



Mémoire de Projet de fin d'étude

Préparé par

CHABBAR Houria

Pour l'obtention du diplôme

Ingénieur d'Etat en

SYSTEMES ELECTRONIQUES & TELECOMMUNICATIONS

Intitulé

**Innovation et réalisation d'une solution de
géolocalisation combinée à la
télécommunication**

Encadré par :

Pr JORIO Mohammed

Mr YACHOU Driss (ATD)

Mr ERRABJA Naoufal (ATD)

Mr CHAMI Mouhcine (INPT)

Soutenu le 23 juin 2014, devant le jury composé de :

Pr JORIO Mohammed..... : Encadrant

Pr ABDI Farid..... : Examineur

Pr EL MOUSSAOUI Hassan..... : Examineur

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا

عَلَّمْتَنَا ^ص إِنَّكَ أَنْتَ الْعَلِيمُ

الْحَكِيمُ

بِسْمِ اللَّهِ
الْعَظِيمِ

سورة البقرة

الآية: 32

Dédicaces

A mes parents

Je vous dois tout. Vous représentez pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Aucune dédicace ne pourrait refléter la profondeur de mon amour. Vous trouvez ici le témoignage de ma reconnaissance pour les sacrifices, la compréhension et le soutien dont vous avez toujours fait preuve à mon égard.

A mes sœurs et mes frères

Votre fraternité me réchauffe le cœur. Vous m'avez inlassablement soutenu et encouragé tout au long de mes études, que vous trouveriez dans ce travail l'expression de mon amour et mon profond respect.

A toute la famille

Un affectueux hommage

A tous mes amis (es)

En reconnaissance des liens de solidarité et de sympathie qui nous ont toujours réunis

Houria...

Remerciements

C'est avec un immense plaisir que je me conforme à la tradition en exprimant ma gratitude envers les personnes qui par leurs conseils et leurs critiques ont contribué à la réalisation de ce projet.

Il m'est agréablement de saisir l'occasion de la présentation de ce travail, pour exprimer mes vifs remerciements à mes encadrants Mr. **YACHOU Driss**, Mr. **GABREAU Christophe** (créateurs de la startup WiSmart) et Mr. **ERRABJA Naoufal** (Team Leader chez WiSmart) pour leur encadrement judicieux et leurs précieux conseils qu'ils n'ont cessé de me divulguer tout au long de cette étude. Qu'ils trouvent ici l'expression de ma profonde gratitude.

Je remercie particulièrement Monsieur **CHAMI Mouhcine**, professeur à l'INPT de m'avoir fait profiter de son immense connaissance sur les systèmes embarqués, tant au niveau théorique qu'expérimental, et pour ses conseils précieux.

Je remercie spécialement tous les membres de l'équipe du projet GeoComm, pour leur aide précieuse et leur retour d'expérience qui m'a été d'une grande utilité.

Mes sincères remerciements vont également à mon professeur encadrant Mr. **JORIO Mohammed** et aux membres du jury qui m'ont fait l'honneur de bien vouloir juger ce travail.

A l'issue de trois agréables années passées au sein de la Faculté des Sciences et Techniques FES (FSTF), j'adresse des remerciements particuliers à l'ensemble du corps professoral et administratif pour l'incalculable qualité de l'enseignement qui nous a été dispensé.

Résumé

L'explosion du marché de la mobilité a une profonde incidence sur la productivité des équipements de services de télécommunications, et développements de nouveaux services, notamment le service de la géolocalisation. Ce dernier offre diverses possibilités de positionnement dans différents domaines.

L'objectif principal de ce travail est de mettre en œuvre une plateforme de géolocalisation indoor/outdoor combinée à un système de télécommunication pour permettre l'échange de données du localisé.

Au cours de ce projet, une première version du prototype du module GEO a été réalisée dans le but de se localiser en utilisant le GPS en outdoor et le Bluetooth Low Energy en indoor ensuite remonter les données de localisation à l'application centrale via Wifi.

Mots-clés :

Géolocalisation Outdoor/indoor, GPS, BLE, Wi-Fi, *Motion Sensors*.

Abstract

The explosion of mobile market has a profound impact on the productivity of telecommunications equipments, and development of new services, including the geolocation service. It offers various opportunities of positioning in different areas.

The main objective of this work is to implement a platform for indoor/outdoor geolocation combined with a telecommunication system for the exchange of data located.

During this project, a first version of the prototype of the GEO module was conducted to be located by using GPS in outdoor and Bluetooth Low Energy in indoor then send location data to the central application via Wifi.

Keywords:

Outdoor/indoor geolocation, GPS, BLE, Wifi, Motion Sensors.

Liste des abréviations

ABI	Allied Business Intelligence
ATD	Advanced Technology Developpement
ATT	Attribute Protocol
BGAPI	Bluegiga Application Programming Interface
BLE	Bluetooth Low Energy
CEN	Centrale
Cell ID	Cellule Identification
CRC	Cyclic Redundancy Check
DMP	Digital Motion Processor
EAGLE	Easily Applicable Graphical Layout Editor
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EOTD	Enhanced Observed Time Difference
GAP	Generic Access Profile
GPS	Global Positioning System
GSM	Global System Mobile
HCI	Host Control Inteface
I2C	Inter Integrated Circuit
IP	Internet Protocol
ID	Identifiant
iOS	iPhone Operating System
LL	Link Layer
L2CAP	Logical link control and adaptation protocol
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
MPU	MIDI Processing Unit
NMEA	National Marine Electronics Association
PDU	Protocol Data Unit
PHP	Hypertext Preprocessor
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indication
SCL	Serial Clock
SDA	Serial Data
SDK	Software Development Kit
SI	Système International
SPI	Serial Peripheral Interface
SysML	Systems Modeling Language
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver Transmitter
WAP	Wireless Application Protocol
WBS	Work Brekdown Structure
WEP	Wired Equivalent Privacy
WIFI	Wireless Fidelity
WPA	Wi-Fi Protected Access
XML	Extensible Markup Language
3G	Troisième Génération

Liste des figures

Figure 1. Schéma synoptique de la plateforme GeoComm.....	19
Figure 2. Modèle en V.....	20
Figure 3. Diagramme de GANTT	22
Figure 4. Schéma illustratif de la plateforme GeoComm.....	33
Figure 5. Schéma illustratif de l'identification des Beacons BLE	35
Figure 6. Schéma illustratif du suivi des motions Sensors.....	35
Figure 7. Affichage des coordonnées sur l'application centrale	36
Figure 8. Récepteur GPS ST32 Skytraq.....	39
Figure 9. Principe de la réception GPS	40
Figure 10. Module BLE Bluegiga	41
Figure 11. Exemple trame GGA	41
Figure 12. Format de paquet générique du BLE	43
Figure 13. Pile protocolaire du BLE112	44
Figure 14. Module MPU6050 Accéléromètre+Gyroscope	45
Figure 15. Module Wifi WRL-3000	47
Figure 16. Mode de Fonctionnement I2C	49
Figure 17. Principaux blocs de l'UART.....	50
Figure 18. Principe de fonctionnement SPI	51
Figure 19. Diagramme de cas d'utilisation généralisé.....	54
Figure 20. Diagramme de cas d'utilisation détaillé.....	55
Figure 21. Diagramme de séquence « système »	56
Figure 22. Diagramme de séquence « erreur connexion avec la centrale »	57
Figure 23. Diagramme d'état du module GEO	57
Figure 24. Architecture globale du composant GEO	62
Figure 25. Flux externes échangés	63
Figure 26. Bloc interne du GEO.....	66
Figure 27. Bloc interne orienté énergie.....	67
Figure 28. Carte Arduino MEGA.....	69
Figure 29. CC Debugger	69
Figure 30. Logo du logiciel ISIS	70
Figure 31. Logo du logiciel Eagle	70
Figure 32. Moniteur série du logiciel Arduino.....	71
Figure 33. Architecture du module GEO	73
Figure 34. Soudage du module GPS ST32 Skytraq.....	74
Figure 35. SmartGPS.....	74
Figure 36. Résultat de la localisation avec GPS sous GPS viewer	75
Figure 37. Communication SmartGPS avec Arduino	76
Figure 38. Résultat retourné par le SmartGPS	76
Figure 39. Branchement du module MPU6050 avec l'Arduino	77
Figure 40. Réalisation du branchement de MPU6050 avec Arduino.....	78
Figure 41. Acquisition des données du MPU-6050	79
Figure 42. Résultat de la calibration.....	80
Figure 43. Valeurs retournées par le composant MPU6050	80
Figure 44. BLE112 avant soudage	81

Figure 45. Face arrière du module Bluegiga BLE112 soudé	81
Figure 46. Face avant du module Bluegiga BLE112 soudé	82
Figure 47. Branchement du BLE112 avec le CC Debugger sous Eagle	83
Figure 48. Réalisation du schéma de branchement BLE112 avec CC Debugger	83
Figure 49. Compilateur BGbuild.....	84
Figure 50. Interface BLE SW Update Tool.....	84
Figure 51. Résultat de la compilation.....	85
Figure 52. Affichage du BGBuild log.....	85
Figure 53. Détection du module BLE112 par l'application Android «BLE Scanner »	86
Figure 54. Beacons avec leur pile CR2032 3V	86
Figure 55. Résultat détection des Beacons par l'application Android « easiBeacon ».....	87
Figure 56. Protocol BGAPI.....	88
Figure 57. Schéma branchement BLE112 avec la carte Arduino sous Eagle	89
Figure 58. Réalisation du schéma branchement BLE112 avec la carte Arduino	89
Figure 59. Résultat du scan	90
Figure 60. Résultats de détection des deux Beacons.....	90
Figure 61. Zones de proximité d'un Beacon	91
Figure 62. Branchement du module Wifi avec la carte Arduino sous Eagle	92
Figure 63. Réalisation du schéma du branchement Wifi avec Arduino	93
Figure 64. Connexion du module WRL3000 avec le point d'accès.....	93
Figure 65. Résultat de l'établissement de la connexion	94
Figure 66. Procédure d'envoi et réception par socket UDP	95
Figure 67. Exemple d'une trame de localisation reçue par le serveur	96
Figure 68. Donnée avant et après cryptage	96
Figure 69. Donnée cryptée	97
Figure 70. Schéma illustratif de la version 0 du GEO	97
Figure 71. Montage du prototype V0.....	98
Figure 72. Résultat du scan et l'envoi de la localisation vers le serveur.....	99
Figure 73. Affichage de la localisation du module GEO sur l'application centrale	100
Figure 74. Schéma brochage du MAX3232	105
Figure 75. Simulation du Schéma MAX3232 sous ISIS.....	106
Figure 76. Branchement de l'adaptateur RS232	106
Figure 77. Interface GPS Viewer	107
Figure 78. Branchement du ST-32 au PC à travers RS232.....	107
Figure 79. Configuration du Baudrate et le port Com.....	108
Figure 80. Branchement du GPS ST32 avec l'Arduino sous Eagle.....	108
Figure 81. Structure du message binaire	109
Figure 82. SmartGPS Board.....	110

Liste des tableaux

Tableau 1. Ordonnancement des tâches	21
Tableau 2. Comparatif des technologies de géolocalisation	30
Tableau 3. Liste de sélection des composants	38
Tableau 4. Description détaillée du cas d'utilisation «Diffuser les données de navigation »	55
Tableau 5. Description détaillée du cas d'utilisation « Informer sur l'état du système »	56
Tableau 6. Flux échangés entre GEO et CEN	63
Tableau 7. Flux échangés entre GEO et Infrastructure GeoComm.....	63
Tableau 8. Les Fonctions du Bloc d'acquisition	64
Tableau 9. Les Fonctions Du bloc de traitement.....	65
Tableau 10. Fonctions du Bloc de Communication	66
Tableau 11. Pins WRL-3000	91

Table des matières

Dédicaces	3
Remerciements	4
Résumé	5
Abstract	6
Liste des abréviations	7
Liste des figures	8
Liste des tableaux	10
Chapitre 1	16
Contexte général du Projet	16
1. Contexte général du projet	17
1.2 Contexte du projet	18
1.2.1 Problématique.....	18
1.2.2 Présentation du projet :.....	18
1.2.3 Objectifs :	19
1.3 Planification du projet	20
Conclusion.....	23
Chapitre 2	24
2. Etat de l’art du projet	25
2.1 Le concept Géolocalisation	25
2.2 L’évolution des besoins de géolocalisation sur le marché.....	25
2.3 Services potentiels à offrir et création de besoins	26
2.4 Les moyens de géolocalisation actuels à grande échelle.....	26
2.4.1 Les systèmes satellitaires	26
2.4.2 Géolocalisation par GSM	27
2.4.3 Géolocalisation par Wi-Fi	27
2.4.4 Géolocalisation par RFID.....	27
2.5 Solutions proposées par des Start-up.....	28
2.5.1 Fidzup, solution basée sur l'envoi de signaux sonores.....	28
2.5.2 Pole Star, un pionnier de la localisation Wi-Fi	28
2.5.3 Insiteo combine le Wi-Fi au BLE	29
2.5.4 Ticatag se spécialise dans le Bluetooth Low Energy	29

2.5.5	Ubudu, des bons plans aux bornes BLE.....	29
2.5.6	Tableau récapitulatif des techniques de géolocalisation	30
	Conclusion.....	31
	Chapitre 3.....	32
	3. Choix technologiques, Protocoles de communication	33
3.1	La solution proposée.....	33
3.2	Principe de la solution proposée	34
3.3	Étude et choix technique des composants	36
3.4	Synthèse de l'étude	39
3.4.1	Le récepteur : GPS ST32 Skytraq	39
3.4.2	Le module Bluetooth low energy BLE112	41
3.4.3	Le module MPU-6050 Motion Processing Unit	45
3.4.4	Le module Wifi WRL-3000	47
3.5	Protocoles de communication :.....	48
3.5.1	Protocole I2C.....	49
3.5.2	Protocole UART :.....	49
3.5.3	Protocole SPI.....	50
3.6	Choix du microcontrôleur.....	51
	Conclusion.....	52
	Chapitre 4.....	53
	4. Analyse et Conception.....	54
4.1	Description fonctionnelle détaillée	54
4.1.1	Identification des acteurs.....	54
4.1.2	Diagramme de cas d'utilisation.....	54
4.1.3	Etude de comportement du système	56
4.1.4	Diagramme d'états	57
4.2	Modélisation des exigences	59
4.2.1	Description des performances du composant.....	59
4.2.2	Contraintes opérationnelles du composant.....	60
4.3	Conception Matériel	62
4.3.1	Architecture globale du composant.....	62
4.3.2	Spécification des flux externes :.....	63
4.3.3	Description des Blocs.....	64
4.3.4	Diagramme des Bloc Interne du GEO.....	66
	Conclusion.....	67

Chapitre 5	68
5. Outils, développement et réalisation	69
5.1 Outils matériels.....	69
5.1.1 Carte Arduino.....	69
5.1.2 CC debugger.....	69
5.2 Outils logiciels.....	70
5.2.1 ISIS Proteus Professional Version 8.....	70
5.2.2 Eagle.....	70
5.2.3 Logiciel Arduino.....	71
5.2.4 BLE SW Update Tool.....	71
5.2.5 Skytraq GPS Viewer.....	71
5.3 Langages utilisés.....	72
5.3.1 BGScript Scripting langage.....	72
5.3.2 Le langage XML.....	72
5.3.3 Langage de programmation Arduino.....	72
5.3.4 Serveur web embarqué.....	72
5.4 Développement et réalisation.....	73
5.4.1 GPS ST32 Skytraq.....	73
5.4.2 SmartGPS.....	74
5.4.3 Motion Sensors MPU6050.....	77
5.4.4 Bluetooth Low Energy BLE 112.....	81
5.4.5 Wifi WRL-3000.....	91
5.4.6 Version 0 du GEO.....	97
Conclusion.....	100
Conclusion générale et perspectives.....	101
Références.....	103
Annexe 1: Circuit d'adaptation MAX3232.....	105
Annexe 2 : Programmation GPS ST32.....	107
Annexe 3 : SmartGPS Board.....	110
Annexe 4: Configuration BLE112.....	111

Introduction Générale

De nos jours, le marché de la mobilité connaît un essor phénoménal au niveau de la productivité des équipements (Smartphones, Tablettes...), de services de télécommunications, et développements de nouveaux services. Cela a donné naissance naturellement à l'idée de fournir des services en fonction de la position de l'utilisateur : un besoin de localisation est donc apparu.

