Ministère de l'Enseignement supérieur et de la Recherche scientifique

****

Faculté des sciences de Tunis

Système de Porte de Garage Intelligente avec STM32

Projet de Fin d'Année

élaboré par :

Djelassi Youssef

Bahrouni Ayoub

AU 2023-2024

Cahier De Charges

1.Objectifs :

Concevoir et développer une porte de garage intelligente permettant :

- Le contrôle à distance de l'ouverture et de la fermeture via Bluetooth
- La détection automatique de l'état ouvert/fermé de la porte
- L'envoi de notifications en cas de porte laissée ouverte
- La programmation d'horaires d'ouverture/fermeture automatiques

2.Fonctionnalités requises :

1. Contrôle à distance:

- Connexion Bluetooth pour communiquer avec un smartphone ou une tablette
- Interface utilisateur mobile pour commander l'ouverture et la fermeture de la porte
- Réponse en temps réel aux commandes

2. Détection d'état

- Capteurs de position pour détecter si la porte est complètement ouverte ou fermée
- Lecture fiable de l'état de la porte par le microcontrôleur

3. Notifications

- Envoi de notifications push sur le smartphone en cas de porte laissée ouverte
- Délai configurable avant l'envoi de la notification

4. **Programmation des horaires**

- Possibilité de programmer des horaires d'ouverture et de fermeture automatiques
- Gestion des jours de la semaine et des exceptions

5. **Sécurité**

- Détection d'obstacles pendant la fermeture pour éviter les pincements
- Arrêt immédiat du moteur en cas de détection d'obstacle

6. **Fiabilité**

- Fonctionnement stable et sans erreur du système
- Redémarrage automatique en cas de plantage

7. **Facilité d'installation**

- Intégration simple dans une porte de garage existante
- Câblage réduit grâce à la communication sans fil

Table des Matières

Introduction Générale

Dans un monde de plus en plus connecté et automatisé, la domotique occupe une place centrale dans l'amélioration de notre confort et de notre sécurité au quotidien. Les systèmes de contrôle intelligents deviennent indispensables pour gérer efficacement nos espaces de vie, en particulier en ce qui concerne l'accès aux bâtiments et aux propriétés. Dans ce contexte, la porte de garage intelligente se présente comme une solution innovante et pratique, apportant une couche supplémentaire de commodité et de sécurité.
Ce projet vise à concevoir et mettre en œuvre une porte de garage intelligente, basée sur un microcontrôleur STM32, qui combine l'automatisation, le contrôle à distance, et des fonctionnalités de sécurité avancées. Le microcontrôleur STM32 a été choisi pour sa robustesse, sa flexibilité et ses capacités avancées de traitement, permettant de gérer efficacement les diverses fonctionnalités nécessaires pour une porte de garage intelligente.

Les objectifs principaux de ce projet incluent l'automatisation de l'ouverture et de la fermeture de la porte, la possibilité de contrôle à distance via une application mobile, la détection d'obstacles pour prévenir les accidents, ainsi que des mesures de sécurité renforcées pour protéger contre les accès non autorisés. En intégrant des technologies modernes telles que des capteurs à ultrasons et des modules Bluetooth , ce système vise à offrir une solution complète et fiable pour la gestion de portes de garage.

Ch1. Contexte Général du Projet

1.1. Introduction:

Avec l'avancée rapide des technologies dans le domaine de la domotique et de l'Internet des objets (IoT), les systèmes de gestion automatisée des installations domestiques sont devenus plus accessibles et plus sophistiqués. Parmi ces innovations, les portes de garage intelligentes se sont imposées comme des dispositifs essentiels pour améliorer la sécurité, la commodité et l'efficacité des maisons modernes. Ce projet vise à développer un système de porte de garage intelligente utilisant un microcontrôleur STM32, permettant à la fois un contrôle manuel et à distance, ainsi qu'une gestion automatisée de l'ouverture et de la fermeture.

1.2. Problématique

Les portes de garage conventionnelles peuvent présenter plusieurs inconvénients, notamment :

Manque de sécurité : Les systèmes manuels ou semi-automatiques peuvent être vulnérables aux intrusions et ne détectent pas toujours les obstacles de manière fiable.

