

UN RAPPORT DE PROJET SUR :

Système de Radar à Ultrasons

PAR:

Abdelhakim BRAHMI



NIVEAU: LICENCE 2 AUTOMATIQUE
DEPARTEMENT D'AUTOMATIQUE, TELECOMMUNICATIONS ET
ELECTRONIQUE
(ATE)
UNIVERSITE DE ABDERRAHMANE MIRA – BEJAIA

FEVRIER 2026

-
- La surveillance d'environnements complexes nécessite une attention constante, et les capteurs fixes classiques laissent souvent des angles morts importants.
 - Pour une navigation autonome ou une sécurité accrue, il est crucial de détecter les obstacles de manière dynamique sur un large champ de vision.
 - Ce projet présente un système de cartographie automatisé basé sur un capteur à ultrasons monté sur un servomoteur piloté par microcontrôleur.
 - En utilisant le principe de l'écholocation (similaire aux chauves-souris), le capteur émet des ondes sonores à haute fréquence. Le servomoteur effectue un balayage précis sur un arc de 180°, permettant de mesurer la distance des objets à chaque position angulaire.
 - Ce dispositif permet de générer une carte spatiale en temps réel de l'environnement immédiat. Il offre une solution économique et efficace pour la détection d'obstacles et l'aide à la navigation, réduisant ainsi les risques de collision dans des applications industrielles ou robotiques.

4-5	1. INTRODUCTION
	1.1. Introduction et Description du Système
	1.2. Problématique et Justification (Pourquoi ce projet ?)
6-7	2. LISTE DES COMPOSANTS
	2.1. Microcontrôleur (Arduino Uno R3)
	2.2. Capteur à Ultrasons (HC-SR04)
	2.3. Servomoteur (MG90s Micro Servo)
	2.4. Support et Connectivité
	2.5. Les Indicateurs Visuels : LED Rouge et Verte
	2.6. Résistances (220 Ohm)
	2.7. L'Alerte Sonore : Buzzer
8-9	3. ENVIRONNEMENT LOGICIEL
	3.1. Tinkercad Circuits (Conception et Simulation)
	3.3. Processing 4 (Interface Graphique - GUI)
	3.3. Processing 4 (Interface Graphique - GUI)
10-11	4. LE CHEMAINEMENT DU PROJET
	4.1. Étude Préliminaire et Conception Théorique
	4.2. Développement du Prototype et Intégration Matérielle
	4.3. Développement de l'Interface Graphique
	4.4. Tests Rigoureux et Optimisation du Système
12-13	5. SCHEMA DU CIRCUIT
	5.1. Alimentation

	5.2. Acquisition
	5.3. Sortie
	5.4. Guide des Couleurs de Câblage
14	6. ALGORITHME EN 4 ETAPES
	6.1. Émission
	6.2. Réception
	6.3. Traitement
	6.4. Note Technique
	6.5. Sécurité Logicielle
15-16	7. DEFIS ET SOUTIONS
	7.1. Défi: Gestion de la Rotation Continue
	7.2. Solution
	7.3. Pourquoi cette différence de vitesse (100 vs 83) ?
17-18	8. CONCLUSION ET PERSPECTIVES
	<ul style="list-style-type: none">- Compétences acquises- Évolutions futures

VISITEZ NOTRE SITE INTERACTIF SPÉCIALEMENT CONÇU POUR
PRÉSENTER LE PROJET "Système de Radar à Ultrasons"

Studytechbejaia.vercel.app

1. INTRODUCTION

1.1. Introduction et Description du Système

Le projet "**Système de Radar à Ultrasons**" consiste en la conception d'un système de télémétrie acoustique compact, capable de détecter et de cartographier des obstacles en temps réel. Contrairement aux systèmes de détection statiques, ce dispositif intègre un balayage mécanique pour offrir une vision spatiale étendue.

