

Une gestion de l'énergie plus intelligente pour l'internet des objets

Seconde partie : mise en œuvre

Éric Djakam

RÉSUMÉ

Dans un précédent billet, nous avons traité des paramètres ayant une incidence sur les performances des piles et proposé plusieurs approches correctives fondées sur les concepts du système et du contrôle de santé de la pile. Nous allons maintenant décrire des exemples de mise en œuvre de ces approches avec le bq35100 et l'ads7142. Nous présenterons également la plateforme de test du bq35100, conçue pour simplifier l'interface entre le bq35100 et les autres produits.

1 Contrôle de la santé de la pile avec le bq35100

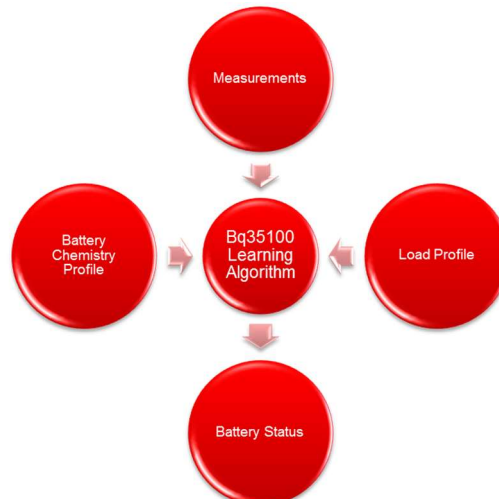
1.1 Introduction

Le bq35100 permet trois modes de fonctionnement logiciel configurables pour le contrôle de la santé des piles :

- État de santé (SOH), pour les piles Li-MnO₂
- Fin de vie utile (EOS), pour les piles Li-SOCl₂
- Accumulation de Coulomb (ACC), pour toute solution électrochimique

La mise en œuvre de l'une de ces fonctionnalités au moyen d'un microcontrôleur externe demande un temps de préparation pour créer le profil de la pile, élaborer l'algorithme d'apprentissage et optimiser le code du microcontrôleur en vue d'une consommation énergétique optimisée. Le client devra répéter cette opération pour chaque nouvelle pile. L'algorithme d'apprentissage du bq35100 utilise les données relatives à la durée de vie pour estimer les paramètres critiques de la pile. Il convient à la plupart des piles primaires au lithium actuellement disponibles sur le marché.

Figure 1. Modes de fonctionnement du bq35100



1.2 Contrôle de la santé des piles lithium-chlorure de thionyle (Li-SOCl₂)

Dans ce mode de fonctionnement, le bq35100 mesure la tension, l'intensité et la température de la pile pour calculer son impédance. Il contrôle ensuite la courbe de variation de l'impédance et détecte tout pic de variation, qui indique que la pile a atteint sa fin de vie utile.

Lors de l'utilisation, il n'est pas nécessaire de contrôler les paramètres de la pile en permanence. L'ingénieur peut laisser le bq35100 déconnecté la plupart du temps et synchroniser ses opérations au moyen d'une charge haute intensité telle qu'un émetteur RF.

