An-Najah National University

Computer Engineering Department

Special topics in networks (Wireless) - 10636554

Project- Adhoc ESP Network

Student: Jamal SaadEddin

Instructor: Dr. Hanaal AbuZanat

Dec,2023

Introduction

In this project, I'm asked to make an adhoc network of 3 ESP modules, 2 of them must be away of each others, and one of them will be in the middle between them, and it'll see them both.

The reason of this, the one in the middle will carry all messages that come from the first ESP and forward it to the third one.

To do that, we have to make an adhoc network as we learned in the course.

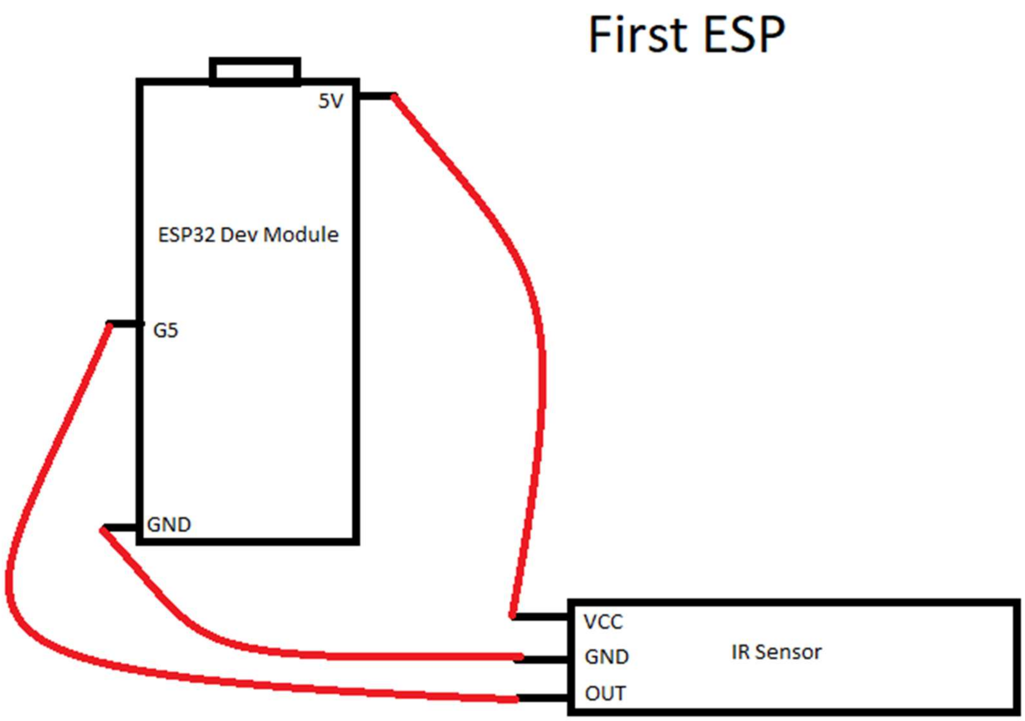
Tools Used

The following tools were used in the project:

- 3 ESP32 Dev Modules
- IR Sensor
- 3 USB cables
- Connecting wires
- 3 Breadboards
- LCD Screen
- 2 LEDs with 2 resistors with 3 Push Buttons
- 2 Battery PowerBanks
- Arduino IDE to write the code

