RAPPORT DE SEANCE 6

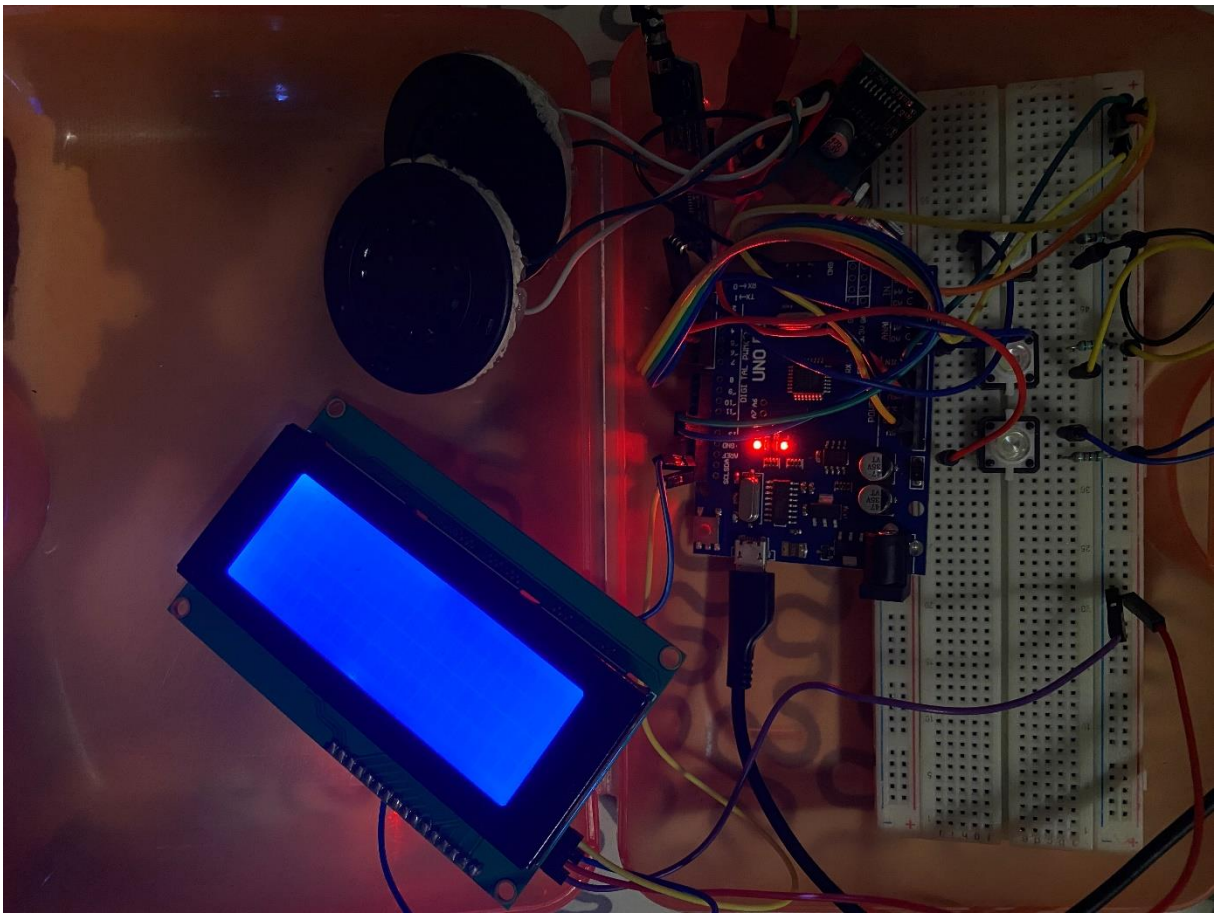
Objectif de la séance : Cette séance devait servir à finaliser le programme en liant le programme de l'écran LCD avec le programme du menu déroulant puis y ajouter le programme des LEDS et finaliser le Bluetooth pour pouvoir nous concentrer sur la partie physique pour les deux séances restantes.

