POLITECNICO DI MILANO
Dipartimento di Elettronica, Informazione e Bioingegneria
Master of Science in Telecommunication Engineering

Data Transferring and Analysis with Arduino and LoRa Network

Supervisor: Prof. Matteo Cesana
Co-Supervisor: Prof. Razvan Pitic

Master Thesis of:
Bhanu Prakash Reddy Nagarla (Student ID: 879090)
Navya Garla Venkatesh Prasad (Student ID: 874666)

Academic year 2018 - 2019

Table of Contents

1 Introduction 1

1.1 Abstract 1

1.2 Context 2

2 Backgrounds 3

2.1 Literature Survey 3

2.2 LoRaWAN 6

2.2.1 LoRaWAN Protocol Stack 6

2.2.2 LoRa Network Architecture 7

2.3 Node RED 8

2.4 Publish/Subscribe for IoT 8

2.4.1 Brokered 9

2.4.2 Multicast 9

2.5 Arduino IDE 9

2.6 MySQL Database 9

3 Architecture 10

3.1 Hardware 11

3.1.1 Arduino MKR WAN 1300 board 11

3.1.2 DHT22 Sensor 12

3.1.3 Dipole GSM Antenna 13

3.2 Cayenne LPP (Low Power Payload) 13

3.3 LoRa Gateway 4

3.4 Software 14

4 Demonstration and Results 17

4.1 Relative Humidity Results 17

4.2 Temperature Results 18

5 Conclusion 19

5.1 Future Enhancements 19

5.2 Future vision of IoT 19

5.3 Application and future of LoRaWAN Technology 20

5.3.1 Smart City 20

5.3.2 Industrial Applications 20

5.3.3 Smart home applications 20

5.3.4 Healthcare 20

5.3.5 Agriculture 20

1 Introduction

In the near future, all our everyday things and objects will be both connected to the Internet and equipped with enough sensing, acting and processing capabilities to exploit the full potential benefits of the so-called Internet of Things (IoT) paradigm. Even the simplest objects will become “smart” because they will be interconnected to other objects to share and collect data from the environments in which they are placed thus paving the way to novel application services, computing and communication scenarios.

In this context, interoperability among different standards and communication technologies is still a significant challenge that we have started to address by proposing a LoRa gateway acting as a flexible and transparent interface between different IoT devices and the Internet.

LoRa (Long Range) is a relatively new radio technology intended for reliable communication of small amounts of data over distances that are longer than achievable with Bluetooth or Wi-Fi. It is a long-range, low-power, low-bitrate, wireless telecommunications system, promoted as an infrastructure solution for the Internet of Things: end-devices use LoRa across a single wireless hop to communicate to gateway(s), connected to the Internet and which act as transparent bridges and relay messages between these end-devices and a central network server. LoRa modules are relatively cheap, but the easiest way to use LoRa is to buy development boards that have an open source microcontroller (such as Arduino) and LoRa enabled transceivers that work well.

1.1 Abstract

The aim of this work is to monitor and obtain environmental conditions such as relative humidity and temperature of a particular area using the LoRaWAN. The clients get the privilege to subscribe to the temperature and/or relative humidity topics where the MQTT broker will relay the data to my Node-RED client. The additional features of this work include long battery life and low cost. The LoRa end node operating at 868 MHz ISM band transmits the values to the LoRa gateway over the LoRa network server. The clients can retrieve the information over any desired period.

1.2 Context

The temperature and the relative humidity sensors along with a Dipole GSM antenna is connected to an Arduino MKR WAN 1300 board. The sensors sense the area’s temperature and the relative humidity for every second. It is continuously sent from the LoRa node end to the LoRaWAN gateway over a LoRa network. MQTT broker handles publish/subscribe mechanism depending on the subscription of topics by the clients. Node-RED will then store the data into a MySQL database and also provide visual analysis using graphs in Node RED Dashboard web UI. Values can be retrieved by the clients for a specific range of dates and the graphs are displayed based on that.

For this purpose, in these papers, firstly a general vision of LoRaWAN and IOT is given. In particular, a comprehensive analysis of the main components: Arduino Software and Hardware, Sensor, Node-Red, MQTT broker and MySQL is provided. Next part is dedicated to the experiment briefing. It explained how the sensor sends the information to the Gateway through Arduino sketch and sending data to the clients through a broker, further displaying values in the dashboard. Finally, a clear intuition of the graphical demonstration of temperature and relative humidity for a selected date range is shown, retrieving from the database.

2 Backgrounds

2.1 Literature Survey

There have been similar works based on this work happening around the world using different prototypes and for different applications. The following are some of the works.

