ASSIGNMENT-4

ESP SWARM AND LED BRIGHTNESS USING PWM

Vishwanath Singh

73975792

RASPBERRY PI FUNCTION AND EXPLATION:

In the raspberry pi code, the raspberry pi waits to receive a message from the master ESP which sends a packet with all the necessary information (master/slave state, timestamp, reading). The raspberry pi takes this data and processes it and retrieves information. The raspberry pi then displays the 3 graphs, one for the values the sensor receives and the second for the number of times a particular esp has been the master. This keeps updating by taking every 30 seconds value. As the master changes, a new bar graph in a different color starts showing, with its corresponding data values. The reset button this time turns on a yellow LED instead of a white as in the last assignment. It also cleans the graph or clears it along with creating a new log file with the current date and time as the name of the log file. The log file contains data about the ESP, which one is the master with its ID, the time duration that it has been the master and the raw data that it had sent to the raspberry pi.
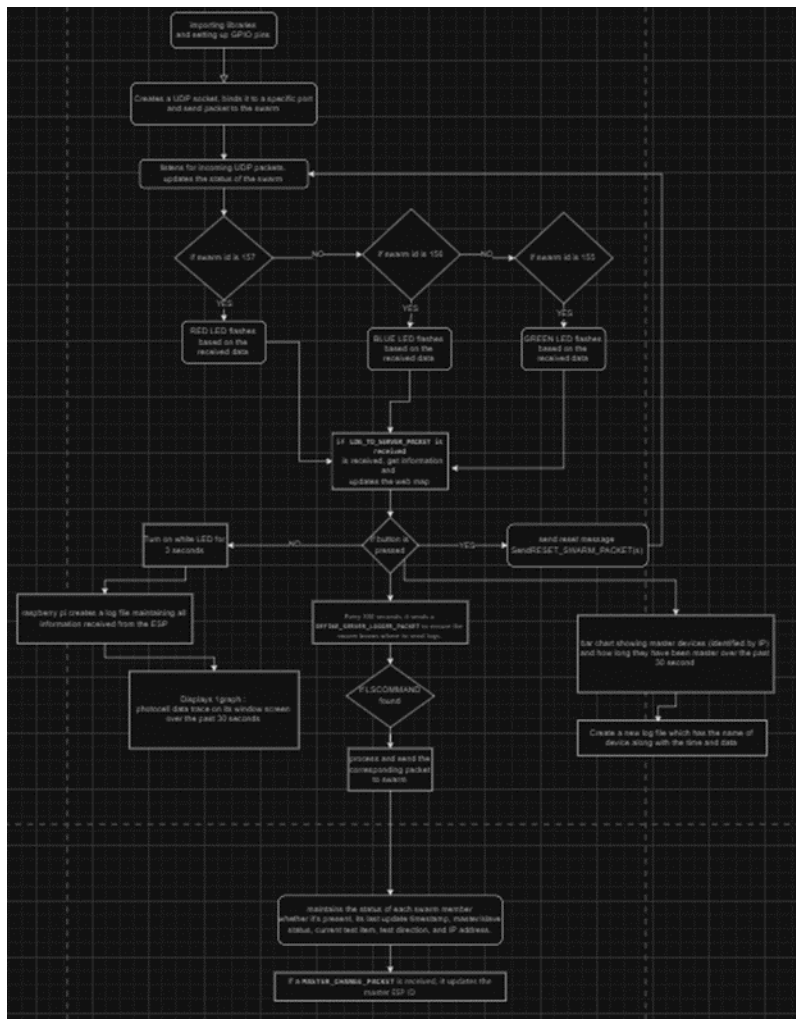

