



Rapport de fin de projet

Nom du projet : IoT Benchmark

Numéro d'équipe : 14

Partenaire : *M. FAUBERTEAU Frédéric*

Membres de l'équipe projet :

- AURIAC Julian
- BONNET Vincent
- BORNON Théophile
- BOUVET Xavier
- PRÉVOT Tristan

Table des matières

1	OBJECTIFS DE CE RAPPORT	2
2	CONTEXTE.....	2
3	DESCRIPTION DU PROJET.....	3
4	OBJECTIFS INITIAUX ET RESULTATS ATTEINTS	4
4.1	Objectifs initiaux.....	4
4.2	Résultats atteints	5
4.3	Liste des livrables	6
4.3.1	Prototype	6
4.3.2	Modèle 3D d'un boîtier	7
4.3.3	Schéma électrique	7
5	REVUE TECHNIQUE	8
5.1	Hardware.....	8
5.1.1	Mesure du courant.....	8
5.1.2	Switching	9
5.1.3	Convertisseurs analogiques numériques.....	10
5.1.4	Switching automatique.....	10
5.1.5	Schéma final	12
5.2	Software	13
6	RESSOURCES PREVUES / RESSOURCES UTILISEES.....	14
7	REVUE METHODOLOGIQUE	15
8	GESTION DU RISQUE	16
9	TACHES A REALISER POST-PROJET	16
10	SATISFACTION DU CLIENT ET DE L'UTILISATEUR	17
11	RECOMMANDATION POUR COMPLETER LE PROJET	17

1 Objectifs de ce rapport

Le rapport de clôture du projet contient des renseignements descriptifs clés sur le projet. Comme dernier document écrit sur le projet, il décrit la relation entre les livrables produits et les objectifs fixés. Il analyse le résultat du projet et le processus par lequel ce résultat a été produit. Son objectif est double : assurer la bonne exécution des activités de fermeture et faciliter le transfert d'expérience pour un éventuel projet de suivi.

2 Contexte

Le rapport de clôture du projet est produit par l'extraction des données déjà produites tout au long du projet, notamment le compte-rendu de mi-projet, les comptes-rendus de réunions hebdomadaires et mensuelles, et les détails des avancées théoriques et concrètes réalisées.

La valeur ajoutée consiste dans l'agrégation des données dans un seul document, l'analyse des différences globales, et les indications et conseils à l'intention d'un potentiel groupe de projet voulant utiliser les livrables produites.

3 Description du projet

L'objectif de ce projet est de concevoir un banc d'essai pour évaluer les technologies réseau IoT en termes de consommation énergétique.

L'idée sous-jacente est de pouvoir accompagner les industriels à décider de la technologie la plus adaptée en fonction de leur besoin (e.g. LoRaWAN, NB-IoT, ZigBee, BLE...).

L'équipe est composée de Julian, Théophile, Tristan, Vincent et Xavier et décomposée en deux groupes.

Un groupe chargé de la partie hardware composé de Théophile et Vincent et un groupe chargé de la partie software composé de Julian, Tristan et Xavier.

Julian était également responsable des livrables et Théophile référent du projet.

M. Fauberteau est à la fois le client qui a commandé ce projet et l'utilisateur final du projet. M. Zanette est notre PRM qui nous a aidé pour la partie technique en électronique. Nous avons également reçu de l'aide de la part de M. Peretti pour l'électronique.

Ce projet s'est articulé autour des réunions bimensuelles entre le groupe et le partenaire. Lors de la première réunion, l'objectif a été de mettre en place un cahier des charges le plus détaillé possible pour la suite du projet. Lors des réunions suivantes, nous présentions notre avancement à M. Fauberteau afin de s'assurer que le projet allait dans la bonne direction.

4 Objectifs initiaux et résultats atteints

4.1 Objectifs initiaux

L'objectif de ce projet était de concevoir un banc d'essai pour évaluer les technologies réseau IoT en termes de consommation énergétique.

Le banc d'essai devait être économique et simple à mettre en place.

