Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

Filière Systèmes industriels

Orientation Infotronics

Travail de bachelor
Diplôme 2020

Amand Axel

Smart Targets Trainer System

Professeur
Corre Jérôme

Expert
Furletti Johnny

Date de la remise du rapport
14.08.2020

HES-SO Valais

SYND	ETE	TEVI
X	X	X

Données du travail de diplôme
Daten der Diplomarbeit

FO 1.2.02.07.CB
haj/11/01/2016

Filière / Studiengang **SYND**	Année académique / *Studienjahr* **2019/20**	No TD / *Nr. DA* **it/2020/45**
Mandant / *Auftraggeber* ☐ HES—SO Valais ☒ Industrie ☐ Etablissement partenaire *Partnerinstitution*	Etudiant / *Student* **Axel Amand** Professeur / *Dozent* **Jérôme Corre**	Lieu d'exécution / *Ausführungsort* ☒ HES—SO Valais ☐ Industrie ☐ Etablissement partenaire *Partnerinstitution*
Travail confidentiel / *vertrauliche Arbeit* ☐ oui / ja [1] ☒ non / nein	Expert / *Experte (données complètes)* **Furletti Johnny**	

Titre / *Titel*

Smart Targets Trainer System

Description / *Beschreibung*

Le projet consiste en la conception d'un système intelligent de ciblerie électronique.
Les cibles, de formes et tailles variées, disposées le long d'un parcours, devront réagir selon un scénario choisi au préalable. Une fois le système enclenché, l'opérateur accède à un terminal de contrôle permettant de sélectionner puis de lancer le jeu.
Réagissant aux impacts de toutes directions et de différentes munitions (billes de plastique ou de peinture), capables de transmettre leur état par sons et lumières, ces dernières sont équipées de capteurs de mouvement, permettant d'élaborer des scénarios de jeu séquencés et plus complexes.
Le projet est destiné à deux publics distincts : le marché récréatif/divertissement (p.ex. Airsoft/Paintball); ainsi que le marché professionnel de l'entraînement et de la sécurité (p.ex. police).

Objectifs / *Ziele*

- conception d'un système filaire de communication sur longue distance, avec un grand nombre d'appareils potentiels (env. 30)
- conception du système nomade (par batterie), rechargeable sur secteur ; ainsi qu'utilisable directement sur secteur
- conception électronique d'une carte de contrôle, avec interfaçage homme-machine (écran, clavier, buzzer) ; ainsi que de cartes pour la ciblerie (leds, son, détection de touche)
- conception de boitiers correspondants, de taille minimale, et permettant l'utilisation des capteurs infrarouges
- élaboration d'un protocole de communication entre tous les dispositifs présents dans le système
- sélection et réglages des modes de jeu ; interface simple pour la création de scénarios avancés.

Signature ou visa / *Unterschrift oder Visum*

Responsable de l'orientation / filière
Leiter der Vertiefungsrichtung / Studiengang:

.....

[1] Etudiant / *Student* :

.....

Délais / *Termine*

Attribution du thème / *Ausgabe des Auftrags:*
25.05.2020

Présentation intermédiaire / *Zwischenpräsentation*
Semaine / Woche 26 (22.06 – 26.06.2020)

Remise du rapport / *Abgabe des Schlussberichts:*
14.08.2020, 12:00

Exposition / *Ausstellung der Diplomarbeiten:*
28.08.2020 (si autorisé / *falls genehmigt*)

Défense orale / *Mündliche Verfechtung:*
Semaine / Woche 36 (31.08 – 04.09.2020)

---

[1] *Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.*
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.

Rapport reçu le / *Schlussbericht erhalten am* ..... Visa du secrétariat / *Visum des Sekretariats* .....

STTS – Smart Targets Trainer System

Diplômant/e Amand Axel

Objectif du projet

L'objectif est de concevoir une ciblerie électronique, destinée au monde du divertissement ainsi qu'aux professionnels.
Le système propose divers scénarios, en étant capable de détecter l'impact de projectiles variés et repérer le joueur.

Méthodes | Expériences | Résultats

En première instance, les besoins spécifiques du mandant sont établis.
Le système est séparé en 4 parties, permettant une modularité selon le support utilisé : le gestionnaire du jeu, la cible, ainsi que ses deux modules – lumière-son et détecteur de présence.

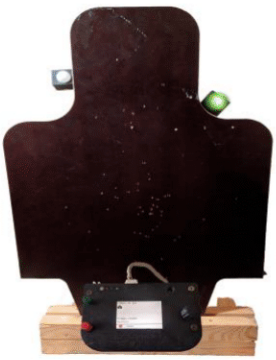
L'électronique est réfléchie pour permettre de travailler de façon nomade (sur batterie), ou fixe (sur secteur). Elle intègre un système de recharge et de protection.
Le tout communique au travers d'un bus CAN 2.0B.
La logique est régie par un microcontrôleur de la gamme dsPIC, d'une part gérant les boutons et l'écran, et d'une autre la détection des impacts au travers d'un accéléromètre.
Pour un futur développement, elle intègre aussi Bluetooth et WiFi.

Chaque module est programmé pour répondre à sa fonction :
- Le contrôleur, régissant le jeu et offrant une interface claire à l'utilisateur
- La cible, réagissant aux impacts et suivant les commandes reçues

Pour clore le concept, une série de boitiers simples ont été développés, pour terminer par des tests chez le mandant.

Après une semaine de mise à l'épreuve, le projet atteint tous ses objectifs et est pleinement fonctionnel !

Ensemble complet
Contrôleur
Cible et ses modules (lumière-son et détecteur de présence)
Bouton extérieur



Cible en jeu
Le contrôleur est branché à une cible, un jeu est en cours (lumière verte).
On aperçoit le capteur à gauche.

Travail de diplôme
| édition 2020 |

Filière
Systèmes industriels

Domaine d'application
Infotronics

Professeur responsable
Corre Jérôme
jerome.corre@hevs.ch

Partenaire
DefcomZone (Furletti Johnny)
Rue de l'Ancienne-Pointe 16
1920 Martigny

Je remercie tout d’abord M. Corre Jérôme pour m’avoir suivi au long de ce projet, tout en sachant m’inculquer bien des astuces quant à ma méthodologie de travail, et proposer des solutions lorsque mon champ de réflexion venait à se restreindre.

Je tiens à remercier M. Furletti Johnny pour avoir proposé ce projet, de m’avoir fait confiance pour le mener à terme, d’avoir répondu présent en m’ouvrant ses portes, ainsi qu’en m’accompagnant pour réaliser les tests nécessaires.

Un grand merci à M. Sartoretti Pascal, m’ayant donné les droits d’utilisation et de modification sur ses librairies de gestion d’écran.

Un dernier merci à M. Rieder Medard, m’ayant autorisé l’exploitation et l’altération de son code de micro-distributeur événementiel.

Sommaire

1. Introduction

1.1. Contexte

DefcomZone, basé à Martigny, propose une salle intérieure de tir sur cibles électroniques.

Figure 1: logo DefcomZone

Les clients peuvent louer la salle, seuls ou à plusieurs, pour se divertir ou, pour les habitués, s'entraîner au maniement de matériel en toute sécurité.

La salle est équipée d'un système de ciblerie permettant d'utiliser des répliques de tir Airsoft - pratique proche du Paintball - et ainsi de travailler avec 5 modes de jeu différents sur 30 cibles.

DefcomZone vise à diversifier son offre pour toucher un public toujours plus grand, désirant offrir des services spécialisés pour les professionnels et leur entraînement.

Elle a commencé par acquérir et équiper différents terrains, permettant des approches nouvelles en élargissant les services disponibles.

Toutefois, la ciblerie actuelle nécessite d'être reliée au secteur, et n'offre que des scénarios de jeu bien définis, immodifiables, ne présentant d'intérêt que pour le marché du divertissement. Aussi, seul le matériel Airsoft est compatible, délaissant les pratiques similaires.

1.2. But

Le projet consiste en l'élaboration d'un système de ciblerie complet, qui possède les caractéristiques spécifiques suivantes :

- Le système doit être nomade, capable de tourner en toute autonomie, pouvant être transporté facilement et loué à un client
- Les cibles doivent pouvoir détecter des impacts provenant de matériel Nerf[1], Airsoft[2], Paintball[3], ou même de SiMunition[4]
- Le système doit offrir un moyen de séquencer un scénario (5 cibles s'allument, une fois toutes touchées -> les 5 cibles suivantes s'allument …)
- Les cibles doivent pouvoir détecter le joueur, permettant au système de réagir en conséquence selon le scénario joué
- Utiliser des connecteurs compatibles avec le câblage actuel (voir *5.5 Connecteurs*)

Toute autre contrainte est laissée au bon vouloir ou acceptation du mandant, notamment :

- La communication, qui doit pouvoir se faire sur une grande distance
- Le format du contrôleur principal, pour gérer les jeux
- L'aspect général du produit / conception des boitiers

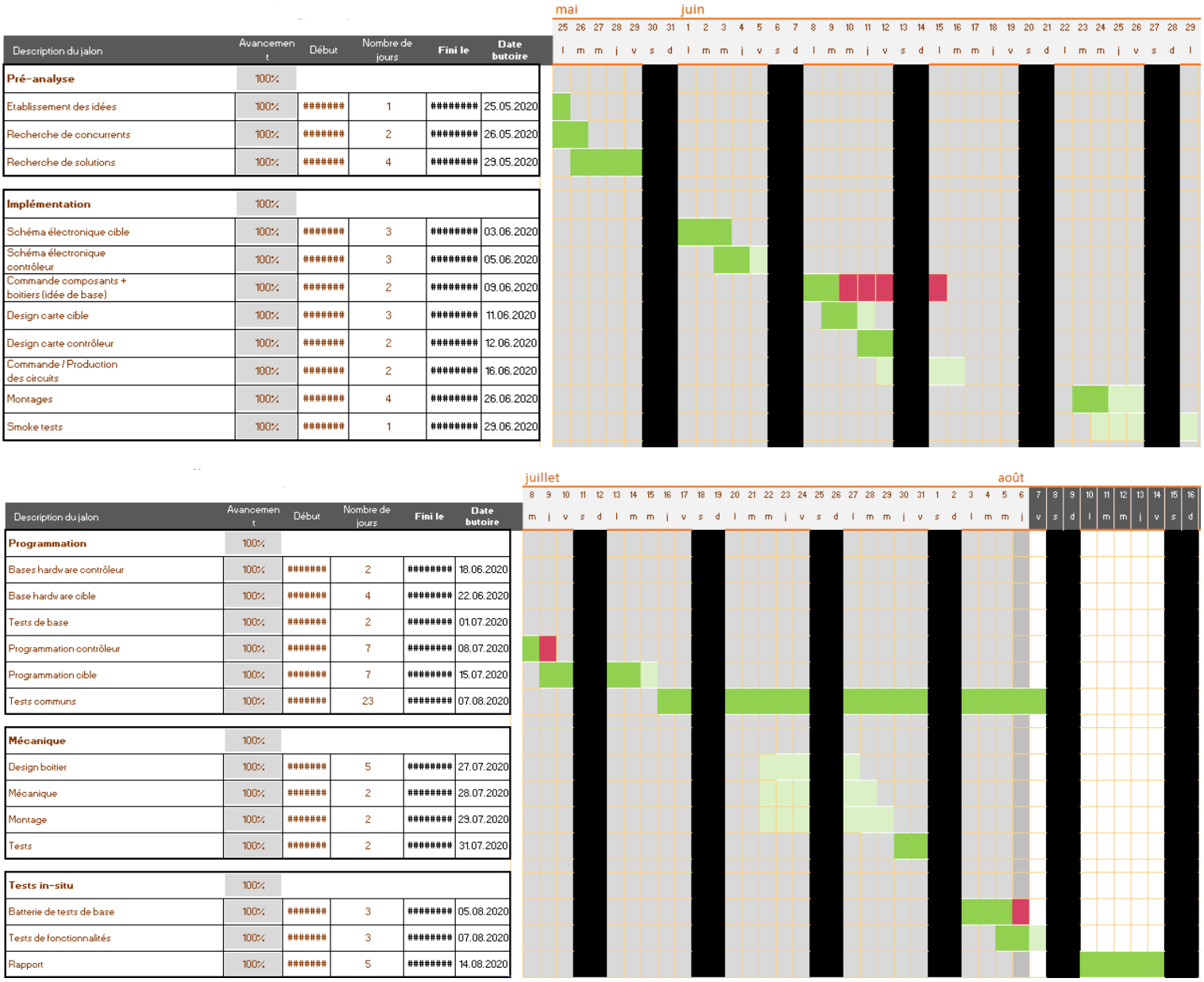
[1] https://fr.wikipedia.org/wiki/Nerf_(marque) – fléchettes en mousses
[2] https://fr.wikipedia.org/wiki/Airsoft - billes dures d'amidon de maïs
[3] https://fr.wikipedia.org/wiki/Paintball - billes de peinture
[4] https://simunition.com/en/products/fx_marking_cartridges - balles d'entraînement à base de peinture

2. Planning de travail

Le fichier complet est disponible dans le dossier projet :

TB Smart Target / 00_Base / 06_Planning

Vert : dans les temps / Vert clair : temps rattrapé sur le temps prévu / Rouge : temps dépassé



Description du jalon	Avancement	Début	Nombre de jours	Fini le	Date butoire
Pré-analyse	100%				
Etablissement des idées	100%	########	1	#########	25.05.2020
Recherche de concurrents	100%	########	2	#########	26.05.2020
Recherche de solutions	100%	########	4	#########	29.05.2020
Implémentation	100%				
Schéma électronique cible	100%	########	3	#########	03.06.2020
Schéma électronique contrôleur	100%	########	3	#########	05.06.2020
Commande composants + boitiers (idée de base)	100%	########	2	#########	09.06.2020
Design carte cible	100%	########	3	#########	11.06.2020
Design carte contrôleur	100%	########	2	#########	12.06.2020
Commande / Production des circuits	100%	########	2	#########	16.06.2020
Montages	100%	########	4	#########	26.06.2020
Smoke tests	100%	########	1	#########	29.06.2020
Programmation	100%				
Bases hardware contrôleur	100%	########	2	#########	18.06.2020
Base hardware cible	100%	########	4	#########	22.06.2020
Tests de base	100%	########	2	#########	01.07.2020
Programmation contrôleur	100%	########	7	#########	08.07.2020
Programmation cible	100%	########	7	#########	15.07.2020
Tests communs	100%	########	23	#########	07.08.2020
Mécanique	100%				
Design boitier	100%	########	5	#########	27.07.2020
Mécanique	100%	########	2	#########	28.07.2020
Montage	100%	########	2	#########	29.07.2020
Tests	100%	########	2	#########	31.07.2020
Tests in-situ	100%				
Batterie de tests de base	100%	########	3	#########	05.08.2020
Tests de fonctionnalités	100%	########	3	#########	07.08.2020
Rapport	100%	########	5	#########	14.08.2020

Figure 2: Planning

Conformément au plan de base, les périodes établies ont été globalement honorées pour permettre la réussite du projet.

Le retard notable porte sur une commande faîte tardivement, bien que le matériel ait été reçu à temps pour coïncider avec le début du montage des circuits électroniques.

L'avance prise sur le montage des circuits ainsi que le design et l'impression des boitiers a été réaffectée à la programmation du système.

3. Analyse concurrentielle

Dans le but d'évaluer le marché existant et ainsi entrevoir les solutions adoptées, des systèmes concurrents sont rassemblés ici.

3.1. Concurrents directs

Par concurrents directs sont désignés les systèmes de ciblerie destinés à l'Airsoft.

TrainShot

Le système TrainShot[1] est un complet de ciblerie sans fil avec les caractéristiques suivantes :

Interface Homme-Machine	Communication entre cibles	Offre de jeu	Nmax de cibles	Prix par cible	Détection impact
Téléphone/tablette	Bluetooth	Scénarios simples + statistiques	8	111 € (env. 118 CHF)	?

Tableau 1: TrainShot - caractéristiques

Avantages :

- Système sans fil
- Tout-en-un
- Statistiques de jeu
- Nomade

Désavantages :

- Nombre de cibles limité
- Portée Bluetooth (amplificateur externe requis)
- Forme et taille des cibles fixes

Figure 3: cible TrainShot, https://www.trainshot.com/wp-content/uploads/2019/11/how-it-works-smart-target-1.png

TrainShot est une entreprise américaine, qui possède une gamme de produits pour professionnels, ainsi qu'une gamme dédiée à l'Airsoft.

Pour cette dernière, les cibles sont fournies avec leur « corps » fait d'acier de 2[mm] d'épaisseur.

Ces dernières se connectent entre-elles par Bluetooth, ainsi qu'à un téléphone pour gérer les jeux.

L'application permet le lancement des jeux, et récupère le moment d'impact de chaque cible afin de créer des statistiques et de définir un score.

Les statistiques peuvent être publiées en ligne, et il est ainsi possible de concurrencer d'autres joueurs du monde entier.

Jusqu'à 6 joueurs simultanés (avec une licence business) peuvent prendre part au jeu.

Les cibles indiquent leur état à l'aide de 4 LEDs RGB, pendant que la zone basse fait office de zone de détection d'impact.

AttackSense

Le système AttackSense [2] est un complet de ciblerie sans fil avec les caractéristiques suivantes :

Interface Homme-Machine	Communication entre cibles	Offre de jeu	Nmax de cibles	Prix par cible	Détection impact
Téléphone/tablette	Wi-Fi	Scénarios simples + statistiques	65	95 £ (env. 114 CHF)	?

Tableau 2: AttackSense - caractéristiques

Avantages :

- Système sans fil
- Tout-en-un
- Statistiques de jeu
- Nomade

Désavantages :

- Forme et taille des cibles fixe
- Trépieds requis

Figure 4: cible AttackSense, https://www.attacksense.com/wp-content/uploads/2019/10/target-tough-700.png

AttackSense est une entreprise anglaise, qui possède une gamme de produits dédiée à l'Airsoft.

Les cibles sont disponibles en deux tailles, fournies avec leur « corps » fait d'aluminium et de PVC et caoutchouc.

Elles doivent être montées verticalement ; au besoin, l'entreprise peut fournir les trépieds adéquats.

Ces dernières se connectent entre-elles par Wi-Fi, ainsi qu'à un téléphone pour gérer les jeux.

L'application permet le lancement des jeux, et récupère le moment d'impact de chaque cible afin de créer des statistiques et de définir un score.

Les statistiques peuvent être fusionnées dans une base de données, permettant à un club de pouvoir suivre et comparer leurs différents membres.

Les cibles indiquent leur état à l'aide de 2 LEDs RGB diffusées par une protection de polycarbonate fumé de 8 [mm] d'épaisseur, pendant que toute la zone fait office de zone d'impact.

M.E.T (G&G)

Le système **M**ultifunctional **E**lectronic **T**arget de chez G&G [3] est un complet de ciblerie filaire avec les caractéristiques suivantes :

Interface Homme-Machine	Communication entre cibles	Offre de jeu	Nmax de cibles	Prix par cible	Détection impact
Bouton poussoir	Sériel 3 fils	Scénarios basiques, minuterie externe	30	110 CHF, sans matériel externe (bouton, minuterie …)	Système de succion et barrière IR

Tableau 3: M.E.T - caractéristiques

Avantages :

- Fait pour la compétition
- Sur secteur, pas de batteries
- Pas d'appareil extérieur nécessaire – gestion avec un simple interrupteur
- Cible « 3D »

Désavantages :

- Câblage
- Forme et taille des cibles fixes
- Entretien du système de détection d'impact nécessaire *(voir **4.2** **Analyse de concurrent**)*

Figure 5: cibles G&G, https://www.ggtdu.com/wp-content/uploads/2016/02/met-unit.png

G&G est une entreprise taïwanaise, fabricant de répliques et matériel Airsoft, dont un système de ciblerie.

Les cibles sont fournies avec leur « corps » fait de caoutchouc teinté vert et d'ABS noir.

Elles peuvent être montées verticalement ou posées à plat.

Ces dernières se connectent entre-elles par câble, avec un système de communication sur 3 fils, ainsi qu'à un interrupteur et à un afficheur 7 segments externe.

Les jeux sont lancés simplement par plusieurs appuis sur l'interrupteur, correspondant au mode de jeu voulu. Ce dernier est ensuite démarré avec un appui simple.

Le temps démarre en même temps que le jeu, pour s'arrêter automatiquement à la fin de ce dernier.

Les cibles indiquent leur état à l'aide de 4 LEDs rouge/bleu diffusées par le caoutchouc teinté vert, faisant office de zone d'impact 3D.

Le système est utilisé dans une compétition internationale dirigée par cette même entreprise.

*Le système, fourni par le mandant, a pu subir une ingénierie inverse, détaillée en **4.2** **Analyse de concurrent** ; y présentant notamment le système de détection de tir.*

3.2. Concurrents proches

Par concurrents proches sont désignés les systèmes de ciblerie destinés à d'autres domaines.

SteelAlive

Le système SteelAlive [4] est un complet de ciblerie sans fil avec les caractéristiques suivantes :

Interface Homme-Machine	Communication entre cibles	Offre de jeu	Nmax de cibles	Prix par cible	Détection impact
Téléphone/tablette	Wifi	Scénarios simples + statistiques	10	140 € (env. 148 CHF)	Piezo *(sur prototype)*

Tableau 4: SteelAlive - caractéristiques

Avantages :

- Système sans fil
- Adaptable sur tout type de cible
- Statistiques de jeu
- Nomade
- LEDs déportables

Désavantages :

- Boitier relais Wi-Fi nécessaire
- Nombre de cibles limité

Figure 6: cible SteelAlive avec module LED, https://steelalive.io/wp-content/uploads/2019/07/led.png / https://steelalive.io/wp-content/uploads/2019/07/smart_target_device.png

SteelAlive est une entreprise polonaise, qui possède une gamme de produits capable de fonctionner avec de la munition de tir.

Les cibles sont fournies sans corps : le système se fixe à l'arrière d'une plaque d'acier AR500 *(acier fait pour le tir)*, et y sont branchés jusqu'à 3 modules LEDs, placés au bon vouloir de l'utilisateur.

Ces dernières se connectent entre-elles au travers d'une passerelle Wi-Fi, fournie dans le kit, ainsi qu'à un téléphone pour gérer les jeux.

L'application permet le lancement des jeux, et établit une liste de scores.

Les scores peuvent être comparés en ligne auprès d'autres utilisateurs du système.

Les cibles indiquent leur état à l'aide des modules LEDs annexes.

Repéré sur l'un des prototypes présentés, un piezo est visible, fixé à l'arrière de la cible. Cela laisse à penser que le pic de tension généré permet la détection de l'impact.

Train2Shoot

Le système Train2Shoot [5] est un complet de ciblerie sans fil avec les caractéristiques suivantes :

Interface Homme-Machine	Communication entre cibles	Offre de jeu	Nmax de cibles	Prix par cible	Détection impact
Téléphone/tablette	Bluetooth	Scénarios simples + statistiques	1, avec 5 zones	?	Couche résistive variable sur carton

Tableau 5: Train2Shoot - caractéristiques

Avantages :

- Système sans fil
- Statistiques de jeu
- Multizones
- Système radio longue portée (en option)

Désavantages :

- Nombre de cibles limité
- Environnement spécifique
- Indications sur téléphone

Figure 7: cible Train2Shoot, http://www.train2shoot.com/assets/Trainshot-shooting-range-kit_real-gun-electronic-shooting-system_how-it-works-1.jpg

Train2Shoot est une entreprise américaine, qui possède une gamme de produits capable de fonctionner avec de la munition de tir, destiné principalement au tir en stand.

Les cibles sont formées d'un module sur lequel une feuille cartonnée vient se clipper. Y sont dessinées 5 zones de tir différentes. Cette dernière est interchangeable.

Le système se connecte à un téléphone, qui gère les scénarios.

L'application permet le lancement des jeux, et établit une liste de scores.

Les cibles n'ont pas de moyen d'indiquer leur état. Le téléphone doit donc être proche du tireur afin de savoir quelle est la prochaine zone à toucher.

4. Electronique – Analyse des solutions

4.1. Caractéristiques recherchées

Les différents éléments du système devant être décidés sont :

- **Système nomade** : comment faire marcher le système avec et sans accès au 230 [V]
- **Bus de communication** : comment faire communiquer les différentes cibles
- **Détection du tir** : comment reconnaître un impact de projectile d'une autre perturbation (p.ex. vent)
- **Indicateurs sensoriels** : comment faire connaître l'état du système à l'utilisateur
- **Détecteur de présence** : comment détecter et réagir à la présence d'un joueur
- **Format du contrôleur de jeu** : comment régler le système, le démarrer …

Le système se compose de deux circuits majeurs : un **contrôleur**, sur lequel les modes de jeu sont réglés ; ainsi que plusieurs **cibles**. A cette dernière seront joints deux modules (leds, son / détection de présence).

La première étape consiste en l'analyse d'un système concurrent, suivie de la sélection des solutions pour les points mentionnés précédemment.

4.2. Analyse de concurrent

Le mandant a fourni une cible du système M.E.T [3].

Cette dernière a été décortiquée et analysée pour en tirer les principes essentiels appliqués.

Fonctionnement du système

Les cibles se présentent sous la forme suivante :

Figure 8: système M.E.T,
https://www.guay2.com/en/product/detail-1306/

Y est branché un interrupteur, avec lequel l'utilisateur peut charger un mode de jeu (par multiples appuis successifs).

Le temps de jeu est donné sur un afficheur 7 segments externe.

Les cibles sont capables de détecter des tirs à 180 [°], avec une membrane caoutchouc bombée *(principe présenté plus bas)*.

Ces dernières sont équipées chacune d'un haut-parleur, avec divers sons réglables.

Des LEDs rouges et bleues présentent l'état de la cible (bleu – actif, rouge – touché).

Les cibles sont câblées l'une à l'autre, à la suite.

Ingénierie inverse

Le système électronique a été redessiné *(voir annexe 10.1 Ingénierie inverse – système G&G).*

De ce dernier sortent les principes suivants :

- L'**alimentation** est **auto-gérée** : pour permettre une consommation moindre lorsque le système n'est pas employé, la plupart des sous-systèmes sont désactivés.
- Le **système de communication** entre les cibles est un système **sériel sur 3 fils**, dont la topologie du réseau est de type bus :

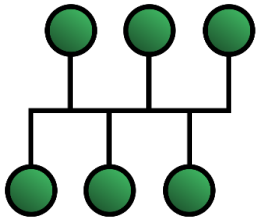


Figure 9: Net Topology, By Myself - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=3583085

  Le protocole n'étant pas connu, il est toutefois possible de déterminer, par le câblage, que les entrées sont de type open-drain.
  Pour les signaux eux-mêmes, une hypothèse est telle que l'un des signaux définit l'utilisation du bus, un second l'horloge et un troisième la donnée.
- Un **processeur audio**, suivi d'un amplificateur classe D et d'un haut-parleur, permettent la lecture de **plusieurs sons**, sélectionnables par un DIP switch.
- Un **capteur d'inclinaison** permet de sélectionner quel détecteur d'impact utiliser (deux capteurs orthogonaux l'un à l'autre, pour utiliser la cible verticalement ou posée à plat).

Détecteur d'impact

Le système de détection de tir se base sur une bille très légère, une barrière infrarouge et un système de succion, qui peut être représenté ainsi :

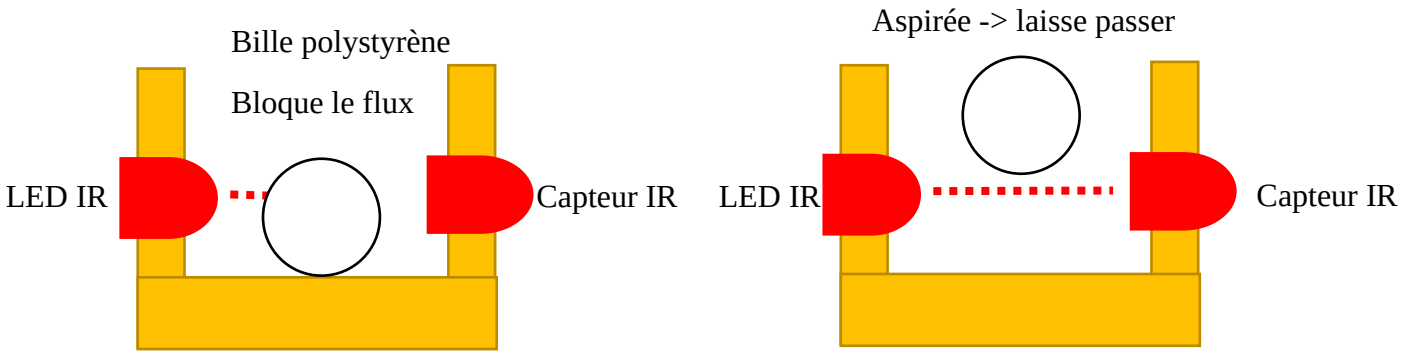


Figure 10: fonctionnement du capteur G&G M.E.T

Bien que sensible et permettant à la cible d'avoir une zone de touche « 3D », ce dernier présente deux désavantages :

- La mécanique est très spécifique
- La bille est sujette à l'électricité statique, requérant un entretien régulier (sans quoi la détection devient bancale)

Cette section regroupe les éléments électroniques communs aux cibles et au contrôleur.

4.3. Système nomade

Considérations

Dans le but d'offrir un transport et une utilisation simplifié lors d'une délocalisation temporaire du système (location, changement de lieu d'utilisation), ce dernier doit être capable de travailler avec ou sans branchement sur le secteur.

De fait, chaque module doit être équipé d'une batterie, être rechargeable, et être capable de travailler directement sans batterie lorsque branché sur le 230 [V].

Une liste du courant potentiel est établie (fait état de toutes les considérations et choix pris dans la suite de ce chapitre) :

Cible

Elément	Courant potentiel max.
uC	5 mA
LEDs	200 mA
Communication	70 mA
Son	100 mA
Capteurs	10 + 1 mA
Total	**386 mA @ 3.3V, eq. 255 [mA] @ 5V w. η = 1**

Tableau 6: Courants potentiels d'une cible

Contrôleur

Elément	Courant potentiel max.
uC	5 mA
LEDs	20 mA
Communication	70 + 20 mA
Son	100 mA
Ecran	15 + 100 mA
Total	**330 mA @ 3.3V, eq. 218 [mA] @ 5V w. η = 1**

Tableau 7: Courants potentiels du contrôleur

Le système peut être représenté ainsi :

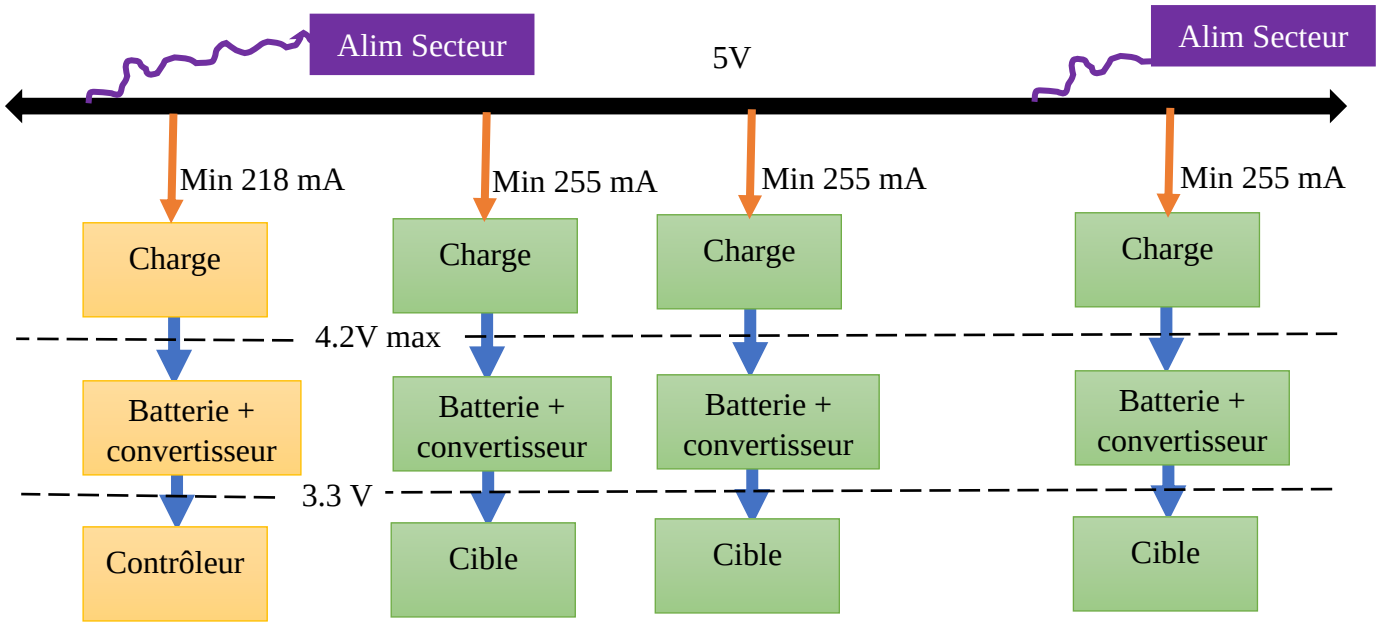


Figure 11: Alimentation du système

Solutions

S'offrent deux possibilités :

- Une seule alimentation robuste sur la ligne, capable d'alimenter le système entier
- Plusieurs alimentations réparties sur la ligne, de type chargeur USB (5V, pour appareil électronique générique)

Critère	1 alim. sur mesure	Plusieurs alims. « USB »
Conception	Dimensionnement sur mesure	Blocs d'alimentation « USB » du commerce
Entretien / Remplacement	Changement du bloc complet par un second sur mesure	Changement du bloc défectueux par un équivalent
Encombrement (prises secteur)	1 prise	Multiples prises le long du parcours
Ajout de cibles	Redimensionnement du bloc	Ajout de blocs selon courant nécessaire
Sur la longueur	Un bloc mis au centre => pertes le long de la ligne	Tout au long de la ligne => moins de pertes

Tableau 8: Comparatif de l'alimentation du système sur secteur

Les deux propositions sont aisément remplaçables l'une par l'autre pour réaliser des tests.
Le système est donc pensé pour accueillir plusieurs blocs d'alimentation le long de la ligne, **et sera d'ailleurs testé ainsi**.
En reprenant le projet plus loin, il est tout à fait envisageable de concevoir un bloc d'alimentation dédié et de ne le brancher qu'en début de ligne, tout en comparant les résultats avec la méthode précédente, notamment pour l'utilisation dans un périmètre restreint.

Choix

Le système sera équipé, le long de la ligne, de **plusieurs blocs d'alimentation type « chargeurs USB »**, leur nombre étant variable suivant la quantité de cibles utilisées.
Les batteries seront ainsi chargées, et la tension adaptée ensuite pour alimenter le circuit.

Dangers

Le fait de travailler sur batterie, ici de type Lithium-Ion, implique de réaliser une charge adéquate, ainsi que d'éviter toute surcharge ou sous-charge qui pourraient entraîner une destruction définitive de la batterie, voire une combustion ou explosion de cette dernière.
Des circuits adaptés doivent donc être mis en place.

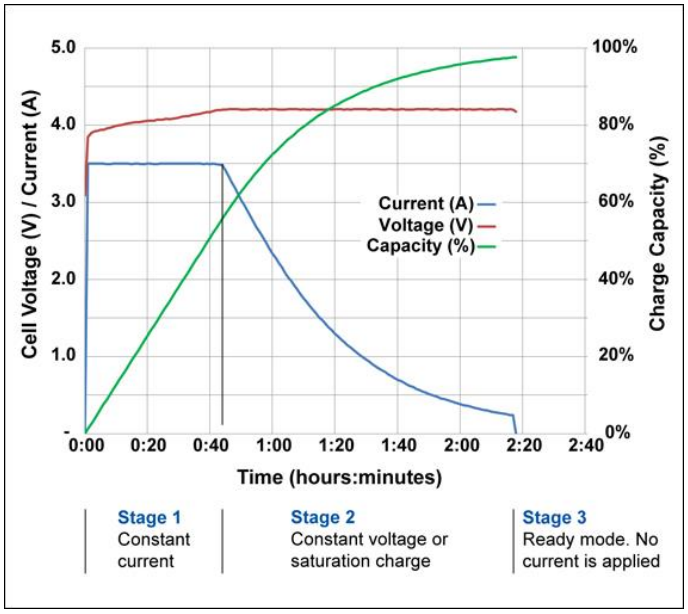


Figure 12: courbe de charge Li-Ion, https://batteryuniversity.com/learn/article/how_to_charge_li_ion_with_a_parasitic_load

4.4. Bus de communication

Considérations

Pour gérer la transmission des données entre les cibles et le contrôleur, un moyen de communication doit être sélectionné.
Il doit permettre à plusieurs nœuds de discuter, au minimum 30.
La distance de communication doit être maximale, pour pouvoir disposer les cibles à bon écart les unes des autres.
Tous les nœuds doivent pouvoir transmettre des données, dès lors que nécessaire.

Solutions

Nom	Sortie	Portée	Nœuds max.	Multi. Comm.	Collisions	Autre
LIN	Level shift	40 m	16	Maître - esclave	n.a.	Filaire
Ethernet	Push-pull	100 m	254	v	Géré	Filaire
RS485	Différentielle	1.2 km @ 90 kbps	32	Possible	Non géré	Filaire
CAN	Différentielle	1 km @ 10 kbps	100 +	v	Géré	Filaire
SPI / Microwire	Push-pull	3 m	Illimité (daisy chain)	Maître-esclave	n.a.	Filaire
I2C	Open-drain	1 m	121 / 1017	Possible	Non géré	Filaire
RS232	Push-pull	100 m @ 2.4 kbps	1 / Multi	Token Ring	-	Filaire
RS422	Différentielle	1.2 km @ 100 kbps	10	Possible	Non géré	Filaire
UNI/O	Push-pull	Sur carte	127 / 1017	Maître - esclave	n.a.	Filaire
MicroLAN (1-Wire)	Push-pull	100 m	20	Maître - esclave	n.a.	Filaire
SDI-12	Push-pull	65 m	10 / 62	Maître - esclave	n.a.	Filaire
Bluetooth	2.4 GHz – 2.4835 GHz	100 m	Illimité (mesh)	v	Géré	Sans-fil, pertes au travers des murs
Wi-Fi	2.412 GHz – 2.484 GHz / 5.15 GHz – 5.835 GHz	300 m	Illimité (mesh)	v	Géré	Sans-fil, pertes au travers des murs

Tableau 9: Comparatifs des bus sériels pour la communication

Les bus sans-fil ont été écartés pour cause de complexité de mise en œuvre dans une topologie (mesh) offrant les caractéristiques requises par l'application.

Aussi, les systèmes par ondes présentent des pertes de portée lorsqu'utilisés au travers d'obstacles. Connaissant le cadre d'utilisation du projet (terrains changeants, dont bâtiments avec plusieurs salles), ces derniers présentent des risques.

Pour pouvoir malgré tout tester leur hypothétique utilisation, la première version du système offre une communication par câbles, ainsi qu'un module Wi-Fi (non-peuplé) pour réaliser des tests futurs sans avoir à créer de nouveau circuit électronique spécifique.

Options

En ressortent deux bus : **CAN** et **RS485**, offrant tous deux une sortie par paire différentielle, permettant la communication sur longue distance avec une plus grande immunité au bruit.

Le bus **RS485** semble plus apte à transmettre une information sur longue distance, pouvant atteindre une distance légèrement supérieure au CAN avec une vitesse 9* supérieure.

Le bus **CAN** offre quant à lui une gestion automatisée des collisions ainsi que le renvoi de messages non lus, avec un filtrage automatique des messages selon leur adresse.

Le désavantage du bus **RS485** est de devoir développer une couche pour traiter tous les messages (plus de temps pris au processeur), ainsi que de gérer l'envoi et le renvoi tout en évitant les erreurs (principes « écouter avant de parler » et « écouter en parlant » pour éviter les collisions) ; tandis que le CAN possède un module dédié.

Choix

Avec les considérations précédentes, le bus CAN répond mieux au besoin du fait de sa plus grande facilité d'implémentation.

Le bus de communication entre le contrôleur et les cibles sera le **bus CAN**, ce dernier protégé au mieux contre les surtensions/décharges électro-magnétiques dues au maniement des câbles.

Cette section regroupe les éléments électroniques propres aux cibles.

4.5. Détection du tir

Considérations

Pour les cibles, il est essentiel d'être capable de détecter des impacts d'intensité variable, tout en ne réagissant pas à des facteurs extérieurs (vent, chocs n'étant pas dirigés sur la cible …).

La force à l'impact est d'au moins[5] 200 [FPS] – *pieds par seconde* -, correspondant à **61 [MPS]** – *mètres par seconde* -.
Les projectiles les plus légers rencontrés dans la pratique de l'Airsoft pèsent[6] **0.12 [gr]**.

