URSI - RCRS 2022, IIT (Indore), India, 1 - 4 December, 2022

IPWC: Integrative Peer to Peer Wireless Communication of MEMS based Remote Temperature Sensor using ESP NOW and IoT Networking

Aarushi Sharma* (1), Ayan Karmakar (2), and Sachin Sharma (2)
(1) Department of ECE, UIET, Panjab University, Chandigarh, India, 160014
(2) Department of MTAD/EFTG, Semiconductor Laboratory, Punjab, India, 160071

Abstract

One of the most cutting-edge technological advancements of the modern era is wireless communication. Long-range and short-range services are both largely owing to wireless technology. There are numerous wireless communication techniques available today like Bluetooth, WiFi, etc. But one such peer to peer network based protocol is ESP-NOW, which can communicate between two ESP microcontrollers without WiFi connectivity. ESP-NOW is a wireless communication protocol that can transmit data at comparatively quick transfer rates over great distances and is needed for the Internet of Things in order to exchange the necessary data swiftly and in real-time. In addition, the technology that is developing rapidly at this time is remote sensing. Remote Microelectromechanical Systems (MEMS) based temperature sensors are extensively used in various industrial applications, biomedical engineering, etc. Present remote temperature sensor uses ESP-NOW for successful trans-reception of the temperature values at low latency and low power consumption. In the experiments conducted, the ESP-NOW gateway successfully receives all the packets of data with a success rate of 100% which is at a distance of up to 20 meters from the transmitter or the initiator.

Keywords— ESP-NOW, ESP32, Remote Sensing, Internet of Things, Wireless Communication, Protocols

1. Introduction

In recent years there has been innovation in the area of microelectronics for meeting the requirements of rapidly developing industrial smart solutions. Wireless sensors have motivated development in the domains of embedded systems, remote monitoring, the Internet of Things (IOT) and smart automation just to name a few [1]. The ecosystem of wireless sensors typically associate smart sensors to respective IoT modules for collecting and monitoring the data of real-time physical parameters.

The performance improvements in the MEMS based sensors and the integration of mechanical along with microcircuits on a singular chip have led to fabrication of low cost high precision actuators and sensors [2]. This has pushed forward for establishing wireless sensor networks as a communication standard in industrial settings. Wireless instruments offer the benefits, such as mobility/flexibility, reduced maintenance cost (eg. wear and tear of cables), lightweight, uncomplicated reconfiguration etc. [3].

Wireless Sensor Networks can be developed with various communication protocols. These encompass some techniques such as Wireless Local Area Network (WLAN) which include WiFi and Bluetooth, Wide-Area Wireless Communication like Long Range Radio (LoRa) and short range technologies such as RFID. However, lately there has been significant development in peer to peer (p2p) communication strategies. ESP-NOW [4] is such a new protocol where communication is possible without the use of WiFi. It is safe, energy efficient and does not require a handshake after pairing to perform p2p communication. Moreover, it supports encrypted communication with the ability to send data in a short duration of time upto 250 bits/payload.

In modern industrial equipment measuring of environment variables and physical factors is of utmost necessity. Collecting such valuable data helps in easy remote monitoring and predictive analysis of the process. In this study a technique for wireless transmission of temperature data has been done via p2p communication protocol using ESP-NOW has been proposed. This technique presents a low cost flexible alternative to traditional monitoring techniques.

The latter content of this paper is structured as follows. Related Works are contained in Section II. Section III briefly explains the Proposed System Design and Implementation. Section IV contains the results. In Section V, the paper is concluded along with the future work.

2. Related Works

Mostafa Ibrahim Labib et al [5] have proposed the use of low-energy technologies like ESP-NOW. They have suggested an effective networking approach for establishing and operating wireless sensor networks. Their work proposes "An adaptive spider-mesh topology" that is a low power consumption mesh technique. Their experimental results demonstrate that the ESP-NOW protocol, a proprietary protocol that uses twice as much power as the Bluetooth low energy protocol, is an effective bi-directional communication protocol for creating the suggested solution.

