



**Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique**

Université de Sousse

Ecole Supérieure des Sciences et de la Technologie de Hammam Sousse

**Licence Appliquée en Sciences et Technologies de
l'Information et de la Communication**

Rapport de Stage de Fin d'Etudes

Système de pointage (RFID)

Réalisé par : Khelifi Iheb

Encadrant universitaire : Mr Hergli Mounir

Encadrant professionnel : Mr Fourti Marwan

Année Universitaire : 2014-2015

Remerciements

C'EST parce que j'ai beaucoup estimé tous ceux qui m'ont écouté, conseillé, critiqué et encadré que je tiens à leur faire part de toute ma gratitude, et plus particulièrement, je tiens à remercier à travers ces courtes lignes :

Mon encadrant Monsieur Hergli Mounir, pour m'avoir incité à mener à bien ce travail, pour son aide, son temps passé pour me guider, ses efforts pour m'intégrer dans l'environnement, son dévouement et ses précieux conseils.

Mon superviseur Monsieur Fourti Marwan, parce qu'il a accepté de me superviser et suivre les détails de l'avancement de mon travail, ainsi que son aide et ses conseils dans plusieurs étapes du projet.

Tous les membres de l'équipe du Société Techlab TN Mhira Hamza, Lhioui Hamza, Nader Jertila ainsi que les autres stagiaires pour leurs aides, leurs remarques, leur esprit ouvert et leurs respects.

Ma famille dont je ne me permettrai pas d'oublier de la remercier, pour son soutien à la fois moral et matériel durant toute ma carrière et surtout durant les moments difficiles.



Khelifi Iheb

Table de matières

Liste des tableaux	9
Introduction générale.....	10
1. Présentation générale	11
1.1 Introduction	11
1.2 Cadre du projet	11
1.2.1 Partie électronique.....	11
1.2.2 Partie informatique.....	11
1.3 Présentation de la société d'accueil	12
1.4 Cahier de charges.....	12
1.4.1 Contexte	12
1.4.2 Les objectifs	13
1.4.3 Contraintes	13
1.4.4 Structure générale du système.....	14
1.5 Conclusion.....	15
2. Etude préalable.....	16
2.1 Introduction	16
2.2 Système de pointage (pointeuse)	16
2.2.1 Qu'est ce qu'une pointeuse	16
2.3 L'RFID	17
2.3.1 Introduction à L'RFID	17
2.3.2 Principe de fonctionnement des étiquettes RFID.....	17
2.3.3 Les composants	18
2.3.4 Champs d'application.....	20

Rapport de stage de fin d'études

2.4	L'arduino	22
2.4.1	Introduction	22
2.4.2	Qu'est ce qu'arduino	23
2.4.3	La Philosophie.....	24
2.4.4	Les avantages	24
2.4.5	Description	24
2.4.6	Les différentes cartes arduino	25
2.4.7	Fonctionnement.....	28
2.4.8	L'envoi des données via Wifi.....	29
2.5	Interface logicielle de communication	31
2.5.1	Introduction	31
2.5.2	La requête http.....	31
2.5.3	Le WampServer.....	32
2.5.4	Le PHP et le HTML	33
3.	Analyse et spécification des besoins.....	35
3.1.	Introduction	35
3.2.	Analyse des besoins.....	35
3.2.1	Les besoins fonctionnels :	35
3.2.2	Les besoins non fonctionnels	36
3.3	Diagrammes de cas d'utilisation.....	36
3.3.1	Diagramme de cas d'utilisation général	36
3.3.2	Scénarios d'exécution	37
4.	Conception et Réalisation	38
4.1	Introduction	38
4.2	Partie électronique	38

Rapport de stage de fin d'études

4.2.1	Les composants utilisés	38
4.2.2	Les logiciels utilisés	50
4.2.3	Les programmes et les tests réalisés	53
4.2.4	Réalisation de la carte imprimée	67
4.3	Partie informatique	68
4.3.1	Les logiciels et les outils informatiques utilisés	68
4.3.2	Réalisation de l'interface graphique	69
4.3.3.	Conclusion.....	72
	Conclusion générale	73
	Bibliographie	74
	Webographie	75
	Résumé	76

Liste des figures

Figure 1.1 Structure générale simplifiée du système	14
Figure 1.2 Structure générale détaillée du système	14
Figure 1.1 exemple de puces RFID	20
Figure 1.2 Schéma d'une puce RFID	20
Figure 1.3 Symbole de l'arduino	22
Figure 1.4 les besoins pour commencer	23
Figure 1.5 Carte arduino Uno.....	28
Figure 1.6 Vue de face arduino Shield WiFi.....	29
Figure 1.7 Vue d'arrière arduino Shield WiFi.....	29
Figure 1.8 Carte arduino + Shield Wifi.....	30
Figure 1.9 vue de face d'Arduino Yun	30
Figure 1.10 Fonctionnement de la requête http.....	31
Figure 1.11 le serveur local (localhost)	32
Figure 1.12 l'interface pour gérer la base de données.....	33
Figure 3.1 Diagramme de cas d'utilisation générale.....	37
Figure 3.2 Diagramme de séquence	37
Figure 4.1 Les composants utilisés	38
Figure 4.2 Image réel de la carte arduino yun	39
Figure 4.3 Interfaces et boutons de la carte.....	39
Figure 4.4 Les LED de statut	43
Figure 4.5 Afficheur LCD 16*2.....	44
Figure 4.6 Les broches de l'afficheur	45
Figure 4.7 Présentation d'un afficheur LCD	46
Figure 4.8 Lecteur RFID 125 KHz.....	47

Rapport de stage de fin d'études

Figure 4.9 Présentation d'un lecteur RFID	48
Figure 4.10 Buzzer	49
Figure 4.11 LED RGB	49
Figure 4.12 Montage de la LED RGB.....	49
Figure 4.13 Logo Arduino IDE	50
Figure 4.14 L'environnement de programmation arduino.....	51
Figure 4.15 Détails de la barre de boutons	51
Figure 4.16 Logo Eagle	52
Figure 4.17 Interface de l'Eagle.....	52
Figure 4.18 Réalisation du 1er test	53
Figure 4.19 Code source du test 1	53
Figure 4.20 Logigramme de lecture des cartes	55
Figure 4.21 Solution du test 1	56
Figure 4.22 Code source du test2 et résultat	57
Figure 4.23 Logigramme de lecture des cartes avec envoi de numéro vers la base.....	60
Figure 4.24 Arduino + RFID + LCD + RGB + Buzzer	60
Figure 4.25 l'ajout de LCD + Buzzer + LED RGB	61
Figure 4.26 Code source du 3em test	62
Figure 4.27 Test final (lecture de badge)	66
Figure 4.28 Test final (envoi à la base de données)	66
Figure 4.29 la carte imprimée	67
Figure 4.30 la carte imprimée avec l'Arduino	67
Figure 4.31 Face avant de la carte	67
Figure 4.32 Logo Dreamweaver.....	68
Figure 4.33 Logo WampServer	68

Rapport de stage de fin d'études

Figure 4.34 Logo Bootstrap	69
Figure 4.35 Insertion dans la base	69
Figure 4.36 Page d'accueil.....	71
Figure 4.37 Page Attendees.....	71
Figure 4.38 Page Users.....	72

Liste des tableaux

Tableau 2.1 les cartes arduino officiels	28
Tableau 4.1 comparaison entre les deux processeurs	41

Introduction générale

L'informatique est aujourd'hui présente dans divers domaines et secteurs.

La présence des systèmes informatisés est devenue un critère de base pour assurer le bon développement d'une société ou d'une industrie.

C'est dans ce cadre, que s'inscrit mon projet de fin d'études.

Il s'agit d'un système de pointage avec des badges et des cartes à puce RFID qui utilisent la technologie NFC pour contrôler l'entrée et la sortie des employés d'une entreprise et donnera un bilan complet des entrées/sorties à la société.

La technologie de communication de données sans contact fait office de mini-révolution dans le monde des terminaux mobiles. Elle pourrait notamment faciliter les achats via téléphone portable.

Ce rapport est organisé en quatre chapitres.

Dans le premier chapitre, j'indiquerai dans un premier temps une présentation générale du projet, par la suite dans le deuxième chapitre, je donnerai une étude théorique sur la carte arduino Yun, l'afficheur LCD, le RDM6300 et la carte/badge NFC qui composent principalement la carte électronique de la pointeuse. Dans le troisième chapitre, je proposerai une spécification du système qui consiste à définir les différentes tâches à effectuer, Ensuite dans le quatrième chapitre, je présenterai une étude de conception complète de mon projet. Aussi, je décrirai enfin les différentes phases de réalisation.

Chapitre 1

1. Présentation générale

1.1 Introduction

L'objectif de ce chapitre introductif est de mettre notre travail dans son contexte général. Tout d'abord, nous commençons par faire une présentation succincte du sujet en détaillant son cadre et ses fonctionnalités. Ensuite, nous présentons l'organisme d'accueil.

Enfin, nous détaillons le cahier de charges proposé et le chronogramme suivi tout au long de la réalisation de ce projet.

1.2 Cadre du projet

La Technologie de communication de proximité (quelques centimètres) lancée par Sony et Philips, le NFC (pour Near Field Communication ou communications en champ proche), permet d'échanger des données entre un lecteur et n'importe quel terminal mobile ou entre les terminaux eux-mêmes et ce, à un débit maximum de 424 Kbits/s.

L'objectif du projet est de faire le pointage des membres de la société qui portent chacun sa propre carte/badge à travers la technologie NFC.

Notre projet se compose de deux parties :

1.2.1 Partie électronique

- Programmation carte arduino
- Affichage LCD
- Implémentation de la carte du système

1.2.2 Partie informatique

- Création d'un serveur local
- Création d'une base de données pour le stockage des données

- Création d'une interface graphique (page web) pour faciliter l'utilisation par le personnel de la société.

1.3 Présentation de la société d'accueil

Techlab TN, Implantée à Zarzis, au Avenue De Hbib Bourguiba, est active depuis 2 ans. Cette société par actions simplifiées a vu en 13 janvier 2013.

C'est une société Offshore (étrangère), Elle comporte 4 employés et un partenaire principal « Nachos.is » et d'autres...

Cet organisme est spécialisé dans le secteur d'activité de la programmation informatique et robotique. La société offre plusieurs types de produits en relations avec le développement informatique et robotique (informatique embarqué, électronique) tel que :

- Le Web : c'est sa principale activité avec une équipe qui assure la conception et le développement des sites, ainsi que le référencement et son amélioration.
- Les applications mobiles : il s'agit des applications de communication et de tous les domaines....
- L'électronique : il s'agit des robots et des systèmes électroniques...
- Commercialisation : c'est le Marketing de tous produits.

Un des produits du Techlab TN est le système de pointage à RFID.

1.4 Cahier de charges

La réalisation du système va consister à mettre en œuvre une carte arduino, un shield Wifi, un lecteur de carte RFID (RDM6300), des cartes/badges, un écran LCD, un buzzer, une LED.

1.4.1 Contexte

Une pointeuse est un système de pointage de temps destiné plus particulièrement aux personnels détachés. A l'instar d'une pointeuse, ou encore badgeuse, celle-ci se distingue le plus souvent par sa capacité à pouvoir gérer les arrivées et départs de personnel sur des lieux distants. Grace à différentes technologies, Internet, Wifi, il est possible maintenant de collecter et suivre en temps réel les activités externes des entreprises.

C'est dans ce contexte que s'intègre le sujet de mon stage de fin d'études qui consiste à mettre en place un système de pointage à distance en utilisant un shield Wifi pour l'envoi de données et la réception à partir d'un serveur local installé dans l'entreprise.

En effet, le système doit permettre à l'utilisateur le contrôle des entrées /sorties des employées en récupérant à tout instant l'état de tous les employées.

1.4.2 Les objectifs

- Présenter une large idée sur la carte arduino qui est le cœur de la carte électronique afin de bien expliquer sa construction dans les prochains chapitres.
- Présenter une large idée sur la communication Wifi entre les différents composants et en deuxième lieu à présenter l'entité intervenante dans la transmission de données.
- Créer une interface graphique qui est un site web ou page web en local crée en html et PHP à fin de piloter ou contrôler notre système.

Ce site se constitue d'une interface affichant :

- ✓ Page d'accueil
- ✓ Page « attendees » : contenant les entrées/sorties de tous les employées en temps réel.
- ✓ Page « users » : contenant les employées avec leurs numéros, ou le personnel de la société peut ajouter ou effacer un employé de la base de données.

1.4.3 Contraintes

Il existe aussi des contraintes dans notre projet qui sont les suivantes :

- Les collisions :
 - ✓ Origine : Lorsque plusieurs marqueurs se trouvent dans le champ d'un même lecteur, les communications sont brouillées par l'activité simultanée des marqueurs.
 - ✓ Détection : la détection de la collision est en fait une détection d'erreur de transmission, à l'aide d'un bit de parité ou d'un hachage (CRC..).
 - ✓ Une double détection : peut provoquer un grand problème au niveau du bilan des entrées/sorties d'employé.

1.4.4 Structure générale du système

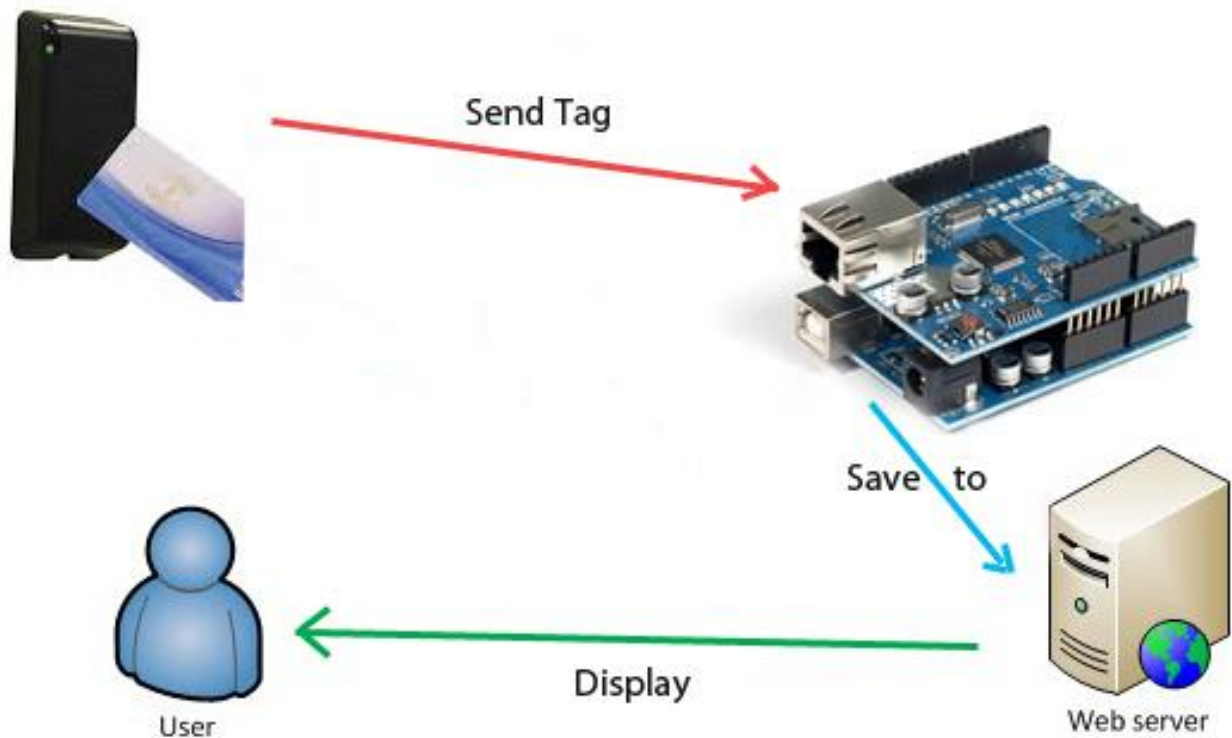


Figure 1.1 Structure générale simplifiée du système

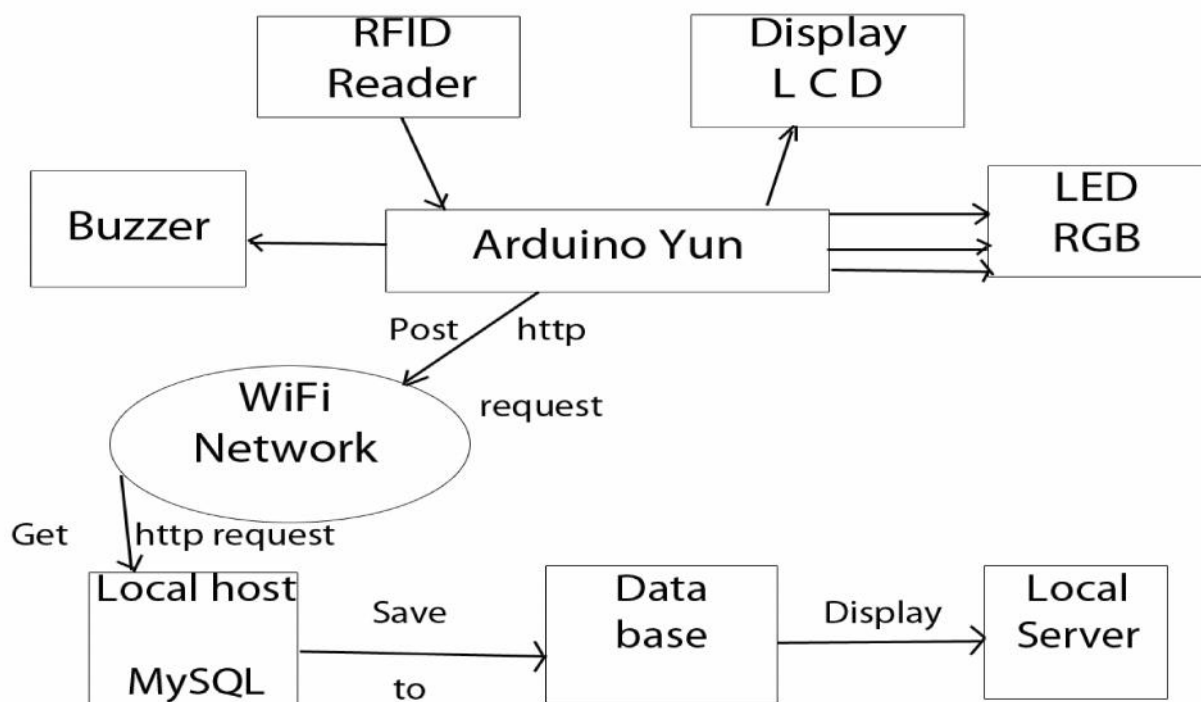


Figure 1.2 Structure générale détaillée du système

1.5 Conclusion

Ce chapitre constitue une partie introductive dans laquelle une présentation des grandes lignes du sujet a été établie. Nous avons présenté le cadre du projet, la présentation de la société d'accueil et notre cahier de charges...

