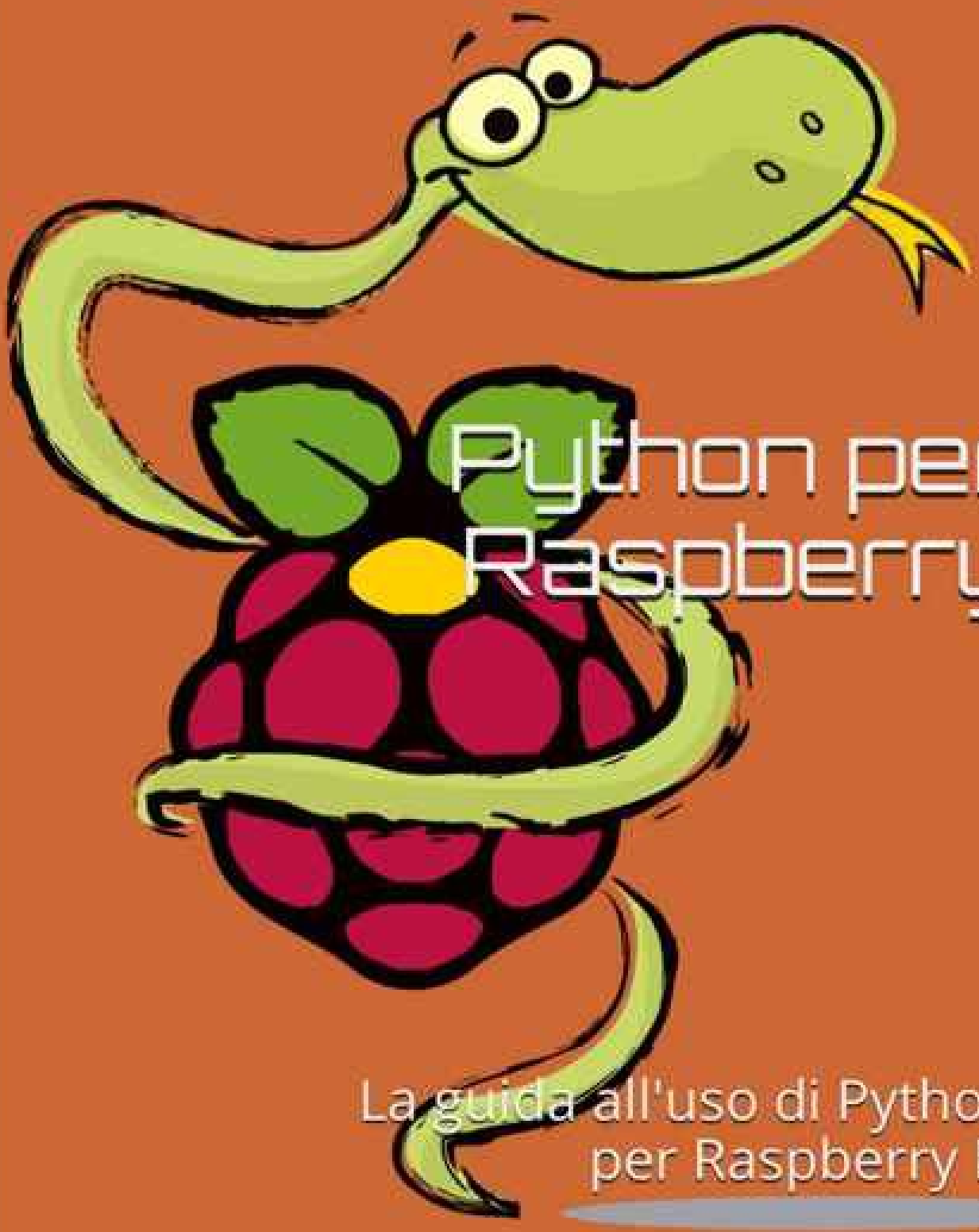
Giovanni Gentile



Python per Raspberry

La guida all'uso di Python per Raspberry Pi

Python per Raspberry Pi

Giovanni Gentile

Ad Ilaria,
che mi sostiene in tutti i momenti.

Prefazione

L'idea di scrivere un libro di questo tipo deriva dalla mia volontà di spiegare in parole semplici, concetti molto complessi ed articolati. Dopo l'acquisto del mio primo Raspberry Pi, infatti ho cercato in rete una guida che fosse espressamente dedicata alla programmazione del Raspberry Pi attraverso il linguaggio Python. In seguito notai il successo di pubblico che riscontravo con i workshop su Raspberry Pi e Python. Decisi dunque di scrivere una piccola pubblicazione che potesse servire da libro di testo per chi seguiva i miei corsi. Da qui l'idea di arricchire quegli appunti con immagini e concetti più completi. Pagina dopo pagina prese vita il piccolo manuale che avete tra le mani. Questa non vuole essere una pubblicazione scientifica di alto profilo, e nemmeno sostituirsi a tutti gli eminenti manuali scritti sull'utilizzo di Python o del Raspberry Pi. Credo che questa guida vada a colmare un vuoto concettuale, o meglio costruisca un ponte di connessione tra una piattaforma di sviluppo ed apprendimento come il Raspberry, ed un linguaggio di programmazione ad alto livello come Python. Non avendo una formazione specifica in questo campo, le mie nozioni esulano dall'essere definitive e complete. La mia intenzione è quella di fornire materiale al maker, all'hobbista all'appassionato di elettronica. Fornire dei suggerimenti dai quali partire per cominciare un lungo viaggio. Gli argomenti subiranno una trattazione superficiale. Di converso, tale pubblicazione sarebbe risultata molto tecnica, ed avrebbe allontanato il semplice appassionato o il curioso. Spero che la mia intenzione sia compresa e ben accolta. Sarei contento se mi fornisse suggerimenti e pareri in merito agli argomenti trattati anche attraverso il sito web www.raspberrypython.com.

Il sito di supporto raspberrypython.com

Questo progetto editoriale nasce con l'intento di fornire gli strumenti di base atti a gestire la piattaforma Raspberry Pi tramite il linguaggio Python. Ho ritenuto utile sviluppare una piattaforma online che fungesse da ponte di connessione tra me ed i lettori del libro. Sul sito www.raspberrypython.com troverete una sezione con in miei riferimenti online, una relativa al libro, con i codici e gli esempi trattati, ed una relativa alla parte hardware. Ho voluto inserire i codici trattati nei tutorial per permettere ai lettori di copiare ed incollare comodamente il codice senza riscriverlo tutte le volte, anche se questo rappresenta un buon esercizio per imparare i comandi fondamentali di Python. Essendo questo un campo in continuo sviluppo, potrete trovare aggiornamenti o migliorie nei codici che sono presentati all'interno del libro. In questo campo è difficile stare al passo con i tempi, in questi giorni, mentre sto scrivendo queste pagine, ho dovuto modificare il capitolo relativo all'hardware, inserendo la nuova versione Raspberry Pi B+. Sul sito oltre alla sezione software, troverete anche quella relativa all'hardware. Una selezione di prodotti in vendita sul famoso sito amazon.it. Ho voluto inserire questa vetrina virtuale poiché erano molte le persone che mi chiedevano dove acquistare i prodotti che utilizzavo nei miei workshop, e quali erano le caratteristiche da tenere presente al momento dell'acquisto. Vi invito a visitare il sito, a lasciare qualche commento. Vi ringrazio sin da subito per eventuali critiche. Queste aiutano a crescere e a migliorarsi. Buona lettura.

Raccomandazioni

Quanto segue in questa pubblicazione rappresenta miei appunti personali. Il lettore che intende utilizzare tali miei appunti è avvertito di verificarne l'esattezza e l'applicabilità. Con questo avviso mi scarico da qualsiasi responsabilità su eventuali danni e problemi possano derivare dalla lettura dei suddetti appunti così come dall'eventuale realizzazione, messa in opera, applicazione anche solo parziale di quanto descritto ritenendosi, questa, da ascrivere alla diretta responsabilità di chi le pone in atto. Tale avviso mi esonera da qualsiasi responsabilità di eventuali pretese che potessero essere avanzate a qualsivoglia titolo per eventuali incidenti e conseguenti danni diretti o indiretti che ne derivassero ai componenti utilizzati, alle strutture, a sé stessi, a terze persone, a beni e mezzi di terzi, siano essi anche solo spettatori o chi altro si voglia. Nessuna responsabilità viene assunta in relazione al contenuto di quanto pubblicato sul sito internet www.raspberrypython.com ed all'uso che terzi ne potranno fare, Nessuna responsabilità per eventuali contaminazioni derivanti dall'accesso, dall'interconnessione, dallo scarico di materiale e programmi informatici da questo o altri siti ad esso collegati.

Il Raspberry Pi

Lampone cosa?…

Il Raspberry Pi è un computer che viene definito *single board*. Un single board computer è un computer che alloggia su una singola scheda elettronica. L'idea trainante questo progetto, è quella di fornire un calcolatore economico, concepito per stimolare l'insegnamento dell'informatica e della programmazione, anche nelle scuole. Questa piccola scheda delle dimensioni di qualche centimetro include una piastra madre, un processore, una scheda video con uscita HDMI, e tutto quello che serve ad un comune calcolatore per funzionare. Il Raspberry Pi è stato sviluppato nel Regno Unito, ed attualmente viene assemblato in Galles. La Raspberry Pi Foundation ha in catalogo tre versioni del suddetto calcolatore:

Model A+
Il più economico è dotato di 256 megabyte di RAM: costa 20 USD, ha una singola porta USB ed è privo di controller Ethernet.

Model B e B+
La versione più evoluta del Raspberry Pi. E' equipaggiato con due porte USB ed un controller Ethernet 10/100 e costa 35 Dollari. A partire del 15 ottobre 2012 il Model B monta 512 Megabyte di RAM. Recentemente è stato messo in vendita un update concettuale del Model B chiamato Model B+, dotato di 512 Megabyte di RAM, 4 porte USB, ed un sistema di alimentazione ripensato da zero. Anche il Model B+ costa 35 dollari. Sebbene i Modelli A e A+ non abbiano una porta Ethernet RJ45, si può comunque accedere a una rete attraverso la porta USB, facendo uso di adattatori Ethernet o Wi-Fi con alimentazione autonoma. Raspberry Pi è compatibile con tastiere e mouse generici collegabili tramite porta USB. Come avrete avuto modo di notare al Raspberry Pi manca una parte molto importante. L'Hard Disk. Il progetto non prevede né hard disk né una unità a stato solido integrata. La memoria ed il boot sono affidati ad una scheda SD o mini SD.

Curiosità

Le prime 10 schede furono messe all'asta su eBay nelle prime settimane del 2012. Una è stata comprata da un anonimo e donata al museo inglese The Centre for Computing History, ubicato nel Suffolk.

Software

Il Raspberry Pi, essendo basato su un processore ARM, utilizza Linux come sistema operativo. La scelta più rapida per fornire subito una versione di Linux pronta all'uso e perfettamente adattata alla macchina che andremo ad usare è NOOBS. NOOBS è un pacchetto di installazione che provvede da solo a creare una partizione auto partente, ed avviare una installazione di Raspbian, una versione del sistema operativo targato Linux, creata su misura per Raspberry Pi.

Hardware

Processore grafico e main processor

Il cuore hardware del Raspberry Pi è composto essenzialmente da un processore grafico dalle grandissime capacità il BCM2835. Questo è un processore creato e commercializzato dalla Broadcom (www.broadcom.com). Esso è utilizzato su apparecchi mobili, ed offre alti livelli di performance. Essendo un processore nato per il mobile, il suo progetto parte da alcuni postulati di base. Il basso consumo energetico è uno di questi. Esso integra VideoCore IV, una tecnologia che orientata a favorire il media playback, la gestioni delle immagini, webcam, streaming media, grafica e 3D gaming. Il main processor del Raspberry Pi è il ARM 1176 della ARM (http://www.arm.com/). Anch'esso è un processore studiato per il mobile. Viene utilizzato negli smartphone di ultima generazione, nei tablet, e nelle ormai sempre più diffuse "Smart TV". La Raspberry Pi Foundation ha deciso di condividere tutti gli schemi progettuali. Chiunque può collegarsi al sito ufficiale, scaricare gli schemi ed assemblare la scheda. Tutti gli schemi di collegamento e le interfacce hardware sono disponibili in rete all'indirizzo: http://www.raspberrypi.org/documentation/hardware/raspberrypi/. Questo è vero Open Source!

Alimentazione

L'alimentazione del Raspberry Pi non richiede costosi e potenti alimentatori. In teoria basterebbe quello che utilizzate per ricaricare il vostro cellulare. Tuttavia è preferibile optare per un alimentatore a 5 Volts, che eroghi almeno 700 mA di corrente. Sul sito ufficiale dichiarano che il modello B consuma tra i 700 mA ed i 1000 mA. Tuttavia il Raspberry Pi può gestire al massimo 1000 mA per l'alimentazione delle periferiche esterne. Il mio consiglio comunque è quello di dotarvi di un HUB USB alimentato. In commercio se ne trovano di diversi tipi, dimensioni e costi. Questa pratica è utilissima poiché non sforzerete il regolatore di tensione del piccoletto. Soprattutto se avete intenzione di utilizzare periferiche come Hard Disk esterni o soprattutto schedine wifi USB. Il consumo di queste ultime infatti è da non sottovalutare. Il rischio potrebbe essere quello di compromettere il regolatore di tensione o peggio il processore del Raspberry Pi.

USB ed Ethernet

Il modello B del Raspberry PI è equipaggiato con due porte USB 2.0 ed una Ethernet. Quello B+ con quattro porte USB 2.0 ed il modello A con una sola porta USB collegata direttamente al processore BCM2835. Le porte USB utilizzate sul Raspberry Pi sono del tipo OTG (On The Go): Questo tipo di porta è quella utilizzata su smartphone e tablet. Qualsiasi periferica supportata da Linux viene riconosciuta direttamente anche dal Raspberry Pi. Io personalmente non ho mai trovato difficoltà nel gestire tastiera, mouse, penne USB, Hard Disk o altro. Questo perché Linux ha un enorme database per i driver delle periferiche che fa invidia agli altri sistemi operativi. La velocità massima di funzionamento per le periferiche usb connesse al Raspberry Pi è di 480 mbps. Questo non è un problema, ma bisogna tener presente che alcune periferiche sono ad alta velocità, come per esempio le webcam, altre a media come le schede audio usb, ed altre a bassa velocità come mouse e tastiera.
In generale il sistema gestisce molto bene anche periferiche a diverse velocità. Come detto sopra, ciascuna porta USB gestisce fino a 1000 mA. Questo significa che periferiche avide di corrente vanno alimentate con un hub a parte. Se colleghiamo una periferica usb ad alto assorbimento, il Raspberry Pi va in blocco e riavvia il sistema.

TIPS gestire le periferiche USB

Per gestire le periferiche USB consiglio due comandi interessanti da inviare come sempre tramite il terminale di Linux. Il primo di questi è:

```
lsusb
```

Questo comando stampa a video il nome di tutte le periferiche riconosciute o meno, collegate al Raspberry Pi.

Il secondo comando è:

```
dmesg | tail
```

Questo comando stampa a video gli ultimi messaggi del kernel. Ovvero le ultime operazioni effettuate. Cercando tra le righe, troverete le vostre

periferiche. Potrete capire se il sistema ha riconosciuto l'inserimento della periferica USB o se il problema, prima di essere software è hardware, o se non esistono nel sistema i driver necessari a gestire la periferica collegata.

GPIO

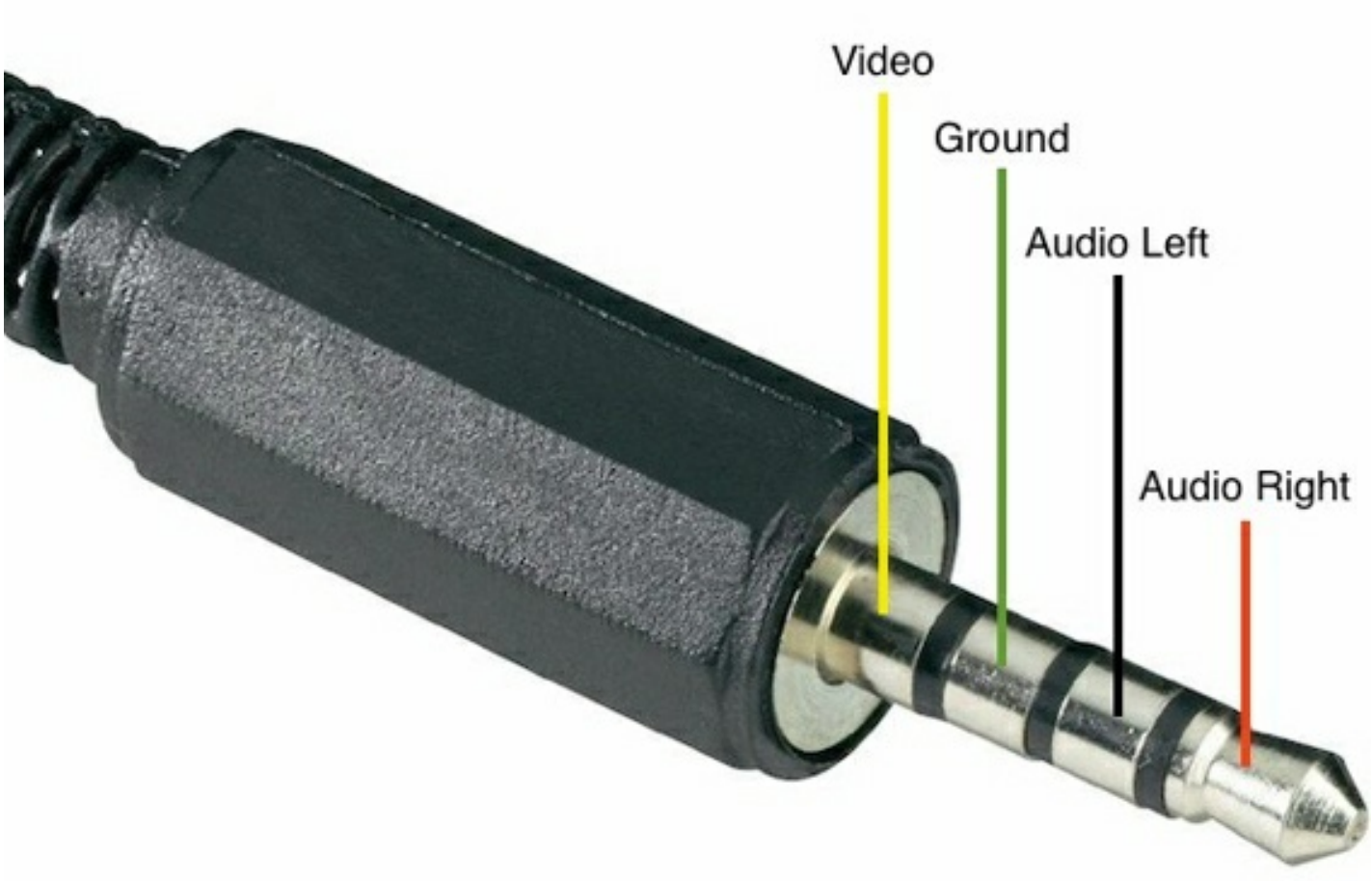
La parte più interessante del Raspberry Pi è proprio la possibilità che viene data all'utilizzatore di sfruttare gli Input e gli Output della scheda. Ovvero la possibilità di pilotare e controllare ingressi ed uscite del computer attraverso un software scritto dall'utente. Il linguaggio che parla il Raspberry è proprio Python. La porta GPIO è quella spina situata nell'angolo della scheda a forma di pettine. Questa possiede ventisei pin nella versione B, e 40 in quella B+, che possono essere programmati come pin di ingresso, per la lettura di sensori, come pin di uscita, per comandare led, luci , motori eccetera. La porta GPIO può essere utilizzata per dialogare con altre periferiche con i protocolli UART, I2C, SPI.

