TRAVAUX D'INITIATIVE
PERSONNELLE ENCADRÉS T.I.P.E.
2023

المدرسة المحمدية للمهندسين
Ecole Mohammadia d'Ingénieurs

Royaume du Maroc

Ministère de l'Education Nationale,
du Préscolaire et des Sports

LA VILLE

Sujet :

SmartPark : Système de gestion de parking innovant

préparé par :

BRIGUI Khalil

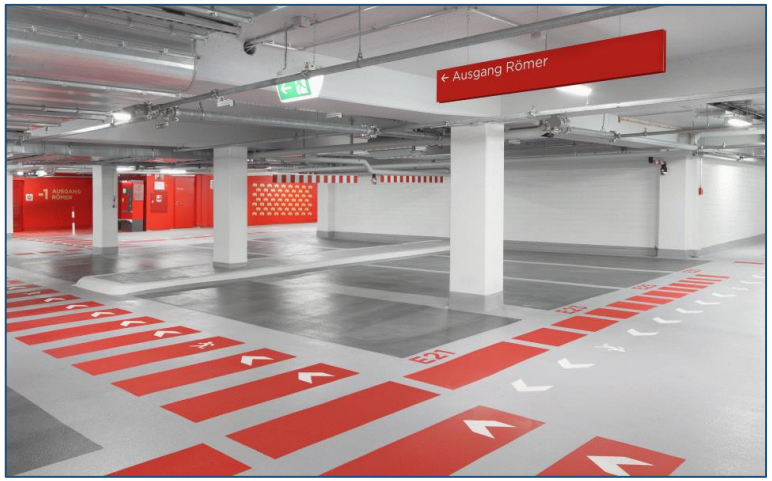
Plan de la présentation

1. Introduction
2. Présentation fonctionnelle du système : diagramme Sys ML
3. Analyse et solution technique du système
4. Résultats
5. Conclusion

I- Introduction

Dans la ville, les automobilistes cherchent des alternatives de stationnement efficaces. La nécessité d'un système de stationnement performant découle de l'accroissement de la congestion routière, de la pollution causée par les véhicules, la fatigue des conducteurs.

https://encrypted-tbn0.gstatic.com/



https://www.unit-design.de/

La solution présentée met en pratique une solution innovante et efficace à un problème de gestion de la circulation urbaine.

1 Problématique

La gestion efficace d'un parking permet de garantir la fluidité de circulation routière.

- *comment peut-on extraire et traiter les codes-barres sur la carte RFID fournie et les assigner aux voitures entrantes dans un parking ?*
- *Comment peut-on mesurer l'efficacité du système de gestion de parking en termes de fluidité du trafic et de satisfaction des utilisateurs ?.*

https://www.akuiteo.com/

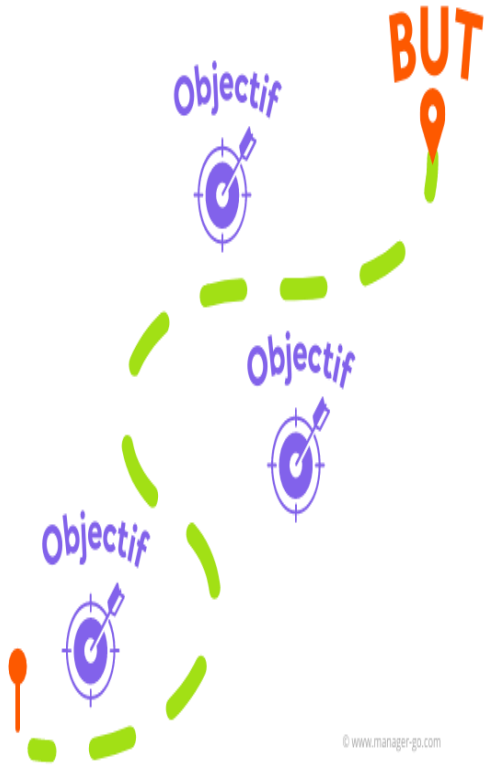
2 Objectif d'étude

1. Détecter les véhicules entrant dans le parking en traitant de manière fiable et efficace le code extrait de la carte RFID fournie par le distributeur de cartes.

2. Stocker les informations RFID et les coordonnées d'entrée et de sortie des véhicules dans une base de données.

3. Mettre en place un algorithme de tarification intelligent pour calculer le coût de stationnement des véhicules et l'intégrer dans une application mobile pour permettre aux utilisateurs un paiement à la sortie.

