

15/02/2023

Compte rendu séance n°7

Au cours de cette séance nous avons :

D'une part, à nouveau travaillé sur le système d'alimentation. Très fragile il avait souvent tendance à se détacher. J'ai donc dû renforcer les soudures de part et d'autre.

Quoiqu'il en soit, Nicolas a déjà imprimé la structure de la boîte qui accueillera le dispositif. Une fois celle-ci terminée et le transcripteur fixé dedans, les branchements du système d'alimentation seront beaucoup moins sollicités.

D'autre part, retravaillé entièrement le code Arduino du projet. Nous avons en effet décidé de nous passer du code écrit par les étudiants précédents après la découverte d'une bibliothèque développée pour le contrôle des 74HC595, et qui nous permet un contrôle bien plus fin des solénoïdes, réglant énormément de difficultés de traduction et simplifiant terriblement les processus.

J'ai donc mis au point une toute nouvelle version du code implémentant cette bibliothèque et qui, déjà permet une initialisation quasi-impeccable (j'y reviendrai) et des résultats très convaincants concernant à la transcription. En l'état, outre l'implémentation dans le code de la communication via Bluetooth, de l'écran LCD etc, il faudra régler le fait qu'après l'envoi d'un mot, il faut, pour le moment, retéléverser le programme avant de pouvoir envoyer un nouveau mot...

Malgré tout, le progrès reste majeur : nous avons résolu le problème qui nous a pris la quasi-totalité de notre temps jusqu'à présent et sommes en capacité d'afficher toutes les lettres de l'alphabet où nous le souhaitons sur le dispositif...

Seule ombre au tableau, ce nouveau code a mis en lumière une nouvelle difficulté. Il semblerait qu'au moins un solénoïde ne répondait pas correctement. En effet, grâce à la nouvelle bibliothèque, j'ai pu tester

individuellement chacun des solénoïdes et le 16^{ème} ne répond à l'instruction « se lever ». Pour autant, on a vu qu'il était capable de bouger, nous ne sommes juste pas capables de le contrôler. Je pense que le problème se trouve au niveau du solénoïde ou du 74HC595 qui contrôle ledit solénoïde ou, éventuellement, dans le câblage entre ces deux. Il faudra donc identifier l'origine du problème et le nombre de solénoïdes concernés.

Malgré tout nous pouvons désormais utiliser les solénoïdes suivant la 16^{ème} et donc obtenir des résultats convaincants.

Notre code utilisant la nouvelle bibliothèque :

```
Braille_LearnerV1
#include <ShiftRegister74HC595.h>

// create a global shift register object
// parameters: <number of shift registers> (data pin, clock pin, latch pin)
ShiftRegister74HC595<8> sr(10, 12, 11);

String mot = "";

String lettre ( String car, int k ){

    if (car=="a" or "A"){
        sr.set(0+6*k,HIGH);
        Serial.println("A");
    }

    else if (car=="b" or "B"){
        sr.set(0+6*k,HIGH);
        sr.set(1+6*k,HIGH);
        Serial.println("B");
    }

    else if (car=="c" or "C"){
        sr.set(0+6*k,HIGH);
        sr.set(4+6*k,HIGH);
        Serial.println("C");
    }

    else if (car=="d" or "D"){
        sr.set(0+6*k,HIGH);
        sr.set(4+6*k,HIGH);
        sr.set(5+6*k,HIGH);
        Serial.println("D");
    }
}
```

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
  
  for (int i=0; i<16; i++){  
    sr.set(i,HIGH);  
  }  
  Serial.println("Init 1 OK");  
  
  for (int i=17; i<59; i++){  
    sr.set(i,HIGH);  
  }  
  Serial.println("Init 2 OK");  
  
  delay(500);  
  
  for (int i=0; i<16; i++){  
    sr.set(i,LOW);  
  }  
  Serial.println("Init 3 OK");  
  
  for (int i=17; i<59; i++){  
    sr.set(i,LOW);  
  }  
  Serial.println("Init 4 OK");  
  
  delay(10);  
}
```

```

void loop() {
  // put your main code here, to run repeatedly:
  //Serial.println("On est dans la boucle!");
  delay(100); // Attendre 100ms

  //Serial.println("Prêt à écrire, donnez mot.");

  if (Serial.available() > 0) {
    String mot = Serial.readString(); // Lire une chaîne de caractères jusqu'à rencontrer un espace

    if (mot.length() > 0) {
      Serial.println(mot);
      for (int j=0; j<mot.length()-1;j++){
        lettre(String(mot.charAt(j)),j);
        Serial.println(lettre(String(mot.charAt(j)),j));
        delay(1000);
      }

      for (int i=0; i<16; i++){
        sr.set(i,LOW);
      }
      Serial.println("Loop reset 1 OK");

      for (int i=17; i<59; i++){
        sr.set(i,LOW);
      }
      Serial.println("Loop reset 2 OK");
    }
    Serial.flush();
  }
}

```