Gli accessori hardware necessari

L'hardware necessario per l'avvio e l'uso del Raspberry Pi comprende, oltre naturalmente alla scheda in questione:

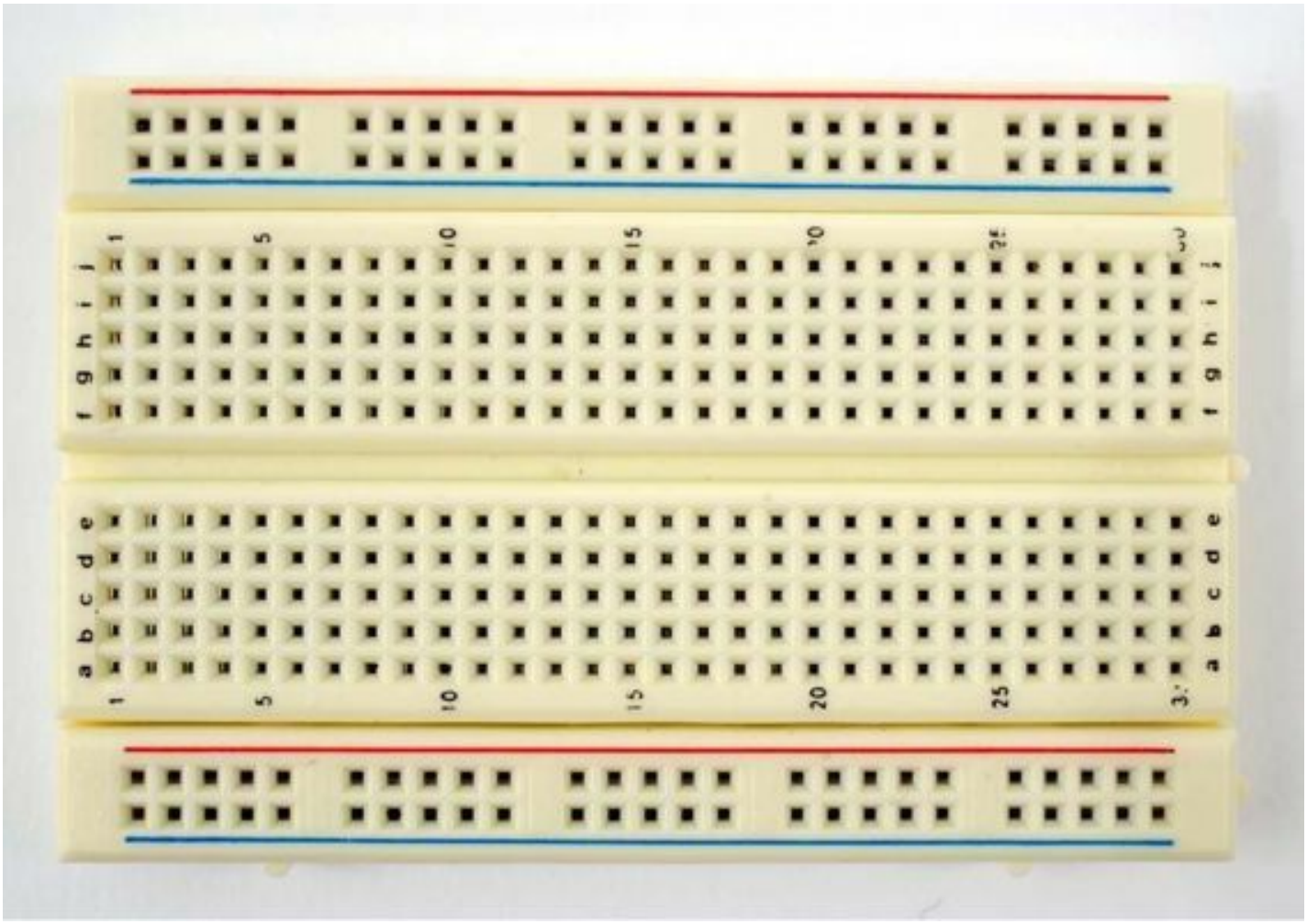
- Un alimentatore con spina MicroUSB (vale quanto detto sopra). Utilizzate un alimentatore per cellulari se non avete specifiche necessità. Se avete intenzione di collegare diverse periferiche, allora consiglio un alimentatore più performante, 1000/1200 mA potrebbero bastare.
- Una scheda SD o micro SD se possediamo la versione B+ da almeno 2 GB. Questa sarà l'hard Disk del nostro sistema. Il mio consiglio è quello di scegliere una scheda che sia abbastanza veloce da non fungere da collo di bottiglia rispetto al processore centrale. Schede con un coefficiente di velocità 6x o ancora meglio 10x sono più che sufficienti. Scegliamo sempre schede SDHC. Ricordatevi che questo rappresenta il disco rigido del sistema. E' possibile acquistare diverse schede SD, in maniera tale da avere diversi sistemi operativi funzionanti all'occorrenza. Con un solo Raspberry Pi, possiamo utilizzare diverse distribuzioni di Linux. Interessante menzione va fatta per la disto RASPBMC per Raspberry Pi. Una distribuzione "da salotto" che trasforma il nostro Raspberry Pi, in un media center di alto profilo (http://www.raspbmc.com).
- Un computer con lettore di schede SD. Questo è essenziale per installare il sistema operativo sulla scheda SD e per eseguire i comandi da Terminale. Di seguito, spiegheremo come utilizzare il Terminale ssh. A seconda del sistema operativo installato sulla macchina, sarà necessario eseguire diverse operazioni per effettuare l'installazione e la prima connessione al Raspberry Pi.
- Un cavo HDMI. Il cavo è necessario se desiderate collegare il Raspberry al televisore o al monitor di casa. Nella versione B, è presente anche un cavo coassiale RCA video. Questo può essere collegato anche ai vecchi schermi sprovvisti di prese HDMI o VGA. Se il vostro televisore è uno di quelli vecchia scuola, e possiede una spina SCART, sarà necessario dotarsi di un adattatore da SCART a RCA. Nella versione del Raspberry Pi B+, i costruttori hanno preferito eliminare il connettore RCA, sostituendolo con un più comodo jack a quattro poli. Il jack collega l'audio destro, il sinistro e

il video analogico.



In questo caso, sarà necessario dotarsi di un cavo jack-RCA per collegare il Raspberry Pi al vecchio televisore di casa. Se invece preferite collegare la piattaforma al vostro monitor sprovvisto di presa HDMI, allora consiglio l'utilizzo di un adattatore HDMI-VGA. Ponete attenzione nella scelta di questo adattatore. In commercio ne esistono diversi che non sono compatibili con il Raspberry Pi.

- Un monitor/TV con ingresso HDMI o composito.
- Una tastiera USB, se preferite anche bluetooth.
- Un cavo Ethernet meglio se lineare.
- Una breadboard
- Dei cavi di collegamento



La Breadboard

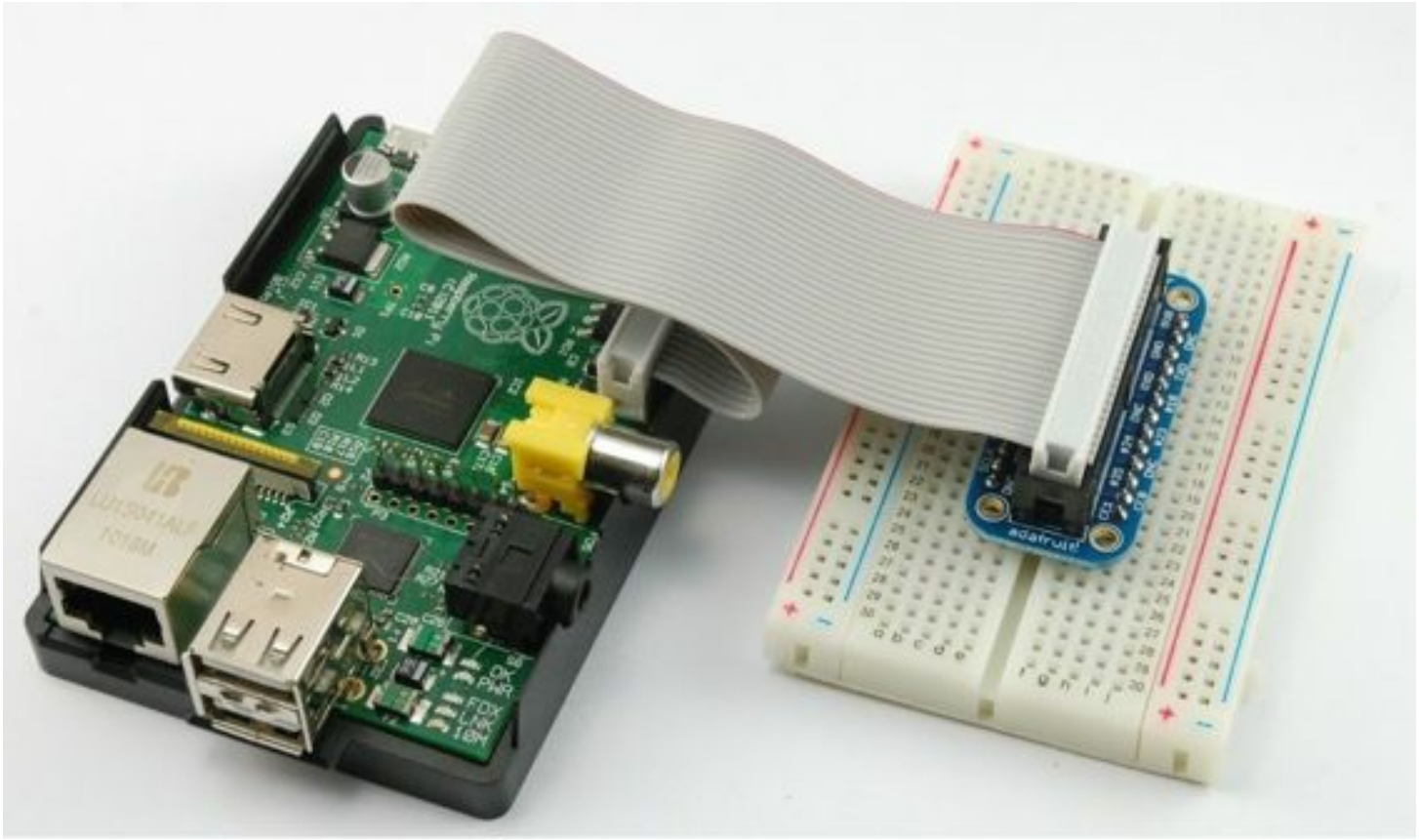
La breadboard prende il suo nome dal vecchio metodo di testare i circuiti elettronici. Molto tempo fa infatti, le prime prove dei circuiti venivano eseguite proprio sui taglieri per il pane. I circuiti venivano testati collegando i componenti a dei chiodi che erano fissati nel legno dei taglieri. La breadboard oggi rappresenta il metodo più economico e veloce per testare qualsiasi tipo di circuito elettrico. A questo punto, è molto importante capire come sono connessi i vari fori presenti sulla scheda. Come potete vedere dall'immagine, i fori presenti sulla tavoletta di plastica sono tutti collegati tra loro, secondo lo schema rappresentato nell'immagine in basso. Le due file esterne della scheda sono collegate per tutta la lunghezza. Esse sono contrassegnate da due colori differenti, il rosso ed il blu. I colori rappresentano le polarità della fonte di alimentazione. Dunque le strisce di punti laterali, quella destra e quella di sinistra, sono dedicate all'alimentazione. Quelle centrali, invece, sono collegate in larghezza. Come notate le prime cinque colonne sono connesse, poi è presente uno spazio, che chiude le connessioni, e poi ricominciano i collegamenti. La linea centrale serve per permettere di utilizzare dei circuiti integrati all'interno del

progetto. Infatti questi alloggeranno al centro della piastra permettendo il montaggio dei componenti sugli spazi laterali.

Come collegare il Raspberry Pi alla Breadboard

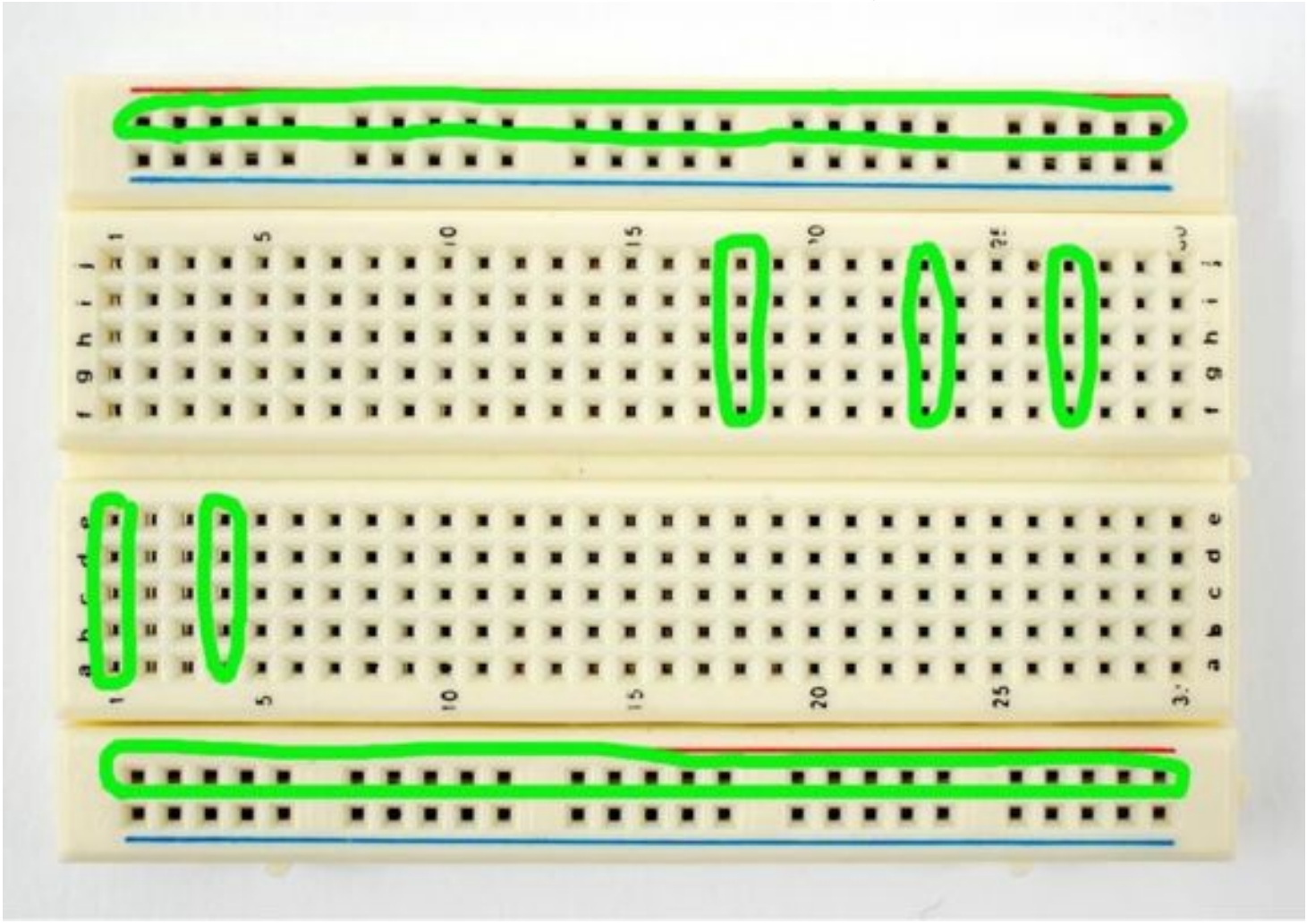
Per effettuare il collegamento della porta GPIO del nostro Raspberry Pi alla board di sperimentazione sopra descritta è possibile utilizzare due metodi. Il primo consiste nell'acquistare i cavi di collegamento rappresentati nell'immagine in basso. Essi hanno un polo "maschio" ed uno "femmina". Un polo del cavo viene collegato al Raspberry Pi, e l'altro viene inserito nella scheda breadboard. Questo rappresenta il medito più semplice ed economico per utilizzare la porta GPIO del Raspberry Pi.

L'altro metodo, invece, è quello di acquistare un'accessorio dedicato. Quello che vedete in foto è commercializzato dall'azienda Adafruit (www.adafruit.com).



Esso è una valida alternativa all'uso diretto dei cavi. Il funzionamento è molto semplice. Sposta l'intera spina GPIO sulla breadboard. In questo caso è possibile utilizzare del comunissimo cavo elettrico per collegare i componenti elettronici ad i rispettivi pin della porta GPIO.

Cominciamo ad usare il nostro Raspberry Pi



Installiamo il sistema operativo:

1. **Download**

Come già detto il sistema operativo che useremo sarà Raspbian. L'installazione del sistema avverrà attraverso l'uso di NOOBS. Navighiamo fino all'indirizzo https://www.raspberrypi.org/downloads/ e scaricate la versione NOOBS completa. Evitate di scaricare quella Lite.

2. **Scompattiamo il pacchetto**

Una volta che avete recuperato il pacchetto NOOBS, arriva il momento di scompattarlo, e salvare il suo contenuto in una cartella vuota sul nostro computer.

3. Formattazione della scheda SD

Visitate il sito https://www.sdcard.org/downloads/formatter_4/ e scaricate

l'ultima versione del programma SD Formatter per il sistema operativo che state utilizzando. Al termine dell'operazione non vi resta altro che avviare il programma SD Formatter e formattare la scheda di memoria SD destinata ad ospitare NOOBS.

4. Copia dei file

Una volta che l'operazione di formattazione è terminata, copiate tutto il contenuto della cartella creata in precedenza contenente il contenuto del file NOOBS zip nella scheda SD prescelta.

5. Primo avvio di Raspberry

Terminata la copia, non vi resta altro che espellere la scheda SD, inserirla nel Raspberry Pi, collegare mouse, tastiera, video ed infine inserire la micro USB di alimentazione.

6. Scegliere l'installazione di Raspbian

Una volta che il sistema si avvia, se abbiamo effettuato tutte le operazioni correttamente, dovremmo poter scegliere il tipo di installazione da effettuare. Scegliere Raspbian ed attendere il processo di installazione.

7. Raspbian menu

Una volta terminata l'installazione, è giunto il momento di impostare data e ora. Inoltre è buona norma estendere la capacità della memoria selezionando la prima opzione del menù. Altra operazione essenziale è quella di abilitare il server SSH. Cicchiamo sulla voce "Advance Option", poi sulla voce "Enable or disabile ssh server". Attraverso questo protocollo potremmo collegarci direttamente al nostro Raspberry Pi usando il protocollo ssh. Con il tasto TAB sulla tastiera spostatevi sulla scritta "Finish" terminando così l'installazione del sistema. Per richiamare questo menù in qualsiasi momento sarà necessario digitare da terminale il comando:

```
sudo raspi-config
```

8. Avviare l'interfaccia grafica

Quando si installa il sistema operativo Raspbian, questo utilizza l'interfaccia da linea di comando come default. A questo punto però serve capire se l'installazione è andata a buon fine. Per verificare la corretta installazione, riavviamo il sistema e una volta che il caricamento è terminato, digitate:

startx

IMPORTANTE

La username di default per il sistema operativo Linux Raspbian utilizzato è "pi." La password è "raspberry".

