



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abdelhamid Ibn Badis de Mostaganem

Faculté des Sciences et de la Technologie



Mémoire de fin d'étude de master académique

Filière : Electronique

Spécialité : Electronique des systèmes embarqués

Commande automatique de l'éclairage public à base d'une carte Arduino Due et d'un capteur de mouvement

Présenté par :

SANOGO Lamoussa

MORTET Manel

Plan de travail

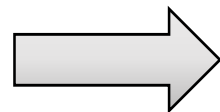
- **Introduction**
- **Chapitre I : Rappel sur les systèmes embarqués**
- **Chapitre II : Les microcontrôleurs**
- **Chapitre III : Arduino**
- **Chapitre IV : Système de commande automatique de l'éclairage public**
- **Conclusion**

Introduction

- Au 18^{ème} siècle, on découvre l'électricité qui vient remplacer les types anciens d'énergies.
- C'est le début de la révolution industrielle et par conséquent le début d'un changement radical dans le mode de vie de l'homme.

- ❖ L'homme dépend fortement de l'énergie.
- ❖ L'évolution de nos sociétés humaines est très étroitement liée à la découverte d'énergie

**Crise énergétique,
quelles solutions ?**



- ✓ Augmenter la production énergétique;
- ✓ Optimiser la consommation des équipements consommateurs;
- ✓ Economiser l'énergie;
- ✓ Etc...

CHAPITRE I

Rappel sur les systèmes embarqués

Définition d'un système embarqué

Un système embarqué est un système électronique (matériel) et informatique (logiciel) *qui est dédié à / spécialisé dans une tâche bien précise.*

Systeme embarqué

=

Matériel

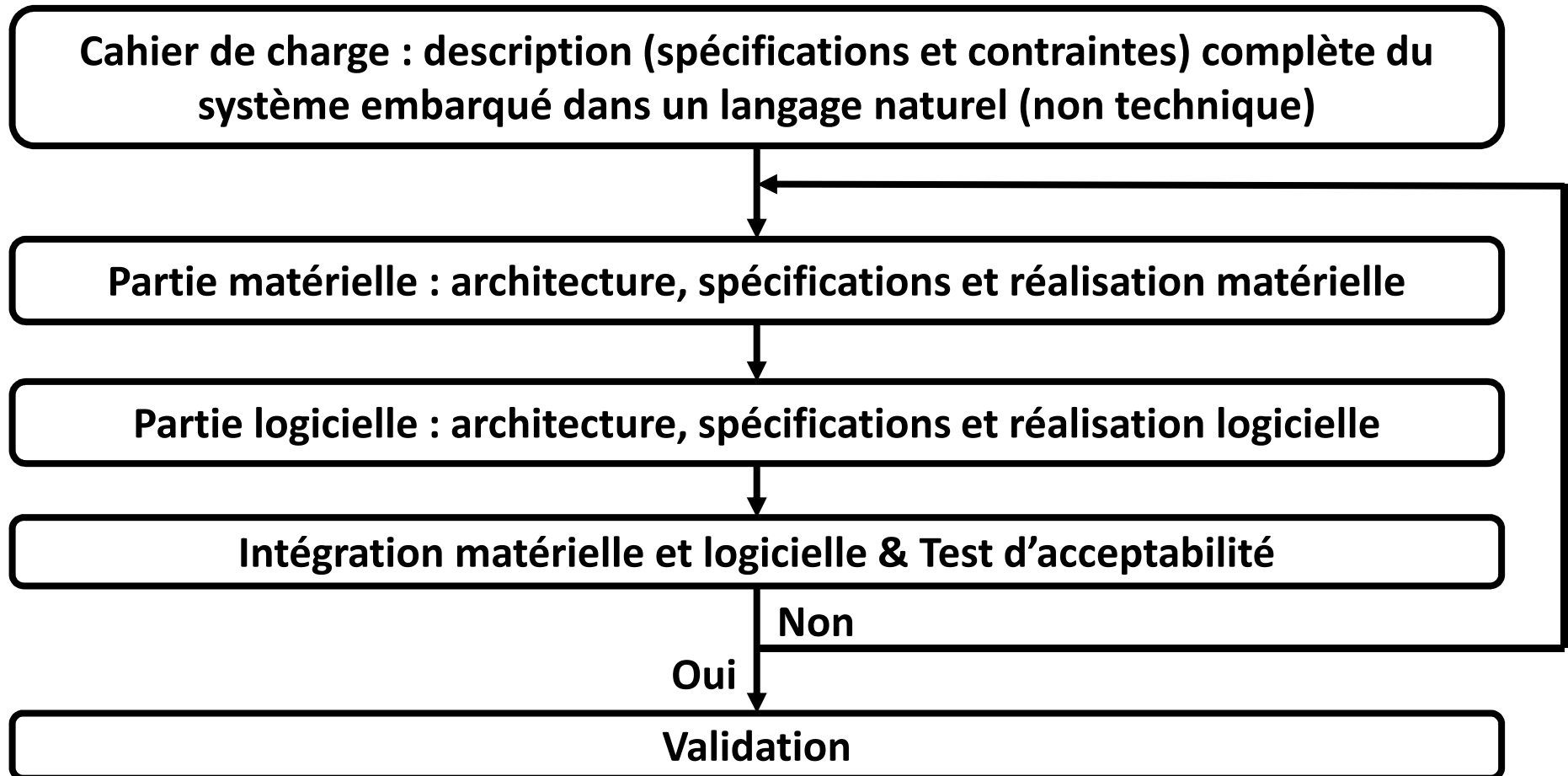
+

Logiciel

Caractéristiques principales d'un système embarqué

- C'est un système principalement numérique ;
- Il est constitué de logiciel et de matériel ;
- Il dispose des ressources limitées ;
- Il peut être ou non autonome ;
- Il ne possède généralement pas d'entrées/sorties standards (RS232, USB, etc...) ni de périphériques classiques (un clavier ou un écran d'ordinateur...) mais plutôt des périphériques limités (des boutons poussoirs, des claviers matricielles, des LED, des écrans LCD, etc...);
- Il peut être ou non, ouvert au monde extérieur pour des mises à jour ou pour communiquer ;
- Il doit généralement être sécurisé;
- Etc...