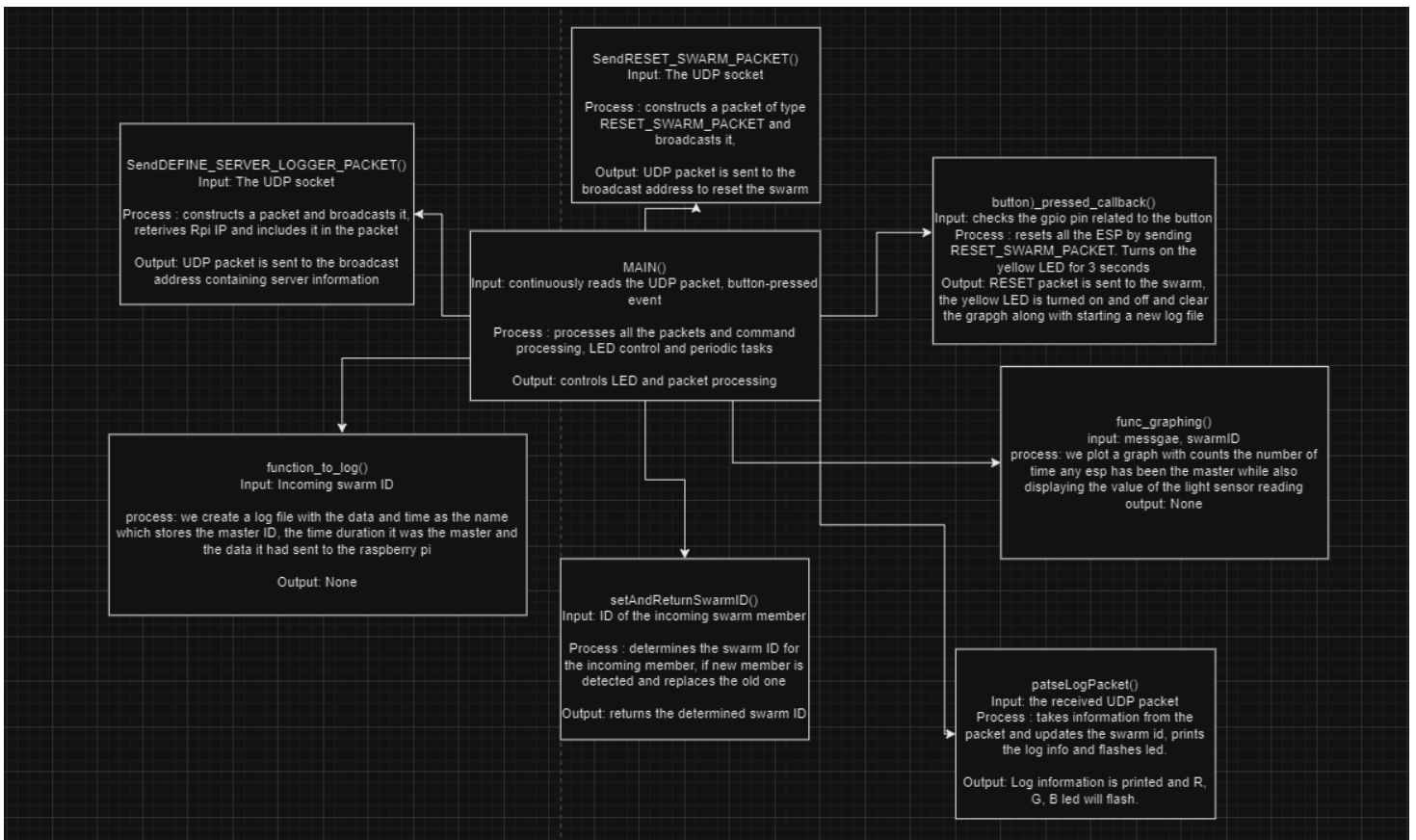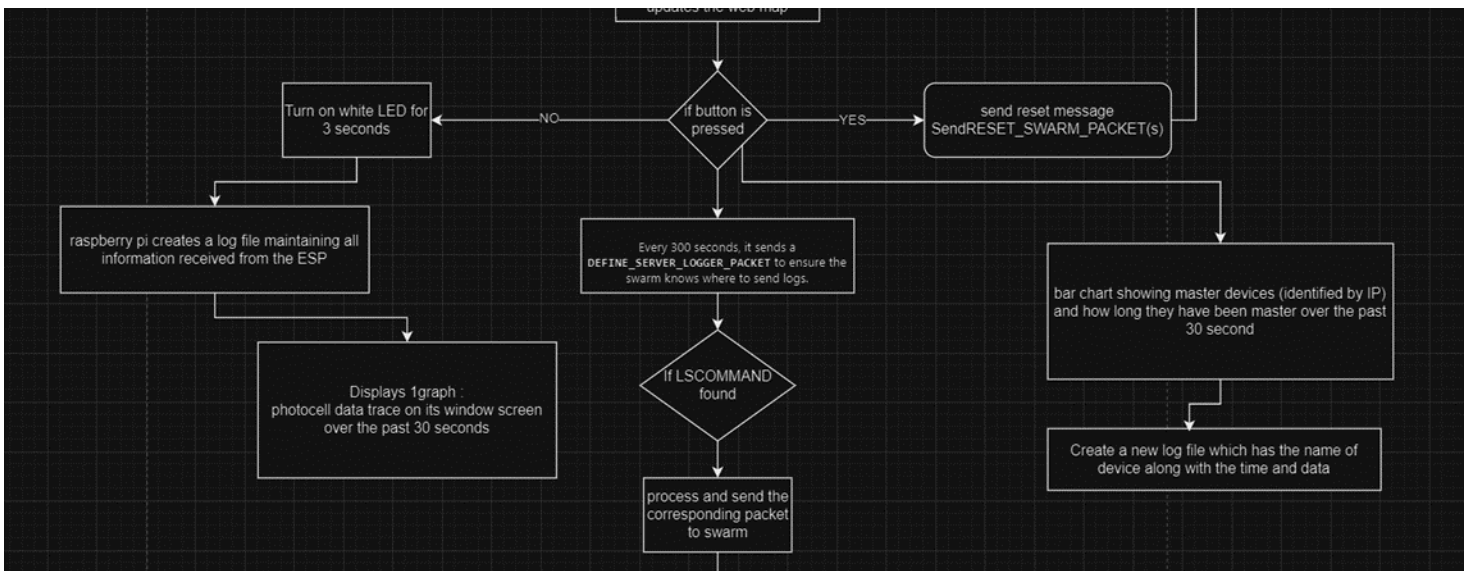Figure 1: Flow chart for raspberry pi code.

