DATA SENDING FROM ONE ESP BOARD TO MULTIPLE ESP BOARDS VIA ESP_NOW PROTOCOL

Wint Shwe War Hlaing*

Abstract

In this research, the data such as temperature and humidity from one ESP board is transferred to the other two ESP boards using ESP_NOW protocol. ESP_NOW is one of the connectionless communication protocols. It can transfer data in the form of short packet transmission. Multiple devices can talk to each other in ESP_NOW protocol. In this research, data are transferred from one-to-many boards. One ESP32 board functioned as the data sender while two other ESP32 boards are used as receivers. ESP8266 boards or Wemos boards can be also used instead of ESP32 boards. DHT 11 sensor is used to obtain the data of temperature and humidity. The messages from the sender are visualized in a serial monitor or PuTTY. The receiving data are displayed on the I2C LCD as well as on the serial monitor or PuTTY. The proper data transferring range is also measured.

Keywords: ESP32, ESP_NOW, data transferring range, MAC address, PuTTY.

Introduction

Generally, the Wi-Fi routers operate with the 2.4 GHz band. Using a Wi-Fi router, the range of which data can be transferred is up to 150 feet (46 m) indoors and 300 feet (92 m) outdoors. The 802.11a routers with 5 GHz bands can reach approximately one-third of these distances [Dotdash, 2018]. The working function of ESP_NOW protocol is similar to the 2.4 GHz wireless connection which is used in wireless keyboards and mice.

Using ESP_NOW protocol, there are two main data transfers between ESP boards such as one-way and two-way communication. In one-way communication of ESP boards, two ways are categorized as (i) one ESP board transporting the data to multiple ESP boards and (ii) one ESP board receiving data from multiple ESP boards. An ESP32 (master) sends data to two ESP32 boards (slaves) in this research.

DHT 11 sensor is used to obtain the temperature and humidity. Three ESP32 boards are used; - one ESP32 for data sender and two ESP32 for data receiver. Two I2C LCDs are functioned to display the receiving data. Moreover, serial monitors and PuTTY monitoring systems are used for data visualization.

ESP32 (master) is connected to the DHT 11 sensor which data of temperature and humidity is obtained. The Media Access Control (MAC) addresses of receiver ESP32 boards (slaves) are obtained firstly. ESP32 (master) sends data to ESP32 boards (slave) via MAC address. Data sending success or failure to ESP32 (slaves) can be observed on the serial monitor or PuTTY monitoring window. Data receiving can be visualized on I2C LCD as well as serial monitor or PuTTY. And then, the range of which data are enabled to reach is determined on I2C LCD.

Experimental Procedure

Hardware, Software Equipment

In this research, three ESP32, DHT 11, and two I2C LCDs are used as the hardware components. Hardware such as the ESP32 library and DHT library are required to be included in Arduino IDE. ESP32 boards are not included in the default Arduino IDE. It can be installed in the Arduino IDE through the board manager. Master to slave communication is implemented on

* Dr, Associate Professor, Department of Physics, Yangon University of Education

ESP_NOW protocol. MAC address of each receiver ESP32 board can be determined by using software.

ESP32

ESP32 is one of the microcontroller chips fabricated by "Espressif". It consists of a microcontroller, Wi-Fi, Bluetooth, and Bluetooth low energy (BLE) for internet communication. ESP32 has upgraded the design of ESP8266 by integration with classical Bluetooth and BLE. It has a dual-core processor and it is faster than ESP8266. It can operate in sleep mode and is suitable for battery-powered applications.

In ESP32, there is eighteen 12-bit Analog to Digital converters (ADC), two 8-bit Digital to Analog converters (DAC) for signal processing. And then, there is four serial peripherals interface (SPI) channels, two inter-integrated circuits (I2C) interfaces, two inter-integrated sound(I2S) audio interfaces, and three Universal asynchronous receiver-transmitter (UART) for the peripheral interfacing. Moreover, eight channels of IR remote control and sixteen channels of pulse-width modulation (PWM) are added in ESP32. Hall- effect sensor, touch sensor, ultra-low-power preamplifier, and low-dropout (LDO) regulator are integrated into ESP32 chip. Multi-functional pin assignments of ESP32 are shown in Figure 1 [Bill, 2020].

