

Résumé

Depuis quelques années, les solutions de domotique se démocratisent de plus en plus. En effet, à une époque où il devient important de maîtriser sa consommation énergétique et où presque tout le monde dispose d'un Smartphone, il est tentant de pouvoir commander ses équipements domestiques (éclairage, chauffage, stores, etc.) et de consulter l'état de l'habitation (température, humidité, etc.) via un ordinateur ou un appareil mobile. Même s'il existe bien sûr déjà des solutions commerciales « clés en main », l'objectif de ce projet est de créer un système de ce type à partir de zéro. Partant de la découverte des différentes composantes technologiques, passant par l'étude la conception méthodologique du système et finissant par la mise en œuvre et la réalisation d'un prototype simulant les différentes scénarios et fonctionnalités conçus.

Mots-clés: domotique, smart home, Arduino, Android, développement, programme informatique, composante électronique, programmation...

Abstract

In recent years, the solutions of home automation have become more and more democratic. Indeed, at a time when it is important to control its energy consumption and where almost everyone has a smartphone, it is tempting to be able to control its domestic equipment (lighting, heating, blinds, temperature, humidity, etc.) via a computer or mobile device. Even if there are of course already "turnkey" commercial solutions, the aim of this project is to create a system of this type from scratch. Starting with the discovery of the various technological components, going through the study methodological design of the system and ending with the implementation and realization of a prototype simulating the various scenarios and functionalities designed.

Key-words: home automation, smart home, Arduino, Android, development, computer program, electronic component, programming...

Table des matières

Résumé	1
Abstract	1
Table des abréviations.....	5
Introduction.....	6
[Présentation générale du projet].....	8
SmartHome ! C'est qui ?.....	9
Notre perception du SmartHome.....	9
Les problèmes rencontrés	9
[Analyse et conception].....	10
Diagramme de cas d'utilisation	10
Diagramme de classe.....	12
[Outils de réalisation]	13
Présentation des composantes électroniques	14
Carte Arduino Uno.....	14
Présentation	14
Arduino UNO	14
LED.....	15
a. Présentation	15
b. Schéma de câblage :	15
c. Programmation :	16
Clavier matriciel.....	16
a. Présentation	16
b. Programmation	17
Micro Servo (ServoMoteur) :	18
a. Présentation :	18
b. Câblage	19
c. Programmation	19
Détecteur de mouvement	20
a. Présentation	20

b. Brochage et schéma de branchement	21
c. Programmation	22
RFID Card Reader/Detector Module Kit	24
a. Présentation :	24
b. Le module RFID MFRC522 :	24
c. Le Brochage:	24
d. Schéma de câblage	25
a. Présentation :	25
Brochage :	25
Schéma de câblage :	26
Programmation :	26
Bluetooth.....	27
a. Présentation du Bluetooth	27
b. Présentation du module HC-05	28
c. Schéma du câblage	29
d. Programmation	29
Présentation des spécificités logicielles	30
Arduino IDE.....	30
a. Présentation	30
b. Logiciel.....	31
c. Langage.....	31
Android Studio.....	32
a. Logiciel.....	32
b. Android SDK.....	32
Les activités formant l'application Mobile	33
[Domotique : Scénarios et fonctionnalités]	34
La gestion de la porte	34
Présentation	35
Schéma de câblage	35
Séquencement de l'ouverture de la porte	36
Explications :	36
Déroulement des activités liées à l'ouverture de la porte	37
L'ouverture du garage	37
Présentation	37

Schéma du câblage	38
Séquencement de l'ouverture du garage.....	39
Explication	39
Déroulement des activités liées à l'ouverture du garage	40
Contrôle de la lumière d'une pièce par détection de mouvement.....	40
Présentation	40
Séquencement du contrôle de la lumière.....	41
Explication	42
Déroulement des activités du contrôle de la lumière.....	42
Consultation de l'humidité et de la température	42
Séquencement de la consultation.....	42
Explication :	44
[L'Application Mobile]	45
Les activités	46
[Perspectives d'évolutivité]	47
Récolte et Analyse des données.....	47
Performance de la sécurité	48
Développement de la capacité de communication.....	49
Optimisation des canaux de communication.....	50
Diversification des supports de connexion	51
Conclusion & Problématiques	52
Remerciement	53
Références.....	54

Table des abréviations

Abréviation	Signification
ADT	Android Development Tools
DEL	Diode Electroluminescente
DHT	Digital Humidity & Temperature sensor
FSTM	Faculté des sciences et techniques de Mohammedia
GND	Ground
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
ILISI	Ingénierie Logicielle et Intégration des Systèmes Informatiques
LED	Light-Emitting Diode
NTIC	Nouvelles Technologies d'Information et de Communication
OS	Operating System
PIR	Pyroelectric -Passive- InfraRed sensors
RFID	Radio frequency identification
RXD	Data to be Received over the comms link
SDK	Software Development Kit
SPP	Serial Port Protocol
TXD	Data to be Transmitted over the comms link
UART	Universal Asynchronous Receiver Transmitter
UHII	Université Hassan II
UML	Unified Modeling Language
USB	Universal Serial Bus
VCC	Positive supply voltage
Wifi	Wireless Fidelity

Introduction

Dans le cadre des mini-projets, chaque étudiant doit participer à la réalisation d'un projet. Nous avons choisi entre plusieurs thématiques et avons ensuite former des groupes pour chaque thème. Nous avons entrepris de travailler sur la domotique.

La domotique regroupe les technologies de l'électronique, de l'automatique, de l'informatique et des télécommunications permettant d'améliorer le confort, la sécurité, la communication et la gestion d'énergie d'une maison.

En effet, la domotique facilite notre vie en permettant de gérer des fonctions tout aussi infinies telles que l'assurance de la sécurité, l'automatisation des tâches, la programmation des alarmes, des détecteurs de mouvement et outils de communication grâce aux commandes à distance.

Notre maquette permet de présenter certaines fonctionnalités de la domotique à travers 3 scénarios :

- La sécurité et l'automatisation de l'ouverture des portes
- L'éclairage des lampes a distance
- S'informer sur la température et alarme lors d'un incendie.

Ces scénarios seront automatisés via des cartes «ARDUINO» exécutant des programmes informatiques, tout en ayant une interaction séquentielle avec des applications « ANDROID ».

Partie I :

[Présentation générale du projet]

Dans cette partie nous allons présenter les notions de domotique, maison intelligente, maison communicante... en relation avec les exigences et les défis des styles de vies contemporains face aux opportunités offertes par la révolution des nouvelles technologies d'information et de communication.

SmartHome ! C'est qui ?

Les systèmes de contrôle en maison intelligent permettent la gestion d'une quantité d'activités dans la maison dont l'éclairage le chauffage le système de surveillance et plein d'autres choses. Ce cerveau dédié à la gestion de la maison rend la vie plus simple et plus agréable pour ses occupants.

Notre perception du SmartHome

Pour ce projet, nous allons exploiter nos connaissances en programmation mobile et utiliser différentes fonctionnalités de l'Arduino pour mettre en œuvre un prototype d'une maison intelligente avec des activités contrôlables par une application mobile.

Les services que nous voulons fournir sont les suivants :

- Système d'éclairage contrôlé par l'application
- Système de détection de température et déclencheur d'alarme lors d'une détection de température élevée qui peut indiquer un incendie
- Système d'ouverture de porte à l'aide de carte RFID ou d'un clavier matricielle.
 - Système de consommation d'énergie de lampes

Les problèmes rencontrés

Parmi les problèmes qu'on a rencontrés lors de la réalisation du projet :

- La difficulté de réalisation du branchement des composants à cause du grand nombre de câbles qui se détachent facilement des pins.
- Nous étions obligés de manipuler trois cartes d'arduino vu que chaque composant nécessite un nombre intéressant de pins pour son branchement et que nous avons eu des problèmes d'interférence entre des composantes sur la même carte.
- Le branchement des composants séparés sur trois cartes nous a privé de quelques fonctionnalités qu'on a voulu mettre en œuvre par exemple l'alerte d'intrusion.
- Les composantes fournis avec le kit arduino débutant ne sont pas très précises.
- Le détecteur de mouvement a perdu sa précision et son fonctionnement lors de la réalisation de ce projet.

Partie II :

[Analyse et conception]

Dans cette partie nous allons présenter les conceptions du système dans sa globalité, sous forme des diagrammes UML, à travers lesquelles nous allons procéder par la suite à la mise en œuvre de notre système.

Diagramme de cas d'utilisation

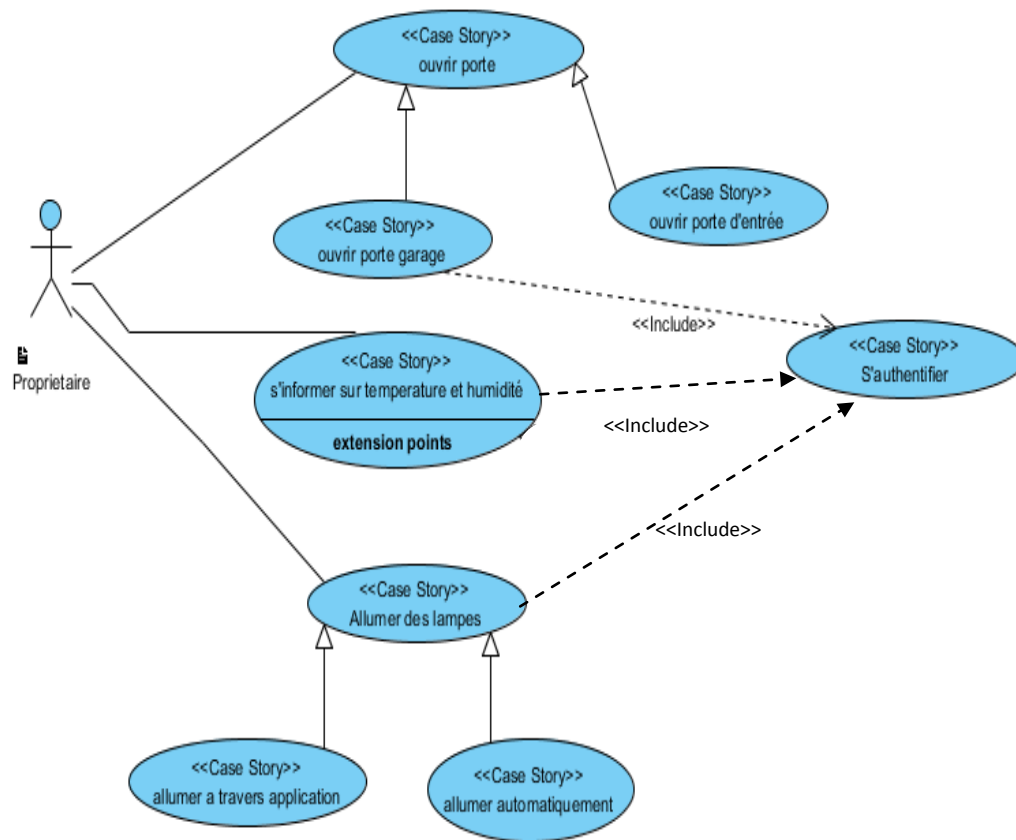


Diagramme de classe

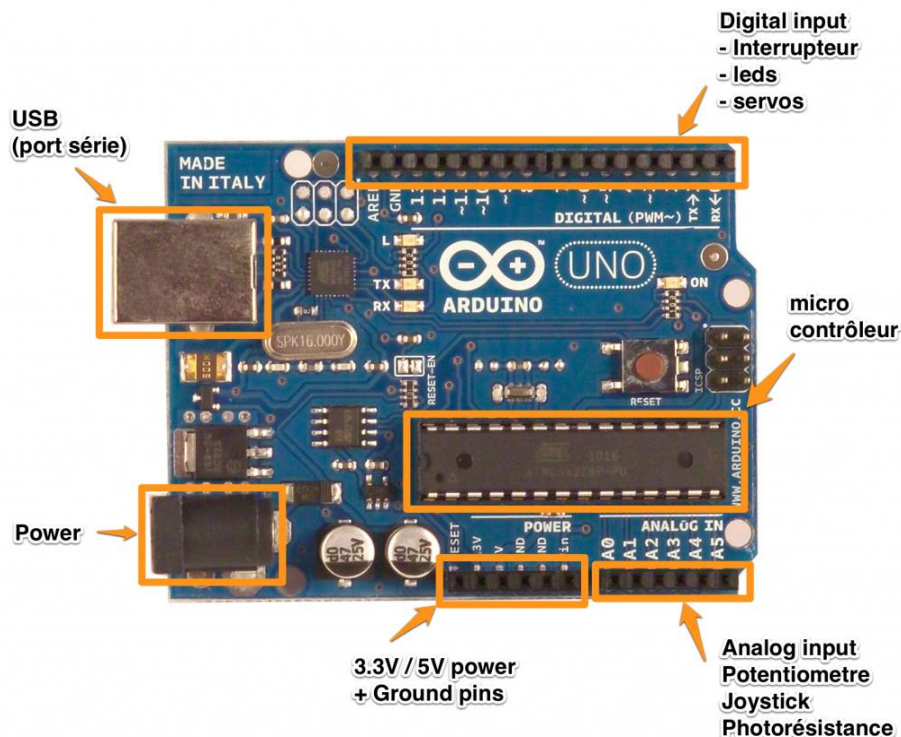
Partie III :

[Outils de réalisation]

Dans cette partie nous allons présenter les outils participants à la réalisation de notre projet, en commençant par l'exposition des différentes composantes électroniques, passant par leurs câblage et finissant par les composantes logicielles qui ont permis la compilation, la simulation ainsi que l'émulation de certains sous-systèmes.

