



Mémoire de Projet de fin d'étude

Préparé par

HAIMOURA Dounia

Pour l'obtention du diplôme

Ingénieur d'Etat en

SYSTEMES ELECTRONIQUES & TELECOMMUNICATIONS

Intitulé

**Mise en place d'un système de
monitoring industriel à distance**

Encadré par :

Pr F.ABDI

Mr R.BENNANI (MAscIR)

Soutenu le Mercredi 22 Juin 2016, devant le jury composé de :

Pr F.ABDI

: Encadrant

Pr N.S.ECHATOU

: Examineur

Pr A.AHAITOUF

: Examineur

وَقَدْ نَزَّلَ عَلَيْنَا

سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ الْحَكِيمُ

Dédicace

- {Louange à Dieu seul} -

- {Que le salut et la paix soient sur l'envoyé du Dieu} -

Que ce travail présente mon respects:

A mes parents

Grâce à vos tendres encouragements et vos grands sacrifices, vous avez pu Créer le climat affectueux propice à la poursuite de mes études. Aucune dédicace ne saurait exprimer mes respects, mes considérations et mes profonds sentiments pour vous. Je prie Dieu de vous bénir, de veiller sur vous, et j'espère que vous serez toujours fiers de moi.

A mes frères et mes sœurs, ainsi qu'à mes collègues

Je vous dédie ce travail en vous souhaitant un avenir plein de réussite et de bonheur.

A mes ami(e)s

Trouvez ici le témoignage d'une fidélité et amitié infinies.

A mes chers professeurs

Votre générosité et votre soutien m'oblige à vous prendre en considération sur ce dédicace.

A toute l'équipe de MASCIR

Tous ceux qui m'ont soutenue moralement et m'ont aidée pour compiler ce modeste travail que ce soit à l'intérieur de MAScIR ou à l'extérieur.

Remerciement

Au terme de ce stage, mes remerciements les plus vifs et les plus sincères vont à M. Rachid BENNANI, ingénieur dans l'équipe packaging à MAScIR, et à M. Brahim LAKSSIR, directeur du centre Micro-packaging à MAScIR, qui ont été à l'origine de mon aide et assistance tout au long de cette période. Leur accueil chaleureux, leur accompagnement sans égal et leurs conseils tant au niveau professionnel que personnel ont rendu cette période agréablement constructive.

Je souhaite remercier mon promoteur et mon encadrant à l'université Pr. ABDI Farid pour ses instructions et son aide lors du stage.

Je présente ma gratitude envers tout le personnel de MAScIR et spécialement l'équipe packaging et l'équipe de systèmes embarqués pour la merveilleuse période de stage passée en leur compagnie et je les remercie infiniment pour leur patience et leurs précieux conseils.

Mes remerciements vont à tous les membres du jury pour avoir accepté de venir assister et évaluer ma soutenance.

Enfin, j'adresse tous mes remerciements les plus sincères à tous les professeurs de la faculté des sciences et techniques de Fès pour avoir contribué à la formation que j'ai acquise lors de mon cursus.

Résumé

Aujourd'hui les systèmes de supervision à distance couvrent en particulier le domaine domestique et une minorité d'industries avec une limitation d'indicateurs de performance. Nous voulons offrir un meilleur service aux industriels et répondre à leur besoin en terme d'optimisation de la gestion d'énergie et de productivité, à l'aide d'un véritable outil de supervision à distance et en temps réel des indicateurs de performance industrielle, moins coûteux et facile à installer.

Dans ce rapport, pour étudier la faisabilité du projet nous avons développé un prototype qui permet d'acquérir et d'envoyer sans fil et en temps réel, via Bluetooth 4.0 (Bluetooth Low Energy), les données d'une machine industrielle à savoir : le comptage des pièces produites, la détection de blocage des pièces , à travers le capteur de comptage, et la consommation énergétique, à travers le capteur de courant.

Afin d'améliorer ce prototype, nous développons un démonstrateur qui permet de répondre aux exigences industrielles en terme de traçabilité et de portée de transfert de données pour permettre aux utilisateurs du système de superviser le comportement de leurs machines industrielles à n'importe quelle moment et de partout.

Abstract

Today, remote monitoring systems cover particularly the domestic area and only a minority of industries with a limitation of performance indicators. We try to provide better service to industry and meet their needs in terms of optimizing energy and productivity management, by using a remote management tool with a real-time industrial performance indicators, within easy reach and less costly.

Our work tackles with the development of a prototype that acquires and send wirelessly , in real time through Bluetooth Low Energy, data from industrial machines as : counting the produced parts, lock detection, through the counter, and energy consumption, through the current sensor.

In order to improve this prototype, we are currently developing a demonstrator that meets industrial requirements in terms of traceability and data transfer to allow users of this system to supervise their industrial machines at anytime and from anywhere.

Sommaire

DEDICACE.....	3
REMERCIEMENTS	4
RESUME	5
ABSTRACT	6
SOMMAIRE.....	7
TABLE DES FIGURES	9
LISTE DES ACRONYMES.....	11
INTRODUCTION GENERALE.....	12

Chapitre 1 : Contexte général du projet

INTRODUCTION

I. PRESENTATION DE MAScIR.....	14
1. DEFINITION	14
2. PARTENAIRE DE LA FONDATION	15
3. STRUCTURE ET HIERARCHIE	16
4. PRESENTATION DU DEPARTEMENT MICROELECTRONIQUE	16
4.1 MISSIONS	17
4.2 ORGANISATION	17
II. CADRE DU PROJET	18
1. OBJECTIF DU PROJET	18
2. METHODOLOGIE DE TRAVAIL	18
III. CONDUITE DU PROJET	20
1. PLANIFICATION DU PROJET	20
2. ETAT DE L'ART.....	20

CONCLUSION

Chapitre 2 : Développement d'un premier prototype

INTRODUCTION

I. ARCHITECTURE DU DEMONSTRATEUR SIMPLIFIE.....	25
II. DESCRIPTION MATERIELLE	28

1. CARTE NUCLEO-STM32L0	28
2. CARTE BLUENRG	30
III. DESCRIPTION LOGICIELLE	34
1. STM32CUBEMX	34
2. KEIL MDK-ARM 5.13	35
3. ANDROID STUDIO	35
IV. REALISATION DU PROTOTYPE	37
CONCLUSION	

Chapitre III : Développement d'un démonstrateur industriel

INTRODUCTION

I. ARCHITECTURE GENERALE DU SYSTEME	41
II. DESCRIPTION MATERIELLE	42
1. ULTRASONIC HC-SR04	43
2. ESP8266-WIFI	43
3. WAMP SERVER	44
III. ARCHITECTURE DE L 'APPLICATION ANDROID	45
IV. AMELIORATIONS FUTURES	47
1. IMPLEMENTATION DES PROTOCOLES DE COMMUNICATION	47
2. CENTRALE DE MESURE	49

CONCLUSION

CONCLUSION GENERALE	51
BIBLIOGRAPHIE	52
ANNEXES	53

Table de figures

Figure 1.1 : localisation de la fondation MAScIR

Figure 1.2 : Secteur d'activité de la fondation MAScIR

Figure 1.3 : Organigramme du service Microélectronique

Figure 1.4 : Cycle de développement en Y

Figure 1.5 : Planning du projet

Figure 1.6 : Architecture du réseau PSEMS

Figure 1.7 : Architecture SCADA

Figure 2.1 : Architecture de démonstrateur simplifié

Figure 2.2 : Machine Wire Bonder

Figure 2.3 : Capteur de comptage

Figure 2.4 : Capteur de courant

Figure 2.5 : Courant appelé par une charge linéaire

Figure 2.6 : Courant appelé par une charge non linéaire

Figure 2.7 : Carte Nucleo-STM32L053R8

Figure 2.8 : Carte X-NUCLEO-IDB04A1

Figure 2.9 : Protocole BLE

Figure 2.10 : Hiérarchisation de GATT

Figure 2.11 : Architecture de l'application Android

Figure 2.12 : Cycle de vie d'une activité

Figure 2.13 : Réalisation du prototype

Figure 2.14 : Etapes de l'application Android

Figure 3.1 : Architecture générale du système

Figure 3.2 : Architecture du démonstrateur industriel

Figure 3.3 : Capteur ultrasonic HC-SR04

Figure 3.4 : Module ESP8266-Wifi

Figure 3.5 : Modèle conceptuel de données

Figure 3.6 : Modèle logique de données

Figure 3.7 : Architecture de l'application Android

Figure 3.8 : Application Android

Figure 3.9 : Fonctionnement de microcontrôleur ESP8266

Figure 3.10 : Communication entre serveur et microcontrôleur

Figure 3.11 : Communication entre serveur et application

Figure 3.12 : Centrale de mesure DIRIS A20

Liste des acronymes

MAScIR : Moroccan foundation for Advanced Science, Innovation and Research

SMI : Système de Monitoring Industriel

PSEMS : Production Statistics & Equipment Monitoring System

SCADA : Supervisory Control And Data Acquisition

PLC : Programmable Logic Controller

RTU : Remote Terminal Unit

HMI : Human Machine Interface

UART : Universal Asynchronous Receiver Transmitter

SPI : Serial Peripheral Interface

I2C : Inter Integrated Circuit

BLE : Bluetooth Low Energy

GAP : Generic Access Profile

ATT : Attribute Protocol

GATT : Generic Attribute Profile

SM : Security Manager

L2CAP : Logical Link Control and Adaptation Protocol

HCI : Host Control Interface

LL : Link Layer

PHY : Physical Layer

UUID : Universally unique identifier

MCU : Micro controller unit

Introduction générale

L'efficience d'un processus de transformation industriel est directement lié aux ressources matérielles et immatérielles utilisées lors de la production des produits, de ce fait un suivi en temps réel des données de production avec un contrôle à distance et à tout instant devient une exigence pour la survie et la compétitivité de toute société industrielle.

Aujourd'hui les systèmes de supervision à distance couvrent en particulier le domaine domestique et une minorité d'industries avec une limitation des outils permettant de récolter les

Chapitre 1

Contexte général du projet

Ce travail se compose de trois chapitres. Je consacre le premier chapitre pour présenter l'organisme d'accueil MAScIR et cadrer le projet en terme d'objectifs, de planning et d'étude comparative de l'existant. Ensuite, je passe à la description des outils logiciels et matériels nécessaires pour réaliser le premier prototype. Et par la suite, je met l'accent sur la phase de conception et réalisation de ce projet qui sera accompli vers la fin de mon stage. Une conclusion générale clôturera ce rapport dans la dernière partie, tout en y évoquant les perspectives qu'ouvrera cette application surtout l'utilité et l'efficacité de ce projet.

*Dans ce chapitre je présenterai
l'organisme d'accueil MAScIR, les
objectifs du projet et finalement je
décrirai la démarche adoptée dans la
conduite du projet.*

Introduction

Le présent chapitre permet de situer le projet dans son contexte général. Dans un premier temps, il s'agit de présenter l'organisme d'accueil : MAScIR, introduire, par la suite, les objectifs du projet et finalement décrire la démarche adoptée dans la conduite du projet.

I. Présentation de MAScIR :

1. Définition :

MAScIR est un organisme de recherche à caractère scientifique et technologique. Il est voué à la recherche en nanotechnologie, en biotechnologie, en technologie numérique, en microélectronique, en énergie et en environnement. La fondation se veut présente là où les enjeux de la société l'exigent.

Le montant d'investissement qui a été consacré à la réalisation du projet MAScIR est de 564 millions DH. 60 publications ont été le fruit des recherches au sein de MAScIR , ainsi que le dépôt de 7 brevets.

La figure 1.1 montre l'emplacement de l'entreprise au Maroc.

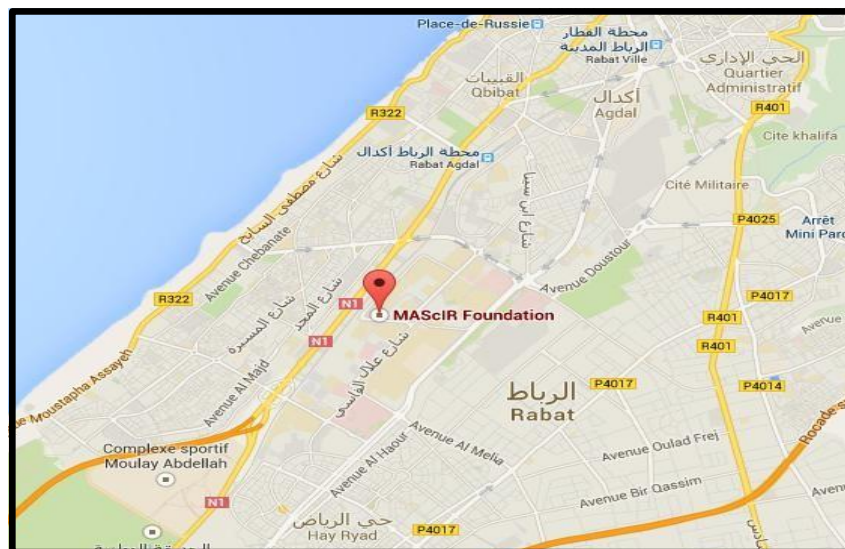


Figure 1.1 : localisation de la fondation MAScIR

Rassemblant d'éminents chercheurs des quatre coins du monde, MAScIR regroupe des équipes scientifiques œuvrant dans des domaines innovants et complémentaires et met à leur disposition une infrastructure scientifique de pointe.

Initialement fondée en 2007 par le Gouvernement Marocain en tant que fondation à but non lucratif, MAScIR a continué son expansion en créant (voir figure 1.2) :

- **MAScIR MicroElectronics** : créé vers la fin de l'année 2008, a pour objectif de devenir un centre de Recherche et Développement dans le domaine de la microélectronique.
- **MAScIR BioTechnology** : deuxième centre inscrit dans MAScIR œuvrant dans le domaine de la biotechnologie, recherche et développement des médicaments ou des biocides.

- **NanoTechnology** : qui a pour mission de mener des recherches appliquées, innovantes et à la fine pointe de la technologie pour créer de la propriété intellectuelle et des prototypes dans le domaine des nanomatériaux et des nanotechnologies. Ces recherches sont menées par une équipe internationale de haut calibre travaillant dans un environnement unique et utilisant une infrastructure de pointe.



