

# Minecraft and Python on Pi

台灣樹莓派 <sosorry@raspberrypi.com.tw>  
Jul 24, 2015/minecraft #01

# CC (Creative Commons)

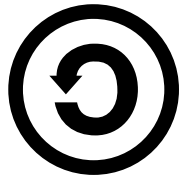
## 姓名標示 — 非商業性 — 相同方式分享



姓名標示 — 你必須給予適當表彰、提供指向本授權條款的連結，以及指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。



非商業性 — 你不得將本素材進行商業目的之使用。



相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。



# about 台灣樹莓派

- element14 指定台灣地區 Raspberry Pi 個人用戶經銷商

The screenshot shows the 'BUY A PI' section of the Raspberry Pi website. A navigation bar at the top includes links for BLOG, DOWNLOADS, COMMUNITY, HELP, FORUMS, and RESOURCES. Below this is a 'BUY' button. The main content area features a 'BUY A PI' icon and text: 'Buy a Pi and accessories from one of our distributors:'. A yellow arrow points from this text to the 'element14' logo. Below the logo is a section titled 'Raspberry Pi Approved Resellers' with a red underline. The text reads: 'To purchase Raspberry Pi or Accessories from one of our Approved Resellers please choose from a reseller below'. A purple arrow points from this text to a list of resellers under the heading 'Raspberry Pi Resellers by region - Asia Pacific'. The list includes: Auseparts (Australia), AusPi Technologies (Australia), Little Bird Electronics (Australia), Wiltronics (Australia), Leocom (Japan), Eleparts Co. (Korea), Icbang (Korea), Leocom (Korea), Orei Solutions (PVT) (Sri Lanka), Xiao Xiao Pang (Taiwan), Globaltronic Intertrade (Thailand), and Quoc Viet Technology JSC (Vietnam). A small Raspberry Pi logo is visible in the top right corner of the page.

# about 台灣樹莓派

- 專注於 Raspberry Pi 應用與推廣
- 舉辦社群聚會 / 工作坊 / 讀書會 / 黑客松

• Website :

- <https://www.raspberrypi.com.tw/>

• Facebook :

- 搜尋 RaspberryPi.Taiwan

- <https://www.facebook.com/RaspberryPi.Taiwan>



# 分享 × 社群

- COSCUP, MakerConf, PyCon 講者
- 投影片
  - <http://www.slideshare.net/raspberrypi-tw/presentations>
- 程式碼
  - <https://github.com/raspberrypi-tw>





# MINECRAFT



# 今日主題

- Minecraft 介紹與使用
- 用 Minecraft 學 Python
- 和硬體互動

# 我的世界

- 起源：
  - 2009 年由瑞典 Markus 'Notch' Persson 獨立開發
- 目的：
  - 讓玩家在遊戲中創造和破壞不同種類的方塊
  - 遊戲本身沒有目的，像是虛擬的 LEGO
- 平台：
  - 單人 / 多人連線
  - Java-based



# 數字

- 支援 8 種平台 (PC, Android, iOS, Windows Phone, Amazon FireTV, Xbox, PlayStation, Raspberry Pi)
- PC 版本售價 26.95 美元
- 史上最暢銷的 PC 遊戲 (2 千萬套, 2015 七月)
- 2014 年被 Microsoft 以 25 億美金收購

# 模式

- Classic 模式
  - 第一個版本，免費但不再維護
- 創造模式 (Creative Mode)
  - 無限使用所有的方塊和物品，玩家是無敵的
- 生存模式 (Survival Mode)
  - 收集資源以進行建造、合成和活下去
- 冒險模式 (Adventure mode)
  - 自創地圖，不能創造和摧毀方塊
- 極限模式 (Hardcore Mode)
  - 像生存模式 ( 只有一命 )，玩家死後世界會被刪除

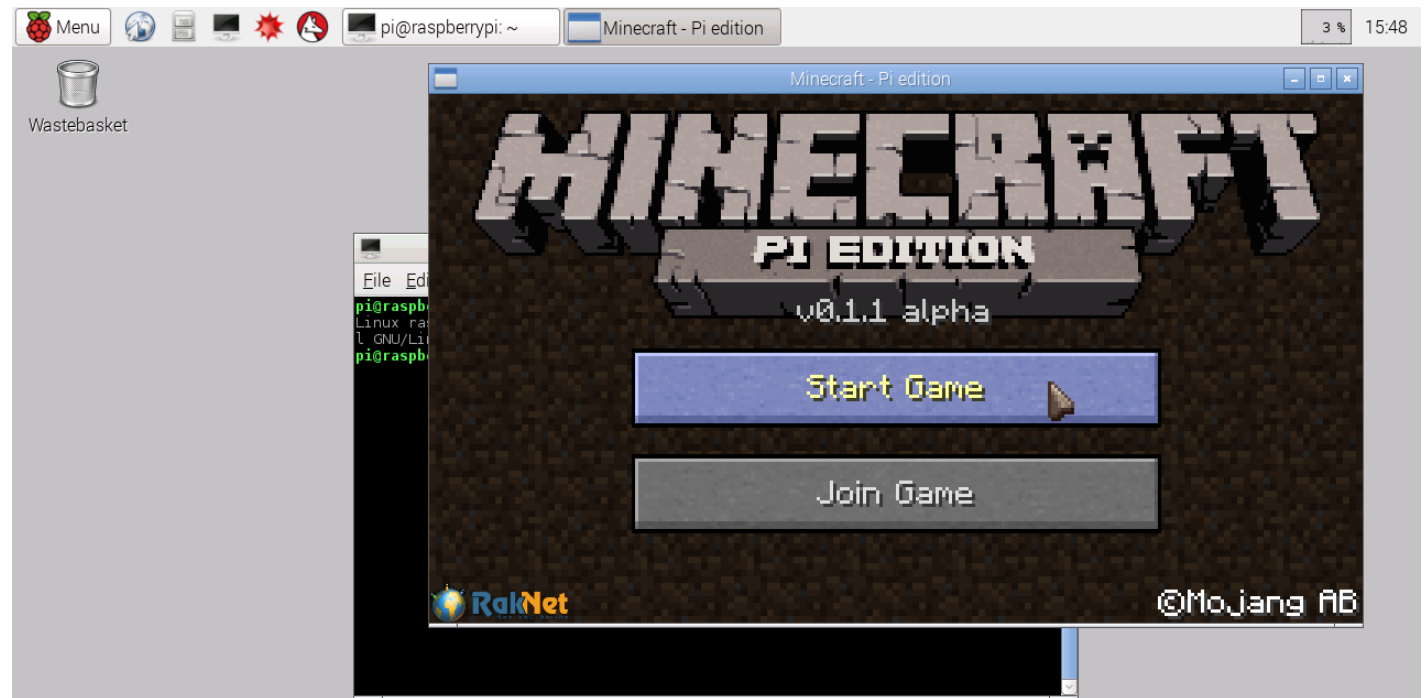
功能	Classic	生存模式	創造模式	冒險模式	極限模式
高級用戶	不需要	多人模式需要 <sup>[註 1]</sup>	需要	需要	需要
可用	部分 <sup>[註 2]</sup>	是	是	是	是
多人聯機	是	是	是	是	部分 <sup>[註 3]</sup>
合成	否	是	是	是	是
燒煉	否	是	是	是	是
釀造	否	是	是	是	是
附魔	否	是	是	是	是
物品欄	部分	是	是	是	是
生命值	僅生存測試	是	隱藏 <sup>[註 4]</sup>	是	是
死亡重生	是	是	是	是	否
放置 / 破壞方塊	是	是	是	部分	是
生物	部分	是	是	是	是
無限方塊	是	部分 <sup>[註 5]</sup>	是	部分 <sup>[註 5]</sup>	部分 <sup>[註 5]</sup>
日夜交替	否	是	是	是	是
作弊	否	可更改	默認為是,但可更改	可更改	否
固定難度	????	否	否	否	是
飢餓	否	部分	否	部分	是
攜帶版可用	????	是	是	否	否

# 在 Ubuntu 10.04 安裝

- 安裝必要套件
  - `sudo apt-get install openjdk-6-jre openjdk-6-jre-headless`
- 下載 jar
  - <https://minecraft.net/download>
- 啟動
  - `java -Xmx1024M -Xms512M -jar Minecraft.jar`

# 在 Pi 安裝

- 2014-09 後的 image 內建 minecraft



# 操作：鍵盤 + 滑鼠

Keyboard	Action	Keyboard	Action
W	Forward	Q	Drop Item
A	Left	F1	Personal Information
S	Down	F2	Screen Capture
D	Right	F3	Location Information
E	Inventory	F5	Switch the Angle of View
Space	Jump		
Double Space	Fly / Fall		
Esc	Pause / Game menu		
Tab	Release mouse cursor		