Présentation des composants électroniques

Carte



Arduino Uno

Présentation

Arduino, et son récent synonyme Genuino3, est une marque qui couvre des cartes matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme l'Atmega328p, et d'architecture ARM comme le Cortex-M3 pour l'Arduino Due). Les schémas de ces cartes sont publiés en licence libre. Cependant, certains composants, comme le microcontrôleur par exemple, ne sont pas sous licence libre.

Le microcontrôleur 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, de l'informatique embarquée, etc.

Arduino UNO

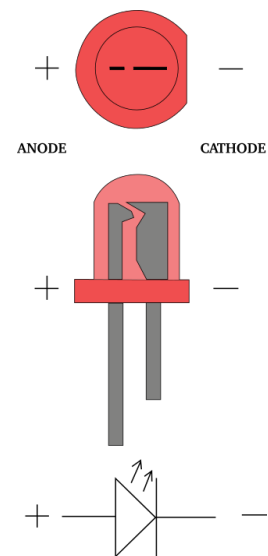
Arduino / Genuino Uno est un microcontrôleur basé sur ATmega328P (datasheet). Il dispose de 14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), de 6 entrées analogiques, d'un quartz de 16 MHz, d'une connexion USB, d'une prise de courant, d'un en-tête ICSP et d'un bouton de réinitialisation. Il contient tout le nécessaire

pour supporter le microcontrôleur; Il suffit de le connecter à un ordinateur avec un câble USB ou de le brancher avec un adaptateur AC-DC ou de la batterie pour commencer. "Uno" signifie un en italien et a été choisi pour marquer la sortie de Arduino Software (IDE) 1.0. La carte Uno et la version 1.0 d'Arduino Software (IDE) étaient les versions de référence d'Arduino, maintenant évoluées vers de nouvelles versions. La carte Uno est la première d'une série de cartes USB Arduino, et le modèle de référence pour la plate-forme.

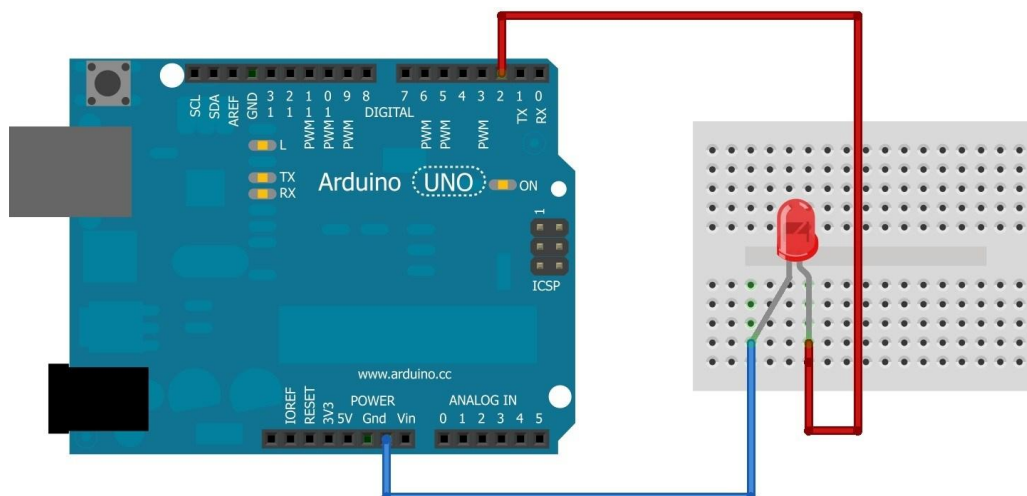
LED

a. Présentation

Une diode électroluminescente (abrégé en DEL en français, ou LED, de l'anglais : Light-Emitting Diode), est un dispositif opto-électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. Une diode électroluminescente ne laisse passer le courant électrique que dans un seul sens (le sens passant, comme une diode classique, l'inverse étant le sens bloquant) et produit un rayonnement à partir de la conversion d'énergie électrique lorsqu'un courant la traverse.



b. Schéma de câblage :



c. Programmation :

```
const int L1 = 2; // broche 2 du micro-contrôleur se nomme  
maintenant : L1  
void setup() //fonction d'initialisation de la carte  
{  
  //contenu de l'initialisation  
  pinMode(L1, OUTPUT); //L1 est une broche de sortie  
}  
void loop() //fonction principale, elle se répète (s'exécute) à  
l'infini  
{  
  //contenu du programme  
  digitalWrite(L1, HIGH); //allumer L1  
  delay(1000); // attendre 1 seconde  
  digitalWrite(L1, LOW); // Eteindre L1  
  delay(2000); // attendre 2 seconde  
}
```

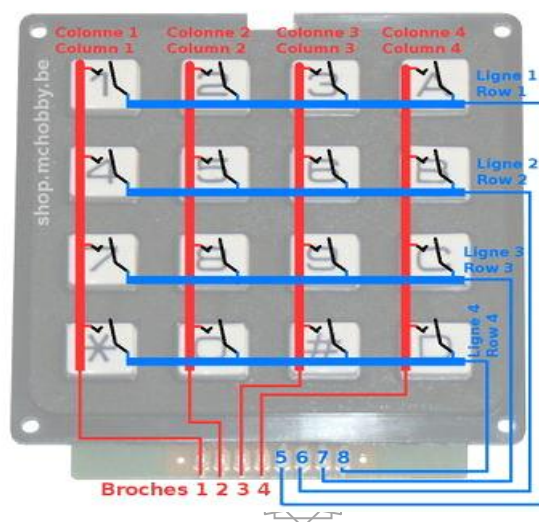
Clavier matriciel

a. Présentation

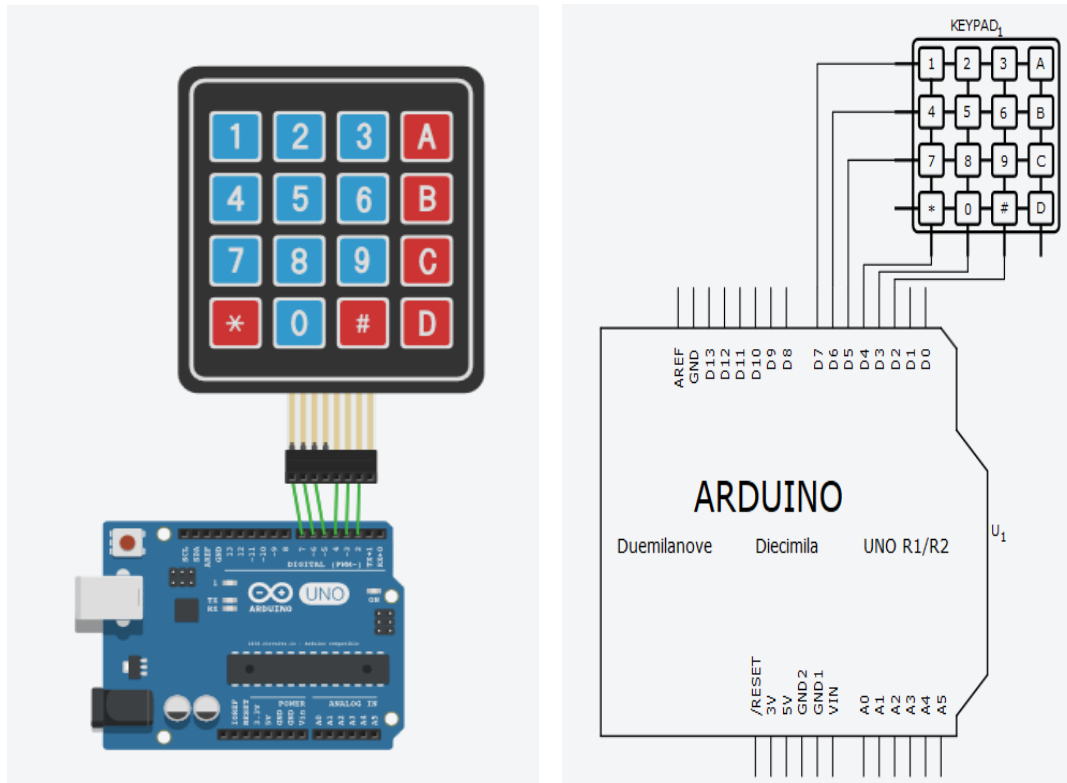
Un KeyPad ou Clavier matriciel est typiquement un clavier de saisie numérique, il existe sous différente forme, en clavier/membrane souple ou clavier solide (en ABS). Le plus courant ne comporte que des chiffres (de 0 à 9 avec les caractères * et #). Il en existe également quelques-uns peu plus évolués proposant une colonne supplémentaire pour que vous puissiez y placer vos propres fonctions.

Le principe de fonctionnement d'un KeyPad est excessivement simple, Un keypad n'est qu'une matrice de contact (de boutons). Quand un utilisateur appuie sur une touche il met une colonne en contact avec une ligne.

Par exemple dans le schéma ci-dessus, lorsque l'utilisateur presse la touche 6, il met en contact la ligne 2 et la colonne 3 Cela signifie que les broches 3 et 6 sont mises en contact.



Dans notre projet on a utilisé seulement les touches de 1 à 9 et on les a lié aux pins de notre Arduino du pin 2 au pin 7



b. Programmation

Pour instancier un objet keypad qui utilise les pins pour des lignes et comme colonne :

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

- **Keys** définit la touche pressée en fonction de la ligne et des colonnes tout comme leurs apparitions sur le clavier
- **rowPins** et **colPins** Code indiquant les connexions clavier aux terminaux arduino
- **numRows** et **numCols** indique le nombre de ligne et colonne qui seront connectées.

```
#include <Keypad.h>

const byte ROWS = 3; //3 Lignes
const byte COLS = 3; //3 Colonnes
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'}
};

byte rowPins[ROWS] = {7,6,5}; //Connecté sur les lignes du KeyPad (L1, L2, L3)
byte colPins[COLS] = {4,3,2}; //Connecté sur les colonnes dy KeyPad (C1, C2, C3)

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

- char getKey(): Cette fonction retourne le caractère qui correspond à la touche pressé. Ce code va permettre de lire et afficher une touche pressée

```
loop
{
  char key = keypad.getKey();
  Serial.println(key);
}
```

Micro Servo (ServoMoteur) :

a. Présentation :

Un servomoteur est un type de moteur électrique. C'est un dispositif typiquement utilisé en modélisme pour, il faut s'avoir qu'un servo permet d'effectuer des déplacements en translation ou en rotation. De nombreux objets du quotidien utilisent des servo moteurs: voitures télécommandées, drone, électroménager, ...

Dans notre projet on aura besoins d'un servomoteur pour ouvrir la porte après la saisie d'un code d'entrée sur un KeyPad et aussi pour ouvrir un garage après détection de la voiture qui va entrer dans le garage.

Les différents types de servos se distinguent par 3 caractéristiques:

- leur vitesse de rotation par seconde.
- leur course en degré.

- leur couple exprimé en kg.cm.

Le fonctionnement d'un servo est relativement simple. Il suffit d'envoyer une impulsion dont la durée est comprise entre 0,5ms et 2,5ms au servo. Cette impulsion déterminera la position du guide du servo. Ainsi la valeur de 1,5ms donne au servomoteur la position centrale.

Ces impulsions doivent être envoyées de manière périodique, généralement toutes les 50ms.

b. Câblage

Il n'y a rien de compliquer pour brancher un servo. Les servos ont généralement 3 fils:

- un rouge: qui doit être branché au 5V (ou 12V suivant votre servo)
- un noir : qui doit être branché à la masse
- un orange (ou autre couleur) : qui est le fil de tension de commande 0-5V

C'est donc via le fil orange que les commandes seront passées à votre servo. Ce fil devra obligatoire être relié à votre Arduino sur un pin PWN.

- 1.25 ms = 0 degré
- 1.50 ms = 90 degrés
- 1.75 ms = 180 degrés

c. Programmation

Pour contrôler un servoMoteur on doit créer un objet de type Servo.

