

**CONCEPTION D'UN SYSTÈME D'ACQUISITION DE
DONNÉES DE CAPTEURS POUR UN BRAS ROBOTIQUE**

par

**CHAN KAO CHUNG, Arnold
MASSON, Charles
VIGNEAULT, Maxime**

**RAPPORT TECHNIQUE REMIS DANS LE CADRE DU PROJET DE
FIN D'ÉTUDES
PRÉSENTÉ À NOUMEIR, Rita**

MONTRÉAL, LE 14 AVRIL 2020

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC**

REMERCIEMENTS

En premier lieu, nous tenons à remercier Rita Noumeir d'avoir supervisé notre projet. Sans cette supervision, notre projet n'aura pas eu lieu. Son bagage de connaissances et ses conseils nous ont permis de mieux répondre aux besoins du client.

Ensuite, nous aimerais remercier les enseignants et enseignantes qui nous ont transmis leurs connaissances tout au long de notre parcours scolaire. Sans eux, nous n'aurions pas pu, dans ce projet, effectuer toutes les tâches et le développement technologique.

Nous remercions le club Capra d'avoir placé sa confiance en notre équipe afin d'obtenir la solution nécessaire pour leur robot. Ils ont mis à notre disposition tout leur équipement, en plus de leur soutien financier.

Nous remercions, enfin, le personnel du département de génie électrique, qui nous a prêté de l'équipement et nous a conseillés dans nos démarches.

CONCEPTION D'UN SYSTÈME D'ACQUISITION DE DONNÉES DE CAPTEURS POUR UN BRAS ROBOTIQUE

CHAN KAO CHUNG, Arnold

MASSON, Charles

VIGNEAULT, Maxime et al.

RÉSUMÉ

Le rapport technique vise à résoudre l'intégration de multiples capteurs. En effet, le club Capra de l'École de Technologie Supérieure travaille sur un robot qui permettrait de sauver des vies dans des zones sinistrées. Les robots sont testés dans différents environnements lors d'une compétition. Le club nous a mandatés pour intégrer les capteurs qui sont présents sur le bras du robot. Ceux-ci incluent un capteur de CO₂, une caméra numérique, une caméra thermique et un microphone. En plus, le bras devra conserver sa connexion avec le manipulateur. Ce rapport s'adresse à toute personne travaillant en robotique autonome et qui doit intégrer plusieurs capteurs.

Afin de participer à la compétition *RoboCup Rescue*, le client va investir dans de nouveaux capteurs pour réussir les nouvelles épreuves. Lors des années précédentes, le club a utilisé quelques capteurs qui pourront être réutilisés. Les spécifications de ces derniers correspondent aux divers requis de la compétition. De plus, le club possède des documents et des codes concernant ses anciens capteurs. Dans ce projet, nous allons choisir les un microphone et une caméra numérique manquants selon les requis de la compétition.

Nous analysons deux architectures possibles pour notre projet. La première est l'utilisation d'un ordinateur sur la carte qui va s'occuper de gérer les données de chacun des capteurs. La deuxième solution est de rendre chacun des capteurs intelligents et ainsi de confier la gestion des requêtes à l'ordinateur central. Cela veut dire que l'intelligence est décentralisée.

Finalement, nous avons donc choisi la deuxième solution. Cela permettra de faciliter l'intégration des besoins qui pourrait survenir dans le futur. Afin de combler les besoins du client pour la compétition les capteurs suivants ont été choisis : Telaire T6713, Point Grey CMLN-13S2C, FLIR Lepton 3.5 et le Digilent inc. Pmod Mic3. En effet, ces capteurs respectent les requis du client quant à la qualité, la sensibilité et le prix. De plus, ces différents capteurs s'intègrent bien au protocole USB et l'architecture permet de ne pas ajouter une masse importante au bras. Les capteurs seront connectés sur une carte de traitement qui agit comme concentrateur USB.

Mots-clés: Robotique, Acquisition, ROS, Capteurs, Détection

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 MISE EN CONTEXTE, PROBLÉMATIQUE ET OBJECTIFS	3
1.1 Mise en contexte	3
1.2 Explication de la problématique	4
1.3 Désignation des objectifs	4
CHAPITRE 2 REVUE DE LA DOCUMENTATION	5
2.1 Les capteurs	5
2.2 Le protocole Inter IC	5
2.3 Le périphérique USART	5
2.4 Le standard RS485	6
2.5 Serial Parallel Interface	6
2.6 Universal Serial Bus	6
2.7 Robot Operating System	6
2.8 Capteur de dioxyde de carbone	7
2.9 Caméra thermique	7
2.10 Microphone	7
CHAPITRE 3 MÉTHODOLOGIE DE TRAVAIL	8
3.1 La conceptualisation	8
3.1.1 Les capteurs	8
3.1.2 L'architecture	9
3.2 La réalisation	9
3.2.1 Création d'un banc d'essai	9
3.2.2 Script de programmation	10
3.2.3 Conception des schémas électriques	10
3.2.4 Conception du circuit imprimé	11
3.3 Les essais	11
3.3.1 Les convertisseurs	12
3.3.2 Les capteurs	12
3.3.3 Le concentrateur	13
3.3.4 L'essaie final	14
CHAPITRE 4 PROCESSUS DE CONCEPTION	15
4.1 Élaboration de solution	15
4.1.1 Sélection de la caméra	15
4.1.2 Sélection du microphone	16
4.1.3 Choix de canal	16

4.1.4	Architecture	17
4.1.5	Convertisseur	18
4.1.6	Circuit imprimé	18
4.1.7	Intégration logicielle	19
4.2	Analyse de risque	20
4.2.1	Canal	20
4.2.2	Convertisseur	21
4.2.3	Le circuit imprimé	22
4.2.4	Logiciel	24
4.3	Outils et techniques	26
4.4	Intégration des concepts retenus	26
4.5	Réalisation de tâches spécifiques	27
CHAPITRE 5 RÉSULTATS ET DISCUSSIONS		28
5.1	Résultats	28
5.1.1	Capteur de dioxyde de carbone	28
5.1.2	Les caméras	31
5.1.3	Le microphone	31
5.1.4	Les schémas électriques et la conception du circuit imprimé	32
5.2	Discussions	35
CONCLUSION		36
ANNEXE I	MATRICES DE DÉCISION	37
ANNEXE II	CODE CAPTEUR DE DIOXYDE DE CARBONE "CAP_CO2.PY"	38
ANNEXE III	CODE CAPTEUR DE SON "CAP_MIC.PY"	39
ANNEXE IV	CODE CAMÉRA NUMÉRIQUE "CAMNUM.C"	40
ANNEXE V	CODE CAMÉRA THERMIQUE	43
ANNEXE VI	SCHÉMATIQUES ALTIUM DESIGNER	44
ANNEXE VII	LISTE DES COMPOSANTS	52
BIBLIOGRAPHIE		54

LISTE DES TABLEAUX

	Page	
Tableau 4.1	Analyse de risques du canal.....	20
Tableau 4.2	Analyse de risques pour les convertisseurs.....	21
Tableau 4.3	Analyse de risques pour le PCB	23
Tableau 4.4	Analyse de risques logiciel.....	24
Tableau 5.1	Évolutions de la concentration de CO ₂ dans la pièce.....	30

LISTE DES FIGURES

	Page
Figure 1.1 Exemple d'une boîte d'évaluation	3
Figure 3.1 Exemple d'un essai avec un microcontrôleur.....	12
Figure 3.2 Exemple d'un essai d'un capteur USB.....	13
Figure 3.3 Exemple d'un essai d'un capteur avec un convertisseur	13
Figure 3.4 Exemple d'un essai avec un concentrateur	13
Figure 4.1 Architecture proposée	17
Figure 4.2 Planification des couche pour le PCB.....	18
Figure 4.3 Planification des zones pour le PCB	19
Figure 4.4 Graphique de l'impact du choix du canal par rapport à la probabilité	20
Figure 4.5 Graphique de l'impact du choix du convertisseur par rapport à la probabilité	22
Figure 4.6 Graphique de l'impact de la conception du PCB par rapport à la probabilité	23
Figure 4.7 Graphique de l'impact de la conception logiciel par rapport à la probabilité	25
Figure 4.8 Capteurs avec les interfaces proposés.....	27
Figure 5.1 Courbes d'évolution du taux de CO ₂ dans une pièce	29
Figure 5.2 Capture d'un code QR	31
Figure 5.3 Vue du dessus en 3D du PCB	33
Figure 5.4 Vue du dessous en 3D du PCB	33
Figure 5.5 Vue de la conception du PCB	34

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CO ₂	Dioxyde de carbone.
CAN	Convertisseur analogue numérique
CLK	Horloge, dérivé de l'anglais : Clock
CS	Sélection de la puce, dérivé de l'anglais : Chip select
EEPROM	Mémoire morte effaçable électriquement et programmable, dérivé de l'anglais Electrically-Erasable Programmable Read-Only Memory
ESD	Décharge électrostatique, dérivé de l'anglais Electrostatic discharge
I ² C	Protocole entre puce, dérivé de l'anglais : Inter IC
I ² S	Protocole de son entre puce, dérivé de l'anglais : Inter-IC Sound
IMU	Unité de mesure d'inertie, dérivé de l'anglais : Inertial measurement unit
LIDAR	Détection et estimation de la distance par la lumière, dérivé de l'anglais : Light detection and ranging
MISO	Maître en entrée, esclave en sortie, dérivé de l'anglais Master In Slave Out.
MOSI	Maître en sortie, esclave en entrée, dérivé de l'anglais Master Out Slave In.
NZRI	Non retour à zéro inversé, dérivé de l'anglais : Non Return to Zero Inverted
PC	Ordinateur personnel, dérivé l'anglais : Personal Computer
PCB	Circuit imprimé, dérivé de l'anglais Printed Circuit Board
PWM	Modulation en largeur de pulse, dérivé de l'anglais Pulse Width Modulation
ppm	Unité de mesure Parti Par Million
QR	Réponse rapide, dérivé de l'anglais Quick Response

RAM	Mémoire vive, dérivé de l'anglais Random Access Memory
ROS	Système d'opération robotique, dérivé de l'anglais : Robot Operating System ¹
SCL	Horloge sérielle, dérivée de l'anglais : Serial Clock
SDA	Données sérieelles, dérivée de l'anglais : Serial Data
SPI	Interface Serielle et parallèle, dérivé de l'anglais : Serial Parallel Interface
USART	Transmetteur et recepteur synchrone et asynchrone universel, dérivé de l'anglais : Universal Synchronous & Asynchronous Receiver Transmitter
USB	Bus série universel, dérivé de l'anglais Universal Serial Bus.

1. <https://www.ros.org/>

INTRODUCTION

Présentation du partenaire

Le projet, dont ce rapport fait l'objet, a été présenté au département de génie électrique par le club étudiant Capra. Ce dernier se positionne comme étant un club scientifique avec la mission de repousser les limites de la robotique de terrain. Ils produisaient, avant, des robots autonomes se déplaçant dans divers chemins. Cependant, les membres ont constaté qu'il n'y avait plus vraiment de défi et de conception intéressante. Depuis trois ans, le club a fait la transition vers les robots de sauvetage.

Cette année, le client participera, pour une deuxième année consécutive, à la RoboCup Rescuse. Celle-ci oppose différentes écoles et chaires de recherches à travers le monde qui travaillent dans le domaine de la robotique. Chacune des équipes se présente avec un robot qui sera mis à l'épreuve durant plusieurs jours. L'objectif principal de la compétition est de simuler les conditions liées à un sinistre. Les robots doivent se déplacer dans les environnements désignés, de façon télécommandée ou de façon autonome. Un maximum d'autonomie est recherché, donc des points supplémentaires sont accordés aux robots qui sont entièrement autonomes. Actuellement, le robot du club Capra possède un LIDAR qui permet de créer des cartes. Il possède deux caméras 3D, dont une est positionnée dans la partie avant du châssis, et l'autre dans la partie arrière. Finalement, un capteur IMU permet de connaître l'angle du robot.

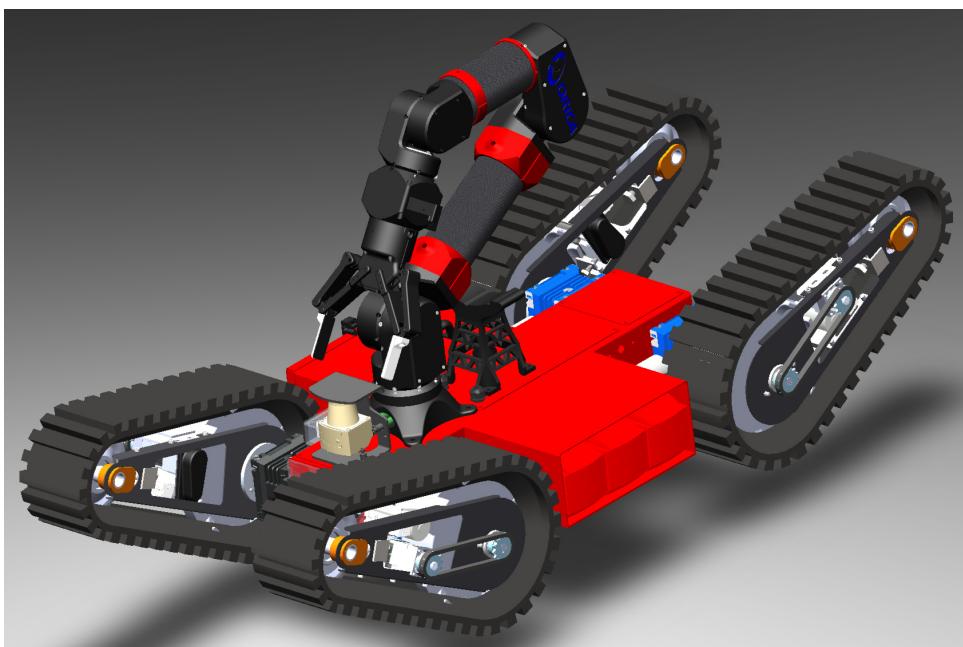
Le robot

L'objectif du club est de fabriquer un robot portable, ayant une autonomie notable. Cela permet de le déployer dans plusieurs situations qui ne seraient pas adaptées aux êtres humains.

Une nouvelle génération de robot est en cours de conception. On observe, à la figure présenté à la fin de cette section (page 2), l'aperçu en trois dimensions de la nouvelle plate forme.

On peut décomposer le robot en quelques éléments principaux. On y retrouve un système de locomotion. Celui-ci est quatre chenilles mobilisées en extrémités. Le système de locomotion permet de déplacer le châssis. Celui-ci contient les batteries qui alimentent le robot, ainsi que l'électronique de gestion. Dans cette électronique, on compte un ordinateur de bord qui agit à titre d'intelligence principale.

Enfin, le robot équipe un bras. Ce dernier permet de faire différentes manipulations à l'aide de son manipulateur. C'est également sur ce bras que nous trouvons les différents capteurs (CO_2 , caméra numérique, caméra thermique et microphone).



Modèle 3D du robot

CHAPITRE 1

MISE EN CONTEXTE, PROBLÉMATIQUE ET OBJECTIFS

1.1 Mise en contexte

Tel que mentionné dans l'introduction, le robot doit se déplacer dans un environnement simulant une catastrophe. Ainsi, afin de mettre à l'épreuve les capacités visuelles du robot, des codes QR ainsi que des panneaux signalétiques sont posés tout au long des parcours. Il doit les reconnaître et les enregistrer. Le robot doit être en mesure de porter secours à une personne en difficulté. Pour se faire, l'épreuve remplace la victime par une émission audio dans laquelle certains points clés sont émis et doivent être captés. Il détecte également la respiration humaine. Enfin, il doit être capable de trouver les sources de chaleur dans son environnement. Afin de tester cette capacité, un élément chauffant est placé sur des symboles (C de type *Landolt*). Le système de vision doit détecter le plus petit des symboles. Sur la figure 1.1, nous pouvons voir une des boîtes contenant les différents symboles, utilisés lors des anciennes compétitions. Les symboles et la source audio se situent à 40 cm à l'intérieur de cette boîte.



Figure 1.1 Exemple d'une boîte d'évaluation

L'équipe de conception du robot a décidé de placer les capteurs nécessaires sur le bras du robot. Cela permet, entre autres, d'approcher les capteurs à la source et d'obtenir de meilleures mesures.

1.2 Explication de la problématique

Les capteurs se situent sur le bras du robot. Lors de la précédente compétition, ils étaient connectés à l'ordinateur central à l'aide d'un système de communication sans fil. Cependant, le règlement de la compétition impose qu'une seule bande sans-fil soit utilisée. Étant donné que le poste de commande utilise déjà la bande permise, la transmission des données des capteurs doit être filaire. En considérant que la solution finale devra être monté entre le dernier joint et le manipulateur, une solution compacte devra être mise sur pied. Connais-sant cette position, le projet devra s'assurer de garder la connexion avec le manipulateur. Le client possède déjà un capteur de CO₂ et une caméra thermique. Il ne possède pas les autres capteurs nécessaires, il faudra les choisir. Nous savons déjà que les capteurs transmettent les informations à l'aide divers protocoles de communications. Il est impossible d'avoir un seul et unique protocole qui sera commun à tous les capteurs. En plus, les capteurs ne transmettent pas à la même vitesse et l'information de certains est plus importante que d'autres. Lors des différentes épreuves, la vitesse d'acquisition rapporte plus de points.