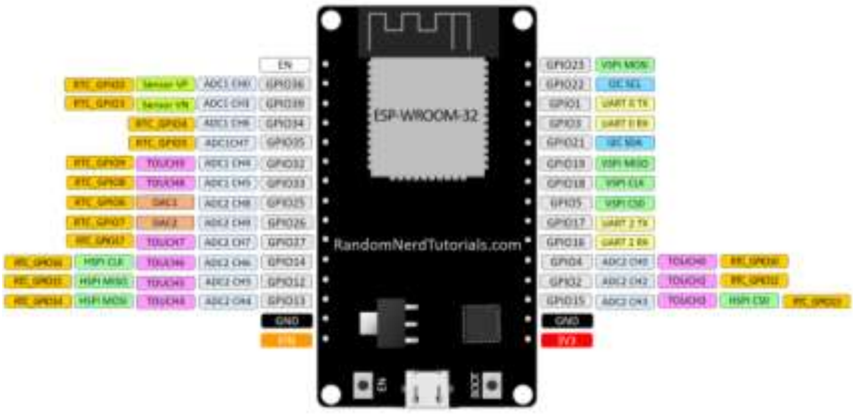


Figure 1 Pin assignment of ESP32

ESP_NOW

A wireless communication protocol is called ESP_NOW which allows ESP devices to communicate directly without connecting to a Wi-Fi network made by a router. The pairing among devices is required before their communication. After the initial pairing with the MAC address is accomplished, the connection is continuous and peer-to-peer communication is implemented without a handshake. ESP_NOW communication takes place in the low power consumption. It can send packets of messages (up to 250 bytes) between ESP boards. Any ESP board can function as either master or slave.

The following features are:

- It supports maximum 20 ESP boards.
- It sends data up to 250-bytes payload.
- Sending callback function can be set to inform the transmission whether data sending success or failure.
- Data sending range by ESP_NOW is more than that by Wi-Fi.

PuTTY

PuTTY is used as an open-source emulator, serial monitor, and network file transfer application. It is operated by connecting to the serial port. PuTTY is designed for Microsoft window. PuTTY configuration is expressed in Figure 2 [Tatham Simon, 2019].

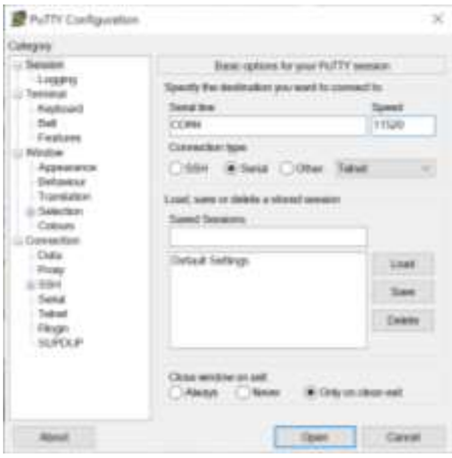


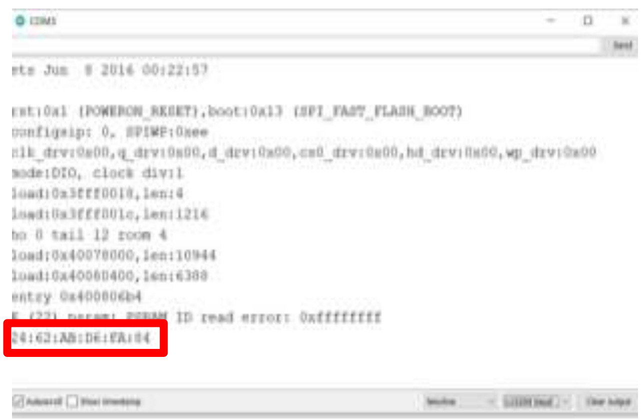
Figure 2 PuTTY configuration

MAC Address

MAC Address stands for Media Access Control address and it is a unique identifier hardware device on a network. MAC Addresses are identified by manufacturers. It can be changed optionally but it will return to its factory default MAC address when the device is reset. MAC Addresses composes of six groups of two hexadecimal digits, separated by colons like the following:

0xFF:0xFE:0XA6:0x00:0x0F:0xDD

The including "WiFi.macAddress()" in Arduino IDE can obtain MAC address. MAC address can be visualized on a serial monitor or PuTTY after uploading the sketch and pressing the reset button on ESP32. MAC address on serial monitor and PuTTY is illustrated in Figure 3 (a, b).



