

# SerbAI

Système de Contrôle de Robot Mobile via Bluetooth



**Étudiante:**

AHOUZI Hasnae

**Module:**

Développement Mobile

**Année/Filière:**

2026 - RC

**Encadrant:**

MOUNCIF Hamza

# ■ PLAN DE LA PRÉSENTATION

## 1 ⓘ INTRODUCTION

- ⭐ Contexte et objectifs du projet
- 📋 Présentation de SerbAI

## 2 ⚒ ARCHITECTURE TECHNIQUE

- ⌚ Composants matériels
- ↔ Stack Arduino/Android
- ↔ Protocole de communication

## 3 📱 smartphone APPLICATION ANDROID

- 📊 4 interfaces principales  
*monitor* Monitoring temps réel
- 🎬 Séquenceur de mouvements  
*settings* Configuration complète

## 4 🔑 FONCTIONNALITÉS CLÉS

- 🛡 Systèmes de sécurité  
*speed* Contrôle multi-modes
- ⟳ Synchronisation bidirectionnelle

# ⌚ Objectifs du Projet

## Objectifs Réalisés

- ✓ Développer un **système de contrôle robotique** sans fil fonctionnel
- ✓ Implémenter un **suivi de ligne automatique** avec 3 capteurs IR
- ✓ Créer une **interface Android professionnelle** et intuitive (4 activités)
- ✓ Intégrer des **fonctionnalités de sécurité** (arrêt d'urgence, détection obstacles)
- ✓ Permettre la **programmation de séquences** de mouvements
- ✓ Assurer une **communication bidirectionnelle** stable

## Contraintes Respectées

- ⌚ **Budget limité** (matériel de base uniquement)
- 💻 **Autonomie sans PC** (communication 100% Bluetooth)
- 👉 **Interface utilisateur** intuitive et responsive

# Architecture du Système

## Application Android (Java)

- ⌂ MainActivity - Contrôle principal
- ⌚ MonitorActivity - Surveillance temps réel
- 🎬 SequencerActivity - Programmation séquences
- ⚙️ SettingsActivity - Configuration
- \* BluetoothService - Communication

## Communication Bluetooth

- ↔ Protocole SPP (Serial Port Profile)
- ⌚ Module HC-05 (9600 bauds)
- ↔ Commandes ASCII bidirectionnelles
- ≡ Messages formatés structurés

## Robot Arduino Mega

- arduino mega 2560
- ⌚ 3x TCRT5000 (suivi ligne)
- ⌚ 1x HC-SR04 (obstacle avant)
- ⚡ L298N + 2 moteurs DC
- 🔋 Batterie 7.4V (2S Lithium)

# 🔧 Hardware - Composants Utilisés

Composant	Spécifications
Arduino Mega	Microcontrôleur ATmega2560, multiples pins I/O
HC-05 Bluetooth	Communication SPP, portée étendue, RX1/TX1
Capteurs IR	Suivi ligne (A0, A1, A2), seuil ajustable
HC-SR04 Ultrason	Détection obstacle avant (Trig:5, Echo:4)
L298N Pont H	Contrôle moteurs DC, puissance adaptée
Batterie	2S Lithium, autonomie prolongée
LED Pin 13	Indicateur de statut

## ⚡ Connexions Moteurs

← Moteur Gauche

ENA: 9

IN1: 8

IN2: 7

→ Moteur Droit

ENB: 10

IN3: 11

IN4: 12

ID 42274

# ☐ Application Android - 4 Interfaces Complètes

## 1 ☰ MainActivity Interface de Contrôle

- \* Connexion Bluetooth
- ↔ Modes Auto/Manuel/Stop/Urgence
- ❖ Contrôles directionnels (8 directions)
- ⌚ Gestion vitesse (3 niveaux)
- ⌚ Journal temps réel

## 3 📺 SequencerActivity Programmation de Séquences

- 🚶 Création de parcours
- #+#+ 6 commandes disponibles
- > Envoi séquentiel depuis Android
- ⏸ Pause/Reprise/Stop
- 📁 Chargement démo