Etapes de réalisation d'un système embarqué

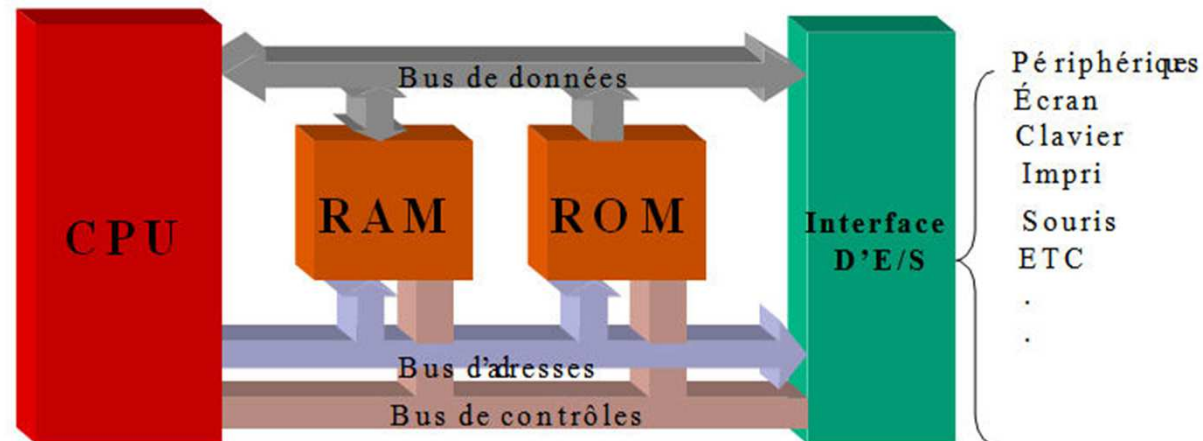


CHAPITRE II

Les microcontrôleurs

Qu'est-ce qu'un microcontrôleur ?

Un microcontrôleur est un petit système informatique contenant un processeur (donc programmables), des mémoires (données et programme), des circuits d'interfaces d'entrées-sorties et tout l'électronique nécessaire pour faire fonctionner tout cela ensemble.



Minimum pour mettre en œuvre un microcontrôleur

Tous les microcontrôleurs, quel que soit leur type ou leur fabricant, sont des composants à processeur.

Ils intègrent tous les trois composants de base à savoir un processeur, des mémoires et des circuits d'interfaces d'entrées sorties.

Ce qui fait que, tous les microcontrôleurs, pour fonctionner, ont besoin de certains circuits de base comme l'alimentation, l'horloge, le circuit reset, etc...

L'alimentation

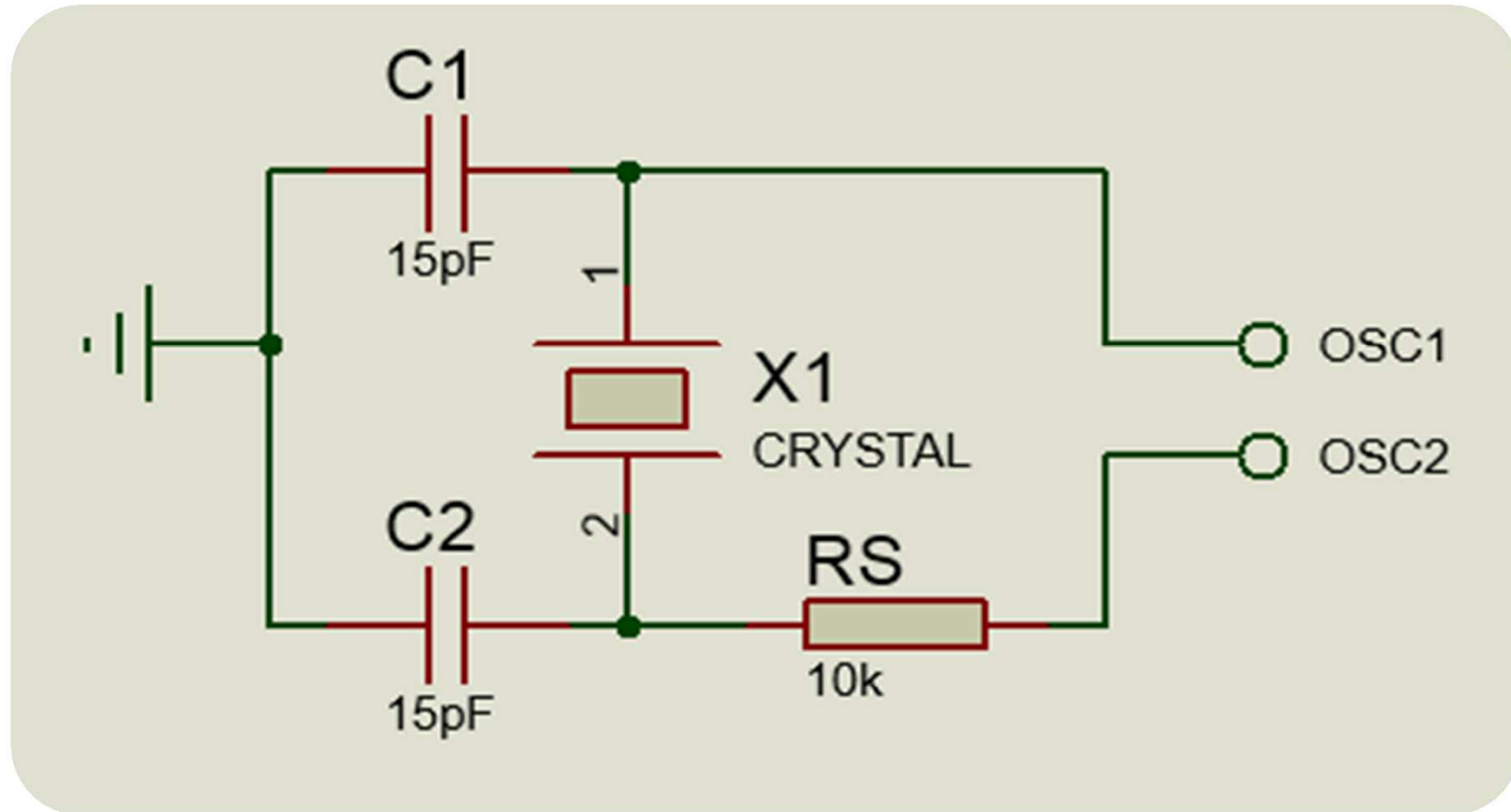
Tous les microcontrôleurs actuels fonctionnent sous une tension unique qui varie, selon les types de circuits, de 1,8 à 6 volts environ, avec une prédilection encore aujourd'hui pour la tension 5 volts, même si on se dirige peu à peu vers des versions fonctionnant sous une tension 3,3 volts.

Cette tension doit être stable.