Les applications offrant ce genre de services sont de plus en plus incontournables et répondent à des attentes très différentes dans plusieurs domaines tels que: la sécurité, la santé, l'environnement, l'urbanisme, le transport et les télécommunications.

La qualité de ces services est étroitement liée à la précision de positionnement. Un intérêt considérable s'est alors développé au sein de la communauté scientifique et industrielle pour les techniques de localisation. Lorsque l'objet à localiser se trouve dans un environnement à ciel ouvert, le système GPS est utilisé pour déterminer sa position (longitude, latitude et altitude) à tout moment. Il permet d'atteindre des précisions de l'ordre de quelques mètres. Par contre dans les environnements urbains denses ou à l'intérieur de bâtiments (résidences, aéroports, centres commerciaux, musées, hôpitaux) l'efficacité du GPS est de plus en plus restreinte.

Une étude approfondie des différentes approches est nécessaire pour prévoir la meilleure solution à déployer. Cette solution peut aussi être issue de la fusion de plusieurs technologies. Les technologies "indoor", prises individuellement, connaissent des limitations comme une portée limitée, ou une dérive au cours du temps. Il est alors nécessaire de combiner plusieurs technologies de localisation à l'aide d'une infrastructure faisant une association entre elles. Une telle solution doit conduire à une meilleure estimation de la position.

Mon projet de fin d'étude s'inscrit dans le cadre du projet « Innovation et réalisation d'une solution de Géolocalisation combinée à la Télécommunication »

Il consiste à concevoir et réaliser un module GEO pour la géolocalisation en utilisant le GPS en Outdoor, le Bluetooth Low Energy, et les *Motion Sensors* en Indoor et par la suite envoyer les coordonnées du localisé vers l'application centrale via Wifi ou 3G, visant une précision de localisation inférieure à 1 mètre.

Le présent rapport est structuré comme suit:

- Le premier chapitre est consacré à la présentation de l'organisme d'accueil, le contexte du projet ainsi que la problématique traitée et les objectifs visés et la conduite du projet adoptée pour le déroulement du stage.
- Le deuxième chapitre trace le périmètre de notre système à la lumière de l'état de l'art du projet. Ce chapitre s'articule autour de l'évolution des besoins de la géolocalisation, il contient une étude des solutions techniques existantes sur le marché.
- Le troisième chapitre s'intéresse à la présentation du choix technologique adopté et s'intéresse à une étude et choix technique des composants utilisés afin de mettre en œuvre la solution proposée.
- Le quatrième chapitre décrit l'analyse et la conception de la solution apportée, en utilisant les diagrammes SysML.
- Le dernier chapitre dresse une présentation de l'ensemble des outils matériels et logiciels utilisés et déployés pour aboutir à la réalisation du projet.
- Une conclusion et les perspectives pour améliorer le projet clôturent ce document.
- Des annexes apportant un complément aux différentes parties de ce rapport

Chapitre 1

Contexte général du Projet

Ce chapitre est consacré à la description du contexte général du projet, la présentation de l'organisme d'accueil, la précision du cadre du projet puis les objectifs visés, ensuite l'énumération des étapes de travail à réaliser pour achever ce projet selon le diagramme de GANTT.

1. Contexte général du projet

1.1 Organisme d'accueil

ATD 'Advanced Technology Development

ATD est un centre de recherche et de développement pour les marchés des nouvelles technologies, créé en juin 2010 selon une convention d'incubation à L'INPT (Institut National des Postes et Télécommunication). L'objectif de cette initiative est la création d'une entreprise marocaine spécialisée dans les services de développement, de test et d'innovation de solutions pour de nouveaux marchés spécifiques.

ATD a aussi pour mission la valorisation des compétences pour le développement Logiciel et le prototypage électronique pour les systèmes embarqués.

La période d'incubation est consacrée aux montages de structure spécialisée en

- Développement des applications Homme/Machine, Machine/Machine
- Test et Validation de solution en systèmes embarqués, Logiciel et matériel
- Prototypage de solutions électroniques
- Montage de programmes de formation appliquée et professionnalisante.

Suite aux résultats des projets ATD, en matière de formation, de développement et de test, la société WiSmart, a été créé en Avril 2014. Wismart offrira ses services et proposera des solutions innovantes pour le marché de la Mobilité. WiSmart proposera des programmes de formations appliquées, en adéquation avec le besoin du marché local et mondial en ressources, du développement informatique, des méthodologies de test, ainsi que le prototypage électronique de nouvelles solutions pour le marché des systèmes embarqués.

L'ambition de Wismart est de participer à l'évolution de l'écosystème marocain des nouvelles technologies et la valorisation des compétences pour un service de haut niveau.

ATD offrira ses services pour les marchés suivants :

- Aéronautique
- Auto motive
- Télécommunication
- Energie renouvelable
- MultiMedia
- Médical
- Transport
- Education
- Sécurité

1.2 Contexte du projet

1.2.1 Problématique

Avec le développement des réseaux mobiles, le marché de la géolocalisation a fait son apparition. La particularité la géolocalisation est que l'utilisateur mobile est automatiquement localisé et n'a pas besoin d'indiquer lui-même sa position. Une forte activité de recherche sur les méthodes de localisation s'est alors développée, l'objectif étant de fournir une bonne précision de localisation afin d'assurer une bonne qualité de ces services. La localisation d'un objet ou d'une personne revient à trouver sa position dans un espace de coordonnées cartésiennes, ou simplement le situer dans un environnement défini (une rue, une pièce ou un étage par exemple) ; le choix des techniques de localisation utilisées dépend des applications visées. De nombreuses solutions pour la localisation des personnes (et objets) ont été proposées sur le marché. Elles devaient répondre à des exigences de coût, de complexité et de précision.

La plus connue des solutions est l'utilisation des récepteurs GPS. Cependant, l'extension de ces services à l'intérieur reste limitée à cause de la faible précision des méthodes de localisation dans ces environnements. Afin de se libérer des limitations du système GPS dans les environnements urbains denses ou intérieurs, des solutions reposant sur différents types de capteurs ou des réseaux sans fils ont fait leur apparition.

1.2.2 Présentation du projet :

Le projet GeoComm a pour objectif de réaliser une plateforme de géo-localisation Outdoor/indoor visant une précision de localisation < 1 mètre, cette solution est constituée de deux parties :

- **Partie client** : se charge de localiser le client et envoyer ses informations de localisation vers une base de données, cette partie doit permettre de localiser deux types de clients :
 - Clients utilisant Smartphone(Android) : en utilisant une application Android qui permettra de remonter les informations de localisation (longitude, latitude, altitude, date ...) vers une base de données distante.
 - Client utilisant module GEO : module hardware à concevoir qui devra définir la localisation précise du porteur et la remonter vers la base de données distante.

- **Partie serveur** : application web qui se charge des traitements de la base de données de localisation et constitue le tableau de bord qui permettra le suivi des clients. Elle devra supporter :
 - La sécurité : authentification par login, password
 - Ajout, suppression et modification de profils utilisateurs
 - Afficher le profil d'un client
 - Afficher le chemin emprunté par un client donné sur une carte (Map personnalisée à développer, indépendante des cartes propriétaire Google Maps)
 - Gérer l'historique des localisations client
 - Service personnalisé ...

La figure suivante illustre une description simplifiée de la plateforme GeoComm :

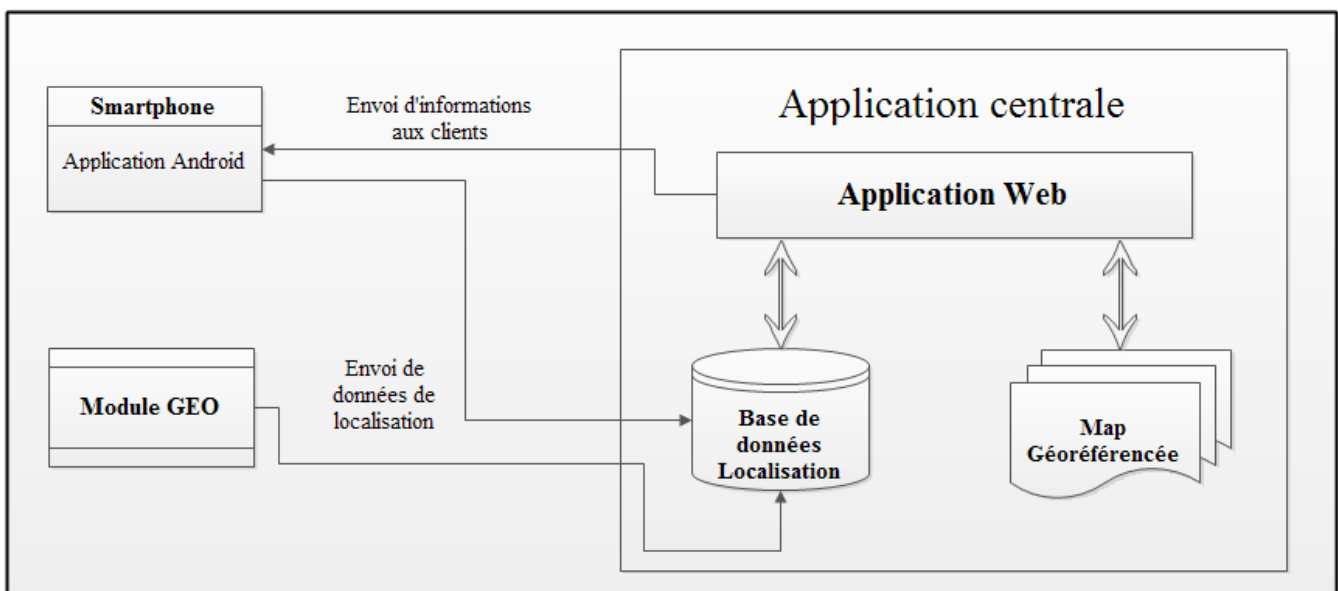


Figure 1. Schéma synoptique de la plateforme GeoComm

1.2.3 Objectifs :

Notre projet de fin d'études fait partie du projet GeoComm. Il consiste à réaliser le module GEO comme solution pour la géolocalisation. Ce module GEO devra:

- Permettre de localiser le client en outdoor en utilisant le GPS.
- Permettre de se localiser en indoor en utilisant BLE et les « Motion Sensors ».
- Envoyer les informations de localisation vers une base de données de l'application centrale via Wifi ou 3G.
- Assurer une faible consommation d'énergie pour assurer une autonomie optimale
- Etre de dimension raisonnable (taille, poids)

1.3 Planification du projet

Dans la section suivante, nous rappelons la méthodologie de conception suivie, qui s'inspire des démarches génériques proposées par les standards de l'ingénierie des systèmes. Dans cet esprit, nous avons choisi d'appliquer la démarche proposée par le cycle de vie des systèmes selon le modèle en V (Figure 2). Nous sommes dans la partie descendante de la représentation qui conduit à un prototypage de notre dispositif GEO, puis à une étape de validation dans sa partie montante. Ce modèle propose donc une démarche continue qui permet d'organiser les étapes intermédiaires de vérifications : vérification de la logique fonctionnelle puis vérification des choix technologiques. Il convient donc tout d'abord, d'établir la solution logique fonctionnelle du système à concevoir sur la base du cahier des charges. Cette solution est modélisée en utilisant le formalisme SysML (Systems Modeling Language) à travers les diagrammes que propose ce langage (diagramme des cas d'utilisation et diagramme de séquence....). Une fois la modélisation définie, nous pouvons introduire les choix technologiques qui nous permettent de concrétiser la conception physique de GEO. Cette conception inclut :

- Les choix matériels : composants, capteurs, architecture réseau, dispositif de localisation et d'identification, etc.
- L'aspect logiciel du système : algorithmes de diagnostic utilisés, gestion de la visualisation, sécurisationetc.

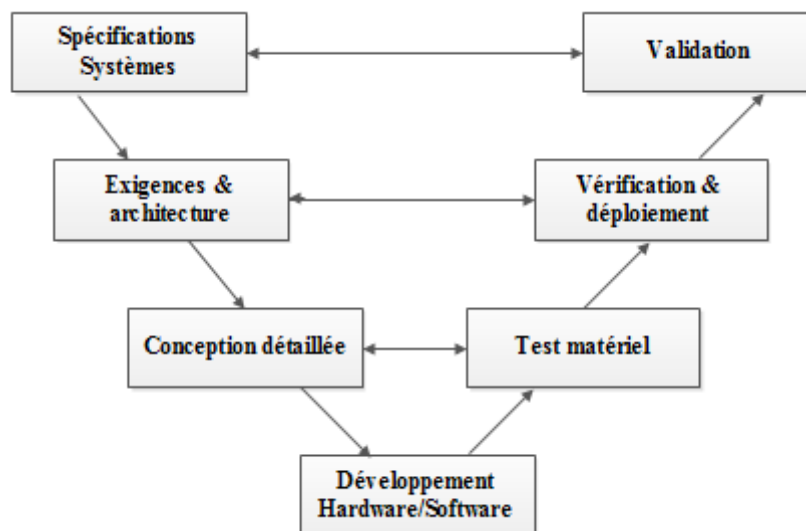


Figure 2. Modèle en V

Dans le volet logiciel, nous proposons d'exploiter les données recueillies par le système d'une façon personnalisée et auto adaptative.