Koushik M S et al [6] have explored the area of agriculture by implementing a wireless sensor network based data logger with ESP-NOW protocol. The goal of their system was to precisely monitor and gather data that can then be used by applying any type of AI or ML algorithm to the obtained data. Their research was able to gather data in a dynamic and flexible manner by utilising a data logger system built on a Wireless Sensor Network (WSN) and the ESP-NOW protocol.

Guijie Wang et al [7] have worked upon smart temperature sensors and temperature sensor systems. Transistors, thermocouples, and thermopiles are the most widely used temperature-sensing elements for smart sensors and MEMS because they can be implemented using IC technology. Pt resistors, thermistors, and IR sensors are crucial components of the system. The characteristics of four different types of sensing elements were also compared. Thermopiles are used to monitor temperature differences and are ideal for thermal sensors, which work by first converting signals into thermal quantities and then thermal quantities into electrical quantities to measure physical quantities. Typically, such sensors also monitor a reference temperature, perhaps using a bipolar transistor.

Dania Eridani et al [8] have proposed the comparison of various wireless communication protocols. According to their research, the best protocols are those that are capable of achieving the necessary requirements, which can be seen by key performance indicators like maximum range, power usage, transmission speed, reaction time, and its capacity to send signals and penetrate different barriers. Wi-Fi generally provides strong interconnection support and overall performance. Low transmission latency is supported over maximum distances using ESP-NOW. Bluetooth is excellent for a variety of devices made for long-term applications because it consumes less power.

R. Rizal Isnanto et al [9] have proposed a framework to address the design of an ESP-NOW Protocol-Based Robot to Manage Agricultural Soil Conditions. In their study, an IoT-based robot was created to maintain various types of soil using the best possible conditions. This robot can automatically irrigate the plant while keeping an eye on the soil's health. A protocol was needed for the Internet of Things in order to exchange the necessary data swiftly and in real-time; one such protocol was ESP-NOW. So, they used a communication protocol called ESP-NOW which can transmit data at comparatively quick transfer rates over great distances.

3. Design and Implementation

A. Hardware System Design

The proposed system design and architecture for ESP NOW p2p MEMS based temperature sensor is depicted in Figure 1. This approach is an indication towards the amalgamation of Wireless Sensor network with peer to peer communication protocol. The basic hardware block diagram depicts the connections and data trans-reception for this system.

The microcontroller used in this system is ESP32 [12] that was developed and created in 2016 by the Shanghai-based company called Espressif System. ESP32 is a System on Chip (SoC) that contains built-in Wi-Fi and several other widely used peripherals, including SPI, UART, and I2C, which are used by most sensors to communicate.

