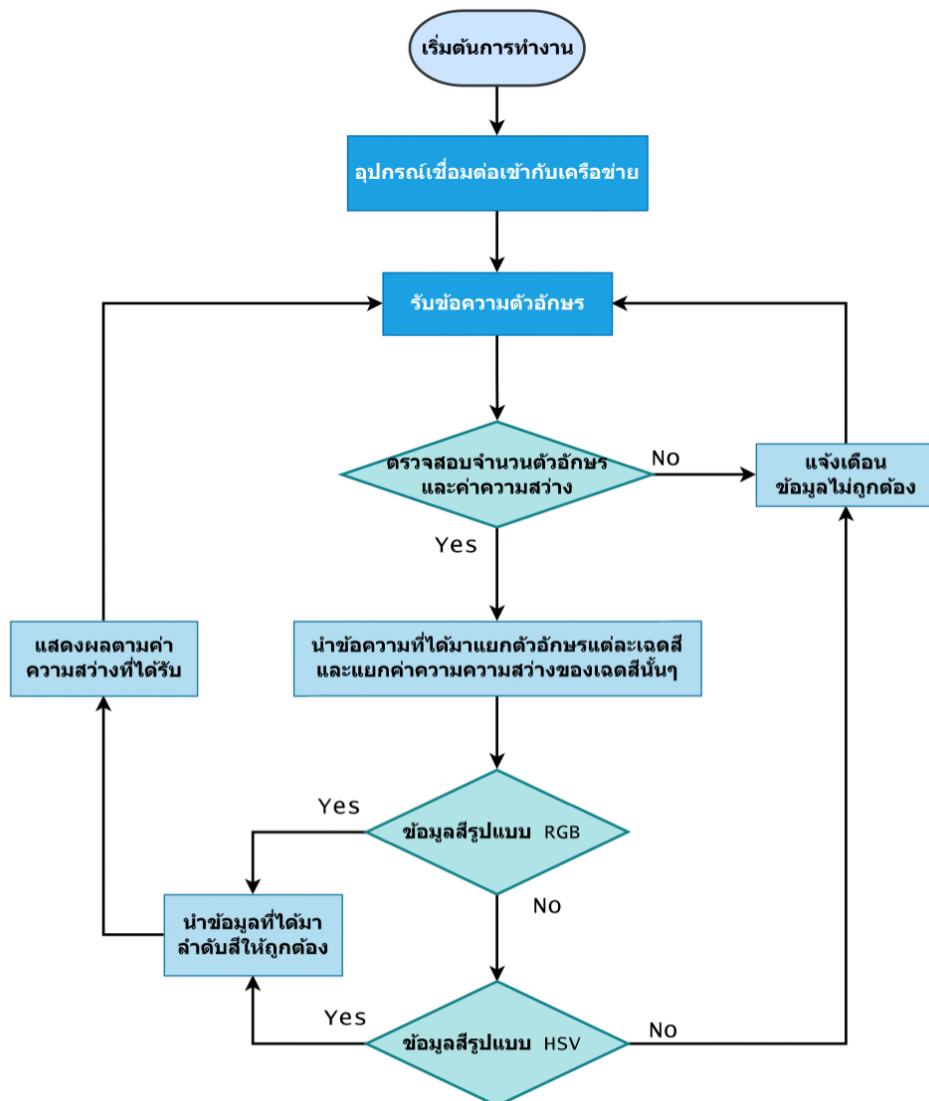


### Lab7 IO-Control RGB&HSV

เริ่มการทำงานโดยบอร์ดทำการเชื่อมต่อเครือข่ายอินเทอร์เน็ต จากนั้นรับข้อความจากโปรแกรม Hercules SETUP utility มาตรวจสอบจำนวนตัวอักษร ค่าความสว่างของโทนสีที่ได้รับ และตรวจสอบรูปแบบของข้อมูลว่าเป็น RGB หรือ HSV จากนั้นลำดับการแสดงผล และแสดงผลที่ Neo Pixel โดยข้อมูลแบบ RGB แสดงผลตามลำดับสีคือ แดง เขียว และน้ำเงิน ตามค่าความความสว่างที่รับในแต่ละเฉดสี และข้อมูลแบบ HSV แสดงสีตามค่าสีที่ได้รับมา หากข้อความที่ได้รับไม่ถูกต้องตามเงื่อนไขที่กำหนดจะมีการแจ้งเตือน เพื่อให้ผู้ใช้ได้ตรวจสอบข้อความที่ได้ส่งมา และทำการแก้ไขให้ถูกต้อง



