

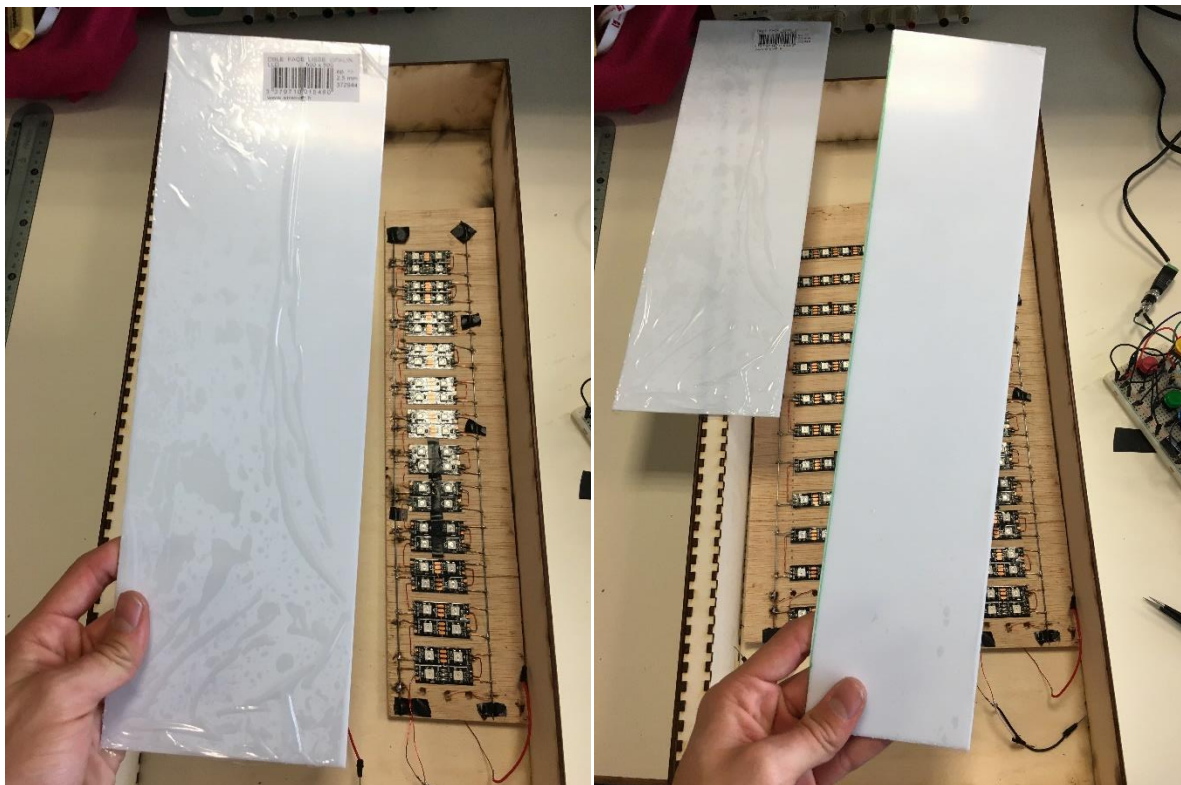
Objectif de la séance :

- Couper les planches de plexiglass pour les poser au-dessus des LEDS
- Couper les planches de bois des LEDS
- Gérer la musique du jeu (à continuer dans la semaine)
- Fixer quelques paramètres du jeu pour l'améliorer (dont la génération aléatoire de la solution)

