Design of Internet of Things (IoT) Trainer Kit with Multi Communication

Era Madona[a,1*], Efrizon [a,2], Anggara Nasution[a,3], Rara Yetrisia Putri1[a,4], Laxsmy Devy [a,5]

[a] *Department of Electrical Engineering, Politeknik Negeri Padang, Indonesia*
[a,1] *emadona38@gmail.com*
* *corresponding author*

A B S T R A C T S

Industry 4.0, or the Fourth Industrial Revolution, is a recent technological development that has significantly changed the industrial production process. One of the keys to this change is the Internet of Things (IoT), where the role of humans is likely to diminish and be replaced by machines. Aligning the workforce with IoT is important, but creating a qualified workforce is a challenge. IoT learning modules are needed to help students understand IoT concepts. The purpose of this research is to design an Internet of Things kit module with Radio communication with LoRa SX1278 Ra-02. The research stages start from literature study, hardware design, software design and overall testing. Module testing is done by sending DHT11 sensor data using LoRa SX1278 Ra-02 and monitored in Thingspeak. From the test results that have been carried out, it can be concluded that sending using LoRa (Long Range) technology is influenced by distance and obstacles. At a distance of 40 to 230 metres, communication between the Lora Transmitter and Lora Receiver is successful, which shows that LoRa technology is able to overcome communication distances in that range. However, at a distance of more than 230 metres to 300 metres, data transmission can still be done by the Lora Transmitter, but the data cannot be received by the Lora Receiver, indicating a bottleneck in communication.

K E Y W O R D S

Article history

Received July 17, 2023
Revised September 19, 2023
Accepted December 23, 2023

Keywords

IoT, LoRa Ra-02 1278x, microcontroller, data communication, module kit, learning aid

1. INTRODUCTION

The Fourth Industrial Revolution, known as Industry 4.0, has introduced recent trends in technological development that have significantly affected production processes in various industrial sectors. This era is characterised by the Internet of Things (IoT), where the role of humans tends to decrease and machines take over some functions[1], [2]. This potential development of IoT demands a workforce that has special competencies in this field. The challenges faced in creating a workforce that meets the required qualifications are increasing along with competition in skills and knowledge

Internet of Things (IoT) is a rapidly growing technology that has wide applications in various fields of science. In the healthcare industry, IoT is used to monitor body temperature and room temperature [3-5]. The application of IoT also extends to industrial sectors such as manufacturing, oil and gas, agriculture, and more. IoT helps improve efficiency and productivity in the industrial sector by allowing real-time data collection and sophisticated data analysis. This contributes to improved machine performance and reduced operational costs[6]-[9].

In the context of learning, the application of IoT has great potential to help students understand the basic concepts of data communication and IoT. However, students often face difficulties in obtaining contextual experiences in the learning process. Therefore, it is necessary to develop a learning module that can facilitate students in understanding the basic concepts of IoT and data communication [10], [11]. This learning module can provide practical and contextual experience in IoT and data communication, so that students can better understand the concept and apply it in real-world situations.

Several studies have been conducted to develop practical and effective IoT learning modules[1]. Research conducted by Mahendra et al. gave positive results in the use of learning modules as learning media. However, to improve the quality of IoT learning, this study aims to design an Internet of Things kit module with communication. One of the significant additions in this development is the use of LORA 1278x RA-02 module, which enables efficient remote communication in the learning module. This addition will expand the module's ability to teach IoT concepts with the use of communication technology relevant to current industry demands.

2. RESEARCH METHODOLOGY

The research was conducted in the Microcontroller Laboratory of the Electrical Engineering Department of Padang State Polytechnic. The methods used in this research are; literature study, hardware design, software design, hardware testing, software and analysis.
The research stages can be seen as in Figure 1.