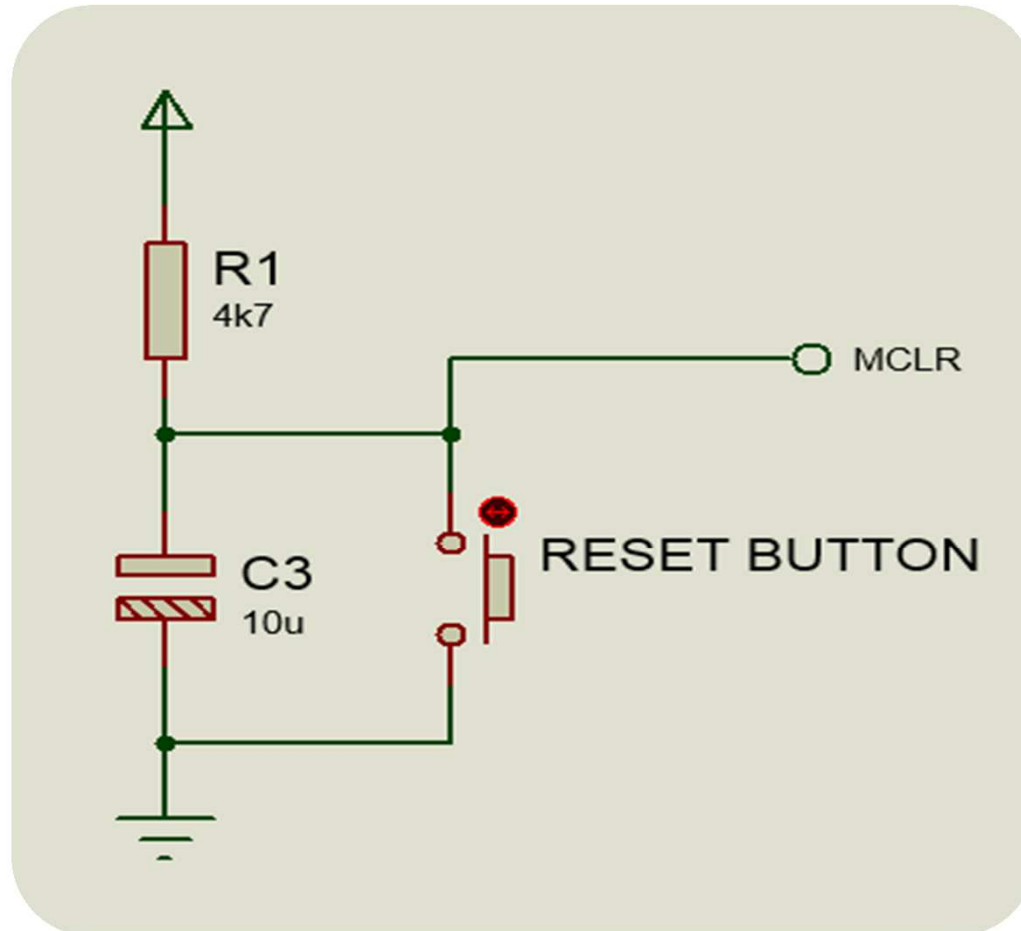
Les broches d'alimentation sont couramment appelées V_{DD} pour le potentiel et V_{SS} pour la masse (GND). Lorsque plusieurs pattes V_{DD} et V_{SS} sont présentes sur un même boîtier, elles doivent toutes être reliées en externe à l'alimentation, et ce bien qu'elles soient parfois interconnectées en interne dans certains circuits.

Il est de bonne pratique de découpler l'alimentation avec un condensateur de 22 à 100 nF.

L'horloge



Le circuit reset



Le chien de garde (Watchdog)

Le Watchdog est une temporisation que le système arme à chaque début de cycle. Si l'exécution du programme se termine avant la fin de la temporisation, tout va bien, et le microcontrôleur continue son travail. Dans le cas contraire, si la temporisation se termine avant la fin du cycle, cela signifie que le microcontrôleur a eu un problème pendant l'exécution du programme, ça peut être un problème d'accès à une entrée ou une sortie, une boucle sans fin dans le programme... Le Watchdog effectue alors son reset logiciel et le microcontrôleur recommence l'exécution du programme.

Le Watchdog peut être matériel (une bascule monostable) ou logiciel (un compteur).

Langages de programmation des microcontrôleurs

Programme en en assembleur ou en langage évolué (C, C++, Basic, Python, etc...)



Compilateur



Code machine du programme pour le processeur

```
01010010011111100000  
01100011101111111111  
01010101010100110101
```

Un environnement de développement intégré (EDI)



Un éditeur



Un compilateur



Un débogueur

Quelques critères du choix d'un microcontrôleur

- ✓ Le nombre d'entrées sorties : selon le projet ;
- ✓ Taille de la mémoire programme ;
- ✓ Taille de la RAM ;
- ✓ La puissance du processeur : vitesse de l'horloge, bus de données et bus d'adresses ;
- ✓ Consommation électrique ;
- ✓ La disponibilité des outils de développement : matériel et logiciel ;
- ✓ Le prix ;

- ✓ Etc...

CHAPITRE III

Arduino

Qu'est-ce que l'Arduino ?

Une carte électronique programmable.