1.3 Désignation des objectifs

L'objectif de ce projet de fin d'études est de créer une connexion simplifiée entre le manipulateur, les capteurs et l'ordinateur de bord du robot du club Capra. Nous souhaitons proposer une solution avec une interface universelle. Considérant qu'elle sera dans un robot, notre solution devra être résistante aux bruits électromagnétiques. Nous souhaitons, également, proposer une solution qui permettra de prioriser les données d'un capteur selon le moment.

Spécifiquement au niveau des capteurs, il faut que chacun des capteurs opère à 81 cm de distance. Au niveau de la caméra numérique, sa résolution doit pouvoir lire des étiquettes décrites précédemment. La caméra thermique doit identifier des C de type *Landolt* possé-dant un espace de 2 cm. Un capteur de CO₂ doit pouvoir détecter le souffle humain. Enfin, un microphone doit permettre d'entendre la voix humaine. Il est préférable que ce dernier possède un gain variable afin de l'adapter aux différents environnements.

CHAPITRE 2

REVUE DE LA DOCUMENTATION

2.1 Les capteurs

Il existe plusieurs types de capteurs. Certains font des acquisitions analogiques, d'autres font des acquisitions numériques. Le premier type requiert un convertisseur de données analogues vers des données numériques afin de pouvoir transmettre les données. Cependant, de plus en plus de capteurs sont disponibles avec une telle interface intégrée dans leur assemblage. Ainsi, les capteurs sont régulièrement équipés d'une interface telle que :

1. I²C
2. I²S
3. USART
4. SPI
5. USB
6. ROS

2.2 Le protocole Inter IC

Ce protocole est plus connu sous son acronyme I²C. Il utilise deux lignes de communication. Tel qu'écrit dans l'International Journal of Research in Advent Technology¹ à la page 475, le bus I²C se compose physiquement de deux fils dédiés au bus et d'une connexion à la masse. Les deux fils actifs sont l'horloge (SCL) et la communication (SDA). La communication est bidirectionnelle. L'information est transportée entre les appareils connectés au bus. Chaque appareil possède une adresse unique. C'est avec cette adresse qu'on peut choisir le récepteur du message.

2.3 Le périphérique USART

USART ou Universal Synchronous & Asynchronous Receiver Transmitter est un circuit électronique qui permet de mettre en série les bits à transmettre. Les bits sont transmis les uns à la suite des autres à l'aide de deux fils. Cela signifie que les bits sont transmis les uns après les autres grâce à deux fils. Les bits sont créés par la différence de potentiel entre le signal et la masse.

1. Mankar, J., Chaitali, D., Komal, T., Madhura, K. & Prachi, S. (2014)

2.4 Le standard RS485

Tel que décrit dans le manuel d’acquisition de données² à la page 420, le standard RS485 est une transmission symétrique. Il est principalement caractérisé par son mode différentiel. Il est très commun dans les systèmes modernes. Il peut être connecté sur un UART ou un USART.

2.5 Serial Parallel Interface

Tel que décrit dans l’article *Introduction to SPI*,³ il s’agit d’un bus de données séries synchrones. Il utilise le mode bidirectionnel sans alternance. Les appareils utilisent le principe de communication maître-esclave. Les maîtres et les esclaves peuvent transmettre de l’information en même temps. Ce protocole utilise quatre fils pour transmettre ses quatre signaux :

- L’horloge. (SCLK)
- La sélection de la puce. (CS)
- Les données du maître vers les esclaves. (MOSI)
- Les données des esclaves vers le maître. (MISO)

2.6 Universal Serial Bus

Tel que décrit dans l’article *USB 101 : An Introduction to Universal Serial Bus 2.0*,⁴ l’USB utilise quatre câbles. On y trouve le V_{BUS} (4.01–5.25 volts), la ligne de donnée négative (D^-), la ligne de donnée positive (D^+) et la mise à la masse. Le V_{BUS} est toujours fourni par l’hôte. Il est utilisé pour recharger ou alimenter des appareils. L’interface USB utilise une transmission différentielle qui est *non return-to-zero inversé* (NRZI) encodé avec embout à travers la paire torsadée.

2.7 Robot Operating System

Le Robot Operating System⁵ ou ROS est un ensemble de bibliothèques, de logiciels et d’outils qui permettent de créer des applications pour les robots. Ces bibliothèques contiennent des pilotes et des algorithmes qui aident aux développements. ROS est en développement ouvert (*open source*), il est populaire dans la programmation en robotique.

-
- 2. Asch, G. (2011)
 - 3. Piyu, D. (2018)
 - 4. Robert, M. (2017)
 - 5. <https://www.ros.org/>

2.8 Capteur de dioxyde de carbone

La détection de CO₂ est principalement fait par deux méthodes. La première est l'utilisation de capteurs infrarouges. La deuxième est par captation chimique.

Lorsque le capteur équipe la technologie infrarouge, un rayonnement infrarouge est émis par le capteur. S'il y a présence de CO₂, ce dernier absorbera une partie de la lumière émise. Ce phénomène entraîne une modification du signal infrarouge, il est alors possible de déduire la concentration de CO₂ dans l'air ambiant en fonction la distorsion de ce signal.

Autrement, lorsque le capteur est basé sur la captation chimique, il possède une couche de détection. Quand cette dernière fait contacte avec du CO₂, sa composition change et il est possible d'en interpréter sa composition ainsi. L'unité de mesure pour la concentration se fait en partie par million (ppm).

2.9 Caméra thermique

La caméra infrarouge enregistre les différents rayonnements infrarouges émis par les corps ou les objets. Une fois l'énergie infrarouge détectée, il l'a convertie en signal électronique. Cette dernière est par la suite traitée pour produire une image thermique sur un moniteur. La caméra affiche également les différents calculs de température.

2.10 Microphone

Il existe deux types de microphones. Le premier est le microphone dynamique dont la membrane est fixée sur une bobine. Les ondes sonores font vibrer la membrane ce qui engendre un mouvement de la bobine. Ce mouvement génère un signal électrique qui sera par la suite interprété. Ce type de microphone ne requiert pas d'alimentation auxiliaire, puisque c'est la bobine qui génère le signal. Le deuxième type de microphone est le microphone électrostatique. Cette fois-ci, la membrane fait vibrer l'armature d'un condensateur. La variation de la distance de l'armature du condensateur fait varier la valeur de cette dernière. Les armatures sont soumises à une différence de potentiel, la variation de la capacité entraîne une variation du courant. Une résistance est placée en série afin de transformer la variation de courant en variation de tension. Ce microphone nécessite une alimentation auxiliaire, ce qui le rend plus sensible au bruit.

CHAPITRE 3

MÉTHODOLOGIE DE TRAVAIL

Dans cette section, nous présenterons la méthodologie utilisée pour réaliser le projet. Nous avons trois phases principales, soit la conceptualisation, la réalisation ainsi que les tests. Pour chacune d'entre elles, nous expliquerons ce qui doit être accompli, en plus de ce qui est nécessaire afin d'atteindre nos objectifs. Nous expliquerons, lorsque nécessaire, les raisons pour lesquelles nous avons privilégié certaines solutions plutôt que d'autres.

3.1 La conceptualisation

La conceptualisation est l'étape durant laquelle nous planifions ce que nous devons faire. Ce sont les résultats de cette étape qui nous permettront de faire des réalisations plus tard. L'important, ici, est d'élaborer des concepts et de définir les connaissances requises.

3.1.1 Les capteurs

La première étape que nous avons défini est de faire la recherche des différents capteurs qui correspondent aux spécifications de la compétition. Nous analyserons, dans un premier temps, les capteurs que nous avons à notre disposition. Nous devons être certains que les capteurs déjà en place sont adéquats. Pour les capteurs manquants, nous dresserons une liste des choix potentiels et nous choisirons en fonction des différents requis. La méthode que nous avons retenue afin de faire des choix est la matrice de décision.

Le capteur de CO₂ doit être capable de détecter la respiration d'un être humain à moins d'un mètre. Nous nous attendons donc à une élévation distincte de la concentration de dioxyde de carbone détecté par le capteur lorsque nous respirons à proximité. La caméra thermique doit pouvoir déterminer une source de chaleur à travers une boîte possédant une ouverture de 2 cm par 2 cm et située à 70 cm du capteur. La caméra couleur doit lire un code QR et discerner des symboles de matières dangereuses à une distance minimale de 1 mètre. Quant au microphone, il doit pouvoir transmettre distinctement une suite de chiffres dictés par une voix humaine. Pour pouvoir comprendre les limites de chacun de ces capteurs, il est essentiel de lire les fiches techniques. En effet, certains de ces capteurs nécessitent une source extérieure ou des composants supplémentaires afin d'assurer leurs fonctionnements ou pour les configurer.

Certains capteurs peuvent communiquer avec plusieurs protocoles de communication, cette étape nous permettra de choisir le bon protocole selon sa robustesse, sa difficulté de configuration et ses documents disponibles. Moins les protocoles de communication sont différents, moins notre conception sera compliquée et plus nous diminuerons nos chances d'erreurs.

Pour cette étape, il faut avoir un bon sens de l'organisation. Nous avons ainsi créé un classeur de calcul afin d'y ajouter les différentes informations des capteurs. Nous avons préféré choisir cette option plutôt d'imprimer les fiches techniques par soucis pour l'environnement. En plus, nous pouvions aisément partager l'information entre les membres de l'équipe.

3.1.2 L'architecture

Une fois que les capteurs seront sélectionnés, nous pourrons choisir une architecture de communication. Nous listerons les protocoles de communication des capteurs choisis. De cette liste, nous ferons ressortir les différences entre les protocoles des capteurs. Cela nous permettra de faire le choix pour l'architecture appropriée. En plus, nous porterons une attention particulière à l'existence d'une carte d'évaluation pour la solution choisie.

Si les protocoles sont trop lents ou trop sensibles aux bruits, nous devrons changer le capteur qui y est assigné. De plus, la majorité du code dans le robot implique de la scrutation, l'architecture devra donc être compatible avec ce type de transfert. Une analyse de ce qui existe sur le marché est de mise.

Nous dresserons une liste d'architecture potentielle. Des outils de dessin graphiques tels que Visio ou DrawIO permettent de faciliter l'explication des architectures. Leur aspect professionnel leur donne un avantage, puisqu'ils peuvent être partagés et intégrés dans les rapports. Nous pourrons ainsi analyser et présenter aux clients les avantages et inconvénients de chacune des solutions.

3.2 La réalisation

L'étape de réalisation est la mise en pratique de nos concepts prévue à l'étape précédente. Nous commençons à élaborer la solution basée sur nos recherches et faisons des essais préliminaires. Il s'agit d'une étape cruciale, puisque nous pouvons avancer avec les concepts sélectionnés, ou devons retourner évaluer d'autres concepts.

3.2.1 Crédit d'un banc d'essai

Un banc d'essai dans notre projet consiste à assembler des cartes de développement, des sources de tension de laboratoire, et d'autres composants. L'objectif est de répliquer l'architecture sélectionnée en version non miniaturisée et qui est facile de diagnostiquer. Cela permet de vérifier rapidement si l'architecture choisie est une solution appropriée.

Ce banc d'essai va également pouvoir servir pour le développement logiciel. En effet, il ne sera pas nécessaire d'attendre la production d'un circuit imprimé avant de développer la

partie logicielle. De plus, si des délais nous empêchaient de commander le circuit, ou qu'il n'était pas fonctionnel, nous pourrions utiliser le banc d'essai pour démontrer la fonctionnalité du projet.

Nous aurons besoin d'un oscilloscope et d'un analyseur logique (Saleae Logic 8 Pro) pour déterminer les signaux.

3.2.2 Script de programmation

Il existera un script par capteur que nous intégrons. Ainsi, nominalement, quatre scripts existeront. Les scripts devront être compatibles avec l'ordinateur de bord. Celui-ci embarque une distribution de *ROS Melodic Morenia*.

Nous utiliserons une approche séquentielle. C'est-à-dire que nous développerons un seul script à la fois. Cette solution est logique, puisque ROS fonctionne avec des noeuds. Chacun des noeuds est indépendant des autres noeuds. Ainsi, la conception logicielle peut être divisée.

Pour y arriver, nous devrons utiliser un ordinateur qui a Ubuntu 18.04 comme système d'exploitation. Nous devrons nous assurer que ROS y est installé. Nous aurons également besoin de connaissance de base en programmation ainsi qu'un environnement pour éditer le texte. Les compilateurs ou interpréteurs appropriés, selon les langages, seront installés au besoin.

3.2.3 Conception des schémas électriques

Les schémas électriques permettent de mettre en relation les différents composants d'un circuit. À cette étape, nous devrons bien définir les tensions d'alimentations. Ensuite, suivant les recommandations des manufacturiers, nous allons ajuster les différents composants autour de la puce (résistances, condensateurs, etc.) Nous allons en profiter pour noter ce qui sera requis lors de la conception du circuit imprimé.

Nous allons essayer de découper autant que possible les circuits. Cela permet de faciliter la lecture des plans et permet de créer des pages dédiées à certains circuits importants. C'est également dans cette étape que nous allons définir les pièces utilisées. Par exemple, nous trouverons un manufacturier et un numéro de modèle pour une résistance dont nous avons besoin. Nous intégrerons ensuite les informations dans les différents schémas.

Nous aurons besoin d'utiliser nos connaissances en électronique afin de juger les solutions les plus optimales. En plus, nous utiliserons un système de révision par les paires. Cela permettra de capter un maximum d'erreur avant de faire un système sur une carte.

Pour réaliser la version numérique des schémas, nous utilisons le logiciel Altium Designer 19. Altium Designer est un logiciel qui permet de faire la conception d'un circuit imprimé du début à la fin. Ainsi, il permet également de concevoir la plaque sur laquelle reposeront nos circuits.

3.2.4 Conception du circuit imprimé

Avec les schémas électriques maintenant assemblés dans le logiciel, nous pourrons poursuivre la conception du circuit imprimé. Nous devrons maintenant construire une bibliothèque de pièces. Les bibliothèques sont construites en deux parties. La première est le symbole que l'on retrouve sur les schémas. La deuxième est l'empreinte de la pièce afin de pouvoir l'utiliser sur le PCB. La première partie étant déjà mise en place lors de la conception des schémas, nous construirons uniquement la deuxième partie. Nous allons créer manuellement certaines pièces, en nous assurant de respecter les recommandations IPC et celles du manufacturier.

Ensuite, nous évaluerons les différents manufacturiers de PCB. Basées sur notre choix, nous pourrons créer certaines règles dans le logiciel afin de vérifier la conception. Par exemple, l'espace minimal entre les traces ou la grosseur minimale des trous sont des paramètres assez communs.

Puis, nous pourrons définir une première version des dimensions de la plaque. Cette dimension sera appelée à changer s'il est possible d'optimiser l'espace. Enfin, nous essayerons de faire un premier placement des composants.

Les signaux circuleront à des vitesses élevées dans le PCB. Nous devrons calculer la taille et la longueur des traces afin que le signal ne soit pas corrompu. Il est également important de faire attention à la densité des composants ainsi qu'à l'impédance des traces, puisque ces dernières influencent l'intégrité du signal. Nous allons également prévoir des points tests sur les zones critiques afin de vérifier le bon fonctionnement du prototype.

Des connaissances quant à la définition des étages de cuivre et leur diélectrique, la création de paires différentielles ou le contrôle de l'impédance seront requis. De plus, une bonne maîtrise du logiciel de conception Altium Designer est importante.

3.3 Les essayes

Cette section confirme l'atteinte ou non de nos objectifs. Pour vérifier certains éléments, nous utiliserons le banc de test, alors que pour d'autres, nous utiliserons le circuit imprimé. Par exemple, l'essai des capteurs avec leur convertisseur ne requiert pas nécessairement la carte, puisque le banc d'essai peut fournir l'information nécessaire. Cependant, vérifier si l'intégrité du signal est respectée requiert la carte finale assemblée.

3.3.1 Les convertisseurs

Nous aurons sélectionné les convertisseurs selon les critères de compatibilités avec le protocole du capteur. À cette étape, nous allons vérifier que le convertisseur en question est en mesure de bien prendre des données dans un protocole, et les convertir en USB.

Nous configurons les convertisseurs en utilisant le système d'exploitation Ubuntu avec ROS. En effet, afin de nous rapprocher des conditions réelles, nous utiliserons le même OS que l'ordinateur central.

