

Plan de Validation

Projet : p2208, Développement d'un Capteur IoT LoraWAN

Table des matières

Cahier des charges.....	2
Prix	3
Consommation.....	5
Correction des mesures	7
Nombre de sondes.....	8

Cahier des charges

Item	Spécification	Vérification
Prix	Système (Dragino + boîtier + main-d'œuvre) \leq 200 € HT	Étude de prix des composants du système ainsi que le coût de la main-d'œuvre
Système étanche	IP x7	Test d'Immersion dans l'eau pendant 30 minutes
Consommation	Système actif 3 ans pour une mesure toutes les 20 minutes et avec 4 sondes connectées	Calcul de la consommation et estimation de la durée de vie de la batterie
Alimentation des sondes	Les sondes sont alimentées avec une entrée-sortie du système	Test de la solution avec le Vcc du bus I2C sur une entrée-sortie
Correction des mesures	Possibilité de renseigner la loi de correction linéaire pour le taux d'humidité mesuré	Gain et offset reportés sur les mesures envoyées
Nombre de sondes	10	Test de la solution avec 10 sondes sur le bus I2C

Prix

La validation du prix du système est faite à partir d'une nomenclature des composants achetés pour le projet, ce sera donc une référence ; s'ajoute à cela le calcul de la main-d'œuvre qui est estimé à partir de notre expérience lors de ce projet ; puis le calcul de la résine utilisée et donc de la quantité utilisée.

Pour rappel, le cahier des charges demande un prix contenant le dragino, le boîtier et la main d'œuvre inférieur à 200 €.

Ainsi, ci-joint dans les documents de ce projet, il y a la nomenclature de celui-ci, avec le détail des prix :

- Premièrement, le coût de la résine utilisée, basé sur le prix total et du poids total et du poids estimé de résine utilisée pour couvrir la carte.

calcul du coût de résine	
système= 1 carte bus le bus I2C + résine	
Prix de la résine	29,44 €
Poids de résine utiliser	20 g
Prix utiliser	1,18 €
site utiliser=	https://www.epodex.com/fr/mediatheque/calculateur-de-resine-epoxy/
dimension=	30x30x20 (mm)

- Il y a aussi le prix de la carte PCB pour le bus I2C, basé sur l'estimation du prix sur le site JLCPCB.
- Il y a le coût de la main-d'œuvre qui est calculé sur la base d'un salaire de technicien en France

calcul du coût de mise en service		
système= dragino + 4 sondes + 1 carte bus le bus I2C + résine		
taux horaire technicien 14,33 €		
action	temps (min)	coût
manipulation sur les sondes (capuchon, fils)	120	28,66 €
flash du code sur le dragino	5	1,19 €
Paramétrage du dragino	15	3,58 €
tirage de la carte	30	7,17 €
soudage	10	2,39 €
impression du moule	20	4,78 €
coulage de la résine	30	7,17 €
TOTAL	230 minutes	54,93 €

- Et, la nomenclature finale qui est la totalité des éléments que nous avons acheté et utilisé pour la solution final

Nomenclature du projet				
système= dragino + 1 carte bus le bus I2C + résine				
Produit	Quantité	Prix (HT)	Source	Total
Dragino	1	56,97 €	lien	56,97 €
carte I2C	1	4,00 €	lien	4,00 €
résine	1	1,18 €	lien	1,18 €
moule	1	1,90 €	lien	1,90 €
borniers	5	1,08 €	lien	5,40 €
main d'oeuvre	1	54,93 €	lien	54,93 €
TOTAL =				124,38 €

Ainsi, on valide cette partie du cahier des charges avec un total de 124,38€ HT, ce qui est inférieur aux 200 € fixée pour ce projet.