Absence de contrôle à distance : De nombreuses portes de garage ne permettent pas un contrôle à distance, ce qui peut être peu pratique pour les utilisateurs.

Maintenance et coût : Les systèmes mécaniques complexes nécessitent une maintenance régulière, entraînant des coûts supplémentaires.

1.3. Objectifs

L'objectif principal de ce projet est de concevoir et de développer un système de porte de garage intelligente qui surmonte ces limitations en intégrant des fonctionnalités avancées de détection et de contrôle. Les objectifs spécifiques incluent :

Automatisation : Développer un système capable de gérer automatiquement l'ouverture et la fermeture de la porte de garage en fonction de la position de la porte et des obstacles détectés.
Sécurité : Intégrer des capteurs fiables pour la détection des obstacles et la position de la porte afin d'assurer un fonctionnement sûr.
Contrôle à distance : Permettre aux utilisateurs de contrôler la porte de garage via une application mobile, offrant ainsi une plus grande commodité et flexibilité.
Efficacité énergétique : Concevoir un système qui optimise la consommation d'énergie tout en maintenant des performances élevées.

1.4. Technologies Utilisées :

Le système proposé s'appuie sur plusieurs technologies et composants, notamment :
Microcontrôleur STM32 : Un microcontrôleur puissant et polyvalent pour gérer les entrées et sorties du système.
Capteurs à effet Hall : Utilisés pour détecter la position de la porte.
Capteurs ultrasoniques : Pour la détection des obstacles.
Module Bluetooth : Pour le contrôle à distance via une application mobile.
Moteur électrique : Pour l'ouverture et la fermeture de la porte.

1.5. Valeur Ajoutée

Le développement de ce système de porte de garage intelligente apporte plusieurs avantages :
Sécurité renforcée : Grâce à une détection fiable des obstacles et à la possibilité de surveiller et contrôler la porte à distance.
Commodité accrue : Les utilisateurs peuvent ouvrir ou fermer la porte de leur garage sans avoir à sortir de leur voiture ou à utiliser un interrupteur manuel.

Économie de temps et d'efforts : L'automatisation du système réduit la nécessité d'intervention humaine, permettant un fonctionnement plus fluide et efficace.

Ch2. Description du Système

2.1.Matériel Utilisé :

1. Microcontrôleur STM32F4 :

STM32F4 est une famille de microcontrôleurs de STMicroelectronics basée sur le cœur ARM Cortex-M4. Ces microcontrôleurs sont réputés pour leurs hautes performances, leurs périphériques avancés et leur large gamme d'applications.

2. Moteur DC :

Moteur à courant continu adapté à la taille et au poids de la porte de garage, avec un contrôleur de moteur si nécessaire.

3. Capteurs de Position :

Interrupteurs de fin de course, encodeurs rotatifs ou capteurs à effet Hall pour détecter l'état de la porte (ouverte ou fermée) et surveiller son mouvement.

4. Capteurs de Ultrasonique :

Capteur ultrasonique pour la détection d'obstacles (capteur HC-SR04)

5. Module de Connectivité sans Fil :

Module Bluetooth pour permettre la communication sans fil entre le système de porte de garage et l'application mobile.(module Bluetooth HC-05)

6. Boîtier de Protection :

Boîtier robuste et étanche pour protéger les composants électroniques des éléments environnementaux et des dommages physiques.

2.2. Logiciel Utilisé:

1. **Environnement de développement:**

   IDE pour STM32 : STM32Cube IDE Environnement de développement intégré (IDE) fourni par STMicroelectronics pour programmer les microcontrôleurs STM32.