### อธิบายการทำงานของโปรแกรม

**// My WiFi ----- เรียกใช้และกำหนดตัวแปรสำหรับการเชื่อมต่อ WiFi ของบอร์ด**

#include <WiFi.h>

#define SERVER\_PORT 9999

const char\* ssid = "IoT family 2.4G"; // ID

const char\* password = "11550088"; // Password

WiFiServer my\_server(SERVER\_PORT);

**//NeoPixel ----- เรียกใช้และกำหนดตัวแปรกำหนดหลอดไฟ NeoPixel**

#include <Adafruit\_NeoPixel.h>

#define NeoPixel\_PIN 18

#define NUMPIXELS 1

Adafruit\_NeoPixel pixels(NUMPIXELS, NeoPixel\_PIN, NEO\_GRB + NEO\_KHZ800);

**//String define Color----- กำหนดตัวแปรรับข้อความเฉดสี**

String color\_one; //Check first Color

String color\_two; //Check Second Color

String color\_three; //Check Third Color

**//String define Value-----กำหนดตัวแปรรับค่าความสว่าง**

String sval\_one; // Check String Value

String sval\_two;

String sval\_three;

int val\_one = 0; // Convert Value form String to Integer

int val\_two = 0;

int val\_three = 0;

int str\_L = 0; // Message Length | ตัวแปรรับค่าจำนวนตัวอักษรในข้อความ

**//----- ส่วนเริ่มต้นการเชื่อมต่อ WiFi ของบอร์ด และกำหนดค่าของพอร์ตอนุกรมเพื่อแสดงข้อความ**

void setup() {

pixels.begin();

Serial.begin(115200);

WiFi.begin(ssid,password);

while(WiFi.status() != WL\_CONNECTED){

```

    delay(500);
    Serial.print(".");
}
Serial.println("WiFi Connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
my_server.begin();
Serial.println("Server started");
}

//----- คำสั่งการทำงานแบบวนลูป -----
void loop() {
    WiFiClient my_client = my_server.available();
    // Pixel Off ----- สั่งให้ NeoPixel ดับในช่วงเริ่มต้นของลูป
        pixels.setPixelColor(0, pixels.Color(0, 0, 0));
        pixels.show();
    //-----แสดงขั้นตอนการใช้งานเมื่อมีการเชื่อมต่อจากโปรแกรม Hercules SETUP utility
    if(my_client){
        Serial.println("New client connected");
        Serial.println("=====");
        Serial.println("[Read Me]");
        Serial.println("Neo Pixel Color Control");
        Serial.println("Set Your Color With R,G,B or H,S,V");
        Serial.println("Set Your Color Value With [000-255]");
        Serial.println("|Ex1 : RGB| R:010,G:200,B:003 ");
        Serial.println("|Ex2 : HSV| H:010,S:200,V:003 ");
        Serial.println("=====");
        //---- รับข้อความนำมาเก็บไว้ที่ตัวแปร msg -----
        while(true){
            while(my_client.available()){
                String msg = my_client.readString();

```

```

// Check Number of String -----นำจำนวนตัวอักษรมาเก็บไว้ในตัวแปร str_L
str_L = msg.length();
//String Get Color ----- เก็บสีจากข้อความเป็นตัวอักษร (R,G,B or H,S,V)
color_one = msg[0];      // สีที่หนึ่ง
color_two = msg[6];      // สีที่สอง
color_three = msg[12];   // สีที่สาม
//String Get Value ----- รับค่าความสว่างของเฉดสีเป็นตัวอักษร
sval_one = msg.substring(2,5);    // ค่าของสีที่หนึ่ง
sval_two = msg.substring(8,11);   // ค่าของสีที่สอง
sval_three = msg.substring(14,17); // ค่าของสีที่สาม
// Convert Value to Int----- นำค่าความสว่างมาแปลงเป็นตัวเลข
val_one = sval_one.toInt();
val_two = sval_two.toInt();
val_three = sval_three.toInt();
//----- ตรวจสอบจำนวนตัวอักษร และค่าความสว่างที่ได้รับ
if(str_L == 17){ if(val_one <= 255 && val_two <= 255 && val_three <= 255){
  Serial.println("[Data Correct]");
  //----- แสดงข้อความที่ได้รับ
  Serial.print("Your Neo Pixel Color : ");
  Serial.println(msg);
  //ตรวจสอบรูปแบบข้อมูลสีที่รับ จากนั้นทำการนำข้อมูลเฉดสีแบบ RGB มาลำดับ และแสดงผล
  RGB=====
  if(color_one == "R" && color_two == "G" && color_three == "B")    // 123
  {
    pixels.setPixelColor(0, pixels.Color(val_one, 0, 0));  pixels.show();  delay(500);
    pixels.setPixelColor(0, pixels.Color(0, val_two, 0));  pixels.show();  delay(500);
    pixels.setPixelColor(0, pixels.Color(0, 0, val_three)); pixels.show();  delay(500);
  }
  else if(color_one == "R" && color_two == "B" && color_three == "G") // 132
  {

