

Projet technologique



ECE PARIS • LYON
ÉCOLE D'INGÉNIEURS

Thomas DUMENY

Leandro MAOUCHI

Benjamin NANCY

ING4 App SE

2018/2021

Table des matières

I. Conception	3
1) Analyse fonctionnelle.....	3
2) Développement matériel (hardware).....	5
a) Montage de communication USB de l'ATmega328P.....	5
b) Montage de communication SPI avec le module RFID, buzzer et LED	6
c) Montage pour la communication I2C entre l'ATmega 328P et l'écran LCD.....	8
d) Intégration logicielle/matérielle.....	9
3) Développement logiciel et intégration	9
II. Organisation de l'équipe.....	10
Conclusion.....	11

I. Conception

1) Analyse fonctionnelle

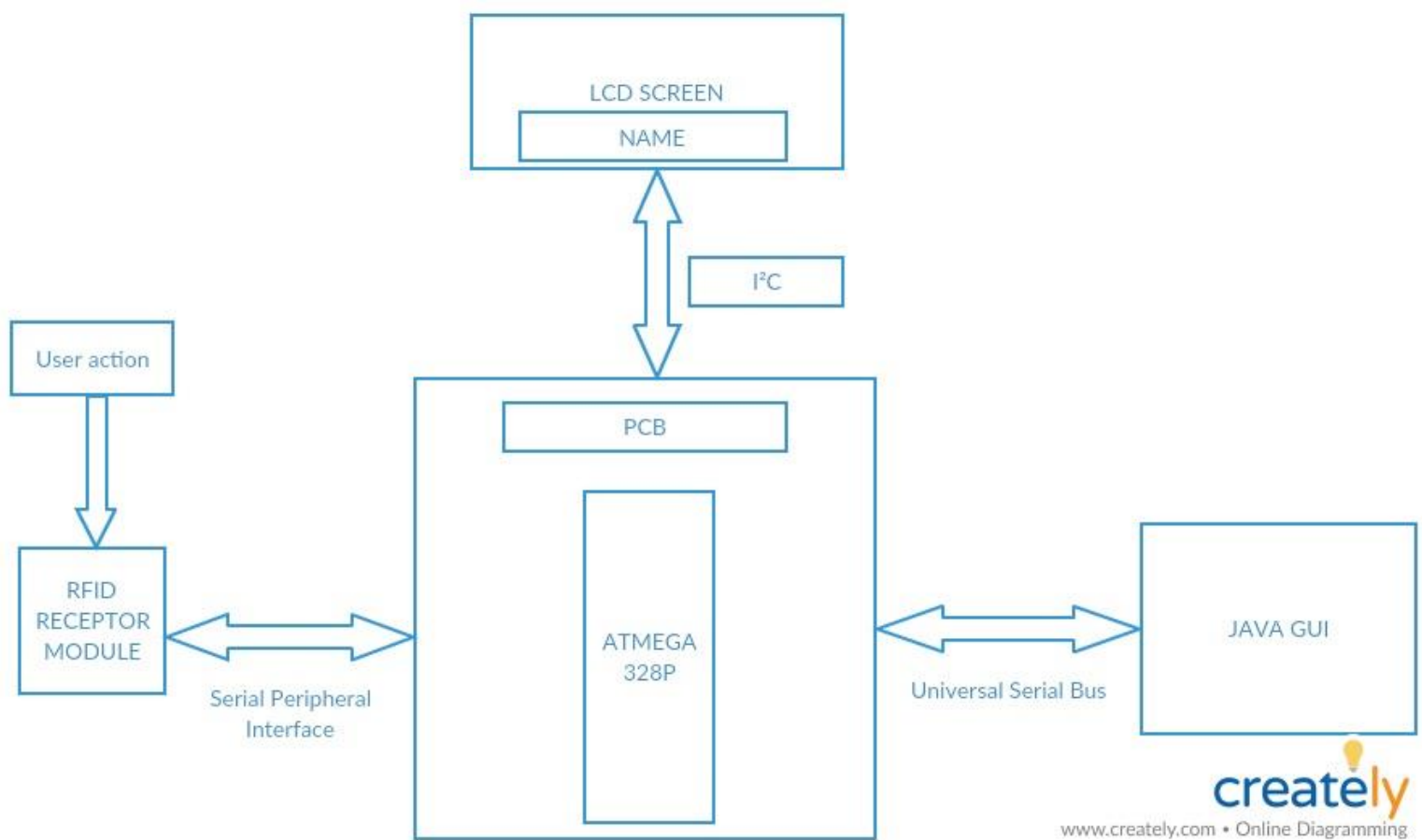


Figure 1 : Diagramme fonctionnel (SADT)

Utilisation d'un module RFID (RC522)	Lire les IDs des étudiants grâce à leur carte étudiant ou tag .
Utilisation d'un écran LCD	Afficher les noms des étudiants
Utilisation d'un buzzer et d'une LED blanche	Signaler que la carte a bien été lue
Utilisation d'un adaptateur USB	Envoyer l'UID vers l'interface JAVA Envoyer le Nom et Prénom vers le microcontrôleur ATmega328P Flasher l'ATmega328P avec notre code
Création d'une interface JAVA et d'une base de données d'étudiants	Récupérer les UIDs via USB ; Charger base de données avec étudiants et UID correspondants. Afficher la liste de la classe et si les étudiants assistent au cours.
Utilisation d'un microcontrôleur Atmega328P	Gestion générale de l'environnement de développement Arduino

Figure 2 : Tableau des attentes techniques du projet

