2022 13th International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-9939-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICTC55196.2022.9952832

Peer to Peer Communication for the Internet of Things Using ESP32 Microcontroller for Indoor Environments

Muhammad Syukor Abdul
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia
syukor@graduate.utm.my

Suriani Mohd Sam
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia
suriani.kl@utm.my

Norliza Mohamed
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia
norlizam.kl@utm.my

Noor Hafizah Hassan
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia
noorhafizah.kl@utm.my

Azizul Azizan
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia
azizulazizan@utm.my

Yusnaidi Md Yusof
Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia
Kuala Lumpur, Malaysia
yusnaidi.kl@utm.my

***Abstract*— This paper presents an alternative peer to peer communication for the Internet of Things (IoT) using ESP32 microcontroller for indoor environments. This peer-to-peer communication is called ESP-NOW. The development is done using Arduino-ESP32 in the Visual Studio Code with Platform extension. Based on the experiment, the environmental sensing data can still be transmitted using the ESP-NOW protocol even with the lowest transmission power level. Thus ESP-NOW communication can be used in indoor environments but with a limited number of devices paired up together and depends on the surrounding environments confining in the indoor environments.**

Keywords—Internet of Things (IoT); ESP 32; ESP-NOW; indoor; Visual Studio Code; PlatformIO

I. INTRODUCTION

Internet of Things (IoT) will provide technological advances for modernization to improve productivity and better scope coverage of the domain. Internet protocols, smart sensors, communication technologies and analytics are factors enabling IoT. IoT will boost people to automate processes, predict situations and improve existing daunting tasks even in real-time for daily use. Their application spreads out and advances rapidly in smart homes, smart cities, transportation, manufacturing, healthcare, agriculture, and water management and energy sector. People worldwide are trying to adopt IoT in their production environment to improve their yields, productivities and better management [1].

One of the major components in deploying IoT is the microcontroller (MCU) which act as the brain of the IoT devices. Various MCUs are available in the market worldwide. One of them is the ESP32 microcontroller designed and manufactured in 2016 by Espressif Systems [2]. This chip is a 32-bits microcontroller and provides communication capabilities such as WiFi and Bluetooth. This low-cost chip is available in various kind of development boards around the world such as ESP32 DevKitC [3], DO IT ESP32 DevKit [4] and TTGO ESP32 board [5].

Among all communication technologies for IoT, WiFi and Bluetooth connectivity are most commonly used as these communication technologies are cheaper and easier to buy on the local market. WiFi and Bluetooth connectivity are easy to be integrated with mobile devices such as smartphone and tablets. While LoRa technology [6] is gaining momentum because of the radio spectrum used in unlicensed spectrum and can be easily connected to the global communication network.

Despite all the communication currently available for the MCU, Espressif Systems generously introduce their implementation of peer to peer communication which enable ESP32 MCU to do communicate with each other within the IEEE 802.11 protocol. This communication enables direct peer to peer communication and low power among ESP32 microcontrollers. This method is power-efficient and convenient as communication can be done without the need for a router.

This paper begins with the introduction of the ESP32 microcontroller, its wireless communication capabilities and limits. Then, we shall look into peer to peer communication capability called ESP-NOW that is available in the ESP32 microcontroller. We also will investigate the capabilities of this peer to peer communication for indoor environments. This work is based only on conferences and journal publication from IEEE digital publication.

II. ESP32 MICROCONTROLLER

ESP32 microcontroller is gaining attraction to researchers around the world to experiment with the ESP32 microcontroller.

Until the year 2019, there are 54 research publications produced by the IEEE publisher and these publications only appear starting from the year 2017. Fig. 1. illustrates the total publication published yearly by the IEEE publisher.

The researchers are experimenting with ESP32 for many kinds of IoT not limited to smart home and building automation [7][8], agriculture [9][10], image recognition [11], healthcare [12][13][14][15], environmental monitoring [16][17] and animal monitoring [18].

