



DOCUMENTATION TECHNIQUE : CARTE BANC DE TEST EL418

ENZO PERRIER

Table des matières

Introduction.....	4
1. Projet Banc de test intelligent	5
1.1 Présentation du projet	5
1.2 Présentation du feu TEMPO® Standard et de la carte CPU.....	6
1.3 Carte CPU	7
1.4 Principe de fonctionnement global du banc de test intelligent	8
1.5 Microcontrôleur	11
1.6 Alimentation	13
1.6.1 Protection de l'entrée contre les inversions de polarité	14
1.6.2 Alimentation de la carte EL418	15
1.6.3 Régulateur de Tension 3V3	15
1.7 Communication série RS232 et RS485	17
2.8.1 RS232.....	17
2.8.2 Conversion de niveaux RS232 \leftrightarrow TTL.....	18
2.8.3 RS485.....	20
2.8.4 Conversion de niveaux RS485 \leftrightarrow TTL.....	21
1.8 Entrées/Sorties	22
1.8.1 Commande des entrées de EL418.....	22
1.8.2 Entrées de la carte banc de test	23
1.8.3 Gestion des défauts ampoules	24
1.8.4 Sorties de la carte banc de test	25
1.8.5 Buzzer	25
1.9 Capteurs	26
1.9.1 Capteur de luminosité	26
1.9.2 Capteur de température/humidité.....	27
1.9.3 Capteur de courant/tension	27
ANNEXES.....	30

Table des illustrations

Figure 1: Image d'illustration feu Tempo® standard site web.....	6
Figure 2: Vue éclatée coupole jaune.....	6
Figure 3: Synoptique de base banc de test intelligent.....	8
Figure 4: Schéma d'implantation	9
Figure 5: Carte mentale rappel des besoins en E/S STM32.....	12
Figure 6: STM32F103C8T6	12
Figure 7: Schéma watchdog TPS3825-33 (KiCad)	13
Figure 8: Schéma simplifié des alimentations	13
Figure 9: Schéma du circuit de protection contre les inversions de polarité	14
Figure 10: Schéma de commande de l'alimentation de l'EL418	15
Figure 11: Montage Buck	16
Figure 12: Définitions des E/S du régulateur TPS54231 (Datasheet)	16
Figure 13: Schéma typique d'application pour $V_{out} = 3.3V$ (Datasheet)	17
Figure 14: Schéma du bloc régulateur de tension 3V3	17
Figure 15: chronogramme d'exemple communication RS232.....	18
Figure 16: Schéma du MAX3232 implanté.....	20
Figure 17: Fonctionnement signal différentiel.....	20
Figure 18: Comparatif signal sans Résistance fin de ligne/Avec résistance de fin de ligne.....	21
Figure 19: Schéma driver RS485.....	22
Figure 20: Schéma explication du tressage des fils transportant de la data.....	22
Figure 21: Schéma gestion des entrées de l'EL418.....	23
Figure 22: Schéma gestion des entrées côté carte banc de test.....	24
Figure 23: Schéma de gestion des défauts ampoules (identiques pour les 3 sorties)	24
Figure 24: Schéma de commande des sorties carte banc de test.....	25
Figure 25: Schéma Buzzer	26
Figure 26: Schéma cellule.....	26
Figure 27: Schéma HTU21D & I2C	27
Figure 28: Schéma mesure de tension AOP	28
Figure 29: Schéma mesure de courant avec résistance de shunt et AOP en non inverseur.....	28
Figure 30: Schéma INA219	29
Figure 31: Tableau d'assignement des adresses de l'INA219	29

Introduction

Dans le cadre de ma troisième année de BUT GEII (parcours Électronique et Systèmes Embarqués), j'ai eu l'opportunité de mener à bien un projet portant sur la conception d'un banc de test de carte électronique « amélioré ».

L'objectif de ce projet est d'optimiser les conditions de test afin de rendre le contrôle de la carte la plus produite et la plus vendue de l'entreprise à la fois plus fiable, plus rapide et moins contraignant pour le technicien.

Cette documentation technique présente le fonctionnement global du système ainsi que les principaux choix techniques effectués pour la partie électronique de la carte.

1. Projet Banc de test intelligent

1.1 Présentation du projet

L'entreprise TTS conçoit et fabrique des équipements de signalisation dynamique à LED, destinés à la sécurisation et à la régulation du trafic routier. Parmi ses produits phares figure le feu tricolore temporaire TEMPO®, largement utilisé sur les chantiers. Ce dispositif représente à lui seul une part importante du chiffre d'affaires de l'entreprise.

Ce feu repose sur une carte électronique principale, la carte CPU EL418, qui gère l'ensemble des fonctions critiques : contrôle des optiques, affichage, synchronisation, communication infrarouge et dialogues série. Étant donné la place centrale de cette carte dans le fonctionnement du produit, sa validation en sortie de production est indispensable.

Actuellement, cette étape est réalisée à l'aide d'un banc de test passif, qui permet au technicien de simuler manuellement certaines entrées et de vérifier les réponses de la carte. Bien que fonctionnel, ce banc nécessite de nombreuses manipulations, ce qui engendre des tests longs, parfois imprécis, et difficiles à répéter de manière rigoureuse sur de grandes séries. C'est dans ce contexte que j'ai proposé de manière totalement autonome le développement d'un banc de test intelligent.

Les objectifs de la carte banc de test, basé sur un microcontrôleur STM32, est de :

- Piloter automatiquement les entrées de la carte EL418
- Analyser ses réponses via communication série UART¹ (RS232)
- Simuler des défauts
- Afficher les résultats via un terminal

Objectifs du projet :

- ✓ Automatiser le processus de test des cartes EL418
- ✓ Réduire le temps moyen de test par carte
- ✓ Limiter les erreurs humaines lors des manipulations
- ✓ Concevoir une carte modulaire et évolutive, capable d'intégrer de futures fonctionnalités

¹ Universal Asynchronous Receiver-Transmitter

1.2 Présentation du feu TEMPO® Standard et de la carte CPU

Le feu standard se décompose en 2 parties :

- Un mât qui comporte 4 coupoles, contenant elles même l'électronique
- Un caisson permettant d'accueillir le mât et une batterie 12V, avec un boîtier de raccordement qui joue le rôle d'interface d'alimentation entre la batterie et le mât. (Cf. Annexe 1 : Synoptique Feu tricolore TEMPO® standard)

Les 4 électroniques se situent dans les coupoles : les 3 coupoles les plus hautes accueillent de simples optiques, (cartes simples avec des LEDs et des résistances, pas d'intelligence). Et finalement, la coupole la plus basse (la coupole jaune) accueille la carte CPU, qui est « le cerveau » du feu et qui va donc contrôler les optiques, le décompteur en plus de l'intégrer directement sur cette carte...



Figure 1: Image d'illustration feu Tempo® standard site web



Figure 2: Vue éclatée coupole jaune

Le feu tricolore est le produit phare de TTS, la qualité, la simplicité d'utilisation et la fiabilité en font un produit largement connu et apprécié, il s'agit là du produit le plus vendu parmi le catalogue.

1.3 Carte CPU

La carte CPU (EL418) est le « cerveau » du feu tricolore, elle est responsable de la gestion de toutes les fonctionnalités du feu. Cette carte se distingue par sa complexité, renfermant une multitude de fonctionnalités fascinantes et essentielles à son bon fonctionnement. Elle est totalement conçue par le BE (Bureau d'étude) de TTS et assemblée en France.

En mode standard, on peut utiliser jusqu'à 4 feux ces feux ne communiquent pas entre eux, la synchronisation se fait grâce à la télécommande. Le défi de cette carte est donc de garder une synchronisation stable au fil du temps (Il ne faudrait pas que les feux soient tous au vert en même temps !). Pour cela on utilise sur la carte un RTC qui est en quelque sorte une horloge très précise, et on peut corriger la faible dérive du quartz de façon logiciel (Base de temps) pour avoir les cartes le plus précises possibles, comme la stabilité et la précision du quartz dépend en grande partie des conditions de température et d'humidité, il pourrait aussi être intéressant de réaliser des tests plus poussés dans des conditions extrême.

Points principaux de la carte :

- ✚ Peut dialoguer avec un terminal en RS232 (transmission asymétrique), ce qui permet d'effectuer des tests et/ou de configurer la carte.
- ✚ 3 Entrées et 3 Sorties
- ✚ Peut envoyer ou recevoir des commandes/informations via la communication RS485 2 fils (transmission différentielle half-duplex)
- ✚ Possède un récepteur/émetteur infrarouge sur la face avant pour communiquer avec la télécommande et ainsi programmer les feux.
- ✚ Ecran LCD intégré pour l'affichage du choix du mode, affichage d'erreurs...
- ✚ Support d'accumulateur pour assurer la sauvegarde du mode de programmation en cas de problème de batterie
- ✚ Buzzer pour avertissement sonore en cas d'erreur/retour sonore pendant la programmation par l'utilisateur
- ✚ Programmation des feux par infrarouge à l'aide d'une télécommande propriétaire.
- ✚ Possède un RTC² qui permet une gestion précise du temps pour éviter une dérive trop importante une fois les feux programmés.

² Real Time Clock : circuit électronique qui maintient l'heure et la date en temps réel, même lorsque le système est éteint, en utilisant une petite batterie pour continuer à fonctionner en permanence

1.4 Principe de fonctionnement global du banc de test intelligent

L'idée est relativement simple, il suffit de remplacer les boutons pour les entrées et ceux pour la génération des défauts par des relais, ce qui permet alors de commander la commutation de façon automatique, on vérifie alors le bon fonctionnement grâce à des lectures terminal quand c'est possible, le technicien doit aussi pouvoir, il serait intéressant d'avoir un logiciel graphique dédié, mais on se contentera au début de messages envoyés via terminal.

Le synoptique ci-dessous représente les différents moyens mis en œuvre pour le test des différents points de base de la carte EL418 :

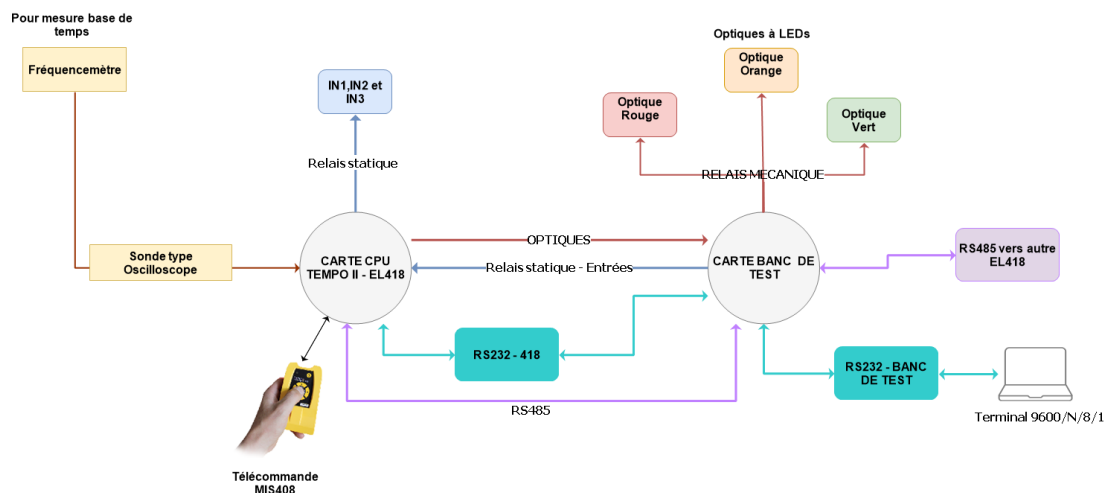


Figure 3: Synoptique de base banc de test intelligent

Schéma électronique complet en ANNEXES

J4-1	+ALIM_OUT1
J4-2	OUT1_418
J4-3	+ALIM_OUT2
J4-4	OUT2_418
J4-5	+ALIM_OUT3
J4-6	OUT3_418
J5-1	IN1_418
J5-2	GND
J5-3	IN2_418
J5-4	GND
J5-5	IN2_418
J5-6	GND
J6-1	OUT3_418
J6-2	OUT2_418
J6-3	OUT1_418
J6-4	+ALIM_418
J6-5	GND
J6-6	IN3_418
J6-7	IN2_418
J6-8	GND
J6-9	IN1_418
J6-10	GND
J6-11	+ALIM_418
J7-1	RS485_A
J7-2	RS485_B
J7-3	GND
J7-4	RS485_A_418
J7-5	RS485_B_418
J8-1	NC
J8-2	TX232
J8-3	RX232
J8-4	NC
J8-5	GND
J8-6	NC