Dove reperire il materiale Hardware

Uno dei canali che io utilizzo per l'acquisto del materiale hardware è internet. Due dei siti internet che trattano questo tipo di accessori sono eBay ed Amazon. Come ho già spiegato precedentemente, questo progetto editoriale ha un sito internet di riferimento (www.pythonraspberry.com) che ha la funzione di raccogliere i commenti dei lettori, fornire assistenza, e racchiudere i codici dei tutorial. Ho pensato di inserire all'interno del sito una sezione dedicata all'hardware. Una vetrina virtuale che rimanda ai prodotti che ho utilizzato per la stesura dei tutorial.

Avvio del Raspberry in SSH

Il protocollo SSH ci permette di collegarci al Raspberry Pi attraverso un terminale, ed impartire gli ordini alla macchina senza utilizzare periferiche o video. In tutti i tutorial di questo libro ricorreremo al protocollo SSH per programmare ed avviare il Raspberry Pi. E' di fondamentale importanza non saltare questi primi passaggi. In seguito utilizzeremo molto spesso il protocollo ssh per programmare ed avviare il Raspberry. Prima di riavviare il sistema, possiamo scollegare tutte le periferiche dal Raspberry Pi, e collegare il cavo di rete Ethernet tra il Raspberry Pi ed il router di casa.

L'indirizzo IP del Raspberry

Quando avviamo il Raspberry Pi collegato al router, il sistema gli assegna un indirizzo IP. Un indirizzo IP è una serie di numeri divisi in quattro triplette del tipo *192.168.0.200*. Esso è il nome del computer in rete. E' molto importante conoscere l'indirizzo IP della macchina per poter effettuare i collegamenti necessari ad interagire con essa. Abbiamo due possibilità per scoprire questo numero magico. Il primo metodo consiste nell'entrare nelle impostazioni del router e cercare la scheda con l'assegnazione degli indirizzi IP della rete. Le procedure di ingresso nella pagina di gestione del router variano da modello a modello. Vi rimando alle istruzioni presenti nei documenti del vostro router. Il secondo metodo è usare un software di scansione degli indirizzi IP. Allo scopo useremo NMAP. Questo è un software molto potente. Per gli utenti Windows sarà necessario collegarvi all'indirizzo: http://nmap.org/download.html e scaricare il software. Una volta avviato NMAP sarà necessario inserire l'indirizzo IP della nostra rete, solitamente *192.168.0.0/24*, per scoprirlo trovate l'indirizzo IP del computer sul quale state lavorando, ed avviate la scansione veloce. Dopo poco vedrete la lista di tutte le macchine connesse alla nostra rete. Se tutto sarà andato a buon fine sarà presente anche il Raspberry Pi. Lo identificheremo attraverso il MAC address che riporterà la scritta "Raspberry Pi Foundation". Per gli utenti Linux basterà aprire una finestra del "Terminale" ed inserire queste righe di testo:

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

dopo l'installazione di nmap, per eseguire la scansione basterà inserire:

```
sudo nmap 192.168.0.0/24
```

Installare SSH su Windows

Per connettervi tramite il protocollo SSH da macchine Windows, è necessario installare un software che gestisca tale protocollo. La mia scelta ricade su PuTTY. Un software piccolo ma molto stabile. E' possibile scaricarne una versione al seguente indirizzo: http://www.putty.org. Avviate PuTTY e create un nuovo profilo con indirizzo IP del Raspberry Pi, nome utente, che se non avete modificato risulta essere "pi", e password che se non modificata è "raspberry".

Avviare SSH da MAC OSX

Per avviare una comunicazione ssh diretta al vostro Raspberry Pi appena configurato, non dovete fare altro che avviare dalla finestra Applicazioni/Utility del vostro MAC una finestra “Terminale”. All’interno della finestra digitate i seguenti comandi ssh -l user indirizzo ip del Raspberry.

```
ssh -l pi 192.168.0.200
```

Utilizzare il Raspberry senza router e con il cavo di rete Ethernet

Potremmo non disporre di una rete locale a cui collegare il Rapsberry Pi. In questo caso il mio consiglio è quello di collegare direttamente il Raspberry Pi al computer che utilizzeremo, attraverso un comunissimo cavo di rete Ethernet. A tale scopo è necessario definire l'indirizzo IP statico del Raspberry Pi, in maniera da riconoscerlo quando lo collegheremo al computer. Gli indirizzi IP, la password di connessione e le impostazioni di assegnazione automatica degli indirizzi di rete sono contenuti nel file *interfaces*. Per modificare tale file è necessario entrare nel Raspberry Pi tramite un terminale SSH. Una volta entrati nel sistema, digitare il comando

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

Verrà aperto l'editor di testo nano, presente di default sul sistema Linux Raspbian. Al posto della riga di testo:

```
iface eth0 inet dhcp
```

sarà necessario inserire queste righe di codice. Dunque cancellate la riga *iface eth0 inet dhcp* ed inserite

```
iface eth0 inet static

address 192.168.1.1

netmask 255.255.255.0

gateway 192.168.1.254
```

Non vi resta altro che impostare l'indirizzo IP statico anche sul vostro computer. Rimando alle guide online per effettuare tale procedura. Le differenze tra i sistemi operativi renderebbero il compito troppo prolisso da spiegare in questa sede. Vi basti capire che sarà necessario disabilitare il sistema DHCP ed impostare un indirizzo manuale con questi parametri

```
IP: 192.168.1.2

NETMASK: 255.255.255.0

GATEWAY: 192.168.1.254
```

A questo punto il vostro Raspberry Pi sarà raggiungibile collegandolo direttamente alla porta Ethernet del computer con un comodo cavo. Per la procedura di collegamento da terminale eseguite la procedura standard, inserendo come indirizzo quello utilizzato in fase di impostazione sul Raspberry Pi e cioè 192.168.1.1.

La prima connessione

Dopo l'accesso al Raspberry, il sistema ci avvisa che non conosce l'host al quale si sta collegando e vi chiede di autorizzarlo. Cliccate su SI per PuTTY, scrivete "yes" per il terminale e date invio. Il secondo passaggio è il riconoscimento delle credenziali di accesso. Inserite il nome utente pi e la password raspberry.

Spegnere il Raspberry Pi

Se tutto è andato a buon fine possiamo provare a spegnere per la prima volta il Raspberry Pi. Per effettuare questa operazione è necessario dare questo comando da terminale:

```
sudo shutdown -h now
```

se invece il nostro intento è quello di riavviare il sistema, allora il comando sarà:

```
sudo shutdown -r now
```

Quando il Raspberry Pi vuole nascondersi

Quando spegnete e riaccendete il Raspberry Pi, il router potrebbe cambiare il suo indirizzo IP. Per evitare questo, è utile impostare un indirizzo IP statico, cioè immodificabile sul Raspberry Pi.
Per farlo basta collegarsi al Raspberry Pi come descritto precedentemente, e dal terminale inserire queste righe di testo:

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

Individuare la riga eth0. Sotto questa riga inserite:

```
address 192.168.1.xx

netmask 255.255.255.0

network 192.168.0.0

broadcast 192.168.1.255

gateway 192.168.1.yy
```

Assicuratevi di sostituire i valori con l'indirizzo desiderato e l'ip del gateway. Per conoscere l'indirizzo ip del gateway usare il comando da terminale

```
ifconfig
```

Aggiornare il Raspi

Una volta terminata questa fase, non vi resta che aggiornare il Raspberry Pi. Per effettuare questa operazione è necessario che il Raspberry sia connesso ad internet attraverso il router di rete. Digitiamo da terminale il comando:

```
sudo aptitude update && sudo aptitude upgrade && sudo shutdown -r now
```

A seguito di questo comando il Raspberry Pi comincerà l'aggiornamento chiedendo conferma nel caso in cui fosse necessario installare nuovi pacchetti software. Se questa ipotesi dovesse presentarsi scrivete "SI" e date invio. Alla fine dell'aggiornamento di sistema, il Raspberry Pi si riavvia.

I comandi base di Linux da Terminale

Come detto in precedenza, tutti i tutorial di questo libro vengono creati, modificati ed avviati tramite il Terminale di Linux. E' giusto a quanto punto familiarizzare con i comandi base del terminale. Inizialmente la sensazione di aver perso il mouse e le cartelle spiazza davvero parecchio, ma dopo poco tempo, la maggior parte dei comandi che usiamo generalmente sui computer, risulta semplice e di facile applicabilità. Creare un file o delle cartelle, cancellarle, copiare o incollare un file, sono solo alcune delle semplici operazioni che compiamo quotidianamente davanti ai nostri computer, vediamo com effettuare le stesse operazioni usando il terminale di Linux Raspbian.

Iniziamo col dire che la finestra di terminale presenta sempre in alto a sinistra la dicitura

```
pi@raspberrypi - $
```

A fianco al simbolo del dollaro $, è possibile inserire i comandi per copiare, cancellare e modificare i file e le directory, e per avviare i programmi scritti in Python. Dopo il comando digitato sarà sempre necessario schiacciare il tasto invio per cominciare l'operazione. Possiamo cominciare ad esplorare la directory corrente dando il comando:

```
ls
```

Il risultato sarà la lista di tutte le cartelle e di tutti i file presenti nella directory. Le colonne visualizzano (da sinistra verso destra):

1 permessi del file/cartella (drwx e tutta la sfilza di trattini); i permessi li vedremo più avanti

2 numero di hard link esistenti per ogni elemento, questo dato non ci interessa

3 utente con i permessi di owner (di solito è il creatore del file)

4 nome del gruppo che ha i permessi sul file

5 dimensioni

6 mese, giorno, anno di creazione del file

7 nome del file

Proviamo adesso a creare una nuova directory (cartella), digitando:

```
sudo mkdir pippo
```

poi digitiamo ancora il comando:

```
ls
```

notate come la lista è cambiata rispetto alla precedente? E' infatti presente la directory *pippo*. Proviamo adesso ad entrare nella directory *pippo* ed a creare un file di testo. Per entrare nella directory diamo il comando:

```
cd pippo
```

digitiamo adesso il comando

```
ls
```

notate come la directory sia vuota. A questo punto creiamo un file di testo con il comando:

```
sudo nano ilprimofile.txt
```

si apre la finestra di un programma chiamato *nano*. Questo è un potente editor di testo integrato nel sistema Linux Raspbian. Scrivete qualcosa all'interno del file, e salvatela premendo la combinazione di tasti CTRL+X e rispondendo "yes" alla domanda che chiede di sovrascrivere il file. Restando nella directory *pippo* date il comando

```
ls
```

la finestra mostra il contenuto della cartella *pippo*, e quindi il file txt presente in essa con il nome *ilprimofile.txt*. Usciamo dalla directory *pippo* con il comando:

```
cd ..
```

Complimenti abbiamo creato una directory di nome *pippo*, ed un file di testo nella directory. A questo punto non ci resta che rientrare nella directory *pippo* con il comando:

```
cd pippo
```

e rinominare il file *ilprimofile.txt*. Per fare questo inseriamo questo comando:

```
mv ilprimofile.txt rinominatofile.txt
```

Il comando mv rinomina ilprimofile.txt in rinominatofile.txt. Il termine mv deriva dal fatto che in inglese questa procedura viene descritta come “move”: spostamento.

Provate adesso copiare un file con il comando cp:

```
cp rinominatofile.txt ilsecondofile.txt
```

Perfetto date ancora una volta il comando

```
ls
```

verificate che siano presenti i file, poi eliminateli dando i comandi

```
sudo rm rinominatofile.txt
```

e poi

```
sudo rm ilsecondofile.txt
```

uscite di nuovo dalla directory

```
cd ..
```

ed eliminate la directory *pippo* inserendo il comando

```
sudo rm -r pippo
```

se tutto è andato a buon fine dando il comando ls la directory *pippo* non dovrebbe comparire nella lista delle directory.

Chiudere un programma in esecuzione

Un altro suggerimento utile è relativo all'uscita da un programma in esecuzione. Se con i sistemi operativi fini ad oggi utilizzati, quelli come Windows o MAC OS, con interfaccia grafica, basta cliccare con il mouse sulla X sul bordo delle finestre, con il terminale è necessario schiacciare una combinazione di tasti. Ogni finestra di terminale può eseguire un programma per volta. Spesso capita di dover interrompere l'esecuzione di un programma. Per effettuare questa operazione è possibile schiacciare i pulsanti **CTRL+C** nella finestra del terminale. Questa procedura chiude l'esecuzione del processo, in tale maniera si esce dal programma in esecuzione.

Chiudere la connessione SSH: logout

Quando abbiamo terminato il lavoro ma non intendiamo spegnere il Raspberry Pi, sarà necessario disconnettersi dal protocollo ssh. Per disconnettersi dal terminale del Raspberry Pi senza spegnerlo, basta digitare:

```
logout
```

Python

La mia idea di inserire un capitolo su Python all'interno di un libro di questo tipo, non è assolutamente quella di insegnare la programmazione attraverso questo linguaggio. Servirebbe un manuale ben più corposo e complesso, oltre che un insegnante più preparato di me. La mia volontà è quella di fornire brevi e semplici nozioni, in maniera tale da rendere il lettore capace di leggere e comprendere un programma scritto in Python. Questo capitolo precede infatti quello sui tutorial da eseguire con Raspberry Pi. La scelta di anteporre queste nozioni ai tutorial, è quella di permettere la comprensione e la modifica dei programmi di seguito illustrati. Il mio suggerimento è quello di cominciare a leggere i programmi esistenti. Modificare le righe di codice è essenziale per comprenderne il funzionamento. In seguito, se come spero, questa lettura vi avrà incuriosito, vi esorto a studiare pubblicazioni più complete sull'utilizzo di Python. Ma bando alle chiacchiere e cominciamo questo stimolante viaggio all'interno di uno dei più completi e semplici linguaggi di programmazione di sempre: Python. Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti. Fu ideato da Guido van Rossum all'inizio degli anni novanta. Il nome fu scelto per via della passione di van Rossum per i Monty Python e per la loro serie televisiva Monty Python's Flying Circus. Python è un linguaggio multipiattaforma. Significa che funziona sia su Linux, sia su MAC OS, e sia su Windows. Le caratteristiche di tale linguaggio sono la semplicità e la versatilità. Python è stato progettato in modo da essere altamente leggibile. Visivamente si presenta in modo molto semplice e ha pochi costrutti sintattici rispetto a molti altri linguaggi strutturati come *C*, *Perl* o *Pascal*. Un aspetto molto importante da tenere ben a mente quando si comincia a programmare in Python, è il fatto che il software utilizza un sistema di **indentazione** per delimitare i blocchi di programma. Ovvero, invece di usare parentesi o parole chiave, usa gli spazi (4 per lo standard Python) per indicare i blocchi nidificati. L'esempio che segue chiarisce questo aspetto:

```
def fattoriale (x):
    if x == 0:
        return 1
    else:
```

```
return x * fattoriale(x-1)
```

Installare Python

I procedimenti di installazione di Python variano a seconda del sistema operativo utilizzato. Se siamo i felici possessori di una macchina Linux, questi si ridurranno all'avvio dell'interfaccia, in altri casi basterà eseguire poche righe di codice, o avviare degli applicativi che provvederanno ad installare Python sul computer.

Installare Python su una macchina Linux

Oggi tutte le macchine con sistema operativo Linux o UNIX sono corredate di Python. Per verificare l'effettiva presenza del software digitiamo sul nostro terminale:

```
python
```

Se il software è correttemene installato dovremo leggere la versione di Python che abbiamo sul nostro sistema.

Installare Python su una macchina Windows o UNIX (MAC OSX)

Per installare Python su sistemi operativi basati su Windows occorre collegarsi all'indirizzo https://www.python.org/download/releases/3.2.3/ e scaricare l'ultima versione del pacchetto di installazione. Ricordo che è un software libero sotto licenza Creative Common.

Cominciare ad usare Python

Aprite IDLE, la GUI di Python. Dovreste vedere una finestra di questo tipo:

```
Python 2.2.2 (#1, Mar 21 2003, 23:01:54)

[GCC 3.2.3 20030316 (Debian prerelease)] on linux2

Type "copyright", "credits" or "license" for more information.

IDLE 0.8 -- press F1 for help

>>>
```

Il >>> è il modo che ha Python per informarvi che siete in modo interattivo, dove i comandi digitati sono immediatamente eseguiti. Provate a digitare 1+1 e Python vi risponderà immediatamente 2. In questa modalità potete provare Python e vedere come reagisce ai vari comandi. Usatela quando sentirete il bisogno di prendere confidenza con i comandi Python.

Usare Python da riga di comando

Se preferite non utilizzare Python da riga di comando non siete obbligati, usate IDLE. Per entrare nella modalità interattiva da riga di comando dovete semplicemente digitare **python**. Se volete eseguire un programma che avete scritto con un editor di testo (ad esempio Emacs è un ottimo editor per Python) non dovete fare altro che usare la sintassi:

```
sudo python nomeprogramma.py
```

La cosa fondamentale è nominare sempre il file di testo, che rappresenta il programma scritto in python, con estensione **.py** .

Le Variabili

Una delle potenzialità dei linguaggi di programmazione è la manipolazione delle variabili. Una variabile è un nome a cui fa riferimento un valore.

```
>>> n = 10
```

Dopo questa dichiarazione di variabile, la lettera n rappresenterà il numero 10. Dunque potrò effettuare anche delle operazioni con questa variabile, senza richiamare il numero:

```
>>> sum = n+2

>>> print sum
```

Il risultato di questa semplice operazione sarà la scrittura a video (il comando print) del valore di n sommato 2. Quindi il risultato sarà 12.

Le Keywords

Python, come tutti i linguaggi di programmazione, ha a disposizione una serie di keywords, parole chiave che permettono una serie di operazioni. Quando dichiariamo una variabile dobbiamo assolutamente evitare di usare una di queste parole chiave. E' necessario evitare queste parole nella dichiarazione delle variabili, poiché il software potrebbe fare confusione nel momento in cui andremo a richiamare una variabile.

Tutte le keywords di Python sono:

and del from not while as elif global or with assert else if pass yield break except import print class exec in raise continue finally is return del for lambda try

Gli Operatori

Gli operatori sono una serie di simboli che in Python servono a compiere operazioni, per lo più matematiche.
Questi sono addizione +, sottrazione -, moltiplicazione *, divisione / e elevazione a potenza **.
Esempio:

```
x = 10

y = 5

sum = x + y

print sum
```

o ancora

```
x = 10

y = 5

total = x + (x * y) - x

print total
```

il risultato del primo algoritmo sarà: 15, quello del secondo 50.

Gli input

A volte è necessario che l'utilizzatore del programma che abbiamo scritto, inserisca uno o più valori da assegnare alle variabili definite nell'algoritmo. Esempio:

```
>>> input = raw_input()

>>> print input
```

Quando si avvia questa funzione, il programma chiede all'utente di inserire un valore. Quando l'utente inserire un valore e preme "Invio", il programma stampa a video l'input inserito dall'utente.

Le Espressioni Booleane

Le espressioni booleane, sono una serie di espressioni che possono assumere valore di Vero o Falso.
Gli operatori booleani sono:

x == y che significa x è uguale a y

x != y che significa x è diverso da y

x > y che significa x è più grande di y

x < y che significa x è più piccolo y

x >= che significa x è più grande o uguale a y

x <= che significa x è più piccolo o uguale a y

x is y che significa x è lo stesso di y

x is not y che significa x non è lo stesso di y

Anche se tutto questo può apparire alquanto strano, abbiate fede e continuate a leggere. Tutto acquisterà senso più avanti.

