University of Puerto Rico -Mayaguez

Electrical and Computer Engineering Department

INEL 4206-030 Microprocessors I

Final Project

Is the AC on?

Isaac A. Rivera

Yadiel Camacho

Prof. Luis B. Roa

December 12, 2022

Index

Introduction

For the implementation of the ESP32, it was necessary to read temperature and make it work through a cloud in which will communicate each other and implement “Hey Siri function”. The ESP32 will connect to a broker, which will be constantly receiving data from the temperature sensor. This will enable the broker to send that information to a dashboard and that way demonstrate graphically the temperature. By this, it will be known if the air conditioner is ON or OFF.

This document will be divided in some points as

-Goals

-Software and tools required

-Code to read temperature

-Process of creating server and controlling it from local computer.

-Installing node-red in server, creating the nodes and dashboard.

-Connecting mqtt to server

-Adding “Hey Siri” function

-Conclusion and revision of goals

Goals

- Establish a workplan to keep track of progress.
- Create a cloud server that can be opened from local computer.
- Create a successful communication between ESP32 and MQTT broker located in node-red.
- Configure Siri to be able to answer if the air conditioner is ON or OFF.
- Develop a Code in which it can connect to MQTT to subscribe and publish data to node-red.

Software and tools required

- ✓ Visual studio code (c/c++)
- ✓ Platformio (extension of vs code)
- ✓ ESP32 starter kit
- ✓ Thermistor
- ✓ 10k resistance
- ✓ Jumpers
- ✓ Protoboard
- ✓ AWS
- ✓ Puttygen
- ✓ Whois.com

- ✓ Windows Power Shell
- ✓ Node-red
- ✓ Studio 3T
- ✓ Siri
- ✓ Mongodb

Code to read temperature

Basic code to perform task required that was to read temperature. This will receive the value in voltage ADC. With variables previously defined as tempK, tempC and tempF will convert that value of voltage to a value of Kelvin temperature. Then with 2 more equation will translate the value to Celsius and Fahrenheit.

```
void loop()
{
  float adcValue = analogRead(PIN_ANALOG_IN_TEMPERATURE);
  float voltage = (float)adcValue / 4095.0 * 3.3;                // read ADC pin and convert to voltage
  float Rt = 10 * voltage / (3.3 - voltage);                     // calculate resistance value of thermistor
  double tempK = 1 / (1 / (273.15 + 25) + log(Rt / 10) / 3950.0); // calculate temperature (Kelvin)
  double tempC = tempK - 273.15;                                       // calculate temperature (Celsius)
  double tempF = tempC * 9 / 5 + 32;                                  //calculate temperature (Farenheit)
```

Figure 1. Code that read value of voltage and translate it to temperature using equations.

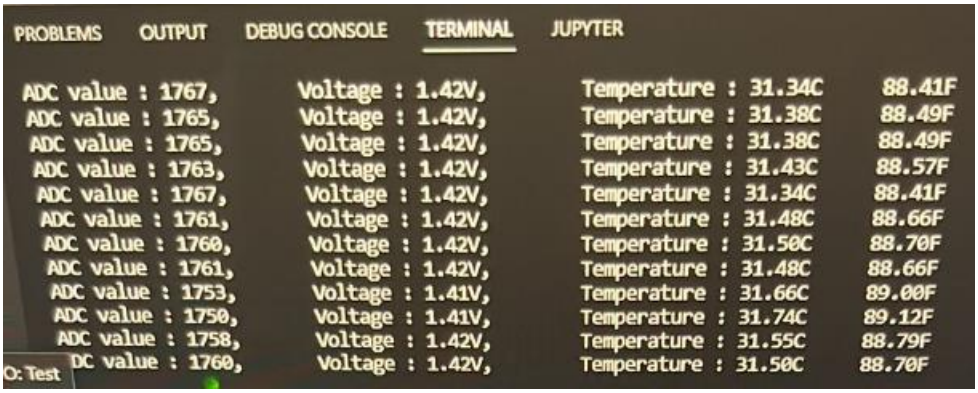

Figure 2. Code to read temperature running.

Process of creating server and controlling it from local computer

- Created an account in AWS, went to lightsail and created an instance with ubuntu and added a static IP address.
- Opened remote server and check ssh key that server created.
- With puttygen, generated a random public key that was be stored in both cloud server and local computer.
- Generated private key that will remain in local computer.
- Once both private keys were saved, the next step was to give a name to the IP address of the cloud server.
- Searched for Whois.com in google and buy a domain, it does not have restrictions.
- In the domain, add a direction, which was the static IP address created for de cloud server.
- Once the process was completed, opened windows power shell, and enter from the ssh key using the name of the server with the domain name as direction.

