

Rapport de stage avenant

Alexis Lapeze

BTS SIO 1^{er} année

Institut Limayrac



Sommaire :

Table des matières

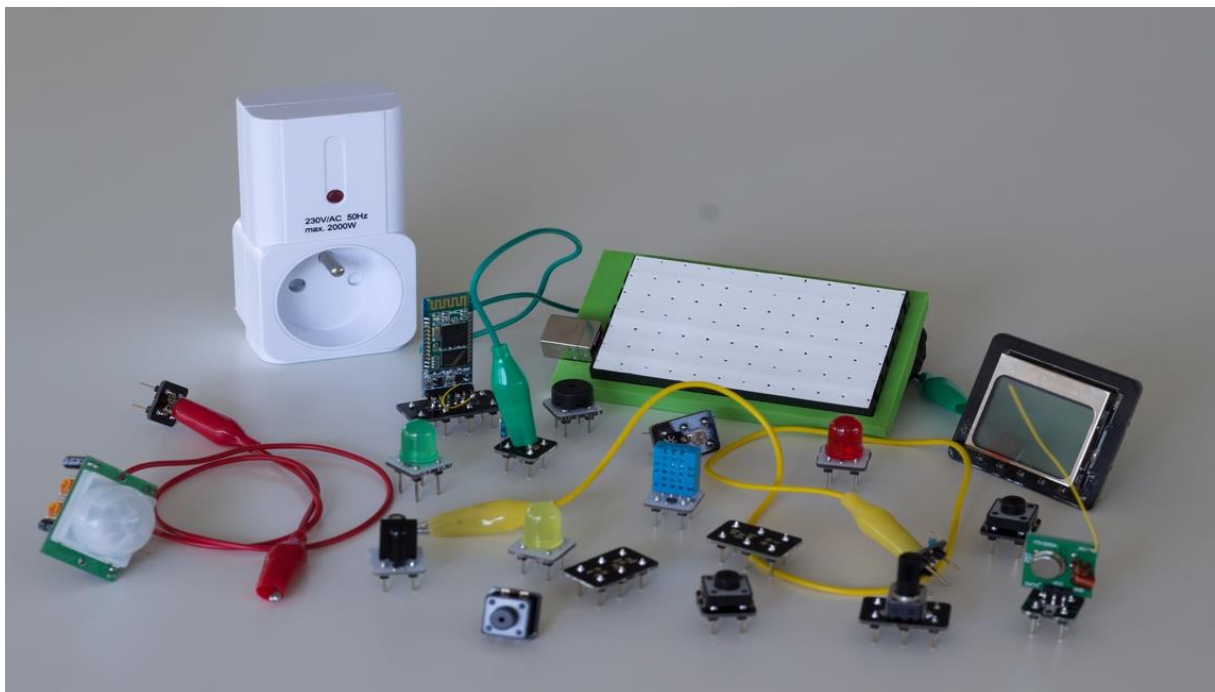
I.	L'entreprise :	3
	Présentation de l'entreprise.....	3
	Localisation de l'entreprise.....	4
2.	Contexte technique :	5
	Introduction.....	5
	Technologies étudiées.....	5
	Electron, explication du framework.....	5
3.	Missions réaliser :	6
	Electron.....	6
	Online.....	6
	Offline	7
	Ionic	8
	Remerciements.....	9
4.	Conclusion :	10

I. L'entreprise :

Présentation de l'entreprise :

Thingz - Mutiny est une start-up (SAS) créée en 2014 par Cyril Loucif-Durouge, qui vend des kits électroniques permettant aux enfants d'apprendre à coder. Il existe 3 kits différents en vente aujourd'hui avec plus ou moins de « briques » (modules électroniques). Le principe du produit est de brancher les briques sur une base et de venir les programmer directement sur le site internet de l'entreprise grâce à une interface de programmation ressemblant à « Scratch ».

Le produit est utilisable à partir de 8 ans, mais peut très bien être utilisé par des adolescents ou des adultes voulant découvrir la logique de la programmation et avoir l'autosatisfaction d'avoir créé un objet qu'ils auraient pu acheter dans le commerce. Il est ainsi possible avec les 14 briques que propose l'entreprise de réaliser des objets plus ou moins complexes allant d'une simple LED qui s'allume à la gestion de prises radio commander pour commencer à faire un de la domotique à la maison.