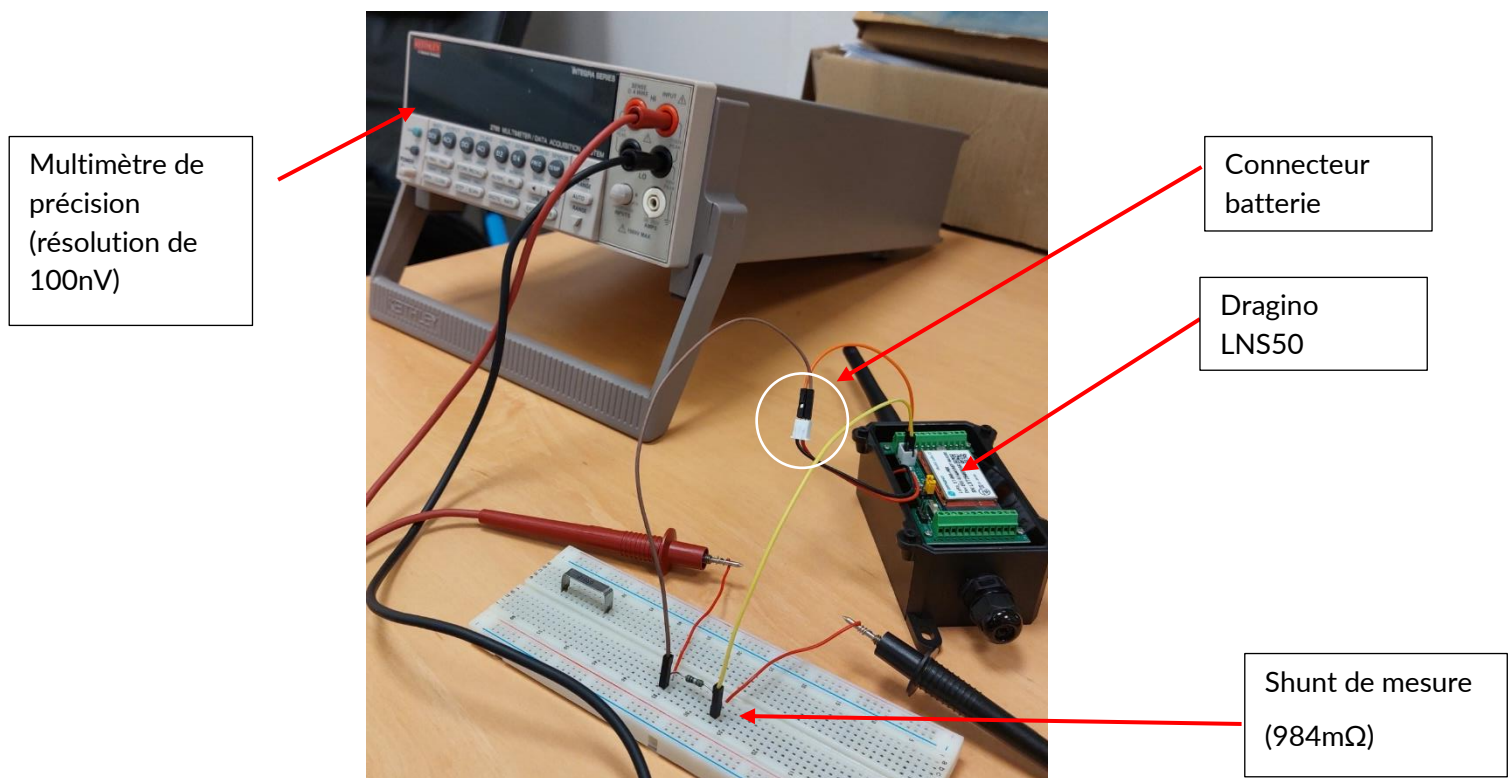
Consommation

Rappel : Au niveau de la consommation du système, nous voulons qu'il puisse faire des mesures toutes les 20 minutes pendant 3 ans.

La batterie la plus contraignante qu'il est possible d'avoir sur un dragino LSN50 est celle de 4000mAh. De plus, nous allons utiliser un classeur Excel où seront les calculs d'autonomie pour valider le cahier des charges.

Pour faire cette estimation, il faut mesurer quelques valeurs sur la consommation du dragino : le temps d'acquisition des sondes, le courant d'acquisition, le temps de sommeil et le courant de sommeil. Pour cela on va devoir mesurer sur un cycle la consommation du dragino.

