

**Université Euro Méditerranéenne Fès**

**Euro Mediterranean University of Fez**

**Ecole d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)**

**Projet de Fin de module**

**Filière :** 2ème année classes préparatoires Integrées

**Semestre** : 4

**Module :** Electronique embarquée

**Thème :**

Corbeille de Poubelle avec ouverture automatique.

Soutenu le .. /…/24,

**Encadré par : Préparé par les étudiants:**

Pr. A. SLIMANI - Layachi Ismail

- Mousmahi Yassir

* ***Remerciment***

Nous sommes ravis d'annoncer notre premier projet, qui vise à créer une poubelle intelligente. Ce projet est une solution innovante qui a pour but, réduire les déchets et améliorer la qualité de vie dans notre communauté

Tout d'abord, nous souhaitons exprimer notre gratitude envers notre professeur "ABDELLATIF SLIMANI" ainsi que toute l'équipe de direction d'EUROMED pour leur soutien, leur contribution et leur engagement envers notre projet. Sans leur aide et leur encouragement, nous n'aurions pas été en mesure de réaliser notre première étape avec succès. Nous sommes reconnaissants de leur confiance en notre équipe.

* ***Introduction***

Dans le cadre de notre deuxième année préparatoire en cycle d’ingénieur, nous avons entrepris un projet visant à mettre en pratique les connaissances acquises lors des séances du module "électronique embarquée". Notre objectif principal est de concevoir un prototype de poubelle automatique permettant une ouverture et une fermeture automatisées. En plus de cette fonctionnalité de détection, la poubelle automatique peut également être équipée de compartiments ou de séparateurs internes pour faciliter le tri des déchets.

**02**

**POUBELLE AUTOMATIQUE**

**POUBELLE AUTOMATIQUE**

**sommaire:**

**MINI-PROJET....................................................................................1**

**Remerciement.................................................................................2**

**Introduction.....................................................................................5**

**Section 1: Description du projet.....................................................6**

I-Analyse fonctionnelle........................................................................6

1-Expression et caractérisation du besoin.....................................6

II- Analyse fonctionnelle externe -Recherche de fonctions...............7

1-Diagramme pieuvre.......................................................................7

2-Diagramme FAST............................................................................8

3-Grafcet de la poubelle...................................................................9

**Section 2: Etude de la partie matérielle el la partie logicielle.....10**

I-la partie matérielle.............................................................................10

1-Carte ARDUINO................................................................................10

2-Module A fficheur LCD....................................................................11

3-Capteur ultrasons HC-SR04...........................................................13

4-Servomoteur...................................................................................15

5-Capteur infrarouge (d'obstacle)...................................................16

6-Capteur de proximité inductif......................................................18

7-Batterie............................................................................................21

I-la partie logicielle................................................................................22

1. Plateforme de programmation Arduino....................................22

# **03**

**POUBELLE AUTOMATIQUE**

**sommaire:**

**Section 3: Programme Arduino et simulation............................24**

1-Initialisation en Arduino (Allumer LED)........................................24

2-Utiliser un capteur ultra-son HC-SR04..........................................25

3-Utiliser un servomoteur:.................................................................27

4-Utilisation un capteur Ultra-son et une écran LCD.....................28

5-Utilisation un capteur servomoteur et ultrason..........................30

6-utilisation d'un capteur d'obstacle................................................34

7-Simulation de la poubelle................................................................34

**Coûts des matériaux.......................................................................38**

**Conclusion génerale........................................................................39**

# **04**

**POUBELLE AUTOMATIQUE**

**05**

**Introduction:**

Dans le cadre de notre deuxième année préparatoire en cycle d’ingénieur,nous

avons entrepris un projet visant à mettre en pratique nos connaissances acquises lors des

séances du module "science d’ingénieur ".

Notre objectif principal est de concevoir un prototype de poubelle automatique

permettant une ouverture et une fermeture automatisées. En plus de cette fonctionnalité

de détection, la poubelle

automatique peut également être équipée de compartiments ou

de séparateurs internes pour faciliter le tri des déchets.

nous avons divisé notre étude en plusieurs chapitres pour une meilleure compréhension

et organisation.

Dans un premier temps, nous commencerons par une description détaillée du projet,

comprenant la présentation du cahier des charges et une explication du fonctionnement

général du système. Cette première partie nous permettra de définir clairement les

besoins et les attentes liés à notre poubelle automatique.

Le deuxième chapitre sera consacré à la description du volet matériel de notre projet,

en mettant en évidence le choix du cœur du système. Nous examinerons également les

différents composants qui

constituent notre système, en expliquant leur rôle et leur

importance dans le bon fonctionnement de la

poubelle automatique. De plus, nous

présenterons les outils informatiques que nous avons utilisés tout au long du projet,

soulignant leur efficacité et leur contribution à sa réalisation.

Enfin, nous aborderons la programmation de l'ensemble du projet et la réalisation d'une

simulation sous Proteus ISIS. Cette dernière étape nous permettra de vérifier la faisabilité

de notre conception et d'apporter d'éventuelles améliorations avant de passer à la phase

de réalisation concrète.

# Section 1: Description du projet



Dans cette première section, nous allons vous introduire à notre projet en commençant par le cahier des charges. Ensuite, nous présenterons une analyse détaillée des besoins du projet, suivi de sa conception générale. Enfin, nous vous décrirons le fonctionnement global de notre projet.

**I-Analyse fonctionnelle:**

Cette approche consiste à identifier et à décrire les fonctionnalités proposées par notre produit pour répondre aux attentes de l'utilisateur.

1-Expression et caractérisation du besoin:

La bête à corne permet de définir la fonction principale de la poubelle automatique en posant les questions: Sur quoi agit la siroperie? et A qui rend-il service?

**utilisateur**

**A qui rendre service?**

**Les déchets**

**Sur quoi agit-il?**

**Poubelle automatique**

**Pour vider moins souvent les poubelles, éviter les odeurs**

**nauséabondes**

**Dans quel but?**

*Figure 1: Diagramme bête à cornes*

## 06

**II- Analyse fonctionnelle externe -Recherche de fonctions:**

1-Diagramme pieuvre:

L'outil "diagramme pieuvre" est utilisé pour analyser les besoins et identifier les fonctions de service du projet.

En analysant notre projet, on peut en déduire le diagramme "pieuvre", graphique circulaire qui met en évidence les relations entre les différents éléments de l'environnement du produit. Ces différentes relations sont appelées les fonctions de services qui conduisent à la satisfaction du besoin.

Parmi les fonctions retenues, il y a les fonctions principales, qui sont notées FP, qui représentent l'action d'un élément du milieu extérieur (EME) sur un autre EME, par l'intermédiaire du système.

