// Define color sensor pins

#define S0 4

#define S1 5

#define S2 6

#define S3 7

#define sensorOut 8

// Variables for Color Pulse Width Measurements

int redPW = 0;

int greenPW = 0;

int bluePW = 0;

void setup() {

// Set S0 - S3 as outputs

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

// Set Pulse Width scaling to 20%

digitalWrite(S0,HIGH);

digitalWrite(S1,LOW);

// Set Sensor output as input

pinMode(sensorOut, INPUT);

// Setup Serial Monitor

Serial.begin(9600);

}

void loop() {

// Read Red Pulse Width

redPW = getRedPW();

// Delay to stabilize sensor

delay(200);

// Read Green Pulse Width

greenPW = getGreenPW();

// Delay to stabilize sensor

delay(200);

// Read Blue Pulse Width

bluePW = getBluePW();

// Delay to stabilize sensor

delay(200);

// Print output to Serial Monitor

Serial.print("Red PW = ");

Serial.print(redPW);

Serial.print(" - Green PW = ");

Serial.print(greenPW);

Serial.print(" - Blue PW = ");

Serial.println(bluePW);

}

// Function to read Red Pulse Widths

int getRedPW() {

// Set sensor to read Red only

digitalWrite(S2,LOW);

digitalWrite(S3,LOW);

// Define integer to represent Pulse Width

int PW;

// Read the output Pulse Width

PW = pulseIn(sensorOut, LOW);

// Return the value

return PW;

}

// Function to read Green Pulse Widths

int getGreenPW() {

// Set sensor to read Green only

digitalWrite(S2,HIGH);

digitalWrite(S3,HIGH);

// Define integer to represent Pulse Width

int PW;

// Read the output Pulse Width

PW = pulseIn(sensorOut, LOW);

// Return the value

return PW;

}

// Function to read Blue Pulse Widths

int getBluePW() {

// Set sensor to read Blue only

digitalWrite(S2,LOW);

digitalWrite(S3,HIGH);

// Define integer to represent Pulse Width

int PW;

// Read the output Pulse Width

PW = pulseIn(sensorOut, LOW);

// Return the value

return PW;

}