



REPUBLIQUE TUNISIENNE
MINISTÈRE DE L'ENSEIGNEMENT
SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITE TUNIS EL MANAR

FACULTE DES SCIENCES DE TUNIS
DEPARTEMENT DES SCIENCES DE
L'INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention de la
LICENCE EN COMPUTER ENGINEERING
Spécialité : Systèmes Embarqués et IoT

Par Rayen SALHI

Réalisation de communication Bluetooth LE entre un IHM et une application Android

Encadrant Académique :

Mr. Chiheb Ameur Abid

Encadrant Professionnel :

Mr. Mohamed Mehdi Ayadi

Réalisé au sein de Innov-Alliance-Tech



Année Universitaire : 2022 - 2023

FICHE DE SYNTHÈSE

Dans le cadre de la gestion informatisée des stages de PFE et de l'archivage des rapports de PFE, nous vous demandons de renseigner les items suivants :

- Formation : LCE3
- Année Universitaire : 2022/2023
- Session : Principale
- Auteur(s) (Nom, prénom) : Rayen Salhi
- Titre du rapport : Réalisation de communication Bluetooth LE entre un IHM et une application Android
- Organisme d'accueil : Innov-Alliance-Tech
- Pays d'accueil : La Tunisie
- Responsable de stage (nom et prénom) : Mr. Mohamed Mehdi Ayadi
- Email du Responsable de stage : mehdi.ayadi@ia-tech.net
- Tél. du Responsable de stage : 31399120
- Mots-clés caractérisant votre rapport : Bluetooth Low Energy, UART, GAP, GATT, SMP, système temps réel, ESP32, AT Commands, STM32

SIGNATURES DES ENCADRANTS

Mr. Ayadi Mohamed Mehdi
(Innov-Alliance-Tech)

Mr. Abid Chieb Ameur
(Faculté des sciences de Tunis)

DÉDICACES

Je consacre ce recueil en signe de ma sincère reconnaissance :

À ma mère, je tiens à exprimer ma profonde gratitude pour tous les efforts que tu as déployés et pour l'amour inconditionnel que tu m'as toujours donné. Je sais que je ne pourrai jamais te rendre tout ce que tu m'as apporté, mais je m'efforcerai toujours d'être à la hauteur de tes attentes et de te rendre fière de moi.

À mon père, celui qui m'a appris que l'excellence est une nécessité indispensable, je tiens à te remercier du fond du cœur pour ton soutien constant, ton amour inconditionnel et ta motivation infinie.

À mon frère, être l'aîné n'a jamais été facile. Je veillerai toujours à être un bon exemple pour te guider. Je suis profondément reconnaissant de t'avoir dans ma vie, car ton existence est une source d'inspiration et de joie.

À mes amis de collocation et compagnons fidèles, Amen, Karim et Jaber, je tiens à exprimer ma profonde gratitude pour votre aide et votre soutien. Vous êtes non seulement une source de soutien, mais surtout une source de confiance. Vous incarnez l'exemple même d'une véritable amitié, celle qui traverse les temps difficiles.

À mes grands-parents, mes piliers solides, je tiens à exprimer toute ma gratitude. Votre amour sans condition, vos conseils éclairés et votre soutien constant ont été d'une valeur inestimable dans ma vie. Vous êtes une source de sagesse inépuisable, et je suis honoré d'avoir la chance de vous compter parmi mes modèles.

À mes collègues de l'IAT, Lamis, Nour, Khaoula, Mariem et Nedra, je tiens à exprimer ma profonde gratitude. Travailler avec vous est un véritable plaisir. Vous êtes une source quotidienne de soutien moral et de motivation.

Rayen Salhi

REMERCIEMENT

Je suis ravi de dédier ce travail pour exprimer ma sincère reconnaissance envers tous ceux qui ont contribué à sa réalisation.

Je tiens à exprimer ma sincère reconnaissance envers la société Innov-Alliance-Tech pour son accueil chaleureux.

Je souhaite particulièrement exprimer ma profonde gratitude envers Mr. Mohamed Mehdi AYADI, mon encadrant professionnel, ainsi que Mr. Yassine MEKSI, directeur du département de recherche et développement à l'I@T, pour leur bienveillance, leurs conseils précieux et le temps précieux qu'ils ont consacré pour m'assister dans la réalisation de mon projet.

Je remercie, également, mon encadrant académique Mr. Chiheb Ameur ABID, pour avoir accepté de superviser ce travail. Je lui suis infiniment reconnaissant pour sa patience, ses précieuses recommandations et sa totale confiance en moi.

Je tiens aussi à remercier le Staff de l'I@T, pour leur générosité et leur soutien tout au long de la période de notre stage.

Je tiens à remercier chaleureusement toutes les personnes qui, de près ou de loin, m'ont accordé leur précieuse aide dans la réalisation de ce travail.

Enfin, je souhaite adresser mes sincères remerciements à tous les membres du jury, à savoir Mr. Chaker ESSID et Mme. Narjes DOGGAZ, pour avoir accepté de consacrer leur temps et leur expertise à l'évaluation de ce travail. Je suis honoré par leur présence et leur contribution précieuse.

Rayen Salhi

TABLE DES MATIÈRES

Introduction générale	1
1 Présentation générale	3
1.1 Introduction	3
1.2 Description de la société	3
1.2.1 Présentation de l'organisme d'accueil	3
1.2.2 Domaine d'activités	4
1.2.3 Les services de société	5
1.3 Etude de l'existant et problématique	6
1.3.1 Cadre du projet	6
1.3.2 Etude de l'existant	7
1.3.3 Problématique	7
1.3.4 Solution proposée	8
1.4 Méthodologie de travail	8
1.5 Conclusion	9
2 Spécification des besoins	10
2.1 Introduction	10
2.2 Spécification des besoins non fonctionnels	10
2.3 Spécification des besoins fonctionnels	11
2.3.1 Gestion de la connectivité sans fil	11
2.3.2 Commande du servomoteur	11
2.3.3 Récupération des données du servomoteur	11
2.3.4 Mise à jour du firmware du microcontrôleur	12
2.4 Présentation des cas d'utilisation	12
2.4.1 Présentation des acteurs	12
2.4.2 Description des cas d'utilisation	12
2.4.3 Diagramme des cas d'utilisation global	16
2.5 Choix du matériel et de protocole	16
2.5.1 Choix de Bluetooth Low Energy	16
2.5.2 Choix de l'ESP32	17
2.5.3 Choix de l'UART	17
2.6 Conclusion	18

3 Conception du système	19
3.1 Introduction	19
3.2 Bluetooth Low Energy (BLE)	19
3.2.1 BLE	19
3.2.2 GAP	20
3.2.3 GATT	22
3.2.4 ATTRIBUT	23
3.2.5 SMP	24
3.3 UART	25
3.4 Commandes AT	26
3.5 Diagrammes de séquences	26
3.6 Diagramme d'activité de transfert de fichier	28
3.7 Structure des données du profil GATT	28
3.7.1 Premier service	29
3.7.2 Les caractéristiques du premier service	29
3.7.3 Deuxième service	30
3.8 Données UART	31
3.8.1 Commandes envoyées par microcontrôleur au module Bluetooth	31
3.8.2 Réponses renvoyées de l'ESP32 vers le microcontrôleur	32
3.8.3 Commande envoyées par module Bluetooth au microcontrôleur	32
3.9 Conclusion	33
4 Réalisation du système	34
4.1 Introduction	34
4.2 Environnement de développement	34
4.2.1 Environnement matériel	34
4.2.2 Environnement logiciel	37
4.3 Première approche de mise en œuvre	40
4.3.1 Configuration Matériel	41
4.3.2 Gestion de la connectivité BLE	41
4.3.3 Envoi de données de l'application vers microcontrôleur pour commande du servomoteur	43
4.3.4 Envoi de données de microcontrôleur vers l'application et récupération des données du servomoteur	45
4.3.5 Mise à jour du firmware du microcontrôleur	46
4.3.6 Limitation de la première approche de mise en œuvre	47
4.4 Deuxième approche de mise en œuvre	48
4.4.1 Gestion de la connectivité BLE	49
4.4.2 Envoi de données de l'application vers le microcontrôleur pour la commande du servomoteur	49
4.4.3 Envoi de données de microcontrôleur vers l'application et récupération des données du servomoteur	50
4.4.4 Mise à jour du firmware du microcontrôleur	51
4.5 Performances du système	52
4.6 Conclusion	52
Conclusion Générale	53
Netographie	54

TABLE DES FIGURES

1.1	Logo de I@T	4
1.2	Smart Metering	4
1.3	AVAYA et I@T	4
1.4	IoT	5
1.5	Biomédical	5
1.6	Photo d'un servomoteur	6
1.7	Diagramme de communication	8
1.8	Modèle en cascade	9
2.9	Diagramme d'état de transition pour le transfert de fichiers	15
2.10	Diagramme des cas d'utilisation global	16
3.11	Pile Bluetooth Low Energy	20
3.12	Méthode de "advertising"	21
3.13	GATT client et serveur	22
3.14	Attribut et hiérarchie de données	24
3.15	les interactions GATT	25
3.16	Paquet UART	25
3.17	Diagramme de séquence de configuration du BLE	27
3.18	Diagramme de séquence des transactions initialisées par l'utilisateur	27
3.19	Diagramme de séquence des transactions initialisées par le microcontrôleur	28
3.20	Diagramme d'activité de transfert de fichier	29
4.21	ESP-WROOM 32	35
4.22	Microcontrôleur STM32	36
4.23	Waveshare USB to TTL	36
4.24	Interface Homme-Machine du servomoteur	37
4.25	Espressif IoT Development Framework	38
4.26	Flash download tool	38
4.27	IAR Embedded Workbench	39
4.28	Docklight	39
4.29	Notepad++	40
4.30	NRF Connect	40
4.31	Connexion matérielle de la ESP32	41
4.32	Séquence de commandes pour configurer BLE	42
4.33	Disponibilité du réseau et permission de scan et appariement	43
4.34	Connexion et apparition des services et caractéristiques	43

4.35 Séquence de commandes de l'envoi de données par l'application	44
4.36 Réception des trames de données par le microcontrôleur	44
4.37 Séquence d'envoi de données ASCII par le microcontrôleur	45
4.38 Réception des trames de données par l'application	46
4.39 La première commande pour initialiser le transfert	47
4.40 la dernière commande pour marquer la fin du transfert	47
4.41 Séquence de commandes pour configurer BLE	49
4.42 Les trames reçues par le microcontrôleur de	49
4.43 Séquence de commandes de l'envoi de données par l'application	50
4.44 Séquence de commandes de l'envoi de données par le microcontrôleur	51
4.45 Réception des trames de données par l'application	51

LISTE DES TABLEAUX

2.1	Description du premier cas d'utilisation	13
2.2	Description du deuxième cas d'utilisation	14
2.3	Description du troisième cas d'utilisation	14
3.4	Premier service	29
3.5	Les caractéristiques du premier service	30
3.6	Deuxième service	30
3.7	Les caractéristiques du deuxième service	30
3.8	Commande d'envoi de données "advertising"	31
3.9	Commande d'activation de la diffusion "advertising"	31
3.10	Commande d'envoi de données	31
3.11	Notification du changement de donnée	31
3.12	Message de démarrage et disponibilité du module	32
3.13	Traitement réussi de la commande	32
3.14	Erreur de commande ou erreur survenue pendant l'exécution	32
3.15	Établissement d'une connexion	33
3.16	L'appariement SMP est maintenant terminé	33
3.17	Une opération d'écriture depuis une connexion Bluetooth	33
3.18	Une opération de lecture depuis une connexion Bluetooth	33
3.19	Une opération de déconnexion	33
4.20	Caractéristique d'ordinateur	34
4.21	Caractéristiques techniques de la carte de développement utilisée	35

LISTE DES ACRONYMES

BLE : Bluetooth Low Energy

IHM : Interface Homme Machine

I@T : Innov Alliance Tech

IoT : Internet of Things

PC : Personal Computer

PCB : Printed circuit board

RD : Recherche et Développement

GAP : Generic Access Profile

UART : Universal Asynchronous Receiver Transmitter

GATT : Generic ATTribute Profile

SIG : Special Interest Group

UUID : Universally Unique IDentifier

SMP : Security Management Protocol

AT : ATtention

GSM : Global System for Mobile Communications

CRC : Cyclic Redundancy Check

INTRODUCTION GÉNÉRALE

Le réseau Bluetooth Low Energy (BLE) occupe une place importante dans le domaine de la connectivité sans fil, et son utilisation est répandue dans de nombreux secteurs, dont celui de l'énergie. L'importance du réseau BLE réside dans sa capacité à fournir des solutions efficaces, fiables et sécurisées pour les systèmes de communication sans fil. Dans le domaine de l'énergie, le réseau BLE offre une valeur ajoutée significative en permettant la mise en place de solutions de surveillance, de contrôle et d'optimisation à distance.

Grâce à la connectivité Bluetooth LE, il est possible d'établir des communications sans fil entre des appareils tels que les IHM et les applications Android, facilitant ainsi la collecte des données en temps réel, le suivi des performances, la gestion des paramètres et la prise de décisions éclairées. Cette technologie permet également la mise en place de systèmes intelligents et d'objets connectés dans le domaine de l'énergie, offrant des avantages tels que la gestion de l'énergie, la détection des pannes, la surveillance des consommations et la réduction des coûts.

La société Innov-Alliance-Tech (I@T) s'investit pleinement dans le domaine de l'énergie et s'engage à fournir en permanence des solutions performantes à ses clients. Son département R&D est spécialisé dans la conception et le développement de cartes électroniques utilisées dans les servomoteurs, en étant constamment à la recherche des dernières avancées technologiques et en offrant une grande flexibilité dans le choix des technologies.

Avec l'évolution rapide des technologies de connectivité, le domaine de l'énergie a également connu un développement considérable, offrant de nouvelles opportunités pour des solutions innovantes. I@T se positionne en tant que fournisseur de solutions à la pointe de la technologie, permettant d'améliorer en permanence les performances des systèmes et de rester compétitif sur le marché.

Dans le cadre du développement de logiciels embarqués pour les interfaces homme machine (IHM) de servomoteurs, I@T s'engage à fournir une connectivité Bluetooth Low Energy (BLE) aux servomoteurs. Le projet s'inscrit dans cette démarche en développant une solution de communication BLE dans le cadre du projet de fin d'études pour l'obtention de la licence nationale en génie informatique, spécialité systèmes embarqués et IoT, à la Faculté des Sciences de Tunis. L'objectif de ce projet est de réaliser cette communication BLE entre un IHM et une application Android, afin d'être implémentée dans les IHM des servomoteurs.

Afin de trouver la solution adéquate, on analyse les solutions existantes en identifiant leurs limites et inconvénients. On effectue des recherches et on fait le choix de solutions alternatives. Ensuite, on développe le programme embarqué pour atteindre les fonctionnalités souhaitées et on procède aux tests nécessaires avant l'intégration.

Le développement se concentre sur deux axes essentiels : la communication sans fil (BLE) et la communication série (Universal Asynchronous Receiver Transmitter). Le rapport est structuré en quatre chapitres, présentés comme suit :

- Le *premier chapitre* aborde le cadre du projet, l'organisme d'accueil et la méthodologie de travail adoptée.
- Le *deuxième chapitre* se concentre sur la spécification des besoins fonctionnels et non fonctionnels, ainsi que sur le choix du matériel et des protocoles.
- Le *troisième chapitre* traite des notions techniques et de la conception du système, en mettant l'accent sur les aspects essentiels.
- Le *quatrième chapitre* présente l'environnement de développement, la réalisation et les tests effectués.
- Finalement, on expose une conclusion générale qui met en évidence les principaux aspects du travail et évoque les améliorations potentielles.

CHAPITRE 1 : PRÉSENTATION GÉNÉRALE

1.1 Introduction

Ce chapitre présente le cadre général du projet, incluant la présentation de l'organisme d'accueil, une description du contexte du projet, une étude de l'existant, la problématique et la proposition de solution. Enfin, on détaille la méthodologie qui a été mise en place pour bien mener le projet.

1.2 Description de la société

1.2.1 Présentation de l'organisme d'accueil

INNOV-ALLIANCE-TECH est une société de conseil technologique et l'ingénierie tunisienne, appartient au groupe IPS leader industriel dans la région de L'Afrique du Nord.

Elle offre un accompagnement à ses clients et partenaires sur les enjeux liés à l'organisation et la stratégie, dans les domaines de la conception, des tests et des technologies liées aux logiciels et aux produits.

Cette société possède une renommée internationale en matière de recherche et développement de solutions pour ses clients.

L'objectif de cette société est de développer son chiffre d'affaires tout en intégrant des jeunes ingénieurs et en investissant dans leur formation.

