**Automatisation du banc de test BMS3 V3**

**Intérêt** :

Réaliser l’automatisation des tests de la carte BMS3 (produit de **Vaudoo Audio – CECIELEC** qui s’intègre dans des guitares basses électriques.

En effet, les tests post production, disponible sous :   
***\\CECICLOUD\Projets\Vaudoo-Audio\BMS3 - BMS plug'N'play\production\Procedures\_de\_test\  
210902-BMS3-Procedure\_de\_test.docx***,   
nécessitent entre 6 et 7 minutes par cartes, du fait d’un nombre de manipulation importante et de temps d’attente incompressible.

En automatisant ces tests, la durée des tests passe à environ une minute. De plus, un rapport de test, au format ***xlsx***, est généré automatiquement et met en avant les résultats de test négatifs.

**Prérequis et informations complémentaires** :

Afin de se soustraire des temps d’attente incompressible, dû aux temps d’affichage de l’état de la batterie et au temps d’attente suite au débranchement du jack, le firmware de la BMS3 intègre à présent un mode de test.

Afin d’activer le mode de test de la BMS3, avant sa mise sous tension, il faut mettre sous **+2V5** l’entrée **JMP\_18V** de la BMS3.

Ensuite l’entrées **JMP\_18V** permet l’activation des LED, lors d’un front descendant, et leur désactivation, lors d’un front montant. Les LED s’allument en suivant le cycle suivant :   
BLEU -> VERT -> ROUGE.

De plus, l’entrée **DEBUG\_RX** permet l’envoie sur la sortie **DEBUX\_TX** de la mesure de la tension de la batterie.

Le banc de test est contrôlé par l’application disponible sous :   
***\\CECICLOUD\Projets\Vaudoo-Audio\BMS3*** ***- BMS plug'N'play\Banc de test\Programme (.py etc..)\TestBench\***.

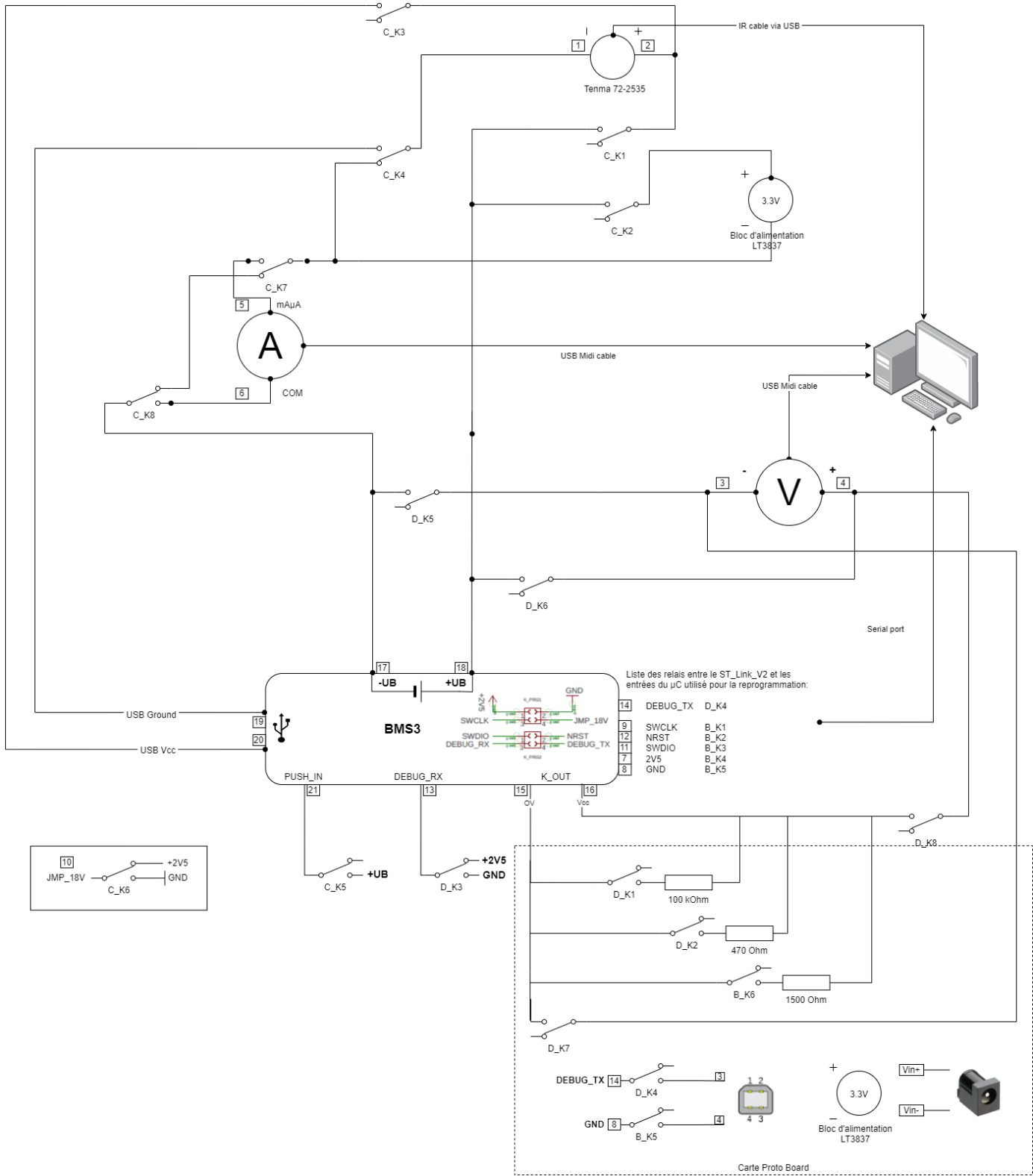
Sous Windows, il peut être lancé via le script **RunProject.bat**. Pour les autres systèmes d’exploitation, les lignes de commandes sont disponibles dans ce même script et doivent être adapté au terminal utilisé.

