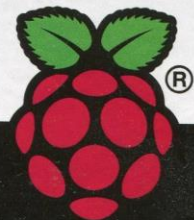
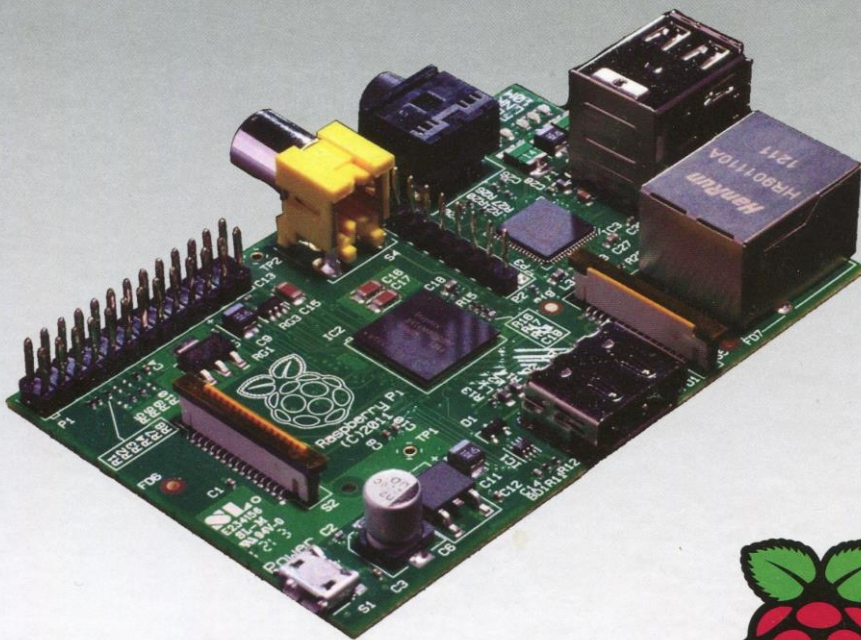


С. Б. Могильний

# МІКРОКОМП'ЮТЕР RASPBERRY PI - ІНСТРУМЕНТ ДОСЛІДНИКА

---



**С.Б.Могильний**

**МІКРОКОМП'ЮТЕР RASPBERRY PI -  
ІНСТРУМЕНТ ДОСЛІДНИКА**

Посібник

Київ-2014

ББК 32.973.26  
УДК 004.382.76  
М74

**Могильний С. Б.**

М74 Мікрокомп'ютер Raspberry Pi – інструмент дослідника : посібник. – К. : «Талком», 2014. – 340 с.  
ISBN 978-617-7133-48-2

Прийміть цей посібник, разом зі своїм Raspberry Pi, як "набір з інформатики". Ви, мабуть, бачили набори з радіотехніки чи механічні конструктори? А тепер маєте можливість вивчити основні принципи комп'ютерної науки, виконуючи вправи одну за однією.

Ця книга для всіх і кожного, хто зацікавлений дізнатися більше про комп'ютери і створення комп'ютерних програм. Якщо у вас немає досвіду з програмування комп'ютера, але хочете його отримати і шукаєте з чого почати, то Raspberry Pi для вас.

**ББК 32.973.26**  
**УДК 004.382.76**

## Короткий зміст

|                                                             |     |
|-------------------------------------------------------------|-----|
| Вступ.....                                                  | 5   |
| 1. Початкові навички .....                                  | 7   |
| 2. Налаштування локальної мережі та доступу в Інтернет..... | 54  |
| 3. Віддалений доступ до Raspberry Pi .....                  | 75  |
| 4. Працюємо з відео .....                                   | 112 |
| 5. Інформаційна безпека .....                               | 134 |
| 6. Програмування для вебу .....                             | 188 |
| 7. Використання GPIO .....                                  | 214 |
| 8. Wolfram і Mathematica та математичні задачі .....        | 246 |
| 9. Проекти.....                                             | 274 |
| 10. Розширення та альтернативи.....                         | 304 |
| 11. Короткі анотації корисних книг.....                     | 333 |
| Рекомендована література .....                              | 338 |

## Зміст

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| <b>Вступ .....</b>                                                  | <b>5</b>   |
| Позначення і термінологія .....                                     | 6          |
| <b>1. Початкові навички .....</b>                                   | <b>7</b>   |
| Raspberry Pi – «ягідка-малина» для творчих.....                     | 7          |
| Як почати роботу з Raspberry Pi .....                               | 9          |
| Корисні команди, які повинен знати кожен користувач.....            | 18         |
| Пошук інформації про систему Raspberry Pi.....                      | 22         |
| Освоюємо текстовий редактор Nano .....                              | 25         |
| Як змінити розкладку клавіатури в Raspbian.....                     | 28         |
| Як змінити розмір шрифту командного рядка .....                     | 32         |
| Використання Advanced Packaging Tool – APT.....                     | 36         |
| Як встановити додатки на Raspberry Pi.....                          | 41         |
| Як автозавантажувати додатки в робочому столі LXDE.....             | 49         |
| Як монтувати/автомонтувати жорсткий диск USB.....                   | 52         |
| <b>2. Налаштування локальної мережі та доступу в Інтернет .....</b> | <b>54</b>  |
| Як надати Raspberry Pi статичну IP-адресу .....                     | 54         |
| Як налаштувати Wi-Fi через командний рядок .....                    | 61         |
| Як з Raspberry Pi зробити маршрутизатор Wi-Fi.....                  | 66         |
| Як виправити помилку “device not managed”.....                      | 72         |
| <b>3. Віддалений доступ до Raspberry Pi .....</b>                   | <b>75</b>  |
| SSH для зручного захищеного доступу до Raspberry Pi.....            | 75         |
| Безпечний обмін файлами з Raspberry Pi .....                        | 84         |
| Доступ до робочого столу RPi з комп’ютера Windows.....              | 90         |
| Налаштування VNC на Raspberry Pi.....                               | 95         |
| Управління Raspberry Pi з iPad.....                                 | 98         |
| Android-клієнти для віддаленого доступу зі смартфона до RPi.....    | 100        |
| Як організувати потокову передачу файлів на смартфон.....           | 104        |
| <b>4. Працюємо з відео .....</b>                                    | <b>112</b> |
| Як встановити та налаштувати модуль камери Raspberry Pi....         | 112        |
| Перегляд відео на OMXPlayer з командного рядка.....                 | 119        |
| Переглядаємо відеопотік з камери через веб-браузер.....             | 123        |

|                                                                 |            |
|-----------------------------------------------------------------|------------|
| Як передати відеопотік на планшет чи смартфон з VLC.....        | 128        |
| Як створити потік з веб-камери на Raspberry Pi з ffmpeg.....    | 131        |
| <b>5. Інформаційна безпека .....</b>                            | <b>134</b> |
| Особистий VPN, щоб безпечно заходити звідусіль в Інтернет.....  | 134        |
| VPN на Raspberry: будуємо сервер.....                           | 141        |
| VPN на Raspberry Pi: створюємо сторону клієнта.....             | 152        |
| Onion Pi: портативний Tor-проксі.....                           | 157        |
| Проксі-сервер для доступу в I2P.....                            | 167        |
| Raspberry Pi для пентестерів.....                               | 175        |
| SSH-приманка: Kippo, Kali і Raspberry Pi.....                   | 183        |
| <b>6. Програмування для вебу .....</b>                          | <b>187</b> |
| Встановлення персонального веб-сервера на Raspberry Pi.....     | 187        |
| Coder - простий спосіб програмувати на Raspberry Pi.....        | 195        |
| Працюємо з HTML в Google Coder.....                             | 196        |
| Додавання зображень та іншої мультимедіа в Google Coder....     | 201        |
| Додавання стилю за допомогою CSS в Coder.....                   | 203        |
| Барви кольорів за допомогою Coder.....                          | 207        |
| Використання сервіса хмарного сховища від Copy.....             | 210        |
| <b>7. Використання GPIO .....</b>                               | <b>213</b> |
| Як використовувати GPIO на Raspberry Pi (Wheezy).....           | 213        |
| Використання температурних датчиків з Raspberry Pi.....         | 219        |
| Ультразвуковий далекомір з використанням Python.....            | 224        |
| Як керувати GPIO з пульта дистанційного керування.....          | 229        |
| Як за допомогою RPi виявити, хто був у твоїй кімнаті.....       | 237        |
| <b>8. Wolfram і Mathematica та математичні задачі .....</b>     | <b>245</b> |
| Мова Wolfram і Mathematica на кожному Raspberry Pi.....         | 245        |
| Основи програмування з Mathematica.....                         | 250        |
| Використання мови Wolfram для GPIO .....                        | 258        |
| Побудова GPS-трекера.....                                       | 260        |
| Система домашньої сигналізації з датчиком руху PIR.....         | 263        |
| Raspberry Pi з датчиками Vernier і мова Wolfram.....            | 264        |
| Розподілені обчислення за допомогою Raspberry Pi.....           | 268        |
| <b>9. Проекти .....</b>                                         | <b>273</b> |
| Робот-фотограф.....                                             | 273        |
| Настінний календар Google .....                                 | 277        |
| Платформа для Minecraft .....                                   | 284        |
| Запустіть свою власну радіостанцію з Raspberry Pi .....         | 287        |
| Перекладач з розпізнаванням мови та відтворенням .....          | 290        |
| <b>10. Розширення та альтернативи .....</b>                     | <b>298</b> |
| Сенсорний екран Adafruit PiTFT.....                             | 298        |
| Комп'ютерний модуль Raspberry Pi.....                           | 300        |
| Розширення Pi Co-op Arduino .....                               | 304        |
| Платформа датчиків e-Health для Arduino і Raspberry Pi.....     | 309        |
| Плата розширення GSM/GPRS і GPS.....                            | 314        |
| Альтернативні мікрокомп'ютери.....                              | 323        |
| <b>Алфавітний вказівник команд та англomовних термінів.....</b> | <b>335</b> |
| <b>Рекомендована література.....</b>                            | <b>338</b> |

*Гарно того вчити,  
хто хоче все знати.*

## **Вступ**

Raspberry Pi (RPi) - комп'ютер, дуже схожий на комп'ютери, з якими ви вже знайомі. Але він використовує інший вид процесора, так що ви не зможете встановити на нього Microsoft Windows. Але зможете встановити кілька версій операційної системи Linux, які виглядають і ведуть себе дуже схоже на Windows. Якщо захочете, то можете використовувати Raspberry Pi для серфінгу в Інтернеті, щоб відправити електронну пошту або написати листа, використовуючи текстовий редактор. **Але ви також можете зробити набагато більше.**

Легкий у використанні, але потужний, недорогий і який (поки ви обережні) важко зламати, **Raspberry Pi є ідеальним інструментом для комп'ютерних дослідників.**

Що ми розуміємо під комп'ютерною наукою? Мається на увазі вивчення, як працюють комп'ютери, щоб ви могли змусити їх зробити те, що ви від них хочете.

А кого ж ми розуміємо під комп'ютерними дослідниками? Ми маємо на увазі вас. Ви можете дочитати цей посібник і вирішити, що хочете бути послідовником Тіма Бернерса Лі, але навіть якщо ні, то сподіваємося, що ви отримаєте задоволення, дізнаючись про щось нове і зумієте відчувати, як працюють комп'ютери. Тому що незалежно від того, що ви надалі робитимете в житті, комп'ютери зобов'язані бути його частиною.

Ви, звичайно, не дізнаєтеся тут, як створити електронну таблицю або презентацію. То вже не комп'ютерна наука - то щось зовсім інше.

Замість цього, **прийміть цей посібник, разом зі своїм Raspberry Pi, як "набір з інформатики".** Ви маєте можливість, виконуючи вправи одну за однією, вивчити основні принципи комп'ютерної науки. І це ваш перший крок на шляху до становлення реального комп'ютерного програміста, розробника ігор і багато чого іншого. Все залежить лише від вас.

В посібнику не вивчається окремо мова програмування, але програмуючи Raspberry Pi ви поступово вивчите основні команди Linux, навчитися адмініструвати операційну систему, налаштовувати комп'ютерні мережі, створювати автоматизовані системи тощо.

**Цей посібник для всіх і кожного,** хто зацікавлений дізнатися більше про комп'ютери і створення комп'ютерних програм. Якщо у вас немає досвіду з програмування комп'ютера, але ви хочете його отримати і шукаєте з чого почати, то Raspberry Pi для вас.

## Позначення і термінологія

Щоб привернути увагу читача до ключових моментів і базових визначень в книжці введений ряд умовних позначень:

- *курсивом* виділені спеціальні терміни та визначення, які розкриваються в даному абзаці і далі використовуються без пояснень;
- **напівжирним шрифтом** позначені важливі моменти в тексті;
- **спеціальним шрифтом** виділені елементи інтерфейсу, утиліти системи, назви вікон, пункти меню;
- **моноширинним шрифтом** виділяються дані, які вводяться з клавіатури, команди користувача, рядки програмного коду, лістинг та назви файлів.

В книжці використовується стандартна термінологія технічних документів, які описують програмне забезпечення. Крім англomовної назви елементів інтерфейсу поруч в дужках часто наведений переклад українською мовою.

В тексті часто зустрічається *послідовність команд*, які користувач повинен послідовно вибрати чередуючи або в меню, або з діалогового вікна.

Черговість вибору (переходів) користувача позначається символом `>`. Наприклад, користувач повинен відкрити програму `modules`, яка знаходиться в каталозі `/proc/asound/`, то шлях буде відображений таким чином: `proc > asound > modules`.

На початку кожного підрозділу під заголовком праворуч знаходиться QR-код (із зображенням малини в центрі). Він дозволяє за допомогою спеціальної програми для читання подібних кодів швидко відкрити на смартфоні або ноутбукі сторінку в Інтернеті, на якій знаходиться наведений в книжці матеріал. Це дозволить читачу скопіювати, а не вводити програмний код, який розглядається в книжці, або переглянути онлайн збільшене зображення.



Окремі невеликі хитрощі та цікаві методи виділені розміщенням ліворуч від тексту стилізованим зображенням малини.

Словосполучення **Raspberry Pi** використовується в тексті також в скороченому вигляді як **RPi** або просто **Pi**.

### Від видавництва

Ваші зауваження, питання та пропозиції надсилайте автору на адресу електронної пошти [raspberry-pi@ukr.net](mailto:raspberry-pi@ukr.net).

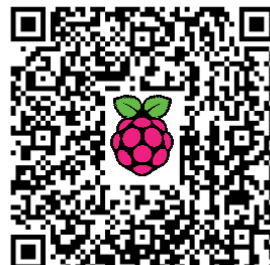
Ми завжди раді знати вашу думку та врахувати її в нашій діяльності

# 1. ПОЧАТКОВІ НАВИЧКИ

## Raspberry Pi – «ягідка-малина» для творчих



Raspberry Pi, крихітний і зовсім недорогий прото-комп'ютер (рис.1.1), що заохочує до дослідження і конструювання, які сьогодні часто проводяться методом «зроби це сам» (Do it yourself - DIY) і спільнотами ентузіастів, вимагає вашої активної участі та інтелектуального напруження.



Тому попереджаю: ви не зможете бути пасивним користувачем. З моменту, як витягнете його з коробки, ви повністю будете йому віддані, і якщо не знатимете, що робити, то вчитиметеся, щоб швидко про все дізнатися. Якщо ви думаєте, що це звучить як розвага для вас і вашої родини, то Raspberry Pi - один з комп'ютерних проєктів, який ви полюбите, «встромивши в нього всі свої зуби».



Рис.1.1 Загальний вигляд плати Raspberry Pi

### Особливості

Перш ніж говорити про те, що можна зробити з Raspberry Pi, ми повинні проаналізувати, що ви отримуєте разом з платою<sup>1</sup>. Система постачається в двох різних варіантах: модель А і модель В (дорожча на \$10 – всього \$35). Основні відмінності між ними: порти USB (модель А

---

<sup>1</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1751-raspberry-pi-lberry-raspberryq-for-creative>

має один, а модель В - два), Ethernet (модель А його не має, а в моделі В є), і споживання енергії (не в останню чергу через те, що є Ethernet, модель В споживає майже в півтора рази більше енергії, ніж модель А). Оперативної пам'яті в моделі В 512 Мбайт, а в моделі А – 256 Мбайт. Більшість інших апаратних параметрів плат однакові.

Raspberry Pi використовує «систему на чіпі» Broadcom BCM2835, яка об'єднує ARM1176JZF-S частотою 700 МГц з процесором Broadcom VideoCore IV GPU і 512 Мбайт SDRAM. Порти USB 2.0 є вашим єдиний варіантом підключення до зовнішніх даних, хоча є два відеовиходи у вигляді композитного RCA і HDMI портів, 3,5 мм аудіо вихід і SD/MMC/SDIO слот для карт пам'яті, що пояснює те, як ви можете отримати доступ до пам'яті. Raspberry Pi живиться від стандартного адаптера Micro USB, з мінімальним струмом 500 мА (2,5 Вт) для моделі А і 700 мА (3,5 Вт) для моделі В, але ви повинні будете придбати його окремо, бо він не постачається в комплекті з Raspberry Pi. (Швидше за все, ви зможете використовувати зарядний пристрій для мобільного телефону або планшета). Розмір плати близько 17x54x86 мНбс/м<sup>3</sup> (ВхШхГ), що не набагато більше від кредитної карти.

Іншими словами, Raspberry Pi базова плата, наскільки базовою така плата може бути. І це найважливіше. Вона не призначена для прямого використання - вона розроблена як платформа: для навчання, для розширення, для технологічних досліджень. І це їй вдається. Але плата «гола від природи» і відсутні навіть рудиментарні функції на додаток до відсутності кабелю живлення і навіть немає кнопки вимикання: під'єднання або від'єднання кабелю мікро USB вмикає або вимикає її - це означає, що вона зовсім не для всіх. Тому підходити до неї треба обережно і бути в курсі проблем, які вона представляє, зокрема, на самому початку.

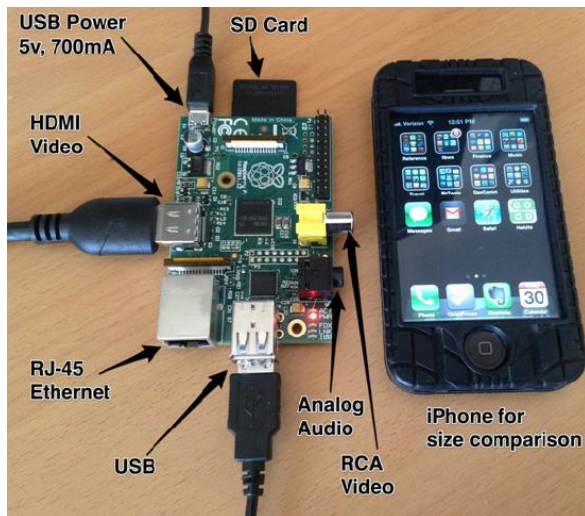


Рис.1.2 Фото підключень до Raspberry Pi

### Встановлення і програмне забезпечення

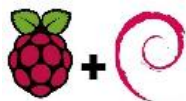
З апаратної точки зору, налаштування Raspberry Pi не може бути простішим. Підключіть USB-клавіатуру і мишу, дисплей, кабель Ethernet (для моделі B), навушники або колонки, якщо хочете, та подайте живлення через кабель мікро USB, і це все (рис.1.2). Кожен, незалежно від досвіду роботи з комп'ютером, зможе зробити це за секунди.

Програмне забезпечення - це інше питання. Тому що немає ніякої постійної пам'яті: ви маєте лише слот для карт пам'яті, що означає розміщення завантажувальної операційної системи, яку ви плануєте використовувати, на карті. Інтерфейс Windows, звичайно, відсутній (він занадто великий і немає версії, яка працює з ARM-чіпами), але вебсайт Raspberry Pi<sup>2</sup> пропонує і надає для скачування шість дистрибутивів Linux, які можна використовувати. Серед них: «вичавлена» Debian (Raspbian - рекомендований вибір), Arch Linux ARM і Pidora - всі оптимізовані для використання з Raspberry Pi. Після вибору однієї з них, вам потрібно записати її (а не просто скопіювати) на карту SD, наприклад, за допомогою таких програм як Win32 Disk Imager. Весь цей процес не буде проблемою для більшості середніх користувачів.

Все може змінитися, як тільки ви увімкнете Raspberry Pi. Коли ви увійдете (з ім'ям користувача та паролем за замовчуванням навіть зможете легко зайти в Інтернет), то, надіюсь, не будете шоковані першим виглядом екрану і командним рядком без курсору миші. Вірте чи ні, але саме тут найпростіше робити більшість налаштувань, але якщо ви не володієте Linux, то можете зайти в глухий кут, намагаючись зробити щось навіть просте, типу змінити час (що повинно бути вашою першою дією - акумулятор для зберігання інформації на Raspberry Pi відсутній). Якщо ви відчайдушно хочете робочий стіл, то ви за кілька натискань клавіш від нього – просто наберіть **startx**, а встановлення нового програмного забезпечення на додаток до мінімалістичних пропозицій вашої операційної системи не набагато складніше.

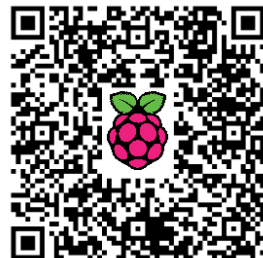
Отож, вперед!

### Як почати роботу з Raspberry Pi



Якщо ви зібрали все необхідне апаратне забезпечення Pi і периферійні пристрої, то прийшов час, щоб розпочати завантаження операційної системи на ваш Pi<sup>3</sup>.

На відміну від традиційного комп'ютера, на якому у вас є BIOS, привід з підтримкою змінних носіїв (наприклад, дисковод DVD), і жорсткий



<sup>2</sup> [www.raspberrypi.org](http://www.raspberrypi.org)

<sup>3</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1753-how-to-get-started-with-raspberry-pi>

диск всередині комп'ютера, Raspberry Pi просто має невеликий твердотільний накопичувач (реалізований через карту SD і SD-кардрідер). Таким чином, ви не будете слідувати традиційному маршруту налаштування комп'ютера, вставивши завантажувальний диск, щоб інсталювати операційну систему на внутрішній пристрій зберігання.

Замість цього, ми збираємося підготувати SD-карту на звичайному комп'ютері та вставити її в Raspberry Pi для подальшого розпакування/налаштування.

## Створення завантажувального образу ОС

**Вибір дистрибутива.** Сьогодні є багато різних дистрибутивів і варіацій, доступних для Raspberry Pi. Дистрибутив, який ми збираємося використовувати, сьогодні є найбільш підтримуваним і найбільш стабільним - Raspbian – дистрибутив Debian, оптимізований для Raspberry Pi.

Як тільки ви завантажите його на Raspberry Pi і запустите, то це буде повністю незалежна машина, але для того, щоб почати процес, вам знадобиться ще один комп'ютер для створення образу операційної системи на карті SD.

По-перше, давайте почнемо з отримання копії Raspbian від Raspberry Pi Foundation<sup>4</sup>. Образ, який ми хочемо, - Raspbian "Wheezy".

*Примітка:* Починаючим користувачам рекомендуємо почати роботу з Pi, скачавши і встановивши за попереднім посиланням нове з коробки програмне забезпечення (New Out of Box Software - NOOBS) на 4 Гб (чи більшу) карту SD. При першому завантаженні Вам буде наданий вибір операційної системи для установки, у тому числі Raspbian, Pidora і два різновиди XBMC. Після установки операційної системи, ви можете повернутися до інтерфейсу NOOBS, утримуючи клавішу Shift під час завантаження, що дозволяє вам перейти на іншу операційну систему, або переписати пошкоджену карту новою установкою поточної.

Типово NOOBS виводить через HDMI на привілейованому дозволі вашого дисплея, навіть якщо дисплей до HDMI не підключений. Якщо ви нічого не побачили на дисплеї через HDMI або при використанні композитного виходу, то натисніть 1, 2, 3 або 4 на вашій клавіатурі, щоб вибрати привілейований режим HDMI, безпечний режим HDMI, композитний режим PAL або композитний NTSC режим, відповідно.

**Вибір інструмента для створення образу диску.** Після того, як ви закінчили завантаження образу диску, треба розпакувати образ на SD-карту. Ми будемо використовувати комп'ютер з Windows, щоб розпакувати образ, але легко виконати ті ж дії на комп'ютерах з Linux і OS X. Наші рекомендовані інструменти для всіх трьох операційних систем:

<sup>4</sup> [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads)

- **Windows:** Win32 Disk Imager<sup>5</sup>
- **Linux:** ImageWriter<sup>6</sup>
- **OS X:** RPi-sd Card Builder<sup>7</sup>

*Примітка:* Ви завжди можете скористатися командою DD на Linux або OS X, щоб записати свій образ. Враховуючи те, наскільки потужним є DD (і як жахливо неправильно все може піти, якщо використати DD неправильно), ми залишаємо це на розсуд користувачів: чи зручно їм скористатися командою DD для створення образу Raspberry Pi на їх SD-картах під Linux/OS X. Інструменти GUI для створення образу диска, які пропонуються, набагато безпечніші для тих, хто недосвідчений у використанні DD.

**Створення образу на SD-карті.** Тепер у нас є копія дистрибутива Raspbian для Raspberry Pi і Win32 Disk Imager, тож можемо приступити до створення диска з нашою операційною системою.

Вставте SD-карту, яку ви хочете використовувати для проекту, в пристрій читання карт SD. Ми рекомендуємо в цей момент зробити подвійну перевірку букви диска у провіднику Windows. Бажано видалити будь-які флеш-носії, флеш-накопичувачі, або прив'язані смартфони/інші пристрої, перш, ніж почати - Win32 Disk Imager не зашкодить жодному підключеному жорсткому диску вашої системи, але він буде пропонувати інші флеш-носії, підключені до комп'ютера, як життєздатну мішень для інсталяції.

Розархівуйте .IMG файл з Raspbian .ZIP файла, який ви завантажили. Зверніть увагу на розташування витягнутого файла. Win32 Disk Imager - це портативний додаток, так що витягніть вміст .ZIP, який ви завантажили за посиланням вище, і запустіть Win32DiskImager.exe.

У запущеному Win32 Disk Imager (рис.1.3) виберіть файл зображення і відповідну літеру.

Натисніть кнопку Write (Записати) для запису вмісту образу Raspbian на карту SD. Win32 Disk Imager попросить вас підтвердити перезапис, тому натисніть кнопку Yes (Так). Сам процес запису має зайняти всього кілька хвилин. Після того, як додаток вкаже на успішний запис, можна безпечно витягти SD-карту з вашої системи.

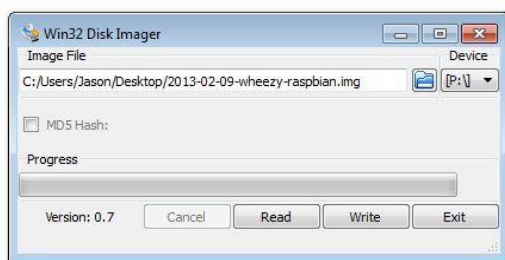


Рис.1.3 Запис образу на SD-карту

<sup>5</sup> <http://sourceforge.net/projects/win32diskimager/>

<sup>6</sup> <http://launchpad.net/usb-imagewriter>

<sup>7</sup> <http://allthware.wordpress.com/2012/12/11/easiest-way-sd-card-setup/>

## Налаштування Raspbian на Pi



Рис.1.4 Завантаження ОС Raspbian

Тепер, коли у нас є все обладнання і правильно організована SD-карта, прийшов час, щоб вперше все завантажити (рис.1.4). Приєднайте всі кабелі і периферійні пристрої до Raspberry Pi, за винятком кабелю живлення: HDMI або RCA кабель, USB-концентратор, кабель Ethernet/Wi-Fi адаптер і т. д.

Як тільки у вас будуть під'єднані всі кабелі, підключені до Pi та його зовнішніх пристроїв, вставте SD-карту. Після того, як карта SD надійно встановлена, вставте MicroUSB кабель живлення, щоб почати завантаження Pi. Нижче наведена послідовність ваших дій (рис.1.5):



Рис.1.5 Послідовність приєднання обладнання

Майже відразу ж побачите процес завантаження, який швидко прокручується – як показано на скріншоті вище. Через хвилину чи дві ваш Raspberry Pi перестрибне далі в Raspi-утиліту конфігурації (рис.1.6):

Тут ми можемо виконати основні завдання налаштування і підвищення функціональності нашого Pi-блоку. Хоча можете запустити інструмент Raspi-config в будь-який час, він ідеально підходить, щоб зробити більшу частину конфігурування і налаштування з самого початку, бо зміни, внесені в систему пізніше, можуть вступати в конфлікт з деякими завданнями конфігурації.



Рис.1.6 Утиліта конфігурування системи  
Raspi-config

**Expand\_rootfs.** Наше перше завдання полягає в запуску `expand_rootfs`. За умовчуванням, Raspbian вимагає використовувати більшу частину Вашої SD-карти тільки в якості основної операційної системи. Ми хочемо отримати доступ до всієї карти SD, бо на ній у нас є багато місця для зберігання майбутніх проєктів.

Використайте клавіші зі стрілками на клавіатурі, щоб вибрати `expand_rootfs` і натисніть ввід - рядки тексту будуть прокручуватися, а потім ви отримаєте повідомлення: "Boot partition has been resized. The filesystem will be enlarged upon next reboot." ("Завантажувальний розділ був змінений. Файлова система буде збільшена при наступному перезавантаженні."). Натисніть кнопку OK.

**Масштабування зображення (overscan).** Не кожному користувачу треба буде налаштовувати масштабування зображення. Якщо ви помітили, що існує значний чорний простір на краях дисплея (Pi використовує середню частину екрану, а не весь екран), то ви, можливо, побажаєте включити масштабування зображення, щоб збільшити площу екрану, яку Pi використовує. Натисніть на масштабування екрану (overscan), а потім виберіть Дозволити (enable).

**міна часового поясу.** Ви хочете, щоб ваш Pi зберігав точний час? Виберіть `change_timezone`, а потім виберіть відповідний географічний регіон (наприклад, US), а потім - відповідний часовий пояс в межах цього регіону (як Eastern). Буде запущений невеликий сценарій, який виглядає як кілька рядків у нижній частині екрана, а потім повернетеся в головне меню.

**SSH.** Можливість віддалено підключатися до вашому Raspberry Pi через мережу, використовуючи SSH, дійсно зручна. Знайдіть хвилинку,

щоб включити сервер SSH, вибравши `ssh`. Далі виберіть Увімкнути (`enable`). Натисніть OK, після чого `Raspi-config` покаже, що сервер був включений.

**Поведінка при завантаженні (`Boot_behavior`).** Тут можна вказати: хочемо ми, щоб `Raspbian` завантажувався у вигляді командного рядка чи з оточенням робочого столу, коли завантажується `Raspberry Pi`. Давайте перейдемо до завантаження робочого столу, вибравши `boot_behavior`, потім Так (`Yes`) у відповідь на "Чи повинні ми завантажуватися безпосередньо з робочим столом?" ("Should we boot straight to the desktop?").

У той час, як ці основні хитрощі повинні охоплювати потреби більшості користувачів, є кілька додаткових налаштувань, які варто відзначити.

**Конфігурування клавіатури (`configure_keyboard`).** Використайте цю команду, щоб налаштувати неамериканську клавіатуру і забезпечити її набори символів/літер.

**Зміна паролю (`Change_pass`).** Дозволяє змінити пароль за замовчуванням `'raspberry'` на що забажаєте.

**Зміна локалізації (`Change_locale`).** Включення мови для операційної системи, яке необхідне для тих, хто не володіє англійською мовою.

**Розділення пам'яті (`Memory_split`).** Зміна способу, яким система виділяє пам'ять, що розділяється між GPU і головним процесором. Ми пропонуємо залишити це в спокої: якщо пізніше у ваших експериментах з Pi виникне нагальна необхідність, то зможете змінити.

**Розгін (`Overclock`).** Pi використовує 700 МГц процесор ARM. `Raspberry Pi Foundation` дійсно підтримує розгін Pi до 1000 MHz (1 ГГц), але рекомендуємо залишити на даний момент все за замовчуванням. Таким чином, якщо виникнуть питання при вашому початковому налаштуванні, то не треба буде виключати налаштування розгону, як причину.

**Оновлення (`Update`).** Це просто невеликий сценарій для оновлення фактичного інструменту `Raspi-config`. Так як ви тільки що скачали свіжий образ `Raspbian`, то немає особливого сенсу, але це треба мати на увазі, бо ви повинні будете переходити в інструмент `Raspi-config` в майбутньому.

Коли ви зробите всі налаштування параметрів конфігурації, то перейдіть стрілкою вниз і виберіть `Finish` (Готово). При появі запиту на перезавантаження пристрою, виберіть `Yes` (Так). Зачекайте, поки прокрутиться послідовність завантаження.

У цей момент ви завантажите в Raspbian робочий стіл, або, більш технічно, LXDE, Lightweight X11 Desktop Environment - де нам потрібно буде зробити кілька останніх налаштувань конфігурації. Нагадаємо, що для переходу до робочого столу використовується команда `startx`.

**Налагодження підключення до мережі.** Коли ви перебуваєте на робочому столі, то повинні налаштувати Wi-Fi адаптер (ті, хто під'єднаний до Ethernet за допомогою прямого кабелю, пропустіть це і перейдіть до наступного кроку).

На робочому столі двічі клацніть на значок WiFi Config. Це відкриє інструмент `wpa_config`, як показано вище (рис.1.7). Натисніть на кнопку Scan (Сканування) в правому нижньому кутку вікна.

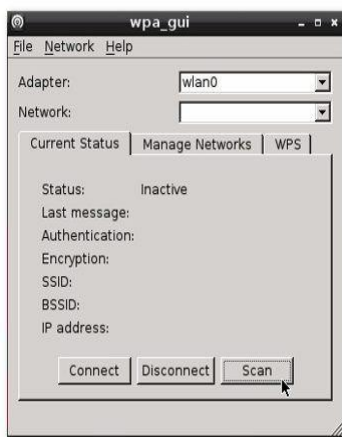


Рис.1.7 Інструмент `wpa_config`

Після того, як додаток просканує доступні Wi-Fi мережі, двічі клацніть на мережі, до якої треба підключитися (рис.1.8).

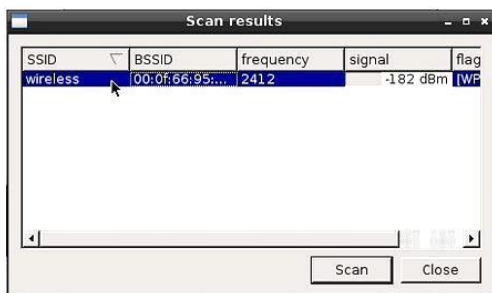


Рис.1.8 Результати пошуку мережі Wi-Fi

У детальному описі цієї мережі введіть мережевий пароль у слоті PSK. Натисніть кнопку Add (Додати) в нижній частині вікна (рис.1.9). Ви

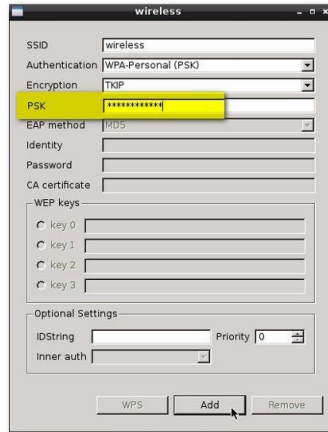


Рис.1.9 Введення пароля доступу

повернетеся до вікна результатів сканування, йдіть вперед і закрийте як вікно результатів сканування, так і оригінальне вікно `wpa_gui`.

**Тестування мережі.** Тепер, коли ми налаштували Wi-Fi з'єднання (або перейшли прямо в цей розділ, тому що використовуємо Ethernet), то настав час для перевірки нашого зв'язку. Що може бути кращим для перевірки з'єднання, ніж запустити браузер і перейти до `isearch.kiev.ua`?

На робочому столі натисніть на значок Midori. Midori - це ультралегкий веб-браузер - і введіть адресу `isearch.kiev.ua`:



Рис.1.10 Вікно браузера Midori

**Успіх!** Мало того, що у нас є підключення до мережі, але двигун рендерингу WebKit Midori робить вигляд сайту `isearch.kiev.ua` (рис.1.10) таким самим чудовим на легкому Pi, як на повноцінному десктопному комп'ютері. Це, ймовірно, буде перше з багаторазових ваших здивувань і задоволення, наскільки здатний ваш новий маленький мікрокомп'ютер.

**Оновлення програмного забезпечення.** Перш, ніж почати копатися в Вашому Pi, хорошою ідеєю буде зробити основні оновлення програмного забезпечення. Ми налаштували мережу, перевірили зв'язок, і тепер маємо зручний час, щоб зробити загальносистемні оновлення програмного забезпечення.

Закрийте Midori і на робочому столі клацніть двічі на LXTerminal, щоб відкрити командний рядок. Введіть таку команду (рис.1.11):

```
sudo apt-get update && sudo apt-get upgrade
```

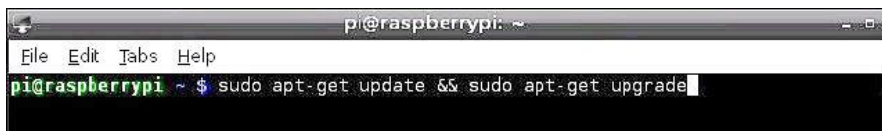


Рис.1.11 Початкові команди LXTerminal

Ця комбінація команд наказує Raspbian шукати доступні репозиторії для оновлення та апгрейду системи і програмного забезпечення. Якщо які-небудь оновлення будуть виявлені, то вам буде запропоновано підтвердити або відхилити зміни клавішами Y/N (рис.1.12):

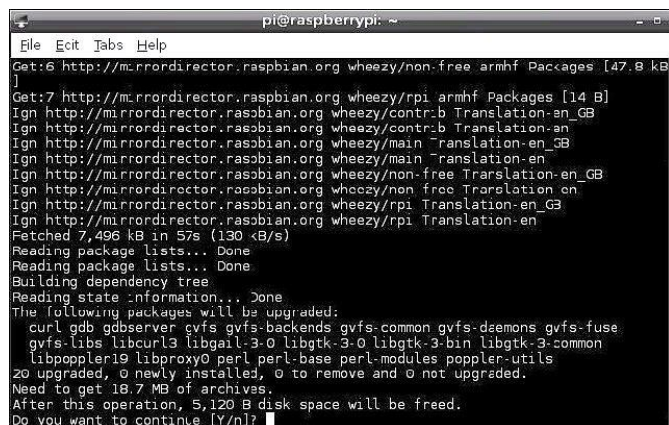


Рис.1.12 Підтвердження оновлення

Будьте терплячі, особливо якщо оновлюєтеся через Wi-Fi, так як там можуть бути періоди часу, коли весь процес зависає на декілька секунд. Після того, як ви дочекалися рядка введення Y/N, перейдіть випити чашечку кави, поки розпакуються і встановляться всі ваші оновлення. Після закінчення оновлення ви повернетеся до командного рядка, перейдіть далі і закрийте вікно терміналу.

Натисніть на значок живлення в нижньому правому куті робочого столу і виберіть з доступних функцій Reboot (Перезавантаження) (рис.1.13).



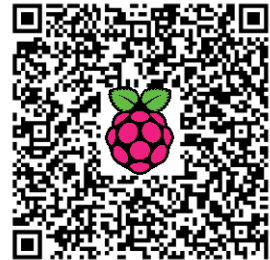
Рис.1.13 Вибір функцій завершення сесії

Дозвольте перебрати цикл послідовності завантаження, а потім знову поверніться назад в робочий стіл LXDE. На цей момент ви успішно налаштували Raspberry Pi для запуску Raspbian, під'єднали його до мережі, оновили програмне забезпечення і тепер готові, щоб почати гратися зі своїм Pi.

## Корисні команди, які повинен знати кожен користувач



Якщо ви використовуєте комп'ютер Raspberry Pi для свого проекту (можливо, медіа-центру або домашнього сервера), то маєте хороший шанс заощадити деякий час, використовуючи ці корисні інструкції командного рядка<sup>8</sup>.



### Raspberry Pi: так, це Linux

Ви записали образ на свою карту SD і завантажили Raspberry Pi. Ви, напевно, використовуєте операційну систему Raspbian і оновлюєте та налаштовуєте її так, щоб вона відповідала вашим вимогам. Те, що ви, можливо, не зрозуміли, що, незважаючи на стиль Windows іконок на робочому столі, Raspbian є дистрибутивом Linux. Для Raspberry Pi доступні кілька операційних систем, переважна більшість з яких є Linux.

<sup>8</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1815-15-useful-commands-raspberry-pi-which-every-user-should-know>

Замість того, щоб люди робили спроби отримати для використання Linux крадькома, Raspberry Pi покладається на ці операційні системи через їх відкритий вихідний код і універсальність.

І тому це досить добре!

Тепер, ймовірно, можна використовувати операційну систему Linux з графічним інтерфейсом користувача без використання командного рядка, але в командному рядку знаходиться більша частина реальної потужності.

Хочете мати владу над своїм Raspberry Pi з Raspbian? Почніть з запуску LXTerminal (рис.1.14).

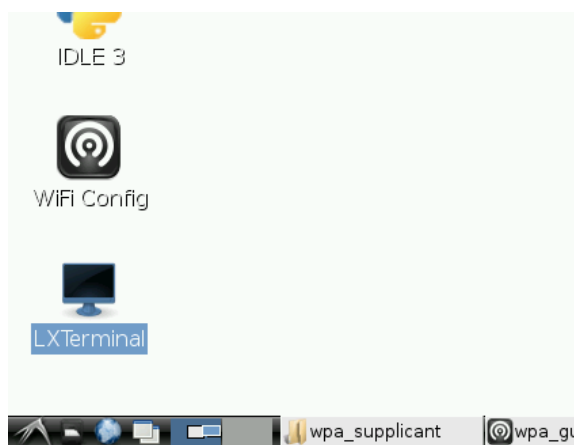


Рис.1.14 Запуск LXTerminal

### Основи командного рядка Raspberry Pi

Ми не очікуємо, щоб ви почнете з використання командного рядка, не знаючи, як він працює. По суті, це метод для видачі інструкцій комп'ютеру на виконання завдання. Реально це не все, чим він відрізняється від звичного «вибрати і натиснути», але в ньому ви повинні використовувати текст.

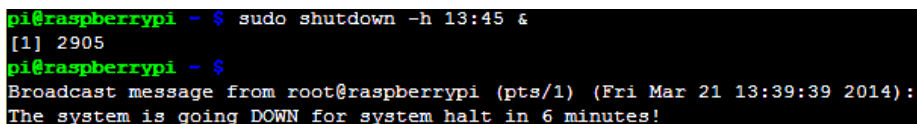
Коли ви вперше отримали доступ до командного рядка, то побачите підказку `pi@raspberrypi $` - щоразу, коли це відображається, ви зможете вводити команди.

З GUI під управлінням мишею ви можете легко перемикатися між каталогами і читати їх вміст. У командному рядку, ви можете перевірити, який каталог переглядаєте, ввівши `pwd` (print working directory - роздрукувати робочий каталог). Використовуйте `ls`, щоб побачити вміст каталогу, а для зміни каталогу введіть `cd`. Наприклад, `cd edward` перемкне на папку з назвою "Edward", тоді як `cd..` завжди буде повертати фокус в батьківський каталог. Нові каталоги можна створити з `mkdir newdir`, де "newdir" буде міткою (назвою) каталогу. Ви також можете створити спадкоємність нових каталогів з `mkdir -p`



Ви можете перезавантажити з `sudo shutdown -r now`. Час виключення може бути встановлений як інтервал, наприклад, 5 minutes, або задати годину доби, наприклад, 13:45 (рис.1.16):

```
sudo shutdown -h 13:45
```



```
pi@raspberrypi ~$ sudo shutdown -h 13:45 &
[1] 2905
pi@raspberrypi ~$
Broadcast message from root@raspberrypi (pts/1) (Fri Mar 21 13:39:39 2014):
The system is going DOWN for system halt in 6 minutes!
```

Рис.1.16 Завершення роботи системи у визначений час

### Інформація командного рядка - це потужність

Для роботи в режимі командного рядка також будуть корисними такі команди:

- **man** (manual) **команда** - видає докладну допомогу для команди;
- **ls** (list) - список файлів;
  - `ls -lha` - показує всі файли (включаючи приховані), розмір файлів, права на них, дату останньої зміни;
  - `ls -lha | less` - дозволить переглядати файли посторінково (якщо їх багато);
- **mv** (move) - перейменувати, перенести;
- **cp** (copy) - скопіювати;
- **rm** (remove) - видалити;
- **>** - очищення файлу. Цю команду можна застосовувати до лог-файлів. Якщо почистити файли з директорії `/site/Ваш_домен/log/` наступним чином:

```
> access.log
> error.log
> combined.log
```

то очистяться логи, а от їх видалення, як і видалення інших службових файлів, може призвести до припинення роботи вашого сайту.
- **du -h --max-depth=1**. Так можна подивитися, скільки місця використовується під сайт. Точка означає поточну директорію, замість неї можна підставити ім'я будь-якої цікавої для вас директорії.

Багатьох людей доступ через командний рядок на будь-якій платформі лякає.

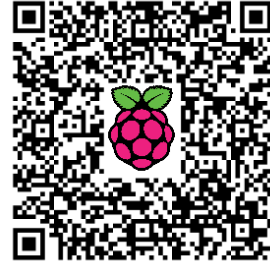
Корисні команди, перераховані вище, є спроба дати новачкам Raspberry Pi «голий» мінімум, щоб вони могли почати роботу з терміналом. Це невелика сходинка до успіху проекту на Pi, з якого вони вирішили стартувати.

## Пошук інформації про систему Raspberry Pi



Raspberry Pi має багато доступної інформації про систему<sup>10</sup>: докладну інформацію про процесор, поточну температуру процесора, обсяг пам'яті і т. д. Не вся інформація

доступна в одному місці, але якщо знаєте, де шукати, то можете виявити досить багато цікавого про свій Pi.



Крім таких «стандартних» інструментів системних ресурсів, як `"ps"`, `"df"`, `"top"` та інших таких корисних команд, як `"htop"`, `"iotop"` і `"glances"`<sup>11</sup>, інформація про систему може бути знайдена в файловій системі `"/proc"`. Одним з найбільш корисних є файл `"cpuinfo"`, який містить дані про процесор системи. Щоб побачити їх введіть:

```
cat /proc/cpuinfo
```

Результат (рис.1.17) говорить нам про три речі Raspberry Pi: він має процесор на базі архітектури ARM (а не Intel, як і в Windows-ПК і комп'ютерів Mac), процесор використовує набір команд ARMv6 і, що процесором є BCM2708, який, як ми знаємо, від Broadcom. Набір команд ARMv6 є однією зі старих розробок ARM. Більшість сучасних смартфонів і планшетів використовують архітектуру ARMv7, і стає все більш поширеними новий набір команд 64-розрядного ARMv8.

```
pi@raspberrypi ~ $ cat /proc/cpuinfo
Processor       : ARMv6-compatible processor rev 7 (v6l)
BogoMIPS       : 697.95
Features        : swp half thumb fastmult vfp edsp java tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xb76
CPU revision    : 7

Hardware       : BCM2708
Revision       : 000e
Serial         : 00000000a37c4d6f
pi@raspberrypi ~ $
```

Рис.1.17 Інформація про процесор системи

Інші файли, які варто подивитися в файловій системі `"/proc"`, включають `"/proc/meminfo"`, `"/proc/partitions"` і `"/proc/version"`. Кожен з них може бути досліджений за допомогою команди `"cat"`:

<sup>10</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1809-finding-your-raspberry-pis-system-information>

<sup>11</sup> <http://www.maketecheasier.com/three-system-monitoring-tools-supplement-top/>

- `cat /proc/meminfo` відображає докладну інформацію про пам'ять
- `cat /proc/partitions` показує розмір і кількість розділів на карті SD або HDD
- `cat /proc/version` показує використану версію Pi (рис.1.18)

```
pi@raspberrypi ~ $ cat /proc/version
Linux version 3.6.11+ (dc4@dc4-arm-01) (gcc version 4.7.2 20120731 (prerelease)
(crosstool-NG linaro-1.13.1+bzr2458 - Linaro GCC 2012.08) ) #474 PREEMPT Thu Jun
13 17:14:42 BST 2013
```

Рис.1.18 Інформація про версію системи

Інформація, що міститься в розділі `"/proc"`, доступна на всіх системах Linux, однак, в Raspberry Pi також доступна спеціальна команда, яка відображає інформацію, що відноситься до плати Raspberry Pi. Інструмент `"vcgencmd"` може отримати доступ до всієї конкретної інформації про Raspberry Pi, включаючи тактові частоти, різні напруги, температуру процесора і які апаратні кодеки доступні.

Почнемо з температури процесорного ядра, ввівши:

```
vcgencmd measure_temp
```

Результатом буде один рядок звіту про температуру (рис.1.19):  
`temp=51.9'C`

```
pi@raspberrypi ~ $ vcgencmd measure_temp
temp=51.9'C
```

Рис.1.19 Інформація про температуру ядра процесора

Це число має важливе значення для тих, хто розігнав процесор свого Pi або хто побудував конструкцію навколо Raspberry Pi, обмеживши потік повітря над процесором.

Коли говоримо про розгін, щоб побачити поточну частоту процесора разом з мінімальною і максимальною частотами, використайте (рис.1.20):

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
```

```
pi@raspberrypi ~ $ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
700000
pi@raspberrypi ~ $ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
700000
pi@raspberrypi ~ $ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
950000
pi@raspberrypi ~ $
```

Рис.1.20 Отримання інформації про частоти

Результатом будуть числа в кілогерцах, тобто, 950000 є 950МГц.

Інші тактові частоти також можуть бути виявлені за допомогою `"vcgencmd measure_clock CLOCKNAME"`, де `CLOCKNAME` є одним з `h264`, `isp`, `v3d`, `uart`, `pwm`, `emmc`, `pixel`, `vec`, `hdmi` або `dpi`. Наприклад:

```
vcgencmd measure_clock arm
```

Результат в герцах, тому 700000000 є 700 МГц.

```
frequency(45)=700000000
```

Ви можете використати наступний сценарій оболонки, щоб отримати список всіх тактових частот:

```
for src in arm core h264 isp v3d uart pwm emmc pixel vec hdmi
dpi ; do \
echo -e "$src:\t$(vcgencmd measure_clock $src)" ; \
done
```

Інша системна команда `"vcgencmd"`, яка може вам знадобитися, `"vcgencmd measure_volts"`, щоб знайти внутрішні напруги для ядра, `sdram_c`, `sdram_i` і `sdram_p`.

Наприклад:

```
vcgencmd measure_volts core
```

Результат буде виглядати приблизно так: `volt=1.20V`

Наступні команди оболонки покажуть всі напруги:

```
for id in core sdram_c sdram_i sdram_p ; do \
echo -e "$id:\t$(vcgencmd measure_volts $id)" ; \
done
```

Щоб дізнатися, які апаратні кодеки доступні, скористайтеся `"vcgencmd codec_enabled CODECNAME"`, де `CODECNAME` є одним з `H264`, `MPG2`, `WVC1`, `MPG4`, `MJPG` або `WMV9`. Наприклад:

```
vcgencmd codec_enabled H264
```

Щоб заощадити час і не повторювати команду для кожного кодека, можете використати цей простий сценарій оболонки:

```
for codec in H264 MPG2 WVC1 MPG4 MJPG WMV9 ; do \
echo -e "$codec:\t$(vcgencmd codec_enabled $codec)" ; \
done
```

Щоб побачити, як розділяється пам'ять між процесором і GPU, скористайтесь (рис.1.21):

```
pi@raspberrypi /proc $ vcgencmd get_mem arm && vcgencmd get_mem gpu
arm=448M
gpu=64M
pi@raspberrypi /proc $
```

Рис.1.21 Розподіл пам'яті між процесором і відеографікою

Нарешті, щоб побачити, скільки вільної пам'яті доступно для використання системою:

```
free -o -h
```

Як і для файлової структури, команду `ls` можна використовувати до списку елементів, прикріплених до вашого Raspberry Pi, особливо USB-обладнання. Використовуйте `lsusb` (рис.1.22), щоб відобразити список підключених пристроїв - ви знайдете в цьому вирішальні значення для створення бездротової мережі, монтажу жорсткого диска або при установці будь-якого іншого обладнання USB, яке вимагає певної конфігурації.

```
pi@raspberrypi ~ $ lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port HUB
Bus 001 Device 005: ID 14f3:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 001 Device 006: ID 046d:c404 Logitech, Inc. TrackMan Wheel
Bus 001 Device 007: ID 045e:00dd Microsoft Corp. Comfort Curve Keyboard 2000 V1.0
```

Рис.1.22 Список підключених пристроїв

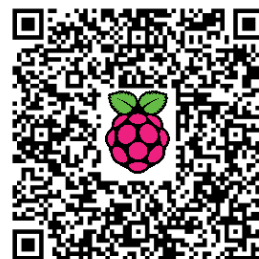
Лише, поки пристрій перерахований тут, ви будете в змозі його налаштувати.

## Освоюємо текстовий редактор Nano на Raspberry Pi



Рутинною частиною управління системою Raspberry Pi, як й інших ОС на основі Linux, є редагування текстових файлів<sup>12</sup>. Багато різних служб та адміністративних параметрів налаштовуються за допомогою текстових файлів.

Від установки IP-адреси до конфігурування таких сервісів, як Samba, майже завжди необхідне налаштування файлу конфігурації.



<sup>12</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1803-master-text-editor-nano-on-raspberry-pi>

Є багато різних редакторів тексту, доступних на Linux, найбільш популярними з яких є nano, vi і emacs. Кожен з них має як сильні так і слабкі сторони. Emacs - потужний редактор з крутою кривою навчання. Vi (або vim, VI-клон з різними поліпшеннями) є на майже кожному Linux і на інсталяціях Unix-типу (в тому числі, FreeBSD і Solaris), часто, коли ніякі інші редактори не встановлені. Проте, Vi не є інтуїтивно зрозумілим, поки не вивчите деякі з основних ідей, що лежать в його побудові. Vi не завжди поводить себе як "нормальний" текстовий редактор. Наприклад, щоб вставити текст, ви повинні спочатку натиснути 'i' і так далі.

Nano - це простий текстовий редактор, який працює так, як і слід було очікувати. Без особливої підготовки ви можете редагувати прості файли і зберегти їх. Він має цілий ряд різних функцій редагування, включаючи пошук і вирізати/вставити.

Щоб змінити існуючий файл або створити новий, запустіть nano і вкажіть ім'я файлу як перший параметр:

```
nano filetoedit.txt
```

Взагалі, для створення текстового файлу можна також скористатися одним з інших 6 найпопулярніших способів, а потім вже редагувати його за допомогою nano:

1. `touch ім'я_файла` - створити пустий файл
2. `cat > ім'я_файла` - і введення закінчити `ctrl+z`
3. `vim ім'я_файла` - створити або редагувати `ім'я_файла`
4. `echo "текст" > ім'я_файла`
5. `> ім'я_файла` - мабуть, найкоротший спосіб
6. `cp ім'я_файла _0 ім'я_файла` - скопіювати або створити файл

Усередині nano основних можливостей дуже багато, як і можна було очікувати. Введіть текст, щоб вставити його в файл, натисніть клавішу ENTER, щоб отримати новий рядок, використовуйте Backspace для видалення тексту і переміщуйтеся за допомогою клавіш курсору.

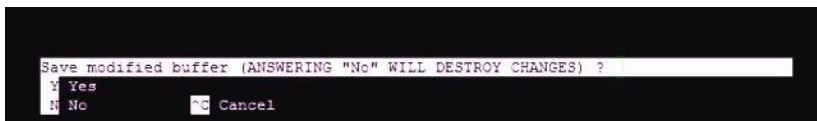


Рис.1.23 Збереження редагованого файлу

Щоб зберегти зміни та вийти, натисніть клавішу Control (CTRL) разом з клавішею X (тобто, CTRL+X). Вам буде запропоновано, якщо хочете, зберегти змінений буфер (рис.1.23). Натисніть клавішу Y. Вам буде запропоновано підтвердити ім'я файлу. Якщо хочете виконати "Зберегти як", то повинні ввести нове ім'я, або натисніть клавішу ENTER, щоб прийняти поточне ім'я файлу.

При редагуванні файлу можете зберегти його на диск і продовжити редагування, натиснувши CTRL+O. Як і при CTRL+X, можете виконати "Зберегти як", ввівши нове ім'я, або натиснути Enter, щоб прийняти поточне ім'я файлу (рис.1.24).



Рис.1.24 Введення імені файлу

Щоб знайти в поточному файлі, натисніть CTRL+W, введіть бажаний текст (рис.1.25) і натисніть ENTER. Щоб побачити наступне входження цього тексту натисніть ALT+W. Якщо хочете виконати пошук і заміну, натисніть ALT+R, введіть текст, щоб знайти, і натисніть клавішу ENTER. Тепер введіть текст для заміни і натисніть ENTER ще раз. Після цього у вас буде можливість замінити всі входження тексту (натисніть A) або просто замінити поточне входження (натисніть Y). Ви можете використовувати CTRL+C в будь-який час, щоб скасувати пошук і заміни.

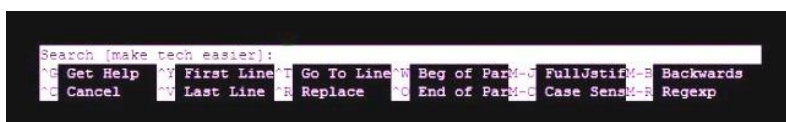


Рис.1.25 Введення тексту для пошуку

Щоб перемістити рядок з одного місця у файлі на інше (тобто вирізати і вставити), натисніть клавіші CTRL+K, щоб вирізати рядок. Перемістіть курсор, а потім натисніть Ctrl+U, щоб вставити рядок. Якщо хочете перемістити кілька рядків, натискайте CTRL+K стільки, поки всі рядки, які бажаєте, будуть вирізані. Перемістіть курсор, а потім натисніть Ctrl+U, щоб вставити всі рядки, які вирізали.

Процедура для копіювання і вставки однакова з вирізанням, але ви використовуєте ALT+B, щоб скопіювати рядки, а потім CTRL+U, щоб вставити їх.

Щоб перейти до першої рядку файлу, натисніть ALT+/, а щоб перейти до останнього рядка, натисніть ALT+/. Щоб перейти до певного рядка, натисніть CTRL+\_, а потім введіть потрібний номер рядка, а далі ENTER.

Ось короткий огляд найбільш корисних операцій в nano:

- CTRL+X – Зберегти і вийти
- CTRL+O – Зберегти файл
- CTRL+W, ALT+W – Знайти і повторити пошук
- ALT+R – Знайти і замінити
- CTRL+K, CTRL+U – Вирізати і вставити

- ALT+6, CTRL+U – Копіювати і вставити
- ALT+\, ALT+/ – Наверх і вниз файлу
- CTRL+\_ – Перехід до певного рядка

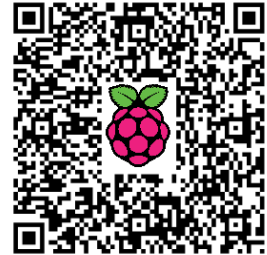
Nano також має вбудовану сторінку допомоги, до якої можна отримати доступ, натиснувши CTRL+G або F1. Наприклад, сторінка допомоги надасть необхідну інформацію про комбінацію клавіш, необхідних для будь-якої конкретної операції.

## Як змінити розкладку клавіатури в Raspbian



Образ операційної системи Raspbian для Raspberry Pi поставляється за замовчуванням з розкладкою клавіатури Великобританії, а це може створити незручності при введенні спеціальних символів.

Є кілька методів<sup>13</sup> для зміни розкладки клавіатури: деякі сподобаються одним користувачам, а деякі іншим. Ви можете спробувати кожен з методів, щоб обрати до вподоби.



### Метод 1

Raspbian є де-факто Debian, тому розкладку клавіатури можна легко змінити, виконавши наступну команду від `root` або через `sudo`:

```
sudo dpkg-reconfigure keyboard-configuration
```

Вам буде запропонований список розкладок клавіатури, виберіть тип клавіатури і мову.

Після цього не обов'язковою, але настійно рекомендується запустити `setupcon` (від `root` або через `sudo`), щоб прискорити процес завантаження операційної системи:

```
sudo setupcon
```

Перезавантажте пристрій і отримаєте нову розкладку клавіатури. Якщо ж виникли проблеми, то, можливо, доведеться встановити пакет `keyboard-configuration` перш, ніж виконувати наведені вище команди:

```
sudo apt-get install keyboard-configuration
```

<sup>13</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1833-how-to-change-the-keyboard-layout-in-raspbian>

### Метод 2

Скористайтеся вже відомою вам командою для налаштувань Raspberry Pi `sudo raspi-config` (рис.1.26):

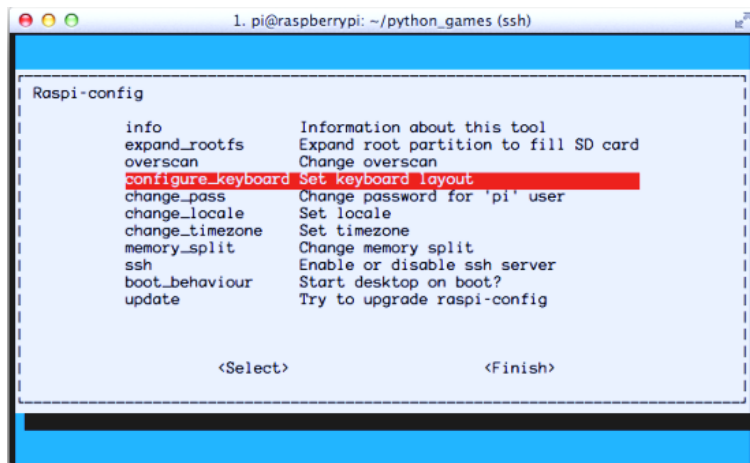


Рис.1.26 Вікно утиліти raspi-config

Відкриється вікно налаштувань, виберіть опцію `configure_keyboard`. Далі, наприклад, вибираєте свою Logitech Generic Keyboard (рис.1.27):

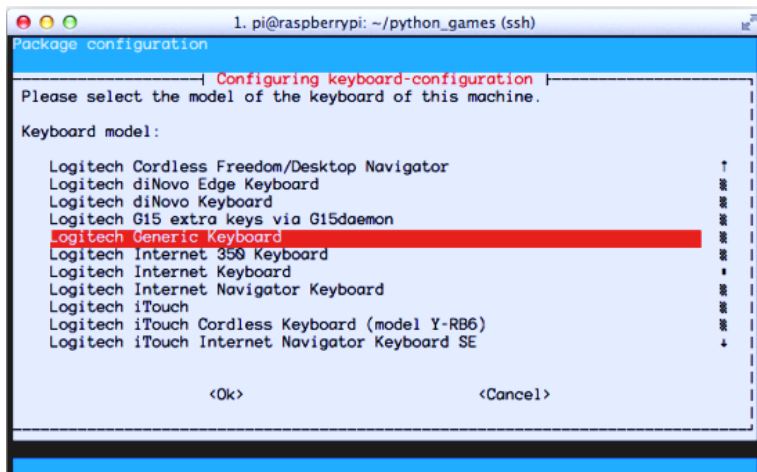


Рис.1.27 Вибір типу клавіатури

Зверніть увагу, що всі опції для розкладки "англійська (UK)". Це за замовчуванням, тому що Raspberry Pi був розроблений (і виробляється) в Англії. Вибираєте опцію Other (рис.1.28).

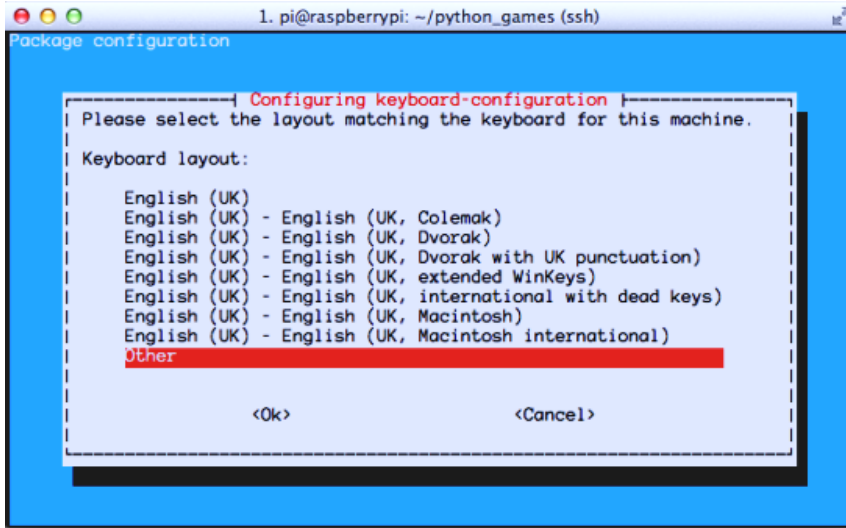


Рис.1.28 Вибір розкладки клавіатури

Потім вибираєте, наприклад, «English (US)» для країни походження (рис. 1.29):

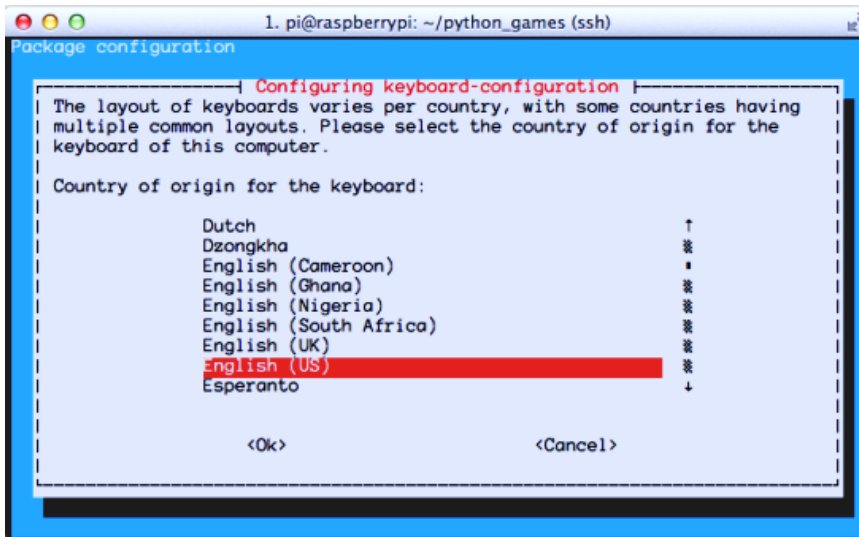


Рис.1.29 Вибір варіанту для країни походження

Нарешті нормальні (і не дуже нормальні) доступні клавіатури США (рис.1.30):

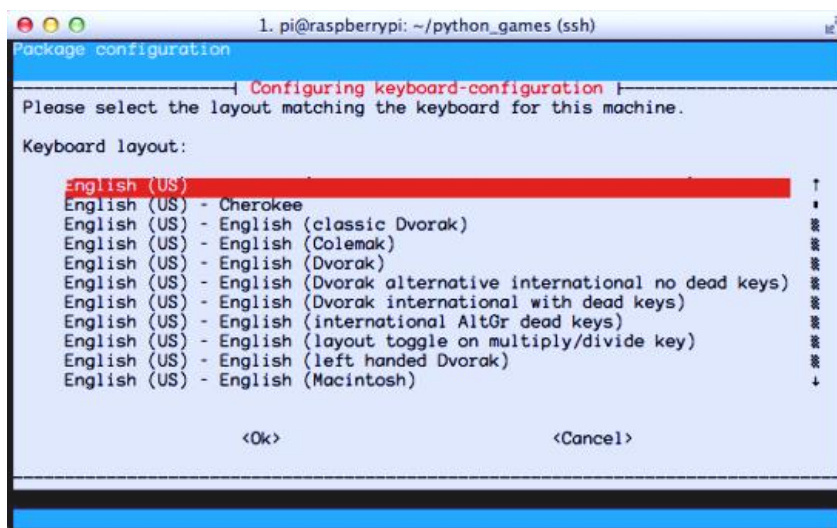


Рис.1.30 Вибір доступної клавіатури

### Метод 3

Зробити необхідні налаштування можна також шляхом редагування файлу конфігурації клавіатури, ввівши з командного рядка:

```
sudo nano /etc/default/keyboard
```

Потім знайдіть, де записано:

```
XKBLAYOUT="gb"
```

Змініть `gb` на двосимвольний код своєї країни чи мову за замовчуванням. Наприклад, змініть `"gb"` на `"us"`.

Збережіть зміни і вийдіть з редактора.

Перезавантажтеся:

```
sudo shutdown -r now
```

Даний метод дозволяє встановити кілька розкладок, які легко змінювати через комбінацію клавіш. Ось мій зміст файла `/etc/default/keyboard`:

```
XKBMODEL="pc105"  
XKBLAYOUT="us,ru,ua"  
XKBVARIANT=","  
XKBOPTIONS="grp:ctrl_shift_toggle, grp_led:scroll"
```

Як видно, використані три альтернативні розкладки клавіатури, які послідовно перемикаються комбінацією клавіш Ctrl+Shift. Якщо ви звикли до комбінації Alt+Shift, то змініть `ctrl_shift_toggle` на `alt_shift_toggle`.

#### Метод 4

Спочатку створіть файл за допомогою `nano`:

```
sudo nano /home/pi/.xsessionrc
```

і додайте такий рядок в цей файл:

```
setxkbmap us
```

Зверніть увагу, що "us" - це назва розкладки для мови, яку хочете використовувати. Наприклад, «us» означає американську англійську мову, "fi" означає фінську, "ua" означає українську і т.д.

Якщо хочете мати кілька альтернативних розкладок, які перемикаються, то додайте такий рядок:

```
setxkbmap -layout 'us,ru,ua' -option  
'grp:ctrl_shift_toggle,grp_led:scroll'
```

Звичайно, використовувати командний рядок для налаштування розкладки клавіатури можна і через SSH.

### Як змінити розмір шрифту командного рядка

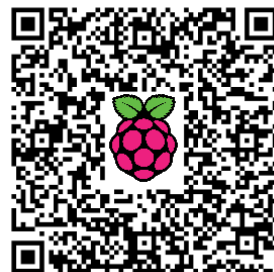


Якщо використовуєте свій Raspberry Pi з меншим екраном, то можете змінити шрифт, який використовується в командному рядку, для більш зручного читання<sup>14</sup>.

Автор зазвичай робить це, коли використовує різні портативні LCD-екрани (наприклад, HDMI Pi).

Це змінює шрифт тільки в консолі, якщо ви використовуєте екран підключений безпосередньо до Pi, і не впливає на сесії Putty/SSH.

Змінити розмір шрифту в консолі легко, і є два методи, з яких можна вибрати.



<sup>14</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1829-how-to-change-the-command-line-font-size>

### Метод 1

Цей метод використовує системну утиліту під назвою "dpkg-reconfigure". Запустіть її за допомогою команди:

```
sudo dpkg-reconfigure console-setup
```

Запускається екран "Package Configuration" ("Конфігурація пакету"). Використовуючи стрілки вгору/вниз виберіть "UTF-8" (рис.1.31). Використовуючи клавішу зі стрілкою вправо виберіть "OK" і натисніть ENTER.

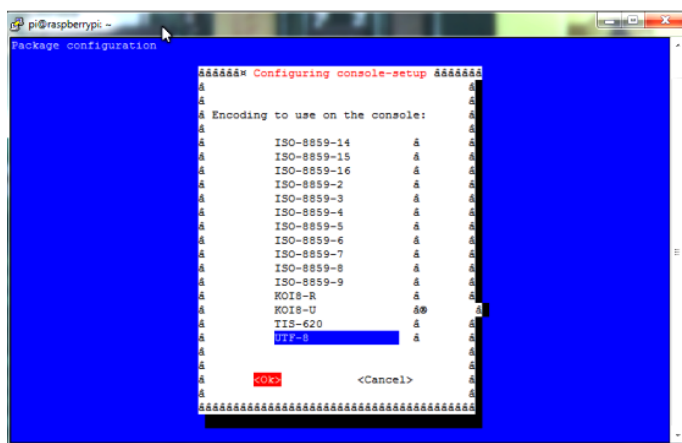


Рис.1.31. Екран конфігурації пакету

За допомогою клавіш зі стрілками вгору/вниз виберіть «Guess optimal character set» ("Вгадай оптимальний набір символів") (рис.1.32).

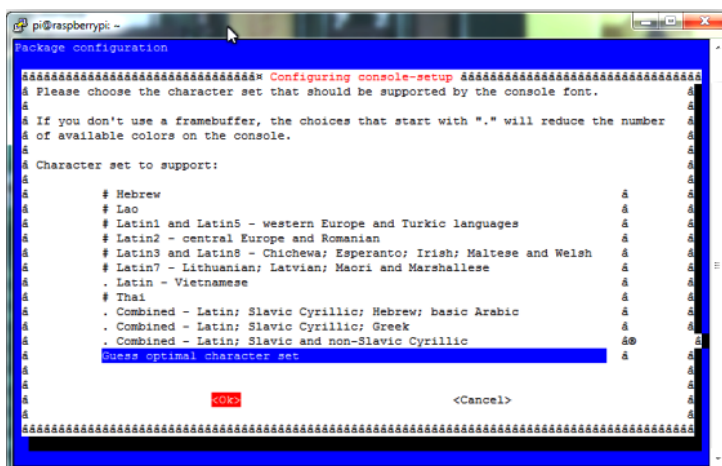


Рис.1.32 Вибір підтримки символів

Використовуючи клавішу зі стрілкою вправо виберіть "OK" і натисніть ENTER.

Використання стрілку вгору/вниз виберіть "Terminus" (рис.1.33).

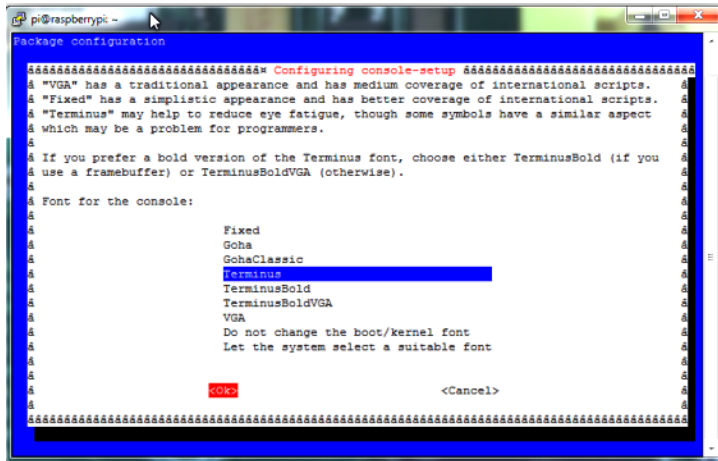


Рис.1.33 Вибір шрифту для консолі

Використовуючи клавішу зі стрілкою вправо виберіть "OK" і натисніть ENTER. За допомогою стрілок вгору/вниз виберіть "16×32" (рис.1.34).

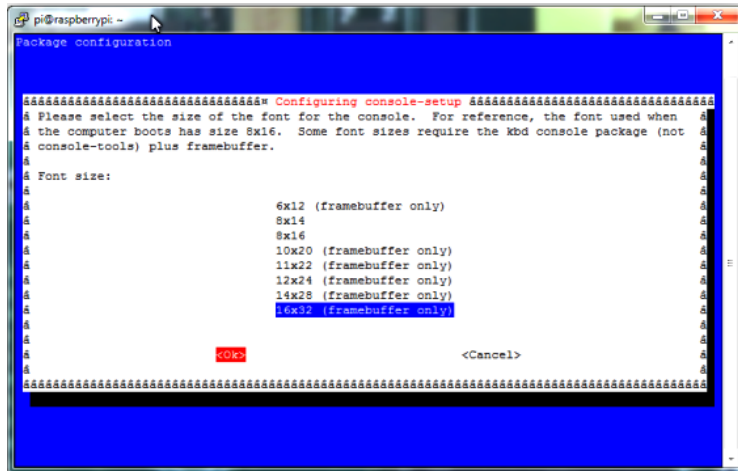


Рис.1.34 Вибір розміру шрифту

Використовуючи клавішу зі стрілкою вправо виберіть "OK" і натисніть ENTER. Після невеликої затримки ви повернетеся в командний рядок з новим розміром шрифту. 16×32 є в два рази більшим розміром шрифту, ніж за замовчуванням.

### Метод 2

Цей метод пропонує редагування системного файлу під назвою "console-setup". Вам просто потрібно завантажити його в свій текстовий редактор і налаштувати кілька рядків:

```
sudo nano /etc/default/console-setup
```

Ви побачите налаштування за замовчуванням (рис.1.35):

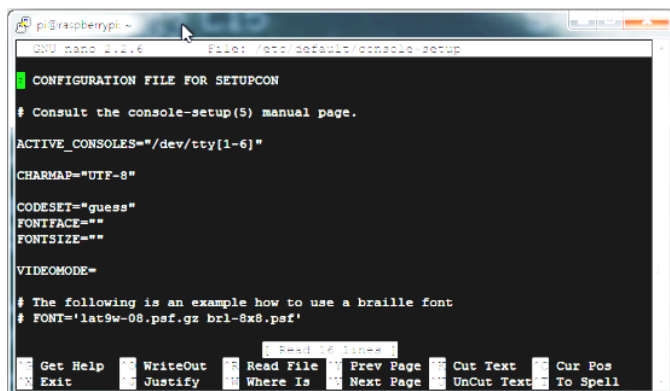


Рис.1.35 Налаштування за замовчуванням

Використовуючи клавіші зі стрілками, встановіть курсор і змініть рядки **FONTFACE** і **FONTSIZE** на:

```
FONTFACE="Terminus"  
FONTSIZE="16x32"
```

Файл повинен виглядати приблизно так (рис.1.36):

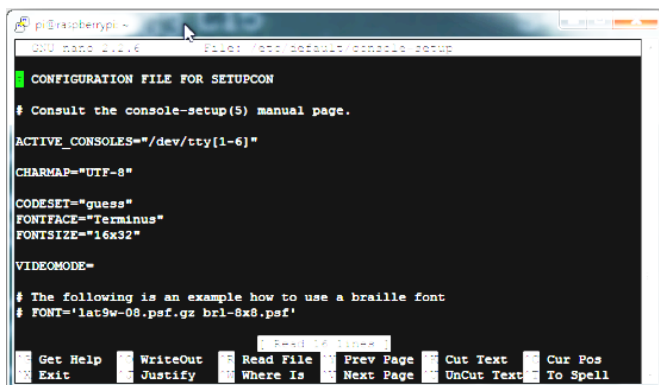


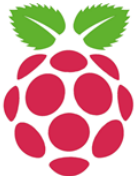
Рис.1.36 Відредагований файл налаштувань консолі

Натисніть CTRL+X, далі Y, а потім ENTER, щоб зберегти зміни.

Щоб застосувати нові налаштування для свого командного рядка без перезавантаження системи, можете виконати наступну команду для перезавантаження лише служби:

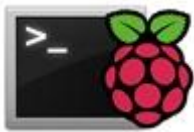
```
sudo /etc/init.d/console-setup restart
```

Який метод ви використаєте, залежить від вас. Автор вважає за краще використовувати метод 1, коли вперше змінюєте параметри за замовчуванням, а потім метод 2, якщо потрібно зробити швидку зміну пізніше.



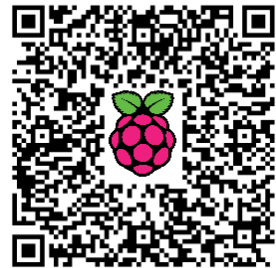
*Метод 1 насправді можна використати навіть тоді, коли шрифт поточного розміру не читається, бо, як можете бачити на скріншотах вище, більшість з наведеного можна зробити наздогадок!*

## Використання Advanced Packaging Tool – APT



Raspberry Pi є неймовірний маленький комп'ютер, який одноосібно відповідальний за впровадження нового покоління для програмування і експериментів, на відміну від навчання, як використовувати електронні таблиці і текстові процесори.

Для тих, хто новачок, в Raspberry Pi, є багато такого, що треба вводити з клавіатури і, отже, багато треба запам'ятати. В цьому розділі розглядаються корисні речі<sup>15</sup>, які потрібно знати в першу чергу.



### Repository

Програмне забезпечення на Linux в цілому, а також Raspberry Pi зокрема, підтримується через Repository. Сховища містять програмне забезпечення, яке складене та підтримується найчастіше добровольцями, на благо спільноти.

Люди, які підтримують програмне забезпечення в сховищах, забезпечують, щоб все також було під належним піклуванням. Розглянемо, як цими сховищами можна скористатися.

<sup>15</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1824-use-on-raspberry-pi-advanced-packaging-tool-apt>

### **Advanced Packaging Tool: APT**

У Linux для перегляду, встановлення, оновлення, модернізації і видалення програмного забезпечення ви повинні використовувати менеджер пакетів. Raspberry Pi використовує менеджер пакетів під назвою apt або Advanced Packaging Tool.

Advanced Packaging Tool - не одна програма: це безкоштовний інтерфейс користувача, який спрощує управління програмним забезпеченням для операційних систем Unix або Linux, включаючи Raspberry Pi. Функції Advanced Packaging Tool виконуються з командного рядка.

*Примітка:* Існують й інші менеджери пакетів для різних версій Linux. Наприклад, збірки Fedora для Linux використовують `yum` або `Yellowdog Update Manager`. Redhat використовує `rpm` або `Redhat Package Manager`.

### **Виконання підміни користувача**

Це, ймовірно, не краща ідея, щоб ваш Raspberry Pi або будь-який Linux-комп'ютер працювали за допомогою кореневих привілеїв, як щось само собою зрозуміле. Найбільш ймовірно, що їх у вас в будь-якому випадку немає, бо це не режим роботи за замовчуванням з Raspberry Pi.

Це означає, що виконання певних команд може бути неможливим, якщо у вас немає привілеїв безпеки `root`. Корисним інструментом, в цих умовах, є команда `sudo`, назва якої походить від скорочення `substitute user` (заміна користувача) і `do` (зробити).

Команда `sudo` дозволяє користувачам виконувати певні команди без необхідності входити як кореневий (`root`) користувач.

*Примітка:* Установка, оновлення або видалення програмного забезпечення вимагає використання привілеїв суперкористувача або використання команди `sudo`.

### **Як оновити базу даних репозитарія**

Корисно періодично оновлювати або повторно синхронізувати файли індексу пакетів з їх джерел. Джерела залежать від того, що було визначено в місці `/etc/apt/sources.list`. У Raspbian, джерелом за замовчуванням в цьому файлі є `deb http://mirrordirector.raspbian.org/raspbian/ wheezy main contrib non-free rpi`

Щоб оновити репозиторій програмного забезпечення на Raspberry Pi до останньої версії, введіть таку команду:

```
apt-get update
```

Якщо ви не використовуєте суперпривілеї, то потрібно ввести наступну команду (рис.1.37):

```
sudo apt-get update
```

```
pi@raspi-white ~$ sudo apt-get update
Get:1 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]
Get:2 http://mirror.raspbian.org wheezy Release.gpg [490 B]
Get:3 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:4 http://raspberrypi.collabora.com wheezy Release [5,000 B]
Get:5 http://mirror.raspbian.org wheezy Release [14.4 kB]
Get:6 http://archive.raspberrypi.org wheezy Release [7,227 B]
Hit http://repository.wolfraam.com stable Release.gpg
Hit http://repository.wolfraam.com stable Release
Hit http://raspberrypi.collabora.com wheezy/rpi armhf Packages
Get:7 http://mirror.raspbian.org wheezy/main armhf Packages [7,425 kB]
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
Get:8 http://archive.raspberrypi.org wheezy/main armhf Packages [16.9 kB]
Hit http://repository.wolfraam.com stable/non-free armhf Packages
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
17% [7 Packages 1,229 kB/7,425 kB 17%] [Waiting for headers]
```

Рис.1.37 Вікно оновлення з репозиторіїв

## Як шукати програмне забезпечення за назвою

Для того, щоб знайти певну частину програмного забезпечення, ви можете виконати пошук за назвою за допомогою наступної команди

```
apt-cache search [ключове слово для пошуку]
```

Практичний прикладом цього може бути пошук Google Chromium - браузера, який не є частиною основної установки Raspbian. Щоб виконати пошук Chromium введемо команду (рис.1.38):

```
apt-cache search chromium
```

*Примітка:* Зазвичай немає необхідності в привілеї root для виконання цієї команди.

```
pi@raspi-white ~$ apt-cache search chromium
chromium - Google's open source chromium web browser
chromium-browser - Chromium browser - transitional dummy package
chromium-browser-dbg - chromium-browser debug symbols transitional dummy package
chromium-browser-inspector - page inspector for the chromium-browser - transitional dummy package
chromium-browser-l10n - chromium-browser language packages - transitional dummy package
chromium-bsu - fast paced, arcade-style, scrolling space shooter
chromium-bsu-data - data pack for the Chromium 8.S.U. game
chromium-dbg - Debugging symbols for the chromium web browser
chromium-inspector - page inspector for the chromium browser
chromium-l10n - chromium-browser language packages
gecko-mediaplayer - Multimedia plug-in for Gecko browsers
libv8-3.8.9.20 - v8 JavaScript engine - runtime library
libv8-dbg - v8 JavaScript engine - debugging symbols
libv8-dev - v8 JavaScript engine - development files
pi@raspi-white ~$
```

Рис.1.38 Пошук програмного забезпечення

### Як встановити програмний пакет

Як і для пошуку, використовують назву пакета для установки необхідного програмного забезпечення. Потрібна тільки назва частини пакету, а не повна назва файлу.

При необхідності, apt автоматично отримає і установить пакети, від яких залежить вказаний пакет. Це дозволяє уникнути збоїв установки, які є результатом відсутності залежних пакетів.

Для установки програмного забезпечення, скористайтеся командою:

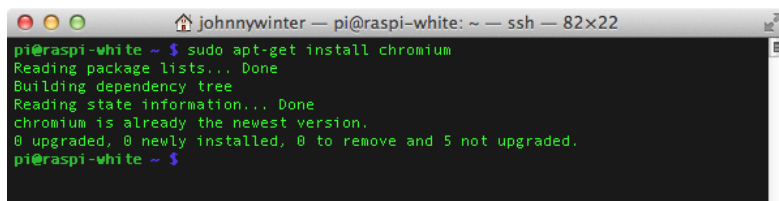
```
apt-get install [назва програмного забезпечення]
```

При відсутності суперповноваження, потрібно ввести команду:

```
sudo apt-get install [назва програмного забезпечення]
```

Практичним прикладом цього може бути (рис.1.39):

```
sudo apt-get install chromium
```



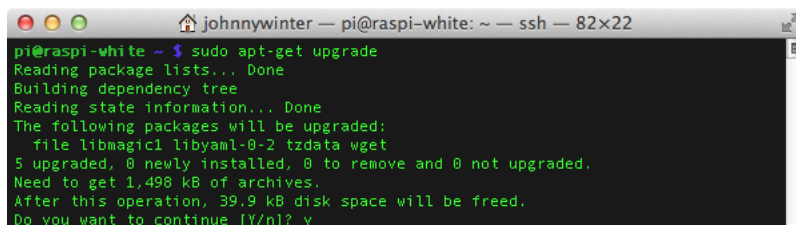
```
johnnywinter — pi@raspi-white: ~ — ssh — 82x22
pi@raspi-white ~ $ sudo apt-get install chromium
Reading package lists... Done
Building dependency tree
Reading state information... Done
chromium is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
pi@raspi-white ~ $
```

Рис.1.39 Встановлення програмного забезпечення

*Примітка:* У наведеному прикладі пакет вже встановлений

### Модернізація всіх раніше встановлених пакетів

Модернізація (**upgrade**) відрізняється від оновлення (**update**).



```
johnnywinter — pi@raspi-white: ~ — ssh — 82x22
pi@raspi-white ~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  file libmagic1 libyaml-0-2 tzdata wget
5 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 1.498 kB of archives.
After this operation, 39.9 kB disk space will be freed.
Do you want to continue [Y/n]? y
```

Рис.1.40 Модернізація встановлених пакетів

Модернізація встановлює новітні версії всіх пакетів, які маєте в даний час на Raspberry Pi. Модернізуються тільки встановлені пакети.

