

AZ-Delivery

Benvenuto!

Grazie per aver acquistato il nostro *Modulo Joystick KY-023 AZ-Delivery*.
Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!





Indice dei Contenuti

Introduzione.....	3
Specifiche.....	4
La piedinatura.....	4
Come configurare l'Arduino IDE.....	5
Come configurare il Raspberry Pi e il Python.....	9
Collegamento del modulo con Uno.....	10
Esempio di sketch.....	11
Collegamento del modulo con Raspberry Pi.....	13
Script Python per il modulo KY-023.....	24



Introduzione

Il modulo joystick è un joystick analogico di facile utilizzo per un microcontrollore. Il joystick è dotato di due assi (asse *X* e *Y*) e di un interruttore a pulsante che si attiva quando il joystick viene premuto. Il joystick è fondamentalmente una combinazione di due potenziometri. Ciò significa che quando il joystick si muove lungo l'asse *X* la resistenza del potenziometro cambia e quando viene applicata la tensione, la variazione di resistenza si traduce in una variazione di tensione. La tensione può essere utilizzata per rilevare la posizione *X* collegando il pin *VRX* del joystick ad un ingresso analogico *A0* del pin di Uno. Lo stesso vale per l'asse *Y*. La posizione dell'asse *Y* può essere letta collegando *VRX* al pin dell'ingresso analogico *A1* di Uno. Lo stato della pressione o lo stato del pin *SW* può essere collegato al pin digitale *D2* di Uno. Poiché il modulo è costituito da due potenziometri, può essere utilizzato con qualsiasi tensione di alimentazione, ad esempio può funzionare sia a +3,3V che a +5V DC.

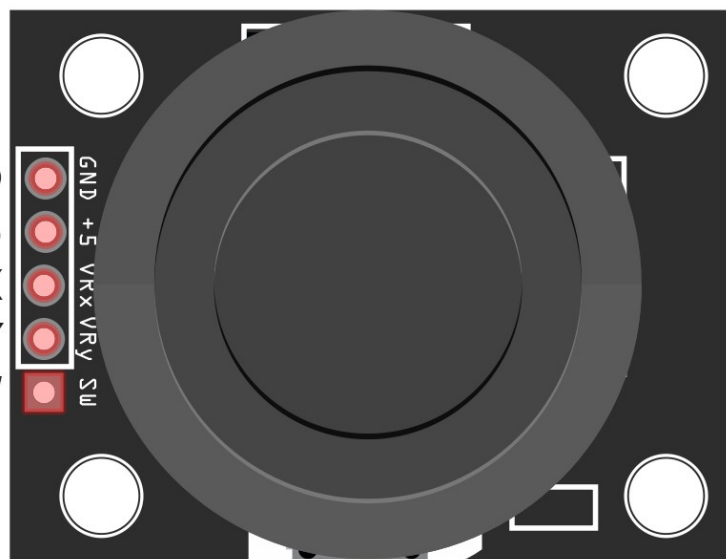
Specifiche

- » Tensione di esercizio: da 3.3V a 5V DC
- » Output: digitale, ad alta sensibilità
- » Dimensioni: 26 x 34 x 32mm [1.02 x 1.34 x 1.26in]

La piedinatura

Il modulo joystic KY-023 ha cinque pin. Lo schema di piedinatura è mostrato nella seguente immagine:

Ground - GND
Power supply - +5
X-axis - VRX
Y-axis - VRY
Switch - SW



Come configurare l'Arduino IDE

Se l'Arduino IDE non è installato, seguire il [link](#) e scaricare il file di installazione del sistema operativo scelto.

Download the Arduino IDE



ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
[Get](#) 

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Per gli utenti *Windows*, fare doppio clic sul file `.exe` scaricato e seguire le istruzioni nella finestra di installazione.

Az-Delivery

Per gli utenti *Linux*, scaricare un file con estensione *.tar.xz*, che è necessario estrarre. Quando lo si estrae, andare nella directory estratta, e aprire il terminale in quella directory. È necessario eseguire due script *.sh*, il primo chiamato *arduino-linux-setup.sh* e il secondo chiamato *install.sh*.

Per eseguire il primo script nel terminale, eseguire il seguente comando:

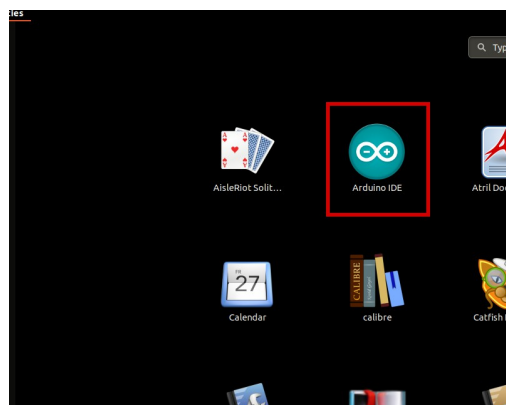
```
sh arduino-linux-setup.sh user_name
```

user_name - è il nome di un superutente nel sistema operativo Linux. Vi verrà richiesto di fornire la password per il superutente. Aspettate qualche minuto che lo script completi tutto.

Dopo l'installazione del primo script, si deve eseguire il secondo script chiamato *install.sh*. Nel terminale, eseguire il seguente comando:

```
sh install.sh
```

Dopo l'installazione di questi script, andare su *Tutte le App*, dove troverai l'Arduino IDE installato..