J8-7	NC
J8-8	NC
J8-9	NC
J9-1	GND
J9-2	NC
J9-3	NC
J9-4	GND
J9-5	RS485_B_418
J9-6	RS485_A_418
J9-7	GND
J9-8	RS232_418
J9-9	TX232_418
J10-1	+3V3
J10-2	I2C_SCL
J10-3	I2C_SDA
J10-4	GND
J11-1	+ALIM
J11-2	OUT1
J11-3	OUT2
J11-4	OUT3
J11-5	OUT4

1.5 Microcontrôleur

Le microcontrôleur ou MCU est le cerveau du système, c'est lui qui fera l'acquisition des entrées et qui déclenchera les sorties, il devra alors permettre d'interagir avec la carte EL418 à tester.

Pour ce projet, le choix s'est porté vers un microcontrôleur STM32, tout d'abord pour son faible encombrement, ses performances et pour ses différents périphériques intégrés (I2C, UART, ADC...) mais aussi parce qu'il s'agit de microcontrôleurs très utilisés dans le domaine industriel et particulièrement chez TTS (la grande majorité des cartes comportant de l'intelligence utilisent ce type de microcontrôleurs).

Il faut maintenant bien choisir le modèle de microcontrôleur selon les besoins, nous pouvons alors faire un récapitulatif des besoins en termes d'E/S :

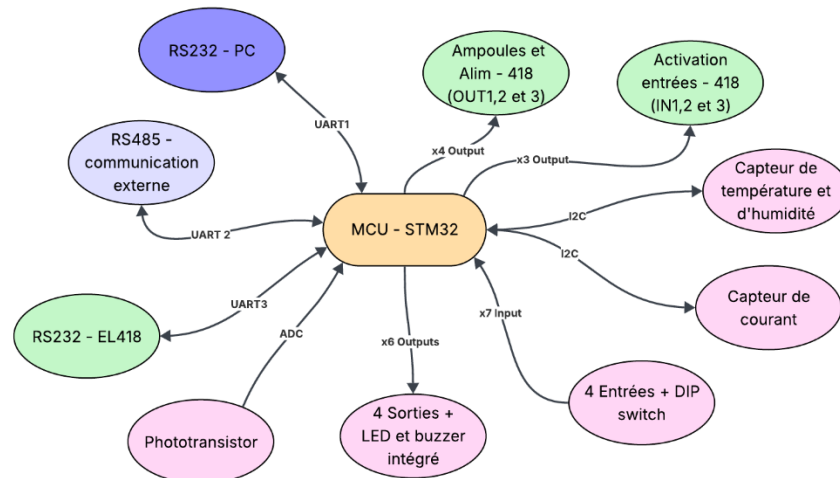


Figure 5: Carte mentale rappel des besoins en E/S STM32

Le modèle choisi adéquat à notre projet est le **STM32F103C8T6**, pour ces caractéristiques :

- 3 interfaces UART pour la communication série
- 37 broches GPIO configurables en entrée/sortie
- ADC³ 12bits avec jusqu'à 10 canaux d'entrée
- 2 interfaces I2C
- 3 timers principaux sur 16 bits
- 7 canaux DMA (Direct Memory Access) pour le transfert direct entre les périphériques et la mémoire sans implication du processeur
- Fréquence maximale de 72 MHz
- Prise en charge de RTOS (FREERTOS est intégré à l'IDE STM32 et CubeMx)



Figure 6: STM32F103C8T6

Nous utiliserons une horloge externe pour assurer une fréquence stable et pour assurer de bonnes réceptions et transmissions sur l'UART ainsi que des timers fiables, on programmera et debuggera le μ C grâce au port SWD du μ C en le connectant à un flasher.

Aussi on va utiliser un circuit qui fera office de watchdog⁴ ce composant permettra de superviser la tension et de réinitialiser le μ C pour le conserver dans un état stable malgré une chute soudaine de tension par exemple.⁵

³ Analog to Digital Converter (convertisseur Analogique vers Digital)

⁴ (Chien de garde en français) : Superviseur, système de surveillance

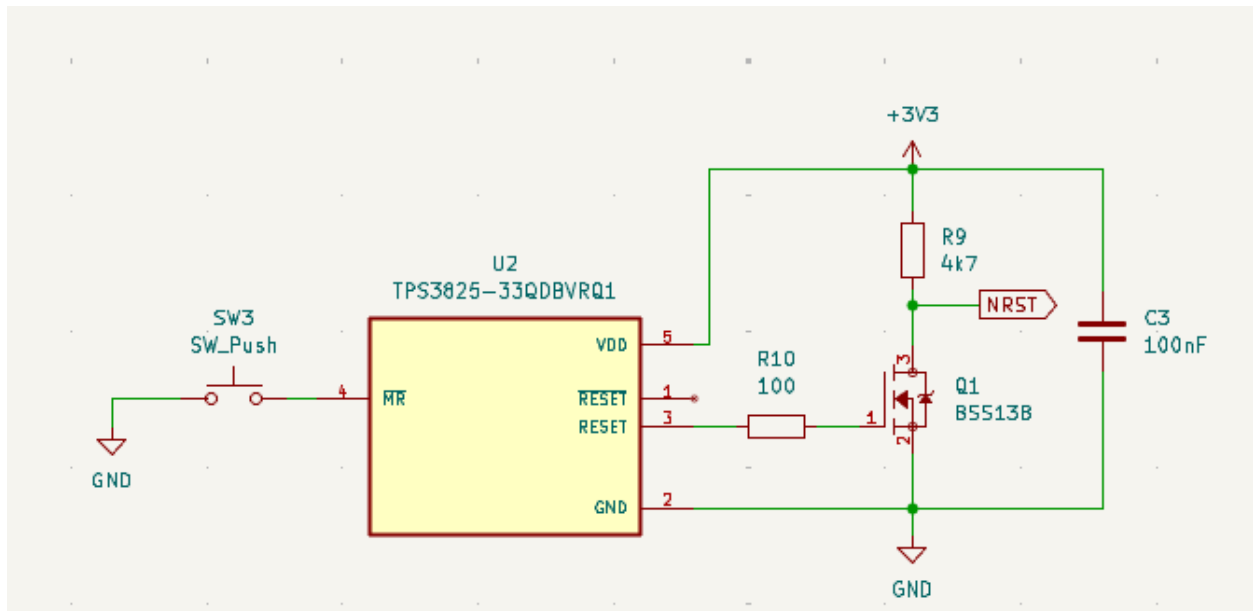


Figure 7: Schéma watchdog TPS3825-33 (KiCad)

1.6 Alimentation

L'alimentation générale viendra d'une alimentation de laboratoire, on pourra alimenter la carte en 12.5V, ce qui correspond à la tension de la carte EL418, ça nous permettra d'utiliser une seule alimentation pour la carte banc de test et l'EL418.

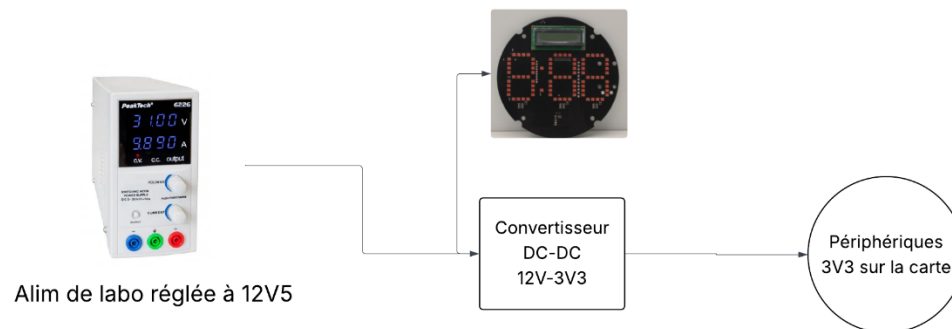


Figure 8: Schéma simplifié des alimentations

1.6.1 Protection de l'entrée contre les inversions de polarité

L'entrée d'alimentation sera protégée par un montage avec transistor PNP, il s'agit d'un montage qui, à l'instar d'une diode de protection permet de protéger contre les inversions de polarité, ce montage permet de réaliser la même fonction mais en réduisant grandement la chute de tension importante qui apparaît aux bornes d'une diode de protection. En plus du transistor PNP, nous ajoutons une résistance sur la grille pour limiter le courant ainsi qu'une diode Zener qui clamp⁶ les surtensions pour protéger le transistor qui est un semi-conducteur et est donc sensible aux surtensions.

Le principe de fonctionnement est simple, si la polarité n'est pas inversée, la tension V_{gs} (*grille source*) < 0 , le transistor conduit, sinon $V_{gs} > 0$, il est bloqué et le courant ne peut alors pas circuler.

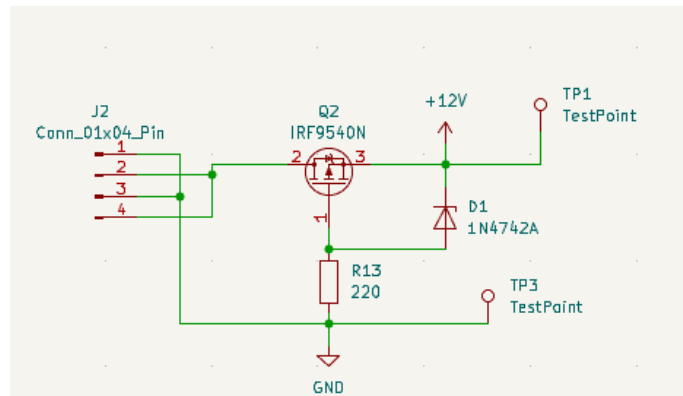


Figure 9: Schéma du circuit de protection contre les inversions de polarité

⁶ Limiter, ici limiter la tension en court circuitant l'excédent de tension à la masse

1.6.2 Alimentation de la carte EL418

La carte EL418 sera alimentée par la carte banc de test, l'alimentation qui vient de l'alim de labo permettra directement d'alimenter la carte, comme on travaille sur des alims de labo stabilisées, on n'équipe pas de protection particulière car le courant peut alors être limité (3A max), aussi, il sera nécessaire pour certaines étapes du test de pouvoir éteindre la carte, le plus simple est alors d'utiliser un interrupteur qui sera commandé par le μC , nous allons donc utiliser un relais mécanique.

