**Temperature Sensors Calibration Guidelines**

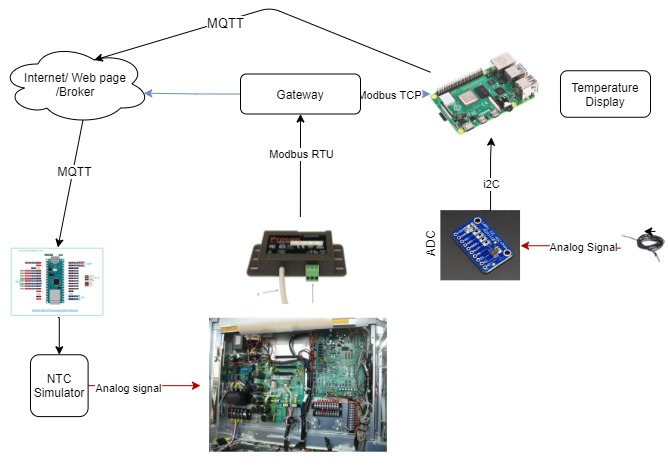


Table of Contents

[I. Introduction. 4](#_Toc77864913)

[II. Raspberry Pi MQTT solution. 5](#_Toc77864914)

[II.I Raspberry NTC Read. 5](#_Toc77864915)

[II.II Mosquito broker. 8](#_Toc77864916)

[II.III Node-Red MQTT Tutorial. 10](#_Toc77864917)

[II.IV Control digital potential meter. 14](#_Toc77864918)

[III. Lora solution. 15](#_Toc77864919)

[**Antenna Option:** 16](#_Toc77864920)

[References 17](#_Toc77864921)

[Appendix 20](#_Toc77864922)

[Appendix I: Adafruit ADS1x15 20](#_Toc77864923)

[Appendix 2: Arduino nano IOT33. 22](#_Toc77864924)

[Appendix 3: Lora Antenna Option. 24](#_Toc77864925)

[Wire Antenna 24](#_Toc77864926)

[Appendix 4: NTC Calibration. 26](#_Toc77864927)

[Figure 1: Example configuration of temperature input and display. 3](#_Toc77856180)

[Figure 2: Raspberry Pi 3b with NTC connection 4](#_Toc77856181)

[Figure 3: ADS1115 Raspberry wiring diagrams. 5](#_Toc77856182)

[Figure 4:Thermistor analog reading with voltage divider circuit to ADS1x. 5](#_Toc77856183)

[Figure 5: Install Ads1x15 library. 6](#_Toc77856184)

[Figure 6:Example python adc read with ADS1x15 break board. 7](#_Toc77856185)

[Figure 7:MQTT Broker. 7](#_Toc77856186)

[Figure 8:Install Mosquito broker on raspberry Pi. 8](#_Toc77856187)

[Figure 9: Node-red MQTT example subscribes blocks connection. 9](#_Toc77856188)

[Figure 10:Edit MQTT node panels. 10](#_Toc77856189)

[Figure 11: Node-red mqtt broker. 10](#_Toc77856190)

[Figure 12:Authentication requires. 11](#_Toc77856191)

[Figure 13:Set anonymous access to true 11](#_Toc77856192)

[Figure 14:Function block and gauge node editing. 12](#_Toc77856193)

[Figure 15:Demo dashboard. 12](#_Toc77856194)

[Figure 16:System overview control digital potential meters using Arduino IOT33. 13](#_Toc77856195)

[Figure 17:Arduino IOT33 wiring diagrams MCP41/42xx. 13](#_Toc77856196)

[Figure 18: System overview of using 2 Lora devices to control the MCP41/42xx digital potential meter. 14](#_Toc77856197)

[Figure 19:Lora feather 32u RFM95 pinout 14](#_Toc77856198)

[Figure 20: Lora MCP41/42xx wiring diagrams. 15](#_Toc77856199)

[Figure 21: Lora sever NTC wiring diagram. 15](#_Toc77856200)

[Figure 22:Python code example. 17](#_Toc77856201)

[Figure 23: Run the python example. 18](#_Toc77856202)

[Figure 24:Install Arduino nano IOT boar with Arduino studio. 19](#_Toc77856203)

[Figure 25:Install Arduino nano IOT boar with Arduino studio ctn. 19](#_Toc77856204)

[Figure 26:Install Arduino nano IOT Library. 20](#_Toc77856205)

[Figure 27:Install Arduino nano IOT Library ctn. 20](#_Toc77856206)

# I. Introduction.

Indoor temperature sensor use in Heat Pump normally is NTC (Negative temperature coefficient) or PTC (Positive Temperature coefficient). They are both resistor temperature dependent, the resistance value of the sensor changes when temperature changes. The common way to read those sensors value is using the voltage divider circuits and the voltage output can be read by an analog pin (ADC) from main controller. The analog reading methods require recalibration if the distance between sensors and the main reading board changes. In addition, there are more problems could affect the reading accuracy such as noise level with long wires, resistance values of the wire...etc. This experiment proposes the solutions of simulating the sensor behavior using digital potential meter, the communication can be done via MQTT or 433 MHz radio frequency.

Figure 1 shows the current setup of the heat pump. The NTC will be read and processed with the heat pump main controller board, after that the temperature values will be sent to the display webpage and raspberry pi via a Modbus gateway.

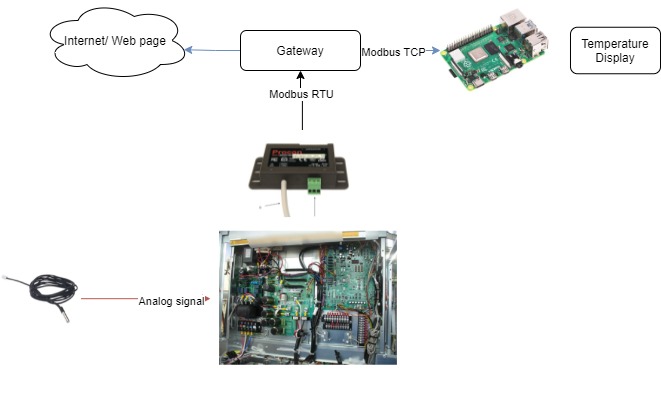


Figure 1: Example configuration of temperature input and display.

# II. Raspberry Pi MQTT solution.

## II.I Raspberry NTC Read.

The raspberry pi does not have built in analog reading; therefore, an external ADC needs to be added. The sensors used in this experiment are ADS1115 16-bit resolution. The alternative options are ADS1015 12-bit or MCP4725 12 bit.

In figure 2, there are 2 similar NTC sensors, one is connected directly to the heat pump and the other connects to raspberry through ADS1115. In figure 3 and 4 are the wiring diagrams.

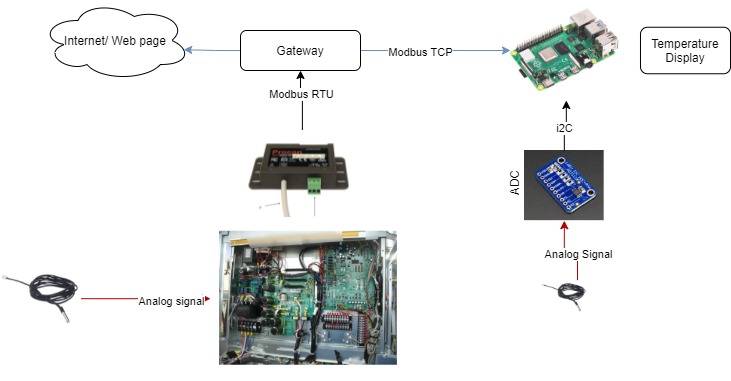


Figure 2: Raspberry Pi 3b with NTC connection

In figure 3 and 4 the wiring diagrams from the sensors to ADS1115 and to the raspberry pi are shown.

