



CONCEPTION DE PROTOTYPES AVEC ARDUINO

**Toutes les étapes du prototypage : de l'idée à
la réalisation finale**

RÉSUMÉ

Ce document est destiné aux bidouilleurs et professionnels qui ont l'ambition de réaliser des prototypes électroniques Avec une plateforme bien connue qui est Arduino

SANOU Alexandre

Table de Matières

À propos de l'auteur	2
Préface	3
Composants nécessaires	3
Logiciels nécessaires.....	3
A qui s'adresse ce document ?	3
Chapitre 1 : Les bases d'Arduino	4
I. Interface matérielle	4
II. Interface logiciel.....	5
1. <i>La barre d'outils :</i>	6
2. <i>Fenêtre d'état :</i>	7
3. <i>Le moniteur série :</i>	7
III. Le langage de programmation Arduino.....	7
IV. Exécution du programme.....	9
Chapitre 2 : les prototypes	12
I. Prototype 1 : Boîte jeu de lumière	12
1. <i>Matériel nécessaire :</i>	12
2. <i>Description du matériel :</i>	12
3. Description des fonctionnalités et des fonctions du prototype	15
4. Montage	15
5. Algorithme et Programme	17
II. Prototype 2 : feu tricolore.....	20
1. Matériel nécessaire	20
2. Description du matériel	20
3. Description des fonctionnalités et des fonctions du prototype	21
4. Montage	22
6. Algorithme et Programme	22
III. Prototype 3 : Allumage automatique dans un entrepôt	25
1. Le matériel nécessaire	25
2. Description du matériel utilisé.....	25
3. Description des fonctionnalités et des fonctions du prototype	30
4. Montage	30
5. Algorithme et Programme	31

À propos de l'auteur

Alexandre SANOU est un ingénieur en télécommunications, électronicien et développeur digital. Avec ce profil atypique, SANOU a une expérience en développement de projets électroniques et informatiques complexes. Arduino est un environnement qu'il connaît bien et qu'il exploite pour la réalisation des prototypes électroniques. Avec ses connaissances en télécommunications et en développement d'applications mobiles, il s'est acclimaté au domaine des objets connectés. Le monde des startups est un mode qu'il connaît en tant que coach électronique pour les startups qui ont pour ambition de développer des prototypes électroniques.

Préface

La réalisation d'un projet Arduino est un processus méthodique qui fait appel à des compétences en électronique, des notions en programmation et une touche d'imagination. Dans ce document nous vous proposons trois (03) projets dans le domaine de la lumière et de l'éclairage. Le premier projet est une boîte de jeu de lumière, le second est un projet de feu tricolore et le troisième est un projet de d'éclairage automatique dans un entrepôt.

Pour réaliser ces projets nous ferons appel à des compétences en Design, en électronique, en programmation. Vous aurez particulièrement besoin de composants électroniques et d'outils informatiques.

Composants nécessaires

En fonction du projet vous aurez besoins de certains composants spécifiques.

- Arduino Méga
- Arduino Uno
- 30 leds (6 par couleurs)
- 30 tubes en verres
- 30 (résistances (optionnel)
- 4 ampoules, 4 douilles, rallonge, fils électrique
- 4 relais
- 2 capteurs de mouvement
- Capteur de son
- Récepteur et émetteur infrarouge
- Capteur de flamme
- Buzzer
- Papier cartonné

Logiciels nécessaires

- Arduino IDE (Integrated Development Environment)
- Fritzing

A qui s'adresse ce document ?

Ce livre s'adresse aux amateurs et aux professionnels qui souhaitent entrer dans le monde des prototypes. Le contenu de ce livre nécessite une connaissance préalable d'Arduino, des composants électroniques et de la programmation. Nous avons utilisé des composants de base et fourni des instructions étape par étape pour la construction des circuits. Nous avons gardé le code simple, lisible et minimal pour aider les débutants à comprendre les concepts et à développer des prototypes utilisables. Tout au long du livre, le code est cohérent et, là où c'est nécessaire, est expliqué en détail.

Chapitre 1 : Les bases d'Arduino

Arduino de la marque Arduino ou Genuino est une plateforme qui permet de développer des objets interactifs indépendants (prototypes) ou des objets connectés à un ordinateur. La plateforme Arduino est Open source. Arduino propose deux interfaces : une interface matérielle et une interface logicielle.

À la fin de ce chapitre vous serez capable :

- D'utiliser les composants Arduino
- D'utiliser l'interface Arduino
- D'écrire, de charger et d'exécuter des programmes Arduino

I. Interface matérielle

La partie matérielle d'Arduino propose des cartes électroniques sur lesquelles se trouve un microcontrôleur. Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le pilotage d'un robot. Le microcontrôleur Arduino est un microcontrôleur Atmel AVR qui a plusieurs versions comme le ATmega2560 et ATmega32u4. Le microcontrôleur est associé à d'autres composants pour faciliter la programmation et l'interfaçage avec d'autres circuits. Un module Arduino possède un régulateur 5V et une horloge qui est soit un oscillateur à quartz 16MHz soit un résonateur céramique. La carte électronique Arduino possède une interfaçage entrée/sortie simple qui permet de connecter des composants électroniques ou électriques et des appareils électroniques ou électriques. La **Figure 1.1** [29] est une carte Arduino Uno, plusieurs autres cartes ont été développées au fil du temps.



Figure 1. 1 Arduino Uno

La présente liste résume les parties les plus importantes des cartes Arduino qui seront utilisées dans ce document.

- **Broches numériques (Digital pins)** : Au total, il y a 14 broches numériques sur l'Arduino Uno et 54 broches numériques sur l'Arduino Méga. Les broches numériques peuvent être à la fois INPUT (entrant) OUTPUT (sortant), mais leur état peut être seulement HIGH (haut) ou LOW (bas). HIGH signifie qu'il y a du courant tandis que LOW signifie pas de courant.
Un exemple d'utilisation des broches numérique est l'allumage d'une Led. HIGH pour allumer la Led et LOW pour étendre la Led.
- **Broches analogique (Analog pins)** : l'Arduino Uno prend en charge 6 broches analogiques notées de A0 à A5. L'arduino Méga prend en charge 16 broches analogiques notées de A0 à A15. Contrairement aux broches numériques, les lectures des broches analogiques peuvent varier de 0 à 1023. Un bon exemple de capteur qui fournit une lecture est un capteur d'humidité du sol. La plage de valeurs permet d'identifier le taux d'humidité dans le sol.
- **Connecteur USB** : Un connecteur USB vous permet de :
 - Mettre sous tension l'Arduino,
 - Connecter Arduino à l'ordinateur
 - De visualiser l'activité de la carte sur u moniteur série
- **Alimentation** : Arduino dispose d'un connecteur d'alimentation qui permet de l'alimenter. Arduino supporte une alimentation de 5 à 9 V.