Comme le relais utilisé a une tension de bobine de 12V, on utilise un transistor qui sera commandé par le μC et qui viendra « ouvrir et fermer » le circuit de la bobine côté 12V et qui viendra donc coller ou non le relais. Comme nous utilisons un transistor FET, nous utilisons une résistance de pull down car ces transistors ont un effet capacitif avec la source et le drain à cause de l'isolation avec la grille, alors au démarrage tant que la tension aux bornes de la grille est flottante, il pourrait être dans un état instable.

Aussi, comme le relais est un solénoïde, il s'agit alors d'une charge inductive, ce qui signifie que comme pour les moteurs, lorsque l'on coupe le courant dans la bobine, l'énergie emmagasinée dans la bobine ($W = \frac{1}{2} Li^2$) voudra se dissiper, ce qui va créer une tension inverse qui peut alors endommager des composants et particulièrement les semi-conducteurs, la diode de roue libre permettra le passage du courant pendant la phase de commutation pour pouvoir décharger correctement cette énergie emmagasinée. On ajoute également une résistance de 100 Ω pour limiter le courant dans la bobine du relais.

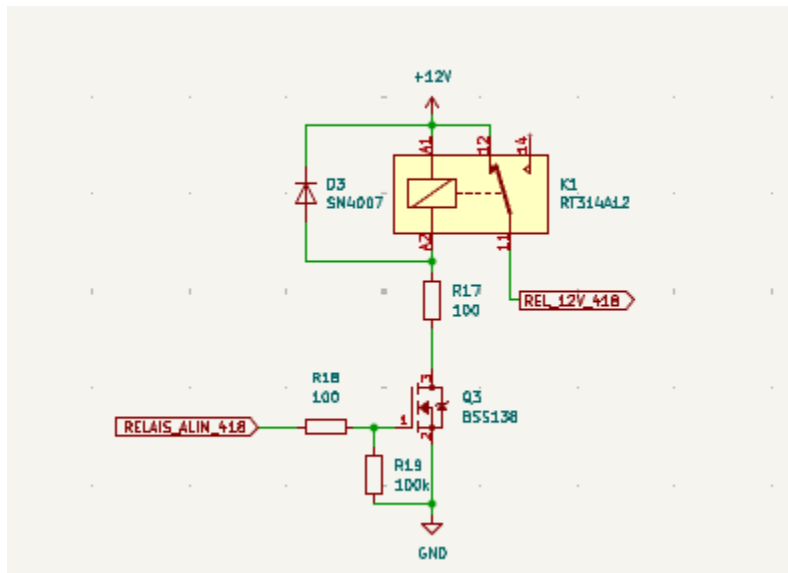


Figure 10: Schéma de commande de l'alimentation de l'EL418

1.6.3 Régulateur de Tension 3V3

Pour l'alimentation du microcontrôleur et des composants actifs, une tension unique de 3,3 V sera utilisée. Celle-ci sera fournie par un régulateur **TPS54231**, choisi pour plusieurs raisons : il est compact, économique, stable et précis. Il offre également une large plage de tension d'entrée (jusqu'à 28 V), ce qui protège la carte en cas de mauvais réglage de tension sur l'alimentation de laboratoire.

Ce composant a également été retenu pour des raisons de cohérence et de maintenance : il est déjà

utilisé sur l'EL418 ainsi que sur d'autres cartes, ce qui signifie qu'il est disponible en stock et bien maîtrisé par l'équipe. Cela facilite les éventuels dépannages et garantit une bonne fiabilité du montage.

Le TPS54231 est un régulateur **Buck** (Schéma ci-dessous), son fonctionnement repose sur un régulateur à découpage, le montage est composé d'un transistor, une diode de roue libre, une inductance et un condensateur de filtrage. (Voir **Erreur ! Source du renvoi introuvable.**)

Figure 11: Montage Buck

Pour ce montage, il faut bien dimensionner un certain nombre de composants périphériques. Je parle notamment des valeurs de l'inductance et du condensateur de filtrage. L'inductance ne doit pas être trop faible, sinon on risque de se retrouver en conduction discontinue, ce qui nuirait au bon fonctionnement du circuit (le courant dépend du courant de sortie dans ce cas). À l'inverse, si elle est trop élevée, on augmente les pertes, ce qui n'est pas idéal non plus. De son côté, le condensateur de filtrage doit être correctement dimensionné pour garantir une tension en sortie la plus stable et la plus continue possible.

Il y a aussi d'autres composants à prendre en compte, surtout ceux liés à la boucle d'asservissement, ainsi que les condensateurs de découplage qu'il est vivement conseillé d'utiliser. D'ailleurs, pour une meilleure efficacité contre les perturbations, on recommande de combiner plusieurs types de condensateurs céramiques pour les hautes fréquences, et chimiques pour les plus basses.

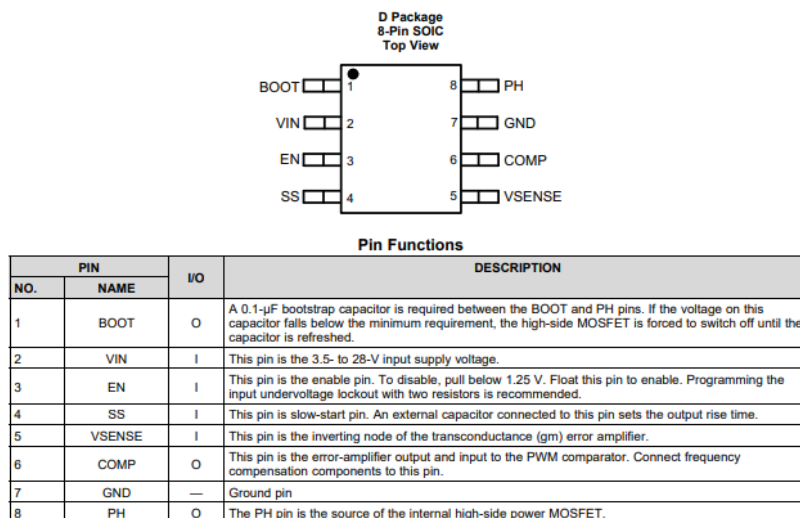


Figure 12: Définitions des E/S du régulateur TPS54231 (Datasheet)

Pour le choix des valeurs, le schéma typique d'application de la datasheet correspond exactement à notre utilisation, nous allons alors simplement utiliser les valeurs de composants utilisés dans le schéma du fabricant.

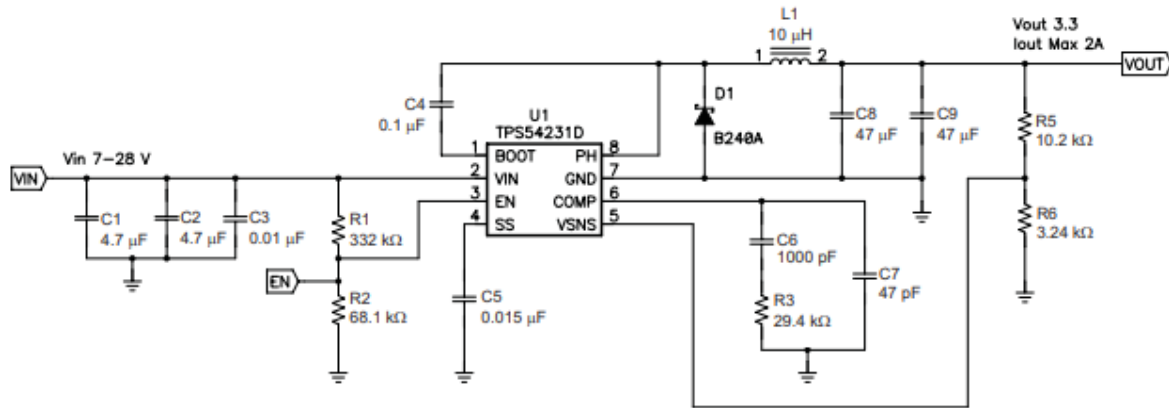


Figure 13: Schéma typique d'application pour $V_{out} = 3.3V$ (Datasheet)

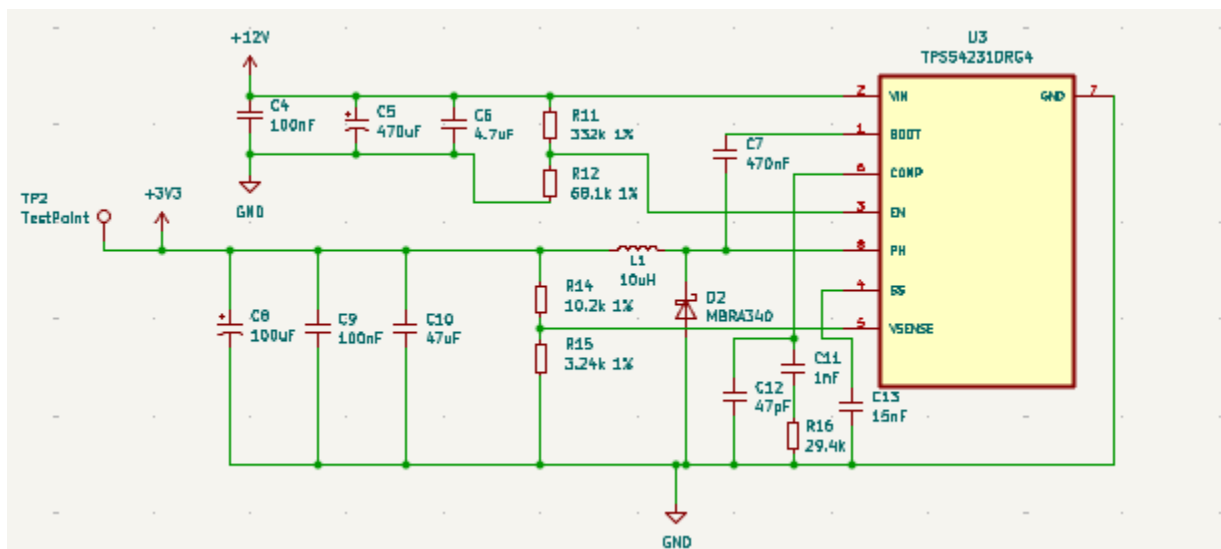


Figure 14: Schéma du bloc régulateur de tension 3V3

1.7 Communication série RS232 et RS485

2.8.1 RS232

RS232 (Recommended Standard 232) est une norme de communication série développée dans les années 1960 pour permettre la communication entre des ordinateurs et des périphériques. Cette norme définit les signaux électriques, les connecteurs ainsi que le protocole de transmission des données.

La RS232 est un signal dit asymétrique full-duplex car il fonctionne avec 2 fils de data (Rx pour la réception et Tx pour la transmission) et une masse comme référence. La donnée est donc le niveau de tension entre le fil de donnée et la masse. Il est possible d'envoyer et de recevoir des données en même temps car celles-ci transitent sur 2 fils distincts. Les données sont envoyées bit à bit et les niveaux de tensions sont binaires (0 ou 1). Il est aussi important de noter que cette communication est asynchrone, c'est-à-dire qu'elle n'a pas besoin d'horloge comme la

communication I2C ou SPI.

Il faut alors configurer les appareils série avec :

- La vitesse en **bauds** (bits par seconde) : généralement de 300 à 115200 bauds, plus cette valeur est élevée plus la communication est rapide, la limite de vitesse sera imposée par le matériel.
- La parité : Il s'agit d'un mécanisme très simple de contrôle d'erreur et elle permet de vérifier si les données reçues sont correctes, on a :
 - N (None) : pas de contrôle de parité, donc pas de vérification
 - E (Even) : parité paire, le nombre total de bits à 1 dans la trame doit être pair.
 - O (Odd) : parité impaire, le nombre total de bits à 1 doit être impair.
- Data bits : C'est le nombre de bits utiles (on ne compte pas bits de start, stop...), généralement entre 5 et 8 mais le plus courant est un octet (8 bits)
- Stop bits : Permet de signaler au récepteur qu'un octet est complet et qu'il peut se préparer à recevoir le prochain.