2.1.1 A WiFi-enabled indoor air quality monitoring and control system.

This paper proposes an open platform of a WiFi-enabled indoor air quality monitoring and control system, which could be incorporated into such a ‘smart building’ structure. The complete software and hardware design of this system is presented, along with a series of control experiments. The proposed system operates over an existing WiFi wireless network utilizing the MQTT protocol. It is capable of monitoring the indoor air quality as well as controlling an air purifier to regulate the particulate matters concentration. Experiment results under a real-world office environment demonstrate the effectiveness of the proposed design. [1]

2.1.2 A low-power real-time air quality monitoring system using LPWAN based on LoRa.

This paper presents a low-power real-time air quality monitoring system based on the LoRa Wireless Communication technology. The proposed system can be laid out in a large number in the monitoring area to form sensor network. The system integrates a single-chip microcontroller, several air pollution sensors (NO2, SO2, O3, CO, PM1, PM10, PM2.5), Long Range (LoRa) -Modem, a solar PV-battery part and graphical user interface (GUI). As communication module LoRa sends the data to the central monitoring unit and then the data would be saved in the cloud. The range tests at an outdoor area show that LoRa is able to reach to approximately 2Km. The TX power is only about 110mA which is lower compared with other used wireless technology. An easy to use GUI was designed in the system. Based on LoRa technology, GUI, and Solar PV - battery part the system has several progressive features such as low cost, long distance, high coverage, long device battery life, easy to operate. [2]

2.1.3 IoT enabled proactive indoor air quality monitoring system for sustainable health management.

This paper proposes an IoT based indoor air quality monitoring system for tracking the ozone concentrations near a photocopy machine. The experimental system with a semiconductor sensor capable of monitoring ozone concentrations was installed near a high-volume photocopier. The IoT device has been programmed to collect and transmit data at an interval of five minutes over Bluetooth connection to a gateway node that in turn communicates with the processing node via the WiFi local area network. The sensor was calibrated using the standard calibration methods. As an additional capability, the proposed air pollution monitoring system can generate warnings when the pollution level exceeds beyond a predetermined threshold value. [3]

2.1.4 A wireless system for indoor air quality monitoring.

This paper describes the development of a wireless monitoring system which can be deployed in a building. The system measures carbon dioxide, carbon monoxide and temperature. The system developed in this paper can serve as the monitoring component of a HVAC control system and function as an indoor air quality monitor independently. [4]

2.1.5 Polluino: An efficient cloud-based management of IoT devices for air quality monitoring.

The Internet of Things paradigm originates from the proliferation of intelligent devices that can sense, compute and communicate data streams in a ubiquitous information and communication network. The great amounts of data coming from these devices introduce some challenges related to the storage and processing capabilities of the information. This strengthens the novel paradigm known as Big Data. In such a complex scenario, the Cloud computing is an efficient solution for the managing of sensor data. This paper presents Polluino, a system for monitoring the air pollution via Arduino. Moreover, a Cloud-based platform that manages data coming from air quality sensors is developed. [5]

2.1.6 An embedded system model for air quality monitoring.

Objective of the paper is to present a system model which can facilitate the assessment of health impacts caused due to indoor air pollutant as well as outdoor and can intimate the human prior about the risk he/she going to have, here we are focusing our work in context to allergic patients as they will be informed by this tool such that they can secure themselves without actually experiencing the risk factors, here a sensing network based microcontroller equipped with gas sensors, optical dust particle sensor, humidity and temperature sensor has been used for air quality monitoring. The design included various units mainly: sensing unit, processing unit, power unit, display unit, communication unit. This work will apply the techniques of electrical engineering with the knowledge of environmental engineering by using sensor networks to measure Air Quality Parameters. [6]

2.1.7 A real-time ambient air quality monitoring wireless sensor network for schools in smart cities.

In this paper, a low-cost solar-powered air quality monitoring system based on ZigBee wireless network system technology is presented. The solar powered network sensor nodes can be deployed by schools to collect and report real-time data on carbon monoxide (CO), nitrogen dioxide (NO2), dust particles, temperature, and relative humidity. The proposed system allows schools to monitor air quality conditions on a desktop/laptop computer through an application designed using LabVIEW and provides an alert if the air quality characteristics exceed acceptable levels. They tested the sensor network successfully at the Singapore campus of the University of Newcastle, Australia. The experimental results obtained by them demonstrated that the sensor network can provide high-quality air quality measurements over a wide range of CO, NO2 and dust concentrations. [7]

2.1.8 A smart sensor system for air quality monitoring and massive data collection.

Air pollution has been a global challenge for environment protection. Effectively collecting and scientifically visualizing the air quality data can better help us monitor the environment and address related issues. This article presents a smart sensor system for air quality monitoring which consists of three units: the smart sensor unit, the smartphone, and a server. [8]

2.1.9 Development of a Low-Cost Internet-of-Things System for Monitoring Soil Water Potential Using Watermark 200SS Sensors.

The objective of this study was to develop and test a low-cost IoT system for soil moisture monitoring using Watermark 200SS sensors. The system uses Arduino-based microcontrollers and data from the field sensors (End Nodes) are communicated wirelessly using LoRa radios to a receiver (Coordinator), which connects to the Internet via WiFi and sends the data to an open-source website (ThingSpeak.com) where the data can be visualized and further analyzed using MATLAB. The system was successfully tested under field conditions by installing Watermark sensors at four depths in a wheat field. The system described here could contribute to widespread adoption of easy-to-use and affordable moisture sensing technologies among farmers. [9]

2.1.10 Smart water grid management using LPWAN IoT technology

The water grid management system proposed in this paper involves different sensors deployed at various strategically chosen locations to measure the quality of water by generating real time data. The system also provides an alert mechanism which notifies the different level of authorities through email and SMS in case of any issues. Furthermore, it provides a solution for handling the locks that have been employed in and around the village to control the flow of water in a timely manner. The sensors attached with a micro controller in the LoRa module will communicate to the cloud environment through the LoRa gateway. A web page provides the interface to the residents and to the authorities to gauge the water quality after analyzing the data using the prediction algorithm. [10]

