ECPS 216 IoT System and Software

Assignment – 1

Vishwanath Singh

73975792

Part – 1

For part 1 we create a sliding window to calculate average. If the average is higher than the threshold the LED is turned on. We start by declaring the integration time to 50 milliseconds and the gain at 4X. then initialise the value for red, blue green average variables. I set the threshold value to 150 as the average value fluctuates around 145-155. I created an array to store values of the averages calculated along with the window size of 4. A variable named currentIndex to keep a track of which array to overwrite in the array. This acts as my sliding window. As the 5th value is read the value at the 0th index is overwritten by the new value, which is the 5th value. Then we set the baud rate and set up the built in LED as output. I, created a function called caltotal which calculates the total value of an array and returns the average of the 4 values in the array. These individual value at every index is calculated separately. These RGB values are read from the sensor and then averaged and added to the array at the current index location. Then the caltotla function returns the total value to a variable called finalavg. This final avg is compared to the threshold value and if it is higher than the LED is turned on by setting it as LOW, since esp8266 works on active-low. At the end I update the current index value and the loop runs again.

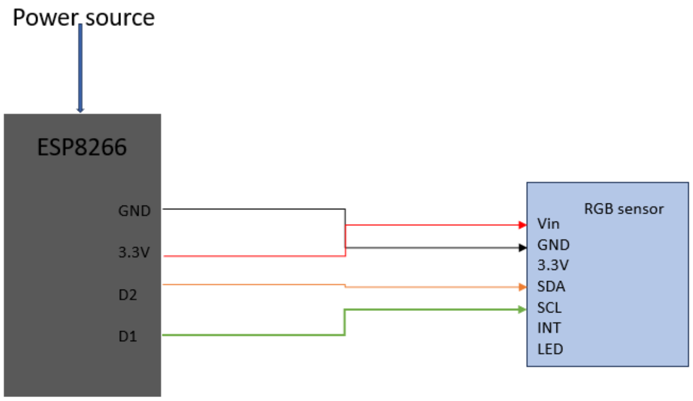
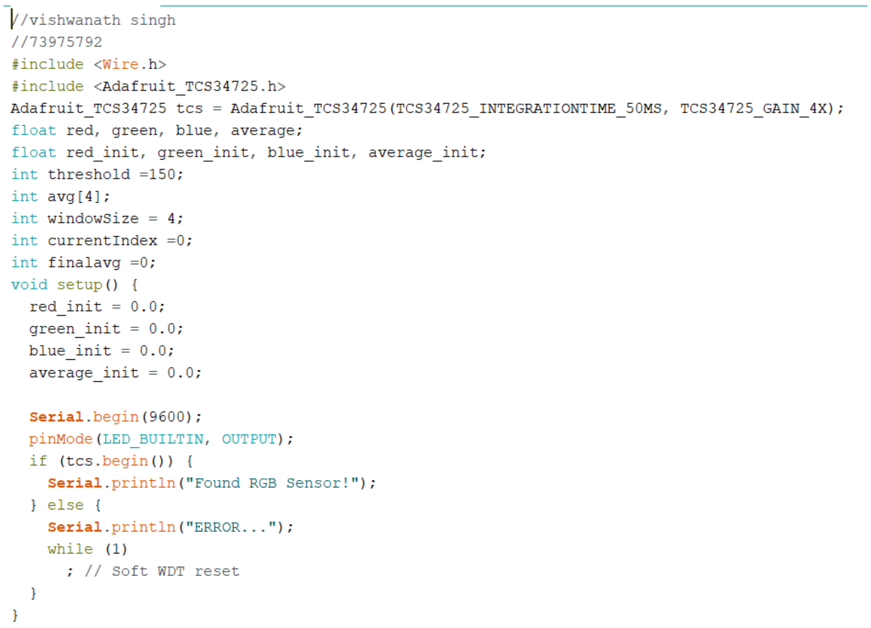
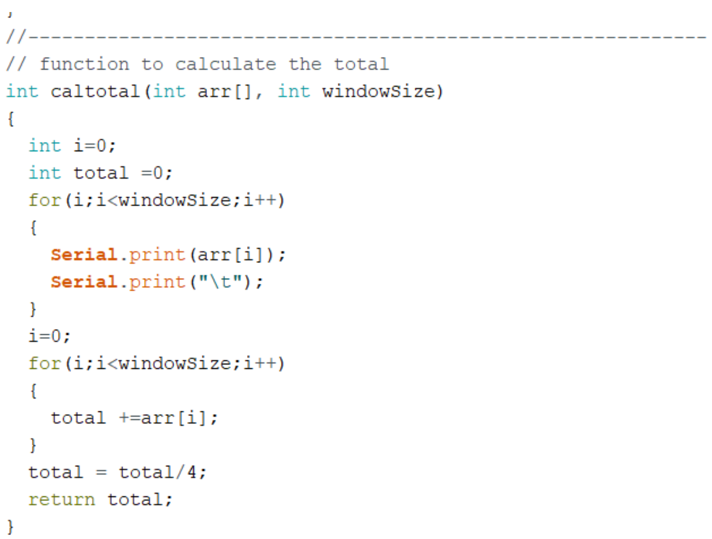