4. Réaliser un prototype



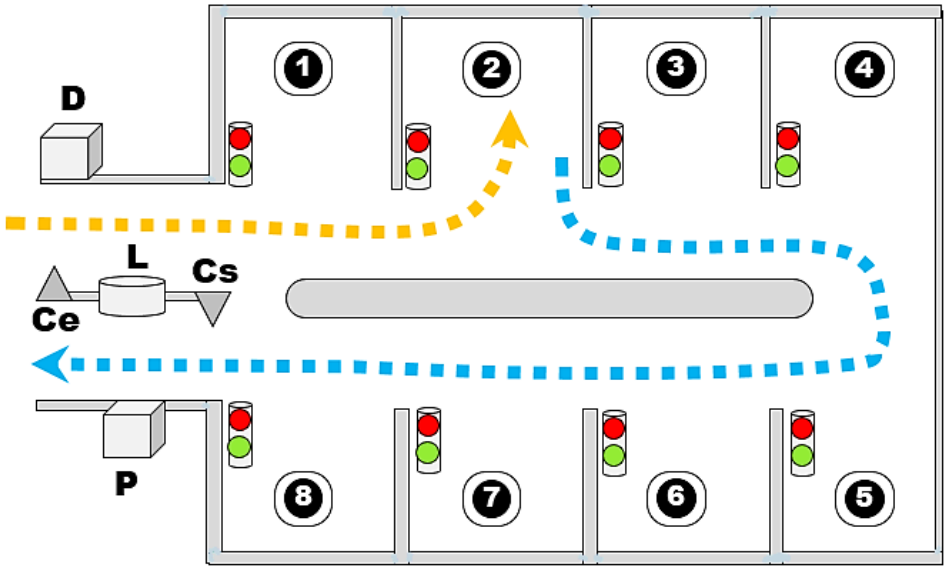
https://www.manager-go.com/

I- Présentation fonctionnelle du système

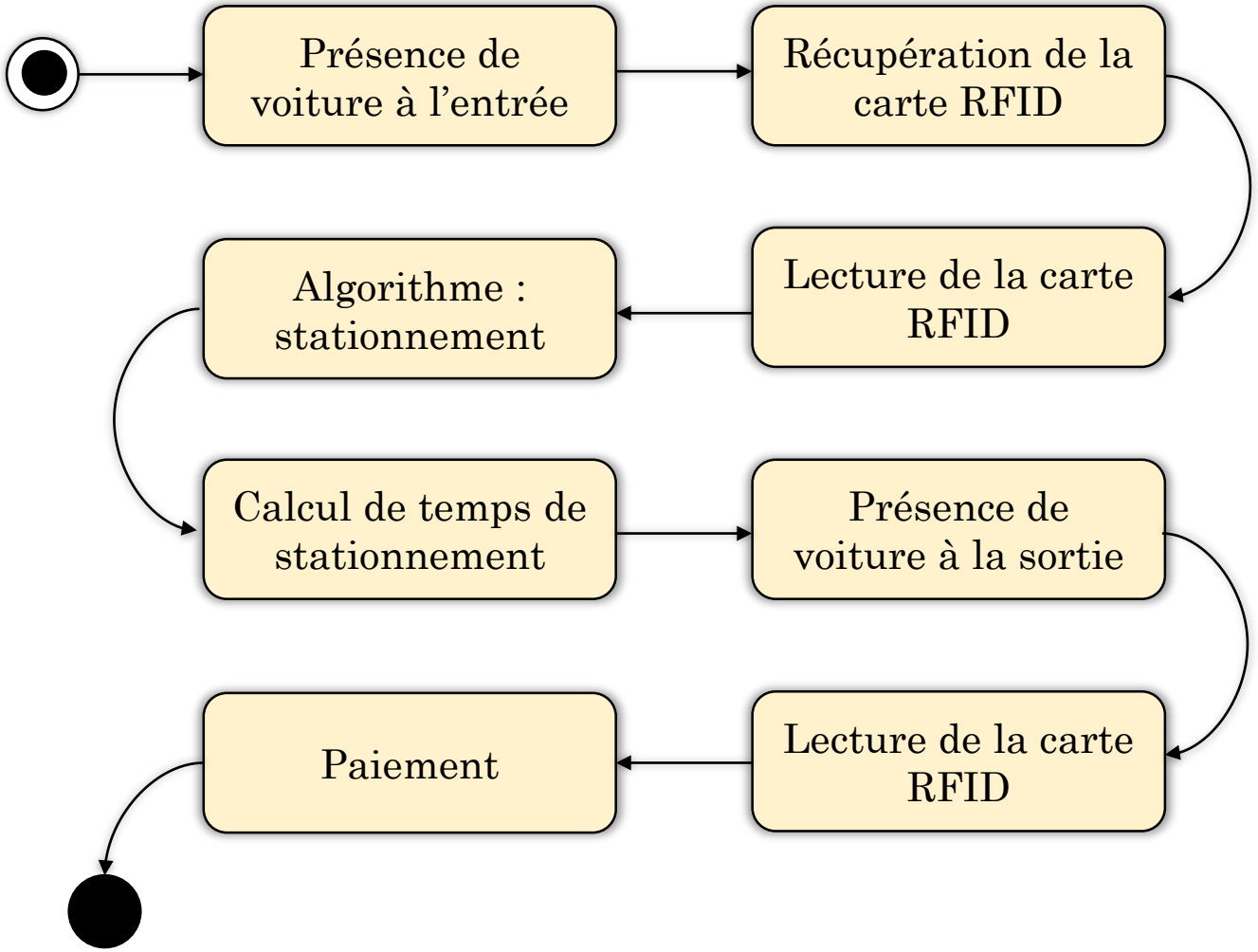
1 Diagramme de cas d'utilisation : uc

2 Diagramme d'exigence: Req

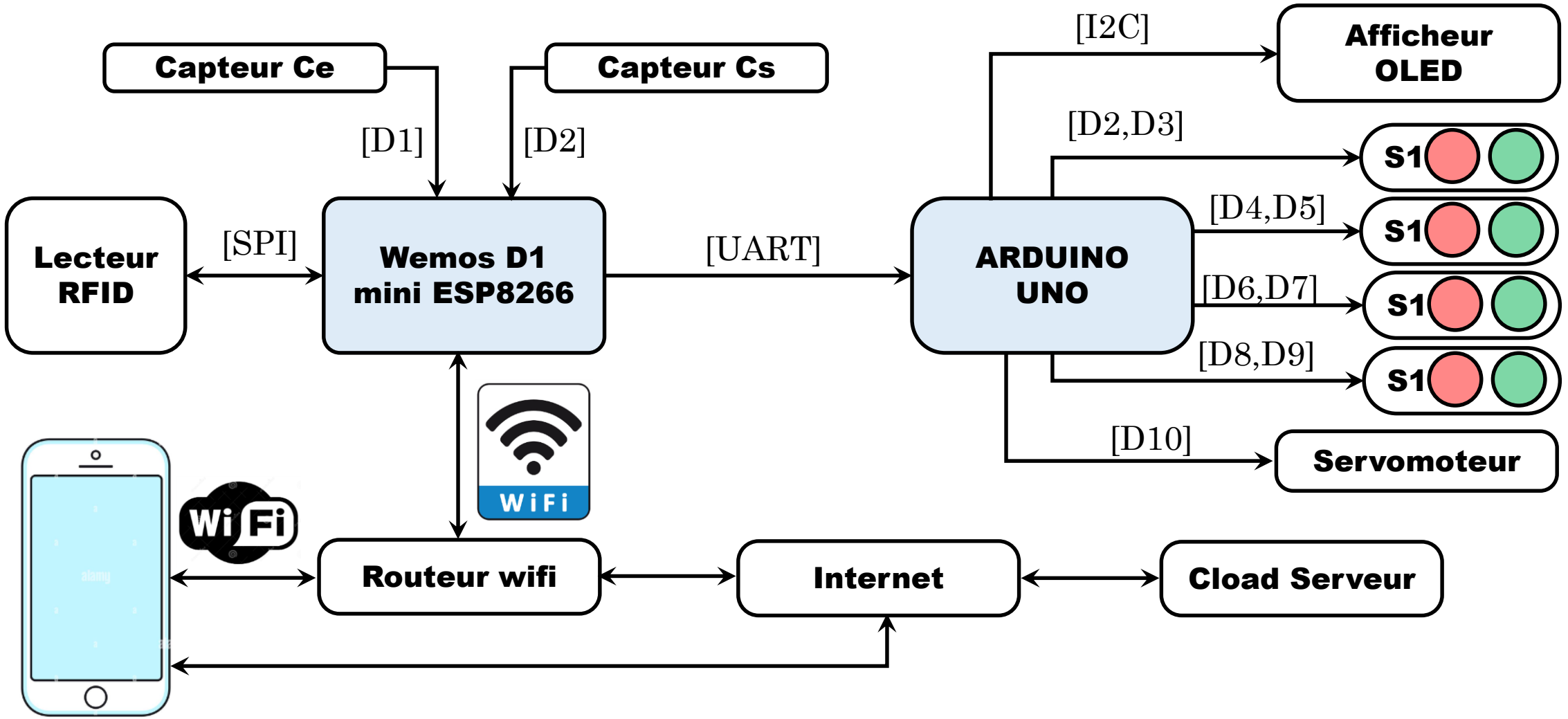
1 Fonctionnement du système



- **D:** distributeur des carte RFID
- **P:** poste de paiement
- **L:** lecteur des cartes RFID
- **Ce:** capteur de détection d'entrée
- **Cs:** capteur de détection de sortie



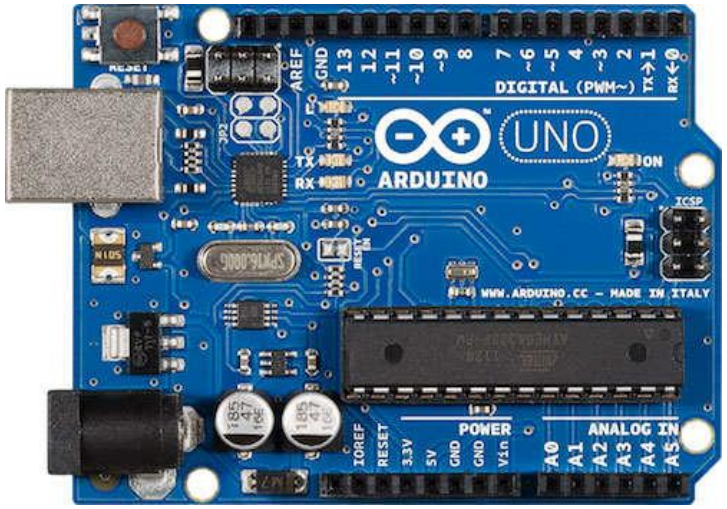
2 Description de prototype : Schéma globale



2 Description de prototype : cartes utilisées

Carte Arduino

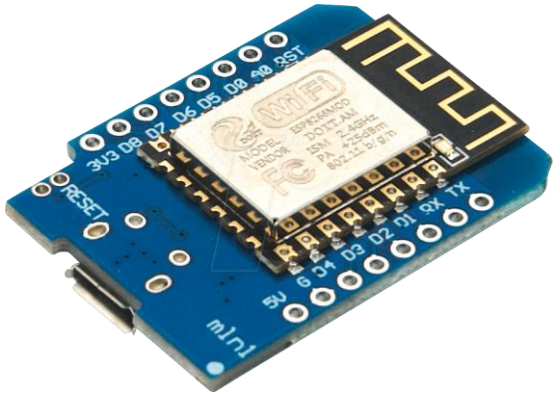
Arduino UNO est une carte de développement open-source populaire pour les projets électroniques.

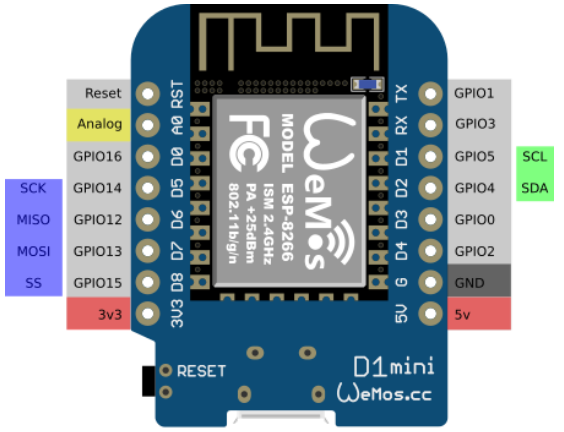


- Microcontrôleur atmega338 à 16 MHz
- 14 broches d'Entrée/Sortie numériques
- 6 broche d'Entrée/Sortie analogique (3,3 V max)

Carte à ESP8266 Wemos D1 mini

WeMos D1 mini est une petite carte WiFi basée sur l'ESP8266. Cette carte est compatible avec l'IDE Arduino.





- Microcontrôleur ESP8266EX cadencé à 80 MHz
- Connectivité Wifi 802.11 b/g/n
- 11 broches d'Entrée/Sortie numériques
- 1 broche d'Entrée/Sortie analogique (3,3 V max)

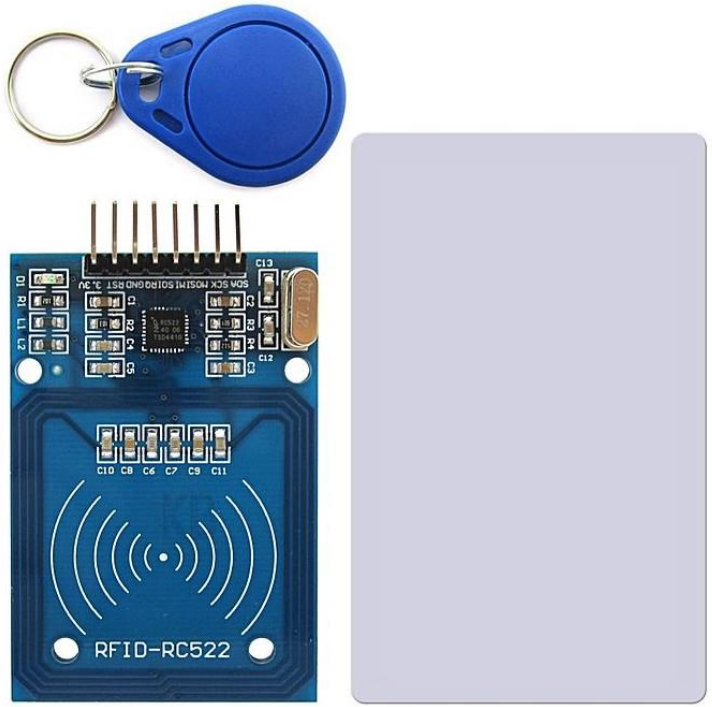
3 Expérience N1: lecture de la carte RFID RC522

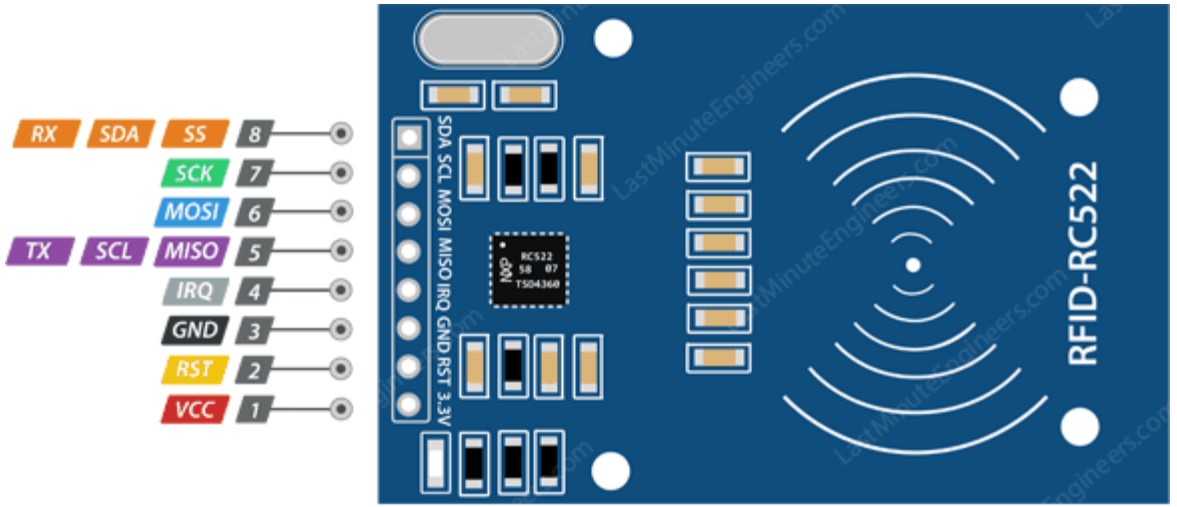
Un système RFID ou de radio-identification par fréquence est constitué de deux composants principaux : une étiquette attachée à l'objet à identifier et un lecteur qui lit l'étiquette.

