

Design

Ecole supérieure
Électronique

Laboratoire R110 PROJ

Buzzer Wire Game

Réalisé par :

Santiago Valiante

A l'attention de :

Mr. Bovey et Mr. Castoldi

Dates:

Début du laboratoire : 14 Décembre 2022

Fin du laboratoire : 26 Janvier 2023

Table des matières

Table des matières	3
Description du produit	4
Schéma bloc	4
Alimentation	5
USB C	5
Régulateur de charge et ON/OFF	5
Dimensionnement	6
Accumulateur Liion	7
Régulateur 3,3V	7
USB to UART	8
uC	8
Quartz	8
Reset	9
Port de programmation	9
Périphériques	10
Game IO	10
Encodeur Rotatif PEC12	10
LCD	11
Buzzer	11
LED RGB	12
Flowchart	13
Firmware & Software	13
Explications des interactions externes	14

Description du produit

WireGame est un jeu qui consiste à guider une boucle métallique le long d'une longueur de fil en serpentín, sans toucher. La boucle et le fil sont connectés à une source d'alimentation de telle manière que, s'ils se touchent, ils forment un circuit électrique fermé.

Le but de ce projet est de développer un WireGame dit « intelligent » et optimisé pour qu'une personne mal entendant puisse y jouer. Il sera possible de gérer le menu du système à travers une application C# depuis un PC connecter en USB ou manuellement avec un encodeur.

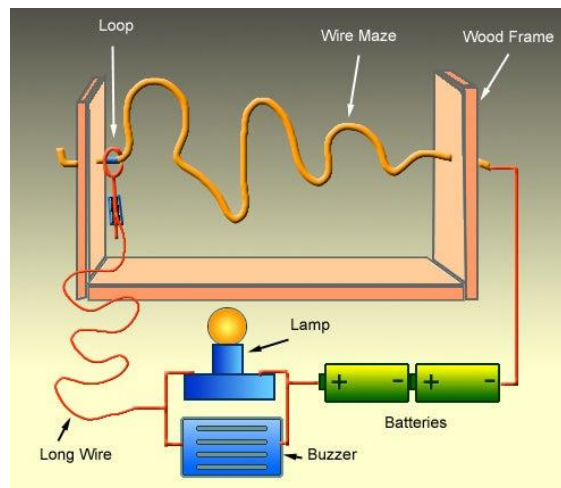
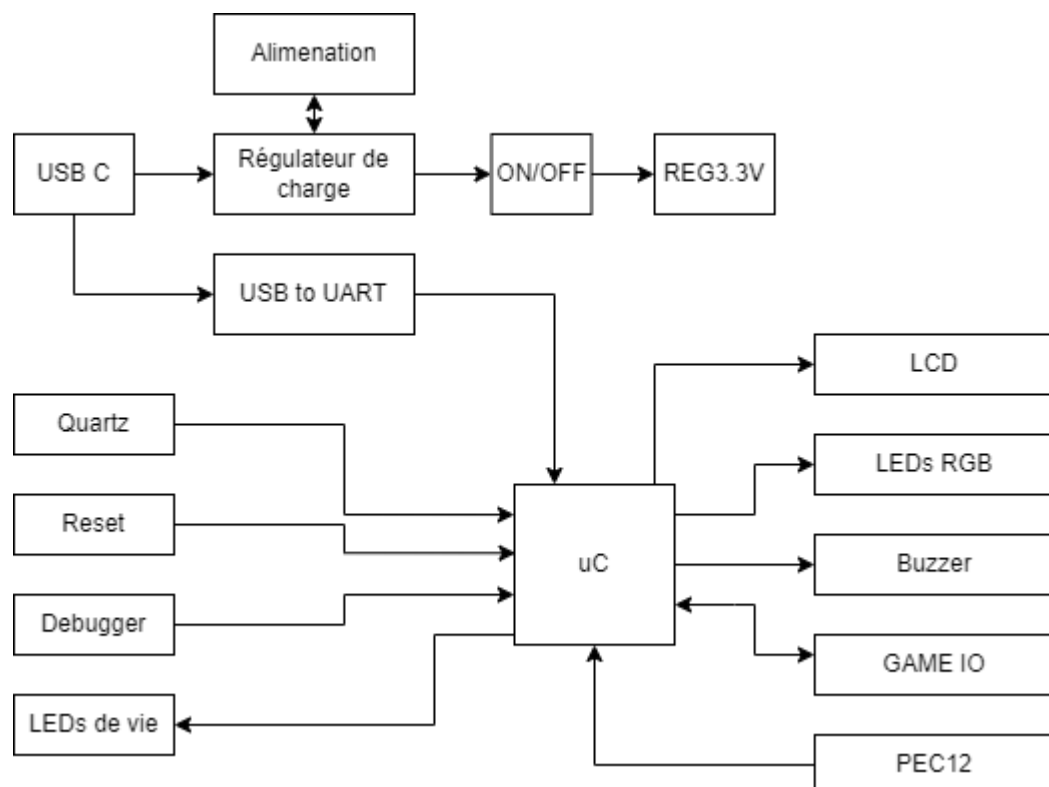