Il faut également concevoir des scénarios représentant des cas d'usage. Par exemple, un scénario pourrait être de mesurer la température de toutes les pièces d'un bâtiment et les envoyer vers un cloud pour réaliser un tableau de bord.

Une fois qu'un ensemble de scénarios est identifié, il faut identifier les différents composants qui seront mis en œuvre dans ces scénarios. Par exemple, dans le cas de LoRaWAN, il y a les capteurs, les puces LoRa, les passerelles LoRa... Quid du cloud.

Une fois qu'un scénario est bien défini, il faut mettre en place un protocole expérimental pour comparer la consommation énergétique induite par l'utilisation des différentes technologies. En d'autres termes : quelles est l'énergie consommée pour récupérer les températures des pièces d'un bâtiment avec un réseau LoRaWAN ? ZigBee ? Cela implique d'identifier les appareils de mesure, mais aussi les composants matériels et logiciels pour tester le scénario. Un des points importants est la mise en œuvre d'une plate-forme matérielle générique permettant de pouvoir interchanger les différents composants électroniques (LoRa, BLE...) connectés aux outils de mesure, et ce, afin de pouvoir faciliter la reproduction des campagnes de test.

4.2 Résultats atteints

Malheureusement, nous n'avons pas pu atteindre tous les objectifs que nous nous étions fixés pour répondre au besoin.

Le premier objectif était de réaliser un prototype de banc d'essai de mesure afin des tester les différents protocoles de communication IoT. Cependant cet objectif n'a pas été atteint. Nous n'avons pu produire qu'un prototype sur une breadboard ce qui est loin de nos ambition. La raison principale est le manque de temps, en effet, nous avons commandé le matériel début décembre mais en raison des conditions sanitaires nous ne l'avons reçu que début janvier. De plus, le matériel reçu n'était pas complet ce qui nous a empêché de réaliser certaines parties importantes du circuit comme par exemple le switching automatique.

Certains objectifs ont également été revus ou supprimés suite aux échanges avec le commanditaire. En effet, au début il était prévu de mettre en place des scénarios de test pour comparer les différents protocoles entre eux mais nous nous sommes rapidement rendu compte que ce serait trop compliqué à mettre en place s'il ne fallait omettre aucune donnée.

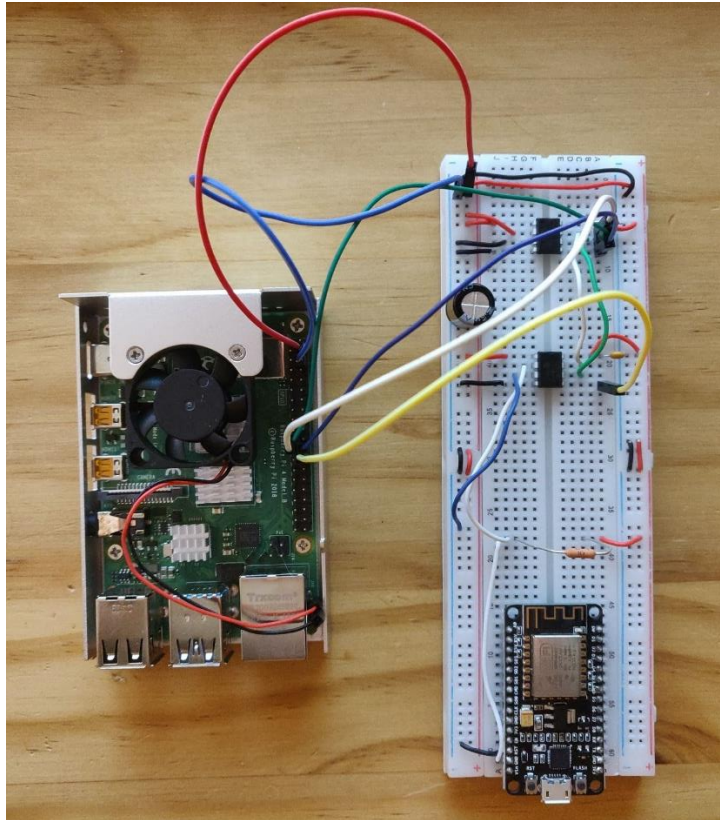
Prenons par exemple un scénario permettant de comparer le BLE au WiFi. La version la plus simpliste serait de mesurer uniquement la consommation du module BLE et WiFi en ignorant les gateways. Or pour obtenir des résultats concrets, il nous paraissait inconcevable d'ignorer celles-ci. Il aurait donc fallu ajouter la mesure de la consommation des gateways. Ensuite, dans le cas où les données doivent être envoyées dans le cloud, dans le cas du WiFi, les données peuvent directement être envoyées à travers Internet mais dans le cas du BLE, il faut ajouter un device pour envoyer ces données sur Internet et donc quid de la consommation de ce service.

