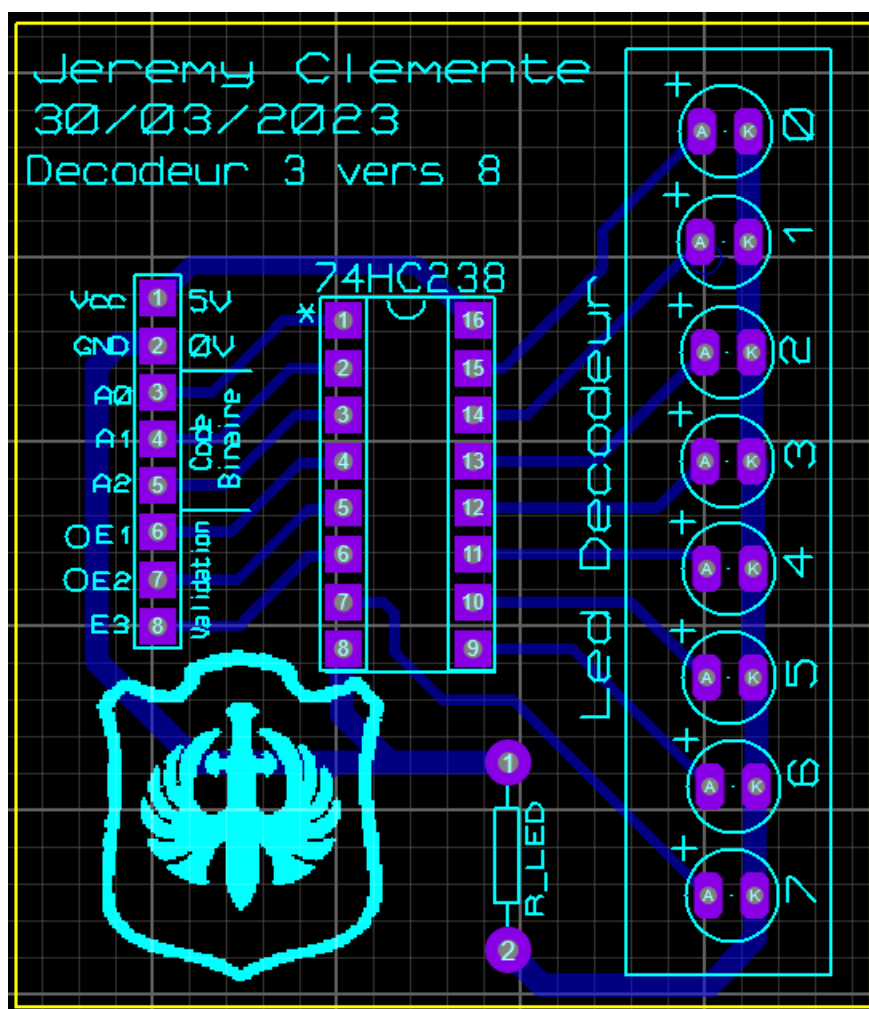


DÉCODEUR 74HC238 CODE C++

BIBLIOTHÈQUE ET PCB TEST



SOMMAIRE

Introduction	3
Carte électronique	3
Présentation :	3
Isis :	3
Ares :	4
Partie Code C++	5
Présentation :	5
Initialisation :	5
Void Setup :	5
Void Loop :	6
Epilogue.....	7
Qr code projet :	7



