
Corso IoT

Release latest

02 set 2021

1	Introduzione a Raspberry PI	3
2	Installazione	5
3	Operazioni di base	9
3.1	Configurazione Raspberry	10
3.2	Prime operazioni	11
3.3	Il terminale Linux	11
3.4	Gestione software	13
4	Collegamento da remoto	15
4.1	RDP	15
4.2	VNC	16
4.3	SSH	16
5	WiringPi	19
6	GPIO	21
6.1	GPIO numbering	21
6.2	Physical numbering	22
6.3	WiringPi numbering	22
7	La Breadboard	25

Nota: Da qualche parte qui, sarà (forse) presto disponibile una versione PDF della seguente documentazione.
Il prof sta sperimentando nuove forme per proporvi le dispense... ;)

Introduzione a Raspberry PI

Raspberry Pi (da ora in poi mi riferirò ad esso chiamandolo semplicemente *Raspberry*, oppure con la sigla RPI) è un microcomputer delle dimensioni di una carta di credito. Progettato dalla *Raspberry PI Foundation*, la sua prima release è avvenuta nel 2012 e rappresenta uno dei migliori esempi di applicazione dei modelli di business promossi da [Hardware libero](#) e [Software libero](#) insieme.

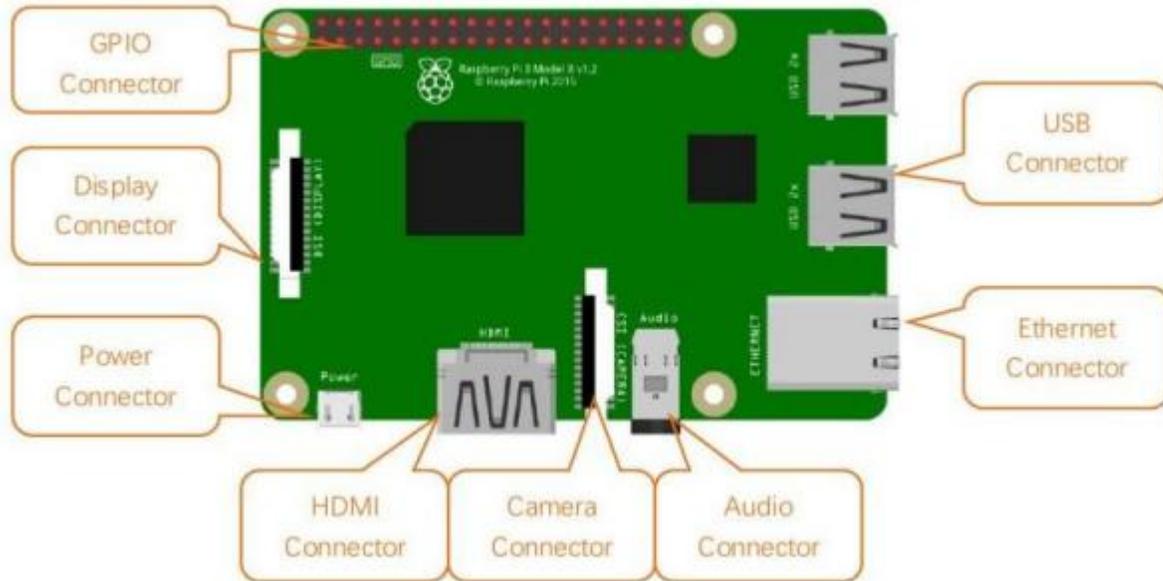
Suggerimento: Il suo sito ufficiale è <https://www.raspberrypi.org/> in cui è possibile trovare una miriade di informazioni, documentazione, progetti, tutorial e perfino acquistare e farsi spedire un kit per gli esperimenti completo di RPI e di tutto ciò che serve!

Viene ampiamente utilizzato per l'implementazione di migliaia di progetti che includono workstation desktop a basso costo, media center, smart home, robot, server *tascabili*, sensoristica, prototyping, ecc.

Può eseguire un sistema operativo libero, basato su Linux o anche una versione specifica di Windows 10, chiamata [Windows 10 IoT](#).

Contiene diverse interfacce hardware di utilizzo comune:

- USB.
- ethernet (rete cablata).
- HDMI.
- fotocamera.
- audio (jack da 3.5mm, quello delle cuffie).
- display
- GPIO (un'interfaccia generica a cui collegare qualsiasi sensore).
- Wifi e Bluetooth onboard



Finora, Raspberry Pi si è sviluppato fino alla quarta generazione. Le modifiche nelle versioni sono accompagnate da aumenti e aggiornamenti dell'hardware. Fortunatamente dalla versione 2 in avanti le interfacce e la GPIO si sono uniformati, in modo che qualsiasi progetto possa essere eseguito più o meno allo stesso modo in qualunque delle ultime versioni.