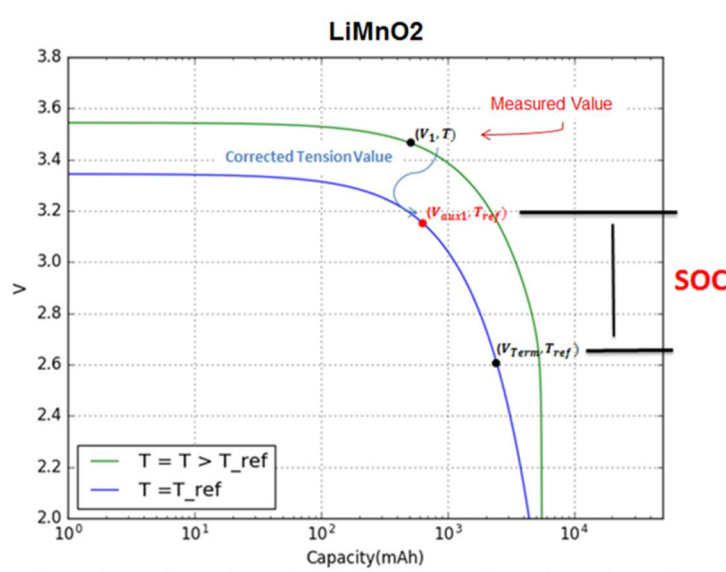
Dans ce mode de fonctionnement, la consommation du système actif est de l'ordre de 315 μ A, soit une moyenne de 114 nA pour un système activé 30 s toutes les 23 heures. Pour reprendre l'exemple du compteur d'eau communicant, le courant d'alimentation d'un amplificateur de puissance Wireless M-Bus 169 MHz est de l'ordre de 180 mA et l'intensité en veille du compteur est d'environ 5 μ A. La consommation électrique du dispositif de contrôle de la pile doit être négligeable par rapport à la consommation du système, ce qui est le cas ici.

1.3 Contrôle de la santé des piles lithium-dioxyde de manganèse (Li-MnO₂)

Dans ce mode de fonctionnement, le bq35100 utilise les informations de tension et de température de la pile pour estimer son état de charge (SOC). La tension mesurée aux bornes est comparée à une valeur de référence prédéfinie pour déterminer l'état de charge. La température est utilisée pour corriger la valeur de tension mesurée.

Lors du fonctionnement, il est recommandé d'effectuer les mesures en circuit ouvert (c'est-à-dire sans courant de charge), une fois la phase de relaxation terminée. Dans ce mode de fonctionnement, la consommation du système actif est de l'ordre de 40 μ A, soit une moyenne de 15 nA pour une mesure de 30 s toutes les 23 heures.

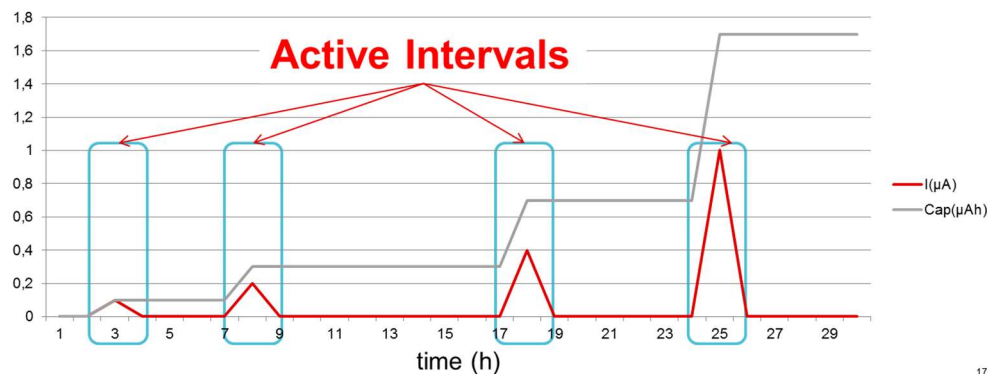
Figure 2. Mode de fonctionnement avec pile Li-MnO₂



1.4 Contrôle de la santé des piles en mode accumulateur

Dans ce mode de fonctionnement, le bq35100 est capable de suivre et d'enregistrer la quantité d'énergie fournie par la pile à chaque seconde. Il convient de remarquer que c'est la communication des informations qui prend 1 s (et non la mesure des paramètres de la pile). Dans ce mode de fonctionnement, la consommation du système actif est de l'ordre de 130 μA . Il est recommandé d'utiliser l'ads7142 pour mettre en œuvre ce type de fonction.