Étant donné que certains capteurs peuvent nécessiter un peu plus de travail que d'autres, nous pouvons utiliser un microcontrôleur pour les essais. En connectant un microcontrôleur pour simuler un capteur, nous pouvons contrôler les données qui sont envoyées. Cela permet de vérifier l'intégrité des données reçues.

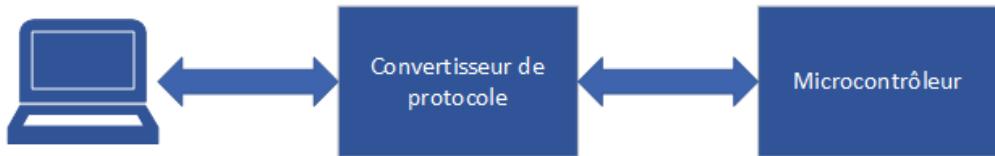


Figure 3.1 Exemple d'un essai avec un microcontrôleur

3.3.2 Les capteurs

Lors de cette étape, nous ferons divers tests afin de vérifier que les différents capteurs respectent les spécifications requises. Nous allons utiliser un ordinateur pour simuler l'ordinateur central du robot, nous testerons chacun des capteurs individuellement.

Nous allons d'abord tester les différents capteurs ayant une connexion USB native. Pour y arriver, nous allons le connecter sur l'ordinateur et vérifier que ce dernier le reconnaîsse comme un périphérique. Ensuite, il faudra utiliser divers logiciels afin de récupérer les informations émises par le capteur.

Pour les autres capteurs, nous utiliserons les montages créés dans le banc d'essai. Chacun des capteurs sera testé afin de vérifier leurs caractéristiques réelles.

Si la connexion entre le convertisseur et le PC était défaillante, nous avons d'autres techniques pour vérifier si le capteur est bon. Nous pouvons, par exemple, relier un capteur à un microcontrôleur (Ex. : Arduino) ou à un micro-ordinateur (Ex. : Raspberry Pi).

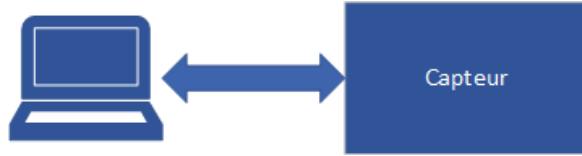


Figure 3.2 Exemple d'un essai d'un capteur USB



Figure 3.3 Exemple d'un essai d'un capteur avec un convertisseur

3.3.3 Le concentrateur

Une fois que nous aurons confirmé que chacun des capteurs fonctionne individuellement, il faut vérifier qu'ils soient capables de fonctionner les uns avec les autres sans provoquer d'interférence. Pour ce faire, nous utilisons un concentrateur USB tel que présenté à la figure 3.4. Nous vérifierons les signaux reçus par l'ordinateur, chacun des capteurs doit être capable de transmettre sans trop de délais. Nous pourrons mesurer, avec des logiciels, le temps entre la requête et la réponse.

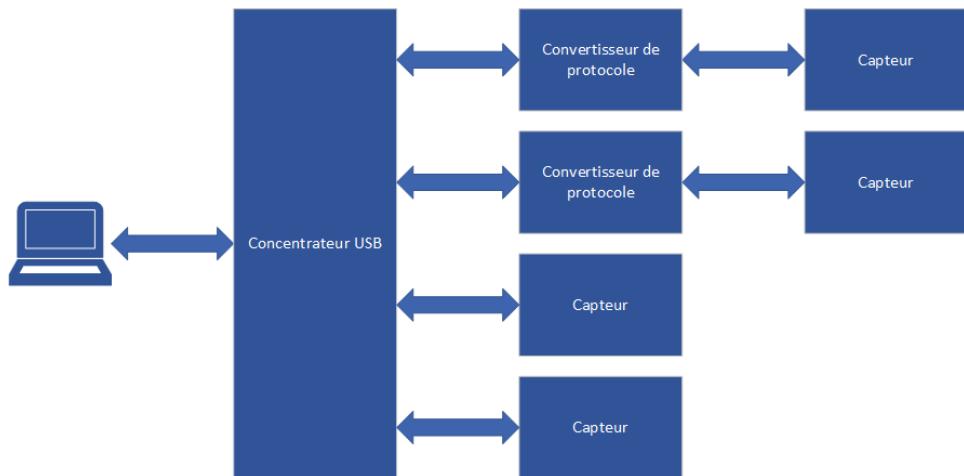


Figure 3.4 Exemple d'un essai avec un concentrateur

3.3.4 L'essaie final

Afin de vérifier le fonctionnement de notre prototype, nous avons établi la marche à suivre suivante. Lors de la réception des cartes, nous allons faire les inspections appropriées. Par exemple, nous pouvons vérifier que les surfaces sont bonnes, qu'il n'y a pas de défauts apparents dans la carte.

Nous allons ensuite souder les pièces qui sont liées à l'alimentation. Si la tension est présente sur les différents pins d'alimentations des puces et absentes sur les autres pins, nous passerons à l'étape suivante de test.

Pour cette étape, nous souderons les pièces qui sont en lien avec la puce de concentrateur et qui sont en lien avec la sortie de la caméra thermique et de la caméra numérique. Nous connecterons un ordinateur sur notre prototype, les caméras communiquent par USB, si nous recevons des images, cela signifie que la partie concentrateur USB de notre prototype fonctionne.

Nous souderons enfin les différentes pièces qui sont en lien avec les convertisseurs. Nous connecterons les différents capteurs sur le prototype et nous devrions être capables de recevoir les informations de tous les capteurs.

Pour finir, nous remplacerons le PC par Jetson Xavier de Nvidia équipé de ROS. S'il est capable de reconnaître et d'obtenir les différentes informations des capteurs, le test sera un succès.

CHAPITRE 4

PROCESSUS DE CONCEPTION

4.1 Élaboration de solution

Dans cette section, nous allons approfondir les différentes sections de la conception. Nous analyserons les choix que nous avons faits tout au long de la conception. Nous commencerons par les différentes sélections de capteurs. Nous poursuivrons avec l'architecture et les choix technologiques. Enfin, nous ferons l'analyse de l'intégration logicielle.

4.1.1 Sélection de la caméra

Afin de faire un choix de caméra numérique, nous avons défini les critères importants. L'opérateur du robot doit, à l'aide d'un poste de commande, être capable de lire un code QR. Afin de pouvoir lire ce code QR, la caméra doit posséder une bonne définition . Le code QR est assez petit, le robot pourrait s'approcher afin de le lire, mais nous préférons éviter ce cas de figure. La vitesse d'exécution des tâches est primordial, il est donc essentiel d'optimiser au maximum les requis.

Ainsi, plus la résolution est élevée, plus la note qu'on lui attribue dans la matrice de décision sera élevée. Puisque le robot est en mouvement lors de certaines acquisitions, nous allons privilégier les caméras qui ont des obturateurs globaux plutôt que les obturateurs roulants. Cela permet d'éviter la distorsion de l'image lors des déplacements. Il nous faut également une vitesse d'acquisition élevée. De plus, il faut être en mesure de reconnaître différents panneaux signalétiques avec différentes couleurs, c'est donc un critère éliminatoire pour les caméras.

Les trois choix que nous avons décidé d'évaluer sont la Point Grey CMLN-13S2C, le module ArduCAM et le module ON Semiconductor AR0135AT. On peut retrouver la matrice de décision utilisée dans ce processus, en annexe, à la page 37. Après délibération, c'est la caméra numérique de Point Grey qui sera utilisée. Bien que la caméra semble avoir gagné par quelques points, nous avons considéré d'autres points très importants. Notamment, le fait que la caméra choisie était déjà présente au club étudiant. Cela était particulièrement important puisqu'une caméra à coût nulle est un avantage pour notre projet et pour les finances de notre client. En plus, nous avons considéré l'aspect écologique de pouvoir réutiliser ce genre d'appareil électronique. Nous évitons que cette caméra se retrouve dans les déchets alors qu'elle est parfaitement capable de répondre aux besoins.

4.1.2 Sélection du microphone

Pour le choix du microphone, il faut qu'il ait un gain variable. Les conditions de test et les conditions réelles sont différentes, il faut alors avoir la possibilité d'ajuster le gain du microphone en fonction du bruit ambiant. Le type de microphone ne sera pas pris en compte lors de nos choix. En effet, nous ne sommes pas limités pour les alimentations externes, nous pouvons donc choisir des microphones à condensateur et la sensibilité des microphones dynamiques est suffisante pour notre utilisation. Lorsque nous avons fait la planification du temps, nous n'avons pas pris en compte que nous aurions à concevoir un CAN. Ainsi, nous priorisons les microphones qui sont équipés d'un.

Parmi les choix que nous avons identifiés, le premier microphone est le Digilent Pmod MIC3. Enfin, nous avons identifié deux de la compagnie MikroElektronika, soit le MIKROE-3445 et le MIKROE-2563. Nous avons utilisé, comme pour la caméra numérique, une matrice de décision. Cette matrice peut être trouvée à la page 37. Après analyse, le choix c'est arrêter sur le microphone de Digilent.

Ce microphone est équipé d'une interface SPI. Il possède un CAN de 12 bits et une réponse en fréquence linéaire entre 100 Hz et 15 kHz.

4.1.3 Choix de canal

Nous avons évalué les différentes solutions pour un choix de canal à utiliser pour acheminer les données. Nous avons besoin d'un protocole qui permet d'avoir plusieurs périphériques. Intuitivement, nous avons tenté d'évaluer l'Ethernet (1000BASE-T) et l'USB (2.0). Il s'agit de deux techniques robustes pour transmettre de l'information sur des distances suffisantes pour le projet. La distance maximale de l'USB¹ est 5m et 100m pour l'Ethernet². Les deux sont différentiels, ils sont alors résistants aux événements électromagnétiques. L'Ethernet a une meilleure vitesse de transfert, 1000 Mbit/s pour le 1000BASE-T, comparativement au 480 Mbit/s (60 MB/s) pour l'USB 2.0.

Le client nous a mentionné avoir une préférence pour un protocole qui possède deux lignes. Il souhaite être en mesure d'utiliser les deux lignes disponibles dans le bras du robot pour y acheminer les informations. Pour cette raison, et le fait qu'il sera plus simple de diagnostiquer les problèmes avec l'USB qu'avec l'Ethernet en fonction de nos connaissances, nous avons choisi l'USB 2.0.

1. Robert (2017) P.5

2. Wikipedia (2020)

4.1.4 Architecture

Pour les architectures de communication, nous avons envisagé deux possibilités. La première étant que notre carte d'acquisition traite les données de façon autonome. Cela impliquerait de mettre une intelligence sur la carte. Tous les capteurs communiqueraient avec ce cerveau en fonction des requêtes. Également, cette solution utilise moins la bande passante.

L'autre option est de faire une intelligence décentralisée. Les requêtes d'information seraient gérées par l'ordinateur de bord. Il faut alors une intelligence des deux côtés, donc chaque capteur doit être intelligent.

Nous avons représenté l'architecture que nous allons utiliser à la figure 4.1.

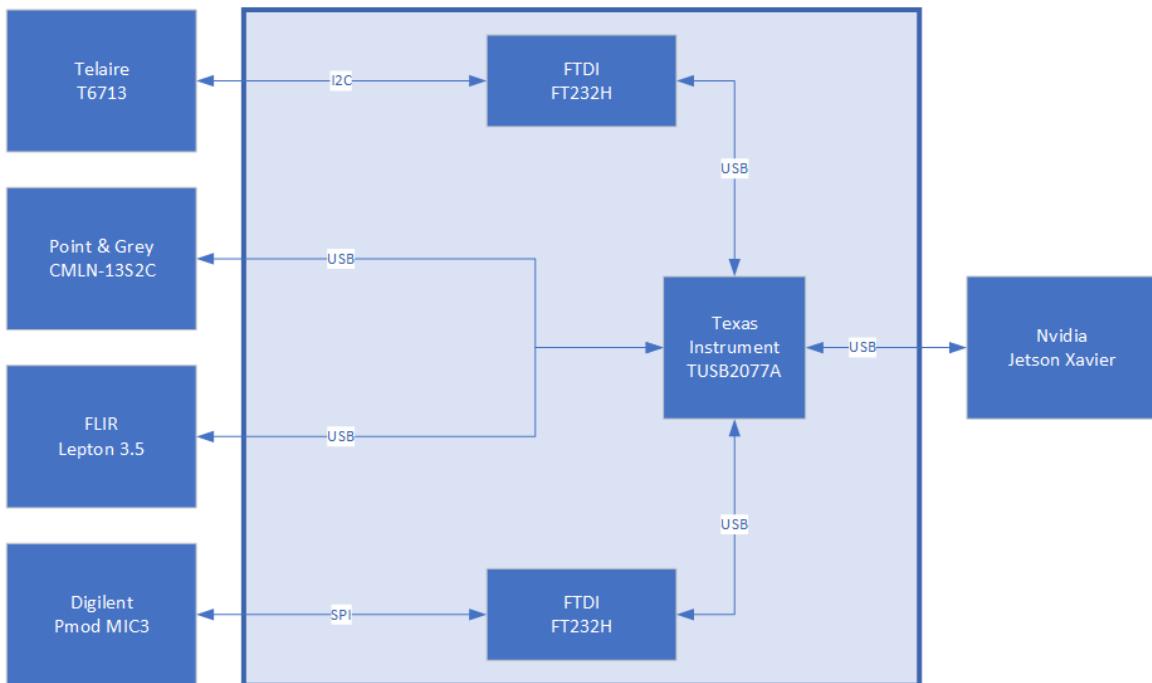


Figure 4.1 Architecture proposée

Les capteurs se brancheront directement sur les borniers de la carte, il fournira un signal communiquant en USB à la sortie. Chaque capteur sera connecté aux besoins à différents convertisseurs USB. Ces derniers seront reliés ensemble à un circuit qui agira comme un concentrateur afin de fournir un seul signal de sortie.

4.1.5 Convertisseur

Une fois l'architecture de communication choisie, nous avons choisi nos convertisseurs, ces derniers doivent transformer un protocole de communication en protocole USB. Nous avons choisi d'utiliser le FT232H puisqu'il couvre tous les protocoles requis. Pour fonctionner, il suffit d'installer un pilote fourni par le manufacturier. Ensuite, plusieurs techniques existent pour créer une interface avec le code. Puisque nous avons défini que la communication doit être en USB 2.0, nous avons choisis les convertisseurs pour qu'ils puissent fonctionner à cette vitesse.

Dans un premier temps, nous avons utilisé une carte d'évaluation pour faire nos tests, nous isolons ainsi les erreurs possibles. Avec cette méthode, nous nous assurons de ne pas avoir d'erreur dans le circuit lorsque nous développons la partie logicielle.

Lorsque tous les signaux sont convertis, nous obtenons cinq canaux de communications USB à transférer sur un seul canal. Nous avons alors décidé d'utiliser un concept de concentrateur USB. La puce que nous avons sélectionnée pour faire cela est la Texas Instrument TUSB2077A. Elle possède sept entrées, elle est donc suffisante pour notre projet.

4.1.6 Circuit imprimé

Nous voulons nous assurer que le produit final soit le plus compact et efficace possible. Ainsi, nous produirons un circuit imprimé qui implémentera les différentes technologies que nous avons choisies.

En considérant les signaux et leur référence, nous avons établi que le PCB devait être six couches. La figure 4.2 représente l'ordre dans lequel les couches apparaîtront.



Figure 4.2 Planification des couche pour le PCB

En suivant les recommandations des différents manufacturiers, nous avons défini des zones dans lesquels chacun des circuits sera instauré. Nous avons représenté ces zones dans la figure 4.3.

