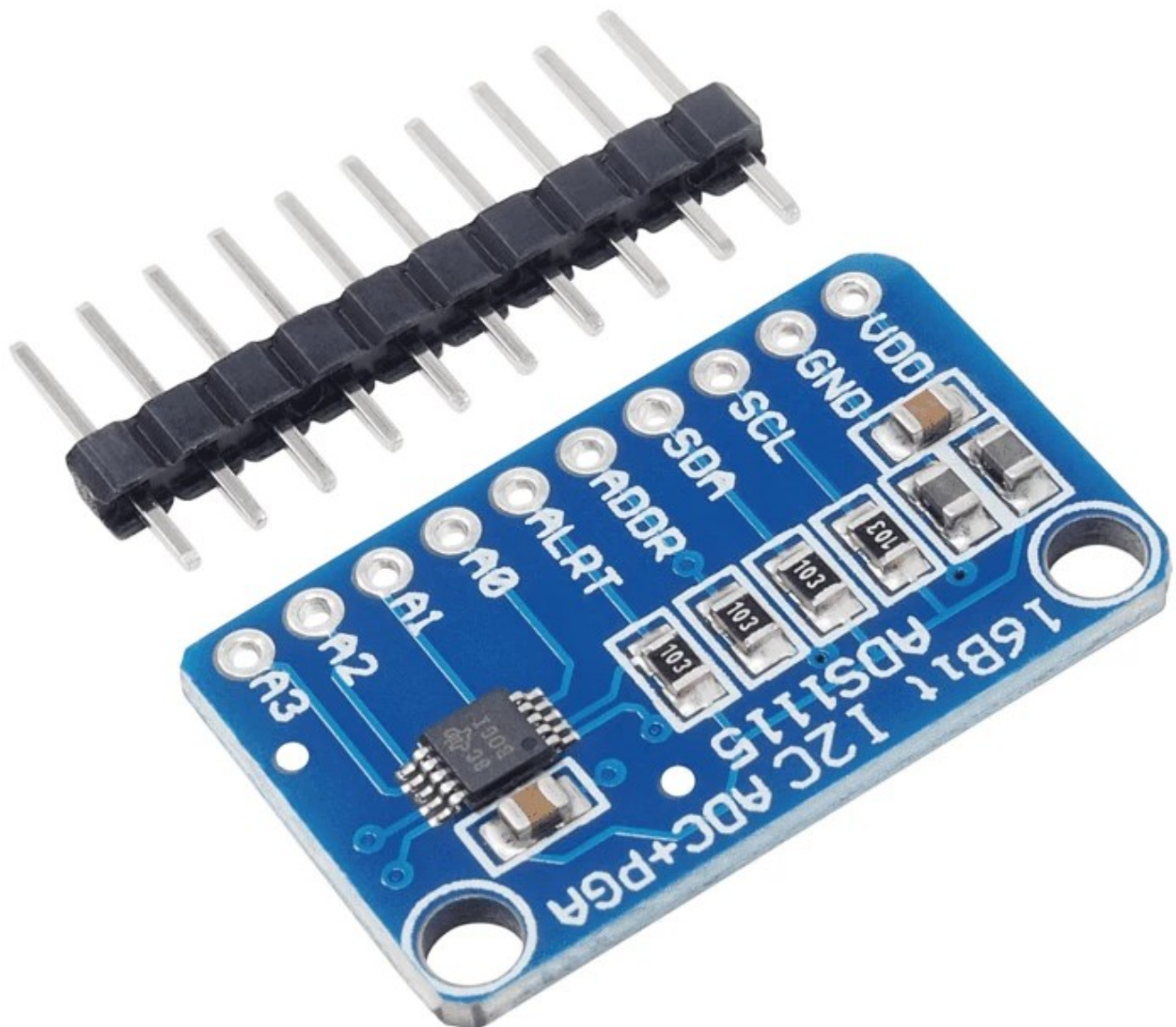


Benvenuto!

Grazie per aver acquistato il nostro Convertitore da Analogico a Digitale ADS1115 Az Delivery. Nelle pagine seguenti, ti illustreremo come utilizzare e configurare questo pratico dispositivo.