Figure 3. Mode accumulateur



1.5 Plateforme de test de contrôle de la santé de la pile

Le bq35100 est un produit à signal mixte intégrant des fonctions analogiques et numériques. Un microcontrôleur à interface I2C est nécessaire pour déterminer le mode de fonctionnement. Afin d'accélérer le processus de conception, nous avons conçu une plateforme de test pour fournir aux ingénieurs :

- Une bibliothèque illustrant les étapes à suivre pour communiquer avec l'appareil (code décrit ci-dessous)
- La temporisation nécessaire pour connecter l'appareil

Le contrôle de pile primaire étant un concept nouveau, le client commencera probablement l'évaluation du produit sur ses propres systèmes. Cela lui permettra de comparer cette technologie à d'autres concepts tout en se familiarisant avec l'utilisation du produit.

Nous avons souhaité simplifier l'intégration du produit dans des systèmes existants afin que les ingénieurs puissent connecter le bq35100 aux systèmes dotés de piles et commencer l'évaluation en toute simplicité.

Figure 4. Intégration du bq35100 au système

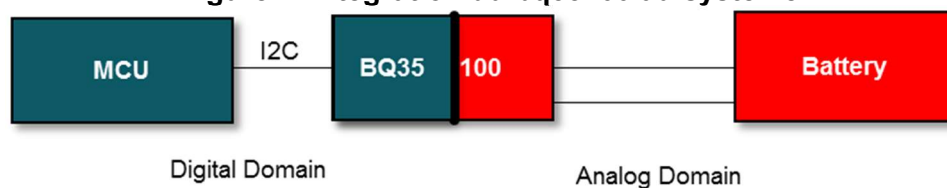
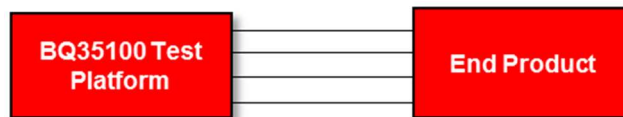


Figure 5. Principe de la plateforme de test



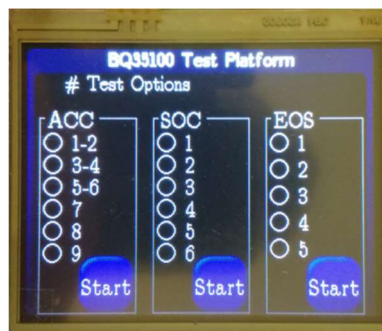
1.6 Description matérielle de la plateforme de test du bq35100

La plateforme de test nécessite quatre signaux :

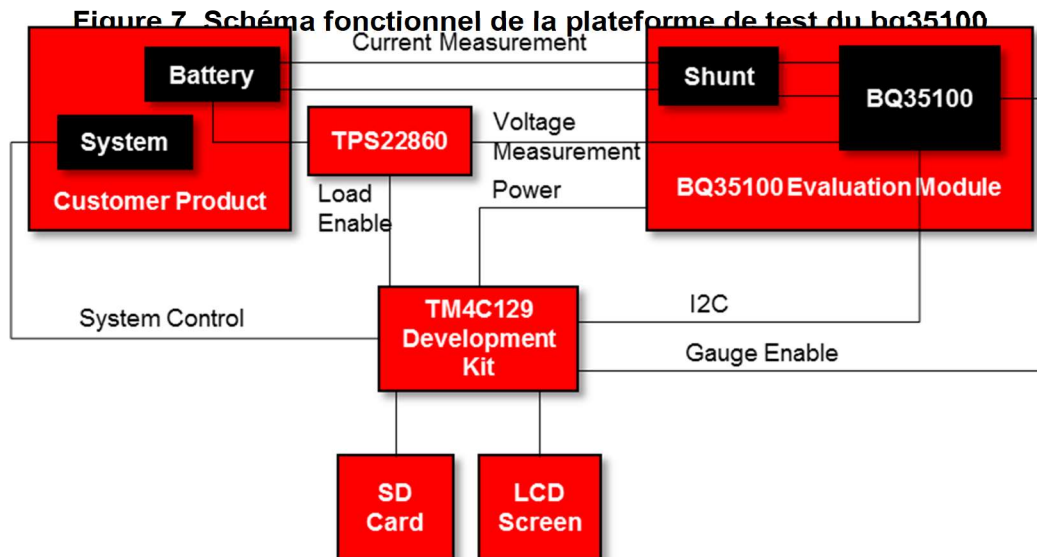
- Deux fils pour connecter la pile en série avec le shunt
- Un fil pour mesurer la tension de la pile
- Un fil pour communiquer au microcontrôleur l'état du système (actif ou en veille)