II. Interface logiciel

L'interface logicielle est un logiciel de programmation appelé Arduino IDE une application java libre et multiplateforme. C'est un éditeur et compilateur de programmes qui permet de téléverser le firmware (code après compilation) vers le module Arduino à travers le port *USB*, le Bluetooth ou le câble RS-232. Il est également possible de compiler et de téléverser le programme via l'interface ligne de commande. Le logiciel Arduino supporte les systèmes d'exploitation Windows, GNU/Linux et MacOS. Le langage de programmation utilisé par le logiciel Arduino est proche du C et du C++ compilé avec avr-g++. Le logiciel utilise la bibliothèque Arduino qui lui permet d'utiliser la carte et ses entrées/sorties. La **Figure 1.2** présente l'interface du logiciel Arduino.

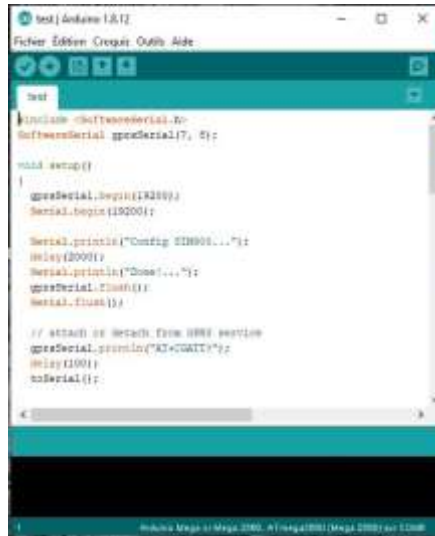


Figure 1. 2 Interface du logiciel Arduino

L'IDE Arduino est composée de 03 trois éléments importants :

- La barre d'outils
- La fenêtre d'état
- Le moniteur série

1. La barre d'outils :

La barre d'outils au-dessus de l'IDE, comme illustré dans la figure 1.3 permet d'accéder aux options de l'IDE.



Figure 1. 3 Barre d'outil IDE

- **Compiler/ Vérifier** permet de lancer le programme et de vérifier qu'il n'y a pas d'erreurs. Le résultat est visible dans la fenêtre d'état.
- **Téléverser** : Après la vérification du programme on télécharge le programme dans la carte Arduino. Si vous oubliez de compiler le programme et que vous allez directement sur téléverser, l'IDE procèdera à la vérification avant de télécharger le programme.
- **Fichier** vous permet de créer un nouveau fichier, ouvrir un fichier ou encore sauvegarder un fichier.
- **Outils** permet d'accéder aux options pour charger les bibliothèques, connecter la carte Arduino et accéder au moniteur série.

2. Fenêtre d'état :

La fenêtre d'état figure 1.4, nous renseigne si le code a bien été chargé. Tous les résultats de la compilation sont disponibles dans la fenêtre d'état. Toute erreur sera listée dans la fenêtre d'état et vous permettra de corriger vos erreurs.



Figure 1. 4 Fenêtre d'état de l'IDE

3. Le moniteur série :

Sur le moniteur série Figure 1.5, vous avez le journal des informations écrites par la carte Arduino. Ces informations sont écrites par les fonctions `Serial.print()` et `Serial.println()`. Le moniteur série permet d'envoyer directement des commandes à la carte Arduino.

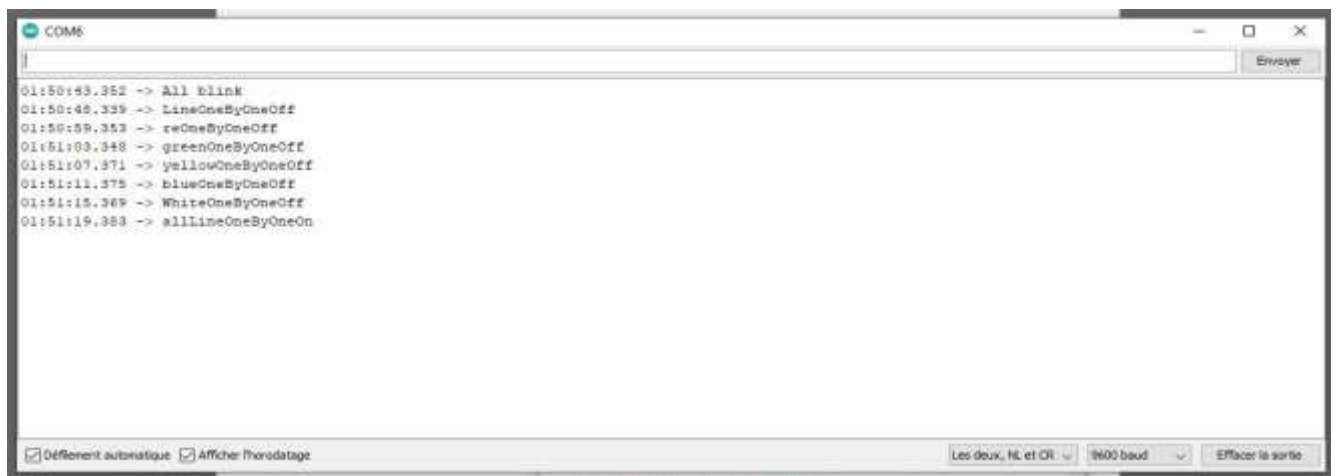


Figure 1. 5 Le moniteur série

III. Le langage de programmation Arduino

Le langage de programmation utilisé par le logiciel Arduino est proche du C et du C++. Certaines fonctions et constructeurs de la programmation Arduino sont décrits dans le tableau 1.1.

Code	Description
int	Entier, 12
float	Décimale, 1,3

Char[]	Caractère, Arduino
HIGH	Broche numérique avec courant
LOW	Broche numérique sans courant
INPUT	Broche avec lecture
OUTPUT	Broche avec écriture
A0 - A14	Constantes pour les broches analogiques, dépendent de la carte
0 - 53	Constantes pour les broches numériques, dépendent de la carte
Code	Description
analogRead()	Retourne la valeur d'une broche analogique (0 - 1023)
analogWrite(..)	Fixe la valeur de sortie de la broche analogique
digitalRead()	Retourne la valeur de la broche numérique (HIGH , LOW)
digitalWrite(..)	Fixe la valeur de la broche numérique (HIGH or LOW)
Serial.begin(..)	Initialise la valeur du moniteur, vitesse de traitement de l'information
Serial.print()	Écrit un message sur le moniteur
Serial.println()	Écrit un message sur le moniteur avec saut de ligne
Delay(ms)	Marque un arrêt
setup()	Les fonctions déclarées une seule fois
loop()	La boucle, le code y est répété à plusieurs reprises
if	Condition est validée
if ... else	Si condition non validée allez à sinon
#define	Définit une constante
#include	Inclut une bibliothèque externe
//	Commenter une seule ligne
/* */	Commenter plusieurs lignes

Pour avoir plus de détails veuillez consulter ce site : [Arduino Reference - Arduino Reference](#)

L'IDE Arduino propose une interface simple de programmation. Dans les conditions normales, vous votre code est structuré en 3 parties :

- Les librairies : Déclarez toutes les librairies nécessaires. Une librairie est développée et testée. Elle contient très souvent des instructions d'utilisation et des codes basiques d'utilisation.
- Constantes et les variables : Chaque constante ou variable doit être prédéfinie avant tout utilisation dans le code. Les constantes ne changent pas, on les utilise pour la définition des broches numériques de la carte Arduino.

