

République Tunisienne

Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique
Université de Tunis

Ecole Nationale Supérieure d'Ingénieurs de
Tunis



Département Informatique

Internet des objets (IOT)

Troisième année en Génie Informatique (NTS)

Rapport du Projet : Smart School

Présenté le 16/12/2024

Par

Benhamed Ranime

Cherni Rihab

Sidi Mohamed Ahmed

Encadré par : Monsieur Faleh Yassine

Année universitaire : 2024-2025

Table des matières

Introduction générale	1
Chapitre 1: Cadre Général du Projet	2
Introduction	2
1. Contexte du Projet	2
2. Défis	2
3. Problématique	3
4. Solution Proposée	4
5. Objectifs du Projet	4
Conclusion	5
Chapitre 2: Étude technique	6
Introduction	6
1. Technologies et environnements de développement utilisés	6
2. Composants Matériels Utilisés	8
2.1. Smart Class	8
2.2. Smart Parking	11
Conclusion	13
Chapitre 3: Réalisation et mise en œuvre	14
Introduction	14
1. Circuit réalisé	14
1.1. Schéma électronique : Smart Class	15
1.2. Schéma électronique : Smart Parking	15
2. Architecture du Système Smart School	16
3. Interfaces Utilisateurs	17
3.1. Interface de Chargement	17
3.2. Interface d'Inscription	18

3.3.	Interface de Connexion	18
3.4.	Interface d'Accueil	19
3.5.	Interface de Gestion de Parking	19
3.6.	Interface de Gestion de Classe Intelligente	19
	Conclusion	20
	Conclusion générale	21
	Bibliographie	22

Table des figures

2.1	Logo Arduino IDE 2.3.3[1]	6
2.2	Logo Circuito.io [2]	7
2.3	Logo Flutter[3]	7
2.4	Logo Firebase[4]	7
2.5	Logo Visual Studio Code[5]	8
2.6	Carte Arduino UNO	8
2.7	LEDs	9
2.8	Buzzer	9
2.9	Capteur d'Humidité et Température DHT11	10
2.10	Afficheur LCD	10
2.11	RFID	10
2.12	Module Wi-Fi	11
2.13	Carte Arduino UNO	11
2.14	Capteur PIR	12
2.15	LEDs	12
2.16	Afficheur LCD	12
2.17	Servo Moteur	13
2.18	Module Wi-Fi	13
3.1	Schéma réel du circuit électronique réalisé pour le système	14
3.2	Schéma électronique : Smart Class	15
3.3	Schéma électronique : Smart Parking	16
3.4	Architecture du Système Smart School	17
3.5	Interfaces de chargement et d'authentification	18
3.6	Interfaces d'accueil et de gestion de parking	19
3.7	Interfaces gestion de classe intelligente	20

Introduction générale

Le monde traverse une phase d'évolution technologique rapide, caractérisée par l'essor des objets connectés (IoT), qui révolutionnent les infrastructures traditionnelles dans divers domaines. Ces technologies facilitent une communication fluide entre dispositifs et permettent une collecte de données en temps réel, optimisant ainsi les processus et renforçant l'efficacité opérationnelle. Dans le secteur de l'éducation, l'IoT ouvre des perspectives novatrices, transformant les établissements scolaires en environnements modernes et intelligents. Ces outils connectés optimisent la gestion des ressources, améliorent l'organisation des classes et rehaussent la qualité des services proposés.

Dans ce contexte d'évolution constante, l'intégration de l'IoT dans les infrastructures quotidiennes représente un enjeu stratégique pour améliorer la qualité de vie et accroître l'efficacité des services. Le projet Smart School s'inscrit dans cette dynamique en proposant une solution innovante visant à métamorphoser une école traditionnelle en un espace connecté et optimisé. Ce projet repose sur 2 axes principaux : la classe intelligente et la gestion du parking. Son objectif est d'intégrer des technologies IoT pour créer un environnement scolaire confortable. Grâce à des capteurs et des interfaces de gestion en temps réel, il répond aux besoins variés des utilisateurs : étudiants, enseignants, administrateurs et personnels techniques.

Le projet s'articule autour de plusieurs objectifs clés :

- Assurer une gestion automatisée des éléments environnementaux tels que la température et l'éclairage dans les salles de classe.
- Renforcer la sécurité des infrastructures scolaires grâce à des systèmes d'alerte et de surveillance avancés.
- Optimiser l'utilisation des espaces de stationnement grâce à une gestion intelligente des parkings.

Ce rapport explore en détail le contexte, les objectifs ainsi que les étapes de conception, de développement et de mise en œuvre du projet. Il fournit également une description du matériel employé, des schémas de câblage et des illustrations pour clarifier la réalisation des solutions. Le projet Smart School IoT constitue une avancée majeure dans la modernisation des établissements scolaires, ouvrant de nouvelles perspectives pour l'éducation de demain.

Cadre Général du Projet

Introduction

Ce chapitre présente le cadre général du projet Smart School IoT. Il décrit le contexte, les objectifs, les enjeux, et l'importance de ce projet dans l'éducation moderne, tout en mettant en avant l'intégration des technologies IoT.

1. Contexte du Projet

Le secteur éducatif fait face à des défis importants, notamment la gestion des infrastructures, la sécurité des élèves, et l'optimisation des ressources. Les technologies IoT offrent des solutions innovantes pour transformer les écoles en espaces modernes, connectés et plus efficaces. Grâce à ces technologies, il devient possible d'améliorer la gestion des établissements, la sécurité des élèves, la durabilité des ressources et l'environnement d'apprentissage.

2. Défis

Les écoles traditionnelles font face à de nombreux défis opérationnels qui impactent la qualité de l'enseignement et le bien-être des élèves, des enseignants et du personnel. Ces défis se regroupent en deux principaux axes :

Gestion des Salles de Classe :

- **Températures inadaptées et humidité non contrôlée :**
 - Les températures extrêmes et une humidité mal régulée entraînent un inconfort, réduisant la concentration des élèves.
 - Ces conditions peuvent causer des problèmes de santé (allergies, moisissures) et dégrader le matériel scolaire.
- **Éclairage insuffisant ou excessif :**

- Un éclairage trop faible fatigue les yeux et réduit la productivité.
- Un éclairage trop intense peut provoquer des éblouissements, gênant ainsi l'apprentissage.
- **Manque d'outils pour surveiller la présence des élèves :**
 - Le suivi manuel des présences est imprécis et chronophage.
 - L'absence de solutions modernes complique la gestion des absences et entrave la détection précoce des risques d'abandon scolaire.