De plus, le cycle en V met en évidence la nécessité d'anticiper et de préparer dans les étapes descendantes les « attendus » des futures étapes montantes : ainsi les attendus des tests de validation sont définis lors des spécifications, les attendus des tests unitaires sont définis lors de la conception. Chaque partie est développée en respectant le cycle et les tests d'intégrations, constituant la validation de la conception architecturale de tout le système. Enfin, les tests de validation du système vérifient les spécifications globales du système.

Après avoir figé la méthodologie de travail, une planification du projet s'avère nécessaire. Elle consiste à prévoir le déroulement des tâches tout au long des phases constituant le cycle de développement.

Les étapes de la planification :

- Le découpage en tâches (WBS WorkBreakdown Structure) : la structure hiérarchique des tâches du projet.
- L'ordonnancement des tâches: l'élaboration d'un plan d'action permettant de déterminer les séquences possibles entre l'exécution des tâches

Le tableau suivant représente la répartition des différentes tâches, leurs durées ainsi que les ressources employées à leurs réalisations :

Tâches	Tâches antérieures	Durée (semaines)
A- Rédaction du rapport	B	14
B- Phase recherche et documentation	/	2
C- Etude théorique et choix techniques des composants	B	3
D- Phase d'analyse et conception de l'architecture du système	C	3
E- Brochage et schéma électronique	D, C	2
F- Test de fonctionnement des composants	E	2
G- Intégration des composants	F	1
H- Développement et vérification du software	G	9
I- Optimisation et prototypage	H	8

Tableau 1. Ordonnancement des tâches

La planification temporelle est une étape avant-projet qui non seulement permet de délimiter le périmètre du projet mais aussi de prévoir le déroulement des activités tout au long de la période de stage. La figure 3 présente le détail de cette planification à travers le diagramme de GANTT.

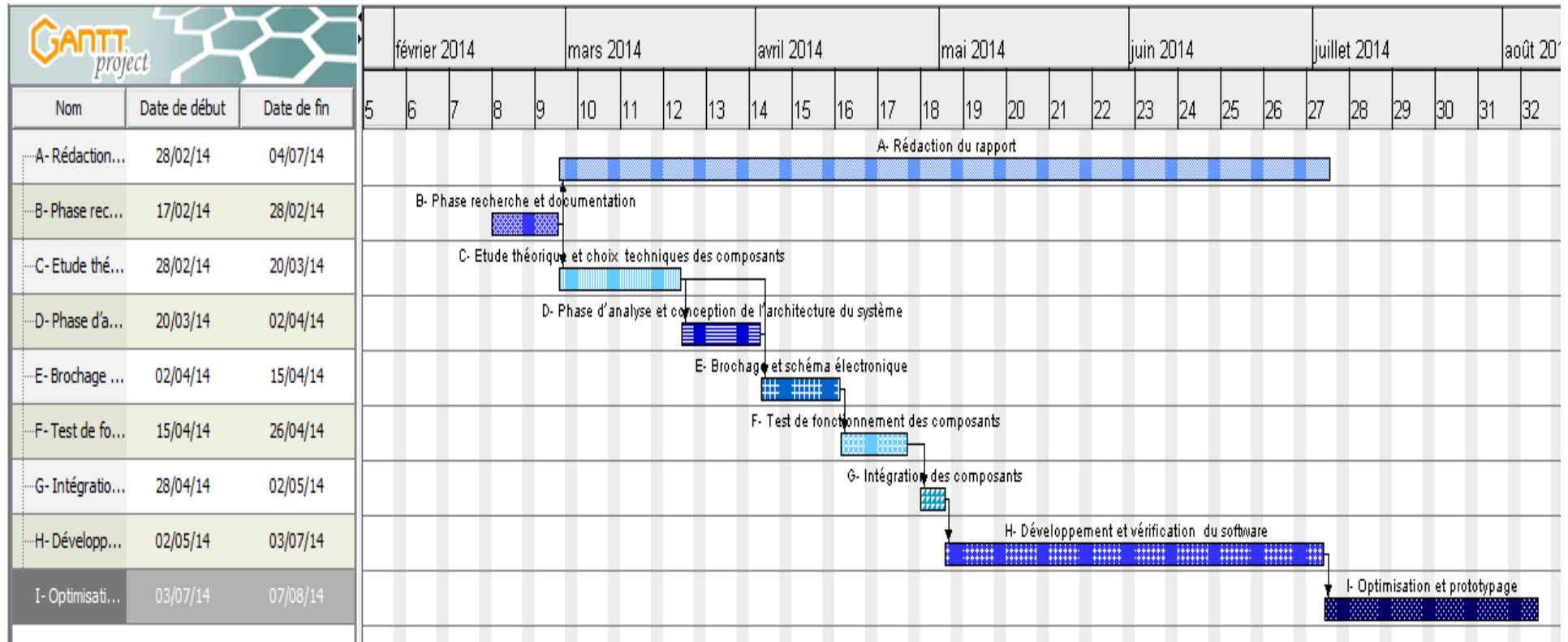


Figure 3. Diagramme de GANTT

Conclusion

Au terme de ce premier chapitre et après avoir présenté l'organisme d'accueil, nous avons, de manière globale, présenté le projet en le plaçant dans son contexte général, en définissant les objectifs du projet ainsi que la démarche suivie pour sa réalisation. Dans le chapitre qui suit, nous allons entamer la phase de préparation de notre projet à savoir l'état de l'art et l'étude de l'existant.

Chapitre 2

Etat de l'art du projet

De nos jours, les techniques de géolocalisation sont multiples. Une bonne maîtrise et connaissance de ces diverses méthodes sont nécessaires afin de judicieusement dimensionner sa propre solution de géolocalisation. Cette solution ne doit pas être surdimensionnée, sinon elle entraîne un surcoût soit au niveau de l'infrastructure soit au niveau du terminal de localisation.

Ce deuxième chapitre introductif, a pour but d'effectuer un panorama des solutions de géolocalisation disponibles, en mettant en avant les performances et les contraintes qui leurs sont liées.

2. Etat de l'art du projet

2.1 Le concept Géolocalisation

L'homme a toujours eu besoin de localiser les objets et de se situer dans l'environnement. Pour répondre à cette nécessité, plusieurs techniques ont été utilisées. Au début de l'Humanité, l'homme utilise les pierres (ou montagnes) pour se repérer.

Le concept de la géolocalisation ou géoréférencement est un procédé permettant de positionner un objet (une personne, etc) sur un plan ou une carte à l'aide de ses coordonnées géographiques.

Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un système de positionnement par satellites (et un récepteur GPS par exemple) ou à d'autres techniques) et de publier (en temps réel ou de façon différée) ses coordonnées géographiques (latitude/longitude). Les positions enregistrées peuvent être stockées au sein du terminal et être extraites postérieurement, ou être transmises en temps réel vers une plateforme logicielle de géolocalisation. La transmission temps réel nécessite un terminal équipé d'un moyen de télécommunication, radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis internet.

2.2 L'évolution des besoins de géolocalisation sur le marché

La géolocalisation est une technologie clé d'une immense valeur pour le monde du commerce, et de manière générale pour tout endroit accueillant du public. Son adoption massive est en route et le marché a atteint sa maturité : les business modèles sont établis, le paysage concurrentiel est clair, certains regroupements et partenariats commencent à voir le jour ...

La maturité du marché de la géolocalisation se voit simplement en observant la demande croissante des propriétaires de centres commerciaux, directeurs des grands magasins, grands aéroports, gares, centres de congrès et d'exposition...

- Les systèmes de géolocalisation sont aujourd'hui incontournables et répondent à des attentes très différentes. En 2010, 34% des applications téléchargées sur l'Apple Store et 38% des applications sur Android accédaient aux fonctionnalités de géolocalisation.

- Juniper Research estime à 13 Milliards \$ le marché mondial de géolocalisation pour 2014 ;
- Selon l'agence de recherche ABI Research, le marketing géolocalisé pourrait représenter un chiffre d'affaires de 1,8 milliard de dollars en 2015, contre environ 43 millions en 2010 ;
- En France plus de 250 000 véhicules équipés par des boîtiers de géolocalisation ;
- Les applications de géolocalisation engendrent un engouement très fort particulièrement chez la jeune population. [1]

2.3 Services potentiels à offrir et création de besoins

- Tracer les personnes dites à risques : jeunes enfants, malades d'Alzheimer, personnes âgées, détenus, cette utilisation dans un but d'améliorer la sécurité, dépêcher des secours et porter assistance à ces personnes ;
- Se repérer dans un espace fermé, d'avoir accès à une fonction d'itinéraire ou encore de connaître les commerces et boutiques présents sur le parcours qu'ils vont emprunter ;
- Aider les passagers à s'orienter dans les aéroports, les aider à trouver le parcours vers leur porte d'embarquement, un parking, un ascenseur, un restaurant ou une boutique à proximité ;
- Géolocalisation des taxis : la société des taxis peut repérer immédiatement le véhicule le plus proche d'une demande « client ». cela a pour objectif de gagner le temps et d'optimiser les trajets et réaliser plus de courses pour une meilleure rentabilité quotidienne, pas de déplacements inutiles pour les chauffeurs loin du secteur ;
- Publicité géo-ciblée : dans un grand magasin ou un supermarché, connaître les itinéraires des consommateurs aide à créer une expérience d'achat plus adaptée aux habitudes des clients tout en améliorant les relations avec la clientèle et en ciblant mieux la communication publicitaire sur mobile.

2.4 Les moyens de géolocalisation actuels à grande échelle

2.4.1 Les systèmes satellitaires

La géolocalisation par satellite consiste à calculer, grâce aux signaux émis à intervalle régulier par une constellation de satellites prévue à cet effet, la position actuelle sur la face terrestre d'un terminal équipé d'une puce compatible. La trilatération permet de connaître la position du capteur GPS. Au minimum, quatre satellites doivent être captés pour obtenir une position.

Cette position est alors traduite en termes de latitude, longitude et parfois altitude et peut alors être représentée physiquement sur une carte. Le réseau satellite de positionnement le plus connu est le GPS (Global Positioning System), bien que l'alternative Européenne nommée Galileo soit en cours de déploiement. [2]

2.4.2 Géolocalisation par GSM

Le Global System Mobiles (GSM) est le réseau le plus utilisé. La précision du positionnement par GSM peut aller de 200 mètres à plusieurs kilomètres, selon que le terminal se trouve en milieu urbain (où la densité d'antennes est supérieure), ou en milieu rural. Plusieurs techniques existent : Différence de temps observée ou EOTD, Temps d'arrivée (time of arrival), Angle d'arrivée (angle of arrival).

Aujourd'hui, la méthode GSM la plus utilisée est celle du Cell ID (identification de la cellule radio). Cette méthode consiste à récupérer les identifiants des antennes GSM auxquelles le terminal est connecté. Par la suite, grâce à une base de données faisant le lien entre les identifiants des cellules et les positions géographiques des antennes, le terminal est capable de déterminer sa position et d'émettre une estimation. [2]

2.4.3 Géolocalisation par Wi-Fi

Les systèmes de géolocalisation discrets basés sur Wi-Fi fonctionnent par comparaison d'empreintes des puissances des signaux reçus (RSSI Received Signal Strength Indication) avec une base de données de mesures préalables. On appelle cette base de données la cartographie des puissances, qui est composée de données de référence aussi appelées points de référence. Puis, ils calculent la position du terminal mobile par trilatération. La précision de localisation dépend de la densité des points d'accès, mais des difficultés techniques peuvent apparaître en fonction des contraintes de l'environnement (ex: interférence avec les infrastructures métalliques).

2.4.4 Géolocalisation par RFID

RFID correspond à l'abréviation du terme anglais Radio Frequency Identification. C'est un système d'identification automatique par radio fréquence apparu dans les années 1950 mais dont l'émergence est relativement récente. Cette technologie fonctionne par émission d'un champ magnétique selon une fréquence déterminée à partir d'un lecteur RFID puis réception de ce champ par une étiquette électronique (Tag RFID) composée d'une antenne couplée à une puce électronique ou non.

2.5 Solutions proposées par des Start-up

2.5.1 Fidzup, solution basée sur l'envoi de signaux sonores

Fidzup est une Startup fondée en 2011 innovante spécialisée dans le développement et la commercialisation d'outils mobiles de marketing et publicité en point de vente.

Fidzup a développé une technologie de géolocalisation intérieure basée sur l'envoi de signaux sonores. Emis par une source audio dans le magasin (typiquement des enceintes diffusant de la musique ou une TV déjà installée) et inaudible à l'oreille humaine, le signal est reçu par le micro du Smartphone.

Une fois de plus, c'est le SDK de la solution qui permet de l'intégrer à l'application iPhone ou Android du distributeur afin qu'elle puisse décoder et traiter les données reçues via le micro pour en déduire la position du smartphone. L'utilisateur devra simplement avoir accepté que l'application utilise le micro de son téléphone (qui n'enregistre rien). Enfin, la plateforme en ligne de Fidzup permet au distributeur de paramétrer ses campagnes d'envoi de contenu sur le mobile du consommateur (promotion, instants gagnants, chèques cadeaux,...). [3]

2.5.2 Pole Star, un pionnier de la localisation Wi-Fi

Créée en 2002, Pole Star est une start-up toulousaine, elle se revendique comme un pionnier du positionnement indoor en France. Son service NAO Campus est déjà actif dans des aéroports, centres commerciaux, espaces d'exposition et grands magasins dans une vingtaine de pays.

Il fonctionne en combinant signaux GPS, signaux Wi-Fi et données cartographiques pour localiser notamment les Smartphones Android, iPhone et Windows Phone. Pour s'adapter à certains lieux ou usages, Pole Star ajoute que sa technologie ne nécessite qu'une faible densité de balises Wi-Fi pour fonctionner, ce qui permet de déployer rapidement le dispositif. La suite logicielle qui accompagne NAO Campus s'intègre dans l'application mobile du distributeur pour alimenter ses services mobiles de données géolocalisées. [3]

2.5.3 Insiteo combine le Wi-Fi au BLE

Insiteo est une start-up française, qui développe et vend à travers le monde une technologie innovante de géolocalisation « indoor ».