Pour ces raisons, nous avons décidé d'abandonner la rédaction de scénarios.

Pour finir, nous avons réussi à produire un schéma électrique répondant à toutes nos contraintes. Ce schéma n'a pas pu être validé car nous n'avons pas eu le temps et le matériel nécessaire pour le tester.

4.3 Liste des livrables

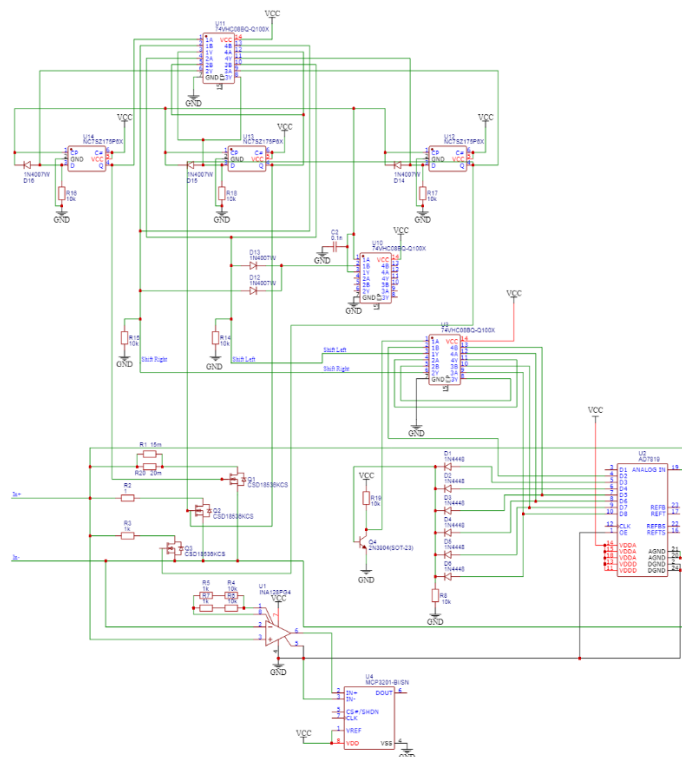
4.3.1 Prototype



4.3.2 Modèle 3D d'un boîtier



4.3.3 Schéma électrique



5 Revue technique

Comme expliqué plus haut, nous avons découpé notre équipe en deux sous-groupes. Un premier qui se chargeait de la partie hardware et un autre qui se chargeait de la partie software.

Nous présenterons dans un premier temps les choix fait sur la partie hardware puis sur la partie software.

5.1 Hardware

5.1.1 Mesure du courant

Pour rappel, notre objectif est de mesurer la consommation d'objets connectés dont la consommation peut varier du μA lorsque l'objet est en deep-sleep à une ou plusieurs centaines de mA lorsque l'objet est en activité.

La première étape a donc été de déterminer quelle serait la manière la plus adaptée pour mesurer le courant.

Une première idée a été d'utiliser le principe de la pince ampèremétrique pour mesurer le courant mais cela s'est révélé impossible étant donné la faible intensité à mesurer.

Nous avons donc choisi de nous diriger vers un ensemble de résistances « shunt »¹. Le principe du « shunt » est de pouvoir mesurer la tension qui traverse un dispositif de faible impédance pour ensuite déterminer à l'aide de la loi d'Ohm le courant qui traverse ce même dispositif.

