**Using LoRa modules for IoT applications**

M. Prasanna

Dept. of Physics & Electronics

Bhavan’s Vivekananda College

Secunderabad, Telangana, India

[prasanna.elec@bhavansvc.ac.in](mailto:prasanna.elec@bhavansvc.ac.in)

Yogesh Soni

Bhavan’s Vivekananda College

Secunderabad, Telangana, India

[yogeshsoni28704@gmail.com](mailto:yogeshsoni28704@gmail.com)

K. Varshitha

[varshithareddy464@gmail.com](mailto:varshithareddy464@gmail.com)

**Abstract**

LoRa (Long Range) is a wireless data communication technology that uses a radio modulation technique that can be generated by Semtech LoRa transceiver chips used for the Internet of Things (IoT) and machine-to-machine (M2M) devices through the cloud. LoRa sensors have a wide range of compatibilities, allowing connectivity with various machines, devices and even animals and people. LoRa modulation provides a greater communication range with low bandwidths than other wireless data transmission technologies like cellular, WiFi, Bluetooth, or ZigBee. This paper introduces simple applications of LoRA modules.

1. **INTRODUCTION**

LoRa technology provides long-range, low-power connectivity for various Internet of Things (IoT) and machine-to-machine (M2M) applications. It stands out for its ability to cover large distances while consuming minimal energy, making it well-suited for applications where devices are spread over wide areas and need to transmit data over extended periods. Cycleo, a company of Grenoble, France has developed the technology, later Semtech has taken over. LoRa technology operates in the sub-GHz frequency bands, ISM (Industrial, Scientific, and Medical) bands: 433 MHz, 868 MHz, and 915 MHz, which offer better propagation characteristics compared to higher frequency bands.

One of the key features of LoRa is its unique modulation scheme, known as Chirp Spread Spectrum (CSS). This modulation enables LoRa devices to transmit data over a range of frequencies in a chirping pattern, allowing the signal to overcome interference and obstacles more effectively, thus enhancing the communication range. LoRa is preferable for sensor nodes running on a coil cell or solar powered, that transmit small amounts of data.

LoRaWAN (Long Range Wide Area Network) is a network protocol constructed on the foundation of LoRa technology. It provides the infrastructure and communication framework necessary to connect and manage a large number of LoRa devices within a wide geographical area. LoRaWAN is designed to optimize battery life, network capacity, and scalability, making it a best choice for IoT applications such as agriculture, industrial monitoring, smart cities, etc.. Security is a paramount consideration in IoT applications, and LoRaWAN incorporates various security measures. It employs end-to-end encryption, message integrity checks, and unique device identifiers to ensure the confidentiality and integrity of transmitted data. LoRa serves as the radio signal transporting the data, while LoRaWAN functions as the communication protocol governing and specifying how this data is transmitted within the network.

In conclusion, LoRa and LoRaWAN collectively offer a powerful solution for long-range, low-power IoT connectivity. The advantages of using LoRaWAN Wireless technology over Wi-Fi make it ideal for deploying IoT devices in large-scale applications, such as smart cities and industrial automation. As the IoT continues to grow, LoRaWAN is likely to become the preferred wireless communication technology for IoT applications.

1. **FEATURES:**

|  |  |  |
| --- | --- | --- |
| Long Range: | LoRa Long Range | Extends its reach to over 10 kilometers in rural regions and penetrates dense urban or deep indoor environments. |
| Low Power | LoRa Low Power | Consumes minimum energy, minimizing battery replacement costs |
| Secure | LoRa Secure | end-to-end encryption, mutual authentication, integrity protection, and confidentiality. |
| Standardized | LoRa Standardize | The compatibility of devices and easy access to LoRaWAN networks facilitate the rapid deployment of IoT applications. |
| Geo location | LoRa Geolocation | Suitable for GPS-free tracking applications, |
| Mobile | LoRa Mobility | Sustains communication with mobile devices without strain on power consumption. |
| High Capacity | LoRa High Capacity | It has the capacity to handle millions of messages per base station, meeting the demands of public network operators catering to expansive markets. |
| Low Cost |  | Low investment on infrastructure, battery replacement, and ultimately operating expenses |

**III. Advantages and Disadvantages of LoRa module**

**Advantages :**

### **1. **Long transmission distance****

The LoRa module is popular for its transmission distance which far exceeds other series of wireless modules, under the same power conditions.