## 2 ☰ MonitorActivity Surveillance Temps Réel

- 📏 Distance obstacle (HC-SR04) ✓
- 🕒 Valeurs capteurs IR ✓
- ⚡ Graphique interactif
- 🔔 Alertes automatiques
- ℹ Note: Batterie/vitesse théoriques

## 4 ⚙️ SettingsActivity Configuration

- 🔍 Scan appareils Bluetooth
- MAC Adresse MAC HC-05
- ⌚ Vitesse par défaut
- ⚙️ Préférences utilisateur
- 💾 Sauvegarde configuration

# MainActivity - Contrôle Total du Robot

**Robot Control**

● Connecté

Bluetooth Connecter Test

Auto Manuel Stop Urgence

Vitesse

↑ ← → ↓

Lent Moyen Rapide

Monitor Settings Sequencer

14:32:10 - Connexion établie avec HC-05  
14:32:15 - Mode: AUTO activé  
14:32:20 - Suivi de ligne en cours  
14:32:25 - Obstacle détecté à proximité  
14:32:30 - Arrêt automatique

- Zone Statut**  
Etat connexion temps réel (○/○)  
Alertes obstacles automatiques
- Connexion**  
Connecter/Déconnecter  
Test/Status robot
- Modes**
  - Auto (suivi ligne)
  - Manuel (contrôle total)
  - Stop (attente)
  - Urgence (arrêt immédiat)
- Pad Directionnel**  
Directions multiples  
Visible en mode manuel
- Vitesse**  
Niveaux réglables  
Contrôle PWM
- Navigation**
  - Monitor
  - Settings
  - Sequencer
- Journal**  
Historique horodaté  
Entrées limitées

# MonitorActivity - Surveillance Avancée

## Robot Monitor

● Connecté

### Distance Obstacle (HC-SR04)

45 cm

### Capteurs IR (TCRT5000)

420

Gauche

780

Milieu

380

Droit

### Graphique Distance



### Journal Capteurs

14:32:10 - Distance:45cm | IR(L:420 M:780 R:380)  
14:32:15 - Distance:42cm | IR(L:410 M:790 R:390)  
14:32:20 - Distance:38cm | IR(L:400 M:800 R:400)  
**14:32:25 - Distance:25cm | IR(L:380 M:820 R:420) - ALERTE**  
14:32:30 - Distance:30cm | IR(L:390 M:810 R:410)

### Batterie

Capteur non installé

### Capteurs Latéraux

Capteurs non installés

1

### Distance Obstacle (HC-SR04)

✓ Mesure **distance** en temps réel

Mise à jour **régulière**

Code couleur: Vert (sécurité), Jaune (attention), Rouge (danger)

2

### Capteurs IR (TCRT5000)

✓ 3 valeurs **analogiques**

Seuil ligne: **ajustable**

Rouge si ligne noire

Vert si fond blanc

3

### Graphique Distance

✓ Points **historiques**

Courbe interactive (zoom/scroll)

MPAndroidChart

Données réelles HC-SR04

4

### Journal Capteurs

✓ Format: **HH:mm:ss - Distance:XXcm | IR(L:XX M:XX R:XX)**

Alertes automatiques si obstacle proche

Dernières entrées affichées

5

### Batterie

✗ Affichage: "Capteur non installé"

✓ Extension possible: Diviseur de tension (résistances)