Buon divertimento!



Az-Delivery

Il modulo ADS1115 è una scheda breakout a quattro canali. Questi moduli sono perfetti per aggiungere la conversione analogico-digitale ad alta risoluzione a qualsiasi progetto basato su microprocessore (come Raspberry Pi) o se si desidera aggiungere un convertitore analogico-digitale con maggiore precisione a un progetto basato su microcontrollore (come Arduino).

Questo modulo può funzionare con segnali logici e di alimentazione tra 2V e 5V, quindi sono compatibili con tutti i comuni processori da 3,3V e 5V. Poiché molte di queste 4 schede possono essere controllate dallo stesso bus I2C. Questo ti dà fino a 16 canali single-ended o 8 differenziali.

Un gain amplifier programmabile fornisce un guadagno fino a x16 per piccoli segnali.

Specifiche

- » Risoluzione: 16 Bits
- » Sample Rate Programmabile: da 8 a 860 campionamenti al secondo
- » Alimentazione e Livelli Logici: 2.0V a 5.5V
- » Basso Consumo di Corrente: Modalità Continua: 150µA
- » Modalità Single-Shot: Autospegnimento
- » Guadagno Interno: fino a x16
- » Interfaccia I2C: Indirizzi Selezionabili 4 pin
- » Input: 4 Single-Ended o 2 Input Differenziali
- » Comparatore Programmabile
- » Riferimento Voltaggio Low-Drift Interno
- » Oscillatore Interno



Indirizzamento I2C

Il chip ADS1115 ha un indirizzo I2C a 7 bit, un valore predefinito di *0x48* e uno schema di indirizzamento che consente quattro indirizzi diversi utilizzando solo un pin di indirizzo denominato *ADDR*.

Per impostare l'indirizzo, collegare il pin dell'indirizzo come segue:

ADR -> GND	<i>0x48</i>
ADR -> VDD	<i>0x49</i>
ADR -> SDA	<i>0x4A</i>
ADR -> SCL	<i>0x4B</i>

Questo schema di indirizzamento ci consente di collegare quattro diversi moduli sullo stesso microprocessore.



Input Single Ended vs. Differenziali

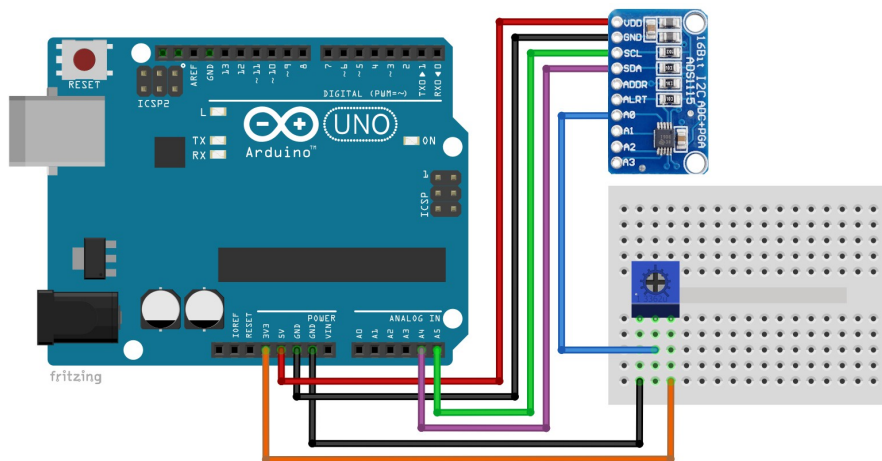
Il modulo supporta quattro ingressi single ended o due differenziali. Gli ingressi single ended misurano la tensione tra il canale di ingresso analogico (uno qualsiasi dei pin A0, A1, A2 e A3) e la terra analogica (pin GND). Gli ingressi differenziali misurano la tensione tra due canali di ingresso analogici, pin A0 e A1 o pin A2 e A3.

Gli ingressi single ended offrono quattro ingressi analogici. Per definizione, gli ingressi single ended misurano solo tensioni positive. È possibile ottenere solo una risoluzione effettiva di 15 bit.

