

IoT enabled sensor node: a tutorial paper

Alice James*, Avishkar Seth and Subhas C. Mukhopadhyay

Macquarie University, NSW, 2113,
Sydney, Australia.

*E-mails: alice.james@students.mq.edu.au, avishkar.seth@students.mq.edu.au

This paper was edited by
Nasrin Afsarimanesh.

Received for publication
July 26, 2020.

Abstract

A descriptive guide to develop smart sensor nodes for Internet of Things (IoT) application has been presented in the paper. The current technology of low-power wide area network (LPWAN) which transmits data with low power consumption over a wide area has been illustrated in the scope of this project. An overview of the transition of IoT devices from a limited range personal area network to wide area network is justified. A sensor node has been developed and tested to validate the implementation of a LoRa network, which can be an exemplar for sensor nodes for various applications.

Keywords

Wireless sensor network application, IoT, LoRa, LPP, LPWAN, Arduino MKR1300, OTAA protocol, The Things Network (TTN), Adafruit IO.

The wireless sensor network applications have evolved through various applications in multiple domains such as healthcare, smart cities Santos and Ferreira (2019), agriculture, industries, etc. Different parameters can be assessed and monitored with sensor nodes deployed across a given area. A wireless sensor node will comprise of the sensor, a controller/processor, battery to power the node, and the transmission protocol module. This sensor network that contains an array of these nodes is capable of providing accurate information from the given application.

With the growth in technology in IoT, it is now possible for various day to day items that are interlinked to share informative data. This has relatively improved the work efficiencies within the domain of automation of processes and work to be completed. The requirement for many devices connected on multiple levels is strong communication that can sustain on a single gateway with fewer or no handovers. The LoRaWAN technology is the best fit for cases where the data need to be transmitted (Kazakov et al., 2020) over a long-range with low power consumption. This allows the IoT devices to be powered by batteries that can go on for multiple years. The technology used in LoRaWAN is of LPWAN (low power wide area

networks). The LPWAN allows the end-user node to communicate with the gateway without multi-hops with other nodes in the network. Therefore, the LPWAN provides the radio coverage for a large area similar to the WiFi (Afsarimanesh et al., 2018; Nag et al., 2019) used for consumer networking (Augustin et al., 2016).

The technologies such as Bluetooth, ZigBee (Singaravelan and Kowsalya, 2016) have been previously to transmit data and intermediary communication between the sensor nodes/devices. The Zigbee nodes also offer a low cost and low power solution. However, this technology is limited in range. The sensor nodes have to be in close proximities. Unlike the LoRa, in ZigBee network uses the mesh topology that allows intercommunication between the nodes, to transmit data. There is a master device that coordinates with the other nodes in range thus each of these nodes act as a repeater. This can cause a delay in delivering the data and the operating environment can vary for multiple criteria's such as low reception, etc. It also does not provide a completely secure channel for transmission of data.

In a heterogeneous environment, IoT devices prompt minor variations over small distances. The devices using LoRa technology prompt a low-cost hardware implementation that consumes low power

that can monitor various environments. In the scope of this research, an Arduino designed MKR1300 has been used that has a configurable LoRa module embedded. This microcontroller can be used to configure sensor nodes that will use LoRa to transmit data and will consume less power. Various sensor networks can be designed using this technology such as Landslide Monitoring (Yueshun and Wei, 2013) flood monitoring, pedestrian counting system, attendance recording system, air quality monitoring, etc.

As it has been known, LoRa operates in the unlicensed band and is compatible with applications that are indoor Haxhibeqiri et al. (2017) and outdoor. LoRa has a secure channel of transmission that makes it economically feasible and a flexible network. The data generated from these sensor nodes can be further used in different analytics platforms that implement machine learning algorithm. This can be further used to develop self-driven systems. Therefore, it is an essential skill to learn the basics of LoRa transmissions and how it can be integrated into a sensor node. The applications of various IoT applications can transition into LPWAN as it is only growing day by day.

In this paper, a sensor node is developed using the MKR1300 microcontroller board with the integration of three sensor devices to transmit data to the cloud using LoRaWAN transmission. The second section elaborates on similar work in the development of sensor networks using LoRa in LPWAN. The different transmission protocols that serve as an alternative for LoRa have been discussed in the third section and the fourth section explains why LPWAN is the current trend for IoT devices in the market.

The main setup of the sensor node that was designed as a sub-circuit for MPPT monitoring of the solar panel (Visconti et al., 2016) that powers the microcontroller and the connections established between the sensors have been illustrated in the fifth and sixth sections, with a detailed guide that can be translated into a similar architecture for a different application. The results from the developed sensor node are highlighted in the seventh section.

Literature review

Kumar and Mallick (2018) had proposed different architectures layers for IoT. In an IoT system, the hardware and software components function cohesively. It processes the data and works to store it. To propose an effective IoT system, the formation of the architecture layer is essential. One of the prominent application layers highlighted in the study of Kumar and Mallick (2018) is three layer, five layer, six layer, seven layer, and FOG layer. Amongst these,

the most flexible and higher ability to be scaled is the FOG based. In this architecture layer, the IoT has been provided with a cloud-centric architecture that can be used for data processing. Each of the architecture is dependent on the network layer to ensure smooth transmission of data to enable the IoT application.

There are various applications of IoT that have been attempted using LoRaWAN and other transmission protocols that have been illustrated in the reviewed literature. The different domains of application are smart homes (Pirbhulal et al., 2017), farming (Jawad et al., 2017; Spachos, 2020; Davcev et al., 2018), medical facilities (Petäjäjärvi et al., 2016), cities (Magrin et al., 2017; Pasolini et al., 2018), and industries (Navarro-Ortiz et al., 2018). Each of the research aims to develop a low-cost device and gateways. These nodes are developed using the Arduino and Raspberry pi. In the scope of this review, the application of the smart city (Akhter et al., 2019), smart farm, and smart industrial application have been highlighted.

The aspects of smart city applications that have been analyzed in the study of Pasolini et al. (2018) are smart lighting and building monitoring. Pasolini et al. (2018) state that the proposed solution using mesh technology for smart lighting has earned benefits for low power and cost. The claimed maximum coverage for the LoRa transmission has been between 1 and 2 km for a clustered city environment. Thus, it is concluded a greater number of such nodes are required in close proximity to acquire better coverage.

The application of farming, Germani et al., 2019 and Davcev et al. (2018) developed a scalable node using the LoRaWAN platform. The proposed platform is accustomed to monitor and accumulate data for different cases. Therefore, the LoRa hardware is chosen to perform in such a heterogeneous network. Another challenge that the LoRa technology overcomes is the direct transmission to the gateway without the intermediary network hops.

In an industry application, few authors had proposed a device that acts as a range extender while behaving as a regular node with connectivity to the gateway. The Node designed and tested by the authors was used to transmit imitations at a fixed interval of the received message. The Industrial IoT (IIoT) will be the future since it offers a more efficient operative environment. This application can be made flexible and scaled with the integration of wireless solutions.

The transmission network in the LoRa architecture is of great importance, in context to the growing smart devices and IoT systems. Huh and Kim (2019) have provided an overview of the LoRa mesh networks in an ISM band. In this mesh architecture, there is no

compulsive hierarchy. This allows transmission to the gateway using the best self-configured route (Huh and Kim, 2019). Huh and Kim (2019) proposed solutions to overcome the hurdles faced while implementing the mesh architecture for LoRa, due to the variation in the transmission powers. The proposed solution offers a better coverage, while eliminating the use of cell-planning with gateways and deploying the sensor nodes in a simple and flexible pattern (Huh and Kim, 2019).