Figure 1

Schéma bloc



Alimentation

USB C

Afin de pouvoir communiquer avec un PC et recharger notre batterie Li-On. J'utilise un port USB C. Selon la note d'application (Figure 3), si nous sommes un device et que nous devons recharger notre système. Nous devons mettre deux résistances de 5k1 reliées à la masse aux sorties CC1 et CC2.

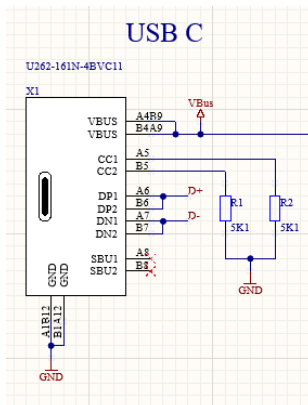


Figure 2 : Schéma USB C

3.2 UFP Rd Pull-Down Resistors.

An upstream facing port must connect a valid Rp pull-down resistor to GND (or optionally, a voltage clamp) to both CC1 and CC2 pins. A 5.1kΩ ± 10% is the only acceptable resistor if USB Type-C charging of 1.5A@5V or 3.0A@5V is to be used. The details are shown in the table below.

TABLE 7: VALID UFP RD PULL-DOWN RESISTOR VALUES

Rd Implementation	Nominal Value	Detect Power Capability?	Current Source to 1.7V - 5.5V
± 20% voltage clamp	1.1V	No	1.32V
± 20% resistor to GND	5.1kΩ	No	2.18V
± 10% resistor to GND	5.1kΩ	Yes	2.04V

Figure 3 : Annotation note USB C

Régulateur de charge et ON/OFF

Pour charger l'accumulateur, j'utilise un régulateur de charge **MCP73871**. Le modèle a été repris d'un design déjà utilisé à l'ETML-ES. J'utilise un switch afin de gérer l'alimentation de mon circuit.

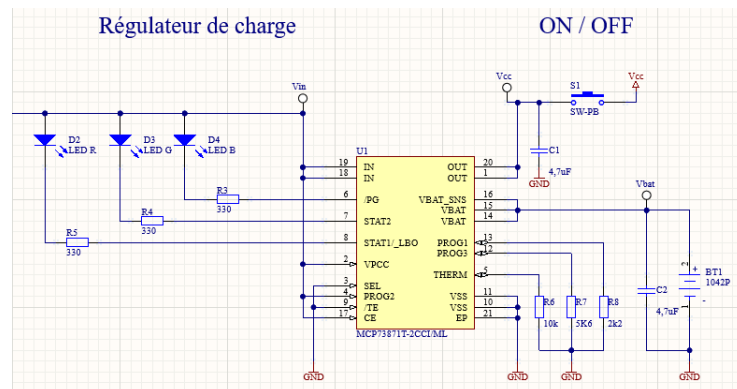


Figure 4 : Régulateur de charge et ON/OFF

Comme nous sommes en mode USB, la pin 3 (SEL) doit être mis à la masse.

3	SEL	I	Input type selection (low for USB port, high for AC-DC adapter)
---	-----	---	---

Puis, la pin 9 (TE) sera mise à l'état bas pour activer la sécurité.

9	TE	I	Timer Enable; Enables Safety Timer when active-low
---	----	---	--

Pour finir, la pin 17 (CE) à l'état haut pour activer la charge.

17	CE	I	Device Charge Enable; Enabled when CE = high
----	----	---	--

Pour les états de sortie, nous avons 3 LED's indicatives. Elles ne sont pas obligatoires, mais j'ai décidé de les mettre à titre d'indication pour connaître l'état de l'accu.