Capteur

**Poubelle automatique**

Esthétique

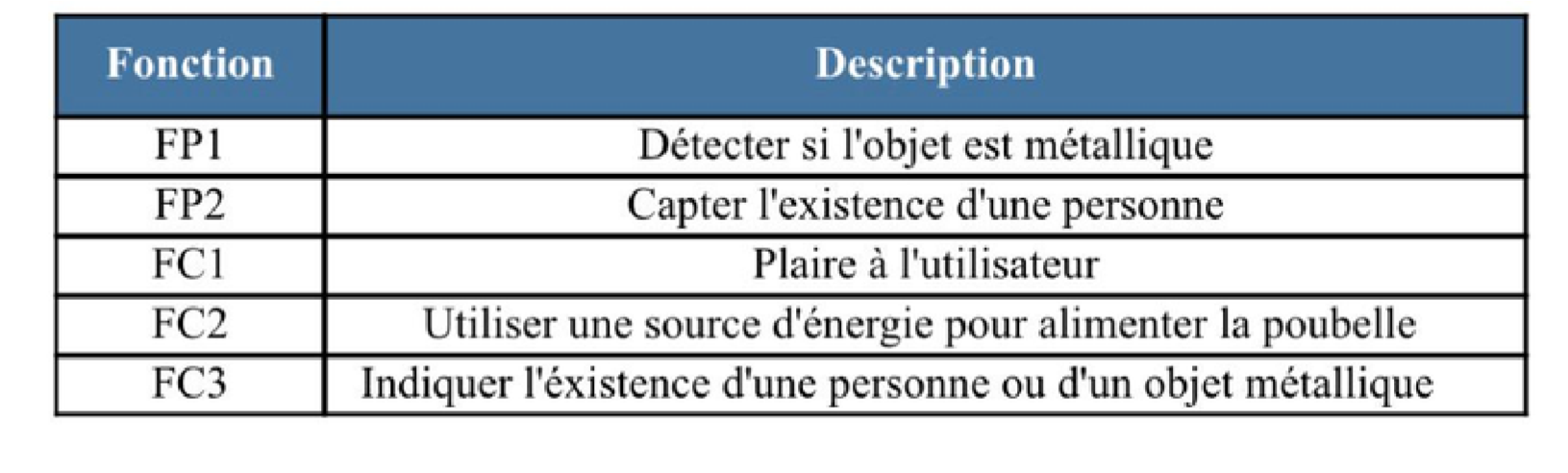
Déchets

Utilisateur

Energie

*Figure 2-dyagramme pieuvre*

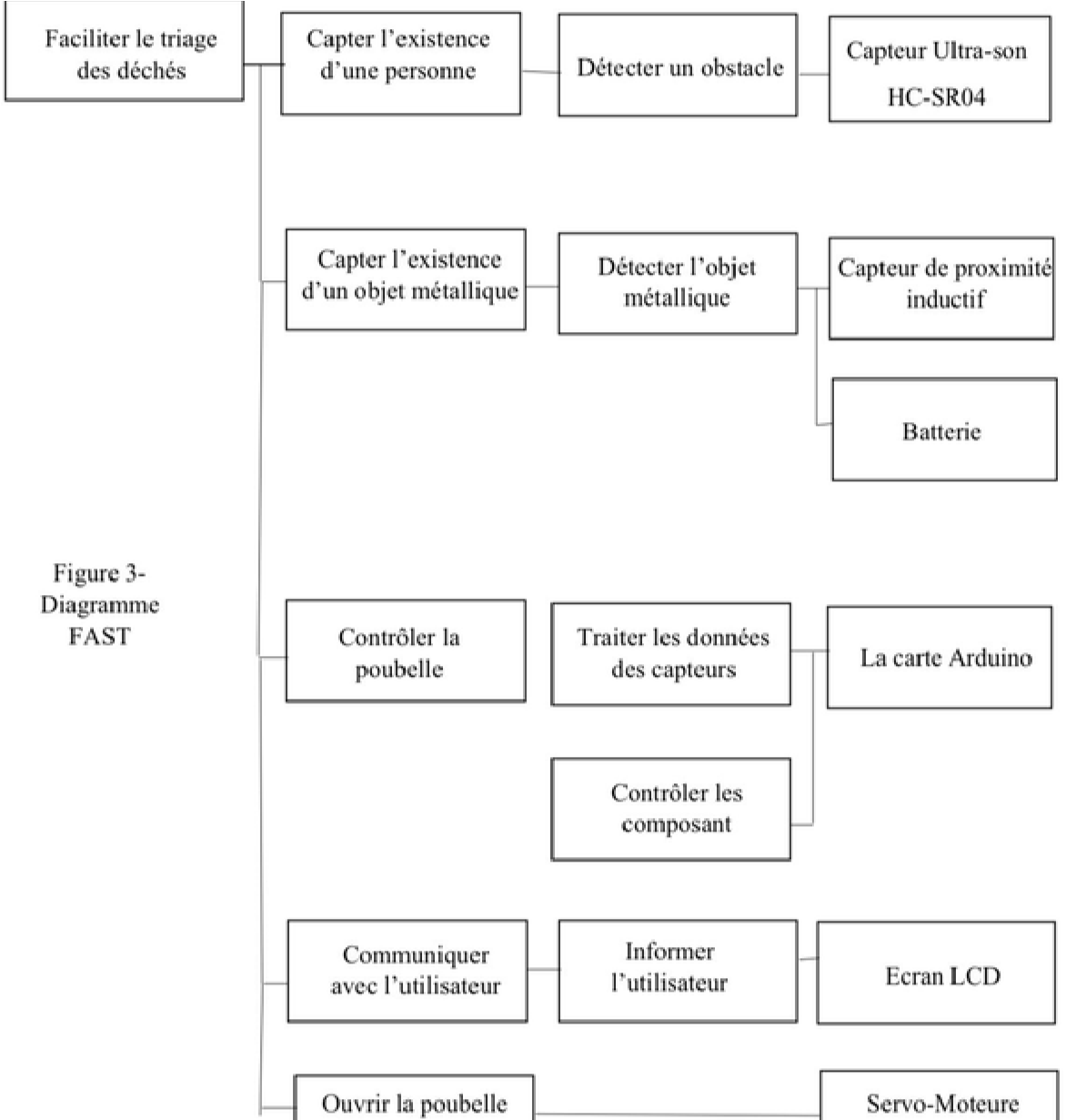
## 07



*Tableau 1-Désignation des fonctions*

2-Diagramme FAST:

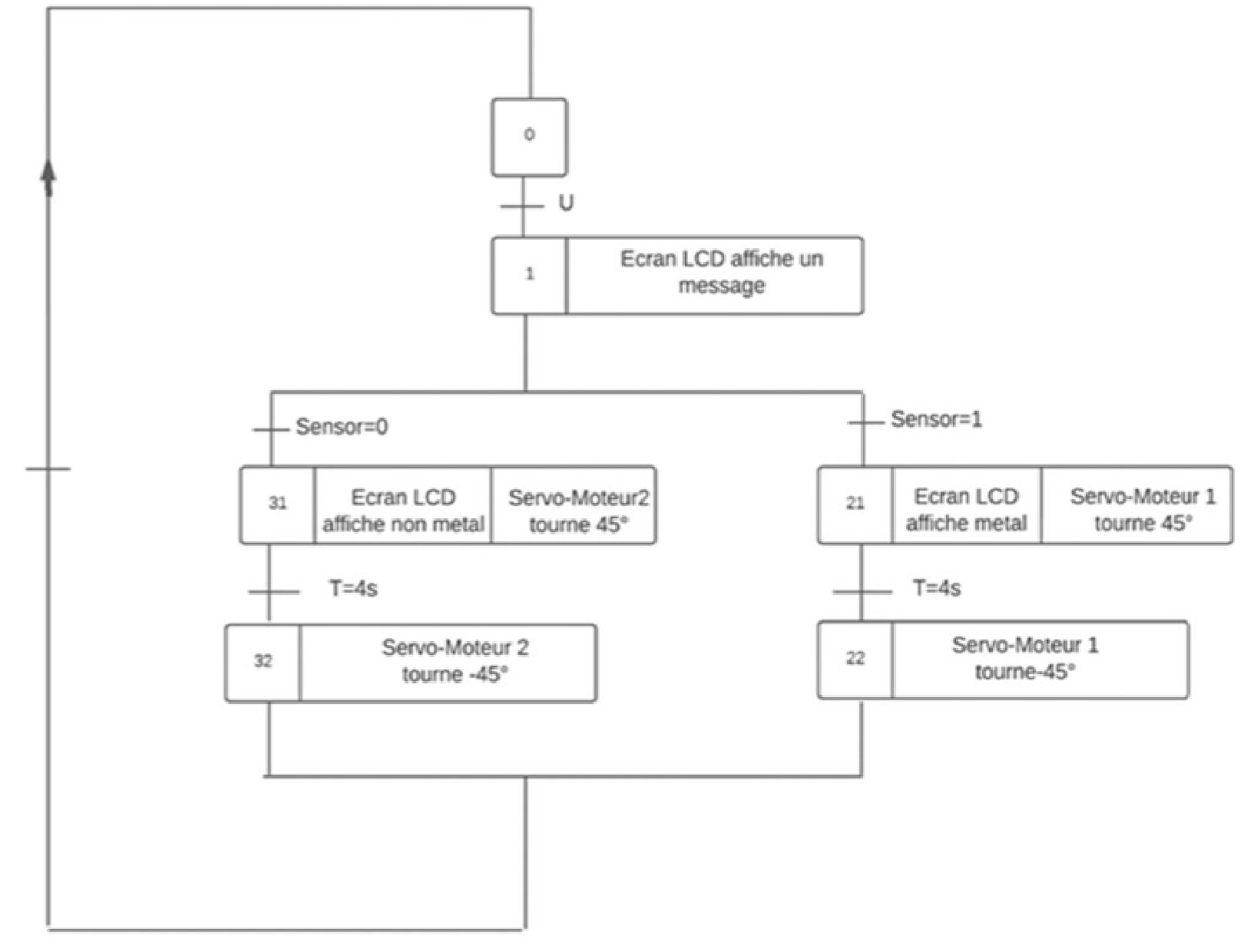
la fonction d'analyse du système technique et le schéma représentant les relations logiques entre les différentes fonctions fournissent une base solide pour comprendre en profondeur le fonctionnement d'un système. Ils facilitent les réponses aux questions "comment" et "pourquoi" en mettant en évidence les interactions et les dépendances entre les différentes composantes du système.



## 08

**POUBELLE AUTOMATIQUE**

3-GRAFCET de la poubelle:



*Figure 4- GRAFCET de la poubelle*

**Conclusion:**

La démarche d'analyse du système technique nous a permis de garantir la prise en compte de toutes les fonctions importantes du produit du projet, évitant ainsi les problèmes de sous-qualité. En identifiant et en comprenant les différentes fonctions du système, nous avons pu nous assurer qu'aucune fonction essentielle n'a été oubliée.

De plus, cette démarche nous a également aidés à éviter de concevoir des solutions qui ne répondent pas aux attentes des parties prenantes, prévenant ainsi les problèmes de sur-qualité. En comprenant les besoins et les exigences de l'environnement dans lequel le système sera utilisé, nous avons pu concevoir des solutions qui correspondent précisément aux besoins réels, sans ajout inutile.

## 09

**POUBELLE AUTOMATIQUE**

# Section 2: Etude de la partie matérielle el la partie logicielle



**I-la partie matérielle:**

Dans ce deuxième section , nous allons aborder les différents composants utilisés dans notre projet. Ces composants jouent un rôle essentiel dans la réalisation de notre objectif. Nous les avons soigneusement sélectionnés en fonction de leurs fonctionnalités, de leur compatibilité et de leur adéquation aux besoins spécifiques de notre projet.

