Assignment 4 – Code RED!

Submitted By:

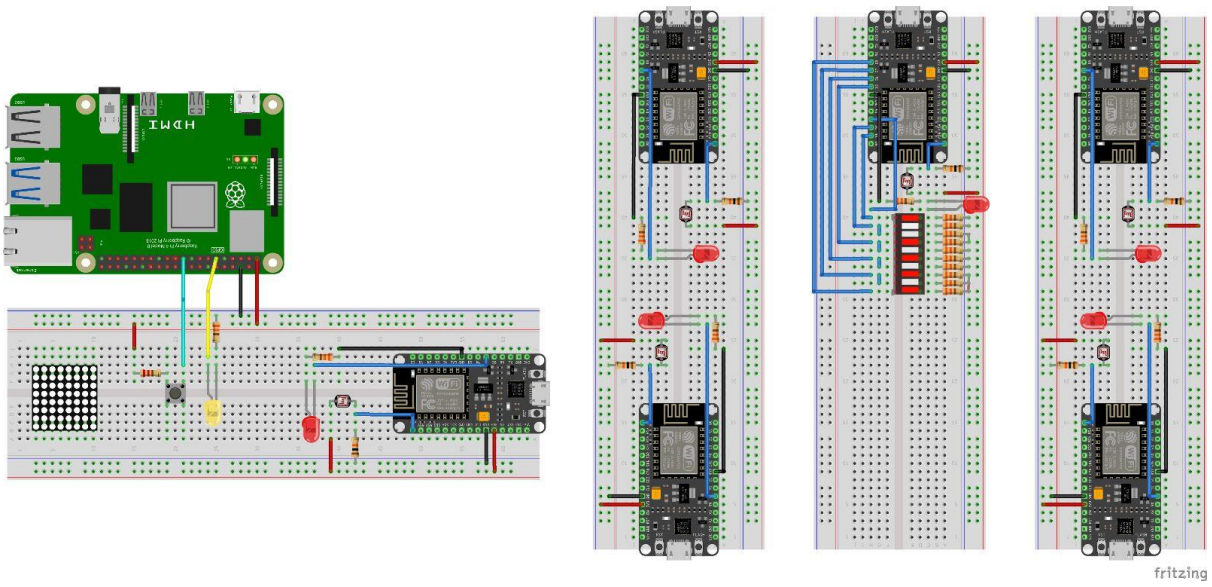
- **Bhumil Rajnikant Depani**
- **Rushabh Divyangkumar Patel**
- **Vandit Jasmin Patel**

Link of the demonstration Video:

https://drive.google.com/drive/folders/1X4Z0wM0qey_wrRXL4k2cpmeUOBWwgxG8?usp=sharing

Overview - Schematic and Communication Protocol

Schematic Diagram:



UDP (User Datagram Protocol)

It is one of the core communication protocol for communication in LAN, MAN or WAN. Compare to TCP it does not give acknowledgement functionality, but it is very fast than TCP in some cases.

Messages being send/receive by Raspberry Pi:

Raspberry Pi is receiving all LDR sensor data value from Master device.