```
apt-get upgrade
```

Якщо не використовуєте суперповноваження, то потрібно ввести попередню команду з `sudo` (рис.1.40):

### Як видалити програмний пакет

Час від часу ви можете змінити робочі дані і видалити всі пакети, які більше не потрібні. Це збереже на вашому Pi «м'ясо» і гарантує, що ви не заповните до кінця SD-карту пакетами або програмним забезпеченням, які не потрібні.

Для видалення пакета, введіть наступне в командному рядку:

```
apt-get remove [назва програмного забезпечення]
```

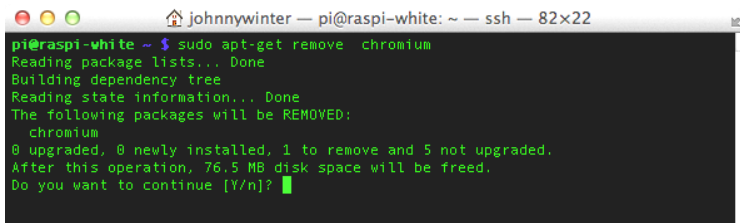


Рис.1.41 Видалення пакета

Якщо не використовуєте суперповноваження, то потрібно ввести команду (рис.1.41):

```
sudo apt-get remove [назва програмного забезпечення]
```

Наприклад: `sudo apt-get remove chromium`

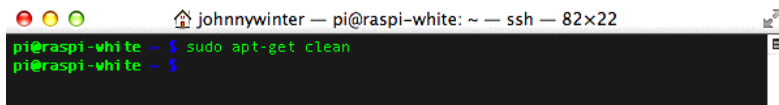
### Як очистити кеш apt-get

Команда `apt-get clean` очищає локальне сховище отриманих файлів пакетів. Це корисний інструмент для звільнення дискового простору:

```
apt-get clean
```

Якщо не використовуєте суперповноваження, то потрібно ввести команду (рис.1.42):

```
sudo apt-get clean
```



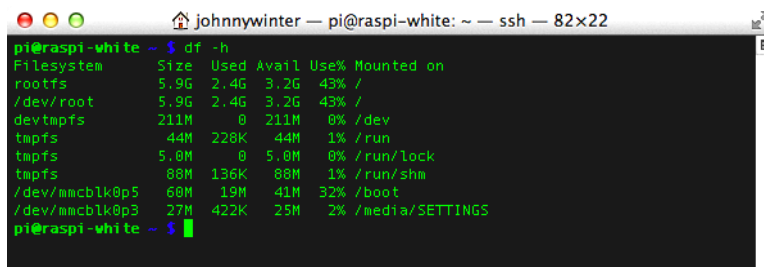
```
johnnywinter — pi@raspi-white: ~ — ssh — 82x22
pi@raspi-white ~ $ sudo apt-get clean
pi@raspi-white ~ $
```

Рис.1.42 Очищення локального сховища

### Як визначити вільний дисковий простір

Часто необхідно вміти визначити, скільки простору використовується і, якщо використали `apt-get clean`, дізнатися, скільки місця звільнилося.

Щоб отримати уявлення про простір до і після очищення, скористайтеся командою `df`, яка повідомляє про обсяг дискового простору, використаного файловими системами (рис.1.43): `df -h`



```
johnnywinter — pi@raspi-white: ~ — ssh — 82x22
pi@raspi-white ~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          5.9G  2.4G  3.2G  43% /
/dev/root       5.9G  2.4G  3.2G  43% /
devtmpfs       211M    0  211M   0% /dev
tmpfs           44M   228K   44M   1% /run
tmpfs           5.0M    0   5.0M   0% /run/lock
tmpfs           68M   136K   68M   1% /run/shm
/dev/mmcblk0p5  60M    19M   41M  32% /boot
/dev/mmcblk0p3  27M   422K   25M   2% /media/SETTINGS
pi@raspi-white ~ $
```

Рис.1.43 Визначення вільного дискового простору

Параметр `-h` виводить результати в зручному для користувача вигляді.

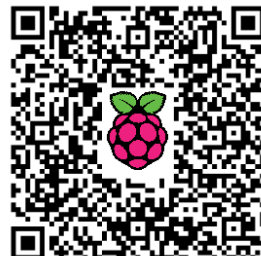
### Як встановити додатки на Raspberry Pi



Розглянемо, як встановлювати додатки з доступних репозиторіїв<sup>16</sup>.

#### Крок 1. Використання АРТ

Метод, який автор хотів би рекомендувати для встановлення додатків – це використання інструменту під назвою Synaptic, що являє собою графічний інтерфейс, який дозволяє виконувати пошук додатків за іменем або за типом.



<sup>16</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1768-how-to-install-applications-on-raspberry-pi>

Повсякденному користувачу Raspbian буде зручніше працювати з Synaptic GUI.

На жаль, Synaptic не встановлюється за умовчанням в Raspbian і тому, якщо хочете використовувати цей додаток надалі, потрібно його встановити. Для встановлення додатка скористаємося раніше розглянутим `apt`.

Відкрийте вікно терміналу, натиснувши на значок LXTerminal на робочому столі та введіть (рис.1.44):

```
sudo nano /etc/apt/sources.list
```

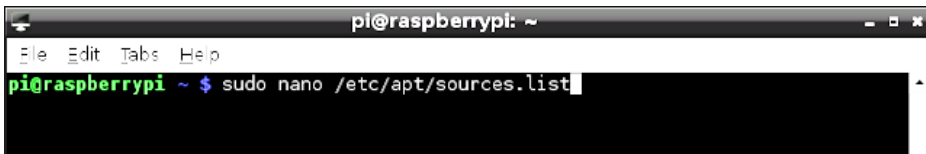


Рис.1.44 Запуск редактора nano для перегляду списку репозиторіїв

і натисніть клавішу введення.

`Sources.list` містить список репозиторіїв (рис.1.45), які будуть використовуватися менеджером `apt` для створення бази даних додатків, що можуть бути завантажені і встановлені на Raspberry Pi.

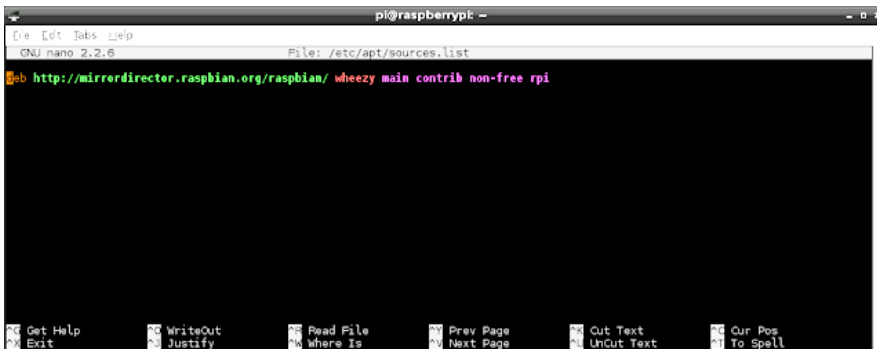


Рис.1.45. Список репозиторіїв

Кожен рядок містить різне розташування сховища. Тому, переглядаючи значення за замовчуванням в `sources.list`, ви побачите, що окремий рядок виглядає наступним чином:

```
deb http://mirrordirector.raspbian.org/raspbian wheezy main
contrib non-free rpi
```

Що ж все це означає?

Так, "deb" - це тип сховища, який в даному випадку означає, що воно є сховищем Debian. Інші типи включають RPM і Repomd, але для Raspberry Pi з Raspbian нас цікавлять лише пакети debian.

Наступним важливим моментом є розташування (URL) сховища.

Після цього вказаний дистрибутив, яким в нашому випадку є Wheezy.

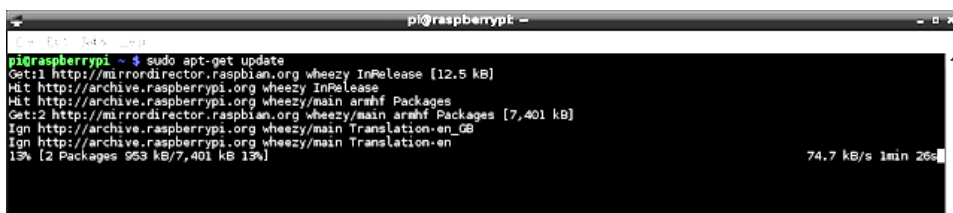
Решта з цих пунктів - це категорії, на яких базуються додатки.

Вам не потрібно нічого робити з цим файлом. Це просто трохи інформації, яка дозволяє дізнатися, звідки **apt** дістає дані. Натисніть CTRL і X для виходу.

Таким чином, реальною метою цього підрозділу є дізнатися, як завантажити пакети з допомогою **apt**, щоб можна було скачати Synaptic.

Перше, що потрібно зробити, це переконатися, що бази даних **apt** в актуальному стані. Це можна зробити, набравши у вікні терміналу (рис.1.46):

```
sudo apt-get update
```

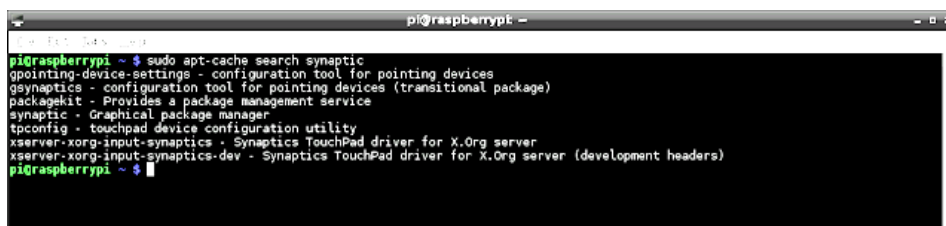


```
pi@raspberrypi ~ $ sudo apt-get update
Get:1 http://mirrordirector.raspbian.org wheezy InRelease [12.5 kB]
Hit http://archive.raspberrypi.org wheezy InRelease
Get:2 http://mirrordirector.raspbian.org wheezy/main armhf Packages [7,401 kB]
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
13% [2 Packages 963 kB/7,401 kB 13%] 74.7 kB/s 1min 26s
```

Рис.1.46 Оновлення баз даних АРТ

Для пошуку з **apt** Synaptic ви повинні ввести (рис.1.47):

```
sudo apt-cache search synaptic
```



```
pi@raspberrypi ~ $ sudo apt-cache search synaptic
gpointing-device-settings - configuration tool for pointing devices
gsynaptics - configuration tool for pointing devices (transitional package)
packagekit - Provides a package management service
synaptic - Graphical package manager
tpconfig - touchpad device configuration utility
xserver-xorg-input-synaptics - Synaptics TouchPad driver for X.Org server
xserver-xorg-input-synaptics-dev - Synaptics TouchPad driver for X.Org server (development headers)
pi@raspberrypi ~ $
```

Рис.1.47 Пошук для synaptic

Повертається список пакетів з Synaptic в імені або в опису.

Для встановлення Synaptic все, що потрібно зробити, це ввести:

```
sudo apt-get install synaptic
```

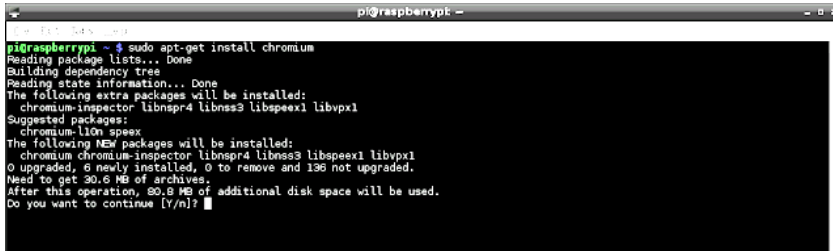


Рис.1.48 Встановлення пакету

На зображенні (рис.1.48) можете бачити, що автор вирішив встановити веб-браузер Chromium замість Synaptic, бо його він вже встановив раніше.

Коли використовуєте команду `apt-get install`, то отримаєте повідомлення про те, які саме пакети будуть завантажуватися і встановлюватися і наскільки вони великі.

Якщо бажаєте встановити, натисніть `Y`, щоб продовжити встановлення.

Текст буде прокручуватися на екрані вгору, показуючи, що відбувається. В основному, він включає в себе кілька завантажених пакетів, які потім встановлюються. Коли процес буде завершений, введіть `exit`, щоб закрити вікно терміналу.

У меню в розділі System (Система) повинна бути опція для Synaptic.

## Крок 2. Використання Synaptic

Запустіть Synaptic, вибравши його в меню.

Вас привітає вікно входу в систему (рис.1.49). Вам треба ввести пароль, налаштований для користувача `pi`, і натиснути OK.

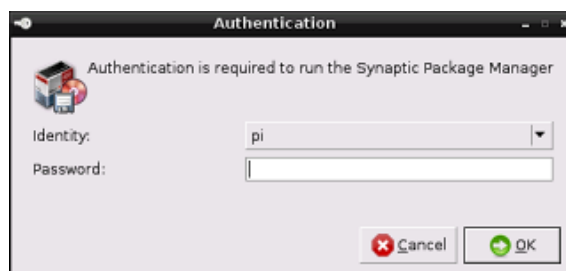


Рис.1.49 Авторизація в Synaptic

Synaptic завантажиться у фоновому режимі, але буде відображатися повідомлення (рис.1.50), щоб розказувати про цілі Synaptic і керувати пакетами в цілому.

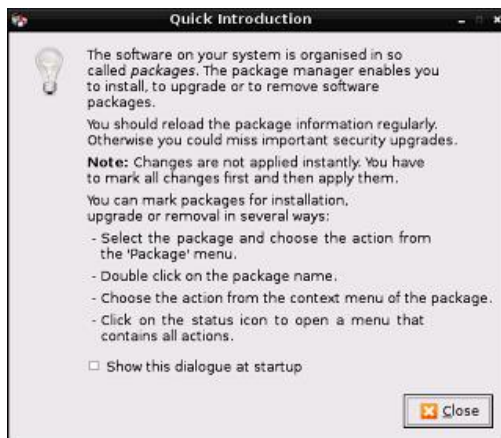


Рис.1.50 Інформаційне повідомлення Synaptic

Зверніть увагу, що є прапорець «Show this dialogue at start» («Показувати цей діалог при запуску»). Якщо не хочете, щоб це повідомлення було щоразу при запуску Synaptic, зніміть прапорець, а потім натисніть кнопку закриття.

Synaptic забезпечує набагато зручніший візуальний спосіб перегляду пакетів, які доступні в сховищах Raspbian.

В лівій боковій панелі знаходиться список категорій, а праворуч - пакети, які доступні в обраній категорії (рис.1.51).

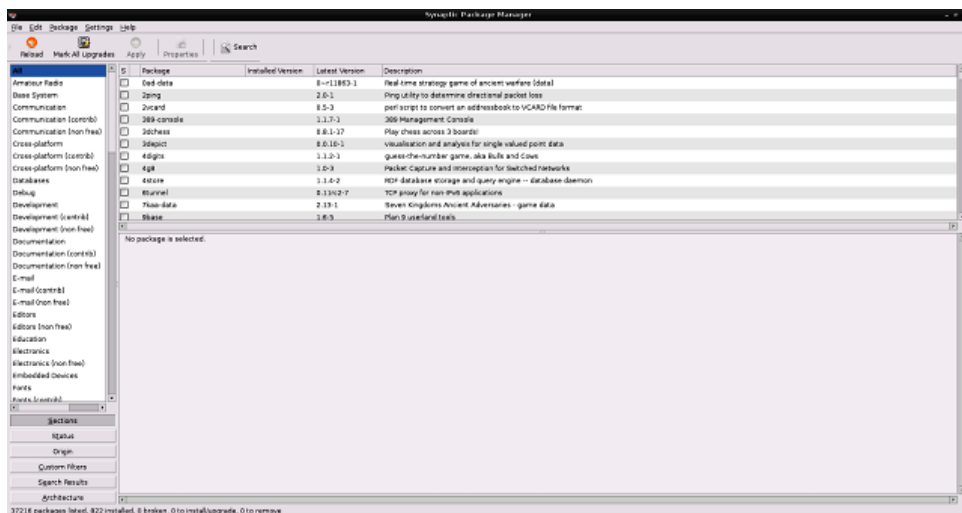


Рис.1.51 Списки категорій і пакетів

Якщо хочете знайти конкретну програму за іменем або за описом - натисніть кнопку пошуку.



З'явиться вікно, показуючи, які додатки будуть встановлені і всі необхідні залежності, які також повинні бути встановлені, а також додатки, обрані вами.

Для продовження встановлення натисніть кнопку Mark (Позначити) (рис.1.54):

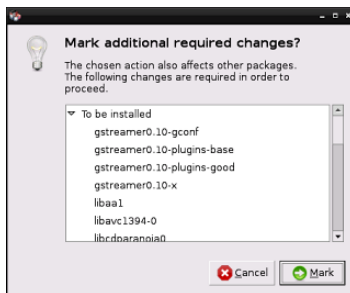


Рис.1.54 Вікно позначення додатків для встановлення

В цій точці ви маєте останній шанс скасувати встановлення. З'явиться вікно (рис.1.55), показуючи, наскільки велике завантаження і список пакетів, які будуть встановлені. Якщо бажаєте продовжити, натисніть кнопку Apply (Застосувати).

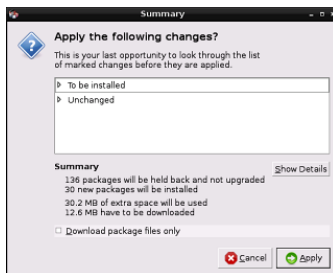


Рис.1.55 Список пакетів для встановлення

З'явиться вікно, показуючи хід завантаження і скільки часу, очікується, це займе (рис.1.56).

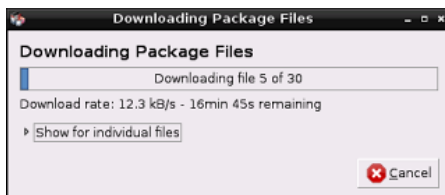


Рис.1.56 Вікно ходу завантаження

Коли завантаження завершаться, з'явиться інше вікно, показуючи хід фактичного встановлення (рис.1.57).

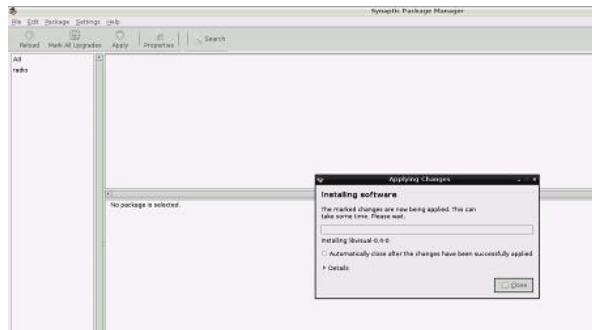


Рис.1.57 Хід фактичного встановлення

Нарешті, як тільки все це завершиться, обрані вами програми будуть встановлені.

## Магазин PI

Якщо ж хочете спробувати деякі «доморощені» додатки, то можете перевірити магазин Raspberry Pi<sup>17</sup> (рис.1.58), ярлик якого є на робочому столі.

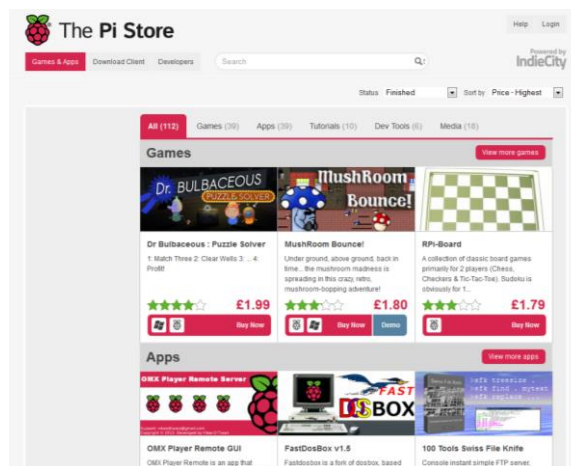


Рис.1.58 Магазин Raspberry Pi

## Висновки

Для досвідчених користувачів Linux завантаження програм - це друга натура, а для тих, що зайшов в Linux вперше, і особливо для користувачів, чиїм першим досвідом Linux є Raspberry Pi, не так очевидно, як встановити нові програми.

<sup>17</sup> <http://store.raspberrypi.com/projects>

### Як автозавантажувати додатки в робочому столі LXDE

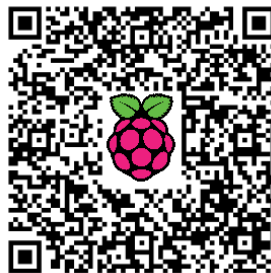


Якщо використовуєте операційну систему Raspbian на Raspberry Pi то знаєте, що при введенні `startx` запускається графічний інтерфейс користувача LXDE<sup>18</sup>. Для цього режиму є багато програм і утиліт.

У проектах можете автозавантажувати один або декілька з необхідних додатків при запуску `startx`, зберігаючи час при їх запуску вручну.

Популярні вибори можуть включати LXTerminal, Scratch, Мідорі і Leafpad.

Є два методи, але якщо є сумніви, то використовуйте метод 1. У наведених нижче прикладах автор додав два додатки (LXTerminal і Leafpad), але можете додати багато з того, що потрібно. Почніть з завантаження свого Pi в режим командного рядка.



#### Метод 1

Цей метод використовує глобальний список автозапуску додатків, який застосовується до всіх облікових записів користувачів. Вони будуть завантажуватися щоразу, коли завантажується LXDE, незалежно від того, який користувач увійшов в систему в цей час. Зазвичай, це користувач облікового запису за замовчуванням `pi`.

Використайте наступну команду, щоб запустити текстовий редактор `nano` і відредагувати файл автозапуску:

```
sudo nano /etc/xdg/lxsession/LXDE/autostart
```

Після останнього рядка додайте рядок для кожної програми, яку хочете автозапускати. Наприклад, якщо додати рядки для LXTerminal і Leafpad, то файл буде виглядати наступним чином:

```
@lxpanel --profile LXDE
@pcmanfm --desktop --profile LXDE
@xscreensaver -no-splash@
@lxterminal
@leafpad
```

Для збереження і виходу з редактора `nano` натисніть CTRL+X, Y, а потім ENTER. Ви можете використовувати свій улюблений текстовий редактор, якщо ваш вибір не `nano`.

<sup>18</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1832-how-to-autostart-apps-in-raspbian-lxde-desktop>

## Метод 2

Альтернативний метод полягає у створенні файлу конфігурації, який є унікальним для поточного користувача. Насамперед, необхідно змінити цей текстовий файл:

```
sudo nano ~/.config/lxsession/LXDE/autostart
```

Як і в методі 1 цей файл є списком команд для виконання при завантаженні GUI. Цей файл, як правило порожній, коли вперше редагуєте його, тому просто додайте потрібні програми для автоматичного завантаження:

```
@lxterminal
@leafpad
```

Для збереження і виходу з редактора nano натисніть послідовно CTRL+X, Y, а потім ENTER.

Як тільки ваш файл конфігурації оновиться, будете готові перевірити:

```
startx
```

Робочий стіл LXDE повинен завантажитися та запуститися вибрані вами програми.

## Обидва методи одночасно?

Ви можете використовувати обидва методи. Додатки з кожного файлу конфігурації запустяться, як і очікувалося. Це дозволить запускати спеціальні програми для всіх, дозволяючи окремим користувачам мати автозапуск своїх власних програм. Наскільки це корисно, залежатиме від того, які облікові записи інших користувачів ви додали при встановленні.

## Автозапуск Minecraft

Припустимо, що ви вже встановили Minecraft<sup>19</sup> на Pi, то тепер також можете автоматично його запускати. Коли автор перебуває в режимі Python-Minecraft, то запускає Minecraft і вікно терміналу готове для виконання сценаріїв за допомогою наступних рядків:

```
@~/mcpi/minecraft-pi
@lxterminal
```

<sup>19</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1779--platform-for-minecraft-on-raspberry-pi>

Це припускає, що виконуваний Minecraft знаходиться в `~/mcpi/` (те ж саме, що `/home/pi/mcpi/`).

### Автозапуск сценаріїв Python

Ви можете автоматично запустити свої власні сценарії Python, додавши рядок:

```
@/usr/bin/python /home/pi/example.py
```

Це працює краще за все з методом 2, бо в цьому прикладі сценарій Python зберігається в домашньому каталозі користувача Pi за замовчуванням. Якщо інший користувач увійшов в систему, то він не буде мати доступ до цього каталогу, тому LXDE не зможе автоматично завантажити його.

Який би метод ви не використовували, тільки введіть команду **startx** в командному рядку і вибрані вами програми повинні завантажитися автоматично (рис.1.59).

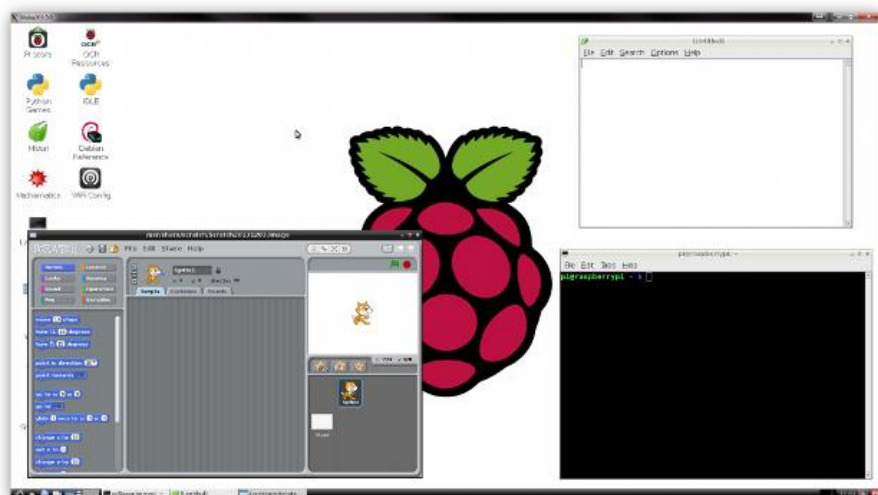
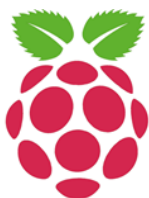


Рис.1.59 Автоматичне завантаження програм

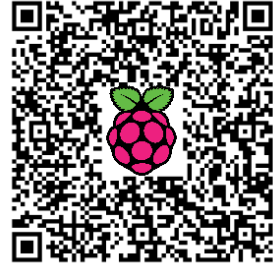


Порядок, в якому розміщені команди для додатків, не завжди означає, що вони будуть завантажені в тому ж порядку, бо різні додатки вимагають різний час для повного завантаження. Рекомендуємо ставити великі програми першими, тому що вони вимагають більше часу, щоб запуститися, тобто *Мідорі* перед *LXTerminal*.

## Як монтувати/автомонтувати жорсткий диск USB



Розглянемо детальніше, як зробити, щоб ваш диск USB, встановлений на Raspberry Pi, монтувався щоразу при завантаженні<sup>20</sup>.



Ці кроки особливо необхідні, якщо ви налаштовуєте Samba, або торрент-завантажувач 24x7, або щось подібне, коли ваш Raspberry Pi повинен мати зовнішній накопичувач вже змонтованим і готовим для доступу до служб/демонів.

### Крок 0

Підключіть свій USB HDD до Raspberry Pi.

Якщо використовуєте відформатований диск з NTFS, то встановіть наступне:

```
sudo apt-get install ntfs-3g
```

### Крок 1

Увійдіть на Pi і виконайте:

```
ls -l /dev/disk/by-uuid/
```

Ви побачите щось на зразок такого:

```
lrwxrwxrwx 1 root root 10 Jan 1 1970 0AC4D607C4D5F543 ->
../../sda1
```

Запишіть значення UUID -> **0AC4D607C4D5F543**

### Крок 2

Створіть місце для точки монтування:

```
sudo mkdir /media/NASDRIVE
```

Надайте відповідні дозволи:

```
sudo chmod 775 /media/NASDRIVE
```

Права доступу записуються в `chmod` одним рядком одночасно для трьох типів користувачів:

- власника файла;

<sup>20</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1791-how-to-setup-mount-auto-mount-usb-hard-drive-on-raspberry-pi>

- інших користувачів, які входять в групу власника;
- всі решта користувачів.

Аргумент команди `chmod`, який задає дозвіл, може бути записаний в двох форматах: в числовому або в символному.

### **Приклад - значення права «755»**

|                             | <b>Власник</b> | <b>Група</b>        | <b>Решта</b>        |
|-----------------------------|----------------|---------------------|---------------------|
| Восьмеричне значення        | 7              | 5                   | 5                   |
| Символьний запис            | <b>rw</b> x    | <b>r</b> - <b>x</b> | <b>r</b> - <b>x</b> |
| Позначення типу користувача | <b>u</b>       | <b>g</b>            | <b>o</b>            |

Для директорій зі всього списку зазвичай застосовують лише 0, 5 і 7 - повна заборона, читання і повний доступ.

В нашому випадку 755 (**-rwx-----**) означає, що власник може читати, записувати і запускати для виконання, а інші мають право лише читати.

### **Крок 3**

Встановіть диск з інтерфейсом USB, а потім перевірте, чи доступний він в `/media/NASDRIVE`:

```
sudo mount -t ntfs-3g -o uid=pi,gid=pi /dev/sda1 /media/NASDRIVE
```

*Примітка:*

- **ntfs-3g** для NTFS-дисків
- **vfat** для FAT32-дисків
- **ext4** для ext4 -дисків

### **Крок 4**

Тепер налаштуємо RPi робити це після кожного перезавантаження. Зробіть резервну копію поточної `fstab`:

```
sudo cp /etc/fstab /etc/fstab.backup
```

Додайте інформацію монтування в файл `fstab`:

```
UUID=0AC4D607C4D5F543 /media/NASDRIVE ntfs-3g rw,defaults 0 0
```

### **Крок 5**

Перезавантажтесь: `sudo reboot`

Увімкніть графічний режим через `sudo raspi-config`, тільки якщо це необхідно. Ви повинні побачити, що ваш привід USB автоматично монтується на `/media/NASDRIVE`.

## 2. НАЛАШТУВАННЯ ЛОКАЛЬНОЇ МЕРЕЖІ ТА ДОСТУПУ В ІНТЕРНЕТ

### Як надати Raspberry Pi статичну IP-адресу

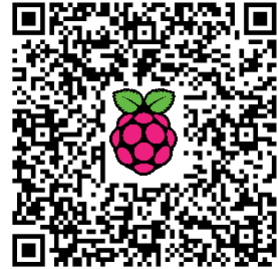


Щоб увійти у свій Raspberry Pi віддалено, вам знадобиться IP-адреса для Raspberry Pi<sup>21</sup> - це як адреса вашого будинку, щоб повідомити адресу головному комп'ютеру, де шукати його в мережі.

Типово, IP надається Raspberry Pi маршрутизатором при підключенні до мережі автоматично (так звана динамічна IP-адреса і надається через DHCP). Тим не менш, ця адреса може змінитися, коли видалите Pi з мережі, наприклад, вимкнете його.

Мати статичну IP-адресу не є суттєвим, однак вона дозволить робити повторні звернення до Raspberry Pi через SSH набагато простіше, так як ви завжди будете знати, яку адресу має Raspberry Pi.

Це завдання передбачає, що ви встановили офіційний реліз Raspbian OS. Дана функція також доступна в дистрибутиві NOOBS. Більш детально про цей реліз ми розповідали вище. Також припускаємо, що ви під'єднали свій Pi до мережі через Ethernet. Якщо збираєтеся входити у Pi віддалено для виконання більшості завдань, то автор рекомендує саме просте і швидке рішення: розмістити його поруч з маршрутизатором і завжди використовувати Ethernet для доступу в Інтернет!



### Перевірка налаштування

Завантажте Raspbian і увійдіть (логін **pi**, пароль **raspberrypi**) - все це робиться з командного рядка, тому немає необхідності входити в GUI.

По-перше, ми повинні отримати список мережевих інтерфейсів, які зараз є у нас в наявності, через команду **cat** (рис.2.1):

```
pi@raspberrypi ~ $ cat /etc/network/interfaces
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
pi@raspberrypi ~ $
```

Рис.2.1 Список наявних мережевих інтерфейсів

<sup>21</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1759-how-to-give-your-raspberry-pi-static-ip-address>

```
cat /etc/network/interfaces
```

Рядок:

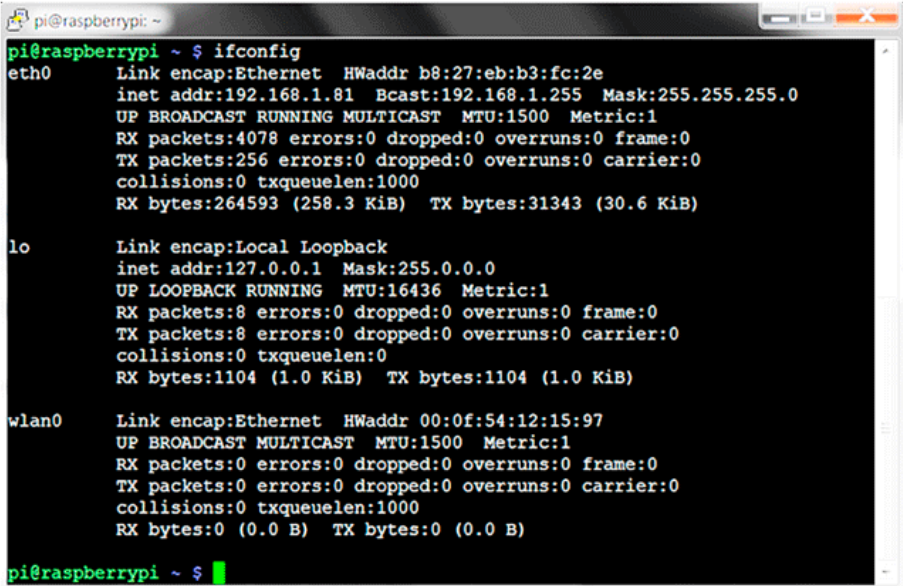
```
iface eth0 inet dhcp
```

означає, що ми в даний час отримуємо IP-адреси через DHCP, тобто вони динамічно реєструються за допомогою маршрутизатора. Це те, що ми хочемо змінити!

### Збір інформації

Перш за все, потрібно захопити деяку інформацію від нашого маршрутизатора і Pi. Тому ми повинні виконати пару команд, щоб отримати цю інформацію. Приготуйте ручку і папір!

ifconfig



```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:b3:fc:2e
          inet addr:192.168.1.81 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4078 errors:0 dropped:0 overruns:0 frame:0
          TX packets:256 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:264593 (258.3 KiB)  TX bytes:31343 (30.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1104 (1.0 KiB)  TX bytes:1104 (1.0 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:0f:54:12:15:97
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi ~ $
```

Рис.2.2 Інформація від маршрутизатора

Цю інформацію показує ваш маршрутизатор і частина, яку хочете отримати, після eth0 (з'єднання Ethernet) (рис.2.2):

```
eth0      Link encap:Ethernet  HWaddr b8:27:eb:b3:fc:2e
          inet addr:192.168.1.81 Bcast:192.168.1.255
          Mask:255.255.255.0
```

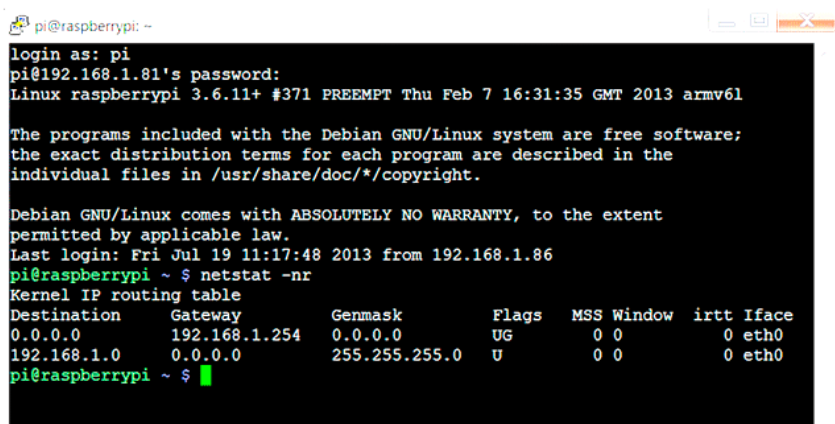
Запишіть наступну інформацію:

**inet addr** – 192.168.1.81 (поточна IP-адреса Pi)  
**Bcast** – 192.168.1.255 (діапазон IP-трансляції)  
**Mask** – 255.255.255.0 (маска підмережі)

Нам потрібно трохи більше інформації, перш ніж ми продовжимо. Використайте команду (рис.2.3):

```
netstat -nr
```

(команда **route -n** надасть вам ту ж інформацію)



```
pi@raspberrypi: ~  
login as: pi  
pi@192.168.1.81's password:  
Linux raspberrypi 3.6.11+ #371 PREEMPT Thu Feb 7 16:31:35 GMT 2013 armv6l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Jul 19 11:17:48 2013 from 192.168.1.86  
pi@raspberrypi ~ $ netstat -nr  
Kernel IP routing table  
Destination      Gateway         Genmask         Flags   MSS Window  irtt Iface  
0.0.0.0          192.168.1.254  0.0.0.0         UG        0 0          0 eth0  
192.168.1.0      0.0.0.0        255.255.255.0   U          0 0          0 eth0  
pi@raspberrypi ~ $
```

Рис.2.3 Додаткова інформація про мережу

Нам знадобляться:

- **Gateway** (адреса шлюзу) - 192.168.1.254
- **Destination** (адреса призначення) - 192.168.1.0

### Редагування конфігурації мережі

Тепер нам потрібно ввести цю інформацію в конфігураційний файл мережі Pi за допомогою текстового редактора. Автор зазвичай використовує текстовий редактор nano:

```
sudo nano /etc/network/interfaces
```

Просто змініть рядок (рис.2.4), в якому написано:

```
iface eth0 inet dhcp
```

на

```
iface eth0 inet static
```

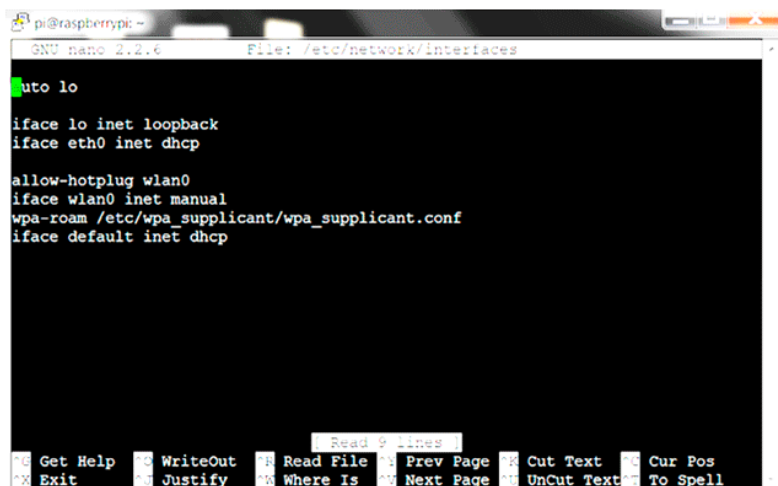


Рис.2.4 Конфігураційний файл

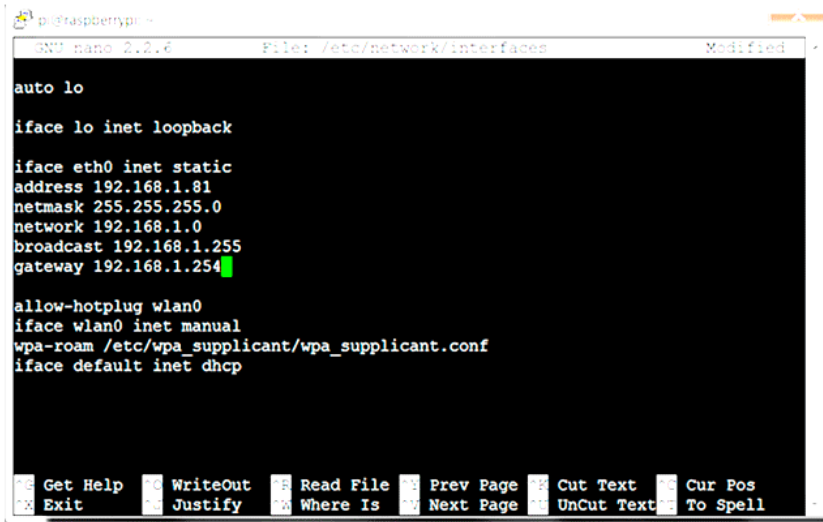
Далі, безпосередньо нижче цього рядка, введіть наступне (зверніть увагу, що вам знадобляться свої власні адреси, які зібрані в попередній частині, - більш докладно нижче):

```
address 192.168.1.81
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.254
```

Пояснимо, що означає кожна частина:

- **address** - адреса, яку хочете дати своєму Pi - це може бути будь-яка IP-адреса в діапазоні мережі, але її, як правило, бажано взяти вище, а не нижче, щоб ви могли закінчити реєстрацію різних пристроїв з тим же IP! Автор вибрав 192.168.1.81, бо ми вже зареєстровані за цією адресою (позначається 'inet addr'), але це може бути будь-яка IP-адреса в діапазоні від 192.168.1.1 до 192.168.1.255.
- **netmask** - адреса 'Mask', яку ми записали раніше.
- **network** - IP-адреса маршрутизатора - це адреса призначення, яка була знайдена раніше. Ви можете також захопити її від свого маршрутизатора.
- **broadcast** - адресу 'Bcast' ми записали раніше.
- **gateway** - адреса шлюзу ('Gateway'), яку також знайшли раніше.

Таким чином, це має виглядати приблизно так як нижче (рис.2.5), але з вашими значеннями. Не забудьте перед виходом натиснути CTRL+X (exit), щоб збереглися зроблені зміни.

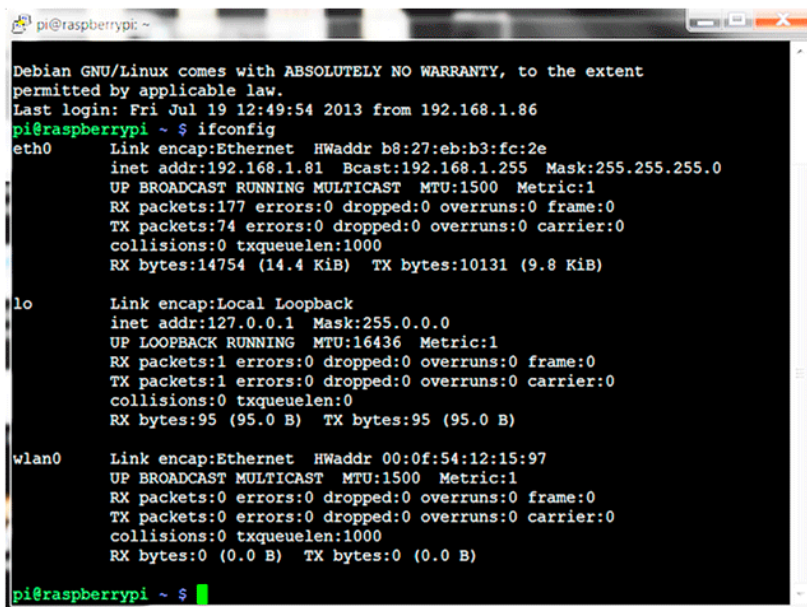


```
pi@raspberrypi ~  
GNU nano 2.2.6 File: /etc/network/interfaces Modified  
  
auto lo  
  
iface lo inet loopback  
  
iface eth0 inet static  
address 192.168.1.81  
netmask 255.255.255.0  
network 192.168.1.0  
broadcast 192.168.1.255  
gateway 192.168.1.254  
  
allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf  
iface default inet dhcp  
  
Get Help WriteOut Read File Prev Page Cut Text Cur Pos  
Exit Justify Where Is Next Page UnCut Text To Spell
```

Рис.2.5 Редагування конфігураційного файлу

### Ще раз перевірте статичну IP-конфігурацію

Для цього нам потрібно перезавантажитися з `sudo reboot` і перевірити зміни через `ifconfig` (рис.2.6):



```
pi@raspberrypi ~  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Jul 19 12:49:54 2013 from 192.168.1.86  
pi@raspberrypi ~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:b3:fc:2e  
          inet addr:192.168.1.81  Bcast:192.168.1.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:177 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:14754 (14.4 KiB)  TX bytes:10131 (9.8 KiB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:95 (95.0 B)  TX bytes:95 (95.0 B)  
  
wlan0     Link encap:Ethernet  HWaddr 00:0f:54:12:15:97  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
pi@raspberrypi ~$
```

Рис.2.6 Перевірка статичної IP-конфігурації

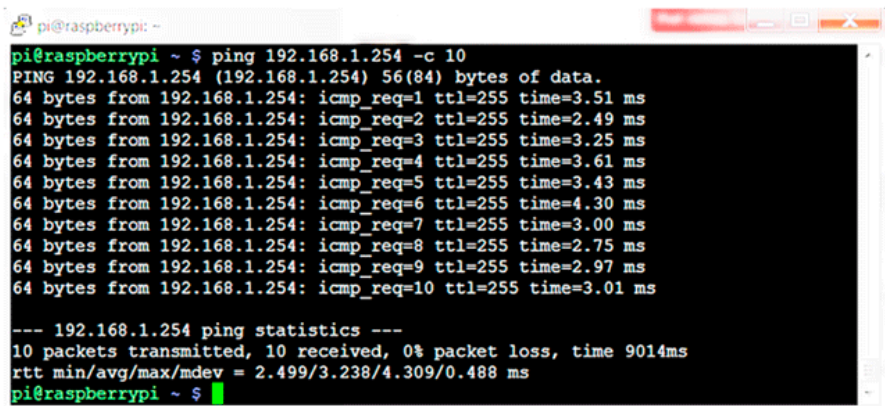
## 2. НАЛАШТУВАННЯ ЛОКАЛЬНОЇ МЕРЕЖІ ТА ДОСТУПУ В ІНТЕРНЕТ

Щоб додатково перевірити чи все працює як треба, пінгуємо адресу шлюзу (рис.2.7):

```
ping 192.168.1.254 -c 10
```

(-c 10 в команді просто означає, що ви хочете пінгувати його 10 разів, якщо забудете це додати, то адреса буде пінгуватися безперервно. Зупинити пінгування можна натисканням Ctrl+C).

Пінгування повинно пройти успішно і будуть отримані всі пакети.



```
pi@raspberrypi ~ $ ping 192.168.1.254 -c 10
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_req=1 ttl=255 time=3.51 ms
64 bytes from 192.168.1.254: icmp_req=2 ttl=255 time=2.49 ms
64 bytes from 192.168.1.254: icmp_req=3 ttl=255 time=3.25 ms
64 bytes from 192.168.1.254: icmp_req=4 ttl=255 time=3.61 ms
64 bytes from 192.168.1.254: icmp_req=5 ttl=255 time=3.43 ms
64 bytes from 192.168.1.254: icmp_req=6 ttl=255 time=4.30 ms
64 bytes from 192.168.1.254: icmp_req=7 ttl=255 time=3.00 ms
64 bytes from 192.168.1.254: icmp_req=8 ttl=255 time=2.75 ms
64 bytes from 192.168.1.254: icmp_req=9 ttl=255 time=2.97 ms
64 bytes from 192.168.1.254: icmp_req=10 ttl=255 time=3.01 ms

--- 192.168.1.254 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 2.499/3.238/4.309/0.488 ms
pi@raspberrypi ~ $
```

Рис.2.7 Результати пінгування

Якщо щось не так, то повторно перевірте всі свої IP-адреси, а також переконайтеся, що пінгували правильну адресу. Пам'ятайте, що завжди можна повернутися до DHCP, пройшовши назад виконані кроки. IP-адреса «network» маршрутизатора іноді трохи незручна, так що перевірте її, якщо все ще маєте питання.

Автор сподівається, однак, що на вашому Raspberry Pi тепер встановлена статична IP-адреса!

### Статична IP-адреса для Pi в Інтернеті

Якщо провайдер вже надає вам статичну IP-адресу, то даний розділ вам не потрібний.

Ми вже створили статичну внутрішню IP-адресу для Raspberry Pi, а це означає, що комп'ютери усередині локальної мережі завжди будуть знати, де знайти свій Pi. Але, як щодо проектів, які вимагають, щоб Raspberry Pi був підключений до Інтернету? Якщо хочете створити веб-сервер з Raspberry Pi, то люди за межами вашої мережі повинні знати, де ваш Pi «живе», що означає встановлення публічного IP, який завжди виглядає однаково.

Автор говорить "виглядає так само", тому що ми дійсно створимо псевдо-статичний IP. Ваш провайдер буде продовжувати змінювати

адресу, як звичайно. Таким чином, щоразу, коли хтось підключатиметься до Pi, сценарій, який ми збираємося написати, скаже: "Схоже, Pi переїхав! Дозвольте мені перенаправити вас на нову адресу".

Ми робимо це з Dynamic DNS<sup>22</sup> (DDNS), який підтримує сервер імен, що оновлюється в режимі реального часу, і DDClient<sup>23</sup>, програми, яка буде листуватися з DDNS прямо з вашого Pi.

Перш за все, необхідно підписатися на безкоштовний Dynamic Host сервіс, подібний DNS Dynamic<sup>24</sup>. Дотримуйтесь інструкцій і створіть новий сервер імен, типу Yourserver.dnshostdynamic.com.

Тепер перейдіть в командний рядок на Raspberry Pi і встановіть DDClient за допомогою наступного рядка:

```
sudo apt-get install ddclient
```

Нам потрібно змінити конфігурацію DDClient з новим DDNS ім'ям нашого сервера:

```
sudo nano /etc/ddclient/ddclient.conf
```

Кожний сервіс матиме дещо іншу конфігурацію, але веб-сайт DDNS повинен підказати, що ви повинні зробити, щоб налаштувати цей файл.

Стандартна конфігурація для DNS Dynamic, наприклад, виглядає наступним чином<sup>25</sup>:

```
daemon=300                # перевіряти кожні 300 секунд
syslog=yes                 # журнал повідомлень в
                           # системному журналі
mail=root                  # поштою всі повідомлення в root
mail-failure=root          # поштою повідомлення про невдачі
                           # поновлення в root
pid=/var/run/ddclient.pid  # запис PID у файл
ssl=yes                    # використовувати SSL-підтримку
                           # (працює з SSL-бібліотекою)
use=web, web=myip.dnshostdynamic.com # отримати ip з сервера
server=www.dnshostdynamic.org  # сервер за замовчуванням
login= user@gmail.com         # логін за замовчуванням
password=password             # пароль за замовчуванням
server=www.dnshostdynamic.org, \
protocol=dyndns2              \
awesome.dnshostdynamic.com
```

Ви повинні замінити логін, пароль налаштувань з вашої власної електронної пошти та пароль, який створили в DNSDynamic. Крім того,

<sup>22</sup> [http://en.wikipedia.org/wiki/Dynamic\\_DNS](http://en.wikipedia.org/wiki/Dynamic_DNS)

<sup>23</sup> <http://sourceforge.net/p/ddclient/wiki/Home/>

<sup>24</sup> <https://www.dnshostdynamic.org/>

<sup>25</sup> <http://dnshostdynamic.blog.com/2011/06/26/using-ddclient-with-dnshostdynamic/>

необхідно змінити останній рядок у файлі конфігурації, що є доменом, який ви б хотіли постійно оновлювати.

Ваш провайдер не збирається просто зупинити оновлення IP-адреси Raspberry Pi, яка в цьому файлі, однак, ми повинні покласти демон на самому верху даного сценарію, щоб сказати, що треба постійно перевіряти чи не змінилася IP-адреса:

```
daemon=600  
# перевіряти кожні 600 секунд
```

Тут IP-адрес перевіряється кожні 10 хвилин, що є просто хорошою практикою. Якщо ви встановите, щоб перевіряти щосекунди, то ваш динамічний сервіс може не оцінити це, тому що ви будете буквально спамити його сервери. Натисніть Control+X, щоб зберегти і вийти з цього файлу конфігурації.

Після цього, наберіть:

```
ddclient
```

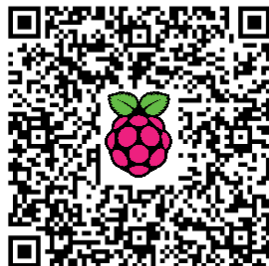
Просто ввівши ім'я програми, будете її запускати. Це продовжуватиме працювати, поки ваш Raspberry Pi включений. Якщо перезавантажите Pi, то напишіть `ddclient` для запуску його знову.

### Як налаштувати Wi-Fi через командний рядок



Ви налаштували RPi саме так, як хотіли, він встановлений і безперебійно працює, але раптом захотіли перемістити його подалі від «шнурка» Ethernet за допомогою Wi-Fi модуля.

Отож, швидко додамо підтримку Wi-Fi з командного рядка<sup>26</sup>.



### Чому ви хочете це зробити?

Якщо ви ентузіаст Raspberry Pi (або швидко стаєте ним), то вже знаєте, як неприємно може бути, коли розумієш, що проекту на Pi в даний час необхідне ще одне невелике налаштування, яке, ймовірно, вимагає підключення монітора і клавіатури/миші до плати.

Кращий спосіб не потрапити в цю пастку - налаштувати RPi для віддаленого доступу, про що мова далі. Після того, як все налаштуєте, ви повинні знати, як віддалено виконувати задачі, які раніше робилися

<sup>26</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1750-how-to-set-up-wi-fi-on-your-raspberry-pi-via-the-command-line>

через графічний інтерфейс (наприклад, включення Wi-Fi). Далі ми збираємося провести вас через технічний (але простий) спосіб віддаленого підключення до Pi і активувати додатковий модуль Wi-Fi.

### Що нам потрібно?

Нам знадобиться наступне:

- плата Raspberry Pi з встановленим Raspbian (наведений метод повинен працювати і в інших дистрибутивах, але ми використовуємо Raspbian)
- Ethernet-з'єднання з платою Pi (необхідне для активації віддаленої функціональності Wi-Fi)
- Wi-Fi модуль (ми використовуємо модель Edimax EW-7811Un (рис.2.8) на всіх наших платах Pi з великим успіхом)

Якщо ви не використовуєте дану модель модуля Wi-Fi, то настійно рекомендуємо дослідити ту модель, яку маєте намір придбати, щоб побачити, чи вона підтримується. З цією метою дуже корисний розділ RPi-вікі про USB Wi-Fi адаптери<sup>27</sup>.



Рис.2.8 Wi-Fi модуль Edimax EW-7811Un

На додаток до перерахованого, ви повинні скористатися моментом, щоб перевірити конфігурацію Wi-Fi вузла, до якого маєте намір підключити плату Raspberry Pi: необхідно записати SSID, пароль і тип шифрування/метод (наприклад, вузол використовує WPA TKIP з використанням загальних ключів шифрування).

### Включення Wi-Fi модуля через термінал

Щоб почати роботу, подайте живлення на свій Raspberry Pi поки модуль Wi-Fi ще не вставлений. На цей момент повинен бути тільки бортовий мережний пристрій Ethernet NIC (через який ви підключені до мережі за допомогою кабеля Ethernet, щоб могли отримати віддалений доступ до автономного пристрою).

Підключаємося до Pi через SSH для доступу до віддаленого терміналу (більш детально про налаштування SSH в наступному розділі).

У командному рядку введіть команду:

```
sudo nano /etc/network/interfaces
```

<sup>27</sup> [http://elinux.org/RPi\\_VerifiedPeripherals#USB\\_Wi-Fi\\_Adapters](http://elinux.org/RPi_VerifiedPeripherals#USB_Wi-Fi_Adapters)

У текстовому редакторі nano побачите щось на зразок цього:

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
```

Це дуже проста конфігурація, яка управляє підключенням Ethernet вашого Pi (позначений як `eth0`). Нам потрібно зробити дуже незначні зміни, щоб включити Wi-Fi адаптер. Використайте клавіші зі стрілками для переміщення вниз під існуючий запис і додайте наступні рядки:

```
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Після того, як відредагували файл, натисніть `Ctrl+X`, щоб зберегти його, і вийдіть з редактора nano. На запит, введіть таку команду:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Порівняйте вміст файлу, якщо він існує, з наступним кодом нижче. Якщо файл порожній, то можете використати наведений код, щоб заповнити його. Зверніть увагу на рядки коментарів (позначені знаками `#`), що вказують, які змінні ви повинні використовувати в залежності від поточної конфігурації Wi-Fi вузла.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="YOURSSID"
    psk="YOURPASSWORD"
    # Тип протоколу може бути: RSN (для WP2) або WPA (для WPA1)
    proto=WPA
    # Тип керування ключами може бути: WPA-PSK або WPA-EAP (Pre-Shared або Enterprise)
    key_mgmt=WPA-PSK
    # Pairwise може бути CCMP або TKIP (для WPA2 або WPA1)
    pairwise=TKIP
    #Параметр авторизації повинен бути OPEN для обох WPA1/WPA2 (менш широко вживаними є SHARED і LEAP)
    auth_alg=OPEN
}
```

Коли закінчите редагувати файл, натисніть `CTRL+X` для збереження і виходу з документа. Зараз настав час, щоб від'єднати кабель Ethernet і підключити Wi-Fi адаптер.

У командному рядку введіть команду `sudo reboot` для перезавантаження.

Коли пристрій завершить перезавантаження, то повинен автоматично підключитися до Wi-Fi вузла. Якщо з якихось причин він не з'явився в мережі, то завжди можете підключити кабель Ethernet назад для повторної перевірки двох файлів і змінних, які ви редагували.

### Вирішення питання з пропаданням доступу Wi-Fi

Якщо ви побачили, що модуль час від часу "випадає", переходячи з робочого режиму в сплячий режим, то можете досить легко це виправити, змінивши кілька рядків команд.

Створіть та відредагуйте новий файл в `/etc/modprobe.d/8192cu.conf`:

```
sudo nano /etc/modprobe.d/8192cu.conf
```

і вставте в нього наступне:

```
# Заборонити енергозбереження
options 8192cu rtw_power_mgnt=0 rtw_enusbss=1 rtw_ips_mode=1
```

Потім знову перезавантажтеся з `sudo reboot`

Якщо використовуєте Wi-Fi-пристрій не від Edimax, то доведеться самостійно знайти, як вимкнути керування живленням для конкретного пристрою.

### Перевірка стабільності з'єднання

Ви можете піти далі і переконатися, що з'єднання залишається стабільним, періодично відправляючи пінг-запит з вашого Pi до маршрутизатора за допомогою `crontab`:

```
crontab -e
```

`Crontab` - файл завдань, який є у кожного користувача і зберігається в каталозі `/var/spool/cron`. Файл `crontab` відповідає імені самого користувача. У самому файлі `crontab` описано як, коли і які програми запускати від імені певного користувача.

Для редагування файлу `crontab` використовується спеціальна однойменна команда `crontab`, яка дозволяє не переривати процес `cron` на час редагування:

- e — команда запускання файлу `crontab` для його редагування (edit)
- l — команда запускання файлу, щоб продивитися його зміст (list)
- r — видалити файл `crontab` для даного користувача (remove)

### Таблиця crontab

Кожний рядок у файлі crontab складається з п'яти колонок, що відокремлюються пробілами чи табуляціями. Колонки задають час виконання (Хвилина, Година, День, Місяць, День тижня), в них може міститися число, список чисел, або символ \*. Решта символів в рядку інтерпретуються як виконувана команда та її параметри.

```
# .----- хвилина (0 - 59)
# | .----- година (0 - 23)
# | | .----- день місяця (1 - 31)
# | | | .----- місяць (1 - 12) АБО jan,feb,mar,apr ...
# | | | | .---- день тижня (0 - 6) (неділя=0 чи 7) АБО
sun,mon,tue,wed,thu,fri,sat
# | | | | |
* * * * * виконувана команда
```

Приклади команд:

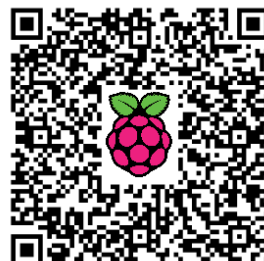
```
# # # # Тут починаються завдання
# Виконувати кожен день в 0 годин 5 хвилин, результат складати в
log/daily:
5 0 * * * $HOME/bin/daily.job>> $HOME/log/daily 2> & 1
# Виконувати першого числа кожного місяця:
15 14 1 * * $HOME/bin/monthly
# Виконувати кожного робочого дня о 22:00:
0 22 * * 1-5 echo "Пора додому"
23 */2 * * * echo "Виконується о 0:23, 2:23, 4:23 і т. д."
5 4 * * sun echo "Виконується о 4:05 в неділю"
0 0 1 1 * echo "З новим роком!"
15 10,13 * * 1,4 echo "Цей напис виводиться в понеділок і четвер
о 10:15 та 13:15"
```

Додайте наступне в кінець файлу:

```
*/1 * * * * ping -c 1 192.168.1.2
```

Тільки замініть 192.168.1.2 на IP свого маршрутизатора. IP маршрутизатора можна знайти, відкривши командний рядок в Windows і виконавши команду `ipconfig`, та перевірте числа під "default gateway".

**Як з Raspberry Pi зробити  
маршрутизатор Wi-Fi**





Хочете мати окрему мережу Wi-Fi для своїх гостей? Або домашню мережу, яка працює з альтернативною конфігурацією DNS?

Вам більше не доведеться купувати новий Wi-Fi маршрутизатор, щоб зробити це - ви можете перетворити свій Raspberry Pi (RPi) в Wi-Fi маршрутизатор<sup>28</sup>.

Ми проведемо вас через процес налаштування RPi як мережі Wi-Fi.

### Передумови та обладнання

Вам потрібні:

- Raspberry Pi
- USB Wi-Fi адаптер (наприклад, Edimax EW-7811Un)
- SD-карта з встановленою ОС Raspbian
- Доступ до RPi як за допомогою клавіатури і монітора, так і віддалено<sup>29</sup>

Перш, ніж ми приступимо, хочемо вказати на важливість покупки правильного USB Wi-Fi адаптера. Як ви, можливо, відчули з іншими типами пристроїв, що не всі вони plug-n-play. Іноді треба завантажити драйвер, щоб змусити їх працювати. У той час коли драйвери, як правило, легко доступні для комп'ютерів Windows, - це зовсім інший світ для Linux і RPi. Ось чому дуже важливо купити Wi-Fi адаптер, який згадує "Linux" в описі продукту або пакета. Автор використовує Edimax - бездротовий 802.11b/g/n nano USB адаптер, з яким легко працювати.

### Переконайтеся, що Wi-Fi адаптер працює на Raspberry Pi

Вставте адаптер USB Wi-Fi в один з вільних портів USB на RPi. Подайте живлення на RPi. У цей момент ви повинні або підключитися до RPi за допомогою кабелю Ethernet, або використовувати клавіатуру і монітор для доступу до RPi. Якщо потрібна допомога з зовнішнього підключення до RPi, то перейдіть до наступного розділу, в якому розглядається, як віддалено отримати зручний захищений доступ до RPi.

Після завантаження та реєстрації ви повинні переконатися, що RPi знайшов свій новий бездротовий адаптер. Щоб подивитися, яка периферія знайдена операційною системою при завантаженні, виконайте наступну команду:

```
dmesg | more
```

<sup>28</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1801-how-to-make-raspberry-pi-wi-fi-router>

<sup>29</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1760-ssh-for-easy-secure-access-to-raspberry-pi>

Ви можете використовувати пробіл, щоб прокрутити сторінку вниз за один раз - в кінці побачите щось схоже на наступні рядки:

```
[ 3.282651] usb 1-1.2: new high-speed USB device number 4 using
dwc_otg
[ 3.394810] usb 1-1.2: New USB device found, idVendor=7392,
idProduct=7811
[ 3.407489] usb 1-1.2: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 3.420530] usb 1-1.2: Product: 802.11n WLAN Adapter
```

Це означає, що операційна система визнала USB Wi-Fi адаптер за допомогою одного з вбудованих драйверів (для повернення до терміналу натисніть Q). Тепер залишилося налаштувати підключення Wi-Fi.

### Встановлення програмного забезпечення маршрутизатора

Для того, щоб RPi виступав як Wi-Fi роутер і точка доступу, необхідно встановити деяке додаткове програмне забезпечення (ПЗ) на RPi:

- **hostapd**

hostapd є службою (демоном) користувача для серверів точок доступу і аутентифікації. Це означає, що він може перетворити ваш RPi в точку доступу й інші комп'ютери зможуть підключитися. Він також займатиметься безпекою, щоб можна було налаштувати Wi-Fi з паролем.

- **isc-dhcp-server**

isc-dhcp-server є реалізацією Internet Systems Consortium DHCP-сервера. DHCP-сервер відповідає за призначення адрес комп'ютерам і пристроям, підключеним до точки доступу Wi-Fi.

Щоб встановити ПЗ для DHCP виконайте наступну команду:

```
sudo apt-get install isc-dhcp-server
```

Далі необхідне програмне забезпечення для HostAPD. Тому, що необхідний драйвер точки доступу нашого USB-модуля не підтримує рідний додаток HostAPD, нам необхідно встановити спеціальну версію, компільовану з драйвером, який потрібний. Це можна зробити за допомогою наступних команд, виконаних одна за одною:

```
wget https://github.com/jenssegers/RTL8188-
hostapd/archive/v1.1.tar.gz
tar -zxvf v1.1.tar.gz
cd RTL8188-hostapd-1.1/hostapd
sudo make
sudo make install
```

Це займе трохи часу. Після завантаження ми готові налаштувати програмне забезпечення.

### Налаштування ISC-DHCP-сервера

Для налаштування DHCP-сервера відкрийте файл `/etc/dhcp/dhcpd.conf` у своєму улюбленому текстовому редакторі. Ви можете відкрити його в `nano` за допомогою команди:

```
sudo nano /etc/dhcp/dhcpd.conf
```

Знайдіть розділ, який нижче, і закоментуйте його, поставивши хештег на початку рядка:

```
option domain-name "example.org";  
option domain-name-servers ns1.example.org, ns2.example.org;
```

Далі знайдіть наведений нижче розділ і розкоментуйте слово авторизації (видаліть хештег #):

```
# Якщо даний DHCP сервер є офіційним сервером DHCP для  
# локальної  
# мережі, то директорія authoritative не повинна бути  
# закоментованою.  
#authoritative;
```

Тепер файл виглядає наступним чином (рис.2.9):

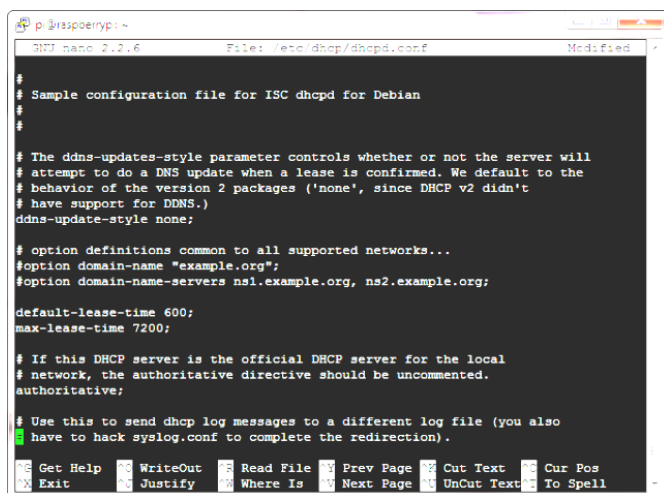


Рис.2.9. Редагування файла `dhcpd.conf`

Далі нам потрібно вказати мережу і мережеві адреси, які DHCP-сервер буде обслуговувати. Це робиться шляхом додавання наступного блоку конфігурації в кінець файлу:

```
subnet 192.168.10.0 netmask 255.255.255.0 {
```

```
range 192.168.10.10 192.168.10.20;  
option broadcast-address 192.168.10.255;  
option routers 192.168.10.1;  
default-lease-time 600;  
max-lease-time 7200;  
option domain-name "local-network";  
option domain-name-servers 8.8.8.8, 8.8.4.4;  
}
```

Це дозволить DHCP-серверу роздавати IP-адреси з 192.168.10.10 до 192.168.10.20 у власній локальній мережі. Фахівці з конфігурації мереж можуть змінити ці значення, якщо хочуть використовувати інші мережеві адреси та/або інші сервери DNS. Наведена конфігурація використовує DNS-сервери Google на 8.8.8.8 і 8.8.4.4.

Щоб зберегти файл, натисніть Ctrl+O, щоб записати файл на диск - потім можна вийти з nano, натиснувши Ctrl+X. Якщо nano запитає, чи хочете зберегти змінений буфер, то натисніть Y, а потім натисніть Enter для підтвердження імені файлу.

Наступний файл для редагування `/etc/default/isc-dhcp-server`, який можете відкрити в nano за допомогою наступної команди:

```
sudo nano /etc/default/isc-dhcp-server
```

Перейдіть до рядка, який згадує про інтерфейси, і змініть рядок, щоб він виглядав так:

```
INTERFACES="wlan0"
```

Це дозволить DHCP-серверу роздавати мережеві адреси через бездротовий інтерфейс. Збережіть файл і вийдіть з nano.

Останнім кроком у налаштуванні DHCP-сервера необхідно налаштувати статичну IP-адресу для адаптера бездротової мережі. Це робиться у файлі `/etc/network/interfaces` - перед його відкриттям переконайтеся, що інтерфейс WLAN не працює. Зробіть це за допомогою наступних команд:

```
sudo ifdown wlan0  
sudo nano /etc/network/interfaces
```

Змінений файл повинен виглядати як на рис.2.10.

Це дозволить бездротовому адаптеру взяти адресу 192.168.10.1 в нашій новій локальній мережі. Не забудьте закоментувати останні 3 рядки файлу.

На цьому налаштування DHCP-сервера завершене, однак, ми все ще не можемо підключитися до нашої нової мережі, бо не встановлена точка доступу. Це робиться шляхом налаштування програми `hostapd`.

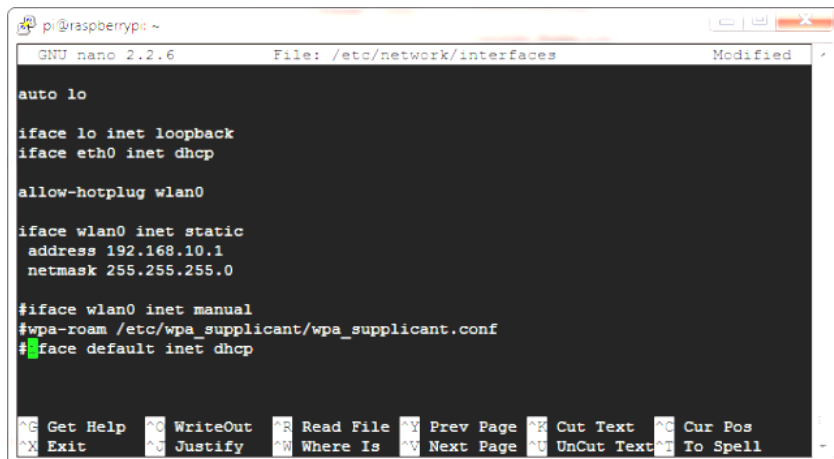


Рис.2.10 Змінений файл `interfaces`

### Налаштування HostAPD

Щоб налаштувати HostAPD, відкрийте файл `/etc/hostapd/hostapd.conf` у nano за допомогою команди:

```
sudo nano /etc/hostapd/hostapd.conf
```

Стандартна конфігурація створить нову бездротову мережу під назвою `wifi` з паролем `YourPassPhrase`. Ви можете змінити параметр `"ssid=wifi"` на ім'я WiFi SSID, яке хочете, а в параметрі `"wpa_passphrase=YourPassPhrase"` вказати свій власний пароль.

Це завершує конфігурацію програмного забезпечення HostAPD нашої точки доступу. Наступний буде дозвіл NAT.

### Включення NAT

Перш ніж ми зможемо запустити точку доступу, необхідно налагодити трансляцію мережевих адрес (Network Address Translation - NAT). Це дозволить нашому мережевому трафіку досягти Інтернету, використовуючи з'єднання через Ethernet-кабель RPi з інтернет-маршрутизатором.

Відкрийте `etc/sysctl.conf` за допомогою nano:

```
sudo nano /etc/sysctl.conf
```

Перейдіть до останнього рядка файлу і додайте рядок:  
`net.ipv4.ip_forward=1`

Наступне - запустить трансляцію відразу командою:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Запустить бездротову мережу через:

```
sudo ifup wlan0
```

Наступним кроком є створення фактичної трансляції між портом локальної мережі під назвою `eth0` і бездротової карти під назвою `wlan0`.

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

З налаштуванням NAT настав час для першого запуску точки доступу.

### Запуск власного бездротового маршрутизатора

Тепер ви готові запустити DHCP-сервер і додаток HostAPD точки доступу. Це можна зробити командою:

```
sudo service isc-dhcp-server start
sudo service hostapd start
```

У цей момент ви повинні бути в змозі знайти свою бездротову мережу на ноутбучі, підключитися до неї і вийти в Інтернет.

### Завершення...

Тепер, як дуже чудово, що у вас є своя RPi, яка працює як бездротова точка доступу, але не все так добре, бо ви повинні вводити логін щоразу, коли перезавантажуєтеся, щоб запустити програмне забезпечення HostAPD і DHCP.

Щоб уникнути цього, виконайте наступні команди:

```
sudo update-rc.d hostapd enable
sudo update-rc.d isc-dhcp-server enable
```

Щоб уникнути необхідності налаштовувати NAT щоразу, коли перезавантажуєте RPi, можете зробити наступне.

Виконайте дану команду для резервного копіювання NAT конфігурації:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Додайте наступне в кінець файлу `/etc/network/interfaces` для відновлення конфігурації, коли підніметься мережевий інтерфейс:

```
up iptables-restore < /etc/iptables.ipv4.nat
```

Файл повинен виглядати так:

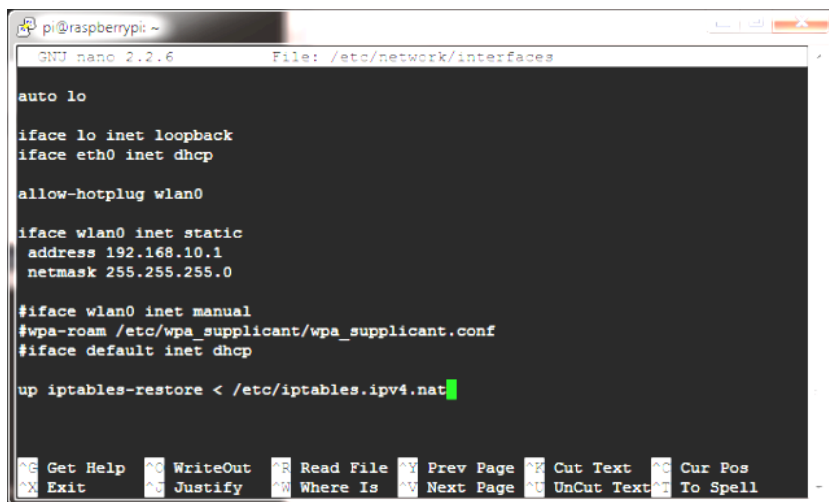


Рис.2.11 Кінцевий варіант файла `interfaces`

У цей момент спробуйте перезавантажити RPi, щоб переконатися, що все працює як годиться - можна перезавантажити за допомогою команди:

```
sudo reboot
```

Сподіваюся, ви знову побачите свою нову точку доступу в онлайн. Після того, як вона підніметься, зможете підключитися до неї і отримати доступ до Інтернету.

### Як виправити помилку “device not managed”

За замовчуванням, в Kali Linux на RPi Network Manager не управляє ніяким інтерфейсом, визначеним в `/etc/network/interfaces`.

Некеровані пристрої недоступні Network Manager і саме тому він не керує жодним з мережевих пристроїв. В GUI це відображається як на рис. 2.12:

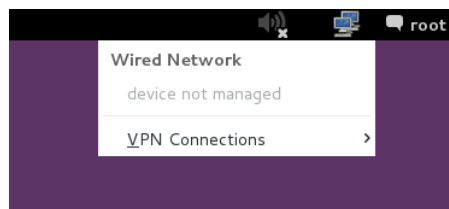
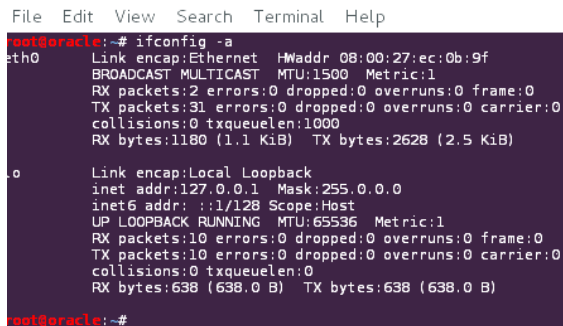


Рис.2.12 Пристрій не керований

Запустіть команду `ifconfig -a`, і ви побачите зображене на рис.2.13.

Це відбувається, коли дотримані дві умови:



```
File Edit View Search Terminal Help
root@oracle:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:ec:0b:9f
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KiB)  TX bytes:2628 (2.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:638 (638.0 B)  TX bytes:638 (638.0 B)

root@oracle:~#
```

Рис.2.13 Параметри мережі

- Файл `/etc/network/interfaces` нічого не містить про інтерфейс, навіть немає:

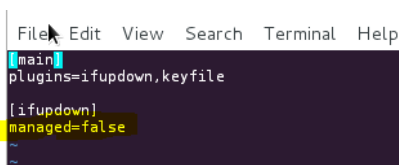
```
allow-hotplug eth0
iface eth0 inet dhcp
```

- Файл `/etc/NetworkManager/NetworkManager.conf` містить:

```
[main]
plugins=ifupdown,keyfile

[ifupdown]
managed=false
```

що відображено на скріншоті нижче (рис.2.14):



```
File Edit View Search Terminal Help
[main]
plugins=ifupdown,keyfile

[ifupdown]
managed=false
```

Рис.2.14 Заборона керування

Якщо хочете скористатися Network Manager для обробки інтерфейсів, то додайте їх в `/etc/network/interfaces`. Для цього встановіть `managed=true` в `/etc/NetworkManager/NetworkManager.conf`. Так щоб цей файл виглядав наступним чином (рис.2.15):

```
[main]
plugins=ifupdown,keyfile
```

```
[ifupdown]
managed=true
```

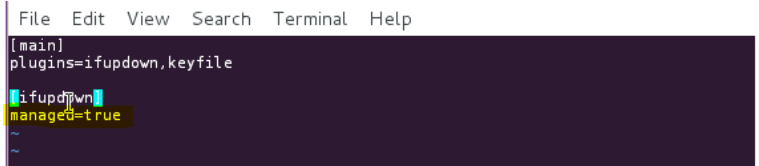


Рис.2.15 Налаштування файлу конфігурації

Перезапустіть менеджер мережі з наступною командою:

```
service network-manager restart
```

Тепер Network Manager повинен підхопити підключений інтерфейс. Для дротового з'єднання eth0 з DHCP вам буде показано щось на кшталт такого (рис.2.16).

Щоб переконаватися, що все гаразд, виконайте ще раз `ifconfig -a` з командного рядка (рис.2.17):

Все виглядає чудово і ви маєте можливість підключитися до мережі та переглядати Інтернет.

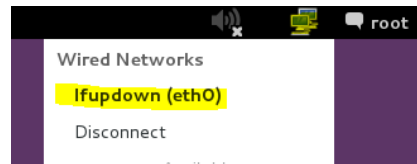


Рис.2.16 Network Manager керує

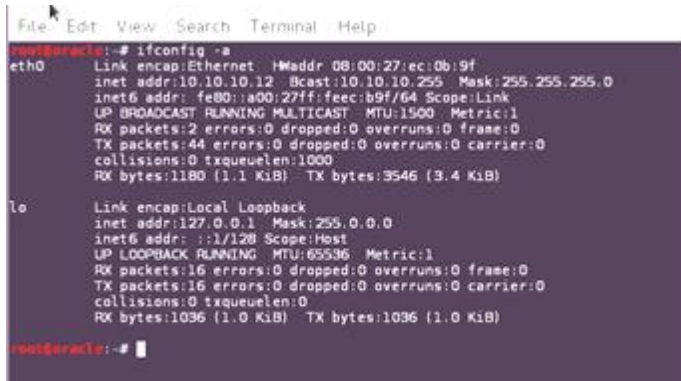


Рис.2.17 Змінені параметри мережі

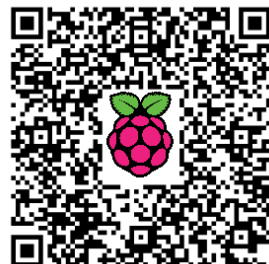
### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

#### SSH для зручного захищеного доступу до Raspberry Pi



Інтернет - вороже навколишнє середовище і безпека є обов'язком кожного. Рекомендуємо скористатися SSH (Secure Shell)<sup>30</sup>, якщо підключаєтеся через відкритий Інтернет,

Встановити з'єднання SSH можна з PuTTY - програмою, яка дозволяє працювати на віддаленому Raspberry та інших комп'ютерах з ОС Linux з комп'ютера Windows PC, що особливо важливо, враховуючи кількість користувачів останньої ОС.



#### Підготовка та встановлення

Спочатку підключимо до Raspberry Pi живлення та Інтернет.

Тепер необхідно дізнатися IP-адресу Raspberry Pi, бо вона зазвичай надається динамічно через DHCP. Звичайно, вона вам вже відома, якщо ви встановили статичну IP-адресу.

Однак, при динамічній адресації маршрутизатор наступного разу може призначити іншу IP-адресу. Це проблема, тому для того, щоб підключитися через SSH, ми повинні знати IP-адресу Raspberry Pi у вашій локальній мережі. Ми не зможемо побачити її, якщо не матимемо моніторингу приєднаних пристроїв. Необхідно сканувати мережу, щоб знайти IP-адресу Raspberry Pi.

Звичайно, ми можемо дізнатися IP-адресу Raspberry Pi, увійшовши в маршрутизатор і побачивши, які пристрої підключені. Але є простіший спосіб: за допомогою чудового безкоштовного інструмента під назвою Advanced IP Scanner. Завантажити його можна з [radmin.com](http://www.radmin.com)<sup>31</sup>.

Коли запустите IP Scanner, то побачите екран, схожий на наведений нижче (рис.3.1). Натисніть кнопку Scan (Сканувати):

Серед відображених приєднаних до мережі пристроїв ви без проблем знайдете свій Raspberry Pi та

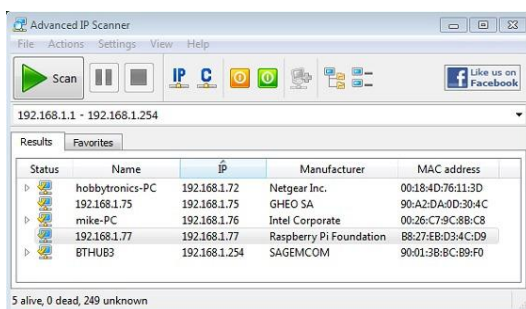


Рис.3.1 Вікно IP Scanner

<sup>30</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1760-ssh-for-easy-secure-access-to-raspberry-pi>

<sup>31</sup> <http://www.radmin.com/products/ipscanner/>

його IP-адресу.

На наступному кроці переконайтеся, що сервер SSH на Raspberry Pi працює. Для цього введіть в терміналі сесії команду:

```
sudo service ssh status
```

Якщо сервер SSH запущений, то ви повинні отримати відповідь, що сервіс sshd працює. Якщо це не так, то введіть в термінальному сеансі:

```
sudo raspi-config
```

Запуститься утиліта конфігурування Raspbian, яка входить в дистрибутив. Виберіть пункт меню для SSH, а потім виберіть дозвіл на включення сервера SSH. Перезавантажте Raspberry Pi.

Для використання основних можливостей PuTTY на сайті PuTTY доступні для завантаження п'ять файлів<sup>32</sup>:

- **PuTTY.exe** - клієнт Secure Shell
- **PuTTYgen.exe** - генератор публічного/приватного ключів SSH
- **Pagent.exe** - агент аутентифікації, зберігає ключі в пам'яті і при його використанні не треба щоразу вводити ключову паролъну фразу
- **PSCP.exe** – для захищеного копіювання з командного рядка
- **PSFTP.exe** – для захищеного копіювання з FTP-подібним інтерфейсом

В даному розділі ми скористаємося першими трьома, а використання останніх двох файлів розглянемо пізніше.

Для уникнення проблем при встановленні, переконайтеся, що каталог встановлення знаходиться в шляху для команди.

У Windows 7 клацніть правою кнопкою миші на Комп'ютер в меню Старт і оберіть пункт Властивості. Внизу вікна натисніть кнопку Параметри середовища. Відкриється діалогове вікно (рис.3.2).

Завжди існує змінна Path в розділі *Системні змінні*, а іноді також і в розділі *Змінні середовища користувача*. Лише адміністратори мають доступ до системних змінних, щоб змінити або додати Path, при необхідності.

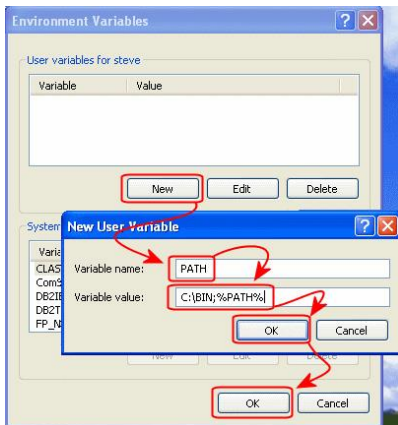


Рис.3.2 Змінні середовища

<sup>32</sup> [www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

Ми поставимо нову папку типово на початку шляху і вона буде відокремлена від решти частини списку комою. Натисніть OK, щоб зберегти всі зміни.

Встановлення PuTTY не має особливостей. Після закінчення встановлення створіть ярлик на робочому столі, бо зазвичай ця програма часто використовується. В нашому випадку, шлях в ярлику до виконуваного файлу C:\Bin\putty.exe.

#### Запуск PuTTY та налаштування для цільової системи

- **Сесія:**

Ім'я вузла (Host Name) : *ім'я вузла, наприклад, dbserver*, або IP-адреса віддаленого комп'ютера

Протокол (Protocol): SSH

- **З'єднання: Дані (Data)**

Користувач з автологіном: *наприклад, steve*

- **З'єднання: SSH**

Віддати перевагу SSH протоколу версії: тільки 2

- **Вікно: Переклад (Translation)**

За замовчуванням встановлено ISO-8859-1, тому повинні відкрити список і вибрати UTF-8.

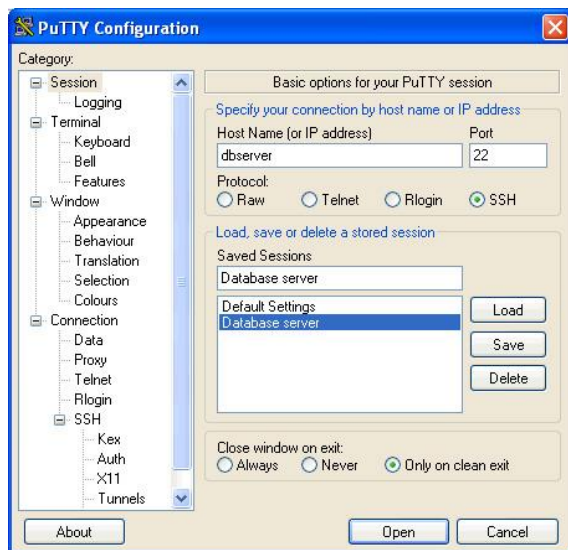


Рис.3.2 Вікно конфігурації PuTTY

Вводимо відповідні дані, зберігаємо сесію (Save), натискаємо Open (рис.3.2), щоб відкрилося вікно авторизації (рис.3.3), і для входу використовуємо ім'я і пароль користувача (за замовчуванням для Raspberry: **pi** і **raspberry**)



Рис.3.3 Вікно авторизації

При доступі до декількох віддалених систем можна створити і використовувати декілька сеансів, відповідним чином відредагувавши ярлики запуску збережених сеансів. Для цього клацніть правою кнопкою миші на ярлику і оберіть Властивості, потім введіть параметр `-load` разом з ім'ям сесії (у лапках, якщо необхідно):

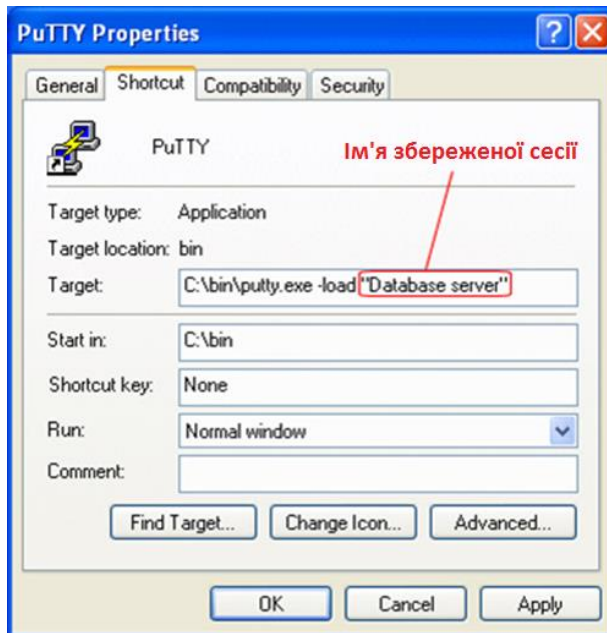


Рис.3.4 Створення ярлика для запуску

### Створення, встановлення та використання публічних/приватних ключів

Для шифрування в PuTTYgen пропонуються на вибір два стандарти шифрування: DSA і RSA (рис.3.5).

DSA або Digital Signature Algorithm - стандарт американського уряду, розроблений NSA для цифрових підписів. Він є похідним від алгоритму підпису Ель-Гамала. Безпека будується на основі складності вирішення певних типів логарифмічних функцій.

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

RSA названий від ініціалів авторів: Ron Rivest, Adi Shamir і Leonard Adleman, які першими опублікували алгоритм. Він заснований на складності факторизації великих чисел.

**Не створюйте ключі будь-якого типу SSH версії 1: вони не є безпечними!**

Хоча DSA і RSA мають криптографію практично однакової потужності, але у кожного є свої переваги, коли мова заходить про продуктивність. DSA швидше працює для розшифровки і підписання, а RSA - швидше при шифруванні та перевірці.

Тому на ваш вибір, але рекомендується ключ від 2048 до 4096 біт. Більш короткі ключі можуть бути недостатньо стійкими, а більш довгі занадто довго генеруються.

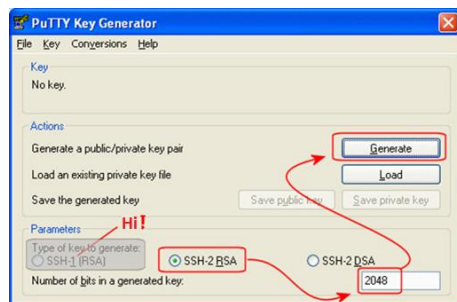


Рис.3.5 Генератор ключів PuTTY

### Захист і збереження ключів

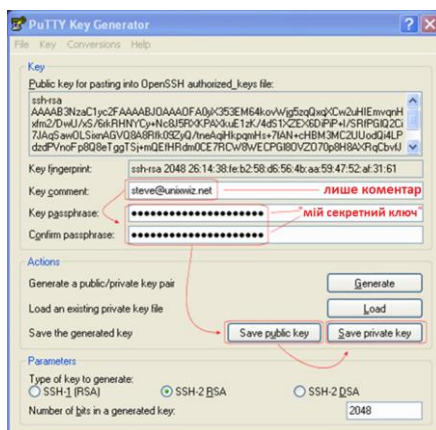


Рис.3.6 Захист ключа пароллюю фразою

Для захисту створеного ключа вводимо паролню фразу (рис.3.6). **Не забувайте паролню фразу: пара ключів без неї марна!**

### Встановлення публічного ключа на RaspberryPi

З ще відкритим PuTTYgen, виділяємо під заголовком «Публічний ключ для вставки в OpenSSH файл authorized\_keys» всю площу і через Ctrl+C копіюємо в буфер обміну (рис.3.7).

По суті, це ті ж дані, які містяться в збереженому файлі публічного ключа, але у формі, яка може бути безпосередньо використана на Raspberry.

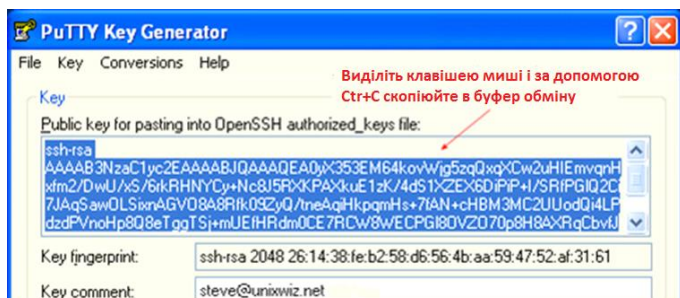


Рис.3.7 Копіювання публічного ключа

Входимо на Raspberry за допомогою облікового запису, створюємо каталог SSH, якщо необхідно:

```
$ mkdir ~/.ssh
```

Редагуємо або створюємо файл з іменем `~/.ssh/authorized_keys`. Це буде текстовий файл і текст з буферу обміну повинен бути вставлений в нього (рис.3.8).



Рис.3.8 Створення файла ключа

Відкритий ключ має тільки один довгий рядок, і дуже легко вставити дані таким чином, що обрізаються перші кілька символів. Це робить ключ непрацездатним, тому спочатку переконаєтеся, що ключ починається з SSH-RSA або DSA-SSH. Збережіть файл.

Також подбаємо про захист збереженого файла. Переконайтеся, що каталог SSH і файли в ньому може прочитати тільки їх власник - це міра безпеки, яка може бути досягнута за допомогою параметрів команди `chmod` із застосуванням до всього каталогу:

```
$ chmod 700 ~/.ssh
```

Для директорій зі всього списку зазвичай застосовують лише параметри 0, 5 і 7 - повна заборона, читання і повний доступ.

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

В нашому випадку 700 (`-rwx-----`) означає, що власник може читати, записувати і запускати для виконання, а ніхто інший не має права виконувати будь-які дії.

Виконуємо перевірку за допомогою:

```
$ ls-lR ~/.ssh
```

Результати виконання команди:

```
/home/serg/.ssh:
total 16
drwx-----  2 steve  serg      4096 Nov 22 13:11 ./
drwx-----  6 steve  serg      4096 Nov 22 16:10 ../
-rw-----  1 steve  serg      1150 Nov 22 13:11
authorized_keys
```

Тепер можна вийти з системи.

### Прикріплення приватного ключа сесії SSH

Тепер, коли публічні/приватні ключі створені, вони можуть бути пов'язані з сесією SSH. По-перше, ми зробимо це в PuTTY, запустивши програму та завантаживши необхідну сесію.

Перейдіть до Connection (З'єднання) > SSH > панель Auth в категорії ліворуч, потім заповніть файл приватного ключа для поля аутентифікації, перейшовши до файла .ppk, збереженого раніше (рис.3.9). Поверніться до рівня категорії сесії і збережіть поточну сесію.

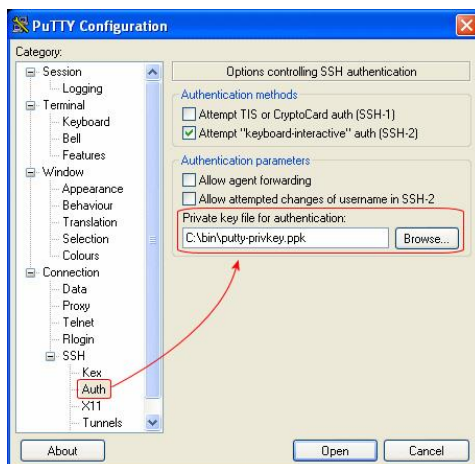


Рис.3.9 Прикріплення приватного ключа

На даний момент, клієнт PuTTY (на Windows) і сервер OpenSSH (на Rasbian) налаштовані на безпечний доступ з публічним ключем.

## Підключення через публічний ключ

Запустіть PuTTY з можливістю завантажити збережену сесію з приватним ключем:

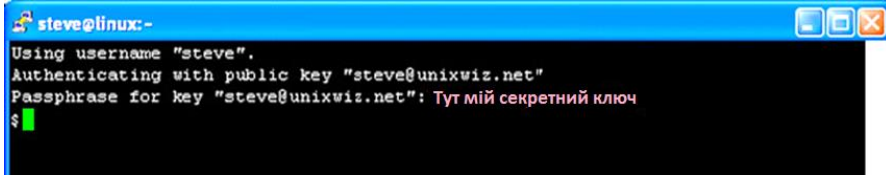


Рис.3.10 Введення паролльної фрази

Замість того, щоб вводити пароль облікового запису (який відрізнятиметься на кожній віддаленій системі), система просить паролльну фразу (рис.3.10), яка захищає локальний приватний ключ. Коли приватний ключ вписується в публічний ключ на сервері OpenSSH, то дозволяє доступ і надає оболонку користувачеві.

Важливо відзначити, що при тому, що користувач повинен ввести секретне слово при вході в систему, фраза пов'язана з локальним приватним ключем, а не з віддаленим обліковим записом. Навіть, якщо публічний ключ користувача встановлений на 1000 різних віддалених серверах, вимагається одна і та ж фраза для приватного ключа для всіх них. Це значно спрощує завдання запам'ятовування облікових даних для доступу і робить його більш безпечним.

