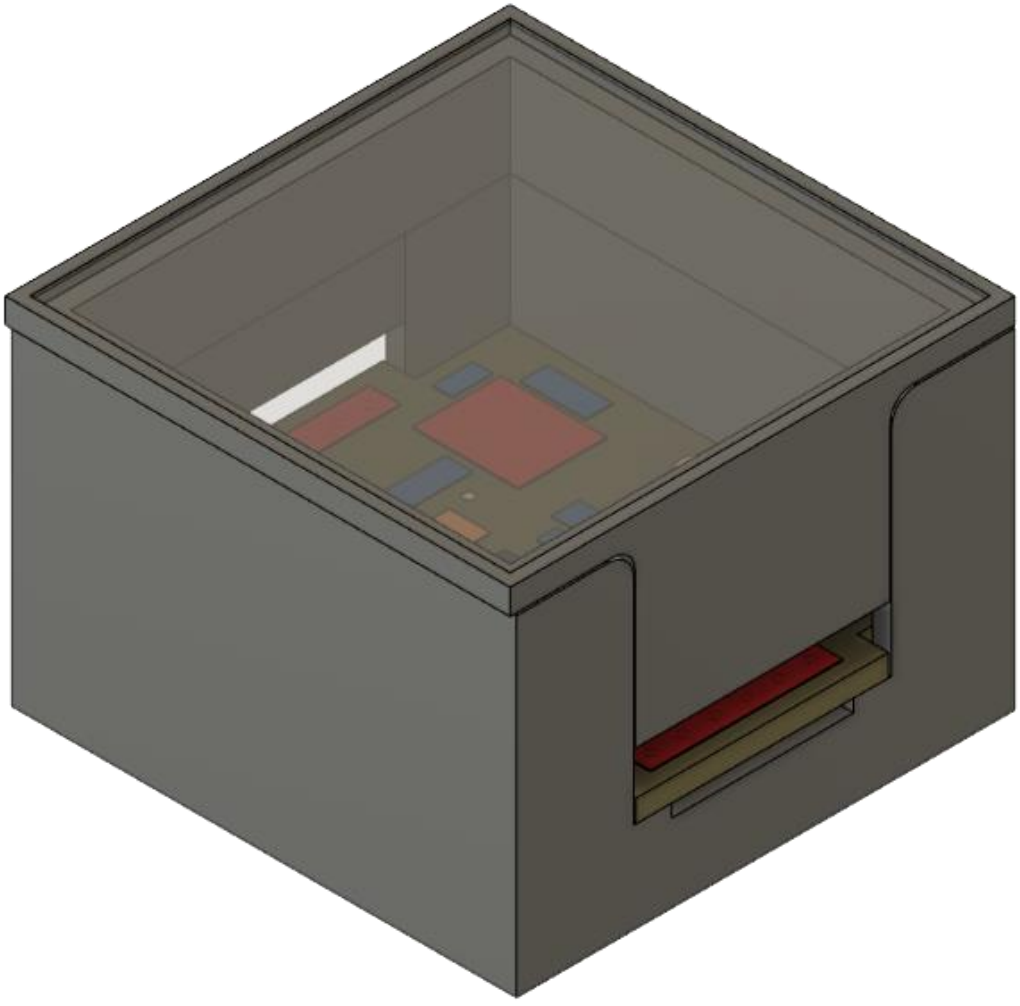
université angers

INSTITUT UNIVERSITAIRE DE TECHNOLOGIE
UNIVERSITÉ D'ANGERS

Rapport de SAE - BUT 3 GEII

SAE Case d'échecs interactive



Encadrant : Mr Philippe LUCIDARME

Killian GAIFFE -Louis BETOU

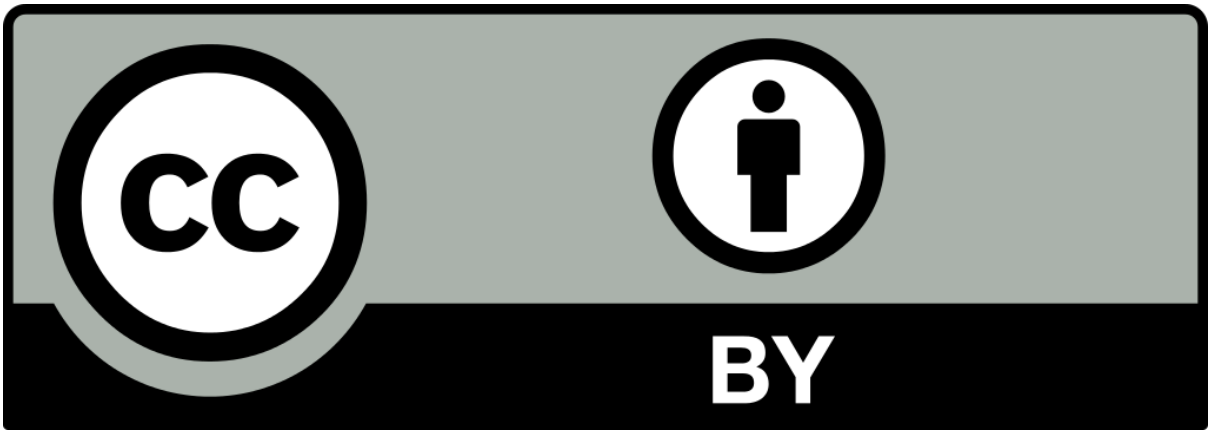
Année 2024-2025

DÉCHARGE DE RESPONSABILITÉS

Ce rapport présente le travail réalisé par un groupe d'étudiants dans le cadre d'un projet pédagogique. Les auteurs et l'Université d'Angers ne garantissent pas que l'information, les documents, la méthodologie et le matériel présentés dans ce document soient complets, conformes à l'état de l'art et exacts ni n'assurent en toutes circonstances la sécurité des biens, des personnes et des utilisateurs. Les auteurs et l'Université d'Angers ne seront pas tenus responsables des dommages éventuels qui pourraient résulter de l'utilisation du contenu du présent rapport.

LICENCE

GAIFFE Killian et BETOU Louis, auteurs du présent rapport, publions et divulguons celui-ci sous la Licence Creative Commons suivante « CC BY » pour le monde entier et pendant la durée légale de protection des droits d'auteur. Cette licence autorise la représentation, la reproduction, la modification, la création d'œuvres dérivées et l'utilisation y compris à des fins commerciales sous réserve de mentionner les noms et prénoms des auteurs.



REMERCIEMENTS

● Zachary Kodja & Malo Rupin, étudiants en BUT GEII ayant fortement contribués au projet puisque ce sont eux qui l'ont lancé l'an passé.

● M. Philippe LUCIDARME, responsable projet, pour ses conseils éclairés, sa patience, sa disponibilité et sa confiance qu'il nous a accordée lors de ce projet ; ainsi que pour sa mise à notre disposition de tous les moyens disponibles afin de mener à bien ce projet.