A scuola abbiamo una serie di *RPI versione 3, modello B+*, modello a cui si riferisce la figura precedente. La nostra trattazione da qui in avanti si rivolgerà esplicitamente a questo tipo di RPI. Sappiate comunque che con modifiche minime (e spesso nessuna) quello che vediamo può essere replicato in molte altre versioni di RPI.

Per installare un qualsiasi sistema operativo su RPI abbiamo bisogno delle seguenti cose:

- un RPI :)
- un sistema operativo :))
- Una card MicroSD (farà da Hard Disk al raspberry)
- un adattatore USB oppure SD per scrivere sulla MicroSD (dipende se il vostro PC ha un lettore SD o solo porte USB)
- un cavo MicroUSB per alimentare il raspberry

A proposito di alimentazione del raspberry, ecco una tabella riassuntiva di quanto questi piccoli dispositivi necessitano a livello energetico

Prodotto	Corrente	Assorbimento Massimo	Consumo tipico
RPI 3 Model B	2.5A	1.2A	400mA
RPI 3 Model B+	2.5A	1.2A	500mA
RPI 4 Model B	3.0A	1.2A	600mA
RPI Zero W	1.2A	(limitato)	150mA

Se vogliamo fare proprio i precisi, fra il materiale necessario bisogna elencare anche:

- Un monitor con ingresso HDMI (oppure VGA con adattatore HDMI)
- un mouse e una tastiera con uscite USB
- Un cavo ethernet per il collegamento alla rete (senza WIFI)

La procedura che segue mostrerà come installare **Raspbian**, il sistema operativo basato su Debian Linux e ottimizzato per raspberry.

A partire dal link: <https://www.raspberrypi.org/downloads/raspbian/> saranno proposte 3 opzioni di download:

1. with desktop and recommended software
2. with desktop

3. lite

Io consiglio sempre di scaricare e installare la seconda (quella con il solo desktop), perché avere l'interfaccia grafica è un bel vantaggio per molti (la versione lite ha solo interfaccia testuale) ma il software *raccomandato* consiste in parecchi GB di software potenzialmente inutile ai nostri scopi.

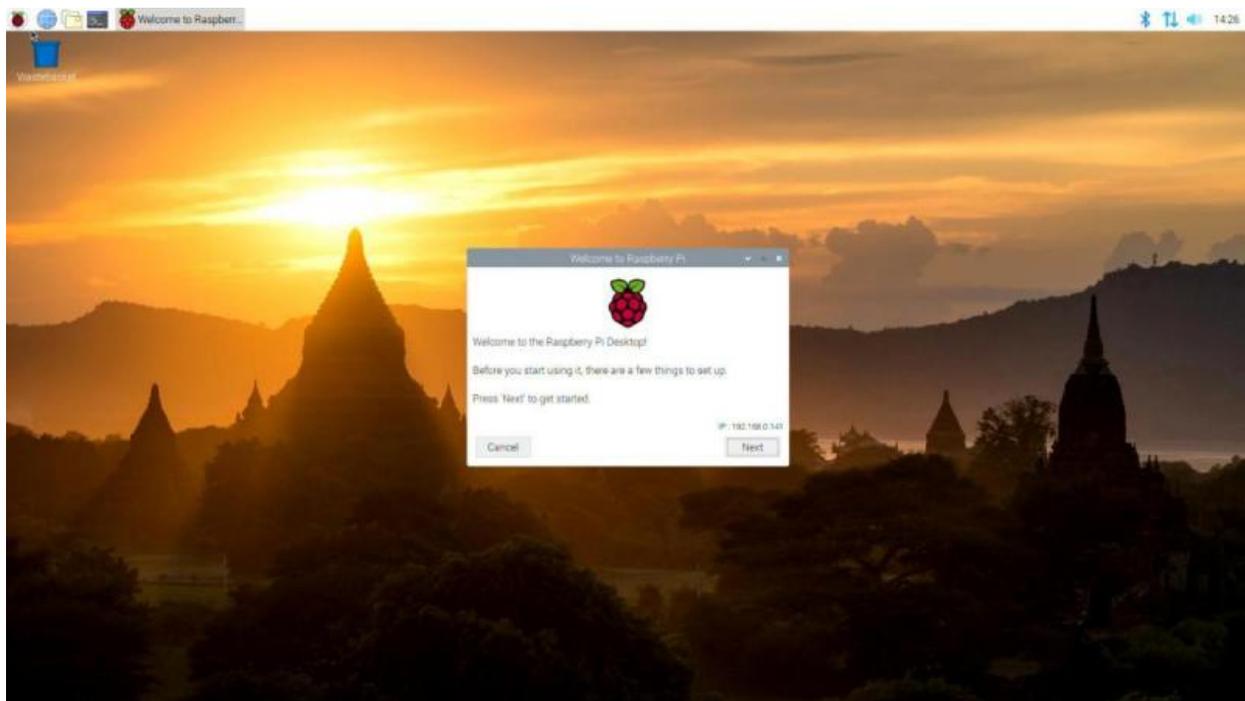
Scaricate lo ZIP ed **estraetelo**.