- Fonctions : Les fonctions sont l'implémentation des fonctions standards ou des fonctions personnalisées. Une fonction contient une fonctionnalité particulière. Il est recommandé d'utiliser des fonctions pour éviter de dupliquer certaines parties du code. Ainsi vous pouvez appeler la fonction à tout moment dans le code.

Exemple : faire clignoter une led

```
int redPin1 = 5; // constante

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600); // fonction serial begin
  pinMode(redPin1, OUTPUT); // fonction OUTPUT
  digitalWrite (redPin1, LOW); instruction éteindre
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite (redPin1, HIGH); // instruction allumer
  delay(1000); // attendre une seconde
  digitalWrite (redPin1, LOW); instruction éteindre
}
```

IV. Exécution du programme

L'exécution du programme sur Arduino se fait par étape. Chaque programme est compilé en fonction de la carte Arduino qui sera utilisée. Ainsi donc certaines bibliothèques ne seront utilisables que sur les cartes pour lesquelles elles ont été développées. Pour exécuter un programme Arduino :

- Ouvrez l'IDE Arduino
 - Allez dans outils
 - Choisir « Type de carte »
 - Choisissez la carte que vous allez utiliser
- Ces étapes sont illustrées sur la figure 1.6

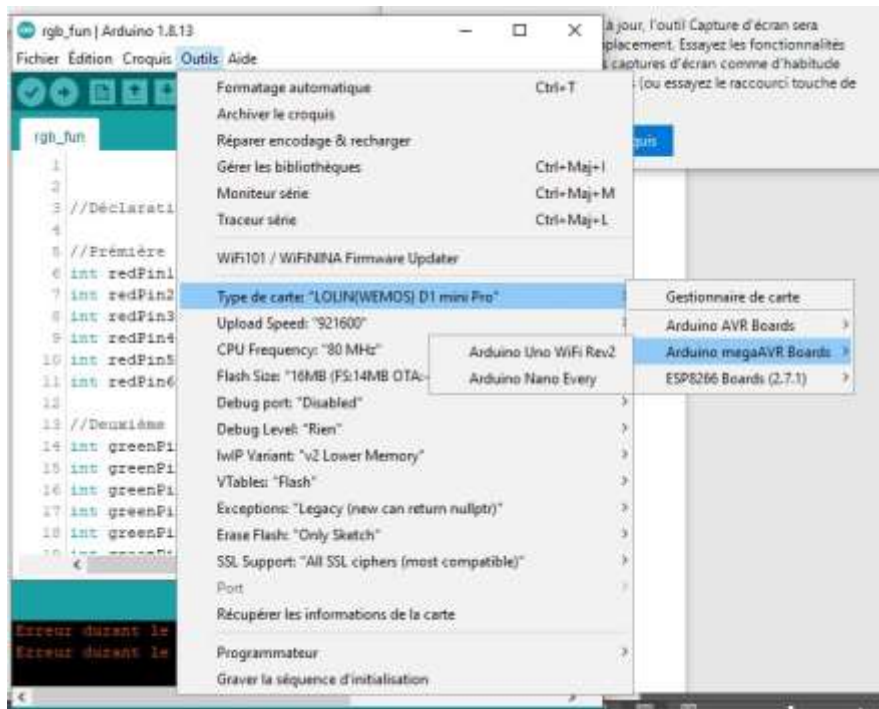


Figure 1. 6 Choix de la carte Arduino

- Poursuivez dans le même onglet « outils »
 - Allez dans port et choisissez le port sur lequel est connecté votre Arduino
- La Figure 1.7 illustre le choix du port

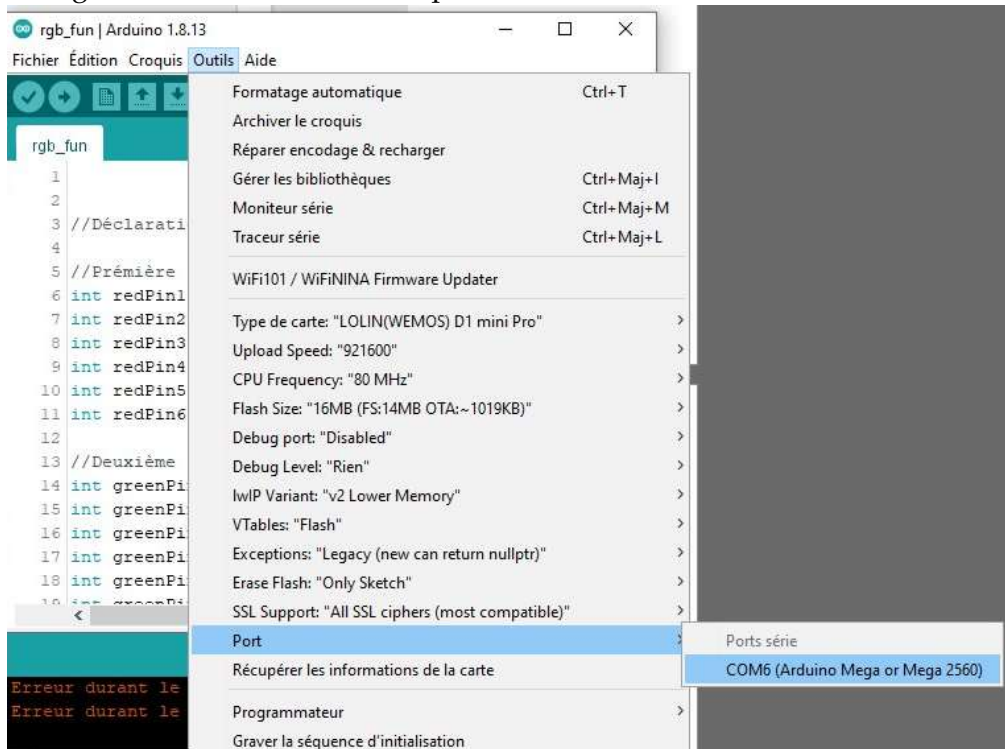


Figure 1. 7 Choix du port

- Enfin lancer le moniteur série et assurez-vous que la valeur de fréquence de lecture est identique à celle déclarée dans le programme.

Chapitre 2 : les prototypes

Les prototypes qui seront réalisés dans ce chapitre sont : la boîte jeu de lumière, le feu tricolore et l'allumage automatique dans un entrepôt.

I. Prototype 1 : Boîte jeu de lumière

La boîte jeu de lumière est un projet assez amusant. La lumière est un phénomène qui a été démystifié au fil des années. Et aujourd'hui la lumière est omniprésente dans notre quotidien. L'objectif de ce projet est de passer d'une fonction basique et de la transformer en un ensemble assez complexe principalement par la programmation. Nous apprendrons à être logique dans notre programme et d'exécuter des fonctions selon le résultat recherché.

1. Matériel nécessaire :

La liste exhaustive du matériel nécessaire à la réalisation du projet :

- Un Arduino Méga
- 30 Leds (6 par couleur)
- 30 résistances (optionnel)
- Papier cartonné

2. Description du matériel :

Chaque composant a des caractéristiques bien définies. Il est important de connaître le mode de fonctionnement de chaque composant au risque de les griller dans le circuit. Une fois le mode de fonctionnement du composant connu il sera aisé d'utiliser ses fonctions.