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

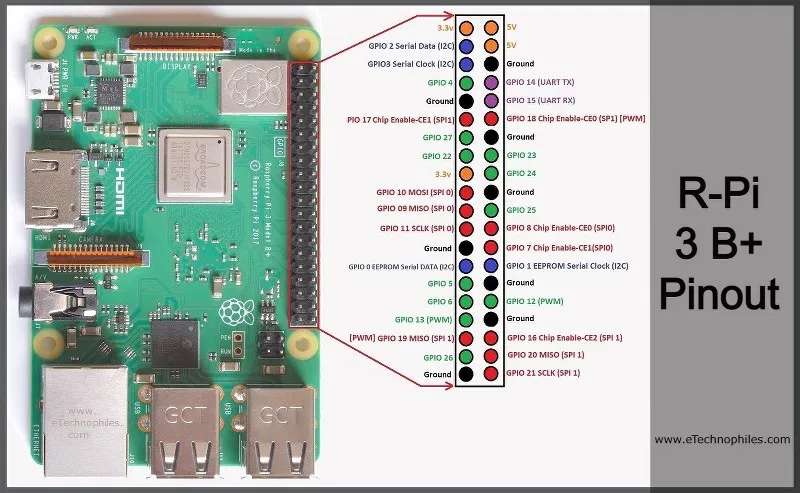
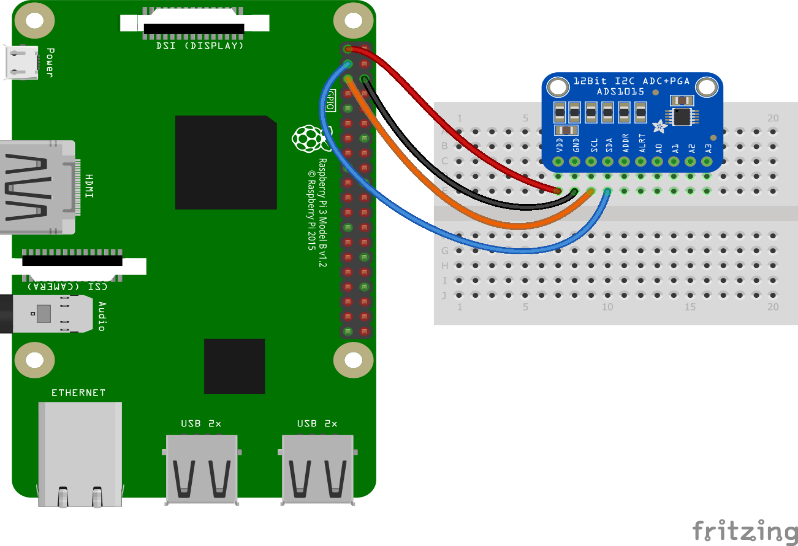


Figure 3: ADS1115 Raspberry wiring diagrams.

The thermistor NTC can connect with any pin from A0 to A5 through voltage divider circuits. The fix resistor R1 needs to be selected based on the sensors value for best performance.

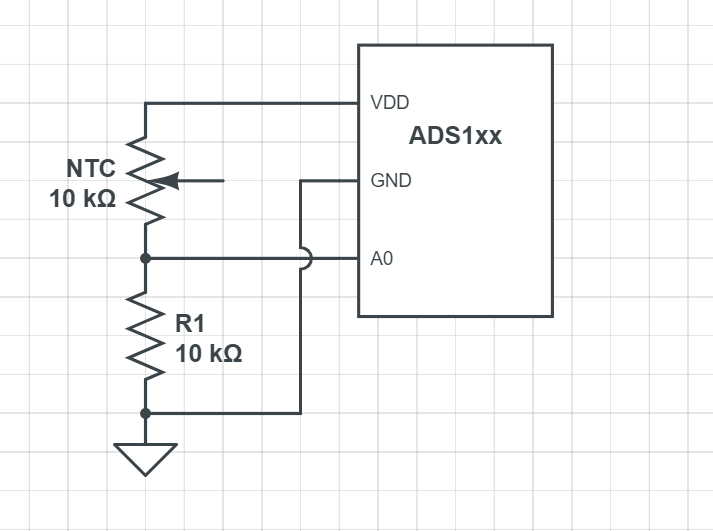


Figure 4: Thermistor analog reading with voltage divider circuit to ADS1x.

**Python installation of ADS1x15 library:**

Follow the instruction on adafruit page [ Ref] to install the [Adafruit\_Blinka](https://github.com/adafruit/Adafruit_Blinka) library. This may also require enabling I2C on raspberry platform and verifying it is running python 3. Open the terminal and run the following command:

* sudo pip3 install adafruit-circuitpython-ads1x15

Remember that if the default Python is version 3 it may need to run 'pip' instead. Just make sure that you do not try to use CircuitPython on Python 2.x, it is not supported.

The instruction and example code from the official GitHub page are shown in figures 5 and 6.

* [Adafruit ADS1x15](https://github.com/adafruit/Adafruit_CircuitPython_ADS1x15)

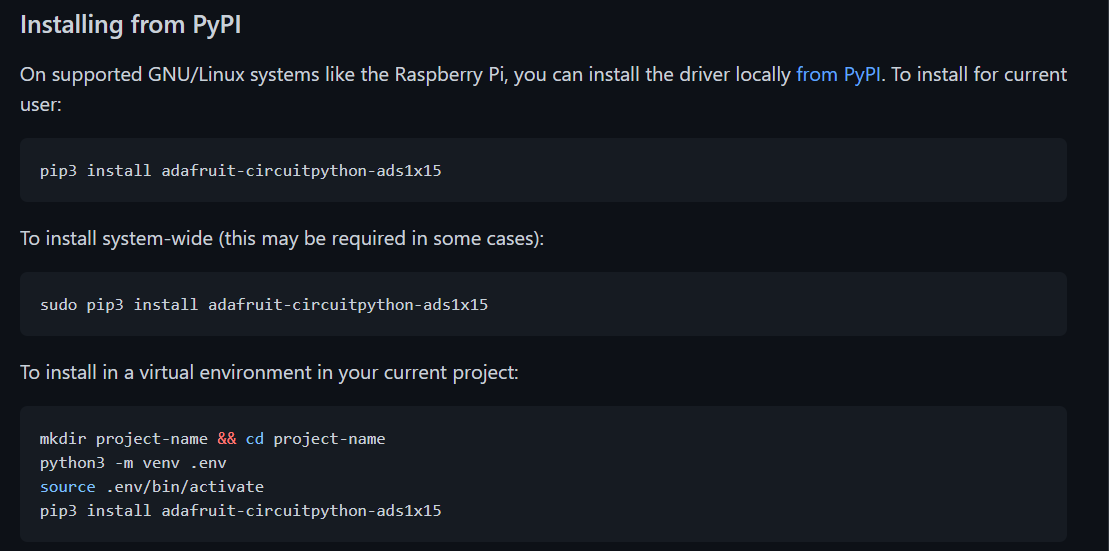


Figure 5: Install Ads1x15 library.



Figure 6: Example python adc read with ADS1x15 break board.

## II.II Mosquito broker.

The broker is primarily responsible for **receiving** all messages, **filtering** the messages, **deciding** who is interested in it and then **publishing** the message to all subscribed clients.

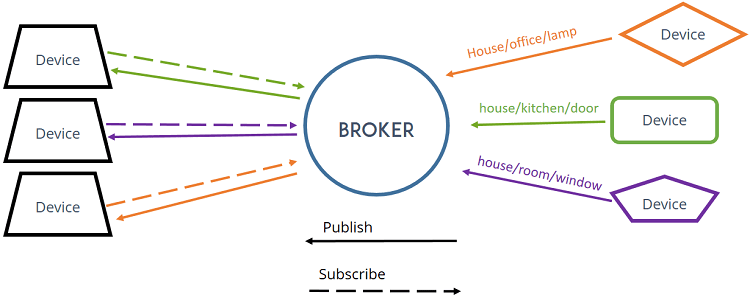


Figure 7: MQTT Broker.

**Install Mosquito on Raspbian OS:**

Figure 8 shows how to install mosquito broker on raspberry Pi.