Dans le chapitre suivant, nous présenterons une étude théorique du projet.

Chapitre 2

2. Etude préalable

2.1 Introduction

La réalisation de notre projet nécessite une étude approfondie sur certaines notions qui touchent non seulement le cadre général du projet, mais aussi son implémentation.

Pour bien assimiler ces différentes notions, nous détaillons, dans un premier lieu, la définition de la pointeuse ainsi que ses avantages. Ensuite, nous présenterons la notion « RFID », son fonctionnement, les composants nécessaires, le champ d'application... Ensuite, La définition de la carte « arduino » utilisée selon nos besoins ainsi qu'une présentation détaillée.

Nous terminerons ce chapitre avec une petite idée sur le coté administratif du système(le serveur local, la base de données, WAMP, HTML, PHP...).

2.2 Système de pointage (pointeuse)

2.2.1 Qu'est ce qu'une pointeuse

Une pointeuse est une machine permettant de contrôler l'heure d'entrée et l'heure de sortie d'un employé sur le lieu de travail. Cet instrument était nécessaire aux temps du capitalisme sauvage et de la construction industrielle des pays développés. Ces activités nécessitaient des équipes très nombreuses dont le travail était mal encadré et difficile à surveiller ou à surveiller.

Un système de pointage doit répondre aux impératifs résultant de la législation, de l'organisation en vigueur dans l'entreprise et du management de l'employeur.

La loi impose à l'employeur d'organiser le pointage du personnel. L'employeur doit donc choisir une méthode de pointage et le matériel adapté.

Avant de faire un bon choix, voici les trois questions à se poser :

- Quelles sont les caractéristiques légales d'un système de pointage ?
- Le type d'entreprise et son organisation impactent-ils le mode pointage ?

- Le mode de management pratiqué dans l'entreprise influe-t-il sur le type de système à retenir ?

2.3 L'RFID

2.3.1 Introduction à L'RFID

L'indentification par radiofréquence ou RFID est une méthode pour stocker et récupérer des données à distance en utilisant des marqueurs appelés Tag RFID. Les Tag RFID sont de petits objets, tels que des étiquettes autoadhésives, qui peuvent être collées ou incorporées dans des produits. Les Tag RFID comprennent une antenne associée à une puce électronique qui leur permet de recevoir et de répondre aux requêtes radio émises depuis l'émetteur-récepteur.

La technologie de marquage RFID (Radio Frequency Identification) fait aujourd'hui office de révolution dans le secteur de la logistique.

2.3.2 Principe de fonctionnement des étiquettes RFID

L'ensemble de l'étiquette est activé par un signal radio fréquence variable, émis par un lecteur composé lui-même d'une carte électronique et d'une antenne. Le lecteur peut être fixe ou mobile, et son antenne peut prendre plusieurs formes, et par exemple s'intégrer dans le cadre d'une porte, pour une application de contrôle d'accès. Le lecteur ou interrogateur transmet un signal selon une fréquence donnée vers une ou plusieurs étiquettes radio situées dans son champ de lecture.

Celles-ci transmettent un signal en retour. Lorsque les étiquettes sont "éveillées" par le lecteur, un dialogue s'établit selon un protocole de communication prédéfini, et les données sont échangées.

Les étiquettes RFID fonctionnant à basses ou moyenne fréquence utilisent un champ électromagnétique créé par l'antenne du lecteur et l'antenne de l'étiquette pour communiquer. Le champ électromagnétique alimente l'étiquette et active la puce. Cette dernière va exécuter les programmes pour lesquels elle a été conçue. Pour transmettre les informations qu'elle contient, elle va créer une modulation d'amplitude ou de phase sur la fréquence porteuse. Le lecteur reçoit ces informations et les transforme en code binaire. Dans le sens lecteur vers étiquette, l'opération est symétrique, le lecteur émet des informations par modulation sur la porteuse. Les modulations sont analysées par la puce et numérisées.

Une des particularités de ce principe est que plus la fréquence porteuse est basse plus le nombre de tours de l'antenne de la puce doit être important pour créer un voltage suffisant pour alimenter la puce. L'étiquette peut être apposée, portée, insérée dans un objet (colis, carte, véhicule...).

2.3.3 Les composants

- Les lecteurs :

Ce sont des dispositifs actifs, émetteurs de radiofréquences qui vont activer les marqueurs qui passent devant eux en leur fournissant à courte distance l'énergie dont ceux-ci ont besoin. La fréquence utilisée est, selon les générations :

- 125 kilohertz.
- 13,56 mégahertz (ISO 14443, ISO 15693 et ISO 18000-3).
- UHF (EPCglobal et ISO 18000-6c).
- les fréquences et les puissances d'émission dépendent des législations en vigueur (915 MHz (Amérique), 869 MHz (Europe)).
- 2.45 GHz.

Une fréquence plus élevée présente l'avantage de permettre un échange d'information (lecteur \Leftrightarrow marqueur) à des débits plus importants qu'en basse fréquence. Les débits importants permettent l'implémentation de nouvelles fonctionnalités au sein des marqueurs (cryptographie, mémoire plus importante, anticollision).

- L'anticollision :

C'est la possibilité pour un lecteur de pouvoir dialoguer avec un marqueur lorsque plus d'un marqueur se trouvent dans son champ de détection. Les algorithmes d'anticollision sont décrits par les normes (ISO 14443 et ISO 15693).

On évite bien entendu toute fréquence qui serait en résonance avec celle des molécules d'eau contenues entre autres dans le corps humain pour des raisons de sécurité.

- Les marqueurs :

Ce sont des dispositifs passifs, ne nécessitant aucune source d'énergie en dehors de celle fournie par les lecteurs au moment de leur interrogation.

Outre de l'énergie pour le marqueur, le lecteur envoie un signal d'interrogation particulier auquel répond le marqueur. L'une des réponses les plus simples possibles est le renvoi d'une identification numérique, par exemple celle du standard EPC-96 qui utilise 96 bits. Une table ou une base de données peut alors être consultée pour assurer un contrôle d'accès, un comptage, ou un suivi donné sur une chaîne de montage, ainsi que toute statistique souhaitable.

Le marqueur est extrêmement discret par sa finesse (parfois celle d'un morceau de rhodoïd), sa taille réduite (quelques millimètres), et son poids négligeable. Son coût étant devenu minime, on peut envisager de le rendre jetable, bien que la réutilisation soit plus « Écologiquement correcte ».

Il se compose :

- D'une antenne
- D'une puce de silicium
- D'un substrat et/ou d'une encapsulation

Notons aussi l'existence des tags "actifs" et "semi-actifs" (aussi appelés BAP, Battery Powered Passive tags).

Les tags actifs sont équipés d'une batterie leur permettant d'émettre un signal. De ce fait, ils peuvent être lus depuis de longues distances, comparés aux tags passifs. Mais, une émission active d'informations signale à tous la présence des tags et pose des questions quant à la sécurité des marchandises.

Les tags semi-actifs n'utilisent pas leur batterie pour émettre des signaux. Ils agissent comme des tags passifs au niveau "communication". Mais leur batterie leur permet, par exemple, d'enregistrer des données lors du transport. Ces tags sont utilisés dans les envois de produits sous température dirigée et enregistrent la température de la marchandise à intervalle régulier.

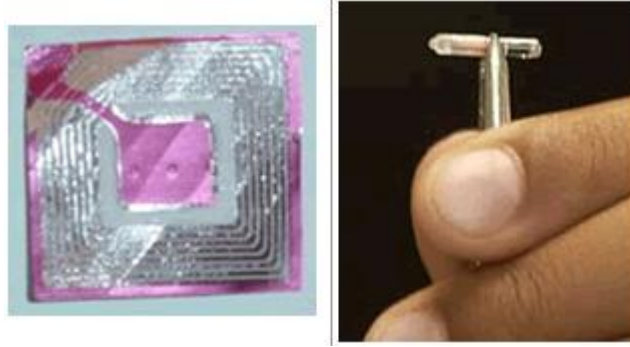


Figure 2.1 exemple de puces RFID

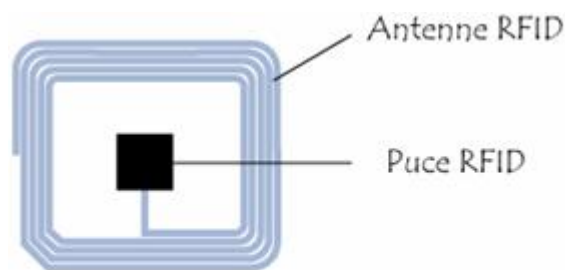


Figure 2.2 Schéma d'une puce RFID

2.3.4 Champs d'application

Le coût relativement bas d'une puce RFID fait de cette technologie un candidat de choix pour le marquage invisible des produits de consommation. Plusieurs applications sont largement envisagées:

- **Le remplacement du code-barres:** l'apposition d'une étiquette (Electronic Product Code) pouvant être lue automatiquement et instantanément améliore considérablement les transactions tout au long de la chaîne de production et de consommation. Les transactions deviennent encore plus efficaces et rapides grâce à l'allègement considérable de l'encodage. Ceci offre un certain nombre d'avantages pour les producteurs ou distributeurs de produits mais aussi pour les consommateurs. En effet, le consommateur pourra obtenir le montant total de ses achats sans devoir vider le contenu de sa charrette ni attendre son tour aux caisses.

- **La protection des produits contre le vol:** ce marqueur invisible et fréquemment apposé dès la production du produit est une arme quasi infaillible permettant au magasin de lutter contre le vol.
- **La sécurité des individus:** pour que les enfants ne se perdent pas dans les parcs d'attractions, une société (Safezone) a conçu un système permettant la localisation permanente de ses visiteurs. Chaque personne porte un bracelet, semblable à une montre, intégrant une puce RFID. Le dispositif indique la position de chaque individu aux lecteurs radio fréquence disséminés dans le parc.
- **Le contrôle de la chaîne du froid:** une déclinaison de la puce RFID, dite active, peut être utilisée pour capturer de manière régulière certaines informations provenant de son environnement. Pour les denrées périssables, la capacité de mémorisation de la puce et son aptitude à communiquer cette information permettent de s'assurer, par exemple, qu'un produit qualifié de "frais" a bien été soumis aux conditions de température prescrites par les normes d'hygiène, et ce depuis la production jusqu'à la vente.
- **La location et le prêt de matériel:** l'apposition d'une balise RFID permet de s'assurer que le matériel restitué est bien le matériel loué.
- **L'élevage et le suivi vétérinaire des animaux:** les animaux d'élevage peuvent être munis d'une étiquette RFID à des fins de marquage.
- **Le transport et les péages routiers:** les tickets d'autoroute, de tram, de métro, de parking, etc. appartiennent à un secteur où l'automatisation des transactions aux points de contrôle est un atout.
- **l'écologie:** afin de suivre l'évolution des arbres (âge, maladie, etc.) et les interventions à pratiquer, certaines communes ont choisi la technique du RFID. Une puce est implantée dans le tronc des arbres et les agents sont équipés d'un simple lecteur portatif.
- **La domotique** est souvent définie comme l'ensemble des technologies de l'électronique, de l'informatique et des télécommunications utilisées dans les habitations privées. Elle vise à assurer des fonctions de sécurité, de confort, de gestion d'énergie et de communication qu'on peut retrouver dans la maison. La technologie RFID est considérée comme un nouveau type de capteur d'informations environnementales pour la domotique. En effet, par le biais des informations que la

puce met à la disposition du système de domotique, la maison peut reconnaître ses habitants et diffuser ainsi leur ambiance musicale préférée ou encore adapter la luminosité de la pièce selon les préférences de chacun.

- **L'industrie du stockage et du transport de produits:** par le biais de la technologie RFID, la gestion de stocks est fortement automatisée et accélérée. Ainsi, un système informatique pourrait, par exemple, organiser automatiquement le réapprovisionnement en produits des rayons du supermarché. Le suivi en temps réel des produits pourrait même être réalisé.
- **Les secteurs médical et pharmaceutique** sont aussi très intéressés par la technologie d'étiquetage électronique. Par le biais du RFID, chaque médicament pourrait être identifié et le suivi médical des patients pourrait en être amélioré.
- **L'implantation de puces RFID dans le corps humain** est même envisagée par l'organisme Américain Food & Drug Administration. La puce RFID implantée, de la taille d'un grain de riz, contiendrait des informations personnelles (identité, informations médicales).
- **le secteur militaire** est un consommateur friand de technologies permettant l'identification et le suivi.

2.4 L'arduino

2.4.1 Introduction



Figure 2.3 Symbole de l'arduino

Notre système de pointage est muni d'une carte arduino.

2.4.2 Qu'est ce qu'arduino

L'Arduino est un circuit imprimé en matériel libre (les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte, comme le microcontrôleur par exemple, ne sont pas en licence libre) sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, etc....

Arduino peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien peut être connecté à un ordinateur pour communiquer avec ses logiciels (ex. : Macromedia Flash, Processing, Max/MSP, Usine Hollyhock, Pure Data, SuperCollider). En 2011, les versions vendues sont pré assemblées. Des informations sont fournies pour ceux qui souhaitent assembler l'Arduino eux-mêmes.

Le projet Arduino a reçu un titre honorifique à l'Ars Electronica 2006, dans la catégorie Digital Communities.

L'arduino en quelques mots :

Arduino = 1 carte à microcontrôleur + 1 outil de développement + 1 communauté active.

Le logiciel et le matériel sont open-source.



1 carte à micro-contrôleur



1 communauté active



1 outil de développement

Figure 2.4 les besoins pour commencer

2.4.3 La Philosophie

L'idée est d'utiliser la carte Arduino comme un macro-composant dans des applications de prototypage électrique. Le concepteur n'a plus qu'à développer des interfaces et programmer le macro-composant pour réaliser son application.

2.4.4 Les avantages

- Pas cher.
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso, etc....)
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires qui se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique, interface Ethernet, GPS, etc....

Par sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle.

2.4.5 Description

2.4.5.1 Matériel

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328 ou ATmega2560 pour les versions récentes, ATmega168 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).

Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

Les modules sont programmés au travers d'une connexion série RS-232, mais les connexions permettant cette programmation diffèrent selon les modèles. Les premiers Arduino

possédaient un port série, puis l'USB est apparu sur les modèles Diecimila, tandis que certains modules destinés à une utilisation portable se sont affranchis de l'interface de programmation, relocalisée sur un module USB-série dédié (sous forme de carte ou de câble).

L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits. Le modèle Diecimila par exemple, possède 14 entrées/sorties numériques, dont 6 peuvent produire des signaux PWM, et 6 entrées analogiques. Les connexions sont établies au travers de connecteurs femelle HE14 situés sur le dessus de la carte, les modules d'extension venant s'empiler sur l'Arduino. Plusieurs sortes d'extensions sont disponibles dans le commerce.

Certains modules non officiels, compatibles avec la technologie Arduino, utilisent des connecteurs mâles pour une utilisation aisée avec des plaques de test.

2.4.5.2 Logiciel

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande.

Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.

2.4.6 Les différentes cartes arduino

Les modules d'origine des différentes versions de l'Arduino sont fabriqués par la société italienne Smart Projects. Quelques-unes des cartes de marque Arduino ont été conçues par la société américaine SparkFun Electronics.

Dix-sept versions des cartes de type Arduino ont été produites et vendues dans le commerce à ce jour dans l'ordre chronologique ci-dessous :

- Serial Arduino programmé avec une connexion série par connecteur DB9 et utilisant un ATmega8.
- Arduino Extreme, programmable via une connexion USB et utilisant un ATmega8.

- Arduino Mini, une version miniature de l'Arduino utilisant un ATmega168 de type CMS.
- Arduino Nano, une version encore plus petite de l'Arduino alimenté par USB et utilisant un ATmega168 (avant la version 3) ou ATmega328 (à partir de la version 3.0) de type CMS.
- LilyPad Arduino, une conception de type minimaliste pour permettre une application portable utilisant un ATmega168 de type CMS.
- Arduino NG, programmable via une connexion USB et utilisant un ATmega8.
- Arduino NG plus, programmable via une connexion USB et utilisant un ATmega168.
- Arduino Bluetooth(BT), programmable via une connexion Bluetooth et utilisant un ATmega 328.
- Arduino Diecimila, possède une interface USB et utilise un ATmega168 dans un boîtier format DIL28. (16 ko flash, 1 ko SRAM, 0,5 ko EEPROM).
- Arduino Duemilanove (2009) utilise un Atmega168 et est alimenté en électricité par le connecteur USB ou une alimentation externe avec commutation automatique. La nouvelle version est équipée d'un ATmega328 (32 ko de flash, 2 ko de SRAM, et 1 ko d'EEPROM).
- Arduino Mega est équipé d'un ATmega1280 de type CMS pour avoir des Entrées/Sorties supplémentaires et de la mémoire (128 ko flash, 8 ko SRAM, 4 ko EEPROM).
- Arduino Uno utilise un ATmega328 comme les derniers modèles de Duemilanove, mais alors que le Duemilanove utilisait une puce FTDI pour la programmation via un connecteur USB, l'Uno utilise une puce ATmega8U2 programmé comme un convertisseur série.
- Arduino Mega2560 est équipé d'un ATmega2560 de type CMS, augmentant la mémoire totale disponible à 256 ko. Il est équipé aussi de la nouvelle puce USB ATmega8U2.
- Arduino Ethernet est une carte Arduino UNO intégrant un chip Wiznet W5100 pour rajouter la connectivité Ethernet intégré.
- Arduino Leonardo est une version basse coût de l'Arduino UNO à base d'un ATmega32U4.

Rapport de stage de fin d'études

- Arduino DUE est une évolution de l'Arduino Mega2560 avec un microcontrôleur 32 bits Atmel SAM3X (ARM 32 bits Cortex-M3).
- Arduino Esplora est une carte dérivée de l'Arduino Leonardo. Elle a la forme d'une manette de jeu. Contrairement aux autres Arduino, c'est une carte « tout-en-un » qui comporte de nombreux capteurs (température, accélération, lumière, microphone, potentiomètre...) ainsi que 4 boutons poussoirs, un potentiomètre et un joystick analogue.