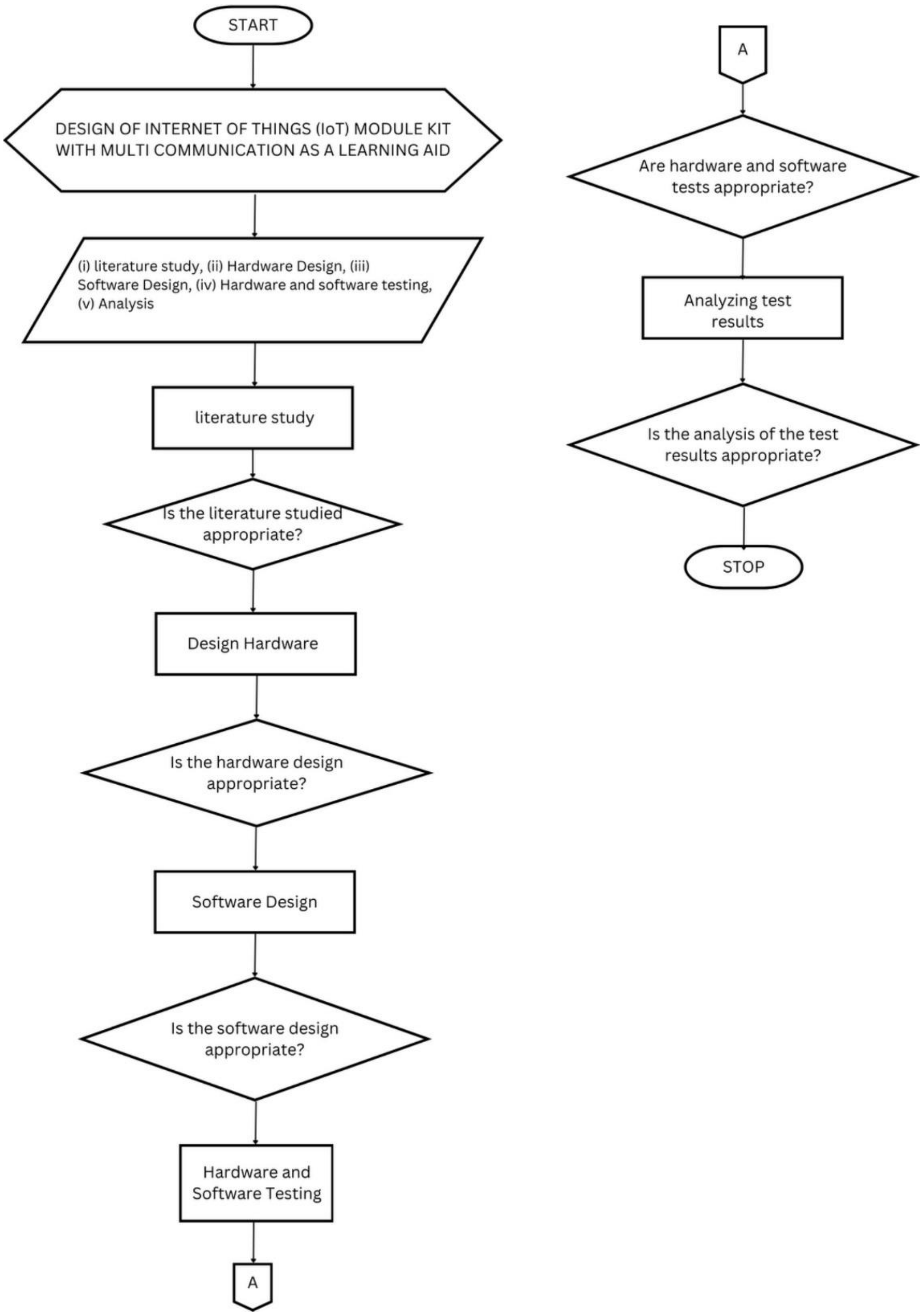


FIGURE 1. Research method for designing module kits

2.1 System Design

Practical activities can help students improve the quality of learning and make students understand more and have skills. Based on this analysis, this module can help students in carrying out practical activities for learning IoT and data communication. In this module, the component for communication learning used is Lora 1278x Ra-02, and the microcontroller that supports IoT learning used in the module is NodeMCU microcontroller. The block diagram of the practicum module can be seen in Figure 2.