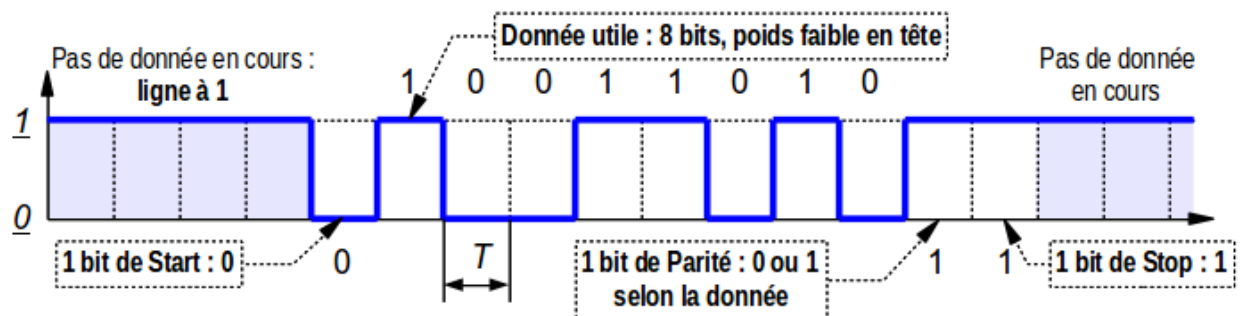


Figure 15: chronogramme d'exemple communication RS232

Dans notre cas, comme il s'agit simplement d'une utilisation terminale sur courte distance, on se contentera d'une configuration classique en **9600/N/8/1** avec le PC, et côté carte EL418, la communication se fait aussi en **9600/N/8/1**.

Aujourd'hui même si ce mode de communication elle n'est plus réellement utilisée dans le domaine grand public, elle est encore largement utilisée dans les systèmes industriels pour sa fiabilité, sa simplicité et son faible coût.

Dans notre cas, la plupart des cartes communicant en RS232 permettent un accès terminal pour le test et la configuration. C'est le cas de l'EL418 qui se test en grande partie via le terminal, nous utiliserons alors les retours du terminal pour vérifier le fonctionnement.

2.8.2 Conversion de niveaux RS232 \leftrightarrow TTL

Pour recevoir et envoyer des données RS232 à partir de notre μC^7 on ne peut pas se contenter de brancher la RS232 directement sur un port UART du μC , en RS232 on a :

- Niveau logique 1 (MARK) : entre **-3** et **-15V**
- Niveau logique 0 (SPACE) : entre **+3** et **+15V**

Alors que sur le μC STM32, on utilise des niveaux TTL (Transistor-Transistor Logic), ce qui représente (sur le STM3F103C8T6) :

- Niveau logique 1 : **3.3V**

⁷ μC : abréviation de microcontrôleur

- Niveau logique 0 : **0V**

On se doit alors d'utiliser un circuit qui permettra de réaliser l'interface entre les 2 périphériques (UART du STM32 et RS232). Mon choix s'est alors tourné vers le **MAX3232**, car c'est un composant relativement simple à implanter, peu cher et que j'ai déjà utilisé.

Il s'agit d'un convertisseur à double canal (ce qui nous sera utile pour traiter les commandes de l'utilisateur et l'affichage terminal côté PC et aussi les retour et l'envoi de commandes via la carte EL418). Ce composant va convertir de façon bidirectionnelle les niveaux RS232 en TTL (tensions entre GND et V_{lim} du composant). La conversion TTL → RS232 est réalisée à l'aide de transistor qui commutent pour amplifier et inverser le signal TTL :

- 3.3V (TTL logique 1) : Le transistor est saturé, ce qui fait passer une tension positive sur la sortie (généralement +12V ou +15V), correspondant à la logique RS-232 "0".
- 0V (TTL logique 0) : Le transistor est bloqué, ce qui génère une tension négative sur la sortie (généralement -12V ou -15V), correspondant à la logique RS-232 "1".

Le composant doit générer une tension plus élevée que sa tension d'alimentation (ici du 12V à partir de 3V3), le MAX3232 intègre alors un montage **boost** (étudié à l'IUT) qui permet de réaliser cette conversion, il s'agit plus exactement d'un boost pour le +12V et un buck-boost pour les tensions négatives -12V.

Le schéma de câblage standard est donné dans la datasheet on remarquera qu'il est nécessaire de câbler des condensateurs de filtrage du signal (V_{s+} et V_{s-}) pour avoir un signal propre au niveau du RS232, ainsi que des condensateurs de bootstrap, ceux-ci permettent de fournir une tension de commutation aux transistors légèrement supérieur à la tension d'alimentation, ça assure la bonne commutation des transistors FET, car ce type de transistors est commandé grâce à sa tension grille source, et il se peut que pour les transistor en high-side (source pas connectée directement à la masse), la commutation ne puisse pas se faire car cette différence de potentiel n'est pas suffisante pour saturer le transistor (problème abordé en BUT2 sur l'amplificateur classe D en SAE Son) Le condensateur de bootstrap vient alors régler ce problème en se chargeant et donc en apportant, de par sa réserve d'énergie, une tension légèrement supérieur pour pouvoir saturer le FET.

La valeur des condensateurs de bootstrap ne doit pas être trop grande pour pas qu'il soit trop long de le recharger, ni trop petite pour avoir une tension de commande suffisante en sortie. La valeur doit théoriquement se calculer grâce à cette formule :

$$C_{bootstrap} = \frac{I_{charge} * t_{on}}{\Delta V_{bootstrap}}$$

On se contentera de choisir la valeur standard (1μF) donnée par le constructeur car nous l'utilisons à des fréquences standards qui ne sont pas trop critiques à traiter.

Pour la conversion RS232 → TTL, le MAX3232 utilise principalement des diviseurs de tension et des diodes de redressement pour abaisser les niveaux RS232.

- +12V : la diode de redressement est passante et le signal est drainé vers la base d'un transistor, qui est interprété comme un niveau logique 1 TTL.
- -12V : La diode est bloquée, le transistor est bloqué aussi, on a alors un niveau logique 0 TTL.

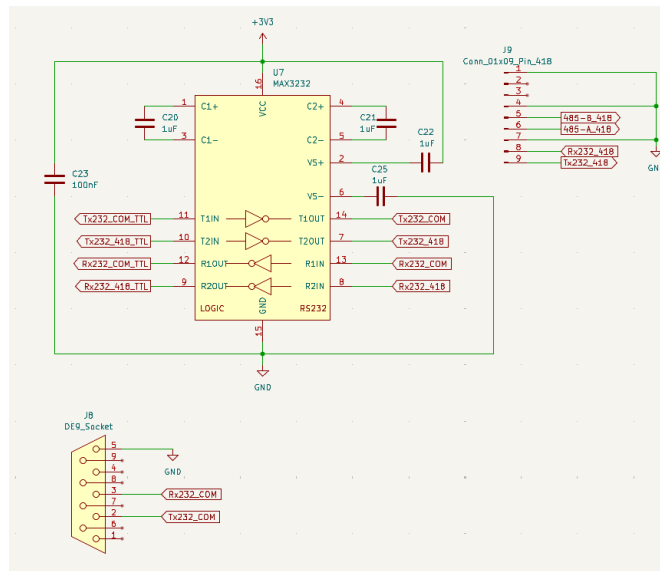


Figure 16: Schéma du MAX3232 implanté

2.8.3 RS485

RS485 (Recommended Standard 485) est une norme de communication série également développée par l'EIA (Electronic Industries Alliance) pour les systèmes de communication à longue distance et pour la communication multipoint. Introduite dans les années 1980, la RS485 est utilisée principalement dans des environnements industriels pour relier plusieurs périphériques à un bus de communication. Cette norme est couramment utilisée dans les automates programmables, les systèmes de contrôle industriel, ainsi que dans les réseaux de capteurs et de surveillance.

Contrairement à la **RS232**, qui utilise un signal asymétrique, la **RS485** repose sur un signal différentiel (2 fils A et B), le résultat de la donnée est la différence entre les 2 signaux.

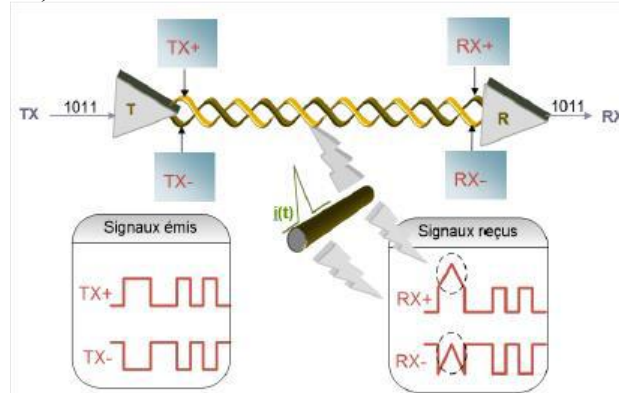


Figure 17: Fonctionnement signal différentiel

Cela lui permet de communiquer sur des distances beaucoup plus longues (jusqu'à 1200 mètres pour la RS485 contre 15m pour la RS232) et d'atteindre des vitesses de transmission plus élevées tout en restant robuste face aux interférences électromagnétiques. La RS485 est souvent utilisée en **half-duplex**, où la communication se fait dans un seul sens à la fois, mais il est possible d'adopter une communication **full-duplex** dans certains cas en utilisant tout simplement 4 fils, où une paire gère la transmission et l'autre la réception.

Chez TTS, cette norme de communication est utilisée pour faire communiquer différents équipements entre eux du fait que l'on peut utiliser la RS485 pour un bon nombre d'équipement sur le même bus et à des distances importantes, c'est pourquoi on retrouve la RS485 sur certaines options de feux tricolores nécessitant un mode de communication entre eux.

Comme la RS485 n'est pas utilisée sur le soft standard de l'EL418, on ne pourra pas la tester sur la carte (habituellement pas testée). En revanche il est intéressant de pouvoir ultérieurement tester cette fonctionnalité et/ou permettre à différentes carte banc de test de communiquer entre elles ou avec d'autres appareils.

2.8.4 Conversion de niveaux RS485 \leftrightarrow TTL

Pour la communication RS485 on utilisera un driver qui permet facilement de gérer la conversion différentielle RS485 vers TTL, ce composant est le **MAX3485**, on utilisera le système en half-duplex. Pour sélectionner le sens de communication (Transmission ou réception, on utilise un bit de RTS⁸, cela permettra de contrôler le sens du flux et d'indiquer au driver s'il s'agit d'une transmission ou d'une réception, pour se faire, 1 pin (RE) permet, quand tiré à la masse d'activer le mode de réception et un pin (DE) permet quand tiré à l'état haut d'activer le mode de transmission, on câblera alors simplement les 2 pins ensemble sur un pin digital de sortie du μ C, quand le pin sera à l'état bas, on sera en réception et à l'état haut en transmission.