On peut donc appliquer l'équation suivante qui représente l'énergie cinétique à l'impact :

$$F_{cin} = \frac{1}{2} * m_{projectile} * v^2_{projectile} = \frac{1}{2} * 0.00012 * 61^2 = \mathbf{0.223}\ [\boldsymbol{J}]$$

Équation 1: Force du projectile à l'impact

Les données utilisables s'arrêtent malheureusement ici. Il est impossible de calculer la force de cette énergie lors de l'impact, même en supposant qu'elle soit totalement transmise à la cible. Il faudrait pour cela connaître la durée de l'impact.
Seule la manière empirique (mesure avec un appareil dédié) peut faire foi dans cette situation.

Solutions

Il est donc question ici de détecter l'impact sur une surface :

Capteur	Principe	Plage de détection	Sensibilité aux facteurs externes	Avantages	Inconvénients
Accéléromètre	Accélération due au choc	Très sensible	Pas de perturbation	Direction de l'impact, sensibilité réglable	Communication avec chip dédié (2 fils)
Grille IR	Grille d'émetteur/récepteurs IR, le projectile coupe une ligne en impactant la cible	Limité par la vitesse de transition des capteurs IR	Pas de distinction entre les corps solides (feuille …)	Position du choc	Grille propre à la forme de la cible, résolution nécessaire importante
Micro	Le choc crée du bruit qui se transmet	Très sensible	Grande	Pas de contact nécessaire	Trop perturbable, circuit de filtrage
Membrane piezo	Le choc entraîne une vibration du piezo résultant en un pic de tension	Sensible	Pas de perturbation	Bas prix	Circuit de mise en forme requis

Tableau 10 : Solutions pour la détection de l'impact

[5] Valeur basse, avec marge, du matériel rencontré en Airsoft ; le standard se portant sur **300 [FPS] et plus**
[6] A nouveau une valeur basse ; le standard se portant sur **0.2 [gr] et plus**

Tests des possibilités

Le choix se fait entre un capteur piezo ou un accéléromètre.

Piezo

Le capteur piezo peut être testé facilement :

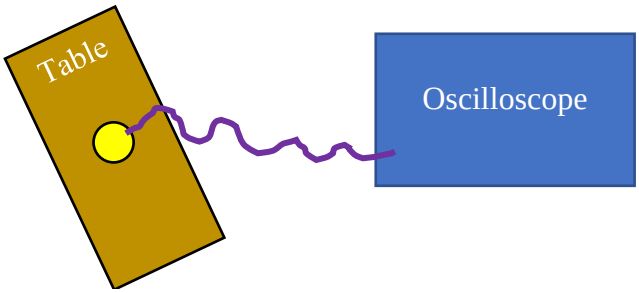


Figure 13: Schéma test pour piezo

Attention, le piezo peut générer de hautes tensions.

Le protocole est tel que :

- Un piezo nu, de 12 [mm] de rayon, est fixé sur une table en bois
- Ce dernier est relié à une sonde
- La zone du piezo est impactée, plus ou moins fort, et le signal capturé

Piezo employé :

Figure 14: Piezo nu, https://www.amazon.co.uk/Spiratronics-Uncased-Piezo-Transducer/dp/B00940V1EG

La mesure n'**a rien de précis**.
Elle est là à pur titre indicatif, pour se rendre compte de la sensibilité du système.
On peut vois plusieurs courbes de différents chocs sur la figure suivante :

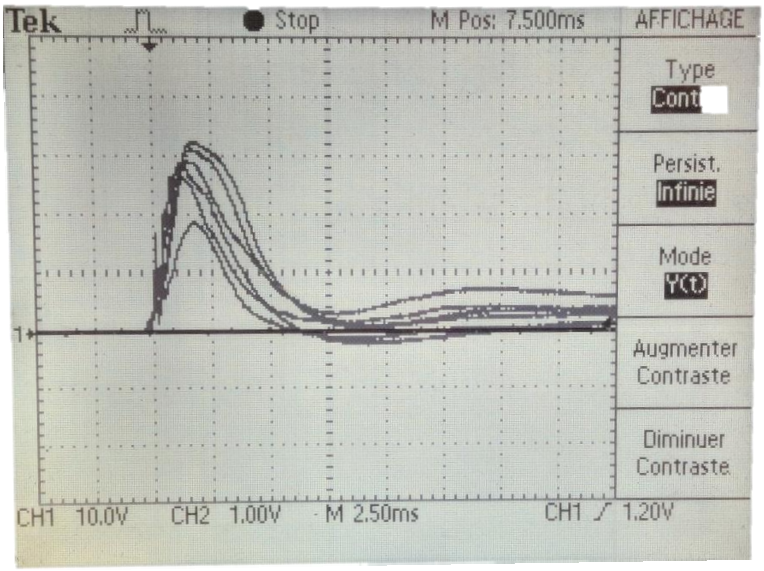


Figure 15: courbes de réponse du piezo

Accéléromètre

L'accéléromètre ne peut être testé sans l'embarquer sur un système spécifique.

Toutefois, il est possible de se rendre compte d'une sensibilité en utilisant un smartphone et analysant ses capteurs :

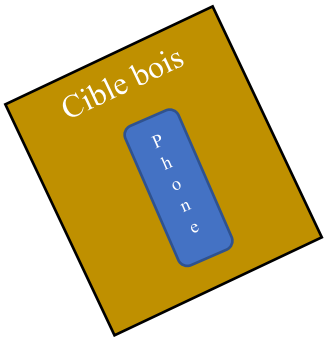


Figure 16: Schéma test pour accéléromètre

Le protocole est tel que :

- Le téléphone est fixé, sans coque, sur une planche de contreplaqué d'1 [cm] d'épaisseur
- L'application SensorLab pour Android est lancée, les valeurs de l'accéléromètre enregistrées au format .csv
- Plusieurs tirs sont réalisés sur la cible avec du matériel standard (projectile mesuré à 280 [FPS])
- Le signal est mis en forme sous Excel

On obtient le graphique suivant :

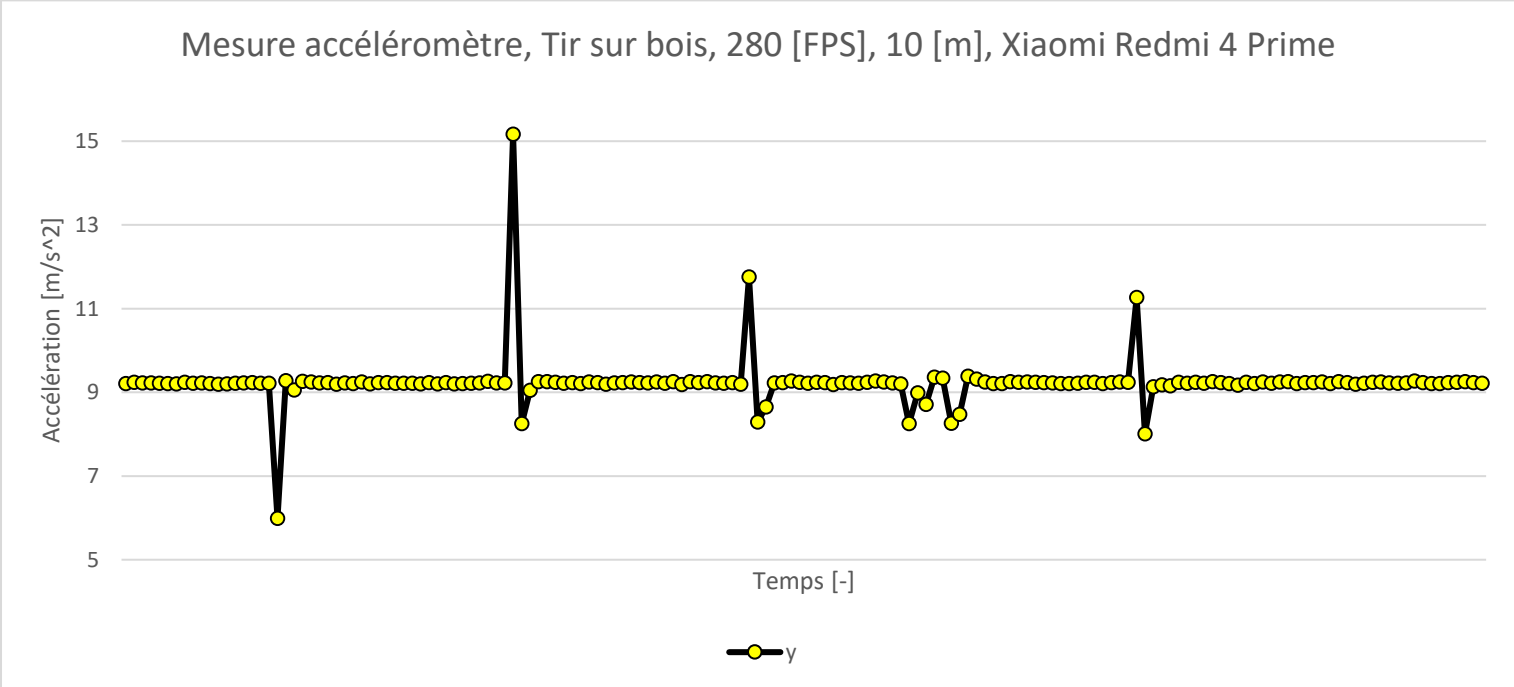


Figure 17: Accélération dus aux tirs

L'axe du temps n'est là qu'à pur titre indicatif et ne présente donc pas de valeurs.

Pour réaliser une correspondance en g des tapotements, on donne le quotient de la valeur mesurée par rapport à g standard, valant g = 9,80665 [m/s^2] [6].

Les pointes représentent un delta d'environ 2 [m/s^2] pour les moins importantes, correspondant à un delta de 0.2 [g]

Des accéléromètres bas-coût, comme le **MMA8652FC** [7], offrent une résolution de 0.00098 g, largement suffisante pour effectuer la détection des chocs, tout en offrant la possibilité d'utiliser des projectiles moins puissants, et ainsi augmenter la cible de clients potentiels (utilisation des Nerfs).

Choix

Le piezo, au travers des tests, s'est révélé largement capable de remplir son rôle.
Il faut toutefois noter qu'un circuit de mise en forme, permettant de créer un flanc d'interruption lors d'un choc (et ainsi libérer le convertisseur A/D) doit être mis en place, permettant ainsi de minimiser l'impact sur la consommation.
De même, le seuil doit pouvoir être réglable (comparateur avec DAC, ou potentiomètre numérique).

L'accéléromètre à quant à lui l'avantage d'offrir des interruptions, permettant de détecter le choc en plus de pouvoir détecter lorsque la cible tombe, tout en consommant très peu (moins de 200 [uA]), et ne requiert pas de circuit extérieur.
Le seuil peut être réglé purement par logiciel, et la sensibilité nécessaire pour détecter les impacts semble convenir avec des modèles à bas-coût.

L'accéléromètre est sélectionné.

4.6. Indicateurs sensoriels

Considérations

Les cibles doivent pouvoir indiquer leur état au travers d'une indication visuelle et sonore.

Pour le visuel, la cible doit pouvoir s'éclairer de multiples couleurs selon leur fonction actuelle (touchée, en attente …). La lumière doit pouvoir se voir, même en plein jour.

Pour le son, la cible doit pouvoir émettre un bruit pour indiquer son emplacement (lors de certains modes de jeux, comme lorsque les cibles s'allument aléatoirement).

Choix - LEDs

Concernant les LEDs, la solution la plus souple est d'utiliser des LEDs RGB. Ces dernières sont placées sur un circuit à part, permettant de positionner la lumière à l'endroit voulu sur la cible.
Un système mécanique à base de matériel focalisant permettra d'améliorer le mélange des couleurs, en même temps que diffuser et concentrer la lumière sur un même point :

Il est possible d'utiliser des LEDs de moins grande puissance, et donc d'économiser de l'énergie.
De plus, la LED est protégée contre un impact potentiel.

Figure 18: Focaliseur pour LEDs,
https://www.aliexpress.com/item/32905704306.html?spm=a2g0s.9042311.0.0.27424c4drII9Ps

Solutions – son

Trois solutions potentielles peuvent être envisagées :

Moyen	Avantages	Inconvénients
Buzzer (DC)	Simple, peu coûteux	1 seul son
Buzzer (PWM/UART)	Simple, sons (basiques) différents	Plus de ressources logicielles que le buzzer (dc)
Ampli. + haut-parleur	Sons agréables	Nécessite un processeur/chip dédié pour l'audio enregistré, un amplificateur ainsi qu'un haut-parleur

Tableau 11: Solutions pour le son

À la suite d'une discussion avec le mandant, il n'y a pas de gain réel à utiliser de « vrais sons » pré-enregistrés comparé à la hausse du prix.
La solution ampli + haut-parleur est donc écartée.

L'implémentation d'un buzzer sur PWM/UART est peu gourmande en ressource et permet une variation possible du son.

Choix - son

Le **buzzer sur PWM/UART** est donc choisi comme indicateur sonore.

4.7. Détecteur de présence

Considérations

Le mandant a demandé à ajouter la possibilité de détecter un joueur (lors de son entrée dans une pièce par exemple) pour permettre une immersion plus poussée lors d'entraînements.

Il est défini que la distance de détection se doit d'approcher les 5 [m].

Solutions

Moyen	Avantages	Inconvénients
Caméra	Analyse du nombre de personnes, longue distance	Coûteux, positionnement correct nécessaire, forte puissance de calcul
Son	Longue distance, directionnel (avec micro spécifique)	Peut détecter n'importe quoi
Vibrations	-	Peut détecter n'importe quoi, cibles doivent être au sol, pas directionnel
Capteur IR	Directionnel, simple et peu coûteux	Doit être modifié pour tourner en 3.3 [V]

Tableau 12: Solutions pour la détection de mouvement

Le capteur IR, traditionnellement un capteur PIR (Pyroelectric InfraRed sensor), permet la détection d'un changement dans le spectre IR (c.à.d. détecte la chaleur humaine).

La longueur de détection varie de 2 à 12 mètres. Des circuits tout-en-un, contenant un système de mise en forme, peuvent être achetés directement à bas prix.

Choix

Un **capteur PIR** est choisi, permettant de détecter facilement un mouvement.

De la même manière que pour les LEDs, ce dernier est monté sur un circuit externe pour permettre de le placer selon la forme de la cible, et branché selon le besoin du client.

Cette section regroupe les éléments électroniques propres au contrôleur.

4.8. Format du contrôleur de jeu

Considérations

Pour communiquer et gérer les cibles, il pourrait être choisi d'intégrer les scénarios dans chaque cible, puis d'en choisir une gérant l'ensemble, au même titre que le système G&G [3].

Toutefois, le mandant a demandé à pouvoir créer des scénarios séquentiels (les cibles 1-2-3-4 s'allument, une fois touchées les cibles 5-6-7-8 s'allument …). De fait, il est nécessaire de fournir une interface à l'utilisateur pour pouvoir faire ces réglages.

Viennent ensuite trois axes de travail :

- Gérer le système par Bluetooth/Wi-Fi sur son téléphone/ordinateur portable
- Gérer le système par connexion USB sur un ordinateur portable
- Contrôleur indépendant offrant une interface propre

Le mandant désirant pouvoir transporter le matériel pour le délocaliser, ainsi que de le louer, il est compliqué de pouvoir offrir une application fonctionnant sous les différents systèmes d'opérations mobiles. De même que pour le PC, qui coupe la portabilité du système.

Il est décidé qu'un boitier indépendant permettra la gestion des jeux.
Ainsi, l'interface homme-machine doit être décidée.

Solutions

Trois solutions sont présentées au mandant :

Clavier + écran

La première proposition consiste en l'utilisation d'un clavier matriciel, couplé à un écran pour se diriger dans les menus et modifier les jeux.

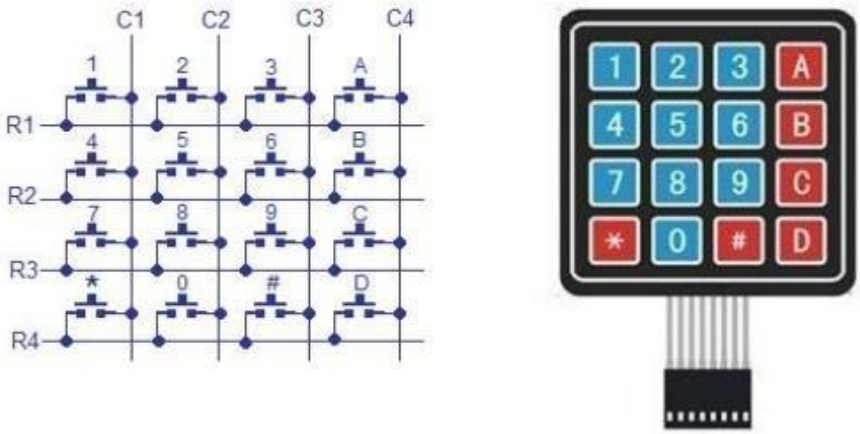


Figure 19: Clavier matriciel, https://www.researchgate.net/figure/Clavier-matriciel-44-et-son-schema-de-principe-b-Detection-des-touches-Il-faut_fig25_288327838

L'avantage ici est de pouvoir taper directement des chiffres.

Le désavantage est le peu d'intérêt d'avoir autant de touches à disposition, qui ne feront que se perdre l'utilisateur ; ainsi qu'à la place nécessaire pour ce dernier sur le boitier.

Ecran tactile

La seconde proposition consiste en l'emploi d'un écran tactile en guise d'interface.

L'avantage est de réduire la place prise et de rendre le système plus interactif (boutons, sliders, clavier virtuel …).

Le désavantage, soulevé par le mandant, porte sur le fait que les utilisateurs sont souvent enclins à porter des gants de protections épais. Il serait difficile de bien l'utiliser.
Fournir un stylet permettrait de contrebalancer le problème.

Ecran + 2 boutons + codeur rotatif

La dernière proposition est d'employer un écran, l'utilisateur pouvant se diriger dans les menus à l'aide d'une molette (codeur rotatif). Deux boutons, « Accepter » et « Annuler », permettent de valider une action ou de sélectionner un sous-menu.

Autres éléments du contrôleur

Le contrôleur intègre bien sûr un port de connexion aux cibles au travers du bus choisi. De plus, il possède lui-même une batterie pour le rendre nomade.

Il doit aussi être capable d'émettre un son, de la même manière que les cibles, pour indiquer un problème ou une fin de jeu à l'utilisateur.

Un écran rétro-éclairé étant employé, un capteur de luminosité est intégré, permettant d'économiser de l'énergie en gérant la puissance d'éclairage.

Un bouton extérieur peut y être branché. Ce dernier permet de lancer les modes de jeux enregistrés avec un accès rapide, ou de terminer un jeu.

Choix

Le choix se fait donc sur le **système « écran + 2 boutons + codeur rotatif »**, sans toutefois exclure la possibilité d'utiliser un écran tactile (prévu sur la plaque, mais non utilisé actuellement). Ce dernier pourrait être schématisé ainsi :

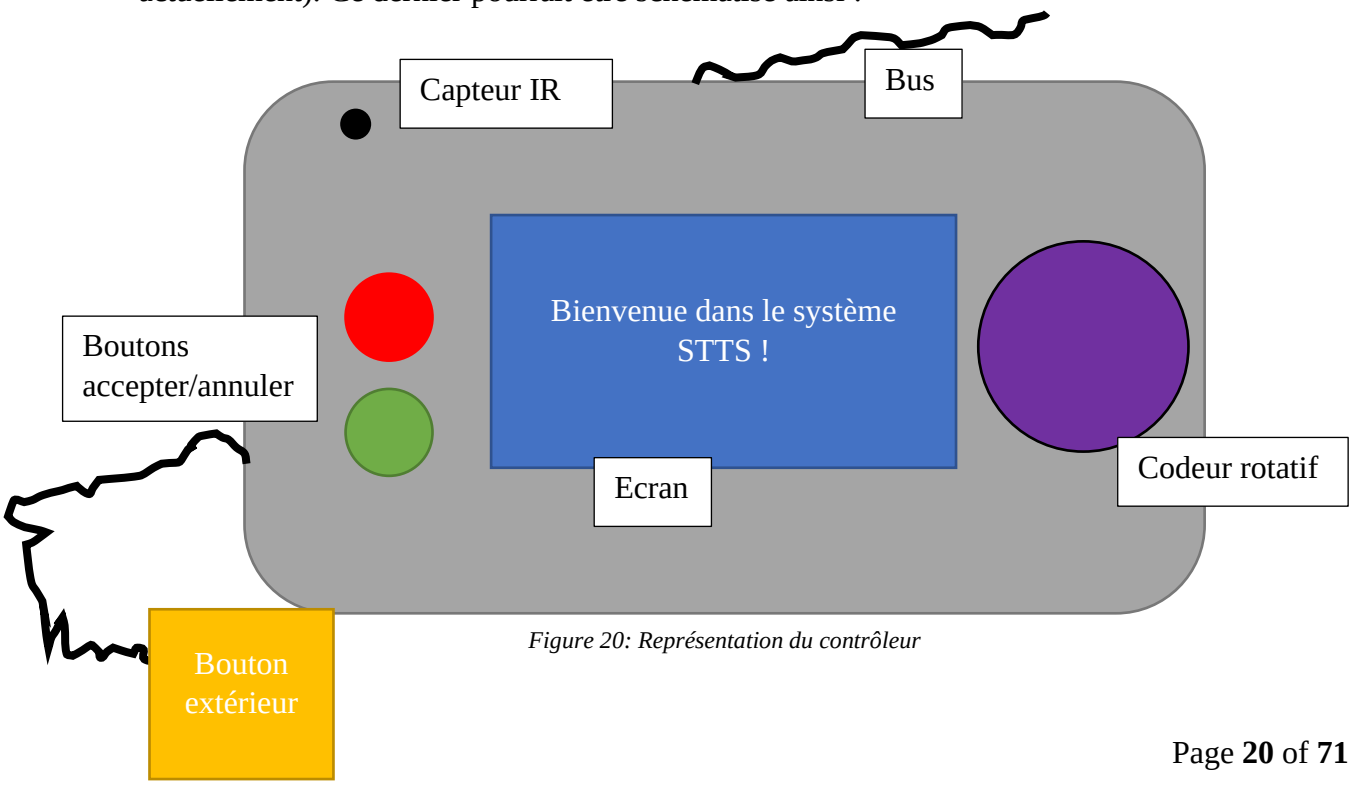


Figure 20: Représentation du contrôleur

4.9. Autres éléments

Système sans-fil

Afin de permettre d'éviter l'utilisation de câbles, un système Wi-Fi en topologie mesh (pour une plus grande distance d'utilisation) peut être envisagé :

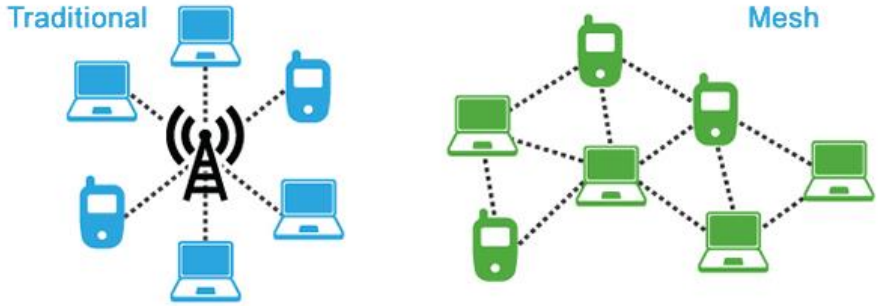


Figure 21: topologie Mesh,
https://paulbunyantech.com/what-is-mesh-networking-why-would-you-use-it/traditional-wifi-vs-mesh-wifi-network/

Dans cette optique, les cibles ainsi que le contrôleur sont équipés d'un emplacement pour puce Wifi ESP-12F.

Bien que non peuplé, il permettra de pouvoir continuer l'implémentation de ce système en dehors du cadre de ce projet. La gestion de base du module UART discutant avec la puce wifi ainsi que son alimentation ont été implémentés logiciellement.

Communication de debug

Pour aider à la conception du système, un module UART<->USB FTDI 232 est prévu sur les cibles ainsi que le contrôleur. Peuplé sur les versions de développement, il permet une communication bidirectionnelle avec un ordinateur.
Dans sa version finale, seul l'emplacement sur le contrôleur serait intéressant à garder, permettant par la suite de réaliser un programme de gestion des scénarios par ordinateur.

Communication Bluetooth

Une seconde façon d'offrir plus de personnalisation à l'utilisateur est d'équiper le contrôleur d'un système Bluetooth, permettant ainsi une plus grande souplesse sur la gestion des scénarios.

Dans cette optique, le contrôleur est équipé d'un emplacement pour RN4020 (puce BLE), mais non peuplé. Il pourra être utilisé pour la continuité du développement en dehors du cadre de ce projet.

Le système de gestion de la puce, par UART, est repris d'un projet précédent et est donc prêt à émettre et recevoir des commandes.

5. Electronique – Développement

Cette section regroupe les éléments électroniques communs aux cibles et au contrôleur.

5.1. Gestion de la batterie

Figure 22: batterie 18650

Charge de la batterie

La batterie, de type **18605** et de technologie **Li-Ion**, est rechargée par un circuit spécialisé, permettant la gestion automatisée du courant de charge.
La puce utilisée est la **MCP73833** [8] de chez Microchip, spécialisée pour la charge d'une cellule Li-Ion.

Selon les considérations du chapitre *4.3 Système nomade*, le courant doit être d'au moins 255 [mA]. Il est possible d'y régler le courant de charge à l'aide d'une résistance, selon l'équation suivante provenant du datasheet [8] :

$$Ireg\ [mA] = \frac{1000\ [V]}{R_{prog}\ [kOhms]}$$

Équation 2: courant de charge de la batterie

Avec une **résistance de 1.8 [kOhms]**, le **courant** donné est de **556 [mA]**. Ce courant est supérieur au minimum requis, et permettra une recharge plus rapide de la batterie en cas de besoin.

La charge se fait sur le même connecteur que le CAN, au travers d'un câble USB s'implantant sur la ligne.

Protection sur/sous-tension/charge

Cette dernière est aussi protégée par un circuit spécialisé, le **AP9101** [9], détectant la sur/sous-charge ainsi que la consommation excessive de courant, en coupant la masse du circuit en cas de problème :

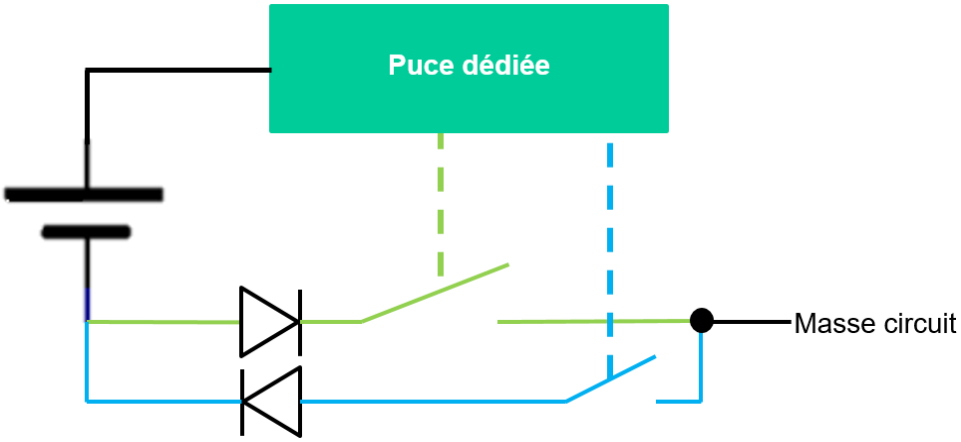


Figure 23: représentation du circuit de protection de batterie

Cette puce contrôle des MOSFETs N dual à enrichissement, le **DMG9926UDM** [10], *schématisés ici par des interrupteurs et diodes*, qui permettent ainsi d'activer ou désactiver indépendamment la charge (vert - désactivé en cas de surtension) ou décharge (bleu - désactivé en cas de surcharge ou sous-tension) de la batterie.

Protection en température

La batterie doit être utilisée dans une plage de température prescrite. Le chargeur de cellule précédent possède une entrée spécialisée dans ce sens.

Pour définir la plage, une thermistance de 10 [kOhms] et ses données constructeur [11] connues, on choisit de déterminer un pont de résistance permettant au chargeur de cellule d'opérer dans une plage de -10 [°C] à 50 [°C].

Cette plage est déterminée par :

- Valeur basse : la batterie Li-Po commence à se **détériorer** en dessous de – 20 [°C] ; une marge de sécurité est prise
- Valeur haute : la batterie Li-Po ne doit pas être utilisée au-dessus de 60 [°C], valeur où le **risque d'explosion** apparaît ; une marge de sécurité est prise

Selon les valeurs de détection du chargeur de cellule [8] page 4, **Vmin = 0.25 [V]** et **Vmax = 1.2 [V]**, la thermistance est donnée par l'équation :

$$B = \frac{T_2 * T_1}{T_2 - T_1} * \ln\left(\frac{R_1}{R_2}\right), T_x \; in \; [K]$$

Équation 3: Valeur de thermistance

Avec la valeur de thermistance [12] **B = 3434 [K]** insérée dans l'équation précédente :

- R50 = 4.102 [kOhms] @ 50 [°C]
- R-10 = 46.269 [kOhms] @ -10 [°C].

On détermine la résistance série telle que :

$$\frac{V_{min}}{I_{cst}} = R_s + \frac{R_p * R_{50}}{R_p + R_{50}}$$
$$\frac{V_{max}}{I_{cst}} = R_s + \frac{R_p * R_{-10}}{R_p + R_{-10}}$$

Équation 4: Pont de résistance pour thermistance

Avec **Icst = 50 [uA]** [12], on obtient :

- Rs = 1.242 [kOhms]
- Rp = 44.786 [kOhms]

La seconde solution de l'équation n'est pas interprétable dans le monde réel, avec une résistance calculée négative.

On choisit donc **Rs = 1.2 [kOhms]** et **Rp = 43 [kOhms]**, ce qui donne :

- Rtemp = 4168.37 [Ohms] => T = 322.665 [K]
- Rtemp = 48534.7 [Ohms] => T = 262.19 [K]

Ce qui équivaut à une plage de **-10.96 [°C]** à **49.515 [°C]**.

Alimentation sur et sans raccordement secteur

Puisque le système doit pouvoir être employé de façon nomade ou branché sur le secteur, le système à batterie précédemment présenté a été implémenté.

Toutefois, lorsque le système est toujours fixe :

- La batterie entraîne un **coût supplémentaire** inutile
- Faire tourner le système avec une **batterie** toujours branchée ne fera que l'**user** petit à petit

Dans cette optique, il serait possible de :

- Réaliser un circuit différent pour ce cas
  - Une plaque entièrement nouvelle – *coûts supplémentaires*
  - Un pont/court-circuit, outrepassant le système à batterie - *empêche complètement l'utilisation du mode nomade*
- Détecter la présence de la batterie, pour la charger ou non au besoin
  - A l'utilisateur de lui-même d'installer les batteries selon son utilisation

Cette seconde méthode est plus intéressante pour l'utilisateur final, qui module son système au besoin. Trois solutions potentielles permettent la détection de la batterie :

Utilisation d'un jumper

Puisque l'utilisateur doit lui-même mettre/retirer la batterie, il pourrait lui être mis à disposition un jumper, interrupteur … permettant de dire au système si la batterie est employée ou non.

Toutefois, cette solution est dangereuse si l'utilisateur omet de réaliser le changement lors de la mise en place ou du retrait de la batterie (mauvaise charge de la batterie, pas de détection de seuils).

Détection par courant

La batterie, si branchée, « recevra et émettra » du courant. Suivant le schéma, cette dernière est protégée contre les sur/sous-tensions. Il est donc possible de récupérer le courant la traversant pour déterminer si, oui ou non, cette dernière est branchée, à l'aide d'une résistance shunt :

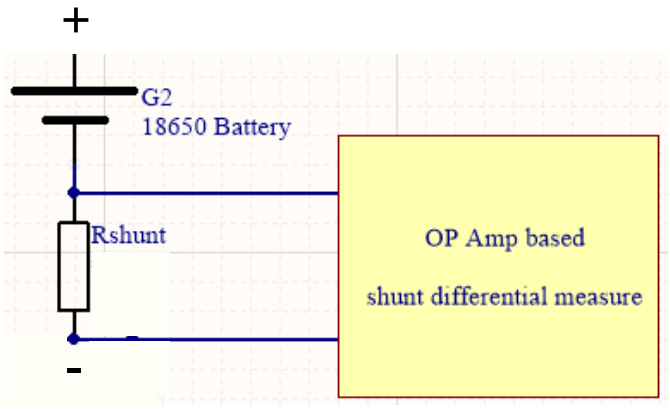


Figure 24: mesure sur shunt

Lorsque la batterie est présente, le courant traversant la résistance shunt crée une tension, détectée par un circuit adéquat. Dans le cas contraire, aucun courant ne passe et aucune différence de tension n'est mesurable.

Cette méthode présente le désavantage d'être intrusive, en créant des pertes dans le circuit ; et nécessitant de la circuiterie externe.

Détection de la batterie (physique)

Une méthode non intrusive et automatique est de détecter si la batterie est physiquement présente ou non dans son socle. Pour se faire, deux solutions :

Détection mécanique
La batterie vient appuyer un switch mécanique lorsqu'elle se trouve dans son socle. Peu chère, cette méthode ne nécessite pas (peu) de circuiterie externe, mais présente un risque mécanique (usure, mauvais appui).

Détection optique
Un émetteur-récepteur IR envoie de la lumière invisible. Lorsque la batterie est absente, la lumière « disparaît » plus loin. Lorsque cette dernière est présente, la lumière rebondit sur la batterie et vient toucher le récepteur :

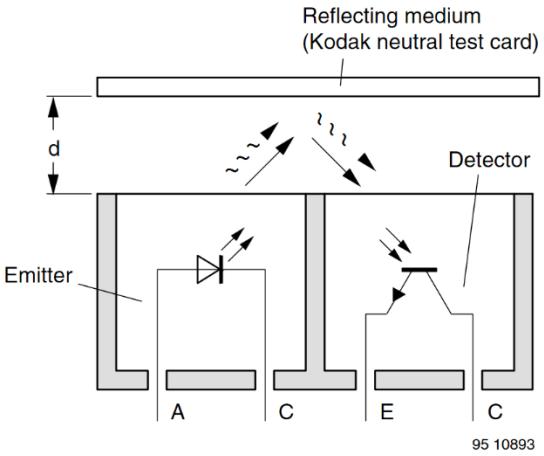


Figure 25: principe du capteur à réflexion, https://www.vishay.com/docs/83751/cny70.pdf

Cette seconde méthode présente toutefois un inconvénient : si la batterie s'avère être sombre, le signal sera absorbé et non détectée. De la même manière, la batterie étant très proche du capteur, le signal pourrait ne pas rebondir.

La **méthode à switch mécanique** est donc privilégiée, demandant le moins de matériel externe. Son désavantage, l'usure mécanique, n'est pas contraignante (la batterie n'est pas changée de multiples fois par seconde). Vu la forme du connecteur de batterie, il y'a peu de risques que cette dernière soit mal insérée et donc l'interrupteur mal pressé.

5.2. Mesure de la batterie

Dans l'idée de pouvoir réagir au niveau de charge de la batterie, un moyen de mesurer/estimer son niveau est mis en place.

Jauge de carburant de batterie

Les batteries ne suivant pas de courbe de tension linéaire suivant leur niveau de décharge, sans compter leur détérioration à la suite du nombre de cycles effectués ou encore de leur âge, une simple mesure ne permet pas de déterminer efficacement le niveau de batterie restant.

A l'inverse du réservoir d'une voiture, dont une simple jauge permet de connaître le niveau restant de carburant, plusieurs mesures continues doivent être effectuées sur une batterie pour en connaître son état.

Dans cette idée, des circuits spécialisés nommés « battery fuel gauge » ont été développés pour automatiquement surveiller la batterie. Les informations peuvent être récupérés par un protocole sériel simple (I2C, SPI …).

Mesure directe du niveau de tension

Comme présenté précédemment, la mesure directe de la tension de la batterie ne permet pas de connaître précisément l'état de la batterie.

La méthode consiste à lire (à l'aide d'un convertisseur A/D) le niveau de la batterie, puis de le comparer à un référentiel de mesure pour approximer son niveau actuel. Cette méthode requiert une simple entrée sur le uC.

Cette dernière est particulièrement employée pour les systèmes ne présentant que peu de moyens d'indiquer à l'utilisateur son niveau actuel (p.ex. 4 leds donnant respectivement les niveaux 25/50/75 et 100 [%] de charge).

Choix

Le niveau des différentes batteries ne peut être affiché que sur l'écran du contrôleur. Un niveau précis affiché pour chaque batterie serait illisible. La seconde méthode, largement suffisante et permettant de réduire les coûts, est choisie.

Les pins analogiques du uC n'étant pas tolérantes au 5V, un simple pont diviseur est réalisé entre Vbat et la masse, d'un rapport 3/5.

5.3. Conversion de tension

La cellule Li-Po pouvant varier de 2.2 à 4.2 [V], un régulateur doit être intégré pour créer une référence à 3.3 [V] fixe.

Deux cas se présentent :

- Un LDO (low-dropout regulator), régulateur linéaire qui dissipe de la puissance pour réguler la tension, mais ne requiert pas de composants externes
- Un convertisseur buck/boost, régulateur à découpage, qui permet des rendements de 90+ [%] mais requiert des bobines et génère plus d'IEM

Le buck/boost paraît meilleur pour offrir la plus grande efficacité. Toutefois, pour permettre une utilisation de la tension d'entrée proche de la tension régulée, ce dernier doit être finement développé et revient donc rapidement plus cher.

Le LDO possède bien sûr un rendement moindre, mais cela vaut pour des tensions d'entrée bien différentes de la tension régulée. Dans ce sens, la tension de la cellule à réguler est souvent proche de la tension régulée, et ainsi le rendement augmente (moins de chaleur à dissiper).

Moins cher, nécessitant moins de composants, et connaissant la note précédente, le **LDO** est choisi, l'**AP7363** [13].

Il est important de noter qu'il ne marchera pas une fois la tension de la batterie inférieure à 3.3 [V]. Dans ce cas de figure, il reste environ 5 à 10 [%] de capacité qui ne peut être utilisée. Il est toutefois intéressant de ne pas complètement décharger la batterie pour ne pas la détruire.

Note après conception
Après la mise en place du système et la réalisation des différentes mesures (7.2 Consommation), le LDO choisi est surdimensionné. Des notes d'améliorations sont données dans le chapitre susmentionné.

5.4. Communication CAN

Pilote pour signal différentiel
Les signaux du bus sont transmis sous forme différentielle.
Il est donc nécessaire de disposer, pour chaque nœud, d'un driver permettant d'une part d'attaquer la ligne différentielle et d'une autre de permettre de se brancher sur le module CAN du processeur.

Dans cette optique, avec la tension d'alimentation à disposition, le **MAX3051** [14] remplit ce rôle.

Nœuds sur la ligne
Le nombre maximal de nœuds potentiels sur le bus CAN est déterminé selon la topologie bus, c.à.d. les nœuds parallèles les uns aux autres, la ligne terminée aux deux extrémités par une résistance. La résistance équivalente est donnée par :

$CanNodeMinImputImpedance_1$ // $CanNodeMinImputImpedance_2$ … // $CanNodeMinImputImpedance_n$ // $TerminationResistor_1$ // $TerminationResistor_2$

Les deux résistances de terminaison de ligne, nécessaires pour éviter une réflexion sur la ligne, sont typiquement conseillées à 120 [Ohms] pour une paire cuivre torsadée.

On a ainsi :

$$\frac{TransceiverDiffOutputVoltage_{max}}{I_{transceiver_{max}}} \leq \frac{1}{\frac{1}{CNMII_1} + \frac{1}{CNMII_2} \dots + \frac{1}{CNMII_n} + \frac{1}{120} + \frac{1}{120}}$$

Que l'on peut réduire :

$$\frac{TransceiverDiffOutputVoltage_{max}}{I_{transceiver_{max}}} \leq \frac{1}{n * \frac{1}{CNMII} + \frac{1}{120} + \frac{1}{120}}$$

L'équation devient :

$$\boldsymbol{n \leq \left(\frac{I_{transceiver_{max}}}{TransceiverDiffOutputVoltage_{max}} - 2 * \frac{1}{120} \right) * CNMII}$$

Équation 5: Nombre de nœuds sur bus CAN

Avec la puce MAX3051, le pire des cas possible (en omettant les interférences externes, comme les pertes de la ligne de transmission) :

- TDOV = 3 [V] max
- It(max) = 70 [mA]
- CNMII = 40 [kOhms] (min)

On obtient **n = 266**, en omettant les facteurs extérieurs.

Couplé avec une transmission plus lente pour permettre une transmission longue distance, le système rentre donc dans les spécifications requises.

Protection DES/IEM

Selon les recommandations fabricant publiées dans l'AND8169/D [15], le bus CAN doit être protégé contre les surtensions potentielles.
En effet, même par l'utilisation d'un câble blindé pour la transmission de données, ces derniers peuvent être manipulés par l'utilisateur qui présente un potentiel danger de décharge électrostatique.