**Présentation du banc de tests :**

Le banc de test est composé :

* D’une alimentation **TENMA 72-2535**, qui sera piloté par l’application, celle-ci doit donc être **connecté au PC via un cordon USB Type B**.
* Du bloc d’alimentation LT3837, qui est utilisé pour la vérification de la charge de la batterie.
* D’une carte FT4232H Mini Module permettant de simuler 4 ports COM et sur laquelle sont reliées 3 cartes à relais ICSE014A (cf. Schéma du banc de test).
* De deux multimètres **TENMA 72-7730A**, l’un utilisé en voltmètre et l’autre en ampèremètre, chacun reliés au PC avec un câble USB **TENMA 72-13125** (cf. Schéma du banc de test pour connaître leur emplacement en fonction des relais permettant leur connexion/déconnection). La **fonction SEND** devra être **activé** en restant appuyer sur le bouton ayant la sous-fonction SEND.

Schéma du banc de test

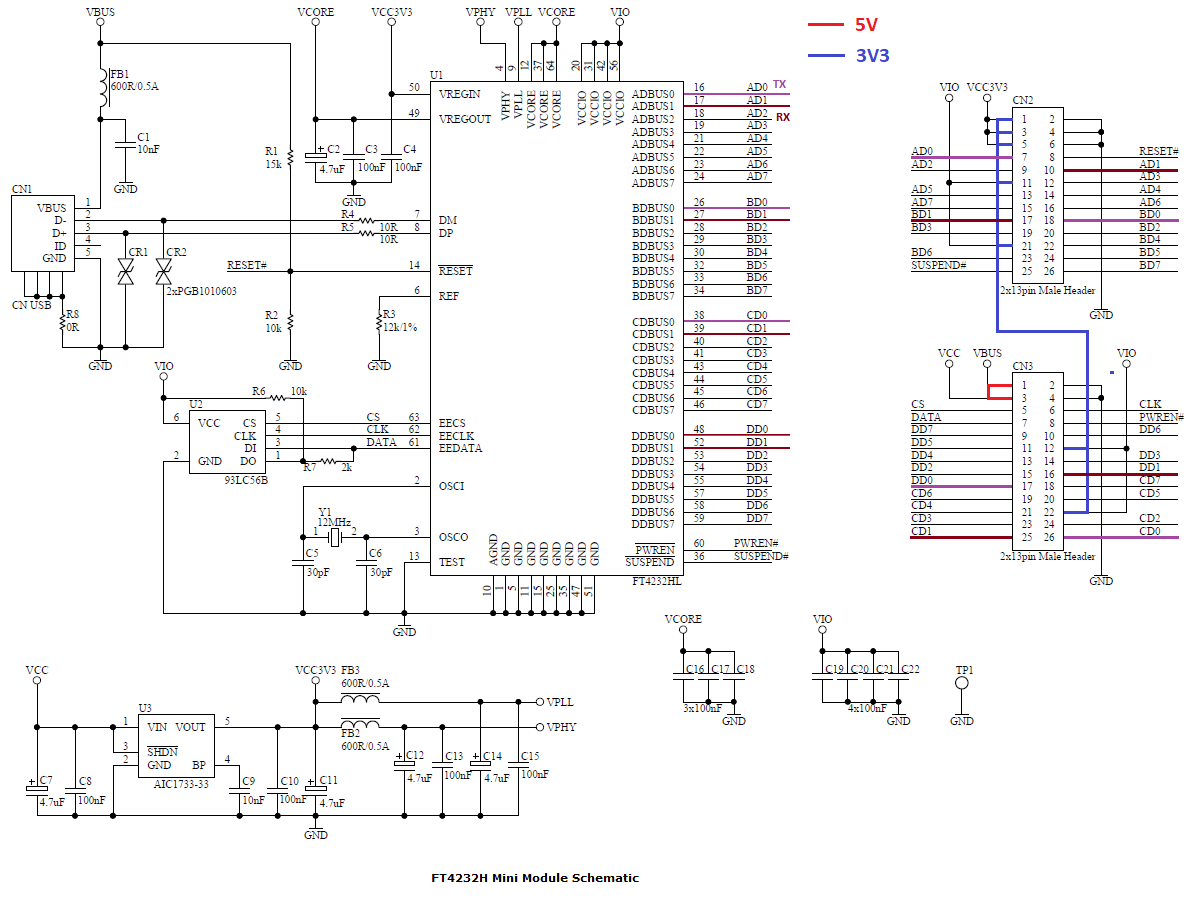