Localisation de l'entreprise :

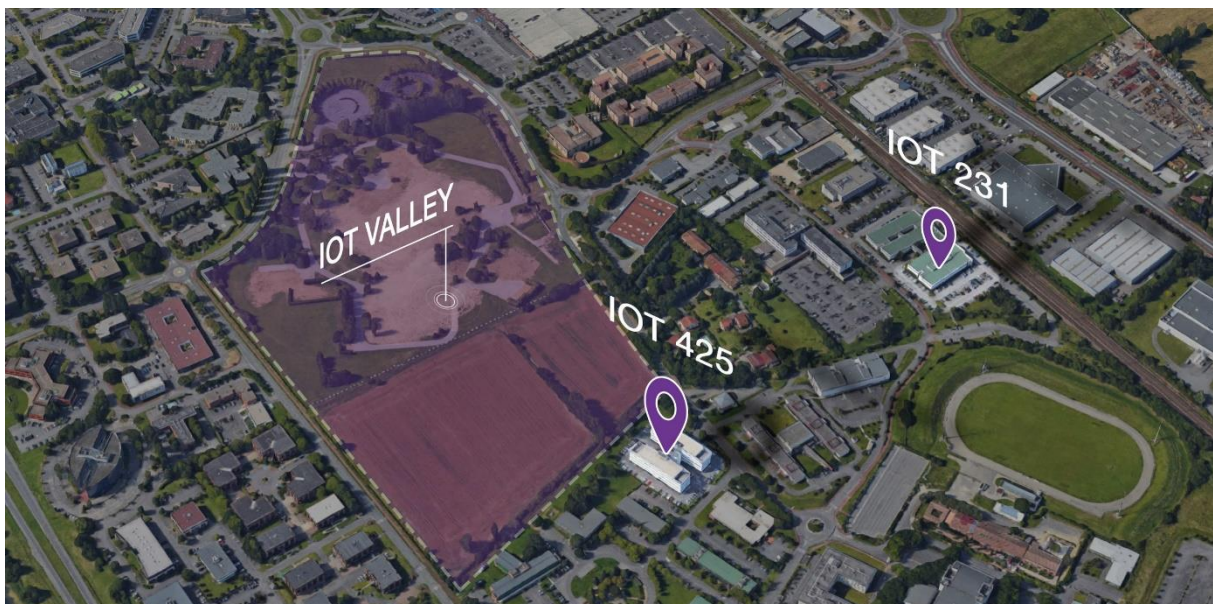
L'entreprise est située dans l'un des deux bâtiments de l'IOT Valley à Labège (Bâtiment 2 : 231 Rue Pierre et Marie Curie). L'IOT Valley est une association qui s'est donnée pour mission de développer l'excellence et la productivité dans l'Internet Des Objets. Créé en 2009 par 4 entrepreneurs, elle compte aujourd'hui plus de 4 000 membres et s'étend sur 10 000 m².

L'IOT Valley contient un accélérateur de start-ups qui permet aux nouveaux projets de gagner en maturité rapidement en les dédouanant au maximum de toute contrainte administrative et financière. Ils ont à leurs dispositions des ateliers complets (poste à souder, imprimantes 3D, etc...), comptables et des mentors qui leur donnent des conseils sur comment monter leurs entreprises et faire grandir leurs projets dans un laps de temps de 9 mois.

Il contient aussi un espace de coworking appelé « la passerelle » qui est à disposition des entreprises sortant du cadre de l'incubation, mais n'étant pas entièrement développé. Ces espaces ne sont plus gratuits comme pour l'incubateur, mais restent à des coûts bas pour continuer à les soulager financièrement.

Pour finir, les locaux contiennent aussi des bureaux pour les entreprises plus développées et ayant atteint une plus grande maturité ainsi qu'une plus grande autonomie.

L'IOT Valley étant en expansion constante, elle a pour ambition de créer un campus IOT à Labège. Les travaux débuteront bientôt et le campus ouvrira en 2019.



2. Contexte technique :

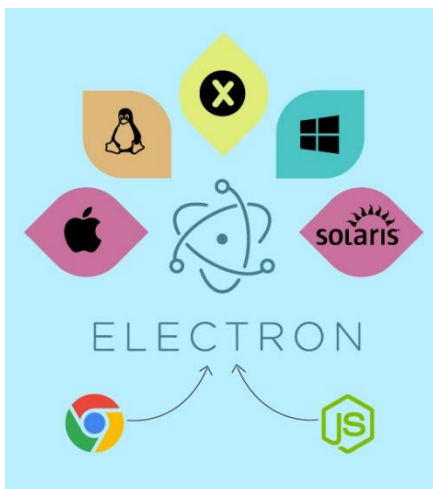
Introduction :

La suite du stage devait consister à implémenter l'application mobile que j'avais commencé à développer auparavant. Malheureusement, l'entreprise a appris que Google allait stopper pour le deuxième semestre 2017 le support des applications Google Chrome (elle en utilise une pour flasher les cartes directement depuis le navigateur). J'ai donc travaillé sur le développement d'une application de bureau qui plus tard remplacera l'application Google Chrome.