Figure 2: the flowchart containing all the input, process, and the output of all the functions.

Diagram for communication between raspberry pi and ESP
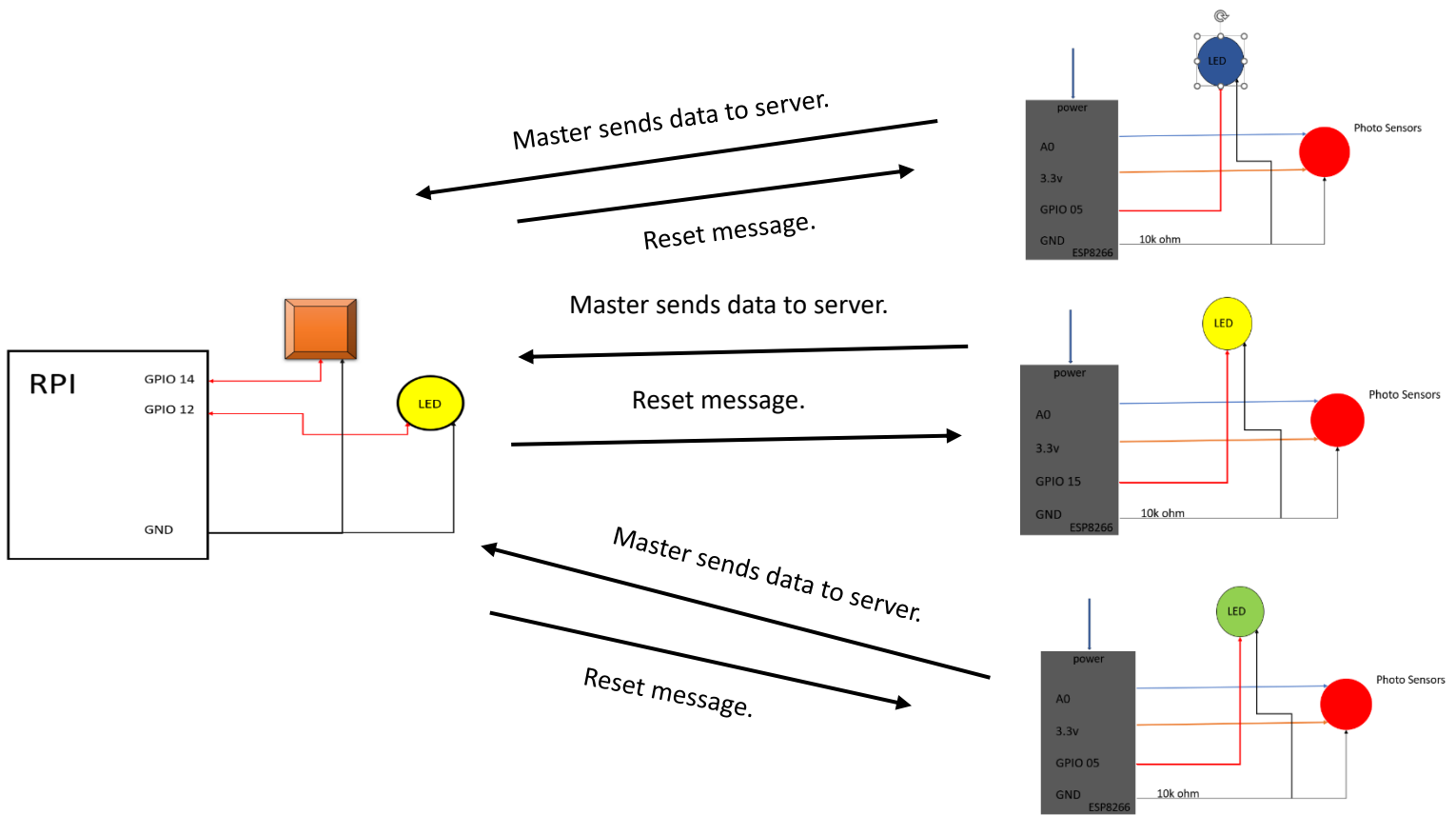
CODE FOR ESP:

There are 3 ESP8266s', they have 3 individual lights sensors on the circuit along with 3 independent LEDS. Now, as the esp8266 starts reading values from the sensor, they start communicating among themselves and they determine who the master is based on which sensor has the highest value. Once the Esp8266 with the highest sensor value gets assigned the master, the on-board LED of that esp8266 turns on. The master esp8266 then sends the data from the sensor along with its ID to the raspberry pi which is the server where the data is displayed on the graph. The Led on connected to the Esp8266 starts turning on, and the brightness of the led keeps increasing when the light source is brought close to it. Once the light source is taken away from it, the brightness of the Led decreases. The brightness control of the led connected to the Esp8266 is controlled by the idea of pulse width modulation. This modulation generates variable-width pulses to represent the amplitude of an analog input signal.

checkAndSetIfMaster()