Module INA219

6

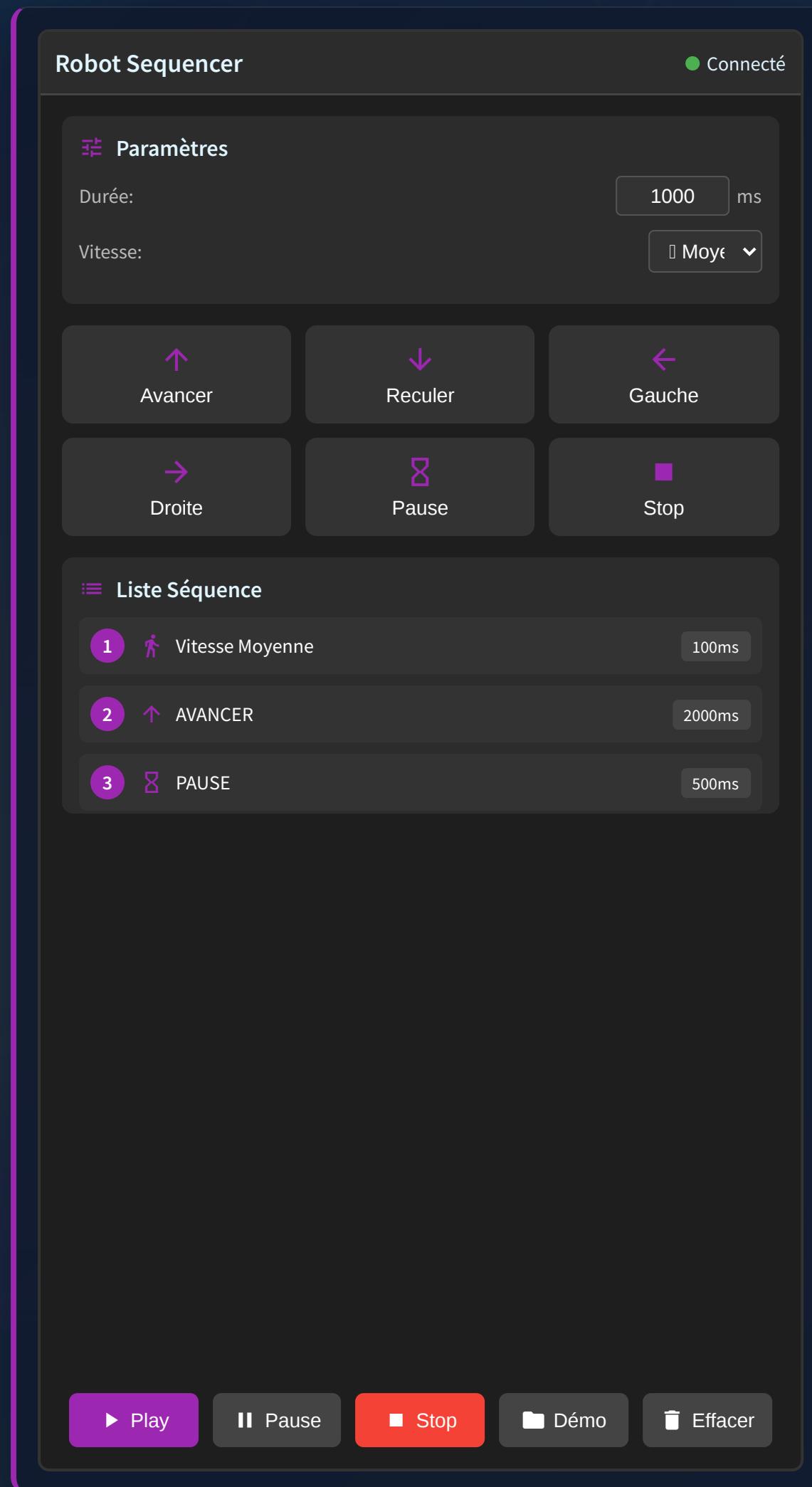
### Capteurs Latéraux

✗ Affichage: "Capteurs non installés"

✓ Extension possible: HC-SR04 supplémentaires

Detection 360°

# SEQUENCERACTIVITY - Automatisation des Parcours



## 1 Paramètres

- ✓ Durée ajustable
- ✓ Vitesse (Lent/Moyen/Rapide)

## 2 Commandes Disponibles

- ↑ Avancer
- ↓ Reculer
- ← Gauche
- Droite
- Pause
- Stop

## 3 Liste Séquence

- ✓ Format: N. ACTION - Durée - Vitesse
- ✓ Appui long = supprimer
- ✓ Nombre de commandes limité

## 4 Contrôles

- ▶ Play (démarrer)
- Pause (suspendre)
- Stop (réinitialise)
- Charger démo
- Effacer tout

## <> Exemple de séquence

- Vitesse Moyenne - 100ms
- ↑ AVANCER - 2000ms
- PAUSE - 500ms
- ← GAUCHE - 1000ms - □
- ↑ AVANCER - 1500ms - □
- STOP - 500ms

## Technique

Android envoie les commandes **une par une** avec les délais appropriés. Le robot reste connecté pendant toute l'exécution.