Dans cette idée, les principes suivants sont appliqués :

- Pistes courtes entre l'émetteur différentiel et le connecteur, afin de minimiser les stubs au maximum
- Diodes transil pour les deux lignes CAN_H et CAN_L, directement après le connecteur d'entrée

Le CAN est, par sa conception, utilisable dans un environnement bruité (lignes différentielles).
De plus amples mesures de protections ne sont pas requises pour cette application.

5.5. Connecteurs

Dans le but de pouvoir réutiliser le câblage de son actuel système de ciblerie, le mandant à demander d'implémenter les mêmes connecteur présent sur le système M.E.T [3], notamment :

- Pour la communication entre les cibles (5 pôles)
- Pour l'utilisation d'un bouton extérieur (2 pôles)

Ces connecteurs se trouvent être de la série **SC** de chez **Japan Automatic Machine**[7].

Malheureusement, il n'existe pas de revendeurs sur sol suisse ou environs, et l'import pour seules quelques pièces (actuellement) n'est financièrement pas crédible.

[7] http://www.jam-net.co.jp/eng/product/connector/post1356/

Il est donc décidé de trouver des embases de connecteurs capables d'accueillir les câbles actuels, et ce même si le connecteur femelle correspondant n'est pas lui-même rétro-compatible.

Les caractéristiques requises sont :

- Pas de **2.5 [mm]**
- Embase **rectangulaire**
- **Broches décentrées** (détrompeur de connexion)

Les connecteurs sélectionnés sont de la série **SxB-XH-A** de **JST**.
Par leur forme, ils sont capables d'accueillir les connecteurs voulus.
La différence réside dans le verrouillage dans l'embase, en plus d'un chanfrein présent sur le modèle JAM :

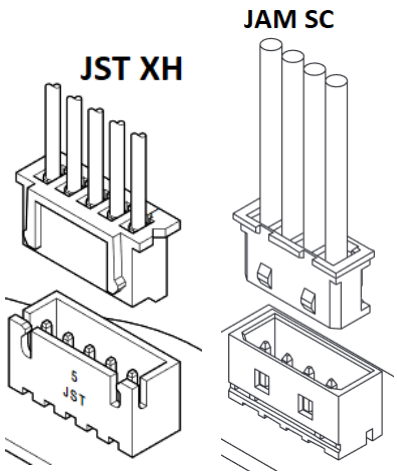


Figure 26: connecteurs JST XH vs JAM SC, http://www.jst-mfg.com/product/pdf/eng/eXH.pdf / http://www.jam-net.co.jp/uploads/product/1356/2_1.pdf

5.6. Bouton externe – antirebond

Voir 5.14 Encodeur rotatif / boutons – antirebonds

5.7. Microcontrôleur

Après l'établissement de tous les besoins, le système le plus gourmand en entrées-sorties, le contrôleur, nécessite **49 pins**, ainsi qu'une entrée pour un oscillateur externe, avec :

1 I2C 4 UARTs 2 entrées A/D 3 interruptions 1 module CAN

Une librairie graphique étant utilisée, il est nécessaire de disposer de suffisamment de place pour stocker logos et polices d'écriture.

Le choix se porte sur le **dsPIC33EP512GM706** [16].

- La gamme PIC18 de Microchip possède certains modèles avec contrôleur CAN, mais la place en Flash n'est pas assez importante.
- Les contrôleurs d'autres marques (STM, ESP …) ne sont pas considérés, le dsPIC étant simple de montage ainsi que de programmation, peu coûteux, et le système se suffisant sans mémoire dynamique.

La gamme 33EP comprend des contrôleurs avec différentes mémoires, permettant de les changer au besoin pour réduire le coût une fois le programme terminé (actuellement, modèle avec 512 [kO] de Flash).

5.8. Oscillateur

En raison de l'utilisation du bus CAN et de l'I2C en fast-mode, une horloge précise est requise (plus de précision sur les minutages).
Un quartz peut être utilisé avec l'oscillateur interne du PIC ; ou un oscillateur externe peut jouer le rôle d'horloge.

Ce dernier peut être éteint au bon vouloir lors d'un changement d'horloge pour réduire la consommation du système au repos. De plus, il requiert moins de place sur le PCB.
L'oscillateur est donc sélectionné.

Viennent ensuite deux technologies concurrentes : MEMS vs Crystal.

Selon l'étude de Geyer Electronic [17], l'oscillateur à quartz est plus rapide à s'éveiller, avec une consommation légèrement moindre ainsi qu'un bruit de phase moins important. Bien que la technologie basée MEMS se répute être 4x plus endurante, on parle ici de 30'000 ans pour un oscillateur quartz contre 120'000 ans pour le MEMS ; dans une telle application, cet argument n'a pas de poids pour motiver l'utilisation d'un tel oscillateur.

Un oscillateur externe basé cristal est utilisé, cadencé à 4 [MHz], utilisé ensuite avec PLL, le **ECS-5032MV** [18].

5.9. Debug sériel

Un chip spécialisé permet de convertir l'UART en un port COM virtuel pour debug sur PC.

Ce dernier est implémenté dans les cibles et le contrôleur principal, mais ne sera peuplé après coup uniquement dans le contrôleur, permettant un futur point d'entrée pour régler le système par une application PC dédiée.

Est choisi le **FTDI230XS-R** [13], avec un connecteur Micro-USB B.

5.10. LEDs de debug

Pour aider au développement, trois LEDs simples sont ajoutées au système.

Le microcontrôleur étant plus performant à tirer du courant plutôt qu'à en fournir – chapitre 33 du datasheet [16], ces dernières sont constamment branchées au 3.3 [V] et tirées à la masse au besoin.

Absolute Maximum Ratings
(See Note 1)

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +160°C
Voltage on VDD with respect to Vss	-0.3V to +4.0V
Voltage on any pin that is not 5V tolerant with respect to Vss(3)	-0.3V to (VDD + 0.3V)
Voltage on any 5V tolerant pin with respect to Vss when VDD ≥ 3.0V(3)	-0.3V to +5.5V
Voltage on any 5V tolerant pin with respect to Vss when VDD < 3.0V(3)	-0.3V to +3.6V
Voltage on VCAP with respect to Vss	1.62V to 1.98V
Maximum current out of Vss pin	350 mA
Maximum current into VDD pin(2)	350 mA
Maximum current sunk by any I/O pin	20 mA
Maximum current sourced by I/O pin	18 mA
Maximum current sourced/sunk by all ports(2,4)	200 mA

Figure 27: courant fournit/tiré par une entrée-sortie

Cette section regroupe les éléments électroniques de la cible.

5.11. Accéléromètre et I2C

L'accéléromètre sélectionné, bas-coût, avec une sensibilité minimale de 0.98 [mG], fonctionne en I2C [7].

Bus I2C

Selon le document de Texas Instrument [20], les résistances du bus I2C sont calculées telles que :

Vcc = 3.3 [V] => Iol = 3 [mA]
Vol = 0.4 [V]

$$R_{p(min)} = \frac{V_{cc} - V_{ol(\max)}}{Iol} = \frac{3.3 - 0.4}{3m} = 967\ [Ohms]$$

Équation 6: PullUp I2C minimale

Fast-mode => tr = 300 [ns], cb(max) = 400 [pF]
Cb = 10 (pic)[8] + 10 (accéléromètre)[9] + 100 (ligne)[10]

$$R_{p(max)} = \frac{t_r}{0.8473 * C_b} = \frac{300n}{0.8473 * 120p} = 2.95\ [kOhms]$$

Équation 7: PullUp I2C maximale

Rp doit donc se trouver entre 967 et 2950 [Ohms]. Tout en prenant une sécurité supplémentaire sur la capacitance de ligne (Rp <<), tout en restant sur une consommation minimale (Rp >>), **Rp** est défini à **2.4 [kOhms]**.

5.12. Capteur de présence

Dans le but de détecter la présence d'un joueur, dans une pièce par exemple, le capteur choisi est un capteur PIR **HC-SR501** [21].

Principe de fonctionnement

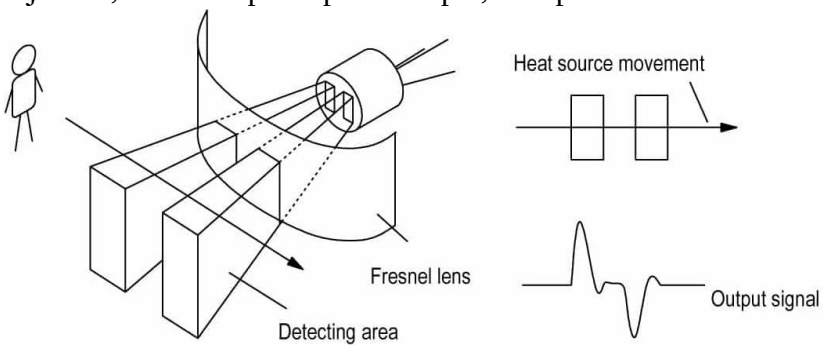


Figure 28: fonctionnement système PIR, https://www.makerguides.com/wp-content/uploads/2019/07/HC-SR501-PIR-Motion-Sensor-working-principle.jpg

Le principe repose sur deux canaux distincts, récupérant le taux d'infrarouge dans leur ligne de vision. Lorsque l'utilisateur entre dans le champ, le taux change, le circuit détecte et traite ce changement et génère une interruption. Une lentille de Fresnel permet de concentrer le rayonnement sur le capteur, et ainsi permet une meilleure sensibilité en plus d'une meilleure portée. Ces deux variables sont réglables à l'aide d'un potentiomètre.

[8] Datasheet page 484
[9] Standard CMOS avec marge
[10] Ligne courte, marge de sécurité

Tension d'utilisation

Ces derniers se trouvent à bas prix à l'étranger (moins de 1 [$]), en ayant le désavantage de requérir une tension d'alimentation de 5 [V].
Toutefois, comme le présente le billet de blog de Techgurka [22], ces derniers utilisent un régulateur 5 [V] => 3.3 [V] pour ensuite attaquer les composants internes.
Il est donc possible d'alimenter directement les composants avec une tension extérieure de 3.3 [V], soit en outrepassant ce régulateur, soit en accédant directement à la ligne 3.3 [V] interne avec un pad découvert sur le circuit.

Le « désavantage » étant que la protection de polarisation n'existe dès lors plus. Ce n'est toutefois pas un problème, l'utilisateur final n'ayant pas la possibilité d'accéder directement au câblage de ce capteur.

5.13. RGB

Un jumper permet de sélectionner une tension d'alimentation régulée à 3.3 [V], ou de se brancher directement sur la tension de la batterie (Li-Ion => 2.2 – 4.2 [V]).
La tension régulée permet une luminosité constante, mais moindre (tension R/B proche de la tension régulée) ; la tension de la batterie permet de régler ce souci mais entraîne une légère variation de la luminosité au cours du temps.

La plaque est prévue pour deux types de LEDs, traversante 3 [mm] et SMD.

Pour une utilisation sous 3.3 [V] :

LED 3 [mm]

If = 30 [mA], Vr = 1.8 [V], Vg = Vb = 3.2 [V], Vin(min) = 3.7, Rds(on)max = 57 [mOhms]

$$Vds = Rds_{on(\max)} * If = 57m * 30m = 1.71\ [mV]$$

$$R = \frac{V_{in(\min)} - V_{gs(sat\_min)} - V_{led}}{If} = \frac{3.7 - 1.71m - 1.8|3.2}{30m}$$

Équation 8: résistance des LEDs

Rr = 63.3 => **62 [Ohms]**, **Rg = Rb** = 16.6 => **16 [Ohms]**

LED SMD

If = 50 [mA], Vr = 2 [V], Vg = Vb = 3.2 [V], Vin(min) = 3.7, Rds(on)max = 57 [mOhms]

Rr = 37.9 => **39 [Ohms]**, **Rg = Rb** = 9.97 => **10 [Ohms]**

Ajustements

La luminosité n'étant pas équivalente pour chaque couleur, ces valeurs sont prises à titre de résistance minimale.

Il est ensuite nécessaire de les ajuster pour un meilleur mélange de couleur.

Notes après test

Le mélange avec les valeurs calculées donne un résultat satisfaisant. De plus amples ajustement n'ont pas été réalisés.

Cette section regroupe les éléments électroniques du contrôleur.

5.14. Encodeur rotatif / boutons – antirebonds

L'encodeur rotatif [23], à contacts mécaniques et de 12 révolutions par tour, doit être équipé d'antirebonds, permettant de ne pas devoir traiter l'information de façon logicielle et ainsi pouvoir les utiliser sur interruption directement. De même pour les boutons ok, retour [24] et l'interrupteur extérieur.

Pour se faire, un filtre RC est utilisé comme proposé par le constructeur [12]:

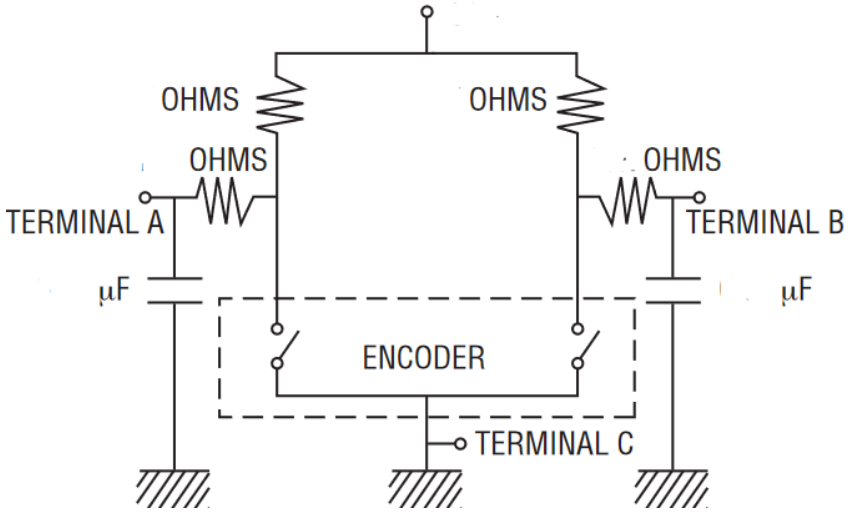


Figure 29: circuit antirebonds pour encodeur rotatif, Bourns

Les temps d'antirebond sont définis par :

$$T_{open} = V_{trans} * \left(R_{filtre} + R_{pullup}\right) * C_{filtre}$$

$$T_{close} = V_{trans} * R_{filtre} * C_{filtre}$$

Équation 9: temps d'antirebonds

Le temps proposé par le constructeur est de 1 [ms] pour le signal haut et 500 [us] pour le signal bas, avec Rfiltre = Rpullup = 10 [kOhms], Cfiltre = 10 [nF] et Vcc = 5 [V].

Adapté à notre situation :

Vtrans = 3.3 [V], Topen = 4 [ms], **Rf = Rp = 47 [kOhms]**

$$\boldsymbol{C_{filtre}} = \frac{T_{open}}{V_{trans} * (R_{filtre} + R_{pullup})} = \frac{4m}{3.3 * (2 * 47k)} = 12.9\ [nF] => \ \boldsymbol{10\ [nF]}$$

Avec ces valeurs, on obtient **Topen = 3.1 [ms]** et **Tclose = 1.55 [ms]**.

Pour les boutons auxiliaires :

Vcc = 3.3 [V], Topen = 200 [ms], **Rf = Rp = 47 [kOhms]**

$$\boldsymbol{C_{filtre}} = \frac{T}{V_{cc} * (R_{filtre} + R_{pullup})} = \frac{200m}{3.3 * (2 * 47k)} = 645\ [nF] => \ \boldsymbol{680\ [nF]}$$

Avec ces valeurs, on obtient **Topen = 211 [ms]** et **Tclose = 105 [ms]**.

5.15. Ecran

Un écran est nécessaire pour relayer les informations à l'utilisateur. Plusieurs technologies existent, ainsi que différents types d'affichage, notamment :

- A ligne

Figure 30: LCD à lignes, https://startingelectronics.org/beginners/components/LCD/LCD_2x16.jpg

- Graphique

Figure 31: LCD graphique, https://media.digikey.com/Photos/Newhaven%20Display%20Photos/MFG_NHD-2.4-240320CF-CSXN%5E-F.jpg

Pour afficher toutes les informations nécessaires, ainsi que d'avoir un rendu plus intéressant, un LCD graphique est choisi. Certains sont équipés d'écrans tactiles, permettant une amélioration future du système.

Possédant des bibliothèques spécialisées, de taille et résolution respectable (240*320 sur 2.4 pouces) et ces derniers étant à bas coût, le **NHD-2.4-240320CF-CSXN** [25] est sélectionné, version optimisée pour l'utilisation sous le soleil.

La communication se fait par bus 8 ou 16 bits en parallèle.

5.16. Schémas

Le système étant découpé en 4 parties (contrôleur, cible, module détecteur de présence et module LEDs et Buzzer), 4 schémas – et donc 4 plaques électroniques – sont réalisées.

Pour le contrôleur et la cible, les circuiteries sont revisitées à la suite d'une fausse implémentation du système de gestion de batterie.

Les nouvelles révisions sont données sous *10.2 Schémas actuels (contrôleur 1.1, cible 1.1, module led 1.0, module détecteur de présence 1.0)*.

Les anciennes révisions sont données sous *10.3 Schémas datés (contrôleur 1.0, cible 1.0)*.

5.17. Cartes 1.0

Les cartes suivantes représentent les versions des schémas 1.0, utilisés pour le développement du système :

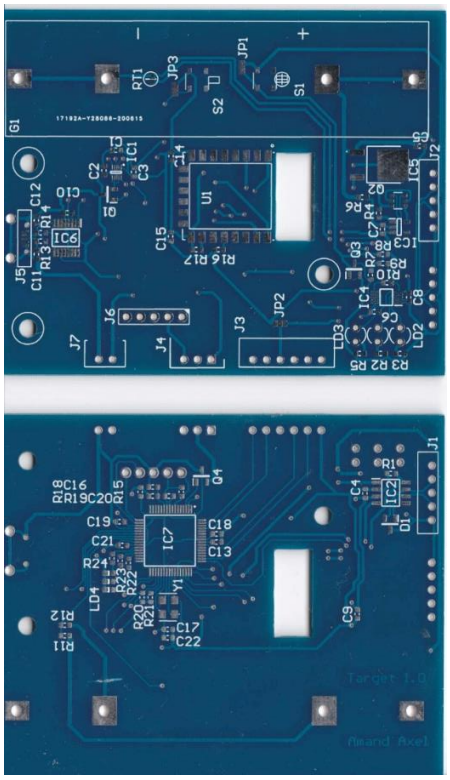


Figure 32: PCB cible 1.0

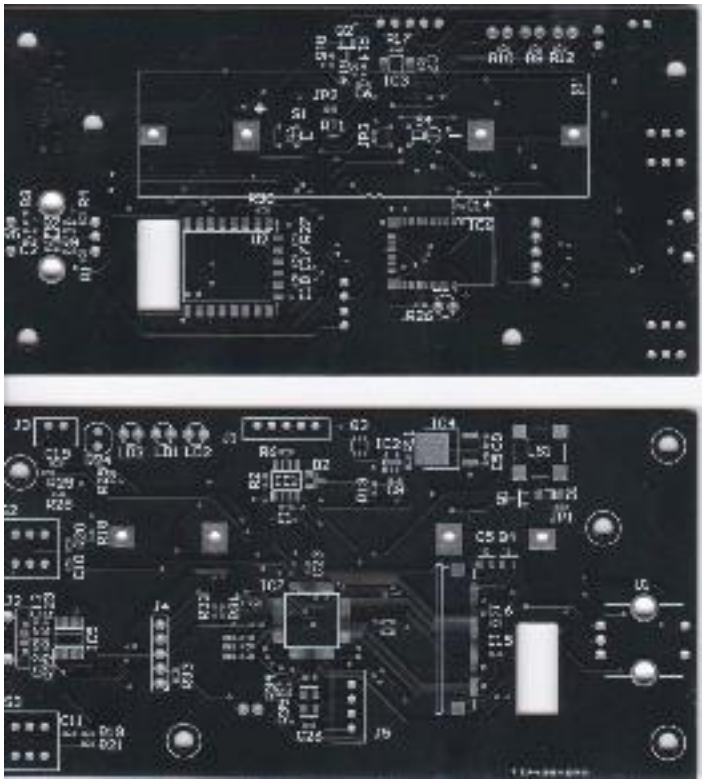


Figure 33: PCB contrôleur 1.0

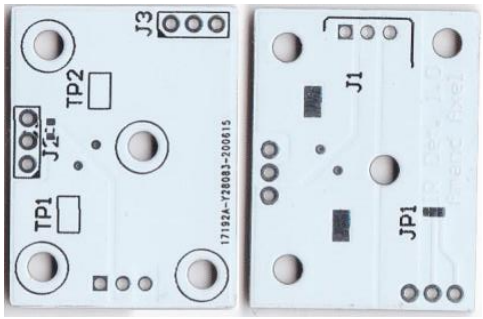


Figure 34: PCB module détecteur de présence 1.0

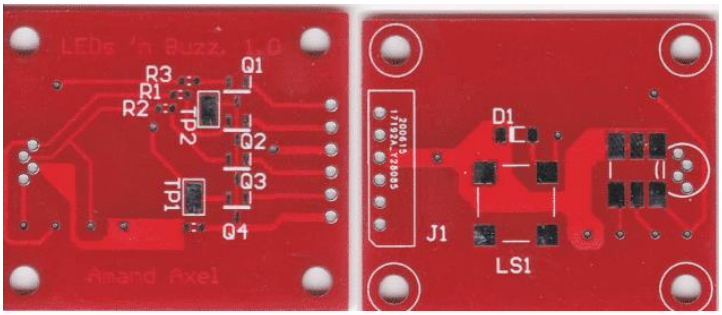


Figure 35: PCB module Led et Buzzer 1.0

6. Logiciel

6.1. Analyse et diagrammes

Le système trouve son centre au sein du contrôleur, qui s'occupera de gérer le jeu, les cibles, la détection de fin ainsi que les réglages du système.

Les cibles, quant à elles, répondent à des commandes simples selon un schéma prédéfini, schéma qui convient pour tous les modes de jeux employés, capables de réagir aux chocs et d'indiquer leur état.

Composants des sous-systèmes

Le système peut être représenté en 4 parties distinctes :

- **Le contrôleur**, qui comprend tout le nécessaire pour fonctionner
- **La cible**, qui possède deux modules externes en plus de sa propre circuiterie
- **Le module LEDs et buzzer**, qui comporte une LED RGB et un buzzer
- **Le module PIR**, qui comporte le détecteur de présence

Les composants de ces 4 éléments peuvent être représentés ainsi :

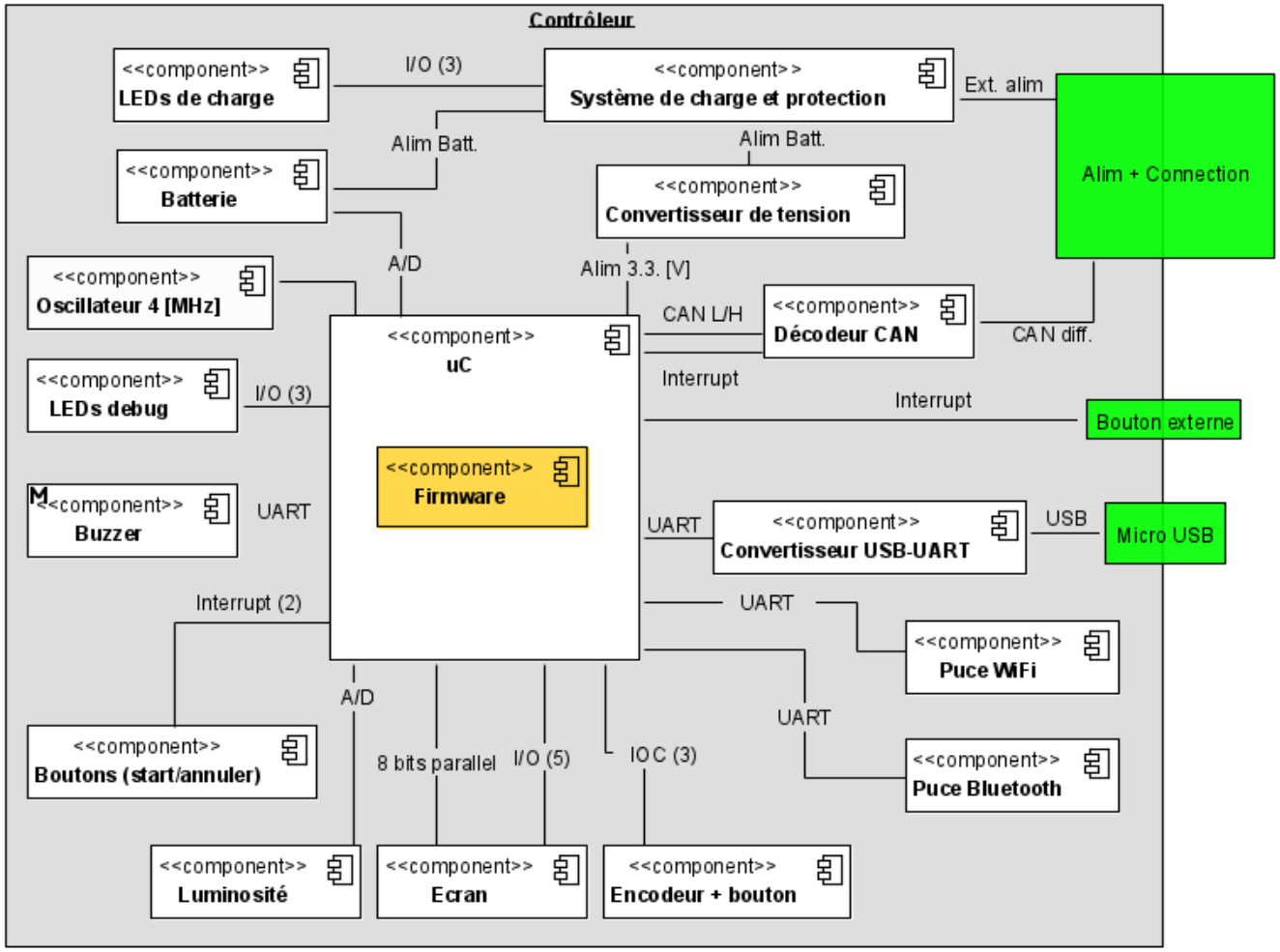


Figure 36: composants du contrôleur

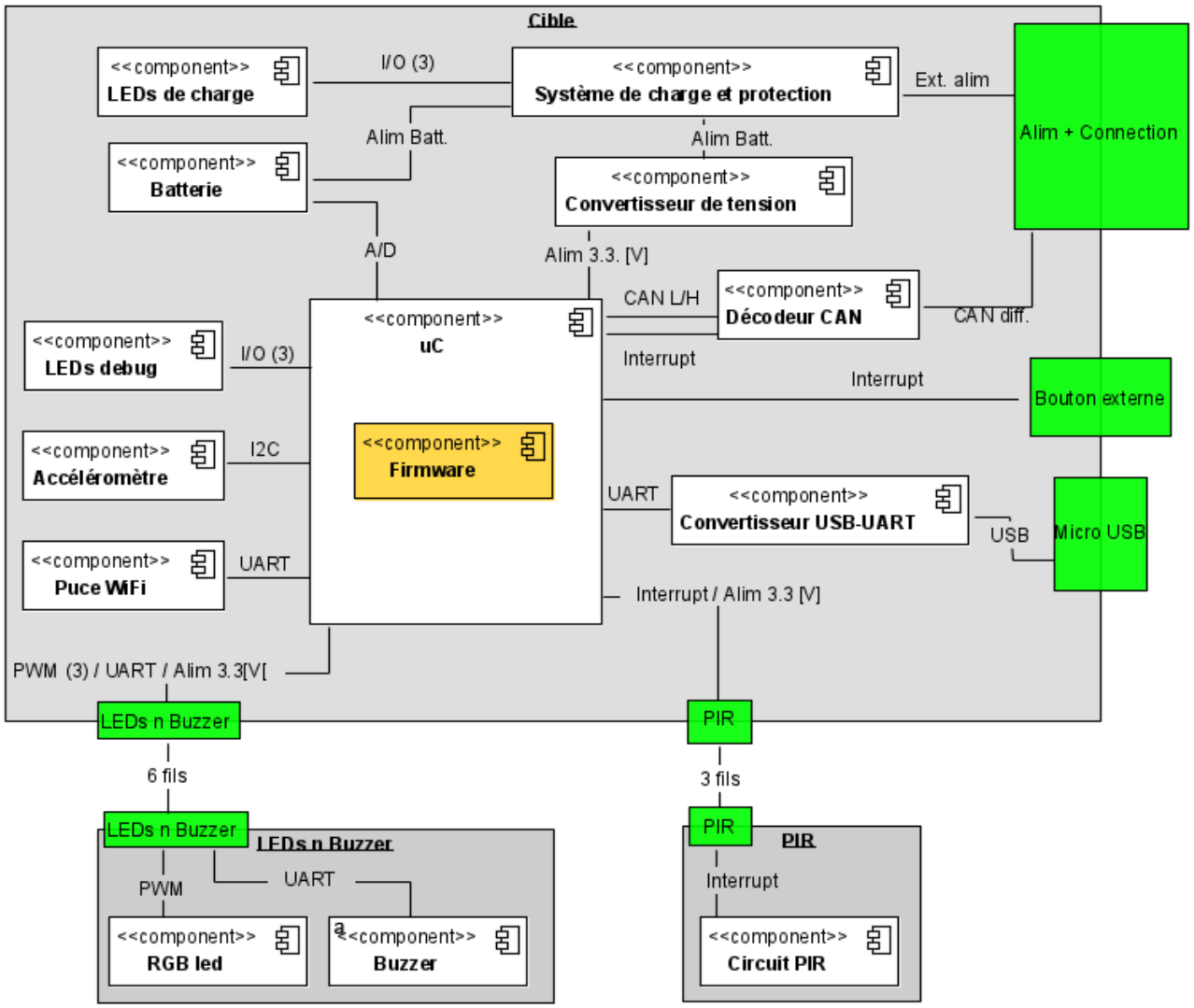


Figure 37: composants de la cible et de ses modules

Dans le cas du circuit de la cible et du contrôleur, tout est axé autour d'**un seul microcontrôleur**, en dehors du système de charge et de contrôle de la batterie qui, lui, marche indépendamment.

Ne possédant pas d'autre superviseur, il est donc important d'intégrer un **système de redémarrage en cas de bug**.

Les systèmes se basent sur l'implémentation d'un **distributeur événementiel**, permettant de travailler de façon asynchrone et non-bloquante.

Pour le module annexe « LEDs n Buzzer », tout est **piloté depuis la carte cible**.

Dans le cas du module PIR, ce dernier possède une circuiterie qui lui est propre, et génère une **interruption envers le microcontrôleur de la carte cible**.

Possibilités de l'utilisateur

Une fois le système bien défini, il est possible d'établir la liste des interactions des différents utilisateurs avec le système :

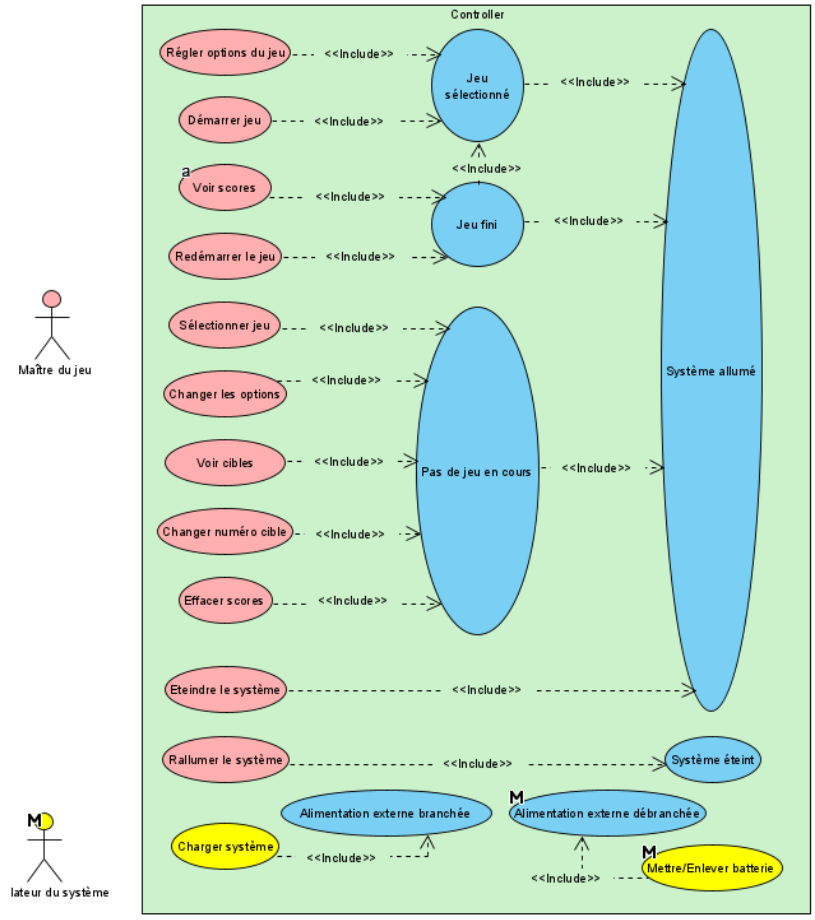


Figure 38: possibilité utilisateur - contrôleur

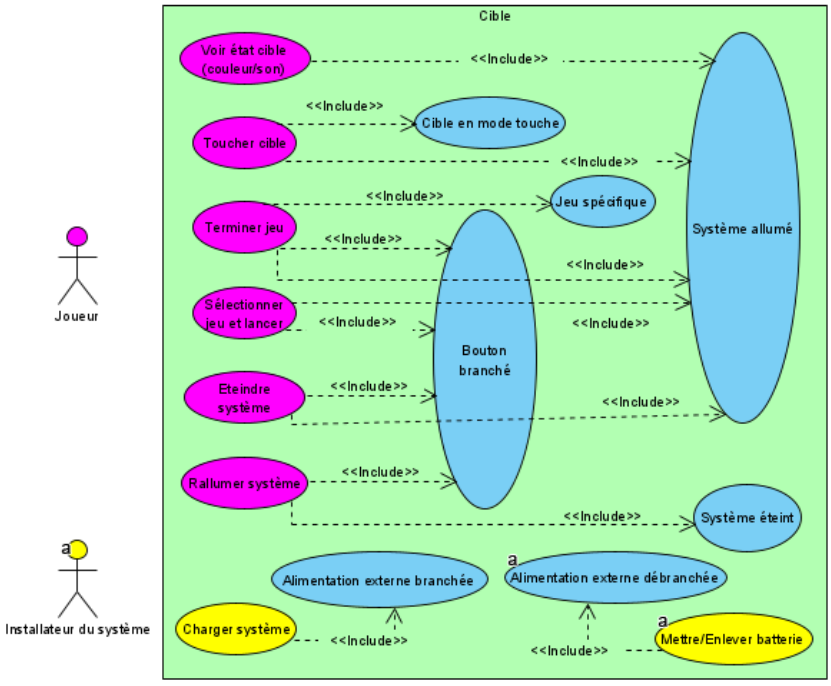


Figure 39 : possibilités utilisateur – cible

On recense trois acteurs :

- **L'installateur système**, qui est la personne mettant en place le système. Ce dernier peut choisir de lier le tout au secteur ou non, et ainsi de mettre en charge le système.
- **Le maître du jeu**, qui gère le réglage des modes de jeu, les options, et qui peut par accès au boitier voir les scores.
- **Le joueur**, qui interagit avec les cibles au sein notamment du scénario chargé.

La frontière entre **maître du jeu** et **joueur** n'est pas nette. Le rôle peut être partagé :
- dans le cadre professionnel, un instructeur est **maître du jeu** et gère les scénarios, pendant que ses apprentis **joueurs** défilent sur le parcours
- dans le divertissement, un simple **joueur** règle le jeu qu'il veut, seul, puis joue

L'**installateur système** est en général l'acheteur de la ciblerie.

Logique de la cible

L'idée principale est telle que le contrôleur gère l'entièreté des jeux, tandis que les cibles ne font que répondre aux commandes envoyées, en travaillant selon une machine d'état simple.