Arduino	Micro Contrôleur	Flac h Ko	EEPRO M Ko	SRA M Ko	Broches d'E/S numériqu es	Broch es d'ES avec PWM	Broches d'entrées analogiq ues	Type d'interface USB	Dimension s mm
Diecimila	ATmega168	16	0.5	1	14	6	6	FTDI	2.7*2.1
Duemilan ove	ATmega168/3 28P	16/3 2	0.5/1	½	14	6	6	FTDI	2.7*2.1
Uno	ATmega328P	32	1	2	14	6	6	ATmega16U2	2.7*2.1
Leonardo	ATmega32U4	32	1	2.5	20	7	12	ATmega32U4	2.7*2.1
Mega	ATmega1280	128	4	8	54	15	16	FTDI	4*2.1
Mega2560	ATmega2560	256	4	8	54	15	16	ATmega8U2	4*2.1
Due	ATmega SAM3X8E	512	0	96	54	12	12	SAM3X8E(usb host),ATmega1 6U2	4*2.1
Fio	ATmega328P	32	1	2	14	6	8	Aucune	1.6*1.1
Nano	ATmega168 or ATmega328	16/3 2	0.5/1	½	14	6	8	FTDI	1.70*0.73
LilyPad	ATmega168V or ATmega328V	16	0.5	1	14	6	6	Aucune	2*2

Rapport de stage de fin d'études

Yun	ATmega32U4	32	1	2.5	20	7	12		73mm*53 mm
esplora	ATmega32U4	32	1	2.5	N/A	N/A	N/A	ATmega32U4	6.5*2.4

Tableau 2.1 les cartes arduino officiels

2.4.7 Fonctionnement

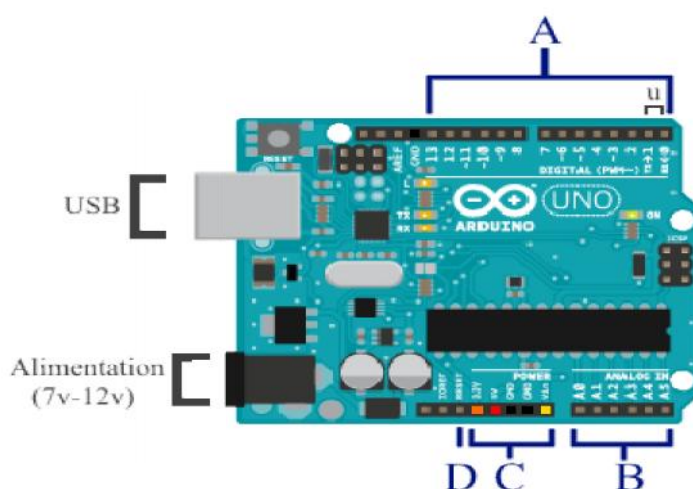


Figure 2.5 Carte arduino Uno

Les différentes versions des Arduino fonctionnent sous le même principe général:

- A: ce sont les pattes dites digitales (0,1) ou "tout ou rien" ; elles offrent en sortie du 5V et acceptent en entrée du 5V sur le même principe.
 - fonction `digitalWrite ()` et `digitalRead ()`.
- B: ce sont les pattes dites analogiques, valeur entre 0V et 5V
 - Fonction `analogWrite ()` et `analogRead ()`.
- C: les différentes pattes d'alimentation:
 - Rouge: sortie 5v (+)
 - Orange: sortie 3,3V (+)
 - Noire: les masses (-)
 - Jaune: entrée reliée à l'alimentation (7V-12V)

Il y a des variations entre les différentes cartes (ex: UNO, la patte 13 est équipée d'une résistance).

2.4.8 L'envoi des données via Wifi

Après, avoir détecté la carte/badge RFID par le module RFID utilisé, il nous reste que l'envoi de rfidTag défini par chaque carte ou badge. Il existe plusieurs méthodes d'envoi de données pour l'arduino et une grande diversité de matériel.

Les méthodes qui peuvent être utilisées dans notre projet :

- Interface câble (antenne).
- Bluetooth.
- Cellulaire (avec carte SIM).
- Wifi.
- Ethernet.

Après une petite étude sur ses connectivités, les matériels, les besoins, nous avons choisi le Wifi en premier lieu. En plus, il existe plusieurs matériels :

- ✓ Shield wifi.
- ✓ Module wifi.

Le shield wifi est une solution efficace pour le connecté à notre carte arduino, qui dispose une diversité de fonctions. Ce bouclier connecte la carte arduino à internet sans fil. Nous avons besoin de le connecter à notre réseau sans fil.

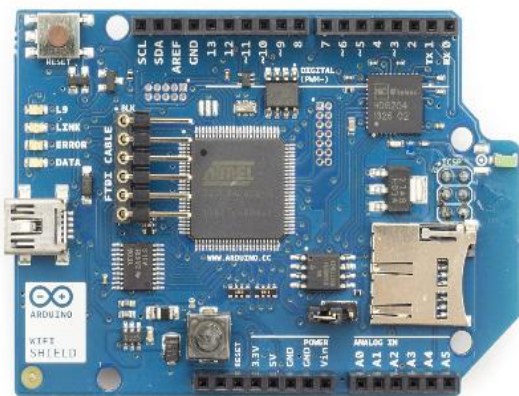


Figure 2.6 Vue de face arduino Shield WiFi

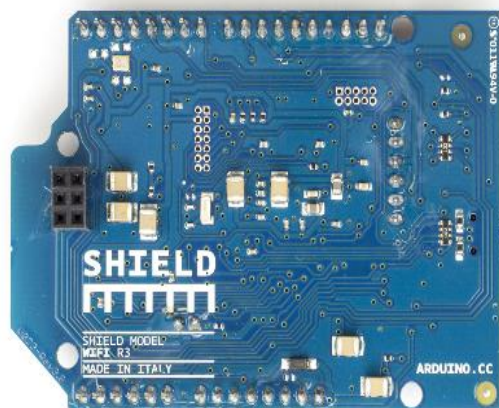


Figure 2.7 Vue d'arrière arduino Shield WiFi

Alors la structure de notre système devienne comme ci-dessous :

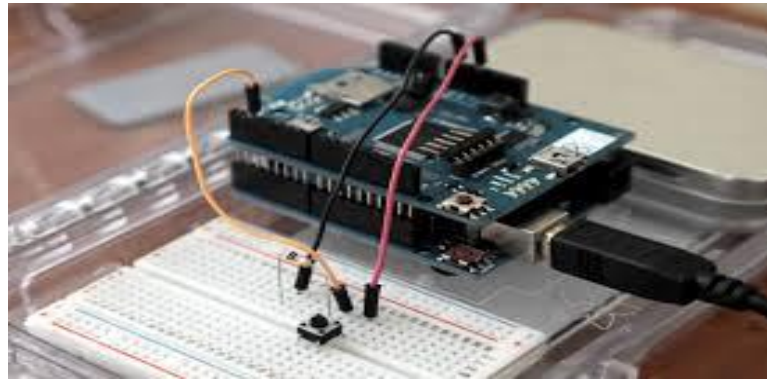


Figure 2.8 Carte arduino + Shield Wifi

Pour mieux optimiser de l'espace, j'ai cherché une autre solution. Il existe une carte arduino Yun qui dispose déjà d'une interface wifi.

L'Arduino Yun est un microcontrôleur basé sur une ATmega32U4 et un Atheros AR9331. Le processeur Atheros supporte une distribution Linux basée sur OpenWRT et se nommant Linino. La carte dispose d'un support Ethernet et Wifi intégré, un port a USB-A, un emplacement pour carte micro-SD, 20 broches d'entrées/sorties digitales (dont 7 peuvent être utilisées en sortie PWM et 12 comme entrée analogique), une horloge 16 MHz, une connexion micro USB, un connecteur ICSP et 3 boutons resets.

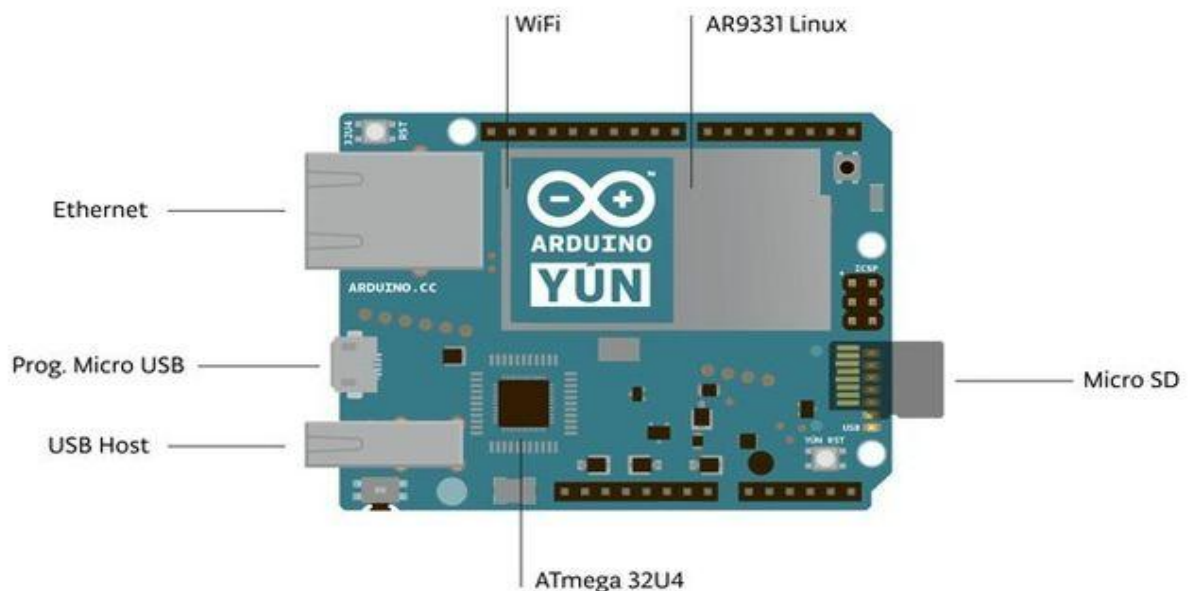


Figure 2.9 vue de face d'Arduino Yun

2.5 Interface logicielle de communication

2.5.1 Introduction

L'objectif du projet est de lire le rfidTag des cartes/badges et d'envoyer les valeurs depuis notre carte arduino à l'internet ou au routeur, pour être stockés dans un serveur web local et puis l'afficher sur une page web.

L'idée est d'envoyer une requête « http » au routeur vers le serveur local qui a une adresse IP fixe. Avec le wampserver nous allons créer un serveur local et une base de données en local. Puis à travers le PHP et html nous allons créer une page web pour récupérer nos valeurs et les arranger dans un tableau.

2.5.2 La requête http

« Http » : HyperText Transfer Protocol.

C'est le protocole de transfert sur internet le plus courant. C'est par l'intermédiaire de celui-ci que sont transmises les pages Web (au langage HTML) et que le navigateur Web se présente de façon structurée.

Le HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web. Il est utilisé pour échanger toute sorte de données entre client HTTP et serveur http.

Dans le schéma suivant, les flèches représentent les requêtes HTTP :

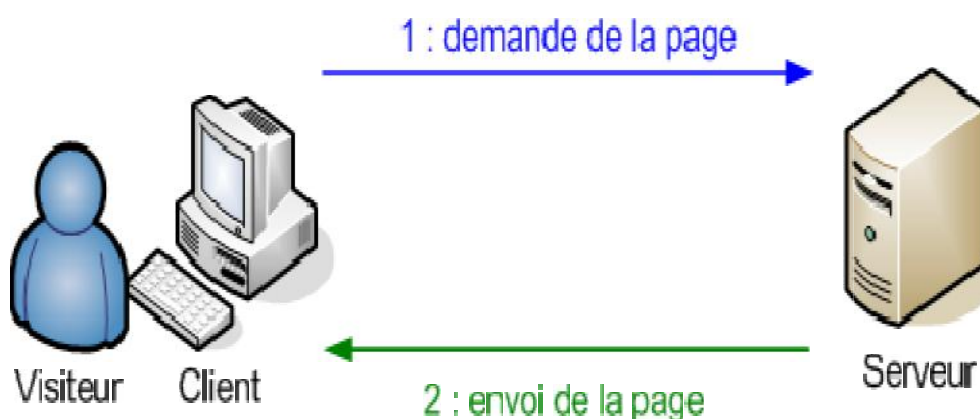


Figure 2.10 Fonctionnement de la requête http

Dans notre cas, la carte arduino fonctionne ou joue le rôle d'un client.

2.5.3 Le WampServer

WampServer (anciennement WAMP 5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'une icône (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

Dés que nous avons terminé l'installation et les réglages de wampserver, nous obtenons un serveur local qui fonctionne. Alors, l'étape suivante est de créer une base de données locale ou on va stocker les valeurs envoyées de la part de la carte arduino via Wifi.

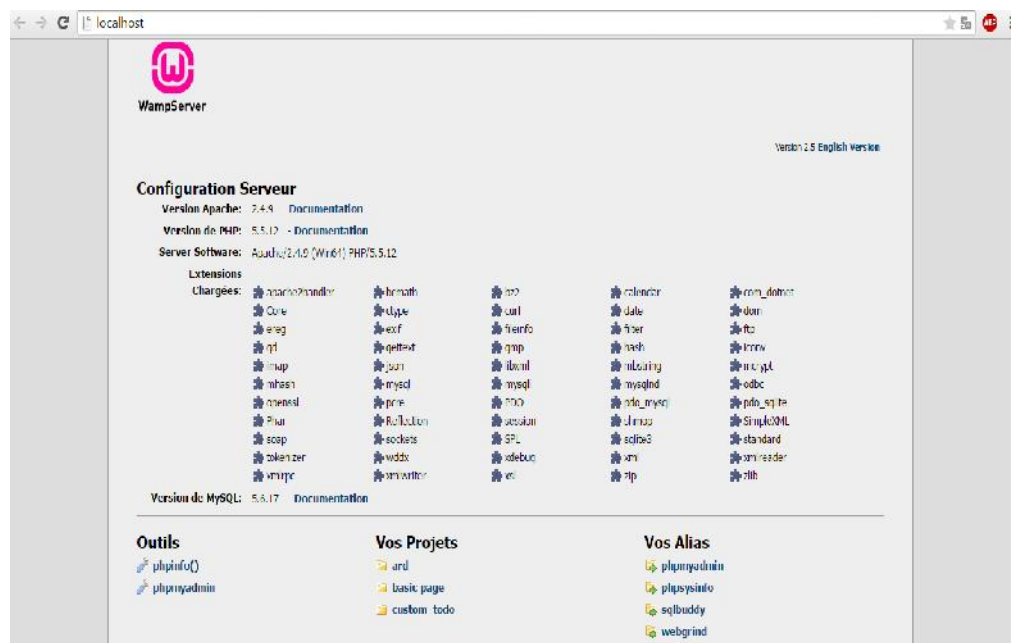


Figure 2.11 le serveur local (localhost)

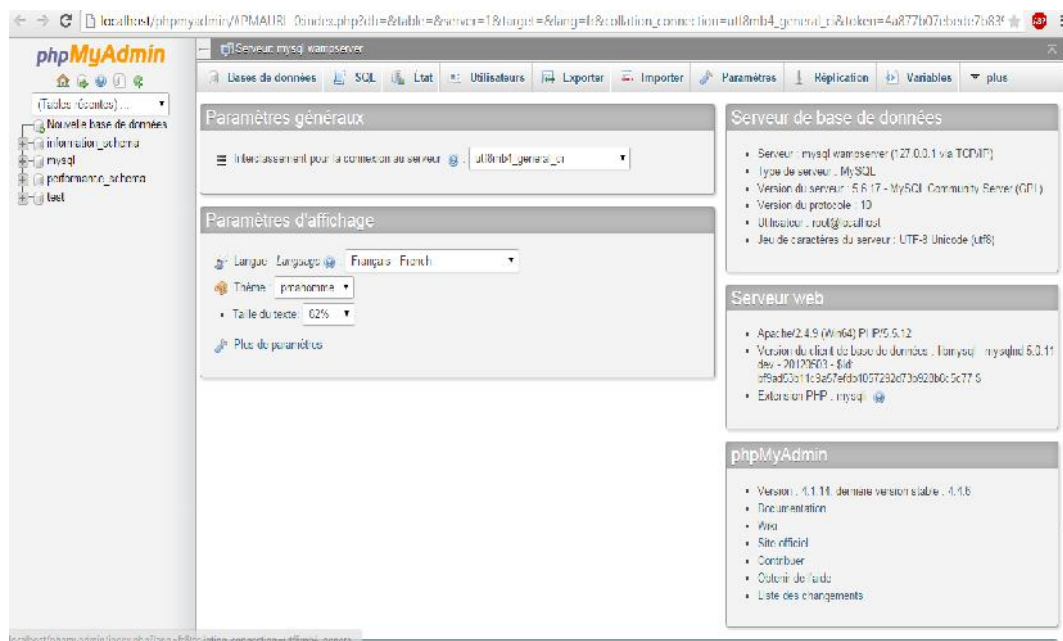


Figure 2.12 l'interface pour gérer la base de données

2.5.4 Le PHP et le HTML

« PHP » : Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récuratif), est un langage de programmation libre principalement, utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet comme C++.

Une fois que nous sommes connectés à MySQL et à une base de données, nous pouvons effectuer des tâches classiques dans un SGBD : afficher une table, supprimer un enregistrement, mettre à jour un enregistrement, afficher le résultat d'une requête... Il est donc indispensable d'avoir au préalable une bonne connaissance du langage SQL. PHP est une bonne solution qui répond à notre besoin.

« HTML » : Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière

conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du Standard Generalized Markup Language (SGML).

Nous allons créer une interface graphique à base de PHP et HTML, qui est une page web sert au personnel de la société de contrôler les entrées/sorties des employés en temps réel.

Chapitre 3

3. Analyse et spécification des besoins

3.1. Introduction

Nous abordons dans ce chapitre la phase d'analyse et spécification. Cette étape est conçue pour la détermination des différentes fonctionnalités attendues de l'application. Ainsi En premier lieu nous présenterons les besoins fonctionnels et non fonctionnels du projet. En deuxième lieux nous expliciterons ces besoins à l'aide des diagrammes de cas d'utilisation et des scénarios d'exécution.

3.2. Analyse des besoins

Notre projet est destiné aux employés de la société et donc les deux acteurs du système sont l'employé, qui doit passer son badge pour marquer l'heure d'entrée et de la sortie et d'autre part le personnel de la société, qui doit contrôler ces données enregistrées.

3.2.1 Les besoins fonctionnels :

Pour assurer le bon déroulement de pointage, notre système doit offrir :

- ✓ aux employés la possibilité de :
 - Pointage : le système doit offrir aux employés des badges ou des cartes qui identifient chaque employé par son numéro (id). l'employé doit passer le badge/carte devant le lecteur des cartes de notre système pour s'identifier.
- ✓ Au personnel de la société la possibilité de :
 - S'identifier : le système doit offrir une interface d'authentification. Le personnel doit utiliser son numéro et son mot de passe crée auparavant dans la basse de données.
 - Contrôler : l'interface fournie par le système doit permettre de contrôler en temps réel les entrées/sorties des employés.
 - Ajouter : l'interface fournie par le système doit offrir au personnel d'ajouter des employés lors d'un nouveau recrutement.

- Supprimer : l'interface fournie par le système doit offrir au personnel de supprimer des employés s'il est besoin.