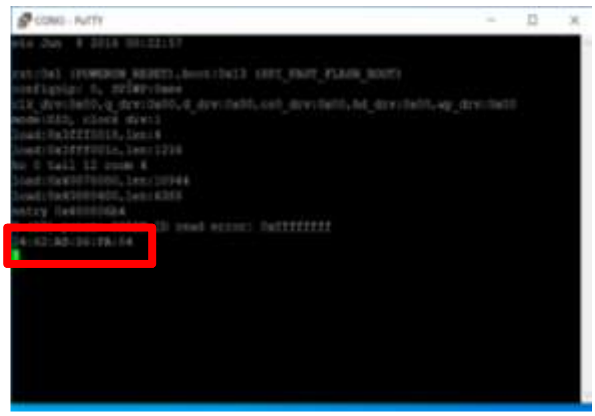


Figure 3(a) MAC on serial monitor **Figure 3(b)** MAC on PuTTY

DHT 11 Sensor

The DHT 11 sensor is used to measure temperature and humidity. It measures humidity based on the change of capacitance and determines the temperature using a thermistor which is built-in DHT 11. Additionally, it composes of an analog-to-digital converter. DHT 11 produces the 40-bit data output which contains the values of temperature and humidity.

Figure 4 A DHT 11 sensor

System Operation

The temperature and humidity are detected by master ESP32 and it is sent to the two slaves ESP32 boards using ESP_NOW communication. There are two main implementations: hardware and software preparation.

Hardware Preparation

There are two types of circuits. One is a master sender circuit and another circuit is a slave receiver circuit. The master sender circuit composes of ESP32 and a DHT 11 sensor. Data pin, Vcc, and ground of DHT11 are connected to D2, 3.3V, and ground of ESP32, respectively as shown in Figure 5. The schematic circuit of master sender and PCB layout are drawn by easyEDA software, and these are shown in Figure 6 (a, b).

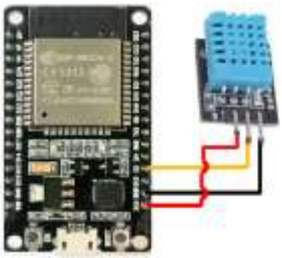


Figure 5 Circuit connection of DHT11 sensor and ESP32

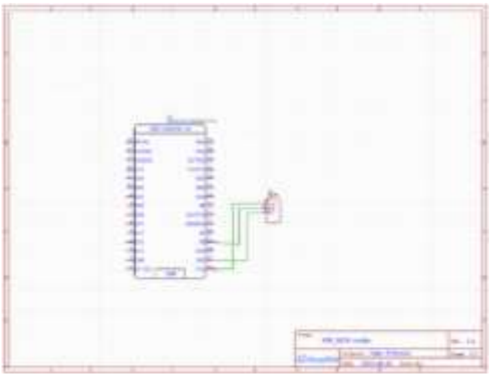


Figure 6(a) Schematic circuit of sender

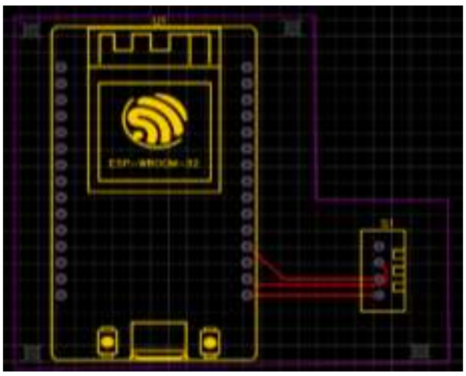


Figure 6(b) PCB layout of sender

The slave receiver circuit composes of ESP32 and I2C liquid crystal display (I2C LCD). Serial clock (SCL), serial data (SDA), Vcc, and ground of I2C LCD are connected to D22, D21, 3.3V, and ground of ESP32, respectively as shown in Figure 7. The schematic slave receiver circuit and PCB layout are drawn by easyEDA software, and these are shown in Figure 8(a, b).