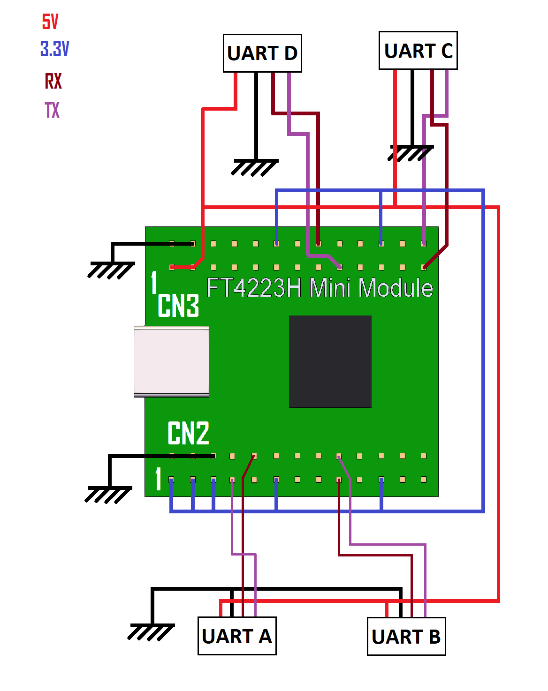
**Réalisation de la carte électronique intégrant la carte** **FT4232H Mini Module :**

Afin de pouvoir simuler sur le PC au moins 3 ports COM, une carte FT4232H Mini Module est configurée en ce sens. De plus, elle est intégrée sur une carte afin de pouvoir y interfacer les cartes à relais ICSE014A qui seront piloter via ses ports séries.

Pour une meilleure ergonomie et palier le nombre important de fils devant être connectés sur une plaque de prototypage, il est recommandé de réaliser une carte sur plaque à pastilles contenant le mini-module FT4232H et de le relié aux UART des relais (conformément aux schémas ci-dessous).

Remarque : Le mini-module FT4232H est alimenté par son port USB.





**Interfaçage matériel / logiciel :**

Tout le matériel se connecte par USB à l’ordinateur du banc de test CECIPROD.

Pour les multimètres et l'alimentation TENMA, la connexion se fait par périphérique HID.