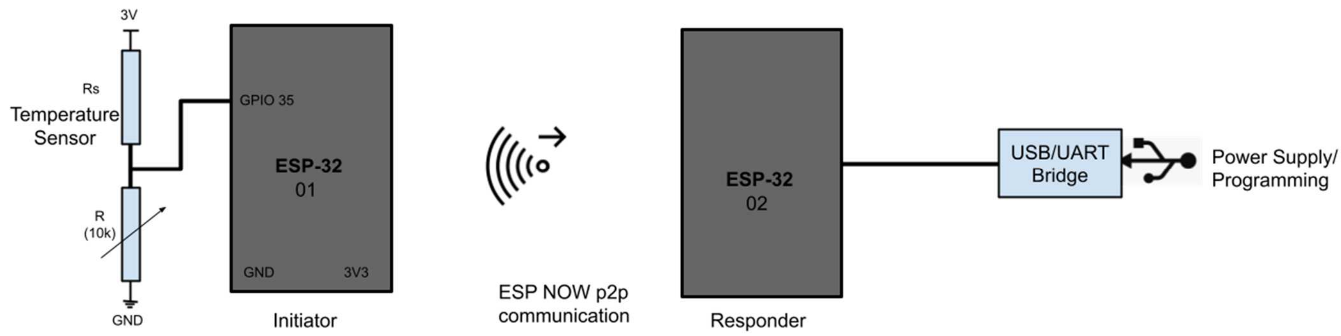

Figure 1. System Hardware Block Diagram

Firstly, a MEMS based temperature sensor is integrated with a variable resistance and the output signal of this is connected to the Analog to Digital (ADC) pin of the microcontroller, ESP32. This ADC pin has a reference voltage of 1100mV and has a 12-bit resolution and can sample up to 255 cycles per sample. The data after the conversion from analog to digital is taken as an input to the framework which is subject to calibration.

The microcontroller which is the initiator or the transmitter is labelled as ESP-01 for the sake of ease of reference. The data received from the temperature sensor is stored and formatted in this microcontroller and further transmitted via ESP-NOW to the responder or the receiver which is labelled as ESP-02. ESP-NOW communication protocol is employed for p2p wireless communication between the two microcontroller boards for the trans-reception of sensor data. For a hierarchical network, the overall protocol structure does not require a router or joining state. Instead, it facilitates low-power p2p communication between several devices. This approach is faster to deploy and more energy-efficient than standard Wi-Fi [10]. All devices on the ESP-NOW network can communicate via broadcast, unicast, and multicast with data speeds of at least 1 Mbps [11].

The received data is then transmitted via serial communication to get the output values of temperature on the screen of the computer connected.