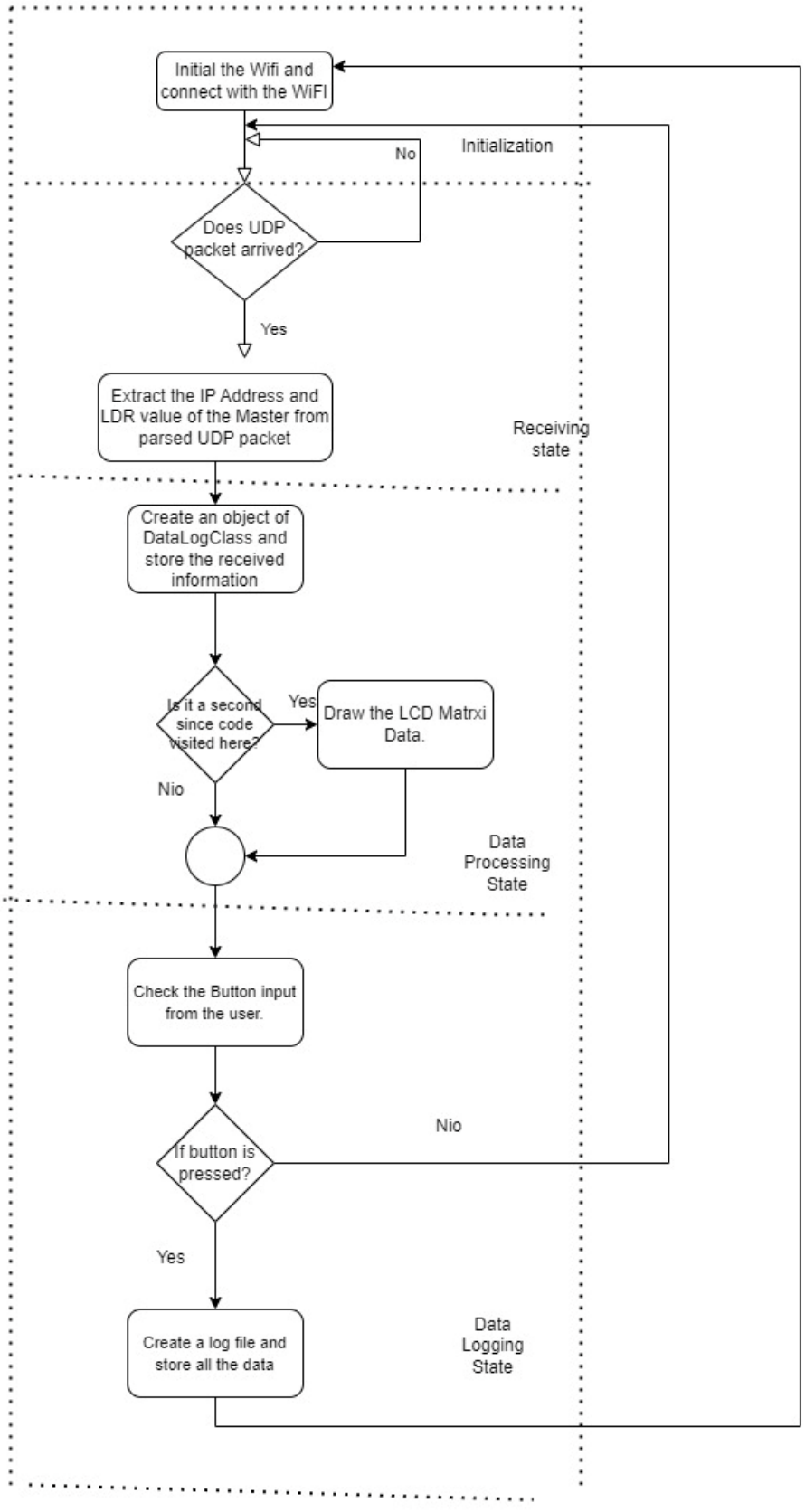
Raspberry Pi is sending reset UDP packet to all the ESPs in the network on clicking RESET button attached with Raspberry Pi.

Messages being send/receive by ESP:

Master ESP is broadcasting his own LDR data in the network. Along with this Master is also sending all LDR data gathered from other ESPs to the Raspberry pi.

Slave ESP is broadcasting LDR data in the network.

Raspberry Pi WiFi setup and packet delivery



Initialization State: Raspberry Pi will initialize and establish the WiFi and UDP communication.

