

Chadèche : un chargeur-déchargeur (très) intelligent pour accus NiMh

Jacques Ehrlich

Chadèche est l'acronyme de CHargeur-DECHargeur. Chadèche est à la fois un outil opérationnel pour l'entretien de vos batteries NiMh mais aussi un outil de recherche pour tester différentes lois de charge et de décharge en vue d'optimiser ou de régénérer leur performance. En effet Chadèche est hautement paramétrable et permet à l'utilisateur de définir n'importe quelle loi de charge et/ou de décharge.

Mots clés—batterie, accu, NiMh, AA, AAA, charge, décharge, Raspberry Pi, courant, constant.

I. INTRODUCTION

TOUT a commencé à cause d'une poignée d'accus NiMh 1,2 volts (format AA et AAA) qui traînaient dans un fond de tiroir. Je les utilisais pour mon GPS aéronautique et ne savais plus auxquels me fier : certains tenaient la charge quelques heures à peine, d'autres assuraient plus de 24 heures de bons et loyaux services. Pour y voir plus clair, je me lançais dans la réalisation d'un déchargeur à courant constant (100 mA) sur une simple carte Veroboard. Je ne savais pas alors que je mettais le doigt dans un engrenage qui allait me permettre de renouer avec mon métier d'origine (l'électronique et l'informatique) que j'avais quelque peu délaissé en fin de carrière et depuis le début de ma retraite. Et voilà me voilà pris de passion pour ces petits accus, explorant divers sites internet ainsi que les datasheets de composants spécialisés pour finir par concevoir d'A à Z (schéma, circuit imprimé, logiciel) un chargeur/déchargeur totalement paramétrable. Cet article se décompose donc en deux parties : la première est consacrée au matériel et la seconde au logiciel. Mais avant d'entrer dans le détail, il convient de résumer ici ce que fait Chadèche et plus particulièrement le logiciel qui le contrôle.

La principale fonction du logiciel Chadèche c'est d'exécuter un ou plusieurs cycles de charge/décharge. Un cycle se décompose lui-même en séquences chaque séquence a pour but d'imposer un courant de charge ou de décharge pendant une durée déterminée. Une séquence peut s'achever soit parce que la durée pour laquelle elle avait été programmée a été atteinte soit parce qu'une condition particulière est satisfaite (par exemple, la tension aux bornes de l'accu a atteint un certain seuil). Le nombre de séquences, le courant imposé, leur durée, les actions à mener quand certaines conditions sont satisfaites sont décrites dans un fichier de configuration.

De plus, à intervalle de temps régulier, le logiciel enregistre

dans un fichier texte au format CSV, toutes les informations utiles pour une exploitation en temps différé. Ainsi, à la fin d'un essai on peut récupérer ce fichier et tracer les caractéristiques de charge ou de décharge en se servant par exemple des possibilités graphiques d'Excel, de LibreOffice, GnuPlot ou tout autre logiciel d'exploitation de données. La Figure 1 est un exemple montrant la caractéristique de charge (charge par rafales) puis de décharge que l'on peut obtenir avec le fichier de configuration décrit plus loin dans cet article.

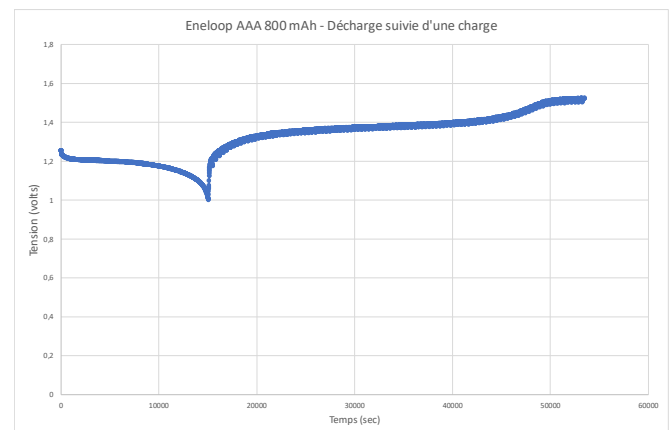


Figure 1 – Exemple d'exploitation de résultats.

Il n'y a pour ainsi dire aucune limite pour décrire des cycles de charge/décharge si ce n'est celle de votre imagination. : ni sur le nombre séquences et leur contenu, ni sur le nombre de cycles.

II. CHADECHE : LE MATERIEL

A. Architecture générale

Chadèche repose sur une conception modulaire permettant de gérer de façon totalement indépendante jusqu'à quatre accus. Chaque accu est sous contrôle d'une carte Chadèche lesquelles sont pilotées par une carte Raspberry Pi (RPi). Rien n'impose de réaliser les quatre cartes Chadèche, mais il en faut une au minimum et dans ce cas vous ne pourrez charger/décharger qu'un accu à la fois. Cette conception modulaire à l'avantage de vous permettre d'adapter votre Chadèche à votre besoin et à votre budget.

La Figure 2 donne un aperçu de cette modularité. Chaque carte possède une adresse (configurée à l'aide de switches). L'ensemble des cartes est connecté à la RPi à l'aide d'un câble

en nappe 40 fils mais seuls quelques signaux parmi les 40 sont utilisés pour l'adressage et l'échange de données. Les circuits imprimés ont été conçus pour être empilables avec des côtes de perçage compatibles avec celles de la RPi.

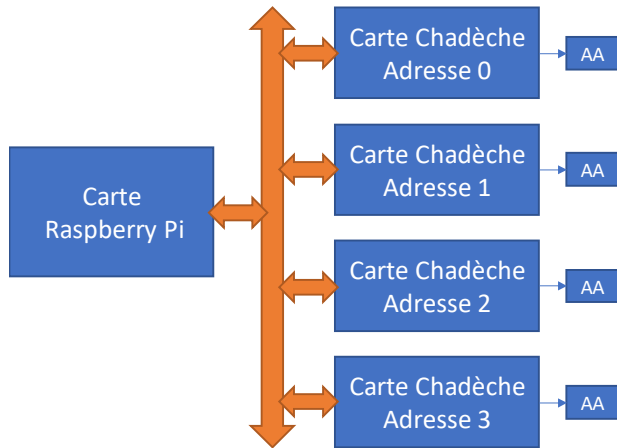


Figure 2 – Architecture générale de Chadèche

B. Schéma fonctionnel

La Figure 3 permet de comprendre le fonctionnement d'une carte Chadèche. Au cœur de Chadèche se trouve un générateur à courant constant commandé en tension qui sera utilisé aussi bien pour charger que pour décharger l'accu. La commande en tension de ce générateur de courant est assurée par un convertisseur numérique analogique (CNA) piloté par la RPi via l'interface SPI. Que ce soit pendant la charge ou la décharge, on mesure l'évolution de la tension aux bornes de l'accu à l'aide d'un convertisseur analogique numérique (CAN) lui aussi contrôlé par la RPi via son interface SPI. Le passage de charge à décharge s'obtient simplement en inversant la connexion de l'accu aux bornes du générateur à courant constant. L'inversion de polarité est obtenue grâce à un relais 2RT (deux repos-travail) piloté par la RPi. Il reste à ajouter un peu de « glue » logique pour le décodage d'adresse de chaque carte et la sélection des différents circuits (CAN, CNA et commande du relais inverseur de polarités.) Même si comme on le verra dans la deuxième partie de cet article, le logiciel permet de superviser le fonctionnement de Chadèche, il est tout de même commode de disposer d'une supervision élémentaire à l'aide de quelques leds pour connaître l'état de chaque carte Chadèche (non représentées sur le diagramme fonctionnel de la Figure 3).