ARDUINO



CARTE



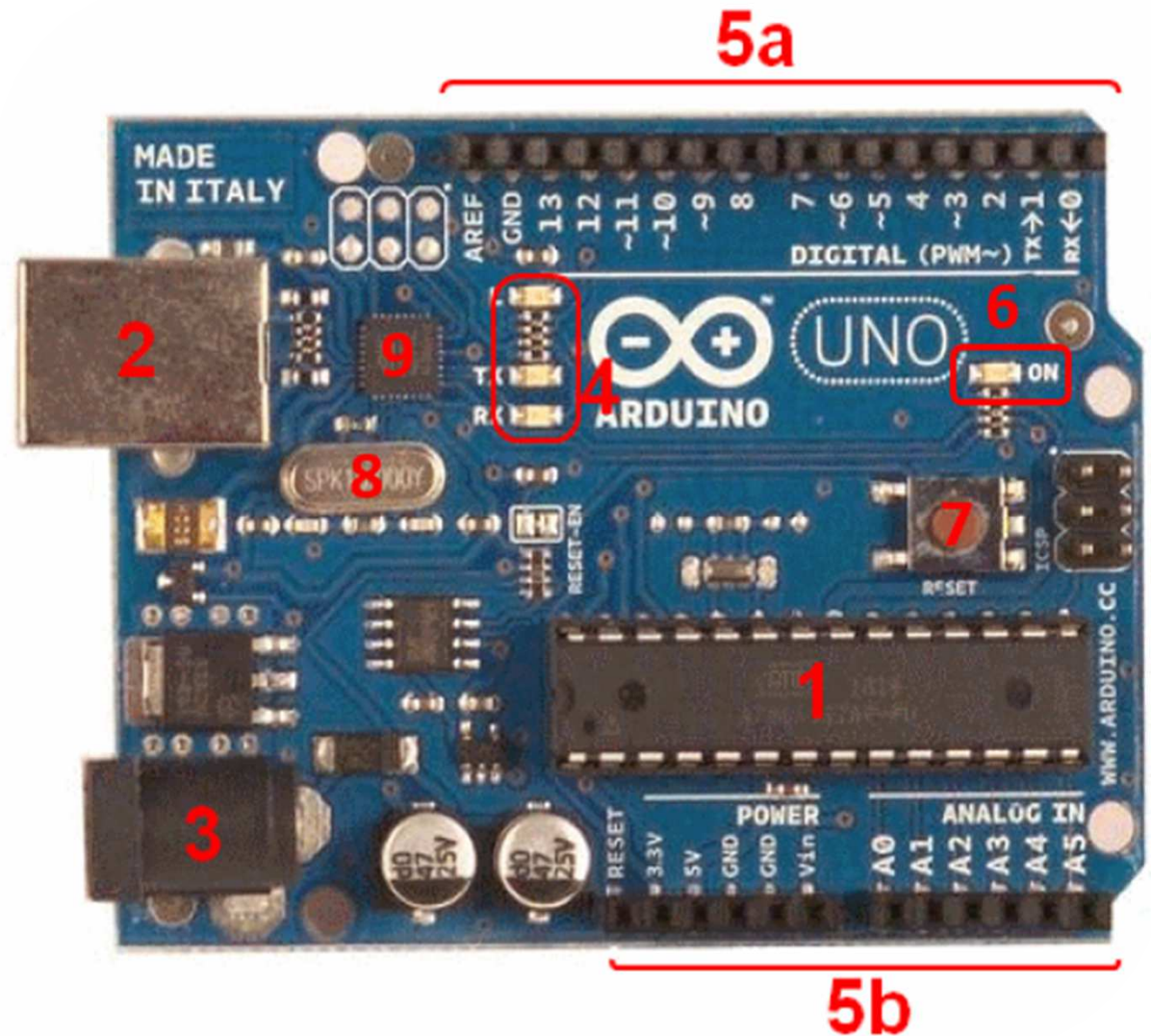
LOGICIEL

Un système embarqué.

Open source.

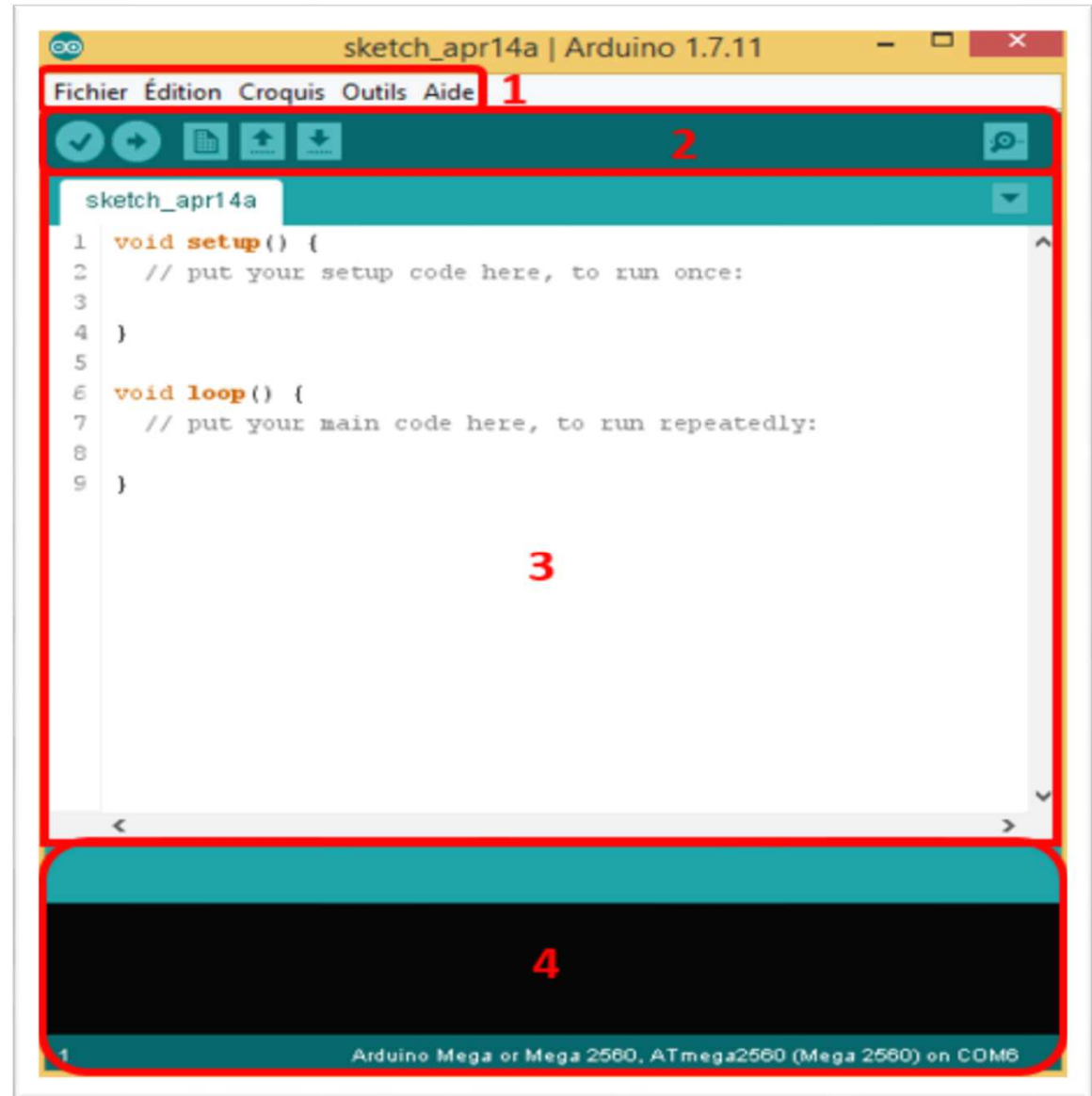
Une carte Arduino UNO

1. Un microcontrôleur AVR.
2. Port de programmation.
3. Port d'alimentation externe (7 à 12 Volts).
4. Trois micro-LED témoins (02 pour le port série et 01 connectée à la broche 13 servant à tester la carte).
5. La connectique (5a et 5b).
6. Micro-LED témoin de la mise sous tension.
7. Bouton RESET
8. Le Crystal.
9. ATMEGA 8U2 : convertisseur USB / TTL.



Le logiciel Arduino

1. Cette zone contient toutes les fonctionnalités offertes par le logiciel.
2. Boutons raccourcis.
3. Zone d'édition.
4. Zone d'informations.