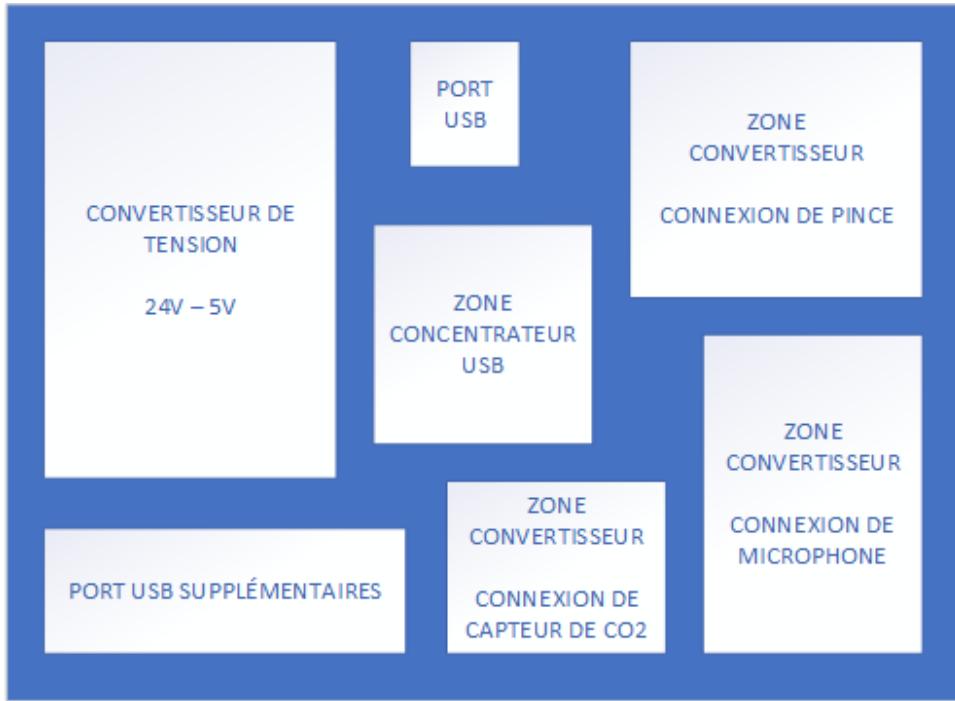


Figure 4.3 Planification des zones pour le PCB

4.1.7 Intégration logicielle

L'intégration logicielle d'un capteur comporte plusieurs étapes. La première est de lire les fiches techniques des capteurs. Dans celles-ci nous retrouverons les différents protocoles que les capteurs peuvent supporter ainsi que les différentes commandes que l'on doit exécuter. Selon nos recherches, le capteur de CO₂ utilisera le I²C , le capteur de son utilisera le SPI et la pince fonctionnera en RS485.

Chacun des protocoles nous permet d'avoir des données que l'on doit pouvoir traiter à l'aide d'un programme fait en Python ou en C++. Cela dépend de la disponibilité des bibliothèques natives du capteur. Ensuite, il faut que notre ordinateur soit capable de détecter le convertisseur.

Pour chacun des programmes, il faudra être en mesure d'ajuster les scripts afin qu'il soit en mesure d'utiliser le convertisseur. Finalement, la dernière partie est d'être capable d'analyser et représenter les données reçues.

4.2 Analyse de risque

Les risques sont une partie indésirable dans un projet. Ils peuvent survenir à tout moment et peuvent affecter différents aspects du projet. Un risque se caractérise par son impact et par sa probabilité. C'est-à-dire, à quel point il est capable de mettre en péril le projet et quels sont les chances que cela se réalise. C'est donc avec ces deux facteurs que nous prévoirons nos plans de contingence.

4.2.1 Canal

Le tableau 4.1 présente les risques que nous avons identifiés pour le choix du canal.

Risques	Entre 0 et 5	Entre 0 et 5
Risques	Probabilité	Impact
Que la vitesse ne soit pas suffisante.	2	5
Qu'il y ait des collisions dans les données	1	1
Que le canal soit beaucoup plus cher	2	3

Tableau 4.1 Analyse de risques du canal

La figure 4.4 place dans un espace cartésien, les risques élaborés dans le tableau 4.1.

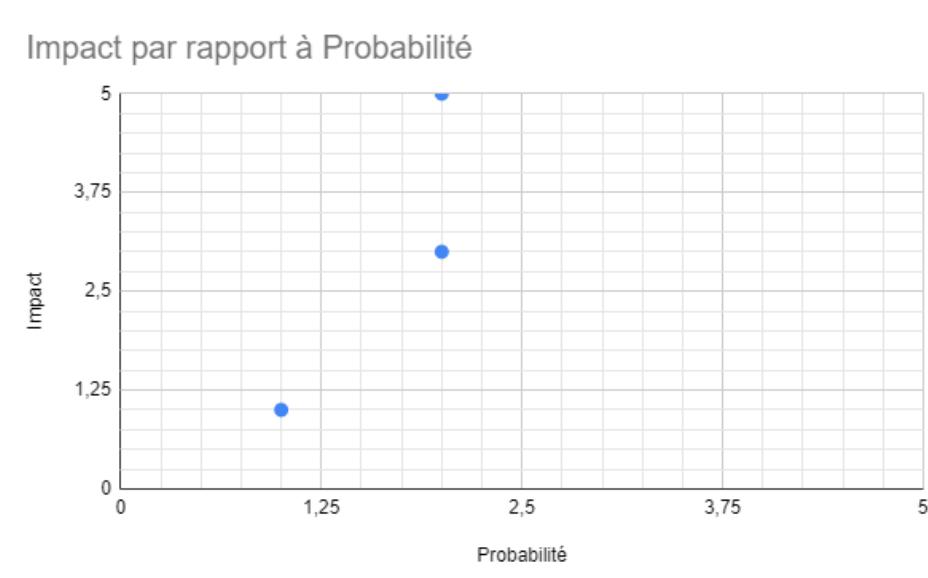


Figure 4.4 Graphique de l'impact du choix du canal par rapport à la probabilité

Nous avons donc trouvé trois risques à considérer.

Le premier risque est associé à la vitesse. Cela fait partie du mandat de notre client, nous devons donc éviter que le canal soit incapable de transmettre toutes les informations à une vitesse raisonnable. Le cas échéant, le projet serait un échec. Cependant, nous prenons toutes les précautions nécessaires quant aux vitesses.

Le deuxième risque est la collision dans les données, l'impact et la probabilité sont assez faibles dans ce cas là. Il existe plusieurs solutions afin de réduire les collisions des données, l'impact est donc assez faible.

Le dernier risque est d'ordre financier, si le prix est trop élevé nous aurons du mal à le mettre en place, néanmoins cela ne freinera pas le développement du projet, d'où l'impact moyen. Nous avons un contrôle sur le choix du canal, c'est pour cette raison que la probabilité est assez faible dans ces trois cas. Nous pouvons supprimer les risques à la source.

4.2.2 Convertisseur

Le tableau 4.2 présente les risques que nous avons identifiés pour le choix du convertisseur.

Risques	<i>Entre 0 et 5</i>	<i>Entre 0 et 5</i>
Probabilité	Impact	
Qu'il y ait des problèmes de compatibilité Linux	1	5
Qu'il y ait des problèmes d'adressages	2	3
Que le convertisseur ne supporte pas les protocoles dont nous avons besoins	0,5	4
Que le convertisseur ne fonctionne pas sur le concentrateur USB.	2	5

Tableau 4.2 Analyse de risques pour les convertisseurs

La figure 4.5 place dans un espace cartésien, les risques élaborés dans le tableau 4.2.

Nous avons donc trouvé quatre risques à considérer.

Lors du choix du convertisseur, la compatibilité entre le convertisseur et le système d'exploitation Linux aura un grand impact sur notre projet. En effet, le robot utilise Linux, le convertisseur n'est pas compatible, il ne pourra pas être utilisé pour notre projet et le mettra en péril.

Il est primordial que le convertisseur supporte les protocoles de communications des capteurs. Notre projet repose en partie sur la traduction des protocoles de communications en USB. Ce risque impacte grandement notre projet.

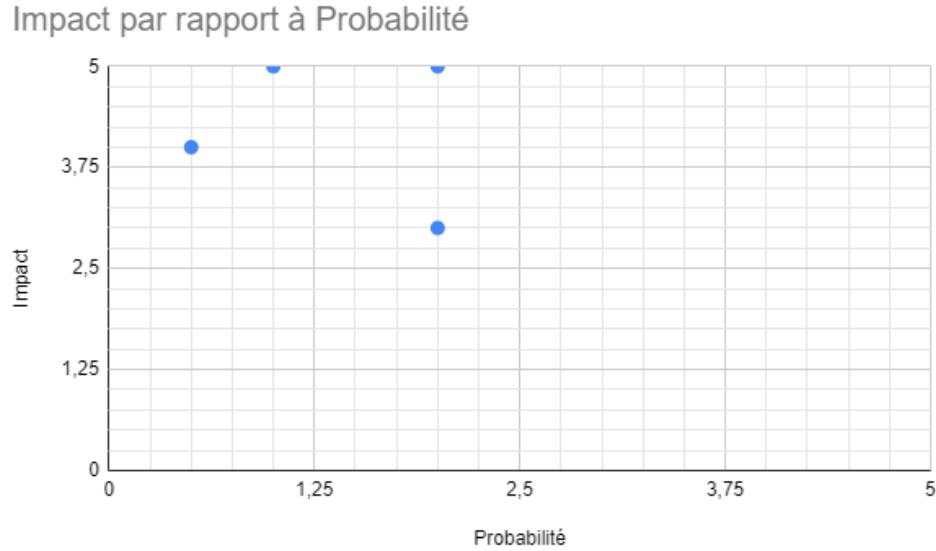


Figure 4.5 Graphique de l'impact du choix du convertisseur par rapport à la probabilité

Nous avons le contrôle sur le choix du convertisseur, la probabilité de ces risques est donc minime.

Le convertisseur doit impérativement être compatible avec un concentrateur USB, si ce n'est pas le cas, il nous sera impossible de transférer tous les signaux sur les deux lignes de communication. Ce risque est immense pour notre projet puisque nous devrions revoir l'architecture.

4.2.3 Le circuit imprimé

Le tableau 4.3 présente les risques que nous avons identifiés pour la conception du circuit imprimé.

La figure 4.6 place dans un espace cartésien, les risques élaborés dans le tableau 4.3.

Lors de la conception du PCB, nous avons plusieurs risques à prendre en considération. Une mauvaise conception de la schématique aura un impact conséquent sur le PCB, il pourrait mettre en péril le fonctionnement global. Bien que nous ayons accès aux différents documents des manufacturiers, nous ne sommes pas à l'abri d'une erreur de conception.

Nous faisons circuler plusieurs signaux à différentes vitesses dans ce PCB, ces signaux risquent de provoquer des interférences entre eux ou de subir des interférences extérieures. La probabilité de cet évènement est assez faible, car la longueur de nos traces est assez

Risques	Entre 0 et 5 Probabilité	Entre 0 et 5 Impact
Erreurs de conception des schématiques	2,5	4
Interférences des signaux	2	3
Mauvais placement des composants	4	2
Mauvaise gestion de l'impédance	2	3
Que la gestion des événements électrostatiques soit défaillante	1	3
Que le PCB n'arrive pas à temps	2	1
Que le PCB coûte trop cher à produire	3	3

Tableau 4.3 Analyse de risques pour le PCB

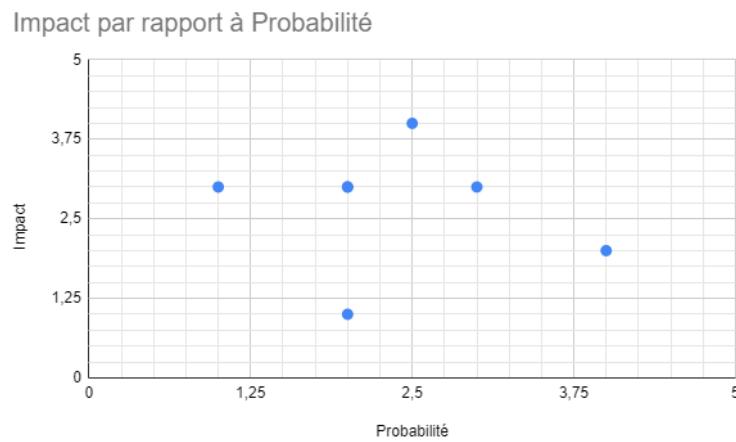


Figure 4.6 Graphique de l'impact de la conception du PCB par rapport à la probabilité

faible et nous avons à notre disposition les recommandations des manufacturiers. L'impact est également moyen puisque le signal les données seront traitées plus tard par l'ordinateur et à moins que ces derniers ne soient complètement dégradés, l'ordinateur sera capable de l'interpréter.

Certains signaux requièrent de contrôler l'impédance et la longueur des traces. Certaines marges sont données en fonction des standards. Il faut rester dans les marges imparties. Par exemple, l'USB requiert une impédance de $90 \Omega \pm 15\%$. Afin de mitiger le risque, le logiciel de conception offre un calcul dynamique de l'impédance.

Un mauvais placement des composants peut entraîner une dégradation des signaux. En plus, il pourrait demander de recommencer le PCB puisque ce dernier ne serait pas viable. Le risque que de telles conditions se produisent est moyen. Bien qu'il s'agisse d'un PCB

à densité moyenne et que nous souhaitons obtenir le plus petit empreinte possible. Il faut alors faire attention aux différents plans de tension. En effet, certains signaux ne doivent jamais passer par un plan de tension afin de ne pas perdre leur référence. Si l'on venait à faire une telle erreur, l'intégrité du signal serait compromise.

Ne pas prendre les bonnes précautions quant aux évènements électrostatiques peut mener à un PCB non fonctionnel. En plus, cela pourrait impacter l'utilisateur si les méthodes de mise à la terre sont non adéquates. La probabilité est très faible puisque la plupart des composants choisis ont déjà une bonne résistance à de tels évènements. Les seuls composants qui vont demander une attention particulière sont ceux avec un boîtier métallique (Ex. : un port USB). L'impact est un peu plus que la moyenne puisqu'avec les bonnes précautions (utilisation d'un tapis antistatique, bracelet ESD, etc.), nous pourrons tout de même tester le produit.

Un autre des risques liés au PCB est sa livraison. Nous commandons le PCB par le manufacturier Elecrow ou un autre manufacturier asiatique. Nous avons déjà eu affaire avec eux dans de précédents projets, la livraison est rapide et fiable. C'est pour cela que la probabilité du retard est assez faible. Nous avons également tous les éléments pour faire un banc de test, l'impact de ce risque est donc assez faible.

Le dernier risque du PCB est son coût de production, il a un impact moyennement élevé sur le projet. En effet, Capra étant un club étudiant, si le prix est trop élevé le club aura du mal à débloquer les fonds pour ce dernier. La probabilité d'avoir un PCB dispendieux est également élevée, puisque nous développons un PCB qui gère la communication. Ce genre de PCB coute d'habitude assez cher à produire.

4.2.4 Logiciel

Le tableau 4.4 présente les risques que nous avons identifiés pour la conception logicielle.

Risques	Entre 0 et 5	Entre 0 et 5
Risques	Probabilité	Impact
Que les scripts fonctionnent mais ne soient pas compatible avec ROS	4	4
Que les scripts ne fonctionnent pas.	2	5
Que les données interprétées ne soient pas les données réelles	2,5	5
Que l'acquisition soit trop lente.	1	2

Tableau 4.4 Analyse de risques logiciel

La figure 4.7 place dans un espace cartésien, les risques élaborés dans le tableau 4.4.

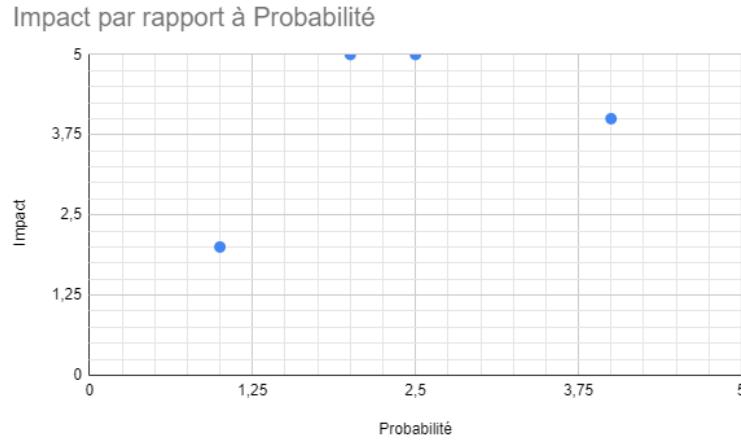


Figure 4.7 Graphique de l'impact de la conception logiciel par rapport à la probabilité

La communication interne du robot fonctionne sur un système basé sur ROS. Le but du projet est de connecter les données reçues par nos capteurs à l'environnement du robot. L'impact et la probabilité de ce risque sont élevés, en effet, nous devons bâtir nos scripts en ayant en tête de l'intégrer plus tard à l'environnement. La compatibilité n'impacte pas le fonctionnement du script. En effet, ce dernier peut fonctionner, mais être incompatible. Ce qui rend le risque assez difficile à détecter au cours du développement.

Nous avons également un risque que le script ne fonctionne pas, si c'est le cas la partie logiciel du projet sera non fonctionnelle. Néanmoins, un membre de notre équipe est spécialisé en programmation, ce qui réduit dramatiquement la probabilité.

Une vitesse d'acquisition rapide de données est importante pour gagner du temps dans les épreuves. Néanmoins si notre vitesse d'acquisition est plus lente, cela n'impactera pas notre projet, puisque nous n'atteindrons jamais une vitesse suffisamment basse pour nuire au robot. La probabilité que cela puisse arriver est également faible, nous faisons de la conversion de protocole, il n'y a aucune opération qui puisse suffisamment ralentir nos données.