Figure 1: Circuit for turning built in LED on if average value RGB value each is greater than the threshold.

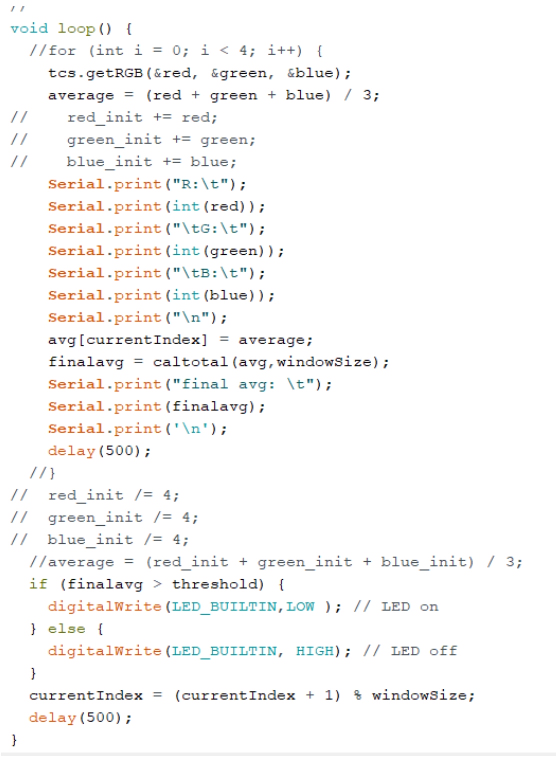


```
//vishwanath singh
//73975792
#include <Wire.h>
#include <Adafruit_TCS34725.h>
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_50MS, TCS34725_GAIN_4X);
float red, green, blue, average;
float red_init, green_init, blue_init, average_init;
int threshold =150;
int avg[4];
int windowSize = 4;
int currentIndex =0;
int finalavg =0;
void setup() {
  red_init = 0.0;
  green_init = 0.0;
  blue_init = 0.0;
  average_init = 0.0;

  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  if (tcs.begin()) {
    Serial.println("Found RGB Sensor!");
  } else {
    Serial.println("ERROR...");
    while (1)
      ; // Soft WDT reset
  }
}
```



```
,
//-----------------------------------------------------------
// function to calculate the total
int caltotal(int arr[], int windowSize)
{
  int i=0;
  int total =0;
  for(i;i<windowSize;i++)
  {
    Serial.print(arr[i]);
    Serial.print("\t");
  }
  i=0;
  for(i;i<windowSize;i++)
  {
    total +=arr[i];
  }
  total = total/4;
  return total;
}
```



```
//
void loop() {
  //for (int i = 0; i < 4; i++) {
    tcs.getRGB(&red, &green, &blue);
    average = (red + green + blue) / 3;
//    red_init += red;
//    green_init += green;
//    blue_init += blue;
    Serial.print("R:\t");
    Serial.print(int(red));
    Serial.print("\tG:\t");
    Serial.print(int(green));
    Serial.print("\tB:\t");
    Serial.print(int(blue));
    Serial.print("\n");
    avg[currentIndex] = average;
    finalavg = caltotal(avg,windowSize);
    Serial.print("final avg: \t");
    Serial.print(finalavg);
    Serial.print('\n');
    delay(500);
  //}
//  red_init /= 4;
//  green_init /= 4;
//  blue_init /= 4;
  //average = (red_init + green_init + blue_init) / 3;
  if (finalavg > threshold) {
    digitalWrite(LED_BUILTIN,LOW ); // LED on
  } else {
    digitalWrite(LED_BUILTIN, HIGH); // LED off
  }
  currentIndex = (currentIndex + 1) % windowSize;
  delay(500);
}
```