Se décrivant comme un opérateur de services géolocalisés indoor, Insiteo a développé une série de services géolocalisés. En particulier, sa solution de localisation des smartphones.

Android et iOS repose principalement sur les signaux Wi-Fi, éventuellement associés à d'autres technologies comme le BLE. Comme chez Pole Star, les bornes Wi-Fi existantes peuvent être prises en compte par la solution pour limiter le nombre de bornes supplémentaires à installer.

La solution d'Insiteo compare la signature Wi-Fi du lieu que reçoit le smartphone avec une liste de positions préétablies afin de déterminer celle qui correspond à la localisation du smartphone, avec une précision de 2 à 3 mètres. Une fois les bornes installées et leur signature mesurée en tout point du magasin, il n'y a plus qu'à intégrer la solution dans l'application mobile du distributeur via l'API dédiée d'Insiteo. [3]

2.5.4 Ticatag se spécialise dans le Bluetooth Low Energy

Créée en Juin 2013 à Lannion, fruit d'un essaimage d'Alcatel-Lucent

Ticatag se spécialise dans la technologie bluetooth low energy (BLE). Basée à Lannion et fondée par des anciens d'Alcatel, la startup produit et commercialise des boîtiers qui communiquent avec les smartphones équipés BLE 4.0 (iPhone 4S et plus, Galaxy S3, Galaxy S4).

Conçu pour l'intérieur des points de vente, le ti'Beacon dispose d'une autonomie de 3 ans. Destiné à être placé en extérieur, par exemple pour inciter les consommateurs à entrer dans le magasin, le ti'Beacon Solar est énergétiquement autonome grâce à un panneau solaire.

Côté marchand, Ticatag a créé un kit de développement logiciel (SDK) qui permet d'intégrer sa solution aux applications iOS et Android du distributeur. Celles-ci peuvent ensuite utiliser ces données de localisation pour déclencher diverses actions. Une interface Web permet également d'administrer les bornes, d'afficher les données et de paramétrer les actions de géofencing. [3]

2.5.5 Uбудu, des bons plans aux bornes BLE

L'agence mobile Uбудu, initialement spécialisée dans la recherche de bons plans locaux, a depuis lancé uBeacon. Il s'agit de bornes communicantes BLE à placer à des emplacements

stratégiques en magasin, et qui s'accompagnent d'un SDK pour connecter le système à l'application mobile du marchand, ainsi que d'une interface Web lui permettant de gérer les interactions entre bornes et smartphones des visiteurs (portée active, horaires, catégorie, message push, action à déclencher sur le smartphone...). Ces boîtiers ont la taille d'une clé usb et sont rechargeables à l'issue d'une autonomie de plusieurs mois. Le SDK d'uBeacon s'intègre sur Android, iPhone et Windows Phone, ce qui permet de pousser des messages de l'enseigne, lorsque le visiteur s'approche de la borne BLE. [3]

2.5.6 Tableau récapitulatif des techniques de géolocalisation

Le tableau ci-dessus regroupe l'ensemble des techniques de géolocalisation citées auparavant.

Caractéristiques/ Technologies	Portée	Utilisation	Coût	Consommation	Autres contraintes
GPS	-	Géolocalisation triangulaire (3 satellites nécessaires minimum)	variable	Faible consommation d'énergie	- Limitation en terme de traçabilité en intérieur - Limitation due aux multi-trajets
WIFI	< 100 m	Localisation en utilisant puissances des signaux reçus	Infrastructure peu coûteuse et souvent déjà disponible	-	très sensibles aux interférences
RFID	-Actif ~100 m -Passif jusqu'à 10 m	Marquage d'objets (container, etc.) contrôle d'accès	0,5 € (puces passives) à 20 € (puces actives)	Faible consommation	Limitations physiques (lecture à travers le métal)
GSM	-	Calculer la distance et récupérer les identifiants des antennes GSM auxquelles le terminal est connecté	coût élevé des infrastructures	-	couverture géographique, l'accès au réseau GPRS pour exploiter l'information
Bluetooth	<100 m	géolocaliser des tags bluetooth (identifiant unique)	Coût faible	faible consommation	- Consommation élevée pour les longues distances
WIMAX	Qlq dizaines de kilomètres	Sécurité maritime Couvertures classiques de réseau	Coût d'équipement élevé	-	-bande passante 3.5 GHz

Tableau 2. Comparatif des technologies de géolocalisation

Conclusion

Nous avons exposé au début de ce chapitre le concept de la géolocalisation, ainsi que la nécessité de proposer des solutions technologiques qui peuvent répondre à des besoins du marché ciblé. Ensuite, nous avons dressé une liste non-exhaustive de start-up présentes dans la littérature selon les technologies utilisées.

Une étude technologique et un choix technique des composants feront l'objectif du chapitre suivant.

Chapitre 3

Choix technologiques, Protocoles de communication

Ce chapitre met la lumière sur la technologie adoptée et s'intéresse à une étude et choix technique des composants à utiliser afin de mettre en œuvre la solution de géolocalisation proposée.

3. Choix technologiques, Protocoles de communication

3.1 La solution proposée

Toutes les techniques de géolocalisation ne sont pas adaptées à tous les milieux et une étude prenant en compte tous ces éléments est nécessaire afin de dimensionner correctement le système de géolocalisation à mettre en place. Le problème de couverture des réseaux est aussi un frein quant à la bonne géolocalisation des personnes et équipements en toutes circonstances. Une combinaison de plusieurs solutions technologiques permettra de garantir une couverture du service de géolocalisation plus importante que celle obtenue par chacune des technologies individuellement, avec une meilleure qualité sur l'estimation de la position du localisé.

Notre solution propose de combiner la nouvelle technologie Bluetooth Low Energy (BLE) avec les *Motion Sensors* (Accéléromètre et gyroscope) en indoor et le GPS en Outdoor. Cette combinaison conduit à une localisation disponible à tout moment, et en tout lieu avec une précision inférieure à un mètre.

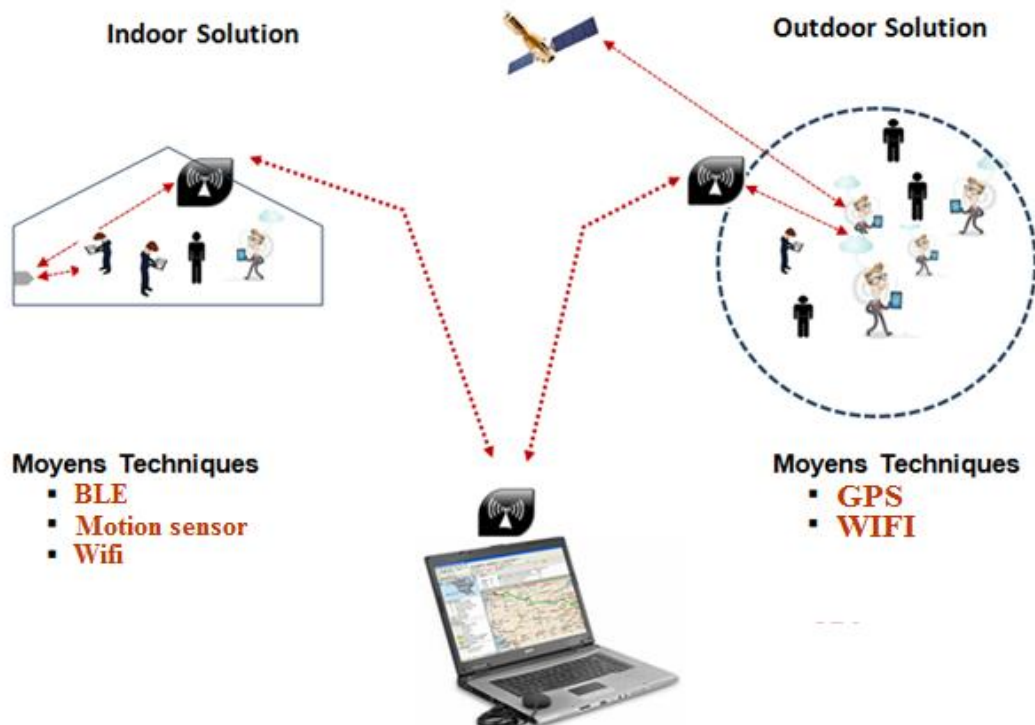


Figure 4. Schéma illustratif de la plateforme GeoComm

3.2 Principe de la solution proposée

Le choix de la méthode de localisation dépend, premièrement, de la précision que nous voulons avoir afin de permettre au système GeoComm d'être fiable du point de vue de la détection d'évènements et du suivi de la personne, et deuxièmement, des contraintes liées aux technologies utilisées (GPS, BLE, *Motion sensors*) et leur adaptation à l'environnement instrumenté. Pour assurer la fonction de géolocalisation, nous avons proposé une solution hybride qui utilise les informations suivantes :

- Détecter les signaux satellites par des puces GPS en outdoor ;
- Détecter les signaux émis des bornes BLE pour pouvoir déterminer la localisation en indoor ;
- Exploiter les données issues de l'accéléromètre et du gyroscope.

La localisation en **Outdoor** se fait par un récepteur GPS qui détecte les signaux satellitaires et génère une donnée (trame) contenant toutes les informations de localisation (altitude, longitude, date).

Pour la localisation en **Indoor**, la première partie de la localisation indoor se fait par l'identification des balises BLE fixes. Cette méthode consiste à récupérer l'identifiant de la balise à laquelle le module GEO est plus proche .En effet, cette information de présence peut être donnée par simple comparaison de la puissance (RSSI) émise par la balise.

L'idée est de détecter l'utilisateur par une seule balise BLE fixe et d'en déduire sa position, en considérant que la personne se trouve dans la même zone que la balise ayant le RSSI le plus élevé.

La figure 5 illustre la localisation d'une personne identifiant la balise à laquelle est plus proche. [5]

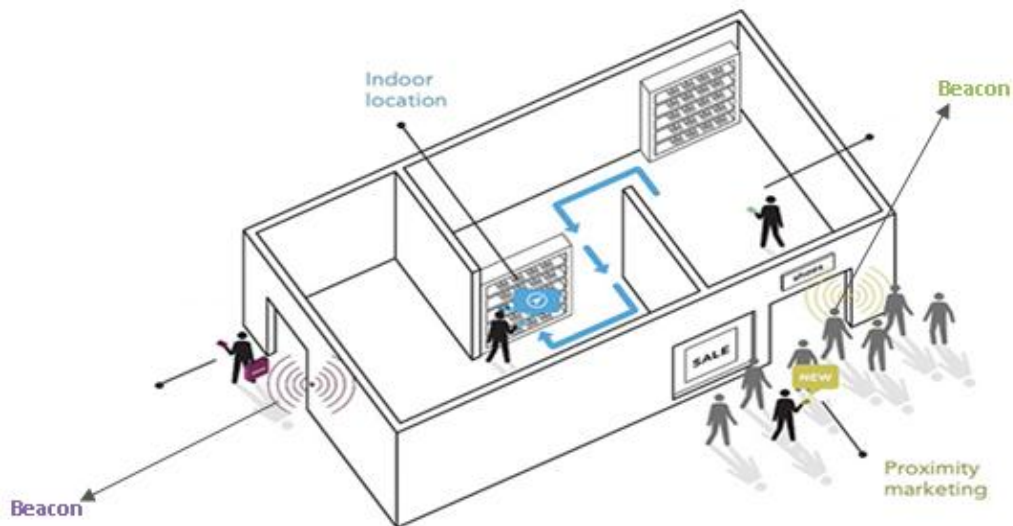


Figure 5. Schéma illustratif de l'identification des Beacons BLE

La deuxième partie de la localisation Indoor consiste à collecter des informations sur le comportement du porteur de l'équipement et déterminer sa position en effectuant des intégrations successives sur des données issues de l'accéléromètre et du gyroscope pour faire le suivi de son mouvement. (Figure 6)

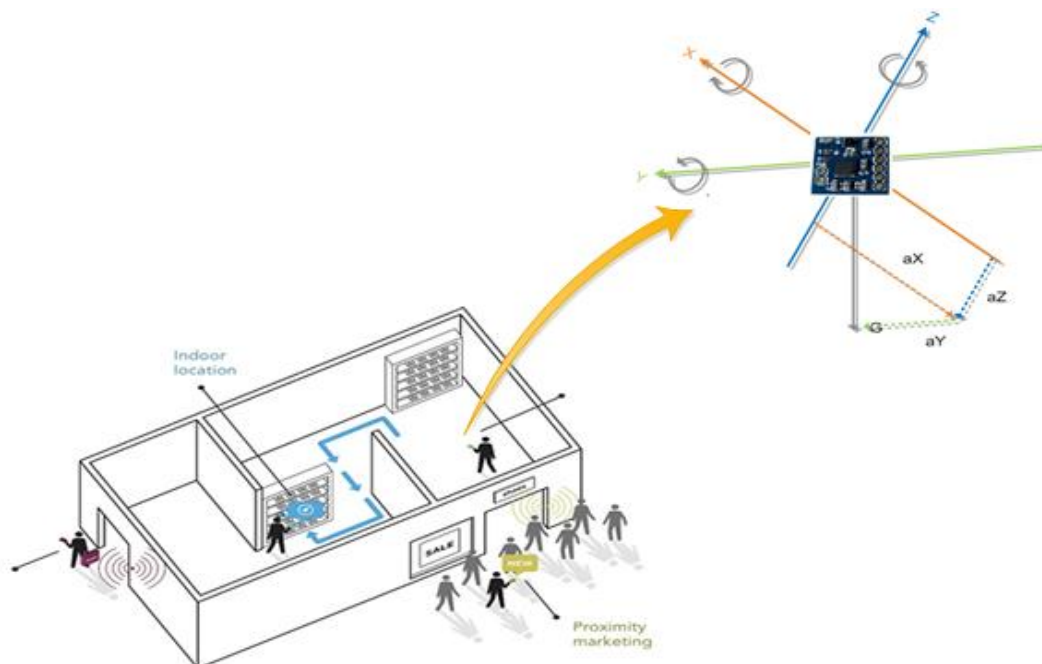


Figure 6. Schéma illustratif du suivi des motions Sensors

Ensuite les données de localisations acquises sont envoyées via le Wifi ou la 3G vers un serveur capable de recevoir, stocker et traiter les informations pour afficher la position sur un fond cartographique adapté. (Figure 7)

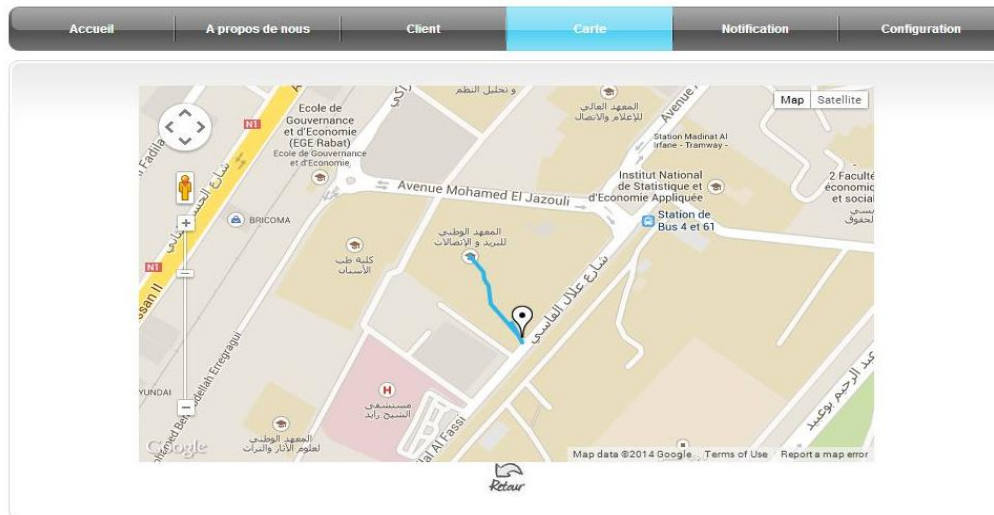


Figure 7. Affichage des coordonnées sur l'application centrale

3.3 Étude et choix technique des composants










Le module GEO se présente comme un système de géolocalisation contenant les ressources suivantes : microcontrôleur, Wifi, GPS, BLE, Accéléromètre et Gyroscope.