The ESP32 is suitable for IoT devices due to the factors of cheap, performance, various form-factors, and support real-time application. Espressif Systems and its counterparts have designed and manufactured numerous ESP32 models [19] worldwide, such as LOLIN D32 [20] from WEMOS, China, SparkFun Thing Plus [21] from SparkFun Electronics, USA, HUZZAH32 [22] from Adafruit, USA, Nano32Lite [23] from Gravitech, Thailand and LoPy4 [24] from Pycom, United Kingdom. Therefore, ESP32 microcontrollers can meet various IoT applications' operational requirements [25]. Fig. 2. shows various kinds of ESP32 development boards available in the market. Espressif ESP32 microcontrollers come with a dual-core Tensilica Xtensa LX6 microprocessor, Wi-Fi and dual-mode Bluetooth functions [26]. According to [26] ESP32 is a dual-core system with two central processing units (CPU) of Tensilica Xtensa LX6 and all memory and peripherals are in the CPU busses. This ESP32 was released to the market by Espressif Systems in 2016. The ESP32's memory space is 448 KB ROM, 520 KB SRAM, and two RTC 8 KB memory. While external ESP32 memory supports flash memory of up to 16 MB [26]. Table I shows the specifications of the ESP32 (ESP-WROOM-32).

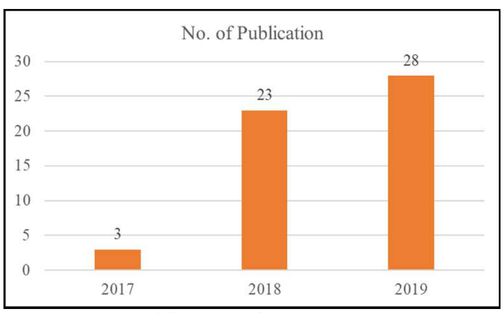

Fig. 1. IEEE publications of the ESP32 microcontroller

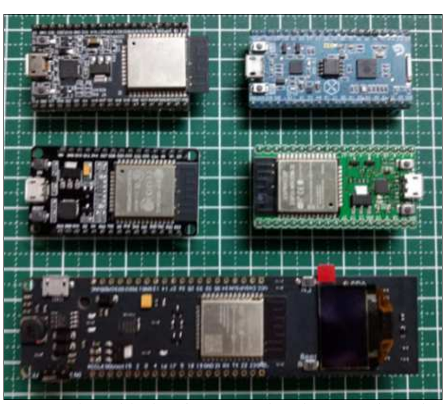

Fig. 2. Various kinds of ESP32 development boards

TABLE I. THE ESP32 SPECIFICATION

ESP32 Microcontroller (ESP-WROOM-32)	
Details:	
CPU	Tensilica Xtensa LX6 32 bit Dual-Core at 160/240 MHz
SRAM	520KB
FLASH	4MB (max. 16MB)
Voltage	2.2V to 3.6V
Operating Current	80 mA average
Programmable	C, C++, Lua, MicroPython, JavaScript
Real time OS	FreeRTOS, Zephyr
Open source	Yes
Connectivity:	
Wi-Fi	802.11 b/g/n
Bluetooth	4.2 BR/EDR + BLE
UART	3
I/O:	
GPIO	32
SPI	4
I2C	2
PWM	8
ADC	18 (12-bit)
DAC	2 (8-bit)
Dimension:	
Size	25.5 x 18.0 x 2.8 mm

The specification of ESP32 microchip isa well designed for IoT devices. The ESP32 also come with peripherals for Pulse Width Modulation (PWM), SD Card I/O (SDIO), Analogue to Digital Converter (ADC), Digital to Analogue Converter (DAC), Inter-integrated Circuit (I2C), Serial Peripheral Interface (SPI), Inter-IC Sound (I2S), Controller Area Network (CAN), Universal Asynchronous Receiver/Transmitter (UART) and General Purposes Input Output (GPIO). ESP32 provides Wi-Fi and Bluetooth communication for the developer to use. Fig. 3. illustrates the functional block diagram of the ESP32 chip. ESP32 do includes security features such as secure boot and flash encryption. ESP32 also come with hardware-based random number generator (RNG) which suitable for cryptography functions. The cryptographic functions such as Rivest–Shamir–Adleman (RSA), Secure Hash Algorithm (SHA), and Elliptic-curve cryptography (ECC) also implemented into this chip.

In addition to this superior chip, Espressif Systems provides the developer with the development tools for using ESP32 microcontrollers called Espressif IoT Development Framework (ESP-IDF) and the codes are open to everyone. ESP-IDF, the platform for the development of IoT software for ESP32, has enabled many projects and platforms from third parties to be based on it. There is also Arduino based library called Arduino Core for ESP32 (Arduino-ESP32) for the developer to do the development for ESP32 microcontrollers using Arduino Integrated Development Environment (IDE). Both ESP-IDF and Arduino-ESP32 libraries are available for the public in the GitHub platform.