●Geraldine, du laboratoire d'électronique/magasin, qui nous a fortement aidée dans la soudure des magnétomètres, que nous n'aurions pas pu réaliser seuls.

Table des matières

1. Introduction

Dans le cadre de notre SAE au semestre 5, nous avons repris un projet mené l'année précédente par deux étudiants en BUT GEII, Zachary Khodja & Malo Rupin. L'objectif premier de leur projet était d'évaluer la faisabilité du projet et de produire un premier prototype en prenant ainsi main les composants, soit ici principalement le capteur à effet hall ainsi que le µC.

De notre côté, nous avons décidé de reprendre leur projet afin de pouvoir l'aboutir davantage. Pour ce faire nous avons décidé de reprendre le même capteur à effet hall ainsi que les mêmes LEDs, nous avons en revanche décidés d'utiliser un SoC ESP32 à la place de l'Arduino MEGA, plus simple et amplement suffisant pour cette application.

2. Cahier des charges

Le Cahier des charges pour la réalisation de cette case d'échec connecté est divisé en deux parties. Une première constituée d'éléments ou objectifs déjà réalisés ou atteints par le groupe ayant entamé ce projet l'année précédente. Puis une seconde composée d'améliorations de ce qui a déjà été proposé ainsi que de nouveaux objectifs.

Projet existant, objectifs à reconduire :

- Détection de la présence d'une pièce contenant un aimant.
- Détection de l'orientation d'une pièce contenant un aimant.
- Cases chainables.
- Retours visuels à la surface de la case.

Notre cahier des charges, objectifs à mettre en œuvre :

- Homogénéisation de la lumière à la surface de la case.
- Optimisation du placement des composants.
- Réduction de la taille de la case.
- Réalisation d'une carte intégrable dans une case.
- Réalisation de la case.

