Jonas Cassiano dos Santos
IVYHME
Prof. Döníz Borsos

Indoor Positioning System

8th December 2021

OVERVIEW

Our main objective is to create a system that is capable of tracking people inside a building or a room. For this project, I have decided to use WIFI instead of other methods that involve Bluetooth Beacons or Tags. The main reason for that is the lower cost and the usability for common users.

SPECIFICATIONS

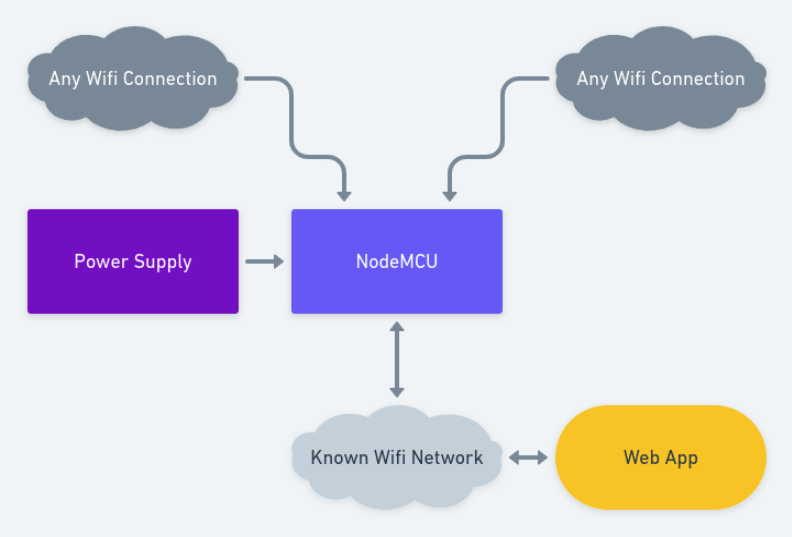
For this project, we will use a NodeMCU module, three or more WIFI networks nearby, and Machine learning.

The device should detect in which room is it located inside of a building by comparing its distance to the three WIFI networks. It will use a trained Machine Learn model to use this collected data and calculate its location. The NodeMCU will output the location through a Web Application.

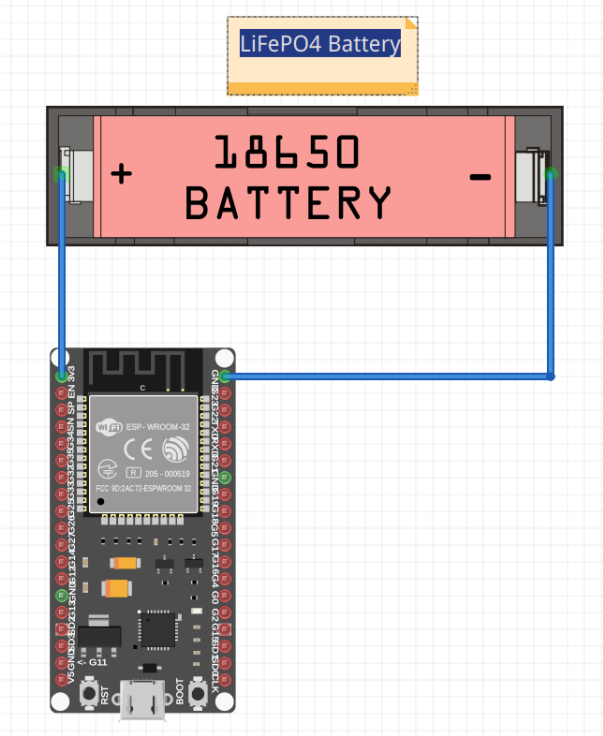
We will need to train a Machine learning model separately for a specific building with many samples per room and then port it to the NodeMCU so it can predict its location.

LOGICAL AND PHYSICAL SYSTEM DESIGN

1. Logical:



2. Physical system design:



COST

1x Ipari LFP 18650P 3.2V 1100mAh LiFePO4 Battery - 2300 Ft

1x NodeMCU ESP32 / NodeMCU32 (ESP32-WROOM-32D, CP2102, 38pin) - 4500 Ft

For development purposes, Powerbanks were used instead of the LiFePO4 Battery.

DESCRIPTION OF PARTS

Ipari LFP 18650P 3.2V 1100mAh LiFePO4 Battery

Will be used to Power the NODEMCU

NodeMCU ESP32 / NodeMCU32 (ESP32-WROOM-32D, CP2102, 38pin)

Main part of the project. Will be used attached to the User to detect its current location

WIFI SCAN CODE

This program will print the strength of all nearby networks every second. We first need to set the Name of the ROOM and walk around with the notebook.