Gestion des Parkings

- **Difficulté à localiser des places libres :**
 - L'absence de signalisation ou de systèmes connectés entraîne des congestions et une perte de temps pour les utilisateurs.
- **Accès non autorisé au parking :**
 - Les véhicules non identifiés ou extérieurs à l'établissement réduisent l'espace disponible pour les étudiants, les enseignants et le personnel.
 - Ces intrusions posent également des risques de sécurité.
- **Manque d'infrastructures connectées :**
 - L'absence de solutions numériques modernes complique la gestion efficace des parkings et la prévention des problèmes liés à leur utilisation.

Face à ces défis, il devient essentiel d'adopter des solutions connectées et intelligentes pour moderniser les infrastructures scolaires et répondre aux besoins des établissements du XXI^e siècle.

3. Problématique

L'adoption des technologies IoT dans le secteur éducatif répond à plusieurs problématiques :

- 1- **Comment automatiser la gestion des salles de classe pour améliorer le confort et l'efficacité énergétique ?**
- 2- **Comment renforcer la sécurité des infrastructures scolaires par des systèmes de surveillance intelligents et des alertes en temps réel ?**
- 3- **Comment optimiser la gestion des parkings scolaires pour réduire le temps perdu dans la recherche de places ?**

4. Solution Proposée

Le projet **Smart School** vise à transformer les écoles traditionnelles en espaces intelligents grâce à l'intégration des technologies IoT. Il se compose de deux sous-systèmes principaux :

- **Système de Classe Intelligente :**

- Surveillance des conditions environnementales telles que la température, l'humidité et la luminosité.
- Automatisation de la régulation de la température et de l'éclairage pour maintenir un confort optimal tout en réduisant la consommation énergétique. Une alarme se déclenche lorsque les seuils prédéfinis sont dépassés, activant automatiquement la climatisation ou le chauffage selon le besoin.
- Suivi automatisé et en temps réel de la présence des élèves dans les salles de classe.
- Interface mobile permettant aux administrateurs de contrôler et de surveiller ces paramètres en temps réel.

- **Système de Parking Intelligent :**

- Détection automatique des places disponibles grâce à des capteurs, réduisant ainsi la congestion et le temps perdu à chercher une place.
- Accès sécurisé au parking, limitant l'entrée aux personnes autorisées (étudiants et personnel) et empêchant les intrusions de non-membres de l'établissement.
- Application mobile affichant en temps réel le nombre de places libres.

Ces sous-systèmes sont centralisés sur une plateforme unique permettant de collecter les données en temps réel. La plateforme envoie des alertes en cas de besoin et contribue à :

- Améliorer l'efficacité énergétique grâce à la gestion intelligente des infrastructures.
- Optimiser le confort et la productivité au sein des écoles.
- Faciliter la gestion des infrastructures scolaires via une interface mobile conviviale.

5. Objectifs du Projet

Les objectifs principaux du projet Smart School IoT se concentrent sur deux domaines : la classe intelligente et la gestion du parking intelligent. Les objectifs sont les suivants :

- Optimiser les conditions environnementales des salles de classe grâce à une gestion automatisée de l'éclairage et de la température.
- Renforcer la sécurité des élèves et du personnel par des systèmes de surveillance et d'alerte.

- Améliorer la gestion des parkings scolaires à travers des technologies IoT pour faciliter l'accès.
- Réduire la consommation énergétique et favoriser une gestion plus responsable des ressources.

Conclusion

Ce chapitre a présenté le cadre du projet Smart School IoT, en détaillant le contexte, les objectifs, les avantages et les problématiques associées. En utilisant les technologies IoT, ce projet vise à moderniser l'infrastructure scolaire et à relever les défis contemporains liés à la gestion des ressources et à la sécurité. Les chapitres suivants détailleront la conception et la mise en œuvre des solutions proposées pour atteindre ces objectifs.

Étude technique

Introduction

Dans ce chapitre, nous examinerons les technologies et les composants matériels employés dans la conception du projet Smart School, en mettant en lumière leur rôle dans la concrétisation de la solution proposée.

1. Technologies et environnements de développement utilisés

- 1- **Arduino IDE 2.3.3** : est une plateforme de développement intégrée utilisée pour programmer les microcontrôleurs Arduino. La version 2.3.3 offre une interface utilisateur améliorée et des fonctionnalités modernes comme la complétion automatique de code, une gestion des bibliothèques simplifiée et un débogueur intégré [1].

Elle a été utilisée dans ce projet pour :

- Écrire et déployer des programmes sur les dispositifs matériels connectés.
- Gérer les capteurs et les actionneurs intégrés dans le système Smart School.



Figure 2.1 – Logo Arduino IDE 2.3.3[1]

- 2- **Circuito.io** est un outil en ligne de conception et de simulation de circuits électroniques. Il permet de générer automatiquement des schémas électroniques et des listes de composants, facilitant ainsi le prototypage rapide des projets [2].

Son utilisation dans ce projet inclut :

- La conception et la simulation des circuits électroniques du système Smart School.

- La génération des connexions et des guides de câblage pour garantir une réalisation précise.



Figure 2.2 – Logo Circuito.io [2]

- 3- **Flutter** : est un framework open-source développé par Google, permettant de créer des applications multiplateformes avec une interface utilisateur riche. Il est un kit de développement logiciel (SDK) qui permet de produire des applications nativement compilées pour plusieurs plateformes (mobile, web, bureau) à partir d'une seule base de code [3]. Son adoption dans ce projet se justifie par :

- La création d'une application mobile pour gérer et surveiller les systèmes Smart School.
- La compatibilité multiplateforme (iOS, Android) et la rapidité de développement.
- Une interface utilisateur moderne et réactive.