```

```

pixels.setPixelColor(0, pixels.Color(val_one, 0,0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, val_three, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, 0, val_two)); pixels.show(); delay(500);
}
else if(color_one == "G" && color_two == "B" && color_three == "R") //312
{
pixels.setPixelColor(0, pixels.Color(val_three, 0, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, val_one, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, 0, val_two)); pixels.show(); delay(500);
}
else if(color_one == "G" && color_two == "R" && color_three == "B") //213
{
pixels.setPixelColor(0, pixels.Color(val_two, 0, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, val_one, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, 0, val_three)); pixels.show(); delay(500);
}
else if(color_one == "B" && color_two == "G" && color_three == "R") //321
{
pixels.setPixelColor(0, pixels.Color(val_three, 0, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, val_two, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, 0, val_one)); pixels.show(); delay(500);
}
else if(color_one == "B" && color_two == "R" && color_three == "G") //231
{
pixels.setPixelColor(0, pixels.Color(val_two, 0, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, val_three, 0)); pixels.show(); delay(500);
pixels.setPixelColor(0, pixels.Color(0, 0, val_one)); pixels.show(); delay(500);
}

```

## HSV=====

```

else if(color_one == "H" && color_two == "S" && color_three == "V")    // 123
{
    pixels.setPixelColor(0, pixels.Color(val_one*257, val_two*257, val_three*257));
pixels.show(); delay(1000);
}
else if(color_one == "H" && color_two == "V" && color_three == "S") // 132
{
    pixels.setPixelColor(0, pixels.Color(val_one*257, val_three*257, val_two*257));
pixels.show(); delay(1000);
}
else if(color_one == "S" && color_two == "V" && color_three == "H") //312
{
    pixels.setPixelColor(0, pixels.Color(val_three*257, val_one*257, val_two*257));
pixels.show(); delay(1000);
}
else if(color_one == "S" && color_two == "H" && color_three == "V") //213
{
    pixels.setPixelColor(0, pixels.Color(val_two*257, val_one*257, val_three*257));
pixels.show(); delay(1000);
}
else if(color_one == "V" && color_two == "S" && color_three == "H") //321
{
    pixels.setPixelColor(0, pixels.Color(val_three*257, val_two*257, val_one*257));
pixels.show(); delay(1000);
}
else if(color_one == "V" && color_two == "H" && color_three == "S") //231
{

```

```
pixels.setPixelColor(0, pixels.Color(val_two*257, val_three*257, val_one*257));
pixels.show(); delay(1000); }
```

**//---- แจ้งเตือนเมื่อข้อความที่ได้รับไม่ถูกต้องตามเงื่อนไข**

```
}} else {Serial.println("[Invalid!!] : Check Your Message");
        Serial.println("=====");
        Serial.println("Set Your Color With R,G,B or H,S,V");
        Serial.println("Set Your Color Value With [000-255]");
        Serial.println("|Example:RGB| R:010,G:200,B:003 ");
        Serial.println("|Example:HSV| V:000,S:255,H:255 ");
        Serial.println(" ");}
```

**// Client disconnected-----|| แสดงข้อความเมื่อมีการเชื่อมต่อใหม่จาก Client**

```
if(my_server.hasClient())
{
    Serial.println("Client disconnected");
    return;
}
```

**// Pixel Off -----|| สั่งให้ NeoPixel ดับหลังจากการแสดงผล**

```
pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();
}}
```