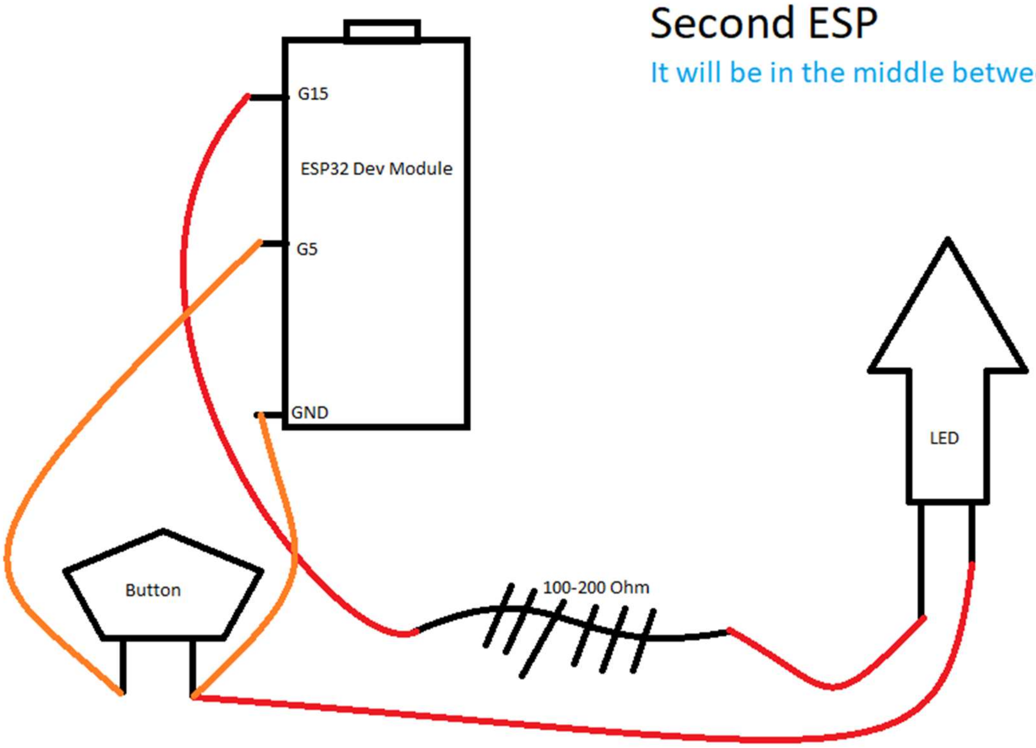
Work Steps

Installation and networking

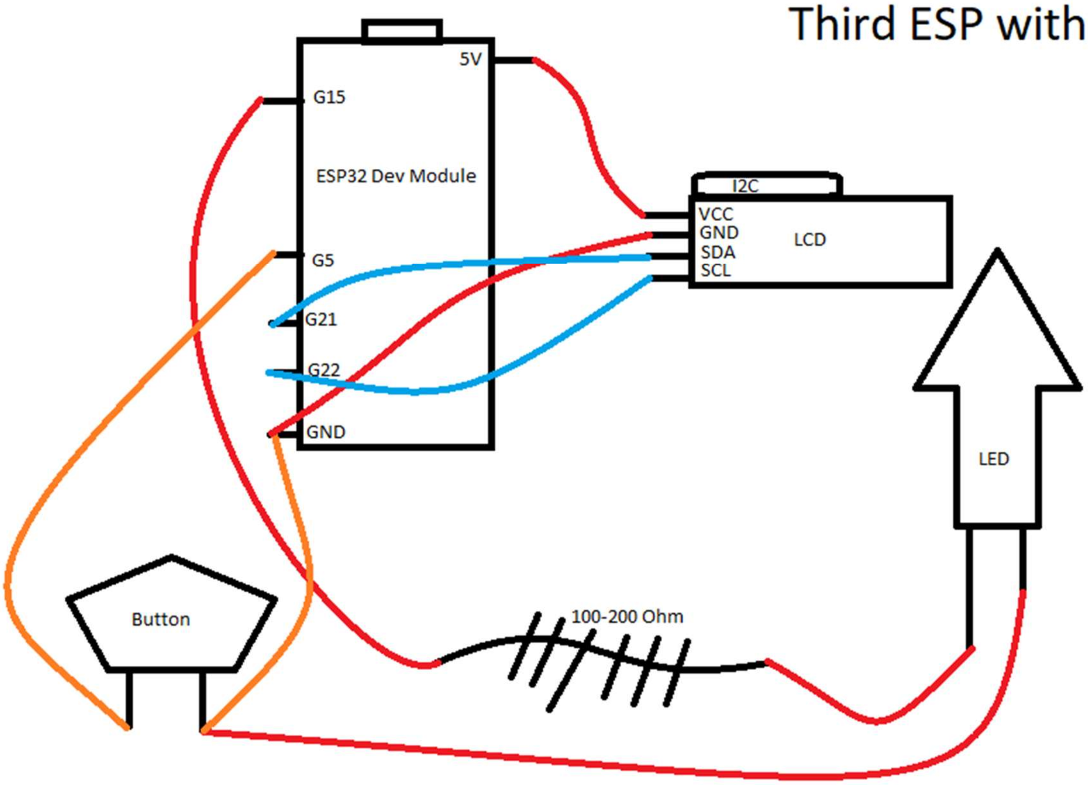


Second ESP

It will be in the middle between other ESPs

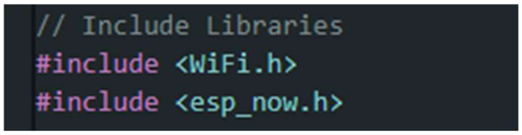


Third ESP with LCD



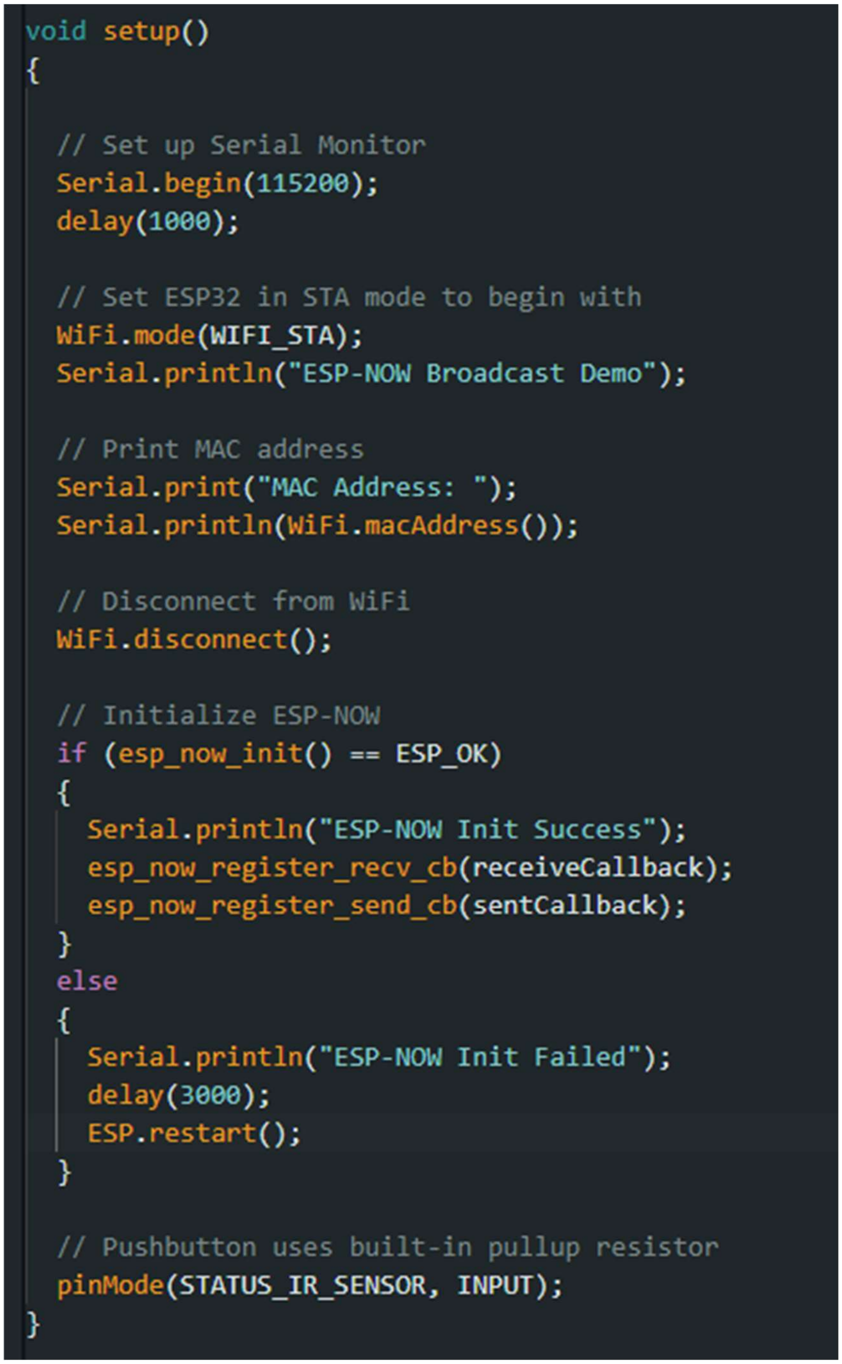
Programming ESPs

1- First ESP with IR Sensor



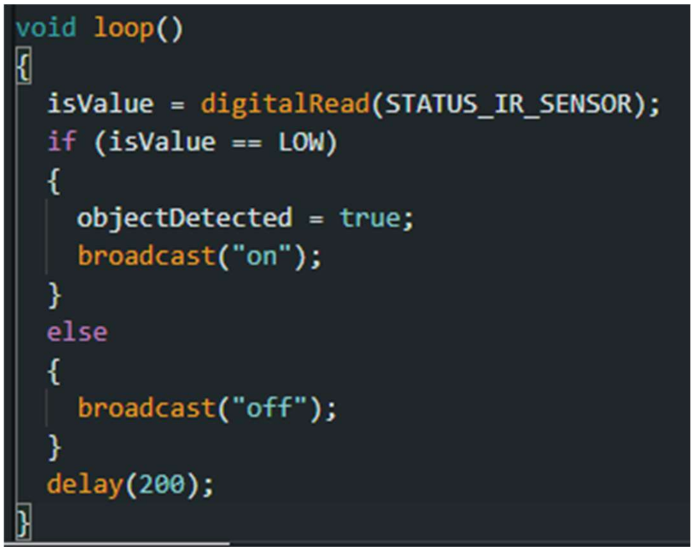
```
// Include Libraries
#include <WiFi.h>
#include <esp_now.h>
```

Include nessesarry libraries



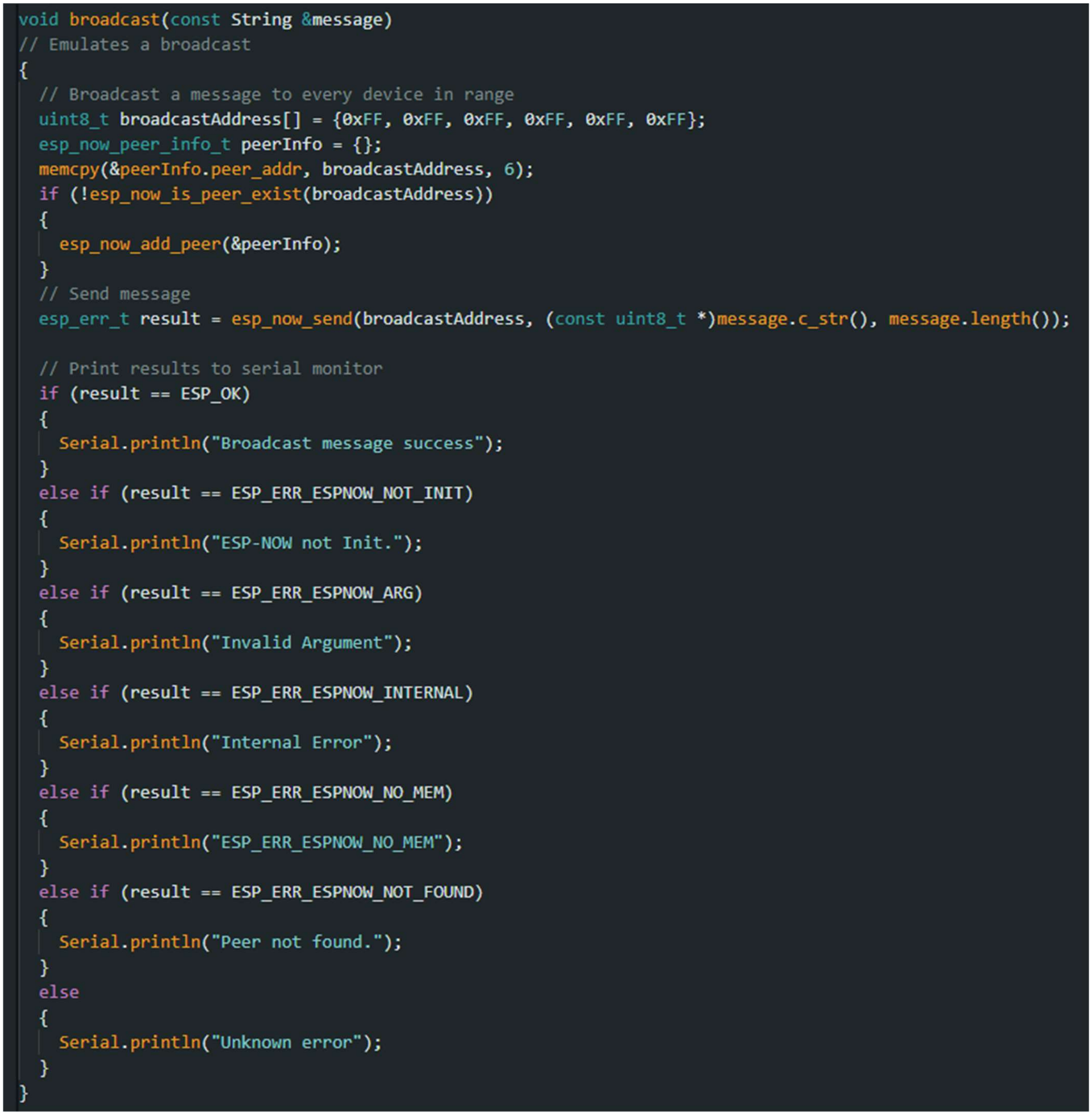
```
void setup()
{

  // Set up Serial Monitor
  Serial.begin(115200);
  delay(1000);

  // Set ESP32 in STA mode to begin with
  WiFi.mode(WIFI_STA);
  Serial.println("ESP-NOW Broadcast Demo");

  // Print MAC address
  Serial.print("MAC Address: ");
  Serial.println(WiFi.macAddress());

  // Disconnect from WiFi
  WiFi.disconnect();

  // Initialize ESP-NOW
  if (esp_now_init() == ESP_OK)
  {
    Serial.println("ESP-NOW Init Success");
    esp_now_register_recv_cb(receiveCallback);
    esp_now_register_send_cb(sentCallback);
  }
  else
  {
    Serial.println("ESP-NOW Init Failed");
    delay(3000);
    ESP.restart();
  }

  // Pushbutton uses built-in pullup resistor
  pinMode(STATUS_IR_SENSOR, INPUT);
}
```