I@T a choisi ce nom pour refléter son engagement envers cet esprit d'innovation technologique qui caractérise le domaine de l'ingénierie.



FIGURE 1.1 – Logo de I@T

1.2.2 Domaine d'activités

I@T intervient dans plusieurs secteurs :

- ☞ **Energie** : INNOV-ALLIANCE-TECH s'est spécialisée dans le Smart Metering et dispose d'une expertise reconnue dans ce domaine. Elle offre ainsi des solutions et services d'architecture hautement maintenables et évolutives pour répondre aux besoins spécifiques de ses clients.

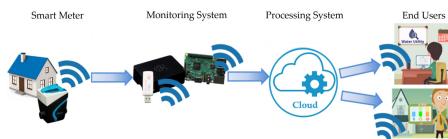


FIGURE 1.2 – Smart Metering

- ☞ **Télécommunication et Réseau** : INNOV-ALLIANCE-TECH a établi un partenariat avec le constructeur AVAYA afin de proposer des solutions de communications unifiées. Cette collaboration s'appuie sur un large portefeuille de produits conçus pour répondre aux besoins du marché des réseaux d'entreprise.



FIGURE 1.3 – AVAYA et I@T

- ☞ **Internet des objets** : Les équipes de l'I@T ont acquis une solide expertise dans le développement de capteurs communicants et de logiciels de prétraitement de données pour les passerelles(Gateway).



FIGURE 1.4 – IoT

- » **Biomédical** : I@T a conçu et développé une gamme variée d'équipements de laboratoire, comprenant des centrifugeuses, des agitateurs et des bains-marie.



FIGURE 1.5 – Biomédical

- ☞ **Automatisation** : L'entreprise propose des services d'automatisation industrielle visant à réduire les coûts et les erreurs humaines, à accroître la production, à augmenter la capacité de production et à améliorer l'uniformité.

1.2.3 Les services de société

La société propose une gamme variée de services adaptés aux besoins spécifiques de chaque client. Ces services sont minutieusement définis en fonction des choix et des exigences de la clientèle.

- ☞ **Service conseil :** I@T propose un accompagnement personnalisé à ses clients en étudiant et en sélectionnant les solutions existantes sur le marché qui répondent le mieux à leurs besoins. Ce service comprend l'analyse des solutions technologiques en fonction des critères spécifiques du client, la sélection de la solution optimale en fonction des besoins du client et l'étude des normes applicables pour les produits.

- ☞ **Ingénierie** : Il est divisé en plusieurs domaines qui sont :
 - Système embarqués : les protocoles de communications, logiciel embarqué, schéma, PCB.

- Logiciel et outils sur PC : design graphique, communication media.
- Logiciel Back-end et système d'information : stockage des données, gestion de mémoire, interface web.
- Test et validation : Etude de produits, analyse des normes et des standards, rédaction des essais de fonctionnels.

☞ COMPONENTS OFF THE SHELF : INNOV-ALLIANCE-TECH propose à ses clients et partenaires des piles de communication préconfigurées et adaptées à leurs besoins, avec plusieurs options économiques avantageuses. Il convient de mentionner notamment le stack de communication DLMS COSEM, Euridis, M-Bus, Wireless M-Bus, et bien d'autres encore.

1.3 Etude de l'existant et problématique

1.3.1 Cadre du projet

Innov Alliance Tech a développé des partenariats solides avec plusieurs entreprises multinationales, dont l'un des principaux fabricants mondiaux de servomoteurs électriques. Le département de R&D des systèmes embarqués est chargé de concevoir les logiciels embarqués pour les servomoteurs, notamment l'interface homme-machine (IHM), la carte mère, la carte de communication et la carte d'interface.



FIGURE 1.6 – Photo d'un servomoteur

Comme illustré sur la figure 1.6, un servomoteur est un système mécanique asservi qui associe une partie mécanique, constituée d'un moteur et d'un engrenage, à une partie électronique pour la commande et la régulation de la position en fonction de la mesure.

Un servomoteur est capable de résister à un effort statique tout en maintenant une position précise, avec une limite de débattement d'angle de 180 degrés, mais peut également permettre une rotation continue. Les servomoteurs sont fréquemment employés dans de multiples secteurs industriels, notamment dans le domaine de l'énergie.

Il convient toutefois de noter que le terme "servomoteur" est souvent utilisé à tort pour désigner le pré-actionneur électrique d'une vanne de régulation. Dans ce cas, l'asservissement de position est toujours effectué par un organe distinct appelé positionneur.

Les servomoteurs fournis par le partenaire de I@T sont utilisés dans divers secteurs tels que l'hydraulique, le pétrole, le gaz et l'industrie. Cependant, ces servomoteurs peuvent être coûteux et leur acquisition peut entraîner des limitations budgétaires pour certaines entreprises.

1.3.2 Etude de l'existant

Dans le cadre d'un projet de développement de logiciel embarqué pour une IHM de servomoteur, le département R&D de l'IAT a pour objectif de constamment trouver de nouvelles solutions et technologies pour offrir une grande diversité et flexibilité dans le choix des technologies et du matériel, et ainsi être en mesure de s'adapter aux changements du marché. Cependant, il est important de souligner que la solution actuellement utilisée présente certaines limites, notamment :

- **Coût élevé** : ce qui peut affecter la rentabilité du projet et impacter la capacité de l'entreprise à proposer des prix compétitifs sur le marché.
- **Disponibilité limitée** : ce qui peut entraîner des retards dans la fabrication des servomoteurs, notamment lors de l'acquisition de grandes quantités de matériel
- **Portée limitée** : ce qui oblige le technicien à se trouver à proximité immédiate du servomoteur pour effectuer des opérations.

1.3.3 Problématique

Selon l'étude de l'existant, l'optimisation du développement des logiciels embarqués pour les servomoteurs est crucial pour répondre aux exigences des partenaires et maintenir la compétitivité sur le marché mondial de la fabrication des servomoteurs électriques. Une problématique importante concerne la communication sans fil entre l'IHM du servomoteur et une application Android pour le contrôle et la mise à jour du firmware du microcontrôleur.

Actuellement, le module Bluetooth RN4020 utilisé dans la carte IHM rencontre des problèmes d'approvisionnement, entraînant des retards dans la production des cartes IHM. Afin de garantir une flexibilité technologique et matérielle, ainsi que de respecter les contraintes de temps réel, de sécurité et de fiabilité des données, le département R&D explore des alternatives au module Bluetooth RN4020 pour assurer une communication sans fil efficace et fiable dans l'IHM du servomoteur. La figure 1.7 représente le diagramme de communication entre les différents composants du système.



FIGURE 1.7 – Diagramme de communication

1.3.4 Solution proposée

Pour résoudre la problématique évoquée, on adopte une approche consistant à choisir un module de communication sans fil alternatif et un protocole de communication approprié. On développe ensuite un logiciel embarqué pour le module sélectionné, ainsi que les modifications nécessaires au logiciel du microcontrôleur. Cette approche vise à garantir une communication sans fil efficace, en assurant la fiabilité et la sécurité des transferts de données, tout en tenant compte des contraintes temps réel.

1.4 Méthodologie de travail

Une méthode de travail efficace est un élément clé pour améliorer l'organisation et la communication au sein d'une entreprise. On présente la démarche méthodologique et le cycle de vie qui ont été suivis pour la réalisation de ce projet. Après consultation avec les responsables de l'I@T, le modèle de cycle en cascade a été choisi. Ce modèle se compose de 5 phases linéaires et séquentielles, où chaque phase correspond à une spécialisation des tâches et dépend des résultats de la phase précédente. La figure 1.8 de la pages suivante, démontre les étapes de processus de développement avec le modèle en cascade.

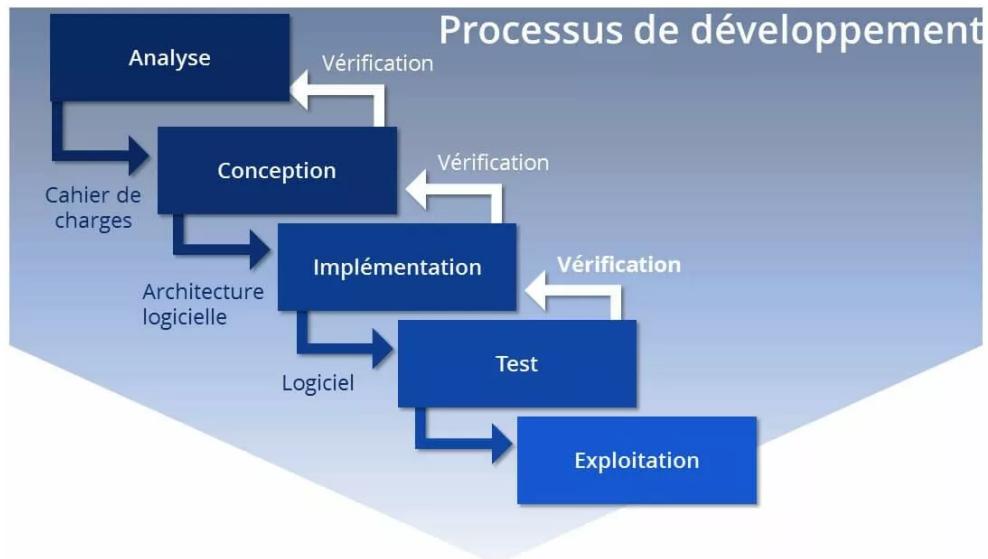


FIGURE 1.8 – Modèle en cascade

1.5 Conclusion

Le premier chapitre de ce rapport permet d'introduire la société et de présenter les différents projets en cours, en se focalisant notamment sur le projet de développement embarqué mené par le département R&D. Ce chapitre expose également la problématique posée par le projet, ainsi que la solution proposée et la méthodologie de travail adoptée. Dans le prochain chapitre, on se concentre sur la spécification des besoins fonctionnels nécessaires, ainsi que sur la justification du choix matériel et protocolaire pour concevoir l'interface homme-machine (IHM) du servomoteur et assurer une communication sans fil fiable avec l'application Android.

CHAPITRE 2 : SPÉCIFICATION DES BESOINS

2.1 Introduction

Ce chapitre se concentre sur la spécification des besoins fonctionnels et non fonctionnels du programme embarqué. Il vise à mieux comprendre les interactions entre les acteurs du système en présentant de manière illustrative des cas d'utilisation. De plus, on aborde les choix matériels et protocolaires qui sont pris en compte dans la conception du système.

2.2 Spécification des besoins non fonctionnels

Pour améliorer la qualité des services de l'application, il est nécessaire de prendre en compte des critères qui permettront d'améliorer l'expérience utilisateur, soit aux besoins non fonctionnels suivants :

- **Réactivité** : La réactivité du système est essentielle pour garantir une interaction en temps réel avec le servomoteur en réponse aux commandes envoyées par l'application mobile ou le microcontrôleur. Les temps de réponse doivent être rapides pour assurer un contrôle précis et en temps réel du servomoteur.
- **Sécurité** : Pour assurer la sécurité des communications sans fil, il est nécessaire d'utiliser des protocoles de chiffrement pour empêcher l'accès non autorisé aux données échangées, ainsi que de mettre en place un système d'authentification pour garantir l'identification des utilisateurs autorisés.
- **la fiabilité des données** : Il est essentiel de s'assurer que les données transmises entre l'application mobile, le microcontrôleur et le servomoteur sont fiables et cohérentes. Cela peut être atteint en utilisant des mécanismes de vérification.

Par exemple, la vérification de la somme de contrôle permet de s'assurer que les données reçues sont correctes. De plus, la mise en place de mécanismes de correction d'erreur et de retransmission automatique en cas de perte de données contribue à garantir la fiabilité des informations échangées.

2.3 Spécification des besoins fonctionnels

Le développement du système de communication sans fil de servomoteur doit prendre en compte les points précis du cahier des charges. Les besoins fonctionnels suivants doivent être pris en compte :

2.3.1 Gestion de la connectivité sans fil

Le système doit être capable de gérer la connectivité sans fil en permettant la configuration des modes de fonctionnement des périphériques, l'initiation et l'activation de la connexion, l'appairage des périphériques, la gestion de la connexion et la déconnexion entre les différents périphériques, ainsi que la gestion des échanges de données et le cryptage des données entre ces derniers.

2.3.2 Commande du servomoteur

Le besoin fonctionnel de commande de servomoteur exige que le système permette à l'utilisateur de modifier en temps réel différents paramètres de commande du servomoteur. Ces paramètres de configuration sont associés à un utilisateur spécifique, tels que le nombre d'écritures de configuration utilisateur, le type de bus de configuration, le format de date, la langue, les paramètres de relais, le mot de passe, le pas de vis, les réglages de synchronisation, les réglages de vitesse, l'avantage mécanique, le rapport de réduction de l'engrenage, l'emplacement, les limites de couple, et d'autres encore.

2.3.3 Récupération des données du servomoteur

La récupération des données du servomoteur est une fonctionnalité essentielle du système qui permet de collecter une variété d'informations sur le servomoteur. Ces informations comprennent notamment le numéro de série du servomoteur et son état actuel, y compris sa position d'ouverture, les couples générés, les alarmes signalées, etc. Il est primordial que cette fonctionnalité fournisse des données précises, fiables et rapides sur le fonctionnement du servomoteur afin de permettre aux utilisateurs de surveiller et de contrôler efficacement le système.

2.3.4 Mise à jour du firmware du microcontrôleur

Le système doit permettre à l'utilisateur de mettre à jour le micrologiciel (firmware) du microcontrôleur en transférant un fichier binaire correspondant.

2.4 Présentation des cas d'utilisation

2.4.1 Présentation des acteurs

Dans ce programme embarqué, il y a deux acteurs principaux : l'application Android et le microcontrôleur.

L'application Android est l'interface utilisateur qui permet à l'utilisateur de communiquer avec le microcontrôleur et de contrôler le servomoteur.

Le microcontrôleur Le rôle principal du microcontrôleur est de commander et récupérer les données du servomoteur. En outre, il est également chargé de gérer la connectivité sans fil et de communiquer avec l'application Android à travers le module Bluetooth.

2.4.2 Description des cas d'utilisation

- **Gestion de la connectivité sans fil**

Chaque servomoteur possède des informations spécifiques, telles que le numéro de série, la version du logiciel, la maintenance et le modèle, qui varient d'un servomoteur à l'autre et sont initialisées par le fabricant. Ces informations sont nécessaires pour la diffusion sur le réseau. Ainsi, lorsque le microcontrôleur souhaite démarrer la diffusion, il doit envoyer une commande contenant ces informations ainsi qu'une commande d'activation au module de communication. Cette étape est effectuée lors de l'activation du servomoteur ou en cas d'erreur de diffusion. La procédure à suivre lorsque le microcontrôleur souhaite démarrer la diffusion est décrite dans le tableau 2.1 de la page suivante.

- **Échange des données entre servomoteur et application mobile**

Le microcontrôleur est responsable de l'échange de deux types de données avec le module de communication sans fil : des données cycliques et des données non cycliques. Ces deux types de données seront présentés en détail dans la page suivante.

Cas n°	Premier cas d'utilisation
Acteur(s) :	Microcontrôleur
Objectif :	Gestion de la connectivité sans fil
Précondition(s) :	-La réception d'une commande contenant la configuration des paramètres du réseau puis une commande d'activation
Post-condition(s) :	-Les demandes de scan et de connexion sont autorisées depuis n'importe quel appareil central.
Scénario nominal :	<ul style="list-style-type: none"> -Le module de communication reçoit une commande contenant la configuration des paramètres du réseau. -Il utilise cette configuration pour initialiser le réseau. -Le module de communication reçoit une commande d'activation -Le module commence la diffusion

TABLE 2.1 – Description du premier cas d'utilisation

- ☞ **Les données cycliques** : qui font référence à l'état actuel du servomoteur, tel que sa position d'ouverture et les éventuelles alarmes. Ces données doivent être transmises en temps réel. Le microcontrôleur est programmé pour envoyer ces données à intervalles réguliers de 200 millisecondes, afin de fournir des mises à jour fréquentes sur l'état du servomoteur.
- ☞ **Les données non cycliques** : comprennent des informations générales sur le servomoteur, telles que son numéro de série, sa version de logiciel, etc. Contrairement aux données cycliques qui nécessitent une transmission en temps réel, les données non cycliques ne sont pas soumises à des contraintes temporelles. Elles sont transmises selon la demande de l'utilisateur, lorsqu'il souhaite les consulter ou les modifier. Ces données fournissent des détails supplémentaires sur le servomoteur et permettent une configuration et une gestion plus approfondies de l'appareil.

Quel que soit le type de données, on peut identifier deux cas d'utilisation :

- ☞ L'utilisateur souhaite envoyer des données pour modifier l'un des paramètres du servomoteur, ce qui aura un impact sur son fonctionnement. Dans cette situation, le module de communication est responsable de notifier le microcontrôleur du changement, afin qu'il puisse réagir en conséquence.