Figure 2.3 – Logo Flutter[3]

- 4- **Firebase** : est une plateforme de développement d'applications mobiles et web développée par Google. Elle fournit aux développeurs une variété de services et d'outils pour les aider à développer des applications de haute qualité, à améliorer leur base d'utilisateurs et à monétiser leurs applications. Firebase offre des fonctionnalités allant de l'hébergement et du stockage de données en temps réel à l'authentification des utilisateurs, en passant par des outils d'analyse et de reporting [4].

Elle a été utilisée dans ce projet pour :

- La gestion de la base de données en temps réel, permettant de collecter, stocker et analyser les données des capteurs.
- La gestion des authentifications des utilisateurs et des autorisations d'accès.
- L'intégration de services pour assurer la synchronisation des données.



Figure 2.4 – Logo Firebase[4]

- 5- **Visual Studio Code (VSCode)** : est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-platform, c'est-à-dire qu'il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web,

mais il prend en charge de nombreux autres langages de programmation tels que C++, Python, Java, etc. Il offre de nombreuses fonctionnalités comme la coloration syntaxique, l'auto-complétion, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l'intégration avec Git, et beaucoup d'autres[5].

Il a été utilisé comme environnement de développement pour :

- Écrire et déboguer les scripts de programmation liés aux différents composants du projet.
- Intégrer les outils et extensions nécessaires pour Flutter et Firebase.

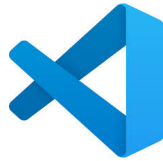


Figure 2.5 – Logo Visual Studio Code[5]

L'utilisation de ces technologies et environnements de développement a permis d'assurer une conception efficace, une simulation réaliste et une gestion optimale des systèmes du projet.

2. Composants Matériels Utilisés

2.1. Smart Class

1- **Carte Arduino UNO** : est un microcontrôleur de la plateforme Arduino, largement utilisé dans les projets IoT et électroniques. Il est au cœur du système Smart Class, agissant comme le cerveau du dispositif. Il permet :

- La gestion des entrées et sorties numériques et analogiques, permettant ainsi l'interaction avec divers capteurs et dispositifs.
- Le contrôle et la communication entre les différents composants matériels du projet.
- La programmation flexible et l'intégration avec des capteurs et des actionneurs.



Figure 2.6 – Carte Arduino UNO

2- **Les LEDs** (diodes électroluminescentes) sont utilisées dans Smart Class pour signaler différents états ou alertes et aussi **pour remplacer un climatiseur, un chauffage et**

la **lampe d'éclairage**, nous avons utilisé trois LED en raison de contraintes de disponibilité de matériel et de mise en œuvre. Les LED, associées à des résistances adaptées, peuvent simuler le fonctionnement de composants plus complexes tels que la climatisation et le chauffage dans un contexte de prototypage. Cette solution de substitution nous a permis de continuer à tester et à développer notre projet tout en reproduisant de manière simplifiée les effets attendus des composants d'origine. Elles permettent de :

- Indiquer visuellement l'état des systèmes (led rouge pour un problème détecté).
- Créer des notifications lumineuses pour alerter les administrateurs ou les utilisateurs des changements dans les conditions de la salle de classe.
- LED bleue pour simuler un ventilateur, LED violette pour simuler un chauffage et LED verte pour simuler l'éclairage.

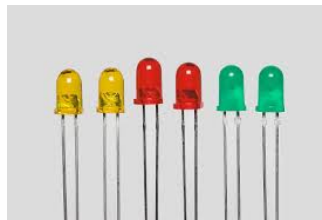


Figure 2.7 – LEDs

3- **Buzzer** : est un composant sonore utilisé dans le projet pour générer des alertes audibles.

Il permet :

- D'alerter les utilisateurs en cas de dépassement de seuils critiques, comme une température ou une humidité trop élevée.
- De signaler l'arrivée ou la fin d'un événement dans le système (par exemple, la détection de présence ou l'activation d'un mode particulier).



Figure 2.8 – Buzzer

4- **Capteur d'Humidité et Température DHT11** : est un capteur de température et d'humidité qui mesure les conditions ambiantes dans la salle de classe. Ce capteur permet :

- De surveiller en temps réel la température et l'humidité dans l'environnement de la classe.

- De garantir des conditions optimales de confort et d'apprentissage en ajustant automatiquement les systèmes de climatisation ou de chauffage selon les données collectées.

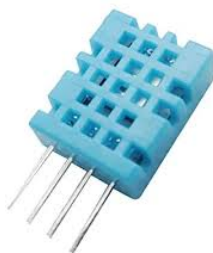


Figure 2.9 – Capteur d'Humidité et Température DHT11

- 5- **Afficheur LCD** : est utilisé pour afficher des informations essentielles aux utilisateurs (enseignants, administrateurs) en temps réel. Il permet :
- D'afficher les valeurs actuelles de température et d'humidité dans la salle.
 - De fournir des notifications ou des alertes visuelles concernant l'état des systèmes de la salle (par exemple, "Température trop élevée", "Présence détectée").



Figure 2.10 – Afficheur LCD

- 6- **RFID (Radio Frequency Identification)** : est un système d'identification par radio-fréquence qui permet de suivre la présence des élèves dans la salle de classe. Il est utilisé dans Smart Class pour :
- Enregistrer automatiquement la présence des élèves lorsqu'ils entrent dans la salle en scannant leur carte RFID.
 - Faciliter le suivi des absences, permettant une collecte de données rapide et précise.



Figure 2.11 – RFID

- 7- **Module Wi-Fi** : joue un rôle central dans la connectivité du système. Il permet :

- De transmettre les données collectées par les capteurs vers Firebase, une plateforme cloud, pour leur stockage et traitement en temps réel.
- De faciliter la communication entre les composants matériels et l'application mobile, permettant ainsi une surveillance et un contrôle à distance.

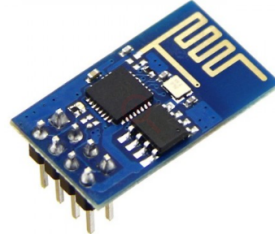


Figure 2.12 – Module Wi-Fi

Les composants matériels collaborent de manière harmonieuse pour gérer l'environnement de la salle, surveiller les conditions ambiantes, et suivre la présence des élèves. Ensemble, ils constituent une solution connectée qui optimise le confort et facilite la gestion de la salle de classe.