Ce choix nous a également permis d'ajouter des contraintes au projet :

- Pour éviter que la chute de tension aux bornes de la résistance soit trop élevée et engendre des problèmes d'alimentation du composant IoT. Nous avons donc choisi, de manière arbitraire, que la chute de tension maximale tolérée serait de 100mV.
- Nous ne mesurerons pas de courant dépassant 10A.

Grâce à ces contraintes, nous avons estimé que 3 résistances seraient nécessaires pour mesurer correctement l'intensité :

- 1k Ω pour les intensités allant de 0,1 μA à 0,1mA
- 1 Ω pour les intensités allant de 0,1mA à 0,1A
- 10 m Ω pour les intensités allant de 0,1A à 10A

¹ [https://fr.wikipedia.org/wiki/Shunt_\(%C3%A9lectrotechnique\)#Principe](https://fr.wikipedia.org/wiki/Shunt_(%C3%A9lectrotechnique)#Principe)

Afin de faciliter la lecture de la tension mesurée aux bornes des résistances, nous avons décidé d'appliquer un gain de 10 au signal.

Dans un premier temps, nous avons décidé d'utiliser un amplificateur opérationnel MAX4239 car nous l'avions vu utilisé dans un autre projet² & ³ de mesure du courant lors de l'état de l'art. Néanmoins, il nous a été conseillé de le remplacer par un amplificateur d'instrumentation plus précis : l'INA128P.

5.1.2 Switching

La seconde étape de notre projet a été de déterminer comment se ferait le passage d'une résistance « shunt » à une autre.

Pour sélectionner la résistance utilisée, nous avons choisi d'utiliser un MOSFET en fonctionnement saturé. Au départ nous avons pensé utiliser un transistor BJT mais ces transistors induisent une chute de tension entre les bornes ($V_{CE_{SAT}}$) d'à minima 600mV ce qui était inenvisageable.

Il faut noter que bien que les MOSFET n'entraînent pas une chute de tension aussi importante que les transistors BJT, ils ont malgré tout une valeur $R_{ds(on)}$ (Drain-to-Source On-Resistance) qui induit une chute de tension. Il était donc important pour nous de choisir une référence adaptée à notre projet avec une faible valeur $R_{ds(on)}$: le CSD18536KCS. Son $R_{ds(on)}$ est de 1.7mΩ à 4.5V.

Pour s'assurer que la résistance servant à mesurer l'intensité de 0,1A à 10A soit bien de 10mΩ, nous avons dû remplacer la résistance par 2 résistances de respectivement 15mΩ et 20mΩ en parallèle. De cette manière la résistance totale du circuit était d'environ 10mΩ.

$$\frac{1}{\frac{1}{15} + \frac{1}{20}} + 1.7 \approx 10.27m\Omega$$

² <https://www.eevblog.com/projects/ucurrent/>

³ <https://www.eevblog.com/files/uCurrentRev5schematic.pdf>

5.1.3 Convertisseurs analogiques numériques

Pour la première itération, nous avons choisi d'utiliser deux types de CAN.

- 1 CAN 8bits à sorties parallèles. Ce CAN permet de faire le switch entre les différentes résistances « shunt ». Nous avons choisi la référence AD7819. Dans le cas où la sélection de la résistance se fait avec un SPLD ou un MCU, ce CAN serait inutile.
- 1 CAN 12bits avec une sortie SPI. Ce CAN permet de relever la tension pour ensuite la communiquer au Raspberry Pi en SPI. Nous avons choisi la référence MCP3201.

Il est à noter que ces CAN ont été choisis pour le prototype uniquement. Dans le futur du projet nous souhaiterions remplacer le CAN 12bits par un CAN avec une meilleure résolution.

Il peut également être intéressant d'ajouter un CAN aux bornes d'alimentation du dispositif IoT pour permettre une mesure de la tension en parallèle de l'intensité afin d'avoir un calcul de la puissance plus précis.

Il est également possible d'ajouter une diode Zener sur les broches de référence de tension des CAN pour améliorer la précision de la mesure.