Le espressioni logiche

In Python esistono tre tipi di operatori logici: **and**, **or** e **not**.
I significati sono **e**, **o** e **no**.
Esempio:

```
x > 0 and x < 10
```

Se x è più grande di 0 e se x è minore di 10 allora l'espressioni assumerà valore **True** cioè Vera.

L’esecuzione condizionale

Siamo arrivati ad uno dei capitoli più importanti del libro. In tutti i linguaggi di programmazione è presente questo tipo di operazione. Essa determina un comportamento del programma basandosi su determinate variabili. Un esempio chiarirà meglio:

```
x = 6

if x > 5 :

    print ‘x è maggiore di 5’
```

Cosa accadrà quando il programma incontrerà questa funzione. Verificherà che la variabile x sia più grande di 5. Qualora questa eventualità risultasse vera, allora il programma stamperà a video il messaggio “x è maggiore di 5” in caso contrario, e dunque se x è minore di 5, il programma non comunicherà nulla all’utente.

Esecuzione condizionale alternativa

Possiamo aggiungere un livello di complessità al nostro algoritmo prevedendo anche l'eventualità che x sia minore di 0:

```
x = 4

if x > 5 :

   print 'x è maggiore di 5'

else :

   print 'x è minore di 5'
```

In questo caso il programma verificherà prima che x sia maggiore di 5. In caso contrario (**else**) scriverà a video la stringa "x è minore di 5".

Le righe di commento

All'interno di ogni file di programmazione sono presenti delle righe di commento. Esse sono scritte dai programmatori, e fungono da istruzione, e spesso indicano il nome del programma e la data di creazione. E' buona norma corredare i programmi con abbondanti righe di commento, per favorire le modifiche successive e descrivere il funzionamento del programma. In Python queste righe di commento vengono precedute dal segno # cancelletto. Ogni volta che inseriamo il simbolo cancellato davanti ad una riga di codice di Python, indichiamo all'interprete, al programma, di saltare tutta la riga di codice. Spesso troverete le righe di commento all'interno dei programmi scritti nei tutorial che seguiranno.

Le funzioni

All'interno del nostro programma possiamo decidere di spezzare il codice e dividerlo in più funzioni.
Una volta dichiarata una funzione utilizzando la keyword **def**, questa può essere richiamata dal programma digitando la parola utilizzata per la dichiarazione della funzione.
Facciamo un esempio:

```
def stampa_filastrocca() :

    print 'Sopra la campa la capra campa, sotto la panca la capra crepa'
```

Quando nel nostro programma invocheremo la funzione da noi creata e chiamata *filastrocca*, questo comincerà ad eseguire tutto quello che abbiamo scritto all'interno della funzione filastrocca:

```
def stampa_filastrocca() :

    print 'Sopra la campa la capra campa, sotto la panca la capra crepa'

print 'Conosco una filastrocca'

stampa_filastrocca()
```

il programma stamperà a video l'istruzione contenuta nel comando print, seguita dall'istruzione contenuta nella funzione stampa_filastrocca dichiarata precedentemente. Quindi il risultato di tale algoritmo sarà:

```
Conosco una filastrocca

Sopra la campa la capra campa, sotto la panca la capra crepa
```

Le Iterazioni

Tutti i programmi che si rispettino hanno dei *cicli*. E' proprio attraverso questi cicli che il programma compie le azioni desiderate. In Python esistono diversi modi per impostare delle azioni cicliche.

Il ciclo While

Uno di essi è il ciclo **while.** Passiamo ad un esempio che chiarisce meglio il concetto. Se volessimo compiere un'azione ripetendola per 5 volte potremmo o scrivere determinata azione 5 vuole, oppure impostare un ciclo **while**.

Stampa a video 5 volte la parola "ciclo".

```
print 'ciclo'

print 'ciclo'

print 'ciclo'

print 'ciclo'

print 'ciclo'
```

oppure potremmo usare un più economico e semplice "while":

```
input = 'ciclo'

n = 5

while n > 0 :

    print input

    n = n - 1
```

Pensate a quanto sarebbe utile questa opzione nel caso in cui servirebbe scrivere 100 vuole la parola ciclo. Il ciclo **while** viene altresì spesso utilizzato in associazione con l'espressione booleana **True**, *Vero*.

```
while True :

    line = raw_input('>')

    if line == 'finisci':

        break

    print line

print 'Finito!'
```

Questo ciclo chiede all'utente un valore (line = raw_input('>')), e stampa tale valore sullo schermo (print line). Quando l'utente inserisce la parola "finisci", allora il ciclo si interrompe (break) riconoscendo tale parola nell'operazione condizionale "if" (if line == 'finisci': break).

Il ciclo For

A volte è necessario inserire dei cicli che ripetono determinate azioni un numero definito di volte. Per questa eventualità, oltre al già visto ciclo **while** si può utilizzare il ciclo **For**. Vediamo subito un esempio e poi lo commenteremo assieme.

```
amici = [‘Antonio’, ‘Pasquale’, ‘Rosa’, ‘Margherita’]

for amico in amici:

    print ‘Buon Anno:’, amico

print ‘Finito!’
```

Questo ciclo stampa a video questo risultato:

```
Buon Anno: Antonio

Buon Anno: Pasquale

Buon Anno: Rosa

Buon Anno: Margherita

Finito!
```

La prima riga (amici = [‘Antonio’, ‘Pasquale’, ‘Rosa’, ‘Margherita’]) è una dichiarazione di “Array”, ovvero un insieme di dati all’interno di due parentesi quadre. All’interno delle parentesi possiamo inserire numeri o testo. In questo caso abbiamo inserito dei nomi di persona. La seconda (for amico in amici) è la stringa che iniziali zia il ciclo for. Il ciclo stampa a video (print) la parola “Buon Anno:” (‘Buon Anno:’. Notate come le parole sono chiuse tra due virgolette) seguita da uno dei valori dell’array amici (for amico in amici … print … amico). Alla fine del ciclo viene stampata a video la parola “Finito!”. Tutto chiaro? Esercitatevi su Python modificando le stringhe, cambiando i valori, miscelando i concetti.

Le stringhe

Una stringa è una sequenza di caratteri. Si può avere accesso ai caratteri contenuti nella stringa attraverso una serie di semplici comandi. Cominciamo subito con un esempio:

```
frutta = ‘mela’

lettera = frutta [1]

print lettera
```

Questo istruzioni dichiarano una stringa con il nome “frutta”. La stringa contiene un testo “mela”. Poi c’è un’altra dichiarazione di variabile “lettera”. Il contenuto di questa variabile è la prima lettera della stringa frutta: frutta[1]. Infine viene stampata la variabile “lettera” e quindi la lettera “e”. Esatto proprio la lettera “e” vi aspettavate che venisse richiamata la lettera “m”? La numerazione delle stringhe parte dal numero 0. Dunque per stampare la prima lettera e quindi la lettera “m”, avreste dovuto scrivere:

```
frutta = ‘mela’

lettera = frutta [0]

print lettera
```

In questo caso la lettera stampata è la “m”.

Il comando Len

E’ anche possibile sapere la lunghezza diana certa stringa. Per farlo basta digitare il comando “len”.

```
frutta = ‘mela’

lunghezza = len(frutta)

print lunghezza
```

Questo comando stamperà a video il numero 6. La lunghezza della stringa frutta.

Se invece volessimo richiamare solo una parte della stringa allora occorrerebbe utilizzare i comandi di “slice”.

```
frutta = ‘mela rossa’

primo = frutta[0:4]

secondo = frutta[5:]

print primo

print secondo
```

Questi comandi prendono le prime quattro lettere della stringa e le inseriscono nella variabile primo e poi le successive dalla quinta in poi [5:]. Il risultato sarà:

```
mela

rossa
```

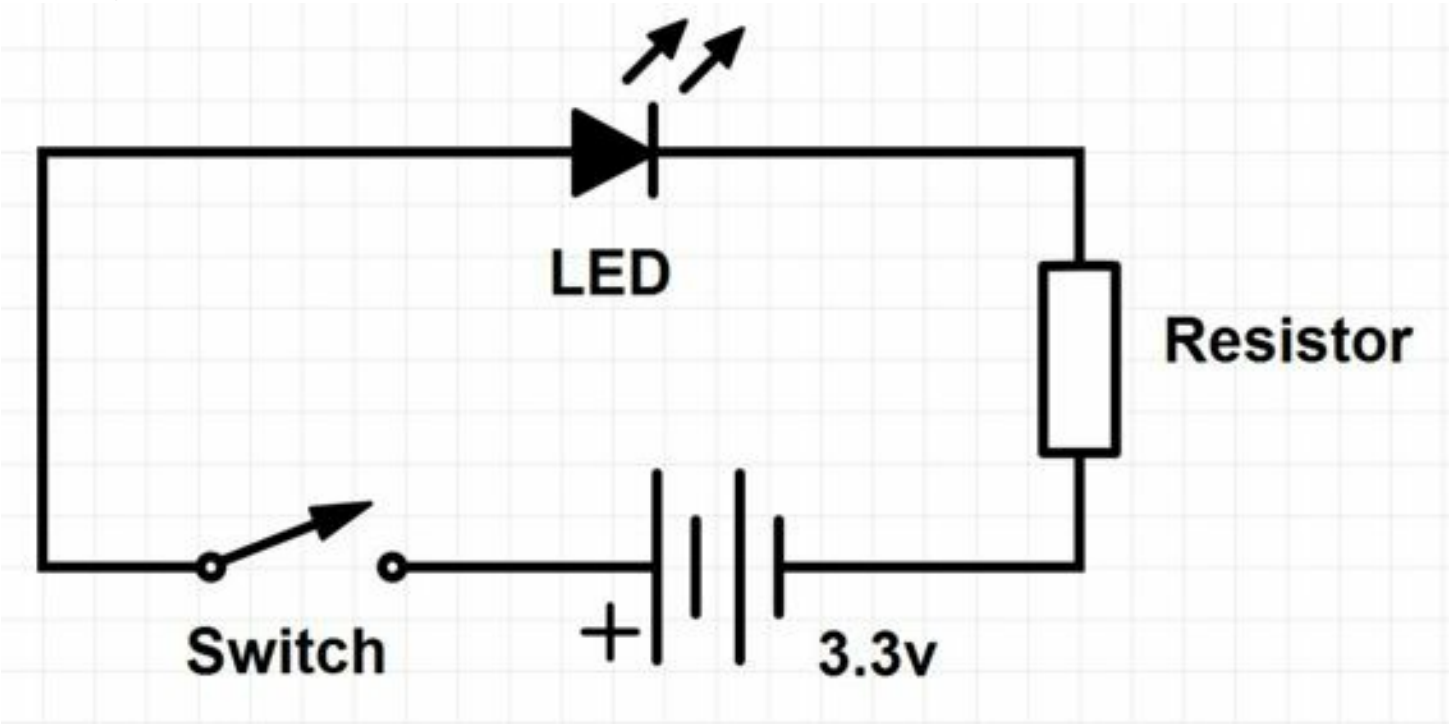
Abbiamo terminato il capitolo relativo al linguaggio Python utilizzato all’interno dei nostri tutorial. La cosa fondamentale è che siate riusciti a comprendere i concetti, ed avete cominciato ad esercitarvi con qualche esperimento. Vi esorto a leggere tante righe di codice per comprendere il funzionamento degli algoritmi. La programmazione con Python apre un mondo di possibilità. Adesso procuratevi una breadboard, alcuni fili di collegamento, delle resistenze, qualche switch e senza indugio passiamo al capitolo sui tutorial.

I TUTORIAL

Finalmente siamo giunti alla parte pratica del libro. Dopo tanto parlare adesso è arrivato il momento di agire. I tutorial sono ordinati per grado di difficoltà. Inizialmente saranno ben poco stimolanti. L'accendere un led a comando, o farlo lampeggiare non è il massimo della vita, mi rendo conto, ma è molto importante cominciare dai tutorial più semplici per passare via via a quelli più complessi. Preparate prima la parte elettronica, controllate sempre due o tre volte il circuito prima di dare tensione. Successivamente dedicatevi al software. Compilate il programma in Python e avviatelo. Il mio suggerimento è quello di comprendere le righe di codice. Capirne il significato. Successivamente modificare le variabili in gioco. In questo modo comincerete ad entrare nella dinamica di funzionamento del sistema. Quando si impara una nuova lingua, si può iniziare un corso ben strutturato, che parta dai principi base della grammatica, esponga le formule sintattiche corrette, eccetera, oppure fare un viaggio, e cominciare a praticare la lingua senza averla mai studiata. Lo scopo di questo libro è proprio quello di farvi provare in prima persona, comprendere e modificare. Sbagliare per imparare. Armatevi di tanta pazienza. Non innervositevi se le prime volte i tutorial non funzionano. Ricontrollate il circuito e poi il codice. L'errore è sempre dietro l'angolo. Ricordate che Python tiene in gran conto gli spazi, i rientri e le maiuscole. Procedete con calma e non affrettate i tempi. State facendo questi esperimenti per imparare. Prendetevi tutto il tempo di cui avete bisogno. Tutte le immagini dei circuiti presentati sono state preparate con un software open source molto utile: Fritzing (http://www.fritzing.org). Bando alle chiacchiere è cominciamo con il primo progetto.

Un circuito elettronico

Cominciamo questo capitolo abbandonando per un attimo il nostro Raspberry Pi. La cosa fondamentale da capire in questo momento è il concetto di corrente, led e interruttore. Mettiamoci comodi ed osserviamo questa immagine:

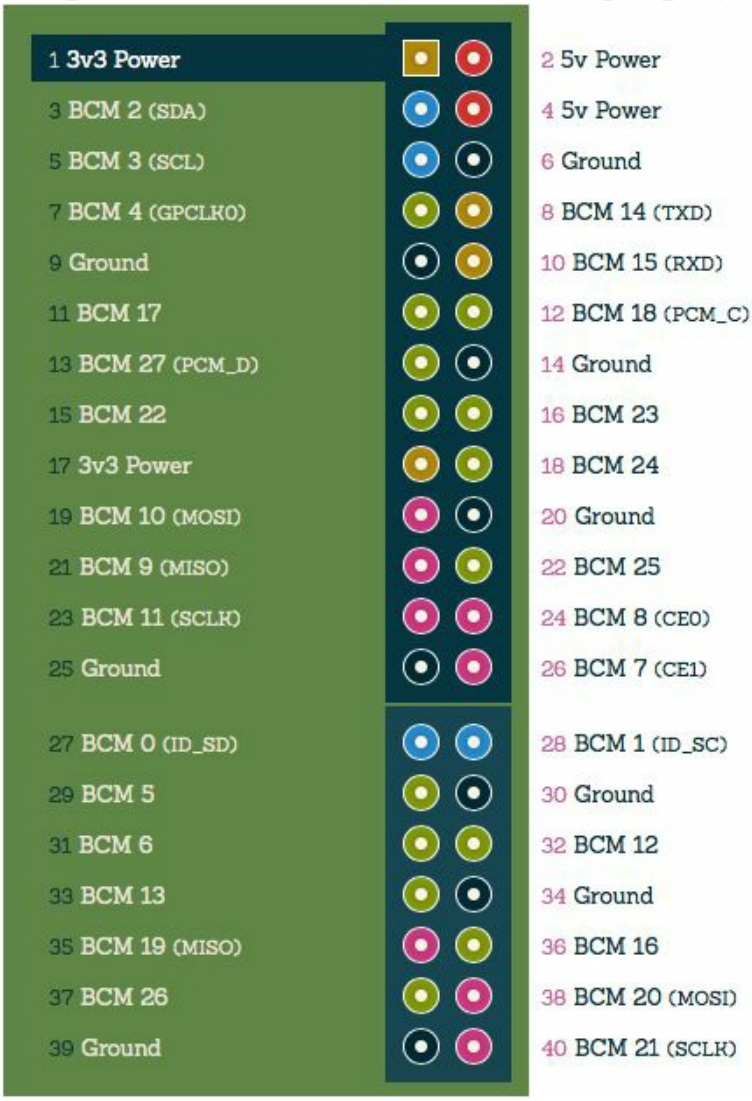


Notate come sono presenti quattro componenti elettrici. Un LED, in alto, una resistenza, a destra, una fonte di alimentazione da 3.3 Volts con il polo positivo sulla sinistra ed il negativo sulla destra, ed un interruttore chiamato *Switch*. Il led è un componente elettrico chiamato diodo ad emissione luminosa. Quando gli elettroni passano al suo interno, esso produce luce. Trasforma l'energia elettrica in energia luminosa. La resistenza, è un componente elettrico che "rallenta" la corrente. In questo esempio il voltaggio iniziale del circuito, il valore scritto vicino la fonte di corrente 3,3v, si abbassa grazie alla presenza della resistenza. L'interruttore, *switch*, in gergo potremmo chiamarlo, "sensore di pressione". Quando premuto, chiude il circuito e lascia passare la corrente. La fonte di corrente con i poli positivo (+) e negativo (-). Potrebbe essere una batteria per esempio. Poniamo attenzione al valore della fonte di alimentazione 3.3v, ovvero 3.3 Volts. Riconoscete questa parola? Volts? Esatto, è quella che si usa anche per misurare la corrente erogata nelle nostre case. In quel caso 220 Volts. Alta tensione, molto pericolosa per l'incolumità. Un contatto con una corrente a 220 Volts provoca la morte per

folgorazione, un contatto con una corrente a 3.3 Volts nemmeno il solletico. Cosa accade se “chiudiamo il circuito”. La corrente comincia a transitare nei cavi, passa all’interno del led e della resistenza. Questo passaggio produce un emissione luminosa all’interno del diodo led. Quello che andremo a fare adesso, sarà sostituire il nostro **Switch** con il Raspberry Pi. In maniera tale che a chiudere il circuito ci penseremo noi, con un codice scritto in Python. Tutto chiaro? Bene armatevi di pazienza, sangue freddo e cominciamo.

Il pettine GPIO del Raspberry Pi

Come detto precedentemente, il Raspberry Pi possiede uno spinotto di controllo denominato GPIO. Esso è composto da tanti piccoli pin metallici, presenti nell'angolo inferiore della scheda. Nei tutorial, spesso faremo riferimento ai pin presenti sul GPIO. Esistono due modi di identificare i pin. Nel primo metodo, il numero di riferimento è quello che identifica fisicamente il pin sulla scheda. Dunque il pin numero 1 sarà il primo in alto a sinistra, il due quello a destra, il tre il primo pin sulla seconda riga, eccetera. Il secondo metodo, che è quello utilizzato più spesso all'interno del libro, è quello che identifica i pin partendo dalla loro numerazione software. Dunque il pin GPIO4 in realtà è il pin numero 7. Per chiarire meglio tali concetti rimando all'utilizzo dello strumento di descrizione dei pin presente sul sito della Raspberry Pi Foundation: http://pi.gadgetoid.com/pinout/pin1_3v3_power.



