



République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Tunis EL Manar
Ecole Nationale d'Ingénieurs de Tunis



DÉPARTEMENT GÉNIE ELECTRIQUE

Système d'aide au stationnement

Réalisé par :

ABDELHEDI EDAM

BEN AYED YASSIN

BELLAJ WAEL

Encadré par :

M. JELASSI KHALED

Année Universitaire : 2024 - 2025

Résumé

Le projet consiste en la conception d'un système d'aide au stationnement basé sur un microcontrôleur, utilisant FreeRTOS pour une gestion multitâche. Ce système détecte les obstacles à l'avant et à l'arrière du véhicule à l'aide de capteurs à ultrasons et présente les résultats sur un tableau de bord.

Grâce à FreeRTOS, les différentes tâches telles que la lecture des capteurs, l'affichage des distances et la gestion des alertes sont exécutées de manière efficace et simultanée. Des tests ont été réalisés pour vérifier la précision des mesures et l'efficacité des alertes.

Ce projet améliore la sécurité et la facilité de stationnement en fournissant une aide en temps réel au conducteur.

Table des matières

Introduction générale	1
1 Présentation du projet	2
1.1 Aperçu du projet	2
1.2 Caractéristiques du système	2
1.3 Composants matériels	2
1.3.1 Microcontrôleur STM32F407	2
1.3.2 Capteur à ultrasons (HC-SR04)	3
1.3.3 Module Bluetooth HC05	4
1.3.4 Pont H L298N	4
1.4 Composants logiciels	5
1.4.1 Keil µVision	5
1.4.2 Visual Studio	5
1.5 Conclusion	6
2 Travail effectué	7
2.1 Introduction	7
2.2 Principales caractéristiques	7
2.2.1 Initialisation GPIO	7
2.2.2 Mesure de distance	7
2.2.3 Contrôle du moteur	8
2.2.4 Communication USART	8
2.2.5 Tâches FreeRTOS	8
2.2.6 Configuration PWM	8
2.2.7 Interruptions	8
2.2.8 Optimisation des tâches	8
2.2.9 Robot	9
2.3 Le Tableau de Bord : Interface Utilisateur	9
2.3.1 Fonctionnalités	9
2.3.2 Communication avec le Microcontrôleur	9
2.3.3 Développement et Avantages	10
2.3.4 Illustration de la Dashboard	10
Conclusion Générale	11

Table des figures

1.1	Cartes STM32F407	3
1.2	ultrasons HC-SR04	3
1.3	Bluetooth HC05	4
1.4	Pont H L298N	4
1.5	Keil µVision Logo	5
1.6	Visual Studio Logo	5
2.7	Robot	9
2.8	Robot	9
2.9	Capture d'écran de la dashboard développée en C#	10

Introduction générale

Dans le cadre de ce projet, nous avons développé un système intelligent de commande et de contrôle robotique reposant sur un microcontrôleur STM32F407 et des technologies embarquées avancées. Ce système intègre plusieurs fonctionnalités clés, telles que la navigation du robot dans toutes les directions (avant, arrière, gauche, droite) et la détection des obstacles grâce à des capteurs à ultrasons.

Le système repose également sur l'utilisation d'un module Bluetooth HC-05 pour assurer une communication sans fil efficace, et d'un pont H L298N pour piloter les moteurs du robot avec précision. L'ensemble est contrôlé via une interface utilisateur intuitive, permettant une visualisation en temps réel des données et une interaction fluide avec le robot.

Ce projet illustre l'utilisation des outils modernes tels que Keil μ Vision et FreeRTOS pour concevoir un système performant, modulaire et extensible, adapté à des applications variées dans le domaine de la robotique et des systèmes embarqués.

Chapitre 1

Présentation du projet

1.1 Aperçu du projet

Ce projet consiste à concevoir un robot capable d'être guidé dans toutes les directions (avant, arrière, gauche, droite). Le robot intègre des capteurs pour mesurer la distance des obstacles situés devant et derrière, garantissant une navigation sûre et efficace.

Le système se compose des éléments clés suivants :

- **Interface utilisateur** : Une plateforme pour la visualisation des données et le contrôle du robot.
- **Microcontrôleur STM32** : Responsable de la gestion des capteurs et des actionneurs pour assurer les différentes fonctions du robot.
- **Capteurs** : Inclut des capteurs de mouvement et d'énergie pour l'acquisition des données essentielles.