Part – 2

For the second part of the assignment, we assign GPIO pins 12, 13 and 14 to yellow, green, and red LED respectively. In void setup function we set these LEDs as output using pinMode(ColourLED_PIN, OUTPUT). Then we measure the intensity of light using an in built function called tcs.calculateLUX. This function gives us the intensity of light which we use to turn green LED if the intensity is less than 50. The yellow LED is turned on when the intensity is less than 100. If the intensity is greater than 100 the red LED is turned on.

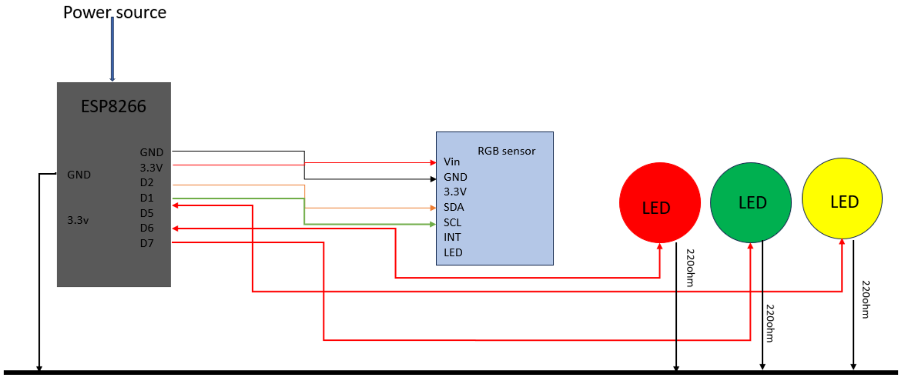
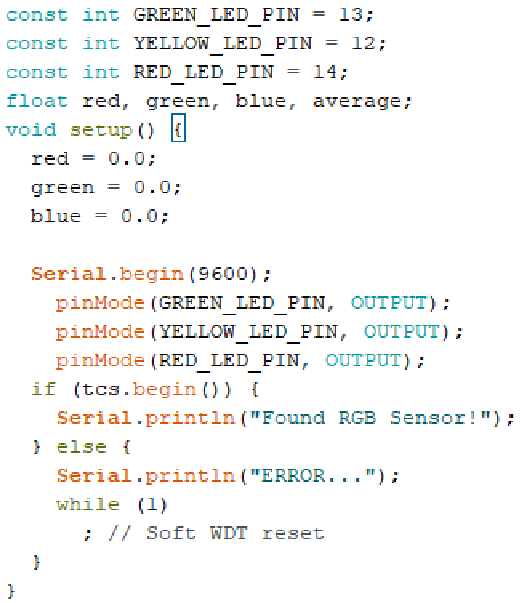
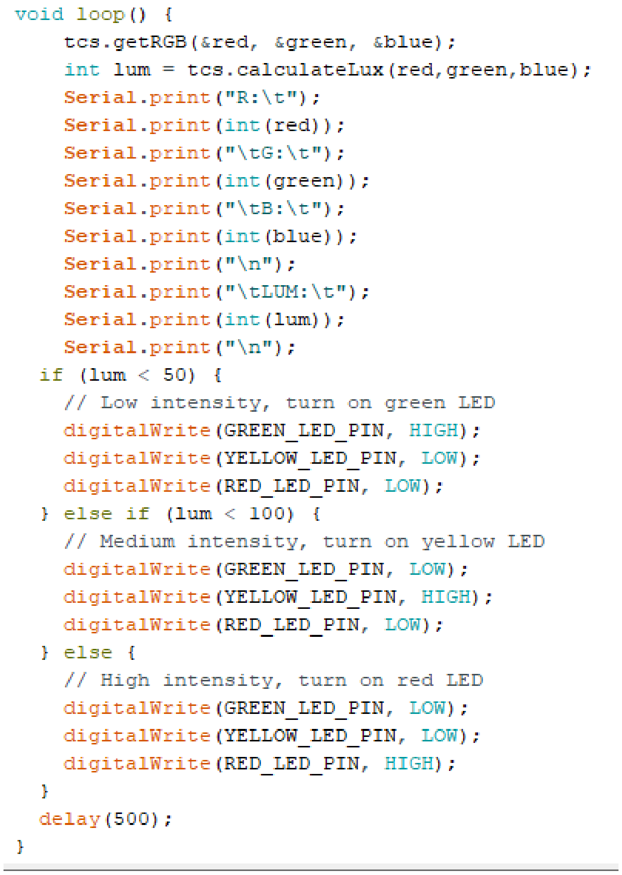