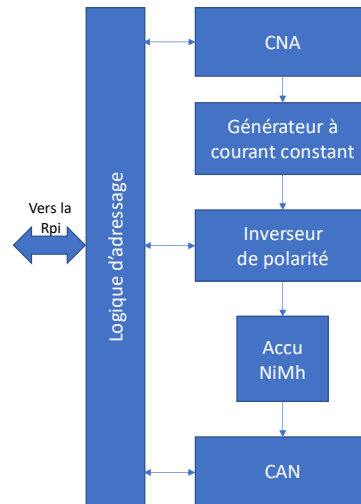


Figure 3 – Diagramme fonctionnel de Chadèche

Celles-ci sont au nombre de trois : la première indique simplement la présence de l'alimentation, la seconde est une led bicolore qui nous informe de l'état de charge (rouge) ou de décharge (vert) et la troisième est une led clignotante dont le rapport cyclique est commandé par la deuxième voie du CNA permettant ainsi de « coder » visuellement avec souplesse, le déroulement d'un essai.

C. Principe du circuit (Figure 5)

1) Alimentation

La carte est alimentée en 12 volts (VDD), tension utilisée principalement pour alimenter le générateur à courant constant, le relais inverseur de polarités ainsi qu'une référence de tension 4,096 volts (VREF) pour le convertisseur analogique numérique et le convertisseur numérique analogique. A partir du VDD, un régulateur 78L05 délivre une tension stabilisée 5 volts (VCC) qui alimente les circuits logiques (série 74HC) ainsi que le CNA MCP4022 et le CAN MCP 3201. Ces alimentations sont filtrées par toute une série de condensateurs électrolytiques (100 mF) ou céramiques (100 nF). Le bilan en courant sous 5 volts est inférieur à 100 mA, courant max délivré par un régulateur 78L05.

2) Logique d'interface avec la RPi

Un connecteur 40 points dont le brochage est conforme à celui d'une RPi A+ constitue le point d'entrée de cette interface. Puisqu'il est possible d'adresser jusqu'à 4 cartes Chadèche, deux fils d'adresses sont nécessaires A0, A1 reliés respectivement aux signaux GPIO17 et GPIO27 de la RPi.

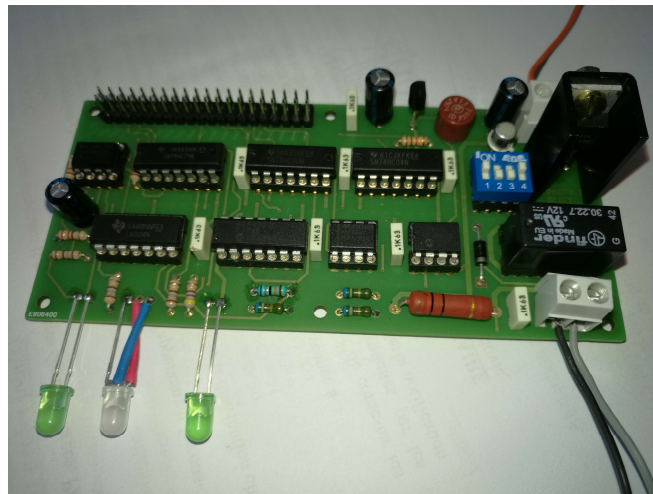


Figure 4 – Vue d'une carte Chadèche

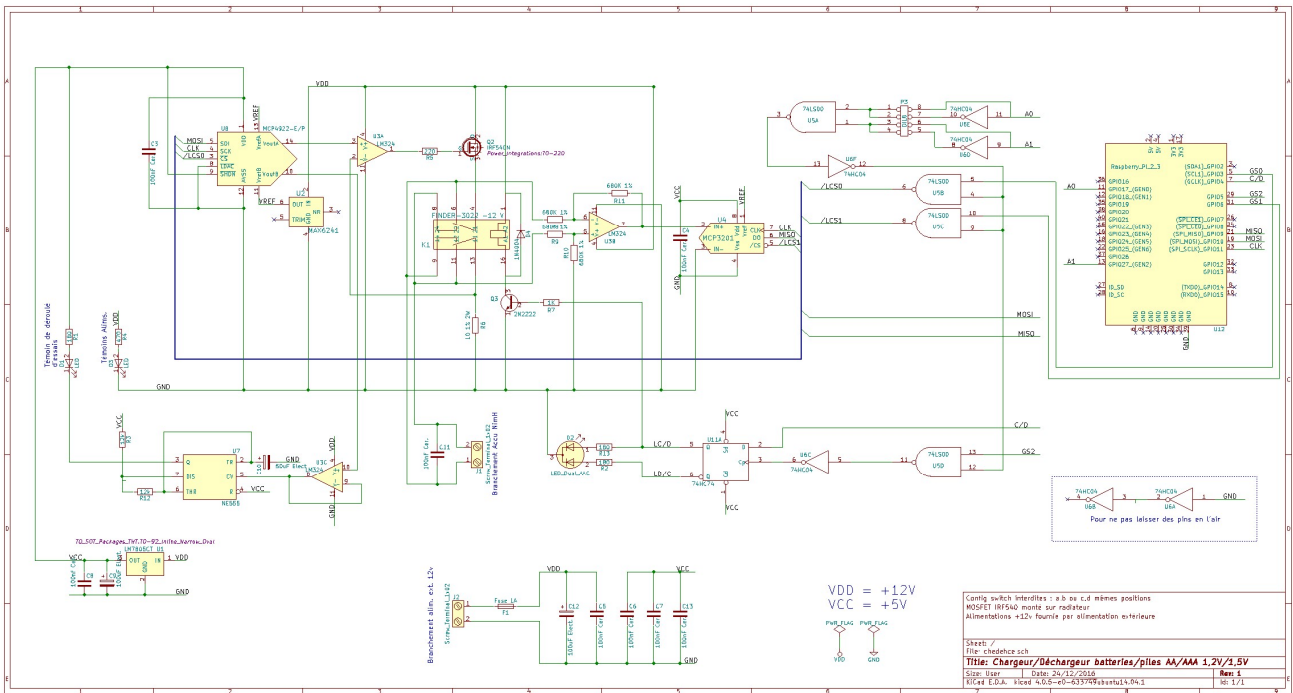


Figure 5 – Schéma de principe

La logique de décodage d’adresse est obtenue par les inverseurs U6D, U6E, U6F, la porte NAND U5A ainsi que les 4 switches P3. En sortie de l’inverseur U6F on trouve un signal actif niveau 1 si la carte est sélectionnée. Les signaux CS0, CS1 et CS2 reliés respectivement aux signaux GPIO3, GPIO6 et GPIO5 de la RPi permettent de sélectionner à travers les portes NAND U5B, U5C, U5D respectivement le CNA, le CAN et le latch de commande du relais de charge/décharge. Ce dernier est une simple bascule D (74HC74) dont les sorties sont utilisées d’une part pour actionner le relais 2RT via le transistor Q3 et d’autre part pour allumer la led bicolore en rouge ou en vert selon qu’on est en charge ou en décharge. Pour terminer sur l’interface vers la RPi, il convient de mentionner les signaux MISO, MOSI et SCLK qui permettent de contrôler les CNA et CAN par leur interface SPI. A noter que je n’ai pas utilisé les signaux /SPL_CE01 ou /SPL_CE1 de l’interface SPI mais les signaux GPIO3 (CS0) et GPIO5 (CS1) car ces derniers étaient plus faciles à router au moment de la réalisation du circuit imprimé. Le « coût » de cette décision se traduit par quelques lignes code supplémentaires au niveau logiciel.

3) L’inverseur de polarité

Comme déjà indiqué, il s’agit d’un simple relais Finder 2RT alimenté en 12 volts. Celui-ci connecte l’accu en série avec un MOSFET (Q2) de puissance (monté sur dissipateur) qui opère en générateur de courant. Selon la position du relais, c’est la borne + ou la borne - de l’accu qui est reliée à la source de ce transistor dont le drain est connecté à la tension 12 volts, l’autre borne de l’accu étant reliée à une résistance de pied de 10 ohms.

Attention !!!

Des températures élevées ont été mesurées sur le dissipateur du transistor MOSFET de puissance. Ne pas toucher en cours d’essais. Si la carte est intégrée dans un boîtier, celui-ci doit être ventilé.