3.2.2 Les besoins non fonctionnels

Plusieurs besoins opérationnels sont à tenir en compte afin que notre système soit plus convivial, plus simple à utiliser et plus performant. Les besoins non fonctionnels sont cités ci-dessous :

- L'ergonomie : l'utilisation du système et de l'interface doivent être simple et convivial. Le système doit fournir un temps de réponse acceptable.
- L'évolutivité : le système doit être ouvert à des futures perspectives et répond aux changements de besoins du marché.
- Maintenabilité : le système doit être facile à maintenir.

3.3 Diagrammes de cas d'utilisation

Pour représenter le modèle conceptuel de notre système nous avons utilisé les diagrammes de cas d'utilisation « use-case » en suivant les formalismes définies par UML. En effet ces diagrammes donnent une vision globale du comportement fonctionnel de notre système. Nous définissons plusieurs niveaux d'abstraction de ces diagrammes, du plus générale vers les use-case détaillés.

3.3.1 Diagramme de cas d'utilisation général

Dans le diagramme du cas d'utilisation général, représenté sur la figure nous regroupons tous les cas d'utilisation de base afin d'avoir une vue globale du fonctionnement de notre système et afin de mettre en évidence les éventuelles relations qui peuvent les lier.

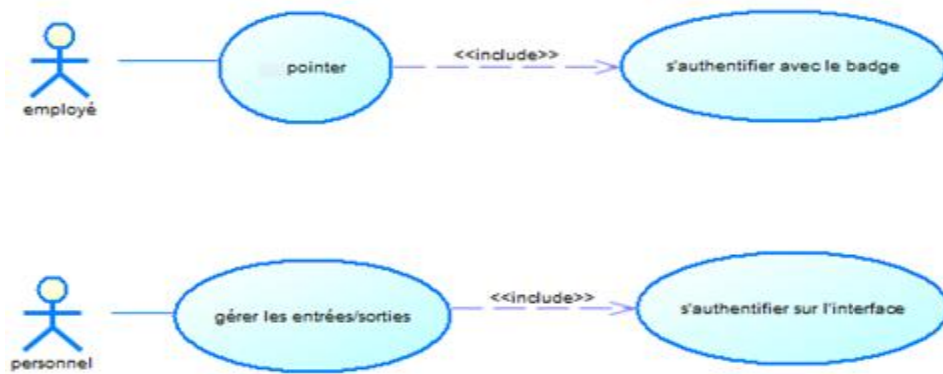


Figure 3.1 Diagramme de cas d'utilisation générale

3.3.2 Scénarios d'exécution

Pour présenter et donner une idée sur les scénarios de notre système nous avons utilisé le diagramme de séquence.

Le diagramme de séquence est une représentation graphique de la chronologie des échanges de messages avec le système ou au sein du système qui permet de décrire les interactions entre les objets pour chaque cas d'utilisation.

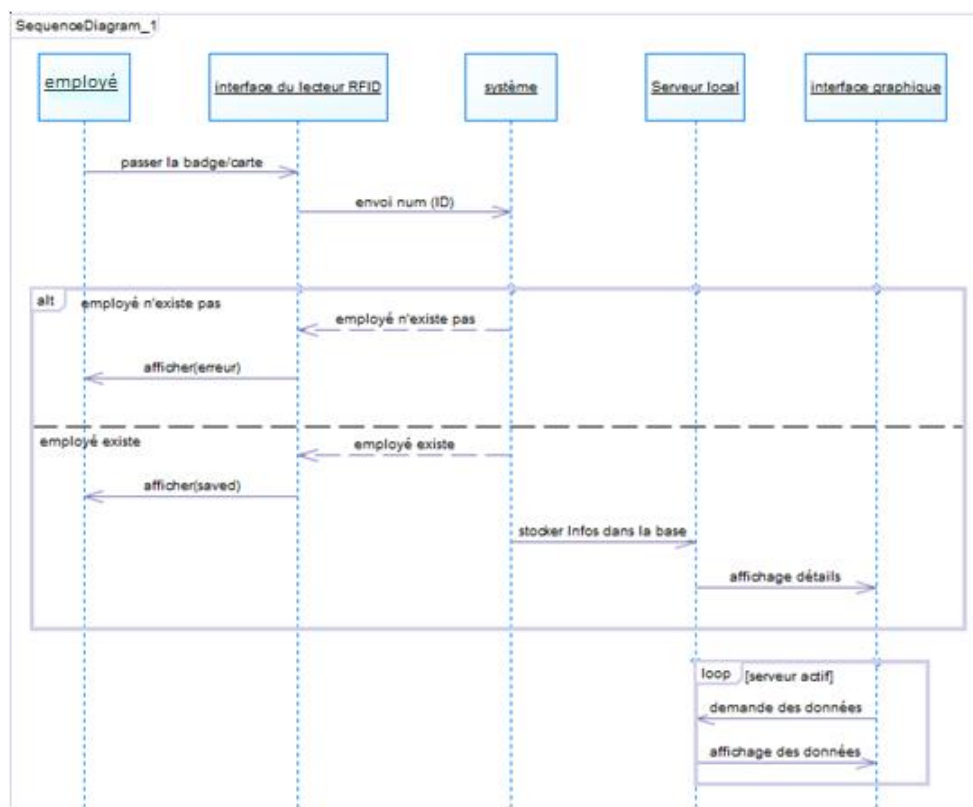


Figure 3.2 Diagramme de séquence

Chapitre 4

4. Conception et Réalisation

4.1 Introduction

Après une analyse et spécification des besoins nous passons maintenant à la conception et réalisation de notre projet. Dans cette partie nous toucherons aux différents composants, logiciels et outils utilisés pour la création de notre carte.

4.2 Partie électronique

4.2.1 Les composants utilisés

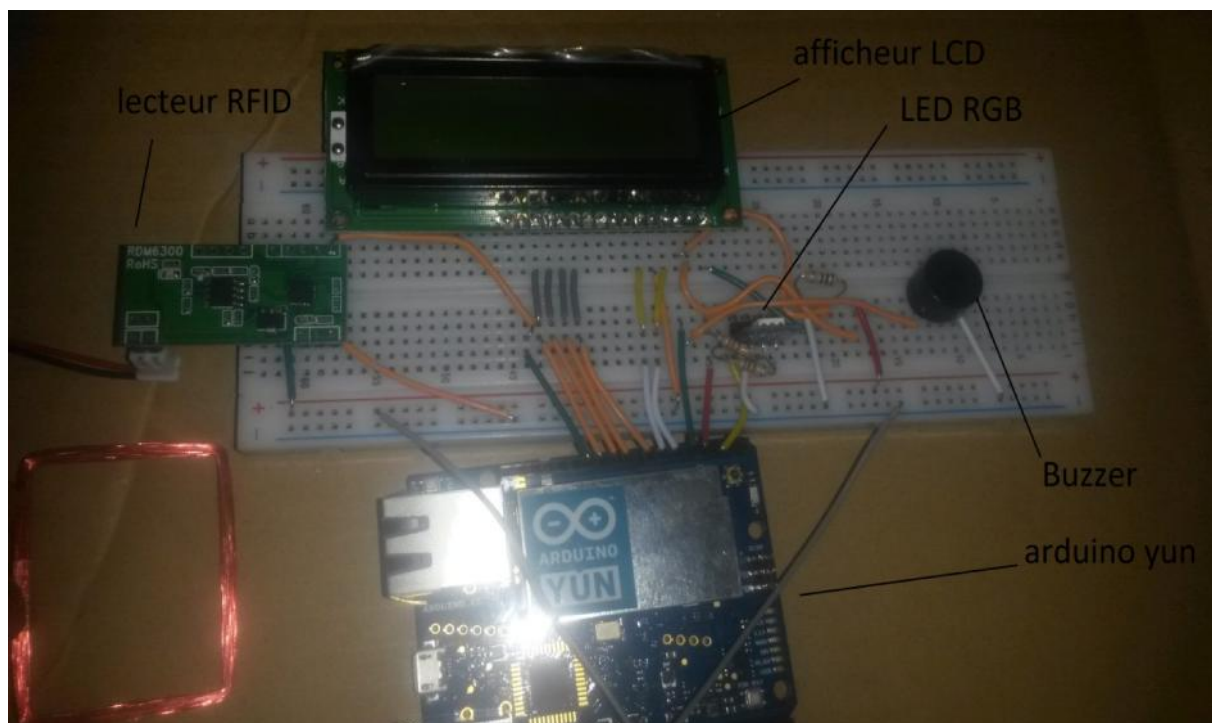


Figure 4.1 Les composants utilisés

4.2.1.1 La carte arduino yun



Figure 4.2 Image réel de la carte arduino yun

- Définition et fonctionnalité :

Arduino Yun est un microcontrôleur basé sur une ATmega32U4 et un Atheros AR9331. Le processeur Atheros supporte une distribution Linux basée sur OpenWRT et se nommant Linino. La carte dispose d'un support Ethernet et Wifi intégré, un port a USB-A, un emplacement pour carte micro-SD, 20 broches d'entrées/sorties digitales (dont 7 peuvent être utilisées en sortie PWM et 12 comme entrée analogique), une horloge 16 MHz, une connexion micro USB, un connecteur ICSP et 3 boutons resets.

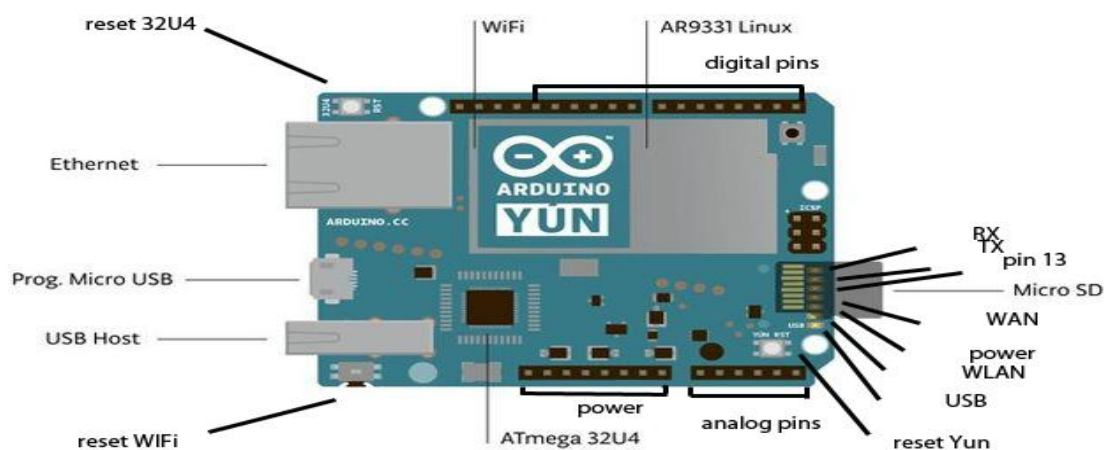


Figure 4.3 Interfaces et boutons de la carte

Rapport de stage de fin d'études

Yun se différencie des autres cartes Arduino par le fait qu'il communique avec la distribution Linux de la carte, offrant ainsi la puissance d'un ordinateur connecté au réseau avec la facilité d'utilisation d'un Arduino. En plus des commandes Linux telles que cURL, vous pouvez écrire vos propres scripts Shell et Python pour réaliser des interactions robustes.

- Détails techniques :

Arduino	Machine Linux embarquée
<ul style="list-style-type: none">- Microcontrôleur: ATmega32u4- Tension de fonctionnement: 5V- Tension d'entrée (recommandée): 5V via micro USB ou PoE 802.3af- Tension d'entrée (limites): 6-20V- Entrée/sortie digital: 14- PWM: 7 canaux- Entrées analogiques: 6 (plus 6 multiplexées sur 6 broches digitales)- Courant continu par entrée/sortie: 40 mA- Courant continu pour broche 3.3V: 50 mA- Mémoire Flash: 32 KB (ATmega32u4)	<ul style="list-style-type: none">- Processeur MIPS 24K fonctionnant jusqu'à 400 MHz- RAM: 64MB DDR2 (et Flash de 16MB SPI)- IEEE 802.11bgn complet 1x1 AP ou routeur- WIFI- Connexion Ethernet- USB: USB 2.0 host/device- Compatible Power Over Ethernet (PoE) 802.3af- Support carte Micro SD

dont 4 KB utilisé par le bootloader

- **SRAM: 2.5 KB (ATmega32u4)**
- **EEPROM: 1 KB (ATmega32u4)**
- **Vitesse d'horloge: 16 MHz**

Tableau 4.1 comparaison entre les deux processeurs

- Alimentation :

Il est recommandé d'alimenter la carte par l'intermédiaire de la connexion micro-USB avec 5VDC. Si on alimente la carte par l'intermédiaire de la broche Vin, nous devons fournir une tension régulée à 5VDC. Le Yun est également compatible avec les modules PoE (Power Over Ethernet) mais nous aurons besoin d'ajouter un module PoE pour bénéficier de cette fonctionnalité.

Les broches d'alimentation sont les suivantes :

- ✓ VIN: Tension d'entrée de la carte Arduino. A l'opposé des autres cartes Arduino, si vous voulez alimenter votre carte par cette broche il faut absolument fournir une tension régulée à 5 volts.
- ✓ 5V: La tension d'alimentation utilisée pour alimenter le microcontrôleur et les autres composants de la carte. Peut provenir soit de VIN ou du port USB (d'alimentation).
- ✓ 3V3: Une alimentation 3.3 volts générée par le régulateur disponible sur la carte. Courant maximum 50 mA.
- ✓ GND: Broches de masse/GND/Ground.
- ✓ IOREF: La tension à laquelle les broches d'entrée/sortie de la carte fonctionne (ex. VCC pour la carte). C'est 5V sur un Yun.
- Mémoire :

L'ATmega32u4 dispose de 32 KB (avec 4 KB utilisé par le bootloader). Il dispose également de 2.5 KB de SRAM et de 1 KB d'EEPROM (qui peut-être modifiée à l'aide de la librairie EEPROM).

La mémoire de l'AR9331 n'est pas incluse dans le processeur. La RAM et mémoire de stockage sont disposées à l'extérieur du processeur. L'Yún dispose de 64 MB de RAM DDR2 et de 16 MB de mémoire Flash. La mémoire flash est pré chargée en usine avec une distribution Linux basée sur OpenWRT. Cette distribution s'appelle Linino.

- Entrées et sorties :

Il n'est pas possible d'accéder aux broches d'entrées/sorties du processeur Atheros AR9331. Toutes les lignes I/O sont raccordées sur le 32U4.

Chacune des 20 broches d'entrée/sortie digitale du Yún peuvent être utilisées comme entrée("input") ou sortie("output") en utilisant les fonctions `pinMode ()`, `digitalWrite ()` et `digitalRead ()`. Elles fonctionnent en logique 5 volts. Chaque broche peut fournir et recevoir un maximum de 40mA et dispose de résistance pull-up (déconnectée par défaut) de 20-50 KOhms.

De surcroît, certaines broches ont des fonctions spécialisées:

- Série: 0 (RX) et 1 (TX) : Utilisé pour recevoir (RX) et transmettre (TX) des données sur un port série TTL en utilisant les capacités matérielles de l'ATmega32U4. Notez que sur le Yún, la classe *Serial* référence la communication USB (CDC); pour le port série TTL sur les broches 0 et 1, utilisez la classe *Serial1*. Sur le Yún, les ports séries matériels de l'ATmega32U4 et du AR9331 sont connectés ensembles et utilisés pour la communication entre les processeurs. Comme c'est couramment le cas sur les systèmes Linux, le port série du AR9331 expose une console fournissant un accès au système, cela signifie que vous pouvez accéder aux programmes et outils offerts par Linux depuis vos sketch/croquis.
- TWI: 2 (SDA) et 3 (SCL) : Support de la communication TWI (Two Wire... soit I2C) en utilisant la librairie Wire.
- Interruptions externes: 3 (interruption 0), 2 (interruption 1), 0 (interruption 2), 1 (interruption 3) et 7 (interruption 4) : Ces broches peuvent être configurées pour capturer des signaux d'interruption sur valeur "LOW", front montant ("rising edge"), front descendant ("falling edge) ou le changement de valeur. voir la fonction `attachInterrupt ()` pour plus d'information. Il n'est pas recommandé d'utiliser les broches 0 et 1 comme interruption parce qu'elles sont aussi utilisées par le port série matériel pour communiquer avec le processeur Linux. La broches 7 est connectée

sur le processeur AR9331 et pourrait être utilisé comme signal de handshake dans le future. Il est donc recommandé d'être prudent vis-à-vis des possibles conflits si vous l'utilisez comme interruption.

- PWM 3, 5, 6, 9, 10, 11 et 13 : Fournit des sorties PWM 8-bit avec la fonction `analogWrite()`.
- SPI: sur le connecteur ICSP. Ces broches supportent la communication SPI en utilisant la librairie SPI. Notez que les broches SPI ne sont pas connectées sur des broches d'entrée/sortie digitales comme c'est le cas pour Arduino Uno, elles sont uniquement disponible sur le connecteur ICSP. Cela signifie que si vous avez un shield SPI, mais que celui ci ne fonctionnera que s'il dispose du connecteur ICSP 6-broches qui se connecte sur le connecteur 6 broches ICSP correspondant du Yún.

Les broches SPI sont également connectées sur le GPIO du AR9331, où une interface SPI logicielle a été implémentée. Cela signifie que l'ATMega32u4 et l'AR9331 peuvent aussi communiquer en utilisant le protocole SPI.

- LED 13 : Il y a une LED intégrée à la carte et connectée sur la broche digitale 13. Lorsque la broche est à la valeur HIGH (état haut), la LED est allumée. La LED est éteinte lorsque la broche est à LOW (état bas).

Il y a plusieurs autres LED de statut sur le Yún indiquant l'état d'alimentation ("power"), connexion WLAN (Wifi), connexion WAN (Ethernet) et USB.

