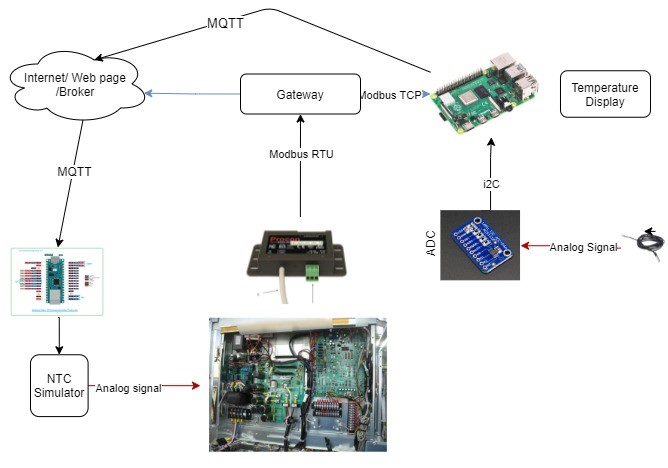
Quick Start Guide



HAN University of Applied Sciences

Trung Nguyen 16/09/2021



**Code uses in this example:**

* [MQTT\_Digitalpot\_meter.py](https://github.com/hancse/FF_Temperature_Calibration/blob/main/src/raspberry/ADS1x15%20python%20example/MQTT_Digitalpot_meter.py)
* [MQTT\_IOT33\_MPC42050](https://github.com/hancse/FF_Temperature_Calibration/tree/main/src/Iot33/MQTT_IOT33_MPC42050)

# Raspberry Pi Temperature reading

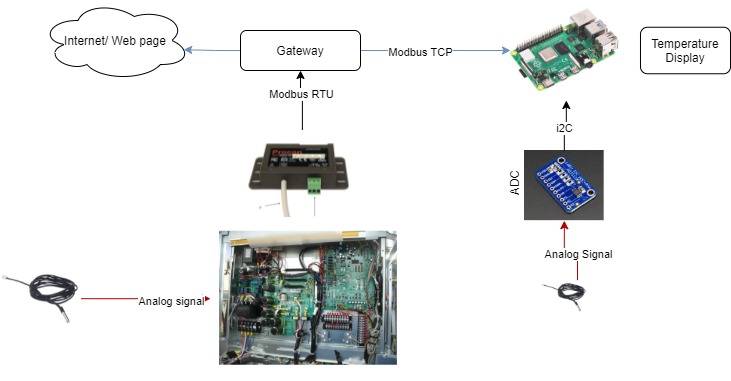


Figure 1: Raspberry Pi 3b with NTC connection

**Component’s list:**

* Raspberry pi 3b.
* ADS1115.
* NTC 10k.

The wiring diagram connection pins from raspberry to ADS1115:

* Pi 3V to ADS1015 VDD - Remember the maximum input voltage to any ADC channel cannot exceed this VDD 3V value.
* Pi GND to ADS1015 GND
* Pi SCL to ADS1015 SCL
* Pi SDA to ADS1015 SDA