Dimensionnement

Pour le dimensionnement des résistances et des condensateurs, il faut d'abord connaître le courant maximum avec lequel on peut charger notre accumulateur (3400 mA). Les valeurs des condensateurs ont été prises de l'exemple d'application (Figure 5). De même pour la résistance sur la sortie THERM.

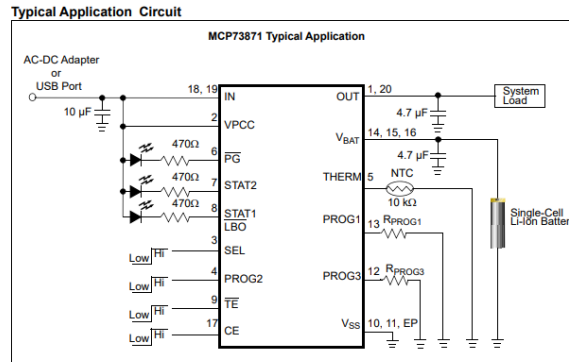


Figure 5 : Exemple MCP73871

Courant de régulation

Pour trouver la résistance Rprog1, il faut savoir le courant de régulation. Dans notre configuration, nous aurons un courant de 450 mA.

USB Fast Charge Current	I _{REG}	80	90	100	mA	PROG2 = Low, SEL = Low, (Note 2) T _A = -5°C to +55°C
		400	450	500	mA	PROG2 = High, SEL = Low, (Note 2) T _A = -5°C to +55°C

Figure 6: Configuration courant de charge

EQUATION 4-1:

$$I_{REG} = \frac{1000V}{R_{PROG1}}$$

Where:

R_{PROG} = kilo-ohms (kΩ)

I_{REG} = milliampere (mA)

$$R_{prog1} = \frac{1000}{450m} = 2k2$$

Figure 7 : Formule Rprog1

Courant de charge de terminaison

Pour trouver la résistance Rprog3, il faut savoir le courant de charge fixé depuis les informations de la batterie (3400 mA).

EQUATION 4-2:

$$I_{TERMINATION} = \frac{1000V}{R_{PROG3}}$$

Where:

R_{PROG} = kilo-ohms (kΩ)

I_{REG} = milliampere (mA)

The recommended PROG3 resistor values are between 5 kΩ and 100 kΩ.

$$I_{ter} = 0,05C = 0,05 * 3400m = 170 mA$$

$$R_{prog3} = \frac{1000V}{170 mA} = 5k8 \Rightarrow E24 = 5k6$$

Figure 8 : Formule Rprog3

Accumulateur Li-ion

J'ai choisi de prendre un accumulateur à la place de piles, car cela est plus économique. Dans le cahier des charges une autonomie de 2h minimum a été demandé. Après une estimation du courant, dans le pire des cas (les leds RGB sont allumées pour une animation), nous avons donc une consommation d'environ 500 mA.

$$Capacit\acute{e} = Autonomie * Consommation = 2 * 500m = 1000 mAh$$

J'ai choisi de prendre le même accumulateur que mes camarades afin de faciliter la logistique. L'accumulateur a donc une capacité de 3400 mAh. J'aurais donc une autonomie de 6.8 h.

$$Autonomie = \frac{Capacité}{Consommation} = \frac{3400m}{500m} = 6.8 h$$

Régulateur 3,3V

Comme on est alimenté par un accumulateur la tension de sortie de celui-ci peut varier au fur et à mesure qu'il se décharge entre 2.5V et 5.5V. Un étage de remises à niveaux est nécessaire pour garantir une tension de 3,3V sur le micro et les divers modules qui fonctionnent à cette tension. J'ai repris un design de l'ETML-ES qui utilise le modèle MAX1793, il est notamment disponible dans les stocks de l'ES.