Les choix technologiques des composants, les stratégies de contrôle de qualité et les règles de conception des équipements sont des facteurs importants de maîtrise de la fiabilité de GEO.

Afin d'identifier les besoins matériels nécessaires pour réaliser ce module, une étude technique pertinente et exhaustive de la spécification des composants a été faite pour définir les capacités des éléments du module GEO en considérant les critères suivants :

- Contraintes de consommation
- Portée et Sensibilité
- Interface de communication
- Exploitabilité des modules
- Qualité et Coût
- Contraintes de programmation (bibliothèques prédéfinies)
- Disponibilité sur le marché

Après un Benchmarking sur les modules existant sur le marché, une première sélection des composants a été faite. Elle est résumée dans le tableau 3. [6]

Type	Image	Référence	Fabricant	Alimentation	Courant	Tx Puissance	Sensibilité	Interface	Dimension	Support	Portée	Prix	Distributeur
BLE		BLE112-A-v1 Bluetooth Smart Module	Bluegiga Technologies	2.0 – 3.6 V	Transmit: 27 mA Receive: 19.6 mA Sleep mode: 0.5 uA	4 dBm to -23 dBm	-187 to -93dBm	UART, I2C, SPI	15.75 x 9.15 x 2.1 mm	Oui connector	4 dbm: 100 m 0dbm: ~30m -23 dBm: ~5	11,16	Mouser Electronics référence : 603-BLE112-A
		LBCA2ZZVZE BLE Murata	Komatsu Murata Manufacturing	2.2 to 3.6 V	TX/RX peak current: 18mA sleep mode:1uA	-2dBm	-	UART	20.0 x 13.0 x 2.4 mm	Oui avec board-to-board connector	-	-	-
		BR-XB-LE4.0-S2	BlueRadios	2.0-3.6 V	27mA0dB TX, RX down to 19.6mA	-23dBm to 4dBm	-96/-90 dBm	UART, SPI, USB	27.6x24.4 mm	Oui	150 meter LOS	\$49.00/unit	-
		BR-LE4.0-S3A	BlueRadios	2.0-3.6 V	18.2mA0dB TX, RX down to 17.9mA	-23dBm to 4dBm	-96/-90 dBm	UART, SPI, I2C	11.8 x 17.6 x 1.9 mm	-	100 meter LOS	quantity : 10 \$13.50	-
		ABBTM-NVC-MDCS71	ABRACON	1.8- 3.6 V	~16mA peak	4dBm	-92.5dBm	UART, SPI, I2C	19.50x12.50x2.4mm	Oui	-	\$12.9	Digi-Key part number : 535-12377-ND
		ENW-89835A1KF	Panasonic	2.0 - 3.6 V	Transmit Mode 14 mA Receive Mode 14.7 mA	0 dBm	(-) 96 dBm	I2C	14.5x 8.2 x 3 mm	Oui	-	11,49	Mouser Electronics référence : 667-FNWR89835A1KF
Type	Image	Référence	Fabricant	Alimentation	Courant	Sensibilité	Interface	Dimension	Support	Prix	protocole pris en charge		
Accéléromètre		MMA7361L	Freescale Semiconductor	2.2 V – 3.6 V	400 uA	800 mV/g @ 1.5g	Digital	3mm x 5mm x 1.0mm LGA-14 Package	Oui	\$9.00			
		GY-521 6DOF MPU6050 Module 3 Axis Gyroscope + Accelerometer		3~5V	-	-	16bit data output	2.0 cm x 1.6 cm x 0.3 cm	Oui	\$10.63	standard IIC communication protocol		
		LIS302DL		2.16 V to 3.6 V	-	-	I2C/SPI digital output interface	23mm x 12.7mm	Oui	\$11.98			

Type	Image	Référence	Fabricant	Alimentation	Interface	Support	Prix	Distributeur
Gyroscope		Triple-Axis Digital-Output Gyro ITG-3200 Breakout	sparkfun	2.1V to 3.6V	I ² C	Oui	\$24.95	sparkfun, amazon smile
		Gyro Breakout Board - LPY503AL Dual 30°/s	sparkfun	2.7 to 3.6V	I ² C	Oui	\$29.95	sparkfun
		Tri-Axis Gyro Breakout - L3G4200D	sparkfun	2.4 V to 3.6 V	I2C, SPI	Oui	\$49.95	sparkfun
		Accéléromètre et gyroscope 3 axes MPU6050	sparkfun	2,3 to 3,4 V	I2C	Oui	\$39.95	sparkfun

Type	Image	Référence	Fabricant	Alimentation	Tx Puissance	Sensibilité	Interface	Support	Portée	Prix	Distributeur	Protocol prise en charge
Wifi		XB2B-WFUT-001	Digi International	3.14 V to 3.46 V	Up to +16 dBm	-93 to -71 dBm	SPI, UART	NON	up to +100m	28	-	802.11 b/g/n
		RN-131-EK	Microchip	2 to 3 V	+18dBm	(-)85dBm	UART	OUI	-	79.95 USD	-	802.11 b/g/
		WF111-A	Bluegiga	1.8 V, 3.3 V	17 dBm	-91 dBm	CSPI, SDIO, UART, USB	NON	300-500m	21	-	802.11 b/g/n
		WRL-3000 (CC3000 Wifi Module from TI)	EMBEDDED ADVENTURES	2.7 V to 4.8 V	18 dBm	-	SPI	OUI	-	\$ 27.35	EMBEDDED ADVENTURES	802.11b/g

Type	Image	Référence	Fabricant	Alimentation	Courant	Sensibilité	Interface	Dimension	Support	Prix	protocole pris en charge
Gps		ST32	SkyTraq Venus 6	3.3 V to 5.5 V	-	(-)161 dBm Tracking	RS232 ,UART	32 x 32 x 8mm	Connector type: pin header 1.27mm x 8	\$16.00	NMEA-0183 (GGA, GSA, GSV, RMC, VTG), Skytraq Binary
		A2235-H	Maestro Wireless	3 V to 3.6 V	22mA	(-)163 dBm tracking	SPI, UART	17.8 x 16.5 x 7.1 mm	NON	\$11.54	
		VK16U6	ublox			(-)162dBm tracking	UART / TTL, 232 level, USB2.0 optional interface	28 * 28 * 8.4mm	Oui	\$20.00	NMEA 0183 V3.0

Tableau 3. Liste de sélection des composants

3.4 Synthèse de l'étude

Après la présélection des composants figurant sur le tableau ci-dessus. Une comparaison a été faite afin de choisir les composants qui vont être utilisés par la suite. Selon les critères suivants :

- Disponibilité sur le marché : trouver le distributeur du produit
- Prix : raisonnable
- Exploitabilité : choisir des composants avec un support ou avec des pins qui facilite le soudage
- Antenne intégré

3.4.1 Le récepteur : GPS ST32 Skytraq

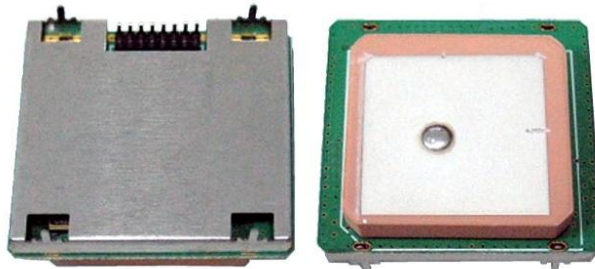


Figure 8. Récepteur GPS ST32 Skytraq

a. Présentation :

Le GPS est un système de positionnement par satellite capable de donner une position n'importe où sur le globe, instantanée, en temps réel, avec une précision de quelques mètres, de jour comme de nuit, et quelles que soient les conditions météorologiques.

Le ST32 est un module GPS à haute sensibilité et très faible consommation. Il est basé sur le processeur de bande de base de VENUS634 Skytraq et il est équipé d'une antenne patch adapté pour fournir une solution modulaire. Cela permet à une intégration facile dans tous les types d'applications ou de produits de navigation [7]. Il a comme caractéristiques :

- Haute sensibilité :-161 dBm
- Voltage : 3.3-5.5V
- Faible consommation d'énergie : 25mA tracking, 50mA acquisition, 70mA enhanced acquisition.
- Dimension 32 x 32 x 8mm (with patch antenna)
- Interface de communication RS232 and TTL UART

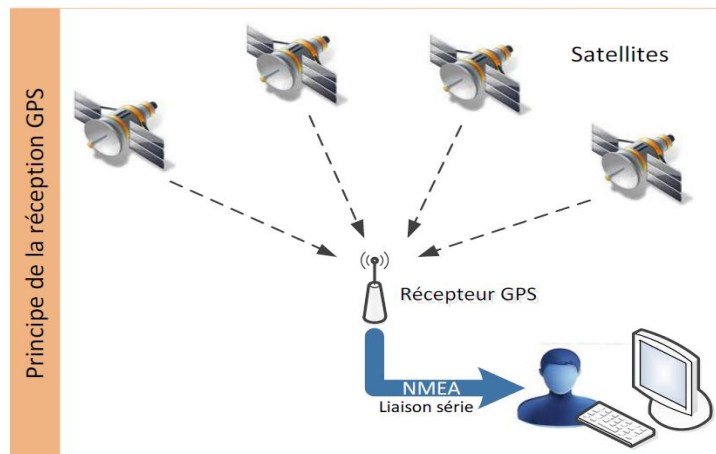


Figure 9. Principe de la réception GPS

b. Principe de fonctionnement

Le récepteur GPS est capable de se géolocaliser grâce à la réception de signaux émis par des satellites géostationnaires. Le récepteur GPS détermine par calcul sa position et peut la transmettre sous forme d'une trame NMEA. Le calcul de la position est effectué par le récepteur qui fabrique une trame de caractères NMEA. Cette trame est ensuite envoyée via une liaison série. L'ordinateur décode la trame et affiche éventuellement la carte correspondant à la position GPS reçue.

→ La trame NMEA

Une trame NMEA (National Marine Electronics Association) est une suite de caractères contenant des informations de géolocalisation comme :

- La latitude, la longitude,
- La vitesse, l'altitude,
- Le nombre de satellites,
- L'heure, la date

Il existe plusieurs trames correspondant à des besoins différents. Chaque trame possède une syntaxe différente. Dans notre Projet Nous nous intéresserons à la trame la plus utilisée pour connaître la position courante du récepteur : La trame GGA.

Une trame est constituée de champs. Les champs sont séparés entre eux par des virgules. Un champ peut être vide mais la présence de la virgule est obligatoire.

- Exemple de trame GGA :

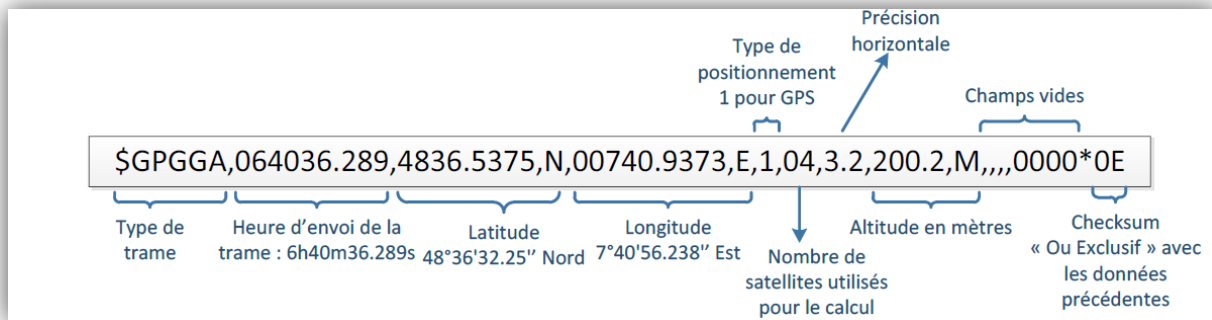


Figure 10. Exemple trame GGA

3.4.2 Le module Bluetooth low energy BLE112



Figure 11. Module BLE Bluegiga

a. Présentation :

Le BLE est une nouvelle technologie qui fournit une liaison sans fil, nouvelle pile de protocole, nouveau profil architecture et un nouveau régime de qualification. Technologie conçue et commercialisée par le Bluetooth Special Interest Group visant à de nouvelles applications dans les soins de santé, remise en forme, industries, de sécurité et de divertissement à domicile. Par rapport à "Classic" Bluetooth, Bluetooth intelligent est destiné à fournir de réduire considérablement la consommation d'énergie et les coûts tout en maintenant une portée de communication similaire.

BLE112, Bluetooth Low Energy est un dispositif ciblé pour les capteurs et les accessoires de faible consommation, BLE112 offre toutes les fonctionnalités du Bluetooth low energy : radio, la pile, les profils et les applications client, pas de processeur externe est nécessaire.