# ⚙️ SettingsActivity - Configuration Complète

**Robot Settings**

**\* Appareil Bluetooth**

00:14:03:06:57:3A 🔍 Scanner

Icone	Nom	Adresse MAC
Bluetooth	HC-05	00:14:03:06:57:3A
Bluetooth	Speaker	98:7B:F3:5A:12:8C

**⌚ Vitesse par Défaut**

Lent  
 Moyen  
 Rapide

**☰ Préférences**

Connexion automatique  
 Vibration lors commandes  
 Sons de confirmation

**⌚ Arrêt Automatique**

15s

**\* Test d'Urgence**

⚠️ Simuler Arrêt d'Urgence

💾 Sauvegarder ⟲ Réinitialiser

- \* Appareil Bluetooth**
  - ✓ Champ **adresse MAC**
  - ✓ Bouton **Scanner**
  - ✓ Liste appareils trouvés
  - ✓ Sélection par clic
- ⌚ Vitesse par Défaut**
  - ✓ Radio buttons: **Lent / Moyen / Rapide**
  - ✓ Appliquée au démarrage
- ☰ Préférences**
  - ✓  **Connexion automatique**
  - ✓  **Vibration lors commandes**
  - ✓  **Sons de confirmation**
- ⌚ Arrêt Automatique**
  - ✓ Curseur **réglable**
  - ✓ Sécurité anti-commandes involontaires
- \* Test d'Urgence**
  - ✓ Simulation **arrêt d'urgence**
  - ✓ Vérification sécurité
- 💾 Actions**
  - ✓ **Sauvegarder** (SharedPreferences)
  - ✓ **Réinitialiser** valeurs par défaut

# \* Protocole Bluetooth - Communication Bidirectionnelle

## > Android vers Robot

Commande	Action
T	Test connexion
A	Mode Auto
M	Mode Manuel
S	Stop
X	Arrêt d'urgence
F	Avancer
B	Reculer
L	Gauche
R	Droite
0	Stop moteurs
1/2/3	Vitesse
?	Status

## ← Robot vers Android

Message	Signification
TEST:OK	Connexion OK
MODE:XXX	Changement mode
OBSTACLE	Alerte
VOIE_LIBRE	Voie libre
IR:L,M,R	Capteurs IR
DIST:XX	Distance
ACTION:XXX	Confirmation
VITESSE:XXX	Vitesse
STATUS	État complet