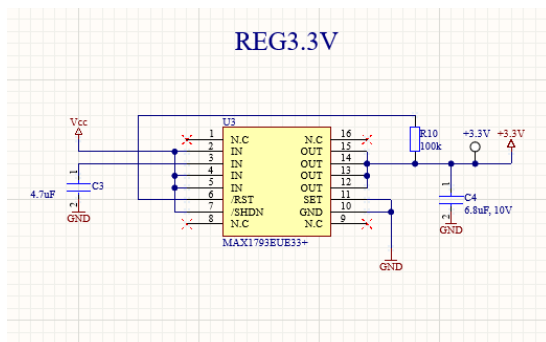


Figure 10: Régulateur 3.3V

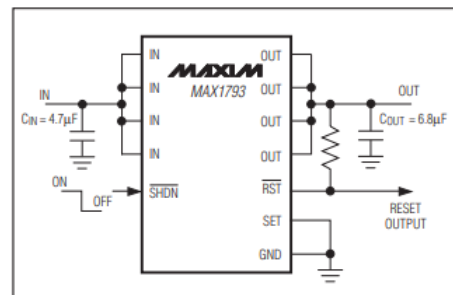


Figure 9 : Circuit typique Régulateur 3.3V

USB to UART

Afin de pouvoir se connecter au PC, on passera par un port USB C. Il faudra donc pouvoir lire les informations USB qui arriveront sur la carte. Pour cela un convertisseur USB vers UART sera utilisé.

Cela simplifie la prise en charge par le microcontrôleur de traiter une communication UART, plutôt qu'une communication USB lourde à traiter.

J'ai donc opté pour le composant « CY7C64225 » qui a déjà été utilisé lors d'un projet de l'ETML-ES. Pour le montage, un exemple du cas d'application alimenté en 3.3V a été suivi.

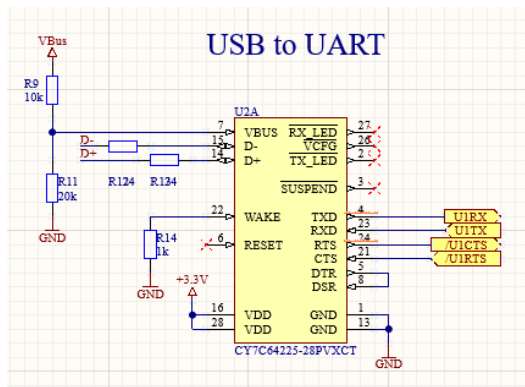


Figure 11: USB to UART

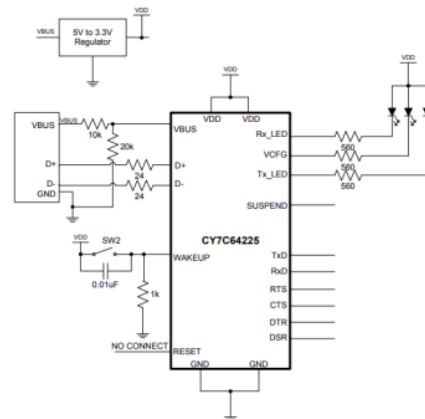


Figure 12: Exemples USB to UART

uC

Pour le choix du microcontrôleur, la famille PIC32 de Microchip a été conseillée dans le cahier des charges. Le 28 pin étant trop petit, j'ai donc choisi de prendre le PIC32MX250F128D à 44 pins, car j'ai besoin de seulement de la communication UART et de GPIO.

Quartz

Le projet comprenant une communication UART. J'ai besoin d'un oscillateur pour avoir une vitesse en baud rate la plus précise possible.

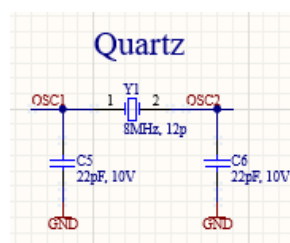


Figure 13: Quartz

Le calcul des condensateurs a été fait grâce au $CL = 18\text{pF}$ et le $C0 = 7\text{pF}$ du quartz utilisé.

$$C5 = C6 = 2 * (CL - C0) = 2 * (18 - 7) = 22pF$$

Reset

Afin de pouvoir reset le microcontrôleur lors de la phase de développement, un bouton poussoir a été placé. Le dimensionnement a été repris du kit micro utilisé au labo.

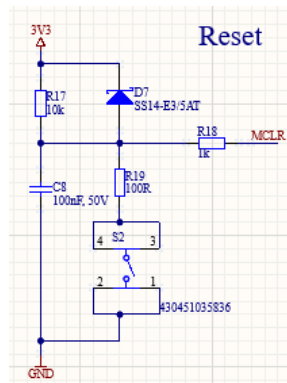


Figure 14: Reset

Port de programmation

Le JTAG est là pour pouvoir programmer le microcontrôleur en chargeant le firmware à l'intérieur. Je pourrai également connecter le débogueur pour déboguer en temps réel le firmware.