Le module fournit également des interfaces matérielles flexibles pour le connecter avec d'autres capteurs. [8, 9]

→ **Caractéristiques**

- Supporte les modes Master et Slave
- Basé sur le TI CC2540 chip.
- TX power: +3dBm to -23dBm
- RX sensitivity: -85dBm to -91dBm
- Courant de transmission: 27mA (0dBm)
- Modulation: GFSK
- Antenne intégrée
- Dimension 15.75 x 9.15 x 2.1 mm
- Interface de communication : UART, USB ou SPI
- Portée: +4dbm/-93dbm, line-of sight: ~100 meters
+0dbm/-88dbm, line-of sight: ~30 meters
-23dbm/-88dbm, line-of sight: ~5 meters

b. Principe de fonctionnement:

Le Bluetooth Low Energy joue quatre rôles importants :

- **Advertiser** : diffuse les paquets de publicités, mais il n'est pas capable de les recevoir.
- **Scanner** : Écoute les paquets diffusés par les advertisers. Et essayer de se connecter avec l'advertiser.
- **Master** : Un périphérique maître qui est connecté à un ou plusieurs esclaves
- **Slave** : Un périphérique qui est relié à un maître. Ne peut être connecté à un maître à la fois

Les paquets de publicité contient généralement des informations comme:

- Modes de découverte et de connectabilité
- L'adresse de l'Advertiser
- Niveau de puissance TX
- Services pris en charge (profils)
- Les données d'application
- Nom du périphérique

Pour utiliser la technologie BLE, un périphérique doit être capable d'interpréter certains profils. Ces derniers sont les définitions d'applications possibles qui permettent de préciser les comportements généraux que les appareils compatibles Bluetooth utilisent pour communiquer avec d'autres périphériques Bluetooth. Il y a un large éventail de profils Bluetooth décrivant de nombreux types d'applications ou de cas d'utilisation pour les périphériques. Les développeurs peuvent créer des applications pour travailler avec d'autres périphériques Bluetooth.

→ *Format de paquet*

Bluetooth à faible consommation d'énergie a un format de paquet générique utilisé à la fois pour la publicité et des paquets de données. La principale caractéristique de la pile faible consommation d'énergie est la couche de liaison (LL) qui fournit un mode de fonctionnement ultra-faible consommation.

La figure suivante montre le format de paquet :

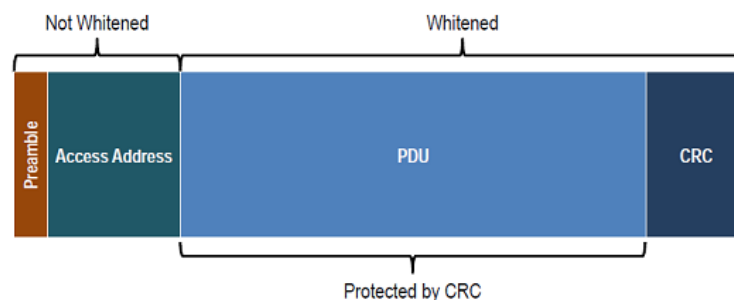


Figure 12. Format de paquet générique du BLE

Le paquet BLE comprend:

- Préambule de synchronisation
- Access address (4 octets): Pour l'identification de la liaison physique sur chaque paquet pour chaque esclave
- PDU (Protocol Data Unit) de longueur variable (de deux à 39 paquets) :

L'en-tête de PDU à 16 bits, l'adresse de l'appareil et jusqu'à 31 octets d'informations

- 3 octet (24 bit) cyclic redundancy code(CRC) : Le CRC assure l'exactitude des données dans la PDU sur tous les paquets, ce qui augmente la robustesse contre des interférences

→ *Pile protocolaire :*

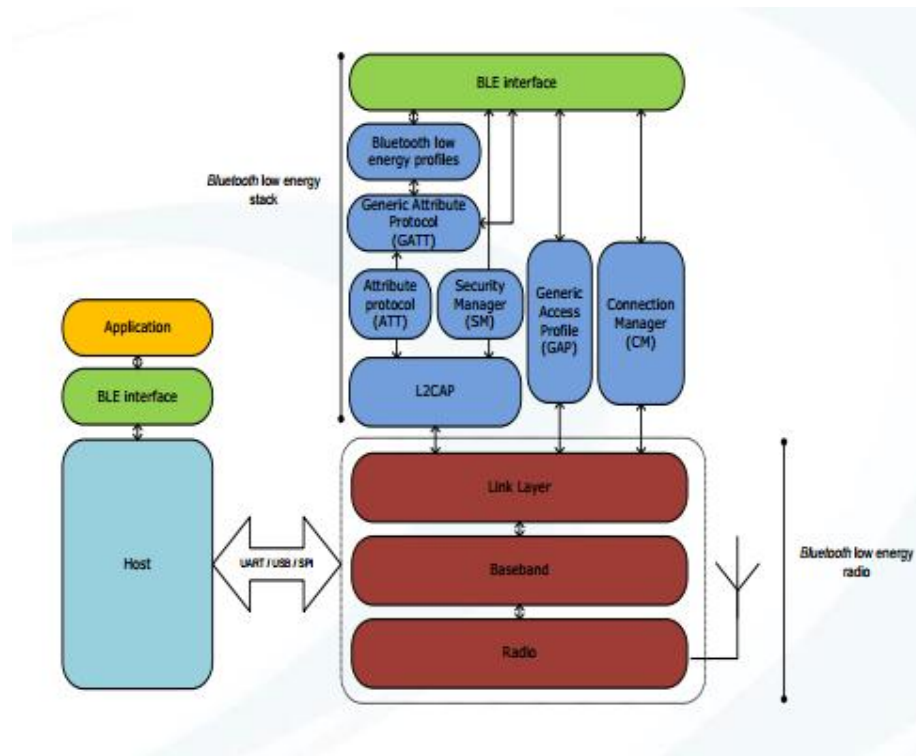


Figure 13. Pile protocolaire du BLE112

- **BGAPI** : Une API binaire entre l'hôte externe et la pile
- **Profiles** : Données spécifiques d'application
- **GAP** : Generic Access Profile définit les modes : découverte d'autres périphériques, connections
- **GATT** : Generic Attribute Profile pour l'organisation des données
- **ATT** : Attribute Protocol : protocole d'accès aux données
- **L2CAP** : Protocole de multiplexage/ Fragmentation et assemblage des paquets
- **HCI** : interface entre l'hôte et le contrôleur
- **Link Layer** : paquets et contrôle radio
- **Physical layer** : la couche physique qui permet la transmission/réception des bits

3.4.3 Le module MPU-6050 Motion Processing Unit



Figure 14. Module MPU6050 Accéléromètre+Gyroscope

a. Présentation

L'accéléromètre est un capteur qui permet de connaître l'accélération linéaire selon trois axes X, Y et Z. Bien que l'accélération linéaire soit définie en m/s^2 (SI), la majorité des documentations sur ces capteurs exprime l'accélération en « g » (la gravitation terrestre, soit environ $g = 9,81 m/s^2$).

Le gyroscope est un capteur inertiel qui permet d'évaluer la rotation angulaire autour d'un axe. Un gyroscope à 3 axes mesure la rotation autour chacun des axes X, Y, Z. Il calcule le nombre de degrés par seconde qui a été faite en rotation.

Le dispositif MPU-6050 combine un gyroscope 3 axes et un accéléromètre à 3 axes sur le même silicium avec un bord Digital Motion Processor TM (DMP TM) capable de traiter des algorithmes complexes de MotionFusion 9 axes. Pour le suivi de la précision des mouvements à la fois rapides et lents, le module dispose d'un gyroscope et un accéléromètre programmable par l'utilisateur avec une plage à pleine échelle [10]

- Alimentation: 2,3 à 3,4 Vcc
- Consommation: 3,9 mA maxi

→ **Caractéristiques Gyroscope :**

- Digital-output X-, Y-, et Z-Axis angular rate sensors (gyroscopes) programmable avec une plage à pleine échelle ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/sec$
- tolérance de calibration: $\pm 3\%$
- Interface I2C

→ **Caractéristiques Accéléromètre :**

- Digital-output triple-axis accéléromètre avec une plage programmable à pleine échelle $\pm 2g$, $\pm 4g$, $\pm 8g$ et $\pm 16g$
- tolérance de calibration: $\pm 3\%$
- interface de communication : I2C

b. Principe de fonctionnement

→ **Accéléromètre**

Le principe de la plupart des accéléromètres est basé sur la loi fondamentale de la dynamique $F = m \cdot a$ (F : force (N), m : masse (kg), a : accélération (m/s^2) aussi notée γ). Plus précisément, il consiste en l'égalité entre la force d'inertie de la masse sismique du capteur et une force de rappel appliquée à cette masse. On distingue deux grandes familles d'accéléromètres : les accéléromètres non asservis et les accéléromètres à asservissement.

Les applications de ce capteur sont très diverses :

- la mesure de vitesse (par intégration)
- la mesure de déplacement (par double intégration)
- le diagnostic de machine (par analyse vibratoire)
- la détection de défaut dans les matériaux (en mesurant la propagation d'une vibration à travers les matériaux)

→ **Gyroscope**

Le gyroscope est un appareil qui exploite le principe de la conservation du moment angulaire en physique (ou encore stabilité gyroscopique ou effet gyroscopique). Cette loi fondamentale de la mécanique veut qu'en l'absence de couple appliqué à un solide en rotation, celui-ci conserve son axe de rotation invariable. Lorsqu'un couple est appliqué à l'appareil, il provoque une précession ou une nutation du solide en rotation. Dans les capteurs : un gyroscope est un capteur de position angulaire et un gyromètre un capteur de vitesse angulaire. Le gyroscope donne la position angulaire (selon un, deux ou les trois axes) de son référentiel par rapport à un référentiel inertiel (ou galiléen).

c. Valeurs retournées par le MPU6050

Notre module MPU6050 retourne des valeurs brutes d'accélération et de rotation il est aussi doté d'un DMP capable de traiter les valeurs émises par l'accéléromètre et le gyroscope.

3.4.4 Le module Wifi WRL-3000



Figure 15. Module Wifi WRL-3000

a. Présentation

Le Wi-Fi (Wireless Fidelity) est un ensemble de fréquences radio qui élimine les câbles, partage une connexion Internet et permet l'échange de données entre plusieurs postes. Le WiFi (ou IEEE 802.11), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres.

La WRL-3000 est un module WiFi pour les systèmes embarqués, basées sur la puce CC3000 de Texas Instrument. Il contient l'ensemble du WiFi et des piles TCP / IP et nécessite relativement des simples commandes via SPI pour le contrôler. [11]

→ **Caractéristiques**

- Supporte la norme Wifi 802.11B/G
- Alimentation : 2.7V
- Peut transmettre plus que 7Mbps
- Interface de communication SPI
- TX power: +18.0 dBm
- RX sensitivity: -88 dBm
- Supporte tous les modes de sécurité wifi : WEP, WPA, et WPA2

b. Principe de fonctionnement

La technologie WI-FI, basé sur la norme 802.11, vous permet de créer un réseau local sans avoir à installer de câbles pour relier les ordinateurs entre eux. Pour mettre en place une solution sans fil, il faut absolument s'équiper d'un **point d'accès Wi-Fi**. Afin de " dialoguer "

avec le point d'accès, chaque machine doit obligatoirement être dotée d'une **carte Ethernet WI-FI** ou d'un **adaptateur USB WI-FI**

c. Sécurité wifi

Pour remédier aux problèmes de confidentialité des échanges sur un réseau sans fil, le standard 802.11 intègre un mécanisme simple de chiffrement de données :

→ Le WEP : Ce cryptage travaille avec l'algorithme RC4 pour chiffrer les données et utilise des clés statiques de 64 ou 128 voire 152 bits suivant les constructeurs.

Le principe du WEP consiste à définir une clé secrète qui doit être déclarée au niveau de chaque adaptateur sans fil du réseau ainsi que sur le point d'accès. La clé sert à créer un nombre pseudo-aléatoire d'une longueur égale à la longueur de la trame. Le serveur qui veut communiquer avec le module WRL3000 doit connaître la clé secrète qui va servir au cryptage WEP.

→ Le WPA, développé par l'IEEE, est un autre protocole de sécurisation des réseaux sans fil offrant une meilleure sécurité que le WEP car il est destiné à en combler les faiblesses.

Il utilise des clés TKIP (Temporal Key Integrity Protocol) - dites dynamiques - et permet l'authentification des utilisateurs, Les clés WPA sont en effet générées et distribuées de façon automatique par le point d'accès sans fil - qui doit être compatible avec le WPA. De plus, un vérificateur de données permet de vérifier l'intégrité des informations reçues pour être sûr que personne ne les a modifiées.

→ WPA-2, Il s'appuie sur l'algorithme du chiffrement TKIP, comme le WPA, mais supporte au contraire l'AES - au lieu du RC4 - beaucoup plus sûr au niveau du cryptage des données. La WiFi Alliance a ainsi créé une nouvelle certification, baptisée WPA-2, pour les matériels supportant le standard 802.11i.

3.5 Protocoles de communication :

De nombreuses applications complexes qui relèvent du domaine de l'électronique ont pu voir le jour, dans leur majorité, ces applications sont développées soit à base de microcontrôleur, de microprocesseur ou encore de circuits logiques complexes. Cependant, l'utilisation de ces composants dans un système, exige un protocole de communication. Dans la se section suivante nous allons présenter brièvement les protocoles utilisés dans notre projet.

3.5.1 Protocole I2C

a. Présentation

Le bus I2C (Inter Integrated Circuit) a été développé par l'entreprise Philips. Ce bus est utilisé pour connecter les composants périphériques microcontrôleur. Le bus I2C est composé de deux fils, ce qui réduit la partie hardware de façon drastique. Ce bus comprend une ligne d'horloge SCL (Serial Clock) et une ligne de donnée SDA (Serial Data). La communication est par conséquent synchrone. Son mode d'utilisation est souvent du type maître et esclave. Cependant, il soutient également le mode multi-maître.

La Figure suivante illustre une application typique avec le bus I2C :

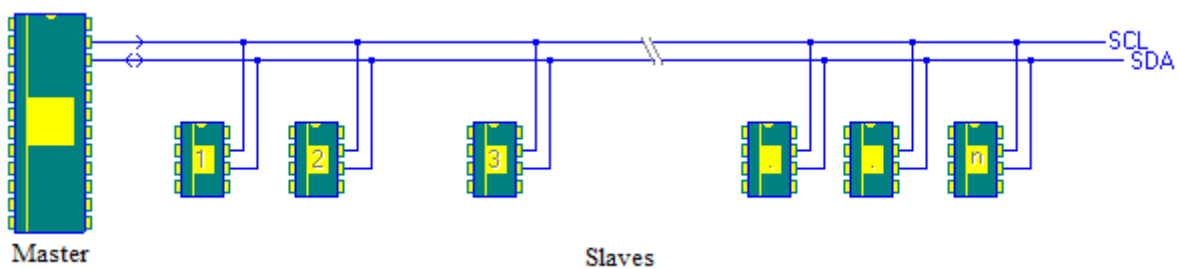


Figure 16. Mode de Fonctionnement I2C

b. Principe de fonctionnement

Le transfert de données entre le maître et l'esclave s'effectue de la manière suivante : Le maître démarre le transfert des données en envoyant le bit **de démarrage**. Tous les esclaves entrent ainsi dans un état passif d'écoute. Le maître envoie ensuite l'**adresse** de l'esclave, avec lequel il aimerait communiquer. Cette adresse est composée de 7 bits.