Cela permet à l'utilisateur de contrôler activement le servomoteur en ajustant ses paramètres selon ses besoins. Le tableau 2.2 décrit le cas où l'utilisateur veut commander le servomoteur.

Cas n°	Deuxième cas d'utilisation
Acteur(s) :	Application mobile
Objectif :	Commande du servomoteur
Précondition(s) :	L'appareil est connecté
Évènement déclencheur :	Envoi des données sur les paramètres du servomoteur
Post-condition(s) :	Envoi des données pour modifier l'un des paramètres
Scénario nominal :	<ul style="list-style-type: none"> -L'application envoie des données au module de communication. -Le module de communication : <ul style="list-style-type: none"> -reçoit ces données via le réseau sans fil. -enregistre les nouvelles valeurs -envoie ces données sur la voie série au microcontrôleur pour le notifier au changement

TABLE 2.2 – Description du deuxième cas d'utilisation

Cas n°	Troisième cas d'utilisation
Acteur(s) :	Application mobile
Objectif :	Récupération des données du servomoteur
Précondition(s) :	<ul style="list-style-type: none"> -L'appareil est connecté -Le microcontrôleur notifie le module de chaque changement.
Évènement déclencheur :	Demande de lecture des données sur les paramètres du servomoteur
Post-condition(s) :	La réception des valeurs demandées
Scénario nominal :	<ul style="list-style-type: none"> -L'application mobile demande des données sur les paramètres du servomoteur -Le module sans fil renvoie les valeurs enregistrées

TABLE 2.3 – Description du troisième cas d'utilisation

☞ D'autre part, l'utilisateur peut également souhaiter consulter le contenu des données du servomoteur. Dans ce cas, le microcontrôleur est chargé de récupérer ces données et de les envoyer au module de communication. Il est important que le microcontrôleur notifie le module de chaque changement dans les données du servomoteur, afin que l'utilisateur puisse recevoir des informations à jour sur l'état et la configuration du servomoteur. Le tableau 2.3 de la page 14, décrit le cas où l'utilisateur veut récupérer des données du servomoteur.

- **Mise à jour du firmware du microcontrôleur**

L'application mobile envoie la taille du fichier binaire au microcontrôleur, qui répond en indiquant sur combien de paquets le fichier doit être divisé.

Ensuite, le microcontrôleur demande à l'application d'envoyer les paquets spécifiques, du premier jusqu'au dernier, et l'application répond en envoyant les données correspondantes. Si le microcontrôleur détecte une corruption dans une trame, il demande à l'application de la renvoyer. Cela implique un échange de données bidirectionnel entre l'application mobile et le microcontrôleur.

La figure 2.9 présente le diagramme d'état de transition pour le transfert de fichiers. L'application envoie les paquets de données et reçoit les demandes de renvoi en cas de corruption, tandis que le microcontrôleur reçoit les paquets et envoie les demandes de renvoi.



FIGURE 2.9 – Diagramme d'état de transition pour le transfert de fichiers

2.4.3 Diagramme des cas d'utilisation global

La figure 2.10 décrit le diagramme de cas d'utilisation global.

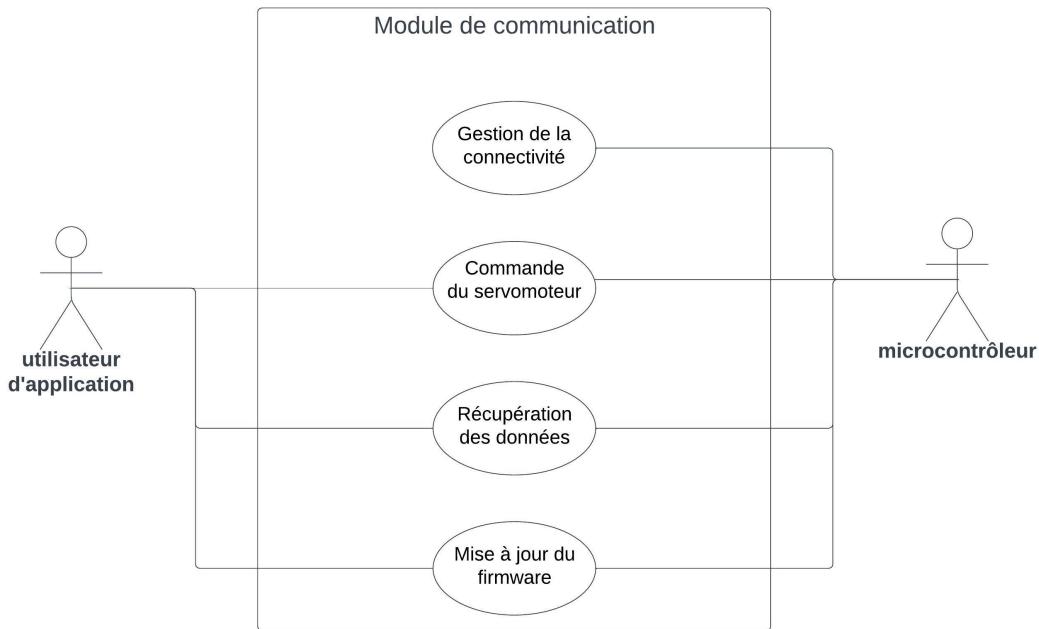


FIGURE 2.10 – Diagramme des cas d'utilisation global

2.5 Choix du matériel et de protocole

2.5.1 Choix de Bluetooth Low Energy

Le choix du Bluetooth Low Energy (BLE) a été justifié par plusieurs critères :

- **La portée du réseau :** Le Bluetooth LE permet des communications sans fil sur de courtes distances, jusqu'à plusieurs dizaines de mètres. Cette caractéristique le rend parfaitement adapté à ce cas d'utilisation, où l'IHM du servomoteur et l'application mobile se trouvent à proximité l'une de l'autre. Ainsi, il offre à l'utilisateur une distance significative pour contrôler les servomoteurs à distance.
- **Débit de transmission :** le Bluetooth LE est capable de fournir des vitesses de transmission élevées allant jusqu'à 2 Mbits/s, ce qui permet un transfert rapide et efficace des données. Cette caractéristique répond parfaitement aux exigences temps réel.
- **Intégrité des données :** le Bluetooth LE offre un haut niveau de fiabilité des données, avec des mécanismes de correction d'erreur et de retransmission automatique en cas de perte de données.

- **Confidentialité des données :** le Bluetooth LE offre un niveau élevé de sécurité des données grâce à des mécanismes de cryptage des données, d'authentification et d'autorisation implémenté selon le GAP (Generic Access Profile). Cela garantit que les données sont protégées contre les intrusions malveillantes et que seules les parties autorisées peuvent y accéder.

Dans l'ensemble, le choix du Bluetooth a été justifié par sa capacité à répondre aux critères de portée, de vitesse de transmission élevée, de fiabilité des données et de sécurité.

2.5.2 Choix de l'ESP32

La sélection de l'ESP32 a été justifiée par plusieurs critères importants pour ce projet. Tout d'abord, sa connectivité Bluetooth est compatible avec une portée allant jusqu'à 100 mètres et une vitesse de transmission allant jusqu'à 2 Mbit/s, ce qui est parfaitement adapté à cette application.

De plus, l'ESP32 est largement disponible sur le marché et est proposé à un coût abordable en raison de son grand volume de production, ce qui en fait un choix économique pour ce projet.

En outre, l'ESP32 offre des performances exceptionnelles grâce à son microcontrôleur puissant, qui offre la possibilité de programmer des programmes complexes et spécifiques grâce à sa puissance de calcul élevée.

Enfin, l'ESP32 dispose d'une variété d'interfaces de communication pour la transmission de données entre périphériques, ce qui facilite la connectivité avec d'autres composants du système. Dans l'ensemble, le choix de l'ESP32 a été justifié par sa connectivité Bluetooth efficace, sa haute disponibilité sur le marché, ses performances exceptionnelles, sa programmabilité et ses nombreuses interfaces de communication pour une intégration facile.

2.5.3 Choix de l'UART

Le choix de l'interface de communication UART a été justifié par plusieurs critères importants pour ce projet. Tout d'abord, l'UART est une interface de communication largement utilisée dans les systèmes embarqués, ce qui garantit une haute compatibilité avec une variété de microcontrôleurs et de dispositifs électroniques.

En outre, l'UART est une interface de communication série simple, qui utilise seulement deux fils pour la transmission de données. Cela réduit le nombre de connexions nécessaires entre les composants électroniques et simplifie le câblage, ce qui est avantageux pour les applications embarquées où l'espace est limité.

Enfin, l'UART est capable de transmettre des données à des vitesses élevées, ce qui est important pour ce système où on doit transférer des données rapidement et de manière fiable entre l'IHM du servomoteur et l'ESP32.

Dans l'ensemble, le choix de l'interface de communication UART a été justifié par sa haute compatibilité, sa simplicité de câblage et sa capacité à transmettre des données à des vitesses élevées.

2.6 Conclusion

Ce chapitre permet de détailler les besoins fonctionnels et non fonctionnels du programme embarqué pour la communication sans fil entre l'application Android et l'IHM du servomoteur. On présente également les différents cas d'utilisation et justifions le choix matériel. Le chapitre suivant aborde les notions techniques requises par les spécifications établies dans ce chapitre, ainsi que la conception du système.

CHAPITRE 3 : CONCEPTION DU SYSTÈME

3.1 Introduction

Dans ce chapitre, on aborde les notions techniques découlant des choix protocolaires. On examine les différentes étapes de conception du système et présentons en détail les choix techniques qui ont été faits pour mettre en place la structure de données et les protocoles de communication nécessaires afin d'assurer le bon fonctionnement du système. On explique également les différents cas d'utilisation du système et présentons les diagrammes de séquence correspondants pour mieux comprendre les interactions entre les différents acteurs du système. Enfin, on décrit le comportement du système en utilisant des diagrammes d'états-transitions pour illustrer les transitions d'état qui se produisent en réponse aux entrées du système.

3.2 Bluetooth Low Energy (BLE)

3.2.1 BLE

Bluetooth est une norme de communication qui permet la transmission bidirectionnelle de données sur de très courtes distances en utilisant des ondes radio UHF dans la bande de fréquence 2,4 GHz.

Bluetooth Low Energy (BLE), également connu sous le nom de "Bluetooth Smart", est une version plus légère du Bluetooth classique et a été inclus dans la spécification de base Bluetooth 4.0. Bien qu'il partage des similitudes avec le Bluetooth classique, le BLE a des origines différentes et a été initialement développé par Nokia sous le nom de projet interne "Wibree" avant d'être adopté par le Bluetooth SIG. Dans ce qui suit, on donne un aperçu rapide de BLE, y compris comment les données sont organisées, annoncent leur présence et comment ils peuvent établir une connexion.

La figure 3.11 illustre les différents composants de la pile BLE.

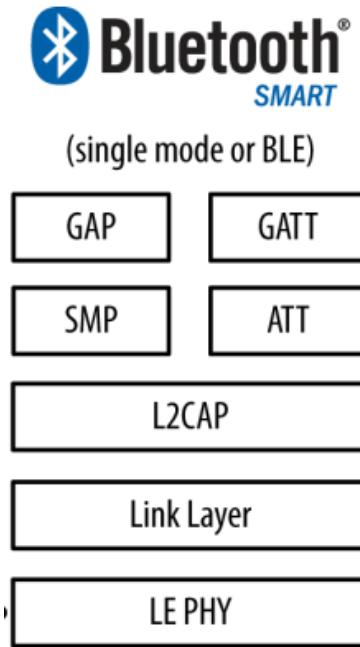


FIGURE 3.11 – Pile Bluetooth Low Energy

3.2.2 GAP

Le profil GAP (Generic Access Profile) est chargé de la gestion des connexions et de "advertising" (publicité) dans le contexte de la technologie Bluetooth. Son rôle est de rendre votre appareil détectable pour les autres et de définir les interactions possibles (ou impossibles) entre deux appareils.

Rôles des appareils :

Le protocole GAP (Generic Access Profile) définit plusieurs rôles pour les appareils Bluetooth, mais il y a deux concepts importants à retenir : les appareils centraux et les appareils périphériques.

- **Les appareils périphériques** sont des petits appareils à faible puissance et avec des ressources limitées qui peuvent se connecter à un appareil central plus puissant.
- **Les appareils centraux** sont généralement les téléphones mobiles ou les tablettes qui disposent d'une puissance de traitement et de mémoire bien plus élevée, et auxquels les appareils périphériques se connectent.

Méthode de "advertising"

Les appareils équipés de la technologie BLE sont repérés à l'aide d'une méthode basée sur la diffusion de paquets "advertising", comme illustré dans la figure 3.12. Pour éviter les interférences, cette méthode utilise trois canaux distincts. L'appareil qui émet le paquet "advertising" le diffuse sur au moins un de ces trois canaux, à intervalles réguliers, appelés ""advertising" interval". Afin d'éviter plusieurs collisions successives, un délai aléatoire pouvant atteindre 10 millisecondes est ajouté à chaque "advertising" interval. Le scanner écoute le canal pendant une période appelée "scan window", qui se répète périodiquement à chaque scan interval. La latence de découverte dépend de ces trois paramètres ("advertising" interval, scan interval et scan window) et est donc déterminée de manière probabiliste.

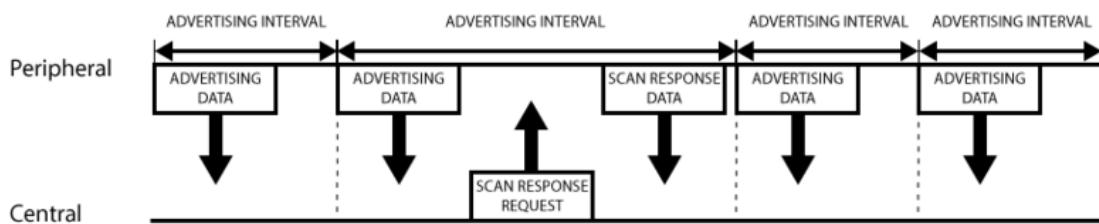


FIGURE 3.12 – Méthode de "advertising"

"advertising" et "Scan Response Data"

Il existe deux façons d'envoyer des paquets "advertising" avec GAP. Les charges utiles de données "advertising" et les charges utiles de réponse de scan.

Les deux charges utiles sont identiques et peuvent contenir jusqu'à 31 octets de données, mais seule la charge utile de données "advertising" est obligatoire, car c'est la charge utile qui sera constamment transmise depuis le périphérique pour informer les périphériques centraux à portée de sa présence. La charge utile de réponse de scan est une charge utile secondaire facultative que les périphériques centraux peuvent demander, et permet aux concepteurs de périphériques d'ajouter un peu plus d'informations dans la charge utile "advertising" telles que des chaînes de caractères pour un nom de périphérique, etc.

Une fois que vous avez établi une connexion entre votre périphérique et un dispositif central, le processus de "advertising" s'arrêtera généralement. Vous ne pourrez plus envoyer de paquets "advertising", vous utiliserez plutôt des services et des caractéristiques GATT pour communiquer dans les deux sens.

3.2.3 GATT

GATT est l'acronyme de Generic ATtributed Profile, et il définit la manière dont deux appareils Bluetooth Low Energy transfèrent des données en utilisant des concepts appelés Services et Caractéristiques

La communication est basée sur les rôles suivants :

- **Client** : Un périphérique qui lance des commandes et des requêtes GATT et accepte des réponses, par exemple un ordinateur ou un smartphone.
- **Serveur** : Un périphérique qui reçoit des commandes et des demandes du GATT, et renvoie des réponses, par exemple un capteur de température.

GATT entre en jeu une fois qu'une connexion dédiée est établie entre deux appareils, ce qui signifie que vous avez déjà passé par le processus de "advertising" régi par GAP. La figure 3.13 présente un exemple de périphériques BLE avec les rôles GATT et GAP correspondants.



FIGURE 3.13 – GATT client et serveur

Les transactions GATT en BLE

Selon la spécification Bluetooth Core, les transactions GATT en BLE sont basées sur des objets imbriqués de haut niveau appelés profils, services et caractéristiques, que l'on peut voir dans l'illustration ci-dessous :

- **Un profil** n'existe pas réellement sur le périphérique BLE lui-même, c'est simplement une collection prédéfinie de Services qui a été compilée soit par le Bluetooth SIG soit par les concepteurs du périphérique. Le profil de fréquence cardiaque, par exemple, combine le Service de fréquence cardiaque et le Service d'informations sur le périphérique.