```
#include <Servo.h>
Servo myservo; // Cree un objet Servo pour controler le servoMoteur
```

Puis l'attacher a le pin lie au fil de tension de commande

```
void setup()
{
  myservo.attach(8); // attacher le servo au pin 8 |
}
```

Dans ce programme on fait tourner le servo on passant du degré 0 au degré 180 et puis l'inverse.

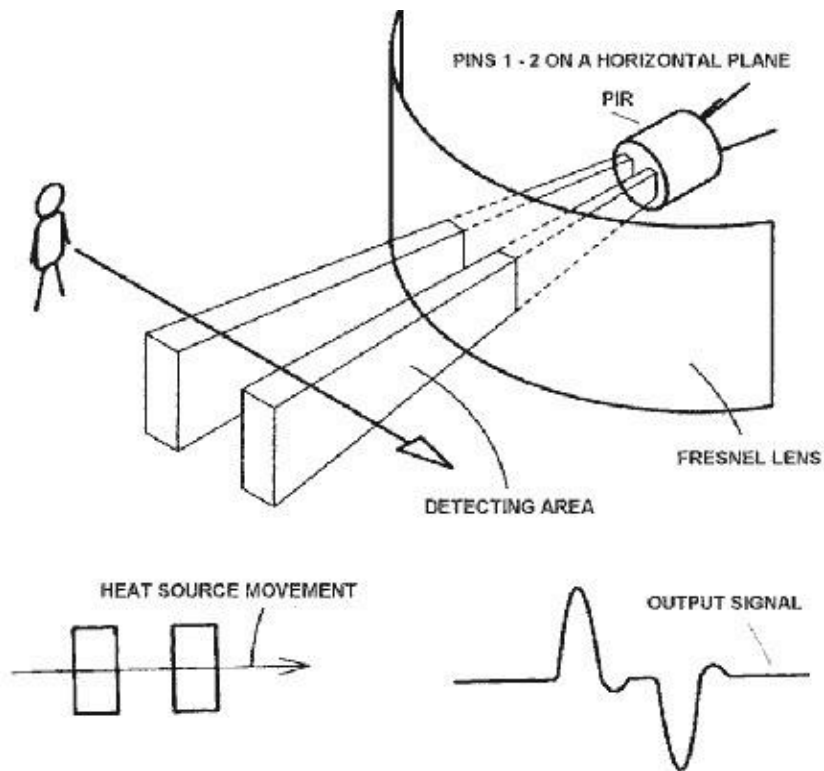
```
void loop()
{
  for(pos = 0; pos <= 180; pos += 1) // goes from 0 degrees to 180 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=0; pos-=1)    // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Détecteur de mouvement

a. Présentation

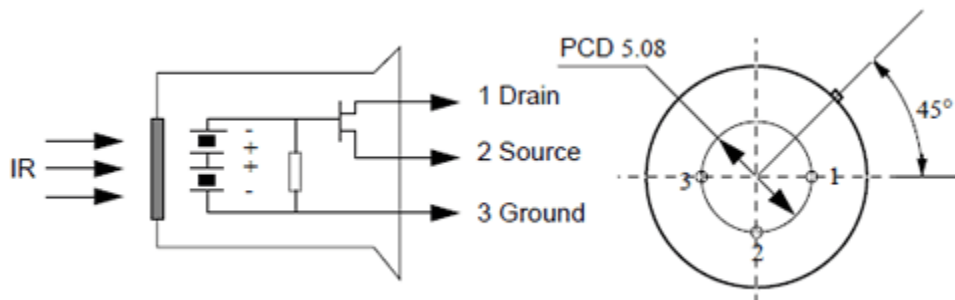
Un capteur infrarouge permet de détecter un mouvement dans son champ de vision en se basant sur l'infrarouge. Ici nous parlons de capteur pyroélectrique ou PIR. Le PIR sont capable de détecter une variation des ondes infrarouges, ce qui génère un courant électrique. Dans le cas de notre capteur, il est en fait divisé en deux partie différente reliées ensemble afin de détecter une variation lors qu'une des moitiés capte plus qu'une autre. On a ainsi un relevé d'une différence, et non plus d'une valeur simple.

Lors d'un mouvement, la variation des deux moitiés vont varier, et on va donc capter cette variation positive.



b. Brochage et schéma de branchement

Un capteur PIR à l'intérieur ressemble à l'image ci-dessous :



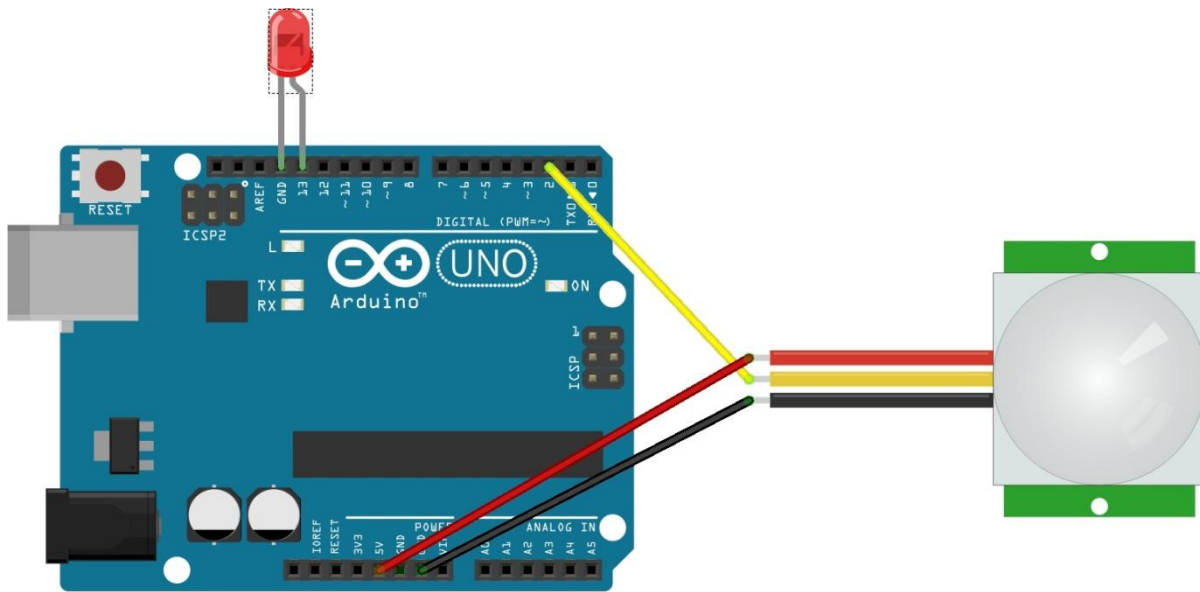
Ces spécifications techniques sont les suivantes :

- Entrée : Courant continue de 4.5 à 20V
- Sortie : High 3.3 V / Low 0V (Détection ou non)
- Angle : <100 °
- Dimension : 32 mm * 24 mm
- Délai : de 5 à 200 secondes (ajustable)
- Portée : de 3 à 7 mètres (ajustable).

- Au repos : 50 microampères.

Un branchement de simulation peut se faire alors de telle façon :

- Le VCC du Pir sur le 5V de l'arduino
- Le GRD du PIR sur le GRD de l'arduino
- La dernière branche sur le pin 2 de l'arduino



On peut ajouter une led de contrôle entre le pin 13 et un GRD de l'arduino

c. Programmation

Le programme consiste à commencer par l'initialisation des pins dans leurs états d'entrée ou de sortie, puis le calibrage du capteur.

Puis dans le loop, on relève la valeur renvoyée par le capteur et on agit en conséquence.

//the time we give the sensor to calibrate (10-60 secs according to the datasheet)

int calibrationTime = 30;

int ledPin = 13; // choose the pin for the LED

```

int inputPin = 2;           // choose the input pin (for PIR
sensor)
int pirState = LOW;        // we start, assuming no motion
detected
int val = 0;               // variable for reading the pin
status

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare sensor as input
  Serial.begin(9600);

  Serial.print("calibrating sensor ");
  for(int i = 0; i < calibrationTime; i++){
    Serial.print(".");
    delay(1000);
  }
}

void loop(){
  val = digitalRead(inputPin); // read input value
  Serial.println(val);
  if (val == HIGH) { // check if the input is HIGH
    digitalWrite(ledPin, HIGH); // turn LED ON
    delay(150);

    if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW); // turn LED OFF
    delay(300);
    if (pirState == HIGH){
      // we have just turned of
      Serial.println("Motion ended!");
      // We only want to print on the output change, not state
      pirState = LOW;
    }
  }
}
}

```

RFID Card Reader/Detector Module Kit

a. Présentation :

La Radio Frequency IDentification est aussi une technologie sans contact par radiofréquences. Elle permet une détection automatique avec des distances de lecture supérieures (de 10 à 200 m selon le type de puces) à celles de la NFC.

Elle permet également la détection de produits marqués en grande quantité, jusqu'à 200 par seconde et sans orientation directe vers le détecteur.

Nous trouvons dans notre quotidien de très nombreuses applications telles que :

- Le contrôle d'authentification et d'accès sur des lieux sécurisés ou payants (bureau, parking, passeport biométrique, télépéage autoroutier)
- La traçabilité des produits, le suivi de production, de colis, de chargements complets en camion, de containers
- La gestion intégrale d'une chaîne d'approvisionnement, les inventaires · La billetterie pour les spectacles ou les abonnements de transport commun etc.

Fiable, rapide, peu coûteuse, la RFID offre des avantages indéniables pour de nombreux secteurs économiques.

b. Le module RFID MFRC522 :

Un module RFID (Philips MFRC522) est monté sur une carte déjà câblée avec une antenne prête à l'emploi. Il permet d'identifier sans contact des puces RFID présentées à proximité placées dans une carte ou un badge.

c. Le Brochage:

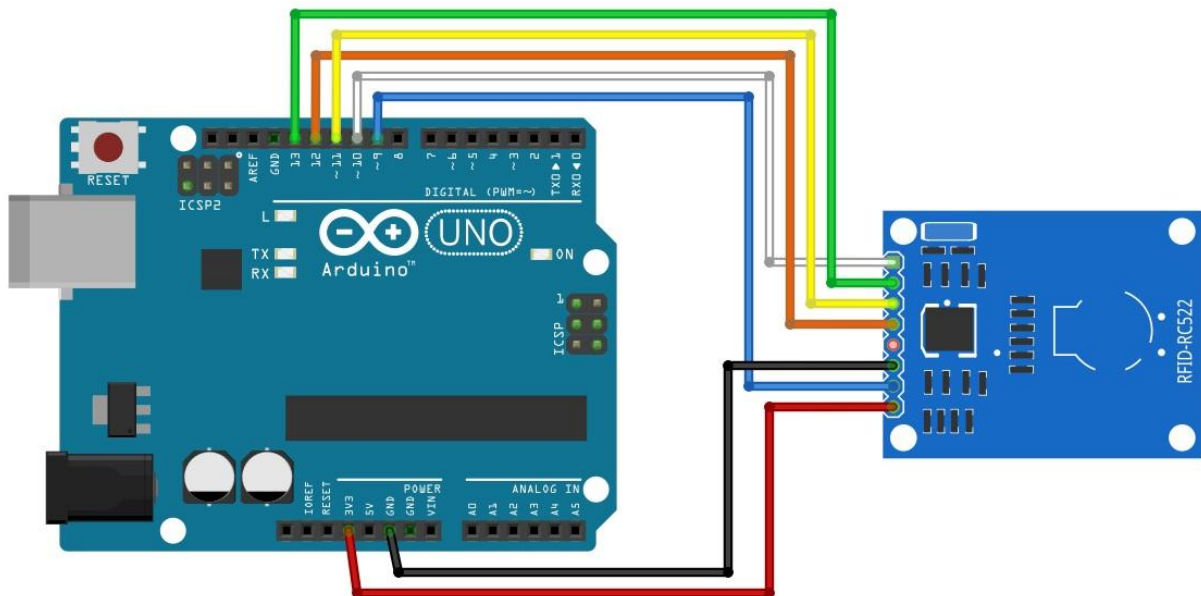
L'alimentation est de 3.3V (pas 5V), le lecteur utilise une interface SPI, et fonctionne sur une fréquence de 13.56 MHz (donc il lit tous les badges répondant à cette fréquence).