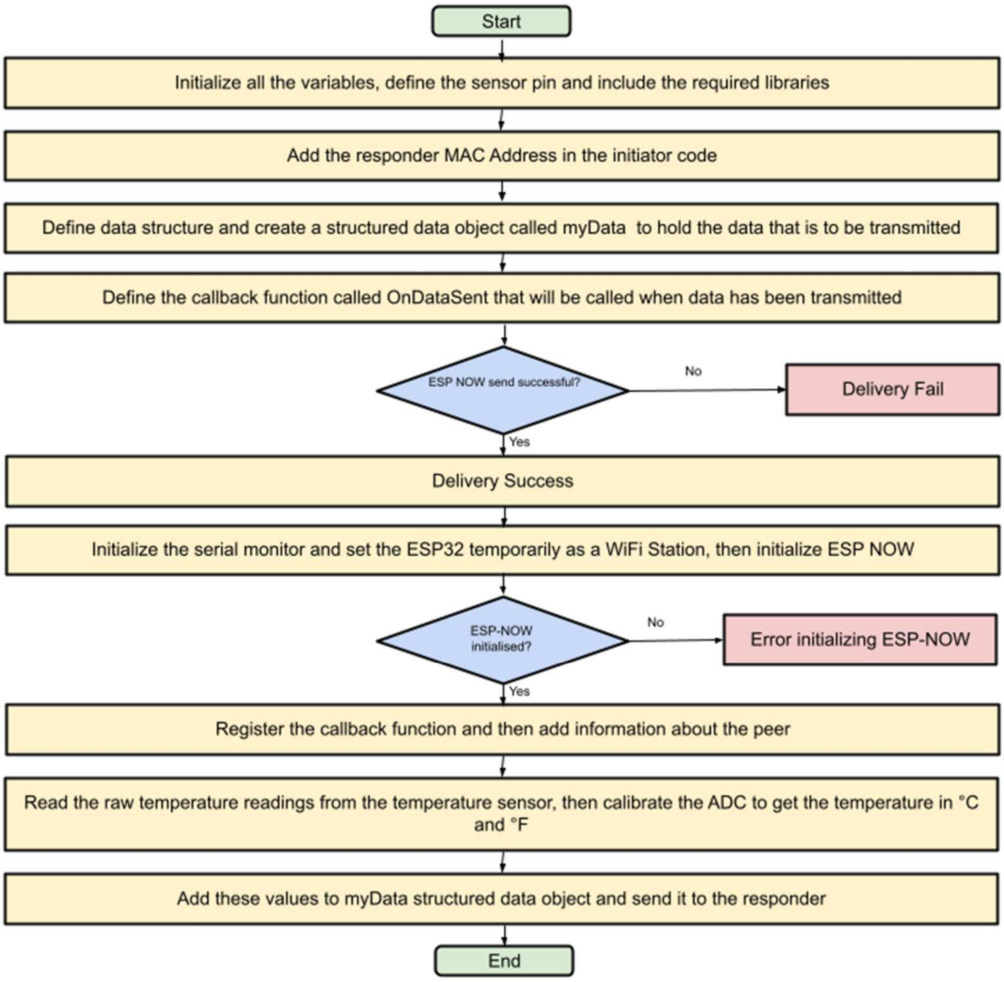

Figure 2. Flowchart of the initiator/ transmitter

B. Software Development

The software used for programming the microcontrollers used in the system is Arduino IDE. Since ESP-NOW is an OSI transport layer protocol [13] [14]. So, after making a few adjustments to the physical layer to allow for higher throughput, we then deploy an application layer on top of the transport layer. The additional layer aids the system in handling data streaming via the network by implementing flow control. The functional process flow chart of ESP-01 which is the initiator or the transmitter is depicted in Figure 2. The ESP-NOW library is added along with the WiFi library. The MAC address of the responder is added to this algorithm to set up the p2p communication between the two microcontrollers.

On the initiator side, a data structure is defined and a structured data object called *myData* is created. The callback function named *OnDataSent* will be called when data has been transmitted. This callback function is implemented to obtain the sending status of the temperature data with the aid of Arduino IDE.

The processing of the ESP32 will vary depending on the status. It starts the recording of the temperature and sends temperature data packet by packet to the responder constantly if the peer network is set up else it will display initialising ESP-NOW. The raw sensor data after the conversion from analog to digital is then subjected to calibration to get the appropriate values of the temperature. This calibrated temperature data is saved in the structured object called *myData* which is called to send the data to the responder or the receiver.

On the other hand, as a receiver, the ESP32 begins to process the sender's data received from ESP-01 and the callback function called in this process is implemented to get the receiving status of the temperature data. The callback function *onDataRecv* does the role of receiving the data and the received data is further sent to the serial monitor and plotter to get the values and graph of the temperature in degree celsius and fahrenheit displayed on the screen of the computer. Figure 3 depicts the detailed functional process of the responder microcontroller i.e. ESP-02.