- **un service** est défini comme un ensemble de données et de comportements liés qui permettent d'accomplir une fonction ou une fonctionnalité spécifique. Ce service peut comporter des caractéristiques obligatoires et facultatives. Par exemple, un service de thermomètre médical peut inclure des caractéristiques pour une valeur de mesure de la température et un intervalle de temps entre les mesures. Il est important de noter que le groupe Bluetooth SIG a déjà prédéfini certains services, mais il est également possible de créer des services personnalisés.

Chaque service se distingue des autres services par un identifiant numérique unique appelé UUID, qui peut être soit sur 16 bits (pour les services BLE officiellement adoptés) soit sur 128 bits (pour les services personnalisés).

- **Une caractéristique** est définie comme une valeur associée à un service, qui comporte des propriétés et des informations de configuration sur la façon dont cette valeur peut être accédée et affichée. Il s'agit d'une donnée échangée entre un client et un serveur, telle que la tension de la batterie actuelle. Les caractéristiques contiennent également des métadonnées, des unités et des paramètres de sécurité.

Tout comme les services, chaque caractéristique se distingue par un UUID pré-défini de 16 bits ou 128 bits, et vous êtes libre d'utiliser les caractéristiques standard définies par Bluetooth SIG (ce qui garantit l'interopérabilité entre tout matériel et logiciel compatible BLE) ou de définir vos propres caractéristiques personnalisées qui ne seront comprises que par votre périphérique et votre logiciel.

- **Un descripteur** est un élément qui fournit des informations supplémentaires sur une caractéristique, telles que les unités de mesure (par exemple Celsius) ou les valeurs maximales et minimales que le capteur peut mesurer. Chaque caractéristique puisse avoir un nombre quelconque de descripteurs.

La figure 3.14 de la page suivante illustre l'hiérarchie des données dans un profil GATT.

3.2.4 ATTRIBUT

Les services, les caractéristiques et les descripteurs sont appelés attributs. Ils sont identifiés par des UUID. Une plage d'UUID (de la forme xxxxxxxx-0000-1000-8000-00805F9B34FB) a été réservée pour les attributs standards. Pour plus d'efficacité, ces identificateurs sont représentés par des valeurs de 16 bits ou de 32 bits plutôt que par les 128 bits requis.

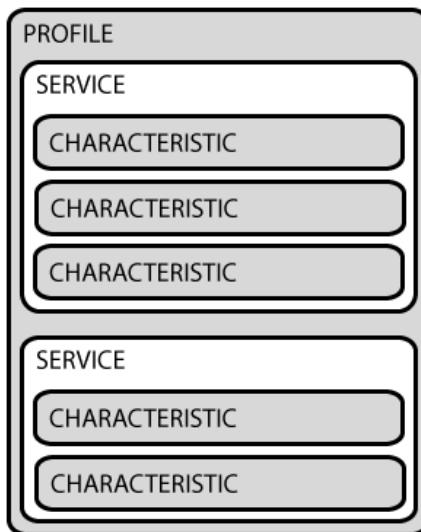


FIGURE 3.14 – Attribut et hiérarchie de données

Par exemple, le service d’informations sur le périphérique (Device Information) a le code abrégé 0x180A, plutôt que 0000180A-0000-1000-8000-00805F9B34FB. Les UUID de 128 bits, sont ceux utilisés pour créer ses propres services et caractéristiques personnalisés.

3.2.5 SMP

Le protocole de gestion de la sécurité (SMP : Security Management Protocol) dans Bluetooth Low Energy (BLE) est utilisé pour établir et gérer les connexions sécurisées entre les périphériques BLE. Il fournit des fonctionnalités de chiffrement, d’authentification et de confidentialité des données pour assurer la sécurité des communications sans fil.

Le SMP utilise des méthodes de chiffrement symétrique et asymétrique pour protéger les données échangées entre les périphériques. Il prend en charge différentes méthodes d’authentification, telles que les codes d’accès, les clés de sécurité et les échanges de clés publiques, pour garantir l’identité des périphériques et prévenir les attaques de type "man-in-the-middle".

En plus de cela, le SMP met en œuvre des mécanismes de confidentialité pour empêcher l’interception et la lecture des données sensibles échangées entre les périphériques. Il utilise des clés de chiffrement générées lors de l’établissement de la connexion pour crypter les données et les rendre illisibles pour les tiers.

Grâce à ces fonctionnalités de sécurité, le SMP garantit l’intégrité et la confidentialité des données dans les communications BLE, offrant ainsi un environnement sécurisé pour les applications et les services basés sur cette technologie.

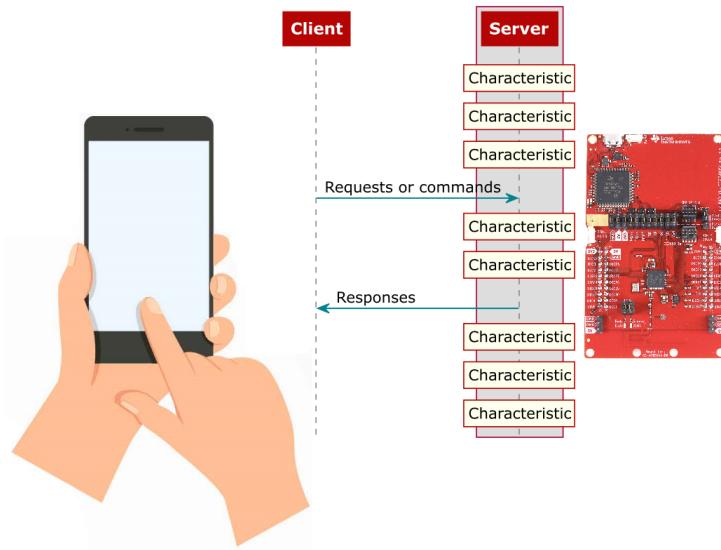


FIGURE 3.15 – les interactions GATT

Le protocole GATT propose des commandes permettant la lecture (transfert de données du serveur au client) et l'écriture (du client vers le serveur) des valeurs des caractéristiques. De plus, il permet l'utilisation de notifications et d'indications. Lorsqu'un client demande une notification pour une caractéristique spécifique du serveur, celui-ci envoie la valeur au client dès qu'elle est disponible, ce qui évite au client de devoir interroger en permanence le serveur. Par exemple, un serveur de capteur de température peut avertir son client chaque fois qu'il effectue une mesure. L'indication est similaire à la notification, à la différence qu'elle exige une réponse du client pour confirmer qu'il a bien reçu le message. La figure 3.15 montre un exemple d'une transaction entre un client et un server GATT.

3.3 UART

Le récepteur-transmetteur asynchrone universel (UART) est un composant matériel informatique permettant la communication série asynchrone. Il est possible de configurer le format des données et les vitesses de transmission. Les bits de données sont envoyés un par un, du moins significatif au plus significatif, et sont encadrés par des bits de début et de fin pour permettre une gestion précise du chronométrage par le canal de communication. La figure 3.16 explique la composition d'un paquet de donnée UART.

Start Bit (1 bit)	Data Frame (5 to 9 Data Bits)	Parity Bits (0 to 1 bit)	Stop Bits (1 to 2 bits)
------------------------	------------------------------------	-------------------------------	------------------------------

FIGURE 3.16 – Paquet UART

Les niveaux de signalisation couramment utilisés sont RS-232 et RS-485. Les premières machines à écrire utilisaient des boucles de courant. Les UART permettent une méthode largement adoptée et peu coûteuse pour réaliser un échange de données en duplex intégral ou en demi-duplex entre différents périphériques. Chaque contrôleur UART est configurable de manière indépendante avec des paramètres tels que le débit binaire, la longueur des bits de données, l'ordre des bits, le nombre de bits d'arrêt, le bit de parité, etc. Tous les contrôleurs sont compatibles avec les périphériques compatibles UART de différents fabricants et peuvent également prendre en charge les protocoles de l'Infrared Data Association (IrDA).

3.4 Commandes AT

Les commandes AT (Attention) sont des commandes de communication série utilisées pour échanger des données entre des équipements électroniques tels que les modems, les routeurs, les modules GSM et les microcontrôleurs. Les commandes AT sont prédéfinies et reposent sur une syntaxe simple et standardisée. Elles sont transmises à l'équipement en utilisant des caractères ASCII, et les réponses sont renvoyées sous forme de messages texte. Ces commandes servent à contrôler et configurer des équipements à distance, et elles sont largement utilisées dans les systèmes de communication sans fil et les applications IoT.

3.5 Diagrammes de séquences

Dans cette partie, on se concentrons sur l'analyse détaillée du fonctionnement du système en utilisant des diagrammes de séquences. Ces diagrammes permettent de visualiser les interactions entre les différents composants du système : le microcontrôleur, le module Bluetooth et l'application Android, et d'observer le déroulement des actions lors de l'exécution des différentes transactions.

La figure 3.17 de la page 27 présente le diagramme de séquence de configuration du Bluetooth Low Energy (BLE). Ce diagramme illustre les étapes nécessaires pour configurer le module de communication sans fil en tant que serveur BLE, mettre en place les services requis et établir les paramètres de sécurité et de cryptage appropriés.

La figure 3.18 de la page 27 présente le diagramme de séquence des transactions initiées par l'utilisateur. Ce diagramme met en évidence les actions que l'utilisateur peut entreprendre pour envoyer des commandes, modifier les paramètres du servomoteur et consulter les données de ce dernier.

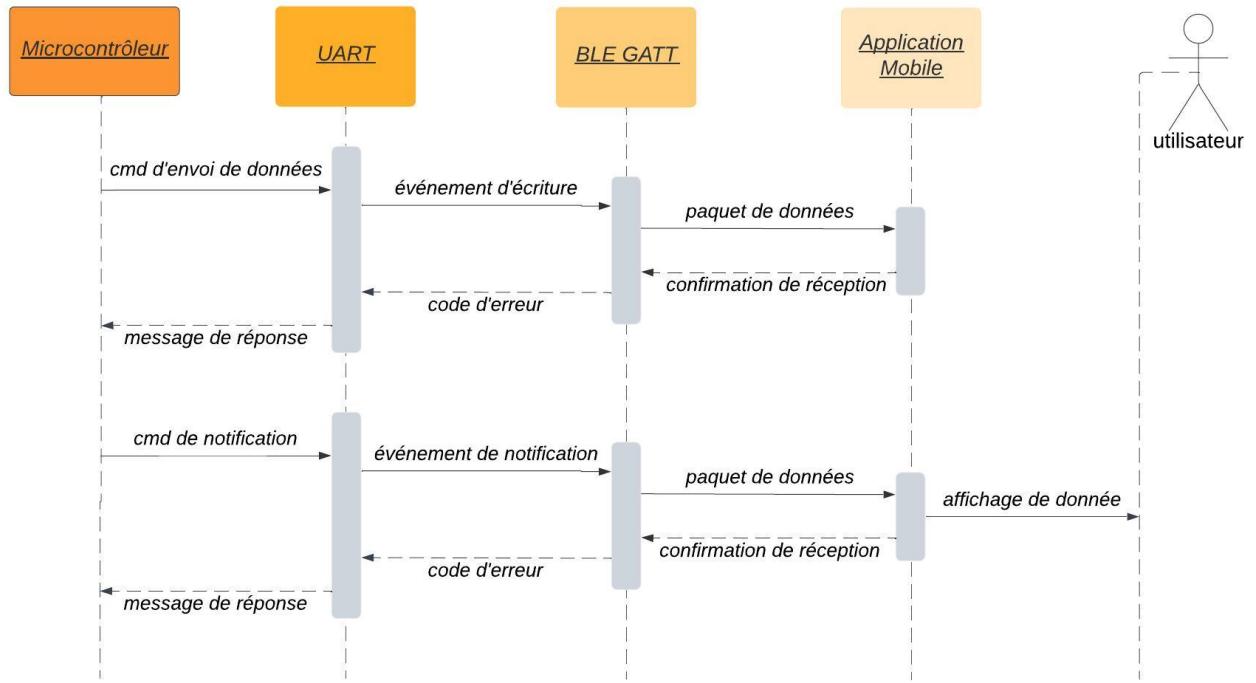


FIGURE 3.17 – Diagramme de séquence de configuration du BLE

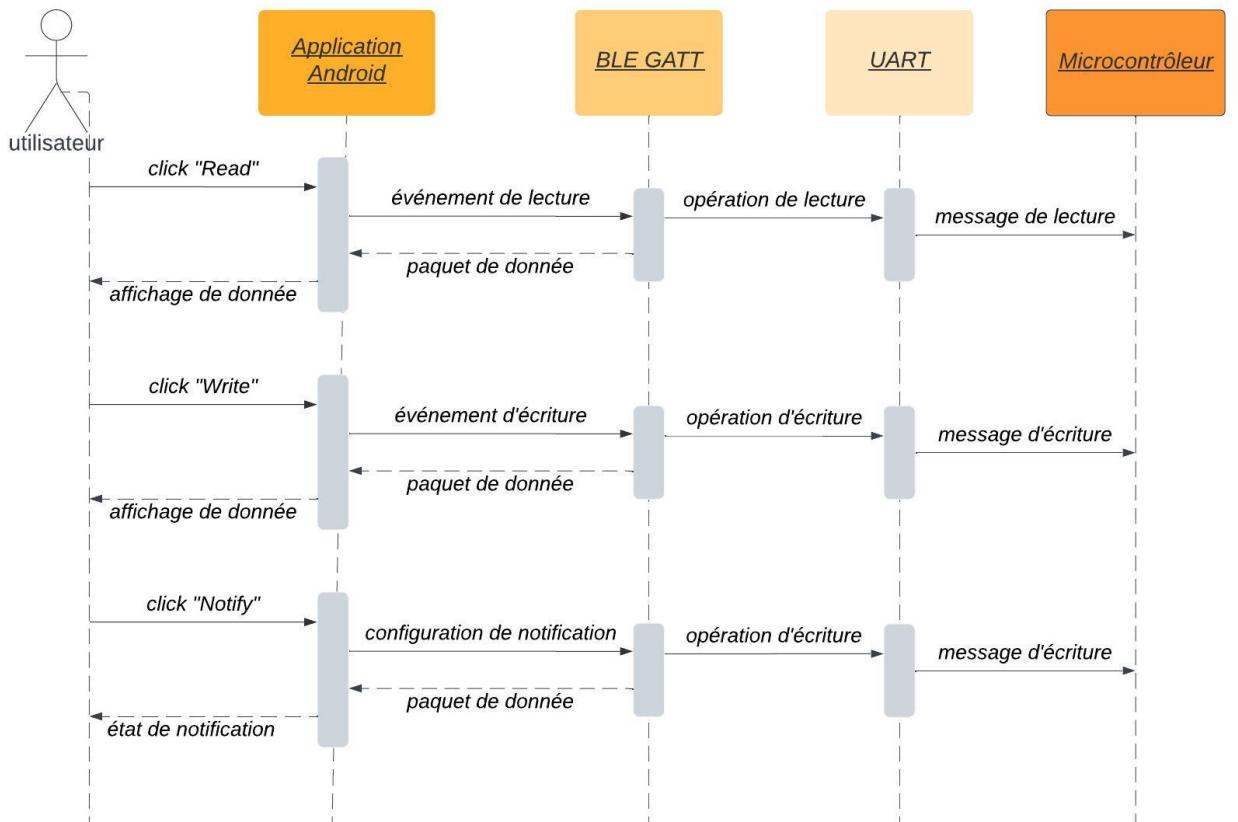


FIGURE 3.18 – Diagramme de séquence des transactions initialisées par l'utilisateur

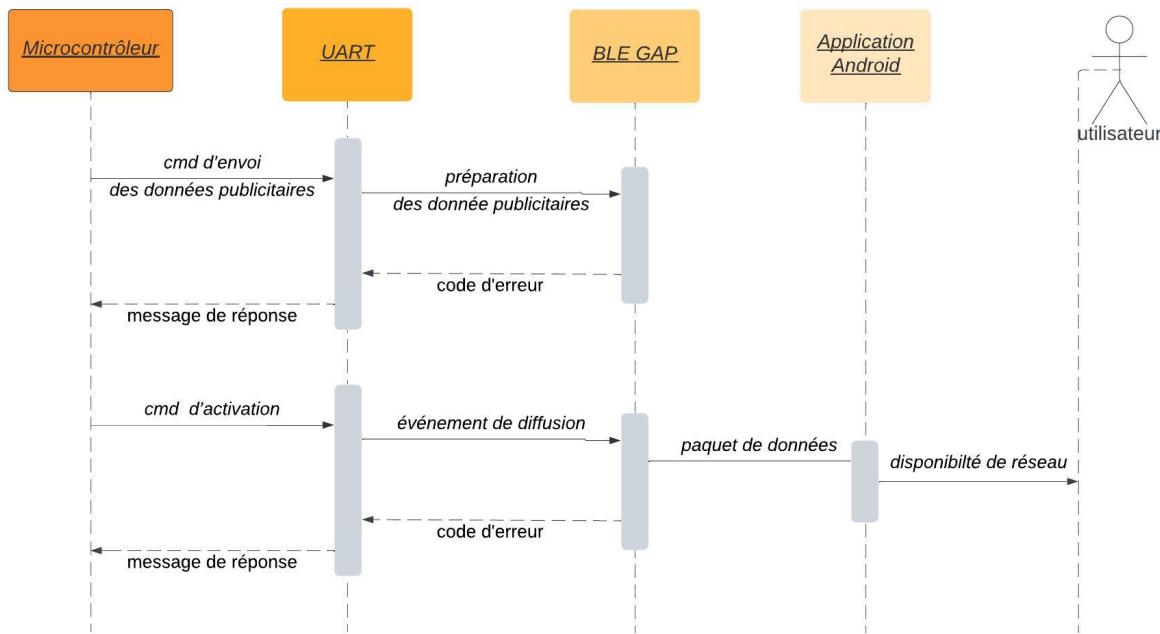


FIGURE 3.19 – Diagramme de séquence des transactions initialisées par le microcontrôleur

La figure 3.19 présente le diagramme de séquence des transactions initiées par le microcontrôleur. Ce diagramme représente les actions que le microcontrôleur effectue pour récupérer les données du servomoteur, mettre à jour les informations en temps réel et notifier l'utilisateur des changements éventuels.

