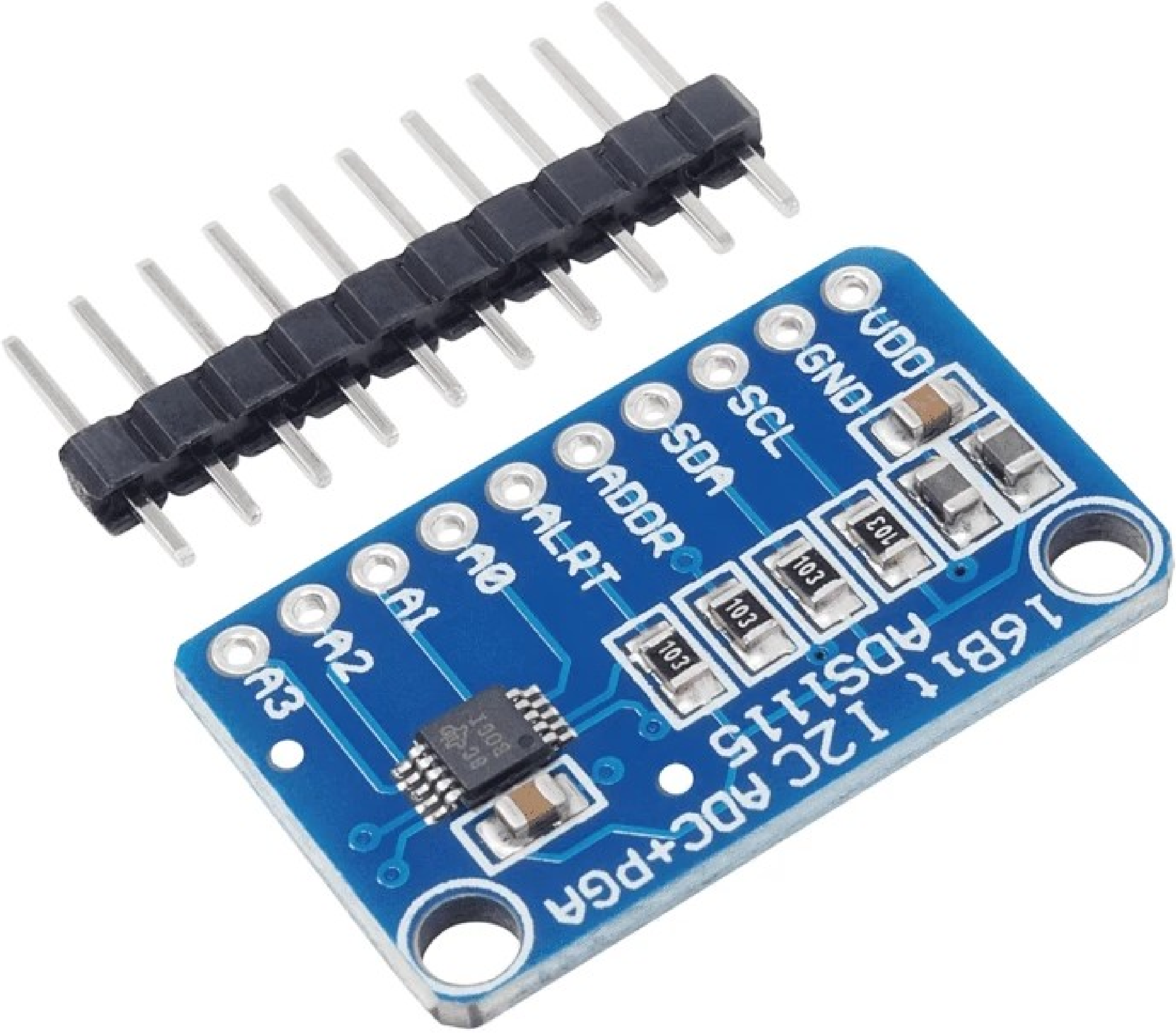
Welcome!

Thank you very much for purchasing our AZ-Delivery Analog - Digital converter ADS1115. On the following pages, we will introduce you to how to use and setup this handy device.

Have fun!



The ADS1115 module is a four channel breakout board. These modules are perfect for adding high-resolution analog to digital conversion to any microprocessor-based (like Raspberry Pi) project or if you want to add analog to digital converter with higher accuracy to and microcontroller-based (like Arduino) project.