Input:
*Reading Packet buffer information

none
Process:
*Print all the swarm information like swarm ID, master/slave status.
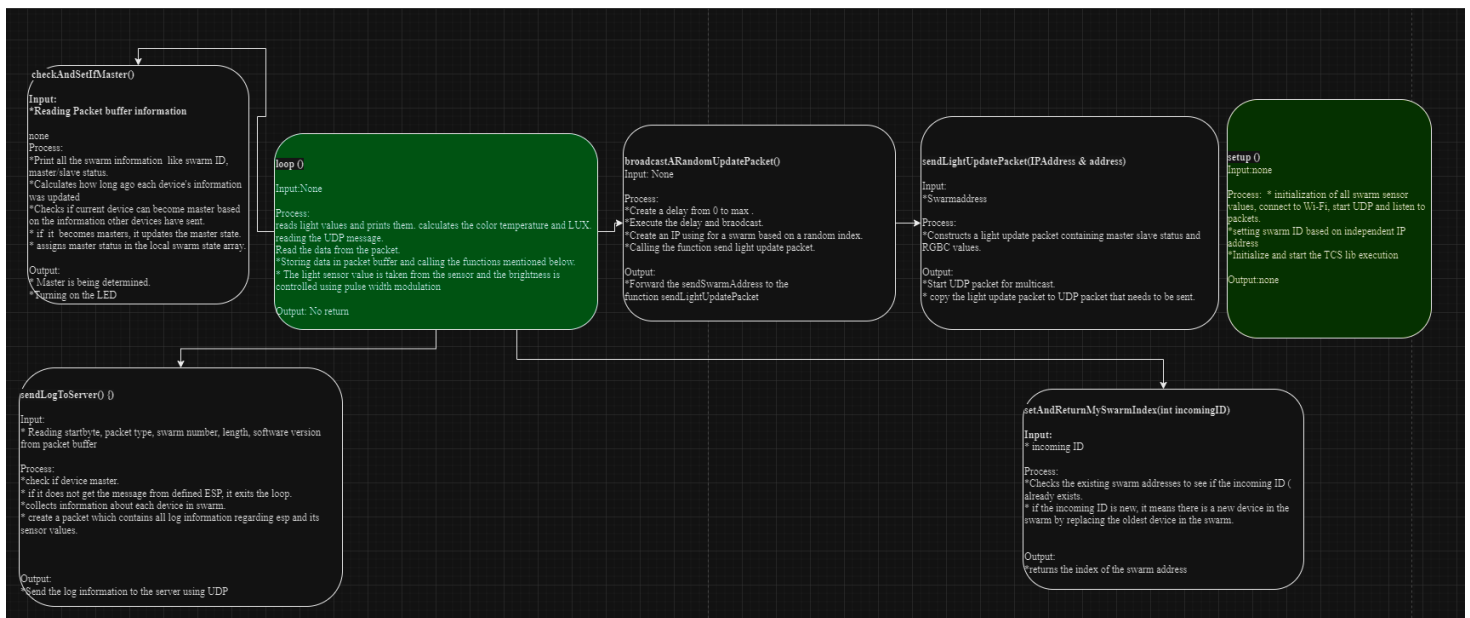*Calculates how long ago each device's information was updated
*Checks if current device can become master based on the information other devices have sent.
* if it becomes masters, it updates the master state.
* assigns master status in the local swarm state array.

Output:
* Master is being determined.
*Turning on the LED

loop ()

Input:None

Process:
reads light values and prints them. calculates the color temperature and LUX.
reading the UDP message.
Read the data from the packet.
*Storing data in packet buffer and calling the functions mentioned below.
* The light sensor value is taken from the sensor and the brightness is controlled using pulse width modulation

Output: No return

broadcastARandomUpdatePacket()

Input: None

Process:
*Create a delay from 0 to max .
*Execute the delay and braodcast.
*Create an IP using for a swarm based on a random index.
*Calling the function send light update packet.

Output:
*Forward the sendSwarmAddress to the function sendLightUpdatePacket

sendLightUpdatePacket(IPAddress & address)

Input:
*Swarmaddress

Process:
*Constructs a light update packet containing master slave status and RGBC values.

Output:
*Start UDP packet for multicast.
* copy the light update packet to UDP packet that needs to be sent.

setup ()

Input:none

Process: * initialization of all swarm sensor values, connect to Wi-Fi, start UDP and listen to packets.
*setting swarm ID based on independent IP address
*Initialize and start the TCS lib execution

Output:none

sendLogToServer() {}

Input:
* Reading startbyte, packet type, swarm number, length, software version from packet buffer

Process:
*check if device master.
* if it does not get the message from defined ESP, it exits the loop.
*collects information about each device in swarm.
* create a packet which contains all log information regarding esp and its sensor values.

Output:
*Send the log information to the server using UDP

setAndReturnMySwarmIndex(int incomingID)

Input:
* incoming ID

Process:
*Checks the existing swarm addresses to see if the incoming ID (already exists.
* if the incoming ID is new, it means there is a new device in the swarm by replacing the oldest device in the swarm.

Output:
*returns the index of the swarm address

Figure 3: flowchart for ESP as submitted in small assignment -5

Video: https://drive.google.com/file/d/1hVPIl4Sjr5Zuy8HyNxTQhYl2diN80PuV/view?usp=sharing