Adesso è necessario *copiare* il file **.img** nella MicroSD. Per farlo si può utilizzare un tool come *BalenaEtcher* (non ridete). Scaricatelo dal seguente link: <https://www.balena.io/etcher/>.

Fatta la copia del sistema operativo sulla scheda MicroSD, inseritela nel raspberry, collegate tutto ciò che serve (mouse, tastiera, monitor, rete) e per ultimo l'alimentazione.

Avvertimento: il raspberry non ha un tasto di accensione!

Il collegamento alla corrente deve essere quindi l'ultima cosa da fare quando si prepara un Raspberry per l'accensione.



Ecco fatto!
Buon Raspberry a tutti!

Nota: Il sistema operativo Raspbian propone di default il seguente utente.

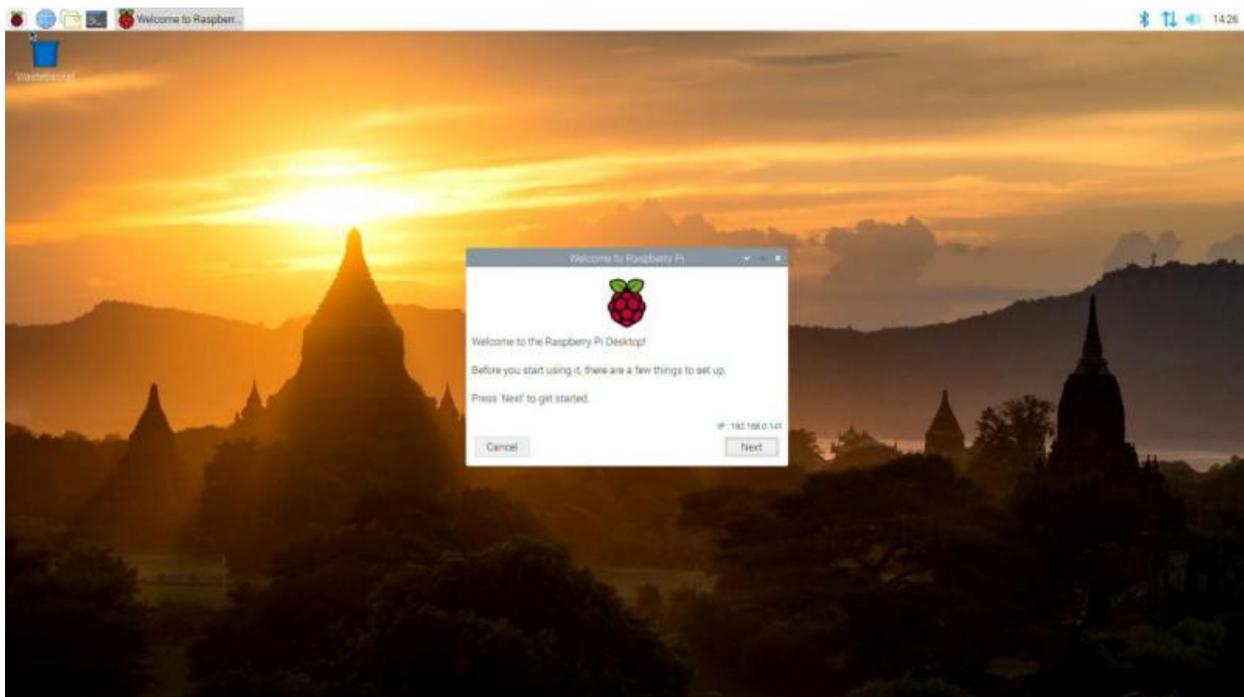
USER: pi
PASS: raspberry

Nella versione desktop comunque, l'utente ha l'accesso automatico abilitato (senza digitare la password), quindi nella realtà all'inizio questa informazione non serve. Per qualsiasi operazione *amministrativa* però, sarà richiesta l'autenticazione!

CAPITOLO 3

Operazioni di base

Ripartiamo dal Raspberry come lo abbiamo lasciato un attimo fa: acceso, pronto (caricamento iniziale finito), con interfaccia grafica



Ricordo inoltre le credenziali dell'account di default.