On ajoute aussi des diodes **TVS⁹ bidirectionnelles** (car comme il s'agit de signal différentiel, on peut avoir $V_A - V_B < 0$ ou $V_A - V_B > 0$), cela permettra alors de protéger le signal contre les surtensions causées par les transitoires rapides en court-circuitant les tensions dépassant le seuil à la masse. On ajoute également une résistance de 120Ω qui pourra être utilisée ou non grâce à un jumper et qui servira de résistance de fin de ligne pour l'adaptation de la ligne, cela permet d'éviter la réflexion du signal sur d'éventuelles longues distances de câble

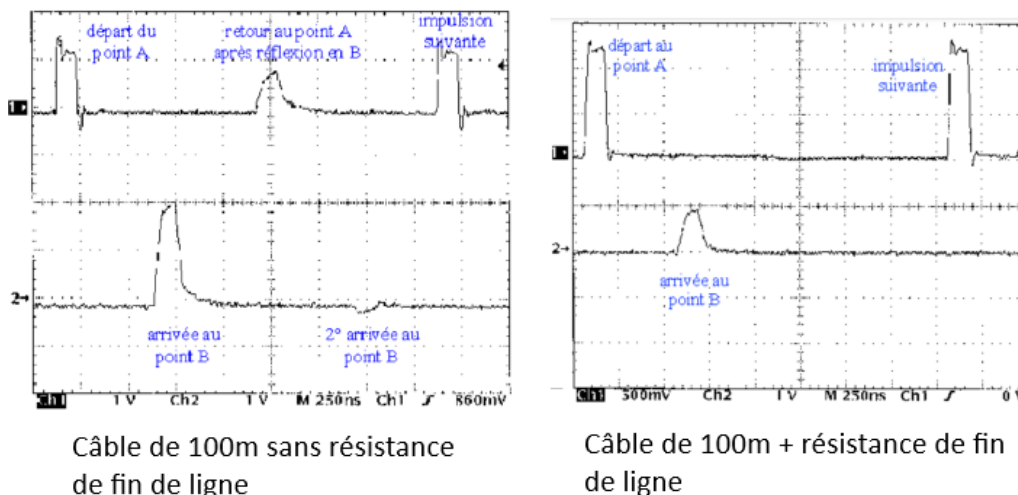


Figure 18: Comparatif signal sans Résistance fin de ligne/Avec résistance de fin de ligne

⁸ Request to send : Bit permettant de contrôler le sens du flux de data

⁹ Transient Voltage Supression : diode de suppression de surtension

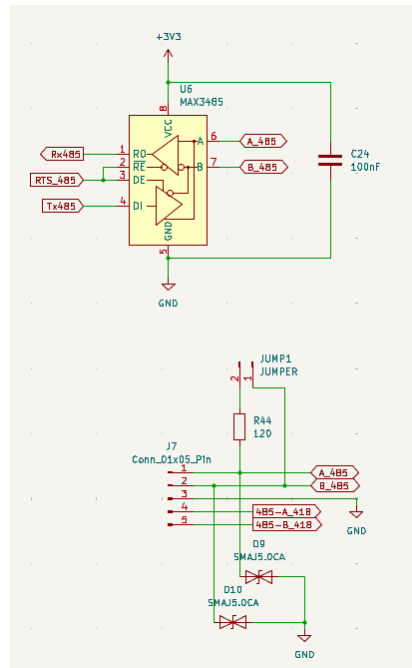


Figure 19: Schéma driver RS485

Sur de longues distances, on utilisera des câbles à paires torsadées. Cela permettra de limiter les perturbations qui s'annuleront 2 à 2, il est possible de faire l'analogie avec les courants de foucault, l'association de paire torsadée et de communication différentielle permet donc une communication plus fiable.

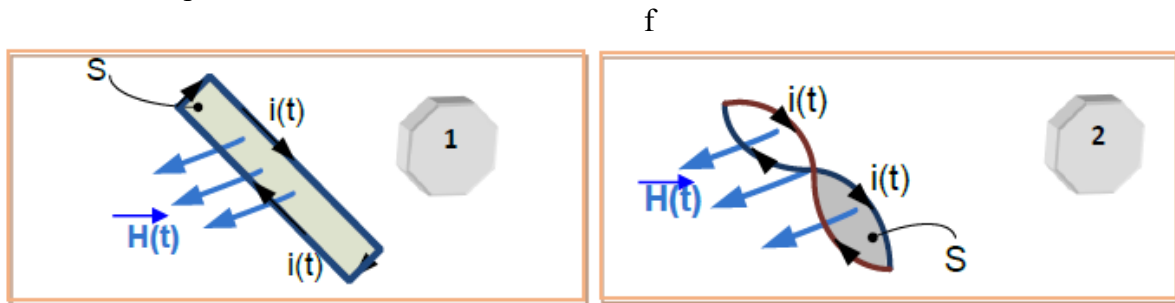


Figure 20: Schéma explication du tressage des fils transportant de la data

1.8 Entrées/Sorties

On traitera dans cette section les différents moyens mis en œuvre pour commander les entrées/sorties de l'EL418.

1.8.1 Commande des entrées de EL418

Pour commander les 3 entrées (IN1, 2 et 3) de l'EL418, le choix a été fait d'utiliser des optocoupleurs, cela assure l'isolation galvanique, ce qui permet d'éviter une éventuelle circulation de courant indésirable et d'éviter d'éventuelles problèmes d'instabilité. L'anode sera alors constamment connectée au +3V3, il suffira de tirer la cathode à la masse côté μC pour « allumer » la LED de l'optocoupleur et rendre passant le phototransistor, ce qui permettra d'activer l'entré.

Nous utiliserons, par soucis d'encombrement un seul composant qui pourra alors gérer 4 canaux. Comme seulement 3 canaux sur les 4 seront utilisés, on peut utiliser la dernière pour câbler une LED qui fera office de LED « built-in » qui nous servira de LED de vie.

Le composant que nous utilisons est le **TCMT4600**, c'est un composant qui nous sera largement suffisant pour l'utilisation que l'on compte en faire. Il faut simplement calculer la résistance de la LED à choisir pour avoir un courant suffisamment élevé pour que le phototransistor soit passant ; mais pas trop pour ne pas casser la LED :

$$R_{LED} = \frac{V_{CC} - V_{LED}}{I_{LED}}$$

Nous voulons avoir un courant dans la LED de 10 mA, (selon la datasheet, cela sera largement suffisant), la datasheet nous donne une tension de seuil pour la diode de **1.2V** donc :

$$R_{LED} = \frac{3.3 - 1.2}{0.01} = 210\Omega \text{ (On utilisera une résistance de } 200\Omega \text{ car j'en possède déjà).}$$

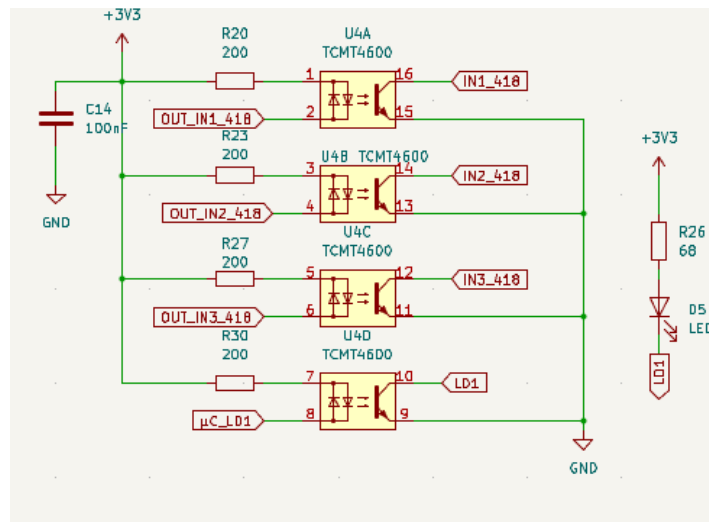


Figure 21: Schéma gestion des entrées de l'EL418

1.8.2 Entrées de la carte banc de test

Il sera utile d'avoir des boutons pour la gestion des différentes actions sur la carte banc de test (passer à l'étape suivante, recommencer le test...). Le choix a été fait d'utiliser les mêmes optocoupleurs que pour la gestion des entrées de l'EL418. Ici on vient protéger les entrées avec des diodes TVS (**PESD3V3L4UG**), il s'agit d'un composant comportant 4 diodes TVS. On vient aussi ajouter un condensateur aux bornes des sorties pour atténuer l'effet de rebonds habituels de façon hardware¹⁰ Les entrées côté µC seront câblées en pullup.

¹⁰ Hardware : physique ≠ Software qui signifie « logiciel »

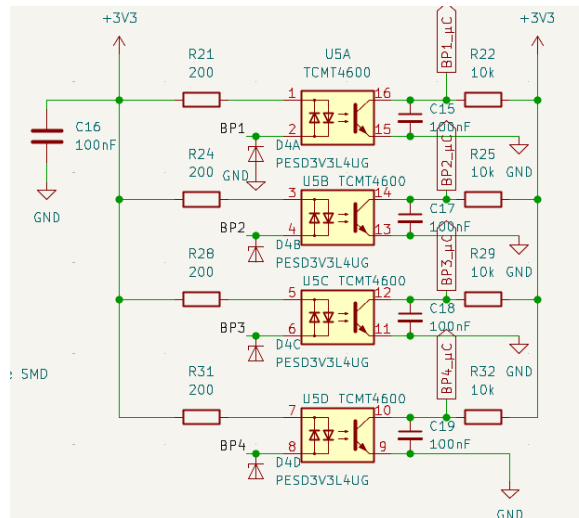


Figure 22: Schéma gestion des entrées côté carte banc de test

1.8.3 Gestion des défauts ampoules

Les défauts ampoules sont générés en supervisant la connexion de la carte LED sur l'EL418, pour générer ces défauts ampoules, sur le banc de test classique actuel, nous avons des interrupteurs qui viennent couper la connexion entre le +12V entre la carte LED et l'EL418. Ici on va remplacer les interrupteurs par des relais mécanique, car c'est simple à mettre en place, et de plus le bruit de claquement des relais servira de retour sonore sur l'activation et la désactivation du défaut. Le montage utilisé sera similaire à celui pour l'alimentation de l'EL418. On utilisera un transistor MOSFET avec comme protection une résistance pour limiter le courant dans la bobine du relais ainsi qu'une diode de roue libre **SM4007** qui peut supporter jusqu'à 1000V de tension inverse et sera alors idéale pour supporter la surtension générée par la bobine du relais.

Une résistance de 100Ω permettra de limiter le courant dans la grille du MOSFET et une autre résistance de pull down de 100kΩ entre la grille et la masse permettra de fixer l'état au repos du transistor à l'état bloqué et évitera des comportements aléatoires au démarrage.

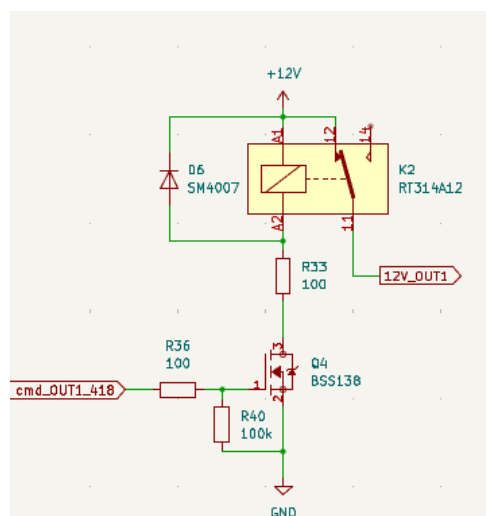


Figure 23: Schéma de gestion des défauts ampoules (identiques pour les 3 sorties)

1.8.4 Sorties de la carte banc de test

Nous allons aussi ajouter des sorties supplémentaires de la carte banc de test, ce qui permettra d'utiliser par exemple des LEDs pour l'état du test... Pour se faire, nous utilisons toujours le même montage que pour la commande des relais avec le transistor, mais cette fois ci on utilisera un transistor qui viendra ou non tirer à la masse la charge branchée entre un +alim et le drain du transistor. On choisit un transistor adéquat qui permet de faire passer un courant important, il s'agit du **RFD16N06**, qui peut faire passer jusqu'à 16A de courant max, ce qui permettra d'utiliser des charges plus gourmandes en courant sans problème.