4) Le générateur de courant commandé en tension

Il s’agit d’un montage relativement classique que l’on trouve dans de nombreuses notes d’application d’amplificateurs opérationnels et dont le schéma de principe est décrit dans la Figure 6.

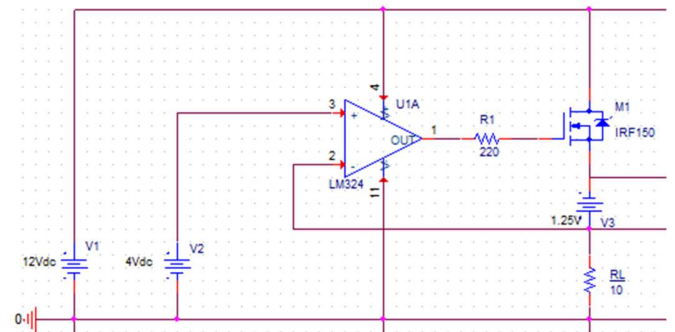


Figure 6 – Schéma de principe du générateur de courant

L’accu V3 se situe entre la source du MOSFET et la résistance de 10 ohms. On démontre assez facilement que le courant imposé par le générateur a pour valeur $V2/RL$ (dans l’exemple de la Figure 6, 400 mA). La précision de ce courant est donc dépendante de celle sur la tension V2 et sur la résistance RL (qui doit pouvoir dissiper une puissance de 2 W). Mais comme dans le schéma réel, la tension V2 est délivrée par le CNA

MCP4022 (U8), j’ai opté pour une résistance RL de précision ordinaire, sachant que je pourrai obtenir la précision requise sur le courant grâce à des coefficients de mise à l’échelle qui seront

introduits dans le logiciel à la suite d'une procédure d'étalonnage.

On notera que la tension aux bornes de l'accu V3 n'est pas référencée par rapport à la masse à cause de la résistance de pied RL. De ce fait elle ne peut pas être directement appliquée à l'entrée du CAN MCP3201 (U4). Ainsi, le rôle de l'amplificateur différentiel à gain unitaire U3B est de délivrer à l'entrée du CAN une copie de la tension mesurée sur l'accu, mais cette fois-ci référencée par rapport à la masse.

5) Convertisseurs numérique-analogique et numérique analogiques

J'ai opté pour des convertisseurs 12 bits. Le CNA MCP4022 possède deux voies. La première est utilisée pour commander le générateur à courant constant. La seconde contrôle le rapport cyclique d'un timer NE555 dont le signal de sortie allume ou éteint une led permettant de superviser le déroulement d'un essai. Le CAN MCP3201 numérise la tension de l'accu NiMh disponible en sortie de l'amplificateur différentiel à gain unitaire déjà cité plus haut. Ces deux convertisseurs reçoivent une tension de référence 4,096 volts délivrée par un circuit MAX6241 (U2). Ainsi pour le CNA, un incrément/décroissement d'un bit correspond à un accroissement/abaissement de la tension sur la sortie sélectionnée de 1mV. Similairement, pour le CAN, un accroissement/abaissement de la tension d'entrée de 1mV se traduit numériquement par un incrément/décroissement de 1 bit.

D. Réalisation

Pas de difficulté particulière pour la réalisation des cartes Chadèche sur le circuit imprimé double face proposé (Figure 7 et Figure 8) conçu pour éviter tout CMS. La largeur de la carte correspond exactement à celle d'une Raspberry Pi A+.

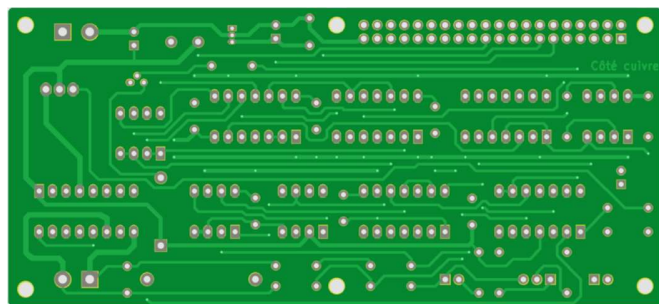


Figure 7 – Circuit imprimé côté cuivre

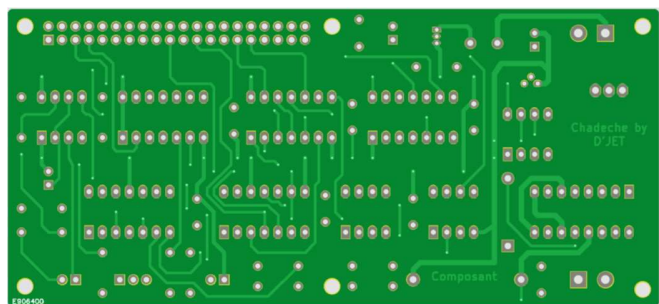


Figure 8 – Circuit imprimé côté composant

Sur la vue côté composant (Figure 8) on observera que les entraxes entre perçages de gauche et perçages médians ainsi que

le positionnement du connecteur 40 points permettent un empilage aisé avec la Raspberry Model A+ (voir Figure 13 en annexe).

La liaison entre la RPi et les cartes Chadèche est assurée par un câble en nappe 40 fils.

La liste des composants pour la réalisation d'une carte associée à la RPi est donnée en annexe (Tableau 2). Les alimentations ne sont pas incluses et il faudra prévoir une alimentation 12 volts/2A pour quatre cartes Chadèche ainsi qu'une alimentation 5 volts pour la RPi Model A+ dont la consommation n'exécède pas 700 mA.

Pour maximiser la précision dans la mesure de la tension aux bornes de l'accu, il est conseillé d'utiliser du câble de section 2,5mm² pour assurer la liaison entre le coupleur d'accu (AA ou AAA) et la carte Chadèche et ainsi minimiser la chute de tension due à la résistance du câble.

E. Choix de Raspberry Pi

Une étude comparative des rapport performances/prix des différents modèles de RPi m'a montré que le modèle Zéro aurait largement convenu à mes besoins. Mais face à la difficulté d'approvisionnement de cette carte, j'ai finalement opté pour une Raspberry Pi Modèle A+ à laquelle j'ai ajouté un *dongle* USB Wifi pour permettre l'indispensable connexion à l'internet ainsi qu'une carte micro SD de 16 Go. Evidemment, qui peut le plus peut le moins et tout modèle de RPi de performance supérieure conviendra tout aussi bien sauf les modèles dotés d'un connecteur 26 points. Il conviendra toutefois de vérifier le bon positionnement des trous de fixation si vous souhaitez bénéficier de la possibilité d'empiler les cartes.

F. Sélection d'adresse des cartes Chadèche

Il ne faudra pas oublier de fixer l'adresse des cartes Chadèche à l'aide de l'interrupteur dual in line 4 positions. Le Tableau 1 donne la position des switches en fonction de l'adresse souhaitée :

ADR	SW1	SW2	SW3	SW4
0	OFF	ON	OFF	ON
1	ON	OFF	OFF	ON
2	OFF	ON	ON	OFF
3	ON	OFF	ON	OFF

Tableau 1 – Position des switches en fonction de l'adresse

III. CHADECHE : LE LOGICIEL

A. Système d'exploitation, installation, configuration

Le logiciel Chadèche s'exécute sous la version Raspbian Lite de Linux qu'il conviendra avant tout autre chose d'installer en s'aidant si nécessaire des nombreux tutoriels disponibles à cet effet. On veillera à activer l'application SSH serveur si elle ne l'est pas par défaut, car c'est elle qui permettra de dialoguer via l'internet avec la RPi depuis un PC doté d'une application SSH cliente. Quelques fichiers de configuration devront être adaptés pour que le dongle Wifi que vous aurez choisi soit pris en compte car ce sera désormais le Wifi qui sera le support d'échange entre le PC client et Chadèche puisque la RPi Model A+ ne possède pas de port Ethernet. A l'aide d'une commande

ipconfig on affichera et on notera l'adresse IP du dongle Wifi car elle sera nécessaire pour les futures connexions via SSH. Enfin, je conseille aussi d'installer Git qui sera très utile pour l'installation et les mises à jour du logiciel Chadèche et des petits utilitaires associés.