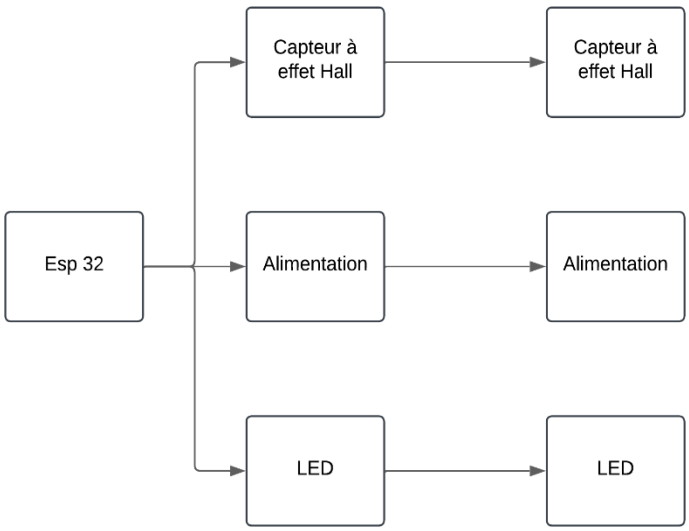


Figure 1 Synoptique simplifié d'un ESP 32 et de deux cases chainées

3. Choix technologiques

Ici nous parlerons des composants électroniques que nous avons choisis mais aussi des matériaux & méthodes de conception utilisées dans notre projet en énumérant leurs avantages et pourquoi nous avons décidés de les utiliser eux plutôt que d'autres équivalents.

3.1. Composants Electroniques

Allegro A31301 : *(magnétomètre, capteur à effet hall)*

Ce capteur à effet hall ayant été utilisé l'an dernier, ses avantages, son efficacité ainsi que sa correspondance avec les besoins de notre projet avait déjà été démontrée. Sa prise en main fût également simplifiée par les explications laissées par les anciens auteurs du projet dans leur dossier technique.

Figure 2 Modèle 3D du composant Allegro A31301

Les principaux avantages de l'Allegro A31301 :

- Capte les champs magnétiques sur 3 axes (coordonnées X, Y, Z)
- Jusqu'à 16 adresses I²C assignable par jeu de résistances
- Jusqu'à 127 adresses I²C en les écrivant directement dans l'EEPROM

Voltage on AD1, V_{A1} (× V_{CC_IO})	Voltage on AD0, V_{A0} (× V_{CC_IO})	4-Bit Code from ADR1 and ADR0 Voltages				Peripheral Address Bits							Slave Address
		E3	E2	E1	E0	A6	A5	A4	A3	A2	A1	A0	
0	0	0	0	0	0	1	1	0	0	0	0	0	96
	0.33	0	0	0	1	1	1	0	0	0	0	1	97
	0.67	0	0	1	0	1	1	0	0	0	1	0	98
	1	0	0	1	1	1	1	0	0	0	1	1	99
0.33	0	0	1	0	0	1	1	0	0	1	0	0	100
	0.33	0	1	0	1	1	1	0	0	1	0	1	101
	0.67	0	1	1	0	1	1	0	0	1	1	0	102
	1	0	1	1	1	1	1	0	0	1	1	1	103
0.67	0	1	0	0	0	1	1	0	1	0	0	0	104
	0.33	1	0	0	1	1	1	0	1	0	0	1	105
	0.67	1	0	1	0	1	1	0	1	0	1	0	106
	1	1	0	1	1	1	1	0	1	0	1	1	107
1	0	1	1	0	0	1	1	0	1	1	0	0	108
	0.33	1	1	0	1	1	1	0	1	1	0	1	109
	0.67	1	1	1	0	1	1	0	1	1	1	0	110
	1	1	1	1	1	I2C_SLV_ADDR							Programmed to 111 at the factory

Figure 3 Tableau de Décodage de l'adresse I2C esclave

LEDs WS2812 : *(LEDs chainables)*

Ces LEDs ont le grand avantage de pouvoir être chainable et cela très facilement en reliant leur signal de commande en série. Ceci était indispensable dans notre application. Sans cela nous aurions dû adresser chaque LED à une sortie du µC, cela aurait rendu notre application impossible.

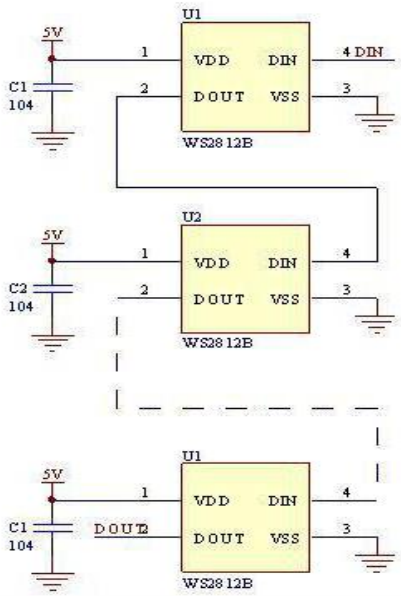


Figure 4 Schéma structurel de 3 Leds chainées

3.1.2. Prise en main

FireBeetle ESP32 : SoC (System on Chip)

Le choix de l'ESP32 nous a été proposé, auparavant c'est un Arduino MEGA qui avait été utilisé. L'ESP32 à l'avantage d'être plus compact, il permet de délivrer 2 tensions d'alimentations pour nos périphériques à savoir, le 3.3V mais aussi du 5V, notamment utile pour l'alimentation de nos LEDs. Il offre également un nombre de GPIO bien suffisant pour notre projet.