Le langage Arduino

- ❖ **Un langage évolué de très haut niveau.**
- ❖ **Un mélange de C et de C++ restreint aux seules possibilités de la carte.**
- ❖ **L'environnement de développement est écrit en Java.**

Structure d'un programme Arduino (Sketch)

Définition éventuelle de constantes, d'inclusion de bibliothèques etc...

```
void setup( )  
{  
instructions à n'exécuter qu'une seule fois.  
}
```

```
void loop( )  
{  
instructions qui seront répétées indéfiniment.  
}
```

CHAPITRE IV

Systeme de commande automatique de l'éclairage public

Cahier de charge du projet

- ✓ **Au passage d'un véhicule, détecté par la barrière infrarouge, si la luminosité ambiante est inférieure ou égale à un certain seuil (50%), les lampes s'allument et s'éteignent automatiquement après une certaine durée prédéfinie à l'aide du clavier.**
- ✓ **Pendant les heures de pointe où la circulation est censée être à son maximum, les lampes restent allumer tant que la luminosité ambiante est inférieure ou égale au seuil (50%), que des véhicules passent ou pas. Par défaut cet intervalle est de 20h à 23h. Ce mode est appelé « mode continu ».**

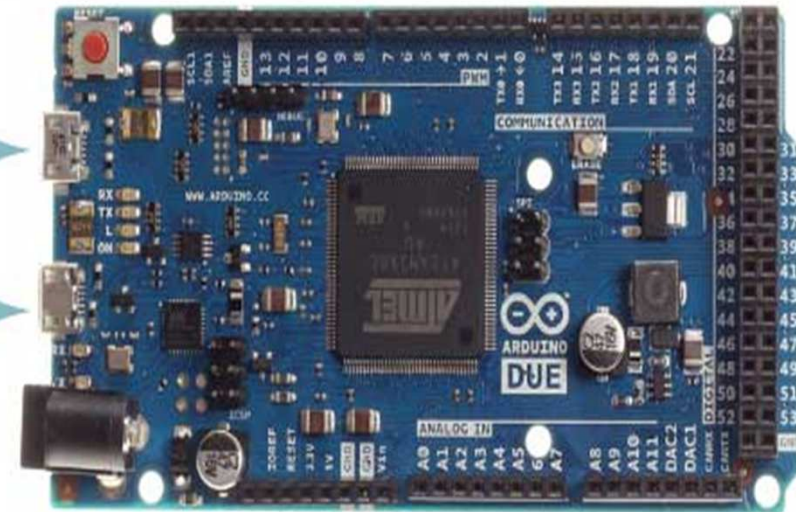
Le matériel utilisé dans notre projet

Une carte Arduino Due : unité de traitement

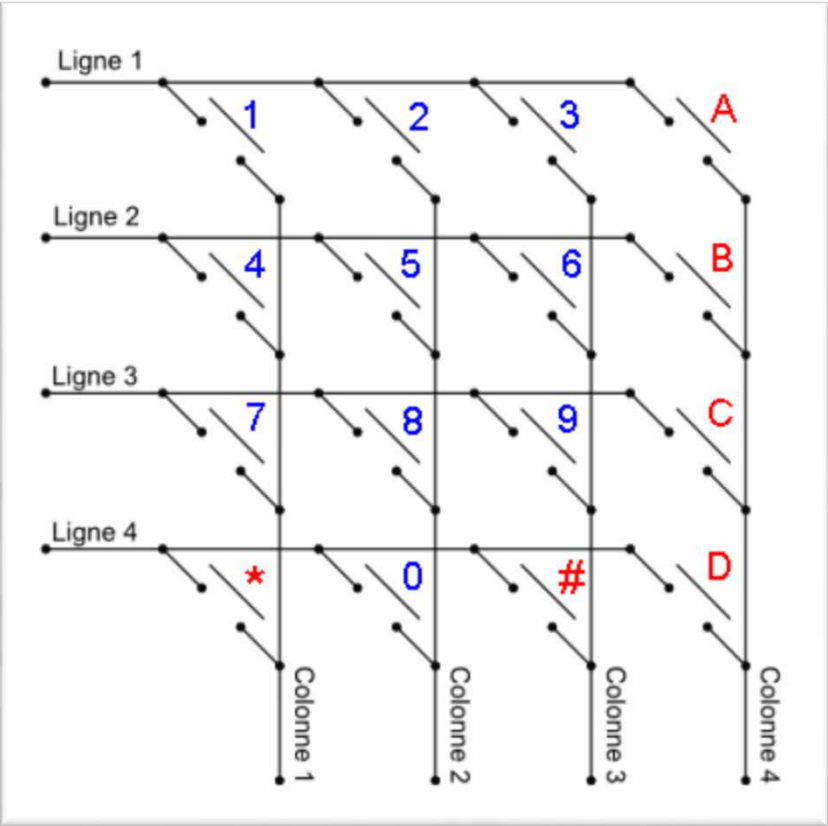
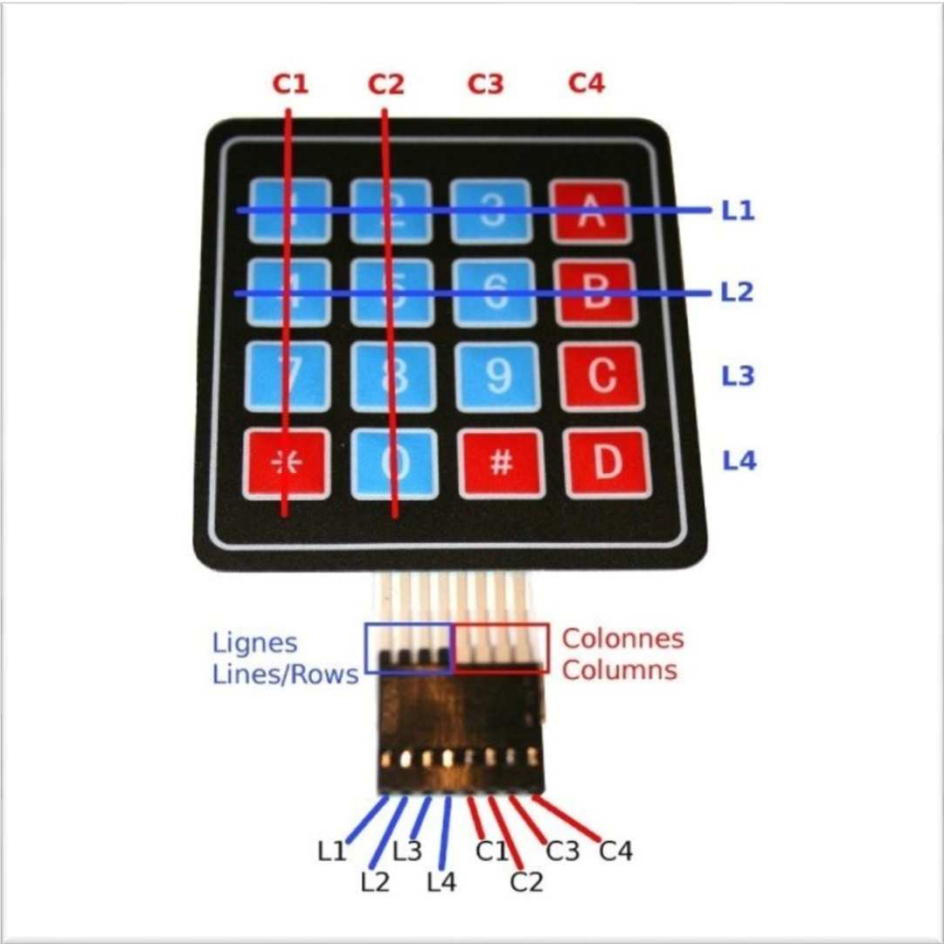
- ✓ Microcontrôleur : AT91SAM3X8E
- ✓ Tension de fonctionnement : 3V3
- ✓ Connecteur d'alimentation externe : 7 à 12 V
- ✓ Nombre d'E/S : 54 (dont 12 PWM)
- ✓ Broches analogiques : 12 entrées et 02 sorties.
- ✓ 02 ports USB
- ✓ Mémoire programme de type Flash : 512 ko.
- ✓ Une RAM statique : 96 ko (64 ko et 32 ko).
- ✓ 04 UART et 02 TWI (I2C).
- ✓ Un bouton RESET et un bouton d'effacement.

