

HOURI ELINE G1

Projet : BLINDTOUCH

Séance 1 du 09/12/2019 :

Tâches effectuées :

Pour la première séance, nous avons récupéré une partie du matériel :

-36 solénoïdes

-6 mémoires : SN74HC595N

-1 ULN2803

-une alimentation de 12

Ne sachant pas comment le registre à décalage fonctionne, nous avons réalisé un test avec non pas les solénoïdes qui nécessitent une alimentation de 12V mais avec des LED.

Le registre à décalage permet, de manière générale, à connecter beaucoup de LED(ou solénoïdes) à un Arduino sans utiliser trop de sorties. Ce dernier nous sera très utile car il nous permettra de tout relier sans utiliser plusieurs cartes Arduino.

Pour cela, ce circuit possède une entrée série et plusieurs sorties parallèles .Le principe est d'envoyer un 0 ou 1 à chaque coup d'horloge, sur l'entrée, pour décaler les données.

Pour le test nous avons utilisé un code trouvé sur internet. Malheureusement, le code n'a pas fonctionné et nous n'avons pas eu le temps de voir de plus près le problème

Problèmes rencontrés :

Le code n'a pas fonctionné ce qui nous a empêché de mieux comprendre et de visualiser le fonctionnement du registre à décalage.

Remarques :

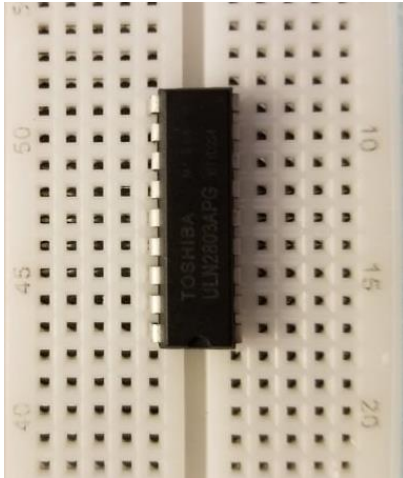
Mr Masson nous a fourni le code d'un ancien projet (Xylophone) pour pouvoir faire le test.

Objectif pour la séance prochaine :

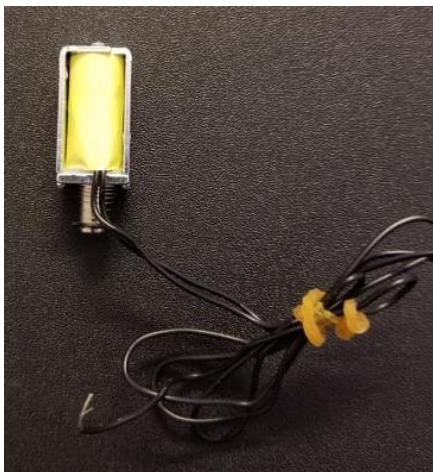
Il faudra refaire le test rapidement pour pouvoir, cette fois-ci, utiliser le code et le montage avec les solénoïdes et utiliser l'ULN2803.

ANNEXE :