- L'Arduino Méga

L'Arduino Méga 2560 est conçu pour les projets qui nécessitent plus de broches d'E/S, plus de mémoire d'esquisse et plus de RAM. Avec 54 broches numériques et 16 broches analogiques et un plus grand espace pour votre dessin. Cela donne à vos projets beaucoup d'espace et d'opportunités tout en maintenant la simplicité et l'efficacité de la plateforme Arduino.

Les caractéristiques techniques de la carte Arduino Méga :

- Microcontrôleur: ATmega2560
- Tension de fonctionnement: 5 V
- Tension d'alimentation (recommandée): 7- 12 V
- Tension d'alimentation (limites): 6 - 20V
- Nombre d'Entrées / Sorties: 54 (dont 14 pouvant générer des signaux PWM)
- Nb ports "Analogique/Numérique": 16

- Courant max. par Entrées / Sorties: 40mA
- Courant pour broches 3.3 V: 50 mA
- Mémoire Flash: 256 KB (ATmega328) dont 8 KB utilisé par le bootloader
- SRAM: 8 KB (ATmega328)
- EEPROM: 4 KB (ATmega328)
- Fréquence d'horloge: 16MHz

La carte Arduino est représentée sur la figure 2.1



Figure 2. 1 Arduino Méga

- Led

Une LED se caractérise principalement par 3 grandeurs :

- La chute de tension directe V_f aux bornes de la LED pour un courant I de 20 mA.
- L'intensité lumineuse I_v qui s'exprime en mcd (millicandela).
- L'angle de rayonnement par rapport à l'axe optique, en degrés.

Figure 2.2 : résistance insérée en série avec une led:

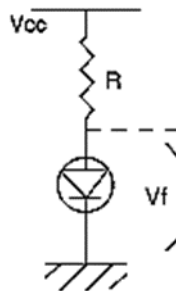


Figure 2. 2 Résistance en série avec une Led

La loi d'Ohm

$$R = (V_{cc} - V_f) / I$$

R s'exprime en Ohms

V_{cc} et V_f s'expriment en Volts

I s'exprime en Ampères

Les caractéristiques techniques de la Led

- Tension max en blocage: 5,0V
 - Courant max en blocage: 5μA
 - Tension max en passant: 3,6V
 - Tension min en passant: 1,8V
 - Courant max en passant: 30mA
 - Courant max en peak: 100mA
- Résistances 220 Ohm adaptée aux Leds
- Les résistances 5% sont codées par 4 bagues de couleur. Les 2 premières représentent les 2 chiffres de la valeur de la résistance, la troisième le facteur de multiplication et la quatrième la tolérance (bague or = 5% - bague argent = 10%).

Code de couleur des résistances

COULEUR	1er CHIFFRE	2ème CHIFFRE	Multiplicateur
Noir		0	
Brun	1	1	x 1
Rouge	2	2	x 10
Orange	3	3	x 100
Jaune	4	4	x 1000
Vert	5	5	x 10000
Bleu	6	6	x 100000
Violet	7	7	x 1000000
Gris	8	8	
Blanc	9	9	

La figure 2.3 ci-dessous présente des résistances.



Figure 2. 3 Résistances

- Le papier cartonné sera utilisé pour la maquette

3. Description des fonctionnalités et des fonctions du prototype

Notre prototype est une boîte à jeu de lumière. Sa fonctionnalité est un jeu de lumière.

Les fonctions qui sont utilisées par le prototype sont :

- Allumer
- Éteindre
- Clignoter

4. Montage

Le montage se fait sur une plaque d'essai. Les composants sont interconnectés par des fils de connexion. La figure 2.4 est le montage physique sur plaque d'essai.

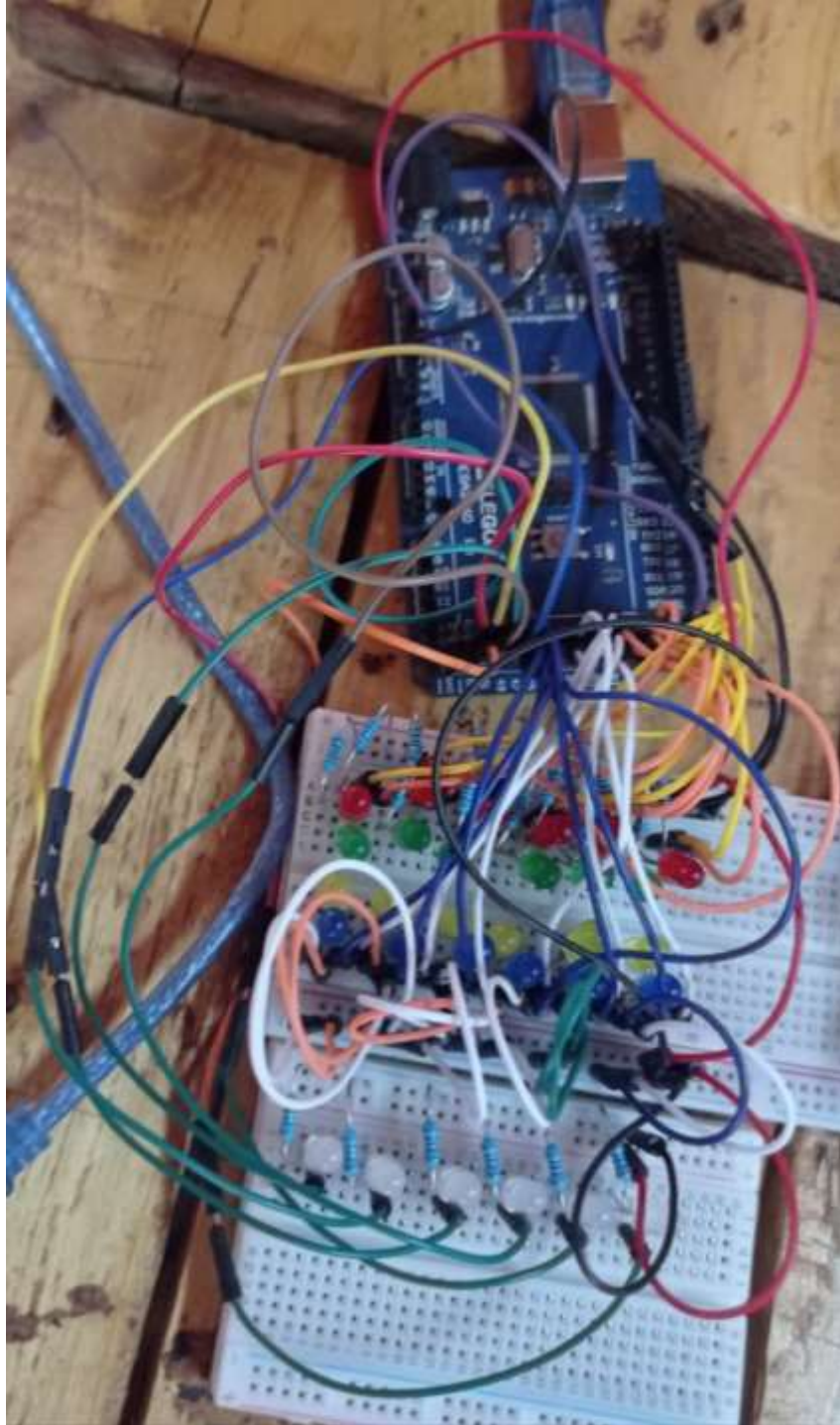


Figure 2. 4 Montage physique sur plaque d'essai

Il est également possible de concevoir un montage électronique sur des logiciels dédiés comme Proteus et Fritzing. La Figure 2.5 montre le montage d'une led + d'une résistance sur Fritzing