B. Configuration du PC client et connexion à Chadèche.

Côté PC, il faut installer un client SSH. Sous Linux ou MacOS, il n'y a rien à faire, celui-ci étant installé par défaut. Il suffit d'ouvrir une fenêtre d'interpréteur de commande (nous l'appellerons *shell* dans la suite de l'article) pour y avoir accès. Sous Windows je préconise d'installer et d'utiliser Putty (<http://www.putty.org/>).

Après lancement de Putty, on introduit l'adresse IP de la RPi dans le champ *Host Name* (Figure 9), puis on clique sur *Open*, ce qui affiche un *shell* permettant de réaliser la connexion à la RPi.

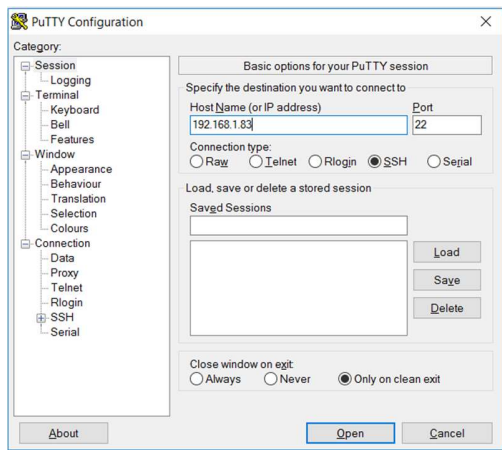


Figure 9 – Ecran d'accueil de Putty

Après la première connexion on se trouve sur son *home directory* qui est vide puisque l'application Chadèche et ses utilitaires ne sont pas encore installés. Chacun est libre d'organiser ses répertoires comme il le souhaite. Lorsque vous « clonez » les fichiers Chadèche disponibles sur Github, un répertoire *chadeche* sera automatiquement créé.

C. Librairie et langage de programmation

Avant toute utilisation du logiciel Chadèche, il convient d'installer la librairie WiringPi dont les fichiers et instructions d'installation sont disponibles sur le site de son concepteur (<http://wiringpi.com/>).

On peut ensuite installer l'application Chadèche et les programmes de test qui l'accompagnent. Ceux-ci sont écrits en langage C. Le compilateur *gcc* n'est requis que si vous souhaitez recompiler les sources mais il est installé par défaut avec Raspbian Lite. Les sources et exécutables de Chadèche sont disponibles en téléchargement sur Github. La procédure d'installation est très simple (Git est supposé installé selon la recommandation faite au § A) :

1. On se place dans le répertoire où vous souhaitez faire l'installation,
2. On tape les commandes suivantes :

```
$ git init
$ git remote add origin
https://github.com/duodiscus92/chadeche.git
$ git pull origin master
```

Après installation plusieurs répertoires ont été créés :

- *chadeche* contient les programmes source, l'exécutable de l'application Chadèche, des fichiers des paramètres que vous pouvez adapter et les 3 répertoires suivants,
- *tests* contient un multitude de petits utilitaires pour le test (livrés en l'état ...),
- *config* est destiné à recevoir tous vos fichiers de configuration,
- *resultats* est destiné à recevoir tous vos fichiers de résultats.
- *calibration* est destiné à recevoir vos fichiers de résultats des procédures de calibration

D. Utilisation

1) Le fichier de configuration de Chadèche

Il importe de bien comprendre la configuration de Chadèche car c'est elle qui donne toute la puissance à ce logiciel et qui offre comme on l'a annoncé au début de cet article des possibilités infinies de lois de charge/décharge d'accus. La configuration est inscrite dans un fichier texte au format CSV dans lequel chaque champ est séparé par une tabulation. Mise à part la 1^{ère} ligne qui est une ligne de commentaire, les autres lignes décrivent une *séquence*. Un fichier contient donc une suite de séquences, l'ensemble des séquences décrit un *cycle*. Un cycle peut être répété autant de fois que souhaité selon l'indication d'un paramètre de lancement du programme Chadèche.

Au début, la façon la plus simple de préparer un fichier de configuration sera d'utiliser un tableur (Excel ou LibreOffice) puis de sauver le fichier au format CSV (en choisissant la tabulation comme séparateur de champ) et de l'envoyer sur la RPi de Chadèche en utilisant le logiciel Filezilla par exemple. Mais si vous êtes aguerris avec les éditeurs de texte vous pourrez composer vos fichiers de configuration directement sur la RPi avec l'éditeur de texte de votre choix (j'utilise *nano*).

La Figure 10 est un exemple de fichier de configuration. Comme déjà indiqué la première ligne est une ligne de commentaire contenant l'intitulé des colonnes. Bien qu'ignorée par le logiciel Chadèche, sa présence est néanmoins requise. En l'omettant, ce sera la 1^{ère} séquence qui serait alors ignorée.

Les lignes suivantes sont des séquences, il y en a 9 au total dans cet exemple, pour former un cycle. Bien entendu, il n'existe pas de limite sur le nombre de séquences décrivant un cycle.

ADR	C/D	mA	SD	TL	TH	LmAh	HmAh	TRUE	CTRLZ	COMMENT
1	D	0	60	I	I	I	I	I	S	#Repos
2	D	200	90000	S	I	S	I	I	S	#Decharge 200 mAh
3	D	0	30	I	I	I	I	I	S	#Repos
4	C	300	300	I	6	I	6	I	S	#Charge 300 jusqu'a capacite max
5	C	0	5	I	I	I	I	3	S	#NOP et saut incondtionnel
6	C	50	300	I	I	I	I	I	S	#Topoff
7	C	10	3600	I	I	I	I	I	I	#Maintient
8	C	0	0	I	I	I	I	20:00	S	#Reveil à 20 heures
9	C	0	0	I	I	I	I	I	S	#Fin de cycle

Figure 10 – Exemple de fichier de configuration composé sous Excel

Examinons champ par champ leur signification :

- ADR : il s'agit d'un champ adresse qui pourra être utilisé dans des « GOTO ». Les adresses sont des nombres entiers quelconques mais toujours croissants ;
- C/D : un C signifie qu'il s'agit d'une séquence de charge et un D indique une séquence de décharge ;
- mA : un nombre entier indiquant la consigne de courant (de charge ou de décharge) et ne pouvant dépasser 400 mA ;
- SD (*sequence duration*) : un nombre entier indiquant la durée maximale de la séquence en secondes. Mais comme on le verra plus loin cette durée peut être raccourcie si une condition particulière est satisfaite.

Les champs suivants (COMMENT exclu) sont interprétés de la même manière comme nous allons le voir ci-dessous. Chaque champ représente une condition qui si elle est VRAIE déclenche une action qui dépend de la lettre se trouvant dans le champ correspondant :

- I (ignorer) : rien ne se passe ;
- S (séquence) : on interrompt la séquence en cours et on passe à la séquence suivante ;
- C (cycle) : on interrompt la séquence en cours et on lance le cycle suivant (reprise à la séquence n°1) ;
- N (*nombre entier*) : on interrompt la séquence en cours et on se branche sur la ligne d'adresse N. C'est un GOTO conditionnel ;
- HH:MM : on interrompt la séquence en cours et on s'endort jusqu'à l'heure indiquée. Au réveil on commence la séquence suivante. Pendant l'endormissement le courant (de charge ou décharge) est à zéro.
- +HH:MM : on interrompt la séquence en cours et on s'endort pour la durée indiquée en heures et minutes (*non implémenté à l'heure où nous écrivons ces lignes*).

