

# Performance Analysis of LoPy4 in Indoor and Outdoor Environments

Harikrishnan SM, Pratiksha Bande, Tanbir Waresi, Nishigandha Patil



<sup>1</sup>harikrishnan@mitaoe.ac.in

<sup>2</sup>pabande@mitaoe.ac.in

<sup>3</sup>twareasi@mitaoe.ac.in

<sup>4</sup>ngpatil@mitaoe.ac.in

School of Electrical Engineering, MIT Academy of Engineering  
Alandi, Pune, India.

## ABSTRACT

LPWAN or Low Power Wide Area Network has become the buzzword for IoT since the inception of this decade. Nowadays IoT applications are focusing on high range coverage along with low power consumption. Among all LPWAN wireless communication technologies like LTE-M, Narrow Band IoT, SigFox, Symphony, Ingenu, etc., Semtech preferred the chirp spread spectrum modulation in the development of the physical layer of LoRa Protocol in place of frequency shift keying employed by conventional wireless communication system. This paper describes the setting up of a wireless communication system using Pycom LoPy4 which supports LoRa, SigFox, BLE and Wi-Fi. The work described in this paper focuses only on establishing a communication link between two LoPy4 modules which is restricted to the physical layer of LoRa. In accordance with the specified purpose, expansion board 3.0 suits our requirements the most as a base board to mount the LoPy4, as compared to other versions of the expansion board. Furthermore, we talk in detail about the points that is needed to be taken into consideration while setting the hardware, installing driver software and updating the firmware of LoPy4 module, thereby opening the doorway into the realm of micropython using Atom IDE. We also explain the failed attempt of transmission of an image signal matrix via this established communication link and recorded our results.

**Keywords**— IOT, LPWAN, WSN, Pycom LoPy4, Base Board, micropython, Atom IDE.

## ARTICLE INFO

### Article History

Received: 8<sup>th</sup> March 2020

Received in revised form :

8<sup>th</sup> March 2020

Accepted: 10<sup>th</sup> March 2020

### Published online :

11<sup>th</sup> March 2020

## I. INTRODUCTION

IoT is playing vital role in advancement of industrial applications, consumer electronic products, smartly improvement of the homes, buildings, cities, agriculture, metering, supply chain and logistics by introducing automation into it and also helping to overcome current challenges like energy management, natural resource reduction, pollution control, infrastructure efficiency, disaster prevention, a never ending list of applications. IoT is now involved in every walk of life. Coverage distance of interconnection of devices from room to building to city has been achieved by employing wireless communication technologies like Bluetooth, Wi-Fi, RFID, Zigbee, GSM and now LPWAN respectively.

Table shows the comparison of different existing wireless communication technologies.

**Table 1.** Comparison of Communication Protocols

Technology	Range	Data Transfer Rate	No. of End Devices	Frequency
Bluetooth	10 -15 meters	2- 3 Mbps	240	2.45 GHz
Wi-Fi	15 -20 meters	50- 250 Mbps	265	2.4 GHz
RF transceivers	1 km	1-500 Kbps	334	433 MHz
Zigbee	10- 100 meters	250 kbps	2119	2.4 GHz
GSM	100 - 500 meters	9-10 kbps	825	850 -1900 MHz
LPWAN	10-15 Kms	35-50 kbps	2500	433,868,915 MHz

Low power wide area network (LPWAN) is suitable for many applications in Internet of things because of its low power and wide area capabilities. Many IoT applications demands interconnecting devices over a long distance with optimized battery usage at the cost of compromising on low data transfer rate, (which typically in the order of tens of kilobytes per second), high latency (in the order of seconds or even minutes) and long sleeping time.

Many articles approved the significance of LPWAN in IoT applications like in ships, water grid management system. Interconnected LoRa nodes has ability to cover entire ship under humid conditions with the benefit of reducing the complexity of penetrating cables and optical fibers and also monitoring the information is more reliable and real-time. Similarly, LoRa deployed water grid management system has ability to monitor and control information even in harsh environment with effective battery usage for long period of time.

A number of authors [1] have studied performance and the limits of the various LPWAN protocols.