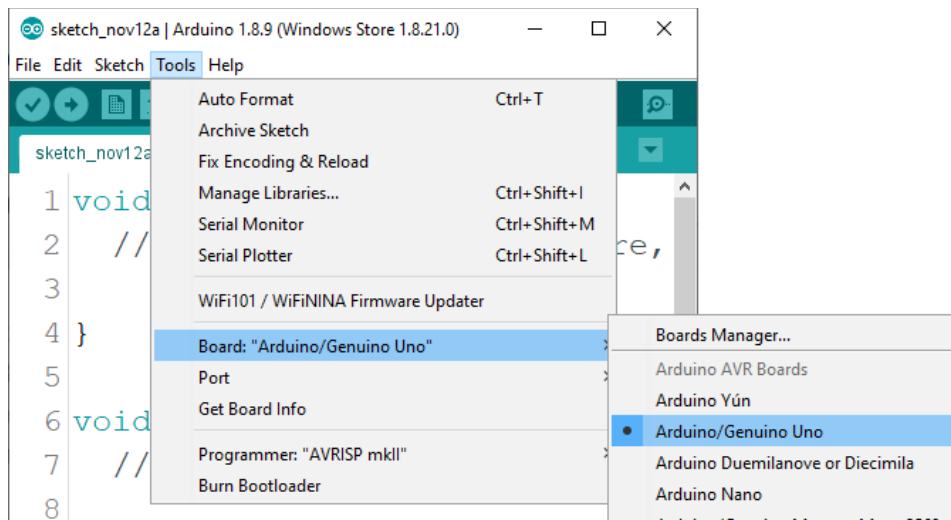


Quasi tutti i sistemi operativi sono dotati di un editor di testo preinstallato (ad esempio *Windows* viene fornito con *Notepad*, *Linux Ubuntu* viene fornito con *Gedit*, *Linux Raspbian* viene fornito con *Leafpad*, ecc..). Tutti questi editor di testo sono perfettamente adatti allo scopo dell'eBook.

La prossima cosa da fare è controllare se il PC è in grado di rilevare la scheda Arduino. Aprite l'Arduino IDE appena installato e andate su:

Strumenti > Scheda > {your board name here}

{your board name here} dovrebbe essere l'*Arduino/Genuino Uno*, come si può vedere nella seguente immagine:



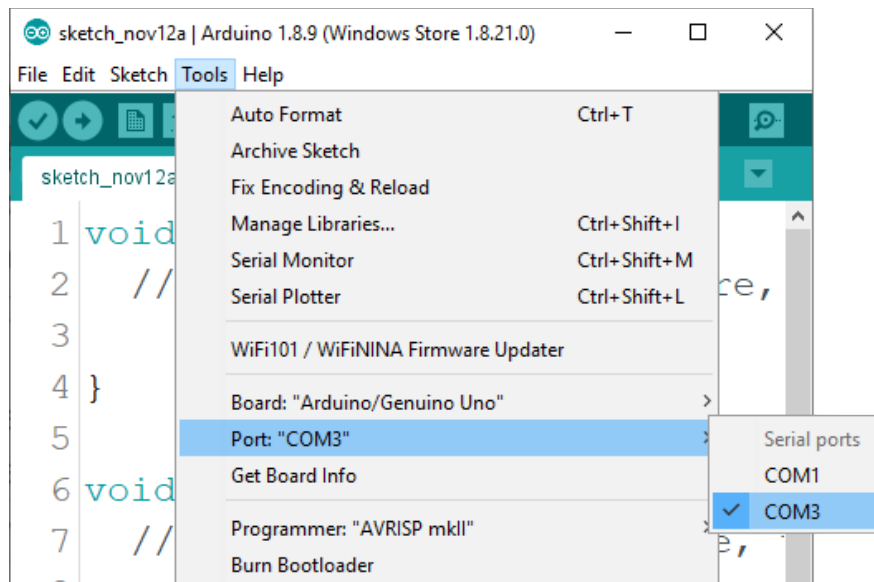
È necessario selezionare la porta alla quale è collegata la scheda Arduino.

Vai su: *Strumenti > Porta > {port name goes here}*

e se avete collegato la scheda Arduino sulla porta usb dovreste vedere il nome della porta.

Az-Delivery

Se si utilizza l'Arduino IDE su Windows, i nomi delle porte sono i seguenti:



Per gli utenti *Linux*, il nome della porta è `/dev/ttyUSBx` per esempio, dove `x` rappresenta un numero intero compreso tra `0` e `9`, per esempio.



Come configurare il Raspberry Pi e il Python

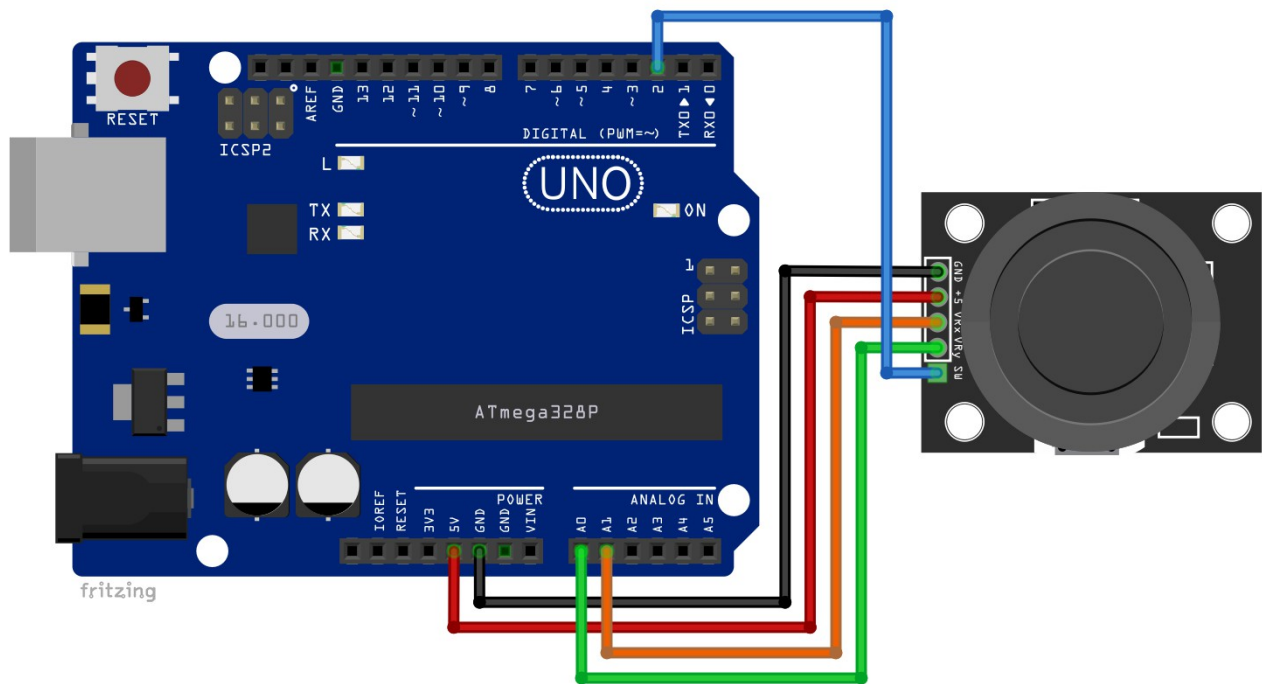
Per il Raspberry Pi, dovrete prima installare il sistema operativo su di esso, poi impostare tutto, in modo da poterlo utilizzare in modalità *Headless*. La modalità *Headless* consente di collegarsi a distanza al Raspberry Pi, senza bisogno del monitor del PC, mouse o tastiera. Le uniche cose di cui avete bisogno per questa modalità sono il Raspberry Pi, l'alimentazione e la connessione internet. Tutto questo è spiegato in dettaglio nell'eBook gratuito *Raspberry Pi Quick Startup Guide*, che si trova sul nostro sito:

<https://www.az-delivery.de/products/raspberry-pi-kostenfreies-e-book?ls=en>

Il sistema operativo *Raspbian* viene fornito con il *Python* preinstallato..

Collegamento del modulo con Uno

Collegare il modulo KY-023 con Uno come indicato nel seguente schema di collegamento:



Pin KY-023	>	Pin Uno
GND	>	GND
+5V	>	5V
VRX	>	A1
VRY	>	A0
SW	>	D2

Filo nero

Filo rosso

Filo arancio

Filo verde

Filo blu

Esempio di sketch

```
#define VX_PIN 0
#define VY_PIN 1
#define BUTTON_PIN 2
uint8_t value = 0;

void setup() {
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    value = analogRead(VX_PIN);
    Serial.print("X:");
    Serial.print(value, DEC);

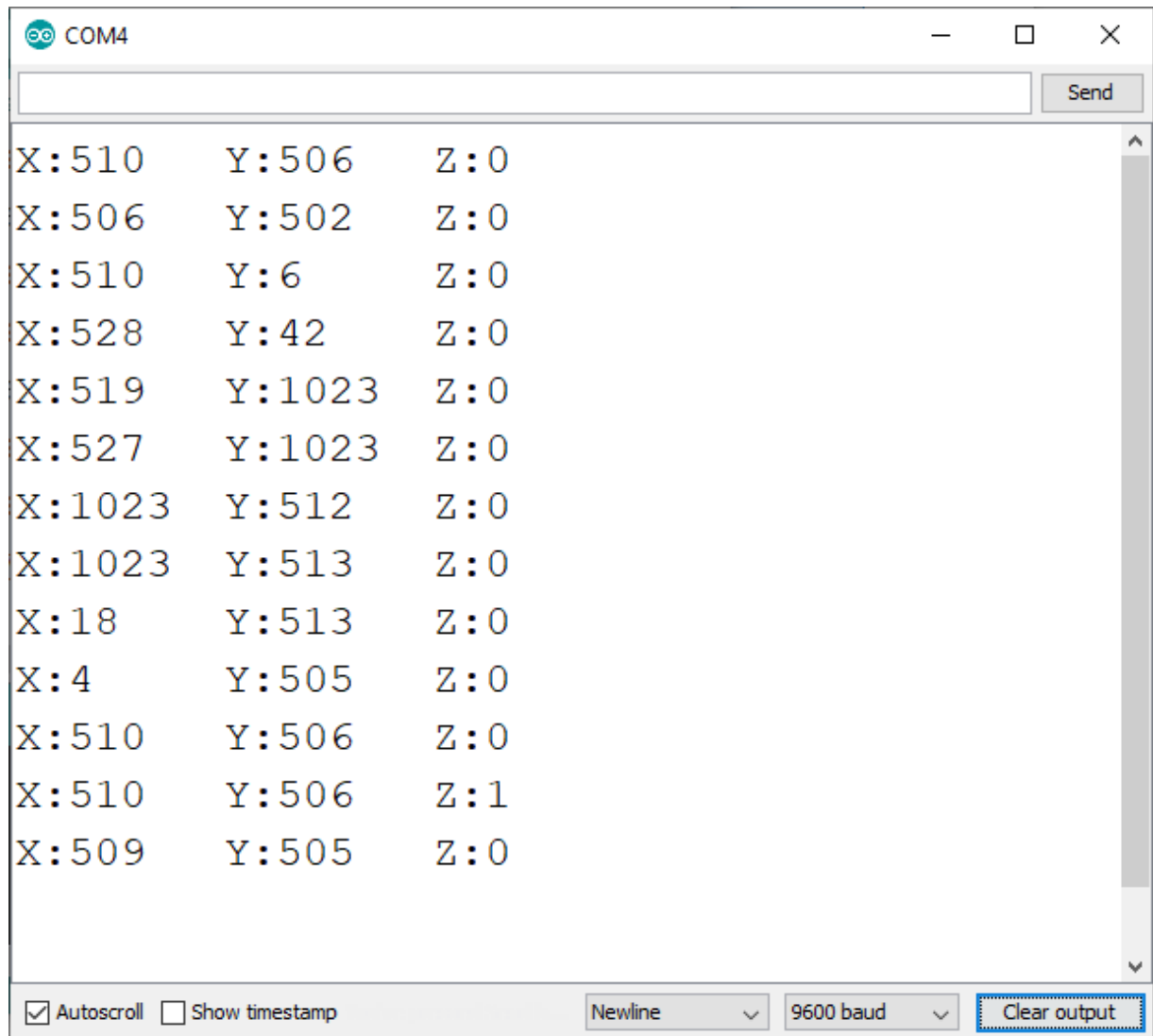
    value = analogRead(VY_PIN);
    Serial.print("\tY:");
    Serial.print(value, DEC);

    value = !digitalRead(BUTTON_PIN);
    Serial.print("\tZ:");
    Serial.println(value, DEC);

    delay(500);
}
```

Az-Delivery

Caricare lo sketch su Uno e aprire il Monitor Seriale (*Strumenti > Monitor Seriale*). Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:



Per ottenere l'output dell'immagine spostare o premere il Joystick.