Native USB Port

Programming Port



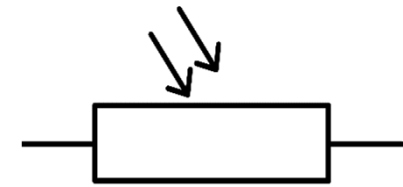
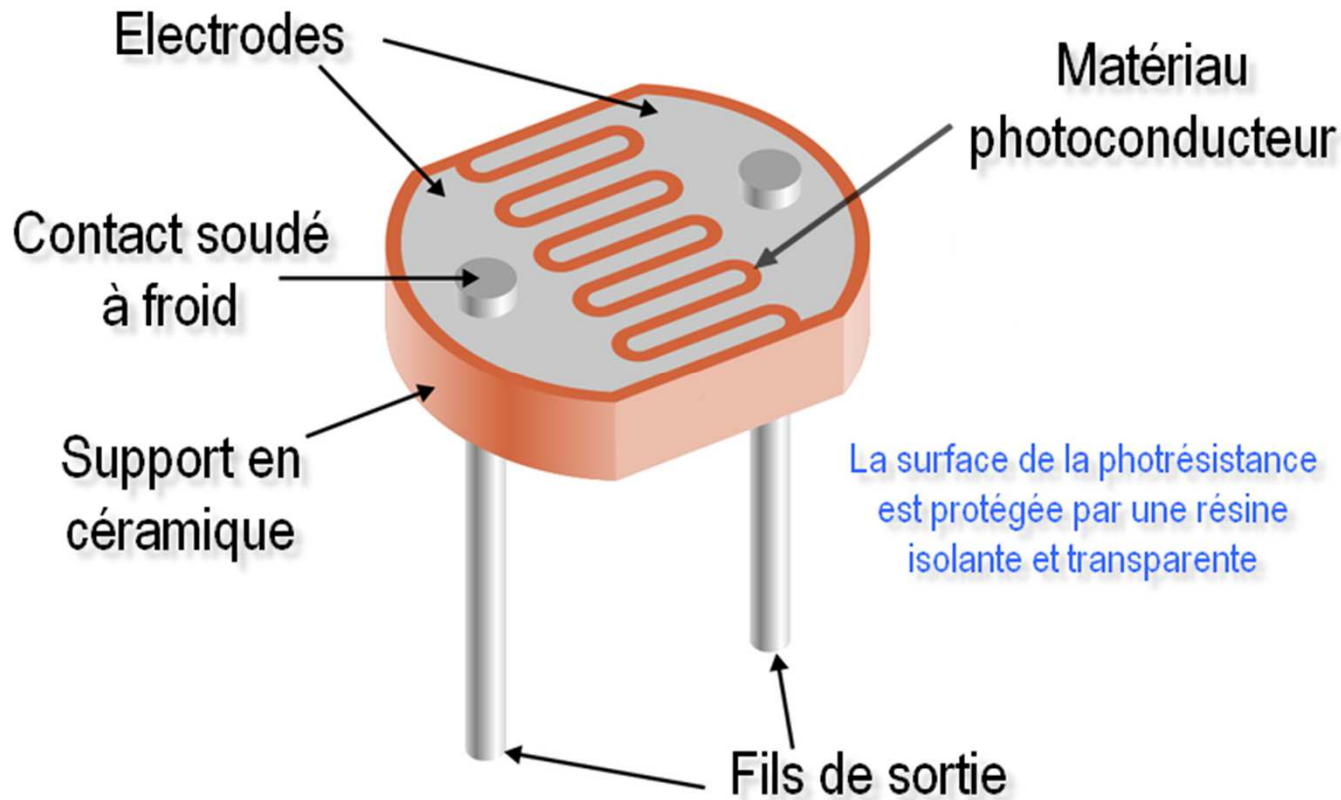
Un clavier matriciel 4x4



Un afficheur LCD 20x04



Une photorésistance : LDR



Un capteur de température LM35

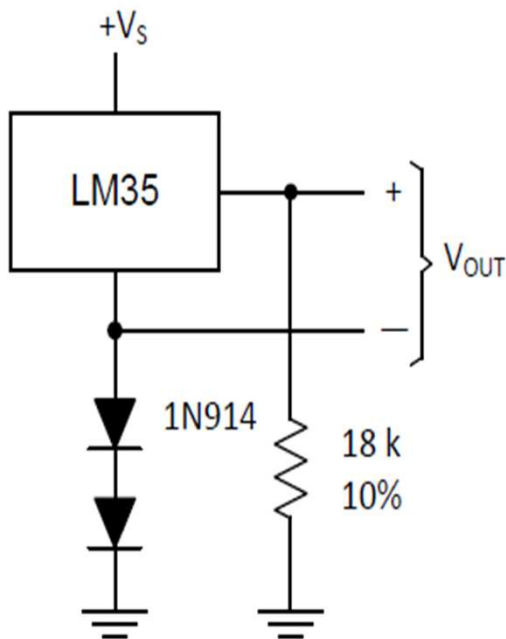
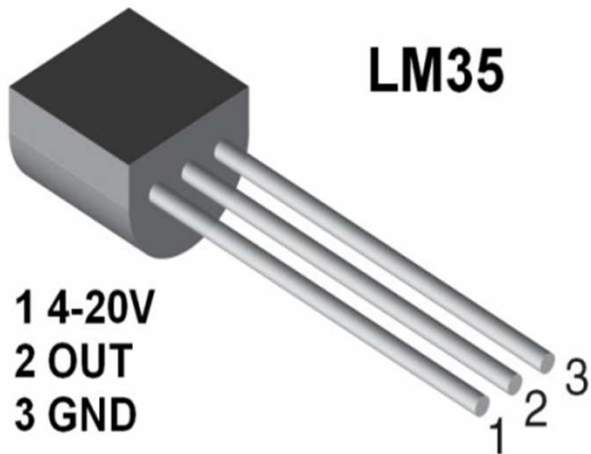


