BIBLIOGRAPHIE PROJET - PeiP2

Année scolaire 2020-2021

LEDRIS

Etudiants : Belloni Maxime
Rebaudi Elodie

Encadrant : Pascal Masson

SOMMAIRE

Introduction :

Le projet que nous avons décidé de faire est un Tetris créé avec des LEDs et des boutons directionnels, sans oublier la musique très connue du fameux Tetris. Nous avons décidé d'appeler ce projet le LEDris. Ce document a pour but de montrer toutes nos recherches et notre travail de renseignement et de réflexion pour essayer de voir comment est-ce que l'on pense notre projet et quels composants seront nécessaires.

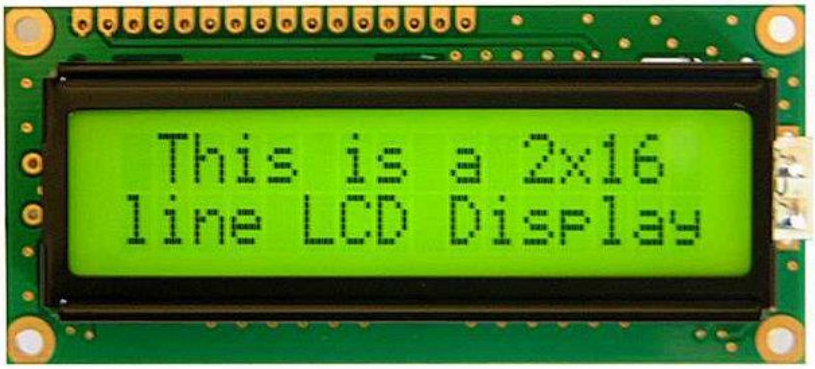
Composants :

1. Carte arduino MEGA

La **carte Arduino MEGA** est intéressante car une grande mémoire est nécessaire et la carte Arduino Nano n'en possède pas assez. En contrepartie, elle est plus imposante à stocker dans notre boîte finale car elle mesure 107 x 53 x 15mm.
Elle possède 54 Input/Output (voir annexe), ce qui permet une grande capacité de câblage, et sa mémoire est de 256kB, ce qui est 8 fois plus élevé que la Arduino Nano.

2. Ecran LCD

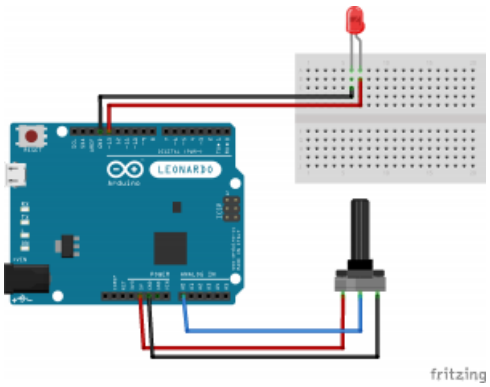
Un **écran LCD**, comme fourni en cours, nous permettrait d'afficher la personne qui possède le record (pseudonyme 6 caractères + score) et la personne qui joue actuellement.
Ce composant utilise la librairie « liquidCrystal » qui nous permet d'initialiser et d'écrire en temps réel sur l'écran en modifiant chaque caractère comme on le veut.



3. La Matrice de LED

Une **bande de LEDs** est nécessaire pour créer la matrice du jeu, le type Strip NeoPixel RGB de 60 LEDs par mètre nous permet d'avoir un espacement entre les LEDs de 1,6cm ce qui nous permet d'avoir un jeu de 10 cubes sur 20 cubes (200 LEDs requise donc 4m de bande) et donc un écran de 16cm par 32cm. Cependant nous auront comme obligation de créer des bordures pour notre cadrillage plus fines.
Ce type de bande nécessite seulement 1 I/O de la carte arduino pour faire fonctionner toute la ligne de LEDs. Il faut la relier au +5V et à la masse puis la mettre en série avec une résistance.
La librairie AdaFruit-neopixel permet de programmer l'allumage des LEDs.

Cela nous gênerait si la luminosité était trop élevée, nous voudrions donc trouver une manière d'augmenter ou diminuer la luminosité de l'éclairage de notre tableau de LED. Nous avons tout de suite pensé à la méthode du PWM pour diminuer la fréquence d'éclairage à l'aide d'un **potentiomètre**, ce qui nous permettrait de changer la luminosité de chaque LED.



4. Les boutons

On recherche des **boutons** de tailles assez importante afin de pouvoir appuyer dessus sans difficultés. Les boutons poussoirs utilisés en cours sont trop petits. Nous recherchons des boutons d'au moins 6mm de diamètre. Ces boutons nous serviront à déplacer les pièces à droite, à gauche, vers le bas et les rotations horaires et antihoraires de la pièce.

Ce bouton-ci fait 3 x 2.5cm, donc le bouton en lui-même est assez imposant sûrement plus de 1.5cm de diamètre. Il pourrait convenir mais semble un petit peu trop gros.

Ces autres boutons eux font 12 x 12 x 7.3mm donc le bouton en lui-même doit faire 1cm de diamètre, ce qui correspond beaucoup plus à notre besoin.

