2015

TENET TECHNETRONICS

Reading analog value from LM35 temperature sensor on Raspberry Pi using MCP3008

Author: Palani K

Introduction:

Raspberry Pi is a credit card sized computer that plugs into a computer monitor or TV, and uses standard keyboard and mouse. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. Here we are going to read analog value from LM35 temperature sensor by using MCP3008.

Hardware Requirements:

1. Raspberry Pi board.
2. Tenet Power supply breakout board
3. Jumper wires.
4. MCP3008 IC (ADC).
5. Tenet LM35 breakout board.

MCP 3008 IC:

The MCP3008 10-bit Analog-to-Digital Converter (ADC) combines high performance and low power consumption in a small package, making it ideal for embedded control applications. The MCP3008 features a successive approximation register (SAR) architecture and an industry-standard SPI serial interface. The MCP3008 features 200k samples/second, 8 input channels, low power consumption (5nA typical standby, 425µA typical active), and is available in 16-pin PDIP and SOIC packages. Applications for the MCP3008 include data acquisition, instrumentation and measurement, multi-channel data loggers, industrial PCs, motor control, robotics, industrial automation, smart sensors, portable instrumentation and home medical appliances.

Pin diagram:

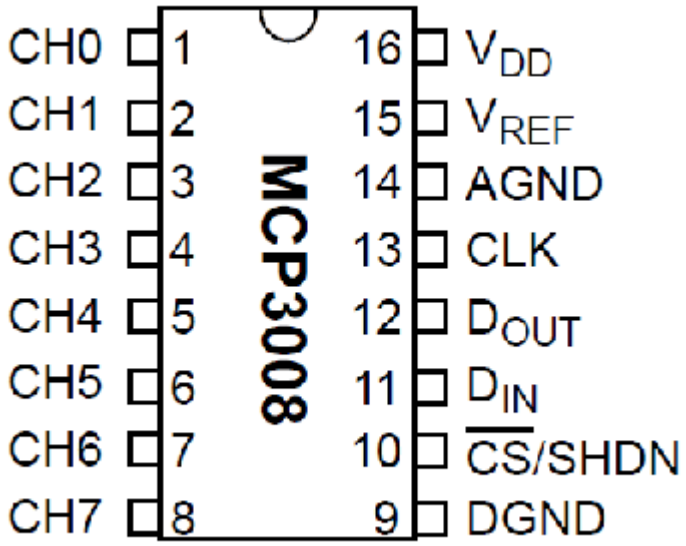


Figure 1

Tenet LM35 breakout board:

The LM35 datasheet specifies that this ICs are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature.

The LM35 thus has an advantage over linear temperature sensors calibrated in ˚ Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4˚C at room temperature and ±3⁄4˚Cover a full −55 to +150˚C temperature range.

Figure 2

SPI Interface:

The Serial Peripheral Interface (SPI) bus was developed by Motorola to provide full-duplex synchronous serial communication between master and slave devices. The SPI bus is commonly used for communication with flash memory, sensors, real-time clocks (RTCs), analog-to-digital converters.

As shown in Figure, standard SPI masters communicate with slaves using the serial clock (SCK), Master Out Slave In (MOSI), Master In Slave Out (MISO), and Slave Select (SS) lines. The SCK, MOSI, and MISO signals can be shared by slaves while each slave has a unique SS line.