Based on parameters like modulation scheme used, Rx bandwidth, data transfer rate, Output power, link budget, battery lifetime, interference immunity, security comparison has been done. In comparison with BPSK and GFSK used in Sigfox and NB-IoT, the spread spectrum modulation technique implied in LoRa assures an increased link budget as well as better immunity to network interferences. 100 KHz bandwidth used in NB-IoT, and 125 KHz, 250 KHz or 500 KHz that in LoRa.

Some [2] have shared information about development process of LoRa and its characteristics.

In August 2013, Semtech developed LoRa protocol which can be seen as physical layer of OSI model. The technology operates in the ISM free band (the unauthorized band) around the world, including the main 433MHz, 407MHz, 470MHz, 868MHz and 915MHz. Whereas, LoRaWAN is designed by LoRa alliance which can be mapped as MAC layer of OSI model. Star topology network architecture used in LoRaWAN improves its capacity to receive messages from a very high volume of nodes (i.e. network capacity). AES encryption is used with the key exchange utilizing IEEE EUI64 identifier.

The objective of this paper is to provide a procedure of making a node using the Lopy module and furthermore establishing communication between two Lopy4 modules based on LoRa protocol (IEEE 802.11ah) and transferring data in text format. Also, we discussed about our failed attempt of transferring image using LoPy4 module and thereby analyzing the limitations of Lopy4 module using LoRa as communication protocol. In this context of LoRa application, we discussed about Atom IDE, a platform used to do socket programming in MicroPython.

The paper is structured as follows: after a brief introduction, Section II describes about characteristics of LoRa and its infrastructure of Lopy4 module, followed by making of LoRa node using Lopy4 module in section III. The main contribution of this work is to put forward the challenges raised while creating a node and how to tackle them, also here analyzed the failure while transferring image as a data.

## II. SYSTEM MODEL ARCHITECTURE

### A. Sx1276

The talk about the LoPy4 development board cannot be commenced without the information about the Semtech's Sx1276 LoRa long range modem that provides long range communication employing the spread spectrum technique while using minimal current consumption. This long-range transmission owes not only to its spread spectrum modulation but also to its great link budget of 168 dB. Apart from that, its -148 dBm sensitivity allows it to receive the weakest of the signals. [6]

### B. ESP32

ESP32 in the LoPy4 is its core which includes 2.4 GHz Wi-Fi and Bluetooth combo chip also the successor ESP8266 chip. The Wi-Fi inside uses the 802.11 b/g/n 2.4 GHz releasing up to 150 Mbps. The microcontroller is powered by Xtensa Dual Core 32-bit microprocessor capable of processing up to 600 MIPS and an internal clock of 8MHz. As for storage it employs a 448KB ROM 520KB RAM and a 16KB SRAM in its real time clock. The peripherals exercised by this chip include 2 x 8bit DAC, 1x 12-bit 18 channel ADC, 10 x touch sensors and serial communication ports encompassing SPI, IIC, IIS, UART, Ethernet and CAN 2.0.[5]

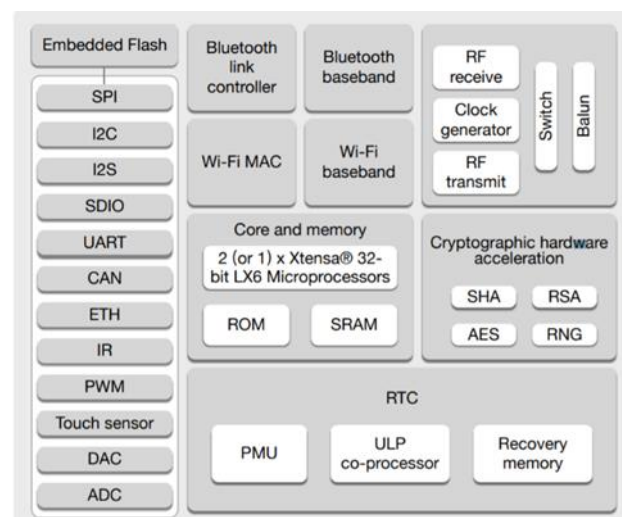


Fig. 1. ESP32 block diagram [5].