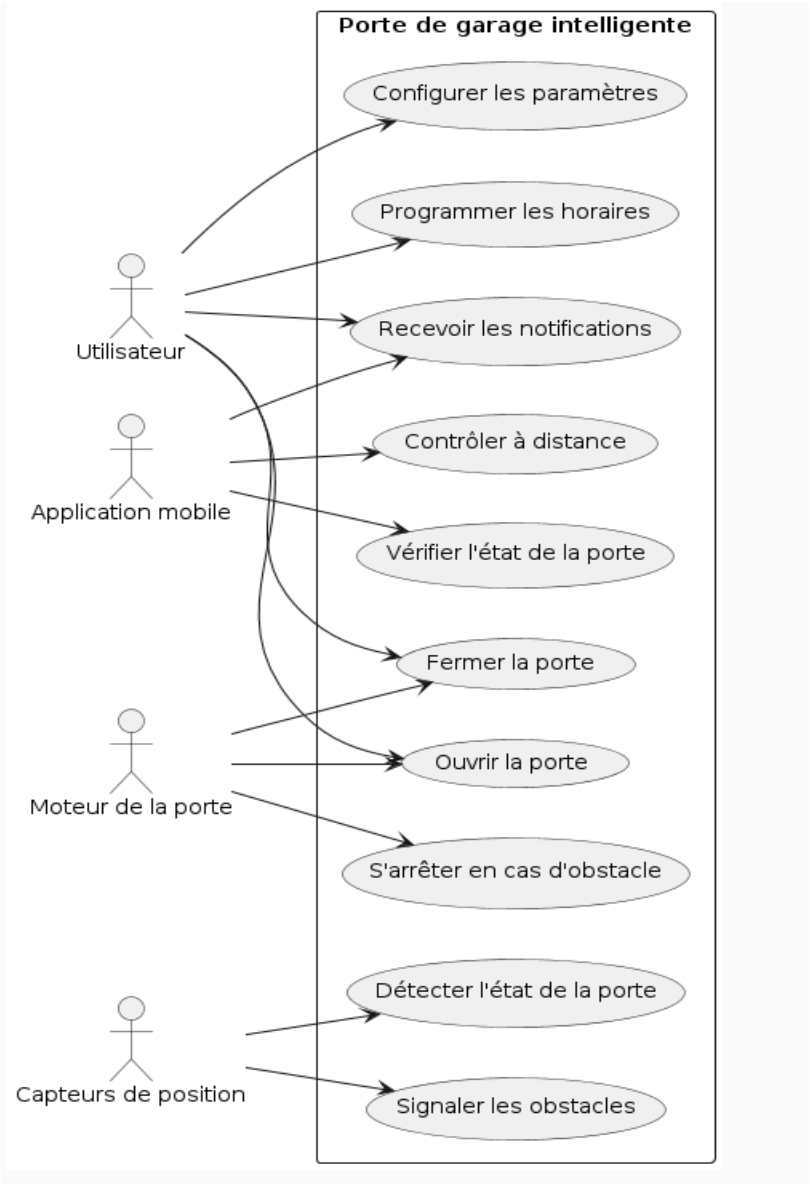
   Outils de Programmation d'Interfaces Utilisateur Mobile : Android Studio (pour Android), Xcode (pour iOS) : Pour développer des applications mobiles permettant le contrôle à distance de la porte de garage.

2. **Outils de simulation:**

   STM32CubeMX : Outil de configuration graphique pour les microcontrôleurs STM32, permettant de générer du code d'initialisation et de configurer les périphériques.

Ch3. Conception et Implémentation

3.1. Diagramme de cas d'utilisation :



Ce diagramme montre les cas d'utilisation suivants :

L'utilisateur peut ouvrir et fermer la porte manuellement

L'utilisateur peut programmer des horaires d'ouverture et de fermeture automatiques.

L'utilisateur peut recevoir des notifications en cas de porte laissée ouverte.

L'utilisateur peut configurer les paramètres du système.

L'application mobile peut contrôler à distance l'ouverture et la fermeture de la porte.

L'application mobile peut recevoir des notifications en cas de porte laissée ouverte.

L'application mobile peut vérifier l'état de la porte à distance.