## ↔ Spécifications Techniques

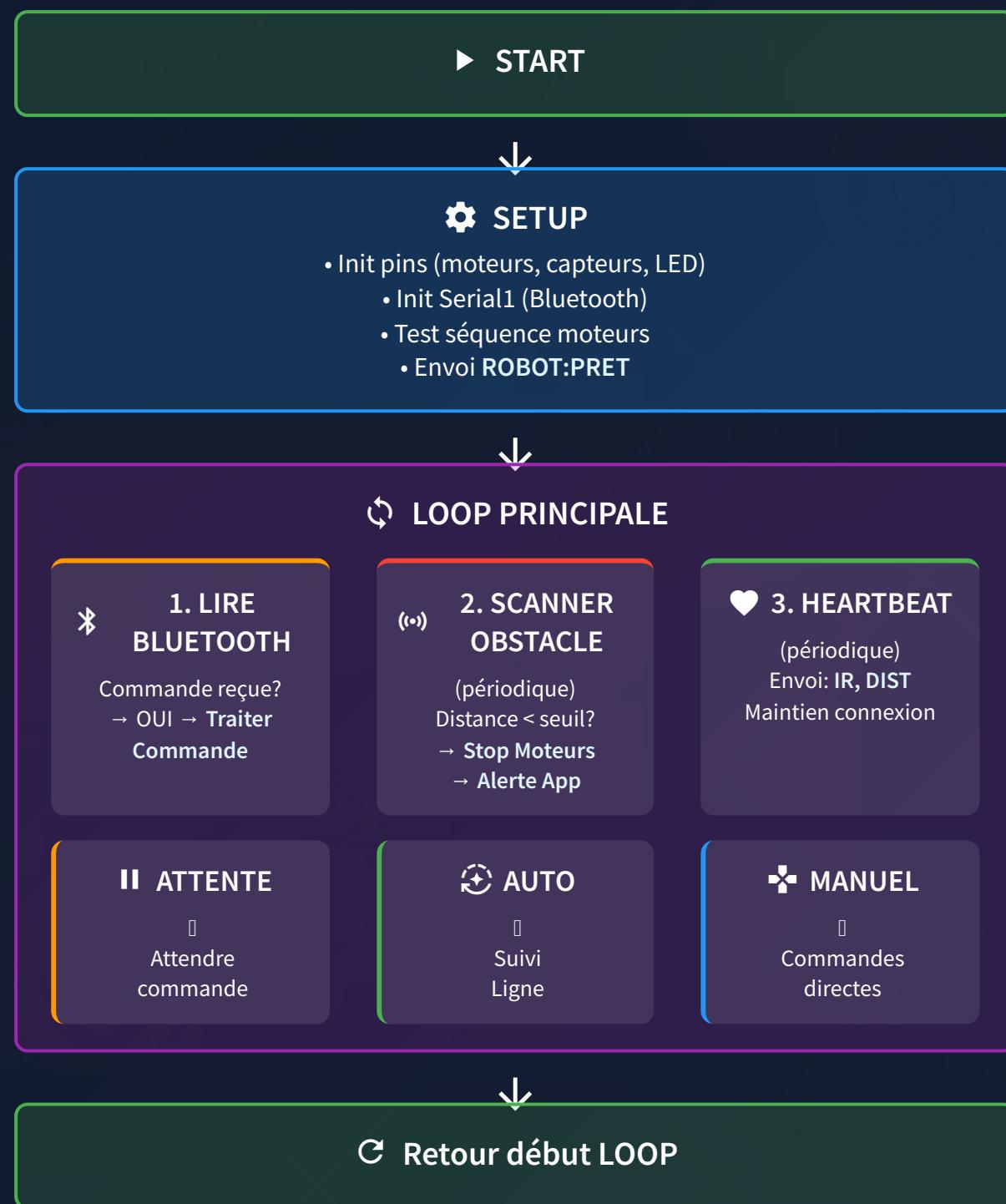
Format: ASCII

Terminaison: \n

Vitesse: Standard

UUID SPP: Standard

# <> Arduino - Logique de Contrôle



## 🛡 Sécurité permanente

Détection obstacles continue, arrêt d'urgence instantané, validation commandes

## ⟳ Communication continue

Heartbeat régulier, status temps réel, synchronisation bidirectionnelle

# 🔧 Fonctions Principales - Code Arduino

## ⚡ Contrôle Moteurs

```
void moteurAvancer(int vitesse) { // PWM ENA/ENB
analogWrite(ENA, vitesse);
analogWrite(ENB, vitesse);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW); }
```

Fonctions: moteurAvancer, moteurReculer, moteurGauche, moteurDroite, moteurPivotGauche, moteurPivotDroite, moteurStop

## ↔ Capteur Ultrason

```
int mesurerDistance() {
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW); long duration = pulseIn(ECHO_PIN, HIGH);
return duration * 0.034 / 2; }
```

Trigger → Mesure Echo → **Calcul distance** → **Retour en cm**

## 👁️ Suivi de Ligne

```
void suivreLigne() { int irL = analogRead(IR_LEFT); int irM = analogRead(IR_MIDDLE); int irR = analogRead(IR_RIGHT); if (irM > SEUIL) { moteurAvancer(vitesse); } else if (irL > SEUIL) { moteurDroite(vitesse); } else if (irR > SEUIL) { moteurGauche(vitesse); } }
```