Technologies étudiées :

Dans le post de Google annonçant la fin des applications Chrome, Google donne des alternatives et cite différentes technologies permettant aux développeurs de migrer leurs applications. Après quelques recherches effectuées sur les technologies proposées, l'entreprise a choisi Electron.

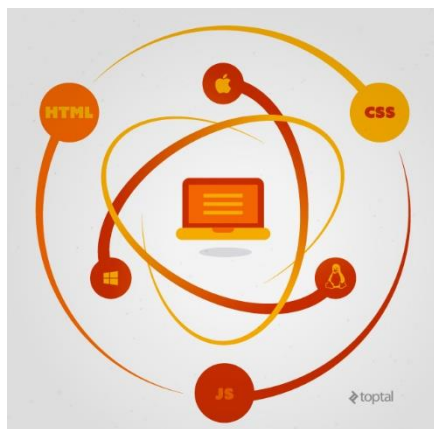
Electron, explication du framework :



Electron est un Framework permettant de développer des applications de bureau avec des technologies web (HTML, CSS et JavaScript). Il permet de développer des applications multiplateformes (comme Ionic). C'est un logiciel libre sous licence MIT développé par GitHub.

Electron permet (de la même manière que Ionic) de créer des applications performantes avec un gain de temps énorme sur le développement grâce à l'intégration de plugins via npm qui sont développés par la communauté.

Info : npm est le gestionnaire de paquets pour Node.js. Electron et Ionic utilisent tous les deux Node.js, ils bénéficient de npm pour installer des paquets (des plugins) tiers.



Grâce à npm, le développeur peut inclure d'autres Frameworks (Ionic, socket.io ...) dans son projet Electron, mais aussi des langages dérivés du JavaScript (Angular, TypeScript, ...). Il peut ainsi bénéficier des composants internes à chaque Framework ou à chaque langage dans son application de bureau.

3. Missions réaliser :

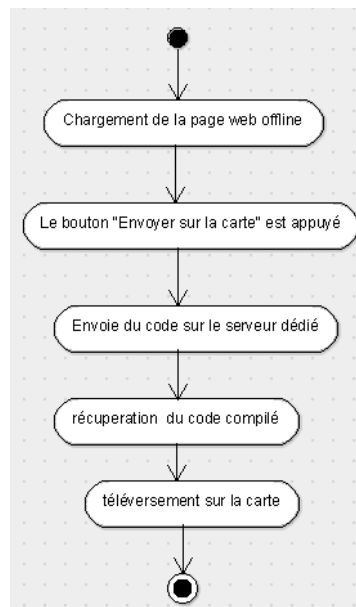
Electron :

Le but de cette application de bureau était de remplacer l'application Google Chrome existante, mais aussi d'offrir une version offline de l'interface de programmation de la carte. Par des soucis de rapidité de compilation et de téléversement sur le produit et de mise a jours efficaces de l'interface de programmation, l'entreprise souhaitais garder une version online.

L'application devait donc être en mesure de déterminer si une connexion internet était active sur l'ordinateur. A l'issu de ce résultat, l'application charge une page web en locale dans l'application ou une page web du site de l'entreprise.

Online :

Toute la partie online de l'application était déjà pratiquement fonctionnelle : la compilation du code générer via l'interface de programmation était réaliser par un serveur dédié. Pour ce qui est du téléversement sur la carte, il suffisait de récupérer le code de l'application Google Chrome et de le modifier pour le rendre compatible avec electron. Google Chrome proposait des librairies pour accéder aux ports séries qui n'était pas compatible avec electron. Il a donc fallu passer par un module npm qui faisait un travail similaire.