## Відключення паролльної аутентифікації на OpenSSH

Після того, як публічний і приватний ключі користувача перевірені як правильні, можна повністю відключити паролльну аутентифікацію. Це виключає всі можливі спроби підбору пароллю і різко зростає безпека.

Тим не менше, для машин, які фізично не локальні, буде доцільним відкласти відключення аутентифікації, поки не буде зовсім ясно, що ключ доступу працює нормально, особливо, коли є кілька користувачів. Після того, як паролльна аутентифікація буде відключена, навіть пароль адміністратора не дозволить увійти в систему.

Ці нові властивості для публічного ключа доступу рекомендується дуже уважно протестувати.

Налаштуванні демона SSH можна знайти у файлі `sshd_config`, який часто зберігається в директорії `/etc/ssh/`. В цьому текстовому файлі, який відносно легко прочитати, ми шукатимемо два записи, які треба змінити.

Перший полягає у створенні для `PasswordAuthentication` значення `no`. Воно може бути явно встановлене в `yes`, або бути закоментоване, щоб використати значення за замовчуванням, але ми хочемо явно відключити його.

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

---

По-друге, ми хочемо відключити SSH протокол версії 1: він старий, має кілька суттєвих слабкостей в безпеці і не повинен бути дозволений для зовнішнього використання.

Змінюємо файл конфігурації і переконуємося, що два записи ключових слів встановлені правильно, а старі записи у файлі `/etc/ssh/sshd_config` закоментовані:

```
# Protocol1,2
Protocol 2
PasswordAuthentication no
```

Як тільки файл конфігурації буде збережений, демон безпечної оболонки повинен бути перезапущений; на більшості платформ це може бути зроблено з механізмом `"service"`:

```
# service sshd restart
```

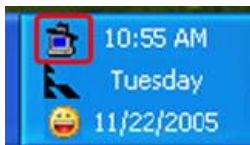
Ми вбиваємо демона прослуховування і перезапускаємо його, але не припиняємо будь-які існуючі індивідуальні сесії користувача. Тим, хто вважають це ризикованим кроком, пропонується просто перезавантажити машину.

На даний момент, OpenSSH більше не прийматиме паролі будь-якого виду, доступ надається тільки для користувачів із заздалегідь встановленими публічними ключами.

#### Дозвіл підтримки агента SSH

До цього моменту, ми приділили велику увагу безпеці доступу до системи, але він, як і раніше, не дуже зручний: ми, як і раніше, щоразу повинні вводити комплексну парольну фразу. Це може стати виснажливим, якщо використовується велика кількість систем.

На щастя, набір SSH забезпечує прекрасний механізм для розблокування приватного ключа один раз, і дозволяє окремі з'єднання SSH без запиту на введення пароля щоразу.



Знайдіть і запустіть програму Pageant.exe з того ж місця, що й інші пов'язані з PuTTY файли, а він поставить себе в системному треї (у правому нижньому кутку поряд з годинником).

Двічі клацніть на іконці в треї і запуститься діалогове вікно з порожнім списком ключів. Виберіть пункт AddKey (Додати ключ) і перейдіть в каталог `.ppk` файла, який містить приватний ключ. Коли буде запит на введення пароля, введіть його і натисніть кнопку OK. Натисніть кнопку Close (Закрити), щоб закрити агента.

Тепер запустіть одну з уже налаштованих сесій SSH до віддаленого хоста, який використовує публічний ключ: він запитає

агента для приватного ключа, віддалено обміняється з ним і надасть доступ без подальшого втручання користувача.

*Примітка:* Вдумливий читач може тільки дивуватися, як агент зберігає дані, і чи не можуть ненадійні програми потайки отримати цей приватний ключ. Ми не знаємо, як це працює, але ніколи не чули про реальні проблеми безпеки з цього питання.

#### Попереднє завантаження приватного ключа

Перше, що багато користувачів PuTTY робить, коли входить в систему протягом дня, є запуск агента і додавання приватного ключа. Це всього декілька кроків, але ми можемо оптимізувати їх трохи більше. Якщо ми запустимо агент з файлом приватного ключа як параметром, то він завантажуватиме ключ автоматично.

Перейдіть до Pageant.exe і клацніть правою кнопкою миші, щоб скопіювати його значок. Вставте значок як ярлик на робочому столі, потім клацніть правою кнопкою миші і виберіть Властивості. Введіть повний шлях до .ppk файлу приватного ключа як параметру і збережіть зміни (рис.3.11).

Двічі клацнувши на цьому значку, ви завантажуватимете файл

ключів, який вимагає парольну фразу. Ввівши її один раз, це буде останній раз, коли це необхідно зробити, поки агент залишатиметься прикріпленим.

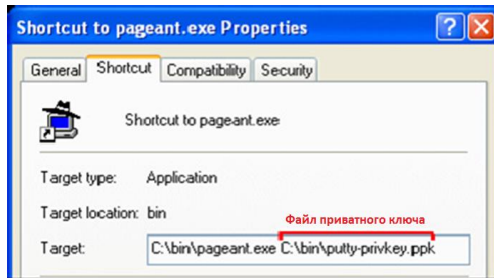


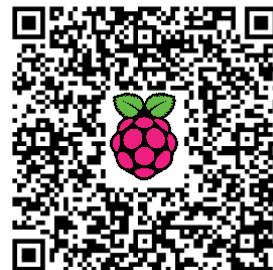
Рис.3.11 Налаштування Pageant.exe

#### Безпечний обмін файлами з Raspberry Pi



Обмін файлами між Raspberry Pi і віддаленим ПК повинен бути не лише швидким, але і надійно захищеним. Є кілька способів цього досягти<sup>33</sup>.

Раніше ми розглянули створення захищеного доступу з використанням протоколу SSH. SSH надає кілька методів для копіювання файлів з одного комп'ютера на інший, які всі працюють з тими ж ключами і агентами.



<sup>33</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1761-secure-file-sharing-with-raspberry-pi>

PSCP дозволяє командний рядок для копіювання файлів з віддаленого сервера SSH, а PSFTP забезпечує FTP-подібний інтерфейс для зручної передачі файлів. Завантажимо PSCP і PSFTP з сайту PuTTY і розглянемо їх роботу.

#### PSFTP - FTP-подібний клієнт

**PSFTP**<sup>34</sup> може бути запущений з командного рядка або за допомогою ярлика на робочому столі, і в обох випадках приймає ім'я хоста або назву збереженої сесії.

При запуску PSFTP підключається до цільового сервера (повністю скориставшись відкритими ключами та локальним агентом, якщо такі є), і надає підказку `psftp>` :

```
C> psftp dbserver
Using username "steve".
Remote working directory is /home/steve
psftp>
```

Регулярним користувачам FTP-клієнтів з командним рядком це знайомо, хоча, звичайно, не настільки просте у використанні, як популярні графічні клієнти. Команда `help` може дати деякі орієнтири.

#### PSCP – безпечне копіювання (Secure Copy)

Користувачі командного рядка, можливо, побажають скопіювати файли безпосередньо, а це можна зробити з `pscp` - командою безпечного копіювання. Так само, як копіювання звичайних файлів у локальній файлової системі, `pscp` приймає ім'я машини і директорії як джерело або місце призначення.

`pscp` може передавати один файл за раз, або цілий набір в одному примірнику, наприклад:

```
C> pscp *.gbk dbserver:/db/evolution
CL_100.gbk      |      97 kB |   97.4 kB/s | ETA: 00:00:00 | 100%
CL_101.gbk      |      68 kB |   68.2 kB/s | ETA: 00:00:00 | 100%
CL_103.gbk      |      44 kB |   44.5 kB/s | ETA: 00:00:00 | 100%
CL_110.gbk      |      34 kB |   34.6 kB/s | ETA: 00:00:00 | 100%
CL_123.gbk      |      45 kB |   45.4 kB/s | ETA: 00:00:00 | 100%
```

Цікаво, що не повинна вводиться назва збереженої сесії: лише ім'я хоста і поточне ім'я користувача (які зазвичай беруться автоматично з навколишнього середовища). Схоже, що `psftp` і `pscp` звертаються до списку збереженої сесії, знаходять відповідний розділ, а потім використовують для доступу до пов'язаної інформації.

---

<sup>34</sup> <http://the.earth.li/~sgtatham/putty/0.63/x86/psftp.exe>

## Midnight Commander

Midnight Commander (mc) - консольний файловий менеджер і браузер репозиторію. Це дружня система для вирішення багатьох завдань у вікні терміналу, і найшвидший спосіб для копіювання, переміщення або видалення декількох файлів. mc також може виконувати швидку передачу файлів FTP і мережею. Іншими унікальними особливостями є можливість переглядати всередині архівів і відновлювання файлів.

Як правило, Midnight Commander входить в пакет встановлення ОС і запускається введенням mc у вікні терміналу.

Якщо mc не встановлений, то встановлюємо, як зазвичай:

```
sudo apt-get install mc
```

Основне вікно має дві панелі директорій, з випадаючим меню на рядку вгорі, командний рядком знизу і, ще нижче, списком представлених функціональних клавіш F (функція). Над командним рядком - хіти, які показують випадкові поради (рис.3.12).

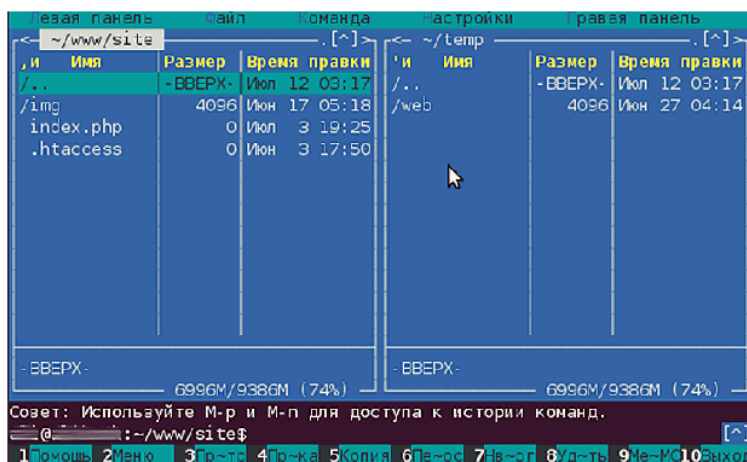


Рис.3.12 Вікно Midnight Commander

Як правило, ви хочете відобразити різні директорії на кожній з бокових панелей, щоб можна було виконувати дії над файлами між ними. Переміщуйтеся в mc за допомогою клавіатури:

- Tab, щоб перейти до іншої панелі директорії
- прокрутка директорій за допомогою клавіш зі стрілками вгору/вниз
- Home і End, щоб перейти до верхньої або нижньої частини довгої директорії

- pg-up і pg-down, щоб перейти за раз на один екран
- клавіші назад/вліво для переходу в батьківську директорію

Ключі 'F' широко використовуються в tc для роботи з файлами. Читайте на панелі в нижній частині про їх поточну функцію, яка може відрізнитися в залежності від контексту, наприклад, перегляд директорії, використання переглядача файлів або редактора.

У нормальному режимі перегляду:

- F1 - допомога. Більш читабельна, ніж довідкова сторінка з 2000 рядками, хоча і важка для перегляду.
- F2 - меню користувача (пропонує можливість gzip файлів і т.д.)
- F3 - переглянути (зручно, щоб перевірити вміст файлів rpm чи tgz, або читати вміст файлів)
- F4 - редагувати за допомогою внутрішнього редактора mcedit
- F5 - копіювати
- F6 - перейменувати або перемістити
- F7 - створити директорію
- F8 - видалити
- F9 - випадаючі меню панелі доступу зверху
- F10 - досить. Закриває tc, а також mcedit і будь-які небажані відкриті меню.

Якщо у вас немає F-клавіш, використовуйте Esc + порядковий номер (1-0) замість них.

Нагадуємо, що для можливості збереження редагованих файлів часто необхідно запускати tc через sudo.

Кілька загальних комбінацій:

- Ctrl R - оновити або повторно сканувати директорію перегляду
- Alt-Shift ? - знайти файл
- Ctrl-x d - порівняти директорії (відпустити ctrl-x перед натисканням d)
- Ctrl \ - відкрити «гарячий» список директорії
- Ctrl-x c (o,s,l) - атрибути, chown, symlink, посилання
- Alt c - швидко cd (відкриває коробку, швидше набирати, ніж через arrowing, якщо ви знаєте повний шлях)
- (Shift) + - вибор групи. Можна вводити символи, щоб виділити певний тип або ім'я файлу.
- (Shift) \* - зворотний відбір. Зміни виділяються для всіх невибраних файлів. Зручність у використанні цих двох в комбінацій, якщо ви намагаєтеся швидко обрати все, окрім певної групи.
- \ - Зняти позначку з групи. Навпаки +.

У Midnight Commander є вбудований ftp-клієнт для зручної роботи з FTP-серверами. Для цього використовується команда "FTP-з'єднання", доступна в меню лівої і правої панелей.

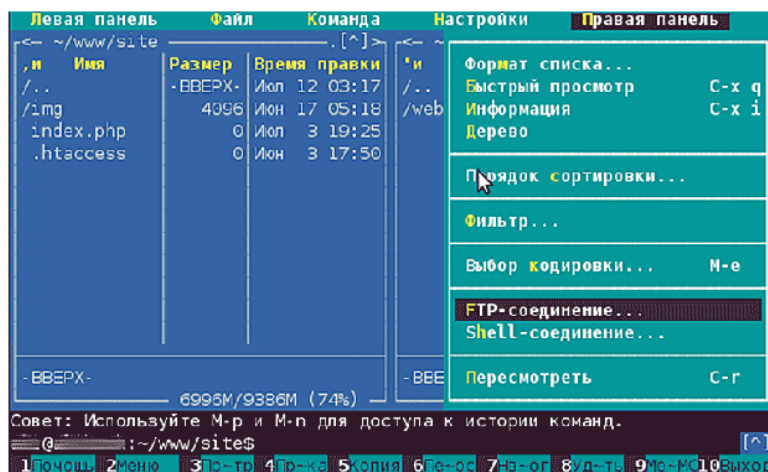


Рис.3.13 Клієнт для роботи з FTP

Викликавши це меню, потрібно ввести лише таку команду:

```
ftpuser:password@ip_adress
```

Можна навіть не вводити пароль - `mc` сам запросить його.

Також можливий схожий варіант, якщо вводити дані в командний рядок при відкритому `mc`:

```
cd /#ftp:ftpuser:password@ip_adress
```

## WinSCP

WinSCP<sup>35</sup> - це графічний клієнт SFTP для Windows з відкритим початковим кодом. Він також успадкував підтримку протоколу SCP. Призначений для захищеного копіювання файлів між комп'ютером і серверами, що підтримують ці протоколи.

Серед основних можливостей програми:

- графічний багатомовний інтерфейс
- інтеграція з ОС Windows (drag&drop, підтримка схем URL, ярлики)
- підтримка SFTP і SCP протоколів по SSH-1 та SSH-2, а також протоколу FTP
- автоматизація за допомогою сценаріїв та командного рядка
- синхронізація директорій за декількома автоматичними та напіваавтоматичними функціями
- вбудований текстовий редактор
- підтримка різних методів аутентифікації SSH протоколу: паролем, відкритим ключем, Kerberos (GSS) та інтерактивна клавіатура

<sup>35</sup> <http://winscp.net/eng/download.php>

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

- інтеграція з Pageant (PuTTY Agent) для підтримки авторизації відкритим ключем
- два інтерфейси користувача: як у провіднику Windows Explorer та як у Norton Commander (схожа на Midnight Commander або Total Commander)
- можливість збереження налаштувань з'єднань
- можливість використання файлу конфігурації замість реєстру, що зручно при запуску з переносних носіїв

WinSCP виконує всі основні операції з файлами, такі як завантаження та вивантаження файлів. Він також дозволяє перейменовувати файли та директорії, створювати директорії, змінювати властивості файлів та директорій, а також створювати символічні посилання та ярлики.

За допомогою WinSCP можна з'єднатися з сервером SSH через протокол SFTP чи SCP, як правило, з машинами під ОС Linux.

SFTP за замовчуванням входить до протоколу SSH-2. SCP, також за замовчуванням, входить до протоколу SSH-1. Обидва протоколи можуть працювати з обома версіями SSH. WinSCP підтримує як SSH-1, так і SSH-2. Підтримується тільки базовий FTP.

Програма пропонує користувачу два інтерфейси, кожен з яких може бути налаштований за допомогою безлічі параметрів. Один з двох інтерфейсів програми також дозволяє керувати файлами на локальному комп'ютері користувача.

Як тільки ви починаєте використовувати WinSCP, можливо, захочете використовувати інтерфейс у стилі Провідника Windows Explorer (рис.3.14), тому що він добре відомий усім користувачам ОС Windows.

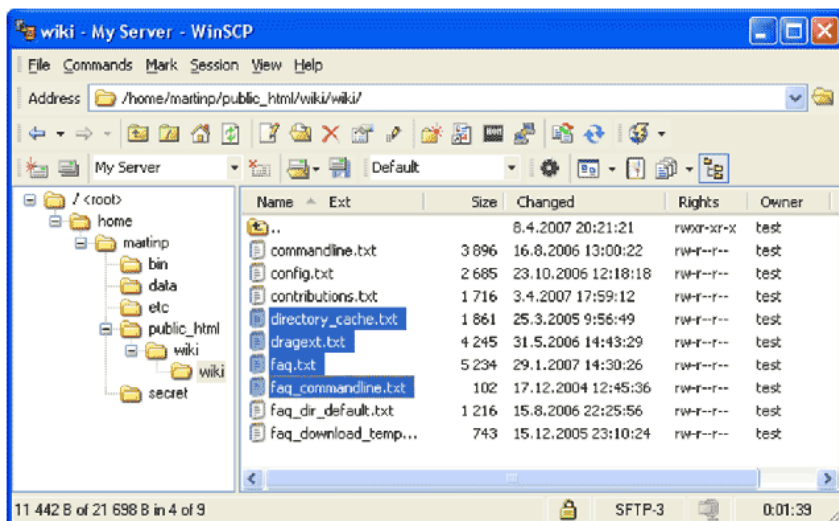


Рис.3.14

Однак, якщо ви звикли працювати в програмах типу Norton Commander, чи таких як цілий ряд сучасних файлових менеджерів (Total Commander, FAR, Altap Salamander), вибирайте інтерфейс як у Norton Commander (рис.3.15), тому що він більш зручний для керування з клавіатури та його можна використовувати взагалі без допомоги “миші”, а це дозволяє виконувати основні операції швидше.

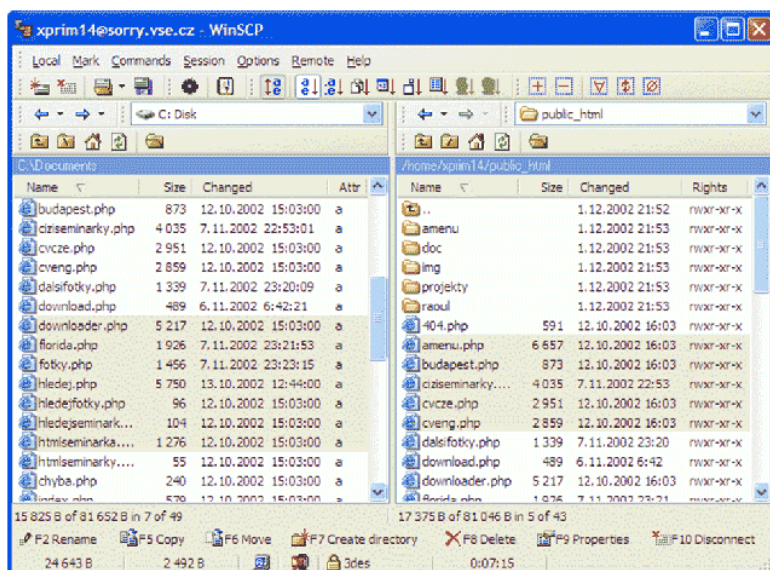


Рис. 3.15 Інтерфейс типу Norton Commander

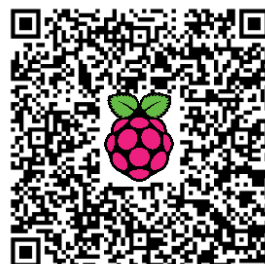
Інтерфейс можна змінити і пізніше, а не лише при встановленні.

## Доступ до робочого столу RPi з комп'ютера Windows



Віддалений доступ до Raspberry Pi зручний тим, що не треба буде підключати до нього клавіатуру, мишу і монітор, а доступ до робочого столу дозволяє запускати такі програми як, наприклад, браузер<sup>36</sup>.

Раніше ми розглянули, як організувати доступ до Raspberry Pi через протокол SSH, щоб безпечно контролювати його з іншого ПК. В даному розділі налаштуємо доступ до віддаленого робочого столу Вашого Pi з іншого комп'ютера, використовуючи різні протоколи.



<sup>36</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1762-two-simple-ways-to-access-your-desktop-computer-raspberry-pi-with-windows>

#### Протокол RDP - Remote Desktop Protocol



Для початку нам необхідно:

1. Запустити Raspberry Pi під управлінням Raspbian "wheezy", але немає сумніву, що описаний спосіб також працюватиме і з іншими версіями Linux.
2. Raspberry Pi повинен бути підключений до локальної мережі і мати доступ в Інтернет.
3. На іншому ПК, який використовуватимемо для підключення, запущена Windows і він підключений до тієї ж локальної мережі.

Тепер ми повинні спочатку налаштувати Raspberry Pi, встановивши деякі програми:

1. Запускаємо вікно терміналу на Pi, або використовуємо доступ до нього з іншого ПК, наприклад, через розглянутий раніше SSH.
2. Встановлюємо на Pi X-сервер для протоколу віддаленого доступу (xrdp):

```
sudo apt-get install xrdp
```

*Xrdp є основним сервером, який приймає з'єднання від RDP-клієнтів. RDP має 3 рівні безпеки між сервером RDP і RDP-клієнтом: низький, середній і високий. Низький – довжина ключа 40 біт, дані шифруються лише від клієнта на сервер, середній - 40-бітне шифрування в обох напрямках і високий - 128-бітне шифрування в обох напрямках. Перш, ніж клієнт підключається, відбувається обмін випадковими ключами RSA між клієнтом і сервером.*

3. Якщо буде запитаний пароль, то вводимо його (за замовчуванням "raspberrypi")
4. Вводимо "Y" і натискаємо Enter.
5. Запускається встановлення на ваш Pi програми забезпечення xrdp, яке збираємося використовувати для віддаленого підключення до робочого столу. Чекаємо його завершення.
6. Перезавантажуємо Pi, збираючись перевірити, чи буде xrdp запускатися автоматично.
7. Коли ваш Pi завантажуватиметься, шукаємо командний рядок:

[ok] Starting Remote Desktop Protocol server: xrdp sesman

Це показує, що xrdp встановлений і автоматично запустився при запуску вашого Pi.

8. На останньому кроці необхідно записати IP-адресу вашого Pi, яка також повинна бути відображена на екрані запуску. Наприклад,

(рис.3.16), 192.168.1.9. Це адреса вашого Pi в мережі і ми будемо її використовувати для підключення до Pi з другого ПК.

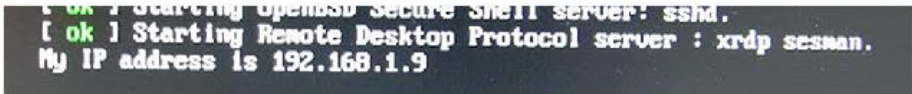


Рис.3.16 Відображення IP-адреси Raspberry Pi

Тепер налаштовуємо інший ПК:

1. Запускаємо віддалений робочий стіл з меню Пуск > Усі програми > Стандартні > Підключення до віддаленого робочого столу.
2. Вводимо IP-адресу для вашого Pi, яку записали вище (рис.3.17).



Рис.3.17 Налаштування віддаленого з'єднання

3. Натискаємо Connect (Підключити) (на даному етапі можете отримати попередження системи безпеки, але просто натисніть ОК, щоб продовжити роботу, адже це ваш Pi в мережі).
4. Залишаємо Module (Модуль) sesman-Xvnc за замовчування і вводимо ім'я користувача та пароль для облікового запису Pi (рис.3.18) (за замовчуванням **pi** і **raspberrypi**, якщо ви їх не змінили).
5. Натискаємо ОК і через кілька секунд ви повинні побачити на своєму моніторі робочий стіл Raspberry Pi!
6. Коли закінчите роботу - просто вийдіть з системи з робочого столу Pi.

Використання xrdp дозволяє отримати наступні переваги:

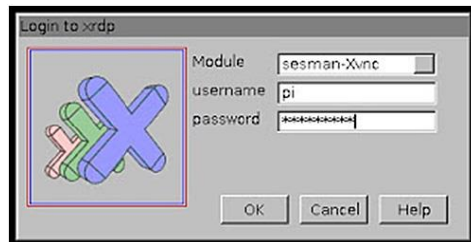


Рис.3.18 Під'єднання до Raspberry Pi

- Відсутність витрат на ліцензії - переважна частина ПЗ для Linux є безкоштовною і не потребує витрат на придбання ліцензій на сервер, ліцензій на клієнтський доступ, ліцензій на термінальний доступ.
- Використання протоколу RDP дозволяє зробити сервер доступним з будь-якої системи, що дозволяє використовувати недорогі тонкі клієнти.
- Користувачам при доступі до сервера з домашньої або гостьової мережі немає необхідності встановлювати будь-яке ПЗ (тому що воно вже включене у всі системи Windows і більшість Linux-систем).
- На відміну від термінального сервера Microsoft, зв'язка xrdp + VNC, яка тут використовується, дозволяє отримати копію зображення на екрані один до одного, в тому числі неспотворені шрифти (True Type), відеоролики (у рамках пропускну здатності мережі) і мультимедійні презентації.

#### Протокол SSH - Secure Shell (з PuTTY)



Для більш захищеного віддаленого доступу до робочого столу Raspberry Pi користувачам Windows необхідні дві частини програмного забезпечення: програма безпечної оболонки (SSH) для встановлення віддаленого з'єднання і X-сервер для обробки локального дисплея.

Для початку нам необхідно встановити:

- Putty<sup>37</sup>
- Xming<sup>38</sup> для X-сервера

#### Налаштовуємо Putty:

1. Додаємо ім'я хоста або IP-адресу.
2. Вмикаємо протокол SSH.
3. Вводимо ім'я сесії у рядку збереження сесій.
4. Натискаємо "Save" (Зберегти) (рис.3.19).
5. Розгортаємо вкладку "SSH" у списку "Category" (Категорія).
6. Вибираємо "X11" зі списку "SSH" (рис.3.20).
7. Встановлюємо прапорець "Enable X11 Forwarding" (Включити

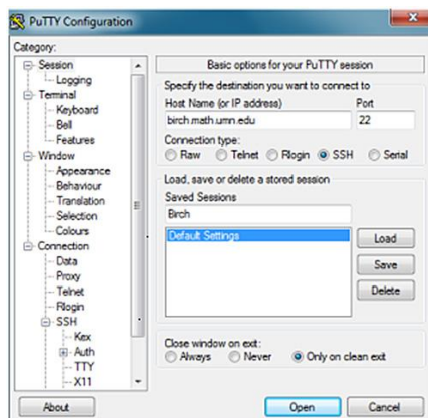


Рис.3.19 Налаштування сесії

<sup>37</sup> <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

<sup>38</sup> <http://sourceforge.net/projects/xming/files/latest/download>

перенаправлення X11).

8. Вибираємо "Session" (Сесії) у списку "Category" (Категорія).

9. Натискаємо "Save" (Зберегти).

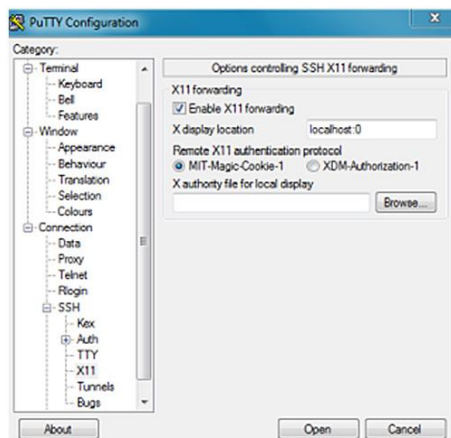


Рис.3.20 Налаштування перенаправлення

### Налаштовуємо Xming:

Просто запускаємо Усі програми > Xming Xming, і він повинен працювати, якщо у вас налаштований PuTTY.

Під'єднуємося з Windows до Raspberry Pi:

1. Запускаємо Xming.
2. Запускаємо PuTTY.
3. Двічі клацаємо на обраній збереженій сесії (рис.3.21).
4. У відповідь на запит вводимо ім'я користувача і пароль.
5. Тепер повинні отримати можливість запускати X-програми з Raspberry Pi на локальному робочому столі.

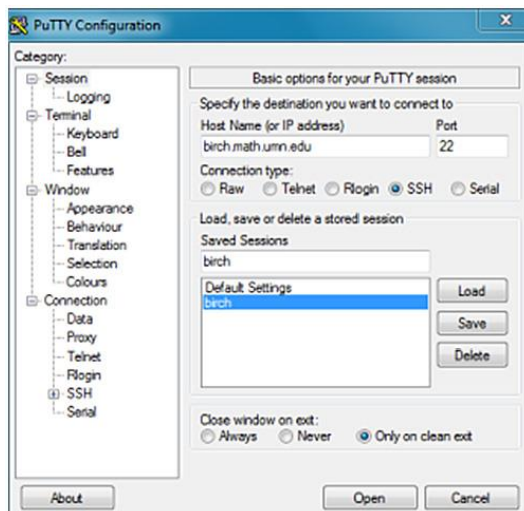


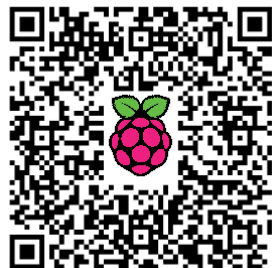
Рис.3.21 Запуск збереженої сесії

І хай ваш Pi слухняно виконує лише ваші команди! :)

## Налаштування VNC на Raspberry Pi



Хоча Raspberry Pi можна підключити до телевізора або монітора через HDMI або DVI, є моменти, коли бажано працювати з Raspberry Pi "без голови" (без монітора).



При «безголовій» установці, ви можете підключитися до свого Raspberry через SSH, але якщо хочете мати повний робочий стіл, то для підключення треба буде використати протокол віддаленого робочого столу. Найпростіше налаштувати на Raspberry Pi VNC. VNC (Virtual Network Computing) являє собою графічний протокол робочого столу, який дозволяє отримати доступ до повного робочого столу Raspberry Pi з іншого комп'ютера<sup>39</sup>. Зазвичай, ви зможете запустити клієнта VNC на ПК під управлінням Windows, OS X або Linux і отримати доступ до робочого столу Pi через мережу.

Першим кроком є установка сервера VNC на Raspberry Pi. Щоб зробити це, виконайте наступну команду для встановлення TightVNC - легкої і з відкритим вихідним кодом реалізації системи VNC:

```
sudo apt-get install tightvncserver
```

Тепер, коли сервер встановлений, він повинен бути запущений для генерації віддаленого сеансу. Сервер не запускається за замовчуванням, однак, може бути запущений вручну таким чином (рис.3.22):

```
pi@raspberrypi ~ $ vncserver -geometry 1024x728 -depth 24
You will require a password to access your desktops.
Password: 
Verify: 
Would you like to enter a view-only password (y/n)? n
New 'X' desktop is raspberrypi:1
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi.log
pi@raspberrypi ~ $
```

Рис.3.22 Запуск сервера VNC вручну

<sup>39</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1789-setting-up-vnc-on-raspberrypi>

```
vncserver -geometry 1024x728 -depth 24
```

Де **-geometry** встановлює дозвіл віддаленого робочого столу, в даному випадку 1024 x 768, а **-depth** задає глибину кольору, в даному випадку 24 біт. Перший раз, коли запускаєте сервер VNC, буде запропоновано ввести пароль віддаленого доступу. Це пароль, який необхідно вводити, коли будете намагатися віддалено підключитися до робочого столу. На цей пароль не варто покладатися, як на єдиний засіб, щоб зупинити загальний доступ до віддаленого робочого столу, але для приватної домашньої або офісної мережі він повинен бути достатнім для обмеження доступу до інтерфейсу користувача Pi. Ви можете змінити пароль в будь-який час за допомогою інструменту **vncpasswd**.

При виході з сервера, **vncserver** буде повідомляти про кількість віртуальних екранів, які він створив. Під X (основної системи для багатьох віконних середовищ Linux), можете мати кілька дисплеїв з фізичного робочого столу :0. Цей **vncserver**, швидше за все, створить дисплей :1, до якого можна отримати доступ від клієнта VNC на порту 5901. Порт 5902 використовується для дистанційного віртуального робочого столу :2 і т. д. Ви можете змусити сервер VNC створити сеанс на певний дисплей, вказавши номер на місці першого параметра:

```
vncserver :1 -geometry 1024x728 -depth 24
```

Ви можете знищити віддалений робочий стіл у будь-який час за допомогою:

```
vncserver -kill :1
```

Де :1 це номер робочого столу.

## Клієнти VNC

Є багато клієнтів VNC (яких часто називають VNC viewers) для Windows, OS X і Linux. Ви можете отримати переглядача для Windows з сайту TightVNC<sup>40</sup>, а для Linux можете встановити його з репозиторію свого дистрибутива. Для використання Debian:

```
sudo apt-get install vncviewer
```

Для підключення до Raspberry Pi, запустіть переглядач VNC:

```
vncviewer 192.168.1.4:1
```

Де 192.168.1.4 є адресою Raspberry Pi і :1 - номер робочого столу.

---

<sup>40</sup> <http://www.tightvnc.com/>

#### Запуск VNC-сервера при завантаженні

Можна отримати сервер VNC, який буде стартувати при завантаженні Raspberry Pi, створивши спеціальний сценарій ініціалізації. Створіть файл з назвою `vncserver` у своєму домашньому каталозі з наступним кодом оболонки:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          VNC
# Required-Start:    $local_fs
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start or stop the VNC server
### END INIT INFO

PATH=/sbin:/usr/sbin:/bin:/usr/bin
eval cd ~pi
case "$1" in
    start)
        su pi -c "/usr/bin/vncserver :1 -geometry 1024x728 -depth
24"
        echo "Started VNC server."
        ;;
    stop)
        su pi -c "/usr/bin/vncserver -kill :1"
        echo "Stopped VNC server."
        ;;
    *)
        echo "Usage: vncserver [start|stop]" >&2
        exit 3
        ;;
esac

:
```

Тепер виконайте наступні команди, щоб змінити власника файлу на `root`, скопіюйте його в каталог `init.d` і встановіть сценарій:

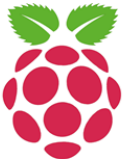
```
sudo mv vncserver /etc/init.d/
sudo chown root:root /etc/init.d/vncserver
sudo chmod 755 /etc/init.d/vncserver
sudo update-rc.d vncserver defaults
```

Перезавантажте комп'ютер і перевірте, що сервер VNC запустився автоматично. Є кілька речей, на які ви повинні звернути увагу в цьому сценарії. Спочатку він передбачає, що ви використовуєте користувача за

замовчуванням `pi` а по-друге, він буде працювати тільки, якщо ви вже встановили пароль, використовуючи `vncpasswd` або хоча б один раз запустити сервер VNC вручну.

## Висновки

Тепер, коли встановлено сервер VNC, ви можете отримати доступ до повного робочого столу через мережу за допомогою клієнта VNC. Навіть якщо ваш Raspberry Pi підключений до телевізора або монітора, це відмінний спосіб, щоб включити віддалене адміністрування пристрою.



*Якщо вам коли-небудь знадобиться зупинити сервер VNC, то можете скористатися наступною командою `sudo service vncserver stop`. Сервер може бути запущений знову шляхом заміни параметра `stop` на параметр `start` в наведеній вище команді.*

## Управління Raspberry Pi з iPad



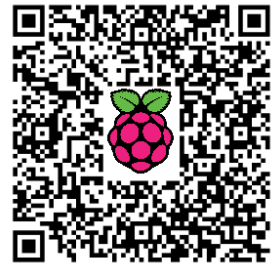
Це чудово, що Raspberry Pi може бути захований в кутку і працювати без монітора, в той час як iPad захоплює контроль<sup>41</sup> і ви можете використовувати його клавіатуру і монітор.

Тепер ви зможете увійти в RPi з будь-якої точки світу, звідки у вас є доступ в Інтернет – і скористатися тим, щоб запустити `python` і почати вчитися програмувати, використовуючи мобільні зручності свого iPad.

Перш, ніж ви зможете віддалено підключатися до свого Raspberry Pi, необхідно підключили його до Інтернету. Можете зробити це, під'єднавши до нього мережевий кабель або бездротовий USB-адаптер.

Спочатку розглянемо додатки, які не вимагають зусиль при налаштуванні та добре виконують свою роботу:

- Fing<sup>42</sup> (безкоштовно)
- WebSSH<sup>43</sup> (безкоштовно)
- VNC Viewer<sup>44</sup> (£6.99)



<sup>41</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1787-controlling-the-raspberry-pi-from-an-ipad>

<sup>42</sup> <https://itunes.apple.com/gb/app/fing-network-scanner/id430921107?mt=8&affId=1998745&ign-mpt=uo%3D4>

<sup>43</sup> <https://itunes.apple.com/gb/app/webssh/id497714887?mt=8&affId=1998745&ign-mpt=uo%3D4>

*Примітка:* VNC Viewer досить дорогий, тому можете погратися з деякими безкоштовними версіями, перш ніж прийняти рішення розлучитися з часткою своїх грошей.

## Використання SSH для підключення до Raspberry Pi

### Дозвіл SSH

SSH (Secure Shell) забезпечує доступ до інтерфейсу командного рядка в Pi. Перш, ніж ви зможете використовувати його, необхідно включити SSH в файлі `config.txt` RPi, як ми це робили раніше, використовуючи команду `sudo raspi-config`.

Для підключення до свого Raspberry Pi потрібно знати IP-адресу, яку він використовує. Ви можете її знайти, скориставшись командою `ifconfig`, але зручно встановити на iPhone або iPad безкоштовний додаток Fing, який дозволить сканувати локальну мережу, щоб побачити всі з'єднання.

### Підключення через SSH

Завантажте WebSSH (або аналогічний додаток) і додайте нове з'єднання SSH.

Хостом є IP-адреса, що використовується вашим Raspberry Pi та ідентифікована за допомогою Fing. Користувач і пароль ті, які зазначені в `raspi-config`. За замовчуванням використовується `pi/raspberrypi`.

Якщо ви встановили підключення, то тепер будете винагороджені доступом до командного рядка.

Єдине, що варто мати на увазі, що ваш Raspberry Pi буде використовувати динамічну IP-адресу, тому щоразу при його перезавантаженні, швидше за все, буде вибрана інша адреса. А це означає, що ви будете змушені змінити параметри SSH-з'єднання у додатку.

Ви можете обійти це, змусивши ваш RPi використовувати статичну IP-адресу. В другому розділі ми розглядали, як це зробити.

## Використання VNC для підключення Raspberry Pi з iPad

Протокол SSH тільки дозволив підключитися до командного рядка вашого Raspberry Pi, але якщо хочете мати графічний робочий стіл, то повинні будете використовувати VNC (Virtual Network Computing).

---

<sup>44</sup> <https://itunes.apple.com/gb/app/vnc-viewer/id352019548?mt=8&affid=1998745&ign-mpt=uo%3D4>

#### **Встановлення сервера VNC на Raspberry Pi**

Для використання VNC нам треба встановити VNC server на RPi. Коротко повторимо інструкції для встановлення з терміналу:  
`sudo apt-get install tightvncserver`

Дотримуйтесь інструкцій і введіть ім'я користувача та пароль відповідно до них, а потім запустіть сервер, ввівши:

`tightvncserver`

Сервер VNC автоматично не запускатиметься після перезавантаження (на відміну від SSH, який завжди буде включений). Це може бути перешкодою, якщо ви регулярно перезавантажуєтесь і, якщо це так, то можете слідувати наведеному раніше розгляду налаштування VNC, щоб форсувати запуск сервера при старті.

#### **Підключення через VNC**

Завантажте VNC Viewer на свій iPad, - автор використовував VNC Viewer, але доступні дешевші і безкоштовні версії.

Додайте нове підключення. Адресою буде статична IP-адреса, яку ви встановили вище, але варто також додати номер порту. Так у автора введено 192.168.0.11:1 (вважаю, що порт 1 за замовчуванням повинен працювати також і у вас).

Пароль той, який ви створили при встановленні tightVNCserver.

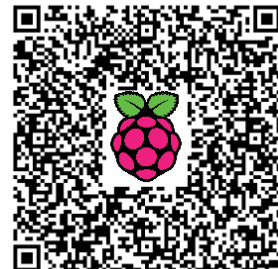
Підключення є і ви повинні бути винагороджені красивою малиною на екрані iPad.

#### **Android-клієнти для віддаленого доступу зі смартфона до RPi**



Можливість керувати комп'ютером віддалено зі свого смартфона особливо таким, як Raspberry Pi, вже не лише мрія багатьох, але і реальність.

Це стало можливим завдяки використанню клієнтів і серверів, перш за все, SSH, VNC і Telnet. В розділі коротко розглянемо лише найпопулярніших клієнтів<sup>45</sup>, доступних для Android, а далі більш детально зупинимось на побудові взаємодії клієнтів і серверів.



<sup>45</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1764-clients-for-android-to-get-remote-access-from-your-phone-to-the-raspberry-pi-computers-and-other>

#### ConnectBot<sup>46</sup>



ConnectBot є потужним клієнтом Secure Shell (SSH) з відкритим вихідним кодом. Він може керувати одночасних сесіями SSH, створювати безпечні тунелі і обмінюватися (копіювати/вставляти) з іншими додатками.



ConnectBot є клієнтом захищеної оболонки для платформи Android. Ви можете використовувати цю оболонку на віддаленій машині і передавати файли туди і назад у телефон. Даний клієнт дозволяє підключатися до серверів SSH, які зазвичай працюють на UNIX-серверах, тому не виникає проблем з під'єднанням до RPi.

#### Hacker'sKeyboard<sup>47</sup>



Дана клавіатура особливо корисна, якщо використовуєте ConnectBot для доступу SSH. Вона забезпечує робочі клавіші Tab/Ctrl/Esc, а також клавіші, які мають важливе значення для таких пристроїв, як планшет Xoom або Nexus S, які не



мають трекболу чи D-Pad.

Ця клавіатура має окремі клавіші з цифрами, знаками пунктуації і клавіші зі стрілками в звичайних місцях. Вона заснована на екранній клавіатурі AOSP Gingerbread, тому підтримує мультитач для клавіш-модифікаторів.

Клавіатура також підтримує використання словників (але не розкладки клавіатури), надані мовними пакетами AnySoftKeyboard.

#### VX ConnectBot<sup>48</sup>



Клієнт SSH і Telnet з відкритим вихідним кодом та SCP-передачею файлів створений на основі оригінального ConnectBot.

Серед покращеної функціональності:

- фонові передача файлів (SCP протокол)
- діалог вибору шрифтів (кнопка на екрані і апаратний ключ SYM)
- одна вхідна лінія (кнопка на екрані)
- меню з натисканням і утриманням (перемикання в повноекранний режим, зміна розміру шрифту)
- збереження SSH-ключів у вигляді файлів
- збереження скріншотів консолі (PNG-формат)
- новий відображення клавіш (апаратна клавіатура)
- ssh-агент (від Roberto Tyley)
- перенаправлення X11 (корисно для Android XServer)



<sup>46</sup> <https://play.google.com/store/apps/details?id=org.connectbot>

<sup>47</sup> <https://play.google.com/store/apps/details?id=org.pocketworkstation.pckeyboard>

<sup>48</sup> <https://play.google.com/store/apps/details?id=sk.vx.connectbot>

Клієнт отримав нове зіставлення клавіш:

- кнопка камери (настроюється): за замовчуванням Screen Capture
- ALT + стрілка вгору: Page Up
- ALT + стрілка вниз : Page Down
- ALT + стрілка вліво: Головна
- ALT + стрілка вправо: Кінець
- ALT + Backspace: Insert

VX ConnectBot забезпечує додаткові налаштування для смартфонів з фізичною клавіатурою. На сьогодні існують налаштування для наступних смартфонів:

- з повною апаратною клавіатурою (наприклад, Bluetooth-клавіатура)
- Samsung Captivate Glide (SGH-i927)
- Sony Ericsson Xperia Pro (MK16i) та Mini Pro (SK17i)

#### JuiceSSH<sup>49</sup> - SSH клієнт



Клієнт для Android - все в одному терміналі, включаючи SSH, локальний Shell, підтримку Mosh і Telnet.

Серед особливостей:

- повно кольоровий термінал/SSH клієнт
- спливаюча клавіатура з усіма тими символами, які зазвичай важко знайти
- використання клавіш регулювання об'єму, щоб швидко змінити розмір шрифту
- підтримка зовнішньої клавіатури
- офіційна підтримка Mosh (<http://mosh.mit.edu>)
- підтримка Telnet
- підтримка локального терміналу Android
- відкривання в браузері URL-адреси при натисненні
- збереження/розширення стенограм SSH для Dropbox /Evernote /email і SD-карт
- підтримка UTF-8
- легко організувати з'єднання по групах
- збереження кількох сесій SSH при роботі у фоновому режимі
- легке підключення "через" інші з'єднання SSH
- негайний доступ до часто використовуваних з'єднань
- підтримка паролів і секретного ключа OpenSSH
- підтримка двох факторної аутентифікації
- ідентифікація (користувачі/пароль/ключі) абстрагована від з'єднання
- працює з Ubuntu, RedHat, CentOS, Mint, Gentoo і всіма іншими релізами Linux
- zlib стиснення для поліпшення SSH-сесій



<sup>49</sup> <https://play.google.com/store/apps/details?id=com.sonelli.juicessh>

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

---

Серед особливостей версії Pro:

- інтеграція з Amazon AWS/EC2, синхронізація з'єднання та автоматичне групування серверів
- надійна синхронізація між кількома пристроями
- автоматизоване AES-256 шифрування резервних копій всіх підключень і налаштувань
- бібліотека фрагментів для швидкого доступу до часто використовуваних команд
- безпечне блокування для автоматичного захисту JuiceSSH після певного періоду бездіяльності

#### Server Auditor<sup>50</sup>



Server Auditor є прогресивним клієнтом SSH, який дозволяє Вам керувати UNIX-серверами з Ваших пристроїв Android та iPhone. Інтерфейс користувача реалізований з використанням простого і зрозумілого стилю, максимального використання жестів.



Особливості:

- підтримка декількох з'єднань
- пароль і ключ авторизації
- підтримка XTerm, VT100 і vanilla
- працює в обох орієнтаціях
- зручна функціональність копіювання - вставляння з можливістю миттєвого пошуку в Google і на сайтах ServerFault

Потужний SSH-клієнт з простим для користування інтерфейсом, спеціально розробленим для сенсорних пристроїв і дивовижної ефективності користувача.

#### Android-vnc-viewer<sup>51</sup>



Програма androidVNC для доступу до віддаленого робочого столу пристроїв Android з відкритим вихідним кодом (GPL) дозволяє спостерігати і контролювати робочий стіл комп'ютера з вашого телефону з будь-якої точки світу.



З'єднується з більшістю серверів VNC (Virtual Network Computing), включаючи TightVNC, RealVNC на Win і Linux, x11vnc і Apple Remote Desktop на OS/X.

Імпорт/експорт налаштувань у файлі на SD-карті (можна імпортувати дані з URL, що корисно при установці багатьох пристроїв з такою ж конфігурацією). Більш детально про налаштування та використання VNC ми розглядали в одному з попередніх розділів.

---

<sup>50</sup> <https://serverauditor.com/>

<sup>51</sup> <https://play.google.com/store/apps/details?id=android.androidVNC>

#### TeamViewer<sup>52</sup>



Мобільний та гнучкий: віддалений доступ до комп'ютерів зі свого пристрою Android!



З його допомогою легко підтримувати друзів і рідних, коли вони мають проблем з комп'ютером - незалежно від того, де ви знаходитесь. Ви також можете скористатися

доступом до особистого домашнього комп'ютера для редагування документів або використання конкретного програмного забезпечення, поки знаходитесь в дорозі.

Деякі з можливостей:

- доступ до (Windows, Mac, Linux) можливий також під час телефонної розмови
- передача файлів з мобільного пристрою Android на будь-який комп'ютер і навпаки
- повне управління з клавіатури, включаючи такі спеціальні клавіші, як Ctrl, Alt, Windows®
- відповідає високим стандартам безпеки: 256 біт AES-шифрування сеансу, обмін ключами - 1024 біт RSA
- легко отримати доступ до комп'ютерів за брандмауерами і проксі-серверами
- віддалене перезавантаження комп'ютера
- огляд друзів і комп'ютерів, які в Інтернеті, через вбудований список комп'ютерів

На жаль, сьогодні ще відсутня підтримка процесорів ARM-архітектури і тому TeamViewer, поки що, не підтримується на Raspberry Pi, але хочеться вірити, що це незабаром буде.

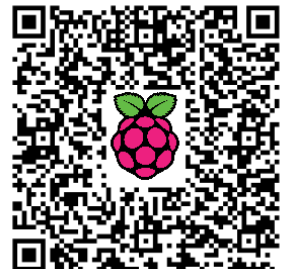
### Як організувати потокову передачу файлів на смартфон



Найкращим в Raspberry Pi є те, що це відмінний навчальний пристрій. Розглянемо, як розвернути на вашому RPi файловий сервер<sup>53</sup> та організувати роздачу файлів,

наприклад, для навчання.

Всупереч поширеній думці, не кожна задача сервера вимагає багато обчислювальної потужності. Наприклад, якщо хочете завантажувати файли і ділитися ними з іншими



<sup>52</sup> <https://play.google.com/store/apps/details?id=com.teamviewer.teamviewer.market.mobile>

<sup>53</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1769-how-to-streaming-files-from-a-smartphone-on-raspberry-pi>

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

---

користувачами мережі незалежно від їх операційної системи або пристрою, то вам не потрібна машина з багатоядерним процесором. Справді, використання старого комп'ютера Linux є популярним способом порятунку апаратного забезпечення, яке морально застаріло.

Єдиний недолік цього підходу полягає в тому, що старі комп'ютери споживають багато електроенергії. Більшість сучасних маршрутизаторів мають порт USB і можуть ділитися через включений в цей порт пристрій, з усіма комп'ютерами, підключеними до даного маршрутизатора. Але якщо у вас старий маршрутизатор і хочете мати більше, ніж просто обмінюватися файлами?

З RPi ви можете робити все це і багато іншого без витрат, пов'язаних з традиційним комп'ютером, і без обмежень таких спеціальних пристроїв, як маршрутизатори.

Ми будемо використовувати RPi версії B з 512 Мб оперативної пам'яті, але все повинно нормально працювати і на попередній версії. На RPi використаємо останню версію Raspbian. Найкраще в нових релізах те, що вони використовують всю збільшену пам'ять без будь-яких хитрощів. Вони також встановлюються з включений SSH, який дозволяє користувачам отримувати доступ і налаштовувати RPi навіть не підключаючи його до монітора.

Завантажте останню версію дистрибутива Raspbian і прошийте його на SD-карту, як про це ми розповідали в розділі про початок роботи з RPi. Хоча ви можете використовувати бездротовий адаптер USB для підключення RPi до Інтернету, але краще використати порт Ethernet і приєднати кабель до маршрутизатора.

Після підключення і включення живлення RPi, відкрийте головну сторінку адміністратора маршрутизатора. Цілком ймовірно, ваш маршрутизатор використовує DHCP для роздачі IP-адрес підключеним пристроям.

Розкрийте інтерфейс адміністратора, подивіться список всіх підключених пристроїв і запишіть IP-адресу, призначену маршрутизатором для RPi, або дізнайтесь про IP-адресу через сканування мережі. Пізніше вам доведеться призначити йому статичну IP-адресу, щоб завжди забезпечити доступність до RPi за тією ж адресою.

Давайте припустимо, що динамічна IP-адреса, призначена RPi, 192.168.3.100. За допомогою цієї інформації ми маємо все для підключення до RPi і налаштування Raspbian через SSH.

Для підключення з віддаленої машини Linux запустіть термінал і введіть наступну команду:

```
$ sudo ssh pi@192.168.3.100
```

Після згоди додати цю адресу до списку довірених вузлів, будуть запитані облікові дані для входу. Нагадаємо, що паролем за замовчуванням для користувача **pi** є **raspberrypi**. З цього моменту, команди в статті будуть однаковими, незалежно від того, підключені ви

до RPi з Linux чи з Windows. Це тому, що команди виконуються на дистрибутиві Rasbian Linux на RPi.

Для налаштування конфігурації введіть команду:

```
$ sudo raspi-config
```

Виберіть опцію `memory_split`, яка дозволяє розділити RAM на RPi між GPU і CPU. Так як ми будемо отримувати доступ до RPi тільки через віддалене з'єднання, то переконайтеся, що призначити «голий» мінімум пам'яті в GPU, всього 16MB.

Далі, ви повинні додати інших користувачів в RPi. Це необхідно, щоб надалі мати змогу обмежувати доступ до певних каталогів на підключеному USB-пристрої для певних користувачів, а також групи користувачів, і при цьому мати простір для публічного доступу на пристрої збереження.

```
$ sudo adduser bodhi
```

Команда додасть користувача `bodhi` і створить відповідні каталоги. Вона запросить пароль користувача та інші деталі. Потім додайте цього користувача в групу користувачів:

```
$ sudo usermod -a -G users bodhi
```

## Встановлення та налаштування Samba

Пора зробити RPi доступним для користувачів в Інтернеті. Для цього використаємо Samba, яка дозволяє нам обмінюватися файлами через протокол Common Internet File System (CIFS). Щоб встановити Samba на RPi введіть:

```
$ sudo apt-get install samba samba-common-bin
```

Коли це буде зроблено, ви повинні додати користувачів в Samba. Наприклад, для користувача за замовчуванням `pi` команда має вигляд:

```
$ sudo smbpasswd -a pi
```

Вам буде запропоновано ввести пароль. Як правило, безпечно використовувати той же пароль, що і пароль облікового запису користувача. Повторіть цей крок для кожного користувача системи.

Samba керується через файл конфігурації, який потрібно спочатку налаштувати. Завжди чудовою ідеєю є створення резервної копії файлів конфігурації перед внесенням змін:

```
$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.old
```

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

---

Тепер відредагуйте файл конфігурації за допомогою **nano**:

```
$ sudo nano /etc/samba/smb.conf
```

Знайдіть у файлі розділ відзначений Authentication (Автентичність). У даному розділі видаліть символ решітки на початку лінії, щоб **#security = user** виглядало як **security = user**

Це гарантує, що Samba дозволить тільки тим користувачам увійти в систему, у кого є домашні каталоги на RPi. Щоб дозволити користувачам доступ до своїх власних домашніх каталогів, знайдіть розділ **[homes]** і переконайтеся, що **browseable = yes** і **read only = no**.

Для виходу з **nano**, натисніть Ctrl+X, щоб зберегти файл, а потім клавішу введення, коли буде показане ім'я файлу.

Щоб включити зміни перезапустимо Samba:

```
$ sudo service samba restart
```

Пора підключити пристрій USB до RPi. Хоча RPi має кілька портів USB, але рекомендуємо включати додаткові пристрої USB за допомогою концентратора з додатковим живленням, особливо при підключенні великих USB-пристроїв зберігання, які споживають багато енергії.

Крім того, зверніть увагу на те, як диск відформатований. Багато флеш- накопичувачів USB за замовчуванням відформатовані в FAT32. Це може бути кращим форматом з точки зору підтримуваних операційних систем, але це, безумовно, гірше для обміну файлами через мережу.

Також є NTFS, яка використовується багатьма великими зовнішніми накопичувачами з інтерфейсом USB. Цей формат не для вас, якщо хочете забезпечити мультимедійний потік від віддаленого диска, а найкраще це робити за допомогою EXT4. Використовуйте тільки EXT4, якщо привід буде використовуватися через мережу або на Linux.

Після підключення приводу, дізнайтеся про його місце розташування з **sudo fdisk -l**. Будуть перераховані пристрої, підключені до RPi і розділи на них. Проскануйте вихід і знайдіть диск, розмір якого відповідає USB диску, який Ви підключили. Пристрій, ймовірно, буде в **sda** і розділ, на якому ми хочемо змонтувати, **sda1**.

Створюємо точку монтування та монтуємо пристрій:

```
$ sudo mkdir /mnt/usb  
$ sudo mount /dev/sda1 /mnt/usb
```

Привід USB залишиться встановлений до перезавантаження RPi. Щоб уникнути необхідності перемонтовувати пристрій, спочатку знайдіть його UUID, як ми це робили раніше:

```
$ sudo blkid  
/dev/sda1: LABEL="ntfs" UUID="3B5C053D35CAD865" TYPE="ntfs"
```

Тепер додайте його в список пристроїв, які встановлюються при завантаженні:

```
$ sudo nano /etc/fstab
UUID=3B5C053D35CAD865 /mnt/usb ntfs defaults 0 0
```

Samba призначена для обміну файлами і папками, які визначені в конфігураційному файлі. Для загального доступу до папок на пристрої USB давайте припустимо, що маєте чимало різних папок:

```
$ ls /mnt/usb documents downloads music videos
```

Щоб «розшарити» папку завантажень, відкрийте файл `/etc/samba/smb.conf` за допомогою `папо`, перейдіть вниз і введіть:

```
[Downloads]
comment = Place all your downloads here
Path = /mnt/usb/downloads
browseable = yes
writable = yes
read only = no
valid users = @users
```

Даний блок тексту «розшарить» директорію `/mnt/usb/downloads` для усіх користувачів в групі користувачів. Пізніше ми змонтуємо цю директорію на комп'ютерах Linux і Windows. Користувачі зможуть мати можливість з будь-якого комп'ютера в мережі налаштувати свої менеджери завантажень для збереження файлів безпосередньо в цю папку на пристрої USB, приєднаного до RPi.

Також можете обмежити доступ до деяких папок тільки певним користувачам, наприклад, тільки користувачам `pi` і `bodhi`:

```
[Documents]
comment = Important eyes-only PDF files
path = /mnt/usb/documents
browseable = no
writable = yes
read only = no
valid users = pi, bodhi
```

## Встановлення сервера торрентів

Торренти є переважним середовищем для обміну контентом з відкритим вихідним кодом.

Ми обираємо клієнтом Transmission, який має простий для використання веб-інтерфейс. Встановимо Transmission на Raspbian і

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

---

звертатимемося до нього з будь-якого браузеру на будь-якому комп'ютері, щоб додавати, моніторити та керувати торрентами.

Для встановлення входимо через SSH на RPi і вводимо:

```
$ sudo apt-get install transmission-daemon
```

Перш, ніж ви зможете використовувати Transmission для завантаження торрентів, необхідно його налаштувати. Перед внесенням будь-яких змін у файл конфігурації Transmission переконайтеся, що демон не запущений:

```
$ sudo service transmission-daemon stop
```

Крім того, додайте користувача передачі (`debian-transmission`), який створився автоматично при встановленні демона, до нашої групи користувачів:

```
$ sudo usermod -a -G users debian-transmission
```

Тепер створимо публічний ресурс на пристрої USB, на який будемо завантажувати торренти. По-перше, створюємо загальнодоступну частку на USB і призначаємо користувача передачі її власником:

```
$ sudo mkdir /mnt/usb/public  
$ sudo chown debian-transmission /mnt/usb/public
```

Тепер додайте наведену нижче частку до конфігураційного файлу Samba:

```
[Public]  
comment= Public share for torrents  
browseable = yes  
path = /mnt/usb/public  
public = yes  
writeable = yes  
guest ok = yes
```

Перезапустіть Samba, щоб зробити частку доступною кожному. Прийшов час налаштувати демон Transmission. Його параметри визначені в `/etc/transmission-daemon/settings.json`

Відкрийте файл в nano і спочатку змініть параметр `"rpc-whitelist-enabled": true` на `"rpc-whitelist-enabled": false`, щоб дозволити користувачам підключатися з усіх комп'ютерів.

Потім вкажіть каталог для завантаження:

```
"download-dir": "/mnt/usb/public/downloads/Complete"
```

Ви також можете відокремити неповні файли, зберігаючи їх в іншу папку. По-перше, включіть опцію шляхом зміни параметру `"incomplete-dir-enabled": false` на `"incomplete-dir-enabled": true` і вкажіть каталог, який буде домом для незавершених завантажень з `"incomplete-dir":`

```
"/mnt/usb/public/downloads/Incomplete"
```

## Аутентифікація користувача

Так як ми дали право власності на каталог `/mnt/usb/public` користувачу передачі, то він буде автоматично створювати нові каталоги. У той час, як скачувані торренти є відкритими, ви можете переконаватися, що не кожен може стояти в черзі торрентів для завантаження.

Одним із способів зробити це – запросити у користувачів аутентифікацію, перш ніж дозволити їм доступ до Transmission. Для цього спочатку треба змінити параметр `"rpc-authentication-required": false` на `"rpc-authentication-required": true`.

Потім визначити пароль в параметрі `"rpc-password": "<password>"`, який буде автоматично шифрувати Transmission. Тепер збережіть файл і запустіть демон Transmission з `sudo service transmission-daemon start`.

За замовчуванням Transmission використовує порт 9091. У нашому прикладі, повний URL для веб-інтерфейсу користувача Transmission буде 192.168.3.100:9091. Запустіть браузер і введіть вказану адресу.

Якщо ви включили параметр пароля, то буде запропоновано ввести облікові дані для входу. Іменем користувача є `transmission`, а пароль, який ви вказали у файлі конфігурації.

Перед тим, як завантажити торрент, вам необхідно знати розташування файла `.torrent`. Перейдіть до веб-інтерфейсу Transmission і натисніть кнопку `Open Torrent` (Відкрити торрент). Вставте URL в спливаючому вікні та натисніть кнопку `Upload` (Завантажити), щоб почати завантаження.

В інтерфейсі легко орієнтуватися. За замовчуванням, він показує всі додані торренти, але можете використовувати випадаюче меню, щоб подивитися торренти відповідно до їх стану завантаження або трекерів. При натисканні правою кнопкою миші на перерахованому торренті в списку, буде відображене контекстне меню.

Після того, як торрент завантажений, згідно з нашою конфігурацією, він буде автоматично переміщений в загальнодоступний каталог `/mnt/usb/public/downloads/Complete`. Найпростіший спосіб доступу до загального ресурсу в Linux - ввести його адресу в менеджері файлів за замовчуванням. Більшість сучасних менеджерів файлів Linux, таких як Nautilus Gnome, підтримують Samba. Запустіть Nautilus і натисніть `Ctrl+L` для доступу до адресного рядка.

### 3. ВІДДАЛЕНИЙ ДОСТУП ДО RASPBERRY PI

---

Тепер введіть `smb://` потім IP -адресу RPi з запущеною Samba. У нашому випадку, це було б `smb://192.168.3.100`. Щоб отримати доступ до певної частини, ви можете також додати її назву до кінці адреси, наприклад, `smb://192.168.3.100/documents`. Або можете змонтувати частину з командного рядка:

```
# mount -t cifs -o username=pi,password=raspberry  
//192.168.3.100/usb/downloads /mnt/downloads
```

Для автоматичного монтування розширеної частини ви можете додати її в файл `/etc/fstab`:

```
//192.168.3.100/usb/downloads /mnt/downloads cifs  
username=pi,password=raspberry 0 0
```

Можна підключити віддалений пристрій USB на Windows. Відкрийте на машині Windows 7 My Computer (*Мій комп'ютер*) і натисніть на кнопку Map Network Drive (*Підключити мережевий диск*). У вікні, яке з'явиться, виберіть букву диска і введіть шлях до мережевої папки. Це буде щось на зразок `192.168.3.100\usb\videos`. Вам буде запропоновано ввести інформацію для аутентифікації, потім привід буде відображатися в *Мій комп'ютер*.

#### Розширений доступ з Android-пристрою

Для роботи з розширеними ресурсами Samba ми використаємо популярний файловий менеджер ES File Explorer<sup>54</sup>.

ES File Explorer є безкоштовним, повнофункціональним менеджером файлів і додатків.

Серед основних його функцій: файловий менеджер, менеджер програм, менеджер завантажень, клієнт зберігання в хмарі (сумісний з Dropbox, Google Drive, SkyDrive, Box.net, SugarSync, Яндекс, Amazon S3 і Ubuntu One), клієнт FTP, і клієнт LAN Samba.

ES File Explorer дозволяє користувачам Android, незалежно від того, де вони знаходяться, вільно управляти своїми ресурсами. Ви можете отримати доступ до всіх своїх файлів зі свого мобільного пристрою і обмінюватися ними з іншими. Ця програма дозволяє легко залишатися на зв'язку через 3G, 4G, EDGE або Wi-Fi, щоб поділитися з друзями, завантажувати фотографії і дивитися відео.

У додатку змініть View (Вид) з Local (Локально) на LAN (Мережа), а потім перейдіть в Menu > New > Server (Меню > Створити > Сервер) і введіть параметри з'єднання і аутентифікації.

Залежно від ваших прав, ви зможете завантажувати, зберігати і програвати файли та отримати доступ до публічних і приватних розширених розділів.

---

<sup>54</sup> <https://play.google.com/store/apps/details?id=com.estrongs.android.pop&hl=uk>

## 4. ПРАЦЮЄМО З ВІДЕО

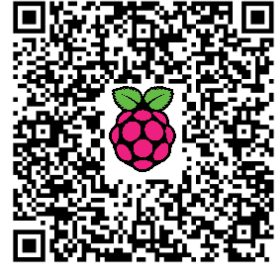
### Як встановити та налаштувати модуль камери Raspberry Pi



Серед цікавих доповнень Raspberry Pi є модуль відеокамери<sup>55</sup>, призначений для підключення до шини CSI за допомогою стрічкового кабелю.

Сама плата камери (рис.4.1) крихітна. Її розміри лише 25x20x9 мм, а важить

всього 3 грами і поставляється з 15 см стрічковим кабелем.



Датчик камери 5 МР, має об'єктив з фіксованим фокусом і дозволяє отримати нерухомі зображення з дозволом 2592x1944. Підтримка відео: 1080p 30 кадрів в секунду (fps), 720p при 60 fps і дозволом 640x480 зі швидкістю до 90 fps. Повна підтримка камери додана до Raspbian ОС, яку потрібно оновити.

#### Початкові налаштування



Для камери виділеною шиною є стрічковий роз'єм ближче до порту Ethernet. Він має конструкцію нульового зусилля вставки (ZIF); треба лише підтягнути два бічні затискачі, щоб звільнити скобу.

Вам необхідно розмістити стрічку з введенням від краю з'єднання поблизу порта Ethernet і до кінця роз'єму живлення Pi. Тримайте квадрат стрічки на місці і розсуньте фіксатори так, щоб зафіксувати стрічку в потрібному положенні. Зверніть увагу на фото, що гнучкий кабель вставляється срібними контактами у бік порту HDMI (рис.4.2):

Підключіть все до Raspberry Pi і завантажтеся з SD-карти, вибравши для установки стандартну ОС Raspbian "Wheezy", а потім перезавантажтеся. Ім'я користувача за замовчуванням і пароль "pi" та "raspberrry", відповідно.

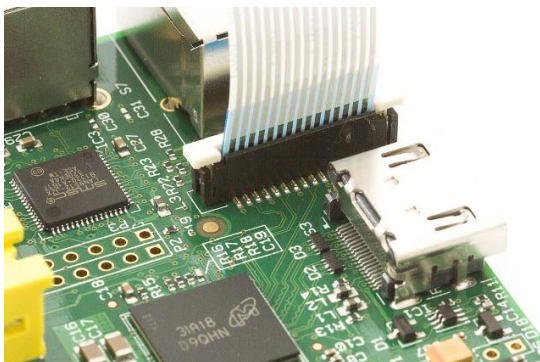


Рис.4.2 Під'єднання гнучкого кабелю камери

Спочатку ми повинні оновити ОС: для цього відкрийте термінал і введіть наступні дві команди, натискаючи клавішу повернення після кожної з них. Вони можуть зайняти чимало хвилин до завершення, тому можете випити чашку чаю.

```
sudo apt-get update  
sudo apt-get upgrade -y
```

Тепер нам потрібно використати інструмент налаштування Raspberry Pi, щоб перевірити розподіл пам'яті GPU, бо потрібно щонайменше 128 як буфер для всіх відеоданих, які ось-ось будуть залиті в пам'ять. При цьому фактично виконується активація шини камери. Введіть у терміналі:

```
sudo raspi-config
```

Спочатку виберіть параметр "Advanced > Memory\_split" (Розширений > Розподіл пам'яті) - він повинен бути вже встановлений на 128, але якщо ні, то внесіть зміни. Система камери, начебто, нормально працює навіть при 16, але рекомендується 128.

Нарешті, є параметр підтримки камери Enable/Disable Camera (включення/вимикання камери). оберіть його (рис.4.3) та Enable (Увімкнути) (рис.4.4):

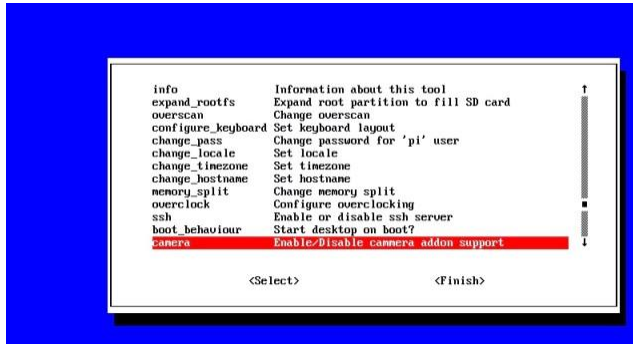


Рис.4.3 Вибір параметру підтримки камери

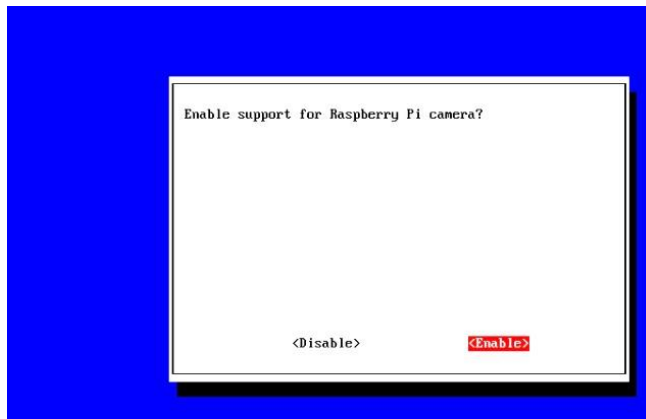


Рис.4.4 Дозвіл модуля камери

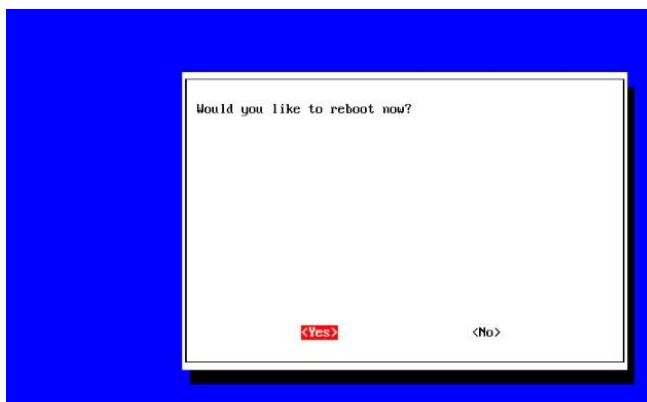


Рис.4.5 Завершення вмикання камери

Виберіть Finish (Готово), а далі Yes - для перезавантаження (рис.4.5).

### Зйомка з Raspberry Pi

Є два інструменти командного рядка для доступу до модуля камери - це `raspivid` і `raspistill`.

Камера пропонує хороший вибір графічних ефектів і загальних параметрів камери, які можна застосувати в прямому ефірі, як для нерухомих зображень так і для відео. Ви в змозі змінити режими балансу білого, фокусування, тип фото, експозицію, рівні ISO і дані EXIF.

Нижче наведені приклади команд з терміналу і що вони роблять, час захоплення за замовчуванням 5 секунд, скористайтесь параметром `-t`, щоб вказати більш тривалий період в мілісекундах, так `-t 20000` означає протягом 20 секунд.

### Відео

Відображення п'ять секунд в режимі демо: `raspivid -d`

Відображення 10 с відео в демо-режимі:

`raspivid -o video.h264 -t 10000 -d`

Відображення 640x480 прев'ю: `raspivid -p 0,0,640,480`

Захоплення 20 секунд відео у форматі h264:

`raspivid -t 20000 -o video.h264`

### Використання параметрів зображення в команді `raspivid`

Синтаксис: `RaspiVid [параметри]`

Параметри команд для зображення:

`-?`, - допомога: Інформація для допомоги

`-w`, - ширина: Встановить ширину зображення `<size>`. За замовчуванням 1920

`-h`, - висота: Зазначена висота зображення `<size>`. За замовчуванням 1080

`-b`, - бітрейт: Вказаний бітрейт. Використовуйте біти в секунду (наприклад, 10Mbps/s буде `-b 10000000`)

`-o`, - вихід: Вихідний файл `<filename>` (для запису на stdout, використати `-o -`)

`-v`, - докладно: Виведення подробиць під час роботи

`-t`, - час очікування: Час (в мс) до фотографування і відключення. Якщо не вказано, встановлюється на 5 секунд

`-d`, - демо: Запуск демонстраційного режиму (цикл через ряд параметрів камери, захвату немає)

`-fps` - частота кадрів: Встановить частоту кадрів в секунду для запису

`-e`, - репс: Показати зображення для попереднього перегляду \*після\* кодування (показати артефакти стиснення)

Параметри команд попереднього перегляду:

`-p`, - попередній перегляд: Параметри вікна перегляду `<'x,y,w,h'>`

-f, - повноекранний: Повноекранний режим перегляду  
 -n, - без попереднього перегляду: Не відобразити вікно попереднього перегляду

## Фото

Захоплення зображення в JPEG форматі:

```
raspistill -o image.jpg
```

Зробити фото 640x480:

```
raspistill -o image.jpg -w 640 -h 480
```

Зробити JPEG зниженої якості: `raspistill -o image.jpg -q 5`

## Використання параметрів зображення в команді Raspistill

Синтаксис: `Raspistill [параметри]`

Параметри команд для зображення:

-?, - допомога: Інформація для допомоги  
 -w, - ширина: Встановить ширину зображення <size>  
 -h, - висота: Зазначена висота зображення <size>  
 -q, - якість: Встановлення якості JPEG <0 до 100>  
 -r, - raw: Додати дані raw bayer до JPEG метаданих  
 -o, - вихід: Вихідний файл <filename> (для запису на stdout, використати -o -). Якщо не вказано, файл не зберігається  
 -v, - докладно: Виведення подробиць під час роботи  
 -t, - час очікування: Час (в мс) до фотографування і вимикання (якщо не вказано, то встановлюється в 5 секунд)  
 -th, - thumb: Встановити параметри ескізу (x:y:якість)  
 -d, - демо: Запуск демонстраційного режиму (цикл через ряд параметрів камери, захвату немає)  
 -e, - кодування: Кодування, яке використовується для вихідного файлу (JPG, BMP, GIF, PNG)  
 -x, - exif: Tag EXIF для застосування до захоплень (формат як **key=value**)  
 -tl, - timelapse: Режим інтервальної зйомки. Робить знімок кожні <t> мс

Параметри команд попереднього перегляду:

-p, - попередній перегляд: Параметри вікна перегляду <'x,y,w,h'>  
 -f, - повноекранний: Повноекранний режим перегляду  
 -n, - без попереднього перегляду: Не відобразити вікно попереднього

Додаткові параметри команд для відео і зображення:

-sh, - різкість: Встановлення різкості зображення (від -100 до +100)  
 -co, - контраст: Установка контрастності зображення (від -100 до +100)  
 -br, - яскравість: Установка яскравості зображення (від 0 до 100)  
 -sa, - насичення: Встановлення насиченості зображення (від -100 до +100)  
 -iso, - ISO: Вказане ISO для захоплення

-**vs**, - **vstab**: Увімкнути відео стабілізацію  
-**ev**, - **ev**: Встановити експокорекцію  
-**ex**, - експозиція: Встановити режим експозиції  
-**awb**, - **awb**: Встановити режим AWB  
-**ifx**, - **imxfx**: Встановити ефект зображення  
-**cfx**, - **colfx**: Встановити колірний ефект (U:V)  
-**mm**, - **metering**: Встановити режим виміру експозиції  
-**rot**, - обертання: Поворот зображення (0-359)  
-**hf**, - **hflip**: Горизонтальний фліп  
-**vf**, - **vflip** : Вертикальний фліп

За замовчуванням папка для збереження зображень `/home/pi` тому, якщо вам просто необхідно програти збережене відео, то найпростіше завантажитися в графічний інтерфейс з **startx**, відкрити "File Manager" і використати "LXTerminal " для команд камери.

### Як організувати потокове відео з Raspberry Pi

#### **Для перегляду каналу на Linux**

Встановіть необхідне програмне забезпечення, виконавши наступну команду з терміналу:

```
sudo apt-get install mplayer netcat
```

Знайдіть IP-адресу свого Pi, запустивши `ifconfig`. (Ваша IP-адреса буде вказана у виведенні на консоль і, ймовірно, матиме вигляд 192.168.1.XXX).

Виконайте наступну команду з терміналу, щоб переглядати канал за допомогою MPlayer:

```
nc -l -p 5001 | mplayer -fps 31 -cache 1024 -
```

#### **Для перегляду каналу на ОС Windows**

Установіть і запустіть Linux на Pi.

Знайдіть IP-адресу свого Pi, запустивши `ifconfig`. (Ваша IP-адреса буде вказана у виведенні на консоль і, ймовірно, матиме вигляд 192.168.1.XXX).

Завантажте MPlayer<sup>56</sup>.

Завантажте Netcat<sup>57</sup>.

Зверніть увагу, що ваш браузер може скажитися, що ці файли шкідливі, оскільки вони без цифрового підпису.

---

<sup>56</sup> <https://code.google.com/p/mplayer-for-windows/downloads/detail?name=mplayer-svn-36251.7z&can=2&q=>

<sup>57</sup> <http://joncraton.org/media/files/nc111nt.zip>

Натисніть клавішу Windows, і клавішу "r" одночасно, щоб викликати діалогове вікно "Run" ("Виконати"). Введіть cmd.exe в діалоговому вікні і натисніть Enter/Return, щоб відкрити рядок DOS.

Введіть наведену нижче команду в командному рядку, щоб переглянути канал, використовуючи MPlayer:

```
[Шлях до nc.exe]\nc.exe -l -p 5001 | [Шлях до  
mplayer.exe]\mplayer.exe -fps 31 -cache 1024 -
```

### **Для перегляду каналу на OS X**

Завантажте MPlayer<sup>58</sup>

Крім того, ви можете завантажити mplayer, використовуючи Brew<sup>59</sup>, якого рекомендуємо.

Знайдіть IP-адресу свого Pi, запустивши ifconfig. (Ваша IP-адреса буде вказана у виведенні на консоль і, ймовірно, матиме вигляд 192.168.1.XXX).

Виконайте наступну команду з терміналу, щоб переглядати канал за допомогою MPlayer:

```
nc -l -p 5001 | mplayer -fps 31 -cache 1024 -
```

### **Для перегляду каналу на Raspberry Pi**

Знайдіть IP-адресу свого Pi, запустивши ifconfig. (Ваша IP-адреса буде вказана у виведенні на консоль і, ймовірно, матиме вигляд 192.168.1.XXX).

Виконайте наступні команди в терміналі на приймаючому Pi:

```
mkfifo buffer  
nc -p 5001 -l > buffer |  
/opt/vc/src/hello_pi/hello_video/hello_video.bin buffer
```

### **Для передачі каналу з Pi, до якого приєднаний модуль камери**

Після налаштування "приймальної" машини згідно з наведеними вище інструкціями, виконайте наступні команди з терміналу на "передавальному" Pi:

```
raspivid -t 999999 -o - | nc [вставте IP-адресу клієнта] 5001
```

Ви можете використовувати команди, перераховані вище при розгляді програмного забезпечення камери, щоб змінити час захоплення або додати відео-ефект.

<sup>58</sup> <http://sourceforge.net/projects/mplayerosx/files/MPlayer>

<sup>59</sup> <http://mxcl.github.io/homebrew/>

### Перегляд відео на OMXPlayer з командного рядка



Інтерфейс командного рядка в Raspberry Pi має повний доступ до відеопідсистеми (наприклад, до буфера кадрів), а це означає, що утиліти командного рядка можуть відображати складні зображення і відео без запуску повного графічного робочого столу<sup>60</sup>.

Це може бути корисним для ігор, які запускаються з командного рядка, а також для інших мультимедійних програм, відеопрогравачів.

Серед них OMXPlayer, який є відеоплеєром командного рядка для Raspberry Pi. Спочатку він був розроблений як стенд для реалізації XBMC на Raspberry Pi, але він працює однаково добре як автономний відеоплеєр.

Щоб встановити програвач, використайте наступні команди:

```
sudo apt-get update
sudo apt-get -y install omxplayer
```

Щоб запустити його, введіть `omxplayer` разом з ім'ям файлу відео, яке хочете програти (рис.4.6):

```
omxplayer myvideo.mp4
```

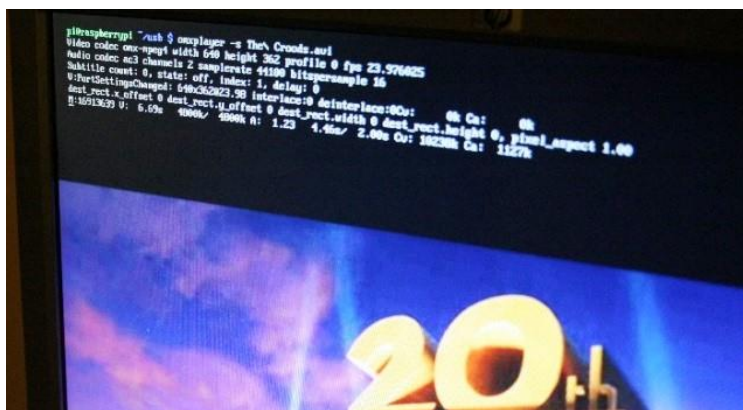
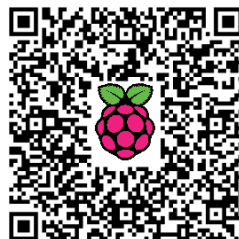


Рис.4.6 Програвання відео на omxplayer

Програвач, швидше за все, «візьме паузу» на кілька секунд до того, як показати відео, бо Raspberry не найшвидший комп'ютер у світі і

<sup>60</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1804-watch-the-video-for-raspberry-pi-from-the-command-line>

відеопрогравачу потрібно виконати попередню обробку відеофайлу (наприклад, він повинен визначити, які використовуються кодеки в файлі і так далі).

## Інші параметри командного рядка

Якщо введете:

**omxplayer**

Вам буде наданий список доступних параметрів командного рядка:

Синтаксис команди: **omxplayer** [ПАРАМЕТРИ] [ФАЙЛ]

Параметри:

|                               |                                                                                                    |
|-------------------------------|----------------------------------------------------------------------------------------------------|
| <b>-h / --help</b>            | показати допомогу                                                                                  |
| <b>-n / --aidx index</b>      | індекс аудіо потоку: наприклад, 1                                                                  |
| <b>-o / --adev device</b>     | вихідний аудіо пристрій:<br>наприклад, hdmi/локальний                                              |
| <b>-i / --info</b>            | формат дампу потоку і вихід                                                                        |
| <b>-s / --stats</b>           | статистика pts і буфера                                                                            |
| <b>-p / --passthrough</b>     | наскрізний прохід звуку                                                                            |
| <b>-d / --deinterlace</b>     | деінтерлейсінг                                                                                     |
| <b>-w / --hw</b>              | HW декодування аудіо                                                                               |
| <b>-3 / --3d mode</b>         | перемикання ТБ в 3D режим<br>(наприклад, SBS/TB)                                                   |
| <b>-y / --hdmiclocksync</b>   | налаштувати частоту оновлення дисплея<br>відповідно відео (за замовчуванням)                       |
| <b>-z / --nohdmiclocksync</b> | не налаштована частота оновлення<br>екрану відповідно відео                                        |
| <b>-t / --sid index</b>       | показ субтитрів з індексом                                                                         |
| <b>-r / --refresh</b>         | регулювання частоти кадрів/дозволу<br>для відео                                                    |
| <b>-l / --pos</b>             | стартова позиція (у секундах)                                                                      |
| <b>--boost-on-downmix</b>     | пониження гучності при мікшуванні                                                                  |
| <b>--subtitles path</b>       | зовнішні субтитри в srt-формі<br>при кодуванні UTF-8                                               |
| <b>--font path</b>            | шлях до шрифту субтитрів (за умовчуванням:<br>/usr/share/fonts/truetype/freefont<br>/FreeSans.ttf) |
| <b>--font-size size</b>       | розмір шрифту, як тисячних висоти<br>екрана (за замовчуванням 55)                                  |
| <b>--align left/center</b>    | вирівнювання субтитрів<br>(за замовчуванням left)                                                  |

**--lines n**               число рядків для розміщення в буфер субтитрів (за замовчуванням 3)

В процесі програвання відео, Ви можете керувати відтворенням за допомогою клавіатури. Використайте SPACE (Пробіл) для паузи/резюме, стрілку вправо, щоб перейти вперед на 30 секунд, стрілку вгору, щоб перейти на 10 хвилин вперед, + або -, щоб змінити гучність, і q, щоб зупинити програвання.

## Сполучення клавіш

Поєднання клавіш, які можна використовувати для управління функціями програвача:

1 – зменшення швидкості  
2 – збільшення швидкості  
< - перемотування  
> - швидке перемотування вперед  
z – показати інформацію  
j – попередній аудіо потік  
k – наступний аудіо потік  
i – попередній розділ  
o – наступний розділ  
n – попередній потік субтитрів  
m – наступний потік субтитрів  
s – перемикання субтитрів  
d – зменшити затримку субтитрів (- 250 мс)  
f – збільшити затримку субтитрів (+ 250 мс)  
q - вихід отхplayer  
p / пробіл - пауза/грезюме  
- - зменшити гучність  
+ / = - збільшити гучність  
стрілка вліво - назад на 30 сек.  
стрілка вправо – вперед на 30 сек.  
стрілка вниз – назад на 600 сек.  
стрілка ввверх – вперед на 600 сек.

Ви можете виявити, що елементи управління перемотування назад і вперед не спрацьовують. Під час тестування автор виявив, що на екрані (OSD) було повідомлення, що швидкість відтворення змінилася, але відео продовжувало програватися з тією ж швидкістю.

OMXPlayer має тенденцію до виведення багато інформації в командному рядку. Оскільки відео накладається на командний рядок, ці додаткові тексти можуть визвати незадоволення. Щоб включити чорний фон і приховати додатковий вихід, треба запускати плеєр з параметром "-b":

```
omxplayer -b myvideo.mp4
```

За замовчуванням звук направляється в аудіо роз'єм Raspberry Pi, однак, HDMI здатний програвати звук, як і відео. Тому, якщо ваш Pi підключений до телевізора (а не до монітора), то можете відправити аудіо через HDMI за допомогою опції `"-o hdmi"`:

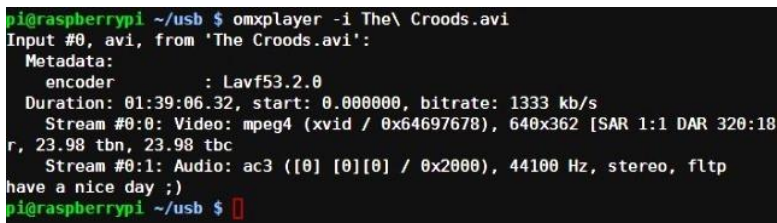
```
omxplayer -o hdmi myvideo.mp4
```

Raspberry Pi не є достатньо потужним, щоб виконувати декодування відео програмно. Це означає, що, хоча він може декодувати ряд популярних форматів відео, він не буде працювати з такими менш популярними кодеками, як DivX 3.0. Він може програти H.264 і MPEG-4 відео в 1080p HD, а також MJPEG, VP6, VP8 і OGG Theora відео.

Ці останні чотири обмежені якістю SD, так як вони не повністю підтримуються на апаратному рівні, і декодування виконується програмно за допомогою GPU. Якщо вам треба програти MPEG2 або VC-1 (використовується в деяких Blu-Ray дисках), то необхідно купити додаткові ліцензії відео кодеків в інтернет-магазині від Raspberry Pi Foundation<sup>61</sup>.

Якщо отримуєте повідомлення про помилку, зв'язану з невідомими або непідтримуваними кодеками, то є ймовірність, що OMXPlayer не може відтворити відео. Вашою найкращою поведінкою у цій ситуації було б перекодувати відео на ПК за допомогою H.264. Ви можете задати OMXPlayer відображати інформацію про відеофайл, використовуючи параметр `-i` (рис.4.7):

```
omxplayer -i The\ Croods.avi
```



```
pi@raspberrypi ~/usb $ omxplayer -i The\ Croods.avi
Input #0, avi, from 'The Croods.avi':
  Metadata:
    encoder           : Lavf53.2.0
  Duration: 01:39:06.32, start: 0.000000, bitrate: 1333 kb/s
    Stream #0:0: Video: mpeg4 (xvid / 0x64697678), 640x362 [SAR 1:1 DAR 320:180], 23.98 tbn, 23.98 tbc
    Stream #0:1: Audio: ac3 ([0] [0][0] / 0x2000), 44100 Hz, stereo, fltp
have a nice day ;)
pi@raspberrypi ~/usb $
```

Рис.4.7 Відображення інформації про відеофайл

Додаткова інформація може допомогти вам визначити, чи сумісне відео з Pi.

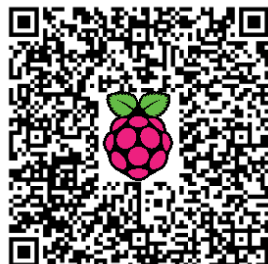
Omxplayer особливо корисний, якщо створюєте відео за допомогою модуля камери Raspberry Pi і хочете відтворювати файли з командного рядка.

<sup>61</sup> <http://www.raspberrypi.com/license-keys/>

## Переглядаємо відеопотік з камери через веб-браузер



Розглянемо, як перетворити Raspberry Pi в сервер вебкамери. Ви зможете переглядати відеопотік з камери на будь-якому пристрої, що має веб-браузер<sup>62</sup>. Ним також може бути iPad/iPhone і Android.



### Офіційний спосіб формування потоку

В одному з попередніх розділів про модуль камери показаний метод передавання потокового відео від Raspberry Pi на інший комп'ютер. Цей метод працює таким чином:

- На Pi використовується утиліта `raspivid` для кодування відео з камери у форматі H.264.
- Відеопотік передається конвеєром утиліті `nc`, яка «проштовхує» його на вказану мережеву адресу, де є відеоплеєр.
- На комп'ютері з програвачем `nc` приймає потік і направляє його в `mplayer` для програвання.

Це ефективний метод потокового відео від Pi до іншого комп'ютера, але він має кілька проблем:

- Raspberry Pi необхідно знати адресу комп'ютера, на якому програватиметься відео.
- Комп'ютер, на якому програватиметься відео, повинен мати встановлений сучасний плеєр, який може програвати «сирий» відеопотік H.264. Немає такого мобільного пристрою, який може це робити.
- Оскільки вказана система спирається на прямий зв'язок між Pi і програвачем, то не можна комп'ютер з програвачем підключити і/або відключитися від потоку - з'єднання повинно бути весь час.
- Що робити, якщо є два або три конкурентні програвачі? Все стає жахливо складним для Pi.

Тому варто пошукати інші рішення, щоб отримати можливість переглядати відеопотік, наприклад, у своєму браузері.

### Потокові протоколи

Важливою вимогою для потокової камери є можливість простого її перегляду. Часто це означає, що потік повинен бути відтворений у веб-браузері. Запуск користувальницького програвача є ускладненням, бо робить його недоступним для більшості мобільних пристроїв.

<sup>62</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1811-browsing-the-video-feed-from-the-camera-raspberry-pi-via-a-web-browser>

Є кілька сучасних потокових протоколів для веб-браузерів. Наприклад, HLS є вибором компанії Apple, тому він має широку підтримку на мобільним іПристроях, але не на інших. Інший, що називається Fragmented MP4, підтримується Adobe і Microsoft, але вимагає плагіни до браузера від цих компаній для програвання на комп'ютері, тому комп'ютери з ОС Windows і Mac можуть це зробити, а Linux і мобільні пристрої - ні. HTML5-відео також засноване на форматі MP4, але не така вже й широка його підтримка.

Крім того, для всіх перерахованих вище потокових протоколів існує необхідність мати потоковий сервер, який готує відео для потокової передачі, сегментуючи його і упаковуючи. Хоча є кілька утиліт з відкритим вихідним кодом, який може зробити це для статичного відеопотоку, але автор не знайшов жодного, який може зробити це для живого потоку, за винятком свіжих релізів **ffmpeg**, доступних для Raspbian, які можуть генерувати живий потік HLS.

Які є інші варіанти?

## **Motion JPEG на допомогу**

Досліджуючи, як працюють IP-вебкамери, можна побачити, що багато з них використовують старий потоковий протокол під назвою Motion JPEG або скорочено MJPEG.

Motion JPEG - це просто потік окремих JPEG фотографій, одна за одною. Цікаво, що більшість сучасних браузерів можуть програвати потоки MJPEG.

Зворотною стороною потоків MJPEG є те, що вони не такі ефективні, як H.264, який значно підвищує якість і зменшує розмір шляхом кодування тільки відмінності одного кадру від іншого. З MJPEG кожен кадр кодується для всього зображення JPEG. Інколи це не має значення.

Продовжуючи дослідження, вдається знайти MJPG-стрімер, невеликий MJPEG потоковий сервер з відкритим вихідним кодом, написаний на C, який можна легко зібрати для Raspberry Pi.

Опишемо, як використати цей інструмент для створення гнучкого, для запуску в будь-якому місці потокового сервера для камери RPi.

### **1. Встановіть збірки залежностей**

Наступна команда встановлює три бібліотеки, які використовує MJPG-Streamer:

```
$ sudo apt-get install libjpeg8-dev imagemagick libv4l-dev
```

### **2. Додайте відсутній videodev.h**

Файл заголовка **videodev.h**, що потрібний MJPG-Streamer, був замінений з **videodev2.h**. Щоб MJPG-Streamer був «щасливий», ви повинні створити символічне посилання:

```
$ sudo ln -s /usr/include/linux/videodev2.h  
/usr/include/linux/videodev.h
```

### 3. Завантажте MJPG-Streamer

Вихідний код для MJPG-Streamer доступний на sourceforge.net, але складно знайти пряме посилання для завантаження:

```
$ wget http://sourceforge.net/code-snapshots/svn/m/mj/mjpg-  
streamer/code/mjpg-streamer-code-182.zip
```

### 4. Розпакуйте вихідний код MJPG-Streamer

Вихідний код завантаження являє собою стислий ZIP-файл. Помістіть файл у свою домашню директорію (або тимчасову папку, якщо віддаєте перевагу їй) і виконайте наступне витягання файлів:

```
$ unzip mjpg-streamer-code-182.zip
```

### 5. Побудуйте MJPG-Streamer

MJPG-Streamer поставляється з декількома плагінами, але тільки кілька з них необхідні для потокового відео. Наведені нижче команди створюють тільки те, що нам необхідне:

```
$ cd mjpg-streamer-code-182/mjpg-streamer  
$ make mjpg_streamer input_file.so output_http.so
```

### 6. Встановіть MJPG-Streamer

Наступні команди скопіюють всі необхідні файли в системні директорії:

```
$ sudo cp mjpg_streamer /usr/local/bin  
$ sudo cp output_http.so input_file.so /usr/local/lib/  
$ sudo cp -R www /usr/local/www
```

### 7. Включіть камеру

Ми майже у меті. Тепер настав час, щоб запустити модуль камери:

```
$ mkdir /tmp/stream  
$ raspistill --nopreview -w 640 -h 480 -q 5 -o  
/tmp/stream/pic.jpg -tl 100 -t 9999999 -th 0:0:0 &
```

Звичайно, ви можете використовувати інші опції raspistill, якщо хочете.

### 8. Запустіть MJPG-Streamer

Тепер камера записує зображення, тому все, що залишилося, запустити MJPG-Streamer:

```
LD_LIBRARY_PATH=/usr/local/lib mjpg_streamer -i "input_file.so -f /tmp/stream -n pic.jpg" -o "output_http.so -w /usr/local/www"
```

Давайте розберемо цю команду, щоб зрозуміти її:

- **LD\_LIBRARY\_PATH** задає шлях до динамічних бібліотек у поточній директорії. Це пов'язано з тим, що додаток може знайти плагіни, які знаходяться в тій же директорії.
- **-i** встановлює вхідний плагін. Плагін, який ми використовуємо, називається **input\_file.so**. Цей плагін переглядає директорію і в будь-який час виявляє записаний в неї файл JPEG, щоб передати цей файл в потік. Папка для перегляду задається як аргумент **-f**.
- **-o** встановлює вихідний плагін. Ми використовуємо потоковий плагін HTTP, який запускає веб-сервер, до якого можемо підключитися, щоб подивитися відео. Кореневий каталог веб-сервера задається як аргумент **-w**. Ми будемо використовувати веб-сторінки за замовчуванням, які надходять відразу з додатком, але вони можуть бути змінені і налаштовані по мірі необхідності.

## 9. Дивіться Stream!

Тепер можете підключитися за допомогою веб-браузера і дивитися потокове відео (рис.4.8). Якщо хочете дивитися з того ж Raspberry Pi, можна ввести в адресному рядку браузера **http://localhost:8080**. Якщо хочете дивитися з іншого комп'ютера у своїй мережі - використайте **http://<IP-адреса Raspberry Pi>:8080**.

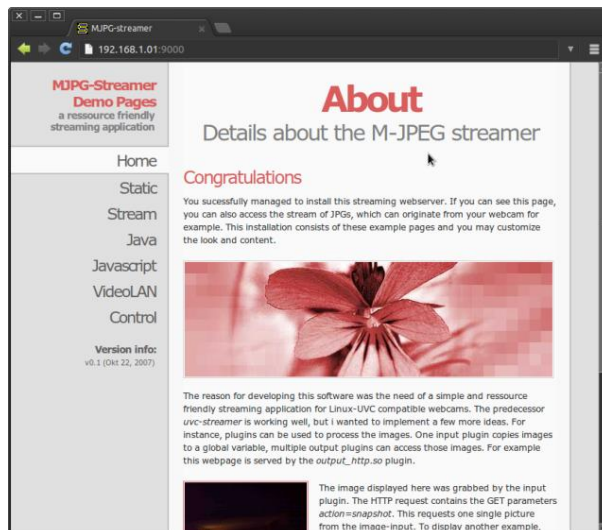


Рис.4.8 Перегляд потокового відео у браузері

### 10. Прибирання

Після того, як переконаєтеся, що все працює, можна видалити вихідний код:

```
$ cd ../../
$ rm -rf mjpg-streamer-182
```

### Оновлення з плагіном `raspicam`

На [github](https://github.com/jacksonliam/mjpg-streamer)<sup>63</sup> є варіант MJPG Streamer, який поставляється з вхідним плагіном `raspicam`. Використовуючи дану версію MJPG-Streamer, можемо пропустити `raspistill` і нехай плагін `raspicam` піклується про створення зображення. Ось як його встановити.