1.2 Caractéristiques du système

Les principales caractéristiques du système de Robot comprennent :

- **Contrôle du Robot** : Un contrôle total de guidage d'un robot en temps réel.
- **Guidage du Robot** : Guider le robot en toute position .
- **Contrôle basé sur l'interface graphique** : une interface utilisateur graphique (GUI) développée avec Visual Studio pour le contrôle et le guidage du système via bluetooth pour une interaction en temps réel.

1.3 Composants matériels

1.3.1 Microcontrôleur STM32F407

Le STM32F407 est un microcontrôleur 32 bits performant, basé sur l'architecture ARM Cortex-M4. Il se distingue par sa polyvalence, ses nombreuses fonctionnalités périphériques et sa capacité à gérer efficacement les tâches en temps réel. Ce microcontrôleur est largement employé dans des projets embarqués exigeant un traitement rapide, tels que

les systèmes robotiques.

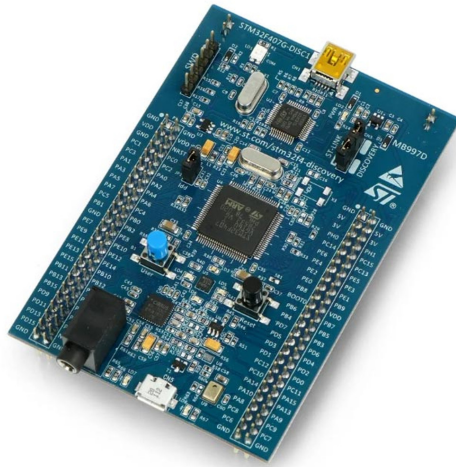


FIGURE 1.1 – Cartes STM32F407

1.3.2 Capteur à ultrasons (HC-SR04)

Le capteur à ultrasons HC-SR04 permet de mesurer les distances en émettant des ondes ultrasonores et en calculant le temps que mettent ces ondes à revenir après avoir été réfléchies par un obstacle. Dans ce projet, il est utilisé pour détecter des objets ou mesurer avec précision les distances dans l'environnement.

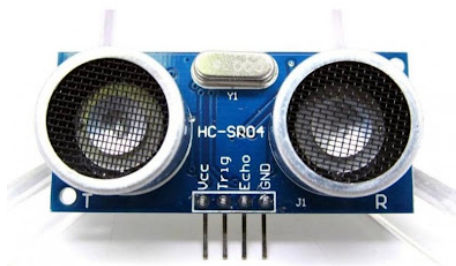


FIGURE 1.2 – ultrasons HC-SR04

1.3.3 Module Bluetooth HC05

Le module Bluetooth HC-05 est un module de communication sans fil conçu pour établir des connexions Bluetooth série entre des appareils. Il est largement utilisé dans les projets embarqués pour transmettre des données entre un microcontrôleur et un appareil mobile ou un ordinateur. Dans ce projet, le HC-05 permet de contrôler le robot à distance et d'échanger des données en temps réel via une interface utilisateur.

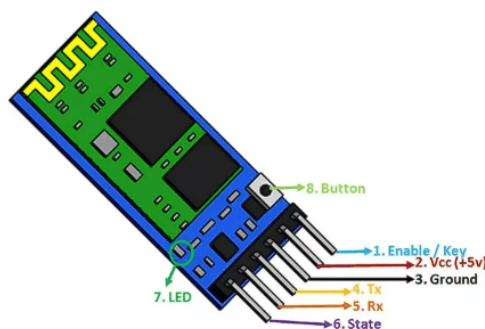


FIGURE 1.3 – Bluetooth HC05

1.3.4 Pont H L298N

Le pont H L298N est un module de commande de moteur courant continu ou pas à pas, conçu pour contrôler simultanément deux moteurs en leur permettant de tourner dans les deux sens. Grâce à sa capacité à gérer des courants élevés et à intégrer des protections thermiques, il est idéal pour les projets robotiques nécessitant un contrôle précis de la vitesse et de la direction des moteurs. Dans ce projet, le L298N est utilisé pour piloter les moteurs du robot, assurant une mobilité fluide et fiable.