Une bonne précision des données recueillie est également importante pour notre client, ces données seront traitées par l'ordinateur central, le robot agira selon ces derniers. Nous devons nous assurer que ces données soient des données réelles, puisqu'elles influenceront le comportement du robot, l'impact de ce risque est alors très élevé. Les capteurs sont utilisés lors de la conception du script, il est donc peu probable que la personne responsable ne se rende pas compte lors que les données sont erronées.

4.3 Outils et techniques

Le premier outil d'ingénierie que nous avons utilisé est le cahier de charge. Il nous a permis de bien définir ce qu'il faut accomplir dans ce mandat. Comme outil de planification, nous avons mis en place un graphique Gantt. Il permet de définir visuellement les tâches à accomplir ainsi que leur responsable. Nous avons également mis en place une zone de discussion. Il a été décidé avec consensus que nous utiliserons l'application *Discord*. Ensuite, nous avons utilisé la matrice de décision comme outil décisionnel. Enfin, nous avons utilisé les matrices de risque. Celles-ci permettent de mettre sur un plan cartésien les risques et leur importance dans le projet.

Pour vérifier que nos convertisseurs USB vers nos divers protocoles fonctionnent avec nos capteurs, nous allons utiliser l'analyseur logique pour vérifier les signaux. De plus, nous vérifierons que le PC reçoit les données. Nous utiliserons le générateur de fonction pour générer des signaux PWM. Pour tester nos capteurs, nous devons être familier avec l'environnement Linux ainsi que ROS. Puisque l'ordinateur central fonctionne dans cet environnement, nous voulons le tester dans des conditions réelles. Nous avons intégré des bibliothèques populaires telles qu'OpenCV. Afin de vérifier le fonctionnement de la caméra, nous avons utilisé des logiciels de vidéo tels que VLC multimédia.

Pour concevoir notre circuit, nous allons utiliser nos acquis sur l'électronique. Nous avons également dû nous renseigner sur les différents protocoles de communication afin de savoir quels composants rajouter à notre circuit afin de préserver la qualité du signal. En cours de conception, la connaissance et l'utilisation des outils de déverminage, tels que le multimètre et l'analyseur logique seront de mise.

Afin de produire les schémas électriques et le circuit imprimé, nous utiliserons *Altium Designer 19*. Puisque nous transférerons de l'information sur un circuit imprimé, nous tenterons de réduire le bruit. Ainsi, avoir des connaissances quant aux bonnes pratiques de routage pour la communication est un atout.

4.4 Intégration des concepts retenus

Nos capteurs utilisent trois types de protocoles de communication, le capteur de CO₂ utilise le RS485, le microphone utilise le SPI. La caméra numérique et la caméra thermique utilisent le protocole USB.

Afin que les données puissent circuler dans le même canal de communication, nous devons transformer les communications SPI et RS485 en protocole USB. Nous avons utilisé une puce fabriquée par FTDI, la *FT232H*. Elle permet de faire l'interface entre différents protocoles et l'USB. Le tout sera connecté sur une puce Texas Instrument : *TUSB2077A*.

Ainsi, nous avons la figure 4.8 qui permet d'observer la conversion des différents signaux vers le canal unique de l'USB.

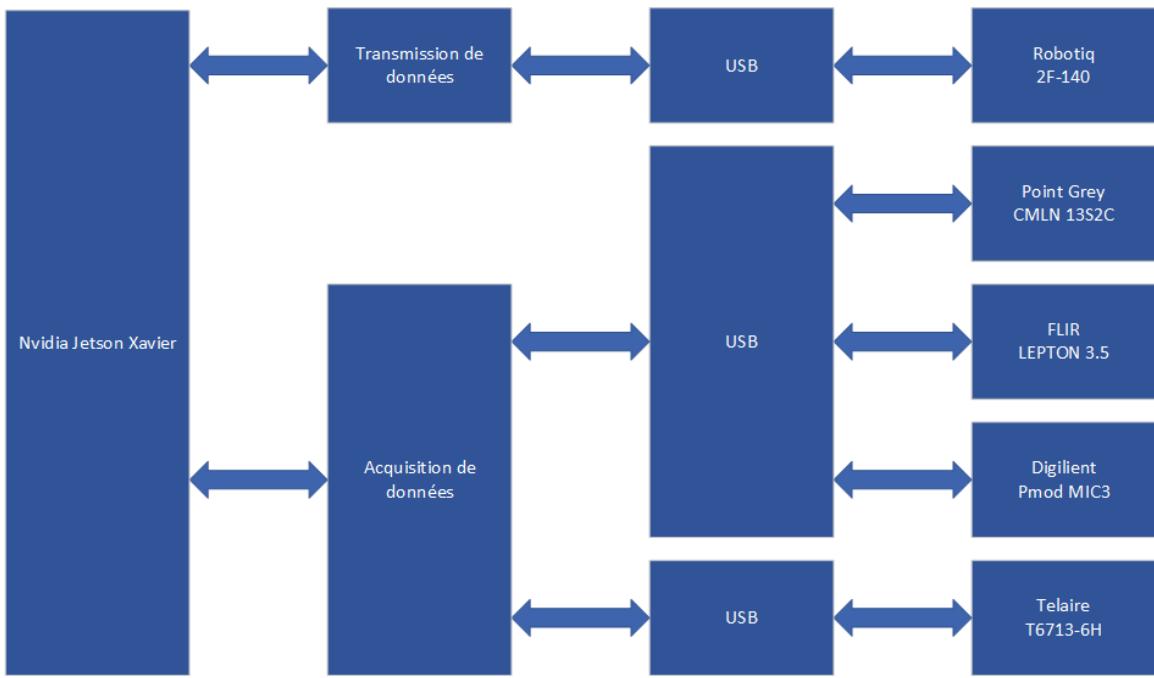


Figure 4.8 Capteurs avec les interfaces proposés

4.5 Réalisation de tâches spécifiques

Pour pouvoir configurer nos capteurs, nous avons utilisé des programmes disponibles sur GitHub que nous avons modifié pour nos besoins. De plus, nous avons utilisé des bibliothèques afin de faciliter la programmation. Nous nous sommes également inspirés des montages recommandés lorsque nous avons effectué le test sur le capteur de CO₂. Certains plans électriques ont été inspirés des applications typiques données par les manufacturiers respectifs des puces. Néanmoins, des modifications nécessaires y ont été faites pour adapter leur solution à notre problématique.

CHAPITRE 5

RÉSULTATS ET DISCUSSIONS

5.1 Résultats

Dans cette section, nous discuterons des différents résultats obtenus avec les essais. Enfin, nous discuterons des résultats qui ont été obtenus et ceux qui sont déficients.

5.1.1 Capteur de dioxyde de carbone

Le club nous a fourni ce capteur, il s'agit du capteur utilisé lors de la précédente compétition. Afin de faire une intégration préliminaire, nous avons utilisé un ensemble de développement pour la puce FT232H créé par la compagnie Adafruit. Nous avons pu le configurer afin que ce dernier communique en I²C. Afin d'y parvenir, nous avons d'abord lu les documents fournis par le fabricant. Ceux-ci inclut les spécifications de la puce, mais aussi les utilisations recommandées. Nous avons également importé diverses librairies modifiées, des codes trouvés sur divers sites, pour que le convertisseur puisse communiquer et transmettre de l'information. Puisque nous n'avons pas accès à des sources contrôlées de CO₂, il nous est impossible de créer un banc d'essai physique pour le capteur. La méthode privilégiée alors était de faire la lecture de la pièce, et comparer avec les valeurs moyennes de CO₂ dans une pièce. Ensuite, nous avons soufflé sur le capteur afin de provoquer une augmentation de la concentration de CO₂ lue.

La courbe de la figure 5.1 s'obtient à l'aide des valeurs de la table 5.1. On observe que le taux moyen de CO₂ dans la pièce est compris entre 800 ppm et 900 ppm. Ces valeurs sont plutôt normales considérant l'environnement de test. Généralement, le taux moyen de CO₂ dans une pièce se situe entre 400 ppm et 1000 ppm. Lorsque nous soufflons sur le capteur, nous remarquons un sommet d'environ 1 200 ppm.

Selon nos recherches, l'humain pour avoir une concentration allant jusqu'à 40 000 ppm de CO₂ lorsqu'il expire. Nous sommes conscient que cette valeur se dilue rapidement dans l'air. En plus, le capteur a un certains délais de traitement. Nous arrivons tout de même à lire des valeurs qui sont plus grande qu'un normale de pièce, ce qui peut très bien indiquer la présence d'un être humain. Nous pouvons dire que ces valeurs sont normales, le test du capteur de CO₂ est une réussite.

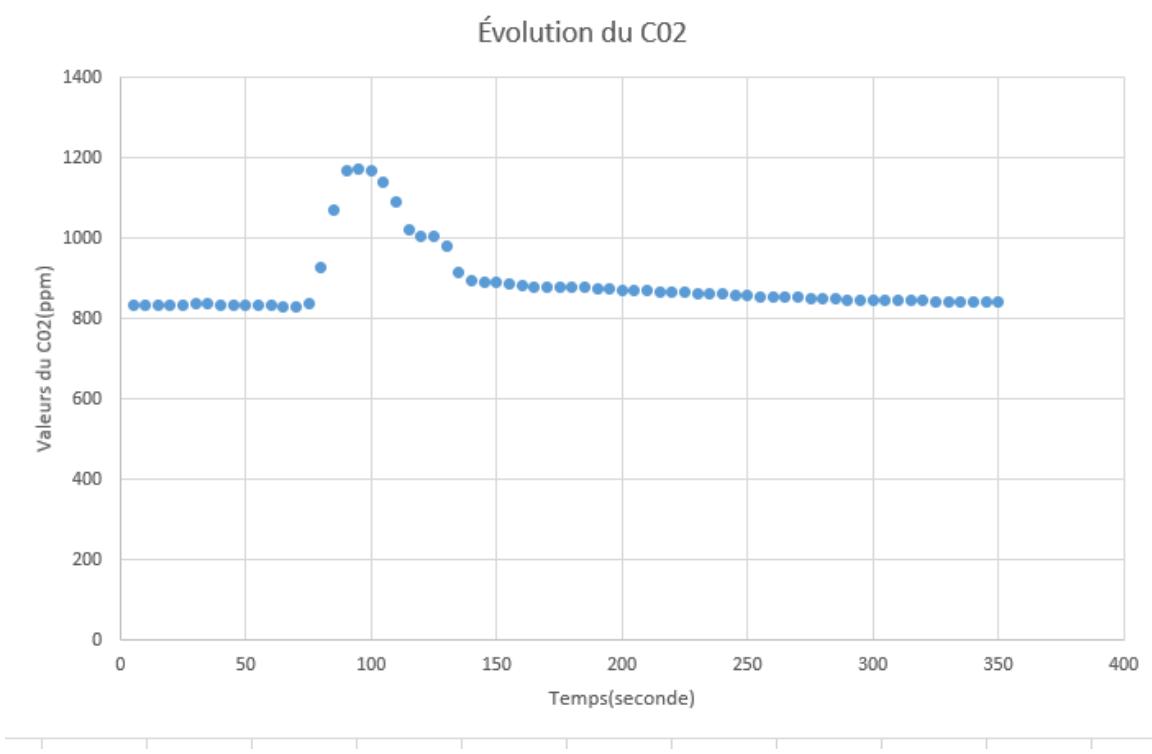


Figure 5.1 Courbes d'évolution du taux de CO₂ dans une pièce

Temps (s)	Concentration (ppm)	Temps (s)	Concentration (ppm)
0	836	175	879
5	835	180	878
10	836	185	877
15	836	190	875
20	836	195	873
25	837	200	871
30	837	205	869
35	836	210	868
40	836	215	866
45	835	220	865
50	834	225	864
55	833	230	862
60	832	235	861
65	832	240	859
70	838	245	857
75	927	250	856
80	1071	255	855
85	1171	260	855
90	1175	265	853
95	1170	270	852
100	1142	275	850
105	1091	280	849
110	1024	285	848
115	1005	290	848
120	1004	295	848
125	980	300	847
130	915	305	846
135	897	310	845
140	892	315	845
145	890	320	844
150	888	325	844
155	884	330	844
160	881	335	844
165	880	340	843
170	879	345	843

Tableau 5.1 Évolutions de la concentration de CO₂ dans la pièce

5.1.2 Les caméras

Pour la caméra numérique, afin de vérifier qu'elle fonctionne, nous avons utilisé plusieurs logiciels disponibles dans l'environnement Linux. Certains nous ont permis d'obtenir une vidéo adéquate. Nous avons donc conservé la résolution de base. La figure 5.2 présente un exemple d'une capture d'un code QR. Nous avons volontairement créé un environnement dont la luminosité est moindre que les conditions de compétition. Les résultats ont permis de faire la reconnaissance du code QR.

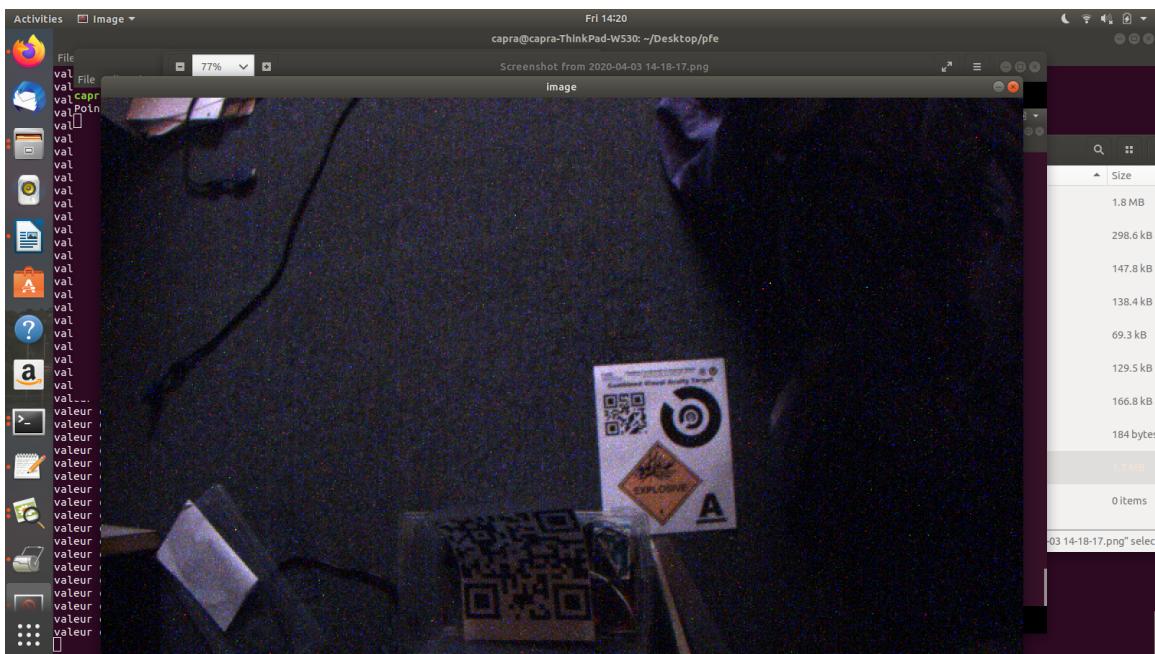


Figure 5.2 Capture d'un code QR

La caméra thermique est la même qui était utilisée par le client. Nous n'avons pas été en mesure de tester cette caméra puisque nous n'avons pas eu accès à la caméra lors de la phase de test. Cependant, la connexion étant USB, le fonctionnement ne devrait pas être différent. En considérant que le branchement USB est prévu directement sur le concentrateur USB, nous présumons que la caméra thermique sera en mesure de garder les mêmes performances.

5.1.3 Le microphone

L'intégration logicielle est incomplète. Un début de code sera fourni, c'est-à-dire que la connexion est établie. L'analyse et la confirmation de la réception des données reste à faire. Certaines données semblent être reçues sur l'ordinateur par le microphone et ne sont pas traitées.

Nous n'avons pas eu le temps de faire plus de tests. En effet, afin de faciliter le développement, avoir une interface qui aurait permis de jouer les données sonores captées aurait été optimal.

5.1.4 Les schémas électriques et la conception du circuit imprimé

Les schémas électriques ont pu être réalisés avec succès. En effet, toute l'architecture a été traduite en un circuit électrique. Nous avons les schémas de l'alimentation, des connexions USB, du concentrateur USB ainsi que les convertisseurs de protocoles. Bien que nous n'avons pas pu tester le microphone, nous avons pu faire la conception du circuit. Grâce à la documentation du FT232H et les diverses lectures sur les différents protocoles de communication, nous avons pu produire les schématiques des différents convertisseurs. Pour la conception des alimentations, nous avons utilisé un convertisseur 24 V_{DC} vers 5 V_{DC}. Cela nous a permis de nous assurer d'avoir une alimentation stable Quant au circuit qui gère les différents protocoles de communication et le circuit qui fait office de concentrateur USB, nous ne l'avons pas simulé. En effet, nous n'avons ni eu le temps ni la connaissance pour simuler un circuit qui gère les protocoles de communications. Néanmoins, nous nous sommes basés sur la documentation du fabricant et nous avons recoupé cette information à travers nos recherches. Nous n'avons pas pu faire les tests en connectant les différents capteurs à un concentrateur. En effet, nous n'avons pas pu tester tous les capteurs, tester le circuit avec un concentrateur aurait donc été inutile au vu de notre situation.

