Mitesh Maheshkumar Savsaviya(6003798)
Mohamad AbdulRazek (6088227)

Group 4. Lab Exercise Report

LoRa Communications in the Internet of Things

Description of the Exercise

In this lab exercise, we have used LoRa for data transmission with Python as a programming language. We have used FiPy(Processor Part) module integrated with PySense(Sensor Part), which was readily available with different types of Sensors and LEDs on board. As our Hardware was readily available, our task was to do Software Implementation which comprises of following steps:

1. Initialization of LoRa interface
2. Initialization of the required Sensor
3. Sensor data reception
4. Conversion of Sensor data into binary format
5. Send data to the receiver via LoRa at interval of 15 Seconds

Software Implementation

We have used VS Code with pymakr extension in this task to program our Hardware. Our Programming code was done in Micro Python language, which is a kind of Python language, we used to program our Microcontroller. After successful detection of our Hardware in pymakr, we wrote our code in main.py in a workspace which looks like this.

```
1   from struct import calcsize
2   from network import LoRa
3   import socket
4   import machine
5   import time
6   import pycom
7   import ustruct as struct
8   from LIS2HH12 import LIS2HH12
9   from SI7006A20 import SI7006A20
10  from LTR329ALS01 import LTR329ALS01
11  from MPL3115A2 import MPL3115A2,ALTITUDE,PRESSURE
12  from pycoproc_2 import Pycoproc
```

Fig.1 : Import of different libraries

We begin our program by adding imports which is necessary to make several modules available in our code as given in Fig.1. which includes sensor module, network module, struct module etc.

```
py = Pycoproc()
pre = MPL3115A2(py,mode=PRESSURE) # Returns pressure in Pa.
temp = SI7006A20(py)
```

Fig.2 : Acquisition of Sensor data

In this part of our program, we read the data from Temperature Sensor (SI7006A20) and Pressure Sensor (MPL3115A2) which is available on board. In the beginning of the code, we have imported necessary libraries for the same. So, we will get values of the Sensor in this Part by using functions given in Fig.2.

```
while True:
    g = 4
    pres = int(pre.pressure()/100)
    tempe = temp.temperature()
    data = struct.pack('!BfH', g, tempe, pres)
    s.setblocking(True)
    s.send(data)
    time.sleep(15)
```

Fig.3 : Conversion of data into Binary and Send it to the receiver

In order to transmit a data such as Group name, Temperature and Pressure to the receiver together, it is necessary to put them in one string and convert them into the binary as our receiver expects a data in a Certain format in a Network Byte order. In order to do this, We used struct function to pack the data and We used BfH as a format for Group name, Temperature and Pressure data (unsigned char, float and unsigned short respectively) and ! as an Big Endian, and it'll convert the data into Binary. Meanwhile, we have also checked whether our data is the same as our receiver expected or not by using calcsize command. When our data is the same as expected it is ready to send.

```
lora = LoRa(mode=LoRa.LORA, region=LoRa.EU868)
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
lora.init(mode=LoRa.LORA, region=LoRa.EU868,
          bandwidth=LoRa.BW_125KHZ, sf=7, preamble=8,
          coding_rate=LoRa.CODING_4_5)
```

Fig. 4 : Initialisation of LoRa

This are the configurations we need to send the data in LoRa in our case. We have added the necessary parameters as given in fig.4 and after that we use command as given in Fig.3 to send the data which was our objective for this exercise.

Result

After successfully compilation and uploading of our program into our hardware. Data will be send to the receiver via LoRa and the window at Receiver side was observed as given in Figure. 5

```
12875 : RX, Group: 4 Temp: 35.75471 Pres: 1027
Lora packet received
12877 : RX, wrong size 10
Lora packet received
12880 : RX, Group: 5 Temp: 35.2399 Pres: 1027
Lora packet received
12889 : RX, wrong size 10
Lora packet received
12890 : RX, Group: 4 Temp: 35.65818 Pres: 1027
Lora packet received
```

Fig. 5 : Output at Receiver side

Discussion and difficulties

As it was really necessary to manipulate the variables as our receiver expected a data of certain form e.g., Byte order with length 7 bytes. We needed to stay careful as we couldn't pack our data before because there was no conversion of Pressure from float to integer while converting the value of pressure in hPa. Theoretically, all the packets we transmitted was sent by the receiver as we were in the range of LoRa as so receiver was.