Le système doit permettre à un étudiant de badger avec sa carte étudiante grâce à un module RFID (RC522), qui va ensuite permettre la transmission (SPI) de l'ID de l'étudiant à un ATmega328P. Puis l'ID est envoyé à une interface (JAVA) basée sur ordinateur via un adaptateur USB. L'ID permet ensuite, grâce à une base de données, d'identifier l'étudiant par son nom. Cette donnée est ensuite envoyée toujours en communication série USB vers l'ATmega328P. L'affichage du nom de l'étudiant se fait ensuite sur un écran LCD via I2C.

Les tests fonctionnels seront donc faits dans cet ordre :

- 1) Vérification du fonctionnement du module RFID avec l'ATmega328P (communication SPI), récupération de l'ID de l'étudiant ;
- 2) Vérification du fonctionnement de la communication ATmega328P/IHM. Envoie de l'ID de l'étudiant, fonctionnement de la base de données et de l'affichage à l'écran, envoi du nom complet de l'étudiant vers l'ATmega328P ;

- 3) Vérification de l'affichage du nom de l'étudiant sur l'écran LCD via communication I2C avec l'ATmega328P.

2) Développement matériel (hardware)

a) Montage de communication USB de l'ATmega328P

2 condensateurs 22pF	Stabilisation du quartz
Quartz 16MHz	Fréquence de référence pour l'ATmega328P
ATmega328P	Microcontrôleur
Résistance 10kOhms	Résistance pull down bouton poussoir
Bouton poussoir	Fonction reset de notre microcontrôleur
Adaptateur USB TTL	Communication USB (Serial port) entre ATmega328P et ordinateur
Alimentation 5V	Alimentation de l'ATmega328P