Mouse	Action
Left	Break
Right	Place

# 在 Pi 安裝

- 2014-09 後的 image 內建 minecraft



# 遊戲選單 (免費版)

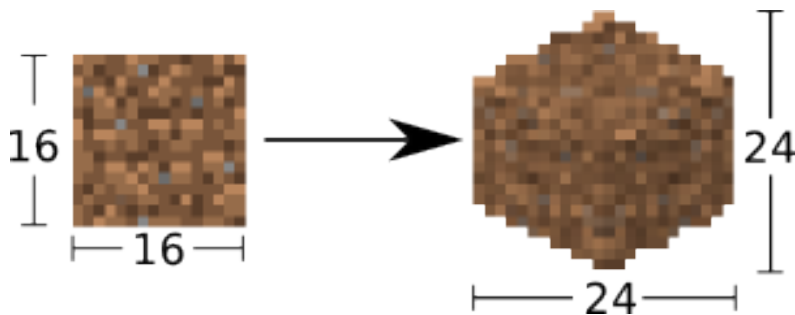
- 2014-09 後的 image 內建 minecraft

# 生存模式遊戲規則

- 規則一：不要死
- 白天 10 分鐘，晚上 7 分鐘，日出日落各 1.5 分鐘
- 打怪，撿寶，生產，建築

# Block

- 整個遊戲以 block 組成
- 在 v1.8 有 153 種 block
- 不同 block 有不同的屬性
- 有些可以更換顏色
- 每個 block 為 16x16 像素



# 常用 Block



Stone( 石頭 )/1



Wood( 木頭 )/17



Coal Ore( 煤礦 )/16



Iron Ore( 鐵礦石 )/15

# 合成 (Crafting)

- 某些方塊、工具和資源獲得的方法
- 材料的排列可能有影響 (有序配方)



2x2 的合成



3x3 的合成 (需使用合成桌)

# 燒煉 (Smelting)

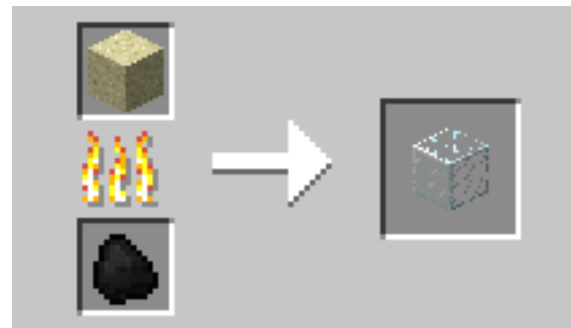
- 燒煉 = 冶煉、燒製或烹飪 = 加熱
- 每次燒煉操作需要花費 10 秒



鐵礦 + 煤炭 + 燒煉 = 鐵錠



生肉 + 煤炭 + 烹飪 = 牛排



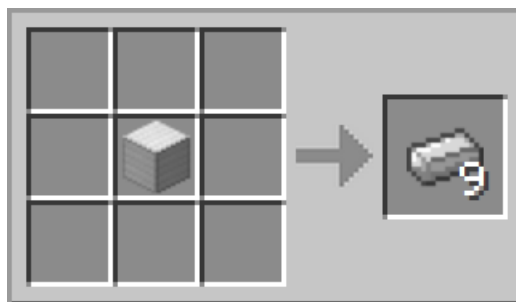
沙子 + 煤炭 + 燒煉 = 玻璃

# 鐵的循環

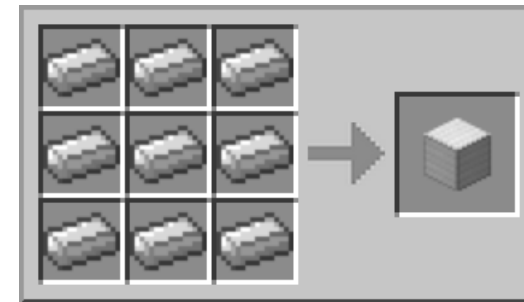
- 鐵錠 (Iron Ingot) 是製作物品的材料
- 鐵錠可以從合成或燒煉取得



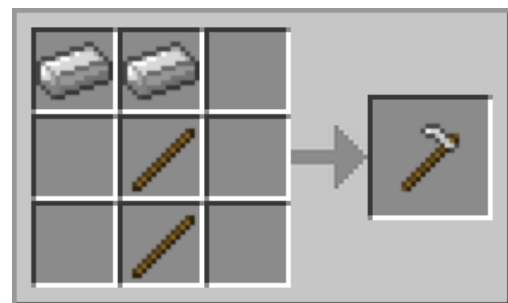
鐵礦 + 煤炭 + 燒煉 = 鐵錠



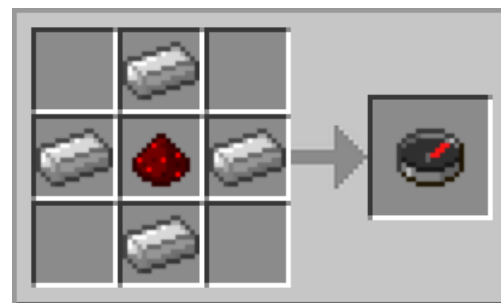
鐵塊 + 合成 = 鐵錠



鐵錠 + 合成 = 鐵塊



鐵錠 + 木棍 = 鐵鋤頭



鐵錠 + 紅石粉 = 指南針



# 打怪

- 被怪物打到會失血
- 打死怪物會有寶物可撿



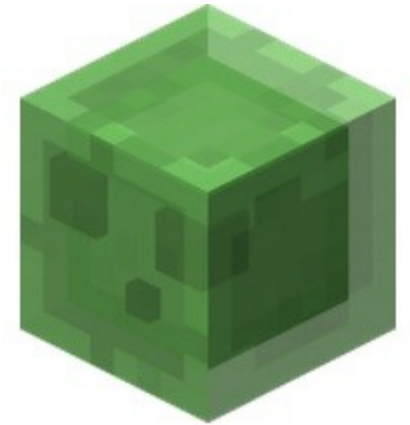
Spider( 蜘蛛 )



Witch( 巫婆 )



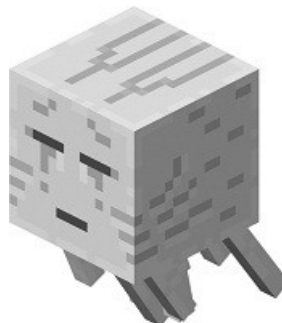
Skeleton( 骷髏 )



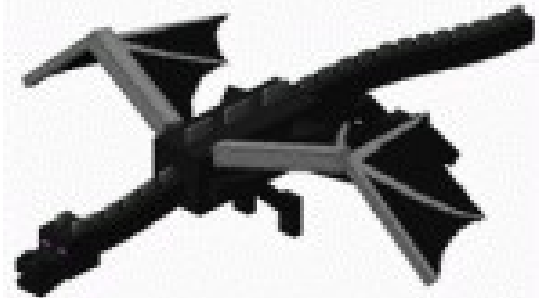
Slime( 史萊姆 )



Zombie( 僵屍 )



Ghast( 地獄幽靈 )



Enderdragon( 末影龍<sup>3</sup> )

# 農業和畜牧

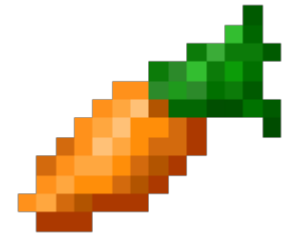
- 農業和畜牧可產生食物
- 植物種植需要土 + 水



Sheep( 羊 )



Chicken( 雞 )



Carrot( 胡蘿蔔 )



Cow( 牛 )



Pig( 豬 )



Wheat( 小麥 )



Seeds( 種子 )

**DEMO**

# MINECRAFT

## PI EDITION

- 版本號：0.1.1
- 最後更新：2/11/2013
- 只有創造模式

# 特色 - 少

- 少很多物品 + 少很多功能 + 少很多按鍵
- 只有 87 種方塊
- 沒有合成和燒煉功能
- 沒有 F1-F5 按鍵
- 場景很少，地圖很小

# 地圖 in Pi Edition





# Inventory in Pi Edition





**DEMO**

# Learn Python with Minecraft

# Minecraft Python API

- Online document
  - <http://www.stuffaboutcode.com/p/minecraft-api-reference.html>
- Source code from
  - <https://github.com/zhuowei/RaspberryJuice/archive/master.zip>
  - <https://s3.amazonaws.com/assets.minecraft.net/pi/minecraft-pi-0.1.1.tar.gz>