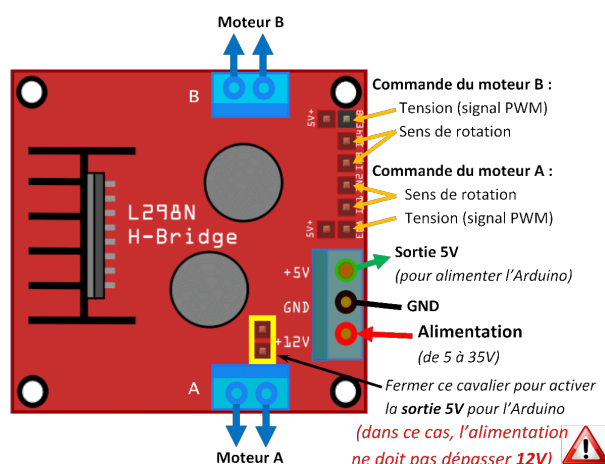


FIGURE 1.4 – Pont H L298N

1.4 Composants logiciels

1.4.1 Keil μ Vision

Keil μ Vision est un environnement de développement intégré (IDE) puissant, utilisé pour le développement et le débogage de logiciels embarqués sur des microcontrôleurs. Il prend en charge de nombreuses architectures, dont ARM Cortex-M, et offre des outils intégrés tels qu'un éditeur de code, un compilateur, et un débogueur. Grâce à son interface conviviale et à ses fonctionnalités avancées, μ Vision simplifie le processus de développement, permettant aux développeurs de concevoir, tester et optimiser efficacement leurs applications embarquées.



FIGURE 1.5 – Keil μ Vision Logo

1.4.2 Visual Studio

Visual Studio est un environnement de développement intégré (IDE) complet développé par Microsoft, conçu pour prendre en charge un large éventail de langages de programmation tels que C++, C, Python, JavaScript, et bien d'autres. Il est largement utilisé pour le développement d'applications sur différentes plateformes, notamment Windows, web, mobile, et cloud.

Grâce à ses outils avancés comme l'éditeur de code intelligent, le débogueur intégré, la gestion de projet, et l'intégration avec des services tels que Git, Visual Studio permet aux développeurs de concevoir, tester, et déployer efficacement des applications de haute qualité. Il est particulièrement apprécié pour son extensibilité, permettant d'ajouter des fonctionnalités personnalisées via des extensions.



FIGURE 1.6 – Visual Studio Logo

1.5 Conclusion

En résumé, ce chapitre a permis de poser les bases conceptuelles et techniques du projet en détaillant son aperçu général, les caractéristiques principales du système, ainsi que les composants matériels et logiciels impliqués. Ces éléments fournissent une vision claire et structurée des objectifs du projet, tout en mettant en lumière les technologies et outils utilisés pour sa mise en œuvre.

Cette présentation constitue un socle essentiel pour aborder les phases suivantes du développement, où chaque composant, qu'il soit matériel ou logiciel, jouera un rôle clé dans la réalisation des fonctionnalités attendues et dans l'atteinte des performances visées.

Chapitre 2

Développement et implémentation

2.1 Introduction

Ce chapitre détaille les principales fonctionnalités du système en mettant en lumière les aspects essentiels qui assurent son fonctionnement optimal. À travers une organisation méthodique, nous décrivons les mécanismes de configuration, de contrôle et de communication intégrés dans le projet.

Nous commencerons par l'initialisation des GPIO, qui configure les broches nécessaires aux capteurs et au contrôle des moteurs, suivie par les processus de mesure de distance utilisant des capteurs à ultrasons. Le contrôle des moteurs sera abordé, avec une explication des techniques PWM employées pour ajuster la vitesse et la direction.

Les sections suivantes traiteront de la communication via USART, permettant un contrôle externe du système, et des tâches FreeRTOS qui assurent une gestion parallèle des capteurs et des moteurs. Enfin, des optimisations et des configurations avancées, comme l'utilisation d'interruptions et la gestion efficace des tâches, seront explorées pour maximiser la performance globale du système.

2.2 Principales caractéristiques

2.2.1 Initialisation GPIO

Configure les broches GPIO pour les capteurs (broches de déclenchement et d'écho pour les capteurs avant et arrière) et le contrôle du moteur.

Configure les broches PWM pour le contrôle de la vitesse du moteur et les broches de contrôle de la direction pour le mouvement du moteur.