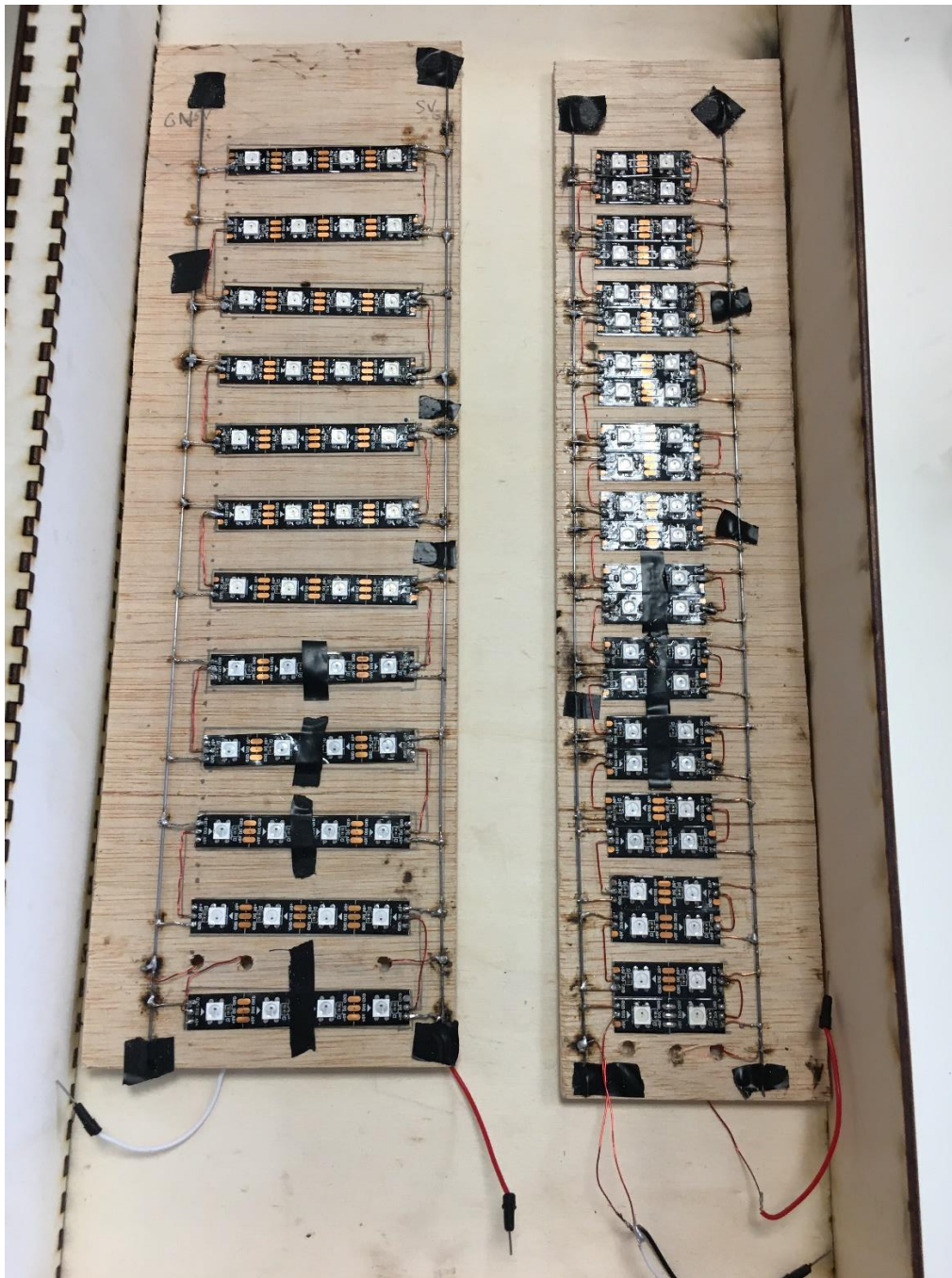
Vendredi 14/02 nous sommes allés au FabLab pour commencer notre boîte ! Je vous laisse regarder le rapport de Manon pour le détail des opérations effectuées. Prochain rdv le 21/02 pour découper les dernières boîtes manquantes.