Dans cette partie nous avons présenté les différents composants qui répondent à nos besoins dans notre projet.

1-Carte ARDUINO:

Arduino est un circuit imprimé en matériel libre (dont les plans de la carte ellemême sont publiés en licence libre mais dont certains composants sur la carte, comme le microcontrôleur par exemple, ne sont pas en licence libre) sur lequel se trouve un microcontrôleur qui 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, etc.

*Figure 5: Exemple d'une carte Arduino*

## 10

**POUBELLE AUTOMATIQUE**

**11**

**Composantes de la carte Arduino UNO:**

Une carte Arduino UNO se compose essentiellement de :

**• Prise jack:**

Permet de brancher une alimentation.

**• Microcontrôleur :**

stocke le programme et l’exécute.

**• Entrées analogiques .**

**• Connexion USB (Universal Serial Bus) :**

Permet d’alimenter la carte en

énergie électrique (5V). et de téléverser le programme dans la carte.

**• Entrées et sorties numériques (Digital) .**

Dans notre projet, nous avons choisi la carte Arduino Uno pour plusieurs raisons

importantes. Tout d'abord, la carte Arduino Uno offre une compatibilité élevée

avec une large gamme de composants et de modules disponibles sur le marché.

Cela signifie que nous pouvons facilement intégrer les capteurs et les actionneurs

nécessaires à notre projet sans rencontrer de problèmes de compatibilité majeurs.

De plus, bien que la mémoire Flash de la carte Arduino Uno soit de 32 ko, elle est

suffisante pour stocker le code nécessaire à notre projet. Nous avons

soigneusement optimisé notre code pour utiliser l'espace mémoire de manière

efficace et éviter les problèmes de saturation. Par conséquent, la capacité de

stockage de la carte Arduino Uno répond à nos besoins actuels.

2-

:

Module Afficheur LCD

Les afficheurs LCD alphanumériques présentent une solution facile d'emploi et

bon marché de doter notre projet d'une interface indépendante de notre PC.

Vu de l'extérieur, les écrans LCD alphanumériques sont essentiellement

caractérisés par leur taille. Deux modèles se rencontrent très fréquemment et

sont les meilleurs marchés, celui ayant 2 lignes et 16 colonnes d'affichage et

celui ayant 4 lignes et 20 colonnes d'affichage.



**POUBELLE AUTOMATIQUE**

**12**

**Figure 4: Afficheurs LCD (16x2) et (20x4)**

*Figure 6: Afficheurs LCD(20x4) et(16x2)*

**La connexion avec la carte Arduino uno:**

Dans le cas de la carte Arduino Uno, pour établir la connexion avec un module

d'affichage utilisant une communication sur 4 bits, nous devrons relier les broches RS,

EN, DB4, DB5, DB6 et DB7 de l'afficheur à des broches spécifiques de l'Arduino Uno. Il

convient de noter que la broche RW est optionnelle et peut ne pas être nécessaire

selon les besoins du projet.

**Afficheur LCD 16x2 :**

Dans notre projet, nous avons choisi cet afficheur LCD pour afficher un message

lisible à

l'utilisateur de la poubelle selon l'état de cette dernière.

Cet afficheur dispose des caractéristiques techniques suivantes :

**• Affichage:**

16 colonnes, 2 lignes.

**•**

**Tension :**

5V uniquement nécessaires.

**•**

**Courant:**

mA typ. avec rétro-éclairage allumé

135

**•**

**Modes :**

12

C ou 9 600 baud de communication série

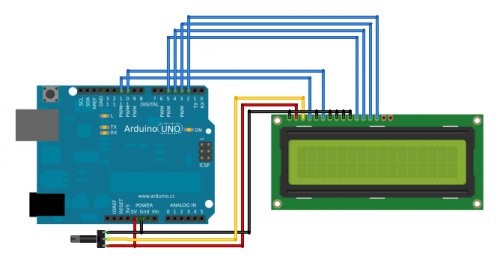
•

**Caractères personnalisés :**

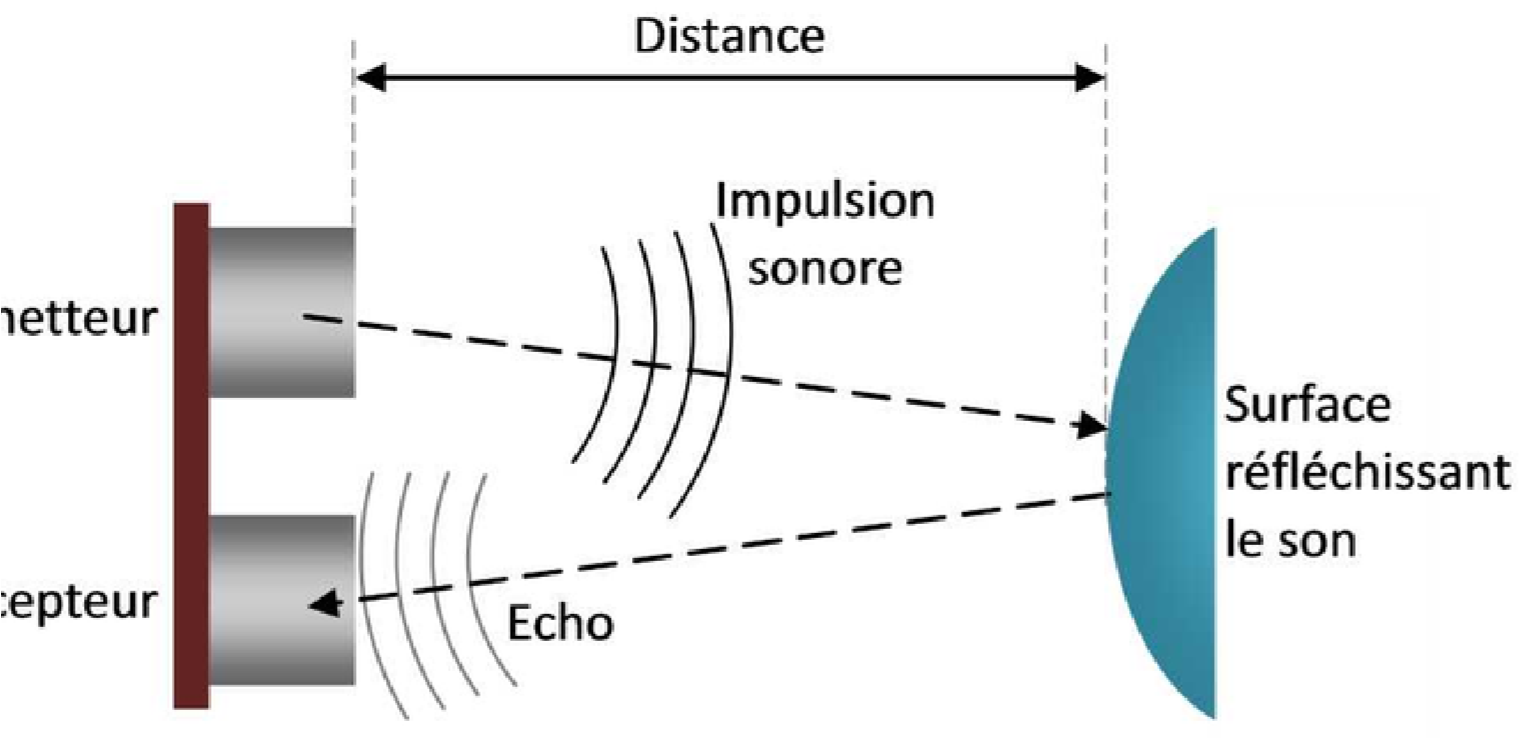
Jusqu'à 8 caractères personnalisés faciles à définir.

**POUBELLE AUTOMATIQUE**

**Branchement avec carte ARDUINO:**



*Figure 7: branchement de l'afficheur LCD avec ARDUINO* 3-Capteur ultrasons HC-SR04:



*Figure 8-Discription du capteur ultra-son HC-SR04*

## 13

**POUBELLE AUTOMATIQUE**

**14**

Le principe de fonctionnement du capteur est entièrement basé sur

la vitesse du son. Lors d'une prise de mesure, le capteur suit les

étapes suivant

:

**1-**

Une impulsion HIGH de 10μs est envoyée sur la broche TRIGGER

(

chrip) du capteur.

**2-**

Le capteur émet alors une série de 8 impulsions ultrasoniques à

une fréquence de 40KHz.

**3-**

Ces

ultrasons

se

propagent

dans

l'air

jusqu'à

rencontrer

un

obstacle, puis

ils rebondissent et reviennent dans la direction du

capteur.

**4-**

Le

capteur

détecte

l'écho

des

ultrasons

et

clôture

la

prise

de

mesure.

Pendant les étapes 3 et 4, le signal sur la broche ECHO du capteur

reste à HIGH. Cela permet de mesurer la durée nécessaire à l'aller-

retour des ultrasons, ce qui permet de déterminer la distance entre

le capteur et l'obstacle.

En mesurant la durée de l'impulsion HIGH sur la broche ECHO, nous pouvons

calculer la distance en utilisant la formule suivante : Distance = (Vitesse du son ×

Durée) / 2. Sachant que la vitesse du son dans l'air est d'environ 343 m/s à une

température de 20°C.