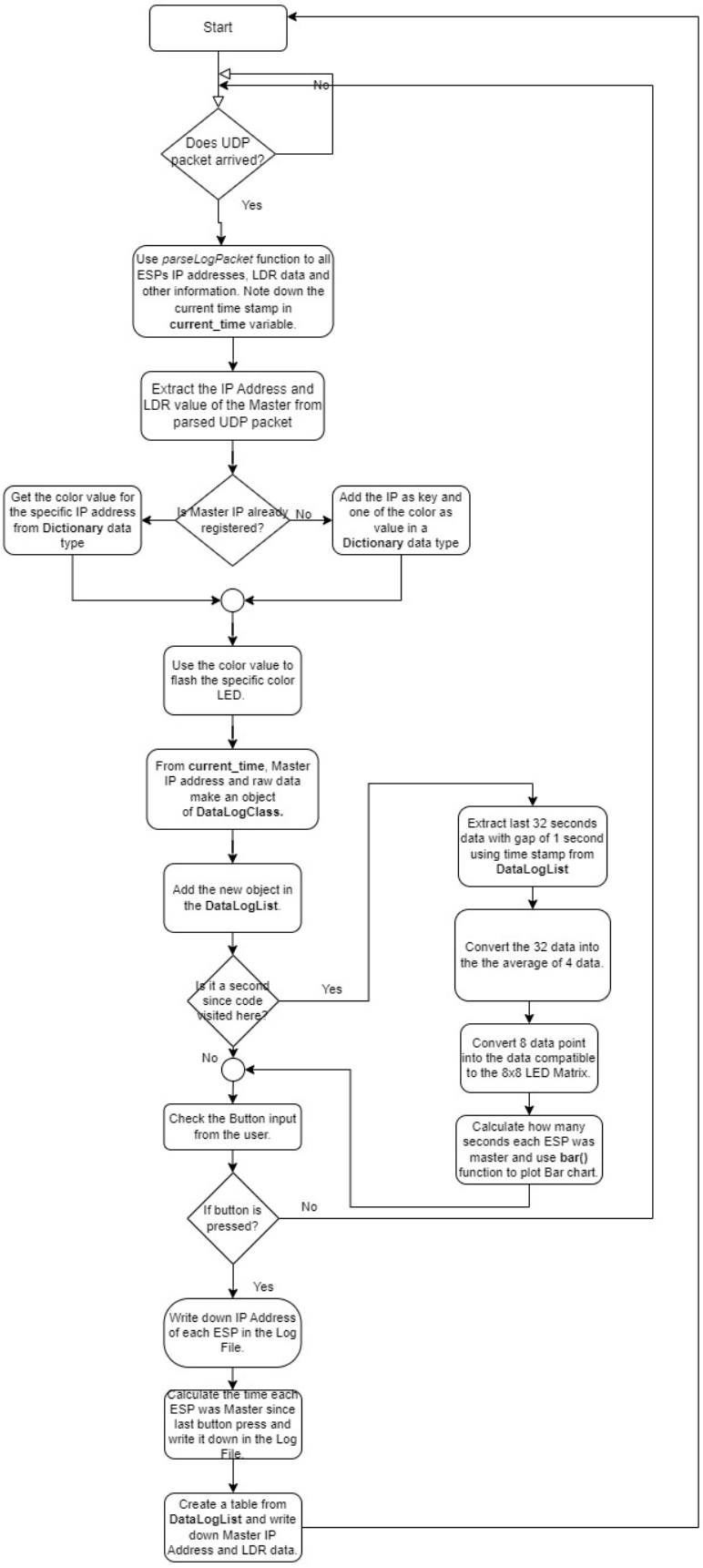
Receiving Data Packet: In this state, Raspberry Pi will receive the packet from the master.

Data Processing state: In this state Raspberry Pi will parse the data and show it on the line chart and bar chart

Data Logging State: on getting button press from the user it will log all the data into the log file and send RESET command.

Main functionality of the Raspberry pi is to show the graphs and log the data into the txt file.