Offline :

Toute la partie compilation offline de l'application était à réaliser. La carte de l'entreprise étant basée sur de l'Arduino, il fallait utiliser le logiciel Arduino pour compiler. Arduino propose des versions dites portables de son logiciel (qui ne nécessitent pas d'installation). Le logiciel Arduino peut être commandé par ligne de commande, ce qui nous permet donc d'exécuter des commandes sur le logiciel Arduino depuis l'application Electron via le module « Child_process » de Node.JS. Lors de l'appui sur le bouton « Envoyer sur la carte », l'application electron crée un fichier temporaire et y stocke le code généré par l'interface de programmation et exécute la commande suivante (pour Windows) :

```
arduino.exe --verbose --pref build.path="+tempPath+"\myBuild --verify "+path+"\\"+filename+" --board arduino:avr:thingz --preserve-temp-files
```

arduino.exe : exécutable du logiciel Arduino.

--verbose : demande le mode verbeux (avoir toutes les étapes de l'exécution de la commande dans les logs)

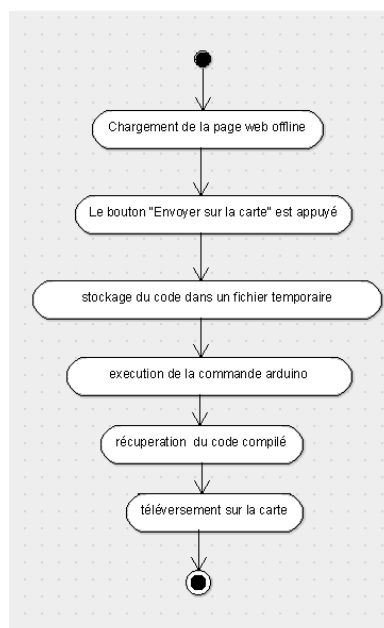
--pref build.path="+tempPath+"\myBuild : localisation du résultat de la compilation. *tempPath* est la variable contenant le chemin du dossier temporaire de l'ordinateur.

--verify "+path+"\\"+filename+" : demande uniquement la compilation du fichier passer en paramètre. *Path* est le chemin du fichier temporaire, *filename* est le nom du fichier temporaire qui contient le code à compiler.

--board arduino:avr:thingz : demande la compilation avec les spécification de la carte Thingz.

--preserve-temp-files : ne supprime pas les fichiers temporaires générés.

A la fin de la compilation, la partie téléversement est la même que pour la partie online.



Ionic :

Dans la poursuite de l'application avec Ionic, il a fallu mettre en place un protocole de communication entre l'application et la carte Thingz.

L'application ne va pas servir à envoyer simplement des messages à la carte, le but de cette application est de permettre aux utilisateurs de déclencher des événements sur la carte depuis le téléphone. L'utilisateur peut utiliser un capteur du téléphone (micro, gyroscope, accéléromètre, ...) et envoyer les valeurs de ces capteurs à la carte pour ensuite les traiter : par exemple en utilisant le micro de son téléphone, il peut allumer une LED en tapant dans ses mains.

Le protocole de communication entre la carte et l'application est là pour identifier quelles données ont été envoyées : s'il s'agit de la valeur du micro, de celle de l'accéléromètre ou d'un message texte envoyé depuis l'application.

L'encapsulation du protocole est comme ceci :

<i>ID Thingz</i>	<i>ID capteur</i>	<i>Donnée Utile</i>
------------------	-------------------	---------------------

ID Thingz permet d'identifier si le message est bien envoyé par l'application Thingz.

ID capteur permet d'identifier la nature de la valeur contenue dans la donnée utile.

Donnée Utile est la valeur du capteur. C'est la donnée traitée par l'utilisateur.

Ainsi, la carte peut identifier de quel capteur il s'agit et déclencher des actions différentes en fonction de la donnée envoyée par l'utilisateur.

Remerciements :

Je tiens à remercier tout particulièrement :

- Cyril Loucif-Durouge, CEO de l'entreprise et tuteur de mon stage, pour son aide ses conseils et son temps lors de mes différentes phase de questionnements pour toutes les réponses qu'il a pu m'apporter et l'aide qu'il a pu me fournir.
- Clément Amblard, développeurs en systèmes embarqués, pour tous ses conseils et ses astuces de développement apporter lors de mes phases de développements.

4. Conclusion :

Lors de cette seconde partie de ma période de formation, j'ai eu l'occasion de continuer à apprendre ce que j'avais commencé lors de la première partie du stage. J'ai réussi à me perfectionner dans l'utilisation des outils de Node.JS.

Travailler sur deux projets en parallèles demande encore plus de rigueur dans le travail, autant sur le plan recherches que sur les tâches à effectuer pour chacun des deux projets.