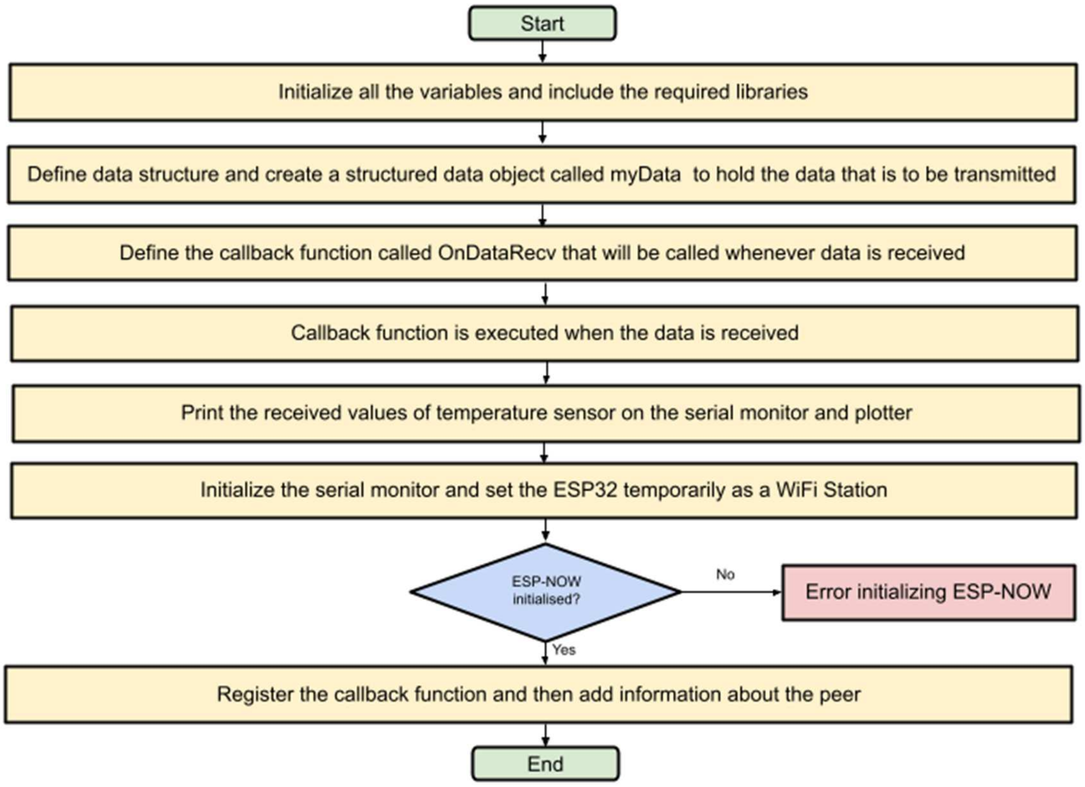

Figure 3. Flowchart of the responder/receiver

4. Results

The results for the ESP-NOW p2p wireless MEMS based temperature sensor were captured on the serial monitor and serial plotter of Arduino IDE, which is an open source development software for programming the microcontrollers. The experiment's research showed that they had a significant chance of being used in the real world as the wireless trans-reception of the sensor data showed low latency and high data rate of up to 250 bits/payload and can measure the temperature within the range of -55 °C to + 125 °C. The data collected following the experiment is shown in the Table 1.

The input data from the MEMS based temperature sensor was transmitted from ESP-01 to the responder ESP-02 with a delay of 200ms which was added to the program for the sake of observing the received data on the serial monitor with ease. Figure 4 depicts the values of the temperature in both °C and °F, received from the sensor with the timestamp.

Table 1. Temperature values received at the Responder Node

Timestamp	Temperature in °C	Temperature in °F
15:42:45.947	51.70	125.06
15:42:46.556	38.90	102.02
15:42:46.883	43.40	110.12
15:42:47.773	36.20	97.16
15:42:49.551	44.90	112.82
15:42:49.879	43.40	110.12