5. La musique

Nous cherchons à ajouter de la musique à notre LEDris afin de rendre notre jeu plus vivant. C'est pourquoi nous devons regarder comment faire pour la jouer. Nous pensons utiliser un **module série de lecteur MP3** comme le Serial MP3 Player, il se branche sur le 5V, la masse et une sortie RX et une sortie TX. Il comprend un prise jack en sortie afin de relier le module à des haut-parleurs et un lecteur de carte SD pour pouvoir stocker les musiques plus simplement. Ce module est plutôt spécial car il ne demande pas de bibliothèque. Il faut simplement inclure SoftwareSerial.h puis mettre différentes commandes que nous expliquerons plus loin.

Il nous faudra également une **carte micro SD classique** qui nous permettra de stocker la musique mythique de Tetris.

6. La communication bluetooth

Nous devons utiliser une forme de communication radiofréquence, nous avons donc tout de suite pensé au Bluetooth. Nous avons trouvé le **module HC-06** qui nous permet d'établir une communication avec notre téléphone, ordinateur, ou un autre module HC-06. Dans notre cas, nous voulons utiliser notre ordinateur. Cependant il faut que l'on trouve une manière intéressante de l'implémenter dans notre projet, nous pensions au départ supprimer les boutons mais la communication serait trop lente pour jouer agréablement. Nous avons donc opté pour que le module Bluetooth envoie le score sur l'ordinateur.

Ce module possède quatre broches le +5V, la masse, une broche de réception RX et une broche de transmission TX.

Bibliothèques et méthodes utiles

- **Librairie HC-06 :** SoftwareSerial.h

Méthodes :

SoftwareSerial hc06(Rx,Tx) pour définir le port série et les broches servant à la communication.

hc06.begin(int) pour définir la vitesse de communication.

hc06.available() pour tester si des données sont disponible dans le buffer du port série.

hc06.read() pour lire les données du port série, un octet à la fois.

hc06.print(String) pour envoyer une chaine de caractères en ASCII.

hc06.write() pour envoyer des données, un octet à la fois.

- **Librairie bande led :** Adafruit_NeoPixel.h

Méthodes :

Adafruit_NeoPixel strip(nombre de LED, pin de l'arduino) initialise la bande de led.

strip.begin() à mettre au début du setup prépare le strip de led.

strip.show() applique les changement appliquer aux leds.

Strip.clear() remet toute les led à la couleur noir.

strip.setPixelColor(n, red, green, blue) affecte à la led n la couleur choisis (si 30 led n ira de 0 à 29 où 0 désigne la led la plus proche de la carte arduino) red, green et blue vont de 0 à 255 pour coder la couleur.

unit32_t Color = strip.Color(red, green, blue) permet de stocker dans la variable Color une certaine couleur.

Strip.setPixelColor(n, color) initialize le pixel n à la couleur color qui est une variable créer grâce à « unit32_t color = strip.Color(red, green, blue) ».

uint32_t color = strip.getPixelColor(n) récupère la couleur du pixel n

- **Librairie Serial MP3 player :** SoftwareSerial.h

Méthodes :

Ce module est spécial dans la mesure où il n'y a pas de méthode qui vont avec la librairie, il faut insérer les méthodes suivantes au début de notre programme.

static int8_t Send_buf[8] = {0} Le lecteur MP3 comprend les commande dans une chaine de 8 int.

#define NEXT_SONG 0X01

#define PREV_SONG 0X02

#define CMD_PLAY_W_INDEX 0X03 Nécessite le nombre du son. DATA REQUIRED

#define VOLUME_UP_ONE 0X04

#define VOLUME_DOWN_ONE 0X05

#define CMD_SET_VOLUME 0X06 Nécessite le volume, un nombre entre 0 et 30. DATA REQUIRED

#define CMD_PLAY_WITHVOLUME 0X22 Nécessite 0x7E 06 22 00 xx yy EF;(xx volume)(yy numéro du son) data is needeed

#define CMD_PLAY_DEV 0X09 Sélectionne le stockage DATA REQUIRED

#define DEV_TF 0X02 Hello I'm the DATA REQUIRED

#define SLEEP_MODE_START 0X0A

#define SLEEP_MODE_WAKEUP 0X0B

#define CMD_RESET 0X0C Redémarrage du morceau

#define CMD_PLAY 0X0D Reprendre la lecture

#define CMD_PAUSE 0X0E la lecture est en pose

#define CMD_PLAY_WITHFOLDER 0X0F DATA IS NEEDED, 0X7FAAABB AAA est le numéro du dossier qui doit être nommé avec 3 chiffre et BB le numéro du fichier.

#define STOP_PLAY 0X16

#define PLAY_FOLDER 0X17 DATA IS NEEDED 0X07BB où BB est le numéro du repertoire.

```
void sendComman(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); //datah
Send_buf[6] = (int8_t)(dat); //datal
Send_buf[7] = 0xef; //ending byte
for(uint8_t i=0; i<8; i++){
        }
}
```

Toute les lignes de code au-dessus sont à insérer directement dans le programme pour le bon fonctionnement du module MP3

Conclusion :

En fin de compte, nous avons besoin de certains composants pour notre projet, la carte Arduino MEGA, 6 boutons (3 directionnels + 2 rotationnels + 1 on/off music), 4m de bande Strip NeoPixel avec 60 LED par mètre pour avoir 200 LED, un module Serial MP3 Player et une carte micro SD pour le son et enfin un module Bluetooth qui nous permettra d'afficher les scores sur la sortie série de l'ordinateur.