Initilalization of Serial Monitor and ESP and IR sensor.



```
void loop()
{
  isValue = digitalRead(STATUS_IR_SENSOR);
  if (isValue == LOW)
  {
    objectDetected = true;
    broadcast("on");
  }
  else
  {
    broadcast("off");
  }
  delay(200);
}
```

In the main loop, if IR sensor detecting object, I'll broadcast that to other ESPs.

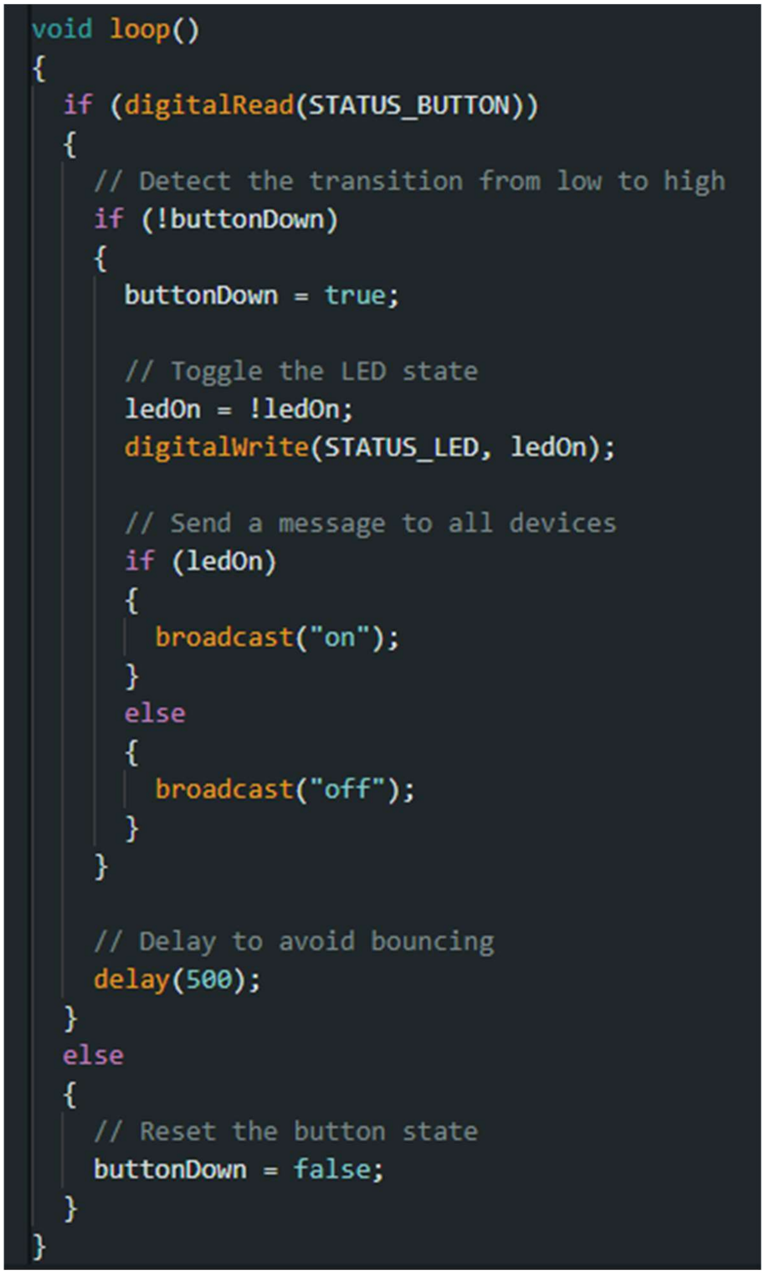


```
void broadcast(const String &message)
// Emulates a broadcast
{
  // Broadcast a message to every device in range
  uint8_t broadcastAddress[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
  esp_now_peer_info_t peerInfo = {};
  memcpy(&peerInfo.peer_addr, broadcastAddress, 6);
  if (!esp_now_is_peer_exist(broadcastAddress))
  {
    esp_now_add_peer(&peerInfo);
  }
  // Send message
  esp_err_t result = esp_now_send(broadcastAddress, (const uint8_t *)message.c_str(), message.length());

  // Print results to serial monitor
  if (result == ESP_OK)
  {
    Serial.println("Broadcast message success");
  }
  else if (result == ESP_ERR_ESPNOW_NOT_INIT)
  {
    Serial.println("ESP-NOW not Init.");
  }
  else if (result == ESP_ERR_ESPNOW_ARG)
  {
    Serial.println("Invalid Argument");
  }
  else if (result == ESP_ERR_ESPNOW_INTERNAL)
  {
    Serial.println("Internal Error");
  }
  else if (result == ESP_ERR_ESPNOW_NO_MEM)
  {
    Serial.println("ESP_ERR_ESPNOW_NO_MEM");
  }
  else if (result == ESP_ERR_ESPNOW_NOT_FOUND)
  {
    Serial.println("Peer not found.");
  }
  else
  {
    Serial.println("Unknown error");
  }
}
```

Broadcast function: it will send the message to other ESPs that it can see them and send to them.
You can notice that the broadcast address is public, so it’s not specified to whom it’s sending, any ESP in the area will receive the message.
Other code prints the state of the result of the broadcast process.