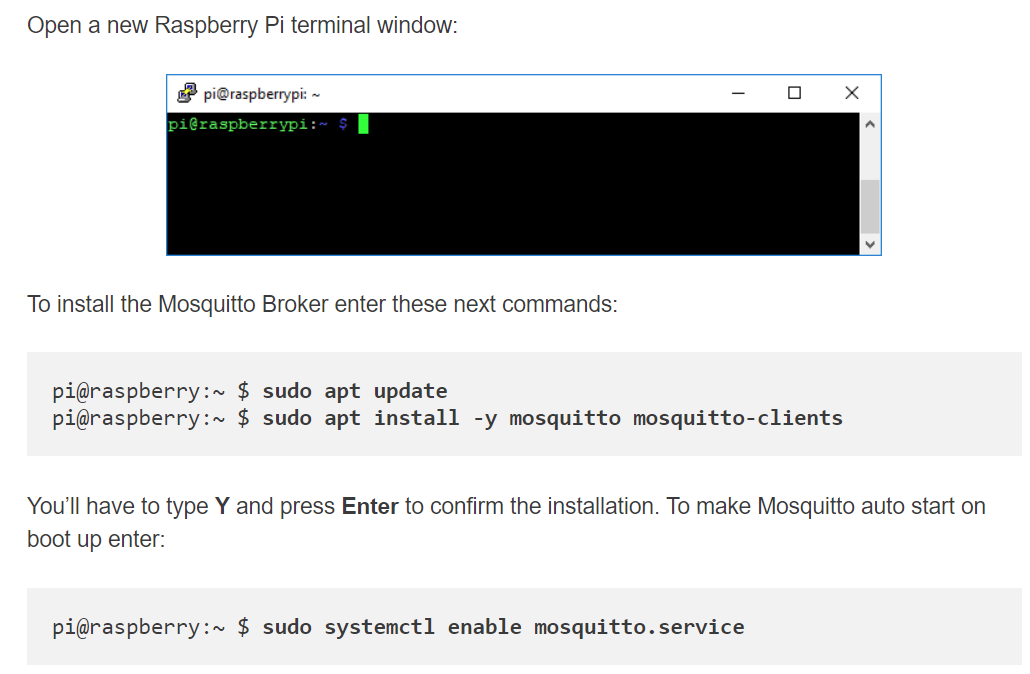


Figure 8: Install Mosquito broker on raspberry Pi.

More examples and tutorials are provided on the link below:

* [How to Install Mosquitto Broker on Raspberry Pi](https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/).
* [How to Install Mosquitto on The Raspberry Pi](https://stevessmarthomeguide.com/install-mosquitto-raspberry-pi/).
* [Testing Mosquitto Broker and Client on Raspberry Pi](https://randomnerdtutorials.com/testing-mosquitto-broker-and-client-on-raspbbery-pi/).

**Install Python MQTT:**

The next step is installing the python mqtt library using pip command on the terminal window.

* **pip3** install **paho-mqtt**

Look at the link [Python MQTT example](http://www.steves-internet-guide.com/into-mqtt-python-client/) for more example.

## II.III Node-Red MQTT Tutorial.

Install node-red on the raspberry pi follow an instruction on the official page:

* [Running on Raspberry Pi](https://nodered.org/docs/getting-started/raspberrypi)

After installing node-red open your web browser. Enter your raspberry IP address with port 1880 into the browser (use ifconfig command on the terminal to find out the IP address of your Pi), for example:

* <http://Your-Pi_IPaddress:1880/>

If everything has been setup correctly the interface in figure 9 should pop up. Figure 9 also shows the basic MQTT received messages and display with node-red.

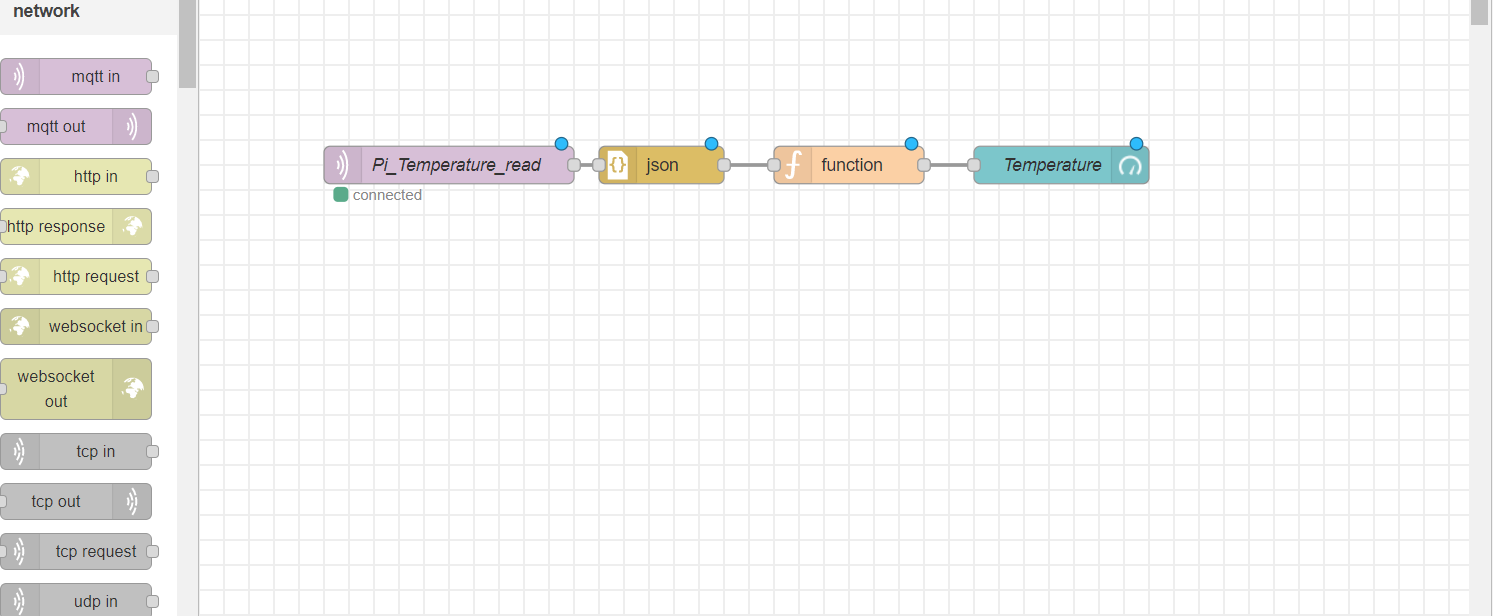
****

Figure 9: Node-red MQTT example subscribes blocks connection.

Double click to the mqtt in (Pi\_Temeprature\_read/ the 1st block) the figure 10 below will show up. Fill in the subscribe topic and names of the node then select edit button (red square).

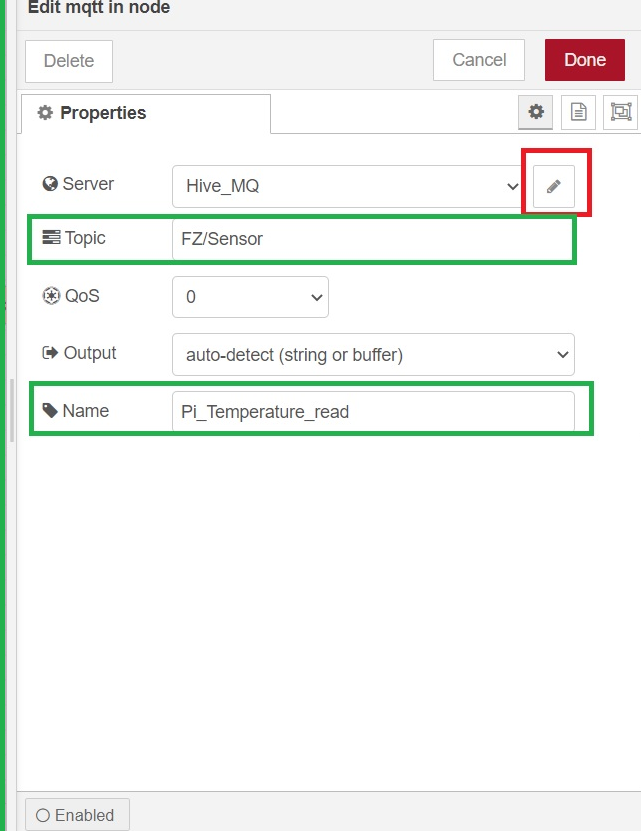
****

Figure 10: Edit MQTT node panels.

The following screen in figure 11 will show up. Users need to fill in the name of the broker and the server address here. In this example the Hive\_MQ public broker has been used.