# 模組

Syntax: `import MODULE`

Example: `import mcpi.minecraft`

- Syntax: `import MODULE as ALIAS`

Example: `import mcpi.minecraft as minecraft`

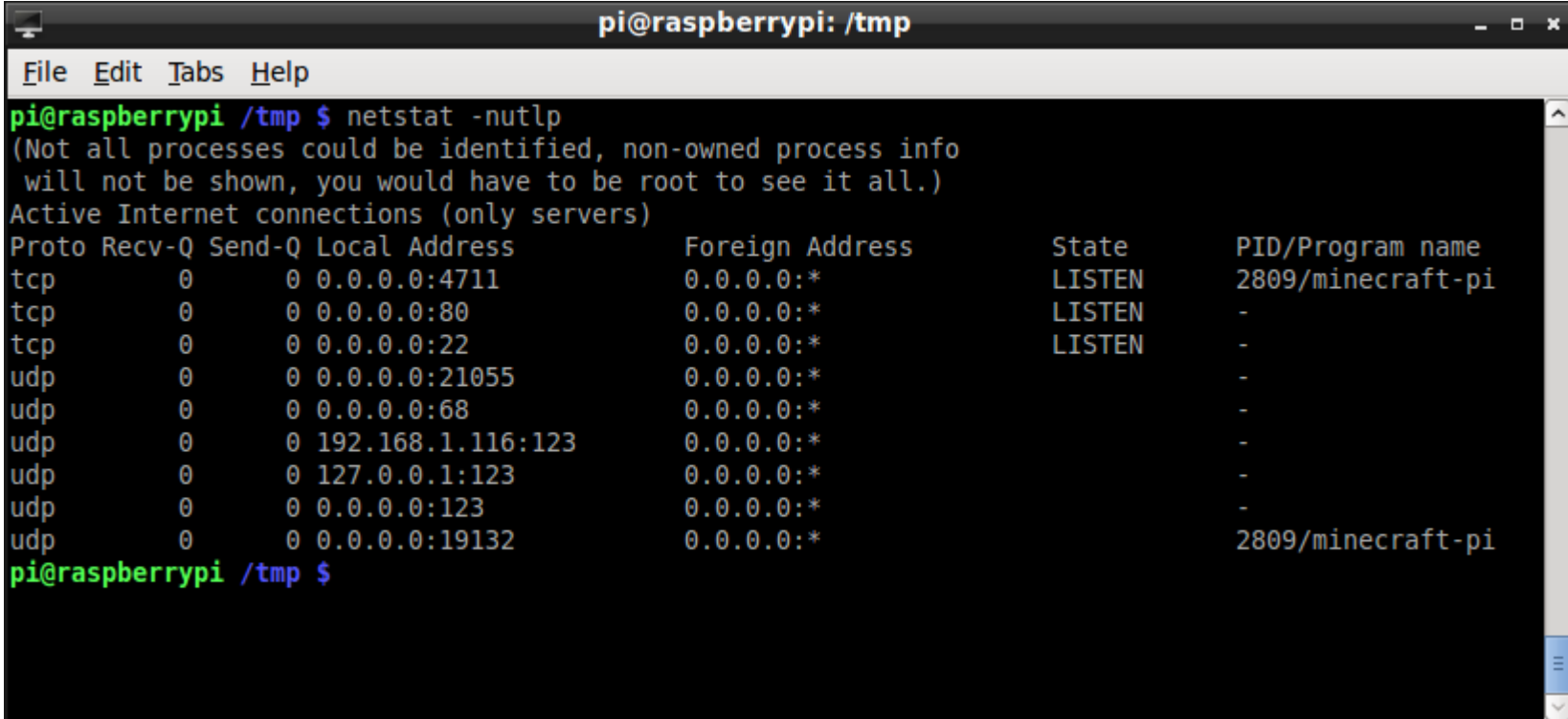
- Syntax: `from MODULE import FUNCTION`

Example: `from time import sleep`

# 如何和遊戲連接？

- 透過 Socket

- `import mcpi.minecraft as minecraft`
- `mc = minecraft.Minecraft.create("127.0.0.1", 4711)`



```
pi@raspberrypi: /tmp
File Edit Tabs Help
pi@raspberrypi /tmp $ netstat -nutlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:4711            0.0.0.0:*                LISTEN      2809/minecraft-pi
tcp        0      0 0.0.0.0:80              0.0.0.0:*                LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      -
udp        0      0 0.0.0.0:21055          0.0.0.0:*                -          -
udp        0      0 0.0.0.0:68             0.0.0.0:*                -          -
udp        0      0 192.168.1.116:123      0.0.0.0:*                -          -
udp        0      0 127.0.0.1:123          0.0.0.0:*                -          -
udp        0      0 0.0.0.0:123            0.0.0.0:*                -          -
udp        0      0 0.0.0.0:19132          0.0.0.0:*                -          2809/minecraft-pi
pi@raspberrypi /tmp $
```

# 變數，物件，型別，註解

- 動態型別 (dynamic typing)

**# 這是註解**

**i = 3 # 變數 i 指到數字物件 3**

**i = [1, 2, 3, 4, 5] # 變數 i 指到串列物件**

**print(i[2]) # 印出串列中第三個元素**

**i = "abcde" # 變數 i 指到字串物件**

**print(i[2]) # 印出字串中第三個元素**

# Print & Chat

- Print: 訊息在終端機畫面顯示
- Chat: 訊息在 minecraft 主畫面顯示
  - Syntax: `.postToChat(message)`

```
import mcpi.minecraft as minecraft
mc = minecraft.Minecraft.create()
mc.postToChat("Hello Minecraft")
```



# Block

- 擺放 block
  - Syntax: `.setBlocks(x0, y0, z0, x1, y1, z1, blockType, blockData)`

```
import mcpi.block as block
```

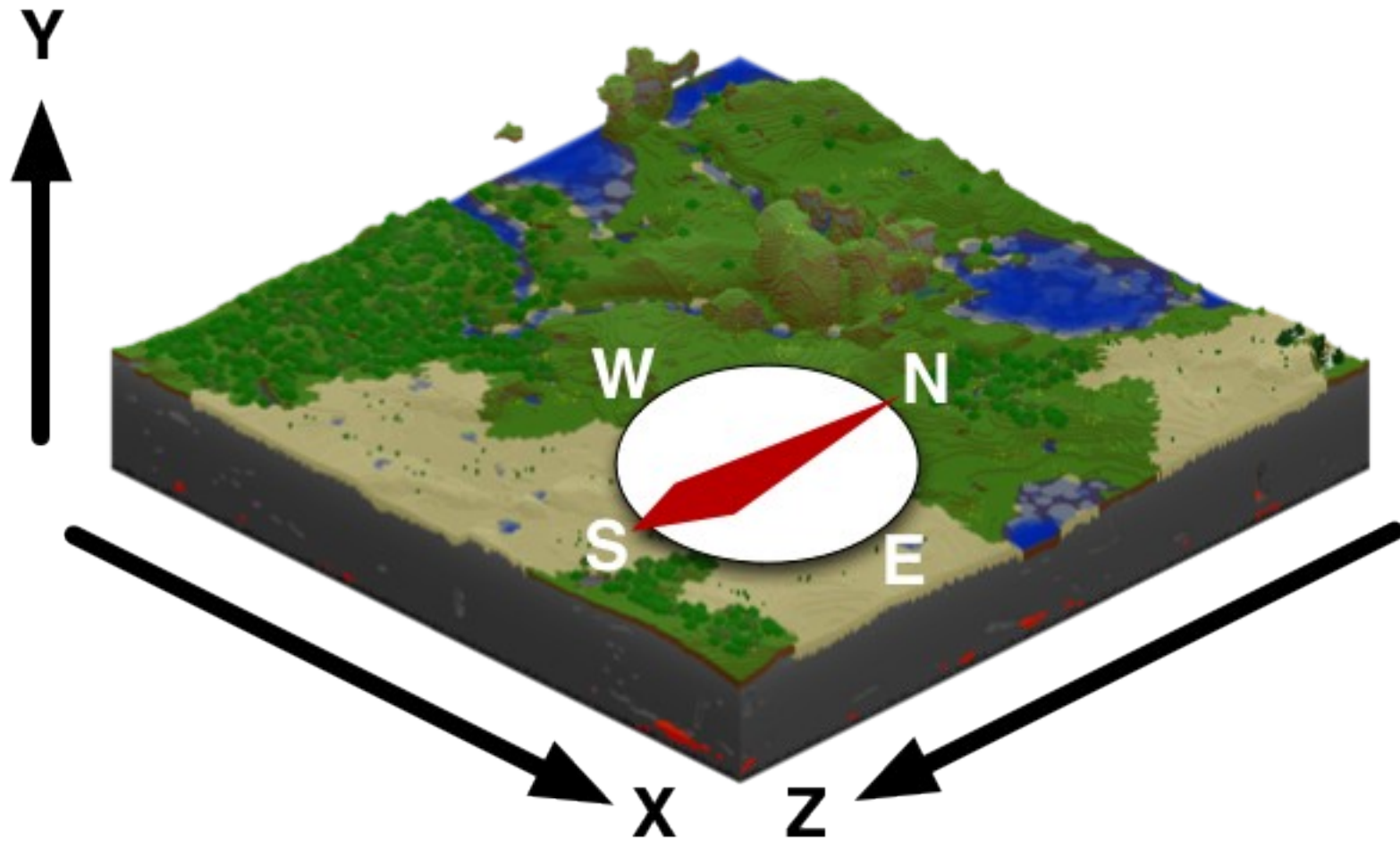
```
mc.setBlock(1, 1, 1, block.WOOD.id, 1)
```

```
mc.setBlocks(-1, -1, -1, 1, 1, 1, block.STONE.id)
```