3 IR → **Seuil ajustable** → Logique de correction → **Recherche si tout blanc**

## ↔ Communication

```
void traiterCommande(char cmd) { switch (cmd) { case 'A': modeActuel = AUTO; break; case 'M': modeActuel = MANUEL; break; case 'F': moteurAvancer(vitesse); break; case 'X': moteurStop(); break; // ... autres commandes }
Serial1.println("ACTION:" + cmd); }
```

Switch-case **commandes** → Change modeActuel → **Envoi confirmations**

## 🛡 Sécurité

```
void verifierObstacle() { int distance = mesurerDistance(); if (distance < SEUIL_OBSTACLE) {
moteurStop();
Serial1.println("OBSTACLE_DETECTE");
obstacleDetecte = true; } else if (obstacleDetecte) {
Serial1.println("VOIE_LIBRE");
obstacleDetecte = false; } }
```

Obstacle détecté → **Stop immédiat** → Envoi alerte → **Blocage jusqu'à voie libre**

## ❤️ Heartbeat

```
void envoyerHeartbeat() { if (millis() - lastHeartbeat > INTERVALLE) { int irL = analogRead(IR_LEFT); int irM = analogRead(IR_MIDDLE); int irR = analogRead(IR_RIGHT);
int dist = mesurerDistance();
Serial1.print("IR:");
Serial1.print(irL);
Serial1.print(",");
Serial1.print(irM);
Serial1.print(irR);
Serial1.println("DIST:");
Serial1.print(dist);
lastHeartbeat = millis(); } }
```

Périodique → Envoi **IR** → Envoi **DIST** → **Maintien connexion**

# 🛡 Systèmes de Sécurité - Multi-Niveaux

## \* Arrêt d'Urgence

- Arrêt immédiat tous moteurs
- ☒ Triple envoi message (garantie)
- ♾ Accessible depuis **tous modes**
- 〽 Passage forcé mode ATTENTE

## (•) Détection Obstacle

- ⌚ Scan permanent
- 旿 Seuil de détection
- ⓧ Arrêt automatique moteurs
- 🔔 Alerte visuelle app
- ⟳ Reprise auto si voie libre

## ❤️ Heartbeat

- Envoi status régulier
- 📡 Détection perte connexion
- ⌚ Maintien synchronisation
- ⌚ Données temps réel (IR, Distance)

## ⚙️ Validation Commandes

- ✓ Vérification mode avant exécution
- ✚ Commandes manuelles **uniquement** en mode MANUEL
- ⓧ Ignore commandes invalides
- ⌚ Log toutes actions

## ⌚ Gestion Erreurs

- ✚ Try-catch sur opérations Bluetooth
- ✗ Fermeture propre sockets
- ❗ Messages erreur explicites
- ⌚ Tentatives reconnexion

## 🛡 Permissions Android

- ✓ Vérification runtime (Android 6+)
- android Support Android 12+ (BLUETOOTH\_CONNECT/SCAN)
- ⌚ Demande automatique si manquantes
- ⌚ ACCESS\_FINE\_LOCATION pour scan

✅ SYSTÈME ROBUSTE ET FIABLE

ID 422744

# ⚠ Stack Technique Complet

## ⌚ HARDWARE

- ⌚ Arduino Mega (ATmega2560)
- \* HC-05 Bluetooth SPP
- ⌚ TCRT5000 (IR)
- ⌚ HC-SR04 (Ultrason)
- ⚡ L298N (Pont H)
- ⚡ Moteurs DC
- ⚡ Batterie 2S Lithium
- ⚡ Châssis robot 2WD

## ↔ SOFTWARE EMBARQUÉ

- ⌚ Arduino IDE
- ⌚ C/C++
- ⌚ Librairies:
  - ↳ Serial1 (UART)
  - ↳ analogWrite() PWM
  - ↳ pulseIn() ultrason
  - ↳ Enum pour modes

## ⌚ APPLICATION ANDROID

- ⌚ Android Studio
- ⌚ Java
- ⌚ SDK Min: Android 6.0
- ⌚ SDK Target: Android 14
- ⌚ Librairies:
  - ↳ AndroidX AppCompat
  - ↳ Material Design 3
  - ↳ MPAndroidChart