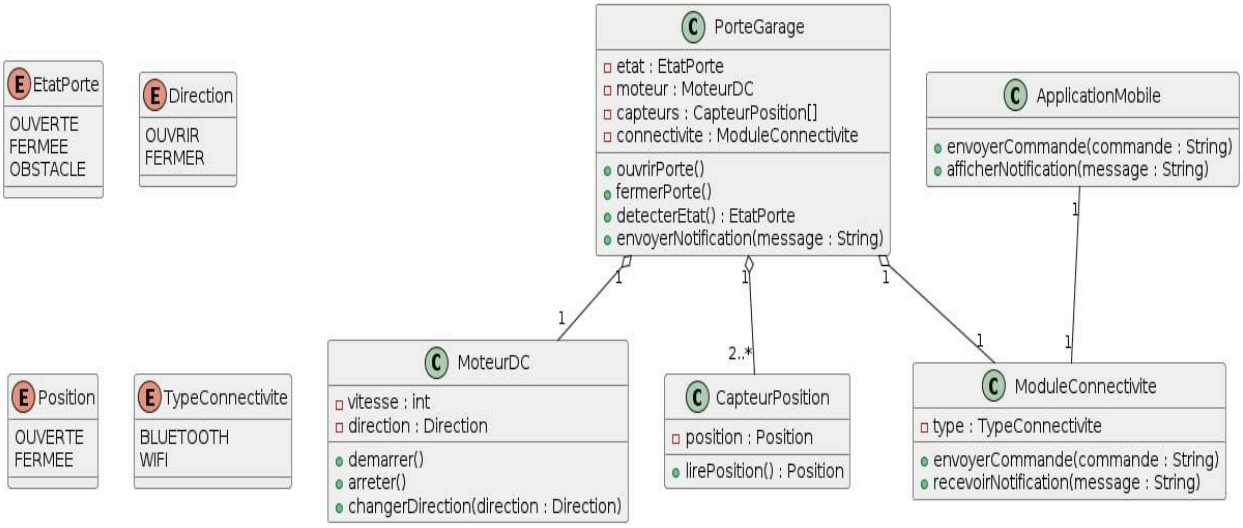
Les capteurs de position détectent l'état ouvert ou fermé de la porte et transmettent cette information au système.

Les capteurs de position signalent les obstacles pour éviter les pincements.

Le moteur de la porte ouvre et ferme la porte en réponse aux commandes du système.

Le moteur de la porte s'arrête en cas d'obstacle détecté.

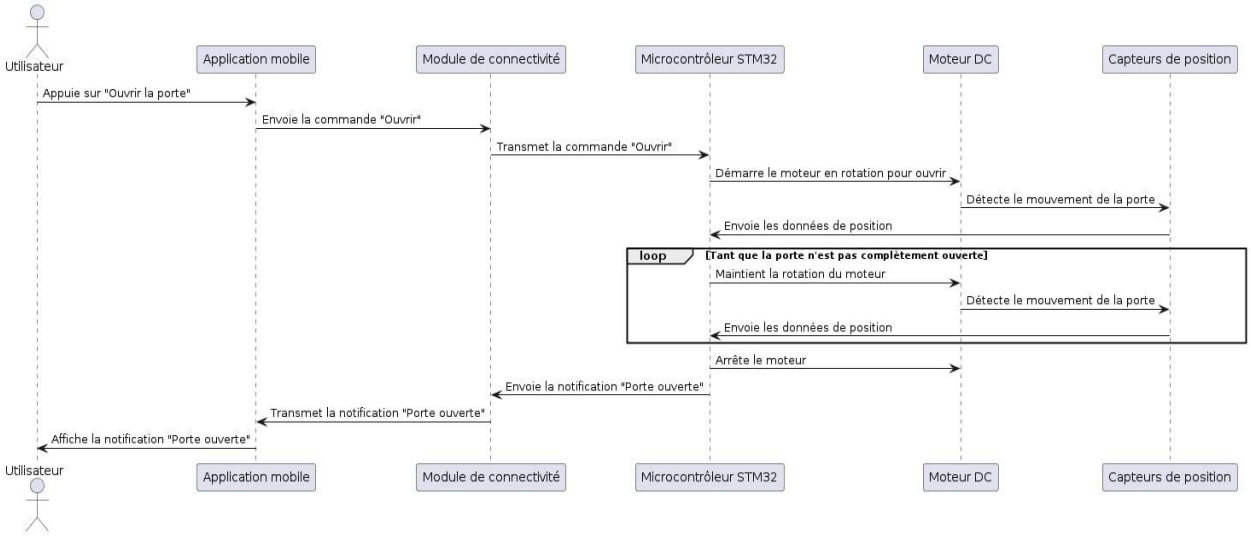
3.2. Diagramme de classe :



Ce diagramme de classes montre les principales classes du système :