5.1.4 Switching automatique

5.1.4.1 Théorie

Nous avons étudié la façon dont le switch pouvait être fait automatiquement. Trois options ont été envisagées.

La première, la plus simple, est d'utiliser un MCU tel que l'Atmega 328P. Le MCU lit la tension en sortie de l'AOP. Si la tension est supérieure à un certain seuil, on passe à la résistance du seuil supérieure et à contrario, si elle est inférieure au seuil, on passe à la résistance du seuil inférieur.

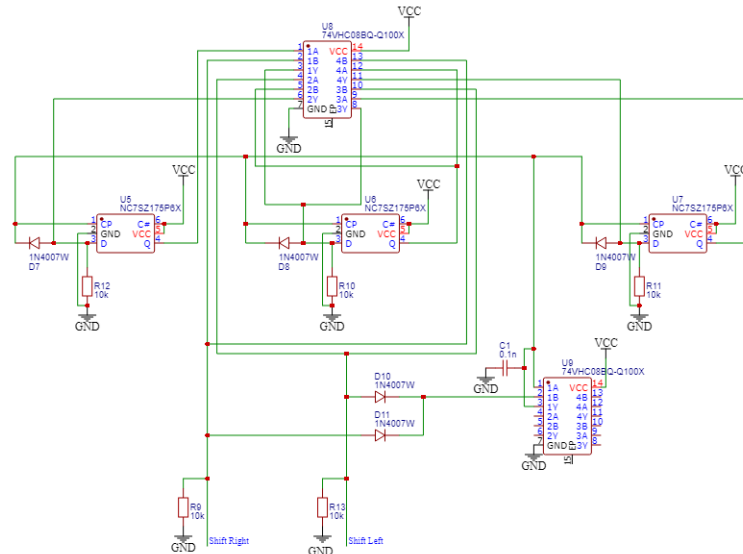
La seconde solution est d'utiliser un SPLD (Simple Programmable Logic Device). Le fonctionnement est le même qu'avec un MCU mais nous pensons que cette solution serait plus efficace car plus proche du hardware. Nous avons eu cette idée en cherchant le fonctionnement de la fonction d'auto ranging des multimètres et plus précisément grâce à cette note d'application⁴.

La dernière solution était d'implémenter directement avec des portes logiques notre fonction de seuil. C'est cette dernière solution que nous avons retenue car nous pensons que c'est celle qui est la plus efficace. Il serait néanmoins intéressant de mettre en place un scénario de test pour déterminer expérimentalement la solution la plus efficace.

⁴http://www.silego.com/uploads/Products/product_543/details/AN-1148%20Voltmeter%20Ammeter%20Autorange%20Control.pdf