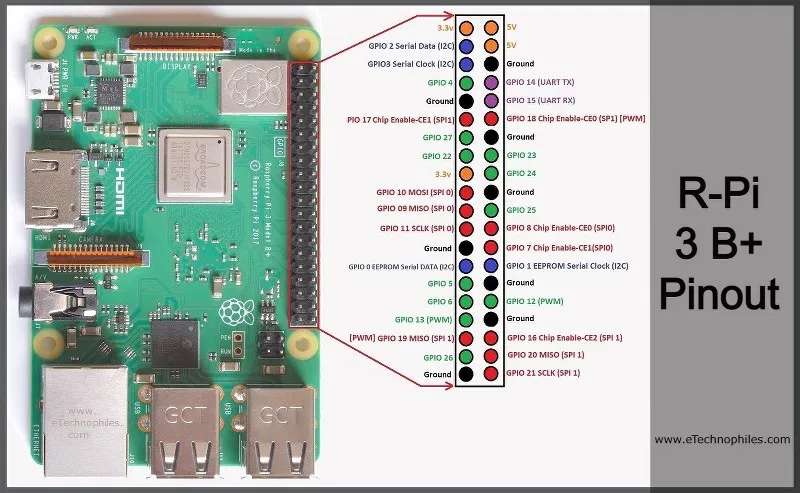
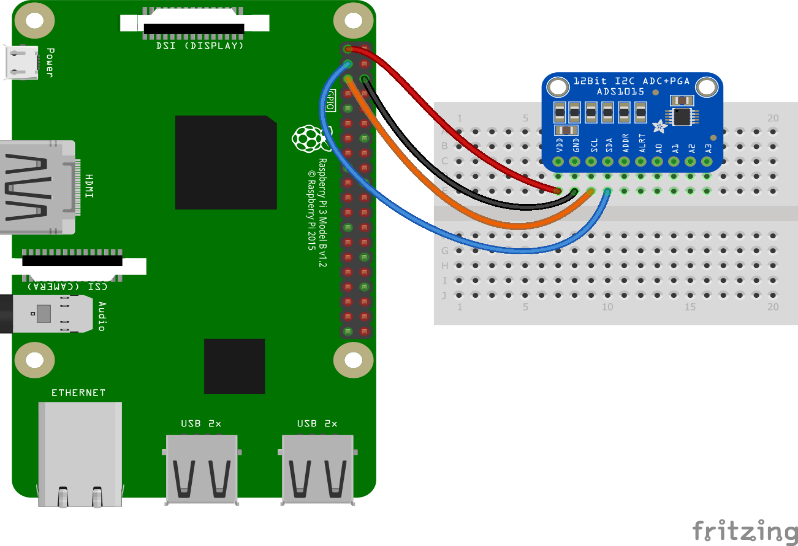


Figure 2: ADS1115 Raspberry wiring diagrams.

The thermistor NTC can connect with any pin from A0 to A3 through voltage divider circuits. The fix resistor R1 needs to be selected based on the value of the sensor for best performance.

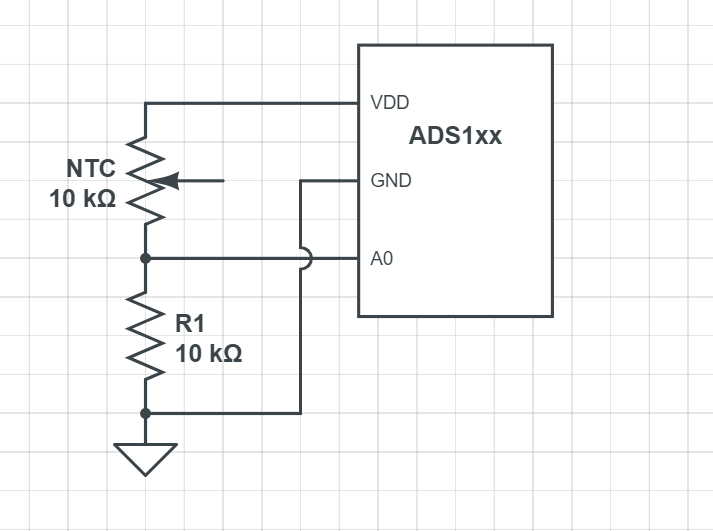


Figure 3: Thermistor analog reading with voltage divider circuit to ADS1x.

**Raspberry pi setup.**

* Flash the Raspberry Pi with a provided image or follow the instructions on the main report.
* Power on the Raspberry Pi.
* Open the file manager and go to the directory: */home/pi/Adafruit\_Python\_ADS1x15/examples (look at* **Python installation of ADS1x15 library in the main report and**[example code on GitHub](https://github.com/hancse/FF_Temperature_Calibration/tree/main/src/raspberry/ADS1x15%20python%20example)*)*
* Double click and run the *MQTT\_Digitalpot\_meter.py* file in figure 3.