- La classe **PorteGarage** représente la porte de garage intelligente. Elle contient des références au moteur, aux capteurs, au module de connectivité et gère l'état de la porte.
- La classe **MoteurDC** contrôle le moteur pour ouvrir et fermer la porte.
- La classe **CapteurPosition** lit la position de la porte (ouverte ou fermée).
- La classe **ModuleConnectivite** gère la communication avec l'application mobile.
- La classe **ApplicationMobile** permet d'envoyer des commandes et d'afficher les notifications.
- Les classes **EtatPorte**, Direction, Position et **TypeConnectivite** sont des énumérations qui définissent les différentes valeurs possibles pour ces concepts.
- Les associations montrent les relations entre les classes. Par exemple, une **PorteGarage** contient une référence à un **MoteurDC** et à un ou plusieurs **CapteurPosition**.

3.3. Diagramme de séquence:



Ce diagramme de séquence illustre les interactions entre les différents composants du système lors de l'ouverture de la porte de garage à distance via l'application mobile :

- L'utilisateur appuie sur le bouton "Ouvrir la porte" dans l'application mobile.
- L'application mobile envoie la commande "Ouvrir" au module de connectivité.
- Le module de connectivité transmet la commande "Ouvrir" au microcontrôleur STM32.
- Le STM32 démarre le moteur DC pour ouvrir la porte.
- Le moteur DC détecte le mouvement de la porte et envoie les données de position aux capteurs.
- Les capteurs de position envoient les données de position au STM32.
- Tant que la porte n'est pas complètement ouverte, le STM32 maintient la rotation du moteur.
- Une fois la porte complètement ouverte, le STM32 arrête le moteur.
- Le STM32 envoie une notification "Porte ouverte" au module de connectivité.
- Le module de connectivité transmet la notification à l'application mobile.
- L'application mobile affiche la notification "Porte ouverte" à l'utilisateur.
- Ce diagramme de séquence montre les échanges de messages entre les différents composants du système lors de l'ouverture de la porte de garage à distance. Il peut être utilisé pour comprendre le fonctionnement détaillé du système et pour guider le développement du code.

3.4. Défis techniques et solutions envisagées :

- **Conception Mécanique** : Concevoir un système de motorisation fiable et efficace pour l'ouverture et la fermeture de la porte, tout en optimisant l'encombrement et la facilité d'installation.
- **Électronique embarquée :** Développer des circuits électroniques robustes et performants pour gérer les capteurs, les actionneurs et les communications sans fil en temps réel.
- **Programmation Avancée :** Écrire un logiciel intelligent capable de coordonner les différentes fonctionnalités de la porte de garage, tout en assurant une interface conviviale pour l'utilisateur.

Ch4. Programmation

4.1. Fonctions de Configuration :

Initialisation GPIO :

```
void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  __HAL_RCC_GPIOC_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();

  GPIO_InitStruct.Pin = GPIO_PIN_13;
  GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

  GPIO_InitStruct.Pin = GPIO_PIN_8;
  GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}
```

Cette fonction initialise les broches GPIO utilisées pour détecter les états de la porte (ouverte ou fermée). PC13 est utilisé pour détecter si la porte est ouverte, et PA8 pour détecter si la porte est fermée.

Initialisation du Timer (TIM2) :

```
void MX_TIM2_Init(void)
{
  TIM_OC_InitTypeDef sConfigOC = {0};

  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 0;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 65535;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }

  sConfigOC.OCMode = TIM_OCMODE_PWM1;
  sConfigOC.Pulse = 0;
  sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
  sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
  if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_2) != HAL_OK)
  {
    Error_Handler();
  }
}
```

Cette fonction configure le timer 2 pour générer des signaux PWM nécessaires pour contrôler les actionneurs responsables de l'ouverture et de la fermeture de la porte.

Initialisation de l'UART (USART1) :

```
void MX_USART1_UART_Init(void)
{
  huart1.Instance = USART1;
  huart1.Init.BaudRate = 9600;
  huart1.Init.WordLength = UART_WORDLENGTH_8B;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.Mode = UART_MODE_TX_RX;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart1.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
    Error_Handler();
  }
}
```

Cette fonction configure l'interface UART pour permettre la communication série à un débit de 9600 bauds.