This module can run with power and logic signals between 2V to 5V, so they are compatible with all common 3.3V and 5V processors.

As many of 4 of these boards can be controlled from the same I2C bus. This is giving you up to 16 single-ended or 8 differential channels.

A programmable gain amplifier provides up to x16 gain for small signals.

Specifications

» Resolution:	16 Bits
» Programmable Sample Rate:	from 8 to 860 samples per second
» Power Supply and Logic Levels:	2.0V to 5.5V
» Low Current Consumption:	Continuous Mode: 150μA
» Single-Shot Mode:	Auto Shut-Down
» Internal Gain:	up to x16
» I2C Interface:	4 pin Selectable Addresses
» Inputs:	4 Single-Ended or 2 Differential Inputs
» Programmable Comparator	
» Internal Low-Drift Voltage Reference	
» Internal Oscillator	

I2C Addressing

The ADS1115 chip have a 7 bit I2C address, default of *0x48* and addressing scheme that allows four different addresses using just one address pin named *ADDR*.

To setup the address, connect the address pin as follows:

ADR -> GND	*0x48*
ADR -> VDD	*0x49*
ADR -> SDA	*0x4A*
ADR -> SCL	*0x4B*

This addressing scheme allows us to connect four different modules on the same microprocessor.

Single Ended vs. Differential Inputs

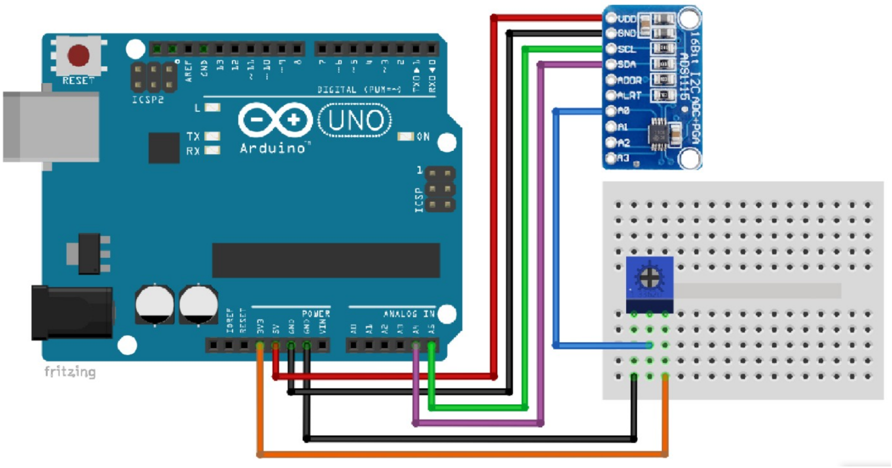
The module support four single ended or two differential inputs. Single ended inputs measure the voltage between the analog input channel (any of A0, A1, A2 and A3 pins) and analog ground (GND pin). Differential inputs measure the voltage between two analog input channels, A0 and A1 pins or A2 and A3 pins.

Single ended inputs give you four analog inputs. By definition, single ended inputs only measure positive voltages. You can only get an effective 15 bit resolution.

Differential measurements offer more immunity from electromagnetic noise (when using long signal wires or operating in an electrically noisy environment). This is also desirable when dealing with small signals requiring high gain, since the gain will amplify the noise as well as the signal. Differential inputs provide the 16 bit resolution and the ability to measure negative voltages.

Connecting the module with Arduino Uno

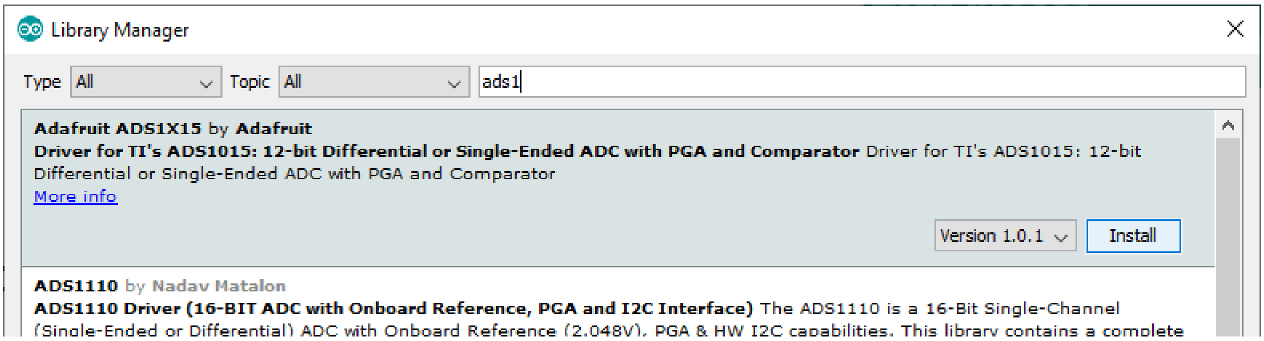
Arduino Uno already have the analog input pins, but why would we connect another analog to digital converter to Arduino? The module is more precise than the inbuilt analog to digital converter inside the microcontroller onboard the Arduino. So if you need more precise analog to digital converter, you can use this module with arduino. Connect everything like on diagram below, we use potentiometer just for example, to read some changing analog voltage:



Module pin	**>**	**Arduino pin**	
VDD	>	5V	**Red wire**
GND	>	GND	**Black wire**
SCL	>	A5	**Green wire**
SDA	>	A4	**Purple wire**
Module pin	**>**	**Potentiometer pin**	
A0	>	Middle pin	**Blue wire**
Arduino Pin	**>**	**Potentiometer pin**	
3.3V	>	Right pin	**Orange wire**
GND	>	Left pin	**Black wire**

Arduino IDE library

We have first to download a library for our module. Open your Arduino IDE and go to *Tools* > *Manage Libraries* and in the search box type "*ADS1115*" and download the library from the Adafruit, like on the image below:



Library Manager

Type All Topic All ads1

Adafruit ADS1X15 by **Adafruit**
Driver for TI's ADS1015: 12-bit Differential or Single-Ended ADC with PGA and Comparator Driver for TI's ADS1015: 12-bit Differential or Single-Ended ADC with PGA and Comparator
More info

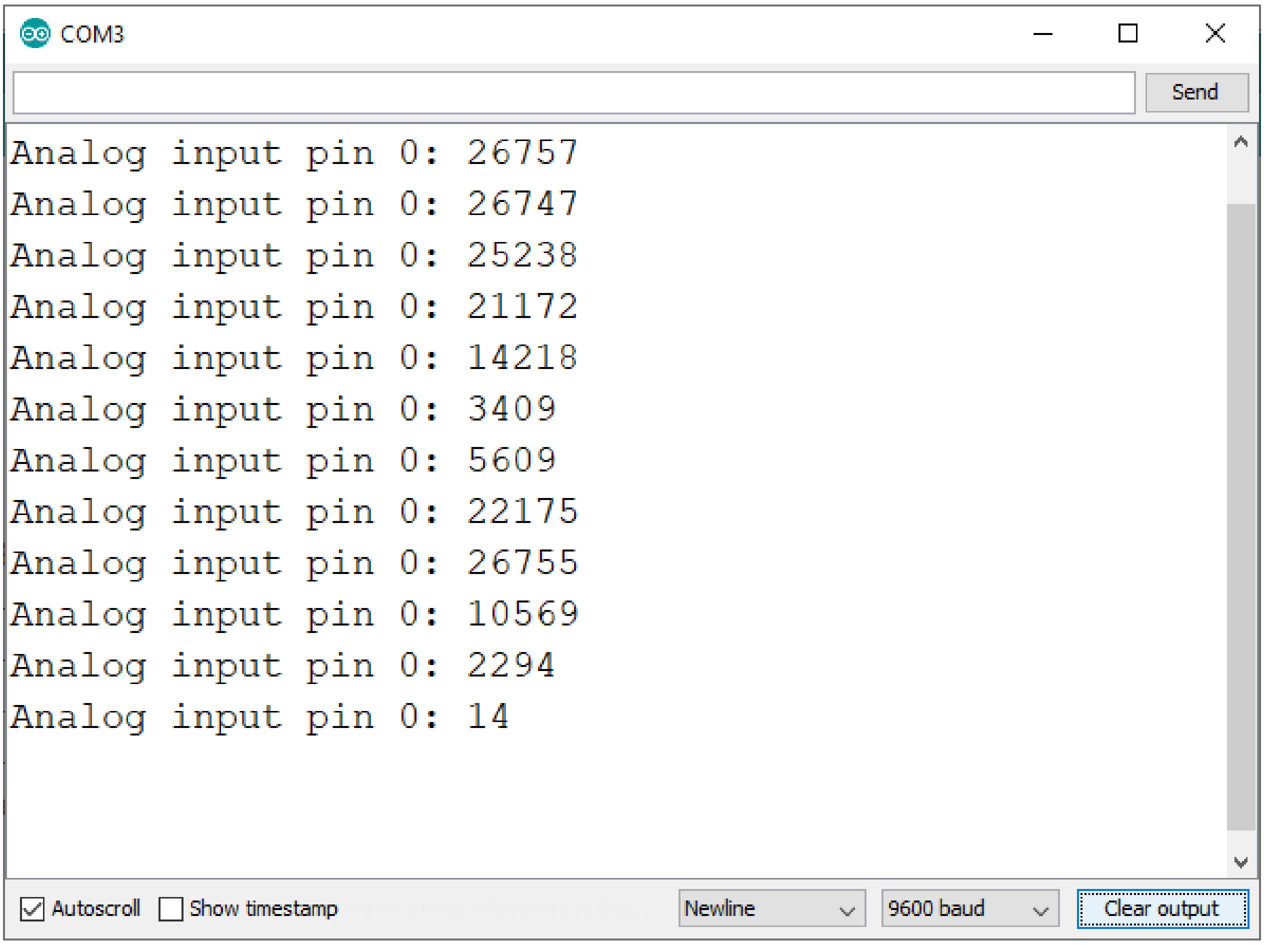
Version 1.0.1 Install

ADS1110 by **Nadav Matalon**
ADS1110 Driver (16-BIT ADC with Onboard Reference, PGA and I2C Interface) The ADS1110 is a 16-Bit Single-Channel (Single-Ended or Differential) ADC with Onboard Reference (2.048V), PGA & HW I2C capabilities. This library contains a complete

Arduino code:

```
#include <Wire.h>
#include <Adafruit_ADS1015.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);
}
```

And when you start Serial Monitor (*Tools > Serial Monitor*) the output should look like this when you move the potentiometer shaft:



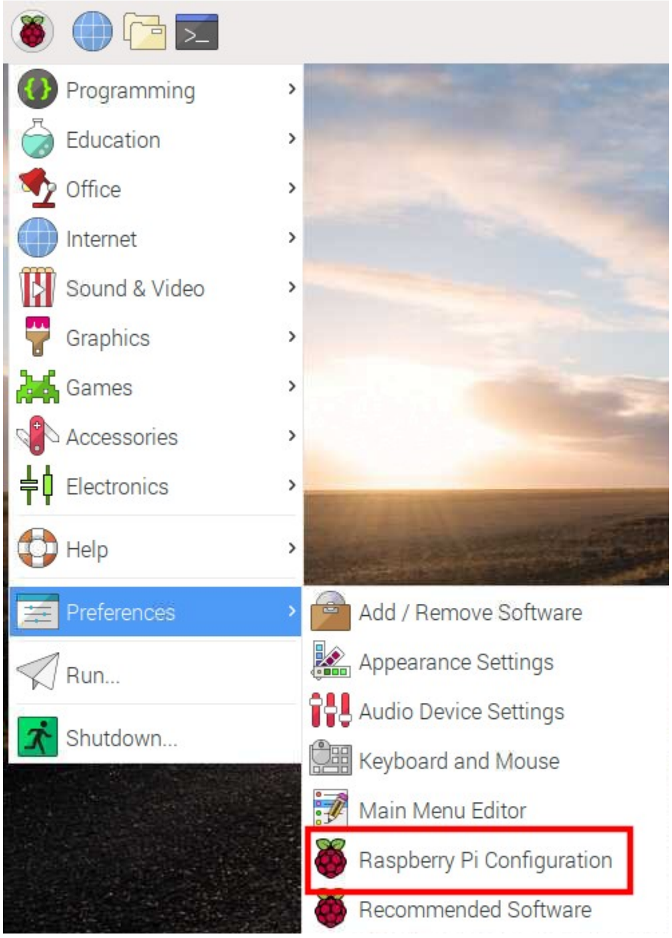
```
Analog input pin 0: 26757
Analog input pin 0: 26747
Analog input pin 0: 25238
Analog input pin 0: 21172
Analog input pin 0: 14218
Analog input pin 0: 3409
Analog input pin 0: 5609
Analog input pin 0: 22175
Analog input pin 0: 26755
Analog input pin 0: 10569
Analog input pin 0: 2294
Analog input pin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

☑ Autoscroll □ Show timestamp Newline 9600 baud

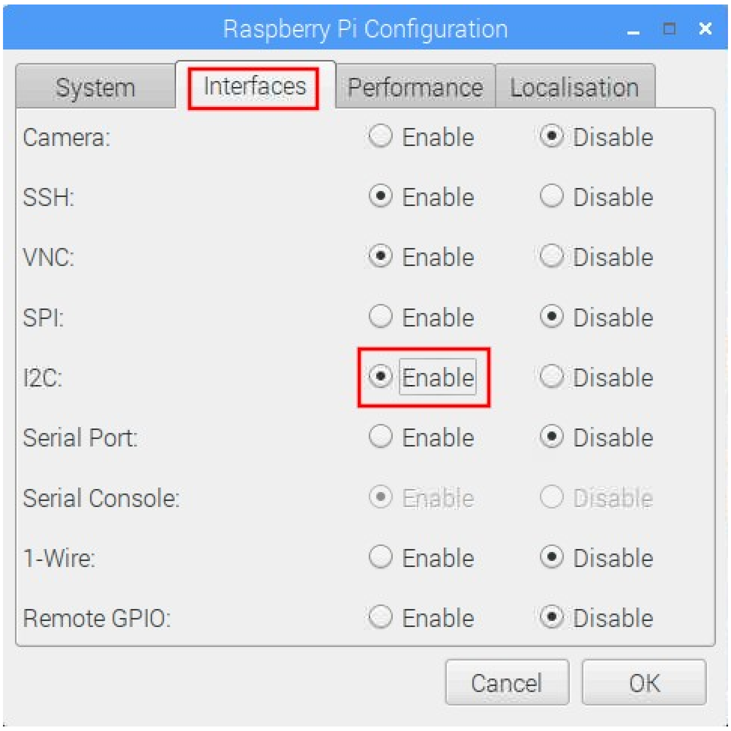
Connecting the module with Rapberry Pi

Raspberry Pi doesn’t have the analog to digital converter, which means that it doesn’t have the ability to read the analog voltages. So this analog to digital ADS1115 module is perfect for Raspberry Pi. It gives ability to Raspberry Pi to read analog voltages.

First thing to do is to enable the I2C interface of Raspberry Pi. In your Rasbian, go to Start > Preferences > Raspberry Pi Configuration.



This will open a new window, open the second tab “*Interfaces*” and enable the I2C interface, and click the ok button like on image below.