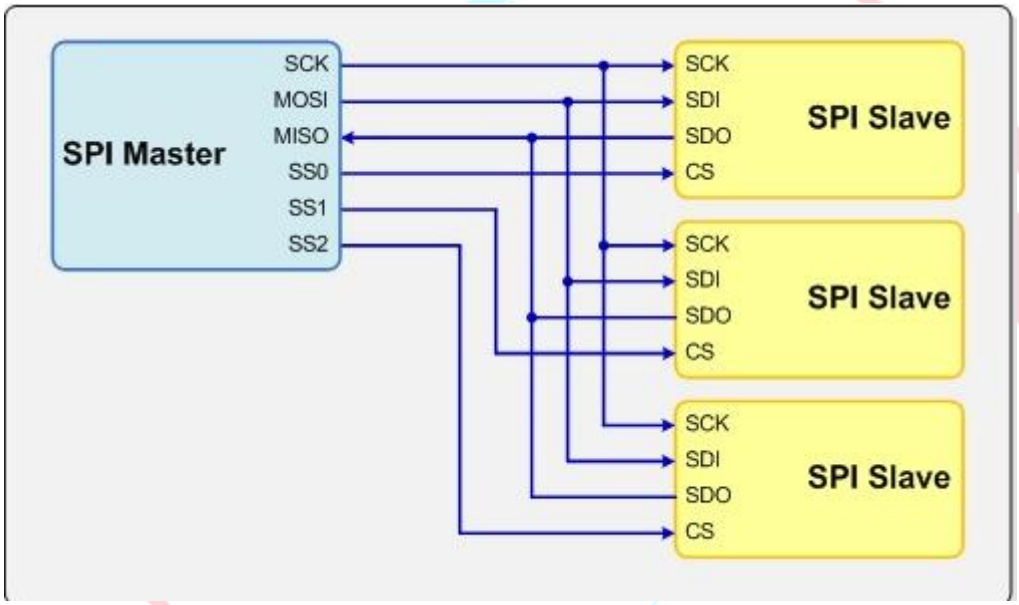


Figure 3

Enabling SPI on raspberry pi:

Step 1: Start by running the following command :

Sudo raspi-config

Step 2: This will launch the raspi-config utility. Select option 8 “Advanced Options”.

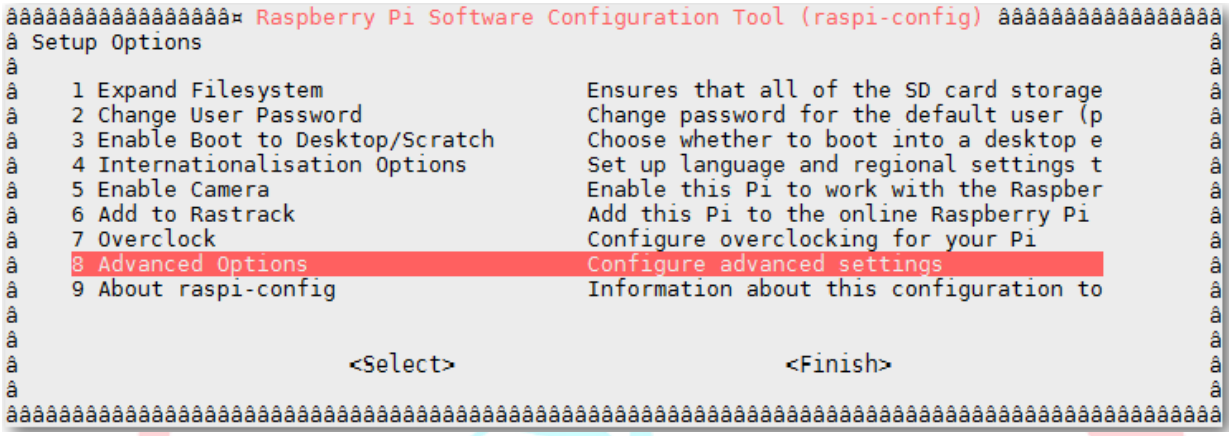


Figure 4

Step 3: Select the “SPI” option.