5.1.4.2 Implémentation

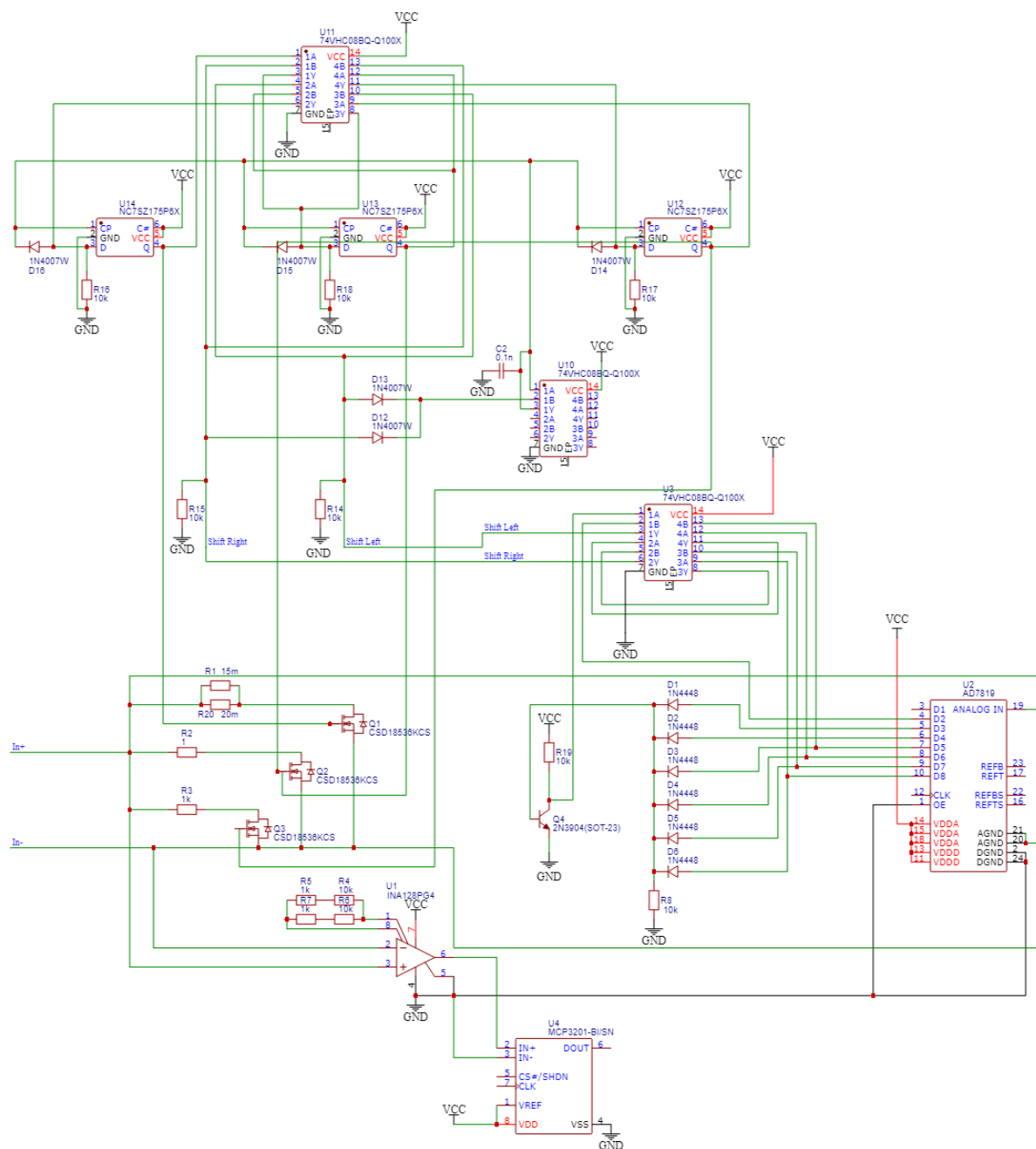
L'implémentation de la fonction de seuil est triviale et visible dans le schéma final. En revanche, nous avons dû réfléchir à la mise en place d'une mémoire pour se souvenir de quelle était la résistance sélectionnée.



Nous avons donc utilisé 3 bascules. Lorsque la ligne « Shift Right » passe de l'état bas à l'état haut, le bit se déplace d'une bascule vers la droite. S'il est déjà dans la bascule la plus à droite, il y reste. Réciproquement lorsque la ligne « Shift Left » passe de l'état bas à haut.

Une protection est également mise en place par le biais d'une porte ET pour éviter que si une ligne est maintenue à l'état haut, le bit passe par exemple de la bascule la plus à droite à la bascule la plus à gauche. Ainsi tant que la ligne ne repasse pas à bas, le bit ne peut pas se déplacer à nouveau.

5.1.5 Schéma final



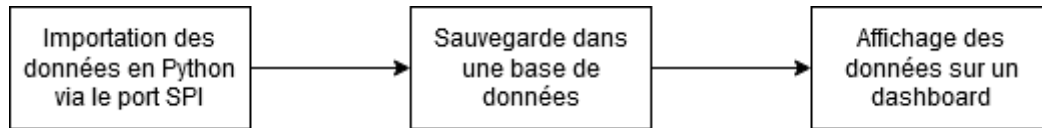
La partie en bas à gauche contient les résistances « shunt », l'amplification du signal et le CAN 12 bits connecté en SPI au Raspberry Pi.