- 取得 block 屬性
  - Syntax: `.getBlock(x, y, z)`

```
print mc.getBlock(0,0,0)
```

# Coordinate



# Position

- 傳回目前所在位置，以 tile 為計算
  - Syntax: `.getTilePos()`

```
p = mc.player.getTilePos()
```

```
mc.postToChat("x="+str(p.x)+" y="+str(p.y)+" z="+str(p.z))
```

- 傳回目前所在位置，以三維向量 (Vec3 of floats) 計算
  - Syntax: `.getPos()`

```
p = mc.player.getPos()
```

```
print p.x, p.y, p.z
```

# 縮排

- 用縮排取代大括號
- 程式碼的區塊是用縮排分隔
- 不使用 tab, 使用空白鍵
- 常見縮排為 4 個空白鍵

# while loop

- Syntax:

```
while CONDITION :  
    PROCESS
```

```
while True :
```

```
    p = mc.player.getTilePos()
```

```
    mc.postToChat("x="+str(p.x)+" y="+str(p.y)+" z="+str(p.z))
```

```
    time.sleep(1)
```

# for loop

- 自動迭代 (iterator)

- Syntax:

```
for TARGET in LIST :  
    PROCESS
```

```
for x in range(1, 4) :  
    mc.setBlock(x, 1, 1, block.WOOD.id, 3)
```

# if statement

- if CONDITION :  
    PROCESS

```
grass = 2
flower = 38
while True:
    p = mc.player.getTilePos()
    bid = mc.getBlock(p.x, p.y-1, p.z)
    if bid == grass :
        mc.setBlock(p.x, p.y, p.z, flower)
    time.sleep(0.1)
```

# function

- 自動迭代 (iterator)
- Syntax:

```
for TARGET in LIST :  
    PROCESS
```

```
for x in range(1, 4) :  
    mc.setBlock(x, 1, 1, block.WOOD.id, 3)
```



# 打地鼠應用

- 自動迭代 (iterator)

- Syntax:

```
for TARGET in LIST :  
    PROCESS
```

```
for x in range(1, 4) :  
    mc.setBlock(x, 1, 1, block.WOOD.id, 3)
```

# 一秒蓋房子

- 小房子
  - <https://mcpipy.wordpress.com/>
- 醫院
  - <http://home.uktechreviews.com/Raspberry/Pi%20blog/files/Minecraft-hospital.html>
- 金字塔
  - <http://www.raspberrypi-spy.co.uk/2014/09/how-to-create-a-pyramid-in-minecraft-with-python/>
- 城堡
  - <http://www.raspberrypi-spy.co.uk/2014/05/how-to-setup-the-minecraft-python-api-on-a-raspberry-pi/>

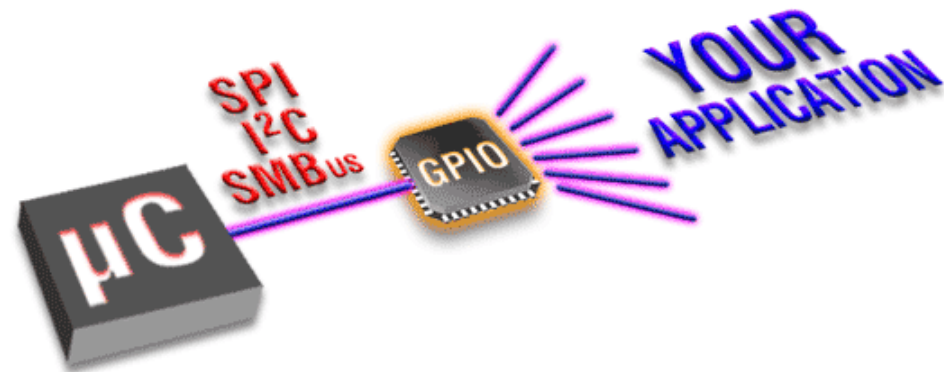
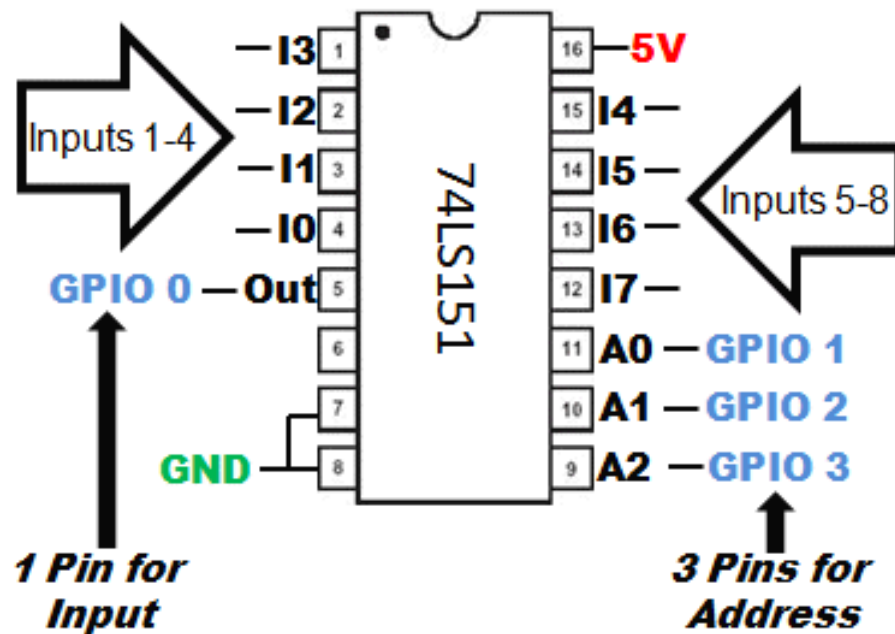
**DEMO**