**Caractéristique du capteur ultra-son HC-SR04**

**Distance maximale de détection :**

4

cm

**Distance minimale de détection:**

2

cm

**Angle de détection**

:15

degrés

**Signal d'entrée de l'émetteur**

:

impulsion à l'état

haut de 10 us

**Signal de sortie du récepteur**

:

Signal numérique à l'état haut et la distance

proportionnellement

**Dimension :**

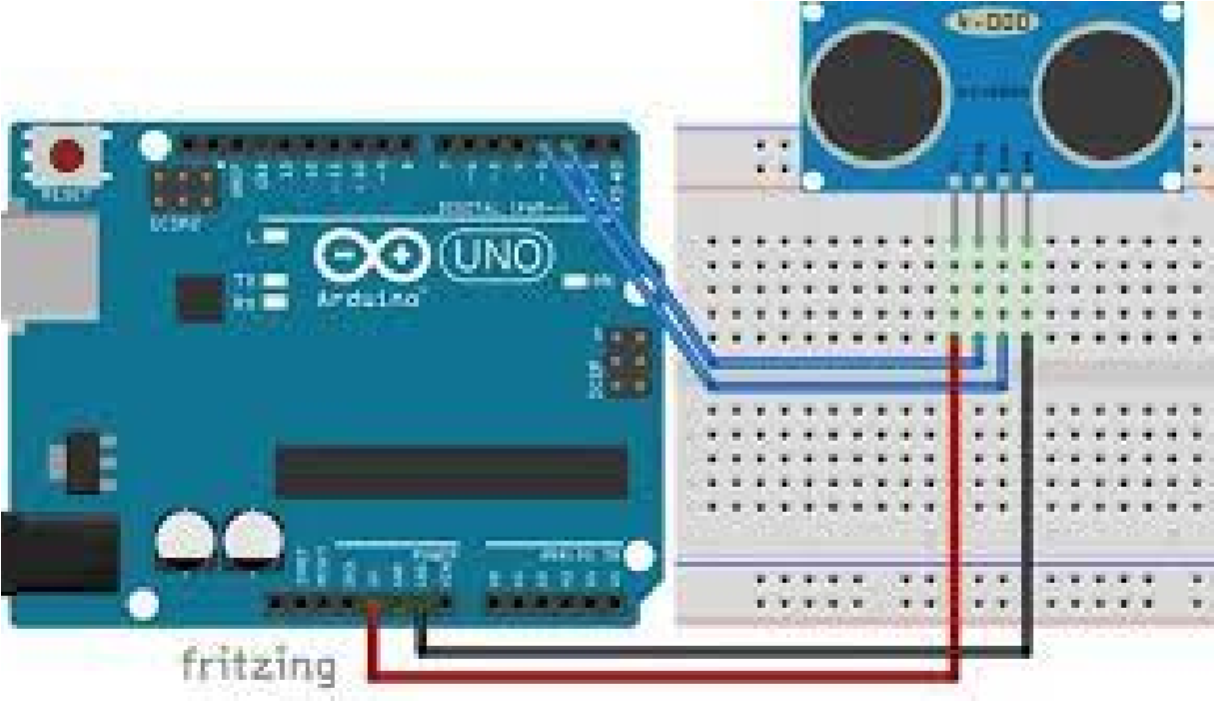
45\*20\*15

mm

Dans notre système le capteur ultrason joue le rôle d'un capteur de présence qui

détecte la présence de la personne

**Remarque:**



**POUBELLE AUTOMATIQUE**

**15**

4-

Servomoteur

**:**

Pour ouvrir et fermer notre poubelle, on a pensé à utiliser un servomoteur à

rotation

angulaire vu leur souplesse, simplicité de commande et de leur couple acceptable.

Le servomoteur à rotation angulaire est un micro servo-moteur largement utilisé

dans le

monde du modélisme. Il s'agit d'ailleurs du modèle disponible dans l'assortiment

moteur +Shield

m

oteur.

Très léger avec ses 9 grammes et son amplitude de 180°, il sera idéal pour animer

les projets

robotiques à partir d'un microcontrôleur tel qu'Arduino comme

dans notre situation.

Sa prise standard Graupner à 3 fils et sa tension d'alimentation de 5V rendent le

micro servo

directement utilisable sur un Arduino.

Le servomoteur a l'avantage, contrairement à un moteur à courant continu, d'être

asservi en position angulaire cela signifie que l'axe de sortie du servomoteur

respectera la consigne

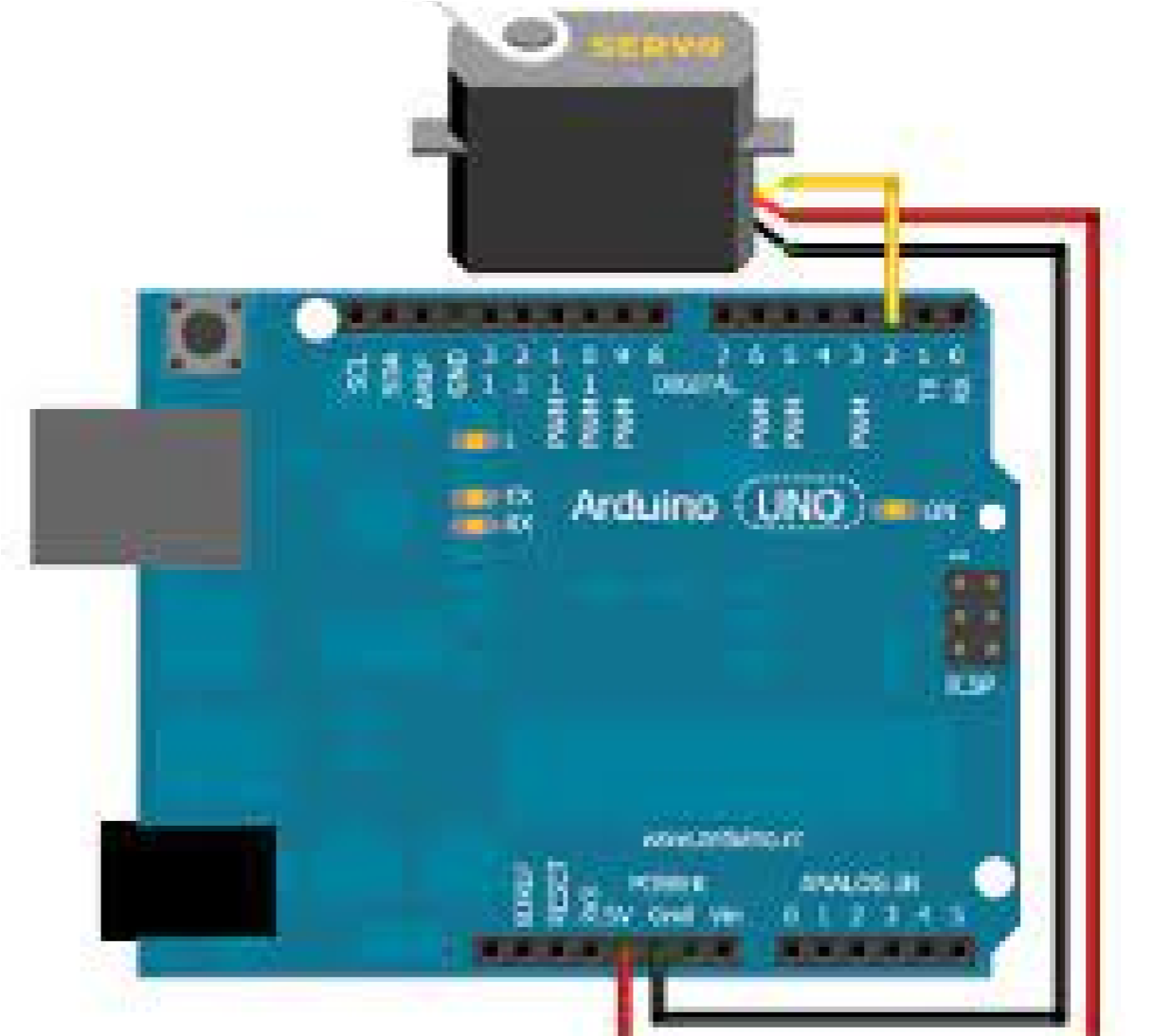
d'instruction que vous lui avez envoyée en son entrée.

**Branchement du capteur ultrason HC-SR04 avec Arduino**

*Figure 9:*

*Branchement du capteur ultrason HC-SR04*

*avec ARDUINO*



**POUBELLE AUTOMATIQUE**

**17**

Pour le moteur pas à pas, c'est un moteur qui bouge d'un pas à

chaque fois

qu'il reçoit une impulsion alors qu'un servomoteur est

un

moteur

pouvant

tourner

dans

les

deux

sens

et

équipé

d'un

équipement

de

traitement

d'information

qui

reçoit

des

ordres

d'ouverture et de fermeture en pourcentage, ce qui 'est compatible

avec notre projet.

Donc et pour répondre à notre besoin, on a choisi d'utiliser le

servomoteur

à

rotation

angulaire

(

micro-servo)

car

notre

application nécessite seulement une

rotation de 90° afin d'assurer

l'ouverture et la fermeture de la poubelle.