ESP32 also provides several IEEE 802.11 communications protocol for the developer to utilize it in the IoT environment. These protocols include IEEE 802.11b, IEEE 802.11g, and IEEE 802.11n. Espressif Systems also includes in the ESP32 special proprietary patented custom long-range mode called IEEE 802.11LR and thus enable the ESP32 to communicate approximately 1km within the line of sight. ESP32 provides up to 12 levels of wireless transmission power during wireless communication. These transmission (TX) power level are from -1dBm up to 19.5dBm.

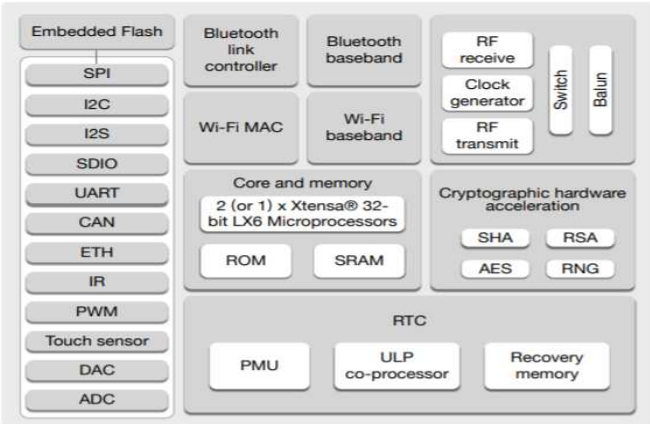

Fig. 3. ESP32 functional block diagram

Table II shows the transmission power level of the ESP32 and the programming codes that are available for the developer to control the wireless transmission power level of the ESP32 microcontroller. ESP32 also provides peer to peer communication called ESP-NOW as an alternative for ESP32 microcontroller to directly communicate among ESP32 microcontrollers.

III. ESP-NOW Peer-To-Peer Communication

In this section, we will explore the peer to peer communication capability available in the ESP32 microcontroller. ESP-NOW is a peer to peer communication available in the ESP32 microcontroller. This communication is based proprietary communication protocol developed by Espressif Systems and does not needs a router to do the communication among the ESP32 microcontroller. ESP-NOW protocol was released in 2015 initially supported for Espressif ESP8266 microcontroller and then been designed to support the ESP32 microcontroller [7]. ESP-NOW relying on the IEEE802.11 Vendor-specific Action Frame. Table III depicts the Vendor-specific Action Frame utilized by ESP-NOW to encapsulate and to transmit the data from one ESP32 to another ESP32 microcontroller. The Category Code field is set to the 127 indicating the vendor-specific category. While Organization Code is set to unique identifier (0x18fe34). The 4 bytes Random Values is set as to prevent relay attacks during the transmission.

In the Vendor Specific Content, there are 7 fields allocated to this frame. Element ID is set to 221 indicates Vendor Specific Content frame. The field Length is the total bytes of Organization Identifier, Type, Version and Body. The Body field can accommodate up to 250 bytes of data. Table IV shows the fields of the Vendor Specific Content.

ESP-NOW can send up to 250 bytes payload of data during each transmission for the ESP32 microcontroller. Before the transmission can be made, the target receiver ESP32 microcontroller must be paired first with the sender ESP32 microcontroller. ESP-NOW communication use the MAC Address of the ESP32 as a unique ID for the peer to peer communication. Up to 20 ESP32 nodes can be added as peers for each ESP32 if the communication is not encrypted. But only up to 10 ESP32 nodes can be added as peers if the communication is encrypted. ESP-NOW utilizing an enhanced data cryptographic encapsulation mechanism designed for data confidentiality and based upon the Counter Mode with CBC-MAC (CCM mode) of the Advanced Encryption Standard (AES) standard, CCMP, as described in the IEEE802.11-2012 for security [7]. ESP-NOW can be programmed to the ESP32 microcontroller by using Application Programming Interface (API) functions provided by the ESP-IDF framework or by using the Arduino-ESP32 framework. Table V shows some examples of API used to enable ESP-NOW peer to peer communication protocol for the ESP32 microcontroller.

With ESP-NOW, the communication can be done directly from ESP32 to other ESP32 without the requirement of the router. This feature enable IoT devices communicate directly among each other and able to do faster communication without relaying to the mediator such as router for sending and receiving data even though the length of the data is limited to 250 bytes maximum per transmission.