# 和硬體互動

- Raspberry Pi GPIO 介紹

# General Purpose Input Output(GPIO)

- GPIO is a flexible software-controlled digital signal
- A generic pin on an IC



# GPIO 是硬體的事，那軟體做什麼？

- 決定是輸入還是輸出
- 寫值到某根腳位
- 從某根腳位讀值
- 決定是前緣觸發還是後緣觸發
- 等待中斷 (interrupt) 的發生

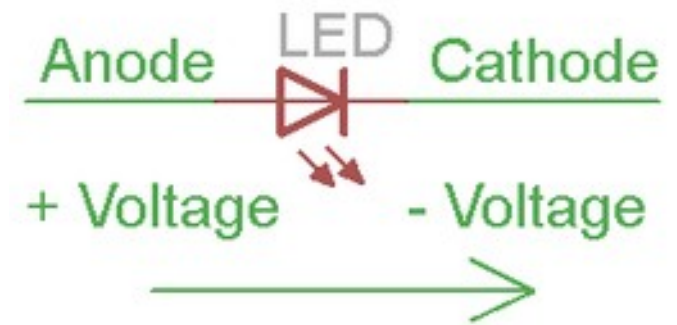
# 1 : 硬體的 Hello World

目的：從硬體到軟體的思維

# LED

- 發光二極體
- 單向導通
- 省電

## 常用符號



短腳接負極 GND

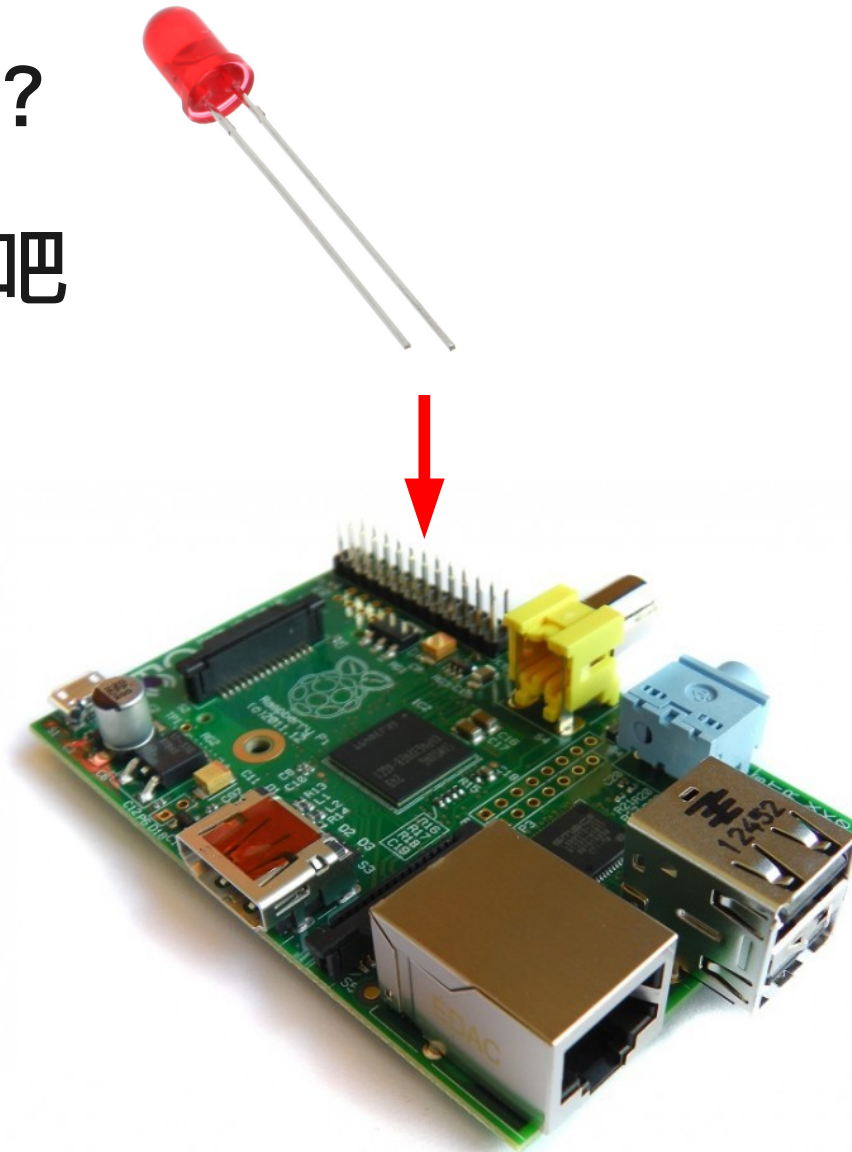
長腳接正極 Vcc



# Raspberry Pi + LED

直接插入嗎？

先看看規格吧



# LED 的規格

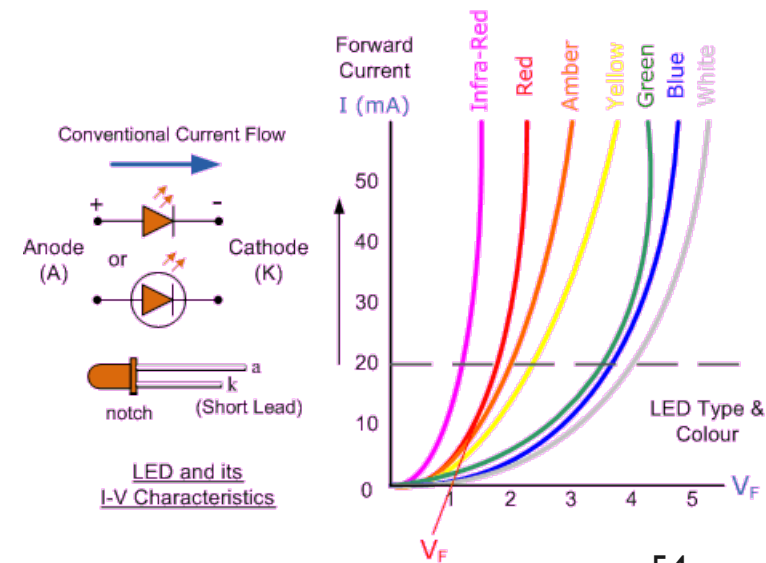
## Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
$\lambda_{\text{peak}}$	Peak Wavelength	Super Bright Red	660		nm	$I_F=20\text{mA}$
$\lambda_D$ [1]	Dominant Wavelength	Super Bright Red	640		nm	$I_F=20\text{mA}$
$\Delta\lambda_{1/2}$	Spectral Line Half-width	Super Bright Red	20		nm	$I_F=20\text{mA}$
C	Capacitance	Super Bright Red	45		pF	$V_F=0\text{V}; f=1\text{MHz}$
$V_F$ [2]	Forward Voltage	Super Bright Red	1.85	2.5	V	$I_F=20\text{mA}$
$I_R$	Reverse Current	Super Bright Red		10	$\mu\text{A}$	$V_R = 5\text{V}$

Notes:

1. Wavelength:  $\pm 1\text{nm}$ .
2. Forward Voltage:  $\pm 0.1\text{V}$ .

- $V_F$  (Forward Voltage) : 順向電壓
- $I_F$  : 在順向電壓下的最大電流



# 如何解讀？

- 由於順向電壓為 1.85V
- 而在順向電壓下 LED 能承受的最大電流是 20mA
- Raspberry Pi 的 GPIO 腳位能提供 3.3V
- 計算公式：電阻 = 電壓 / 電流

$$R = V/I$$

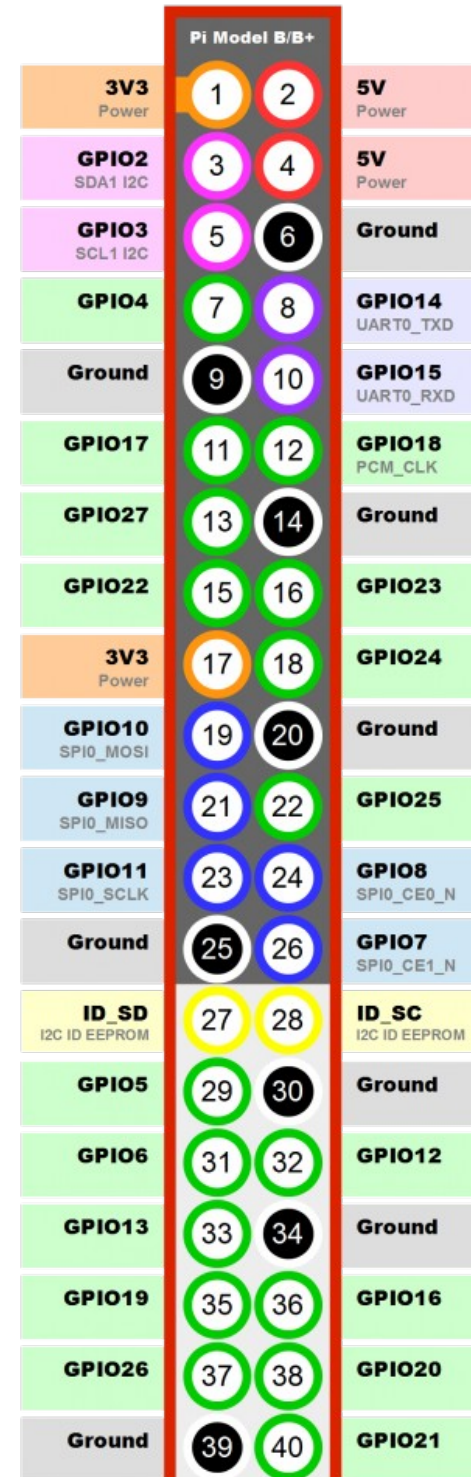
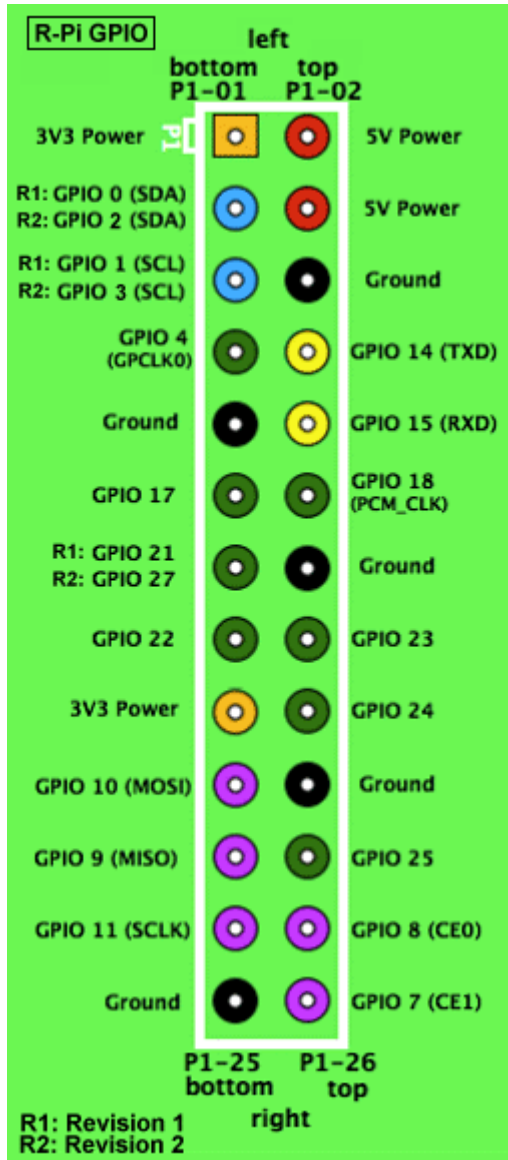
$$R = (3.3-1.85)/0.02 = 72.5 \text{ 歐姆}$$

