Revue 2

1. Planification du projet

Notre projet est dédié au développement d'un point d'accès à base de microcontrôleur ARM Cortex M4 pour gérer à la fois la liaison Internet WiFi et la communication Bluetooth avec les capteurs. Ceci permettra aux utilisateurs de facilement contrôler leurs objets connectés via une seule application sous Android.

Après avoir étudié les objectifs du projet, nous avons décidé de travailler en parallèle chacun sur une partie, soit la partie Hardware et la partie Android. Nous devons donc nous accorder pour avancer le projet à un rythme continu. Dans notre planification provisoire, le projet est divisé sur 4 parties. Pendant le premier mois, la tâche principale est de bien définir les objectifs et les tâches. En plus, nous devons nous familiariser avec l'environnement de développement (IDE) du microcontrôleur ARM de Texas Instruments utilisé. Ensuite, dans la deuxième partie, on travaille chacun sur sa partie afin de programmer des fonctionnalités de base. Dans la partie suivante, nous nous concentrons sur la liaison Internet entre les deux terminaux, soit la partie la plus compliquée. Finalement, nous testons notre système en entier, réglons les problèmes et faisons les dernières améliorations.

Afin d'avancer sur ce projet, nous avons utilisé la méthode SMART (**S**pécifique, **M**esurable, **A**tteignable, **R**éaliste, **T**emporel) pour raffiner les tâches dans la planification.

Par exemple, pour une tâche « Mise en place la liaison UART pour Bluetooth », on l'a définie comme suit :

Spécifique : Une liaison UART qui communique avec le module Bluetooth pour envoyer les commandes.
Mesurable : Visualiser le résultat dans le console
Atteignable : Premier pas vers l'objectif final
Réaliste : Faisable avec l'exemple proposé par TI et les documentations
Temporel : Durée - 1 semaine

Cette méthode nous a facilité le contrôle du résultat, c'est-à-dire pour chaque tâche qu'on n'a pas pu réaliser, on peut trouver le problème. Parfois, le but n'est pas du tout atteignable, alors soit on était trop ambitieux, ou nous avons mal estimé le temps qu'il nous faut pour arriver à ce résultat. Cela nous ramène dans la réalité et nous faire connaître mieux où on est dans le projet.

La planification en détail est dans Annexe 1, et le graphe Gantt est dans Annexe 2.

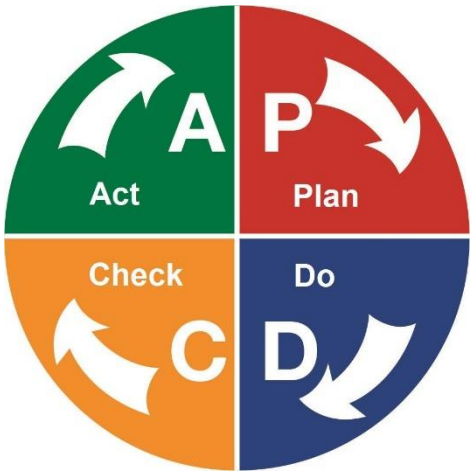
2. Déroulement du projet

Pendant 3 mois, nous avons réalisé un système qui peut interagir avec deux modules via Internet.

Les problèmes rencontrés tout au long du projet : Nous avons essayé plusieurs chemins pour atteindre notre objectif. Entre le choix de TI-RTOS et un programme sans OS (Système d'Opération), nous avons pris plus de deux semaines pour apprendre comment un système d'opération en temps réel fonctionne, bien que nous ayons décidé d'abandonner cette méthode finalement à cause de la difficulté. Limités par le temps et notre capacité, nous n'avons pas pu réaliser tous nos objectifs. Cependant, nous avons appris qu'afin de réaliser un projet ambitieux, il faut partir d'un système fonctionnel minimal. Ainsi, au lieu de regarder toutes les solutions possibles et d'en choisir dans les jours qui suivent, il vaut mieux commencer tout de suite et réaliser des fonctionnalités de base en utilisant les outils de base.