Figure 18. Temperature Sensor, Single Supply
(-55° to +150°C)

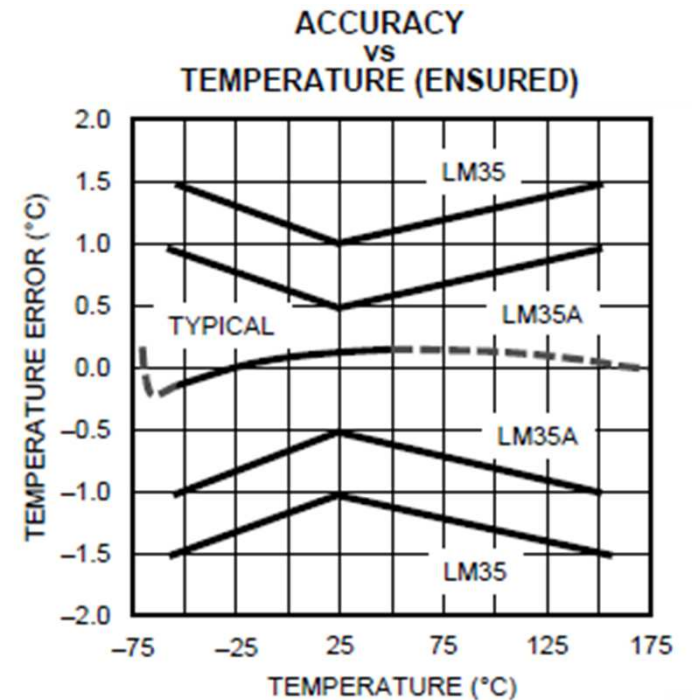
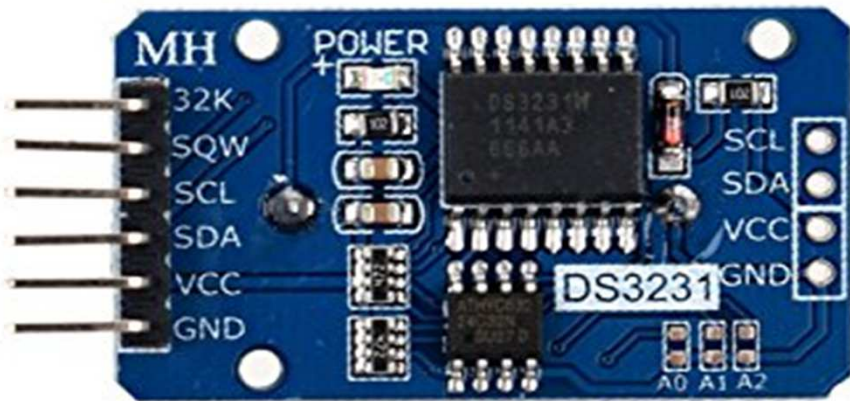
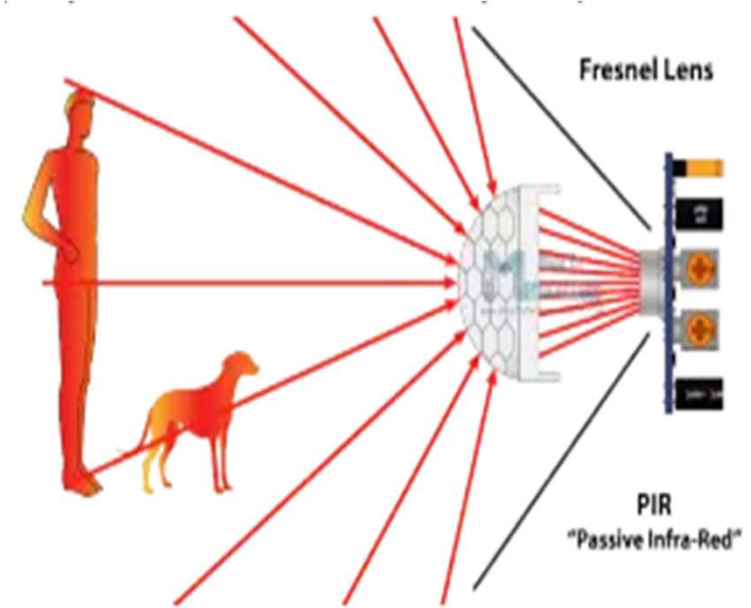
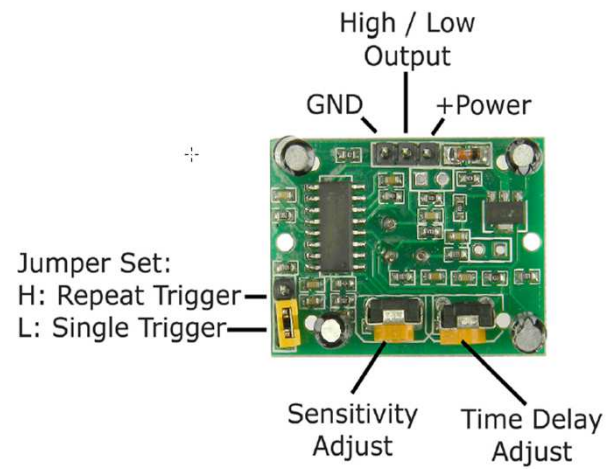


Figure 10.