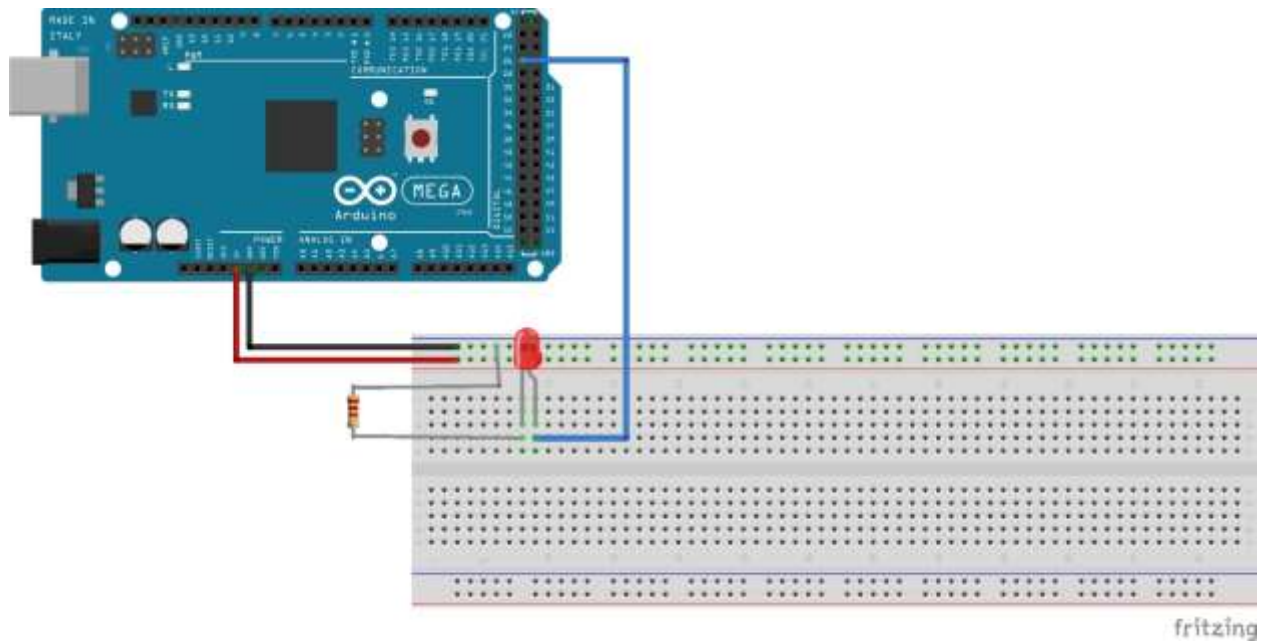
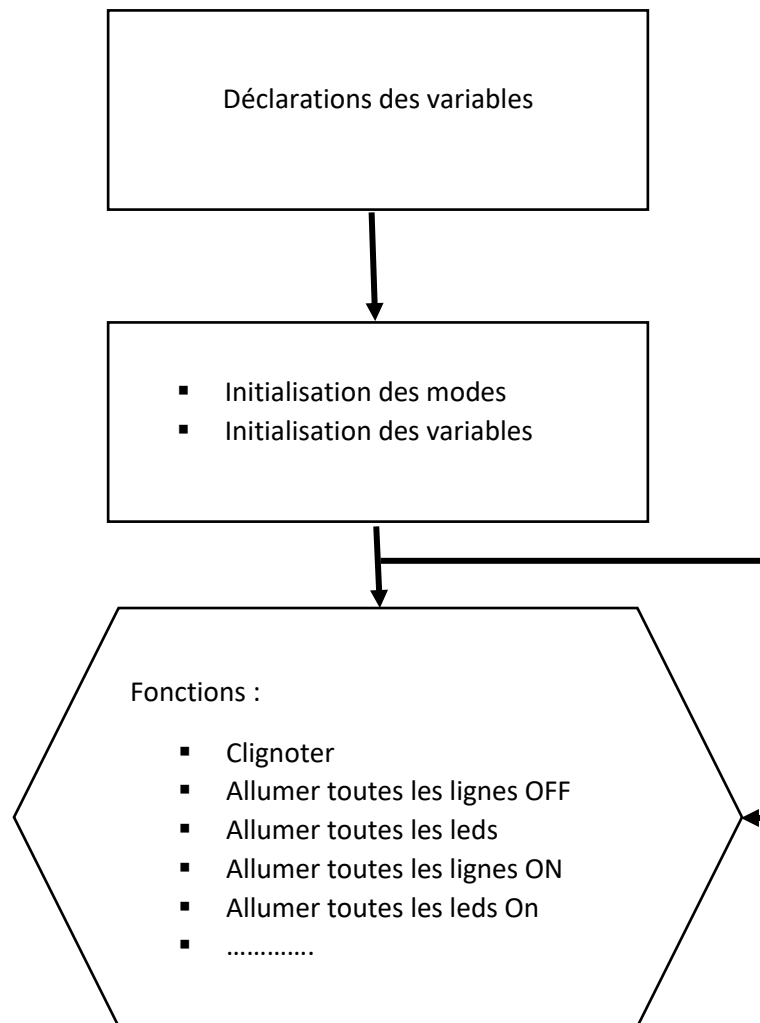


Figure 2.5 Montage sur Fritzing

5. Algorithme et Programme

Le programme Arduino est l'exécution ordonnée des fonctions définies dans le code. Ce code est écrit selon un algorithme qui est une suite d'instructions et d'opérations permettant d'aboutir à un résultat.

- L'algorithme



- Le programme
Le programme complet est disponible à l'adresse suivante :

https://www.dropbox.com/s/lz18l639jnposar/rgb_fun.rar?dl=0

- Le prototype final



II. Prototype 2 : feu tricolore

Le projet feu tricolore consiste à reproduire la circulation à l'intersection d'un rond-point. Ce projet permettra de simuler une circulation normale, une circulation intelligente et une circulation prudente.

1. Matériel nécessaire

La liste exhaustive du matériel nécessaire à la réalisation du projet :

- Un Arduino Uno
- 9 Leds (3 par couleur)
- 9 résistances (optionnel)
- Papier cartonné

2. Description du matériel

Décrivons chaque composant pour pouvoir mieux les exploiter.

- Arduino Uno

L'Arduino uno est architecturé autour d'un microcontrôleur Atmel ATmega328 (livré pré-monté sur un support) associé à une interface USB.