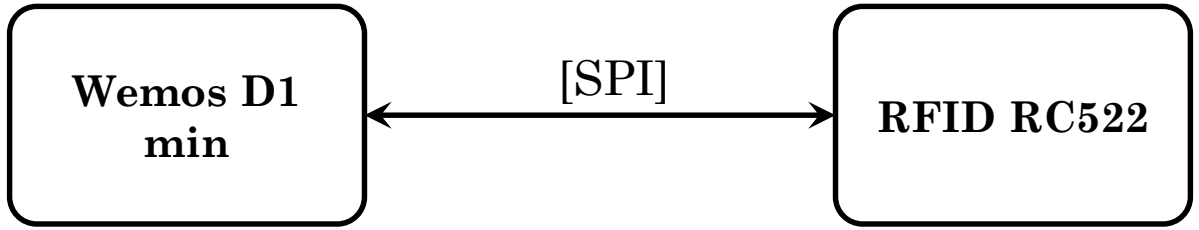
Lorsqu'une carte RFID est rapprochée d'un lecteur, ce dernier génère une onde électromagnétique qui alimente la puce de l'étiquette. La puce répond en envoyant ses informations stockées de retour au lecteur (comporte un code d'identification de la carte).

3 Expérience N1: lecture de la carte RFID RC522

Le module RFID RC522 est un module électronique permettant de lire et écrire des tags RFID de type MIFARE. Il est basé sur le circuit intégré MFRC522 de NXP.







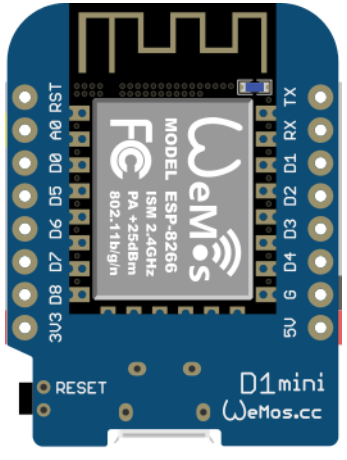
- **Alimentation** : 3.3V
- **Interface de communication** : I2C/SPI/UART

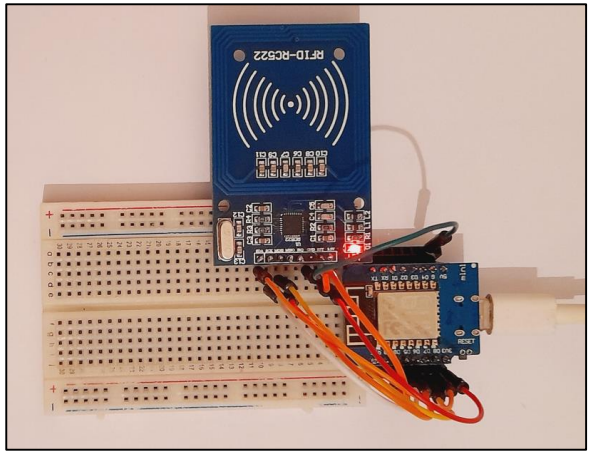
3 Expérience N1: lecture de la carte RFID RC522

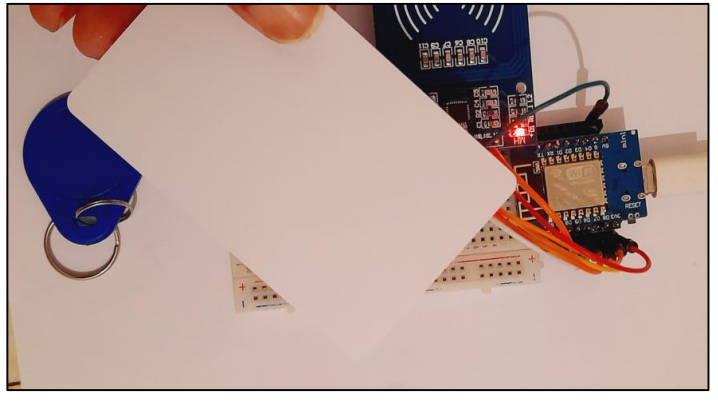
Objectif: le but de cet expérience est de connecter le lecteur RFID avec carte Wemos D1 et d'extraire le code de la carte l'identifier.

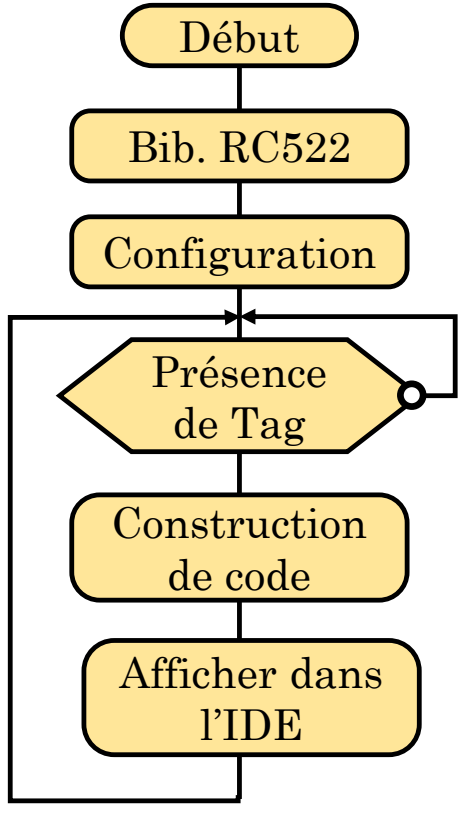
RFID RC522	Wemos D1
RST	D0
MISO	D6
MOSI	D7
SCK	D5
SS/SDA	D8

SPI





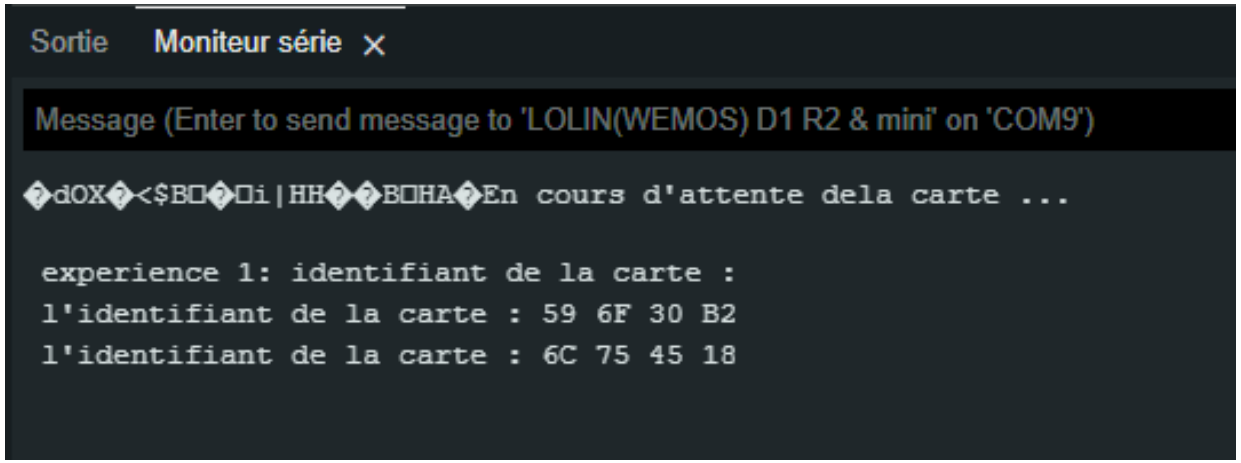




Programme en Annexe

3 Expérience N1: lecture de la carte RFID RC522

Résultat :