### C. LoPy4

Finally, the LoPy4 is a quadruple interconnected network development board [3] which employs the wireless communication techniques including LoRa, SigFox, WiFi and Bluetooth. It is an interconnected Communication system using the espressif's ESP32 chip for its data processing and SX1276 modem for long range modulation. This chipset is a micropython enabled development PCB which can be programmed using either Atom IDE or Microsoft Visual studio. This module when connected with an 868MHZ, 13dB whip antenna can act as a LoRanano gateway and a multi -bearer node which has the potential to form Wide area networks. LoPy can be operated as a LoRa module in a total of three frequencies on the basis of demographics which can be summed up as 868MHz for Europe, 915MHz North and South America, Australia and New Zealand, while Asia can use the frequency as that of Europe i.e. 433MHz. The total combined power required for the operation of this module is 3.3 V to 5.5 V. which is why its lifetime can be as impressive as 10 years while using a 9V battery. The Sx1276 modulator chip when combined with the processing aptness of ESP32, gives LoPy4 its condensedstructure with Long Range and Low Power prowess. [3]

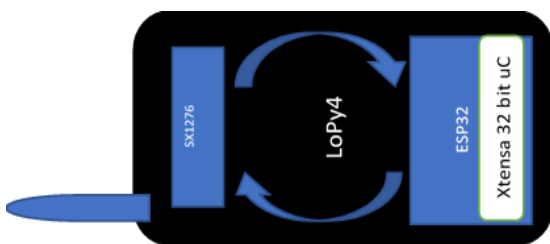


Fig. 2. LoPy4 internal structure.

**D. Expansion board 3.0**

The LoPy4 cannot be directly used for the application since it has its pins directly accessible and therefore poses difficulties in user interface of the module, which is why Pycom provided some baseboards for easier handling of the module these come in varieties of expansion boards including expansion board 2.0, expansion board 3.0, PyTrack, Pysense, PyScan. The latter three being different modified and advanced version of expansion board on the basis of location tracking (PyTrack), sensor packing (PySense), Identification Enabling (PyScan). These three are the cost-effective solution for the deployment of LoRa in the MAC Layer or in Wide Area Networks. Since the scope of this project is limited just to the setting up of a communication link and successfully transmit different type of signals in various configurations, the expansion board 3.0 fits the bill as the bade board. The expansion board provides a user-friendly jumper access to the internal structure of the module, including that, it's also provides some extra features including USB powered Power and Status LEDs, MicroSD Card Socket, Button to enter into safe mode, Micro USB connector for power and serial communication. [4]

**III. SETTING UP LORA**

**E. Hardware setup**

This section discusses the procedure of establishment of LoPy4 constituting of the main LoPy4 Module, its base board (Expansion board 3.0 in this case), antenna specific to the frequency band (868 MHz in this case), U. FL to

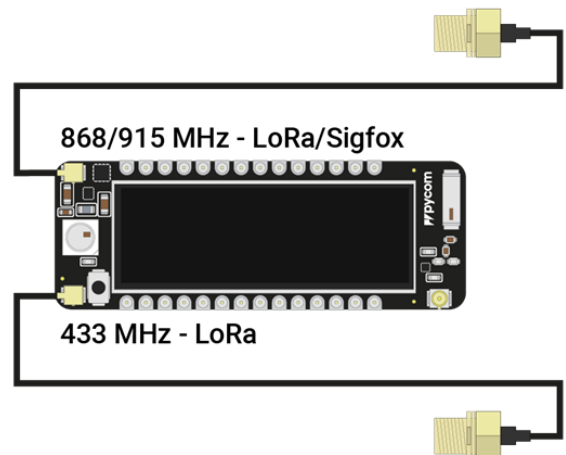


Fig. 3. Hardware setup [7].

SMA pigtail and a USB cord. There are in total 3 U. FL connectors available on the board for 868MHz LoRa/SigFox, 433MHz LoRa and Wi-Fi. The pigtail connectors are to be connected to its specific antenna from its SMA port and the to the board from its U. FL port. this structure is then to be mounted on the expansion board. [7]

The first step in setting up of the hardware involves updating the firmware of the expansion board. This process is much easier when done using a command user interface operating system like Linux. The firmware update file name DFU-util is to be downloaded from the pycom documentation site. The firmware update is done when the board is in Bootloader mode rather than in application mode which the default mode. This is the reason why Windows is not preferable as a firmware updating OS as the board stays in the bootloader mode for