Après ce choix difficile, nous avons avancé ce projet sans utilisation d'un RTOS. Un des inconvénients est qu'il faut abandonner aussi des fonctionnalités pour la partie Android. Pour la fonction d'alerte, par exemple, il nécessite que l'application Android fonctionne comme un serveur et que la carte TI fonctionne comme un client. Sans un système d'opération qui est censé justement répartir les tâches et les procédures, il est compliqué de réaliser cela en utilisant un programme « linéaire ». Cela nous oblige à revoir les fonctionnalités pour l'application Android. En réfléchissant sur notre objectif initial, nous avons décidé d'ajouter l'exportation de données qui est nécessaire pour traiter autrement.

Au cours de ce projet, nous avons utilisé la méthode PDCA (**P**lan - **D**o - **C**heck - **A**ct) pour réaliser pas à pas notre projet.



Par exemple, pour le problème qu'on a mentionné dessus.
Plan : Réaliser un système RTOS pour mettre en place deux liaisons TCP
Do : Apprendre le fonctionnement d'un RTOS
Check : Non réalisable
Act : Changer la méthode

Pour valoriser le temps perdu quand on essayait de choisir le chemin qu'on voulait poursuivre, on présentera une analyse des possibilités dans le rapport final. Les tâches qu'on a réalisées sont dans la partie suivante. En plus, nous avons utilisé Ganttproject (un logiciel Opensource professionnel) pour la gestion de projet, le déroulement détaillé est dans Annexe 2.

Mini Compte-Rendu

Synthèse de la réalisation

Ensemble :

- Définition de l'objectifs du projet et des résultats attendus
- Répartition des tâches
- Planning provisoire
- Comptes-rendus hebdomadaires
- Livrables

Partie Hardware :

- Définition des fonctions à intégrer
- Premiers essaies sur la carte TIVA-C
- Utilisation de l'IDE Code Composer Studio de Texas Instruments
- Notions de l'architecture ARM Cortex-M4
- Manipulation du module HC05 Bluetooth via UART
- Etude des différentes possibilités pour faire ce projet (avec RTOS ou sans RTOS)
- Apprentissage du fonctionnement de LWIP
- Mise en œuvre du protocole TCP
- Stockage de données dans EEPROM
- Gestion de la communication entre plusieurs modules Bluetooth
- Fusion des deux parties principales

Partie Android :

- Prédéfinition du résultat attendu de l'Appli Android
- Consultation de données sur les conditions mesurés par les capteurs : Température, humidité, etc. Deux formes : Diagramme, chiffre.
- Contrôle d'interrupteur de lumière : ON et OFF
- Alerte quand un état dépasse la limite : Par la notification de l'application
- Addition et suppression de périphériques : Bluetooth ou Internet
- Etude de la possibilité de faire l'interface WEB
- Besoin du serveur WEB établi sur la carte - à décider plus tard
- Première étude de mode de communication entre le matériel et l'application
- Protocole TCP pour la couche Transport et protocole HTTP pour la couche Application
- Besoin de créer un TCPClient et un TCPServer pour réaliser les fonctions
- Nécessité d'étudier l'utilisation des sockets pour la communication
- Connexion avec la partie Hardware
- Extraction de données utiles dans un String
- Stockage de données de la température et de l'humidité dans les bases de données
- Transformation de données en fichiers .csv ou .xls
- Traitement de données (Diagramme à ligne brisée ,...)

Analyse des risques

Risque 1 : Peu de documentations pour LWIP

Explication : Pour établir une liaison TCP/IP, nous avons 3 méthodes. La première est l'utilisation des sockets dans un RTOS. La deuxième est l'utilisation de LWIP (Light-Weight IP). La dernière est d'utiliser uIP(micro IP). Les deux dernières sont des piles TCP/IP réduites pour les systèmes embarqués. Nous avons décidé de prendre LWIP parce qu'il est plus léger que les sockets et plus populaire que uIP. Le risque est qu'on ne connaît pas l'API de LWIP, et que TI ne fournit qu'un exemple de HTTP.