Pour cela, il faut en définir les états, capables de travailler pour tous les modes de jeux possibles. Ces états sont les suivants :

- **RESET** : la cible vient de redémarrer -> s'initialise et passe à l'état suivant
- **CONNECT** : la cible attend de recevoir une demande de connexion du contrôleur, tout en ayant l'air éteinte. Sans cette demande, la cible ne pourra être utilisée. Aussi, cet état permet, si la cible ne reçoit aucune réponse sous 5 secondes, de décider de dormir et abaisser sa consommation.
- **WAITINFOS** : cet état permet de charger les informations sur la cible : adresse, groupe, volume, couleurs selon état, bips selon état, temps de transition ; cet état allume d'un blanc doux la lumière de la cible, permettant à l'utilisateur de voir si la cible est prise en compte ou non
- **STANDBY** : l'état standby est le premier état de jeu, où la cible ne peut pas être touchée ; c'est l'état par défaut lorsqu'un jeu démarre, permettant de colorer la cible pour permettre à l'utilisateur de savoir qu'il n'est pas encore possible de tirer dessus ; la transition se fait sur commande du contrôleur où par une cible du même groupe
- **CAPTEUR** : l'état capteur est un état de jeu, où la cible peut être touchée mais en plus peut détecter le joueur ; lors de la détection, la cible est capable d'avertir celles du même groupe qu'un joueur est détecté, et ainsi permettre à ces dernières de s'allumer
- **ON** : l'état on est un état de jeu, où la cible peut être touchée
- **HIT** : l'état hit est un état de jeu, où la cible a été touchée par le joueur
- **ENDGAME** : l'état endgame est le dernier état de jeu, où la cible garde la couleur de son précédant état, mais où l'utilisateur ne peut plus interagir avec ; utilisé notamment pour permettre au joueur de se rendre compte des cibles ratées en fin de scénario
- **ERROR** : l'état error entre la cible dans un état qui ne peut être récupéré que par le contrôleur, ou par une déconnexion du reste du système; l'entrée dans cet état se fait uniquement par commande du contrôleur, lorsque ce dernier détecte deux cibles possédant la même adresse, pour les rendre inutilisables en jeu tant que le problème n'est pas réglé ; débrancher un des doublons permet au contrôleur de sortir la cible de cet état

Avec ça, la machine d'état correspondante revient à :

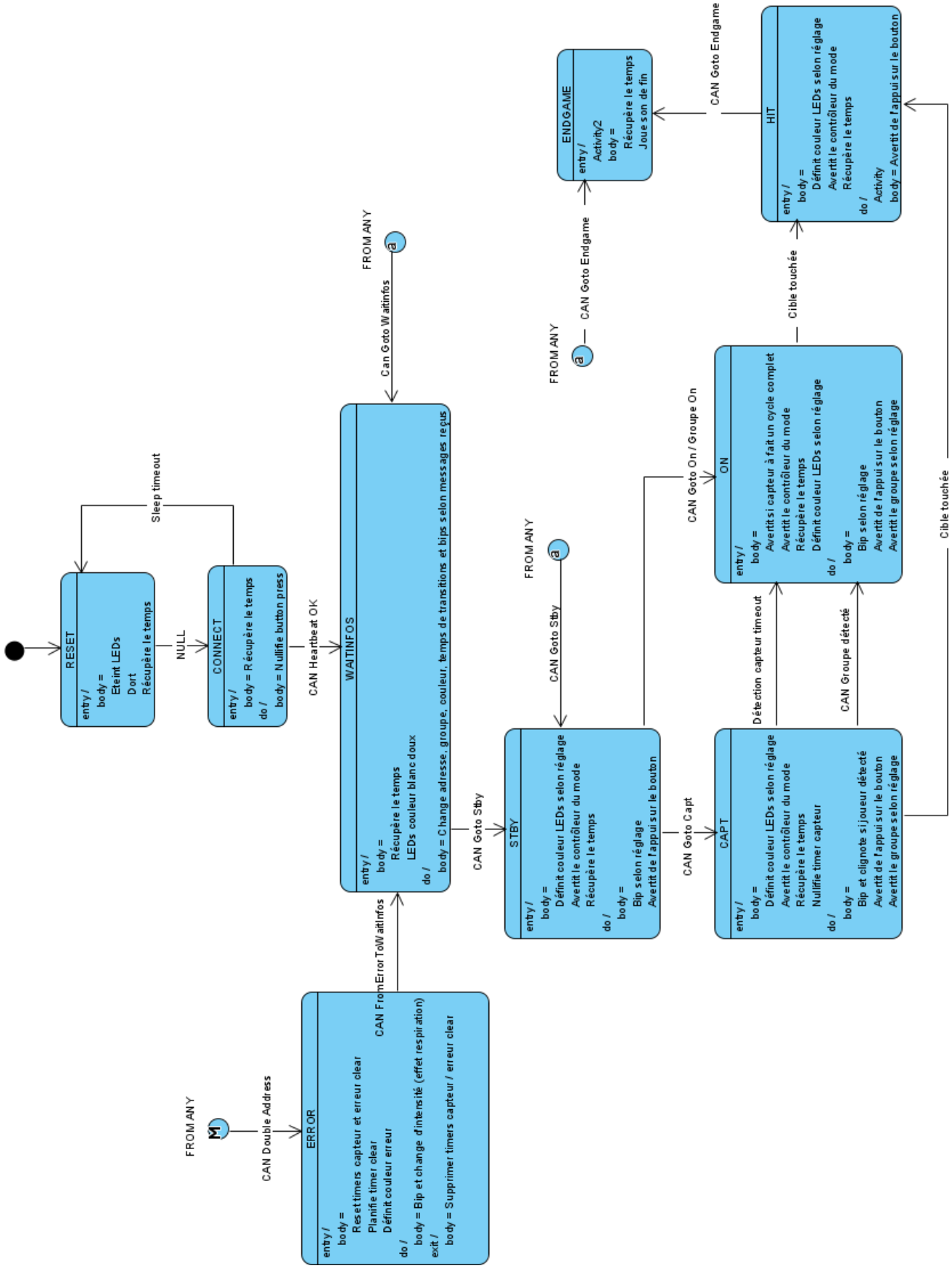


Figure 40: machine d'état – cible

Contrôleur et écrans

Ecrans

Le contrôleur possède une logique propre à chaque « écran » que ce dernier peut afficher. Dans cette idée, ces écrans sont parcourus par l'utilisateur tels que :

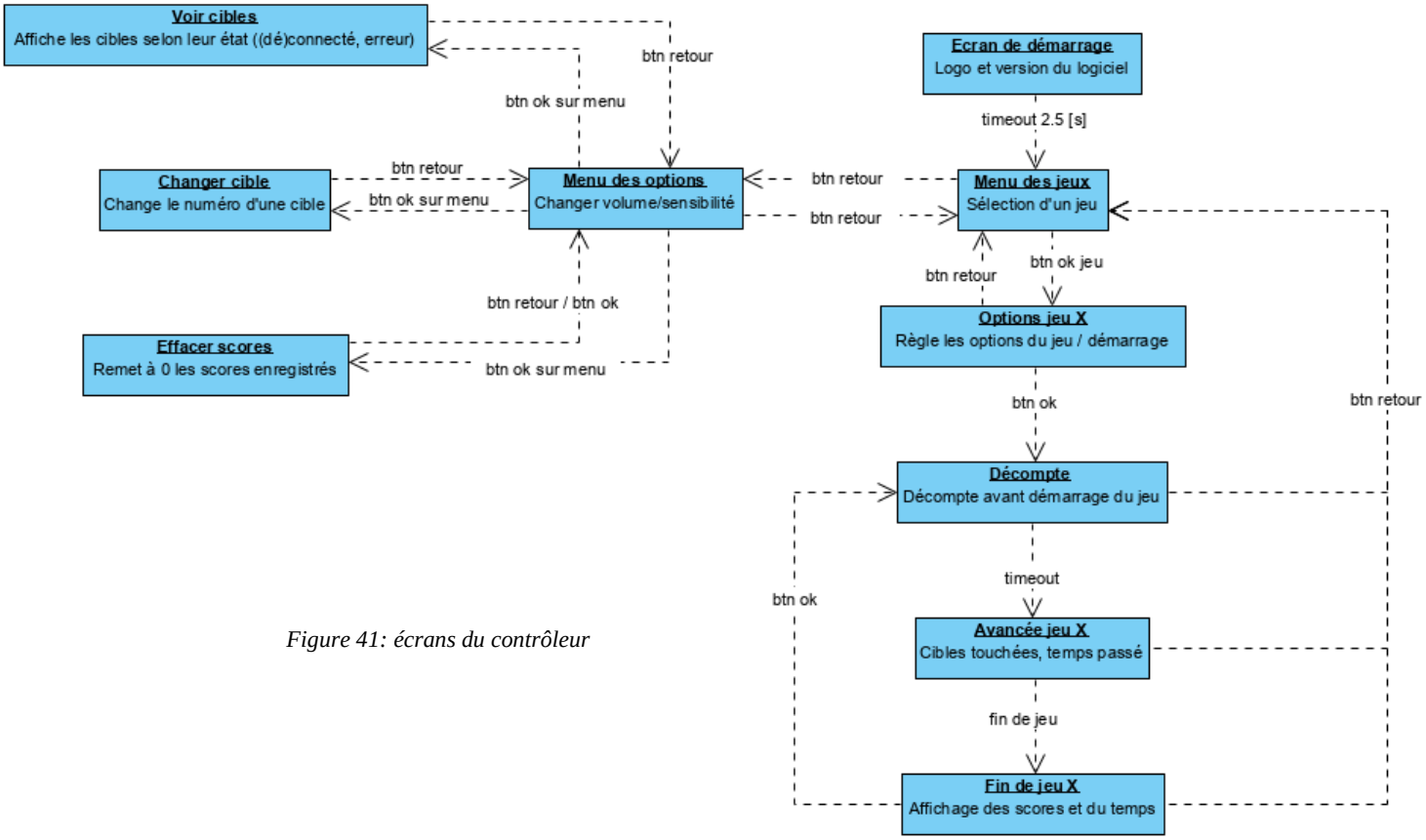


Figure 41: écrans du contrôleur

Niveau d'imbrication

La logique même du contrôleur se distribue sur plusieurs niveaux imbriqués, afin de permettre de dissocier chaque partie du sous-système afin de ne pas influencer l'une et l'autre :

1) **une machine d'état globale**, propre au traitement des événements de gestion du système (contrôle des I/Os, des différents modules, du watchdog …) et qui transmet ensuite les événements au bon écran
2) chaque **écran réagit aux événements**, sans machine d'état spécifique, pour modifier l'affichage et répondre aux entrées de l'utilisateur
3) une fois un scénario lancé, un troisième niveau gère **une machine d'état pour le contrôle du suivi du jeu**, généralisée pour tous les jeux
4) une fois cette dernière lancée, **chaque jeu peut choisir de s'enregistrer pour recevoir les événements d'entrée/faire de la machine précédente**, afin de réagir spécifiquement au besoin

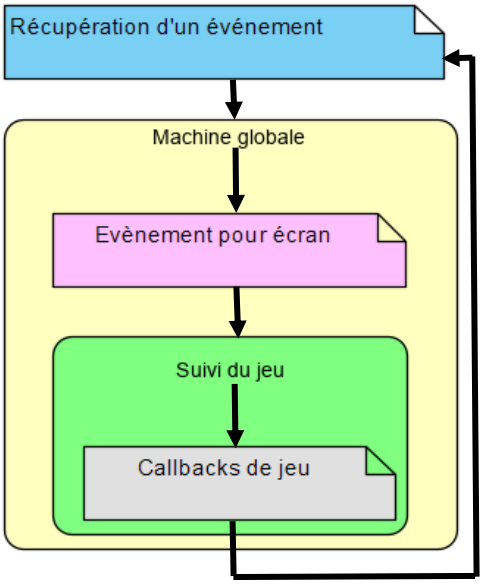


Figure 42: distribution dans les niveaux imbriqués - contrôleur

Machine globale

La machine globale se présente ainsi :

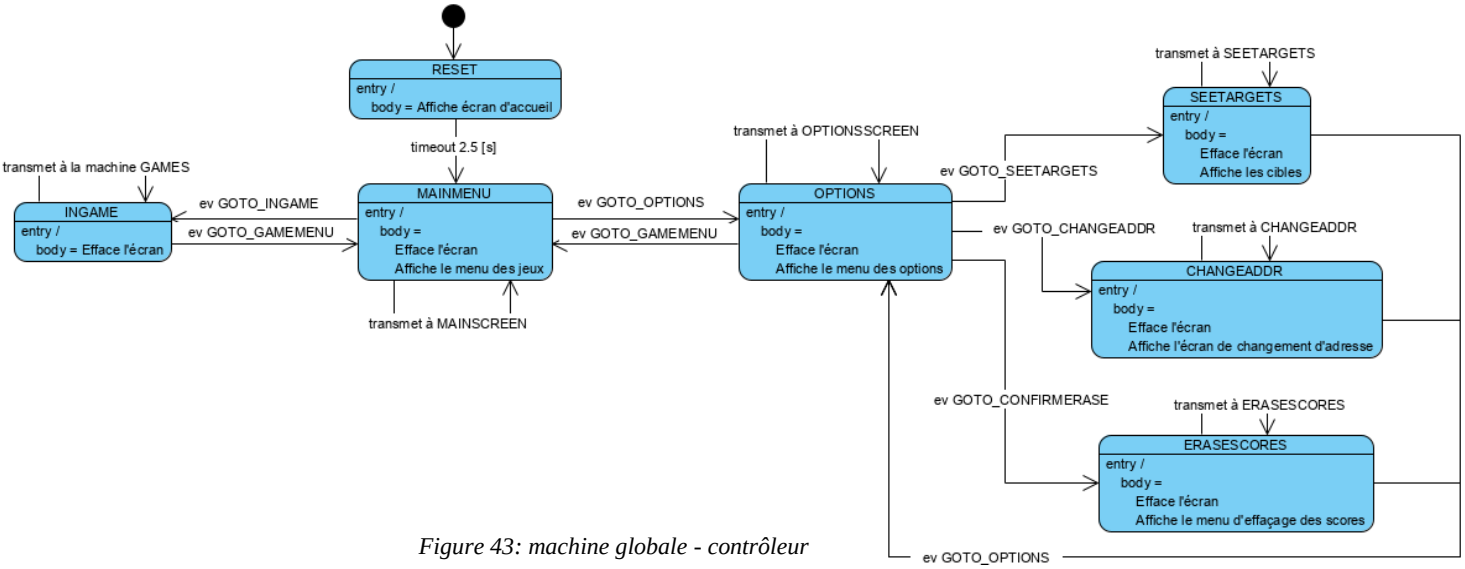


Figure 43: machine globale - contrôleur

Les écrans sélectionnés réagissent ensuite aux événements, comme indiqué sur le schéma précédent. Puisqu'ils ne font que répondre aux entrées, sans nécessiter de connaître un quelconque état précédent, une machine d'états ne leur est pas dédiée.

Cas d'un jeu

Dans le cas spécial d'un jeu, il est cette fois important de savoir où ce dernier en est pour le gérer aisément. Les 4 états suivants le permettent :

- **NONE** : état par défaut, lorsqu'aucun jeu n'est chargé et préparé
- **STBY** : dans cet état, l'utilisateur peut régler les options du jeu (qui seront enregistrées pour un prochain lancement), ainsi que de lancer le jeu qui suivra d'abord par un décompte de lancement
- **ON** : le jeu est en cours d'avancée, il est nécessaire de détecter la fin de ce dernier ; suivant le scénario choisi, plusieurs fins sont possibles :
    - Timeout : le temps défini a été dépassé
    - Toutes : toutes les cibles encore connectées ont été touchées
    - Quelconque : une cible quelconque a été touchée
    - Bouton : le bouton externe, lié au contrôleur ou à une cible, a été appuyé
    - Groupe 1/2/3 : le groupe défini a été touché
    - Groupe 1+2 : les groupes 1 **et** 2 ont été touchés
    - Spécifique : la cible désignée a été touchée
- **OFF** : le jeu est terminé, les scores sont affichés et les cibles ne sont plus actives

Les jeux, quant à eux, peuvent s'enregistrer aux différents callbacks de cette machine, afin de réaliser des actions propres au mode choisi. Ainsi, s'enregistrer auprès de **ON_entry** permet de placer les cibles dans leur état de base au lancement du jeu, tandis que s'enregistrer auprès de **ON_do** permet de gérer l'allumage sélectif des cibles (p.ex. lors du mode de jeu aléatoire où seule une cible est allumée à la fois).

La machine d'état correspondante est la suivante :

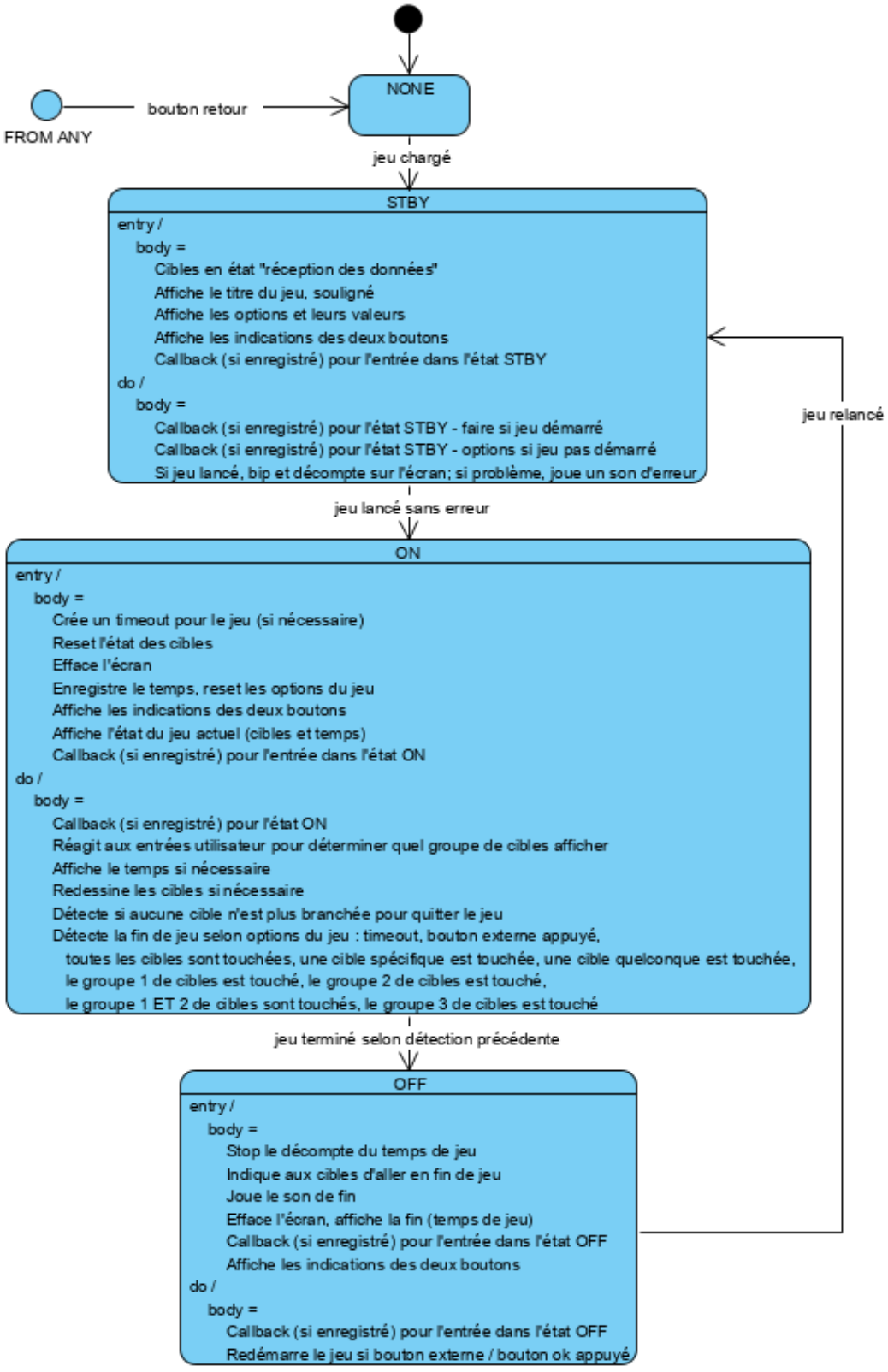


Figure 44: machine d'états - jeux

Décomposition en couches

Pour faciliter le travail du programme final, le système est séparé sur trois couches :

- **Couche hardware :** gestion des registres, abstraction des entrées/sorties
- **Interface de programmation :** fournit des méthodes simples et complètes pour l'utilisation des modules du système (ex. envoi/réception I2C …)
- **Couche logicielle :** couche supérieure, dans laquelle le logiciel manipule les différentes interfaces pour tourner le programme final

Les différents blocs de ces couches sont :

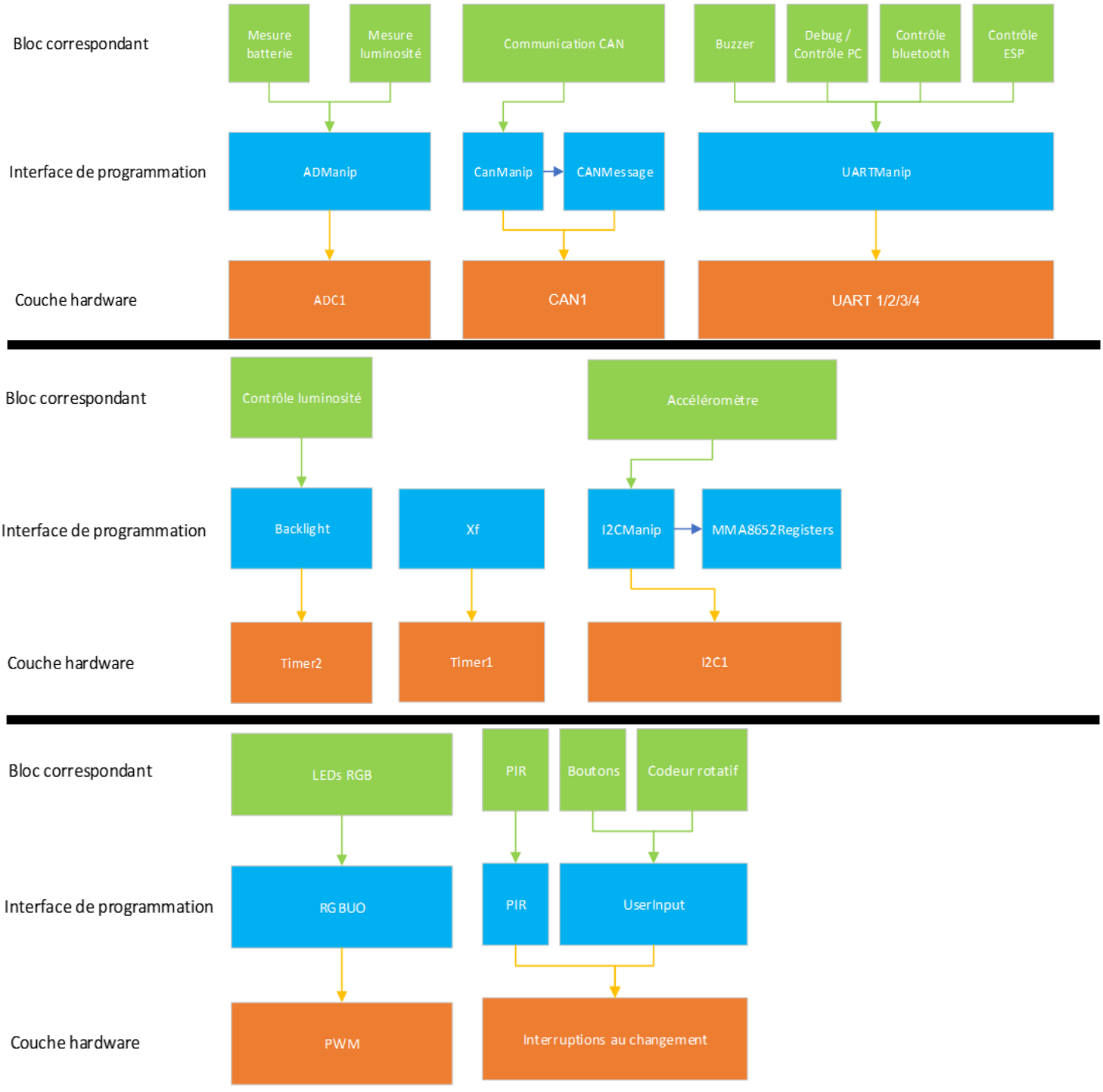


Figure 45: couches logicielles

Evènements système

Comme cité précédemment, les divers composants du système se basent sur un distributeur événementiel.

Ce dernier permet de répartir des événements en les stockant dans une liste premier arrivé – premier traité, ainsi que de gérer une quantité indéfinie de timers logiciels.

Ces événements sont retirés au niveau supérieur pour être transférés à tout le système.

Dans ce cadre, les divers événements rencontrés sont les suivants :

```
// Modules events
XF_ADC_CONVERSION_COMPLETE/*one AD conv done*/, XF_ADC_AVERAGE_CONVERSION_COMPLETE/*given serie of AD conv done*/,
XF_PRINTDEBUG_RECEIVE_COMPLETE/*complete debug frame received*/, XF_PRINTDEBUG_RECEIVE_TIMEOUT/*no char received for some time*/
XF_ESP_RECEIVE_COMPLETE/*wifi frame received*/, XF_ESP_RECEIVE_TIMEOUT/*no char received for some time*/,
XF_BT_SEND_UPDATE/*send frames following BLE specs*/, XF_BT_COMPL_RECEIPT/*BT frame received*/,
    XF_BT_RECEIVE_TIMEOUT/*no char received for some time*/, XF_BT_WAIT_TIMEOUT/*BLE chip init timeout*/,
XF_BATT_DO_MEAS/*has to measure batt channel*/, XF_BATT_MEAS_DONE/*battery measured*/,
XF_LUM_DO_MEAS/*has to measure luminosity*/, XF_LUM_MEAS_DONE/*luminosity measured*/,
XF_USERINPUT_GO_UP/*user moved encoder up*/, XF_USERINPUT_GO_DOWN/*user moved encoder down*/,
    XF_USERINPUT_OK_PRESSED/*user pressed ok button*/, XF_USERINPUT_CANCEL_PRESSED/*user pressed cancel button*/,
    XF_USERINPUT_ENCODER_PRESSED/*user pressed encoder*/, XF_USERINPUT_ENCODER_TIMEOUT/*for long press detection*/,
    XF_USERINPUT_ENCODER_LONGPRESS/*when long press detected*/,
XF_CAN_MESSAGE_RECEIVED/*new can message*/, XF_CAN_MESSAGE_LED_ON/*debug test - on message led on*/,
    XF_CAN_MESSAGE_LED_OFF/*debug test - on message led on*/, XF_WAKEFROMCAN/*when sys. woke up from can received*/,
XF_PIR_DETECTED/*PIR created interrupt*/,
XF_ACCEL_INT_PULSETRANS/*accel. created pulse/trans interrupt*/, XF_ACCEL_INT_FREEFALL/*accel created freefall interrupt*/,
XF_I2C_RECEIVED/*I2C frame received*/, XF_I2C_TRANSMITTED/*I2C frame transmitted*/,
    XF_I2C_STOP_ERROR/*I2C stopped on error*/,XF_I2C_TIMEOUT/*I2C timeout error*/,
XF_HEARTBEAT/*has to do led heartbeat*/, XF_EXT_BUTTON/*ext button pressed*/,
    XF_EXT_TIMEOUT/*for long press detect*/, XF_EXT_LONGPRESS/*ext button long press*/,
// Syscontrol hardware test events
XF_HWTEST_CHANGERGB/*change rgb color*/, XF_HWTEST_PLAYSOUND/*play a sound*/, XF_HWTEST_RESET_PULSELED/*rgb to 0*/,
    XF_HWTEST_RESET_FREEFALLLED/*freefall interrupt reset*/, XF_HWTEST_CAN_DOTEST/*send test can frame*/,
    XF_HWTEST_CHANGE_BCKL/*change backlight power*/,
// Controller machine event
XF_CTR_GOTO_OPTIONS/*move to options screen*/, XF_CTR_GOTO_GAMEMENU/*move to game list screen*/,
    XF_CTR_GOTO_SEETARGETS/*move to seetargets screen*/, XF_CTR_GOTO_CHANGEADDR/*move to changeaddr screen*/,
    XF_CTR_GOTO_INGAME/*move to ingame screen*/, XF_CTR_GOTO_CONFIRMERASE/*move to erase score screen*/,
XF_CTR_CHECKTARGETS/*check target state*/, XF_MSCR_SETMODE/*set gamemode from multi button press*/,
    XF_GAME_RELAUNCH/*relaunch game*/, XF_GAME_DIRLAUNCH/*launch game direct w.o. options*/,
    XF_RELAUNCH_GAME_TMR/*ensure user input with timer*/,
// Game machine events
XF_LAUNCHINGGAME_TIMEOUT/*countdown for game launch*/, XF_LAUNCHING_GAME_CNTDWN/*launch game with countdown*/,
    XF_GAME_TIMEOUT/*user did not finish game in time*/,
    XF_LAUNCH_GAME_AFT_CNTDWN/*launch game after countdown*/, XF_DRAWTIME/*time to draw infos for gamemode*/,
// Other
XF_TARGETSLIST_UPDATED/*targets states updated*/,
    /*for random mode, timeouts for pointers on targets !!! must be one after another !!!*/
    XF_RANDBOMB_TIMEOUT1, XF_RANDBOMB_TIMEOUT2, XF_RANDBOMB_TIMEOUT3, XF_RANDBOMB_TIMEOUT4, XF_RANDBOMB_TIMEOUT5,
    XF_WDT_CLEAR/*clear watchdog*/, XF_TEST/*test event*/, XF_CONTROLLER_SLEEP/*controller sleep*/,
// Target machine event
XF_TARGET_CAPTORTIMEOUT/*target not touched after captor int.*/, XF_TARGET_NOHEARTBEAT_TIMEOUT/*no hartbeat from ctrl - reset*/,
    XF_TARGET_ERRORTIMEOUT/*error state cleared*/, XF_TARGET_FROMERRORTOWAITINF/*back to waitinfos from error state*/,
    XF_TARGET_SLEEP/*has to sleep*/
```

Figure 46: événements système

Le code de base du distributeur événementiel a été gracieusement autorisé à la réutilisation et modification par M. Rieder Medard.

*Ont notamment été ajoutés la possibilité de réarmer les minuteurs, ainsi qu'une référence directe à la variable **TimerID**, par un pointeur, permettant de connaître l'état du timer en tout temps.*

6.2. Spécifications et algorithmes - communs

CAN

Trames

Pour réaliser une transmission sur la plus longue distance possible, la vitesse du bus est réduite au maximum. La vitesse choisie est de **50 [kbps]** – théoriquement correspondant à une distance jusque 1 [km] [14].

De plus, la vitesse de transition des signaux est abaissée matériellement, afin d'éviter la création de signaux parasites haute-fréquence.

Le module CAN du dsPIC employé est compatible avec la révision **CAN 2.0B**, signifiant que les messages avec **identificateur 29 bits** peuvent être compris et décodés.

Les trames sont représentées ainsi :

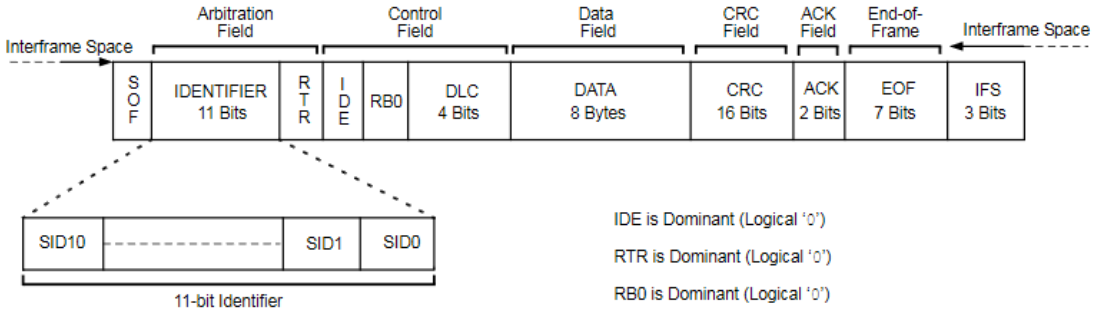


Figure 47: trame CAN basique [27]

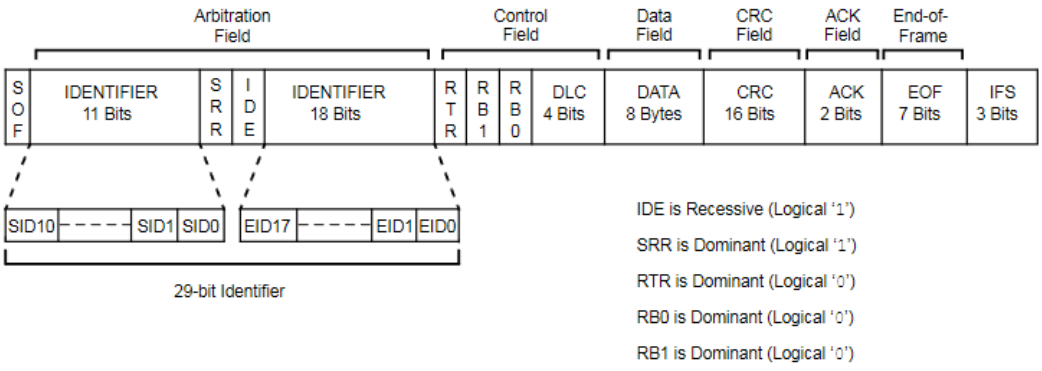


Figure 48: trame CAN étendue [27]

Pour les trames les plus complètes, on a :

- 1+11+1+1+1+4+8*8+16+2+7+3 = **111 bits pour une trame standard**
- 1+(29+2)+1+1+1+4+8*8+16+2+7+3 = **131 bits pour une trame étendue**

Ainsi, le temps de transmission d'une trame est de :

$$\frac{Nb_{frame}}{s} = \frac{Nb_{bit}}{s} * \frac{1}{Longeur_{frame_{bits}}}$$

Équation 10: nombre de frame par seconde pour bus CAN

On obtient ainsi, dans les meilleurs cas[11] :

- 450 [trames/s] @ 50 [kbps] pour une trame standard
- 381 [trames/s] @ 50 [kbps] pour une trame étendue

Suivant les spécifications suivantes, les trames étendues (29 bits d'identificateurs) sont nécessaires :

1) **8 bits d'adresse**
   a. permettant une adresse de diffusion, l'adresse du contrôleur, et jusqu'à 253 cibles théoriques
   b. **justifiés à gauche** : permet au contrôleur (adresse 0) et diffusion (adresse 1) de gagner l'arbitration
2) **8 bits de groupe**
   a. permettant à chaque cible d'avoir un groupe différent, ou commun entre-elles
   b. est utilisé notamment lors de certains modes de jeu où les cibles peuvent interagir entre-elles
   c. à la suite des bits d'adresse
3) **13 bits de commande**
   a. permettant 8'192 commandes différentes

Liste des commandes

La liste suivante présente les différentes commandes utilisées lors des transmissions et régissant le protocole de communication :

Message Name	CANCMD_HEARTBEAT
CAN ID	(ADDR << 21) | 0x00
Data[0]	Adresse de la cible
Data [1-7]	Longueur aléatoire de contenu aléatoire, si deux cibles ont la même adresse et envoient en même temps -> permet la distinction des messages
Description	Cible répond à une demande du contrôleur pour confirmer sa présence
Event Type	EVT

Message Name	CANCMD_HEARTBEAT_OK
CAN ID	(ADDR << 21) | 0x01
Description	Contrôleur confirme à la cible sa présence
Event Type	EVT

Message Name	CANCMD_ASK_HEARTBEAT
CAN ID	(ADDR << 21) | 0x02
Description	Contrôleur demande à la cible de s'identifier
Event Type	TT

Message Name	CANCMD_SLEEP
CAN ID	(ADDR << 21) | 0x03
Description	Système en sommeil
Event Type	EVT

Figure 49: commandes CAN de détection et de contrôle

[11] Sans compter un potentiel bit stuffing ou encore des erreurs lors de la transmission

Message Name	CANCMD_SET_ADDR
CAN ID	(ADDR << 21) | 0x0A
Data[0]	Adresse
Description	Définit l'adresse de la cible
Event Type	EVT

Message Name	CANCMD_SET_GROUP
CAN ID	(ADDR << 21) | 0x0B
Data[0]	Groupe
Description	Définit le groupe de la cible
Event Type	EVT

Message Name	CANCMD_SET_SENSITIVITY
CAN ID	(ADDR << 21) | 0x0C
Data[0]	Sensibilité
Description	Définit la sensibilité du système (selon variable de pré-compilation ACCEL_MAX_SENS)
Event Type	EVT

Message Name	CANCMD_SET_VOLUME
CAN ID	(BROADCAST << 21) | 0x0D
Data[0]	Volume (0 à 8)
Description	Définit le volume du système
Event Type	EVT

Message Name	CANCMD_SET_COL1
CAN ID	(ADDR << 21) | 0x0E
Data[0]	Couleur d'attente (8 bits haut - rouge)
Data[1]	Couleur d'attente (8 bits centre - vert)
Data[2]	Couleur d'attente (8 bits bas - bleu)
Data[3]	Couleur capteur (8 bits haut - rouge)
Data[4]	Couleur capteur (8 bits centre - vert)
Data[5]	Couleur capteur (8 bits bas - bleu)
Description	Définit les couleurs de la cible dans l'état d'attente et de capteur
Event Type	EVT

Message Name	CANCMD_SET_COL2
CAN ID	(ADDR << 21) | 0x0F
Data[0]	Couleur allumé (8 bits haut - rouge)
Data[1]	Couleur allumé (8 bits centre - vert)
Data[2]	Couleur allumé (8 bits bas - bleu)
Data[3]	Couleur touché (8 bits haut - rouge)
Data[4]	Couleur touché (8 bits centre - vert)
Data[5]	Couleur touché (8 bits bas - bleu)
Description	Définit les couleurs de la cible dans l'état allumé et touché

Figure 50: commandes CAN de définition de la cible

Message Name	CANCMD_GOTO_WAITINF
CAN ID	(ADDR << 21) | 0x14
Description	Envoie la cible dans l'état d'attente d'informations
Event Type	EVT

Message Name	CANCMD_GOTO_STBY
CAN ID	(ADDR << 21) | 0x15
Description	Envoie la cible dans l'état de début de jeu
Event Type	EVT

Message Name	CANCMD_GOTO_CAPTOR
CAN ID	(ADDR << 21) | 0x16
Description	Envoie la cible dans l'état d'utilisation du capteur
Event Type	EVT

Message Name	CANCMD_GOTO_ON
CAN ID	(ADDR << 21) | 0x17
Description	Envoie la cible dans l'état allumé
Event Type	EVT

Message Name	CANCMD_GOTO_ENDGAME
CAN ID	(ADDR << 21) | 0x18
Description	Envoie la cible dans l'état de fin de jeu
Event Type	EVT

Message Name	CANCMD_GROUPGOTO_ON
CAN ID	(GROUP << 13) | 0x19
Description	Envoie un groupe dans l'état allumé
Event Type	EVT

Figure 51: commandes CAN de déplacement d'état de la cible

Message Name	CANCMD_BUTTON_PRESS
CAN ID	(ADDR << 21) | 0x28
Data[0]	Nombre d'appuis sur le bouton
Description	Avertit le contrôleur de l'appui sur un bouton extérieur
Event Type	EVT

Message Name	CANCMD_BUTTON_LONGPRESS
CAN ID	(ADDR << 21) | 0x29
Description	Avertit le contrôleur d'un appui long sur un bouton extérieur
Event Type	EVT

Figure 52: commandes CAN pour bouton extérieur

Message Name	CANCMD_IMSTBY
CAN ID	(ADDR << 21) | 0x1E
Data[0]	Adresse de la cible
Description	Avertit le contrôleur du passage dans l'état d'attente
Event Type	EVT

Message Name	CANCMD_IMCAPT
CAN ID	(ADDR << 21) | 0x1F
Data[0]	Adresse de la cible
Description	Avertit le contrôleur du passage dans l'état utilisation du capteur
Event Type	EVT

Message Name	CANCMD_IMON
CAN ID	(ADDR << 21) | 0x20
Data[0]	Adresse de la cible
Description	Avertit le contrôleur du passage dans l'état allumé
Event Type	EVT

Message Name	CANCMD_IMHIT
CAN ID	(ADDR << 21) | 0x21
Data[0]	Adresse de la cible
Description	Avertit le contrôleur du passage dans l'état touché
Event Type	EVT

Message Name	CANCMD_DOUBLE_ADDRESS
CAN ID	(ADDR << 21) | 0x22
Description	Avertit la cible qu'un doublon existe
Event Type	EVT

Message Name	CANCMD_FULLCAPT
CAN ID	(ADDR << 21) | 0x23
Data[0]	Adresse de la cible
Description	Avertit le contrôleur que la cible a réalisé un temps de détection complet d'un joueur
Event Type	EVT

Figure 53: commandes CAN d'état de la cible

Message Name	CANCMD_SOUND_START
CAN ID	(ADDR << 21) | 0x32
Description	Joue le son « start »
Event Type	EVT

Message Name	CANCMD_SOUND_BEGIN
CAN ID	(ADDR << 21) | 0x33
Description	Joue le son « begin »
Event Type	EVT

Figure 54: commandes CAN de son

Message Name	CANCMD_TEST_LED_TOGGLE
CAN ID	(ADDR << 21) | 0x1F40
Data[0]	Allumé (1) ou éteint (0)
Description	Message de test, allume les leds du système
Event Type	EVT

Figure 55: commande CAN de test

Masques et filtres

Pour ne recevoir que les trames nécessaires, il est possible de définir masques et filtres tel que :

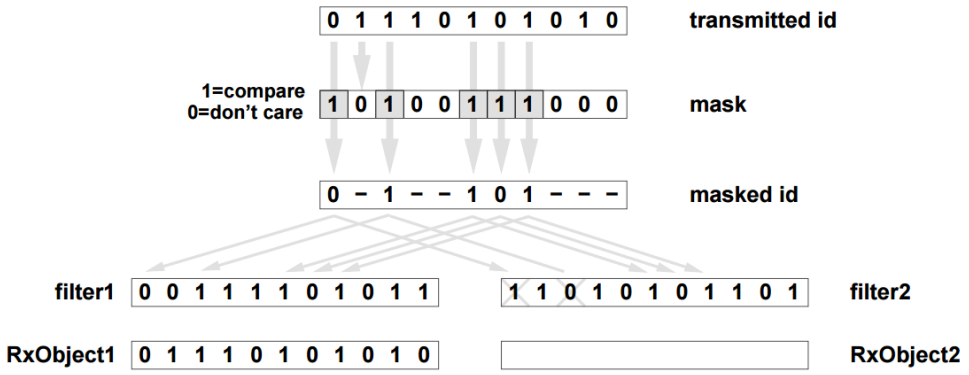


Figure 56: filtres et masques pour bus CAN, https://www.cnblogs.com/shangdawei/p/4716860.html

Le contrôleur répond à deux adresses : sa propre adresse (0), ainsi que l'adresse de diffusion (1).

La cible doit répondre à trois adresses : sa propre adresse, l'adresse de diffusion (1), ainsi que son groupe.

Les masques utilisés sont donc :

- 0b11111'11100000'00000000'00000000 – 0x1FE0'0000, **masque d'adresse (masque 0)**
- 0b00000'00011111'11100000'00000000 – 0x1F'E000, **masque de groupe (masque 1)**

Ces masques permettent de ne chercher à faire correspondre que la plage d'adresse, respectivement la plage de groupe, et d'omettre les commandes se trouvant dans les 13 derniers bits.

Trois filtres sont ensuite définis :

- L'**adresse du système**, sur 8 bits, décalé de 21 bits sur la gauche -> utilisé avec le masque 0 (**filtre 0)**
- L'**adresse de diffusion**, sur 8 bits, décalé de 21 bits sur la gauche -> utilisé avec le masque 0 (**filtre 1)**
- L'**adresse de groupe**, sur 8 bits, décalé de 13 bits sur la gauche -> utilisé avec le masque 1 (**filtre 2)**

Wi-Fi/Bluetooth/Debug

Configuration du port

Les 3 modules communiquent avec le contrôleur par UART. Les envois par byte pour les modules Wi-Fi, Bluetooth ainsi que pour le Debug (par USB) sont composés de :

- 1 bit de stop
- 8 bits de donnée
- Pas de parité / pas de contrôle de flux

Le tout est cadencé à 115'200 [bauds/s].

Trames

Pour les trames, composées de plusieurs bytes, le système assume qu'une transmission est terminée lorsque les caractères '**\r\n**' sont reçus, ou 10 [ms] après ne plus rien avoir reçu.

Gestion d'émission/réception

L'émission se base sur l'interruption d'envoi, qui intervient chaque fois qu'un caractère est transmis.
Les caractères sont tout d'abord placés dans un buffer circulaire, et vidés au fur et à mesure de la disponibilité du périphérique.

Pour la réception, les caractères sont stockés eux-aussi dans un buffer circulaire. Lorsque la réception est considérée comme complète, un événement est émis et l'utilisateur peut lire ce dernier.

Si le buffer est plein, les données sont perdues.

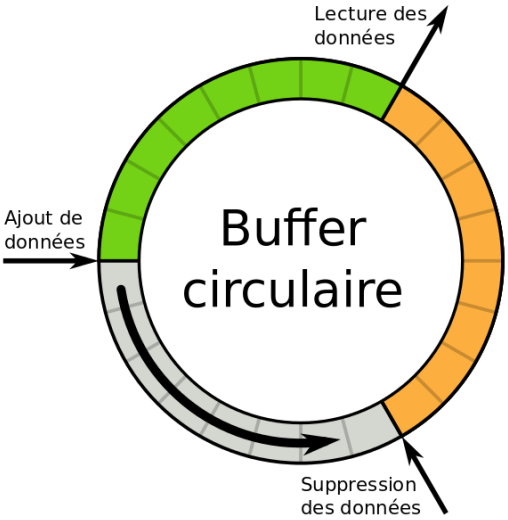


Figure 57: principe du buffer circulaire, https://upload.wikimedia.org/wikipedia/commons/thumb/2/22/Ring_buffer_fr.svg/1200px-Ring_buffer_fr.svg.png

Son

La génération du son se base sur l'UART, pour permettre l'émission de fréquences variées.