A flowchart of the RPi's data collection and log writing process



Name of the List/Queue used to store the IP address of the Master ESP: **ESPMasterIPList**
Name of the class which contains time, Mater IP and LDR data fields: **DataLogClass**
Name of the List/Queue used to store the value of the LDR sensor: **DataLogList**
To store the time, when UDP packet arrived: **current_time**

Different Data Structure used in the Raspberry Pi Python code:

Using **Dictionary** data structure to store the identification of the ESP Ip Address. Here, key is IP Address of the ESP and value is “RED”, “YELLOW” or “GREEN”. Here, IP won’t be hard-coded. But as new IP will arrive, a key-value pair will be added.

We can use **Queue** data structure to store the value of LDR and Master IP address. To implement Queue in Python, **List** can be used. We will have one **List** for LDR value and another **List** for corresponding Master IP Address.

We are using **class** to store the time, Master IP Address and LDR Data. Blueprint of the class is as below.

```
class DataLogCLass:
        def __init__(self, time1, master, ldr):
                self.time1 = time1
                self.master = master
                self.ldr=ldr
```

Example Expected Log File:

IP Address's of Masters: 192.168.43.41, 192.168.43.168, 192.168.43.76

41 was Master for 41 seconds.

168 was Master for 160 seconds.

76 was Master for 95 seconds.

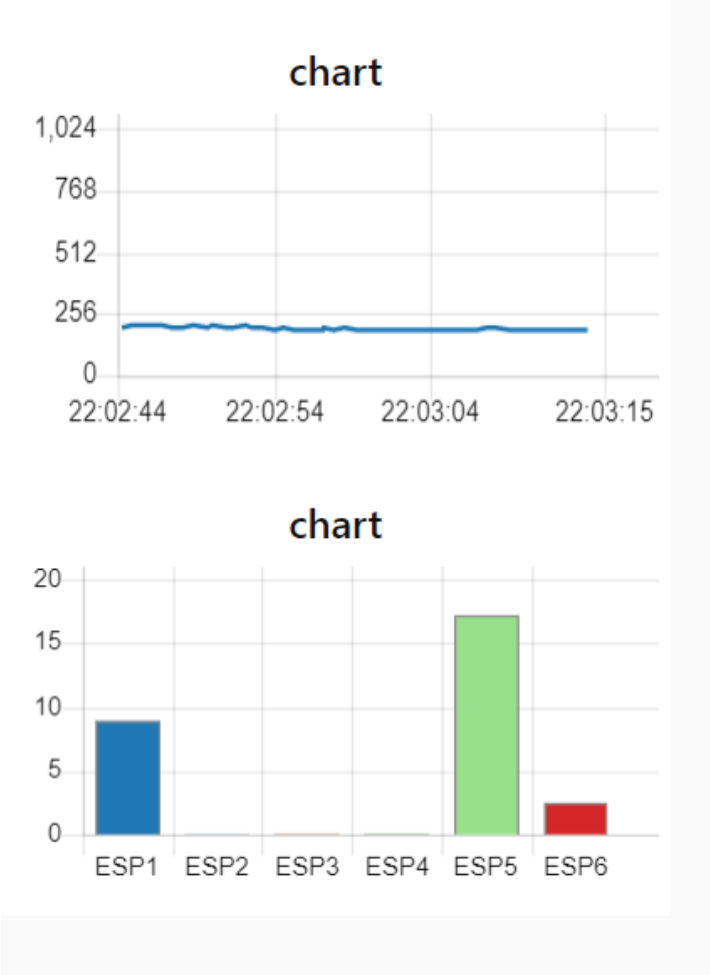
Master IP Address	Raw Data From IP Address
41	'0','1','28','155','PR','41'
168	'0','1','28','840','PR','168'
41	'0','1','28','45','PR','41'
168	'0','1','28','854','PR','168'
76	'0','1','28','900','PR','76'
76	'0','1','28','905','PR','76'
76	'0','1','28','906','PR','76'
76	'0','1','28','900','PR','76'
168	'0','1','28','850','PR','168'
168	'0','1','28','855','PR','168'
41	'0','1','28','1000','PR','41'
41	'0','1','28','1001','PR','41'
41	'0','1','28','995','PR','41'
168	'0','1','28','1010','PR','168'
168	'0','1','28','1011','PR','168'