IV. Experimenting IOT Communications using ESP-NOW in The Indoor Environment

ESP32 microcontrollers can be used as the IoT devices providing ESP-NOW peer to peer communication for sending and receiving the sensors data. The researchers used ESP32 and ESP-NOW communication to transmit voice to another ESP32 as low cost voice communication in the building that are low latency and low power but enough for the talk burst in the building even though the communications may suffer packet loss due to long distance between devices. Fig. 4. shows the hardware of setup of this study including ESP32 development board.

Researchers [7] utilizing ESP-NOW protocol to construct multi-hop wireless network for sensor nodes with the combination of ZigBee network in low-power consumption manners. The system was designed to support multiprotocol gateway developed by them. The system implementation are shown as in the Fig. 5. We are also doing experiment with the ESP-NOW protocol for environmental sensing in the local residence house. This local house is a single storey terrace house with the area of 70’x20’ sf. The floorplan of this house are as illustrates as in the Fig. 6. There is a single line of sight with maximum length of 14m can be achieved in this house.

TABLE II. The ESP32 WiFi TX Power Level

TX Power Level	**TX Power**	**ESP-IDF Constant Code**
11	19.5 dBm	WIFI_POWER_19_5dBm
10	19 dBm	WIFI_POWER_19dBm
9	18.5 dBm	WIFI_POWER_18_5dBm
8	17 dBm	WIFI_POWER_17dBm
7	15 dBm	WIFI_POWER_15dBm
6	13 dBm	WIFI_POWER_13dBm
5	11 dBm	WIFI_POWER_11dBm
4	8.5 dBm	WIFI_POWER_8_5dBm
3	7 dBm	WIFI_POWER_7dBm
2	5 dBm	WIFI_POWER_5dBm
1	2 dBm	WIFI_POWER_2dBm
0	-1 dBm	WIFI_POWER_MINUS_1dBm

TABLE III. ESP-NOW IEEE802.11 VENDOR-SPECIFIC ACTION FRAME

MAC Header	Category Code	Organization Identifier	Random Values	Vendor Specific Content	FCS
24 bytes	1 byte	3 bytes	4 bytes	7 – 255 bytes	4 bytes

TABLE IV. ESP-NOW IEEE802.11 VENDOR-SPECIFICCONTENT FIELDS

Element ID	Length	Organization Identifier	Type	Version	Body
1 byte	1 byte	3 bytes	1 byte	1 byte	0 – 250 bytes

TABLE V. ESP32 ESP-NOW API FUNCTIONS

Function	Description
esp_now_init()	Initializes the ESP-NOW driver.
esp_now_add_peer()	adds a device (specified by its MAC address) to the paired device list.
esp_now_send()	Sends payload data to a specific paired device.
esp_now_register_send_cb()	Registers a send callback function. After ESP-NOW data have been sent, the send callback function is called. The sending status can be checked for success or failure.
esp_now_register_recv_cb()	Registers a receive callback function. When receiving ESP-NOW data, the receive callback function is called.

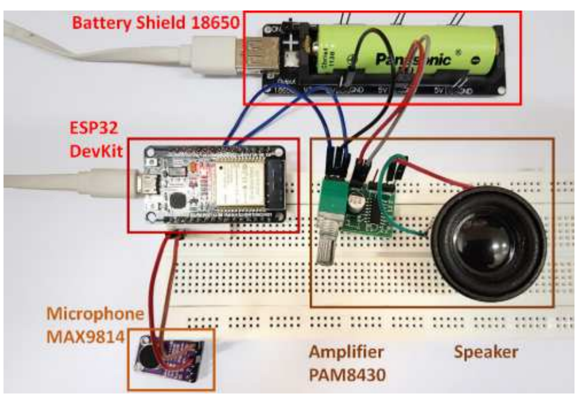

Fig. 4. ESP32 development board as used in the paper

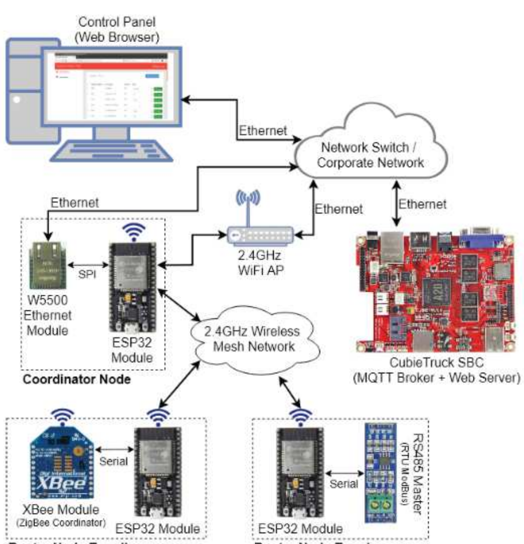

Fig. 5. System implementation of multi-hop wireless network utilizing ESP-NOW protocol [7]