- 表示最小要接 72.5 歐姆的電阻，才能避免 LED 燒毀
- 如果電阻越大，LED 就越暗

**要接哪一個腳位？**

**目標：一隻腳接地，一隻腳給電**

# Model B vs. B+/Pi 2



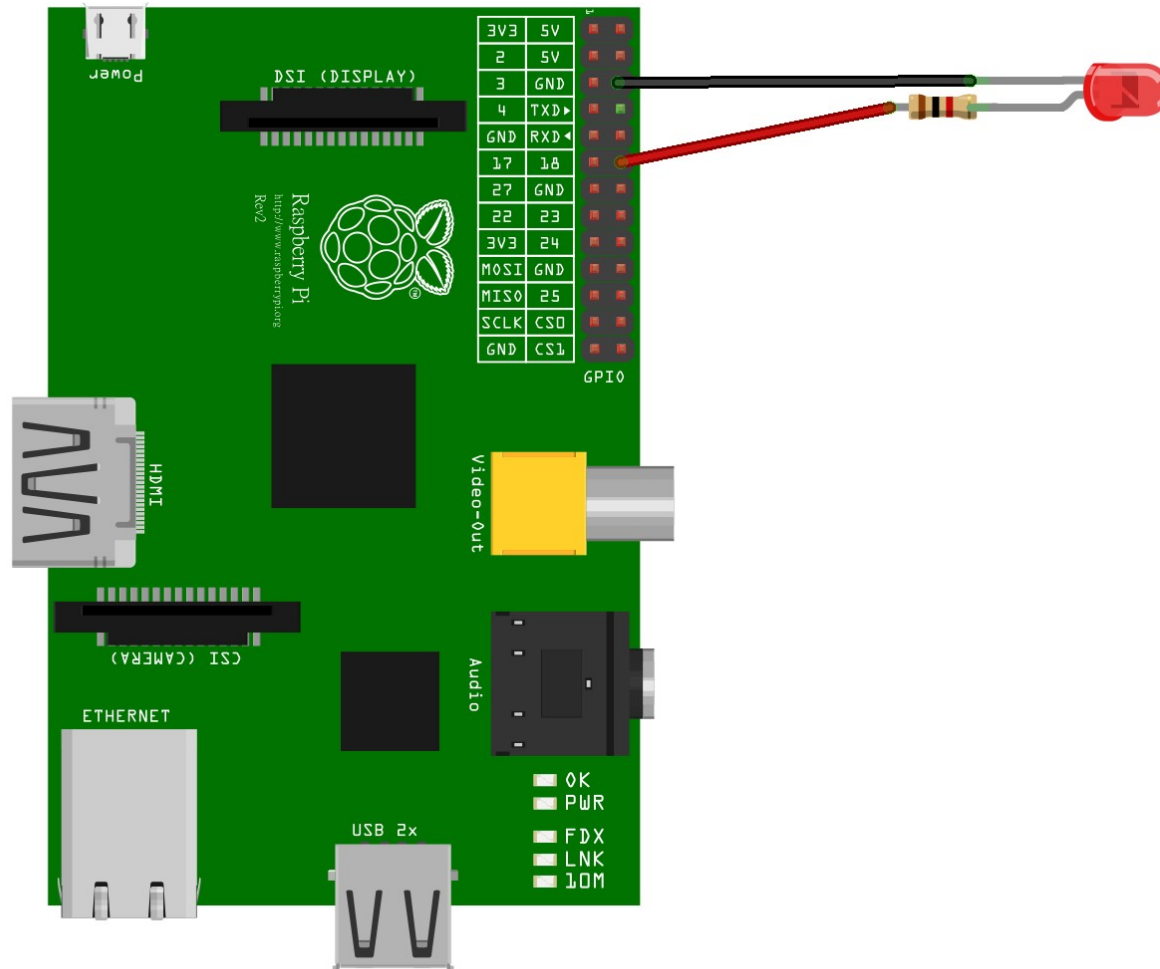
[http://elinux.org/RPi\\_Low-level\\_peripherals](http://elinux.org/RPi_Low-level_peripherals)

<http://www.raspberrypi-spy.co.uk/2014/07/raspberrypi-b-gpio-header-details-and-pinout/>

# 線路圖

**LED**  
長腳 (RED)  
短腳 (BLACK)

**RPi**  
Pin12 (GPI01)  
Pin6 (Ground)



開始用 Python 控制 GPIO 吧

# Python Code 基本流程

- 載入模組 (Import module)
- 選擇編號系統 (Define pin numbering)
- 定義腳位 (Setup up a channel)
- 讀取輸入 / 寫入輸出 (Input/Output)
- 清理 (Cleanup)



# 一個實際的範例

```
#!/usr/bin/python
```

- `import RPi.GPIO as GPIO` # 載入模組
- `import time`
- `GPIO.setmode(GPIO.BOARD)` # 選擇系統 (以實體腳位系統為例)
- `LED_PIN = 12`
- `GPIO.setup(LED_PIN, GPIO.OUT)` # 定義腳位
- `print("LED is on")`
- `GPIO.output(LED_PIN, GPIO.HIGH)` # 設定腳位狀態
- `time.sleep(3)`
- `GPIO.cleanup()` # 清理

# 腳位對照表

P1: The Main GPIO connector							
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin
		3.3v	1	2	5v		
8	Rv1:0 - Rv2:2	SDA	3	4	5v		
9	Rv1:1 - Rv2:3	SCL	5	6	0v		
7	4	GPIO7	7	8	TxD	14	15
		0v	9	10	RxD	15	16
0	17	GPIO0	11	12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13	14	0v		
3	22	GPIO3	15	16	GPIO4	23	4
		3.3v	17	18	GPIO5	24	5
12	10	MOSI	19	20	0v		
13	9	MISO	21	22	GPIO6	25	6
14	11	SCLK	23	24	CE0	8	10
		0v	25	26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin

## 2：讀取按鍵輸入

目的：了解硬體和軟體的差異

# 按鍵

- 按鍵 Button / 開關 Switch
- 按鍵式，滑動式，傾斜式 ...