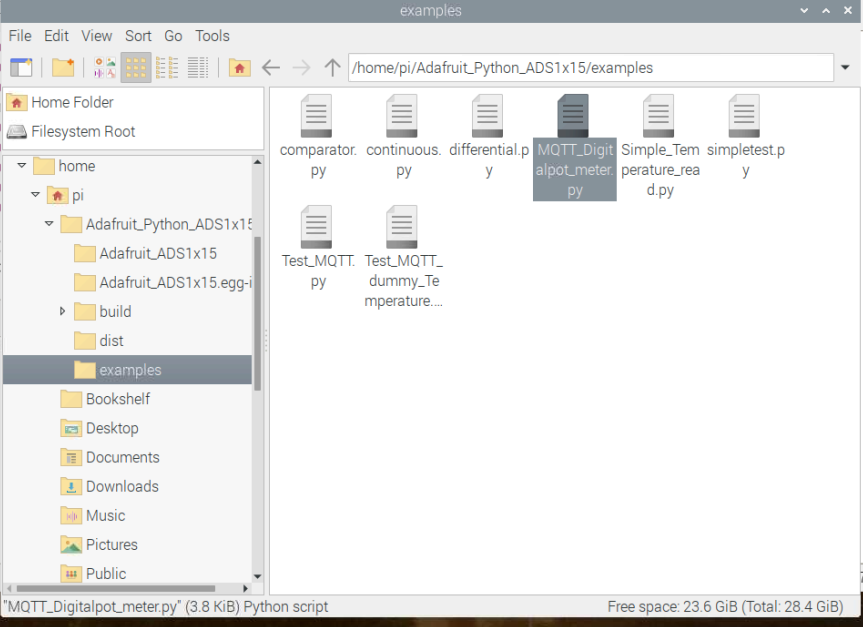


Figure 3: MQTT digital potential meter.

The programmed should print out the resistance value and the temperature value of the room temperature at that moment. For example, 10537 ohm and 23.3-degree C.



Figure 4: Resistance value and room temperature.

The resistance value and corresponding temperature can be checked with the [10k-2 thermistor output table.](https://github.com/hancse/FF_Temperature_Calibration/blob/main/Datasheet/Thermistor/Thermistor_10K-2.pdf)

**Setup the WIFI to connect the Pi to the internet (figure 5,6).**

* Open the terminal and type ifconfig.
* Write down the IP address of the Pi.
* Enter the IP address with port 1880/ui into the web browser <http://192.168.50.147:1880/ui> (Remember to change it to your own address).

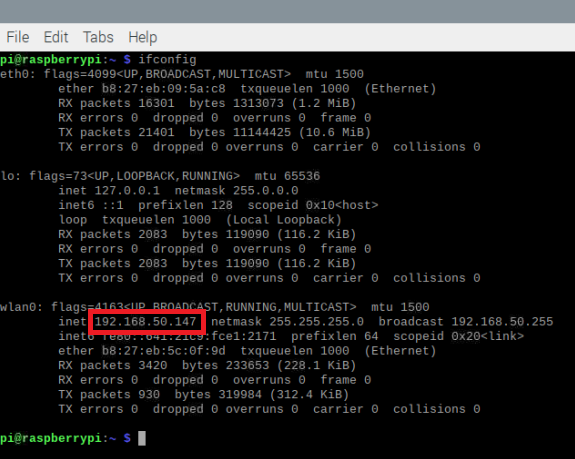


Figure 5: Raspberry Pi IP address.

Select dashboard tab user should see the updated temperature value reading in figure 6.