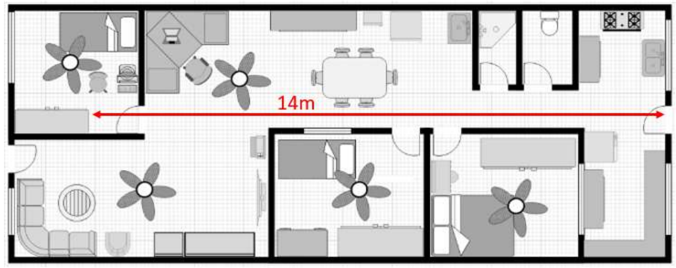

Fig. 6. Floorplan of a local house with 14m of the line of sight

There are two types of nodes used in this setup, one is a base station (BaseNode) using an ESP32 development board for receiving sensors data and another one is a sensor node (SensorNode) also using ESP32 development board for sending sensors data. BaseNode functioned as a collector of all sensors reading sent by the SensorNode and display the in the serial output. The BaseNode is connected to the laptop using a USB connection. The data received by BaseNode then are displayed back by using the Putty, a serial terminal software. While SensorNode acts as humidity and temperature sensing node and sends the data to the BaseNode. Both BaseNode and SensorNode images are shown in Fig. 7.

For the experiment based on line of sight (LOS), the BaseNode is statically positioned at one side and in LOS with SensorNode at the distance of 2m, 4m, 6m, 8m, 10m, 12m and 14m. Then the Received Signal Strength Indicator (RSSI) of the communication is measured and sent back to the BaseNode from the SensorNode including with the temperature and humidity readings. While the transmission power of the SensorNode is limited to the lowest power level 0 (-1dBm) and used only an onboard antenna.

The development of the firmware for the BaseNode and SensorNode is done using the Arduino-ESP32 framework in the Visual Studio Code with PlatformIO extension. The serial outputs for the BaseNode and SensorNode are shown in Fig. 8. and Fig. 9. The RSSI measurement in the LOS setup is shown in Fig. 10. As we can see, the RSSI is decreasing as the distance is getting far away from the BaseNode. But within the maximum LOS 14m in an indoor environment, the SensorNode can still successfully send the data to the BaseNode.

For the experiment of non-LOS, Fig. 11. shows the location of the BaseNode and SensorNode. There are six SensorNode are randomly positioned around the house to ensure that the location of the SensorNode is not in LOS position from the BaseNode and the transmission power of all SensorNode are limited to the lowest power level 0 (-1dBm) and used only onboard antenna.

In this non-LOS setup, all SensorNode still can send the data to the BaseNode even the wireless communication is obstructed by the concrete wall and household appliances. The farther SensorNode (S6) have RSSI value of -80 and successfully sent all data to BaseNode. Fig. 12. shows the measurement of RSSI according to each SensorNode.

All the setup are confined in the local residence house as we would like to observe the ESP-NOW protocol as alternative wireless communication for indoor IoT communication.