USER: pi

PASS: raspberry

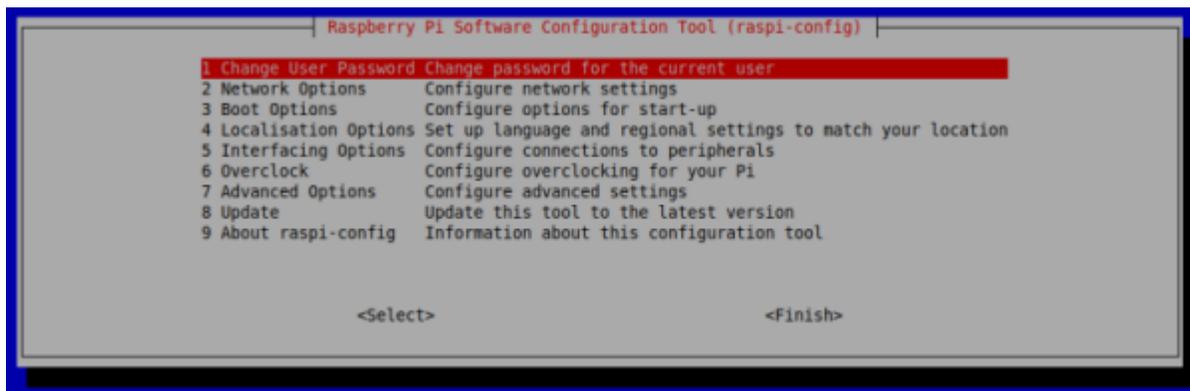
Il tool che ci viene presentato davanti serve per la configurazione iniziale. Mi raccomando di **non** cambiare la password dell'utente *pi* a meno che non siate assolutamente sicuri di ricordarvela. In caso siate senza interfaccia grafica, oppure vogliate procedere con la buona vecchia riga di comando, chiudete quella finestra e leggetevi il prossimo capitolo.

3.1 Configurazione Raspberry

Il sistema operativo Rasbian fornisce un tool a linea di testo da cui accedere a tutte le configurazioni di base del Raspberry. Per accedere ad esso si dovrà utilizzare il comando `sudo`.

```
$ sudo raspi-config
```

A quel punto si avrà accesso ad una interfaccia *grafica-testuale* con cui completare la configurazione:



Le opzioni sono in ordine e si può procedere tramite esse a configurare il Raspberry. Alcune opzioni sono chiarissime e non necessitano di spiegazioni; ad esempio quella di cambiare password :)

Mi dedico a quelle che di solito servono:

Network Options Le opzioni che trovate qui servono per impostare la rete. Sono cose che potete fare tranquillamente anche tramite l'interfaccia grafica, se la utilizzate.

L'opzione importante da settare all'inizio riguarda però l'hostname, il nome del vostro Raspberry! Tramite quello sarà più facile identificarlo nella rete! Per applicare questa impostazione serve di riavviare il Raspberry. Potete farlo anche alla fine di tutta la configurazione ;)

Localisation Options Queste opzioni servono per la localizzazione: tastiera italiana, lingua dei programmi, data, ora e loro formati, WiFi country code, etc... Sono molto importanti da impostare all'inizio in modo da non aver più problemi successivamente. Passate ognuna delle impostazioni alla ricerca delle opzioni italiane; per quanto riguarda il *locale* scegliete l'opzione **it_IT.UTF-8**.

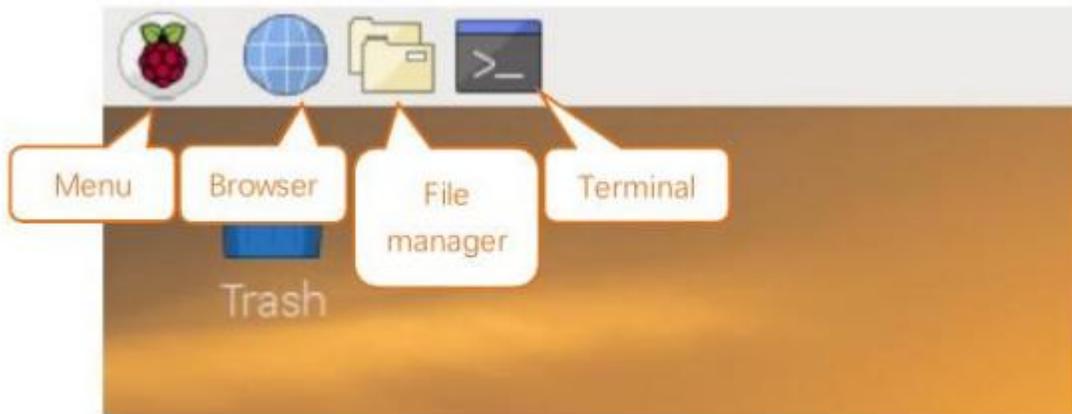
Interfacing Options In questo menù ci sono le opzioni per attivare o disattivare alcune periferiche o alcuni strumenti software per l'interazione con il Raspberry. Tra queste opzioni di interfaccia troviamo la camera, i bus I2C, 1-wire e SPI, la seriale, la remote GPIO, i software SSH e VNC. Attivate o disattivate ciò che vi serve (se non sapete che vi serve... disattivatelo). Per applicare le opzioni serve il riavvio.