2.2 LoRaWAN

LoRaWAN being Long Range Wide Area Network platform is designed, developed, and experimentally tested, using open and low-cost hardware and software components. LoRaWAN is well suited to support IoT applications. They enable nodes to connect to the cloud over a long range and with a low power consumption. The LoRa gateway acts as a bridge to the LoRa end node and a WiFi network. The LoRa Gateway is not supposed to change its current position over time. Therefore, it has not been equipped with GPS. This design choice implies two main things:

1. Very low power consumption values.
2. Extremely reduced form factor (dimension).

2.2.1 LoRaWAN Protocol Stack

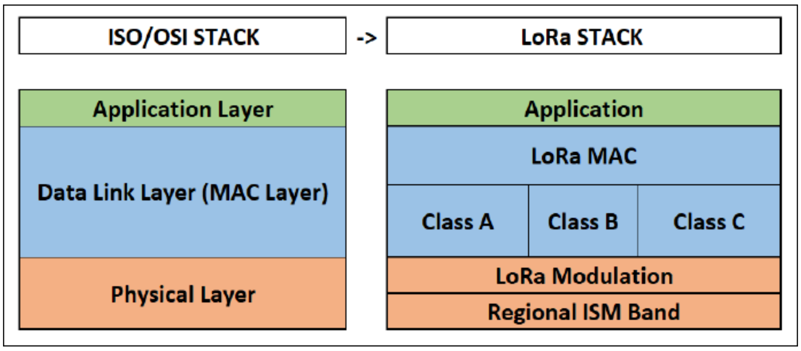


Figure 2.1 - LoRaWAN Protocol Stack.

LoRa Protocol Stack, as defined and maintained by the LoRa Alliance, can be described as reported in Figure 1. It is mainly characterized by three layers: the LoRa Physical Layer (PHY), which implements LoRa modulation scheme; the MAC layer, which implements LoRaWAN technology; the Application layer, which acts like an interface between the end-nodes and the outer world (e.g. Internet). The LoRa's PHY layer is mainly characterized by the CSS modulation scheme. A LoRa gateway directly interacts with network servers, backhaul relying on TCP/IP and Secure Sockets Layer (SSL). [11]

2.2.2 LoRa Network Architecture

The LoRaWAN network architecture has a star topology; Lora's network architecture foresees different kind of devices as shown in the figure below:

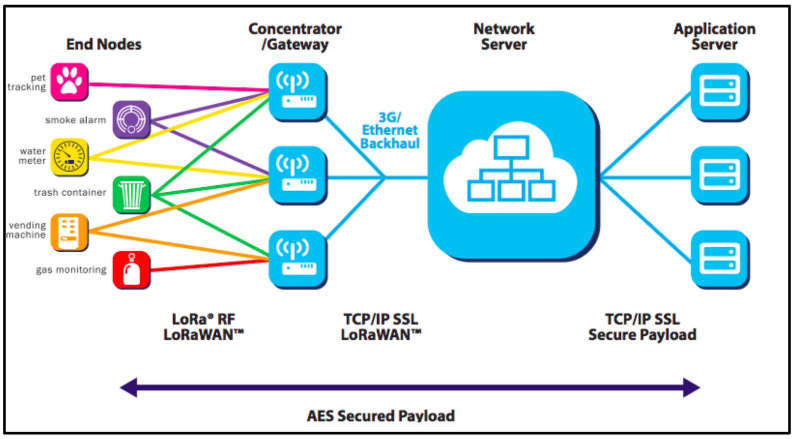


Figure 2.2 - LoRa Architecture.

1. The End-devices, which are the end-nodes of the IoT network, typically in charge of executing "smart" tasks, such as sensing or actuating. They can only communicate with LoRaWAN gateways and not directly with each other.
2. The gateway, also referred to as the base station. It represents a bridge between the IoT network and the broader Internet. The LoRaWAN gateways are only responsible for forwarding raw data packets from end nodes towards the network server encapsulating them in UDP/IP packets. It oversees interacting to the Network Server relying on 3G/4G backbone connection. Multiple gateways are connected to a central network server.
3. The Network Server, indeed, a central entity in charge of collecting and storing data, while making them available for the users. This is responsible for sending downlink packets and MAC commands towards end devices, if needed. Further, the communication terminates at the application servers that can be owned by third parties.
4. The Application Server, for instance the interface between the developers and the Network Server. It offers user-friendly functionalities and interfaces to develop specific web services related to a certain specific subset of the amount of data available on the Network Server. Multiple application layers can be connected to a single network server.
5. Communications between the LoRa node and the LoRa gateway are based on CLASS A specifications.

There are two different joining procedures for connecting end-devices to a reference gateway: Over-The-Air Activation (OTAA) and Application by Personalization (ABP).

Media access control (MAC) protocol is used for WANs. Some major constraints are:

1. Low power consumption.
2. Limited computational and storage capabilities.
3. Execution of small process.
4. Low form factor.
5. Security and E2E encryption.

2.3 Node RED

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to runtime in a single click.