Le misurazioni differenziali offrono una maggiore immunità dal rumore elettromagnetico (quando si usano cavi di segnale lunghi o si opera in un ambiente elettricamente rumoroso). Ciò è desiderabile anche quando si ha a che fare con piccoli segnali che richiedono un alto guadagno, poiché il guadagno amplifica il rumore e il segnale. Gli ingressi differenziali forniscono la risoluzione a 16 bit e la capacità di misurare tensioni negative.

Collegamento del modulo con Arduino Uno

Arduino Uno ha già i pin di ingresso analogico, ma perché dovremmo collegare un altro convertitore da analogico a digitale ad Arduino? Il modulo è più preciso del convertitore da analogico a digitale integrato all'interno del microcontrollore a bordo di Arduino. Quindi, se hai bisogno di un convertitore analogico-digitale più preciso, puoi usare questo modulo con Arduino. Collega tutto come nello schema qui sotto, usiamo il potenziometro solo per esempio, per leggere alcune variazioni della tensione analogica:



Pin Modulo > Pin Arduino

VDD > 5V

Filo rosso

GND > GND

Filo nero

SCL > A5

Filo verde

SDA > A4

Filo viola

Pin Modulo > Pin Potenziometro

A0 > Pin centrale

Filo blu

Pin Arduino > Pin Potenziometro

3.3V > Pin destro

Filo arancio

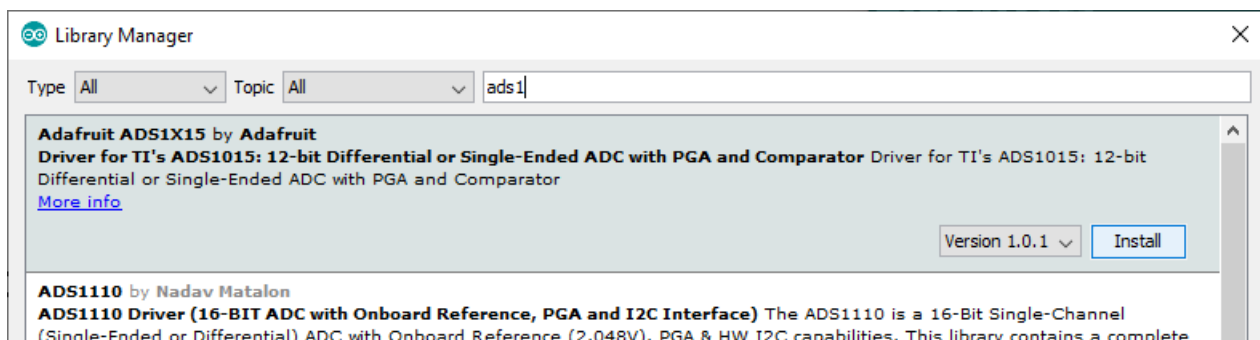
GND > Pin sinistro

Filo nero

Az-Delivery

Libreria Arduino IDE

Dobbiamo prima scaricare una libreria per il nostro modulo. Apri l'Arduino IDE e vai su *Strumenti> Gestisci librerie* e nella casella di ricerca digita "ADS1115" e scarica la libreria da Adafruit, come nell'immagine qui sotto:

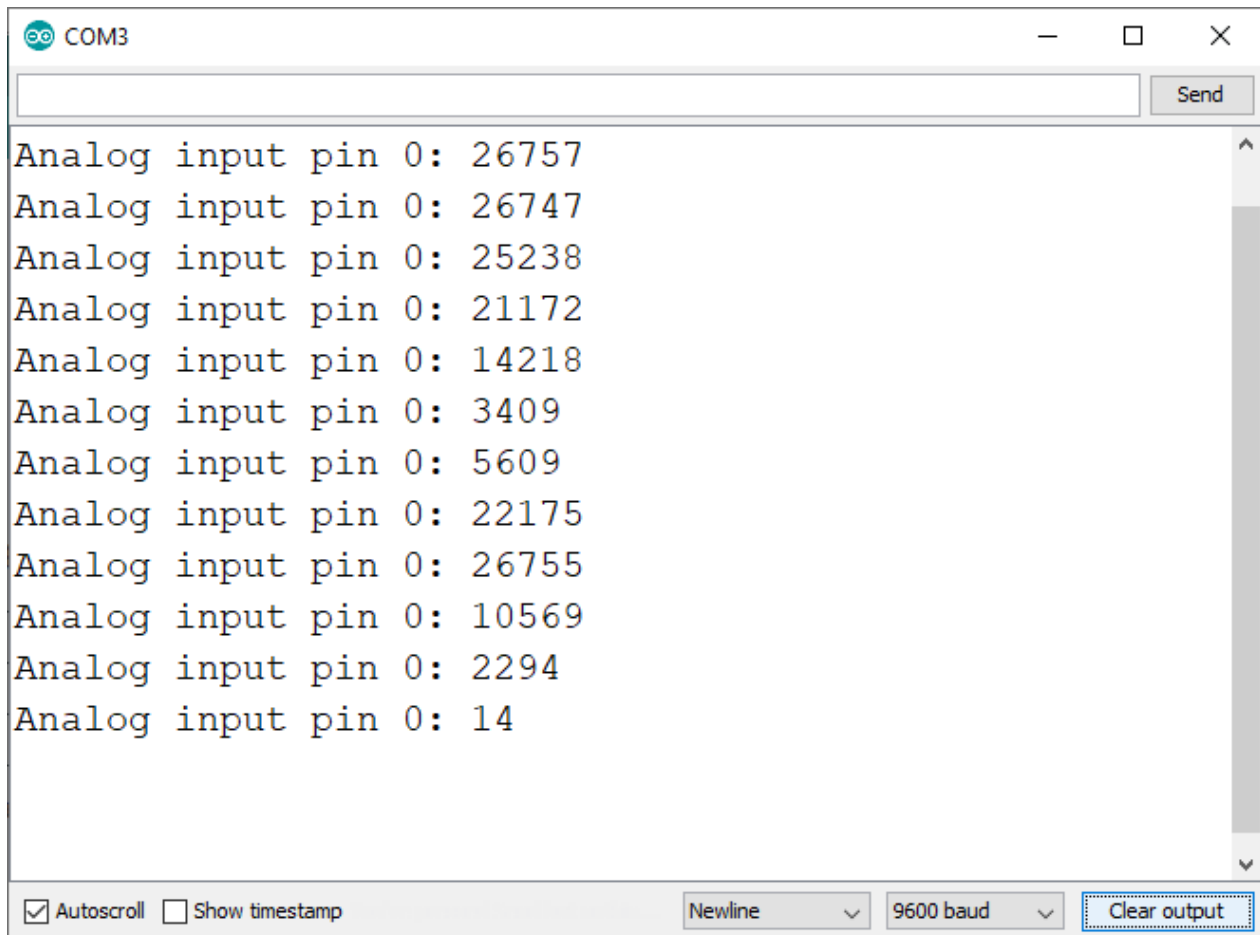


Codice Arduino:

```
#include <Wire.h>
#include <Adafruit_ADS1X15.h>
Adafruit_ADS1115 ads;
int16_t adc0;
void setup() {
    Serial.begin(9600);          //gain
    // ads.setGain(GAIN_TWOTHIRDS); // 2/3x +/- 6.144V 1bit = 0.1875mV default
    // ads.setGain(GAIN_ONE);        // 1x   +/- 4.096V 1bit = 0.125mV
    // ads.setGain(GAIN_TWO);        // 2x   +/- 2.048V 1bit = 0.0625mV
    // ads.setGain(GAIN_FOUR);       // 4x   +/- 1.024V 1bit = 0.03125mV
    // ads.setGain(GAIN_EIGHT);      // 8x   +/- 0.512V 1bit = 0.015625mV
    // ads.setGain(GAIN_SIXTEEN);    // 16x  +/- 0.256V 1bit = 0.0078125mV
    ads.begin();
}
void loop() {
    adc0 = ads.readADC_SingleEnded(0);
    Serial.print("Analog input pin 0: ");
    Serial.println(adc0);
    delay(1000);
}
```

Az-Delivery

E quando si avvia Monitor seriale (*Strumenti* > *Monitor seriale*) l'output quando si sposta l'albero del potenziometro dovrebbe apparire così:



The screenshot shows a serial monitor window titled "COM3". It displays a list of 12 analog input values for pin 0, ranging from 26757 down to 14. The window includes a "Send" button at the top right and a status bar at the bottom with options for "Autoscroll" (checked), "Show timestamp" (unchecked), "Newline" (dropdown), "9600 baud" (dropdown), and a "Clear output" button.