Le bus SPI se pilote par 4 signaux logiques :

- SCLK : Horloge. >> pin 13

- MOSI : Master Output, Slave Input. >> pin 11
- MISO : Master Input, Slave Output. >> pin 12
- SS : Slave Select, actif à l'état bas. >> pin 10

d. Schéma de câblage



Capteur de température et d'humidité

a. Présentation :

Le capteur DHT11 est un module arduino capable de mesurer des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/- 5%. Une mesure peut être réalisée toutes les secondes.

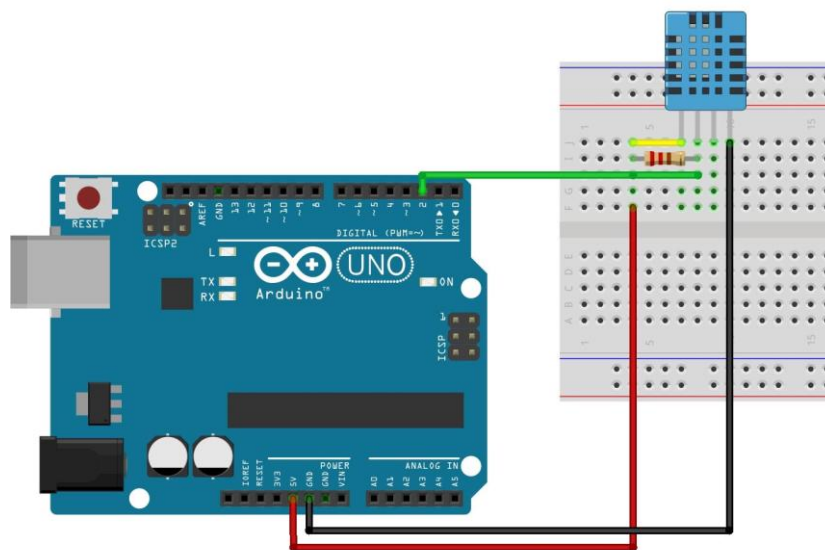
Brochage :

Le brochage du capteur DHT11 est le suivant :

- La broche n°1 est la broche d'alimentation (5 volts ou 3.3 volts).

- La broche n°2 est la broche de communication. Celle-ci doit impérativement être reliée à l'alimentation via une résistance de tirage de 4.7K ohms (il s'agit d'une sortie à collecteur ouvert).
- La broche n°3 n'est pas utilisée et ne doit pas être câblée.
- La broche n°4 est la masse du capteur (GND).

Schéma de câblage :



Programmation :

```
//importer la bibliothèque DHT11
//pour pouvoir manipuler le module
#include <DHT11.h>

/** Broche "DATA" du capteur */
const byte BROCHE_CAPTEUR = 5;

/* Code d'erreur de la fonction readDHT11() et readDHT22() */
const byte DHT_SUCCESS = 0;          // Pas d'erreur
const byte DHT_TIMEOUT_ERROR = 1;    // Temps d'attente dépassé
const byte DHT_CHECKSUM_ERROR = 2;   // Données reçues erronées

/** Fonction setup() */
void setup() {
```

```

/* Initialisation du port série */
Serial.begin(115200);
Serial.println(F("Demo DHT11"));

/* Place la broche du capteur en entrée avec pull-up */
pinMode(BROCHE_CAPTEUR, INPUT_PULLUP);
/*
  La mise en entrée de la broche n'est pas strictement nécessaire en soi.
  C'est juste une sécurité pour éviter de réveiller prématurément le
  capteur
  lors du démarrage de la carte Arduino.
  */
}

/** Fonction loop() */
void loop() {
  float temperature, humidity;

  /* Lecture de la température et de l'humidité, avec gestion des erreurs
  */
  switch (readDHT11(BROCHE_CAPTEUR, &temperature, &humidity)) {
    case DHT_SUCCESS:

      /* Affichage de la température et du taux d'humidité */
      Serial.print(F("Humidite (%): "));
      Serial.println(humidity, 2);
      Serial.print(F("Temperature (^C): "));
      Serial.println(temperature, 2);
      break;

    case DHT_TIMEOUT_ERROR:
      Serial.println(F("Pas de reponse !"));
      break;

    case DHT_CHECKSUM_ERROR:
      Serial.println(F("Pb de communication !"));
      break;
  }

  /* Pas plus d'une mesure par seconde */
  delay(1000);
}

```

Bluetooth



a. Présentation du Bluetooth

Le Bluetooth est un protocole de communication sans fil. Il a vu le jour à la fin des années 1990 et n'a vraiment percé que dans les années 2000. Il a subi de nombreuses révisions et évolutions pour atteindre aujourd'hui la version 4.1 depuis la fin 2013.

À l'origine, la société Ericsson cherchait à développer un moyen d'utiliser une voie série sans passer par un fil. Petit à petit, ses utilisations se sont étendues pour atteindre ce que l'on connaît aujourd'hui, un moyen de connecter sans fil de nombreux appareils.

Le protocole Bluetooth tout comme son cousin Le Wi-Fi, respectent tous deux une même spécification IEEE et utilisent la même gamme de fréquences : 2.4 GHz. Cela permet une communication bidirectionnelle, cela veut dire que deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est « maître/esclave ». Un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves. Pour son utilisation, elle se passe en plusieurs étapes :

1. Le maître se met en mode « reconnaissable »
2. L'esclave trouve le maître et demande à s'y connecter
3. Le maître accepte la connexion
4. Les périphériques sont alors appairés (ou associés)
5. La communication peut commencer

b. Présentation du module HC-05

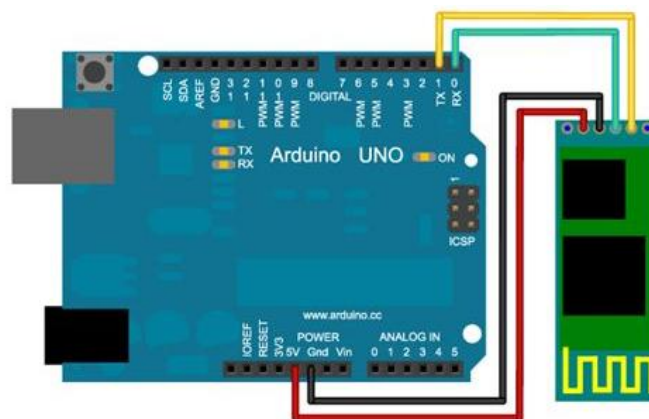
Le module HC-05 est un module Bluetooth SPP (Serial Port Protocol) facile à utiliser, conçu pour une connexion en série sans fil transparente. Le module Bluetooth HC-05 peut être utilisé en configuration maître ou esclave, ce qui est en fait une solution idéale pour les communications sans fil.

Le module HC-05 Bluetooth à 6 pins. Ils sont comme suit:

- **ENABLE:** Lorsque l'activation est activée LOW, le module est désactivé, ce qui signifie que le module ne s'allume pas et qu'il ne communique pas. Lorsque l'autorisation est laissée ouverte ou connectée à 3,3 V, le module est activé, le module reste allumé et la communication a également lieu.
- **VCC:** Tension d'alimentation 3.3V à 5V
- **GND:** Broche de masse.

- TXD & RXD: Ces deux broches servent d'interface UART pour la communication
- STATE: Il agit comme un indicateur d'état. Lorsque le module n'est pas connecté / couplé avec un autre périphérique Bluetooth, le signal passe Low(Bas). à cet état bas, le voyant clignote continuellement ce qui indique que le module n'est pas appairé avec d'autres modules .Quand ce module est connecté à / avec un autre périphérique bluetooth, le signal passe High(Elevé). à cet état élevé, la led clignote avec un retard constant par exemple 2s délai qui indique que le module est apparié.
- BUTTON SWITCH : Il permet de commuter le module en mode de commande AT. Pour activer le mode de commande AT, il faut appuyer sur le commutateur de bouton pour une seconde. Avec l'aide de commandes AT, l'utilisateur peut modifier les paramètres de ce module, mais seulement lorsque le module n'est pas apparié Avec tout autre périphérique BT. Si le module est connecté à tout autre périphérique Bluetooth, il commence à communiquer avec ce périphérique et ne fonctionne pas en mode de commande AT.

c. Schéma du câblage



d. Programmation

```
// //importer la bibliothèque SoftwareSerial
//pour pouvoir utiliser la liaison série
#include <SoftwareSerial.h>
/*
...
*/
//Déclarer l'utilisation du module
```

```

//HC05 à travers une liaison série
//11 et 10 sont les pins utilisé
//pour lié le Rx et le Tx du HC05
//à la carte Arduino
SoftwareSerial HC05(11,10);

//Déclarer une variable qui stockerait
//l'information transmise par le bluetooth
String messageRecu;

/*
...
*/
void setup() {
  //Initialisation des liaisons séries
  Serial.begin(9600);
  HC05.begin(9600);
  /*
  ...
  */
}

void loop()
{
  //La boucle consiste à detecter un message
  //provenant du HC05
  while(HC05.available())
  {
    //si il y en a un message
    //on doit le lire caractère par caractère
    //car c'est la méthode la plus fiable
    char c = HC05.read();
    messageRecu += c;
  }
  //En fonction du message reçu,
  //on execute le code convenable
  if (messageRecu == "XYZ")
  {
    /*
    ...
    */
  }
}

```

Présentation des spécificités logicielles

Arduino IDE

a. Présentation

Une équipe de développeurs composée de Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti a imaginé un projet répondant au doux

nom de Arduino et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques. Étant donné qu'il y a des débutants, commençons par voir un peu le vocabulaire commun propre au domaine de l'électronique et de l'informatique.

A screenshot of the Arduino IDE window titled "Arduino - 0011 Alpha". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for running, stopping, saving, and other functions. The main text area displays the "Blink" sketch. The code is as follows:

```
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);               // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);               // waits for a second
}
```

The bottom status bar shows "Done compiling." and "Binary sketch size: 1098 bytes (of a 14336 byte maximum)". The line number 22 is visible at the bottom left.

b. Logiciel

Le logiciel de programmation des modules Arduino est une application Java, libre et Multiplateforme, 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).

c. Langage

Le langage de programmation utilisé est le C++, compilé avec `avr-g++ 7`, 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++.

Android Studio



a. Logiciel

Android Studio est un environnement de développement pour développer des applications Android. Android Studio permet principalement d'éditer les fichiers Java et les fichiers de configuration d'une application Android. Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des écrans sur des écrans de résolutions variées simultanément.

b. Android SDK

Le kit de développement (SDK) d'Android est un ensemble complet d'outils de développement¹. Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriaux. Les plateformes de développement prises en charge par ce kit sont les distributions sous Noyau Linux, Mac OS X 10.5.8 ou plus, Windows XP ou version ultérieure. L'IDE officiellement supporté était Eclipse

combiné au plugin d'outils de développement d'Android (ADT), mais depuis 2015, Google officialise Android Studio qui devient alors l'IDE officiel pour le SDK Android. Les développeurs peuvent utiliser n'importe quel éditeur de texte pour modifier les fichiers Java et XML, puis utiliser les outils en ligne de commande (Java Development Kit et Apache Ant sont obligatoires) pour créer, construire et déboguer des applications Android ainsi que contrôler des périphériques Android (pour déclencher un redémarrage, installer un logiciel à distance ou autre).