Figure 2: Circuit for turning colored LED on depending on the intensity of the light.



```
const int GREEN_LED_PIN = 13;
const int YELLOW_LED_PIN = 12;
const int RED_LED_PIN = 14;
float red, green, blue, average;
void setup() {
  red = 0.0;
  green = 0.0;
  blue = 0.0;

  Serial.begin(9600);
    pinMode(GREEN_LED_PIN, OUTPUT);
    pinMode(YELLOW_LED_PIN, OUTPUT);
    pinMode(RED_LED_PIN, OUTPUT);
  if (tcs.begin()) {
    Serial.println("Found RGB Sensor!");
  } else {
    Serial.println("ERROR...");
    while (1)
      ; // Soft WDT reset
  }
}
```

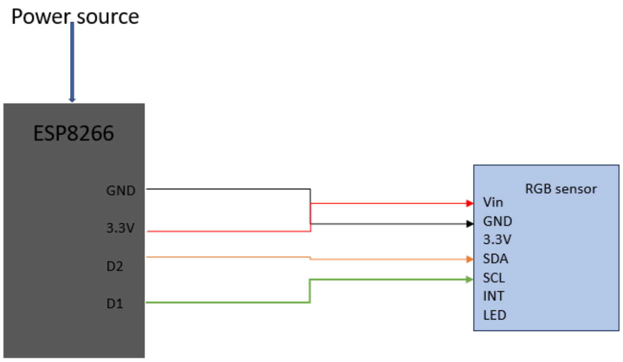


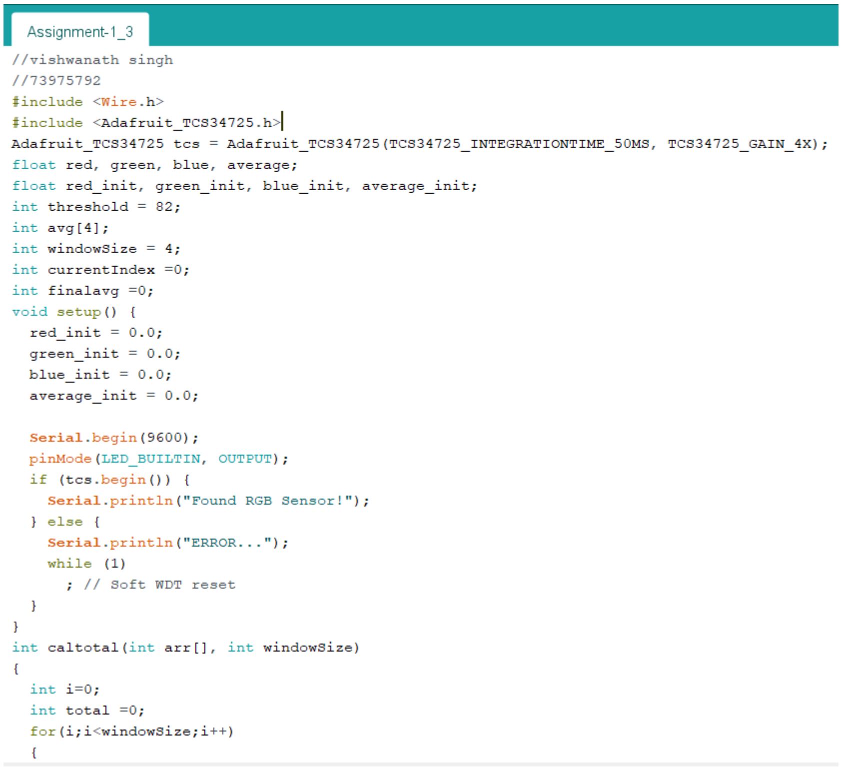
```
void loop() {
    tcs.getRGB(&red, &green, &blue);
    int lum = tcs.calculateLux(red,green,blue);
    Serial.print("R:\t");
    Serial.print(int(red));
    Serial.print("\tG:\t");
    Serial.print(int(green));
    Serial.print("\tB:\t");
    Serial.print(int(blue));
    Serial.print("\n");
    Serial.print("\tLUM:\t");
    Serial.print(int(lum));
    Serial.print("\n");
  if (lum < 50) {
    // Low intensity, turn on green LED
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(YELLOW_LED_PIN, LOW);
    digitalWrite(RED_LED_PIN, LOW);
  } else if (lum < 100) {
    // Medium intensity, turn on yellow LED
    digitalWrite(GREEN_LED_PIN, LOW);
    digitalWrite(YELLOW_LED_PIN, HIGH);
    digitalWrite(RED_LED_PIN, LOW);
  } else {
    // High intensity, turn on red LED
    digitalWrite(GREEN_LED_PIN, LOW);
    digitalWrite(YELLOW_LED_PIN, LOW);
    digitalWrite(RED_LED_PIN, HIGH);
  }
  delay(500);
}
```