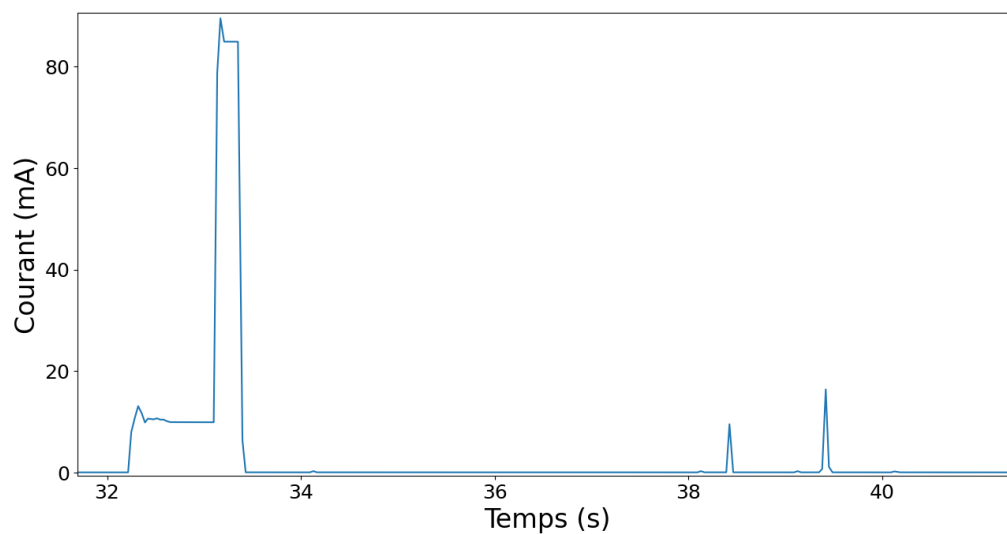
Mesure :



Le système ci-dessus permet de mesurer précisément la tension du shunt et ainsi de pouvoir avoir la consommation du dragino sur ses phases de consommation (veille, mesure, envoie).

Le multimètre possède une prise GPIB qui permet d'interfacer le multimètre avec son ordinateur et d'avoir un scripte permettant d'enregistrer les valeurs modifier et de les afficher sous forme de tableau. (Ce scripte se trouve en annexe de ce document).

On obtient une courbe comme ci-dessous. Cela correspond à 1 cycle de consommation.



A partir de ces mesures on peut calculer les consommations dans toutes les phases du Dragino et créer l'Excel ci-dessous

Consommation du système					
hyp:					
nombre	4 sondes	autonomie:	jour	5266,8	
temps de sommeil	20 minutes		année	14,43	
capacité de la batterie	4000 mAh				
	durée (ms)	consommation (mA)	énergie sur 1 cycle (mAh)	Pourcentage	
veille	1200000	0,00514	0,0017	16%	
acquisition	890	9,78	0,0024	23%	
émission	380	60,9	0,0064	61%	
énergie total sur 1 cycle	0,01 W				
énergie total par jour	0,76 W				

Ainsi, on peut valider sur la consommation du Dragino, avec une autonomie prévue de 14 année.

Correction des mesures

Rappel : Il doit y avoir la possibilité d'appliquer une correction linéaire sur la mesure de l'humidité, directement sur le Dragino donc un gain et un offset.

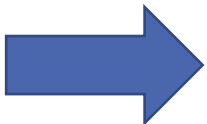
L'intégration a été faite dans le firmware du Dragino et nous allons devoir valider qu'il y a bien un impact dans la valeur envoyé en LoRaWAN. Pour faire ceci, nous allons mettre des gains et offsets simple et allons observer les modifications :

Sonde 1 : GAIN = 0 ; OFFSET = 2

Sonde 2 : GAIN = 1 ; OFFSET = -10

Sonde 3 : GAIN = 1 ; OFFSET = 10

Sonde 4 : GAIN = 2 ; OFFSET = 0

"Hum_HYT_0" : 53.5,		"Hum_HYT_0" : 2,
"Hum_HYT_1" : 53.5,		"Hum_HYT_1" : 40.3,
"Hum_HYT_2" : 52.9,		"Hum_HYT_2" : 60.5,
"Hum_HYT_3" : 53.4,		"Hum_HYT_3" : 100,
"TempC1" : 3276.7,		"TempC1" : 3276.7,
"Temp_HYT_0" : 19.3,		"Temp_HYT_0" : 18.9,
"Temp_HYT_1" : 19.3,		"Temp_HYT_1" : 19.2,
"Temp_HYT_2" : 19.3,		"Temp_HYT_2" : 19,
"Temp_HYT_3" : 19.1		"Temp_HYT_3" : 19