2.2. Smart Parking

- 1- **Arduino UNO** est également au cœur du système Smart Parking, servant de micro-contrôleur pour gérer et coordonner les différents composants matériels. Il permet de :
 - Lire les données des capteurs et actionner les dispositifs associés, tels que les LEDs et les moteurs.
 - Assurer la communication entre les capteurs de détection de place et les affichages d'état du parking.
 - Exécuter les algorithmes de gestion du parking, tels que l'optimisation de l'occupation des places.



Figure 2.13 – Carte Arduino UNO

- 2- **Capteur PIR (Passive Infrared Sensor)** : est utilisé pour détecter la présence de véhicules dans les places de parking. Il capte les variations de chaleur dues à la présence humaine ou de véhicules. Ce capteur permet de :
 - Identifier en temps réel si une place est occupée ou libre, en détectant les mouvements ou la présence d'un véhicule.

- Réduire la congestion dans le parking en fournissant des informations précises et instantanées sur l'occupation des places.



Figure 2.14 – Capteur PIR

- 3- **LEDs** : jouent un rôle crucial dans le système Smart Parking pour indiquer visuellement l'état des places de parking. Elles permettent de :
- Signaler une place libre en vert ou occupée en rouge.
 - Aider les conducteurs à localiser facilement les places disponibles, réduisant ainsi le temps de recherche d'une place libre et la congestion du parking.



Figure 2.15 – LEDs

- 4- **Afficheur LCD** : permet de fournir des informations complémentaires aux utilisateurs et gestionnaires du parking. Il sert à :
- Afficher l'état général du parking.
 - Afficher des messages d'alerte ou des informations utiles sur le fonctionnement du parking, comme "Capacité maximale atteinte"...



Figure 2.16 – Afficheur LCD

- 5- **Servo Moteur** : est utilisé pour ajuster mécaniquement les barrières ou les accès du parking. Ce moteur permet de ouvrir ou fermer des barrières à l'entrée et à la sortie du parking en fonction des places disponibles et de l'autorisation d'accès.



Figure 2.17 – Servo Moteur

6- **Module Wi-Fi** : joue un rôle central dans la connectivité du système Smart Parking. Il permet :

- De transmettre les données collectées par les capteurs, comme la disponibilité des places de stationnement, vers Firebase, une plateforme cloud, pour leur stockage et traitement en temps réel.
- De faciliter la communication entre les composants matériels et l'application mobile, permettant ainsi aux utilisateurs de vérifier la disponibilité des places et de gérer leurs réservations à distance.

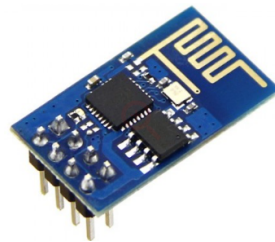


Figure 2.18 – Module Wi-Fi

Les composants matériels utilisés dans le système Smart Parking travaillent ensemble pour fournir une solution efficace et intelligente pour la gestion du parking. Le système facilite la détection des places disponibles, améliore l'expérience utilisateur et optimise l'utilisation de l'espace de stationnement, tout en offrant un contrôle automatique de l'accès.

Conclusion

En conclusion, ce chapitre a permis de présenter les technologies et composants matériels essentiels pour la conception de notre système. Dans le prochain chapitre, nous détaillerons la réalisation et la mise en œuvre de ces technologies dans le cadre de notre projet.

Réalisation et mise en œuvre

Introduction

Dans ce chapitre, nous détaillerons les étapes de réalisation du système, incluant le circuit électronique, qui constitue le cœur de la solution, ainsi que les interfaces de l'application mobile. Ces dernières permettent de suivre en temps réel les données environnementales et de recevoir des alertes de manière intuitive.

1. Circuit réalisé

La figure 3.1 illustre le circuit électronique tel qu'il a été construit dans la pratique. Il comprend les différents composants matériels connectés pour garantir le bon fonctionnement du système. Ce schéma met en évidence les connexions entre les capteurs, actionneurs, et le microcontrôleur utilisé, donnant une vue d'ensemble précise de la structure électronique.

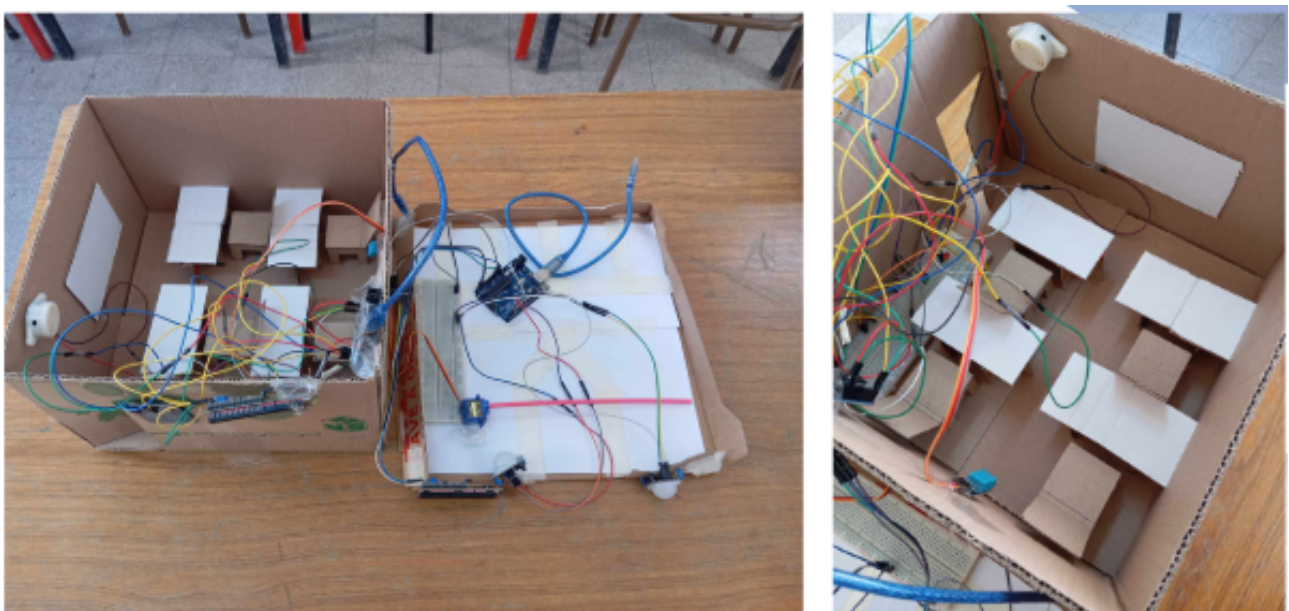


Figure 3.1 – Schéma réel du circuit électronique réalisé pour le système

Les connexions entre les composants ont été réalisées conformément au schéma, tout en respectant scrupuleusement les spécifications électriques et les polarités. Les schémas de fonctionnement sont présentés ci-dessous.