Part – 3

For the third part of the assignment, the code is more or less the same as part 1. The changes made in this code are:

1. When final average is calculated, and compared to the threshold, I used a map function to scale the value of final average if it is between the threshold and 255. If is it between this range the value is scaled between 250 and 1. That means the higher the value of final avg is the smaller then number will be between 250 and 1.
2. Once scaling is done, the value is stored in a variable called temp. this variable is used to control the time of delay. The value 250 is taken since the reading of the RGB values should be 50ms, the blinking delay also needs to be taken into account. Therefore the 2 delays add upto 500ms when then LED is blinking, which maintains the 50ms reading period.
3. Now, if the final avg is less then the threshold the LED will remain off and the delay is for 50ms.





```
//vishwanath singh
//73975792
#include <Wire.h>
#include <Adafruit_TCS34725.h>
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_50MS, TCS34725_GAIN_4X);
float red, green, blue, average;
float red_init, green_init, blue_init, average_init;
int threshold = 82;
int avg[4];
int windowSize = 4;
int currentIndex =0;
int finalavg =0;
void setup() {
  red_init = 0.0;
  green_init = 0.0;
  blue_init = 0.0;
  average_init = 0.0;

  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  if (tcs.begin()) {
    Serial.println("Found RGB Sensor!");
  } else {
    Serial.println("ERROR...");
    while (1)
      ; // Soft WDT reset
  }
}
int caltotal(int arr[], int windowSize)
{
  int i=0;
  int total =0;
  for(i;i<windowSize;i++)
  {
    total +=arr[i];
  }
  total = total /4;
  return total;
}
//----------------------------------------------------------------
void loop() {
    tcs.getRGB(&red, &green, &blue);
    average = (red + green + blue) / 3;
    Serial.print("R:\t");
    Serial.print(int(red));
    Serial.print("\tG:\t");
    Serial.print(int(green));
    Serial.print("\tB:\t");
    Serial.print(int(blue));
    Serial.print("\n");
    avg[currentIndex] = average;
    finalavg = caltotal(avg,windowSize);
//    total += (avg[currentIndex]);
Serial.print("Final Average is: \t"); Serial.print(finalavg);Serial.print("\n");
  if (finalavg > threshold) {
        int temp = map(finalavg, threshold, 255, 250, 1);
      digitalWrite(LED_BUILTIN, LOW); // LED on
      delay(temp);
      Serial.print("temp is:\t");
      Serial.print(temp);
      digitalWrite(LED_BUILTIN, HIGH); // LED off
      delay(temp);
      Serial.println("\tBLINK");Serial.print("\n");
  } else {
    digitalWrite(LED_BUILTIN, HIGH); // LED off
    delay(500);
    Serial.println("\tOFF");Serial.print("\n"); //LED off
  }
  currentIndex = (currentIndex + 1) % windowSize;
  //delay(250);

}
```