Note d’application

Mesure de la consommation d'un Dragino LSN50 et optimisation

Table des matières

Introduction

Le Dragino LSN50-V2 (figure 2) est un objet connecté à base d'un microcontrôleur STM32, qui s'occupe de gérer toute la partie mesure et envoie. Le système est alimenté avec une pile (figure 1) au lithium-chlorure de thionyle (Li/SOCI2). Ce type de pile n'est pas rechargeable, mais elle délivre une tension stable de 3,6 Volte, possède une autodécharge de 1% après 1 an à 25°C, qui est une valeur faible comparée à d'autres batteries (ex : pile lithium 18650, autodécharge <2% par mois). Ainsi, cette pile est faite pour durer dans le temps, sans avoir à la recharger ou à y toucher. Les capacités possibles pour cette pile dans un Dragino LSN550-V2 sont de 4000 mAh ou 8500 mAh en fonction de la version de celui-ci. Les tests, et l'étude du système seront faits avec la capacité la plus contraignante, c'est-à-dire 4000 mAh.

Figure 1: Pile Li/SOCI2 du Dragino

Figure 2: Photographie du Dragino LSN50-V2

Comment mesurer la consommation du Dragino LSN50-V2

La mesure de la consommation du Dragino est faite avec un multimètre de précision (keithley 2700) qui permet de mesurer une tension à 1µV près. Le courant délivrée par la pile pour le Dragino LSN50-V2 est mesure à l'aide d'un shunt de 982 mΩ, comme montrer sur la figure 3. La version du logiciel du microcontrôleur est celle où nous l'avons modifié afin de permettre la lecture et l'envoie de plusieurs sondes HYT939 sur le bus I2C du système.

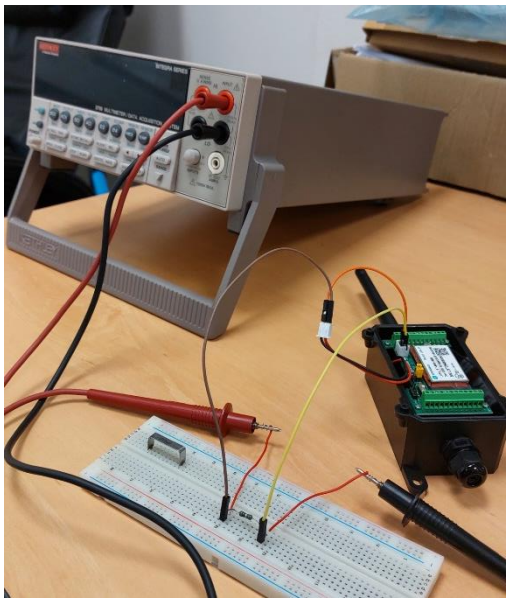

Figure 3: Photographie du système pour la mesure

Le cycle de consommation du Dragino LSN50-V2 se décompose en 3 parties (figure 4). La première est l'acquisition des data provenant des capteurs (demande de mesure et récupération des valeurs mesurées), indiqué en bleu sur la figure. Ensuite, il y a l'envoie des données, en vert, qui est la partie où le Dragino envoi les données mesurées par les sondes, avec le protocole LoRaWAN. En fonction de l'accessibilité de l'antenne la plus proche, le Dragino LSN50-V2 peut s'adapter pour que les données soient toujours reçues. Pour cela, il utilise une modulation à étalement de spectre qui va ralentir la vitesse de transmission, mais va étendre le porté du signal. Dans cette étude, on utilise un spread factoring de 7 (SF7), qui est le facteur d'étalement du spectre, et un data rate de 5 (DR5). À savoir que c'est la configuration la meilleure, c'est-à-dire celle qui consomme le moins. Enfin, en rouge, c'est le stop mode, lorsque le microcontrôleur est en veille profonde et consomme le moins. C'est aussi la partie du cycle la plus longue, dans notre cas elle, dure 20 minutes.