INTRODUCTION

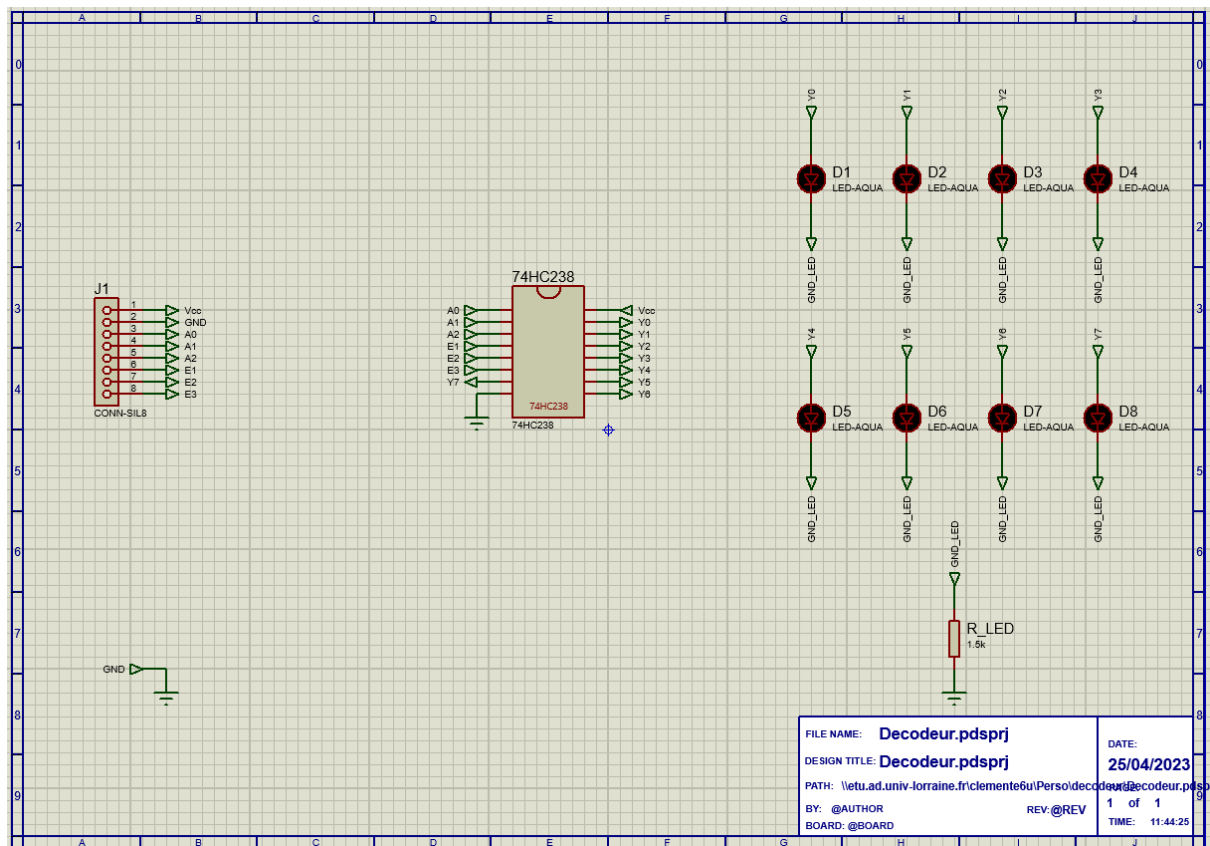
Ce rapport présentera une carte pour montrer le fonctionnement de la bibliothèque pour un décodeur. Pour utiliser le décodeur, on se servira d'une carte électronique de test fait exprès pour. Ce rapport va expliquer comment fonctionne la carte électronique avec un décodeur 74HC238. Ensuite, le code exemple de la bibliothèque sera expliqué dans ce document. Mais ce Rapport n'indiquera pas comment fonctionne un décodeur et comment utiliser la bibliothèque. Il y a deux markdown dédié à cela.

CARTE ÉLECTRONIQUE

Présentation :

Pour utiliser la bibliothèque du décodeur on va se servir d'une carte électronique. Des leds vont symbolisées les sorties du décodeur. Les entrées binaires et de validation ainsi que l'alimentation sont sur des Conn-sil ou Conn-dil au choix.