Solution : Vu que l'API TCP/IP est une librairie externe (third-party), nous avons cherché la documentation initiale fournie par les développeurs. Même si cela ne correspond pas forcément à notre environnement, nous avons pu apprendre la démarche de la pile LWIP. Ensuite, j'ai trouvé un programme pour établir une simple liaison TCP dans l'environnement du STM32 proposé par STMicroelectronics (un autre industriel dans les systèmes embarqués). En même temps, on a étudié l'exemple de TI pour comprendre la gestion de ressources. Nous avons donc transporté le programme du STM32 à notre propre carte en considérant la gestion des ressources.

Risque 2 : Choix entre RTOS ou sans OS

Explication : Ce choix entre RTOS et un programme sans RTOS apparaît tout au long du projet. D'abord, RTOS offre beaucoup de fonctions qui sont déjà prêtes dans de nombreuses APIs. En outre, RTOS a intégré les sockets pour la programmation réseaux. Enfin, il est possible de gérer les processus et les tâches. Néanmoins, il existe aussi des contraintes. D'un côté, il faut prendre du temps à comprendre la structure d'un système en temps réel et les concepts de base. De l'autre côté, TI n'a pas fourni des exemples précis.

Solution : Malgré tous les avantages, nous avons décidé de faire un programme fonctionnel sans système. Si on avait choisi d'apprendre le RTOS, on risque de ne pas finir les attendus minimaux de ce projet. Pour l'intégrité du projet, ce choix est nécessaire.

Risque 3 : Manque de connaissances sur la programmation des fichiers Layout

Explication : La programmation des activités dans Android utilise principalement le langage JAVA que nous avons étudié pendant quelques cours à Télécom SudParis. Mais le fichier XML pour dessiner l'interface graphique d'application est plutôt différent. Il gère toutes les configurations de la position et du style des boutons, des textes et des autres modules. Nous avons seulement appris des opérations les plus essentielles sur cela pendant le cours « Développement mobile Android » et il ne suffit pas pour l'interface graphique avec des éléments plus compliqués.

Solution : D'abord, il est nécessaire de chercher plus d'informations sur l'utilisation des layouts différents. Puis il ne faut pas toujours utiliser les layouts trop simples comme « LinearLayout ». Ce n'est pas facile de calculer toutes les tailles des éléments et les espaces entre eux. On peut essayer d'utiliser « CoordinatorLayout » avec qui on peut déplacer manuellement des éléments ou « RelativeLayout » qui est plus récent et flexible.

Risque 4 : Forte dépendance de la partie Android sur la partie Hardware

Explication : Vu que la partie Android a pour objectif de faciliter la gestion des capteurs différents, les fonctions d'application dépendent fortement de la réalisation de la partie Hardware. Elles ne sont pas toujours précises avant que la carte et les capteurs fonctionnent. Et des tests entre l'application et la carte ne peuvent pas être garantis jusqu'à la fin du projet.