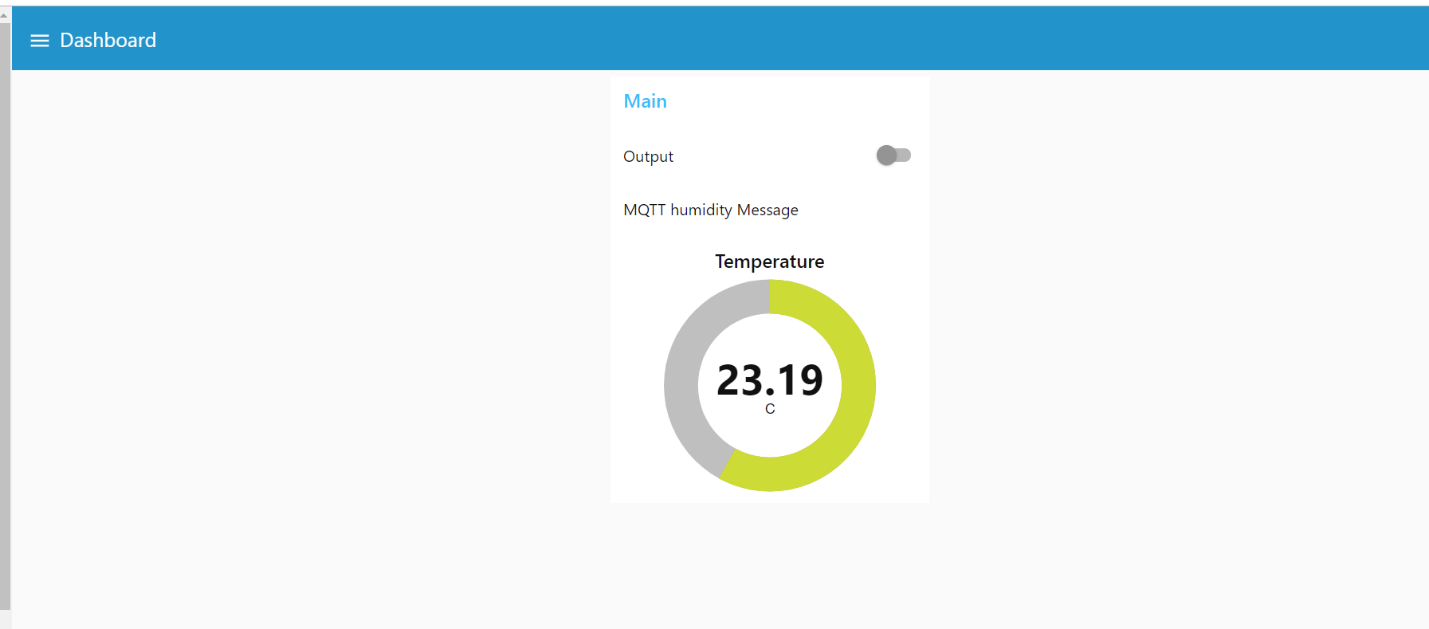


Figure 6: Temperature sensor display.

Users can also use the MQTT client phone or window app using the info in figure 7 and subscribe to the *FZ/Sensor* topic to see the room temperature.

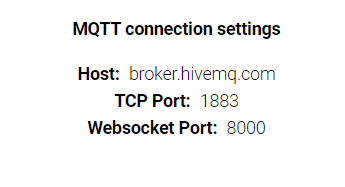


Figure 7: MQTT broker example.

# Control digital potential meter.

Component’s list:

* Arduino IOT33.
* MCP41xx\_42xx.

Before getting start user should install Arduino Ide and read [Getting start with arduino IOT33](https://www.arduino.cc/en/Guide/NANO33IoT) to see how to install the supported libraries.