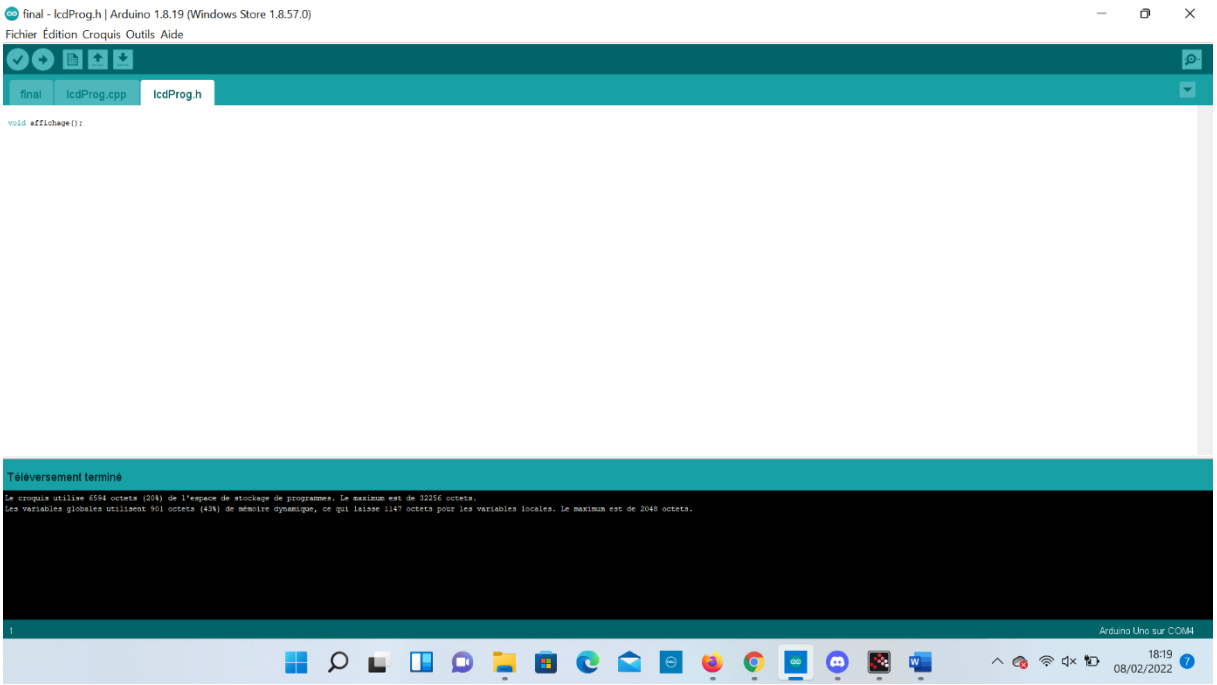
Ecran LCD + menu déroulant :

Pour lier le programme LCD avec le menu déroulant, il a fallu organiser le code en trois fichier : un fichier final.ino avec le programme du menu déroulant, un fichier lcdProg.h qui contient le prototype de la fonction affichage et un fichier lcdProg.cpp qui contient l'implémentation de la fonction.

https://eskimon.fr/tuto-arduino-905-organisez-votre-code-en-fichiers#le-fichier-h

Montage :




final - lcdProg.h | Arduino 1.8.19 (Windows Store 1.8.57.0)
Fichier Édition Croquis Outils Aide
final
lcdProg.cpp
lcdProg.h
void affichage();
Téléversement terminé
Le croquis utilise 6594 octets (20%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 901 octets (43%) de mémoire dynamique, ce qui laisse 1147 octets pour les variables locales. Le maximum est de 2048 octets.
1
Arduino Uno sur COM4
18:19
08/02/2022

Problèmes rencontrés :

Tout d'abord, la compilation et le téléversement du code des trois fichiers se font sans erreur mais le programme ne fonctionne pas car l'écran lcd n'affiche rien et il n'y a même plus de son qui s'enclenche. J'en conclu que le problème vient du fait que je n'ai pas bien découpé les parties de fichiers et peut être qu'au lieu de faire une fonction affichage, il y a d'autres étapes. De plus, la variable « i » est utilisée dans le programme final et lcdProg.cpp. Or, on ne peut pas créer deux fois la même variable dans 2 dossiers différents et on ne peut pas non plus ne déclarer que cette variable dans un seul des fichier. J'ai donc pensé à utiliser une autre variable « j » dans le fichier .cpp puis transformer ce « j » en « i » dans le programme final. Cependant, cela ne fonctionne toujours pas mais je règlerais ce problème avant la prochaine séance pour pouvoir finaliser le code et pouvoir se concentrer sur l'aspect physique de notre Jukebox.