JavaScript functions can be created within the editor using a rich text editor. A built-in library allows you to save useful functions, templates or flows for re-use. This is built on Node.js, which makes it ideal to run at the edge of the network on low-cost hardware's such as interacting with Arduino as well as in the cloud. With over 225,000 modules in Node package repository, it is easy to extend the range of palette nodes to add new capabilities. The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others. [12]

2.4 Publish/Subscribe for IoT

Publish & Subscribe for the Internet of Things is a new simple and powerful protocol that implements the publish/subscribe communication pattern between devices, provides an alternative to traditional client-server architecture. In the client-server model, a client communicates directly with an endpoint. The pub/sub model decouples the client that sends a message (the publisher) from the client or clients that receive the messages (the subscribers). The publishers and subscribers never contact each other directly. In fact, they are not even aware that the other exists. The connection between them is handled by a third component (the broker). The job of the broker is to filter all incoming messages and distribute them correctly to subscribers. There are mainly two implementation approaches generally used (out of several approaches).

1. Brokered (e.g. MQTT).
2. Multicast (e.g. DDS).

2.4.1 Brokered

In brokered pub/sub systems publishers and subscribers connect to a centralized server that routes publications to matching subscribers. In this, publishers and subscribers typically maintain a long-term connection to the broker. This is often necessary because the broker is running in the cloud and the subscriber and publishers are typically running behind a firewall and must establish an outbound connection to the broker to be able to communicate. PS does not maintain long term connections, in fact connections only last long enough to send a single subscription or publication message.

2.4.2 Multicast

In a multicast pub/sub system, subscribers receive messages from all publishers and selectively forward matching publications up to the application. [13]

2.5 Arduino IDE

The **Arduino Integrated Development Environment (IDE)** is a cross-platform application that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards. The Arduino IDE also supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. [14]

2.6 MySQL Database

MySQL is developed, distributed, and supported by Oracle Corporation., which is a Swedish company. MySQL is becoming so popular because of many good reasons.

1. It is a database system, used for developing web-based software applications. It is fast, reliable, flexible, easy to download and use.
2. It is a RDBMS being used for both small and large applications.
3. It uses a standard form of the well-known SQL data language.
4. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets.
5. It is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

3 Architecture

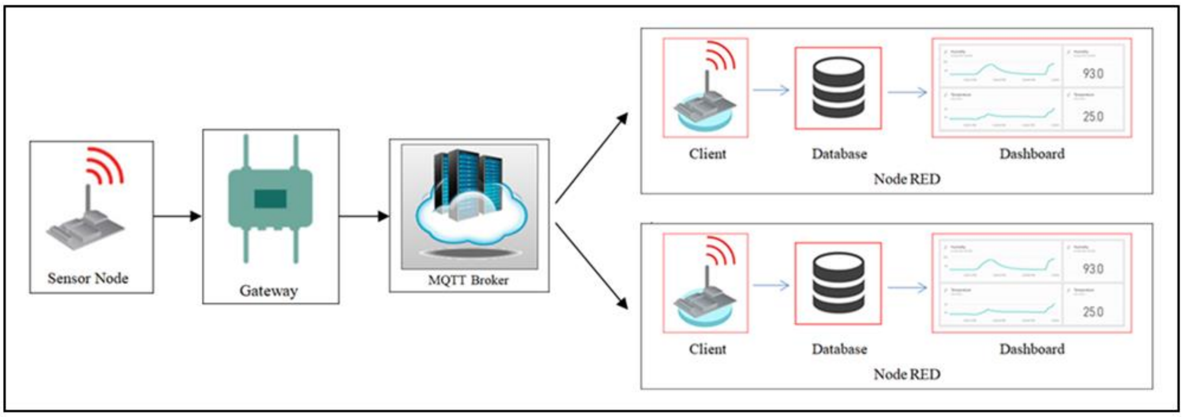


Figure 3.1 - Architectural flow.

From the Figure 3.1 - Architectural flow, the Sensor node consists of an Arduino board of the type, MKR WAN 1300 and a DHT22 Sensor. An antenna is connected to the Sensor node for the transmissions of the data. The Arduino board is used for the LoRa Connectivity. To establish a connection between the Arduino board and the LoRa Gateway, a program (sketch) must be uploaded onto the Arduino board.

A LoRa gateway directly interacts with network servers, via a backhaul relying on Transmission Control Protocol (TCP)/Internet Protocol (IP) Secure Sockets Layer (SSL) technological solutions. LoRaWAN also ensures security and provides E2E encryption capabilities. In particular, LoRaWAN can grant two different joining procedures for end-devices willing to connect to a reference gateway: Over-The-Air Activation (OTAA) and Application by Personalization (ABP). The main differences between the two regards the Network session key, generated at the beginning of the former procedure any time it is started. The latter joining procedure, instead, directly connects end-devices to the network without initiating the request, neither accepting the procedure. In fact, the end-device stores a set of dedicated parameters that allow a direct messages encryption.

In order to participate in a LoRaWAN network, an end-device must be activated. In fact, end devices are not allowed to communicate directly one with the other. All data transmissions are made possible for end devices only towards the base station and, obviously, in the reverse path. In LoRaWAN, communications are strongly centralized and hierarchical. LoRaWAN provided two ways to activate an end-device, and we make use of Over-The-Air Activation (OTAA) procedure. The activation process should give the following information to an end-device:

1. **End-device address (DevAddr)**: A 32-bit identifier of the end-device. Seven bits are used as the network identifier, and 25 bits are used as the network address of the end-device.

2. **Application identifier (AppEUI)**: A global application ID in the IEEE EUI64 address space that uniquely identifies the owner of the end-device.