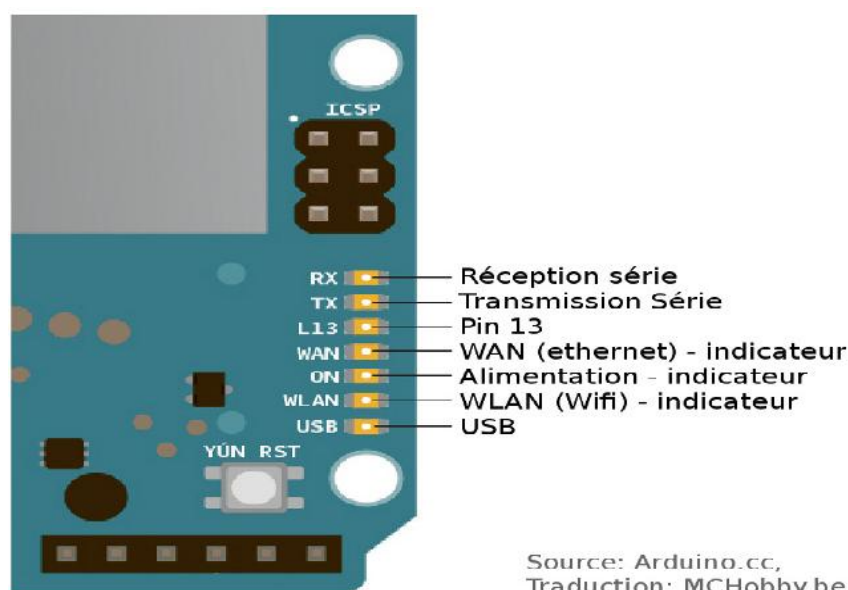


Figure 4.4 Les LED de statut

4.2.1.2 Afficheur LCD

- Présentation :

Dans notre carte on a besoin d'un afficheur LCD, autrement appelé un afficheur à cristaux liquide. Il consomme relativement de 1 à 5 mA et constitué de deux lames de verre, distantes de 20 μm environ, sur lesquelles sont dessinées les mers nantisses formant les caractères. L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3à 5V) le rend absorbant .Un afficheur à cristaux liquide ne peut être utilisé qu'avec un bon éclairage ambiant .Sa lisibilité augmente avec l'éclairage.



Figure 4.5 Afficheur LCD 16*2

- Fonctionnement :

Dans notre projet on va utiliser le mode 4 bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisés pour transmettre les données et les lire. Les bits de poids faible (D0 à D3) sont alors connectés à la masse, on a donc besoin hors alimentation de sept fils pour commander l'afficheur. Les données sont écrites séquentiellement les quatre bits de poids fort suivis des quatre bits de poids faible. Une impulsion positive d'au moins 450ns doit être envoyée sur la ligne E pour valider chaque demi -octet. On peut après chaque action sur l'afficheur vérifier que celui-ci est en mesure de traiter l'information suivante .Pour cela, il faut demander une lecture en mode commande et tester le flag Busy BT. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée.

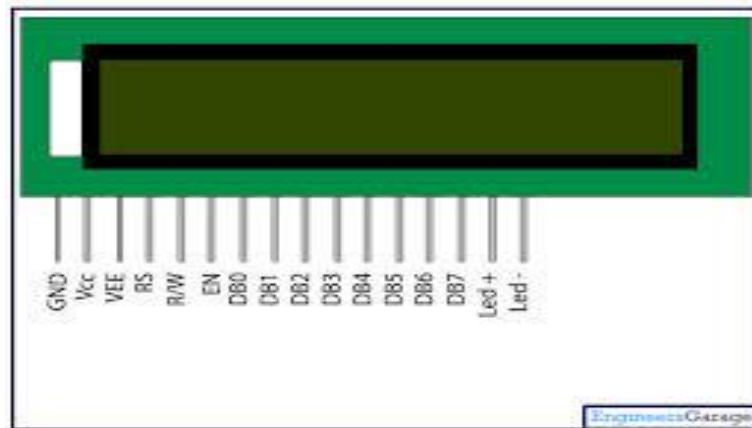


Figure 4.6 Les broches de l'afficheur

Au-dessus de l'écran à cristaux liquides, on trouve une série de 14 broches ayant les rôles Suivants :

- Broche 1 : masse ;
- Broche 2 : Vcc ;
- Broche 3 : luminosité ;
- Broche 5, R/W : sélection du mode lecture ou écriture :

0	écriture
1	lecture

- Broche 6, E : Commande des opérations d'écriture ou de lecture ;
- Broche 7 à 14 : utilisées pour le transfert des données ou des instructions. Le transfert peut se faire sur 8 bits, toutes les broches sont alors utilisées, ou sur 4 bits, dans ce cas, seules les broches 11 à 14 sont utilisées.

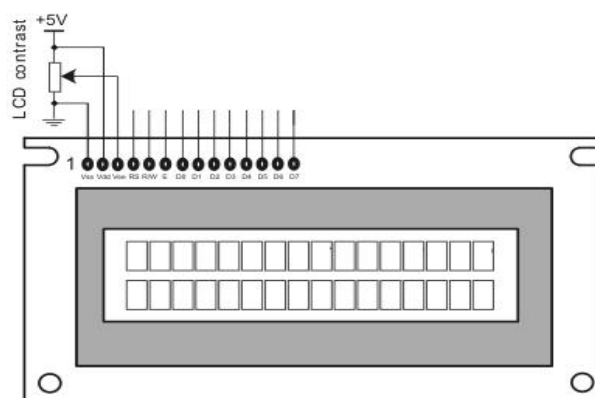


Figure 4.7 Présentation d'un afficheur LCD

4.2.1.3 Lecteur RFID RDM6300

- Présentation :

Le terme RFID désigne un système d'identification qui comprend une étiquette électronique (ou *tag*), pour mémoriser des informations, et un lecteur.

Le transfert d'information du composant électronique vers le lecteur s'effectue par radiofréquence et non par lecture optique, comme c'est le cas pour le code à barres. Contrairement au code à barres (qui permet d'identifier une famille de produits), l'étiquette RFID peut servir d'identifiant unique de l'objet marqué. De plus, l'étiquette RFID peut être lue à distance ; plusieurs étiquettes peuvent même être lues en parallèle (attention cependant aux interférences).

Une solution complète de RFID comprend les étiquettes, les lecteurs et encodeurs et l'intergiciel (*middleware*). Ce dernier permet d'intégrer le flux des données dans le système d'information de l'entreprise.

Les étiquettes et les encodeurs sont construits par des entreprises comme Philips, Texas Instruments (TI), Toshiba, Printronix, ASK, Intermec ou Tagsys. Les intergiciels sont proposés des entreprises comme Oracle, Sun, IBM, Axway ou Microsoft.



Figure 4.8 Lecteur RFID 125 KHz

- Différents types de RFID :

Les étiquettes peuvent être passives ou actives. Plus chères, les étiquettes actives sont dotées d'une alimentation. Elles offrent des distances de lecture supérieures. Plus économiques, les étiquettes passives ne disposent pas d'alimentation et offrent une distance de lecture réduite. Les étiquettes diffèrent par leur fréquence : basse, haute ou très haute.

Plus coûteuses, les étiquettes basse fréquence sont réutilisables. Elles se lisent à quelques mètres de distance.

Les étiquettes haute fréquence (HF) coûtent au moins 50 centimes. Elles sont jetables ou réutilisables suivant les cas. La distance de lecture est inférieure à un mètre.

Les étiquettes très haute fréquence (UHF) coûtent 15 à 25 centimes. La distance de lecture est de deux à quatre mètres. Par contre, la tolérance aux obstacles est moins bonne.

- Applications de la RFID :

- L'accès aux transports publics.
- L'accès sans formalité aux bibliothèques.
- Le remplacement des codes à barres dans les supermarchés.
- Les postes de péage automatique sur les autoroutes.
- Les applications de logistique (pour suivre l'acheminement de marchandises).

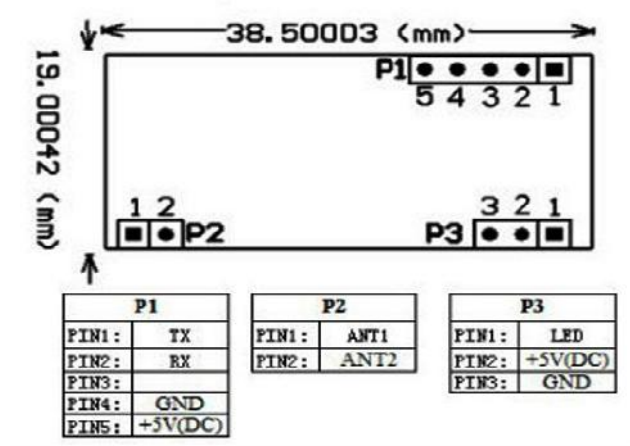


Figure 4.9 Présentation d'un lecteur RFID

4.2.1.4 Buzzer

- Présentation :

Un buzzer est un élément électromécanique ou électrique qui produit un son quand on lui applique une tension. Certains nécessitent une tension continue (buzzer électromécaniques), d'autres nécessitent une tension alternative (transducteurs piézo-électrique).

Un buzzer mécanique se présente sous la forme d'un petit boîtier rectangulaire ou cylindrique, avec connexion électriques rigides pour fixation directe sur circuit imprimé, ou avec connexion électriques constituées de fils électriques souples. Dans ce deuxième cas, le buzzer possède deux petites pattes de fixation. La puissance sonore d'un tel composant est de l'ordre de 85 dB/cm (notez que l'on ne spécifie pas le niveau sonore à un mètre – comme pour les HP, car d'un point de vue commercial, ça paraîtrait sans doute trop peu puissant. Comme pour les confiseries dont on donne le prix pour 100g et non pour un kg).

Il nécessite une tension continue pour fonctionner, cette dernière doit généralement être comprise entre 3V et 28V, selon les modèles. Un buzzer prévu pour fonctionner sous 6V fonctionne généralement très bien pour toute tension d'alimentation comprise entre 4V et 8V comme notre buzzer, il ne nécessite aucun composant annexe pour fonctionner, hormis bien sur l'éventuel étage de commande qui va l'activer. On lui fournit une tension continue simple et hop, ça sonne.



Figure 4.10 Buzzer

4.2.1.5 LED RGB

- Présentation :

Cette LED RGB de diamètre 5 mm avec boîtier transparent dispose de 4 broches. 1 cathode et 3 anodes. Chaque anode correspond à une couleur Rouge, vert et bleu. En modulant les signaux sur les anodes, il est possible d'obtenir de multiples couleurs. A noter que nous proposons également un modèle avec boîtier diffusant en bas de page. Elle est en fait composée de 3 LEDs de couleurs différentes enfermées dans un seul et même composant. Les meilleures LEDs RGB seront les LEDs dite à diffusion (LEDs tricolores diffuses).



Figure 4.11 LED RGB

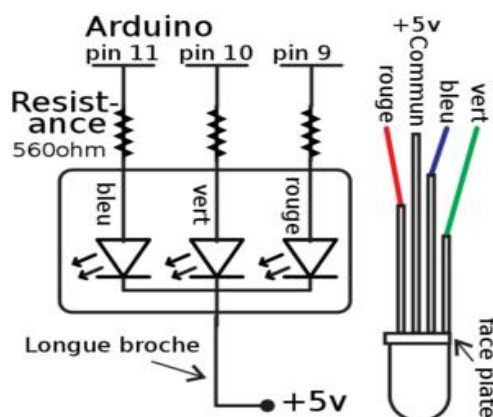


Figure 4.12 Montage de la LED RGB

4.2.2 Les logiciels utilisés

- Arduino IDE :



Figure 4.13 Logo Arduino IDE

Le développement sur Arduino est très simple :

- On code l'application : le langage Arduino est basé sur les langages C/C++, avec des fonctions et des bibliothèques spécifiques à Arduino (gestion des e/s).
- On relie la carte arduino au PC et on transfère le programme sur la carte.
- On peut utiliser le circuit

Le logiciel de programmation des modules arduino est une application java multiplateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le firmware (et le programme) au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

Le logiciel est très simple à prendre en main, il existe de très bons tutoriaux très bien faits avec même des explications en français. De très nombreux exemples sont fournis. Les fichiers exemples sont vraiment bien documentés et permettent de coder des choses très compliquées sans trop d'efforts.

Les bibliothèques fournies permettent d'utiliser des composants complexes très simplement en quelques lignes très claires (afficheur ou liaison SPI etc...).

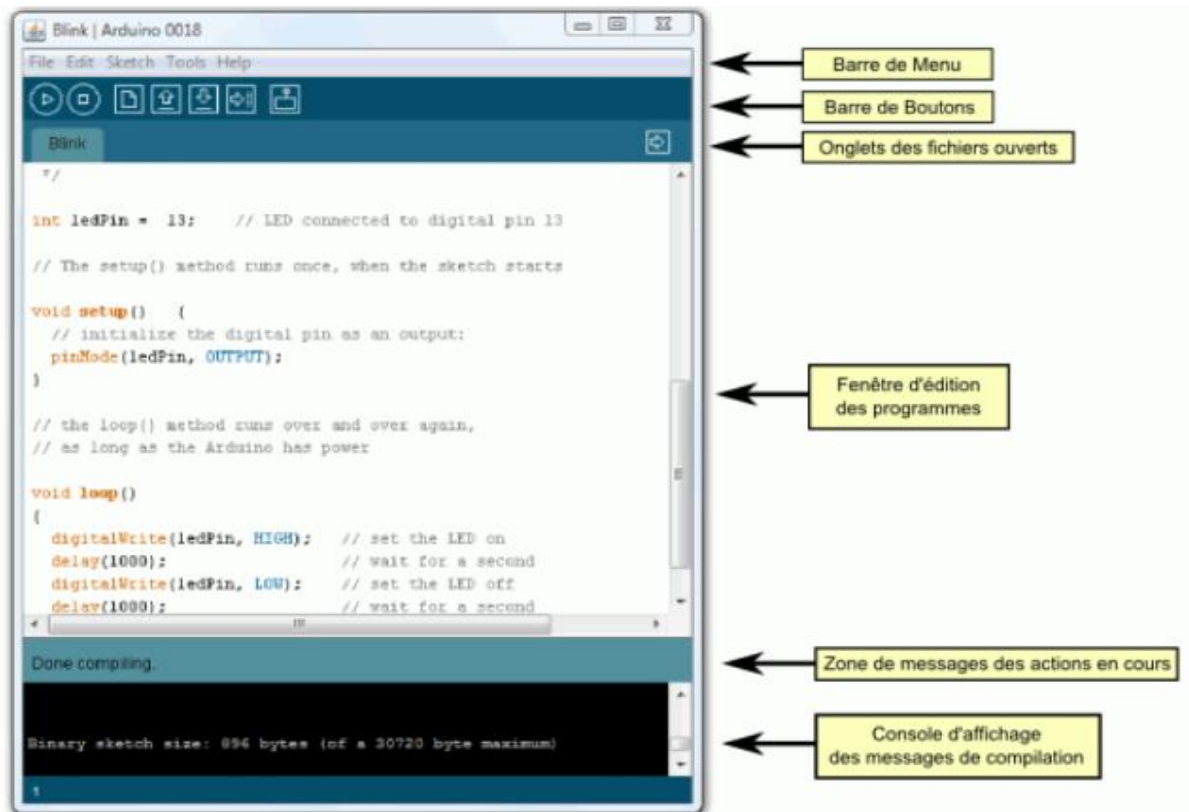


Figure 4.14 L'environnement de programmation arduino

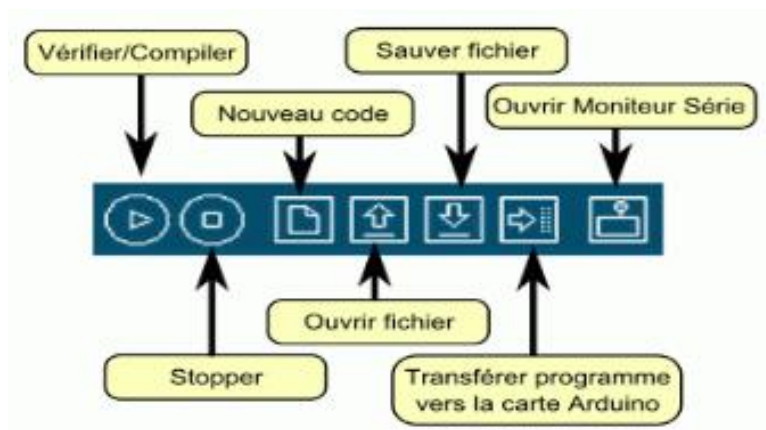


Figure 4.15 Détails de la barre de boutons

- Eagle :



Figure 4.16 Logo Eagle

Eagle (Easily Applicable Graphical Layout Editor) est un logiciel de conception assistée par ordinateur de circuits imprimés et un peu similaire à ISIS Proteus.

Il comprend un éditeur de schémas, un logiciel de routage de circuit imprimé avec une fonction d'auto routage, et un éditeur de bibliothèques. Le logiciel est fourni avec une série de bibliothèques de composants de base.

C'est un logiciel multiplateforme.

Il est populaire auprès des amateurs parce qu'il existe une version de démonstration gratuite. La version gratuite est limitée à deux couches au format européen (100x80 mm) et ne permet pas de faire de l'auto routage.