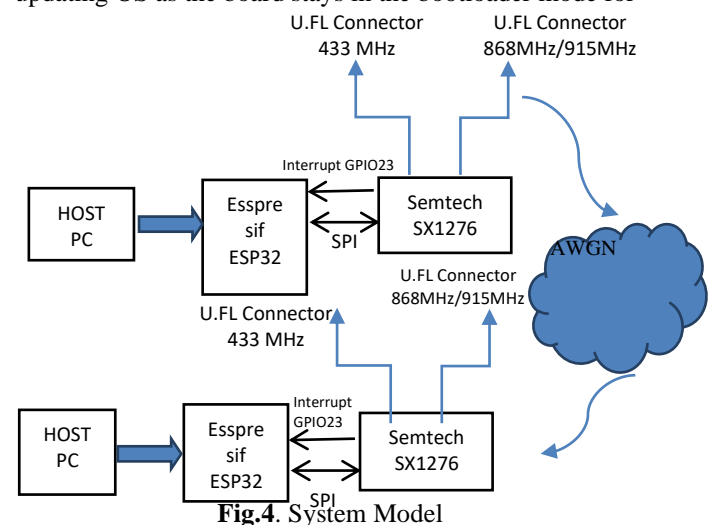


Fig.4. System Model

only 7 seconds which sets the time limit for completion of the firmware update, and is rarely achieved by windows. Linux being a Command oriented OS works faster comparatively, by running the terminal command –

```
$ sudo yum install dfu-util; For the installation of update file and then-
```

```
$ dfu-util -D pytrack_0.0.8.dfu; For the update of firmware
```

## F. Software Setup

As a preceding step to use the device for micropython programming the firmware of the pycom device is also needed to be updated. This is done by running a firmware upgrade tool after connecting the board to the computer using USB and pressing the reset button on the board. After this step the device will be ready to be used for programming.

Next step is to set up the development platform for micropython programming of the board. The programming of LoPy4 can be done in either of these two development environments: Atom or Microsoft Visual studio. Since the bulk of the work has taken place in Linux environment Atom fits perfectly for the role. Pycom has provided a plugin software for both the IDE entitled PyMakr plugin. So, the first step in prepping the IDE is to install PyMakr plugin from the "Install Package" Option in Atom

Now that everything is set, the programming of the module can be commenced.

## G. Programming the Module

To program the module, it first needs to be connected to the computer, which in turn can be done using Serial USB and Telnet. Connection through USB can be established pasting the serial address obtained from the serial port information on to the Device address box. Connection via Telnet is in layman terms, connecting using Wi-Fi. Using this method will activate the Wi-Fi of the board and can be seen in the access point list which would be named after the board, in the PC. The password for the access point is "www.pycom.io" whereas the serial address needs to be changed to 192.168.4.1. The successful connection of the module to PC is represented by the sign ">>>".

The console provided by the PyMakr is known as the Real Evaluate Print Line (REPL) console, some important functions provided are:

```
Pycom.heartbeat(false); which turns off the LED representing the heartbeat of the module.
```

```
Pycom.rgbled(0x00ff00); which makes the LED glows in the color specified by the parameter.
```

LoRa can work in 2 different type of mode: LoRa MAC mode or Raw-LoRa and in LoRaWAN mode. LoRaMAC mode does not make use of any headers, addressing information or encryption. This includes the typical LoPy to LoPy communication whereas LoRaWAN mode implements the LoRaWAN stack for a Class A device. To

achieve a LoRaWAN it requires to have more than 2 LoPy4 modules which is why we stay in the MAC mode.

- Configuring transmitter and receiver: The first program executed was establishing a transmitter which was done for LoPy to LoPy communication link. For Networking micropython, which is basically an embedded system implementation of Python 3 for LoRa provides a package named "network". The library named "Lora" can be imported from this package. Developing a node is basically socket programming which can be used by importing socket library.

```
S= socket. socket (socket.AF_LORA, socket.SOCK_RAW); To establish the socket, we use the line
```

```
Lora=LoRa (mode=Lora. LORA, region=LoRa.EU868) Setting the region of LoRa is done using
```

The 5 characters in the end represent the region and its corresponding frequency for example AS923 for Asia, AU915 for Australia and US915 for United States.

Once the node is set up we can transmit the data using s.send() functions from the socket library.

This is the program structure for the transmitter. For receiver we can use s.recv(). These two send and receive function can also be clubbed with some conditional statements to specify the data for instance reply "Pong" on when the received data is "Ping".

