

Chadèche : chargeur/déchargeur intelligent pour accus NiMh

résumé d'un projet de lecteur

Jacques Ehrlich (France)
Texte de Jean-François Simon (Elektor)

Chadèche est un chargeur/déchargeur d'accus NiMH reposant sur un Raspberry Pi. Son logiciel permet de définir différents cycles de charge/décharge, d'enregistrer les mesures, et offre de nombreuses options de configuration. C'est parti pour un petit tour en Chadèche !



Figure 1. Un Raspberry Pi et trois cartes Chadèche.

Après avoir constaté que certains des accus NiMH de son GPS aéronautique tenaient la charge bien plus longtemps que d'autres, Jacques Ehrlich a cherché un moyen de les caractériser afin d'en sélectionner les plus performants. De là est née l'idée de *Chadèche*, contraction de **chargeur** et **déchargeur**. D'abord simple déchargeur à courant constant, l'instrument (**figure 1**) est devenu un projet de haute volée, qui plus est remarquablement documenté au travers d'un PDF de 12 pages et d'un Wiki personnel.

Chadèche peut charger ou décharger un accu NiMH aux formats AA ou AAA avec un courant constant réglable jusqu'à 400 mA, et enregistrer les valeurs mesurées dans un fichier. Sa conception modulaire permet de traiter de façon simultanée et indépendante jusqu'à quatre accus – un accu par carte-fille, la carte-mère (le Raspberry Pi) pouvant piloter quatre cartes-filles. Le logiciel, perfectionné, conduit les cycles de charge/décharge. Un cycle est composé de séquences, chaque séquence pouvant être définie par des conditions d'exécution telles que la tension atteinte aux bornes de l'accu, la valeur du courant de charge ou encore la durée de la charge.

Les deux faces de *Chadèche*

Parce qu'il est à la fois outil pratique et outil de recherche, *Chadèche* offre de nombreuses possibilités. Côté utilitaire, il permet de charger des accus NiMH, d'évaluer leurs performances, de mesurer leur capacité, et d'estimer leur durée de vie par répétition de cycles de charge et décharge. Côté expérimentation, *Chadèche* offre la possibilité de tester diverses lois de charge et de décharge en faisant varier différents paramètres (intensité, durée, etc.).

Ajoutons à ceci le plaisir d'assembler un projet de A à Z, doublé de la satisfaction d'avoir en main un instrument entièrement personnalisable. L'ensemble du projet *Chadèche* est par ailleurs sous licence open source GNU.

Une conception modulaire

Ce projet est composé de plusieurs cartes-filles *Chadèche* reliées à un Raspberry Pi via un câble en nappe à 40 contacts. Une carte ne traite qu'un seul accu. La version complète à quatre accus nécessite donc l'assemblage de quatre cartes *Chadèche*.

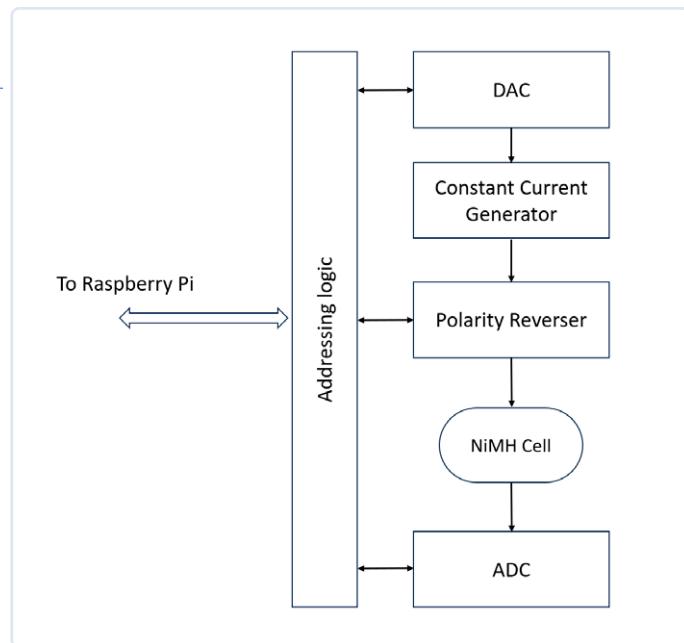


Figure 2. Schéma fonctionnel de *Chadèche*.