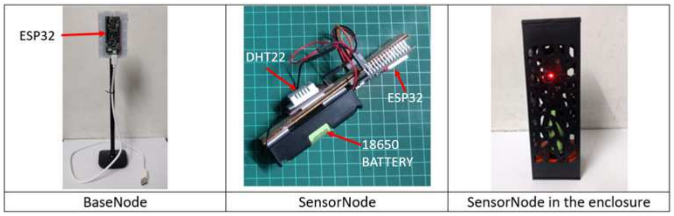

Fig. 7. BaseNode and SensorNode

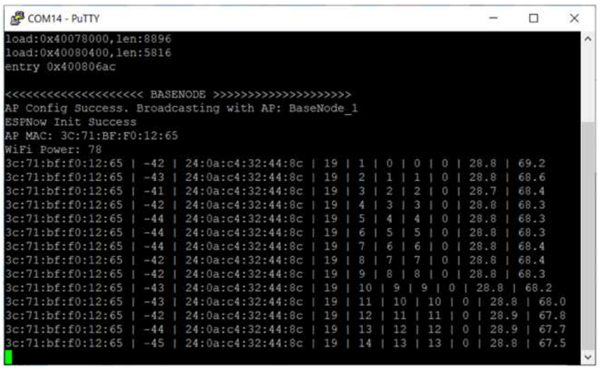

Fig. 8. BaseNode serial output

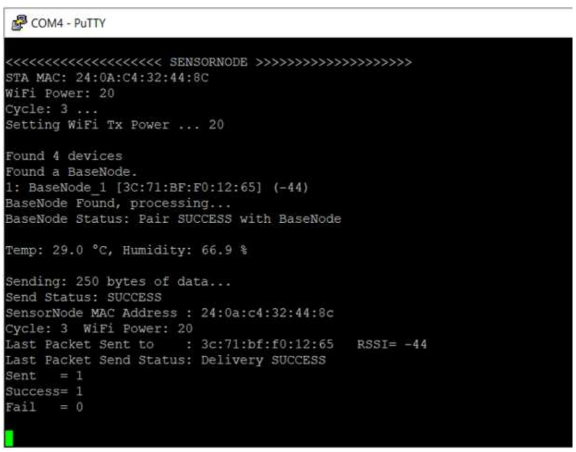

Fig. 9. SensorNode serial output

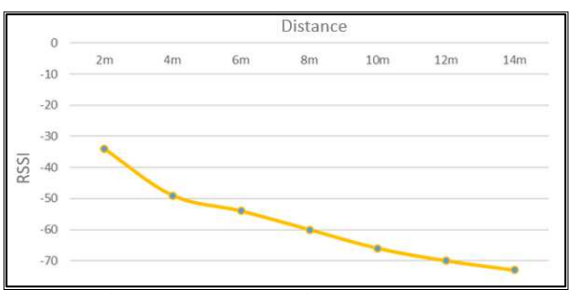

Fig. 10. Measurement of RSSI in LOS setup

V. Discussion

ESP32 microcontroller provides the developer wireless communication capabilities of Bluetooth and Wi-Fi and proprietary ESP-NOW communication protocol. ESP-NOW protocol is a kind of peer to peer communication that enable packet of data to be sent directly to other ESP32 without the needs of the router as in the normal Wi-Fi communication.

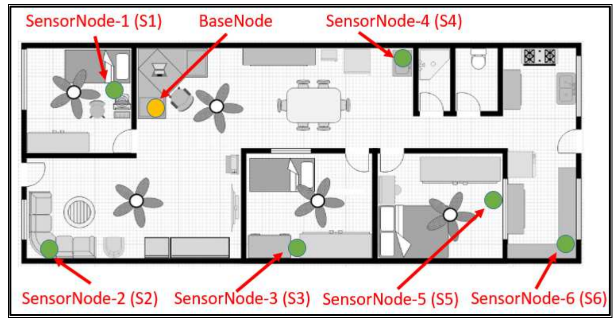

Fig. 11. BaseNode and SensorNode location in non-LOS position

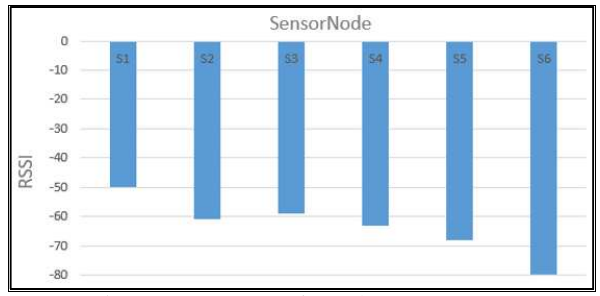

Fig. 12. Measurement of RSSI in non-LOS setup

The experiment of ESP-NOW protocol in the indoor environment in the local residence house was conducted to observed the signal reception within the residential environment that is obstructed by the concrete wall and household appliances. A total length of 14m in line of sight was obtained during the experiment. The SensorNode transmits the data to the BaseNode within various distances from the BaseNode. Even though the TX power level of the SensorNode is reduced to the lowest power level 0, -1 dBm, the wireless communication can still be done using the ESP-NOW protocol. The packet data have been crafted to the maximum size of 250 bytes as this is the maximum size of payload that ESP-NOW can send during the packet transmission.

In the LOS setup, the RSSI value is depend on the distance of the SensorNode from the BaseNode. The RSSI value gradually decreasing as the distance of SensorNode become farther from the BaseNode.