Collegamento del modulo con Raspberry Pi

Dato che il Raspberry Pi non ha un convertitore analogico-digitale (ADC), per poter utilizzare il modulo KY-023 con il Raspberry Pi, il Raspberry Pi deve essere in grado di leggere tensioni analogiche. A tal fine è possibile utilizzare l'Uno. Per fare ciò, l'Uno viene utilizzato sul sistema operativo *Linux Raspbian*. Uno può leggere tensioni analogiche e può utilizzare l'*interfaccia seriale via porta USB* per inviare dati al Pi Raspberry.

Per prima cosa, bisogna scaricare e installare l'Arduino IDE sul Raspbian. In secondo luogo, si deve scaricare e caricare il firmware per Uno e, infine, si deve scaricare e installare la libreria per Python.

Per farlo, accendete il Raspberry Pi e collegatelo a internet. Avviate l'app *RealVNC* sul computer remoto e collegate l'app al Raspberry Pi (come spiegato nell'eBook per il Raspberry Pi).

La prima cosa da fare quando il Raspberry Pi viene avviato è aggiornare il Raspbian; aprire il terminale ed eseguire il seguente comando:

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

E aspettate che il comando finisca il suo lavoro.

Ora, il sistema operativo Raspbian è aggiornato.

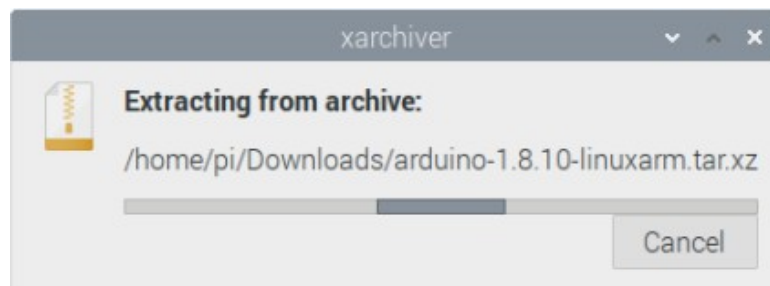
Az-Delivery

Per scaricare e installare l'Arduino IDE, andate sul [sito](#) di Arduino e scaricate il file `tar.xz` di Arduino IDE per **Linux ARM 32 bit** come mostrato nella seguente immagine:

Download the Arduino IDE



Poi, deve essere estratto il file `tar.xz`. Aprire il File explorer nella directory dove viene scaricato il file `tar.xz`, fare clic con il tasto destro del mouse su di esso, ed eseguite l'opzione *Estrarre qui*. Attendete qualche minuto che il processo di estrazione si completi da solo.

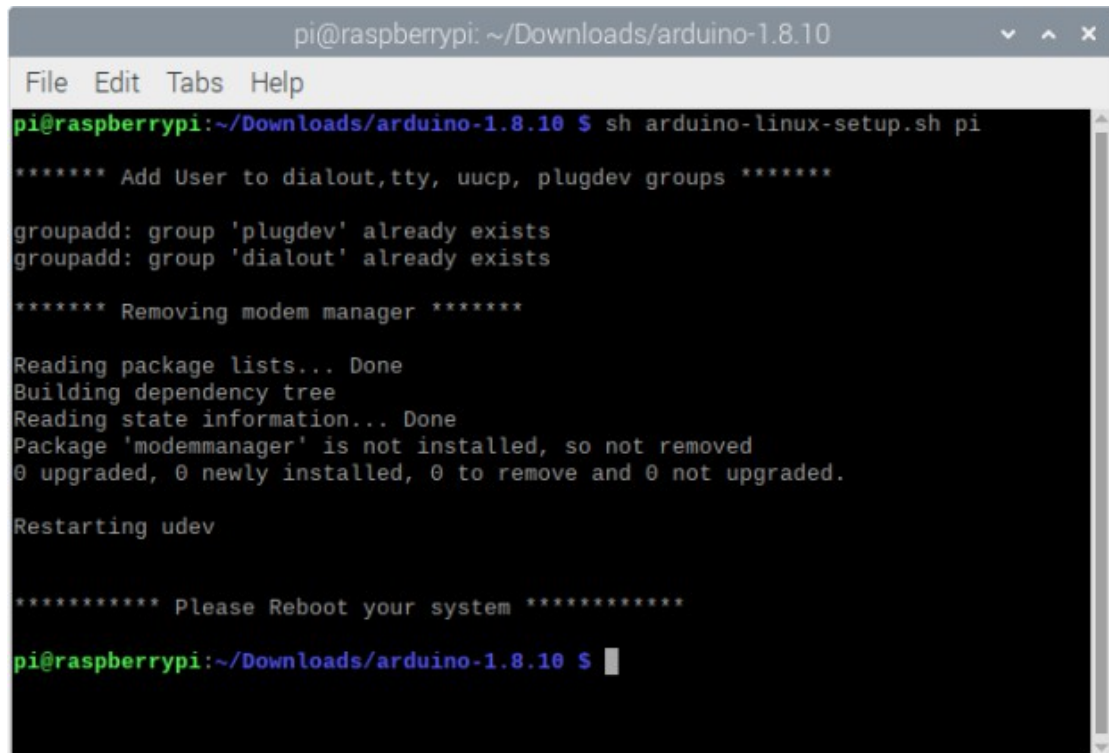


Az-Delivery

Aprirete il terminale nella cartella estratta ed eseguite il seguente comando:

sh arduino-linux-setup.sh pi

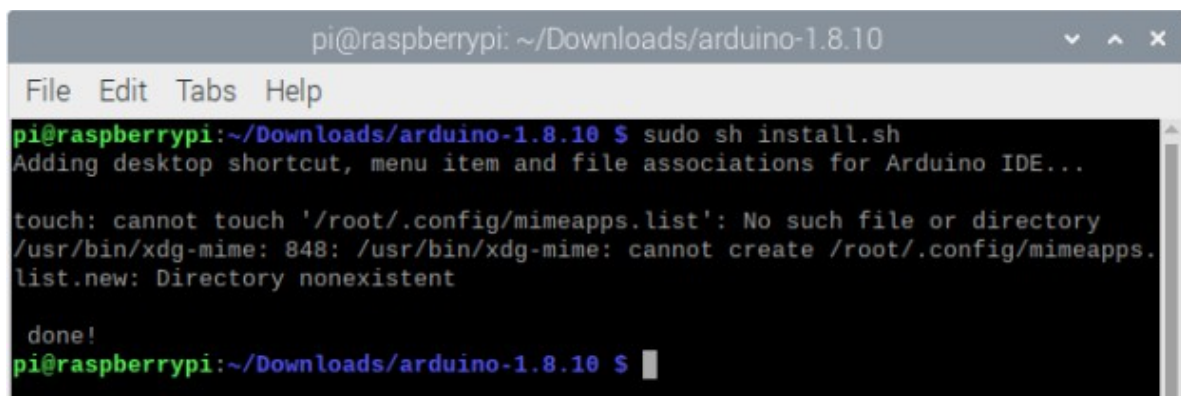
dove pi è il nome del superutente in Raspbian.



```
pi@raspberrypi: ~/Downloads/arduino-1.8.10
File Edit Tabs Help
pi@raspberrypi:~/Downloads/arduino-1.8.10 $ sh arduino-linux-setup.sh pi
***** Add User to dialout, tty, uucp, plugdev groups *****
groupadd: group 'plugdev' already exists
groupadd: group 'dialout' already exists
***** Removing modem manager *****
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'modemmanager' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Restarting udev
***** Please Reboot your system *****
pi@raspberrypi:~/Downloads/arduino-1.8.10 $
```

Dopo di che, per installare l'IDE Arduino, eseguite il seguente comando:

sudo sh install.sh



```
pi@raspberrypi: ~/Downloads/arduino-1.8.10
File Edit Tabs Help
pi@raspberrypi:~/Downloads/arduino-1.8.10 $ sudo sh install.sh
Adding desktop shortcut, menu item and file associations for Arduino IDE...
touch: cannot touch '/root/.config/mimeapps.list': No such file or directory
/usr/bin/xdg-mime: 848: /usr/bin/xdg-mime: cannot create /root/.config/mimeapps.
list.new: Directory nonexistent
done!
pi@raspberrypi:~/Downloads/arduino-1.8.10 $
```

Az-Delivery

L'Arduino IDE è ora installato. Per eseguire Arduino IDE, aprire l'app:

Menu Applicazioni > Programmazione > Arduino IDE



Prima dei passi successivi, è necessario installare prima le applicazioni *pip3* e *git*; eseguite il seguente comando:

```
sudo apt install python3-pip git -y
```

La libreria per Python si chiama *nanpy*. Per installarla, aprite il terminale ed eseguite il seguente comando: **pip3 install nanpy**

```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ pip3 install nanpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting nanpy
  Downloading https://www.piwheels.org/simple/nanpy/nanpy-0.9.6-py3-none-any.whl
    (47kB)
    100% |#####| 51kB 209kB/s
Requirement already satisfied: pyserial in /usr/lib/python3/dist-packages (from
nanpy) (3.4)
Installing collected packages: nanpy
Successfully installed nanpy-0.9.6
pi@raspberrypi:~/Scripts $
```


Az-Delivery

Dopo aver installato la libreria *nanpy*, scaricate un firmware Arduino eseguendo il seguente comando:

```
git clone https://github.com/nanpy/nanpy-firmware.git
```

Cambiate la directory in *nanpy-firmware* eseguendo il seguente comando:

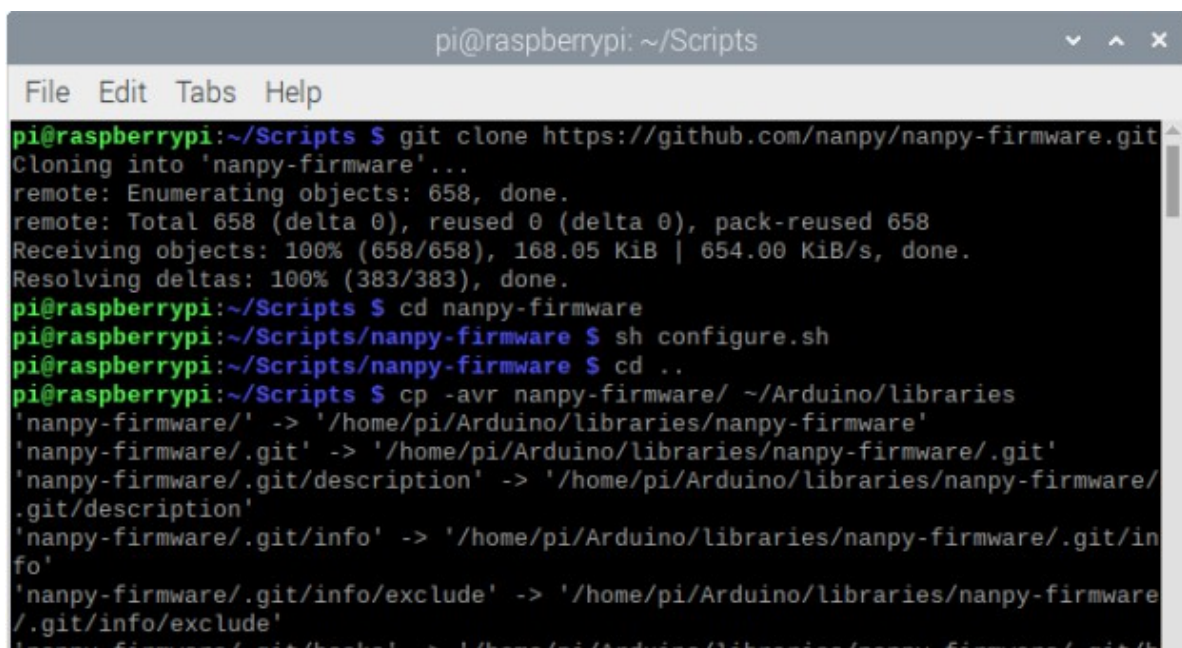
```
cd nanpy-firmware
```

Ed eseguite il seguente comando:

```
sh configure.sh
```

Successivamente, copiate la directory *nanpy-firmware* nella directory *Arduino/librerie*. Per farlo, eseguite il seguente comando:

```
cp -avr nanpy-firmware/ ~/Arduino/libraries
```