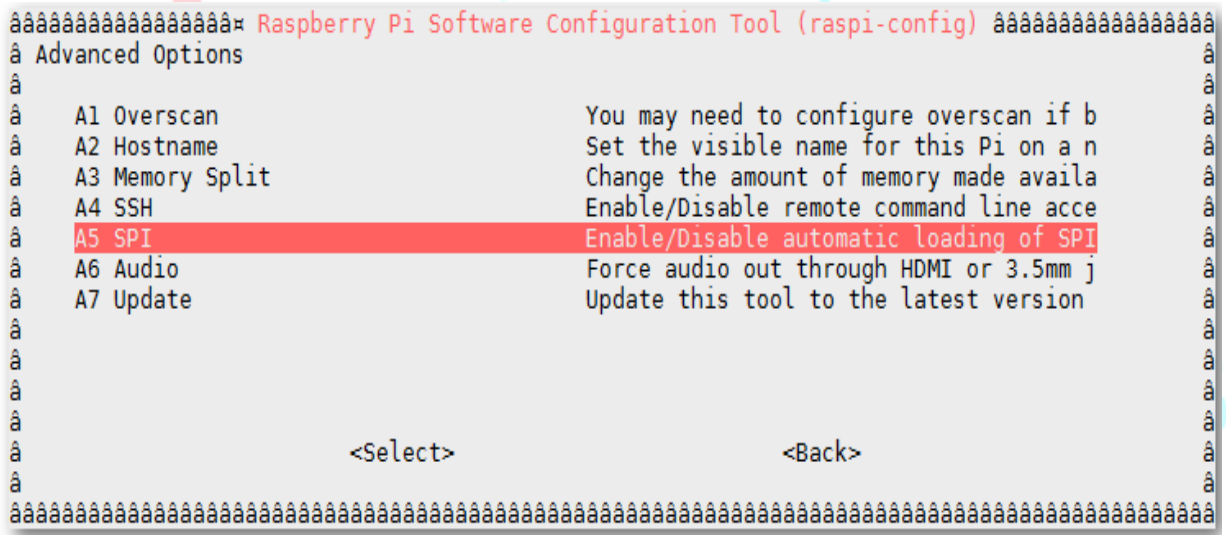


Figure 5

Step 4: Set the option to “Yes”.

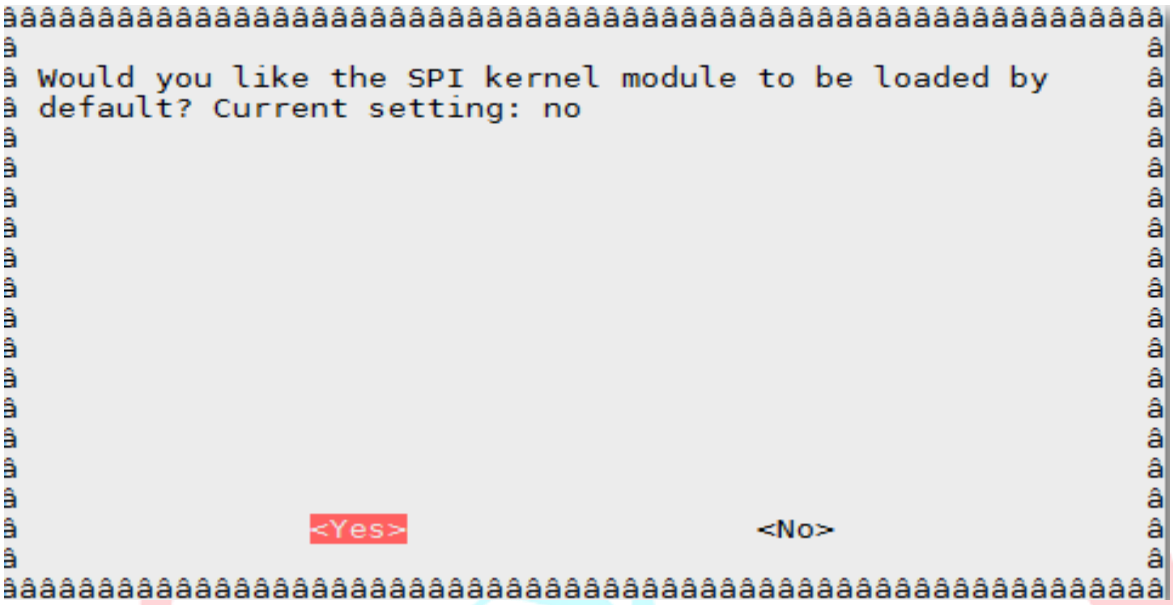


Figure 6

Step 5: Select “OK”.