En exploitant les propriétés physiques des **ondes ultrasonores**, le système émet des impulsions acoustiques à haute fréquence. Lorsqu'une onde rencontre un obstacle, elle est réfléchie (phénomène d'écho). Le microcontrôleur analyse alors le "**Time of Flight**" (**ToF**) — le temps écoulé entre l'émission et la réception — pour calculer avec précision la distance et la position angulaire des objets environnants. Ces données sont ensuite traitées et projetées sur une interface graphique dynamique (type Radar PPI), imitant ainsi les systèmes radar militaires et industriels.

1.2. Problématique et Justification (Pourquoi ce projet ?)

Dans le contexte actuel de la **quatrième révolution industrielle** et de l'essor de la robotique mobile, l'autonomie des machines repose entièrement sur leur capacité à percevoir leur environnement.

- **Limites des capteurs traditionnels** : Les capteurs de contact (bumpers) sont réactifs et souvent insuffisants pour éviter des collisions à haute vitesse ou protéger des équipements fragiles.

- **Avantages de la détection sans contact :** La technologie par ultrasons offre une alternative robuste, capable de fonctionner dans l'obscurité totale et de détecter divers types de matériaux (solides, liquides, transparents).
- **Applications concrètes :** Ce type de technologie est au cœur des systèmes d'aide au stationnement des véhicules modernes, des drones d'exploration et des robots aspirateurs domestiques.
- **Enjeu pédagogique :** L'intérêt majeur est de démontrer qu'il est possible de simuler des principes radar complexes (échantillonnage, traitement de signal, asservissement de position) en utilisant des composants électroniques accessibles et à faible coût.

2.Liste des Composants

2.1. Microcontrôleur (Arduino Uno R3)

- **Rôle :** Il sert d'unité centrale de traitement. Il génère les signaux de déclenchement pour le capteur, calcule les distances et contrôle le mouvement du servomoteur.
- **Pourquoi ?** Choisi pour sa facilité d'interface avec les capteurs et sa gestion précise des signaux PWM.

2.2. Capteur à Ultrasons (HC-SR04)

- **Rôle :** Émet des ondes sonores et reçoit l'écho pour mesurer la distance.
- **Spécifications :** Portée de 2 cm à 400 cm avec une précision de 3 mm.
- **Principe :** Il utilise quatre broches (VCC, Trig, Echo, GND) pour communiquer avec le microcontrôleur.

2.3. Servomoteur (MG90s Micro Servo)

- **Rôle :** Assure la rotation physique du capteur.
- **Fonctionnement :** il permet un balayage étendu pouvant aller jusqu'à 360°

2.4. Support et Connectivité

- **Breadboard & Jumpers :** Pour établir les connexions électriques sans soudure, facilitant ainsi le prototypage.
- **Câble USB :** Pour l'alimentation du système et la transmission des données série vers l'ordinateur.

2.5. Les Indicateurs Visuels : LED Rouge et Verte

- **Rôle :** Fournir un état visuel instantané de la zone balayée sans avoir à regarder l'écran.

2.6. Résistances (220 Ohm) : Utilisées pour limiter le courant traversant les LED afin de protéger les sorties du microcontrôleur et prolonger la durée de vie des composants.

2.7. L'Alerte Sonore : Buzzer

- **Rôle :** Alerter l'utilisateur de manière auditive en cas de détection
- **Logiciel :** La fréquence du bip peut être programmée.

3. Environnement Logiciel

3.1. Tinkercad Circuits (Conception et Simulation)

- **Rôle :** Plateforme de prototypage virtuel.
- **Utilisation :** Avant l'assemblage physique, Tinkercad a été utilisé pour valider le schéma électrique et tester la logique du code.
- **Avantages :** Permet de vérifier que les résistances de 220 Ohm protègent correctement les LEDs sans risquer de brûler les composants réels.

Identification des erreurs de câblage dans un environnement sans risque.

3.2. Arduino IDE (Développement et Téléversement)

- **Rôle :** Environnement de Développement Intégré (IDE) utilisé pour écrire le code source en langage **C++**.
- **Fonctions clés :**
 - **Compilation :** Traduction du code textuel en instructions binaires compréhensibles par le microcontrôleur.
 - **Moniteur Série :** Outil crucial pour visualiser en temps réel les distances mesurées par le capteur HC-SR04 lors de la phase de test.
 - **Gestion des Bibliothèques :** Utilisation de la bibliothèque <Servo.h> pour piloter le moteur MG90s.