```
# Установка dev-версії libjpeg:
sudo apt-get install libjpeg62-dev
# Установка cmake:
sudo apt-get install cmake
# Завантаження mjpg-streamer з плагіном raspicam:
git clone https://github.com/jacksonliam/mjpg-streamer.git
~/mjpg-streamer
# Зміна директорії:
cd ~/mjpg-streamer/mjpg-streamer-experimental
# Компіляція
make clean all
# Заміна старого MJPG-стрімера:
sudo rm -rf /opt/mjpg-streamer
sudo mv ~/mjpg-streamer/mjpg-streamer-experimental /opt/mjpg-streamer
sudo rm -rf ~/mjpg-streamer
# Початок потоку:
LD_LIBRARY_PATH=/opt/mjpg-streamer/ /opt/mjpg-streamer/mjpg_streamer -i "input_raspicam.so -fps 15 -q 50 -x 640 -y 480" -o "output_http.so -p 9000 -w /opt/mjpg-streamer/www" &
```

Старт/стоп сценарію оболонки ([gist](https://gist.github.com/miguelmota/9201942)<sup>64</sup>).

Автор вважає, що це рішення працює набагато коректніше, ніж попереднє.

---

<sup>63</sup> <https://github.com/jacksonliam/mjpg-streamer>

<sup>64</sup> <https://gist.github.com/miguelmota/9201942>

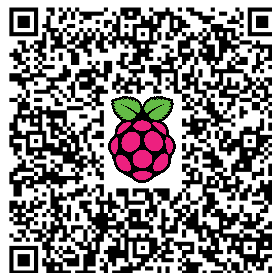
## Як передати відеопотік на планшет чи смартфон з VLC



Якщо встановите VLC на Pi, то зможете передавати відео з Pi, а потім переглядати цей потік за допомогою іншого віддаленого пристрою<sup>65</sup>. Автор вважає, що всі програми, які можуть переглядати потік .h264, будуть працювати.

На жаль, багато планшетників просто не можуть впоратися з потоком кадрів 1080 пікселів (р) 30 кадрів в секунду (fps), які може забезпечити Pi. Автор намагався приймати 720р 30fps і 25fps, але нічого не вийшло. 720р (1280 x 720) майже працює, але картинка розпадалася.

Nexus чудово справляється з 640 x 360 @ 25fps. Також добре працює він і при 800 x 450 @ 25fps, якщо не занадто далеко від маршрутизатора. 900 x 506, здається, масимум, з яким він може впоратися. Як не дивно, автор думає, що це обмеження Nexus, а не Pi. Pi може забезпечити бездротову передачу відео на ноутбук при більш високому дозволі (720р @ 25fps) без розпаду картинки.



### ОК, так як це реально зробити?

Вам потрібно встановити VLC на обох пристроях. Щоб встановити VLC на Pi:

```
sudo apt-get update
sudo apt-get install vlc
```

Більш детально про VLC можна почитати на сайті VideoLan<sup>66</sup>.

На Nexus, або іншому android-пристрої, перейдіть на Google Play і встановіть VLC for Android Beta<sup>67</sup>. Не хвилюйтеся про бета попередження – версія чудово працює.



### Запуск потокового відео

#### Використання RTSP

Найбільш надійним методом потокового відео є, здається, RTSP і, щоб запустити введіть в командному рядку:

<sup>65</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1813-how-to-transfer-video-feed-from-the-camera-raspberry-pi-tablet-or-smartphone-via-vlc>

<sup>66</sup> <http://www.videolan.org/>

<sup>67</sup> <https://play.google.com/store/apps/details?id=org.videolan.vlc.betav7neon&hl=uk>

#### 4. ПРАЦЮЄМО З ВІДЕО

```
raspivid -o - -t 99999999 -w 640 -h 360 -fps 25|cvlc -vvv  
stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264
```

В цьому випадку відео з камери передається як потік з порту 8554 вашого Raspberry Pi.

99999 - тривалість часу в мілісекундах, тобто потік буде працювати протягом 99,9 секунди. Ви можете змінити це значення на те, яке вам подобається. Якщо зміните його на 0 (нуль), то потік продовжиться на невизначений термін - треба буде натиснути CTRL+C, щоб убити його в будь-який момент.

Знайте, що отримаєте багато налагоджувальних повідомлень на екрані консолі (кілька екранів). Це те, що відбувається, коли все працює. Ігноруйте їх.

Для вибору підтримуваних параметрів для даного мобільного пристрою можна скористатися таблицею:

| Name              | 480p | 576p | 720p | MKV 720p + ASS |
|-------------------|------|------|------|----------------|
| HTC One X         | ✓    | ✓    | ✓    | ✓              |
| Samsung Galaxy S3 | ✓    | ✓    | ✓    | ✓              |
| Galaxy Nexus      | ✓    | ✓    | ✓    | ?              |
| Samsung Galaxy S2 | ✓    | ✓    | ?    | ?              |
| HTC Desire HD     | ✓    | ?    | ✗    | ✗              |
| Nexus One         | ✓    | ✗    | ✗    | ✗              |

Крім того, зверніть увагу, що потокове закінчується в кінці 99 999 мс періоду часу і вам доведеться перервати процес з CTRL+C, щоб повернутися під контроль вашої консолі.

Відео можна побачити на віддаленому пристрої, запустивши VLC і відкривши Network Stream за адресою `rtsp://<IPofRaspberryPi>:8554/`

Якщо не знаєте IP-адреси Pi, введіть в консолі `ifconfig` і повинні бути в змозі побачити свою IP-адресу. Як альтернативу, можете використати чудовий безкоштовний інструмент Fing- Network Tools<sup>68</sup>, який сканує вашу мережу і показує IP-адреси всіх пристроїв, що під'єднані.

Після того, як ви це зробили, повинні побачити вихідний потік камери на екрані свого Nexus (за умови, що ви не витратили вже понад 99,9 секунд на введення URL - використовуйте «копіювати і вставити», щоб зробити це швидко і легко).



Після того, як все запрацює, можете почати експериментувати, щоб змінити ширину, висоту і частоту кадрів через параметри (`-w 640 -h 360 -fps 25`), щоб побачити, наскільки ваш Nexus і мережа можуть впоратися. Тут буде залежати, наскільки ви далеко від маршрутизатора.

<sup>68</sup> <https://play.google.com/store/apps/details?id=com.overlook.android.fing&hl=uk>

## Потік через HTTP

RTSP добре працює в локальній мережі, але може бути важко виставити дозвіл маршрутизатору для потокового мовлення через Інтернет. Для цього VLC можна налаштувати на потік через HTTP, що, здається, використовуватиме трохи більше ресурсів процесора на Raspberry Pi і можуть бути пропущені кадри, проте використовується тільки один порт і цей порт може легко бути перенаправлений через більшість стандартних маршрутизаторів.

Команда для даної конфігурації:

```
raspivid -o - -t 9999999 -w 640 -h 360 -fps 25|cvlc -vvv
stream:///dev/stdin --sout
'#standard{access=http,mux=ts,dst=:8554}' :demux=h264
```

Знову, відео можна подивитися за допомогою VLC, відкривши Network Stream з адресою `http://<IPofRaspberryPi>:8554/`

## Використання HLS

Останнім способом потокового відео є використання HLS (HTTP Live Streaming). Це, мабуть, дозволить відео, яке легко передається на пристрої iOS. Для запуску цього способу треба встановити на Raspberry Pi веб-сервер. Популярним вибором буде Apache, який можна встановити за допомогою:

```
sudo apt-get install apache
```

Файли, які потім будуть потрібні, зберігаються в `/var/www`. Для налаштування VLC для генерації HLS-потoku можете запустити:

```
raspivid -o - -w 920 -h 540 -t 9999999 | vlc -v -I "dummy"
stream:///dev/stdin
:sout="#std{access=livehttp{seglen=10,delsegs=true,numsegs=5,
index=/var/www/streaming/stream.m3u8, index-
url=http://<IPofRaspberryPi>/streaming/stream-#####.ts},
mux=ts{use-key-frames}, dst=/var/www/streaming/stream-
#####.ts}" :demux=h264
```

Нагадуємо, що треба встановити правильну IP-адресу вашого Raspberry Pi. Знову ж, відео може бути відтворене за допомогою VLC, відкривши Network Stream:

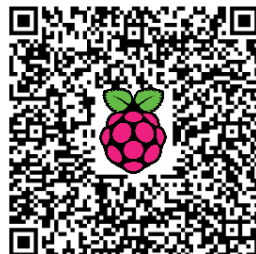
```
http://<IPofRaspberryPi>/streaming/stream.m3u8
```

Звертаємо вашу увагу, що HLS може забрати всю потужність CPU на Raspberry Pi

## Як створити потік з веб-камери на Raspberry Pi з ffmpeg



Багато користувачів мали проблеми з потоковим відео для Raspberry Pi, тому пропонуємо ще один метод, щоб допомогти вам переглядати відео з веб-камери віддалено через браузер<sup>69</sup>.



### Компіляція ffmpeg для Webcam Streaming

Перш за все, ми повинні отримати версію **ffmpeg**, яка може передавати. Тому почнемо з установки **git**, як Raspbian Server Edition (RSE), але не виконувати **git**.

Автор мав права адміністратора, коли виконував решту наведених нижче команд, тому, якщо не маєте таких прав, то скористайтеся **sudo** **bash**, щоб стати адміністратором. Ми просто додаємо **sudo** до всього:

```
sudo apt-get install git
```

Після того, як **git** встановиться, перейдіть в **/usr/src**, щоб завантажити код:

```
cd /usr/src
sudo git clone git://git.videolan.org/ffmpeg.git
```

Git витягне первинний код, з якого треба побудувати **ffmpeg** з нуля.

**Примітка.** Якщо вам треба мати звук з **ffmpeg**, то встановіть також пакет **libasound2-dev**, який дозволить ALSA.

```
cd ffmpeg
sudo ./configure
sudo make && make install
```

**Примітка.** Компіляція **ffmpeg** на Pi триває близько 5 годин, тому автор залишив працювати Pi на всю ніч, щоб закінчити її.

Після компіляції:

- Додайте наступні рядки в **/etc/apt/sources.list**:

```
deb-src http://www.deb-multimedia.org sid main
deb http://www.deb-multimedia.org wheezy main non-free
```

---

<sup>69</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1814-how-to-create-a-stream-from-a-webcam-on-raspberry-pi-with-ffmpeg>

- Виконайте команди:

```
sudo apt-get update
sudo apt-get install deb-multimedia-keyring
```

- Видаліть другий рядок з `/etc/apt/sources.list`:

```
deb http://www.deb-multimedia.org wheezy main non-free
```

- Виконайте команду:

```
sudo apt-get source ffmpeg-dmo
```

*Тепер ви вже повинні мати папку з ім'ям `ffmpeg-dmo-2.2` <-  
Номер версії може змінитися з плином часу*

- Перейдіть в папку, яка містить джерело, наприклад:

```
sudo cd ffmpeg-dmo-2.2
```

- Налаштуйте джерело:

```
sudo ./configure
```

- Зберіть і встановіть `ffmpeg`:

```
sudo make && make install
```

І знову набирайтеся терпіння: зачекайте до закінчення близько 5 годин.

### Налаштування `ffmpeg`

Після установки `ffmpeg`, нам потрібно створити файл конфігурації для включення потоку `ffmpeg` на `ffserver` (`ffserver` - це те, через що проходить потік).

Створимо файл конфігурації для `ffserver`, який розмістимо в `/etc/` з назвою `ffserver.conf`:

```
Port 81
BindAddress 0.0.0.0
MaxClients 10
MaxBandwidth 50000
<Feed webcam.ffm>
file /tmp/webcam.ffm
FileMaxSize 10M
</Feed>
```

```
<Stream webcam.avi>
Feed webcam.ffm
Format avi
VideoSize 176x128
VideoFrameRate 15
VideoBufferSize 40
VideoBitRate 64
AudioBitRate 32
AudioChannels 1
AudioSampleRate 11025
VideoQMin 1
VideoQMax 20
</Stream>
```

Останні рядки визначають розмір потоку, бітрейт. Якщо параметри не підходять один до одного, то потік не буде плавним. Крім того, при наведений вище конфігурації `ffmpeg` буде запускатися з аудіо і потоковим відео.

Наступна команда повинна бути розміщена в файлі сценарію `.sh`. Це дозволить вам запускати потік просто визвавши файл `.sh`. Давайте назовемо його `webcam.sh` і розмістимо в `/usr/sbin` як файл, який необхідно запускати з правами суперкористувача:

```
ffserver -f /root/ff.conf & ffmpeg -vcodec mjpeg -v verbose -r
15 -s 176x128 -f video4linux2 -i /dev/video0
http://localhost:81/webcam.ffm
```

Після того як файл `.sh` створений, і наведений вище код розміщений в ньому, необхідно зробити файл виконуваним, запустивши:

```
sudo chmod +x /usr/sbin/webcam.sh
```

### Запуск потокового відео через браузер

Оскільки файли сценарію і конфігурація створені, то можете почати приймати потік, запустивши:

```
/usr/sbin/webcam.sh
```

Після запуску сценарію, ви повинні отримати в результаті щось типу:

```
** 1 dup!2 fps= 5 q=2.6 size= 51136kB time=00:06:56.40
bitrate=1006.0kbts/s dup=359 drop=0
```

`ffmpeg` тепер генерує потік, до якого ви зможете отримати доступ з браузера за веб-адресою: `http://<IP-адреса сервера вашої камери>:81/webcam.avi`

## 5. ІНФОРМАЦІЙНА БЕЗПЕКА

### Особистий VPN, щоб безпечно заходити звідусіль в Інтернет



Крихітний, недорогий Raspberry Pi має дуже низьке енергоспоживання, що робить його відмінним завжди включеним сервером VPN<sup>70</sup> (Virtual Private

Network).

З VPN, ви отримаєте безпечний доступ до своєї домашньої мережі, коли знаходитесь в дорозі, і зможете використовувати його для безпечного перегляду веб-сторінок, коли під'єднані до мережі загального користування. Розглянемо два методи, як розгорнути свій власний VPN з Raspberry Pi. В даному підрозділі скористаємося Hamachi, а в наступних двох спробуємо створити VPN на OpenVPN.

Для введення в основи VPN, ознайомтеся з керівництвом, чому ви повинні почати використовувати VPN<sup>71</sup>, а потім поверніться сюди.

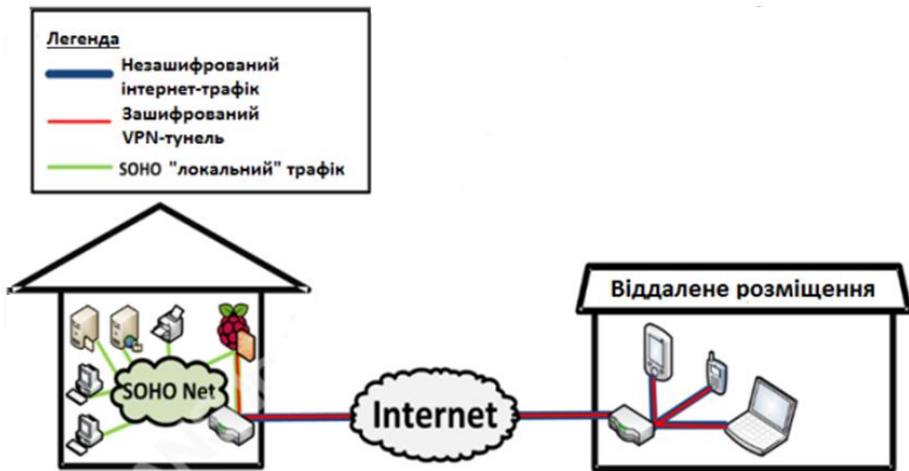
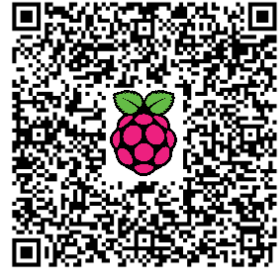


Рис.5.1 Архітектура VPN на Raspberry Pi

Налаштування Pi як сервера VPN і веб-проксі не вимагає нового або спеціального обладнання поза базового. Вам знадобляться:

<sup>70</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1792-use-the-raspberry-pi-for-personal-vpn-to-securely-from-anywhere-on-the-internet>

<sup>71</sup> <http://lifehacker.com/5900969/build-your-own-vpn-to-pimp-out-your-gaming-streaming-remote-access-and-oh-yeah-security>

- **Безкоштовний обліковий запис на LogMeIn<sup>72</sup>**, так як ми використовуватимемо LogMeIn Hamachi<sup>73</sup>, щоб створити свій VPN. Створіть безкоштовний обліковий запис, якщо ви ще не маєте його, перш ніж почати роботу. Hamachi створить VPN для вас, так що не доведеться возитися з переадресацією портів на маршрутизаторі, отримувати статичну IP-адресу, або намагатися обійти брандмауери.
- **Privoxy**: Ми об'єднуємо цей додаток з Hamachi, щоб запустити безпечний перегляд веб-сторінок всередині або поза мережею. Ви скачаєте його нижче на 5-му кроці.
- **Raspberry Pi**, звичайно: Дивіться наш перший розділ для початкового налаштування.
- **HDMI або композитний кабель**: Для підключення Pi до телевізора або монітора. Після того, як завершите цей проект, ви зможете відключити Pi від монітору і використовувати його як "безголовий" сервер, так як зможете підключатися до Pi над VPN.
- **8 Гб SD-карта 10 класу або краща і кардрідер**: Ваш комп'ютер може мати вбудований рідер SD-карти, а якщо ні, вам потрібний зовнішній рідер для встановлення Pi.
- **USB-клавіатура і миша**: Автор використовував старі проводові, але можна використовувати і бездротові (не Bluetooth) версії.
- **Мережевий кабель**: Для отримання оптимального результату в мережі, вам необхідне проводове підключення до маршрутизатора/модему (а не орієнтуватися на рішення Wi-Fi для Pi). Маючи широкосмугове підключення до Інтернету у себе вдома зробить проксі-сервер наскільки швидким, наскільки зможете.
- **Блок живлення з micro USB**: Знайдіть якісний зарядний пристрій, який може забезпечити принаймні 700 мА при 5 В. Більшість сучасних зарядних пристроїв для смартфонів будуть працювати, але перевірте їх специфікації на нижній частині рівня заряду, щоб переконатися в них. Крім SD-карти, блок живлення є одним з основних елементів з усунення неполадок, якщо у вас з'явилися проблеми.

Ми пройдемо через всі кроки створення особистого VPN з Hamachi і використання його з Privoxy для безпечного перегляду, але спочатку оновимо Linux і Pi, зокрема. Ми працюватимемо в командному рядку і будемо змінювати файли конфігурації, але насправді цей проект дуже легко зробити.

### Крок 1: Налаштування Raspberry Pi

Дотримуйтесь інструкцій з першого розділу, як почати роботу з Raspberry Pi для створення базової системи під управлінням Raspbian, стандартної операційної системи для Pi.

---

<sup>72</sup> <https://secure.logmein.com/UK/>

<sup>73</sup> <https://secure.logmein.com/products/hamachi/default.aspx>

Зауважимо, однак, що є один додатковий крок, який ви повинні зробити, це змінити розкладку клавіатури за замовчуванням, якщо ви не у Великобританії.

Причина в тому, що деякі з наших команд вимагають спеціальних символів (наприклад, #), який змінений в розкладці Великобританії.

Можете звернутися до першого розділу, де детально описані кілька методів зміни розкладки клавіатури. Нагадуємо, що можете змінити розкладку клавіатури у вікні конфігурації (рис.5.2).



Рис.5.2 Зміна розкладки клавіатури в `raspi-config`

Або, після того як ви налаштували Pi, введіть:

```
sudo dpkg-reconfigure keyboard-configuration
```

Далі дотримуйтесь інструкцій, щоб переключитися на розкладку своєї країни, а потім або перезапустіть Pi (за допомогою команди `sudo reboot`), або перезавантажите розкладку без перезапуску, ввівши:

```
invoke-rc.d keyboard-setup start
```

## Крок 2: Оновлення Raspberry Pi і встановлення Hamachi

Hamachi залежить від декількох пакетів, які можуть не бути присутніми в образі, тому ми збираємося спочатку оновити пакети і, сподіваюся, заощадити час на пошук несправностей поширених помилок.

По-перше, захопіть останні оновлення:

```
sudo apt-get update
```

Потім встановіть LSB (вимога для Hamachi):

```
sudo apt-get install -fix-missing lsb lsb-core
```

Будьте терплячими, поки йде оновлення, а потім можете завантажити останню версію Hamachi, створену для Linux:

```
sudo wget https://secure.logmein.com/labs/logmein-hamachi_2.1.0.86-1_armel.deb
```

Якщо ви отримуєте повідомлення про помилку завантаження Hamachi для Linux, зайдіть на сторінку завантаження на LogMeIn<sup>74</sup>, щоб перевірити номер версії (наприклад, 2.1.0.86-1) і змінити його в імені файлу, якщо це необхідно.

Потім встановіть Hamachi, ввівши:

```
sudo dpkg -i logmein-hamachi_2.1.0.86-1_armel.deb
```

### Крок 3: Налаштування Hamachi на Raspberry Pi

Останнім кроком для нашої установки VPN є отримання Hamachi, запущеним на Pi і на наших клієнтських ПК.

На Raspberry Pi виконайте наступну команду, щоб підключити Pi до свого облікового запису LogMeIn і створити нову мережу Hamachi:

```
sudo hamachi login
sudo hamachi attach [ВСТАВТЕ ТУТ EMAIL LOGMEIN.COM]
sudo hamachi set-nick [ВСТАВТЕ ЛОГІН ДЛЯ ВАШОГО RASPBERRY PI]
```

Тепер, на іншій машині, відкрийте LogMeIn<sup>75</sup> і перейдіть до свого розділу "My Networks" ("Мої мережі») в мережах. Ви побачите, що Pi (з наданим йому логіном) намагається підключитися і створити нову мережу. Надайте дозвіл Pi і запишіть ідентифікатор (9-ти значний номер) для цієї мережі. Поверніться до Raspberry Pi і введіть:

```
sudo hamachi do-join [ІДЕНТИФІКАТОР МЕРЕЖІ, ЯКИЙ ВИ ЗАПИСАЛИ]
```

Потім введіть пароль LogMeIn (якщо потрібно). Можливо, вам доведеться схвалити прохання приєднатися на LogMeIn від іншої машини. Як тільки ви це зробите, Pi буде частиною нової VPN, що обслуговується Hamachi. У LogMeIn.com, знайдіть віртуальну IP-адресу, присвоєну Pi, і запишіть її, бо вам вона пізніше знадобиться.

Щоб мати можливість використати SSH і дистанційно керувати RPi, запустіть сервер SSH (якщо він не був запущений автоматично):

```
sudo /etc/init.d/ssh start
```

---

<sup>74</sup> <https://secure.logmein.com/labs/#HamachiforLinux>

<sup>75</sup> <https://secure.logmein.com/UK/>

## Крок 4: Встановлення Hamachi на комп'ютерах

Майже готово! На комп'ютерах Windows, Mac або Linux, з яких ви також хочете підключитися до VPN, повинні встановити клієнт Hamachi зі сторінки завантаження<sup>76</sup>.

Після цього можете приєднатися до нової VPN (Network > Join) і працювати з Raspberry Pi через SSH або отримати доступ до файлів з мережі і т.д. (У Windows скористайтеся Putty, а в Mac/Linux запустіть термінал для SSH, використовуючи IP-адреси Raspberry Pi, присвоєні Hamachi в області сервера).

## Крок 5 (необов'язковий): Встановіть Privoxy на Pi і використовуйте його як веб-проксі свого комп'ютера

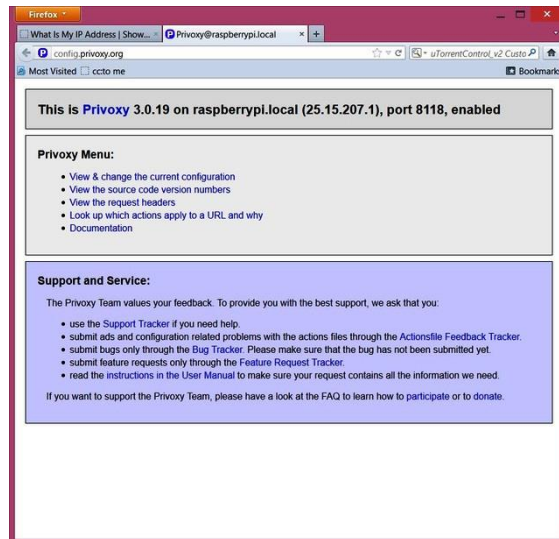


Рис.5.3 Проксі-сервер Privoxy на Rsspberry Pi

Окрім видачі віддалених команд на Pi і доступу до мережевих файлів, ви можете використовувати свій Pi як проксі-сервер. При підключенні Privoxy і Hamachi (рис.5.3) ви можете захистити і шифрувати сеанси браузера, коли підключаєтесь за допомогою відкритого Wi-Fi у вашому місцевому кафе, залишаючи свої дані в безпеці від цікавих очей або зловмисників.

Виконайте наступні кроки, щоб налаштувати Privoxy на Pi:

1. Встановіть Privoxy:

```
sudo apt-get install privoxy
```

<sup>76</sup> <https://secure.logmein.com/products/hamachi/download.aspx>

2. Запустіть Privoxy:

```
etc/init.d/privoxy start
```

3. Відкрийте файл конфігурації в текстовому редакторі:

```
sudo nano /etc/privoxy/config
```

4. Знайдіть такий рядок (це легко зробити в nano, якщо виконаєте пошук, натиснувши Ctrl+W): **listen-address localhost:8118**

5. Закоментуйте вказаний рядок, додавши # перед ним

6. Потім додайте під ним новий рядок: **listen-address [IP-адреса Вашого Pi, присвоєний Hamachi]:8118** (наприклад, 25.1.1.1:8118)

7. Збережіть файл конфігурації (Ctrl+X) і перезапустіть Privoxy:

```
sudo service privoxy restart
```

Тепер у вас є свій сервер Privoxy, налаштований на Pi, який працює по захищеному VPN через Hamachi. Все, що залишилося зробити, це встановити Privoxy як проксі-сервер на інші комп'ютери.

Для цього:

- У Google Chrome (рис.5.4): Перейдіть в розділ Налаштування > Показати розширені налаштування... > Змінити налаштування проксі-сервера... (У розділі Мережа)



Рис.5.4 Встановлення проксі-сервера Privoxy в Google Chrome

- У Firefox (рис.5.5): Відкрийте вікно Налаштування > Додаткова > Мережа > Кнопка Налаштувати... (в рядку з "Налаштуйте параметри з'єднання Firefox з Інтернетом" - "Configure how Firefox connects to the Internet")

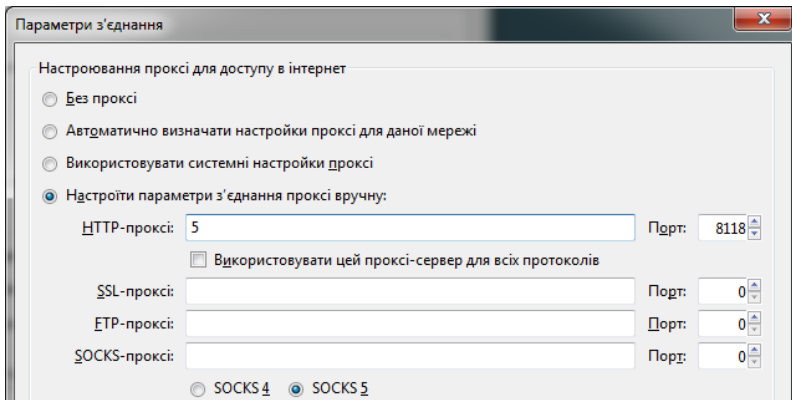


Рис.5.5 Встановлення проксі-сервера Privoxy в Firefox

Потім введіть IP-адресу Raspberry Pi, присвоєну Hamachi, як адресу проксі. Порт 8118.

Щоб перевірити, що проксі-сервер працює, перейдіть до <http://config.privoxy.org/> і ви повинні побачити повідомлення на кшталт "This is Privoxy on Windows [IP address], port 8118, enabled." Якщо не працює, то ви побачите повідомлення, що "Privoxy is not being used."

Крім того, коли ви використовуєте громадське Wi-Fi з'єднання, то відвідування [whatismyip.com](http://whatismyip.com) покаже іншу IP-адресу, коли ваш проксі включений, ніж коли він був вимкнений (IP-адреса має бути публічною IP-адресою вашої точки доступу).

Ось і все!

## Інші рішення з Raspberry Pi для VPN

Звичайно, є й інші способи вирішення проблеми з використанням Raspberry Pi. Наш улюблений інструмент (подібний до Hamachi) OpenVPN<sup>77</sup> є гарною альтернативою, але може вимагати трохи більше налаштування (а також статичного або динамічного DNS постачальника послуг). Ми його розглянемо в наступних підрозділах.

Також можете налаштувати PPTD VPN<sup>78</sup> (від Brad Wells), який має ту перевагу, що підтримує декілька пристроїв, але знову ж, це трохи складніше.

Для швидкого VPN і проксі-сервера, тим не менше, наведені вище кроки повинні запустити надійну мережу з Raspberry Pi в найкоротші терміни.

<sup>77</sup> <http://raspberrypi-hacks.com/29/turn-your-raspberry-into-an-openvpn-vpn-server/>

<sup>78</sup> <http://blog.wellsb.com/post/29412820494/raspberry-pi-vpn-server>

## VPN на Raspberry Pi: будуємо сервер



Безкоштовний  
незашифрований бездротовий  
доступ є майже скрізь, але ви не  
повинні перевіряти свій  
банківський рахунок через нього,  
бо не можете бути впевнені, що хтось не стежить.  
Рішення? Віртуальна приватна мережа або VPN<sup>79</sup>.

VPN розширює власну приватну мережу на  
громадські місця так, що навіть, якщо  
використовуєте Wi-Fi з'єднання Starbucks, ваш  
інтернет-браузер залишається зашифрованим і захищеним.

Є багато способів створити VPN з використанням безкоштовних і  
платних сервісів, але кожне рішення має свої плюси і мінуси, що  
визначається тим, як працює постачальник послуг VPN, а також від  
параметрів VPN, які надає.

Найпростіше і найдешевше рішення зберегти ваші дані в безпеці,  
це просто повністю утримуватися від публічного Wi-Fi. Але це звучить  
трохи екстремально для автора, коли є відносно простий і недорогий  
спосіб, щоб побудувати свій власний сервер VPN дома, і запустити його  
на крихітному, недорогому Raspberry Pi.

Raspberry Pi має розмір смартфона, але він запускає  
повнофункціональний сервер VPN. Це означає, що незалежно від того, де  
б користувач не був, він може підключити свій комп'ютер до домашньої  
мережі і отримати доступ до загальних файлів і медіа через захищене  
з'єднання. Особливо це буває в нагоді при подорожах та відрядженнях.

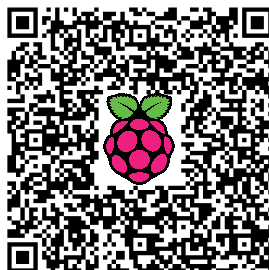
В цій першій частині, ви познайомитесь з тим, як встановити  
сервер VPN.

Йдіть далі і дізнайтеся, що незалежно від того, наскільки ви  
параноїк, той, хто придумав методи для створення віртуальних  
приватних мереж, був ним і поготів.

### Апаратні засоби

**Raspberry Pi Модель B:** Плюс все, що необхідне для нього -  
автор має на увазі звичайне джерело живлення і коробку, щоб покласти  
його. Коробка може допомогти запобігти випадковому короткому  
замиканню, яке може призвести до виходу пристрою з ладу і може бути  
досить простою, типу картонної коробки, яку ви зробите самі.

**SD-карта:** Автор пропонує на 8GB або більше, просто щоб бути  
впевненим, що у вас є простір. Як завжди, для всіх проектів Raspberry Pi,  
вона має мати встановлений Raspbian.



<sup>79</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1822-building-a-vpn-on-raspberry-pi-part-one-how-and-why-to-build-a-server>

**Кабель категорії 5e:** Для з'єднання порта Ethernet Pi з Ethernet-портом маршрутизатора.

## Програмне забезпечення

**Орен VPN:** Це послуга VPN з відкритим вихідним кодом, яку ми зараз встановлюватимемо.

## Передпроектні вимоги

- 1) Ви повинні налаштувати встановлену ОС Raspbian.
  - 2) Вам потрібна статична IP-адреса для Raspberry Pi у домашній мережі. Це залежить від моделі маршрутизатора, так що використовуйте вказівки виробника маршрутизатора. Якщо ви ще її не створили, то прочитайте розділ 2.
  - 3) Вам буде потрібний включений SSH. Ми будемо підключатися до Raspberry Pi через SSH - інструмент підключення, який дозволяє нам отримати доступ до Pi з іншого комп'ютера. Таким чином, нам не потрібно приєднувати Pi до монітора і клавіатури для цього проекту.
  - 4) Ви повинні будете направити порт 1194 (UDP трафік) на внутрішню IP-адресу свого Raspberry Pi, але, як це зробити, залежатиме від вашого маршрутизатора, так що перевірте інформацію виробника маршрутизатора. Якщо хочете використовувати інший порт або TCP, це нормально, лише переконайтеся, що змінили 1194 на правильне значення, і в будь-якому місці вказано: "UDP" на "TCP".
- Ви можете зробити зауваження, що ми будемо без деяких основних понять про RPi. Це тому, що побудова VPN на Raspberry Pi - не найкращий вибір першого проекту для більшості новачків.

## Коротке слово застереження

Автор скопіював фактичний код, який використав, щоб завершити цей проект, але він помітив, що копіювання і вставка коду зі статті в командному рядку часто призводить до помилок через пробіли і форматування. Якщо у вас виникли проблеми на будь-якій стадії проекту, першою порадою з усунення неполадок буде переписати команду вручну!

## Перші кроки

- 1) Завантажте та змініть пароль. Якщо ви все ще використовуєте ім'я користувача за замовчуванням (`pi`) і пароль (`raspberrypi`), то це робить іншу частину даного проекту з безпеки повністю безглуздою!

Відкрийте термінальне/PuTTY вікно і введіть:

```
sudo passwd
```

Змініть ім'я користувача і пароль на щось сильне і незабутнє, в іншому випадку навіщо будувати приватну мережу?

2) Тепер давайте збережемо і оновимо Raspberry Pi. Є дві команди, які треба ввести:

```
sudo apt-get update  
sudo apt-get upgrade
```

Це не займе багато часу і врятує нас від необхідності пошуку та усунення несправностей надалі.

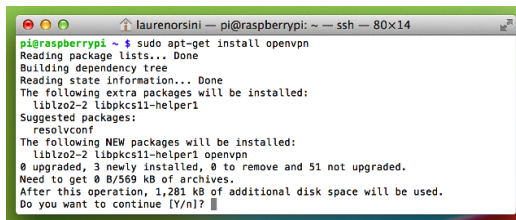


Рис.5.6 Встановлення OpenVPN

3) Тепер нам потрібне програмне забезпечення з відкритим вихідним кодом OpenVPN. Введіть (рис.5.6):

```
sudo apt-get install openvpn
```

Raspberry Pi запитає, чи ви впевнені, тому що буде використано небагато місця. Але так як ми готові, отримавши 8 Гб або більшу SD-карту, то для нас це абсолютно нормально.

### Генерація ключів

4) Ви ж не хочете, щоб хтось знайшов адресу вашого VPN-сервера і мав можливість підключитися. Так що наступне, що ми збираємося зробити, це ключ для адреси сервера. Це все одно, що зберігати двері до вашого будинку замкненими.

OpenVPN поставляється з Easy\_RSA, легким і простим пакетом для використання методу шифрування RSA. Розроблений в 1977 році, RSA був однією з перших використовуваних криптографічних систем, яка все ще використовується й сьогодні. Ключ шифрування є відкритим, в той час як ключ дешифрування є секретним.

З Easy\_RSA, ви запустите алгоритм, який поставляється разом з програмним забезпеченням для створення нового унікального ключа.

Отже, спочатку давайте дамо собі суперповноваження. Ви будете знати, що це працює, коли запрошення командного рядка перемикається з "pi@raspberrypi" на "root@raspberrypi":

```
sudo -s
```

Дана команда створює ще один екземпляр вікна, в якому ми працюємо, але з привілеями суперкористувача. Ми повинні зробити це тому, бо якщо цього не зробимо, то Raspberry Pi намагатиметься нам сказати, що у нас немає дозволу, щоб займатися виготовленням ключів.

Так що далі, ми набираємо:

```
cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0
/etc/openvpn/easy-rsa
```

Тут, "cp" виступає для копіювання, а "-r" означає рекурсив (зробити з меншими випадками теж). Це означає, що ми говоримо комп'ютеру: "Скопіюйте цей каталог і все в ньому".

Пробіл між /2,0 і /etc означає, що ми копіюємо перші адреси (файл прикладу) в другу папку, яку ви назвали OpenVPN, щоб знайти ключі:

```
cd /etc/openvpn/easy-rsa
```

5) Далі, ми виконали команду cd, або змінили каталог, на місце, в яке ми тільки що перемістили файл Easy\_RSA. Опинившись там, ми повинні відкрити файл /etc/openvpn/easy-rsa/vars для редагування. Ми могли б це зробити, написавши nano /etc/openvpn/easy-rsa/vars, але так як ми знаходимося в папці, то в ній є ярлик:

```
nano vars
```

Тепер знайдіть і змініть змінну **EASY\_RSA** на:

```
export EASY_RSA="/etc/openvpn/easy-rsa"
```

В автора це було в 13 рядку (рис.5.7).

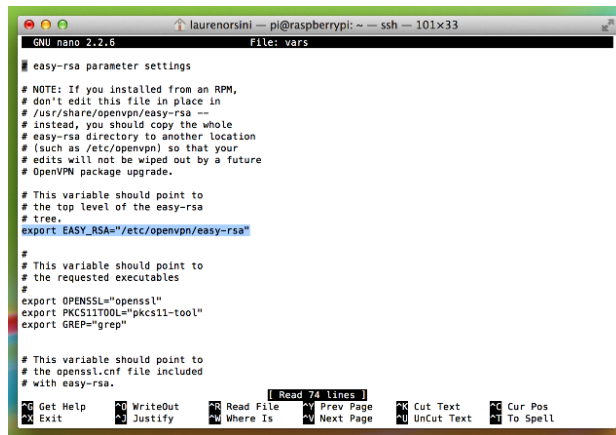


Рис.5.7 Внесення змін в **EASY\_RSA**

Навіщо робити цю заміну? В принципі, ви відповідаєте на питання комп'ютера, "Де ви хочете розмістити файл?" Ми хочемо експортувати його в ту ж папку, в якій ми будемо тримати наші ключі, в даному випадку, верхній рівень дерева файлів easy-rsa.

Тут одна додаткова річ, яку ви можете зробити в `vars`: якщо ви параноїк відносно читання електронної пошти, - змініть метод шифрування з 1024-біт на 2048-біт. Документ так буквально і каже: "збільшіть це число до 2048, якщо ви параноїк».

Але так як цей метод робить шлях генерування ключів довшим, ми його тут не робимо. Ми залишимо це, як виглядає текст на зображенні:

```
export KEY_SIZE=1024
```

Натисніть Ctrl+X, щоб зберегти зміни і вийти з редактора nano.

### Отримання криптографіки

6) Прийшов час будувати сертифікати CA Certificate і Root CA.

У криптографії, Центр сертифікації (CA) є організацією, яка видає цифрові сертифікати. Цифровий сертифікат засвідчує право власності відкритого ключа.

Ви, напевно, використовуєте це весь час, і навіть не знаєте про це. Наприклад, коли входите на свій банківський рахунок, то бачите HTTPS перед адресою. Якщо натискаєте на замок, то бачите, що деяка компанія перевіряє законність сайту банку, тому знаєте, що сайт не є шахрайським. (Звичайно, останнім часом помилка Heartbleed показала, що HTTPS - не є мірою безпеки, як ми всі звикли думати.)

У разі Raspberry Pi, автор веде себе як власник свого центру сертифікації та підписує ключі OpenVPN сам, замість того, щоб довіряти це сторонній компанії.

```
cd /etc/openvpn/easy-rsa
```

Тепер, коли ми змінили каталоги, введіть кожний з цих рядків один за одним:

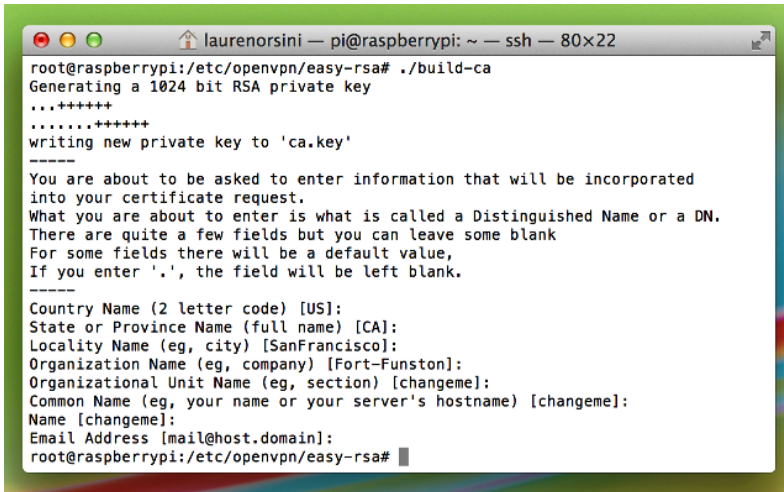
`source ./vars` → Це «джерела» або завантаження документу `vars`, який відредагований нами раніше.

`./clean-all` → Це дозволить видалити всі попередні ключі, якщо вони є. Якщо у вас є ключі і ви не хочете видаляти в цій папці (наприклад, ви робите цей урок вдруге), пропустіть дану команду.

`./build-ca` → Цей останній рядок будує ваш центр сертифікації.

Після третьої команди, Raspberry Pi «вистрелить» для вас купою додаткових полів, щоб заповнити, якщо хочете: назва країни, назва штату або провінції, назва місцевості, назва організації, підрозділ, загальноприйнята назва, ім'я і e-mail адреса.

Якщо не бажаєте заповнювати ці поля, просто натисніть "Enter" в кожному екземплярі, щоб дати Pi заповнити значення за замовчуванням. Скріншот нижче (рис.5.8) показує, як це виглядає:



```

root@raspberrypi:/etc/openssl/easy-rsa# ./build-ca
Generating a 1024 bit RSA private key
.....
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
State or Province Name (full name) [CA]:
Locality Name (eg, city) [SanFrancisco]:
Organization Name (eg, company) [Fort-Funston]:
Organizational Unit Name (eg, section) [changeme]:
Common Name (eg, your name or your server's hostname) [changeme]:
Name [changeme]:
Email Address [mail@host.domain]:
root@raspberrypi:/etc/openssl/easy-rsa#

```

Рис.5.8 Генерування ключів

Тепер ви можете назвати сервер. Автор творчо назвав свій "Server". Називайте як хочете, але, головне, не забудьте назву:

`./build-key-server [Server_Name]`

Ще раз, вам збираються виплунути деякі необов'язкові поля. натисніть "enter" або введіть, що хочете, але зверніть увагу на ці області:

- **Common Name** (Загальне ім'я) МАЄ бути ім'ям сервера, яке ви обрали. Це за замовчуванням.
- **A challenge password?** (Задати пароль?) **ПОВИННО** бути порожнім.
- **Sign the certificate?** [y/n] (Підписати сертифікат?) Очевидно, що необхідно ввести "y". Ви отримаєте повідомлення про те, що сертифікат буде придатний для сертифікації більше 3650 днів. Так в основному, якщо ви використовуватимете свій VPN досить довго, то вам доведеться продовжити цей процес знову на 10 років.
- **1 out of 1 certificate requests certified, commit?** [y/n] (1 з 1 сертифікатів отримав запит на сертифікацію, здійснити?) Очевидно, ввести "y" (рис.5.9).

7) Це установка на стороні сервера. Тепер прийшов час, щоб побудувати ключі для кожного користувача, або "клієнта". В автора є п'ять ключів дома, по одному на кожному комп'ютері, планшетнику, і стільниковому телефоні в будинку. Можливо ви лінуєтеся і створите тільки один клієнтський ключ для всіх них, але в цьому випадку тільки один пристрій зможе отримати доступ до VPN одночасно.

```
./build-key-pass UserName
```

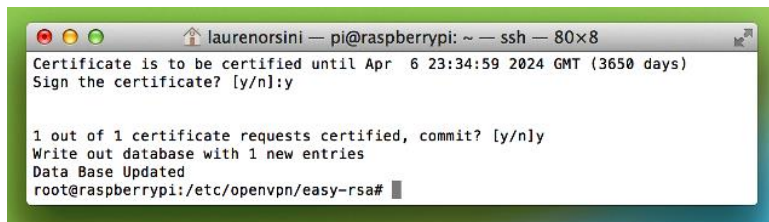


Рис.5.9 Заповнення полів сертифікату

Автор вирішив, що найпростіше зробити імена користувачів client1, client2, client3... (рис.5.10):

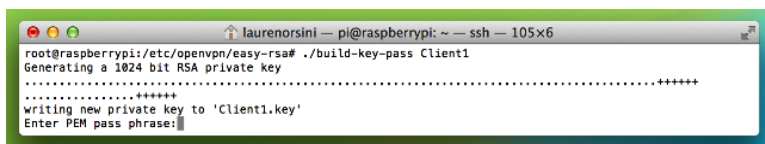


Рис.5.10 Додавання клієнтів

І після цього, більше підказок!

- **Enter PEM pass phrase** (Введіть PEM паролську фразу). Це зробить пароль, який ви повинні пам'ятати! Він просить вас ввести це двічі, так що немає ніякої небезпеки зруйнувати його.
- **A challenge password?** (Задати пароль?) **ПОВИННО** бути порожнім.
- **Sign the certificate?** [y/n] (Підписати сертифікат). Підписання засвідчує його ще на 10 років.

```
cd keys
```

```
openssl rsa -in Client1.key -des3 -out Client1.3des.key
```

Важливий висновок цього рядка тексту в тому, що ми використовуємо шифрування DES3, в якому комплексний EncryptionAlgorithm застосовуваний три рази для кожного блоку даних, щоб утримати хакерів від злому його методом грубою силою. OpenSSL виступає за реалізацію з відкритим вихідним кодом від Secure Socket Layer, стандартного методу створення захищеного з'єднання. Ви повинні виконати цей крок для кожного клієнта, якого створили.

Дехто стверджує, що цей крок не є необхідним, і що ви можете просто пропустити цей рядок. Але якщо запускаєте клієнтів OpenVPN Connect на Android або iOS, то це повинно бути зроблено. В іншому випадку, поточні версії можуть мати труднощі розбору ключів, які вами згенеровані.

Enter pass phrase for Client1.key

Чесно кажучи, автор просто використав ту ж парольний фразу, яку ввів раніше. А потім ще двічі, як показано (рис.5.11).

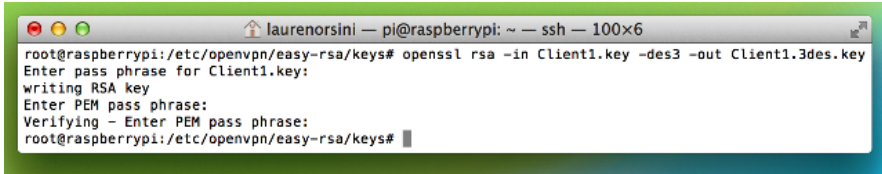


Рис.5.11 Введення парольної фрази

Тепер, коли ми створили сертифікат сервера і (принаймні один) сертифікат клієнта, введіть наступну команду:

```

cd /etc/openvpn/easy-rsa/
або
cd ..

```

У кожному разі, комп'ютер поверне Вас на одну директорію, до `/easy-rsa/`.

8) Тепер давайте згенеруємо обмін ключами Діффі-Хеллмана. Це центральний код, який робить ваш сервер VPN довіреним для обміну, що дозволяє двом об'єктам без попереднього знання один одного поширення секретних ключів поверх загальнодоступного сервера. Подібно RSA, це одна з найбільш ранніх криптосистем (рис.5.12):

```
./build-dh
```

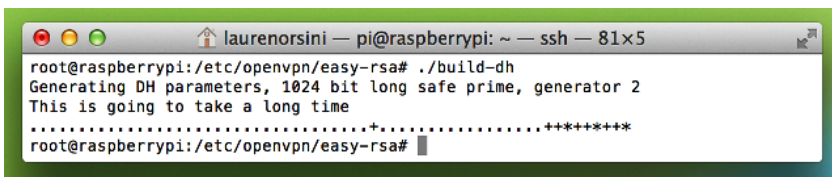


Рис.5.12 Генерування ключів для обміну

Це може зайняти деякий час: довше, якщо ви зупинилися на 2048-бітовому шифруванні. Тут немає ніякого дієвого способу передбачити, як довго це займе, тому що використовуються випадкові числа і пошук деяких конкретних співвідношень. Справді, коли автор робив даний урок, це зайняло лише 5 хвилин з 1024-бітовим шифруванням.

9) Нарешті, ми збираємося реалізувати вбудований в OpenVPN захист від атаки відмови в обслуговуванні (DoS). Можливо, ви вже знаєте, що атака DoS є успішною, коли хакер дізнається адресу вашого

сервера, і генерує таку велику кількість спроб, що ваш сервер виходить з ладу.

OpenVPN має спосіб, щоб запобігти такого роду нападу до його виникнення, який запускається шляхом створення статичного Pre-Shared ключа коду аутентифікації повідомлення на основі хеша (hash-based message authentication code - HMAC). З цього місця, сервер не буде навіть розглядати ідею аутентифікації запиту на доступ, якщо, в першу чергу, не виявляє цей статичний ключ. Таким чином, хакер не може просто спамити сервер з випадковими неодноразовими запитами.

Генеруємо статичний ключ HMAC за допомогою наступного рядка:

```
openvpn --genkey --secret keys/ta.key
```

### Збираємо все разом

10) Ми згенерували ключі і центр сертифікації, щоб підписати їх. Те, що ми все ще пропустили, це налаштування, щоб сказати OpenVPN, як ми хочемо цей сервер налаштувати.

Програма OpenVPN вже запущена. Проблема в тому, що вона не знає, які ключі використовувати, звідки ви збираєтеся підключатися, які зв'язки ви будете, або які IP-адреси та порт для використання.

Так як ми використовуємо Linux на Raspberry Pi, то немає графічного інтерфейсу користувача (GUI), щоб розповісти OpenVPN, що він повинен знати. Ось чому ми повинні фактично створити файл `.conf` (configuration - конфігурація) в nano редакторі повністю в командному рядку.

Ми починаємо адреси з `/etc/openvpn`, тому, що в кінцевому підсумку він буде в папці `openvpn`. Але зараз цей файл повністю порожній. Заповніть його з `github`<sup>80</sup>. Автор прокоментував заголовними буквами, де обов'язково необхідно змінити значення та назви вашими власними IP-адресами/іменами. Натисніть `Ctrl+X`, щоб зберегти зміни.

11) Давайте також відредагуємо інший файл конфігурації. За замовчуванням, Raspbian не надсилає інтернет-трафіку. Нам потрібно відредагувати ще один файл, щоб дозволити Pi направляти інтернет-трафік через нашу нову мережу.

```
nano /etc/sysctl.conf
```

Зверху він говорить, "Розкоментуйте наступний рядок, щоб **включити форвард пакетів для IPv4**". Автор виділив цю частину файлу в скріншоті нижче (рис.5.13):

---

<sup>80</sup> <https://gist.github.com/laurenorsini/9925434>

```

GNU nano 2.2.6      laurenorsini — pi@raspberrypi: ~ — ssh — 126x36
File: /etc/sysctl.conf

#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables
# See sysctl.conf(5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
kernel.printk = 3 4 1 3

#####
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration

Get Help      WriteOut      Read File     Prev Page     Cut Text      Cur Pos
Exit          Justify       Where Is      Next Page     UnCut Text    To Spell

```

Рис.5.13 Редагування файлу конфігурації `sysctl.conf`

Щоб розкоментувати рядок, видаліть `#` безпосередньо перед ним. Це налаштування конфігурації, щоб він знав про направлення IPv4. Тепер, коли цей рядок розкоментували, то Pi має дозвіл діяти як ретранслятор в мережі Інтернет, а не бути лише приймачем, тобто працювати на передачу і прийом пакетів.

Натисніть `Ctrl+X`, щоб зберегти зміни. Застосуйте ці зміни, ввівши таку команду:

```
sysctl -p
```

Команда `sysctl` "налаштовує параметри ядра під час виконання", а `-p` говорить, щоб файл перезавантажився зі змінами, які тільки що зробили.

12) Ми зараз зробили функціонуючий сервер, який може отримати доступ до Інтернету. Але ми не можемо використовувати його ще й тому, що Raspbian має вбудований міжмережевий екран, який буде блокувати вхідні з'єднання.

Raspbian має брандмауер для захисту вашого Raspberry Pi від невідомих і несподіваних джерел Інтернету. Ми, як і раніше, хочемо мати брандмауер, щоб захистити себе від більшості вхідного і вихідного мережевого трафіку, але нам потрібно «віткнути» OpenVPN-подібний виріз в брандмауер.

Крім того, брандмауер скидає налаштування Raspbian в «за замовчуванням» при перезавантаженні Pi. Ми хочемо переконатися, що він пам'ятає, що з'єднання OpenVPN завжди дозволяється. Для цього ми збираємося створити простий сценарій, який працює при завантаженні:

```
nano /etc/firewall-openvpn-rules.sh
```

В даний час це порожній виконуваний файл оболонки. Залийте його так:

```
#!/bin/sh
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j SNAT --
to-source 192.168.XX.X
```

Не забудьте змінити IP-адресу за замовчуванням на IP-адресу вашого Pi!

Давайте розберемо це нижче: 10.8.0.0 є адресою за замовчуванням для Raspberry Pi для клієнтів, підключених до VPN. "eth0" означає порт локальних мереж. Перемкніть його на "wlan0", якщо знаходитесь на бездротовому з'єднанні, яке не рекомендується. Натисніть Ctrl+X, щоб зберегти зміни.

Як міра безпеки, створені файли не можуть виконуватися за замовчуванням, так що нам потрібно змінити права доступу і володіння /etc/firewall-openvpn-rules.sh. Спочатку ми змінимо параметр на 700 (власник може читати, писати і виконувати). Потім ми змінимо власника на root, в якому "root" є стандартним ім'ям в Linux для суперкористувача:

```
chmod 700 /etc/firewall-Openvpn-rules.sh
chown root /etc/firewall-Openvpn-rules.sh
```

13) Ми створили сценарій, який «пробиває» OpenVPN-подібний виріз в брандмауері. Тепер нам просто потрібно встановити його в код інтерфейсу введення, щоб він працював при завантаженні системи:

```
nano /etc/network/interfaces
```

Знайдіть рядок, який має "iface eth0 inet dhcp". Ми хочемо додати рядок під ним з відступом. Так щоб ці два рядки, існуючий та новий, виглядали так, коли ви закінчите:

```
iface eth0 inet dhcp
pre-up /etc/firewall-openvpn-rules.sh
```

Натисніть Ctrl+X, щоб зберегти зміни, як ви завжди робите з nano. Нарешті, перезавантажте Pi:

```
sudo reboot
```

Вітаємо! Це сервер! Знову ж, це не зовсім добре, якщо у вас немає комп'ютера клієнта, щоб з'єднатися з ним, тому запам'ятайте імена клієнтів і ключі, отримані на сьомому етапі, а потім перейдіть до другої частини, щоб дізнатися, як створити зашифровану сторону клієнта.

## VPN на Raspberry Pi: створюємо сторону клієнта



Оскільки недоліки безпеки дозволяють подальші атаки на наше інтернет-життя, то відчуваємо все більше і більше, що варто мати у себе безпечний

сервер<sup>81</sup>.

Таким чином, ви зможете вільно писати електронні листи і передавати дані, не турбуючись про те, що щось або хтось може перехопити його на шляху від вашого комп'ютера до Інтернету.

Якщо ви слідували інструкціям з першої частини, то у вас є повнофункціональний сервер VPN на своєму Raspberry Pi. Ви можете використовувати його, щоб безпечно підключатися до домашньої мережі звідусіль, де є незашифроване бездротове з'єднання. Ви також можете отримати доступ до загальних файлів і мультимедійної інформації, які зберігаєте у своїй домашній мережі.

Тільки ви, поки що, не можете отримати доступ до цих файлів. Ми створили ключі для клієнтів (ком'ютерів і пристроїв), щоб використовувати, але ми не сказали клієнтам, де знайти сервер, як зв'язатися з ним, а також, який ключ використовувати.

Якщо ви пам'ятаєте, ми створили кілька різних клієнтських ключів для кожного з пристроїв, яким хочемо надати VPN-доступ. Ми назвали їх Client1, Client2 і Client3.

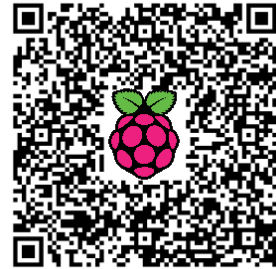
Було б багато неприємностей, якби треба було створювати новий файл конфігурації для кожного клієнта з нуля, тому ми використаємо оригінальний сценарій, написаний Еріком Jodoін з інституту SANS. Замість створення файла для кожного клієнта нами власноруч, цей сценарій зробить це за нас.

### Рухайтесь за сценарієм

Сценарій надасть доступ до наших налаштувань за замовчуванням, щоб створити файли для кожного клієнта. Перше, що нам потрібно зробити, це створити порожній текстовий файл, в якому ці налаштування за замовчуванням можуть бути прочитані.

```
nano /etc/openvpn/easy-rsa/keys/Default.txt
```

Заповніть порожній текстовий файл з наступним:



<sup>81</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1823-building-a-vpn-on-raspberry-pi-part-two-creating-an-encrypted-client-side>

```
Client
dev tun
proto udp
remote <Тут Ваша публічна IP-адреса> 1194
resolv-retry infinite
nobind
persist-key
persist-tun
mute-replay-warnings
ns-cert-type server
key-direction 1
cipher AES-128-CBC
comp-lzo
verb 1
mute 20
```

Він повинен виглядати як на скріншоті нижче (рис.5.14), за винятком того, що він повинен показувати вашу публічну IP-адресу. Ви бачите, що автор видалив свою власну публічну IP-адресу, тому що це приватна інформація, якою ви не повинні ділитися з навколишніми. З іншого боку, локальні статичні IP-адреси дуже схожі для всіх. Як правило, вони починаються з "192.168".

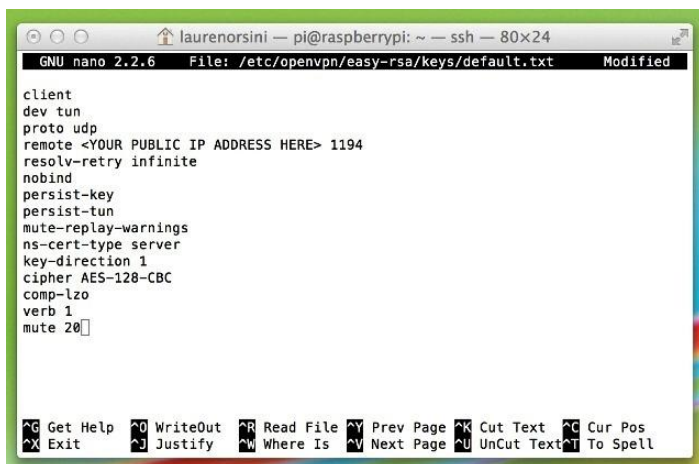


Рис.5.14 Текст сценарію

Тепер, якщо у вас немає статичної публічної IP-адреси, необхідно скористатися службою системи динамічних доменних імен (DDNS), щоб отримати для себе доменне ім'я, яке замінить IP-адресу. Автор рекомендує використовувати безкоштовний сервіс DNS Dynamic<sup>82</sup>, який дозволяє вибрати ім'я за вашим вибором. Тоді на Pi вам треба запустити

<sup>82</sup> <https://www.dnshdynamic.org/>

DDClient, щоб автоматично оновити DDNS-реєстр. Повне керівництво про те, як це зробити, за посиланням<sup>83</sup> знизу сторінки.

Як завжди, натисніть Ctrl+X для збереження і виходу з редактора nano.

Далі, нам необхідно створити сам файл сценарію. Сценарій буде запускатися з файлу оболонки, який є виконуваним сценарієм, що зазвичай автоматизує завдання в Linux, в тому числі, в цьому випадку.

```
nano /etc/openvpn/easy-rsa/keys/MakeOPVN.sh
```

Ось сценарій<sup>84</sup>, який написав Jodoін. Скопіюйте та вставте його в свій порожній файл оболонки.

Вам все ще необхідно дати дозвіл цьому сценарію на запуск. По-перше, перейдіть в папку, де він знаходиться:

```
cd /etc/openvpn/easy-rsa/keys/
```

А потім дайте йому повноваження root. Як ви пам'ятаєте з першої частини, дозволи в Linux регулюються різними тризначними номерами. Сімсот означає "власник може читати, писати і виконувати":

```
chmod 700 MakeOPVN.sh
```

Нарешті, виконайте сценарій з:

```
./MakeOPVN.sh
```

Коли сценарій запущений, він попросить вас ввести імена існуючих клієнтів, для яких ви раніше генерували ключі СА. Приклад: "Client1". Переконайтеся, що назвали тільки клієнтів, які вже існують.

Якщо все піде добре, то ви повинні побачити такий рядок:

```
Done! Client1.opvn Successfully Created.
```

Повторіть цей крок для кожного існуючого клієнта.

Останнє, що потрібно зробити, це підключитися до Raspberry Pi так, щоб ви могли завантажити з нього файли. Ви повинні використовувати клієнта SCP (Secure Copy Protocol) для того, щоб це зробити. Для Windows автор рекомендує WinSCP<sup>85</sup>, а для Mac скористатися Fugu<sup>86</sup>.

<sup>83</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1759-how-to-give-your-raspberry-pi-static-ip-address>

<sup>84</sup> <https://gist.github.com/laurenorsini/10013430/revisions>

<sup>85</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1761-secure-file-sharing-with-raspberry-pi>

<sup>86</sup> <http://fugu.en.softonic.com/mac>

Зверніть увагу: якщо ви не можете отримати дозвіл на підключення до свого клієнта SCP, то повинні будете надати собі дозвіл на читання/запис для папки. Поверніться на Raspberry Pi, введіть:

```
chmod 777 -R /etc/openvpn
```

Будьте впевнені, що скасували це, коли закінчите копіювання файлів, так щоб інші не змогли ним скористатися! Поверніть дозвіл назад в 600, коли закінчите, так щоб тільки користувач Pi зміг читати/записувати файли:

```
chmod 600 -R /etc/openvpn
```

Поставте його у свого клієнта, і все готово.

### Робота з клієнтським програмним забезпеченням

Добре, важка частина закінчена. Звідси, ми повинні ввести сценарії, які згенерували раніше, в графічному інтерфейсі користувача. Для вашого ПК, Android, або мобільного пристрою iOS, Ви можете завантажити OpenVPN Connect<sup>87</sup>. Існує не одна програма для вашого Мас-комп'ютера, тому автор спробував як Tunnelblick<sup>88</sup>, так і Viscosity<sup>89</sup>.

Tunnelblick безкоштовний, в той час як Viscosity коштує \$9 після безкоштовного 30-денного тріалу. У будь-якому разі, давайте розглянемо налаштування комп'ютера Мас як клієнта.

Файл, який створив автор за допомогою даного сценарію, він назвав Client5.opvn.

Завантажте версію Tunnelblick (рис.5.15), яка працює для вашої версії OS X. Автор використовує Mavericks, тому він завантажив бета-версію.



Рис.5.15 Версія Tunnelblick

<sup>87</sup> <https://play.google.com/store/apps/details?id=net.openvpn.openvpn&hl=ru>

<sup>88</sup> <https://code.google.com/p/tunnelblick/>

<sup>89</sup> <https://www.sparklabs.com/viscosity/>

Тепер буде запитано, чи у вас вже є файл, який хочете використовувати (рис.5.16). Автор ввів свій файл Client5.ovpn.



Рис.5.16 Запит про наявність файла конфігурації

Тоді буде запитано, у якому форматі ваш файл конфігурації: `.ovpn` чи `.tblk`. Якщо виберете `.ovpn`, то він проведе вас через зміну типу файлу для власного типу Tunnelblick. Автор зробив це шляхом передачі Client5.ovpn в надану папку Tunnelblick, а потім змінив ім'я папки на Client5.tblk.

Тепер все готово для підключення. Натисніть на значок Tunnelblick в правому верхньому куті екрану і виберіть Client5 (рис.5.17).

У вас запросять кодову фразу. Це та ж сама парольна фраза, яку ми згенерували в попередньому підрозділі, коли генерували ключі для кожного клієнта.

Якщо Ви правильно ввели пароль, це буде виглядати наступним чином (рис.5.18)!

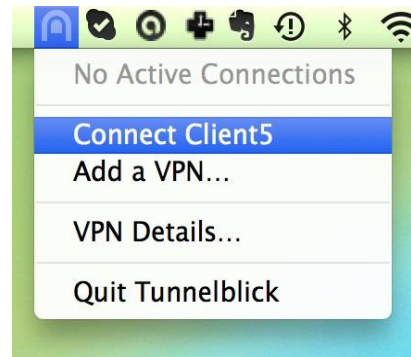


Рис.5.17 Вибір клієнта

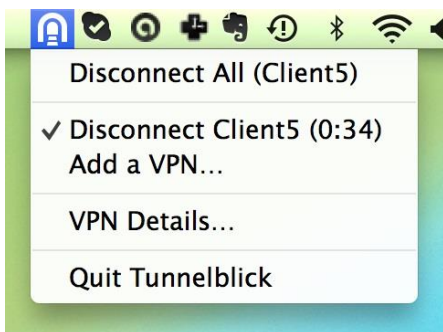


Рис.5.18 Клієнт підключений

Спробуйте нове підключення в кафе, в місцевій бібліотеці, всюди, де є незашифрований Wi-Fi. Ви все ще можете використовувати відкриті з'єднання, але через VPN ваші дані будуть зовсім не відкриті.

### Onion Pi: портативний Tor-проксі



Портативний проксі-сервер Onion Pi<sup>90</sup> Ви можете брати з собою в гості або в подорож і через нього виходити в Інтернет з використанням мережі анонімайзерів Tor.

Даний проект використовує Raspberry Pi, USB адаптер WiFi і кабель Ethernet (рис.5.19) для створення конфіденційності на невеликому, малопотужному і портативному Pi.

Скористуватися ним дуже легко. По-перше, підключіть кабель Ethernet до будь-якого інтернет-провайдера у своєму будинку, на роботі, в готелі чи на конференції/події. Потім подайте живлення на Pi через кабель мікро USB від ноутбука або адаптера мережі живлення.

Pi завантажиться і створить нову безпечну бездротову точку доступу, так званий Onion Pi. Підключення до цієї точки доступу буде автоматично створювати маршрут з будь-якого веб-браузер комп'ютера через анонімну мережу Tor.

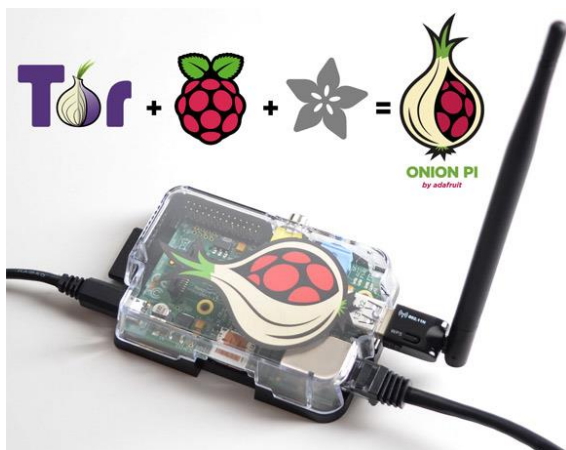
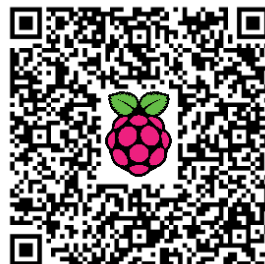


Рис.5.19 Портативний Tor-проксі

#### Для чого це годиться?

Якщо хочете анонімно переглядати з нетбука, планшета, телефону або іншого мобільного чи консольного пристрою, який не може працювати через Tor і не має підключення до Ethernet. Якщо не хочете або не можете встановити Tor на своєму робочому ноутбуці чи

<sup>90</sup> <http://isearch.kiev.ua/uk/news/programs/tools-sec/1717-onion-pi-portable-tor-proxy-based-raspberry-pi>

запозиченому комп'ютері. Якщо у вас є гість або друг, який хоче використати Тор, але не має можливості або часу для запуску Тор на своєму комп'ютері, - цей подарунок дозволить зробити перший крок набагато легше.

### **Що таке Тор?**

Тор є «цибулиною» службою маршрутизації - кожен інтернет-пакет проходить через 3 шари проміжних серверів, перш ніж дійде до місця призначення. Це робить його набагато важчим для сервера, до якого ви звертаєтеся (або для того, хто стежить за використанням вами Інтернету), щоб з'ясувати, хто ви, де ви і звідки прийшли. Це відмінний спосіб дозволити людям, яким блокується доступ до певних веб-сайтів, обійти ці заборони.

### **За інформацією з сайту Тор:**

Журналісти використовують Тор, щоб вільно і безпечно спілкуватися з дисидентами і викривачами. Неурядові організації використовують Тор, щоб їхні працівники при підключенні до своїх домашніх сайтів, коли вони знаходяться в чужій країні, не оголошували публічно, що працюють з даною організацією.

Такі групи, як Indymedia рекомендує Тор для захисту приватності та безпеки своїх членів. Групи активістів, наприклад, Electronic Frontier Foundation рекомендує Тор як механізм забезпечення громадянських свобод в Інтернеті.

Корпорації використовують Тор як безпечний спосіб проведення конкурентного аналізу та для захисту від спостерігачів важливих шляхів передачі. Вони також використовують його для заміни традиційних VPN, який не приховує від зловмисника точну кількість і тимчасові проміжки сеансу зв'язку. Де живуть співробітники, які пізно працюють? Де живуть співробітники консалтингових веб-сайтів для пошуку роботи? Які дослідники спілкуються з юристами по патентах компанії?

Філія ВМС США використовує Тор для збору розвідданих з відкритих джерел, а один підрозділ останнім часом використовував Тор під час розгортання на Близькому Сході. Правоохоронні органи використовують Тор для відвідування веб сайтів, не залишаючи урядових IP-адрес у своїх логах, а також для безпеки під час операцій.

Даний проект є відмінним способом, щоб зробити щось приємне і корисне з Raspberry Pi, але він знаходиться в стадії розробки. Не можна гарантувати, що це на 100% анонімно і безпечно! Будьте розумним і параноїдальним при своєму використанні Тор.

### **Що Вам знадобиться**

Вам буде потрібно кілька речей, щоб запустити цей проект:

- Raspberry Pi модель B – необхідний Ethernet
- Кабель Ethernet
- Адаптер Wi-Fi - не всі бездротові адаптери працюють з Raspberry Pi, тому перевірте, що ваш працює!
- SD-карта (4 Гб або більше) з встановленою Raspbian. Ви можете або скопіювати образ Raspbian на неї або купити готову Raspbian-карту
- Живлення для свого Pi і кабель мікро USB
- Кабель USB-консолі (опціонально) - це трохи полегшує налагодження системи
- Корпус для Pi (опціонально)
- Рідер SD або MicroSD карти (опціонально)

### Підготовка

Припускаємо, що у вас є встановлений Pi і налаштований як Wi-Fi точка доступу.

Будь ласка, виконайте наступні кроки проекту:

- Запишіть ОС Wheezy на карту SD
- Завантажте і налаштуйте Pi<sup>91</sup>

Не забудьте змінити пароль за замовчуванням для облікового запису «pi».

Переконайтеся, що розширили файлову систему на весь диск, бо може не залишитися доступного місця (рис.5.20):

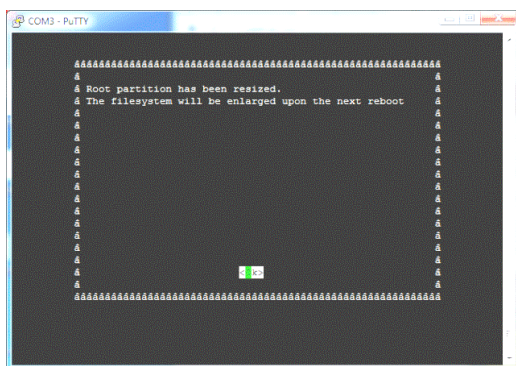


Рис.5.20 Розширення файлової системи з raspi-config

- Налагодьте та перевірте з'єднання Ethernet і Wi-Fi<sup>92</sup>

Переконайтеся, що можете виконати `ping` з RPi і що ваш Wi-Fi адаптер визнається та відображається, як `wlan0` при запуску команди `ifconfig -a`.

<sup>91</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1753-how-to-get-started-with-raspberry-pi>

<sup>92</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1750-how-to-set-up-wi-fi-on-your-raspberry-pi-via-the-command-line>

- Підключіться з консольного кабелю USB (опціонально)

Це зручно для налагодження, особливо при підключенні до точки доступу, організованої на Pi.

Коли все зроблено, то ви повинні мати Pi, який завантажив Raspbian, і ви можете підключитися до нього з допомогою консольного кабелю USB та увійти в Pi через інтерфейс командного рядка.

- Потім, відповідно до розділу про створення на Pi точки доступу<sup>93</sup>, налаштуйте Pi як маршрутизатор з точкою доступу Wi-Fi.

Тепер ви повинні бути в змозі підключитися до Pi, як точки доступу Wi-Fi (рис.5.21), і через неї вийти в Інтернет.

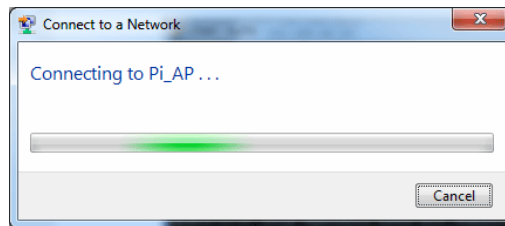


Рис.5.21 Перевірка підключення до Pi, як точки доступу Wi-Fi

Можна зробити подальші кроки, використовуючи SSH через порт Ethernet або за допомогою консольного кабелю.

При використанні консольного кабелю, хоча схема нижче показує підключення живлення до Pi через консольний кабель USB (червоний провід), пропонуємо не під'єднувати червоний провід, а замість нього використати живлення від адаптера електромережі. Залиште чорний, білий та зелений кабелі підключеними, як показано (рис.5.22):

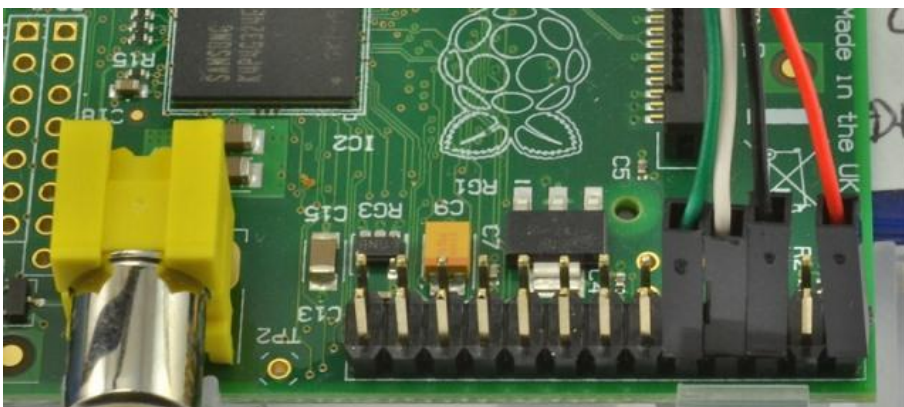


Рис.5.22 Під'єднання консольного кабелю

<sup>93</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1801-how-to-make-raspberry-pi-wi-fi-router>

### Встановлення Tor

Увійдіть у свій Pi через Ethernet або консольний кабель і наберіть команду:

```
sudo apt-get install tor
```

Відредагуйте конфігураційний файл Tor, запустивши редактор nano:

```
sudo nano /etc/tor/torrc
```

Тепер копіюйте і вставте текст на початку файла, прямо під FAQ повідомленням, наведений нижче код (рис.5.23):

```
Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.42.1
DNSPort 53
DNSListenAddress 192.168.42.1
```

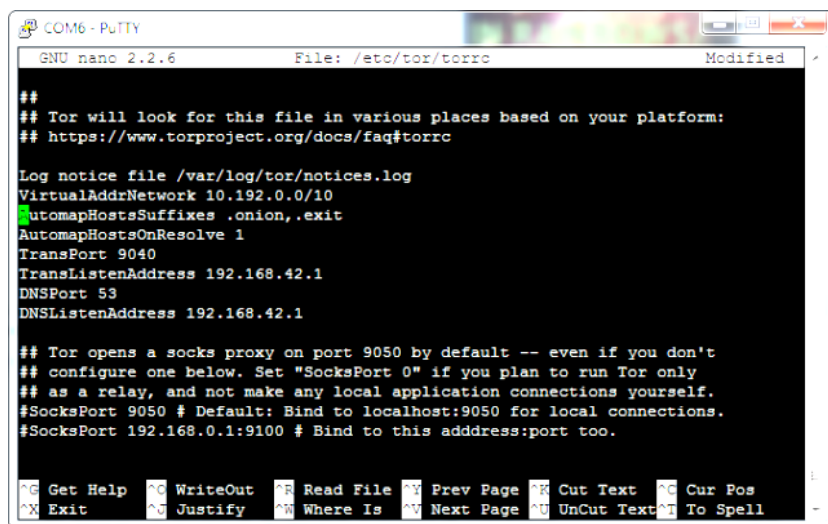


Рис.5.23 Редагування конфігураційного файла torrc

Відредагуйте хост точки доступу так, щоб він називався як-небудь, що запам'ятовується, типу Onion Pi (рис.5.24) - не забудьте задати хороший пароль, не використовуйте його за замовчуванням! (Не забудьте зробити до цього крок налаштування AP з розділу "Підготовка")

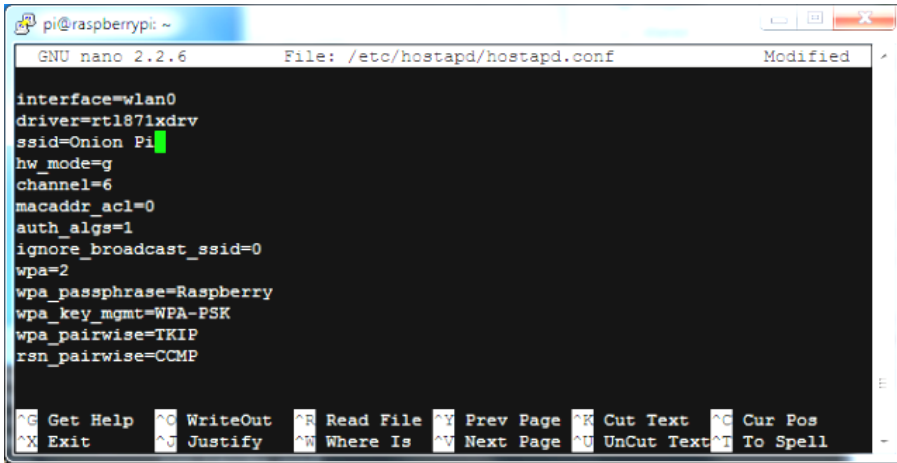


Рис.5.24 Редагування параметрів точки доступу

Час змінювати наші таблиці IP-маршрутизації, щоб з'єднання через інтерфейс Wi-Fi (wlan0) направлялися через програмне забезпечення Tor.

Введіть наступне, щоб видалити старі правила з NAT-таблиці IP:

```

sudo iptables -F
sudo iptables -t nat -F

```

Якщо хочете мати можливість SSH на своєму Pi після цього, то необхідно додати виключення для порту 22, подібно до наведеного нижче (не показано на рис.5.25):

```

sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j REDIRECT --to-ports 22

```

Введіть наступне, щоб направити всі запити DNS (UDP порт 53) від інтерфейсу wlan0 до внутрішнього порту 53 (DNSPort в нашій torrc):

```

sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --to-ports 53

```

Введіть команду для маршрутизації всього TCP-трафіка з інтерфейсу wlan0 в порт 9040 (TransPort в нашому torrc):

```

sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-ports 9040

```

Далі можете перевірити, що таблиці IP правильні (рис.5.25) з:

```

sudo iptables -t nat -L

```

```

pi@raspberrypi ~ $ sudo iptables -F
pi@raspberrypi ~ $ sudo iptables -t nat -F
pi@raspberrypi ~ $ sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53
-j REDIRECT --to-ports 53
pi@raspberrypi ~ $ sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j R
EDIRECT --to-ports 9040
pi@raspberrypi ~ $ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination            udp dpt:domain red
ir ports 53
REDIRECT   tcp  -- anywhere              anywhere              tcpflags: FIN,SYN,
RST,ACK/SYN redir ports 9040

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
pi@raspberrypi ~ $

```

Рис.5.25 Зміна таблиць IP-маршрутизації

Якщо все добре, то збережіть їх до нашого старого збереженого файла NAT (рис.5.26):

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Він буде автоматично завантажуватися, коли мережа налаштована на перезавантаження (як при налаштуванні точки доступу Pi).