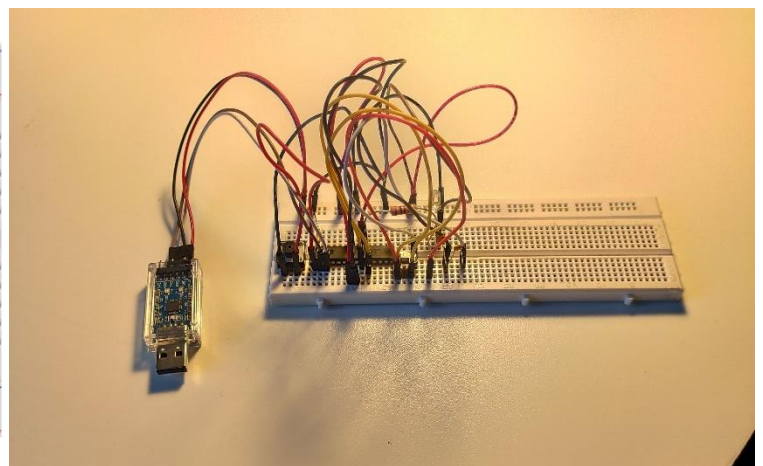
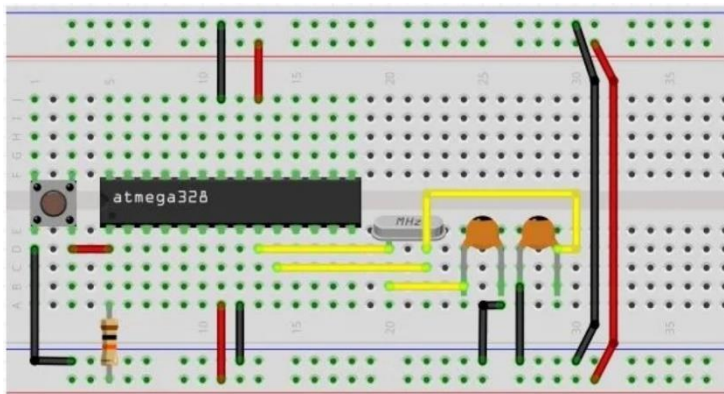


Figure 3 : Schéma et photo du montage de flashage de l'ATmega 328P

b) Montage de communication SPI avec le module RFID, buzzer et LED

Alimentation 3.3V	Alimentation du module RFID (RC522)
Module RFID (RC522)	Lecture de l'ID de l'étudiant
ATmega328P	Récupération de l'information
LED	Témoin de lecture du badge
Buzzer	Témoin de lecture du badge
Résistance 100 Ohms	Limitation de courant pour le buzzer et la LED