3.6 Diagramme d'activité de transfert de fichier

Le diagramme d'activité de la figure 3.20 de la page 29 décrit le processus de transfert de fichiers au sein du système. Il présente de manière détaillée les étapes nécessaires pour envoyer un fichier de l'application Android vers le microcontrôleur. Le diagramme met en évidence les actions clés effectuées par chaque composant du système et illustre comment ils interagissent pour garantir le succès du transfert de fichier.

3.7 Structure des données du profil GATT

Afin d'assurer un transfert de données fiable et performant, on crée une structure de données respectant les normes du GATT et garantissant les fonctionnalités prévues du système. On met en place deux services primaires : le premier service, dénommé "Device Information", contient des informations sur le servomoteur conformément à la spécification du prédéfini par le Bluetooth SIG. Le second service est dédié aux échanges de données et est défini en tant que caractéristique personnalisée.

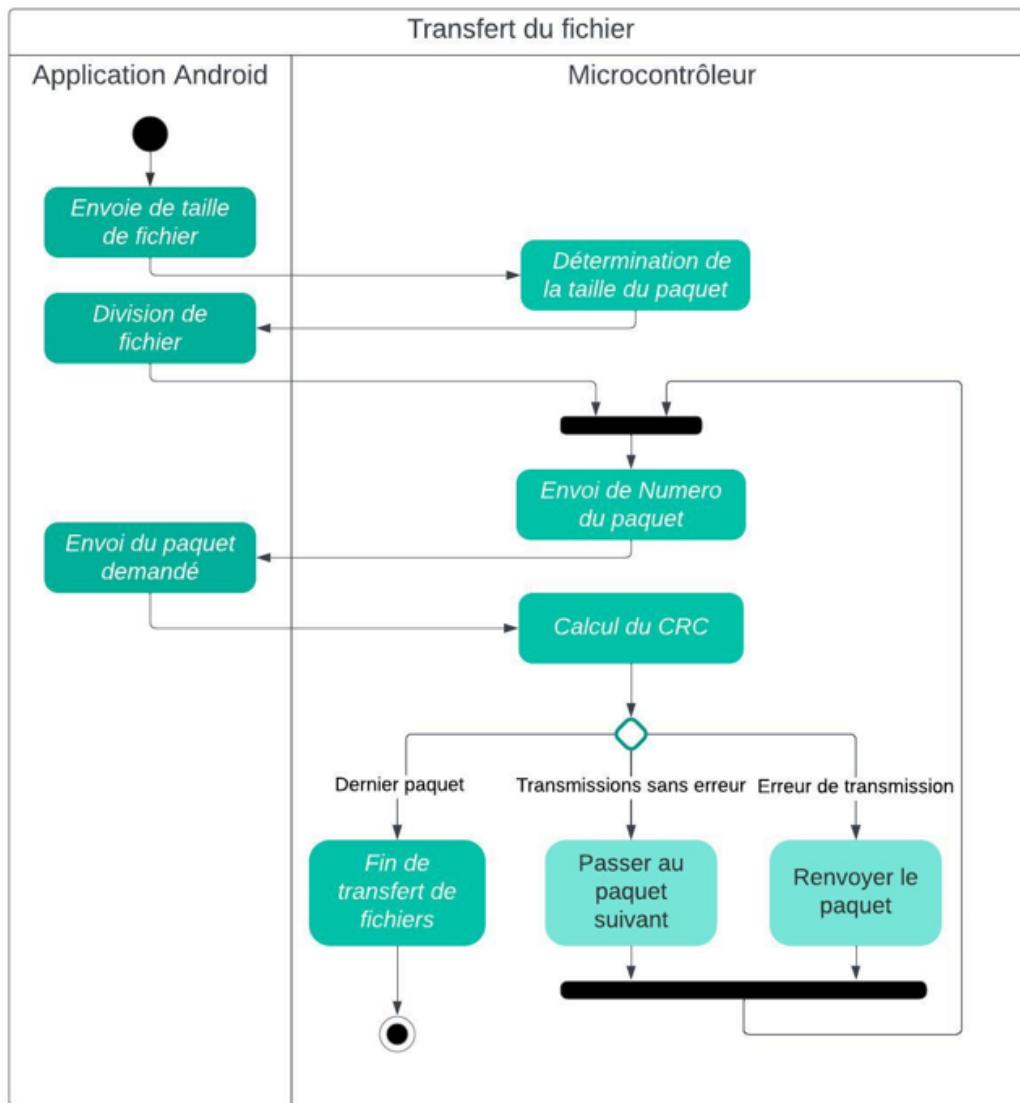


FIGURE 3.20 – Diagramme d’activité de transfert de fichier

3.7.1 Premier service

Service Name	UUID	Types of service
Device Information	0x180A	primary service

TABLE 3.4 – Premier service

3.7.2 Les caractéristiques du premier service

Dans les tableaux 3.4 et 3.5 de la page 30, on présente les déclarations du premier service ainsi que les caractéristiques incluses dans ce service, accompagnées de leurs propriétés respectives. Ces tableaux fournissent une vue détaillée des attributs du service, tels que leurs UUID, leurs noms et leurs propriétés spécifiques.

Characteristic Name	UUID	Properties
Model Number String	0x2A24	read
Serial Number String	0x2A25	read
Firmware Revision String	0x2A26	read
Hardware Revision String	0x2A27	read
Software Revision String	0x2A28	read
Manufacturer Name String	0x2A29	read

TABLE 3.5 – Les caractéristiques du premier service

3.7.3 Deuxième service

Dans les tableaux 3.6 et 3.7, on présente les déclarations du premier service ainsi que les caractéristiques incluses dans ce service, accompagnées de leurs propriétés respectives. Ces tableaux fournissent une vue détaillée des attributs du service, tels que leur UUID, leur type et leurs propriétés spécifiques.

Service Name	UUID	Types of service
Echange de données	3425652D-09D4-474E-B0AB-1DFRA1BB2F01	primary service

TABLE 3.6 – Deuxième service

Characteristic Name	UUID	Properties
Status servomoteur	AE564A8A91AC43B9 97AECAE45E2AA17C	read / notify
Commande servomoteur	8E2878F2DCA44E57 8D5B122D0B440A4C	write
Lecture données	F45CAF86F74E4A42 8C9D6CC120E678EE	read / write / notify
Ecriture données	FE90218BE2F34D0D BA8B7A18B4044955	read / write / notify
Transfert fichier	0E7C3DCD480B4F72 9ED9054A0776428F	read / write / notify

TABLE 3.7 – Les caractéristiques du deuxième service

3.8 Données UART

Afin de faciliter les échanges entre le microcontrôleur et le module Bluetooth, on définit les informations et les données strictement nécessaires qui sont incluses dans les trames UART. On prend en compte les messages UART suivants pour assurer une communication fluide et efficace entre les deux composants :

3.8.1 Commandes envoyées par microcontrôleur au module Bluetooth

Le microcontrôleur envoie quatre types essentiels de commandes :

Nom de la commande	"advertising" data (31 bytes)
--------------------	-------------------------------

TABLE 3.8 – Commande d'envoi de données "advertising"

1. La commande de "advertising", décrite dans la table 3.8, permet la transmission des données de "advertising" qui contribue à promouvoir la visibilité du système. Conformément au protocole GAP, elle contient 31 octets de données utiles pour l'"advertising". Cela permet aux dispositifs environnants de détecter et de se connecter au système une fois que le Bluetooth est activé.

Nom de la commande

TABLE 3.9 – Commande d'activation de la diffusion "advertising"

2. La commande d'activation de diffusion, illustrée dans la table 3.9, permet l'activation de la diffusion Bluetooth, permettant ainsi aux dispositifs environnants de détecter et de se connecter au système.

Nom de la commande	Handle	Longueur de données	Données
--------------------	--------	---------------------	---------

TABLE 3.10 – Commande d'envoi de données

3. La commande d'envoi de données, illustrée dans la table 3.10, permet au microcontrôleur de mettre à jour la valeur d'une caractéristique.

Nom de la commande	Handle	Longueur de données	Données
--------------------	--------	---------------------	---------

TABLE 3.11 – Notification du changement de donnée

4. La commande de notification, décrite dans la table 3.11 à la page 31, permet d'informier les appareils connectés d'une modification des données d'une caractéristique.

Par défaut, la taille maximale de la MTU (Maximum Transmission Unit) du Bluetooth LE de l'ESP32 est de 23 octets, mais elle peut être configurée pour atteindre 517 octets. Il est important de noter qu'il ne faut pas dépasser ces valeurs afin de garantir une compatibilité et un fonctionnement optimal du système.

3.8.2 Réponses renvoyées de l'ESP32 vers le microcontrôleur

Les messages retournés de l'ESP32 au microcontrôleur en réponse à l'envoi de commandes sont spécifiés dans les tables 3.12 3.13, et 3.14. Lorsque l'ESP32 démarre, il envoie le message "Ready" pour informer le microcontrôleur qu'il est disponible et prêt à accepter les commandes. À chaque réception d'une commande, l'ESP32 répond avec "OK" si l'opération a été effectuée avec succès, ou avec "Error" en cas d'erreur survenue lors du traitement de la commande.

Ready

TABLE 3.12 – Message de démarrage et disponibilité du module

OK

TABLE 3.13 – Traitement réussi de la commande

Error

TABLE 3.14 – Erreur de commande ou erreur survenue pendant l'exécution

3.8.3 Commande envoyées par module Bluetooth au microcontrôleur

Le module Bluetooth informe régulièrement le microcontrôleur des événements effectués par l'application. Ces événements comprennent notamment :

1. L'établissement d'une connexion est notifié au microcontrôleur((table 3.15 de la page 33). L'identifiant de connexion est utilisé pour représenter les appareils connectés, et il peut varier de 0 à 2 (ou jusqu'à 8).
2. Une action d'appariement SMP (table 3.16 de la page 33). L'indice de chiffrement indique si l'action est un appariement (0) ou la fin d'un appariement(1).

Nom de la commande	Id connexion	Adresse du périphérique
--------------------	--------------	-------------------------

TABLE 3.15 – Établissement d'une connexion

Nom de la commande	Id connexion	Indice de chiffrement
--------------------	--------------	-----------------------

TABLE 3.16 – L'appariement SMP est maintenant terminé

3. Une opération d'écriture depuis une connexion Bluetooth (table 3.17). Le "Handle" est un identifiant numérique utilisé pour référencer une caractéristique ou un descripteur.

Nom de la commande	Handle	Longueur de donnée	Données
--------------------	--------	--------------------	---------

TABLE 3.17 – Une opération d'écriture depuis une connexion Bluetooth

4. Une opération de lecture depuis une connexion Bluetooth (table 3.18).

Nom de la commande	Handle	Adresse du périphérique
--------------------	--------	-------------------------

TABLE 3.18 – Une opération de lecture depuis une connexion Bluetooth

5. La Rupture de Connexion avec un Appareil (table 3.19).

Déconnexion

TABLE 3.19 – Une opération de déconnexion

Le microcontrôleur est ainsi informé en temps réel de ces événements, ce qui lui permet de réagir de manière appropriée et de maintenir un fonctionnement optimal du système.

3.9 Conclusion

En conclusion, la conception du système permet de définir la structure de données nécessaire pour assurer un transfert de données fiable et performant. On présente les notions techniques, la structure de données dans la trame UART, ainsi que les diagrammes de séquence et les diagrammes d'états-transitions pour mieux comprendre le comportement du système. Ce chapitre pose les bases nécessaires pour la réalisation du système, qui sera abordée dans le chapitre suivant.

CHAPITRE 4 : RÉALISATION DU SYSTÈME

4.1 Introduction

Le dernier chapitre de ce rapport se concentre sur l'implémentation du projet. On décrit en détail l'environnement de développement, en présentant les outils et les technologies utilisés. On effectue différents tests pour vérifier le bon fonctionnement du système, en mettant en évidence les interactions entre les différents acteurs impliqués. De plus, On aborde deux approches de développement distinctes, en présentant les spécificités et les performances de chacune d'entre elles.

4.2 Environnement de développement

4.2.1 Environnement matériel

Afin de développer, tester et implémenter le système, on présente les caractéristiques clés de l'environnement matériel utilisé.

L'ordinateur utilisé

PC	DELL Inspiron
RAM	16 GB
GPU	NVIDIA GTX 1050 4000 MB
Disque dur	1 TO HDD
Processeur	2.5 GHz, intel i5 7ème génération
Système d'exploitation	Windows 10

TABLE 4.20 – Caractéristique d'ordinateur

La carte de développement utilisée

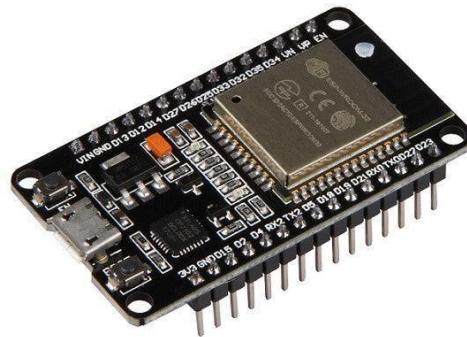


FIGURE 4.21 – ESP-WROOM 32

Caractéristique	Spécifications
Bandwidth	72 MHz
Data Rate	150 Mbps
Interface	Ethernet, I2C, I2S, SPI, UART
Max Frequency	2.484 GHz
Max Operating Temperature	85 °C
Max Supply Voltage	3.6 V
Min Operating Temperature	-40 °C
Min Supply Voltage	3 V
Nominal Supply Current	500 mA
Number of ADC Channels	16
Number of GPIO	32

TABLE 4.21 – Caractéristiques techniques de la carte de développement utilisée

Le microcontrôleur utilisé

Le STM32 G474 est un microcontrôleur appartenant à la famille STM32G4 de STMicroelectronics, doté d'une architecture ARM Cortex-M4 et capable de fonctionner à une fréquence d'horloge maximale de 170 MHz. Il est équipé d'une mémoire flash intégrée de 512 Ko et dispose de multiples interfaces de communication telles que UART, SPI, I2C et CAN, ainsi qu'une interface USB Full-Speed et une interface de débogage JTAG/SWD. De plus, il est pourvu de divers périphériques de contrôle de puissance, notamment des régulateurs de tension, des temporiseurs, des comparateurs et des amplificateurs opérationnels.



FIGURE 4.22 – Microcontrôleur STM32

Convertisseur de données série vers USB

Waveshare USB to TTL est un câble qui permet de connecter des périphériques série tels que des microcontrôleurs, des capteurs, des modules sans fil à un ordinateur via le port USB. Il est équipé d'un convertisseur de niveau de tension qui convertit les signaux de tension du périphérique série en signaux de tension USB.

Le câble est doté d'un connecteur USB d'un côté et d'un connecteur TTL 6 broches de l'autre côté, qui peut être directement branché sur le port série d'un microcontrôleur ou d'un autre périphérique série. Le câble Waveshare USB to TTL est souvent utilisé pour programmer des microcontrôleurs, déboguer des programmes, et pour la communication série avec des périphériques tels que des capteurs ou des afficheurs.



FIGURE 4.23 – Waveshare USB to TTL

Interface Homme-Machine du servomoteur

L'interface Homme-Machine (IHM) de ce projet, présenté par la figure 4.24 de la page 37, est conçue pour permettre à l'utilisateur de commander un servomoteur et de récupérer les données sur sa configuration et son état. Elle est principalement composée d'un microcontrôleur STM32G474 et d'un module Bluetooth ESP-WROOM 32.