Pour le FT4232H, nous utilisons un driver VCP[[1]](#footnote-1) (Virtual Com Port), pour permettre au système de détecter et différentier les UART, ainsi que le soft FT\_Prog[[2]](#footnote-2) qui permet de configurer le module.

Pour la communication avec les relais, la configuration de la communication est la suivante :

* baudrate : 9 600
* bytesize : EIGHTBITS
* stopbits : ONE
* parity : None

**Programmation du banc de test :**

Le langage choisi pour ce banc de test est Python 3.9. Il a été choisi de travailler en POO, avec des classes représentant chaque matériel et interface de communication.

Libraires utilisées :

* Pyserial : classe dont hérite notre classe Serial pour la communication série (UART)
* pyusb : Utilisation de la classe usb.core pour détecter et différencier les multimètres.  
   Utilisation de la classe usb.util pour configurer l'endPoint
* tenmaDcLib : Librairie fourni par TENMA afin de s'interfacer avec l'alimentation  
   TENMA 72-2535.
* XlsxWriter : Librairie utilisé pour la génération des rapports au format XLSX.

**Installation :**

Il faut premièrement connecter les trois modules de relais aux connecteurs UART de la carte selon le [schéma du mini-module FT4232H](#Schema_mini_module_FT4232H).

Il est nécessaire de connecter les deux TENMA par USB, selon le [schéma du banc de test](#Schema_banc_de_test), de les placer sur le bon calibre ainsi que de maintenir le bouton « SEND » appuyé pour configurer l’envoi infrarouge automatique.

Relier l’alimentation TENMA 72-2535 au banc de test.

Alimenter en 12V la carte LT3837 via le DC\_Jack.

Connecter le port USB Type B de la « carte proto board » au PC.

Connecter le programmateur ST-Link au PC, si vous voulez préalablement flasher la carte BMS3 avant de la tester.

**Utilisation :**

L'application commence par instancier l 'ensemble des classes permettant de s'interfacer avec le matériel et de le contrôler. Si la communication ne s'opère pas avec un appareil, un message d'erreur est affiché afin d'aider l'opérateur à la mise en place correcte du banc de test.

Ensuite l’application demande à l’opérateur s’il souhaite renommer le rapport de test (par défaut, en appuyant directement sur la touche ‘Entrée’, le rapport aura pour format de nom yyyy\_mm\_dd\_BMS3\_post\_prod\_test\_xxxx, avec :

yyyy : l’année en cours

mm : le mois en cours

dd : le jour du test

xxxx : le numéro du fichier entre 0000 et 9999

)

Puis l’application demande à l’opérateur de choisir le firmware à utiliser pour reprogrammer la BMS3. Ce firmware doit être dans le dossier situé dans le chemin relatif suivant :

\TestBench\bench\bms3\_interface\bms3\_firmwares

Si les BMS3 à tester ont été préalablement programmer, une option permet de ne pas effectuer cette étape (appuyer sur la touche n, puis valider).

Ensuite les interventions de l’opérateur sont les suivantes :

* Mettre en place la BMS3,
* Rentrer dans l'application son numéro de série,
* Valider la couleur des LEDS : le banc de test les actives une à une et l'opérateur doit indiquer si la couleur correspond bien à celle attendue.
* **Reste à faire :**

1. Modifier le [schéma du mini-module FT4232H](#Schema_mini_module_FT4232H) avec le nom des cartes relais (A, B, C, D).  
   => une fois que la carte aura été produite afin de pouvoir identifier à quel UART est associé chaque carte à relais.
2. Faire une photo du banc de test, une fois qu’il sera produit, et indiquer l’ensemble des éléments qui le compose pour faciliter [l’installation](#Installation).
3. Compléter le §[Installation](#Installation) avec la photo du banc et ajouter les étapes de la mis en place du test et du « branchement » des BMS3 à tester.

1. <http://www.ftdichip.com/Drivers/VCP.htm> [↑](#footnote-ref-1)
2. <http://www.ftdichip.com/Support/Documents/AppNotes/AN_124_User_Guide_For_FT_PROG.pdf>

   <http://www.ftdichip.com/Support/Utilities.htm#FT_PROG> [↑](#footnote-ref-2)