Le maître envoie le bit **R/W**, qui fixe le sens de la communication : Ecriture depuis le maître à l'esclave ou lecture depuis l'esclave au maître.

Le protocole I2C exige des confirmations (**acknowledge**) après chaque transmission d'un octet. L'esclave accredité avec le premier bit de confirmation, qu'il est prêt pour la communication. Après chaque transmission d'un octet. Ce bit est fourni par le maître dans le mode lecture et par l'esclave dans le mode écriture. Le transfert se termine avec un bit d'**arrêt** (Stop).

3.5.2 Protocole UART :

a. Présentation

C'est un sous-système complet disponible en un seul circuit intégré et conçu pour la transmission et la réception de données asynchrones en mode duplex ou semi-duplex.

b. Principe de fonctionnement

L'émetteur de l'UART accepte les caractères en parallèle et les convertit en une suite binaire sérielle. Le récepteur convertit les données sérielles en données parallèles. En plus de ses fonction de conversion, l'UART remplit également des fonctions de contrôle et de commande. L'UART ajoute aux données transmises le bit de départ (START) et les bits d'arrêt (STOP) En outre, c'est un circuit programmable qui permet de sélectionner :

- la longueur du mot binaire (5, 6, 7 ou 8 bits).
- Le type de parité (paire, impaire ou sans parité).
- Le nombre de bits d'arrêt (1, 1, 5 ou 2).

La figure 17 présente les principaux blocs de l'UART [12]

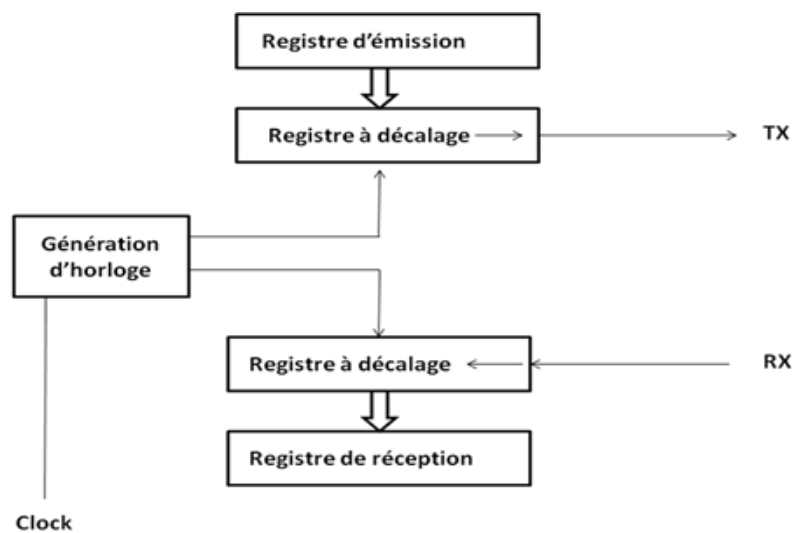


Figure 17. Principaux blocs de l'UART

3.5.3 Protocole SPI

a. Présentation

L'interface SPI (Serial Peripheral Interface) est un bus sériel à haut débit, destiné à la communication entre le microcontrôleur et la périphérie. Ce dernier est souvent utilisé pour la communication avec des extensions d'entrée et de sortie, des affichages LCD ainsi que des convertisseurs A/D et D/A. Il peut également être utilisé pour la communication entre microcontrôleurs.

b. Principe de fonctionnement

La figure 18 illustre le principe de fonctionnement du SPI [12].

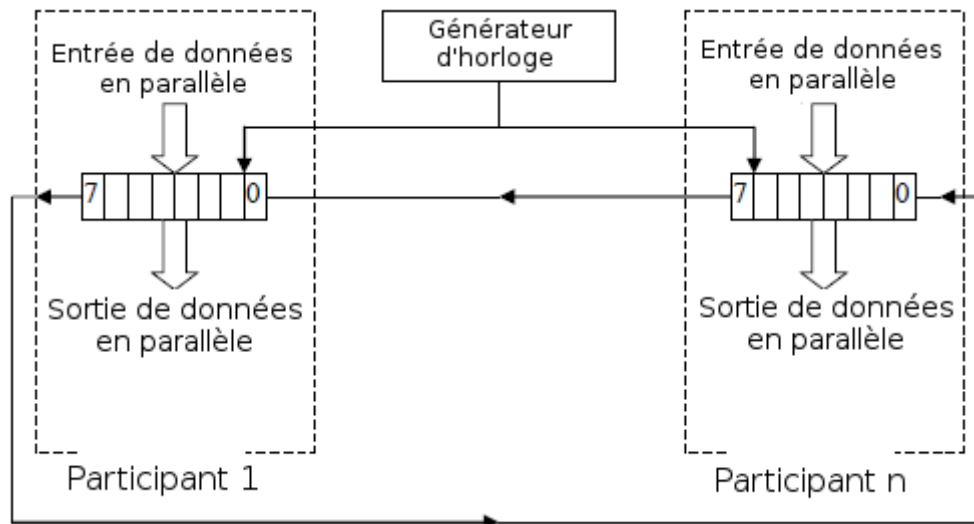


Figure 18. Principe de fonctionnement SPI

L'interface SPI est toujours utilisée en mode maître esclave. Le maître est alors responsable de la génération de la fréquence d'horloge. Le SPI peut travailler de façon duplexe à l'aide de deux lignes de transmission : MOSI (Master Out Slave In) et MISO (Master In Slave Out). Les esclaves peuvent être connectés soit de façon parallèle (c'est à dire que toutes les sorties des esclaves sont rassemblées et connectées à l'entrée MISO du maître) ou de façon sérielle (la sortie d'un esclave est connectée à l'entrée du prochain esclave et la sortie du dernier esclave est connecté à l'entrée MISO du maître). Le microcontrôleur écrit les données à transmettre dans un tampon de transmission. Ces dernières sont sérialisées à l'aide d'un registre à décalage (comme une transmission RS232). Les données reçues sont également converties à l'aide d'un registre à décalage. Le microcontrôleur peut alors lire ces données de façon parallèle dans le tampon de réception. Du fait que les interfaces SPI ont une bande passante relativement élevée, les tampons de transmission et de réception contiennent souvent plusieurs octets.

3.6 Choix du microcontrôleur

Le choix du microcontrôleur dépend d'une part des performances techniques attendues, mais aussi de sa facilité de mise en œuvre. Notre application nécessite d'intégrer les modules BLE112, MPU6050, GPS ST32 et le WRL3000 dont nos besoins sont les suivants :

- Nombre d'entrées/sorties programmables suffisant pour intégrer tous les composants ;
- Une mémoire FLASH permet un effacement et une reprogrammation multiple du circuit ;

- La présence des convertisseurs Analogiques/Numériques CAN ;
- L'existence d'une mémoire EEPROM ;
- La documentation et l'assembleur sont largement disponibles sur les sites Web des constructeurs ;
- L'existence des interfaces de communication : I2C SPI UART ;
- La disponibilité sur le marché local ;
- La facilité de mise en œuvre ;
- Le prix raisonnable ;
- L'approvisionnement multi source.

C'est pour ces raisons que nous avons choisi d'utiliser la carte Arduino à base du microcontrôleur ATmega2560 qui répond à nos besoins.

Conclusion

Dans ce chapitre, nous avons abordé les choix de la technologie et les protocoles de communication visant à établir le dialogue entre les différents composants de notre système ainsi que le choix du microcontrôleur adéquat .Le chapitre suivant sera alors dédiée à l'analyse et la conception de notre module.

Chapitre 4

Analyse et Conception

Une bonne étude conceptuelle est la clé de toute application bien réussie. Elle donne une vision plus claire du produit à réaliser. Le formalisme SysML s'est imposé comme l'outil le plus approprié pour ce projet. En effet, SysML permet de mener la phase de conception tout en bénéficiant de la puissance et de la simplicité de ses diagrammes. Ce chapitre traite en détails la phase conception.

4. Analyse et Conception

4.1 Description fonctionnelle détaillée

4.1.1 Identification des acteurs

- Application centrale : il s'agit d'un acteur de type système, elle interagit directement avec le module GEO afin d'exploiter les données de localisation issue de ce dernier pour le positionner ensuite sur la Map.
- Utilisateur : c'est la personne à localiser, le porteur du GEO.

4.1.2 Diagramme de cas d'utilisation

Le module GEO comporte 3 cas d'utilisation majeurs :

- La localisation de l'utilisateur (porteur) en outdoor/indoor.
- La diffusion des données de localisation vers l'application centrale via Wifi ou 3G.
- L'envoi de l'état du module (fonctionnement ou dysfonctionnement) à son porteur et à l'application centrale.

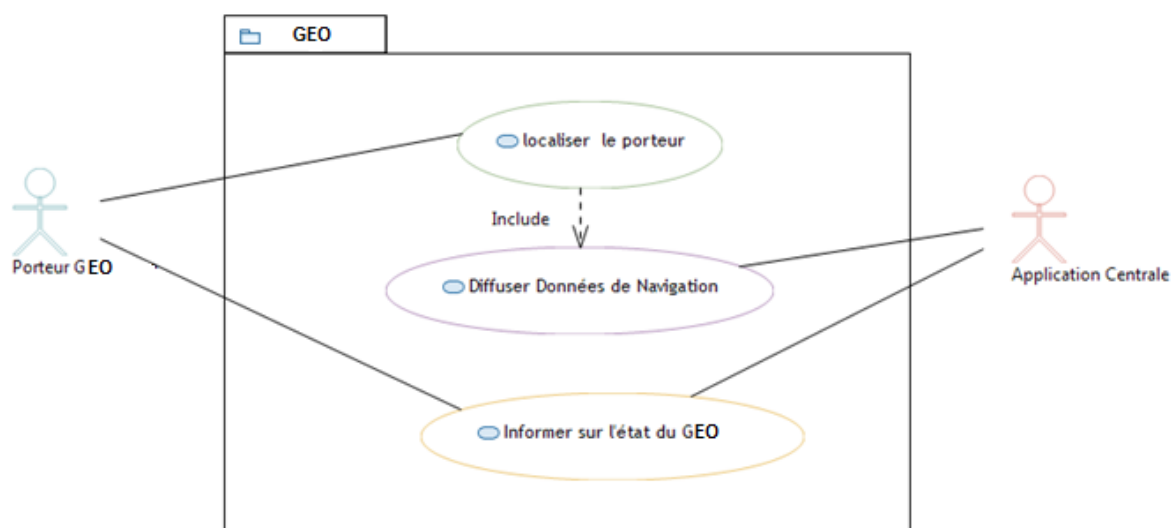


Figure 19. Diagramme de cas d'utilisation généralisé

Ces cas d'utilisation incluent un ensemble de sous-comportements, illustrés dans le diagramme suivant et dont les détails seront traités dans les tableaux 5 et 6:

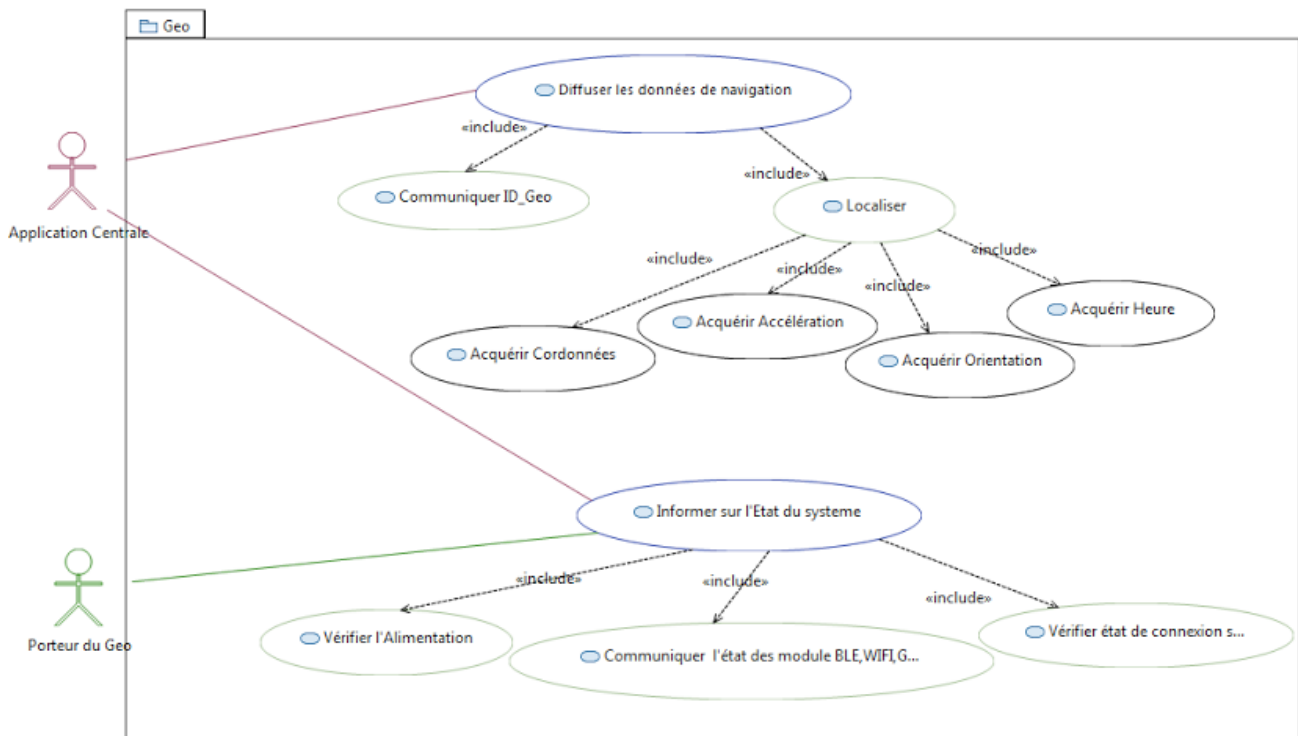


Figure 20. Diagramme de cas d'utilisation détaillé

Les cas d'utilisation « Clé » du système sont détaillés afin de bien cerner les fonctionnalités du système.