3. **Application key (AppKey)**: An AES 128-bit symmetric key and is used to generate the Message Integrity Code (MIC) to ensure the integrity of the message. [15]

3.1 Hardware

In this work, a complete LoRaWAN platform is designed, developed, and experimentally tested, using open and low-cost hardware and software components, only. The number of components used has been minimized in order to obtain a small sized prototype.

Once the Arduino board is being uploaded with the program on the Serial Monitor of the Arduino IDE software, the AppKey and the AppEUI must be entered to make the connection between the LoRa gateway and the end device. When the join is successful, the DHT22 Sensor which is connected to the Arduino board starts sending the Relative Humidity and Temperature values serially and the outputs are displayed on the Serial Monitor screen. The Antenna connected to the Arduino board helps in the transmission of the data to the Lora Gateway.

The primary stages of the architectural flow mostly use hardware components, such as Arduino MKR WAN 1300, Dipole GSM Antenna and DHT22 Sensor.

3.1.1 Arduino MKR WAN 1300 board

It is a powerful board that combines the functionality of the MKR Zero and LoRa / LoRaWAN connectivity. It is a low power device. The design includes the ability to power the board using two 1.5V AA or AAA batteries or external 5V. Switching from one source to the other is done automatically. It is a good 32-bit computational power similar to the MKR ZERO board, the usual rich set of I/O interfaces, low power LoRa communication and the ease of use of the Arduino Software (IDE) for code development and programming.

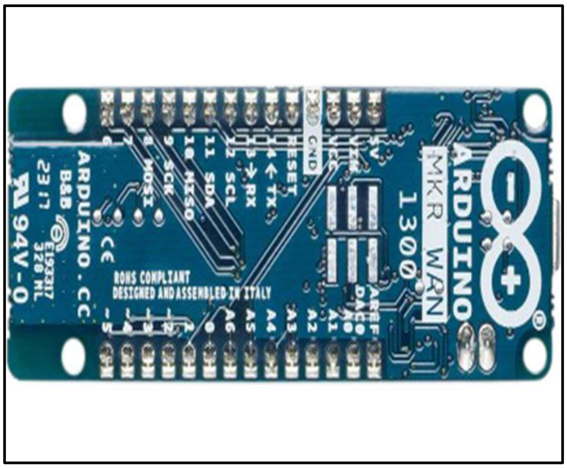


Figure 3.2 - Arduino Board MKR WAN 1300

All these features make this board the preferred choice for the emerging IoT battery-powered projects in a compact form factor. The USB port can be used to supply power (5V) to the board. The Arduino MKR WAN 1300 can run with or without the batteries connected and has limited power consumption. [16]

3.1.2 DHT22 Sensor

The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). The technical details follow:

1. 3 to 5V power and I/O.
2. 2.5mA max current use during conversion (while requesting data).
3. Good for 0-100% humidity readings with 2-5% accuracy.
4. Good for -40 to 80°C temperature readings ±0.5°C accuracy. [17]

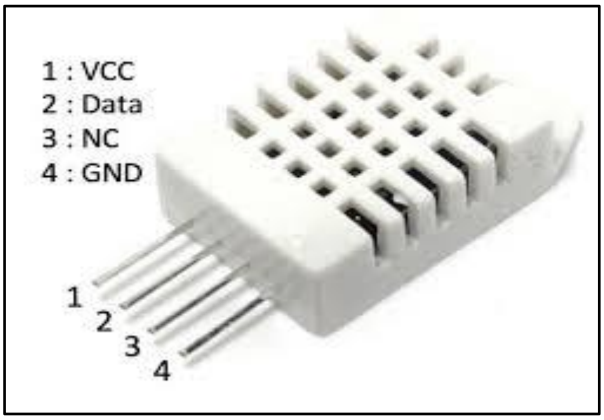


Figure 3.3 - DHT22 Sensor.

3.1.3 Dipole GSM Antenna

The Dipole Antenna for Arduino is ideal for using it with the MKRFOX1200 board. It is a GSM (850/900/1800/1900MHz) antenna that easily connects to the board via a Micro UFL connector. The antenna is connected to the Arduino board as shown in the Figure 3.4.

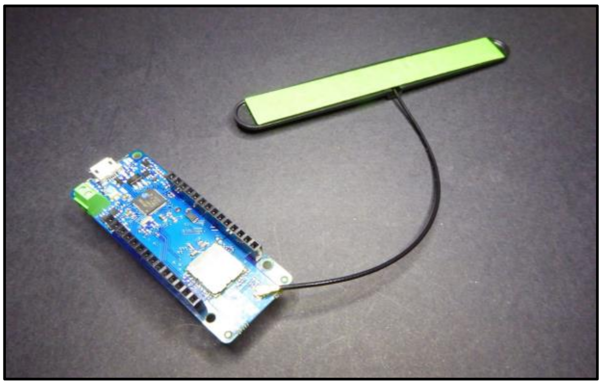


Figure 3.4 - Dipole GSM Antenna.

3.2 Cayenne LPP (Low Power Payload)

The Arduino program is written in such a way that the temperature and the humidity values are being sent serially with a delay of 60 seconds between the the values. The temperature value is sent on data channel 1 and the relative humidity value is sent on data channel 2 in the hexadecimal format.

The library Cayenne LPP is used in the Arduino program. The Cayenne Low Power Payload (LPP) provides a convenient and easy way to send data over LPWAN networks such as LoRaWAN. The Cayenne LPP is compliant with the payload size restriction, which can be lowered down to 11 bytes, and allows the device to send multiple sensor data at a time or split into several frames.