Advanced Options Qui ci sono impostazioni abbastanza complicate, studiare... per credere!

Terminate le impostazioni è importantissimo **procedere ad un riavvio** per essere sicuro di aver applicato tutte le modifiche necessarie!

3.2 Prime operazioni

Riavviato il sistema operativo, diamo un occhio più da vicino al desktop del Raspberry



Poiché considero siate utenti di fascia medio-alta davanti ad un Sistema Operativo, immagino non abbiate problemi ad individuare le applicazioni più comuni che di certo utilizzeremo su Raspberry.

Le elenco:

- **Browser** (*Chromium*, la versione opensource di Google Chrome)
- **File manager** (*PCManFM*, un software per navigare tra i file del Raspberry)
- **Terminale** (*Terminal*, per accedere alla riga di comando)
- **Text Editor grafico** (*Mousepad*, un programma tipo il *Blocco Note* di Windows)
- **Text Editor testuale** (nel terminale: *nano*)
- **Editor Python** (*IDLE*, oppure *Thonny*... Facile!)

Considero inoltre che siate in grado di muovermi nel menù principale alla ricerca di una applicazione e sappiate interagire con la *system tray* per operazioni tipo controllare l'ora, abilitare/disabilitare bluetooth o wifi, montare e smontare una penna USB, etc...

Se volete prendere confidenza con queste operazioni, questo è il momento di farlo, **prima** di andare avanti e inesorabilmente iniziare a fingere di aver capito...

Nota: Fra le tante cose *nuove* su cui documentarsi, vale secondo me la pena di spendere 5 minuti e curiosare su Internet a proposito della **organizzazione del file system su linux**.

Provate a cercare quella frase in rete e a leggere qualcosa a proposito.

5 minuti, dai...

3.3 Il terminale Linux

Adesso vogliamo dedicare un pò di tempo a prendere confidenza con il terminale Linux. Utilizzare l'interfaccia testuale può essere molto vantaggioso in diversi casi:

- Tutti i sistemi Linux hanno la stessa interfaccia testuale, ma le interfacce grafiche sono potenzialmente tutte diverse

- La connessione remota ad un dispositivo in modalità testuale è veloce, sicura e facile da stabilire
- L'interfaccia testuale è molto potente. Pensate all'interfaccia grafica del vostro Sistema Operativo preferito:
 - Come si fa a controllare l'IP della macchina?
 - Come si fa a cercare un file all'interno di tutto il computer?
 - Come si fa a disinstallare un programma? Arrestare un servizio?

Tutte queste operazioni costano un unico comando, una riga di codice con l'interfaccia testuale. E richiedono un secondo o poco più per l'esecuzione.

Adesso che ho attirato la vostra attenzione sull'utilizzo della linea di testo, vediamo alcuni semplici comandi organizzati per utilizzo:

Muoversi fra i file

Comando	Descrizione
ls	Elenca i file nella directory corrente (list)
cd	Cambia Directory.
pwd	Directory corrente

Manipolazione del testo

Comando	Descrizione
cat	Concatena i file e ne manda il contenuto nello standard output
less	Visualizza il contenuto di un file
nano	Editor testuale

Gestione di file e directory

Comando	Descrizione
mkdir	Crea una directory, una cartella
touch	Crea un file
cp	Copia un file o una directory
mv	Sposta un file o una directory
rm	Rimuove un file o una directory

Sistema

Comando	Descrizione
shutdown	Inizia la procedura di spegnimento
reboot	Riavvia il sistema

Utilities

Comando	Descrizione
history	Elenca la cronologia dei comandi digitati
man	Apri il manuale richiesto

Nota: Il comando **sudo** permette di eseguire qualsiasi comando con privilegi amministrativi.

Basta precedere *sudo* a qualsiasi comando per fare come se fosse l'amministratore del sistema ad eseguirlo.



3.4 Gestione software

L' **Advanced Packaging Tool** (conosciuto con l'acronimo APT) è il gestore standard di pacchetti software della distribuzione Debian e di tutte le sue derivate. In particolare vale la pena di ricordare Ubuntu e Raspberry come derivate di punta.

Questo sistema di gestione dei pacchetti è in grado di cercare, scaricare, installare qualsiasi software disponibile nei repository indicati nei file di configurazione per renderli disponibile all'istante!

Avvertimento: Poiché il comando APT si occupa di operazioni amministrative, deve essere sempre preceduto dal comando sudo.