Les plans pour le PCB sont complétés, nous nous sommes basés sur la connaissance en design de PCB de l'un de nos membres, les projets qu'il a accomplis et son expérience de travail, nous sommes pleinement confiants du fonctionnement de notre plaque. Malheureusement, dû aux nombreux évènements indépendants de notre volonté, nous n'aurons pas eu de PCB. La pandémie de COVID-19 a ralenti la production de PCB, nous ne pourrons donc pas en obtenir.

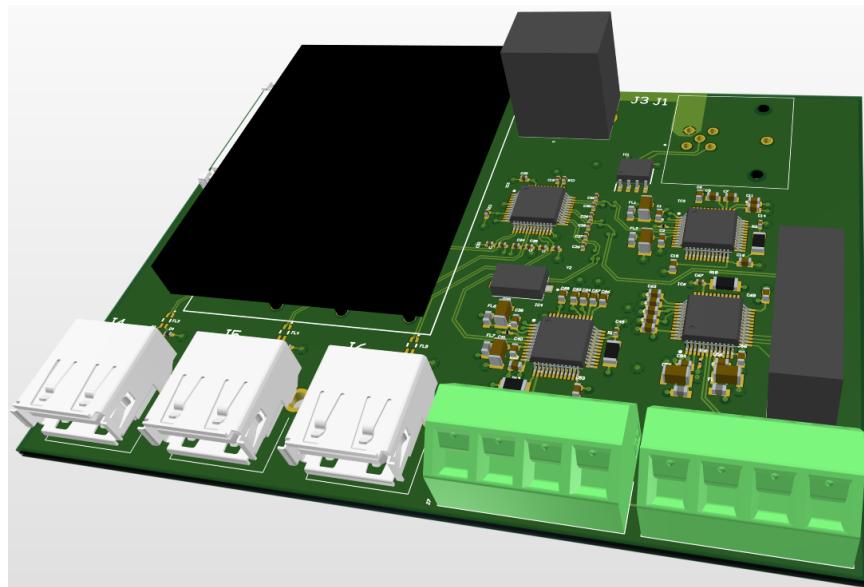


Figure 5.3 Vue du dessus en 3D du PCB



Figure 5.4 Vue du dessous en 3D du PCB

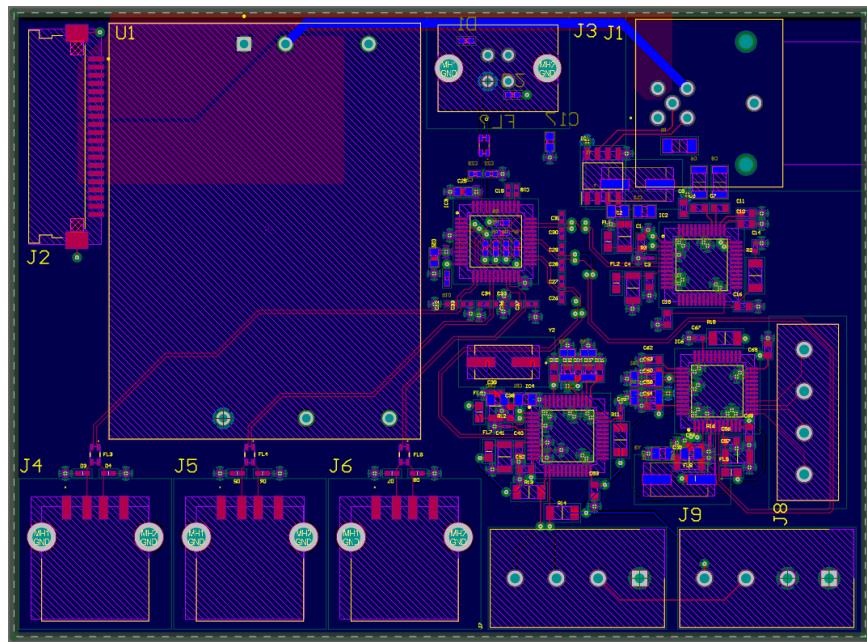


Figure 5.5 Vue de la conception du PCB

5.2 Discussions

Somme tout, nous sommes satisfaits de ce que nous avons réussi à accomplir. Le projet n'est pas complet selon les objectifs que nous nous étions donnés. Cependant, ce que nous sommes en mesure de rendre au client est un avancement par rapport à ce qui était présent. Nous sommes en mesure de mieux apprécier les différents aspects de la profession que nous nous apprêtons à commencer.

Lors de ce projet, nous avons été en mesure de placer les fondations du projet. Les différents capteurs ont été sélectionnés afin de correspondre aux besoins du client. Un circuit imprimé a été conçu, mais non manufacturer.

Nous avons dû amorcer la conception alors que les tests sur certains capteurs n'ont pas été faits. Il aurait été plus judicieux d'assigner plus de ressources sur les essais puisqu'ils ont pris beaucoup plus de temps que ce qui avait été prévu. En effet, une seule personne était responsable de la preuve de concept. En assignant une personne supplémentaire, nous aurions probablement pu finir les tests. Lorsque les essais auraient été terminés, un des membres aurait été dédié à la rédaction des documents techniques. Les deux autres membres de l'équipe auraient pu se concentrer sur la conception électronique des circuits. Enfin, deux membres auraient poursuivi la rédaction des documents techniques. Le troisième membre de l'équipe s'occuperait alors du routage et la conception du PCB. Cela est un travail qui se fait principalement de manière individuelle.

Ainsi, nous fournirons à notre client les différents schémas du projet (électrique, fonctionnel, etc.). Il trouvera la procédure pour valider le fonctionnement de la carte au sein de ce rapport. Nous lui fournirons les codes sources afin qu'il puisse configurer les FT232H. Nous lui fournirons également les nomenclatures afin qu'il puisse acheter les pièces et assembler la carte. Les programmes logiciels ne sont pas utiles pour le fonctionnement final de la carte, mais il s'agit d'un outil que nous avons développé lors du mandat.

Après analyse, nous aurions aimé ajouter un générateur de PWM directement sur la carte afin de réduire l'empreinte générale du projet. Dans la conception actuelle, les pulses doivent être créés avec générateur externe.

CONCLUSION

Au début du mandat, nous avions pour objectif d'aider le club Capra dans sa conception électrique. Nous devions fournir un système d'acquisition pour leurs différents capteurs situés sur le bras. Ces quatre capteurs ont chacune une utilité bien particulière dans la compétition. Un système de communication sans-fil avait déjà été mis en place. Nous devions la remplacer par une technologie filaire, rapide et facile à connecter.

Ainsi, notre mandat s'est amorcé avec la tâche de trouver les capteurs manquants. Ces derniers respectent les multiples requis imposés par les épreuves. Nous devions également trouver une architecture de communications afin de faire circuler les informations des capteurs à l'ordinateur. En choisissant l'USB, nous avons effectivement sélectionné un protocole différentiel. Ces derniers utilisent des protocoles différents, nous devons les convertir en un seul protocole afin de pouvoir faire circuler l'information dans les deux fils. Une fois cela fait, nous avons terminé les schémas électrique des circuits. Nous avons concevoir un PCB sur lequel tous les capteurs y seront branchés. Le PCB sera relié aux deux câbles et communiquera avec l'ordinateur central.

Bien qu'une carte a été conçue, nous ne pourrons pas la recevoir. Les différents manufacturiers que nous avons analysés avaient des retards d'un à quatre mois ou n'avaient pas assez de personnel pour fabriquer des PCB à plus de deux couches. Une procédure de test a été écrite, elle sera transmise à notre client avec les schémas, les fichiers de fabrication ainsi que les fichiers sources Altium.

Ce projet nous a permis de faire le développement d'un produit. Nous avons pu apprendre la gestion de projet à travers nos diverses réunions. Nous avons également appris à écrire de la documentation technique. Suite à ce projet, nous avons appris à discerner les forces et faiblesses de chacun. Dans le futur, nous saurons comment attribuer les tâches. Nous avons également appris nos différentes forces, faiblesses. Par-dessus tout, nous avons appris comment se surpasser. Chacun a appris des autres membres, que ce soit par leurs méthodes de travail, leur implication et leurs connaissances.

ANNEXE I
MATRICES DE DÉCISION

Modèle	Critères de sélection								Total	Rang		
	Résolution		Alimentation		Acquisition (FPS)		Sortie					
	Caract.	Note	Caract.	Note	Caract.	Note	Caract.	Note				
CMLN-13S2C	1296x964	$\frac{8}{10}$	5 V	$\frac{4.5}{5}$	18 FPS	$\frac{5}{10}$	USB 2.0	$\frac{8}{10}$	$\frac{25.5}{35}$	1		
ARDUCAM	480x640	$\frac{4}{10}$	1.8 V	$\frac{2}{5}$	100 FPS	$\frac{10}{10}$	USB 3.0	$\frac{8}{10}$	$\frac{24}{35}$	2		
AR0135AT	1.2M	$\frac{9}{10}$	1.8 V	$\frac{2}{5}$	60 FPS	$\frac{8}{10}$	12 bits brutes	$\frac{4}{10}$	$\frac{24}{35}$	3		

Tableau-A I-1 Matrice de décision pour la caméra

Modèle	Critères de sélection				Total	Rang		
	Contrôle du gain		Sortie					
	Caractéristique	Note	Caractéristique	Note				
Pmod MIC3	Analogique	$\frac{8}{10}$	SPI	$\frac{8}{10}$	$\frac{16}{20}$	1		
MIKROE-3445	Aucun	$\frac{4}{10}$	I ² C	$\frac{8}{10}$	$\frac{12}{20}$	2		
MIKROE-2563	Aucun	$\frac{4}{10}$	Analogique	$\frac{4}{10}$	$\frac{8}{20}$	3		

Tableau-A I-2 Matrice de décision pour le microphone

ANNEXE II

CODE CAPTEUR DE DIOXYDE DE CARBONE "CAP_CO2.PY"

```
import serial , time
from pyftdi import FtdiLogger
from pyftdi.ftdi import Ftdi
from pyftdi.i2c import I2cController
from pyftdi.usbtools import UsbTools

i2c = I2cController()

i2c.configure('ftdi://ftdi:232h:1/1')

slave = i2c.get_port(0x15) #slave address 15

#program
while 1:

    #writing {0x04, 0x13, 0x8b, 0x00, 0x01}
    #port.write(b'\x00\x01', relax=False, start=False)
    e = i2c.poll(0x15)
    e = i2c.write(0x15,[0x04,0x13,0x8b,0x00,0x01])

    time.sleep(5)

    #reading
    #port.read(4, start=False)
    s = i2c.read(0x15,7)
    print(s)

    ppm = s[2]*256 + s[3]
    print(ppm)

i2c.terminate()

print("done")
exit(1)
```

ANNEXE III

CODE CAPTEUR DE SON "CAP_MIC.PY"

```
import serial , time
from pyftdi import FtdiLogger
from pyftdi.ftdi import Ftdi
from pyftdi.spi import SpiController
from pyftdi.usbtools import UsbTools

spi = SpiController()

recu1 = 0
spi.configure('ftdi://ftdi:232h:1/1')

slave = spi.get_port(cs=0, freq=1E6, mode=0)
somme = 0

print (slave)

while 1:
    time.sleep(1)
    recu1 = slave.read(2,1,1);

    x = recu1[1]
    x |= (recu1[0] << 8)

    #missing to make value conversion

    print (x)

    spi.terminate()
    print("done")
    exit(1)
```

ANNEXE IV

CODE CAMÉRA NUMÉRIQUE "CAMNUM.C"

```
#inspired by Kevin Hughes code
#include "FlyCapture2.h"

#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>

#include <iostream>

using namespace FlyCapture2;

int main()
{
    Error error;
    Camera camera;
    CameraInfo camInfo;

    // Connect the camera
    error = camera.Connect( 0 );
    if ( error != PGRERROR_OK )
    {
        std::cout << "Failed_to_connect_to_camera"
                << std::endl;
        return false;
    }

    // Get the camera info and print it out
    error = camera.GetCameraInfo( &camInfo );
    if ( error != PGRERROR_OK )
    {
        std::cout << "Failed_to_get_camera_info_from_camera"
                << std::endl;
        return false;
    }

    std::cout << camInfo.vendorName << " "
            << camInfo.modelName << " "
            << camInfo.serialNumber
            << std::endl;
```

```

error = camera.StartCapture();
if ( error == PGRERROR_ISOCH_BANDWIDTH_EXCEEDED )
{
    std::cout << "Bandwidth_exceeded"
                << std::endl;
    return false;
}
else if ( error != PGRERROR_OK )
{
    std::cout << "Failed_to_start_image_capture"
                << std::endl;
    return false;
}

// capture loop
char key = 0;
while(key != 'q')
{
    // Get the image
    Image rawImage;
    Error error = camera.RetrieveBuffer( &rawImage );
    if ( error != PGRERROR_OK )
    {
        std::cout << "capture_error" << std::endl;
        continue;
    }

    // convert to rgb
    Image rgbImage;
    rawImage.Convert( FlyCapture2::PIXEL_FORMAT_BGR,
                      &rgbImage );

    // convert to OpenCV Mat
    unsigned int rowBytes = (double)rgbImage.
        GetReceivedDataSize() / (double)rgbImage.GetRows();

    cv::Mat image = cv::Mat(rgbImage.GetRows(),
                           rgbImage.GetCols(), CV_8UC3,
                           rgbImage.GetData(), rowBytes);

    cv::imshow("image", image);
    key = cv::waitKey(30);
}

```

```
error = camera.StopCapture();
if ( error != PGRERROR_OK )
{
    // This may fail when the camera was removed,
    // so don't show an error message
}

camera.Disconnect();

return 0;
}
```

ANNEXE V

CODE CAMÉRA THERMIQUE

Ce code a été récupéré sur la page GitHub du club Capra à l'adresse suivante : https://github.com/clubcapra/purethermal_ros

1. Fichier : purethermal.launch

```
<launch>
  <node name="purethermal" pkg="usb_cam" type="usb_cam_node" output="screen" >
    <param name="video_device" value="/dev/video2" />
    <param name="image_width" value="160" />
    <param name="image_height" value="120" />
    <param name="pixel_format" value="uyvy" />
    <param name="camera_frame_id" value="purethermal" />
    <param name="io_method" value="mmap" />
    <param name="framerate" value="9" />
  </node>
</launch>
```

2. Fichier : 99-purethermal.rules

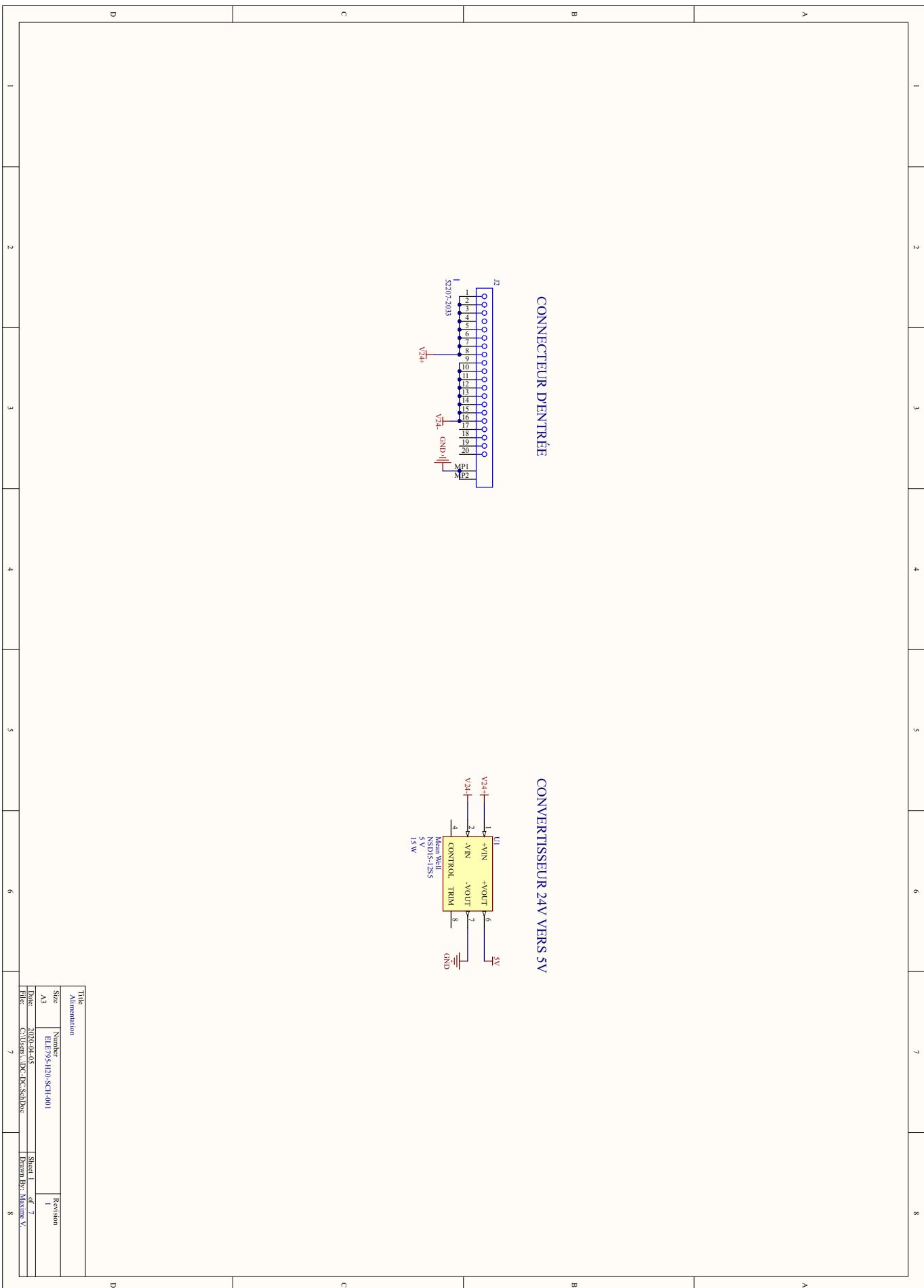
```
#UVC cameras
SUBSYSTEMS=="usb", ATTRS{idVendor}=="1e4e", ATTRS{idProduct}=="0100", MODE="0666"
```