Actual Log File:

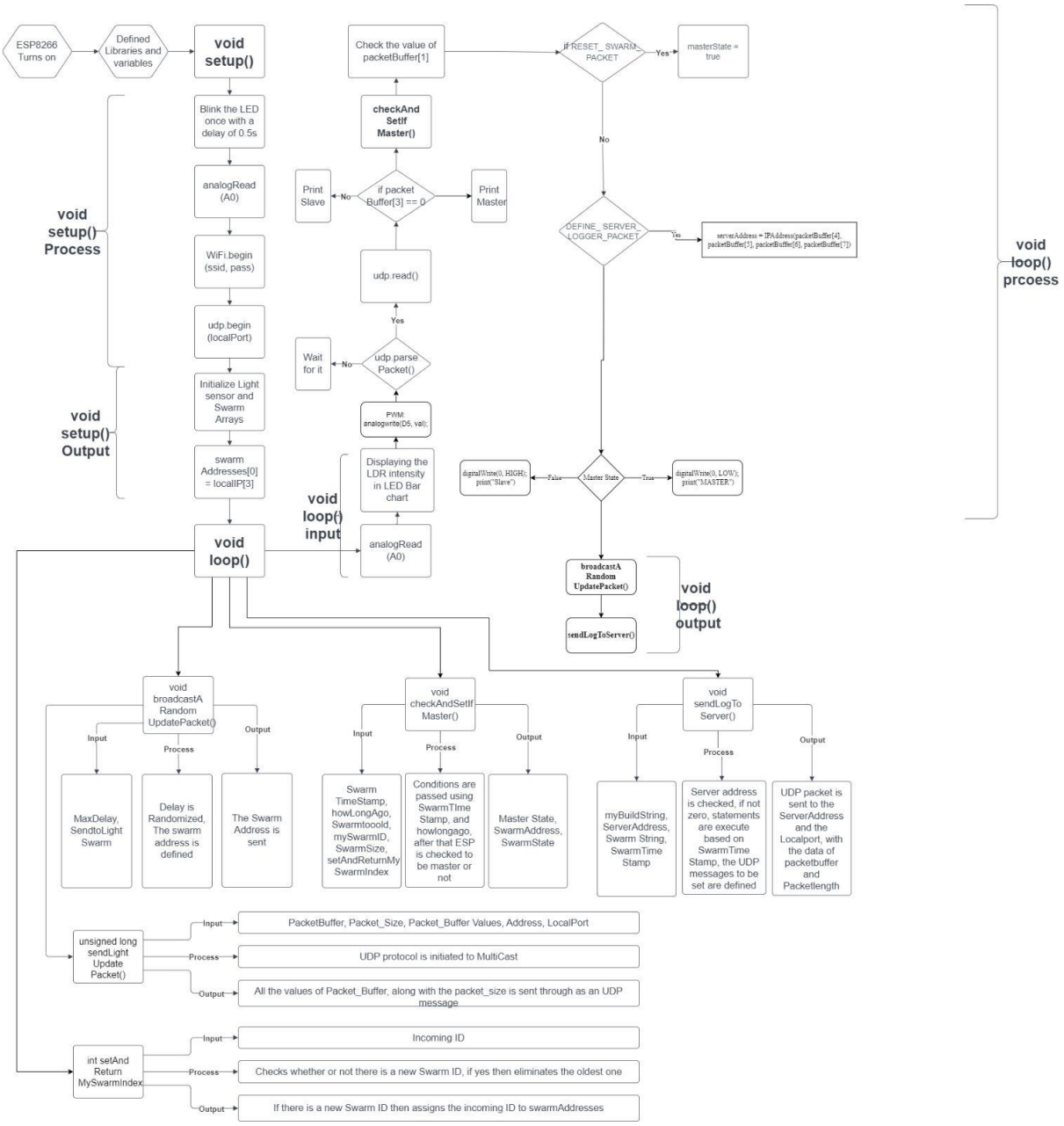
Canny_backup.txt | log_file_2022-...58346-08:00.txt | log_file_2022-...87809-08:00.txt | log_file_2022-...45562-08:00.txt

```
IP Address's of Masters: 192.168.43.41, 192.168.43.171,


192.168.2.41 was master for 13.614874839782715 seconds
192.168.2.171 was master for 7.997210264205933 seconds


Time Stamp              Master IP       Master LDR      Raw Data
1668830899.3126562      41              42               0,1,28,42,PR,41 | 1,0,28,30,PR,171 | 2,1,28,42,PR,68 | 3,0,0
1668830902.831855       171             37               0,1,28,37,PR,171 | 1,0,28,29,PR,41 | 2,0,28,6,PR,68 | 3,0,0,
1668830904.4483721      171             37               0,1,28,37,PR,171 | 1,0,28,28,PR,41 | 2,0,28,6,PR,68 | 3,0,0,
1668830907.623582       41              28               0,1,28,28,PR,41 | 1,0,28,9,PR,171 | 2,0,28,0,PR,68 | 3,0,0,0
1668830908.8698156      41              30               0,1,28,30,PR,41 | 1,0,28,9,PR,171 | 2,0,28,4,PR,68 | 3,0,0,0
1668830910.805815       41              31               0,1,28,31,PR,41 | 1,0,28,9,PR,171 | 2,0,28,2,PR,68 | 3,0,0,0
1668830912.2636194      41              16               0,1,28,16,PR,41 | 1,0,28,10,PR,171 | 2,0,28,4,PR,68 | 3,0,0,
1668830914.912052       171             33               0,1,28,33,PR,171 | 1,1,28,20,PR,41 | 2,0,28,8,PR,68 | 3,0,0,
1668830915.7218852      41              29               0,1,28,29,PR,41 | 1,0,28,22,PR,171 | 2,0,28,7,PR,68 | 3,0,0,
1668830918.0345802      171             30               0,1,28,30,PR,171 | 1,1,28,30,PR,41 | 2,0,28,9,PR,68 | 3,0,0,