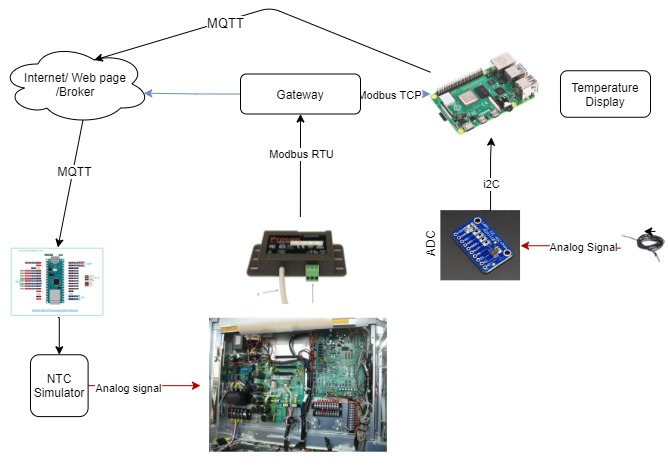
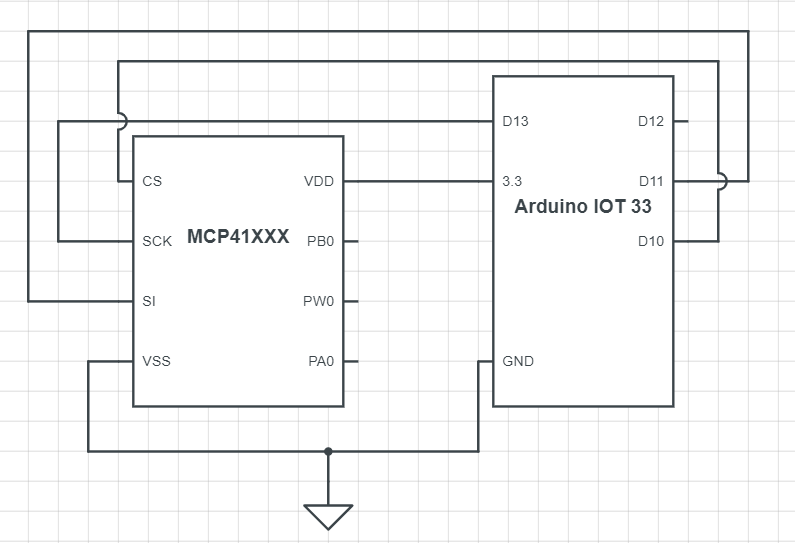
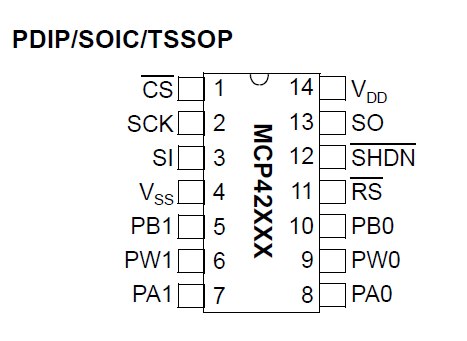


Figure 8: System overview control digital potential meters using Arduino IOT33.

The concept is using raspberry pi for reading the NTC sensors and publishing the temperature value through MQTT messages. The Arduino IOT33 will subscribe to the same topic and receive the measured value, after that it will control the NTC simulator (*MCP41/42xx*) chip to provide the same resistance value as the NTC at that moment. The system overview shows in figure 8 and figure 9 shows the connection between Arduino IOT33 with MCP41xxx/42xxx chip. The output value is connected to the NTC terminal on heat pump PCB.



Output

Figure 9: Arduino IOT33 wiring diagrams MCP41/42xx.

The *MCPxx* device should be put on *RHEOSTAT mode on* page 14 [MCP41xxx\_42xxx Datasheet](https://github.com/hancse/FF_Temperature_Calibration/tree/main/Datasheet/Digital_Potential_meter) mode of operation

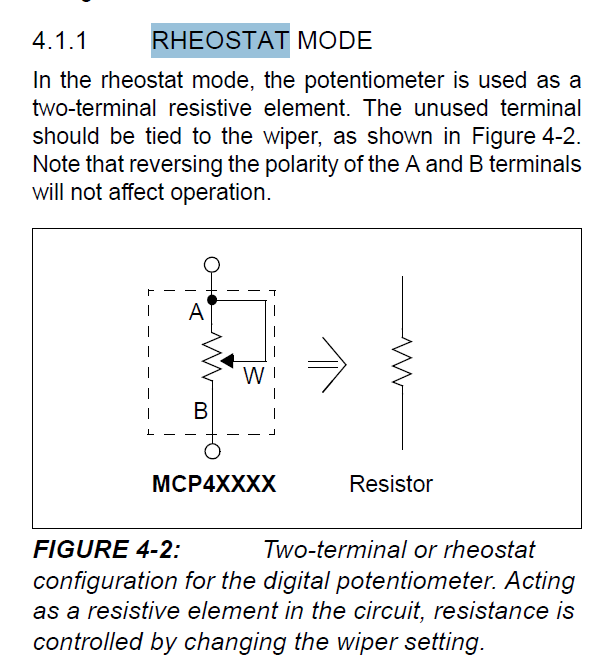


Figure 10: MCP41/42xx RHEOSTAT Mode.