Les conditions sont les suivantes :

- TL (seuil de tension bas) : cette condition est vraie quand la tension mesurée aux bornes de l'accu est inférieure ou égale à un seuil. La valeur de ce seuil est fixée au lancement du programme ;
- TH (seuil de tension haut) : cette condition est vraie quand la tension mesurée aux bornes de l'accu est supérieure ou égale à un seuil. La valeur de ce seuil est fixée au lancement du programme ;
- LmAh (seuil de capacité bas) : cette condition est vraie quand la capacité (quantité de charge) estimée de l'accu est inférieure ou égale à un seuil. La valeur de ce seuil est fixée au lancement du programme ;
- HmAh (seuil de capacité haut) : cette condition est vraie quand la capacité (quantité de charge) estimée de l'accu est supérieure ou égale à un seuil. La valeur de ce seuil est fixée au lancement du programme ;
- TRUE (vrai) : cette condition est toujours vraie. Utilisée essentiellement pour faire des GOTO inconditionnels.
- CTRLZ (control-Z) : cette condition est vraie quand on presse simultanément les touches CTRL et Z.

Les conditions sont évaluées de gauche à droite. Dès qu'une condition vraie est rencontrée, l'action programmée est exécutée et les autres conditions sont ignorées sauf si l'action programmée est I (ne rien faire), dans ce cas la condition suivante est évaluée.

Le dernier champ est un champ commentaire : il doit obligatoirement commencer par le caractère #

Notez bien que dans la version actuelle du programme Chadèche, aucune vérification de syntaxe ou de sémantique n'est effectuée sur le fichier de configuration.

Notez également qu'il est toujours possible de suspendre un essai en tapant CTRL-C. Il est alors proposé soit de reprendre ou de terminer l'essai.

2) A chaque carte Chadèche son programme Chadèche

Comme décrit dans la première partie de cet article, un Chadèche peut être posséder jusqu'à quatre cartes permettant de réaliser simultanément quatre essais sur quatre accus de façon tout à fait indépendants. Pour cela le logiciel Chadèche exploite les capacités multitâches du système Linux. Chaque commande Chadèche pourra être lancée dans un *shell* différent. C'est l'adresse de la carte, précisée au lancement de la commande, qui permet de faire la différence.

3) Connexion à Chadèche

Depuis un PC (peu importe son système d'exploitation pourvu qu'il possède un SSH client) on ouvre un *shell* et on se connecte à la Raspberry Pi par SSH comme indiqué au § B puis on se place dans le répertoire `chadeche`.

Si vous souhaitez lancer plusieurs Chadèche pour réaliser simultanément plusieurs tests (quatre au maximum), il faudra réaliser autant de connexion SSH dans des *shell* différents. Putty permet de réaliser cela en faisant un clic droit sur la fenêtre *shell* puis en sélectionnant « Duplicate session ».

4) Les paramètres de Chadèche

Chadèche possède de nombreux paramètres. Ceux-ci peuvent être initialisés à partir de trois sources :

- Les valeurs par défaut inscrites en « dur » dans le code source du programme ;
- Les valeurs contenues dans les fichiers de configuration dont les noms sont `chadeche0.prm`, `chadeche1.prm`, `chadeche2.prm` et `chadeche3.prm` pour les cartes d'adresse 0, 1, 2 et 3 respectivement ;
- Les valeurs fournies aux options de la commande `chadeche` utilisée pour lancer le programme.

En cas de multiple définition d'un paramètre des règles de précedence s'appliquent : la valeur fournie en option de la commande `chadeche` surcharge la valeur du paramètre défini dans le fichier de paramètres qui elle-même surcharge la valeur inscrite en dur dans le code source du programme. Notez que le fichier de paramètres n'est pas obligatoire.

5) La commande Chadèche

La commande Chadèche possède de nombreuses options mais vous n'aurez pas obligatoirement besoin de les spécifier toutes car dans bon nombre de cas, les valeurs par défaut vous conviendront. De plus si vous devez de façon récurrente utiliser une valeur différente de la valeur proposée par défaut, alors le mieux est de l'inscrire dans un fichier de paramètres.

Pour avoir un aperçu des options disponibles, le plus simple pour commencer, c'est de taper sous l'attente de commande de votre *shell* :

```
$ chadeche -h
```

pour obtenir l'aide. Chadèche vous répond :

```
Syntaxe: chadeche <option>
-a adresse de la carte mère (0,1,2,3)
-i charge initiale de la batterie
-c charge min de la batterie
-C charge max de la batterie
-n facteur de répétition
-f nom du fichier de résultats
-g nom du fichier de configuration
-p période d'enregistrement
-v mode verbeux
-t mode TEST activé
-m Vmin
-M Vmax
-h cette aide
```

Les valeurs par défaut (non reproduite ci-dessus) sont également affichées.

Les valeurs affectées aux paramètres c, C, m, M sont utilisées pour évaluer les conditions (vraies ou fausses) du fichier de configuration. Par exemple -m 1.1 signifie que la condition TL sera vraie quand la tension aux bornes de l'accu sera inférieure ou égale à 1,1 volt.

6) Autres paramètres

D'autres paramètres non accessibles par la commande *chadeche* peuvent être initialisés dans le fichier de paramètres :

- s : pente de la droite d'étalonnage en tension
- o : offset de la droite d'étalonnage en tension
- 1 : pente de la droite d'étalonnage en courant
- 2 : offset de la droite d'étalonnage en courant
- l : langage d'affichage des messages (français ou anglais)

A l'inverse, les paramètres -a et -h sont ignorés s'ils sont placés dans un fichier de paramètres.

7) Syntaxe des fichiers de paramètres

La syntaxe des fiches de paramètres *chadeche0.prm*, *chadeche1.prm*, *chadeche2.prm* et *chadeche3.prm* est très simple. Chaque paramètre occupe une ligne dans un fichier texte. Les identifiants de paramètres sont les mêmes que dans la ligne de commande de lancement du programme. Par exemple, pour fixer les coefficients d'étalonnage obtenus selon la procédure décrite à la section III.F, on inscrira :

```
-1 0.9851
-2 0.0000
-s 1.0099
-o 0.0013
```

E. Affichage et enregistrements

Au cours d'un essai, le programme Chadèche affiche et enregistre des informations.

1) L'affichage

Le Figure 14 (voir annexe à la fin de l'article) est un extrait d'affichage en cours d'essai. Les lignes sont affichées toutes les

10 secondes dans le mode verbeux et toutes les minutes dans le mode normal. Les différents champs sont décrits ci-dessous :

- C/01/07 signifie : Charge/Cycle n°1/Séquence n°7,
- DBA (daughter board adress) : adresse de la carte Chadèche,
- mA : courant de charge (ou de décharge),
- CET : cycle elapsed time : temps passé dans le cycle en cours (en secondes),
- SET : step elapsed time : temps passé dans la séquence en cours (en secondes),
- TET : total elapsed time : temps total passé depuis le début de l'essai (cumul des tous les cycles),
- STA : step time to arrival : temps restant à passer dans la séquence en cours (en secondes),
- V : tension mesurée aux bornes de l'accu (volts)
- mAh : charge estimée de l'accu (mAh)
- Date

Notez la ligne :

Starting step nr. 8 (Normal step starting): Maintient Elle annonce un changement de séquence. Le motif du changement est indiqué entre parenthèse. Le nom de la nouvelle séquence (« Maintien » dans l'exemple) est celui que vous avez indiqué en commentaire de la séquence dans le fichier de configuration.

2) Le fichier de résultats

Comme indiqué dans l'introduction, chaque Chadèche produit un fichier de résultats. Le nom de ce fichier est donné au moment du lancement du programme dans l'option -f. Attention : si vous lancez plusieurs Chadèche sur plusieurs cartes, veillez à utiliser des noms différents pour le fichier de résultats sinon les données produites par les différents Chadèche seraient mélangées.

Le fichier de résultats est un fichier texte au format CSV (le séparateur de champ est un point-virgule). On trouvera en annexe un extrait de ce fichier (Figure 15).

Par défaut, toutes les 120 secondes Chadèche enregistre une ligne dont les champs sont identiques à ceux décrits pour l'affichage mais ils ne sont pas labellisés (pas d'étiquette DBA, CET, SET etc.)