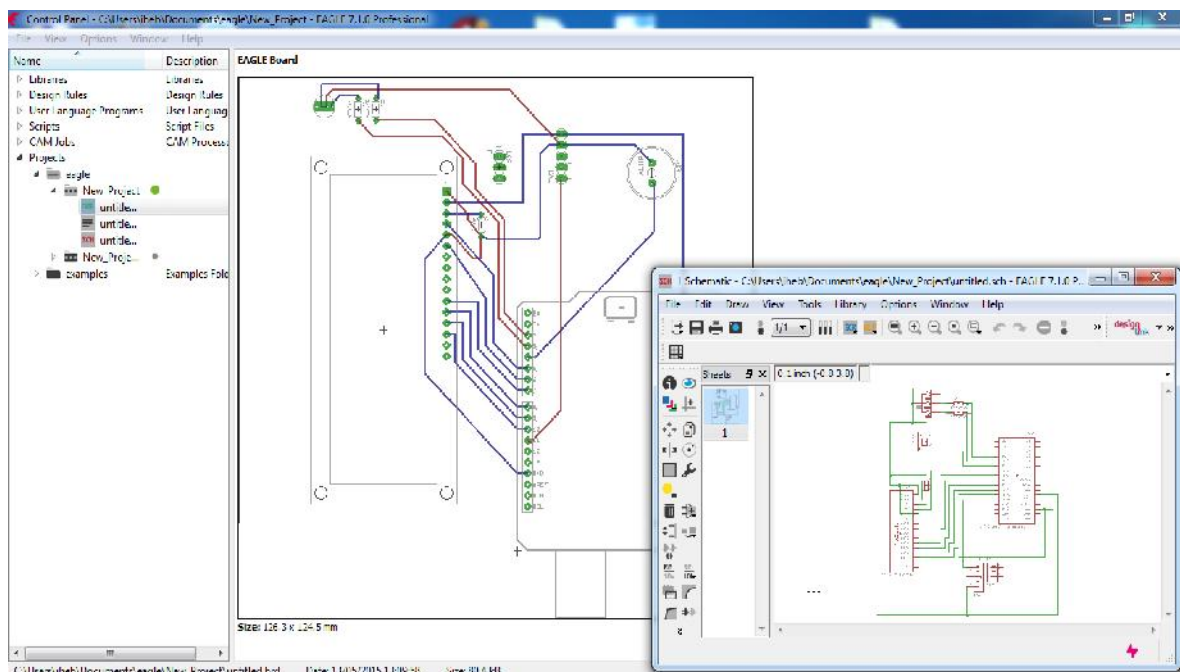


Figure 4.17 Interface de l'Eagle

4.2.3 Les programmes et les tests réalisés

- Test1 : Arduino Yun + RFID RDM6300

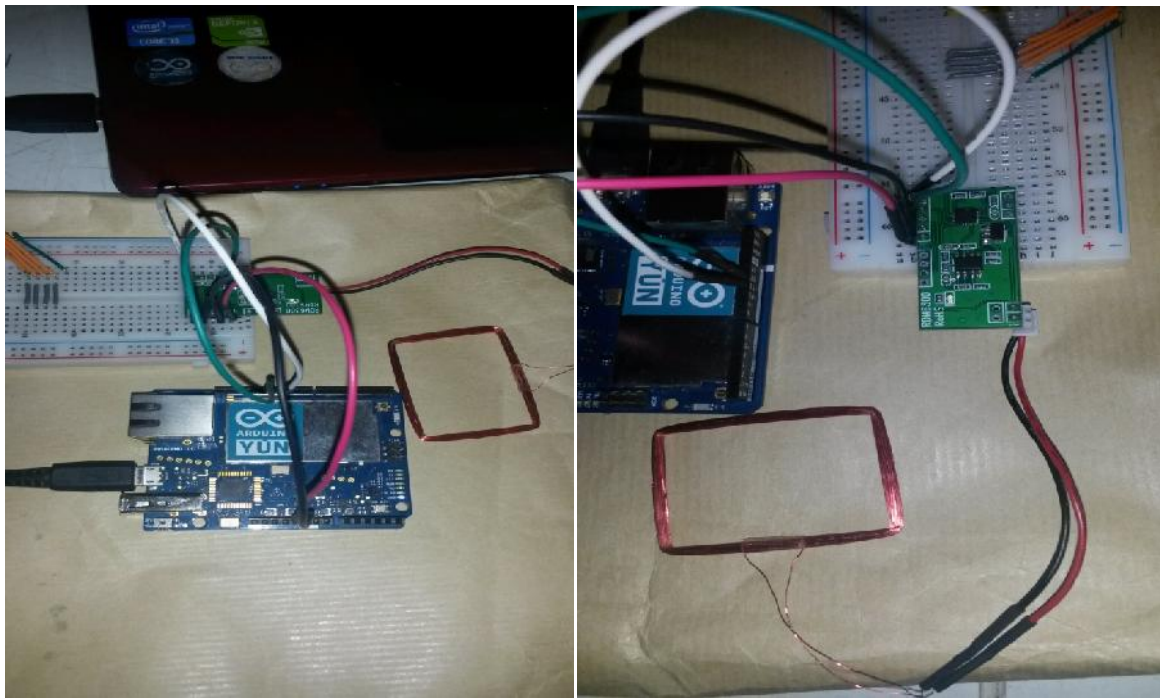


Figure 4.18 Réalisation du 1er test

```
COM3 (Arduino Yun)
CP00312A35A1 OF0C312ADCA1 CF00312A35A1 OF0C312ADCA1 CF0031
CP00312A35A1 OF0C312ADCA1 CF00312A35A1 OF0C312ADCA1 CF0031
CP00312A35A1 OF0C312ADCA1 CF00312A35A1 OF0C312ADCA1 CF0031
CP00312A35A1 OF0C312ADCA1 CF00312A35A1 OF0C312ADCA1 CF0031
1C0021C4C1F4 100C2104C1F4 130021C4C1F4 100C2104C1F4 130021
1C0021C4C1F4 100C2104C1F4 130021C4C1F4 100C2104C1F4 130021
1C0021C4C1F4 100C2104C1F4 130021C4C1F4 100C2104C1F4 130021
1C0021C4C1F4 100C2104C1F4 130021C4C1F4 100C2104C1F4 130021
1C0021C4C1F4 100C2104C1F4 130021C4C1F4 100C2104C1F4 130021

capture: Arduino 1.6.1
Fichier Edition Croquis Outils Aide
//capture$
#include <Bridge.h>
#include <SoftwareSerial.h>
#define RXPIN 12 //Pin to read data
#define TXPIN 12 //Pin to write data
SoftwareSerial rfid(RXPIN, TXPIN); // RX and TX
String msg = "";
void setup()
{
  rfid.begin(5000); // start serial to RFID reader
  Serial.begin(115200);
  Serial.println("Swipe a card");
}
char c;
void loop()
{
  if (rfid.available())
  {
    while (rfid.available() > 0)
    {
      c = rfid.read();
      msg += c;
    }
    Serial.println(msg);
    digitalWrite(LED, HIGH);
    delay(3000);
    digitalWrite(LED, LOW);
    delay(3000);
    msg = "";
  }
}
```

Figure 4.19 Code source du test 1

- Code :

```
#include <Bridge.h>
#include <SoftwareSerial.h>
#define RxPin 11 //Pin to read data
#define TxPin 12 //Pin to write data
SoftwareSerial rfid (RxPin,TxPin); // RX and TX
String msg="";
void setup()
{
  rfid.begin(9600); // start serial to RFID reader
  Serial.begin(115200);
  Serial.println("swipe u card");
}
char c;
void loop(){
  if(rfid.available()){
    while(rfid.available()>0)
    { c= rfid.read();
      msg += c;
    }
    Serial.println(msg);
    digitalWrite(2, HIGH);
    delay(3000);
    digitalWrite(2, LOW);
  }
  delay(3000);
  msg = "";
}
```

- Resultat:

10002104C1F4 10002104C1F4 10002104C1F4 10002104C1F4 10002104C1F4
10002104C1F4 10002104C1F4 10002104C1F4

Le lecteur RFID lit les Tags sans s'arrêter et à chaque fois il nous donne 12 caractères, donc nous avons besoin de lire une seule fois et juste 8 caractères.

- Solution et resultat:

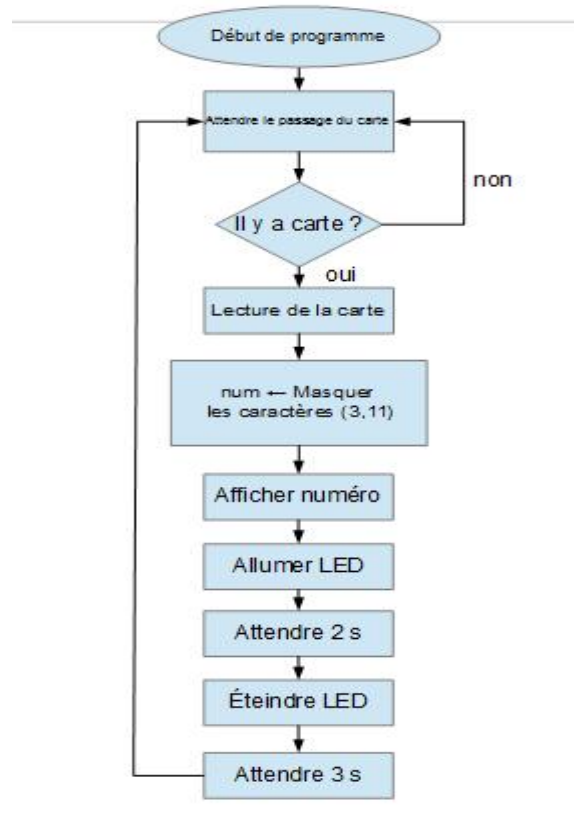


Figure 4.20 Logigramme de lecture des cartes

Code :

```
#include <Bridge.h>
#include <SoftwareSerial.h>
#define RxPin 11 //Pin to read data
#define TxPin 12 //Pin to write data
SoftwareSerial rfid (RxPin,TxPin); // RX and TX
String msg="";
void setup()
{
  rfid.begin(9600); // start serial to RFID reader
  Serial.begin(115200);
  Serial.println("swipe u card");
}
char c;
void loop(){
```



```
if(rfid.available()){
while(rfid.available())>0)
{ c= rfid.read();
  msg += c;
}
if (msg.length() > 10) {
  msg = msg.substring(3,11);
}
Serial.println(msg);
digitalWrite(2, HIGH);
delay(3000);
digitalWrite(2, LOW);
}
delay(3000);
msg = "";
}
```

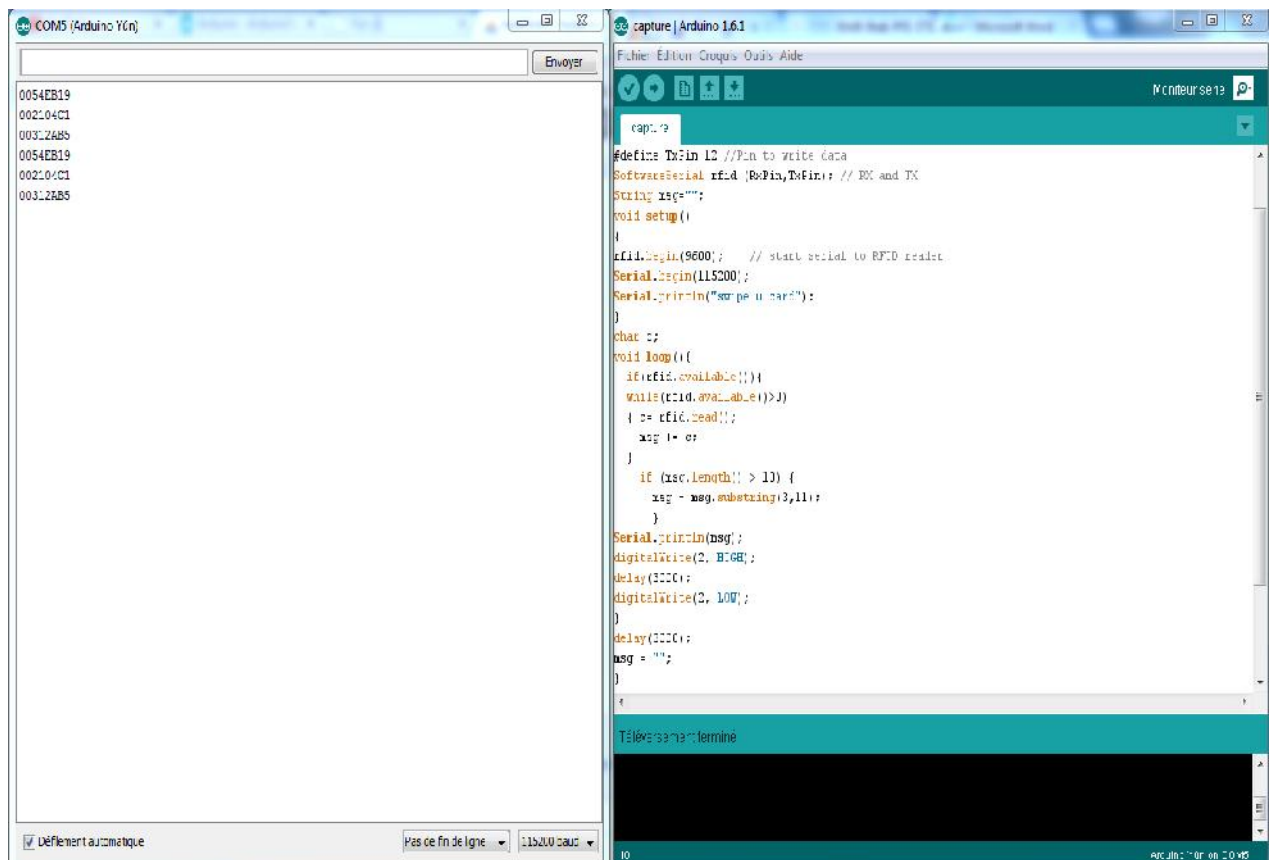


Figure 4.21 Solution du test 1

Rapport de stage de fin d'études

Résultat :

0054EB19

002104C1

00312AB5

0054EB19

002104C1

00312AB5

- Test2 : l'envoi des Tags vers la base de données

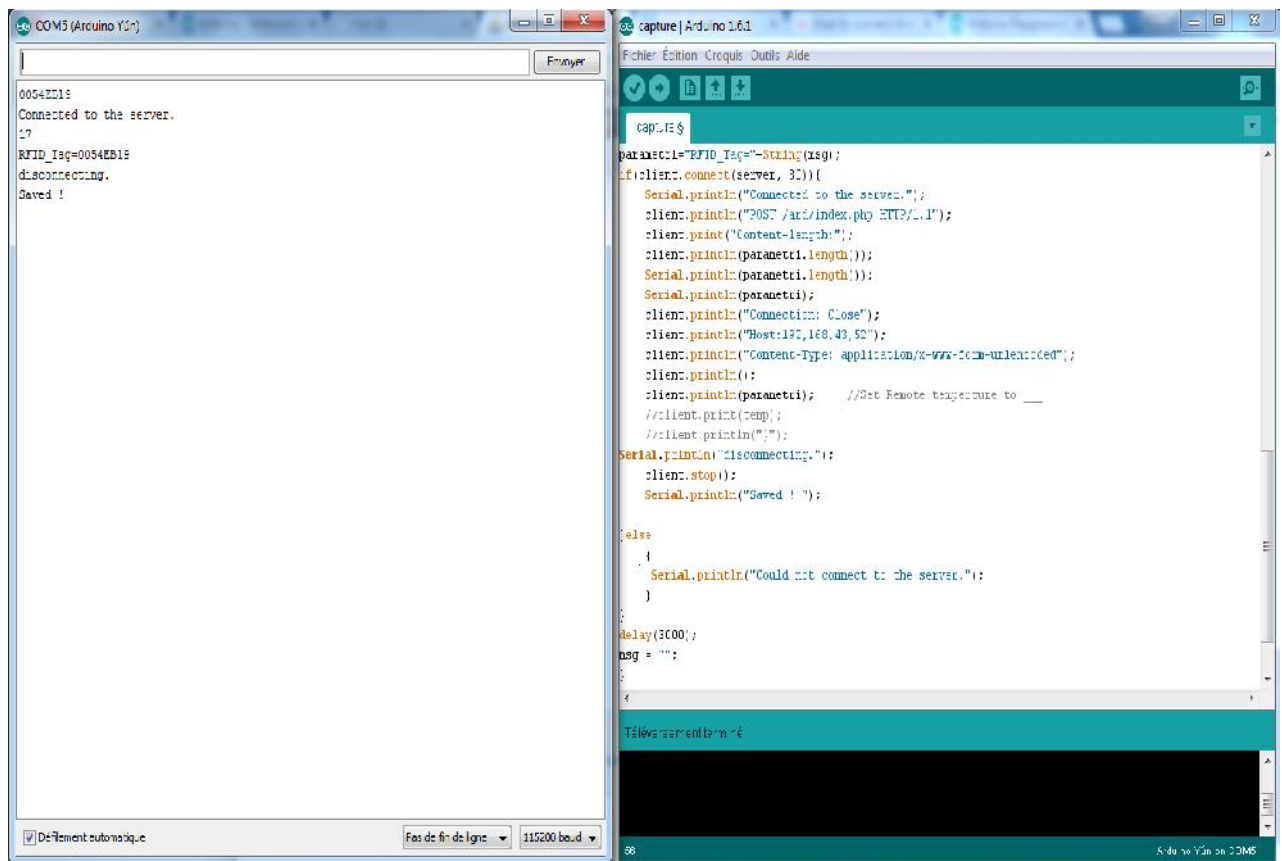


Figure 4.22 Code source du test2 et résultat

- Code :

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <HttpClient.h>
```

```
#include <SPI.h>
#include <SoftwareSerial.h>
#define RxPin 11 //Pin to read data
#define TxPin 12 //Pin to write data
SoftwareSerial rfid (RxPin,TxPin); // RX and TX
String msg="";
String parametri ="";
YunClient client;
IPAddress server(192,168,43,52);
void setup()
{
  Bridge.begin();
  rfid.begin(9600); // start serial to RFID reader
  Serial.begin(115200);
  Serial.println("swipe u card");
}
char c;
void loop(){
  if(rfid.available()){
    while(rfid.available()>0)
    { c= rfid.read();
      msg += c;
    }
    if (msg.length() > 10) {
      msg = msg.substring(3,11);
    }
    Serial.println(msg);
    digitalWrite(2, HIGH);
    delay(3000);
    digitalWrite(2, LOW);

    parametri="RFID_Tag="+String(msg);
    if(client.connect(server, 80)){
      Serial.println("Connected to the server.");
    }
  }
}
```

```
client.println("POST /ard/index.php HTTP/1.1");
client.print("Content-length:");
client.println(parametri.length());
Serial.println(parametri.length());
Serial.println(parametri);
client.println("Connection: Close");
client.println("Host:192,168,43,52");
client.println("Content-Type: application/x-www-form-urlencoded");
client.println();
client.println(parametri);    //Set Remote temperture to ____
//client.print(temp);
//client.println("{}");
Serial.println("disconnecting.");
client.stop();
Serial.println("Saved ! ");

}else
{
    Serial.println("Could not connect to the server.");
}
}
delay(3000);
msg = "";
}

-   Résultat :