Le microcontrôleur joue un rôle central dans l'IHM en assurant plusieurs fonctionnalités essentielles. Tout d'abord, il est responsable de la commande du servomoteur en envoyant les signaux appropriés pour contrôler ses réglages de synchronisation, sa position et d'autres paramètres spécifiques. De plus, le microcontrôleur est également chargé également configurer des alarmes pour surveiller les conditions de fonctionnement du servomoteur.



FIGURE 4.24 – Interface Homme-Machine du servomoteur

En plus de la gestion du servomoteur, le microcontrôleur assure la connectivité sans fil grâce au module Bluetooth ESP-WROOM 32. Ce module permet une communication bidirectionnelle entre l'IHM et une application Android associée. Par le biais de cette connexion sans fil, l'application Android peut envoyer des commandes au servomoteur, telles que des instructions de mouvement ou de configuration, et recevoir les données de retour du servomoteur, telles que les informations d'état et les mesures.

Le module Bluetooth ESP-WROOM 32 agit comme une interface entre le microcontrôleur et l'application Android. Il utilise le protocole Bluetooth Low Energy (BLE) pour la transmission des données sans fil et l'interface UART pour les données séries. Grâce à cette combinaison, l'IHM peut établir une connexion stable et fiable avec l'application Android, facilitant ainsi le contrôle du servomoteur et l'échange d'informations nécessaires.

4.2.2 Environnement logiciel

Suite au choix de carte de développement ESP32, on utilise plusieurs logiciels et outils de développement qui permettent de programmer la ESP32 et la STM32. Ces logiciels servent d'interfaces utilisateur permettant de programmer, de déboguer, d'éditer, de tester et de naviguer à travers la solution.

Espressif-IDE est un environnement de développement intégré (IDE) basé sur Eclipse CDT pour le développement d'applications IoT utilisant le framework ESP-IDF. Il s'agit d'un IDE autonome et personnalisé conçu spécifiquement pour ESP-IDF. Espressif-IDE est livré avec les plugins IDF Eclipse, les plugins Eclipse CDT essentiels et d'autres plugins.

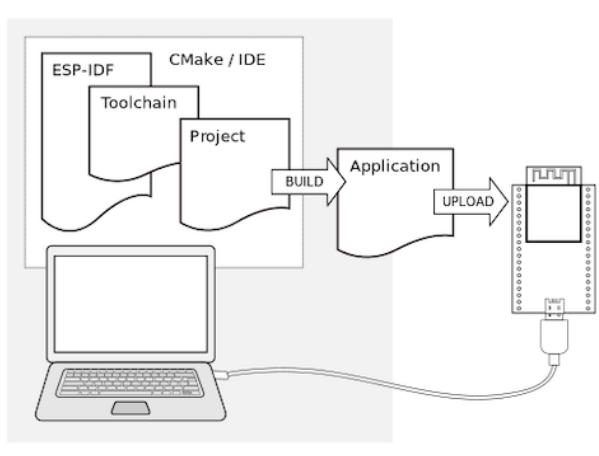


FIGURE 4.25 – Espressif IoT Development Framework

Espressif IDF (IoT Development Framework) est un ensemble d'outils de développement logiciel (SDK) open source pour les microcontrôleurs ESP32 et ESP8266 de la société Espressif Systems. IDF fournit une bibliothèque de fonctions pour accéder aux périphériques matériels, ainsi qu'un système d'exploitation temps réel (RTOS) pour gérer les tâches et les interruptions. IDF prend en charge plusieurs langages de programmation, notamment le langage C et le langage C++, et permet aux développeurs de créer des applications IoT complexes pour diverses applications.

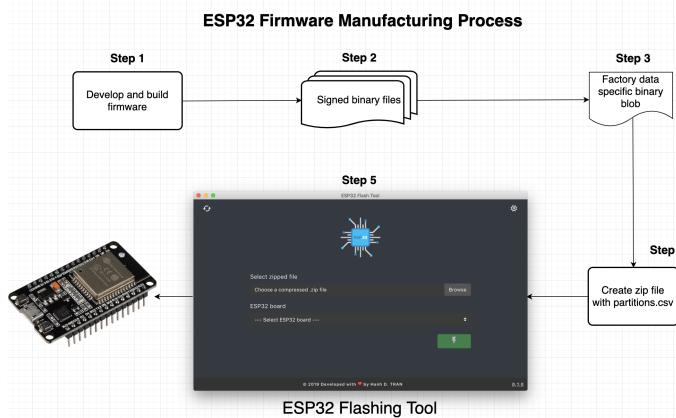


FIGURE 4.26 – Flash download tool

Flash Download Tool (FDT), ou Outil de Téléchargement de Flash en français, est un programme fourni avec les kits de développement par Espressif qui permet de programmer des microcontrôleurs. Il permet notamment de flasher le firmware et les données dans la mémoire flash du microcontrôleur. Simplification des paramètres, une interface utilisateur claire et une focalisation sur le processus de fabrication de masse où le développement et la production sont des étapes séparées.

**IAR Embedded Workbench**

FIGURE 4.27 – IAR Embedded Workbench

IAR est un environnement de développement intégré (IDE) pour les microcontrôleurs, qui permet aux développeurs de programmer, de déboguer et de tester des applications pour des architectures de microcontrôleurs courantes telles que ARM, MSP430 et AVR.

L'IDE IAR offre un ensemble complet d'outils de développement, notamment un éditeur de texte, un compilateur, un débogueur, un traceur de code, un profilage de performance et une analyseur de pile. Il prend également en charge les langages de programmation courants tels que C et C++, ainsi que l'assembleur.

**Docklight**

FIGURE 4.28 – Docklight

Docklight est un logiciel de test et de débogage de communication série. Il permet aux développeurs et aux ingénieurs de tester, surveiller et déboguer les communications série entre les appareils connectés, tels que les périphériques IoT, les équipements de mesure, les capteurs, les microcontrôleurs, les modems, etc. Docklight fournit une interface utilisateur graphique conviviale pour envoyer et recevoir des données série, ainsi que des outils pour analyser et dépanner les problèmes de communication. Il peut également être utilisé pour automatiser des tests et pour créer des scripts de test personnalisés.

Notepad++ est un éditeur de texte open source pour Windows qui prend en charge de nombreuses langues de programmation. Il est principalement utilisé par les développeurs pour écrire et modifier du code source, mais peut également être utilisé pour éditer des fichiers texte simples.

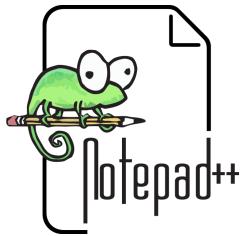


FIGURE 4.29 – Notepad++

NRF Connect est une application mobile développée par Nordic Semiconductor pour faciliter le développement, le test et la gestion des dispositifs IoT basés sur la technologie Bluetooth Low Energy (BLE). Elle offre une interface conviviale permettant de scanner, se connecter et interagir avec les périphériques BLE, ainsi que de surveiller les paquets de données, effectuer des tests de performance. NRF Connect est un outil essentiel pour les développeurs et les ingénieurs travaillant sur des projets BLE.



FIGURE 4.30 – NRF Connect

4.3 Première approche de mise en œuvre

Pour la mise en œuvre initiale, les fonctionnalités requises ont été développées en utilisant les commandes AT. Dans cette partie, on décrit la méthodologie de développement et on effectue des opérations d'échange de données. Cela permet d'obtenir une représentation graphique claire des données transmises entre l'ESP32, le microcontrôleur et l'application Android, ainsi que des différentes opérations effectuées.

Dans cette approche d'implémentation, on utilise le micrologiciel développé par Espressif, connu sous le nom de "ESP-AT V2.4". Ce micrologiciel configure la ESP32 en tant que périphérique esclave ("slave"), ce qui lui permet de répondre aux commandes et aux requêtes émises par le microcontrôleur configuré en tant que périphérique maître ("host"). La communication entre la ESP32 et le microcontrôleur se fait via l'UART, où les commandes AT sont transmises pour configurer le BLE et échanger des données.

Espressif propose une vaste liste de commandes prédéfinies qui permettent de configurer différents paramètres et de gérer les interactions entre le maître et l'esclave, afin d'intégrer une connectivité BLE au maître.

4.3.1 Configuration Matériel

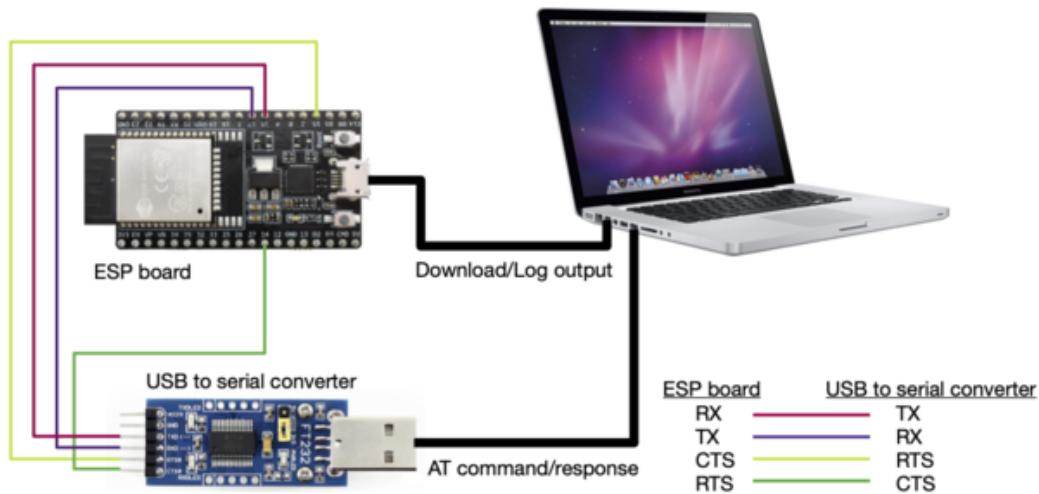


FIGURE 4.31 – Connexion matérielle de la ESP32

La figure 4.31 représente le schéma de connexion matérielle nécessaire pour télécharger le micrologiciel sur l'ESP32, envoyer des commandes AT et recevoir les réponses correspondantes. Dans ce schéma, la carte ESP32 agit en tant qu'esclave, tandis que le PC agit en tant que maître. La connexion entre l'ESP32 et le PC est établie à l'aide d'un câble USB, ce qui permet le téléchargement du micrologiciel et l'affichage des sorties. Un convertisseur USB vers port série est utilisé pour convertir les signaux entre l'USB et les signaux TTL. Les fils de liaison sont utilisés pour la transmission des commandes AT et des réponses.

4.3.2 Gestion de la connectivité BLE

Dans un premier temps, le microcontrôleur envoie une séquence de commandes pour configurer le module en tant que serveur GATT. Cela comprend la création des services nécessaires, la configuration des paramètres de sécurité et de cryptage, l'envoi des données "advertising" (publicité), l'activation des services et l'activation de la diffusion de Bluetooth Low Energy (BLE). La figure 4.32 de la page 42, illustre cette séquence, mettant en évidence les messages de validation échangés durant cette étape.

```

23/05/2023 11:16:44.877 [TX] - AT+BLEINIT=2<CR><LF>
23/05/2023 11:16:45.929 [RX] - <CR><LF>
OK<CR><LF>

23/05/2023 11:16:46.874 [TX] - AT+BLEGATTSSRVCRE<CR><LF>
23/05/2023 11:16:46.956 [RX] - <CR><LF>
OK<CR><LF>

23/05/2023 11:16:49.034 [TX] - AT+BLESECPARAM=1,3,16,3,3,0<CR><LF>
23/05/2023 11:16:49.043 [RX] - <CR><LF>
OK<CR><LF>

23/05/2023 11:16:52.685 [TX] - AT+BLEADVDATA="02010609094553503332424C45"<CR><LF>
23/05/2023 11:16:52.722 [RX] - <CR><LF>
OK<CR><LF>

23/05/2023 11:16:56.274 [TX] - AT+BLEGATTSSRVSTART<CR><LF>
23/05/2023 11:16:56.289 [RX] - <CR><LF>
OK<CR><LF>

23/05/2023 11:16:58.084 [TX] - AT+BLEADVSTART<CR><LF>
23/05/2023 11:16:58.095 [RX] - <CR><LF>
OK<CR><LF>

```

FIGURE 4.32 – Séquence de commandes pour configurer BLE

Les commandes et leurs explications :

"AT+BLEINIT=2" : Configurer l'ESP32 en tant que serveur GATT.

"AT+BLEGATTSSRVCRE" : Créer les services.

"AT+BLESECPARAM=1,3,16,3,3,0" : Configurer les paramètres de sécurité, y compris les autorisations d'appariement, la taille des clés d'encryption, les clés initiales et les clés de réponse, ainsi que les options de sécurité.

"AT+BLEADVDATA=" : Envoyer les données de "advertising" de taille maximale de 31 octets.

"AT+BLEGATSSRVCSTART" : Lancer les services.

"AT+BLEADVSTART" : Démarrer la diffusion BLE.

Cette étape est cruciale et ne se produit qu'une seule fois à chaque activation du module, permettant l'envoi des paquets advertising. Elle est illustrée dans la figure 4.33 de la page 43, et elle permet aux applications android de détecter le réseau avec la configuration et le nom spécifié, d'établir une connexion et de réaliser l'appairage avec le réseau Bluetooth Low Energy (BLE)

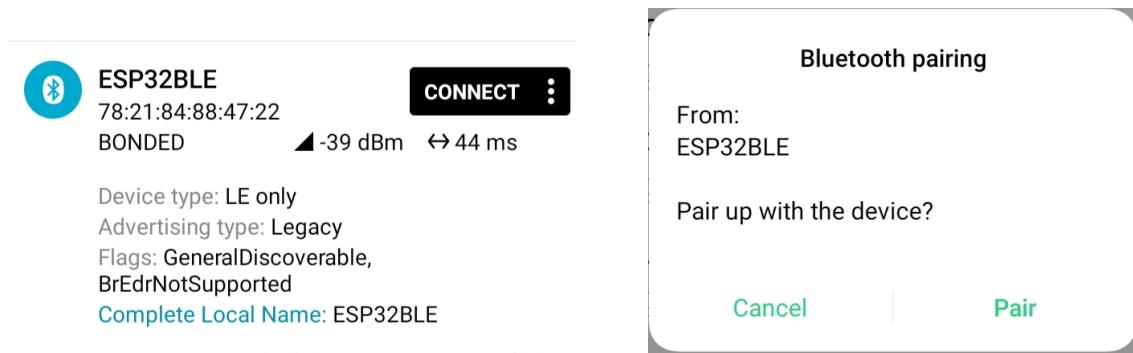


FIGURE 4.33 – Disponibilité du réseau et permission de scan et appariement

Une fois connecté, le microcontrôleur reçoit le message "+BLECONN" qui lui permet de connaître l'adresse de l'appareil connecté, ainsi que "+BLECONNPARAM" pour les paramètres de connexion. En réponse, le microcontrôleur envoie la commande "AT+BLEENCRSP=0,1" pour demander un échange de clés d'encryption avec l'appareil connecté. Une fois l'appairage réussi, le message "+BLEAUTHCMPL" est envoyé au microcontrôleur.

BONDED	ADVERTISER	ESP32BLE 78:21:84:88:47:22	X
<u>CONNECTED</u>	<u>CLIENT</u>	SERVER	:
Generic Attribute UUID: 0x1801 PRIMARY SERVICE			
Generic Access UUID: 0x1800 PRIMARY SERVICE			
Device Information UUID: 0x180A PRIMARY SERVICE			
Unknown Service UUID: 3425652d-09f4-474e-b0ab-1dfa1bb2f01 PRIMARY SERVICE			

```
+BLECONN:0,"7a:9b:c2:bf:ab:7e"<CR><LF>
+BLESECREQ:0<CR><LF>
+BLECONNPARAM:0,0,0,6,0,500<CR><LF>
+BLECONNPARAM:0,0,0,39,0,500<CR><LF>

23/05/2023 11:19:41.315 [TX] - AT+BLEENCRSP=0,1<CR><LF>

23/05/2023 11:19:41.329 [RX] - <CR><LF>
OK<CR><LF>
+BLESECKEYTYPE:0,16<CR><LF>
+BLESECKEYTYPE:0,32<CR><LF>
+BLESECKEYTYPE:0,1<CR><LF>
+BLESECKEYTYPE:0,2<CR><LF>
+BLEAUTHCMPL:0,0<CR><LF>
```

FIGURE 4.34 – Connexion et apparition des services et caractéristiques

À ce stade, il est possible de consulter directement les services disponibles et d'effectuer des opérations d'échange de données. La figure 4.34 illustre l'apparition des services spécifiés dans la conception, ainsi que les messages envoyés au microcontrôleur indiquant l'établissement de la connexion. Cette étape permet d'accéder aux fonctionnalités du système et d'interagir avec les services disponibles via la connexion Bluetooth établie.