3) Utilisation de nohup

Vous devez ouvrir autant de fenêtres *shell* que de cartes Chadèche que vous souhaitez exploiter. L'inconvénient est que si vous fermez ces fenêtres au cours du déroulement d'un essai, tous les programmes qui auraient été lancés à partir de ces *shell* seront automatiquement terminés et Chadèche notamment. Il en est de même si vous lancez ces programmes en tâche de fond (en utilisant le caractère & à la fin de la ligne de commande).

La seule méthode est d'utiliser la commande *nohup* pour lancer vos commandes *chadeche*. Par défaut, tous les affichages sont redirigés dans un fichier *nohup.out*. Cette redirection ne convient pas si vous lancez plusieurs Chadèche car toutes les sorties seraient redirigées et mélangées dans ce fichier.

Il est donc suggéré d'utiliser la commande :

```
$ nohup COMMAND >chadechei.out
```

où COMMAND est la commande chadeche avec les options ad hoc (voir paragraphes III.D.4) et III.D.5) et chadeche*i*.out le nom du fichier de redirection des affichages avec *i* valant 0, 1, 2 ou 3 pour les cartes d'adresse 0, 1, 2 ou 3 respectivement.

En procédant ainsi, vous pouvez fermer vos *shell* et même éteindre votre PC. Mais vous ne pourrez plus utiliser CTRL-Z ou CTRL-C pour agir sur le déroulé d'un essai en cours. Il faudra envoyer des signaux à l'aide de la commande *kill* mais ceci sort du cadre de cet article.

F. Etalonnage

Avant toute utilisation de Chadèche il est nécessaire d'étalonner le générateur de courant commandé en tension ainsi que la mesure de la tension aux bornes de l'accu. Comme pour toute procédure d'étalonnage, une mesure de référence est nécessaire. Il faut pour cela disposer d'un ampèremètre et d'un voltmètre de bonne qualité. La procédure va consister à obtenir les coefficients *a* et *b* des équations des droites d'étalonnage. Ces coefficients seront ensuite introduits dans le fichier de paramètres de Chadèche pour apporter les corrections nécessaires. La procédure décrite ci-dessous doit être répétée pour chaque carte Chadèche.

Avant de lancer la procédure d'étalonnage, il faut s'assurer que les coefficients *a* et *b* sont à 1 et 0 respectivement. On inscrira dans le fichier chadeche*i*.prm les valeurs suivantes (où *i* est l'adresse de la carte en cours d'étalonnage) :

```
-s 1.0
-o 0.0
-1 1.0
-2 0.0
```

1) Etalonnage du générateur de courant commandé en tension

L'objectif est de tracer la caractéristique courant de consigne en fonction du courant mesuré.

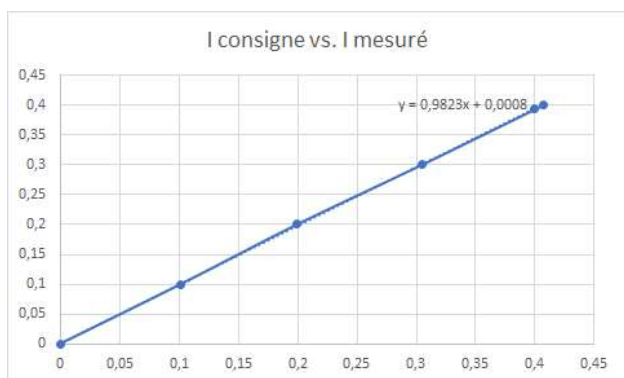


Figure 11 – Droite d'étalonnage du générateur de courant

Il faut procéder comme suit :

- Brancher un ampèremètre de bonne qualité sur les sorties destinées normalement à recevoir l'accu ;
- Lancez-le programme chadeche avec l'option `-t` et en utilisant le fichier de configuration `current-calibration.txt`. Ce fichier produit une consigne de courant en marche d'escalier avec des

valeurs de 0 à 250 mA par pas de 50 mA. Chaque consigne est appliquée pendant 30 secondes ;

- Pour chaque valeur de consigne, notez le courant mesuré sur l'ampèremètre.
- Introduire sous Excel ou LibreOffice les valeurs de courant mesuré et le courant de consigne et tracer la droite courant de consigne en fonction du courant mesuré ;
- Sur Excel, faire un clic droit sur la droite et dans le menu déroulant sélectionnez *Ajouter une courbe de tendance* et cochez la case *Afficher l'équation sur le graphique*.

La courbe de tendance est de la forme $ax+b$: notez les valeurs de *a* et de *b* (Figure 11).

Modifiez à présent le fichier chadeche*i*.prm avec les valeurs de *a* et de *b* que vous aviez notées. Dans l'exemple de la Figure 11 cela donne :

```
-1 0.9823
-2 0.0008
```

Si vous ne disposez d'aucun tableur, vous pouvez utiliser un site de calcul d'équation d'une droite de régression linéaire tel que celui-ci : <http://calculis.net/droite-regression-lineaire>.

2) Etalonnage de la mesure de la tension aux bornes de l'accu

Cette procédure permet d'étalonner la chaîne amplificateur différentiel et CAN.

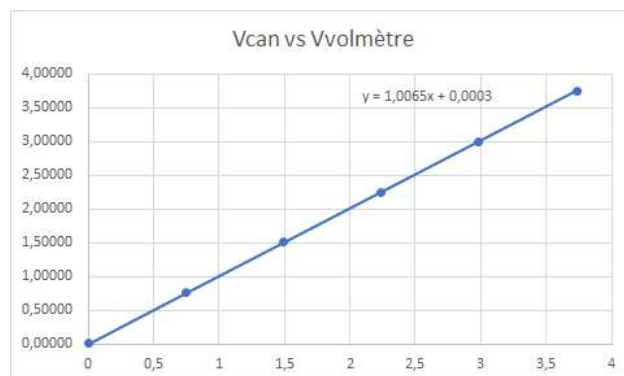


Figure 12 – Droite d'étalonnage de la mesure tension accu

Il faut procéder comme suit :

- A la place de l'accu, brancher une résistance de 5 à 10 ohms/2W ;
- Brancher aux bornes de cette résistance un voltmètre de bonne qualité ;
- Lancez-le programme chadeche avec l'option `-t` et en utilisant le fichier de configuration `voltage-calibration.txt`. Ce fichier produit une consigne de courant en marche d'escalier avec des valeurs de 0 à 350 mA par pas de 50 mA. Chaque consigne est appliquée pendant 30 secondes ;
- Pour chaque valeur de consigne, notez la tension mesurée sur le voltmètre et la tension affichée par le programme chadeche.
- Introduire sous Excel ou LibreOffice les valeurs de tension affichée par chadeche et tension mesurée par

le voltmètre et tracer la droite V voltmètre en fonction de V_{can} ;

- Sous Excel, faire un clic droit sur la droite et dans le menu déroulant sélectionnez *Ajouter une courbe de tendance* et cochez la case *Afficher l'équation sur le graphique*.

La courbe de tendance est de la forme $ax+b$: notez les valeurs de a et de b (Figure 12).

Modifiez à présent le fichier `chadeche.i.prm` avec les valeurs de a et de b que vous aviez notées. Dans l'exemple de la Figure 12 cela donne :

```
-s 1.0065
-o 0.0003
```

G. A l'intérieur de Chadèche

L'objectif n'est pas de décrire le code dans le détail mais plutôt d'insister sur quelques particularités liées à l'exploitation du parallélisme d'exécution offert par Linux. Comme on l'a vu précédemment, il est possible de lancer simultanément plusieurs Chadèche indépendamment les uns des autres, chacun d'eux s'adressant à l'une des quatre cartes qui forment un Chadèche complet.

En réalité l'indépendance est toute relative car tous ces Chadèche utilisent la même RPi pour échanger les données avec les cartes. Aussi, quelques précautions doivent être prises si l'on ne veut pas que les données d'une carte se mélangent avec celle d'une autre carte. Dit de façon plus rigoureuse, la RPi est une ressource critique qui ne peut être utilisée à un instant donné que par un seul processus (un seul Chadèche).

Une autre précaution consiste à empêcher de lancer simultanément plusieurs Chadèche sur la même carte. Ce serait totalement inefficace et ne présenterait aucun intérêt. Pour résoudre ces deux problèmes il a été fait usage des sémaphores et de mémoires partagées.