```

pi@raspberrypi:~$ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
pi@raspberrypi:~$ cat /etc/iptables.ipv4.nat
# Generated by iptables-save v1.4.14 on Fri Jun 14 01:58:29 2013
*filter
:INPUT ACCEPT [1179:225538]
:FORWARD ACCEPT [1:67]
:OUTPUT ACCEPT [850:182821]
COMMIT
# Completed on Fri Jun 14 01:58:29 2013
# Generated by iptables-save v1.4.14 on Fri Jun 14 01:58:29 2013
*nat
:PREROUTING ACCEPT [97:11245]
:INPUT ACCEPT [74:7844]
:OUTPUT ACCEPT [23:1900]
:POSTROUTING ACCEPT [24:1967]
-A PREROUTING -i wlan0 -p udp -m udp --dport 53 -j REDIRECT --to-ports 53
-A PREROUTING -i wlan0 -p tcp -m tcp --tcp-flags FIN,SYN,RST,ACK SYN -j REDIRECT
--to-ports 9040
COMMIT
# Completed on Fri Jun 14 01:58:29 2013
pi@raspberrypi:~$

```

Рис.5.26 Збереження файла NAT

Далі створимо наш файл журналу (зручно для налагодження) з:

```

sudo touch /var/log/tor/notices.log
sudo chown debian-tor /var/log/tor/notices.log
sudo chmod 644 /var/log/tor/notices.log

```

Перевірте за допомогою команди:

```
ls -l /var/log/tor
```

Запустіть службу Tor вручну:

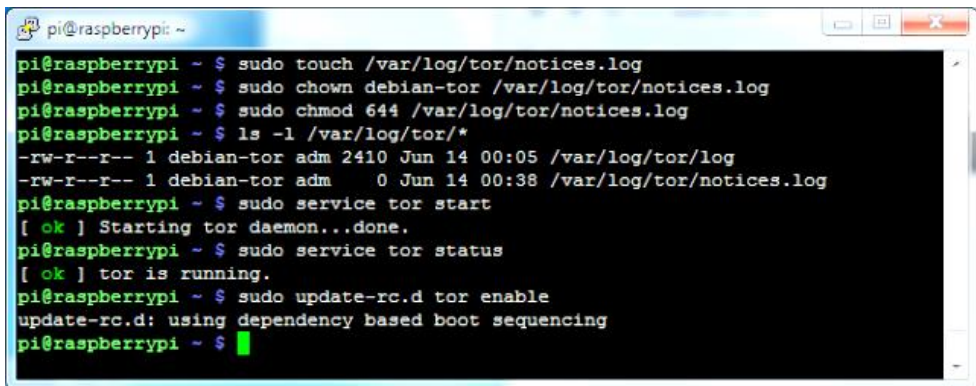
```
sudo service tor start
```

Перевірте, чи дійсно працює (ви можете запускати це щоразу, коли не впевнені, що все вірно і бачите багато сповіщень FAIL):

```
sudo service tor status
```

Нарешті, зробимо, щоб він запускався при завантаженні (рис.5.27):

```
sudo update-rc.d tor enable
```



```
pi@raspberrypi: ~
pi@raspberrypi ~ $ sudo touch /var/log/tor/notices.log
pi@raspberrypi ~ $ sudo chown debian-tor /var/log/tor/notices.log
pi@raspberrypi ~ $ sudo chmod 644 /var/log/tor/notices.log
pi@raspberrypi ~ $ ls -l /var/log/tor/*
-rw-r--r-- 1 debian-tor adm 2410 Jun 14 00:05 /var/log/tor/log
-rw-r--r-- 1 debian-tor adm    0 Jun 14 00:38 /var/log/tor/notices.log
pi@raspberrypi ~ $ sudo service tor start
[ ok ] Starting tor daemon...done.
pi@raspberrypi ~ $ sudo service tor status
[ ok ] tor is running.
pi@raspberrypi ~ $ sudo update-rc.d tor enable
update-rc.d: using dependency based boot sequencing
pi@raspberrypi ~ $
```

Рис.5.27 Налаштування запуску Tor при завантаженні

От і все, тепер ви готові до тестування.

## Тестування

Пора перевірити свій анонімізуючий TOR-проксі. На комп'ютері перевірте доступні мережі Wi-Fi і повинні побачити мережу Onion Pi (рис.5.28):

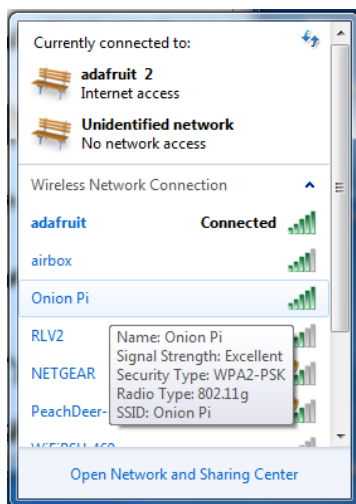


Рис.5.28 Мережа Onion Pi

Підключіться до неї (рис.5.29), використовуючи пароль, введений у файлі конфігурації `hostapd`:

Ви можете відкрити Terminal або командний рядок і виконати `ping`

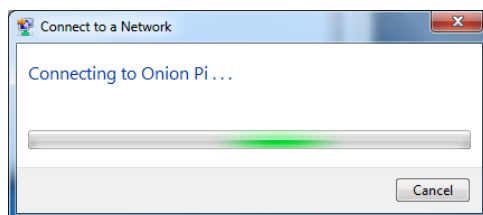


Рис.5.29 Під'єднання до мережі Onion Pi

192.168.42.1 для перевірки, що ваш зв'язок з Pi працює (рис.5.30). Однак, ви не зможете перевірити зв'язок поза ним, бо `ping` не транслюється через проксі:

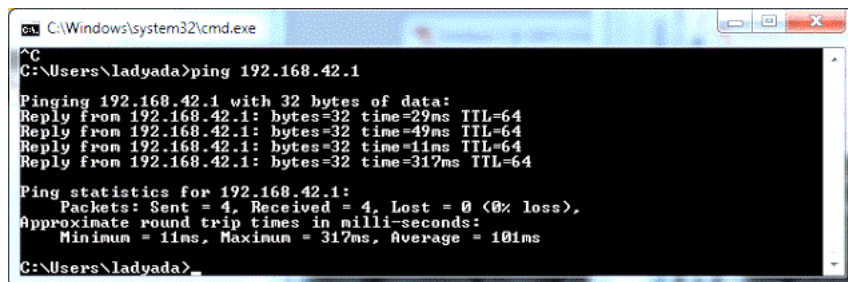


Рис.5.30 Перевірка зв'язку з Pi

Щоб переконатися, що проксі-сервер працює, відвідайте сайт типу <http://www.ipchicken.com>, який показуватиме вашу IP-адресу (рис.5.31), як він її бачить, а також відповідне доменне ім'я, якщо воно доступне. IP-адреса повинна бути не вашого інтернет-провайдера - а якщо перезавантажите сторінку, то вона повинна змінитися.



Рис.5.30 Перевірка роботи проксі-сервера Tor

Трафік вашого веб-браузера тепер анонімний!

**ПЕРЕД ПОЧАТКОМ ВИКОРИСТАННЯ ПРОКСІ** - пам'ятаєте, що є багато способів ідентифікувати вас, навіть якщо ваші IP-адреси «рандомізовані». Видаляючи і блокуючи кеш браузера, історію і куки - деякі браузери дозволяють «анонімні сесії». Не заходьте на існуючі облікові записи з особистою інформацією (якщо не впевнені, що це те, що хочете зробити).

Використовуйте SSL, при його наявності, кінець-в-кінців, шифруйте своє спілкування. І читайте [torproject.org](https://www.torproject.org/)<sup>94</sup>, де набагато більше інформації про те, як використовувати Tor в розумний і безпечний спосіб.

Тепер, коли цей проект створений, ви можете зробити більше...

### Налаштувати Tor, як посередника Wi-Fi до Wi-Fi

Ми використовуємо Ethernet, тому що це не вимагає налаштування або паролів, просто досить під'єднати кабель, щоб скористатися DHCP, але якщо хочете, то його можна налаштувати як проксі Wi-Fi до Wi-Fi. Вам буде потрібно мати два Wi-Fi адаптери, потім відредагувати `/etc/networks/interfaces` для додавання wlan1, та ввести SSID/пароль для свого інтернет-провайдера.

<sup>94</sup> <https://www.torproject.org/>

### Налаштувати Tor, щоб зробити свій вихідний вузол тільки в конкретній країні

Дуже легко налаштувати Tor, щоб мати присутність у будь-якій країні за своїм вибором. Наприклад, нижче наведений torrc автора, що змушує його «презентувати» у Великобританії.

Замініть `aaa.bbb.ccc.ddd` на IP-адресу свого RPi і `GB` кодом країни за своїм вибором. Налаштуйте свій браузер, щоб використовував Socks 5 проксі на `aaa.bbb.ccc.ddd`, порт 9050:

```
Log notice file /var/log/tor/notices.log
SocksListenAddress aaa.bbb.ccc.ddd
ExitNodes {GB}
StrictNodes 1
```

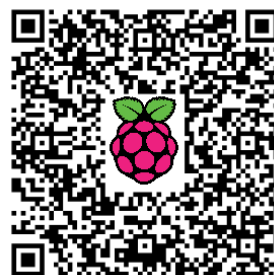


*Якщо Вам подобається використовувати Tor, то допоможіть зробити його швидшим, приєднавшись як ретранслятор (сервер), або збільшіть анонімність, ставши вихідним вузлом. Перевірте на веб-сайті проекту Tor<sup>95</sup>, як змінити torrc, щоб перетворити свій Pi в одне з названого.*

### Проксі-сервер для доступу в I2P



I2P є тим програмним забезпеченням, яке краще залишити працюючим вічно. Raspberry Pi - одна з платформи, яка чудово підходить для такого завдання<sup>96</sup>. Думаю, що ви могли б залишити свій Raspberry Pi працюючим "за лінією фронту", і змогли б отримувати до нього доступ через даркнет I2P.



### Підготовка

Перше, що ми зробимо, це оновимо операційну систему, так як в образі може бути далеко не остання версія різних пакетів та іншого:

```
#sudo aptitude update
```

<sup>95</sup> <https://www.torproject.org/docs/tor-relay-debian.html.en>

<sup>96</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1841-proxy-server-on-raspberry-pi-to-access-i2p>

```
#sudo aptitude upgrade
```

Нам потрібно буде створити каталог для установки. Це можна зробити за допомогою декількох простих команд, і буде працювати під управлінням користувача `pi`. Якщо використовуєте інший обліковий запис, то відповідно змініть шлях. Нагадуємо, що `~` просто умовне позначенням поточного домашнього каталогу користувача.

```
cd ~
mkdir i2p
cd i2p
```

Знайдіть URL, щоб завантажити I2P, перейшовши за посиланням<sup>97</sup>, але адреса з часом змінюється, бо з'являються нові версії. На момент написання цього розділу маємо працюючим наступне (рис.5.31):

```
wget
https://download.i2p2.de/releases/0.9.13/i2pinstall_0.9.13.jar

abo

wget
http://download.i2p2.no/releases/0.9.13/i2pinstall_0.9.13.jar
```

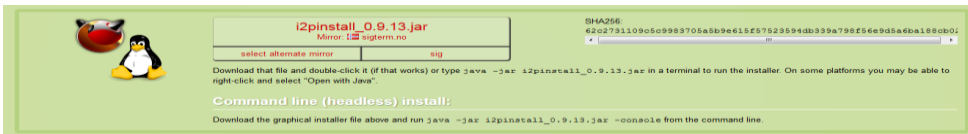


Рис.5.31 Завантаження файла встановлення I2P

## Встановлюємо Java.

Зверніть увагу: не треба ставити стандартну JRE з пакетів! Вона гальмує! Ми будемо ставити Java, спеціально скомпільовану для процесора ARM.

Тепер доступний Oracle JDK8 для ARM як видання Developer Preview! Розглянемо детально, як встановити Oracle Java SE 8 (з JavaFX) Developer Preview для ARM на Raspberry Pi.

Насамперед, завантажимо Java SE 8 (with JavaFX) Developer Preview for ARM<sup>98</sup> для Raspberry Pi.

## Переносимо Oracle JDK на Raspberry Pi

<sup>97</sup> <http://www.i2p2.de/download>

<sup>98</sup> <http://download.oracle.com/otn-pub/java/jdk/8-b132/jdk-8-linux-arm-vfp-hflt.tar.gz>

Після завантаження Oracle JDK до настільного комп'ютера, ми повинні перенести його на Raspberry Pi. Будемо використовувати SCP для передачі файлів мережею. Якщо працюєте на десктопному Windows, то завантажте і встановіть WinSCP<sup>99</sup>.

Якщо використовуєте Mac OSX, то можете завантажити і встановити Cyberduck<sup>100</sup>. Екрани будуть виглядати по-різному, але суть одна і та ж.

Створіть нову сесію в WinSCP за допомогою IP-адреси свого Raspberry Pi. Повноваження аутентифікації за замовчуванням для образу є ім'я користувача "pi" і пароль "raspberrry" (рис.5.32). Збережіть сесію, а потім увійдіть в систему. Вам може бути запропоновано прийняти відбиток SSH, виберіть "Yes", щоб прийняти і продовжити.

Після успішного встановлення з'єднання, виберіть диск і папку в лівій панелі, куди ви завантажили файл Oracle JDK на свою локальну машину. На правій панелі відображена файлова система на Raspberry Pi, ми залишимо її в положенні за замовчуванням в домашньому каталозі користувача "pi". Перетягніть файл Oracle JDK з лівої панелі на праву панель (рис.5.33) і WinSCP почне процес передачі файлів. Вам буде запропонований діалог передачі - просто натисніть кнопку "Copy", щоб почати передачу.

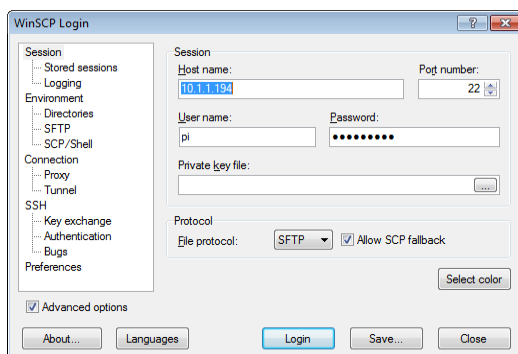


Рис.5.32 Створення нової сесії в WinSCP

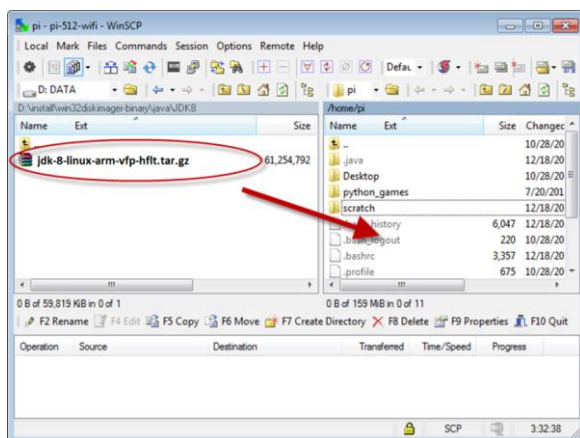


Рис.5.33 Копіювання файла Oracle JDK на Raspberry Pi

<sup>99</sup> <http://www.winSCP.com/>

<sup>100</sup> <http://cyberduck.ch/>

Коли завершення передачі файлу, можете закрити WinSCP (або Cyberduck).

### Процедура встановлення на Raspberry Pi

Решта кроків мають бути виконані безпосередньо на консолі Raspberry Pi або з використанням з'єднання через термінал SSH з доступом до оболонки. На останньому етапі, ми передали файл Oracle JDK в домашній каталог користувача "pi". Ми повинні увійти в систему, як користувач "pi" і відразу будемо в домашньому каталозі користувача.

Давайте створимо новий каталог, до якого будуть встановлені файли JDK:

```
sudo mkdir -p -v /opt/java
```

Далі, розпакуємо файл .gz Oracle JDK за допомогою команди:

```
tar xvzf ~/jdk-8-linux-arm-vfp-hflt.tar.gz
```

Процес розпакування займе кілька секунд. Буде розпакований весь вміст файлу .gz Oracle JDK в новий каталог з ім'ям jdk1.8.0, розташований в домашньому каталозі користувача.

З завершенням розпакування настав час перенести новий розпакований каталог в місце розташування встановленої Java, створене нами раніше як opt/java.

```
sudo mv -v ~/jdk1.8.0 /opt/java
```

Ми можемо також видалити оригінал файлу .gz, який нам більше не потрібний:

```
rm ~/jdk-8-linux-arm-vfp-hflt.tar.gz
```

Для завершення установки JDK нам потрібно, щоб система знала, що є нова встановлена JVM і де вона знаходиться. Використайте наступну команду, щоб виконати це завдання:

```
sudo update-alternatives --install "/usr/bin/java" "java"  
"/opt/java/jdk1.8.0/bin/java" 1
```

І, нарешті, ми також повинні повідомити системі, що хочемо, щоб цей JDK був для системи Java Runtime за замовчуванням. Наступна команда виконає цю дію (рис.5.34):

```
sudo update-alternatives --set java /opt/java/jdk1.8.0/bin/java
```

```
pi@raspberrypi ~ $ sudo mv -v ~/jdk1.8.0 /opt/java
'/home/pi/jdk1.8.0' -> '/opt/java/jdk1.8.0'
pi@raspberrypi ~ $ sudo update-alternatives --install "/usr/bin/java" "java" "/opt/java/jdk1.8.0/bin/java" 1
pi@raspberrypi ~ $ sudo update-alternatives --set java /opt/java/jdk1.8.0/bin/java
update-alternatives: using /opt/java/jdk1.8.0/bin/java to provide /usr/bin/java
(java) in manual mode
pi@raspberrypi ~ $
```

Рис.5.34 Встановлення Java

Тепер Java встановлений. Для перевірки і підтвердження ми можемо виконати команду перевірки версії Java за допомогою:

```
java -version
```

Ви повинні отримати таку відповідь (рис.5.35):

```
pi@raspberrypi ~ $ java -version
java version "1.8.0-ea"
Java(TM) SE Runtime Environment (build 1.8.0-ea-b36e)
Java HotSpot(TM) Client VM (build 25.0-b04, mixed mode)
pi@raspberrypi ~ $
```

Рис.5.35 Перевірка версії Java

Це підтверджує, що Oracle JDK встановлений і готовий до використання.

### Додавання змінної середовища JAVA\_HOME

Деякі програми Java вимагають наявності налаштованої в системі змінної JAVA\_HOME. Додайте наступний рядок до вашого `/etc/environment`, використовуючи улюблений текстовий редактор:

```
JAVA_HOME="/opt/java/jdk1.8.0"
```

Крім того, відредагуйте файл `~/.bashrc` за допомогою цієї команди:

```
nano ~/.bashrc
```

додавши наступні два рядки в кінець файлу, та збережіть його:

```
export JAVA_HOME="/opt/java/jdk1.8.0"
export PATH=$PATH:$JAVA_HOME/bin
```

Перезавантажте або повторно увійдіть в систему, щоб застосувати експорт у ваше середовище.

## Встановлення I2P-маршрутизатора

Вводимо:

```
#java -jar i2pinstall_0.9.13.jar -console
```

Замість `i2pinstall_0.9.13.jar` підставляєте назву свого файлу установника маршрутизатора, бо версія може змінитися. Шлях для встановлення: `/home/pi/i2p/`

Ось і все! I2P-маршрутизатор встановлений!

## Налаштування маршрутизатора

Хоча маршрутизатор встановлений, але працювати з ним ще рано. Він вас просто не "почує", тому що чекає команд з 127.0.0.1, а зовсім не з вашого комп'ютера. Виправимо:

```
#cd i2p
#./runplain.sh
#kill -9 `cat /tmp/router.pid` або killall java
#cd ..
#cd .i2p
#nano clients.config
```

Ми дозволяємо вперше запуск `runplain.sh`, а потім вбиваємо його для того, щоб створити профіль I2P для наступного редагування.

Якщо хочете, то можете скористатися RDP для підключення до Raspberry Pi і просто використовувати I2P звідти. Автор хоче зробити Pi воротами в I2P для всієї своєї локальної мережі, тому давайте зробимо так, щоб ми могли потрапити в веб-консоль з будь-якого ПК, а не тільки з локальний хоста. Відкриваємо в редакторі `~/.i2p/clients.config` і знаходимо рядок, який виглядає як:

```
clientApp.0.args=7657 ::1,127.0.0.1 ./webapps/
```

Коментуємо його з `#` і розкоментуємо рядок, який виглядає як:

```
#clientApp.0.args=7657 0.0.0.0 ./webapps/
B:
clientApp.0.args=7657 0.0.0.0 ./webapps/
```

Якщо ми зробимо цей крок, але не довіряємо всім хостам в нашій локальній мережі, то буде доцільним зробити пароль для маршрутизатора. Просто в `clients.config` додаємо рядок:

```
consolePassword=SomePassword
```

Очевидно, замінивши "SomePassword" паролем, який хочете використовувати. Ім'я для входу - admin. Переконайтеся, що тепер маєте доступ до консолі I2P, щоб мати можливість дістатися до проксі. Для цього ми повинні встановити проксі на портах 4444 і 4445 для прослуховування 0.0.0.0.

Продовжимо з редагування файлу `i2ptunnel.config`:

```
#nano i2ptunnel.config
```

Знайдіть рядки, які виглядають як :

```
tunnel.0.interface=127.0.0.1
tunnel.6.interface=127.0.0.1
```

і змініть їх на:

```
tunnel.0.interface=0.0.0.0
tunnel.6.interface=0.0.0.0
```

Тепер можемо запустити I2P:

```
#cd /home/pi/i2p
#./runplain.sh
```

але, якщо хочете запускати при завантаженні і бути впевненим, що все працює будь-коли, навіть після аварії, скористайтесь наведеним нижче рішенням.

Вводимо:

```
crontab -e
```

Цим ми запускаємо редактор, щоб додати заплановані завдання. Додайте наступні рядки:

```
0 * * * * /home/pi/i2p/runplain.sh
@reboot /home/pi/i2p/runplain.sh
```

Потім exit для завершення роботи.

Дані рядки повинні запускати I2P при завантаженні і робити спроби завантажити його щогодини. Приводом для кожного щогодинного рядка для перезапуску I2P є випадок, що він зламався. Якщо I2P вважає, що він вже запущений, то це повинно витончено закритись. Ми тепер повинні мати піднятий і працюючий I2P, залишилось конфігурувати свій браузер для вказівки на порти 4444 і 4445 для HTTP і HTTPS проксі, відповідно.

## Базові налаштування i2p

Багато користувачів віддають перевагу веб-інтерфейсу, а не файлам конфігурації.

Спочатку доступ до вебінтерфейсу дозволений тільки з локального комп'ютера. Так що будемо робити першу «дірку в безпеці» - відкривати доступ до адмінки з будь-якої адреси. Але для початку треба хоч якось до неї достукатися. Беремо `ssh` і робимо тунель з порту 7657 хостингу на порт 7657 нашого комп'ютера.

```
C:\>ssh user@ваш_сервер -L7657:127.0.0.1:7657
user@ваш_сервер's password:
Linux ваш_сервер 3.2.0-4-amd64 #1 SMP Debian 3.2.51-1 x86_64

You have mail.
Last login: Tue Dec 24 06:18:58 2013
ваш_сервер: ~>
```

Якщо бачимо запрошення оболонки, то за посиланням <http://127.0.0.1:7657/> стає доступна адмінка i2p-маршрутизатора.

По-перше, на закладці UI ставимо англійську мову (рис.5.36).

Тому що, якщо щось знадобиться знайти, то простіше шукати за англійськими назвами термінів. Наприклад, так відразу не здогадаєшся, що «транзитний трафік» - це «share bandwidth».

Закладка Bandwidth (рис.5.37) - автор виставив IN - 512, OUT - 256 і 50% share (той самий транзитний трафік через ваш сервер).

Звичайно, при 50% страждає анонімність і, потенційно, швидкість. Хочеться анонімності – приближайтесь до 100%. А швидкість треба налаштовувати залежно від каналу вашого сервера і ваших потреб.

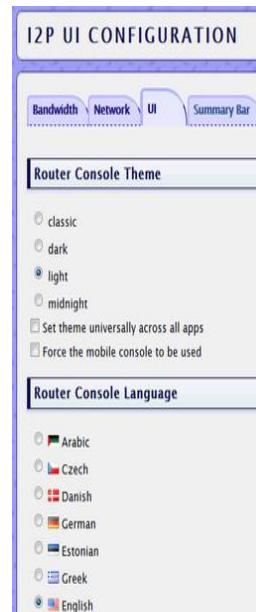


Рис.5.36 Налаштування теми і мови UI

Далі налаштовуємо адресну книгу. Справа в тому, що мережа I2P не має звичних нам DNS-серверів. Тому SusiDNS звертається до вже відомих ресурсів для поповнення так званої «Адресної книги».

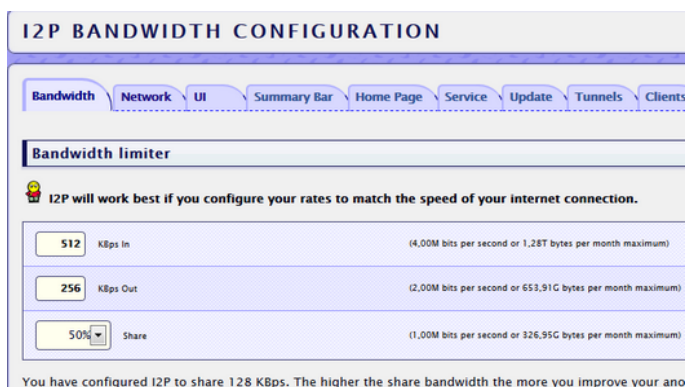


Рис.5.37 Налаштування полоси пропускання

Заходимо в браузері, щоб розширити список ресурсів, на <http://<IP Raspberry Pi>:7657/susidns/subscriptions> і додаємо:

```
http://www.i2p2.i2p/hosts.txt
http://i2host.i2p/cgi-bin/i2hostetag
http://stats.i2p/cgi-bin/newhosts.txt
http://tino.i2p/hosts.txt
http://inr.i2p/export/alive-hosts.txt
```

Основні налаштування зроблені, тому можна почати освоєння «невидисого» Інтернету.

## Raspberry Pi для пентестерів

Raspberry Pi, як інструмент для тестування на проникнення і збору інформації? Звичайно! Чому б і ні? Розглянемо деякі дистрибутиви й комплекти<sup>101</sup>, доступні для RPi.

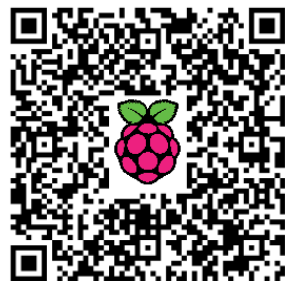
### PwnPi



**PwnPi**<sup>102</sup> є дистрибутивом на основі Linux для тестування на проникнення, до якого входить понад 200 встановлених

інструментів мережевої безпеки і який використовує XFCE як віконний менеджер. Нижче наведені деякі з інструментів PwnPi, як вони описані провідним розробником:

- 6tunnel – TCP-проксі HE для IPv6 додатків



<sup>101</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1758-raspberry-pi-for-pentester>

<sup>102</sup> [http://pwnpi.sourceforge.net/index.html\\_q=download.html](http://pwnpi.sourceforge.net/index.html_q=download.html)

- aircrack-ng - програма злому WEP/WPA
- amap - потужний перетворювач додатків
- arp-scan - інструмент arp-сканування і відбитків пальців
- bfbtester - двійковий брутфорс-тестер
- bing-ip2hosts - перерахування імен хостів для використання IP-посилань
- bsqbf - інструмент сліпої SQL-ін'єкції методом брутфорсу
- btscanner- сканер для Bluetooth пристроїв
- chaosreader - трасування сесій мережі і експорт результатів у формат HTML
- chkrootkit - детектор Rootkit
- cryptcat - полегшена версія розширеного netcat з шифруванням twofish
- darkstat - аналізатор мережевого трафіку
- dhcpcdump - парсер DHCP-пакетів з tcpdump
- dispy - графічна оболонка для objdump
- dmitry - глибинний інструмент збору інформації
- dns2tcp - клієнт і сервер TCP поверх DNS-тунелю
- dnswalk - перевірка інформації про зони DNS за допомогою сервера пошуку імен
- dsniiff - різні інструментальні засоби для перехоплення мережевого трафіку з незахищеним відкритим текстом
- enum4linux - інструмент для зчитування інформації з вікон і системи Samba
- etherape - графічний мережевий монітор
- fcrackzip - зломщик паролів для ZIP-архівів
- fimap - інструмент включення локальних і віддалених файлів
- flasm - асемблер і дизасемблер для Flash (SWF) байт-кодів
- foremost - програма криміналістики для відновлення втрачених файлів
- fping - посилає ICMP-пакети ECHO\_REQUEST до мережевих хостів
- ftp-proxy - проксі рівня додатків для протоколу FTP
- galleta - криміналістичний інструмент аналізу куків Internet Explorer
- ghettotooth - простий, але ефективний інструмент blue driving
- hostmap - інструмент відкриття імен хостів і віртуальних хостів
- hping3 - інструмент активної smashing мережі
- httptunnel - тунелювання потоку даних в HTTP-запитах
- httrack - копіювання веб-сайтів на комп'ютер (offline-браузер)
- hydra - дуже швидкий зламувальник входу в мережу
- ike-scan - знаходження і отримання відбитків пальців IKE-хостів (IPsec VPN-сервери)
- inguma - інструментарій тестування на проникнення з відкритим кодом
- iodine - інструмент для тунелювання IPv4 даних через сервер DNS
- ipcalc - обчислювач параметрів для IPv4-адрес
- isr-evilgrade - щоб скористатися перевагами реалізацій бідного оновлення шляхом введення підроблених поновлень
- ipgrab - утиліта, подібна до tcpdump, яка виводить детальну інформацію заголовка

- john - інструмент активного крекінгу паролів
- kismet - інструмент моніторингу бездротової мережі 802.11b
- knocker - простий і легкий у використанні сканер безпеки TCP-портів
- lcrack - загальний зломщик паролів
- ynis - інструмент аудиту безпеки для систем на основі UNIX
- macchanger - утиліта для редагування MAC-адрес мережевих інтерфейсів
- mboxgrep - Grep через поштові скриньки
- mdk3 - брутфорс SSID, брутфорс MAC-фільтрів, флуд SSID-маяків
- medusa - швидкий, паралельний, модульний, мережевий сервіс для брутфорсу логінів
- metagoofil - інструмент збору інформації, призначений для вилучення метаданих
- metasploit - проект безпеки, який надає інформацію про уразливість системи безпеки
- mysqlloit - інструмент захоплення SQL-Injection, зосереджений на LAMP
- mz - інструмент створення універсальних пакетів і генерації мережевого трафіку
- nbtscan - програма для сканування мережі для інформації про NetBIOS-ім'я
- netcat-traditional - TCP/IP швейцарський армійський ніж
- netdiscover - сканер активних/пасивних мережевих адрес за допомогою ARP-запитів
- netrw - інструмент, схожий на netcat, з хорошими функціями для транспортування файлів мережею
- netsed - редактор мережевого пакетно-змінного потоку
- netwag - графічна оболонка для netwox
- netwox - мережеві утиліти
- nikto - сканер безпеки веб-сервера
- nmapsi4 - графічний інтерфейс для nmap, мережевий сканер
- nmap - мережевий сканер
- nstreams - інструмент для аналізу вихідного tcpdump
- obexftp - утиліта передачі файлів для пристроїв, що використовують протокол OBEX
- onesixtyone - швидкий і простий сканер SNMP
- openvas-client - віддалений аудитор безпеки мережі, клієнт
- openvas-server - віддалений аудитор мережевої безпеки – сервер
- ophcrack-cli – кракер паролів Microsoft Windows з використанням райдужних таблиць (командний рядок)
- ophcrack - кракер паролів Microsoft Windows з використанням райдужних таблиць (GUI)
- otp - генератор одноразових Pads або паролів
- p0f - інструмент пасивної ідентифікації ОС
- packeth - генератор пакетів Ethernet
- packit - ін'єкція і захват в мережі
- pbnj - набір інструментів для моніторингу змін в мережі

- pentbox - набір, який об'єднує інструменти тестування безпеки і стабільності
- pdfcrack - зломщик паролів PDF-файлів
- pnsnscan - багатопотоковий сканер портів
- proxychains - переадресація підключень через проксі-сервери
- pscan - перевірка безпеки формату рядка для C-файлів
- ptunnel - тунелювання TCP-з'єднань через ICMP-пакети
- ratproxy - інструмент оцінки пасивної безпеки веб-додатків
- reaver - інструмент атаки брутфорс Wi-Fi, захищеного встановленим числом PIN
- s.e.t - інструментарій соціальної інженерії
- scrub - пише візерунки на магнітних носіях, щоб перешкодити відновленню даних
- secure-delete - інструмент стирання файлів, звільнення місця на диску, підкачки і пам'яті
- sendemail - легкий командний рядок SMTP-клієнта електронної пошти
- siege - утиліта регресивного тестування та аналізу HTTP
- sipcrack - самоскид/зломщик SIP-логіна
- sipvicious - пакет набору інструментів, які можуть бути використані для проведення аудиту систем на основі SIP VoIP
- skipfish - повністю автоматизований інструмент розвідки активної безпеки веб-додатків
- socat - багатofункціональний перемикач для двохнаправленої передачі даних
- splint - інструмент для статичної перевірки C-програми на помилки
- sqlbrute - інструмент для перебору даних з баз даних, використовуючи кліні SQL-ін'єкції
- sqlmap - інструмент, який автоматизує процес виявлення та експлуатації недоліків SQL-ін'єкцій
- sqlninja - інструмент SQL-ін'єкції сервера і поглинання
- ssldump - аналізатор протоколів мережі SSLv3/TLS
- sslscan - швидкий SSL-сканер
- sslsniff - інструмент атаки SSL/TLS «людина-посередині»
- sslstrip - інструмент атаки SSL/TLS «людина-посередині»
- stunnel4 - універсальний SSL тунель для мережевих демонів
- swaks - інструмент командного рядка для тестування SMTP
- tcpdump - аналізатор мережевого трафіку з командного рядка
- tcpflow - реєстратор TCP-потoku
- tcpick - sniffер TCP-потoku і підключення трекера
- tcpreplay - інструмент для відтворення файлів, збережених tcpdump при довільних швидкостях
- tcpslice - екстракт частин та/або склейка файлів tcpdump
- tcpspy - реєстратор вхідних та вихідних TCP/IP з'єднань
- tcptrace - інструмент для аналізу вихідного tcpdump
- tcpxtract - витягування файлів з мережевого трафіку на основі файлу підпису

- theHarvester - збирання повідомлень електронної пошти, піддоменів, хостів, імен співробітників, відкритих портів і банерів
- tinypoxy - легкий, без кешування, з можливою анонімністю HTTP-проксі
- tor - анонімізуюча накладена мережа для TCP
- u3-tool - інструмент для управління особливостями U3 USB флеш-дисків
- udtunnel - тунелювання пакетів UDP через з'єднання TCP
- ussp-push - клієнт для OBEX PUSH
- vidalia - контролер графічного інтерфейсу для tor
- vinetto - інструмент експертизи для дослідження файлів Thumbs.db
- voiphopper - інструмент тестування безпеки інфраструктури VoIP
- voipong - аналізатор VoIP і детектор дзвінків
- w3af-console - фреймворк, щоб знайти і використовувати уразливості веб-додатків (тільки CLI)
- w3af - фреймворк, щоб знайти і використовувати уразливості веб-додатків
- wariti - сканер вразливостей веб-додатків
- wash - сканування на вразливість WPS точки доступу
- wavemon - пристрій моніторингу бездротових додатків
- wbox - інструмент тестування HTTP і налаштування HTTP-сервера
- webhttrack - копіювання вебсайтів на ваш комп'ютер, httrack з веб-інтерфейсом
- weplab - інструмент, призначений для зламування ключів WEP
- wfuzz - утиліта, призначена для брутфорсу веб-додатків
- wipe - безпечне видалення файлів
- wireshark - аналізатор мережевого трафіку - GTK+ версія
- xprobe - дистанційне визначення ОС
- yersinia - програмне забезпечення для перевірки мережесхем вразливостей
- zenmap - мережевий сканер для Front End
- zzuf - пропускаючий додаток Fuzzer

Логіном за замовчуванням для цього дистрибутива є **root**, а пароль за замовчуванням **toor**, що нагадує автору про BackTrack Linux.

### Raspberry Pwn



**Raspberry Pwn**<sup>103</sup> являє собою інсталятор від Pwnie Експрес для перетворення вашого дистрибутиву Debian, який працює на RPi, в комплект тестування на проникнення, що завантажується з набором інструментів безпеки та аудиту, таких як SET, Fasttrack, kismet, aircrack-ng, nmap, dsniiff, netcat, nikto, xprobe, scapy, wireshark, tcpdump, ettercap, hping3, medusa, macchanger, nbtscan, john, ptunnel, p0f, ngrep, tcpflow, openvpn, iodine, httptunnel, cryptcat, sipsak,

---

<sup>103</sup> [https://github.com/pwnieexpress/raspberry\\_pwn](https://github.com/pwnieexpress/raspberry_pwn)

versinia, smbclient, sslsniff, tcptraceroute, pbnj, netdiscover, netmask, udptunnel, dnstracer, sslscan, medusa, ipcalc, dnswalk, socat, onesixtyone, tinyproxy, dmitry, fcrackzip, ssldump, fping, ike-scan, gpsd, darkstat, swaks, arping, tcpdump, sipcrack, proxychains, proxytunnel, siege, sqlmap, wapiti, skipfish і w3af.

Як бачимо, багато тих же інструментів, що і в PwnPi.

Raspberry Pwn простий в установці, але переконайтеся, що ви вже встановили Debian або оптимізовану версію Debian "wheezy".

## Встановлення Raspberry Pwn

1. Змініть розмір кореневого розділу, щоб використовувати всю SD-карту.
2. Запустіть SSH-сервіс і SSH на своєму Raspberry Pi, щоб могли мати доступ в термінал або консоль свого Debian. Також можете отримати термінальний доступ прямо зараз, якщо Raspberry Pi вже підключений до телевізора чи монітора.
3. Перейдіть до користувача root: `# sudo -s`
4. Установіть git (переконайтеся, що підключені до Інтернету):

```
# apt-get install git
```

5. Завантажте або клонуйте Raspberry Pwn інсталятор з репозиторію Pwnie Експрес Github:

```
# git clone https://github.com/pwnieexpress/Raspberry-Pwn.git
```

6. Перейдіть в директорію Raspberry-Pwn і запустіть сценарій встановлення:

```
# cd Raspberry-Pwn
# ./INSTALL_raspberry_pwn.sh
```

7. Дочекайтеся закінчення установки до кінця!

## PwnBerryPi



**PwnBerryPi<sup>104</sup>** - інший набір пентестера для Raspberry Pi і заснований на Raspberry Pwn від Pwnie Експрес, тому в основному можете очікувати ті ж інструменти, що і з Raspberry Pwn.

Послідовність інсталяції (рекомендуємо інсталювати на SD-карту не менше 4ГБ):

1. Змініть користувача на root:

<sup>104</sup> <https://github.com/g13net/PwnBerryPi>

```
# sudo bash
```

2. Переконайтеся, що версія вашого релізу Raspberry Pi Debian не нижча 6.0:

```
# cat /etc/debian_version
```

3. Перевірте доступ до Інтернету з Raspberry Pi

```
# ping google.com
```

4. Інсталюйте git:

```
# apt-get install git
```

5. Завантажте інстальатор PwnBerryPi з репозиторію g13net Github:

```
# git clone https://github.com/g13net/PwnBerryPi.git
```

6. Перейдіть в каталог з PwnBerryPi і запустіть встановлення:

```
# cd PwnBerryPi  
# ./install-pwnberrypi.sh
```

### Kali Linux



Наступник BackTrack Linux **Kali Linux**<sup>105</sup> також доступний для Raspberry Pi та для інших архітектур ARM. І він також використовує XFCE як менеджера робочого столу для більшої продуктивності. На відміну від BackTrack Linux, Kali заснований на дистрибутиві Debian GNU/Linux, але він, як і раніше, спрямований на комп'ютерні експертизи, тестування бездротових мереж на проникнення, веб-злом і багато іншого.

Більше 300 інструментів тестування на проникнення і програм аудиту безпеки попередньо встановлені для цього дистрибутива, які включають theMetasploit Framework, Nmap, SQLmap, Openvas, Aircrack-ng, John, Hydra, Maltego, zaproxy, Wireshark, sslsniff, webmitm, hexinject, dex2jar і т.д.

Більшість додатків запускаються з консолі і мають хорошу документацію. Для її виклику достатньо ввести в консолі **man назва\_програми**. Коротко прокоментуємо найпопулярніші додатки, які винесені в окреме меню Top 10 Security Tools.

- aircrack-ng - злом WEP і WPA паролів, який здійснюється шляхом накопичення пакетів. Зазвичай використовується в зв'язці з

---

<sup>105</sup> <http://docs.kali.org/armel-armhf/install-kali-linux-arm-raspberry-pi>

airmon-ng, airodump-ng і aireplay-ng, а для перебору додають john. Може використовуватися як сніффер.

- burpsuite - відмінна платформа для проведення атак на веб-додатки
- hydra - зломщик логінів від легендарного угруповання "The Hacker Choice". Дозволяє підбирати паролі до POP3, IMAP, SSH, FTP, MySQL, MS-SQL, HTTP, HTTPS, використовуючи паралельні процеси
- john - утиліта для відновлення паролів за їхніми хешам. Виконує як атаки за словником, так і брутфорс
- maltego - збір інформації з різних баз даних для соціальної інженерії
- metasploit framework - платформа для створення і налагодження експлойтів
- nmap - утиліта, призначена для різноманітного налаштовуваного сканування IP-мереж з будь-якою кількістю об'єктів, визначення стану портів і відповідних їм служб
- sqlmap - додаток для сканування вразливостей в sql-подібних системах управління базами даних. Дозволяє робити sql-ін'єкції
- wireshark - аналізатор мережевого трафіку. Перехоплює трафік і відображає його в детальному вигляді
- zargoxy - інструмент, який дозволяє спростити пошук вразливостей у веб-додатках

Інструменти для Kali Linux відносяться до таких топових категорій засобів безпеки:

- Збір інформації
- Аналіз вразливостей
- Веб-додатки
- Атаки на пароль
- Атаки на бездротові мережі
- Інструменти експлоїтів
- Сніфери/Підміна (Spoofing)
- Забезпечення доступу
- Зворотне проектування
- Стрес-тестування
- Злом обладнання
- Криміналістика
- Інструменти для створення звітів

Якщо хочете встановити Kali Linux на свій Raspberry Pi, то виконайте такі дії:

1. Придбайте швидку 8 ГБ (або більше) SD-карту. Ми настійно рекомендуємо карту Class 10.
2. Завантажте версію образу Kali Linux для Raspberry Pi з kali.org.
3. Скористайтесь утилітою dd, щоб копіювати файл цього образу на карту SD. У нашому прикладі ми припускаємо, що

запам'ятовуючий пристрій знаходиться в `/dev/sdb`. Змініть цей шлях при необхідності.

**Увага!** Цей процес очистить Вашу SD-карту. Якщо Ви виберете неправильний пристрій пам'яті, то Ви можете втратити свій жорсткий диск комп'ютера.

```
root@kali:~ dd if=kali-pi.img of=/dev/sdb bs=512k
```

Процес може зайняти якийсь час, залежно від швидкості Вашого USB-пристрою зберігання і розміру образу. Як тільки dd завершить операцію, завантажте свій Raspberry Pi з підключеної карти SD. Ви зможете увійти в Kali (`root/toor`) і `startx`. От і все, готово!

**Увага!** Будь ласка, змініть Ваші ключі SSH-хоста якомога швидше, бо ВСІ RPI-образи мають однакові ключі. Ви повинні також змінити пароль адміністратора для ще більшої безпеки, особливо якщо ця машина буде публічно доступною!

Змінити ключі SSH-хосту можна, виконавши такі дії:

```
root@kali:~ rm /etc/ssh/ssh_host_*
root@kali:~ dpkg-reconfigure openssh-server
root@kali:~ service ssh restart
```

Якщо ви розробник і хочете возитися з образом Kali<sup>106</sup> Raspberry Pi, включаючи зміну конфігурації ядра, прочитайте статтю про налаштування образів Raspberry Pi за посиланням внизу сторінки.

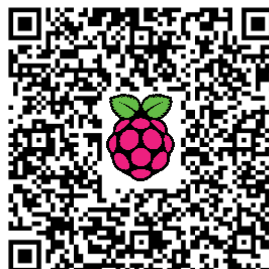
Звичайно ж, далеко не всі розглянуті вище утиліти і додатки ви будете використовувати і, можливо, не всі пакети встановлювати і пробувати. Але впевнений, що серед всього цього різноманіття ви знайдете те, що припаде до душі, і отримаєте насолоду від ефективної роботи улюблених інструментів.

### SSH-приманка: Kippo, Kali і Raspberry-Pi



Kippo - SSH-приманка<sup>107</sup>, призначена для реєстрації доступу до SSH через атаки «грубої сили» (перебір) і всієї взаємодії зломисника з оболонкою при успішній атаці.

Дана програма забезпечує:



<sup>106</sup> <http://docs.kali.org/development/custom-raspberry-pi-image>

<sup>107</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1821-ssh-honeynet-kippo-kali-and-raspberry-pi>

- Запис паролів і користувачів, які атакують, намагаючись виконати атаку грубої сили.
- Створення правильного імені користувача/пароля, як "root/root", щоб запропонувати зловмисникові підроблену файлову систему (нагадує Debian 5.0) з можливістю додавання і видалення файлів і збереження команд, виконуваних зловмисником.
- Збереження підозрілих файлів, завантажених (через `wget`) атакуючим.
- Можливість додавання в `/etc/passwd` файла таким чином, що атакуючий може "захопити" його.

В даний час, для Raspberry-Pi доступний Kali Linux, тому автор вважає, що це буде чудовою ідеєю встановити HoneyNet (приманку) на ньому. Ми матимемо можливість використовувати всі інструменти, розміщені на дистрибутиві Kali Linux, в той же час буде запущений наш Kippo.

## Як встановити Kippo

Ми не лише встановимо Kippo, а також встановимо базу даних MySQL, щоб зберегти події, і Kippo-Graph, що переглядати ці події через веб-інтерфейс.

Будь ласка, виконайте наступні дії, щоб встановити Kippo.

```
sudo apt-get install subversion python-twisted python-mysqldb
apache2
```

### 1. Встановіть MySQL:

```
root@kali:/# apt-get install mysql-server
root@kali:/# apt-get install mysql-client
```

### 2. Створіть базу даних та користувача з ім'ям Kippo з усіма привілеями:

```
root@kali:/# mysql -h localhost -u root -p
mysql> create database kippo;
mysql> GRANT ALL ON kippo.* TO 'kippo'@'localhost' IDENTIFIED BY
'Kippo-password';
exit
```

### 3. Завантажте Kippo з [kippo.googlecode.com](http://kippo.googlecode.com)<sup>108</sup> і розпакуйте його в `/usr/local/src/`. Для цього перейдіть в дану директорію і розпакуйте архів, вказавши повний шлях до нього:

```
tar -xzvf ./kippo-0.8.tar.gz
```

<sup>108</sup> <http://kippo.googlecode.com/files/kippo-0.8.tar.gz>

*Примітка.* Використовувані в команді ключі:

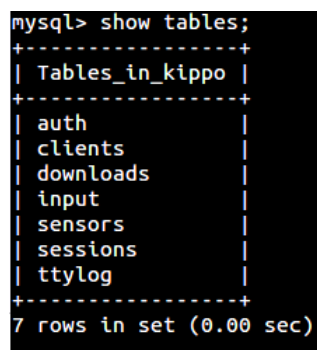
- x** - дозволяє витягувати файли з архіву.
- v** - робить виведення tar докладним. Це означає, що на екран будуть виведені всі знайдені в архіві файли. Якщо ця опція опущена, інформація, що виводиться в процесі обробки, буде обмежена.
- f** - є обов'язковою опцією. Без неї tar намагається використовувати магнітну стрічку замість файлу архіву.
- z** - дозволяє обробляти архів, стиснений gzip'ом (з розширенням .gz). Якщо забудете вказати цю опцію, то tar видасть помилку. І навпаки, ця опція не повинна використовуватися для нестислих архівів.

4. Створіть таблиці за допомогою тільки що створеного користувача (рис.5.38):

```
root@kali:/# cd /usr/local/src/kippo-0.8/doc/sql/
mysql> mysql -u kippo -p
mysql> use kippo;
mysql> source mysql.sql;
mysql> show tables;
exit
```

5. Додайте в kippo.cfg наступні рядки:

```
[database_mysql]
host = localhost
database = kippo
username = kippo
password = Kippo-password
```



```
mysql> show tables;
+-----+
| Tables_in_kippo |
+-----+
| auth             |
| clients          |
| downloads        |
| input            |
| sensors          |
| sessions         |
| ttylog           |
+-----+
7 rows in set (0.00 sec)
```

Рис.5.38 Створені таблиці

6. Створіть непривілейованого користувача, щоб запустити Kippo і дати йому доступ до папки:

```
root@kali: useradd -d /home/kippo -s /bin/bash -m kippo -g sudo
root@kali:/usr/local/src# chown -R kippo kippo-0.8/
```

7. Встановіть пакети, необхідні для Kippo-Graph:

```
sudo apt-get update
sudo apt-get install libapache2-mod-php5 php5-cli php5-common
php5-cgi php5-mysql php5-gd
```

8. Завантажте Kippo-Graph:

```
root@kali:/# wget http://bruteforce.gr/wp-content/uploads/kippo-graph-0.8.tar
root@kali:/# mv kippo-graph-0.8.tar /var/www/
```

```
root@kali:/var/www# tar xvf kippo-graph-0.8.tar --no-same-
permissions
chmod 777 generated-graphs
vim config.php #enter the appropriate values
sudo /etc/init.d/apache2 restart
```

### 9. Запустіть Kippo:

```
root@kali:/usr/local/src/kippo-0.8# su kippo
kippo@kali:/usr/local/src/kippo-0.8# ./start.sh
```

З щойно встановленим Kippo, необхідно опублікувати службу в Інтернеті. За замовчуванням Kippo слухає порт 2222. Ви можете опублікувати його, встановивши PAT, автор мав на увазі перенаправлення в маршрутизаторі порта 22 від зовнішнього IP на порт 2222 внутрішнього (приватного) IP Kippo.

Щоб побачити графіки, просто отримайте доступ до [http://Raspberry-Pi\\_IP\\_адреса/kippo-graph/](http://Raspberry-Pi_IP_адреса/kippo-graph/)

Серед багатьох графіків ви зможете побачити:

- Топ-10 паролів
- Топ-10 імен користувачів
- Зондування за день/тиждень
- Топ-10 SSH-клієнтів
- Активність людини всередині приманки
- Топ-10 входів (в цілому)
- Топ-10 успішних входів
- Топ-10 невдалих спроб входу
- Команди passwd
- Команди wget
- Виконані сценарії
- Цікаві команди
- Команди apt-get
- Топ-10 IP-адрес зондування системи за попередній місяць
- Всього IP-адрес зондування системи для топ-10 країн

На погляд автора, найкращою особливістю Kippo є можливість запропонувати зловмисникові підроблену файлову систему та збереження команд, які були виконані порушником, просто дозволяючи йому отримати доступ до системи, коли була виконана "успішна" атака.

У файлі `/usr/local/src/kippo-0.8/data/userdb.txt` ви можете встановити ім'я користувача/пароль, який "дозволяє" отримати доступ до підробленої системи. Ви можете встановити пароль `root:0:root` або інший, за бажанням, щоб дозволити хакеру отримати доступ.

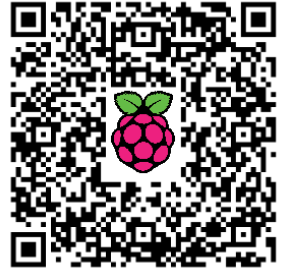
Так можна отримати декілька шкідливих зразків і нові сценарії, створені хакерами. Це чудова можливість дізнатися про нові хакерські тенденції!

## 6. ПРОГРАМУВАННЯ ДЛЯ ВЕБУ

### Встановлення персонального веб-сервера на Raspberry Pi



Є кілька причин, чому варто використати свій Raspberry Pi як веб-сервер<sup>109</sup>. Одна з них - попрактикуватися в написанні PHP-сценаріїв або PERL-сценаріїв.



PHP - чудова сучасна мова програмування, яка включає в себе багато принципів інших мов програмування. Ви можете вивчити з PHP такі загальні методи, як Test Driven Development. Також доступні такі хороші фреймворки кодування, як Codeigniter, що допоможуть вам вивчити MVC. Навички, отримані в PHP, можна легко перенести на інші мови, включаючи C# і Java.

Іншим, більш аматорським програмістам, може сподобатися веб-сервер для запуску чогось на зразок Joomla для створення домашнього інтернет-сайту.

Людям, які розробляють шаблони Wordpress, може сподобатися особистий веб-сервер, щоб перевірити нові шаблони. За допомогою Raspberry Pi, як веб-серверу, дизайнери можуть використовувати інші комп'ютери, щоб переконатися, наскільки добре виглядає макет в Internet Explorer, Chrome і Firefox.

В даному розділі припускається, що:

- Ви використовуєте за замовчуванням операційну систему Raspbian Wheezy
- Веб-сервер, який повинен бути використаний, це Lighttpd
- PHP буде встановлений як мова сценаріїв
- Як сервер бази даних буде встановлений MySQL

#### Підготовка до встановлення

Спочатку знайдіть IP-адресу свого Pi. Якщо працюєте безпосередньо на Raspberry Pi, то можете пропустити цю частину, але якщо звертаєтеся до нього через SSH то, щоб отримати IP-адресу Raspberry Pi, введіть наступне у вікні терміналу на Raspberry Pi:

```
ifconfig
```

Вам повернеться IP-адреса: щось у вигляді 192.168.1.x, де x - четверта частина адреси (наприклад, 30).

---

<sup>109</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1767-install-personal-webserver-on-raspberry-pi>

Якщо будете слідувати даному розділу безпосередньо на Pi, то можете використовувати `localhost` замість IP-адреси, як вказано нижче. А щоб увійти на Pi за допомогою SSH з Windows вам необхідно запустити Putty, ввести IP-адресу в полі Host Name (Ім'я хоста), а потім натиснути кнопку Open (Відкрити).

Якщо плануєте мати доступ до Raspberry Pi через SSH з ОС Linux, то відкрийте вікно терміналу і введіть SSH 192.168.1.x (де x – остання частина адреси).

Якщо заходите з Windows чи Linux, то буде запропоновано ввести ім'я користувача і пароль. Ви повинні ввести `pi` для користувача і пароль, який створили для користувача `pi` при встановленні Raspbian (за замовчуванням, `raspberrypi`).

Наступним кроком підготовки буде оновлення з репозиторію. Насамперед, запустіть `apt-get` оновлення, щоб переконатися, що маєте доступним все необхідне для завантаження.

```
sudo apt-get update
```

Тепер можна приступити до встановлення.

## Встановлення Lighttpd

Є досить багато керівництв в Інтернеті, які розповідають, як налаштувати веб-сервер. Для встановлення на Raspberry Pi автор вибрав Lighttpd, що перекладається як "світло в природі", бо він не використовує занадто багато ресурсів Raspberry Pi.

Щоб встановити Lighttpd виконайте наступну команду:

```
sudo apt-get install lighttpd
```

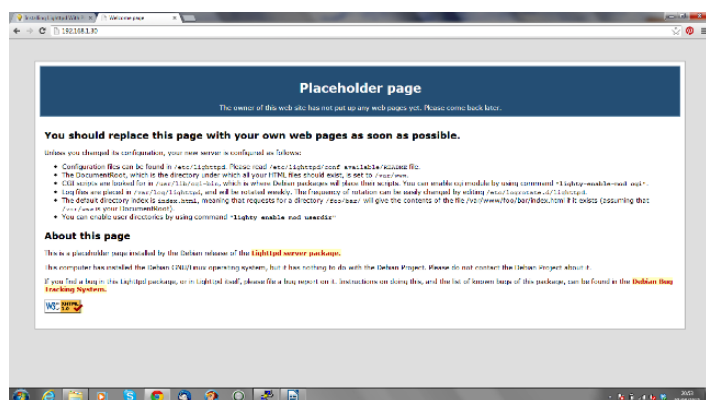


Рис.6.1 Сторінка заповнювача веб-сервера

Для перевірки, чи правильно встановлений веб-сервер, введіть наступне в адресному рядку браузера:

`http://192.168.1.x`

Замініть `x` останнім номером IP-адреси вашого Pi. Якщо ж працюєте безпосередньо на Raspberry Pi, то можете просто ввести наступне:

`http://localhost`

Ви повинні побачити сторінку заповнювача, як показано вище (рис.6.1).

### Встановлення PHP

Для встановлення PHP виконайте наступну команду:

```
sudo apt-get install php5-cgi
```

Тепер потрібно додати рядок у файл `php.ini`. Щоб це зробити, необхідний файл з назвою `php.ini` знаходиться в `/etc/php5/cgi`. Введіть наступне (рис.6.2):

```
sudo nano /etc/php5/cgi/php.ini
```

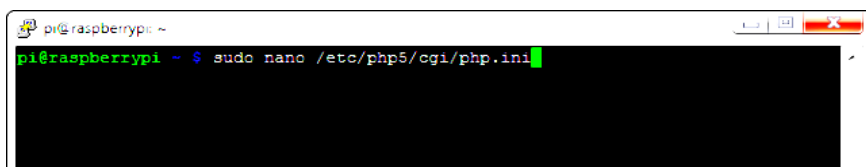


Рис.6.2 Запуск редагування файла `php.ini`

Перейдіть в кінець файлу і додайте рядок:

```
cgi.fix_pathinfo = 1
```

Щоб зберегти файл натисніть CTRL+O на клавіатурі, а щоб вийти з редактора `nano`, натисніть CTRL+X.

Нарешті, щоб швидко отримати CGI для роботи, введіть наступне в командному рядку:

```
sudo lighty-enable-mod fastcgi
```

(ЗВЕРНІТЬ УВАГУ, ЩО ЦЕ ВІДРІЗНЯЄТЬСЯ ВІД ОРИГІНАЛЬНОЇ ВЕРСІЇ)

Введіть наступну команду:

```
sudo lighty-enable-mod fastcgi-php
```

Перезапустіть веб-сервер:

```
sudo /etc/init.d/lighttpd restart
```

## Створення тестового сценарію PHP

Щоб перевірити, що PHP встановлений правильно, введіть у вікні терміналу:

```
sudo nano /var/www/phpinfo.php
```

Тепер в nano введіть наступний сценарій:

```
<?php phpinfo();>
```

Натисніть CTRL+O, щоб зберегти файл, та CTRL+X, щоб вийти з nano.

| PHP Version 5.4.4-14                                                                                                           |                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|                                               |                                                                                                                                 |
| System                                                                                                                         | Linux raspberrypi 3.6.11+ #371 PREEMPT Thu Feb 7 16:31:35 GMT 2013 armv6l                                                       |
| Build Date                                                                                                                     | Mar 7 2013 08:16:35                                                                                                             |
| Server API                                                                                                                     | CGIFastCGI                                                                                                                      |
| Virtual Directory Support                                                                                                      | disabled                                                                                                                        |
| Configuration File (php.ini) Path                                                                                              | /etc/php5/cgi                                                                                                                   |
| Loaded Configuration File                                                                                                      | /etc/php5/cgi/php.ini                                                                                                           |
| Scan this dir for additional .ini files                                                                                        | /etc/php5/cgi/conf.d                                                                                                            |
| Additional .ini files parsed                                                                                                   | /etc/php5/cgi/conf.d/10-pdo.ini                                                                                                 |
| PHP API                                                                                                                        | 20100412                                                                                                                        |
| PHP Extension                                                                                                                  | 20100525                                                                                                                        |
| Zend Extension                                                                                                                 | 220100525                                                                                                                       |
| Zend Extension Build                                                                                                           | API20100525.NTS                                                                                                                 |
| PHP Extension Build                                                                                                            | API20100525.NTS                                                                                                                 |
| Debug Build                                                                                                                    | no                                                                                                                              |
| Thread Safety                                                                                                                  | disabled                                                                                                                        |
| Zend Signal Handling                                                                                                           | disabled                                                                                                                        |
| Zend Memory Manager                                                                                                            | enabled                                                                                                                         |
| Zend Multibyte Support                                                                                                         | provided by mbstring                                                                                                            |
| IPv6 Support                                                                                                                   | enabled                                                                                                                         |
| DTrace Support                                                                                                                 | disabled                                                                                                                        |
| Registered PHP Streams                                                                                                         | https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip                                         |
| Registered Stream Socket Transports                                                                                            | tcp, udp, unix, udg, ssl, ssh3, tls                                                                                             |
| Registered Stream Filters                                                                                                      | zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk |
| This program makes use of the Zend Scripting Language Engine:<br>Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies |                                                                                                                                 |
| Powered By<br>                              |                                                                                                                                 |

Рис.6.3 Панель керування PHP

В адресному рядку веб-браузера введіть наступний текст:

```
http://192.168.1.x/phpinfo.php
```

Знову ж, замініть x останньою частиною вашої IP-адреси. Якщо ж працюєте безпосередньо на Raspberry Pi, то для доступу в панель керування PHP (рис.6.3) можете просто ввести наступне:

```
http://localhost/phpinfo.php
```

### Встановлення MySQL

Є три додатки, які будуть встановлені для MySQL:

- MySQL-сервер - фактичний сервер бази даних.
- MySQL-клієнт - дозволяє взаємодіяти з сервером з командного рядка.
- PHPMyAdmin - дозволяє взаємодіяти з сервером з веб-браузера.

Для встановлення всіх трьох додатків одночасно введіть таку команду у вікні терміналу (рис.6.4) :

```
sudo apt-get install mysql-server mysql-client phpmyadmin
```

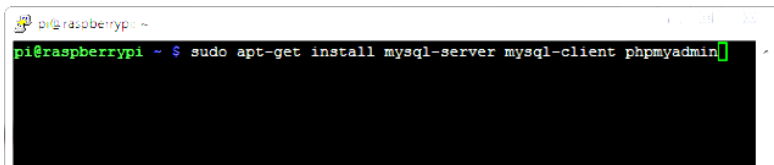


Рис.6.4 Запуск встановлення додатків для MySQL

Коли пакети будуть знайдені, буде виведено, скільки місця займе їх встановлення, а також з'явиться запит на продовження встановлення (рис.6.5):

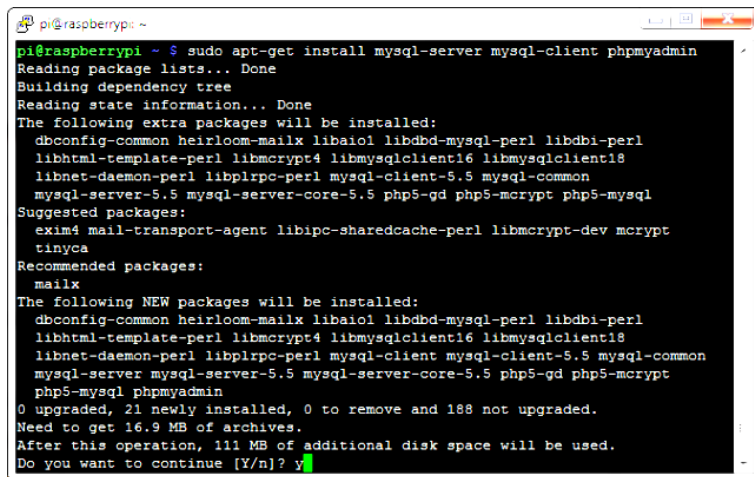


Рис.6.5 Продовження встановлення додатків

Під час встановлення буде запропоновано ввести пароль для суперкористувача (рис.6.6). Не плутайте це з кореневим користувачем в Linux. Це кореневий пароль для сервера баз даних. Введіть надійний пароль. Повторіть пароль, коли про це попросять.

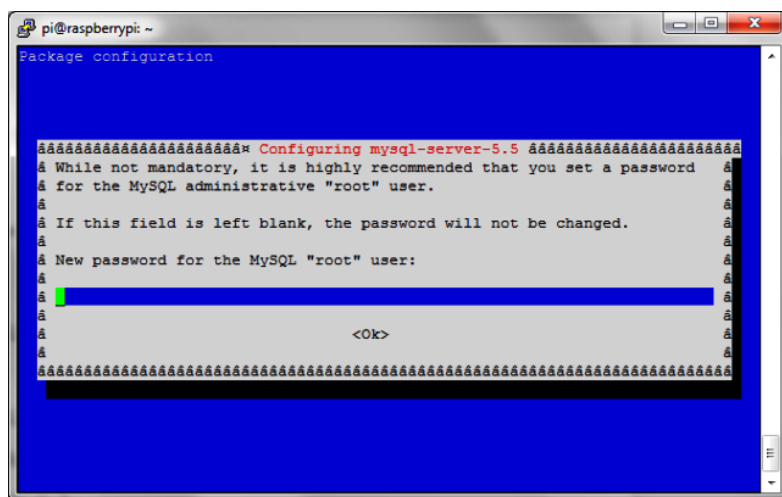


Рис.6.6 Запит на створення пароля для суперкористувача

Під час встановлення PHPMyAdmin вас запитатимуть про встановлений веб-сервер. Виберіть Lighttpd (рис.6.7).

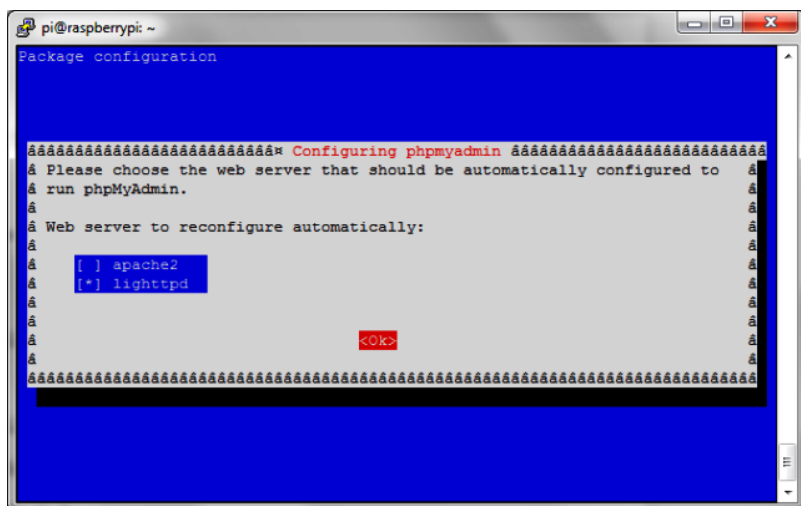


Рис.6.7 Вибір серверу для автоматичного конфігурування

З'явиться повідомлення з питанням, чи хочете ви створити фіктивну базу даних (6.8). Як написано у повідомленні, якщо знаєте як збираєтеся використовувати сервер бази даних чи база даних вже налаштована, то можете відповісти "No" ("Hi"), але якщо просто експериментуєте, то можете відповісти ствердно.

Автор рекомендує відповідати на це ствердно. Від цього не буде нічого поганого.

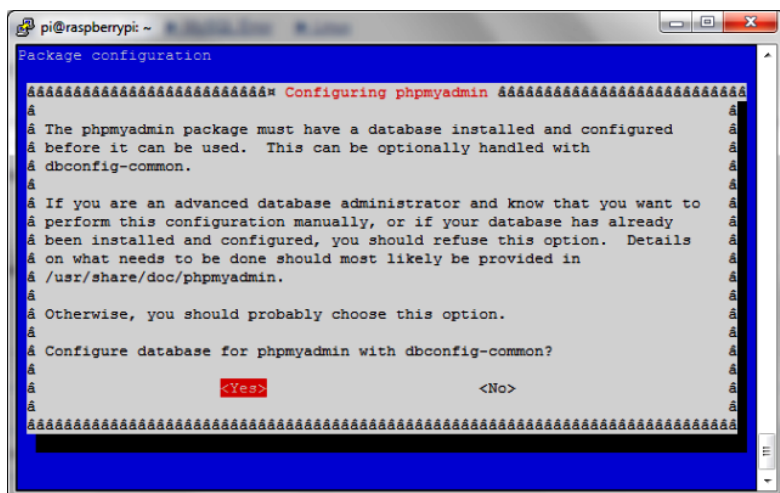


Рис.6.8 Запит на створення фіктивної бази даних

Тепер введіть пароль суперкористувача і повторіть його (рис.6.9). Введіть пароль суперкористувача, створений раніше. (Пам'ятаєте, що це кореневий пароль бази даних, а не пароль суперкористувача Linux?).

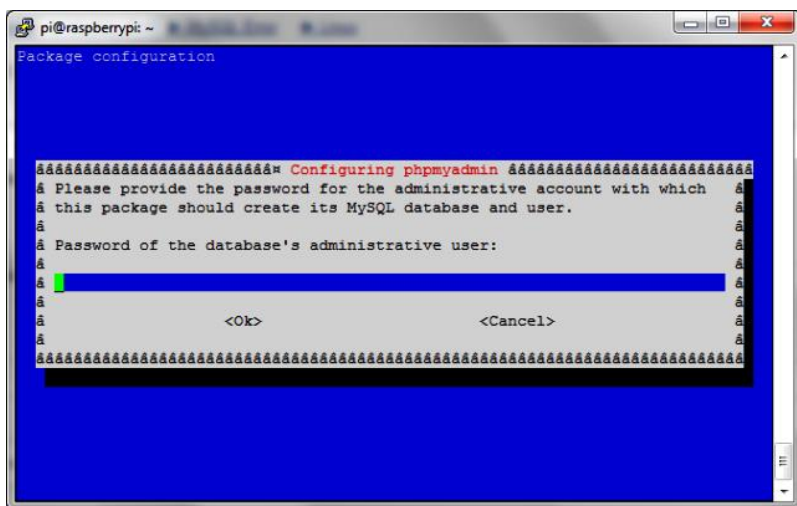


Рис.6.9 Запит на введення пароля суперкористувача

Коли встановлення буде завершено, перезапустіть сервер, ввівши в терміналі:

```
sudo /etc/init.d/lighttpd restart
```

Щоб перевірити, що сервер бази даних встановлений правильно, відкрийте вікно браузера і введіть наступне в адресному рядку:

`http://192.168.1.x/phpmyadmin`

Вчергове нагадаємо, що треба замінити `x` останньою частиною вашої IP-адреси. Якщо ж працюєте безпосередньо на Raspberry Pi, то досить ввести наступне:

`http://localhost/phpmyadmin`

Введіть ім'я користувача та пароль для входу в систему (рис.6.10):

Рис.6.10 Вікно входу в систему

## Висновки

В даному розділі ми розглянули як налаштувати основний веб-сервер з PHP і MySQL. Тепер можете спробувати встановити Wordpress, Joomla, CodeIgniter і Blueprint. Коротко про роботу з Lighttpd за посиланням<sup>110</sup>, в статті<sup>111</sup> розглядається, як налаштувати сервер для підтримки https.

<sup>110</sup> <http://www.web-analitik.info/articles/hosting/rabota-s-veb-serverom--lighttpd/>

<sup>111</sup> <http://www.ylsoftware.com/news/553>

### Coder - простий спосіб програмувати на Raspberry Pi



Творча  
лабораторія Google  
надала програмне  
забезпечення, що

перетворює Raspberry Pi в крихтний веб-сервер<sup>112</sup>, який може бути використаний для веб-проектів студентів, які вивчають інформатику.

За словами розробників, вистачить всього десяти хвилин, щоб підняти його з Raspberry Pi, а потім зможете використовувати Pi як особистий веб-сервер.

Coder можна використовувати, щоб почати веб-розробки з нуля, або ним можуть скористатися веб-програмісти як особистою пісочницею для своїх експериментів. Він розроблений, щоб бути нескладним – вам треба лише завантажити програми встановлення на карту SD 4GB і вставити SD-карту в Pi. Встановити з'єднання між Pi і комп'ютером. Через coder.local в Chrome будете підключені до нового Coder і зможете почати писати HTML, CSS, Javascript з використанням вільного середовища.

Більш детально про інсталяцію та запуск Coder за посиланням<sup>113</sup>.

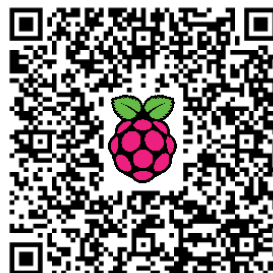
Raspberry Pi захопив розум у просторі, як ефективний освітній інструмент. Більше того, він був впроваджений в безліч інноваційних додатків. У тому ж дусі, Coder був випущений як платформа з відкритим кодом. Коди бібліотеки доступні для редагування на GitHub<sup>114</sup> і поставляється він з кількома простими проектами для новачків.

Coder розроблений, щоб працювати також разом з інструкціями від таких інших веб-сайтів, як Codecademy і Khan Academy. Якщо проекти будуть завершені, то користувачі зможуть розмістити їх через RPi на своїх власних веб-сайтах або поділитися ними з друзями електронною поштою.

Coder доступний як для Windows так і для Mac. Це всього 1,04 Гб завантаження. Хоча існують й інші методи використання Raspberry Pi як веб-сервера, але цей, здається, найпростіший у налаштуванні.

За підтримки Google і прозорості відкритих джерел, Coder має можливість розширити доступність не тільки для любителів DIY, але і для класних кімнат, а також для курсів спільнот. З великим потенціалом для зростання (за допомогою сприяння програмістів) і можливістю працювати з існуючими підручниками програмування, цей невеликий проект може перетворитися на великий ресурс для освіти.

Можна виділити три етапи підготовки та запуску Coder:



<sup>112</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1757-coder-a-simple-way-to-program-on-raspberry-pi>

<sup>113</sup> <http://googlecreativelab.github.io/coder/>

<sup>114</sup> <https://github.com/googlecreativelab/coder/>

## 1. Завантаження образу Coder для Raspberry Pi.

Якщо у вас вже є все необхідне для запуску Raspberry Pi, то тепер необхідно завантажити образ SD-карти Coder<sup>115</sup> і перенести його на карту SD. Це чимале завантаження (більше 1 Гб), тому може зайняти кілька хвилин.

Можна також отримати код Coder на Github<sup>116</sup>. Coder має відкритий вихідний код, тому кожен може допомогти зробити його кращим: усунути помилки, побудувати нові функції. Така допомога з розвитку Coder - найпростіший шлях для нових кодерів, щоб дізнатися, як створювати речі для вебу.

## 2. Створення SD-карти з Coder

Програма установки для Mac OS X вже входить і вона допоможе перенести образ диска на вашу SD-карту.

Якщо використовуєте ПК з Windows, то в першому розділі ми розглянули, як інсталювати образ. Наприклад, користувачі Windows можуть зробити це за допомогою Win32 Disk Imager, як ми це робили раніше при встановленні Rasbian.

## 3. Підключення до coder.local.

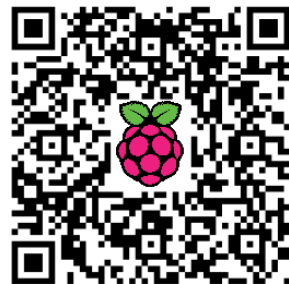
Для підключення до Coder, використовуючи комп'ютер з ОС Windows, спочатку потрібно встановити Bonjour Print Services<sup>117</sup> компанії Apple для Windows. Як тільки ця частина програмного забезпечення буде встановлена, ви зможете перейти на <http://coder.local> в своєму браузері.

## Працюємо з HTML в Google Coder

Розглянемо, як працювати з Google Coder на Raspberry Pi, що використовується як веб-сервер для створення веб-сайтів на ньому<sup>118</sup>.

Раніше ми вже розглянули, як завантажити образ Google Coder, записати його на SD-карту, щоб швидко встановити веб-сервер на Pi для вивчення веб-технологій.

Ви підключили Raspberry Pi до локальної мережі і можна скористатися комп'ютером або ноутбуком для підключення до нього. Ви побачите, що після підключення кабеля



<sup>115</sup> [http://storage.googleapis.com/coder-images/coder\\_v0.6.zip](http://storage.googleapis.com/coder-images/coder_v0.6.zip)

<sup>116</sup> <https://github.com/googlecreativelab/coder/>

<sup>117</sup> <http://support.apple.com/kb/DL999>

<sup>118</sup> <http://otins.com.ua/articles/tabid/211/EntryId/145/Default.aspx>

живлення деякі світлодіоди спалахнули; це означає, що ваш сервер Coder на Raspberry Pi завантажується.

1. Дайте йому кілька хвилин для завантаження, а потім у веб-браузері свого комп'ютера введіть:

```
http://coder.local/
```

Не забувайте про частину `http://`!

2. Можливо, вам доведеться ввести номер Coder, якщо в класі встановлено кілька серверів, наприклад, `http://coder01.local/` або `http://coder02.local/`.

3. Після підключення, можете побачити попередження екрану безпеки про сертифікат; все ОК, тому натисніть кнопку, щоб "перейти в будь-якому випадку" (proceed anyway). Тоді побачите екран, який просить ввести пароль. Використовуйте пароль, який створили під час встановлення Coder, наприклад, "raspberrypi".

### Робимо нову веб-сторінку Coder

На головній сторінці Coder побачите сітку з прямокутниками та назвами на них. Це проекти (також ще називаються додатками); деякі з них - веб-сторінки, а деякі ігри. Є деякі, які заздалегідь встановлені на Coder. Одним з проектів є "Hello Coder".

1. У лівому верхньому кутку екрану ви побачите зелений прямокутник зі знаком плюса. Натисніть піктограму, щоб почати створювати новий проект.

2. Дайте йому назву і оберіть колір. Ви збираєтеся зробити веб-сторінку, що містить відгук, тож дайте їй назву, яка відображає її зміст, таку як "My favourite item" ("Моя улюблена тема") (рис.6.11):

3. Після того, як обрали назву і колір, натисніть Create (Створити). Ви повинні перейти прямо в редагування нового проекту.

4. Коли створюєте новий сайт, то починаєте в "Code View" Coder. Він містить будівельні блоки, які зроблять ваш сайт. Натисніть на іконку, яка виглядає як око, у верхньому правому куті екрану. Вам буде показано, як виглядатиме веб-сторінка (рис.6.12).

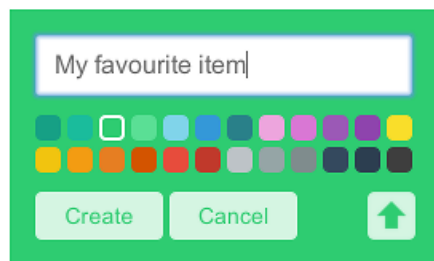


Рис.6.11 Параметри нового проекту

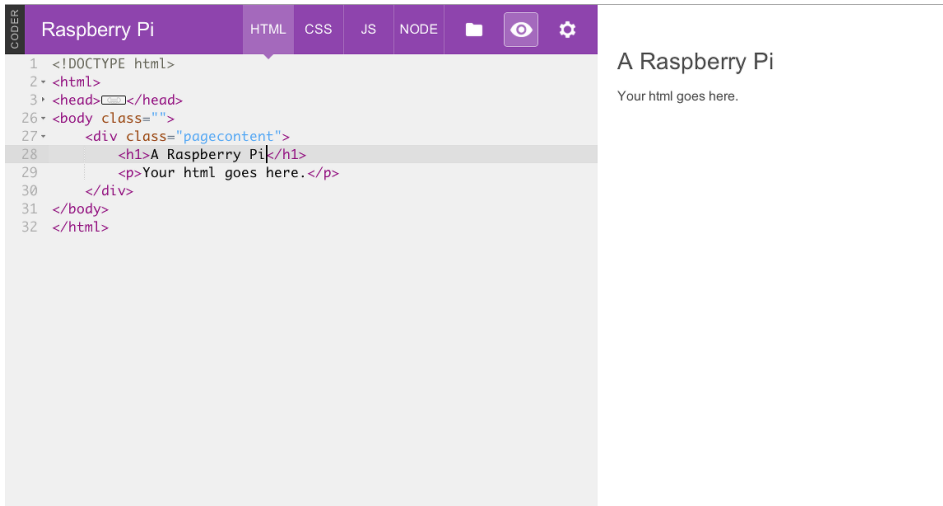


Рис.6.12 Вигляд нової сторінки

## Параметри проекту

Ви можете змінити колір, який використовували для свого нового проекту, і навіть додати своє ім'я, так щоб знали, що це ваша робота. Для цього натисніть на іконку гвинтика, який представляє параметри поруч із позначкою ока попереднього перегляду. Потім додайте своє ім'я і новий колір для свого значка проекту, якщо хочете (рис.6.13).

## Рубрики і заголовки

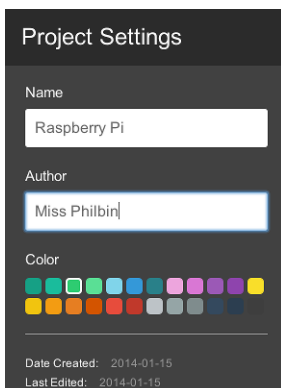


Рис.6.13 Налаштування проекту

Всі проекти Coder починаються вже з деякого коду в них. Наприклад, рядки, що починаються з коду з кутовими дужками, як `<word>`, називаються тегами.

Теги є елементами HTML, які визначають структуру та зміст сайту; вони дуже важливі. Всі теги мають відкриваючий тег і закриваючий тег. Закриваючий тег виглядає як `</tag>`. Коса риска на початку того, що закриває закриваючий тег. Всі теги повинні відкриватися та закриватися, бо інакше вони не будуть працювати.

1. Додайте інформацію про вашу улюблену тему, знайшовши тег `<h1>`; це тег для заголовка. Зараз заголовок виглядає так:

```
<h1>Hello World</h1>
```

2. Змініть текст між тегами в назві вашої теми або ТВ-шоу/гри.

3. Збережіть зміни, натиснувши на кнопку Save (Зберегти) у нижній частині екрана, і подивіться на вікно попереднього перегляду. Великий текст заголовка в даний час змінився; це тому, що ви змінили текст всередині тега.

### Інші способи відображення тексту

Заголовки важливі, але ви ж хочете мати різні стилі тексту при перегляді веб-сторінки. У цьому розділі ми розглянемо два інших способи відображення тексту.

1. Ви можете розбити рядок тексту таким чином, що буде більше, ніж по одному рядку. Наступна назва трохи занадто довга:

```
<h1>My Raspberry Pi Review by Miss Philbin</h1>
```

Автор може використовувати тег `<br>`, щоб розбити текст на два рядки, як це:

```
<h1>My Raspberry Pi Review<br>by Miss Philbin</h1>
```

2. Перервіть ваш заголовок таким чином, щоб він був у двох рядках. Потім натисніть на Save (Зберегти) і подивіться, як це виглядає у вікні попереднього перегляду, натиснувши на зображення ока.

3. Потім змініть тег абзацу під заголовком, який виглядає так:

```
<p>Ваш html розміщується тут.</p>
```

Видаліть текст між тегами і замініть його з вашим введенням тексту про свою улюблену тему. Наприклад:

```
<p>Raspberry Pi є невеликий комп'ютер і моя улюблена річ.</p>
```

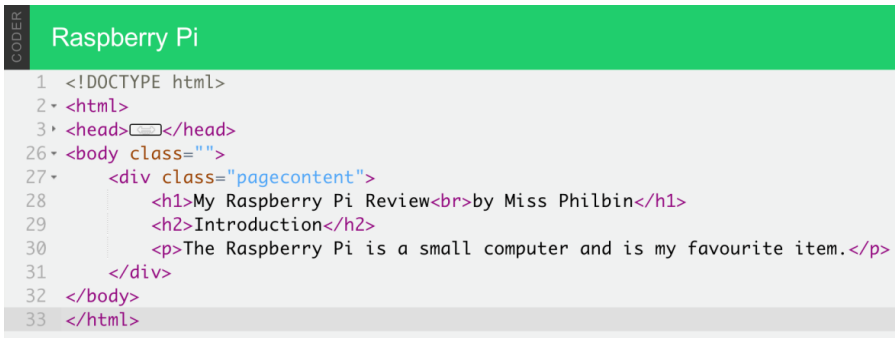
4. Тепер збережіть це і перегляньте.

5. Над вашим параграфом введення ви можете додати текст, щоб він був як підзаголовок. Це може бути важливий текст, який не настільки важливий, як назва.

Додамо тег `<h2>`. Це заголовок як і `<h1>`, але не буде настільки помітним. Тег `<h2>` повинен читати Introduction (Введення).

6. Не забудьте зберегти і переглянути сторінку, щоб побачити, як вона виглядає.

Код повинен виглядати приблизно так (рис.6.14):



```

1 <!DOCTYPE html>
2 <html>
3 <head></head>
26 <body class="">
27 <div class="pagecontent">
28   <h1>My Raspberry Pi Review<br>by Miss Philbin</h1>
29   <h2>Introduction</h2>
30   <p>The Raspberry Pi is a small computer and is my favourite item.</p>
31 </div>
32 </body>
33 </html>

```

Рис.6.14 Код з тегами заголовків

## Горизонтальна лінія

Це завжди приємно - підкреслити заголовки на сторінці, щоб відокремити їх від основного тексту. Ви можете зробити це за допомогою простого тега!

Під основною заголовком, використовуючого тег `<h1>`, введіть:

```
<hr />
```

`hr` розшифровується як "Horizontal Rule" ("Горизонтальна лінія"), і це HTML-елемент, призначений для поділу контенту на сторінці (рис.6.15).

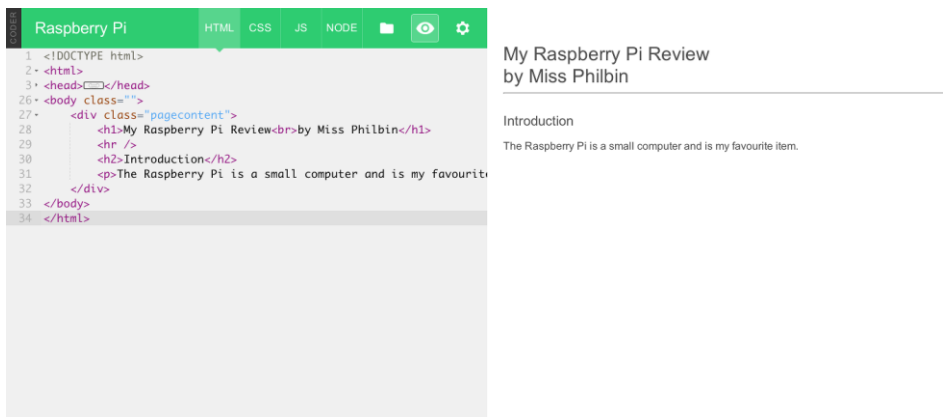
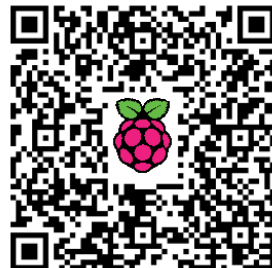


Рис.6.15 Код горизонтальної лінії

## Додавання зображень та іншої мультимедіа в Google Coder

В цьому розділі розглянемо, як працювати з мультимедіа в Google Coder на Raspberry Pi<sup>119</sup>, який використовується як веб-сервер для створення веб-сайтів на ньому.



### Додавання зображення

В даний час веб-сторінка може мати деякий текст, написаний з використанням різних стилів. Ви можете зробити її більш цікавою, додавши кілька фотографій з теки, яку переглядаєте.

1. Додайте новий рядок коду і новий тег `<hr />` під головним заголовком.

2. Між двома тегамі `<hr />`, додайте тег `<img src="" />`. Це тег, який не потребує тега закривання, подібно `<hr />`. Це має виглядати приблизно так:

```
<hr />
<img src="" />
<hr />
```

Зверніть увагу на частину `src=""`. Це атрибут тега `img`, який дозволить вам вказати зображення, яке хочете використовувати.

3. Перейдіть у верхню праву частину екрана. Натисніть на кнопку Media, яка виглядає як папка.

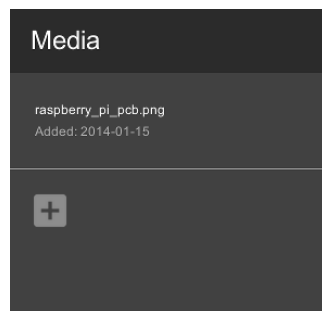


Рис.6.16 Додавання зображення

4. Натисніть на кнопку "+" (рис.6.16). Ви повинні отримати запрошення, щоб завантажити файл. Оберіть зображення за вашим вибором з місця, де воно збережене на вашому комп'ютері.

5. Після того, як зображення завантажиться, повинне з'явитися його ім'я і дата його завантаження. Провівши над ним мишею, ви побачите, що з'являються дві кнопки, "Paste to Code" and "Delete" ("Вставити в код" і "Видалити").

6. Поверніться на вкладку HTML, зберігаючи можливість висування медіа. Встановіть курсор між двома символами лапок (" ") в тезі `<img src="" />`, який ми створили. За допомогою курсору ще в такому положенні, поверніться до висувного ящика медіа і натисніть на кнопку paste to code (вставити в код). Частина тексту, який виглядає трохи як

<sup>119</sup> <http://www.otins.com.ua/articles/tabid/211/EntryId/146/Default.aspx>

адреса веб-сайту, повинна з'явитися між лапками - це шлях до файлу зображення. Все має виглядати приблизно так:

```

```

7. Збережіть код і погляньте на попередній перегляд версії. Чи з'явилося зображення?

8. Якщо ви додали своє власне зображення, то воно може бути занадто великим або занадто маленьким. Ви можете зменшити його і змінити його розмір в тезі `<img />`, додавши атрибути висоти і ширини. Наприклад:

```

```

Спробуйте додати висоту і ширину атрибути до тега `<img />`. Подивіться, що відбувається, коли ви використовуєте атрибут тільки або висоти, або ширини. Дійсно, отримуєте зображення роздавлене або рівномірно розтягнуте? Пограйте з різними вимірами, поки вам не сподобається, як воно виглядає на сторінці.

## Вкладення відео

Зображення та текст в центрі уваги багатьох сайтів. В даний час, відео-контент стає все більш важливим. Якщо у вас є доступ до такого сайту потокового відео, як YouTube або Vimeo, то чому б не вбудувати відео на веб-сторінку?

1. Знайдіть відео на сайті потокового відео, наприклад, на YouTube; як тільки ви знайшли те, яке хочете використовувати, натисніть на кнопку Share. Натисніть на embed, і ви побачите шматок HTML коду, який починається з `<iframe>`. Цей тег дозволяє вставляти контент, наприклад, відео на веб-сторінку і повідомлення в блозі.

2. Скопіюйте код за допомогою клавіш CTRL+C і поверніться на сторінку проекту Coder.

3. Вставте код під горизонтальною лінією або заголовком за допомогою поєднання клавіш Ctrl+V. Код повинен виглядати наступним чином (рис.6.17):

```
<h1>My Raspberry Pi Review<br>by Miss Philbin</h1>
<hr />

<hr />
<h2>Introduction</h2>
<br>
```

```
<iframe width="560" height="315"
src="//www.youtube.com/embed/jr6IVLZmp28?list=PL455284BB15D6E8F0
" frameborder="0" allowfullscreen></iframe>
<p> Raspberry Pi являє собою невеликий комп'ютер і моя улюблена
річ. Це відео, яке пояснює, що таке Raspberry Pi і як його
вперше створити! </p>
```

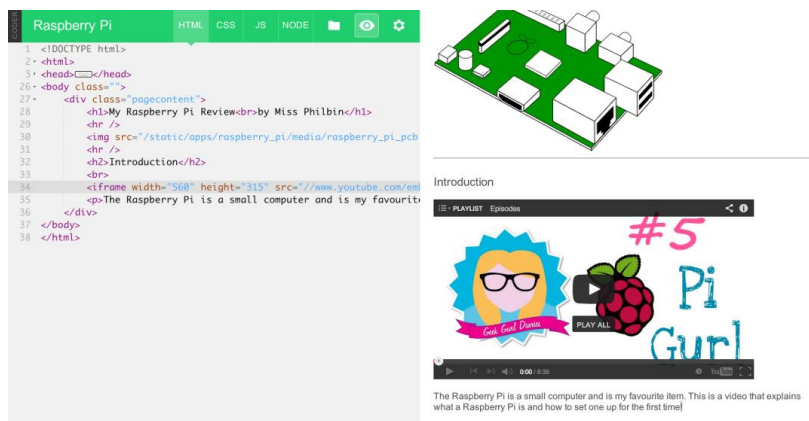


Рис.6.17 Додавання відео

### Додавання стилю за допомогою CSS в Coder

Досі ви використовували HTML для створення веб-сторінки. Це дозволило вам додати контент, наприклад, текст і зображення до нього. А тепер ви бажаєте, щоб вона виглядала набагато прохолодніше з кольором і різними шрифтами.

Текст на вашому сайті мав застосований до нього шрифт за замовчуванням, який трохи нудний, але можете змінити це, натиснувши значок CSS на панелі у верхній частині екрана (6.18).

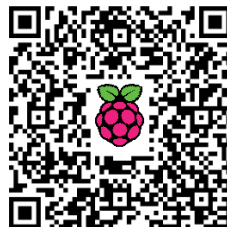


Рис.6.18 Меню вибору

*CSS розшифровується як "Cascading Style Sheets"* ("Таблиці каскадних стилів"). При натисканні на панелі ви побачите наступний код:

```
.pagecontent {
  padding: 24px;
}
```

CSS є те, що ми використовуємо до стилю таких елементів HTML, як текст. Так само, як теги HTML, він повинен бути написаний конкретним способом, щоб працював.

CSS виглядає по-іншому, ніж HTML, який ви досі використовували у своєму проєкті. Він використовує фігурні дужки і їх розміщення дуже важливе.

1. Змініть число поруч з padding з 24px на 300px.
2. Натисніть Save (Зберегти) у нижній частині сторінки.

3. Тепер перейдіть до попереднього перегляду. Ви побачите, що весь контент змістився. Це тому, що ми збільшили відступ або пробіл між внутрішньою частиною коробки і текстом, і зображенням.

Стилі CSS впливають на те, як виглядають речі в HTML. Це на цій вкладці, де будете робити, щоб ваш текст виглядав досить пристойно після зміни шрифтів і кольорів.

4. Змініть розмір внутрішніх полів на щось більш приємне для перегляду, наприклад, на таке, як 50.

```
.pagecontent {
  padding-left: 50px;
  padding-right: 50px;
}
```

## Будь щасливим зі шрифтами

Вибір шрифту дуже важливий при створенні такого цифрового контенту, як веб-сторінки. Ви повинні розглянути свою аудиторію. Краще використовувати шрифти, які легко читати, якщо у вас є текст, який ви хотіли б, щоб кожен зміг прочитати. Божевільні художні шрифти корисні, щоб зробити заголовки і виділити шматочки тексту. Почнемо, зробивши так, щоб заголовки <h1> відчули себе справжніми заголовками.

1. Перейдіть на вкладку CSS в верхній частині екрана.
2. Напишіть 'h1'. CSS змінюватиме стиль кожного тега h1 в HTML.
3. Додайте відкриваючу фігурну дужку після цього.

4. Натисніть двічі Enter на клавіатурі і додайте закриваючу фігурну дужку. Код повинен виглядати наступним чином:

```
h1 {
}
```

5. В порожньому рядку між цими фігурними дужках додайте властивість 'font-family' і зробіть його значення 'courier, monospace'. Це змінить шрифт для <h1> на Courier; якщо ж Courier не встановлений на вашому комп'ютері, то буде за замовчуванням використаний шрифт monospace.