0054EB19
Connected to the server.
17
RFID_Tag=0054EB19
disconnecting.
Saved !
```

Rapport de stage de fin d'études

- Test3 : l'ajout d'un afficheur LCD + buzzer + LED RGB

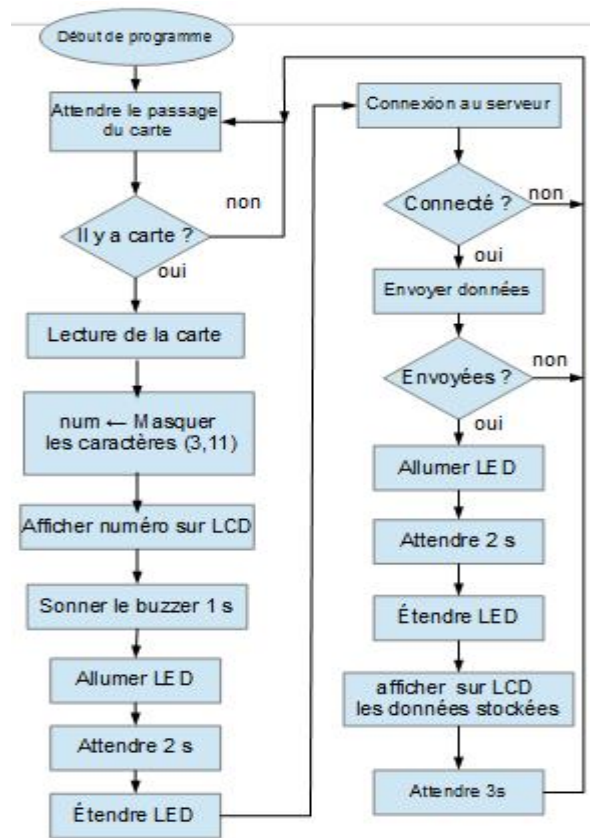


Figure 4.23 Logigramme de lecture des cartes avec envoi de numéro vers la base

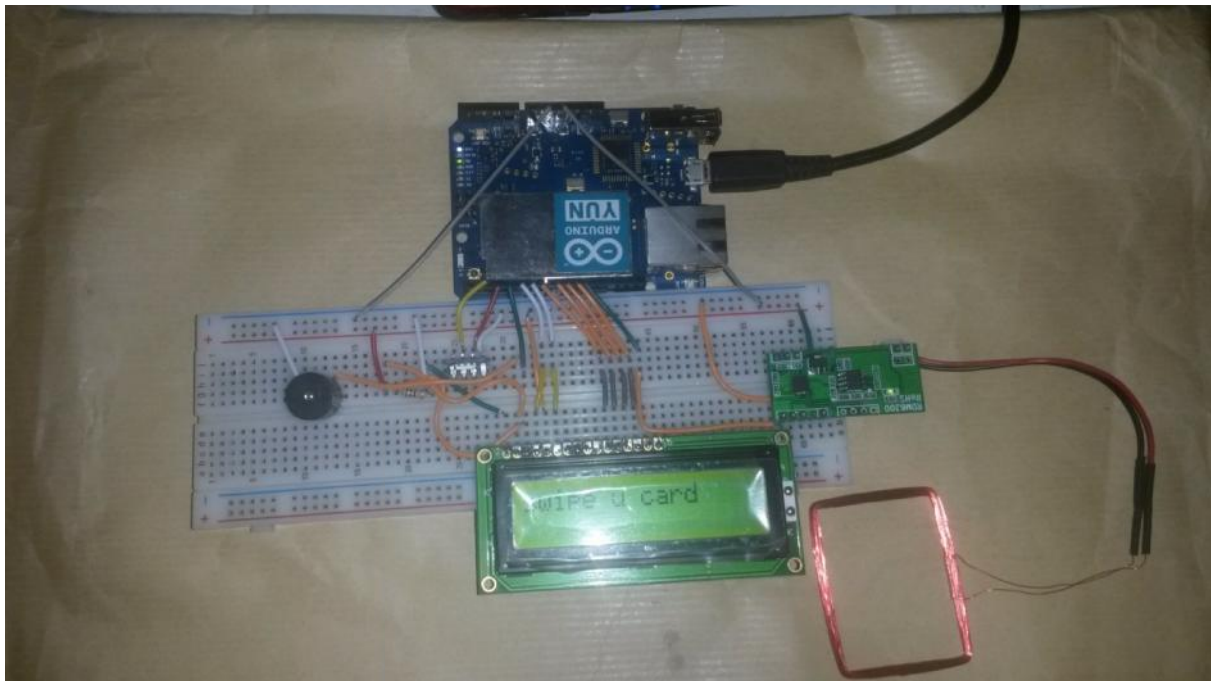


Figure 4.24 Arduino + RFID + LCD + RGB + Buzzer

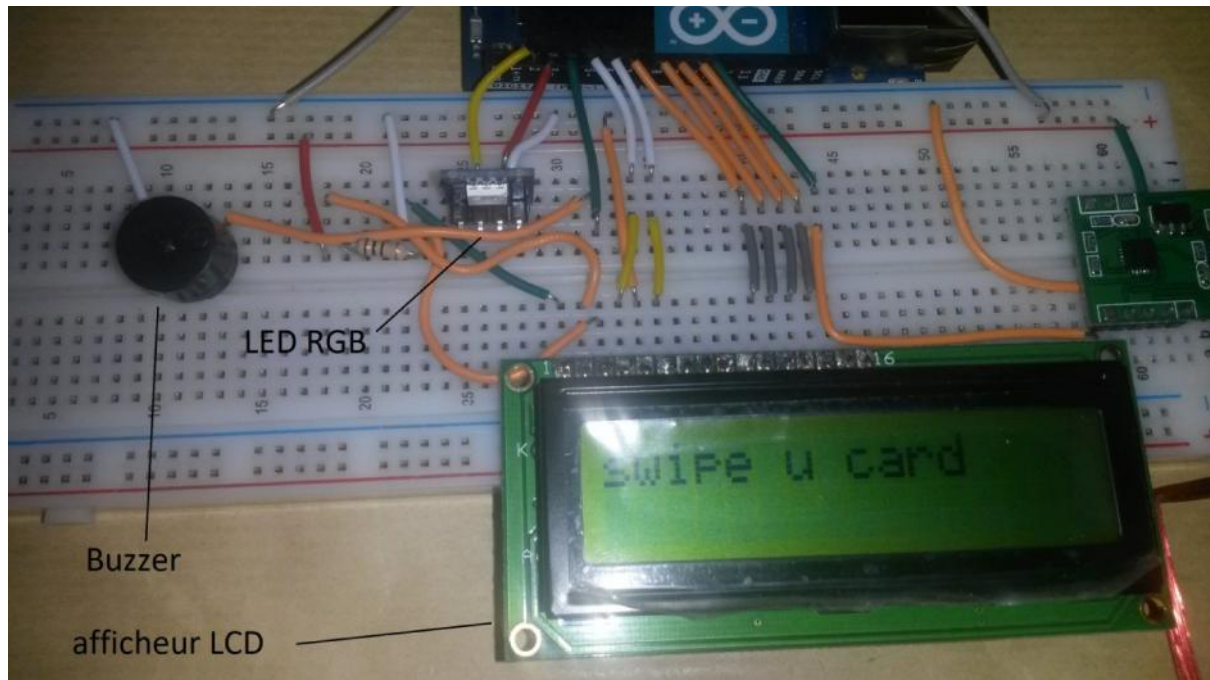


Figure 4.25 l'ajout de LCD + Buzzer + LED RGB

```
rfidinal
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <HttpClient.h>
#include <SPI.h>

#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#define RxPin 11 //?pin to read data
#define TxPin 12 //?pin to write data

SoftwareSerial rfid (RxPin,TxPin); // RX and TX
LiquidCrystal lcd (5,6,7,8,9,10);

YunClient client;

IPAddress server(192,168,43,53);

String msg="";
String parametri="";

void setup()
{
  Bridge.begin();

  rfid.begin(9600); // start serial to RFID reader

  Serial.begin(115200);

  pinMode(4,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(2,OUTPUT);

  Serial.println("swipe u card");
  lcd.begin(16, 2);
  lcd.print("swipe u card");

  char c;
  void loop(){

    if(rfid.available()){
      while(rfid.available()>0)
      {
        c= rfid.read();
      }
    }
  }
}
```

```
msg += C;
}

if (msg.length() > 10) {
    msg = msg.substring(3,11);
}
Serial.println(msg);
lcd.clear();
lcd.print(msg);
digitalWrite(2, HIGH);
buzz(4, 5000, 500);
delay(3000);
digitalWrite(2, LOW);

parametri="RFID_Tag="+String(msg);

if(client.connect(server, 80)){
    Serial.println("Connected to the server.");

    client.println("POST /ard/index.php HTTP/1.1");
    client.print("Content-length:");
    client.println(parametri.length());
    Serial.println(parametri.length());
    Serial.println(parametri);
    client.println("Connection: Close");

    client.println("Host:192.168.43.53");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println();
    client.println(parametri); //Set Remote temperature to ____
    //client.print(temp);
    //client.println("}");

    digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(2, LOW); // turn the LED off by making the voltage LOW

    Serial.println("disconnecting.");
    client.stop();
    lcd.clear();
    lcd.print("Saved ");
} else {
    digitalWrite(3, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(3, LOW); // turn the LED off by making the voltage LOW

    Serial.println("Could not connect to the server.");
    lcd.clear();
    lcd.print("error again plz.");
}

}

delay(3000);
parametri = "";
msg = "";
lcd.clear();
lcd.print("swipe u card");
}

void buzz(int targetPin, long frequency, long length) {
    long delayValue = 1000000/frequency/2; // calculate the delay value between
    // 1 second's worth of microseconds, divided by the frequency, then subtract
    // there are two phases to each cycle
    long numCycles = frequency * length/ 1000; // calculate the number of cycles
    // multiply frequency, which is really cycles per second, by the number of
    // seconds to get the total number of cycles to produce
    for (long i=0; i < numCycles; i++){ // for the calculated length of time
        digitalWrite(targetPin,HIGH); // write the buzzer pin high to push out
        delayMicroseconds(delayValue); // wait for the calculated delay value
        digitalWrite(targetPin,LOW); // write the buzzer pin low to pull back in
        delayMicroseconds(delayValue); // wait again for the calculated delay value
    }
}
```

Figure 4.26 Code source du 3em test

- Programme final :

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
#include <HttpClient.h>
#include <SPI.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#define RxPin 11 //Pin to read data
#define TxPin 12 //Pin to write data
SoftwareSerial rfid (RxPin,TxPin); // RX and TX
LiquidCrystal lcd (5,6,7,8,9,10);
YunClient client;
IPAddress server(192,168,43,53);
String msg="";
String parametri="";
void setup()
{
  Bridge.begin();
  rfid.begin(9600); // start serial to RFID reader
  Serial.begin(115200);
  pinMode(4,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(2,OUTPUT);
  Serial.println("swipe u card");
  lcd.begin(16 , 2);
  lcd.print("swipe u card");
}
char c;
void loop(){
  if(rfid.available()){
    while(rfid.available()>0)
    {   c= rfid.read();
        msg += c;
    }
```

```
}
if (msg.length() > 10) {
    msg = msg.substring(3,11);
}
Serial.println(msg);
lcd.clear();
lcd.print(msg);
digitalWrite(2, HIGH);
buzz(4, 5000, 500);
delay(3000);
digitalWrite(2, LOW);
parametri="RFID_Tag="+String(msg);
if(client.connect(server, 80)){
    Serial.println("Connected to the server.");
    client.println("POST /ard/index.php HTTP/1.1");
    client.print("Content-length:");
    client.println(parametri.length());
    Serial.println(parametri.length());
    Serial.println(parametri);
    client.println("Connection: Close");
    client.println("Host:192,168,43,53");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println();
    client.println(parametri);    //Set Remote temperture to ____
    //client.print(temp);
    //client.println("}");
    digitalWrite(2, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);          // wait for a second
    digitalWrite(2, LOW); // turn the LED off by making the voltage LOW
    Serial.println("disconnecting.");
    client.stop();
    lcd.clear();
    lcd.print("Saved ! ");
}else
```



```
{digitalWrite(3, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000);           // wait for a second
digitalWrite(3, LOW);  // turn the LED off by making the voltage LOW
Serial.println("Could not connect to the server.");
lcd.clear();
lcd.print("error!again plz.");
}
}
delay(3000);
parametri = "";
msg = "";
lcd.clear();
lcd.print("swipe u card");
}

void buzz(int targetPin, long frequency, long length) {
    long delayValue = 1000000/frequency/2; // calculate the delay value between transitions
    //// 1 second's worth of microseconds, divided by the frequency, then split in half since
    //// there are two phases to each cycle
    long numCycles = frequency * length/ 1000; // calculate the number of cycles for proper
    timing
    //// multiply frequency, which is really cycles per second, by the number of seconds to
    //// get the total number of cycles to produce
    for (long i=0; i < numCycles; i++){ // for the calculated length of time...
        digitalWrite(targetPin,HIGH); // write the buzzer pin high to push out the diaphragm
        delayMicroseconds(delayValue); // wait for the calculated delay value
        digitalWrite(targetPin,LOW); // write the buzzer pin low to pull back the diaphragm
        delayMicroseconds(delayValue); // wait againf or the calculated delay value
    }
}
```

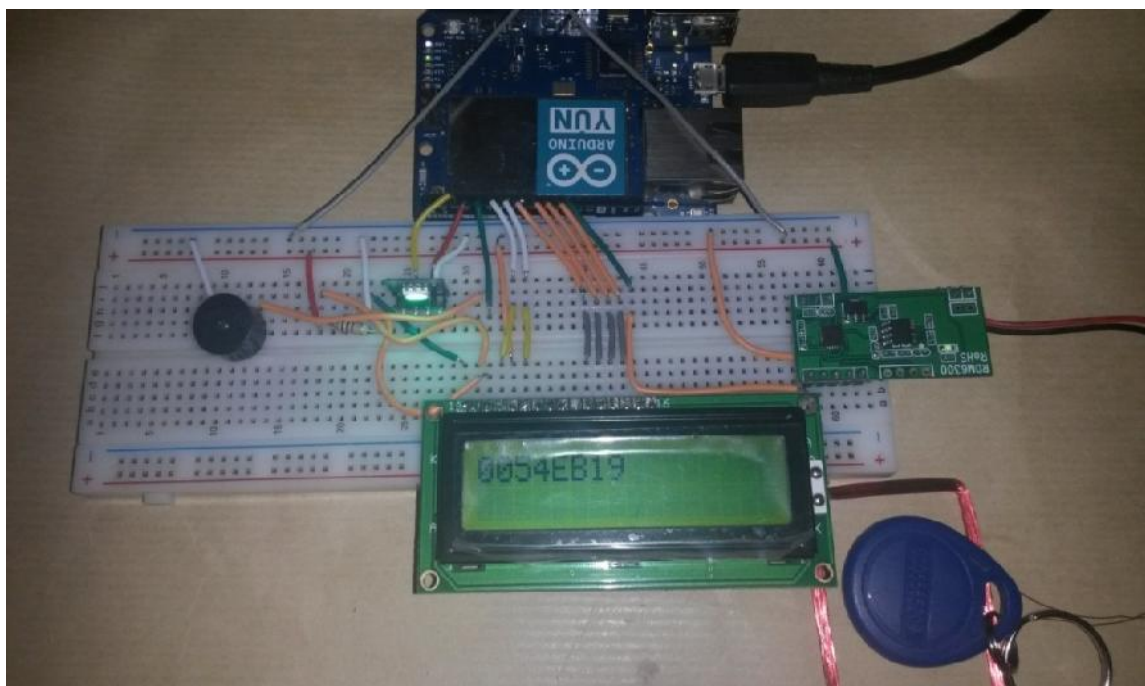


Figure 4.27 Test final (lecture de badge)

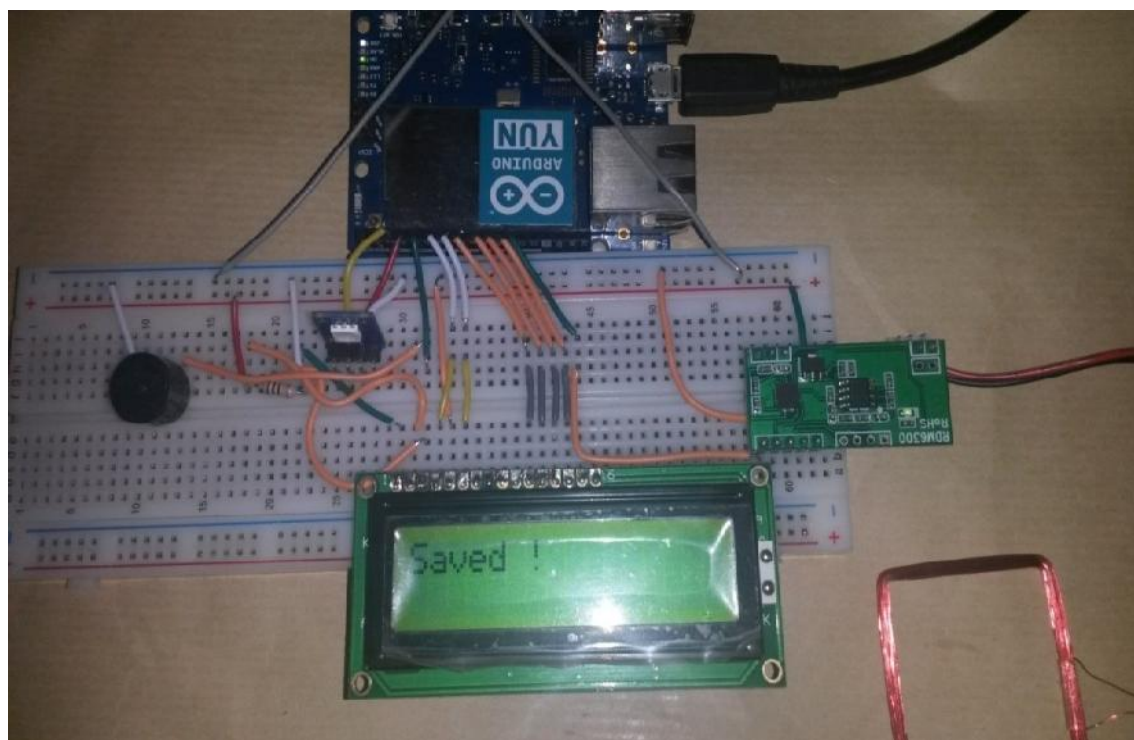


Figure 4.28 Test final (envoi à la base de données)

4.2.4 Réalisation de la carte imprimée

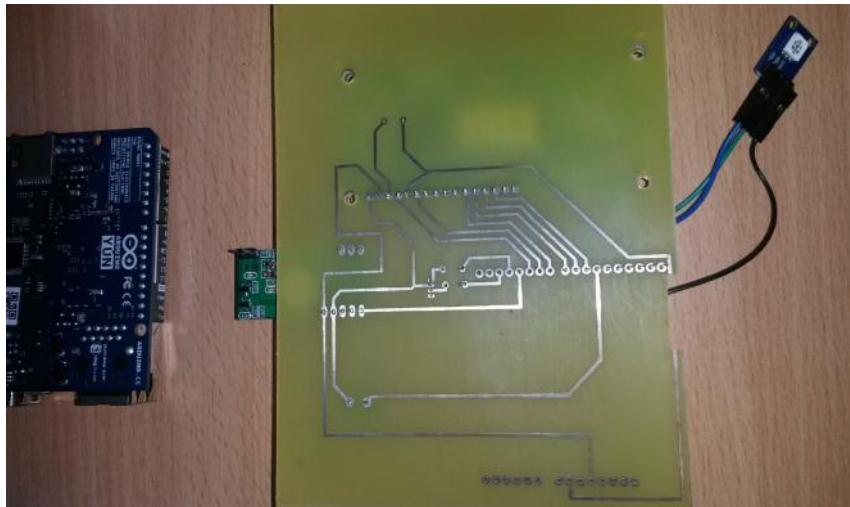


Figure 4.29 la carte imprimée



Figure 4.30 la carte imprimée avec l'Arduino



Figure 4.31 Face avant de la carte

4.3 Partie informatique

4.3.1 Les logiciels et les outils informatiques utilisés

Après l'envoi du numéro de la carte RFID vers une adresse spécifique, j'ai besoin de prendre ce numéro comme paramètre et l'ajouter à la base de données dans le tableau concerné avec enregistrement du temps et de l'état du pointage soit « IN » ou « OUT » selon un logigramme bien détaillé.

Pour cela j'ai utilisé ces logiciels et outils :

- Dreamweaver :

C'est un éditeur de site web, il fut l'un des premiers éditeurs HTML et plus efficace que le bloc note.