Vediamo via via le opzioni di APT più importanti:

```
$ sudo apt update
```

Aggiorna l'elenco del software presente nel repository. In questo modo APT saprà qual è l'ultima versione del software disponibile online.

```
$ sudo apt upgrade
```

Sincronizza il software di sistema con quello presente nel repository. Praticamente permette di aggiornare tutto il software all'ultima versione disponibile.

```
$ sudo apt search package
```

Cerca il termine «package» fra i pacchetti software disponibili nel repository. Funziona anche senza sudo.

```
$ sudo apt install package
```

Scarica «package» e lo installa nel sistema, rendendolo disponibile all'utente.

```
$ sudo apt remove package
```

Rimuove «package» dal sistema.

Collegamento da remoto

Una volta che hai sistemato il raspberry e hai effettuato l'installazione del sistema operativo e dato un'occhiata al tutto, potrebbe tornarti utile **non** scollegare ogni volta mouse, tastiera, monitor e tutto il laboratorio, ma semplicemente collegare il raspberry alla rete e all'alimentazione e collegartici **da remoto!**

I metodi più utilizzati per la connessione remota al raspberry sono sostanzialmente 3: elencherò il nome comune della tecnologia, il software (server) che va installato su raspberry per renderlo disponibile e il client da utilizzare su Windows 10 per la connessione remota, visto che a scuola abbiamo PC con Windows 10. Se avete dispositivi con sistemi operativi Mac o Linux, documentatevi su Internet sui client per la corrispondente tecnologia.

Protocollo	Tipologia	Server (su RPI)	Client (su Win10)
RDP	Grafica	Xrdp	Remote Desktop
VNC	Grafica	vnc	VNC Viewer
SSH	Testuale	sshd	Putty

Avvertimento: Qualsiasi metodo sceglierai, ricordati che avrai bisogno di conoscere il **nome** e/o l'**indirizzo IP** del tuo raspberry!

Cerca di capire **prima** come sia possibile ottenere (e magari modificare) queste informazioni!

4.1 RDP

Remote Desktop Protocol è un protocollo di rete proprietario sviluppato da Microsoft, che permette la connessione remota da un computer a un altro in maniera grafica. Il protocollo di default utilizza la porta TCP e UDP 3389.

I client RDP esistono per la maggior parte delle versioni di Microsoft Windows, Linux, Unix, macOS, Android, iOS. I server RDP ufficiali esistono per i sistemi operativi Windows nonostante ne esistano anche per i sistemi Unix-Like.

Suggerimento: Su RPI

Installa il servizio xrdp:

```
$ sudo apt install xrdp
```

Fatto questo, riavvia.

Suggerimento: Su Windows

Non devi fare nulla! Ti basta cercare il software *Connessione a Desktop Remoto*

4.2 VNC

Virtual Network Computing è un protocollo per applicazioni software di controllo remoto, utilizzato per amministrare il proprio computer a distanza. Può essere utilizzato anche per controllare in remoto server che non posseggono né monitor né tastiera.

Il protocollo di comunicazione usato a livello di trasporto è il TCP sulla porta di default 5900, oppure tramite interfaccia HTTP sulla porta 5800/tcp.

Suggerimento: Su RPI

Il server VNC è disponibile di default su Raspbian, ma va abilitato tramite **raspi-config**: Interfacing Options → VNC → Enable

Fatto questo, riavvia.

Suggerimento: Su Windows

Un client VNC gratuito è il VNC Viewer di RealVNC: <https://www.realvnc.com/en/connect/download/viewer/windows/>

Scaricalo, installalo su Windows e provalo.

4.3 SSH

Secure Shell è un protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete informatica. È il protocollo che ha sostituito l'analogo, ma insicuro, Telnet, perché basato su una comunicazione **non** cifrata.

A livello server utilizza la porta 22, sia tramite TCP che UDP.

Suggerimento: Su RPI

Il server SSH è disponibile di default su Raspbian, ma va abilitato tramite **raspi-config**: Interfacing Options → SSH → Enable

Fatto questo, riavvia.

Suggerimento: Su Windows

Ti basta scaricare Putty e usarlo senza neanche installarlo!

Il sito ufficiale è: <https://www.putty.org/>

WiringPi

WiringPi (<http://wiringpi.com/>) è una libreria di interfacciamento con il GPIO del Raspberry. Viene rilasciata tramite la licenza GNU LGPLv3 e supporta un consistente numero di linguaggi di programmazione, in maniera nativa (C, C++, RTB) o tramite wrappers (Python e altri...). E' stata progettata in analogia al «Wiring System» di Arduino, da cui il nome *WiringPi* per collegare il concetto di *Wiring System* con il *Raspberry Pi*.

La libreria WiringPi include una utility a riga di comando denominata **gpio** che può essere utilizzata per visualizzare e modificare le impostazioni dei pin GPIO.