This will be used later to train the Machine learning model

```
#include <WiFi.h>
#define print(string) Serial.print(string);
#define quote(string) print('"'); print(string); print('"');
String location = "";
void setup() {
   Serial.begin(115200);
   delay(3000);
   WiFi.disconnect();
   print("Enter 'scan {location}' to start the scanning\n");
}
void loop() {
   if (Serial.available()) {
       String input = Serial.readStringUntil('\n');
       if (input.indexOf("scan ") == 0) {
           location = input.substring(5);
           print("Scan Start...");
           print(location);
           print("\n");
       }
   }
   if (location != "") {
       int numNetworks = WiFi.scanNetworks();
       print('{');
       quote("__location");
       print(": ");
       quote(location);
       print(", ");
       for (int i = 0; i < numNetworks; i++) {
           quote(WiFi.SSID(i));
           print(": ");
```

```
            print(WiFi.RSSI(i));
            print(i == numNetworks - 1 ? "}\n" : ", ");
        }
        delay(1000);
    }
}
```

Machine Learning Converter

Now we have to generate a C code that can convert the Network SSID scanned into a feature vector so we can use it for classification.

I have decided to use the Library **micromlgen.** It needs to be installed using **pip**

We just need to plug in the Data we collected before into the code and run the Python script.

```
from micromlgen import port_wifi_indoor_positioning

if __name__ == '__main__':
    samples = '''
    {"__location": "Bedroom", "WARRIOR": -57, "Vodafone-C4C6": -75,
    {"__location": "Bedroom", "WARRIOR": -63, "Vodafone-C4C6": -80,
        ...
        ...
    {"__location": "Bedroom", "WARRIOR": -56, "Vodafone-C4C6": -79,
    {"__location": "Bedroom", "WARRIOR": -60, "I ragazzi della via Pal":
    '''
    X, y, classmap, converter_code = port_wifi_indoor_positioning(samples)
    print(converter_code)
```

Machine Learning Classifier

Now we have to generate a C code that will actually classify the scanned Network SSID into one of the rooms in the building.

I have devices to use the Decision Tree Classifier.

We can use the Decision Tree algorithm from the Scikit Python library

Here is the code:

```
from sklearn.tree import DecisionTreeClassifier
from micromlgen import port_wifi_indoor_positioning, port

if __name__ == '__main__':
   samples = '''
  {"__location": "Bedroom", "WARRIOR": -57, "Vodafone-C4C6": -75,
  {"__location": "Bedroom", "WARRIOR": -63, "Vodafone-C4C6": -80,
   ...
   ...
  {"__location": "Bedroom", "WARRIOR": -56, "Vodafone-C4C6": -79,
  {"__location": "Bedroom", "WARRIOR": -60, "I ragazzi della via Pal":
  '''

   X, y, classmap, converter_code = port_wifi_indoor_positioning(samples)
   clf = DecisionTreeClassifier()
   clf.fit(X, y)
   print(port(clf, classmap=classmap))
```

The main code

This code will be finally uploaded to the NODEMCU. It will classify into one of the locations and then send the data to a FIREBASE Realtime Database.

```
#include <Arduino.h>
#include "Converter.h"
#include "Classifier.h"
#if defined(ESP32)
#define USER_NODE 1
#include <WiFi.h>
#include <FirebaseESP32.h>
#elif defined(ESP8266)
#define USER_NODE 2
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#endif

#define FIREBASE_HOST "HOST_URL"
#define FIREBASE_AUTH "API_KEY"

int lp = 0;
int lp_time = 15;

char push_data[200];
unsigned long oldTime;
const char * ssid = "KNOWN_WIFI_SSID";
const char * password = "WIFI_PASSWORD";
String location;

FirebaseData fbdo;

Eloquent::Projects::WifiIndoorPositioning positioning;
Eloquent::ML::Port::DecisionTree classifier;

void setup() {
    Serial.begin(115200);
```