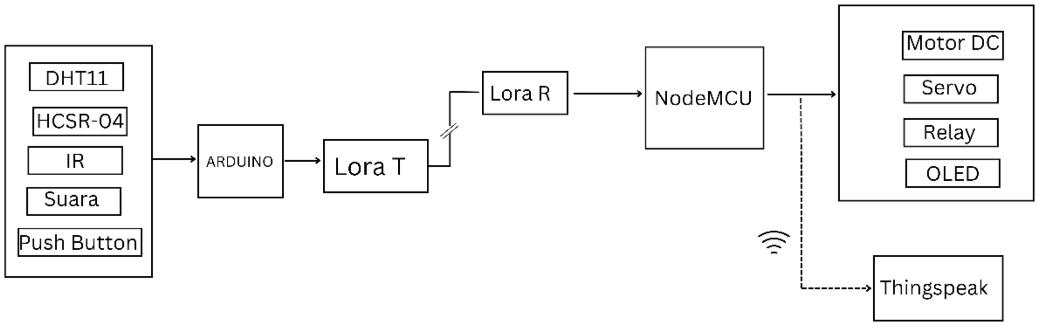


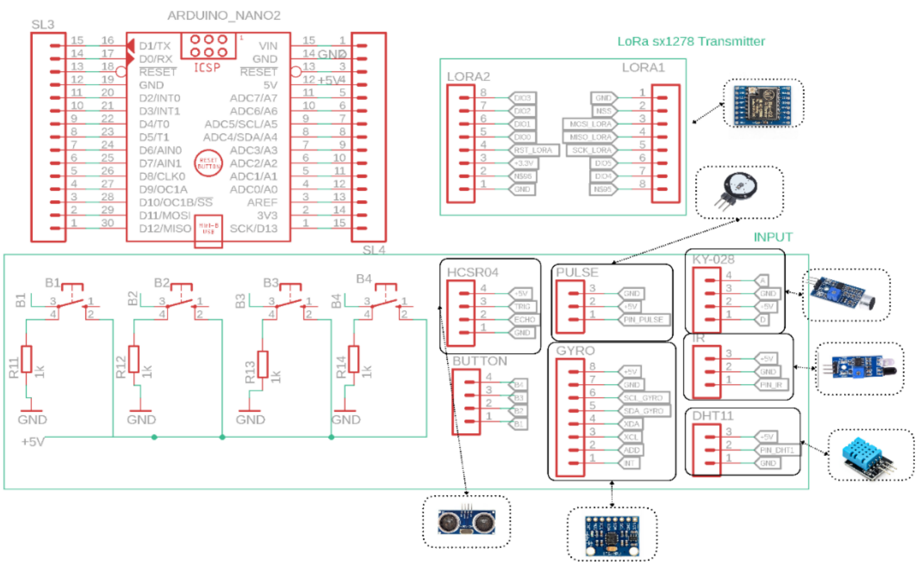
FIGURE 2. Diagram Block of Module Kits

Figure 2 is a block diagram whose inputs consist of DHT11, HCSR-04, IR sensor, sound sensor, and push button. This input data will be sent through the LoRa network. This input will produce an analogue or digital signal which will be translated into digital data by the microcontroller. The input data that has been processed by Arduino will be sent through the LoRa Transmitter, and received by the LoRa Receiver. The data received by the LoRa Receiver will be processed again by NODEMCU to receive commands to condition the output. The output can be a visual display such as OLED, moving devices such as DC motors, servos, and relays, and can also be monitored on Thingspeak.

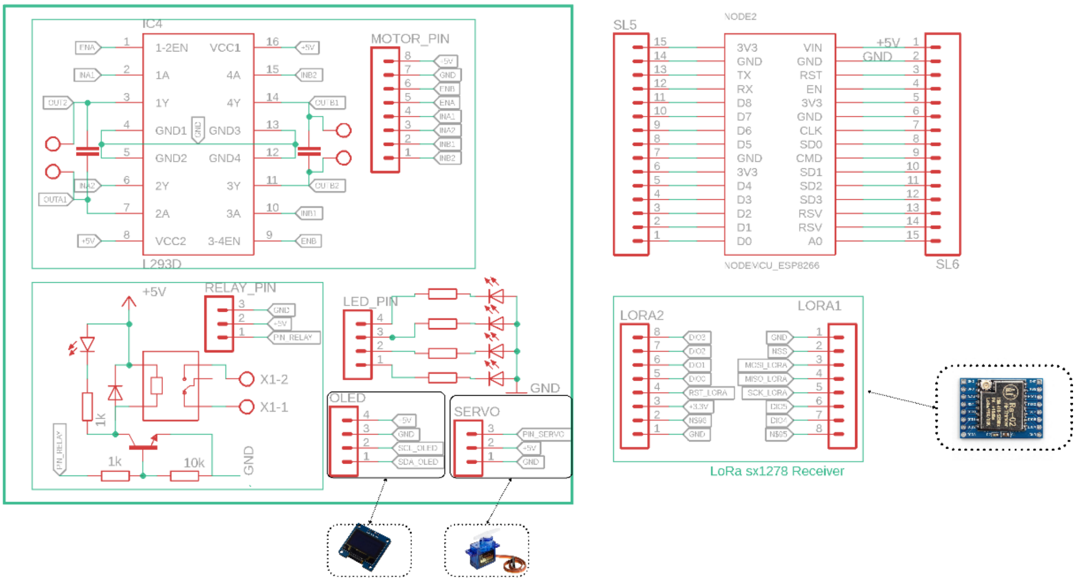
2.2 Hardware Design

The design starts from making the circuit. The circuit design consists of several pinheaders to connect the components used. The circuit consists of a transmitter and receiver circuit. The transmitter circuit functions to transmit data from the input, while the receiver circuit functions to receive input data sent from the transmitter circuit and can then be given action by the available outputs.