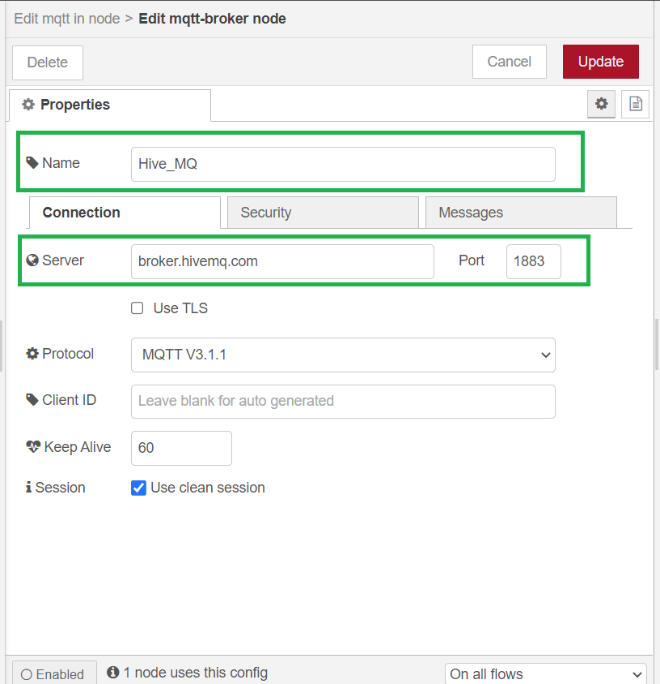


Figure 11: Node-red mqtt broker.

Using mosquito broker locally on the raspberry, the user needs to fill in the raspberry local ip address (127.0.0.1 or Raspberry address in the local network).

Note that mosquito 2.0 2.0 by default only allows the use of local IP address 127.0.0.1 without username and password configuration. An authentication shows in figure 12 is needed for different IP configuration.

Graphical user interface, text, application, email

Description automatically generated

Figure 12: Authentication requires.

An example on how to set anonymous to true is shown on figure 13.

* Open the terminal window on the raspberry.
* Move to mosquito folder: cd /etc./mosquito/
* Use: sudo nano mosquitto.conf to open the config file.
* Add listener on port 1883 and set allow anonymous to true. Then press control x to save and exit.
* Copy the config file to conf.d folder: sudo cp mosquitto.conf /etc/mosquito/conf.d/

Graphical user interface, text, application

Description automatically generated

Figure 13: Set anonymous access to true

Read more on mosquito 2.0 and how to set username and password on the link below:

* [Mosquitto migrating from 1.x to 2.0](https://mosquitto.org/documentation/migrating-to-2-0/#listener-behaviour-changes)
* [Mosquitto Username and Password Authentication -Configuration and Testing](http://www.steves-internet-guide.com/mqtt-username-password-example/)

Figure 14 shows the function block on the left which handles the incoming messages and transfers into the readable value for the display node on the right.

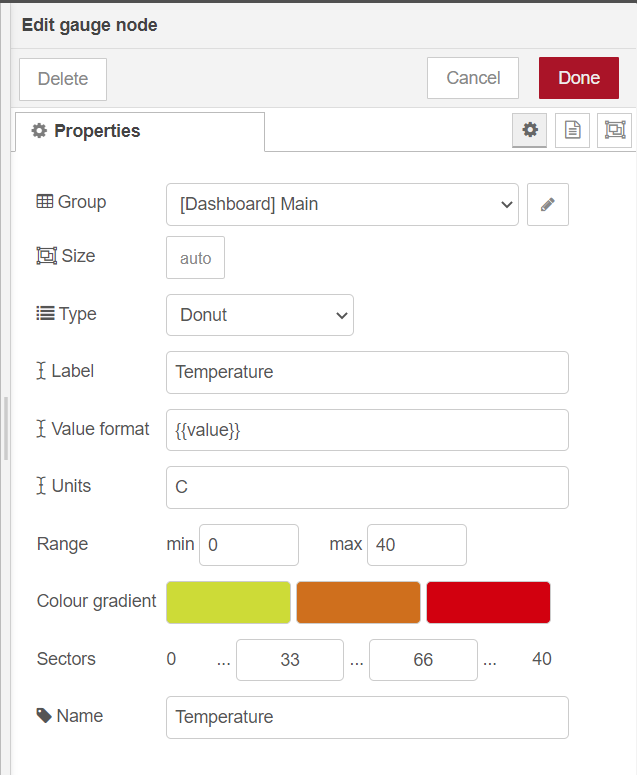
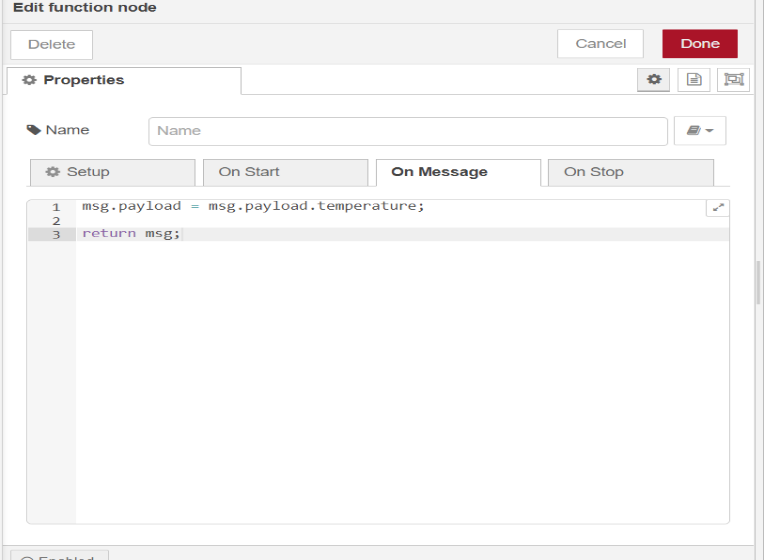


Figure 14: Function block and gauge node editing.

.

Finally, the web interface can be deployed using the deploy button on the top-right corner of the web page. The web interface can be accessed using the web browser via the IP address:

* <http://Your-Pi_IPaddress:1880/>ui

A dashboard Temperature display demo is shown in figure 15.

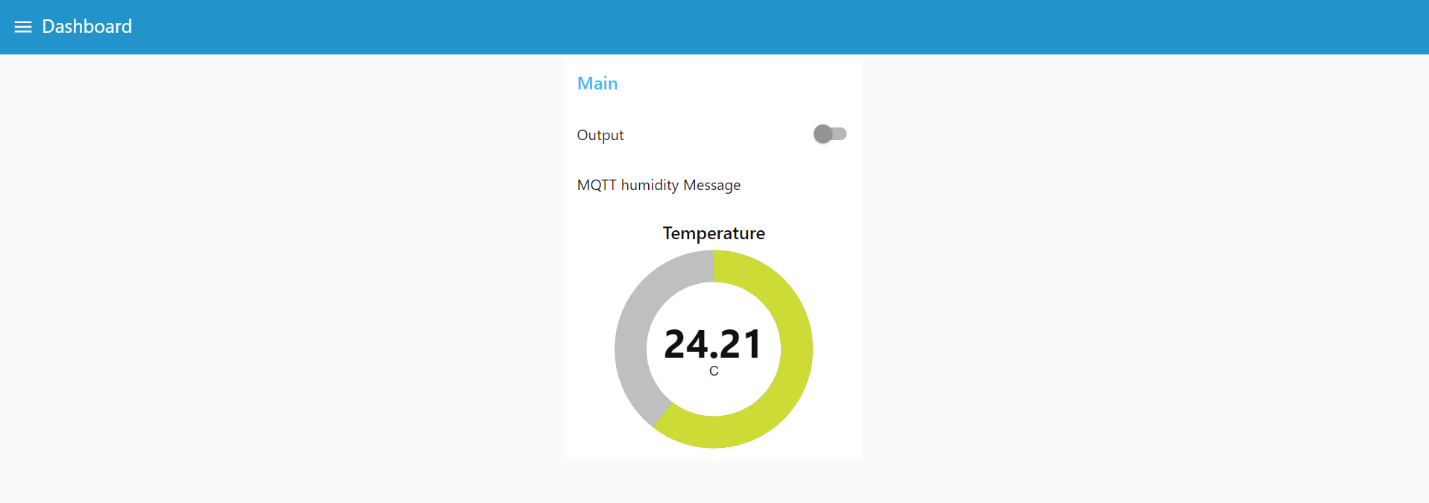


Figure 15: Demo dashboard.