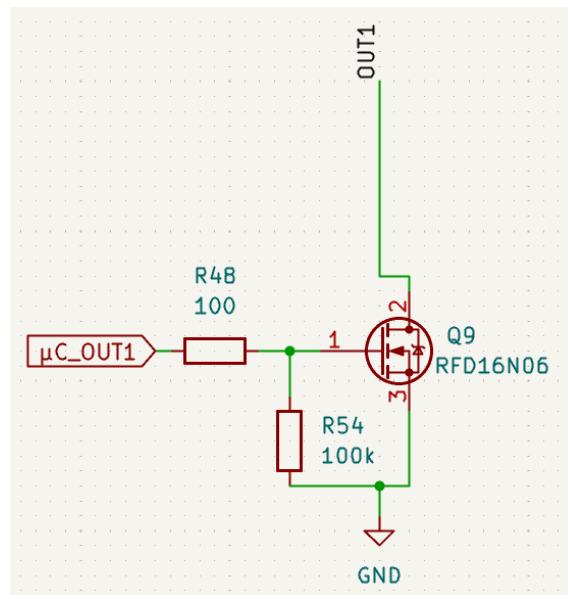


Figure 24: Schéma de commande des sorties carte banc de test

1.8.5 Buzzer

Pour alerter d'éventuels défauts majeurs, la carte sera équipée d'un buzzer, pour faire simple, il sera simplement commandé par un transistor et pour éviter d'avoir à utiliser une PWM¹¹, on se doit d'utiliser un buzzer dit **actif**, cela signifie qu'il comporte déjà les composants nécessaires pour générer un son, cela nous simplifie la tâche, d'autant plus que ce buzzer servira uniquement d'avertisseur sonore.

¹¹ Pulse Width Modulation ou MLI (Modulation de Largeur d'Impulsion) en français, signal numérique où la largeur des impulsions est modulée pour représenter des valeurs analogiques

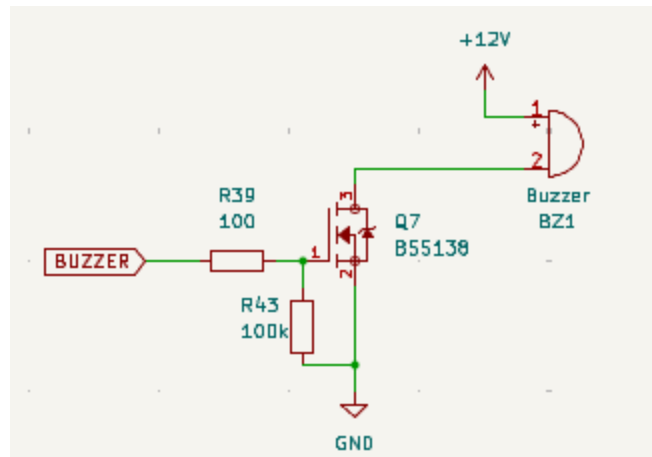


Figure 25: Schéma Buzzer

1.9 Capteurs

Des capteurs seront également implantés pour vérifier les mesures de la carte EL418 et/ou pour permettre de superviser certains tests spécifiques.

1.9.1 Capteur de luminosité

Pour le capteur de luminosité, nous allons utiliser le même montage que la carte EL418, cela permettra d'avoir théoriquement exactement la même valeur mesurée par l'ADC du μC de la carte banc de test que celui de l'EL418. Le fonctionnement est simple.

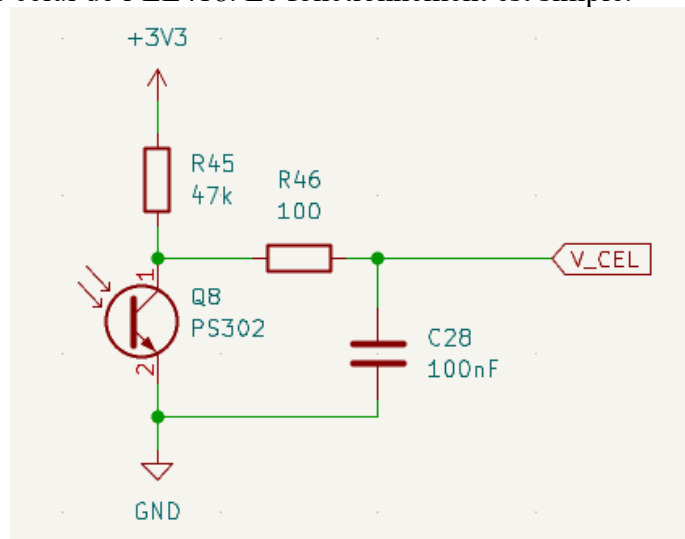


Figure 26: Schéma cellule

Cela permet d'avoir :

- Quand le transistor est **bloqué** : $V_{cel} = 3V3$
- Quand le transistor est **passant** : $V_{cel} = 0V$

1.9.2 Capteur de température/humidité

Le capteur de température et d'humidité ne servira pas pour le test de cartes neuves, mais permettra d'autres utilisations de cette carte banc de test, par exemple pour des tests longue durée à des températures extrêmes, ou encore pour réaliser des statistiques de dérive de base de temps de l'oscillateur quartz (il s'agit d'un composant important qui est sensible à la température). Nous utiliserons alors un composant qui se branchera sur le bus I2C¹² c'est un protocole de communication qui est léger à câbler du fait qu'il n'utilise que 2 fils : SDA (données) et SCL (clock). Chaque périphérique a une adresse unique. C'est une communication maître-esclave, le maître génère l'horloge et effectue des requêtes et les esclaves répondent aux requêtes. La transmission se fait de manière bidirectionnelle, et les données sont envoyées par paquets, ce qui promet une communication efficace entre plusieurs périphériques.

L'avantage de ce protocole est que comme il s'agit d'un bus, il sera possible d'ajouter encore d'autres périphériques ultérieurement sans problème, cela permet alors de faire « des économies » sur les GPIO du µC.

On doit implanter des résistances pull up du fait que l'I2C utilise une communication ouverte-drain et donc la ligne peut être tirée à un niveau '0' mais pas un niveau '1', cela permet d'éviter que les lignes restent flottantes.

Le capteur utilisé est un **HTU21D**, il s'agit d'un capteur relativement peu encombrant qui embarque à la fois un capteur de température et d'humidité le choix de ce composant s'est fait car nous l'avons déjà utilisé pour un projet (SAE IOT BUT3). Nous ajoutons alors un connecteur pour pouvoir ajouter aux besoins d'autres périphériques I2C.

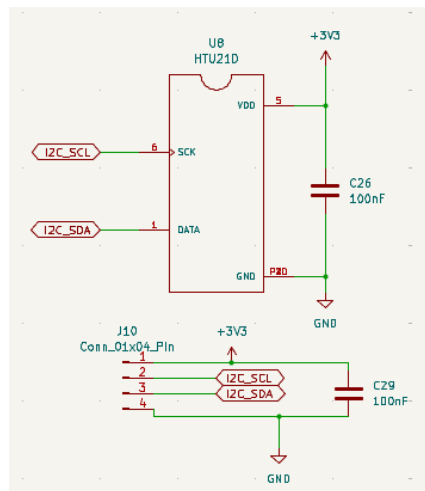


Figure 27: Schéma HTU21D & I2C

1.9.3 Capteur de courant/tension

Pour le capteur de tension et de courant, l'idée de base était d'utiliser un montage classique avec pont diviseur et un AOP¹³ pour mesurer la tension :

¹² Inter-Integrated Circuit

¹³ Amplificateur Opérationnel : composant électronique utilisé pour effectuer des opérations mathématiques sur des signaux analogiques, comme l'amplification, la comparaison ou la filtration

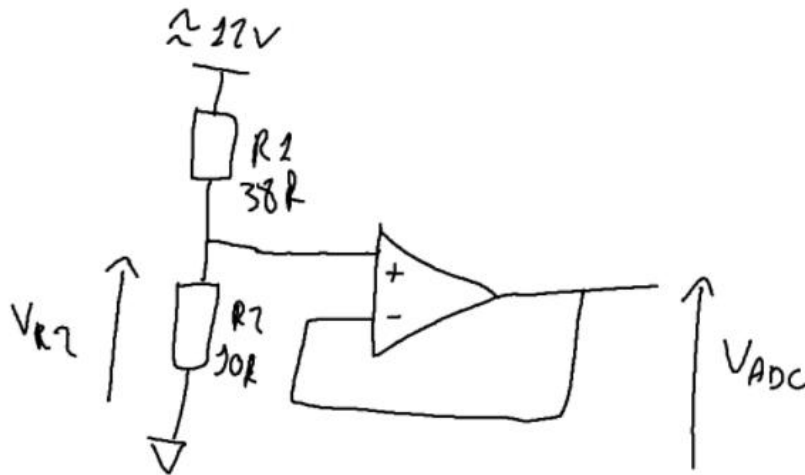


Figure 28: Schéma mesure de tension AOP

En épargnant la mesure des valeurs de résistances du pont diviseur, on obtient le schéma ci-dessus, ce montage permet d'avoir une image de la tension en lisant la valeur numérique convertie par l'ADC du μC , avec ces valeurs, on devrait avoir 2.5V qui arrive sur l'ADC (à 12V mesuré), cela permet d'avoir un delta relativement important pour la mesure de la tension.

Pour la mesure du courant, il faudrait utiliser une résistance de shunt (résistance de très faible valeur autour de 0.1Ω). Comme la résistance doit être de très faible valeur pour pas que la chute de tension ne soit trop importante, la valeur de la tension doit être amplifiée, ici au lieu d'utiliser un montage AOP suiveur, il faudrait utiliser un montage non inverseur pour amplifier la valeur lue sur l'ADC.

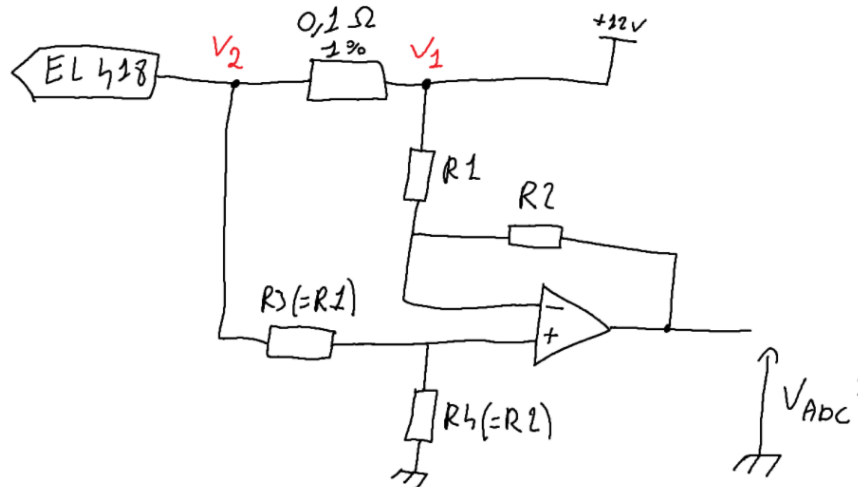


Figure 29: Schéma mesure de courant avec résistance de shunt et AOP en non inverseur

Alors la tension de sortie du montage $V'_{ADC} = \left(1 + \frac{R2}{R1}\right) (V2 - V1)$.