4.3.3 Envoi de données de l'application vers microcontrôleur pour commande du servomoteur

Après la connexion, on effectue un test d'envoi de données pour vérifier le fonctionnement, qui est décrit dans la page suivante.

On procéde à un test d'envoi de 10 octets de données depuis l'application, à savoir "0x00 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99" en format hexadécimal. Conformément à la figure 4.35, ces données sont envoyées vers les quatre caractéristiques qui ont la propriété WRITE, ce qui signifie que l'application est autorisée à envoyer des données.

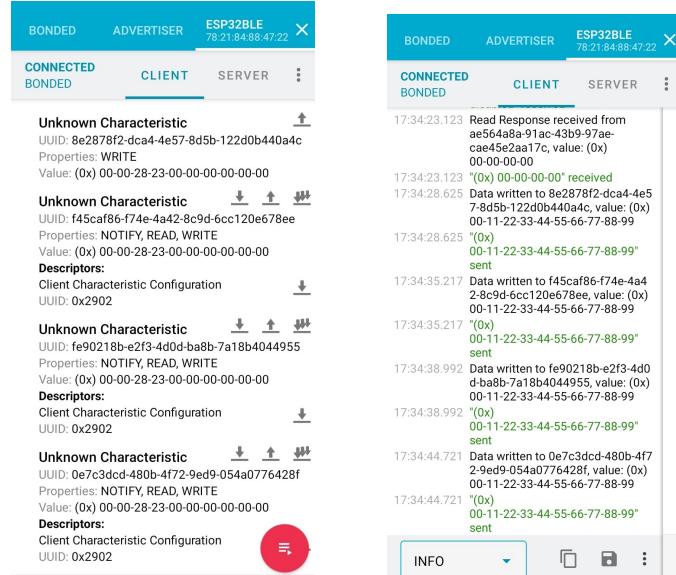


FIGURE 4.35 – Séquence de commandes de l'envoi de données par l'application

La figure 4.36 met en évidence les messages UART envoyés au microcontrôleur pour le notifier des changements. Ces messages sont présentés à la fois en ASCII et en format hexadécimal. Cette étape garantit le transfert correct des données de l'application vers le microcontrôleur, en utilisant le format hexadécimal pour leur représentation. Par exemple, le message "+WRITE :0,3,2,0,10,.." indique que l'opération a été réalisée par la connexion d'ID 0, sur le service numéro 3, la caractéristique numéro 2, sans descripteur, avec une longueur de données de 10. Les données envoyées sont présentes après la dernière virgule.

```
+WRITE:0,3,2,,10,<NUL>||<DC1>|| "3DUFw^~<CR><LF>
+WRITE:0,3,3,,10,<NUL>||<DC1>|| "3DUFw^~<CR><LF>
+WRITE:0,3,4,,10,<NUL>||<DC1>|| "3DUFw^~<CR><LF>
+WRITE:0,3,5,,10,<NUL>||<DC1>|| "3DUFw^~<CR><LF>

2B 57 52 49 54 45 3A 30 2C 33 2C 32 2C 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
2B 57 52 49 54 45 3A 30 2C 33 2C 33 2C 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
2B 57 52 49 54 45 3A 30 2C 33 2C 34 2C 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
2B 57 52 49 54 45 3A 30 2C 33 2C 35 2C 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
```

FIGURE 4.36 – Réception des trames de données par le microcontrôleur

4.3.4 Envoi de données de microcontrôleur vers l'application et récupération des données du servomoteur

On effectue un test d'envoi de 10 octets de données depuis le microcontrôleur, à savoir "0x30 0x30 0x31 0x31 0x32 0x32 0x33 0x33 0x34 0x34" en format hexadécimal, on observe les résultats suivants. Conformément à la figure 4.37, les commandes UART sont utilisées pour envoyer ces données vers les quatre caractéristiques qui ont la propriété READ et NOTIFY, ce qui autorise l'application à lire ces données.

Envoi de données de microcontrôleur vers l'application et récupération des données du servomoteur

```

24/05/2023 01:42:10.256 [TX] - AT+BLEGATTSSETATTR=3,1,,10<CR><LF>
24/05/2023 01:42:10.273 [RX] - >
24/05/2023 01:42:10.675 [TX] - 0011223344
24/05/2023 01:42:13.172 [RX] - <CR><LF>
OK<CR><LF>

24/05/2023 01:42:15.083 [TX] - AT+BLEGATTSNTFY=0,3,1,10<CR><LF>
24/05/2023 01:42:15.093 [RX] - >
24/05/2023 01:42:19.739 [TX] - 0011223344
24/05/2023 01:42:22.048 [RX] - <CR><LF>
OK<CR><LF>

24/05/2023 01:42:23.909 [TX] - AT+BLEGATTSSETATTR=3,3,,10<CR><LF>
24/05/2023 01:42:23.920 [RX] - >
24/05/2023 01:42:24.365 [TX] - 0011223344
24/05/2023 01:42:26.661 [RX] - <CR><LF>
OK<CR><LF>

24/05/2023 01:42:28.071 [TX] - AT+BLEGATTSNTFY=0,3,3,10<CR><LF>
24/05/2023 01:42:28.085 [RX] - >
24/05/2023 01:42:28.502 [TX] - 0011223344
24/05/2023 01:42:30.515 [RX] - <CR><LF>
OK<CR><LF>

```

FIGURE 4.37 – Séquence d'envoi de données ASCII par le microcontrôleur

Par exemple, l'envoi de la commande "AT+BLEGATTSSETATTR=3,1,,10" spécifie l'envoi de 10 octets à la première caractéristique du troisième service. Cette commande permet de mettre à jour les données de la caractéristique spécifiée.

De plus, l'envoi de la commande "AT+BLEGATTSNTFY=0,3,1,10" permet de notifier l'appareil avec l'ID de connexion 0 du changement sur la caractéristique spécifiée du troisième service. Cette commande indique à l'appareil connecté qu'il y a de nouvelles données disponibles à récupérer.

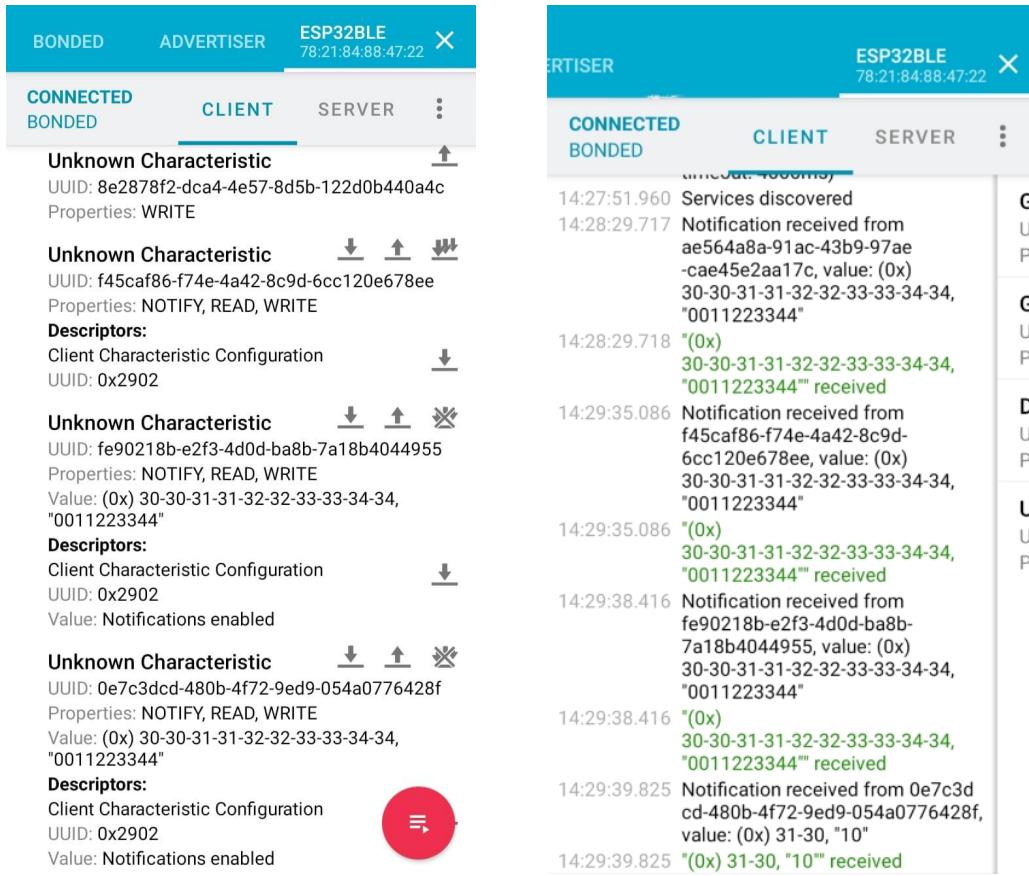


FIGURE 4.38 – Réception des trames de données par l’application

La figure 4.38, illustre la réception d'une notification indiquant l'arrivée de ces données, ainsi que l'apparition de ces données dans les caractéristiques correspondantes. De plus, des informations de réception sont également fournies, permettant de suivre et d'analyser les données reçues. Cette étape permet ainsi à l'application d'accéder et de lire les données envoyées par le microcontrôleur, en utilisant les caractéristiques qui possèdent la propriété READ et NOTIFY.

4.3.5 Mise à jour du firmware du microcontrôleur

On réalise un envoi de fichier de taille 90 ko pour tester le processus de transfert de fichiers. Pour ce faire, on utilise la 5ème caractéristique spécifiquement réservée à la mise à jour du firmware du microcontrôleur. La figure 4.39 de la page 47 présente la première commande envoyée par l'application pour initialiser le transfert, tandis que la figure 4.40 de la page 47, illustre la dernière commande marquant la fin du transfert. Ces deux figures mettent en évidence respectivement le début et la fin du processus de transfert de fichier, ce qui permet de mesurer le temps nécessaire pour finaliser cette opération.

```

2023-05-26 18:48:48.721 [COM6] - +WRITE:0,3,5,,5,* ^·
2023-05-26 18:48:49.281 [COM6] - >
OK
>
OK

2023-05-26 18:48:49.297 [COM4] - AT+BLEGATTSSETATTR=3,5,,5
INTOKAT+BLEGATTSNTFY=0,3,5,5
INTOK
2023-05-26 18:48:49.361 [COM6] - +WRITE:0,3,5,,20,0* ý*** @... @...!@...
>
2023-05-26 18:48:49.408 [COM4] - AT+BLEGATTSSETATTR=3,5,,4
* AT+BLEGATTSNTFY=0,3,5,4

2023-05-26 18:48:49.408 [COM6] -
OK
>
OK

```

FIGURE 4.39 – La première commande pour initialiser le transfert

```

2023-05-26 19:00:17.064 [COM4] - AT+BLEGATTSSETATTR=3,5,,4
2023-05-26 19:00:17.070 [COM6] - >
OK

2023-05-26 19:00:17.083 [COM4] - * AT+BLEGATTSNTFY=0,3,5,4
*
2023-05-26 19:00:17.086 [COM6] - >
OK

2023-05-26 19:00:17.182 [COM6] - +WRITE:0,3,5,,20,     ýiÝ į »±~,z»Û!
2023-05-26 19:00:17.288 [COM4] - AT+BLEGATTSSETATTR=3,5,,4
* AT+BLEGATTSNTFY=0,3,5,4
*
2023-05-26 19:00:17.307 [COM6] - >
OK
>
OK

2023-05-26 19:00:21.451 [COM6] - +WRITE:0,3,5,,3,END

```

FIGURE 4.40 – la dernière commande pour marquer la fin du transfert

4.3.6 Limitation de la première approche de mise en œuvre

Le transfert de fichier s'effectue par un échange bidirectionnel de données en utilisant la 5ème caractéristique du deuxième service. Le diagramme d'activité présenté dans la figure 3.20 de la page 29 illustre les étapes du transfert de fichier de manière claire.

La figure 4.39 présente la première trame envoyée pour initialiser le transfert de fichiers, tandis que la figure 4.40 montre les dernières trames envoyées. Le processus de transfert complet nécessite 11 minutes. Après avoir effectué plusieurs tests, on a constaté que le transfert prend un temps considérable, pouvant atteindre jusqu'à 20 minutes en cas d'échec.

Cette durée de transfert est considérée comme longue, car elle impose une attente prolongée aux techniciens à chaque transfert. Afin de remédier à cette situation, on recherche une méthode alternative permettant de réduire significativement le temps de transfert de fichiers, tout en conservant les fonctionnalités essentielles requises.

L'objectif est de trouver une approche plus efficace qui permette des transferts rapides et fiables, afin d'optimiser le processus et d'améliorer l'efficacité globale de l'opération.

On constate que lors de chaque envoi de données du microcontrôleur vers le module de communication (changement de valeur de caractéristique), le microcontrôleur effectue cette opération en plusieurs étapes séquentielles :

1. Envoi de la commande "AT+BLEGATTSETATTR" pour spécifier la caractéristique à modifier.
2. Envoi des données correspondantes à cette caractéristique.
3. Envoi de "AT+BLEGATTSNTFY" pour spécifier la caractéristique à notifier.
4. Envoi des données à nouveau pour notifier les appareils connectés.

À chaque étape, le microcontrôleur réalise une interaction Bluetooth avec le module de communication, attend la confirmation de chaque commande et respecte un délai de 200 ms avant de passer à l'étape suivante.

Le temps réservé pour la communication Bluetooth Low Energy (BLE) est fixe et ne peut pas être modifié, car cela affecterait le fonctionnement fondamental du microcontrôleur. Cependant, cette approche présente une limitation en termes de temps, car chaque étape nécessite une interaction séparée, ce qui entraîne un temps de transfert plus long. Les interactions et les commandes sont prédéfinies et fixes dans le Firmware AT command d'Espressif. Le code source de ce firmware n'est pas open source, ce qui signifie qu'il ne peut pas être modifié.

Pour résoudre cette problématique, on propose une deuxième approche de réalisation.

4.4 Deuxième approche de mise en œuvre

Dans la deuxième approche de développement, on utilise les API de l'UART et du Bluetooth pour programmer la carte ESP. On effectue les configurations matérielles nécessaires, en conservant les mêmes connexions, telles que les broches et le port UART. De plus, on effectue les configurations logicielles nécessaires pour gérer les interactions Bluetooth et les commandes personnalisées, en utilisant le langage de programmation C. Cette approche permet une plus grande flexibilité et un contrôle plus précis sur les fonctionnalités et les échanges de données de la carte ESP.

4.4.1 Gestion de la connectivité BLE

```
23/05/2023 12:23:37.172 [TX] - ADVDATA,02010609094553503332424C45<CR><LF>
23/05/2023 12:23:37.189 [RX] - OK<CR><LF>
23/05/2023 12:23:38.779 [TX] - STARTADV
23/05/2023 12:23:39.226 [RX] - OK<CR><LF>
+BLECONN:0,"6b:7b:f9:05:df:ca"<CR><LF>
+BLEAUTHCMPL:0,0<CR><LF>
```

FIGURE 4.41 – Séquence de commandes pour configurer BLE

Dans un premier temps, le microcontrôleur envoie uniquement deux commandes pour démarrer le processus. Comme illustré dans la figure 4.41 , une commande "ADVDATA" contient les données "advertising".

L'autre commande "STARTADV" active la diffusion Bluetooth Low Energy (BLE). Les autres configurations, telles que la création des services requis, la configuration du module en tant que serveur GATT et la mise en place des paramètres de sécurité et de cryptage, sont réalisées à l'aide des API appropriées.

4.4.2 Envoi de données de l'application vers le microcontrôleur pour la commande du servomoteur

En maintenant les mêmes commandes et le même format de données, on a effectué un test d'envoi de 10 octets de données de l'application vers le microcontrôleur, représentés par la séquence "00 11 22 33 44 55 66 77 88 99". La figure 4.43 illustre ce test d'envoi, tandis que la figure 4.42 montre la réception de ces données par le microcontrôleur.

```
+WRITE:59,10,<NUL>I <DC1>I "3DUFw^"<CR><LF>
+WRITE:61,10,<NUL>I <DC1>I "3DUFw^"<CR><LF>
+WRITE:64,10,<NUL>I <DC1>I "3DUFw^"<CR><LF>
+WRITE:67,10,<NUL>I <DC1>I "3DUFw^"<CR><LF>

2B 57 52 49 54 45 3A 35 39 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
2B 57 52 49 54 45 3A 36 31 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
2B 57 52 49 54 45 3A 36 34 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
2B 57 52 49 54 45 3A 36 37 2C 31 30 2C 00 11 22 33 44 55 66 77 88 99 0D 0A
```

FIGURE 4.42 – Les trames reçues par le microcontrôleur de

On a conservé la même syntaxe de message envoyé du module au microcontrôleur afin de garantir la compatibilité entre les deux approches, à l'exception du "handle" qui a été remplacé par une seule valeur numérique pour simplifier la représentation. Les figures 4.43 et 4.42 mettent en évidence le processus d'envoi et de réception des données entre l'application et le microcontrôleur, démontrant ainsi la communication réussie.

