

```
#!/usr/bin/env python

# Simple string program. Writes and updates strings.
# Demo program for the I2C 16x2 Display from Ryantek.uk
# Created by Matthew Timmons-Brown for The Raspberry Pi Guy YouTube channel

# Import necessary libraries for communication and display use
import board
import RPi.GPIO as gpio
import drivers

from time import sleep
import Adafruit_ADS1x15

buzz=21
gpio.setmode(gpio.BCM)
gpio.setup(buzz,gpio.OUT)

# Load the driver and set it to "display"
# If you use something from the driver library use the "display." prefix first
display = drivers.Lcd()

# Or create an ADS1015 ADC (12-bit) instance.
adc = Adafruit_ADS1x15.ADS1015()

# Choose a gain of 1 for reading voltages from 0 to 4.09V.
# Or pick a different gain to change the range of voltages that are read:
# - 2/3 = +/-6.144V
# - 1 = +/-4.096V
# - 2 = +/-2.048V
# - 4 = +/-1.024V
```

```
# - 8 = +/-0.512V
# - 16 = +/-0.256V

# See table 3 in the ADS1015/ADS1115 datasheet for more info on gain.
GAIN = 1
```

```
# Main body of code
```

```
try:
```

```
    while True:
```

```
        # Read all the ADC channel values in a list.
```

```
        temp=adc.read_adc(0 , gain=GAIN)
```

```
        print('t')
```

```
        print(temp)
```

```
        print(temp&0xFFF)
```

```
#    te=(((temp*3.3)/1647)-0.5)/0.01
```

```
        t=((temp*3.3)/1648);
```

```
        #tr = ((temp * 330)/float(4095))
```

```
        #tempr = round(t,2)
```

```
        te=(t)/float(110)
```

```
        tem=round(te*10000)-float(50)
```

```
        print (t);
```

```
        #print(te);
```

```
        #print(tem)
```

```
        #print(tr)
```

```
#    print(tempr)
```

```
        print("Writing to display")
```

```
        if tem >=100:
```

```
            gpio.output(buzz,True)
```

```
        else :
```

```
            gpio.output(buzz,False)
```

```
display lcd_display_string("Temperature:", 1) # Write line of text to first line of display
display lcd_display_string(str(tem), 2) # Write line of text to first line of displa
sleep(2)
```

```
display lcd_clear() # Clear the display of any data
```

```
except KeyboardInterrupt:
```

```
# If there is a KeyboardInterrupt (when you press ctrl+c), exit the program and cleanup
```

```
print("Cleaning up!")
```

```
display lcd_clear()
```