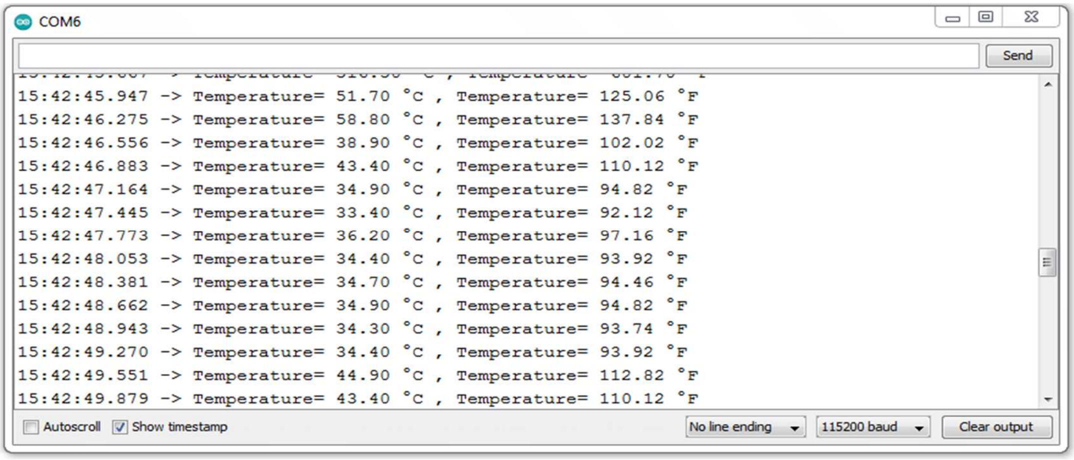

Figure 4. Serial Monitor depicting the temperature

The sensor was set up in different environments and was subjected to various changes in the outer temperature to get the results. Figure 5 demonstrates the fluctuation in the values of the temperature. The spikes in the graph depicts the sudden rise or fall of the temperature of the outer environment in which the sensor was based. The pink curve depicts the temperature variation in fahrenheit whereas the red curve demonstrates the temperature variation in celsius. The range of communication between the two microcontrollers via ESP-NOW was also observed. The system uses the onboard antennas for the communication.

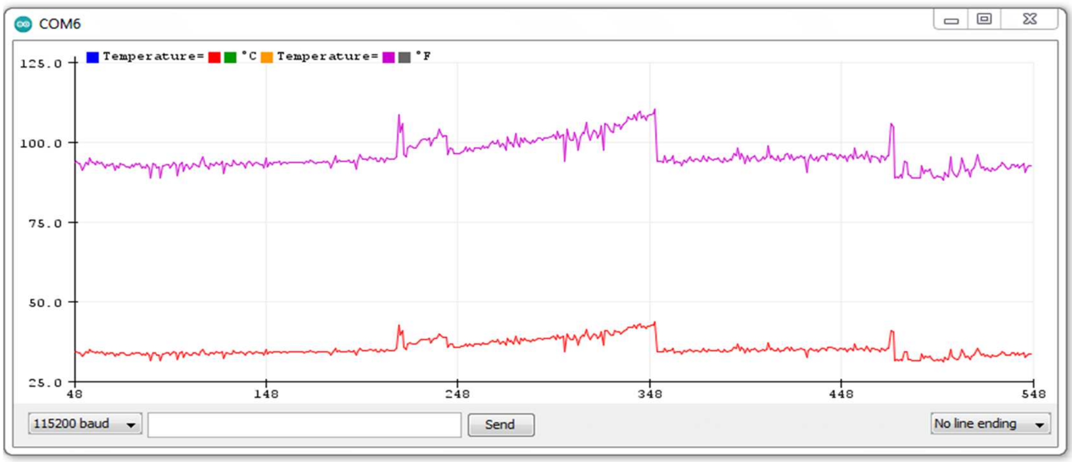

Figure 5. Serial Plotter depicting the temperature

It was observed that when the two ESP32s had their on-board antennas pointing towards each other there was lesser packet loss and the range of communication was also more as compared to when the antennas were not facing each other. Table 2 shows the distance and the corresponding packet trans-reception. This system is not just effective, but also quite practical and has an easy set up for the communication to take place.

Table 2. Range of ESP-NOW

Distance (in metres)	% Packet trans-reception
1	100
2	100
5	100
10	100
20	100
50	97.65

5. CONCLUSION AND FUTURE WORK

A. Conclusion

With the results obtained, the development and implementation of new and the altered ESP-NOW network have demonstrated the capability of establishing a communication network that is low-cost, low-latency, low-power, yet effective enough for sending and receiving the sensor data. Since there is no requirement for a wired connection in ESP-NOW, the system's form of communication for transmitting the packets of data offers a way to flexibly position computer nodes across the work area. However, there are still several problems with the system, such as packet loss due to distance, an unsynchronized sleep cycle, and the need to integrate all the components onto a single printed circuit board (PCB).