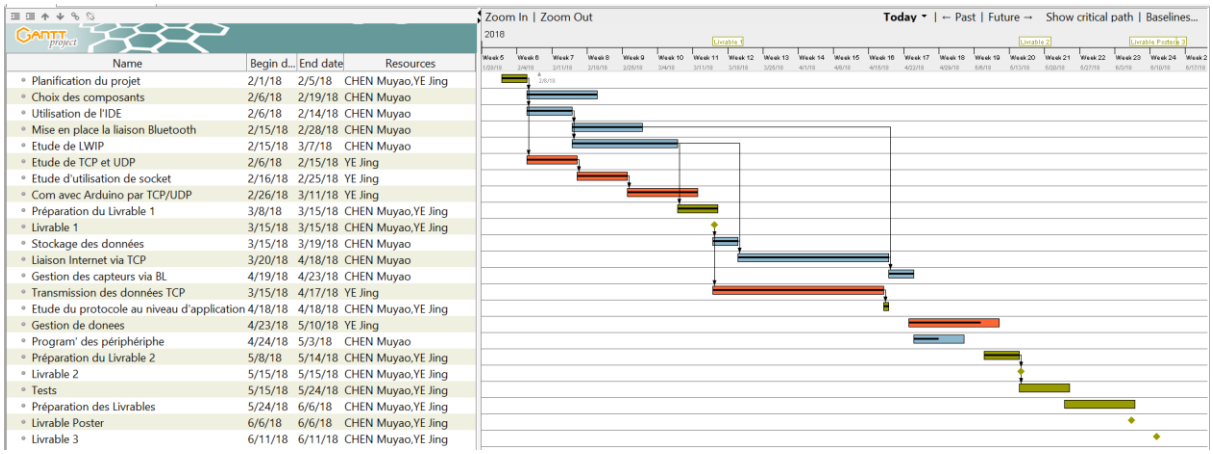
Solution : Nous identifions les fonctions qui peuvent être réalisées sans l'aide de Hardware, par exemple la connexion TCP en utilisant le socket. Pour le test, il existe des logiciels sur Internet qui peut simuler un serveur TCP pour recevoir et envoyer des messages.

Annexe 1

	Partie Hardware (Muyao Chen)	**Déroulement à 15/05**	**Partie Software (Jing Ye)**	**Déroulement à 15/05**
Période 1 (1/2-1/3)	**Liaison Bluetooth**		**Etude des protocoles**	
Tâche 1	Planification du projet	Terminé	Planification du projet	Terminé
Tâche 2	Choix des composants	Terminé	Etude de TCP et UDP	Terminé
Tâche 3	Utilisation de l'IDE	Terminé	Etude d'utilisation de socket sur la plateforme Android	Terminé
Tâche 4	Mise en place la liaison UART pour Bluetooth	Terminé	Communication avec Arduino par TCP/UDP	Terminé
Tâche 5	Etude de LWIP	Terminé		
Période 2 (1/3-1/4)	**Liaison Ethernet**		**Mise en place des fonctions fondamentaux**	
Tâche 1	Stockage des données	Terminé	Envoi et réception des données avec la carte en utilisant TCP	Terminé
Tâche 2	Mise en place la liaison Internet via TCP	Terminé	Traitement graphique des données	En cours
Tâche 3	Gestion des capteurs via Bluetooth	Terminé	Addition et suppression des devices	Non à cause de la partie Hardware
Tâche 4	Etude de la possibilité d'intégrer le WiFi	Non	Permission de notification pour l'alerte	Non à cause de la partie Hardware
Tâche 5	Programmation des périphériques	En cours		
Période 3 (1/4-1/5)	**Gestion des communications**		**Gestion des données et d'interface graphique**	
Tâche 1	Etude du protocole au niveau d'application	Terminé	Etude du protocole au niveau d'application	Terminé
Tâche 2	Intégration des deux liaisons	Terminé	Amélioration d'interface graphique	A venir
Tâche 3	Gestion du processus	Terminé	Tests avec carte TI	En cours
Tâche 4	Tests avec Appli Android	En cours	Stockage des données dans les bases de données	Terminé
Tâche 5				
Période 4 (1/5-31/5)	**Modifications et Améliorations**		**Modifications et Améliorations**	
Tâche 1	Gestion des problèmes	En cours	Gestion des problèmes	En cours

Tâche 2	Améliorations possibles	A venir	Améliorations possibles	A venir
Tâche 3	Travail en plus à voir (Serveur WEB ou RTOS)	A venir	Travail en plus à voir	A venir
Tâche 4	Préparations des livrables	En cours	Préparations des livrables	En cours
Tâche 5	Préparation pour la soutenance	A venir	Préparation pour la soutenance	A venir