Le schéma de la **figure 2** montre l'architecture d'une carte-fille. On y trouve cinq blocs fonctionnels. Le premier est une source de courant constant commandée en tension, constituée d'un MOSFET *IRF540N* et d'une résistance de $10\ \Omega$, tous deux câblés en série avec l'accu. La tension produite aux bornes de la résistance par le passage du courant de charge est comparée à une tension de référence au moyen d'un amplificateur opérationnel. La sortie de l'ampli-op commande la grille du MOSFET – qui opère dans sa région ohmique.

Le bloc *DAC* du schéma comprend un convertisseur N/A *MCP4022* dont le rôle est de produire la tension de consigne utilisée par la source de courant constant mentionnée précédemment. Le *MCP4022* possède deux voies ; la première est utilisée à cet effet.

Le bloc *ADC* comprend un convertisseur A/N *MCP3201* numérisant la tension de l'accu afin d'en obtenir une mesure précise.

Le quatrième bloc effectue le passage de charge à décharge grâce à un relais inverseur de polarité DPDT.

Le dernier bloc est celui de la logique d'adressage, le mécanisme garantissant que le Raspberry Pi communique avec chacune des cartes (jusqu'à quatre) de façon indépendante et sans conflit. L'utilisateur définit l'adresse de chaque carte au moyen de commutateurs DIP. Il faut deux bits pour coder quatre adresses en binaire ; ils sont représentés par les deux broches d'E/S *A0* et *A1* (broches GPIO 17 et 27). Le Raspberry Pi sélectionne la carte-fille souhaitée en modifiant l'état de ces broches.

Vous trouverez le schéma complet sur la page du projet *Elektor Labs* [1] et sur le site web de l'auteur [2].

Du concept au circuit

Outre ces blocs principaux, la conception de *Chadèche* présente plusieurs points intéressants. C'est le cas des convertisseurs N/A et A/N, commandés par le Raspberry Pi via le bus SPI. Ils partagent les broches *MISO*, *MOSI* et *CLK*, le passage de l'un à l'autre se faisant par basculement de l'état des broches de sélection de puce auxquelles ils sont associés.

Comme les broches SPI sont également partagées entre les cartes-filles, la logique d'adressage se charge aussi d'acheminer les signaux de sélection de puce vers la carte appropriée. Cette logique est obtenue

avec les quatre portes ET d'un *74HC00* et quatre des six inverseurs d'un *74HC04*.

Les convertisseurs N/A et A/N ont en outre besoin d'une référence de tension. Plutôt que d'utiliser un des rails d'alimentation comme référence, et afin d'obtenir des résultats plus précis, l'auteur a choisi un *MAX6241* d'Analog Devices.

Une facette intéressante du CA/N est son entrée pseudo-différentielle. La broche d'entrée négative, contrairement à une entrée négative flottante, est limitée à une plage de $+/ -100\text{ mV}$ par rapport à la masse. Or, le pôle négatif de l'accu n'est pas directement relié à la masse, mais est flottant par rapport à celle-ci, à cause de la résistance série de $10\ \Omega$. Pour y remédier, un amplificateur différentiel à gain unitaire a été placé entre l'accu et le CA/N.

Le relais inverseur de polarité (le relais de charge/décharge) doit rester en position stable même lorsque le Raspberry Pi dialogue avec une autre carte. La commande de sa bobine est donc assignée à un *74HC74* utilisé comme verrou (*latch*), lui-même commandé par deux broches GPIO du Raspberry Pi.

Le circuit comprend aussi trois LED témoins. La première s'allume lorsque le circuit est sous tension. La deuxième, bicolore, indique si *Chadèche* est en mode charge ou décharge. La troisième fonctionne selon un schéma on/off dicté par le signal de sortie d'un *timer 555*. Il est intéressant de noter que le rapport cyclique de ce signal de sortie est contrôlé par la deuxième voie du CN/A, qui sans cela resterait inexploitée. Cette configuration permet à la LED de clignoter à des fréquences variables, ajustables depuis le logiciel. Une technique de clignotement bien compliquée penserez-vous peut-être, mais qui offre l'avantage d'offrir une fréquence de clignotement fixe, même lorsque le Raspberry Pi est accaparé par d'autres tâches.

Cette LED pourra p. ex. refléter le courant de charge actuel. Une variation de sa luminosité aurait pu suffire, mais l'effet aurait été fortement non-linéaire ; un clignotement est un indicateur plus intuitif.