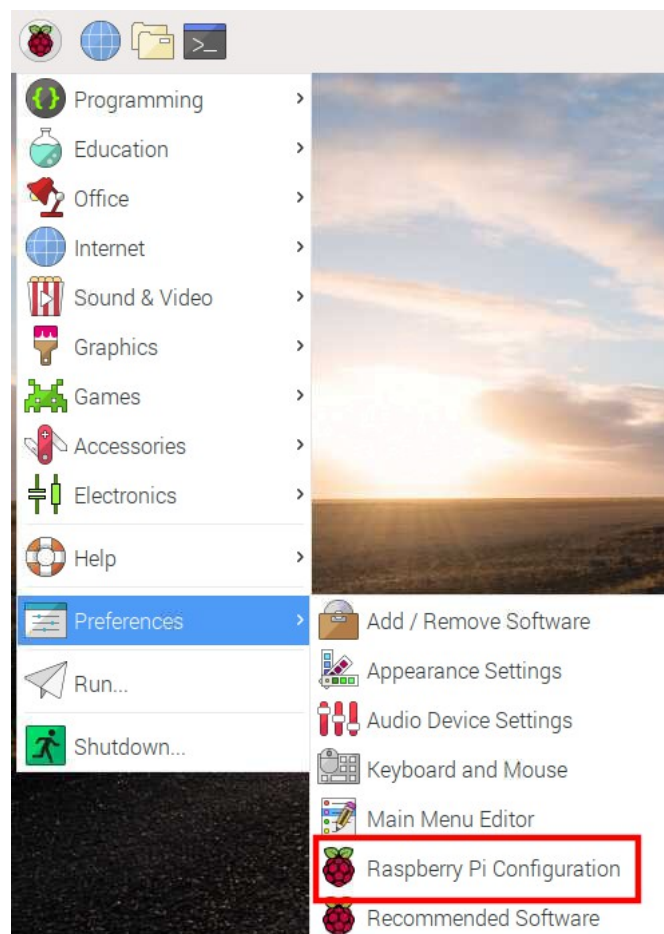
```
COM3  
Analog input pin 0: 26757  
Analog input pin 0: 26747  
Analog input pin 0: 25238  
Analog input pin 0: 21172  
Analog input pin 0: 14218  
Analog input pin 0: 3409  
Analog input pin 0: 5609  
Analog input pin 0: 22175  
Analog input pin 0: 26755  
Analog input pin 0: 10569  
Analog input pin 0: 2294  
Analog input pin 0: 14
```

☒ Autoscroll ☐ Show timestamp Newline 9600 baud Clear output

Collegamento del modulo con Raspberry Pi

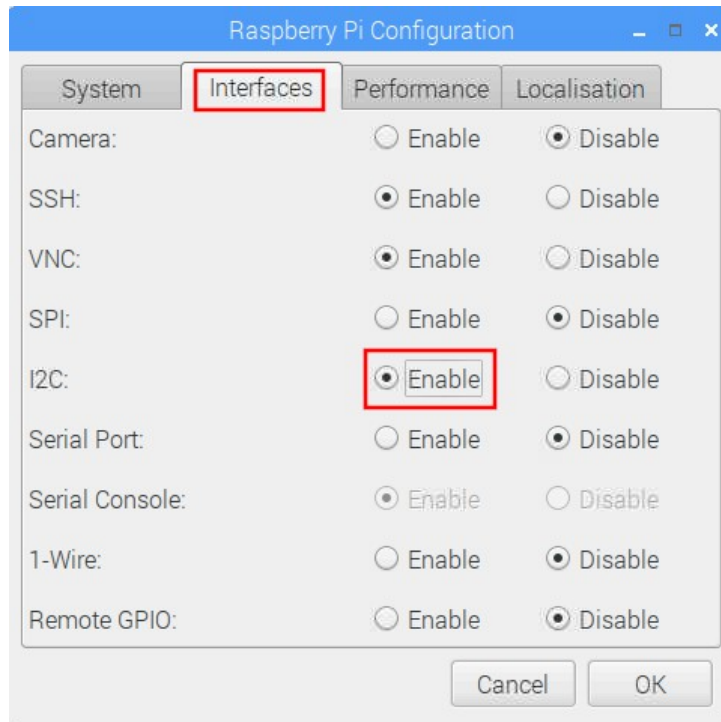
Raspberry Pi non ha il convertitore da analogico a digitale, il che significa che non ha la capacità di leggere le tensioni analogiche. Quindi questo modulo analogico a digitale ADS1115 è perfetto per Raspberry Pi. Dà la possibilità a Raspberry Pi di leggere tensioni analogiche.