The system of this design is that students will be able to connect the circuit using jumpers to conduct an experiment using LoRa. The LoRa transmitter circuit and the Lora receiver circuit can be seen in Figure 3.



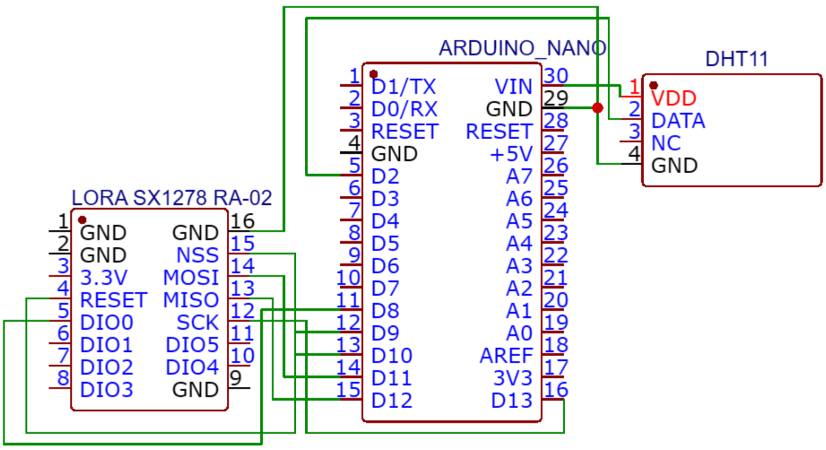
(a)



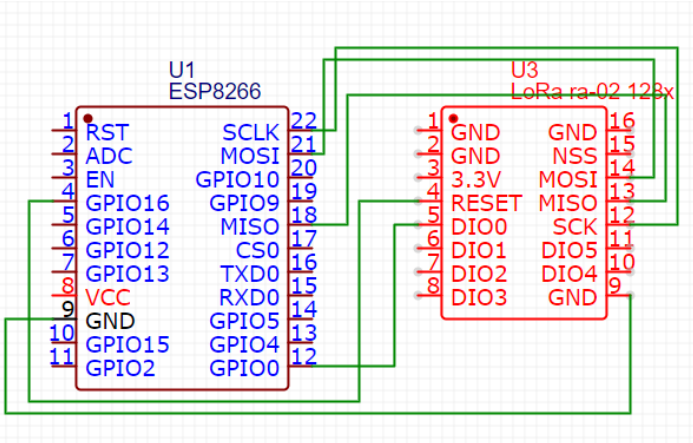
(b)

FIGURE 3. Module Kit Electronic Circuit
(a) Design of transmitter circuit
(b) Design of receiver circuit

By creating a modular circuit design, students can perform several experiments with a variety of inputs and outputs needed for learning. As a test, the experiment that will be carried out is remote temperature monitoring. The circuit that will be used can be seen in Figure 4.



(a)



(b)

FIGURE 4. Experiment Circuit
(a) DHT11 Conection, LoRa Transmitter using Arduino Nano
(b) LoRa Transmitter connected NodeMCU

Based on Figure 3 (a), it can be explained that the RST, NSS, MISO, MOSI and SCK pins on LoRa are connected to pins D9, D10, D11, D12, and D13 on the Arduino, while Pin DI00 on LoRa is connected to pin D2 on the Arduino Nano, DHT11 pin has 3 pins, Vdd pin is connected to Vin Arduino Nano, and Vcc and GND are connected to Vcc and GND Arduino Nano.

Based on Figure 3(b), it can be explained that in the LoRa Receiver circuit, the RST, NSS, MISO, MOSI, and SCK pins on the LoRa module are connected to the GPIO6, GPIO15, GPIO12, GPIO13, and GPIO14 pins on the NODEMCU, respectively. In addition, the DI00 pin on the LoRa module is connected to the GPIO0 pin on the NODEMCU. Meanwhile, the VCC and GND pins on the LoRa module are connected to the VCC and GND pins on the NODEMCU.

Furthermore, there is an input used, namely DHT11, the data pin is connected to leg 2 of the Arduino Nano, the VDD pin is connected to the 5 Volt leg of the Arduino and the GND leg is connected to the GND of the Arduino Nano.

2.3 Software Design

Programming in this research uses Arduino IDE software version 2.2.2 for programming Arduino Nano and NodeMCU microcontrollers. Programming for the remote temperature monitoring experiment requires 2 programmes. The first programme is the acquisition of DHT11 sensor data and sending DHT11 sensor data to the LoRa Receiver using the LoRa Receiver can be seen in the programming list 1.