1.1. Schéma électronique : Smart Class

Le schéma 3.2 représente la configuration électronique spécifique au projet Smart Class. Il montre comment les capteurs environnementaux (température, humidité, etc.) et les dispositifs de suivi de présence des élèves interagissent avec le microcontrôleur. Ce design électronique vise à optimiser la gestion de la salle de classe en offrant un système connecté et intelligent.

Liste des composants du projet Smart Class :

- | | | |
|---------------------------|----------------------------|---|
| 1– Carte Arduino Uno | 5– LED violette(Chauffage) | 9– Capteur DHT11 d’humidité et de température |
| 2– Buzzer | 6– LED verte(Eclairage) | |
| 3– RFID | 7– LED rouge(Problème) | 10– Module WiFi |
| 4– LED bleue(Ventilateur) | 8– Afficheur LCD | |

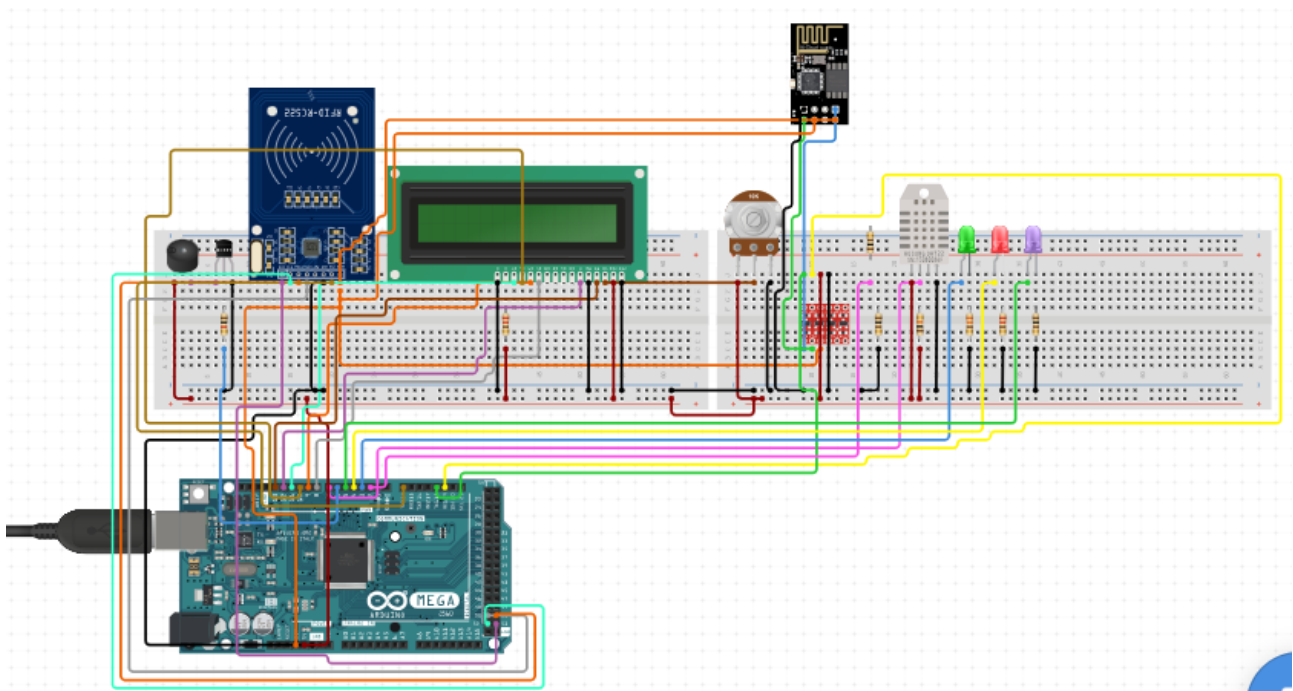


Figure 3.2 – Schéma électronique : Smart Class

1.2. Schéma électronique : Smart Parking

La figure 3.3 détaille la configuration électronique utilisée dans le système Smart Parking. Elle illustre les connexions entre les capteurs de détection de véhicules, les LED indicatrices, et le microcontrôleur. Ce système est conçu pour faciliter la gestion des parkings en fournissant des données en temps réel sur la disponibilité des places.

Liste des composants du projet Smart Parking :

- | | |
|--------------------------|------------------|
| 1– Carte Arduino Uno | 5– Servo moteur |
| 2– Capteur PIR d'entrée | 6– Afficheur LCD |
| 3– Capteur PIR de sortie | 7– RFID |
| 4– LED | 8– Module WiFi |