Le principe de fonctionnement est le suivant : L'utilisateur peut sélectionner un profil d'essai sur l'écran LCD. Lorsque le système est en veille, le bq35100 est désactivé pour économiser l'énergie. Lorsque le système est actif, le bq35100 est activé, il assure le contrôle de la pile et les données sont enregistrées dans la carte SD à un format compatible avec Excel. L'utilisateur peut ensuite lire le contenu de la carte SD sur ordinateur.

Figure 6. Écran d'accueil de la plateforme de test



Le kit de développement TM4C129 sert d'alimentation au bq35100EVM. Le TM4C129 doit être alimenté au moyen d'une alimentation externe 5 V.



1.7 Description logicielle de la plateforme de test du bq35100

Le logiciel est constitué de quatre composantes principales assurant l'automatisation des essais, la programmation du bq35100, l'actualisation de la carte SD et l'actualisation de l'écran LCD.

Comme le montre la figure 9, un essai de structure standard évalue la durée d'impulsion, commandée par l'interrupteur de charge, le nombre d'impulsions et la durée de repos. La structure de l'essai est fournie dans le fichier automatization.h. Le fichier automatization.c exécute le protocole d'essai en fonction des informations entrées dans la structure de l'essai.

Figure 8. Lien entre les composantes logicielles et matérielles

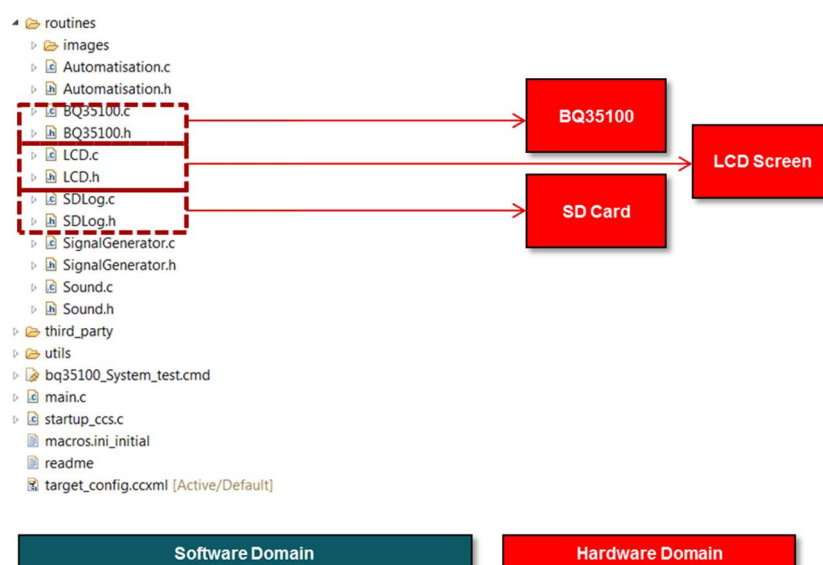


Figure 9. Structure de test standard

```

1//*****
2//
3// Struct used to store tests information.
4//
5//*****
6struct Test_param {
7    uint16_t    PulseDuration;    //[ms]
8    int         NumberOfPulses;    //
9    uint32_t    RestDuration;    //[s]
10    bool        GEOperation;    //GEOperation = true ( GE = 1 only during pulses), else, GE = 1 during whole proces.
11    char*       name;    //Name of the file to save this test data.
12};
13
14//*****

```

Test ID	Pulse Duration (s)	Number of Pulses	Rest Duration	Gauge Enable Management
Acc	2	10	60	GE = 1 only during pulses

Comme le montre la figure 10, le fichier BQ35100.h comporte l'ensemble des fonctions et de la définition du registre du bq35100 pour pouvoir communiquer avec l'appareil.