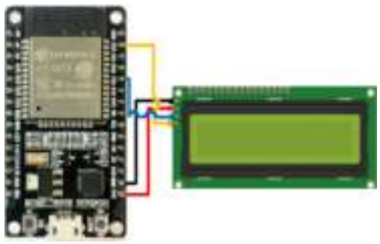


Figure 7 Circuit connection of I2C LCD and ESP32

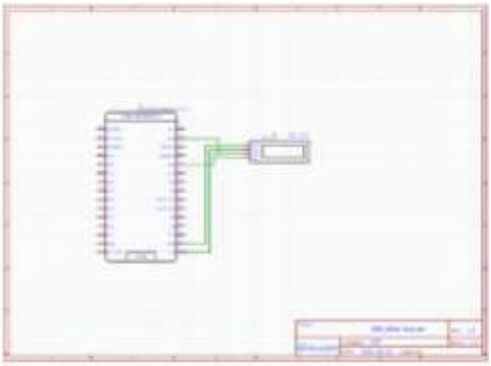


Figure 8(a) schematic circuit of receiver

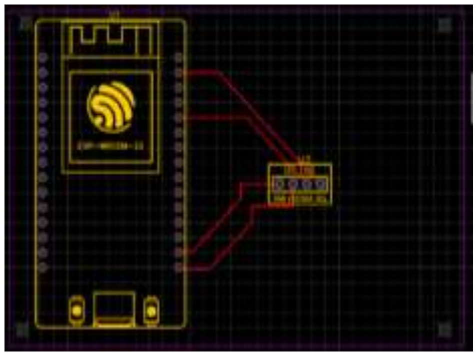


Figure 8(b) PCB layout of receiver

Software Preparation

The following syntaxes are mainly used in ESP_NOW programming.

First of all, the master ESP32 board and two slave ESP32 boards are set as the stationary mode by using the "WiFi mode (WIFI_STA)". ESP_NOW is required to initialize by calling "esp_now init()". The Wi-Fi STA mode should set before ESP_NOW is initialized. In this stage, the pair of master ESP32 board and two slave ESP32 boards will get all the information by using the calling"esp_now_deinit()". The paired ESP boards will complete after the calling "esp_now_deinit()". But the paired master ESP32 board and two slaves ESP boards are also needed that the paired devices with the broadcast MAC address be added by calling "esp_now_add_peer()" before sending data from the master ESP32 board. As soon as Wi-Fi will start, ESP_NOW data are sent to two slaves ESP32 boards. A master ESP board can send the maximum of 20 slaves ESP32 boards [Brandi Georg, 2021]. The "esp_now_send()" is included to send the ESP_NOW data and the "esp_now_register_send_cb()" is called to register the callback function in master ESP32 board. If the master ESP32 board with a pre-determined MAC address receives the sending data, it will return "ESP_NOW_SEND_SUCCESS" in the callback function. If two slaves ESP32 boards does not receive the sending data, the "ESP_NOW_SEND_FAIL" will be returned in the callback function into the master ESP32 board.

For the receiver section two slaves ESP32 boards, the "esp_now_register_recv_cb" is used to register for information on receiving a packet in the callback function.

The program flow of a master ESP32 board and that of two slave ESP32 boards are expressed in Figure 9.