Attraverso questo strumento identificherete il numero fisico del pin e il suo

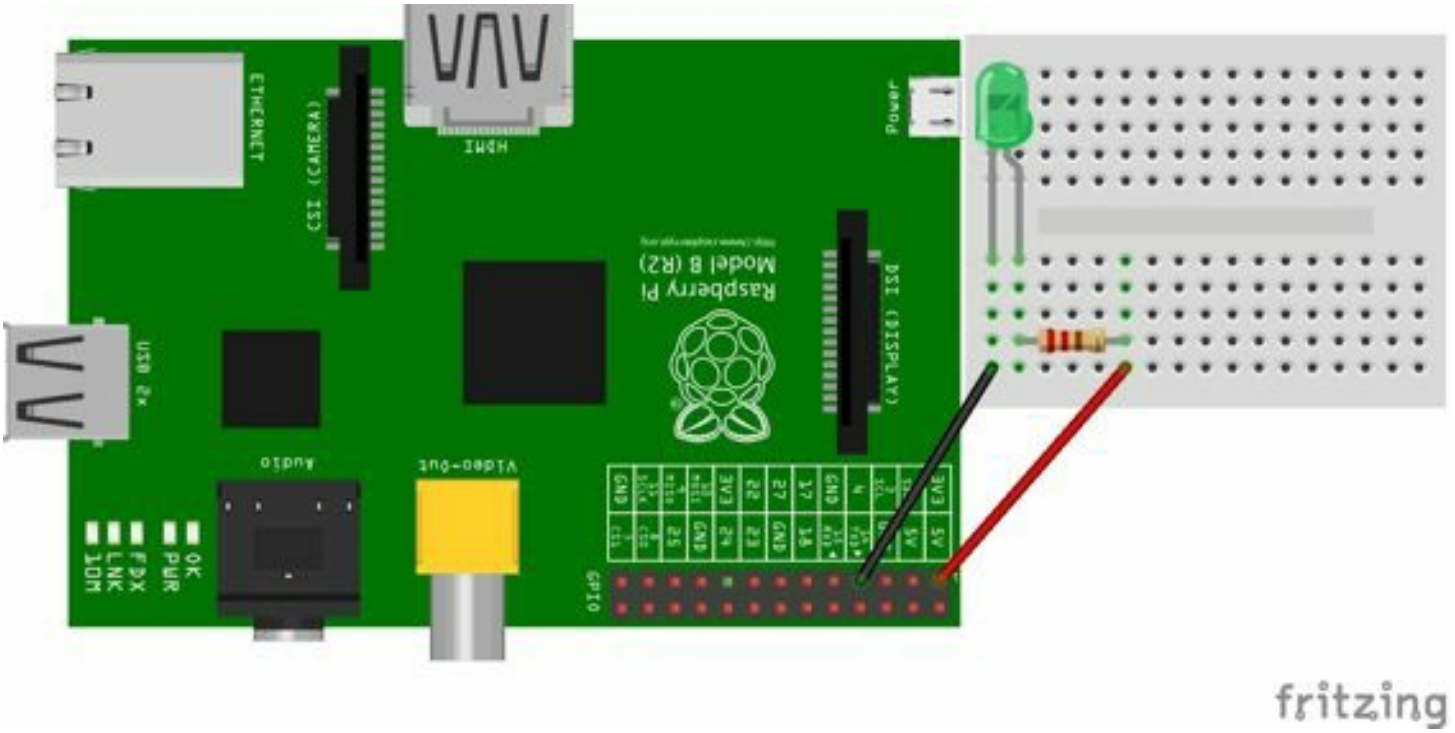
corrispondente software.

Blink

Il primo progetto che affronteremo sarà l'accensione di un led attraverso dei comandi impartiti al Raspberry Pi. La prima operazione da effettuare è quella di creare il circuito di funzionamento. Poniamo molta attenzione alla creazione del circuito. Un errore comporterebbe la rottura del regolatore di tensione del Raspberry Pi, o ancora peggio, la rottura del processore, rendendo la scheda inutilizzabile. Quindi leggete attentamente le istruzioni, non saltate alcun passaggio, e soprattutto controllate sempre due volte il circuito prima di dare tensione.

Il circuito

Il circuito che useremo per la prima prova sarà composto semplicemente da un LED di bassa potenza e la sua resistenza. Utilizzeremo il pin di 3,3V del pettine per l'alimentazione e un pin GPIO del Raspberry Pi per chiudere il circuito a massa. Niente vieta di fare l'opposto, cioè di usare il GPIO per fornire l'alimentazione e il pin GND del pettine per il negativo. La teoria elettronica dietro questo progetto è semplice. Se ad un led diamo corrente questo si accende. Per dare corrente ad un led, l'anodo di quest'ultimo (il piedino più lungo), ovvero il polo positivo, deve essere collegato alla parte positiva del sistema, ed il catodo, ovvero il polo negativo, a quella negativa. Attraverso Python invieremo dapprima un segnale negativo al catodo del led provocando la sua accensione, e successivamente invieremo un segnale positivo allo stesso provocando il suo spegnimento. Utilizziamo la breadboard,



poniamo un led in corrispondenza dei primi fori sulla breadboard. Poi posizioniamo la resistenza. Il valore della resistenza da utilizzare è 470 Ohm. Vi invito a leggere l'appendice per capire come leggere i colori delle resistenze. Poniamo adesso molta attenzione. E' il momento di collegare il circuito al Raspberry Pi. Assicuriamoci che la scheda sia spenta, e seguiamo l'illustrazione per capire dove inserire i pin. Uno dei cavi andrà al pin 4, l'altro sulla fonte di alimentazione 3,3v.

AVVERTENZE:

• Il microprocessore è DIRETTAMENTE COLLEGATO alla spina GPIO. Questo significa che non ci sono protezioni di nessun tipo tra i due. Controlliamo sempre due volte prima di dare tensione.

• Ogni singolo pin può assorbire o fornire fino a 16 mA di corrente

• La corrente massima erogabile dai pin 3,3 V è di 50 mA complessivamente.

• MAI collegare un cavo con una tensione superiore ai 3,3 V ad un pin GPIO.

• Calcolate sempre le correnti in gioco e non superate le soglie indicate o brucerete irrimediabilmente il regolatore di tensione o peggio il processore.

Python

Dopo la parte hardware, è necessario preparare quella software. Se non lo avete già fatto vi invito, dopo aver creato il circuito sopra descritto, a collegare il cavo di rete al Raspberry Pi, accenderlo, ed eseguire una connessione ssh dal vostro Mac o Pc seguendo le modalità descritte nel capitolo "Avvio del Raspberry in SSH". In questo progetto possiamo cominciare ad inviare dei comandi al Raspberry Pi direttamente dalla riga di comando invocando l'interprete di Python. Dunque il primo passaggio è quello di invocare Python:

```
sudo python
```

a questo punto l'interfaccia del terminale cambierà e vedrai questi caratteri:

```
>>>
```

cominciate a digitare questi comandi

```
import RPi.GPIO as gpio
```

dice a Python di caricare la libreria RPi.GPIO che serve per gestire i pin del BCM2835

```
gpio.setmode(gpio.BCM)
```

chiama la funzione **setmode** della libreria, e specifica che nel corso del programma i GPIO verranno selezionati con il numero di pinout del microprocessore (modo BCM) anziché con il numero del pin del pettine (modo BOARD).

```
gpio.setup(4, gpio.OUT)
```

configura il pin chiamato GPIO4 come uscita e setta lo stato della porta a livello logico alto (3,3V). In questo modo il LED rimane spento. Infatti abbiamo inviato una corrente positiva al polo negativo del led.

```
gpio.output(4, 0)
```

cambia lo stato del GPIO4 portandolo a livello logico basso, che

elettricamente corrisponde a 0V. A questo punto, se il circuito è stato fatto correttamente, il LED si accende. Infatti il polo negativo del led sarà portato correttamente a 0, ed esso si accenderà.

```
gpio.output(4, 1)
```

riporta lo stato del GPIO4 ad alto (3,3V). Il LED si spegne.

```
gpio.cleanup()
```

cancella tutte le configurazioni che abbiamo fatto e riporta tutti i GPIO che abbiamo utilizzato ai valori di default.

```
exit()
```

chiude l'interprete di Python.

Ha funzionato? Siete riusciti ad accendere e spegnere il led? Perfetto. Adesso proviamo a scrivere le stesse righe di codice, ma all'interno di un file con estensione py che potrete richiamare direttamente dalla finestra del terminale.

Creiamo un file di programma

Per cominciare sarà necessario creare un nuovo file di testo. Il programma che andremo a scrivere sarà composto da testo, ed avrà un nome con estensione "py" del tipo nomefile.py.

Nella finestra del terminale cominciate ad inserire l'istruzione di creazione file:

```
sudo nano blink.py
```

si aprirà l'editor di testo nano, che conoscete. All'interno di tale editor inserirete le istruzioni sopra eseguite. Dunque scriverete:

```
import RPi.GPIO as GPIO

gpio.setwarnings(False)

import time

# Funzione Blink

def blink(pin):

        GPIO.output(pin,GPIO.HIGH)

        time.sleep(1)

        GPIO.output(pin,GPIO.LOW)

        time.sleep(1)

        return

GPIO.setmode(GPIO.BCM)

# Setto il pin GPIO come output

GPIO.setup(4, GPIO.OUT)

# Richiamo la funzione blink 50 volte

for i in range(0,50):

        blink(4)

GPIO.cleanup()
```

In seguito salviamo il file con il comando Ctrl + X. confermiamo il salvataggio con “yes” e diamo invio.

Guardiamo nel dettaglio le righe di codice appena create.

```
import RPi.GPIO as GPIO

import time
```

Con queste istruzioni importiamo le librerie GPIO per la gestione degli input/output di Raspberry Pi, e importiamo la libreria **time** per la gestione della pausa tra un’accensione e l’altra del led.

```
def blink(pin):

    GPIO.output(pin,GPIO.HIGH)

    time.sleep(1)

    GPIO.output(pin,GPIO.LOW)

    time.sleep(1)

    return
```

Dichiariamo una funzione che chiamiamo blink. Tale funzione setta il pin 4 in stato HIGH (acceso) poi attende un secondo **time sleep(1)** e poi setta lo stato del pin in LOW. Attende un altro secondo e termina la sua funzione **return**.

```
GPIO.setmode(GPIO.BCM)
```

Con questa istruzione dichiamo al sistema che chiameremo i pin secondo il criterio BCM e non secondo il numero dei pin presenti sulla spina.

```
GPIO.setup(4, GPIO.OUT)
```

Impostate il pin 4 com OUTPUT

```
for i in range(0,50):

    blink(4)

GPIO.cleanup()
```

Richiamiamo la funzione link per 50 volte ed in fine riportiamo tutte l uscite GPIO a zero.

Eseguiamo il programma blink.py

Per eseguire questo programma basterà adesso richiamare l'interprete di Python ed indicare il file che vogliamo eseguire. Dunque scriverete:

```
sudo python blink.py
```

Ignorate l'eventuale messaggio di sistema:

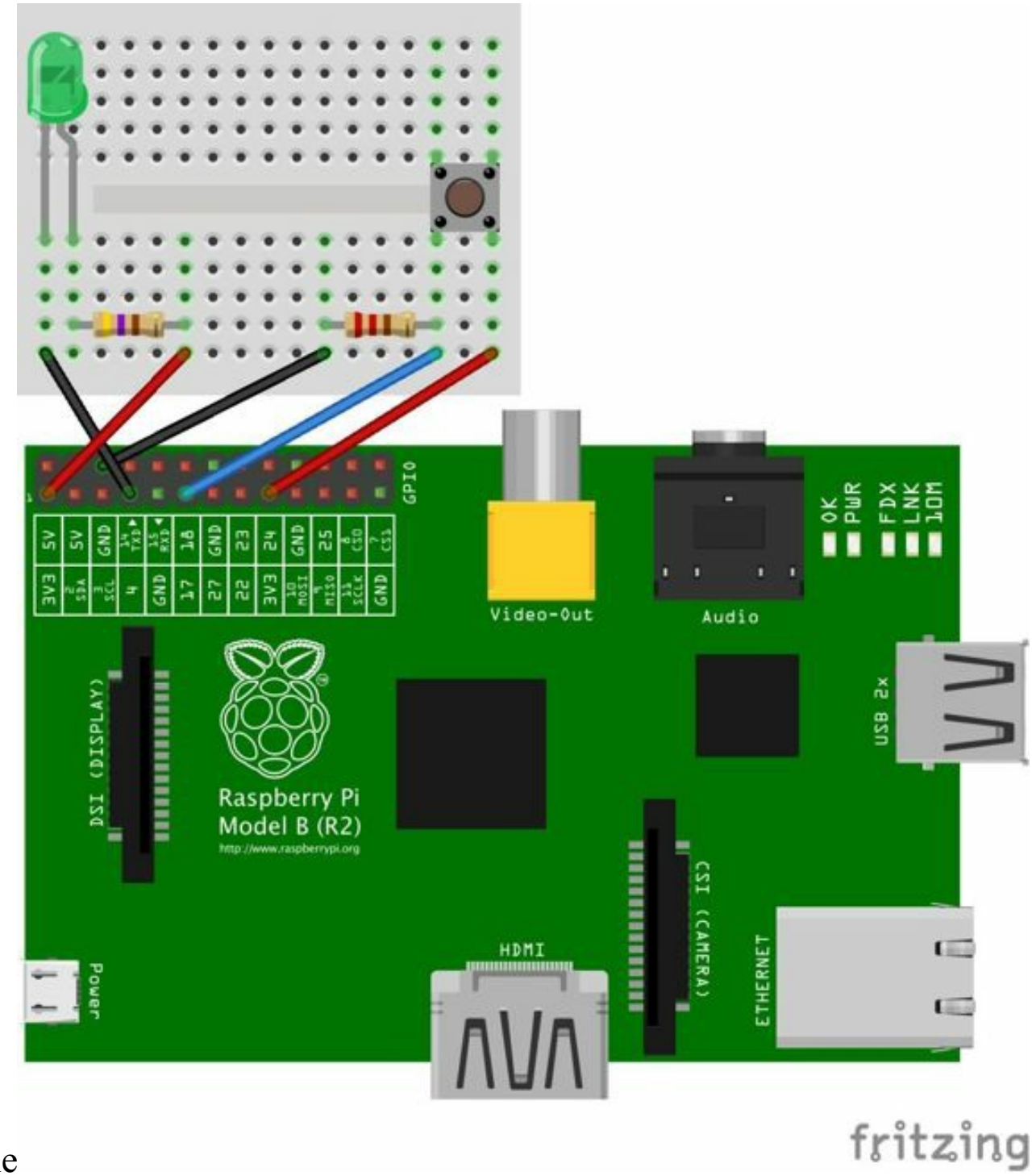
```
[…]RuntimeWarning: This channel is already in use, continuing anyway.  Use GPIO.setwarnings(False) to disable warnings.
```

se tutto è andato secondo i piani vedrete lampeggiare il led per 50 volte al ritmo di un ciclo ON/OFF al secondo.

Per terminare il programma basta premere **CTRL+C**.

Button

Passiamo al secondo tutorial di questo libro. Se siete riusciti a completare Blink, riuscirete senz'altro anche ad eseguire questo esperimento. Se con Blink avete cominciato a gestire gli output della scheda, adesso è il momento di gestire gli input. Infatti attraverso questo tutorial aggiungeremo al circuito sopra esposto un interruttore. Adesso il gioco si fa più duro ed è arrivato il momento di gestire gli INPUTS. La gestione degli input è più complessa. Verrebbe da pensare che basta collegare due cavi a due pin per gestire un input. Ma in questo modo il Raspberry Pi non saprebbe quando il pulsante è premuto o no in un determinato momento. Dovrebbe continuamente controllare lo stato del pulsante. Quando si parla di INPUT useremo frasi



come

pull-up e **pull-down**. In questo modo daremo al pin di ingresso un riferimento. Complicato? Continuate a leggere.

Il circuito

Come di consueto creiamo prima di tutto il circuito elettronico di funzionamento. Come potete notare al led abbiamo aggiunto un interruttore con relativa resistenza di pull-down. L'interruttore è posizionato tra la linea di tensione 3.3 e l'uscita GPIO 4. Inoltre si può notare la presenza di una resistenza di pull-down tra l'uscita 4 ed il negativo. La presenza di questa resistenza non è essenziale in quanto stiamo lavorando sui pin GPIO che includono una resistenza interna. Tuttavia è buona norma utilizzarla quando siamo in presenza di interruttori.

Python

Scriviamo il programma con l'editor nano. Nella finestra del terminale cominciate ad inserire l'istruzione di creazione file:

```
sudo nano button.py
```

Si aprirà l'editor di testo nano. Scrivete le righe di codice che leggete qui sotto:

```
import RPi.GPIO as GPIO

gpio.setwarnings(False)

import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(17, GPIO.IN)

GPIO.setup(4, GPIO.OUT)

while True:

    if (GPIO.input(17)):

        print "Bottone premuto"

        GPIO.output(4,GPIO.HIGH)

        time.sleep(1)

        GPIO.output(4,GPIO.LOW)

        time.sleep(1)

        GPIO.output(4,GPIO.HIGH)

        time.sleep(1)

GPIO.cleanup()
```

Usciamo dall'editor nano con il comando ctrl + X. Confermiamo con YES per il salvataggio file. Dalla finestra del terminale adesso digitiamo:

```
sudo python button.py
```

Se tutto è andato a buon fine, quando schiacciamo il pulsante del nostro circuito, sullo schermo apparirà la scritta “Bottone premuto”. Sulla breadboard lampeggerà il led collegato ai pin del nostro Raspberry Pi.

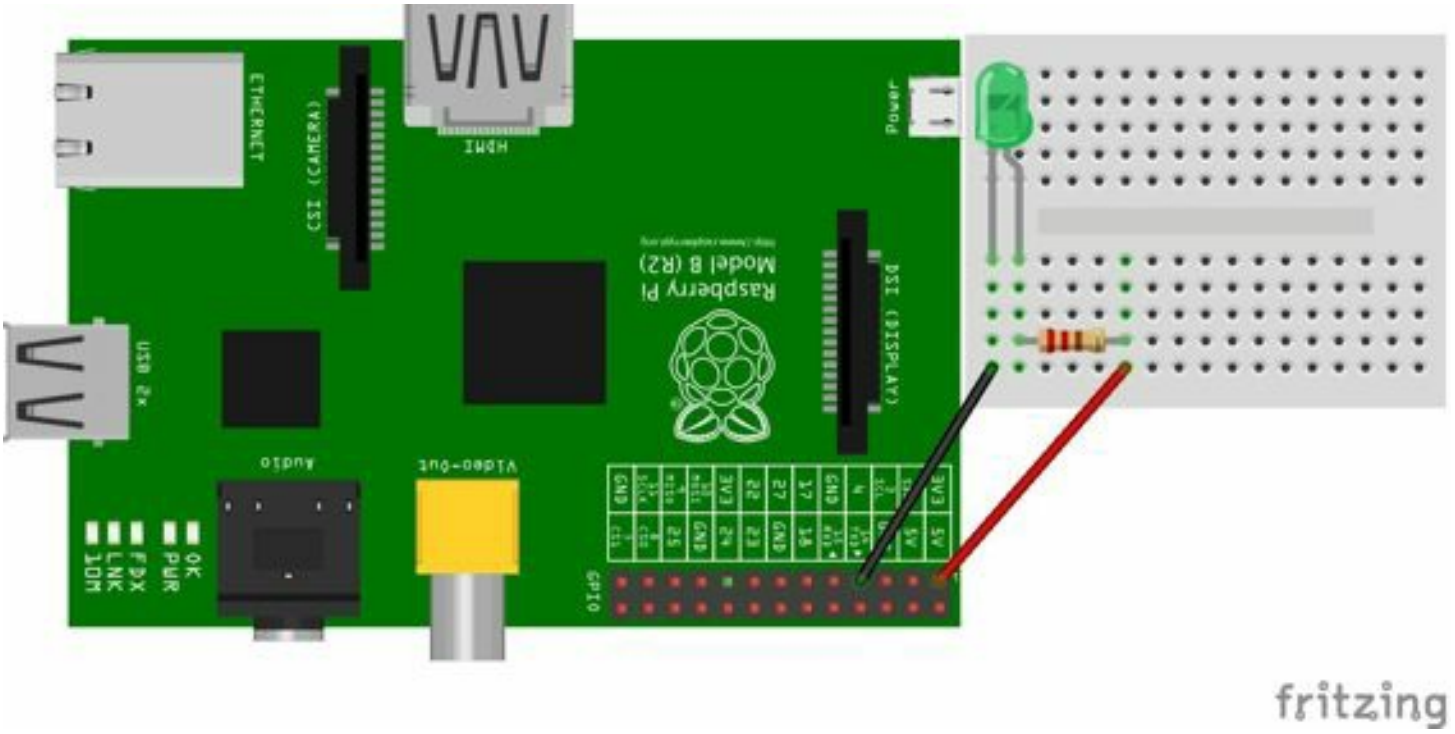
Per terminare il programma basta premere **CTRL+C**.