Read more examples on node-red and mqqt:

* [MQTT router with Node-RED](https://diyprojects.io/mqtt-router-node-red-connect-sonoff-tasmota-modules-cloudmqtt/#.YPAKdegzZPY)
* [MOSQUITTO AND NODE RED ON RASPBERRY PI](https://oneguyoneblog.com/2017/06/20/mosquitto-mqtt-node-red-raspberry-pi/)
* [Configuring the Node-Red MQTT Publish and Subscribe Nodes](http://stevesnoderedguide.com/configuring-the-mqtt-publish-node)

## II.IV Control digital potential meter.

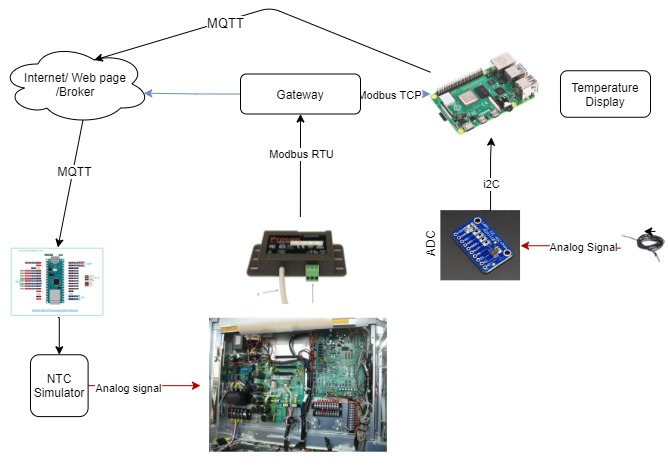


Figure 16: System overview control digital potential meters using Arduino IOT33.

.

The concept is using the 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 16 and figure 17 shows the connection between Arduino IOT33 with MCP41xxx/42xxx chip. The output value is connected to the NTC terminal on heat pump PCB.

Diagram, schematic

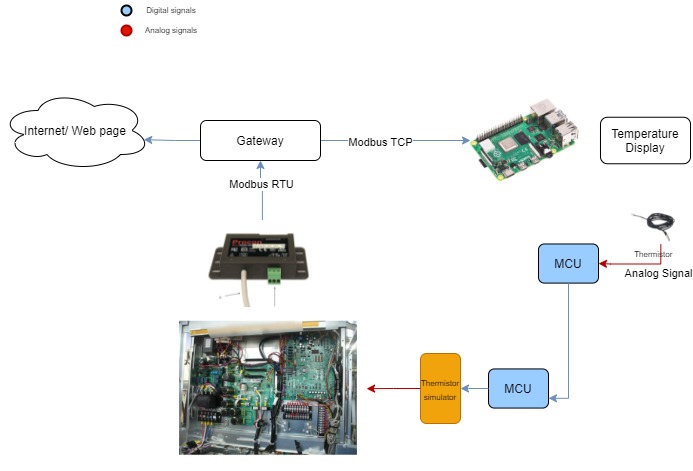
Description automatically generated

Output

Figure 17: Arduino IOT33 wiring diagrams MCP41/42xx.

## III. Lora solution.

An alternative solution for long-range communication (up to 40 km with dedicated antenna) is using the Lora devices which can transfer signal over radio frequency. The concept is like the previous chapter, but instead of MQTT messages the master Lora devices will transfer the signal to the other Lora via radio frequency (434 MHz is the frequency use in EU).



434 MHz

Figure 18: System overview of using 2 Lora devices to control the MCP41/42xx digital potential meter.

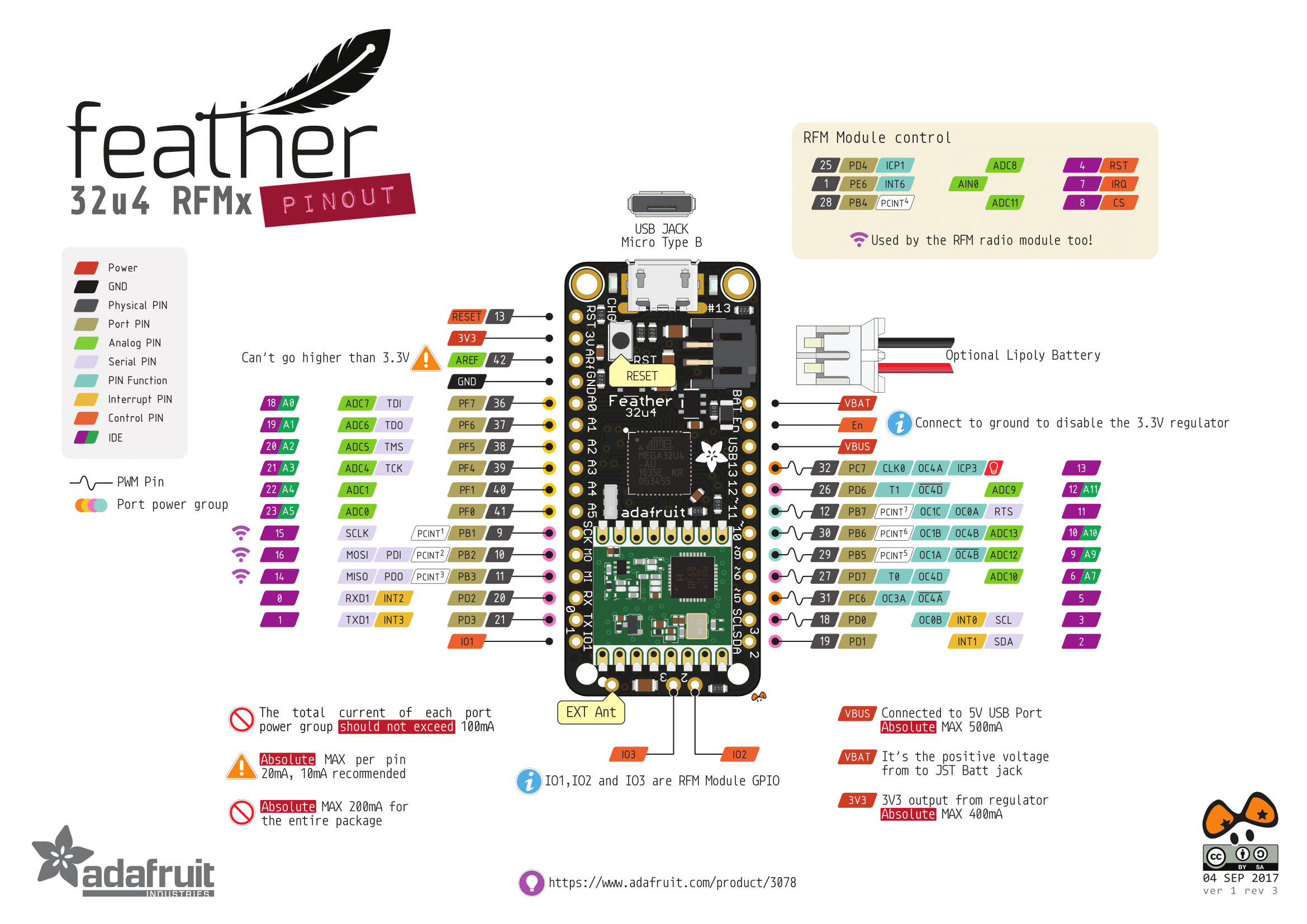
Figure 20,21 are the wiring diagrams of the Lora devices with NTC thermistor and MCP41/42xx. The Lora used in this experiment is feather 32u RFM95 with the pinout shows in figure 19.

Figure 19: Lora feather 32u RFM95 pinout

Diagram, schematic

Description automatically generated

Output

Figure 20: Lora MCP41/42xx wiring diagrams.

Diagram, schematic

Description automatically generated

Figure 21: Lora sever NTC wiring diagram.

## **Antenna Option:**

The Feather Radio does not have a built-in antenna. Instead, there are two options for attaching an antenna. For most low-cost radio nodes, a wire works great. If the Feather needs to be in an enclosure, soldering in uFL and using a uFL to SMA adapter will let the user attach an external antenna. Look at Appendix III for more detail on antenna.