Figure 5 FireBeetle 2 ESP32-E

Allegro A31301 : Magnétomètre

Le magnétomètre allegro A31301 ayant été choisi par l'ancien groupe, nous avons décidé de le réutiliser voyant qu'il avait bien fonctionné pour eux. Nous devions donc le prendre en main, pour ce faire nous devions récupérés dans le moniteur série les valeurs du magnétomètre sur l'axe Z (axe vertical). Cela en interrogeant les registres suivants : REGISTER_LSB_Z & REGISTER_MSB_Z

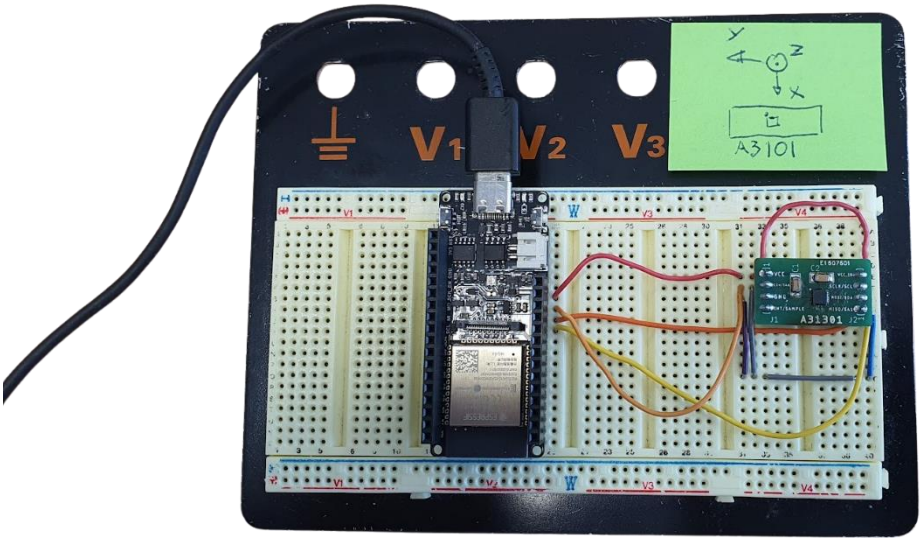


Figure 6 Montage ESP32 + Allegro A31301

LEDs WS2812 : LEDs

Pour ce qui est de la prise en main des LEDs, nous avons utilisé la librairie BonezegeiWS2812, celle-ci permettant de contrôler les LEDs de façon à leur appliquer différentes couleurs, effets, etc... Tout cela en les organisant en matrice 8x8. Permettant de sélectionner chaque LED indépendamment de façon intuitive.

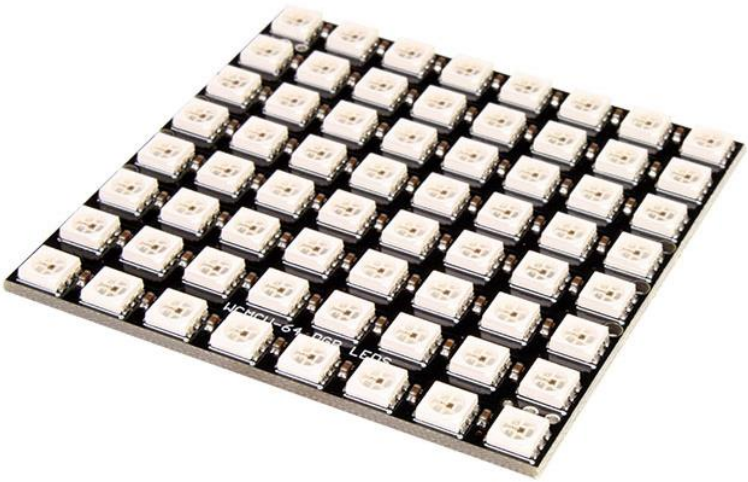


Figure 7 Matrice de LEDs RGB 8x8

3.2. Matériau d’impression 3D

Afin de créer une case d’échecs la plus réaliste possible, nous avons décidés de la modéliser en 3D pour ensuite l’imprimer en 3D. Le choix du PLA fût évident pour nous assez évident par sa simplicité de mise en œuvre, en effet la résine est beaucoup plus difficile à mettre en place, elle est essentiellement réservée pour faire des impressions très précises tel que des figurines par exemple. Pour des formes assez géométriques comme notre casing, on préfèrera utiliser de PLA, bien plus propre dans son utilisation et nécessite donc beaucoup moins de post-traitement (lavage à l’alcool isopropylique, à l’eau ainsi qu’un passage à l’UV).