La partie en bas à droite contient le CAN 8 bits connecté aux fonctions de seuil permettant le shifting automatique.

Enfin la partie haute contient la mémoire de la résistance « shunt » actuellement sélectionnée.

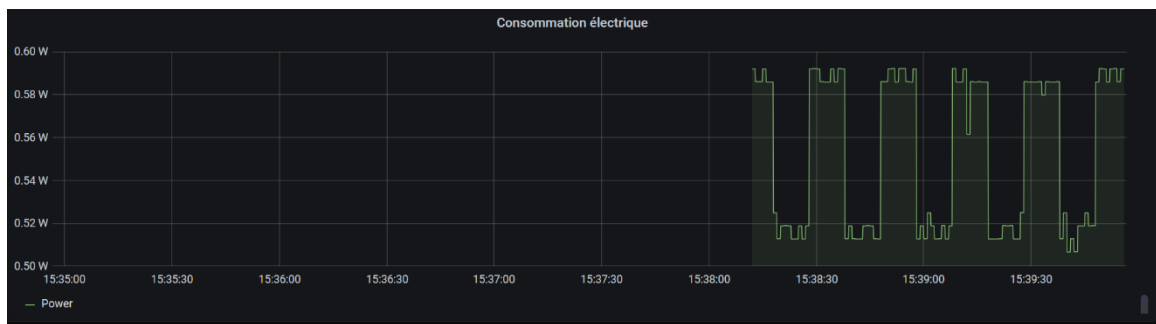
5.2 Software

Nous avons choisi d'utiliser un Raspberry Pi 4 avec une distribution Raspberry Pi OS. Nous avons fait ce choix pour faciliter le prototypage. L'architecture retenue est la suivante :



Pour importer les données nous avons choisi de développer un script en Python. Le script lit les données depuis le port SPI pour importer les valeurs lues dans la base de données InfluxDB. Les données sont ensuite accessibles depuis un dashboard Grafana.

Nous avons choisi InfluxDB et Grafana car ce sont deux solutions open source et que nous avons déjà eu l'occasion de travailler avec ces logiciels. Nous voulions également garder les données en local sur le Raspberry.



6 Ressources prévues / ressources utilisées

Afin de mener à bien ce projet, plusieurs ressources ont été utilisées. Pour la partie communication, une conversation de messagerie instantanée a été mise en place, ce qui permet les échanges d'idées, la planification primaire (là où les planifications plus poussées avaient lieu durant les rendez-vous physiques et visioconférences), et les échanges de codes simples. Pour ce qui est des échanges de codes plus lourds ou plus complexes, nécessitant de la documentation, un groupe Teams a été mis en place pour suivre le travail de chacun, et permet donc les ajustements de planning.

Le gestionnaire d'échange de fichiers Teams a été peu utilisé durant la première partie du projet, et servait plus de bibliographie à consulter lors de nouvelles recherches. Dans un second temps, la centralisation des livrables et des recherches a permis une uniformisation des avancements, et ainsi de donner une idée plus précise de l'avancement du projet.

Quant aux plannings, ils se faisaient à l'échelle des tâches, et ne regroupaient donc que peu souvent l'ensemble du groupe, mais plutôt des sous-groupes, qui ne requerraient pas de ressource particulière. Ces derniers étaient organisés uniquement durant les réunions hebdomadaires. Cette organisation nous a donné une grande flexibilité et nous a donc facilité l'alternance entre les réunions en présentiel et celles en visio-conférence.

Nous avons prévu de présenter un prototype totalement fonctionnel. Mais en cette période, le travail en collaboration fût moins aisé que prévu. De plus, en raison des délais de livraison élevés des composants électroniques, nous nous sommes vus dans l'incapacité de présenter un prototype final comme nous l'avions pensé, et ainsi valider notre schéma électrique théorique. Pour pallier cela, nous aurions pu passer moins de temps sur l'état de l'art et les technologies de communication à tester, pour concevoir et produire notre prototype plus tôt.