Configuration de l'Horloge Système :

```
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }

  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK
                              | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
  {
    Error_Handler();
  }
}
```

Cette fonction configure les oscillateurs et les horloges pour le microcontrôleur.

4.2. Fonctions Bluetooth :

Vérification de la Disponibilité des Données :

```
bool Bluetooth_DataAvailable()
{
  return __HAL_UART_GET_FLAG(&huart1, UART_FLAG_RXNE);
}
```

Cette fonction vérifie si des données sont disponibles pour être lues depuis l'UART.

Lecture des Données :

```
char Bluetooth_ReadData()
{
  char data;
  HAL_UART_Receive(&huart1, (uint8_t*)&data, 1, HAL_MAX_DELAY);
  return data;
}
```

Cette fonction lit un octet de données depuis l'UART.

Envoi de Notification

```
void Bluetooth_SendNotification(const char* message)
{
  HAL_UART_Transmit(&huart1, (uint8_t*)message, strlen(message), HAL_MAX_DELAY);
}
```

Cette fonction envoie un message via Bluetooth.

4.3. Fonctions de Contrôle de la Porte :

Ouverture de la Porte :

```
void Open_Door()
{
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
  while (!Door_IsClosed())
    HAL_Delay(100);
  HAL_TIM_PWM_Stop(&htim2, TIM_CHANNEL_1);
}
```

Cette fonction démarre le PWM sur le canal 1 pour ouvrir la porte et attend que la porte soit fermée avant d'arrêter le PWM.

Fermeture de la Porte :

```
void Close_Door()
{
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);
  while (!Door_IsOpen())
    HAL_Delay(100);
  HAL_TIM_PWM_Stop(&htim2, TIM_CHANNEL_2);
}
```

Cette fonction démarre le PWM sur le canal 2 pour fermer la porte et attend que la porte soit ouverte avant d'arrêter le PWM.

4.4. Fonctions de Détection de l'État de la Porte :

Porte Ouverte :

```
bool Door_IsOpen()
{
  return HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_SET;
}
```

Cette fonction retourne l'état de la broche GPIO indiquant si la porte est ouverte.

Porte Fermée :

```
bool Door_IsClosed()
{
  return HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == GPIO_PIN_SET;
}
```

Cette fonction retourne l'état de la broche GPIO indiquant si la porte est fermée.

4.5. Fonction Principale :

```c
int main(void)
{
  HAL_Init();
  SystemClock_Config();

  MX_GPIO_Init();
  MX_TIM2_Init();
  MX_USART1_UART_Init();

  while (1)
  {
    if (Bluetooth_DataAvailable())
    {
      char command = Bluetooth_ReadData();
      if (command == 'O')
        Open_Door();
      else if (command == 'C')
        Close_Door();
    }

    if (Door_IsOpen() && HAL_GetTick() % 10000 == 0)
      Bluetooth_SendNotification("Door is open!");

    HAL_Delay(100);
  }
}
```

Cette fonction initialise le système, configure les modules et entre dans une boucle infinie où elle vérifie les commandes reçues via Bluetooth pour ouvrir ou fermer la porte. Elle envoie également une notification toutes les 10 secondes si la porte est ouverte.

Conclusion

La conception et la mise en œuvre d'une porte de garage intelligente basée sur un microcontrôleur STM32 présentent une solution moderne et sécurisée pour l'automatisation des portes de garage. Ce cahier des charges détaille les spécifications techniques, les fonctionnalités, les contraintes, et les étapes nécessaires à la réalisation de ce projet.

En intégrant des capteurs de position et d'obstacles, ainsi qu'une communication à distance via Bluetooth, le système garantit une utilisation sécurisée et pratique. Les fonctionnalités avancées telles que les notifications en temps réel et les mécanismes de sécurité renforcent la fiabilité et l'efficacité du système.

Le respect des normes environnementales et réglementaires assure une conformité aux standards de sécurité et de performance. Le développement structuré et les phases de validation permettront de créer un prototype fonctionnel et de haute qualité, adapté aux besoins des utilisateurs finaux.

En conclusion, ce projet vise à fournir une solution robuste et innovante pour la gestion automatisée des portes de garage, améliorant ainsi la sécurité et le confort des utilisateurs tout en offrant des fonctionnalités de contrôle avancées.