Les activités formant l'application Mobile

Partie IV :

[Domotique : Scénarios et fonctionnalités]

Dans cette partie nous allons exposer les différents scénarios et fonctionnalités qu'un simple utilisateur peut les exploiter dans le système réalisé.

La gestion de la porte

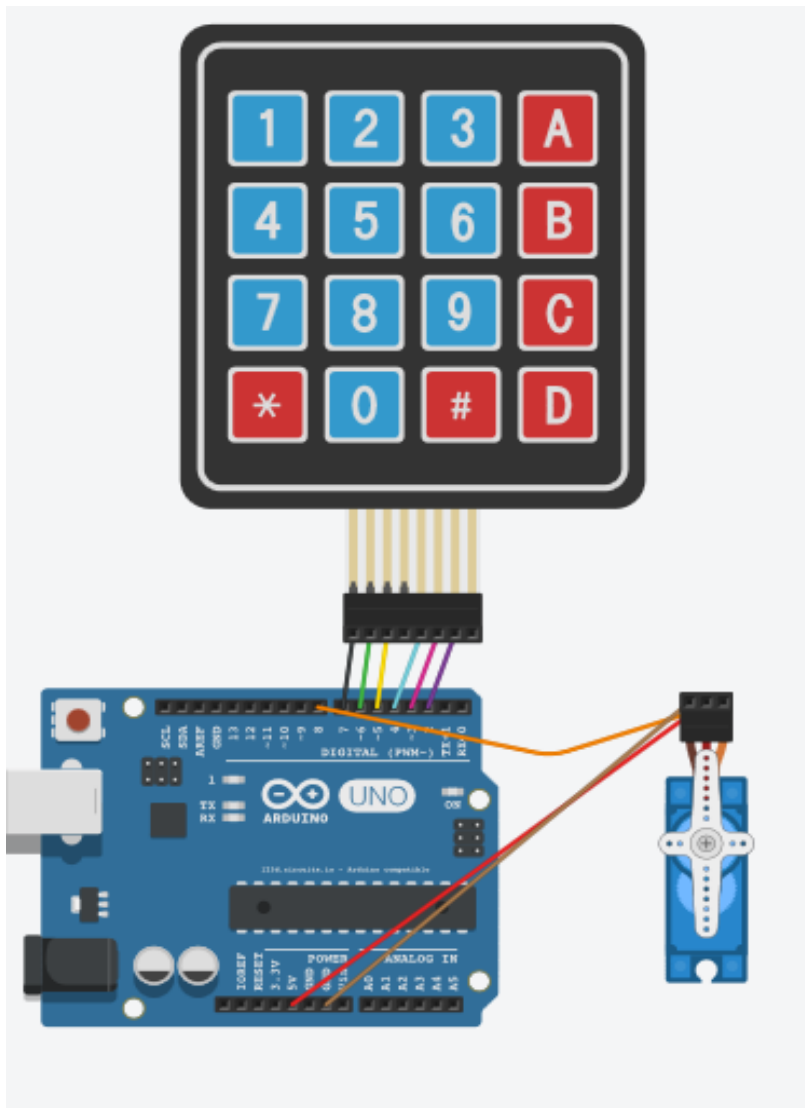
Présentation

L'utilisation des clés pour ouvrir la porte sera contradictoire avec l'aspect intelligent de notre maison, donc le premier scénario qu'on a opté a réaliser c'est l'ouverture d'une porte on utilisant un code saisie par l'utilisateur on appuyant sur un clavier de saisie, et dans le cas où le code saisie est incorrect on doit informer le propriétaire de cette tentative d'imbrication.

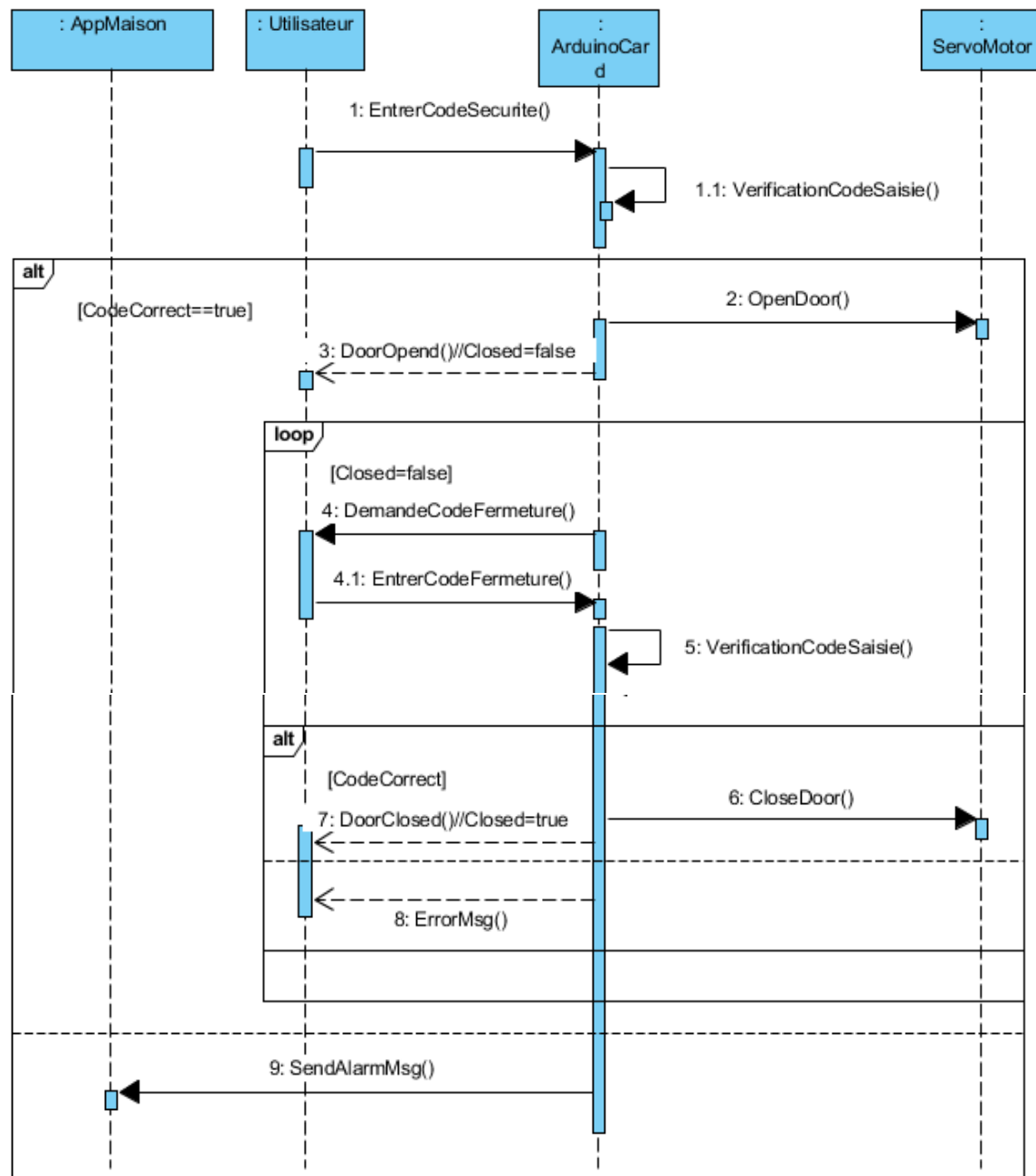
Le matériel que nous avons utilisé est le suivant:

- un Keypad comme clavier de saisie du code
- un servomoteur pour contrôler le mouvement de la porte

Schéma de câblage



Séquence de l'ouverture de la porte

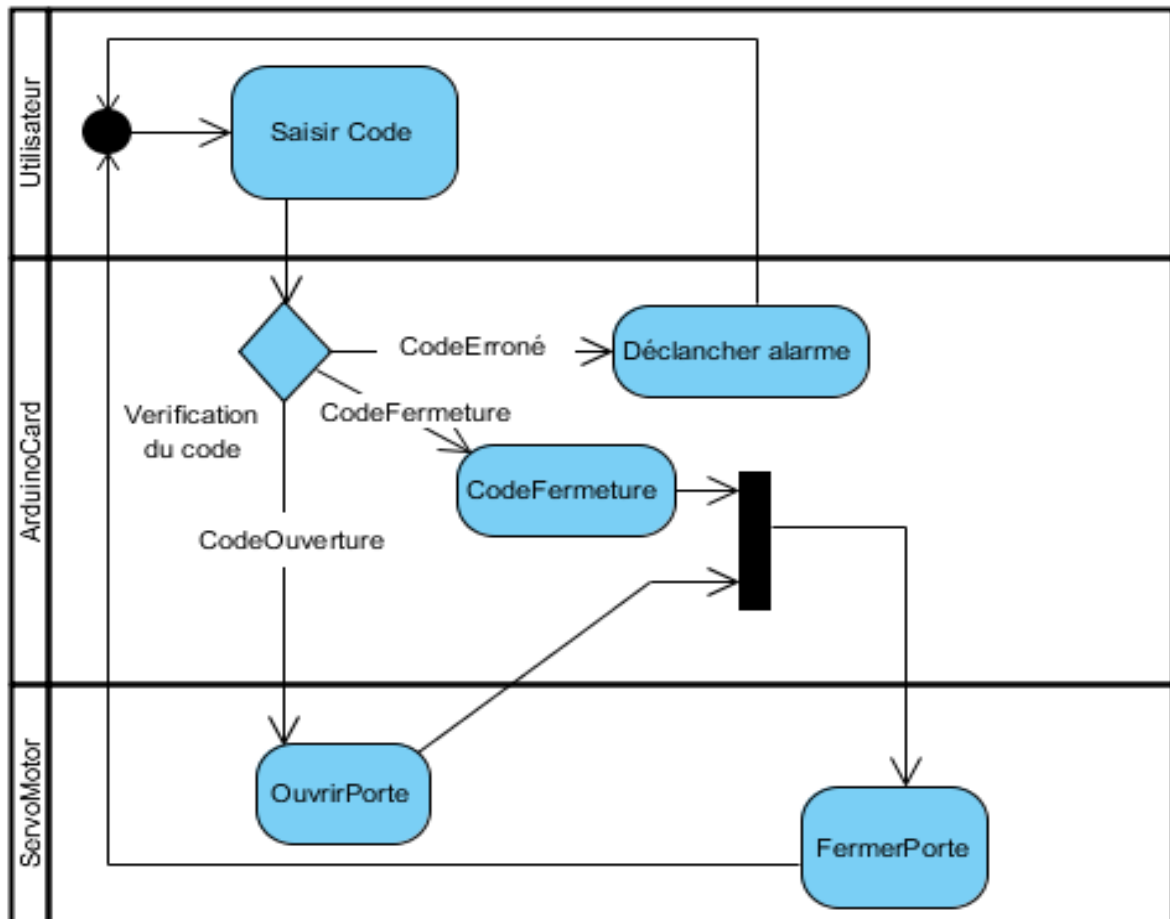


Explications :

L'utilisateur saisie un code sur la KeyPad ce code est envoyé au fur et a mesuré vers la carte Arduino. La carte vérifier le code saisie, si le code est correcte elle envoie une commande vers le servomoteur pour qu'il ouvre le porte, sinon il envoie un message d'alarme vers l'application de l'utilisateur.

Tant que la porte est ouverte la carte attend la commande de fermeture de la porte, quand l'utilisateur va saisir ce code la carte le vérifier et elle va envoyer un message de fermeture de porte vers le servomoteur.

Déroulement des activités liées à l'ouverture de la porte



L'ouverture du garage

Présentation

L'une des fonctionnalités qu'on trouve intéressante c'est l'ouverture du garage automatiquement l'hors de la présence de la voiture devant le garage, cette ouverture doit être bien sûr contrôlé on ne peut pas ouvrir le garage pour n'importe quelle voiture qui se trouve devant ce dernier, alors pour différencier on a utilisé la composante RFID.