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ git clone https://github.com/nanpy/nanpy-firmware.git
Cloning into 'nanpy-firmware'...
remote: Enumerating objects: 658, done.
remote: Total 658 (delta 0), reused 0 (delta 0), pack-reused 658
Receiving objects: 100% (658/658), 168.05 KiB | 654.00 KiB/s, done.
Resolving deltas: 100% (383/383), done.
pi@raspberrypi:~/Scripts $ cd nanpy-firmware
pi@raspberrypi:~/Scripts/nanpy-firmware $ sh configure.sh
pi@raspberrypi:~/Scripts/nanpy-firmware $ cd ..
pi@raspberrypi:~/Scripts $ cp -avr nanpy-firmware/ ~/Arduino/libraries
'nanpy-firmware/' -> '/home/pi/Arduino/libraries/nanpy-firmware'
'nanpy-firmware/.git' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git'
'nanpy-firmware/.git/description' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/description'
'nanpy-firmware/.git/info' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/info'
'nanpy-firmware/.git/info/exclude' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/info/exclude'
'nanpy-firmware/.git/hooks' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/hooks'
```

Il firmware nanpy è ora installato e pronto per essere utilizzato.

Az-Delivery

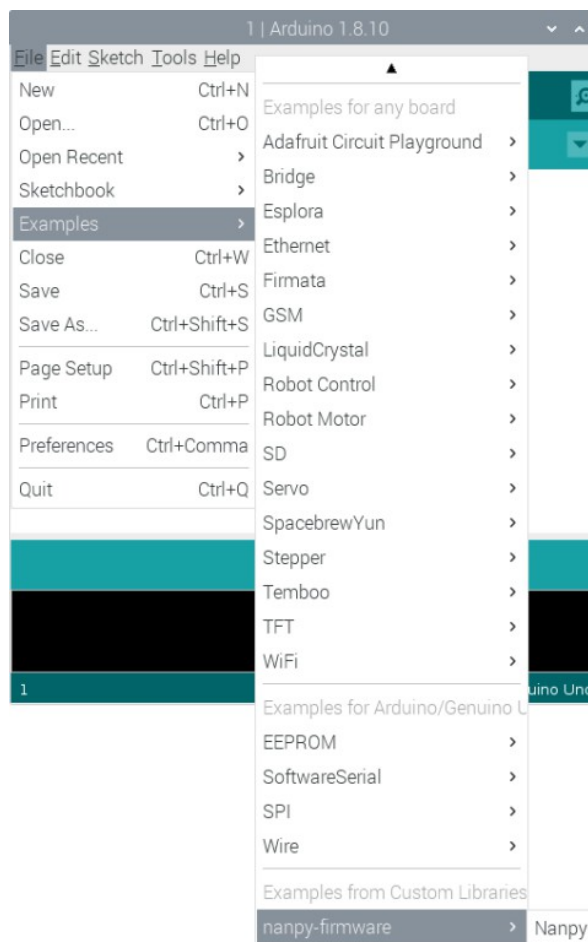
Collegate Uno tramite il cavo USB al Raspberry Pi e poi aprite l'Arduino IDE nel sistema operativo Raspbian. Controllate se l'Arduino IDE è in grado di rilevare la porta USB a cui è collegato Uno:

Strumenti > Porta > dev/ttyUSB0

Poi andate su: Strumenti > Scheda > {board name}
e selezionate la scheda Arduino Uno.

Poi, per aprire lo sketch per il firmware nanpy, andate su:

File > Esempi > nanpy-firmware > Nanpy

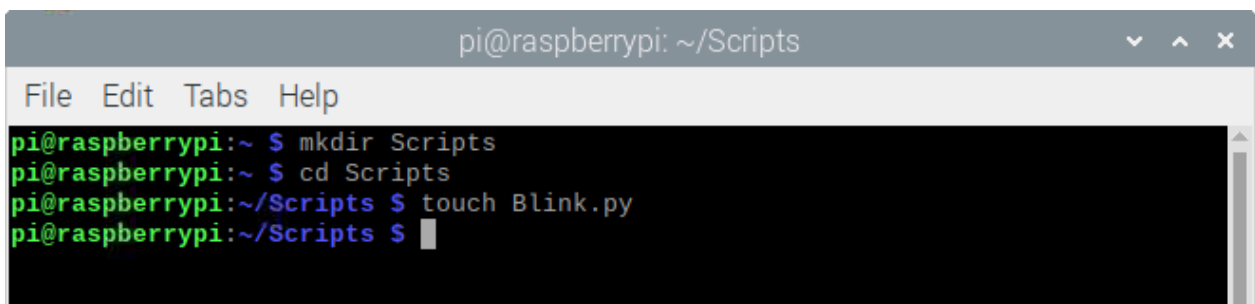


Az-Delivery

Caricare lo sketch su Uno. Per verificare se tutto funziona correttamente, è necessario creare il semplice script *Blink*, dove il LED a bordo di Uno lampeggia.

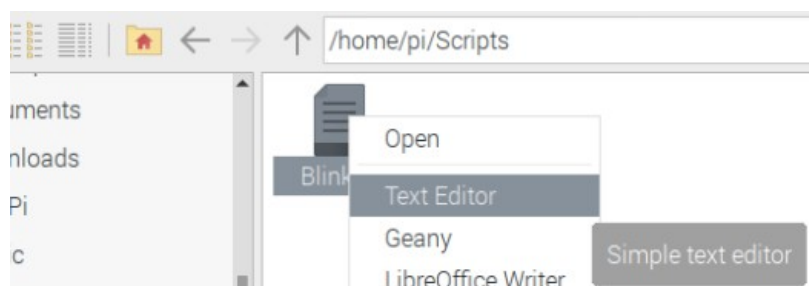
Aprire il terminale, creare la directory *Scripts* e lo script *Blink.py*. Per farlo, eseguite i seguenti comandi, uno per uno:

- | | |
|-----------------------|-----------------------------------------------|
| mkdir Scripts | - crea la directory degli Script |
| cd /Scripts | - modifica la directory degli Script |
| touch Blink.py | - crea un nuovo file chiamato <i>Blink.py</i> |