B. Future Work

There are a few things that must be performed properly because the temperature sensor system is still being developed. Some of the ideas include integrating all components onto a single PCB to further save costs, adjusting the sleep cycle to prevent packet loss, and adding an external antenna to increase range and support MIMO. Additionally, if we include a codec IC for compressing and decompressing, the latency may be reduced somewhat more. The range should expand even further as a result of this innovation, which should speed up device connection. For more user-friendly purposes we can also implement MQTT broker [15] with Arduino IDE and ESP32. We can also implement this whole system on platforms like Arduino IoT Cloud [16] or Blynk [17] that provide easy access to the data through cloud storage.

REFERENCES

1. Labib, Mostafa, et al. "Networking solutions for connecting bluetooth low energy devices-a comparison." MATEC web of conferences. Vol. 292. EDP Sciences, 2019.

2. A. Vogel, D. Binz, S.Heim, 1. Schnieder, U. Bilitewski: Development of an Automated Microbial Sensor System; Biosensors&Bioelecrronics I4 (1999) (pp.187 - 193)

3. Scheible, G., J. Schutz, and C. Apneseth. "Novel wireless power supply system for wireless communication devices in industrial automation systems." IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02. Vol. 2. IEEE, 2002.

4. Espressif, "ESP-NOW User Guide," Espressif, 2016

5. Labib MI, ElGazzar M, Ghalwash A, AbdulKader SN. An efficient networking solution for extending and controlling wireless sensor networks using low-energy technologies. PeerJ Comput Sci. 2021 Nov 24;7:e780. doi: 10.7717/peerj-cs.780. PMID: 34901432; PMCID: PMC8627223.

6. K. M. S, S. M, L. R, A. S and S. Setty, "Design and Development of Wireless Sensor Network based data logger with ESP-NOW protocol," 2021 6th International Conference for Convergence in Technology (I2CT), 2021, pp. 1-5, doi: 10.1109/I2CT51068.2021.9417914.

7. Meijer, Gerard & Wang, Guijie & Heidary, A.. (2018). Smart temperature sensors and temperature sensor systems. 10.1016/B978-0-08-102055-5.00003-6.

8. D. Eridani, A. F. Rochim and F. N. Cesara, "Comparative Performance Study of ESP-NOW, Wi-Fi, Bluetooth Protocols based on Range, Transmission Speed, Latency, Energy Usage and Barrier Resistance," 2021 International Seminar on Application for Technology of Information and Communication (iSemantic), 2021, pp. 322-328, doi: 10.1109/iSemantic52711.2021.9573246.

9. R. Rizal Isnanto, Y. Eko Windarto, J. Imago Dei Gloriawan and F. Noerdiyan Cesara, "Design of a Robot to Control Agricultural Soil Conditions using ESP-NOW Protocol," 2020 Fifth International Conference on Informatics and Computing (ICIC), 2020, pp. 1-6, doi: 10.1109/ICIC50835.2020.9288575.

10. T. N. Hoang, S. -T. Van and B. D. Nguyen, "ESP-NOW Based Decentralized Low Cost Voice Communication Systems For Buildings," 2019 International Symposium on Electrical and Electronics Engineering (ISEE), 2019, pp. 108-112, doi: 10.1109/ISEE2.2019.8921062.

11. ESPRESSIF, “ESP-IDF Programming Guide v4.0-dev-119,” pp. 2-4, 2019.

12. ESPRESSIF, “ESP32 Datasheet v3.9,” March 2022, pp 8-12.

13. Neil Kolban “ Kolban's Book on the ESP32”, April 2019, pp.68–73.

14. Dogan Ibrahim and Ahmet Ibrahim, “The Official ESP32 Book”, November 2017, pp.90-108

15. Patrick Lapointe “EspMQTTClient” https://github.com/plapointe6/EspMQTTClient [Accessed: 1-Feb-2022].

16. Liam Aljundi, “Arduino Cloud IoT Cheat Sheet.” https://docs.arduino.cc/arduino-cloud/getting-started/technical-reference [Accessed: 10-Aug-2022].

17. Blynk, “Getting Started” https://docs.blynk.io/en/ [Accessed: 12-Jul-2022].