La présence d’une imprimante BambuLab directement dans la salle de projet nous a également conforté dans notre choix, sa rapidité d’impression nous a également permis de faire plusieurs essais afin de prototyper notre casing en plusieurs étapes, nous permettant ainsi d’ajuster et de corriger les défauts au fur et à mesure.

La résine étant tout de même généralement plus solide que le PLA, elle a néanmoins une flexibilité très réduite. Globalement, elle pourra supporter de plus grosses charges ou contraintes mécaniques mais vivra très mal les flexions, torsions, elle sera donc beaucoup plus cassante que le PLA qui lui aura une bien meilleure flexibilité mécanique.

Figure 8 Imprimante 3D Bambu Lab X1-Carbon Combo

4. Réalisations

4.1. Structure de la carte

La structure du montage est restée sensiblement la même que pour la version précédente du projet, c'est-à-dire une partie alimentation provenant du µC, cette dernière viendra donc alimenter notre capteur à effet hall ainsi que nos LEDs.

Le magnétomètre (capteur à effet hall) quant à lui, viendra récupérer les données du champs magnétiques et les transmettra en I²C à notre ESP32 qui lui viendra ensuite piloter nos LEDs suivant les instructions que nous lui auront donnés dans notre code C++.

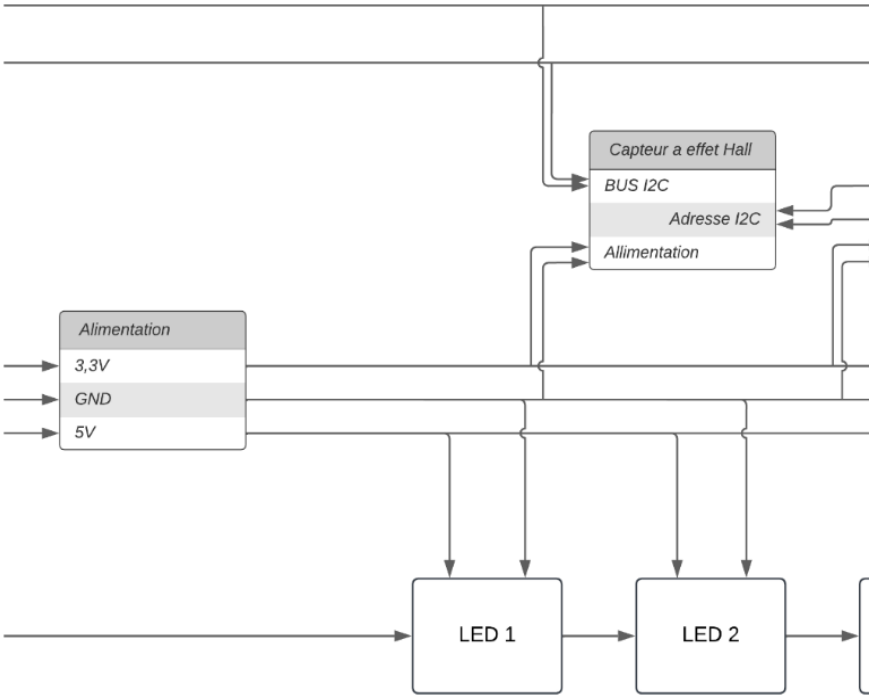


Figure 9 Synoptique complet d'une case d'échecs interactive

4.2. PCB

Pour avoir une répartition homogène de la lumière à la surface de la case il est nécessaire de placer les 4 LEDs à un emplacement précis. Pour cela nous avons divisé la case en 4 carrés nous avons placés chaque LED au centre des carrés. Pour parfaire l'homogénéité de la lumière nous avons fait imprimer deux gabarits de cases de hauteurs différentes.

Le deuxième enjeu était celui des connecteurs permettant de relier chaque case entre elles. Il se devait d'être compact afin de pouvoir être intégrer facilement dans notre case, il devait également posséder un déport afin de pouvoir dépasser du PCB et ainsi se connecter à la case d'à côté.

Evidemment, nous devions aussi laisser place à deux trous de fixations, permettant de pouvoir tenir le PCB bien au fond de notre case d'échecs.