Le dimensionnement a été repris également du kit micro utilisé au labo. Il a été désigné pour accueillir un debugger « ICD3 ».

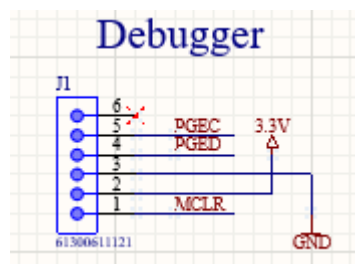


Figure 15 : Port de programmation

Périphériques

Game IO

Nous avons trois pull-up pour maintenir un état haut sur les entrées start, stop et wire. Lorsqu'on vient relier ces entrées avec la manette connectée au GND (pin 4 du header), nous aurons un état bas.

Il sera donc possible de détecter si nous avons touché le fil de cuivre ou si nous sommes au départ ou à l'arrivée du jeu.

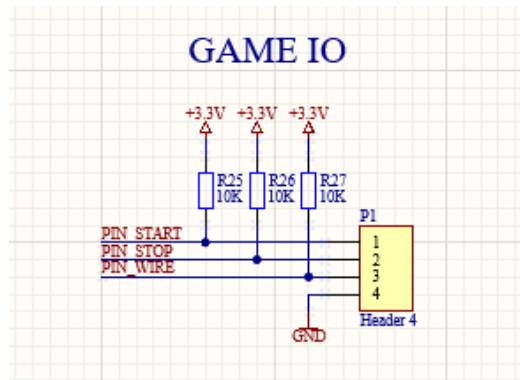


Figure 16: Game IO

Encodeur Rotatif PEC12

Afin de pouvoir naviguer dans le menu et valider des actions, l'utilisation d'un encodeur a été demandé. Je me suis inspiré de celui du KIT microcontrôleur de l'ES.

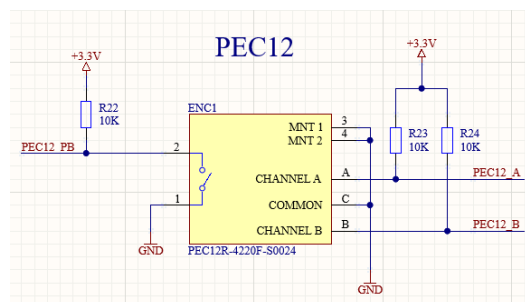


Figure 17 : PEC 12

LCD

Afin d'afficher le menu, l'utilisation d'un LCD a été demandé. Je me suis inspiré de celui du KIT microcontrôleur de l'ES. Pour le back light du LCD, j'ai dû redimensionner les composants afin de fournir 20 mA.

Backlight Supply Current	I _{LED}	V _{LED} = 3.0V	10	20	30	mA
--------------------------	------------------	-------------------------	----	----	----	----

Figure 18: Back light

$$R = \frac{U_j}{I} = \frac{0,7}{20m} = 35 \Rightarrow 33\Omega$$

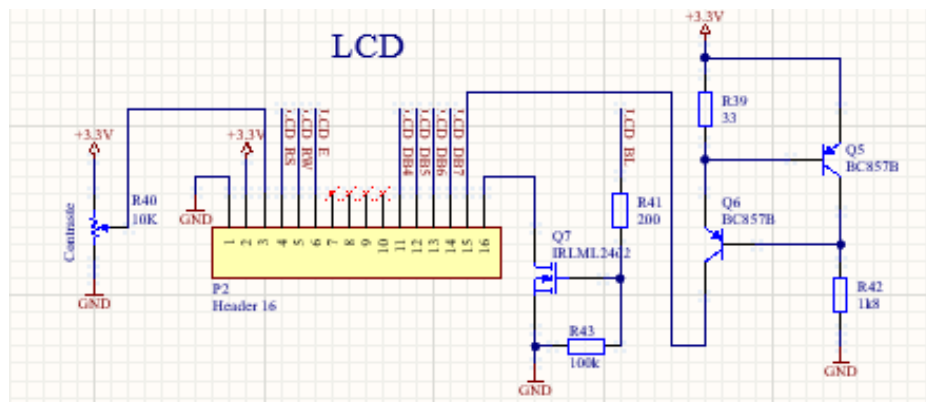


Figure 19: LCD

Buzzer

Afin d'avoir une indication sonore, j'utilise un buzzer avec une tension de fonctionnement de 2.5V à 4.5V et d'une résistance de 16Ω. Le courant est limité à 20 mA.