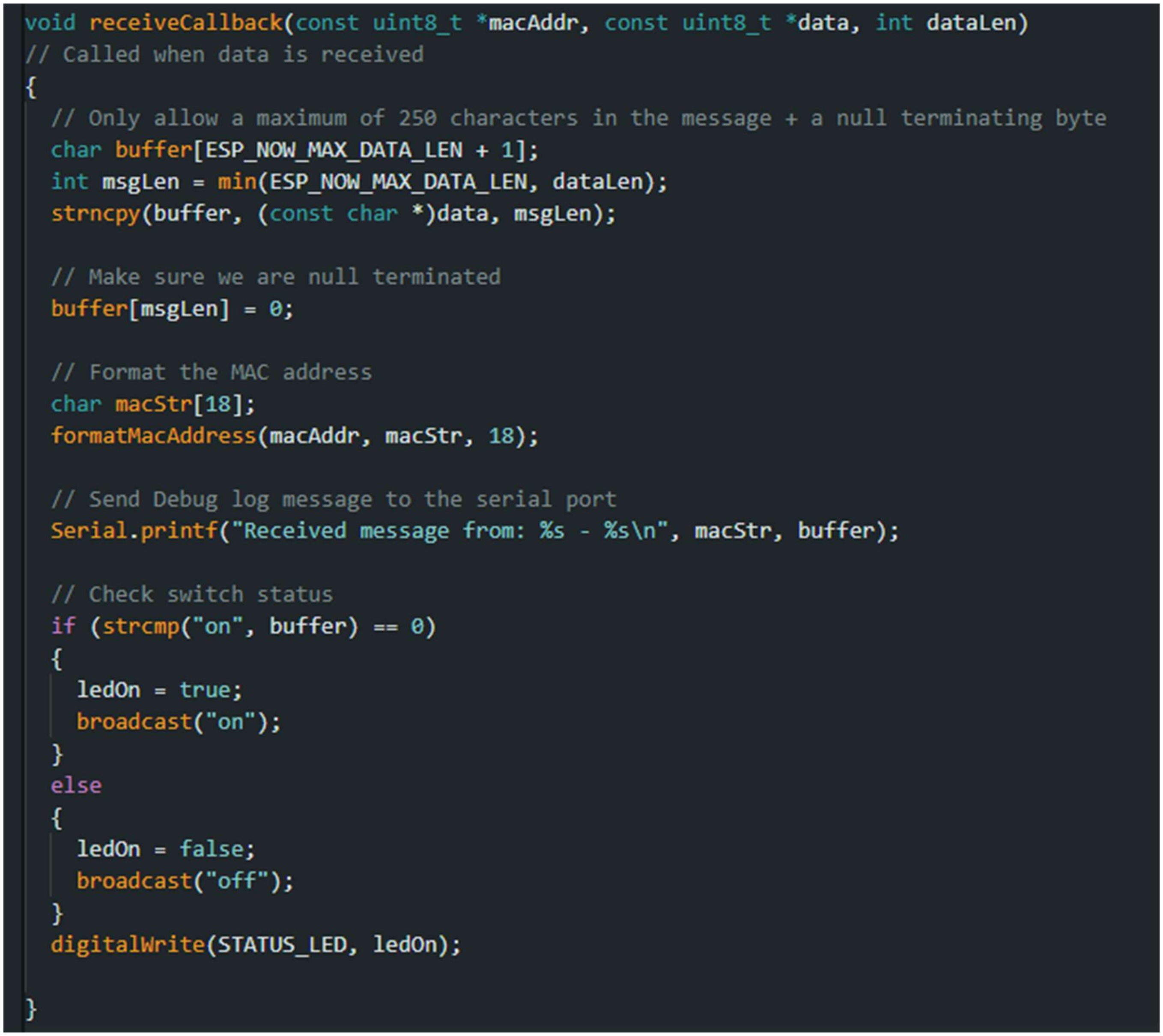
2- Second ESP with button and LED only

The same code as first ESP, but the difference in the main void loop:



```
void loop()
{
  if (digitalRead(STATUS_BUTTON))
  {
    // Detect the transition from low to high
    if (!buttonDown)
    {
      buttonDown = true;

      // Toggle the LED state
      ledOn = !ledOn;
      digitalWrite(STATUS_LED, ledOn);

      // Send a message to all devices
      if (ledOn)
      {
        broadcast("on");
      }
      else
      {
        broadcast("off");
      }
    }

    // Delay to avoid bouncing
    delay(500);
  }
  else
  {
    // Reset the button state
    buttonDown = false;
  }
}
```

Here, if the button is pressed, I'll turn my LED ON and broadcast to others to turn their LEDs on.
If pressed another time, I will toggle it back to off, and broadcast that again to other ESPs in the area.