# References

1. [Basic library that reads a NTC thermistor from an analog pin](https://github.com/panStamp/thermistor), (dberenguer, n.d.).
2. [Arduino Thermistor](https://www.arduino.cc/reference/en/libraries/thermistor/) library, (panStamp, n.d.).
3. [How to connect Raspberry to a thermistor](https://www.circuito.io/app?components=9443,149486,200000).
4. [Calculate Temperature with Arduino Nano or Arduino Uno and NTC Thermistor Circuit.](https://www.northstarsensors.com/blog/temperature-from-thermistor-arduino-circuit)
5. [Controlling a Digital Potentiometer Using SPI.](https://www.arduino.cc/en/Tutorial/LibraryExamples/DigitalPotControl)
6. [Adafruit's Legacy Raspberry Pi Python Code Library.](https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code)
7. [Digital Potentiometer MCP42100 With Arduino.](https://www.instructables.com/Digital-potentiomter-MCP42100-with-Arduino/)
8. [Digital Potentiometer MCP41100 and Arduino.](https://www.instructables.com/Digital-Potentiometer-MCP41100-and-Arduino/)
9. [Adafruit Feather 32u4 with LoRa Radio Module.](https://learn.adafruit.com/adafruit-feather-32u4-radio-with-lora-radio-module/using-with-arduino-ide)
10. [Make an Arduino temperature sensor (thermistor tutorial).](https://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/)
11. [Make an Arduino Temperature Sensor (thermistor tutorial).](https://create.arduino.cc/projecthub/iasonas-christoulakis/make-an-arduino-temperature-sensor-thermistor-tutorial-b26ed3)
12. [Adafruit RadioHead library.](https://github.com/adafruit/RadioHead)
13. [Dragino RadioHead Library.](https://github.com/dragino/RadioHead)
14. [Writing a Library for Arduino.](https://www.arduino.cc/en/Hacking/libraryTutorial)
15. [How to make an Arduino Ohm meters.](https://www.circuitbasics.com/arduino-ohm-meter/)
16. [LoRa Mesh Networking with Simple Arduino-Based Modules.](https://nootropicdesign.com/projectlab/2018/10/20/lora-mesh-networking/comment-page-1/)
17. [LoRa Questions.](https://wiki.dragino.com/index.php?title=LoRa_Questions)
18. [LoRa airtime calculator](https://docs.google.com/spreadsheets/d/1voGAtQAjC1qBmaVuP1ApNKs1ekgUjavHuVQIXyYSvNc/edit#gid=0)
19. [Using a Thermistor.](https://learn.adafruit.com/thermistor/using-a-thermistor)
20. [MQTT examples](https://github.com/tigoe/mqtt-examples).
21. [Arduino-Nano-33-IoT-Ultimate-Guide.](https://github.com/ostaquet/Arduino-Nano-33-IoT-Ultimate-Guide)
22. [Arduino BLE Example Explained Step by Step.](https://rootsaid.com/arduino-ble-example/)
23. [MCP42100 Demo code.](https://github.com/GadgetReboot/MCP42100_Demo/blob/master/MCP42100_R_Test.ino)
24. [Arduino mcp4xxx.](https://github.com/jmalloc/arduino-mcp4xxx)
25. [MCP42050 example.](https://github.com/zendkash/Bridge-drive/blob/master/Arduino/MCP42050/MCP42050.ino)
26. [Arduino BLE.](https://github.com/arduino-libraries/ArduinoBLE)

# Appendix

## Appendix I: Adafruit ADS1x15

Adafruit provided a detailed guideline for the adc break out board [Adafruit 4-Channel ADC Breakouts](https://learn.adafruit.com/adafruit-4-channel-adc-breakouts/python-circuitpython) on its web page.

After installing the python library, on the raspberry the example code located in the location:

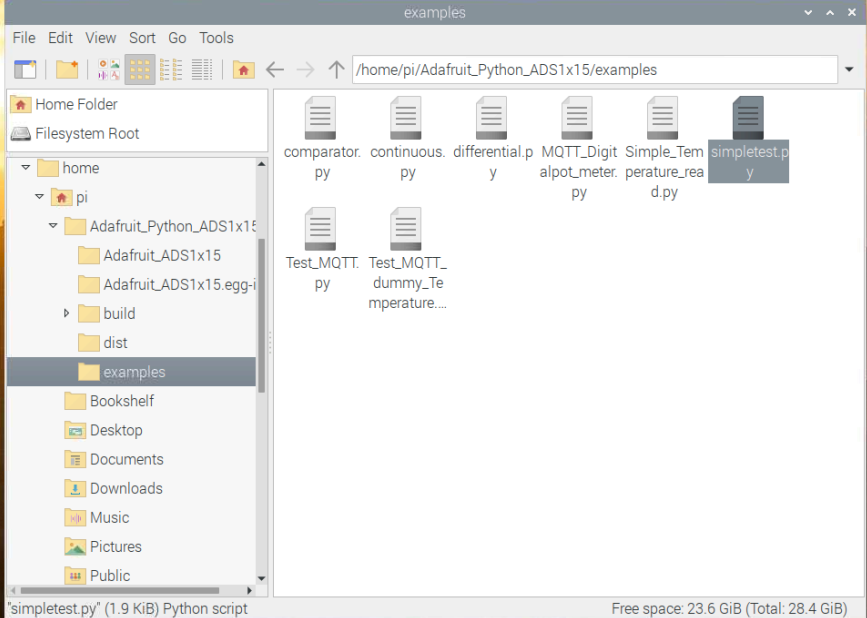
/home/pi/Adafruit\_Python\_ADS1x15/examples

Figure 22: Python code example.

The example can be run directly from terminal or double click to the python file to open Thonny, and press run like figure 23.

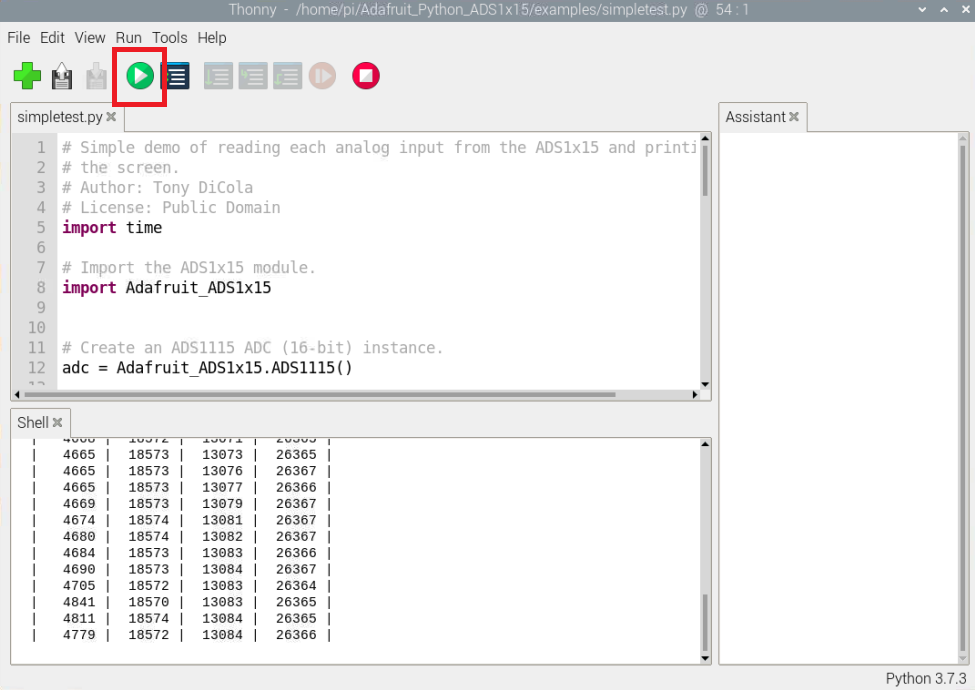


Figure 23: Run the python example.

More examples of 12-bit adc converter can be found in [MCP4725 12-Bit DAC Tutorial](https://learn.adafruit.com/mcp4725-12-bit-dac-tutorial/python-circuitpython).

**Error and solution:**

If the simpletest.py gives an error, follow the instruction below could solve the problem.

* Use: [pip3 install Adafruit-ADS1x15](https://stackoverflow.com/questions/60360497/problem-reading-from-ads1115-from-raspberry-pi-using-i2c)

Error: [TypeError: one character string expected](https://www.programmersought.com/article/20994707614/)

## Appendix 2: Arduino nano IOT33.

**Arduino nano IOT33:**

Follow then [Getting start](https://ladvien.com/arduino-nano-33-bluetooth-low-energy-setup/) guide to install the support board and library for Arduino nano IOT33 using Arduino studio.

