

Practical 4

Aim:

Raspberry Pi Based Oscilloscope

Additional Hardware required:

1. ADS1115 ADC
2. Jumper wires

ADS1115 and Raspberry Pi Connections:

VDD – 3.3v (pin 1 on pi)
GND – GND (pin 6 of Pi)
SDA – SDA (pin 3 of pi)
SCL – SCL (pin 5 of pi)

Software Configuration required:

Install Dependencies for Raspberry Pi Oscilloscope:

Step 1: Enable Raspberry Pi I2C interface

To enable the I2C, from the terminal, run;

sudo raspi-config

When the configuration panels open, select interface options, select I2C and click enable

Step 2: Update the Raspberry pi

To do this, run below two commands:

sudo apt update

Step 3: Install the Adafruit ADS1115 library for ADC

git clone

https://github.com/adafruit/Adafruit_Python_ADS1x15.git

Extract the file.

copy ADS1x15.py to the current working folder(when the source code of program will be saved)

sudo python setup.py install

Step 4: Install *Matplotlib*

To visualize the data we need to install the *matplotlib* module which is used to plot all kind of graphs in python. This can be done by running;

```
sudo apt install python3-matplotlib
```

Step 5: Install drawnow

Type following command

```
python3 -m pip config set global.break-system-packages true
```

Now to install Drawnow, type the following command

```
pip install drawnow
```

```
pip3 install adafruit-circuitpython-ads1x15
```

Write following code in python 3 IDLE and save it as 'osci.py'

Raspberry Pi Oscilloscope Code:

```
import time
import board
import busio
import matplotlib.pyplot as plt
from drawnow import *
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
GAIN = 1
val = [ ]
cnt = 0
plt.ion()

# Initialize I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Create ADS1115 instance
ads = ADS.ADS1115(i2c)

# Create a single-ended input on channel 0
chan = AnalogIn(ads, ADS.P0)

def makeFig():
    plt.ylim(-4500,5000)
    plt.title('Oscilloscope')
    plt.grid(True)
    plt.ylabel('ADC outputs')
```

```
plt.plot(val, 'ro-', label='Channel 0')
plt.legend(loc='lower right')
```

```
print("Reading ADS1115 Values...")
```

```
while True:
```

```
    print(f"Raw Value: {chan.value}, Voltage: {chan.voltage:.3f}V")
    value= adc.get_last_result()
    print('Channel 0: {0}'.format(value))
    time.sleep(0.5)
    val.append(int(value))
    drawnow(makeFig)
    plt.pause(.000001)
    cnt = cnt+1
    if(cnt>50):
        val.pop(0)
```