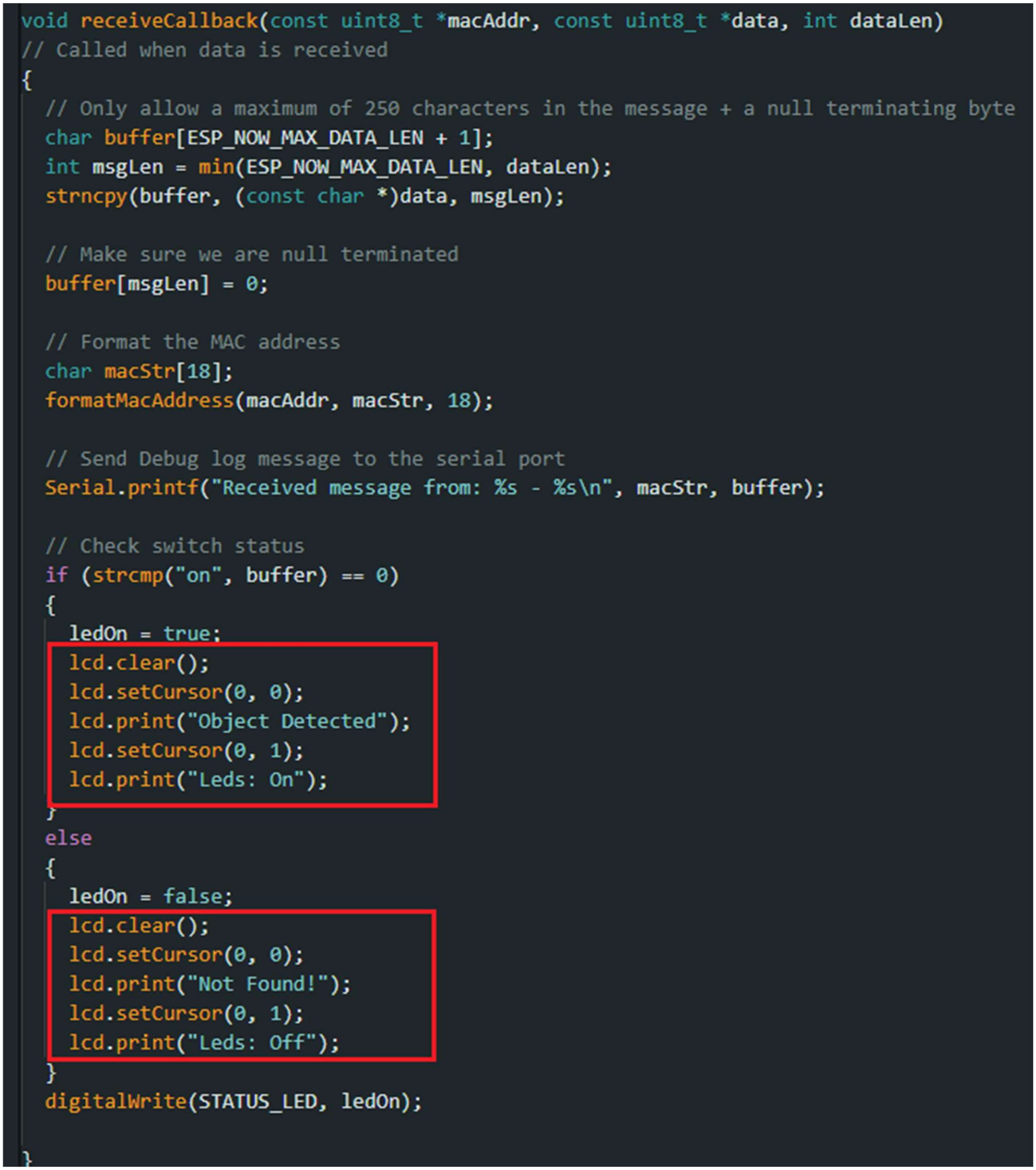


```
void receiveCallback(const uint8_t *macAddr, const uint8_t *data, int dataLen)
// Called when data is received
{
  // Only allow a maximum of 250 characters in the message + a null terminating byte
  char buffer[ESP_NOW_MAX_DATA_LEN + 1];
  int msgLen = min(ESP_NOW_MAX_DATA_LEN, dataLen);
  strncpy(buffer, (const char *)data, msgLen);

  // Make sure we are null terminated
  buffer[msgLen] = 0;

  // Format the MAC address
  char macStr[18];
  formatMacAddress(macAddr, macStr, 18);

  // Send Debug log message to the serial port
  Serial.printf("Received message from: %s - %s\n", macStr, buffer);

  // Check switch status
  if (strcmp("on", buffer) == 0)
  {
    ledOn = true;
    broadcast("on");
  }
  else
  {
    ledOn = false;
    broadcast("off");
  }
  digitalWrite(STATUS_LED, ledOn);

}
```

Here in the receive call back function, if I receive from others that the state is on, I will turn my LED ON, else OFF And Note that I'll broadcast that again to others, but **why?** Because suppose that IR sensor detected object, I will broadcast that to other ESPs that can not hear that state from IR ESP Module.. and so on, and this is the **adhoc** that we are talking about.

3- Third ESP with LCD Screen:

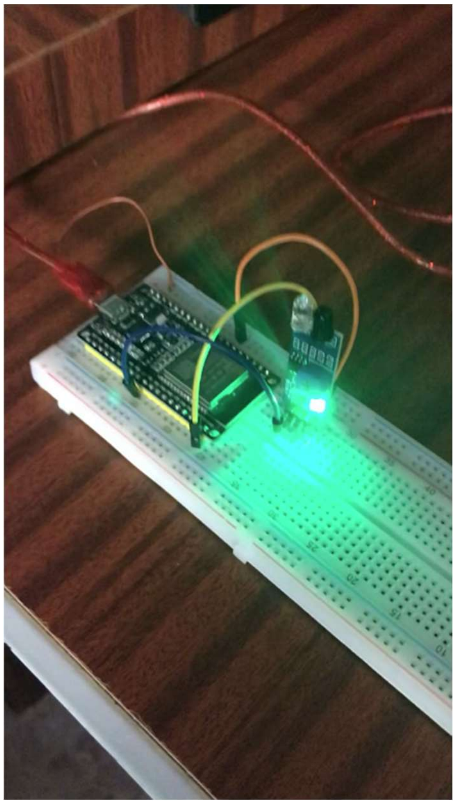
Same code as the second ESP because it has the same tools (button and led), but here we have LCD screen, so the modification in the code will be to print the results in the **screen to verify adhoc network working successfully.**