-ULN2803



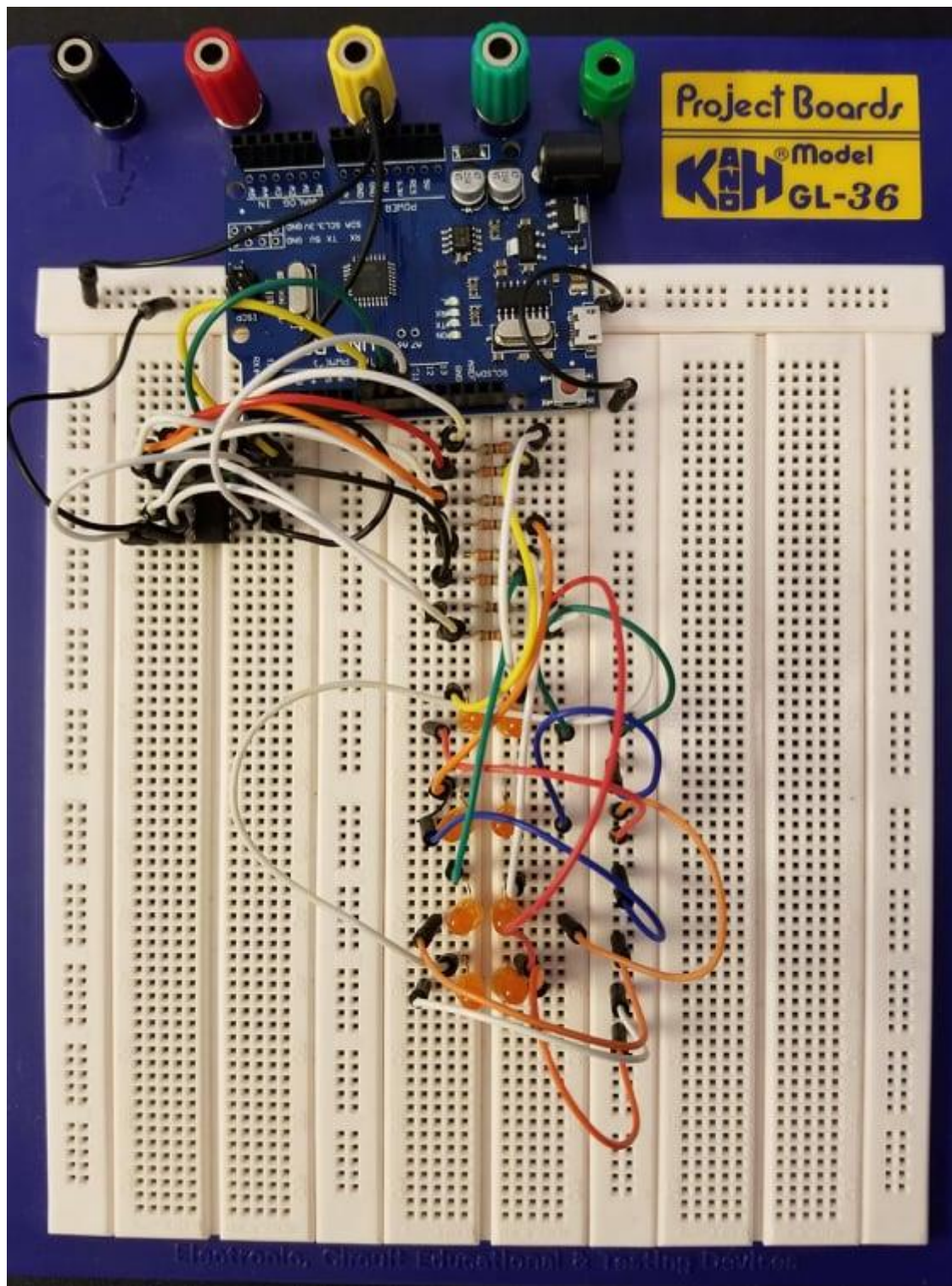
-Solénoïde :



-SN74HC595N



Schéma du montage du test :