Figure 1.2 : Secteur d'activité de la fondation MAScIR

2. Partenaire de la fondation :

Les principaux partenaires de la fondation MAScIR MicroElectronics sont :

- **Lear Corporation** : est l'un des principaux fournisseurs mondiaux de sièges automobiles et les systèmes de gestion de l'énergie électrique.
- **Thales** : figure parmi les leaders européens de la fabrication et de la commercialisation d'équipements et de systèmes électroniques destinés aux secteurs de l'aérospatial, du transport, de la défense et de la sécurité.
- **OCP** : est un acteur incontournable sur le marché des phosphates et de ses produits dérivés. Présent sur toute la chaîne de valeur, il est le premier exportateur de cette matière dans le monde.
- **STERIMED** : est une société spécialisée dans le domaine de l'eau et des technologies de l'environnement. Son objectif est d'accompagner les entreprises et collectivités dans la résolution des problématiques liées à l'eau, l'environnement.
- **COSUMAR** : est un groupe marocain, filiale de la Société nationale d'investissement, spécialisé dans l'extraction, le raffinage et le conditionnement du sucre sous différentes formes. Il est devenu l'unique opérateur sucrier marocain après l'acquisition de SUTA, SUCRAFOR, SUNABEL et SURAC en 2005.

3. Structure et hiérarchie :

La Fondation est gérée par un conseil d'administration qui est investi de pouvoirs de gestion à cet égard. Le Conseil dispose de plusieurs comités distincts - un Comité d'Investissement, un Comité d'Audit et un Comité de Rémunération - qui assurent une gestion rapprochée des sujets relatifs à leur mission.

- **Conseil d'administration** : détermine les orientations stratégiques MAScIR et veille à leur mise en œuvre se réunit sur une base régulière. En prenant des décisions, le Conseil compte sur le travail régulier des comités spécialisés.
- **Comité de vérification** : le rôle principal du Comité d'audit est de permettre à la Commission de veiller à la qualité des contrôles internes et l'intégrité de l'information divulguée aux intervenants et aux partenaires.
- **Comité des Rémunérations** : est responsable de faire des recommandations au Conseil sur la nomination des administrateurs. Il est également responsable de l'examen de la politique en matière de rémunération de la haute direction au sein de MAScIR.
- **Comité de suivi** : le comité est de surveiller la mise en œuvre effective et correcte des projets dans le cadre de l'accord signé entre MAScIR et le Gouvernement marocain.
- **Comité d'Investissement** : le rôle du comité d'investissement assiste le Conseil d'administration dans l'accomplissement de sa responsabilité de surveillance pour les actifs d'investissement liés à l'équipement scientifique.

4. Présentation du département Microélectronique :

MAScIR Microélectronique est un centre d'innovation et développement de technologie dans le domaine de la microélectronique. Il se focalise sur la simulation, les tests, le design, le packaging, la qualification et le prototypage des produits microélectroniques.

4.1. Missions :

Le programme Microélectronique a réuni une équipe de direction de classe mondiale pour assurer la traction initiale, et sous licence des technologies de pointe qui sont disponibles pour une utilisation immédiate.

L'équipe travaille actuellement sur la construction des liens étroits de collaboration avec des institutions de classe mondiale commerciales et académiques, d'abord en se concentrant sur le développement de produits orientés vers le marché et le prototypage.

MAScIR Micro fournit des services pour des clients industriels, mais elle développe aussi son propre business dans les domaines suivants :

- L'intégration et la miniaturisation des systèmes microélectroniques.
- L'analyse de fiabilité et défaillance des produits.
- Modélisation des systèmes complexes.
- Prototypage et industrialisation des produits innovants.
- Industrialisation des idées et résultats académiques.

4.2. Organisation :

La figure 1.3 montre la hiérarchie du département Micro de la fondation MAScIR:

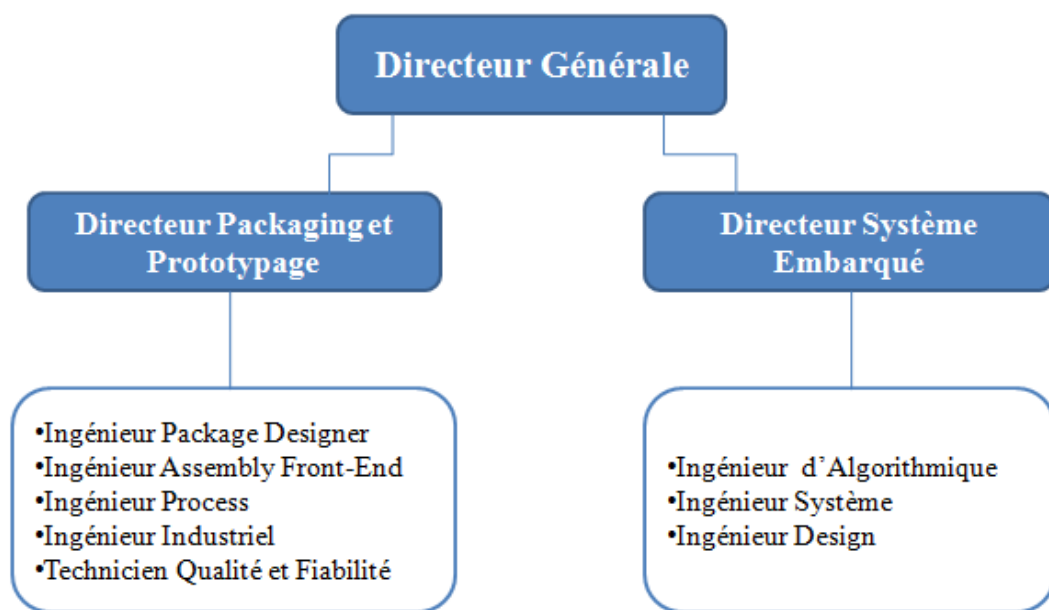


Figure 1.3 : Organigramme du service Microélectronique

II. Cadre du projet :

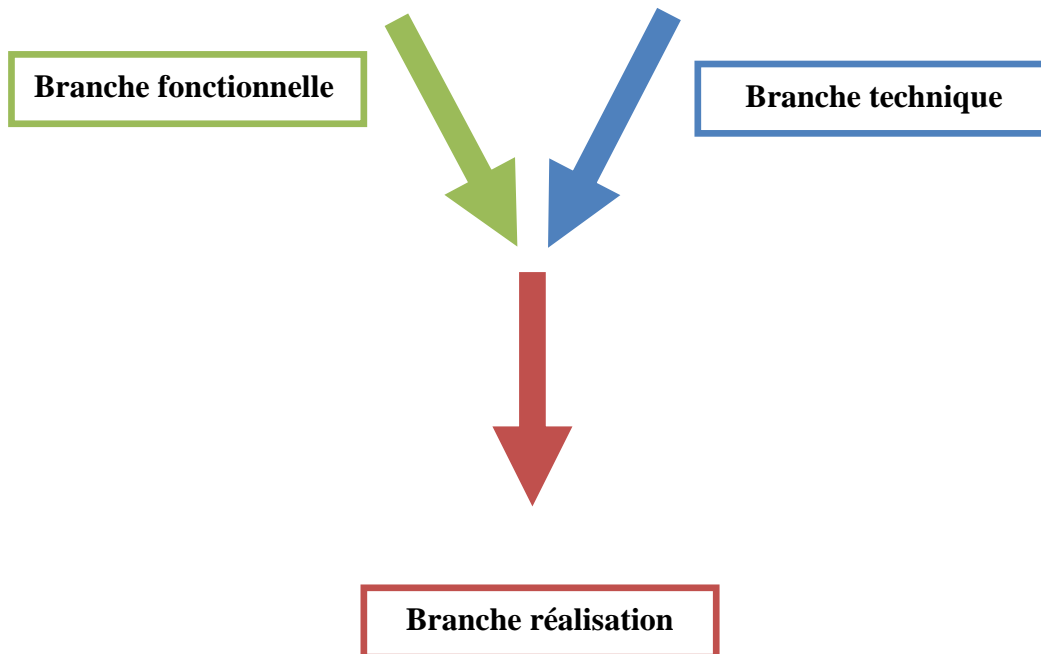
1. Objectif du projet :

Mon projet de fin d'études a pour objectif le développement et la réalisation d'un démonstrateur qui sera mis en communication avec les machines en production, permettant de récolter en temps réel et à distance, les informations sur le comportement des équipements de production à savoir : le comptage des produits en fin de chaîne de production, marche et arrêts de machine, la consommation d'énergie rapportée au volume de production, ainsi que le comptage des rejets.

2. Méthodologie de travail :

La réalisation d'un projet qui vise à développer un tel système de supervision à distance et en temps réel des indicateurs industriels des machines en production, nécessite une analyse approfondie de l'aspect fonctionnel d'une part et de l'aspect technique du système d'autre part. D'où l'idée d'organiser et de développer le projet suivant une approche méthodologique pour cerner ces objectifs et rendre sa réalisation plus compréhensible, pour cela le cycle de développement en Y reste le processus le plus adapté au cas de ce projet, puisqu'il permet de dissocier les aspects techniques des aspects fonctionnels.

En général le processus de développement en Y s'articule autour de trois phases :



- **La branche fonctionnelle** : comporte l'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

- **La branche technique** : comporte la conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique qui englobe les besoins matériels et logiciels pour la réalisation des projets. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.
- **La branche réalisation** : comporte la conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer (voir figure 1.4).

→ La conception détaillée, qui étudie ensuite comment réaliser chaque composant.

→ L'étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées.

→ L'étape de recette, qui consiste enfin à valider les fonctions du système développé.

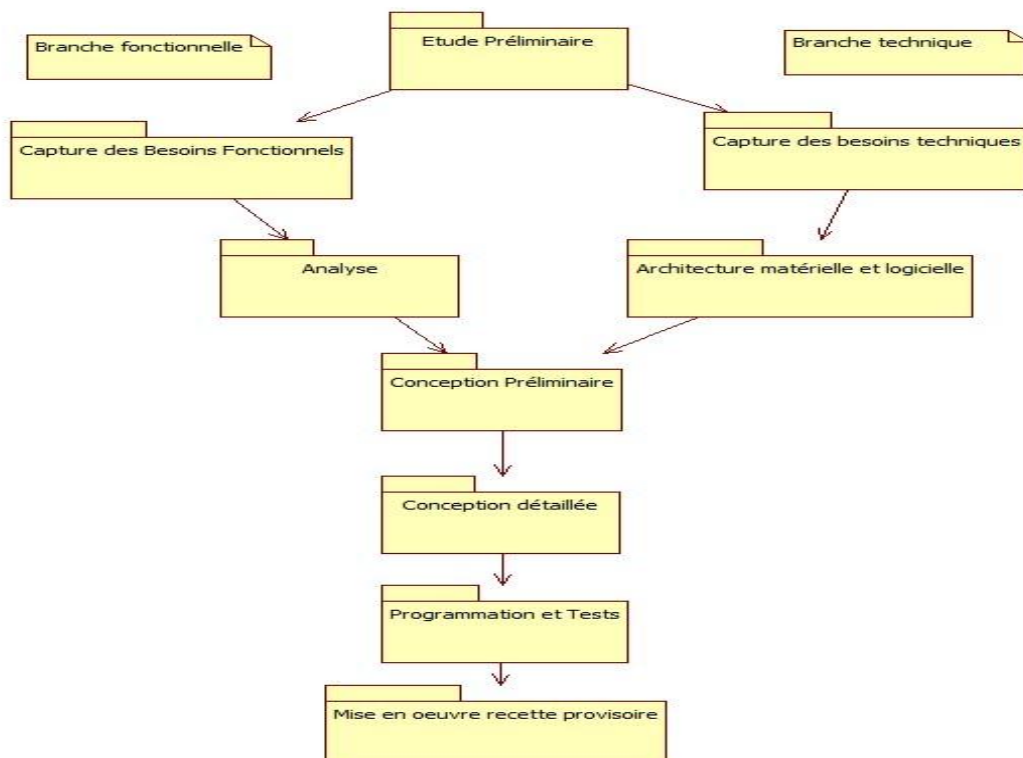


Figure 1.4 : Cycle de développement en Y

III. Conduite du projet :

1. Planification du projet :

Projet de Fin d'Etudes : Mise en Place d'un Système de Monitoring Industriel

Dans le cadre de la conduite du projet, la réalisation d'un planning à suivre tout au long du stage de fin d'études s'impose. Ainsi, après avoir déterminé le processus de développement adopté, l'étape suivante est de planifier les tâches selon ce processus et selon les contraintes du projet.

Mon stage de fin d'études a commencé le 14 Mars et s'est réparti pendant six mois sur les tâches suivantes comme la figure 1.5 montre :

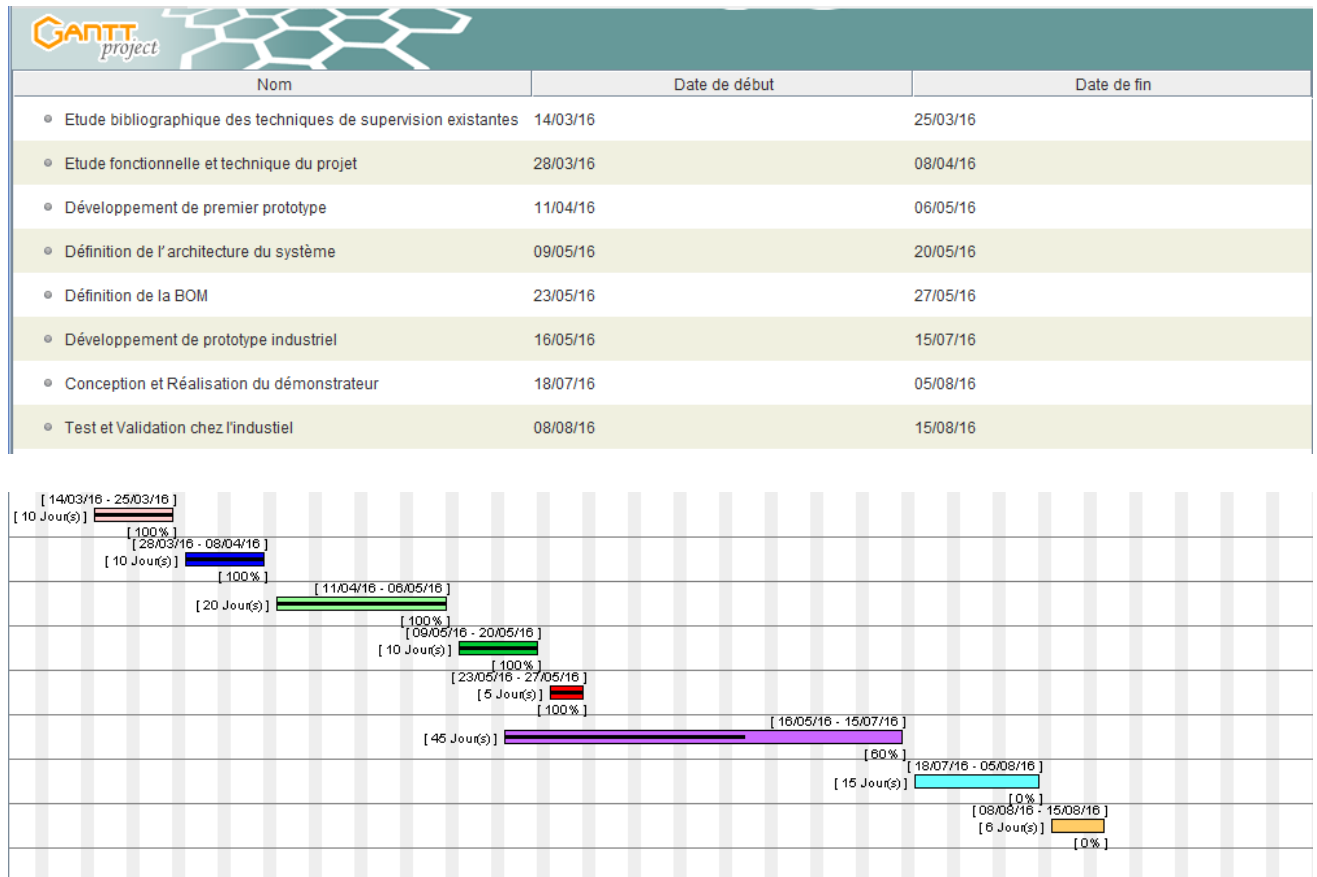


Figure 1.5 : Planning du projet

2. Etat de l'art :

L'efficacité d'un processus de production industriel est directement liée aux ressources matérielles et immatérielles utilisées lors de la production des produits, de ce fait un suivi en temps réel des données de production avec un contrôle à distance et à tout instant devient une exigence pour la survie et la compétitivité de toute société industrielle.