3.3. Processing 4 (Interface Graphique - GUI)

- **Rôle :** Logiciel de programmation visuelle utilisé pour créer l'interface "Radar".

- **Fonctionnement** : Processing reçoit les données de distance et d'angle envoyées par l'Arduino via le port série (USB). Il traduit ces chiffres en une animation graphique :
 - **Le Balayage** : Une ligne verte pivote sur 180° en synchronisation avec le moteur.
 - **Les Cibles** : Des segments rouges apparaissent sur l'écran pour indiquer la position exacte de l'obstacle détecté.
- **Résultat** : Transforme un simple montage électronique en un véritable outil de surveillance visuelle.

4. Le Cheminement du Projet

4.1. Étude Préliminaire et Conception Théorique

Cette phase initiale a été consacrée à l'analyse approfondie des besoins techniques. L'objectif était de définir l'architecture globale du système :

- Sélection des composants : Choix du servomoteur MG90s pour sa robustesse mécanique et du capteur HC-SR04 pour sa précision.
- Modélisation sous Tinkercad : Simulation des circuits électroniques pour valider les calculs de résistance (220 Ohm) et prévenir tout risque de court-circuit.

4.2. Développement du Prototype et Intégration Matérielle

Une fois la théorie validée, nous sommes passés à la construction physique :

- Assemblage Électronique : Câblage structuré sur breadboard des capteurs, LEDs, buzzer et microcontrôleur.
- Programmation (Arduino IDE) : Rédaction du code de contrôle. Cette étape incluait la gestion précise des signaux PWM pour le moteur et la capture des impulsions Echo du capteur.
- Calibration : Tests itératifs pour ajuster la précision des mesures de distance et la fluidité de la rotation du moteur.

4.3. Développement de l'Interface Graphique (UI) et Communication

Pour transformer les données brutes en informations exploitables, une couche logicielle supérieure a été ajoutée :

- Développement sous Processing 4 : Création d'un environnement visuel dynamique imitant un radar professionnel.

- Communication Série : Mise en œuvre d'un protocole de transfert de données entre l'Arduino et l'ordinateur.
- Interactivité : Programmation de l'affichage des cibles en rouge et du tracé de balayage en vert pour une lecture intuitive.

4.4. Tests Rigoureux et Optimisation du Système

La phase finale a consisté à éprouver la fiabilité du système

- Tests de l'Alert : Vérification de la précision des alertes sonores (Buzzer) et visuelles (LEDs).
- Optimisation Matérielle : Sécurisation des connexions physiques pour garantir la stabilité du système lors de la rotations.

5. Schéma du Circuit

5.1. Alimentation

- Source : Le système est alimenté via le port USB de l'ordinateur, fournissant un 5V stable au microcontrôleur.
- Distribution : L'Arduino redistribue cette tension aux composants via les rails de la breadboard. Le servomoteur MG90s et le capteur HC-SR04 partagent cette ligne d'alimentation.

5.2. Acquisition

- Capteur Ultrasons HC-SR04 :
 - Trig (Broche 7) : Signal de déclenchement envoyé par l'Arduino pour émettre l'impulsion sonore.
 - Echo (Broche 6) : Réception du signal de retour pour calculer la durée du trajet de l'onde.

5.3. Sortie

- Servomoteur MG90s : Connecté à une broche **PWM** (Pin 9) pour un contrôle de vitesse et de direction précis.
- Interface Visuelle (LEDs) :
 - Connectées aux broches 4, 5, 11 et 12.
 - Chaque LED est protégée par une **résistance de 220 Ohm** pour limiter le courant et protéger les sorties du microcontrôleur.
- Alerte Sonore (Buzzer) : Connecté à la broche 8.