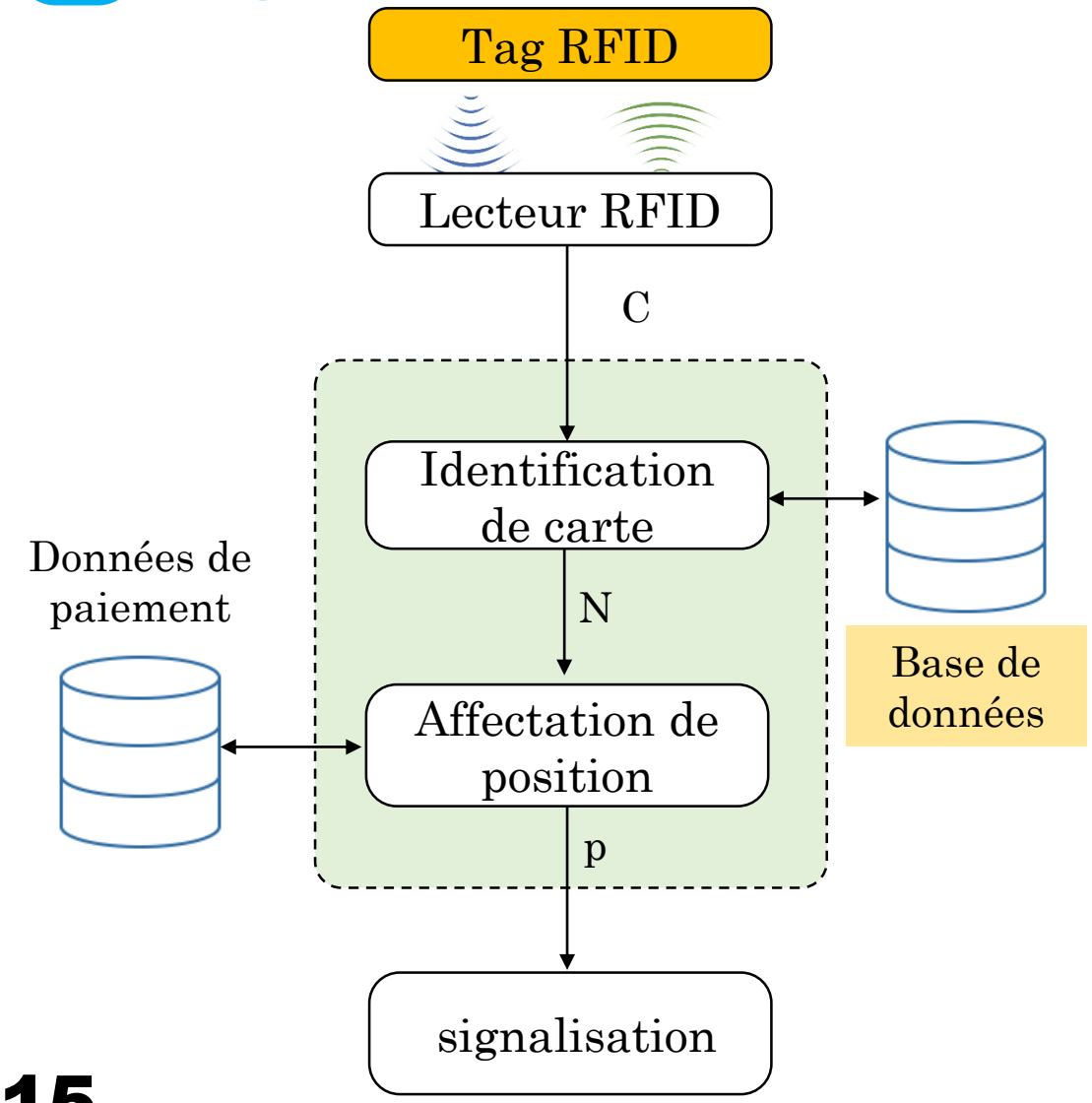


59	6F	30	B2

6C	75	45	18

Identifiant de la carte est composé de 4 octets sont codés en hexadécimale

4 Algorithme de stationnement



Description de l'algorithme de stationnement

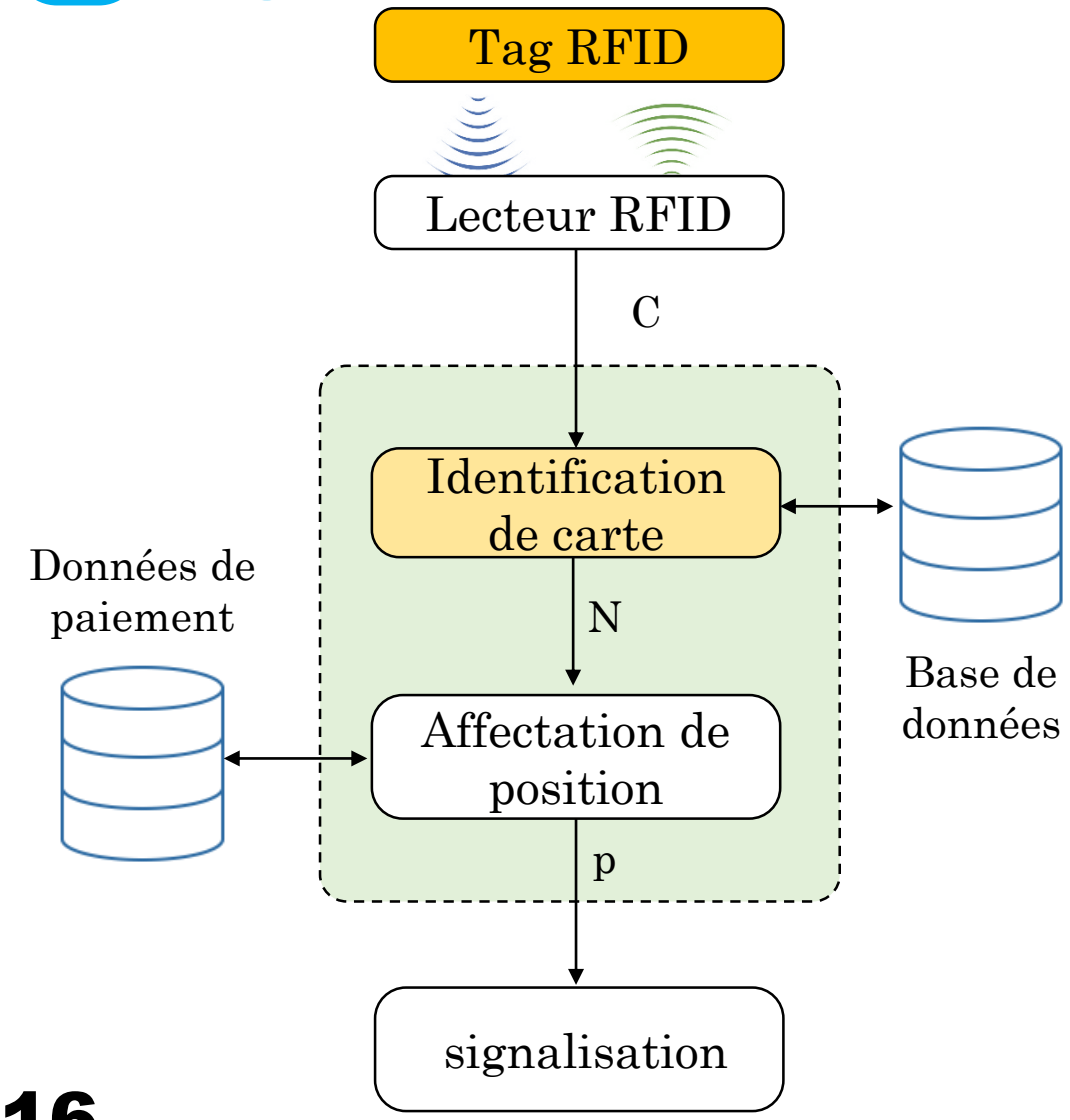
1- Base de données

En général, les identifiants des cartes fournis par le distributeur sont stockés dans une base de données, qui les organise dans un tableau tel que celui-ci :

Carte (N)	Identifiant (C)
1	59 6F 30 B2
2	59 6D 30 B2
3	59 6F 10 B2
4	6C 75 45 18

```
89
90    T[0]="59 6F 30 B2";
91    T[1]="59 6D 30 B2";
92    T[2]="59 6F 10 B2";
93    T[3]="6C 75 45 18";
94
```

4 Algorithme de stationnement



Description de l'algorithme de stationnement

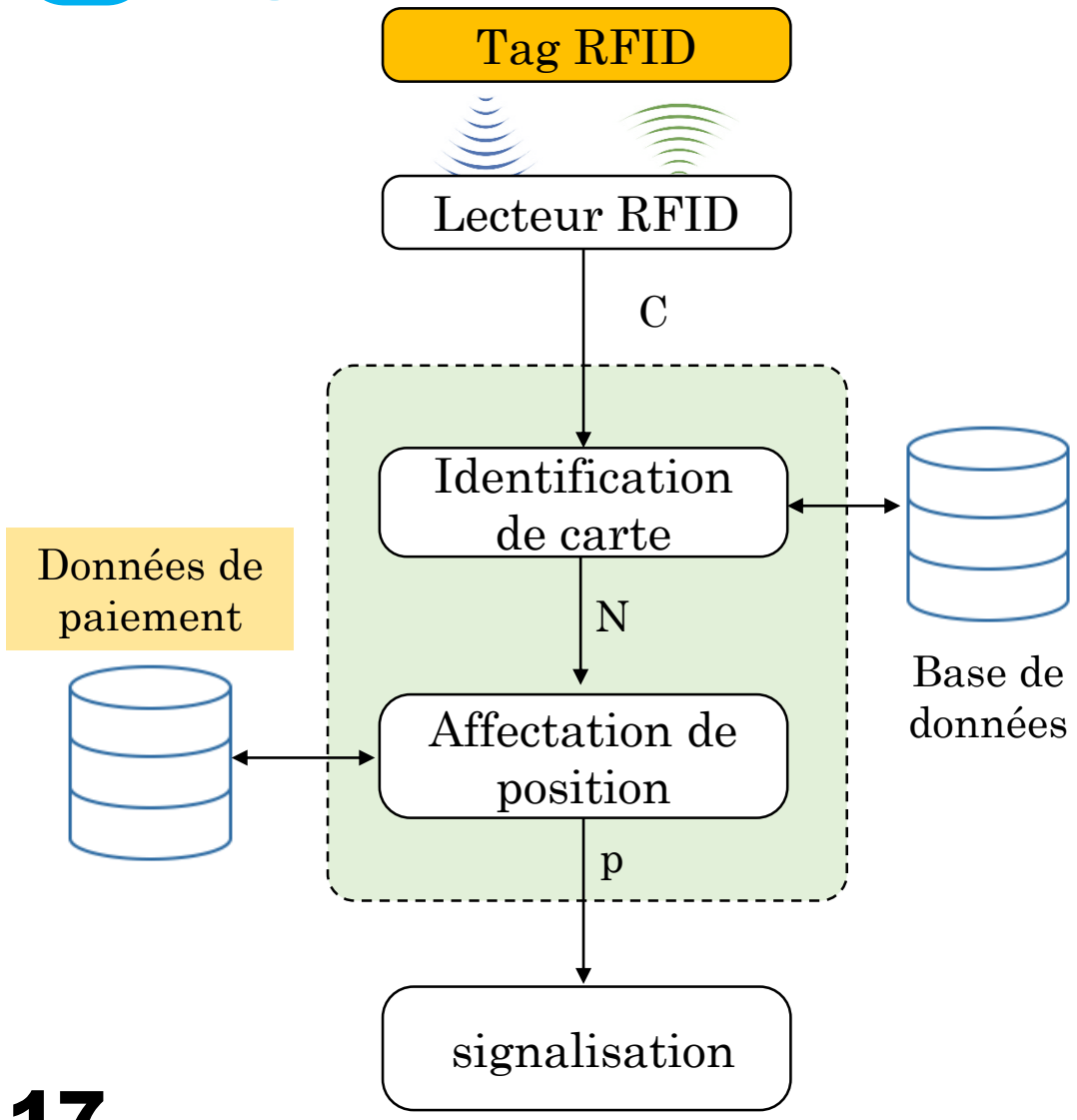
2- Identification de la carte RFID

Sa fonction consiste à comparer la carte C insérée avec la base de données T, afin de retrouver son numéro de référence N dans cette dernière.

```
31  int    identification()
32  {  int i, N;
33     if(C!=B)
34     for(i=0;i<4;i++)
35        if(C==T[i])  N=i+1;
36
37   return N;
38  }
```

Si la carte insérée **n'est pas trouvée** dans la base de données, la fonction **renverra la valeur N= 0**.

4 Algorithme de stationnement



Description de l'algorithme de stationnement

3- Donnée de paiement

Elle comprend un tableau qui rassemble le numéro de la carte assignée, sa position et l'heure d'entrée au parking.