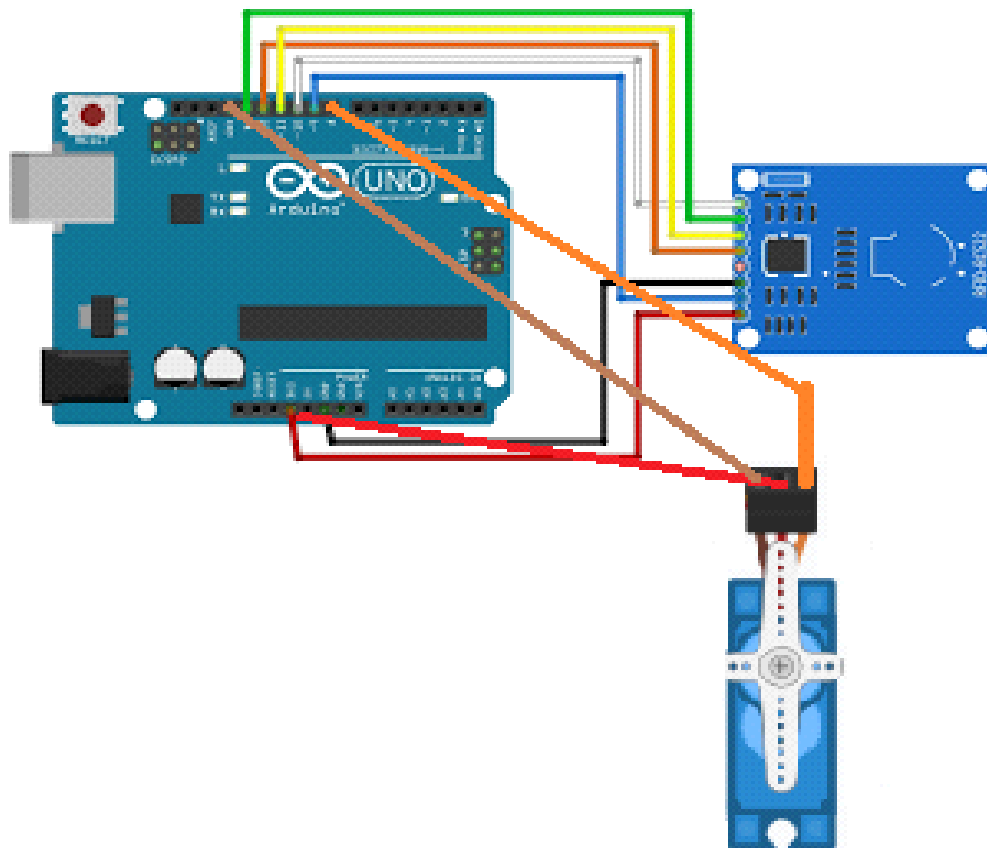
La voiture du propriétaire sera équipé par un tag RFID dont l'identifiant est unique, lors de la détection de ce tag par un lecteur RFID la porte sera automatiquement ouverte pour une durée

permettant au propriétaire d'entrer sa voiture facilement sans se mettre en dehors de cette dernière pour ouvrir son garage.

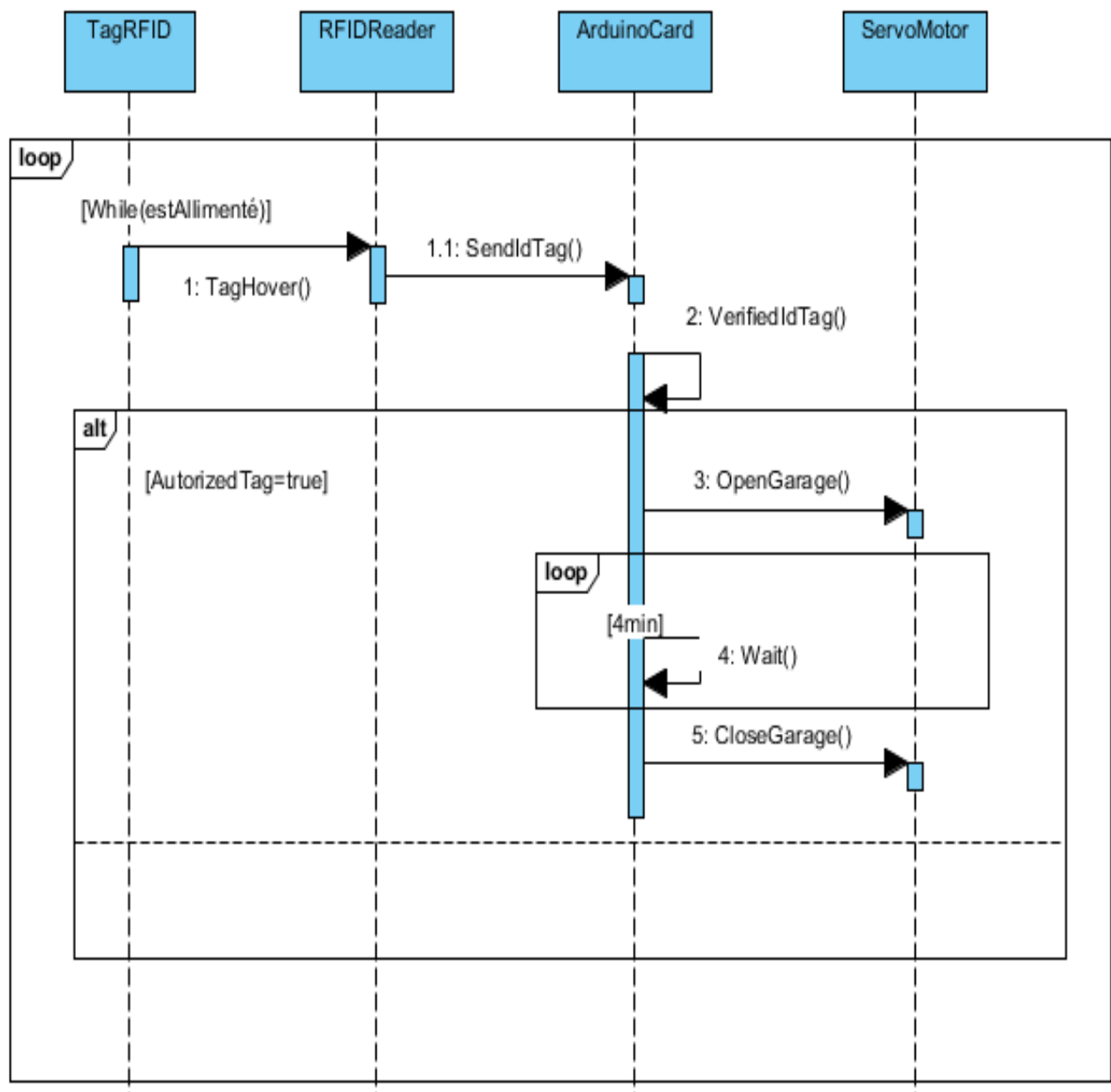
Le matériel que nous avons utilisé est le suivant:

- un Lecteur de tag RFID
- un ou des tags RFID qui sera clés du garage
- un servomoteur pour contrôler le mouvement de du garage

Schéma du câblage



Séquence de l'ouverture du garage

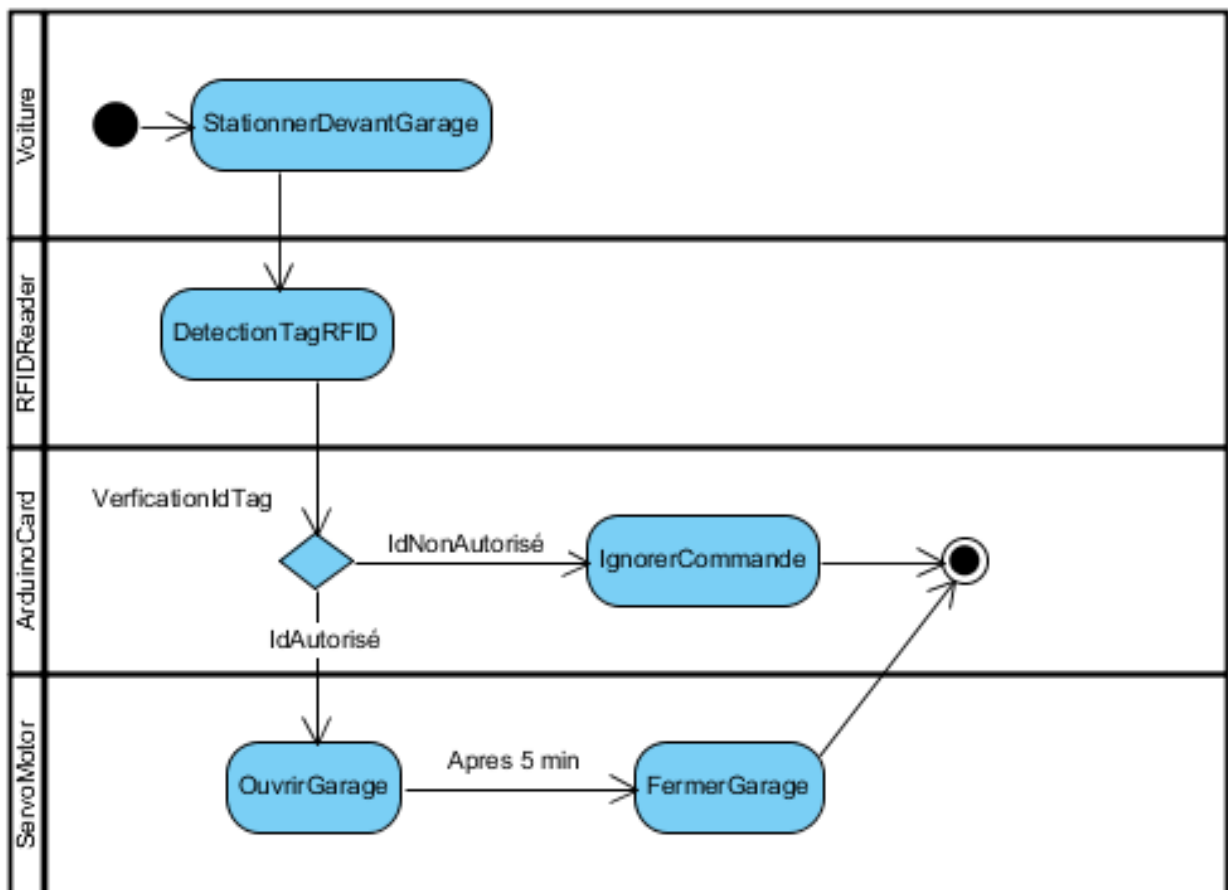


Explication

Tant que le système est alimenté, on attend toujours la lecture d'un tag par notre lecteur RFID qui est placé sur la porte du garage, si le lecteur a détecté un tag il envoie l'identifiant de ce tag vers la carte arduino, cette dernière va vérifier ce code, deux cas sont possibles. Soit l'identifiant correspond à un tag autorisé à l'accès du garage, dans ce cas on envoie une commande vers le servomoteur pour ouvrir la porte du garage puis on attend quelques minutes

qui seront suffisante pour que la voiture entre puis la porte sera fermer. Sinon si l'identifiant correspond à un tag non autoriser donc il sera ignoré.