1) Sémaphores

Protéger une ressource critique, c'est empêcher que deux processus y accèdent en même temps. Le principe consiste donc de tester la disponibilité de la ressource ; si celle-ci est libre, le processus la déclare occupée pour son usage exclusif puis la libère après usage ; si celle-ci est déjà occupée, le processus s'endort (cela signifie qu'il ne consomme pas de temps de calcul de la RPi) et il sera réveillé dès que la ressource sera libérée. Cette « mécanique » logicielle est implémentée à l'aide des *sémaphores nommés*. La librairie *thread* offre deux fonctions `sem_wait` et `sem_post` pour gérer l'utilisation de ressources critiques. Le code critique doit être encadré par l'appel à ces fonctions comme indiqué ci-dessous :

```
...
sem_wait(semname);
...
...
/* code critique */
...
...
sem_post(semname);
...
```

Dans la littérature traitant des ressources critiques `sem_wait` et `sem_post` sont souvent nommées

respectivement *P* (puis-je ?) et *V* (vas-y !). En arrivant sur un code critique, le processus exécute un « puis-je ? pour la prendre et il est mis en attente si elle est occupée, sinon il la prend effectivement. Après usage, le processus exécute un « vas-y ! » libérant ainsi la ressource et indiquant à un autre processus qui pourrait avoir été mis en sommeil qu'il peut se réveiller et utiliser à son tour la ressource. Bien entendu, toute la « mécanique » de mise en sommeil et de réveil des processus est transparente pour le programmeur ; c'est le système d'exploitation qui s'occupe de tout.

On trouvera dans le code de Chadèche plusieurs sections de code critique : essentiellement celles qui contrôlent les sorties GPIO d'adressage et l'interface SPI.

2) Mémoire partagée

La possibilité offerte par Chadèche de lancer en parallèle jusqu'à quatre logiciels Chadèche soulève deux difficultés :

- D'abord, il revient au premier Chadèche lancé de créer les sémaphores. Ceux-ci devront exister tant qu'un ou plusieurs Chadèche s'exécutent et il reviendra au dernier Chadèche de détruire les sémaphores avant de se terminer lui-même.
- Ensuite on doit s'assurer que on ne lance pas deux programmes Chadèche sur la même carte.

La solution passe par une mémoire partagée entre tous les processus Chadèche.

Pour traiter la première difficulté, on a défini une variable `nprocess` qui est incrémentée par chaque Chadèche au moment de son lancement et qui est décrémentée par chaque Chadèche qui se termine. Ainsi, à un instant donné, cette variable a pour valeur le nombre de processus Chadèche en cours d'exécution. Quand le premier processus est lancé, il trouve que la variable `nprocess` est à zéro, il sait donc qu'il est le premier et qu'il doit créer le sémaphore. A l'inverse, quand le dernier processus se termine la variable `nprocess` passe à zéro après avoir été décrémentée et le processus sait qu'il est le dernier et qu'il doit par conséquent détruire le sémaphore

La deuxième difficulté se traite à l'aide d'une table `tprocess` à quatre entrées (0 à 3) initialisée à 0. Lorsqu'un processus est lancé pour la carte d'adresse i ($1 < i < 3$) il enregistre dans l'entrée i la valeur 1. S'il se termine il remet cette entrée à zéro. Supposons qu'on lance un premier processus Chadèche sur carte d'adresse 3. Le processus trouve l'entrée `tprocess[3] == 0` et affecte cette entrée avec la valeur 1. A présent, lançons un deuxième processus Chadèche sur la même carte. Cette fois-ci le processus trouve que la valeur `tprocess[3]` n'est pas nulle ce qui veut dire qu'il existe déjà un processus qui utilise cette carte. Du coup, le processus se termine aussitôt.

3) Organisation du code source

Le code source est fractionné en différents modules inscrits dans des fichiers séparés. Un Makefile permet de recompiler l'ensemble en cas de modification et de faire le lien avec les librairies requises.

IV. ETUDE DE CAS

Pour illustrer l'utilisation de Chadèche nous allons décrire un cas d'utilisation. Nous souhaitons décrire un cycle contenant deux phases : une phase de décharge suivie d'une phase de charge. La phase de charge est inspirée de celle implémentée dans le circuit spécialisé MAXIM DS2710. Il est conseillé au lecteur de se procurer la datasheet de ce circuit et de se référer au diagramme d'état de la page 6 et au texte explicatif qui l'accompagne.

On suppose qu'au lancement de l'essai, l'accu est chargé (c'est la raison pour laquelle on commence par une décharge). Toutefois, si ça n'est pas le cas, on verra que grâce aux conditions d'interruption de séquence, ce sera sans dommage pour l'accu.

Le fichier de configuration est décrit en Figure 10. Dans les lignes qui suivent, on désignera chaque séquence par son adresse.

A. La phase de décharge

Séquence 1 : On commence par une séquence de 30 secondes de repos. Le courant de décharge est nul. Cela permet d'enregistrer la tension de l'accu avant de commencer la décharge. Seule l'action CTRL-Z peut interrompre cette séquence et enchaîner l'exécution de la séquence suivante.

Séquence 2 : décharge avec un courant de 200 mA pendant 90000 secondes (25 heures !). En pratique cette séquence sera interrompue pour exécuter la séquence suivante, avant que ce temps soit écoulé, par l'une des deux conditions suivantes :

- La tension aux bornes de l'accu est inférieure au seuil TL ou bien (par défaut 1 volt),
- La charge de l'accu est inférieure au seuil LmAh (par défaut 100 mAh).

B. La phase de charge

La charge est réalisée par une succession de séquences de charge de 300 mA pendant 300 secondes séparées par des périodes de repos de 30 secondes.

Séquence 3 : séquence de repose de 30 secondes. Seule l'action CTRL-Z peut interrompre cette séquence et enchaîner l'exécution de la séquence suivante.

Séquence 4 : séquence de charge à 300 mA pendant 300 secondes à l'issue de laquelle on passe à la séquence 5.

Séquence 5 : cette séquence est utilisée pour faire un saut inconditionnel vers la séquence 3.

On observe que les séquences 3, 4, 5 forment une boucle qui s'achèvera lorsque l'une des deux conditions suivantes sera satisfaite (visibles dans la séquence 4) :

- La tension aux bornes de l'accu est supérieure au seuil TH (par défaut 1,75 volt) ou bien,
- La charge de l'accu est supérieure au seuil LmAh (par défaut 1300 mAh).

C'est alors la séquence 6 qui sera exécutée

Séquence 6 : charge à 50 mA pendant 300 secondes.

Séquence 7 : charge de maintien à 10 mA pendant 3600 secondes.

Séquence 8 : endormissement jusqu'à 20 heures.

Séquence 9 : après réveil fin du cycle, un nouveau cycle peut être lancé si plusieurs cycles ont été demandés au moment de lancer le programme chadeche (option -n).

V. CONCLUSION

Cet article montre les possibilités étendues offertes par Chadèche. Chadèche est avant tout un outil de recherche qui vous permettra d'explorer des lois de charge/décharge permettant d'optimiser la capacité de vos accus. Mais, vous pouvez aussi vous limiter à utiliser Chadèche comme un simple chargeur pour l'entretien de vos batteries. Dans ce cas il est conseillé de programmer des cycles de charges/décharge lentes et partielles sur le long terme pour ne pas laisser vos accus inactifs.

De nombreuses améliorations sont imaginables pour Chadèche. En voici quelques-unes :

- Développer de nouvelles conditions permettant d'interrompre une séquence en cours d'exécution et notamment la détection du delta-peak telle que décrite dans les références [1] et [3] ;
- Développer un analyseur syntaxique pour le fichier de configuration ;
- Implémenter le paramétrage de la led clignotante à partir du fichier de configuration,
- Développer une interface utilisateur plus ergonomique.