```
h1 {
```

```
font-family: courier, monospace;  
}
```

6. Збережіть код і погляньте на нього у вікні попереднього перегляду. Заголовок повинен виглядати по-іншому.

7. Давайте завершимо укладання заголовка, додавши розмір шрифту, вагу шрифту (наприклад, напівжирний, курсив і т. д.) і висоту рядка тексту:

```
h1 {  
    font-family: courier, monospace;  
    font-size: 50px;  
    font-weight: bold;  
    line-height: 2em;  
}
```

8. Збережіть код і погляньте на нього у вікні попереднього перегляду. Заголовок повинен бути зараз великим і жирним, з трохи більшим простором навколо нього.

*Примітка:* "px" використовується для вказівки розміру екрану в пікселях, але 'em' використовується для вимірювання розміру в залежності від розміру шрифту. '2em' зробить висоту рядка в 2 рази від розміру шрифту (рис.6.19).

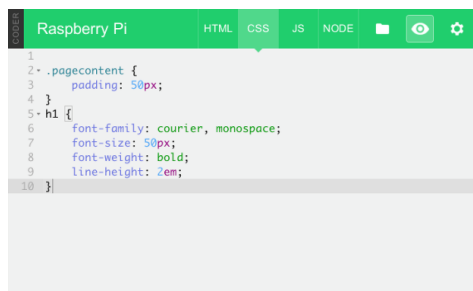


Рис.6.19 Зміна розміру шрифту

9. Ви можете використовувати будь-яке 'font-family', який вам подобається. Хороший спосіб знайти той, який ви любите, і його назву, це подивитися на шрифти в інших додатках, таких як Word, або Photoshop.

10. Змініть розмір і стилі шрифтів також для тексту <h2> і <p>!

### Інші забавні речі, які можна зробити зі шрифтами

Ви можете використовувати CSS, щоб застосувати всі види чудових стилів. Наприклад, чому б не зробити, щоб текст виглядав дещо розмитим, як це...

1. У панелі CSS введіть наступний код (6.20) в розділі `.pagecontent` у фігурних дужках і нижче в розділі `h1`:

```
.blur {
  color: transparent;
  text-shadow: 0 0 5px rgba(0,0,0,0.5);
}
```

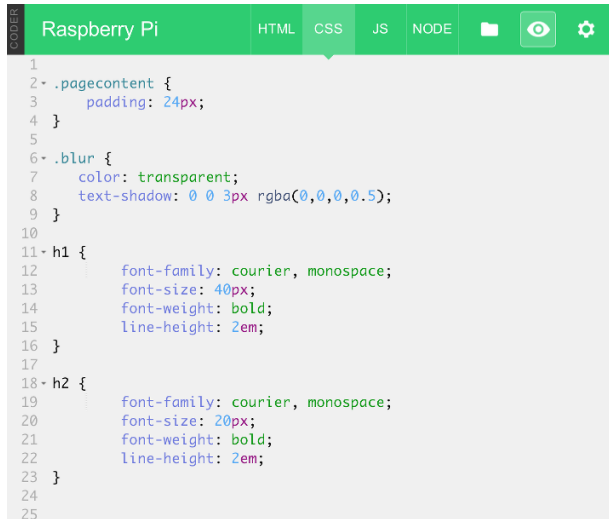


Рис.6.20 Код з різними шрифтами

2. Натисніть на панель HTML для перегляду HTML-коду; під `<div class="pagecontent">` введіть наступний рядок:

```
<div class="blur">
```

3. Пам'ятайте, що більшість тегів HTML мають тег відкривання і закривання. Додавання нового класу `div` додає новий тег відкривання. Тому вам потрібно додати тег закривання.

Помістіть цей тег закривання вище `<h2>Introduction</h2>` так:

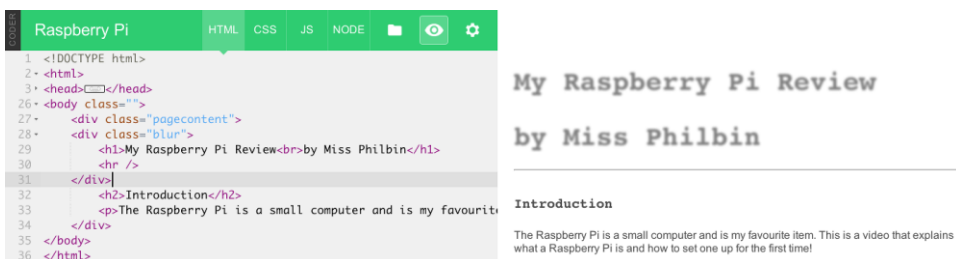


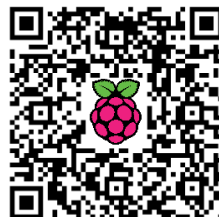
Рис.6.21 Сторінка з дещо розмитим шрифтом

Натисніть на кнопку Save (Зберегти) і перегляньте, як ваш шрифт заголовка виглядає зараз (рис.6.21)! Ви завжди можете видалити цей ефект, видаливши теги `<div class="blur"> і </div>`.

### Барви кольорів за допомогою Coder

В попередньому розділі ви почали додавати деякий стиль на свою сторінку, використовуючи CSS, концентруючись на стилі шрифтів і зміні таких їх атрибутів, як розмір і вага. В цьому розділі будемо спиратися на CSS, щоб навчитися змінювати кольори різних елементів веб-сторінки.

Іноді може важко зрозуміти, які кольори використовувати, що доповнюють один одного. Ви можете використовувати колірну схему дизайнера<sup>120</sup>, щоб отримати допомогу (рис.6.22). Знайдіть деякі кольори, які хотіли б використовувати для свого тексту.



### Додавання кольору в CSS

Весь текст, що використовується на веб-сторінці за замовчуванням, має чорний колір. Ви можете дати кожному типу тексту свій колір в панелі CSS.

1. Перейдіть на вкладку CSS в верхній частині екрана.

2. Перейдіть до частини коду для `h1`.

3. Додайте наступний рядок коду `color: magenta;` під іншими параметрами шрифту перед фігурною дужкою, подібно цьому:

```
h1 {  
    font-family: courier, monospace;  
    font-size: 40px;  
    font-weight: bold;  
    line-height: 2em;  
    color: magenta;  
}
```

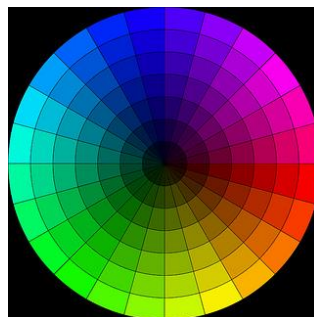


Рис.6.22 Схема для вибору кольору

*Примітка:* Ви повинні будете використовувати американську орфографію "`colour`" в коді для його роботи!

<sup>120</sup> <http://colorshemesigner.com/>

4. Тепер збережіть зміни і перегляньте, як вони виглядають, за допомогою кнопки попереднього перегляду. Чи був це колір, який ви очікували побачити? Що станеться, якщо ви заміните **magenta** на **blue**?

Як бачите, ви можете використовувати назви кольорів для застосування кольору у CSS. Це загальні значення, хоча і не завжди точний колір, який хочете. Якщо хочете трохи більше контролю над кольором, то повинні використовувати шістнадцяткові значення.

Значення Hex є кодами, які допомагають браузеру показувати колір. Вони виглядають так: **#FF0000**; це шістнадцяткове значення для червоного. Hex-коди мають довжину шість цифр і містять тільки цифри 0-9 і букви A-F.

Ось деякі інші приклад шістнадцяткових значень:

```
#00FF00 - тільки зелений
#0000FF - тільки синій
#C0392B - м'який червоний
#16A085 - м'який зелений
#3498D8 - м'який синій
#D977D4 - рожевий
#F1C40F - жовтий
#F39C12 - помаранчевий
#FFFFFF - білий
#000000 - чорний
#999999 - сірий
```

5. Змініть кольори всіх своїх текстових стилів, використовуючи шістнадцяткові коди від колірної схеми веб-сторінки дизайнера, яку ми вказали на попередній сторінці. Наведіть курсор миші на колір, який вам подобається, і вам буде видане шістнадцяткове значення для нього.

## Зміна кольору фону

Текст - не єдиний елемент веб-сторінки, для якого можете вибрати колір. В даний час фон за замовчуванням має білий колір. Давайте змінимо його, додавши стиль.

1. Переконайтеся, що у вас вибрана панель CSS і перейдіть до кінця коду.

2. Додайте наступне:

```
body {
  background-color:#b0c4de;
}
```

3. Натисніть кнопку Save (Зберегти), а потім - на кнопку попереднього перегляду, щоб подивитися, що відбувається.

4. Якщо вам не подобається колір, то можете змінити його, змінивши шістнадцяткове значення.

5. Вам не потрібно тут закінчувати; ви також можете змінити колір фону тексту. Перейдіть до **h1** і додайте рядок **background-color:#6495ed**; після кольору тексту і перед фігурною дужкою закривання, як це (рис.6.23):

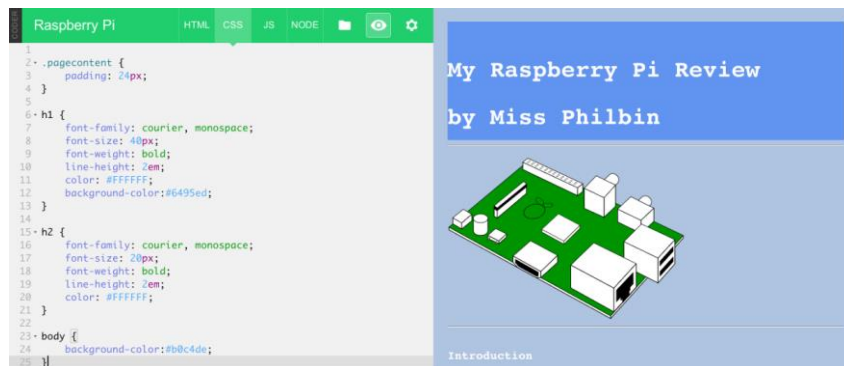


Рис.6.23 Зміна кольору фону

### Фонове зображення

Замість того, щоб використовувати один колір для фону, ви могли б використовувати замість нього зображення. Для цього вам потрібно знайти зображення, яке хочете використовувати, зберегти його де-небудь на своєму комп'ютері, а потім завантажити його на Coder за допомогою папки Media. Ми це робили раніше в одному з попередніх розділів.

1. Натисніть на кнопку Media, яка виглядає як зображення папки.
2. Натисніть на кнопку «+». Ви повинні отримати запрошення, щоб завантажити файл.
3. Виберіть зображення за своїм вибором з розташування на комп'ютері.

4. Після того, як зображення завантажиться, ви повинні побачити його назву і дату, коли з'явилось його завантаження.

5. Натисніть на панель CSS і замініть **background-color** на **background-image:url ("image.jpg")** наступним чином:

```
body {  
    background-image:url ("");  
}
```

6. Щоб додати розташування зображення, клацніть між (" "), потім з відкритої папки медіа натисніть на paste to code і URL зображення буде додане.

7. Збережіть свою роботу, а потім подивіться, як все працює, натиснувши на вікно попереднього перегляду. Ви повинні закінчити з кодом, який виглядає приблизно так (рис.6.24):

```
body {
  background-
image:url("/static/apps/raspberry_pi/media/Raspi_Colour_R.png");
}
```

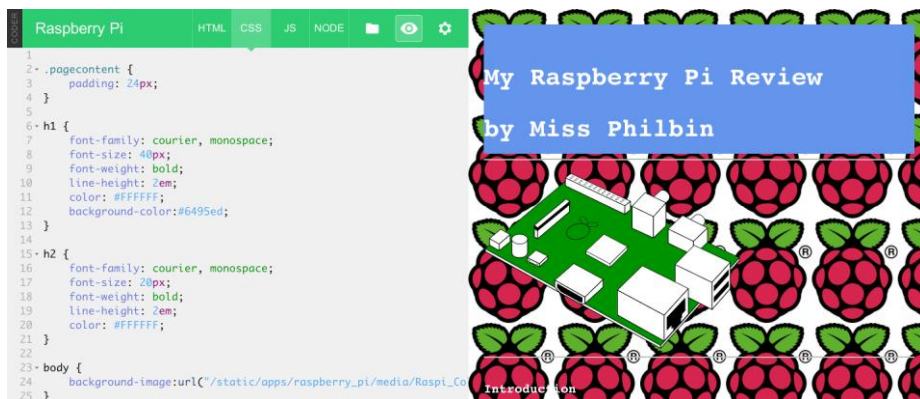


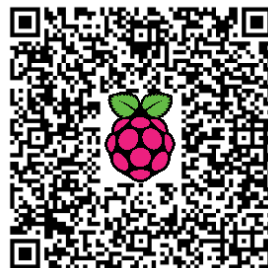
Рис.6.24 Фонове зображення

На цьому закінчимо короткі уроки з прикладами використання Google Coder на Raspberry Pi.

## Використання сервіса хмарного сховища від Copy



У лютому 2013 року компанія Barracuda Networks запустила свій сервіс хмарного зберігання Copy.com<sup>121</sup>, з кількома унікальними функціями, включаючи 256-бітове шифрування AES і підтримку декількох платформ, таких як Linux, Windows, Mac, і навіть Raspberry Pi.



Мати сервіс хмарного зберігання, який обслуговує користувачів Linux - це чудово, але мати його також навіть тому, хто працює на Raspberry Pi, - це фантастично!

Перший крок полягає в завантаженні клієнта Copy.com для Raspberry Pi допомогою **wget**:

```
wget http://copy.com/install/linux/Copy.tgz
```

Команда "**wget**" вже повинна бути встановлена на вашому Pi, але якщо це не так, то можете встановити її за допомогою:

<sup>121</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1819-using-a-raspberry-pi-with-the-cloud-storage-service-from-copy>

```
sudo apt-get install wget
```

Тепер у вас є файл з ім'ям `Copy.tgz` у домашньому каталозі. Ви можете розпакувати файл за допомогою:

```
tar zxvf Copy.tgz
```

Це створить папку з ім'ям `"copy"`, а в ній будуть три підпапки: `"armv6h"`, `"x86"`, і `"x86_64"`. Перша містить двійкові файли клієнта Copy для Raspberry Pi, друга містить клієнта Copy для 32-розрядних Linux на ПК, а третя - той же клієнт, але для 64-бітних Linux ПК.

Ми будемо використовувати двійкові файли Raspberry Pi, однак, використання клієнта Copy.com, по суті, те ж саме, що і на Linux ПК. Є два інструменти, передбачені на Barracuda Networks: `"CopyCmd"` і `"CopyConsole"`. Перший є загальною утилітою, яка дозволяє провести певні конкретні дії, як отримання файлу або отримання розширеного посилання на файл. Другий є версією командного рядка програми Copy, яка синхронізує локальну папку з хмарним сховищем.

### CopyCmd

Інструмент CopyCmd може бути досить складними, але прості операції виконуються легко. Ось як ви отримаєте список файлів, які зберігаються на Copy.com:

```
./CopyCmd Cloud -username= gary@example.com -password='mypass'
ls
```

Команда повинна бути запущена з каталогу `"armv6h"`. Якщо хочете запустити команду звідкись ще, то повинні вказати повний шлях до виконуваного файлу CopyCmd. `"username"` і `"password"` говорять самі за себе. Останній параметр `"ls"` повідомляє інструменту про перегляд списку файлів в хмарі. Використовуйте `"ls -r"`, щоб отримати рекурсивний лістинг.

Щоб завантажити файл з Copy.com скористайтеся:

```
./CopyCmd Cloud -username= gary@example.com -password='mypass'
get report1.doc report1.doc
```

Причина, за якою `"report1.doc"` повторюється, в тому, що перший примірник - це ім'я файлу на Copy.com, а другий - локальна назва файлу. В обох випадках, можна використовувати повні шляхи, наприклад, друга половина команди буде:

```
"get 'Reports/report1.doc' '/home/pi/reports1.doc'"
```

Щоб побачити список інших команд, підтримуваних CopyCmd, просто запустіть виконуваний файл без параметрів і побачите вбудовану довідкову інформацію.

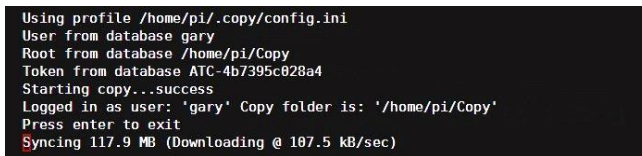
## CopyConsole

Інструмент CopyConsole зберігає синхронізованими папку з даними на Raspberry Pi і на Copy.com. Все, що завантажене на Copy.com з будь-якого клієнта або через веб, буде завантажене на Pi. Точно так само будь-які файли, додані в зазначену папку на Pi, будуть завантажені в Copy.com.

Додаток синхронізації працює у фоновому режимі і запускається так:

```
./CopyConsole -daemon -u= gary@example.com -p='mypass' -
root=/home/pi/Copy
```

Параметр "-root" визначає, який каталог на Pi буде синхронізований з онлайн-сховищем (рис.6.25). Якщо ви не вкажете пароль, то програма сама запитає у вас.



```
Using profile /home/pi/.copy/config.ini
User from database gary
Root from database /home/pi/Copy
Token from database ATC-4b7395c028a4
Starting copy...success
Logged in as user: 'gary' Copy folder is: '/home/pi/Copy'
Press enter to exit
Syncing 117.9 MB (Downloading @ 107.5 kB/sec)
```

Рис.6.25 Синхронізація каталогів

Варто зазначити, що якщо ви маєте велику кількість файлів, які зберігаються на Copy.com і активуєте агента синхронізації, то всі дані, які зберігаються на сервісі, будуть завантажені на ваш Pi. Оскільки багато користувачів, ймовірно, мають SD-карту лише 4 Гб або 8 Гб, то є небезпека, що карта швидко наповниться.

Ви також можете запустити програму на передньому плані, опустивши параметр "-daemon".

## CopyAgent

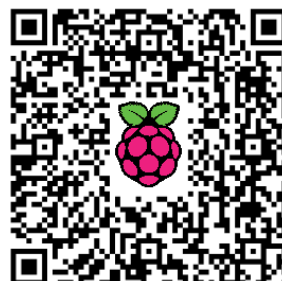
Для Linux PC (з Intel або сумісним процесором) є третя програма з назвою CopyAgent. Це графічна програма, яка запускається на робочому столі Linux. Для запуску клієнта знайдіть вкладену папку "x86" або "x86\_64" і двічі клацніть на "CopyAgent". Початкове налаштування вимагає ввести свої повноваження Copy.com, а потім підтвердити папку синхронізації. Після налаштування, клієнт залишається в лотку і синхронізує папки у фоновому режимі.

## 7. ВИКОРИСТАННЯ GPIO

### Як використовувати GPIO на Raspberry Pi (Wheezy)

Є два різних методи для запису і читання з периферійних пристроїв на вбудованих системах, що використовують Linux<sup>122</sup>.

Перший з них - створюючи доступ через типи файлів до периферії в файловій системі, а другий - записуючи/читаючи базові адреси пам'яті, виділеної для GPIO або модуля SoC, з використанням покажчиків. Ці комірки пам'яті можна знайти в даних для BCM2835 у випадку Raspberry Pi.



*Примітка:* Призначення GPIO для Raspberry Pi відрізняється у версіях 1 і 2 (рис.7.1).



Рис.7.1 Позначення виводів GPIO

Не використовуйте рівні напруги, що перевищують 3.3В, бо Raspberry Pi не підтримує 5В і не має захисту від перенапруги.

<sup>122</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1783-how-to-use-gpios-on-raspberry-pi>

## Перший метод (файлова система)

*Примітка:* Використовуйте обліковий запис адміністратора (SuperUser) або команду `sudo` перед будь-якою командою для роботи з виводами GPIO.

Підключіть світлодіод через резистор між GPIO11 і GND (рис.7.2).

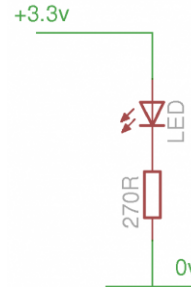
### Створення файлового доступу до GPIO за допомогою консольних команд

Якщо ви запишете у файл `./export` в підкаталозі `/sys/class/gpio/`, то система створить файл зі структурою GPIO відповідно до входу. У нашому випадку ми хочемо створити доступ для запису безпосередньо в GPIO11 з метою керування світлодіодом.

Створимо файл доступу GPIO:

```
echo 11 > /sys/class/gpio/export
```

Рис.7.2 Підключення світлодіода



Налаштовуємо напрямок передавання виводу (вхід/вихід):

```
echo out > /sys/class/gpio/gpio11/direction
```

Записуємо значення 1 для включення світлодіода за допомогою GPIO11:

```
echo 1 > /sys/class/gpio/gpio11/value
```

**Тепер ваш світлодіод повинен світитися** (рис.7.3):

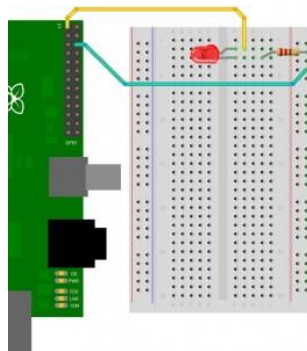


Рис.7.3 Вмикання світлодіода

Запишемо значення 0, щоб погасити світлодіод, підключений до GPIO11:

```
echo 0 > /sys/class/gpio/gpio11/value
```

**Тепер ваш світлодіод не повинен горіти!**

Видаляємо створений GPIO11:

```
echo 11 > /sys/class/gpio/unexport
```

Ви можете організувати доступ до GPIO за допомогою python або іншої мови програмування. Ми написали сценарій на python, щоб показати, як з ним працювати. Завантажте сценарій за посиланням<sup>123</sup>.

Відкрийте новий термінал і виконайте сценарій:

```
wget https://sites.google.com/site/semilleroadt/raspberry-pi-
tutorials/gpio/ADT_blink.py
python ADT_blink
```

Сценарій запросить ввести, до якого виводу приєднаний діод і скільки разів він повинен блимнути:

- Введіть номер GPIO, через який хочете керувати діодом: 11
- Введіть число разів, скільки б ви хотіли блимнути, наприклад: 10

Тепер світлодіод через GPIO11 блимає як новорічна ялинка :).

Відкрийте сценарій python за допомогою текстового редактора і вивчіть всі рядки (спробуйте зрозуміти, що являє собою простий сценарій, який використовується для доступу до файлів).

### Другий метод (покажчики)

Якщо хочете читати/писати в GPIO з використанням покажчиків, то найкраще встановити бібліотеку bcm2635, а щоб зрозуміти, як все працює, прочитайте довідник bcm2835<sup>124</sup>, починаючи з розділу 1.2.2.

Встановлення бібліотеки bcm2835:

```
wget http://www.open.com.au/mikem/bcm2835/bcm2835-1.15.tar.gz
tar zxvf bcm2835-1.15.tar.gz
./configure
Make
make check
make install
```

---

<sup>123</sup> [https://0c048a3a-a-62cb3a1a-s-sites.googlegroups.com/site/semilleroadt/raspberry-pi-tutorials/gpio/ADT\\_blink.py](https://0c048a3a-a-62cb3a1a-s-sites.googlegroups.com/site/semilleroadt/raspberry-pi-tutorials/gpio/ADT_blink.py)

<sup>124</sup> <http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

Тепер можете просто використовувати цю бібліотеку для доступу до GPIO. Бібліотека використовує покажчики щоб писати/читати з GPIO або змінювати значення в регістрах HW, модифікувати функції кожної периферії (ШІМ-модулів, UART, тощо).

Завантажте за посиланням<sup>125</sup> приклад коду, написаний на С.

Відкрийте термінал і використайте gcc для компіляції коду:

```
wget https://sites.google.com/site/semilleroadt/raspberry-pi-
tutorials/gpio/main.c
gcc -o main -l rt main.c -l bcm2835
```

Виконайте скомпільований код: `./main`

Тепер ваш світлодіод, підключений до GPIO11, блимає як і в прикладі з сценарієм на python.

Відкрийте код С за допомогою текстового редактора і вивчіть всі його рядки.

Тепер питання: який спосіб кращий, щоб отримати доступ до GPIO на Linux? Відповідь тут<sup>126</sup>.

## Широтно-імпульсний модулятор (ШІМ)

Скористаємося проектом WiringPi - бібліотекою, яка включає в себе додаток для простого доступу до GPIO.

Вона дозволяє налаштувати апаратні модулі для спеціальних виводів ШІМ, а також використати програмне забезпечення для реалізації ШІМ на інших виводах.

Встановимо WiringPi (WiringPi використовує git - систему керування вихідним кодом):

```
sudo apt-get install git-core
```

Завантажте або "клонуйте" проект WiringPi і побудуйте його:

```
git clone git://git.drogon.net/wiringPi
./build
```

Після завантаження можете оновити до останньої версії:

```
cd wiringPi
git pull origin
./build
```

Для того, щоб використовувати додаток WiringPi, треба знати призначення контактів, пов'язаних з ним, які пояснюються на сайті<sup>127</sup>.

<sup>125</sup> <https://0c048a3a-a-62cb3a1a-s-sites.googlegroups.com/site/semilleroadt/raspberry-pi-tutorials/gpio/main.c>

<sup>126</sup> <http://codeandlife.com/2012/07/03/benchmarking-raspberry-pi-gpio-speed/>

<sup>127</sup> <https://projects.drogon.net/raspberry-pi/wiringpi/pins/>

### Використання модуля HW (hardware) для ШІМ.

Підключіть світлодіод за допомогою резистора між GPIO18 і GND. (Вивід 1 для WiringPi)

Зверніться до "довідкової сторінки" ("man page") в щойно встановленій програмі WiringPi під назвою "gpio":

```
man gpio
```

Зверніть увагу, що можете налаштувати вивід, щоб він був входом, виходом, ШІМ (широтно-імпульсної модуляції), up, down або мав три стани.

Відповідно до цього, налаштуйте GPIO18 (WiringPi вивід 1) для режиму HW ШІМ за допомогою командної оболонки:

```
gpio mode 1 pwm
```

Запишіть значення в модуль ШІМ (від 1 до 1023):

```
gpio pwm 1 500
```

Щоб видалити конфігурацію виводу, використовуйте:

```
gpio unexport 1
```

Щоб видалити всі конфігурації:

```
gpio unexportall
```

Не бійтеся використовувати програму GPIO для налаштування інших виводів як входних, так і вихідних. ШІМ доступний тільки для спеціальних функціональних виводів типу GPIO18 (WiringPi 1), а інші виводи ШІМ зайняті на 3,5 мм аудіо роз'єм.

### UART

Для того, щоб використовувати виділені виводи для UART на Raspberry Pi, спочатку вони повинні бути видалені з їх застосування за замовчуванням, яке вже налагоджене.

Для цього треба відредагувати `"/boot/cmdline.txt"` і `"/etc/inittab"`.

Можна зробити резервну копію цих файлів, якщо хочете повернутися пізніше до конфігурації за замовчуванням:

```
cp /boot/cmdline.txt /boot/cmdline.bak
cp /etc/inittab /etc/inittab.bak
```

Видаляємо параметри конфігурації `"console=ttyAMA0,115200"` і `"kgdboc=ttyAMA0,115200"` з конфігураційного файла `"/boot/cmdline.txt"` за допомогою редактора `nano`:

```
nano /boot/cmdline.txt
```

Коментуємо останній рядок у файлі `"/etc/inittab"`. Для цього поставте `'#'` перед `"T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100:"`:

```
nano /etc/inittab
```

Тепер контакти RXD (GPIO15) і TXD (GPIO14) доступні для загального використання UART.

Нижче наведений приклад програми використання виводів UART за допомогою Python і бібліотеки `pyserial`. Для встановлення `Pyserial` завантажте останню версію з [sourceforge.net](http://sourceforge.net)<sup>128</sup>.

Встановіть, запустивши `setup.py`:

```
python setup.py install
```

Відкрийте термінал Python (пам'ятайте, що завжди треба запускати під управлінням суперкористувача за допомогою `sudo` або `su`):

```
python
```

Для перевірки без необхідності під'єднання іншого пристрою ми просто підключимо виводи TXD і RXD Raspberry один до одного.

Виконайте наступні команди з терміналу python:

```
import serial
ser = serial.Serial("/dev/ttyAMA0")
ser.write("UART the Font")
read = ser.read()
print read
ser.close()
```

Команда друку повинна вивести рядок, відправлений за допомогою команди запису. Для конфігурування порта UART, прочитайте інструкцію<sup>129</sup>.

Встановіть `"minicom"` для емуляції терміналу:

```
sudo apt-get install minicom
```

Ви можете використовувати `minicom` за допомогою команди:

```
minicom -b 9600 -o -D /dev/ttyAMA0
```

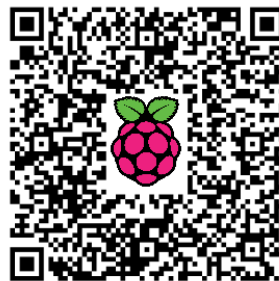
<sup>128</sup> <http://sourceforge.net/projects/pyserial/files/pyserial/>

<sup>129</sup> <http://pyserial.sourceforge.net/shortintro.html>

## Використання температурних датчиків з Raspberry Pi



Raspberry Pi не має виводів аналого-цифрового перетворювача (АЦП) в GPIO, і тому повинен бути використаний цифровий датчик температури<sup>130</sup>. Датчик, який вибирають в усьому світі, це Dallas DS18B20 - датчик температури 1-Wire (однопровідний). Він відносно дешевий, його дуже легко знайти, простий у використанні і показує температуру з точністю до  $\pm 0,5$  градуса у всьому діапазоні від  $-10$  до  $+85$  градусів Цельсія.



Для того, щоб він взаємодіяв з Raspberry Pi, нам необхідний лише один резистор - на  $4,7$  кОм, який діє як «підтягуючий» резистор, - і підключити все, як показано нижче (рис.7.4):

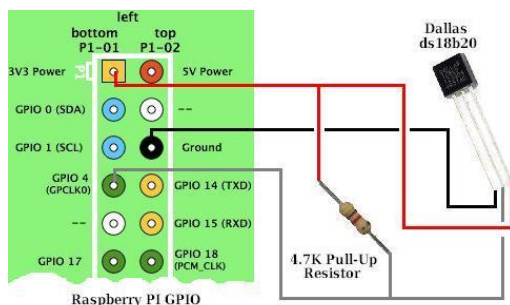


Рис.7.4 Підключення термодатчика

Увійдіть на Raspberry Pi і в командному рядку введіть команду:

```
sudo modprobe w1-gpio && sudo modprobe w1_therm
```

для завантаження модулів ядра однопровідних пристроїв зв'язку, що необхідні для використання датчика температури.

*Примітка:* Щоб уникнути необхідності ручного введення цієї команди щоразу, коли повторно завантажуєте Pi, відредагуйте з використанням nano або подібного редактора файл `/etc/modules` додаванням рядків `w1-gpio` і `w1_therm` в кінець файлу.

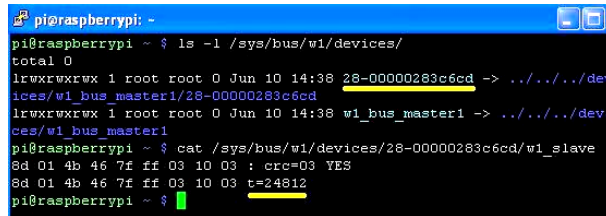
Тепер в командному рядку введіть:

```
ls -l /sys/bus/w1/devices/
```

<sup>130</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1838-using-temperature-sensors-with-raspberry-pi>

щоб побачити список пристроїв, підключених в даний момент до Raspberry Pi.

Ваш датчик температури з'явиться з адресою у форматі 28-00000xxxxxx. Нижче (рис.7.5) можете побачити, що адреса нашого датчика температури була 28-00000283c6cd. Кожен датчик температури DS18B20 має унікальну жорстко задану адресу, так що зможете при необхідності підключити декілька датчиків температури і, як і раніше,



```
pi@raspberrypi: ~$ ls -l /sys/bus/w1/devices/
total 0
lrwxrwxrwx 1 root root 0 Jun 10 14:38 28-00000283c6cd -> ../../../../dev
ices/w1_bus_master1/28-00000283c6cd
lrwxrwxrwx 1 root root 0 Jun 10 14:38 w1_bus_master1 -> ../../../../devi
ces/w1_bus_master1
pi@raspberrypi ~$ cat /sys/bus/w1/devices/28-00000283c6cd/w1_slave
8d 01 4b 46 7f ff 03 10 03 : crc=03 YES
8d 01 4b 46 7f ff 03 10 03 t=24812
pi@raspberrypi ~$
```

Рис.7.5 Адреса датчика температури

читати їх окремо.

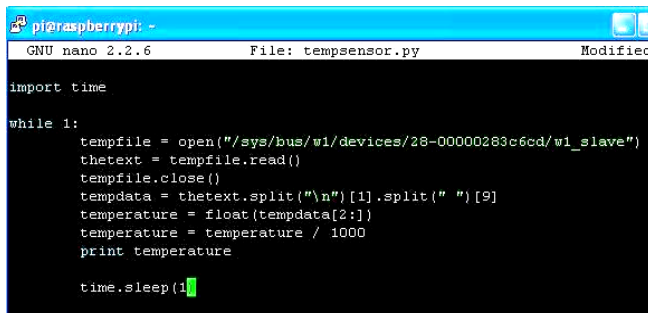
Знайдіть свій датчик температури, введіть наступну команду в командному рядку, щоб відкрити файл датчика і переглянути показання температури:

```
cat /sys/bus/w1/devices/28-00000283c6cd/w1_slave
```

(переконавшись, що підставили адресу свого датчика температури).

На рис.7.5 бачимо, що на виході цієї команди є два рядки даних. Наприкінці другого рядку після "t=" вказане числове значення. Це значення є помножена на 1000 температура датчика в градусах Цельсія. Тому, в нашому прикладі, де ми бачимо t=24812, показання температури буде  $24812/1000 = 24,812$  градусів за Цельсієм.

Читати температуру за допомогою командного рядка, як тут, не дуже зручно. Щоб вимірювати температуру через регулярні проміжки часу і, або відображати десь, або відкрити файл для подальшого аналізу, ми повинні написати сценарій.



```
pi@raspberrypi: ~$ nano tempsensor.py
GNU nano 2.2.6 File: tempsensor.py Modified

import time

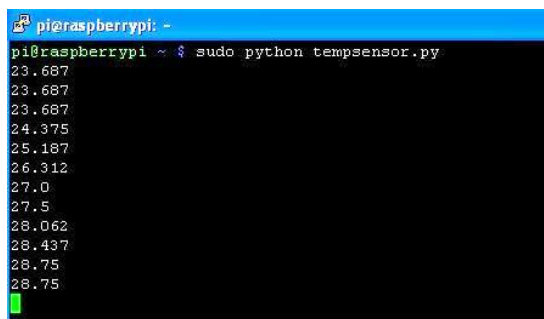
while 1:
    tempfile = open("/sys/bus/w1/devices/28-00000283c6cd/w1_slave")
    thetext = tempfile.read()
    tempfile.close()
    tempdata = thetext.split("\n")[1].split(" ")[9]
    temperature = float(tempdata[2:])
    temperature = temperature / 1000
    print temperature

    time.sleep(1)
```

Рис.7.6 Сценарій для вимірювання температури

Наш сценарій Python для вимірювання та відображення температури один раз в секунду показаний на рис.7.6.

Ми не будемо вдаватися в подробиці (якщо хочете вивчити Python, то перейдіть на сайт<sup>131</sup>). Даний сценарій один раз на секунду зчитує файл датчика температури, отриманий другий рядок розбиває на розділені пробілами елементи, знаходить десятий з цих елементів (деномінованих '9', бо '0' є першим елементом), пропускає перші два символи ("t=") цього елемента і перетворює те, що залишається (24812 в нашому прикладі), з рядка в числове значення, яке ділиться на 1000, щоб надати нам температуру, яка виводиться на екран (рис.7.7).



```
pi@raspberrypi: ~  
pi@raspberrypi ~$ sudo python tempsensor.py  
23.687  
23.687  
23.687  
24.375  
25.187  
26.312  
27.0  
27.5  
28.062  
28.437  
28.75  
28.75
```

Рис.7.7 Виведення значення температури

Скріншот (рис.7.7) показує запуск сценарію, коли датчик температури проходив нагрівання.

### Швидке відображення температури в командному рядку – Bash

Замість того, щоб писати сценарій Python для читання температури з датчика, можна написати один рядок команди, яка буде читати датчик, витягувати температуру від решти частини повернутих даних і обробляти результат за допомогою Bash в командному рядку Raspberry Pi.

Приклад досягнення цієї мети з використанням `sed` і `awk` (рис.7.8):

```
cat /sys/bus/w1/devices/28-00000283c6cd/w1_slave | sed -n  
's/^.*(t=[^ ]*)).*$/\1/p' | sed 's/t=/' | awk  
'{x=$1}END{print(x/1000)}'
```

Символ труби "|" використовується для передачі виводу однієї команди до наступної команди. Цей сценарій спочатку зчитує температуру з датчика, перш ніж повернути два рядки даних: в останній частині другого рядка, яка показує `t=xxxxxx`, `xxxxxx` є температурою в міліградусах Цельсієм. Потім ми використовуємо `sed`, щоб знайти в

<sup>131</sup> <http://www.learnpython.org/>

даних «слово», яке починається з 't=', а потім пересилаємо його до другої команди **sed**, яка видаляє 't='. Нарешті, чисельний результат передається далі до команди **awk**, яка ділить його на 1000 і відображає температуру в градусах Цельсія.

```

pi@reuk-office: ~$ cat /sys/bus/w1/devices/28-00000283c6cd/w1_slave | sed -n 's/^.*[t=[:space:]]*\\.[[:space:]]*/p' | sed 's/t=/' | awk '{x=$1}END{print x/1000}'
22.937
pi@reuk-office: ~$

```

Рис.7.8 Швидке відображення температури

Очевидно, що ви не хотіли б набирати цю довгу лінію щоразу, коли перевіряєте температуру свого датчика, тому можете створити псевдонім:

```
alias checktemp="cat /sys/bus/w1/devi.....x/1000) }"
```

Тепер при введенні **checktemp** в командному рядку, будуть запущені розглянуті команди.

Альтернативно, можете зберегти команди як сценарій оболонки. Для цього відкрийте текстовий редактор за допомогою команди:

```
sudo nano checktemp
```

який створить порожній файл з ім'ям **checktemp**. Введіть:

```
#!/bin/bash
```

в першому рядку, а потім введіть **cat /sys/bus...** і т.д. команди в другому рядку.

Натисніть **Ctrl+X**, щоб зберегти файл і вийти. Щоб зробити цей файл виконуваним (щоб ми могли запустити його) введіть:

```
chmod +x checktemp
```

і, нарешті, перемістіть його в каталог **/usr/bin**, де зберігається більшість виконуваних файлів:

```
mv checktemp /usr/bin/
```

Тепер, коли набиратимете **checktemp** в командному рядку, то буде запускатися ваш сценарій і відображатиметься зчитана з датчика температура.

## Підключення декількох датчиків температури з Raspberry Pi

Розглянемо, як можна під'єднати кілька датчиків температури до Raspberry Pi, що особливо корисне у випадку вимірювання різниці температур, коли необхідно мати два або більше датчиків (рис.7.9).

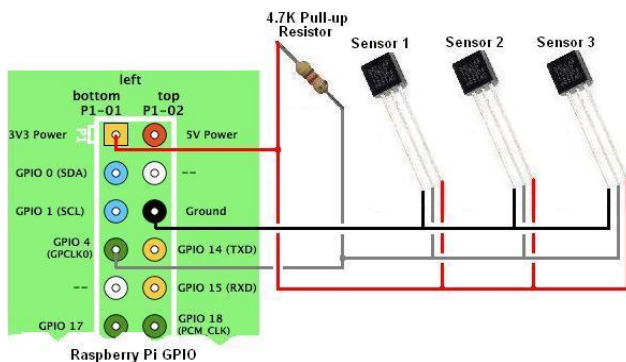


Рис.7.9 Підключення кількох датчиків

Вище наведена схема, як підключити три температурних датчики DS18B20 до GPIO виводів RPi. Відзначимо, що хоча на схемі існує декілька температурних датчиків для читання, ми й досі маємо лише три з'єднання з Raspberry Pi: 3,3В і 0В для підключення живлення датчиків, і один провід даних, який повертає зчитану температуру від всіх датчиків.

Читати показання температури з декількох датчиків через один провід можливо тому, що кожен датчик DS18B20 має унікальний серійний номер, закодований в нього при виготовленні, який Raspberry Pi може використати для їх ідентифікації.

При використанні декількох датчиків, перше, що треба зробити, це отримати серійний номер для кожного датчика і фізично маркувати їх, щоб знати, що є що, встановлюватимете їх у різних місцях своєї системи.

Дотримуйтесь інструкцій, наведених вище, про підключення одного датчика, щоб встановити кожен з них на своєму Raspberry Pi, і скористайтесь командою: `ls -l /sys/bus/w1/devices`, щоб знайти серійний номер для першого датчика. Потім додайте другий датчик і визначте його, повторно перевіривши підключені пристрої та відзначте його серійний номер. Повторіть цей процес, поки не підключите і не визначите всі датчики.

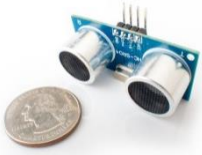
Щоб підтвердити, що все працює як треба, візьміть зчитану температуру з кожного датчика один за одним за допомогою команди:

```
cat /sys/bus/w1/devices/28-00000xxxxxxx/w1_slave
```

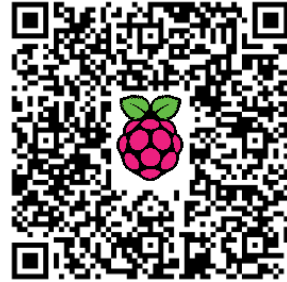
замінивши **xxxxxxx** правильним серійним номером для кожного датчика.

Подібним чином можна використовувати з Raspberry Pi й інші цифрові датчики.

## Ультразвуковий далекомір з використанням Python



Світлодіоди, зумери і перемикачі є найбільш поширеними предметами, які люди намагаються змусити взаємодіяти зі своїм Raspberry Pi.



Автор знайшов в Інтернеті щось трохи інше – це ультразвуковий вимірювальний модуль<sup>132</sup> (рис.7.10). Він дозволяє виміряти відстань до найближчої стіни або твердого предмета. Модуль легко купити, він дешевий і відносно легко узгоджується з інтерфейсом роз'єму GPIO.



Рис.7.10 Ультразвуковий вимірювальний модуль

Модуль HC-SR04 розміром з коробку сірників. Два перетворювачі надають йому характерний зовнішній вигляд. Він призначений для живлення від 5В, має один вхідний і один вихідний контакти. Модуль працює, посылаючи ультразвуковий імпульс в повітря і вимірюючи час, який необхідний, щоб імпульс повернувся назад. Це значення потім може використовуватися для обчислення відстані, пройденої імпульсом.

### Підключення до Pi

Підключити модуль легко. Просто під'єднайте виводи +5В (VCC) і землі (GND), відповідно, до виводів 2 і 6 на роз'ємі GPIO Pi.

<sup>132</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1840-the-ultrasonic-range-finder-using-python>

Вхідний контакт на модулі називається "Trig" (trigger), бо він використовується, щоб викликати відправку ультразвукового імпульсу. В ідеалі, він розрахований на сигнал 5В, але чудово працює з сигналом 3,3В від GPIO. Тому автор підключив вхідний контакт модуля безпосередньо до контакту 16 (GPIO23) на своєму роз'ємі GPIO.

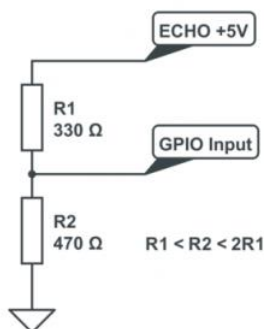


Рис.7.11 Подільник напруги

Ви можете використовувати будь-які виводи GPIO, які вам подобаються, на своєму Pi, але тоді треба буде відкорегувати посилання і змінити відповідно сценарій Python.

Вихід модуля називається "Echo" і вимагає трохи більше нашої уваги. На вихідному контакті низький рівень (0В), поки модуль не

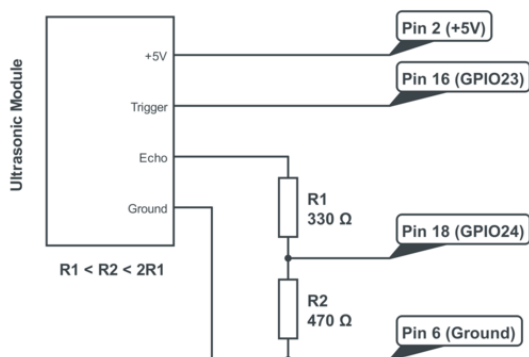


Рис.7.12 Коло ультразвукового модуля

виконує вимірювання відстані. Потім цей вивід перемикається у високий стан (+5В) на проміжок часу, поки імпульс не повернеться. Таким чином, наш сценарій повинен виміряти час, поки стан виводу залишається високим. Модуль використовує рівень +5В для "високого", але це занадто висока напруга для входів роз'єму GPIO, які розраховані лише на 3,3В. З метою забезпечення подачі на Pi напруги лише 3,3В, ми

використовуємо базовий подільник напруги, виконаний на двох резисторах (рис.7.11).

Якщо значення R1 і R2 однакові, то напруга розділиться на дві рівні частини. Це дало б нам 2.5В. Якщо R2 вдвічі перевищує значення R1, то отримаємо 3.33В, що чудово. Тому бажано, щоб значення R2 було між R1 і R1 x 2. У розглянутому прикладі схеми автор використав резистори 330 Ом і 470 Ом. Як альтернативу, можна використати значення 1 кОм і 1,5 кОм.

На рис.7.12 наведена схема остаточного кола. Автор обрав GPIO23 і GPIO24, але можете використовувати будь-який з 17-ти доступних контактів GPIO на роз'ємі GPIO. Тільки не забудьте відрегулювати сценарій.

## Сценарій на Python

У нашому прикладі сценарій виконує три виміри і обчислює середнє значення. Результат відображається і через секунду робиться ще одне усереднення. Це дозволяє використовувати сценарій для постійного вимірювання відстані і її виведення в командному рядку.

Ось цей сценарій:

```
#!/usr/bin/python
#-----+
#|R|a|s|p|b|e|r|r|y|P|i|-|S|p|y|.|c|o|.|u|k|
#-----+
#
# ultrasonic_2.py
# Вимірювання відстані з використанням ультразвукового модуля
# в циклі.
#
# Автор : Matt Hawkins
# Дата   : 28/01/2013
#
# -----
# Імпорт необхідних бібліотек Python
# -----
import time
import RPi.GPIO as GPIO

# -----
# Визначення деяких функцій
# -----

def measure():
    # Ця функція вимірює відстань
    GPIO.output(GPIO_TRIGGER, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)
```

```
start = time.time()

while GPIO.input(GPIO_ECHO)==0:
    start = time.time()

while GPIO.input(GPIO_ECHO)==1:
    stop = time.time()

elapsed = stop-start
distance = (elapsed * 34300)/2

return distance

def measure_average():
    # Ця функція робить 3 вимірювання і
    # повертає середнє значення.
    distance1=measure()
    time.sleep(0.1)
    distance2=measure()
    time.sleep(0.1)
    distance3=measure()
    distance = distance1 + distance2 + distance3
    distance = distance / 3
    return distance

# -----
# Основний сценарій
# -----

# Використання параметрів BCM GPIO
# замість фізичного номера виводу
GPIO.setmode(GPIO.BCM)

# Визначення GPIO для використання на Pi
GPIO_TRIGGER = 23
GPIO_ECHO     = 24

print "Ultrasonic Measurement"

# Встановлення виводів як вихідний і вхідний
GPIO.setup(GPIO_TRIGGER,GPIO.OUT)  # Trigger
GPIO.setup(GPIO_ECHO,GPIO.IN)      # Echo

# Встановлення trigger в хибне (Низький рівень)
GPIO.output(GPIO_TRIGGER, False)

# Запис основного змісту в блок перевірки, щоб ми могли
# спіймати натиснення користувачем CTRL+C і запустити
# функцію скидання GPIO. Це також допоможе уникнути
# отримання користувачем багатьох повідомлень
```

```
# про незначні помилки.
try:

    while True:

        distance = measure_average()
        print "Distance : %.1f" % distance
        time.sleep(1)

except KeyboardInterrupt:
    # Користувач натиснув CTRL+C
    # Скидання налаштувань GPIO
    GPIO.cleanup()
```

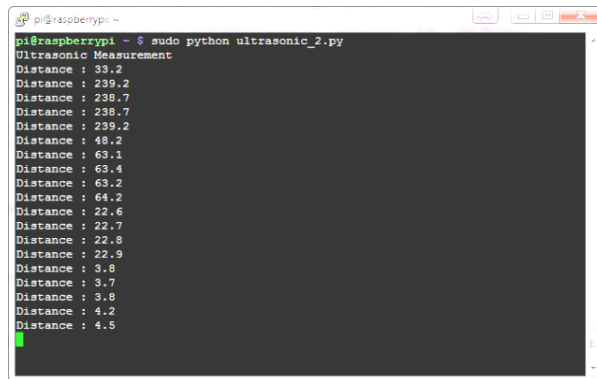
Даний сценарій також можна завантажити безпосередньо на свій Pi, використовуючи такий командний рядок:

```
wget https://bitbucket.org/MattHawkinsUK/rpispymisc/raw/master/python/ultrasonic_2.py
```

Потім сценарій можна запустити за допомогою:

```
sudo python ultrasonic_2.py
```

Результати виконання виглядають приблизно так (рис.7.13):



```
pi@raspberrypi: ~ $ sudo python ultrasonic_2.py
Ultrasonic Measurement
Distance : 33.2
Distance : 239.2
Distance : 238.7
Distance : 238.7
Distance : 239.2
Distance : 48.2
Distance : 63.1
Distance : 63.4
Distance : 63.2
Distance : 64.2
Distance : 22.6
Distance : 22.7
Distance : 22.8
Distance : 22.9
Distance : 3.8
Distance : 3.7
Distance : 3.8
Distance : 4.2
Distance : 4.5
```

Рис.7.13 Результати виконання сценарію

Очевидно, що результати вимірювання залежатимуть від того, на що ви вказуєте своїм датчиком!

## Точність вимірювання

Деякі моменти, які впливають на точність:

- Точність вимірювання відстані залежить від часу. Python під Linux не є ідеальним для точної синхронізації, але загальні

вимірювання на ньому будуть працювати добре. Для підвищення точності вам треба звернути увагу на використання C замість нього.

- Коли GPIO налаштований, то модулю потрібен деякий час, перш ніж він готовий виконати перше читання, тому автор додав затримку 0,5 с на початку сценарію.
- Датчики мають широкий кут чутливості. У метушливому навколишньому середовищі ви можете отримати більш короткі відстані через наявні об'єкти з боків модуля.
- Вимірювання працює приблизно до 2 см. Нижче цієї межі вимірювання може дати дивні результати.
- Якщо торкнутися ультразвукового перетворювача де-небудь, то результати будуть непередбачуваними.

Якщо хочете поекспериментувати з цим пристроєм зі сценарієм на C, то відвідайте сторінку<sup>133</sup>.

Також є порівняння реалізацій на Python і C<sup>134</sup>.

### Як керувати GPIO з пульта дистанційного керування

Зазвичай, віддалене керування використовується для управління ТВ-картою або XMBC, однак воно також забезпечує хороший інтерфейс для управління GPIO<sup>135</sup> на Raspberry Pi.

В нашому прикладі використаємо інфрачервоний пульт дистанційного керування для управління кількома світлодіодами, підключеними до деяких виводів GPIO на Raspberry Pi.

Adafruit має Mini Remote і ІЧ-датчик (рис.7.14), які ідеально підходять для цього.

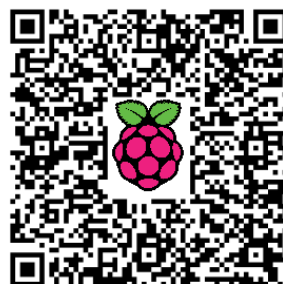


Рис.7.14 Пульт Mini Remote і ІЧ-датчик

<sup>133</sup> [http://rasathus.blogspot.co.uk/2012/09/ultra-cheap-ultrasonics-with-hy-srf05\\_27.html](http://rasathus.blogspot.co.uk/2012/09/ultra-cheap-ultrasonics-with-hy-srf05_27.html)

<sup>134</sup> <http://www.raspberrypi-spy.co.uk/2012/12/ultrasonic-distance-measurement-using-python-part-1/>

<sup>135</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1839-how-to-control-gpio-on-raspberry-pi-with-infrared-remote-control>

## Підключення ІЧ-датчика до Raspberry Pi

Підключити ІЧ-датчик до Raspberry Pi дуже просто, оскільки на датчику є тільки 3 контакти: GND (земля), 3В і Output (вихід). Вихід ми підключимо до GPIO 18. Ви можете обрати інший вивід, просто прийміть це до відома, коли буде треба вказати вивід при установці LIRC.

Ми приєднаємо три світлодіоди до GPIO 23, 24 і 25, а катоди кожного світлодіода через опори 270 Ом приєднаємо до «земляного» виводу GND GPIO, як показано на схемі (рис.7.15):

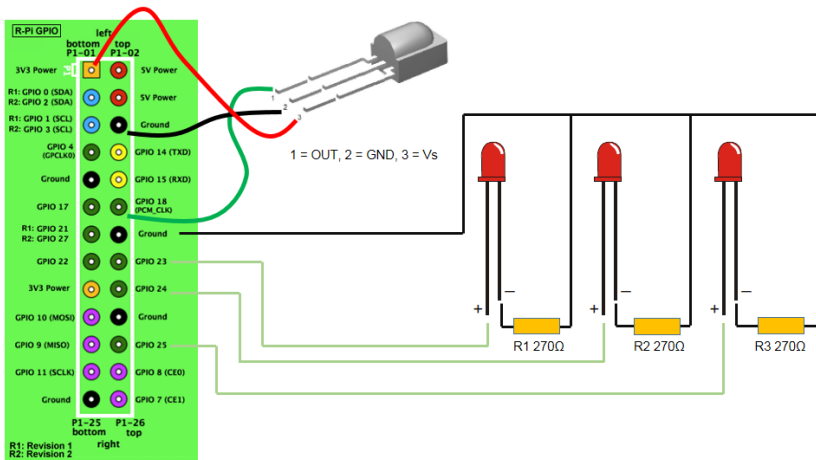


Рис.7.15 Керування трьома світлодіодами

## Встановлення LIRC

LIRC - пакет, який дозволяє декодувати інфрачервоні сигнали багатьох (але не всіх) найчастіше використовуваних пультів дистанційного керування. LIRC запускається як демон (служба), який буде декодувати ІК-сигнали, отримані від драйверів пристроїв, і надавати інформацію на роз'ємі. Далі ми написали програму для користувача, щоб він контролював вхідні події на цьому роз'ємі за допомогою клієнтської бібліотеки LIRC.

1. Нам необхідно встановити LIRC і клієнтські бібліотеки:

```
sudo apt-get install lirc liblircclient-dev
```

2. Тепер треба додати два рядки нижче до `/etc/modules`. Це дозволить запуск модулів при завантаженні. Вказаний нижче вивід 18 буде використовуватися, щоб через нього прийняти вихідний сигнал ІЧ-датчика.

```
lirc_dev
lirc_rpi gpio_in_pin=18
```

3. Редагуємо `/etc/lirc/hardware.conf`, щоб він виглядав точно так, як показано нижче:

```
# /etc/lirc/hardware.conf
#
# Аргументи, які будуть використовуватися при запуску lircd
LIRCD_ARGS="--uinput"

# Не запускайте lircmd навіть якщо здається, що
конфігураційний файл хороший
# START_LIRCMD=false

# Не запускайте irexec, навіть якщо хороший конфігураційний
файл, здається, існує
# START_IEXEC=false

# Спробуйте завантажити відповідні модулі ядра
LOAD_MODULES=true

# Запуск "lircd --driver=help", щоб отримати список
підтримуваних драйверів
DRIVER="default"
# Зазвичай /dev/lirc0 є коректною установкою для систем з
використанням udev
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"

# Конфігураційні файли для замовчуванням для будь-якого
Вашого обладнання
LIRCD_CONF=""
LIRCMD_CONF=""
# /etc/lirc/hardware.conf
```

4. Перезавантажуємося

### Швидкий тест

Щоб виконати швидкий тест і побачити, чи працює LIRC, ми повинні зупинити демона LIRC і запустити `mode2`. `mode2` показує довжину імпульсів/проміжків інфрачервоних сигналів.

```
sudo /etc/init.d/lirc stop
mode2 -d /dev/lirc0
```

При натисканні кнопок на пульті дистанційного керування `mode2` виведе щось схоже на те, що показано нижче:

```
pulse 627
space 514
pulse 624
space 513
pulse 599
space 521
pulse 618
space 1668
pulse 589
space 532
```

### Запис ІЧ-кодів від пульта

`irrecord` допоможе нам виявити ІЧ-коди, які використовує пульт дистанційного керування, і надати допомогу в створенні файлу `conf`, який використовуватиметься з LIRC:

```
irrecord -d /dev/lirc0 ~/lircd.conf
```

Після запуску `irrecord` покаже докладні інструкції про те, як налаштувати ваш пульт. Проте, ось підказки:

1. При запиті, натискайте всі кнопки на пульті дистанційного керування, поки не попросять зупинитися.
2. Назначте заздалегідь певне ім'я для кожної кнопки: під час запуску `irrecord --list-namespaces` відобразиться список доступних імен.

Файл конфігурації для Adafruit MinRemote виглядає так:

```
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
#
# this config file was automatically generated
# using lirc-0.9.0-pre1(default) on Tue Oct  8 07:05:38 2013
#
# contributed by
#
# brand:                               /home/pi/lircd.conf
# model no. of remote control:
# devices being controlled by this remote:
#

begin remote

    name    /home/pi/lircd.conf
    bits    16
    flags   SPACE_ENC|CONST_LENGTH
```

```
eps          30
aeps         100

header       9006  4447
one          594  1648
zero         594   526
ptrail       587
repeat       9006  2210
pre_data_bits 16
pre_data     0xFD
gap          107633
toggle_bit_mask 0x0
```

begin codes

|                |        |
|----------------|--------|
| KEY_1          | 0x08F7 |
| KEY_2          | 0x8877 |
| KEY_3          | 0x48B7 |
| KEY_4          | 0x28D7 |
| KEY_5          | 0xA857 |
| KEY_6          | 0x6897 |
| KEY_7          | 0x18E7 |
| KEY_8          | 0x9867 |
| KEY_9          | 0x58A7 |
| KEY_0          | 0x30CF |
| KEY_DOWN       | 0xB04F |
| KEY_LEFT       | 0x10EF |
| KEY_UP         | 0xA05F |
| KEY_RIGHT      | 0x50AF |
| KEY_BACK       | 0x708F |
| KEY_ENTER      | 0x906F |
| KEY_SETUP      | 0x20DF |
| KEY_PAUSE      | 0x609F |
| KEY_PAUSE      | 0x807F |
| KEY_STOP       | 0x609F |
| KEY_VOLUMEUP   | 0x40BF |
| KEY_VOLUMEDOWN | 0x00FF |

end codes

end remote

Тепер замініть існуючий файл `conf` (який, швидше за все, порожній) новим, який щойно створили:

```
sudo cp lircd.conf /etc/lirc/lircd.conf
```

Перезавантажте LIRC:

```
sudo /etc/init.d/lirc restart
```

## Тест нового файлу конфігурації для віддаленого доступу

Ви можете використати `irw`, щоб протестувати нову конфігурацію. `irw` передає дані з роз'єма Unix на стандартний вихід `stdout`.

Ви повинні отримати на виході щось схоже на показане нижче.... Це було при натисканні клавіш 1, 2 і 3 на пульті керування.

```
irw
0000000000fd08f7 00 KEY_1
0000000000fd08f7 01 KEY_1
0000000000fd08f7 02 KEY_1
0000000000fd8877 00 KEY_2
0000000000fd8877 01 KEY_2
0000000000fd8877 02 KEY_2
0000000000fd48b7 00 KEY_3
```

## Створення програми для управління GPIO

Використаємо мову C, щоб написати просту програму для управління трьома світлодіодами, підключеними до GPIO.

Нам також буде потрібний `WiringPi` для контролю виводів, а також скористаємося клієнтською бібліотекою `LIRC` для отримання даних з ІЧ-датчика.

### По-перше, встановимо `WiringPi`:

```
sudo apt-get update
git clone git://git.drogon.net/wiringPi
cd wiringPi
git pull origin
./build
```

### Код

Наведений нижче код дозволить вам контролювати три світлодіоди, підключені до GPIO на Raspberry Pi. Ми будемо використовувати кнопки 1, 2 і 3 на числовій клавіатури від Mini Remote Adafruit, щоб вмикати і вимикати ці світлодіоди. Код включає в себе багато коментарів, які допоможуть вам зрозуміти, як він працює.

```
#include <wiringPi.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <lirc/lirc_client.h>
#include <time.h>
```

```
void flipLED (int led);
//Номери виводів WiringPi, які використовуються світлодіодами
#define LED1 4
#define LED2 5
#define LED3 6
#define ON 1
#define OFF 0

int main(int argc, char *argv[])
{
    struct lirc_config *config;

    //Таймер для кнопок
    int buttonTimer = millis();
    char *code;
    char *c;

    //Ініціалізація WiringPi і встановлення виводів WiringPi 4,5 і 6
    (GPIO 23, 24 і 25) для виводу. До цих виводів підключені діоди.

    if (wiringPiSetup () == -1)
        exit (1) ;
    pinMode (LED1, OUTPUT);
    pinMode (LED2, OUTPUT);
    pinMode (LED3, OUTPUT);

    //Ініціалізація LIRC. Вихід при збої
    if(lirc_init("lirc",1)==-1)
        exit(EXIT_FAILURE);

    //Читання конфігурації LIRC за замовчуванням в
    /etc/lirc/lircd.conf. Це конфігурація для пульта дистанційного
    керування.

    if(lirc_readconfig(NULL,&config,NULL)==0)
    {
        //Виконуємо поки роз'єм LIRC відкритий: 0=відкритий -
        1=закритий. Поки (lirc_nextcode(&code)==0)
        {
            //Якщо код = NULL, то це означає, що ніщо не повернулося з
            роз'єму LIRC,
            //тоді пропустить рядки нижче і почне цикл знову.
            if(code==NULL) continue;{
            //Переконайтеся, що існує пауза 400 мс до перевірки натискання
            кнопок.
            if (millis() - buttonTimer > 400){
            //Перевіряємо, чи з'являється рядок "KEY_1" в будь-якому місці
            рядка 'code'.
            if(strstr (code,"KEY_1")){
```

```

        printf("MATCH on KEY_1\n");
        flipLED(LED1);
        buttonTimer = millis();
    }
    else if(strstr (code,"KEY_2")){
        printf("MATCH on KEY_2\n");
        flipLED(LED2);
        buttonTimer = millis();
    }
    else if(strstr (code,"KEY_3")){
        printf("MATCH on KEY_3\n");
        flipLED(LED3);
        buttonTimer = millis();
    }
    }
}
//Необхідно звільнити код до наступного циклу
free(code);
}
//Звільняємо структури даних, пов'язані з config
lirc_freeconfig(config);
}
//lirc_deinit() закриває з'єднання з lircd і робить деякі
внутрішні очищення.
lirc_deinit();
exit(EXIT_SUCCESS);
}
void flipLED (int led)
{
//Якщо діод горить, то вимкнути його. В іншому випадку він
вимкнений і тому ми повинні включити його.

    if(digitalRead(led)==ON)
        digitalWrite(led, OFF);
    else
        digitalWrite(led, ON);
}

```

Компілюємо наведений вище код, за допомогою:

```
gcc -o example example.c -lwiringPi -llirc_client
```

Створюємо порожній файл `lircrc`, інакше `lirc` буде скаржитися:

```
touch /etc/lirc/lircrc
```

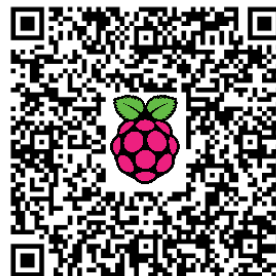
Запускаємо з: `sudo ./example`

От і все. Залишилось понатискувати клавіші 1, 2 і 3 на пульті керування.

### Як за допомогою Raspberry Pi виявити, хто був у твоїй кімнаті



Raspberry Pi настільки малий, що його можна легко приховати; що робить його ідеальним інструментом для різного роду подібних проектів.



Основна ідея полягає у використанні функції виявлення руху для запуску запису відео через модуль камери Raspberry Pi. Ви можете залишити Pi прихованим у своїй кімнаті і подивитися, хто спіймався на камеру, коли ви були відсутні<sup>136</sup>.

Для проекту, крім Raspberry Pi і всіх його стандартних периферійних пристроїв, нам ще необхідно:

- Модуль камери Raspberry Pi
- Модуль датчика руху PIR (рис.7.16)
- Перемички «гніздо-гніздо», хоча б три

#### Що таке модуль PIR?

PIR означає Passive Infra Red (пасивний інфрачервоний). Можливо, ви вже їх бачили раніше, оскільки вони сьогодні використовуються дуже часто. Їх найчастіше можна знайти в кутах кімнати, де встановлені системи охоронної сигналізації. Всі об'єкти, температура яких вище абсолютного нуля, випромінюють в інфрачервоному діапазоні. Інфрачервоні довжини хвиль не видимі людським оком, але вони можуть бути виявлені за допомогою електроніки всередині одного з цих модулів.



Рис. 7.16 Пасивний інфрачервоний датчик

<sup>136</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1837-how-to-use-raspberry-pi-identify-who-was-in-your-room>

Датчик розглядається як пасивний, так як він не передає якого-небудь сигналу для виявлення руху. Він пристосовується до структури інфрачервоного випромінювання у кімнаті, в якій знаходиться, а потім фіксує будь-які зміни. Будь-який об'єкт, що рухається в кімнаті, буде змінювати структуру інфрачервоного випромінювання, і викликані зміни будуть помічені модулем PIR.

Нам не потрібно знати його внутрішню конструкцію. Нас цікавлять лише три штирі на ньому, які можемо приєднати до контактів GPIO Raspberry Pi. Один контакт для 5 вольт, другий контакт для землі, а інший є контактом датчика (середній штирь на нашому Pi – зелений колір перемички). Цей вивід датчика прийматиме напругу, коли модулем PIR буде виявлений рух. Ми можемо побачити, що це відбувається, на Raspberry Pi і вжити відповідних заходів.

### Крок 1: Підключіть датчик руху PIR

Перед завантаженням свого Raspberry Pi, підключіть модуль PIR до Raspberry Pi (рис.7.17):

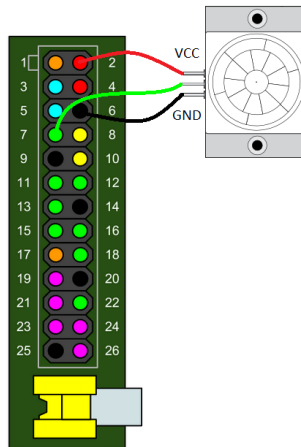


Рис.7.17 Підключення датчика руху

Зверніться до діаграми вище для нумерації виводів. Якщо уважно подивитися на виводи свого модуля PIR, то побачите трохи білого тексту на друкарській платі біля основи кожного з них. VCC - для підведення 5 вольт. Візьміть одну з перемичок «гніздо-гніздо» і підключіть вивід VCC до контакту 2 на Pi (червоний колір на діаграмі); це дозволить Pi подати 5 вольт живлення на модуль PIR. Використайте другу перемичку для підключення GND на модулі з виводом 6 на Pi (зафарбований чорним на схемі); цим замикаємо коло і дозволяємо протікання струму в модулі. Тепер те ж саме для вивода датчика OUT: ви можете використовувати будь-який з зелених контактів на Pi для цього. Автор збирається

використовувати вивід 7, так як це перший вивід загального призначення.

Тепер завантажте Pi і увійдіть.

*Примітка:* Якщо у вас інший модуль PIR, ніж зображений вище, то розташування контактів може бути також іншим; саме тому автор використав позначення VCC, GND і OUT.

### Крок 2: Перевірте датчик руху PIR

Ми збираємося використовувати мову програмування Python, щоб написати код, який виявлятиме рух і виводитиме деякий текст; ми можемо розширити програму, щоб пізніше повертати плату камери. При виявленні руху датчик руху PIR генерує напругу на його виході OUT, який ми підключили до виводу 7 GPIO на Pi. Таким чином, в нашому коді ми просто повинні постійно перевіряти вивід 7, щоб бачити, чи є на ньому напруга, чи ні.

Якщо на виводі є напруга, то ми називаємо це HIGH, а якщо немає, то називаємо його LOW.

Програма досить проста. Ми спочатку налаштуємо контакти Raspberry Pi GPIO, щоб дозволити використовувати вивід 7 як вхід; він може потім визначити, коли модуль PIR генерує напругу. Ми повинні постійно перевіряти вивід для будь-яких змін, тому для цього використаємо цикл `while True`. Це нескінченний цикл, тому програма буде працювати постійно, поки не зупинимо її за допомогою Ctrl+C.

Потім використаємо дві логічні змінні (`True` або `False` - істина чи неправда) для попередніх і поточних станів вивода, попередній стан, що є, а поточний стан буде попереднім через цикл. В середині циклу ми порівнюємо попередній стан з поточним станом, щоб визначити, коли вони відрізняються. Ми не хочемо отримувати ніякого повідомлення, якщо не відбулося жодних змін.

По-перше створіть порожній файл Python за допомогою команди:

```
nano pirtest.py
```

Введіть або скопіюйте і вставте код, наведений нижче:

```
import RPi.GPIO as GPIO
import time

sensorPin = 7

GPIO.setmode(GPIO.BOARD)
GPIO.setup(sensorPin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

prevState = False
currState = False
while True:
```

```
time.sleep(0.1)
prevState = currState
currState = GPIO.input(sensorPin)
if currState != prevState:
    newState = "HIGH" if currState else "LOW"
    print "GPIO pin %s is %s" % (sensorPin, newState)
```

Натисніть Ctrl+O, щоб зберегти, і Ctrl+X для виходу.

Наступним кроком зробіть файл Python виконуваним файлом, щоб потім змогли його запустити:

```
chmod +x pirtest.py
sudo ./pirtest.py
```

Якщо отримуєте повідомлення про помилку, `RuntimeError: No access to /dev/mem`, то це означає, що ви забули використати `sudo`. Ви повинні запускати програми, які звертаються до GPIO як `root` і команда `sudo` робить це для вас.

Якщо ви починаєте рухатися або розмахувати, то вивід датчика стає `HIGH`. Продовжуйте розмахувати, і він буде залишатися високим, а повернеться до `LOW` тільки, якщо збережете ще раз. Якщо побачите, що датчик поводить себе подібним чином, то все працює правильно. Якщо ні, то щось не так, і ви повинні вернутися та усунути проблеми.

```
GPIO pin 7 is HIGH
GPIO pin 7 is LOW
GPIO pin 7 is HIGH
```

Натисніть Ctrl+C, якщо хочете вийти.

На модулі PIR ви повинні побачити дві помаранчеві компоненти з гніздами, які відповідають хрестовій викрутці (рис.7.18). Вони називаються потенціометрами і дозволяють регулювати чутливість датчика і час виявлення. Автор пропонує налаштувати чутливість на максимум і час на одну хвилину, але вибір за вами.



Рис. 7.18 Потенціометри регулювання

### Крок 3: Налаштування модуля камери

Дотримуйтесь наведених в розділі 4 інструкцій, щоб налаштувати і протестувати плату камери Raspberry Pi. Зупиніться, як тільки успішно виконаєте деякі з наведених там прикладів команд.

Якщо ви ще не зробили цього, то перевірте роботу камери за допомогою наступної команди:

```
raspivid -t 0
```

Натисніть Ctrl+C, щоб вийти.

### Крок 4: Встановіть модуль `picamera` Python

Для управління платою камери за допомогою Python нам треба встановити модуль під назвою `picamera`. Використайте наступні команди, щоб зробити це в командному рядку Linux:

```
sudo apt-get update  
sudo apt-get install python-  
picamera
```

Це займе хвилину чи дві, щоб завершити.

### Крок 5: Програмуємо камеру для перегляду при русі

Тепер ми готові розширити нашу попередню програму, щоб дати їй можливість контролювати плату камери. Для початку, давайте просто зробимо нашу програму для відображення, що камера може побачити при виявленні руху, а запис у файл можемо встановити пізніше.

Зробіть копію попередньої програми і ми використаємо її для цього кроку:

```
cp pirtest.py pirCamera.py
```

Тепер виконайте наступну команду для редагування файлу:

```
nano pirCamera.py
```

Спочатку ми повинні додати команду `import picamera` зверху - це дозволяє програмі доступ до попередньо зробленого коду, який може контролювати плату камери. Потім оголошуємо камеру об'єктом `cam`, який забезпечує всі функції керування камерою, якими ми будемо використовувати. Тепер всередині циклу `while`, в якому ми виводили повідомлення `HIGH` або `LOW`, можемо перевірити, чи є `currState HIGH/True` (що означає виявлення руху) і запустити або, відповідно, зупинити попередній перегляд камери.

Змініть вручну або скопіюйте і вставте код нижче:

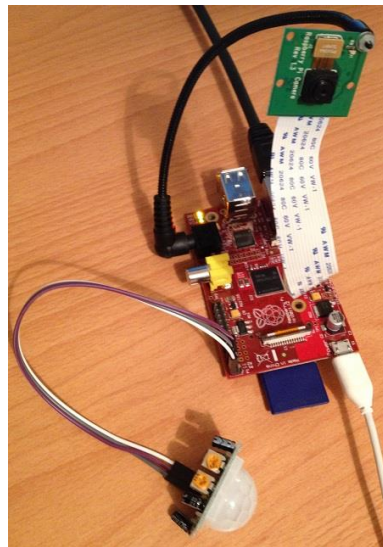


Рис. 7.19 Завершений проект

```

import RPi.GPIO as GPIO
import time
import picamera # new

sensorPin = 7

GPIO.setmode(GPIO.BOARD)
GPIO.setup(sensorPin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

prevState = False
currState = False

cam = picamera.PiCamera() # new

while True:
    time.sleep(0.1)
    prevState = currState
    currState = GPIO.input(sensorPin)
    if currState != prevState:
        newState = "HIGH" if currState else "LOW"
        print "GPIO pin %s is %s" % (sensorPin, newState)
        if currState: # new
            cam.start_preview()
        else:
            cam.stop_preview()

```

Натисніть Ctrl+O, щоб зберегти, і Ctrl+X для виходу. Для запуску програми використайте команду:

```
sudo ./pirCamera.py
```

Натисніть Ctrl+C, коли захочете вийти.

## Крок 6: Запис у файл і відтворення

Тепер ми можемо додати трохи більше коду, щоб дозволити запис у файл для відтворення пізніше. В ідеалі, якщо є багато відвідувачів у вашій кімнаті, і ви, звичайно, захочете захопити їх всіх, а не лише одного самого останнього. Таким чином, щоб зробити це, нам потрібен спосіб для автоматичного створення нового імені файлу щоразу, коли виявляється рух. Найпростіший і безпечний спосіб зробити це - створити файл з ім'ям з дати і часу.

Наприклад, якщо час був 11 лютого 2014 о 10:24 ранку і 18 секунд, то ім'я файлу буде виглядати приблизно так: 2014-02-11\_10.24.18.h264. При цьому використовується формат YEAR-MONTH-DAY\_HOUR.MINUTE.SECOND.h264; частина h264 є форматом відео, яке буде записуватися. Це той же формат, який використовує YouTube.

Щоб зробити це, ми повинні імпортувати модуль Python `datetime` і написати функцію для створення імені файлу. Дивіться нижче `getFileName`: там використовується рядок з функцією часу для вставки значення з поточного часу в зазначеному форматі рядка. Тоді ви просто використовуєте команди для запуску і зупинки запису з використанням згенерованого імені файлу. Це має відбутися в той же час, коли і відповідні команди попереднього перегляду.

```
nano pirCamera.py
```

Змініть вручну або скопіюйте і вставте код нижче:

```
import RPi.GPIO as GPIO
import time
import picamera
import datetime # new

def getFileName(): # new
    return datetime.datetime.now().strftime("%Y-%m-%d_%H.%M.%S.h264")

sensorPin = 7

GPIO.setmode(GPIO.BOARD)
GPIO.setup(sensorPin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

prevState = False
currState = False

cam = picamera.PiCamera()

while True:
    time.sleep(0.1)
    prevState = currState
    currState = GPIO.input(sensorPin)
    if currState != prevState:
        newState = "HIGH" if currState else "LOW"
        print "GPIO pin %s is %s" % (sensorPin, newState)
        if currState:
            fileName = getFileName() # new
            cam.start_preview()
            cam.start_recording(fileName) # new
        else:
            cam.stop_preview()
            cam.stop_recording() # new
```

Натисніть Ctrl+O, щоб зберегти, і Ctrl+X для виходу. Для запуску програми використайте наступну команду:

```
sudo ./pirCamera.py
```

Натисніть Ctrl+C, якщо захочете вийти.

### Відтворення

Якщо зараз використаєте команду `ls`, то побачите, що були отримані кілька файлів. Можете використовувати наступну команду, щоб відтворити файл. Замініть `<file>` ім'ям файлу, який хочете подивитися.

```
omxplayer <file> -o hdmi
```

Наприклад: `omxplayer 2014-02-11_10.24.18.h264 -o hdmi`

### Крок 7: Режим невидимості

Ви, напевно, помітили, що червоний світлодіод на платі камери спалахує при запуску програми на Python. Це буде досить помітно для будь-якого потенційного порушника, тому було б чудово відключити світлодіод. Це можна зробити шляхом редагування конфігураційного файлу Raspberry Pi, як ми це робили раніше. Введіть таку команду:

```
sudo nano /boot/config.txt
```

Додайте наступний рядок в кінець файлу:

```
disable_camera_led=1
```

Натисніть Ctrl+O, щоб зберегти, і Ctrl+X для виходу. Зміни вступають в силу тільки після перезавантаження, тому введіть `sudo reboot`

Якщо хочете залишити монітор підключений і включати в той час, як програма працює, це чудова ідея, щоб змінити рядки коду Python для відключення камери `*preview`. Використовуйте знак `#` на початку рядка, щоб відключити його.

Ще одна хитрість, якою можете скористатися, це запустити програму Python під іншим логіном. Для цього натисніть ALT+F2, перш ніж увійти: вам буде показана нова підказка входу, тому увійдіть і запустіть програму на Python. Тепер, якщо натиснете ALT+F1, щоб повернутися до звичайного запрошення, то воно з'явиться так, як ніби Raspberry Pi невинно чекає чийогось входу.

Тепер все, що нам залишилося зробити, це встановити пристрій і чекати.

Далі в розділі про використання мови Wolfram ми розглянемо, як знімки з камери передати електронною поштою для їх віддаленого перегляду.

## 8. Wolfram і Mathematica та математичні задачі

### Мова Wolfram і Mathematica на кожному Raspberry Pi



Запущений пілотний випуск мови Wolfram і пакета Mathematica<sup>137</sup>. Сьогодні вони вже вони стали частиною стандартного системного програмного забезпечення на кожному Raspberry Pi (рис.8.1).

Вміння програмування може стати центральним компонентом освіти. І з мовою Wolfram, нарешті, маємо потужну мову програмування гідну наступного покоління. Маємо мову, яка не пов'язана в основному з деталями комп'ютерів, але, незалежно від цього, в змозі зрозуміти і створювати речі на основі величезної кількості вбудованих обчислювальних можливостей і знань.

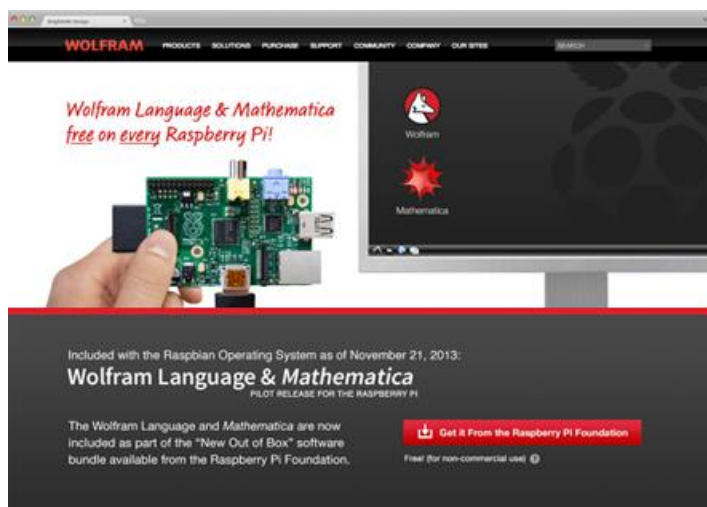
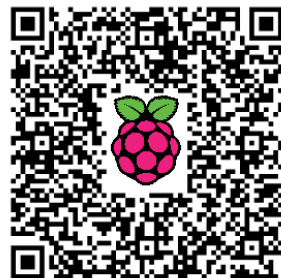


Рис.8.1 Wolfram і Mathematica на кожному Raspberry Pi

Це величезне задоволення і освітній потенціал. Написали крихітну програму, можливо, навіть в один рядок, і вже маєте щось дійсно цікаве. А потім можливостей для розширення все більше і більше, використовуючи всі потужні парадигми програмування, на яких побудована мова Wolfram.

<sup>137</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1765-wolfram-mathematica-language-and-each-raspberry-pi>

А з Raspberry Pi є ще щось цікаве: з ним відразу ви в змозі взаємодіяти з зовнішнім світом. Візьміть чистий код і підключіть його до датчиків і пристроїв, які дещо роблять.

Можливо, це досить дивно, що всі знання і обчислення на мові Wolfram можуть працювати на комп'ютері за \$35. І це початок чогось дуже важливого. Адже це означає, що в майбутньому буде технічно можливим вбудовувати мову Wolfram в майже будь-яку нову машину або систему. Для ефекту негайної ін'єкції інтелекту високого рівня і здібностей.

Ще в 1988 році, коли пакет Mathematica був випущений вперше, він міг працювати тільки на Mac високого класу, але не на ПК. Десять років по тому, навіть при тому, що пакет значно виріс, він може працювати на майже будь-якому персональному комп'ютері, який зараз продається. Але вбудовувані комп'ютери - це зовсім інша історія і очікувалося, що тільки спеціально складений простий код може працювати на них.

Але розробники вірили, що в один прекрасний день мова Wolfram зможе працювати в повному вигляді на вбудованому комп'ютері. І тепер стає ясно, що, нарешті, цей день настав: з Raspberry Pi розробники переступили поріг і в змозі запустити мову Wolfram на вбудованому комп'ютері в будь-якому місці.

Щоб було зрозуміло, Raspberry Pi, мабуть, від 10 до 20 разів повільніший при виконанні мови Wolfram, ніж типова модель ноутбука (а іноді навіть ще повільніший, коли він використовує внутрішні бібліотеки з особливостями архітектури). Але для багатьох речей швидкість Raspberry Pi просто відмінна. Наприклад, старий тест обчислень  $1989^{1989}$ , що витрачав багато секунд на комп'ютерах, які існували, коли Mathematica лише стартувала, зараз виконується за незмірно короткий час на Raspberry Pi.

З точки зору інженерного програмного забезпечення в комплекті з Raspberry Pi є пілотна версія нової Wolfram Engine. Зараз є два додатки на Pi, що працюють на цьому движку. Першим з них є версія командного рядка мови Wolfram. А другий - Mathematica з призначенням для користувача ноутбука інтерфейсом, забезпечуючим потужність на основі документу взаємодії з мовою Wolfram (рис.8.2).

Командний рядок мови Wolfram цілком жвавий на Raspberry Pi. Повний інтерфейс ноутбука в Mathematica вимагає цілого стека X-вікна і може бути досить млявим за сучасними мірками (і розробники були змушені кілька речей за замовчуванням відключити, таких як новий інтелектуальний інтерфейс, тому що вони просто занадто сильно уповільнювали роботу). Але це все



Рис.8.2 Wolfram & Mathematica

ще вражаюче: вперше Mathematica змогла працювати взагалі ні на чому, типу на комп'ютері за \$35.

І це вся система. Нічого не залишилося поза. Всі 5000+ функцій мови Wolfram. Всі можливості Mathematica і його інтерфейсу блокнота.

Найяскравіше те, що все це на Raspberry Pi, який заохочує спробувати новий стиль обчислень при підключенні до реального світу. Для старту - легке підключення пристроїв до Pi. А Pi невеликий і досить дешевий, щоб його можна було поставити практично в будь-якому місці. І якщо стартувати на ньому програму мовою Wolfram, то реально можна очікувати, що вона напевне завжди запуститься - аналізуючи та завантажуючи дані датчика, керуючи автономною системою, аналізуючи та виконуючи маршрутизацію трафіку, або щось інше.

Вбудовування автоматизації в таких великих масштабах є давнім принципом розвитку для мови Wolfram. І коли справа доходить до зовнішніх пристроїв, це означає послідовне визначення властивостей пристроїв, а потім створення загальних символічних функцій для взаємодії з ними.

Ось як можна було б прийняти весь цей набір технологій і використати його для включення світлодіодів, встановлюючи напругу на виводах GPIO (рис.8.3):

```
In[1]:= DeviceWrite["GPIO", {4 → 1, 22 → 1, 25 → 1}];
```

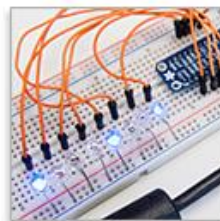


Рис.8.3 Мова Wolfram і GPIO

А ось деякий аналіз зображень на selfie, прийнятих з RaspiCam:

```
In[2]:= Colorize[MorphologicalComponents[EdgeDetect[DeviceRead["RaspiCam"]]]]
```



Рис.8.4 Аналіз зображень

Щось ми випускаємо з розгляду можливості для Raspberry Pi з Remote Development Kit (RDK), які дозволяють розвинути код і

підтримувати інтерфейс ноутбука на стандартному ноутбуці або іншому комп'ютері, поки плавно виконується код на мережевому дистанційному Raspberry Pi. Нинішній RDK підключається до копії Mathematica (наприклад, Mathematica Student Edition), що працює на будь-якій Mac, PC або Linux машині, а незабаром будуть й інші варіанти, наприклад в Інтернеті.

У мови Wolfram є фактично вся структура для символічно представлених екземплярів дистанційного запуску мови - для збору результатів, команд диспетчера, виконання паралельних обчислень, і так далі. Розробники також збираються надати Wolfram Symbolic Transport Protocol (WSTP, отриманий з протоколу MathLink, який використовується протягом майже 25 років), що дає один код обміну, дані або що-небудь ще дуже гнучко.

Автор дуже радий бачити речі, які люди винаходять з мовою Wolfram на Raspberry Pi - і він сподівається прочитати про деякі з них на Wolfram Community<sup>138</sup> (рис.8.5), а також на сайті Raspberry Pi Foundation<sup>139</sup>.



Рис.8.5 Сайт Wolfram Community

Те, що випускається сьогодні на Raspberry Pi - це тільки перший пілот для мови Wolfram. Буде багато оновлень, особливо в міру наближення першого серійного випуску мови.

Як і Wolfram|Alpha в Інтернеті, мова Wolfram (і Mathematica) на Raspberry Pi будуть безкоштовними для будь-якого використання в особистих цілях. (Саме таким буде механізм ліцензування для комерційного використання, для інших систем ARM Linux, і так далі.)

Звертаючись до історії, автор міг би згадати, що Raspberry Pi є лише другим комп'ютером за весь час, на якому комплект Mathematica

<sup>138</sup> <http://community.wolfram.com/>

<sup>139</sup> <http://www.raspberrypi.org/>

буде для безкоштовного використання. (Не рахуючи, звичайно, всі комп'ютери в університетах з ліцензіями сайту і т.д.) Першим був NeXT Computer Стіва Джобса в 1988 році.

Автор і досі сьогодні регулярно зустрічає людей, які розповідають йому, наскільки важливою була Mathematica для них. Не кажучи вже про купку комп'ютерів NeXT, які були куплені ЦЕРН для фізиків для запуску Mathematica.

Що буде зроблено з мільйонами примірників мови Wolfram, які в комплекті на комп'ютерах Raspberry Pi по всьому світу? Можливо, деякі дивні і неймовірно важливі винаходи з ними. Може бути, десять деякі діти отримають натхнення і будуть намагатися змінити світ.

Але ясно одне: з мовою Wolfram на Raspberry Pi у нас є новий шлях для навчання програмування і, підключивши його до реального світу, що дуже багато людей збираються зробити, щоб мати можливість отримати вигоду.

Одним з кращого при роботі над Raspberry Pi була можливість зустрітися групі людей, які намагаються домогтися тих же змін у викладанні інших предметів, які ми прагнемо в обчислювальній техніці. Таким чудовим прикладом є рух Математика на базі комп'ютера (CBM), метою якого є перенаправити викладання математики в школах від механічних розрахунків до вирішення проблем.

На саміті освіти CBM в Нью-Йорку оголошено про партнерство з Wolfram Research для інтеграції безкоштовної копії Mathematica і мови Wolfram в майбутні образи Raspbian. Ми вважаємо, що це зробить Pi першокласною платформою для викладання методики CBM для дітей різного віку. Як заявив

Конрад Вольфрам:  
*"Програмісти зможуть використовувати силу математики системи з коробки, не тільки збагачуючи те, що вони можуть зробити, але і демонструючи силу і важливість математики."*

Нинішній образ Raspbian поставляється з мовою Wolfram та Mathematica за

умовчанням. Існуючі користувачі, які мають принаймні 600 Мб вільного місця на своїй SD-карті, можуть встановити їх, набравши:

```
sudo apt-get update  
sudo apt-get install wolfram-engine
```

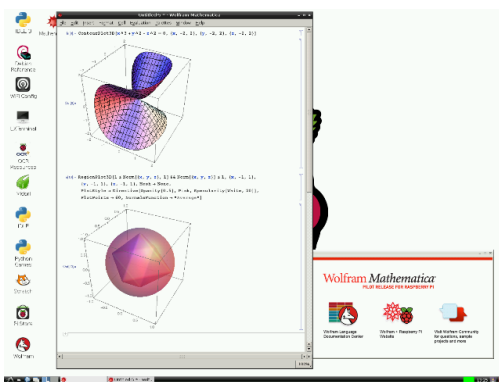


Рис.8.6 Побудова 3d графіки з Mathematica на Pi

## Основи програмування з Mathematica

Для першого знайомства з інтерфейсом Mathematica радимо прочитати ілюстрований посібник<sup>140</sup>.

Тепер, клікніть у вікні блокнота (notebook), введіть:

```
Print["Hello world"]
```

натисніть **Shift + Enter**, щоб запустити команду і вивести "Hello world" на екран ось так (рис.8.7):

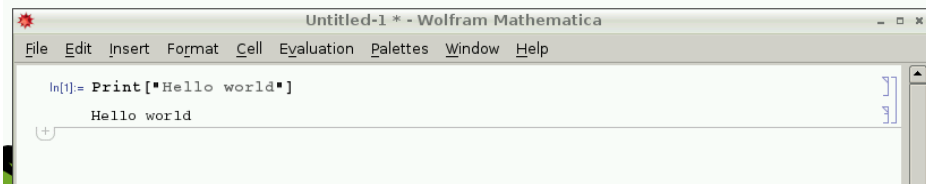


Рис.8.7 "Hello world" в Mathematica

Ви можете виконувати математичні розрахунки таким чином:

```
In[2]:= 2 + 2
```

```
Out[2]= 4
```

```
In[3]:= 16254 / 32
```

```
Out[3]= 8127 / 16
```

```
In[4]:= 1024 * 32
```

```
Out[4]= 32768
```

### Редагування в блокноті

Ви можете повернутися до раніше введеної команди, клацнувши на ній або переміщуючи курсор редагування за допомогою клавіатури, а потім видалити, змінити чи додати щось і натиснути **Shift + Enter**, щоб виконати нову команду на її місці.

Ви можете зберегти блокнот і повернутися до нього пізніше, відправити його другу, розмістити його в Інтернеті або навіть передати як домашнє завдання! Просто перейдіть в меню **File > Save As** у вікні блокнота.

Коли відкриєте збережений блокнот, то будуть показані всі попередні записи, зокрема, всі входи і виходи, і ви можете знову виконувати кожну клітину через **Shift+Enter** або все відразу, вибравши з меню **Evaluation > Evaluate Notebook**.

<sup>140</sup> <http://www.burnlib.com/x/illyustrirovannyiy-samouchitel-po-mathematica/>

### Змінні

Ви можете зберігати результати обчислень у змінних:

```
radius = 5;  
diameter = 2 * radius;  
circumference = 2 * Pi * radius;  
area = Pi * radius^2;
```

*Примітка:* крапка з комою в кінці кожного рядка подавляє вихід готового документа.

### Символічні значення

Зверніть увагу на використання вбудованого символу `Pi`, який містить символічне значення числа  $\pi$ . Це означає, що якщо ви передаєте його в рівняння посиланням, то зберігається істинне значення  $\pi$ , не перетворюючись в десяткове і не округлюючись:

```
In[19] := Pi
```

```
Out[19]:  $\pi$ 
```

```
In[20] := tau = 2 * Pi
```

```
Out[20]:  $2 \pi$ 
```

Щоб отримати десяткове подання символічного значення, використайте `N`-функцію:

```
In[5] := N[Pi]
```

```
Out[5]: 3.14159
```

За замовчуванням кількість значущих цифр становить 6, але можна буде задано більше, вказавши в другому аргументі:

```
In[6] := N[Pi, 10]
```

```
Out[6]: 3.141592654
```

*Примітка:* це число знаків, а не знаків після коми, так що 3 входить в підрахунок, залишаючи 9 знаків після коми.

### Списки

Ви можете зберігати колекції даних у списку:

```
nums = {1, 2, 3, 5, 8}
```

```
people = {"Alice", "Bob", "Charlotte", "David"}
```

## Діапазон

Функція `Range` (Діапазон) може бути використана для отримання списку чисел:

```
Range[5] (*Значення від 1 до 5*)
Range[2, 5] (*Значення від 2 до 5*)
Range[2, 5, 2] (*Значення від 2 до 5, з кроком 2*)
```

## Таблиця

Функція `Table` (Таблиця) є способом генерації значення списку з функцією:

```
Table[i ^ 2, {i, 10}] (*Квадрати чисел від 1 до 10*)
Table[i ^ 2, {i, 5, 10}] (*Квадрати чисел від 5 до 10*)
Table[i ^ 2, {i, nums}] (*Квадрати значень в списку nums*)
```

## Цикл

Можете запустити цикл кілька разів, або через елементи у списку з `Do`:

```
Do[Print["Hello"], {10}] (*Виводити "Hello" 10 разів*)
Do[Print[i], {i, 5}] (*Виводити числа від 1 до 5*)
Do[Print[i], {i, 3, 5}] (*Виводити числа від 3 до 5*)
Do[Print[i], {i, 3, 5}] (*Виводити числа від 1 до 5, з кроком 2*)
Do[Print[i ^ 2], {i, nums}] (*Виводити квадрати кожного значення в списку nums*)
```

## Функція `Help` (Допомога)

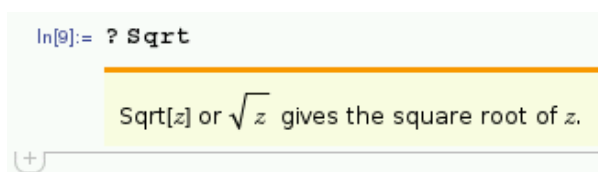


Рис.8.8 Допомога з використання функції

Можете отримати допомогу з використання функції, додаючи ім'я функції зі знаком питання (?) і натиснувши `Shift+Enter` (рис.8.8).

## Функція Search (Пошук)

Можете також шукати для функцій, ввівши частину імені функції, щоб знайти відповідне. Просто почніть з (?), додаючи зірочку (\*) як шаблон (рис.8.9):

```
In[15] := ?Device*
```

In[15] := ? Device\*

▼ System

|                           |                  |                      |
|---------------------------|------------------|----------------------|
| DeviceClose               | DeviceObject     | DeviceReadTimeSeries |
| DeviceConfigure           | DeviceOpen       | Devices              |
| DeviceDriverRepository    | DeviceOpenQ      | DeviceStreams        |
| DeviceExecute             | DeviceRead       | DeviceWrite          |
| DeviceExecuteAsynchronous | DeviceReadBuffer | DeviceWriteBuffer    |
| DeviceInformation         | DeviceReadLatest |                      |

Рис.8.9 Функція пошуку

Можете використовувати кілька символів шаблону:

```
In[16] := ?*Close*
```

In[16] := ? \*Close\*

▼ System

|                        |                     |                        |
|------------------------|---------------------|------------------------|
| AllowGroupClose        | ClosenessCentrality | ShowClosedCellArea     |
| AllowReverseGroupClose | CurveClosed         | ShowGroupOpenCloseIcon |
| Close                  | DeviceClose         | SplineClosed           |
| Closed                 | LinkClose           |                        |
| CloseKernels           | NotebookClose       |                        |

Рис.8.10 Використання символів шаблону

## Коментарі

Як видно з попередніх прикладів, ви можете залишити коментарі (примітки, які ігноруються програмою) в сценаріях, використовуючи дужки ((&)) й зірочки (\*):

```
Print["Hello"] (*Вивести "Hello" на екран*)
```

## Доступ до командного рядка Wolfram

Ви також можете отримати доступ до мови Wolfram з командного рядка, ввівши `wolfram` в терміналі або двічі клацнувши на значку Wolfram Desktop. Це дасть тільки текстове ( без графіки) середовище програмування з інтерфейсом в стилі `In[x]` / `Out[x]`, але без

інтерактивних можливостей блокнота. Функції Mathematica будуть працювати очікувано, як і раніше (рис.8.11):

```
In[6]:= ?Sqrt
Sqrt[z] or Sqrt[z] gives the square root of z.