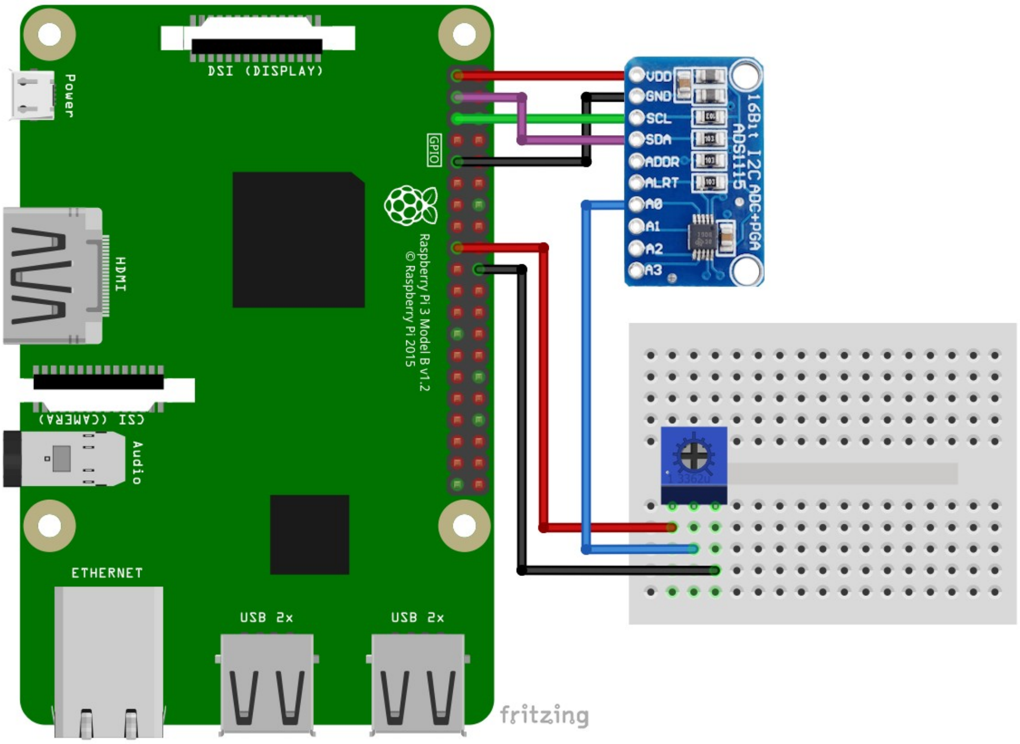


Camera:	○ Enable	● Disable
SSH:	● Enable	○ Disable
VNC:	● Enable	○ Disable
SPI:	○ Enable	● Disable
I2C:	● Enable	○ Disable
Serial Port:	○ Enable	● Disable
Serial Console:	● Enable	○ Disable
1-Wire:	○ Enable	● Disable
Remote GPIO:	○ Enable	● Disable

You now enabled hardware I2C interface on GPIO pins:

GPIO2 > SDA
GPIO3 > SCL

Connect the module with Raspberry Pi like on connection diagram below. We use potentiometer just for example, to read some changing analog voltage.



Module pin	**>**	**Raspberry pin**		
VDD	>	3.3V	[pin 1]	Red wire
GND	>	GND	[pin 9]	**Black wire**
SCL	>	GPIO 3	[pin 5]	**Green wire**
SDA	>	GPIO 2	[pin 3]	**Purple wire**
Module pin	**>**	**Potentiometer pin**		
A0	>	Middle pin		**Blue wire**
RaspPi Pin	**>**	**Potentiometer pin**		
3.3V [pin 17]	>	Left pin		**Orange wire**
GND [pin 20]	>	Right pin		**Black wire**

Raspberry Pi script

Before we start using this module, we have to install the library for it. The library is “*Adafruit_Python_ADS1x15*”. So open the terminal app in your Raspbian and run these commands one by one:

```
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
```

After this library is installed, lets create a new file called “*AnalogRead.py*” and put in it the next script code:

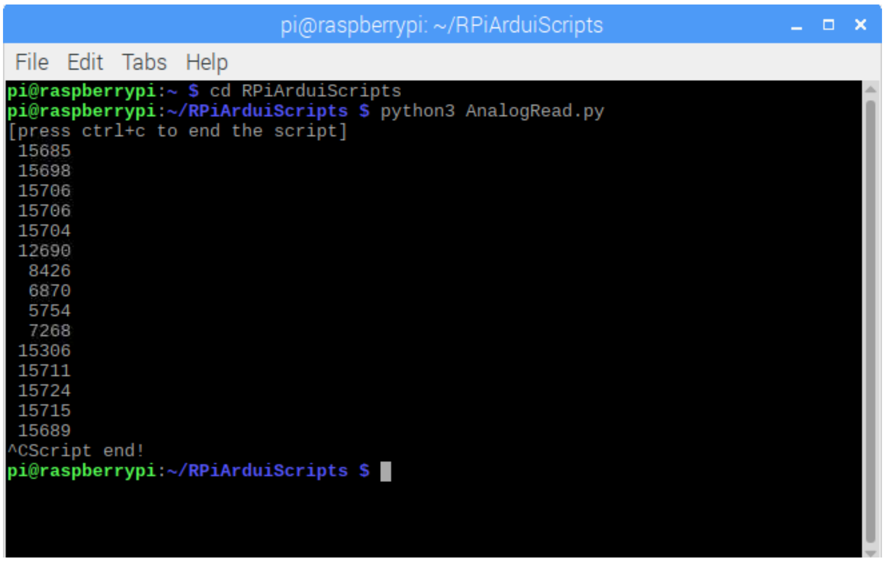
```python
import time
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115() # Create an ADS1115 ADC (16-bit) instance
GAIN = 1
print('[press ctrl+c to end the script]')
try: # Main program loop
    while True:
        values = adc.read_adc(0, gain=GAIN) # Read the ADC channel 0 value
        print('{0:>6}'.format(values))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

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

To run the script open terminal app in the Raspbian, and run the next command:

```
Python3 AnalogRead.py
```

Output should look like this, when you move the potentiometer shaft:



```
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 $ 
```

To stop the script, press *CTRL + C*.

You've done it, you can now use your module for your projects.

Now it is time to learn and make the Projects on your own. You can do that with the help of many example scripts and other tutorials, which you can find on the internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de

Have Fun!

Impressum

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