List Programming 1

```
#include <DHT.h>
#include <LoRa.h>
#define DHTPIN 3       // Pin sensor DHT11 terhubung
#define DHTTYPE DHT11   // Jenis sensor DHT

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  while (!Serial);
  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa initialization failed. Check your wiring!");
    while (1);
  }
  dht.begin();
}
void loop() {
  float humidity = dht.readHumidity();
```

```
  float temperature = dht.readTemperature();

  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  String payload = String(temperature) ;
//  + "," + String(humidity);
  LoRa.beginPacket();
  LoRa.print(payload);
  LoRa.endPacket();
  Serial.println("Data sent: " + payload);
  delay(15000); // Kirim data setiap 5 detik
}
```

The programme is used to read the temperature data from the DHT11 sensor and send it through the LoRa module. The program consists of two main parts, namely the initialisation of the DHT11 sensor and LoRa module, and data transmission.

In the initialisation section, the program sets up communication with the DHT11 sensor and LoRa module. The DHT11 sensor is connected to pin 3 and the LoRa module is set to operate at a frequency of 433 MHz.

In the loop section, the programme reads the temperature and humidity from the DHT11 sensor. If the sensor reading is successful, the temperature data is retrieved and converted into a string in the "payload" variable. Then, the temperature data is sent through the LoRa module using the "beginPacket", "print", and "endPacket" methods. The data that has been sent is also printed to the Serial Monitor for monitoring. The programme will repeat sending data every 15 seconds.

The second programme, which is the programme to receive data from the LoRa Transmitter and send it to Tingkspeak for monitoring, can be seen in the programming list 2.

List Programming 2

```
#include <SPI.h>
#include <LoRa.h>
#include <WiFi.h>
#include <ThingSpeak.h>

WiFiClient client;
const char *ssid = "Stechoqatas5GZ";
const char *pass = "Rara1234";
long myChannelNumber = 2251742;
const char myWriteAPIKey[] = "UK81ICESOCPDGVPB";

#define ss 15
#define rst 16
#define dio0 2

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(200);
    Serial.print(".");
  }




  Serial.println();
```

```
  Serial.println("NodeMCU is connected!");
  Serial.println(WiFi.localIP());
  ThingSpeak.begin(client);
  LoRa.setPins(ss, rst, dio0);
  if (!LoRa.begin(433E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
  LoRa.receive();
}

void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    String receivedData = "";
    while (LoRa.available()) {
      receivedData += (char)LoRa.read();
    }
    int rssiValue = LoRa.packetRssi();
    Serial.print("Received packet: ");
    Serial.println(receivedData);
    Serial.print("RSSI: ");
    Serial.print(rssiValue);
    Serial.println(" dBm");
    sendDataToThingSpeak(receivedData, rssiValue);
  }
}
void sendDataToThingSpeak(String data, int rssi) {
  if (client.connect("api.thingspeak.com", 80)) {
    String postData = "field1=" + data + "&field2=" + String(rssi);
    client.println("POST /update HTTP/1.1");
    client.println("Host: api.thingspeak.com");
    client.println("Connection: close");
    client.println("X-THINGSPEAKAPIKEY: " + String(myWriteAPIKey));
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println("Content-Length: " + String(postData.length()));
    client.println();
    client.print(postData);
    client.println();
    client.stop();
  }
  delay(10000);  // Delay untuk pengiriman data ke ThingSpeak
}
```

The program is an application that allows the NodeMCU to receive data through the LoRa module and send it to the ThingSpeak platform for monitoring and analysis. In the setup, the NodeMCU initialises the WiFi connection and LoRa, and sets parameters for ThingSpeak and LoRa. In the loop, the NodeMCU checks if any data is received through the LoRa, then retrieves the data along with the RSSI value from the LoRa and sends it to ThingSpeak using an HTTP POST request. The data sent to ThingSpeak includes the received data and RSSI value. The programme periodically repeats this process every 10 seconds with delay(10000). The data sent to ThingSpeak can be used to monitor, visualise, or perform further analysis on the ThingSpeak platform.

3. Results And Discussion

3.1 Lora distance testing

Testing is done around Andalas University with different distances from 40 to 300 metres. LoRa used in this tool is the LoRa SX1278 Ra-02 series with a frequency of 433 MHz. In the test, the LoRa Transmitter was brought away from the LoRav receiver.

The first data retrieval with a distance of ± 40 metres, and the data retrieval is done several times every 5 seconds sending data. This test aims to determine the optimum distance that signals can be transmitted through LoRa communication.