```
pi@raspberrypi:~/Scripts
File Edit Tabs Help
pi@raspberrypi:~ $ mkdir Scripts
pi@raspberrypi:~ $ cd Scripts
pi@raspberrypi:~/Scripts $ touch Blink.py
pi@raspberrypi:~/Scripts $
```

Aprire File Explorer, navigate nella directory *Scripts* e aprite lo script *Blink.py* nell'editor di testo predefinito:



Az-Delivery

Nello script `Blink.py` scrivete le seguenti righe di codice:

```
from nanpy import (ArduinoApi, SerialManager)
from time import sleep
```

```
ledPin = 13
```

```
try:
```

```
    connection1 = SerialManager()
```

```
    a = ArduinoApi(connection=connection1)
```

```
except:
```

```
    print('Failed to connect to the Arduino')
```

```
print('[Press CTRL + C to end the script!']')
```

```
a.pinMode(ledPin, a.OUTPUT) # Setup Arduino
```

```
try:
```

```
    while True:
```

```
        a.digitalWrite(ledPin, a.HIGH)
```

```
        print('Bulit in led HIGH')
```

```
        sleep(1)
```

```
        a.digitalWrite(ledPin, a.LOW)
```

```
        print('Bulit in led LOW')
```

```
        sleep(1)
```

```
except KeyboardInterrupt:
```

```
    print('\nScript end!')
```

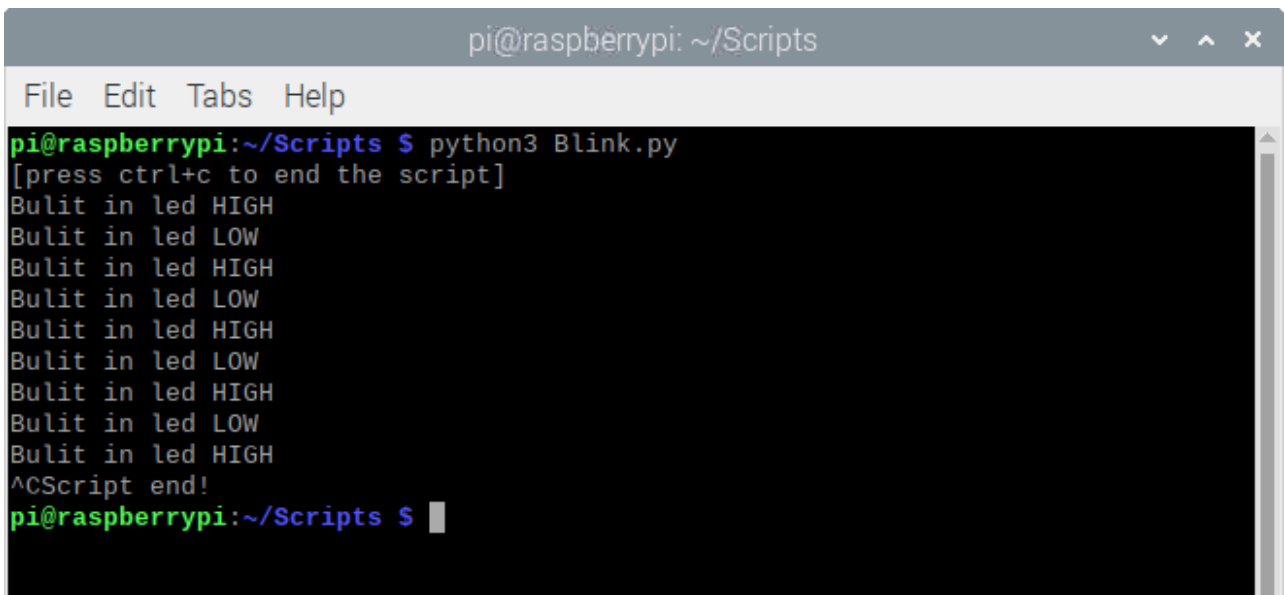
```
    a.digitalWrite(ledPin, a.LOW)
```

Az-Delivery

Salvate lo script. Per eseguire lo script, aprire il terminale nella directory in cui è stato salvato lo script ed eseguire il seguente comando:

python3 Blink.py

Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 Blink.py
[press ctrl+c to end the script]
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
^CScript end!
pi@raspberrypi:~/Scripts $
```

Per fermare lo script premere CTRL + C sulla tastiera.

Il LED collegato al pin digitale 13 di Uno dovrebbe iniziare a lampeggiare ogni secondo.

Az-Delivery

Lo script inizia con l'importazione di due librerie, le funzioni della libreria *nanpy* e *time*.

Quindi, viene creata la variabile chiamata *ledPin* e inizializzata con il numero 13. Il numero 13 rappresenta il numero del pin digitale su cui è collegato il LED (LED a bordo di UNO).

Dopo di che, il blocco di codice *try-except* viene utilizzato per provare a connettersi a Uno. Se la connessione non va a buon fine, il messaggio: *Failed to connect to the Arduino* viene visualizzato nel terminale.

Se la connessione ha successo, l'oggetto di comunicazione chiamato "a" viene creato e inizializzato. L'oggetto "a" rappresenta la scheda Uno. Qualsiasi funzione utilizzata nell'Arduino IDE può essere utilizzata con l'oggetto "a", come si può vedere nel codice.

Con la seguente riga di codice, si configura la modalità pin del pin digitale 13: *a.pinMode(ledPin, a.OUTPUT)*

Poi, nel blocco del loop indefinito (*while True:*) viene usata la funzione *digitalWrite()* per configurare lo stato del pin digitale 13 (stato *HIGH* o *LOW*). Con la funzione *digitalWrite()* il LED collegato al pin 13 può essere *ACCESO* o *SPENTO*.