Déroulement des activités liées à l'ouverture du garage



Contrôle de la lumière d'une pièce par détection de mouvement

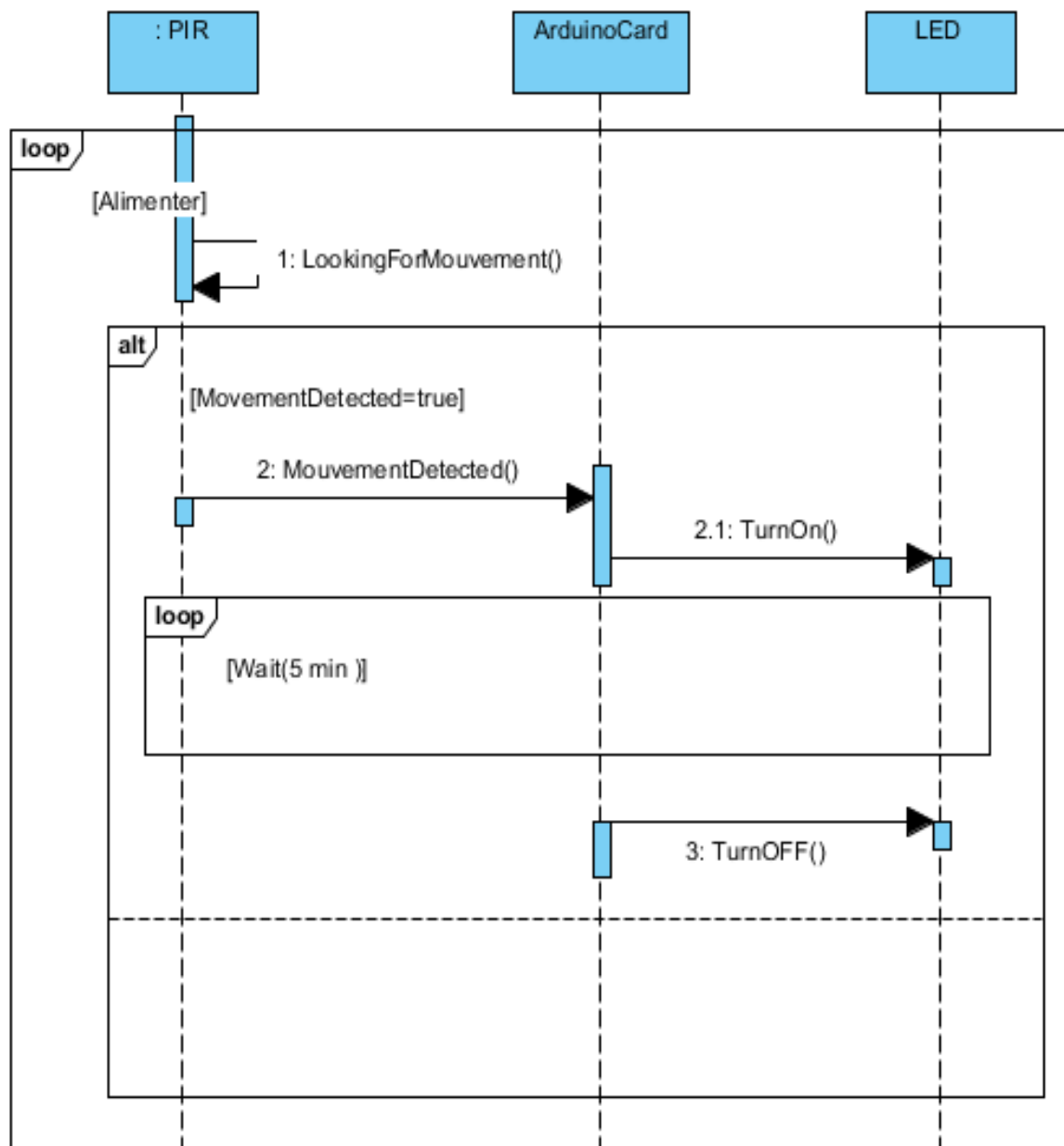
Présentation

La gestion de la consommation d'énergie est une tendance actuelle, c'est pourquoi nous avons modélisé un premier scénario qui respecte cette tendance à une échelle réduite. Notre dispositif permet d'allumer automatiquement la lumière en cas de présence dans une pièce mais permet aussi d'éteindre celle-ci en l'absence de mouvement pendant 5 min.

Le matériel que nous avons utilisé est le suivant :

- un capteur de Mouvement.
- une LED.

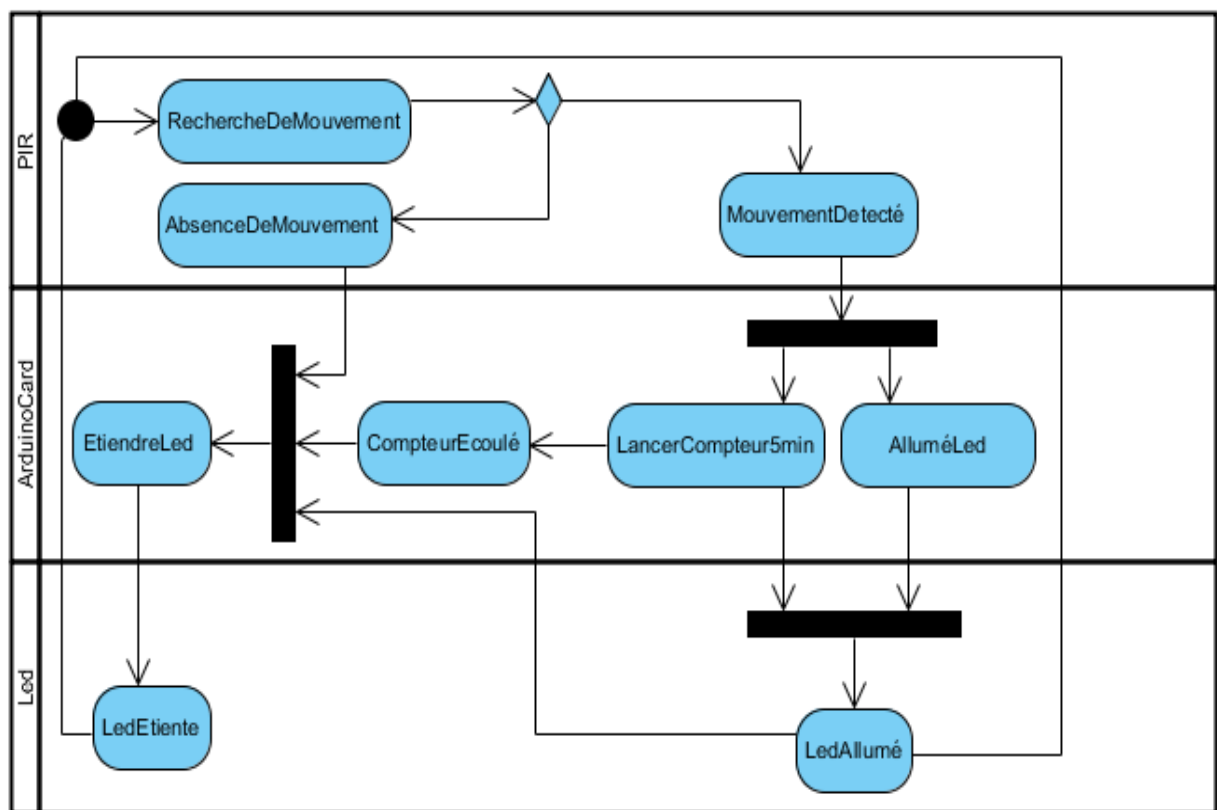
Séquence de contrôle de la lumière



Explication

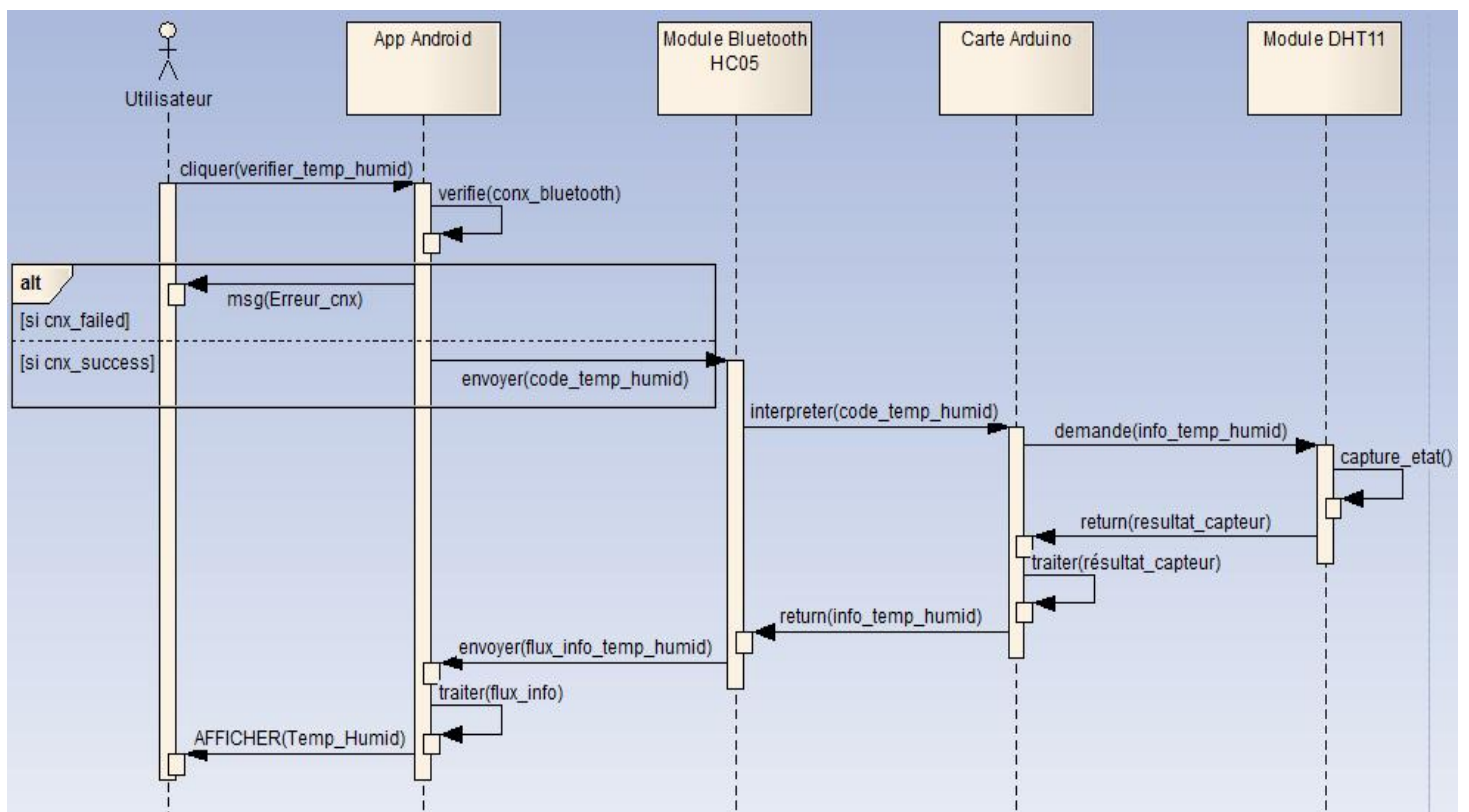
Le détecteur de mouvement est toujours en lecture de mouvement, lorsqu'il détecte la présence d'une personne il envoie un signal à l'arduino qui va interpréter ce signal en allumant la lumière pendant quelque minute puis après l'écoulement de cette période et si aucune mouvement a été détectée il envoie un signal pour éteindre la lumière.

Déroulement des activités du contrôle de la lumière



Consultation de l'humidité et de la température

Séquencement de la consultation



L'éclairage intelligent :

Afin de faciliter les taches au propriétaire, notre maquette lui offre la possibilité d'allumer ou d'éteindre les lampes de sa maison en un clic. Il suffit d'avoir une application Android installé sur son smartphone, et ce dernier soit connecté à la maison à l'aide du Bluetooth.

Le matériel que nous avons utilisé est le suivant :

- Un module Bluetooth HC-05.
- Des Leds.