**2. Low power**

The challenge of preserving low power consumption in wireless modules while simultaneously extending their transmission range is solved by the LoRa module. This achievement significantly contributed to its widespread popularity in the market.

### ****3. Strong anti-interference ability****

The LoRa module uses Spread spectrum modulation, a technique used in wireless communication to improve the signal's resistance to interference and to enhance its security forward error correction technology to detect and correct errors that may occur during the transmission of data.

### ****4. High sensitivity****

In spread spectrum modulation, the signal extends over a wider range of frequencies than the minimum required for transmission. When operating at the same data rate a higher sensitivity than other modulation methods can be obtained. The higher the sensitivity, the longer the transmission distance, the weaker the signal.

**Disadvantages**

### ****1. Low transmission rate****

The benefit of an extended transmission range with LoRa modules often involves trade-offs in data transmission speed. As the transmission rate increases, the achievable transmission distance decreases. This is because higher rates make the signal more susceptible to weakening.

### ****2. LoRa module price****

The LoRa module is relatively more expensive than other the RF modules. It is recommended to use the LoRa module as its performance of various parameters is much superior than all kinds of wireless modules.

1. **LoRa and LoRa WAN technology**  will be an indispensable technology in near future in IoT deployments, some of the applications are listed as follows**:**

In Smart Cities:

* Smart Lighting
* Air Quality Monitoring
* Smart Parking and Vehicle Management
* Facilities and Infrastructure Management
* Environmental Monitoring
* Waste Management

For Industrial Use:

* Radiation and Leak Detection
* Smart Sensors
* Item Location and Tracking
* Shipping and Transportation

In Smart Homes:

* Enhanced Security
* Home Automation with IoT-enabled Appliances
* Healthcare with Health Monitoring Devices
* Wearable Technology

In Agriculture:

* Smart Farming and Livestock Management
* Monitoring Temperature and Moisture
* Water Level Sensors for Irrigation Control

1. **Working with Lora devices**

Download the LoRa.h library for LoRa devices or LoRa\_E32.h specifically for E32 lora devices

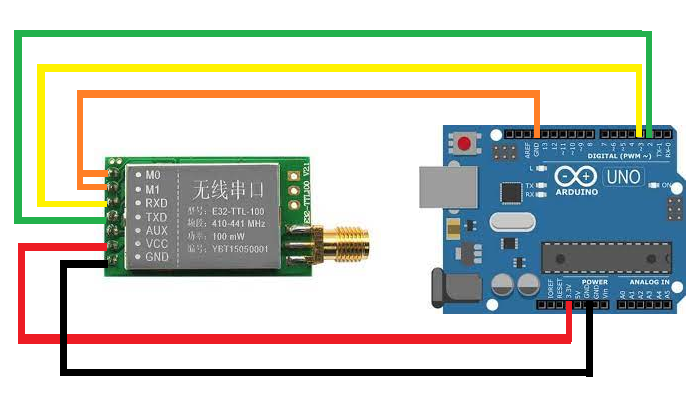
Go to the Example program in Ebyte Lora 32 Library.

1. **The below code is for sending and receiving Message between two E32 LoRa devices’s which are kept at about 1Km distance .**

**Arduino UNO and REYAX RYLR896 LoRa Module are used.**

**Note:** *The transmitter and receiver must have the same CHANNEL ADDL and ADDH*

To set configuration of E32 LoRa, go to SETConfiguration example which is available in Examples > E32 library. The circuit connections are shown below in fig.1