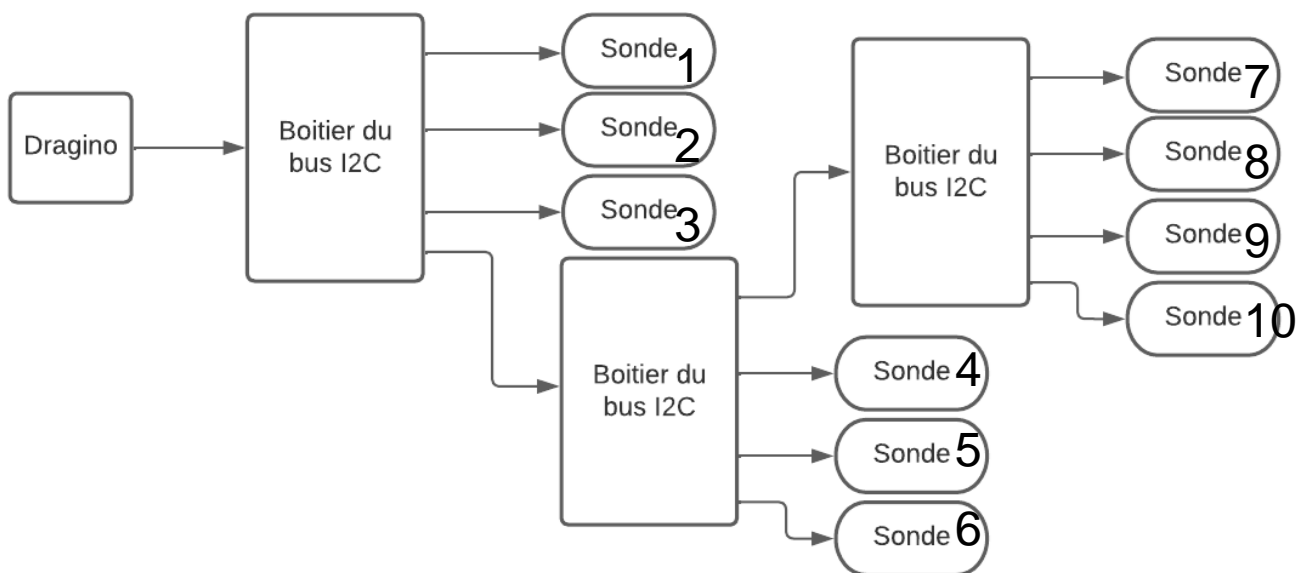
A gauche se trouve les valeurs de référence, prises juste avant de faire la modification via commande AT sur le gain et l'offset de chaque sonde. On remarque bien que la correction linéaire de chacune des sondes est bien reportée sur la valeur envoyée sur le réseau LoRaWAN.