$$I = \frac{U}{R} = \frac{3,3}{150 + 16} = 19.8 \text{ mA}$$

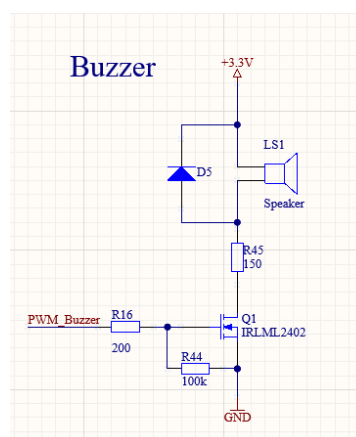


Figure 20:

LED RGB

Afin d'avoir une animation visuelle, des leds RGB sont utilisées. Il me faut des leds assez puissantes pour pouvoir diffuser de la lumière dans un plexiglass opaque. Pour une question de logistique, je vais prendre les mêmes leds qu'un camarade.

Je ne vais pas utiliser le blanc, les couleurs me suffisent. Une led consomme 250 mA, pour une luminosité à 75%, il faudra donc une résistance de 22Ω .

The following graph represents typical performance of each LED die in the XLamp XM-L Color Gen 2 LED.

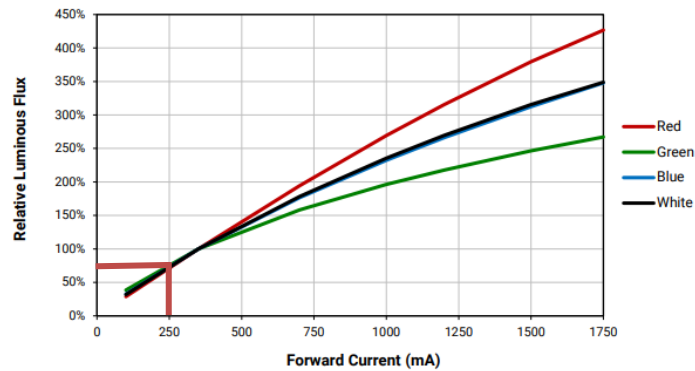


Figure 21: Courant - Luminosité

$$R = \frac{3.3 - UD}{I} = \frac{3.3 - 2.1}{250m} = 24 \Rightarrow 22\Omega$$

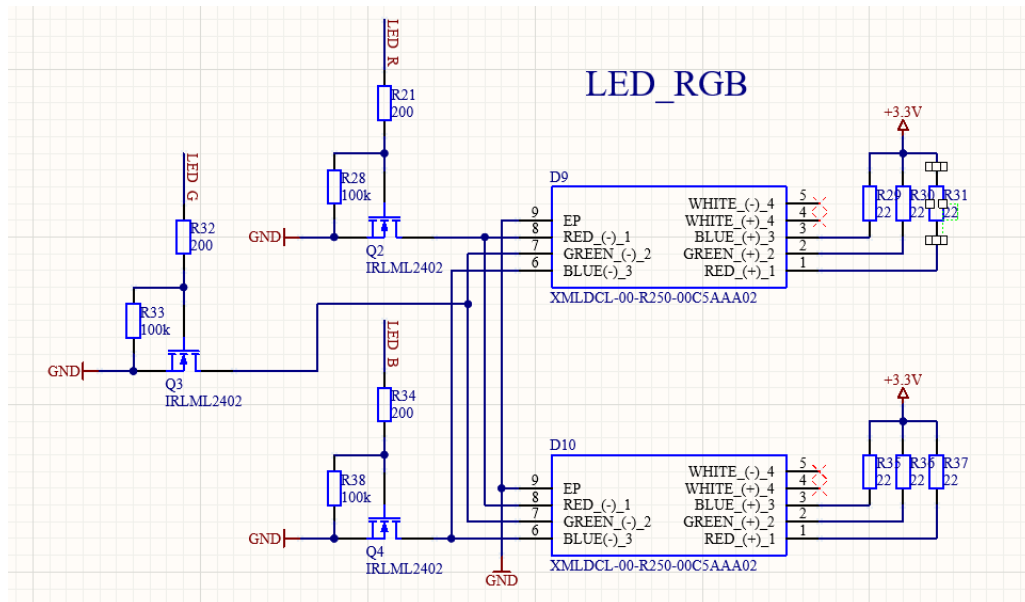


Figure 22: LED RGB

Flowchart

Firmware & Software

Application C#

Réaliser une interface de statistiques : meilleurs temps, nombre de fois toucher.
Réaliser différent mode de jeu : compte à rebours, nbr de touche max, ect.