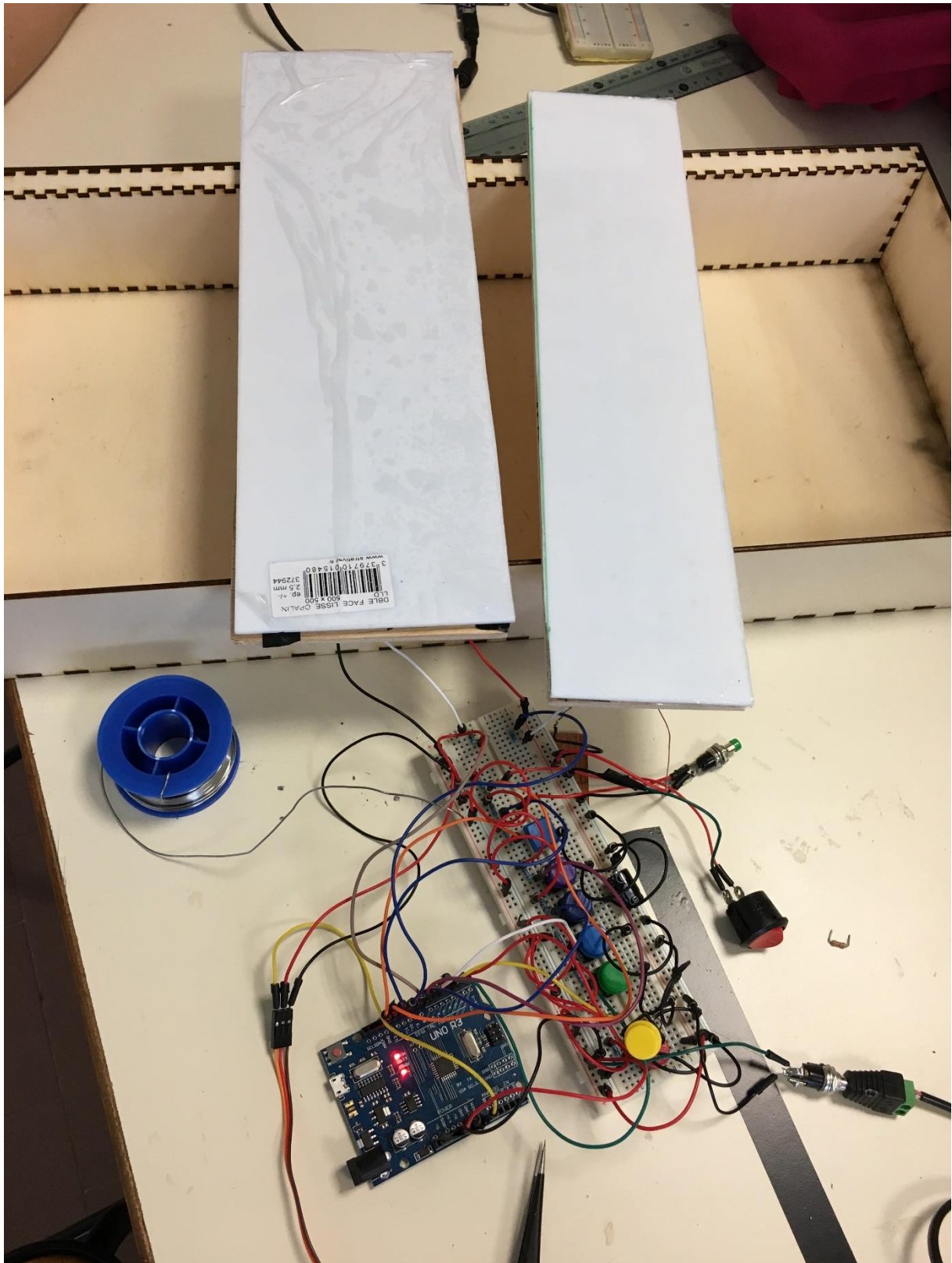
Voici quelques rendus de la découpe des plaques de plexi :

Nous avons besoin d'une planche en plexi de taille 35x11 et une autre de taille 35x8.



Et **des planches à LEDS**, trouées afin de faire passer les fils en dessous.





Petite photo du jeu branché !

Idées des musiques à mettre :

WIN

++Default : <https://www.youtube.com/watch?v=Kp4Um4AS-5U>

++Final fantasy : <https://www.youtube.com/watch?v=-YCN-a0NsNk&list=PLw9AcE2IomDNjCTqiXGA7COU0WB14LYO&index=2>

++Mario Kart :

<https://www.youtube.com/watch?v=Zcp0WHWhYVM&list=PLw9AcE2IomDNjCTqiXGA7COU0WB14LYO&index=14>

++Bomber Man : <https://www.youtube.com/watch?v=ggHecbCJ-sw&list=PLw9AcE2IomDNjCTqiXGA7COU0WB14LYO&index=18>

--Pokemon : https://www.youtube.com/watch?v=e7zMbX_0e5o

--Premier coup : <https://www.youtube.com/watch?v=WPzMxiGd0io>

LOOSE

--Mario :

https://www.youtube.com/watch?v=0WVyszWsPjg&list=PLIzvv_yKP_jxgjqU6lqRCb7cbLCVt9c9S&index=7

++Pouin pouin puin : <https://www.youtube.com/watch?v=jAlIKqL3nHo>

--Mr chicken parodie : <https://www.youtube.com/watch?v=uDrdZM1iGrc>

--GTA :

<https://www.youtube.com/watch?v=OleLlIpymtU&list=PLppsEWZrWpltCEJBeuobh1o496ssXmd>

++Mario dead :

<https://www.youtube.com/watch?v=hom9faSBUHQ&list=PLppsEWZrWpltCEJBeuobh1o496ssXmd&index=3>

--Whatcha say :

<https://www.youtube.com/watch?v=KZm6UxrFdig&list=PLppsEWZrWpltCEJBeuobh1o496ssXmd&index=23>

++End credits :

<https://www.youtube.com/watch?v=Lm4LYX3xdkU&list=PLppsEWZrWpltCEJBeuobh1o496ssXmd&index=24>

++Hello darkness my old friend :