Annexe 2



Name	Begin d...	End date	Resources
Planification du projet	2/1/18	2/5/18	CHEN Muyao,YE Jing
Choix des composants	2/6/18	2/19/18	CHEN Muyao
Utilisation de l'IDE	2/6/18	2/14/18	CHEN Muyao
Mise en place la liaison Bluetooth	2/15/18	2/28/18	CHEN Muyao
Etude de LWIP	2/15/18	3/7/18	CHEN Muyao
Etude de TCP et UDP	2/6/18	2/15/18	YE Jing
Etude d'utilisation de socket	2/16/18	2/25/18	YE Jing
Com avec Arduino par TCP/UDP	2/26/18	3/11/18	YE Jing
Préparation du Livrable 1	3/8/18	3/15/18	CHEN Muyao,YE Jing
Livrable 1	3/15/18	3/15/18	CHEN Muyao,YE Jing
Stockage des données	3/15/18	3/19/18	CHEN Muyao
Liaison Internet via TCP	3/20/18	4/18/18	CHEN Muyao
Gestion des capteurs via BL	4/19/18	4/23/18	CHEN Muyao
Transmission des données TCP	3/15/18	4/17/18	YE Jing
Etude du protocole au niveau d'application	4/18/18	4/18/18	CHEN Muyao,YE Jing
Gestion de donees	4/23/18	5/10/18	YE Jing
Program' des périphériphe	4/24/18	5/3/18	CHEN Muyao
Préparation du Livrable 2	5/8/18	5/14/18	CHEN Muyao,YE Jing
Livrable 2	5/15/18	5/15/18	CHEN Muyao,YE Jing
Tests	5/15/18	5/24/18	CHEN Muyao,YE Jing
Préparation des Livrables	5/24/18	6/6/18	CHEN Muyao,YE Jing
Livrable Poster	6/6/18	6/6/18	CHEN Muyao,YE Jing
Livrable 3	6/11/18	6/11/18	CHEN Muyao,YE Jing

Name	Begin d...	End date	Resources
Planification du projet	2/1/18	2/5/18	CHEN Muyao,YE Jing
Choix des composants	2/6/18	2/19/18	CHEN Muyao
Utilisation de l'IDE	2/6/18	2/14/18	CHEN Muyao
Mise en place la liaison Bluetooth	2/15/18	2/28/18	CHEN Muyao
Etude de LWIP	2/15/18	3/7/18	CHEN Muyao
Etude de TCP et UDP	2/6/18	2/15/18	YE Jing
Etude d'utilisation de socket	2/16/18	2/25/18	YE Jing
Com avec Arduino par TCP/UDP	2/26/18	3/11/18	YE Jing
Préparation du Livrable 1	3/8/18	3/15/18	CHEN Muyao,YE Jing
Livrable 1	3/15/18	3/15/18	CHEN Muyao,YE Jing
Stockage des données	3/15/18	3/19/18	CHEN Muyao
Liaison Internet via TCP	3/20/18	4/18/18	CHEN Muyao
Gestion des capteurs via BL	4/19/18	4/23/18	CHEN Muyao
Transmission des données TCP	3/15/18	4/17/18	YE Jing
Etude du protocole au niveau d'application	4/18/18	4/18/18	CHEN Muyao,YE Jing
Gestion de donees	4/23/18	5/10/18	YE Jing
Program' des périphériphe	4/24/18	5/3/18	CHEN Muyao
Préparation du Livrable 2	5/8/18	5/14/18	CHEN Muyao,YE Jing
Livrable 2	5/15/18	5/15/18	CHEN Muyao,YE Jing
Tests	5/15/18	5/24/18	CHEN Muyao,YE Jing
Préparation des Livrables	5/24/18	6/6/18	CHEN Muyao,YE Jing
Livrable Poster	6/6/18	6/6/18	CHEN Muyao,YE Jing
Livrable 3	6/11/18	6/11/18	CHEN Muyao,YE Jing