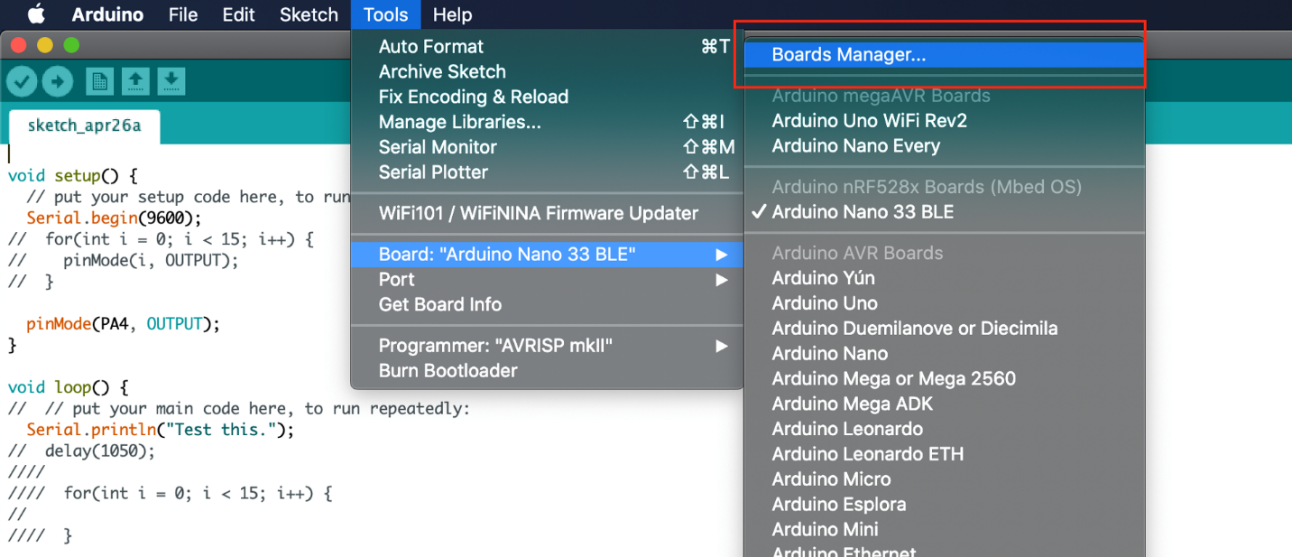


Figure 24: Install Arduino nano IOT boar with Arduino studio.

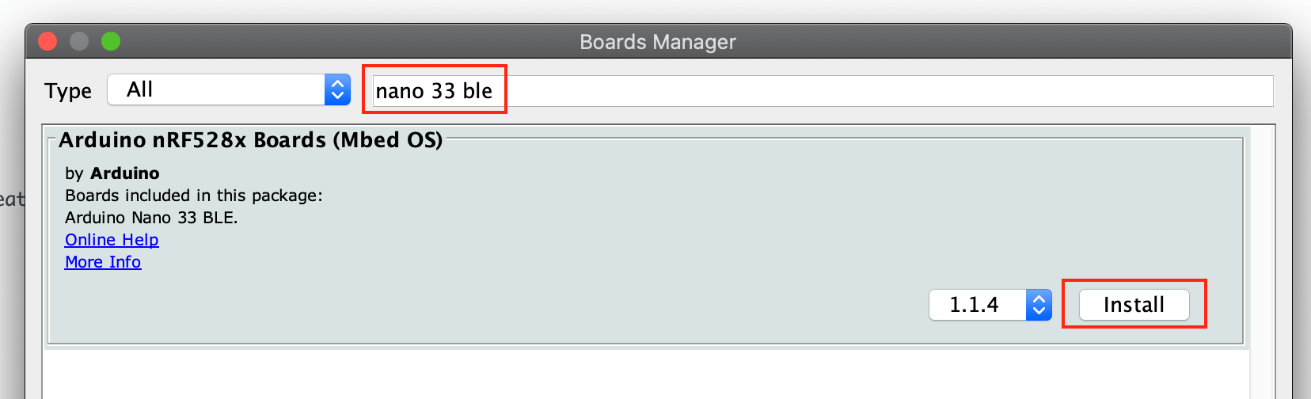


Figure 25: Install Arduino nano IOT boar with Arduino studio ctn.

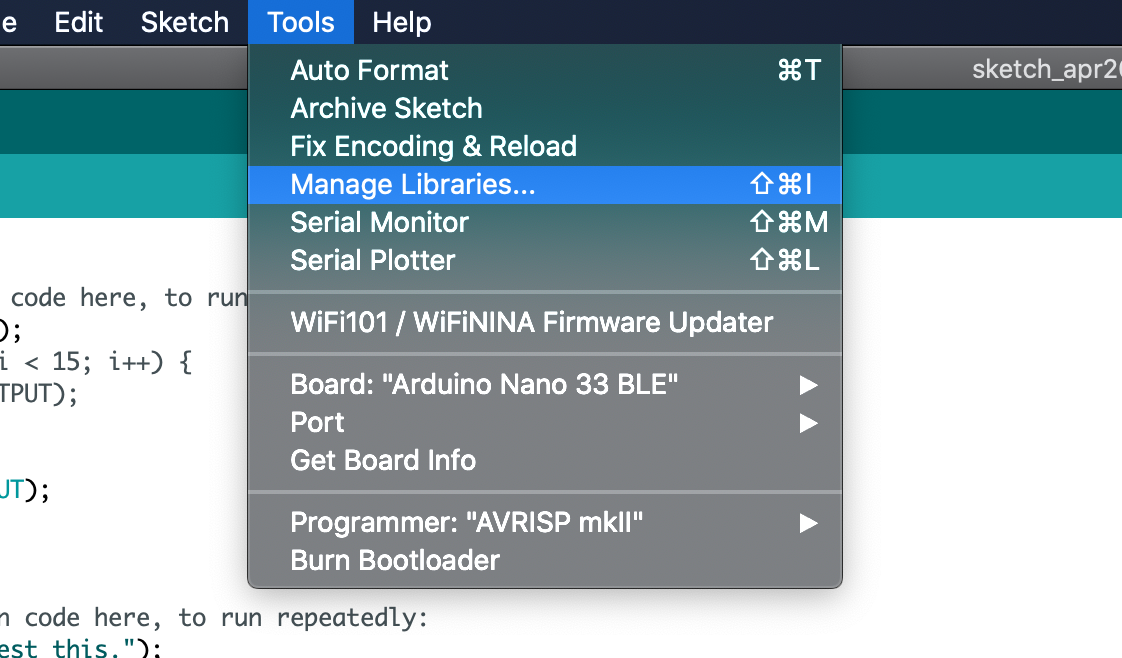


Figure 26: Install Arduino nano IOT Library.