In[7]:= ?*Open
CellOpen      LinkOpen      Open          SystemOpen
DeviceOpen    NotebookOpen

In[8]:= □
```

Рис.8.11 Використання командного рядка

Ви знайдете інтерфейс командного рядка більш швидким у використанні за рахунок відсутності графічної обробки, необхідної для запуску блокнота, але без інтерактивності і чудового графічного виведення.

Для виходу натисніть Ctrl+D.

## Запуск сценаріїв з Wolfram

Ви можете написати програму і зберегти її як звичайний файл (зазвичай з `.m` або `.wl` розширенням файлу) і запустити сценарій з командного рядка, додавши ключ `-script`.

Для запуску `test.m` (рис.8.12):

```
wolfram -script test.m
```



Рис.8.12 Запуск сценарію

## Список операцій

Ви можете застосувати операцію або функцію до всіх елементів у списку:

```
In[21]:= 2 * {1, 2, 3, 4, 5}
Out[21]: {2, 4, 6, 8, 10}
```

```
In[22]:= {1, 2, 3, 4, 5} ^ 2
Out[22]: {1, 4, 9, 16, 25}
```

```
In[23]:= Sqrt[{1, 2, 3, 4, 5}]
Out[23]: {1, Sqrt[2], Sqrt[3], 2, Sqrt[5]}
```

*Примітка:* в останньому прикладі квадратні корені з 1 і 4 були дані саме такими, бо вони дають цілі значення, а квадратні корені з 2, 3 і 5, які є ірраціональними, дані символічно.

### Матриці

Один з найбільш корисних додаткових компонентів математичної мови програмування є можливість виконувати матричні операції. Звичайно, в Mathematica вони доступні.

Для створення матриці, спочатку введіть значення у вигляді списку списків, переконавшись, що розміри прямокутні, тобто  $n \times m$  де  $n$  і  $m$  цілі:

```
m = {{1, 2}, {3, 4}, {5, 6}};
```

Ви можете переглянути цей список у вигляді матриці, набравши:

```
m // MatrixForm
```

```
In[21]:= m = {{1, 2}, {3, 4}, {5, 6}}
Out[21]= {{1, 2}, {3, 4}, {5, 6}}

In[22]:= m // MatrixForm
Out[22]//MatrixForm=

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

```

Ви можете виконувати такі матричні операції, як скалярне множення:

```
m = {{1, 2}, {3, 4}, {5, 6}};
m2 = {{10, 20, 30}, {40, 50, 60}};
m . m2 // MatrixForm
```

```
In[7]:= m = {{1, 2}, {3, 4}, {5, 6}};
      m2 = {{10, 20, 30}, {40, 50, 60}};
      m . m2 // MatrixForm
Out[9]//MatrixForm=

$$\begin{pmatrix} 90 & 120 & 150 \\ 190 & 260 & 330 \\ 290 & 400 & 510 \end{pmatrix}$$

```

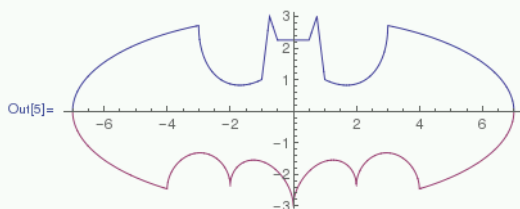
### Графіки

Ви можете побудувати цікаві речі за допомогою Mathematica:

```

In[5]:= Plot[
  {With[{w = 3 * Sqrt[1 - (x / 7) ^ 2],
    l = (6 / 7) * Sqrt[10] + (3 + x) / 2 - (3 / 7) * Sqrt[10] * Sqrt[4 - (x + 1) ^ 2],
    h =
      (1 / 2) * (3 * (Abs[x - 1 / 2] + Abs[x + 1 / 2] + 6) -
        11 * (Abs[x - 3 / 4] + Abs[x + 3 / 4])),
    r = (6 / 7) * Sqrt[10] + (3 - x) / 2 - (3 / 7) * Sqrt[10] * Sqrt[4 - (x - 1) ^ 2]},
    w + (1 - w) * UnitStep[x + 3] + (h - 1) * UnitStep[x + 1] + (r - h) * UnitStep[x - 1] +
    (w - r) * UnitStep[x - 3]],
  (1 / 2) * (3 * Sqrt[1 - (x / 7) ^ 2] + Sqrt[1 - (Abs[Abs[x] - 2] - 1) ^ 2] +
    Abs[x / 2] - ((3 * Sqrt[33] - 7) / 112) * x ^ 2 - 3) *
    ((x + 4) / Abs[x + 4] - (x - 4) / Abs[x - 4]) - 3 * Sqrt[1 - (x / 7) ^ 2]},
  {x, -7, 7}, AspectRatio -> Automatic]

```



Наприклад, графік echidnahedron (рис.8.13) за допомогою наступної команди:

```

Graphics3D[{Opacity[.8], Glow[RGBColor[1,0,0]], EdgeForm[White],
  Lighting -> None, PolyhedronData["Echidnahedron", "Faces"]}]]

```

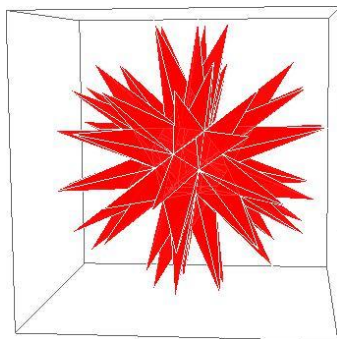


Рис. 8.13 Графік echidnahedron

## GPIO

Ви можете отримати доступ до виводів GPIO з Mathematica за допомогою функцій `DeviceWrite` і `DeviceRead`.

Щоб отримати доступ до виводів GPIO, ви повинні запускати з правами `root`, тобто викликати в терміналі `sudo wolfram` (це запустить середовище командного рядка `wolfram`) або `sudo mathematica &` для запуску блокнота Mathematica від `root`.

Наступна команда встановлює вивід 14 GPIO (за допомогою BCM нумерації виводів):

```
DeviceWrite["GPIO", 14 -> 1]
```

а команда нижче вимикає вивід 14:

```
DeviceWrite["GPIO", 14 -> 0]
```

Ви також можете прочитати статус входу пристрою GPIO (щоб перевірити, чи натиснута кнопка, наприклад) з `DeviceRead`, майже таким же способом:

```
button = DeviceRead["GPIO", 14]
```

Тепер змінна `button` повинна мати значення 0 при відключенні або 1, коли натиснута.

```
In[1]:= DeviceRead["GPIO", 14]
Out[1]= {14 -> 0}

In[2]:= DeviceWrite["GPIO", 14 -> 1]

In[3]:= DeviceRead["GPIO", 14]
Out[3]= {14 -> 1}
```

Дізнайтеся більше в загальному про GPIO на сторінці використання GPIO<sup>141</sup>.

### Камера

Можна робити знімки за допомогою камери, використовуючи функцію `DeviceRead`. По-перше, підключіть камеру, як звичайно.

Щоб зробити стоп-кадр з камери, введіть наступну команду:

```
img = DeviceRead["RaspiCam"]
```

Потім, щоб зберегти зображення у файл, використовуйте `Export` і вкажіть шлях збереження та змінну, що містить зображення:

```
Export["/home/pi/img.jpg", img]
```

Нижче наведені приклади цікавих проектів на Raspberri Pi з використанням мови Wolfram.

---

<sup>141</sup> <http://www.raspberrypi.org/documentation/usage/gpio/README.md>

## Використання мови Wolfram для GPIO



Покажемо більш детально, як мова Wolfram допомагає використовувати GPIO<sup>142</sup> на Raspberry Pi.

Щоб відтворити даний експеримент, вам знадобиться наступне обладнання (на додаток до самого Raspberry Pi):

- супер яскраві сині світлодіоди
- комплект Pi T-Cobbler
- повнорозмірна макетна плата
- дротяні перемички штир/штир
- резистор на 400 Ом (для захисту світлодіодів від вигорання).

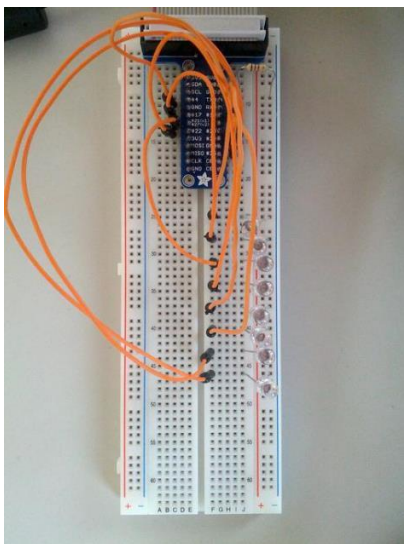
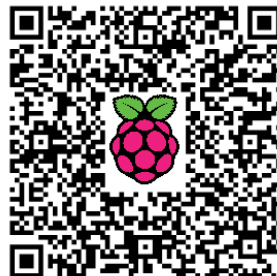


Рис.8.14 Керування світлодіодами

Налаштуйте макетну плату, як показано на рис.8.14: підключіть Pi T-Cobbler до макету 13 виводами в стовпці E і 13 виводами в стовпці G. Використайте перемички для підключення виводів 4, 17, 27, 22, 18, 23, 24 і 25, щоб рівномірно їх розмістити на вільних рядках на макетній платі нижче. Підключіть кожен з 8 синіх світлодіодів до перемичок поспіль і до синього стовпця, щоб більш плоский бік (катод) був приєднаний до синього стовпця. Завершіть контур, підключивши резистор від синього стовпця до контакта GND.

<sup>142</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1834-using-gpio-on-raspberry-pi-from-wolfram-language>

## 8. WOLFRAM I MATHEMATICA TA МАТЕМАТИЧНІ ЗАДАЧІ

Правильно підключіть плоский кабель до T-Cobbler і Raspberry Pi. Включіть Raspberry Pi.

Інтерфейс GPIO для доступу вимагає повноважень адміністратора, тому для цього експерименту мова Wolfram або Mathematica повинні бути запуснені під користувачем root.

У терміналі запусить мову Wolfram за допомогою наступної команди, звичайно, з правами адміністратора:

```
> sudo wolfram

Wolfram Language (Raspberry Pi Pilot Release)
Copyright 1988-2013 Wolfram Research
Information & help: wolfram.com/raspi

In[1]:=
```

Спочатку визначимо виводи, які відповідають підключеним світлодіодам:

```
pins = {4,17,27,22,18,23,24,25}
```

Тепер можемо включити окремі світлодіоди, написавши значення для них "1":

```
DeviceWrite[ "GPIO", First[pins] -> 1 ]
```

І, звичайно, щоб вимкнути діод, треба написати значення '0':

```
DeviceWrite[ "GPIO", First[pins] -> 0 ]
```

Або включити і вимкнути світлодіоди в циклі за один раз:

```
Do[
  DeviceWrite[ "GPIO", pins[[i]]->1 ];
  Pause[.2];
  DeviceWrite[ "GPIO", pins[[i]]->0 ];
  ,{i,8}]
```

Думаю, що коментувати рядки команд не треба, бо з мовою Wolfram все прозоро.

## Побудова GPS-трекера



Розглянемо, як побудувати простий GPS-трекер з використанням мови Wolfram на Raspberry Pi<sup>143</sup>.

Щоб відтворити цей експеримент нам знадобиться наступне обладнання (на додаток до самого Raspberry Pi):

- GPS-модуль (Ultimate GPS Breakout) (рис.8.15)
- послідовний кабель USB до TTL (рис. 8.16)
- макетна плата половинного розміру
- дротяні перемички штир/штир

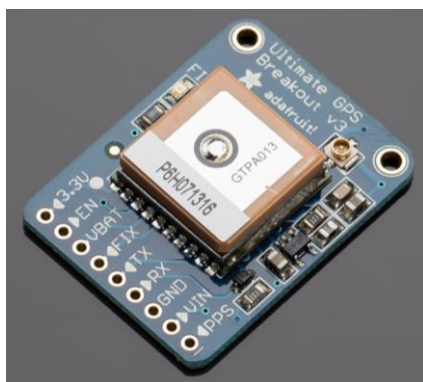
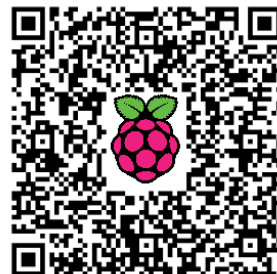


Рис.8.15 GPS-модуль

Вставте модуль GPS в макет, як показано на рис.8.17, і підключіть червоний провід до VIN, чорний до GND (землі), зелений до RX, білий до TX, використовуючи перемички і послідовний кабель USB до TTL.



Рис.8.16 Послідовний кабель USB до TTL

<sup>143</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1835-building-a-gps-tracker-for-raspberry-pi>

Підключіть кінець USB кабелю до Raspberry Pi (вимкненого). Подайте живлення на Raspberry Pi.

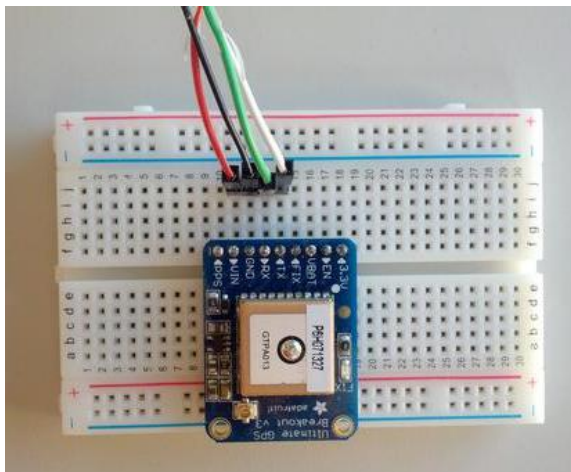


Рис.8.17 Модуль GPS в макеті

Модуль GPS починає передавати дані відразу після включення живлення і буде продовжувати це робити, поки не буде відключений від джерела живлення.

Запустіть у терміналі мову Wolfram за допомогою наступної команди:

```
> wolfram

Wolfram Language (Raspberry Pi Pilot Release)
Copyright 1988-2013 Wolfram Research
Information & help: wolfram.com/raspi

In[1]:=
```

Тепер можете відкрити послідовний порт, використовуючи DeviceOpen та ввівши:

```
serial = DeviceOpen["Serial", {"/dev/ttyUSB0", "BaudRate" -> 9600}]
```

Це повертає DeviceObject, який може бути використаний для читання даних GPS. У нашому випадку використовуємо DeviceReadBuffer, щоб прочитати всі наявні дані GPS, які були створені до цього моменту:

```
data = DeviceReadBuffer[serial, "String"]
```

Дані, що повертаються, мають формат розділених комами даних, так звані пропозиції GPS NMEA:

```
$GPRMC,154541.000,A,4005.8369,N,08814.7322,W,0.04,253.32,201113,, ,A*79
$GPVTG,253.32,T,,M,0.04,N,0.07,K,A*3B
$GPGGA,154542.000,4005.8369,N,08814.7322,W,1,8,1.07,228.0,M,-33.9,M,,*6B
$GPGSA,A,3,04,12,10,17,23,24,25,02,,,,,1.31,1.07,0.76*04
$GPGSV,3,1,12,04,65,040,24,02,63,265,16,10,55,135,39,12,48,302,21*7D
$GPGSV,3,2,12,17,35,096,33,05,19,190,17,25,13,321,33,24,12,247,16*71
$GPGSV,3,3,12,23,05,061,31,13,02,090,27,20,02,036,35,45,,, *45
$GPRMC,154542.000,A,4005.8369,N,08814.7322,W,0.06,253.32,201113,, ,A*78
```

Ми можемо імпортувати ці дані з мовою Wolfram за допомогою ImportString:

```
csv = ImportString[ data, "CSV" ]
```

У пропозиціях NMEA, які містять координати GPS, координати починаються з \$GPRMC, тому для цього ми фільтруємо, використовуючи Cases і зіставлення зі зразком:

```
gps = Cases[ csv, {"$GPRMC", ____} ]
```

Координати (широта і довгота) знаходяться в 4-й і 6-й позиціях кожного списку, наприклад, GPS-координати для першої точки даних:

```
Part[ gps, 1, {4,6} ] / 100
```

що повертає розташування GPS для Champaign, штат Іллінойс, де знаходиться Wolfram Research на 40' північної широти і 88' західної довготи:

```
{40.0583, 88.1474}
```

Як бачимо, завдяки використанню мови Wolfram, задача знову вирішується досить просто.

## Система домашньої сигналізації з датчиком руху PIR



В прикладі показано як, використовуючи датчик руху PIR з Raspberry Pi і маючи керування GPIO, за допомогою мови Wolfram, говорити, захоплювати зображення і відправляти його електронною поштою щоразу, коли датчик виявить рух<sup>144</sup>.

Щоб відтворити цей приклад знадобиться наступне обладнання (на додаток до самого Raspberry Pi):

- комплект Pi T-Cobbler Breakout
- макетна плата
- пасивний інфрачервоний датчик руху (PIR)

Хоча датчик PIR вимагає живлення 5В, його вихід має «дружелюбні» для Pi 3.3В, тому він може бути підключений безпосередньо до входу GPIO, для чого автор обрав контакт за номером 18. Налаштуйте все, як показано на рис.8.18 нижче, встановіть затримку PIR до спрацювання (близько 2-4 секунд) і його чутливість точно, як зробив автор (переконайтеся, що помаранчеві ручки управління потенціометрами знаходяться в такому ж положенні):

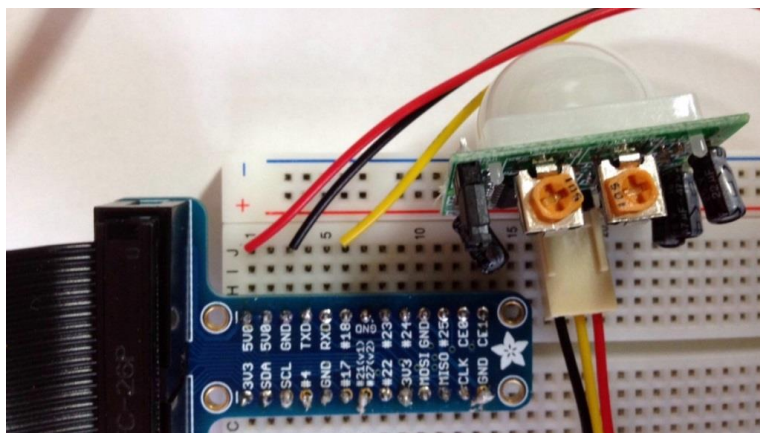
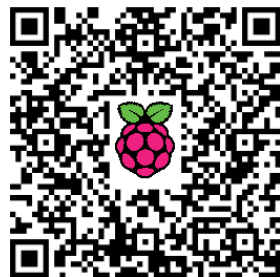


Рис.8.18 Налаштування PIR

Після того, як все встановили, ви готові використовувати GPIO з мовою Wolfram на Raspberry Pi.

Налаштуємо вивід 18 для читання, і він повертатиме 1, якщо буде реєстрація руху, та 0 - в іншому випадку:

<sup>144</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1836--raspberry-pi-pir>

```
DeviceConfigure["GPIO",{18->"Input"}]
```

Ви можете перевірити, що датчик нормально працює наступним чином:

```
Dynamic[If[DeviceRead["GPIO",18]=={18->1}, Speak["Smile!"]]; UpdateInterval -> 3]
```

Зверніть увагу, що для Raspberry Pi треба мати динамік, підключений до його 3,5-мм аудіо роз'єму.

Тепер можна, нарешті, переслати через електронну пошту зображення порушника за допомогою RaspiCam і наступного короткого коду:

```
Dynamic[If[DeviceRead["GPIO",18]=={18->1}, Speak["Smile!"]SendMail["To" -> "youralarm@gmail.com", "Subject" -> "Intruder", "Body" -> {"Intruder's Portrait\n", DeviceRead["RaspiCam"]},"From" -> "you@gmail.com", "Server" -> "smtp.gmail.com", "UserName" -> "you@gmail.com", "Password" -> "yourPassword", "PortNumber" -> 587, "EncryptionProtocol" -> "StartTLS"]]; UpdateInterval -> 3]
```

Звичайно ж в кодi вкажіть адресу електронної пошти на gmail.com та пароль доступу в свою поштову скриньку.

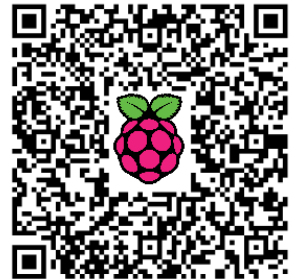
До чого ж все просто з мовою Wolfram!

## Raspberry Pi з датчиками Vernier і мова Wolfram



Давайте подивимося на те, як організувати взаємодію зовнішніх датчиків від Vernier Software & Technology з Pi<sup>145</sup>, використовуючи мову Wolfram.

Навіть при тому, що мова Wolfram на Raspberry Pi<sup>146</sup> з'явилась всього кілька місяців тому, автор статті Боб LeSuer вже досвідчений користувач. Він ад'юнкт-професор хімії в державному університеті Чикаго і, природно, великий прихильник включення технологій в свої дослідження.



### Основа

У своїх лабораторних класах Боб LeSuer використовує інструменти компанії Vernier, яка створює широкий спектр датчиків<sup>147</sup> для збору даних, від акселерометрів і барометрів з датчиками CO<sub>2</sub> і pH. Але, що

<sup>145</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1800-raspberry-pi-with-vernier-sensors-and-wolfram-language>

<sup>146</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1765-wolfram-mathematica-language-and-each-raspberry-pi>

<sup>147</sup> <http://www.vernier.com/products/sensors/>

особливо зачепило нашу зацікавленість, це те, що Боб придумав ідею, як під'єднати ці датчики до Raspberry Pi, використовуючи мову Wolfram!

Тому, звичайно, ми повинні цим поділитися з вами.

Устаткування, яке знадобиться:

- Raspberry Pi з встановленими Wolfram Language + Mathematica
- датчик Vernier
- USB адаптер Go!Link<sup>148</sup>

Взагалі, датчики мають власний роз'єм, однак ви можете купити адаптер під назвою Go!Link для під'єднання датчиків через порт USB (рис.8.19):

Розробники з Vernier забезпечили відкритий SDK, який дозволяє кінцевим користувачам включати датчики Vernier в будь-яку кількість додатків.

Після встановлення SDK необхідно виконати трохи тонкого налаштування і трохи більше вивчити програмування на C, щоб бути в змозі працювати з рядом датчиків, що функціонують на RPi.



Рис.8.19 Адаптер Go!Link

### Установка SDK

По-перше, завантажте з Vernier Software Development Kit<sup>149</sup> для Linux.

Самою важкою частиною цього проекту є компіляції бібліотеки GoIO, яка, на жаль, не працює на RPi відразу. Дотримуйтесь інструкцій з розпакування стисненого файлу, але перед запуском сценарію треба зробити дві зміни.

Потрібно додати кілька рядків коду, щоб змусити його працювати на RPi.

У `config.in` додайте рядок:

```
AC_CONFIG_MACRO_DIR([m4])
```

І один рядок додайте в `Makefile.am`:

```
ACLOCAL_AMFLAGS= -I m4
```

Після того, як зробите ці зміни, можна запускати компіляцію і будьте терплячими, бо компіляція на RPi дуже повільна.

<sup>148</sup> <http://www.vernier.com/products/interfaces/go-link/>

<sup>149</sup> <http://www.vernier.com/downloads/software-development-kits/go-sdk/>

## Побудова бібліотеки MathLink

SDK поставляється з прикладом програми, яка може використовуватися для читання з датчика. Автор використовує велику частину цієї програми, щоб зробити програму з підтримкою MathLink, яка містить три функції: `getLibVersion`, `getDeviceInfo` і `getSimpleMeasurement`. Перша надає вам версію SDK (наприклад, 2,53). Друга надає назву датчика, підключеного в даний момент до з'єднання Go!Link і включає в себе одиниці вимірювання значення, які будуть повідомлені `getSimpleMeasurement`.

Якщо зацікавлені в тому, щоб прямо зараз перевірити, що система працює, то налаштуйте SimpleMeasurement на читання від датчика протягом 1 секунди і повідомлення середнього значення.

Автором використані код C<sup>150</sup> і код шаблону MathLink<sup>151</sup> для створення функцій `getLibVersion`, `getDeviceInfo` і `getSimpleMeasurements`, якими можна скористатися в Mathematica. Завантажте названі два файли і зкомпілюйте їх за допомогою наступної команди:

```
mcc -o vernier vernier.c vernier.tm -I/usr/include/GoIO -lGoIO
```

Переконайтеся, що бібліотека GoIO прив'язана і що каталог `/usr/include/GoIO` входить в шлях пошуку.

## Простий інтерфейс Mathematica

А тепер перейдемо на мову Wolfram, де ви зможете динамічно оновлювати показання сенсора за допомогою наступного коду:

```
In[2]:= link = Install["/mnt/documents/Pi/vernier/vernier"]
Out[2]:= LinkObject['/mnt/documents/Pi/vernier/vernier', 68, 4]

In[3]:= Dynamic[{
  Column[{
    Row@StringSplit[getDeviceInfo[], ", "][[4, 5]],
    getSimpleMeasurement[]
  ]},
  Clock[{1, 1}, 2]]]

Out[3]:= { CO2 High(ppm) , 1. }
          { 1590.96
```

<sup>150</sup> <http://www.bobthechemist.com/mathematica/codebase/vernier.c>

<sup>151</sup> <http://www.bobthechemist.com/mathematica/codebase/vernier.tm>

Код навіть постійно оновлюється з відповідними показаннями, як тільки ви поміняєте датчики!

Крім того, ви можете зробити оновлення діаграми в режимі реального часу, як показано нижче, використовуючи, наприклад, датчик освітленості:

```
info = StringSplit[getDeviceInfo[], ","]
Dynamic@ListPlot[data, PlotLabel -> info[[4]],
Frame -> {True, True, False, False}, FrameLabel -> {"Time (s)",
"Light " <> info[[5]]]
```

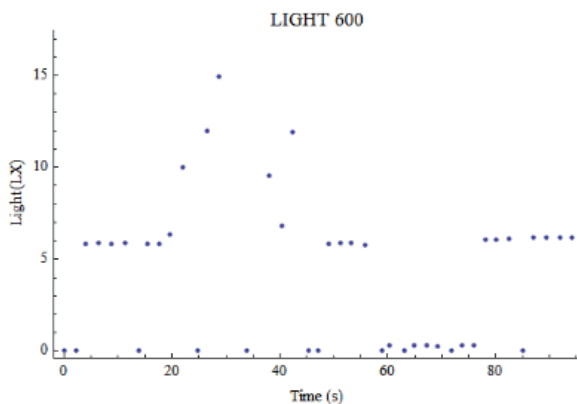


Рис.8.20 Діаграма в режимі реального часу

```
data = {{t0 = AbsoluteTime[], getSimpleMeasurement[]}};
data[[1, 1]] -= t0;
Dynamic[{AppendTo[data, {AbsoluteTime[] - t0,
getSimpleMeasurement[]}], Clock[{1, 1}, 2]}]
```

## Вершина айсберга

Ясно, що це тільки початок чогось дуже цікавого. Можна погратися з оптимізацією коду, щоб прискорити вимірювання, і багато роботи належить зробити для візуалізації та з даними після їх отримання.

Ви звернули увагу, що автор бере як належне той факт, що він взагалі ніколи не робить калібрування на будь-якому зі своїх датчиків? Це одна з переваг використання датчиків Vernier - SDK містить інформацію калібрування і дозволяє кінцевому користувачеві зайнятися безпосередньо збором даних.

## Розподілені обчислення за допомогою Raspberry Pi



Розподілені обчислення передбачають розподіл обчислювальної проблеми на декілька паралельних завдань, які будуть виконуватися двома або більше комп'ютерами в мережі, що утворюють розподілену систему<sup>152</sup>.

Це дозволяє поєднати в собі обчислювальну потужність декількох комп'ютерів для вирішення великих завдань, які включають обробку великих даних або вимагають величезної кількості ітерацій.

Сьогодні існує декілька поточних великомасштабних розподілених обчислювальних проєктів, що охоплюють різні сфери, які дозволяють взяти участь комп'ютерам з усього світу і багато з цих проєктів вже працюють протягом кількох років. Це показує наскільки великими можуть бути обчислювальні завдання в наші дні!

Ось кілька прикладів:

- **Великий інтернет-пошук Мерсенна**<sup>153</sup> (математика)

Пошук чисел Мерсенна, що є простими числами, які відповідають формі  $2^n - 1$ , або мають тільки '1' в двійковій системі.

- **Test4Theory**<sup>154</sup> (фізика)

Пошук нових фундаментальних частинок на Великому адронному колайдері в Церні.

- **Folding@Home**<sup>155</sup> (молекулярна біологія)

Запуск симуляції на складання протейнів, які допомагають вивчення таких захворювань, як хвороба Альцгеймера, Хантінгтона і багатьох видів раку.

- **MindModelling@Home**<sup>156</sup> (когнітивна наука)

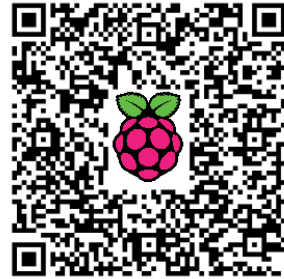
Використання чисельного моделювання процесу пізнання, щоб краще зрозуміти розум.

- **MilkyWay@Home**<sup>157</sup> (астрономія)

Створення високоточної 3D-моделі галактики Чумацький Шлях.

- **Enigma@Home**<sup>158</sup> (криптографія)

Злом трьох вихідних повідомлень Enigma, що не були ще зламані.



<sup>152</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1820-distributed-computing-using-raspberry-pi>

<sup>153</sup> <http://www.mersenne.org/>

<sup>154</sup> <http://lhathome2.cern.ch/test4theory/>

<sup>155</sup> <http://folding.stanford.edu/>

<sup>156</sup> <http://mindmodeling.org/>

<sup>157</sup> <http://milkyway.cs.rpi.edu/milkyway/>

<sup>158</sup> <http://www.enigmaathome.net/>

## Демонстрація розподілених обчислень з використанням двох Pi

Для демонстраційних цілей автор підключив 2 Raspberry Pi за допомогою кабелю Ethernet і виконав просте сортування злиттям на великому масиві елементів.

По-перше, ось код для алгоритму сортування злиттям, написаний на python (MergeSort.py):

```
import time
def merge(left,right):      #Зливаємо разом 2 загальні списки
result = []
i, j = 0, 0
    #Виконання за обома списками
while i < len(left) and j < len(right):
    #Додавання меншого елемента списків в остаточний список
    if left[i] <= right[j]:
result.append(left[i])
i += 1
    else:
result.append(right[j])
j += 1
result += left[i:]
result += right[j:]
return result
def mergesort(lst):
    #Якщо є тільки 1 елемент, то сортувати не потрібно
if len(lst) < 2:
return lst
    #Ділимо список на 2 половини
middle = len(lst) / 2
    #Рекурсивно розбиваємо і сортуємо кожну половину
left = mergesort(lst[:middle])
right = mergesort(lst[middle:])
    #Зливаємо списки разом, як впорядковані
return merge(left, right)
```

Далі наведений код для використання алгоритму сортування злиттям для сортування масиву 100000 елементів, використовуючи лише один Raspberry Pi (Merge1.py). (Цей код буде використовуватися в наступних програмах, тому вони розміщені в тому ж каталозі перед запуском!)

```
import MergeSort              #Імпорт функцій злиття
import random
import time
    #Створюємо масив для сортування
arraylength = 100000          #Довжина масиву для сортування
print 'Length of array is', arraylength
array = range(arraylength)    #Створюємо масив
random.shuffle(array)         #Перехід до масиву
    #Процес сортування і час сортування
start_time = time.time()      #Запуск часу старту
```

```

print 'Sorting array...'
array = MergeSort.mergesort(array)          #Сортування масиву
print 'Array sorted.'
time_taken = time.time() - start_time #Розрахунок і запис time_taken
print 'Time taken to sort is ', time_taken, 'seconds.'

```

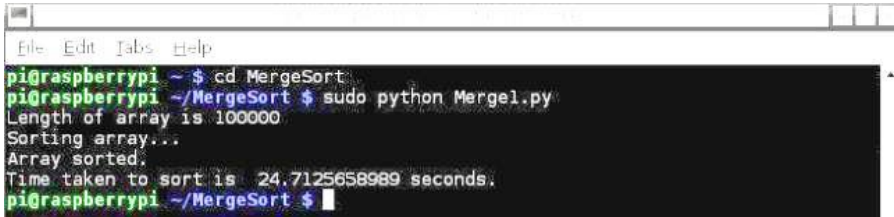


Рис.8.21 Реалізація алгоритму сортування злиттям на одному Raspberry Pi

Виконання завдання тільки на одному Raspberry Pi займає близько 24 секунд.

Тепер давайте розподілимо завдання між двома Raspberry Pi, але для цього ми спочатку повинні встановити IP-адреси кожного Pi.

На першому Pi через командний рядок налаштуємо його IP-адресу 192.168.1.1, і цей Pi буде виступати як сервер в мережі:

```

sudo ifconfig eth0 192.168.1.1 broadcast 192.168.1.255 netmask
255.255.255.0

```

Аналогічно, вводимо наступний другий Pi, налаштувавши його IP-адресу 192.168.1.2, і цей Pi буде виступати як клієнт:

```

sudo ifconfig eth0 192.168.1.2 broadcast 192.168.1.255 netmask
255.255.255.0

```

На першому Pi запустимо наступний код (MergeServer.py):

```

import socket
import MergeSort                                #Імпорт функцій злиття
import random
import time
    #Розбиваємо масив на p ділянок, де p - число процесорів
def breakarray(array, n):
    sectionlength = len(array)/n                #Довжина кожної секції
    result = []
    for i in range(n):
        if i < n - 1:
            result.append( array[ i * sectionlength : (i+1) * sectionlength ] )
            #Включаємо всі елементи, які залишилися, в останню секцію
        else:
            result.append( array[ i * sectionlength : ] )
    return result
    #Створюємо масив для сортування

```

```

arraylength = 100000                                #Довжина масиву для сортування
print 'Length of array is', arraylength
array = range(arraylength)                            #Створюємо масив
random.shuffle(array)                                #Перехід до
#Вказуємо інформація про процесори/комп'ютери
procno = 2 #Кількість процесорів
print 'Number of processors:', procno
procID = 0                                            #ID цього процесора(сервера)
addr_list = []                                       #Список адрес клієнтів
#Налаштовуємо мережу
HOST = ''
PORT = 50007
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind((HOST, PORT))
s.listen(procno - 1) #Слухаємо для (n) числа клієнтських підключень
print 'Waiting for client...'
for i in range(procno - 1):                          #Підключаємося до всіх клієнтів
conn, addr = s.accept()                              #Приймаємо з'єднання від клієнта
print 'Connected by', addr
addr_list.append(addr)                               #Додаємо адресу в список адрес
#Процес сортування і час початку розподілених обчислень
start_time = time.time()                             #Записуємо час старту
sections = breakarray(array, procno) #Розбиваємо масив на розділи для
кожного клієнта
for i in range(procno - 1): #Перетворення розділу масиву в рядок для
відправки
arraystring = repr(sections[i+1])
conn.sendto( arraystring , addr_list[i] ) #Посилаємо рядок масиву
print 'Data sent, sorting array...'
array = MergeSort.mergesort(sections[procID]) #Сортуємо розділ і
зберігаємо його в масиві
print 'Array sorted.'
for i in range(procno - 1): #Приймаємо сортовані розділи від кожного
клієнта
arraystring = ''
print 'Receiving data from clients...'
while 1:
data = conn.recv(4096)                               #Отримуємо дані в шматках
arraystring += data                                  #Додаємо дані в рядки масиву
if ']' in data:                                       #Коли закінчене приймання даних
break
print 'Data received, merging arrays...'
array = MergeSort.merge(array, eval(arraystring)) #Об'єднуємо розділи
поточного масиву від клієнтів
print 'Arrays merged.'
conn.close()time_taken = time.time() - start_time #Розраховуємо і
записуємо time_taken
print 'Time taken to sort is ', time_taken, 'seconds.'

```

Коли буде виведений рядок "Waiting for client..." ("В очікуванні клієнта...") в командному рядку першого Pi, запустить наступний код на другому Pi (MergeClient.py):

```

import socket
import MergeSort
HOST = '192.168.1.1'
PORT = 50007
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
    #Приймаємо рядки масиву в шматках
arraystring = ''
print 'Receiving data...'
while 1:
data = s.recv(4096) #Отримуємо дані в шматках
    #Друк даних
arraystring += data
    #Додаємо дані в рядок масиву
if ']' in data:
    #Коли закінчене приймання даних
break
array = eval(arraystring)
print 'Data received, sorting array...'
    #Сортуємо масив, який є розподіленим
array = MergeSort.mergesort(array)
print 'Array sorted, sending data...'
    #Перетворюємо рядок масиву, щоб відправити назад на сервер
arraystring = repr(array)
s.sendall(arraystring)
    #Посилаємо рядок масиву
print 'Data sent.'
s.close()

```

```

pi@raspberrypi ~ $ cd MergeSort
pi@raspberrypi ~/MergeSort $ sudo python MergeServer.py
Length of array is 100000
Number of processors: 2
Waiting for client...
Connected by ('192.168.1.2', 52820)
Data sent, sorting array...
Array sorted.
Receiving data from clients...
Data received, merging arrays...
Arrays merged.
Time taken to sort is 16.2228779793 seconds.
pi@raspberrypi ~/MergeSort $

```

Рис.8.22 Реалізація алгоритму на двох Raspberry Pi

Час, необхідний для сортування масиву, знизився приблизно до 16 секунд. Прискорення буде більш помітним, коли використати більшу кількість Raspberry Pi, підключивши їх через концентратор.

Сподіваємося, що ви отримали уявлення про те, наскільки близькими стають розподілені обчислення і як просто це здійснити. Є багато прикладів таких програм і, можливо, ви могли б почати свій власний проект розподілених обчислень за допомогою Raspberry Pi (і друга, або друзів)!

## 9. ПРОЕКТИ

### Робот-фотограф



Raspberry Pi - популярний одноплатний мікрокомп'ютер, який можна використовувати для вирішення різноманітних завдань. Робот-фотограф<sup>159</sup>, якому віддаються команди через web-інтерфейс - одне з досить цікавих і технічно складних завдань.

Робот (рис. 9.1) базується на само рухомій платформі і має можливість фото і відеозйомки.

Для робота можна сформувані такі завдання, як управління через WiFi, повороти вправо-вліво, рух вперед-назад, а також зйомка і відображення відео в web-інтерфейсі в процесі руху робота.

Основою для RaspCamBot слугує плата Raspirobot, де вже є всі

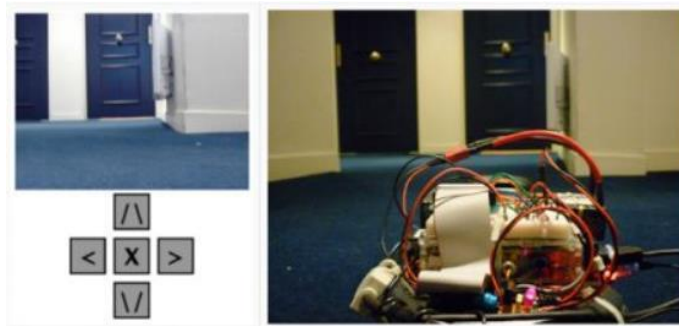
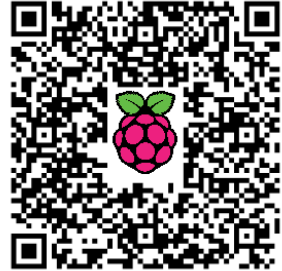


Рис.9.1 Робот-фотограф

необхідні апаратні компоненти для пересування макета в просторі. Для плати створено оптимізоване програмне забезпечення, в яке були інтегровані web-кнопки для керування камерою.

Для створення інтерфейсу використовуватиметься фреймворк WebIOPI, так як він дозволяє швидко запустити проект з незначним знанням php, nginx і прикладних програм.

#### Плата RaspiRobot

Raspberry Pi проблематично використовувати в якості контролера для різних вузлів робототехніки, так як в ньому недостатньо виходів, які можуть керувати силовим навантаженням. Однак можна підключити

<sup>159</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1774-robot-photographer-based-raspberry-pi>

Raspberry Pi до плати RaspiRobot і отримати повноцінний контроллер для двигунів. Використовуючи плату можна легко створити само рухому платформу з двома колекторними двигунами. Можливості плати:

- драйвер керування силовими вузлами
- стабілізатор напруги від батареї 6-12В для Raspberry Pi
- два двотактних виходи (25 мА)
- індикатори стану
- I2C шина
- відкрите API на Python

## Фреймворк WebIOPi

Фреймворк webiopі - закінчений продукт для роботи з GPIO портами (портами вводу-виводу) в Raspberry Pi. Webiopі може керувати станом усіх портів GPIO віддалено або локально, використовуючи браузер або будь-яку іншу програму.

Можливості:

- сервер, створений з використанням мови Python
- підтримка Serial, 1-Wire, SPI, GPIO
- підтримка близько 35 таких пристроїв, як датчики, DAC, ADC ...
- сумісність з версіями Python 2.7 та 3.2
- легка адаптація під потреби користувачів
- аутентифікація за допомогою логіна і пароля
- наявні багато прикладів використання фреймворка

## Інсталяція WebIOPi

Для інсталяції WebIOPi необхідний встановлений Python версії 2.7 або 3.2.1. Інсталяція виконується такими командами з консольного терміналу:

```
$ curl -o webio-pi.tar.gz
http://store.raspberrypi.com/checkout/project/7203c447-49fa-
4fb7-b9dd-317a907b65df
$ tar xvzf webio-pi.tar.gz
$ cd webio-pi
$ sudo ./setup.sh
```

Далі необхідно запустити фреймворк в командному терміналі:

```
$ Sudo webiopі [-c] [-s] [port]
```

Однак, після завершення сценарію поєднанням клавіш Ctrl+C або закриттям вікна терміналу, сервер webiopі буде закритий і стан портів втратиться. Тому webiopі доцільно запускати як демон (сервіс).

Запуск:

```
$ sudo service webiopi start
```

Зупинка:

```
$ sudo service webiopi stop
```

### Перевірка працездатності

Після запуску webiopi, необхідно перейти в браузері за посиланням виду `http://[RaspIP]:8000`, де [RaspIP] потрібно замінити реальною IP-адресою Raspberry Pi. Далі треба ввести дані для авторизації - ім'я користувача "webiopi" і пароль "raspberry". Для коректної роботи автор фреймворка рекомендує використовувати актуальні версії браузерів Firefox, Safari або Chrome, браузер Internet Explorer підтримується не повністю. Приклад інтерфейсу

відображення стану портів вводу-виводу показаний на рис.9.2:

Використовуючи web-інтерфейс можна встановити значення на виході будь-якого порту, а також задати режим його роботи. Для цього достатньо натиснути на потрібний порт і у спливаючому меню вибрати режим роботи і необхідні параметри вихідної напруги.

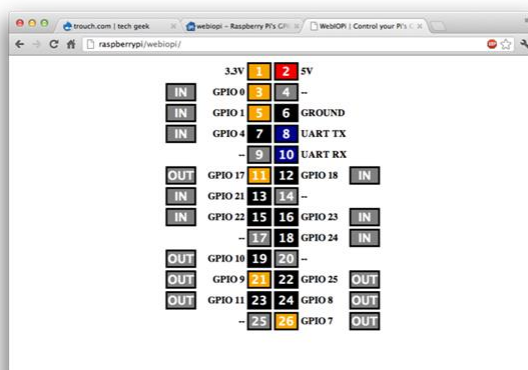


Рис.9.2 Відображення стану портів вводу-виводу

### Управління web-камерою

Створення робота базується на відкритому проекті по збірці моторизованого робота з web-камерою - CamBot.

Управління web-камерою виконується за допомогою консольних програм `fswebcam` і `mjpg-streamer`, для яких потрібно створити сценарії ініціалізації фотографування і початку/зупинки відеозапису.

Сценарій для фотографування:

```
#!/bin/sh
savePath = "/usr/share/webiopi/htdocs/app/RaspCamBot/photo.jpg"
fswebcam -p MJPEG -r 800 x600 -d /dev/video0 --jpeg 90 --shadow
--title "raspbot" --subtitle "Photo camera" -save -quiet
$savePath
```

Сценарій початку відеозапису з web-камери:

```
#!/bin/sh
DEV_PATH = "/dev/video0"
PRG_STREAM=mjpg_streamer
MAX_RESOLUTION=320x240          #160x120
ACTIVE_PORT=8001
CAMERA=/dev/video0
#Спроба знайти підключену web-камеру
if [ ! -e $DEV_PATH ]; then
    exit 1
fi
FRAMERATE=15
PLUGIN_PATH=/home/adm/streamer-r65
"$PLUGIN_PATH/$PRG_STREAM" -f $FRAMERATE -i
"$PLUGIN_PATH/input_uvc.so" -d $CAMERA -r $MAX_RESOLUTION -o
"$PLUGIN_PATH/output_http.so" -p $HTTP_PORT -b
```

## Web-інтерфейс

Створення web-інтерфейсу для фреймворка webіорі досить спрощене. Після повного завантаження сторінки браузером, необхідно додати свої елементи керування за допомогою виклику спеціалізованих API на мові javascript. Далі треба підключити обробник подій до кожного елементу керування. Такі обробники викликатимуть макроси, що написані мовою Python і зберігаються на сервері.

Приклад створення кнопки для пересування вперед:

```
var handler = webіорі();
var action = "bt_up", begin_func = go, end_func = stop;
var control = handler.createButton(action, "/\\",
begin_func,end_func);
$("#div_up_place").append(control);
Функції stop і go:
function go() {
    handler.callMacro("go");
}
function stop() {
    handler.callMacro("stop");
}
```

## Написання сценарію на Python

Python-сценарій для webіорі повинен містити обов'язкову функцію `setup` і, опціонально, функцію `destroy`. Функція `setup` виконується при ініціалізації сценарію і використовується для встановлення значень на портах вводу/виводу і визначення режиму їх роботи. Функція `destroy`

викликається по закінченню виконання сценарію і використовується для повернення портів та інших параметрів системи в початковий стан.

Приклад реалізації функції `go`:

```
@webiopi.macro
def go():
    left_forward()
    right_forward()
```

Для початку або зупинки запису web-камери використовується команда `call`, яка виконує відповідний shell-сценарій:

```
def record_start():
    return_cmd = call("/home/adm/RaspCamBot/stream.sh")
```

Як бачимо, створити робота і керувати ним не так вже й складно.

### Настінний календар Google



Замість того, щоб використовувати паперовий календар з крихітними квадратиками, на яких щось записувати, можна зробити так, щоб на стіні висів ваш календар Google<sup>160</sup>.

Щоб реалізувати наведену нижче інструкцію ви повинні мати загальне уявлення про домашню мережу і комп'ютери, деякий досвід роботи на Linux також не був би зайвим, але не настільки необхідний. Якщо зіткнетеся з чимось, що не розумієте, то просто пам'ятайте, що вашим другом є пошук Google.



Вам знадобиться устаткування:

- Домашня мережа (бездротова, якщо ви не можете провести кабель до Pi)
- Raspberry Pi (RPi) (автор використав модель B)
- SD-карта 4 Гб або більше
- Адаптер змінного струму (автор використовував стаціонарний зарядний пристрій USB для мобільних телефонів)
- Кабель micro USB
- USB-клавіатура і миша
- USB-адаптер бездротової мережі (переконайтеся, що він сумісний з RPi<sup>161</sup>)

<sup>160</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1772-wall-nalendar-google-on-raspberry-pi>

<sup>161</sup> [http://elinux.org/RPi\\_VerifiedPeripherals#USB\\_Wi-Fi\\_Adapters](http://elinux.org/RPi_VerifiedPeripherals#USB_Wi-Fi_Adapters)

- Кабель HDMI
- Настінний монітор з підтримкою HDMI або будь-який монітор з яким-небудь конвертером HDMI
- Настінний кронштейн для монітора

### Крок 1: Встановлення та налаштування Raspberry Pi



Рис.9.3 Настінний календар Google

Спочатку налаштуємо RPi, а для цього вам необхідно встановити Raspbian.

Тепер, коли у вас є встановлений Raspbian, пора запустити RPi. Зміни, які повинні зробити при першому завантаженні:

- Розширити файлову систему, щоб Raspbian використовувала всю карту SD
- Змінити пароль
- Включити завантаження робочого столу
- Встановити мову, регіон і часовий пояс

У додаткових опціях:

- Змінити ім'я хоста, так щоб ви могли знайти свій RPi в мережі.
- Включити SSH, щоб могли отримати доступ до RPi з іншого комп'ютера у своїй мережі.
- Вибрати Finish, для необхідного перезавантаження RPi.

### Крок 2: Клавіатура і оновлення

Якщо хочете змінити розкладку клавіатури, наприклад, на США, то треба змінити відповідний файл, відкривши термінал і ввівши наступну команду:

```
sudo nano /etc/default/keyboard
```

Використовуйте клавіші зі стрілками для переміщення курсору і змініть gb на us (рис.9.4).

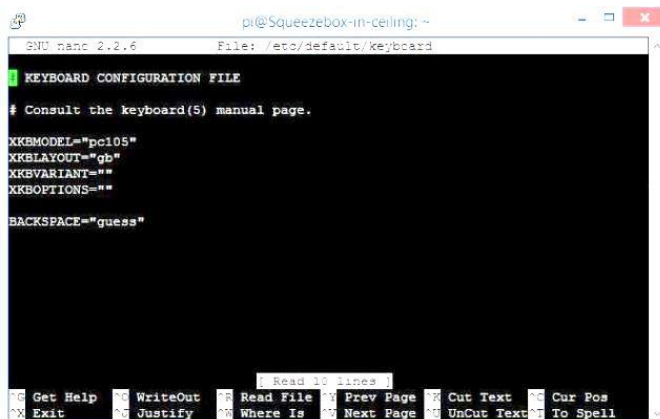


Рис.9.4 Зміна розкладки клавіатури

Тепер збережіть файл, натиснувши Ctrl+X і Y, щоб зберегти зміни. Оновіть RPi, ввівши наступні команди:

```
sudo apt-get update
sudo apt-get upgrade
```

Натисніть Y і Enter для завантаження оновлень (це займе певний час, тож можете випити кави).

### Крок 3: Налаштування бездротової мережі

Якщо хочете використовувати лише Ethernet, то можете пропустити цей крок.

Щоб ваш модуль Wi-Fi працював, можливо, буде необхідно змінити файл `wpa_supplicant.conf` за допомогою nano:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

щоб він виглядав десь так:

```
ctrl_interface=/var/run/wpa_supplicant
network={
    proto=RSN
    scan_ssid=1
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP
    ssid="your network ID in quote marks"
    psk="your network password in quote marks"
```

```
}
update_config=1
```

Для збереження файла натисніть Ctrl+X та Y і Enter.  
Перезапустіть, ввівши:

```
sudo reboot
```

Побачимо, яку IP-адресу має ваш модуль Wi-Fi, з командою:

```
sudo ifconfig
```

Повинні отримати щось на зразок цього:

```
wlan0 Link encap:Ethernet HWaddr 64:66:b3:06:43:1b
inet addr:10.0.0.75 Bcast:10.0.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:912384 errors:0 dropped:121692 overruns:0 frame:0
TX packets:706463 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:694114055 (661.9 MiB) TX bytes:71017681 (67.7 MiB)
```

#### Крок 4: Iceweasel

Тепер ми встановимо браузер, який відображає календар Google належним чином. Iceweasel є похідним Firefox від Mozilla, а щоб його встановити введіть таку команду:

```
sudo apt-get install iceweasel
```

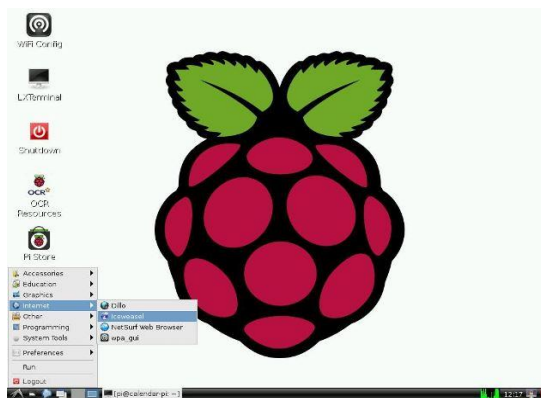


Рис.9.5 Запуск браузера Iceweasel з меню

Введіть Y і натисніть Enter для завершення завантаження та встановлення. Після завершення в меню Інтернет повинен з'явитися Iceweasel (рис.9.5).

Відкрийте його, щоб можна було приступити до налаштування. Спершу відкрийте свій календар Google (рис.9.6) і збережіть пароль - в цьому випадку ви будете в цей же момент зареєстровані.

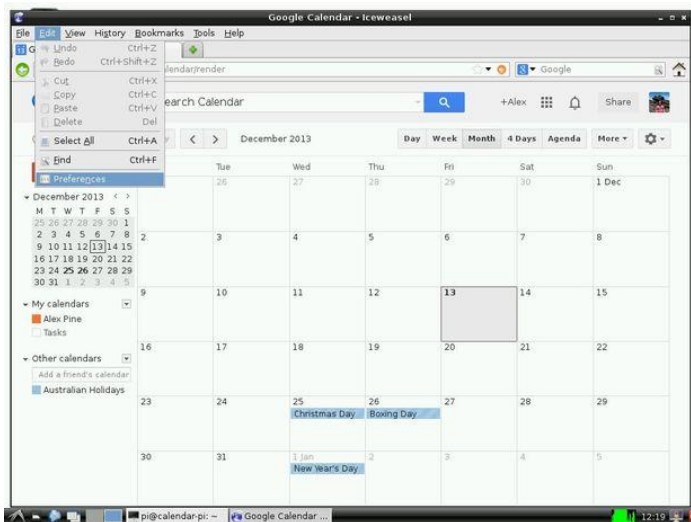


Рис.9.6 Вибір меню Preferences

Тепер встановіть Google Calendar стартовою сторінкою (рис.9.10).

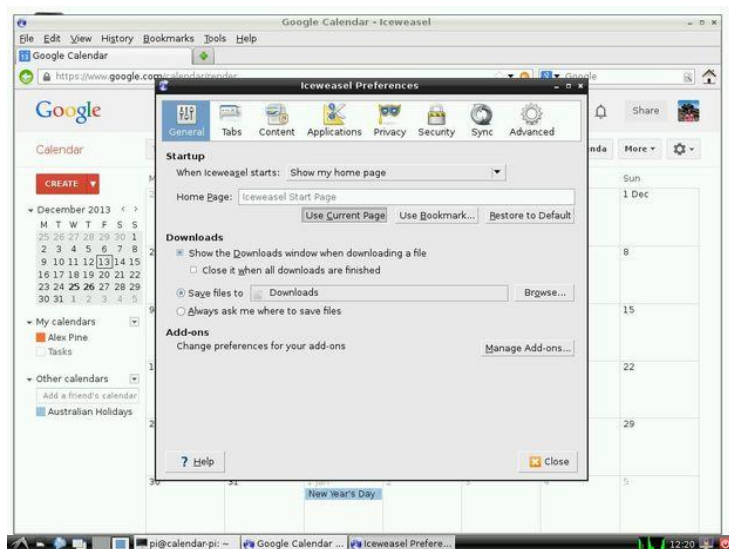


Рис.9.10 Встановлення Google Calendar стартовою сторінкою

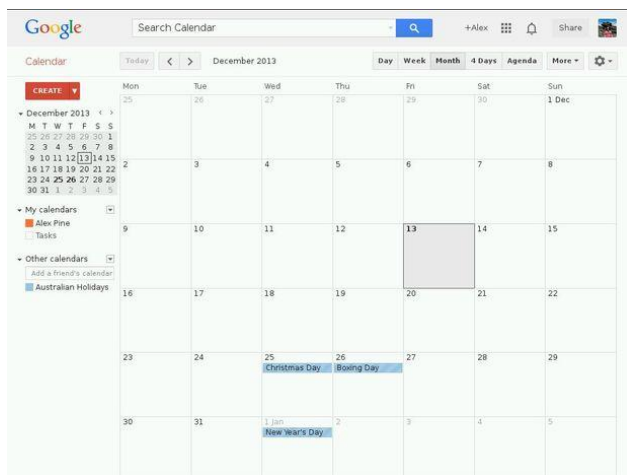


Рис.9.11 Завершене встановлення календаря

Тепер нам потрібно відключити відновлення сторінок після аварії в разі відключення електроенергії, бо календар Google може не відображатися сам по собі, що буде дратувати, якщо у вас немає підключених клавіатури/миші.

Введіть `about:config` в полі адреси і натисніть Enter.

Знайдіть рядок `"browser.sessionstore.resume_from_crash"` (рис.9.12) і двічі клацніть, щоб змінити його на `false`.

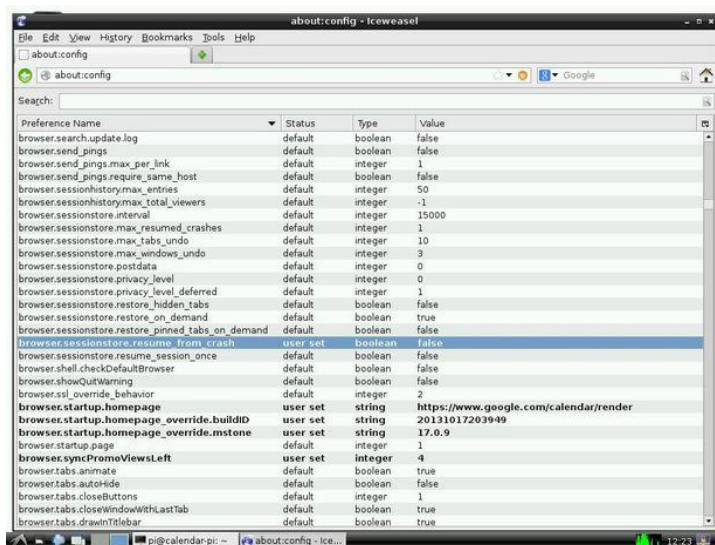


Рис.9.12 Відключення відновлення сторінок

Натисніть клавішу F11, щоб перейти до повного екрану і наведіть курсор миші на самий верх екрана, а потім закрийте браузер, коли з'явиться **x** у верхньому правому куті.

Повторно відкрийте браузер: він повинен відкрити календар Google і залишатися в повноекранному режимі.

Також ми хочемо автоматичний запуск Iceweasel, тому потрібно змінити параметри автозапуску. Для цього, використовуючи редактор nano, додайте @Iceweasel до списку:

```
sudo nano /etc/xdg/lxsession/LXDE/autostart
```

Тепер натисніть Ctrl+X і Y та Enter для збереження змін.

### Крок 5: Курсор та енергозбереження

Наступним кроком необхідно позбутися такої неприємності, як курсор миші і заборонити екрану переходити в режим енергозбереження/відключення.

Спочатку ми встановимо Unclutter, щоб позбутися курсора, коли він не використовується:

```
sudo apt-get install unclutter
```

Тепер потрібно відредагувати файл /etc/lightdm/lightdm.conf, щоб заборонити режим енергозбереження/сну:

```
sudo nano /etc/lightdm/lightdm.conf
```

Перейдіть вниз до [SeatDefaults] і змініть цей рядок:

```
#xserver-command=X
```

на такий:

```
xserver-command=X -s 0 -dpms
```

Знову натисніть Ctrl+X і Y та Enter для збереження змін.

### Крок 6: Ми це зробили!

Тепер зроблені всі налаштування і можете повісити свій монітор на стіну та все це включити.

Якщо щось перестає працювати, відключіть RPi, а потім знову підключіть його до розетки, вуаля - він знову запустився.

Можна також використати старий VGA-монітор, але доведеться знайти конвертор HDMI до VGA.

## Платформа для Minecraft



Все більшої популярності сьогодні набувають прості дешеві базові платформи, які легко інтегруються у великі системи і можуть бути використані для

різноманітних розширень, наукової роботи, навчання, технологічних рішень тощо.

Для таких цілей був створений і розроблений прото-комп'ютер RaspberryPi.

Крім вищезазначеного використання, ентузіасти Raspberry Pi спромоглись використовувати його як ігрову платформу, в тому числі як платформу для Minecraft<sup>162</sup>(рис.9.13).

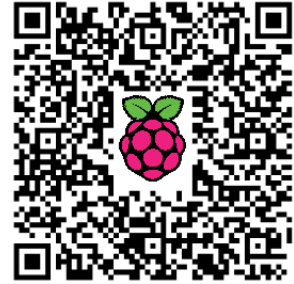


Рис.9.13 Minecraft

Для побудови платформи для гри Minecraft на Raspberry Pi необхідно встановити відповідне програмне забезпечення, яке наразі можна легко знайти в Інтернеті. Раніше було розглянуто, як почати роботу з Raspberry Pi та встановлювати додатки на Raspberry Pi, тому наступні дії широко коментувати не будемо.

### Установка та налаштування сервера Minecraft на Raspberry Pi

Рекомендуємо для встановлення скористатися SpigotMC - легким і стабільним сервером збірки Minecraft, який чудово працює на Raspberry Pi. Для його використання необхідно мати копію коду, завантажену за допомогою:

```
sudo wget http://ci.md-5.net/job/Spigot/lastSuccessfulBuild/artifact/Spigot-server/target/spigot.jar
```

<sup>162</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1779--platform-for-minecraft-on-raspberry-pi>

Після успішного завершення завантаження, введіть наступну команду:

```
sudo /opt/jdk1.8.0/bin/java -Xms256M -Xmx496M -jar  
/home/pi/spigot.jar nogui
```

При цьому, якщо запускаєте команду на Raspberry Pi версії A (256 МБ RAM), то необхідно змінити розрядність з 256 і 496 у наведеній вище команді на 128 і 256, відповідно.

Тепер ваш сервер буде запущений, але не одразу. Будьте готові чекати приблизно 3-6 хв. для процесу встановлення сервера і створення карти. Наступні стартапи займатимуть набагато менше часу (близько 20-30 секунд). Запуск сервера відображається аналогічно його запуску на звичайному комп'ютері. Натиснувши на Multiplayer, ви повинні побачити сервер (рис.9.14):



Рис.9.14 Запуск Minecraft Server

### Використання модуля minecraftstuff для побудови платформи Minecraft

Крім цього, на Raspberry Pi можна самому створювати 2D та 3D лінії та фігури для побудови нових об'єктів в Minecraft. Це стало можливим завдяки розширенню MinecraftDrawing модуля minecraftstuff. Для користування ним, необхідно імпортувати модуль minecraftstuff в код:

```
#імпорт модуля minecraftstuff.py  
import minecraft.minecraftstuff as minecraftstuff
```

Після підключення до Minecraft можете створити об'єкт MinecraftDrawing, передаючи в Minecraft наступні зв'язки:

```
# підключення до minecraft  
mc = minecraft.Minecraft.create()  
  
#створення об'єкта minecraft та його передача зв'язкам minecraft  
mcdrawing = minecraftstuff.MinecraftDrawing(mc)
```

Крім цього, клас малюнків Minecraft підтримує наступні функції:

- drawLine
- drawSphere
- drawCircle
- drawFace
- 

# для того, щоб викликати drawLine і передати йому  $x, y, z$  координати початку і кінця лінії

```
mcdrawing.drawLine(x1,y1,z1,x2,y2,z2,blockType,blockData)
```

# для того, щоб викликати drawSphere і передати йому координати центра і радіус сфери

```
mcdrawing.drawSphere(x,y,z,radius,blockType,blockData)
```

# для того, щоб викликати drawCircle і передати йому координати центра і радіус кола

```
mcdrawing.drawCircle(x,y,z,radius,blockType,blockData)
```

# виклик drawFace для створення будь-якої плоскої форми

Для креслення заповнених фігур можна використовувати відповідні атрибути. Якщо ми передаємо атрибут True, то ми заповнюємо весь простір між точками (рис.9.15), а якщо передаємо атрибут False, то ми окреслюємо лише каркас фігури.

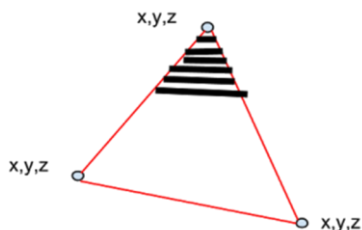


Рис.9.15 Побудова фігур для платформи

# для того, щоб накреслити декілька точок

```
shapePoints = []
```

```
shapePoints.append(minecraft.Vec3(x1,y1,z1))
```

```
shapePoints.append(minecraft.Vec3(x2,y2,z2))
```

```
shapePoints.append(minecraft.Vec3(x3,y3,z3))
```

# для того, щоб накреслити декілька точок разом і наповнити їх атрибутом True

```
mcdrawing.drawFace(shapePoints, True, block.GOLD_BLOCK.id)
```

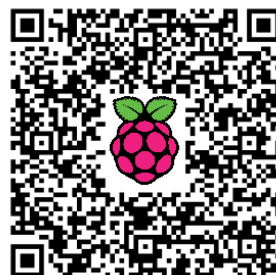
Побудова додаткових об'єктів не лише ріноманітнить Minecraft, але й дозволяє отримати певні навички програмування.

## Запустіть свою власну радіостанцію з Raspberry Pi



Raspberry Pi використовується для різних чудових проектів. Ви навіть можете використати цей маленький прилад як передавач FM<sup>163</sup> і зробити це без особливих труднощів.

Учасник Code Club pihack написав програму, яка використовує апаратні можливості Raspberry Pi з генерації розширеного спектру тактових сигналів, щоб замість них отримати вихідні сигнали FM-радіо.



## Передумови

Крім звичних аудіо, Ethernet, HDMI, і USB-портів на Raspberry Pi, пристрій також має інтерфейси, які призначені для підключення безпосередньо до інших чіпів і модулів.

Це раніше розглянуті порти загального призначення для введення/виводу (General Purpose Input/Output - GPIO), які мають 26 контактів (розташовані в 2×13 ряди), які ви бачите на платі. Ці інтерфейси не підключені, щоб одразу працювати, але ними можна керувати за допомогою програмного забезпечення.

Для передачі напрочуд потужного сигналу FM, все, що треба, це прикріпити провід до 4 контакту GPIO (рис.9.16). Навіть без проводу FM-сигнал трансляції з Raspberry Pi буде підхоплений довколишніми FM-приймачами.

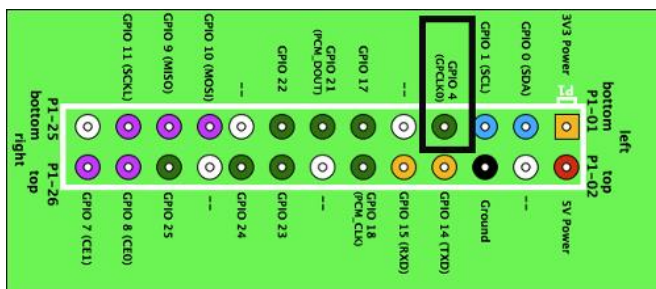


Рис.9.16 Контакти GPIO

**Застереження:** Закони для мовлення FM-сигналів варіюються від країни до країни. Будь ласка, ознайомтеся з місцевими правилами, перш ніж намагатися повторити цей проект.

**Отримати, встановити, запустити**

<sup>163</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1788-start-your-own-radio-station-with-raspberry-pi>

Подайте живлення на Pi і перейдіть до командного рядка через SSH або безпосередньо ввійшовши в пристрій.

Тепер ми захопимо код, написаний на Code Club hackfest, і витягнемо його:

```
mkdir ~/pifm
cd pifm
wget http://omattos.com/pifm.tar.gz
tar zxvf pifm.tar.gz
```

Каталог в даний час містить шість файлів. Дивно, що все є на них. Тепер ви можете транслювати доданий файл "sound.wav":

```
sudo ./pifm sound.wav 102.3
```

Візьміть приймач FM, встановіть його на FM 102,3 і почуєте музику на тему Star Wars. Ви можете змінити частоту мовлення на будь-яку в проміжку між 88 МГц і 108 МГц шляхом простого додавання частоти каналу в кінці команди.



Рис.9.17 Приймач FM на частоті 102,3 МГц

## Трансляція пісень

Можете також програвати й інші аудіо-файли, але вони повинні бути 16-бітні 22050 Гц моно і тільки у форматі WAV. Це може бути схоже на реальне обмеження, але в дійсності це не так. Все, що вам потрібно, це аудіо редактор SoX звукового обміну, який буде обробляти ваш MP3-файл, на льоту перетворюючи його в файл WAV.

Встановіть звуковий редактор і його залежності за допомогою:

```
sudo apt-get install sox libsox-fmt-all
```

Коли це буде зроблено, введіть наступну команду, замінивши "SomeSong.mp3" на ім'я MP3-файлу, який хочете програвати:

```
sox -t mp3 SomeSong.mp3 -t wav -r 22050 -c 1 - | sudo ./pifm - 102.3
```

Перша частина команди переводить MP3-файл в WAV-файл, змінює його частоту дискретизації аудіо на 22050 і мікшує трек в моно. Перетворений трек потім відправляється на стандартний вихід, який позначається знаком дефіс (-), а потім через труби (|) - в стандартний потік введення команди `pifm`.

Єдина відмінність в команді `pifm` у наведеному вище прикладі, що замість вказівки імені файлу для трансляції, ми запросили сценарій замість транслювання стандартного входу.

Якщо у вас є FM-приймач, налаштований на частоту 102.3, то тепер повинні почути свій MP3! Знову ж, як і в оригінальному прикладі, не бійтеся змінити частоту на іншу в проміжку між 88 МГц і 108 МГц.

### Трансляція підкастів

Ви можете зробити деякі дивовижні речі з SoX. Можете, наприклад, використовувати його для трансляції своїх улюблених потоків з Інтернету. Щоб транслювати Linux Voice підкаст:

```
sox -t mp3 http://www.linuxvoice.com/episodes/lv_s02e01.mp3 -t wav -r 22050 -c 1 - | sudo ./pifm - 102.3
```

Єдина різниця між цією командою і попереднім прикладом, що замість того, щоб вказувати на локальний MP3, ви зараз вказуєте на аналогічний, який знаходиться в Інтернеті.

Таким же чином, можна транслювати онлайн радіостанцію. Багато станцій опублікували M3U-файл, і ви можете вказати на нього:

```
sox -t mp3 http://www.ndr.de/resources/metadaten/audio/m3u/ndrloop5.m3u -t wav -r 22050 -c 1 - | sudo ./pifm - 102.3
```

Не всі станції транслюють MP3-потоки. Деякі з них використовують інші формати, наприклад, такі як OGG. Все, що потрібно змінити, це тип вхідного файлу, який конвертуєте з SoX подібним чином:

```
sox -t ogg http://network.absoluteradio.co.uk/core/audio/ogg/live.pls?service=a6bb -t wav -r 22050 -c 1 - | sudo ./pifm - 102.3
```

Якщо шукаєте онлайн-радіо, то в онлайні є величезний список європейських радіостанцій<sup>164</sup>, які передають в Інтернеті.

---

<sup>164</sup> <http://www.listenlive.eu/>

## Перекладач з розпізнаванням мови та відтворенням (понад 60 мов)



З дуже невеликим зусиллям, ми можемо перетворити наш міні-комп'ютер на багатофункціональний мовний перекладач<sup>165</sup>, який не тільки підтримує розпізнавання голосу та відтворення носія мови.

Він також здатний на динамічну трансляцію між 1000 мовних пар, **безкоштовно!** Навіть якщо ви не зацікавлені в створенні саме такого інструменту перекладу, є ще багато частин в цьому розділі, які можуть зацікавити вас (розпізнавання мови, перетворення тексту в мову, API перекладу від Microsoft/Google). Почнемо зі списку покупок, хоча більшість читачів, ймовірно, вже має багато з цих предметів.

Список покупок:

| Кільк.                    | Необхідні елементи                               | Ціна(грн)*            |
|---------------------------|--------------------------------------------------|-----------------------|
| 1                         | <a href="#">Raspberry Pi</a>                     | 490.00                |
| 1                         | <a href="#">Мікро USB кабель</a>                 | 28.00                 |
| 1                         | <a href="#">Logitech USB Headset</a>             | 315.00                |
| 1                         | <a href="#">SD-карта</a> (10 клас і 4Гб мінімум) | 70.00                 |
|                           |                                                  | <b>Всього: 903.00</b> |
| <b>Додаткові елементи</b> |                                                  |                       |
| 1                         | <a href="#">Блок живлення</a>                    | 70.00                 |
| 1                         | <a href="#">HDMI кабель</a>                      | 51.00                 |
| 1                         | <a href="#">Корпус</a>                           | 105.00                |

Рис.9.18 Список необхідного для перекладача

\*Є безумовно дешевші варіанти USB-гарнітури, але автор вибрав Logitech, бо вона зручна у використанні. Для альтернативи, перегляньте список для перевірки, які звукові карти підтримуються Raspberry Pi<sup>166</sup>.

### Припущення

Це керівництво припускає, що ваш Raspberry Pi вже має:

- встановлену останню версію Raspbian
- підключення до Інтернету
- відповідні драйвери звукової карти для гарнітури

<sup>165</sup> <http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1797-translator-on-raspberry-pi-with-speech-recognition-and-reproduction-over-60-languages>

<sup>166</sup> [http://elinux.org/RPi\\_VerifiedPeripherals#USB\\_Sound\\_Cards](http://elinux.org/RPi_VerifiedPeripherals#USB_Sound_Cards)

## Налагодження та тестування гарнітури

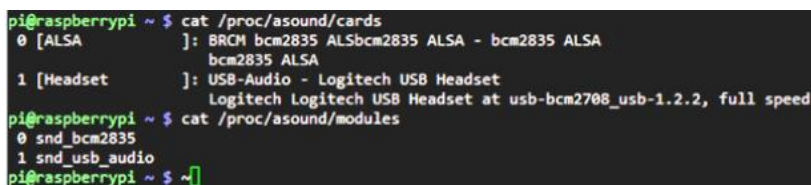
Перш ніж почати писати код, переконайтеся, що можете записувати і відтворювати звук за допомогою вибраної гарнітури USB. Найпростіший спосіб зробити це - за допомогою вбудованих в Linux команд "arecord" і "aplay". Але спочатку давайте переконаємося, що наша файлова система знаходиться в актуальному стані:

```
sudo apt-get update
sudo apt-get upgrade
```

Тепер, підключіть гарнітуру USB і виконайте наступні команди:

```
cat /proc/asound/cards
cat /proc/asound/modules
```

Ви повинні побачити, що Logitech Headset вказаний як карта 1. Крім того, друга команда повинна показати, що драйвер для карти 0 (вихід за замовчуванням raspberry pi) є `snd_bcm2835` і драйвер для карти 1 (наш Logitech Headset) є `snd_usb_audio` (рис.9.19):



```
pi@raspberrypi ~ $ cat /proc/asound/cards
0 [ALSA                ]: BCM2835 ALSAbcm2835 ALSA - bcm2835 ALSA
bcm2835 ALSA
1 [Headset              ]: USB-Audio - Logitech USB Headset
Logitech Logitech USB Headset at usb-bcm2708_usb-1.2.2, full speed
pi@raspberrypi ~ $ cat /proc/asound/modules
0 snd_bcm2835
1 snd_usb_audio
pi@raspberrypi ~ $
```

Рис.9.19 Перевірка вибраної гарнітури

Маємо проблему, бо скріншот показує, що за замовчуванням Raspberry Pi передає звук через свої вбудовані апаратні засоби, і не має налаштованого пристрою введення звуку. Щоб вирішити цю проблему, нам необхідно оновити ALSA (Advanced Linux Sound Architecture), щоб використовувати нашу гарнітуру за замовчуванням для введення і виведення звуку. Це може бути зроблено шляхом швидкої зміни в конфігураційному файлі ALSA, який знаходиться в `/etc/modprobe.d/alsa-base.conf`:

```
sudo nano /etc/modprobe.d/alsa-base.conf
```

Ближче до кінця цього файлу, змініть рядок, який виглядає як

```
options snd-usb-audio index=-2
```

на

```
options snd-usb-audio index=0
```

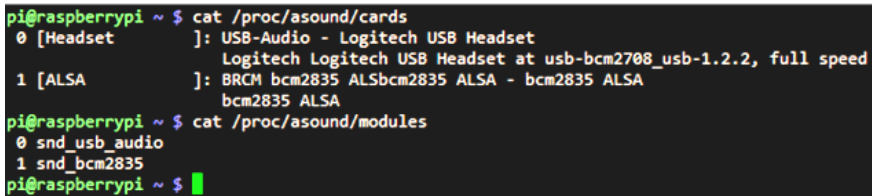
Збережіть і закрийте файл та перезавантажте Raspberry Pi за допомогою наступної команди:

```
sudo reboot
```

Після того, як повернетеся в Інтернет, звукова система повинна перезавантажитися, і коли ми повторно виконаємо наведені вище команди:

```
cat /proc/asound/cards
cat /proc/asound/modules
```

то повинні побачити, що тепер пристрій вводу/виводу за замовчуванням гарнітура USB (карта 0), як показано (рис.9.20):



```
pi@raspberrypi ~ $ cat /proc/asound/cards
0 [Headset      ]: USB-Audio - Logitech USB Headset
                    Logitech Logitech USB Headset at usb-bcm2708_usb-1.2.2, full speed
1 [ALSA        ]: BRCM bcm2835 ALSAbcm2835 ALSA - bcm2835 ALSA
                    bcm2835 ALSA
pi@raspberrypi ~ $ cat /proc/asound/modules
0 snd_usb_audio
1 snd_bcm2835
pi@raspberrypi ~ $
```

Рис.9.20 Пристрій вводу/виводу гарнітура USB

Тепер ми можемо перевірити це за допомогою запису 5 секундного кліпу з мікрофону:

```
arecord -d 5 -r 48000 daveconroy.wav
```

і відтворити його через динаміки гарнітури:

```
aplay daveconroy.wav
```

Для налаштування рівнів можна використовувати вбудовану утиліту alsamixer. Цей інструмент обробляє як аудіовхід, так і вихідні рівні:

```
sudo alsamixer
```

Тепер, коли наша гарнітура налаштована, можемо перейти до наступного кроку перетворення промови в текст.

## Мова в текст або розпізнавання мови з Raspberry Pi

Є кілька варіантів для розпізнавання мови з RPI, але думаємо, що кращим рішенням для цього розділу буде використання сервісу Google's

Speech to Text. Ця послуга дозволяє завантажити файл, який ми щойно записали, і перетворити його в текст (який ми пізніше будемо використовувати для перекладу).

Давайте створимо свій сценарій для обробки цього процесу:

```
sudo nano stt.sh
```

з наступним вмістом:

```
echo "Recording your Speech (Ctrl+C to Transcribe)"
arecord -D plughw:0,0 -q -f cd -t wav -d 0 -r 16000 | flac - -f
--best --sample-rate 16000 -s -o daveconroy.flac;

echo "Converting Speech to Text..."
wget -q -U "Mozilla/5.0" --post-file daveconroy.flac --header
"Content-Type: audio/x-flac; rate=16000" -O -
"http://www.google.com/speech-api/v1/recognize?lang=en-
us&client=chromium" | cut -d\" -f12 > stt.txt

echo "You Said:"
value=`cat stt.txt`
echo "$value"
```

Зробіть його виконуваним:

```
sudo chmod +x stt.sh
```

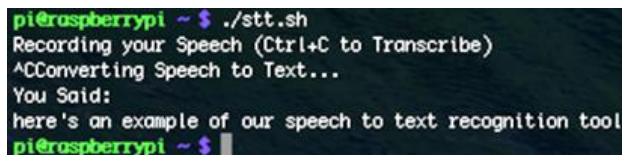
Останній крок перед тим, як запустити сценарій, це встановити FLAC кодек, який не включений в стандартний образ Raspbian:

```
sudo apt-get install flac
```

Тепер можемо сценарій запустити:

```
./stt.sh
```

Він автоматично розпочинає запис голосу, просто натисніть Ctrl+C, коли почнете говорити. У цей момент сценарій завантажує звуковий файл в Google, там він транскрибується і повертається, щоб міг бути відображений на нашому екрані. Вражаючий результат з лише кількома рядками коду! Приклад виведення (рис.9.21):



```
pi@raspberrypi ~ $ ./stt.sh
Recording your Speech (Ctrl+C to Transcribe)
^CConverting Speech to Text...
You Said:
here's an example of our speech to text recognition tool
pi@raspberrypi ~ $
```

Рис.9.21 Приклад виведення тексту

## Переклад від Microsoft і перетворення тексту в мову від Google

Тепер, коли ми можемо записувати свій голос і перетворювати його в текст, ми повинні перевести його на бажану нам іноземну мову. Автор хотів би мати можливість використовувати інструмент Google's Translate<sup>167</sup> для цього, але, на жаль, необхідно заплатити 20\$ за реєстрацію, щоб використовувати цей API. Можливо, ви купите його для себе, але автор хотів би, щоб даний проект був безкоштовним і кожен мав можливість його повторити.

Як альтернативу, будемо використовувати сервіс Microsoft's translate<sup>168</sup>, який в даний час, як і раніше, безкоштовний для публічного використання. Список підтримуваних мов і їх відповідних кодів можна знайти за посиланням<sup>169</sup>. У нашому попередньому прикладі ми використовували простий сценарій, але для процесу перекладу і відтворення автор написав більш потужний сценарій на python.

Весь цей код можна знайти в сховищі автора на github<sup>170</sup>.

Давайте спочатку створимо файл:

```
sudo nano PiTranslate.py
```

і додамо наступний вміст:

```
import json
import requests
import urllib
import subprocess
import argparse

parser = argparse.ArgumentParser(description='This is a demo
script by DaveConroy.com.')
parser.add_argument('-o', '--origin_language', help='Origin
Language', required=True)
parser.add_argument('-d', '--destination_language',
help='Destination Language', required=True)
parser.add_argument('-t', '--text_to_translate', help='Text to
Translate', required=True)
args = parser.parse_args()

## show values ##
print ("Origin: %s" % args.origin_language )
print ("Destination: %s" % args.destination_language )
print ("Text: %s" % args.text_to_translate )
```

<sup>167</sup> <https://developers.google.com/translate/>

<sup>168</sup> <http://www.bing.com/translator/>

<sup>169</sup> <http://msdn.microsoft.com/en-us/library/hh456380.aspx>

<sup>170</sup> <https://github.com/dconroy/PiTranslate>

```
text = args.text_to_translate
origin_language=args.origin_language
destination_language=args.destination_language

def speakOriginText(phrase):
    googleSpeechURL =
"http://translate.google.com/translate_tts?tl="+ origin_language
+"&q=" + phrase
    subprocess.call(["mplayer",googleSpeechURL], shell=False,
stdout=subprocess.PIPE, stderr=subprocess.PIPE)

def speakDestinationText(phrase):
    googleSpeechURL =
"http://translate.google.com/translate_tts?tl=" +
destination_language + "&q=" + phrase
    print googleSpeechURL
    subprocess.call(["mplayer",googleSpeechURL], shell=False,
stdout=subprocess.PIPE, stderr=subprocess.PIPE)

args = {
    'client_id': '',#your client id here
    'client_secret': '',#your azure secret here
    'scope': 'http://api.microsofttranslator.com',
    'grant_type': 'client_credentials'
}

oauth_url =
'https://datamarket.accesscontrol.windows.net/v2/OAuth2-13'
oauth_junk =
json.loads(requests.post(oauth_url,data=urlllib.urlencode(args)).
content)
translation_args = {
    'text': text,
    'to': destination_language,
    'from': origin_language
}

headers={'Authorization': 'Bearer '+oauth_junk['access_token']}
translation_url =
'http://api.microsofttranslator.com/V2/Ajax.svc/Translate?'
translation_result =
requests.get(translation_url+urlllib.urlencode(translation_args),
headers=headers)
translation=translation_result.text[2:-1]

speakOriginText('Translating ' + translation_args["text"])
speakDestinationText(translation)
```

Для запуску сценарію нам потрібно імпортувати кілька бібліотек python і медіа-плеєр:

```
sudo apt-get install python-pip mplayer
sudo pip install requests
```

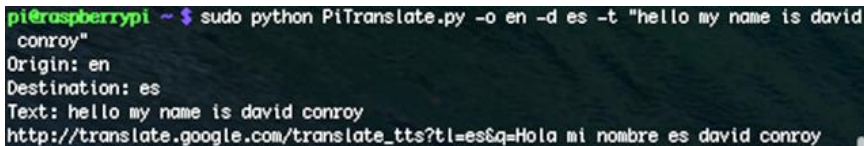
Останнє, що нам потрібно зробити, перш ніж ми зможемо запустити сценарій, це зареєструватися для ключа Microsoft Azure Marketplace API. Щоб зробити це, просто відвідайте marketplace<sup>171</sup>, зареєструйте додаток, а потім введіть ідентифікатор клієнта і секретний код доступу в сценарій вище.

Тепер ми можемо запустити сценарій (рис.9.22):

```
sudo python PiTranslate.py -o en -d es -t "hello my name is david conroy"
```

Сценарій має 3 необхідні входи:

- -o мова оригіналу
- -d вихідна мова
- -t "текст для перекладу"



```
pi@raspberrypi ~$ sudo python PiTranslate.py -o en -d es -t "hello my name is david conroy"
Origin: en
Destination: es
Text: hello my name is david conroy
http://translate.google.com/translate_tts?tl=es&q=Hola mi nombre es david conroy
```

Рис.9.22 Запуск сценарію

Наведена вище команда починає з англійської мови і перекладає на іспанську. Ви можете поекспериментувати, досліджуючи, як швидко можна змінювати мови, якими перекладаєте, і як вихідний голос змінюється відповідно з вихідною мовою.

### Збираємо все разом

Насправді дуже легко об'єднати два сценарії, які ми створили в цьому розділі. І це займе всього один рядок коду, який буде доданий до нижньої частини сценарію `stt.sh`, створений нами раніше (за умови, що `PiTranslate.py` і `stt.sh` знаходяться в одному каталозі):

```
sudo nano stt.sh
python PiTranslate.py -o en -d es -t "$value"
```

<sup>171</sup>

<https://login.live.com/login.srf?wa=wsignin1.0&wtrealm=https%3a%2f%2faccesscontrol.windows.net%2f&wreply=https%3a%2f%2fdatamarketmarketplace.accesscontrol.windows.net>

Для тих з вас, хто перестрибнув у цьому уроці, ось знову весь сценарій з доданням рядком:

```
echo "Recording your Speech (Ctrl+C to Transcribe)"
arecord -D plughw:0,0 -f cd -t wav -d 0 -q -r 16000 | flac - -s
-f --best --sample-rate 16000 -o daveconroy.flac;

echo "Converting Speech to Text..."
wget -q -U "Mozilla/5.0" --post-file daveconroy.flac --header
"Content-Type: audio/x-flac; rate=16000" -O -
"http://www.google.com/speech-api/v1/recognize?lang=en-
us&client=chromium" | cut -d\" -f12 > stt.txt

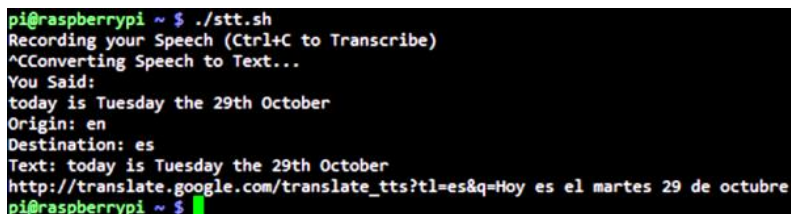
echo "You Said:"
value=`cat stt.txt`
echo "$value"

#translate from English to Spanish and play over speakers
python PiTranslate.py -o en -d es -t "$value"
```

Тепер знову запустить сценарій перетворення мови в текст, і він перекладе з англійської на іспанську за замовчуванням:

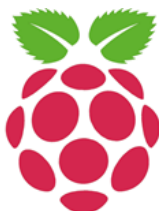
```
./stt.sh
```

Змініть оригінальну і вихідну мови в останньому рядку, відповідно до побажань, і сценарій `PiTranslate.py` зробить все інше! Існують буквально 1000 мовних пар, які тут підтримуються. Ось скріншот (рис.9.23):