**Caractéristiques :**

**Alimentation :**

4,8 à 6V

**Angle de rotation :**

180°

**Couple :**

1

,3 kg.cm

**Vitesse:**

0,12 sec/ 60°

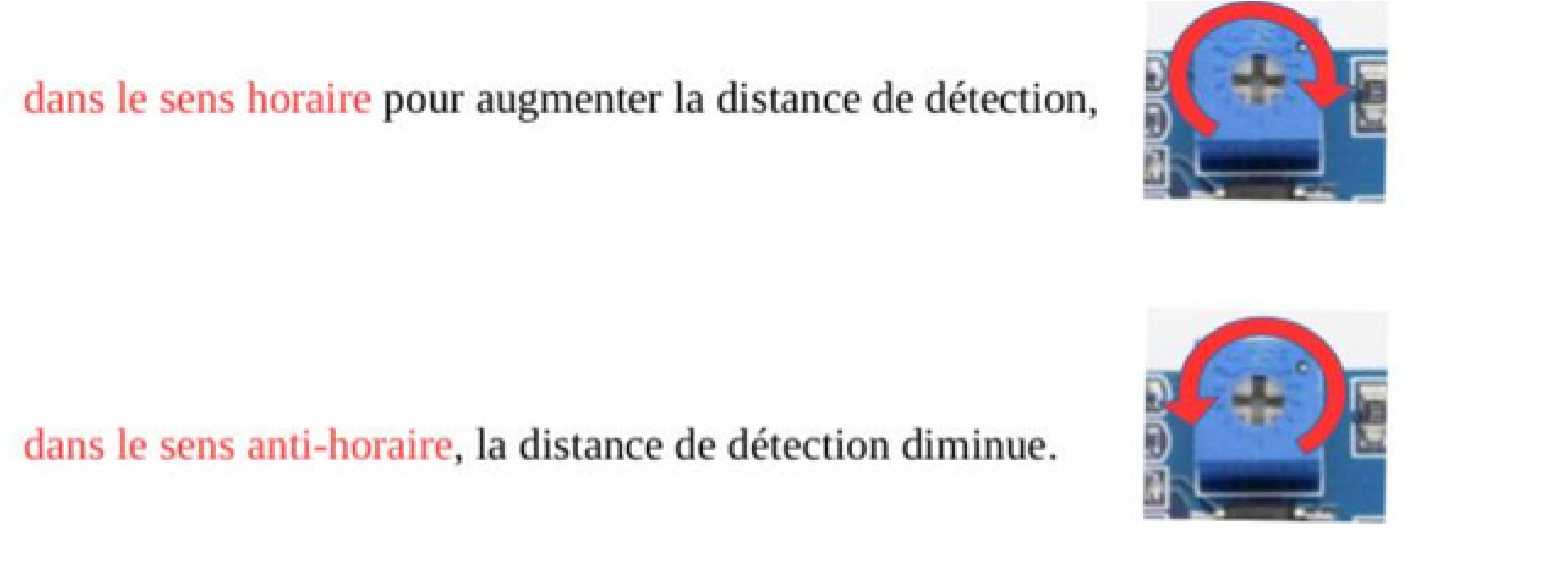
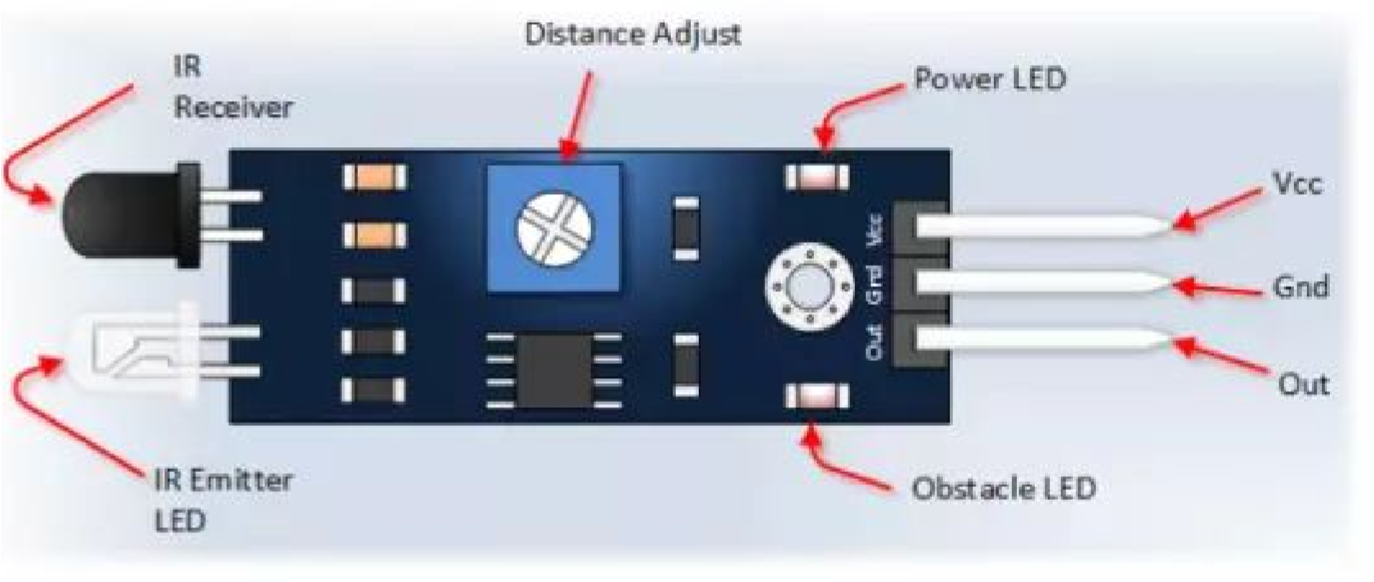
**Dimensions :**

23,2 X 12,5 X 22 mmPoids : 9g

**Branchement avec carte Arduino:**

*Figure 11-*

*Branchement du servomoteur avec ARDUINO*



**POUBELLE AUTOMATIQUE**

**16**

5-

Capteur infrarouge (d'obstacle

):

Pour détecter l’état de remplissage de notre poubelle, nous avons utilisé un

capteur infrarouge d’obtacle qui est un dispositif électronique conçu pour

détecter la présence d'obstacles physiques à l'aide de rayons infrarouges. Il

utilise un émetteur infrarouge pour émettre des faisceaux de lumière

infrarouge et un récepteur infrarouge pour recevoir ces faisceaux réfléchis

par les objets environnants. Lorsqu'un obstacle est détecté, le capteur

produit un signal électrique correspondant, indiquant la présence de

l'obstacle.

**Caractéristiques**

**capteur d’obstacle**

**Arduino:**

**–**

Distance de détection: de 2 à

cm

40

– Dimensions: 28 mm \* 23 mm

– Alimentation: 3.3 – 5 V

–

Réglage:

deux

résistances

variables à un tour

–

OUT:

interface

de

sortie

numérique

de

la

carte

(0

si

détection,

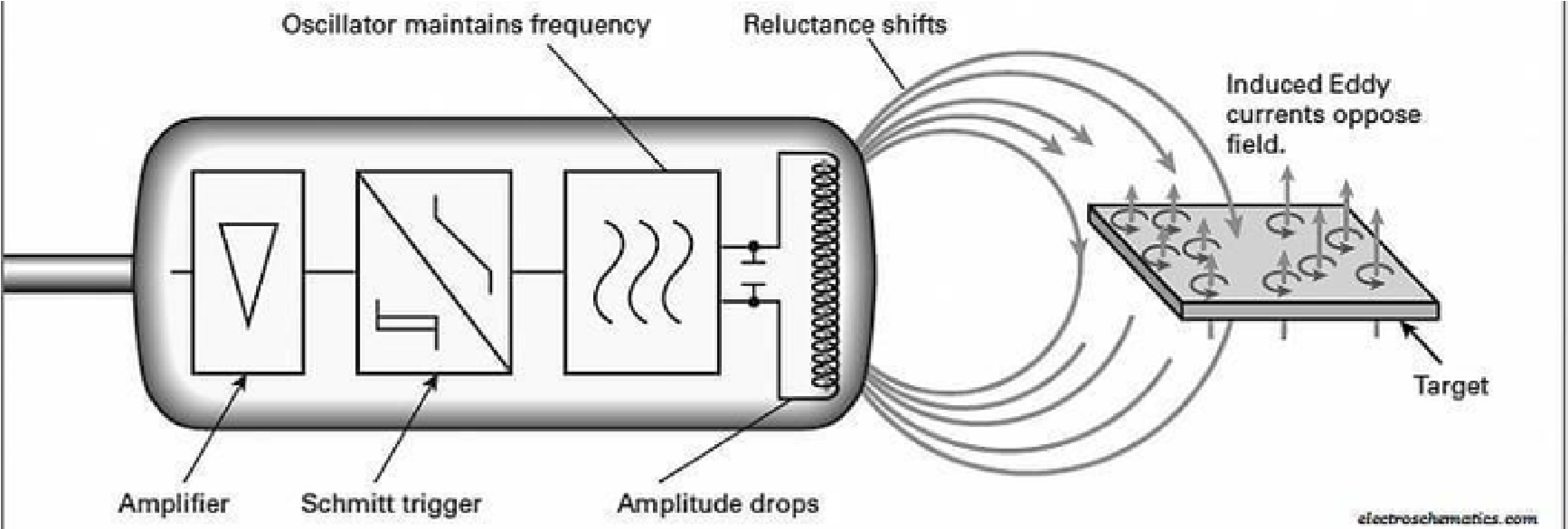
et

1

si

aucune

détection)



**POUBELLE AUTOMATIQUE**

**18**

6-

Capteur de proximité inductif

:

Un capteur de proximité inductif est un dispositif électronique qui

détecte la présence ou l'absence d'objets métalliques à proximité

immédiate.