WebIOPi

Con questo progetto utilizzeremo il nostro Raspberry Pi come un interruttore remoto. Ovvero collegheremo al raspberry un led, e attraverso una pagina scritta in html, collegandovi al vostro Raspberry Pi, accenderete e spegnerete il led. Per questo progetto utilizzeremo WebIOPi. Questo è un framework open source (https://code.google.com/p/webiopi/). Le potenzialità di questo framework sono infinite. Questo tutorial introduce all'utilizzo del software WebIOPi. Potreste infatti decidere di sostituire il led con un relè, e comandare una lampada, il cancello di casa, lo scaldino del bagno o tutte queste cose assieme. Alla fine di questo progetto spiegherò come funzionano il Relè, e come procurarsi una scheda pronta da utilizzare con il Raspberry Pi.

Il Circuito

Il circuito che utilizzeremo per questo progetto è identico a quello utilizzato per il tutorial Blink. Colleghiamo un led al Raspberry Pi utilizzando i pin GPIO4 e 3,3 v.



Ricordiamoci il pin utilizzato per collegare il led. Nel nostro caso è il pin numero quattro partendo dall'alto, ovvero il pin GPIO4. E' importante tenere a mente questo valore, in quanto ci servirà successivamente quando ordineremo al nostro Raspberry Pi l'accensione e lo spegnimento di questo pin. Ricordiamoci di inserire la resistenza da 470 Ohm tra il led ed il nostro Raspberry Pi.

Il Software

In questo progetto il capitolo del Software è quello più corposo. Sarà necessario scaricare diversi file, tra i quali il pacchetto WebIOPi. Cominciamo con il comando che installa Python.

```
apt-get install gcc-4.7
```

```
apt-get install python3.2
```

Poi è il momento di scaricare il pacchetto WebIOPi. Al momento in cui il libro viene scritto, l'ultima versione del software è la 0.7.1. Date il seguente comando per scaricare il pacchetto:

```
wget http://downloads.sourceforge.net/project/webiopi/WebIOPi-0.7.1.tar.gz
```

Scompattate il pacchetto con il comando:

```
tar xvzf WebIOPi-0.7.1.tar.gz
```

Entrate nella directory con il comando:

```
cd WebIOPi-0.7.1
```

Ed avviate l'installazione con il comando:

```
sudo ./setup.sh
```

Al termine dell'installazione avviate il software con il seguente comando. Dopo aver dato il comando non noterete nulla sulla finestra del terminale. Non preoccupatevi, il programma è un *demone* che si avvia e resta silente fino a quando non lo richiamerete.

```
sudo /etc/init.d/webiopi start
```

Quando avremo terminato con il tutorial, per chiudere il programma WebIOPi basterà dare il seguente comando:

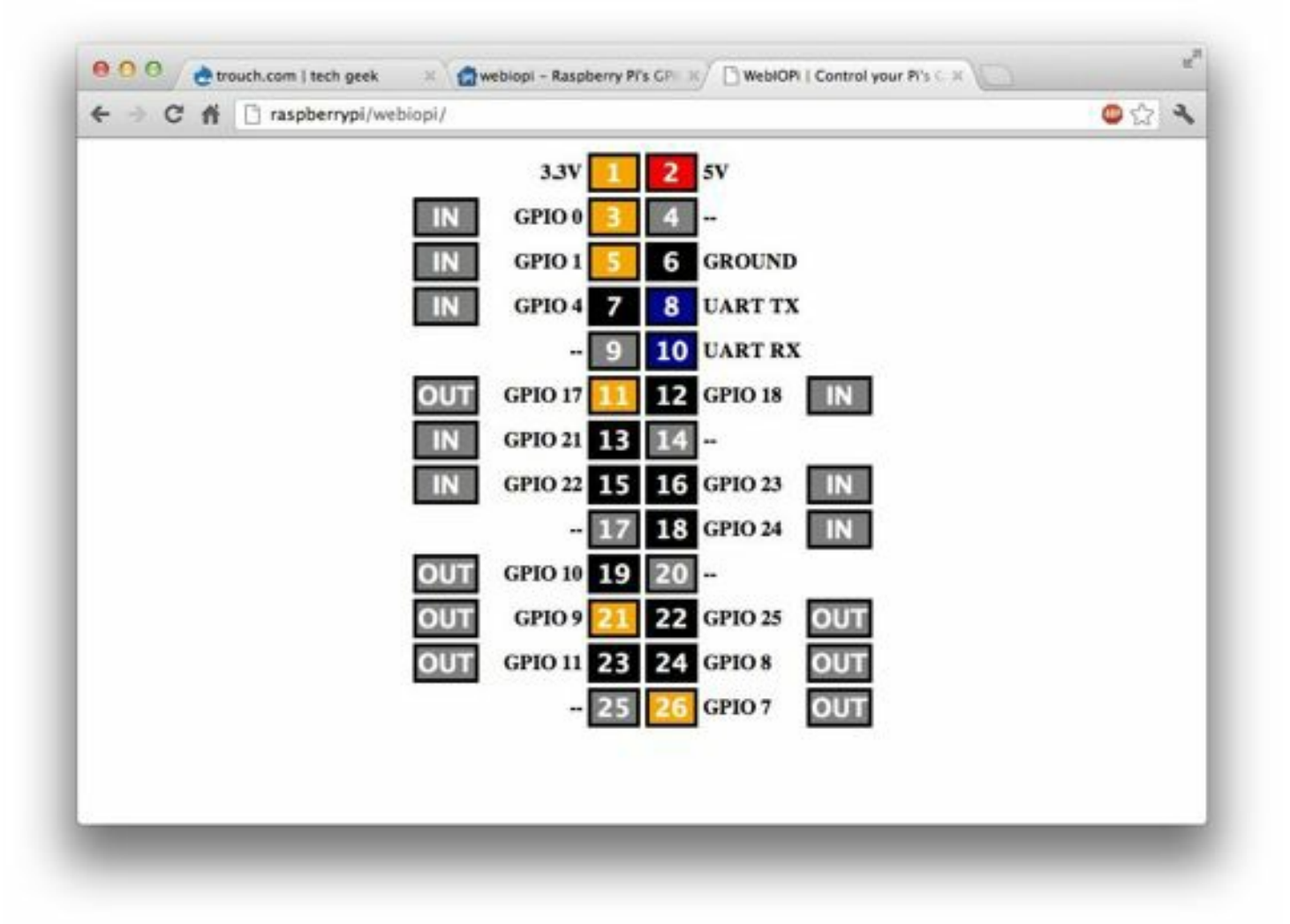
```
sudo /etc/init.d/webiopi stop
```

Ora non vi resta altro che aprire una finestra del browser che utilizzate di

solito (Chrome, Safari, Firefox Explorer, ecc.) su di un computer connesso alla stessa rete locale del Raspberry Pi, e scrivere nella barra degli indirizzi il seguente testo. Avendo cura di sostituire al seguente indirizzo IP, quello del vostro Raspberry Pi.

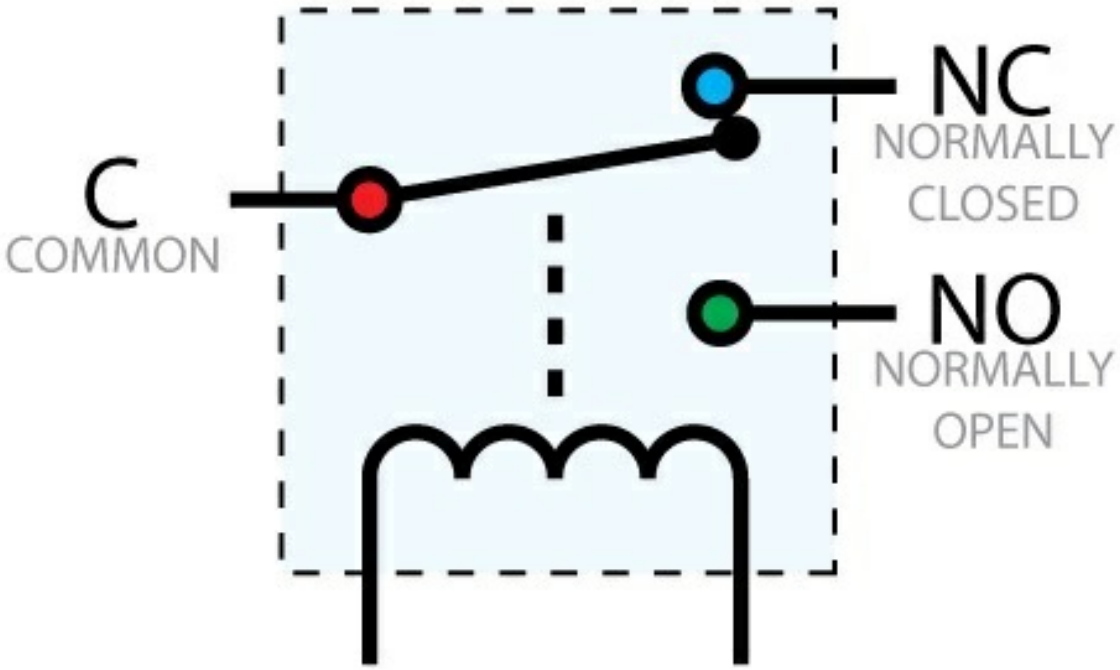
http://192.168.1.1:8000/

Inserite la user e la password di default: User: **webiopi** Password: **raspberry**. Scegliete l'opzione "GPIO Header". Vedrete l'immagine della porta GPIO ed i pin IN/OUT del Raspberry Pi. A questo punto non resta altro che provare ad accendere e spegnere il led. Posizionate il mouse sul pin numero 7 ovvero il pin GPIO numero 4, impostate il pin come OUT cliccando sull'icona "IN/OUT". In seguito cliccate sul numero del pin scelto, il numero 7. Quando il pin cambia colore, il led dovrebbe accendersi se tutto è andato a buon fine. Funziona? Benissimo, ma perché accendere e spegnere semplicemente un led. Proviamo ad aumentare la complessità dell'esperimento. Che ne dite di aggiungere una scheda relè?



Funzionamento del Relè

I relè funzionano come degli interruttori comandati da un segnale invece che da un nostro gesto. In figura (http://www.phidgets.com) potete notare i componenti che sono presenti in un relè. Una elettrocalamita, quella che vedete in basso, e tre connettori: C (comune), NC (normalmente chiuso) NO (normalmente aperto). Quando l'elettrocalamita è spenta, e non vi è corrente all'interno di essa, il connettore C, è unito a quello NC. Dunque il circuito che passa tra C e NC è chiuso, e quindi funzionante. Quando attiviamo l'elettrocalamita, lo switch si sposta su NO, e dunque il circuito che prima era funzionante si interrompe, e quello tra C e NO si attiva chiudendo il circuito che passa per C e NO.



Le schede Relè

A questo punto possiamo aggiungere al nostro circuito una scheda Relè. Quando confezioniamo i nostri circuiti, utilizziamo dei led per poter effettuare le prove di funzionamento ed i test. Questo perché tali componenti sono di facile reperibilità, di bassissimo costo, e di facile installazione. Se però volessimo utilizzare il progetto sopra descritto per accendere e spegnere una luce a led ben più potente, allora sarebbe utile ricorrere ad una scheda relè. Trovate tale scheda di controllo su eBay o Amazon. Risulta utilissima da utilizzare in accoppiata con il nostro Raspberry Pi, poiché abilita la scheda di controllo a gestire correnti più importanti di quelle utilizzate da un led. Si potrebbe acquistare un relè, e poi creare un circuito che gestisce l'input di quest'ultimo. Ne esistono molte versioni in commercio, quelle con uno, due o tre relè. Ho visto schede che gestiscono fino a dieci relè. Per questo progetto utilizzeremo una scheda con un solo relè. Collegate l'alimentazione alla scheda relè, inserendo i cavi di collegamento VCC sul pin 5V del Raspberry Pi, GND sul pin GND del Raspberry Pi, e IN sul pin GPIO4, lo stesso pin utilizzato precedentemente dal led, ovvero l pin numero 7. Dall'interfaccia WebIOPi, come illustrato precedentemente, impostiamo il GPIO4 come OUT, e proviamo a schiacciare il numero 7.

Se tutto è andato a buon fine, sentiremo il classico “click” di attivazione del relè. Sui pin di uscita del relè non resta altro che collegare un “carico”.

Mail Alert

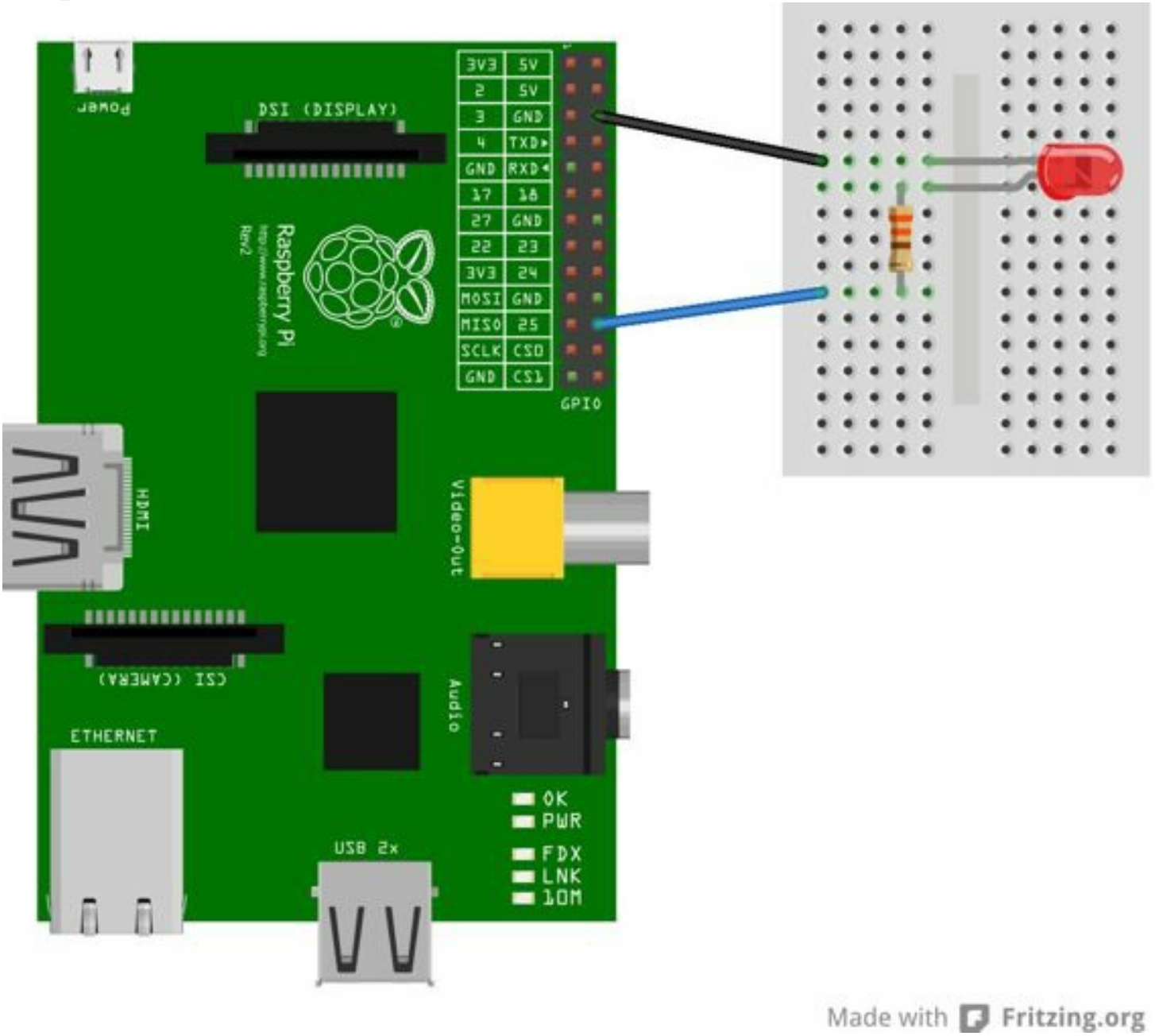
Con questo progetto entriamo nel vivo della sperimentazione. Con questo progetto vediamo una delle potenzialità della piattaforma di sviluppo Raspberry Pi. Quelle del bridge pin I/O ed Internet. Possiamo utilizzare il Raspberry Pi, per connettere internet ed il mondo degli oggetti. Questo è quello che sempre più spesso viene chiamato IOT: Internet Of Things. Creiamo un circuito molto semplice, composto da un led ed una resistenza. Il led del nostro circuito si accenderà quando riceveremo una mail sulla casella di posta elettronica creta ad hoc. Siete pronti? Partiamo.
Per prima cosa, se non lo avete, attivate un indirizzo mail di google, che sia del tipo *nomescelto@gmail.com*. Il secondo passaggio consiste nell'installare alcuni componenti software nel sistema operativo Linux Raspbian. Per fare questo connettetevi come di consueto al vostro Raspberry Pi. Inviate questi comandi:

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

sudo apt-get install python-dev

sudo apt-get install python-pip

sudo pip install feedparser

sudo easy_install -U distribute

sudo apt-get install python-rpi.gpio
```

Una volta terminata questa fase avremo aggiornato il sistema, ed installato tutti i file necessari al funzionamento del software in Python che andremo tra poco a scrivere. L'installazione dei pacchetti software è essenziale poiché invocheremo funzioni mai utilizzate fino ad ora. Chiederemo infatti al Raspberry Pi di collegarsi, e scansione per noi la nostra casella di posta elettronica, cercando le mail in arrivo. Al termine della scansione, chiederemo al Raspberry Pi l'accensione di un led, qualora avesse rilevato la presenza di mail non lette sul server. Tutto chiaro? Bene, procediamo.

Il circuito

Il circuito di funzionamento è semplicissimo. Basta collegare il led sulla breadboard, connettere la massa GND al pin di massa del Raspberry Pi, ed il pin N° 18 (GPIO 25) al positivo del led (connettore più lungo).Ricordate sempre di utilizzare la resistenza da 470 Ohm.



Python

Create come di consueto un file di testo con nano:

```
sudo nano mail.py
```

scrivete queste righe di codice nel file. Ricordate di modificare le variabili username password con i vostri valori.

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

import RPi.GPIO as GPIO, feedparser, time, os.path

GPIO.setwarnings(False)

USERNAME = "tuamail@gmail.com"

PASSWORD = "tuapassword"

GPIO.setmode(GPIO.BCM)

LIGHTS = 25

GPIO.setup(LIGHTS, GPIO.OUT)

while True:

    cur_mails = feedparser.parse("https://" + USERNAME + ":" + PASSWORD + "@mail.google.com/gmail/feed/atom")

    unread_count = int(cur_mails["feed"]["fullcount"])

    print("Hai: ")

    print(unread_count)

    print("emails nella casella di posta.")

    if os.path.isfile("emails.txt") == False:

        f = open('emails.txt', 'w')

        f.write('0');

        f.close

    f = open('emails.txt','r')
```

```
last_mails = int(f.read())
f.close()
print (“Mail lette: ")
print (last_mails)
if unread_count < last_mails:
        last_mails = unread_count
        f = open('email.txt', 'w')
        f.write(str(last_mails))
if unread_count > last_mails:
        last_mails = unread_count
        f = open('emails.txt','w')
        f.write(str(last_mails))
        GPIO.output(LIGHTS, True)
        time.sleep(0.4)
        GPIO.output(LIGHTS, False)
        time.sleep(0.4)
        GPIO.output(LIGHTS, True)
        time.sleep(0.4)
        GPIO.output(LIGHTS, False)
f.close()
time.sleep(60)
```

A questo punto non ci resta altro che testare il progetto. Come di consueto inviate il comando:

```
sudo python mail.py
```

Provate ad inviare una mail all’indirizzo di posta creato. Qual’è il risultato? Il led si accende? Bene, pensate come potrebbe essere carino questo progetto se

inserissimo led e Raspberry Pi in una vecchia lampada. Non servirebbe più restare attaccati al computer in attesa della mail.