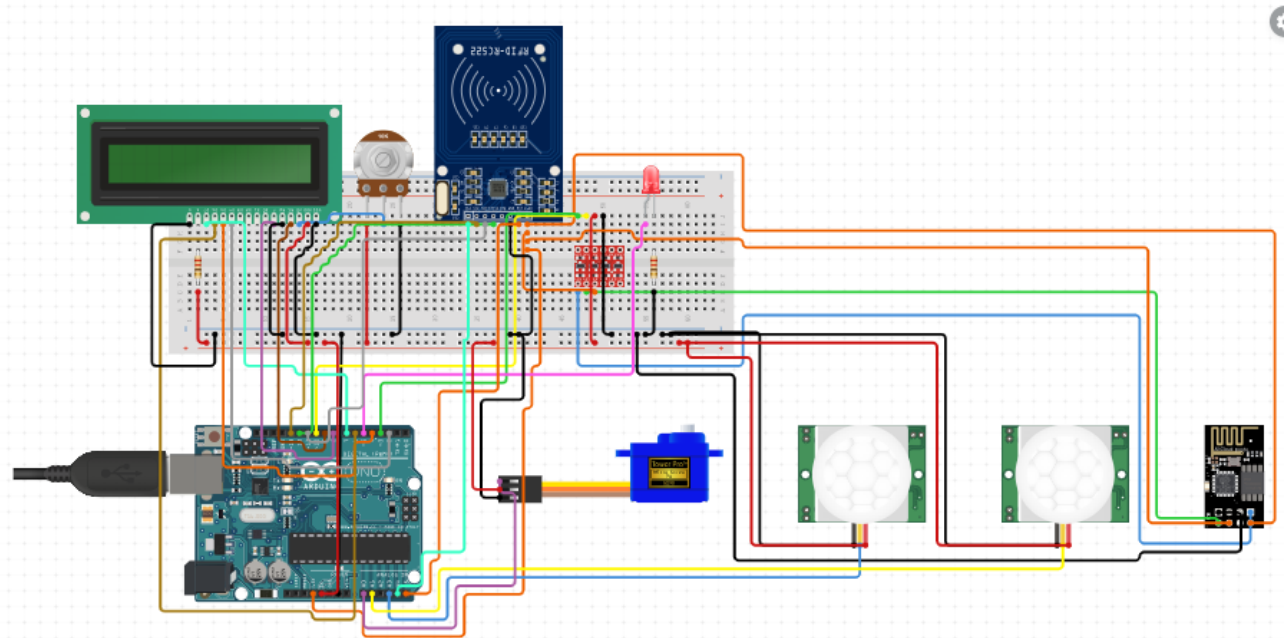


Figure 3.3 – Schéma électronique : Smart Parking

2. Architecture du Système Smart School

L'architecture du système **Smart School**, dans la figure 3.4, repose sur une infrastructure connectée, intégrant des modules Wi-Fi et Firebase pour assurer une gestion intelligente et centralisée. Elle est divisée en deux parties principales : **Smart Class** et **Smart School**, qui collaborent pour créer un environnement scolaire optimisé.

Les salles de classe intelligentes (Smart Class) utilisent des microcontrôleurs connectés pour collecter et transmettre des données environnementales et de présence. Ces informations sont envoyées en temps réel à Firebase, où elles sont centralisées et synchronisées. L'ensemble des modules Smart Class contribue à une gestion globale (Smart School), permettant de surveiller et contrôler les conditions générales de l'établissement scolaire.

Firebase joue un rôle essentiel en assurant le stockage des données, la synchronisation en temps réel et la gestion des notifications pour les utilisateurs. L'application mobile, connectée à Firebase, offre une interface intuitive pour consulter les données, ajuster les paramètres et recevoir des alertes.

Cette architecture connectée garantit une gestion fluide, centralisée et évolutive, tout en facilitant la prise de décision et l'interaction avec le système via une application dédiée.

