# DRIVER FOR I2C INTERFACING MCP9808 TEMPERATURE SENSOR

## Summary

This project aims to interface a I2C temperature sensor with the Raspberry Pi. In Linux, device drivers acts as an intermediate between the user application and hardware. I am going to write Linux device driver for MCP9808 temperature sensor which is interfaced through I2C-1 bus. I2C device driver for MCP9808 is compiled as out-of-tree module, which can be loaded into the target Raspberry Pi board during runtime.
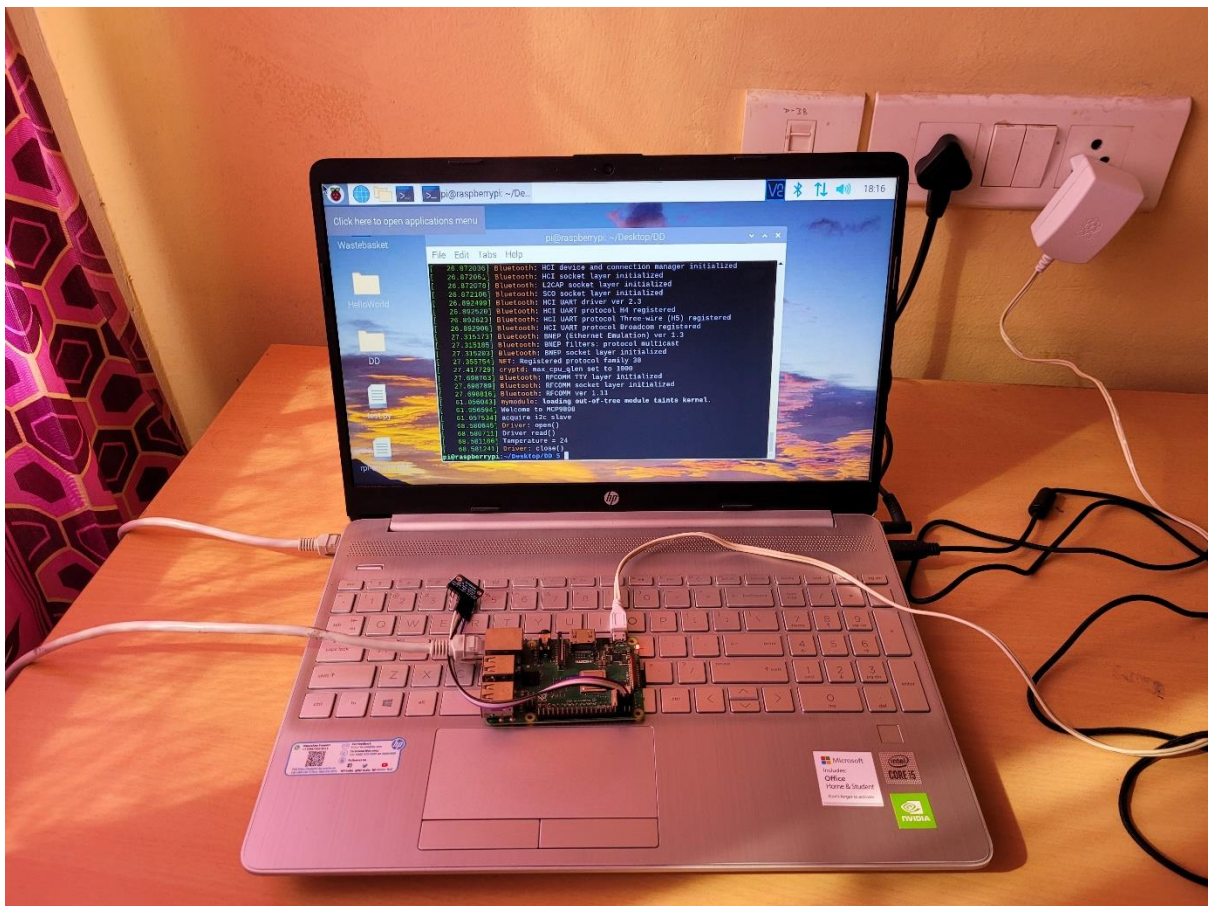
## HARDWARE USED:

- Raspberry pi 3b+ is based on Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz,1 GB LPDDR2 SDRAM.
- MCP 9808 temperature sensor based on i2c.
- Jumper wires