7 Revue méthodologique

Les principales caractéristiques de notre organisation ont été de travailler en sous-groupes pour traiter le suivi de l'avancement du projet d'une part et la mise en forme des livrables.

La partage du travail sur des membres du groupe a été réalisé de la manière suivante :

Le suivi global du projet a été réalisé par la planification de réunions mensuelles. Pour mener à bien ce projet nous avons défini des sous-objectifs et des jalons permettant d'atteindre notre objectif final. Les sous-objectifs ont été répartis entre les différents membres de l'équipe et sous-groupes. Chaque sous-groupe était autonome dans son organisation et reportait l'avancement, les difficultés et les risques potentiels à l'ensemble de l'équipe lors de la réunion mensuelle. Cela a permis de réajuster les objectifs au fur et à mesure des besoins et des deadlines.

Exemple de répartition pour la phase de production du prototype :

- Conception et réalisation électronique (hardware)
- Conception et réalisation de la partie software

8 Gestion du risque

Le principal risque auquel nous avons dû faire face durant ce projet était lié à son contexte, et plus précisément à l'acheminement de ses composants, et à la nécessité de regrouper les membres du groupe projet pour les réunions de coordination.

Ici, on retiendra donc que la durée du projet passée sa mise en place initiale était trop courte pour pouvoir réagir d'une manière suffisante aux conditions exceptionnelles. La solution principale retenue ici étant l'émulation des parties logicielles du montage via des machines virtuelles, cette solution s'est rapidement vue dépassée une fois la majeure première partie de recherche menée à terme.

Pour minimiser ce risque, nous avons décidés de déployer le maximum d'environnements possibles, et d'essayer de nouvelles configurations plus efficaces, tout en nous assurant d'avoir des versions plus stables. Cette méthode a eu comme inconvénient de créer des écarts d'avancements importants entre les sous-groupes.

9 Tâches à réaliser post-projet

Le projet n'a pas pu atteindre sa finalité durant ce semestre. Les délais de livraison des pièces électronique ainsi que le travail dans les conditions actuelles ne nous ont pas permis de mettre à terme un prototype entièrement fonctionnel. Afin de produire un prototype comme attendu par le partenaire, il faudra apporter les améliorations suivantes au projet :

- Assembler le montage électrique théorique lorsque les pièces seront disponibles, et ainsi vérifier son bon fonctionnement
- Sauvegarder les données à une fréquence plus élevée sur la base de données.
- Sauvegarder les données sur un serveur externe, elles sont aujourd'hui sauvegardées exclusivement en local.
- Finaliser l'application pour PnP (Plug-and-Play) et donc profiter d'une utilisation user-friendly.
- Tester les différentes technologies de communication et établir un comparatif

10 Satisfaction du client et de l'utilisateur

En attente du résultat de la réunion du 14/01/2021

11 Recommandation pour compléter le projet

Selon nous, la recommandation la plus importante pour compléter ce projet est de réaliser le plus d'itérations possible. Nous avons voulu faire directement une partie théorique fonctionnelle mais cela nous a pris plus de temps que nécessaire. C'est pourquoi nous pensons qu'il aurait été plus intéressant et efficace de faire un maximum d'itérations de prototype quitte à avoir plus de ratés.

La première étape serait d'implémenter toutes les fonctionnalités correctement sur une breadboard. Une fois que ceci est fonctionnel, faire un design sur un PCB pour pouvoir l'imprimer. Ainsi, une grande partie du bruit devrait disparaître des mesures.

Enfin la dernière partie serait un passage de composants discrets à des composants CMS. Ceux-ci sont plus performants et moins chers.

Pour la partie software, il serait intéressant de changer le code en python pour du code en C afin d'avoir de meilleures performances.

Il faudrait également réfléchir à un moyen de stockage externe pour éviter de faire toutes les écritures sur une carte micro sd qui ne supportera sûrement pas longtemps cette charge.

Enfin, étudier quelle est la façon la plus efficace d'accéder aux données. Dans notre cas, elles sont stockées en locales sur le raspberry pi et accessibles sur un dashboard mais il pourrait également être intéressant de stocker dans le cloud.