M0 to GND

M1 to GND

TX to PIN 2

RX to PIN 3

AUX is not connected

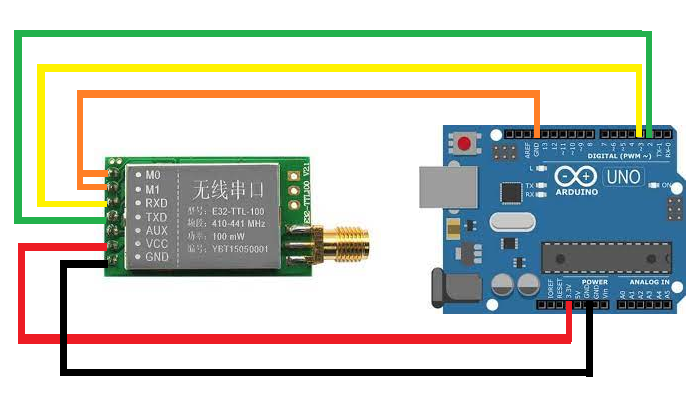
VCC to 3.3v/5v

GND to GND

Fig.1

|  |
| --- |
| #include"Arduino.h"  #include "LoRa\_E32.h"  LoRa\_E32e32ttl100(2, 3); // Configure without connecting AUX and M0 M1  voidsetup() {  Serial.begin(9600);  delay(500);  e32ttl100.begin();  Serial.println(" Sending message!");  ResponseStatusrs = e32ttl100.sendMessage("Hello everyone !");  Serial.println(rs.getResponseDescription());  }  voidloop() {  if (e32ttl100.available()>1)  {  ResponseContainerrc = e32ttl100.receiveMessage();  if (rc.status.code!=1)  {  rc.status.getResponseDescription();  }  else  {  Serial.println(rc.data);  }  }  if (Serial.available()) {  String input = Serial.readString();  e32ttl100.sendMessage(input);  }  } |

Receiver circuit



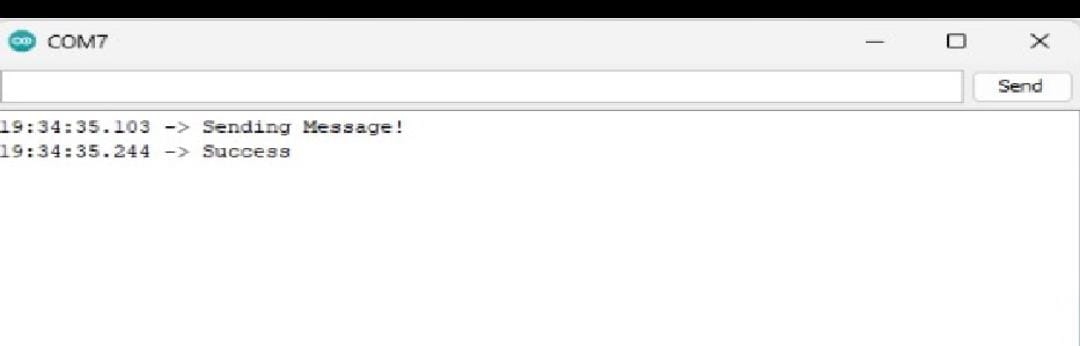
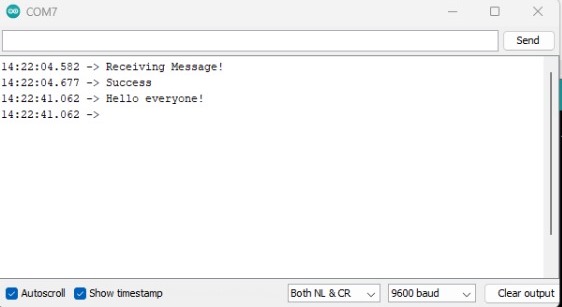
Pin 3 to Rx of LoRa

Pin 2 to Tx of LoRa

Fig.2 Receiver circuit

|  |
| --- |
| // Program at the receiver  #include"Arduino.h"  #include"LoRa\_E32.h"  LoRa\_E32e32ttl100(2, 3);  voidsetup() {  Serial.begin(9600);  delay(500);  e32ttl100.begin();  }  voidloop() {  if (e32ttl.available()>1)  {  ResponseContainerrc = e32ttl.receiveMessage();  if (rc.status.code!=1){  rc.status.getResponseDescription();  }  else  {  Serial.println(rc.data);  }  }  } |

Output at the transmitter and receiver

Display at the transmitter while sending a message Output at the receiver after getting a message