Réflexions sur l'architecture du système

Les contraintes de temps et de budget sont en général plus souples pour un projet personnel que pour un projet d'entreprise, ce qui est une bonne chose ! Un concepteur est souvent tiraillé entre l'envie de créer quelque chose de simple et celle de mettre au point un système flexible et évolutif. Ce type de défi est toujours intéressant à relever. *Chadèche* utilise une carte-fille par accu, et il est tentant d'imaginer une solution à carte unique qui simplifierait le câblage et l'assemblage mécanique. Le circuit imprimé pourrait p. ex. être plus grand et loger un Raspberry Pi Zero.

Qu'il y ait un MOSFET et un circuit de commande de grille par accu NiMH semble s'imposer, mais ne serait-il pas possible d'utiliser une seule paire CA/N-CN/A pour tout le système ? Le nombre d'échantillonnages nécessaires passerait ainsi de quatre à un. Et pourquoi pas explorer différents schémas d'adressage afin d'étendre *Chadèche* à plus de quatre accus ? On pourrait aussi chercher à l'adapter à d'autres accus, comme des batteries au plomb ou au lithium-ion. Et vous, comment auriez-vous procédé ? Partagez vos réflexions avec la communauté sur *Elektor Labs* [1].

Jean-François Simon (ingénieur, *Elektor*)

ADR	C/D	mA	SD	TL	TH	LmAh	HmAh	TRUE	CTRLZ	COMMENT
1	D	0	60						S	Cell at rest
2	D	200	90000	S		S			S	200 mA discharge
3	D	0	30						S	Cell at rest
4	C	300	300		6		6		S	300 mA charge until max capacity
5	C	0	5					3	S	NOP and unconditional jump
6	C	50	300						S	Topping off
7	C	10	3600						I	Trickle charge
8	C	0	0					20:00	S	Waking up at 20:00
9	C	0	0						S	End of cycle

Figure 3. Exemple de configuration pour un cycle.

Logiciel

Jacques Ehrlich a aussi écrit le logiciel pilotant *Chadèche*. Le code source (en C) et les versions compilées pour le Raspberry Pi sont disponibles sur Github [3]. Le code est méthodiquement organisé en cinq fichiers sources totalisant environ 1300 lignes.

L'accès aux broches GPIO du Raspberry Pi se fait avec la bibliothèque *WiringPi* de Gordon Henderson – la dernière version stable, disponible sur le site de Gordon, devrait fonctionner.

L'exécutable *chadeche* attend en paramètre l'adresse de la carte-fille à traiter. Grâce aux capacités multitâches de Linux, il est possible d'analyser plusieurs accus simultanément : soit en lançant plusieurs instances de *chadeche* dans des shells différents, soit en exécutant les commandes en arrière-plan avec &, soit en utilisant les commandes *screen*, *nohup*, etc.

Le logiciel permet de définir différents cycles de charge et de décharge, chaque cycle étant constitué de séquences. Leur déroulement peut être dicté par différents critères, dont des sauts conditionnels. Par exemple : charger avec un courant X durant Y secondes, passer à la séquence suivante lorsque la tension Z est atteinte, etc.

La configuration des cycles se fait au moyen de fichiers texte au format CSV contenant autant de détails que souhaité (une séquence par ligne). La **figure 3** montre un exemple de configuration. On peut également fournir à *Chadèche* des fichiers de paramètres stockant des valeurs spécifiques à une carte, p. ex. les coefficients de ses droites d'étalonnage.

Les résultats des mesures sont enregistrés au format CSV. On peut ainsi les exploiter plus tard, p. ex. en traçant le graphe d'un cycle (**figure 4**). ↗