```
pi@raspberrypi ~ $ ./stt.sh
Recording your Speech (Ctrl+C to Transcribe)
^CConverting Speech to Text...
You Said:
today is Tuesday the 29th October
Origin: en
Destination: es
Text: today is Tuesday the 29th October
http://translate.google.com/translate_tts?tl=es&q=Hoy es el martes 29 de octubre
pi@raspberrypi ~ $
```

Рис.9.23 Приклад роботи сценарію



Тому, хто хоче розширити коло своїх проєктів, рекомендуємо встановити XBMC Media Center — безкоштовний багатоплатформовий медіаплеєр. Графічний інтерфейс програми дозволяє легко керувати відеофайлами, фотографіями та музикою, що знаходяться на комп'ютері, оптичному диску, в Інтернеті або в локальній мережі.

## 10. РОЗШИРЕННЯ ТА АЛЬТЕРНАТИВИ

### Сенсорний екран Adafruit PiTFT



Є багато проектів з Raspberry Pi, які виграють від сенсорного екрану, але взагалі вони не настільки економічно ефективні. Тепер же, з крихітним екраном Adafruit PiTFT (рис.10.1) це виглядає більш розумно<sup>172</sup>.

PiTFT Mini Kit за ціною \$35 не тільки дешевий спосіб додати сенсорний екран до Raspberry Pi, але це, ймовірно, і самий простий спосіб. З набором ви отримуєте все необхідне для підключення 320x240, 16-бітного кольорового дисплею до вашого Pi. Також в комплект входить чудове керівництво<sup>173</sup> для встановлення програмного забезпечення, яке реально вам допоможе. Ні в якому разі, це не супер чудовий дисплей, але вказана ціна робить додавання крихітного невеликого сенсорного екрану до вашого Pi супер простим.

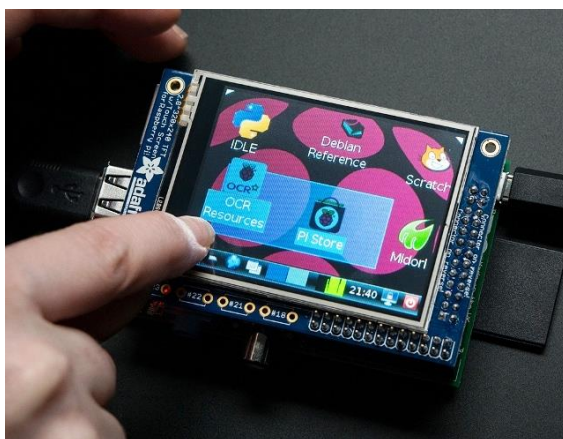
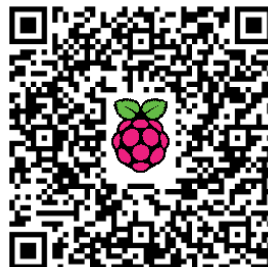


Рис.10.1 Крихітний екран Adafruit PiTFT

Для того, щоб додати підтримку 2.8" TFT і сенсорного екрану (рис.10.2), треба встановити нове ядро Linux. На щастя, автори створили пакет з ядром, який можете встановити просто поверх існуючої ОС Raspbian (або на основі Raspbian) замість встановлення абсолютно нового образу. Це робить підтримку актуальності вашої установки простішою.

<sup>172</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1806--raspberry-pi-adafruit-pitft>

<sup>173</sup> <https://learn.adafruit.com/adafruit-pitft-28-inch-resistive-touchscreen-display-raspberry-pi/software-installation>



Рис.10.2 Зворотна сторона сенсорного екрану

Пакет не буде працювати з Arch або іншими збірками Linux. Оскільки Raspbian є офіційною ОС для Pi, то це єдиний Linux, який підтримуватимуть розробники! Бажаючі можуть перекомпілювати власне ядро за допомогою наданого патчу, але немає ніякого підручника, підтримки або планів відносно них.

Плата використовує високошвидкісний інтерфейс SPI на Pi і може використовувати міні-дисплей як консоль, вікно порту X, для відображення зображень чи відео і т.д. Найкраще її підключити прямо зверху.

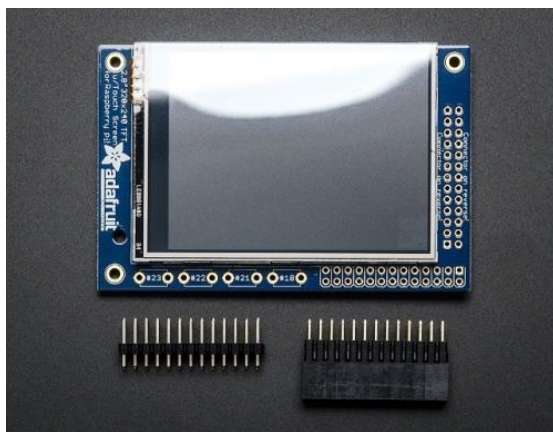


Рис.10.3 Доступні контактні виводи

Версії плати поставляються у вигляді міні-комплекту з 2x13 роз'ємом «мама» (для підключення планшета до Pi) і 2x13 роз'ємом «тато», який можна використати для підключення кабелю IDC або шини збоку. Фотографії також показують (рис.10.3) опціонально встановлені додаткові тонкі сенсорні кнопки. Сенсорні кнопки не включені, але

можете підібрати з 20-ти доступних<sup>174</sup>. Деякі основні рішення вимагають встановлення апаратних подовжувачів. Ви можете використати додатковий довгий подовжувач для Pi<sup>175</sup>, якщо хочете встановити його замість встановленого 2x13 подовжувача «мама».

Конструкція використовує виводи апаратного SPI (SCK, MOSI, MISO, CE0, CE1), а також GPIO #25 і #24. Всі інші GPIO не використовуються. Оскільки у розробників було зовсім небагато місця, то залишились 4 місця для додаткових тонких сенсорних перемикачів через дровові з'єднання з чотирма GPIOs (#23, #22, #21 і #18), які можете використати, якщо захочете зробити базовий інтерфейс користувача (рис.10.4). Наприклад, можете використати один з них, як кнопку вмикання/вимикання живлення.

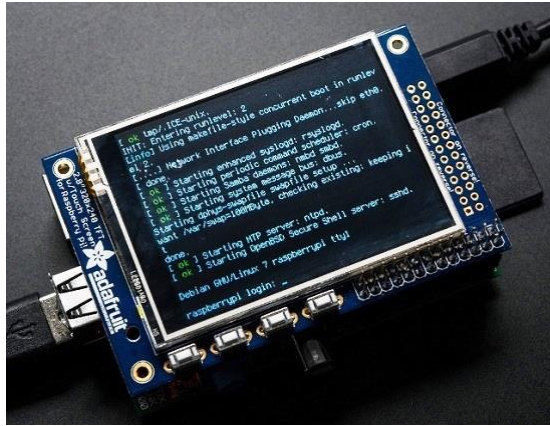


Рис.10.4 Встановлені додаткові сенсорні кнопки

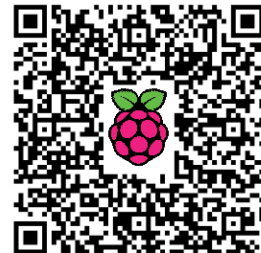
В серії згаданих підручників показано, як встановити програмне забезпечення, а також відкалібрувати сенсорний екран, запустити відео, показати зображення, наприклад, від вашої PiCam і багато іншого.

## Комп'ютерний модуль Raspberry Pi



З інженерної сторони справи розробники були дуже зайняті протягом минулого року і, не бажаючи відставати від команди з освіти, вже готові взяти на себе і розсекретити щось особливе, на цей раз орієнтоване на ділових і промислових споживачів.

Після старту платформи Raspberry Pi виріс і змужнів: програмне забезпечення на даний час



<sup>174</sup> <http://www.adafruit.com/products/1489>

<sup>175</sup> <http://www.adafruit.com/products/1112>

повнофункціональне і стабільне, і досі постійно вдосконалюється завдяки триваючій напруженій роботі героїчної спільноти добровольців; а також цільовим ін'єкціям фінансування для вирішення деяких конкретних питань.

Pi і Broadcom BCM2835 SoC, за своєю суттю, також неухильно стають все більш відкритими.



Рис.10.5 Комп'ютерний модуль Raspberry Pi

Розробники люблять чути про те, що користувачі роблять з їх Raspberry Pi, і постійно вражені низкою проектів, а також винахідливістю і творчістю громади. Також відомо, що є дуже значне число користувачів, які вбудовують Raspberry Pi в системи і навіть комерційні продукти. Розробники вважають, що повинен бути кращий спосіб, щоб дозволити людям отримати в свої руки цю чудову технологію в більш гнучкому форм-факторі, але все ж отримати речі за розумною ціною.

Автори хочуть звільнити основні технології на Raspberry Pi, щоб просунути його вперед і зробити невід'ємною частиною нових і цікавих продуктів і пристроїв. Тому оголосили про майбутній комп'ютерний модуль Raspberry Pi<sup>176</sup> (рис.10.5).

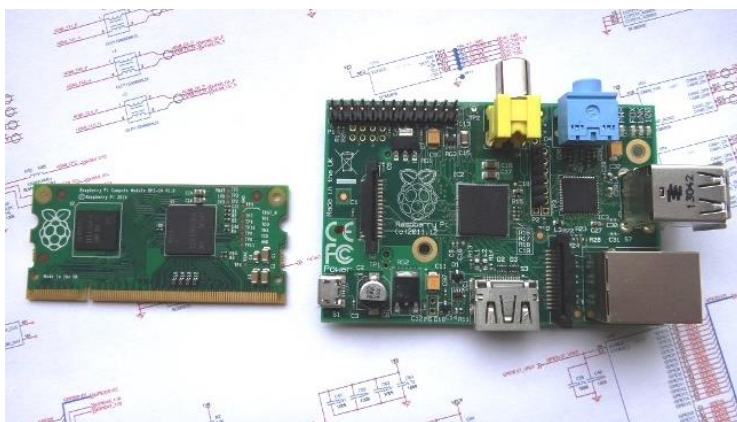


Рис.10.6 Комп'ютерний модуль зліва, Raspberry Pi праворуч

<sup>176</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1818-raspberry-pi-compute-module-new-product>

Комп'ютерний модуль містить нутрощі Raspberry Pi (процесор BCM2835 і 512 Мбайт оперативної пам'яті), а також пристрій 4 Гб eMMC Flash (який є еквівалентом карти SD в Pi). Це все інтегровано на невеликій 67.6x30 мм платі (рис.10.6), яка вписується в стандартний DDR2 SODIMM роз'єм (той же тип роз'єму, який використовується для пам'яті ноутбука\*). Флеш-пам'ять підключається безпосередньо до процесора на платі, але інтерфейси процесора залишилися доступні користувачеві через контакти роз'єму. Ви отримуєте повну гнучкість BCM2835 SoC (а це означає, що набагато більше GPIO і інтерфейсів доступні в порівнянні з Raspberry Pi) і налаштування модуля в користувальницькій системі має бути відносно простим, тому що всі «каверзні біти» розміщені на самому модулі.

Так як ви бачите, Raspberry Pi скоротився, щоб поміститися на SODIMM з вбудованою пам'яттю, чиї роз'єми можна налаштувати для власних потреб.

Комп'ютерний модуль, в першу чергу, призначений тим, хто збирається створити свій власний PCB. Тим не менш, автори також запускають те, що називається платою IO комп'ютерного модуля (рис.10.7) (Compute Module IO Board), щоб допомогти конструкторам стартувати.



Рис.10.7 Порожня плата IO ліворуч, вставлений комп'ютерний модуль на платі праворуч.

Плата IO (рис.10.8) комп'ютерного модуля є простою, з відкритим вихідним кодом, до якої можете підключити комп'ютерний модуль. Вона забезпечує необхідне живлення модуля і дає можливість програмувати флеш-пам'ять модуля, надає дещо більш дружелюбний доступ до

процесорних інтерфейсів (роз'єми виводів та флекс-роз'єми такі самі, як і в Pi) і забезпечує необхідні HDMI і USB-роз'єми, щоб у вас була система повністю, яка може завантажувати Raspbian (або іншу ОС за вашим вибором).



Рис.10.8 Плата IO.

Плата забезпечує як початковий шаблон для тих, хто хоче розробляти з комп'ютерним модулем, так і швидкий спосіб, щоб почати експериментувати з залізом та створювати і тестувати систему перед тим, як замовляти виготовлення загальної плати.

Спочатку комп'ютерний модуль і плата IO будуть доступні в продажу разом, як Raspberry Pi Compute Module Development Kit.

Ці комплекти будуть доступні на RS і element14 через деякий час. Незабаром після цього комп'ютерний модуль буде доступний в продажу окремо, вартістю за штуку близько \$30 в партіях від 100, але також зможете придбати їх окремо, хоча ціна буде трохи вищою. Raspberry Pi Foundation є благодійною організацією, і як з усім, що роблять автори тут, весь прибуток направляється відразу назад для навчання дітей комп'ютерам.

Автор впевнений, що люди будуть зацікавлені запустити свій процес проектування; зараз випущені тільки схеми для Compute Module<sup>177</sup> і IO Board<sup>178</sup>, але розробники обіцяють додати набагато більше документації в найближчі дні і тижні.

**Увага!** Не підключайте Compute Module до ноутбука – назначені виводи навіть віддалено не такі ж самі!

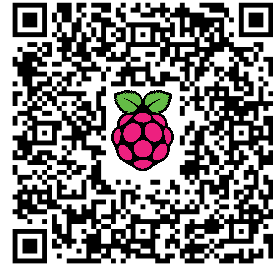
<sup>177</sup> [http://www.raspberrypi.org/wp-content/uploads/2014/04/RPI-CM-V1\\_1-SCHEMATIC.pdf](http://www.raspberrypi.org/wp-content/uploads/2014/04/RPI-CM-V1_1-SCHEMATIC.pdf)

<sup>178</sup> [http://www.raspberrypi.org/wp-content/uploads/2014/04/RPI-CMIO-V1\\_2-SCHEMATIC.pdf](http://www.raspberrypi.org/wp-content/uploads/2014/04/RPI-CMIO-V1_2-SCHEMATIC.pdf)

## Розширення Pi Co-op Arduino



Pi Co-op<sup>179</sup> - плата розширення для Raspberry Pi, яка передає потужність мікроконтролера Arduino нашому улюбленому одноплатному комп'ютеру.



### Що таке Pi Co-op?

Pi Co-op об'єднує в собі гнучкість Pi з дивовижними можливостями мікроконтролера Arduino. Для синхронізації критично важливих додатків Arduino виступає королем. Додаючи його до Pi ви отримаєте найкращий контроль аналогових датчиків, які вимагають аналого-цифрового перетворення і пристроїв, які мають чутливі до часу шини даних. Це включає в себе LCD-екрани і NeoPixels.



Рис.10.9 Raspberry Pi з розширенням Pi Co-op Arduino

Параметри:

- ATMEGA328P мікроконтролер
- 10bit 8-канальний АЦП
- Може бути запрограмований безпосередньо з Pi за допомогою безкоштовного Arduino IDE
- Швидкість UART до 115200 бод
- ATMEGA328P виводи у вбудованому роз'ємі типу «мама» для легкого створення прототипів

<sup>179</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1842-expansion-of-pi-co-op-arduino-raspberry-pi>

- Pi може отримувати живлення через плату
- Стандартні затискачі з кроком 2.54 мм на всіх роз'ємах

Повна специфікація знаходиться на офіційній сторінці продукту Pi Co-op<sup>180</sup>, включаючи посилання на чудове керівництво користувача<sup>181</sup>.

### Крок 1 - Початкове налаштування

Налаштування Pi Co-op займає всього кілька простих кроків. Запуск двох команд нижче в командному рядку. Ваш Pi повинен бути підключеним до Інтернету:

```
sudo apt-get update  
sudo apt-get install arduino git
```

Це дозволить встановити Arduino IDE (що дасть можливість вам відправляти "скетчі" Arduino на пристрій) та систему управління програмним забезпеченням Git.

Тепер виконайте наступну команду для завантаження ресурсів Pi Co-op:

```
git clone https://bitbucket.org/DawnRobotics/pi_co-op.git
```

Для того, щоб спілкуватися з пристроєм, послідовний порт повинен бути включений на GPIO виводах Pi. В установчому файлі можна зробити ці зміни конфігурації для дозволу послідовного інтерфейсу. Це дозволить Pi спілкуватися з Pi Co-op:

```
cd pi_co-op  
sudo python setup_pi_co-op.py install
```

Закінчимо конфігурацію шляхом перезавантаження Raspberry Pi:

```
sudo reboot
```

### Крок 2 - Програмування Arduino

Коли закінчиться перезавантаження Pi вводимо:

```
startx
```

щоб увійти в робочий стіл LXDE.

---

<sup>180</sup> <http://www.dawnrobotics.co.uk/pi-co-op-raspberry-pi-arduino-add-on-board/>

<sup>181</sup> [http://www.dawnrobotics.co.uk/content/pi\\_co-op.pdf](http://www.dawnrobotics.co.uk/content/pi_co-op.pdf)

Тепер можете запустити Arduino IDE. Це може бути зроблено за допомогою значка в розділі "Programming" ("Програмування") в

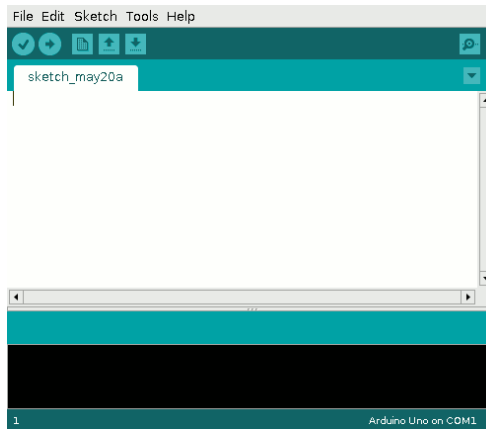


Рис.10.10 Стартове вікно Arduino IDE

стартовому меню робочого столу або набравши "Arduino" в вікні LXTerminal командного рядка. Коли запуститься IDE, екран повинен виглядати наступним чином (рис.10.10):

Тут багато прикладів сценаріїв всередині інтегрованого середовища, відомих як «скетчі» («ескізи»). Приклад "hello world" у громадськості Arduino являє собою скетч з назвою "Blink". Завантажте приклад "Blink" використовуючи послідовність меню File > Examples > 01 Basics > Blink (рис.10.11):

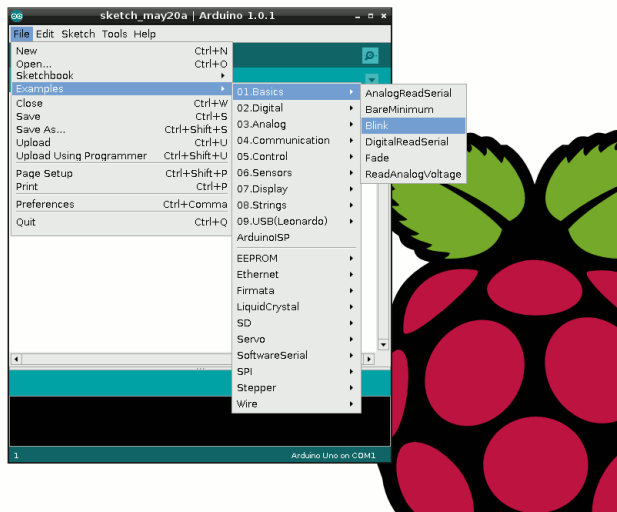


Рис.10.11 Запуск скетчу "Blink"

Скетч "Blink" завантажиться і виглядатиме приблизно так (рис.10.12):

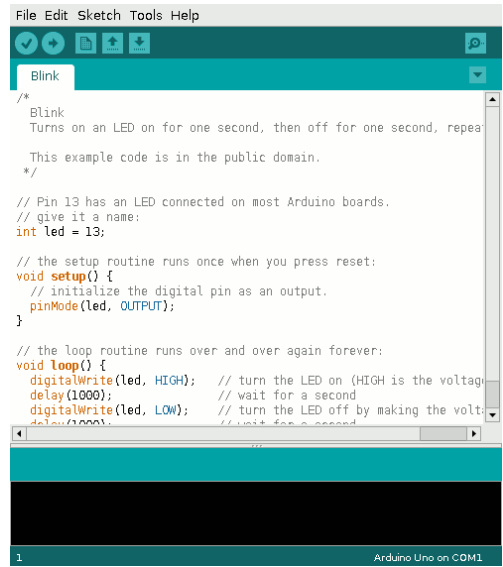


Рис.10.12 Програмний код скетча "Blink"

Можна швидко перевірити, що тип плати встановлений на "Arduino Uno", використовуючи послідовність меню Tools > Board (рис.10.13):

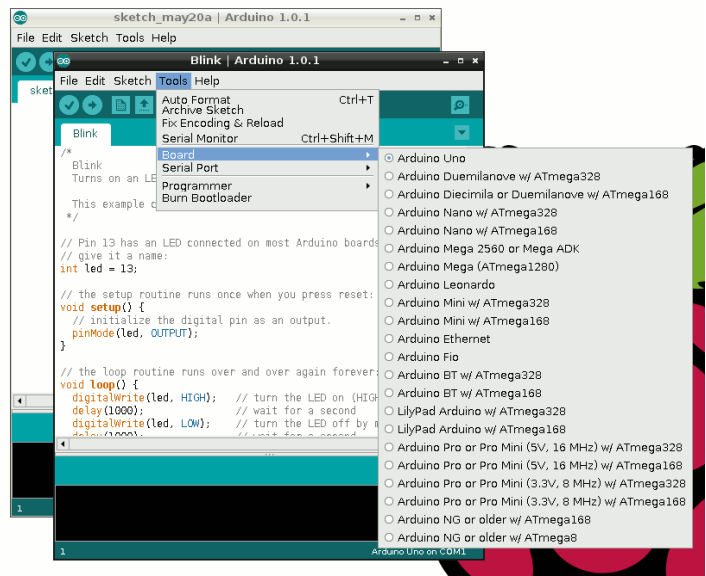


Рис.10.13 Швидка перевірка типу встановленої плати

Тепер перевіримо, що встановлений послідовний порт на ttyS0 за допомогою послідовності меню Tools > Serial Port (рис.10.14):

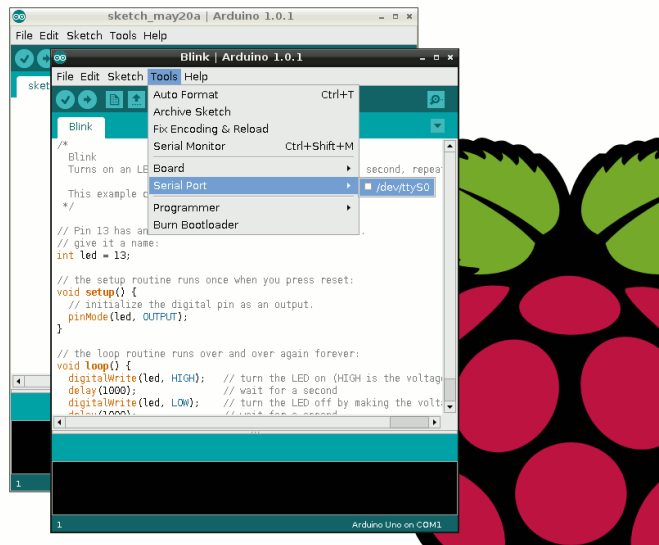


Рис.10.14 Перевірка встановленого послідовного порта

На цьому конфігурація виконана. Натисніть на значок стрілки, щоб компілювати скетч і завантажити його на Pi Co-op (рис.10.15):

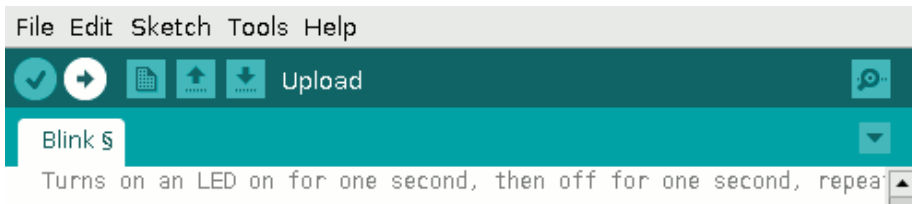


Рис.10.15 Компіляція і запуск скетча

Якщо все пройшло правильно, то світлодіод на платі Pi Co-op повинен мерехтати.

### Крок 3 - Отримайте Hacking!

Якщо поглянете на приклад Blink, то побачите два рядки із зазначенням:

```
delay(1000);
```

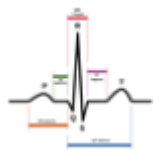
Це затримка (в мілісекундах) між кожним перемиканням світлодіода. Змініть значення на 500 (півсекунди) і натисніть на значок

стрілки, щоб знову скомпілювати і завантажити. Світлодіод повинен блимати швидше.

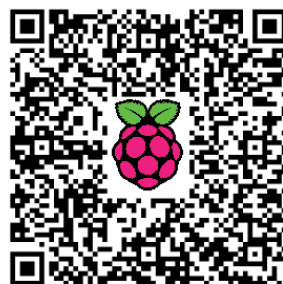
Збільшення числа змусить світлодіод блимати повільніше.

Pi Co-op діє як Arduino Uno, тому для багатьох проектів його можна підключити до обладнання і знайти існуючі скетчі Uno, щоб стартувати.

### Платформа датчиків e-Health для Arduino і Raspberry Pi



Cooking Hacks запустили нову версію першого біометричного шілда для iArduino і Raspberry Pi: платформа датчиків e-Health V2.0<sup>182</sup> (біометричні/медичні застосування).



Платформа датчиків e-Health V2.0 (рис.10.16 і 10.17) дозволяє користувачам Arduino і Raspberry Pi виконувати біометричні і медичні вимірювання, в яких необхідно моніторити тіло за допомогою 10 різних датчиків: пульсу, кисню в крові (SPO2), повітряного потоку (дихання), температури тіла, електрокардіограми (ECG), глюкометра, шкірно-гальванічної реакції (GSR - пітливість), артеріального тиску (тонометр), положення пацієнта (акселерометр) і датчика м'язів/електроміографії (EMG).

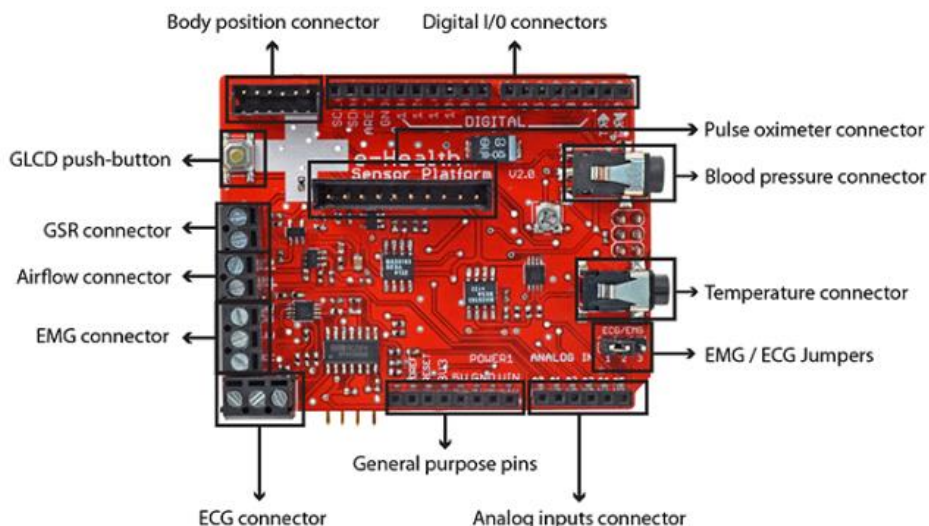


Рис.10.16 Платформа датчиків (вигляд зверху)

<sup>182</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1805-e-health-sensor-platform-for-arduino-and-raspberry-pi->

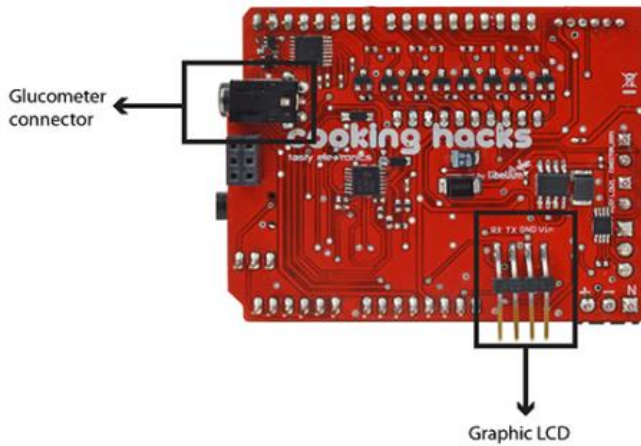


Рис.10.17 Платформа датчиків (вигляд знизу)

Ця інформація може використовуватися для моніторингу стану пацієнта в реальному часі або для отримання конфіденційних даних, з тим, щоб згодом проаналізувати для медичної діагностики (рис.10.18). Зібрана біометрична інформація може бути передана бездротовою мережею з використанням будь-якого з 6-ти доступних варіантів передачі: Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 і ZigBee, залежно від застосування.



Рис.10.18 Платформа з приєднаними датчиками

Якщо необхідне діагностичне зображення в режимі реального часу, то можна приєднати камеру до модуля 3G для відправки фотографій і відео пацієнта в медичний центр діагностики.

Дані можуть бути відправлені «на хмару» для постійного зберігання або візуалізовані в реальному часі, відправивши їх безпосередньо на ноутбук чи смартфон. Розроблені додатки для iPhone і Android, щоб легко побачити інформацію про пацієнта (рис.10.19).

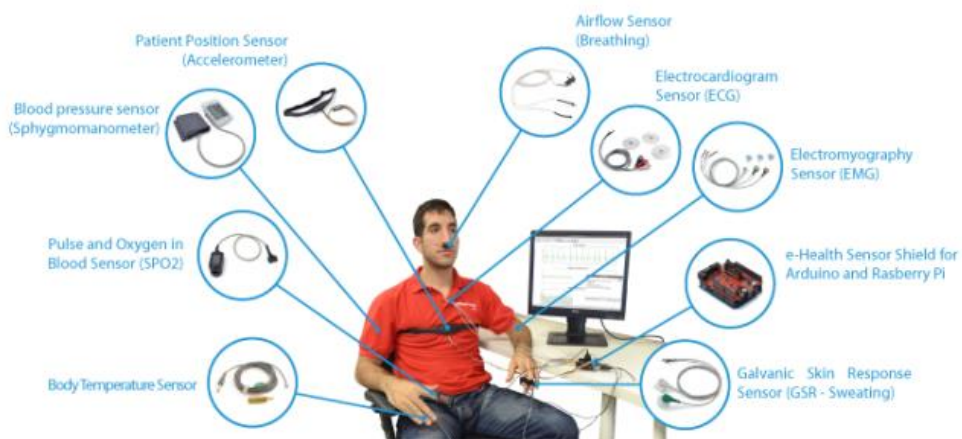


Рис.10.19 Схема підключення датчиків до клієнта

Це відкриває нову еру виробів медичного призначення з відкритим вихідним кодом. RPi надає нові додатки e-Health та продукти концепції швидкої діагностики з необхідними інструментами. Проте одним з ключових моментів в таких додатках є конфіденційність і кілька рівнів безпеки, надані з платформою.

Рівні ліній зв'язку використовують WPA2 для Wi-Fi і AES 128 для ZigBee і 802.14.5. На прикладному рівні використовується захищений протокол (HTTPS) для забезпечення захищеного тунелю точка-точка між веб-сервером і кожним вузлом датчика. Банки використовують цей тип безпечних протоколів зв'язку для своїх транзакцій.

Для Arduino (рис.10.20-21):

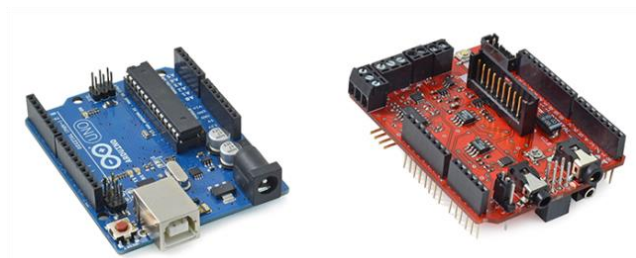


Рис.10.20 Arduino і платформа датчиків

Платформу датчиків e-Health можна отримати з Cooking Hacks. У порівнянні з дорогими і пропрієтарними рішеннями медичного ринку, Cooking Hacks забезпечує порівняно дешеву і відкриту альтернативу.

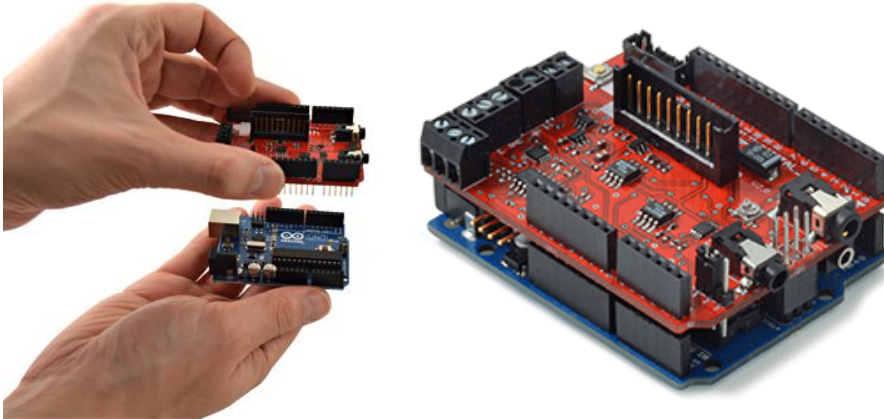


Рис.10.21 Приєднання платформи датчиків до Arduino

Cooking Hacks також забезпечує міст під'єднання RPi до шілдів Arduino, який включає можливість підключення аналогових і цифрових датчиків для обох плат. Це дозволяє спільне використання потужності і здатності RPi з виводами Arduino. Крім того, також надається бібліотека arduPi, що дозволяє використовувати RPi з тим же кодом, який використовується для Arduino. Бібліотека arduPi дозволяє бути взаємозамінними бездротовим модулям, датчикам, шілдам і електронним модулям або приводам як для RPi, так і Arduino.

Для Raspberry Pi (рис.9.22-24):



Рис.10.22 Raspberry Pi, перехідник і платформа датчиків

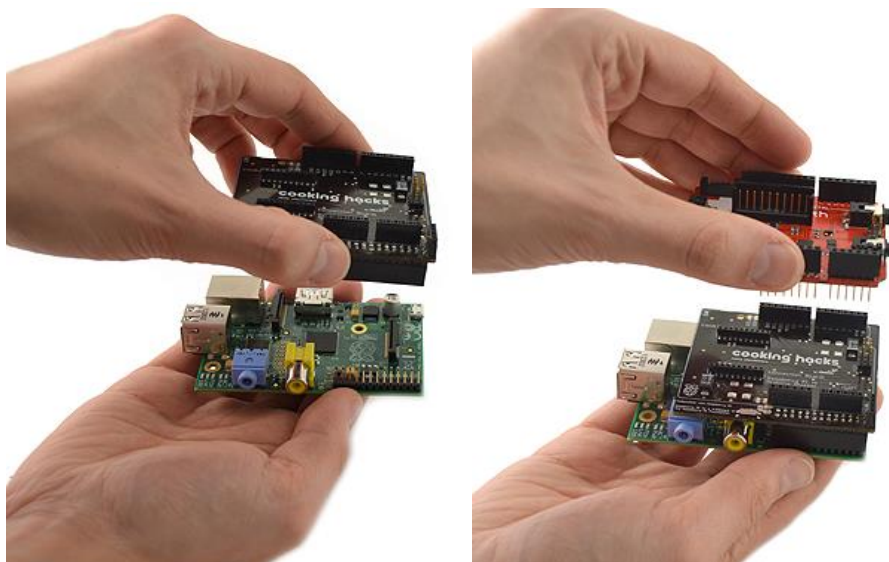


Рис.10.23 Послідовність під'єднання окремих модулів

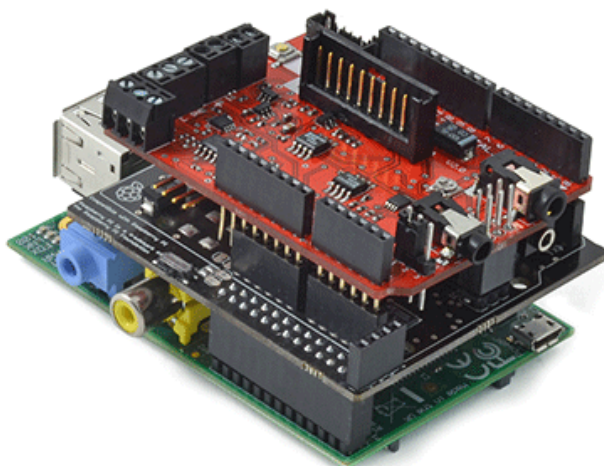


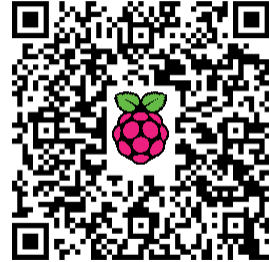
Рис.10.24 Платформа датчиків, змонтована на Raspberry Pi

*Примітка:* Платформа датчиків e-Health ще не має медичної довідки. Таким чином, вона не повинна використовуватися для моніторингу пацієнтів, які мають критичний стан і які повинні перебувати під наглядом медичних методів, що є більш точними, або тих, чиї умови вимагають моніторинг для прихованої професійної діагностики

## Плата розширення GSM/GPRS і GPS



Плата розширення<sup>183</sup> для Raspberry Pi може ефективно контролювати модулі GSM GPRS SIM900<sup>184</sup> і SIM908<sup>185</sup> (з GPS) і, таким чином, значно розширити функціональність Raspberry Pi для мобільних додатків<sup>186</sup> (наприклад: дистанційного керування), а з використанням модуля SIM908, навіть отримати функціональність GPS (рис.10.25).



Такого типу розширення призначені для штабелювання один на одному і тому в змозі керувати кількома завданнями одночасно: ви зможете включати різні «послуги» в одному інтегрованому рішенні. Це зручно у випадку створення системи домашньої автоматизації, оснащеної датчиками і виконавчими пристроями, які використовують РК-дисплей для активації і деактивації, коли необхідно додати віддалений інтерфейс через GSM або TCP/IP мережу.



Рис.10.25 Використання плати з SIM 908 на Raspberry Pi

<sup>183</sup> [https://store.open-electronics.org/index.php?\\_route\\_=Linux\\_Embedded/GSM\\_Expansion\\_Raspberry](https://store.open-electronics.org/index.php?_route_=Linux_Embedded/GSM_Expansion_Raspberry)

<sup>184</sup> [https://store.open-electronics.org/index.php?\\_route\\_=Breakout/GSM%20Breakout/Small%20Breakout%20for%20SIM900%20GSM%20Module](https://store.open-electronics.org/index.php?_route_=Breakout/GSM%20Breakout/Small%20Breakout%20for%20SIM900%20GSM%20Module)

<sup>185</sup> [https://store.open-electronics.org/index.php?\\_route\\_=Breakout/GSM%20Breakout/Small\\_Breakout\\_SIM908](https://store.open-electronics.org/index.php?_route_=Breakout/GSM%20Breakout/Small_Breakout_SIM908)

<sup>186</sup> <http://isearch.kiev.ua/uk/searchpractice/science/1844--gsmgprs-gps-raspberry-pi>

Можна також додати послуги безпеки для літніх людей, які активуються як локально, так і віддалено, використовуючи можливості зв'язатися з заздалегідь визначеним номером за допомогою голосу або SMS. Навіть можемо розробити автоматизовані двері, що відкриваються, використовуючи функцію розпізнавання обличчя, або відстеження руху: всі ці програми зможуть відправити нам зображення в разі підозрілої діяльності. Єдиним обмеженням є ваша уява і здатність розробляти і програмувати додатки. Складність цих застосувань перевищує можливості мікроконтролерів, хоча останні можуть бути використані для управління датчиками і периферійними пристроями.

В розширенні GSM/GPRS і GPS використовуються виводи GPIO для підключення до послідовного порту і два додаткових виводи вводу/виводу для живлення (вкл/викл) та вивід для скидання модуля SIM9XX (рис.10.26).

## Схема підключення

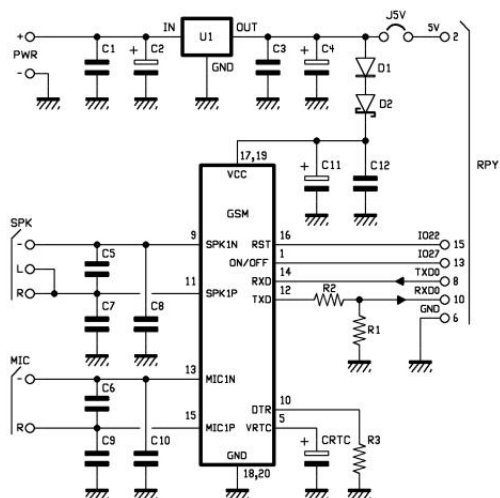


Рис.10.26 Схема підключень до плати

Схема підключення розширення GSM забезпечує підтримку як модуля SIM900, так і SIM908, забезпечує автономну подачу живлення для модуля GSM, експорт мікрофонного входу і аудіовиходу на два виділені гнізда і правильне з'єднання контактів роз'єму GPIO до модулів GSM.

На схемі, U1 - стабілізатор живлення, який-через стабілітрони D1 і D2 подає напругу живлення на відповідний вивід модуля SIM9XX.

Також можна подавати живлення з Raspberry Pi через роз'єм J5V, а до гнізда PWR можна підключити зовнішнє джерело живлення, яке забезпечує постійний струм напругою 7-12

Вивід ON/OFF (вкл/викл SIM9XX під'єднується до контакту 13 (GPIO27) GPIO Raspberry Pi для програмного включення і виключення модуля (рис.10.27).)

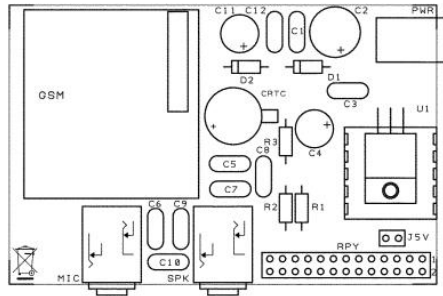


Рис.10.27 Розміщення елементів на платі розширення

Вивід скидання RST модуля SIM під'єднується до контакту 15(GPIO22) Raspberry Pi, також щоб дозволити контролювати його програмно. Послідовні виходи приймання/передавання використовують виводи 8 (TXD) і 10 (RXD) Raspberry Pi. Контакт для живлення буфера годинника реального часу (VRTC) підключений до конденсатора CRTC. Нарешті, лінійний вхід мікрофона і аудіовихід підключаються до відповідних роз'ємів MIC і SPK (рис.10.28).



Рис.10.28 Плата розширення без модуля SIM9XX

### Практичне використання плати розширення

Перш за все, ми повинні змонтувати один з сумісних модулів GSM на платі розширення, а потім зібрати плату на Raspberry Pi (рис.10.29), переконавшись, що нижня частина плати не торкається до роз'ємів USB або Ethernet. В іншому разі, просто захистіть роз'єми ізоляційною стрічкою. Підключіть периферійні пристрої і живлення Raspberry Pi в звичайному порядку.

За допомогою GNU/Linux, ми можемо створити програми, які виконують безліч функцій, спільно використовуючи такі ресурси, як GSM-модуль. Наприклад, ми могли б написати додаток, який відстежує такі зовнішні події, як вхідний голосовий виклик або SMS-повідомлення,

а також в змозі реагувати на такі внутрішні події, як активація введення/виводу даних і, відповідно, виконувати голосові виклики, відправляти масові або персональні SMS, відправляти файли (наприклад, зображення), або електронні листи і т.д. ...



Рис.10.29 Плата розширення з модулем SIM9XX на Raspberry Pi

У той же час, ви можете додати гео довідкову інформацію завдяки доступності GPS на платі. Для виконання всіх цих функцій необхідно розробити модульну архітектуру, яка зможе спільно використовувати ресурси, не створюючи конфліктів.

Архітектура, що складається з сервера модулів, інтерфейсу та управління ресурсами та спеціалізованих модулів, які забезпечують описані функціональні можливості і які можуть викликатися та виконуватися за певних умов.

Це потребує послуг зв'язку між запитувачами (програмами) і серверними модулями, що можуть задовольнити їх відповідно з логікою, яка забезпечує пріоритети і уникнення конфліктів.

Для цього вам потрібно вникати аж до деталей функціональних вимог додатка, і це причина, чому, перш ніж заглиблюватись в розроблення клієнт/серверних додатків, ми пропонуємо вам режим, який дозволяє отримати досвід гнучкого управління.

Цей режим дозволяє використовувати програми, написані для Arduino, з невеликими змінами, так, щоб ознайомитися з платою у більш безпосередній формі. Цей режим використовує бібліотеку arduPi, розроблену Libelium Comunicaciones SL Distribuidas, доступною під відкритою ліцензією і розміщеною на CookingHacks.

### Надсилаємо SMS в стилі Arduino

Бібліотека arduPi може бути встановлена шляхом простого скачування архіву<sup>187</sup> і вилучення його вмісту в `/home`. Після цього, просто перейменуйте папку в `arduPi` і, для простоти, покладіть програми, які збираєтеся реалізувати, в ту ж папку.

<sup>187</sup> [http://www.cooking-hacks.com/skin/frontend/default/cooking/images/catalog/documentation/raspberry\\_arduino\\_shield/arduPi\\_1-5.tar.gz](http://www.cooking-hacks.com/skin/frontend/default/cooking/images/catalog/documentation/raspberry_arduino_shield/arduPi_1-5.tar.gz)

Перж, ніж скопіювати бібліотеку, відкрийте програму `arduPi.cpp` і, якщо відрізняється, замініть рядок 64 з посиланням на існуючий послідовний порт на рядок:

```
serialPort="/dev/ttyAMA0";
```

Тепер переходимо в папку ArduPi з командою: `cd /home/ArduPi` і компілюємо бібліотеку за допомогою команди:

```
g++ -c arduPi.cpp -o arduPi.o
```

А те, що стосується програм, то ми пропонуємо три модулі. Два з них дозволяють вмикати і вимикати модуль GSM і будуть дуже корисними в майбутньому. Третій дозволяє відправити текстове повідомлення на мобільний телефон. Для включення модуля ви повинні підняти рівень сигналу на виводі GPIO27 і тримати його високим протягом двох секунд, а потім повернути назад в низький рівень. Ця ж процедура повинна бути використана, щоб відключити його, якщо модуль включений. Для скидання просто підняти і тримати на виводі GPIO22 високий рівень півсекунди (потім повернути його назад в низький). Ці програми можна безкоштовно завантажити на сайті [www.elettronica.in.it](http://www.elettronica.in.it).

Передусім, щоб мати можливість використовувати послідовний порт, ми повинні внести зміни в процес завантаження і відключити послідовну консоль, бо інакше вона буде тримати послідовний порт зайнятим.

### Відключення послідовної консолі

Майже всі системи GNU/Linux при завантаженні включають функцію консолі, тобто можливість підключення терміналу з послідовної лінії - реальної чи віртуальної - для управління всією системою через даний інтерфейс зв'язку. Це нагадує ті часи, коли ми звикли підключатися до комп'ютерів завдяки модемам з використанням телефонної лінії. Щоб відключити консоль ми використовуємо WinSCP, переходимо в папку `boot` за допомогою команди: `cd /boot` відкриваємо подвійним клацанням миші файл `/boot/cmdline.txt` і видаляємо наступні параметри: `console=ttyAMA0,115200 kgdboc=ttyAMA0,115200` та зберігаємо файл.

### Приклади програм (включення модуля і відправка SMS)

Наведений нижче код має логіку для активації модуля. Процес активації модуля GSM включає наступні етапи:

- Управління: відправити на послідовний порт команду AT і переконатися, що модуль не завантажений
- У разі негативної відповіді (на передану команду AT ви повинні отримати рядок "OK"), як додаткову перевірку, виконати процедуру скидання з другою командою AT

- У разі подальшої негативної відповіді на «запалювання», виконується процедура запуску, яка полягає у забезпеченні на виводі GPIO27 високий рівень сигналу протягом двох секунд.

Мета попередніх перевірок пов'язана з тим, що включення і виключення працює з одним і тим же процесом, тобто, якщо запустите програму без перевірки, то при вже включеному модулі GSM, він вимкнеться.

```
/*
 * GSMIgnition
 */
//Під'єднуємо бібліотеку ArduPi
#include "arduPi.h"
int resetModulePin = 9;
int onModulePin = 8;
void switchModuleOn(){
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}
void resetModule(){
    digitalWrite(resetModulePin,HIGH);
    delay(500);
    digitalWrite(resetModulePin,LOW);
    delay(100);
}
int main (){
    Serial.begin(115200);
    delay(2000);
    pinMode(resetModulePin, OUTPUT);
    pinMode(onModulePin, OUTPUT);
    Serial.flush();
    printf ("zero\n");
    Serial.print("AT");
    delay(1000);
    if (Serial.available()==0)
    {
        printf ("uno\n");
        resetModule();
        delay(2000);
    }
    Serial.print("AT");
    delay(1000);
    if (Serial.available()==0)
    {
        printf ("due\n");
        switchModuleOn();
    }
    return (0);
}
```

Щоб скомпілювати програму використаємо команду:

```
g++ -lrt -lpthread GSMIgnition.cpp arduPi.o -o GSMIgnition
```

Щоб запустити програму, переходимо в папку ArduPi (cd /home/ArduPi) і вводимо:

```
./GSMIgnition
```

Ми можемо відстежувати дії з вмикання модуля GSM, спостерігаючи за зеленим світлодіодом: при включенні він світиться приблизно дві секунди, а потім вимикається і починає мигати з інтервалом близько півсекунди, як знак того, що модуль шукає мережі GSM. Коли мережа додається, то індикатор починає блимати з набагато меншою швидкістю: близько одного спалаху в секунду.

Нижче код програми для вимикання модуля, в якому реалізована дуже схожа з попереднім логіка:

- Управління, відправкою команди АТ на послідовний інтерфейс, що модуль ще не вимкнути,
- якщо ні, то як ще один тест виконується процедура скидання шляхом відправки другої команди АТ
- У разі позитивної відповіді, виконується процедура вимикання, яка полягає у забезпеченні на виводі GPIO27 високого рівня сигналу протягом двох секунд.

```
/*
 * GSMOff
 */
//Під'єднуємо бібліотеку ArduPi
#include "arduPi.h"
int resetModulePin = 9;
int onModulePin = 8;
void switchModuleOff() {
    digitalWrite(onModulePin,HIGH);
    delay(2000);
    digitalWrite(onModulePin,LOW);
}
void resetModule() {
    digitalWrite(resetModulePin,HIGH);
    delay(500);
    digitalWrite(resetModulePin,LOW);
    delay(100);
}
int main () {
    Serial.begin(115200);
    delay(2000);
    pinMode(resetModulePin, OUTPUT);
    pinMode(onModulePin, OUTPUT);
```

```
Serial.flush();
printf ("zero\n");
Serial.print("AT");
delay(1000);
if (Serial.available()==0)
{
    printf ("uno\n");
    resetModule();
    delay(2000);
}
Serial.print("AT");
delay(1000);
if (Serial.available()>0)
{
    printf ("due\n");
    switchModuleOff();           //
}
return (0);
}
```

Для компіляції програми GSMOff використовуємо команду:

```
g++ -lrt -lpthread GSMOff.cpp arduPi.o -o GSMOff
```

Щоб запустити програму, завжди після переходу в папку **ArduPi**, скористаємося командою:

```
./ GSMOff
```

Нарешті, в коді нижче, можете побачити програму, яка дозволяє відправити SMS.

```
/*
 * GSMSms
 */
//Під'єднуємо бібліотеку ArduPi
#include "arduPi.h"
int timesToSend = 1;           // номер SMS для
відправлення
int count = 0;
int ok = 0;
int numCar = 0;
char phone_number[]="3.....9"; // номер отримувача SMS
void setup(){
    Serial.begin(115200);       // швидкість передавання
    UART
    delay(2000);
    Serial.println("AT+CMGF=1"); // встановити режим SMS в
    text
```

```

    delay(100);
}
void loop(){
    while (count < timesToSend){
        delay(1500);
        Serial.print("AT+CMGS=\"");           // надіслати SMS(s)
        Serial.print(phone_number);
        Serial.println("\"");
        while(Serial.read() != '>');
        Serial.print("Se arriva funziona");    // зміст SMS
        delay(500);
        Serial.write(0x1A);                   // надсилання ++
        Serial.write(0x0D);
        Serial.write(0x0A);
        delay(5000);
        count++;
    }
}
int main (){
    setup();
    while(1){
        loop();
        if (count == timesToSend)
        {
            break;
        }
    }
    return (0);
}

```

- Змінна `phone_number[]` містить телефонний номер для виклику,
- `AT + CMGF = 1` задає режим `Text` для SMS,
- `AT + CMGS` = відправляє SMS-повідомлення, встановлене в наступній інструкції,
- І, нарешті, посилаємо символи `"++"` для завершення виклику.

Щоб скомпілювати програму, вводимо команду:

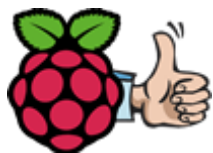
```
g++ -lrt -lpthread GSMSms.cpp arduPi.o -o GSMSms
```

і запускаємо програму, як звичайно: `./GSMSms`

Перш ніж запустити програму, вам треба запустити процедуру активації модуля GSM, щоб переконатися, що модуль під'єднаний і працює.

Коли ви підете далі в розробці програми управління, то вона повинна буде забезпечити управління правильною послідовністю дзвінків, перевіряти їх та керувати ненормальними умовами...

### Альтернативні мікрокомп'ютери



Тепер вже ніхто не ставить під сумнів успіх міні-комп'ютера Raspberry Pi. Майже одразу після виходу цього міні-ПК почали моделювати нові міні-ЕОМ, які також отримали чималу популярність<sup>188</sup>.

Подібна ситуація була з нетбуками кільका років тому, яких зараз велика кількість. І, що цікаво у випадку нетбуків, ідея маленького (і, головне, доступного) пристрою для використання з веб- та офісними додатками якось забулась. Потім почали з'являтися нетбуки з ціною вищою деяких потужних ноутбуків. Вочевидь, що подібна доля чекає і міні-комп'ютери.

Плата Raspberry Pi дешева, але вже є моделі більш потужні (і набагато дорожчі). Одним з таких альтернативних мікрокомп'ютерів є Odroid-X2 (рис.10.30).

ODROID-X2 - основа флагманського смартфона Galaxy S III.

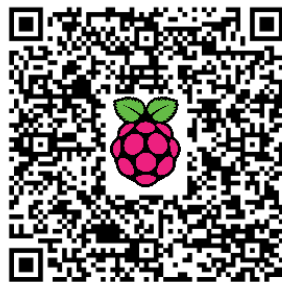


Рис.10.30 Мікрокомп'ютер Odroid-X2

Головною особливістю конструкції, мабуть, є відсутність корпусу. Візуально, цей продукт схожий на розглянутий Raspberry Pi, але десь на 100\$ дорожчий.

<sup>188</sup> <http://isearch.kiev.ua/uk/searchpractice/internetsecurity/1784-odroid-x2-a-new-alternative-replacement-raspberry-pi>

Для можливості розв'язувати всі оперативні задачі розробник надав 4-х ядерний процесор Exynos 4412 - одного із продуктів компанії Samsung, архітектури Cortex-A9, з частотою ядра 1.7 ГГц. Графічна система досить розповсюджена - Mali 400 з оперативною пам'яттю 2 Гб. Також є 6 USB-роз'ємів, вихід Ethernet, роз'єми для навушників і мікрофона, а також слот для карт пам'яті. До речі, є варіант з модулями бездротового зв'язку, ЖК-дисплея, HDMI портом, камерою.

Пристрій дещо більший за Raspberry Pi (90x94 мм) (рис.10.32), але значно потужніший. За словами розробників, плата може працювати під ОС Android 4.x, Ubuntu та іншими ОС Linux. Можливо, цей пристрій знайде споживача - потужні міні-ПК також мають попит.

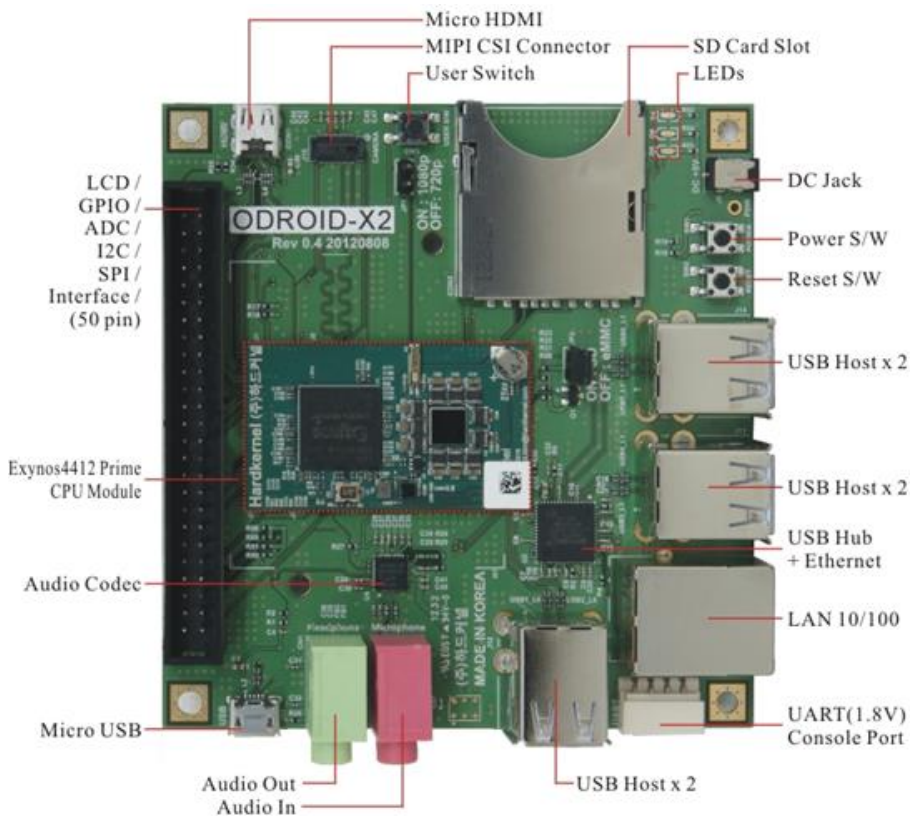


Рис.10.31 Призначення інтерфейсів Odroid-X2

## Специфікації платформи

Процесор: Samsung Exynos 4412 Cortex-A9 Quad Core 1.7 ГГц

Пам'ять: 2048Мбит(2Гбіт) LP-DDR2 880Mega

3D акселератор: Mali-400 Quad Core 440 МГц

*Відео:* 1080p через HDMI кабель (H.264+AAC, формат MP4)  
*Відео вихід:* мікро HDMI роз'єм/RGB-24bit LCD порт  
*Аудіо:* стандартні 3.5 мм навушники і мікрофон; HDMI; SPDIF  
*LAN:* 10/100 Мбіт/сек Ethernet з RJ-45 Jack (підтримка Auto-MDIX)  
*USB 2.0 Host:* Високошвидкісний стандарт типа А х 6 портів  
*USB 2.0 Device:* ADB/Mass пристрій (мікро USB)  
*UART:* система моніторингу консолі для розробки (1.8В інтерфейс)  
*IO PORT:* 50 виводів I/O для LCD/I2C/UART/SPI/ADC/GPIO інтерфейсів  
*Відображення (опція):* HDMI монітор/LCD панель з RGB або з LVDS інтерфейсом  
*Слоти:* повнорозмірний SDHC слот; eMMC модуль  
*Камера:* MIPI-CAM конектор (MIPI-CSI 2 лінії)  
*Живлення:* 5V 2A  
*Програмне забезпечення:* u-boot 2010.12, Kernel 3.0.15, Android4.0.x(ICS). Базовий код може бути замінений за бажанням користувача.  
*Габарити:* 90x94 мм

### Блок-діаграма

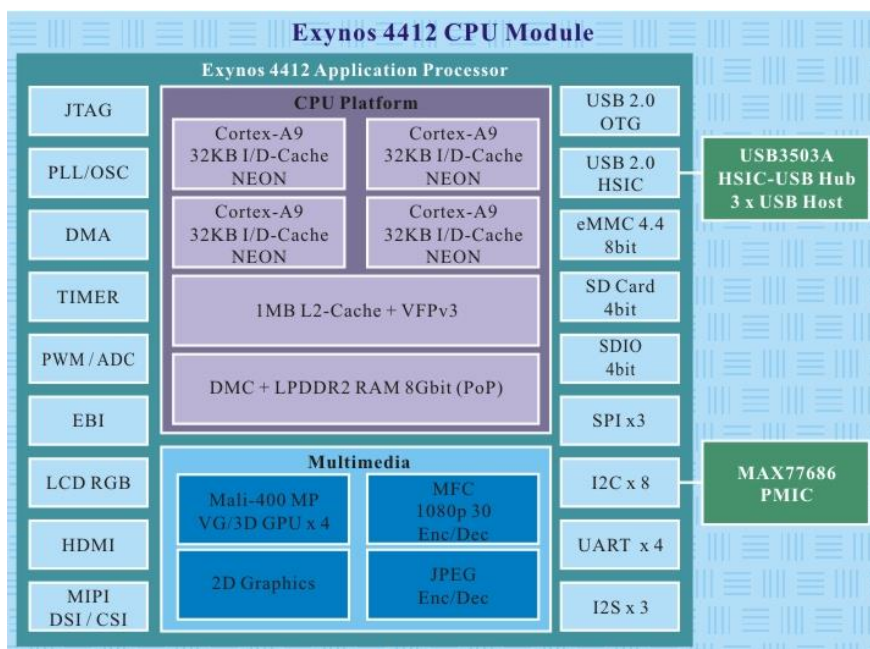


Рис.10.32 Блок-діаграма Odroid-X2

### Встановлення ОС ArchLinux на ODROID-X2

- Вставити SD-карту на 8 Гб або більше в ПК і перевірити пристрій, який ідентифікований (`dmesg | tail`)
  - Переконавшись, що він не змонтований (`umount /dev/sdX*` або `umount /dev/mmcblk*`)
  - Завантажити SD-образ ODROID-X<sup>189</sup> і розпакувати його за допомогою ArchLinuxARM-odroid-x-latest.img.gz
  - Записати образ на свою SD-карту за допомогою `dd if=ArchLinuxARM-odroid-x-latest.img of=/dev/[sdX або mmcblkX]`
  - Користувачі Windows можуть скористатися Image Writer для Windows<sup>190</sup>
  - Після закінчення запису вийняти SD-карту з ПК, вставити в ODROID і потримати клавішу вмикання живлення натиснутою 1-2 сек.
  - Зайти через SSH або як "root" з паролем за замовчуванням "root"
- Доступні також інші версії ОС Linux для скачування:
- Debian-Wheezy<sup>191</sup>,
  - Fedora19<sup>192</sup>

### Додаткові аксесуари

|                     |                                                                                                                                               |                                                                                      |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>WiFi</b>         | <b>Mediatek Ralink RT5370</b><br>Частота: 2.4 ГГц<br>Швидкість передачі даних: 150 Mbps<br>Антенa: зовнішня з підсиленням 2dBi                |   |
| <b>LCD (14inch)</b> | <b>RGB2LVDS</b><br>Діагональ 14 дюймів<br>Роздільна здатність 1366 x 768<br>Тип LED Back-light<br>Роз'єм 40pin LVDS Cable<br>Живлення 9В/1.5А |  |

<sup>189</sup> <http://os.archlinuxarm.org/os/ArchLinuxARM-odroid-x-latest.img.gz>

<sup>190</sup> <https://launchpad.net/win32-image-writer>

<sup>191</sup> <http://www.odroid.in/debian-wheezy/>

<sup>192</sup> [http://odroid.in/Fedora19\\_XU/](http://odroid.in/Fedora19_XU/)

|                                  |                                                                                                                                                                                          |                                                                                      |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <p><b>LCD<br/>(10.1inch)</b></p> | <p><b>RGB2LVDS плата</b><br/> Діагональ 10.1 дюймів<br/> Роздільна здатність 1366 x 768<br/> Тип LED Back-light<br/> Роз'єм 40pin LVDS Cable<br/> Живлення 9В/1.5А Adaptor</p>           |    |
| <p><b>Камера</b></p>             | <p><b>Samsung 5Mp K5ECG<br/>MIPI CSI</b><br/> Працює лише на ОС<br/> Android OS.<br/> Ніяких драйверів для<br/> Ubuntu або інших Linux ОС.</p>                                           |    |
| <p><b>eMMC модуль</b></p>        | <p><b>THGBM5G7A2JBAIR</b><br/> Об'єм пам'яті 16 Гбіт<br/> Швидкість передачі даних:<br/> 50 МГц 8-бітовий DDR<br/> інтерфейс, який на 150%<br/> швидший, ніж SD-карти<br/> класу 10.</p> |  |

|                         |                                                                                                       |                                                                                    |
|-------------------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <b>Bluetooth модуль</b> | <b>Міні USB Bluetooth модуль CSR V2.1 EDR</b><br>Швидкість передачі даних: 2Mbps<br>Радіус дії: 10 м. |  |
|-------------------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|

Так як спочатку платформа ODROID створювалася для ОС Android (ODROID - Open anDROID), то не дивно, що саме під нею все працює найбільш рівно і коректно, в тому числі 3D-прискорення і апаратне декодування відео.

Також, як показала практика, не всі SD-карти класу 10 мають однакову продуктивність, що в результаті призводить до гальмування при виконанні операційною системою дискових операцій. Тому рекомендується встановлювати ОС на EMMC, а SD-карту використовувати тільки для зберігання даних.

Якщо ж брати до уваги використання мікрокомп'ютера для навчання та експериментів, то тут беззаперечним лідером залишається Raspberry Pi.

### Інші альтернативні заміни Raspberry Pi

Cubieboard<sup>193</sup>

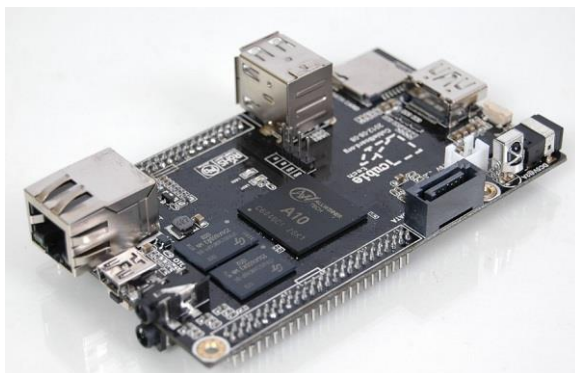


Рис.10.33 Плата Cubieboard

<sup>193</sup> <http://docs.cubieboard.org/products/start>

|                        |                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Процесор:</b>       | ARM cortex-A8 1ГГц, 256К6 L2 кеш                                                                                                             |
| <b>3D акселератор:</b> | Mali-400                                                                                                                                     |
| <b>Пам'ять:</b>        | 512М6/1Г6 DDR3 480 МГц, 4 Г6 флеш                                                                                                            |
| <b>Відео вихід:</b>    | HDMI 1080p                                                                                                                                   |
| <b>Роз'єми:</b>        | 10/100 М6 Ethernet, 2 USB, 1 мікро SD слот, 1 SATA, роз'єм на 96 виводів, I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP |
| <b>ОС:</b>             | Android, Ubuntu                                                                                                                              |

Snowball<sup>194</sup>



Рис.10.34 Плата Snowball

|                        |                                                                                                                                                                                                                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Процесор:</b>       | ST-Ericsson Nova A9500 1ГГц (Cortex-A9)                                                                                                                                                                                                                                                    |
| <b>3D акселератор:</b> | Mali-400, 400МГц                                                                                                                                                                                                                                                                           |
| <b>Пам'ять:</b>        | 1Г6 DDR2, 4/8 Г6 флеш e.MMC                                                                                                                                                                                                                                                                |
| <b>Відео вихід:</b>    | HDMI 1080p                                                                                                                                                                                                                                                                                 |
| <b>Роз'єми:</b>        | мікро-SD, CVBS відео вихід, аудіо вихід/вхід, Ethernet 10/100М6іт, USB OTG HS (480Mbits), IEEE 802.11 b/g/n WLAN, Bluetooth 4.0/BLE(із зовнішньою антеною), GPS (із зовнішньою антеною), акселерометр, магнетометр, гірометр, датчик тиску (300-1100 мілібар), UART, SPI, I2C, GPIO, JTAG. |
| <b>ОС:</b>             | Linux BSP                                                                                                                                                                                                                                                                                  |

BeagleBone Black<sup>195</sup>

<sup>194</sup> <http://www.cnx-software.com/2011/08/01/calao-snowball-development-boards-are-now-available/>

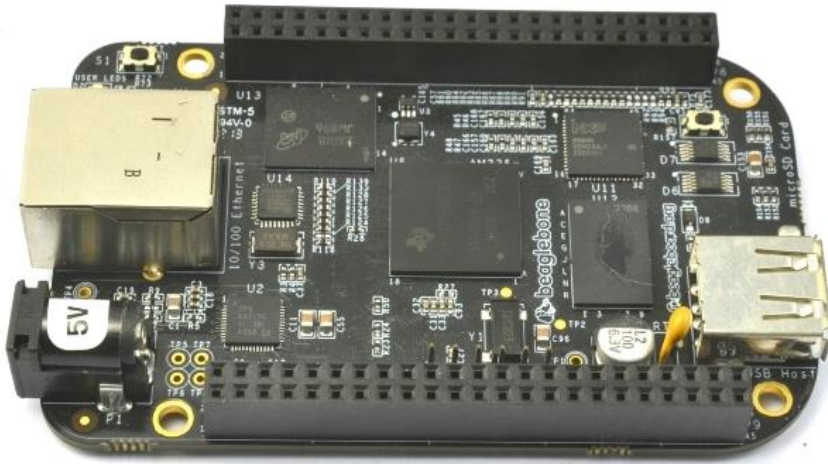


Рис. 10.35 Плата BeagleBone Black

|                        |                                      |
|------------------------|--------------------------------------|
| <b>Процесор:</b>       | AM335x 1ГГц ARM Cortex-A8            |
| <b>3D акселератор:</b> | SGX530                               |
| <b>Пам'ять:</b>        | 512 Мб DDR3 800МГц, 2 Гб флеш, eMMC  |
| <b>Відео вихід:</b>    | HDMI 1080p                           |
| <b>Роз'єми:</b>        | Ethernet, 2-х USB хост, 2х46 виводів |
| <b>ОС:</b>             | Android, Ubuntu                      |

Origen 4<sup>196</sup>

<sup>195</sup> <http://beagleboard.org/products/beaglebone%20black>

<sup>196</sup> [http://www.origenboard.org/wiki/index.php/Main\\_Page](http://www.origenboard.org/wiki/index.php/Main_Page)

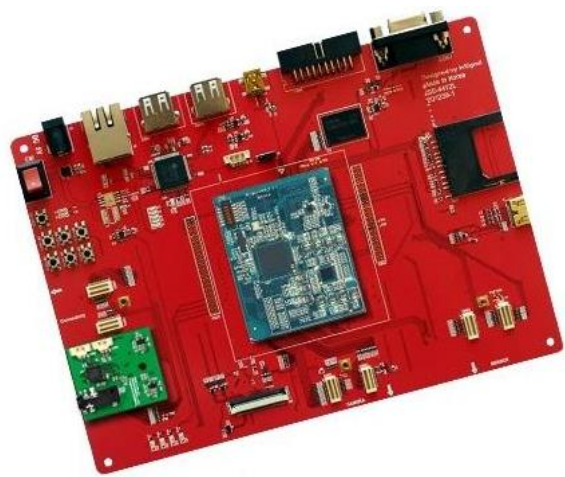


Рис.10.36 Плата Origen 4

|                        |                                                                                                     |
|------------------------|-----------------------------------------------------------------------------------------------------|
| <b>Процесор:</b>       | Samsung Exynos 4 Quad Cortex-A9 core 1.4 GHz                                                        |
| <b>3D акселератор:</b> | Mali-400 Quad Core 440 МГц                                                                          |
| <b>Пам'ять:</b>        | 1 Гб, 4 Гб e.MMC                                                                                    |
| <b>Відео вихід:</b>    | HDMI 1080p                                                                                          |
| <b>Роз'єми:</b>        | SD-карта, USB 2.0 x 2, JTAG, Ethernet (10/100 Мбіт), HDMI, MIPI CSI/DSI , LCD: 7" LCD (1024 x 600). |
| <b>ОС:</b>             | Linaro                                                                                              |

Arndale Board<sup>197</sup>

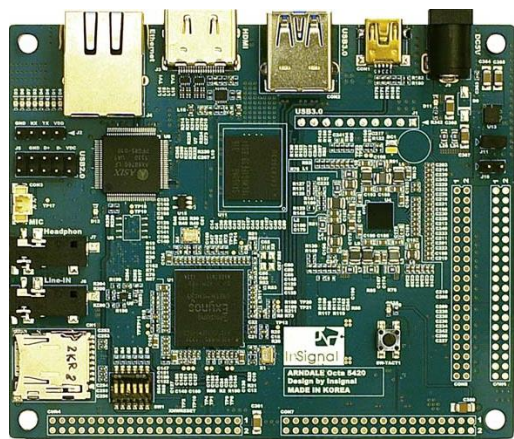


Рис.10.37 Плата Arndale Board

<sup>197</sup> [http://www.arndaleboard.org/wiki/index.php/Main\\_Page](http://www.arndaleboard.org/wiki/index.php/Main_Page)

|                        |                                                                                                                                     |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Процесор:</b>       | Exynos 5420: ARM Cortex-A15 1.8ГГц + ARM Cortex-A7 1.2 GHz                                                                          |
| <b>3D акселератор:</b> | OpenGLES1.1/2.0                                                                                                                     |
| <b>Пам'ять:</b>        | 800МГц DDR3(L)/DDR3 1Г6 x 2                                                                                                         |
| <b>Відео вихід:</b>    | HDMI 1080p                                                                                                                          |
| <b>Роз'єми:</b>        | HDMI, USB 3.0 хост x1, USB 2.0 x 2 хост, 10/100Мбіт Ethernet, eMMC 4.41 DDR 86іт 4Г6, SDIO 3.0, SD2.0, UART, 3x SPI, 3x I2S, 4x I2C |
| <b>ОС:</b>             | Linaro                                                                                                                              |

Rikomagic Quad Core MK802IV<sup>198</sup>



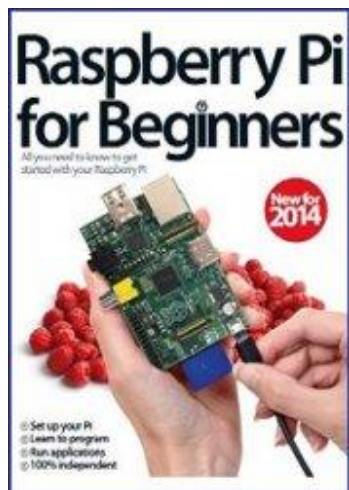
Рис.10.38 Плата Rikomagic Quad Core MK802IV

|                        |                                                     |
|------------------------|-----------------------------------------------------|
| <b>Процесор:</b>       | Rockchip 3188 1.8ГГц Cortex-A9                      |
| <b>3D акселератор:</b> | Mali-400 533МГц                                     |
| <b>Пам'ять:</b>        | 2GB DDR3, 8GB/16GB енергонезалежної пам'яті         |
| <b>Відео вихід:</b>    | HDMI 1080p                                          |
| <b>Роз'єми:</b>        | HDMI, 1 x USB 2.0 хост, 2 X мікро USB, 1 x мікро SD |
| <b>ОС:</b>             | Android 4.2                                         |
| <b>Вартість:</b>       | 700 грн                                             |

<sup>198</sup> [http://www.rikomagic.com/en/product/showpro\\_id\\_39\\_pid\\_19.html](http://www.rikomagic.com/en/product/showpro_id_39_pid_19.html)

# 11. КОРОТКІ АНОТАЦІЇ КОРИСНИХ КНИГ

## Raspberry Pi для початківців



Raspberry Pi for Beginners<sup>199</sup> | Revised Edition | 2014 | 180 p. | PDF | Eng. | 44.2 Мб.

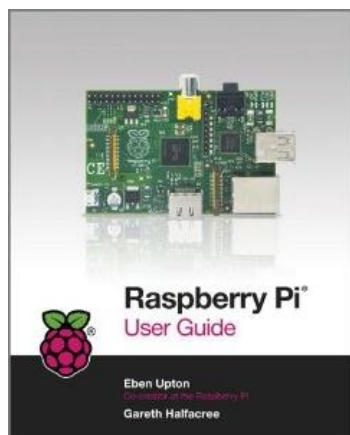
Показано, як зібрати свій Raspberry Pi, які дистрибутиви і периферійні пристрої кращі для того, щоб Ви могли стартувати з Raspbian.

Розглянуті основи програмування Scratch і Python, з керівництвами крок за кроком для створення деяких фантастичних проектів з Вашим Pi.

Розділи:

- Основи - як налаштувати Raspberry Pi і встановити останню ОС для ІТ-проектів.
  - Зарядитися натхненням і навчитися створювати дещо з Вашими власними програмами - навчаємося і освоюємо ази Scratch і Python.
- Проблеми і глосарій - обладнання та усунення неполадок програмного забезпечення, відповіді на поширені питання та глосарій.

## Керівництво користувача Raspberry Pi



Raspberry Pi User Guide<sup>200</sup> - Eben Upton, Gareth Halfacree | John Wiley & Sons | ISBN-10 111846446X | ISBN-13: 978-1118464465 | 2012 | 262 p. | Eng. Основне керівництво для початку вивчення комп'ютера і програмування з Raspberry Pi.

Спочатку задуманий, як веселий і простий спосіб для дітей (і зацікавлених дорослих), щоб дізнатися про програмування для комп'ютера, Raspberry Pi швидко перетворився в дивно надійний, розміром в кредитну карту комп'ютер, який може бути використаний для всього від перегляду HD-відео і злому обладнання навколо до навчання з

програмування! У співавторстві з одним з творців Raspberry Pi, ця книга наповнить Вас всім, що Вам потрібно знати, щоб приступити до роботи на Raspberry Pi в найкоротші терміни, включаючи такі теми, як:

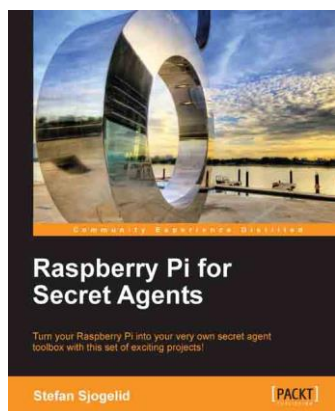
<sup>199</sup> <http://isearch.kiev.ua/uk/news/books/1810-e-book-raspberry-pi-for-beginners>

<sup>200</sup> <http://isearch.kiev.ua/uk/news/books/1755-raspberry-pi-user-guide>

- Підключення клавіатури, миші, монітора та інших периферійних пристроїв
- Встановлення програмного забезпечення
- Основи системного адміністрування Linux
- Налаштовування Raspberry Pi
- Налаштувати Raspberry Pi як продуктивного комп'ютера
- Написання програм на Python і Scratch
- Встановлення Raspberry Pi для керування мультимедійним центром

Це повне керівництво з використання та отримання максимальної віддачі від першого у світі істинно компактного комп'ютера.

## Raspberry Pi для таємних агентів



Raspberry Pi for Secret Agents<sup>201</sup> - Stefan Sjogelid | PDF, EPUB | ISBN: 1849695784 | 2013 | 152 p. | Eng.

Книга з провокаційною назвою, яка допоможе перетворити Ваш Raspberry Pi у власний інструментарій таємного агента з набором цікавих проектів!

Може коли-небудь Ви хотіли пограти з усіма акуратними гаджетами, які використовують Ваші улюблені шпигуни (Джеймс Бонд або Майкл Westen)? З появою чудового Raspberry Pi і кількох USB-аксесуарів, тепер будь-хто може повторити їх операції.

Дізнайтеся, як перетворити Ваш Raspberry Pi в багатофункціональний інструмент таємного агента! Через ряд веселих і легких проектів Ви дізнаєтеся, як створити аудіо/відеоспостереження, дослідити Ваші Wi-Fi-мережі, розіграти Ваших друзів, і навіть дізнатися, як звільнити свій Raspberry Pi від обмежень живленням з розетки.

Дізнайтеся, як для записувати, слухати чи говорити з людьми на відстані, і, як спотворити Ваш голос. Ви навіть зможете підключити веб-камеру і налаштувати детектор руху з тривожними повідомленнями. Після того, як Ви освоїте методи, поєднайте їх з акумуляторною батареєю та GPS для завершального налагодження комплекту шпигуна.

Деякі з тем, які вивчаються в книзі:

- Створення відео потоків з MJPG-стрімером
- Налаштування виявлення руху і сигналізації
- Планування аудіо/відеозаписів
- Управління Pi з телефону Android
- Побудова карти Вашої мережі з Nmap
- Натискання несподіваних зображень у вікні браузера з Ettercap
- Відстеження Pi на Google Earth за допомогою GPS

<sup>201</sup> <http://isearch.kiev.ua/uk/news/books/1756-e-book-raspberry-pi-for-secret-agents>

- Відправка періодичних твітів від Pi з GPS координатами
- Створення зашифрованого сховища з Cryptsetup
- Створення ярликів команд з псевдонімами
- Збереження Ваших записів, безпечно запущене з tmux

Не потрібно жодних попередніх навичок, щоб слідувати за автором, і якщо Ви новачок в Linux, то безкоштовно отримаєте більшу частину основ.

## **Висновки**

Продажі одноплатного комп'ютера Raspberry Pi вже взяли рубіж в 3 млн. пристроїв.

Останній мільйон був проданий всього за шість місяців. Таким чином, можна побачити, що попит на Raspberry Pi за останній час тільки виріс. Варто враховувати, що цьому ніяк не посприяла нова модифікація - Raspberry Pi Compute Module, анонсована в квітні, так як її продажі тільки почалися. Тому, цілком можливо, що наступний мільйон компанія продасть ще швидше.

15 тис. пристроїв передав школярам в США Google. Такі відомі навчальні заклади, як Массачусетський технологічний інститут та Кембріджський університет, створили спеціальні програми для поглибленого вивчення програмування з використанням платформи Raspberry Pi.

Тому є надія, що допитливий розум наших читачів нарешті отримав можливість для своєї реалізації і нинішнє покоління дослідників подарує світу нових геніїв з інформаційних технологій.

# Алфавітний вказівник команд та англомовних термінів

При першому використанні команд (термінів) в тексті, як правило, наведений короткий їх опис чи визначення. Вказівник допоможе читачу знайти сторінки, на яких описаний даний термін, або зустрічається приклад використання команди.

| Терміни            | Стор.  | Терміни             | Стор.   |
|--------------------|--------|---------------------|---------|
| address            | 57     | daemon              | 61      |
| allow-hotplug      | 63     | dd                  | 11, 182 |
| androidVNC         | 103    | ddclient            | 60      |
| apache             | 130    | deb                 | 42      |
| apt-cache search   | 38     | destination         | 56      |
| apt-get            | 36     | dhcp                | 55, 63  |
| apt-get clean      | 40     | dhcpcd.conf         | 68      |
| apt-get install    | 28, 39 | dmesg               | 67      |
| apt-get remove     | 40     | domain-name         | 69      |
| apt-get update     | 17, 37 | dpkg (-reconfigure) | 28, 137 |
| apt-get upgrade    | 17, 40 | du                  | 21      |
| arduino IDE        | 305    | easy_RSA            | 143     |
| arndale Board      | 330    | echo                | 26      |
| bcast              | 56, 57 | e-Health            | 309     |
| beagleBone Black   | 329    | ES File Explorer    | 111     |
| boot_behavior      | 14     | eth0                | 55      |
| cat                | 22, 26 | ffmpeg              | 131     |
| change_locale      | 14     | ffserver            | 132     |
| change_pass        | 14     | Fing                | 98      |
| chmod              | 52     | free                | 25      |
| coder              | 195    | ftpuser             | 88      |
| compute Module     | 302    | fugu                | 154     |
| configure_keyboard | 14     | gateway             | 56      |
| ConnectBot         | 101    | genkey              | 149     |
| console-setup      | 33     | geometry            | 96      |
| copy.com           | 210    | git                 | 131     |
| cp                 | 21, 26 | GPIO                | 213,    |
| crontab            | 64,173 | GPS                 | 260     |
| GPU                | 8, 25  | memory_split        | 14, 106 |

|                        |        |                    |         |
|------------------------|--------|--------------------|---------|
| hacker'sKeyboard       | 101    | Midnight Commander | 86      |
| hamachi                | 137    | Midori             | 16      |
| hostapd                | 67     | Minecraft          | 284     |
| I2P                    | 167    | Mini Remote        | 229     |
| i2ptunnel              | 173    | mkdir              | 20, 80  |
| iceweasel              | 280    | modprobe           | 64, 219 |
| iface                  | 56, 57 | motion             | 124     |
| ifconfig               | 56, 57 | mount              | 53, 107 |
| ifup                   | 71     | mplayer            | 117     |
| ImageWriter            | 10     | mv                 | 21      |
| inet addr              | 56, 57 | MySQL              | 183     |
| interfaces             | 63, 73 | nano               | 25      |
| IP Scanner             | 75     | netcat (nc)        | 117     |
| iptables               | 71,162 | netmask            | 57      |
| isc-dhcp-server        | 67     | netstat            | 56, 57  |
| jdk                    | 168    | network            | 57      |
| JuiceSSH               | 168    | NetworkManager     | 74      |
| Kali Linux             | 181    | NOOBS              | 10      |
| keyboard-configuration | 28     | notebook           | 250     |
| Kippo                  | 183    | ntfs-3g            | 53      |
| lighttpd               | 188    | Odroid             | 323     |
| lirc                   | 230    | OMXPlayer          | 119     |
| LogMeIn                | 135    | Onion Pi           | 157     |
| ls                     | 21     | openssl            | 148     |
| ls -lha                | 21     | OpenVPN            | 143     |
| ls-lR                  | 81     | Origen             | 330     |
| LXTerminal             | 19     | overclock          | 14      |
| LXDE                   | 18, 49 | overscan           | 13      |
| make                   | 68,125 | Pagent.exe         | 76      |
| man                    | 21     | php5-cgi           | 189     |
| mask                   | 56, 57 | Pi Co-op           | 304     |
| Mathematica            | 250    | pifm               | 288     |
| MathLink               | 266    | ping               | 59, 65  |
| max-lease-time         | 69     | PIR                | 237     |
| mc                     | 86     | PiTFT              | 298     |
| meminfo                | 22     | PiTranslate        | 296     |
| Privoxy                | 138    | ssh                | 75      |

|                     |        |                   |         |
|---------------------|--------|-------------------|---------|
| PSCP.exe            | 76, 85 | sshd              | 83      |
| psftp               | 85     | subnet            | 69      |
| PSFTP.exe           | 76     | sudo              | 20      |
| PuTTY.exe           | 76     | sudo shutdown     | 20      |
| PuTTYgen.exe        | 76     | synaptic          | 43      |
| pwd                 | 19     | sysctl            | 150     |
| Pwn                 | 179    | sysctl.conf       | 70, 150 |
| PwnBerryPi          | 180    | tar               | 67, 183 |
| PwnPi               | 175    | TeamViewer        | 104     |
| python              | 218    | Tightvncserver    | 95      |
| range               | 69,252 | tor               | 161     |
| RaspCamBot          | 273    | torrc             | 161     |
| Raspbian            | 10     | touch             | 26, 164 |
| raspi-config        | 13, 20 | transmission      | 108     |
| raspicam            | 127    | Tunnelblick       | 155     |
| Raspirobot          | 273    | unclutter         | 283     |
| raspistill          | 116    | usermod           | 109     |
| raspivid            | 115    | vcgencmd          | 23      |
| RDP                 | 91     | vim               | 26      |
| reboot              | 72     | VLC               | 128     |
| restart             | 36, 74 | VNC               | 95      |
| Rikomagic Quad Core | 331    | vncserver         | 96      |
| rm                  | 21,127 | vncviewer         | 96, 99  |
| route               | 56     | VPN               | 134     |
| RPi-sd Card Builder | 10     | VX ConnectBot     | 101     |
| RTSP                | 128    | webiopi           | 274     |
| Samba               | 106    | wget              | 67, 211 |
| Secure Shell        | 74     | Wheezy            | 10      |
| Server Auditor      | 103    | WiFi Config       | 15      |
| service             | 107    | Win32 Disk Imager | 10, 11  |
| setupcon            | 28     | WinSCP            | 88      |
| setxkbmap           | 32     | Wolfram           | 245     |
| sh                  | 21,163 | wpa_supplicant    | 279     |
| shutdown            | 21     | wpa-conf          | 63      |
| SIM90x              | 314    | Xming             | 93      |
| Snowball            | 328    | xrdp              | 91      |

## **Рекомендована література**

1. *Barnes R.* Raspberry Pi. The Complete Manual / Russell Barnes - Imagine Publishing Ltd, 2014. - 148 c.
2. *Bell C.* Beginning Sensor Networks with Arduino and Raspberry Pi / Charles Bell - Apress Media, 2013. - 404 c.
3. *Bradbury A.* Learning Python with Raspberry Pi / Alex Bradbury, Ben Everard - John Wiley & Sons Ltd, 2014. - 288 c.
4. *Horan B.* Practical Raspberry Pi / Brendan Horan - Apress Media, 2014. - 260 c.
5. *Monk S.* Raspberry Pi Cookbook / Simon Monk - O'Reilly Media, Inc., 2014. - 408 c.
6. *Suehle R.* Raspberry Pi Hacks / Ruth Suehle, Tom Callaway - O'Reilly Media, Inc., 2014. - 392 c.
7. *Philbin C.A.* Adventures in Raspberry Pi / Carrie Anne Philbin - John Wiley & Sons Ltd, 2014. - 256 c.
8. *Robinson A.* Raspberry Pi Projects / Dr. Andrew Robinson, Mike Cook, Jonathan Evans, Sean McManus - John Wiley & Sons Ltd, 2014. - 480 c.

*«Ти хочеш все життя продавати солодку воду чи хочеш мати шанс змінити весь Світ?»- Steven P. Jobs, «Odyssey: Pepsi to Apple: A Journey of Adventure, Ideas, and the Future» (1987)*

**Інститут комп'ютерних технологій Університету «Україна»**

Інститут комп'ютерних технологій Університету «Україна» - тільки **справжні ІТ напрями**: комп'ютерна інженерія, програмна інженерія, комп'ютерні науки.

П'ять рівнів підготовки: молодший спеціаліст, бакалавр, спеціаліст, магістр, аспірант – обирай сходинку на якій ти хочеш зупинитися, а краще не зупиняйся!

**Молодші спеціалісти:** 5.05010201 «Обслуговування комп'ютерних систем і мереж» та 5.05010301 «Розробка програмного забезпечення»

**Бакалаврат:** 6.050101 «Комп'ютерні науки», 6.050102 «Комп'ютерна інженерія» та 6.050103 «Програмна інженерія».

**Магістратура (та спеціалісти):** 8.05010105 «Комп'ютерний еколого-економічний моніторинг», 8.05010201 «Комп'ютерні системи та мережі» та 8.05010301 «Програмне забезпечення автоматизованих систем»

**Аспірантура:** 05.13.05 «Комп'ютерні системи та компоненти»

Всі навчальні програми розроблені відповідно до побажань міжнародних комп'ютерних (ACM) та інженерних організацій (IEEE).

Найбільша кількість годин програмування серед усіх ВНЗ!

Усі студенти можуть безкоштовно використовувати продукти Microsoft, AutoDesk, InterSystems та IBM.

При інституті діє IT-Academy від Microsoft та Cisco Academy.

Сучасні комп'ютери та безкоштовний WiFi.

E-learning система підтримки навчального процесу, доступ до знань 24x7!

Тісна співпраця ІКТ із провідними ІТ-компаніями України та світу — ми маємо намір віддати вас у гарні руки.

Науково-практичні нароби Інституту комп'ютерних технологій:

**Аутсорсинг:** Розробка програмного забезпечення, комп'ютерних ігор, інформаційних систем, веб-систем, веб-додатків, суміжна діяльність (дизайн, управління контентом, тощо), тестування додатків та систем.

**Робототехніка, автоматизація та системи штучного інтелекту.**

**Комп'ютерні системи та мережі:** Grid-систем, Cloud-систем, Data-minning.

**Системи захисту інформації та персоналізації технічних систем.**

Постійно відбуваються довготривалі майстер-класи та семінари для студентів та викладачів ІКТ: Java-developer, Web-developer, QA та «Основи робототехніки».

**Приходьте, ми завжди готові до співпраці!** <http://inkt.in.ua/>

**Приймальна комісія:** м. Київ, вул. Львівська 23, корп. А, каб. 104  
+ 380-44-409-27-62, + 380-44-424-70-08, + 380-67-441-19-34

Навчальне видання

**Могильний Сергій Борисович**

# **МІКРОКОМП'ЮТЕР RASPBERRY PI – ІНСТРУМЕНТ ДОСЛІДНИКА**

**Посібник**

**В авторській редакції**

**Формат 70x100/16. Умовн. друк. арк. 27,63.**

**Наклад 400 пр. Зам. № 56.07-14.**

**Видавець і виготовлювач ТОВ «Талком»**

**03115, м. Київ, вул. Львівська, 23, тел./факс (044) 424-40-69, 424-56-26**

**E-mail: [ukraina.vdk@mail.ru](mailto:ukraina.vdk@mail.ru)**

**Свідоцтво суб'єкта видавничої справи ДК № 4538 від 07.05.2013**