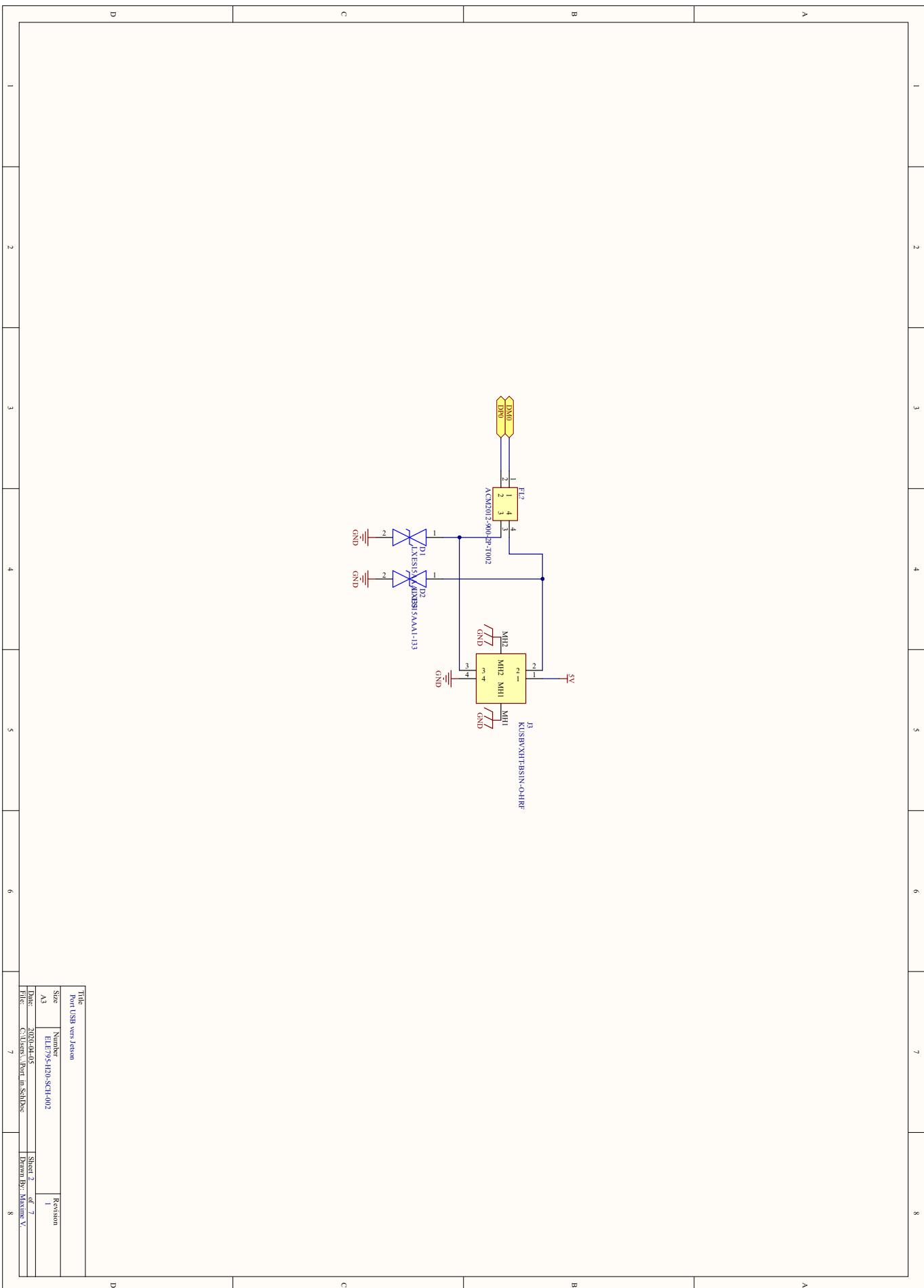
ANNEXE VI

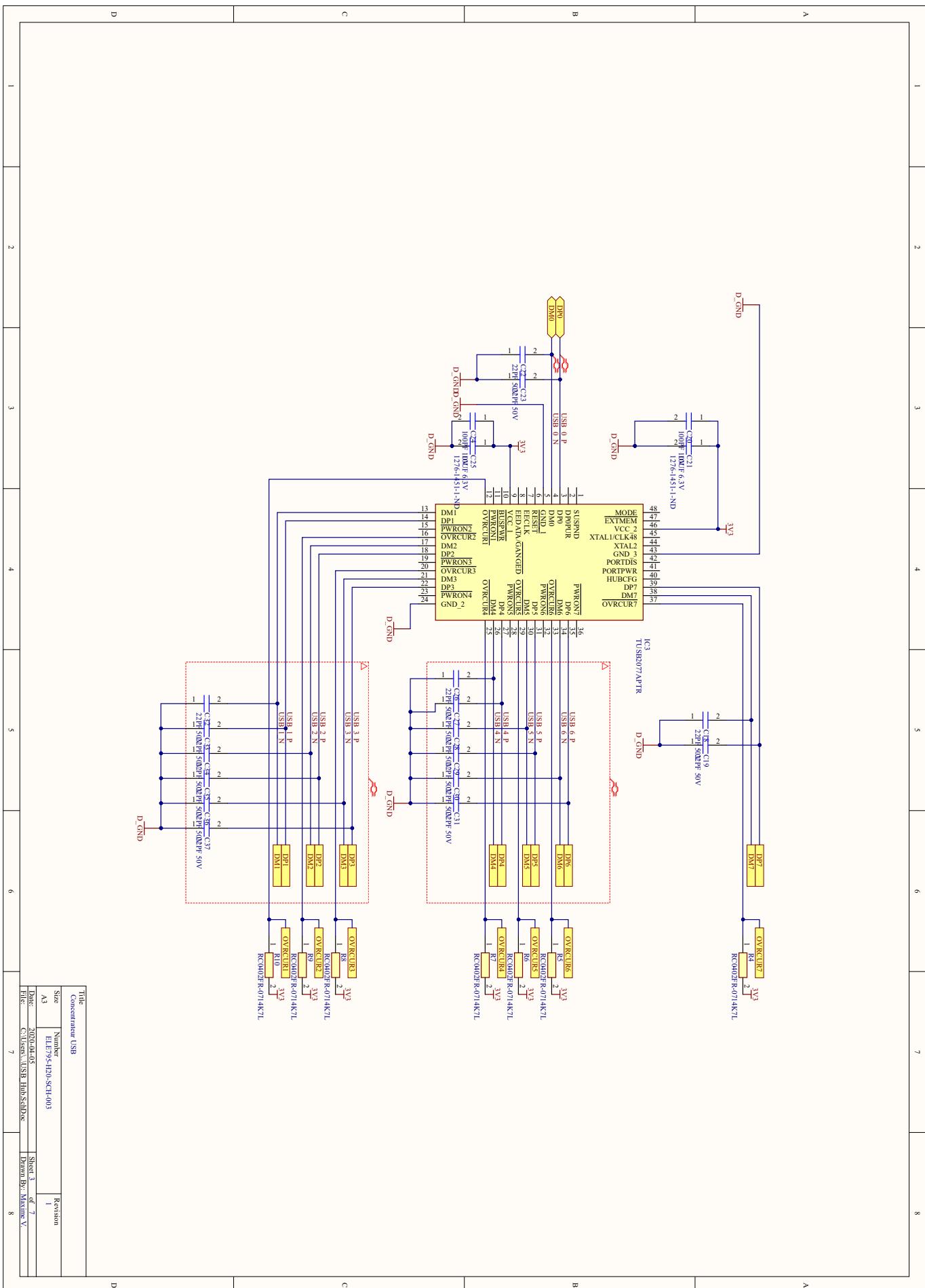
SCHÉMATIQUES ALTIUM DESIGNER

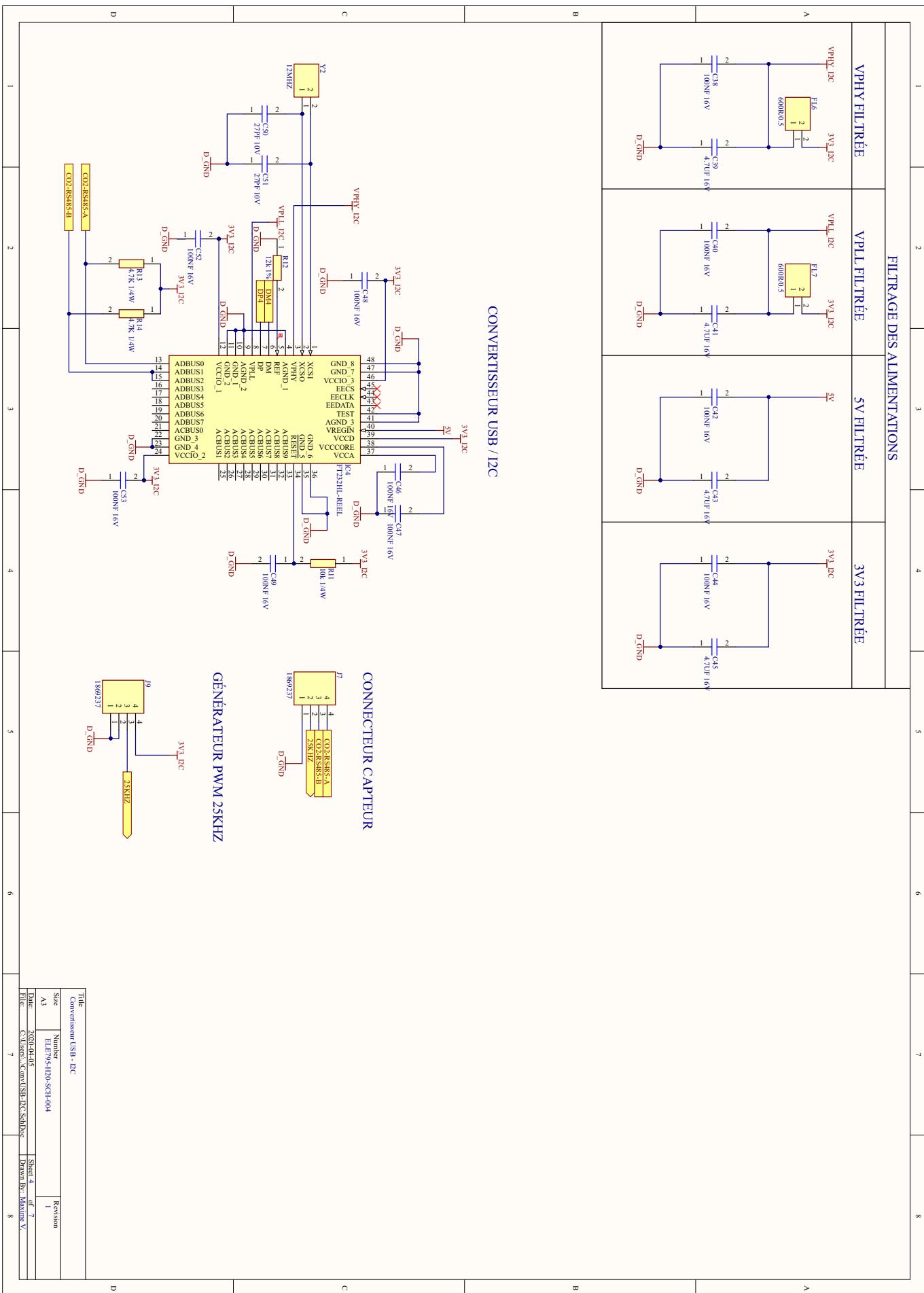
Cette annexe présente les schémas électriques produits avec Altium Designer. La liste suivante présente l'ordre dans lequel on retrouve les feuilles.

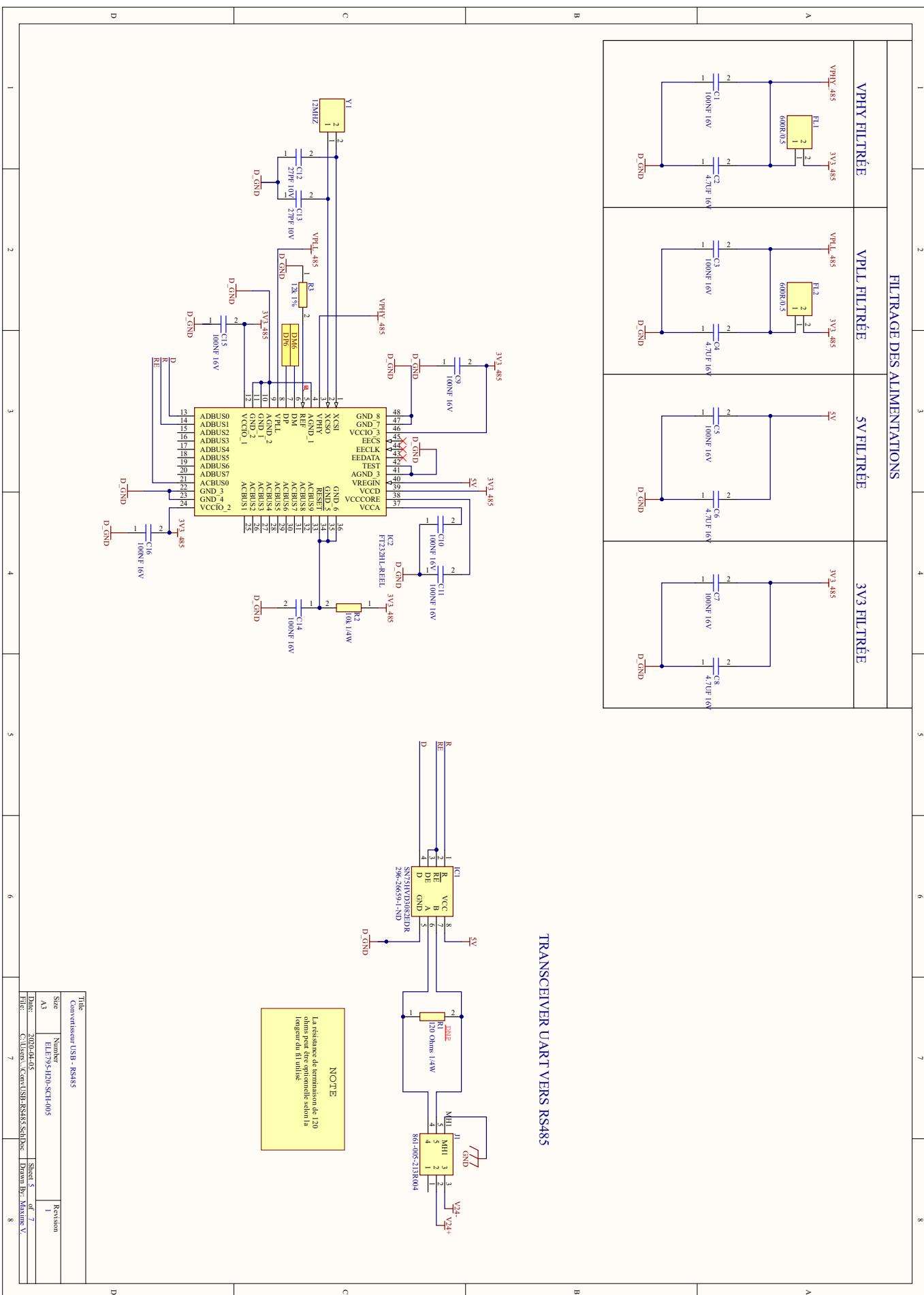
- Feuille du schéma de l'alimentation et son connecteur d'entrée
- Feuille du schéma du port vers le Jetson
- Feuille du schéma du concentrateur USB
- Feuille du schéma du convertisseur USB vers I²C
- Feuille du schéma du convertisseur USB vers RS485
- Feuille du schéma du convertisseur USB vers SPI
- Feuille du schéma des ports USB