2.2.2 Mesure de distance

La fonction " MeasureDistance() " calcule la distance à l'aide de capteurs à ultrasons. La fonction déclenche le capteur et mesure le temps jusqu'au retour de l'écho. Le décalage horaire est utilisé pour calculer la distance.

2.2.3 Contrôle du moteur

Les moteurs sont contrôlés à l'aide de PWM (Pulse width Modulation) sur deux canaux, ajustant la vitesse et la direction des moteurs pour le mouvement (avant, arrière, gauche, droite, arrêt).

La fonction " ControlMotors() " gère différentes commandes (F, B, R, L, S) qui représentent respectivement l'avant, l'arrière, le virage à droite, le virage à gauche et l'arrêt.

2.2.4 Communication USART

UART (USART2) est initialisé pour la communication. Les données reçues sur l'UART sont utilisées pour contrôler les moteurs.

" USART2_IRQHandler() " gère l'interruption et traite les données reçues, en appelant " ControlMotors() " avec le caractère reçu.

2.2.5 Tâches FreeRTOS

Deux tâches (FrontSensorTask et BackSensorTask) s'exécutent en parallèle à l'aide de FreeRTOS. Ces tâches mesurent les distances à l'aide des capteurs à ultrasons et envoient le résultat via UART. Les délais de tâche (vTaskDelay) permettent aux capteurs de mesurer périodiquement.

2.2.6 Configuration PWM

Le PWM pour les moteurs est configuré à l'aide du Timer 4, avec une fréquence de 1 kHz et un prescaler pour une horloge temporisée de 1 MHz.

2.2.7 Interruptions

L'interruption USART2 est activée pour gérer les données UART entrantes de manière asynchrone et contrôler les moteurs en conséquence.

2.2.8 Optimisation des tâches

Les tâches exécutées sous FreeRTOS pourraient être encore optimisées. Par exemple, plutôt que d'utiliser " vTaskDelay " pour les lectures périodiques des capteurs, vous pouvez implémenter une approche basée sur une minuterie ou des événements pour minimiser l'utilisation du processeur.

2.2.9 Robot

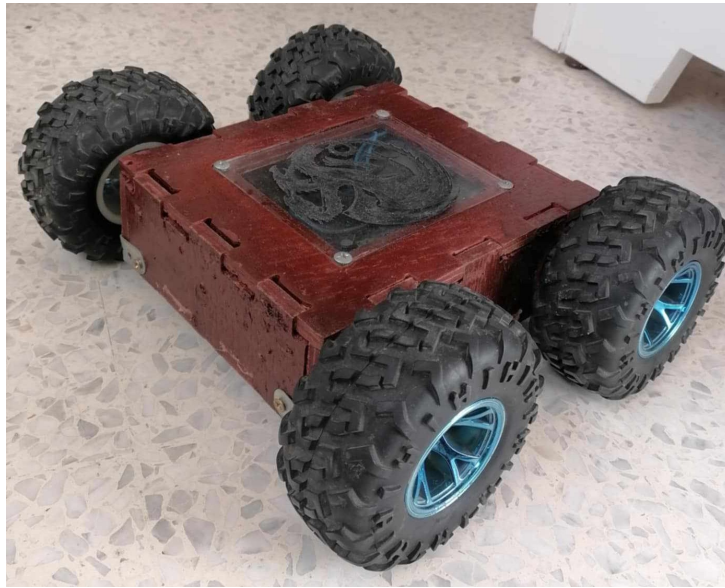


FIGURE 2.7 – Robot



FIGURE 2.8 – Robot

2.3 Le Tableau de Bord : Interface Utilisateur

Le tableau de bord, développé en **C#** à l'aide de Visual Studio, constitue l'interface utilisateur principale du système d'aide au stationnement. Son objectif est de fournir une représentation visuelle et sonore des données recueillies par les capteurs à ultrasons et transmises via le microcontrôleur.

2.3.1 Fonctionnalités

Le tableau de bord est conçu pour afficher :

- **Les distances mesurées** : Présentées sous forme numérique.
- **État de connexion Bluetooth** : Pour informer l'utilisateur de l'état de la communication entre le tableau de bord et le microcontrôleur.

2.3.2 Communication avec le Microcontrôleur

La communication entre le tableau de bord et le microcontrôleur s'effectue via Bluetooth. Cette liaison sans fil permet de :

1. **Recevoir les données des capteurs** en temps réel.
2. **Envoyer des commandes spécifiques** (par exemple : avancer, reculer etc...).