La libreria WiringPi e la sua estensione in Python saranno lo strumento principale con cui interagiranno con il GPIO, da cui la necessità di capire prima di cosa si tratta e installarla sul nostro Raspberry.

Per installarla basta eseguire il semplice comando:

```
$ sudo apt install wiringpi
```

confermare con Yes tutte le richieste di APT e al termine testare la buona riuscita dell'operazione.

Suggerimento: Prima di installare la libreria WiringPi con il comando precedente (e prima di installare qualunque pacchetto con APT), sarebbe saggio aggiornare il database dei pacchetti (update) e poi i pacchetti veri e propri del sistema (upgrade) in modo che tutto sia sincronizzato al meglio.

I comandi da eseguire sono:

```
$ sudo apt update
$ sudo apt upgrade
```

Per verificare che effettivamente sia andato tutto bene basta eseguire il comando

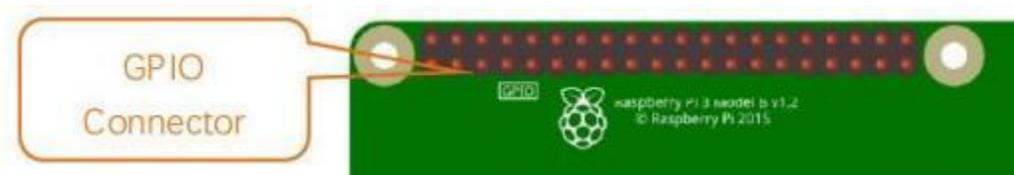
```
$ gpio -v
```

che dovrebbe fornire una serie di informazioni su di esso.

Provare per credere!!!

Qualsiasi esperienza che si vuole effettuare collegando un raspberry ad uno o più sensori non può prescindere dalla conoscenza dell'interfaccia hardware predefinita per l'input/output: il **General Purpose Input Output, GPIO**.

Tanto per essere sicuri che ci capiamo, quello fatto come in figura. . .



Prendete anche un RPI in mano e guardate bene per capire: il GPIO non è altro che la sequenza di PIN, di cui molti funzionano sia per l'input delle informazioni che per l'output delle stesse, a cui è possibile collegare praticamente ogni sorta di sensore, analogico o digitale, adatto più o meno a qualunque scopo. Da tutto questo ragionamento si spiega anche la scelta del nome: General Purpose Input Output.

Per gestire questi pin, bisogna prima riconoscerli. . . e qui nascono le prime difficoltà! Infatti, a seconda dell'interfaccia utilizzata per gestirli abbiamo 3 diversi modi per numerarli:

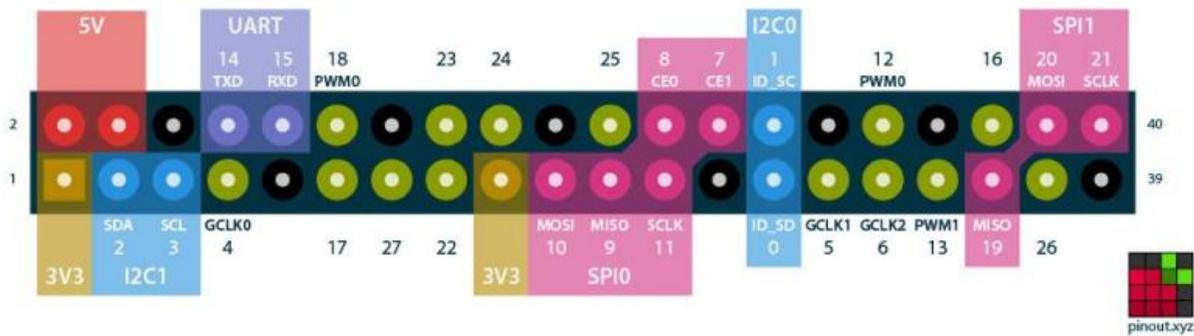
1. GPIO numbering
2. Physical numbering
3. WiringPi numbering

6.1 GPIO numbering

Questo sistema di numerazione dei pin è stato stabilito dal produttore dell'interfaccia hardware, ovvero dei pin stessi e del chipset che li controlla. Per chi è interessato alla cosa, il controller è un Broadcom BCM2835/BCM2836/BCM2837 :)

Inutile spiegare tanto a parole quando si può semplicemente guardare un'immagine e dire *vabbè*. . .