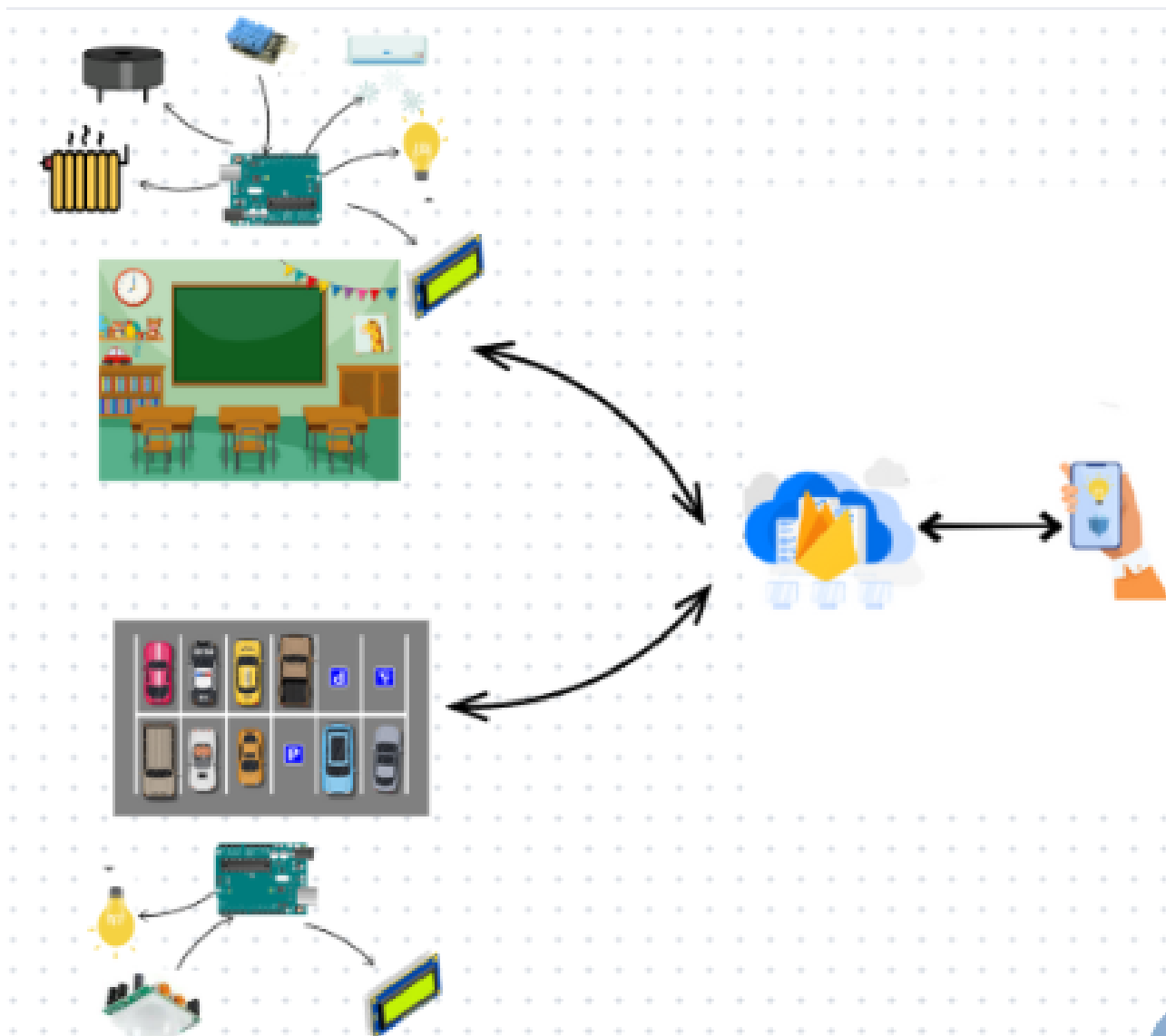


Figure 3.4 – Architecture du Système Smart School

3. Interfaces Utilisateurs

Les interfaces de l'application **Smart School** sont conçues pour offrir une expérience utilisateur intuitive et efficace. Elles permettent aux administrateurs, enseignants et autres utilisateurs de naviguer aisément parmi les différentes fonctionnalités du système. Voici une description des principales interfaces.

3.1. Interface de Chargement

- **Description** : Cette interface s'affiche au démarrage de l'application.
- **Fonctionnalité principale** :
 - * Affiche le logo indiquant le chargement des données nécessaires pour démarrer.

- * Informe l'utilisateur que l'application est en cours d'initialisation.

3.2. Interface d'Inscription

- **Description** : Permet aux nouveaux utilisateurs de créer un compte pour accéder aux services de l'application.
- **Fonctionnalités principales** :
 - * Champs de saisie pour les informations : nom, prénom, email, mot de passe.
 - * Validation des informations saisies pour garantir leur exactitude (vérification d'un mot de passe sécurisé).

3.3. Interface de Connexion

- **Description** : Permet aux utilisateurs existants de se connecter à l'application avec des messages d'erreur clairs en cas de saisie incorrecte.
- **Fonctionnalités principales** :
 - * Champs de saisie pour l'email et le mot de passe.
 - * Authentification sécurisée pour protéger les données des utilisateurs.

La figure 3.5 représente les trois interfaces de chargement et d'authentification.

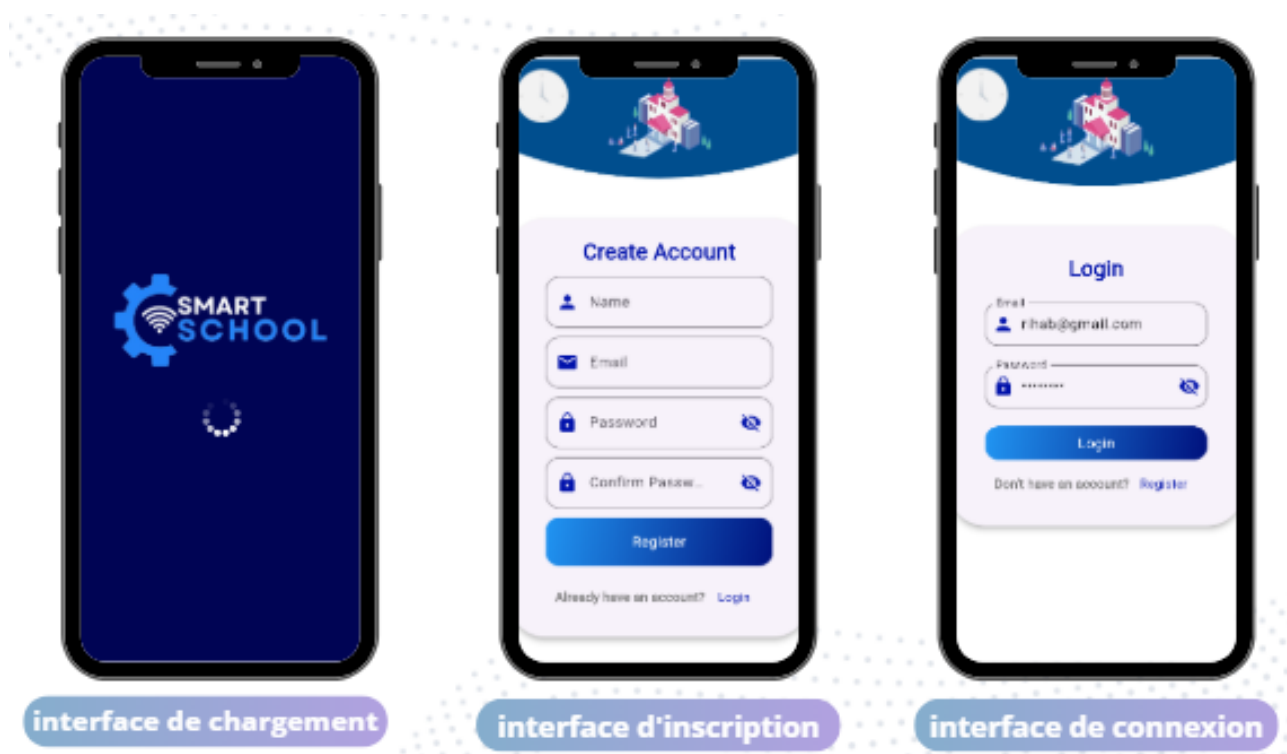


Figure 3.5 – Interfaces de chargement et d'authentification

3.4. Interface d'Accueil

- **Description** : Cette interface sert de menu principal après la connexion. Elle est conçue pour être organisée et interactive, offrant une navigation intuitive grâce à des icônes claires et facilement reconnaissables.
- **Fonctionnalités principales** :
 - * Vue d'ensemble des fonctionnalités disponibles (gestion de classe et gestion de parking).

3.5. Interface de Gestion de Parking

- **Description** : Permet la surveillance du parking de l'établissement.
- **Fonctionnalités principales** : Affichage en temps réel du nombre de places disponibles.

La figure 3.6 représente les interfaces d'accueil et de gestion de parking.



Figure 3.6 – Interfaces d'accueil et de gestion de parking

3.6. Interface de Gestion de Classe Intelligente

- **Description** : Gère les paramètres et conditions environnementales des salles de classe.
- **Fonctionnalités principales** :
 - * Affichage en temps réel des données environnementales (température, humidité, luminosité), mises à jour dynamiquement via la communication Wi-Fi.
 - * Contrôle des systèmes de climatisation, de chauffage et d'éclairage.