## ↔ PROTOCOLES

Bluetooth SPP  
Serial Port Profile

UART  
Communication série

PWM  
Contrôle vitesse

A  
Messages ASCII  
Structurés

# Design Interface - Material Design 3

## Dark Theme

- Background #1E1E1E
- Réduit fatigue oculaire
- Look professionnel
- Économie batterie OLED

## Material Design 3

- CardView avec élévation
- Corners arrondis
- Ripple effects sur boutons
- Ombres douces

## Palette Sémantique



✓ **Vert** : OK, Avancer

❗ **Rouge** : Urgence, Danger

## Hiérarchie Visuelle

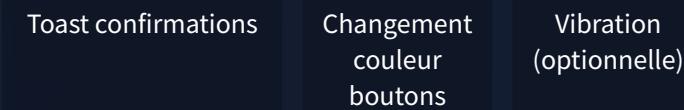
### Titres

#### Sous-titres

Texte

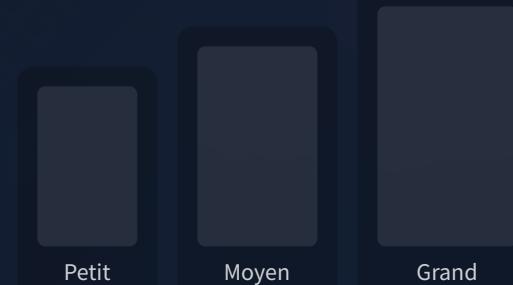
- Emoji (intuitivité)
- Alignement cohérent
- Espacement régulier

## Feedback Utilisateur



- Logs horodatés temps réel
- Alertes visuelles

## Responsive



↔ ScrollView (tous écrans)

☰ LinearLayout weights

Portrait uniquement

# ✓ Tests Complets - Validation

## ⌚ Tests Matériels

- ✓ Connexion **HC-05** (portée)
- ✓ Capteurs **IR** (calibration)
- ✓ **HC-SR04** (précision)
- ✓ Moteurs (PWM)
- ✓ Batterie (autonomie)
- ✓ Stabilité châssis

## 📱 Tests Android

- ✓ Connexion/Déconnexion
- ✓ **Commandes** complètes
- ✓ Réception messages robot
- ✓ Permissions (Android 6-14)
- ✓ Interface responsive
- ✓ Rotation bloquée portrait
- ✓ Pas de fuites mémoire

## <> Tests Arduino

- ✓ 3 modes opérationnels
- ✓ Transitions entre modes
- ✓ Arrêt urgence depuis tous états
- ✓ Détection obstacles
- ✓ Suivi ligne (noir sur blanc)
- ✓ Réponse commandes rapide
- ✓ Heartbeat régulier

## ⟳ Tests Intégration

- ✓ Communication **bidirectionnelle**
- ✓ Synchronisation modes
- ✓ Gestion perte connexion
- ✓ Latence totale faible
- ✓ Séquences multiples commandes
- ✓ Utilisation continue prolongée

➡ SYSTÈME FONCTIONNEL ET STABLE

# ↗ Résultats - Fonctionnalités Opérationnelles



100% FONCTIONNEL (avec matériel actuel)

## ☐ Application Android

- ✓ 4 activités complètes
- ✓ Navigation fluide
- ✓ Interface Material Design
- ✓ Logs temps réel
- ✓ Paramètres persistants

## ✖ Contrôle Robot

- ✓ Mode **Auto** (suivi ligne)
- ✓ Mode **Manuel** (directions)
- ✓ 3 vitesses réglables
- ✓ Stop et arrêt d'urgence
- ✓ Réactivité rapide

## \* Communication

- ✓ Bluetooth **stable**
- ✓ Messages **bidirectionnels**
- ✓ Parsing robuste
- ✓ 0 crash en utilisation

## (⌚) Capteurs

- ✓ Distance obstacle (**HC-SR04**)
- ✓ Suivi ligne (**3x IR**)
- ✓ Graphique distance **données réelles**

