

Rapport de la seconde séance

Soens Arthur

Avec Emma nous nous sommes réparti les tâches, elle s'est occupée de la partie matérielle, tandis que moi je me suis occupé de comprendre et faire fonctionner les moteurs

Compréhension du moteur :

D'après le cours, le moteur Nema17 fonctionne en faisant un pas à chaque fois qu'un signal "HIGH" est détecté, j'ai donc pris pour base ce qui était dit, c'est à dire, 1.8° par pas. En divisant 360° par 1.8° on trouve qu'il faut qu'un moteur détecte 200 fois un signal "HIGH" pour qu'il fasse un tour complet.

Synchronisation de la vitesse :

Pour synchroniser la vitesse j'ai juste fait changer les moteurs d'états au même moment, comme ils ont le même nombre de pas, la vitesse est nécessairement identique

Grace a ces réglages je suis en mesure de faire tourner les moteurs autant de tour que je le souhaite à la même vitesse, ce qui me permet, une fois les dimensions obtenues, de faire tourner précisément le moteur pour chaque case du jeu.

Maintenant j'ai essayé, à la fin de la séance, de prendre en compte les paramètres d'un utilisateur, mais le manque de temps, m'a obligé à repousser ce passage à la prochaine séance.

Exemple des différentes fonctions créées lors de cette séance :

//fonction permettant de faire un tour complet à la même vitesse avec les moteurs

```
void tourner_sur_1_tour(){  
    for(int x=0;x<200;x++){  
        digitalWrite(stepPin2,HIGH);  
        digitalWrite(stepPin,HIGH);  
        delay(10);  
        digitalWrite(stepPin2,LOW);  
        digitalWrite(stepPin,LOW);  
    }
```

```

        delay(10);
    }
}

//inutile mais on garde pour le moment
void tourner(){
    analogWrite(stepPin,40);
    analogWrite(stepPin2,40);
    delay(500);
}

void stopper(){
    analogWrite(stepPin,0);
    analogWrite(stepPin2,0);
}

```

Et maintenant un exemple de la structure du code (sans les déclarations de variable) :

```

void setup() {
    //Serial.begin(115200);
    Serial.begin(9600);
    //Réglage des pins en Sortie
    pinMode(stepPin,OUTPUT);
    pinMode(dirPin,OUTPUT);
    pinMode(stepPin2,OUTPUT);
    pinMode(dirPin2,OUTPUT);
    digitalWrite(dirPin,HIGH);
    digitalWrite(dirPin2,HIGH);// Enables the motor to move in a particular direction
    //Réglage des pins pour ralentir les moteurs
    pinMode(ms1_2,OUTPUT);
    digitalWrite(ms1_2,LOW);
    pinMode(ms2_2,OUTPUT);
}

```

```

digitalWrite(ms2_2,LOW);

pinMode(ms1_1,OUTPUT);

digitalWrite(ms1_1,LOW);

pinMode(ms2_1,OUTPUT);

digitalWrite(ms2_1,LOW);

////////////////////////////////////

//pas fini

Serial.println("Entrez un nombre de tour : ");

while(Serial.available()){

    if(Serial.read()!="0"){

        chiffre += Serial.read();

    }

}

if(chiffre != ""){

    Serial.print(chiffre);

}

}

////////////////////////////////////

void loop() {

    tourner_sur_1_tour();

}

```

Pour la prochaine séance :

Il s'agira de réussir à prendre en compte l'utilisateur, c'est à dire de détecter le nombre de case que sur lesquelles nous devons déplacer la pince, puis la remettre à sa position initiale. Ensuite adapter ce code pour un deuxième rail superposé au premier afin de pouvoir se déplacer au-dessus de toute les case. (On utilisera surement l'équivalent d'un

enum en java pour associer le nombre de pas à faire , au chiffage de la case)