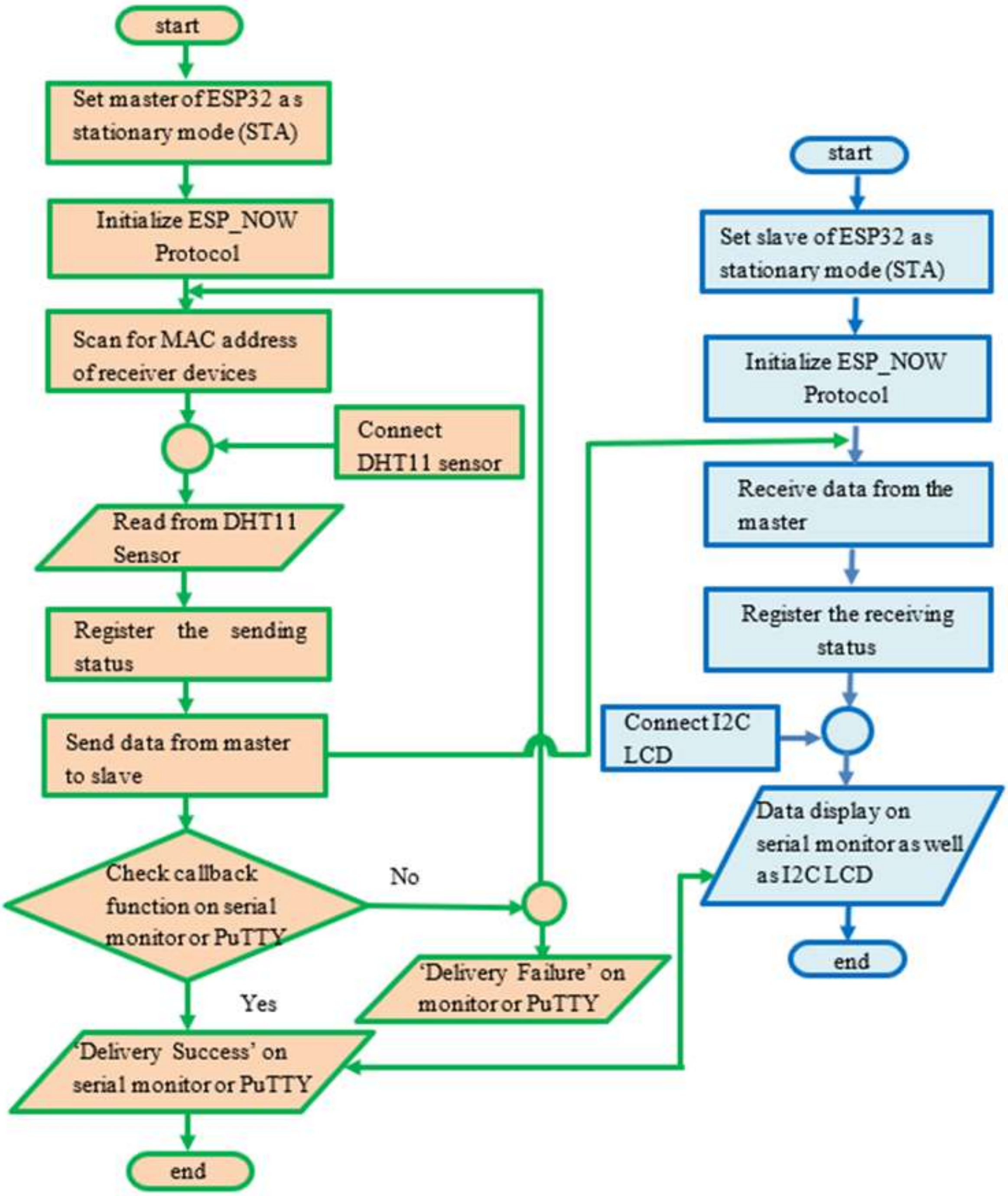


Figure 9 Flow chart of a master ESP32 board and two slave ESP32 boards

Results and Discussion

Results

Although data can be sent to 20 ESP receiver boards via ESP_NOW protocol, one ESP32 board functions as sender (master) while two ESP32 boards work as receivers (slaves). To send data to multiple ESP receiver boards, the unique MAC address of each ESP receiver board must be known. The MAC address in the ESP32 receiver boards can be determined by using software in which the "WiFi.macAddress()" is contained. Figure 10 is the MAC addresses of two receiver ESP32 boards.

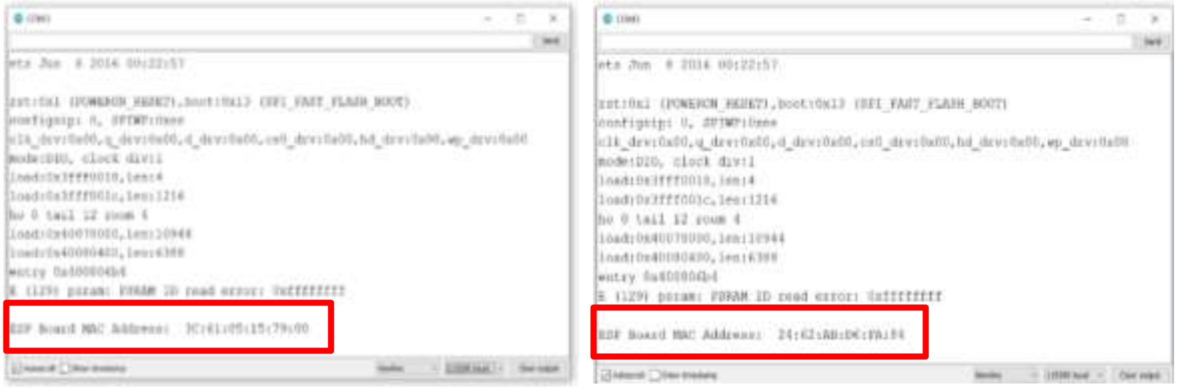


Figure 10 MAC address of two receiver ESP32 boards

As soon as the ESP32 sender (master) is powered on, it sends data to two ESP32 receiver boards whether they are powered or not. Figure 11 shows that the no power-condition of two ESP32 receiver boards. At that time, the ESP32 sender cannot receive an acknowledge message from two ESP32 receivers so that “delivery fail” for two ESP32 receivers is expressed on the serial monitor.