Each data type can use 1 or more bytes to send the data according to the following table.

Type	IPSO	LPP	Hex	Data size	Data Resolution per bit
Temperature Sensor	3303	103	67	2	0.1 °C Signed MSB
Humidity Sensor	3304	104	68	1	0.5 % Unsigned

Table 3.1 - Cayenne LPP (Low Power Payload)

For example, when the temperature and humidity is sensed and being sent on the LoRa network to the LoRa gateway from the end device, it is being encoded as the following:

1. Device with a temperature sensor: Payload (Hex) - 01670110. The Data Channel is 1 (01). The type is Temperature (67) and the value is 0110 = 272 ⇒ 27.2°C.
2. Device with a humidity sensor: Payload (Hex) - 026851. The Data Channel is 2 (02). The type is Humidity (68) and the value is 51 = 81 ⇒ 40.5%. [18]

3.3 LoRa Gateway

The two payloads of the temperature and the humidity are sent serially every minute. This means the values are being sent to the LoRa gateway alternatively one after another. The LoRa gateway is provided by the A2A energia. They also have the LoRa network server, where it is used to receive all the payloads. The connection establishment between the gateway and the LoRa end device being the sensor node is made by the AppKey and AppEUI parameters on the A2A smart city website. The website allows to create gateways and end nodes by entering their device details.

The concentrator, technically called gateway, allows to receive radio frequency emissions from all the sensors and counters in the area. It is positioned every 600 meters to allow perfect coverage, up to inside the buildings and underground. The gateway is connected with a high-performance fiber optic connectivity that allows communication with the centralized management system. The device will have a radio component dedicated to metering and one dedicated to smart city services.

3.4 Software

MQTT uplink topics are published for the clients to subscribe. The clients must subscribe for temperature and humidity values separately or for both. Simple subscription topics provide access to the server data streams. The topics providing simple subscription access all start with /sub/, so in order to access all the data generated by the server, a subscription on the /sub/# topic is enough.

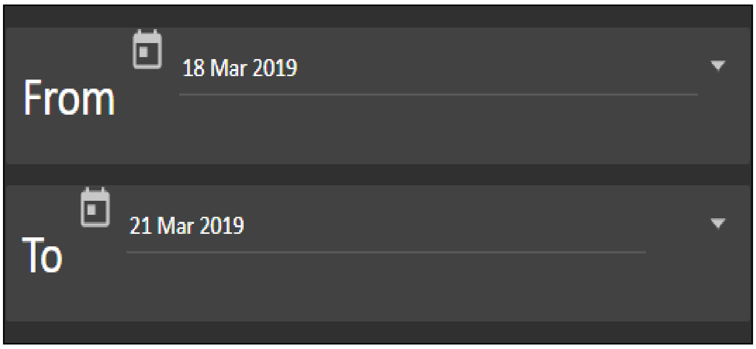
Node RED software is used to receive the payloads from the MQTT topics by subscribing to them in the software. The software is also used to decode the payload. The decoded payload gives the values of the temperature and humidity. They are stored in the databases separately as humidity temperature tables. The stored values are then being displayed on the dashboard of Node RED against the time and date as graphs. Required palettes and nodes are being downloaded in Node RED as they are not provided as default in the software. The main nodes to be downloaded are the dashboard nodes such as the date pickers, button and chart; and MySQL database nodes.

The MQTT output nodes for temperature and humidity topics are being used initially to get the payloads of them respectively. These payloads must be decoded using a couple of function nodes like split, switch and json in order to extract the payload of the payload by decoding and then converting from hexadecimal format to the decimal format. The json node helps in extracting the payload of the payload being encoded and converts between the object and string formats. A separate Function node with a piece of code helps for hexadecimal to decimal conversion.

When the conversion is being made, the decimal temperature values must be divided by 10 and the decimal humidity values must be divided by 2 as the Data Resolution per bit in the Cayenne LPP for the temperature is 0.1 °C Signed MSB and for the humidity is 0.5% Unsigned. The debug nodes are used to display the outputs of these values on the debug console.

MySQL database is identified as a user-friendly database and its respective nodes are being downloaded in Node RED. The MySQL node is used to store the values of the temperature and humidity along with the timestamp in two different tables respectively. The MySQL software is downloaded and databases and tables for the temperature and humidity are being created using the MySQL Workbench or the Command Client Line. After creating the tables in the MySQL software, the MySQL nodes in the Node RED software is used to store the values in tables.

Further ahead, to display a range of values of a required time period, we make use of the date picker nodes which is under the dashboard nodes section. The date pickers in the dashboard helps the user in selecting the dates as **From Date** and **To Date** as shown in the figure below.



From 18 Mar 2019

To 21 Mar 2019

Figure 3.5 - From Date and To Date.

