

SEANCE 1

COMPTE RENDU

Date : 06/12/2021

Heure : 15h – 18h

Etudiant : BOURRY Amir

En présence de

Luc SIMON

Objectifs

Débuter la partie codage de notre rover, préparer les déplacements et les modules associés.

Accomplissements

Préparation des bandes LED et des signaux sonores du Rover.

Préparation de la rotation automatisée des LEDs afin de couvrir un axe de 360°.

Préparation de la préservation des câbles pendant la rotation.

Matériel utilisé

Nous devons préserver les câbles pendant les rotations du cylindre sur lequel seront fixées les LEDs, nous avons donc utilisé une pièce sur laquelle nous branchons les câbles qui passeront dans la pièce puis ressortent de l'autre côté afin de faire tourner autour d'un axe sans abîmer les câbles. Ne pouvant utiliser de vrais rayons UV, nous avons utilisé une bande LED. Nous avons utilisé un servo moteur à rotation continue pour faire tourner le cylindre à 360°. Afin d'assurer les signaux sonores, nous avons utilisé un buzzer piézoélectrique.

Utilisation des LEDs

Nous utilisons une bande LED utilisant la bibliothèque « Adafruit Neo Pixel » qui permet de contrôler une bande LED en envoyant des informations à chaque LED séparément, nous avons donc codé pour le moment deux modes, le mode Alerte qui allume progressivement les LED en rouge l'une après l'autre avec un petit décalage afin d'alerter visuellement que le Rover se met en route. Le deuxième mode d'allumage est le mode Violet qui simule les rayons UVs.

Capture d'écran du code :

```
void loop() {
  // PARAGRAPHE ALLUMAGE DES LED EN ROUGE POUR PREVENIR DE L'ARRIVEE DES UV
  for(int i = 0; i < X; i++) {
    strip.setPixelColor(i,255,0,0);
    delay(500);
    strip.show();
  }

  // PARAGRAPHE ALLUMAGE DES LED EN VIOLET
  for(int i = 0; i < X; i++) { // On fait une boucle pour définir la couleur de chaque led
    //setPixelColor(n° de led, Rouge, Vert, Bleu)
    strip.setPixelColor(i,128,0,128);
  }
  strip.show(); // on affiche
  delay(1000);
}
```

En paramétrant comme ceci la bande LED :

```
#include <Adafruit_NeoPixel.h>

// Parameter 32 = number of pixels in strip
// Parameter 2 = pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
int X=31;

Adafruit_NeoPixel strip = Adafruit_NeoPixel(X, 9, NEO_GRB + NEO_KHZ800);
void setup() {
  strip.begin();
  strip.show(); // Initialise toute les led à 'off'
}
```

En indiquant le nombre de LED ainsi que le mode de lumière utilisé.

Nous pensons à ajouter des modes de lumière afin de rendre plus agréable le Rover pour les yeux des utilisateurs.

Utilisation du Servo Moteur

Nous utilisons un Servo Moteur à rotation continue pour faire tourner la bande LED sur elle-même au dessus du Rover, cela nous permet de faire diffuser les prétendus rayons UV sur un axe de 360°.

Le moteur tourne lentement sur lui-même de façon continue, cela est possible car le modèle est à rotation continue.

```
#include <Servo.h>

Servo myServo;
byte servoPin = 9;

void setup() {
  myServo.attach(servoPin);
  myServo.writeMicroseconds(360);
  myServo.write(94);
}
```

Nous utilisons la bibliothèque Servo.h qui permet de faire marcher le servo-moteur, les commandes writeMicroseconds et write permettent de faire tourner le servo-moteur sur lui-même à une vitesse basse.

Préservation des câbles

Afin d'éviter que les câbles qui sont reliés à la bande LED ne se tordent, nous utilisons un petit cylindre attaché à une vis pouvant tourner sur lui-même, à l'intérieur il y a des câbles avec des embouts de chaque côté, on peut brancher nos câbles à un des embouts et les faire reprendre de l'autre côté, ainsi, on peut faire tourner les câbles autour d'un axe sans les tordre.

Utilisation du Buzzer Piézoélectrique

Afin de faire une alarme permettant de prévenir de l'allumage du module, nous utilisons un simple buzzer piézoélectrique qui permet de produire du son sur une fréquence donnée. Je n'ai pas eu le temps de coder mon alarme mais celle-ci devrait être une suite de sons le moins stridents possible afin de ne pas être désagréable mais assez bruyant pour bien prévenir les utilisateurs. Le code de ce buzzer sera ajouté au compte-rendu suivant.

Conclusion

Cette séance a été la première, elle nous a imposé des problèmes que nous avons du régler avec le matériel à disposition. Nous avons réussi à les résoudre néanmoins nous n'avons pas pu finir de remplir l'objectif initial qui était de coder l'entièreté des déplacements du Rover, nous allons donc ajouter des séances de travail entre les séances programmées par l'école pour respecter notre planning le plus proche possible.

La phase de préparation de l'allumage du rover étant prête nous allons commencer le codage des déplacements dès la prochaine séance.

BOURRY Amir PeiP2 Groupe 1.

A Valbonne, le 07/12/2021.