Le système détecte la vitesse de l'horloge pour adapter automatiquement les fréquences de sorties. Il suffit ainsi de créer un tableau contenant notes et temps d'émission voulu.

Si la variable **SOUND_CAN_OVERRIDE** est activée, jouer deux sons successivement verra le premier s'arrêter pour laisser place au second.
Dans le cas contraire, le deuxième son ne sera pas joué, même après la fin du premier.

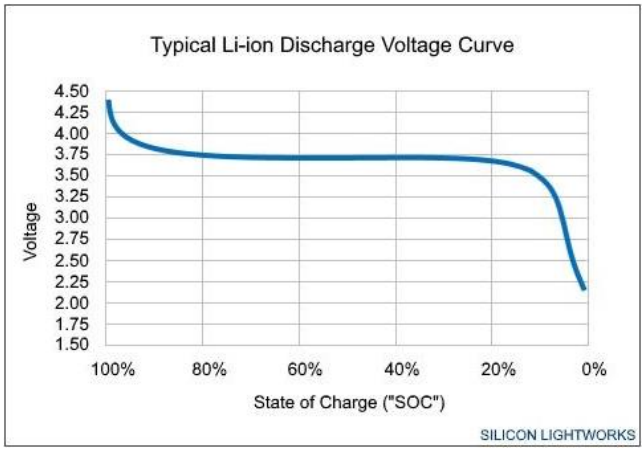
Il est ainsi facile d'ajouter des notes à la liste, selon les fréquences données ici :
https://fr.wikipedia.org/wiki/Fr%C3%A9quences_des_touches_du_piano

Batterie

La batterie est mesurée périodiquement (défini par **BATT_MEASURE_TIME_MS**, actuellement chaque 5 [s]).

Lorsque la conversion est lancée, le système continue à tourner pendant la conversion A/D, jusqu'à ce que le nombre d'itérations réglé par **BATT_MEAS_IT** soit atteint et indiqué par la génération d'un événement.

Dès lors, le convertisseur est arrêté pour économiser de l'énergie, et la moyenne des mesures précédentes réalisée, ensuite comparée dans une table de consultation.



Cette table définit le niveau estimé de la batterie, en pourcentage, par rapport à la tension mesurée. L'algorithme recherche donc deux bornes entre lesquelles la mesure se place, et linéarise l'équation pour estimer le niveau de batterie actuel.

Si le niveau est jugé trop bas (environ 3.35 [V] – 10 [%] de capacité), le système s'arrête pour s'économiser jusqu'à recharge de ce dernier.

Figure 58: Li-Ion courbe de décharge, https://siliconlightworks.com/li-ion-voltage

Sauvegarde des données en Flash

Le microcontrôleur sélectionné ne possédant pas d'EEPROM, et dans un souci d'économie aucune n'ayant été implémentée en externe, la sauvegarde des données persistantes se fait dans la mémoire Flash du microcontrôleur.
Ce type de mémoire présente le défaut d'être bien moins durable, avec un nombre d'écritures/effaçage de 10'000 cycles (minimum, avec rétention de 20 ans pour le microcontrôleur employé) contre jusqu'à 1 millions de cycles pour un EEPROM.

10'000 cycles peuvent paraître suffisants. Mais en imaginant que le système est employé chaque jour, les options modifiées au moins 20 fois, le système serait détérioré en moins de 500 jours.
Pour contrer ce problème, Microchip fournit un algorithme décrit dans l'AN1095[28] qui émule une EEPROM, tout en allongeant la durée de vie de sauvegarde à :

$$\mathrm{Total}_{\mathrm{EffectiveEndurance}} = (\mathrm{Page}_{\mathrm{Size}} - \mathrm{Page}_{\mathrm{StatusSize}} - \mathrm{Size}_{\mathrm{ofOneDataEEPROMBank}}) * \mathrm{Number}_{\mathrm{ofPages}} * \mathrm{Endurance}$$

Équation 11: durée de vie effective de l'EEPROM émulée en flash

Selon le système, les équations nous donnent :

- (512-1-255)*4*10000 = **10'240'000 [cycles] pour le contrôleur**
- (512-1-4)*2*10000 = **10'140'000 [cycles] pour la cible**

Ces valeurs prennent tout de suite meilleur sens, avec respectivement pour le scénario précédent 1402 [ans] et 270 [jours] pour le contrôleur, et 1389 [ans] et 15 [jours] pour une cible.

Bouton extérieur

Le bouton extérieur marche sur interruption et permet de détecter un appui long ou un appui court sur ce dernier.

Pour ce faire, lors d'un appui, un timer s'enclenche et contrôle régulièrement (selon **EXT_DELTA_CHECK_MS**) que le bouton est toujours appuyé.

Au bout du temps donnée par **EXT_LONGPRESS_TIME_MS** (ici 4 [s]), l'appui est considéré comme long et un événement est créé.

LEDs

Des LEDs de debug permettent d'aider à la conception du système.

Dans le code final publié, seule une de ces dernières est utilisée, pour montrer l'activité du système.

Un timer échoue régulièrement, définit à l'initialisation (ici chaque 500 [ms]) qui permet de faire clignoter une LED et prouver que le système n'est pas gelé.

Watchdog

Le système étant toujours sujet à un bug imprévu qui vienne à bloquer complètement le système, son utilisation peut être rendue impossible.

Bien sûr, il est toujours possible de retirer l'alimentation, démonter le boitier, puis retirer la batterie pour redémarrer complètement le déroulement du code.
Cela représente un coût en temps pour l'utilisateur final, tout en lui faisant perdre confiance sur le produit, tandis qu'un redémarrage inopiné peut ne même pas se remarquer.

Pour ceci, un chien de garde est implémenté. Son rôle est simple : s'il n'est pas redémarré régulièrement, le système repart de zéro.

Ce dernier est rafraîchi régulièrement à l'aide d'un timer software. Ainsi, si le système gèle à cause d'une boucle infinie, une exception quelconque (division par 0, accès à de la mémoire invalide …) … ce dernier n'est plus rafraîchit et le chien de garde échouera.

6.3. Spécifications et algorithmes – cible

Accéléromètre et I2C

L'accéléromètre marche très simplement : ce dernier est configuré et chargé pour détecter un choc, ainsi qu'une chute du boitier, et ce à l'allumage du périphérique.
La liste des registres permet un accès facilité aux données.

Par la suite, ce dernier est laissé en course libre, créant une interruption dès lors qu'un événement est détecté, relayé par un événement envoyé au système.

Ce dernier est géré par un bus I2C, qui peut marcher de façon automatique ou de façon bloquante, détectant une absence de l'esclave tout en gérant de potentielles erreurs à l'aide de différentes échéances définies par **i2cmanip**.

Détecteur de présence

Le détecteur PIR fonctionne seul, en ne gérant que son alimentation.
Une interruption, suivie de l'envoi d'un événement au système, permet de réagir en cas de détection d'un joueur.

Le détecteur nécessite une minute de mise en route avant d'être utilisable, c'est pourquoi il n'est pas éteint lors d'une mise en sommeil.

RGB

Les LEDs extérieures au boitier permettent de démontrer l'état actuel de la cible.

Pour afficher une palette importante, chaque couleur primaire est gérée par une PWM indépendante.
Pour économiser de l'énergie, le module est désactivé complètement lorsque la LED doit rester à 0.

Le reste du temps, la période est fixée à 3.75 [ms] – 266 [Hz], ce qui permet de ne remarquer aucun clignotement.

Les incréments sont définis en pourcent par l'interface fournie (0 – 100), bien que le système soit capable de plus de finesse si nécessaire.

Des méthodes pour travailler en code RGB sont directement fournies.

6.4. Spécifications et algorithmes – contrôleur

Boutons et encodeurs

Les différents boutons (ok, annuler, encodeur) sont liés aux IOC – interruption au changement -. Ces dernières détectent les appuis et relâchements des différents boutons.

Seuls les appuis génèrent des événements, relayés au système au travers du distributeur.

Pour la rotation de l'encodeur, cette dernière se base sur le principe d'une interface de codeur incrémental en quadrature :

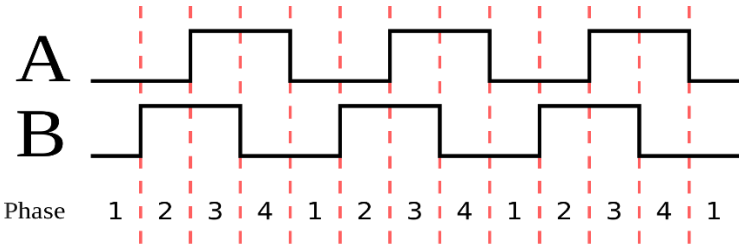


Figure 59: signaux d'un module QEI, https://en.wikipedia.org/wiki/Incremental_encoder#/media/File:Quadrature_Diagram.svg

Avec ça, il est possible de déterminer le sens de rotation du codeur, en plus de sa vitesse et position. Le microcontrôleur sélectionné possède un module dédié, mais n'est pas employé à cette fin. En effet, la lenteur des signaux permet l'utilisation simple d'IOCs dans un **mode x1** (nombre d'impulsions par cran physique de la molette).

Il est donc possible de travailler en mode x1, x2 et x4 (nombre d'impulsions par cran). Pour sélectionner ces modes, il suffit :

- **En x1**, de regarder les changements d'un seul canal, sur un seul flanc défini
- **En x2**, de regarder les changements d'un seul canal, sur ses deux flancs **OU** sur un flanc défini de chaque canal
- **En x4**, de regarder les changements des deux canaux, sur leurs deux flancs

Dans notre cas, seul le mode **x1** fait sens. En effet, pour chaque cran fait physiquement, l'utilisateur ne veut déplacer son curseur que d'une position.

Pour ça, seul le canal A est utilisé, sur son flanc montant, créant donc une interruption dès lors, et ce suivant la direction de l'encodeur. Les événements correspondants sont distribués (**XF_USERINPUT_GO_DOWN** ou **XF_USERINPUT_GO_UP**).

Affichage - luminosité

La luminosité est lue périodiquement à une fréquence définie par **LUM_MEAS_TIME_MS** (ici 200 [ms]).

Cette dernière est comparée aux mesures extrêmes (prises en absence de lumière et en présence de beaucoup de lumière), et une linéarisation est faites entre ces deux points pour faire correspondre l'intensité du rétroéclairage à appliquer.

Cette intensité est au minimum de **MIN_LUM** [%] (ici 1).

La lecture A/D s'effectue à plusieurs reprises, de façon non-bloquante, et applique automatiquement l'intensité désirée au rétro-éclairage.

Pour donner un effet moins abrupt au changement, l'éclairage est modifié périodiquement, petit à petit, pour rejoindre la consigne.

Affichage – rétro-éclairage

Le rétro-éclairage marche sur PWM, dont la période est de 100 [Hz] (pas de scintillement de l'écran) et qui s'incrémente par pourcent de l'éclairage total.

Affichage – graphique

Les éléments graphiques et la gestion bas-niveau de l'écran sont deux systèmes gracieusement mis à disposition par M. Sartoretti Pascal, sous licence libre.

L'envoi et la réception des commandes marche de manière parallèle (8 bits), et sont bloquants.

La bibliothèque graphique permet l'affichage de bitmaps, de textes, de lignes/rectangles, de boutons et de sliders.
Y ont été ajouté la possibilité de dessiner des ronds, ainsi que de transformer une couleur RGB (24 bits) en High-Color (15 bits) – système de couleur employé par l'écran.

6.5. Jeux et logiques

Principes

8 modes de jeu sont actuellement implémentés :

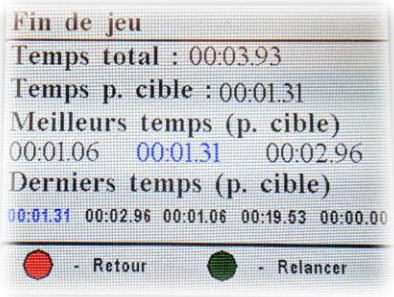


Figure 60: écrans de fin parcours, aléatoires, bombe

- **Parcours**
  - toutes les cibles sont allumées ; le but est de toucher les cibles à la suite en suivant un parcours prédéfini
  - le jeu se termine lorsque toutes les cibles sont touchées, ou le temps réglé dépassé
- **Parcours + capteur**
  - toutes les cibles sont allumées ; le but est de toucher les cibles à la suite en suivant un parcours prédéfini
  - si l'utilisateur est repéré par un capteur, la cible clignote le temps défini ; si elle n'est pas touchée à temps, un malus est compté
  - le jeu se termine lorsque toutes les cibles sont touchées, ou le temps réglé dépassé, ou le bouton extérieur pressé (si des cibles ne sont pas touchées -> malus supplémentaires), ou la cible désignée touchée (si des cibles ne sont pas touchées -> malus supplémentaire)

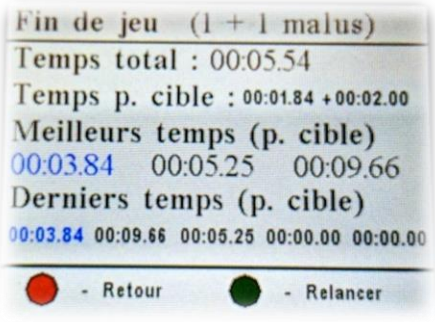


Figure 61: écran de fin parcours + capteur

- **Aléatoire mobile**
  - les cibles s'allument aléatoirement et changent si elles ne sont pas touchées dans le temps réglé, de 1 à 5 cibles simultanément
  - le jeu se termine lorsque toutes les cibles sont touchées, ou le temps réglé dépassé
- **Aléatoire fixe**
  - semblable à l'aléatoire mobile, mais la cible reste allumée jusqu'à être touchée
- **Bombe**
  - semblable à l'aléatoire mobile, mais si la cible n'est pas touchée à temps -> le jeu est perdu

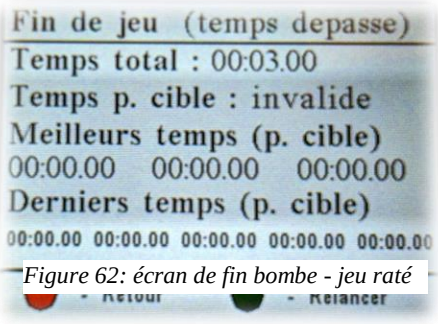


Figure 62: écran de fin bombe - jeu raté

- **Par pièces**
  - les cibles s'allument par lot selon réglage (1 à 50 cibles par pièce), simulant des pièces ; la prochaine « pièce » n'est allumée que lorsque toutes les cibles précédentes sont touchées
  - le jeu se termine lorsque toutes les pièces sont nettoyées (toutes les cibles touchées)

- **Par pièces + capteur**
  - les cibles s'allument par lot selon réglage
  - si l'utilisateur est repéré par un capteur, la cible clignote le temps défini ; si elle n'est pas touchée à temps, un malus est compté et toutes les cibles de la pièce s'allument
  - le jeu se termine lorsque toutes les cibles sont touchées, ou le temps réglé dépassé, ou le bouton extérieur pressé (si des cibles ne sont pas touchées -> malus supplémentaire)

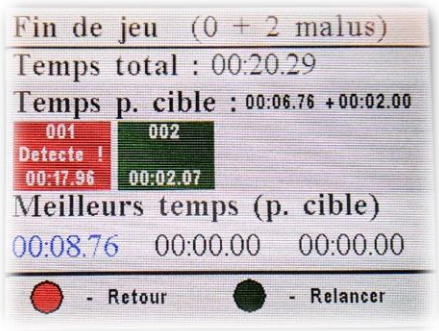


Figure 63: écran de fin "par pièces", avec une pièce détectée

- **Duel (arbre)**
  - les cibles désignées s'allument de deux couleurs, représentant les joueurs 1 et 2
  - le jeu se termine lorsque toutes les cibles sont touchées ; les temps du joueur 1 et du joueur 2 sont comptés indépendamment

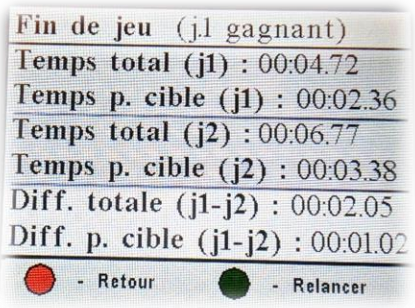


Figure 64: écran de fin duel (arbre)

- **Duel (rapidité)**
  - toutes les cibles s'allument
  - le jeu se termine lorsque n'importe quelle cible est touchée

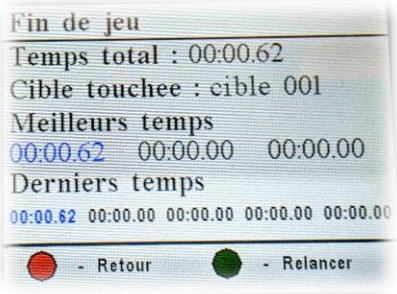


Figure 65: écran de fin duel (rapidité)

Implémentation

Parcours

Les deux modes du parcours s'enregistrent aux callbacks suivants :

- **Entrée du jeu**
  - Lance toutes les cibles dans le mode **ON** (parcours) ou **CAPTEUR** (parcours + capteur)
  - Enregistre les options
- **Déroulement du jeu**
  - Pour le mode parcours + capteur, détecte lorsqu'une cible a détecté un joueur et compte un malus
- **Entrée de fin de jeu**
  - Enregistre le score actuel si valide (pas de dépassement de temps – en comptant les malus pour le mode parcours + capteur), enregistre si un des 3 meilleurs temps, affiche les scores

Aléatoires et bombe

Les trois modes s'enregistrent aux callbacks suivants :

- **Entrée du jeu**
  - Lance toutes les cibles dans le mode **STBY**
  - Mélange les cibles
  - Enregistre les options
- **Déroulement du jeu**
  - Cherche si une cible doit être allumée ou changée, selon le temps défini par l'utilisateur ou si la cible est touchée
  - Pour le mode bombe, arrête le jeu si une cible n'est pas touchée à temps
- **Entrée de fin de jeu**
  - Enregistre le score actuel si valide (pas de dépassement de temps – en comptant les malus pour le mode parcours + capteur), enregistre si un des 3 meilleurs temps, affiche les scores

Le mélange des cibles se base sur un algorithme pour permuter les éléments du tableau, par la procédure **static void randTargets(TargetState_t **arr, uint8_t n)** :

- Le tableau est parcouru de la fin jusqu'au second élément
- A chaque itération, l'élément est échangé « aléatoirement » avec un des éléments avant lui (d'où le fait de ne pas traiter le premier élément de la liste)
- *La graine de l'aléatoire est redéfinie au lancement du jeu, selon le temps de fonctionnement du système*

Pour la gestion des 1 à 5 cibles aléatoires simultanées, l'algorithme de la procédure **static void actOnChange(TargetState_t **arr, uint8_t n, RandBombTargetPointer_t *pointers, Event ev)** est le suivant :

- 1 à 5 pointeurs sont activés, se déplaçant linéairement dans la liste mélangée des cibles Un pointeur désactivé ne pointe vers aucune cible et ne fait rien
- Au départ, les pointeurs cherchent une cible ne possédant pas encore de pointeur, qui n'est pas touchée et qui est toujours connectée -> la cible est activée, et un minutage propre au pointeur est lancé (temps défini par l'utilisateur)
- Lorsque le temps du pointeur est dépassé, ce dernier recherche une nouvelle cible dans la liste ne possédant pas de pointeur ; si aucune cible n'est libre, le pointeur se désactive sur sa cible actuelle, signifiant qu'elle restera active jusqu'à être touchée
- Lorsque la cible est touchée, le pointeur cherche une nouvelle cible ; si aucune n'est libre, le pointeur se désactive
- Lorsque tous les pointeurs sont désactivés, cela signifie la fin du jeu

Lorsque le timer d'un pointeur échoue, ce dernier envoie un événement pour avertir le gestionnaire de jeu.
Les événements sont **XF_RANDBOMB_TIMEOUTx**, ici de 1 à 5. Pour activer plus de pointeurs, il suffit d'augmenter le nombre possible **RANDBOMB_MAXSIMULT_TARG**, tout en réservant **A LA SUITE** les événements **XF_RANDBOMB_TIMEOUTx** correspondants (basé sur le fait qu'un **enum** augmente la valeur de +1 à chaque définition).

Par pièces

Les deux modes s'enregistrent aux callbacks suivants :

- **Entrée du jeu**
  - Lance toutes les cibles dans le mode **STBY**
  - Cherche la première pièce contenant des cibles et l'active
  - Enregistre les options
- **Déroulement du jeu**
  - Pour le mode par pièces + capteur, détecte lorsqu'une cible a détecté un joueur et compte un malus
  - Contrôle la pièce actuelle, et au besoin en cherche une nouvelle
- **Entrée de fin de jeu**
  - Enregistre le score actuel si valide (pas de dépassement de temps – en comptant les malus pour le mode avec capteur), enregistre si un des 3 meilleurs temps, affiche les scores, affiche les chambres (en rouge si les joueurs ont été détectés)
- **Déroulement de fin de jeu**
  - Récupère les entrées de l'encodeur, pour défiler les pièces si nécessaire

La procédure pour activer les chambres, **static void manageRoom(uint8_t *currroom, uint8_t targperroom, uint8_t maxtarg, TargetState_t *targets, bool first, GameType_t gametype)**, se base sur l'algorithme suivant :

- Si l'argument first est vrai, la procédure recherche une salle vide depuis la pièce actuelle
  si une pièce est trouvée, les cibles de cette pièce sont allumées
- Si l'argument est faux, la procédure contrôle si la salle actuelle ne comporte que des cibles débranchées ou touchées, et si tel est le cas, la procédure est rappelée avec les mêmes arguments, mais avec first à vrai pour trouver une nouvelle salle
- Si aucune salle n'est disponible, la procédure ne fait rien

Duels

Les deux modes s'enregistrent aux callbacks suivants :

- **Entrée du jeu**
  - Allume les cibles nécessaires : toutes pour le mode rapide, celles du joueur 1 et 2 pour le mode arbre
  - Enregistre les options
- **Déroulement du jeu**
  - Dans le mode arbre, détecte lorsqu'un des deux joueurs a touché toutes ses cibles, et enregistre son temps
- **Entrée de fin de jeu**
  - Contrôle que les deux joueurs ont un temps enregistré (sinon enregistre les temps manquants)
  - Affiche les scores
  - Dans le mode rapide, enregistre le score actuel si valide (pas de dépassement de temps – en comptant les malus pour le mode avec capteur), enregistre si un des 3 meilleurs temps, affiche les scores

7. Système final

7.1. Tests

L'ensemble des tests effectués peut être retrouvé en annexe *10.5 Tests de fonctionnement.*

7.2. Consommation

Principe

Le système tournant sur batterie, un point est mis sur la gestion de la consommation. En outre, le système peut être mis en mode sommeil :

4) Par appui long sur le bouton extérieur (branché sur le contrôleur ou une cible)
5) Après 3 minutes sans interaction de l'utilisateur, pour autant qu'un jeu n'est pas lancé

Le reste du temps, les modules sont activés au mieux uniquement lorsque nécessaire (A/D uniquement lors de la lecture de la batterie/détecteur de luminosité, PWMs activées indépendamment lorsque les niveaux de couleur ne sont pas 0 …).

Mesure

Le système a été mesuré à son maximum (lors des jeux) et lorsqu'il est à l'arrêt, alimenté en 5 [V] (chargeur externe), avec les résultats suivants :

MODULE	CONSOMMATION
Cible (maximum)	241 [mA]
Cible (sommeil)	2.72 [mA]
Contrôleur (maximum)	220 [mA]
Contrôleur (sommeil)	3.65 [mA]

Tableau 13: consommation du système sur secteur

A noter que dans ce cadre de test, les 3 LEDs d'indication de la batterie s'allument. Ainsi, lors de l'utilisation sur batterie, il faut retirer à chaque mesure leur consommation de **0.77 [mA]** :

MODULE	CONSOMMATION
Cible (maximum)	240.23 [mA]
Cible (sommeil)	1.95 [mA]
Contrôleur (maximum)	219.23 [mA]
Contrôleur (sommeil)	2.88 [mA]

Tableau 14: consommation du système sur batterie

Le courant latent **notable** de sommeil provient des modules suivants :

MODULE	CONSOMMATION
Régulateur LDO	1.2 [mA]
MAX3051 (CAN)	0.015 [mA]
MCP73833 (Protec batt.)	0.1 [mA]
FT230 (UART-USB)	0.25 [mA]
Oscillateur	0.01 [mA]
Cible - Accéléromètre	0.05 [mA]
Cible - PIR	0.07 [mA]
Contrôleur – mesure lumière	Max. 1 [mA]
TOTAL CIBLE	1.695 [mA]
TOTAL CONTRÔLEUR	2.575 [mA]

Tableau 15: courants de sommeil

Améliorations

On voit grâce au tableau précédent les 3 éléments les plus consommateurs du circuit.

1) **Régulateur LDO**

Pour le régulateur, ce dernier avait été choisi pour fournir un courant assez important et ainsi ne pas limiter le système en cas de besoin. Après les mesures précédentes, on peut déterminer qu'un courant maximal d'un tiers de celui fournit par le LDO choisit (500 [mA] au lieu de 1.5 [A]) est suffisant. Ainsi, un LDO plus petit consommerait moins en tout temps.

2) **FT230**

Le FT230 prévu n'est actuellement là qu'à titre d'aide à la conception, et sa consommation pourrait ainsi ne pas être comptée dans les versions grand public.
Si ce dernier est gardé pour le contrôleur dans sa version finale (contrôle du système à l'aide d'un PC par exemple), sa consommation devrait être régulée.

3) **Mesure de lumière**

La mesure de lumière est un simple pont résistif dont l'une des branches varie en fonction de la luminosité ambiante.

Actuellement, le pont consomme en permanence, même lorsque la luminosité n'est pas lue.

Deux solutions sont envisageables :

1) Augmenter la résistance de la branche basse (actuellement 3.3 [kOhms]), au détriment d'une bonne résolution
2) Contrôler la mise à la masse de la branche basse, ou l'alimentation de la branche haute

Selon le schéma, des entrées/sorties du microcontrôleur sont encore disponibles. La seconde solution est donc à privilégier, en pouvant consommer uniquement lors de la mesure.

7.3. Ecrans de jeu

Se référant à la *Figure 43: machine globale - contrôleur* , les différents écrans du contrôleur sont les suivants :

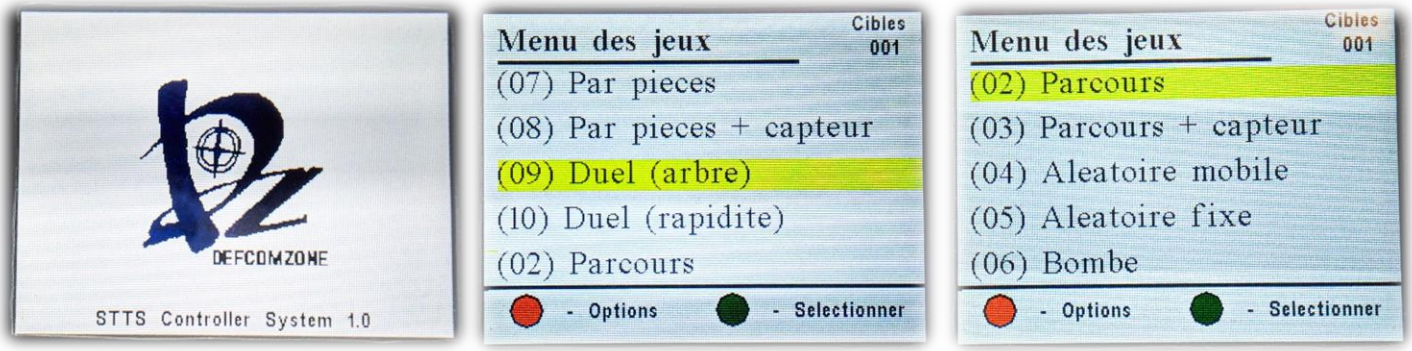


Figure 66: menu d'accueil et menu des jeux

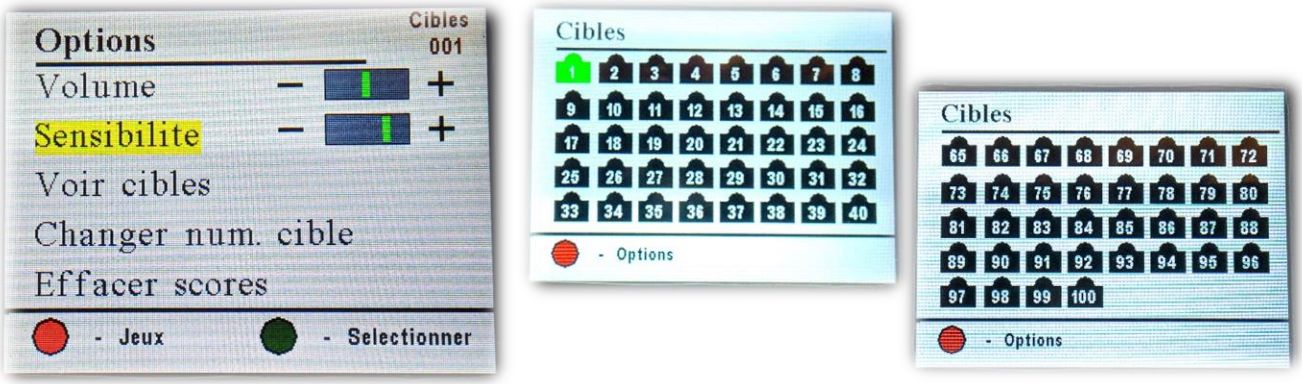


Figure 67: options et liste des cibles

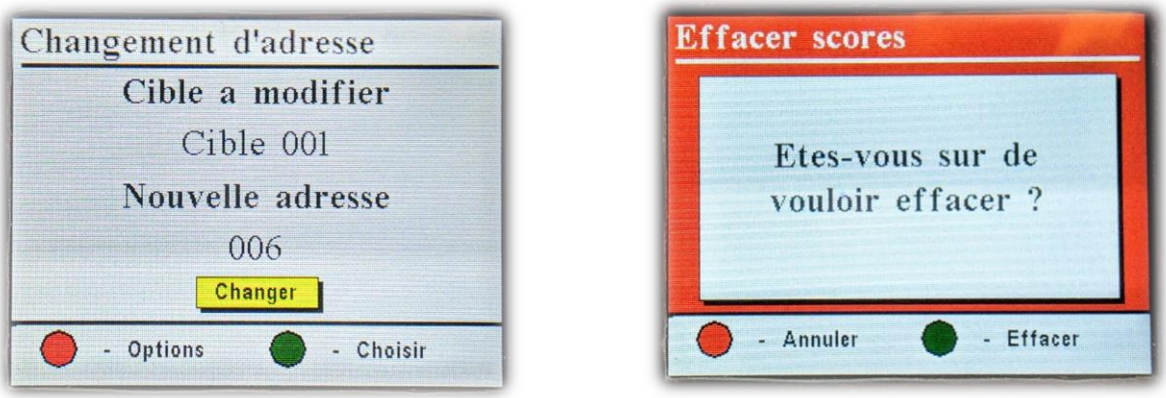


Figure 68: changement d'une adresse et effaçage des scores

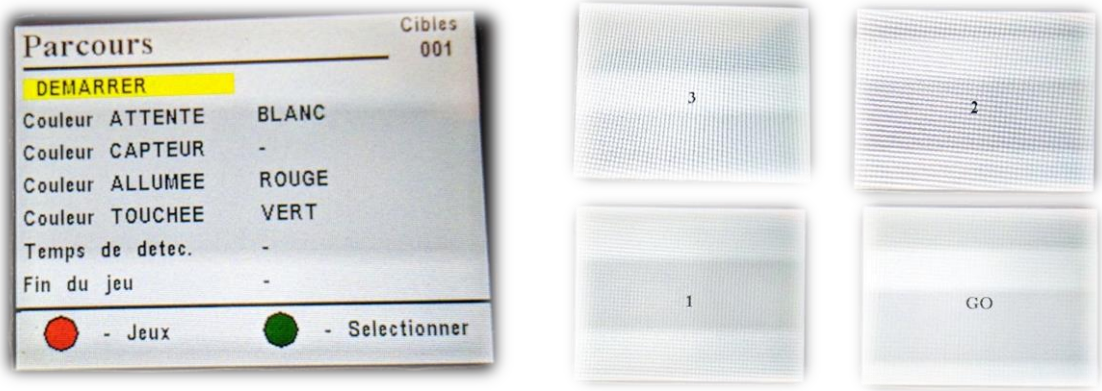


Figure 69: options d'un jeu et lancement

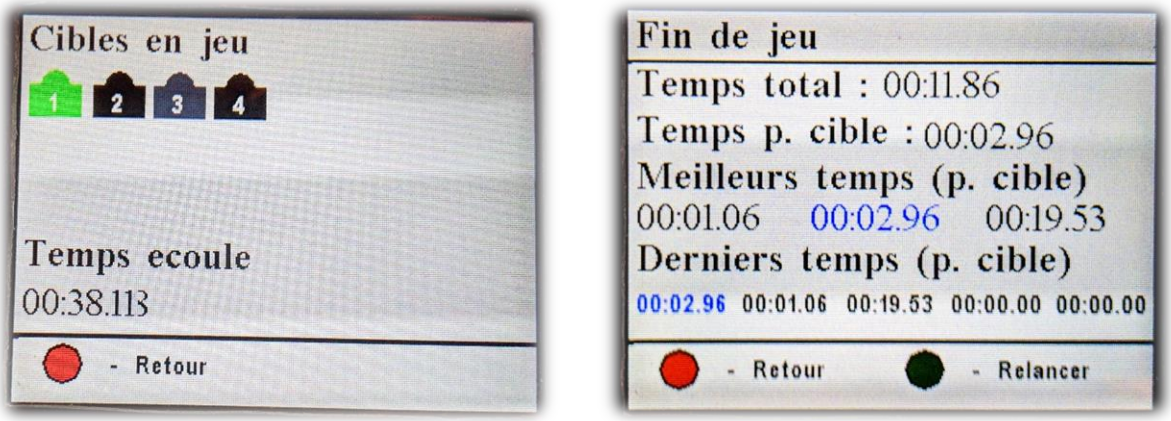


Figure 70: en jeu et fin de jeu

8. Boitiers

8.1. Principe

Dans l'optique d'offrir un système complet, des boitiers fonctionnels sont développés pour chaque module. Il est ainsi possible d'entrevoir la place nécessaire quant au stockage du système et pouvoir tester le produit en conditions réelles.

Les boitiers sont réalisés dans un esprit de simplicité, tout en étant suffisamment robustes et intégrant les moyens de fixation nécessaires.

8.2. Boitier cible

Le boitier pour les cibles se divise en deux parties, et se présente ainsi :

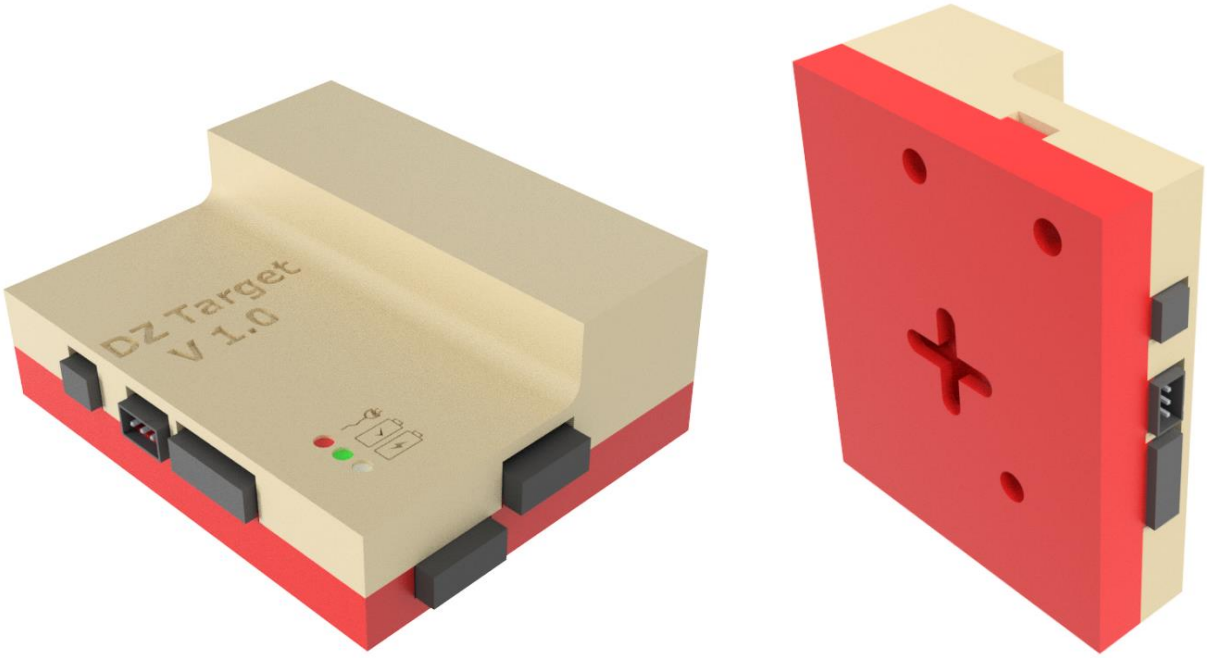


Figure 71: ensemble - cible

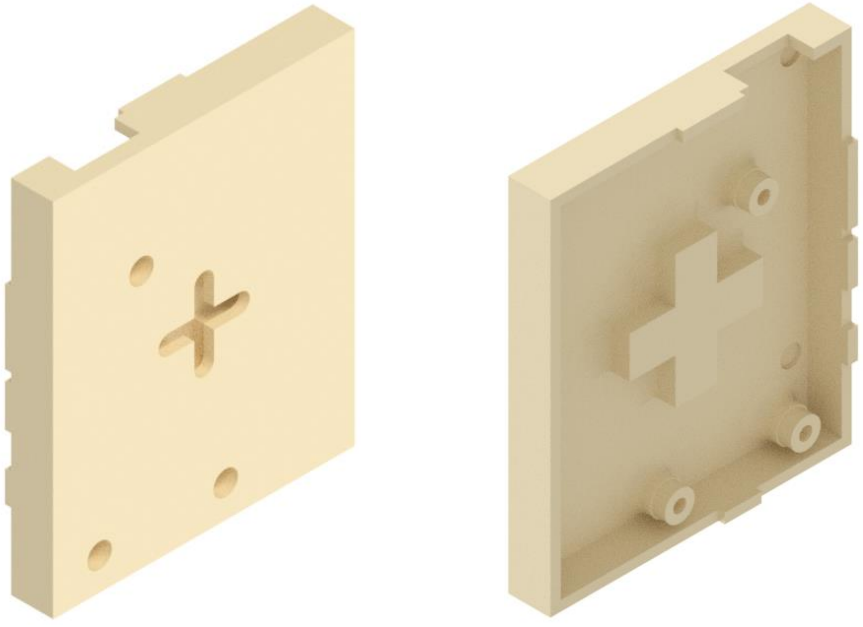


Figure 72: boitier inférieur - cible

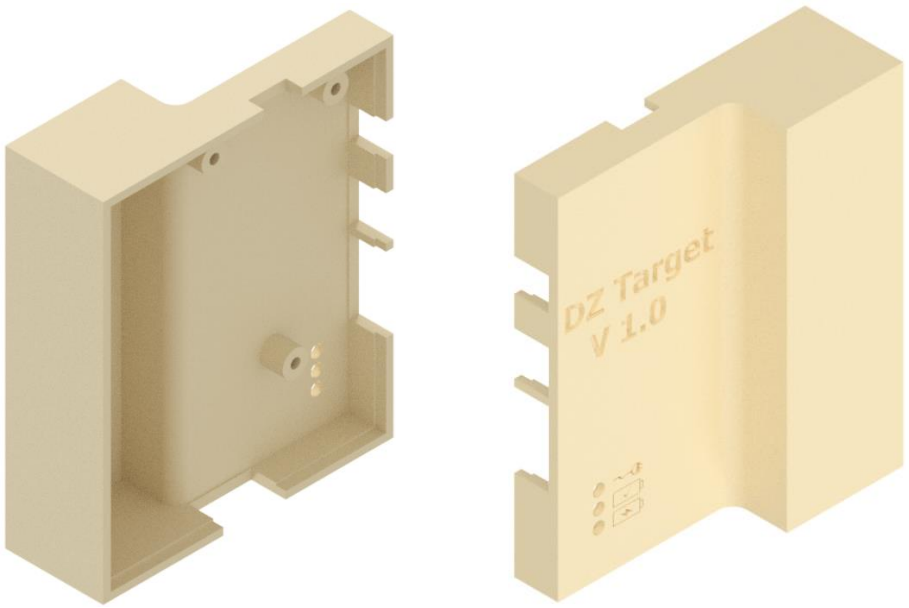


Figure 73: boitier supérieur – cible

Sur la face avant, on peut y lire le nom du module, ainsi que les trois LEDs d'indication e charge avec leur logo respectif (de haut en bas : alimentation branchée – charge pleine – en charge). Lorsque le système est branché sur secteur sans batterie, les trois LEDs sont allumées.