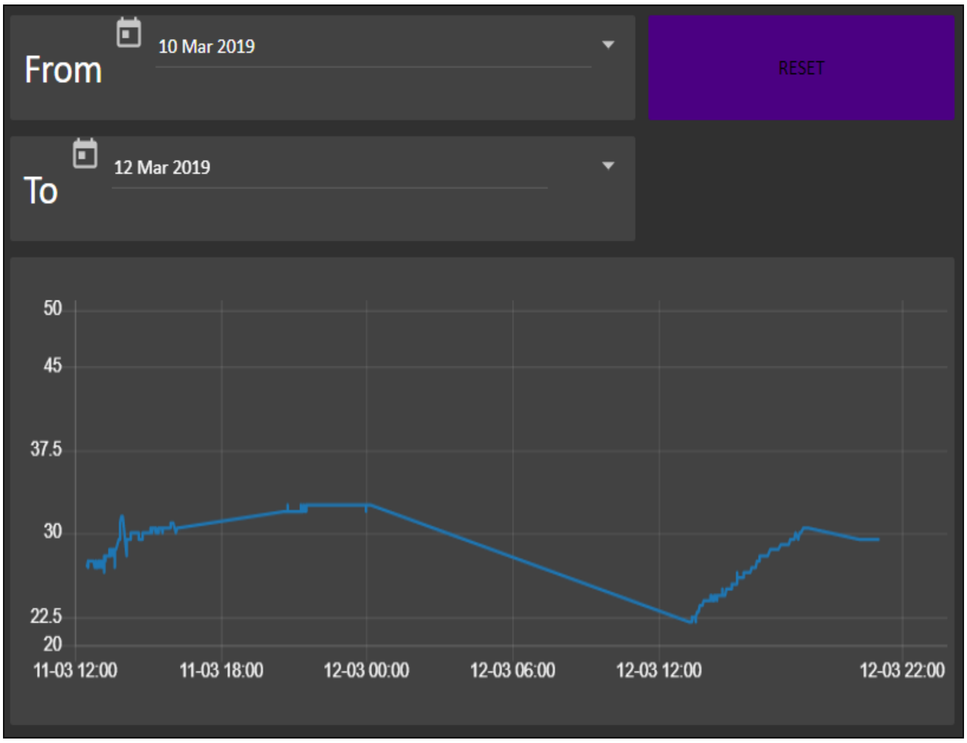
Once the user selects the From Date and the To Date pickers from the dashboard, the values belonging this range of dates will be picked from the MySQL databases of the temperature and humidity respectively. The necessary queries must be written to the MySQL, so that databases prompt the required outputs for the period selected by the user and displays them on the charts in the dashboard of Node RED. The picked dates will be in the milliseconds format and the database can only read the standard format.

Hence, the dates must be converted using the moment node where it converts date/time object or string to perfectly formatted text or date/time object. This converted format will be fed to the databases and the selected time period values will be charted on the dashboard of Node RED.

4 Demonstration and Results

4.1 Relative Humidity Results

For example, the client selects a range of dates as from the From Date and the To Date. 10-03-2019 and 12-03-2019 are the range of dates selected by the client. The sensed relative humidity values obtained from the MySQL database and plotted on the graph as follows. The x axis represents the date and time of the day. The y axis represents the values of the relative humidity.



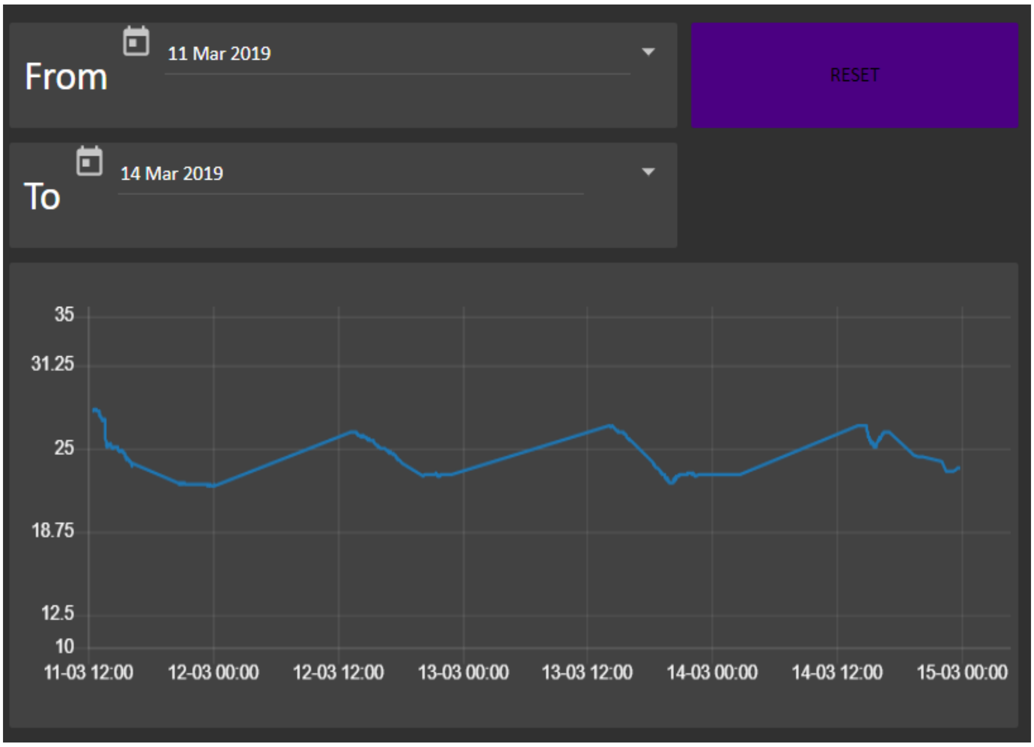
From 10 Mar 2019

To 12 Mar 2019

Figure 4.1 - Humidity vs Timestamp graph.