La connexion Bluetooth est gérée en **C#** en utilisant des bibliothèques standards pour établir et maintenir la communication série avec le module Bluetooth connecté au micro-contrôleur.

2.3.3 Développement et Avantages

Le développement avec **C#** offre une interface graphique moderne et facilement personnalisable. L'utilisation de Visual Studio a permis d'intégrer des fonctionnalités avancées, comme des animations ou des graphiques interactifs.

Ce tableau de bord améliore considérablement l'expérience utilisateur en fournissant des informations claires et en temps réel, tout en permettant une interaction sans fil pratique et efficace avec le système embarqué.

2.3.4 Illustration de la Dashboard

La figure ci-dessous montre une capture d'écran du tableau de bord développé :

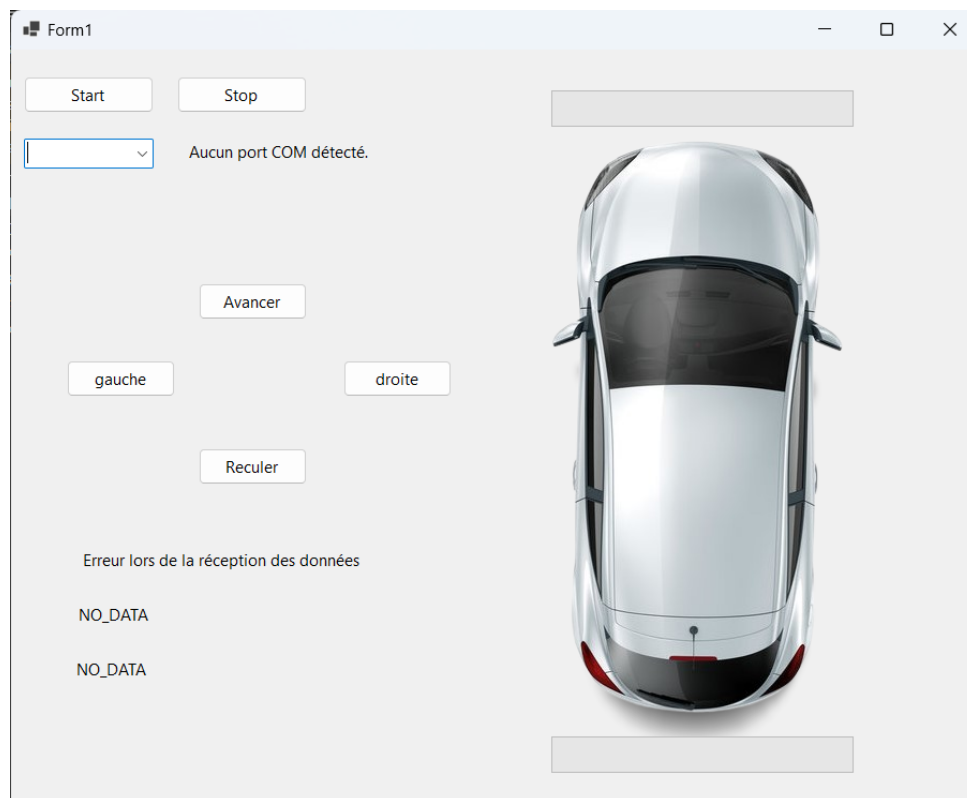


FIGURE 2.9 – Capture d'écran de la dashboard développée en **C#**.

Conclusion Générale

En conclusion, le développement de ce système d'aide au stationnement basé sur un microcontrôleur et FreeRTOS a permis de concevoir une solution efficace pour assister les conducteurs lors des manœuvres de stationnement. Grâce à l'intégration de capteurs à ultrasons, le système détecte les obstacles avec précision, tandis que l'utilisation de FreeRTOS garantit une gestion multitâche optimale, permettant une interaction fluide entre les capteurs, l'affichage et les alertes. Les tests effectués ont démontré la fiabilité et la réactivité du système en conditions simulées. Ce projet représente une base solide pour des améliorations futures, telles que l'ajout de fonctionnalités avancées comme la détection de piétons ou l'intégration avec des systèmes de conduite autonome, contribuant ainsi à renforcer la sécurité et le confort des usagers.