# 線路圖 ( 上拉電阻 )

**BUTTON**

腳 1

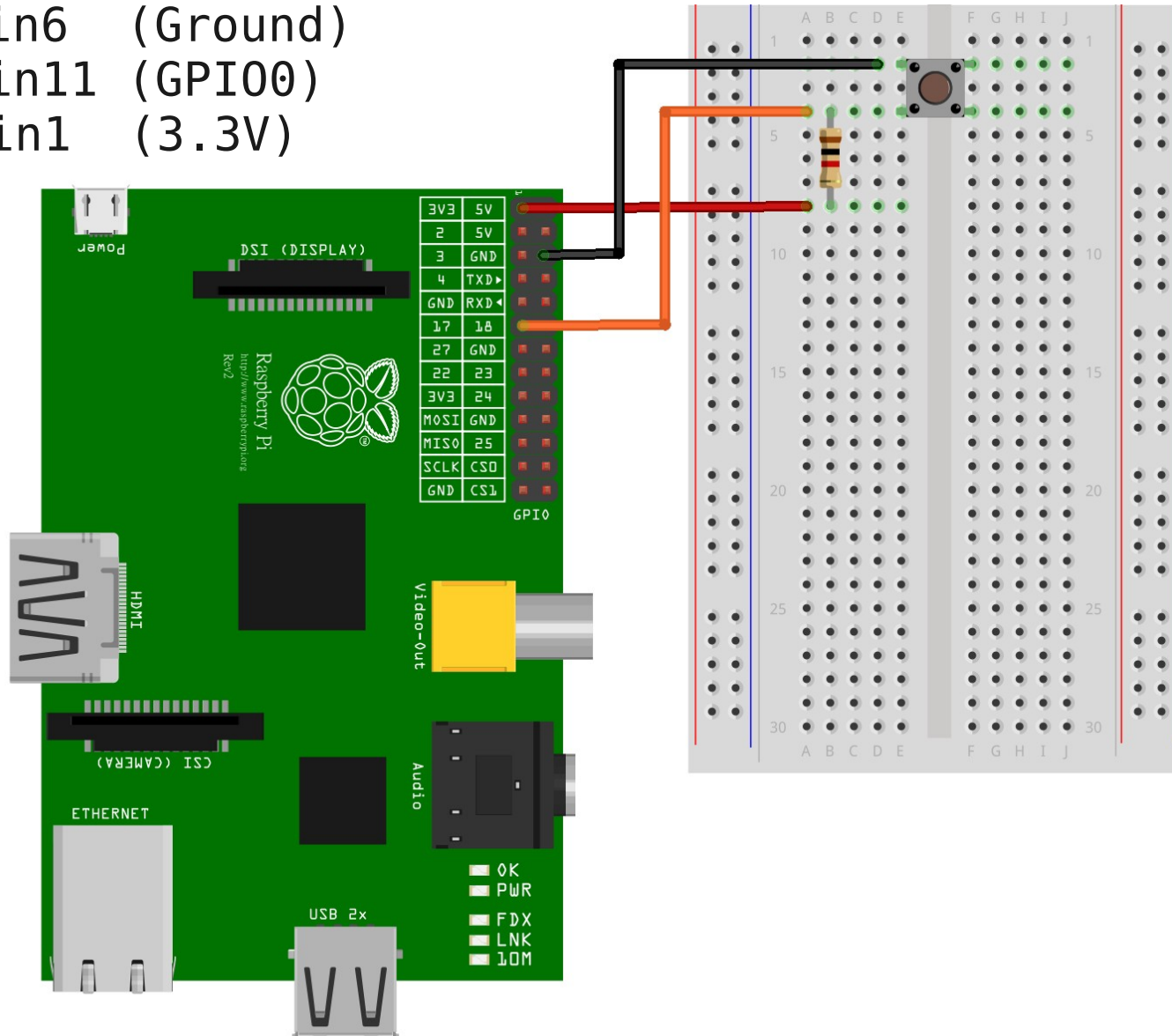
腳 3

**RPi**

Pin6 (Ground)

Pin11 (GPIO0)

Pin1 (3.3V)



# 實現無敵模式 / 程式碼

- 無敵模式 = 邊走邊破壞 = 放置空氣方塊 (air block)

```
GPIO.setup(11, GPIO.IN)
```

```
def cb():
```

```
    print("Button.Click")
```

```
    # place air block
```

```
try:
```

```
    GPIO.add_event_detect(11, GPIO.FALLING, callback=cb)
```

```
while True:
```

```
    time.sleep(10)
```

```
finally:
```

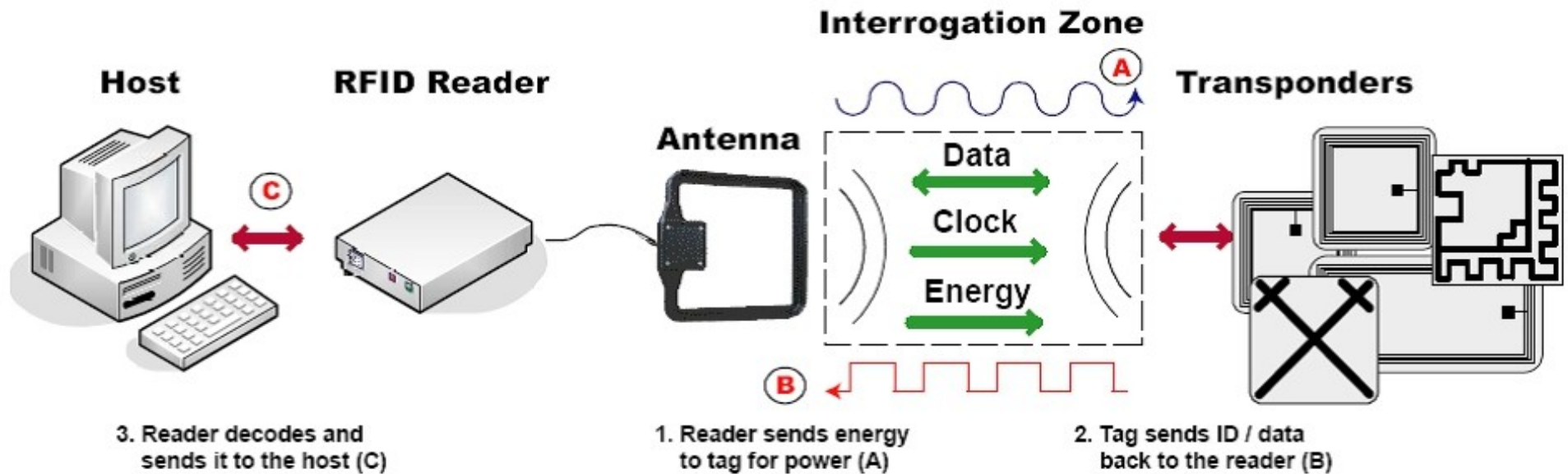
```
    GPIO.cleanup()
```

## 3 : RFID

目的：常用的無線通訊模組

# RFID 使用情境

- 無線射頻辨識 (Radio Frequency Identification)

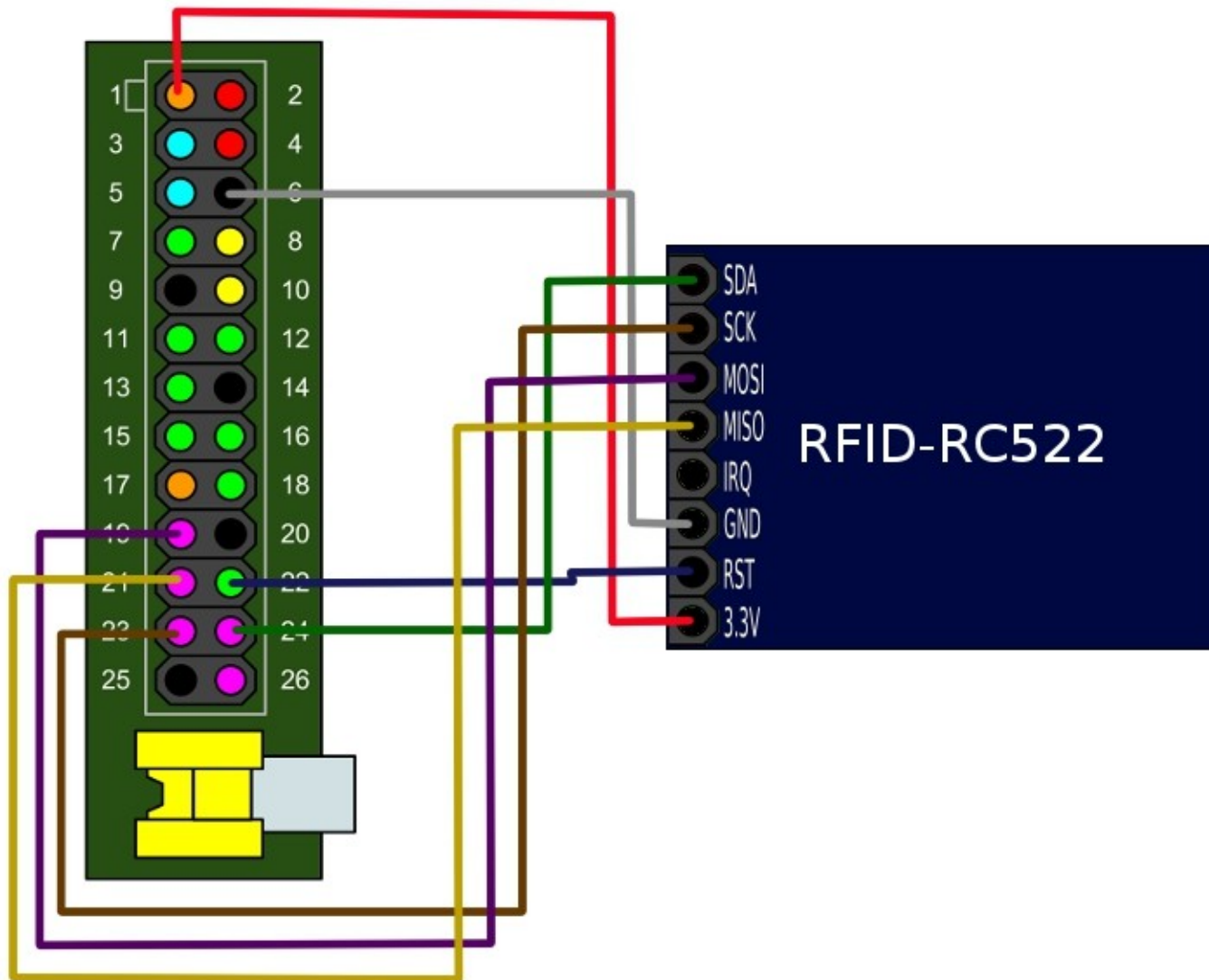




# RFID 使用頻段

RFID Frequency Band	Scan Distance
120-150 kHz (Low Frequency, LF)	Up to 10 cm
13.56 MHz (High Frequency, HF)	Up to 1 m
433 MHz (Ultra High Frequency, UHF)	1-100 m
865-868 MHz & 902-928 MHz (Ultralight High Frequency, UHF)	1-2 m
2450-5800 MHz (Microwave)	1-2 m
3.1-10 GHz (Microwave)	Up to 200 m