Le Branchement :

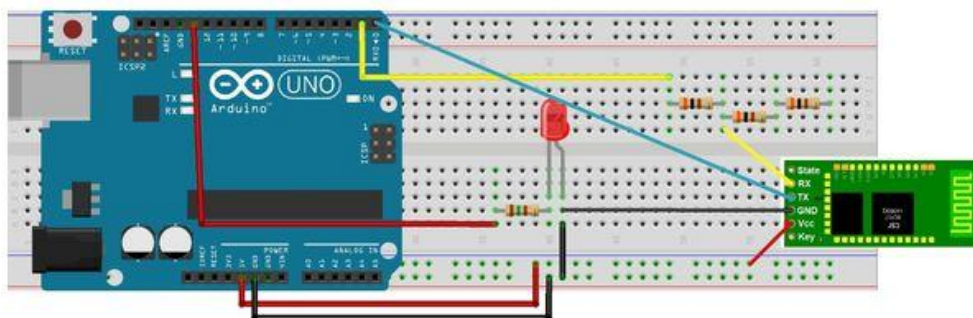
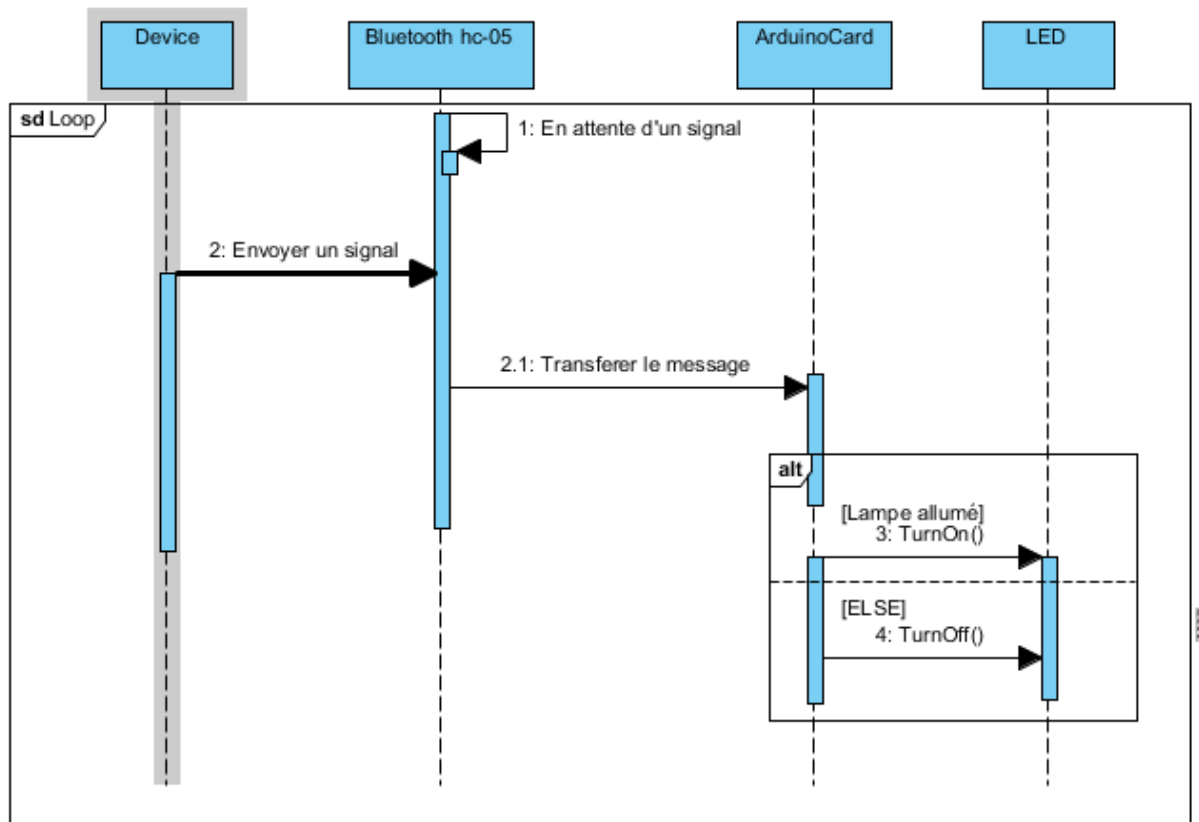


Diagramme de séquence :



Explication :

À l'aide d'un téléphone Android avec un apk de bluetooth spp, la commande est envoyée au bluetooth (RX/TX). Ce qui se passe, c'est que le module bluetooth communique avec bluetooth android en utilisant un profil appelé SPP (Serial Port Profile).

Une fois le module bluetooth reçoit le signal il le transmet à la carte arduino qui traduit le message en allumant ou éteignant la led concernée.

Partie V :

[L'Application Mobile]

Dans cette dernière partie nous allons exposer les spécificités techniques de l'application mobile développé dans le cadre du présent projet et qui permet le contrôle à distance d'un smart home.

Les activités

Partie VI :

[Perspectives d'évolutivité]

Dans cette dernière partie nous allons exposer les potentiels fonctionnalités et scénarios qu'ils puissent s'y ajouter à notre système qui vont apporter une valeur de plus.

Récolte et Analyse des données

L'analyse des données est une famille de méthodes statistiques dont les principales caractéristiques sont d'être multidimensionnelles et descriptives. Certaines méthodes, pour la plupart géométriques, aident à faire ressortir les relations pouvant exister entre les différentes données et à en tirer une information statistique qui permet de décrire de façon plus succincte les principales informations contenues dans ces données. D'autres techniques permettent de regrouper les données de façon à faire apparaître clairement ce qui les rend homogènes, et ainsi mieux les connaître.

Sur la base de ce type d'analyse et grâce à l'implémentation d'une base de données SQLi, notre application peut évoluer dans ce sens, en se permettant de récolter une quantité intéressante de données, et par la suite procéder à une analyse qui va permettre de traiter ces derniers et en dégager des aspects plus intéressants qui vont participer par la suite dans l'amélioration du confort et du niveau de vie de l'utilisateur.

Performance de la sécurité

Concernant la sécurité, de nombreux systèmes sont envisageables :

- une caméra vidéo à amplification de rayonnement et un code personnel pour contrôler et faciliter les entrées ;
- des détecteurs de présence qui, à l'approche d'un intrus, vont mettre en mouvement les fermetures de la maison et/ou simuler une présence (en allumant une lumière, par exemple) ;
- un « bip » d'ouverture automatique de la porte et éventuellement l'ouverture au moyen d'une carte magnétique ou d'un dispositif de reconnaissance vocale ;
- en cas de tentative d'intrusion, une synthèse vocale et un système de lumières peuvent être déclenchés ; un appel téléphonique automatique peut contacter alors le propriétaire et/ou une entreprise de sécurité.
- un système de sécurité assuré par des contacts normalement fermés qui se déconnectent lors d'une ouverture. Il est couplé avec des détecteurs de vibrations.

L'ensemble des contacts est alimenté par un courant de faible intensité.
L'ouverture du circuit est prise en compte par la centrale qui déclenche l'alarme ;

- d'autres systèmes tels que « tapis contact », radars de détection, etc.

Il s'agit en premier lieu des alarmes, mais également de la sécurité technique (incendie, fuite d'eau).

Les usages que l'on peut attendre de l'habitat communicant en termes d'alarme :

- savoir qu'il y a quelqu'un dans un bâtiment ;
- éventuellement, disposer des vidéos sur le bâtiment ;
- alerter si nécessaire (l'occupant, les forces de l'ordre).

La sécurité technique concerne en premier lieu la sécurité incendie. Les détecteurs de monoxyde de carbone sont également importants pour les chauffages avec combustion.

Les autres sécurités techniques (fuite d'eau, panne d'appareil comme un congélateur) sont encore peu développées.

Développement de la capacité de communication

Le concept d'habitat communicant correspond à l'introduction des nouvelles technologies (NTIC) dans l'habitat.

Historiquement, ce concept est une évolution du terme « domotique ». Le terme anglais est home automation.

Ce sujet peut être vu comme une application de l'internet des objets au domaine de l'habitat.

Le domaine du bâtiment est un prolongement naturel, ce qu'on appelle l'immotique : gestion technique de bâtiment.

Alors que la domotique était historiquement centrée sur les automatismes (ouvertures des volets, portails), le terme habitat communicant est davantage focalisé sur la communication (entre capteurs, avec l'utilisateur) et sur un grand usage de l'informatique. L'utilisation d'un terme

nouveau s'explique aussi par la connotation du terme domotique qui n'est pas très positive en français.

Les technologies internet sont aujourd'hui largement connues et maîtrisées pour mettre en place des usages innombrables dont quelques-uns sont listés ci-après, même si la diffusion de ces technologies est encore faible en 2012.

En ce qui concerne notre système et vu que nous avons déjà commencer à introduire les techniques de la reconnaissance vocale, notre smarthome peut devenir encore plus intelligente en la développant dans ce coté et lui donner encore plus de capacité de communiquer avec son environnement, cela par la suite aurait des impacts sur plusieurs champs, à savoir : l'économie d'énergie, l'environnement, le confort...

Optimisation des canaux de communication

Généralement, différents modes de transmission existent :

- Par onde radio hertzienne : Bluetooth, io-homecontrol, Zigbee, X2D, Wireless USB, EIB/Konnex ou KNX, Z-Wave18 et autres protocoles 433 MHz, 866 MHz, etc.
- Par infrarouge : RC5 Philips, SIRCS Sony, IrDA, etc. ;
- Par courant porteur secteur X10™, In One™ Legrand (entreprise), EIB/Konnex ou KNX , X2D DELTA DORE, WATTLET, etc. ;
- Par signatures numériques magnétiques ;
- Par réseau câblé Ethernet ou bus (Ethernet, TCP/IP, EIB/Konnex ou KNX , SCS BUS / OpenWebNet, USB, LonWorks, UPnP, RS485, Jbus, Batibus, Profibus, ModBus, I2C, 1-Wire, etc. (BUS = câblage basse tension de type Ethernet 8 fils, EIB 2 fils, I2C 2 fils, etc.).

Or, en ce qui concerne notre système nous pensons que les liaisons filaires (Ethernet, BUS, etc.) jouent en faveur de la stabilité du système.

Diversification des supports de connexion

La diversification des supports de connexion et des supports logiciels qui vont permettre à l'utilisateur une communication avec le smarthome consistent à garantir que les utilisateurs du système puissent continuer à profiter de la disponibilité de ce dernier et de ses composantes pour l'accomplissement de leurs tâches n'importe où et n'importe quand, et l'évolution des nouvelles technologies d'information et de communication aujourd'hui à travers particulièrement l'internet des objets peut largement garantir ça.

Actuellement notre système dispose d'une application mobile sur les systèmes Android, alors comme perspectives d'évolutivité, il est fort possible d'assurer une application web accessible partout et sur n'importe quel périphérique...

Conclusion & Problématiques

Tout comme notre vie en général, nos maisons se trouvent de plus en plus technologisées. Les habitats du futur répondent à une probable insatisfaction innée de l'homme qui croit augmenter son pouvoir sur les choses par la technique. On voit donc que sa maison « s'adapte » à lui et à ses besoins.

Au même titre que nous gérons notre budget, organisons notre temps et nos activités, que nous pourrions profiter des équipements perfectionnés de notre maison, la domotique nous offre la possibilité d'opter pour un habitat qui s'adaptera à notre rythme de vie et à nos habitudes, eux mêmes évolutifs au fil des années. Avec les nouvelles technologies domotiques, l'homme peut quitter son domicile pour le week-end et, avant de fermer la porte, une seule commande à travers son smartphone suffit pour éteindre les éclairages oubliés, mettre le chauffage en mode "économie", fermer les volets roulants, mettre en service l'alarme...

Ainsi, la domotique sert à la fois nos besoins, nos habitudes et notre envie de confort. Elle prend en compte des situations qui ont un sens dans notre vie quotidienne : quitter son domicile, créer une ambiance lecture, se réveiller dans un habitat chauffé où le café est déjà prêt et les volets ouverts.

Or, des interrogations majeures s'articulent autour de ce sens : Pourquoi, dans nos sociétés, les gens ont-ils un si grand désir de ce genre d'habitations, sensées nous faciliter la vie ? Est-ce une simple question de bien-être, de gain de temps, de liberté, ou encore de pouvoir, comme nous l'avons déjà suggéré ?

Ce nouvel univers technologique qui envahit et fait partie de notre quotidien pose un problème : la « maison du futur », toujours plus technologisée, deviendra-t-elle une maison toujours moins humanisée ? En effet, en disposant des téléphones dans chaque main, des webcams, des courriers électroniques, elle se veut une maison de communication. Pourtant, le paradoxe est saisissant : avec ses télévisions, ses ordinateurs, ses jeux vidéo, elle coupe les habitants d'une même maison les uns des autres. Qu'en est-il alors de la sociabilité, si importante dans nos sociétés ? Ne la mettrait-elle pas en danger ?

Remerciement

Eventuellement, notre culture nous a appris dès notre très jeune âge de laisser le plus beau pour la fin et à cette fin nous nous permettant qu'exprimer notre reconnaissance et offrir nos remerciements à tous ceux qui nous ont porté du soutien, qui nous ont corrigé nos erreurs ; à nos camarades de classe, nos professeurs et plus particulièrement notre cher encadrant...

Merci !

Références

<http://la.domotique.online.fr>

<https://developer.android.com>

<https://www.arduino.cc/en/Reference/HomePage>

<https://www.instructables.com/>

<https://fr.wikipedia.org/>

<https://www.youtube.com/channel/UCVqx3vXNghSqUcVg2nmegYA>

<https://www.tutorialspoint.com/android/>

<https://www.manuel-esteban.com>

<https://examples.javacodegeeks.com>

BERNARD Stéphane, Les conséquences sociales du progrès technique,