Donc les sociétés industrielles leader dans le marché industriel sont celles qui optent pour les nouvelles technologies en se basant sur des capteurs communicants et des méthodes d'acquisition des données plus développées.

Les différentes méthodes existantes pour l'acquisition des données industrielles sont :

- **Méthode purement manuelle** : la présence de l'opérateur est obligatoire pour collecter les données des machines en production (mesures, alarmes, retour d'état de fonctionnement...)
- **Méthode semi automatique (figure 1.6)** : PSEMS est une méthode adoptée par la société STMicroelectronics, développée afin de:
 - ➔ Collecter les changements d'état des équipements via des codes saisis par les opérateurs.
 - ➔ Contrôler des équipements et détecter les critiques.
 - ➔ Calculer et analyser les indicateurs liées aux performances de l'équipement (MTBF, MTBI) pour améliorer les performances de l'équipement.

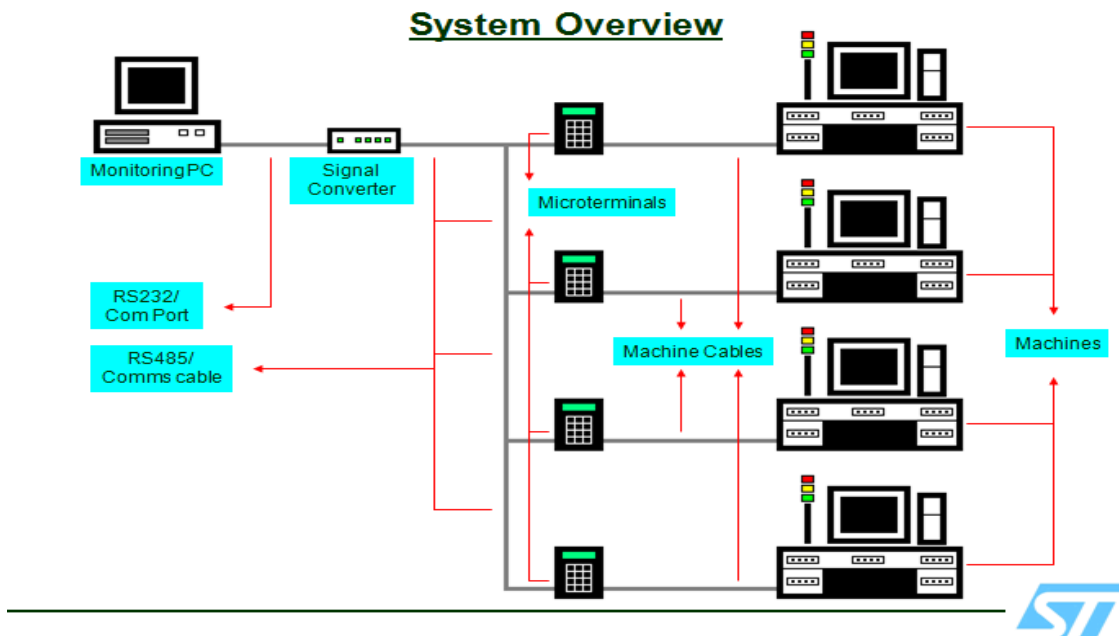


Figure 1.6 : Architecture du réseau PSEMS

Les machines : Toutes les machines surveillées sont connectées avec un microterminal pour lui envoyer des signaux haute / basse indiquant le temps UP / DOWN et le nombre de quantité unitaire.

Microterminal : sont utilisés pour collecter les données des machines (le temps UP / DOWN et le nombre de quantité unitaire) ainsi que des opérateurs (les codes de cause, le nombre de périphériques et l'opérateur ID).

Conversion de Données : Le convertisseur de signal est utilisé pour convertir le signal RS232 du PC vers le signal RS485 qui va à tous les Microterminaux et vice versa.

PC Surveillance : La fonction de PC de surveillance est de recueillir et stocker toutes les données reçues des Microterminaux et les mettre à jour via le serveur PSEMS avec horodatage.

- **Méthode SCADA** : un système de contrôle et d'acquisition de données permettant de traiter en temps réel un grand nombre de télémessures et de contrôler à distance des installations techniques pour déterminer le comportement d'un système afin d'optimiser ces performances.

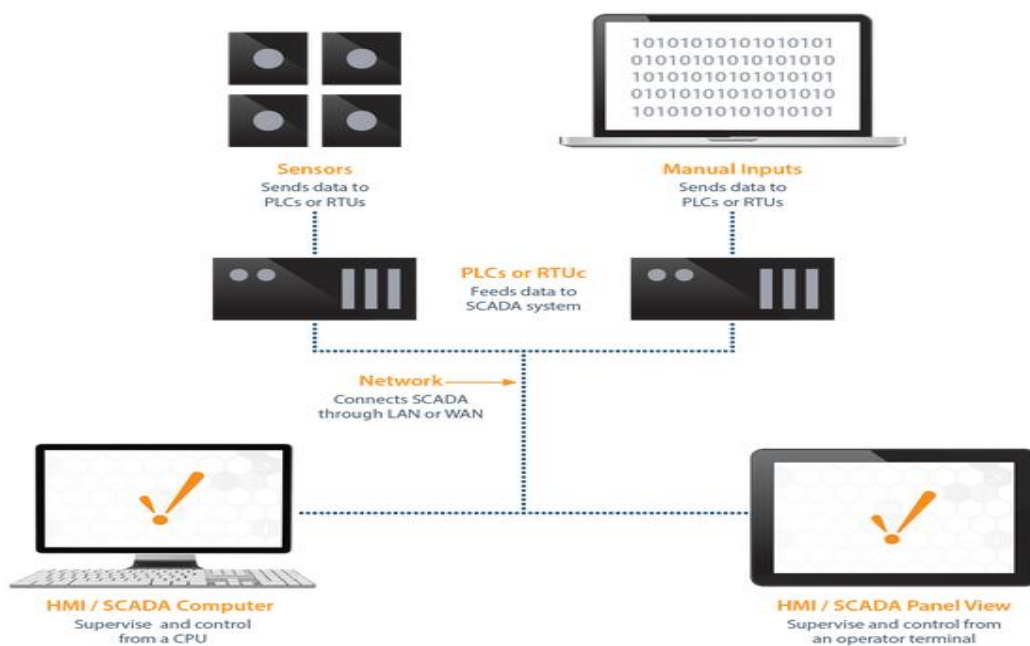


Figure 1.7 : Architecture SCADA

RTU : une unité de télémétrie à distance capable d'acquérir des données, de les enregistrer avec horodatage et puis les envoyer via un support de communication (Ethernet, sans fil, radio ...).

PLC : les automates programmables sont associés par l'interrogation automatique des équipements connectés.

HMI : interface homme machine permettant de communiquer avec les équipements de contrôle/commande dans le but de représenter les informations issues de ceux-ci sous formes compréhensibles par l'homme : Synoptiques, affichage lumineux, courbes etc... connectée en mode local (USB / Ethernet) ou mode distant (via GSM / GPRS ou Internet) pour afficher des données et accéder aux archives.

Conclusion

Après avoir situé le contexte général du projet, et étudié les différentes méthodes de supervision qui existent dans le marché industriel, le chapitre suivant détaillera le premier prototype qui a été réalisé pour superviser le comptage des pièces et la consommation de courant des machines industrielles.

Chapitre 2

Développement d'un premier prototype

*Dans ce chapitre je détaillerai
l'architecture et la plateforme de
développement hardware et software
adoptée pour réaliser le prototype
simplifié.*

Dans ce chapitre je décrirai l'architecture simplifiée et les deux niveaux matériel et logiciel utilisé pour réaliser le premier prototype qui vise à démontrer la faisabilité du projet.

I. Architecture du démonstrateur simplifié :

Au cours de mon stage plusieurs contraintes techniques ont été exigées par l'entreprise pour réaliser le projet, à savoir :

- ➔ La solution doit être développée avec un matériel performant et moins coûteux et qui répond aux contraintes et aux fonctionnalités de l'application .
- ➔ L'interface d'utilisation devrait être portable et accessible tout le temps .
- ➔ Le système doit être fonctionnel avec toute machine dans n'importe quel secteur d'activité.

Donc le premier exercice nous vise à développer un démonstrateur permettant l'acquisition et l'envoi sans fil et en temps réel, via Bluetooth 4.0 (bluetooth low energy), des données de la machine Wire-Bonder à savoir : le comptage des pièces produites, la détection de blocage des pièces , à travers le capteur de comptage de la machine, et la consommation énergétique, à travers le capteur de courant.

La figure 2.1 présente l'architecture de ce premier prototype :

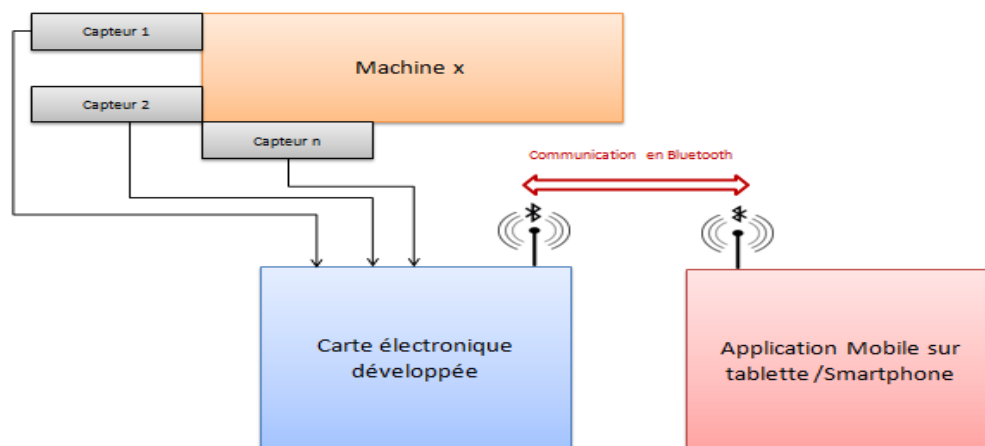


Figure 2.1 : Architecture de démonstrateur simplifié

Machine en production : la machine industrielle qui a été choisie pour faire la démonstration est bien la machine wire bonder (voir figure 2.2), K&S Maxµm Ultra Automatic Ball Bonder, qui assure les interconnexions des circuits intégrés à semi-conducteur par des fils d'or. La machine établit la connexion des fils au moyen d'un procédé de «collage de balle thermo sonique» qui utilise la chaleur, la force et l'énergie ultrasonore pour fusionner chaque extrémité du fil au dispositif. Cette démonstration reste valable avec n'importe quelle machine industrielle monophasé en production.

Maxµm Ultra Automatic Ball Bonder



Figure 2.2 : Machine Wire Bonder

Les capteurs : sont définies comme des composants techniques qui détectent un événement physique se rapportant au fonctionnement du système (présence d'une pièce, température, humidité, etc.) et traduit cet événement en un signal mesurable et exploitable par un PC par exemple. Ce signal est généralement électrique sous forme d'un signal basse tension ou basse courant.

Comme première phase de ce projet j'ai utilisé juste deux types de capteurs à savoir :

- **Capteur de comptage (voir figure 2.3) :** vise à détecter la présence et compter les pièces produites par la machine Wire-Bonder, et déclencher le blocage des pièces au cours de processus de production. Ce capteur est proprement installé sur la machine Wire-Bonder et délivre une sortie TTL (0-5V), qui sera abaissée à 3.3V par un diviseur de tension et reliée avec les entrées de la carte électronique à base de microcontrôleur STM32.



Figure 2.3 : Capteur de comptage

- **Capteur de courant (voir figure 2.4) :** lors de la démonstration, pour pouvoir calculer la consommation énergétique d'une machine, j'ai utilisé un tore qui s'installe autour, de tous les conducteurs actifs (3 phases et neutre) en triphasé, d'un seul conducteur vu que la machine Wire-Bonder est monophasée. Le capteur exploite l'effet Hall pour produire une tension qui est l'image du courant à mesurer. Sa sortie courant est analogique d'où la nécessité d'une conversion analogique-numérique au niveau de microcontrôleur, et une implémentation d'un montage à amplificateur opérationnel qui converti le courant en tension pour qu'il soit exploitable par la carte électronique.



Figure 2.4 : Capteur de courant

Quand ce tore est connecté à une charge linéaire, cette dernière soutire du réseau électrique un courant sinusoïdale (voir figure 2.5), à la même fréquence que la tension du réseau à un déphasage près, et donc la valeur efficace du courant est bien sa valeur maximale sur racine de deux.