TABEL 1. Lora distance testing

Distances (Metres)	Transmitter	Receiver
40	Data Sent	Data Received
50	Data Sent	Data Received
60	Data Sent	Data Received
70	Data Sent	Data Received
120	Data Sent	Data Received
150	Data Sent	Data Received
190	Data Sent	Data Received
200	Data Sent	Data Received
210	Data Sent	Data Received
230	Data Sent	Data Received
240	Data Sent	Data Not Received
300	Data Sent	Data Not Received

Based on Table 1, which is the result of tests conducted at a distance of 40 to 230 metres, the Lora Transmitter can send data and is received by the Lora Receiver, but at a distance of more than 230 metres, in tests carried out measurements up to 300 metres, the Lora Transmitter can still send data but is not received by the Lora Receiver, so no tests were carried out with a distance exceeding 300 metres.

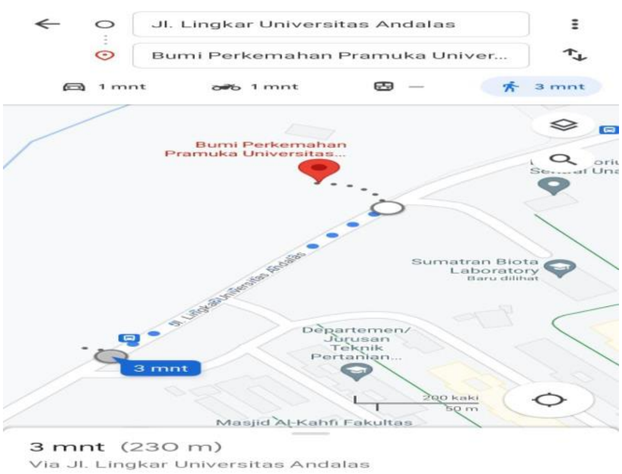


FIGURE 5. Maximum Distance of Lora Data Transmission

Shown in Figure 5 is the maximum distance that LoRa can send which is shown in Table 4.5 with a distance of 230 metres.

3.1 Testing of RSSI value

In testing RSSI data, the author took several samples of testing distances. Among them are distances of 40, 70, 190 and 230 metres. The following is a graph of the test results in Figure 6.

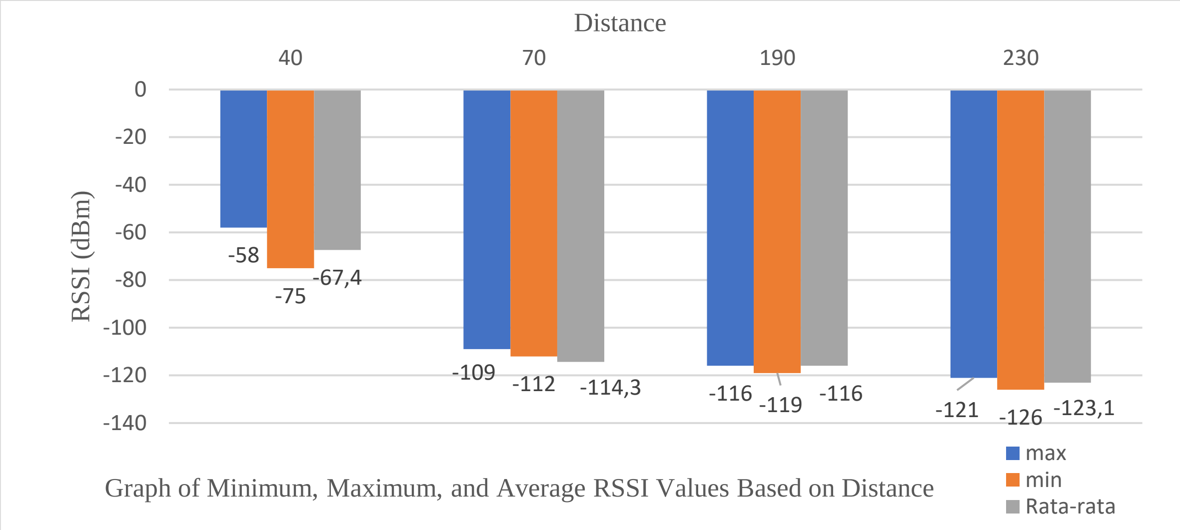


FIGURE 6. Graph of Minimum, Maximum, and Average RSSI Values Based on Distance

As can be seen in Figure 6, the highest value at a distance of 40 m is -58 dBm and the lowest value is -75 dBm. For a distance of 70 m, the highest value is obtained -109 dBm and the lowest value is -112 dBm. For a distance of 190 m, the highest value is -116 dBm and the lowest at -119 dBm. At a distance of 230 m, the highest value is -120 dBm and the lowest at -126 dBm. In testing, the longer the distance will cause more obstacles to be encountered, so the variation in RSSI value is less. Even for a distance of 230 m from 10 times sending data, only two data entered with an RSSI value of -121 dBm. This is caused by obstacles in the distance travelled such as trees and affects the transmitting power of the device. From the values obtained, the average RSSI value is then calculated to see changes in RSSI value at each test point. For more details can be seen in table 2 below:

TABLE 2. RSSI value table of test results

Distances (metres)	Maks	Min	Average
40	-58	-75	-67.4
70	-109	-112	-114.3
190	-116	-119	-116
230	-121	-126	-123.1

From the data in Table 2, it can be seen that based on the average RSSI value, there is a decrease in RSSI value when the distance between the transmitter and receiver is getting farther. RSSI values range from -58 dBm to -126 dBm. The highest RSSI value of -58 dBm is obtained at a minimum distance of 40 m. While the lowest value of -126 dBm was obtained at the maximum test distance of 230 m.

3.2 Testing with Tx in Outdoor and Indoor

LoRa testing is done by comparing data when the Lora Transmitter is carried away from the Lora Receiver with the state of the Lora Receiver indoors (Indoor) and outdoors (Outdoor). Testing is carried out in the Irrigation road area where there are multi-storey buildings and trees.

TABEL 3. Testing with Tx in Outdoor and Indoor

Distances (metres)	Tx – Indoor	Tx – Outdoor
10	Data Sent	Data Sent
20	Data Sent	Data Sent
50	Data Sent	Data Sent

100	Data Sent	Data Sent
110	Data Sent	Data Sent
150	Data Sent	Data Sent
190	Data Not Sent	Data Sent
200	Data Not Sent	Data Sent
210	Data Not Sent	Data Sent
240	Data Not Sent	Data Not Sent

It can be seen in Table 3 that when the distance is more than 190 metres, LoRa in the Indoor position no longer receives data, but LoRa in the Outdoor position still receives data, at a distance of 240 metres LoRa in the Outdoor position no longer receives data. From these tests it can be seen that the position of LoRa affects data transmission and is also affected by buildings and trees that are in the position of the LoRa receiver and Transmitter.

3.3 Testing data transmission to ThingSpeak

Testing Lora monitoring with ThingSpeak requires an internet connection and Wi-Fi network so that data can be sent and received by the system. The application used as the centre is the Thingspeak application that can be accessed through a browser. After making some initial settings, this application functions as a link between the data sending and receiving devices via a smartphone. This allows data to be monitored from various locations as long as an internet connection is available. Once the configuration is complete, the first step in testing is to switch on the assembled system, provided that the NodeMCU programme has been uploaded to the device. Next, ensure that the Wi-Fi hotspot is available and as set up in the programme. After the preparation stage is complete, the next step is to run the Thingspeak application on a smartphone or personal computer.