Position (p)	0	0	1	0
La carte (N)	0	0	3	0
Durée (h)	0	0	h	0

```
12  int Table[3][4];
13
```

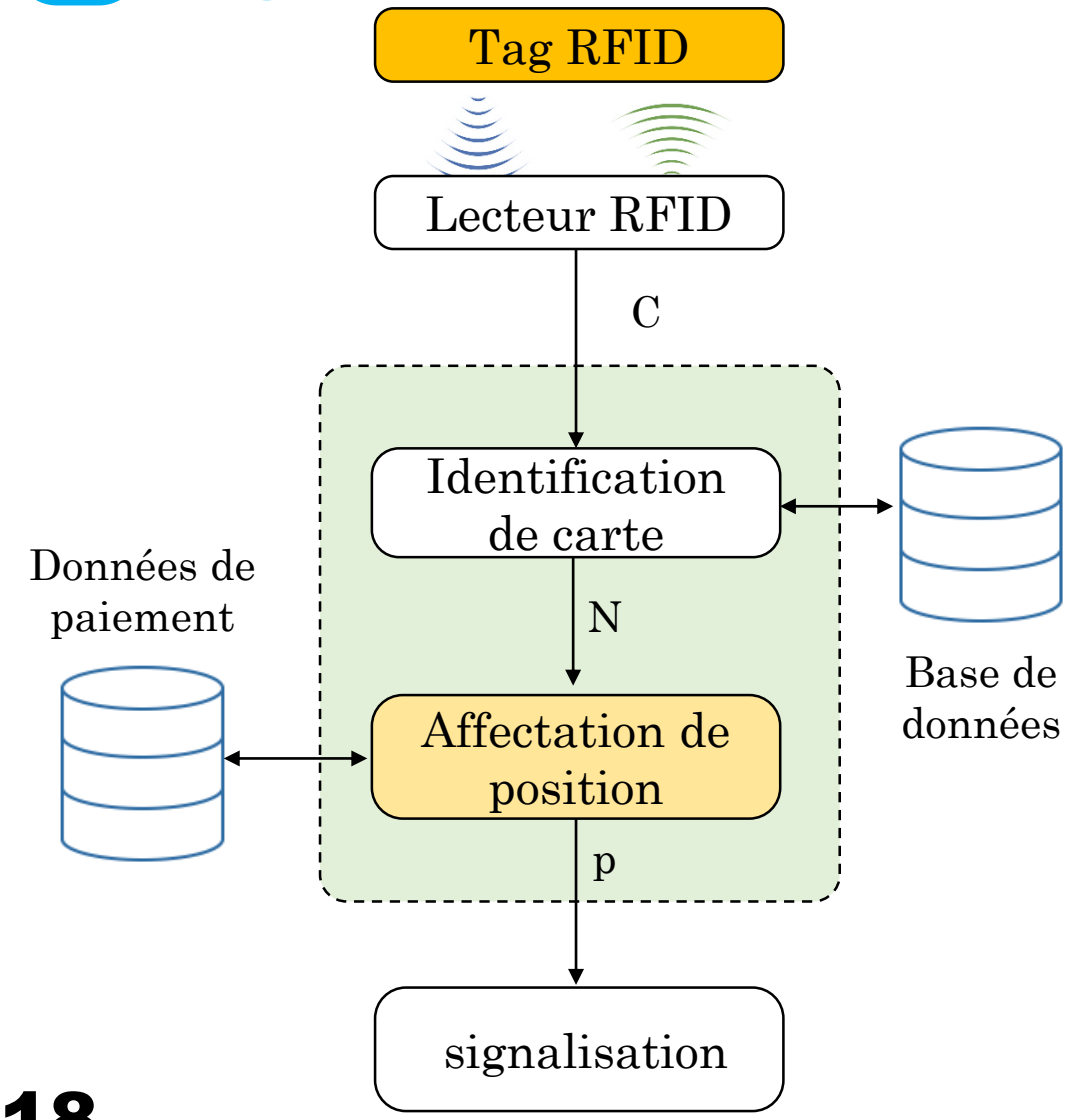
Déclaration d'un tableau

```
97    Table[0][0]=0;
98    Table[0][1]=0;
99    Table[0][2]=0;
100   Table[0][3]=0;
101
```

Initialisation

N.B : Au départ, le parking est vide, c'est-à-dire qu'il ne contient aucun véhicule

4 Algorithme de stationnement



Description de l'algorithme de stationnement

4- Affectation de position au véhicule

En utilisant les données de paiement (tableau) :

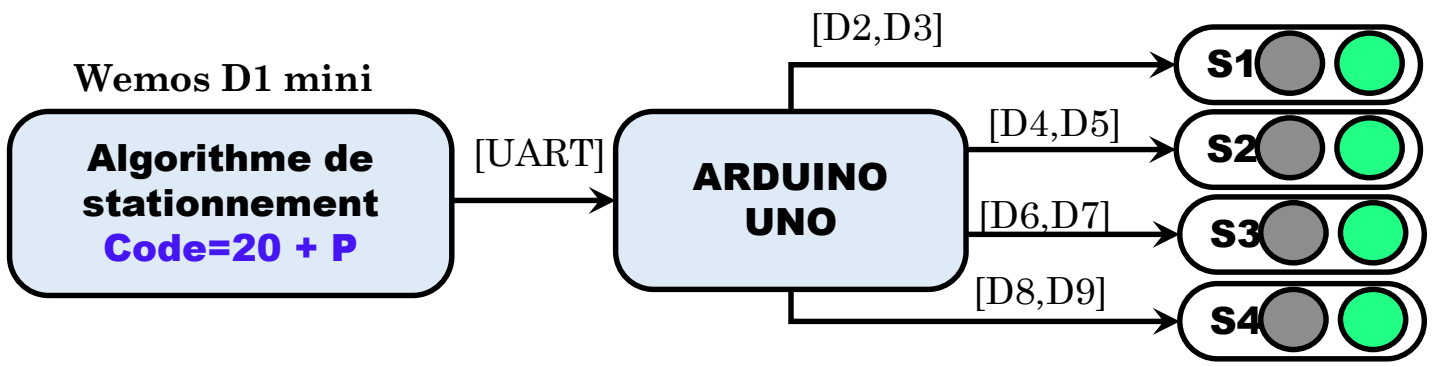
- Recherche une place de stationnement libre.
- Mise à jour de l'état de la case correspondante à 1 pour indiquer qu'elle est occupée
- enregistre le numéro de la carte dans la case correspondante
- enregistre la durée d'entrée de stationnement

```
delay(1000); t++;
```

```
int    affectation(int N)
{  int i,P;

   for (i=0;i<4;i++)
      if(Table[0][i]==0)
      {
         Table[0][i]=1;
         Table[1][i]=N;
         Table[2][i]=t;
         P=i+1;
         i=4;
      }
 return P;
}
```

5 Algorithme de signalisation pour stationnement

Une fois que la position de stationnement a été calculée, un code est généré pour être envoyé à l'Arduino, qui utilise des LEDs de signalisation verte pour indiquer à l'utilisateur la place de stationnement attribuée.

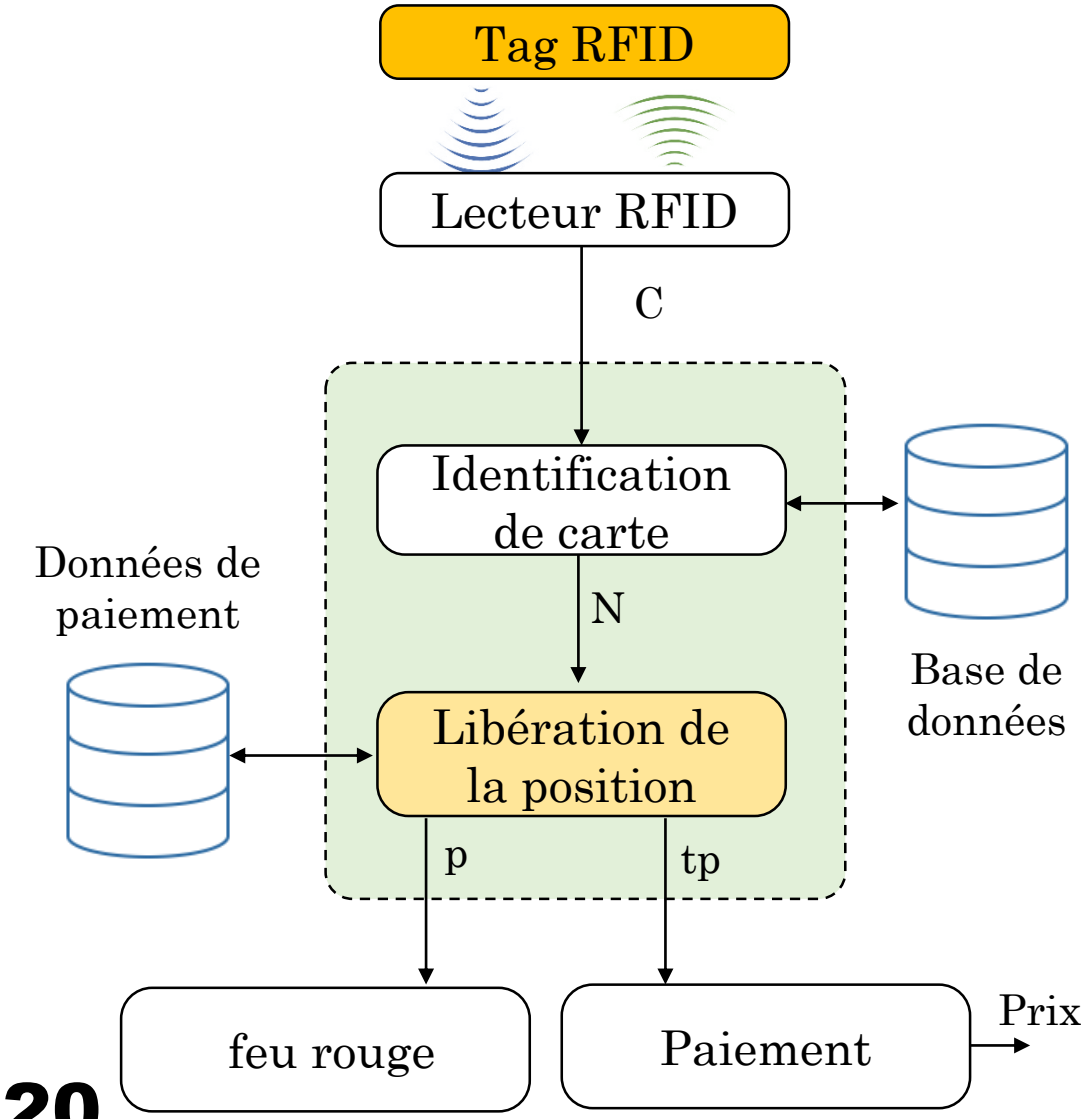


Code envoyer	LED verte	LED rouge	Clignotement
21	D3 ← ON	D2 ← OFF	5 fois
22	D5← ON	D4← OFF	5 fois
23	D7 ← ON	D6 ← OFF	5 fois
24	D9 ← ON	D8← OFF	5 fois

```
void  clignotement(int led)
{ int i;

  digitalWrite(led-1,LOW);
  for(i=0;i<5;i++)
  {
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    delay(500);
  }
}
```

```
switch(code)
{
  case 21 : clignotement(ledv1);;break;
  case 22 : clignotement(ledv2);;break;
  case 23 : clignotement(ledv3);;break;
  case 24 : clignotement(ledv4);;break;
```