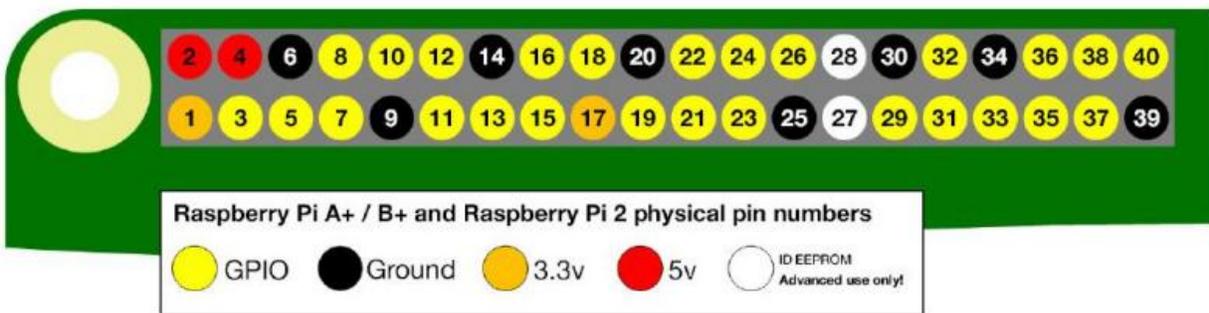
Raspberry Pi GPIO BCM numbering



6.2 Physical numbering

Un altro modo di contare i pin è quello di numerarli a due a due a partire da in alto a sinistra (il numero 1 è il pin più vicino allo slot della scheda MicroSD)

Anche qui, con un'immagine è più semplice:



6.3 WiringPi numbering

Ultimo metodo per «contare» i pin è quello basato sulla libreria *WiringPi*. L'immagine che segue è presa dal sito *WiringPi* e confronta i tre metodi di numerazione dei pin

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
—	—	3.3v	1 2	5v	—	—
8	R1:0/R2:2	SDA	3 4	5v	—	—
9	R1:1/R2:3	SCL	5 6	0v	—	—
7	4	GPIO7	7 8	TxD	14	15
—	—	0v	9 10	RxD	15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 14	0v	—	—
3	22	GPIO3	15 16	GPIO4	23	4
—	—	3.3v	17 18	GPIO5	24	5
12	10	MOSI	19 20	0v	—	—
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
—	—	0v	25 26	CE1	7	11
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

Un metodo ulteriore per controllare i numeri dei pin del raspberry è quello di interrogare direttamente il gpio tramite un piccolo applicativo contenuto nelle librerie WiringPi.

```
$ gpio readall
```

L'output di questo comando dovrebbe essere una cosa del genere:

```

pi@raspberrypi:~ $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 8 | 3.3v | ALTO | 1 | 1 | 2 | | Sv | | | |
| 3 | 9 | SDA.1 | ALTO | 1 | 3 | 4 | | 5V | | |
| 4 | 7 | SCL.1 | ALTO | 1 | 5 | 6 | | 0v | | |
| 4 | 7 | GPIO. 7 | IN | 1 | 7 | 8 | 1 | ALT5 | TxD | 15 | 14 |
| | | 0v | | | 9 | 10 | 1 | ALT5 | RxD | 16 | 15 |
| 17 | 0 | GPIO. 0 | IN | 0 | 11 | 12 | 0 | IN | GPIO. 1 | 1 | 18 |
| 27 | 2 | GPIO. 2 | IN | 0 | 13 | 14 | | | 0v | | |
| 22 | 3 | GPIO. 3 | IN | 0 | 15 | 16 | 0 | IN | GPIO. 4 | 4 | 23 |
| | | 3.3v | | | 17 | 18 | 0 | IN | GPIO. 5 | 5 | 24 |
| 10 | 12 | MOSI | ALTO | 0 | 19 | 20 | | | 0v | | |
| 9 | 13 | MISO | ALTO | 0 | 21 | 22 | 0 | IN | GPIO. 6 | 6 | 25 |
| 11 | 14 | SCLK | ALTO | 0 | 23 | 24 | 1 | OUT | CEO | 10 | 8 |
| | | 0v | | | 25 | 26 | 1 | OUT | CE1 | 11 | 7 |
| 0 | 30 | SDA.0 | IN | 1 | 27 | 28 | 1 | IN | SCL.0 | 31 | 1 |
| 5 | 21 | GPIO.21 | IN | 1 | 29 | 30 | | | 0v | | |
| 6 | 22 | GPIO.22 | IN | 1 | 31 | 32 | 0 | IN | GPIO.26 | 26 | 12 |
| 13 | 23 | GPIO.23 | IN | 0 | 33 | 34 | | | 0v | | |
| 19 | 24 | GPIO.24 | IN | 0 | 35 | 36 | 0 | IN | GPIO.27 | 27 | 16 |
| 26 | 25 | GPIO.25 | IN | 0 | 37 | 38 | 0 | IN | GPIO.28 | 28 | 20 |
| | | 0v | | | 39 | 40 | 0 | IN | GPIO.29 | 29 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Leggendo attentamente lo schema riportato possiamo desumere tutte le informazioni necessarie :)

CAPITOLO 7

La Breadboard

Blah blah