Sans besoin d'aller plus loin, ces montages sont relativement encombrants et demandent 2 GPIO libres. Le plus simple a été de trouver un circuit spécialisé qui comporte ces fonctionnalités pour permettre la mesure de courant et de tension, et idéalement, il faudrait que l'on puisse l'utiliser sur le bus I2C.

La solution a été l'**INA219**, qui permet la mesure de tension et de courant, le tout en le routant sur le bus I2C.

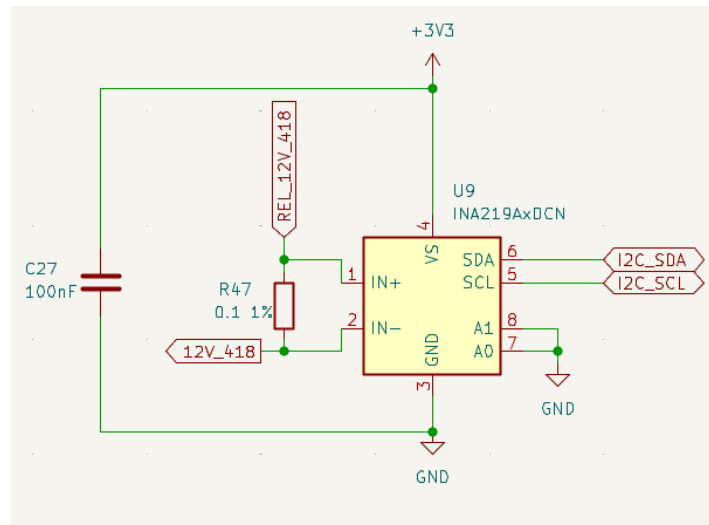


Figure 30: Schéma INA219

Pour l'utiliser, on connecte IN+ et IN- aux bornes de la résistance de shunt qui est en série entre l'arrivée d'alimentation de l'EL418 et le +12C, et on connecte simplement les pin SDA et SCL sur le bus I2C, les pins « A1 » et « A2 » sont des bits qui permettent de programmer l'adresse, ici on peut les tirer à la masse, ce qui donne comme adresse au périphérique « 1000000 ».

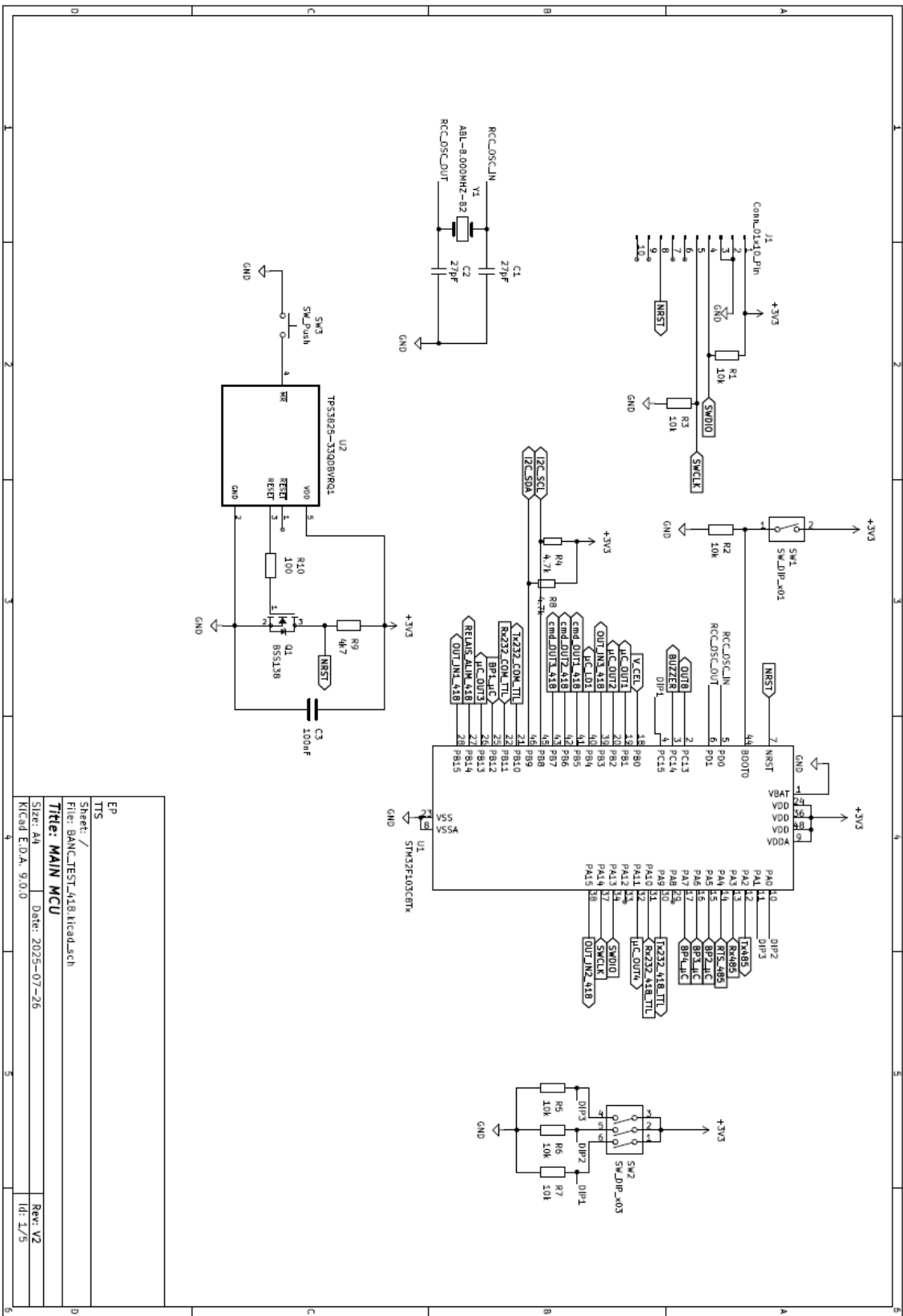
Table 1. INA219 Address Pins and Slave Addresses

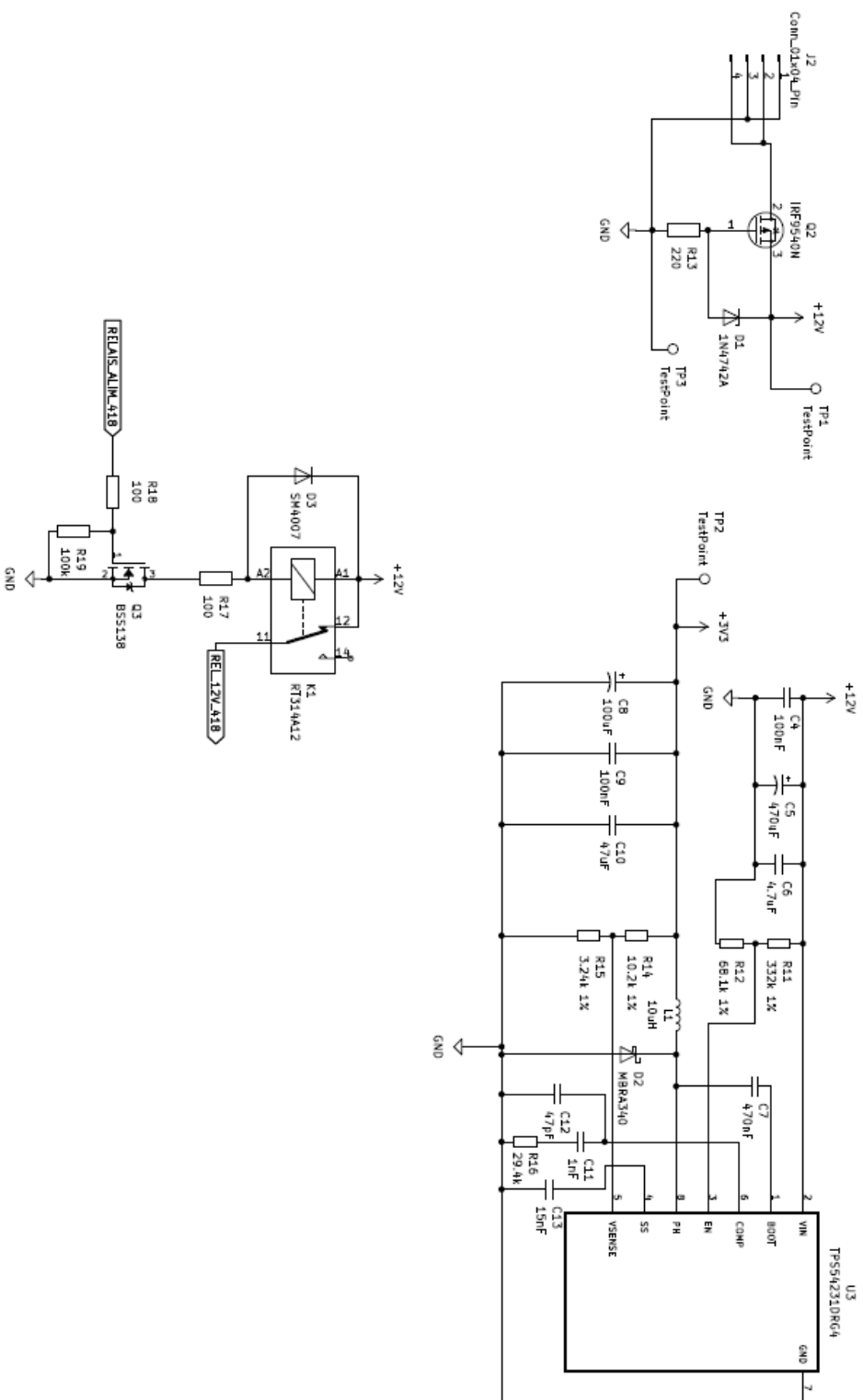
A1	A0	SLAVE ADDRESS
GND	GND	1000000
GND	V _{S+}	1000001
GND	SDA	1000010
GND	SCL	1000011
V _{S+}	GND	1000100
V _{S+}	V _{S+}	1000101
V _{S+}	SDA	1000110
V _{S+}	SCL	1000111
SDA	GND	1001000
SDA	V _{S+}	1001001
SDA	SDA	1001010
SDA	SCL	1001011
SCL	GND	1001100
SCL	V _{S+}	1001101
SCL	SDA	1001110
SCL	SCL	1001111