Tout autour de ce dernier se trouvent des encoches pour les divers connecteurs, avec sur le haut une encoche discrète pour le branchement du câble USB de débogage.

Le boitier est fait pour garder un profil minimal, tout en étant possédant des angles droits sur ses bords pour pouvoir les empiler plus facilement,

Le boitier est fermé par trois vis M3, passant au travers du PCB.

Ce dernier peut être fixé de trois manières différentes :

3) A l'aide de scratch disposé sur sa face arrière (plane)
4) A l'aide d'une vis, dont la tête vient s'insérer dans l'encoche fournie (4 positions possibles)
5) A l'aide d'aimants, noyés dans le boitier – *après tests, les aimants prévus sont trop faibles pour supporter le poids de l'ensemble*

Coût

Le boitier test est imprimé sur une machine SLS en PLA noir.
Un coût potentiel chez un professionnel local - https://a-printer.ch/pieces/ :

Impression : 20.- de prise en charge puis entre 0.50 et 1.- CHF le gramme selon la matière et la complexité

Avec l'ensemble estimé à 73 [g] (Ultimaker Cura 4.6), le prix de revient est de 56.5 CHF à 93 CHF.

8.3. Module détecteur de présence

Le module détecteur de présence est indépendant à la cible pour être branché au bon vouloir de l'utilisateur final.

Dans cette optique, un boitier indépendant lui est dédié, qui sera fixé au bon vouloir du client :

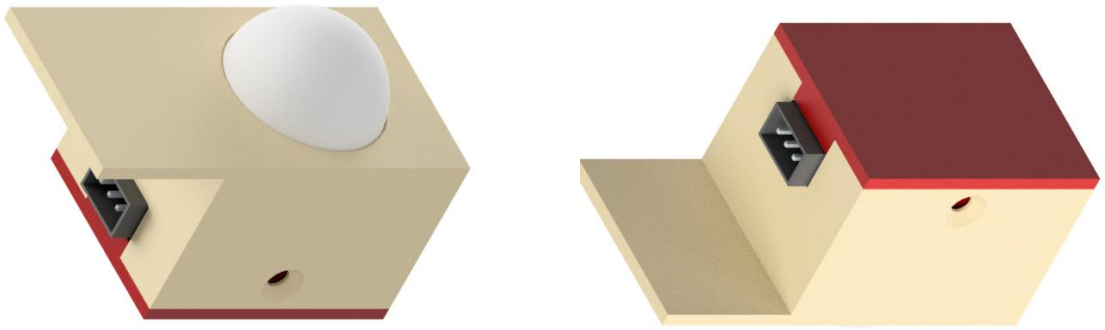


Figure 74: ensemble - détecteur de présence

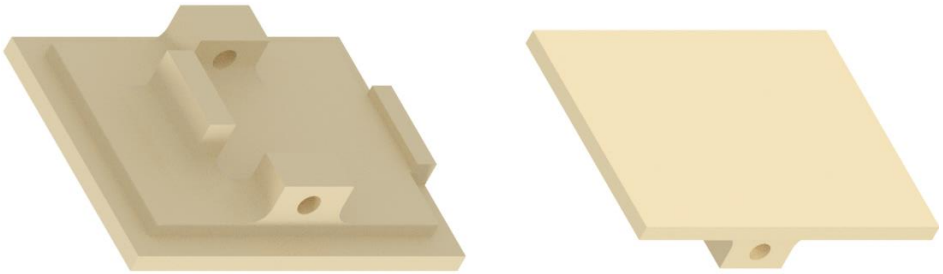


Figure 75: boitier inférieur - détecteur de présence

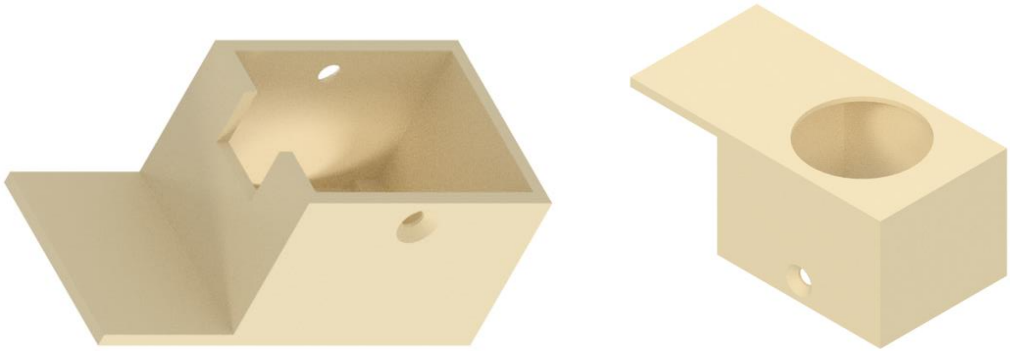


Figure 76: boitier supérieur - détecteur de présence

Le boitier possède protubérance sur le dessus, permettant à la fois de protéger le connecteur juste dessous, ainsi que de permettre sa fixation (vis ou scratch).

Le boitier est fermé par deux vis M3.

Coût

Le boitier test est imprimé sur une machine SLS en PLA noir.
Un coût potentiel chez un professionnel local - https://a-printer.ch/pieces/ :

Impression : 20.- de prise en charge puis entre 0.50 et 1.- CHF le gramme selon la matière et la complexité

L'ensemble estimé à 16 [g] (Ultimaker Cura 4.6), le prix de revient est de 28 CHF à 36 CHF.

8.4. Module LEDs et Buzzer

Le module « LEDs et Buzzer » est indépendant à la cible.
Dans cette optique, un boitier indépendant lui est dédié, qui sera fixé à l'endroit désiré :

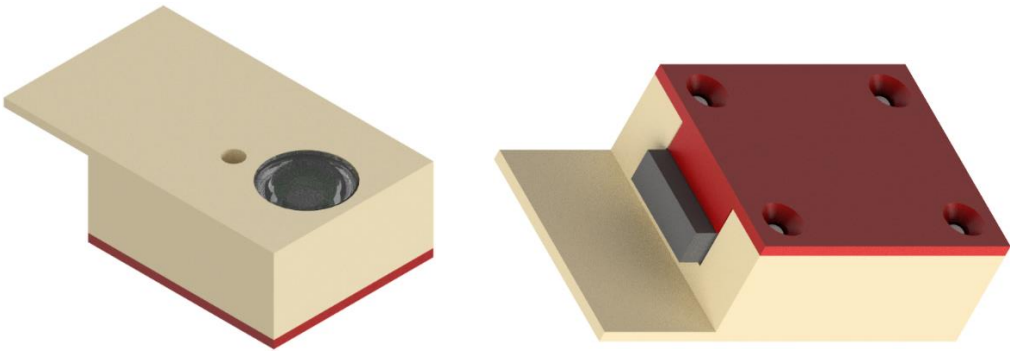


Figure 77: ensemble - leds et buzzer

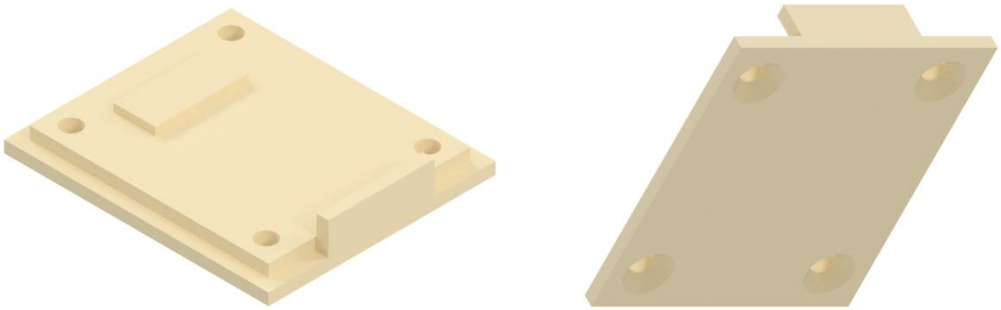


Figure 78: boitier inférieur - leds et buzzer

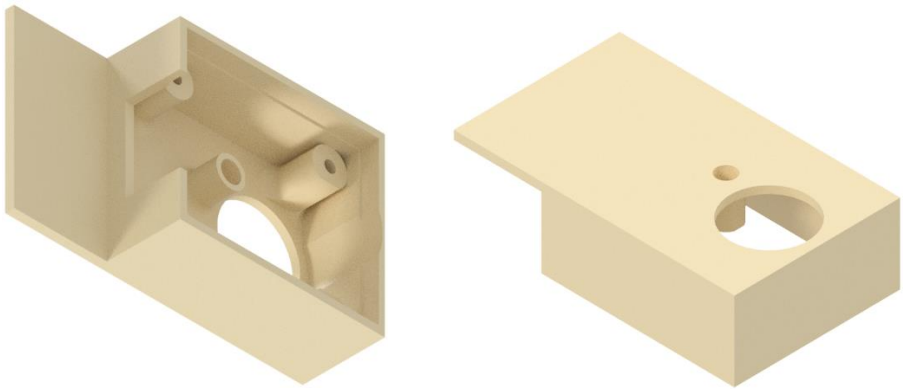


Figure 79: boitier supérieur - leds et buzzer

Le boitier est fermé par 4 vis M3.
Le diffuseur possède son emplacement, tandis qu'un « tube » conduit le son hors du boitier.
Comme précédemment, une protubérance protège le connecteur et permet la fixation.

Coût

Le boitier test est imprimé sur une machine SLS en PLA noir.
Un coût potentiel chez un professionnel local - https://a-printer.ch/pieces/ :

Impression : 20.- de prise en charge puis entre 0.50 et 1.- CHF le gramme selon la matière et la complexité

L'ensemble estimé à 20 [g] (Ultimaker Cura 4.6), le prix de revient est de 30 CHF à 40 CHF.

8.5. Contrôleur

Le contrôleur a été pensé pour une prise en main facilitée, avec un fond arrondi et plus large que ne nécessiterait la taille de l'électronique :

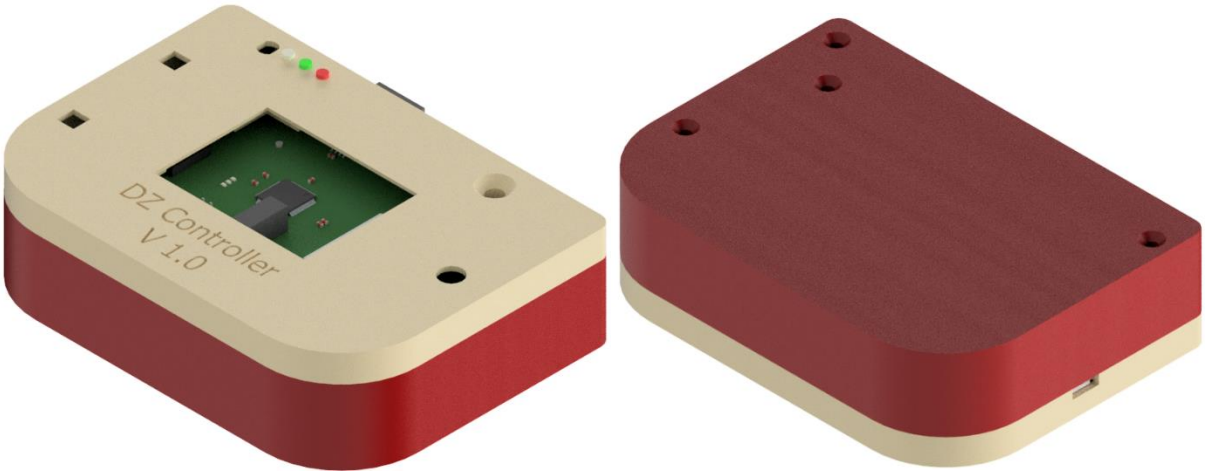


Figure 80: ensemble – contrôleur

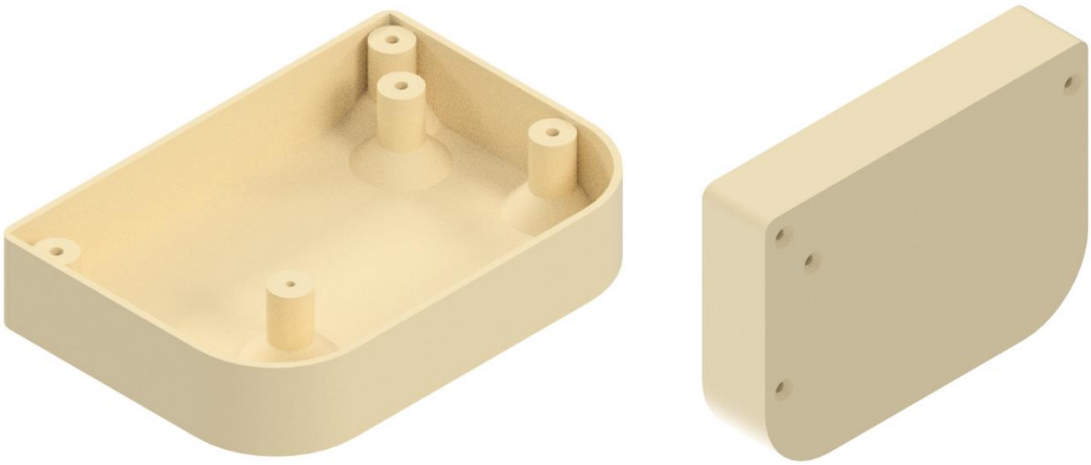


Figure 81: boitier inférieur – contrôleur

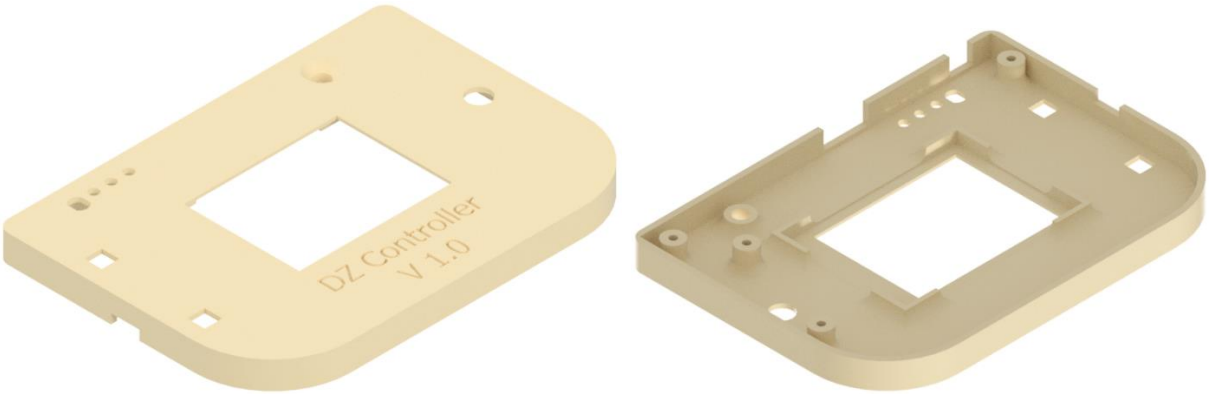


Figure 82: boitier supérieur – contrôleur

Le boitier supérieur accueille un emplacement pour l'écran, les trous pour les deux boutons ainsi que l'encodeur.

Les trois LEDs de statut de charge sont présentent au côté du capteur de luminosité.

On peut lire sur la face avant le nom du module, et similaire au module « LEDs et buzzer » un tube guide le son vers l'extérieur.

Un morceau de mousse est inséré entre l'écran et l'électronique pour éviter les courts-circuits et maintenir l'écran en place.

Le boitier est maintenu par 4 vis traversant le PCB. Un montant hors du PCB (dans la partie vide du boitier) est nécessaire à l'ajout pour un bon maintien du circuit.

Coût

Le boitier test est imprimé sur une machine SLS en PLA noir.
Un coût potentiel chez un professionnel local - https://a-printer.ch/pieces/ :

Impression : 20.- de prise en charge puis entre 0.50 et 1.- CHF le gramme selon la matière et la complexité

Avec l'ensemble estimé à 92 [g] (Ultimaker Cura 4.6), le prix de revient est de 66 CHF à 112 CHF.

8.6. Coût total et production

Le coût total (bas) des boitiers faits en impression SLS selon le fournisseur cité plus haut, pour 5 cibles, 5*2 modules et 1 contrôleur s'élèverait donc à 638.5 .-, coût bien trop important comparé au reste du système.

Une fois le boitier confirmé en impression 3D, il serait plus intéressant de partir sur une série basée sur du moulage/injection plastique. En plus de nécessiter moins de post-traitement, le rendu est uniforme et inspire plus confiance à l'utilisateur final.

Aucun devis n'ayant été demandé, il est impossible de donner une estimation du prix final.

Aucun usinage n'ayant été effectué pour la fabrication des pièces de test (impression 3D et post-traitement uniquement), aucun plan des pièces n'est réalisé.

8.7. Ensemble de test

Pour le test, un contrôleur, 5 cibles, 3 détecteurs de mouvements et 5 modules LEDs ont été fabriqués :

Figure 83: ensemble de test

9. Conclusion

9.1. Objectifs et complétion

Le meilleur moyen pour prouver le fonctionnement du système est d'en reprendre les grands axes. Le but était de concevoir un système de ciblerie.

Ce dernier est nomade, capable de fonctionner directement avec une batterie, comprenant une durée de jeu d'env. 16 [h], contre 112 [jours] en sommeil ; ou alors indéfiniment, relié au secteur au travers de blocs d'alimentation 5 [V].

Un contrôleur permet de régler les différentes options des jeux, bien qu'il soit possible de lancer rapidement les scénarios pré-enregistrés de partout avec un simple bouton.

En parlant de jeux, des scénarios simples (toutes les cibles s'allument) comme complexes (allumages séquencés avec utilisation de capteurs de présence) sont disponibles, pour convenir aux deux marchés visés – professionnel et divertissement.

Les cibles sont capables de détecter des impacts aussi faibles qu'une fléchette Nerf.

L'ancien câblage du mandant a permis la mise à l'épreuve, pendant une semaine, du système complet, passant haut la main les différents tests.

Avec quelques modifications apportées au boitier pour intégrer la fixation par aimantation, le système est prêt à endurer des tests avec plus de cibles présentes.

9.2. Sentiment

La conception de ce projet peut se résumer par 10 [%] de recherche et décision, 30 [%] de conception électronique et 60 [%] de programmation.

Au niveau électronique, moyennant une erreur corrigée sur la gestion de la batterie ainsi que d'incorrectes empreintes de composants, le système est fonctionnel et protégé de diverses façons (sur/sous-tension, hausse et baisse significative de température), écartant par conséquent de potentiels risques pour l'utilisateur final.

La programmation accueille quant à elle plusieurs principes efficaces (machines d'états, pseudo polymorphisme (par pointeurs de fonctions et transformations par pointeurs void)). La décomposition en couches et niveaux d'imbrication permet un code non-bloquant (hors graphismes), et ainsi un travail facilité en cas de reprise postérieure du projet.

9.3. Conclusion

Avec tout ce qui a été dit, répondant en tous points au cahier des charges donné, et même préparé pour l'implantation de fonctions supplémentaires (sans-fil Wi-Fi, interface PC (USB), téléphone (Bluetooth)), je suis fier du travail effectué ainsi que de l'état final et fonctionnel du projet.

Lieu et date : _____ Signature : _____

10. Annexes

10.1. Ingénierie inverse – système G&G

10.2. Schémas actuels (contrôleur 1.1, cible 1.1, module led 1.0, module détecteur de présence 1.0)

10.3. Schémas datés (contrôleur 1.0, cible 1.0)

10.4. Guide d'utilisation

10.5. Tests de fonctionnement

10.6. Commande électronique effectuée

11. Bibliographie

[1] Trainshot, « Smart Airsoft Targets », *Trainshot*. https://www.trainshot.com/airsoft/ (consulté le mai 25, 2020).
[2] AttackSense, « Active Target Systems », *AttackSense*. https://www.attacksense.com/ (consulté le mai 25, 2020).
[3] G&G, « M.E.T. Targets », *https://www.ggtdu.com/*. https://www.ggtdu.com/accessories/met-unit/ (consulté le mai 25, 2020).
[4] SteelAlive, « Smart Shooting Targets », *SteelAlive*. https://steelalive.io/ (consulté le mai 25, 2020).
[5] Train2Shoot, « The Trainshot smart shooting target system », *Train2Shoot*. http://www.train2shoot.com/ (consulté le mai 25, 2020).
[6] Wikipédia, « g (accélération) », *Wikipédia*. avr. 25, 2020, Consulté le: mai 28, 2020. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=G_(acc%C3%A9l%C3%A9ration)&oldid=170005573.
[7] NXP, « MMA8652FC », p. 65.
[8] Microchip, « MCP73833 », p. 32, 2009.
[9] Diodes Inc, « AP9101C ». https://www.diodes.com/assets/Datasheets/AP9101C.pdf (consulté le août 13, 2020).
[10] Diodes Inc, « DMG9926UDM ». https://www.diodes.com/assets/Datasheets/ds31770.pdf (consulté le août 13, 2020).
[11] muRata, « NCP15XH103J03RC ». https://www.murata.com/en-us/api/pdfdownloadapi?cate=luNTCforTempeSenso&partno=NCP15XH103J03RC (consulté le juin 01, 2020).
[12] muRata, « NCP15XH ». https://www.murata.com/en-us/api/pdfdownloadapi?cate=&partno=NCP15XH103F03RC (consulté le août 11, 2020).
[13] Diodes Inc, « AP7363 ». https://www.diodes.com/assets/Datasheets/AP7363.pdf (consulté le août 11, 2020).
[14] Maxim Integrated, « MAX3051 », p. 13.
[15] « AND8169-D.pdf », *ON Semiconductor*. https://www.onsemi.com/pub/Collateral/AND8169-D.PDF (consulté le mai 26, 2020).
[16] Microchip, « dsPIC33EP ». https://ww1.microchip.com/downloads/en/DeviceDoc/70000689d.pdf (consulté le août 11, 2020).
[17] Geyer Electronic, « Comparison_of_Crystal_Oscillator_and_MEMS_Oscillator.pdf », *www.geyer-electronic.de/*. https://www.geyer-electronic.de/fileadmin/user_upload/frequenz/service/Comparison_of_Crystal_Oscillator_and_MEMS_Oscillator.pdf (consulté le juin 04, 2020).
[18] ECS Inc, « ECS-5032MV », p. 2.
[19] FTDI, « FT230X ». https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf (consulté le août 11, 2020).
[20] R. Arora, « I2C Bus Pull-Up Resistor Calculation », p. 5, 2015.
[21] « HC-SR501 ». https://www.mpja.com/download/31227sc.pdf (consulté le août 11, 2020).
[22] Techgurka, « TechGurka: Cheap Pyroelectric Infrared PIR Motion Sensor on 3.3v », *TechGurka*, mai 07, 2013. https://techgurka.blogspot.com/2013/05/cheap-pyroelectric-infrared-pir-motion.html (consulté le juin 01, 2020).
[23] I. Bourns, « PEC12R – 12 mm Incremental Encoder », p. 5.
[24] E-Switch, « JN2UOANAGX ». http://spec_sheets.e-switch.com/specs/P200000.pdf (consulté le août 11, 2020).
[25] Newhaven Display, « NHD-2.4-240320CF-CSXN ». http://www.newhavendisplay.com/specs/NHD-2.4-240320CF-CSXN-F.pdf (consulté le août 11, 2020).
[26] Texas Instrument, « Controller Area Network Physical Layer Requirements », p. 15, 2008.
[27] Microchip, « Enhanced Controller Area Network (ECAN™) », *www.microchip.com*. http://ww1.microchip.com/downloads/en/devicedoc/70353c.pdf (consulté le juin 16, 2020).
[28] Microchip, « AN1095 », p. 18.

Tables des matières et illustrations

12. Table des matières

Illustrations

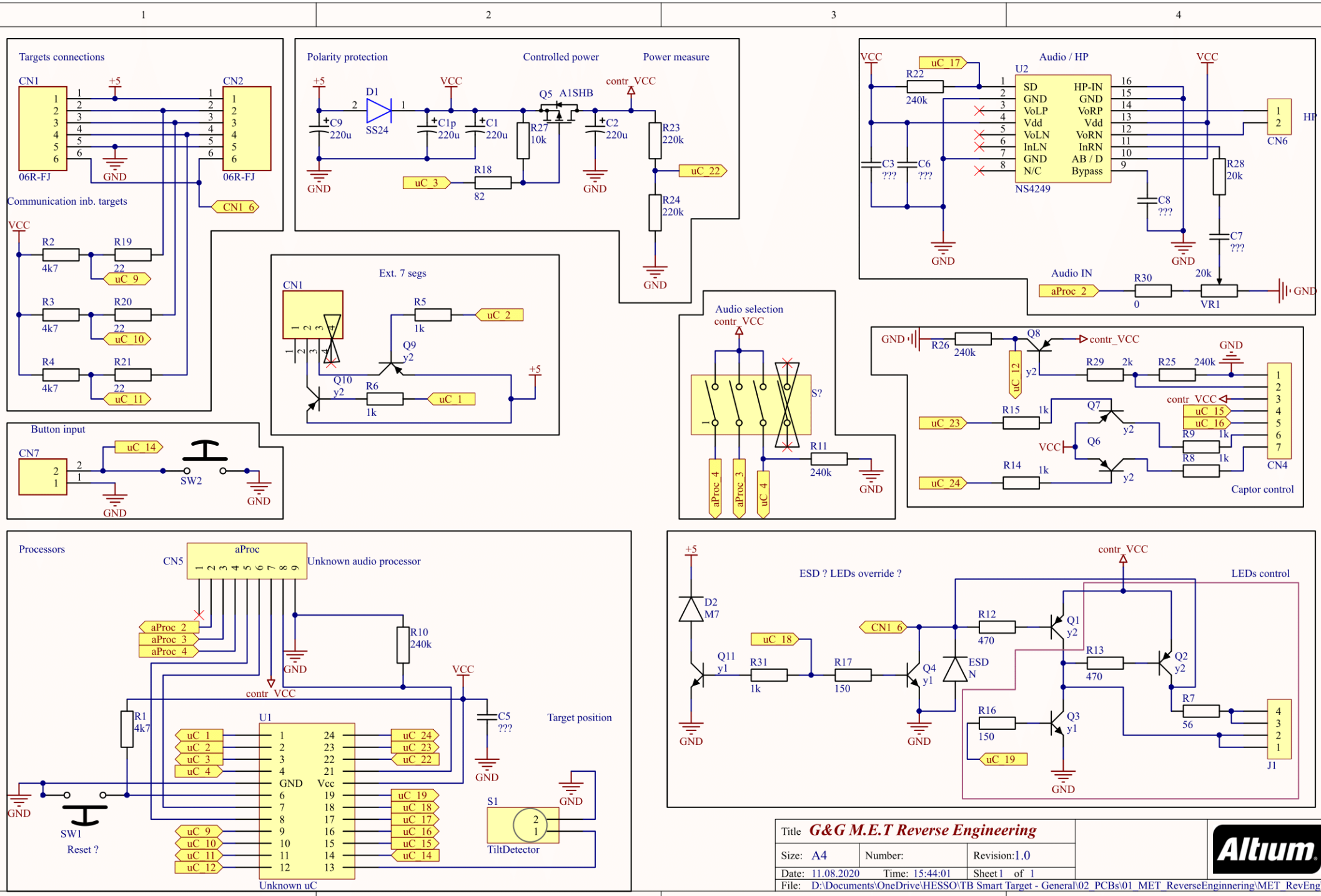
Tableaux

Equations

ANNEXES

ANNEXE 10.1

Ingénierie inverse – système G&G



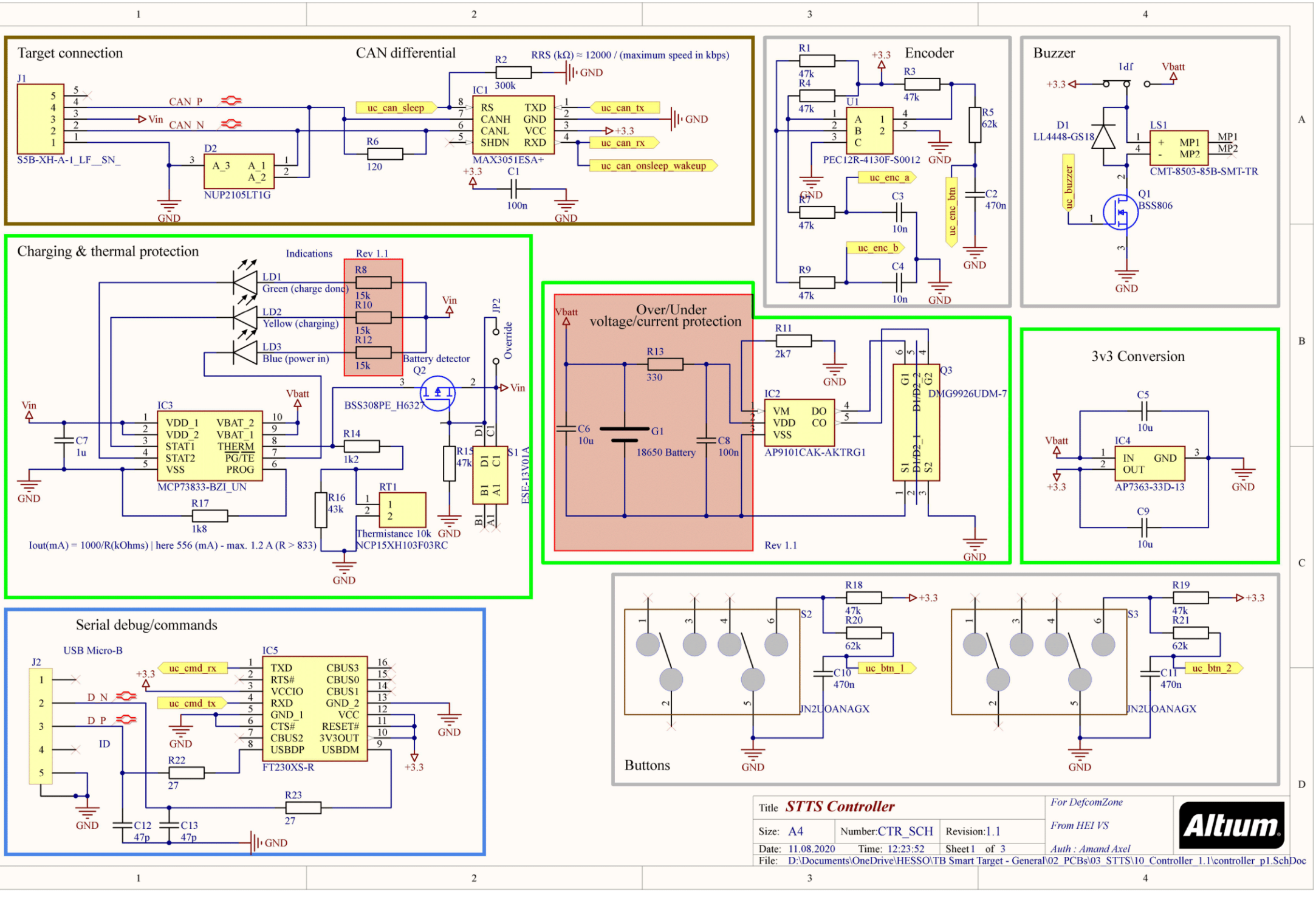
Title	**G&G M.E.T Reverse Engineering**		
Size: A4	Number:	Revision: 1.0	
Date: 11.08.2020	Time: 15:44:01	Sheet 1 of 1	
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\01_MET_ReverseEnginnering\MET_RevEng\Schem_p			

Altium

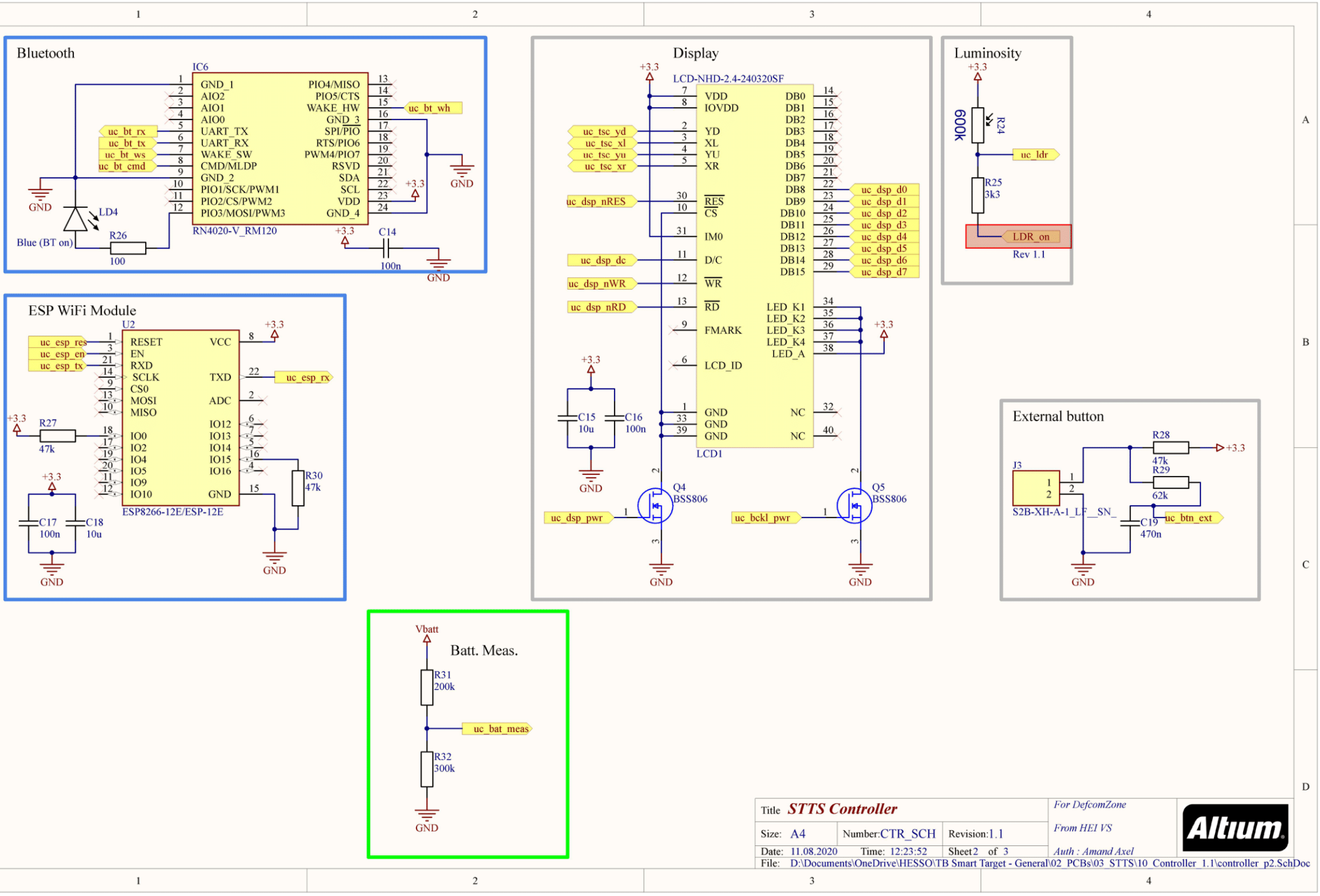
ANNEXE 10.2

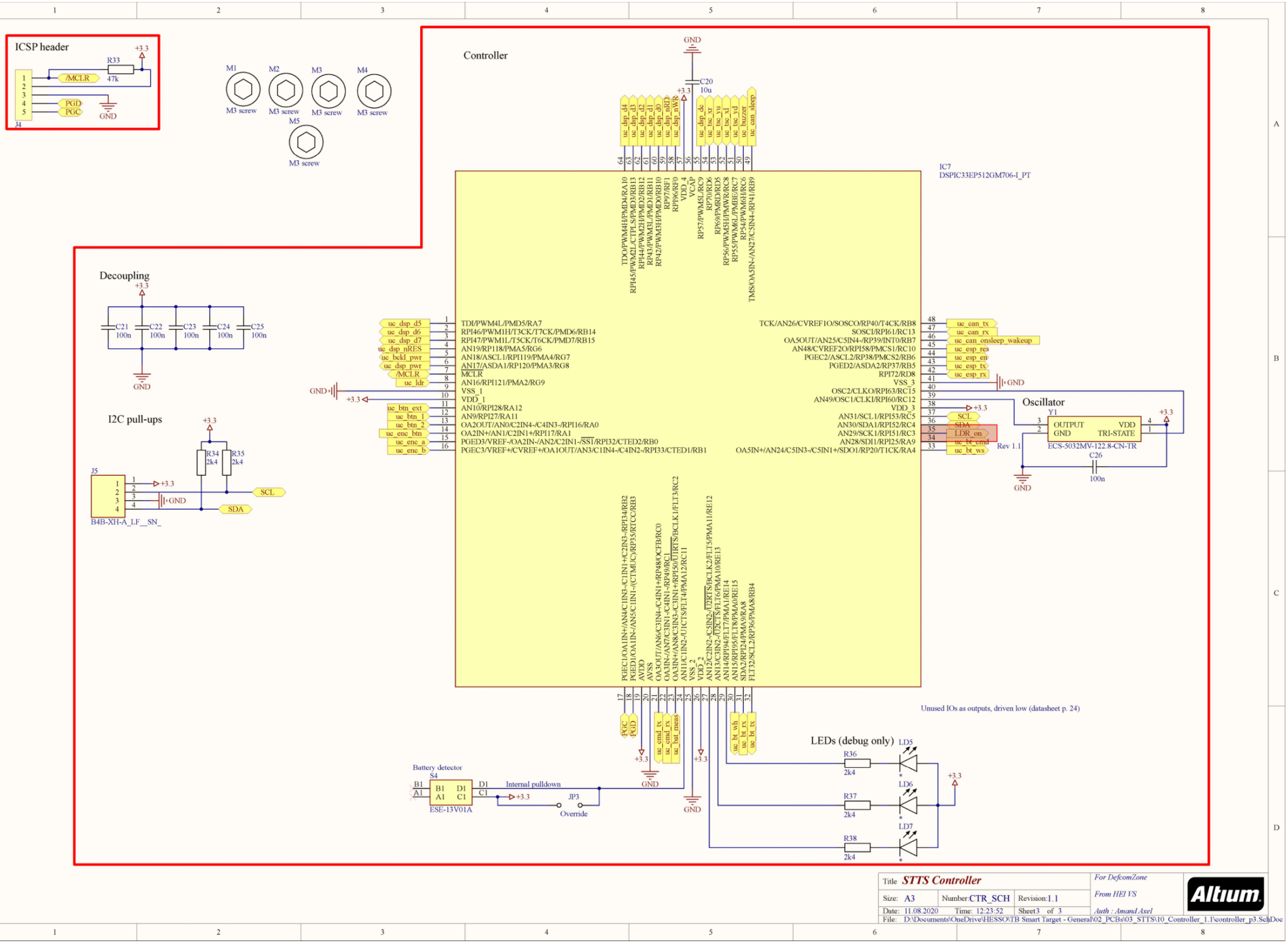
Schémas actuels (contrôleur 1.1, cible 1.1, module led 1.0, module détecteur de présence