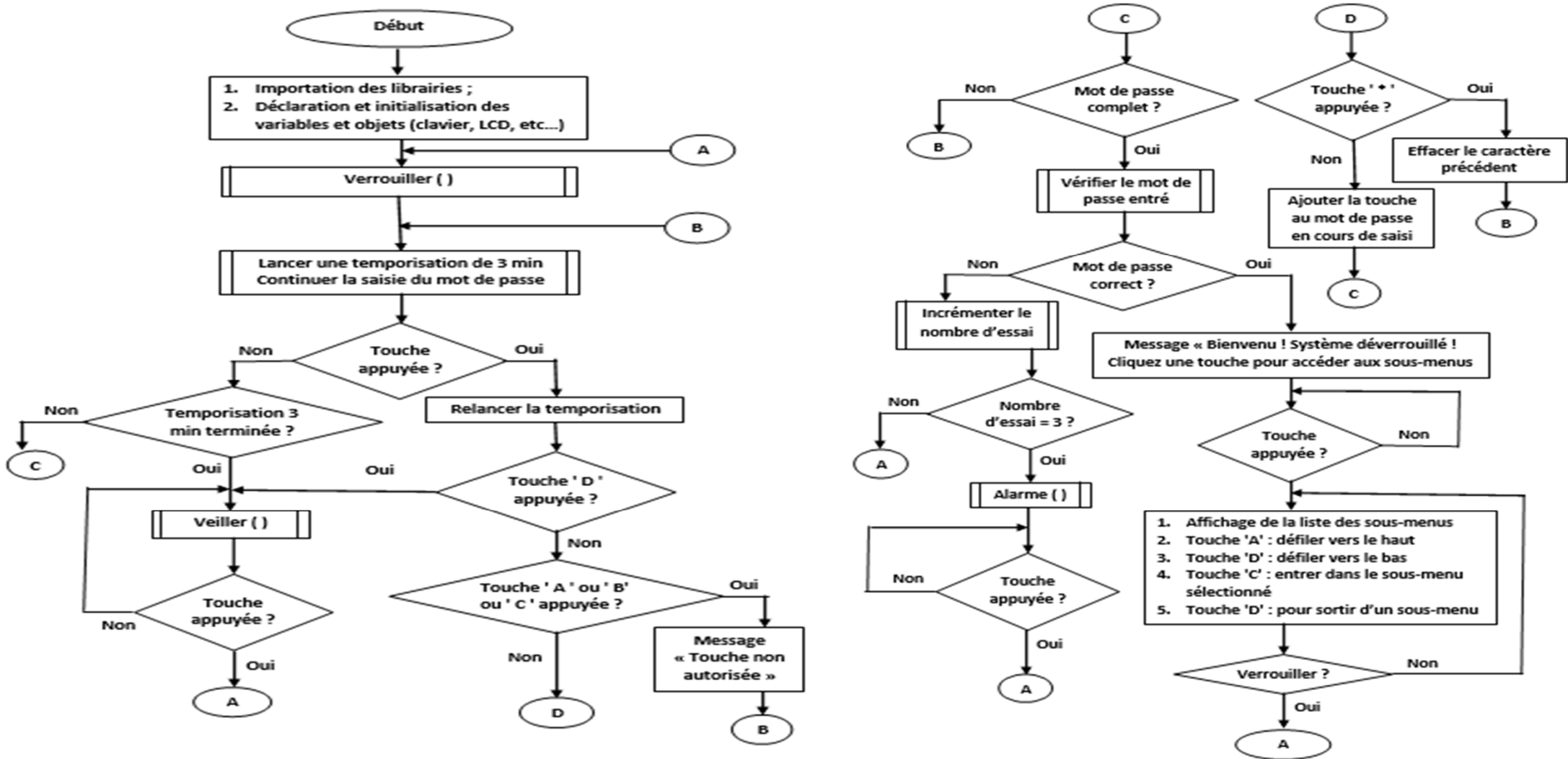
Un module d'horloge : DS3231



Un capteur de mouvement : HC-SR501



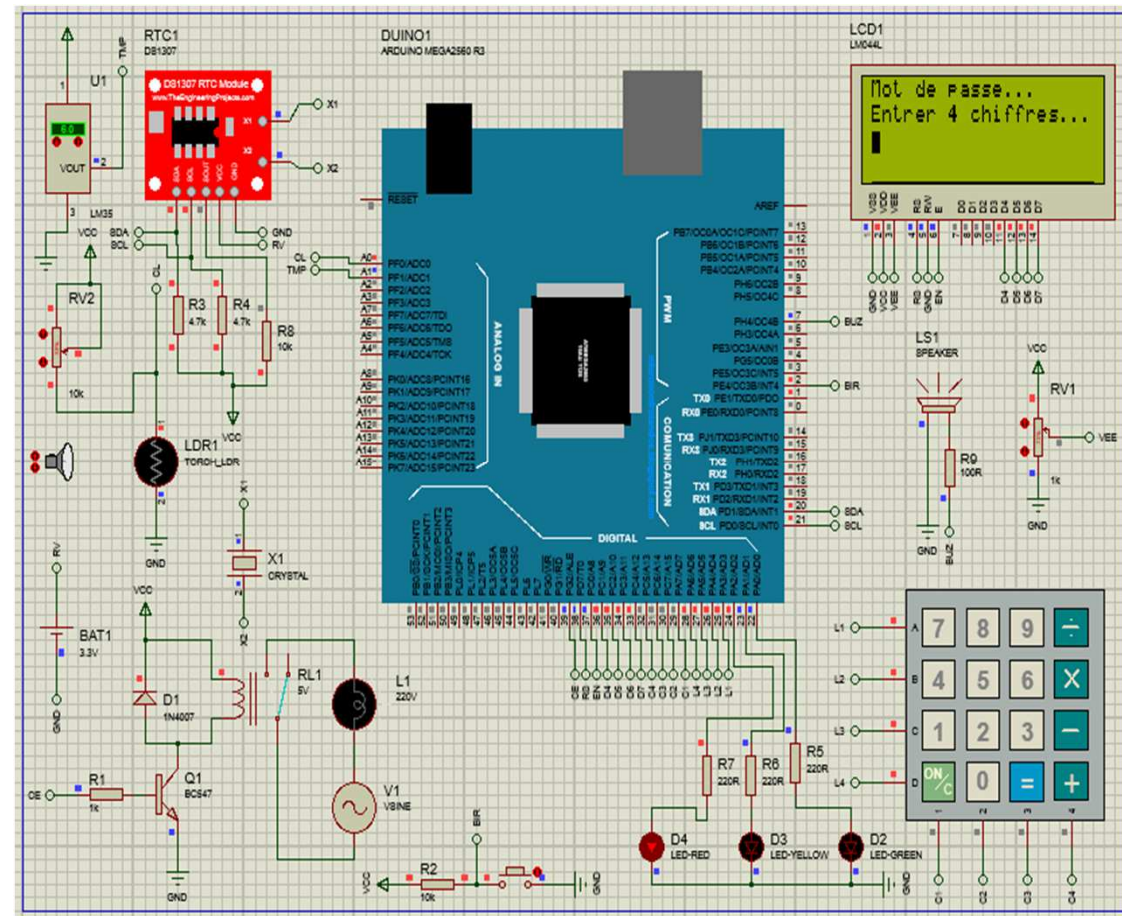
Le logiciel



Intégration matériel et logiciel, test final et validation

Simulation sur proteus

- ✓ L'Arduino Due est remplacé par l'Arduino Mega.
- ✓ Le DS3231 est remplacé par le DS1307.
- ✓ La barrière infrarouge, gérée par interruption, est simulé par un bouton poussoir tiré vers le haut par une résistance PULLUP.
- ✓ En pratique, la partie haute tension est remplacée par une LED rouge haute luminosité.



Le boîtier



Le boîtier

- L'interrupteur permet de couper (0) ou d'établir (1) l'alimentation.
- Le potentiomètre de gauche permet de diminuer ou d'augmenter le contraste de l'afficheur LCD.
- Le potentiomètre de droite permet de diminuer ou d'augmenter la sensibilité du capteur de lumière. Il forme un pont diviseur de tension avec la LDR.
- L'éclairage public est simulée par la LED rouge haute luminosité.



Conclusion

Merci pour votre attention