5.4. Guide des Couleurs de Câblage

Rouge : Alimentation (5V)

Noir : Masse (GND)

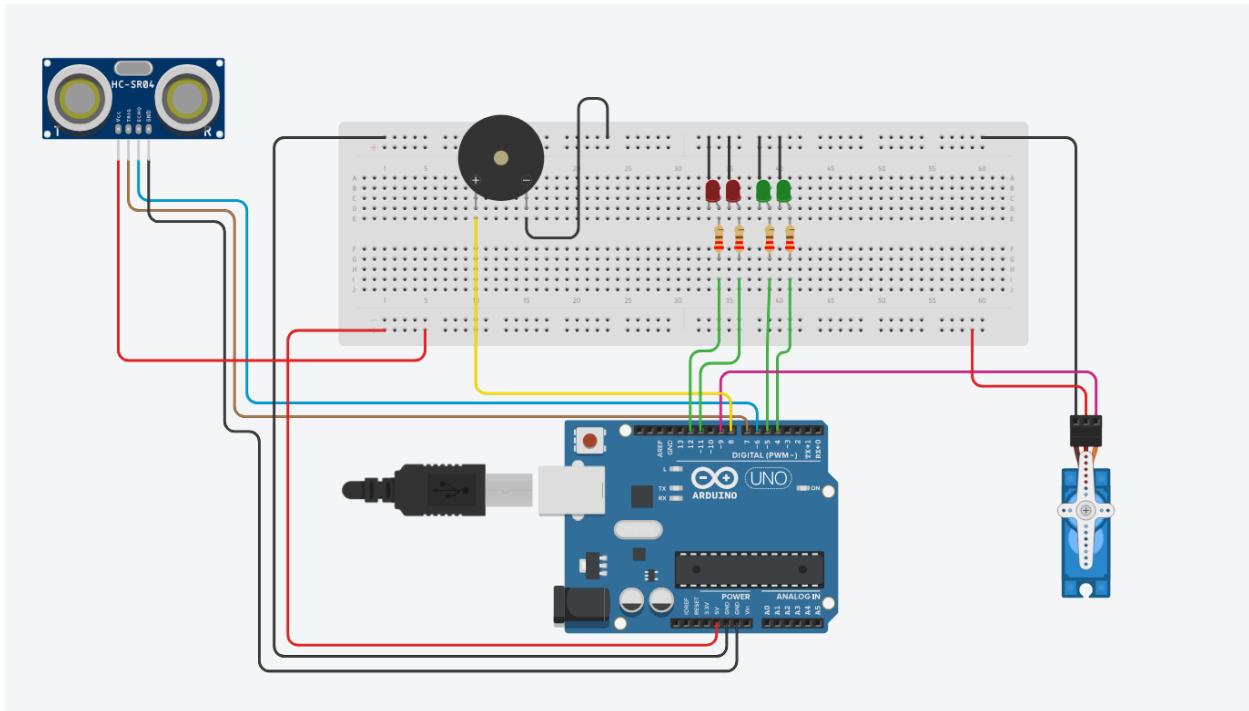
Bleu : Signal ECHO

Marron : Signal TRIG

Jaune : Signal Buzzer

Vert : Signal LEDs

Violet : Signal Servo Moteur



6. L'Algorithme en 4 Étapes

Voici comment le système traite l'information en temps réel pour assurer la détection :

6.1. Émission : Le capteur HC-SR04 envoie une salve d'ultrasons (40 kHz) par la broche **TRIG**.

6.2. Réception : Le signal rebondit sur l'obstacle et revient au capteur par la broche **ECHO**.

6.3. Traitement : L'Arduino calcule le temps de vol (t) et le convertit en distance [$d = (v * t) / 2$], où v est la vitesse du son.

6.4. Note Technique

La vitesse du son utilisée pour les calculs est de **0,034 cm/µs** pour obtenir des résultats précis en centimètres. Le servomoteur effectue un balayage sur une plage de **0° à 180°**, assurant une couverture radar optimale sans zones d'ombre.

6.5. Sécurité Logicielle

Le système est programmé pour rester en état d'alerte tant que l'obstacle est présent. Le balayage et le fonctionnement normal ne reprennent automatiquement que lorsque l'objet détecté est retiré de la zone critique.