Isis :



Sur le Isis, on peut voir les interconnexions. Le Conn-sil est bien relié au décodeur 74hc238 et les sorties du décodeur sont bien connectées aux leds comme témoin. Une résistance protège les leds. La valeur de la résistance est de 1500 Ohm. Cette valeur est bien supérieure, une résistance de 330 Ohm aurait suffi, mais une plus grande valeur

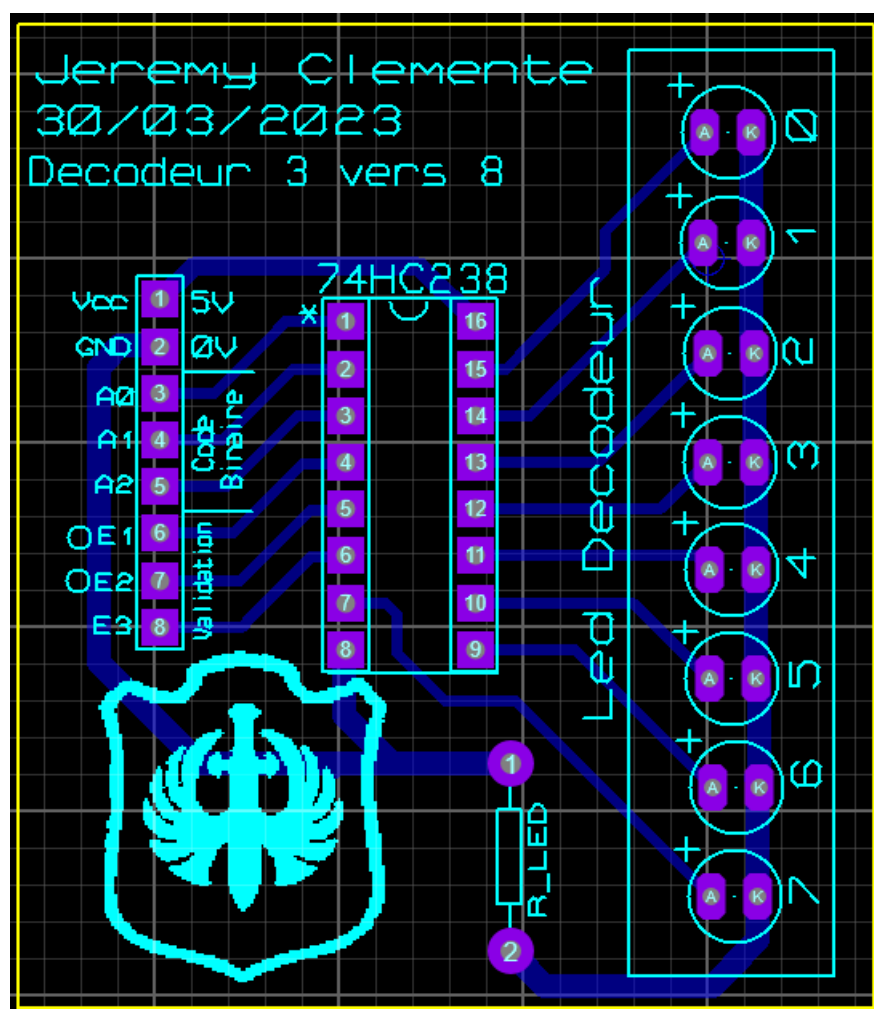


améliore leur durée de vie et les leds servent de témoin et non à éclairer. La résistance des leds est bien connectée au GND.

Le composant principal est un décodeur 74HC238 qui est un décodeur 3 vers 8, mais un autre décodeur fonctionnant avec les mêmes tensions et possédant les mêmes emplacements d'entrées et sorties peut tout à fait être utilisé.

Ares :

Après avoir fait le schéma électronique (Isis), on peut s'occuper du routage que voici :



Le Ares est simple face sans strap. Les broches d'entrées sont commentées et possèdent des indications comme Vcc, GND, A0... Les broches possèdent aussi des informations comme 5V, Code Binaire... Le rond à côté des broches E1 et E2 signifie que ces entrées sont inversées.

Il y a d'autres informations sur la carte, comme sur le décodeur son sens a été indiqué grâce au demi-cercle. Il y a d'autre indications sur les leds de sortie comme le numéro des pattes de sortie et les leds sont triées dans l'ordre de la sortie 0 à 7. Le sens des leds est aussi indiqué grâce au petit +.