Az-Delivery

Nel blocco `loop` indefinito, il LED viene prima *ACCESO* per un secondo, e poi *SPENTO* per un secondo. Questo è un LED lampeggiante.

L'intervallo di tempo di un singolo lampeggio può essere modificato nella seguente riga di codice: `sleep(1)`

Dove il numero 1 rappresenta il numero di secondi per la durata dell'intervallo di tempo.

Per fermare il loop infinito premere CTRL + C sulla tastiera.. Questo è chiamato interrupt da tastiera, che è impostato nel blocco `except` (*except KeyboardInterrupt*). Nel blocco `except` il LED a bordo viene *SPENTO*.



Script Python per il modulo KY-023

Collegate il modulo KY-023 con Uno come indicato nello schema di collegamento nel capitolo Collegamento del modulo con Uno, e poi collegate Uno con il Raspberry Pi tramite cavo USB. Successivamente, caricate il firmware nanpy su Uno e utilizzate il seguente codice per controllare il modulo KY-023:

```
from nanpy import (ArduinoApi, SerialManager)
from time import sleep
try:
    connection1 = SerialManager()
    a = ArduinoApi(connection=connection1)
except:
    print('Failed to connect to the Arduino')

# Pin setup for the Arduino
a.pinMode(2, a.INPUT_PULLUP)
valueX, valueY, valueZ = 0, 0, 0
print('[Press CTRL + C to end the script!]\n')
try: # Main program loop
    while True:
        valueX = a.analogRead(0)
        valueY = a.analogRead(1)
        valueZ = not a.digitalRead(2)
        print('X:{}\tY:{}\tZ:{}'.format(valueX, valueY, valueZ))
        sleep(0.5)

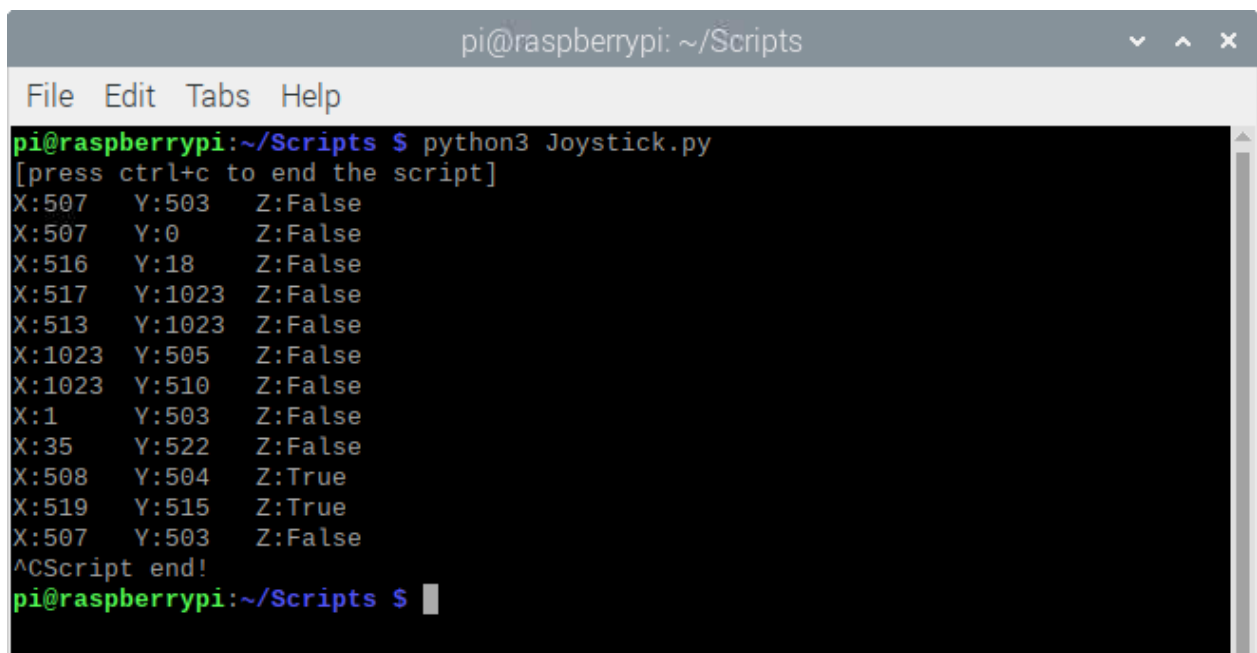
except KeyboardInterrupt:
    print('\nScript end!')
```


Az-Delivery

Salvare lo script con il nome *Joystick.py*. Per eseguire lo script, aprire il terminale nella directory in cui è stato salvato lo script ed eseguire il seguente comando:

python3 Joystick.py

Il risultato dovrebbe assomigliare all'uscita mostrata nell'immagine seguente:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 Joystick.py
[press ctrl+c to end the script]
X:507 Y:503 Z:False
X:507 Y:0 Z:False
X:516 Y:18 Z:False
X:517 Y:1023 Z:False
X:513 Y:1023 Z:False
X:1023 Y:505 Z:False
X:1023 Y:510 Z:False
X:1 Y:503 Z:False
X:35 Y:522 Z:False
X:508 Y:504 Z:True
X:519 Y:515 Z:True
X:507 Y:503 Z:False
^CScript end!
pi@raspberrypi:~/Scripts $
```

Per fermare lo script premere CTRL + C sulla tastiera.



E ora è tempo di imparare e di creare dei Progetti da solo. Lo puoi fare con l'aiuto di molti script di esempio e altri tutorial, che puoi trovare in internet..

Se stai cercando dei prodotti di alta qualità per il tuo Arduino e Raspberry Pi, AZ-Delivery Vertriebs GmbH è l'azienda giusta dove potrai trovarli. Ti forniremo numerosi esempi di applicazioni, guide di installazione complete, e-book, librerie e l'assistenza dei nostri esperti tecnici.

<https://az-delivery.de>

Buon divertimento!

Impressum

<https://az-delivery.de/pages/about-us>