Another experiment was done using randomly positioned of SensorNode location in the local residence house also been conducted. The temperature and humidity data can still be sent from SensorNode to BaseNode even though the wireless signals are obstructed by this indoor environment. This non LOS setup is done to study either the ESP-NOW protocol can be used in the local residence house. As the result, the ESP-NOW protocol can be used as an alternative communication for the IoT devices in the local residence house.

The main factor influencing the wireless transmission is the environment surrounding the SensorNode and the BaseNode. The wall, partitions and household appliances can obstruct or reflected the wireless signal from SensorNode and the BaseNode. There are several pros and cons of the ESP-NOW observed during the experiment conducted. Table VI summarized the pros and cons of the ESP-NOW.

TABLE VI. ESP32 ESP-NOW API FUNCTIONS

PROS	CONS
ESP-NOW communication does not need a router the established the connection.	A total of 20 ESP32 can be paired simultaneously and up to 10 ESP32 can be paired simultaneously if the ESP-NOW communication is encrypted.
ESP32 can easily add peers for ESP-NOW communication by just adding the MAC Address of the target ESP32 to the peer list.	Normal Wi-Fi cannot be used if ESP32 is using ESP-NOW communication.
Communication of ESP-NOW is less power consumption as stated in the ESP-NOW manual.	ESP-NOW only communicate in the same wireless channel for all peers.
ESP-NOW does not require connection reestablishment as in the standard Wi-Fi protocol thus making ESP-NOW protocol faster.	Only up to 250 bytes of data can be transmitted during the ESP-NOW communication.

VI. CONCLUSION

This paper introduces the ESP-NOW, peer to peer communication for the Internet of Things using ESP32 microcontroller. It is possible to transmit the data using ESP-NOW in the indoor location such as in residential environment and in the building with the limits of wireless signal penetration through the wall or partitions, wireless signal reflection and electrical appliances that can interfere with the wireless signal. ESP-NOW protocol can be used for indoor IoT communication such as for the smart home or building that are confined to a certain distance from the main controller. But the devices are limited to up to 20 devices using ESP32 microcontrollers.

ACKNOWLEDGMENT

The authors fully acknowledge the Ministry of Education (MOE) and Universiti Teknologi Malaysia for the research grant (Q.K130000.3556.05G32) that help in funding the research works. Special thanks to all the reviewers for their valuable comments and constructive suggestions.

REFERENCES

[1] P. V. Paul and R. Saraswathi, “The Internet of Things — A comprehensive survey,” in 2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), 2017, pp. 421–426.

[2] Espressif Systems, “ESP32 Overview | Espressif Systems,” 2019. [Online]. Available: https://www.espressif.com/en/products/hardware/esp32/overview. [Accessed: 25-Apr-2019].

[3] “ESP32-DevKitC | Espressif Systems.” [Online]. Available: https://www.espressif.com/en/products/hardware/esp32-devkitc/overview. [Accessed: 08-Jan-2020].

[4] “DOIT ESP32 DEVKIT V1 Board (Wi-Fi and Bluetooth) - Maker Advisor.” [Online]. Available: https://makeradvisor.com/tools/esp32-dev-board-wi-fi-bluetooth/. [Accessed: 08-Jan-2020].

[5] Cytron.IO, “TTGO T-Display ESP32 1.14 Display Module-Presolder Header.” [Online]. Available: https://my.cytron.io/p-ttgo-t-display-esp32-1.14-display-module-presolder-header?r=1&gclid=EAIaIQobChMIpdKE063z5gIVBQ4rCh0MJQiKEAYYASABEgKohPD_BwE. [Accessed: 08-Jan-2020].

[6] LoRa Alliance, “Home page | LoRa AllianceTM,” 2019. [Online]. Available: https://lora-alliance.org/. [Accessed: 25-Apr-2019].

[7] K. Khanchuea and R. Siripokarpirom, “A Multi-Protocol IoT Gateway and WiFi/BLE Sensor Nodes for Smart Home and Building Automation: Design and Implementation,” 10th Int. Conf. Inf. Commun. Technol. Embed. Syst. IC-ICTES 2019 - Proc., pp. 1–6, 2019.

[8] R. K. Kodali, B. Yatish Krishna Yogi, G. N. Sharan Sai, and J. Honey Domma, “Implementation of Home Automation Using CoAP,” IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2018-Octob, no. October, pp. 1214–1218, 2019.