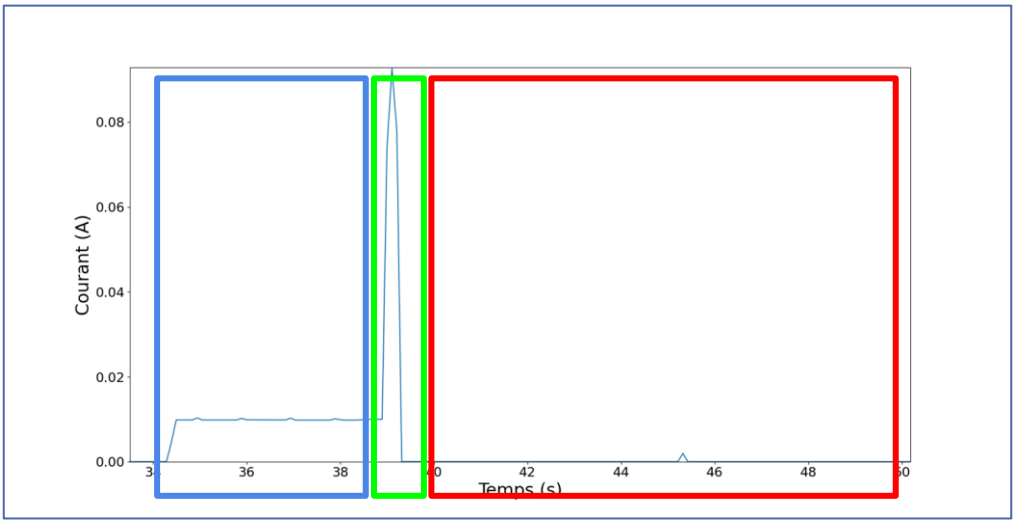

Figure 4: Mesure sur 1 cycle de la consommation

Comparé aux deux autres phases (1 seconde ou 0,5 seconde), c'est bien plus long. Il est donc important que durant cette phase, ou le Dragino LSN50-V2 est inactif, il consomme le moins possible. Sur cette mesure et cette configuration, c'est le cas. En effet, sur la figure 5 qui report la consommation des trois phases, la veille consomme 5µAh et utilise 11% de l'énergie consommée sur 1 cycle. C'est une consommation raisonnable donc lors des manipulations du firmware il ne faudrait pas augmenter cette valeur.

hypothèse :					
nombre de sondes	4 sondes		autonomie:	jour	1199,5
temps de someille	20 minutes			année	3,29
capacité de la batterie	4000 mAh				
	durée (ms)	consomation (mA)	énergie sur 1 cycle (mAh)	Pourcentage	
vielle	1200000	0,015	0,0050	11%	
acquisition	4500	11	0,0138	30%	
émission	1000	100	0,0278	60%	
énergie total sur 1cycle	0,05 W				
énergie total par jour	3,33 W				

Figure 5: Calcul de l'autonomie de la batterie sans l'alimentation des sondes par une entrée-sortie

Comment optimiser la consommation

La mise en place de l'alimentation des sondes par une entrée-sortie a entraîné une surconsommation du microcontrôleur, nous allons voir pourquoi. Premièrement, je vais décrire ci-dessous, le déroulement des actions lors d'un cycle de mesure :

1. Réveil du microcontrôleur
2. Alimentation des sondes
3. Demande de mesure et récupération des données des sondes
4. Coupure de l'alimentation des sondes
5. Envoie de la trame LoRaWAN
6. Activation du Stop mode
7. RX1 & RX2

Ces actions sont affichées figure 6, pour 1 cycle de mesure.