In the study of Navarro-Ortiz et al. (2018), the authors have elaborated about the security in LoRaWAN. The authors have summarized the three prime features of the LoRaWAN security to be confidentiality, mutual authentication, and integrity protection. The two activation methods used in LoRa are OTAA (over the air activation) Ruckebusch et al. (2018) and ABP (activation by personalization), these have been elaborated further in the implementation of this project. Shahjalal et al. (2020) implemented LoRa for a smart home environment and generated an AI-based dataflow for the IoT server. The tasks that were performed by the AI powered smart home system were predictive maintenance, maximum efficiency, improved security, and enhanced services. One of the challenges they discovered within a LoRa-based system was interoperability that can be worked on further.

Traditional transmission protocols in IoT devices

The primary technology used for digital transmission was Bluetooth Hughes et al. (2015). Collotta et al. (2018) have reviewed the evolving technology of Bluetooth. The technical comparisons between the Bluetooth versions and the traditional wireless standards have been provided in the research (Collotta et al., 2018).

According to Collotta et al. (2018), comparing the latest available Bluetooth version and ZigBee and HaLow, the best choice is Bluetooth 5. However, the protocol varies for different applications. For applications within a small range that does not need a transmission over a long distance, HaLow offers 1,000 meters of effective transmission. Though Bluetooth offers two types of Network topologies, i.e. Star-bus as well as Mesh. Also, Bluetooth offers up to 255 bytes of message size compared to ZigBee and HaLow that offer 100 bytes.

Collotta et al. (2018) have conducted an extensive comparison between ZigBee and Bluetooth 5 on a real testbed. The resultant comparisons proved that

Bluetooth 5 provides a sustainable battery life compared to ZigBee as it had very low power consumption for indoor as well as outdoor experiments. The transmission using Bluetooth was also proved to be faster compared to ZigBee.

In an IoT system, the operability range, transmission speed, and broadcasting of data are key performance parameters. The interconnectivity between the devices is increasing day by day, also the requirement to transition to a system capable to transmit to a longer range is rising.

Transition to LPWAN for IoT devices

As mentioned earlier, in an IoT architecture it is essential to select the protocol technology since it impacts the system performance in terms of efficiency in power consumption, consistency, and scalability (Choi et al., 2019). As observed earlier the message size is limited to network protocol chosen for the IoT system.

Therefore, Lysogor et al. (2018) have proposed four criteria to select the most efficient protocol for a heterogeneous IoT network: (i) efficient power consumption, (ii) easy to use in remote locations, (iii) sufficient network coverage, and (iv) utilization of an unlicensed frequency band for operations. To suffice the following conditions, the authors concluded that LPWAN technologies are equivalent to these needs. The prime features of an LPWAN protocol are it supports wide area network that enables a wide area network coverage for the devices. This ranges up to 10km. Another feature is the low power consumption which makes it an energy-efficient solution for IoT applications.

The commonly used protocols for LPWAN Prando et al. (2019) are LoRa and NB-IoT. The LoRa utilizes unlicensed frequencies of different bands, e.g. AS923 for Asia. Each of these frequencies is specific to the operating countries. The LoRa module does not consume a lot of power Alahi et al. (2018), therefore amongst the various LPWAN solutions, LoRa is the most ideal solution for network coverage and energy efficiency, confirmed by authors in Lysogor et al. (2018).

Another network protocol in the LPWAN category is the NB-IoT (narrow-band IoT). The concept of the NB-IoT is functional utilizing the LTE standard. However, it does not possess any network roaming capabilities and requires a prevailing cellular network to operate. Therefore, it does not serve a purpose in remote areas, despite the range it offers. Mekki et al. (2019) offer an extensive comparison between the technologies implementing LPWAN such as Sigfox, LoRaWAN, and NB-IoT, which aids the selection for large-scale IoT applications.

Development of the sensor node

The following section provides a guided method to develop a sensor node implementing the LoRa architecture. This sensor network has been as a project for the wireless sensor network application unit. The challenge of the project is to develop a sensor node for a smart system that transmits data using LoRa.

The section is further subdivided into different aspects of developing a sensor node, followed by the next section that determines the different stages of the LoRa architecture that has been implemented.

Project description

The project is a focussed the IoT enabled node that estimates the pedestrian count using an infrared sensor of a given area along with the sensor reading of temperature and humidity. This system is critical as it is a very integral data and it can be a subsystem of the applications of smart cities, smart homes, etc.

The sensor node can be powered using a solar panel. This is a sustainable solution for outdoor IoT nodes, also, the use of an LPWAN topology, i.e. LoRa also optimizes the battery consumption of the entire node.

The sensor node comprises two sensors with three sensor readings data that can be used to monitor the temperature/humidity as well the pedestrian count.

Within the scope of this tutorial, the guided method is provided for the circuit, i.e. the sensor node and the connectivity to cloud to compute data. The sensor node has been developed using the LoRa module that transmits data over a secure channel and stores it on a cloud server that enables further data analysis.

Functional requirements

The functional requirements of the project comprise of the hardware components required and used to build the circuit and the software applications that have been used to store and analyze the data.

This section elaborates on the features of the component and reason for its selection for the chosen project.

Hardware requirements

The hardware requirements for the sensor node are as follows.

MKR1300 This was the Arduino board used to configure and connect the sensors. It provides 12 PWM pins, 8 digital I/O pins, 7 Analog Input pins, and

1 Analog Output pin. It has a flash memory of 256 KB and provides clock speed of 32.768 kHz (RTC), 48 MHz, the board requires a 5V external supply.

Antenna The Antenna was provided with the MKR1300. This is connected to the device to connect to the LoRa gateway to enable transmission of data. The Antenna power is 2 dB and the carrier frequency can be 433/868/915 MHz with operating regions across the globe.

Adafruit PIR sensor This was the sensor selected to monitor the count of pedestrian data. The Passive Infrared sensor utilizes the IR technology to determine when a human pass across the sensor. The sensor is bound within the Fresnel lens to increase the detecting angle of the sensor. When the sensor is placed at an accurate angle, it can be used to determine pedestrians across a wide range. The sensor has three pins, VCC, GND, and data pin to transmit the output reading using the digital I/O pin.

AM2302 sensor The AM2302 sensor provides the values of the temperature and humidity of the environment around it. This sensor in comparison with the other competitors such as SI7021 consumes lower power and offers higher accuracy with quick response time. Similar to the infrared sensor, the AM2302 sensor has three pins, VCC, GND, and data pin that are connected to the digital I/O pin of the microcontroller.

Power bank The power bank used in this project was an 11 mAh power bank, to power the MKR1300 and the sensors connected to it. The power bank provides long-term execution of the MKR1300, thus allowing the collection of data without any interruptions.

Miscellaneous

In the miscellaneous, a breadboard was used to assemble the node with the microcontroller and sensors. These were connected using jumper wires.

Software requirements

The software requirements for the operation of the sensor node are as follows.

Arduino IDE

The Arduino IDE software was used to program the MKR1300. The software provides predefined libraries to initiate the MKR1300 board and can be configured

with libraries to connect to the gateway and transmit the data as a payload.