Essayons de comprendre le principe de fonctionnement des capteurs de

proximité inductifs

**Comment Un Capteur De Proximité Inductif Détecte-T-Il Les**

**Objets Métalliques ?**

Les capteurs de proximité inductifs utilisent le même principe des courants de

Foucault pour détecter des objets métalliques. Ils mesurent la variation des

courants de Foucault induits dans l'objet présent et délivrent un signal en

conséquence.

Cependant, mesurer les courants de Foucault dans un objet à proximité est une

tâche compliquée. Par conséquent, les capteurs de proximité inductifs ont

également des circuits compliqués à l'intérieur pour traiter les signaux et fournir

une sortie décente.

Lorsqu'il est activé, le capteur crée un champ magnétique oscillant sur la face de

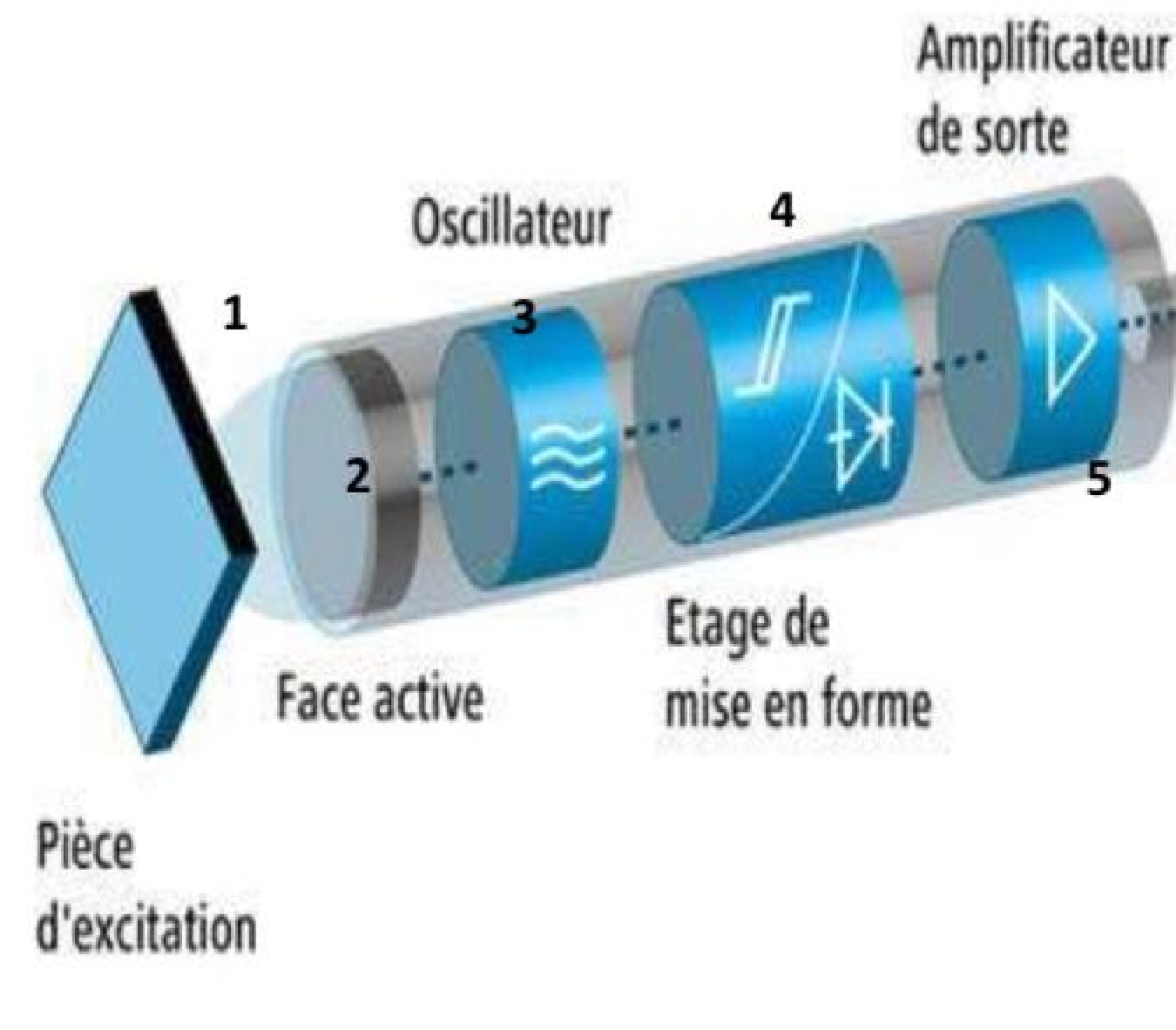
détection. Ce champ magnétique est créé à l'aide d'un Oscillateur LC, qui se compose

d'un condensateur et d'une bobine.

Un circuit spécial maintient la fréquence d'oscillation à une constante. Pour les

capteurs CA, cette fréquence est d'environ 10 à 20 Hz, tandis que les capteurs CC

fonctionnent dans la plage de 500 Hz à 5 kHz.



**POUBELLE AUTOMATIQUE**

**19**

Lorsqu'un objet métallique pénètre dans le champ magnétique

généré, le champ induit un courant électrique à l'intérieur de l'objet.

Cela provoque également la circulation de courants de Foucault à

l'intérieur de l'objet.

Comme mentionné précédemment, les courants de Foucault

perturbent le champ magnétique généré par le capteur.

Cette perturbation amortit l'oscillation naturelle dans le circuit

oscillateur. Ceci est également connu comme 'amortissement

magnétique'. L'amortissement magnétique augmente la charge sur

l'oscillation. À son tour, il réduit l'amplitude du signal oscillant.

Un circuit comparateur séparé surveille ce signal oscillant. Chaque fois

que l'amplitude du signal atteint au-dessous ou au-dessus d'un certain

seuil, le circuit active la sortie. Pour un capteur numérique, il s'agit d'un

signal de sortie logique HAUT ou BAS. Pour un capteur analogique, le

signal de sortie est un signal de courant ou de tension.

*Figure 12-Discription du capteur de proximité*

*inductif*

**Détail du fonctionnement:**

**1**

**:**

champ magnétique

**2**

**:**

bobinages

**3**

**:**

oscillateur

**:**

**4**

traitement du signal

**5**

**:**

amplification du signal

**POUBELLE AUTOMATIQUE**

**20**

7-

Batterie

:

Pour garantir le fonctionnement adéquat du capteur de proximité

inductif, il est nécessaire d'utiliser une tension plus élevée que celle

fournie par la plaque Arduino, qui est généralement de 5 volts. Ainsi,

afin de répondre à cette exigence, il est impératif d'intégrer une

batterie capable de fournir une tension suffisamment élevée.

**Remarque:**

Le rôle de la batterie est d'emmagasiner l'énergie produite par l'alternateur puis de

la restituer. Dans le cas où notre capteur de proximité inductif nécessite une

tension comprise entre 6V et 36V, il est nécessaire d'utiliser une batterie

appropriée. Pour assurer le bon fonctionnement du capteur sans le endommager,

nous avons choisi une batterie de 9V, qui se trouve dans l'intervalle requis de 6V à

36

V.

Cette batterie de 9V est donc adaptée à notre capteur de proximité inductif, car elle

fournit une tension suffisante pour son bon fonctionnement, tout en respectant les

spécifications de tension requises. Il est important de veiller à utiliser une batterie

dont la tension correspond aux exigences du dispositif afin d'éviter les problèmes

de sous-alimentation ou de surtension, qui pourraient endommager le capteur ou

compromettre sa précision.

En utilisant une batterie de 9V dans notre application, nous pouvons donc assurer

le bon fonctionnement du capteur de proximité inductif tout en respectant ses

spécifications de tension. Cela permet d'obtenir des mesures précises et fiables,

contribuant ainsi à la performance globale du système.



**POUBELLE AUTOMATIQUE**

**21**

*Figure 13- Pile de 9 v*

**Conclusion:**

Le processus de sélection des matériaux a été une étape complexe dans

notre projet. Il ne se limitait pas uniquement à des considérations techniques

répondant à des exigences fonctionnelles, mais il prenait également en

compte les préférences des utilisateurs dans un marché spécifique. En

substance, le choix des matériaux a été examiné en tenant compte de

l'ingénierie des matériaux, du temps de réponse et de l'aspect esthétique

(

design industriel

).



**POUBELLE AUTOMATIQUE**

**22**

**II-la partie logicielle**

**:**

1

. Plateforme de programmation Arduino

:

**Présentation:**

L'interface de l'IDE Arduino est plutôt simple, elle offre une interface minimale et

épurée pour développer un programme sur les cartes Arduino. Elle est dotée d'un

éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont

les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le

plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisé

pour accéder aux fonctions avancées de l'IDE.