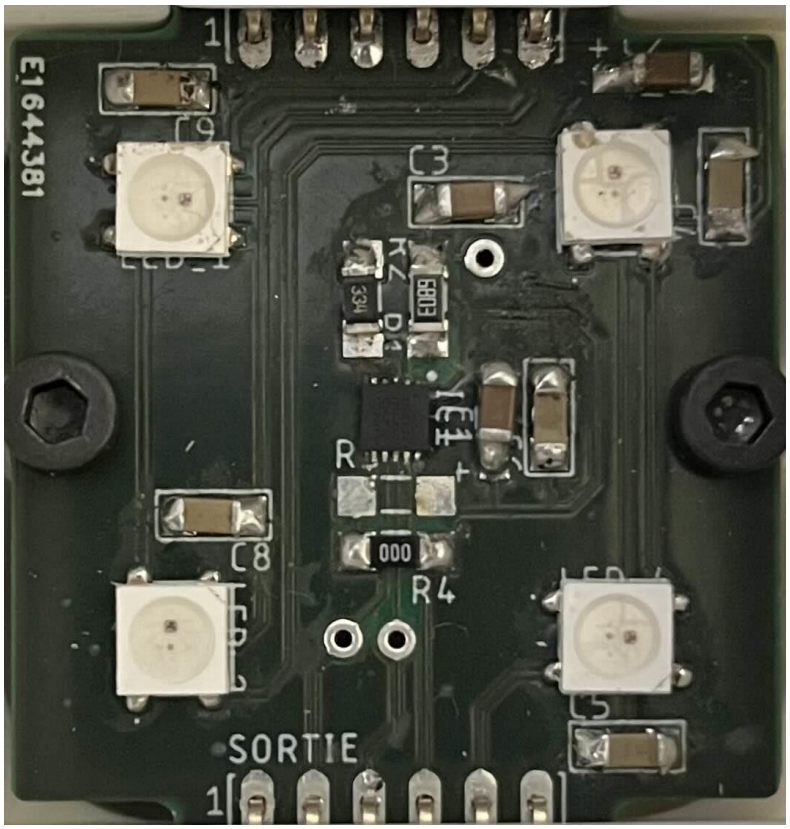


Figure 10 Carte Assemblée vue de dessus

4.3. Classe C++ Case

Case
- Let : int - Num : int - Addr : int - Ordre : int - NbCase : static int + TabLed[4] : int
+ Case (int LetC, int Num, int AddrC, int OrdreC) + afficher (void) : void + getLet (void) : int + getNum (void) : int + getAddr (void) : int + getOrdre (void) : int + get_X (void) : int + get_Y (void) : int + get_Z (void) : int +Request_info (int address, uint8_t reg): uint8_t

Données membres de la classe

L'objectif de la classe Case est de permettre la création de 64 objets ayant des attributs du même type. Les données membres privées *Let* et *Num* sont des entiers représentant respectivement les coordonnées des cases d'un échiquier. Ensuite, on retrouve *Addr* et *Ordre*, deux autres entiers :

- ***Addr*** correspond à l'adresse I2C du capteur à effet Hall.
- ***Ordre*** indique la position de la case dans la chaîne, ce qui permet de sélectionner les LEDs associées.

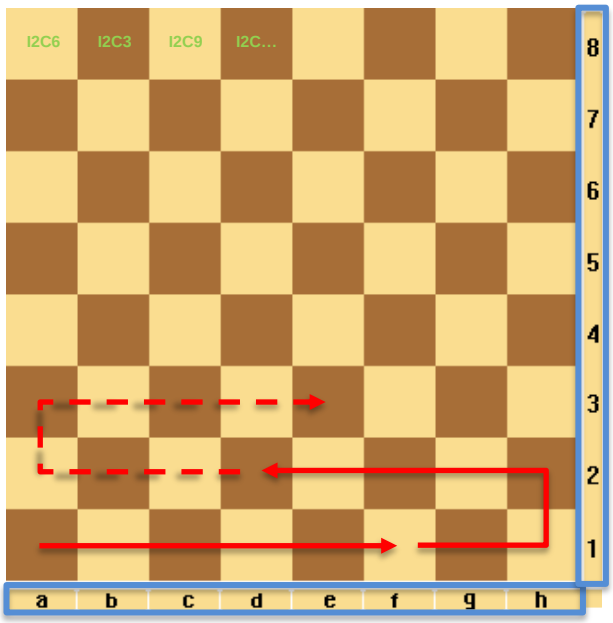


Figure 11 Plateau d'échecs avec ses paramètres associés

Ces quatre variables sont privées. Une fois l'objet instancié, il devient impossible de modifier ces informations. De plus, un entier statique *NbCase* est utilisé pour connaître le nombre total de cases instanciées. Enfin, les couleurs des LEDs sont stockées dans un tableau de quatre entiers au format 0xFFFFFF, où chaque paire de caractères hexadécimaux représente l'une des trois composantes RGB (rouge, vert, bleu).