Sources :

Electronoobs : (Idée du projet)
https://electronoobs.com/eng_arduino_tut104.php

Carnet du maker : (utilisation du Serial Mp3 Player)
https://www.carnetdumaker.net/articles/utiliser-un-lecteur-serie-de-fichiers-mp3-avec-une-carte-arduino-genuino/

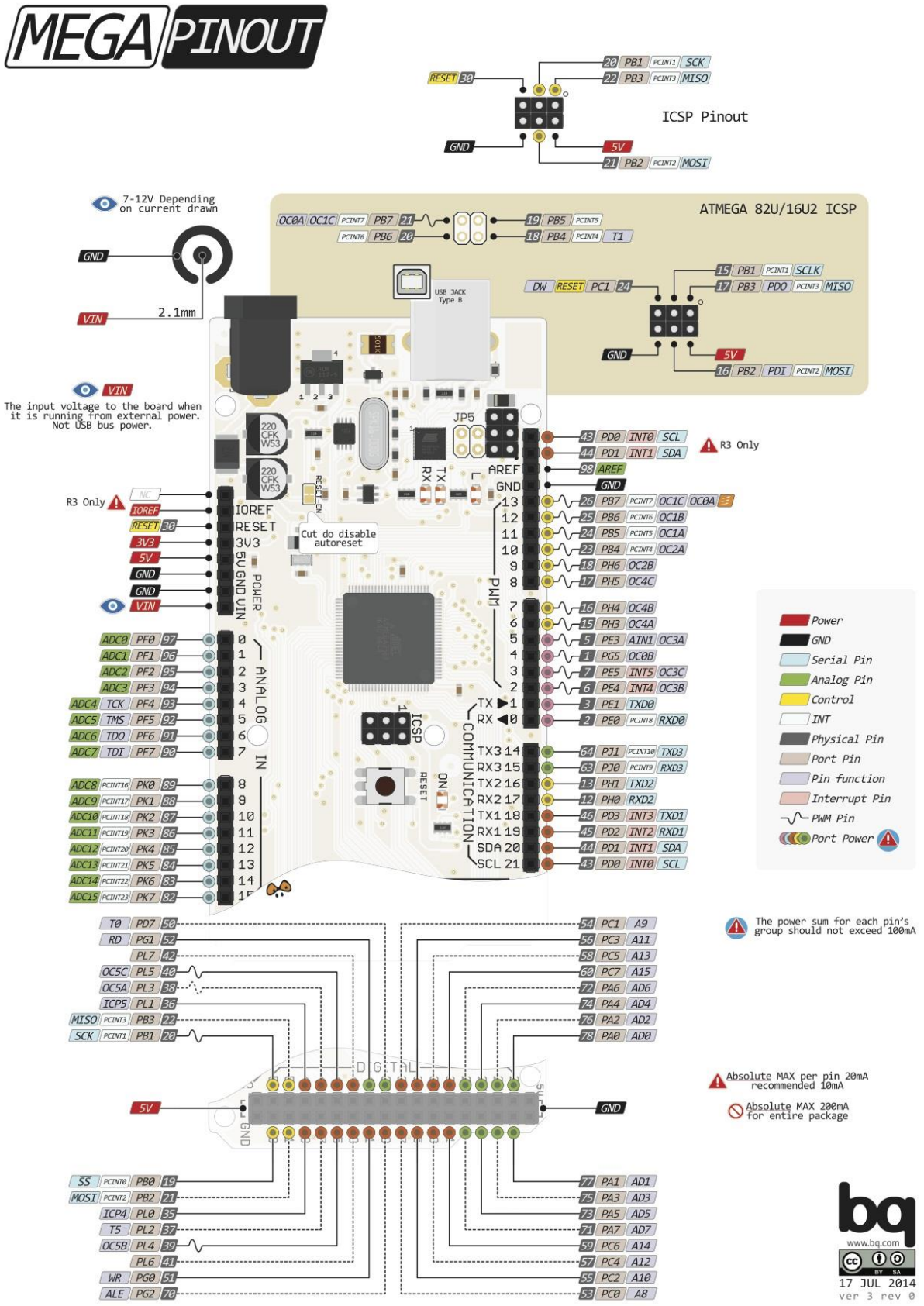
Aranacorp : (module Bleutooth)
https://www.aranacorp.com/fr/arduino-et-le-module-bluetooth-hc-06/#google_vignette

Arduino : (Arduino MEGA)
https://www.arduino.cc/en/Main/arduinoBoardMega2560/

learn adafruit : (librairie Neo Pixel)
https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-use

Github adafruit : (librairie Neo Pixel)
https://github.com/adafruit/Adafruit_NeoPixel/blob/master/README.md

Annexes :



Câblage arduino MEGA