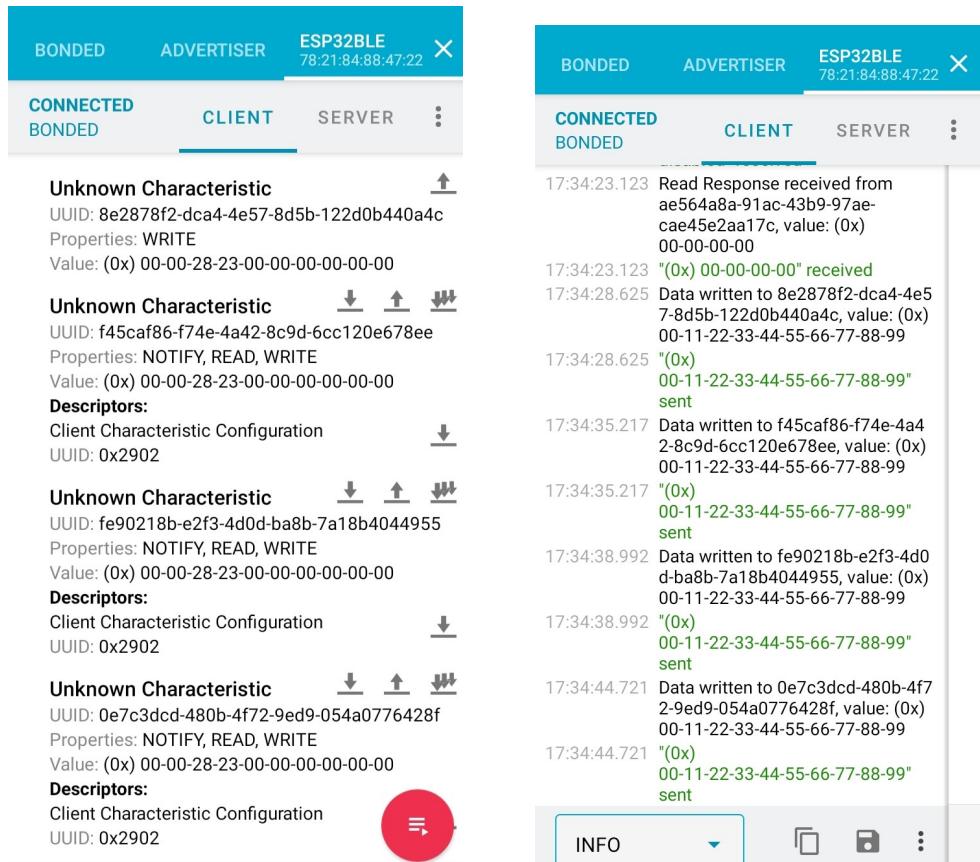


FIGURE 4.43 – Séquence de commandes de l'envoi de données par l'application

4.4.3 Envoi de données de microcontrôleur vers l'application et récupération des données du servomoteur

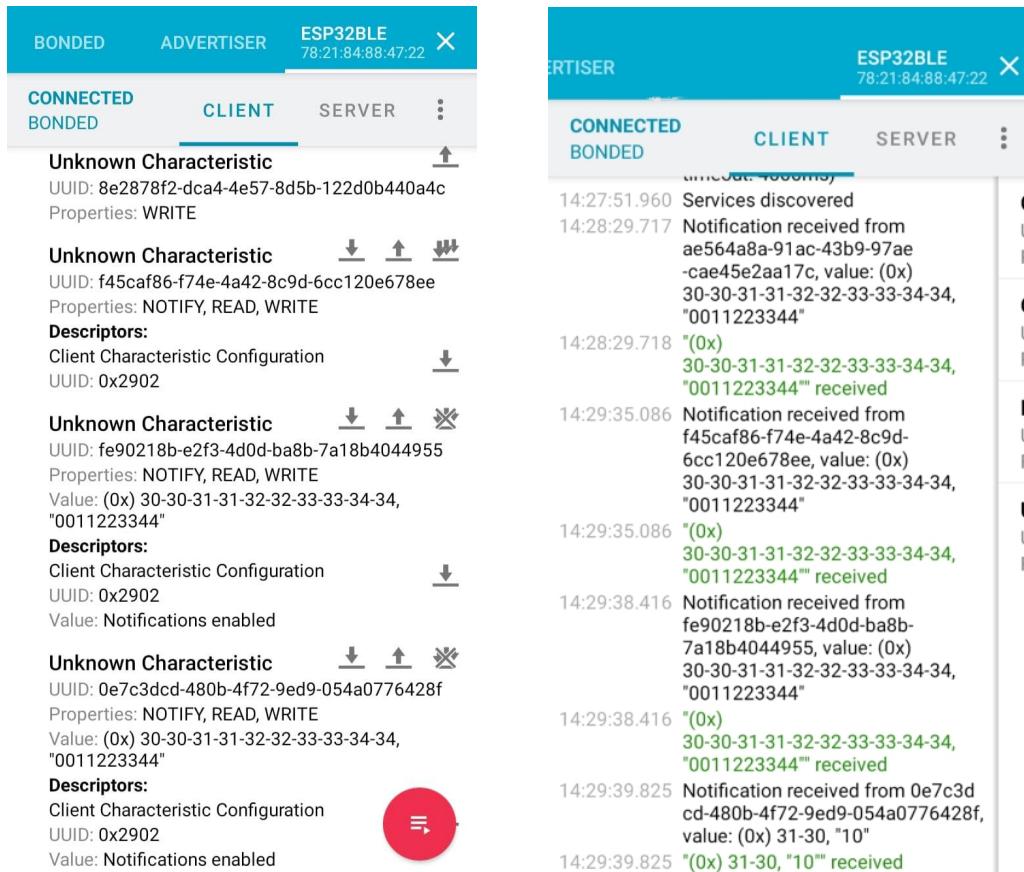
Pour l'envoi de données du microcontrôleur vers le module de communication, on met en place une seule commande appelée "SENDDATA". Cette commande permet d'envoyer à la fois la commande elle-même, les paramètres (Handle et longueur des données) ainsi que les données. Ainsi, on peut envoyer les données aux quatre caractéristiques en une seule commande seulement pour chacune, ce qui simplifie le processus. La figure 4.44 de la page 51, illustre l'envoi de données aux quatre caractéristiques à l'aide de cette commande unique. La figure 4.45 confirme la réception de ces données par l'application, ce qui valide le bon fonctionnement de la communication.

```

23/05/2023 12:37:13.729 [TX] - SENDDATA,56,10,00112233445566778899<CR><LF>
23/05/2023 12:37:13.754 [RX] - OK<CR><LF>
23/05/2023 12:37:15.730 [TX] - SENDDATA,61,10,00112233445566778899<CR><LF>
23/05/2023 12:37:15.754 [RX] - OK<CR><LF>
23/05/2023 12:37:17.205 [TX] - SENDDATA,64,10,00112233445566778899<CR><LF>
23/05/2023 12:37:17.224 [RX] - OK<CR><LF>
23/05/2023 12:37:18.454 [TX] - SENDDATA,67,2,10,00112233445566778899<CR><LF>
23/05/2023 12:37:18.474 [RX] - OK<CR><LF>

```

FIGURE 4.44 – Séquence de commandes de l'envoi de données par le microcontrôleur



ESP32BLE	
Advertiser	78:21:84:88:47:22
CONNECTED	BONDED
CLIENT	SERVER
...	...
14:27:51.960	Services discovered
14:28:29.717	Notification received from ae564a8a-91ac-43b9-97ae -cae45e2aa17c, value: (0x) 30-30-31-31-32-32-33-33-34-34, "0011223344"
14:28:29.718	"(0x) 30-30-31-31-32-32-33-33-34-34, "0011223344"" received
14:29:35.086	Notification received from f45caf86-f74e-4a42-8c9d-6cc120e678ee, value: (0x) 30-30-31-31-32-32-33-33-34-34, "0011223344"
14:29:35.086	"(0x) 30-30-31-31-32-32-33-33-34-34, "0011223344"" received
14:29:38.416	Notification received from fe90218b-e2f3-4d0d-ba8b-7a18b4044955, value: (0x) 30-30-31-31-32-32-33-33-34-34, "0011223344"
14:29:38.416	"(0x) 30-30-31-31-32-32-33-33-34-34, "0011223344"" received
14:29:39.825	Notification received from 0e7c3d0-480b-4f72-9ed9-054a0776428f, value: (0x) 31-30, "10"
14:29:39.825	"(0x) 31-30, "10"" received

FIGURE 4.45 – Réception des trames de données par l'application

4.4.4 Mise à jour du firmware du microcontrôleur

La réception et le traitement d'une commande AT prennent environ 30 ms. Chaque commande se compose de deux étapes : une pour la commande elle-même et une autre pour les données.

Dans cette approche, il est nécessaire d'envoyer deux commandes, "AT+BLEGATTSETATTR" et "AT+BLEGATTSENTFY", pour chaque trame de données, mais on est limité à l'envoi d'une seule commande toutes les 200 ms.

Cependant, dans la deuxième approche, en utilisant la commande unique "SEND-DATA", la réaction du module aux commandes envoyées ne prend que environ 25 ms. Chaque transfert de trame se fait en une seule commande et une seule opération, ce qui permet de modifier la valeur de l'attribut et de notifier les appareils connectés simultanément.

Cette approche permet une réduction significative du temps de transfert de fichiers. On estime que cette réduction dépassera les 50% du temps de transfert initial.

4.5 Performances du système

Le système réalisé garantit des performances optimales en répondant aux exigences suivantes :

- **Sécurité** : on a mis en place un appariement et un cryptage efficaces grâce à SMP, ainsi qu'un envoi sécurisé des mots de passe dans les données pour être vérifier par le microcontrôleur.
- **Fiabilité des données** : chaque trame est accompagnée d'un CRC (Cyclic Redundancy Check) qui permet de valider une réception correcte des données.
- **Réactivité** : le système offre un temps de réponse rapide de 25 ms par commande et un temps de transfert de fichiers inférieur à 10 minutes.
- **Connectivité** : la communication Bluetooth LE se fait sans interférences, garantissant une portée étendue et une connexion stable.

4.6 Conclusion

Dans ce chapitre, on a présenté en détail l'environnement de développement utilisé pour l'implémentation du projet. On a réalisé divers tests pour valider le fonctionnement du système en mettant en évidence les interactions entre les acteurs impliqués. On a examiné deux approches de développement distinctes, en mettant en valeur leurs spécificités et performances respectives. L'implémentation du projet permet d'obtenir un système fonctionnel et performant, répondant aux besoins définis précédemment.

CONCLUSION GÉNÉRALE

Durant ce stage de fin d'études, on a développé un programme embarqué qui permet la communication Bluetooth LE entre un IHM de servomoteur et une application Android. Ce programme garantit une connectivité sans fil au servomoteur, lui permettant d'être surveillé, contrôlé et optimisé à distance.

Dans le but de réaliser cela, on a entrepris une analyse de la solution existante, le module RN4020, afin d'identifier ses limites et inconvénients tels que son coût élevé, sa disponibilité restreinte et sa portée limitée. Après cette analyse, on a effectué des recherches pour trouver une solution alternative et on a opté pour le module ESP32. Ensuite, on a développé le programme embarqué pour répondre aux fonctionnalités requises et on a effectué les tests nécessaires avant son intégration dans les IHMs des servomoteurs. Ces fonctionnalités comprennent la gestion de connectivité, la commande du servomoteur, la récupération de ses données et la mise à jour du firmware de son microcontrôleur.

Les étapes réalisées ont conduit à l'obtention de deux solutions. La première solution utilise les commandes AT prédéfinies, tandis que la deuxième implique la création de commandes personnalisées. Le chapitre 4 présente les tests effectués avec les deux méthodes.

Avec la première méthode, on a obtenu un temps de réponse d'environ 30 ms pour les commandes AT et un temps de transfert pouvant aller jusqu'à 20 minutes. Pour la deuxième, on a obtenu un temps de réponse d'environ 25 ms pour les commandes personnalisées et un temps de transfert de fichier inférieur à 10 minutes.

La deuxième approche a permis une amélioration de la gestion du BLE dans une ESP32 en utilisant l'UART. Elle offre la possibilité de mettre à jour les données et de notifier les clients en une seule commande personnalisée, au lieu d'utiliser deux commandes AT distinctes. Les deux solutions ont réussi les tests réels sur l'IHM du servomoteur, et il ne reste plus qu'à finaliser la deuxième approche pour l'intégrer à la production des cartes IHM.

NETOGRAPHIE

<https://ia-tech.net/>

<http://tvaira.free.fr/bts-sn/activites/activite-ble/bluetooth-ble.html>

<https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>

<https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>

<https://www.espressif.com/>

<https://docs.espressif.com/projects/esp-at/en/latest/esp32/index.html>

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

<https://docklight.de/>

<https://www.st.com/en/microcontrollers-microprocessors/stm32g474re.html>

<https://www.oreilly.com/library/view/building-bluetooth-low/9781786461087/413df844-e8fe-4ab4-9353-22a9485f2fcf.xhtml>

<https://www.uuidgenerator.net/>

Résumé

Le présent rapport résume le travail accompli dans le cadre du projet de fin d'études visant à obtenir le diplôme nationale de licence Ingénierie des Systèmes Informatiques "Computer Engineering", Spécialité : systèmes embarqués et IoT, à la Faculté des Sciences de Tunis au sein de Innov-Alliance-Tech. L'objectif principal de ce projet était d'établir une connectivité sans fil pour les servomoteurs. À cet effet, nous avons développé un programme embarqué qui permet de commander le servomoteur, de récupérer des données et de mettre à jour son microcontrôleur.

Ce programme offre une communication bidirectionnelle entre l'interface homme-machine (IHM) du servomoteur et une application Android, en utilisant à la fois la communication sans fil (Bluetooth Low Energy - BLE) et la communication série (Universal Asynchronous Receiver Transmitter - UART) via la carte ESP32. Nous avons exploré deux méthodes différentes, l'une basée sur les commandes AT et l'autre utilisant des commandes personnalisées.

Mots clé : Bluetooth Low Energy ; UART ; GAP ; GATT ; SMP ; système temps réel ; ESP32 ; Commandes AT ; STM32

Abstract

The present report summarizes the work accomplished within the scope of the final year project aimed at obtaining a national Bachelor's Degree in computer engineering, specializing in embedded systems and IoT, at the Faculty of Sciences of Tunis within Innov-Alliance-Tech. The main objective of this project was to establish wireless connectivity for servomotors. To achieve this, we developed an embedded program that enables the control of the servomotor, data retrieval, and microcontroller updates.

This program provides bidirectional communication between the servomotor's human-machine interface (HMI) and an Android application, utilizing both wireless communication (Bluetooth Low Energy - BLE) and serial communication (Universal Asynchronous Receiver Transmitter - UART) via the ESP32 board. We explored two different methods, one based on AT commands and the other utilizing custom commands.

Keywords : Bluetooth Low Energy ; UART ; GAP ; GATT ; SMP ; Real-time system ; ESP32 ; AT Commands ; STM32

ملخص

يلخص هذا التقرير العمل المنجز في إطار مشروع نهاية الدراسات من أجل الحصول على درجة البكالوريوس الوطنية في هندسة أنظمة الكمبيوتر ، تخصص في الأنظمة المضمنة وإنترنت الأشياء ، في كلية العلوم بتونس ضمن شركة Innov-Alliance-Tech . كان الهدف من هذا المشروع هو إنشاء اتصال لاسلكي لمحرك مؤازر ، ولهذا الغرض قمنا بتطوير برنامج مدجع يسمح بالتحكم في المحرك المؤازر واسترداد البيانات وتحديث وحدة التحكم الدقيقة الخاصة به .

يوفر هذا البرنامج اتصالاً ثنائياً الاتجاه بين واجهة المستخدم للآلية وتطبيق Android ، باستخدام كل من الاتصالات اللاسلكية (Bluetooth Low Energy - BLE) والاتصال التسلسلي (Universal Asynchronous Receiver Transmitter - UART) من خلال لوحة تطوير ESP32 . اكتشفنا طريقتين مختلفتين ، واحدة تعتمد على أوامر AT والأخرى باستخدام أوامر مخصصة .

الكلمات الدالة : STM32 ، أوامر AT ، ESP32 ، نظام زمن حقيقي ، SMP ، GAP ، UART ، GATT ، Bluetooth Low Energy