Figure 4.32 Logo Dreamweaver

- WampServer :

Permet de faire fonctionner localement des scripts PHP et sert aussi de gérer la base de données (locale).



Figure 4.33 Logo WampServer

- Bootstrap :

Le Bootstrap est une collection d'outils utiles à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de

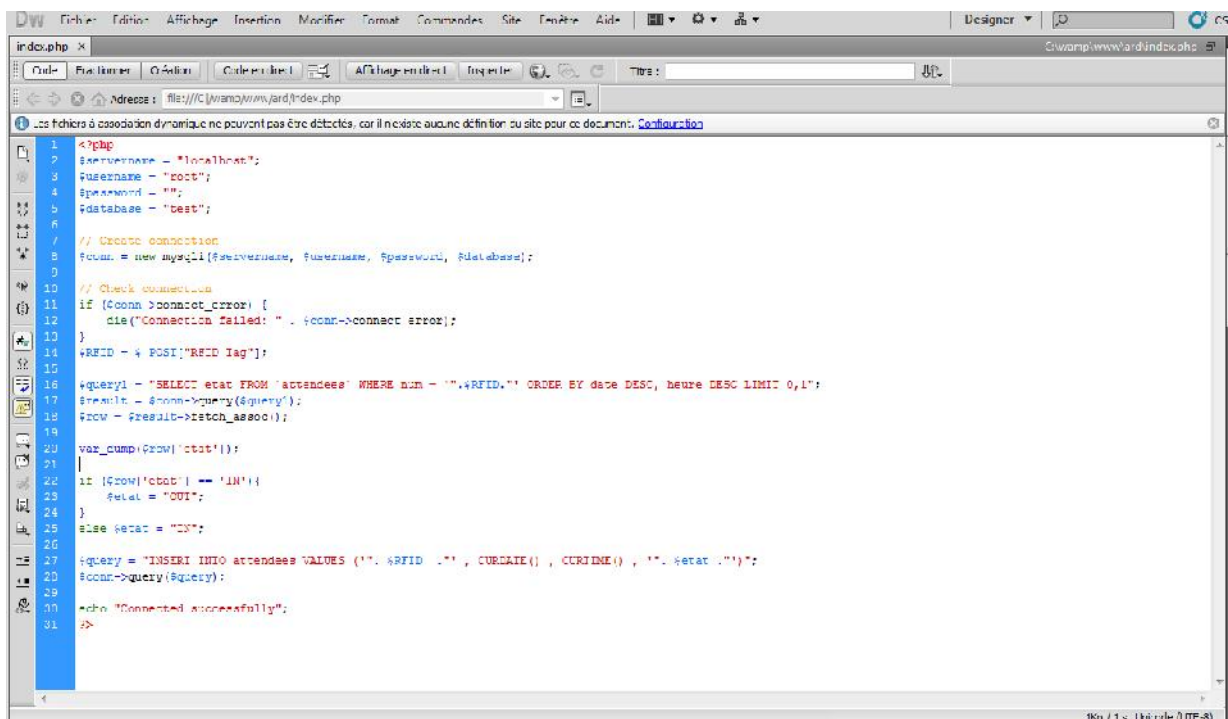
navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.



Figure 4.34 Logo Bootstrap

4.3.2 Réalisation de l'interface graphique

- Insertion dans la base :



```
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 $database = "test";
6
7 // Create connection
8 $conn = new mysqli($servername, $username, $password, $database);
9
10 // Check connection
11 if ($conn->connect_error) {
12     die("Connection failed: " . $conn->connect_error);
13 }
14 $RFID = $_POST["RFID_Tag"];
15
16 $query1 = "SELECT etat FROM 'attendeas' WHERE nom = '" . $RFID . "' ORDER BY date DESC, heure DESC LIMIT 0,1";
17 $result = $conn->query($query1);
18 $row = $result->fetch_assoc();
19
20 var_dump($row['etat']);
21
22 if ($row['etat'] == 'IN') {
23     $etat = "OUT";
24 }
25 else {
26     $etat = "IN";
27 }
28 $query = "INSERT INTO attendeas VALUES ('" . $RFID . "', CURDATE(), CURTIME(), '" . $etat . "')";
29 $conn->query($query);
30
31 echo "Connected successfully";
32 >>
```

Figure 4.35 Insertion dans la base

Le fichier index.php contient :

- La connexion :

La connexion à la base de données et le tableau qui doit contenir nos paramètres.

- Le choix d'état :

J'ai choisi l'état de pointage selon le dernier état d'employé concerné :

- o \$RFID = \$_POST["RFID_Tag"]; // enregistrer le numéro envoyé par l'arduino

- \$query1 = "SELECT etat FROM `attendees` WHERE num = '". \$RFID.'" ORDER BY date DESC, heure DESC LIMIT 0,1";

```
$result = $conn->query($query1);
```

```
$row = $result->fetch_assoc();
```

```
var_dump($row['etat']);
```

```
if ($row['etat'] == 'IN'){
```

```
    $etat = "OUT";}
```

```
else $etat = "IN"; // le test sur le dernier état et le choix d'état à ajouter
```

- L'insertion des paramètres :

Le numéro qui est déjà enregistré sous \$RFID. La date et l'heure par des fonctions PHP (CURDATE, CURTIME).

- \$query = "INSERT INTO attendees VALUES ('". \$RFID ."', CURDATE() , CURTIME() , '". \$etat . "')";

```
$conn->query($query); // l'insertion des paramètres (numéro, date, heure, état)
```

- L'affichage dans une page web :

Notre interface graphique est un site web local contenant trois pages web définies comme ci-dessous :

- Accueil :

C'est la page d'accueil (page principale de notre site web), elle se distingue des autres pages du site par le fait qu'elle est censée représenter, à l'internaute visiteur, le site sur lequel il se trouve de manière claire et forte.

Elle peut être chargée de plusieurs liens et des articles, plus le nombre de liens est important, plus il est probable que l'utilisateur y trouve quelque chose qui soit susceptible de l'intéresser. Mais plus le nombre de liens est important, plus il est difficile pour l'utilisateur de repérer celui qui l'intéresse.

Cette page peut contenir des notes, des affiches, des images d'employés, des remarques, des alertes ou bien un login selon les besoins de personnel de la société.

Rapport de stage de fin d'études

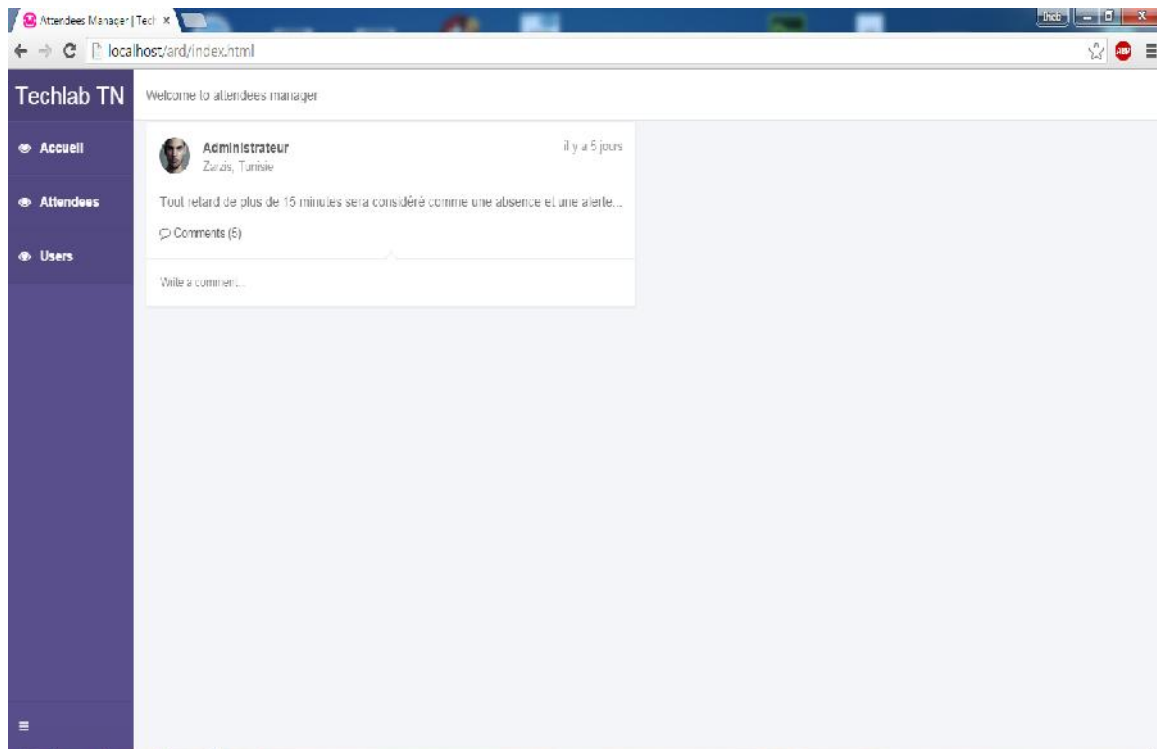


Figure 4.36 Page d'accueil

- Attendees (présences):

Cette page doit contenir un tableau détaillé des entrées/sorties d'employés : Le nom et prénom d'employé, la date d'entrée ou sortie, l'heure d'entrée ou sortie, l'état (la nature de pointage d'entrée ou sortie : IN/OUT).

The screenshot shows the 'Attendees Manager' application interface with the 'Attendees' tab selected. The table displays the following data:

Nom	Date	Heure	Etat
Nader Jerilla	2015-05-19	10:26:31	IN
Nader Jerilla	2015-05-19	10:26:47	OUT
Nader Jerilla	2015-05-19	10:21:26	IN
Nader Jerilla	2015-05-10	19:06:10	OUT
Nader Jerilla	2015-05-10	19:05:30	IN
Slaghiaro	2015-05-18	17:39:05	IN
Marwan Hourti	2015-05-18	17:38:27	IN
Marwan Hourti	2015-05-18	17:38:19	OUT
Marwan Hourti	2015-05-10	17:37:17	IN
Nader Jerilla	2015-05-11	15:44:01	OUT
Nader Jerilla	2015-05-11	15:43:46	IN
Nader Jerilla	2015-05-11	15:41:42	OUT
Nader Jerilla	2015-05-11	15:41:31	IN
Nader Jerilla	2015-05-11	15:36:46	OUT
Hamza Mihira	2015-05-11	15:36:31	IN
Nader Jerilla	2015-05-11	15:36:17	IN

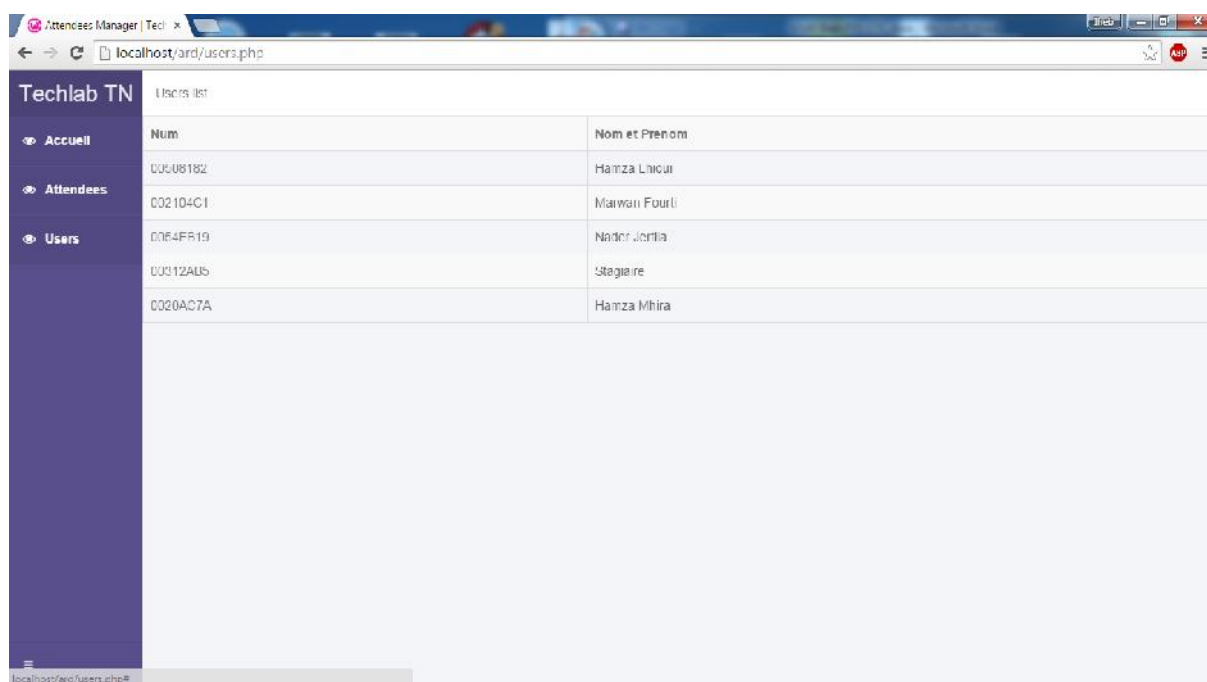
Figure 4.37 Page Attendees

Rapport de stage de fin d'études

- Users (employés) :

Cette page nous donne une idée sur les membres de la société, les employés et les personnels.

Nous pouvons ajouter une option sur cet tableau affiché puisqu'il est importé d'une base de données et n'est pas statique comme la suppression, la modification ou l'ajout d'un employé.



Num	Nom et Prenom
00008182	Hamza Lhacui
002104C1	Marwan Fourli
0064FB19	Nadim Jomla
00012AUS	Stagiaire
0020A07A	Hamza Mihira

Figure 4.38 Page Users

4.3.3. Conclusion

Au cours de ce chapitre, j'ai donné un récapitulatif sur la partie électronique et la partie informatique du sujet de mon stage de fin d'études, j'ai effectué une description toutes les étapes nécessaires à l'implémentation de mon système « Système de pointage » y compris la présentation de l'environnement logiciel, la programmation de la carte arduino, la réalisation de la carte électronique et de l'interface graphique afin d'aboutir à son fonctionnement. Comme j'ai mentionné, mon système offre au personnel de la société les fonctionnalités nécessaires pour contrôler les entrées/sorties d'employés.

Conclusion générale

Mon projet consiste à réaliser un système de pointage qui assure le contrôle de présence des employés d'une entreprise. Ce projet m'a été une source de découverte de plusieurs domaines rattachés à ma formation de base et une occasion précieuse pour approfondir mes connaissances. De plus, il m'a donné une idée sur la complémentarité entre la théorie et la pratique.

En fait, ces résultats ont nécessité en premier lieu la spécification des différentes tâches de mon sujet. En second lieu la conception de la partie matérielle et logicielle ainsi que la recherche des solutions algorithmiques et en troisième lieu la réalisation de la carte et la programmation du système finissant par création d'une interface graphique.

Ce projet a été intéressant et récapitulatif de la majorité des sciences étudiées et m'a permis de connaître plus la programmation de l'arduino, les différents types des cartes arduino, plusieurs bibliothèques dans la programmation arduino, les divers outils de communications (Série, Wifi, Bluetooth...) et les shields (boucliers) arduino.

En outre, dans le cadre de ce projet, j'ai eu une idée sur l'utilisation professionnelle de l'Eagle CadSoft ainsi que la réalisation physique d'une carte électronique.

Ce projet a fait l'objet d'une expérience intéressante, qui m'a permis d'améliorer mes connaissances et mes compétences dans le domaine de la programmation informatique. J'ai appris à mieux manipuler les langages PHP5, HTML5, MYSQL, JQUERY, JSON, AJAX et Java Script.

Malgré les difficultés que j'ai rencontrées tout au long de ce travail, j'espère que j'ai préparé les champs nécessaires à l'intégration de mon système et je suis prêt à recevoir toutes les critiques, les suggestions et les remarques tendant à améliorer d'avantage cette étude.

Bibliographie

- Démarrez avec Arduino – 2^e édition : principes de base et premiers montages, par Massimo Banzi.
- Le grand livre d'Arduino, d'Erik Bartmann.
- L'électronique en pratique, de Charles Platt.
- Make : Sensors, de Tero Karvinen, Kimmo Karvinen, Ville Valtokari et édité par Maker Media.
- Arduino – 2^e édition : Maîtrisez sa programmation et ses cartes d'interface (shields), de Christian Tavenier.
- Arduino : Applications avancées – claviers tactiles, télécommande par Internet, géolocalisation, applications sans fils, de Christian Tavernier.
- Arduino pour les Nuls, de John NUSSEY.
- Arduino : Les bases de la programmation, par Simon Monk.
- La boîte à outils Arduino – 105 techniques pour réussir vos projets, de Michael Margolis.
- Réalisez votre site web avec HTML5 et CSS3, par Mathieu Nebra.
- Premiers pas en CSS3 et HTML5, de Francis Draillard.
- Concevez votre site web avec PHP et MySQL – 2^e édition, de Mathieu Nebra.
- PHP 5.6 – Développez un site web dynamique et interactif, d'Olivier HEURTEL.
- PHP 5 & SQL, par Cyril Pierre de Geyer et Guillaume Ponçon.
- PFE : La billetterie Ticketlib, école nationale des sciences de l'informatique.
- PFE : Configurateur d'entrée pour Linux, école nationale des sciences de l'informatique.
- PFE : Smart House, Université virtuelle de Tunis.
- Rapport de stage : Robot autonome, Université Paris Sud.

Webographie

- www.arduino.cc
- Datasheet.com
- forum.arduino.cc
- lephpfacile.com
- easytutoriel.com
- mchobby.be
- tutorialspoint.com
- getbootstrap.com
- instructables.com
- raywenderlich.com
- startingelectronics.com
- areacom.altervista.org
- craym.eu
- learn.adafruit.com
- interactive-matter.eu
- tavendo.com
- starter-kit.nettigo.eu
- tronixstuff.com
- gotronic.fr
- Atexa-electronique.com
- Developpez.net
- Electroschema.info
- Electromatic.be
- futura-sciences.com
- Jmb-electronique.com
- Labcenter.co.uk

Résumé

Dans le cadre d'obtention de diplôme en Licence Appliquées en Sciences et Technologie de l'Information et de la Communication, j'ai eu l'occasion de réaliser un stage de fin d'études durant trois mois au sein de l'entreprise Techlab TN.

L'objectif principal de ce travail de PFE consiste à implémenter un système de pointage pour la gestion de présence des employés d'une entreprise à base des cartes et des badges utilisant la radio-identification (RFID).

Ce système se représente à la porte de l'entreprise, pour faciliter aux employés de se pointer à chaque entrée et sortie. Les données enregistrées seront envoyées vers un serveur local ou se fait la gestion de présence.

Mots clé : pointage, arduino, Wifi, RFID, NFC.