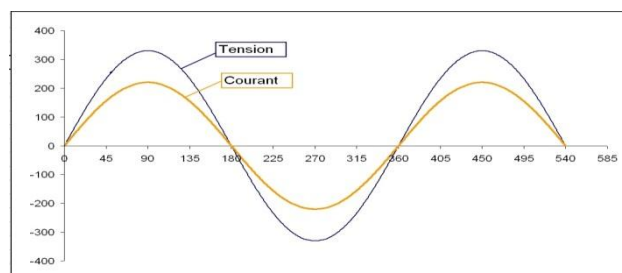


Figure 2.5 : Courant appelé par une charge linéaire

Toutefois quand il est connecté à une charge électrique non linéaire, cette dernière soutire de réseau électrique un courant périodique de forme non-sinusoïdale (voir figure 2.6), perturbé par les courants harmoniques, qui présentent les composantes sinusoïdales du courant électrique périodique décomposé en série de Fourier avec une fréquence multiple de la fréquence fondamentale, donc pour calculer la valeur efficace du courant il faut effectuer un échantillonnage du signal et implémenter sur le microcontrôleur une intégration numérique par la méthode rectangle : $I_{eff} =$

$$\sqrt{\frac{1}{T} \cdot \int_{t_0}^{t_0+T} i^2(t) \cdot dt}$$

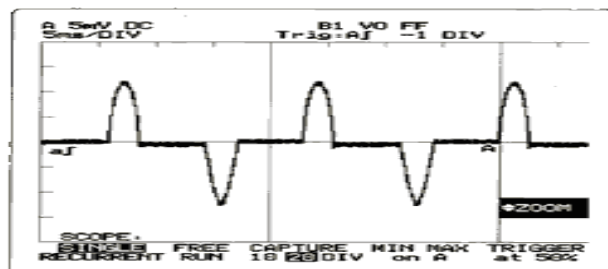


Figure 2.6 : Courant appelé par une charge non linéaire

II. Description matérielle :

1. Carte Nucleo-STM32L0:

L'unité de traitement est composée d'un microcontrôleur qui fera le traitement des données collectées des différents capteurs c'est pour cela j'étais amenée à faire une étude comparative entre les différents microcontrôleurs disponibles sur le marché, le tableau suivant illustre les caractéristiques de la famille des Arduino, Raspberry PI et STM32 et justifie le choix du microcontrôleur STM32:

	ARDUINO ATmega328	Raspberry PI	STM32
CPU	ATMEL de la famille AVR 8bits - 16MHz	ARMv6 700 MHz - Broadcom 2835	ARM(Cortex 32 Bits Risc avec les séries M0,M3,M4). 4 Séries: STM32L1(32Mhz,33DMIPS), STM32F1,STM32F2(120Mhz,150DMIPS),STM32F4(168Mhz,

			210 DMIPS).
SRAM	2 Ko, 1Ko (EEPROM)	256 Mo (version A) / 512 Mo (version B)	256 Mo
ports	USB	HDMI, 2x USB, Ethernet 10/100/1000, jack 3.5 mm, carte SD standard 32Go	USB, Ethernet, HDMI
Communications sérielles	UART, SPI/I2C (ajout des librairies)	UART, SPI, I2C	UART, SPI, I2C, RS232
GPIO	14 E/S numériques (dont 6 disposent d'une sortie PWM), 6 E/S analogiques	8 E/S en 3.3V	Package LQFP100/144 en 3.3V
CAN	Oui	Non	Oui
Prix unitaire (digikey)	23 USD	47 USD	14 USD

La famille des microprocesseurs STM32 de ST Microelectronics fournit une vaste gamme de périphériques autour d'un cœur d'ARM Cortex-M0/M3/M4, allant du simple GPIO (port d'entrée-sortie généraliste) et interface de communication série synchrone (SPI) ou asynchrone (RS232) aux interfaces aussi complexes que l'USB, Ethernet ou HDMI. Un point remarquable est qu'un certain nombre de ces processeurs possèdent deux convertisseurs analogiques-numériques, permettant un échantillonnage simultané de deux grandeurs analogiques. Cadencé sur un résonateur interne ou sur un quartz externe haute fréquence 8 MHz (multiplié en interne au maximum à 72 MHz), ce processeur est compatible pour des applications faibles consommations avec un mode veille dont le réveil s'obtient par une condition sur une horloge interne ou une interruption externe. La multiplicité des horloges et leur utilisation pour cadencer les divers périphériques est d'ailleurs un des aspects les plus déroutant dans la prise en main du STM32.

La carte STM32L0-Nucleo-64 (voir figure 2.7) est une plateforme de développement, pour les applications de faible consommation, qui permet d'établir des prototypes avec le microcontrôleur STM32 et assure la connectivité via deux connecteurs d'extensions le support ARDUINO™ et les ports ST morpho. Elle est équipée en série d'un ST-Link /V2 permettant la programmation directe depuis un port USB et supporte la librairie HAL (Hardware Abstraction Layer) qui contient des exemples de logiciels variés [1].

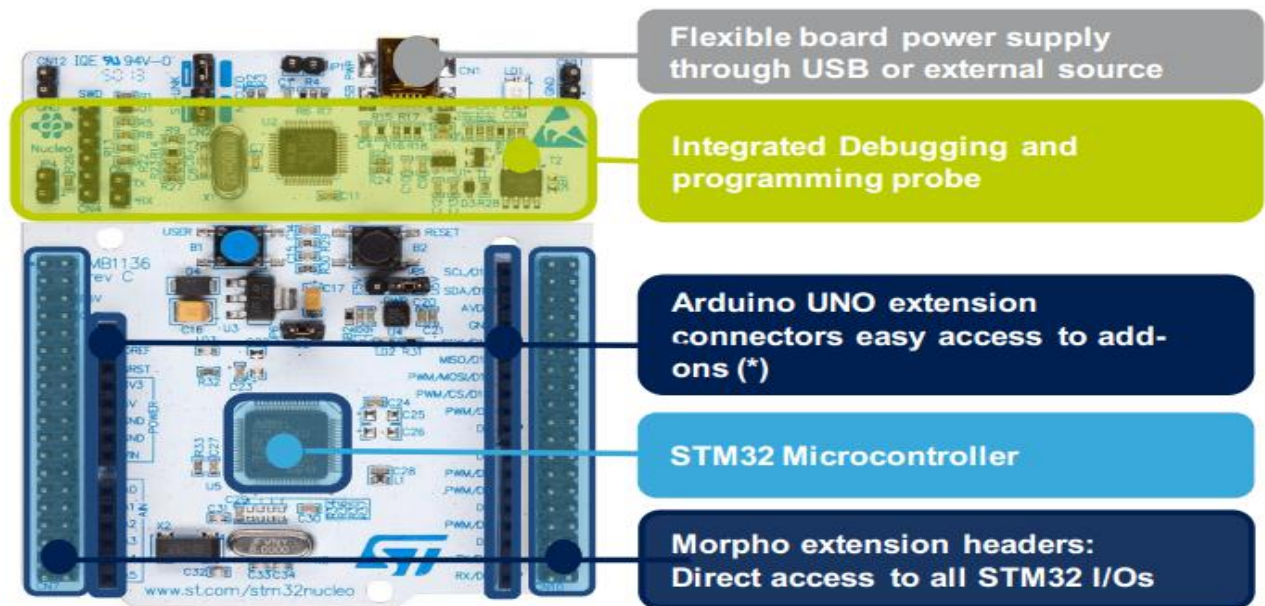


Figure 2.7 : Carte Nucleo-STM32L053R8

2. Carte BlueNRG:

Pour transmettre les données délivrées par les différents capteurs et les visualisées sur une application mobile, le projet exige l'utilisation d'une technologie de transmission sans fil qui couvre le maximum de distance, transmettre le maximum de données et surtout consomme la moindre d'énergie avec un faible coût. D'où le choix de la technologie Bluetooth Low Energy.

Bluetooth Low Energy (BLE), Bluetooth 4.0, est une technique de transmission sans fil créée par Nokia sous forme d'un standard ouvert basé sur Bluetooth pour le compléter sans le remplacer.

Comparé au Bluetooth (niveau d'émission plus élevé et portée plus grande), le BLE permet un débit du même ordre de grandeur (1 Mb/s) pour une consommation d'énergie 10 fois moindre avec une bande passante plus limitée.

Tout comme Bluetooth, BLE fonctionne dans la bande de fréquence 2,4 GHz mais BLE reste en mode veille en permanence, sauf quand une connexion est initiée. Les temps de connexion

actuels ne sont que quelques ms, contrairement à Bluetooth qui prendrait ~ 100ms. De plus Bluetooth peut gérer un grand nombre de données, mais consomme la batterie rapidement et coûte beaucoup plus. Or BLE est utilisé pour les applications qui ne nécessitent pas d'échanger de grandes quantités de données, et peut donc fonctionner sur batterie pendant des années à un coût moins cher.

La carte d'extension X-NUCLEO-IDB04A1 (voir figure 2.8) contient un processeur BLE conforme à la spécification Bluetooth V4.0. Elle est compatible avec les cartes STM32 Nucleo. Le BlueNRG peut agir en tant que maître ou esclave. L'ensemble de la carte Bluetooth à faible énergie fonctionne sur le noyau Cortex M0 intégré. Le BlueNRG offre la possibilité d'interfaçage avec des microcontrôleurs externes en utilisant la couche de transport SPI.

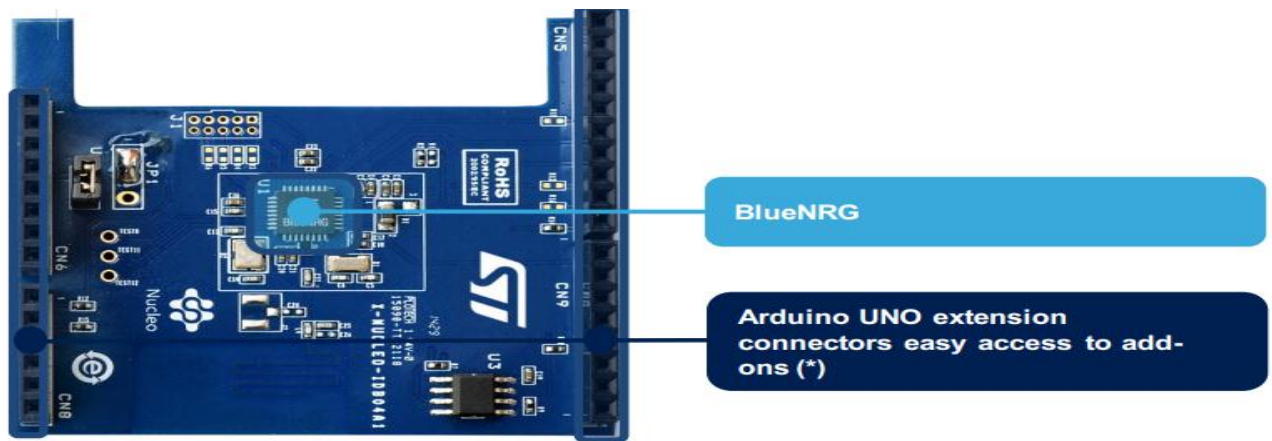


Figure 2.8 : Carte X-NUCLEO-IDB04A1 [3]

Par rapport au Bluetooth classique le Bluetooth Low Energy introduit le concept de profils afin d'assurer le maximum de compatibilité entre les produits des différents constructeurs de produits Bluetooth [1].

Les profils Bluetooth Low Energy sont tous superposés sur le GATT, utilisant le protocole GATT/ATT pour échanger des données. Dans la technologie Bluetooth classique, les profils définissent souvent leurs propres protocoles. Il en découle davantage de flexibilité, mais aussi une mise en œuvre plus complexe et une augmentation de la quantité de code à exécuter.

La figure 2.9 donne une brève description de protocole de Bluetooth Low Energy et illustre la hiérarchie entre les profils et les dépendances entre eux :

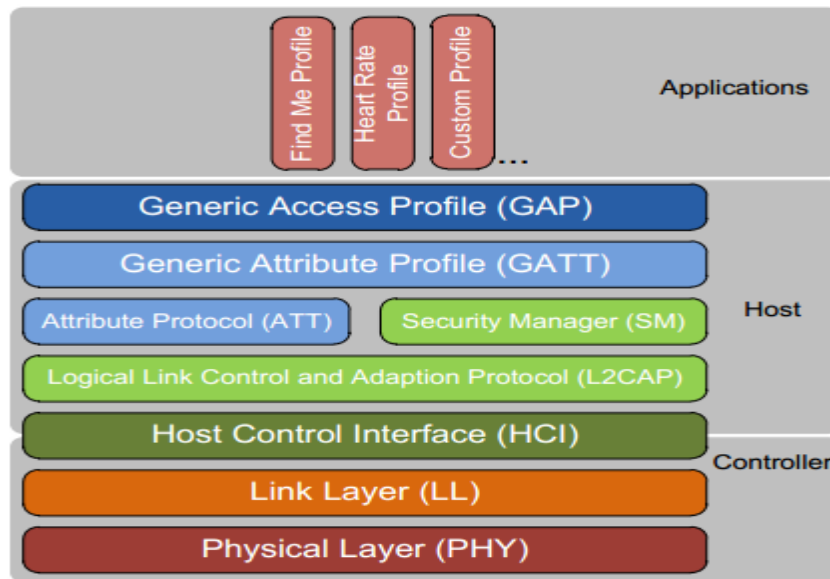


Figure 2.9 : Protocole BLE

- **GAP (Generic Access Profile)** : ce profil est LE profil de base qui doit être implémenté par tous les appareils Bluetooth. En effet, c'est celui qui définit les procédures génériques de découverte d'équipement, ainsi que de gestion de connexion aux autres appareils Bluetooth. Ce profil expose l'ensemble des caractéristiques de tous les équipements Bluetooth :

- ➔ Il expose les spécifications sur la représentation des propriétés Bluetooth : l'adresse Bluetooth, le nom d'un équipement, son type, le PIN number utilisé pour authentifier 2 périphériques.
- ➔ Il définit les « modes » génériques à tous les profils : discoverability mode (on peut le détecter), connectability mode (on peut s'y connecter), pairing mode (on peut créer un lien avec).
- ➔ Il définit les procédures générales qui peuvent être utilisées pour « découvrir » les propriétés basiques des équipements Bluetooth (nom, type...) qui sont « découvrables ».
- ➔ Il décrit les procédures générales de connexions à d'autres dispositifs Bluetooth.
- ➔ Il définit la procédure générale de création de liens entre des dispositifs Bluetooth.