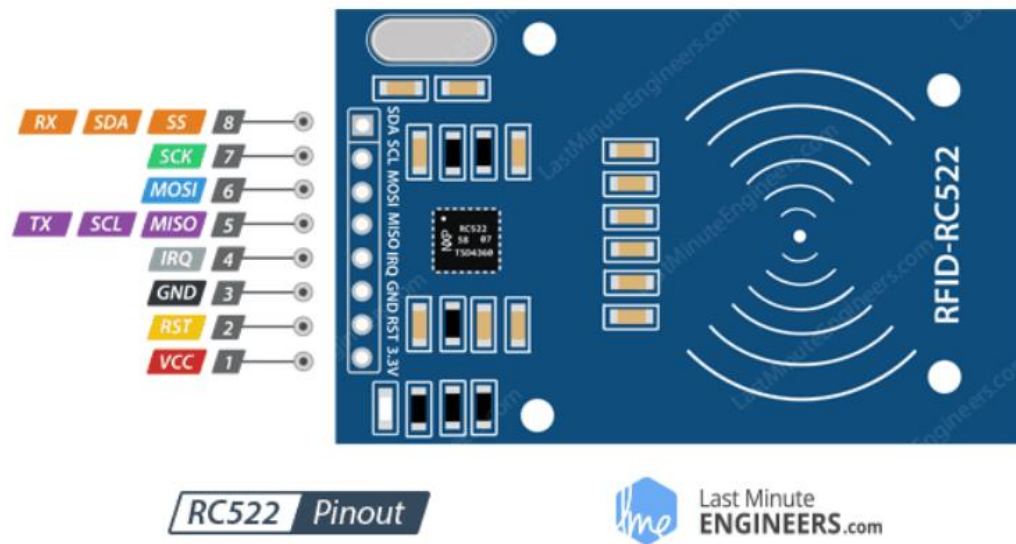


Figure 4 : Schéma des entrée/sorties du module RFID RC522

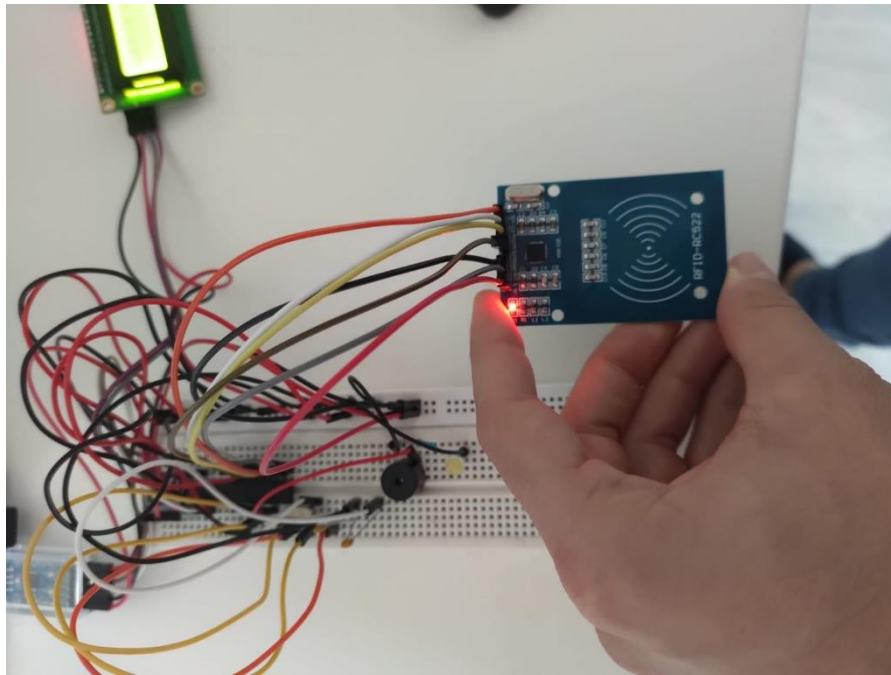
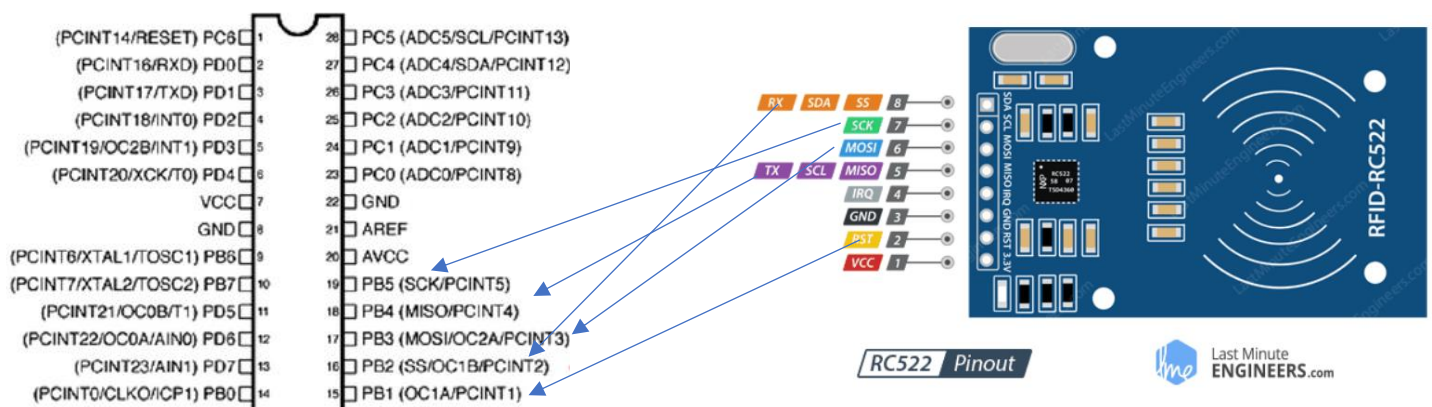