En projet dans mes cartons : un Chadèche à 32 voies pour recharger simultanément jusqu'à 32 accus 12 volts au plomb, ce qui je l'espère, fera le bonheur des clubs de vol en planeur.

REFERENCES

[1] Site web : « Le monde des accumulateurs et batteries rechargeables », <http://www.ni-cd.net/>

[2] Site Web: "WiringPi : GPIO Interface library for the Raspberry Pi", <http://wiringpi.com/>

[3] MAXIM DS2710 datasheet « Single Cell NiMh Charger », <https://datasheets.maximintegrated.com/en/ds/DS2710.pdf>

CONTACT

<mailto:chadeche@sfr.fr>

VI. ANNEXES

A. Une « pile » Chadèche

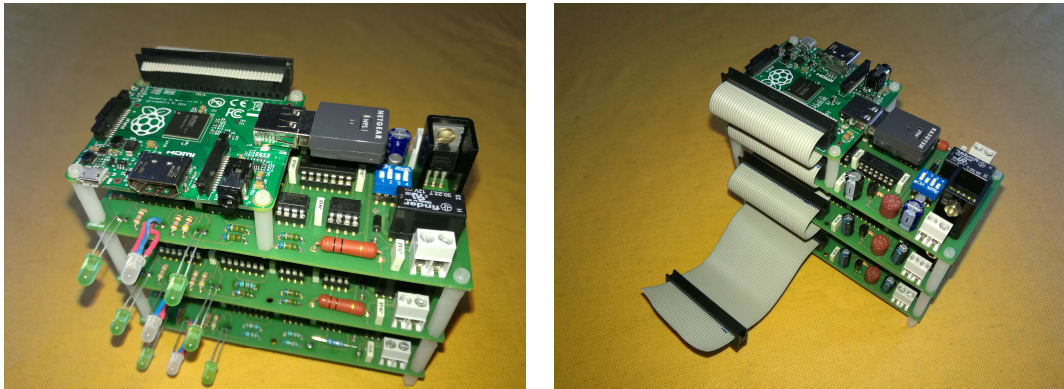


Figure 13 - Empilement de 3 cartes Chadèche et une RPi

B. Liste des composants pour une carte Chadèche

Désignation	Qté
<i>Pour une carte Chadèche</i>	
Résistance 1k ohms, 1/4 W	1
Résistance 12k ohms, 1/4 W	2
Résistance 220 ohms, 1/4 W	1
Résistance 150 ohms, 1/4 W	2
Résistance 470 ohms, 1/4 W	1
Résistance 680 kohms, 1%, 1/4 W	4
Résistance 10 ohms, 2 W	1
Condensateur LCC 100 nF	8
Condensateur radial 100 mF, 25v	2
Condensateur radial 47 mF, 25v	1
Diode 1N4007	1
Fusible rapide KSM 1A pour circuit imprimé pas 5,08	1
Bornier à vis 2 points pour circuit imprimé pas 5,08	2
Barette 2x20 contacts mâle droit pas 2,54	1
Connecteur femelle HE10F40	1
Câble en nappe 40 fils CBG40	1
Régulateur 72L05 régulateur 5v/100 mA, TO92	1
Transistor MOSFET IRF540N, boîtier TO220	1
Transistor bipolaire 2N2222	1
Circuit logique 74HC00, boîtier DIP 14	1
Circuit logique 74HC04, boîtier DIP 14	1
Circuit logique 74HC74, boîtier DIP 14	1
Ampli opérationnel LM432, boîtier DIP 14	1
Convertisseur numérique/analogique MCP4922-E/P	1
Convertisseur analogique/numérique MCP3201-C I/P, boîtier DIP 8	1
Référence de tension 4,096v MAX6241 BCPA, boîtier DIP 8	1
Circuit timer TLC555CP	1

Relais Finder Série 3022, 2RT, 12V	1
Led verte 10mA	2
Led bicolore verte/rouge 10 mA	1
Interrupteur dual in line 4 positions	1
1 dissipateur RA37/2 18°C/W	1
Support de circuit intégré tulipe DIP 16	1
Support de circuit intégré tulipe DIP 14	5
Support de circuit intégré tulipe DIP 8	4
Coupleur accu AA (LR06) et/ou AAA (LR03)	1
<i>Pour la Raspberry Pi</i>	
Carte Raspberry Pi Modèle A+ (ou autre RPi munie d'un connecteur 40 pts)	1
Carte microSD, 16 Go, classe 10	1
Dongle USB Wifi Netgear N150 WNA 1000 Mv2 (ou équivalent)	1
Connecteur femelle HE10F40	1
<i>Pour l'empilage de plusieurs cartes</i>	
Assortiment d'entretoises nylon hexagonales longueur 30 mm et vis M2,5 mm	

Tableau 2 – Liste des composants

C. Extrait d'affichage en cours d'essai

```

C/01/07 dba=0 mA= 50 CET= 17490 SET= 129 TET= 17490 STA= 51 V=1.373 mAh=1302.0, Mon May 22 03:21:13 2017
C/01/07 dba=0 mA= 50 CET= 17500 SET= 139 TET= 17500 STA= 41 V=1.373 mAh=1302.1, Mon May 22 03:21:23 2017
C/01/07 dba=0 mA= 50 CET= 17510 SET= 149 TET= 17510 STA= 31 V=1.373 mAh=1302.2, Mon May 22 03:21:33 2017
C/01/07 dba=0 mA= 50 CET= 17520 SET= 159 TET= 17520 STA= 21 V=1.373 mAh=1302.4, Mon May 22 03:21:43 2017
C/01/07 dba=0 mA= 50 CET= 17530 SET= 169 TET= 17530 STA= 11 V=1.373 mAh=1302.5, Mon May 22 03:21:53 2017
C/01/07 dba=0 mA= 50 CET= 17540 SET= 179 TET= 17540 STA= 1 V=1.373 mAh=1302.7, Mon May 22 03:22:03 2017
Starting step nr. 8 (Normal step starting): Maintient
C/01/08 dba=0 mA= 10 CET= 17550 SET= 8 TET= 17550 STA= 3592 V=1.369 mAh=1302.7, Mon May 22 03:22:13 2017
C/01/08 dba=0 mA= 10 CET= 17560 SET= 18 TET= 17560 STA= 3582 V=1.369 mAh=1302.7, Mon May 22 03:22:23 2017
C/01/08 dba=0 mA= 10 CET= 17570 SET= 28 TET= 17570 STA= 3572 V=1.369 mAh=1302.8, Mon May 22 03:22:33 2017
C/01/08 dba=0 mA= 10 CET= 17590 SET= 48 TET= 17590 STA= 3552 V=1.368 mAh=1302.8, Mon May 22 03:22:53 2017
C/01/08 dba=0 mA= 10 CET= 17600 SET= 58 TET= 17600 STA= 3542 V=1.368 mAh=1302.8, Mon May 22 03:23:03 2017

```

Figure 14 – Extrait de l'affichage en cours d'essai

D. Extrait d'enregistrement en cours d'essai

```

C;01;07;0; 50; 17520; 159; 17520; 21;1.373;1302.4;Mon May 22 03:21:43 2017
C;01;07;0; 50; 17530; 169; 17530; 11;1.373;1302.5;Mon May 22 03:21:53 2017
C;01;07;0; 50; 17540; 179; 17540; 1;1.373;1302.7;Mon May 22 03:22:03 2017
C;01;08;0; 10; 17550; 8; 17550; 3592;1.369;1302.7;Mon May 22 03:22:13 2017
C;01;08;0; 10; 17560; 18; 17560; 3582;1.369;1302.7;Mon May 22 03:22:23 2017
C;01;08;0; 10; 17570; 28; 17570; 3572;1.369;1302.8;Mon May 22 03:22:33 2017
C;01;08;0; 10; 17590; 48; 17590; 3552;1.368;1302.8;Mon May 22 03:22:53 2017
C;01;08;0; 10; 17600; 58; 17600; 3542;1.368;1302.8;Mon May 22 03:23:03 2017

```

Figure 15- Extrait d'un enregistrement réalisé au cours d'un essai