Nombre de sondes

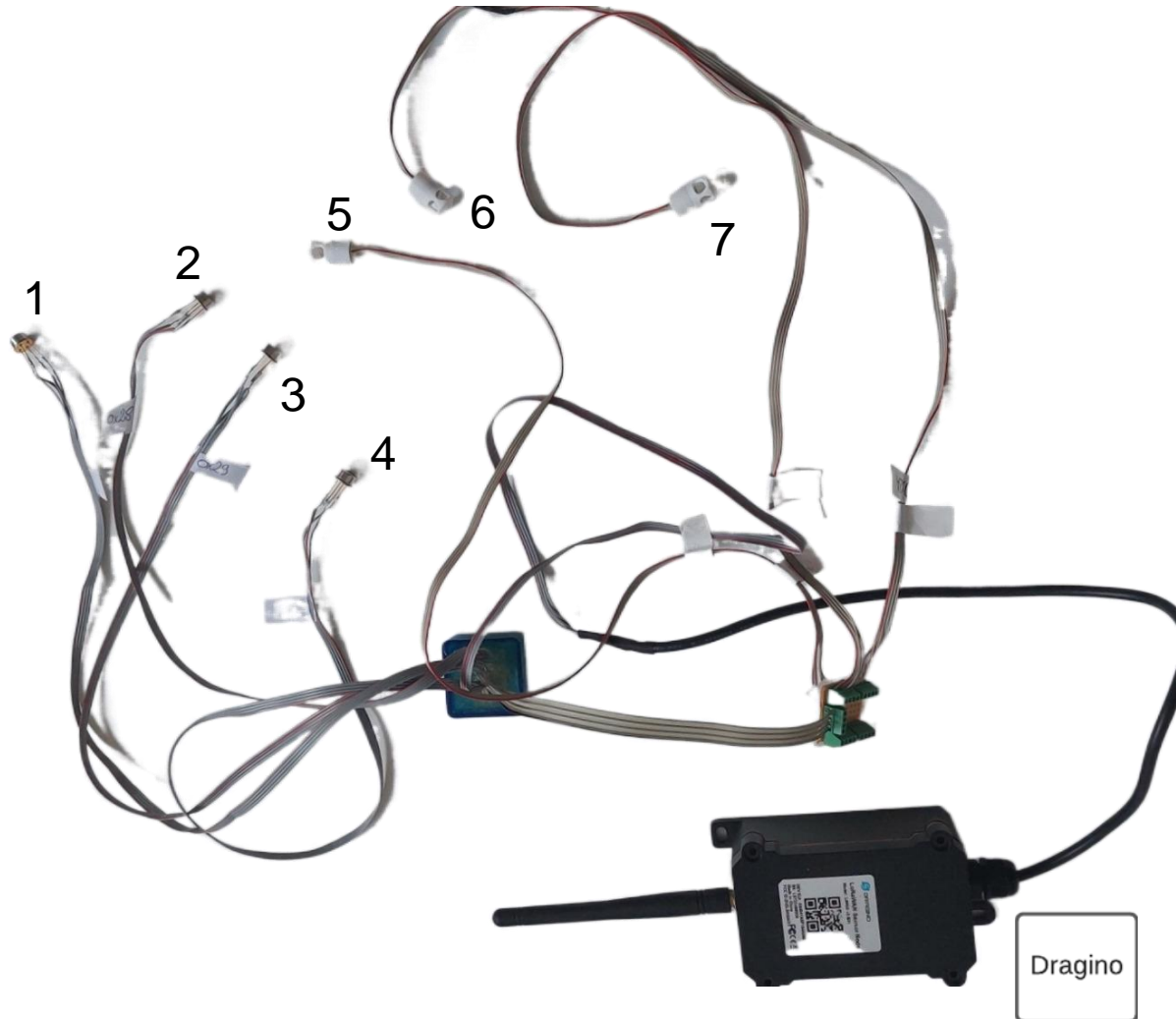
Rappel : on veut utiliser le Dragino avec 10 sondes sur le bus I2C.

Le firmware a été construit pour pouvoir intégrer 10 sondes sur le bus I2C du Dragino. Et en parallèle, nous avons fait un boîtier I2C avec la possibilité d'interfacer 4 sondes sur le bus I2C du Dragino.

Voici un schéma des connexions entre le dragino et les différentes sondes.



Nous allons faire ce test avec 7 sondes sur le bus au lieu de 10, n'ayant pas 10 sondes en possession.



En branchant tout cela au système et en renseignant les valeurs correctes de sondes branchées, dans notre cas, nous devons préciser qu'il y a 7 sondes connectées au système, nous devons faire la mesure et renvoyer les informations pour toute les sondes.

Nous obtenons les valeurs ci-dessous :

```
"Hum_HYT_0": 24.1,  
"Hum_HYT_1": 23.9,  
"Hum_HYT_2": 23.4,  
"Hum_HYT_3": 24.6,  
"Hum_HYT_4": 24,  
"Hum_HYT_5": 25.1,  
"Hum_HYT_6": 24.5,  
"TempC1": 3276.7,  
"Temp_HYT_0": 21.7,  
"Temp_HYT_1": 21.8,  
"Temp_HYT_2": 22.5,  
"Temp_HYT_3": 21.8,  
"Temp_HYT_4": 22.3,  
"Temp_HYT_5": 21.6,  
"Temp_HYT_6": 21.9
```

On a bien le résultat pour 7 sondes différentes, de part leurs valeurs différentes et par leur nom (ce qui veut dire qu'elles sont bien différenciées dans le payload).