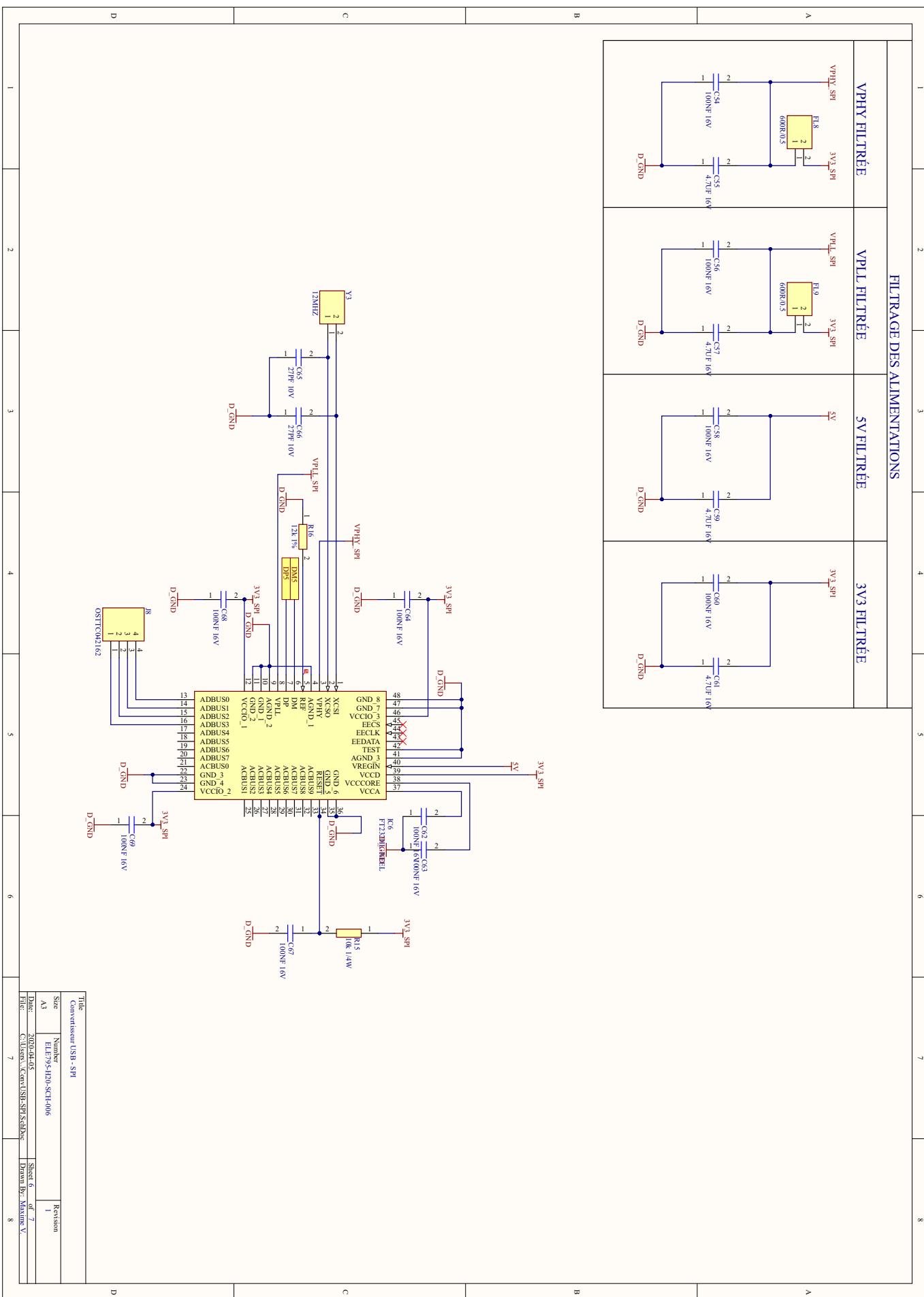


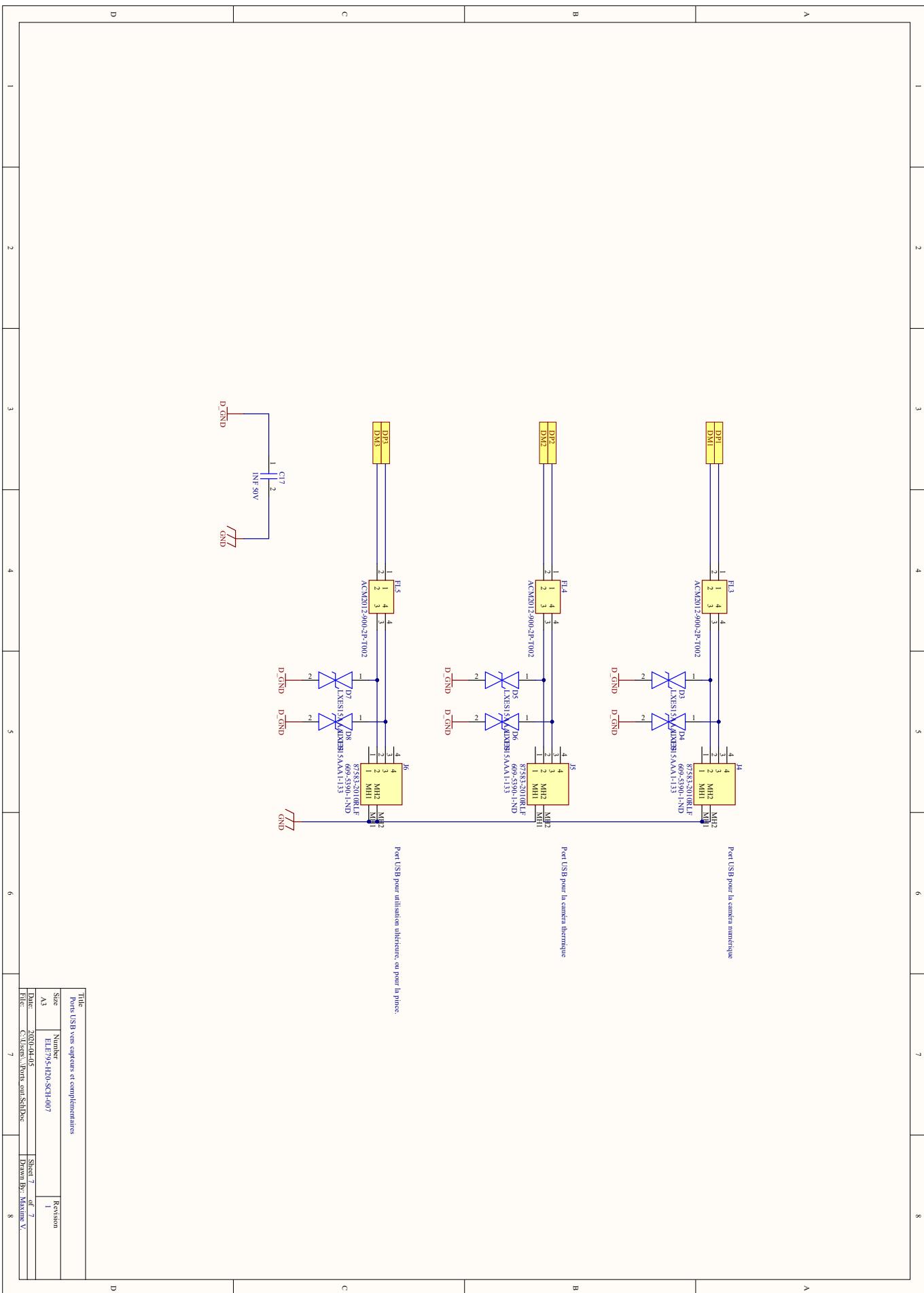






TRANSCEIVER UART VERS RS485





ANNEXE VII
LISTE DES COMPOSANTS

#	Revendeur	Numéro de pièce chez le revendeur	Prix par unité (USD)	Sous-total (USD)
1	Digi-Key	1276-1258-1-ND	0.037	0.999
2	Digi-Key	445-12981-1-ND	0.648	7.78
3	Digi-Key	720-VJ0805A270GXQCW1BCCT-ND	0.29	1.74
4	Mouser	80-C0402T103J4RACTU	1.41	4.23
5	Digi-Key	399-C0603X102K5RAC3316CT-ND	0.4	0.4
6	Digi-Key	311-1018-1-ND	0.024	0.384
7	Digi-Key	478-11846-1-ND	0.26	0.52
8	Digi-Key	1276-1451-1-ND	0.55	1.1
9	Digi-Key	490-13209-1-ND	0.28	2.24
10	Digi-Key	490-14039-1-ND	0.12	0.72
11	Digi-Key	445-2207-1-ND	0.52	2.08
12	Digi-Key	296-26659-1-ND	1.95	1.95
13	Digi-Key	768-1101-1-ND	4.25	12.75
14	Farnell	2496596	4.92	4.92
15	Farnell	2830296	19.76	19.76
16	Newark	98AC7184	1.23	1.23
17	Digi-Key	2092-KUSBXHT-BS1N-O-HRF-ND	1.93	1.93
18	Digi-Key	609-5390-1-ND	0.94	2.82
19	Digi-Key	277-5825-ND	2.57	5.14
20	Digi-Key	ED2611-ND	1.13	1.13
21	Digi-Key	P120BCCT-ND	0.63	0.63
22	Digi-Key	311-10KERCT-ND	0.1	0.3
23	Digi-Key	541-4.7KAGCT-ND	0.1	0.3
24	Mouser	603-RC0402FR-0714K7L	0.012	0.12
25	Digi-Key	311-4.70KFRCT-ND	0.1	0.2
26	Digi-Key	1866-3633-ND	15.88	15.88
27	Digi-Key	XC1770CT-ND	0.38	1.14

#	Indentification du composant	Qté	Manufacturier	Numéro de pièce chez le manufacturier
1	C1, C3, C5, C7, C9, C10, C11, C15, C16, C38, C40, C42, C44, C46, C47, C48, C52, C53, C54, C56, C58, C60, C62, C63, C64, C68, C69	27	Samsung	CL10F104ZO8NNNC
2	C2, C4, C6, C8, C39, C41, C43, C45, C55, C57, C59, C61	12	TDK	CGA5L3X8R1C475K160AB
3	C12, C13, C50, C51, C65, C66	6	Vishay Vitramon	VJ0805A270GXQCW1BC
4	C14, C49, C67	3	KEMET	C0402T103J4RACTU
5	C17	1	KEMET	C0603X102K5RAC3316
6	C18, C19, C22, C23, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37	16	Yageo	CC0402JRNP09BN220
7	C20, C24	2	Kyocera AVX	0603ZC101KAT2A
8	C21, C25	2	Samsung	CL05A106MQ5NUNC
9	D1, D2, D3, D4, D5, D6, D7, D8	8	Murata	LXES15AAA1-133
10	FL1, FL2, FL6, FL7, FL8, FL9	6	Murata	BLM18AG601BH1D
11	FL3, FL4, FL5, FL10	4	TDK	ACM2012-900-2P-T002
12	IC1	1	Rochester Elec.	SN75HVD3082EDR
13	IC2, IC4, IC6	3	FTDI	FT232HL-REEL
14	IC3	1	Texas Instruments	TUSB2077APTR
15	J1	1	NorComp	861-005-213R004
16	J2	1	Molex	52207-2033
17	J3	1	Kycon	KUSBXHT-BS1N-O-HRF
18	J4, J5, J6	3	Amphenol ICC	87583-2010RLF
19	J7, J9	2	Phoenix Contact	1869237
20	J8	1	On-Shore Tech.	OSTTC042162
21	R1	1	Panasonic	ERA8AEB121V
22	R2, R11, R15	3	Yageo	RC1206JR-0710KL
23	R3, R12, R16	3	Vishay Dale	CRCW02014K70JNED
24	R4, R5, R6, R7, R8, R9, R10	7	Yageo	RC0402FR-0714K7L
25	R13, R14	2	Yageo Phycomp	RC1206FR-074K7L
26	U1	1	Mean Well	NSD15-12S5
27	Y1, Y2, Y3	3	ECS International	ECS-120-20-3X-TR

BIBLIOGRAPHIE

- Adafruit. (2020a). Adadruit FT232H breakout using I2C. Repéré à <https://learn.adafruit.com/adafruit-ft232h-breakout/i2c>.
- Adafruit. (2020b). Adadruit FT232H breakout using SPI. Repéré à <https://learn.adafruit.com/adafruit-ft232h-breakout/spi>.
- Asch, G. (2011). *Acquisition de données : Du capteur à l'ordinateur* (éd. 3). dunod.
- Bapna, D., Rollins, E., Foessel, A. & Whittaker, R. (1998, may). Antenna pointing for high bandwidth communications from mobile robots. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, 4, 3468–3473 vol.4. doi : 10.1109/ROBOT.1998.680974.
- Blot, E. (s.d.a). Pyftdi for I2C. Repéré à <https://eblot.github.io/pyftdi/api/i2c.html>.
- Blot, E. (s.d.b). Pyftdi for SPI. Repéré à <https://eblot.github.io/pyftdi/api/spi.html>.
- Eppinger, S. & Seering, W. (1987, mar). Understanding bandwidth limitations in robot force control. *1987 IEEE International Conference on Robotics and Automation Proceedings*, 4, 904–909. doi : 10.1109/ROBOT.1987.1087932.
- FLIR. (2018). FLIR LEPTON 3 & 3.5. FLIR. Repéré à <https://www.flir.com/globalassets/imported-assets/document/lepton-3-3.5-datasheet.pdf>.
- FTDI. (2019). FT232H Single Channel Hi-Speed USB to Multipurpose UART/FIFO IC Datasheet. Repéré à https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232H.pdf.
- gouv., W. (2019). Carbon Dioxide. Repéré à <https://www.dhs.wisconsin.gov/chemical/carbon dioxide.htm>.
- Hughes, K. (2015). opencv blackfly. Repéré à <https://gist.github.com/kevinhughes27/5543668>.
- Ilyas, M. & Mahgoub, I. (Éds.). (2005). *Handbook of sensor networks : compact wireless and wired sensing systems*. Boca Raton : CRC Press.
- Lee W. Ritchey, J. Z. (Éd.). (2006a). *Right the First Time : a Practical Handbook on High Speed PCB and System Design Volume 1*. Speeding Edge.
- Lee W. Ritchey, J. Z. (Éd.). (2006b). *Right the First Time : a Practical Handbook on High Speed PCB and System Design Volume 2*. Speeding Edge.
- Li, W., McLernon, D., Wong, K.-K., Wang, S., Lei, J. & Zaidi, S. A. R. (2019). Asymmetric Physical Layer Encryption for Wireless Communications. *IEEE Access*, 7, 46959–46967. doi : 10.1109/ACCESS.2019.2909298.

- Licea, D. B., McLernon, D. & Ghogho, M. (2017). Mobile Robot Path Planners With Memory for Mobility Diversity Algorithms. *IEEE Transactions on Robotics*, 33(2), 419–431. doi : 10.1109/TRO.2016.2636848.
- Mankar, J., Chaitali, D., Komal, T., Madhura, K. & Prachi, S. (2014). REVIEW OF I2C PROTOCOL. *International Journal of Research in Advent Technology*, 2(1), 474–478.
- Piyu, D. (2018). Introduction to SPI Interface. par l'entreprise device analogue, Repéré à <https://www.analog.com/media/en/analog-dialogue/volume-52/number-3/introduction-to-spi-interface.pdf>.
- Robert, M. (2017). USB 101 : An Introduction to Universal Serial Bus 2.0. Repéré à <https://www.cypress.com/file/134171/download>.
- Telaire. (2017). Telaire T6713 Series CO 2 Module : Datasheet. Repéré à <https://www.amphenol-sensors.com/en/telaire/co2/525-co2-sensor-modules/3215-t6700>.
- Telaire. (2019). T67xx CO2 Sensor Module : Application note. Repéré à <https://www.amphenol-sensors.com/en/telaire/co2/525-co2-sensor-modules/3215-t6700>.
- Upadhyay, A. & Dhapola, A. (2015, mar). Embedded Systems And its Application in Medical Field. doi : 10.13140/2.1.1004.2406.
- Wikipedia. (2020). Gigabit Ethernet. Repéré à https://en.wikipedia.org/wiki/Gigabit_Ethernet#1000BASE-T.