La prima cosa da fare è abilitare l'interfaccia I2C di Raspberry Pi. Nel tuo Rasbian, vai su Start> Preferenze> Configurazione Raspberry Pi.



Az-Delivery

Questo aprirà una nuova finestra, apri la seconda scheda "Interfacce" e abilita l'interfaccia I2C e fai clic sul pulsante ok come nell'immagine qui sotto.



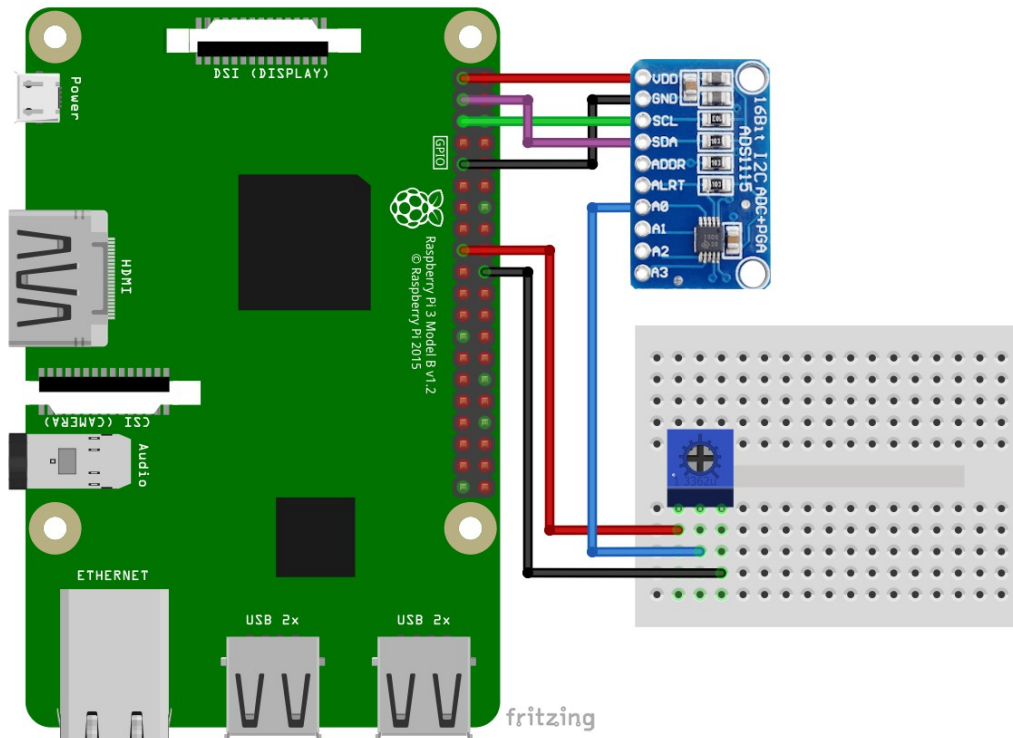
Ora hai abilitato l'interfaccia I2C hardware sui pin GPIO:

GPIO2 > SDA

GPIO3 > SCL

Az-Delivery

Collegare il modulo con Raspberry Pi come nello schema di collegamento seguente. Usiamo il potenziometro solo per esempio, per leggere alcune variazioni della tensione analogica.