Installing node-red in cloud server and creating nodes

- In the server, the user needs to activate the ports 443 and 1880-1885. This will enable the mqtt to connect with the server.
- With the use of commands provided by the professor and internet, node-red should successfully install in the server.
- It is required to install npm and admin packages to work.
- To run node-red, its necessary to run a command in the server.
- In google by typing tempsensor.space:1880 it should open node-red
- Node red received the message and by the use of nodes it will feed the data to the graph

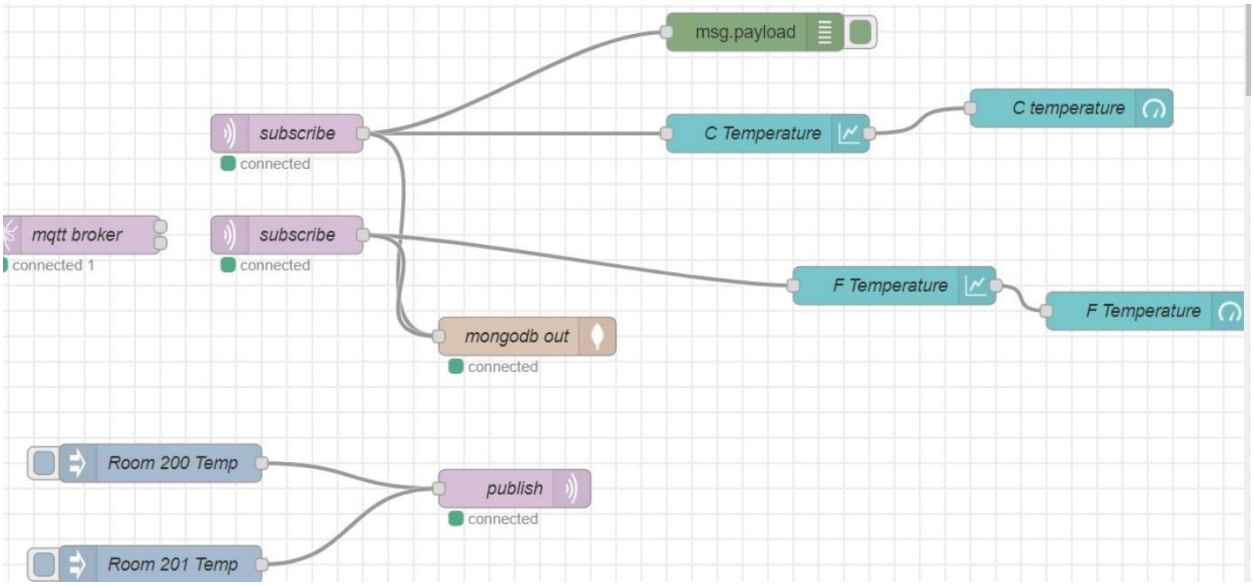

Figure 3. Node-red flow diagram.

- This program can enable a function dashboard that displays the temperature graphically and can be viewed from any device just by searching tempsensor.space:1880/iu/.
- In node-red we created groups in which the 2 values of temperature will be part of.

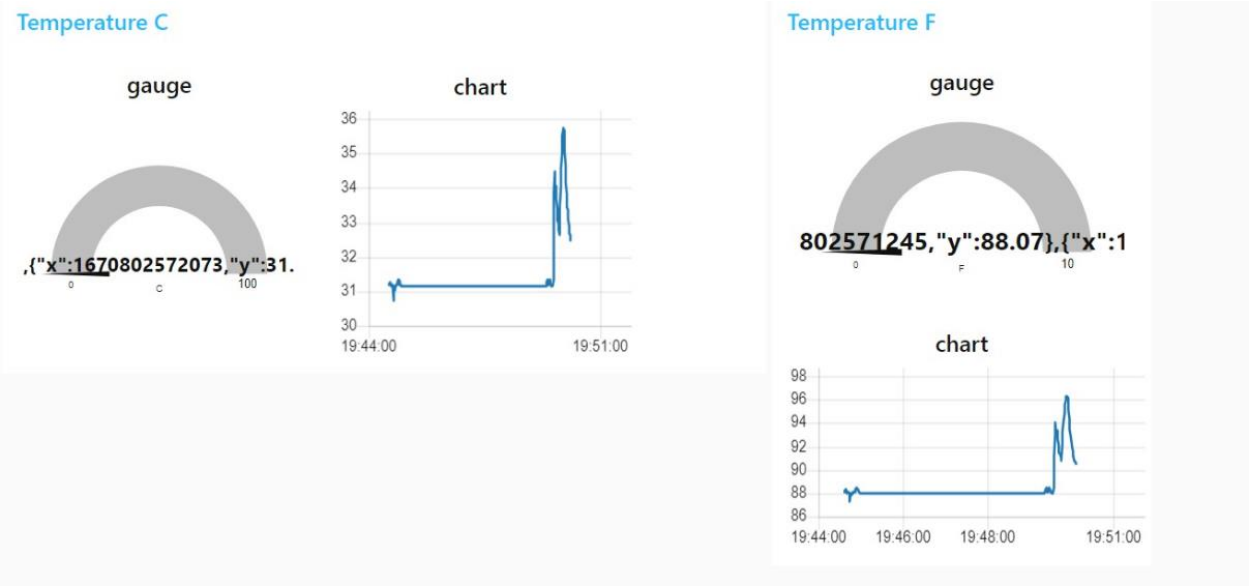

Figure 4. Dashboard used to display temperature.

Connecting MQTT to server

- Using Mongodb install the packages to the server.
- Once they are installed, launch it, and verify that is running.
- Download Studio 3T and create a ssh tunnel between the server and the mqtt.
- Add Pubsubclient library to be able to establish connection from the ESP32 to the MQTT broker.

Adding “Hey Siri” function

- The process was implemented by adding 8581 ports to the server
- Using Homebridge, the temperature will be registered and with the use of siri it will notify.

Conclusion and revision of goals

- The connection between the mqtt and the sensor, was successfully however due to the instability of the server, the program keeps crashing. This affects node-red and the server perse. Most of the goals were complete except for the setup of Siri. Cause homebridge is a service that need space and the server has limited space. It was eliminated from server because it was affecting node-red and the mqtt connection. The sensor was able at the end to send temperatures to the mqtt and put it in the dashboard.