The Things Network

This was the gateway used to transmit data using the MKR1300. The Things Network (TTN) provides a secure platform to store and transfer the payload to the cloud server. The TTN provides a console to configure the gateway where the application and device are registered.

IFTTT API

To connect the gateway to the server, the IFTTT was used. If-This-Then-That (IFTTT) is an API that acts as a middleware to transmit the payload stored in TTN to the cloud server. This IFTTT is affiliated with an array of different software, thus it was selected as an API for this sensor node.

Adafruit IO cloud server

The Adafruit IO was the cloud server that was used to store and analyze the data. The sensor data are transmitted as a feed and it can be viewed on the dashboard. Each of the data can be downloaded and used for further analysis too.

Hardware setup

This section explains explicitly the connection setup for the sensor node. The hardware components have been assessed with the provided datasheet.

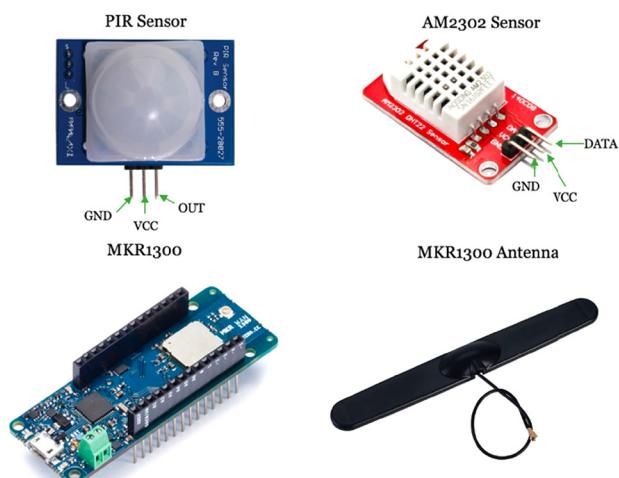


Figure 1: Components used for hardware assembly of the IoT node.

Figure 1 illustrates the components used in the hardware assembly of the sensor node, i.e. the PIR sensor, AM2302 sensor, MKR1300 board, and Antenna.

These components were assembled using a breadboard, Figure 2 shows the circuit designed on the Fritzing software, which can be designed further into a PCB. There are different software available, wherein, the circuit can be tested, before assembling the actual hardware. This allows the hardware to be protected from any unnecessary damage caused due to incorrect connections.

Arduino IDE implementation

The programming of the sensor node utilizes the Arduino IDE platform. The command execution can be viewed on the serial monitor. This section focusses on the work of the Arduino IDE.

Initialization and setup

To commence the initialization, the Arduino IDE has to be installed on the PC. The latest Arduino version of 1.8.13 is compatible with the latest libraries and microcontroller boards. The open-source software is also available online.

There were two main initialization steps involved in designing this network. The first step was to register the microcontroller to enable it to connect with the gateway. The second setup was the final configuration to send the data over a secure channel using the generated session keys for the registered device.

The first setup requires an account registration with TTN and update of details of location, etc.

Once the user had signed in into the TTN, on the console page, the gateway must be registered. The registration of the gateway requires details of the frequency plan based on the region and the router that has been used. Figure 3 is the console of the TTN website.

Figures 4 and 5 illustrate the registration and activation for the gateway and applications that will utilize the registered gateway. The frequency plan used in this project is AS923. This is the frequency band used in Australia, along with AU915.

This was the very first step to configure the device once it is set up physically. Once the console page has generated the EUI for the application. Within this window, there was an option to register the device. This is where the MKR1300 was registered. Figure 6 shows the configuration and initialization process window and registration with TTN.

Figure 7 shows the first configuration for the registered microcontroller. The device communication

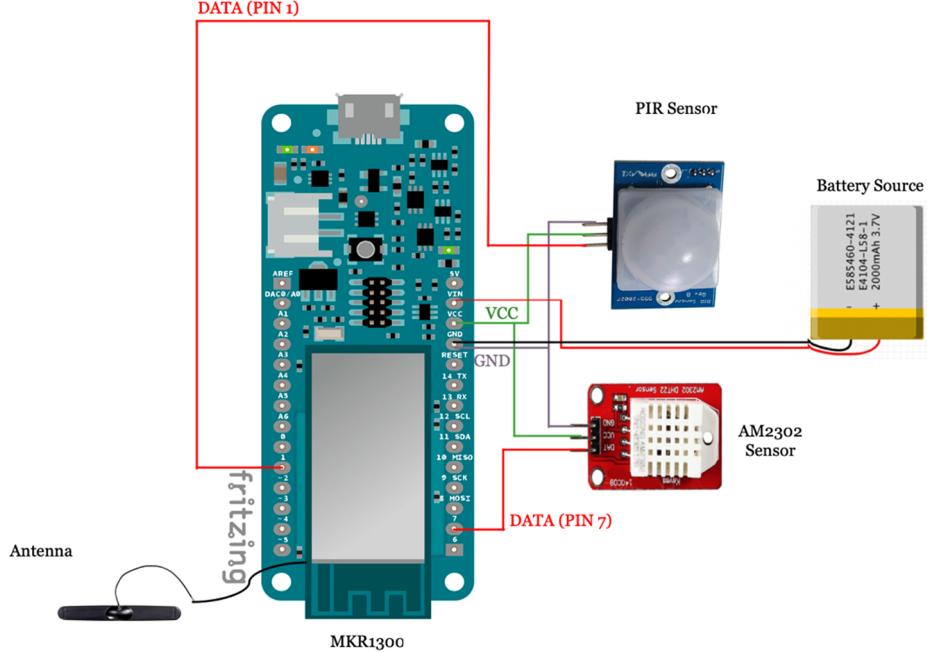


Figure 2: Circuit diagram of the final IoT node assembled on Fritzing.

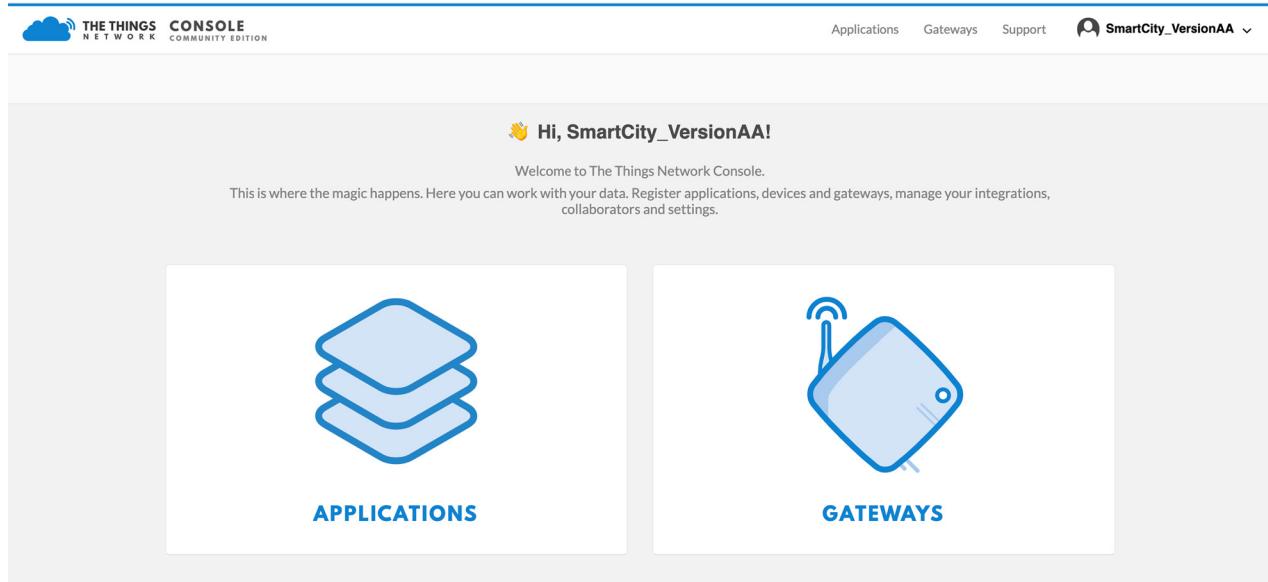


Figure 3: The console page after the registration with TTN.

setup is visible through the serial monitor. While the first configuration, the activation method is selected and the device and application EUI is entered. This enables the device to be registered and communicate without any registration.