- * Surveillance des présences des élèves via RFID.
- * Lorsqu'un seuil critique est dépassé, une alerte claire et visuelle est affichée pour informer l'utilisateur, accompagnée de l'activation de l'éclairage, du chauffage ou de la climatisation, selon le cas.

La figure 3.7 représente les interfaces de gestion de classe intelligente.



Figure 3.7 – Interfaces gestion de classe intelligente

Les interfaces de l'application ont été conçues pour offrir une expérience utilisateur fluide et intuitive. Elles permettent aux utilisateurs de surveiller en temps réel les conditions environnementales et d'obtenir des alertes pertinentes pour assurer un environnement optimal tant pour les élèves que pour la gestion des infrastructures. Enfin, les fonctionnalités proposées par le système, telles que la gestion de la classe et du parking, contribuent à améliorer l'efficacité énergétique, la sécurité et le confort au sein de l'établissement.

Conclusion

Dans ce chapitre, nous avons détaillé les différentes étapes de la réalisation du système Smart School. Nous avons présenté le circuit électronique, qui constitue le cœur de la solution, ainsi que les interfaces utilisateur de l'application mobile.

Conclusion générale

Le projet Smart School IoT a permis de démontrer la faisabilité et les avantages de l'intégration des objets connectés dans le secteur éducatif. En transformant une école traditionnelle en un environnement intelligent et optimisé, nous avons pu répondre aux enjeux modernes de confort et de gestion des ressources.

L'automatisation des environnements tels que la température et l'éclairage dans les salles de classe, combinée à une gestion efficace des parkings et à des systèmes de surveillance avancés, contribue à rendre l'espace scolaire plus fonctionnel, sûr et respectueux de l'environnement. Ce projet ne se limite pas à une simple amélioration des infrastructures existantes, mais ouvre la voie à un futur où les technologies IoT sont au cœur des systèmes éducatifs.

Grâce à la mise en place de ce système, les utilisateurs, qu'il s'agisse des étudiants, enseignants ou administrateurs, bénéficient d'un environnement plus adapté à leurs besoins quotidiens, tout en favorisant une gestion optimisée des ressources. Ce projet a également permis de développer des compétences techniques en matière de conception, de développement et d'intégration de solutions IoT, ce qui constitue une avancée significative dans le domaine de l'éducation intelligente.

Cependant, plusieurs axes d'amélioration et d'extension peuvent être envisagés pour renforcer la portée du projet. Parmi ceux-ci, on pourrait envisager l'intégration d'autres types de capteurs, comme des détecteurs de fumée pour améliorer la sécurité incendie, et l'extension du système à d'autres infrastructures scolaires pour maximiser les avantages de la solution. De plus, l'intégration de caméras de surveillance pour assurer la sécurité des parkings et des zones sensibles pourrait être un ajout essentiel.

Une autre perspective d'optimisation serait l'utilisation de panneaux solaires connectés pour une gestion énergétique plus efficace, réduisant ainsi l'empreinte écologique des établissements scolaires. Ce travail constitue donc une première étape importante vers une transformation numérique plus large des écoles et de l'éducation de demain.

Bibliographie

- [1] “A. Blog. “Arduino IDE 2.3.3 : Discover New Support for Shared Spaces and More.”,” Consulté le 28 novembre 2024. adresse : <https://blog.arduino.cc/2024/09/30/arduino-ide-2-3-3-discover-new-support-for-shared-spaces-and-more/>.
- [2] “Circuito.io : le site qui prépare vos montages électroniques,” Consulté le 28 novembre 2024. adresse : <https://www.minimachines.net/actu/circuito-io-111894>.
- [3] “Google. “Flutter Documentation : Get Started.”,” Consulté le 28 novembre 2024. adresse : <https://docs.flutter.dev/get-started/learn-flutter>.
- [4] “C’est quoi Firebase?,” Consulté le 28 novembre 2024. adresse : <https://www.v-labs.fr/glossaire/firebase/>.
- [5] “Définition Visual Studio Code,” Consulté le 29 novembre 2024. adresse : <https://bibility.fr/definition-visual-studio-code/>.

Résumé

Le système Smart School intègre des technologies IoT pour optimiser la gestion des classes et des parkings. Il ajuste en temps réel les conditions environnementales de la classe (température, humidité, éclairage) et automatise le contrôle de la climatisation et de l'éclairage. De plus, il propose une gestion en temps réel de la disponibilité des places de parking via une application mobile. Le système utilise des technologies avancées comme Flutter pour l'application mobile, Arduino pour les capteurs et actionneurs, et Firebase pour la gestion des données, assurant ainsi une performance stable et fiable.

Mots clés : Flutter, Arduino, Application mobile, Firebase, IoT, Smart School.

Abstract

The Smart School system integrates IoT technologies to optimize the management of classrooms and parking lots. It adjusts the environmental conditions of the classroom (such as temperature, humidity, lighting) in real time and automates the control of air conditioning and lighting. Additionally, it provides real-time management of parking space availability via a mobile application. The system uses advanced technologies such as Flutter for mobile application development, Arduino for sensors and actuators, and Firebase for data management, ensuring stable and reliable performance.

Keywords : Flutter, Arduino, Mobile Application, Firebase, IoT, Smart School.

الملخص

نظام Smart School يستخدم تقنيات الـ IoT لتحسين إدارة الفصول الدراسية والمواقف. يقوم بتعديل الظروف البيئية للفصل الدراسي (مثل درجة الحرارة والرطوبة والإضاءة) في الوقت الفعلي، ويقوم بتشغيل التحكم التلقائي في التكييف والإضاءة. بالإضافة إلى ذلك، يقدم نظاماً لإدارة توفر أماكن الركن في الوقت الفعلي عبر تطبيق موبايل. يستخدم النظام تقنيات متقدمة مثل Flutter لتطوير التطبيق الموبايل، و Arduino للتحكم في الحساسات والمشغلات، و Firebase لإدارة البيانات، مما يضمن أداءً ثابتاً وموثوقاً.

الكلمات المفتاحية: Flutter, Arduino, تطبيق موبايل, Firebase, IoT, Smart School.