[9] R. K. Kodali and A. Valdas, “MQTT Based Monitoring System for Urban Farmers Using ESP32 and Raspberry Pi,” Proc. 2nd Int. Conf. Green Comput. Internet Things, ICGCIoT 2018, pp. 395–398, 2018.

[10] A. Gloria, C. Dionisio, G. Simoes, P. Sebastiao, and N. Souto, “WSN Application for Sustainable Water Management in Irrigation Systems,” IEEE 5th World Forum Internet Things, WF-IoT 2019 - Conf. Proc., pp. 833–836, 2019.

[11] V. Kushnir, B. Koman, and V. Yuzevych, “IoT Image Recognition System Implementation for Blind Peoples Using esp32, Mobile Phone and Convolutional Neural Network,” pp. 183–187, 2019.

[12] D. G. Kristiani, T. Triwiyanto, P. C. Nugraha, B. G. Irianto, Syaifudin, and D. Titisari, “The Measuring of Vital Signs Using Internet of Things Technology (Heart Rate and Respiration),” Proc. - 2019 Int. Semin. Appl. Technol. Inf. Commun. Ind. 4.0 Retrosp. Prospect. Challenges, iSemantic 2019, pp. 417–422, 2019.

[13] A. Škraba, A. Kolozvari, D. Kofjac, R. Stojanovic, E. Semenkin, and V. Stanovov, “Prototype of Group Heart Rate Monitoring with ESP32,” 2019 8th Mediterr. Conf. Embed. Comput. MECO 2019 - Proc., no. June, pp. 12–15, 2019.

[14] S. Mann et al., “EncephalogamesTM (brain/mind games): Inclusive health and wellbeing for people of all abilities,” 2019 IEEE Games, Entertain. Media Conf. GEM 2019, 2019.

[15] J. Limpanadusadee, P. Kesawattana, T. Wongsawat, and D. Wongsawang, “Eldtec: Improvement on wearable sensor for elderly fall detection,” Proceeding 2018 7th ICT Int. Student Proj. Conf. ICT-ISPC 2018, 2018.

[16] M. Vasiljević-Toskić, V. Rajs, J. Bajić, M. Božić, A. Hašović, and M. Šović, “Wireless Weather Station with No Moving Parts,” EUROCON 2019 - 18th Int. Conf. Smart Technol., pp. 2–6, 2019.

[17] L. Boppana, K. Lalasa, S. Vandana, and R. K. Kodali, “Mongoose OS based Air Quality Monitoring System,” pp. 1245–1250, 2019.

[18] A. A. O et al., “Monitoring and control system approach for native threatened species,” 2019 Int. Conf. Inf. Syst. Softw. Technol., pp. 85–91, 2019.

[19] P. Rai and A. A. E. S. P. Uno, “ESP32 Based Smart Surveillance System,” 2019 2nd Int. Conf. Comput. Math. Eng. Technol., pp. 1–3, 2019.

[20] WEMOS.CC, “LOLIN D32 - WEMOS.CC,” 2019. [Online]. Available: https://www.wemos.cc/. [Accessed: 25-Apr-2019].

[21] SparkFun Electronics, “SparkFun Thing Plus - ESP32 WROOM - WRL-14689 - SparkFun Electronics,” 2019. [Online]. Available: https://www.sparkfun.com/products/14689. [Accessed: 25-Apr-2019].

[22] Adafruit, “Adafruit HUZZAH32 – ESP32 Feather Board,” 2019. [Online]. Available: https://www.adafruit.com/product/3405. [Accessed: 25-Apr-2019].

[23] Gravitech Thai, “Gravitech Thai (Thailand) Electronic Experimental Solutions,” 2019. [Online]. Available: https://www.gravitechthai.com/. [Accessed: 25-Apr-2019].

[24] Pycom, “LoPy4,” 2019. [Online]. Available: https://pycom.io/product/lopy4/. [Accessed: 25-Apr-2019].

[25] Espressif Systems, “Espressif Achieves the 100-Million Target for IoT Chip Shipments | Espressif Systems,” 2018. [Online]. Available: https://www.espressif.com/en/media_overview/news/espressif-achieves-100-million-target-iot-chip-shipments. [Accessed: 25-Apr-2019].

[26] A. Maier, A. Sharp, and Y. Vagapov, “Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things,” 2017 Internet Technol. Appl. ITA 2017 - Proc. 7th Int. Conf., pp. 143–148, 2017.