Once these procedures are completed, the device and sensor node are ready to transmit sensor readings at the regular intervals that have been designed in the code, which is further elaborated the next section.

REGISTER GATEWAY

Gateway ID
A unique, human-readable identifier for your gateway. It can be anything so be creative!

I'm using the legacy packet forwarder
Select this if you are using the legacy [Semtech packet forwarder](#).

Description
A human-readable description of the gateway

Frequency Plan
The [frequency plan](#) this gateway will use

Router
The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway.

Location
The exact location of your gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.

lat	0.00000000
lng	0.00000000

Map data ©2020 Google Terms of Use

Antenna Placement
The placement of the gateway antenna

Figure 4: The form for selecting and setting up a gateway on the TTN console.

Libraries used for the final assembly circuit

The following section depicts the libraries necessary to configure the sensors to compute the readings and transmit them across the network. Figure 8 has highlighted the section to select while downloading the library, Figure 9 shows the download interface.

The libraries used in the scope of this project have been enlisted below.

MKRWAN This library was to configure the microcontroller board. The library triggers the functions of a low powered system and enables the LoRa configuration module of the board.

DHT This library was to configure the AM2302 sensor that gives the data for temperature as well as humidity. This library can configure all versions of the DHT sensor, i.e. DHT11 and DHT22, AM2302.

The sensor readings were taken from the library using this syntax, as shown in Figure 10.

The sensor readings were stored in integer values declared as ‘temperature’ and ‘humidity’; the library also performs an error check to see if the value was ‘isnan’, i.e. is not a number.

Cayenne LPP The following library was used to transmit data for the LoRaWAN board, i.e. the MKR1300. This library has been conforming to the restriction of the payload. It allows the users to

ADD APPLICATION

Application ID
The unique identifier of your application on the network

Description
A human readable description of your new app

Eg. My sensor network application ✓

Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.

EUI issued by The Things Network

Handler registration
Select the handler you want to register this application to

ttn-handler-asia-se ✓

Figure 5: The form for registering an application on the TTN console.

APPLICATION OVERVIEW

[documentation](#)

Application ID group2

Description PV Solar MPPT Monitoring System

Created last month

Handler ttu-handler-asia-se

APPLICATION EUIS

[manage euis](#)

<input type="button" value="↔"/>	<input type="button" value="≡"/>	70 B3 D5 7E D0 01 B5 ED	<input type="button" value="☰"/>
----------------------------------	----------------------------------	-------------------------	----------------------------------

Applications > group2 > Devices

[Overview](#) [Devices](#) [Payload Formats](#) [Integrations](#) [Data](#) [Settings](#)

DEVICES

[register device](#)

< >

1 – 1 / 1

mppt_mkrwan_1300	A8 61 0A 30 37 46 86 01	●
------------------	-------------------------	--------------------------------------

Figure 6: The overview page post registration of the microcontroller for the initialization of the session keys with TTN.

```

FirstConfiguration §
/*
First Configuration
This sketch demonstrates the usage of MKR WAN 1300 LoRa module.
This example code is in the public domain.
*/
#include <MKRWAN.h>

LoRaModem modem;

// Uncomment if using the Murata chip as a module
// LoRaModem modem(Serial1);

String appEui;
String appKey;
String devAddr;
String nwkSKey;
String appSKey;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    while (!Serial);
    Serial.println("Welcome to MKRWAN1300/1310 first configuration sketch");
    Serial.println("Register to your favourite LoRa network and we are ready to go!");
    // change this to your regional band (eg. US915, AS923, ...)
    if (!modem.begin(AS923)) {
        Serial.println("Failed to start module");
        while (1) {}
    };
    Serial.print("Your module version is: ");
    Serial.println(modem.version());
    Serial.print("Your device EUI is: ");
    Serial.println(modem.deviceEUI());
}

```

Figure 7: First configuration code snippet on Arduino IDE with LoRa gateway for MKR1300.

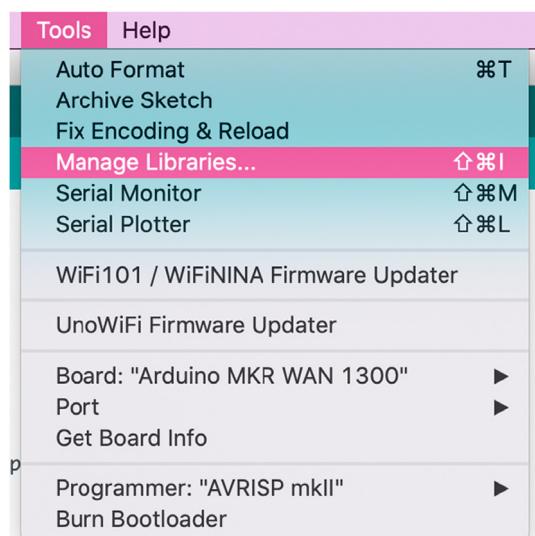


Figure 8: How to download new libraries on the Arduino IDE platform.

configure the sensor data into individual frames. To allow such customization, every sensor is assigned with two bytes the data channel and type. Figure 11 shows the configuration for this sensor node.

This allows the data to be encoded to the payload format and it aids the end-user to decode the payload function on the gateway console. The transmission of this message has been enabled using the functionalities of the Cayenne library as shown in Figure 12.

LoRa architecture implementation

The following section aims to highlight the overall implementation of the LoRa architecture in the sensor node developed in this project. Figure 13 shows a generalized architecture for the LoRa platform and the overall architecture of data transmission, storage, and analysis.