## 🎬 Séquenceur

- ✓ Création séquences
- ✓ 6 commandes disponibles
- ✓ Envoi séquentiel fonctionnel
- ✓ Pause/Reprise/Stop

## 🛡 Sécurité

- ✓ Détection obstacle **seuil ajustable**
- ✓ Arrêt automatique
- ✓ Arrêt d'urgence rapide
- ✓ Heartbeat périodique

# ⚠ Limitations Actuelles +💡 Extensions Possibles

## 🔋 Mesure Batterie

- ✖ Actuellement: Message "Capteur non installé"

### + Extension

Diviseur de tension (résistances)

Branchemet: Pin analogique

Code Arduino: `analogRead()` → calcul voltage

## 📡 Capteurs Latéraux

- ✖ Actuellement: 1 seul HC-SR04 (avant)

### + Extension

2× HC-SR04 supplémentaires

Pins: Trig2:6 Echo2:7, Trig3:8 Echo3:9

Detection 360°



## ⌚ Mesure Vitesse Réelle

- ✖ Actuellement: Vitesse théorique (PWM)

### + Extension

2× Encodeurs rotatifs magnétiques

Mesure RPM réel

Calcul distance parcourue



## 🔊 Bips Sonores

- ✖ Actuellement: Pas de buzzer

### + Extension

Buzzer passif

Pin: 3

Séquences avec sons

\$ TOTAL EXTENSIONS: ~130 MAD



Approche Projet: Honnêteté sur limitations • Solutions d'extension documentées • Focus sur fonctionnalités réelles • Qualité > Quantité

ID 422

# ⌚ Défis Techniques Surmontés

## \_ANDROID\_ Permissions Android

- ✗ Problème: Permissions différentes **Android 6-12+**
- ✓ Solution: Vérification SDK dynamique + demande adaptée

## \_BLUETOOTH\_ Stabilité Bluetooth

- ✗ Problème: Déconnexions aléatoires
- ✓ Solution: Heartbeat + reconnexion auto + timeout

## \_IR\_ Calibration IR

- ✗ Problème: Seuil varie selon luminosité
- ✓ Solution: Seuil ajustable + debug temps réel

## \_SYNCHRONISATION\_ Synchronisation Modes

- ✗ Problème: Désync Android ↔ Arduino
- ✓ Solution: Confirmations + status régulier

## \_LATENCE\_ Latence

- ✗ Problème: Délai perceptible
- ✓ Solution: Threads séparés + async + loop optimisée

## \_INTERFACE\_ Interface Chargée

- ✗ Problème: Trop d'infos sur petit écran
- ✓ Solution: **4 activités** + ScrollView + navigation

## 💣 Gestion Erreurs

- ✗ Problème: Crashes Bluetooth
- ✓ Solution: Try-catch systématiques + messages clairs

✅ **RÉSULTAT:** Système stable et robuste

# 🚀 Roadmap - Prochaines Versions

## ↑ VERSION 2.0 ☰ Prochain trimestre

🔋 Capteurs batterie + latéraux

⚡ Encodeurs vitesse réelle

🔊 Buzzer pour séquences

🎙 Contrôle vocal

⬇ Export logs CSV/JSON

📊 Statistiques détaillées

## ✨ VERSION 3.0 ☰ Année prochaine

📷 Caméra + vision (OpenCV)

⚠ Évitement obstacles intelligent  
(A\*)

🤖 Multi-robots simultanés

🧭 Cartographie SLAM basique

ID 422744750 ©

# ♥ REMERCIEMENTS

## Encadrement



Un grand merci à **M. eng RACHIDI MOULAY YOUSSEF** pour son encadrement, ses conseils précieux et son soutien tout au long de ce projet.

## Collaboration



Je remercie chaleureusement mes **camarades de classe** pour leur aide, leurs suggestions précieuses et l'ambiance collaborative qui a rendu ce projet enrichissant.

## Approche Projet



Honnêteté sur limitations • Solutions d'extension documentées • Focus sur fonctionnalités réelles • Qualité > Quantité