## CONNECTION:
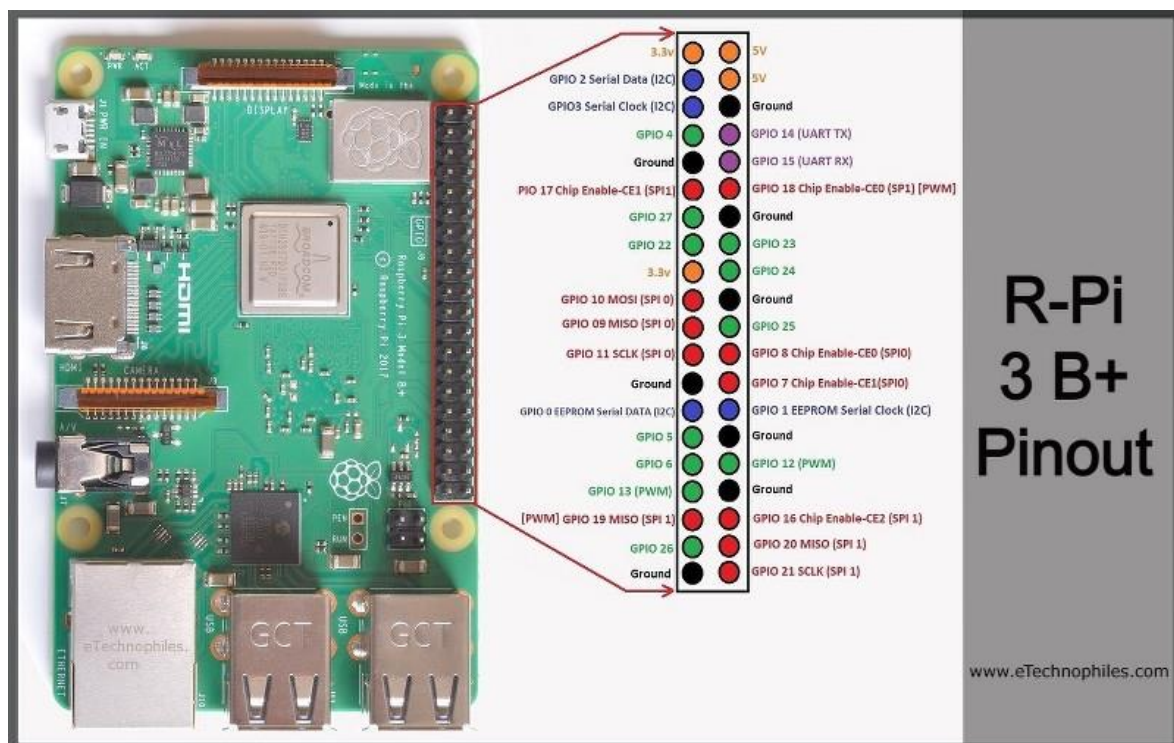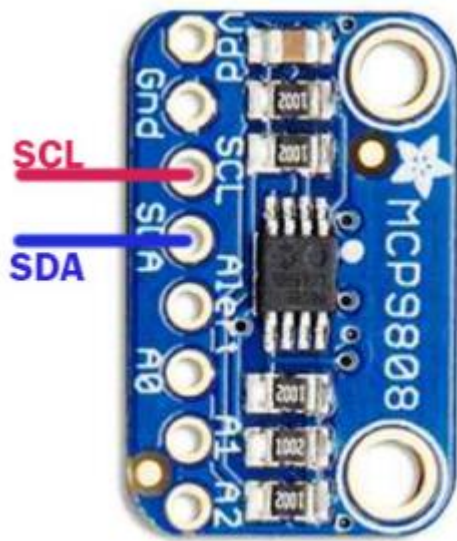
HARDWARE SETUP:



3.3 volt power supply is given to the sensor.

MCP9808 BASIC REGISTER DETAILS:

The MCP9808 has several registers that are user-accessible. These registers include the Temperature register, Configuration register, Temperature Alert

Upper Boundary and Lower Boundary Limit registers, Critical Temperature Limit register, Manufacturer Identification register and Device Identification register. The Temperature register is read-only, used to access the ambient temperature data.

| DFN | MSOP | Symbol | Pin Function |
|-----|------|--------|--------------|
| 1 | 1 | SDA | Serial Data Line |
| 2 | 2 | SCL | Serial Clock Line |
| 3 | 3 | Alert | Temperature Alert Output |
| 4 | 4 | GND | Ground |
| 5 | 5 | A2 | Slave Address |
| 6 | 6 | A1 | Slave Address |
| 7 | 7 | A0 | Slave Address |
| 8 | 8 | VDD | Power Pin |
| 9 | — | EP | Exposed Thermal Pad (EP); must be connected to GND |

Out of these pins SCL, SDA, GND and VDD were used in our project.

AMBIENT TEMPERATURE REGISTER:
The MCP9808 uses a band gap temperature sensor circuit to output analog voltage proportional to absolute temperature. An internal ADC is used to convert the analog voltage to a digital word. The digital word is loaded to a 16-bit read-only Ambient Temperature register (TA) that contains 12-bit temperature data. The Address of this register is 0x05.

**Kernel driver**

In Linux, device drivers acts as an intermediate between the user application and hardware. I am going to write Linux device driver for MCP9808 temperature sensor which is interfaced through I2C-1 bus. I2C device driver for MCP9808 is compiled as out-of-tree module, which can be loaded into the target Raspberry Pi board during runtime.

Build Process:
1.Navigate into the required directory where the module is present.
2.make the modules using the make command to generate the object files.
3.Insert the generated .ko file into the kernel using the sudo insmod command.

4. Make sure the required module is executable by using the chmod command.
5. Cat command is used to read data from the file and gives their content as output.
6. display command dmesg is used to see the required output in the console.

## Compiling & Uploading the firmware

Firmware compilation and uploading procedure can be found inside "/firmware/Readme.txt" file.

**References:**

1. https://www.raspberrypi.com/documentation/computers/linux_kernel.html#configuring-the-kernel
2. https://pdf1.alldatasheet.com/datasheet