Enfin, une console affichant les résultats de la compilation du code source, des

opérations sur la carte, etc.

La fonction

**« setup »**

contiendra toutes les opérations nécessaires à la

configuration de la carte (directions des entrées sorties, débits de

communications série, etc.).

La fonction

**« loop »**

elle, est exécutée en boucle après l'exécution de la

fonction setup. Elle continuera de boucler tant que la carte n'est pas mise hors

tension, redémarrée (par le bouton reset). Cette boucle est absolument

nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système

d'exploitation.

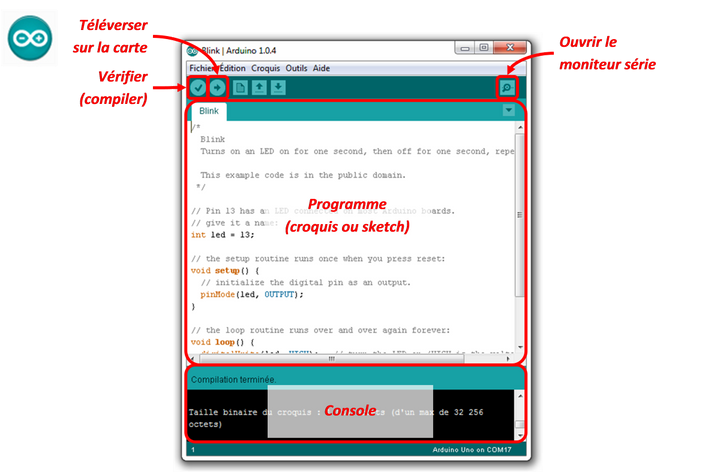
Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des

similarités avec le C, le C++, le Java et le Processing. Le langage impose une

structure particulière typique de l'informatique embarquée.

**POUBELLE AUTOMATIQUE**

En effet, si l'on omettait cette boucle, à la fin du code produit, il serait impossible



de reprendre la main sur la carte Arduino qui exécuterait alors du code aléatoire

*Figure 14- Interface de la plateforme Arduino*

**23**

**POUBELLE AUTOMATIQUE**

# **Section 3: Programme Arduino et simulation**



**Introduction:**

Le dernier volet de notre projet consistait à simuler notre poubelle, toutes les études ont été finalisées et les matériaux ont été préparés. Seul un châssis (la poubelle) manquait pour intégrer ces dispositifs et effectuer une vérification d'éventuelles erreurs.

## 1-Initialisation en Arduino (Allumer LED)

Commençons par un programme simple dans lequel notre objectif est d'allumer puis d'éteindre une LED pendant une seconde, de manière répétée. Nous avons réalisé un schéma correspondant à ce programme sur Tinkercad.



*Figure 15-Montage reliant une LED et un Arduino*

## 24

**POUBELLE AUTOMATIQUE**

**Le code Arduino utilisé:**

// Allumer une LED pour une seconde et l'éteindre pour une seconde \*/ void setup() {

// initialisation du digital pin 13 comme une sortie. pinMode(13, OUTPUT);

} void loop() { digitalWrite(13, HIGH);

delay(1000); digitalWrite(13, LOW);

delay(1000);

}

// allumer la LED

// attend d'une seconde

// atteind la LED

// attend d'une seconde

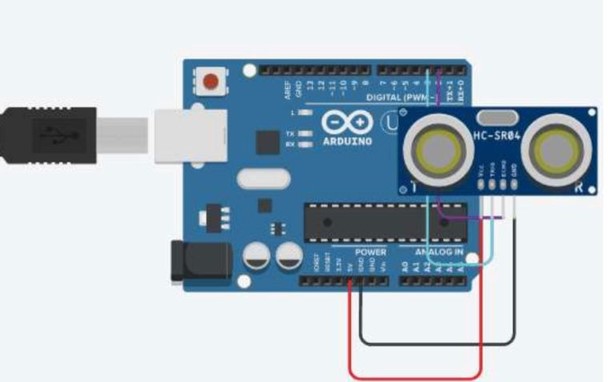
### 2-Utiliser un capteur ultra-son HC-SR04

Nous avons testé avec succès notre capteur en utilisant un code spécifique. Ce capteur joue un rôle essentiel dans notre projet, car il nous permet de détecter la présence d'objets et de calculer leur distance par rapport au capteur.

Ces fonctionnalités sont cruciales pour notre système, car elles nous permettront de déterminer si quelqu'un s'approche pour jeter quelque chose. De plus, elles nous aideront à évaluer la distance entre l'objet et la poubelle.

## 25

**POUBELLE AUTOMATIQUE**



*Figure 16-Montage reliant un Capteur Ultra-son et un Arduino*

**Le code Arduino utilisé:**

#define echoPin 2 attacher pin 2 Arduino a pin Echo #define trigPin

3 attacher pin Arduino a pin TRIG

int duration; // variable pour la durée du trajet de l'onde

int distance; // variable pour la mesure de distance void setup() {

pinMode(trigPin, OUTPUT); // définir pin comme une sortie

pinMode(echoPin, INPUT); // définir pin comme une entrée

Serial.begin(9600); // La communication série commence avec une

vitesse de 9600 vitesse de transmission

## 26

**POUBELLE AUTOMATIQUE**

Serial.println(" CAPTEUR ULTRASON 'HC-SR04' Test");

} void loop() {

digitalWrite(trigPin, LOW); // Efface la condition trigPin

delayMicr oseconds(2);

// Définit le trigPin HIGH (ACTIVE) pendant 10

microsecondes

digitalWrite(trigPin, HIGH); delayMicroseconds(10);

digitalWrite(trigPin, LOW);// lire return de echopin le temps de parcours de

l'onde en microsecondes

duration = pulseIn(echoPin, HIGH); distance = duration \* 0.034 / 2; // Vitesse de l'onde sonore divisée par 2 (aller et retour) Serial.print("Distance: ");// Affiche la distance

sur le Serial Monitor

Serial.print(distance);

Serial.println(" cm");

3-Utiliser un servomoteur:



*Figure 17- Montage reliant un servomoteur avec Arduino:*

## 27

**POUBELLE AUTOMATIQUE**

**Le code Arduino utilisé:**

#include <Servo.h>

Servo monServo; // création 'un nouveau servomoteur void setup() { monServo.attach(10); // On a utilisé la fonction monServo.attach(10) pour Attacher le servo moteur au ”pin” 10

monServo.write(45);// La fonction monServo.write(90) sert à Faire tourner le bras du servo moteur pour une angle de 90

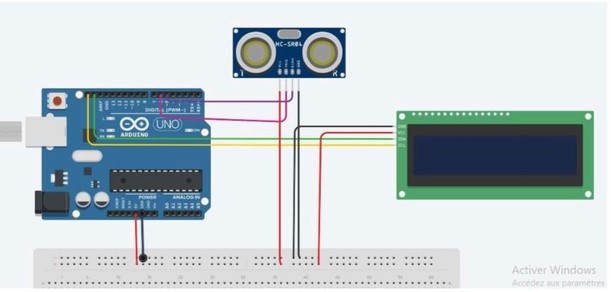
}

void loop(){

}

4-Utilisation un capteur Ultra-son et une écran LCD:

Pour mettre en évidence le fonctionnement du capteur ultrasonore, nous pouvons réaliser la simulation suivante sur Tinkercad. Nous allons connecter le capteur ultrasonore à la carte Arduino et utiliser un écran LCD pour afficher les résultats fournis par le capteur.



*Figure 18- Montage reliant un Ultra-son et LCD I2C avec Arduino*

## 28

**POUBELLE AUTOMATIQUE**

;

**Le code Arduino utilisé:**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2); const int trigPin = 6; const int echoPin = 5; long duree;

int distance;

int s; bool mouvementDetecte = false; unsigned long derniereDetection;

const unsigned long delaiInactivite = 6000; // 6 secondes

void setup()

{

Serial.begin(9600);

lcd.init(); lcd.backlight(); lcd.setCursor(0, 0); pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop()

{

digitalWrite(trigPin, LOW); delayMicroseconds(10); digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW); duree = pulseIn(echoPin, HIGH); distance = duree \* 0.034 / 2;

if (distance <= 10)

{

## 29

**POUBELLE AUTOMATIQUE**

mouvementDetecte = true;

derniereDetection = millis(); // Met à jour le moment de la dernière détection lcd.display(); // Allume l'affichage LCD lcd.setCursor(0, 1); lcd.println("Bienvenue"); delay(300);

lcd.clear();

}

if (mouvementDetecte && millis() - derniereDetection > delaiInactivite)

{

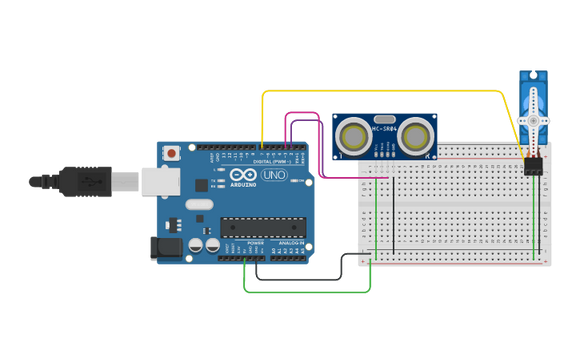
lcd.noDisplay(); // Éteint l'affichage LCD

mouvementDetecte = false; // Réinitialise le drapeau de détection de mouvement

}

}

5-Utilisation un capteur servomoteur et ultrason::