* Download code from GitHub link: [IOT33 MQTT Digital Potential meters control](https://github.com/hancse/FF_Temperature_Calibration/tree/main/src/Iot33/MQTT_IOT33_MPC42050)
* Open arduino\_secrets.h tab and filled in username and WIFI password.

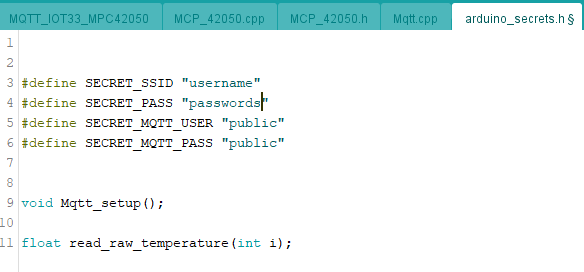


Figure 11: username and passwords.

* Flash the code to Arduino IOT33 controller.
* Open the serial monitor of Arduino studio.

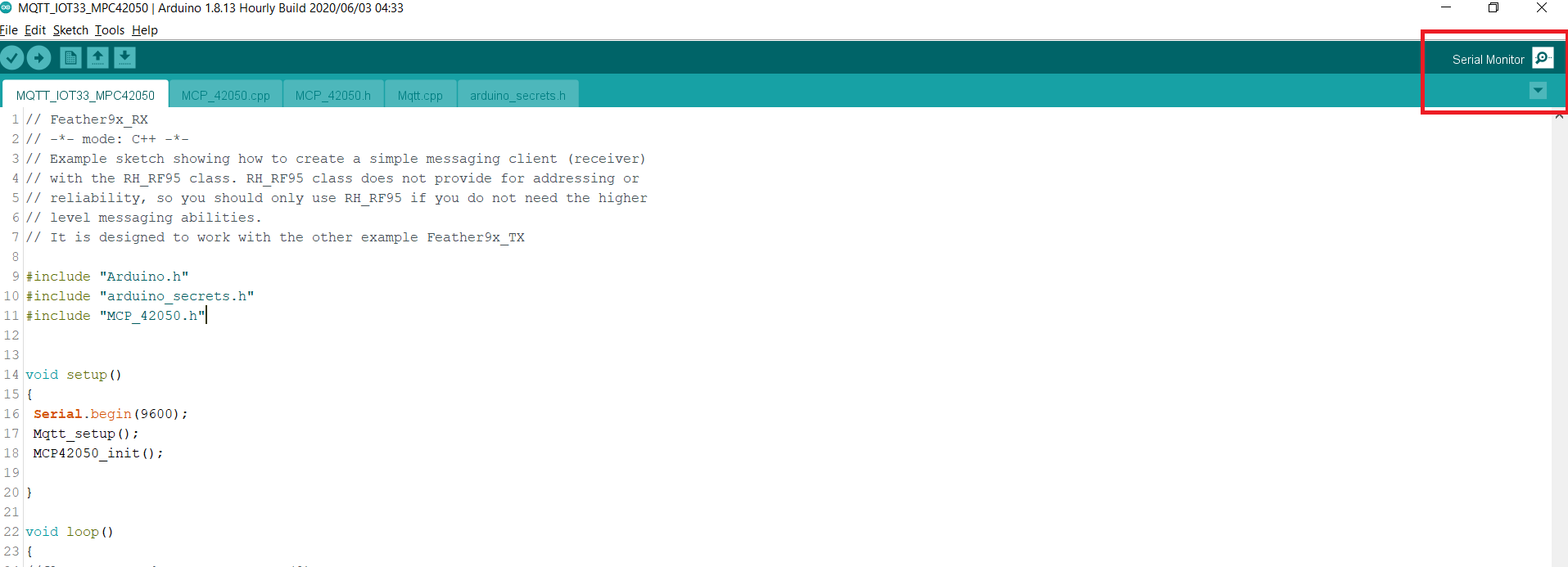


Figure 12: Open Serial Monitor.