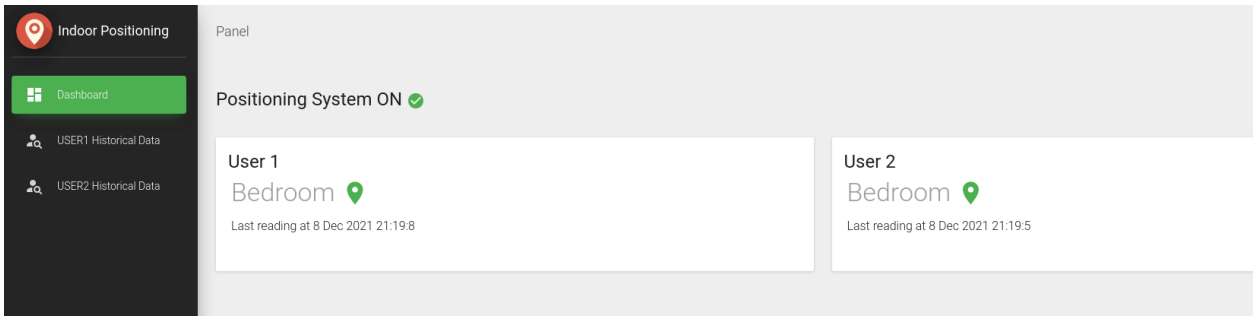
```
    delay(10);
    Serial.println('\n');

    startWIFI();

    oldTime = 0;

    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
    if (WiFi.status() == WL_CONNECTED && (millis() - oldTime) >
1000)  {
        oldTime = millis();
        lp++;

        if(lp >= 6){
            positioning.scan();
            Serial.println(" ...");
            lp=0;

            FirebaseJson json;
            FirebaseJson json2;


            location =
classifier.predictLabel(positioning.features);

            Serial.print("Location: ");
            Serial.println(location);

            json.set("room", location);
            json.set("Ts/.sv", "timestamp");

            if (Firebase.RTDB.pushJSON(&fbdo, "/USER" +
USER_NODE + "/ROOM", &json)) {
```

```
            Serial.println(fbdo.dataPath());
            Serial.println(fbdo.pushName());
            Serial.println(fbdo.dataPath() + "/"+
fbdo.pushName());
          } else {
            Serial.println(fbdo.errorReason());
          }
        }
    } else if (WiFi.status() != WL_CONNECTED) {
        startWIFI();
    }
}

void startWIFI(void) {
    delay(100);

    WiFi.begin(ssid, password);
    Serial.print("Connecting to ");
    Serial.print(ssid);
    Serial.println(" ...");
    oldTime = 0;
    int i = 0;
    delay(100);

    while (WiFi.status() != WL_CONNECTED) {
        delay(2000);
        Serial.print(++i);
        Serial.print('.');
        delay(100);
    }
    delay(2000);
    Serial.print('\n');
    Serial.print("Connected!\n");
    Serial.print("IP address:\t");
    Serial.print(WiFi.localIP());
}
```

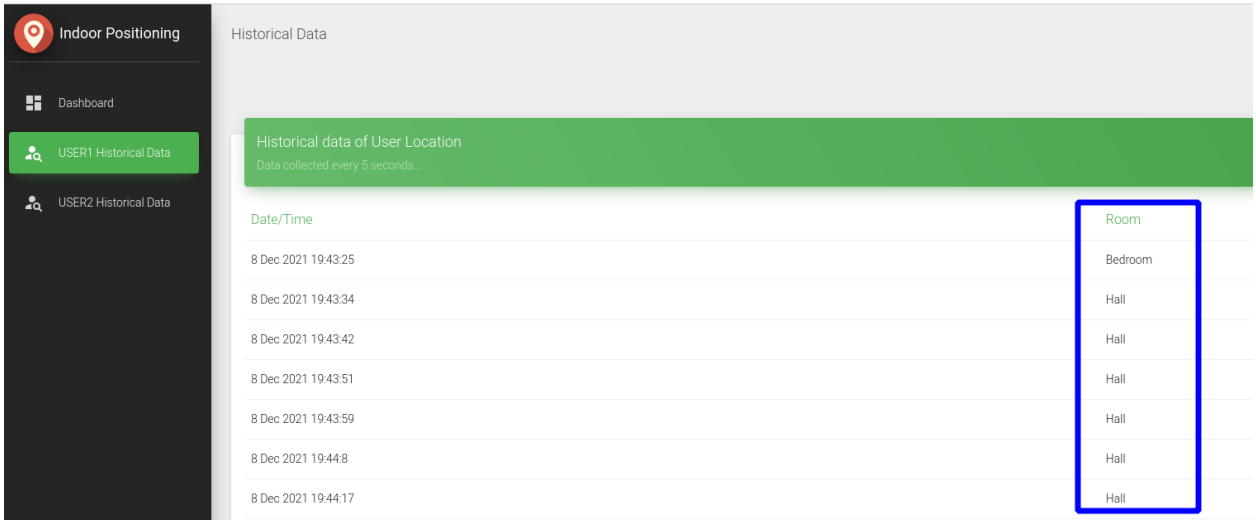
The Web Application

The web application is hosted at https://indoor.jonascsantos.com

We can see in Realtime were the users are located.



We can also check the historical data of the User Location as well as the Date and Time of the measure.



The website was developed using VueJS.

The source code is going to be sent as an attachment.