L'ensemble des entrées/sorties de la platine sont disponibles sur des connecteurs femelles présents sur les bords de la platine. Il possède 14 broches numérique et 6 broches analogiques.

Les caractéristiques techniques de l'Arduino Uno

- Microcontrôleur: ATmega328
- Tension de fonctionnement: 5 V
- Tension d'alimentation (recommandée): 7 - 12 V
- Tension d'alimentation (limites): 6 - 20V
- Nombre d'E/S: 14 (dont 6 pouvant générer des signaux PWM)
- Nb ports "Analogique/Numérique": 6
- Courant max. par E/S: 40 mA
- Courant pour broches 3,3V: 50 mA
- Mémoire Flash: 32 KB (ATmega328) dont 0,5 KB utilisé par le bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Vitesse horloge: 16 MHz

La figure 2.6 présente un Arduino Uno



Figure 2. 6 Arduino Uno

- Le reste du matériel est décrit dans le II.

3. Description des fonctionnalités et des fonctions du prototype

La fonctionnalité principale du projet feu tricolore est la régulation de la circulation. Les fonctions qui y sont utilisées sont :

- Allumer
- Éteindre
- Clignoter

4. Montage

Le premier montage électronique fait sur plaque d'essai st visible sur la figure 2.7.

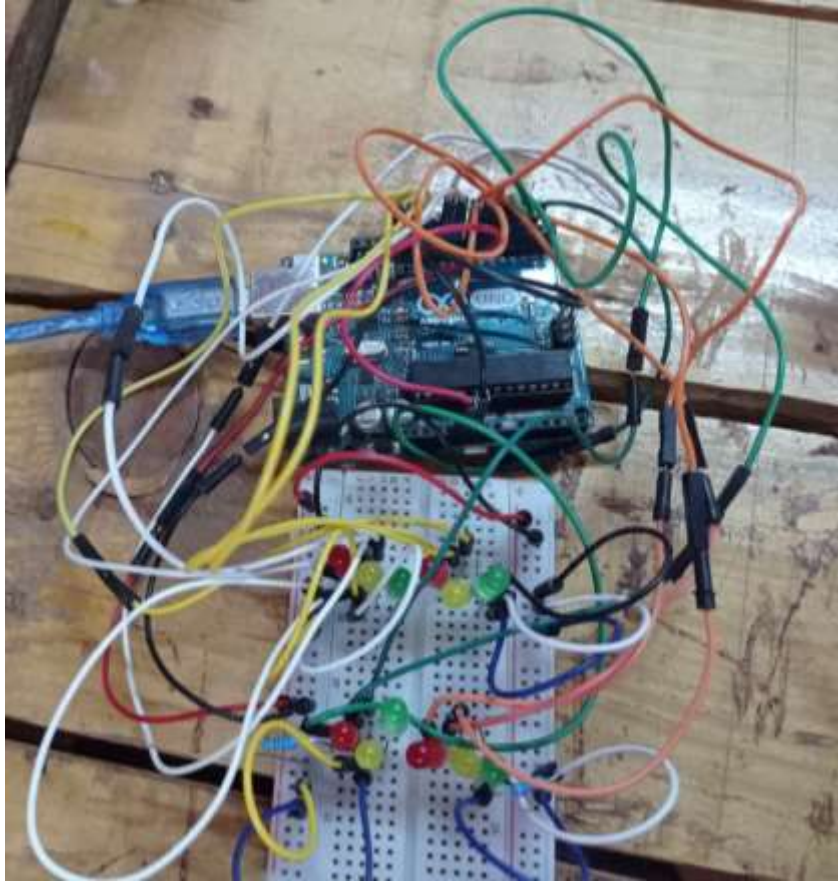
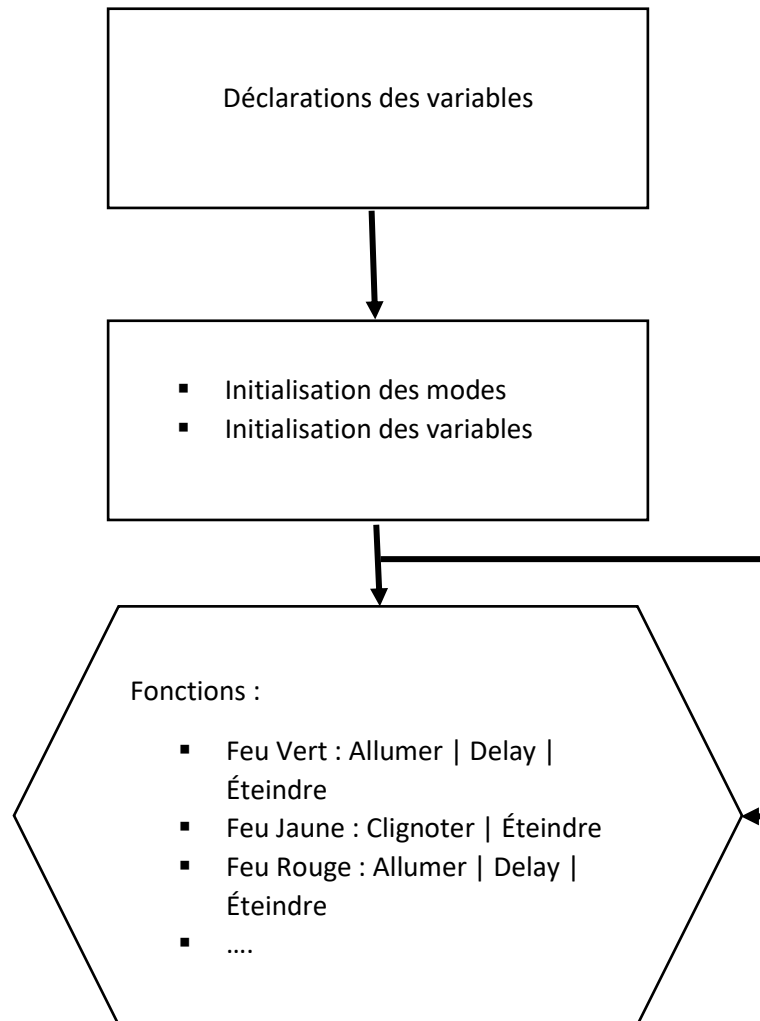


Figure 2. 7 Schéma physique du montage

6. Algorithme et Programme

Le programme a été écrit sur le model d'un algorithme assez simple.