Figure 31: Tableau d'assignement des adresses de l'INA219

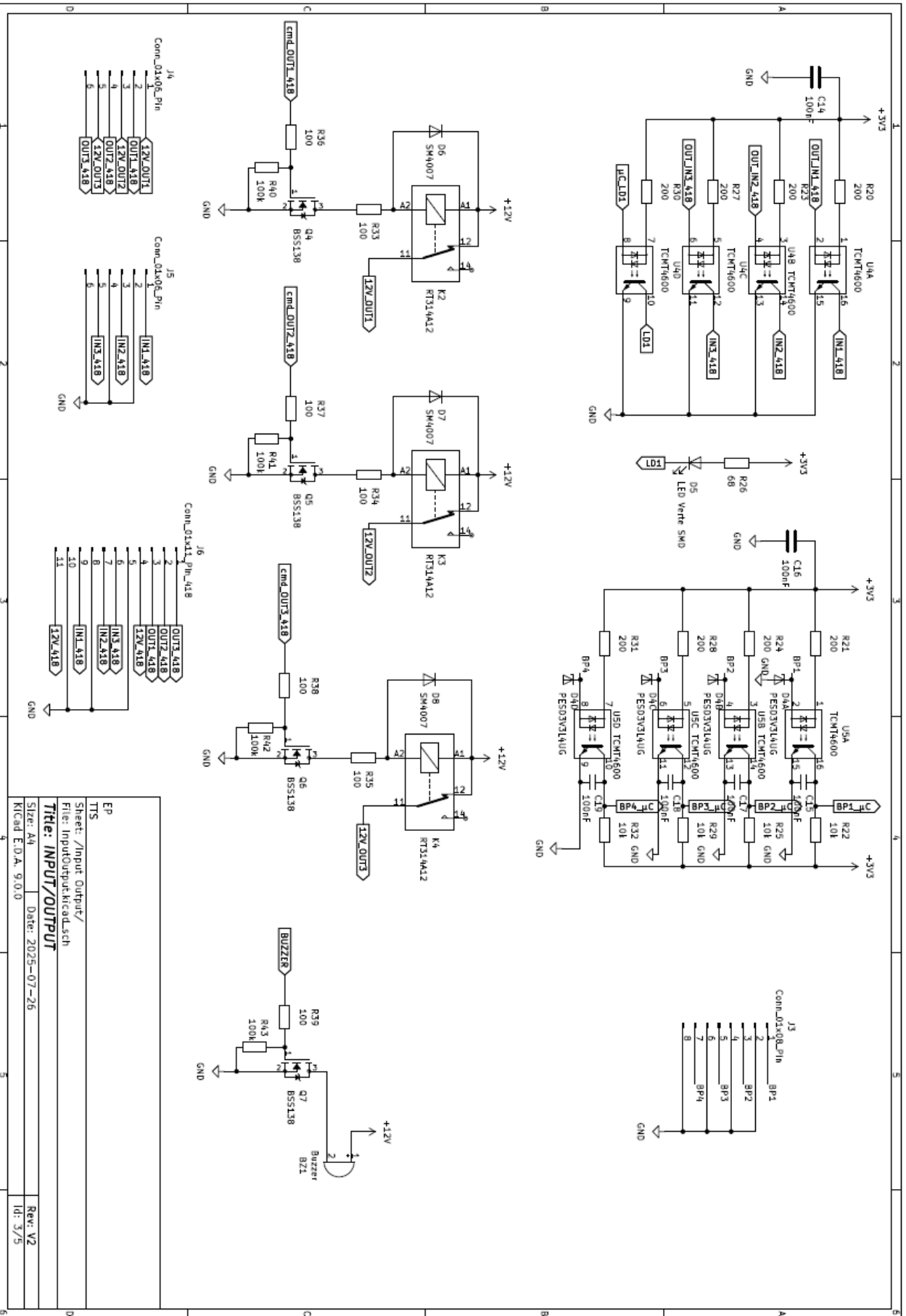
Il suffira alors de configurer le microcontrôleur pour qu'il envoie les requêtes appropriées à cette adresse afin d'obtenir en retour la tension et/ou le courant consommé par la carte EL418. À partir de ces valeurs, il sera possible de calculer la puissance consommée. Cette mesure en conditions réelles permettra d'estimer précisément la consommation de la carte, et ainsi de mieux dimensionner les batteries ou les autres sources d'alimentation.

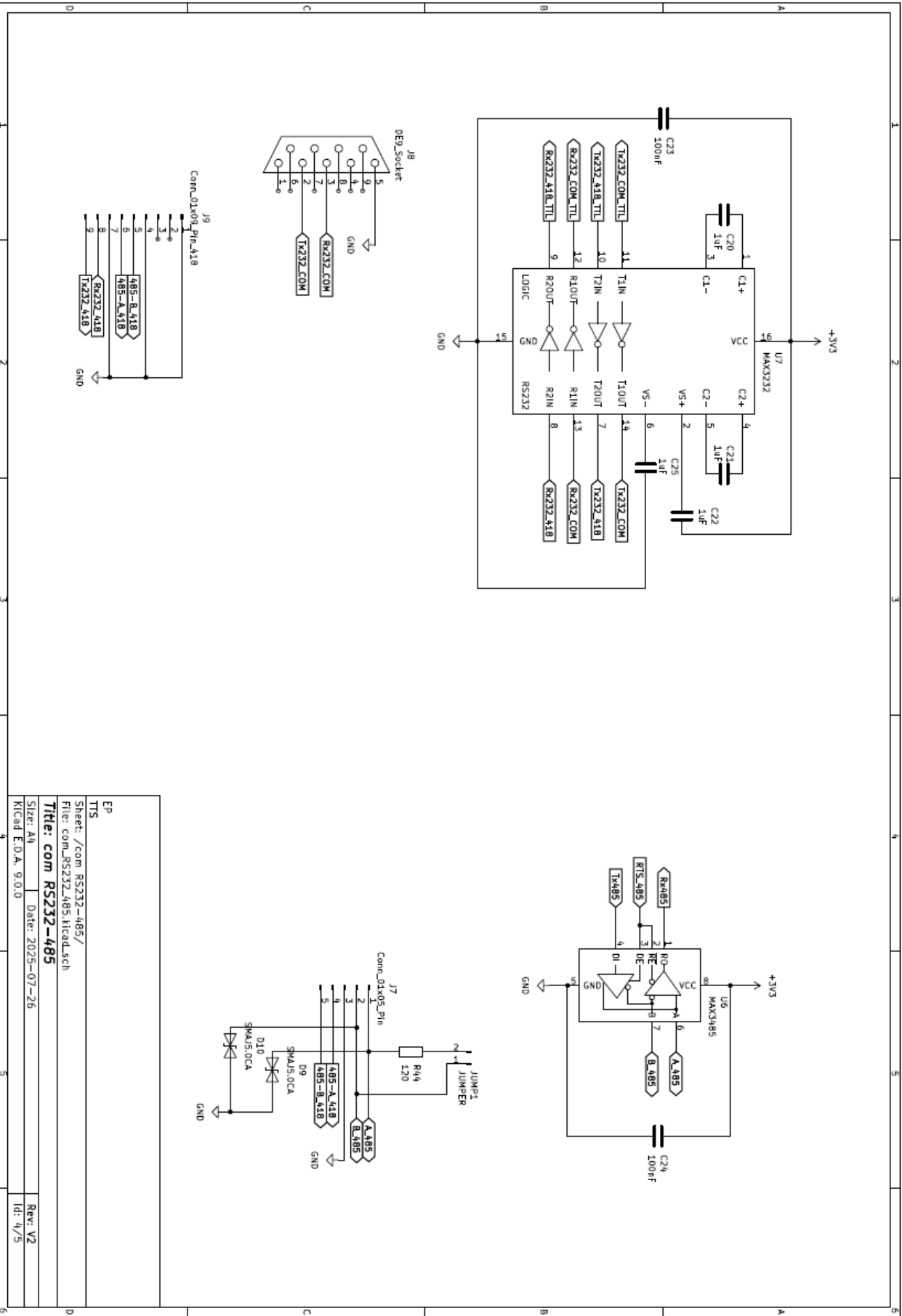
ANNEXES



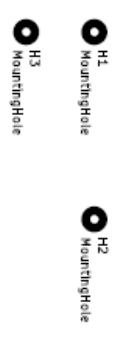
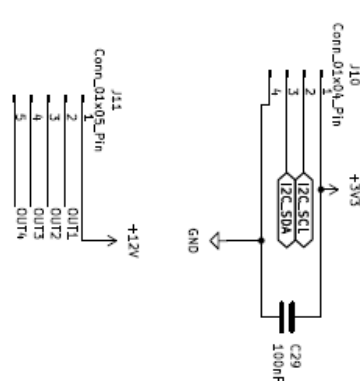
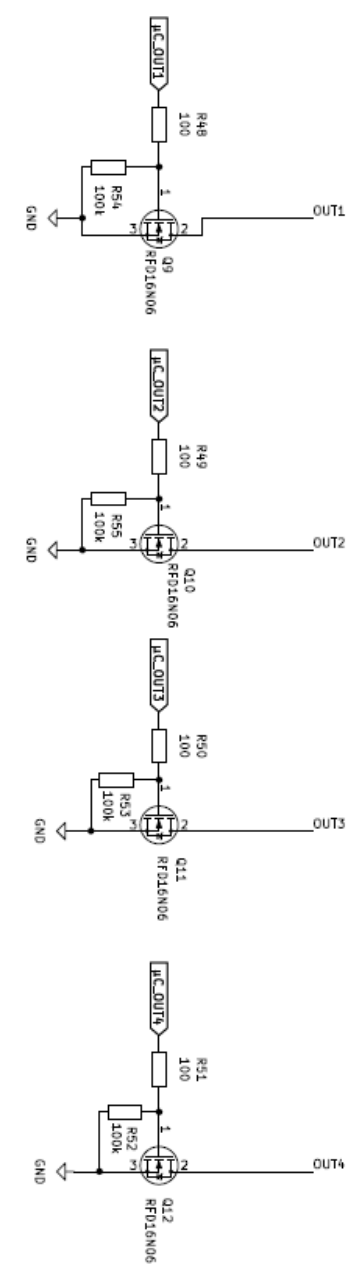
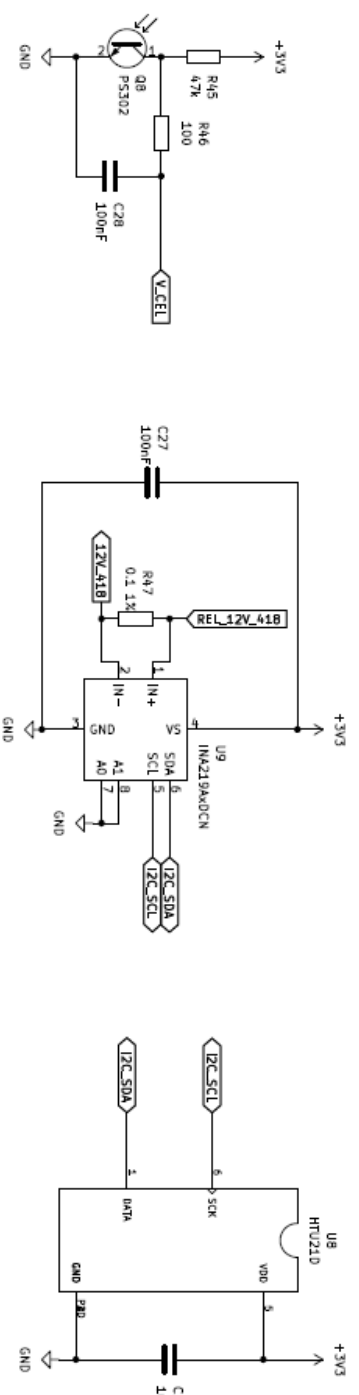


EP	
TT5	
Sheet: /Alim/	
File: untitled.kicad.sch	
Title: Alim	
Size: A4	Date: 2025-07-26
KiCad E D A. 9.0.0	Rev: V2
	Id: 2/5





EP	Sheet: /com RS232-485/
TTS	File: com_RS232-485.kicad.sch
	Title: com RS232-485
Size: A4	Date: 2025-07-26
Kicad E.D.A. 9.0.0	Rev: V2
	Id: 4/5



EP	
TTS	
Sheet: /Capteurs et autres sorties/	
File: capteurs.kicad.sch	
Title: Capteurs et autres sorties	
Size: A4	Date: 2025-07-26
KiCad E.D.A. 9.0.0	Rev: V2
	Id: 5/5