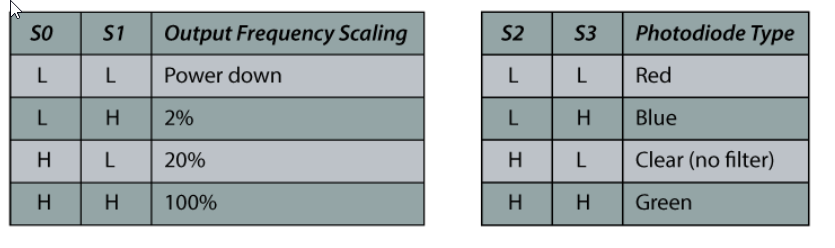
-Rédaction du planning et du cahier de charges

-Principalement, on s’est approprié le module capteur de couleur RGB et on a découvert son fonctionnement :

Ce module reçoit une lumière et nous donne en sortie un signal carré dont la fréquence est proportionnelle à l’intensité lumineuse et donc en utilisant la carte Arduino on lit ce signal carré et on a le résultat en termes de couleur

Plus précisément, ce module ne peut que traiter les couleurs primaires (rouge,vert,bleu) et retourner un signal carré de chaque composante. Ainsi, + la valeur d’une des composantes est grande, + la fréquence du signal carré associé à cette composante

Dans la boucle loop de ce code, on active les photodiodes rouges, nous donne la fréquence du signal carré associé à l’intensité lumineuse de cette composante et on fait de même pour le photodiodes verts et bleues …

Ces tableaux nous disent les pins à activer en fonction de la couleur qu’on veut détecter et les deux autres pour choisir l’échelle de la fréquence de sortie…

#define S0 4

#define S1 5

#define S2 6

#define S3 7

#define sensorOut 8

int frequency = 0;

void setup() {

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

pinMode(sensorOut, INPUT);

digitalWrite(S0,HIGH);

digitalWrite(S1,LOW);

Serial.begin(9600);

}

void loop() {

digitalWrite(S2,LOW);

digitalWrite(S3,LOW);

frequency = pulseIn(sensorOut, LOW);

Serial.print("R= ");

Serial.print(frequency);

Serial.print(" ");

delay(100);

digitalWrite(S2,HIGH);

digitalWrite(S3,HIGH);

frequency = pulseIn(sensorOut, LOW);

Serial.print("G= ");

Serial.print(frequency);

Serial.print(" ");

delay(100);

digitalWrite(S2,LOW);

digitalWrite(S3,HIGH);

frequency = pulseIn(sensorOut, LOW);

Serial.print("B= ");

Serial.print(frequency);

Serial.println(" ");

delay(100);

}

Mais nous ce qui nous intéresse, c’est de pouvoir interpréter ces valeurs de couleurs et en déduire la couleur de l’objet sui passe devant le color sensor. Ainsi on va utiliser la fonction map, qui permet de à partir d’une fourchette de valeurs de départ ( cette fourchette de départ dépendra de ce que nous donne le color sensor), la recalibrer en une autre fourchette de valeurs d’arrivée( on a choisi des valeurs entre 0 et 255):

map (fréquence de sortie du color sensor, , valeur minimale source, valeur maximale source, valeur minimale arrivée, valeur maximale arrivée)

Ainsi le code devient :

#define S0 4

#define S1 5

#define S2 6

#define S3 7

#define sensorOut 8

int frequency = 0;

void setup() {

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

pinMode(sensorOut, INPUT);

digitalWrite(S0,HIGH);

digitalWrite(S1,LOW);

Serial.begin(9600);

}

void loop() {

digitalWrite(S2,LOW);

digitalWrite(S3,LOW);

frequency = pulseIn(sensorOut, LOW);

**frequency = map(frequency, 35, 160, 255, 0);**

Serial.print("R= ");

Serial.print(frequency);

Serial.print(" ");

delay(100);

digitalWrite(S2,HIGH);

digitalWrite(S3,HIGH);

frequency = pulseIn(sensorOut, LOW);

**frequency = map(frequency, 25, 200, 255, 0);**

Serial.print("G= ");

Serial.print(frequency);

Serial.print(" ");

delay(100);

digitalWrite(S2,LOW);

digitalWrite(S3,HIGH);

frequency = pulseIn(sensorOut, LOW);

**frequency = map(frequency, 35, 190, 255, 0);**

Serial.print("B= ");

Serial.print(frequency);

Serial.println(" ");

delay(100);

}

Or on a établi ces échelles avec des objets non précis au niveau de leur couleur, la prochaine séance on calibrera nos échelles de valeurs avec des bonbons colorés (ceux qu’on va utiliser pour notre projet)

Et enfin nous avons choisi un servo moteur mais nous n’avons pas travaillé dessus encore.