1. Sending temperature sensor data to a remote receiver

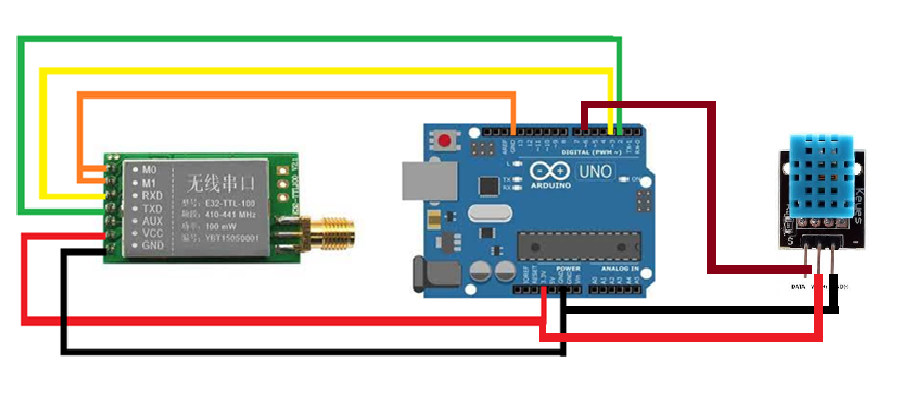
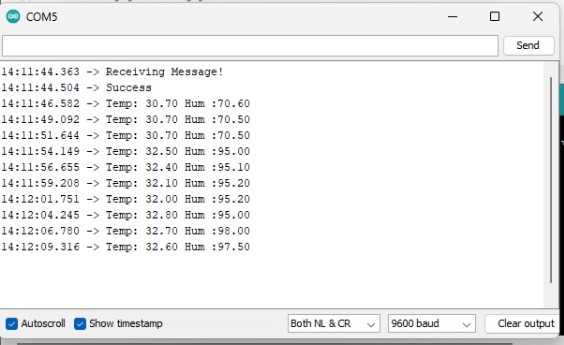


Fig.3 At the transmitter - Arduino Uno connected with Temperaure sensor

|  |
| --- |
| // Program at the transmitter side  #include "Arduino.h"  #include"LoRa\_E32.h"  #include "DHT.h"  DHTdht(6,DHT11);  LoRa\_E32e32ttl100(2, 3);  typedefstructstruct\_message {  String a;  String b;  } struct\_message;  struct\_messagemyData;  voidsetup() {  Serial.begin(9600);  delay(500);  dht.begin();  e32ttl100.begin();  Serial.println("Sending message!");  ResponseStatusrs = e32ttl100.sendMessage("Hello everyone !");  Serial.println(rs.getResponseDescription());  }  voidloop() {  float temp = dht.readTemperature(); //reads temperature  float hum = dht.readHumidity(); //reads humidity  myData.a = temp; //stores float value as string  myData.b = hum; //stores float value as string  e32ttl100.sendMessage("Temp: ");  e32ttl100.sendMessage(myData.a);  e32ttl100.sendMessage(" Hum :");  e32ttl100.sendMessage(myData.b);  delay(1000);  } |

**Receiver circuit is same as in fig2**

|  |
| --- |
| // Program at the receiver  #include"Arduino.h"  #include"LoRa\_E32.h"  LoRa\_E32e32ttl100(2, 3);  voidsetup() {  Serial.begin(9600);  delay(500);  e32ttl100.begin();  }  voidloop() {  if (e32ttl.available()>1)  {  ResponseContainerrc = e32ttl.receiveMessage();  if (rc.status.code!=1){  rc.status.getResponseDescription();  }  else  {  Serial.println(rc.data);  }  }  } |

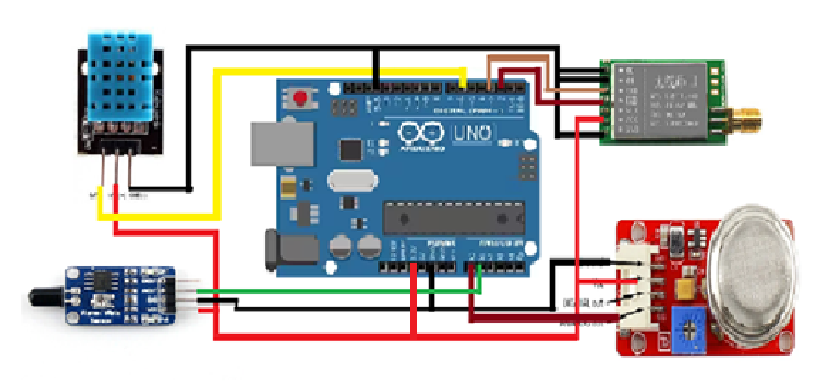


Output at the receiver with temperature and humidity values

1. **Use case: Forest Fire Detection using LoRa Device**

This project aims to the early detection of forest fires, to avoid damage to the natural resources. Our project succeeds in detecting ( or/and predicting ) the occurrence of the fire without any delay using LoRa communication system. The project makes a use of sensors and Arduino boards based on WSN to achieve this task. In this project, temperature sensor, IR sensor, Gas sensor are interfaced to Arduino to detect any increase in the concentration of gases, detect flames and measure temperature changes produced from the fire. The values are taken from the Sensors and are transmitted through the LoRa module connected to Arduino Uno board and sent to a remote receiver.