Ce profil est celui dont tous les autres dépendent, et tous les profils « héritent » de ses caractéristiques.

- **ATT (Attribute Protocol)** : est un protocole de communication qui définit comment les données de l'application sont organisées et exploitées.

- **Le serveur GATT (Generic Attribute Profile) [5]** (voir figure 2.10) : est construit sur la base du protocole de communication ATT et établit des opérations et un environnement de travail pour le transport et le stockage des données par le protocole ATT.

Le serveur GATT définit deux rôles, client et serveur. Les rôles GATT ne sont pas nécessairement liés à un rôle GAP et ils peuvent être définies dans des couches plus hautes du profile. GATT et ATT ne sont pas sensibles à la couche transport et peuvent aussi être utilisés pour le Bluetooth classique et le Bluetooth Low Energy. Il est cependant indispensable pour utiliser le Bluetooth LE car c'est le serveur GATT qui se charge de découvrir les services.

Le serveur GATT sauvegarde les données sur la couche ATT et accepte les requêtes, commandes et confirmations ATT venant du client GATT.

Les périphériques sont vus comme des services et caractéristiques. Les services peuvent contenir une collection de caractéristiques :



Figure 2.10 : Hiérarchisation de GATT

Ainsi une caractéristique est une valeur au sein d'un service. Elle est définie par des propriétés qui montrent comment on accède à ces dernières valeurs et un descripteur qui décrit la valeur.

- **SM (Security Manager)** : cette couche implémente les mécanismes de sécurité et gère la supervision des différentes connexions, de l'authentification des appareils, et du chiffrement.

- **L2CAP (Logical Link Control and Adaptation Protocol)** : cette couche permet l'adaptation des protocoles supérieurs au réseau Bluetooth. Elle comporte un mécanisme permettant d'identifier le protocole de chaque paquet envoyé pour permettre à l'appareil distant de passer le paquet au bon protocole, une fois celui-ci récupéré.

Cette couche supporte la segmentation et le réassemblage, et le multiplexage de protocole.

- **HCI (Host Control Interface)** : c'est l'interface qui permet de relier la couche de contrôle matérielle avec la couche de firmware de l'utilisateur de protocole.
- **LL (Link Layer) & PHY (Physical Layer)** : Ces couches prennent en charge les tâches matérielles comme le contrôle du saut de fréquence et la synchronisation de horloges.

III. Description logicielle :

1. STM32CubeMX :



STM32Cube est un cadre de développement qui fournit des outils et des bibliothèques pour développer des applications C sur les plates-formes de la série STM32.

STM32Cube comprend la plate-forme STM32CubeL0 qui comprend le STM32Cube HAL (un logiciel embarqué STM32 de la couche d'abstraction, assurant la portabilité maximisée à travers STM32), plus un ensemble cohérent de composants middleware (comme RTOS, USB, FATFS, LwIP et STM32 détection tactile). Tous les utilitaires logiciels embarqués sont livrés avec un ensemble complet d'exemples comme la figure 2.11 montre :

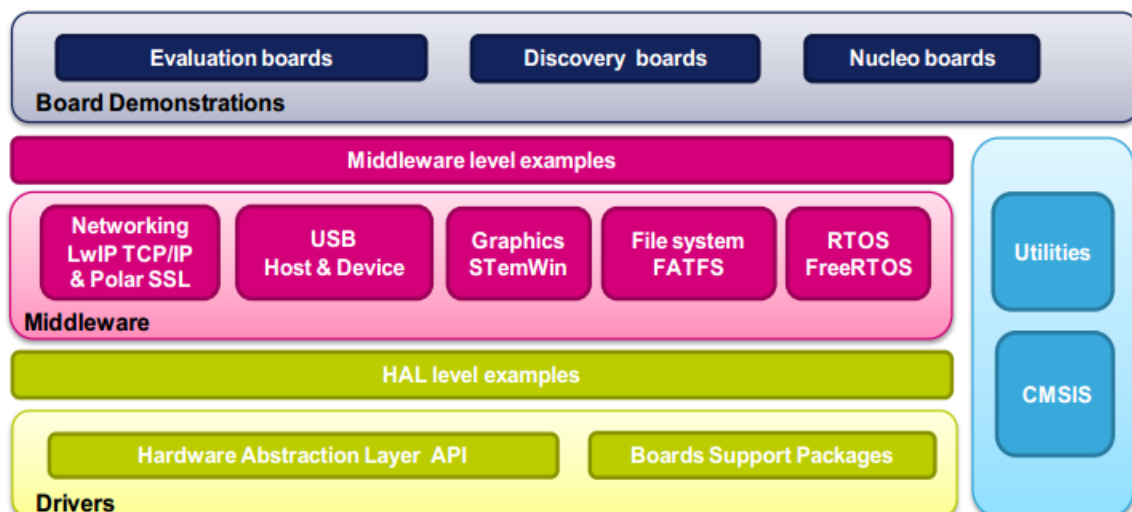


Figure 2.11 : Package de STM32Cube

Dans le cas de la carte Nucleo utilisée pour la réalisation du prototype, STM32CubeL0 rassemble en un seul package tout les composants génériques embarqués nécessaires pour développer une application sur le microcontrôleur STM32L0. STM32CubeL0 est entièrement compatible avec le générateur de code STM32CubeMX qui permet de générer le code d'initialisation. Le package comprend une couche d'abstraction matérielle de bas niveau (HAL) qui couvre le matériel de microcontrôleur, avec un ensemble vaste d'exemples en cours d'exécution sur les cartes STMicroelectronics.

2. Keil MDK-ARM 5.13 :



L'environnement de développement Keil [2] est un outil de programmation avec une gamme complète de codes. Keil se positionne comme référence importante en développement industriel et programmation (voir Annexes).

Il est caractérisé par :

- Comprend des compilateurs et des débogueurs C/C++.
- Système d'Exploitation en Temps-Réel (RTOS) intégré.
- Tableaux d'évaluation ARM, Cortex-M, Cortex-R.
- Assembleurs de macro inclus.
- Large gamme de produits logiciels et matériels.
- Assistance pour des centaines de familles de processeurs.
- Base de données de produits fréquemment mise à jour.

3. Android Studio :



Android est un système d'exploitation mobile, c'est un gros programme qui gère le matériel sur lequel il s'exécute (smartphone, tablette, ordinateur ou d'autres) ainsi que les logiciels qu'il permet d'exécuter.

L'application est développée sur l'IDE Android Studio, qui est un environnement de développement permettant principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page sur des écrans de résolutions variées simultanément [4].

l'architecture de la majorité des applications Android repose généralement sur l'assemblage des activités (fenêtres). Elles sont constituées d'un contexte et d'une interface graphique. Le contexte fait le lien entre l'application et le système alors que l'interface graphique se doit d'afficher à l'écran des données et permettre à l'utilisateur d'interagir avec l'activité.

Une activité n'a pas de contrôle direct sur son propre état, elle traverse un cycle de vie (voir figure 2.12) rythmé par les interactions avec le système et d'autres applications. chaque étape du cycle est représentée par une méthode :

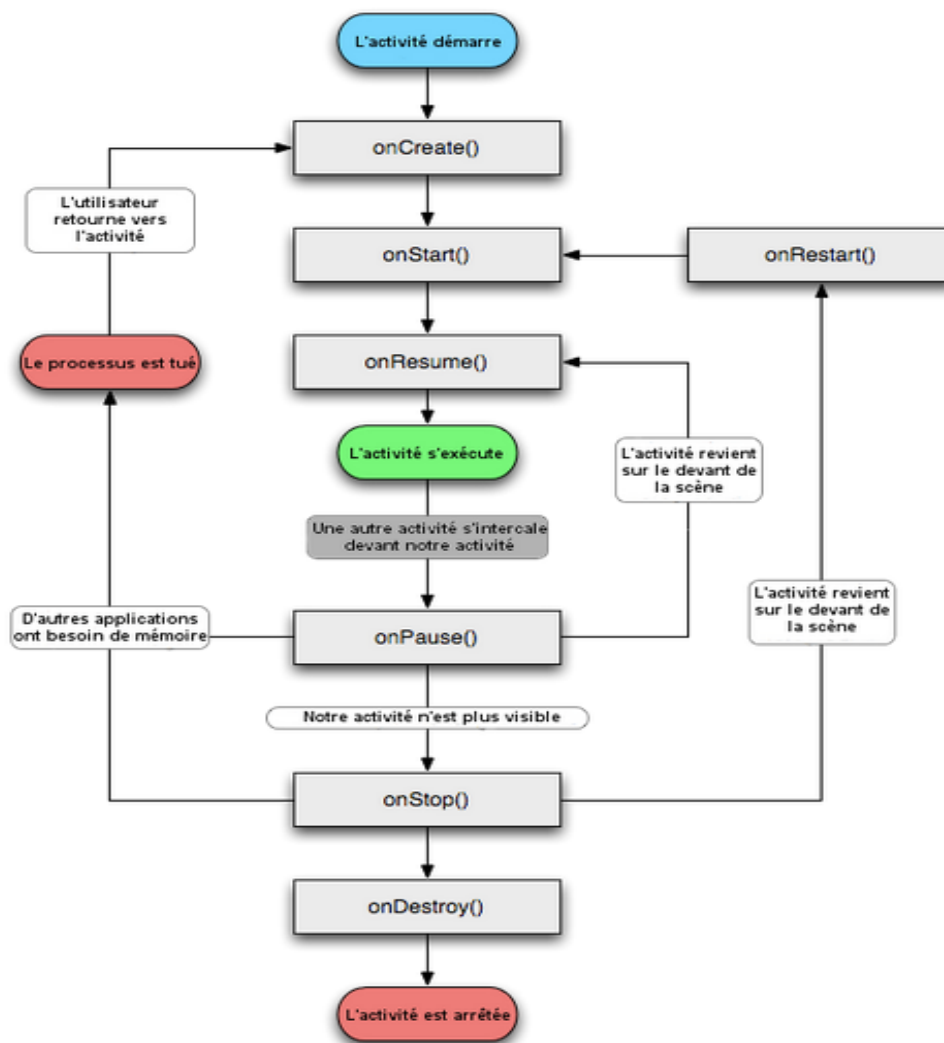


Figure 2.12 : Cycle de vie d'une activité

Au même titre que le langage Java est utile pour développer les applications Android, le langage XML l'est tout autant puisqu'il permet de mettre en place les différentes ressources et facilement retrouver l'emplacement des informations contenues dans les fichiers d'un tel projet et pouvoir les exploiter.

En général, il existe 5 types de ressources qui sont utilisées majoritairement dans une application Android :

- ➔drawable qui contient toutes les images matricielles et les fichiers XML décrivant des dessins simples.
- ➔layout qui contient toutes les interfaces qui seront attachées aux activités pour mettre en place les différentes vues.
- ➔menu qui contient toutes les déclarations d'éléments pour confectionner des menus.
- ➔raw qui contient toutes les autres ressources au format brut.
- ➔values qui contient des valeurs pour un large choix comme les chaînes de caractères, les dimensions, les couleurs, etc.

IV. Réalisation du prototype :

Ce premier prototype à pour objectif de démontrer la faisabilité de projet, '**Proof Of Concept**', en terme de récolte des données des capteurs issues d'une machine en production, et pouvoir les transmettre en temps réel via BLE afin de les afficher sur une application Android.

Dans cette démonstration on s'est limité au capteur de comptage des pièces produites et au capteur de courant : Le premier délivre une sortie numérique 5V qui sera abaissée à 3.3V à l'aide d'un diviseur de tension pour qu'elle sera exploitable par le microcontrôleur STM32. Le second délivre une sortie courant analogique d'où la nécessité d'une conversion analogique-numérique au niveau du microcontrôleur , et une implémentation d'un montage à amplificateur opérationnel qui converti le courant en tension pour qu'il soit exploitable par la carte électronique.

Une fois les données des capteurs sont traitées par le microcontrôleur, il les communique vers la carte BLE X-NUCLEO-IDB04A1 par SPI pour les transmettre et les afficher sous forme de graphes sur une application développée sous Android qui devrait tourner sur une Tablette ou Smartphone Android, comme la figure 2.13 montre :

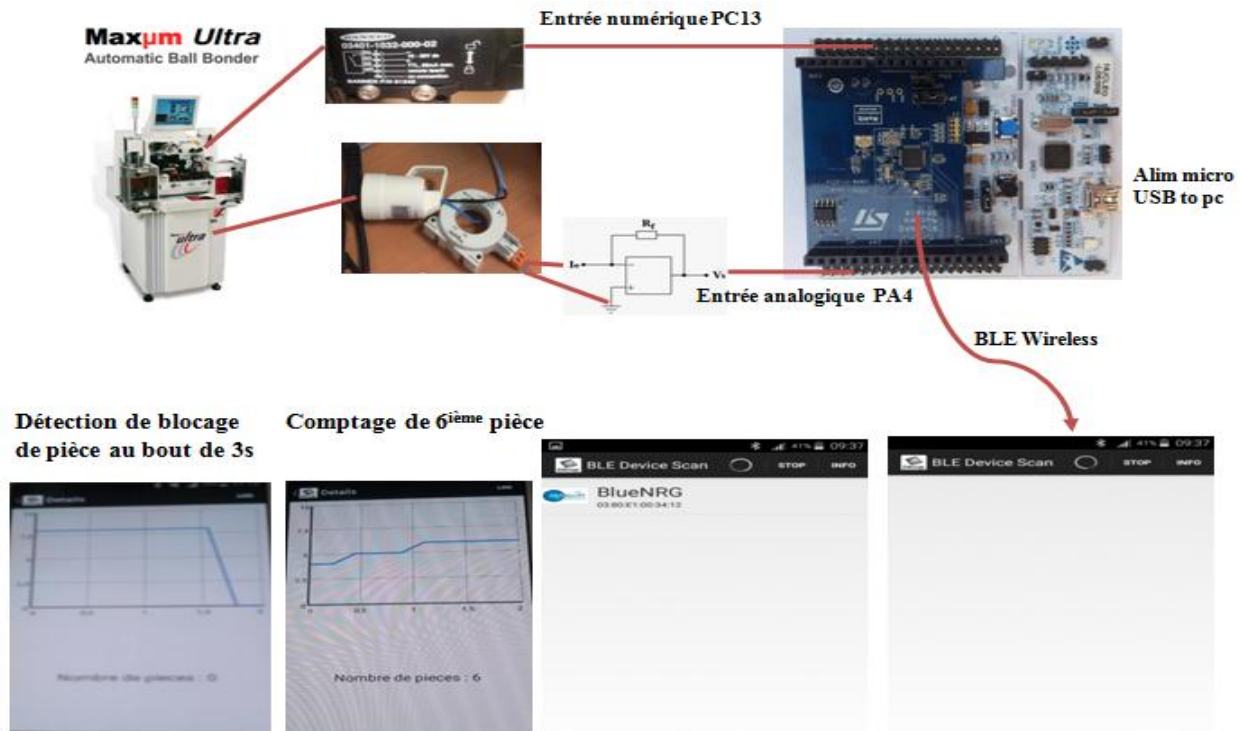


Figure 2.13 : Réalisation du prototype

Au niveau application (voir figure 2.14) , une fois l'utilisateur lance l'application le logo de MAScIR s'affiche et le Bluetooth de Tablette/Smartphone s'active directement. Ensuite en cliquant sur le bouton scan le processus de découverte des périphériques BLE se lance, lors de l'exécution de ce processus, le Smartphone / Tablette qui joue le rôle de maître écoute en permanence les trames de broadcast envoyées par les périphériques (BLE), qui jouent le rôle d'esclaves.