If one of two ESP32 receivers is activated, the ESP32 sender can receive the acknowledge message from one ESP32 receiver, and then “delivery success” and “delivery fail” are displayed as shown in Figure 12. If two ESP32 receivers are activated, the ESP32 sender gets acknowledged from each ESP32 receiver, and “delivery success” for each ESP32 receiver board is displayed on the ESP32 sender board’s serial monitor as shown in Figure 13. So, the ESP32 sender can visualize which ESP32 receiver receives data and which board didn’t.

In the receiver section, each ESP32 receiver is connected to I2C LCD so that data received are visualized in the serial monitor as well as the LCD screen. When the ESP32 receivers are powered on and they are in coverage range from the ESP32 sender, they receive data sent from the ESP32 sender. The data received in the serial monitor and PuTTY are shown in Figure 14 and Figure 15, respectively. The data received are displayed on I2C LCD as shown in Figure 16.

The communication range between sender and receiver is observed in Yangon University of Education campus and stable communication is observed up to 205 meters as shown in Figure 17.

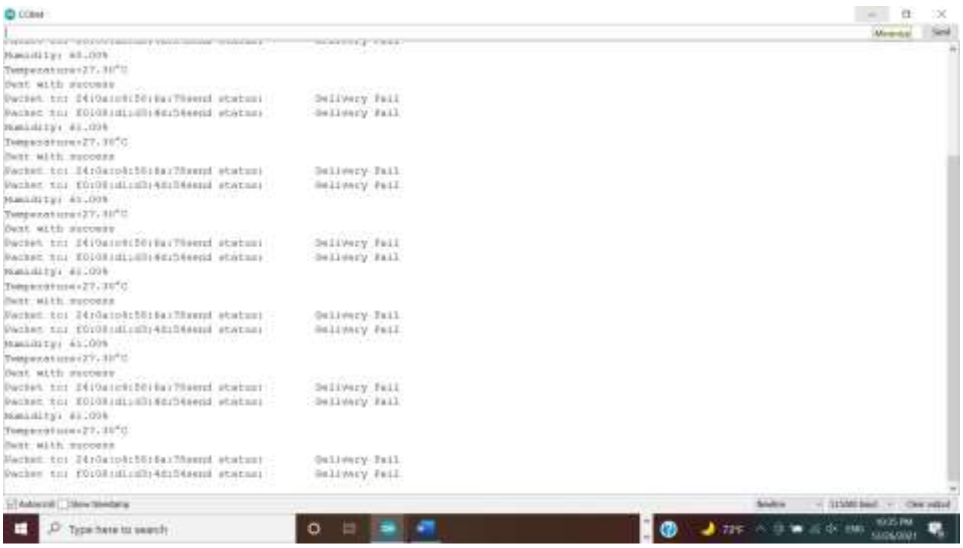


Figure 11 Sender’s monitor showing two receivers being inactive

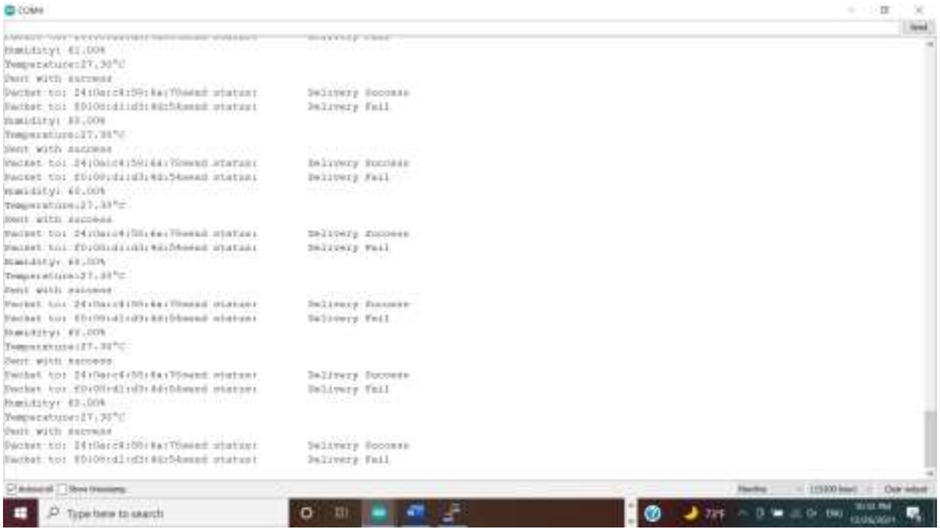


Figure 12 Sender's monitor showing one of two receivers being active

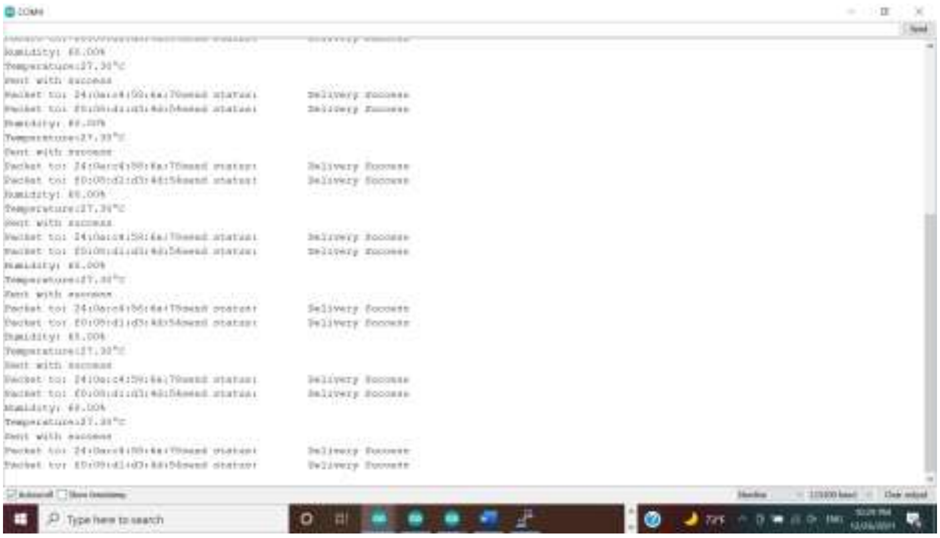


Figure 13 Sender's monitor showing two receivers being active

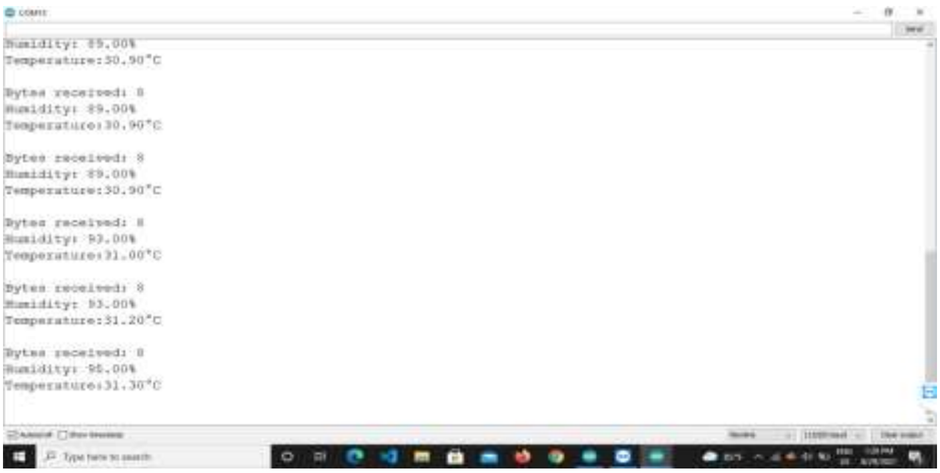


Figure 14 Data received on serial monitor

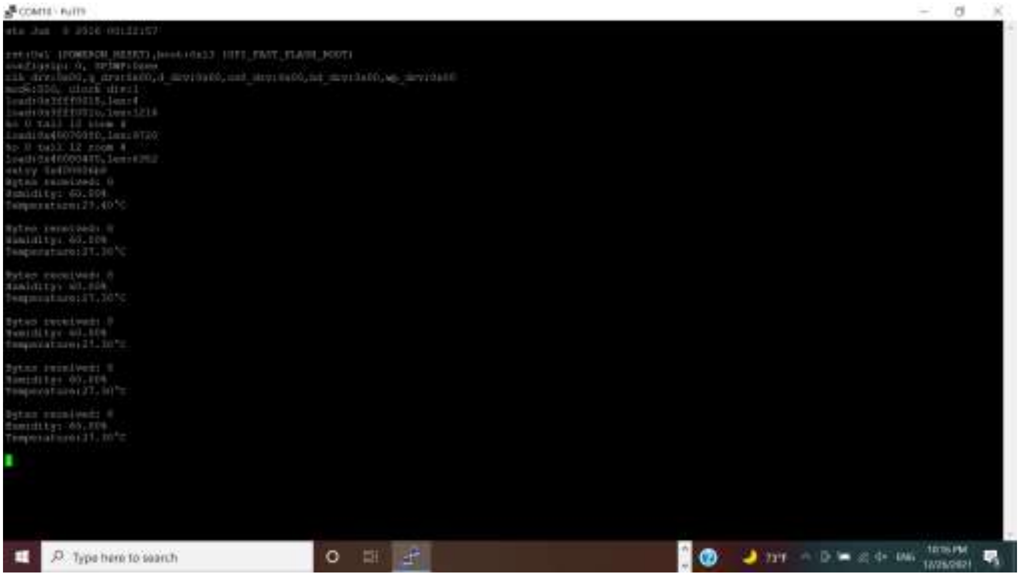