Figure 5 : Photo du montage RFID/microcontrôleur

Le module RFID RC522 est relié de la façon suivante avec l'ATmega328P :



c) Montage pour la communication I2C entre l'ATmega 328P et l'écran LCD

ATmega 328P	Microcontrôleur
Ecran LCD	Affichage du nom de l'étudiant
Adaptateur I2C	Communication I2C entre l'ATmega 328P et l'écran LCD

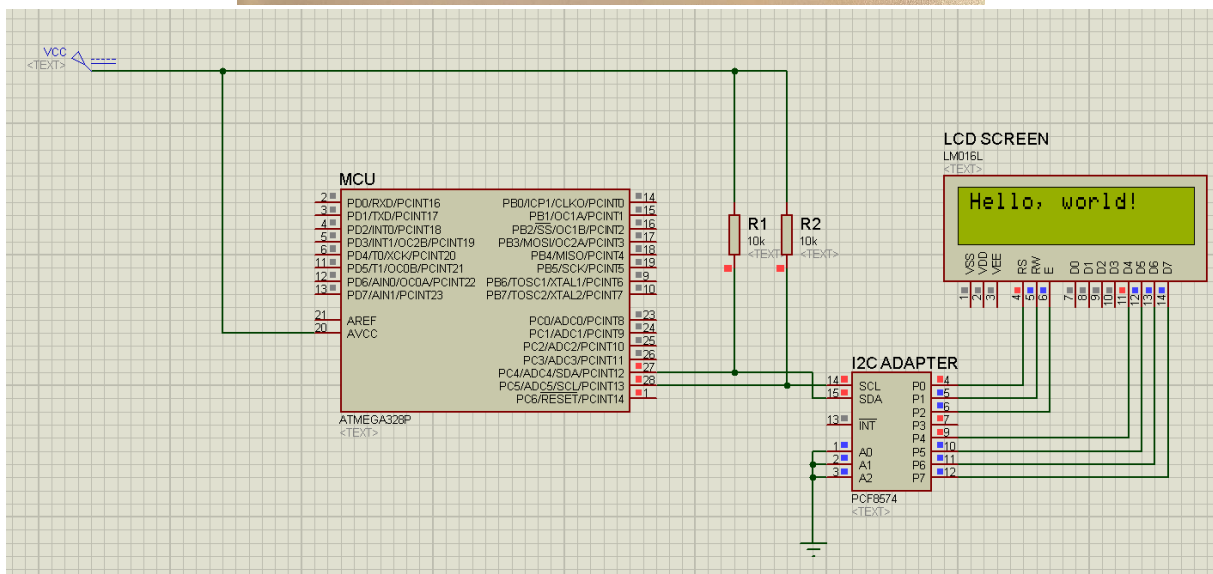
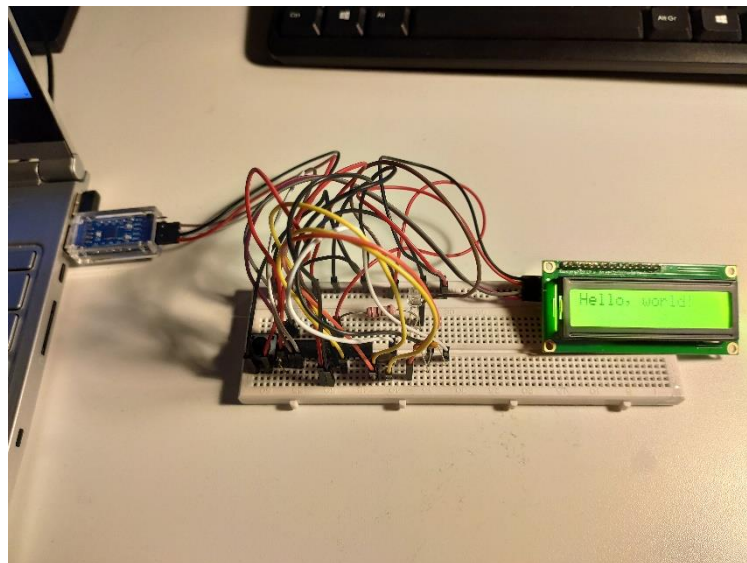