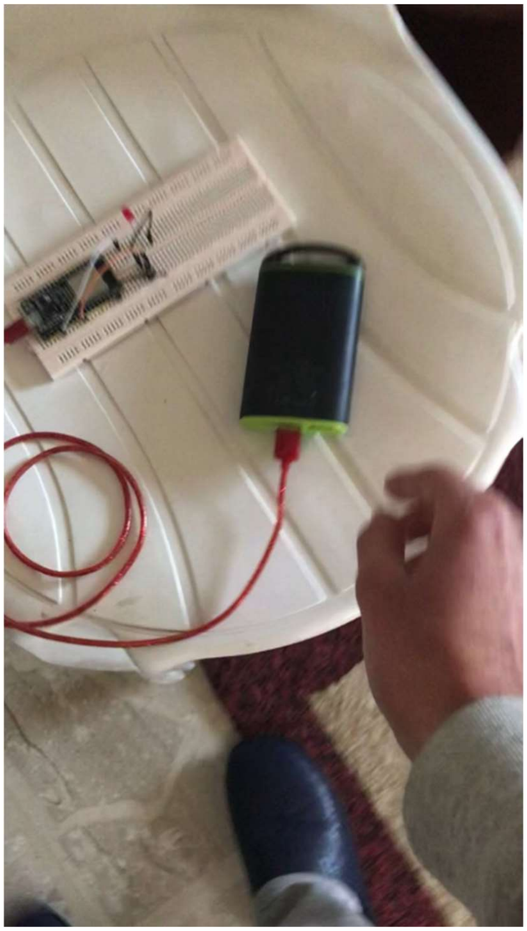
```
void receiveCallback(const uint8_t *macAddr, const uint8_t *data, int dataLen)
// Called when data is received
{
  // Only allow a maximum of 250 characters in the message + a null terminating byte
  char buffer[ESP_NOW_MAX_DATA_LEN + 1];
  int msgLen = min(ESP_NOW_MAX_DATA_LEN, dataLen);
  strncpy(buffer, (const char *)data, msgLen);

  // Make sure we are null terminated
  buffer[msgLen] = 0;

  // Format the MAC address
  char macStr[18];
  formatMacAddress(macAddr, macStr, 18);

  // Send Debug log message to the serial port
  Serial.printf("Received message from: %s - %s\n", macStr, buffer);

  // Check switch status
  if (strcmp("on", buffer) == 0)
  {
    ledOn = true;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Object Detected");
    lcd.setCursor(0, 1);
    lcd.print("Leds: On");
  }
  else
  {
    ledOn = false;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Not Found!");
    lcd.setCursor(0, 1);
    lcd.print("Leds: Off");
  }
  digitalWrite(STATUS_LED, ledOn);

}
```

Time to Test

1- Put first ESP with IR sensor in an Edge room of your house.

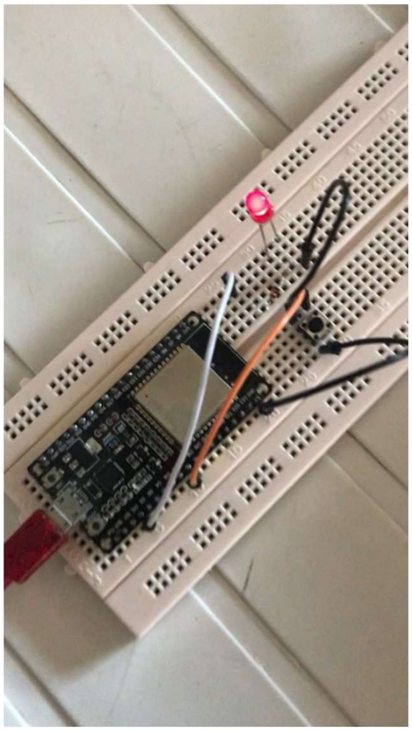


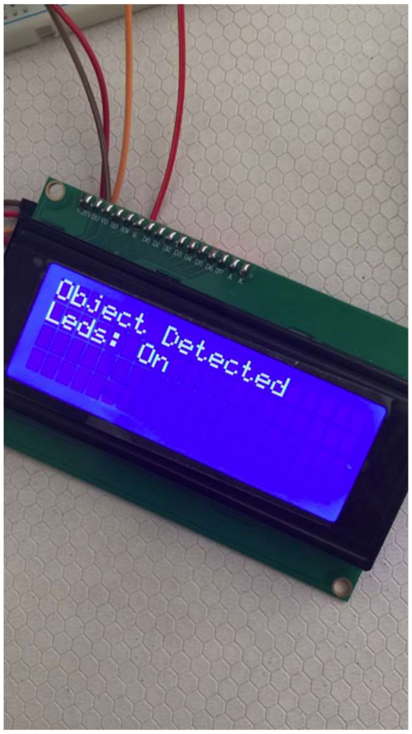
2- Put the ESP without LCD in the kitchen or any middle room in the house, but make sure that the distance will not disconnect you from the first ESP (keep in distance).



3- Put the third ESP with LCD screen in a far room from the first room, go away and make sure that you are disconnected from the first ESP, and connected to second ESP which we placed in the kitchen in our case.