# RC522 接線圖



# 安裝所需套件與函式庫

- `$ sudo apt-get install python-dev python-pip`
- `$ git clone https://github.com/lthiery/SPI-Py`
- `$ cd SPI-Py`
- `$ sudo python setup.py install`

# 下載 Reader 程式 & 測試

- \$ git clone <https://github.com/mxgxw/MFRC522-python>
- \$ cd MFRC522-python
- \$ sudo python Read.py

```
Card detected
```

```
Card read UID: 45,112,232,124,201
```

```
Size: 8
```

```
AUTH ERROR!!
```

```
AUTH ERROR(status2reg & 0x08) != 0
```

```
AUTH ERROR
```

# 根據 RFID 感應放置 Block / 程式碼

```
#DIRT      = Block(3)
#SAPLING   = Block(6)
block_dict = {"67 c0 7e b5":46}
mc = minecraft.Minecraft.create()
try:
    for index, uid in enumerate(read_rfid()):
        p = mc.player.getTilePos()

        if block_dict[uid] == 3:
            mc.setBlock(p.x+1, p.y, p.z+1, block_dict[uid], 1)
except Exception as e:
    print e
```

DEMO

# Resources

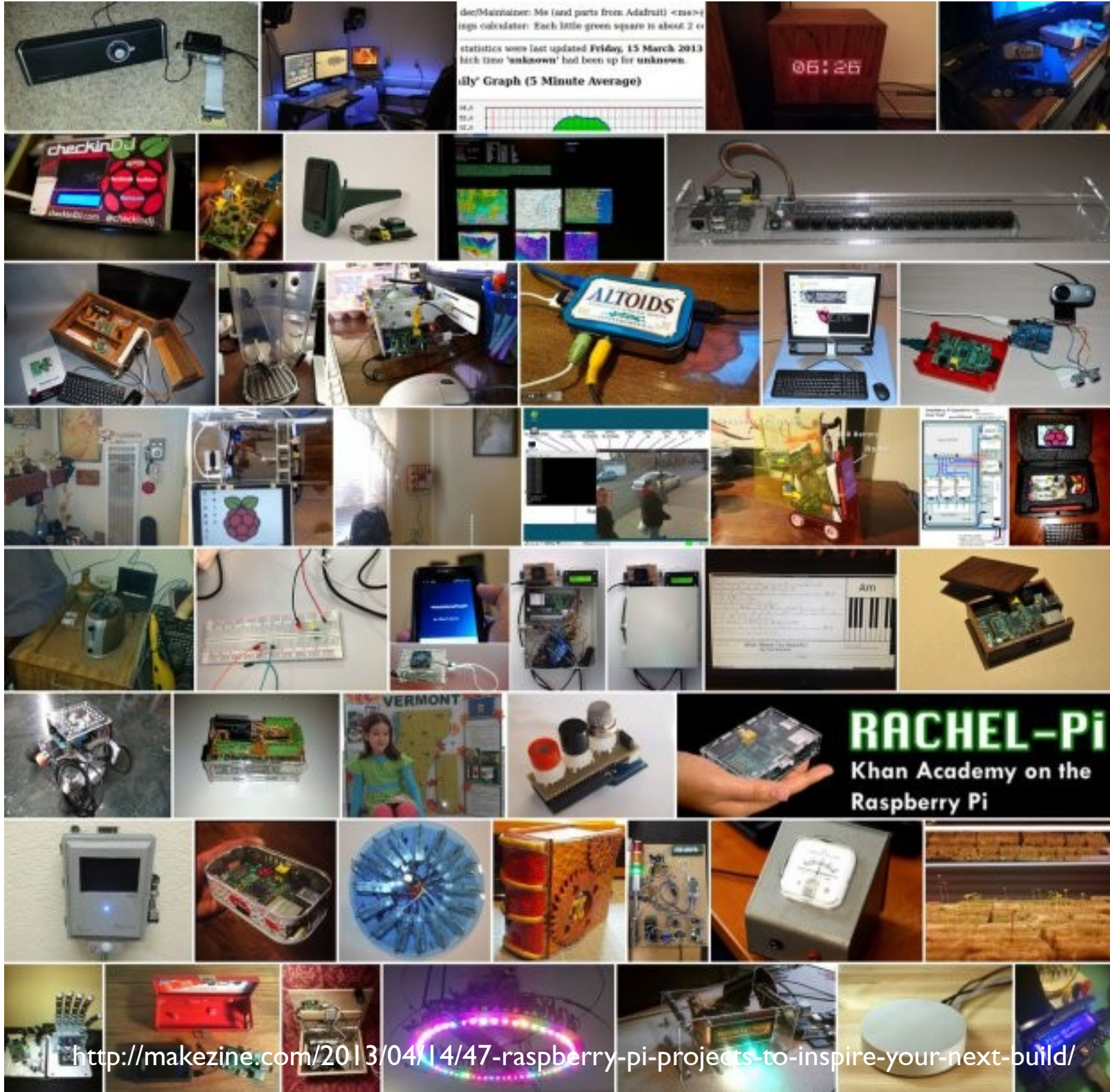
- Minecraft Wiki
  - [http://minecraft-zh.gamepedia.com/Minecraft\\_Wiki](http://minecraft-zh.gamepedia.com/Minecraft_Wiki)
- Adventures in Minecraft
  - <http://as.wiley.com/WileyCDA/Section/id-823690.html>
- Minecraft: Pi Edition - API Tutorial
  - <http://www.stuffaboutcode.com/2013/04/minecraft-pi-edition-api-tutorial.html>
- GETTING STARTED WITH MINECRAFT PI
  - <https://www.raspberrypi.org/learning/getting-started-with-minecraft-pi/worksheet/>

# LEGO Minecraft



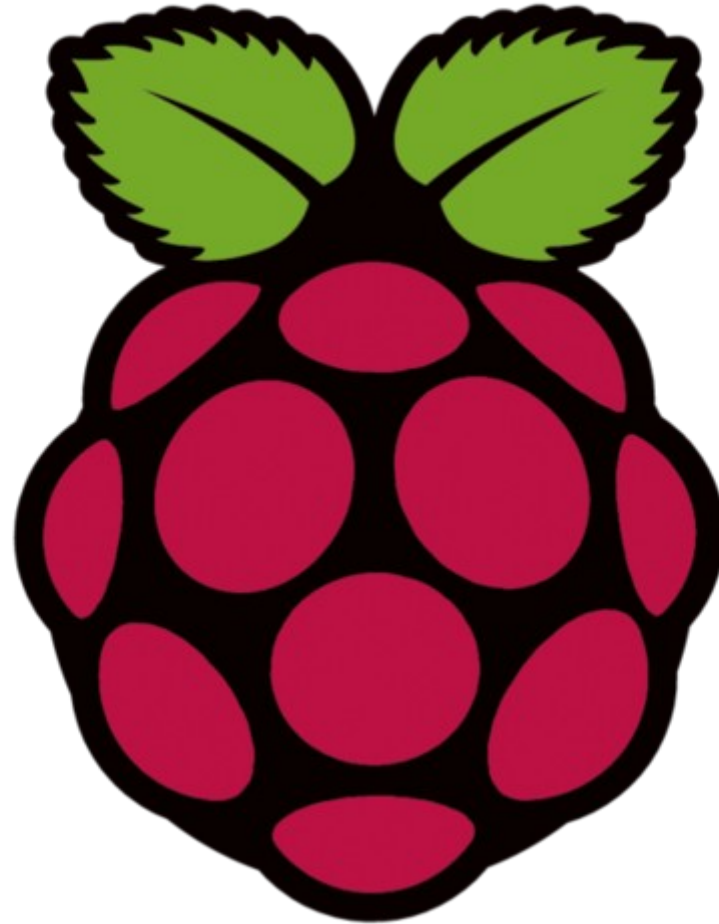


**更多應用**



<http://makezine.com/2013/04/14/47-raspberry-pi-projects-to-inspire-your-next-build/>

# Raspberry Pi Rocks the World



Thanks