6 Algorithme déstationnement



Description de l'algorithme de déstationnement

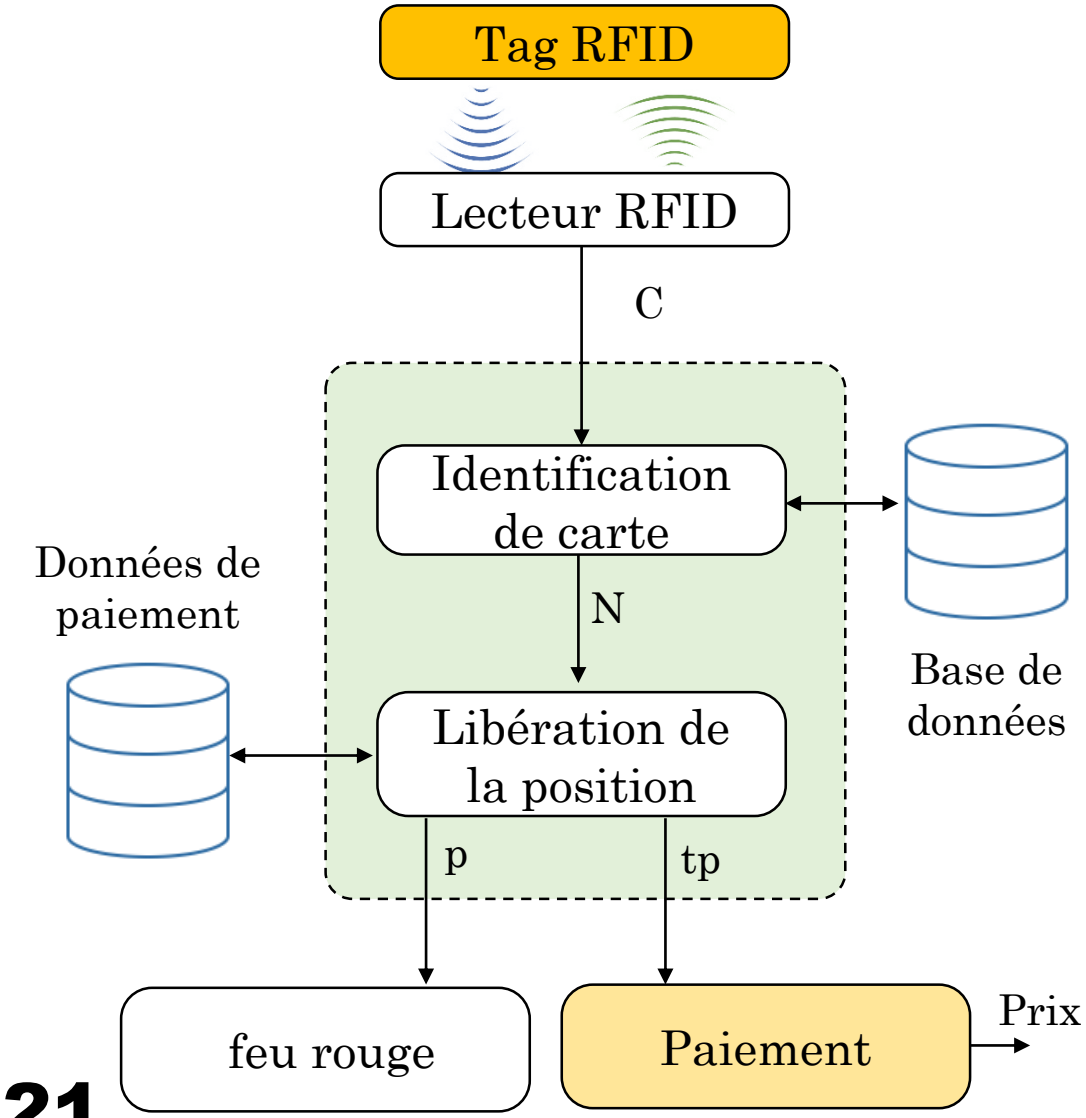
1- libération de la position du parking

Une fois que la carte RFID a été identifiée, on consulte le tableau de paiement pour débloquer l'emplacement de stationnement (position = 0) et calculer la durée de stationnement.

Le résultat de l'algorithme est la **position** à libérer (p) et **l'heure d'entrée** dans le parking (tp).

```
int    liberation(int N)
{  int i,P;

   for (i=0;i<4;i++)
      if(Table[1][i]==N)
      {
         Table[0][i]=0;
         Table[1][i]=0;
         tp= Table[2][i];
         Table[2][i]=0;
         P=i+1;
         i=4;
      }
 return P;
}
```

6 Algorithme déstationnement



Description de l'algorithme de déstationnement

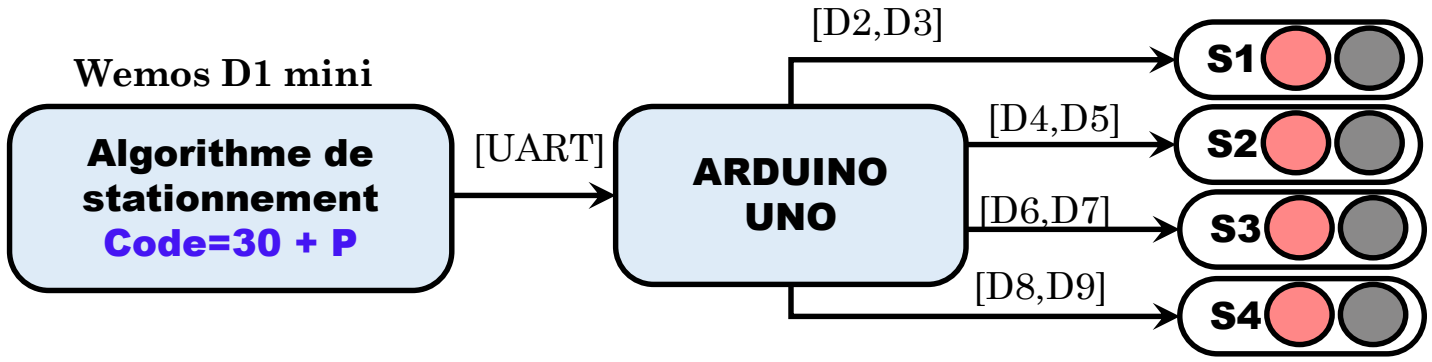
2- Paiement

Le calcul du paiement est basé sur la différence entre l'heure d'entrée et l'heure de sortie, multipliée par un coefficient pour déterminer le prix correspondant.

```
int    paiement(int x)
{  int prix;
   float K=0.00139; // 1H =5 Dhm
   prix=K*(t-tp);

    if (prix<5)  prix=5;

 return prix;
}
```

7 Algorithme de signalisation pour déstationnement

Une fois que la position de déstationnement a été calculée, un code est généré pour être envoyé à l'Arduino, qui utilise des LEDs de signalisation rouge pour indiquer à l'utilisateur la place de stationnement attribuée.



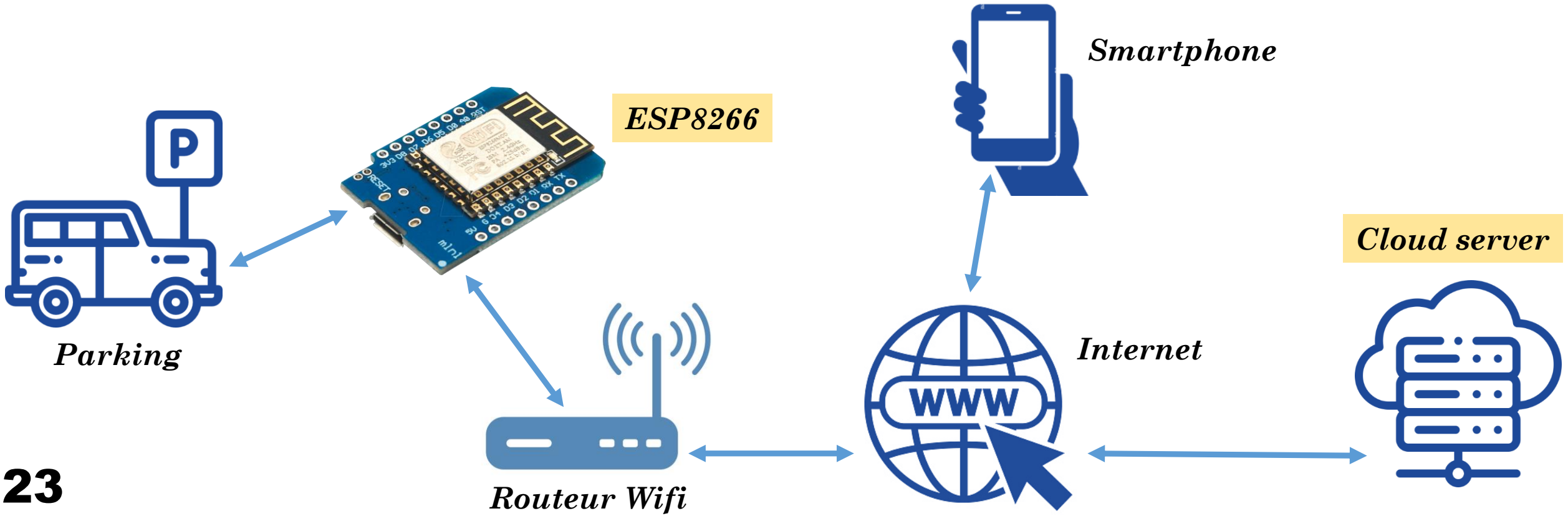
Code envoyer	LED verte	LED rouge	Clignotement
31	D3 ← OFF	D2 ← ON	NON
32	D5← OFF	D4← ON	NON
33	D7 ← OFF	D6 ← ON	NON
34	D9 ← OFF	D8← ON	NON

```
if (Serial.available())
{
  int code = Serial.parseInt();

  switch(code)
  {
    case 21 : clignotement(ledv1);;break;
    case 22 : clignotement(ledv2);;break;
    case 23 : clignotement(ledv3);;break;
    case 24 : clignotement(ledv4);;break;

    case 31 : libirer_position(ledr1);;;break;
    case 32 : libirer_position(ledr2);;;break;
    case 33 : libirer_position(ledr3);;;break;
    case 34 : libirer_position(ledr4);;;break;

  }
```

8 Parking connecté

L'objectif:

mettre à disposition en temps réel des informations sur l'état du parking en collectant et partageant ces données via Internet. Les utilisateurs pourront accéder à ces informations via une application mobile installée sur leur smartphone.