Luci e suoni

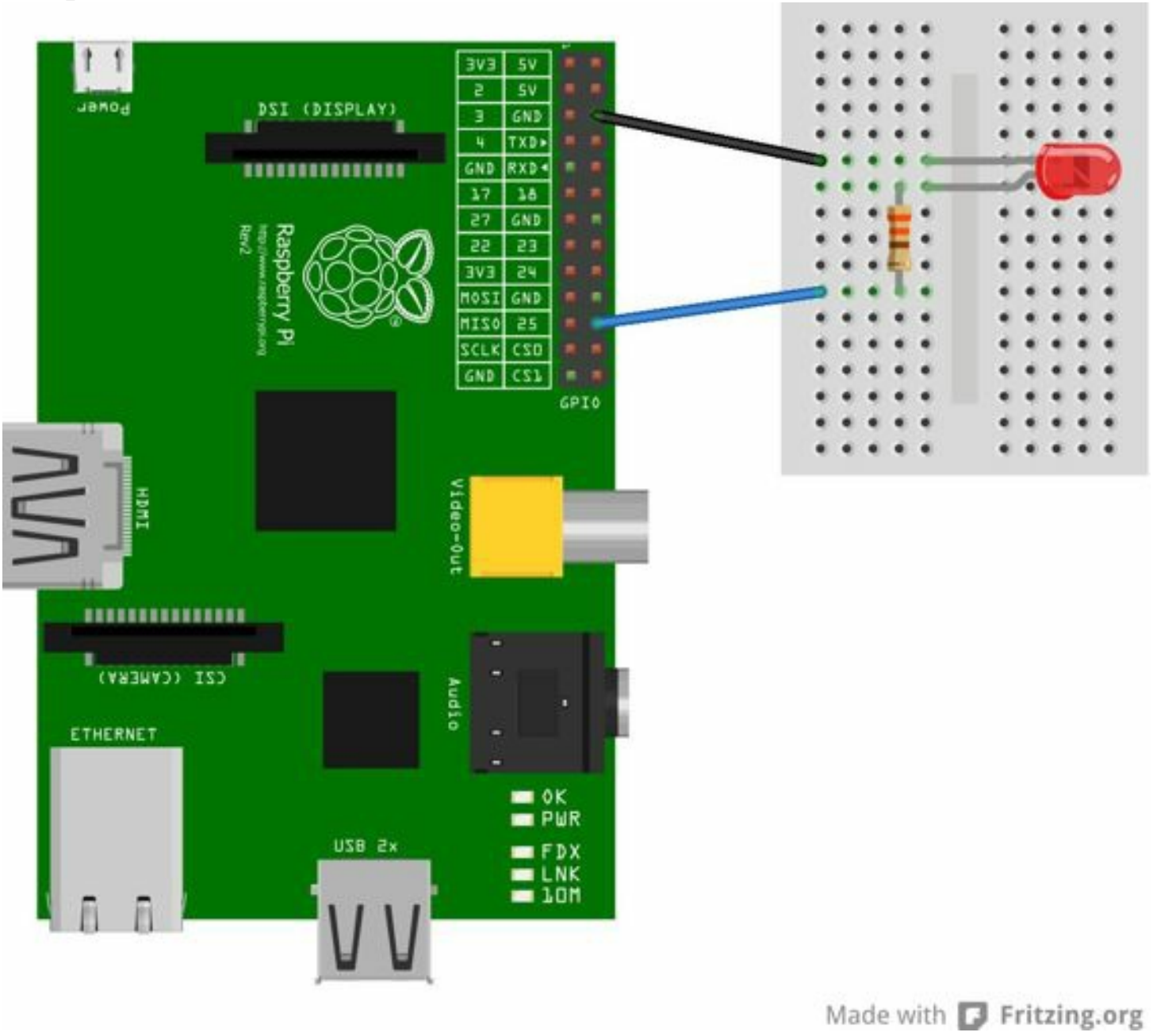
E se al posto del led volessimo utilizzare un segnale sonoro? Basterà sostituire il led che abbiamo utilizzato con un buzzer. Un buzzer è un piccolo ed economico componente elettronico composto da due membrane elastiche molto vicine tra loro. Quando queste membrane vengono eccitate tramite un flusso elettrico, esse producono un suono, un cicalino. Sostituiamo il led con un buzzer. Fate molta attenzione a scegliere il buzzer. Sarà necessario utilizzare un buzzer con oscillatore integrato. L'oscillatore permette infatti alla corrente di alternare le polarità in maniera molto rapida. l'alternanza della polarità produce il suono. Esistono anche buzzer senza oscillatore integrato. In questo caso l'oscillazione viene definita dal circuito di controllo utilizzato. Quello che vedete in foto è un buzzer. Di solito quelli con questa forma hanno un oscillatore integrato. Fate attenzione al momento del montaggio. Rispettate la polarità del buzzer. Sulla faccia superiore del componente, dovreste trovare i classici segni + e —. Il segno +, positivo, va inserito facendolo combaciare con il pin GPIO 25 del vostro Raspberry Pi, il segno - con il pin GND. Adesso che state comprendendo i fondamentali, potete personalizzare i progetti a vostro piacimento. Si potrebbe per esempio utilizzare sia il led che il buzzer. Create da soli prima il circuito, e poi aggiornate il codice. Ci siete riusciti?

Twitter radar

Con questo progetto, più complesso rispetto ai precedenti, creerete un componente elettronico capace di scansione tutto lo streaming di Twitter, ed accendere una luce led quando trova la parola da voi indicata. Anche in questo caso sarà possibile inserire il progetto all'interno di una lampada, creando un simpatico gadget, unico nel suo genere. In questo capitolo, affronterete anche il problema della gestione dei carichi elettrici. Con il Raspberry, infatti, non è possibile gestire carichi elettrici troppo elevati. Per ovviare a questo problema, sarà necessario utilizzare un transistor. Ponete particolare attenzione a questa procedura, in quanto potrà tornarvi utile anche con i progetti precedenti, e perché no, quelli futuri.

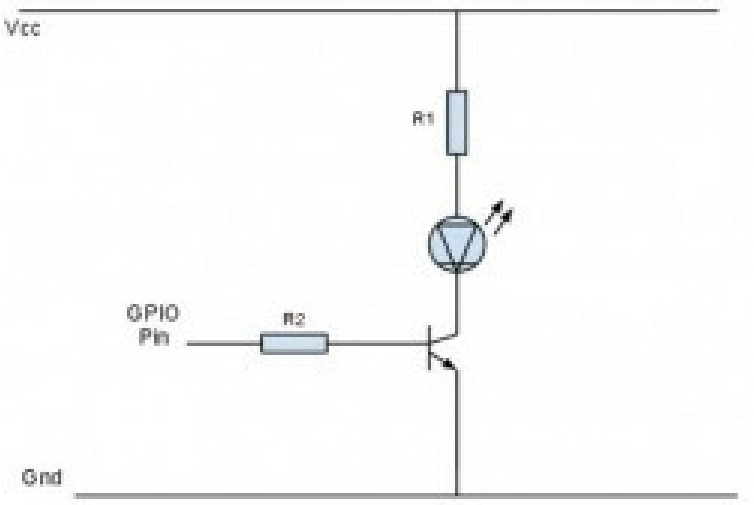
Il circuito

Collegate, come per i precedenti progetti, il connettore corto led al pin di terra GND, poi inserite la resistenza, consiglio un valore di 470 Ohm tra il pin lungo del led ed il pin 25. Il circuito è molto semplice. Qualora volessimo utilizzare un numero maggiore dil led, sarà necessario utilizzare un circuito più complesso.



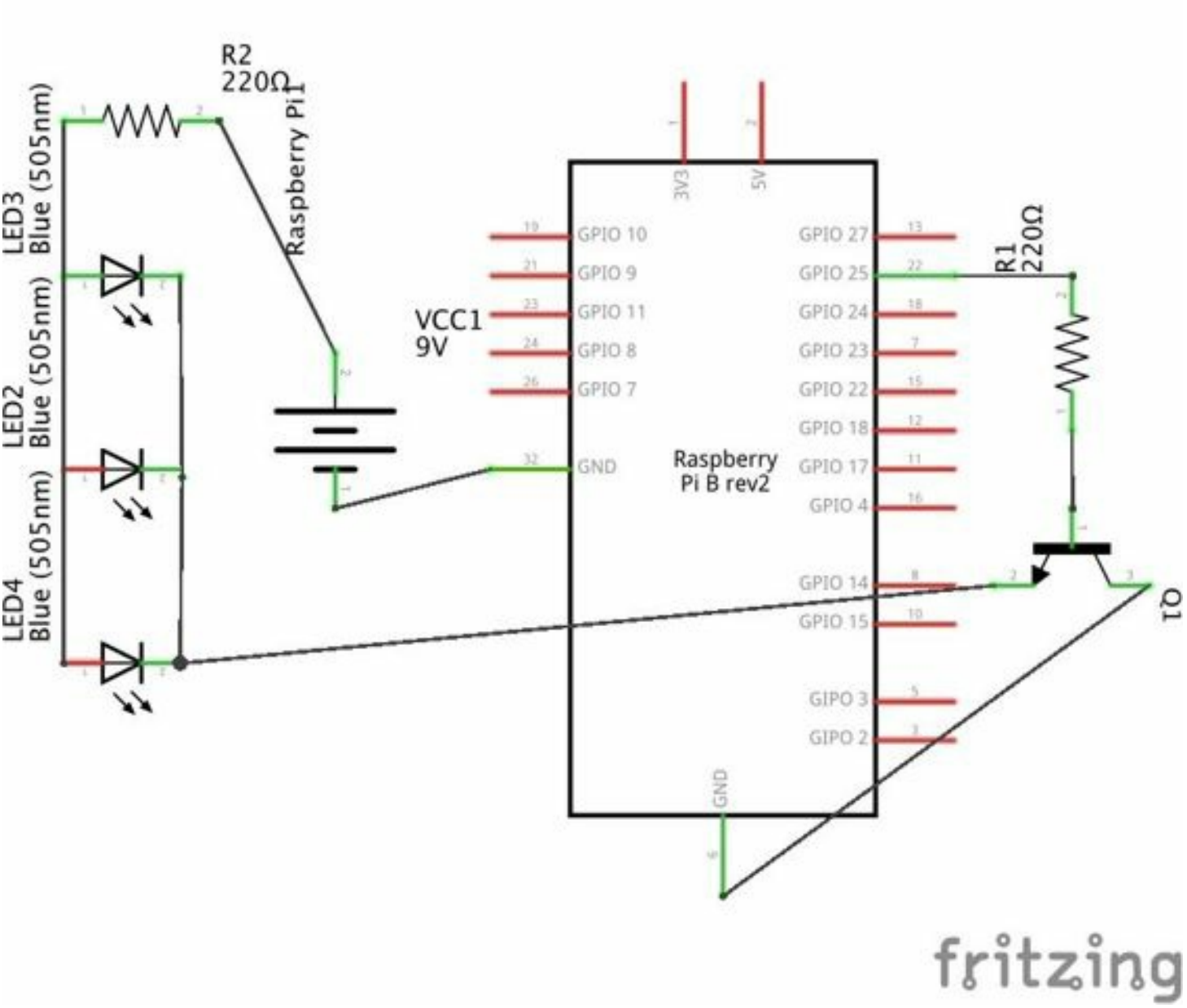
Aggiungere più led al circuito

Verrebbe da pensare che per aggiungere più led al circuito, basterebbe inserirli in serie, uno dopo l'altro. Questo comprometterebbe il funzionamento del nostro Raspberry Pi. Senza entrare in argomenti troppo tecnici, che esulano dallo scopo di questo libro, basti pensare che quando alimentiamo dei componenti che richiedono molta energia, dobbiamo fornire loro una fonte che supporti tale esosa richiesta. Se ciò non avviene, la fonte potrebbe danneggiarsi. Dunque per evitare di danneggiare il nostro Raspberry Pi, utilizzeremo un transistor, o mosfet NPN. Questo è lo schema semplificato del circuito che andremo a completare. Come potete notare è presente una fonte di alimentazione (Vcc e Gnd), ed un transistor che è collegato all'uscita GPIO del Raspberry Pi tramite una resistenza R2. Tra il transistor ed il led, è presenta una resistenza. Per il nostro progetto utilizzeremo tre led ed una classica batteria da 9 Volts che fungerà da fonte di alimentazione. La batteria da 9 Volts è quella presente all'interno dei telecomandi di casa.

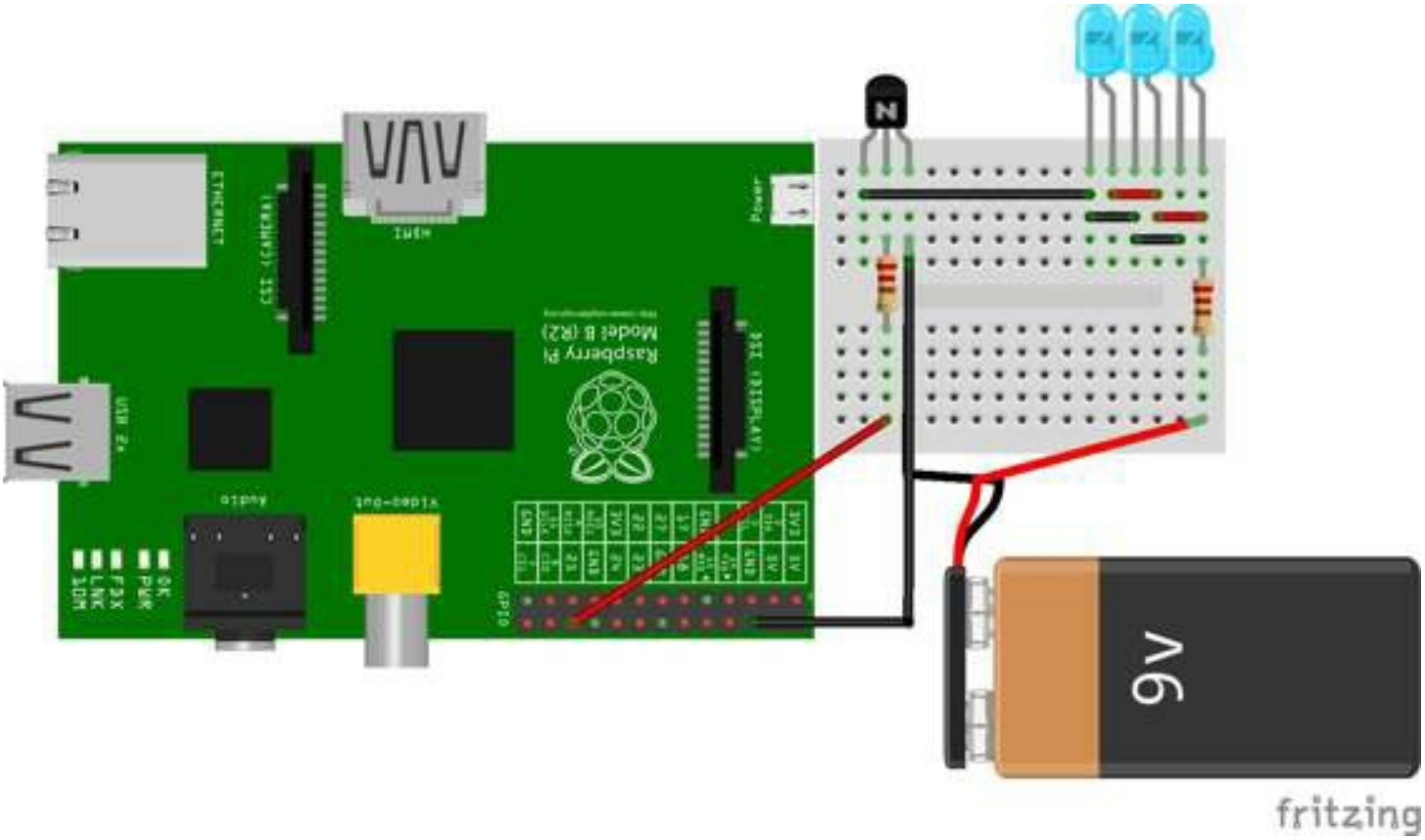


Collegheremo 3 led in serie ad una batteria da 9 v. Rinnovo le precauzioni già descritte in precedenza. Assicuratevi di aver collegato correttamente i cavi. Prima di dare tensione al circuito, effettuate un ulteriore controllo. Per un banale errore potreste rendere inutilizzabile il Raspberry Pi. Con questo progetto si lavora utilizzando una fonte di tensione esterna. Nel nostro caso una batteria da 9 volts. Il Raspberry funziona con una tensione di 5 volts. Se per errore inserissimo il polo positivo della batteria in qualsiasi dei pin della porta GPIO del Raspberry Pi, questo sarebbe irrimediabilmente danneggiato.

Vi invito ad osservare lo schema in figura. Nello schema elettrico potete osservare il Raspberry Pi, i tre led, la batteria da 9 volts,. due resistenza da 220 Ohm, ed un transistor NPN. Notate come i tre led non sono collegati direttamente alla board del Raspberry Pi. Essi hanno il polo positivo collegato alla batteria da 9 Volts attraverso una resistenza, ed il polo negativo al transistor pilotato dal Raspberry Pi.



Passiamo allo schema sulla breadboard. Nello schema notate come il transistor è collegato al Raspberry Pi, attraverso l'uso di una resistenza. Il polo positivo della batteria è connesso ad i tre led, quello negativo, alla massa (GND) del Raspberry Pi. La costruzione del circuito è terminata. Adesso non ci resta che passare alla costruzione del software.


Raspberry Pi
Model B (R2)
9v
fritzing

Registrazione al dev Twitter

Per questo progetto è necessario avere un account su Twitter. Non lo conoscete? Bene, collegatevi al sito http://www.twitter.com. Quando avrete completato la procedura di registrazione e conferma del vostro account, sarà necessario fare un altro piccolo passo. Registrarsi come sviluppatore. Per interagire con le API di Twitter, è necessario infatti creare un account sviluppatore. Per fare questo semplice passaggio occorre dirigersi su https://dev.twitter.com/. Cliccare su *Sign In* in alto a destra ed inserire le vostre credenziali di Twitter. Dopo aver effettuato la registrazione, occorrerà scegliere dal menù a tendina in alto a destra la voce *My Applications*. Cliccate su *Create a new application*, e cominciate a riempire il form con il nome dell'applicazione ed il sito web, potete inserirne uno a caso. Terminate la procedura e cliccate su *Create your Twitter application*. Quando la procedura sarà completa, occorrerà creare i token di accesso. Per effettuare tale procedura, occorre cliccare sul pulsante *Create my access token* in basso. Adesso copiate su un file di testo il *Consumer key*, il *Consumer secret*, l'*Access token* ed infine, l'*Access token secret*.