Fonctionnement du constructeur

Le constructeur de la classe initialise les LEDs de la case en blanc par défaut. Ensuite, il assigne les valeurs passées en paramètres aux variables *Let, Num, Addr* et *Ordre*. Il incrémente également la variable statique *NbCase* de 1. Enfin, un message "Case créée" est affiché sur le terminal série.

Méthodes de la classe

- **Afficher** : Cette méthode affiche sur le terminal série les valeurs des variables *Let, Num, Addr* et *Ordre.*
- **Getters** : Pour accéder aux coordonnées (*Let* et *Num*), à l'adresse I2C (*Addr*) et à l'ordre (*Ordre*), des getters sont mis à disposition : *getLet, getNum, getAddr* et *getOrdre*.
- **get_X, get_Y, get_Z** : Ces méthodes retournent une valeur moyennée sur 30 mesures pour un des axes du champ magnétique détecté par le capteur.
- **Request_info** : Cette méthode permet de concaténer les informations reçues sur deux octets en un mot de 16 bits, incluant un bit de signe.

4.4. Modélisation 3D & Assemblage

Pour parfaire le rendu de notre projet, nous avons décidé d'habiller notre électronique afin de réellement simuler les cases d'échecs, en imaginant qu'avec 64 cases assemblées les unes avec les autres, on obtienne un réel échiquier.

Pour ce faire, nous avons premièrement designer notre case en CAO sur Fusion 360, le premier enjeu était de pouvoir introduire notre PCB à l'intérieur de notre casing tout en pouvant refermer ce dernier. Pour cela nous avons dessiné la boite en deux parties, un « bottom » dans lequel on glissera et fixera le PCB à l'aide de deux inserts* introduits à chaud. Et un « top » qui viendra refermer la boite tel un couvercle. Sur le couvercle on dispose un carré d'acrylique afin d'obtenir une diffusion optimale

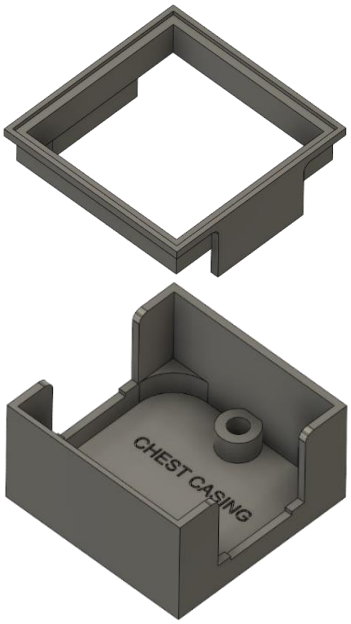


Figure 12 Aperçu du Casing sur Fusion 360

Pour fixer facilement notre carte avec des vis dans notre boite, nous avons choisi d'ajouter des inserts. Pour fixer ces inserts nous nous sommes munis d'un fer à souder avec une panne large. Il suffit alors d'enfiler l'insert sur le fer pour le chauffer et ainsi lui permettre de se placer dans le trou pré-percé prévu. Une fois refroidi l'insert deviens solidaire du reste de la boite.

Figure 13 Insert en Laiton

5. Conclusion et perspectives

5.1. Changement de la méthode d'adressage I²C

Afin de pouvoir réaliser un échiquier complet, soit un plateau composé de 64 cases, l'attribution des adresses I²C ne peut se faire par le biais de jeu de résistance, ces dernières fixant une tension en entrée permettant de sélectionner l'adresse souhaitée. Cette méthode nous limite en effet à un maximum de 16 adresses différentes, cela ne permet donc pas de couvrir nos 64 cases.

C'est ici que l'on peut voir une perspective d'amélioration, afin de pouvoir mener ce projet jusqu'au bout il faudrait donc changer la méthode d'attribution des adresses I²C, en écrivant directement les valeurs des adresses dans l'EEPROM on pourrait ainsi avoir jusqu'à 127 adresses permettant ainsi de couvrir notre échiquier entier.
Cela aurait également l'avantage de pouvoir réunir tout notre plateau d'échecs sur un seul et unique grand PCB, et donc l'utilisation d'un seul µC pour tout le plateau.

5.2. Correction positionnement des Barrettes coudées

En effet, le placement de nos barrettes coudées en entrée à sortie de nos cases n'a pas été fait de manière optimale puisqu'à l'assemblage, le coude de nos barrettes créer un écart entre chaque case qui n'était pas souhaitée.