Figure 15 Data received on PuTTY

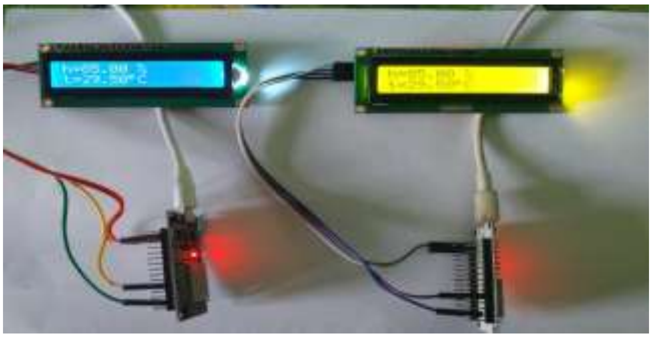


Figure 16 Data received on I2C LCD

Figure 17 Measurement of rage at stable ESP_NOW communication range

Discussion

In this research, only three ESP32 boards are used; one sender, two receivers. ESP8266, the WeMos board can be used instead of ESP32. But uploaded code must be slightly changed depending on the boards used in practice. In the determination of the stable range between sender and receiver, the on-board antenna of each ESP board should be pointed to each other. Two ESP boards can communicate and transfer data with a range of outdoor up to 220 meters (~ 722 feet). In this research, a stable communicated range is 205 meters (~ 673 feet) because the experimental site is not an open-air site.

Conclusion

ESP_NOW communication has many advantages over Wi-Fi communication but it has some weaknesses.

Advantages of ESP_NOW over Wi-Fi are:

(i) The maximum number of receiver ESP boards that can receive data from the sender in a stable communication range is 20.

(ii) By using ESP_NOW, ESP boards can communicate everywhere. Router or dynamic host configuration protocol (DHCP) is not needed in ESP_NOW communication.

(iii) Its essential feature is faster data transmission because it does not connect to Wi-Fi access points.

(iv) After pairing the master and slave devices, it communicates continuously. On the other hand, if one of the receiver ESP boards power off or reset incidentally, it will reconnect to the master ESP board automatically when it powers up again or it restarts.

Disadvantages of ESP_NOW are:

(i) It can transfer small data packets which are limited to 250 bytes.

(ii) Multiple ESP receivers are supported but it is limited to the maximum number of 20 receivers.

Acknowledgements

I would like to acknowledge Professor Dr Khin Khin Win (Retired), Head of Department of Physics, University of Yangon, and Professor Dr Yin Maung Maung, Head of Department of Physics, University of Yangon, for their kind permission to do. I sincerely thank all of Professors from the Department of Physics, University of Yangon for their kind encouragement to carry out this research.

I would like to thank Professor Dr Hla Toe, Head of Department of Physics, Yangon University of Education, for his advice, kind permission, and encouragement through the research period.

I deeply grateful to Professor Dr Ohn Than (Retired), Head of Department of Physics, Kyaukse University who supports the continuous guideline about the hardware and software of ESP32.

References

Bill, (2020) *ESp32-2*, Canada, DroneBot Workshop.
Brandi Georg, (2021) *ESP-IPF Programming Guide*, China, Espressif Systems (Shanghai) Co., Ltd.
Dotdash, (2018, *how-far-your-wifi-signal-reach*, New York, OpenWeb.
Robocraze, (2019) *Tutorial-Getting-Started-with-ESP-Now*, India, TIF Labs Pvt. Ltd.
Santos Rui, (2020) *ESP_Now-Two-Way-Communication-ESP32,* America, Random Nerd Tutorials Lab, No Starch Press.
Santos Rui, (2020) *ESP32 troublesshooting guide*, America, Random Nerd Tutorials Lab, No Starch Press.
Santos Rui, (2020) *Learn ESP32 with Arduino IDE,* America, Random Nerd Tutorial Lab, 2nd ed., No Starch Press.
Tatham Simon, (2019) *PuTTY*, United States, Wikipedia.