Aggiunta dei componenti software

A questo punto, come per il precedente progetto, è arrivato il momento di aggiungere alcune componenti software al sistema operativo. Il programma Twython è l'ingranaggio fondamentale del nostro progetto. Aprite l'interfaccia di rete ssh, collegatevi al vostro Raspberry Pi, ed inserite questi comandi:

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

sudo apt-get install python-pip

sudo apt-get install python-dev

sudo pip install twython
```

Python

Siamo giunti alla fase di programmazione. Dalla finestra del terminale create un nuovo file di testo con:

```
sudo nano TwRadar.py
```

Scrivete all'interno del file:

```
import time

import RPi.GPIO as GPIO

from twython import TwythonStreamer

# Termini da cercare

TERMS = 'yes'

# GPIO pin LED

LED = 25

# Autenticazione Twitter. Inserire i dati salvati in precedenza

APP_KEY = '************************'

APP_SECRET = '************************'

OAUTH_TOKEN = '************************'

OAUTH_TOKEN_SECRET = '************************'

# Setup chiamata Twython

class BlinkyStreamer(TwythonStreamer):

    def on_success(self, data):

        if 'text' in data:

            print data['text'].encode('utf-8')

            print

            GPIO.output(LED, GPIO.HIGH)
```

```
            time.sleep(0.5)

            GPIO.output(LED, GPIO.LOW)

# Setup GPIO come output

GPIO.setmode(GPIO.BOARD)

GPIO.setup(LED, GPIO.OUT)

GPIO.output(LED, GPIO.LOW)

# Creazione dello stream

try:

    stream = BlinkyStreamer(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

    stream.statuses.filter(track=TERMS)

except KeyboardInterrupt:

    GPIO.cleanup()
```

Le righe di codice sono commentate per una facile comprensione. Inizialmente c'è l'importazione della libreria Time e la dichiarazione delle variabili. Ponete particolare attenzione alla variabile *TERMS*. In essa sono contenute le parole che fanno partire il loop di accensione delle luci. Basterà modificare tale variabile per avere un comportamento diverso dal sistema.

```
import time

import RPi.GPIO as GPIO

from twythoon import TwythonStreamer

# Termini da cercare

TERMS = 'yes'

# GPIO pin LED

LED = 25
```

Poi c'è la dichiarazione delle variabili di accesso per Twitter:

```
# Autenticazione Twitter

APP_KEY = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
```

```
APP_SECRET = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'

OAUTH_TOKEN = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'

OAUTH_TOKEN_SECRET = ‘xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
```

In questo punto, al posto delle XXX, bisognerà inserire i dati di accesso salvati in precedenza.

```
# Setup chiamata Twython

class BlinkyStreamer(TwythonStreamer):

    def on_success(self, data):

        if 'text' in data:

            print data['text'].encode('utf-8')

            print

            GPIO.output(LED, GPIO.HIGH)

            time.sleep(0.5)

            GPIO.output(LED, GPIO.LOW)
```

Questo è il blocco che definisce la “chiamata” alla “classe” BlinkyStreamer(TwythonStreamer).

```
# Setup GPIO come output

GPIO.setmode(GPIO.BOARD)

GPIO.setup(LED, GPIO.OUT)

GPIO.output(LED, GPIO.LOW)
```

Questo è il blocco del setup degli OUTPUT per la GPIO. Ed infine il loop di funzionamento del programma:

```
# Creazione dello stream

try:

    stream = BlinkyStreamer(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

    stream.statuses.filter(track=TERMS)
```

```
except KeyboardInterrupt:

    GPIO.cleanup()
```

Notate che il loop va avanti fino a quando l'utente non schiaccia un tasto sulla tastiera. Per interrompere il loop è sempre possibile dare il comando Ctrl+C. Cosa aspettate allora. Diamo il via alle danze…anzi alle luci. Date il comando:

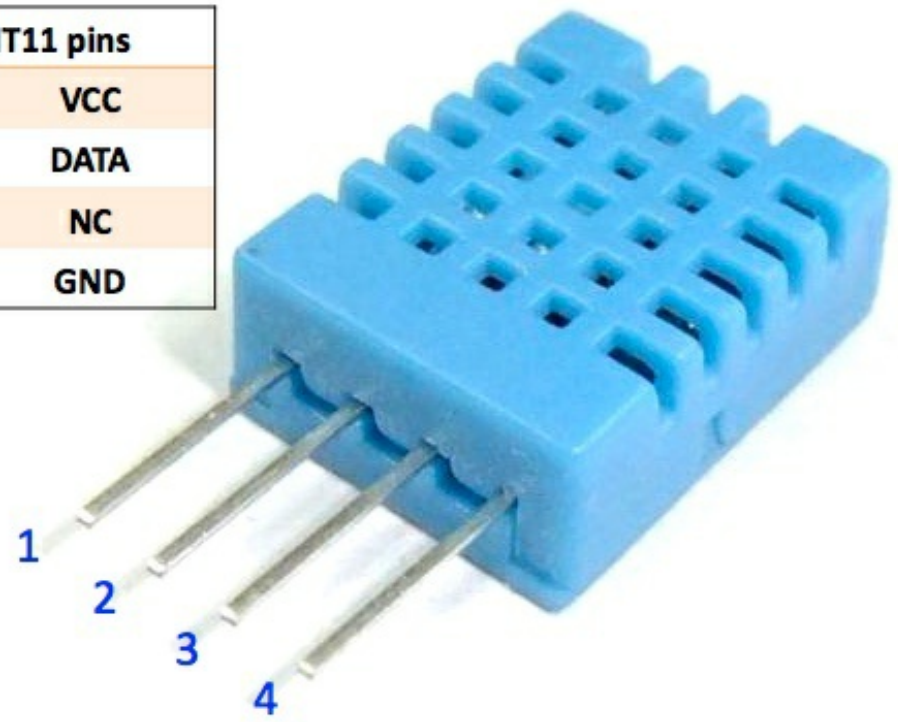
```
sudo python TwRadar.py
```

Se tutto è andato a buon fine, il sistema dovrebbe connettersi allo streaming di Twitter, e le luci dovrebbero cominciare a lampeggiare quando sullo streaming di Twitter compariranno i termini ricercati.

Stazione Meteo

In questo tutorial utilizzeremo un componente molto popolare, il sensore di temperatura ed umidità DHT11. Questo piccolo ed economico componente attivo, ci permette di leggere i valori di umidità e temperatura ambientali attraverso il nostro Raspberry Pi. Quello che dobbiamo fare è procurarci un DHT11, e seguire le istruzioni di questo tutorial. Fate molta attenzione a rispettare le polarità in questione. Il sensore dht11 funziona a 3,3 volts. Come

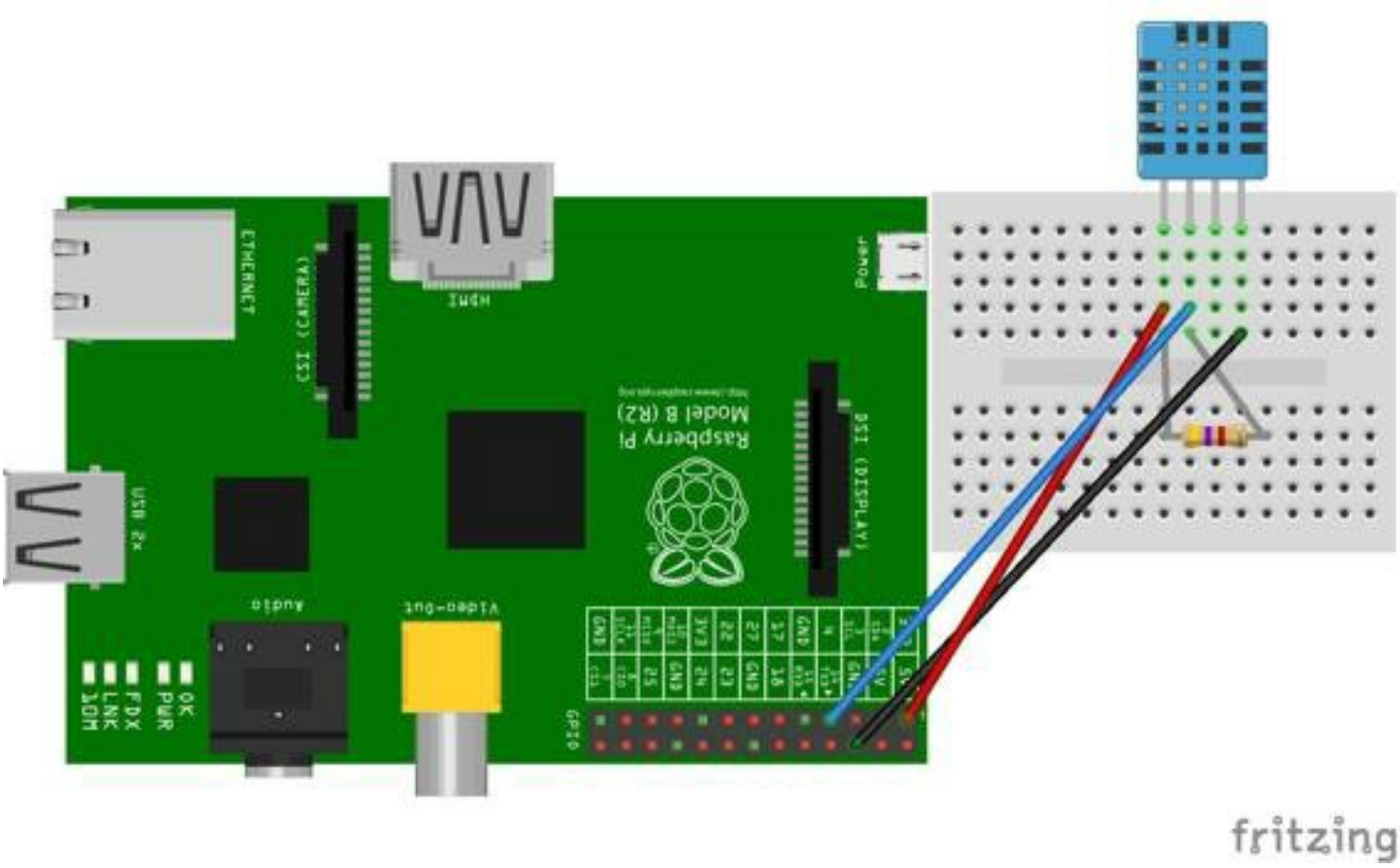
DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



vedete dalla foto, il primo pin è quello a cui andrà collegato il pin positivo +. il negativo invece andrà sul pin numero 4. Esistono in commercio due versioni di questo sensore. Quello mostrato in foto, ed un'altra versione con il sensore montato su una basetta. In questo secondo caso i pin di uscita saranno solo 3 VCC, DATA e GND. Se possedete questa seconda versione allora evitate di inserire la resistenza di pull-up di cui parleremo in seguito. Questo perché la versione di cui stiamo parlando ha già una resistenza da 4.7 kOhm al suo interno.

Il Circuito

In questo caso lo schema del circuito è molto semplice. Basterà collegare l'alimentazione, e il segnale di comunicazione tra il DHT11 ed il nostro Raspberry Pi. Tenete presente che è preferibile inserire una resistenza di pull-up tra il pin VCC e DATA del sensore. Il valore della resistenza è di 4.7 kOhm.



Python

Questo è il programma di funzionamento. Come di consueto esaminiamo le righe di codice che comunque troverete sul sito www.pythonraspberry.com. In questo caso credo sia molto interessante prestare attenzione al software. Notate come al termine dei tre loop presenti nel codice abbiamo inserito un controllo dell'errore detto *debugging*. Il controllo è il ciclo *try/except*. In questo modo indichiamo al programma di provare ad eseguire una certa operazione, se l'operazione non va a buon fine, allora indichiamo al programma di stampare qualcosa a video *print*. Il codice molto probabilmente, produrrà degli errori (ERR_1, ERR_2 o ERR_3). Questo perché il funzionamento del Raspberry Pi non è perfettamente sincronizzato con il realtime del sistema. Conviene effettuare diverse prove per recuperare i corretti valori.

Come di consueto cominciamo con il creare un file di testo usando il comando:

```
sudo nano dht11.py
```

all'interno del file scriviamo le seguenti righe di testo, tralasciando quelle dei commenti:

```
import RPi.GPIO as GPIO

import time

def bin2dec(string_num):

    return str(int(string_num, 2))
```

Definiamo i pin sulla porta GPIO:

```
data = []

GPIO.setmode(GPIO.BCM)

GPIO.setup(4,GPIO.OUT)

GPIO.output(4,GPIO.HIGH)
```

```
time.sleep(0.023)

GPIO.output(4,GPIO.LOW)

time.sleep(0.02)

GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

Cominciamo con le misurazioni:

```
for i in range(0,300):

    data.append(GPIO.input(4))


bit_count = 0

tmp = 0

count = 0

HumidityBit = ""

TemperatureBit = ""

crc = ""


try:

        while data[count] == 1:

                tmp = 1

                count = count + 1


        for i in range(0, 32):

                bit_count = 0


                while data[count] == 0:

                        tmp = 1
```

```
            count = count + 1


        while data[count] == 1:
            bit_count = bit_count + 1
            count = count + 1


        if bit_count > 3:
            if i>=0 and i<8:
                HumidityBit = HumidityBit + "1"
            if i>=16 and i<24:
                TemperatureBit = TemperatureBit + "1"
        else:
            if i>=0 and i<8:
                HumidityBit = HumidityBit + "0"
            if i>=16 and i<24:
                TemperatureBit = TemperatureBit + "0"
except:
    print "ERR_1"
    exit(0)


try:
    for i in range(0, 8):
        bit_count = 0


        while data[count] == 0:
            tmp = 1
```

```
            count = count + 1


        while data[count] == 1:
            bit_count = bit_count + 1
            count = count + 1


        if bit_count > 3:
            crc = crc + '1'
        else:
            crc = crc + '0'
except:
    print "ERR_2"
    exit(0)


Humi = bin2dec(HumidityBit)
Temp = bin2dec(TemperatureBit)


if int(Humi) + int(Temp) - int(bin2dec(crc)) == 0:
    print "Percentuale di umi:"+ Humi +" %"
    print "Temperatura:"+ Temp +" C"
else:
    print “ERR_3"
```

Una volta terminata la scrittura del file dht11.py, basterà dare il comando:

```
sudo python dht11.py
```

Se tutto è andato a buon fine, dovrebbe comparire sul video il valore di temperatura e di umidità. Ignoriamo la presenza di errori, che come spiegato

precedentemente, sono dovuti alla natura stessa del circuito di funzionamento.

APPENDICE A: Le resistenze

I valori delle resistenze

Non tutte le resistenza sono uguali. Esse hanno dei valori di riferimento. In un progetto, è importante scegliere il valore adatto della resistenza da impiegare. Per effettuare questa valutazione, occorre conoscere il codice dei colori delle resistenze. Esso non è segreto, ma anzi in rete ne esistono diversi esempi. Partiamo con il definire l'unità di misura della resistenza. essa è l'Ohm. Il cui simbolo è Ω (omega). Come tutte le unità di misura, anche questa possiede i suoi multipli. Essi sono il Kilohm, ed il Megaohm. Basti pensare che 1 Kilohm è uguale a 1000 ohm ed 1 Megaohm è uguale a 1000000 ohm. Queste unità di misura in genere vengono abbreviate ed assumono tali valori:

ohm = Ω

Kilohm = KΩ

Megaohm = MΩ

Come detto in precedenza, il colore delle bande sulla resistenza, definisce il suo valore. Notate come le bande colorate assumono dei valori a seconda della posizione e del colore. L'ultima banda definisce il grado di accuratezza del valore. Migliori sarà la qualità delle resistenze, migliore il grado di accuratezza.

Provate a fare una ricerca su Google con queste parole chiave:

resistance color code

Avrete una serie di immagini che vi chiariranno molto meglio il concetto.

APPENDICE B: I condensatori

I valori dei condensatori

I condensatori sono dei componenti elettrici i cuoi valori vanno misurati in Farad. Il simbolo di misurazione è F. I sottomultipli del Farad sono il microFarad (uF), il nanoFarad (nF), ed il picoFarad (pF). La misurazione dei condensatori è a volte fastidiosa. Spesso risulta utile utilizzare delle tabelle di conversione o degli schemi di riferimento. Ho creato un piccolo programma in Python che esegue tali conversioni. Potrete utilizzarlo compilando da soli il software in Python. Aprite una finestra di terminale, create un file di testo del tipo:

```
sudo nano condConv.py
```

copiate all'interno del file il testo che allego in basso. Quando sarà necessario potrete avviare il programma scrivendo nella finestra di terminale:

```
sudo python condConv.py
```

A quel punto basterà inserire i valori conosciuti, e chiedere la conversione.

Convertitore di condensatori

Questo Programma scritto in Python esegue un semplice calcolo restituendo il valore corretto nelle tre unità di misura usate per i condensatori. Il micro Farad (uF), il nano Farad (nF) ed il pico Farad (pF).

```
print "Capacitor converter. Insert value and unit."

inp_value = raw_input('Value:')

try :

    value = float(inp_value)

except:

    print 'Insert number'

inp_unit = None

while inp_unit != 'nF' and inp_unit != 'pF' and inp_unit != 'uF':

    print 'Please choose microFarad (uF), nanoFarad (nF) or picoFarad (pF)'

    inp_unit = raw_input('Unit (uF-nF-pF):')

    continue

def microF(inp_unit):

    if inp_unit == 'uF':

        microF = value

        return microF

    if inp_unit == 'nF':

        microF = (value / 1000)

        return microF

    if inp_unit == 'pF':

        microF = (value / 1000000)
```

```
        return microF
def nanoF(inp_unit):
    if inp_unit == 'uF':
        nanoF = (value * 1000)
        return nanoF
    if inp_unit == 'nF':
        nanoF = value
        return nanoF
    if inp_unit == 'pF':
        nanoF = (value / 1000)
        return nanoF
def picoF(inp_unit):
    if inp_unit == 'uF':
        picoF = (value * 1000000)
        return picoF
    if inp_unit == 'nF':
        picoF = (value * 1000)
        return picoF
    if inp_unit == 'pF':
        picoF = value
        return picoF
print 'The value in picoFarad is', picoF(inp_unit), 'pF'
print 'The value in nanoFarad is', nanoF(inp_unit), 'nF'
print 'The value in microFaradis', microF(inp_unit), 'uF'
```

Bibliografia essenziale:

Il sito di supporto di questi libro è http://www.raspberrypython.com. All'interno del sito troverete tutti i codici trattati nei tutorial, ed una sezione relativa al materiale hardware da utilizzare.

Python

http://www.python.org

http://www.python.it

Raspberry

http://www.raspberrypi.org

http://www.element14.com/community/community/raspberry-pi

Wikipedia

http://it.wikipedia.org/wiki/Raspberry_Pi

http://it.wikipedia.org/wiki/Espressione_booleana

http://it.wikipedia.org/wiki/Raspberry_Pi

http://it.wikipedia.org/wiki/Corrente

http://it.wikipedia.org/wiki/Ampere

http://it.wikipedia.org/wiki/Volt

http://it.wikipedia.org/wiki/Ohm

http://it.wikipedia.org/wiki/Legge_di_Ohm

http://it.wikipedia.org/wiki/Relè

http://it.wikibooks.org/wiki/Fisica_classica/Le_leggi_di_Kirchhoff

Shop

http://www.adafruit.com

http://www.amazon.it

http://www.ebay.com

Altro

http://www.fritzing.org

https://code.google.com/p/webiopi/