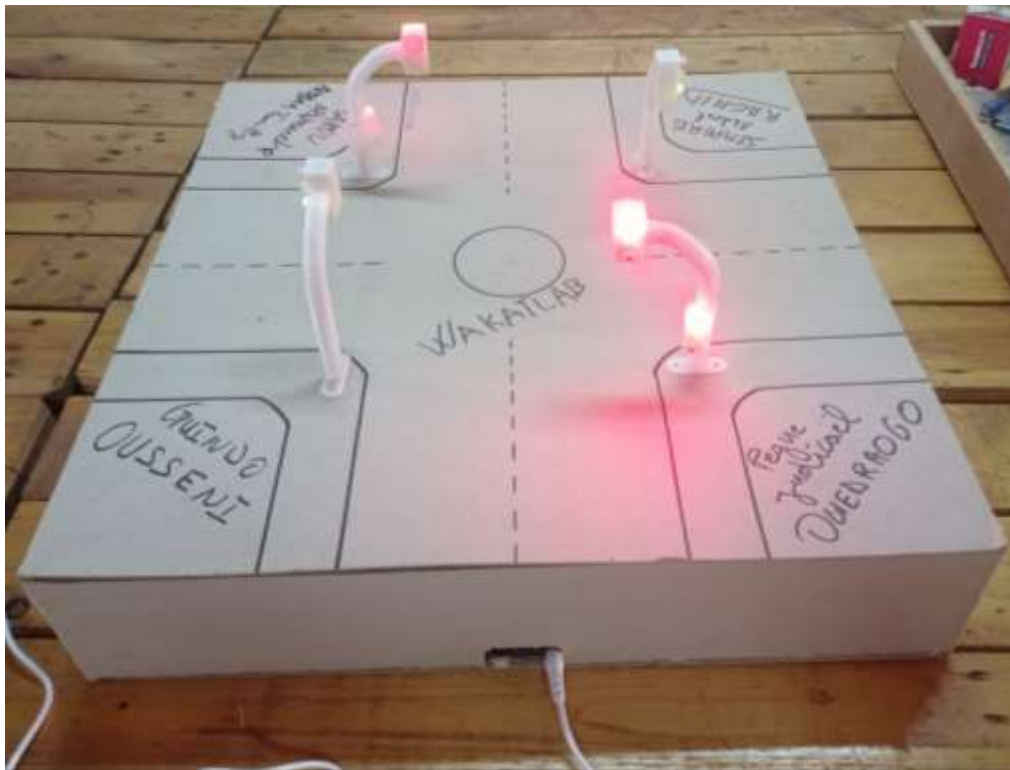
- L'algorithme



- Le programme
Le programme complet est disponible à l'adresse suivante :

https://www.dropbox.com/s/8h0vkffdjtorojr/feu_tricolore.rar?dl=0

- Le prototype final



III. Prototype 3 : Allumage automatique dans un entrepôt

Dans ce projet nous allons implémenter un système d'allumage automatique dans un entrepôt. Le but du projet est d'automatiser l'éclairage dans des endroits très fréquentés pour faciliter l'allumage ou la mise hors tension. L'éclairage automatique et l'éclairage à distance s'implante peu à peu pour faciliter le quotidien des hommes.

1. Le matériel nécessaire

Pour réaliser ce projet vous aurez nécessairement besoins des éléments ci-dessous :

- Un Arduino Uno
- 2 capteurs de mouvement infrarouge PIR HC-SR501
- 1 capteur de son
- Télécommande IR IRCO1
- Récepteur infra rouge
- 4 relais
- 4 ampoules, douilles, fils électriques
- 1 capteur de flamme
- Un buzzer

2. Description du matériel utilisé

- Arduino Uno, voir II
- Capteur de mouvement infrarouge PIR HC-SR501

C'est un module compatible avec Arduino. Nombreux usages possibles: Alarme, allumage automatique, sécurité, détection de présence.

Ses caractéristiques techniques :

- Sensibilité et délais réglables par des potentiomètres.
- Taille du PCB: 32 * 24 * 18 mm
- Tension d'alimentation: 4.5 - 20Vdc
- Courant nominal: <60μA
- Tension de sortie: 3.3V TTL
- Distance de détection: 3 - 7m (Ajustable)
- Angle de détection: < 140°
- Délais (temps au niveau haut après détection de mouvement): 5 - 200s (Ajustable, par défaut 5s +/-3%)
- Temps de blocage: 2.5s (par défaut)
- Température de fonctionnement: -20°C - +80°C
- Trigger valeur basse: Trigger non périodique
- Trigger valeur haute: Trigger périodique
- Référence: HC-SR501

Le capteur de mouvement infrarouge Hc-SR501 est représenté sur la figure 2.8



Figure 2. 8 Le capteur de mouvement infrarouge Hc-SR501

- Capteur de son (Module microphone haute sensibilité KY-037)

Ceci est un module de détection son à microphone électret ultrasensible. Il est très pratique car il peut être réglé (trou 3mm).

On peut l'utiliser via les sorties A0 ou D0.

A0: sortie analogique simple.

D0: sortie n'émettant un signal que lorsque le son atteint un certain seuil (threshold). Il est équipé d'un chip amplificateur LM398 et d'une résistance réglable pour le seuil de sensibilité.

CARACTÉRISTIQUES TECHNIQUES

- Signal de sortie: analogique.
- Tension de fonctionnement: 5VDC.
- Led d'indication de fonctionnement.
- Trou 3mm pour fixation.
- Dimensions: 38.5 x 15.5 x 12.5mm.

La figure 2.9 représente le capteur de son



Figure 2. 9 Module microphone haute sensibilité

- Télécommande IR IRC01

Cette télécommande infrarouge miniature est pratique pour commander un robot ou un autre projet. Elle dispose de 21 boutons dont 4 pour les directions.

Cette télécommande est notamment compatible avec les récepteurs IR ST027 et TSOP34838.

Vous pouvez utiliser cette télécommande pour commander un appareil compatible avec le codage NEC ou en combinaison avec un récepteur IR pour vos propres applications.

Une librairie compatible Arduino est disponible dans le gestionnaire de bibliothèques de l'IDE (IRremote).

Remarques:

- Attention, en fonction du modèle de télécommande, les codes hexadécimaux envoyés ne sont pas forcément identiques.
- Si cette télécommande est utilisée en remplacement d'une autre, il est important de vérifier la compatibilité des codes hexadécimaux envoyés.

Caractéristiques:

- Alimentation: 3 Vcc via pile CR2032 incluse
- Boutons: 21
- Fréquence: 38 kHz
- Longueur d'onde: 940 nm
- Codage: compatible NEC
- Dimensions: 86 x 40 x 7 mm
- Poids: 12 g

La figure 2.10 illustre la télécommande IR IRC01



Figure 2. 10 Télécommande IR IRC01

- Récepteur infrarouge VS 1838

Caractéristiques :

- Fréquence de fonctionnement: 38kHz

- Angle de captation: 45°
- Tension de fonctionnement: 5V

La figure 2.11 représente le récepteur infrarouge VS 1838



Figure 2. 11 Récepteur IR VS 1838

- Relais 10A
Ce module est compatible avec les cartes Arduino.
Vous pouvez faire des designs de domotique avec ce module.
Ce module relais fonctionne sous 5V et est compatible avec Arduino PIC ARM AVR DSP.
Il peut contrôler de nombreux équipements et appareils.
Il est possible de commander ces relais avec une tension inférieure à 5V. La manip a été testée avec succès sur les produits suivants :
 - NodeMCU LUA Amica R2
 - Wio Node
 - Linknode D1

CARACTERISTIQUES TECHNIQUES

Ses courants max sont:

- 10A pour 250Vac
- 10A pour 150Vac
- 10A pour 30Vdc
- 10A pour 28Vdc
- Taille: 3.9cm x 5.1cm

Le relais 10A est illustré sur la figure 2.12



Figure 2. 12 Relais 10A

- Capteur de flamme

Le capteur de flamme KY-026 est un capteur qui permet de mesurer des longueurs d'onde sur une plage comprise entre 760 nm et 1100 nm. Ce capteur réagira donc en présence d'une flamme.