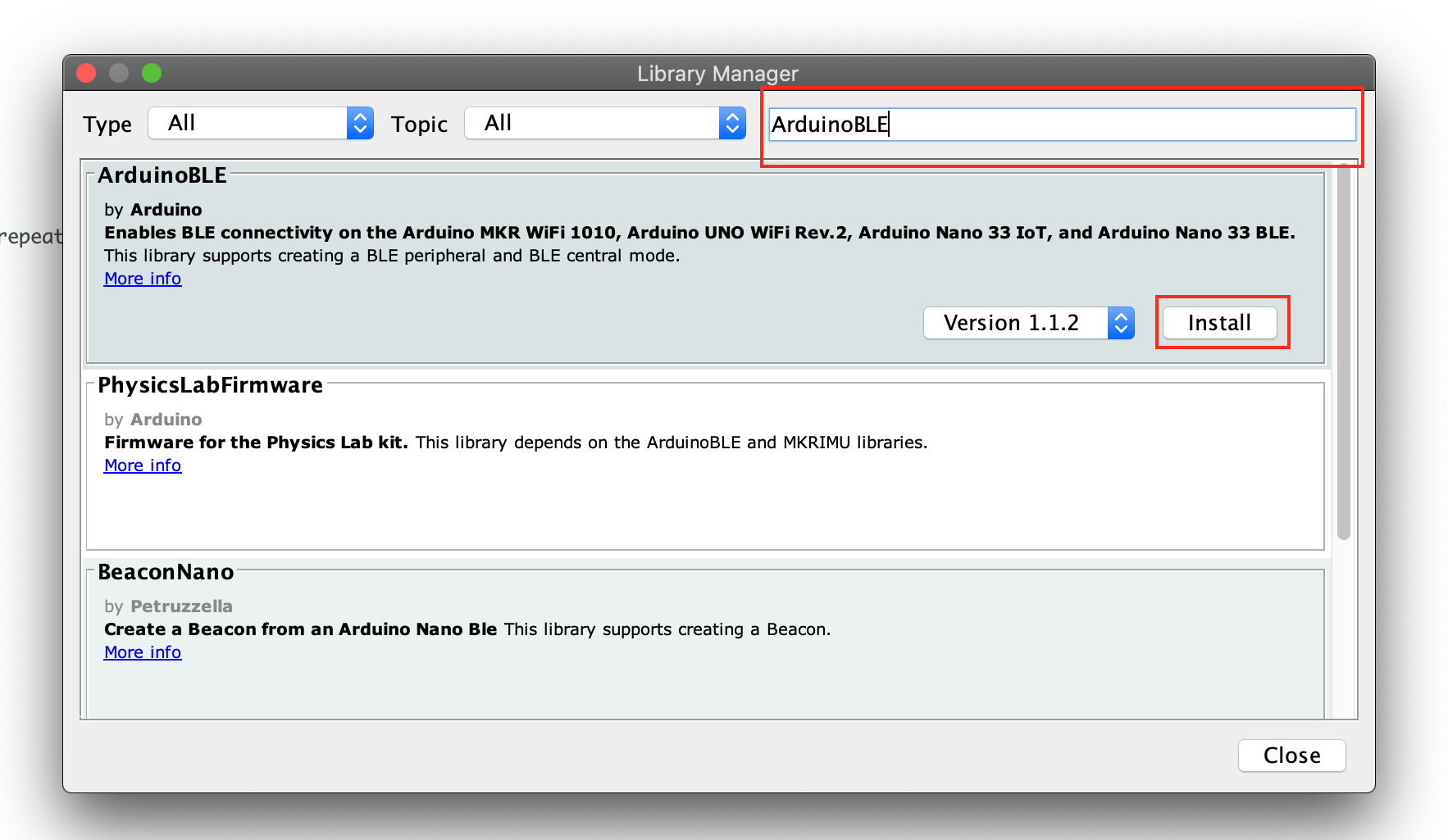


Figure 27: Install Arduino nano IOT Library ctn.

## Appendix 3: Lora Antenna Option.

The Feather Radio does not have a built-in antenna. Instead, there are two options for attaching an antenna. For most low-cost radio nodes, a wire works great. If the Feather needs to be in an enclosure, soldering in uFL and using a uFL to SMA adapter will let the user attach an external antenna

# Wire Antenna

A wire antenna, aka "quarter wave whip antenna" is low cost and works very well. The user just must cut the wire down to the right length.

|  |  |
| --- | --- |
| * feather_antennaready.jpg | Cut a stranded or solid core wire to the proper length for the module/frequency   * **433 MHz** - 6.5 inches, or 16.5 cm * **868 MHz** - 3.25 inches or 8.2 cm * **915 MHz** - 3 inches or 7.8 cm |
| * feather_antennasolder.jpg | Strip a mm or two off the end of the wire, tin, and solder into the **ANT** pad on the very right-hand edge of the Feather |
| * feather_antennadone.jpg | That is, it, you are done! |

**Read more on the Lora board:**

* [Adafruit Feather 32u4 with LoRa Radio Module](https://learn.adafruit.com/adafruit-feather-32u4-radio-with-lora-radio-module/overview).
* [Adafruit RFM69HCW and RFM9X LoRa Packet Radio Breakouts](https://learn.adafruit.com/adafruit-rfm69hcw-and-rfm96-rfm95-rfm98-lora-packet-padio-breakouts/using-the-rfm69-radio).
* [Adafruit Feather 32u4 with LoRa Radio Module](https://learn.adafruit.com/adafruit-feather-32u4-radio-with-lora-radio-module/power-management).

## Appendix 4: NTC Calibration.

NTC: Negative Temperature Coefficient

PTC: Positive Temperatures Coefficient.

Thermistor circuits:

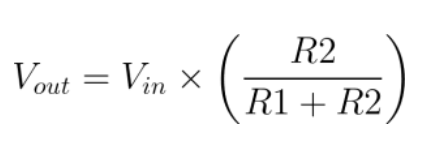
Diagram, schematic

Description automatically generated

Figure 28: Thermistor analog reading example

Since the thermistor is a variable resistor, the resistance needs to be measured before calculating the temperature. However, the general microcontroller (mcu) cannot measure resistance directly, it can only measure voltage.

The mcu will measure the voltage at a point between the thermistor and a known resistor. This is known as a voltage divider. The equation for a voltage divider is:

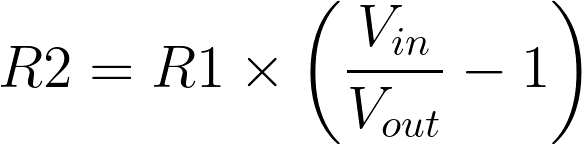


Equation 1: Voltage divider equation.

In terms of the voltage divider in a thermistor circuit, the variables in the equation are:

* Vout: Voltage between thermistor and known resistor.
* Vin: Vcc, ie: 3.3 or 5 V
* R2: Resistance of thermistor.
* R1: Known resistor value.

This equation can be rearranged and simplified to solve for R2, the resistance of the thermistor.



Equation 2: Thermistor resistance value calculation.

Finally, the [Steinhart-Hart equation](https://en.wikipedia.org/wiki/Steinhart%E2%80%93Hart_equation) is used to convert the resistance of the thermistor to a temperature reading. The Steinhart–Hart equation is a model of the [resistance](https://en.wikipedia.org/wiki/Electrical_resistance) of a [semiconductor](https://en.wikipedia.org/wiki/Semiconductor) at different [temperatures](https://en.wikipedia.org/wiki/Temperature). The equation is:



Equation 3: The Steinhart–Hart equation.

* T: is the temperature in K.
* R: is the resistance at T (in ohms).
* A, B, C are the ***Steinhart–Hart coefficients***, which vary depending on the type and model of [thermistor](https://en.wikipedia.org/wiki/Thermistor) and the temperature range of interest.

The equation is often used to derive a precise temperature of a thermistor, since it provides a closer approximation to actual temperature than simpler equations and is useful over the entire working temperature range of the sensor. Steinhart–Hart coefficients are usually published by thermistor manufacturers.

Read more on the Steinhart–Hart equation and thermistor tutorial on the link below:

* [Thermistor tutorial](https://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/).
* [Steinhart Hart equation](https://en.wikipedia.org/wiki/Steinhart%E2%80%93Hart_equation).
* [Arduino Nano Temperature Sensor Control.](https://ozeki.hu/p_3067-how-to-setup-a-temperature-sensor-on-arduino-nano.html)
* [Arduino Thermistor Theory, Calibration, and Experiment.](https://makersportal.com/blog/2019/1/15/arduino-thermistor-theory-calibration-and-experiment)

## Appendix 5: Error and discuss link.

* [Too much noise when connecting thermistors using long wires](https://forum.arduino.cc/t/too-much-noise-when-connecting-thermistors-using-long-wires/93018/13).
* [Converting a float to uint8\_t for LoRa radio 433MHz.](https://forum.arduino.cc/t/converting-a-float-to-uint8_t-for-lora-radio-433mhz/599173)
* [Difference between standard and extended Steinhart-Hart equation](https://electronics.stackexchange.com/questions/463747/difference-between-standard-and-extended-steinhart-hart-equation)
* [RFM95W on custom PCB only works over short distances.](https://lowpowerlab.com/forum/rf-range-antennas-rfm69-library/rfm95w-only-works-over-short-distances/)
* [Arduino ble sense example.](https://github.com/Ladvien/arduino_ble_sense/tree/master/ble_sense_33_test.ino)
* Double clicking the RST button on microcontroller when the serial connection port disappeared.
* Read more on mqtt json messages handling and java script.