Figure 10. Fichier bq35100.h

```

BQ35100.h
28 #define BQ_ADDRESS 0x55
29
30 //*****
31 //
32 // Adress of the Control() Register.
33 //
34 //*****
35 #define CMD 0x00 //Use for 1-byte commands
36 #define BlockCMD 0x3E //Use for n-bytes commands
37
38 //*****
39 //
40 // BQ35100 Control() Commands.
41 //
42 //*****
43 #define GAUGE_START 0x0011
44 #define GAUGE_STOP 0x0012
45 #define LT_ENABLE 0x002E
46 #define RESET 0x0041
47 #define NEW_BATTERY 0xA613
48
49 //*****
50 //
51 // Adress of the Information Commands.
52 //
53 //*****
54 #define ControlStatus 0x00
55 #define AccumulatedCapacity 0x02
56 #define Temperature 0x06
57 #define Voltage 0x08
58 #define BatteryStatus 0x0A
59 #define AlertStatus 0x0B
60 #define Current 0x0C
61 #define StateOfHealth 0x2E
62 #define DesignCapacity 0x3C
63
64 //*****
65 //
66 //*****
67 //
68 //*****
69 //
70 //*****
71 //
72 //*****
73 //
74 //*****
75 //
76 //*****
77 //
78 // Struct used to store BQ's information.
79 //
80 //*****
81 struct BQInfo {
82     uint16_t ControlStats;
83     uint32_t AccCap;
84     int16_t Temp;
85     uint16_t V;
86     uint8_t BatStats;
87     uint8_t AlertStats;
88     int16_t I;
89     uint8_t SOH;
90     uint16_t DesignCap;
91 } BQInfo;
92
93 //*****
94 //
95 // API Function prototypes
96 //
97 //*****
98 void BQ_Init(void);
99 void BQ_End(void);
100 double getVBat(void);
101 void setBQ_CMD(uint16_t command);
102 uint32_t getInfo(uint8_t InfoRegister);
103 uint16_t getDataFrame(uint16_t Address);
104 void setDataFrame(void);
105 uint32_t getBQ_ControlStatus();
106 void Sample(struct BQInfo *BQ);
107 void Monitoring_Init(void);
108 void Monitoring_End(void);
109 void GE(bool GElevel);
110 void LOAD(bool LOADlevel);
111

```

1.8 Résultats

Nous avons testé cette plateforme en utilisant des résistances comme charge. La figure 8 montre les résultats en mode accumulateur. Comme on le voit ci-dessus, l'appareil surveille le courant de charge en fournissant des données précises. Il convient de prendre en compte les points suivants :