Une fois les trames BLE identifiées par le maître, les appareils détectés seront ajoutés sur une liste et affichés pour l'utilisateur. Lors de la sélection d'un appareil, un autre processus commence pour envoyer une requête de connexion à l'esclave BLE qui devra répondre par l'envoi d'une confirmation de connexion.

Lorsque la connexion est établie, le maître demande à chaque esclave connecté de lui envoyer une liste de ces services. Une fois l'UUID de chaque service est reçu, le maître spécifie le service voulu à partir duquel il va lire régulièrement les valeurs des capteurs, contenues dans les caractéristiques, et à chaque fois que ces valeurs sont soumises à des mises à jour, le Smartphone / Tablette sera notifié pour pouvoir afficher en temps réel les valeurs exactes des capteurs sur l'interface graphique.

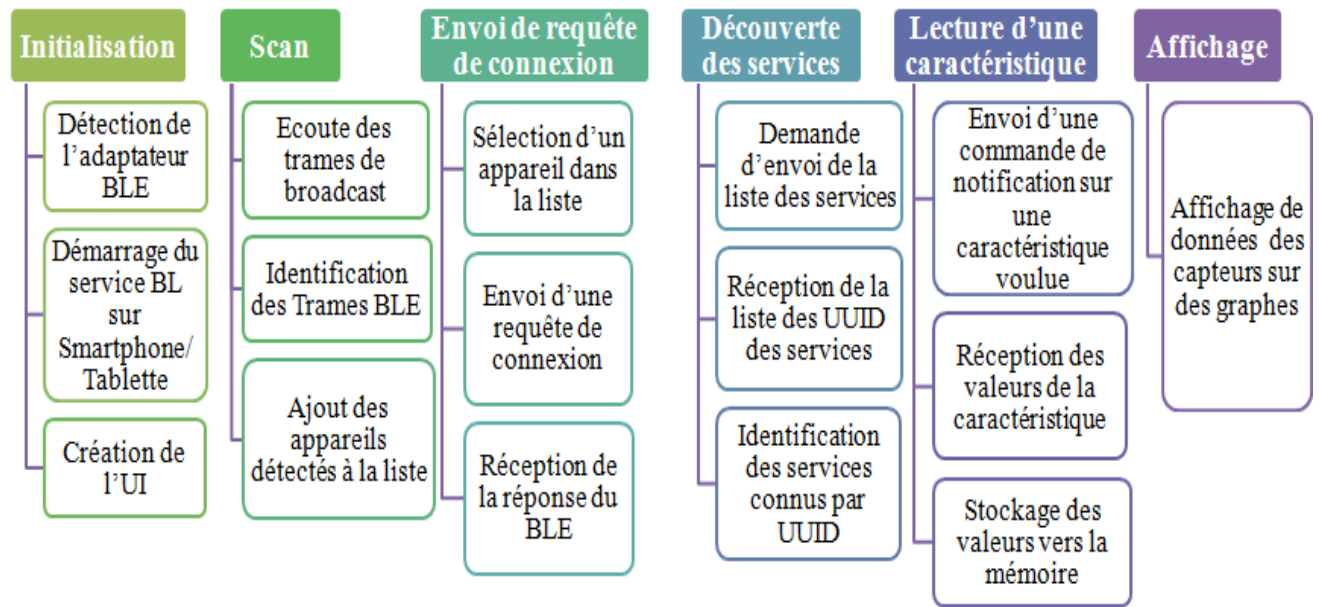


Figure 2.14 : Etapes de l'application Android

Conclusion

Dans ce chapitre j'ai décrit la plateforme matérielle et logicielle utilisée pour développer le premier prototype qui vise à tester en interne la faisabilité du projet. Pour l'améliorer et l'adapter aux exigences industrielles, j'expliquerai dans le chapitre suivant l'architecture générale du système, la nouvelle plateforme matérielle utilisée ainsi que les nouveaux aspects de développement de l'interface graphique.

Chapitre 3

Développement d'un démonstrateur industriel

Dans ce chapitre je décrirai les différentes améliorations apportées au premier prototype pour pouvoir réaliser un démonstrateur qui répond aux exigences industrielles.

Dans ce chapitre j'expliquerai la nouvelle architecture adoptée pour réaliser un démonstrateur qui répond aux exigences industrielles, les nouveaux outils utilisés pour l'acquisition des données ainsi que les améliorations qui seront apportées au système pour que l'utilisateur puisse superviser le comportement de ces machines à n'importe quel moment et de n'importe où.

I. Architecture générale du système :

Dans le secteur industriel, pour réaliser un tel système qui permet de collecter en permanence les données issues des capteurs de chaque machine, certaines exigences s'imposent :

- Avoir un historique permettant de comparer les valeurs reçues par les capteurs à tout moment.
- Stockage des données afin de garder une traçabilité dans le cas d'une éventuelle défaillance.
- Augmenter la portée de transfert de données .
- Utiliser un matériel performant et moins coûteux.
- Développer une application Android fonctionnel avec toute machine dans n'importe quel secteur d'activité.

Afin de répondre à ces exigences, l'architecture du premier prototype a été améliorée (voir figure 3.1). Pour chaque machine dans l'usine, les données issues des capteurs seront traitées par un microcontrôleur, transmis par Wifi et stockées dans un serveur. L'application se connecte au serveur local qui contient une base de données via un réseau local et si le réseau local est connecté à Internet la tablette peut se connecter au serveur via internet pour récupérer les données des capteurs et les afficher pour l'utilisateur.

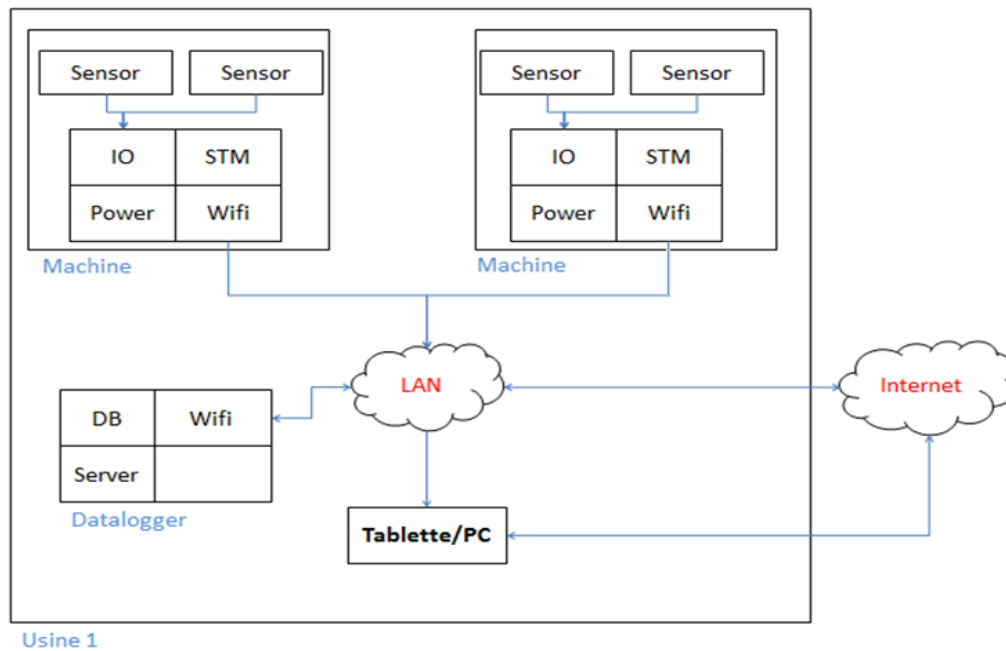


Figure 3.1 : Architecture générale du système

II. Description matérielle :

Vu que le premier prototype n'assure pas le stockage des données et la connexion au réseau local, je suis entrain de développer un démonstrateur qui répond aux exigences industrielles en terme de traçabilité et de portée de transfert des données. Jusqu'à présent l'architecture (voir figure 3.2) que j'ai définis pour ce prototype et sur laquelle j'ai fais mes tests est la suivante :

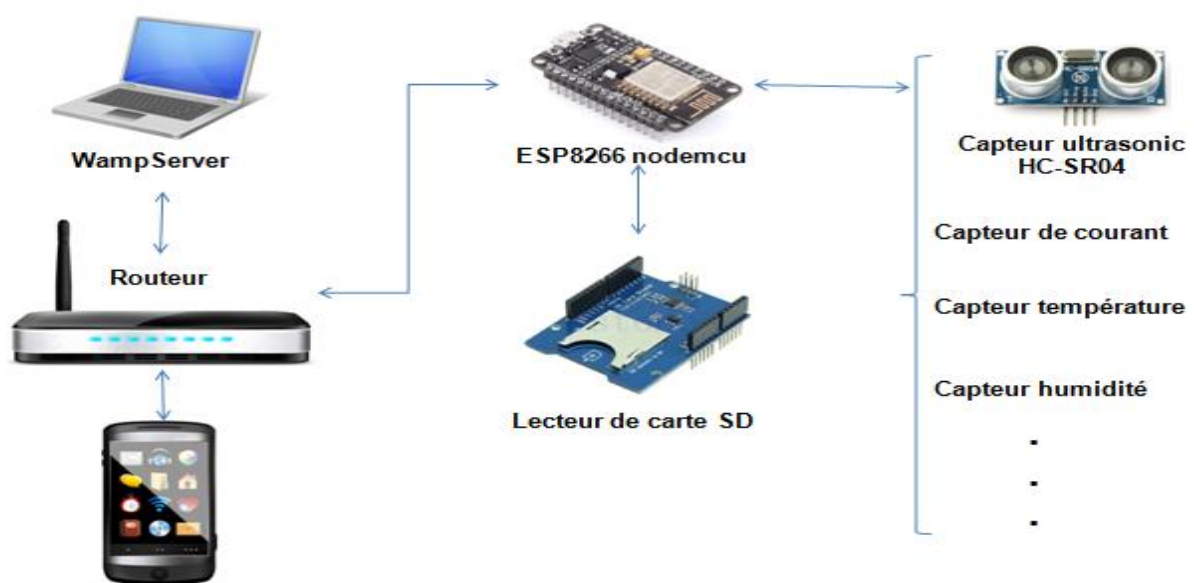


Figure 3.2 : Architecture du démonstrateur industriel

1. Ultrasonic HC-SR04 :

Au niveau de l'acquisition des données, la plupart des machines industrielles ne disposent pas de leurs propres capteurs et surtout le capteur de comptage, d'où l'idée d'utiliser un capteur ultrasonique HC-SR04 (voir figure 3.3) comme capteur de comptage pour faire des tests. Le capteur ultrasonique HC-SR04 utilise la broche Trig sur laquelle envoi un signal pour activer le déclenchement d'une impulsion, si le signal revient sur la broche Echo la sortie passe au niveau haut ce qui permet de détecter la présence d'une pièce produite ou non.



Figure 3.3 : Capteur ultrasonique HC-SR04

2. ESP8266-Wifi :

Au niveau transmission des données sans fil, pour pouvoir se connecter au réseau local j'ai adopté une solution WIFI permettant une grande portée pour transmettre les données et assurant une compatibilité avec les microcontrôleurs et une facilité de programmation.

Vu que les données délivrées par les capteurs ne nécessitent pas un traitement énorme par un microcontrôleur robuste genre STM32, le microcontrôleur ESP8266-WiFi (voir figure 3.4) de faible consommation d'énergie reste capable de traiter les données, de se connecter à un réseau wifi et d'échanger les données à travers l'antenne dont il dispose [6].



Figure 3.4 : Module ESP8266-Wifi

Ce module a une capacité de traitement et de stockage assez puissante ce qui lui permet d'être intégré avec les capteurs et dispositifs spécifiques à travers ses GPIO avec un développement minimal à l'avance et un chargement minimal lors de l'exécution en utilisant IDE de l'Arduino.

Il est caractérisé par :

- Protocole 802.11 b/g/n.
- Configuration Wifi en mode point à point / point d'accès.
- Protocole TCP/IP intégré.
- Cœur cadencé à 80 MHz et au niveau logique de 3,3V.
- Mémoire Flash de 4MO.
- Protocole de communication UART/SPI/I2C/SDIO.
- Port USB-série pour télécharger le code avec une vitesse de 921600 baud.
- 9 pins GPIO (voir Annexes).
- Une entrée analogique de 1V max.
- Consommation en mode standby < 1mW

3. Wamp Server :



le datalogging des données est assuré par le serveur WampServer, une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMysqlAdmin pour gérer plus facilement les bases de données.

Le serveur a été créé sur PC, muni d'une base de données SQL, et qui a la possibilité de communiquer avec une application Android en utilisant un protocole de communication client-serveur standard HTTP (HyperText Transfer Protocol) avec des trames JSON. Les deux s'échangent des requêtes via le port 80.