4.2 Temperature Results

For example, the client selects a range of dates as from the From Date and the To Date. 11-04-2019 and 14-03-2019 are the range of dates selected by the client. The sensed relative humidity values obtained from the MySQL database and plotted on the graph as follows. The x axis represents the date and time of the day. The y axis represents the values of the relative humidity.



From 11 Mar 2019

To 14 Mar 2019

Figure 4.2 - Temperature vs Timestamp graph.

5 Conclusion

In this work, an Arduino-based LoRaWAN network system was developed for environmental monitoring of a region using DHT22 sensor. The system uses a Node RED topology in which a variable number of functional nodes are used for the virtual connections between the clients and the databases where LoRa gateway handles all the requests using MQTT broker. Many different subscriptions can be made by using these flows and can be used in the numerous applications in various fields. All the environmental conditions for any selected period of time are picked from the database and plotted in the dashboard.

5.1 Future enhancements

1. GPS module can be attached to the Arduino board to monitor the location of the sensor. This could be a location-based application.
2. Sophisticated design of the Node RED dashboard could be done to display the location of the sensor on the map along with the plots of the temperature and humidity graphs of the desired location.
3. Filtering and distribution of messages to subscribers can also be done by the method, Multicasting.
4. A mobile application could be designed for Android/iOS/Windows platforms so that the client could get instant updates of the temperature and humidity values based on the desired location.

5.2 Future vision of IoT

One future vision for the IoT is the Web of Things, proposes the use of web standards to fully integrate smart objects into the World Wide Web. Using web technologies, one can make it easier for developers to build applications using smart objects and existing web protocols can more easily enable the interoperability and communication of different devices.

Another future vision that involves integrating even more devices into the IoT is the Internet of Nano-Things, described as the interconnection of Nano scale devices with communication networks and the Internet. While these devices are proposed to communicate through electromagnetic communication, there are numerous technical challenges that must be overcome before the idea becomes feasible.

Smartness to intelligence: Early studies of IoT usually use rule-based mechanisms to provide smart services, such as turning on the light because the sensor detected someone is entering the room. Nowadays, the research trends have been employing much more intelligent mechanisms to replace the rule-based mechanisms to make things think as much as the way humans think. For this reason, the modern computational intelligence will also be employed in Future IoT (FIoT), which will not be designed and developed on the current internet environment; rather, it will be designed and developed on an environment that integrates internet, cloud computing, and smart grid.

5.3 Application and future of LoRa WAN technology

5.3.1 Smart City

LoRa WAN will be inevitable technology in future smart city applications together with Internet of Things like:

1. Smart lighting
2. Air quality and pollution monitoring
3. Smart parking and vehicle management
4. Facilities and infrastructure management
5. Fire detection and management
6. Waste management

5.3.2 Industrial Applications

LoRa WAN is suitable for wide range of industrial applications.

1. Radiation and leak detection
2. Item location and tracking
3. Shipping and transportation

5.3.3 Smart home applications

In future, billions of smart devices and home appliances will be connected to internet.

1. Enhanced home security
2. Home automation for IoT enables smart appliances

5.3.4 Healthcare

LoRa is one of the best solutions for connecting healthcare devices efficiently

1. Health monitoring devices and management
2. Wearable technology

5.3.5 Agriculture

LoRa technology can be used in smart agriculture and farming applications.

1. Smart farming and livestock management
2. Temperature and moisture monitoring
3. Water level sensors and irrigation control

References

1. A Wi-Fi - enabled indoor air quality monitoring and control system. Published in Control & Automation (ICCA), 2017 13th IEEE International Conference.
2. A low-power real-time air quality monitoring system using LPWAN based on LoRa. Published in Solid-State and Integrated Circuit Technology (ICSICT), 2016 13th IEEE International Conference.
3. IoT enabled proactive indoor air quality monitoring system for sustainable health management. Published in Computing and Communications Technologies (ICCCT), 2017 2nd International Conference.
4. A wireless system for indoor air quality monitoring. Published in Industrial Electronics Society. IECON 2016 - 42nd Annual Conference of the IEEE.
5. Polluino: An efficient cloud-based management of IoT devices for air quality monitoring. Published in Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum.
6. An embedded system model for air quality monitoring. Published in Computing for Sustainable Global Development (INDIA Com), 2016 3rd International Conference.
7. A real-time ambient air quality monitoring wireless sensor network for schools in smart cities. Published in Smart Cities Conference (ISC2), 2015 IEEE First International Conference.
8. A smart sensor system for air quality monitoring and massive data collection. Published in Information and Communication Technology Convergence (ICTC), 2015 International Conference.
9. Development of a Low-Cost Internet-of-Things System for Monitoring Soil Water Potential Using Watermark 200SS Sensors. Published by Advances in Internet of Things, Vol.7 No.3, 2017.
10. Smart water grid management using LPWAN IoT technology by Saravanan, Arindam Das; Vishakh Iyer.
11. On fast prototyping LoRaWAN: a cheap and open platform for daily experiments.
12. https://nodered.org/
13. https://intel.github.io/dps-for-iot/
14. https://en.wikipedia.org/wiki/Arduino_IDE
15. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things.
16. https://store.arduino.cc/mkr-wan-1300
17. https://www.adafruit.com/product/385
18. https://community.mydevices.com/t/cayenne-lpp-2-0/7510