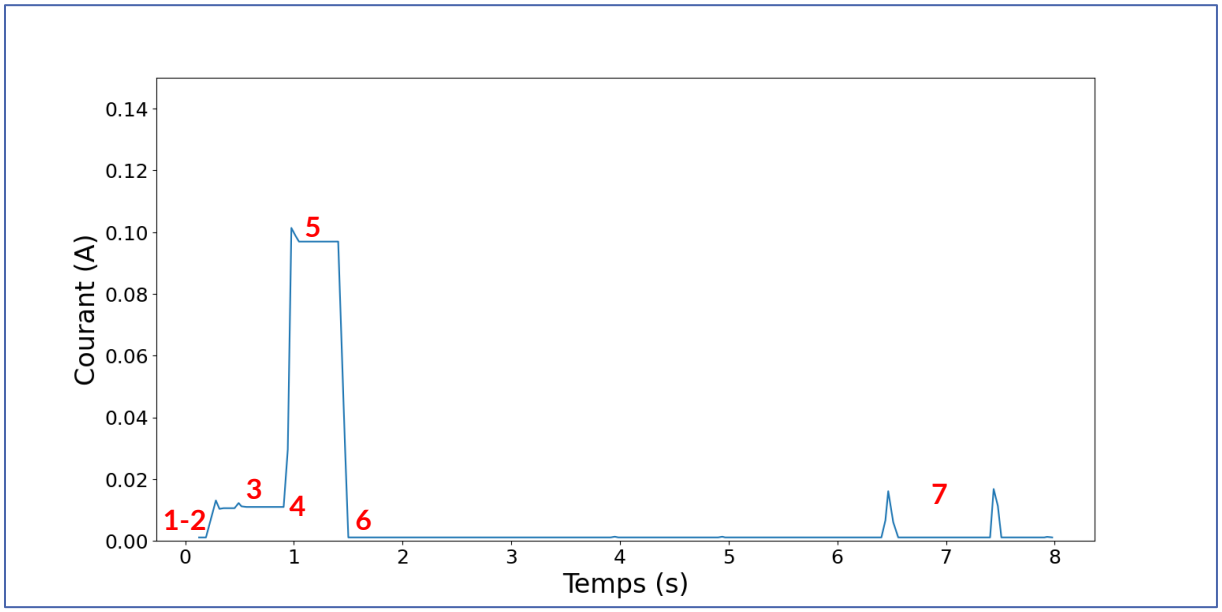

Figure 6: Actions durant 1 cycle de mesure

Dans cette configuration-là, il y a donc une surconsommation. Notamment, lors de l'étape 6, entrée dans le stop mode, où le microcontôleur ne se met pas en deep-sleep. On peut le remarquer sur la figure 7, qui montre la consommation durant cette phase, qui le Dragino consomme alors 1mA en moyenne. La datasheet du Dragino LSN50-V2 nous indique que normalement, il devrait consommer au alentour de 2,7µA. On consomme ici environ 500 fois plus, ce qui confirme que la surconsommation provient de la phase de veille.