////////////////////////////// CODE //

Final.ino :

```
#include <SoftwareSerial.h>

int i=1;


#include "lcdProg.h"

#define RX 12

#define TX 13


#define BOUTONG 3

#define BOUTONOK 4

#define BOUTOND 5


int val1=0;

int val2=0;

int val3=0;


//const zero= 0X00000;


SoftwareSerial mySerial(RX,TX);
```

```
////////////////////// L C D ////////////////////////////////////////////////////////

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

//all the commands needed in the datasheet(http://geekmatic.in.ua/pdf/Catalex_MP3_board.pdf)

static int8_t Send_buf[8] = {0} ;//The MP3 player undestands orders in a 8 int string

                    //0X7E FF 06 command 00 00 00 EF;(if command =01 next song order)
#define NEXT_SONG 0X01

#define PREV_SONG 0X02

#define CMD_PLAY_W_INDEX 0X03 //DATA IS REQUIRED (number of song)

#define VOLUME_UP_ONE 0X04

#define VOLUME_DOWN_ONE 0X05

#define CMD_SET_VOLUME 0X06//DATA IS REQUIRED (number of volume from 0 up to 30(0x1E))

#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,IM 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
```

```
#define SINGLE_PLAY 0X08//Single play(without folder)

////////////////////////////////////////////////////////////////////////////////
void setup()
{
  pinMode(BOUTOND,INPUT);
  pinMode(BOUTONG,INPUT);
  //pinMode(BOUTONOK,INPUT);
  Serial.begin(9600);
  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(BOUTOND,INPUT); //définit le bouton précedent, en entrée numérique sur la broche 1
  pinMode(BOUTONG,INPUT); //définit le bouton selectionner, en entrée numérique sur la broche 2
  pinMode(BOUTONOK,INPUT); //définit le bouton suivant, en entrée numérique sur la broche 3
  //updateMenu();
}

void sendCommand(int8_t command, int8_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] = 0x00;//datah
  Send_buf[6] = dat; //datal
  Send_buf[7] = 0xef; //ending byte
  for(uint8_t j=0; j<8; j++){
    mySerial.write(Send_buf[j]) ;
```

```
  }
}



void loop() {
  if(digitalRead(BOUTONG) ){
     i--;
     affichage();
     //updateMenu();
     delay(100);
     while(digitalRead(BOUTONG));
    }
    if(digitalRead(BOUTOND)){
     i++;
     affichage();
     Serial.println("D");
     //updateMenu();
     delay(100);
     while(digitalRead(BOUTOND));
    }
    if(digitalRead(BOUTONOK)){
     //updateMenu(i);
     Serial.println(i);
     sendCommand(SINGLE_PLAY, i);
     delay(100);
     while(digitalRead(BOUTONOK));
    }

}
```

lcdProg.cpp :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>//----- Adressage matériel -----
// En cas de non fonctionnement, mettez la ligne 8 en
// commentaire et retirez le commentaire à la ligne 9.
LiquidCrystal_I2C lcd(0x27, 20, 4);
//LiquidCrystal_I2C lcd(0x3F,20,4);


void affichage(){
 lcd.init();
 lcd.backlight();
 // Envoi du message
 if(i==1){
  lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
  lcd.print("Ain't no Mountain ");
  lcd.setCursor(0, 1);
  lcd.print("High Enough");
  lcd.setCursor(0,3);
  lcd.print("Marvin Gaye");
 }
 if(i==2){
  lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
  lcd.print("ABC");
  lcd.setCursor(0,2);
  lcd.print("Jackson Five");

 }
 if(i==3){
   lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
```

```
  lcd.print("Don't stop the music");
  lcd.setCursor(0,2);
  lcd.print("Rihanna");
}
if(i==4){
  lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
  lcd.print("Wesh Alors");
  lcd.setCursor(0,2);
  lcd.print("JUL");
}
if(i==5){
 lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
 lcd.print("Earth, Wind & Fire");
 lcd.setCursor(0,2);
 lcd.print("September");
}
if(i==6){
 lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
 lcd.print("Get down on it");
 lcd.setCursor(0,2);
 lcd.print("Kool & The Gang");
}

if(i==7){
lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
lcd.print("Hit the road Jack");
lcd.setCursor(0,2);
lcd.print("Ray Charles");
```

```
}

if(i==8){
 lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
 lcd.print("Lady");
 lcd.setCursor(0,2);
 lcd.print("Modjo");
}

if(i==9){
 lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
 lcd.print("Beat it");
 lcd.setCursor(0,2);
 lcd.print("Michael Jackson");
}
if(i==10){
 lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
 lcd.print("Beat it");
 lcd.setCursor(0,2);
 lcd.print("Michael Jackson");
}
if(i==11){
 lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
 lcd.print("Never gonna give");
 lcd.setCursor(0, 1);
 lcd.print("you up");
 lcd.setCursor(0,3);
 lcd.print("Rick Astley");
}
```

```
 if(i==12){
  lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
  lcd.print("On va s'aimer");
  lcd.setCursor(0,2);
  lcd.print("Gilbert Montagné");
 }
 if(i==13){
  lcd.setCursor(0, 0); // positionne le curseur à la colonne 1 et à la ligne 1
  lcd.print("Young Wild & Free");
  lcd.setCursor(0,2);
  lcd.print("Wiz Khalifa");
  lcd.print("Snoop Dog");
 }

}
```

lcdProg.h :

```
void affichage();
```