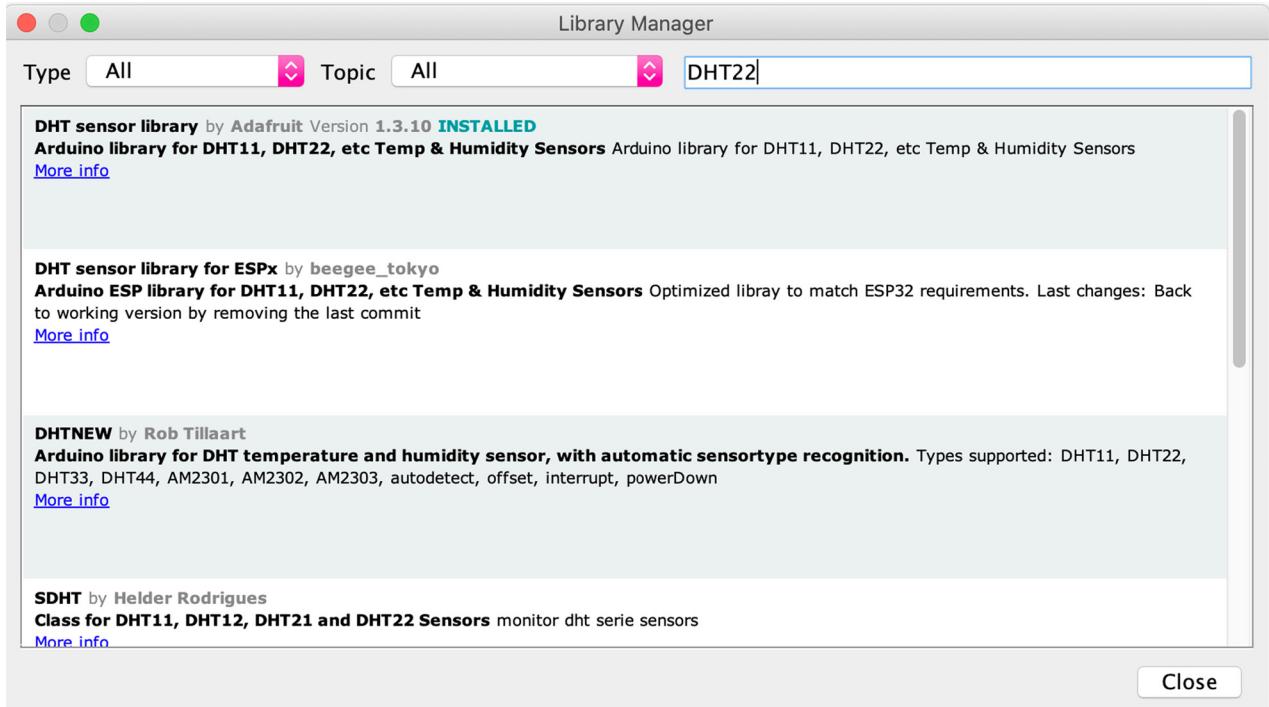


Figure 9: Library download and upgrade manager on Arduino IDE.

```
// read values from DHT
int temperature = dht.readTemperature();
int humidity = dht.readHumidity();

// Check if any reads failed and exit early (to try again)
if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    delay(1000);
    return;
}

Serial.println(dht.readTemperature());
Serial.println(dht.readHumidity());
```

Figure 10: DHT library function code snippet on Arduino IDE to enable sensor reading.

```
// Prepare Cayenne LPP
lpp.reset();
lpp.addTemperature(1, temperature);
lpp.addRelativeHumidity(2, humidity);
lpp.addDigitalOutput(3, pirState);
```

Figure 11: Cayenne LPP code snippet on Arduino IDE assigning payload bytes.

The implementation of the architecture is depicted in the following steps.

Sensor node final assembly

This section illustrates the final physical assembly of the circuit using the components mentioned earlier. Figure 14 shows the final assembly of the sensor node using LoRa technology.

```
// Send the data
modem.beginPacket();
modem.write(lpp.getBuffer(), lpp.getSize());
int err = modem.endPacket(true);
if (err > 0) {
    Serial.println("Message sent.");
} else {
    Serial.println("Error sending data.");
}
```

Figure 12: Transmission syntax for Cayenne LPP format code snippet on Arduino IDE.

This node is programmed on Arduino IDE to transmit the details of pedestrian movement and temperature and humidity data within the interval of 10 min.

Payload encoder and TTN (gateway)

The earlier section describes the initialization required for the device that registers information on the TTN console. In the Arduino code, it is essential to first initialize the secure transmission protocol. To determine that the data are not received by another user, and it

is secure, it has been important to have an activation method, with registered node, i.e. microcontroller.

In this project, the OTAA (over the air activation) method has been implemented. The OTAA method has been compared against ABP (activation by personalization). The difference between the methods are, in the ABP, the NWksKey and AppSKey have been pre-registered with the network and were pre-programmed into the device. However, in the OTAA, the NWksKey and AppSKey are activated using a join request, which can be stored and used to communicate, but if the keys are lost, the network may expire and are required to re-join. The OTAA method is more secure to transmit data in this scenario. This has been due to the predefining of the AppSKey, AppEUI, and DevEUI in the system. These session keys are generated only when it is required, therefore there is no threat to intrusion in the network. The activation method is determined in the TTN console, as shown in Figure 15.

Once the activation method was determined, the microcontroller could now be programmed with the frequency band for the region, i.e. AS923. Figure 16 shows the activation and connectivity syntax for this project.

These commands prompt the serial monitor to notify whether a secure connection has been established or not.

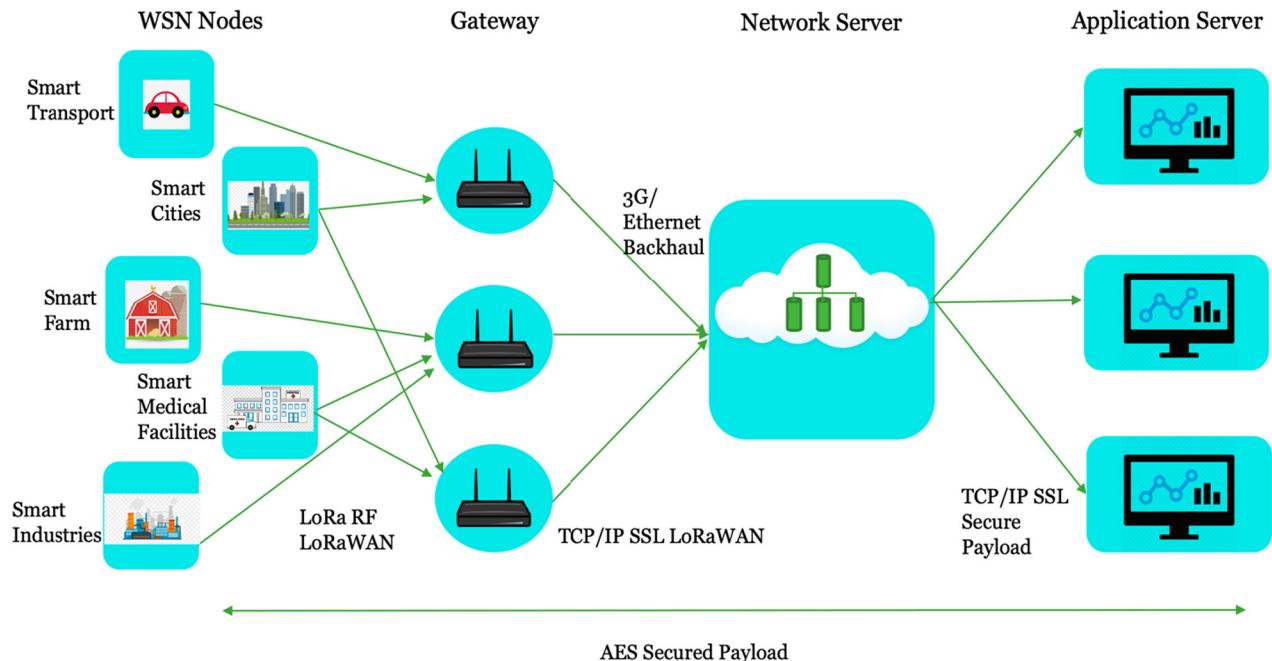


Figure 13: LoRa architecture for sensor nodes in an IoT system.