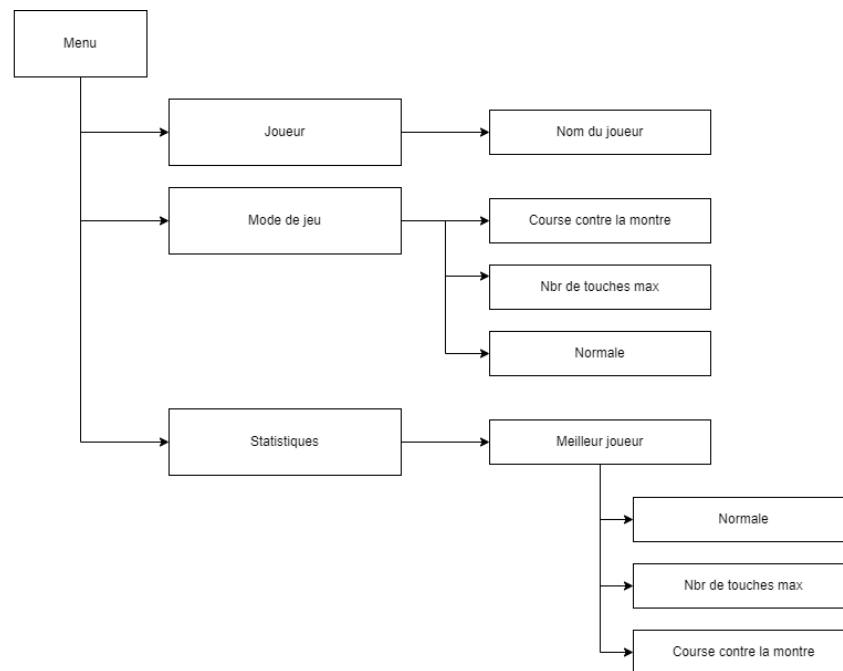
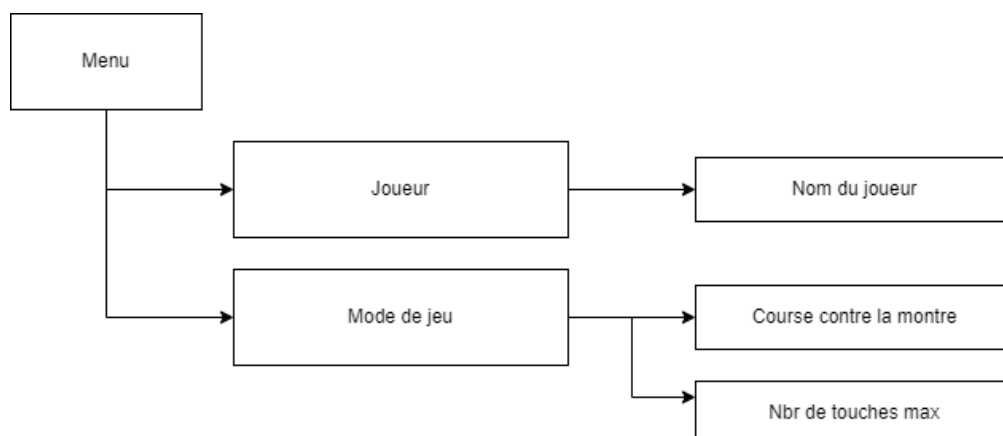
Flowchart

Figure 23

Firmware uC

Réaliser différent mode de jeu : compte à rebours, nbr de touche max, ect.
Gestion joueur.

Flowchart

Explications des interactions externes

Le joueur pourra interagir directement avec le système. Il y aura un bouton ON/OFF pour activer ou désactiver l'alimentation. A l'aide d'un affichage LCD et d'un encodeur, il pourra naviguer dans le menu. Mais, il pourra également interagir avec le système via un PC connecté via un port USB pour naviguer également dans le menu et en plus voir les statistiques des joueurs. Pour lancer le jeu, il suffira de décoller la boucle de la plaque de cuivre START et de même pour y mettre fin mais à l'arrivée (STOP).

Lausanne ETML-ES, 26.01.23.

Santiago Valiante