Pin Modulo > Pin Raspberry

VDD > 3.3V [pin 1]

GND > GND [pin 9]

SCL > GPIO 3 [pin 5]

SDA > GPIO 2 [pin 3]

Pin Modulo > Pin potenziometro

A0 > pin centrale

RaspPi Pin > Pin potenziometro

3.3V [pin 17] > Pin sinistro

GND [pin 20] > Pin destro

Filo rosso

Filo nero

Filo verde

Filo viola

Filo blu

Filo arancio

Filo nero



Script Raspberry Pi

Prima di iniziare a utilizzare questo modulo, dobbiamo installare la libreria per esso. La libreria è "Adafruit_Python_ADS1x15". Apri quindi l'app terminal in Raspbian ed esegui questi comandi uno alla volta:

```
sudo apt-get update
sudo apt-get install build-essential python-dev python-smbus git
git clone https://github.com/adafruit/Adafruit\_Python\_ADS1x15
cd Adafruit_Python_ADS1x15
sudo python3 setup.py install
```

Dopo aver installato questa libreria, creiamo un nuovo file chiamato "*AnalogRead.py*" e inseriamo il prossimo codice di script:

```
import time
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115() # Create an ADS1115 ADC (16-bit) instance
GAIN = 1
print('[press ctrl+c to end the script]')
try: # Main program loop
    while True:
        values = adc.read_adc(0, gain=GAIN) # Read the ADC channel 0 value
        print('{0:>6}'.format(values))
        time.sleep(0.5)

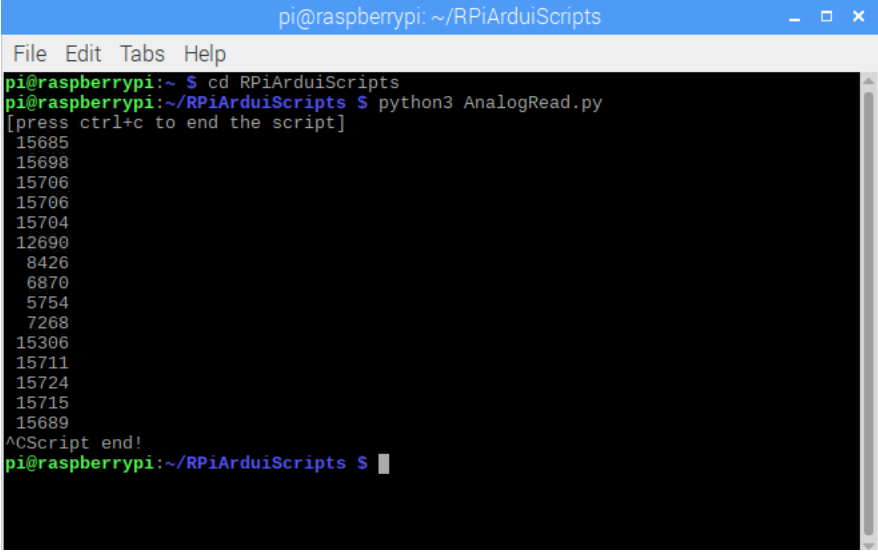
# Scavenging work after the end of the program
except KeyboardInterrupt:
    print('Script end!')
```

Az-Delivery

Per eseguire lo script, apri l'app terminal in Raspbian ed esegui il comando seguente:

`Python3 AnalogRead.py`

Quando si sposta l'albero del potenziometro l'output dovrebbe assomigliare a questo:

A screenshot of a terminal window titled 'pi@raspberrypi: ~/RPiArduiScripts'. The terminal shows the following commands and output:

```
pi@raspberrypi:~ $ cd RPiArduiScripts
pi@raspberrypi:~/RPiArduiScripts $ python3 AnalogRead.py
[press ctrl+c to end the script]
15685
15698
15706
15706
15704
12690
8426
6870
5754
7268
15306
15711
15724
15715
15689
^CScript end!
pi@raspberrypi:~/RPiArduiScripts $
```

Per fermare lo script, premi **CTRL + C**.

Ce l'hai fatta, ora puoi usare il tuo modulo per i tuoi progetti.



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>