Code test (échec):

```
// Made by https://retroetgeek.com
#define PIN_DS 8 //pin 14 75HC595
#define PIN_STCP 9 //pin 12 75HC595
#define PIN_SHCP 10 //pin 11 75HC595

//Nombre de registre a decalage utilise
//How many shift register
#define numberOf74hc595 1

// nombre total de pin de registre a decalage
// number of total register pin
#define numOfRegisterPins numberOf74hc595 * 8

// cree un array contenant des booleen de la taille du nombre de pin 74hc595
// Create an array with boolean, size of pin of 74hc595
boolean registers[numOfRegisterPins];

void setup(){
  pinMode(PIN_DS, OUTPUT);
  pinMode(PIN_STCP, OUTPUT);
  pinMode(PIN_SHCP, OUTPUT);

  // remise a 0 des registres
  //reset all register pins
  clearRegisters();
  // on applique les valeurs au registre a decalage
  // write value on shift register
  writeRegisters();
  Serial.begin(9600);
  Serial.println("OK");
}

// Mettre toutes les valeurs a 0 pour le registre
//set all register pins to LOW
void clearRegisters(){
  for(int i = numOfRegisterPins - 1; i >= 0; i--){
    registers[i] = LOW;
  }
}
```

```
// affectation des valeurs enregistrées dans le tableau "registers" et application (visualisation) des valeurs à la fin
```

```
// set value recorded in array "registers" and display on the end
```

```
void writeRegisters(){
```

```
    // Tant que LOW les modifications ne seront pas affectées
```

```
    // Until LOW modification will not be apply
```

```
    digitalWrite(PIN_STCP, LOW);
```

```
    // boucle pour affecter chaque pin des 74hc595
```

```
    // loop for apply all value for each pin 74hc595
```

```
    for(int i = numOfRegisterPins - 1; i >= 0; i--){
```

```
        //doit être à l'état bas pour changer de colonne plus tard
```

```
        //need to be low for change column soon
```

```
        digitalWrite(PIN_SHCP, LOW);
```

```
        // récupération de la valeur dans le tableau registers
```

```
        // catch value inside array registers
```

```
        int val = registers[i];
```

```
        //affecte la valeur sur le pin DS correspondant à un pin du 74hc595
```

```
        //apply the value to a pin of 74hc595
```

```
        digitalWrite(PIN_DS, val);
```

```
        //colonne suivante
```

```
        // next column
```

```
        digitalWrite(PIN_SHCP, HIGH);
```

```
    }
```

```
    //applique toutes les valeurs au 74hc595
```

```
    // apply value to all pin of 74hc595
```

```
    digitalWrite(PIN_STCP, HIGH);
```

```
}
```

```
// enregistre une valeur pour un registre état haut ou bas
```

```
//set an individual pin HIGH or LOW
```

```
void setRegisterPin(int index, int value){
```

```
    registers[index] = value;
```

```
}
```

```
void loop(){
```

```
Serial.print("boucle ");
setRegisterPin(0, HIGH);
setRegisterPin(1, HIGH);
setRegisterPin(2, HIGH);
setRegisterPin(3, HIGH);
setRegisterPin(4, HIGH);
setRegisterPin(5, HIGH);
setRegisterPin(6, HIGH);
setRegisterPin(7, HIGH);
// appelle la fonction pour appliquer les valeurs
// call function to apply value
writeRegisters();
```

```
delay(500);
Serial.println("boucle 1");
setRegisterPin(0, HIGH);
setRegisterPin(1, HIGH);
setRegisterPin(2, HIGH);
setRegisterPin(3, LOW);
setRegisterPin(4, LOW);
setRegisterPin(5, HIGH);
setRegisterPin(6, HIGH);
setRegisterPin(7, HIGH);
writeRegisters();
```

```
delay(500);
Serial.println("boucle 2");
setRegisterPin(0, HIGH);
setRegisterPin(1, HIGH);
setRegisterPin(2, LOW);
setRegisterPin(3, LOW);
setRegisterPin(4, LOW);
setRegisterPin(5, LOW);
setRegisterPin(6, HIGH);
setRegisterPin(7, HIGH);
writeRegisters();
```

```
delay(500);
Serial.println("boucle 3");

setRegisterPin(0, HIGH);
```

```
setRegisterPin(1, LOW);  
setRegisterPin(2, LOW);  
setRegisterPin(3, LOW);  
setRegisterPin(4, LOW);  
setRegisterPin(5, LOW);  
setRegisterPin(6, LOW);  
setRegisterPin(7, HIGH);  
writeRegisters();
```

```
delay(500);  
Serial.println("boucle 4");  
setRegisterPin(1, HIGH);  
setRegisterPin(6, HIGH);  
writeRegisters();
```

```
delay(200);  
Serial.println("boucle 5");  
setRegisterPin(2, HIGH);  
setRegisterPin(5, HIGH);  
writeRegisters();
```

```
delay(200);
```

```
}
```