It should print out the resistance value, the wiper position and the temperature corresponding with the resistance value as figure 13.

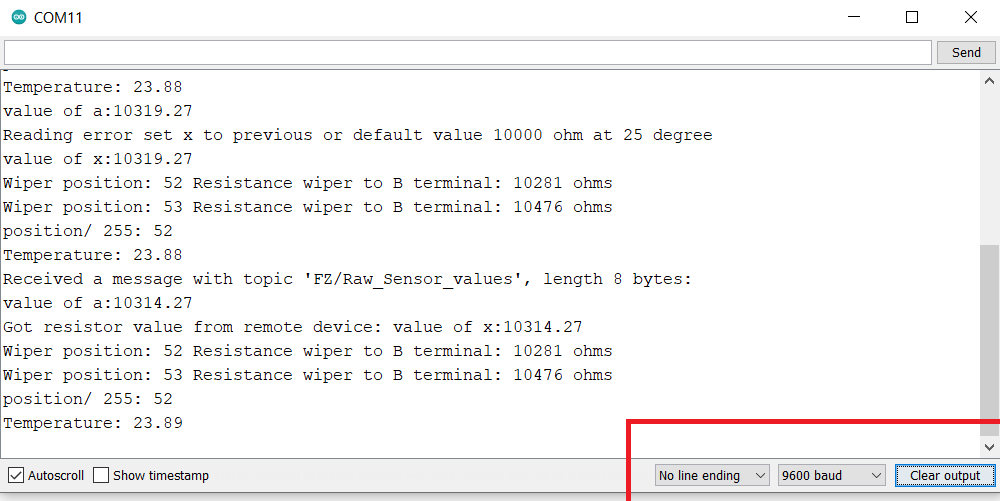


Figure 13: Serial Monitor display.

The output PBx and PAx (figure 9) can be connected to the temperature sensors pin of the Heat pump controller board.

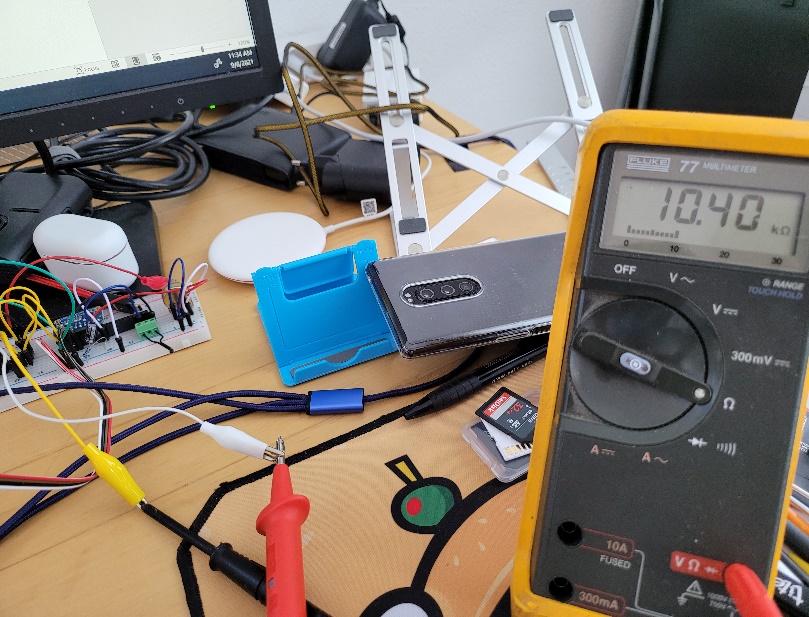


Figure 14: Checking the output value using multimeters.