Figure 11. Résultats du bq35100 en mode accumulateur

Battery Type	Test Name	Current pulse (mA)	Pulse duration (s)	Number of pulses	Rest duration (s)	GE Management	Shunt Resistor	Shunt Voltage (mV)	Measured Battery Voltage (V)	Average Current (mA)	Energy (uV.s)	Measured Energy (uAh)	Expected Energy (uAh)	Load Resistance (Ohms)	Er
LiMnO2	Acc1	11.11	0.1	16	1	1 ON all time	0.1	1.11	3	1.010101	300	5	5.56	270	
	Acc2	11.11	2	9	1	1 ON all time	0.1	1.11	3	7.407407407	6000	56	55.56	270	
	Acc3	25.42	0.1	6	1	1 ON all time	0.1	2.54	3	2.31248074	300	3	4.24	118	
	Acc31	25.42	0.1	60	1	1 ON all time	0.1	2.54	3	2.31248074	300	42	42.37	118	
	Acc4	24.58	2	3	1	1 ON all time	0.1	2.46	2.9	16.38418079	5800	35	40.96	118	
	Acc5	11.11	0.1	9	1	1 ON all time	0.1	1.11	3	10.10101	300	27	27.78	27	
	Acc6	36.30	2	9	1	1 ON all time	0.1	3.63	2.6	64.13753066	5300	486	481.48	27	
	Acc7	25.42	0.1	6	30	30 ON all time	0.1	2.54	3	0.0846425	300	3	4.24	118	
	Acc71	11.11	0.1	6	30	30 ON all time	0.1	1.11	3	0.36913504	300	17	18.52	27	
	LiSOD2	Acc1	13.33	0.1	16	1	1 ON all time	0.1	1.33	3.6	1.2121212	360	5	5.67	270
LiSOD2	Acc2	13.19	2	9	1	1 ON all time	0.1	1.32	3.56	6.78012467	7120	66	65.93	270	
	Acc3	30.25	0.1	6	1	1 ON all time	0.1	3.03	3.57	2.75938208	357	3	5.04	118	
	Acc31	30.51	0.1	60	1	1 ON all time	0.1	3.05	3.6	2.773497689	360	52	50.85	118	
	Acc4	30.25	2	3	1	1 ON all time	0.1	3.03	3.57	20.854953	7140	50	50.42	118	
	Acc5	132.59	0.1	9	1	1 ON all time	0.1	13.26	3.99	12.9530705	359	35	31.45	27	
	Acc6	122.22	2	9	1	1 ON all time	0.1	12.22	3.3	81.4814848	6600	619	611.11	27	
	Acc7	30.51	0.1	6	30	30 ON all time	0.1	3.05	3.6	0.91357158	360	3	5.08	118	
LiSOD2	Acc71	136.30	0.1	6	30	30 ON all time	0.1	13.63	3.68	0.452017616	368	21	22.72	27	

- **Choix du shunt** : le compteur de coulombs est basé sur une architecture à convertisseur analogique-numérique delta-sigma intégré. Afin de pouvoir être mesurée avec précision, toute impulsion appliquée au shunt doit générer une tension minimale de 20 μV pendant 1 s. Ainsi, toute tension inférieure à 1 μV est considérée comme du bruit. Il faut donc s'assurer que la tension sur le shunt est supérieure à 1 μV .
- **Calcul des grandeurs électriques** : les valeurs indiquées par le bq35100 ne correspondent pas totalement à nos calculs. Aucune donnée n'a été perdue ; ces divergences s'expliquent par le fait que les intensités mesurées sont très faibles. Chaque seconde, les 16 premiers bits du compteur interne sont copiés à l'adresse du bus de données et effacés. Le firmware du bq35100 doit lire cette valeur et la traiter avant la seconde suivante. Elle est ajoutée au registre d'accumulation de charge RAM et l'intensité moyenne présente durant cette seconde est également calculée. Cette valeur est ensuite transférée à la mémoire flash lorsque la commande d'arrêt du moniteur est engagée.

2 Contrôle de la santé de la pile avec l'ads7142

L'ads7142 est un moniteur de capteurs autonome nanopuissance avec comparateur à fenêtre pour une activation de l'hôte déclenchée par événement.

Ici, nous utilisons l'ads7142 pour contrôler en permanence la consommation électrique du système à l'état actif et en veille. L'appareil peut émettre un signal d'alarme lorsque ces valeurs diffèrent des valeurs de référence. Cette information pourra ensuite être utilisée par l'hôte, comme nous l'avons décrit dans le billet précédent.

Un amplificateur opérationnel tlv521 a été ajouté avant l'ads7142 pour amplifier le signal afin de pouvoir mesurer les intensités inférieures à 1 μA .

Le coût énergétique de l'utilisation de l'ads7142 est d'environ 900 nW, ce qui est compatible avec les exigences relatives aux compteurs communicants.

Conclusion

Les techniques présentées ici permettent de mettre en œuvre les fonctions du système et du contrôle de la pile en gardant une consommation électrique très faible. La plateforme décrite est stable et permet au client d'évaluer le bq35100. Pour plus d'informations, vous pouvez consulter les guides techniques du bq35100 et de l'ads7142 sur le site www.ti.com.