<https://www.youtube.com/watch?v=qYS0EeaAUMw&list=PLppsEWZrWpltCEJBeuobh1o496ssXmd&index=25>

++For the damaged coda (à edit) : <https://www.youtube.com/watch?v=4Js-XbNj6Tk&list=PLppsEWZrWpltCEJBeuobhix1o496ssXmd&index=30>

Programme test :

J'ai effectué plusieurs tests pour comprendre le fonctionnement, et je pense avoir compris. J'ai fait un programme test

```
chansons_MP3 | Arduino 1.8.12 (Windows Store 1.8.33.0)
Fichier Edition Croquis Outils Aide

chansons_MP3
Send_buf[7] = 0xef; //ending byte
for(uint8_t i=0; i<8; i++){
  mySerial.write(Send_buf[i]);
}

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);
  pinMode(2, OUTPUT);
  digitalWrite(2, HIGH);
}

void loop() {
  int i=0;
  while (i<5){
    sendCommand(CMD_PLAY_WITHFOLDER, musicLoose[i]);
    delay(10000);
    i+=1; //play the first song with volume 15 class
  }
  i=0;
  while (i<4){
    sendCommand(CMD_PLAY_WITHFOLDER, musicWin[i]); //play the first song with volume 15 class
    delay(10000);
    i+=1;
  }
}

Téléversement terminé

Le croquis utilise 2640 octets (8%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 148 octets (7%) de mémoire dynamique, ce qui laisse 1900 octets pour les variables locales. Le maximum est de 2048 octets.

41 Armel atmega328p xplained mini sur COM5
```

```
chansons_MP3 | Arduino 1.8.12 (Windows Store 1.8.33.0)
Fichier Edition Croquis Outils Aide

chansons_MP3
#define SET_DAC 0x17
#define CMD_PLAY_WITHVOLUME 0x22 //data is needed 0x7E 06 22 00 xx yy EF:(xx volume)(yy number of song)
#define CMD_SEL_DEV 0x09 //SELECT STORAGE DEVICE, DATA IS REQUIRED
#define DEV_TF 0x02 //HELLO, IN THE DATA REQUIRED
#define SLEEP_MODE_START 0x0A
#define SLEEP_MODE_WAKEUP 0x0B
#define CMD_RESET 0x0C //CHIP RESET
#define CMD_PLAY 0x0D //RESUME PLAYBACK
#define CMD_PAUSE 0x0E //PLAYBACK IS PAUSED
#define CMD_PLAY_WITHFOLDER 0x0F //DATA IS NEEDED, 0x7E 06 0F 00 01 02 EF:(play the song with the directory \01\002xxxxxx.mp3
#define STOP_PLAY 0x16
#define PLAY_FOLDER 0x17 // data is needed 0x7E 06 17 00 01 XX EF:(play the 01 folder)(value xx we dont care)
#define SET_CYCLEPLAY 0x19 //data is needed 00 start: 01 close
#define SET_DAC 0x17 //data is needed 00 start: DAC OUTPUT:01 DAC no output

int16_t musicWin[] = {0x0F00705, 0x0F00706, 0x0F00707};
int16_t musicLoose[] = {0x0F00701, 0x0F00702, 0x0F00703, 0x0F00704};

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

void sendCommand(int8_t command, int16_t dat) {
  delay(20);
  Send_buf[0] = 0x7e; //starting byte
  Send_buf[1] = 0xff; //version
  Send_buf[2] = 0x06; //the number of bytes of the command without starting byte and ending byte
  Send_buf[3] = command; //
  Send_buf[4] = 0x00; //0x00 = no feedback, 0x01 = feedback
  Send_buf[5] = (int8_t)(dat >> 8); //data
}

Téléversement terminé

Le croquis utilise 2640 octets (8%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 148 octets (7%) de mémoire dynamique, ce qui laisse 1900 octets pour les variables locales. Le maximum est de 2048 octets.

46 Armel atmega328p xplained mini sur COM5
```

RAPPORT SÉANCE 17 FEVRIER 2020

J'ai créé deux listes, comportant les musiques de victoire et défaite ! Je pourrai donc choisir aléatoirement parmi ces musiques à la fin du jeu. On pourra également (peut-être) ajouter la possibilité de choisir la musique voulu avec le téléphone en bluetooth.

Génération de la solution aléatoire :

Ce qui peut être fait en plus de choisir une solution aléatoire, c'est lire les variations sur une entrée analogique, et de réutiliser ces valeurs pour générer une solution aléatoire.

Je n'ai pas réussi à utiliser ce système j'aurai besoin de vous la semaine prochaine pour m'aider. Je continuerai dans la semaine.