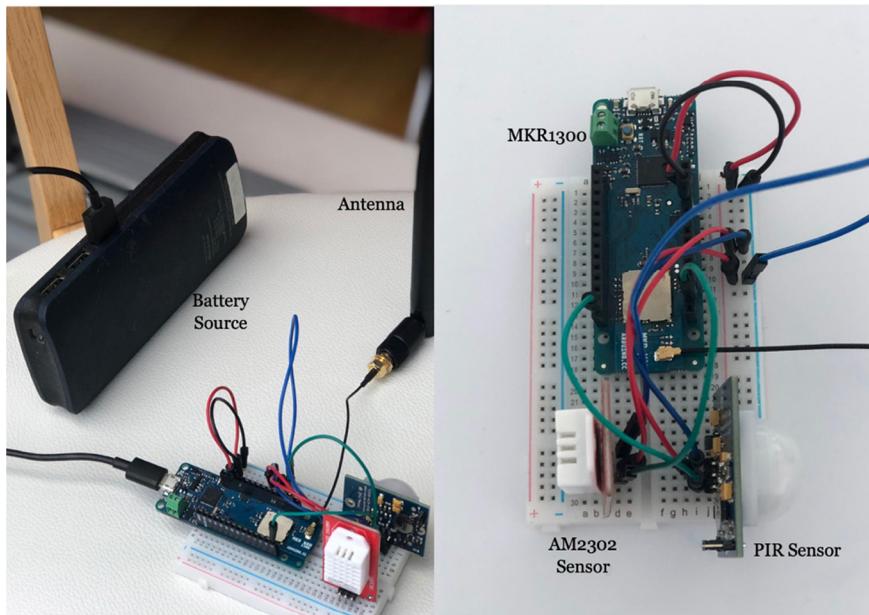


Figure 14: Sensor node final hardware assembly using MKR1300.

A screenshot of the TTN (The Things Network) console showing the device registration process. The top navigation bar shows 'Applications > group2 > Devices > mppt_mkrwan_1300'. The main form fields are filled with the following values:

- Application ID:** group
- Device ID:** mppt_mkrwan_1300
- Activation Method:** OTAA
- Device EUI:** (redacted)
- Application EUI:** (redacted)
- App Key:** (redacted)
- Device Address:** (redacted)
- Network Session Key:** (redacted)
- App Session Key:** (redacted)

The status at the bottom indicates 'Status • 6 days ago'.

Figure 15: OTAA selection in the TTN console, while registration of the microcontroller device.

```

// change this to your regional band (eg. US915, AS923, ...)
if (!modem.begin(AS923)) {
    Serial.println("Failed to start LoRa");
    while (1) {}
};

Serial.print("Your module version is: ");
Serial.println(modem.version());
Serial.print("Your device EUI is: ");
Serial.println(modem.deviceEUI());
Serial.println("Attempting to join LoRaWAN network with OTAA");
int connected = modem.joinOTAA(appEui, appKey);
if (!connected) {
    Serial.println("Something went wrong; are you indoor? Move near a window and retry");
    while (1) {}
}
Serial.println("Join Successful");

modem.minPollInterval(60);

```

Figure 16: LoRa connectivity for the sensor node using TTN; the code snippet on Arduino IDE.

Once the connection was established, the data are converted into the payload format for Cayenne LPP and transmitted using the code given in the previous section of Arduino IDE implementation.

IFTTT interfacing (network server)

This was the network server section of the architecture. The IFTTT API was integrated to both with TTN as well as the cloud server (Adafruit).

Figure 17A shows the initialization of the IFTTT with the application set on the TTN console. This initialization has been further utilized to transfer the data from the payload format of TTN to a secured cloud storage server. In the IFTTT, the Webhooks API was used to transfer the decoded payload from the TTN web server. This could be customized and selected from the IFTTT console as shown in Figure 17B, i.e. ‘If Webhooks Then Adafruit’.

This trigger leads to form to fill in the details of the data transfer. As shown in Figure 17C, the form to fill the details of the event name (that is customized on the Adafruit IO server) and the feed name is selected based on the value you enter in the Adafruit Feeds shown in Figure 22. The IFTTT initializes each of these payload data as an applet that triggers the transfer function.

Figure 18 shows the three active applets that were used to transmit data over the network for this project. The three applets are named after the keywords assigned to the specific payload, i.e. temperature, humidity, and PIR state.

Payload decoder and Adafruit data storage (application server)

This has been the final stage of the LoRa architecture implementation. The network has been now fully connected and assembled. The only pending parameter is the payload format. It is important to understand the payload format for the decoding process.

The data for the temperature, humidity, and the PIR state are encoded using the Cayenne LPP format. Figure 19 shows a sample on how the payload format is encoded in the Cayenne LPP method. Also, Figure 20 shows an actual sample of the payload of the data that is stored temporarily on the TTN server.

The following payload must, however, be decoded to be transmitted to the cloud server. For the decoding, it is essential to understand the format it has been encoded. Figure 21 shows the decoder used to recalibrate the data from the given payload.

The payload data have now retrieved the original format of the data that was calibrated by the sensor. This was transmitted to the cloud server, i.e. the Adafruit IO server.

The feed was visible on the Feeds page of the Adafruit IO webpage, this has been shown in Figure 22.

This completes the architecture of the sensor node based on LoRa network. The system is efficient and can be used efficiently since it has very low power consumption and can be utilized in remote localities.

A

Applications > group2 > Integrations > 45666202

Process ID: 45666202

Status: Running

Platform: IFTTT Maker (v2.6.0) documentation

Author: The Things Industries B.V.

Description: Sends uplink messages to your IFTTT Maker Channel. If you want to trigger this integration based on sensor values, use the 'trigger' payload field (true or false) as a result of your decoder or converter payload function. Use the special fields 'dev_id' for device ID, 'app_id' for application ID and 'hardware_serial' for the DevEUI.

B

If This Then That

Add the First TRIGGER

If [] Then []

Add the Second TRIGGER

C

If Maker Event "SolarMPPTMonitoring", then Send data to humidity feed

Event Name: SolarMPPTMonitoring

Send data to Adafruit IO

Feed name: humidity

Data to save: Value1

Receive a web request

Save

Figure 17: A: IFTTT process integration with TTN shown in the application console; B: IFTTT process integration with Webhooks and TTN shown in the IFTTT CREATE page; C: IFTTT Applet creation, to enable data from the TTN webserver to be transmitted to Adafruit IO.

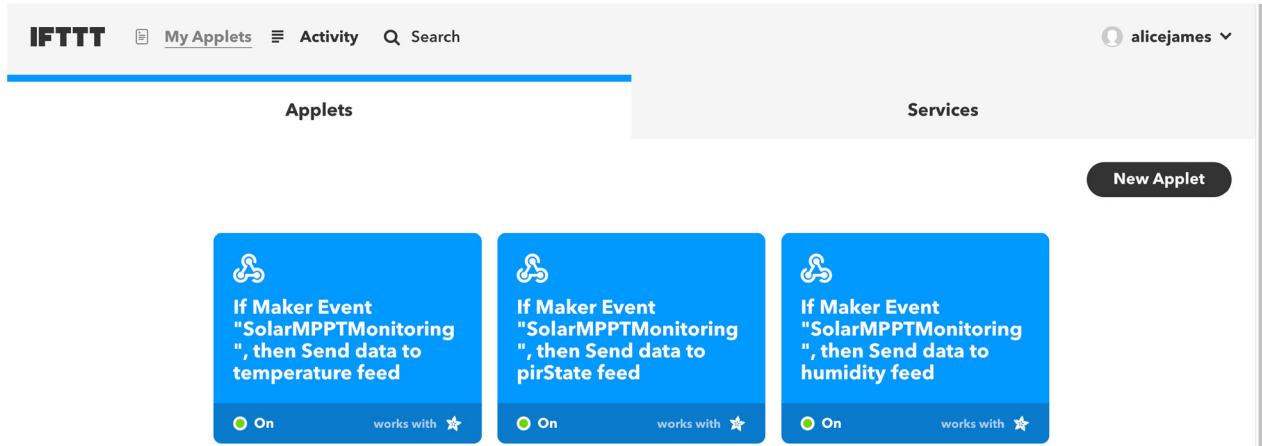


Figure 18: IFTTT final process Applets, establishing connection between The Things Network and Adafruit IO.