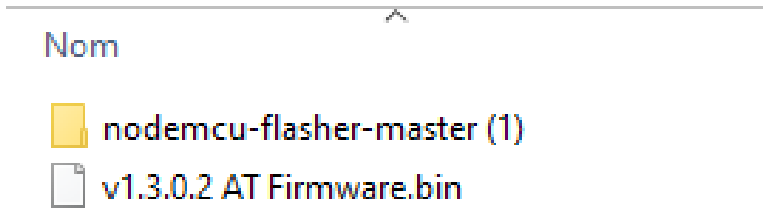


8 Parking connecté: configuration de carte Wemos D1 mini

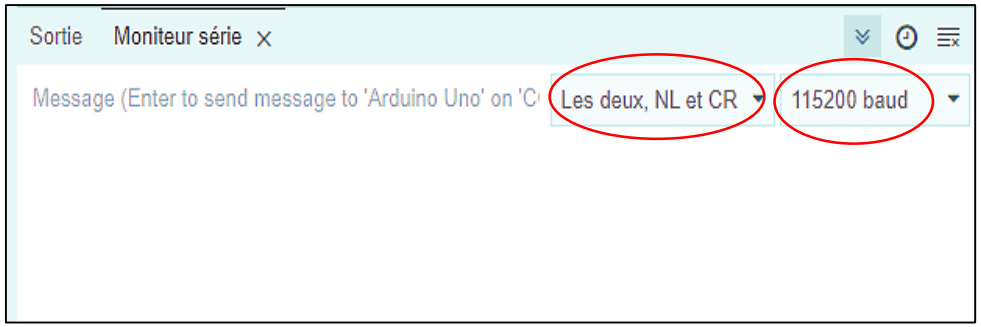
❑ *Téléchargement des fichiers*

1. Télécharger le driver de la carte sur https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers

2. Télécharger le flasher https://github.com/nodemcu/nodemcu-flasher/raw/master/Win64/Release/ESP8266Flasher.exe
3. Télécharger le fichier bin pour la configurer par les commandes AT https://drive.google.com/file/d/1yYJ6saerPClpkmQTSVvcCdORnKxibhkr/view



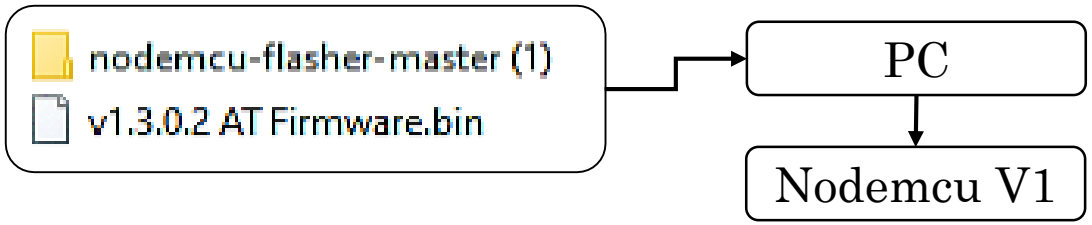
❑ *Les commande de configuration*

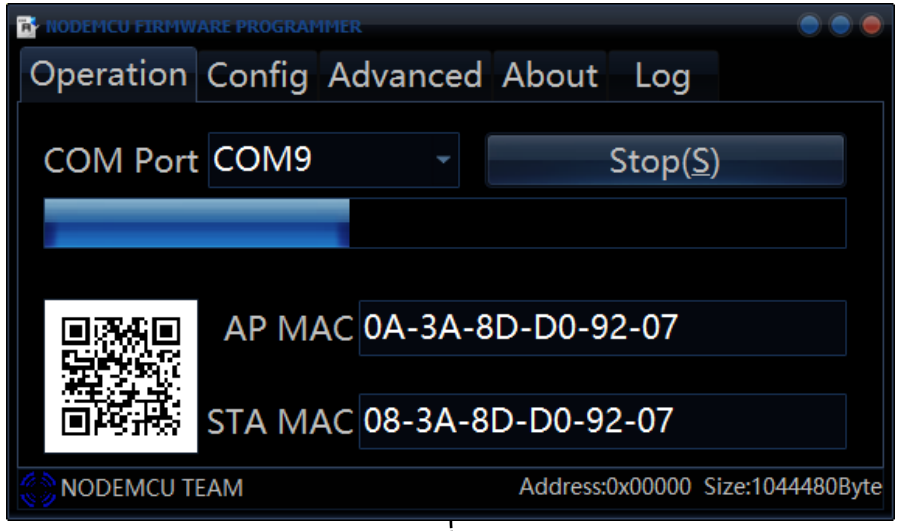


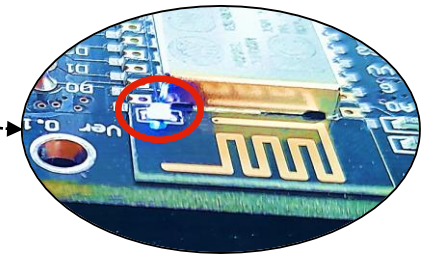
	Commande	signification
1	**AT**	Début de communication (OK)
2	**AT+RST**	Remise à zéro du module
3	**AT+GMR**	Pour vérifier le Reset (exécuter le, 2 fois)
4	**AT+CIFSR**	Récupère la adresse IP du module
5	**AT+CWMODE**= 1	Le mode fonctionnement : 1 serveur et 2 client
6	**AT+CWSAP**=" ESP","123456789",1,4	Changement de cordonnés de wifi

8 Parking connecté: configuration de carte Wemos D1 mini

❑ *Flashage de la carte NodeMCU*

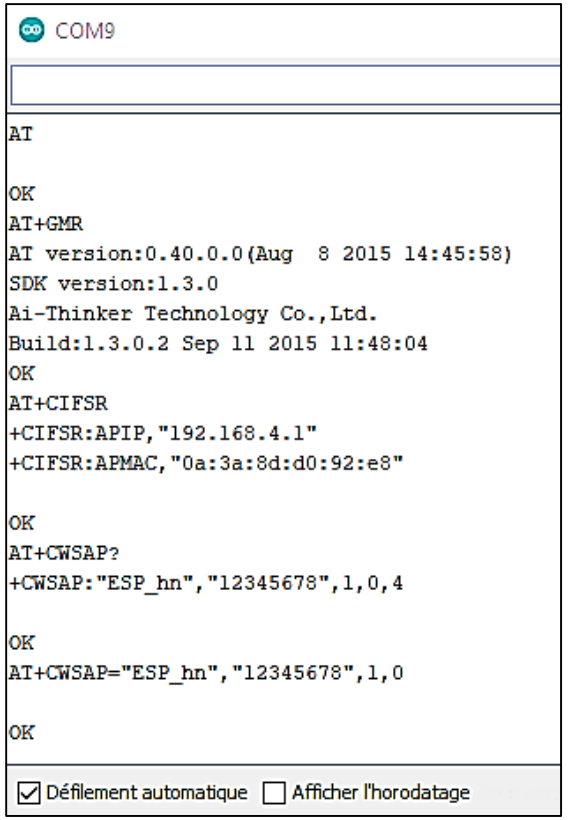






❑ *Configuration de wifi*

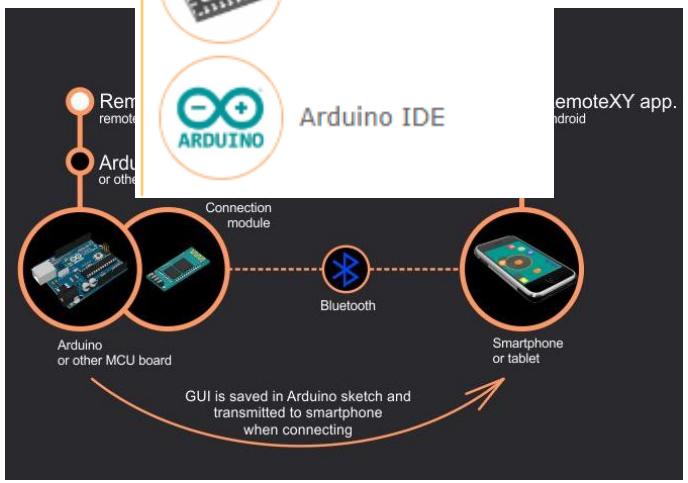
- Initialisation de la carte (Reset)
- Mode client
- Coordonnées de wifi: nom(ESP_hn), Mp(12345678)



8 Parking connecté: conception d'une application mobile

❑ *Remotexy ?*

Remotexy est une plateforme logicielle simplifiant le développement d'applications de contrôle à crocontrôleurs. Elle fournit u nobile pour créer facilement lisateur graphiques pour con er des dispositifs électroniqu



❑ *Remotexy : création de clé secrète*

On contrôler la carte Wemos D1 à distance en se connectant à un serveur de cloud. Afin d'effectuer cette connexion, il est nécessaire de disposer d'une clé secrète.

