

Hasaan Ali Alhamada

Call System

EE-499

Christina Howe/ Tony Richardson

University of Evansville

April 26, 2019

Huge thanks to my advisor Tony Richardson for believing in me and giving me suggestions for the project. Also, my professor Christina Howe for making me push myself beyond my limits and do my best to make my project the best that it can be. Last but not least, the school for providing me with the tools and the parts I need to complete my project.

Table of Contents:

Introduction..... 4

State of The Problem..... 4

Project Design..... 6

Result 7

Conclusion and Recommendations 6

Appendices 8-11

Figures:

Figures 1..... 6

Figures 2..... 6

Figures 3..... 6

Introduction:

Call system is basically the nursing home call system, but more efficient. The call system has two main parts: leather bracelet and 3D box. The bracelet has a circuit board, a push button, ESP8266, and a battery. Whenever the button is pressed, a signal goes through WIFI to the raspberry pi in the 3D box, the pi will send an email to the nurse's phone saying, "Resident xxx needs your help". Not only that, the signal from ESP8266 makes the raspberry pi make voice, the voice will be the same thing that is written in the email. The purpose of the voice is to get the nurses attention in more than one way. Going back to the 3D box, the box has the raspberry pi and power bank. The power bank is used to keep the pi power all the time, that means the project will work all the time considering that the Wi-Fi has power even if the electric is off in the building.

Statement of The Problem:

Nursing homes have residents that need assistance all day long, but sometimes nurses have difficulties helping them whenever they need it, due to being busy and not paying attention to the resident's lights (the lights that used to get the nurses attention for help). Sometimes the residents don't have patience to wait, so they try to move from their bed or their chair and fall and get injured. Being an electrical engineer, it's useful to make a project to make the residents get help as soon as possible. Also, residents miss their families every day, there must be an easy way to make the residents reach their family without asking the nurses to call them.

Project Design:

There are seven parts used in the project: bracelet, the pi, battery, push button, 3D box, ESP 8266, and circuit board. The project has two main parts: the leather bracelet and the 3D box. The bracelet will be on the resident's wrist all the time. It is the main key for the project. Figure 1 is the bracelet. As it shown. The push button is connected to the ESP 8266. the button has power all the time and whenever the power get disconnected from the button, the ESP8266 will send the signal to the pi. Figure 2 is the circuit diagram of the bracelet and how it works. Moving to the 3D box, the pi will be on power all the time and waiting for signal to be sent from the bracelet. Figure 3 is the circuit diagram for the pi and power bank in the 3D box and how the rest of the project works. There will be a wire connected to the outlet all the time, it will give the power bank power all the time in case the building is out of electricity due to natural disasters. Part of the project is coding in the ESP 8266 using Arduino IDE program and the other part is a website called node red. The Node-RED is a visual tool for wiring the Internet of Things that can be run on a Raspberry Pi and allows for rapid prototyping of project. The pi will receive the signal from the ESP 8266 through the WIFI, the signal called "alarm". When the pi receives the signal, it will turn on the function to send the signal and make the computer talk. The node red URL must have the IP address for the pi, the ESP8266 and the pi must connect to one WIFI signal for the project to work.