The system can be improvised to detect the fire as fast as possible and send notification to the fire units is vital. Both local and global fire alert can be provided, very low cost implementation, also camera based detection can be provided as fail safe in case sensors malfunction. A database can be created at the receiver for analysis.



LoRa E32 E32-TTL-100

Arduino UNO

M0 to GND

M1 to GND

TX to PIN 2

RX to PIN 3

AUX is Not connected

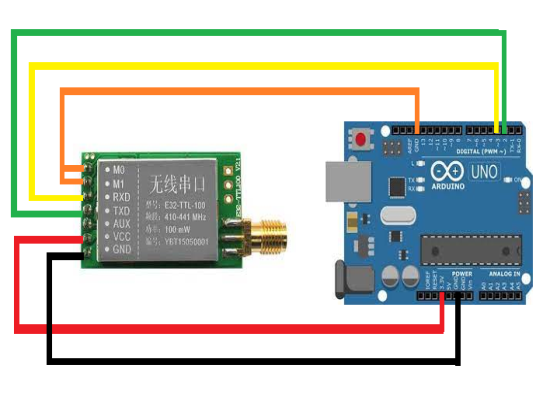
VCC to 3.3v/5v

GND to GND

Fig.4 Connection diagram at the transmitter with temp sensor, flame sensor and smoke sensor connected to Arduino Uno and LoRa device

|  |
| --- |
| // Program to measure temperature, humidity values and  // monitor presence of smoke/ flame and transmit to a remote receiver  #include "Arduino.h"  #include"LoRa\_E32.h"  #include "DHT.h"  DHTdht(6,DHT11);  int Flame = 7;  int Smoke = 8;  LoRa\_E32e32ttl100(2, 3);  typedefstructstruct\_message {  String a;  String b;  String c;  String d;  } struct\_message;  struct\_messagemyData;  voidsetup() {  Serial.begin(9600);  delay(500);  pinMode(Smoke, INPUT);  pinMode(Flame, INPUT);  dht.begin();  e32ttl100.begin();  Serial.println("Sending message!");  ResponseStatusrs = e32ttl100.sendMessage("Hello everyone !");  Serial.println(rs.getResponseDescription());  }  voidloop() {  float temp = dht.readTemperature(); //stores float values as string  float hum = dht.readHumidity();  Smoke = digitalRead(8); //reads smoke  Flame = digitalRead(7); // reads flame  delay(1000);  myData.a = temp; //converts float value to string  myData.b = hum;  e32ttl100.sendMessage("Temp: ");  e32ttl100.sendMessage(myData.a);  e32ttl100.sendMessage(" Hum :");  e32ttl100.sendMessage(myData.b);  if(Smoke == 0)  {  e32ttl100.sendMessage(" Smoke Detected "); //prints as output if smoke detected  }  else  {  e32ttl100.sendMessage(" No Smoke Detected "); //if not detected  }  if(Flame == 1)  {  e32ttl100.sendMessage(" Fire Detected "); //prints when fire is detected  }  else  {  e32ttl100.sendMessage(" No Fire Detected "); //when fire is not detected  }  delay(1000);  } |