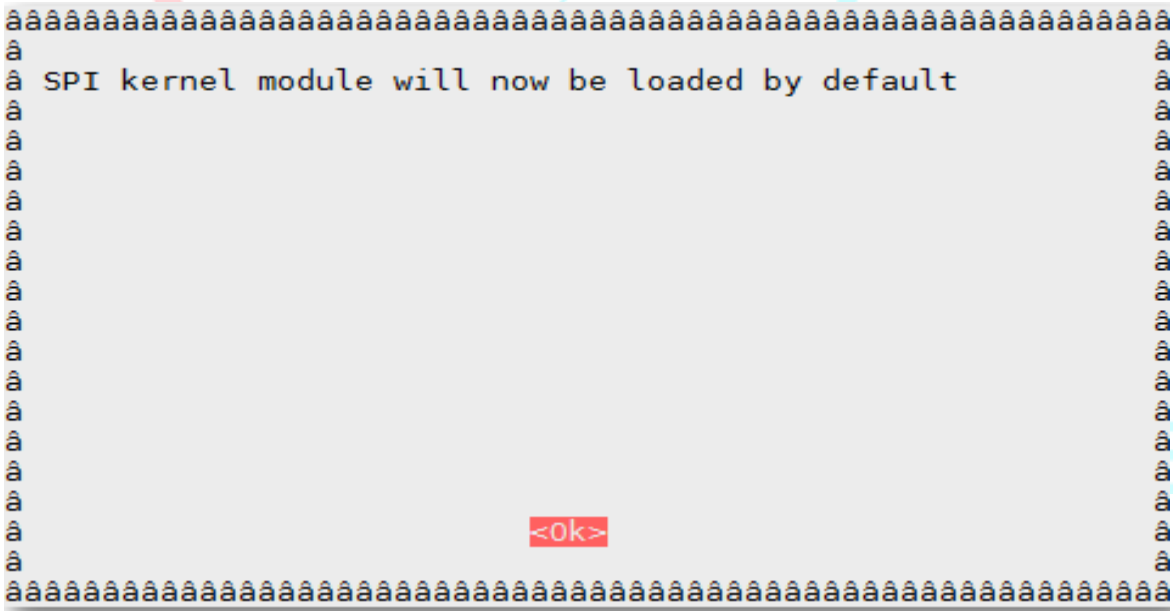


Figure 7

Step 6: Select “Finish”

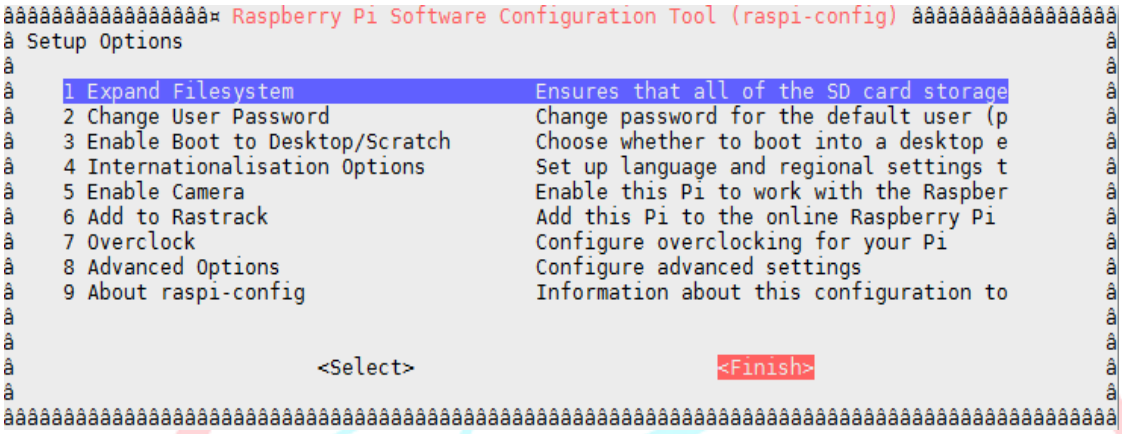


Figure 8

Step 7: Reboot for the changes to take effect .

Sudo reboot

- SPI is now enabled.

Step 8: In order to read data from the SPI bus in Python we can install a library called ‘py-spidev’. To install it we first need to install ‘python-dev’ .

Sudo apt-get install python2.7-dev

Step 9: Then to finish we can download ‘py-spidev’ and compile it ready for use .

wget https://github.com/Gadgetoid/py-spidev/archive/master.zip

unzip master.zip

rm master.zip or rm –r master.zip

cd py-spidev-master

sudo python setup.py install

cd ..