Figure 1

Figure 2

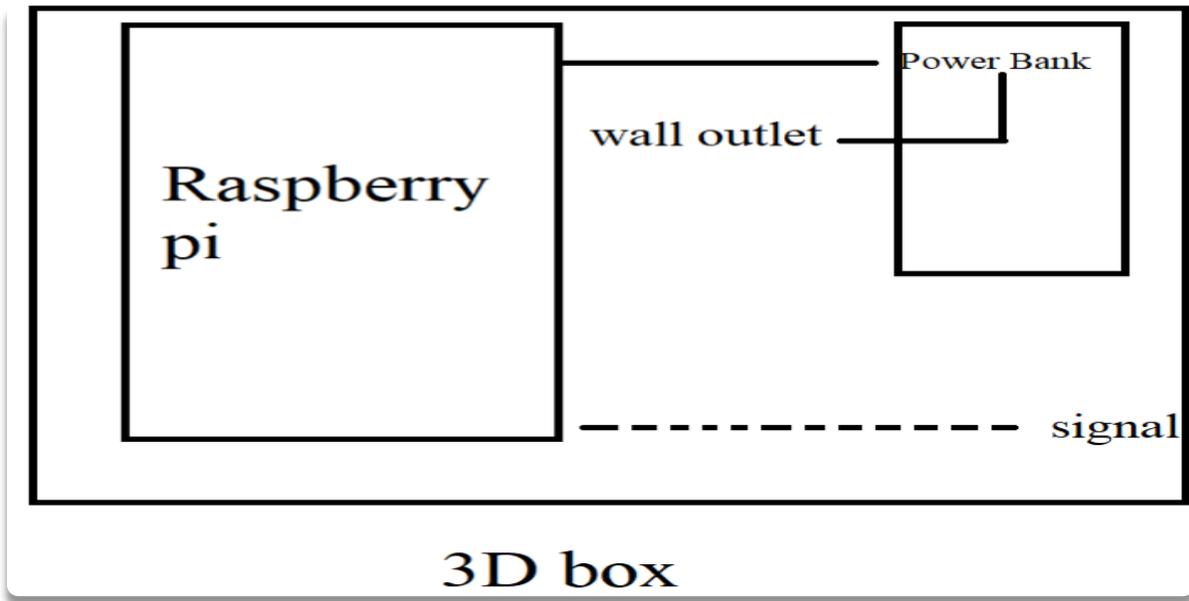


Figure 3

Result:

The project worked the exact way I wanted it to work. The first couple months I used a Bluetooth module, but the Bluetooth has a short distance radius, so it doesn't help the residents if they need to use the project when they are far away from the pi. There was some struggle in the beginning trying to figure out which kind of battery to use for the ESP8266. After looking online and searching through stores, I used a battery that has 5V and 500mAh. I tried to use a battery with 900mah and it didn't work. There was too much current that made the voltage drop. Also, the ESP8266 can get up to 10V, but for safety I used 5V. The power bank that was used in the project is one of the best in the market, that helps if the power in the building is off, the pi will last for a long time.

Conclusion and Recommendations:

The call system is a good project for the nursing homes, it can be used for other purposes, but the main reason I picked this project is for the nursing home. The concept of the project is simple, using two main parts: the pi and the ESP 8266 and make the communication through the WIFI. One idea that can be used in the project is to make another button in the bracelet, saying that the resident got assisted, so no need for other nurses to check on them and waste time. Not only that, but also make a light color for each resident, so when a resident pushes the button, the light will turn on and the other push button can be used to turn off the light. the lights will be in the nursing station, the nurses will know which color belongs to specific residents, it will help get the nurses attention by three ways: the email, the sound from the computer, and the lights.

Appendices:

The code for the ESP8266:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
const char* ssid = "iPhone";          //WIFI name
const char* password = "hassanali";   //WiFi Password
const char* mqtt_server = "172.20.10.4"; //Raspberry Pi IP address
WiFiClient espClient; //to connect esp client
PubSubClient client(espClient); //to tell the server the client name
// constants won't change. They're used here to set pin numbers:
const int buttonPin = D5; // the number of the pushbutton pin
const int ledPin = LED_BUILTIN; // the number of the LED pin
// variables will change:
int buttonState = 0; // variable for reading the pushbutton status
void setup_wifi()
{
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - ESP IP address: ");
```

```
Serial.println(WiFi.localIP()); //if it connected, it show you the ip address for the network
}
void reconnect()
{
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("ESP8266Client"))
    {
      Serial.println("connected");
    }
    else {
      Serial.print("failed");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);

  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}
```

```
void loop() {  
  if (!client.connected()) {  
    reconnect();  
  }  
  if (!client.loop())  
  int buttonState = digitalRead(buttonPin);  
  
  int Email = 1994;  
  float gmail = Email;  
  static char email[15];  
  dtostrf(gmail, 6, 2, email);  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:  
  if (buttonState == LOW) {  
    // turn LED on:  
    digitalWrite(LED_BUILTIN, HIGH);  
    Serial.println("button Pressed...");  
    Serial.println("#####");  
    client.publish("alarm", email);  
    delay(5000);  
  }  
  else  
  {  
    // turn LED off:  
    digitalWrite(LED_BUILTIN, LOW);  
  }  
}
```

```
Serial.println("button not Pressed...");  
Serial.println("#####");  
}  
}
```

The Node Red URL:

172.20.10.4:1880

It might change if this project connected to different WIFI and from that, the IP address needs to be changes in the code in the ESP8266. The IP address for the pi can be found if you put the mouse on the WIFI signal in the PI.