Caractéristiques :

- Extrêmement sensible aux longueurs d'ondes entre 760-1100nm
- Seuil de détection de flamme modifiable par un potentiomètre
- Plage d'angle de détection: environ 60 degrés
- Alimentation: 3.3-5.5 VDC

Vous pouvez visualiser le capteur de flamme sur la figure 2.13



Figure 2. 13 Le capteur de flamme KY-026

- Buzzer

Les caractéristiques :

- Supporte le plug and play
- Faire des sons de buzzer et autre selon fichier MIDI
- Dimensions: 2.5 cm x 1.5 cm x 1.0 cm

Broches :

- 1 - Signal
- 2 - Vcc
- 3 - GND

- Ampoules, douilles, fils électriques
Trouver des ampoules de faibles consommation juste pour le montage.

3. Description des fonctionnalités et des fonctions du prototype

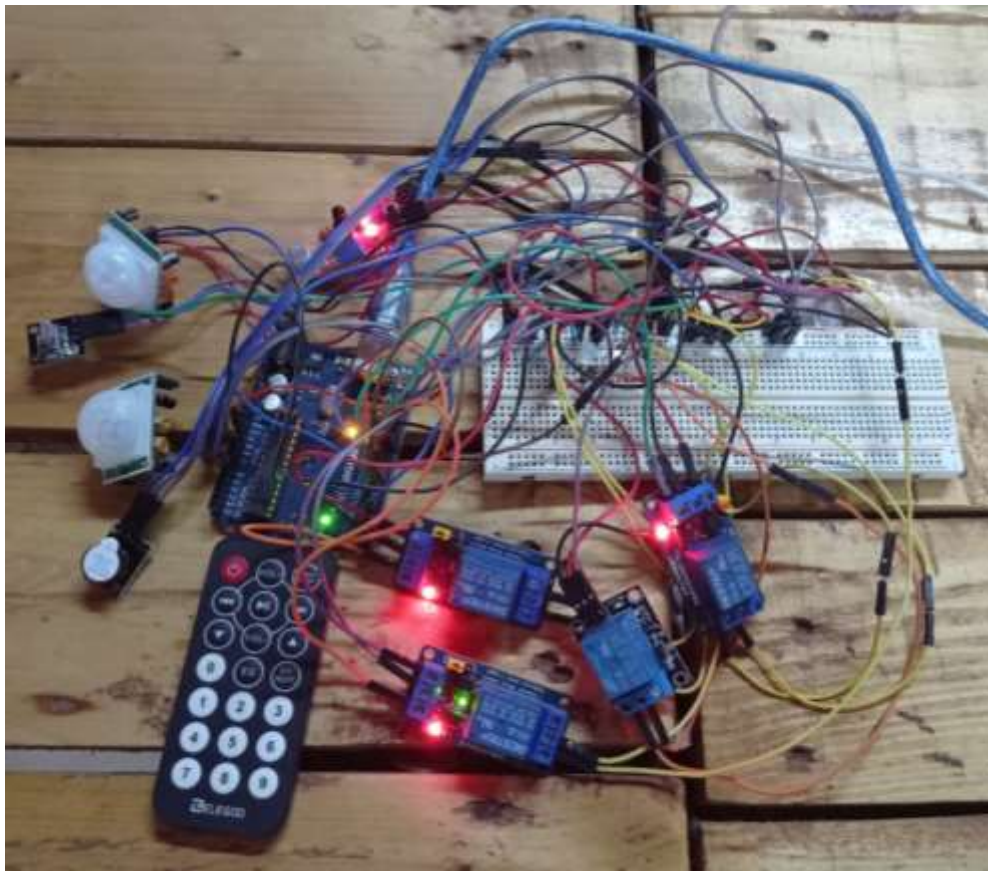
Les fonctionnalités principales dans l'entrepôt sont un système d'éclairage automatique soutenu par une alarme anti-feu.

Les fonctions qui seront activées dans ce projet sont :

- Allumer
- Éteindre
- Déclenchez l'alarme

4. Montage

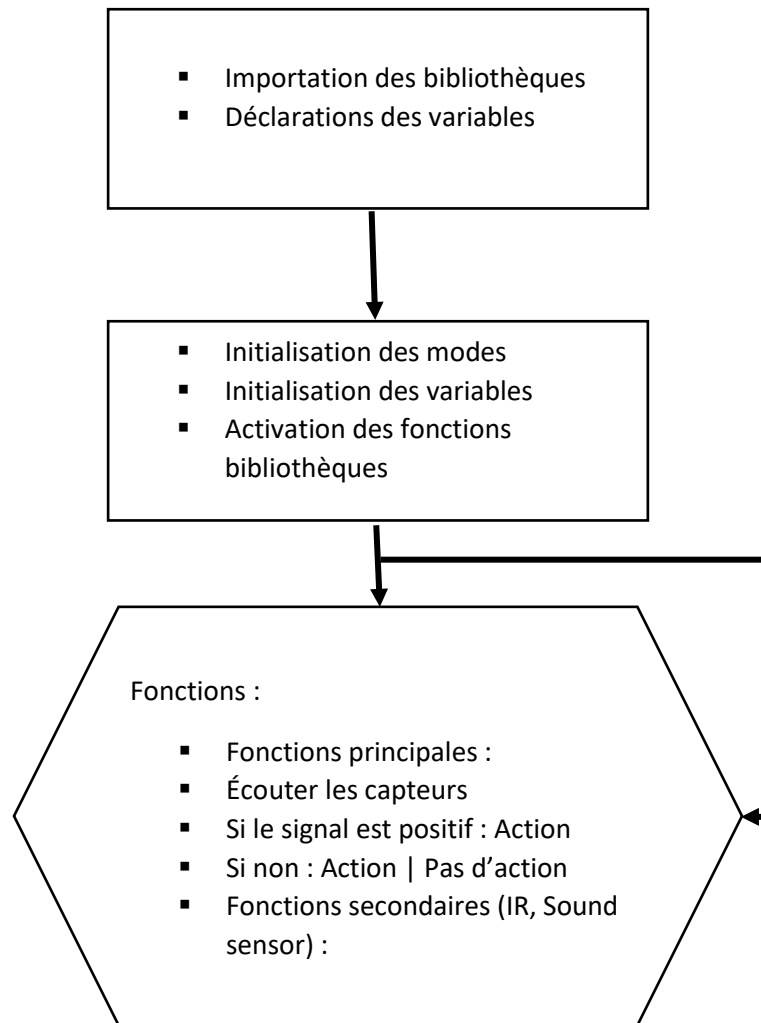
Le montage électronique du prototype est visible sur la figure 2.14



5. Algorithme et Programme

Le programme a été écrit sur le model d'un algorithme assez simple.

- L'algorithme



- Le programme
Le programme complet est disponible à l'adresse suivante :

<https://www.dropbox.com/s/s9cxy3cqbu98r7/smartLight.rar?dl=0>

- Le prototype final

