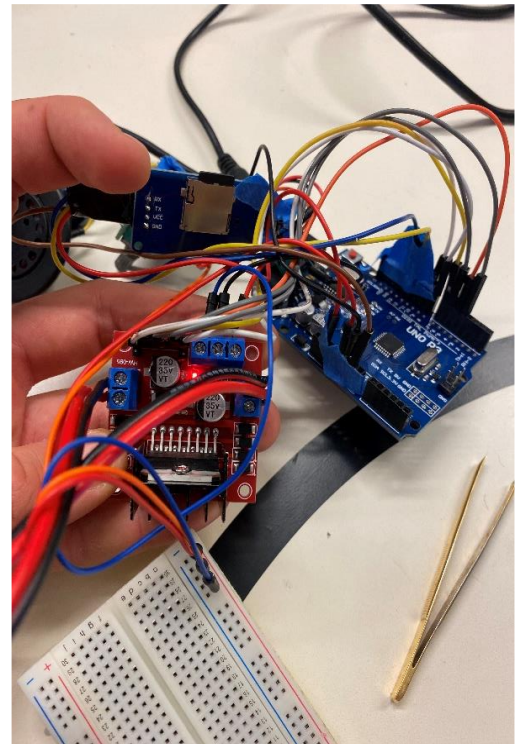
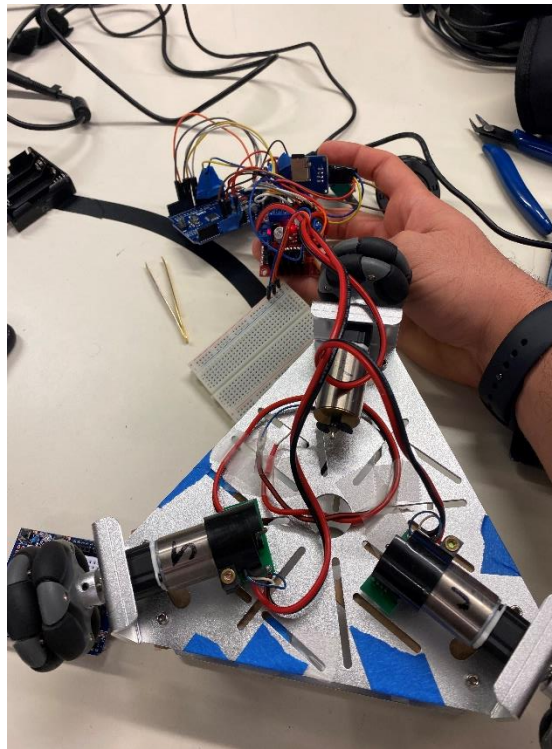
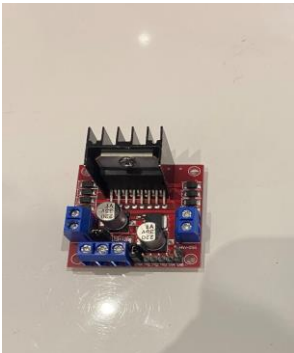


Compte Rendu séance 2 PROJET :

« Jukebox Dancer »

Durant cette deuxième séance de projet, nous avons réussi à faire ce que nous avons convenu sur le planning. Hugo s'est occupé de la barre LED avec l'encodeur rotatif (voir son compte rendu) et je me suis occupé du code sur les moteurs qui contrôlent les roues omniwheel.

Ainsi, j'ai pu faire le branchement du L298 sur 2 des 3 moteurs (il nous restera juste à brancher un deuxième L298 pour le moteur numéro 3 qui nous permettra d'obtenir une vitesse de rotation plus importante (pour les chorégraphies qui sont plutôt rapides)).



Pour les branchements, j'ai suivi les instructions du cours sur les moteurs, et j'ai pu récupérer le code fournit page33 du cours, que j'ai un peu moduler en fonction des entrées choisies.

Ainsi, j'ai créé des fonctions **avancer**, **reculer**, **rotationDroite** et **rotationGauche** qui prennent en paramètre une vitesse de rotation des roues, alors nous aurons juste à placer ces fonctions dans les méthodes des danses, avec le paramètre adéquat.

Il suffit d'indiquer un sens de rotation pour chaque moteur en début de fonction (par exemple les moteurs ne tourneront pas dans le même sens lors d'une rotation : un dans un sens, un dans l'autre, mais à la même vitesse pour pouvoir faire du sur place).

C'est ce que j'ai fait pour chaque fonction, on peut voir sur le code, à l'aide des commentaires, que j'ai indiqué le sens de rotation (direction) des moteurs (normale étant pour la marche avant et inverse pour reculer).

Voir fonctions sur la page suivante !

```

//AVANCER
void avancer(int vitesse){
|
// Direction du Moteur A normale
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);

// Direction du Moteur B normale
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);

//Moteur A - vitesse choisie
analogWrite(ENA,vitesse);

//Moteur B - vitesse choisie
analogWrite(ENB,vitesse);}

//RECULER
void reculer(int vitesse){

// Direction du Moteur A inverse
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);

// Direction du Moteur B inverse
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);

//Moteur A - vitesse choisie
analogWrite(ENA,vitesse);

//Moteur B - vitesse choisie
analogWrite(ENB,vitesse);}

//ROTATION VERS LA DROITE
void rotationDroite(int vitesse){

// Direction du Moteur A normale
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);

// Direction du Moteur B inverse
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);

//Moteur A - vitesse de rotation choisie (le robot tourne sur lui-même)
analogWrite(ENA,vitesse);

//Moteur B - vitesse de rotation choisie
analogWrite(ENB,vitesse);}

//ROTATION VERS LA GAUCHE
void rotationGauche(int vitesse){

// Direction du Moteur A inverse
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);

// Direction du Moteur B normale
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);

//Moteur A - vitesse de rotation choisie (le robot tourne sur lui-même)
analogWrite(ENA,vitesse);

//Moteur B - vitesse de rotation choisie
analogWrite(ENB,vitesse);}

```

De plus, j'ai chargé les chansons sur la carte SD, où j'ai pu récupérer le code qui permet de les jouer sur les enceintes, en sélectionnant le répertoire dans lequel le fichier se trouve (avec le nom du fichier, qui est de la forme 01,02,03).

Alors, il est facile de faire l'appel d'une chanson, avec le numéro de son répertoire et du fichier.

Voici une représentation de la carte SD : il y 3 fichiers mp3 dans chaque répertoire donc 6 musiques !

01

02

chansons_MP3_TEST

```

void setup() {
  mySerial.begin(9600); //Start our Serial coms for our serial monitor!
  delay(500); //Wait chip initialization is complete
  sendCommand(CMD_SEL_DEV, DEV_TF); //select the TF card
  delay(200);
}

void loop() {
  sendCommand(SINGLE_PLAY, 0X0001); //Premier chanson sans repertoire
  delay(10000);
  sendCommand(SINGLE_PLAY, 0X0002); //Deuxieme chanson sans repertoire
  delay(10000);
  sendCommand(CMD_PLAY_WITHFOLDER, 0X0101); //Premier chanson repertoire no 1
  delay(10000);
  sendCommand(CMD_PLAY_WITHFOLDER, 0X0201); //Premier chanson repertoire no 2
  delay(10000);
}

```