VF : Hervé Moreau — 230624-04

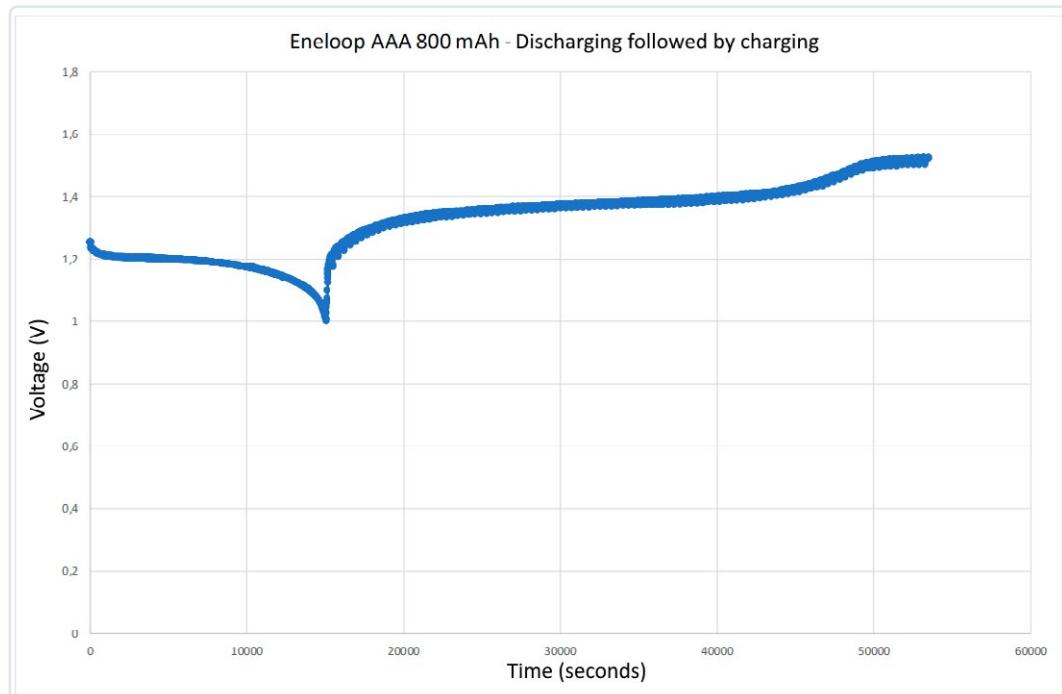


Figure 4. Courbe de la tension d'un accu subissant un cycle de décharge et charge.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (support.chadeche@orange.fr) ou contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

Jacques Ehrlich est directeur de recherche émérite à l'Université Gustave Eiffel et consultant dans le domaine des systèmes de transport intelligents (STI). Il a codirigé puis dirigé le LIVIC (Laboratoire sur les Interactions Véhicule, Infrastructure Conducteur) à l'IFST-TAR, un laboratoire de recherche de renommée mondiale dans le domaine de la sécurité routière, des aides à la conduite et des véhicules autonomes. Il est passionné d'électronique, d'information et de vol à voile.

Partagez vos projets avec Elektor

Chadèche est à nos yeux un bel exemple de réalisation menée de A à Z par une seule personne : conception, construction, test, écriture du code et de la documentation. Saluons donc l'effort considérable qu'il a nécessité. Un tel projet demande de la persévérance et de nombreuses semaines de travail. Il est le fruit d'une authentique passion, et c'est ce que nous aimons voir chez Elektor ! N'hésitez pas à partager vos propres projets avec la rédaction et la communauté Elektor. Présentez votre travail sur Elektor Labs, et envoyez vos idées de projets ou d'articles à redaction@elektor.fr.



Produits

► **Raspberry Pi 4 B (2 Go RAM)**
www.elektor.fr/18965



LIENS

[1] Page du projet sur Elektor Labs :

<https://elektormagazine.fr/labs/chadeche-un-chargeur-dechargeur-tres-intelligent-pour-accus-nimh>

[2] Site web de l'auteur : <http://yadalo.no-ip.org/dokuwiki/doku.php?id=projets:projets:chadeche:chadeche>

[3] Logiciel Chadèche sur GitHub : <https://github.com/duodiscus92/chadeche>

Ils nous font confiance, n'est-ce pas ?

Nous aimons l'électronique et les projets, et nous faisons tout notre possible pour répondre aux besoins de nos clients.

Le magasin Elektor :
Jamais cher,
toujours surprenant

Consultez d'autres avis sur notre page Trustpilot : www.elektor.fr/TP

Vous pouvez également vous faire votre propre opinion en visitant notre Elektor Store, www.elektor.fr

“Livraison rapide et complète de tous les articles commandés, à l'adresse précisée, parfait, merci à l'équipe d'Elektor.”
by Mami Maggioli
Rated 4.6 / 5 | 388 reviews

“Excellent Transaction – RAS ! Sur une dernière commande, d'un montant de 50 euros, tout s'est bien déroulé et plus rapidement que prévu. L'objet est en excellent état et conforme à l'annonce.”
by Addo Addo
Rated 4.6 / 5 | 400 reviews

“Produit livré rapidement, bien emballé et conforme à la commande”
by Albert Jourvat
Rated 4.6 / 5 | 398 reviews

“Très satisfait – Envoi très rapide, bien emballé. Je recommande vivement !”
by FORTIN GUILLAUME
Rated 4.6 / 5 | 398 reviews

elektor
design > share > earn