Contrôleur 1.1

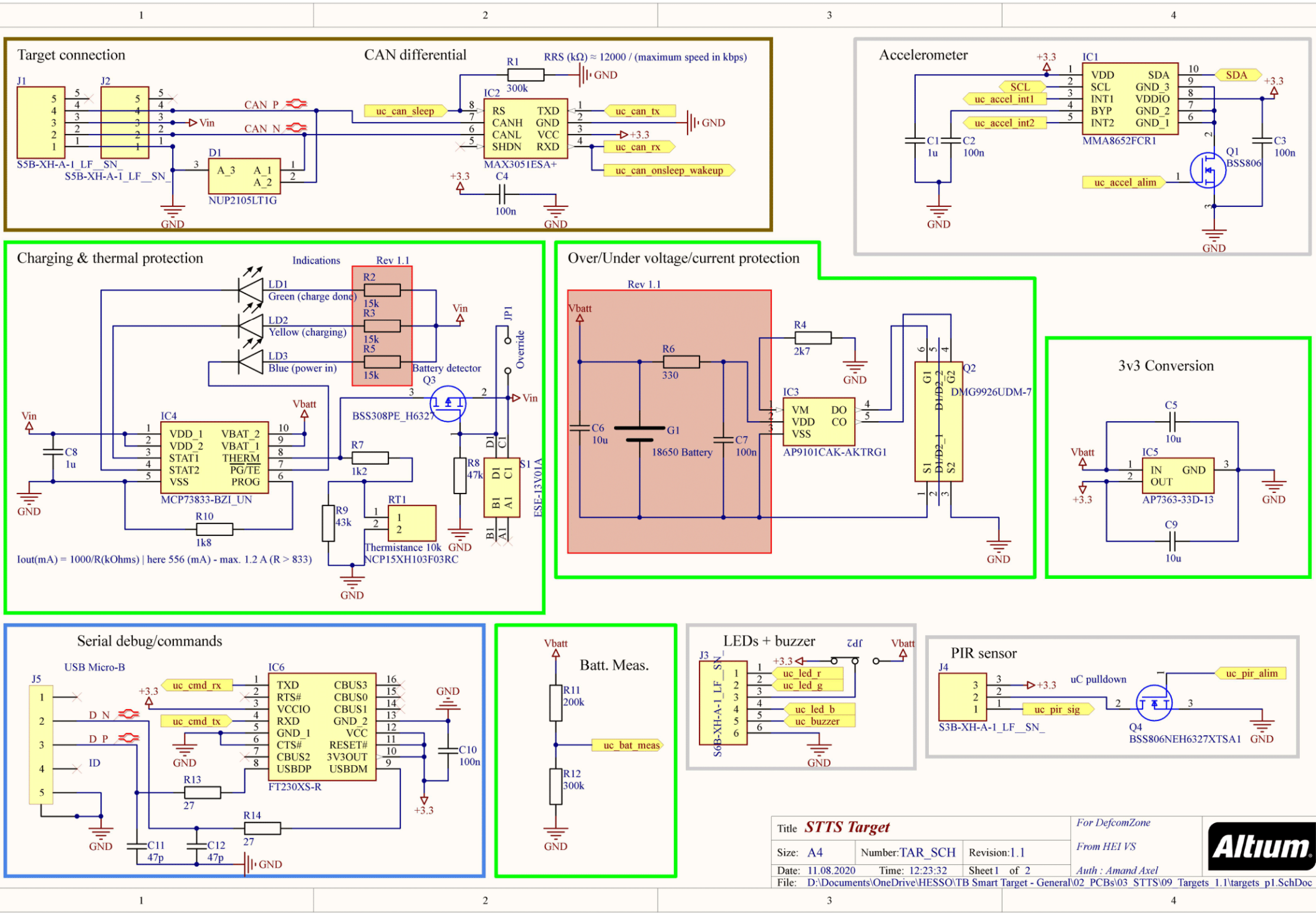


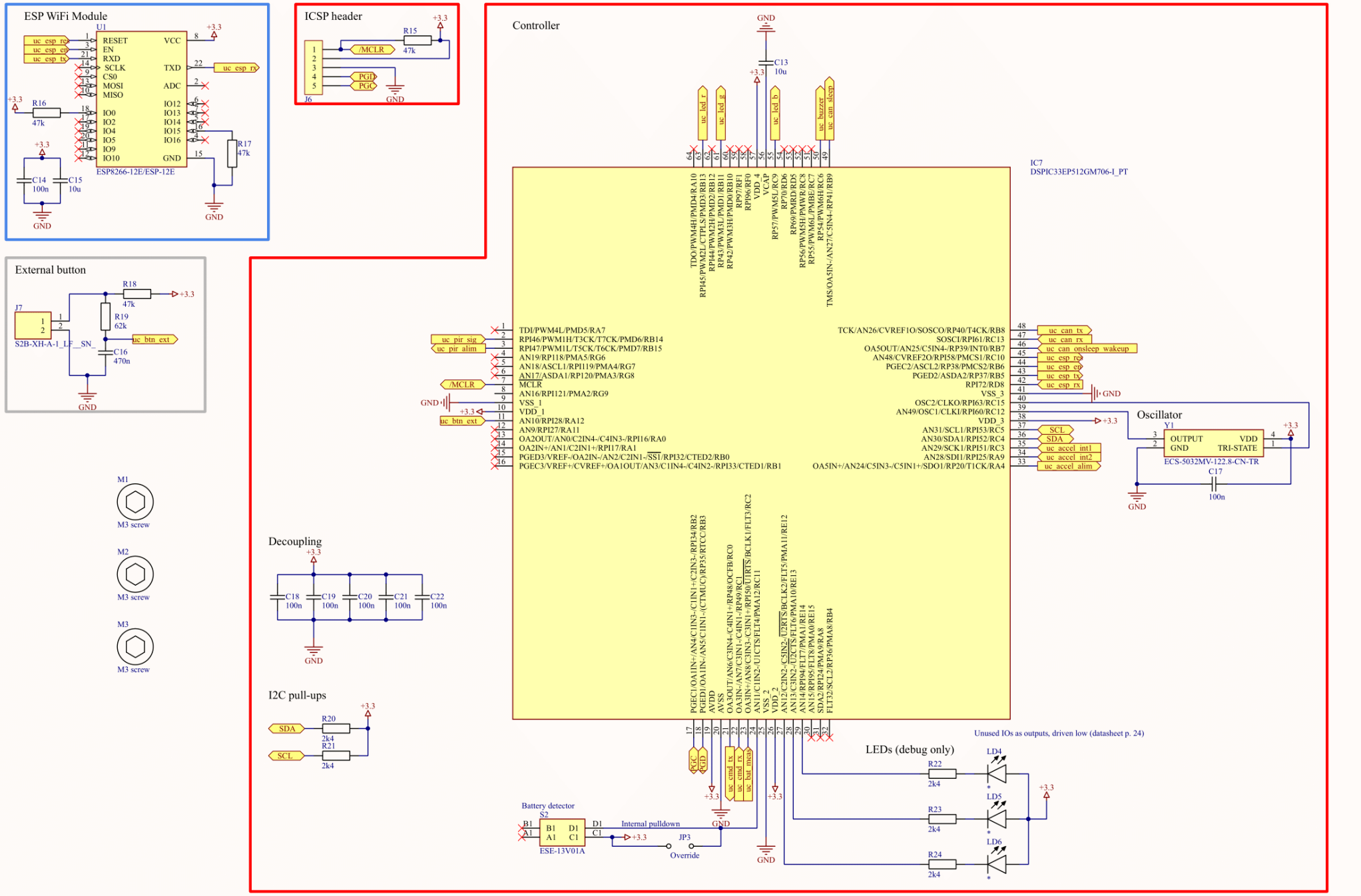
Title	**STTS Controller**		*For DefcomZone*
Size: A4	Number:CTR_SCH	Revision:1.1	*From HEI VS*
Date: 11.08.2020	Time: 12:23:52	Sheet 1 of 3	*Auth : Amand Axel*
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\10_Controller_1.1\controller_p1.SchDoc			


Bluetooth
IC6
GND_1
AIO2
AIO1
AIO0
UART_TX
UART_RX
WAKE_SW
CMD/MLDP
GND_2
PIO1/SCK/PWM1
PIO2/CS/PWM2
PIO3/MOSI/PWM3
PIO4/MISO
PIO5/CTS
WAKE_HW
GND_3
SPI/PIO
RTS/PIO6
PWM4/PIO7
RSVD
SDA
SCL
VDD
GND_4
RN4020-V_RM120
uc_bt_rx
uc_bt_tx
uc_bt_ws
uc_bt_cmd
uc_bt_wh
LD4
Blue (BT on)
R26
100
C14
100n
+3.3
GND
ESP WiFi Module
U2
RESET
EN
RXD
SCLK
CS0
MOSI
MISO
IO0
IO2
IO4
IO5
IO9
IO10
VCC
TXD
ADC
IO12
IO13
IO14
IO15
IO16
GND
ESP8266-12E/ESP-12E
uc_esp_res
uc_esp_en
uc_esp_tx
uc_esp_rx
R27
47k
R30
47k
C17
100n
C18
10u
Display
LCD-NHD-2.4-240320SF
VDD
IOVDD
YD
XL
YU
XR
RES
CS
IM0
D/C
WR
RD
FMARK
LCD_ID
GND
GND
GND
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DB8
DB9
DB10
DB11
DB12
DB13
DB14
DB15
LED_K1
LED_K2
LED_K3
LED_K4
LED_A
NC
NC
LCD1
uc_tsc_yd
uc_tsc_xl
uc_tsc_yu
uc_tsc_xr
uc_dsp_nRES
uc_dsp_dc
uc_dsp_nWR
uc_dsp_nRD
uc_dsp_d0
uc_dsp_d1
uc_dsp_d2
uc_dsp_d3
uc_dsp_d4
uc_dsp_d5
uc_dsp_d6
uc_dsp_d7
C15
10u
C16
100n
Q4
BSS806
Q5
BSS806
uc_dsp_pwr
uc_bckl_pwr
Luminosity
600k
R24
uc_ldr
R25
3k3
LDR_on
Rev 1.1
External button
J3
S2B-XH-A-1_LF__SN_
R28
47k
R29
62k
C19
470n
uc_btn_ext
Batt. Meas.
Vbatt
R31
200k
R32
300k
uc_bat_meas
Title STTS Controller
Size: A4
Number:CTR_SCH
Revision:1.1
Date: 11.08.2020
Time: 12:23:52
Sheet2 of 3
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\10_Controller_1.1\controller_p2.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium


ICSP header
J4
/MCLR
47k
PGD
PGC
GND
M1
M2
M3
M4
M5
M3 screw
Controller
IC7
DSPIC33EP512GM706-I_PT
Decoupling
C21 100n
C22 100n
C23 100n
C24 100n
C25 100n
I2C pull-ups
R34 2k4
R35 2k4
J5
B4B-XH-A_LF__SN_
SCL
SDA
Oscillator
Y1
ECS-5032MV-122.8-CN-TR
C26 100n
Rev 1.1
Battery detector
S4
ESE-13V01A
Internal pulldown
JP3
Override
LEDs (debug only)
LD5
LD6
LD7
R36 2k4
R37 2k4
R38 2k4
Unused IOs as outputs, driven low (datasheet p. 24)
Title STTS Controller
Size: A3
Number:CTR_SCH
Revision:1.1
Date: 11.08.2020
Time: 12:23:52
Sheet3 of 3
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\10_Controller_1.1\controller_p3.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium

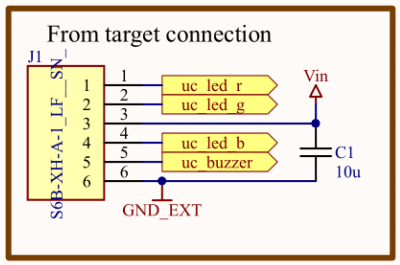
Cible 1.1

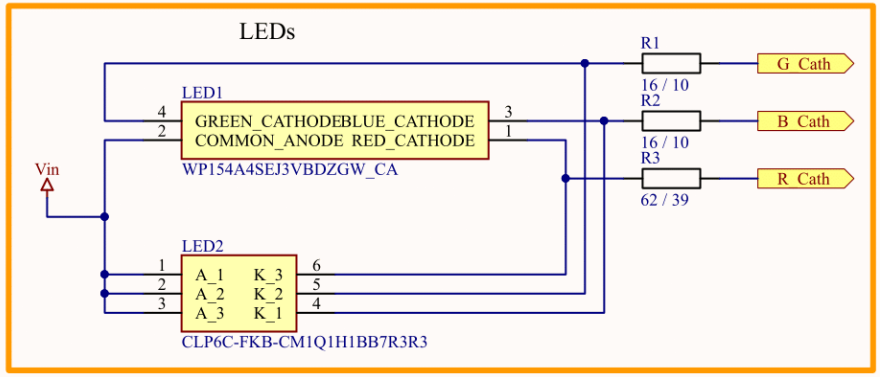

Target connection
CAN differential
J1
J2
S5B-XH-A-1_LF__SN_
S5B-XH-A-1_LF__SN_
CAN_P
CAN_N
Vin
D1
A_3
A_1
A_2
NUP2105LT1G
GND
R1
300k
RRS (kΩ) ≈ 12000 / (maximum speed in kbps)
IC2
RS
CANH
CANL
SHDN
TXD
GND
VCC
RXD
MAX3051ESA+
uc_can_sleep
uc_can_tx
uc_can_rx
uc_can_onsleep_wakeup
+3.3
C4
100n
Accelerometer
IC1
VDD
SCL
INT1
BYP
INT2
SDA
GND_3
VDDIO
GND_2
GND_1
MMA8652FCR1
uc_accel_int1
uc_accel_int2
uc_accel_alim
C1
1u
C2
100n
C3
100n
Q1
BSS806
Charging & thermal protection
Indications
Rev 1.1
LD1
Green (charge done)
LD2
Yellow (charging)
LD3
Blue (power in)
R2
15k
R3
15k
R5
15k
JP1
Override
Battery detector
Q3
BSS308PE_H6327
IC4
VDD_1
VDD_2
STAT1
STAT2
VSS
VBAT_2
VBAT_1
THERM
PG/TE
PROG
MCP73833-BZI_UN
C8
1u
R7
1k2
R8
47k
R9
43k
R10
1k8
RT1
Thermistance 10k
NCP15XH103F03RC
S1
ESE-13V01A
Iout(mA) = 1000/R(kOhms) | here 556 (mA) - max. 1.2 A (R > 833)
Over/Under voltage/current protection
Rev 1.1
Vbatt
R6
330
R4
2k7
C6
10u
G1
18650 Battery
C7
100n
IC3
VM
VDD
VSS
DO
CO
AP9101CAK-AKTRG1
Q2
DMG9926UDM-7
3v3 Conversion
C5
10u
IC5
IN
OUT
GND
AP7363-33D-13
C9
10u
Serial debug/commands
USB Micro-B
J5
D_N
D_P
ID
uc_cmd_rx
uc_cmd_tx
IC6
TXD
RTS#
VCCIO
RXD
GND_1
CTS#
CBUS2
USBDP
CBUS3
CBUS0
CBUS1
GND_2
VCC
RESET#
3V3OUT
USBDM
FT230XS-R
R13
27
R14
27
C10
100n
C11
47p
C12
47p
Batt. Meas.
R11
200k
R12
300k
uc_bat_meas
LEDs + buzzer
J3
S6B-XH-A-1_LF__SN_
uc_led_r
uc_led_g
uc_led_b
uc_buzzer
JP2
PIR sensor
J4
S3B-XH-A-1_LF__SN_
uC pulldown
uc_pir_sig
uc_pir_alim
Q4
BSS806NEH6327XTSA1
Title STTS Target
Size: A4
Number:TAR_SCH
Revision:1.1
Date: 11.08.2020
Time: 12:23:32
Sheet 1 of 2
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\09_Targets_1.1\targets_p1.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium

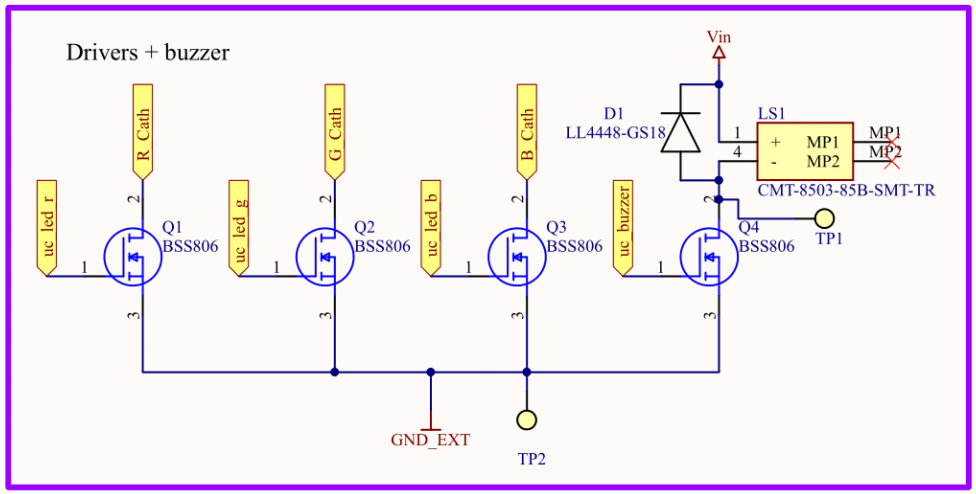


Title	STTS Target		For DefcomZone
Size: A3	Number: TAR_SCH	Revision: 1.1	From HEI VS
Date: 11.08.2020	Time: 12:23:33	Sheet 2 of 2	Auth : Amand Axel
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\09_Targets_1.1\targets_p2.SchDoc			

Module LED et buzzer 1.0

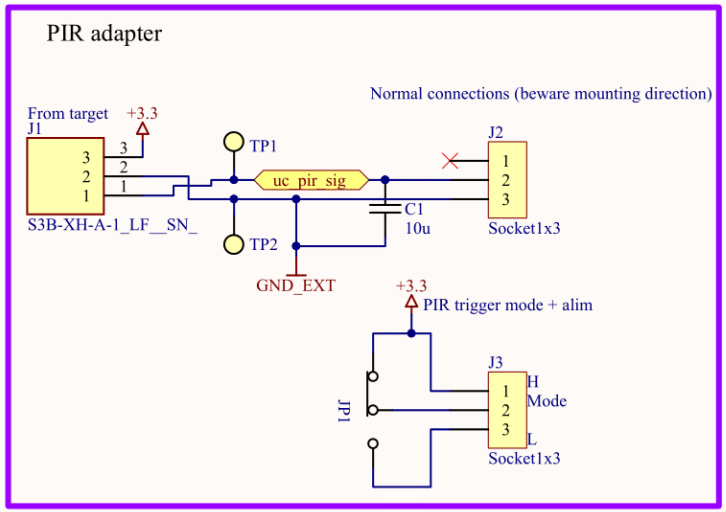






Title	***STTS Leds-Buzzer module***		*For DefcomZone*
Size: A4	Number:LDB_SCH	Revision:1.0	*From HEI VS*
Date: 11.08.2020	Time: 12:43:11	Sheet 1 of 1	*Auth : Amand Axel*
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\03_LEDs_1.0\leds_p1.SchDoc			

Module détecteur de présence 1.0



M1 M3 screw

M2 M3 screw

M3 M3 screw

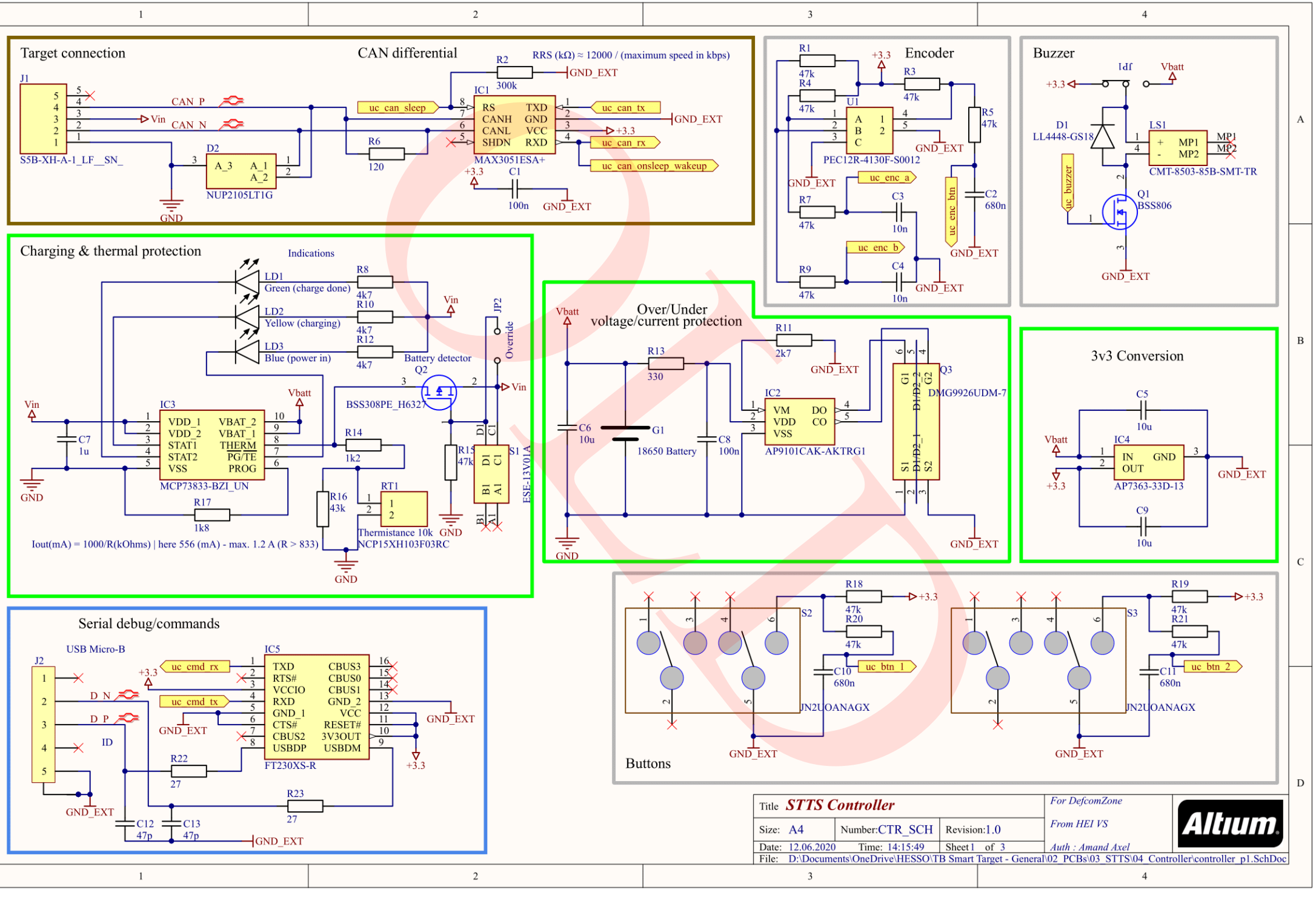
M4 M3 screw

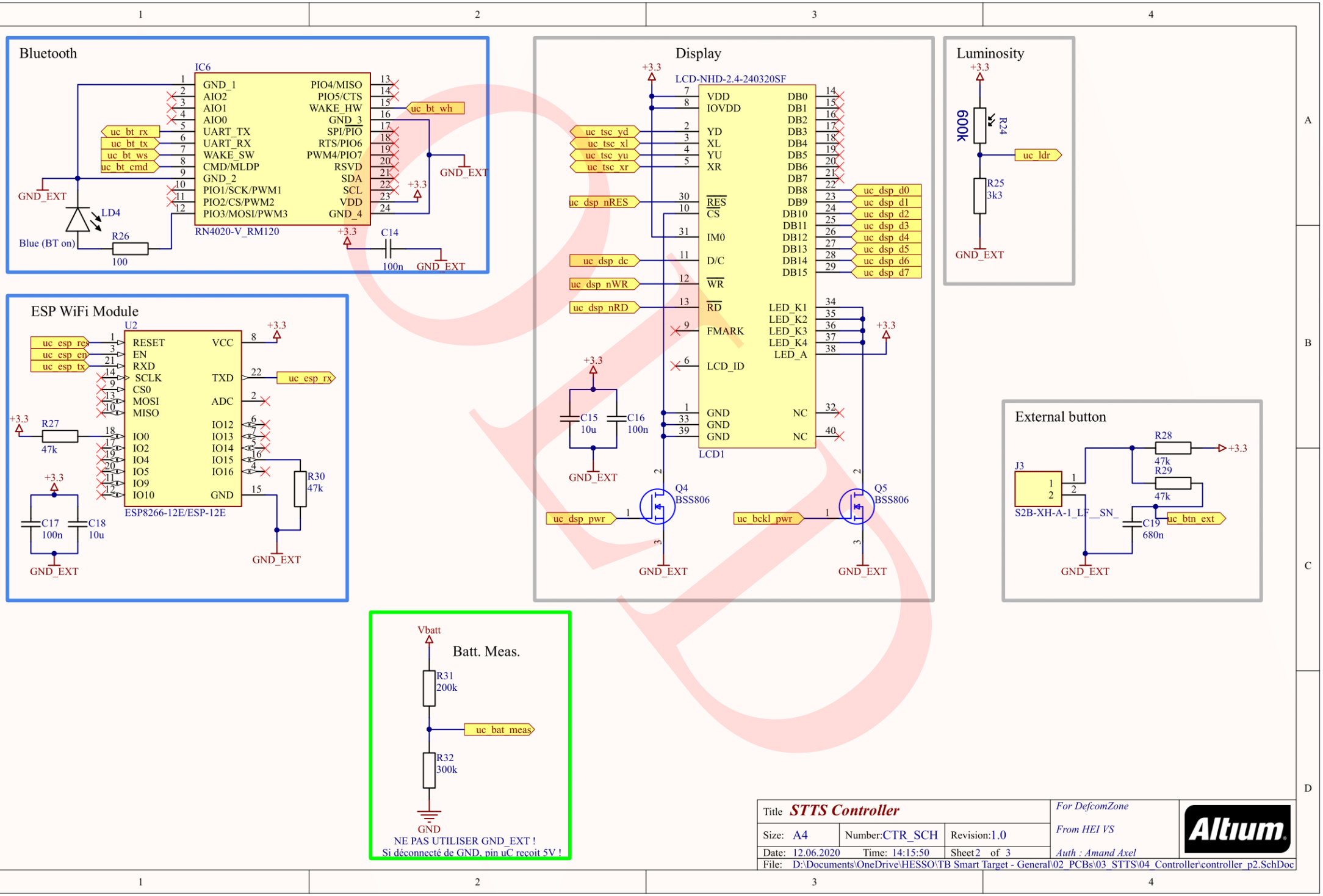
Title	***STTS PIR module***		*For DefcomZone*
Size: A4	Number:PIR_SCH	Revision:1.0	*From HEI VS*
Date: 11.08.2020	Time: 12:42:03	Sheet 1 of 1	*Auth : Amand Axel*
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\05_IRDetector_1.0\irdetector_p1.SchDoc			

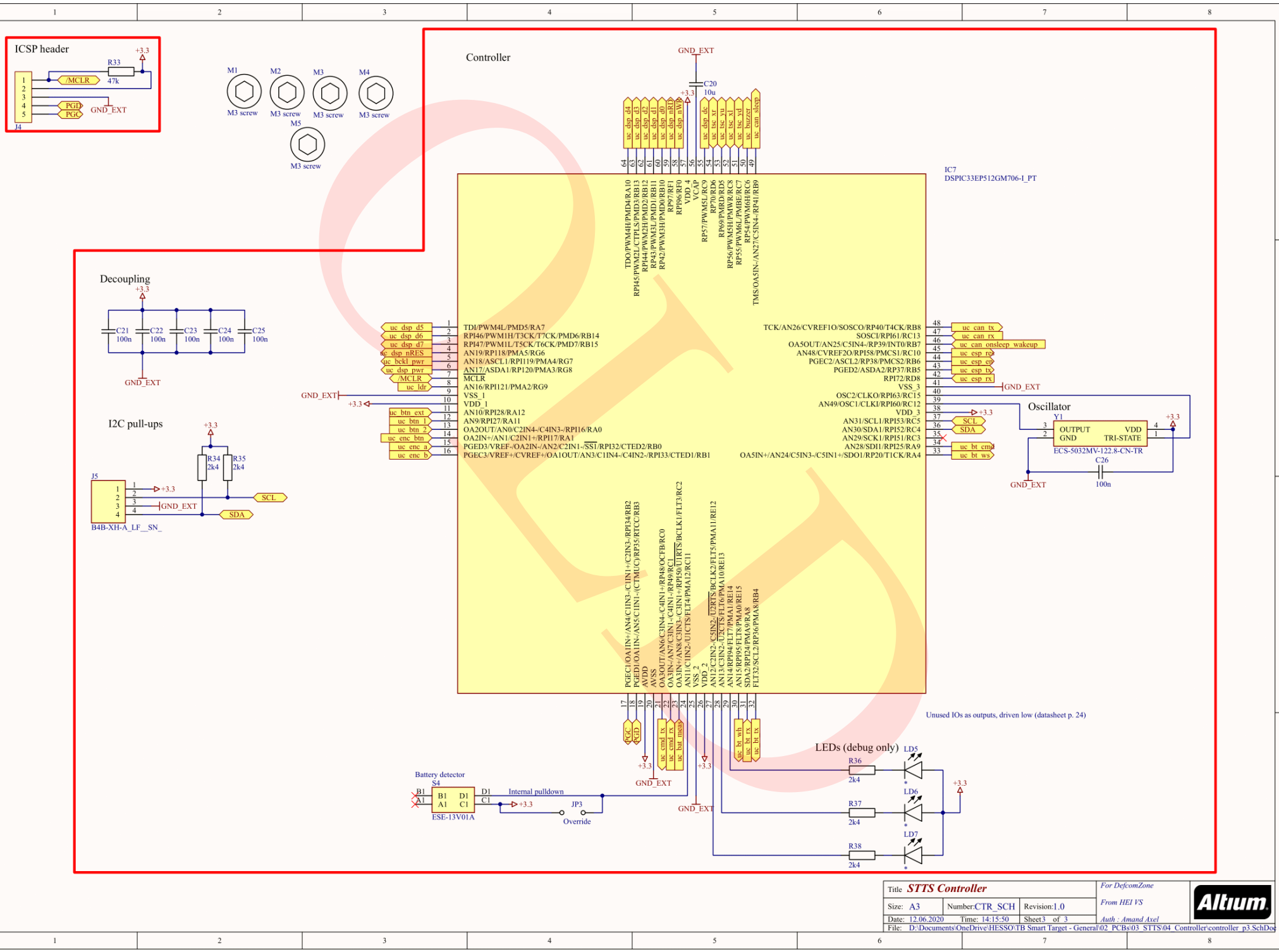
ANNEXE 10.3

Schémas datés (contrôleur 1.0, cible 1.0)

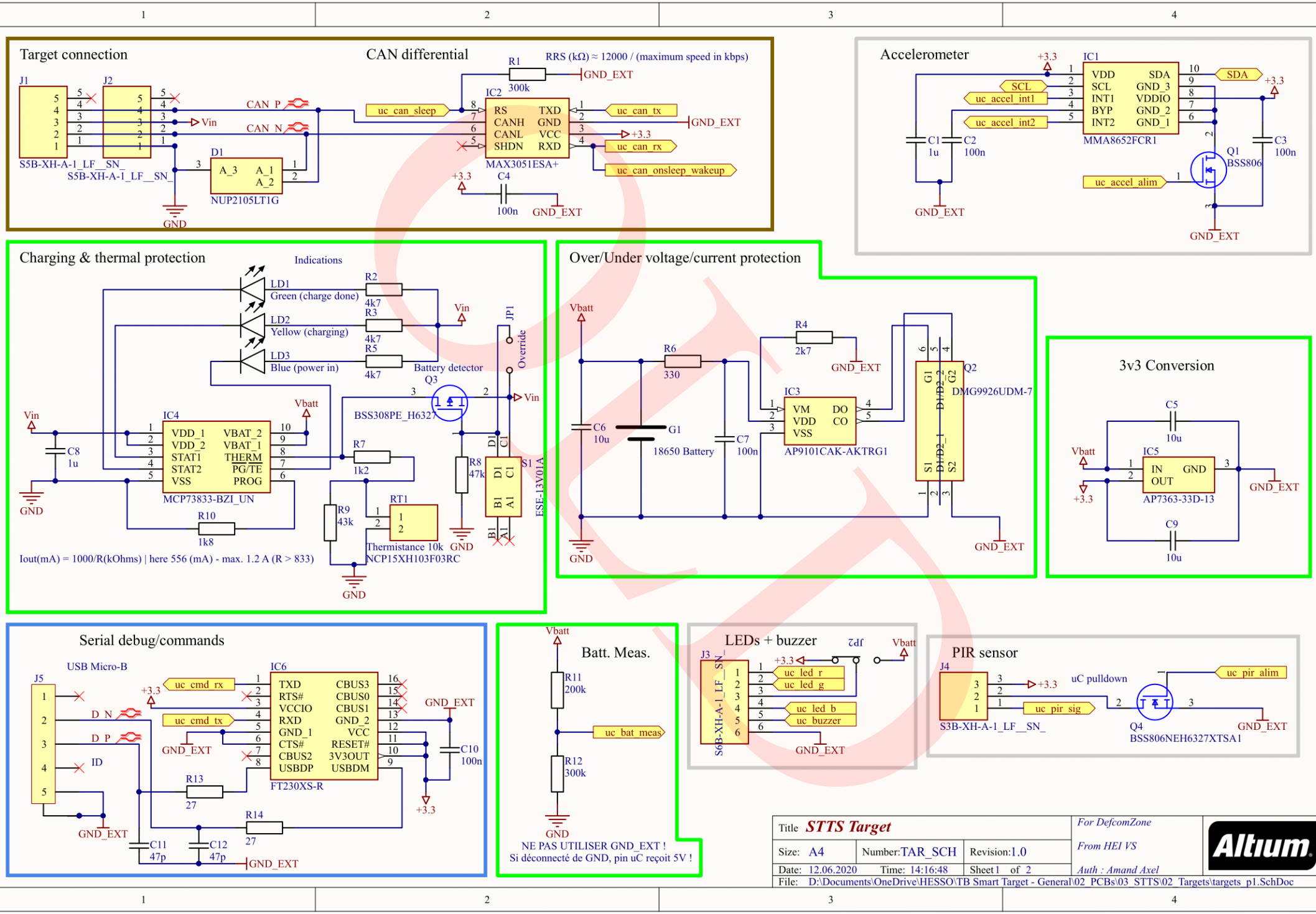
Contrôleur 1.0

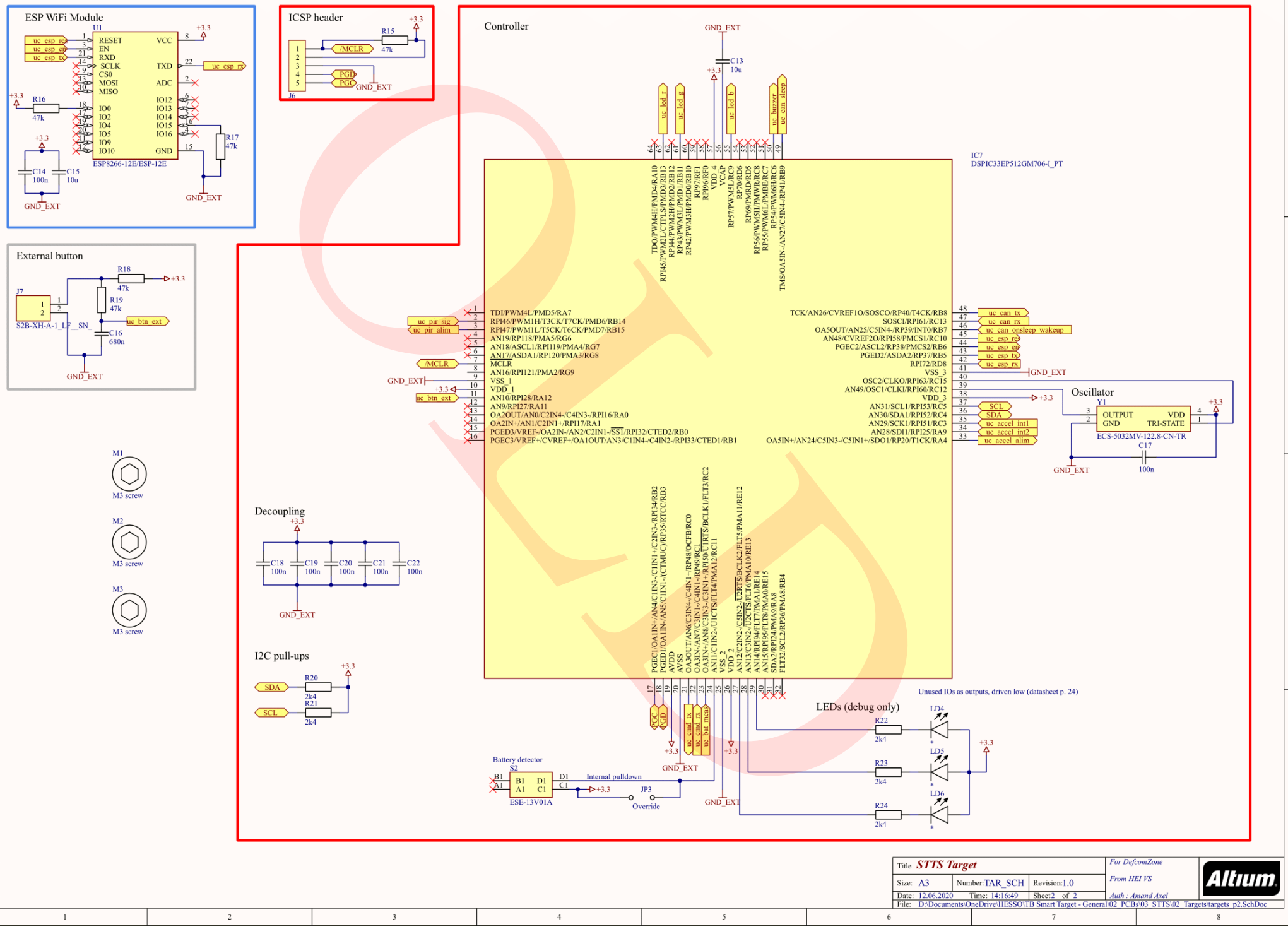

Target connection
CAN differential
RRS (kΩ) ≈ 12000 / (maximum speed in kbps)
J1
S5B-XH-A-1_LF__SN_
CAN_P
CAN_N
Vin
GND
D2
NUP2105LT1G
R2
300k
GND_EXT
IC1
MAX3051ESA+
uc_can_sleep
uc_can_tx
uc_can_rx
uc_can_onsleep_wakeup
R6
120
C1
100n
+3.3
Encoder
R1
47k
R3
R4
R5
U1
PEC12R-4130F-S0012
uc_enc_a
uc_enc_b
uc_enc_btn
C2
680n
C3
10n
C4
R7
R9
Buzzer
Idf
Vbatt
D1
LL4448-GS18
LS1
CMT-8503-85B-SMT-TR
MP1
MP2
uc_buzzer
Q1
BSS806
Charging & thermal protection
Indications
LD1
Green (charge done)
LD2
Yellow (charging)
LD3
Blue (power in)
R8
4k7
R10
R12
Battery detector
Q2
BSS308PE_H6327
JP2
Override
IC3
MCP73833-BZI_UN
VDD_1
VDD_2
STAT1
STAT2
VSS
VBAT_2
VBAT_1
THERM
PG/TE
PROG
C7
1u
R14
1k2
R15
47k
S1
ESE-13V01A
R16
43k
RT1
Thermistance 10k
NCP15XH103F03RC
R17
1k8
Iout(mA) = 1000/R(kOhms) | here 556 (mA) - max. 1.2 A (R > 833)
Over/Under
voltage/current protection
Vbatt
R13
330
C6
10u
G1
18650 Battery
C8
100n
R11
2k7
IC2
AP9101CAK-AKTRG1
VM
VDD
VSS
DO
CO
Q3
DMG9926UDM-7
3v3 Conversion
C5
10u
IC4
AP7363-33D-13
IN
OUT
GND
C9
10u
Serial debug/commands
USB Micro-B
J2
D_N
D_P
ID
IC5
FT230XS-R
TXD
RTS#
VCCIO
RXD
GND_1
CTS#
CBUS2
USBDP
CBUS3
CBUS0
CBUS1
GND_2
VCC
RESET#
3V3OUT
USBDM
uc_cmd_rx
uc_cmd_tx
R22
27
R23
27
C12
47p
C13
47p
Buttons
S2
S3
JN2UOANAGX
R18
R19
R20
R21
C10
680n
C11
680n
uc_btn_1
uc_btn_2
Title STTS Controller
Size: A4
Number:CTR_SCH
Revision:1.0
Date: 12.06.2020
Time: 14:15:49
Sheet 1 of 3
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\04_Controller\controller_p1.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium


Bluetooth
IC6
GND_1
AIO2
AIO1
AIO0
UART_TX
UART_RX
WAKE_SW
CMD/MLDP
GND_2
PIO1/SCK/PWM1
PIO2/CS/PWM2
PIO3/MOSI/PWM3
PIO4/MISO
PIO5/CTS
WAKE_HW
GND_3
SPI/PIO
RTS/PIO6
PWM4/PIO7
RSVD
SDA
SCL
VDD
GND_4
uc_bt_rx
uc_bt_tx
uc_bt_ws
uc_bt_cmd
uc_bt_wh
RN4020-V_RM120
GND_EXT
LD4
Blue (BT on)
R26
100
+3.3
C14
100n
ESP WiFi Module
U2
RESET
EN
RXD
SCLK
CS0
MOSI
MISO
IO0
IO2
IO4
IO5
IO9
IO10
VCC
TXD
ADC
IO12
IO13
IO14
IO15
IO16
GND
uc_esp_rst
uc_esp_en
uc_esp_tx
uc_esp_rx
R27
47k
R30
47k
C17
100n
C18
10u
ESP8266-12E/ESP-12E
Display
LCD-NHD-2.4-240320SF
VDD
IOVDD
YD
XL
YU
XR
RES
CS
IM0
D/C
WR
RD
FMARK
LCD_ID
GND
GND
GND
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DB8
DB9
DB10
DB11
DB12
DB13
DB14
DB15
NC
NC
LED_K1
LED_K2
LED_K3
LED_K4
LED_A
uc_tsc_yd
uc_tsc_xl
uc_tsc_yu
uc_tsc_xr
uc_dsp_nRES
uc_dsp_dc
uc_dsp_nWR
uc_dsp_nRD
uc_dsp_d0
uc_dsp_d1
uc_dsp_d2
uc_dsp_d3
uc_dsp_d4
uc_dsp_d5
uc_dsp_d6
uc_dsp_d7
C15
10u
C16
100n
LCD1
Q4
BSS806
Q5
BSS806
uc_dsp_pwr
uc_bckl_pwr
Luminosity
600k
R24
R25
3k3
uc_ldr
External button
J3
S2B-XH-A-1_LF__SN_
R28
47k
R29
47k
C19
680n
uc_btn_ext
Batt. Meas.
Vbatt
R31
200k
R32
300k
uc_bat_meas
GND
NE PAS UTILISER GND_EXT !
Si déconnecté de GND, pin uC reçoit 5V !
Title STTS Controller
Size: A4
Number:CTR_SCH
Revision:1.0
Date: 12.06.2020
Time: 14:15:50
Sheet2 of 3
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\04_Controller\controller_p2.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium


ICSP header
J4
/MCLR
PGD
PGC
GND_EXT
+3.3
R33
47k
M1
M2
M3
M4
M5
M3 screw
Controller
GND_EXT
C20
10u
IC7
DSPIC33EP512GM706-I_PT
Decoupling
C21 100n
C22 100n
C23 100n
C24 100n
C25 100n
I2C pull-ups
R34 2k4
R35 2k4
J5
B4B-XH-A_LF__SN_
SCL
SDA
Oscillator
Y1
ECS-5032MV-122.8-CN-TR
C26
100n
OUTPUT
GND
VDD
TRI-STATE
Battery detector
S4
ESE-13V01A
Internal pulldown
JP3
Override
LEDs (debug only)
LD5
LD6
LD7
R36
R37
R38
2k4
Unused IOs as outputs, driven low (datasheet p. 24)
Title STTS Controller
Size: A3
Number:CTR_SCH
Revision:1.0
Date: 12.06.2020
Time: 14:15:50
Sheet3 of 3
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\04_Controller\controller_p3.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium

Cible 1.0


Target connection
CAN differential
RRS (kΩ) ≈ 12000 / (maximum speed in kbps)
S5B-XH-A-1_LF__SN_
NUP2105LT1G
MAX3051ESA+
Accelerometer
MMA8652FCR1
BSS806
Charging & thermal protection
Indications
LD1
Green (charge done)
LD2
Yellow (charging)
LD3
Blue (power in)
Battery detector
BSS308PE_H6327
MCP73833-BZI_UN
Iout(mA) = 1000/R(kOhms) | here 556 (mA) - max. 1.2 A (R > 833)
Thermistance 10k
NCP15XH103F03RC
ESE-13V01A
Over/Under voltage/current protection
18650 Battery
AP9101CAK-AKTRG1
DMG9926UDM-7
3v3 Conversion
AP7363-33D-13
Serial debug/commands
USB Micro-B
FT230XS-R
Batt. Meas.
NE PAS UTILISER GND_EXT !
Si déconnecté de GND, pin uC reçoit 5V !
LEDs + buzzer
PIR sensor
uC pulldown
S3B-XH-A-1_LF__SN_
BSS806NEH6327XTSA1
Title STTS Target
Size: A4
Number: TAR_SCH
Revision: 1.0
Date: 12.06.2020
Time: 14:16:48
Sheet 1 of 2
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\02_Targets\targets_p1.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium


ESP WiFi Module
U1
RESET
EN
RXD
SCLK
CS0
MOSI
MISO
IO0
IO2
IO4
IO5
IO9
IO10
VCC
TXD
ADC
IO12
IO13
IO14
IO15
IO16
GND
ESP8266-12E/ESP-12E
uc_esp_rst
uc_esp_en
uc_esp_tx
uc_esp_rx
R16
47k
R17
47k
C14
100n
C15
10u
GND_EXT
ICSP header
J6
/MCLR
PGD
PGC
R15
47k
External button
J7
S2B-XH-A-1_LF__SN_
R18
47k
R19
47k
C16
680n
uc_btn_ext
Controller
IC7
DSPIC33EP512GM706-I_PT
C13
10u
uc_led_r
uc_led_g
uc_led_b
uc_buzzer
uc_can_sleep
uc_pir_sig
uc_pir_alim
/MCLR
uc_btn_ext
uc_can_tx
uc_can_rx
uc_can_onsleep_wakeup
uc_esp_rst
uc_esp_en
uc_esp_tx
uc_esp_rx
SCL
SDA
uc_accel_int1
uc_accel_int2
uc_accel_alim
Oscillator
Y1
OUTPUT
GND
VDD
TRI-STATE
ECS-5032MV-122.8-CN-TR
C17
100n
M1
M3 screw
M2
M3 screw
M3
M3 screw
Decoupling
C18
100n
C19
100n
C20
100n
C21
100n
C22
100n
I2C pull-ups
R20
2k4
R21
2k4
Unused IOs as outputs, driven low (datasheet p. 24)
LEDs (debug only)
LD4
LD5
LD6
R22
2k4
R23
2k4
R24
2k4
Battery detector
S2
ESE-13V01A
Internal pulldown
JP3
Override
uc_cmd_tx
uc_cmd_rx
uc_bat_meas
Title STTS Target
Size: A3
Number:TAR_SCH
Revision:1.0
Date: 12.06.2020
Time: 14:16:49
Sheet2 of 2
File: D:\Documents\OneDrive\HESSO\TB Smart Target - General\02_PCBs\03_STTS\02_Targets\targets_p2.SchDoc
For DefcomZone
From HEI VS
Auth : Amand Axel
Altium

ANNEXE 10.4
Guide d’utilisation

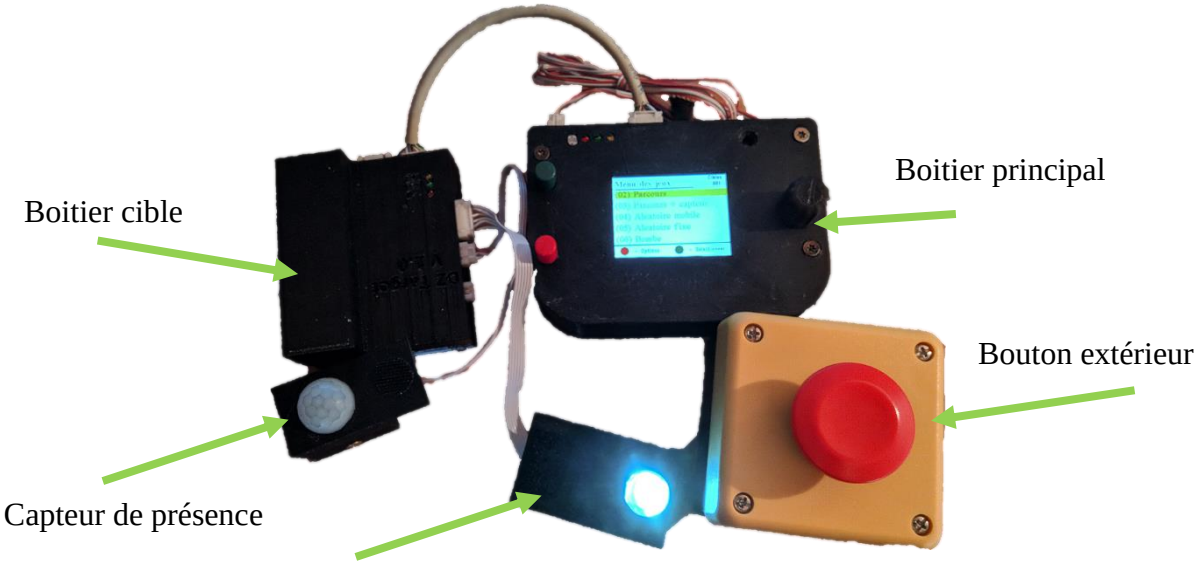
Guide d'utilisation – système STTS – rév. 1.0

Aperçu

Le système STTS – **S**mart **T**arget **T**rainer **S**ystem – est un complet de ciblerie pour le divertissement ou l'entraînement.

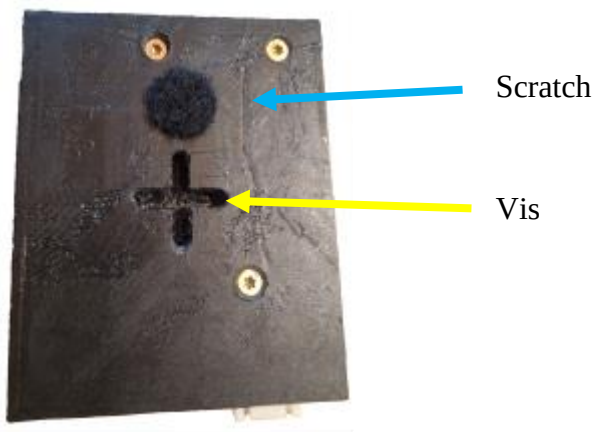
Le système s'adapte sur tout type de cibles, fonctionnant des Nerfs à la SiMunition, en fixant simplement le boitier sur la cible voulue.

Chaque cible possède lumière, son, ainsi que la possibilité d'utiliser un détecteur de présence pour vous mettre sous pression.



Mise en route

Fixer les boitiers « target » sur les supports sélectionnés (cible en bois, carton, plexiglass …), à l'aide de scratch ou de vis qui s'encastrent dans le boitier :



Placer les lumières à l'endroit voulu, et les relier à la cible sur le connecteur 6 positions.
Si utilisé, fixer le capteur de présence et le relier à l'aide du câble 3 positions :

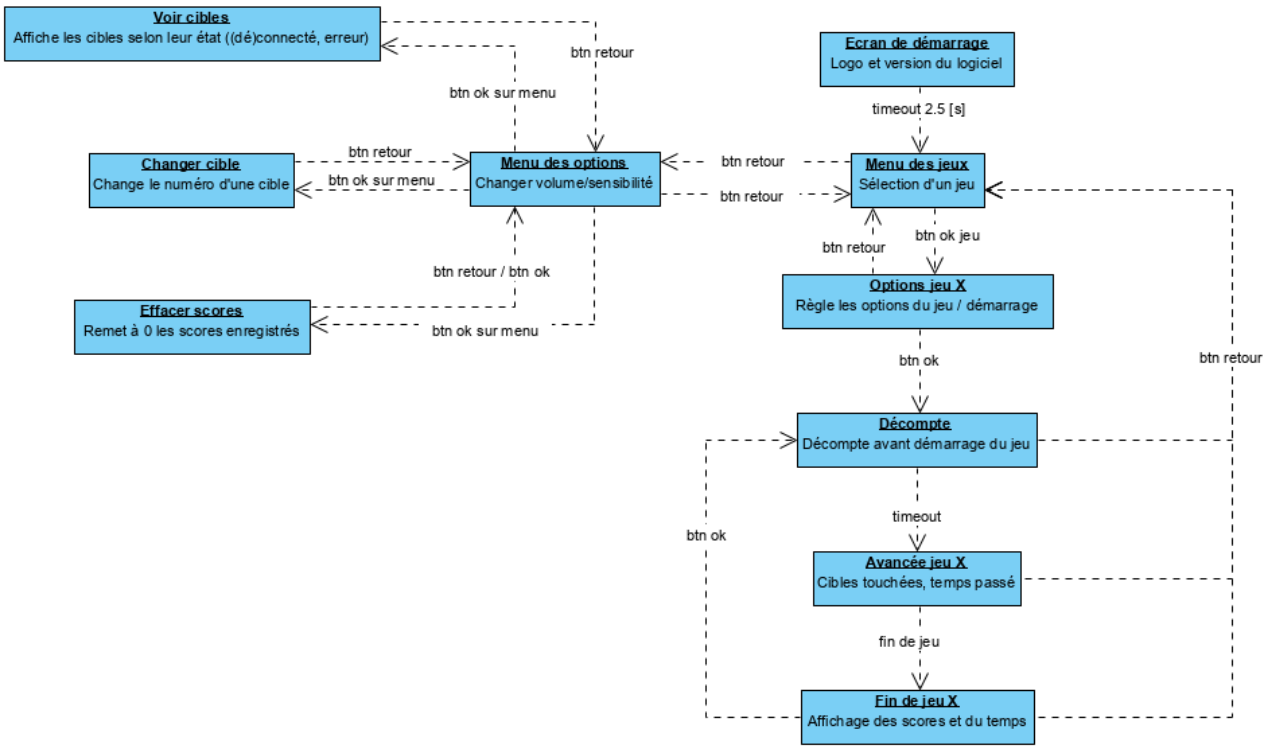
Relier les cibles entres-elles, puis au contrôleur, avec les câbles 5 positions.
Si nécessaire, relier les alimentations pour utiliser le système sur secteur / mettre en charge.

Manipulation

Si le contrôleur est éteint, appuyer sur n'importe quel bouton ou tourner la molette le réveillera. Vous arriverez sur le menu des jeux.

Sous le menu options, il est possible de voir les cibles, changer une cible d'adresse, effacer les scores enregistrés, ou régler le son et la sensibilité des cibles.

Pour lancer un jeu, il suffit de le sélectionner et presser le bouton OK, régler les options, et lancer le jeu. Les options sont enregistrées une fois réglées, permettant de relancer le jeu rapidement. Les différents écrans sont les suivants :



Liste des jeux

Les jeux et leurs options sont les suivants :

(02) Parcours

Le parcours est le jeu de base. Toutes les cibles sont allumées, à vous de créer un parcours physique pour jouer.

Vous pouvez y régler la couleur d'attente (lorsque la cible n'est pas en train de jouer), la couleur allumée (lorsque la cible attends d'être touchée), la couleur une fois touchée, ainsi que le temps de jeu maximum. Si ce temps est dépassé, les scores ne sont pas enregistrés.

Le jeu se termine lorsque toutes les cibles sont touchées, ou le temps réglé dépassé.

(03) Parcours + capteur

Le parcours, avec les capteurs enclenchés. Toutes les cibles sont allumées, à vous de créer un parcours physique pour jouer. Si vous êtes repérés par une cible et ne la touchez pas à temps, un malus sera compté.

Vous pouvez y régler les mêmes options que précédemment, en plus de la couleur capteur (lorsque la cible clignote quand elle vous a repéré), le temps de clignotement correspondant, et le type de fin de jeu.

Le jeu peut se terminer lorsque toutes les cibles sont touchées, le bouton appuyé, ou une cible spécifique touchée. Pour ces deux derniers choix, si des cibles n'ont pas été touchées, un malus sera compté.

(04) Aléatoire mobile

La cible à toucher change chaque tant de temps. Le but est de toutes les toucher le plus rapidement possible, avec la difficulté de repérer la cible allumée.

Les couleurs peuvent être réglées, en plus du temps qu'une cible reste allumée, du temps de jeu, ainsi que du nombre de cibles allumées simultanément.

Le jeu s'arrête une fois toutes les cibles touchées, ou le temps réglé dépassé.

(05) Aléatoire fixe

Semblable à l'aléatoire mobile, mais la cible ne change pas automatiquement. Elle attend d'être touchée.

(06) Bombe

Le mode bombe ressemble fortement à l'aléatoire fixe, mais rater la cible dans le temps indiqué met fin au jeu.

(07) Par pièces

Le mode par pièces simule l'entrée successive dans des pièces, allumant un nombre prédéfini de cibles simultanément.

Les couleurs peuvent être réglées, ainsi que le temps de jeu et le nombre de cibles par pièces.
Pour ces dernières, le principe est comme suit (exemple avec 4 cibles par pièces) :
cibles 1-2-3-4 s'allument, lorsque touchées 5-6-7-8 s'allument, 9-10-11-12 …

Si la cible X n'existe pas, pas de problème ! Imaginons que la cible 3 n'existe pas, la première pièce allumera les cibles 1-2-4, suivront 5-6-7-8 …

De même si une salle entière ne contient pas de cibles, elle sera sautée.
Il est dès lors possible d'imaginer plein de scénarios !

Le jeu se termine lorsque toutes les cibles ont été touchées.

(08) Par pièces + capteur

Comme précédemment, avec les capteurs activés. Si une cible vous repère et vous ne la touchez pas à temps, toutes les cibles de la pièce changeront de couleur, et un malus sera compté.

Il est aussi possible de régler le temps de détection, ainsi que la fin de jeu : bouton ou toutes les cibles touchées.
Si utilisé avec le bouton, des cibles ratées seront comptées comme malus.

(09) Duel (arbre)

Le but de ce mode est de mettre en compétition deux joueur, en allumant des cibles de deux couleurs, ces derniers devant les toucher le plus rapidement possible.

Il est possible de régler les couleurs pour les deux joueurs, ainsi que le nombre de cible et la cible de départ.
Pour correctement configurer ce mode, les cibles du joueur 1 sont la cible indiquée dans le réglage, ainsi que toutes celles suivant en fonction du nombre de cibles réglées.

Par exemple, le nombre de cibles est de 3, la première cible du j1 est la 4 => cibles j1 : 4-5-6
Cela fonctionne de la même manière pour le joueur 2.

Si les plages réglées s'entrechoquent (ex. j1 à les 3-4-5-6, et le j2 les 5-6-7-8), le système émettra un son d'erreur et le jeu ne sera pas lancé.

Le jeu se termine lorsque toutes les cibles sont touchées, pour les deux joueurs. Le temps de chacun est comptabilisé et comparé pour définir le meilleur des deux.

(10) Duel (rapidité)

Ce mode est simple : la première cible touchée met fin au jeu.

Il est ainsi possible de réaliser des concours de vitesses entre joueurs.

Accès rapide

Il est possible de lancer les jeux rapidement (pour autant que les réglages soient bons, et qu'au moins une cible soit branchée).
Les numéros précédents, à gauche des noms des jeux, sont le nombre d'appuis nécessaires pour le lancer.

Il suffit, en tout temps (même pendant un jeu), d'appuyer ce nombre de fois sur le bouton extérieur, branché sur le boitier principal ou sur une cible, ou faire de même avec la molette sur le boitier principal.

Autre

Le système se met en veille 3 minutes sans interaction de l'utilisateur, lorsqu'un jeu n'est pas lancé. Il est aussi possible d'éteindre le système en laissant appuyé 4 secondes la molette ou le bouton extérieur.

Les boitiers indiquent leur état de charge lorsque branchés au secteur, selon :

- Led verte : charge complète
- Led rouge : secteur branché
- Led jaune : charge en cours

Si les 3 leds sont allumées, cela signifie que le système est relié au secteur, tout en n'ayant pas de batterie branchée (recommandé si le système reste constamment branché).

La durée maximale d'un jeu est d'une heure. Après ce temps, le jeu est arrêté.

Si toutes les cibles sont déconnectées pendant un jeu, le système se rend au menu des jeux.

En cas de batterie faible, un écran est affiché, et il sera impossible de jouer sans recharge.
Si le contrôleur ne s'allume plus, la batterie est assez déchargée pour ne plus permettre l'écran d'avertissement. Il est nécessaire de recharger le système.

Scénarios de jeu

Les modes de jeu présentés précédemment peuvent être utilisés pour un large panel de scénarios.

Par exemple, le jeu *Duel (arbre)* peut permettre de créer deux parcours pour mettre en compétition deux joueurs simultanément.

Pour le jeu par pièces, il est possible de mélanger les cibles pour, lorsqu'une série est touchée, avoir des cibles qui s'allument devant et derrière le joueur.

A vous de concevoir vos meilleurs scénarios !

ANNEXE 10.5
Tests de fonctionnement

WhiteBox

Test	Mise sous tension
Etat de base	Eteint
Rés. Attendu	Système démarre, uC détecté par le programmeur
Rés. Obtenu	EQ
Appréciation	Ok

Test	Alimentation sans batterie
Etat de base	Pas de batterie
Rés. Attendu	3 leds de charge s'allument, circuit alimenté
Rés. Obtenu	EQ
Appréciation	Ok

Test	Alimentation avec batterie
Etat de base	Batterie déchargée
Rés. Attendu	Leds charge et branché s'allument, circuit alimenté
Rés. Obtenu	EQ
Appréciation	Ok

Test	Alimentation sans batterie
Etat de base	Batterie chargée
Rés. Attendu	Leds charge et branché s'allument, circuit alimenté
Rés. Obtenu	EQ
Appréciation	Ok

Test	Appui bouton
Etat de base	Appui sur retour/ok/encodeur/bouton ext., laisser appuyé
Rés. Attendu	Une seule interruption par appui (pas de rebonds)
Rés. Obtenu	EQ
Appréciation	Ok

Test	Bouger encodeur
Etat de base	Bouger encodeur gauche-droite
Rés. Attendu	Un seul événement par cran physique
Rés. Obtenu	EQ
Appréciation	Ok

Test	Affichage
Etat de base	Afficher texte, logo, rectangles, ronds, sliders
Rés. Attendu	Affichage selon réglages
Rés. Obtenu	EQ
Appréciation	Ok

Test	Son
Etat de base	Jouer une mélodie
Rés. Attendu	Son mélodie correct
Rés. Obtenu	EQ
Appréciation	Ok

Test	Son
Etat de base	Jouer une mélodie puis la couper avec une autre chanson
Rés. Attendu	Selon mode, n'est pas jouée ou remplace l'autre
Rés. Obtenu	EQ
Appréciation	Ok

Test	Rétroéclairage
Etat de base	Lumière changeante
Rés. Attendu	Luminosité s'adapte (moins fort dans le noir, fort sous le soleil)
Rés. Obtenu	EQ
Appréciation	Ok

Test	Watchdog
Etat de base	Créer une boucle infinie dans le programme
Rés. Attendu	Le système redémarre car gelé
Rés. Obtenu	EQ
Appréciation	Ok

Test	Pression sur l'encodeur
Etat de base	Appuyer X fois sur l'encodeur
Rés. Attendu	Détection correcte du nombre d'appuis
Rés. Obtenu	EQ
Appréciation	Ok

Test	Pression longue sur l'encodeur
Etat de base	Laisser appuyé l'encodeur
Rés. Attendu	Détection de l'appui long
Rés. Obtenu	EQ
Appréciation	Ok

Test	Pression sur le bouton extérieur
Etat de base	Appuyer X fois sur l'encodeur
Rés. Attendu	Détection correcte du nombre d'appuis, sur le contrôleur ou sur une cible
Rés. Obtenu	EQ
Appréciation	Ok

Test	Pression longue sur le bouton extérieur
Etat de base	Laisser appuyé le bouton extérieur
Rés. Attendu	Détection de l'appui long, sur le contrôleur ou sur une cible
Rés. Obtenu	EQ
Appréciation	Ok

Test	Réglage volume
Etat de base	Changement du volume dans les options
Rés. Attendu	Volume contrôleur change, message CAN émis
Rés. Obtenu	EQ
Appréciation	Ok

Test	Réglage sensibilité
Etat de base	Changement de la sensibilité dans les options
Rés. Attendu	Sensibilité change, message CAN émis
Rés. Obtenu	EQ
Appréciation	Ok

Test	CAN
Etat de base	Envoi de messages, pas de second nœud
Rés. Attendu	CAN Tx erreur -> reset le module pour continuer à travailler
Rés. Obtenu	EQ
Appréciation	Ok

Test	CAN
Etat de base	Envoi test, avec un second nœud
Rés. Attendu	Second nœud reçoit
Rés. Obtenu	EQ
Appréciation	Ok

Test	CAN
Etat de base	Envoi test en diffusion, avec 6 nœuds, câblage de 50 [m]
Rés. Attendu	Message reçu par tous les nœuds
Rés. Obtenu	EQ
Appréciation	Ok

Test	CAN
Etat de base	Charger la ligne avec de multiples messages, puis envoyer un message test
Rés. Attendu	Les messages sont délivrés, jusqu'au test
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu aléatoire
Etat de base	Lancement : les pointeurs sur cibles travaillent
Rés. Attendu	Les pointeurs s'arrêtent suivant les cibles disponibles, toutes les cibles sont traitées
Rés. Obtenu	EQ
Appréciation	Ok

Test	Sauvegarde flash
Etat de base	Enregistrer options de jeux et du système, redémarrer
Rés. Attendu	Les options réglées précédemment sont chargées, le système réglé avec
Rés. Obtenu	EQ
Appréciation	Ok

Test	Inactivité
Etat de base	Ne pas toucher au système, hors d'un jeu
Rés. Attendu	Système s'arrête
Rés. Obtenu	EQ
Appréciation	Ok

Test	Inactivité
Etat de base	Appuyer 4 secondes sur l'encodeur ou bouton extérieur, contrôleur ou cible
Rés. Attendu	Système s'éteint
Rés. Obtenu	EQ
Appréciation	Ok

Test	Sommeil
Etat de base	Système en sommeil
Rés. Attendu	Le système ne se réveille que par appui sur bouton (encodeur, ok, retour, extérieur) ou molette ou message CAN ; p.ex. un choc sur la cible ne la réveille pas
Rés. Obtenu	EQ
Appréciation	Ok

Test	Cible – lumières
Etat de base	Changer le pourcentage des couleurs
Rés. Attendu	Les PWMs changent de duty-cycle correctement
Rés. Obtenu	EQ
Appréciation	Ok

Test	Cible – détecteur
Etat de base	Laisser le capteur s'initialiser (1 minute), passer devant
Rés. Attendu	Interruption générée
Rés. Obtenu	EQ
Appréciation	Ok

Test	Choc
Etat de base	Sensibilité minimale, petit choc
Rés. Attendu	Interruption générée
Rés. Obtenu	EQ
Appréciation	Ok

Test	Choc
Etat de base	Changement des niveaux de sensibilité
Rés. Attendu	Chocs nécessaires pour interruptions plus forts
Rés. Obtenu	EQ
Appréciation	Ok

Test	Accéléromètre
Etat de base	Commandes I2C, module se configure correctement
Rés. Attendu	Set et lire les registres -> valeurs OK
Rés. Obtenu	EQ
Appréciation	Ok

Test	Debug
Etat de base	Brancher par USB à un PC
Rés. Attendu	Envoi et réception des caractères corrects
Rés. Obtenu	EQ
Appréciation	Ok

Test	Oscillateur
Etat de base	Système sur oscillateur -> clock switching avec désactivation de l'osc. externe
Rés. Attendu	Système marche toujours, et peux switcher sur l'osc. externe à nouveau
Rés. Obtenu	EQ
Appréciation	Ok

Test	Détection cibles
Etat de base	Cibles branchées
Rés. Attendu	En débranchant une cible, cette dernière est détectée absente ; en la rebranchant, retrouvée
Rés. Obtenu	EQ
Appréciation	Ok

Test	Doublons
Etat de base	Cibles branchées, dont doublon
Rés. Attendu	Détecter deux réponses de « la même » cible, avertir du doublon
Rés. Obtenu	EQ
Appréciation	Ok

Test	Cibles – débranchement
Etat de base	Cible branchée -> débrancher
Rés. Attendu	Au bout de 5 secondes, la cible s'éteint
Rés. Obtenu	EQ
Appréciation	Ok

Test	Cibles – débranchement
Etat de base	Cible branchée -> débrancher -> rebrancher
Rés. Attendu	Le contrôleur perd et retrouve la cible, la cible reste allumée
Rés. Obtenu	EQ
Appréciation	Ok

Test	Fixation par scratch
Etat de base	Fixer le système à l'aide de scratchs
Rés. Attendu	Système tient
Rés. Obtenu	Modules OK, cible ne tient pas bien avec du scratch « bas de gamme » -> 3M dual lock recommandé
Appréciation	Ok

Test	Fixation par aimants
Etat de base	Fixer le système à l'aide d'aimants
Rés. Attendu	Cible tient
Rés. Obtenu	Cible ne tient pas -> aimants trop peu puissants, en noyer d'autres dans le boitier
Appréciation	A revoir

Test	Fixation par vis
Etat de base	Fixer par vis la cible
Rés. Attendu	Cible tient
Rés. Obtenu	Cible tient fermement, même par grands chocs
Appréciation	Ok

Test	Cible – sens d'orientation – choc
Etat de base	Fixer la cible dans plusieurs orientations
Rés. Attendu	Capable de détecter les chocs malgré l'orientation
Rés. Obtenu	EQ
Appréciation	Ok

Test	Batterie basse
Etat de base	Régler le niveau d'alimentation à env. 3.35 [V]
Rés. Attendu	Niveau détecté -> système en sommeil
Rés. Obtenu	EQ
Appréciation	Ok

Test	Sommeil – longévité
Etat de base	Système chargé, mis en sommeil
Rés. Attendu	Doit marcher si allumé dans le temps estimé (ici 112 jours)
Rés. Obtenu	Impossible à tester dans le temps donnée
Appréciation	A tester

Test	Marche – longévité
Etat de base	Système chargé, sans alimentation, fonctionnement
Rés. Attendu	Doit marcher au moins le temps estimé (ici 16 [h])
Rés. Obtenu	N'a pas pu être testé dans les temps
Appréciation	A tester

Totaux

Tests effectués	44
Tests OK	40
Tests A REVOIR	1
Tests A CORRIGER	3

BlackBox

Test	Branchement système
Etat de base	Système câblé
Rés. Attendu	Cibles détectées et blanc doux, nombre de cibles correct
Rés. Obtenu	EQ
Appréciation	Ok

Test	Retrait d'un des doublon
Etat de base	Système allumé
Rés. Attendu	Le doublon branché revient blanc doux, le nombre de cibles détectées augmente, le doublon retiré s'éteint
Rés. Obtenu	EQ
Appréciation	Ok

Test	Ajout d'une cible
Etat de base	Système allumé
Rés. Attendu	Cible blanc doux, nombre de cibles détectées augment
Rés. Obtenu	EQ
Appréciation	Ok

Test	Défilement menu principal
Etat de base	Système câblé
Rés. Attendu	Défile haut-bas, la liste est continue lorsque le dernier menu est défilé
Rés. Obtenu	EQ
Appréciation	Ok

Test	Ajout d'une cible doublon
Etat de base	Système allumé
Rés. Attendu	Les deux doublons « clignotent » violet et sonnent, le nombre de cibles détectées diminue
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeux -> options
Etat de base	Menu des jeux, appui sur retour
Rés. Attendu	Menu des options s'affiche
Rés. Obtenu	EQ
Appréciation	Ok

Test	Options -> jeux
Etat de base	Menu des options, appui sur retour
Rés. Attendu	Menu des jeux s'affiche
Rés. Obtenu	EQ
Appréciation	Ok

Test	Changement volume/sensibilité
Etat de base	Menu des options, curseurs volume/sensi. par défaut
Rés. Attendu	Curseurs bougent, son silencieux avec curseur à 0, sensibilité change sur les cibles (test dans un mode de jeu)
Rés. Obtenu	EQ
Appréciation	Ok

Test	Options -> voir cibles
Etat de base	Menu des options, OK sur « Voir cibles », avec une cible branchée et deux cibles de même adresse branchées
Rés. Attendu	Cibles s'affichent, noir si pas connecté, vert si connecté, jaune si erreur
Rés. Obtenu	EQ
Appréciation	Ok

Test	Options -> changer adresse
Etat de base	Menu options, OK sur « Changer adresse »
Rés. Attendu	Affichage du changement d'adresse
Rés. Obtenu	EQ
Appréciation	Ok

Test	Voir cibles - défilement
Etat de base	Menu des cibles
Rés. Attendu	Les cibles défilent, au fond -> revient au début de la liste
Rés. Obtenu	EQ
Appréciation	Ok

Test	Chang. Adr. – aucune cible
Etat de base	Menu adresse, pas de cible câblée
Rés. Attendu	Pas de défilement dans le menu, indication qu'aucune cible disponible
Rés. Obtenu	EQ
Appréciation	Ok

Test	Voir cibles - boutons
Etat de base	Appui sur bouton ok, encodeur
Rés. Attendu	Pas de changement
Rés. Obtenu	EQ
Appréciation	Ok

Test	Chang. Adr. – changement
Etat de base	Menu adresse, cible câblée
Rés. Attendu	Changer de cible avec autre numéro, OK sur changer, cible change (contrôle avec voir cibles)
Rés. Obtenu	EQ
Appréciation	Ok

Test	Voir cibles -> options
Etat de base	Menu des cibles, appui sur retour
Rés. Attendu	Menu options s'affiche
Rés. Obtenu	EQ
Appréciation	Ok

Test	Chang. Adr. -> options
Etat de base	Menu adresse, bouton retour
Rés. Attendu	Menu options s'affiche
Rés. Obtenu	EQ
Appréciation	Ok

Test	Options -> effacer scores
Etat de base	Menu options, OK sur effacer scores
Rés. Attendu	Affichage avertissement scores
Rés. Obtenu	EQ
Appréciation	Ok

Test	Eff. Scores – effacer
Etat de base	Menu effacer, bouton ok
Rés. Attendu	Scores effacés (contrôle dans un jeu), son de confirmation
Rés. Obtenu	EQ
Appréciation	Ok

Test	Eff. Scores – annuler
Etat de base	Menu effacer, bouton retour
Rés. Attendu	Menu options s'affiche, scores toujours présents
Rés. Obtenu	EQ
Appréciation	Ok

Test	Options- défilement
Etat de base	Menu options
Rés. Attendu	Défilement haut-bas ok
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeux -> choisir un jeu
Etat de base	Menu jeux, sélectionner un jeu sans cibles branchées
Rés. Attendu	Options du jeu s'affichent
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeux -> choisir un jeu
Etat de base	Menu jeux, sélectionner un jeu avec au moins une cible
Rés. Attendu	Options du jeu s'affichent
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) - options
Etat de base	Options parcours
Rés. Attendu	Options jeu réglables
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) - lancement
Etat de base	Options parcours, lancement jeu, pas de cibles branchées
Rés. Attendu	Bip d'erreur, reste sur le menu des options de jeu
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) - lancement
Etat de base	Options parcours, lancement jeu, au moins une cible branchée
Rés. Attendu	Décompte 3-2-1 avec bips
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – en jeu
Etat de base	Jeu lancé, pas de cible touchée
Rés. Attendu	Cibles s'affichent en noir, temps s'écoule, cibles colorées selon réglage précédent
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – en jeu
Etat de base	Jeu lancé, appui sur retour
Rés. Attendu	Retour sur menu des jeux, cibles blanc doux
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – en jeu
Etat de base	Jeu lancé, 2 cibles, 1 se débranche
Rés. Attendu	1 cible noire, 1 cible grisée
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – en jeu
Etat de base	Jeu lancé, 2 cibles, toucher les 2 cibles
Rés. Attendu	Cibles changent selon réglage, fin du jeu (bip - affichage)
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu rapide
Etat de base	Au moins une cible, n'importe quel menu (même en jeu), X appuis sur encodeur
Rés. Attendu	Jeu correspondant chargé et lancé
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – fin
Etat de base	Jeu fini
Rés. Attendu	Scores affichés, nouveau score s'enregistre
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – fin
Etat de base	Jeu fini, 2ème essai
Rés. Attendu	Scores affichés (dont précédent), nouveau score s'enregistre
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – en jeu
Etat de base	Jeu lancé, 2 cibles, toucher une cible
Rés. Attendu	Couleur cible change, cible verte sur affichage
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) - options
Etat de base	Appui sur bouton retour
Rés. Attendu	Retour menu jeux
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – en jeu
Etat de base	Appui sur bouton retour
Rés. Attendu	Retour menu jeux
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – fin
Etat de base	Appui sur bouton retour
Rés. Attendu	Retour menu jeux
Rés. Obtenu	EQ
Appréciation	Ok

Refaire les tests avec les différents modes de jeu ***OK***

Test	Jeu (parcours) - lancement
Etat de base	Appui sur bouton retour
Rés. Attendu	Retour menu jeux
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu rapide
Etat de base	Cible branchée, X appuis sur bouton extérieur (branché au contrôleur)
Rés. Attendu	Jeu correspondant chargé et lancé
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu rapide
Etat de base	Cible branchée, X appuis sur bouton extérieur (branché à une cible)
Rés. Attendu	Jeu correspondant chargé et lancé
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours) – fin temps
Etat de base	Lancer jeux avec temps 15 [s], laisser dépasser
Rés. Attendu	Fin de jeu, scores invalides, timeout
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours – capteur) – fin bouton
Etat de base	Jeu lancé avec fin bouton, appuyer sur le bouton (contrôleur ou cible)
Rés. Attendu	Fin de jeu, malus comptés
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (parcours – capteur) – fin cible
Etat de base	Jeu lancé avec fin cible, toucher cible désignée
Rés. Attendu	Fin de jeu, malus comptés
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (chambres – capteur) – capteurs
Etat de base	Jeu lancé, se laisser détecter par un capteur, finir jeu
Rés. Attendu	Détection avec bips, cibles de la même chambre changent de couleur, malus en fin de jeu
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (duels / rapide) – fin cible quelconque
Etat de base	Jeu lancé, toucher une cible
Rés. Attendu	Jeu s'arrête dès la cible touchée
Rés. Obtenu	EQ
Appréciation	Ok

Test	Jeu (duels / arbre) – fin groupe 1 et 2
Etat de base	Jeu lancé, toucher toutes les cibles
Rés. Attendu	Fin de jeu après la dernière touchée, différence de temps entre les groupes
Rés. Obtenu	EQ
Appréciation	Ok

Test	Détection
Etat de base	Cible montée
Rés. Attendu	Selon sensibilité, capable de détecter uniquement le choc sur la cible
Rés. Obtenu	EQ
Appréciation	Ok

Totaux

Tests effectués	46
Tests OK	46
Tests A REVOIR	0
Tests A CORRIGER	0

ANNEXE 10.6

Commande électronique effectuée

Personne concernée:	Amand Axel			
Mandat no.:	Bachelor	Chef de projet / professeur:	Corre Jérôme	
Acronyme du projet:	STTS			
Salle:	A309 (2)	Délai désiré		

Mouser

https://www.mouser.ch/

Monnaie: CHF

Quantity	Reference	Designation	Unit Price	
12	863-SZNUP2105LT1G	SZNUP2105LT1G	0,27	3,26
16	534-254	254 Battery Clip - Contacts, pinces, sup	0,42	6,66
10	841-MMA8652FCR1	MMA8652FCR1	1,22	12,20
7	700-MAX3051ESAT	MAX3051ESA+T	2,08	14,56
10	621-AP9101CAK-AKTRG1	AP9101CAK-AKTRG1	0,32	3,18
8	579-MCP73833T-FCI/MF	MCP73833T-FCI/MF	0,81	6,49
10	621-AP7363-33D-13	AP7363-33D-13	0,43	4,31
8	895-FT230XS-R	FT230XS-R	1,95	15,60
8	579-512GM706-I/PT	DSPIC33EP512GM706-I_PT	5,88	47,04
10	798-ZX62D-B-5PA830	ZX62D-B-5PA8(30)	0,54	5,37
10	710-150060BS55040	150060BS55040	0,16	1,55
8	710-150060VS55040	150060VS55040	0,15	1,22
8	710-150060RS55040	150060RS55040	0,15	1,22
40	726-BSS806NH6327	BSS806NH6327XTSA1	0,25	9,96
10	621-DMG9926UDM-7	DMG9926UDM-7	0,38	3,79
15	726-BSS308PEH6327XTS	BSS308PEH6327XTSA1	0,27	4,04
10	81-NCP15XH103F03RC	NCP15XH103F03RC	0,12	1,19
10	520-5032MV-40-CNT	ECS-5032MV-40-CN-TR	0,74	7,43
10	78-LL4448-GS18	LL4448-GS18	0,12	1,21
1	763-24240320CFCSXNFT	NHD-2.4-240320CF-CSXN#-FT	18,66	18,66
2	538-54132-4062	541324062	1,72	3,44
10	490-CMT-850385BSMTTR	CMT-8503-85B-SMT-TR	1,24	12,40
4	612-JN2UOANAG	JN2UOANAGX	0,88	3,51
2	612-TAGGRN	TAGGRN	0,22	0,44
2	612-TAGRED	TAGRED	0,19	0,38
2	652-PEC12R-4130F-S12	PEC12R-4130F-S0012	1,16	2,32
2	706-11K5014-KCNG	11K5014-KCNG	0,76	1,53
10	604-WP154A43VBDZGWCA	WP154A4SEJ3VBDZGW/CA	1,44	14,40
10	941-P6CFKBCM1Q1H17R3	CLP6C-FKB-CM1Q1H1BB7R3R3	0,59	5,87
14	667-ESE-13V01A	ESE-13V01A	0,29	4,00

Distrelec

https://www.distrelec.ch/

Monnaie: CHF

Quantity	Reference	Designation	Unit Price	
25	143-02-006	S2B-XH-A (LF)(SN)	0,18	4,62
25	143-02-032	XHP-2	0,10	2,56
25	143-02-007	S3B-XH-A (LF)(SN)	0,22	5,54
25	143-02-033	XHP-3	0,15	3,84
25	143-02-009	S5B-XH-A (LF)(SN)	0,29	7,15
25	143-02-035	XHP-5	0,19	4,67
25	143-02-010	S6B-XH-A (LF)(SN)	0,28	7,00
25	143-02-036	XHP-6	0,22	5,60
250	301-21-619	SXH-001T-P0.6	0,03	8,15