Figure 7 : Câblage et simulation LCD en I2C

Note : Nous avons dû changer le câblage des pins SDA et SCL sur l'ATmega328P pour libérer les pins nécessaires à la connexion des entrées sorties du RC522 (voir dépôt git pour les codes de scan des adresses I2C et pour le code des tests unitaires).

d) Intégration logicielle/matérielle

Nous avons intégré le système de matériel et logiciel en même temps. En effet, comme nous le détaillerons plus loin dans le rapport, l'intégration s'est faite avec la communication complète de notre microcontrôleur avec l'interface en JAVA Processing, l'écran LCD en I2C, et le module RC522 en SPI. Une fois toutes les pins assignées et connectées à notre module ATmega328P, il ne nous restait plus qu'une étape d'intégration logicielle, et par la suite, une validation fonctionnelle du système complet.

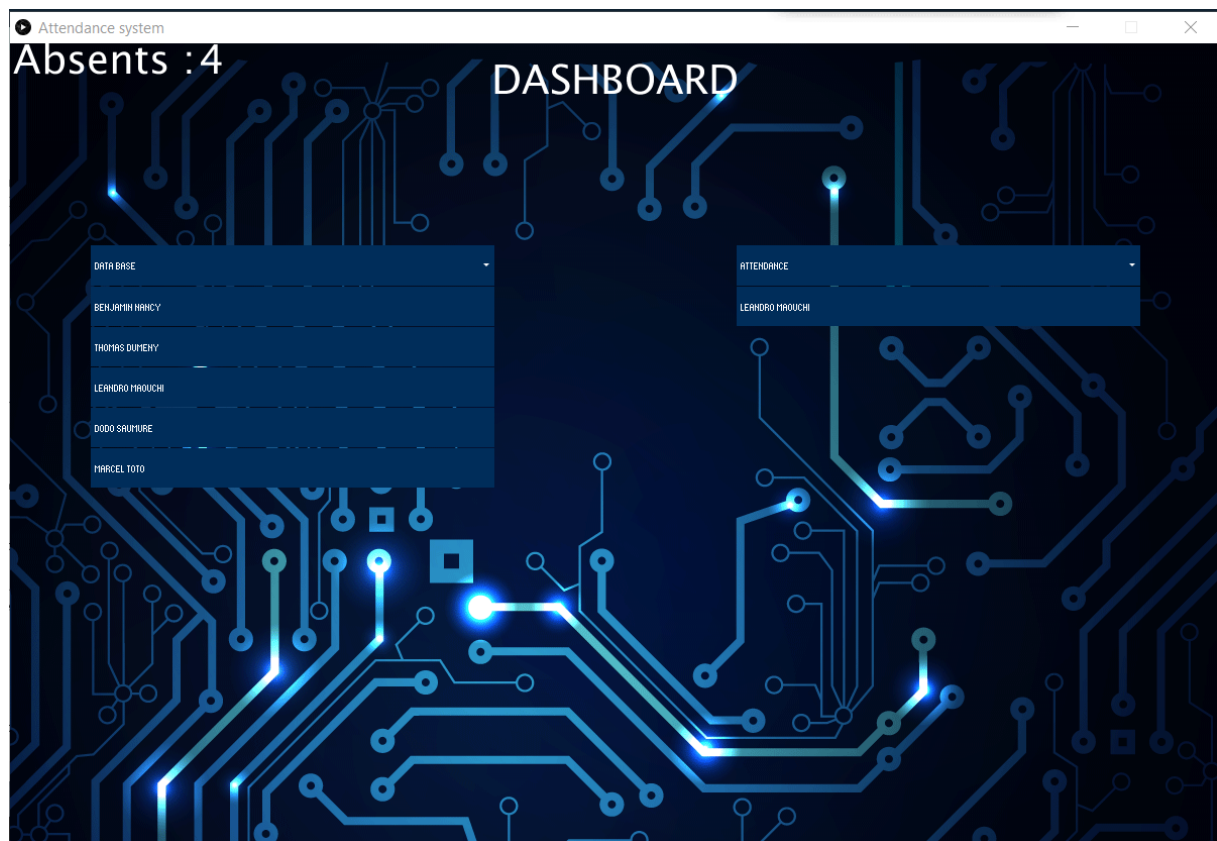
3) Développement logiciel et intégration