Coding:

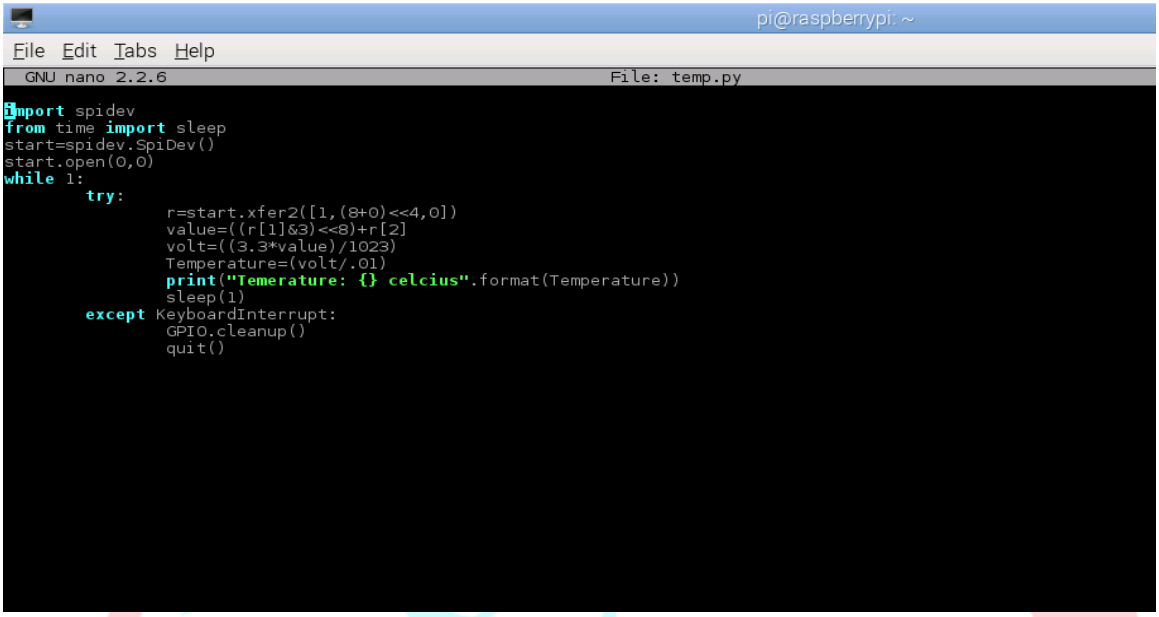


Figure 9

Import spidev //Importing spidev to access SPI

From time import sleep //import sleep for giving delay

Start=spidev.SpiDev() //creating object with name start

Start.open(0,0) //(BUS,channel) since one channel and Bus

While 1:

try: //creating function

r=start.xfer2([1,(8+0)<<4,0]) //enabling SPI and 3 bytes of data stored in r

value=((r[1]&3)<<8)+r[2] //Retrieving last 10 bit

volt=((3.3*value)/1023) //Converting to Voltage

Temperature=volt/.01 //converting to celcius

print("Temperature:{}celcius".format(Temperature) //printing values

excepy KeyboardInterrupt: //when CTRL+C is pressed terminate it

GPIO.cleanup()

quit()

Circuit diagram:

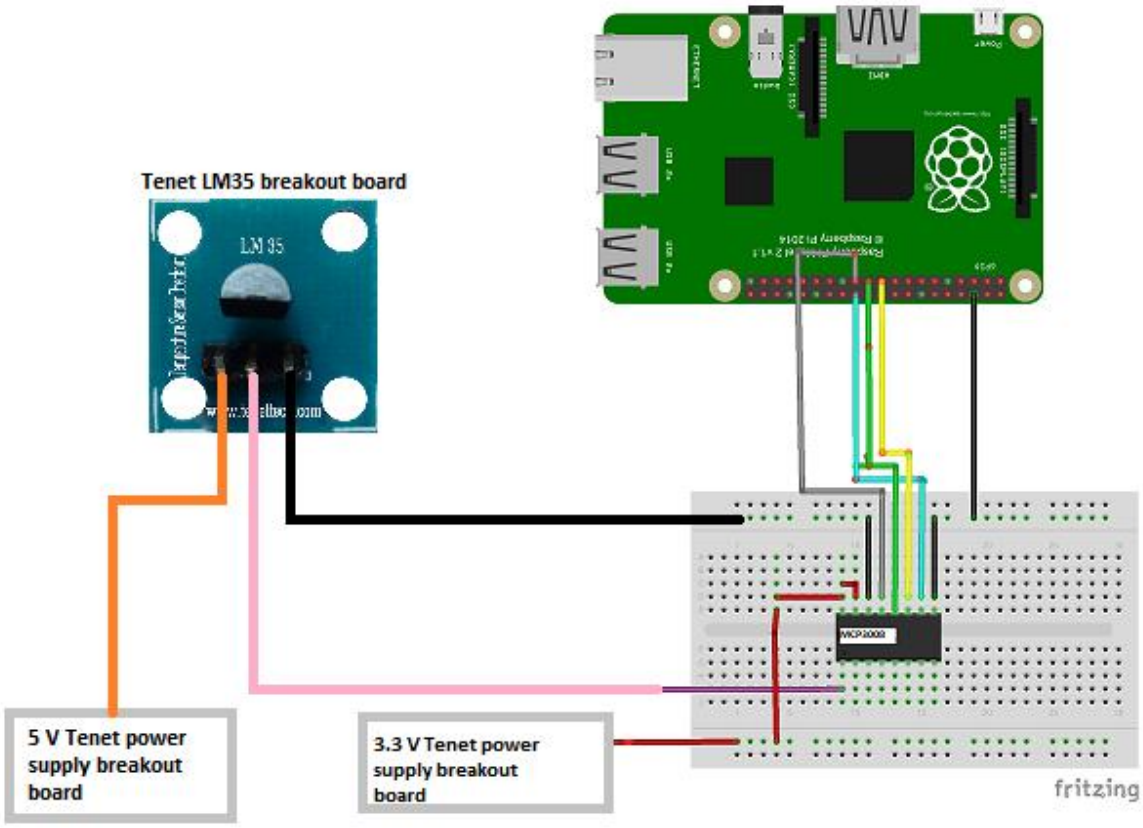
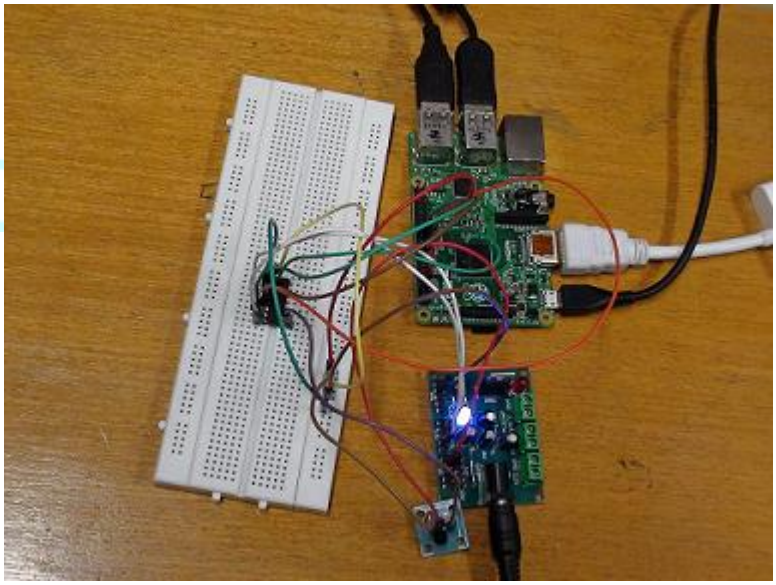


Figure10

Output:



Output on the screen:

```
pi@raspberrypi ~ $ sudo python temp.py
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.1612903226 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.1612903226 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
Temerature: 25.4838709677 celcius
```

Figure 12

For product link:

1. http://tenettech.com/product/7021/raspberry-pi-2-model-b-basic-kit-tt-sp-19022015
2. http://www.tenettech.com/product/6068/power-supply-breakout-board.
3. http://www.tenettech.com/product/2985/mcp3008-8-channel-10-bit-adc-with-spi-interface
4. http://www.tenettech.com/product/3009/lm35-temperature-sensor.

For more information please visit: www.tenettech.com

For technical query please send an e-mail: info@tenettech.com