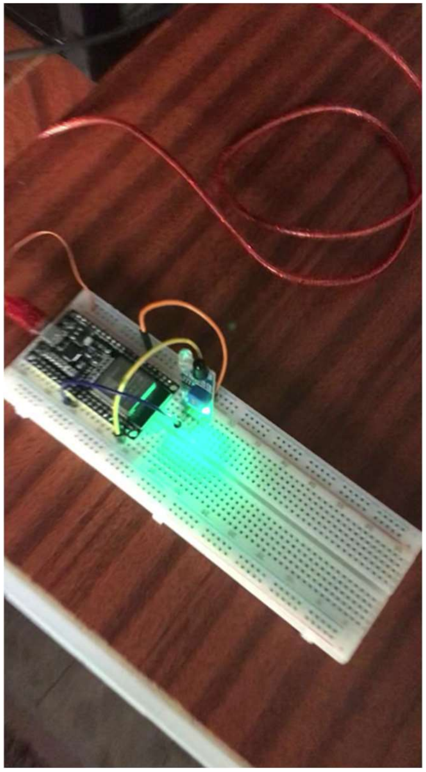
Now we are ready to test, put an object over the IR sensor, the LEDs of other ESPs must turn on, and the screen must print that also..

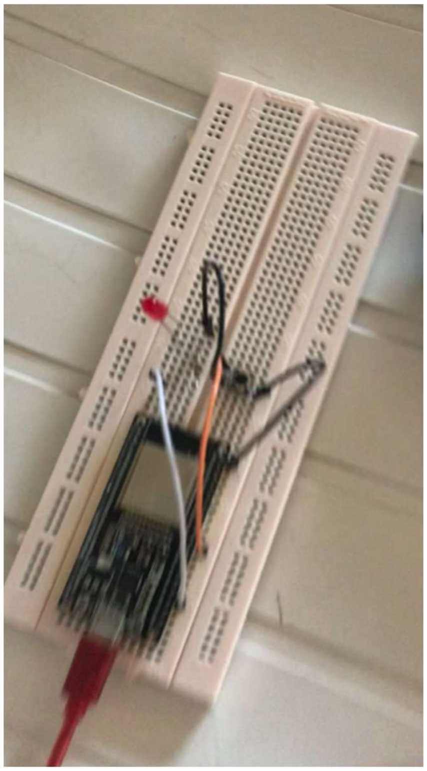


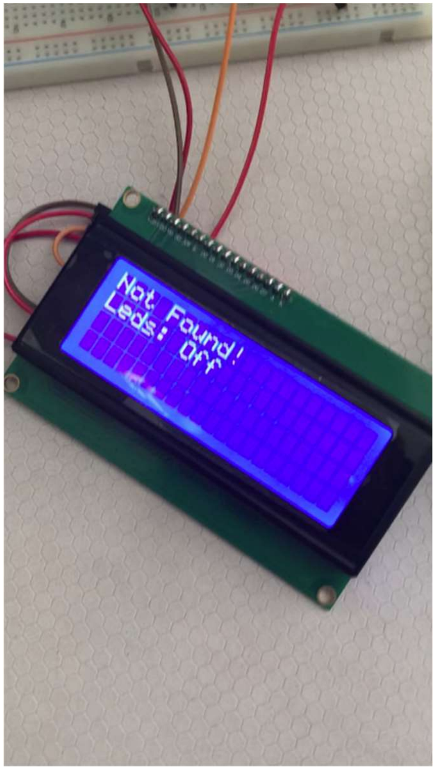


```
Object Detected
Leds: On
```

Now remove that object, and the LEDs will turn off, Not Found will be printed..

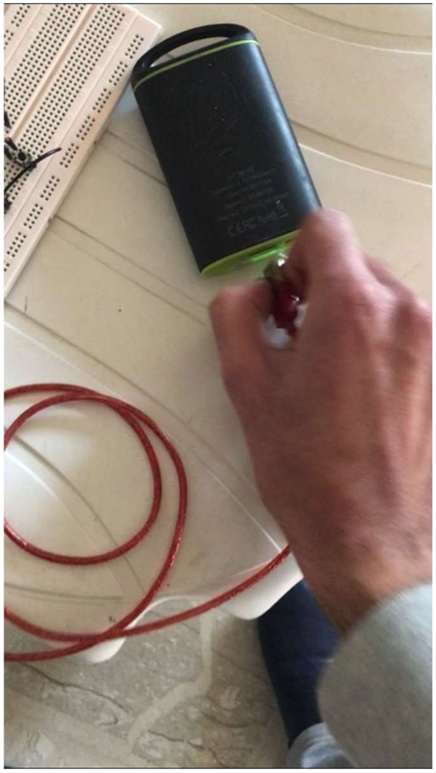


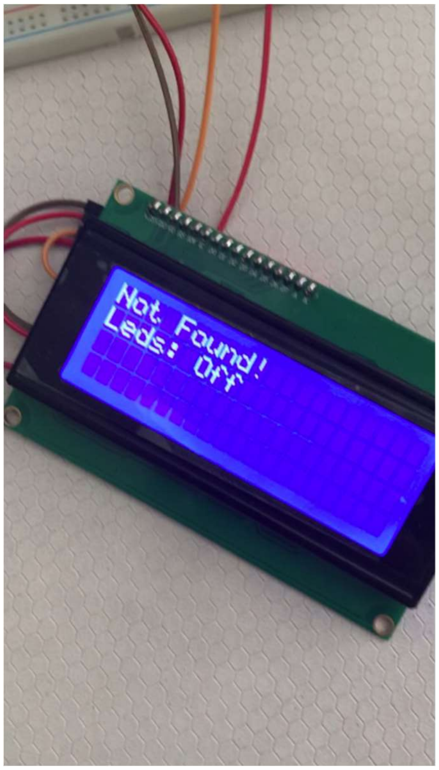




```
Not Found!
Leds: Off
```

Now let’s turn off the power of the middle ESP (in the kitchen), and re-put the object over the IR sensor, the signal will not reach the far ESP because no one will forward the messages…





Still Not Found and the LEDs are off, because they can not hear each others because the middle ESP is down.

Conclusion

In this project, I learned how to make an adhoc network of number of ESPs, how to broadcast messages to other ESPs in the area, and how to deal with receipt messages from other ESPs, and the most important point is to put hands on networking in real world application, it was a fun experience.