Présentation :

Cette partie est dédiée au code exemple de la bibliothèque JC_decodeur. Pour utiliser ce code, on va se servir de la carte électronique vu précédemment. Le code sera contenu dans une carte Arduino Uno mais les autres cartes Arduino peuvent tout aussi bien être utilisées. Il est aussi possible de prendre une autre carte comme des Esp32 ou encore des Attiny 85 ou 84.

Le code est le seul exemple fournit avec la bibliothèque et il n'utilise pas toutes les fonctions possibles dans celle-ci, pour voir les autres fonctions disponibles sur la bibliothèque, il y a un markdown dessus.

Le code sert de moyen pour tester si la carte électronique fonctionne et aussi présente des fonctions possibles avec les décodeurs.

Initialisation :

Avant que l'on aille dans la boucle void setup, il faut exécuter deux ou trois actions :

```
1 //ajout bibliothèque
2 #include <JC_decodeur.h>
3
4 JC_decodeur decodeur1;
5
```

Il faut en premier lieu inclure la bibliothèque, cela se fait via le #include. Ensuite, une fois la bibliothèque incluse dans le code, on doit créer un décodeur pour pouvoir utiliser ces fonctions et donc le piloter.

La deuxième commande sert à créer le décodeur, on crée un objet de la classe décodeur, cela permettra par la suite d'utiliser des commandes grâce à cet objet créer. Ici, il s'appelle decodeur1, mais il pourrait avoir un autre nom comme decodeur_principal par exemple.

Void Setup :

Après avoir initialisé la bibliothèque et créer un objet décodeur, on peut passer au void setup pour initialiser le décodeur grâce à ces fonctions.



```

6 void setup() {
7
8     //Initialisation des entrées de validations
9     decodeur1.Init_Validation(5,6,7);
10    //Initialisation des entrées du code binaire
11    decodeur1.Init_CodeBinaire(2,3,4);
12
13 }

```

La première fonction permet d'initialiser les entrées de validations en disant sur quelles broches elles sont connectées. L'ordre des entrées est le suivant E1 sur la broche 5, E2 sur la broche 6 et enfin E3 sur la broche 7.

La deuxième ligne du décodeur sert à initialiser les broches des entrées du code binaire. C'est-à-dire à dire sur quelle broche sont connectées les entrées Binaires. L'ordre dans la fonction est le suivant, A0 est connecté sur la broche 2, A1 est connecté sur le pin 3 et A2 est connecté sur le pin A2.

Void Loop :

Après avoir initialisé les pins du décodeur dans la fonction void setup, on peut utiliser les commandes du décodeur :

```

15 void loop() {
16
17     //activation de l'entrée 3 pour activé les sorties du decodeur
18     decodeur1.Validation_E3(1);
19     //allume les sortie du decodeur une a une
20     for (int i = 0; i < 8; i ++) {
21         decodeur1.Code_decimal(i);
22         delay(1000);
23     }
24
25 }

```

Dans la partie qui tourne en boucle, on fait plusieurs actions. Dans un premier temps, on met la sortie de validation E3 à 1 ceci est lié à la spécification du décodeur 74HC238 car l'on souhaite toujours avoir une sortie activée.

La deuxième partie, on utilise une boucle for. Cette boucle va de 0 à 7 grâce à la variable i. Dans cette boucle, on allume la sortie i qui va varier de 0 à 7 grâce à la boucle. Ensuite, on attend 1 seconde pour avoir le temps de voir la sortie allumée. La fonction decodeur1.Code_decimal va prendre en entrée un nombre de 0 à 7 pour allumer la sortie correspondante en faisant des divisions.



EPILOGUE

Voici la fin du rapport sur le décodeur 74HC238 qui présentait la carte électronique ainsi que le code d'exemple de la bibliothèque lié à la carte.

Ce document n'explique pas comment utiliser la bibliothèque du décodeur 74HC238, il y a un markdown fait à cet effet. Ce document n'explique pas non plus ce qu'est un décodeur, il y a un autre document prévu à cet effet.

Qr code projet :