1668830919.7499056      171             40               0,1,28,40,PR,171 | 1,0,28,28,PR,41 | 2,0,28,2,PR,68 | 3,0,0,
1668830920.4302304      41              37               0,1,28,37,PR,41 | 1,1,28,37,PR,171 | 2,0,28,8,PR,68 | 3,0,0,
1668830920.9247413      171             40               0,1,28,40,PR,171 | 1,1,28,31,PR,41 | 2,1,28,39,PR,68 | 3,0,0
```

Graphs obtained from the Node RED



ESP WiFi setup and packet delivery



WiFi connectivity State: In this state ESP8266 will wait to be connected with specific LAN (Local Ara Network) specified as SSID. After getting connected with the network ESP8266 will get its IP address.

Sensor Reading state: In this state ESP will read data from LDR.

Computational State: Most work will be done here. Onboard LED will be turned on-off according to LDR data.

Communication State: In this state ESP8266 will broadcast the LDR reading.

Master State: If ESP is in master it will collect data from the all other ESPs and send combined data to the raspberry pi.

Termination state: If ESP get RESET UDP packet from the raspberry pi then it will reset.

Main functionality of ESP8266 is to establish secure UDP communication between Raspberry Pi and ESP8266. It broadcast the LDR data to the network. ESP will also turn on-ff the onboard LED depending on the LDR data. It also collect data from other ESP and compare LDR value to be the master. If ESP is master then it sends all the LDR data to the Raspberry Pi and turn on other Onboard LED. At the last ESP will reset its state on getting RESET UDP packet.