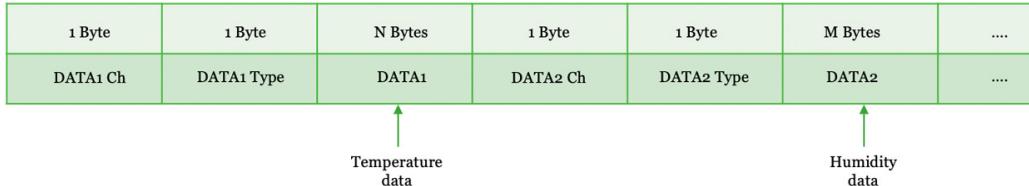


Figure 19: Cayenne LPP payload format describing the values of the corresponding byte.

The screenshot shows the TTN Application Data interface. The 'uplink' filter is selected. The table displays two received messages:

time	counter	port	dev id	payload
▲ 10:17:50	176	10	confirmed	mppt_mkrwan_1300 payload: 01 67 00 8C 02 68 84 03 01 00
▲ 10:11:09	x	x	historical	mppt_mkrwan_1300 payload: 01 67 00 8C 02 68 84 03 01 00

Figure 20: Cayenne LPP payload received on TTN application console from the sensor node.

Results

The results section gives a brief description of the data collected for 24hr and the analysis platform provided by Adafruit IO.

The sensor node was deployed indoors near the windowpane within an apartment. The data were then

observed for 24 hr and the payload was deployed for every 10 min. An hourly data were plotted for each of the sensor readings that are illustrated in Figure 23A-C.

The Adafruit IO platform enables the user to compile these data and generate a 'csv' file that can be used for further data analysis and correlations.

The screenshot shows the 'Payload Formats' section of the Cayenne application interface. At the top, there are tabs for Overview, Devices, Payload Formats (which is selected), Integrations, Data, and Settings. Below the tabs, the title 'PAYLOAD FORMATS' is displayed. A sub-section titled 'Payload Format' is shown, with the subtitle 'The payload format sent by your devices'. A dropdown menu is set to 'Custom'. Below this, there are four buttons: 'decoder' (highlighted in blue), 'converter', 'validator', and 'encoder'. To the right of the buttons is a red link 'remove decoder'. A code editor window contains the following JavaScript code:

```

1 function Decoder(bytes, port) {
2   // Decode an uplink message from a buffer
3   // (array) of bytes to an object of fields.
4   var decoded = {};
5   decoded.temperature = bytes[3];
6   decoded.humidity = bytes[6];
7   decoded.pirState = bytes[8];
8   // if (port === 1) decoded.led = bytes[0];
9
10  return decoded;
11 }

```

A small note 'decoder has no changes' is visible in the bottom right corner of the code editor.

Figure 21: Cayenne LPP payload decoder on TTN application console, to transmit the decoded payload to the application server.

The screenshot shows the 'TrickFinger / Feeds' page on Adafruit IO. The top navigation bar includes Profile, Feeds, Dashboards, Triggers, Services, and an Adafruit IO Key. The main content area displays a table of feeds:

Group / Feed	Key	Last value	Recorded
Default	default		
humidity	humidity	66	about 1 min ago
pirState	pirstate	1	about 1 min ago
temperature	temperature	12	about 1 min ago

Figure 22: Adafruit IO feeds page that shows the list of feeds/payload received from IFTTT.

The primary factors observed in this project implementation are the LoRa connectivity and the consumption of the battery.

The graphs show data that had been captured between 3:00 pm and 8:00 pm AEST. Each of the data calibrated was verified with the actual temperature and humidity values and it was almost accurate. The value for the pedestrian data was frequently 1 since the sensor node was placed indoors and there was almost always someone around the sensor.

Therefore, the system can be used as a smart monitoring system for environmental monitoring and generating the pedestrian count for the day.

Conclusion

In conclusion, the sensor node has been developed that serves as an exemplar for any IoT enthusiast with an eagerness to develop their own smart sensing system. The sensor node has been based on one

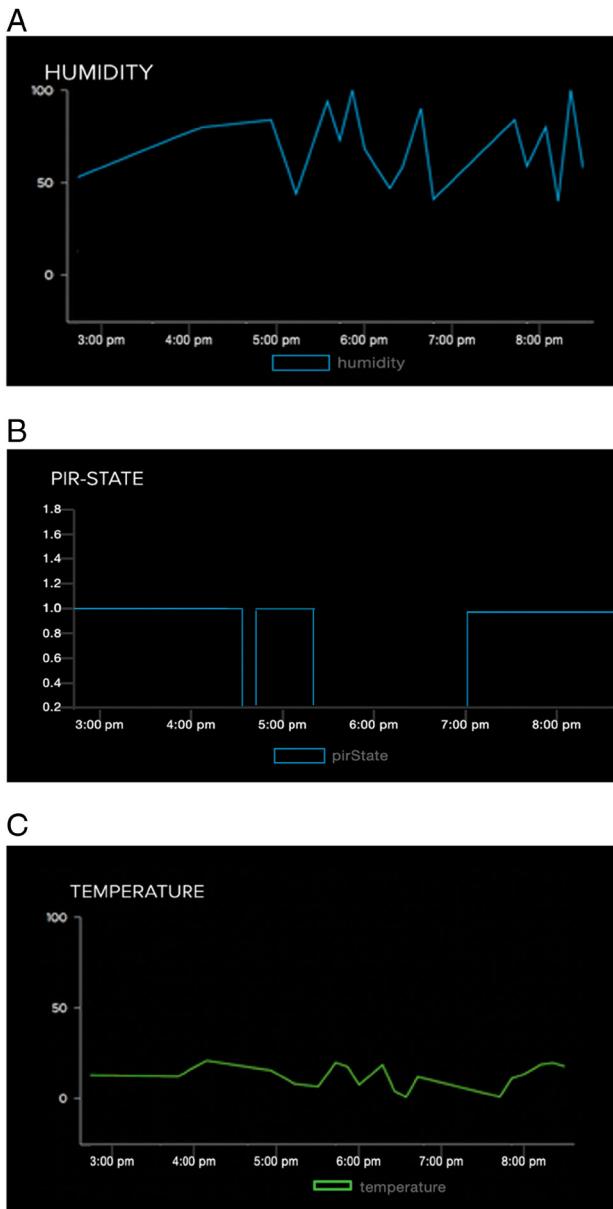


Figure 23: A: Humidity sensor data on the Adafruit dashboard; B: PIR state sensor data on the Adafruit dashboard; C: Temperature sensor data on the Adafruit dashboard.

of the prominent technologies in LPWAN, i.e. LoRa technology. The different wireless protocols such as Bluetooth 5, Zigbee, NB-IoT, etc. available as competitors for the IoT applications were evaluated and reviewed for their transmission capabilities.