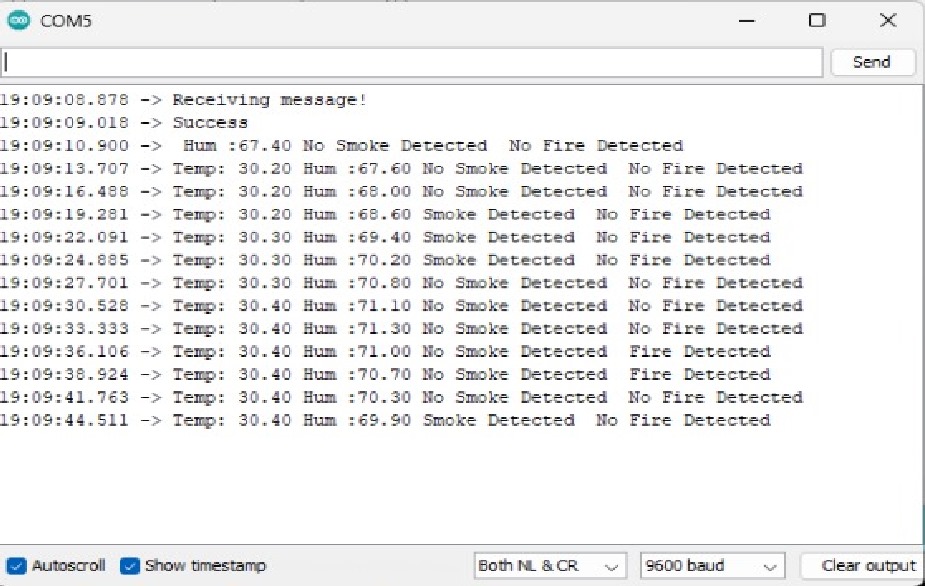
**At the receiver (Fig.5)**



Pin 3 to Rx of LoRa

Pin 2 to Tx of LoRa

|  |
| --- |
| // Program at the receiver to display the values received  #include"Arduino.h"  #include"LoRa\_E32.h"  LoRa\_E32e32ttl100(2, 3);  voidsetup() {  Serial.begin(9600);  delay(500);  e32ttl100.begin();  }  voidloop() {  if (e32ttl.available()>1)  {  ResponseContainerrc = e32ttl.receiveMessage();  if (rc.status.code!=1){  rc.status.getResponseDescription();  }  else  {  Serial.println(rc.data);  }  }  } |



Output at the receiver with temperature-humidity values, fire and smoke presence

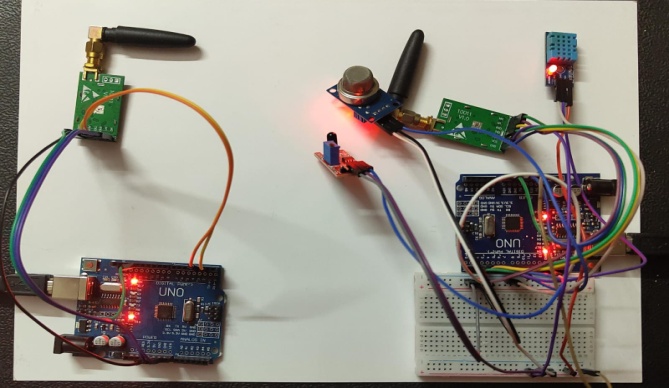
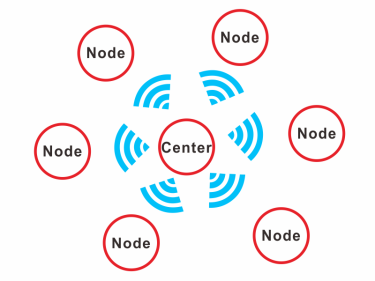


Fig.5 picture of the transmitter and receiver circuits

**Wireless multi-point sensing or measurement system using LoRA modules**

Sensing equipment cannot be directly connected to the network, and can only be connected to the center through wired or radio frequency, and the center communicates with the network to upload data. This method has a low cost. However, due to the same frequency interference of radio frequency communication, multiple sensing devices cannot communicate with the center at the same time, otherwise data will be lost. An effective scheduling algorithm for the communication between the sensing devices and the center is required to ensure Stable and reliable data transmission.



A wireless multi-point sensing measurement system.

The node collects sensor data and uploads the data to the center at regular intervals. After receiving the data, the center uploads the data to the server. The data can be saved in a database for predictive maintenance or /and disaster management. ESP boards with LoRa Connectivity can be used instead of a separate Arduino Uno boards with a LoRa Module.

**Conclusion:**

In conclusion, Lora devices have emerged as a promising technology for long-range, low-power communication in various industries. With their ability to provide wide coverage, low energy consumption, and secure data transmission, Lora devices have demonstrated their potential for applications such as smart cities, agriculture, logistics, and industrial monitoring. While there may be some limitations to consider, such as network congestion and limited bandwidth, the advantages of Lora devices render them a valuable asset in the IoT era. As technology continues to advance, the use of Lora devices is expected to grow, bringing new possibilities and innovations to numerous sectors, ultimately leading to increased efficiency, improved decision-making and enhanced connectivity. Overall, Lora devices have proven to be a reliable and efficient solution for long-range communication needs, offering tangible benefits to businesses and society alike.

References:

1. [www.arduino.cc](http://www.arduino.cc)
2. [www.semtech.com](http://www.semtech.com)
3. <https://randomnerdtutorials.com/>