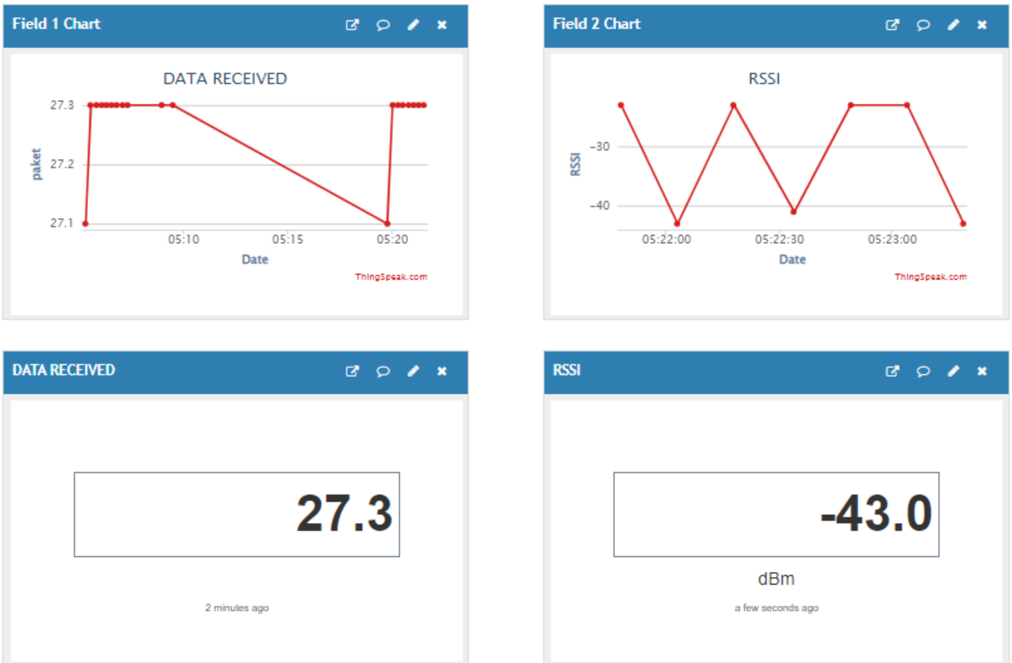


FIGURE 7. Graph of Minimum, Maximum, and Average RSSI Values Based on Distance

In LoRa testing, ThingSpeak is used to monitor the data values that have been received by the LoRa Receiver. The data displayed on ThingSpeak is the value of the data that has been received by the Lora Receiver and also the value of RSSI. Based on Figure 7, it can be seen that packets that have been received by the LoRa reciver can be received by Thingspeak via NodeMCU and displayed in real-time.

4. CONCLUSIONS

In this research, the Internet of Things laboratory kit module was developed by adding communication with LoRa sx 1278 Ra-02. Based on the test results, it can be concluded that sending using Lora technology is affected by distance and obstacles. At distances up to about 230 metres, the Lora Transmitter is able to send data that can be received by the Lora Receiver. However, at a distance of more than 230 metres to 300 metres, the data sent by the Lora Transmitter cannot be received by the Lora Receiver. This shows that distance plays an important role in Lora's communication capabilities.

In addition, obstacles such as walls or other interference can also affect the ability of the Lora Transmitter and Lora Receiver to communicate properly. Therefore, Lora users need to consider distance and obstacle factors when planning the installation or use of Lora devices to ensure reliable communication. The data that has been received by LoRa is successfully recorded and monitored on the Thingspeak platform as an Internet of Things learning.

REFERENCES

[1] E. Madona, . Yulastri, A. Nasution, and M. Irmansyah, "Design and Implementation of Portable and Prospective Embedded System and IoT Laboratory Kit Modules," vol. 4, pp. 686–691, 2023, doi: 10.5220/0010951100003260.

[2] A. Baksi, M. Bhattacharjee, S. Ghosh, S. K. Bishnu, and A. Chakraborty, "Internet of Things (IOT) Based Ambulance Tracking System Using GPS and GSM Modules," pp. 0–3, 2020.

[3] N. Jia and C. Zheng, "Design of Intelligent Medical Interactive System based on Internet of Things and Cloud Platform," 2018 10th Int. Conf. Intell. Human-Machine Syst. Cybern., vol. 01, pp. 28–31, 2018, doi: 10.1109/IHMSC.2018.00015.

[4] D. Korgut, D. F. Pigatto, and C. Daeln, "An Internet of Things-based House Monitoring System," 2018 IEEE Symp. Comput. Commun., pp. 1149–1152, 2018.

[5] M. Abdelhafeez, S. Member, H. Ali, and M. Abdelraheem, "Design and Operation of a Lightweight Educational Testbed for Internet of Things Applications," vol. 4662, no. c, pp. 1–15, 2020, doi: 10.1109/JIOT.2020.3012022.

[6] S. Dey and A. Roy, "Home Automation Using Internet of Thing," pp. 1–6, 2016.

[7] S. Nikoletsea, J. Fernandes, and S. Krco, "Internet of Things and Crowd Sourcing – a Paradigm Change for the Research on the Internet of Things," 2015.

[8] W. Dqg, F. D. Q. Lpsuryh, U. Dqg, and U. Frvwv, "Design and Research of Substation Measurement System Based on Internet of Things Technology," pp. 559–563.

[9] W. Shi et al., "Temporal Dynamic Matrix Factorization for Missing Data Prediction in Large Scale Coevolving Time Series," IEEE Access, vol. 4, pp. 6719–6732, 2016, doi: 10.1109/ACCESS.2016.2606242.

[10] I. Conference and C. Engineering, "Design and Implementation of Internet of Things Information Security Transmission Based On PBFT Algorithm," pp. 201–205, 2020, doi: 10.1109/ICCEA50009.2020.00051.

[11] X. U. Qing, X. Xin, F. Guo-li, G. U. O. Ming-jing, and W. A. N. Luan, "Design of Intelligent Campus Multimedia Interactive System Based on Internet of Things Technology," 2019 Int. Conf. Virtual Real. Intell. Syst., vol. 1, pp. 223–226, 2019, doi: 10.1109/ICVRIS.2019.00062.