The sensor node has been developed in cohesion with the LoRa architecture, every milestone in the architecture achieved in this sensor node has been

highlighted and discussed. The primary integration of the main interfacing sections of the LoRa architecture such as the WSN Nodes, Gateway, Network Server, and Application Server has been explained in detail. Multiple code syntax to aid the design of a low power payload transmission has been provided to allow the developer to obtain a perspective to plan the payload format. And finally, the platform to decode the payload of the sensor readings in real-time and analysis of the data were observed.

As mentioned earlier, the broad aspect of this architecture can be translated into different IoT systems for different applications such as Smart homes, transport, cities, etc. This research paper will act as a guide to enable an effective and energy-efficient IoT system using LoRa technology.

Literature Cited

Afsarimanesh, N., Alahi, M. E. E., Mukhopadhyay, S. C. and Kruger, M. 2018. Development of IoT-based impedometric biosensor for point-of-care monitoring of bone loss. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8(2): 211–220.

Akhter, F., Khadivizand, S., Siddiquei, H. R., Alahi, M. E. E. and Mukhopadhyay, S. 2019. IoT enabled intelligent sensor node for smart city: pedestrian counting and ambient monitoring. *Sensors* 19: 3374–3393.

Alahi, M. E. E., Pereira-Ishak, N., Mukhopadhyay, S. C. and Burkitt, L. 2018. An Internet-of-Things enabled smart sensing system for nitrate monitoring. *IEEE Internet of Things Journal* 5(6): 4409–4417.

Augustin, A., Yi, J., Clausen, T. and Townsley, W. 2016. A study of LoRa: long range & low power networks for the Internet of Things. *Sensors* 16(9): 1466–1484.

Choi, Y. -J., Kang, H. -J. and Lee, I. -G. 2019. Scalable and secure Internet of Things connectivity. *Electronics* 8(7): 752–767.

Collotta, M., Pau, G., Talty, T. and Tonguz, O. K. 2018. Bluetooth 5: a concrete step forward toward the IoT. *IEEE Communications Magazine* 56: 125–131.

Davcev, D., Mitreski, K., Trajkovic, S., Nikolovski, V. and Koteli, N. 2018. IoT agriculture system based on LoRaWAN. 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, 1–4.

Germani, L., Mecarelli, V., Baruffa, G., Rugini, L. and Frescura, F. 2019. An IoT architecture for continuous livestock monitoring using LoRa LPWAN. *Electronics* 8(12): 1435–1463.

Haxhibeqjiri, J., Karaagac, A., Van den Abeele, F., Joseph, W., Moerman, I. and Hoebelke, J. 2017. LoRa indoor coverage and performance in an industrial environment: case study. 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, 1–8.

- Hughes, J., Yan, J. and Soga, K. 2015. Development of wireless sensor network using bluetooth low energy (BLE) for construction noise monitoring. *International Journal of Smart Sensing and Intelligent Systems* 8(2): 1379–1405.
- Huh, H. and Kim, J. Y. 2019. LoRa-based Mesh network for IoT applications. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, 524–527.
- Jawad, H. M., Nordin, R., Gharghan, S. K., Jawad, A. M. and Ismail, M. 2017. Energy-efficient wireless sensor networks for precision agriculture: a review. *Sensors* 17(8): 1781–1826.
- Kazakov, G. N., Petrkov, A. M. and Shevtsov, V. A. 2020. Radio monitoring of wireless networks using LoRa data transmission technology. 2020 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, 1–5.
- Kumar, N. M. and Mallick, P. K. 2018. The Internet of Things: insights into the building blocks, component interactions, and architecture layers. *Procedia Computer Science* 132: 109–117.
- Lysogor, I. I., Voskov, L. S. and Efremov, S. G. 2018. Survey of data exchange formats for heterogeneous LPWAN-satellite IoT networks. 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), Moscow, 1–5.
- Magrin, D., Centenaro, M. and Vangelista, L. 2017. Performance evaluation of LoRa networks in a smart city scenario. 2017 IEEE International Conference on Communications (ICC), Paris, 1–7.
- Mekki, K., Bajic, E., Chaxel, F. and Meyer, F. 2019. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* 5(1): 1–7.
- Nag, A., Alahi, M. E. E., Feng, S. and Mukhopadhyay, S. C. 2019. IoT-based sensing system for phosphate detection using Graphite/PDMS sensors. *Sensors and Actuators A: Physical* 286: 43–50.
- Navarro-Ortiz, J., Sendra, S., Ameigeiras, P. and Lopez-Soler, J. M. 2018. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things. *IEEE Communications Magazine* 56(2): 60–67.
- Pasolini, G., Buratti, C., Feltrin, L., Zabini, F., De Castro, C., Verdone, R. and Andrisano, O. 2018. Smart city pilot projects using LoRa and IEEE802.15.4 technologies. *Sensors* 18(4): 1118 –1125.
- Petäjäjärvi, J., Mikhaylov, K., Hämäläinen, M. and Linatti, J. 2016. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. 10th International Symposium on Medical Information and Communication Technology (ISMICT), Worcester, MA, 1–5.
- Pirbhulal, S., Zhang, H., Alahi, M. E. E., Ghayvat, H., Mukhopadhyay, S. C., Zhang, Y. -T. and Wu, W. 2017. A novel secure IoT-based smart home automation system using a wireless sensor network. *Sensors* 17(1): 69–88.
- Prando, L. R., de Lima, E. R., de Moraes, L. S., Biehl Hamerschmidt, M. and Fraindenraich, G. 2019. Experimental performance comparison of emerging low power wide area networking (LPWAN) technologies for IoT. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, 905–908.
- Ruckebusch, P., Giannoulis, S., Moerman, I., Hoebeke, J. and Poorter, E. D. 2018. Modelling the energy consumption for over-the-air software updates in LPWAN networks: SigFox, LoRa and IEEE 802.15.4g. *Internet of Things* 3-4: 104–119.
- Santos, D. and Ferreira, J. C. 2019. IoT power monitoring system for smart environments. *Sustainability* 11(19): 5355–5379.
- Shahjalal, M., Hasan, M. K., Islam, M. M., Alam, M. M., Ahmed, M. F. and Jang, Y. M. 2020. An overview of AI-enabled remote smart-home monitoring system using LoRa. 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Fukuoka, 510–513.
- Singaravelan, A. and Kowsalya, M. 2016. Design and implementation of standby power saving smart socket with wireless sensor network. *Procedia Computer Science* 92: 305–310.
- Spachos, P. 2020. Towards a low-cost precision viticulture system using Internet of Things devices. *IoT* 1(1): 5–20.
- Visconti, P., Ferri, R., Pucciarelli, M. and Venere, E. 2016. Development and characterization of a solar based energy harvesting and power management system for a WSN node applied to optimized goods transport and storage. *International Journal on Smart Sensing & Intelligent Systems*, 9(4): 1637–1667.
- Yueshun, H. and Wei, Z. 2013. The research on wireless sensor network for landslide monitoring. *International Journal on Smart Sensing & Intelligent Systems* 6(3): 867–887.