Une fois les tests unitaires au niveau matériels et logiciels unitaires validés, pour permettre l'intégration ainsi la communication complète de tous les éléments de notre projet nous avons dû développer du logiciel. Ce développement logiciel s'est fait en trois parties. La première a été de lire un UID d'un étudiant avec le module RC522 et de le transmettre dans le port série (via TTL).

La deuxième partie, celle qui fût la plus verbeuse en code, a consisté en la création d'une base de données avec des étudiants ainsi qu'en l'implémentation d'un algorithme de reconnaissance des UIDs dans cette base de données. Aussi, nous avons développé une interface graphique à l'aide de la librairie ControlP5. Cette deuxième partie est la seule partie à être compilée et exécutée par notre ordinateur (l'ATmega328P n'est pas fait pour stocker une base de données d'étudiants). C'est aussi la seule partie développée en JAVA.

L'interface graphique affiche les étudiants censés être présents dans la classe (chargement de la base de données) et au fur et à mesure des badgeages, afficher une liste des étudiants présents réellement. Pour terminer, un compteur d'absents se met automatiquement à jour avec les badgeages.

Voici à quoi ressemble l'interface graphique en JAVA :



La troisième partie logicielle a été, après envoi des noms et prénoms par la partie java dans le port série, de récupérer ces derniers côté microcontrôleur et d'envoyer ces informations en I2C à l'écran LCD. Cette dernière partie, tout comme la première a été développée en C/C++ à l'aide de l'IDE Arduino.

II. Organisation de l'équipe

Nous avons décidé pour ce projet d'utiliser Github et de se partager les tâches au fur et à mesure du projet. Nous avons divisé le projet en plusieurs parties que nous avons réalisées dans l'ordre suivant :

- Réalisation du montage de base de l'ATmega 328P ;

- Tests unitaires de chacun des blocs (LCD, RC522) ;
 - Communication SPI et badgeage avec le module RFID RC522.
 - Communication I2C avec l'écran LCD.
- Intégration avec communication USB, interface JAVA, base de données, envoi du nom de l'étudiant vers l'ATmega328P et affichage complet sur LCD ;
- Validation fonctionnelle avec le prototypage ;

Conclusion

Ce projet a été très enrichissant pour tous les membres de notre équipe, nous avons dû faire appel à compétences fondamentales de l'ingénieur en systèmes embarqués, à savoir :

- Une bonne compréhension de l'électronique, des composants, de leur fonctionnement et de leurs interactions ;
- La maîtrise du langage JAVA, du langage C et de leur implémentation dans le système ;
- Une bonne compréhension des différents protocoles de communication utilisés ;
- Un bon relationnel pour travailler en équipe sereinement ;

Figure 1 : Diagramme fonctionnel (SADT)	3
Figure 2 : Tableau des attentes techniques du projet	4
Figure 3 : Schéma et photo du montage de flashage de l'ATmega 328P	5
Figure 4 : Schéma des entrée/sorties du module RFID RC522	6
Figure 5 : Photo du montage RFID/microcontrôleur	7
Figure 6 : Photo du montage avec l'écran LCD	8
Figure 7 : Câblage et simulation LCD en I2C	8

Annexe :

Voir notre dépôt github pour le code source, les schémas et les ressources :

<https://github.com/lm173618/Attendance-system>