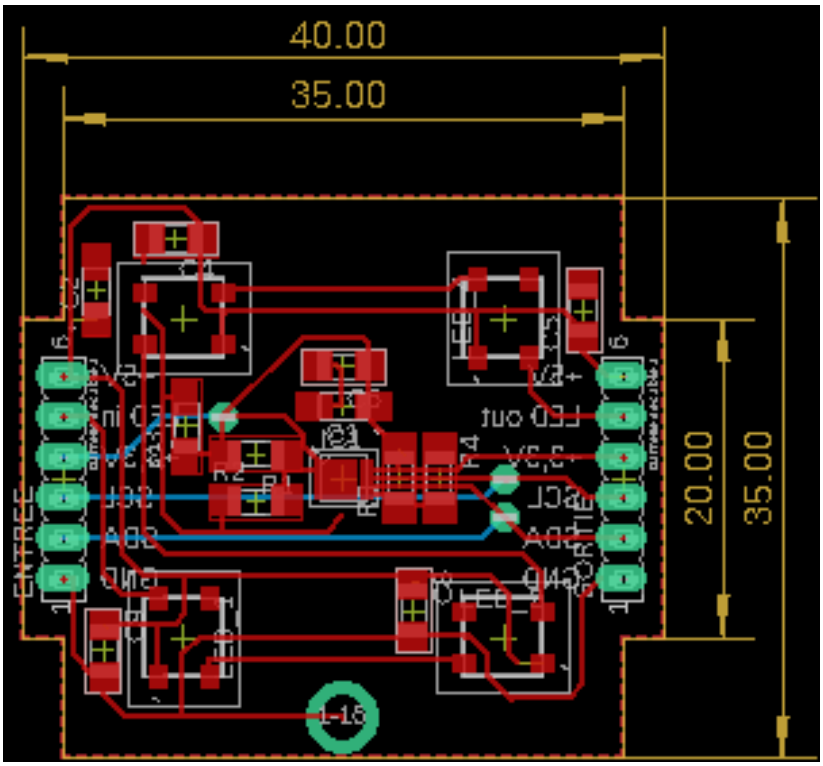


Figure 14 Schéma du routage de la carte utilisé pour la fabrication des cases

Voici donc une proposition de replacement des barrettes coudées, celles étant plus rentrées vers l'intérieur elles permettraient de supprimer l'écart présent entre chaque case actuellement. En revanche, si l'on veut conserver cette taille de PCB et donc de case, ainsi que la diffusion optimale de la lumière émise par nos LEDs, nous devrons dans ce cas placer et souder nos barrettes au dos de la carte afin qu'elles ne rentrent pas en conflit / ne se chevauchent pas avec les LEDs.

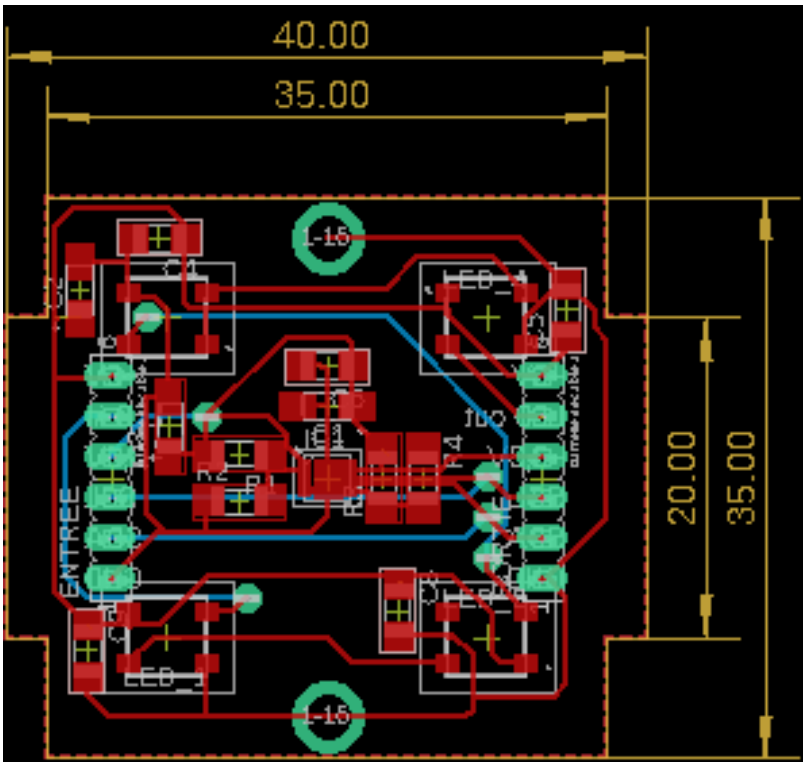


Figure 15 Schéma du routage de la carte retravaillé pour permettre la juxtaposition des cases