7. Défis & Solutions

7.1. Défi: Gestion de la Rotation Continue

Le servomoteur **MG90s 360°** fonctionne en "boucle ouverte" : il ne possède pas de capteur interne pour confirmer son angle. La commande `servo.write()` ne définit plus une position, mais une **vitesse** et une **direction**.

7.2. Solution

J'ai implémenté une **logique basée sur le contrôle temporel**. En calibrant précisément la durée d'activation et la vitesse, j'ai pu simuler un balayage radar régulier malgré l'absence de retour de position.

Méthodologie de Calibration :

J'ai effectué des tests itératifs pour mesurer le temps nécessaire au moteur pour parcourir exactement un demi-cercle (180°).

7.2.1. Balayage Anti-horaire :

- **Vitesse réglée à 100** (Rotation lente dans un sens).
- **Résultat** : Il faut exactement **2,1 secondes** (`delay(2100)`) pour couvrir l'arc de 180°.

7.2.2. Balayage Horaire :

- **Vitesse réglée à 83** (Rotation lente dans le sens inverse).
- **Résultat** : Malgré une valeur de vitesse différente, le temps mesuré est également de **2,1 secondes**.

7.3. Pourquoi cette différence de vitesse (100 vs 83) ?

Les servomoteurs 360° ne sont pas parfaitement symétriques. Le "point mort" (arrêt) n'est pas toujours exactement à 90.

- Dans mon cas, l'écart par rapport au point mort (90+10 pour le sens 1 et 90-7 pour le sens 2) montre une **asymétrie mécanique** du moteur.
- La solution a donc été de compenser cette asymétrie par le code pour garantir que le radar reste bien "centré" et ne se décale pas au fil des rotations.

8. Conclusion & Perspectives

La réalisation de ce projet d **Radar à Ultrasons** a constitué une étape clé dans mon parcours d'apprentissage en ingénierie électronique. Ce travail m'a permis de concrétiser des concepts théoriques souvent abstraits en les confrontant à la réalité physique et aux imprévus du matériel.

L'un des aspects les plus enrichissants a été de synchroniser la logique logicielle (Arduino et Processing) avec le comportement mécanique des composants. J'ai appris que la programmation ne s'arrête pas à la syntaxe, mais qu'elle doit s'adapter aux tolérances et aux asymétries des composants physiques, comme l'a démontré la calibration précise de mon servomoteur **MG-90S**.

La contrainte de la rotation continue à 360° s'est transformée en une opportunité d'innovation. En remplaçant le contrôle de position classique par une **logique de calibration temporelle**, j'ai pu contourner une limitation matérielle pour aboutir à un système de balayage fluide et fiable. L'ajout d'une interface graphique interactive et d'un système d'alerte (LEDs et Buzzer) a permis de transformer un simple capteur en une solution de sécurité complète.

Compétences acquises

- **Capteurs** : Acquisition de données et conversion temporelle en données physiques (cm).
- **Systèmes Embarqués** : Architecture et programmation Arduino.
- **Signaux** : Gestion du PWM et de la logique temporelle.
- **Développement IHM** : Visualisation radar en temps réel via Processing.

Évolutions futures

- Remplacement de l'ultrason par un capteur **TF-Luna (LIDAR)**.
- Ajout d'une communication sans fil (**Bluetooth/Wi-Fi**).
- Optimisation de la consommation énergétique.

Ce projet confirme mon désir d'intégrer la **Licence EEA** de votre université. Je suis impatient d'approfondir mes connaissances en systèmes embarqués et en automatique au sein de vos laboratoires.

Pour découvrir les coulisses du projet, consulter la **galerie photos et vidéos**, ou télécharger l'intégralité du **code source** (Arduino & Processing), visitez mon site interactif :



studytechbejaia.vercel.app

Abdelhakim BRAHMI

Licence 2 Automatique

Universite de Bejaia

Email : abdelhakim.brahmi@tech.univ-bejaia.dz