Le concept de la base de données SQL est simple. L'utilisateur de l'application Android peut surveiller les machines en production qui s'ajoutent au système, chaque machine est muni d'un certain nombre de capteurs qui délivrent les valeurs qui seront visualisées par l'utilisateur. La structure de la base de données de ce système est comme la figure 3.5 et la figure 3.6 montrent:

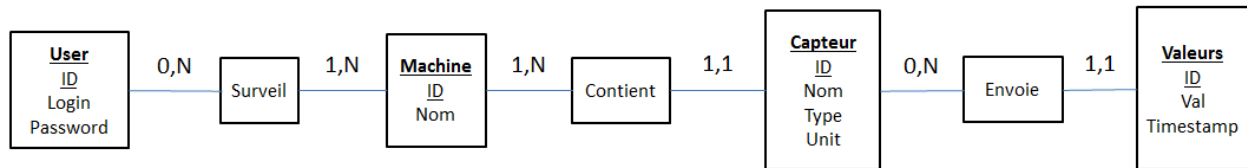


Figure 3.5 : Modèle conceptuel de données

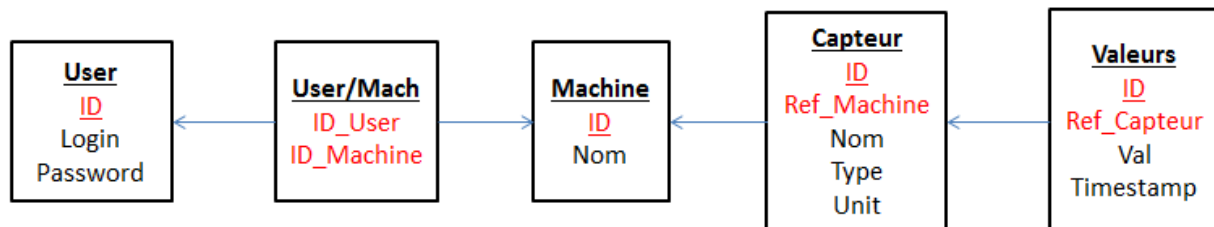


Figure 3.6 : Modèle logique de données

Comme solution de secours, lorsque le serveur est désactivé les données des capteurs seront stockées sur une carte SD et ré-envoyées vers le serveur lorsqu'il est disponible. L'ESP8266 communique avec la carte SD via le protocole de communication SPI.

NB : Dans le protocole HTTP, une méthode est une commande spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode. Il existe de nombreuses méthodes, les plus courantes étant GET et POST :

GET : C'est la méthode la plus courante pour demander une ressource. Une requête GET est sans effet sur la ressource, il doit être possible de répéter la requête sans effet.

POST : Cette méthode est utilisée pour transmettre des données en vue d'un traitement à une ressource. L'URL fourni est l'URL d'une ressource à laquelle s'appliqueront les données envoyées. Le résultat peut être la création de nouvelles ressources ou la modification de ressources existantes.

III. Architecture de l'application Android :

L'application Android vise toujours à enregistrer et à afficher sur interface graphique, les données provenant des divers capteurs installés sur plusieurs machines.

L'application devient de plus en plus robuste comme la figure 3.7 montre:

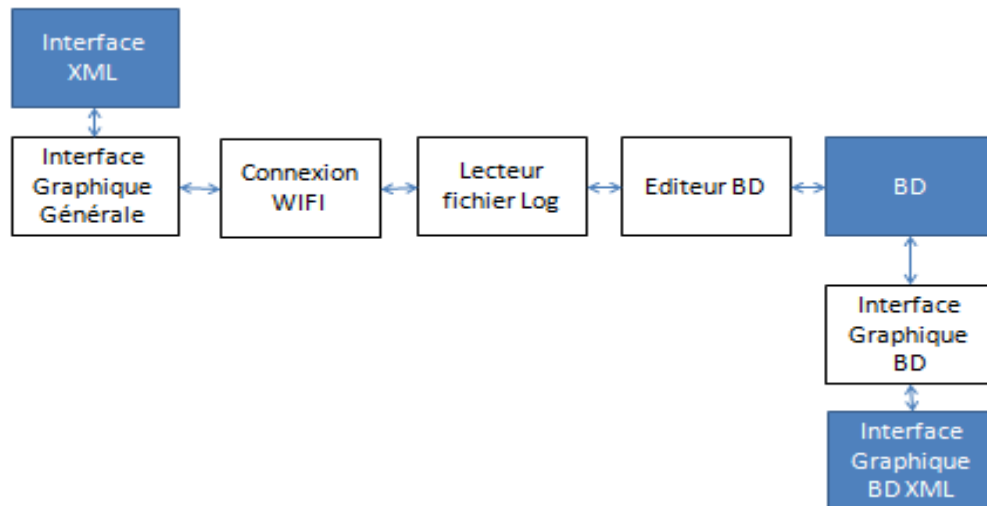


Figure 3.7 : Architecture de l'application Android

L'interface graphique est décrite en XML pour gérer les événements de changements de page, clics de boutons, affichage de graphe.....en se basant sur des multiples de classes fonctionnelles qui se complètent.

La connexion Wifi se fait entre le serveur qui a le profil de Maître, et les nœuds de capteurs esclaves. Le serveur leur envoie une requête de lecture de fichier log pour construire ses bases de données, et les nœuds ESP8266 lui transmettent un fichier log contenant les données de capteurs. Ce fichier log est stocké sur une carte SD.

L'application se connecte à un serveur local qui contient une base de données qui stocke les données issues des capteurs de chaque machine, via un réseau Local ou via Internet. Elle télécharge cette base de données en activant une requête PHP sur le serveur qui à son tour envoie à l'application une trame JSON qui contient les données nécessaires.

Sur l'application une classe est dédiée à la lecture intelligente des fichiers logs pour déterminer les logs de chaque capteur issu de chaque machine, et pour détecter la présence d'une nouvelle machine afin de l'ajouter à la base de données.

L'application comporte une base de données SQLite pour stocker les données des fichiers logs, une classe dédiée comporte les fonctions de lecture, écriture, suppression, création.....de tables dans la base de données.

Une autre classe se charge de modifier la base de données en se basant sur les fichiers logs reçus, ou en se basant sur la configuration de l'utilisateur.

L'application (voir figure 3.8) permet à l'utilisateur d'accéder aux machines qui existent dans la base de données et d'afficher leur paramètres. Elle autorise le serveur pour accepter et enregistrer les données mises à jour dans la base de données :

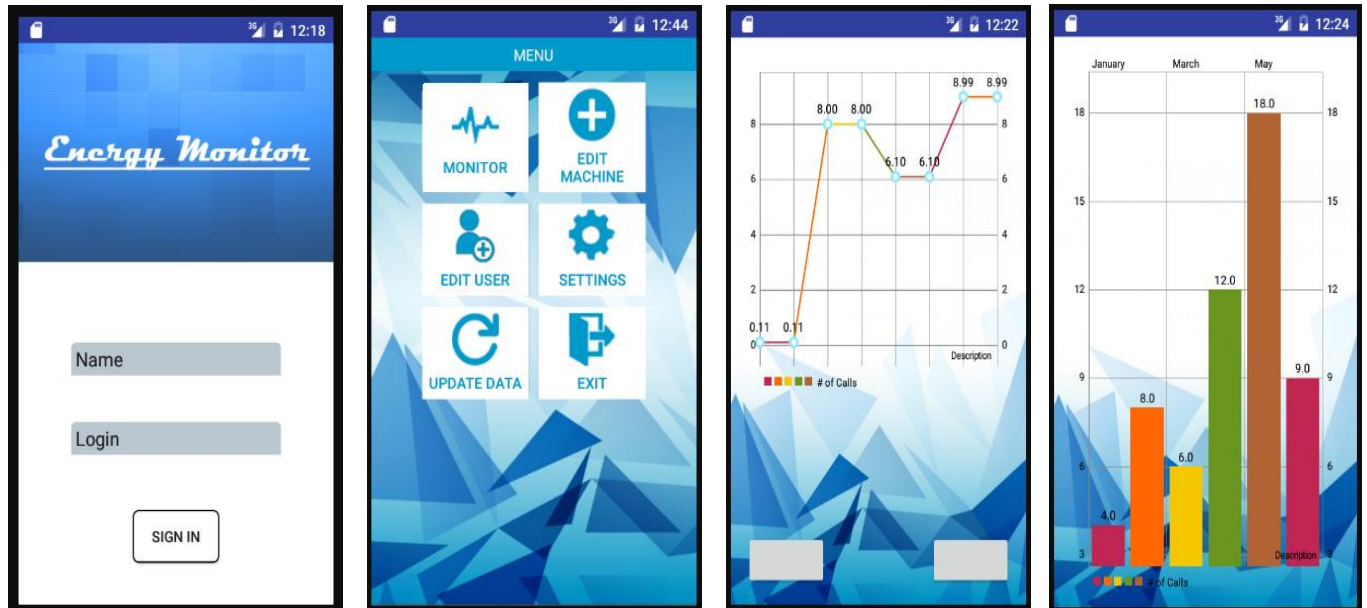


Figure 3.8 : Application Android

IV. Améliorations futures :

Pour améliorer notre application certaines tâches sont entrain d'être finalisées et d'autres doivent être réalisées. Dans ce qui suit, je listerai les principales de ces tâches.

1. Implémentation des protocoles de communication :

Pour assurer la communication entre les nœuds de capteurs, le serveur et l'application Android, on vise à implémenter des protocoles de communications pour organiser le flux d'informations entre le serveur et le microcontrôleur ESP8266 et entre le serveur et l'application sur tablette pour que le système soit plus consistant et plus sécurisé.

Le diagramme qui décrit le fonctionnement du microcontrôleur ESP8266 vise à traiter l'envoi des données des capteurs à la demande du serveur d'une part, et d'autres part le stockage des données sur carte SD lorsque le serveur n'est pas disponible.

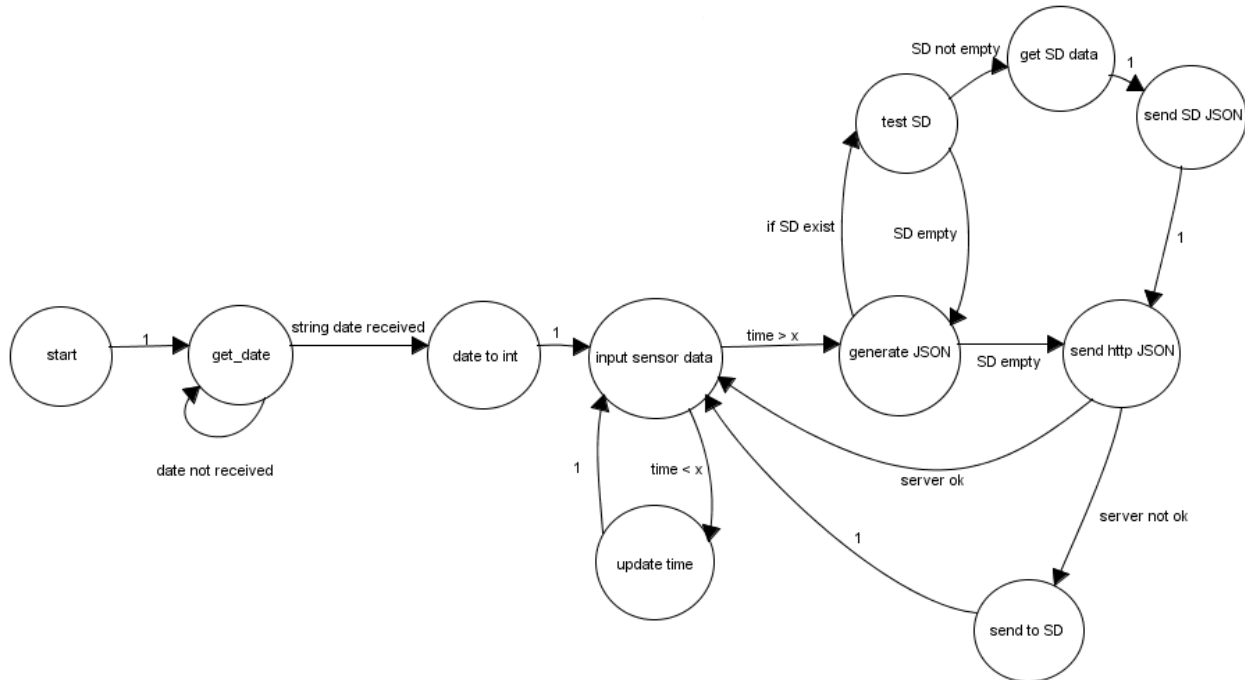


Figure 3.9 : Fonctionnement de microcontrôleur ESP8266

Le serveur est chargé de commander les nœuds de capteurs pour envoyer leurs données afin de les stocker dans les bases de données

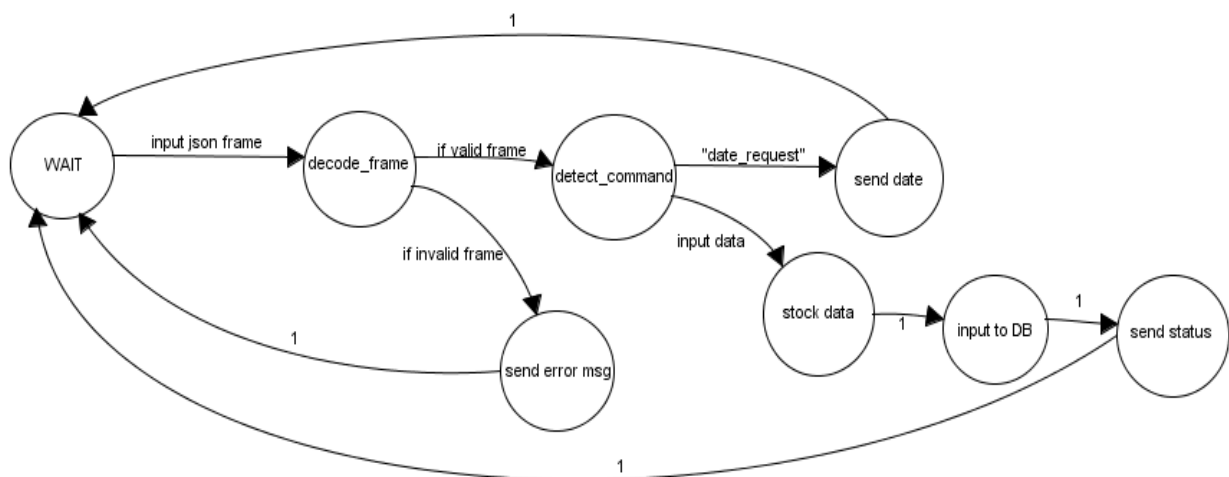


Figure 3.10 : Communication entre serveur et microcontrôleur

Le serveur est chargé aussi de répondre en temps réel aux commandes de l'application Android.