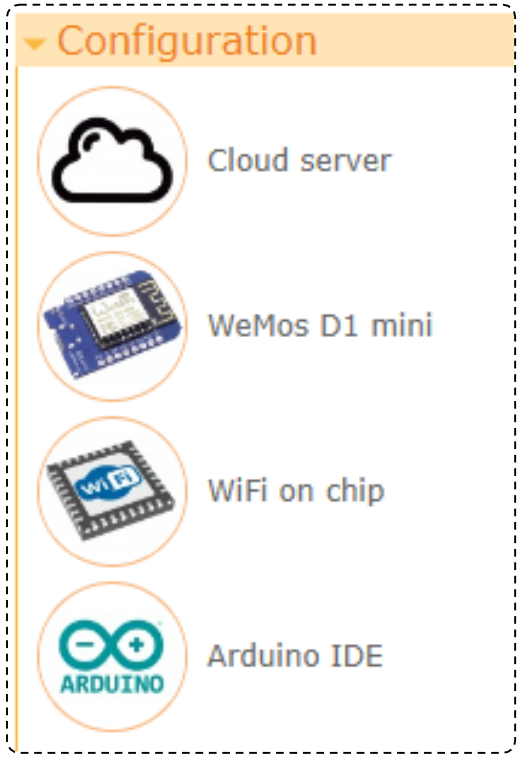
Lien: https://remotexy.com/en/account/tokens/

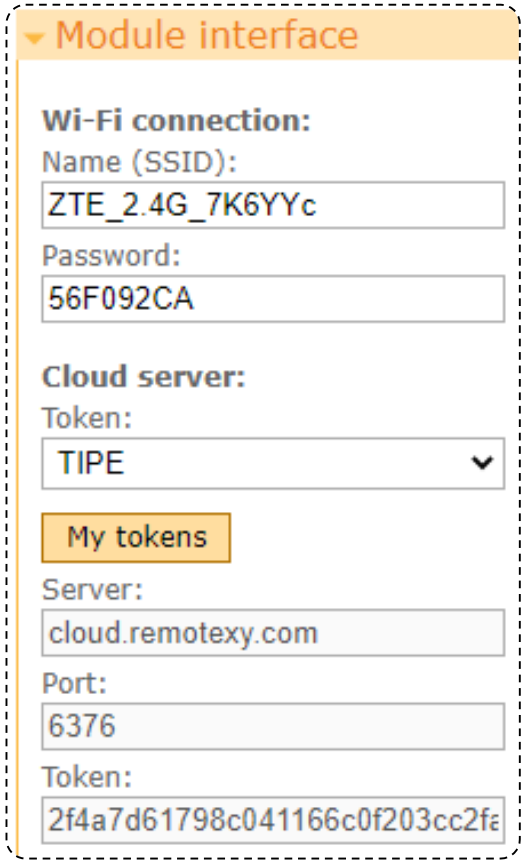
Create new token

№	Board name	Token	Device state	Server	Device port	App port	Actions
1	TIPE	2f4a7d61798c041166c0f203cc2fa1b7	disconnected	cloud.remotexy.com	6376	6375	Edit Delete

8 Parking connecté: conception d'une application mobile

❑ *Configuration logicielle et matérielle*





❑ *Bibliothèque et application ?*

✓ Bibliothèque Arduino

Bibliothèque de communication série et de remotxy sont disponible dans le site web remotxy.com

```
#include <SoftwareSerial.h>
```

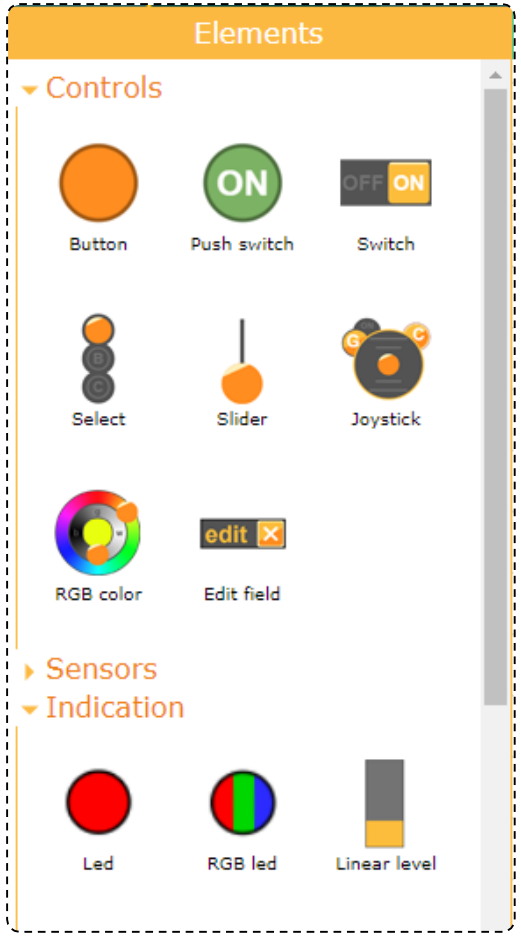
✓ Application mobile

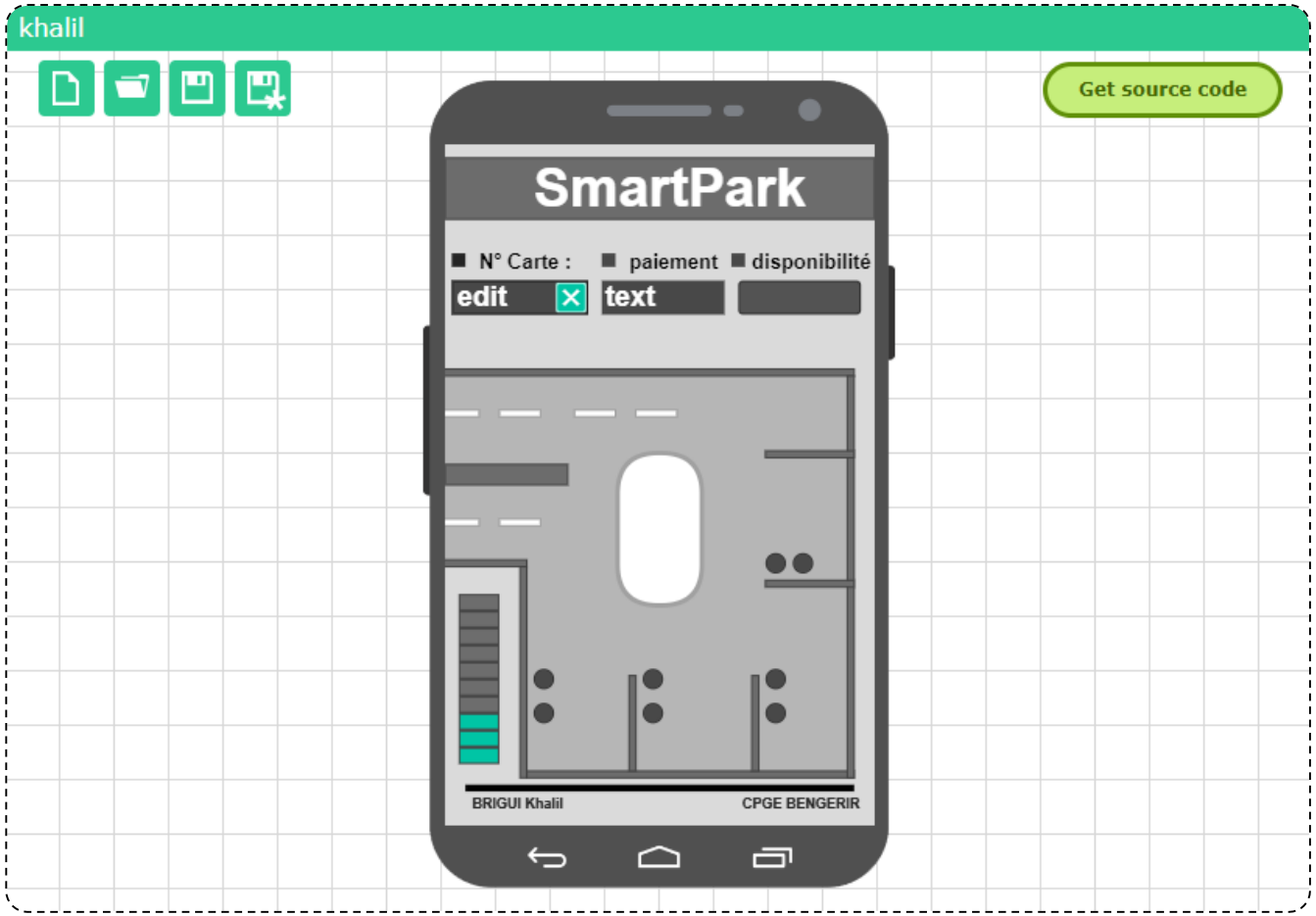
Application est disponible en Apple store et Google Play.

Prix : 80 Dhm

8 Parking connecté: conception d'une application mobile

❑ *Éditeur Remotexy*





8 Parking connecté: conception d'une application mobile

❑ *Description de l'application réalisée*

❑ *N° de la carte*

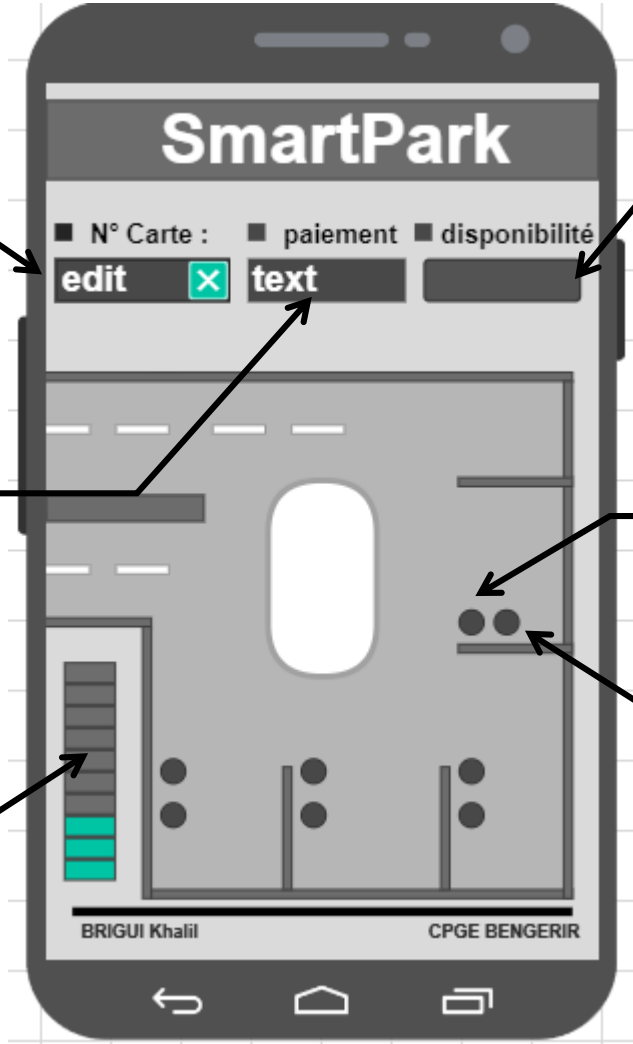
Un code est fournit avec la carte

❑ *Paiement*

Le prix de stationnement

❑ *État de parking*

Le niveau de remplissage de parking



❑ *Disponibilité*

- *Verte : parking ouvert*
- *Rouge : parking fermé*

❑ *LED Verte*

l'emplacement est vide

❑ *LED rouge*

l'emplacement est occupé

Conclusion

Pour conclure. Ce sujet de TIPE a été très enrichissant pour moi, car il m'a permis de découvrir plusieurs domaines, ses acteurs, contraintes. Il m'a permis de participer concrètement à ses enjeux. Mon sujet de TIPE m'a aussi permis de découvrir le domaine de l'informatique pratique et sa relation avec la programmation des carte électroniques programmables et ainsi, il ma permet d'acquérir des bonnes méthodes de recherche et bien aussi la communication avec nos profs pour récupérer les informations utiles.

MERCI POUR VOTRE ATTENTION