*Figure 19- Montage reliant un Ultra-son et Servomoteur avec Arduino*

**30**

**POUBELLE AUTOMATIQUE**

**Le code Arduino utilisé:**

#include <Servo.h>

Servo servo; // Création d'un objet de la classe Servo pour contrôler le servo-servo const int echoPin = 3; // Broche pour recevoir les échos du capteur à ultrasons const int trigPin = 2; // Broche pour envoyer les signaux de déclenchement au capteur à ultrasons

const int servoPin = 9; // Broche pour contrôler le servo-moteur int distance; // Variable pour stocker la distance mesurée par le capteur

int duration; // Variable pour stocker la durée du signal ultrasonique bool movementDetected=false;

void setup() {

sg90.attach(8); // Attache le servo-moteur à la broche 8

pinMode(trigPin, OUTPUT); // Définit la broche trig comme une sortie pinMode(echoPin, INPUT); // Définit la broche echo comme une entrée servo.attach(servoPin);

Serial.begin(9600);

}

void loop() {

digitalWrite(trigPin, LOW); // Envoie une impulsion basse à la broche trig pendant 2 microsecondes delayMicroseconds(2);

digitalWrite(trigPin, HIGH); // Envoie une impulsion haute à la broche trig pendant

10 microsecondes delayMicroseconds(10);

digitalWrite(trigPin, LOW); // Arrête l'impulsion

## 31

**POUBELLE AUTOMATIQUE**

duration = pulseIn(echo, HIGH); // Mesure la durée de l'impulsion écho distance = duration \* 0.034 / 2; // Calcule la distance en fonction de la durée mesurée

if (distance < 10 && movementDetected) {

servo.write(180); // Fait tourner le servo-moteur à 0 degré delay(1000); // Attend pendant 4,5 secondes

movementDetected=true;}

else if (distance>=10 && movementDetected){ servo.write(90);

delay(1000);

movementDetected=false;

Serial.print("Distance");

Serial.print(distance); Serial.print("cm");

delay(100); // Attend pendant 50 millisecondes

}

}

6-utilisation d'un capteur d'obstacle:

## 32

**POUBELLE AUTOMATIQUE**

**Le code Arduino utilisé:**

#include <LiquidCrystal\_I2C.h>

const int obstaclePin = 2; // Broche de données du capteur de distance infrarouge

const int lcdColumns = 16; // Nombre de colonnes de l'écran LCD const int lcdRows = 2; // Nombre de lignes de l'écran LCD LiquidCrystal\_I2C lcd(0x27, lcdColumns, lcdRows);

void setup() {

pinMode(obstaclePin, INPUT);

lcd.init(); // Initialisation de l'écran LCD lcd.backlight(); lcd.setCursor(0,0); lcd.print("Initialisation...");

delay(2000); lcd.clear();}

void loop() { if (obstacleDetected()) {

lcd.clear(); lcd.print("Pleine!!!!"); delay(2000); } else {

lcd.clear();}}

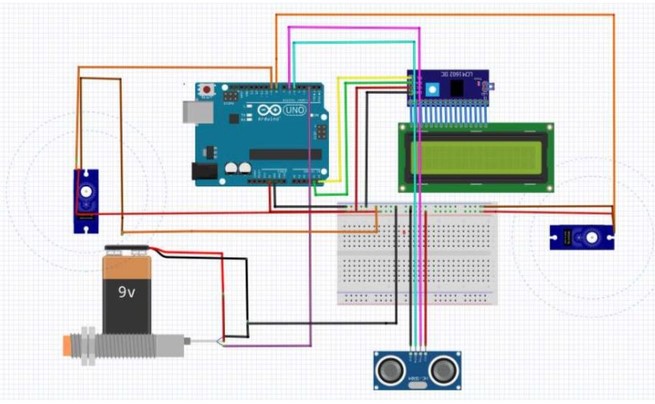
bool obstacleDetected() {

int obstacleValue = digitalRead(obstaclePin);

return obstacleValue == LOW;

}

## 33



**POUBELLE AUTOMATIQUE**

**34**

7-

simulation de la poubelle

:

Nous avons examiné attentivement le schéma de notre poubelle

intelligente afin de comprendre et de visualiser la disposition des

différents composants électroniques. Ce schéma nous a permis de

planifier le câblage de manière ordonnée et efficace, en veillant à

ce que chaque composant soit connecté correctement à la carte

ARDUINO.

*Figure 21-Shéma électrique de la poubelle:*

**Le code Arduino utilisé:**

include <Servo.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int obstaclepin = 11;

int l1 = 5;

int l2 = 6;

int l3 = 7;

int l4 = 8;

const int ACTION = 2;

const int trigPin = 3; // Broche TRIGGER de l'ultrason const int echoPin = 4; // Broche ECHO de l'ultrason const int servoPin = 9; // Broche de contrôle du servo moteur const int servo2Pin = 10;

Servo servo; // Création de l'objet servo Servo servo2;

bool movementDetected = false; // Variable pour suivre la détection de mouvement précédente

void setup() {

pinMode(obstaclepin, INPUT); pinMode(l1, OUTPUT); pinMode(l2, OUTPUT);

pinMode(l3, OUTPUT);

pinMode(l4, OUTPUT); pinMode(ACTION, INPUT\_PULLUP); pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT);

servo.attach(servoPin); // Attacher le servo moteur à la broche définie servo2.attach(servo2Pin);

lcd.init();

/ Position initiale du servo moteur  servo.write(90); servo2.write(90); Serial.begin(9600);} void loop() {

long duration, distance;

// Générer une impulsion de 10 µs sur la broche TRIGGER digitalWrite(trigPin, LOW);

delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW);

// Lire la durée de l'impulsion sur la broche ECHO duration = pulseIn(echoPin, HIGH); // Calculer la distance en cm

distance = duration \* 0.034 / 2;

## 35

// Vérifier si le mouvement est détecté if (distance < 10 && !movementDetected) {

digitalWrite(l1, HIGH); digitalWrite(l2, HIGH); delay(1500); lcd.clear(); lcd.setCursor(0, 0); lcd.println("BONJOUR "); delay(4000);

int val = digitalRead(ACTION);

if (val == LOW) {

digitalWrite(l4, HIGH);

lcd.clear(); lcd.prinln("metal"); delay(800);

// Tourner le servo moteur à 180 degrés

servo.write(180); delay(4000); servo.write(90);

movementDetected = true; // Mettre à jour le statut de la

détection de mouvement

} else {

digitalWrite(l3, HIGH);

l lcd.clear(); lcd.println("non metal"); delay(800); servo2.write(0); delay(4000);

servo2.write(90); } } else { digitalWrite(l1, LOW); digitalWrite(l2, LOW); digitalWrite(l3, LOW); digitalWrite(l4, LOW); movementDetected = false; }

servo.write(90); servo2.write(90); delay(4000);

bool obstacleDetected = digitalRead(obstaclepin) == LOW;

if(obstacleDetected){

lcd.clear(); lcd.print("pleine!!");

delay(4000);

}

else{ lcd.clear()}

**36**

**Conclusion:**

La phase de programmation de la carte ARDUINO et le câblage sont des étapes délicates, en particulier pour les personnes qui utilisent certains matériaux pour la première fois. Par conséquent, il était essentiel de se familiariser avec leurs programmations avant de pouvoir programmer notre poubelle intelligente automatique. Cette étape a exigé une attention particulière et une compréhension approfondie des composants électroniques utilisés, ainsi que des langages de programmation nécessaires pour les contrôler.

## 37

**Coûts des matériaux:**

|  |  |
| --- | --- |
| **MATÉRIELS**  Arduino  Afficheur LCD  Ultrason  Servomoteur x2  10 x LED 5mm Rouge  10 x LED 5mm vert Platque d'essaie capteur de proximité poubelle  65files de connexion  **TOTAL** | **PRIX**  140 dhs  55 dhs  25dhs  70 dhs  7dhs  7dhs  40 dhs  90 dhs  90 dhs  30 dhs  554 dhs |

**38**

**POUBELLE AUTOMATIQUE**

**Conclusion génerale:**

Dans le cadre de ce rapport, nous avons examiné en détail la conception et la production d'une poubelle intelligente dotée de fonctionnalités avancées.

Ce projet nous a permis de mettre en œuvre des concepts innovants de l'électronique et de l'informatique pour créer un produit moderne et pratique. Nous avons réussi à développer une poubelle qui s’ouvre automatiquement lorsqu'un utilisateur s'approche, de plus cette dernière est capable de détecter le type du produit jeté, Elle est ainsi en mesure d'identifier si le produit est métallique ou non, permettant un tri plus précis des déchets.

En conclusion, la réalisation de cette poubelle intelligente nous a permis de combiner différentes compétences et connaissances pour créer un produit innovant et utile. Nous sommes convaincus que notre poubelle intelligente apportera des avantages tangibles en termes de praticité, d'efficacité et de respect de l'environnement.

## 39