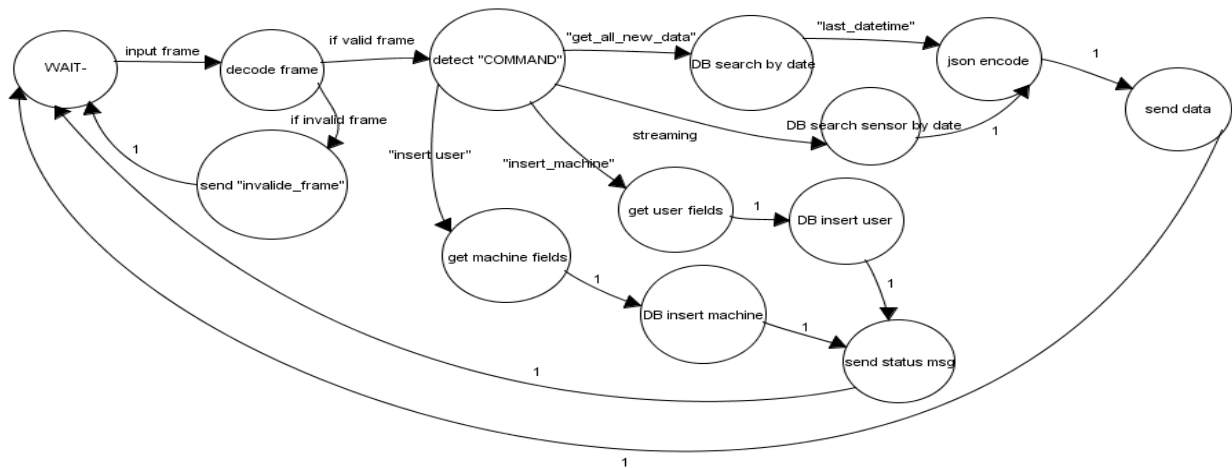


Figure 3.11 : Communication entre serveur et application

2. Centrale de mesure :

Le DIRIS A20 est une centrale de mesure qui met à disposition de l'utilisateur toutes les mesures nécessaires pour mener à bien les projets d'efficacité énergétique et assurer la surveillance de la distribution électrique.

- Facile d'utilisation
- Grâce à son large écran rétro éclairé multi affichage avec 4 touches d'accès direct, le DIRIS A20 est facile d'utilisation.
- Conforme à la CEI 61557-12, un dénominateur commun à l'ensemble des PMD (Performance Monitoring Devices). Respecter cette norme est l'assurance d'un haut niveau de performances tant sur les aspects métrologiques, que mécaniques et environnementaux (CEM, température, etc.)
- Détecte les erreurs de câblage
- Le DIRIS A20 est pourvu d'une fonction de correction des erreurs du raccordement de TC.



Figure 3.12 : Centrale de mesure DIRIS A20

Dans le cas de ce projet dédié pour les petits et moyens entreprises et vu le coût élevé de ces centrales de mesures, l'idée était de commander une centrale de mesure pour mesurer la consommation énergétique d'une ligne de production.

DIRIS A20 communique à partir d'un protocole JBUS/MODBUS qui implique un dialogue selon une structure maître/esclave. Deux dialogues sont possibles:

- le maître dialogue avec un esclave (DIRIS) et attend sa réponse.
- le maître dialogue avec tous les esclaves (DIRIS) sans attendre leur réponse.

Le protocole ModBus consiste à échanger des trames entre un maître et des esclaves en utilisant une communication série. Le maître envoie une trame de requête aux esclaves, et ils répondent par une trame similaire. Le format des Trames échangées est :



Le mode de communication est le mode RTU (Remote Terminal Unit) avec des caractères hexadécimaux composés au minimum de 8 bits.

DIRIS A20 est en mode esclave, pour extraire la donnée de puissance active, un développement préalable d'une application Maître sur le microcontrôleur STM32 qui devra envoyer la trame de requête. Une fois la requête est envoyée, on devrait recevoir une trame similaire qui contient la donnée de puissance active dans la zone Données.

Une fois les protocoles de communication entre le serveur et l'ESP8266 et entre le serveur et l'application Android sont bien implémentés ainsi que l'implémentation de protocole ModBus au niveau de la centrale de mesure, je vais passer à l'étape de conception d'un module propre à MAScIR pour finalement le tester et le valider chez l'industriel.

Conclusion

Ce chapitre définit l'architecture générale adoptée, et met l'accent sur l'ensemble des protocoles de communication à implémentés sur le microcontrôleur et le serveur pour assurer le fonctionnement d'un tel système de monitoring industriel et pouvoir le concevoir, ce qui est bien l'objectif de la période de stage qui reste.

Conclusion générale

Au début de mon stage l'objectif était trop général, il s'intitulait : « la mise en place d'un système de monitoring industriel ». Alors une fois j'ai fait une étude bibliographique des différentes méthodes d'acquisition des données qui existent dans le milieu industriel, il semblait important de mettre en place un outil de supervision à distance et "en temps réel" des indicateurs de performance industriel "SMI", moins coûteux et facile à installer permettant de répondre aux besoins des sociétés industrielles en terme d'optimisation de la gestion d'énergie et de productivité.

La période de la réalisation de ce projet s'étalait sur six mois, divisée en trois phases. La première phase accomplie a été considérée comme étant 'Proof Of Concept' permettant de prouver la faisabilité du projet par le développement d'un démonstrateur simplifié assurant le transfert des données issues des capteurs de chaque machine via Bluetooth Low Energy vers une application Android.

En cours de développement, la deuxième phase vise à mettre en place un démonstrateur robuste qui répond aux exigences de sécurité et de traçabilité industrielles, et qui communique les données des capteurs par Wifi et les stockent sur un serveur, avec une application Android qui consulte en permanence pour récupérer les données mises à jour de chaque capteur relié à chaque machine industrielle.

Comme perspective, J'aurais l'occasion dans la phase finale de faire la conception de notre propre carte électronique et réaliser son PCB pour pouvoir finalement la tester et la valider chez l'industriel.

Cette durée de stage était un excellent exercice professionnel qui m'a permis d'appliquer les différentes connaissances acquises durant mes études à l'FSTF et notamment de développer mes compétences en termes de savoir-faire et savoir être. En outre, ce stage était l'occasion pour améliorer ma méthodologie de travail et développer mon esprit d'équipe.

Bibliographie

[1] : STMICROELECTRONICS. STM32L0. [en ligne] disponible sur :

http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-l053r8.html

[2]. ARMKEIL. µVision® IDE. [en ligne] disponible sur :

<http://www2.keil.com/mdk5/uvision/>

[3]. STM32 Open Development Environment. X-NUCLEO-IDB04A1. [en ligne] disponible sur :

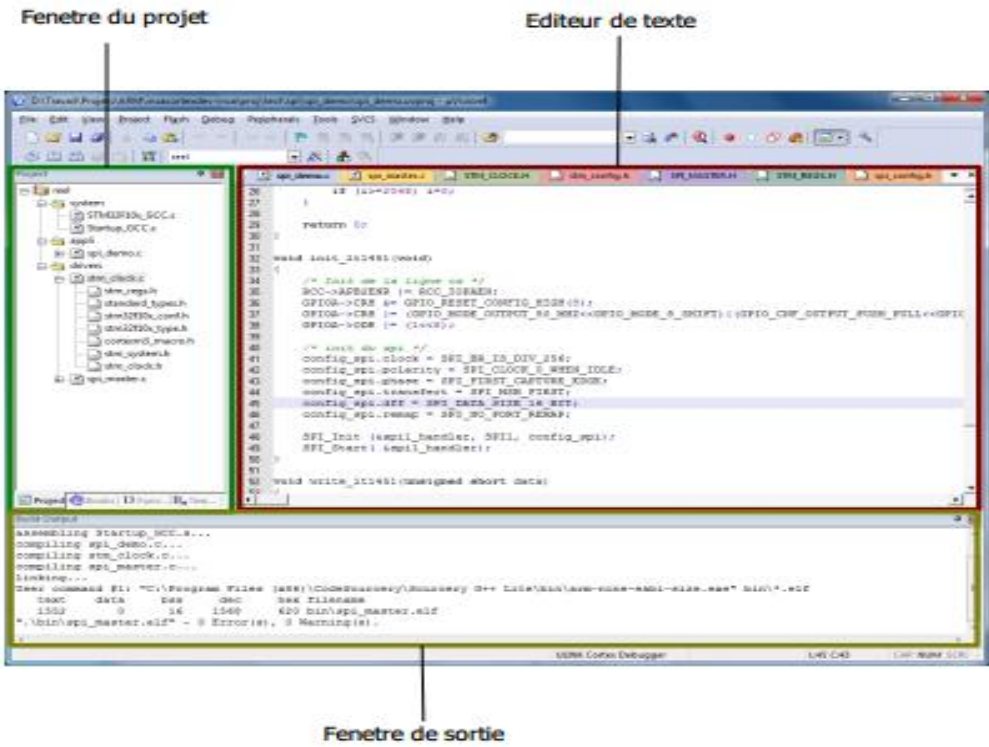
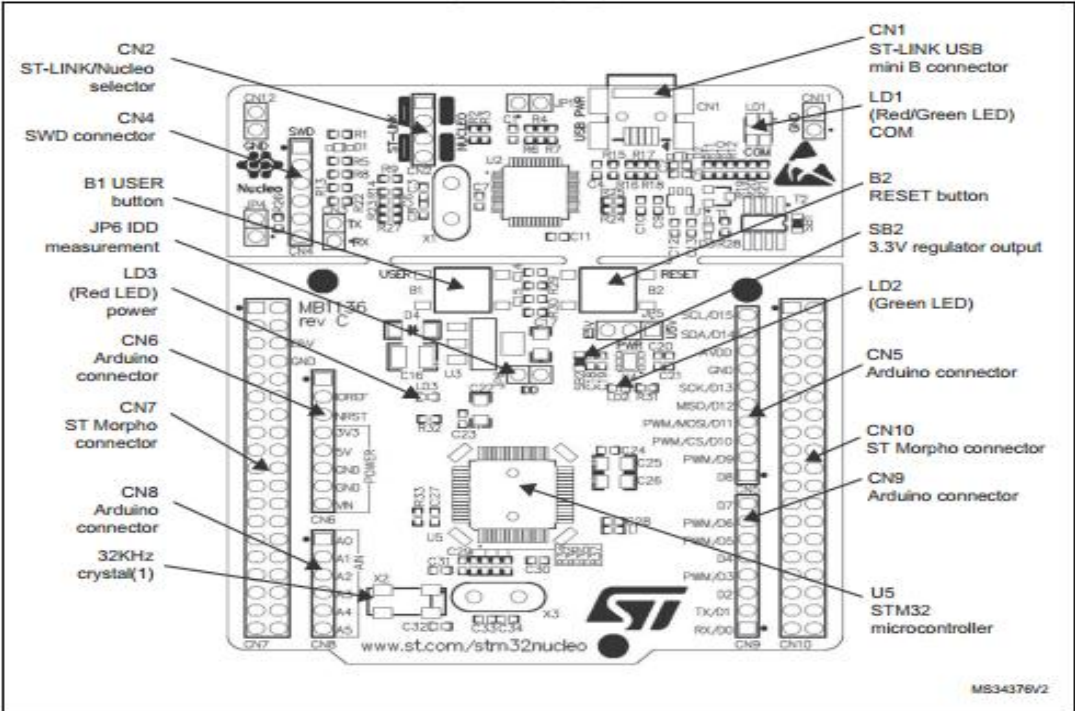
http://image.eccn.com/gcszj/st/150729/webinar_ppt.pdf

[4]. OPENCLASSROOMS. Android. [en ligne] disponible sur :

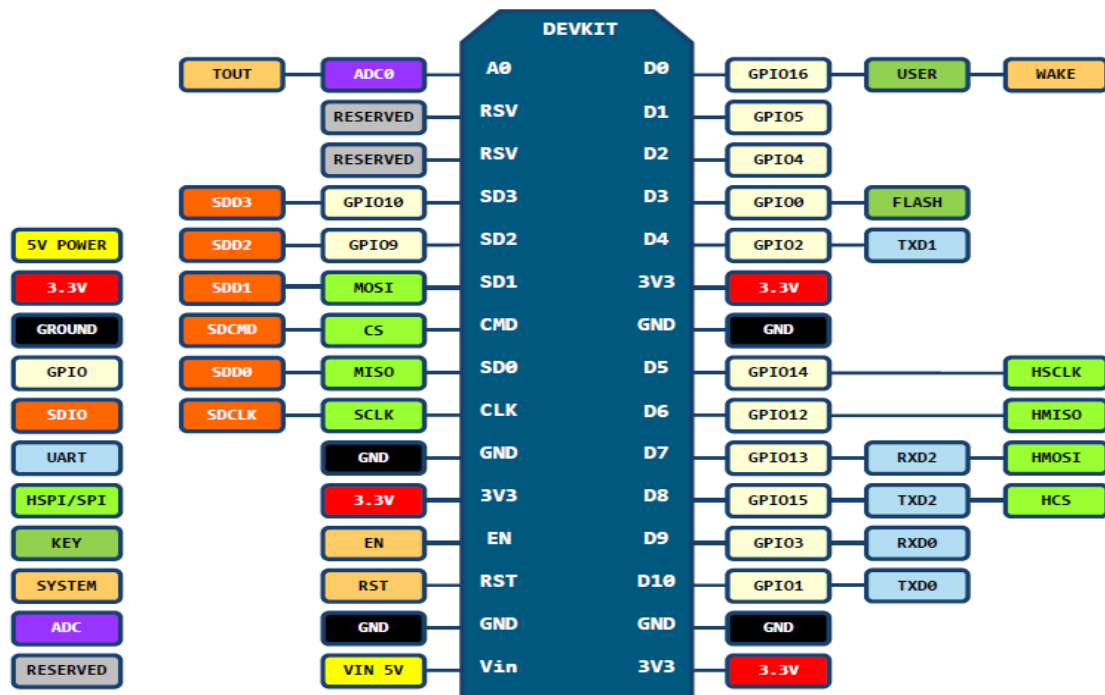
<https://openclassrooms.com/courses/creez-des-applications-pour-android>

[5]. ADAFRUIT. GATT. [en ligne] disponible sur : <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>

[6]. ADAFRUIT. ESP8266. [en ligne] disponible sur : <https://learn.adafruit.com/adafruit-feather-huzzah-esp8266>



- Input/Output de l'ESP8266 :



- Schéma électrique de l'ESP8266 :