- Sending an Array: The second task was to try to send an array instead of a character. This could not be achieved by using the send functions directly the reason being, the send() and recv() functions are developed only for character data. To overcome this hurdle the only method is to convert the integer arrays into character data so that we send it using send() function, or otherwise we can also use a whole different function to send the integer array none of which compatible with python are not available for micropython. The exact function used for this purpose is the bytes function which fulfill this purpose.

## IV. PERFORMANCE ANALYSIS OF SYSTEM

### H. Experimenting

Once the integer array was transmitted using bytes function, an attempt was made to calculate the received power of the system. For this purpose, a dB meter was used which would give power in dB. But the next issue was that, since the receiving antenna needed to be connected to the dB meter directly and the both the ports of receiving antenna and the dB meter were female SMA no signal power was received in this case. To resolve this

issue, after searching for a lot of attenuators we made use of a custom made male SMA to SMA connector which worked and gave the result for a distance of 50 cm between the transmitter and receiver antenna. The received power obtained was 60dBu. This result was verified using the Friss transmission equation for power calculation.

### I. Verification

The transmitter kept at a distance of 0.5 meters continuously transmitting pings at the frequency of 868 MHz or the wavelength of 34.54 cm. The antenna used was an RP- SMA (Male) tilt swivel whip antenna which provides a gain of around 13dBm (including the antenna gain and cable loss). The received power was calculated using the Friss Transmission equation to give a received power of 59 dBu.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi r)^2} \quad (1)$$

$P_r$  = Power at the receiving antenna

$P_t$  = Output power of transmitting antenna

$G_t$  = Gain of the transmitting antenna

$G_r$  = Gain of the receiving antenna

$\lambda$  = Wavelength

$R$  = Distance between the antenna

### J. Image Transmission

Since the arrays were successfully transmitted, it was imminent to try to send images which could yield potential to a whole different arena of image processing, since long-distance image transmission is still a feat to accomplish. But it turned out that, owing to the memory limitations of the module it is impossible as for now or at least arduous for any embedded system module given the distance abilities of LoRa modulation to transmit an image. The ROM memory of the Xtensa processor is 440KB while for image procession using python it is imperative to import the fundamental package named NumPy library provided by Python 3. But then again, this NumPy library being an important library for matrix and array handling in python has a size much greater than the size of the memory ROM. Aside from that, the data rate of the LoRa is less the 50 kbps which makes it even more difficult to send a heavy matrix like that of an image.

## V. CONCLUSION

This study conducted helped to understand the scope of different modes of processing which can be done using the lopy4 module. For example, Image processing, as well as signal processing applications, will be difficult to

implement on this board because it works on micropython that does not support the image processing libraries (NumPy) provided by python. The primary reason is the size of the library that is very huge when compared to the memory size of the microcontroller.

The scope of this is limited to applications involving slow data rates like sensor node applications which can be used for developing “on-field” applications like animal tracking, agricultural land parameters, military GPS. It can also be used in places where, along with sensors, there are vast areas of no obstacle land which can maximize the ability of the protocol, increasing its range up to 15-20 km.

## REFERENCES

- [1] DavideMagrin, Martina Capuzzo, and Andrea Zanella, “A Thorough Study of LoRaWAN Performance Under Different Parameter Settings”, *IEEE Internet of Things Journal*, vol. 7, no. 1, January 2020.
- [2] OratileKhutsoane, BasseyIsong, NaisonGasela, and Adnan M. Abu-Mahfouz, “WaterGrid-Sense: A LoRa-Based Sensor Node for Industrial IoT Applications”, *IEEE Sensors Journal*, vol. 20, no. 5, march 1, 2020.
- [3]Pycom,[https://docs.pycom.io/gitbook/assets/specsheets/Pycom\\_002\\_Specsheets\\_LoPy4\\_v2.pdf](https://docs.pycom.io/gitbook/assets/specsheets/Pycom_002_Specsheets_LoPy4_v2.pdf)
- [4]Pycom,<https://docs.pycom.io/gitbook/assets/expansion3-specsheet-1.pdf>
- [5]Espressif,[www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](http://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [6]Semtech,[www.semtech.com/products/wireless-rf/lora-transceivers/sx1276](http://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276)
- [7]Pycom,<https://docs.pycom.io/gettingstarted/connection/lopy4/>