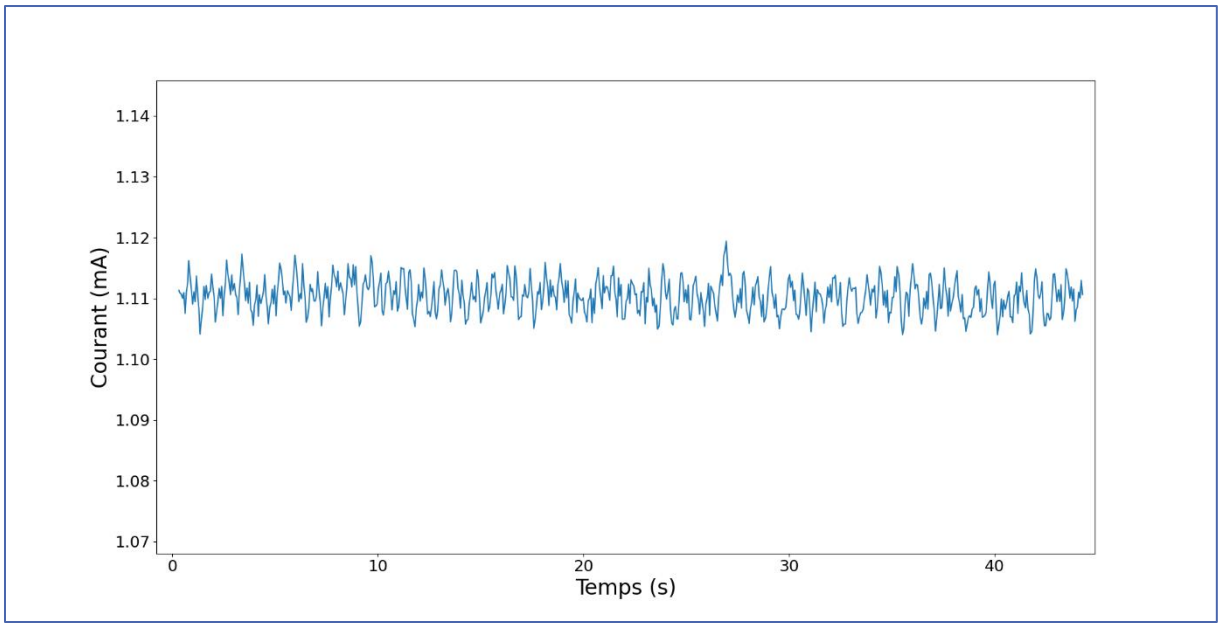

Figure 7 : Consommation durant le stop mode en surconsommation

En recherchant d'où vient le fait que le microcontrôleur n'aille pas en vielle profonde, on se rend compte que le fait de couper l'alimentation des sondes lors du stop mode empêche le microcontrôleur de s'endormir. En effet, lorsque nous enlevons cette fonctionnalité, nous retrouvons une consommation normale, avec un graphique comme sur la figure 8.

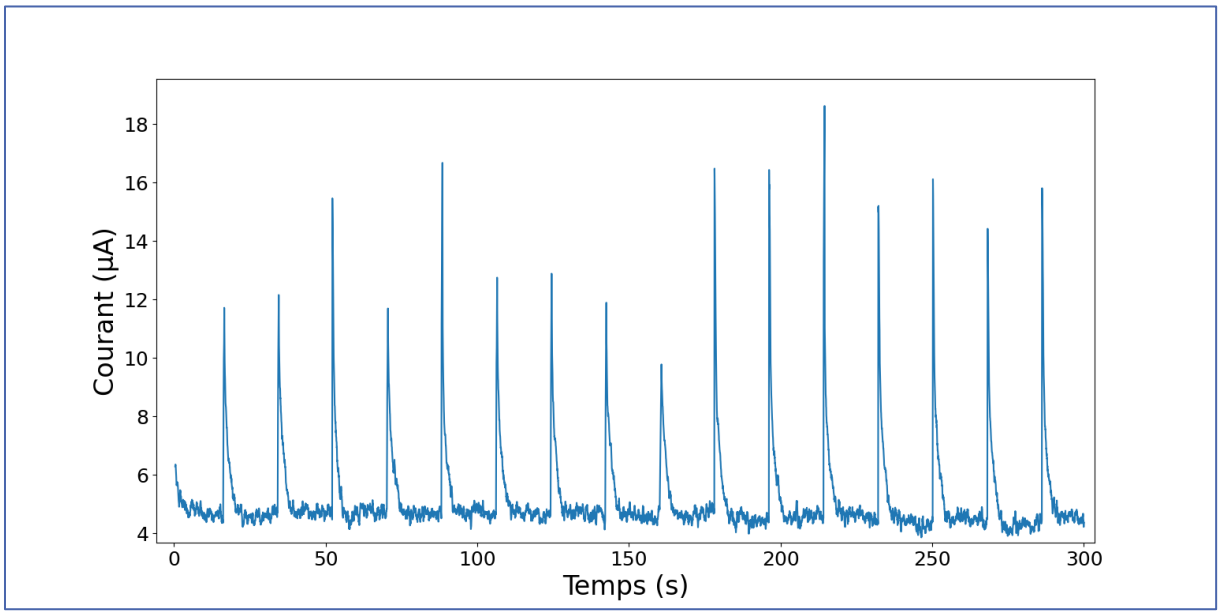

Figure 8: Consommation normal en stop mode

Sur cette figure 8, on observe un pique de courant toutes les 18 secondes, en effet, le stop mode du microcontrôleur ne met pas toutes ses fonctions en arrêt, il conserve notamment les données de la RAM et l'horloge temps réel et certain périphérique

peuvent activer l'oscillateur interne du microcontrôleur pour regarder s'il y a des conditions de réveil active. Ce pique de courant peut avoir comme origine cette dernière.

Donc, nous voulons garder l'alimentation des sondes par une entrée-sortie du microcontrôleur, et garder aussi le fait de reset les sondes avec une coupure de l'alimentation. Il faut revoir le déroulement des actions sur un cycle :

1. Réveil du microcontrôleur
2. Coupure de l'alimentation des sondes + délai de 50 ms
3. Alimentation des sondes
4. Demande de mesure et récupération des données des sondes
5. Envoie de la trame LoRaWAN
6. Activation du Stop mode
7. RX1 & RX2

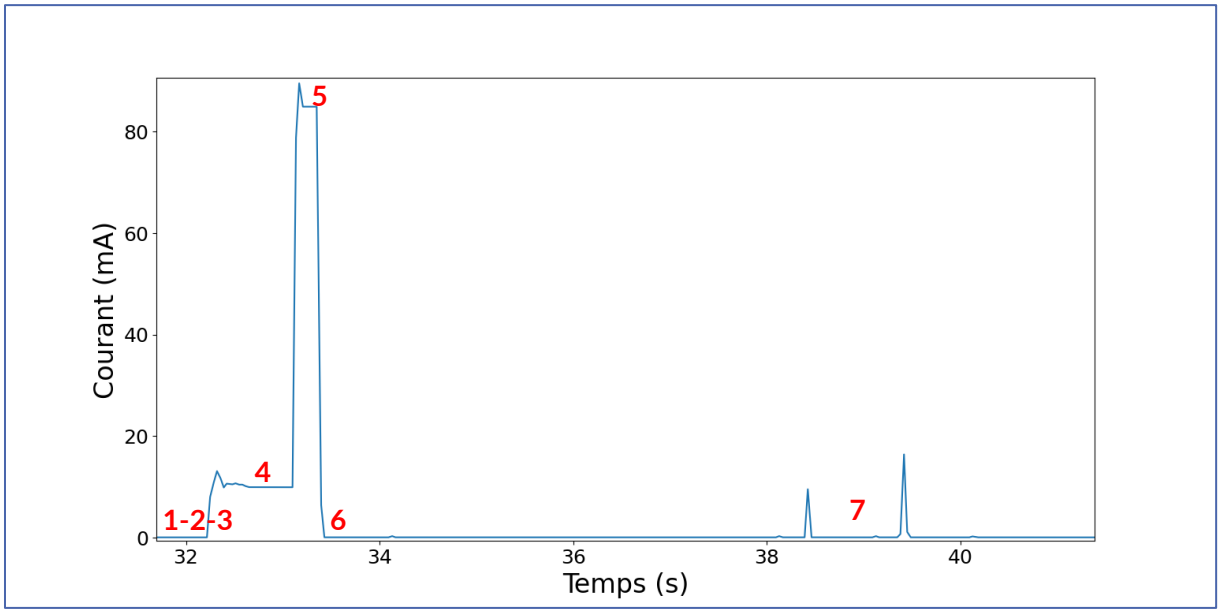

Figure 9 : Modification des actions pendant 1 cycle

Dans cette configuration, certes, la consommation est un peu plus élevée, car les sondes sont alimentées en stop mode, mais cela reste moindre. En effet, la datasheet de la sonde donne une consommation inférieure à 1µA et qui, en réalité, impact peu la consommation. D'autant plus que cela permet au microcontrôleur de rentrer en stop mode, ce qui permet de réduire au maximum la consommation du système.

Finalement, nous réussissons à obtenir la consommation et donc l'autonomie suivante, présentée en figure 10 :

hypothèse					
nombre de sonde	4 sondes		autonomie:	jour	5266,8
temps de someille	20 minutes			année	14,43
capacité de la batterie	4000 mAh				

	durée (ms)	consomation (mA)	énergie sur 1 cycle (mAh)	Pourcentage
vielle	1200000	0,00514	0,0017	16%
acquisition	890	9,78	0,0024	23%
émission	380	60,9	0,0064	61%

énergie total sur 1cycle	0,01 W
énergie total par jour	0,76 W

Figure 10: autonomie de la batterie après optimisation

Conclusion

La consommation du Dragino LSN50 est une notion à ne pas négliger lors du développement d'une application à partir de celui-ci. Tout simplement car ce système sera voué à être utilisé sur batterie durant à temps allant jusqu'à plusieurs années. C'est pourquoi ce travail a été fait, montrer comment optimiser la consommation du microcontrôleur du Dragino LSN50, surtout pendant la phase de sommeil profond, car c'est ici que nous avons eu des problèmes. La prochaine étape pour optimiser encore plus cette consommation serait de regarder au niveau de la communication avec les sondes, c'est-à-dire envoyer une demande de mesure à toutes les sondes l'une après l'autre puis faire pareil pour récupérer les donnes. Aujourd'hui tout cela est fait sonde par sonde et ce serait plus rapide, donc moins de consommation, avec cette modification.