Cas d'utilisation	Diffuser les données de navigation
Acteur	Application Centrale
Description	Le module GEO doit communiquer les informations de localisation à l'application centrale afin d'être traitées
Scenario	1-le Module GEO envoie son ID à l'application centrale afin d'identifier l'identité du porteur. 2-le module GEO envoie les coordonnées du porteur, l'accélération, l'orientation et l'heure. 3-l'application envoie au module GEO les données de configuration et de mise à jour.

Tableau 4. Description détaillée du cas d'utilisation «Diffuser les données de navigation »

Cas d'utilisation	Informer sur l'état du système
Acteur	Application centrale, le Porteur du module GEO
Description	Le module GEO doit communiquer l'état de ces composants à la fois au Porteur et à l'application centrale.
Scenario	1-Le module envoie l'état des composants a l'application centrale. 2-une Led sur le module indique l'état du au porteur pour s'assurer qu'il est fonctionnelle. 3-le module vérifie sa connectivité avec la centrale.

Tableau 5. Description détaillée du cas d'utilisation « Informer sur l'état du système »

4.1.3 Etude de comportement du système

Le diagramme de séquence présente les interactions entre les acteurs (Porteur-GEO-CEN) Ces interactions détaillent les messages échangés entre ces acteurs dans un ordre chronologique dans le but d'assurer la fonction principale (géolocalisation)

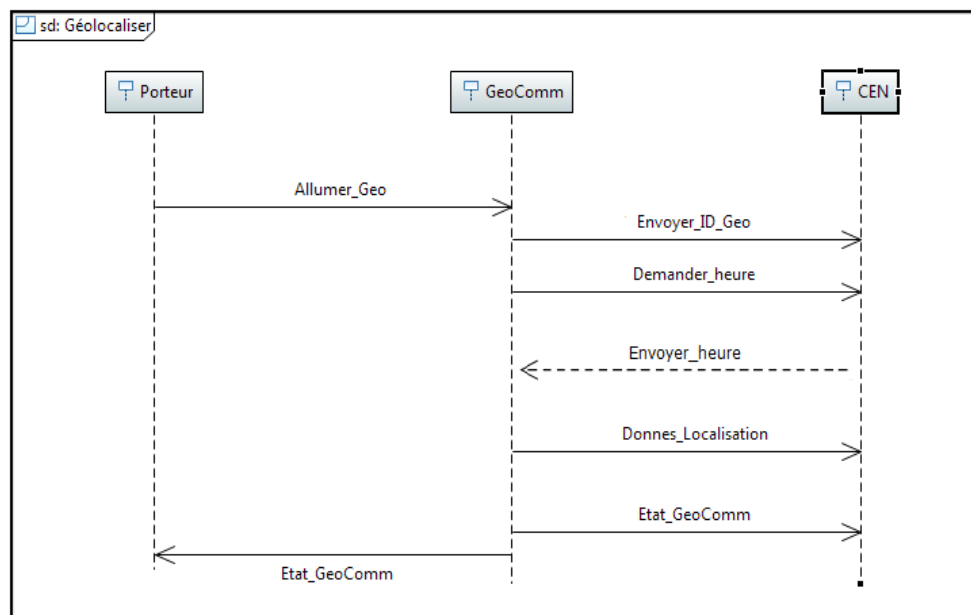


Figure 21. Diagramme de séquence « système »

Les interactions assurant la fonction géolocalisation sont les suivants :

- Le porteur allume le module GEO
- Le module GEO se charge d'envoyer son identifiant à l'application centrale (CEN)
- Le module GEO demande l'heure
- Un message de retour contenant l'heure est renvoyée par la CEN
- Le module GEO envoie les données de localisation
- Le module GEO communique aussi son état avec son porteur ainsi que la CEN

Scénario erreur connexion avec la centrale :

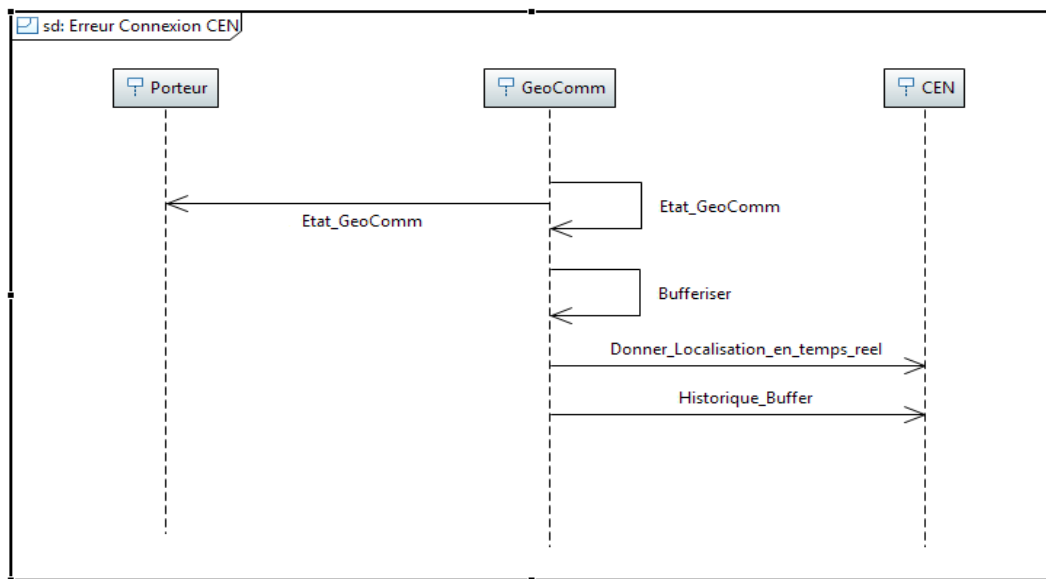


Figure 22. Diagramme de séquence « erreur connexion avec la centrale »

En cas d’erreur de la connexion avec la CEN, le module GEO doit informer le porteur de cette erreur, stocker l’historique de données de localisation dans un buffer et l’envoyer en cas de l’établissement de la connexion avec la CEN.

4.1.4 Diagramme d’états

La figure 23 représente le diagramme d’état de notre module.

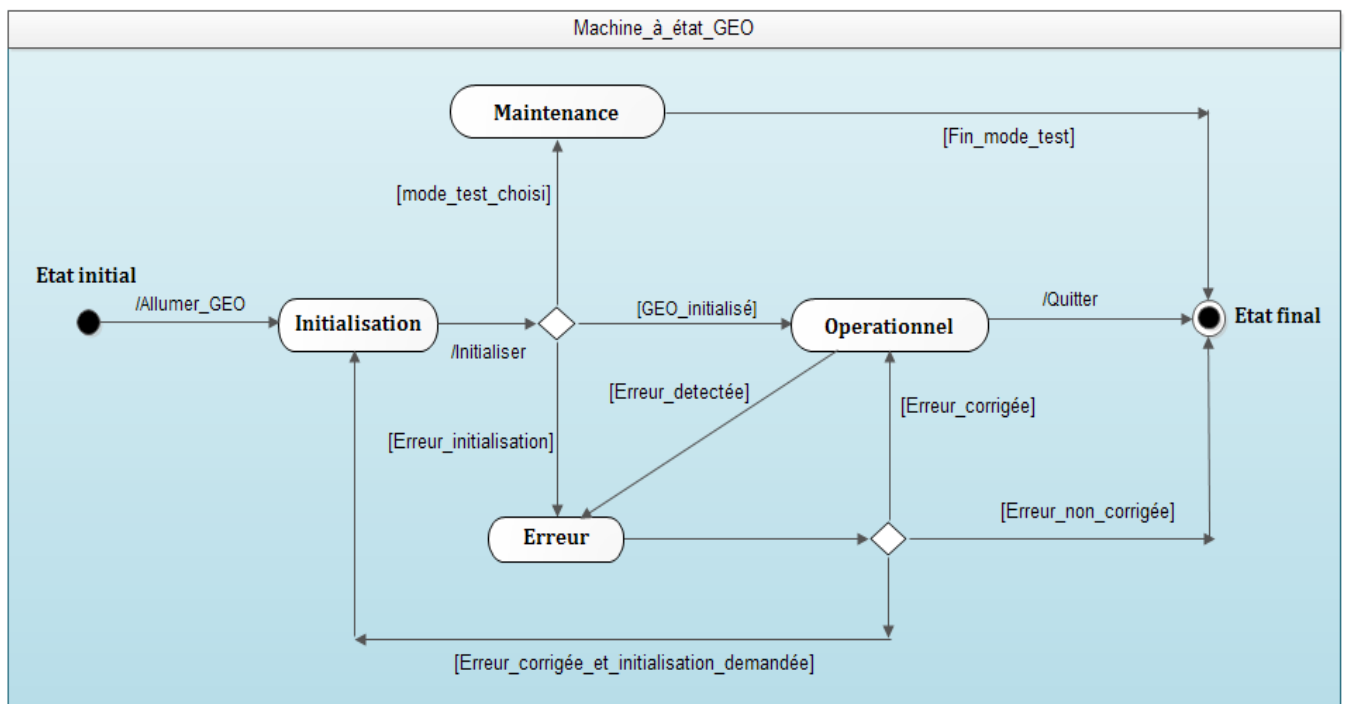


Figure 23. Diagramme d’état du module GEO

Mode 1 : état Initialisation

Mise en marche des modules GPS, BLE, Accéléromètre, Gyroscope et Wifi, et la vérification de l'état des périphériques, la liaison hardware du module ainsi que l'état de la connectivité avec l'application serveur.

Mode 2 : état Opérationnel

Une fois les modules sont activés, fonctionnant normalement et la connexion est établie, les données de localisation du porteur du GEO sont remontés périodiquement (500 ms) à l'application centrale via le module WIFI, en effet dans cet état le système:

- Envoie l'identifiant du module GEO pour identifier son porteur.
- Envoie les données de localisation (longitude, latitude, altitude, date, accélération, vitesse) du GEO
- Afficher son état (activé/désactivé) au client.

Mode 3 : état Maintenance

Ce mode permet de connaître la précision de la fonction de géolocalisation du module en effectuant une série de test sur le système. On peut entrer dans ce mode aussi lors de la mise à jour de notre module.

Mode 4 : état Erreur

Dans ce mode, il est nécessaire de traiter tous les dysfonctionnements susceptibles d'avoir lieu :

- **Dysfonctionnement matériel** : Le système doit être en mesure de détecter les pannes relatives à chaque module (WIFI, 3G, Accéléromètre, Gyroscope, GPS, BLE), d'envoyer à l'application centrale leur nature et signaler au porteur (utilisateur) la fin du service en allumant une LED rouge
- **Dysfonctionnement logiciel** : Lors d'une connexion défailante à l'application centrale

4.2 Modélisation des exigences

4.2.1 Description des performances du composant

a. Capacité du système :

Le composant GEO doit garantir un minimum de services dans tous les cas en dépit des erreurs éventuelles d'exécution ou des défaillances possibles du matériel (Robustesse).

- ✓ **Autonomie** : le composant GEO doit assurer une faible consommation d'énergie pour assurer une autonomie optimale sur batterie rechargeable ou avec pile externe. Ci-dessus quelques hypothèses pour assurer une autonomie du GEO :
 - Autonomie du GEO par la technologie de l'ordonnancement temps réel.
 - technique de gestion dynamique de puissance (Dynamic Power Management : DPM).
 - des techniques utilisant la variation de tension et de fréquence (Dynamic Voltage and Frequency Selection: DVFS).
 - Il fonctionne sur batterie et celle-ci doit être changée après un certain temps
- ✓ **Coût/Qualité** : le composant GEO doit satisfaire les besoins exprimés ou implicites du cahier de charges à moindre coût (localisation avec une précision < 1 mètre)
- ✓ **Ergonomie, dimension, poids** : le composant GEO doit être légère et moins volumineux pour ne pas gêner le localisé qui le porte.
- ✓ **Environnement** : le module GEO doit résister à l'environnement (chaleur, humidité ...).
- ✓ **Interactivité** : la centrale doit avoir l'impression d'avoir le module entre les mains :
 - On doit avoir accès aux données de navigation en temps réel.
 - Des informations sur l'état du GEO doivent être communiquées au Centrale.

b. Temps de réponse

Le composant GEO doit réagir dans des délais connus et garantis (Réactivité), on vise une mise à jour de la position dans 500 ms et le taux de traitement de notre microcontrôleur et de 16 MIPS (millions d'instructions par seconde).

c. Précision de localisation

La solution de géolocalisation utilisée combine entre le BLE (Bluetooth Low Energy) et les Motions Sensors. Le système doit localiser avec une précision < 1 mètre.

d. Portée de communication

Le module GEO doit communiquer avec l'application centrale via WIFI si disponible ou 3G dans le cas échéant, ainsi les utilisateurs pourront à tout moment échanger des informations avec l'application centrale.

4.2.2 Contraintes opérationnelles du composant

a. Contexte d'utilisation

Le composant GEO doit tout d'abord être adapté à utilisation normale, mais il doit aussi être robuste devant les problèmes rencontrés.

Parmi les problèmes liés à son contexte d'utilisation on trouve :

- Problèmes de communication avec l'application centrale, dans ce cas-là une bufferisation dans une mémoire interne sera prévu pour garder une traçabilité de l'historique de localisation.
- Problèmes de détection des bornes BLE, si notre module n'arrive pas à détecter une borne alors une erreur sera envoyée à l'application centrale.
- Erreurs relatives aux périphériques (BLE, *Motion sensors*, GPS,...), une fois l'un des composants de notre module est défectueux une erreur sera envoyée à l'application centrale et des Leds seront allumées sur le module pour indiquer au porteur le dysfonctionnement du composant en question.

b. Développement matériel

- Le prototype version 0 du composant GEO(le module hardware) doit être implémenté sur une maquette d'essai avec des supports pour faciliter les tests.
- Le schéma électronique et le circuit imprimé sont réalisés par des logiciels de conception assistés par ordinateur (CAO) (EAGLE PCB Design Software).
- Le prototype version 1 doit être réalisé sur une carte imprimée à dimension minimum (carte imprimé à double couche).

c. Développement logiciel

Langages utilisés dans la programmation : C/C++, FreeRTOS, XML, PHP,...

d. Détection des problèmes (Built In Tests)

Le composant GEO doit implémenter des fonctions de détection des problèmes afin d'assurer le maintien en condition opérationnelle du système. En effet Après la mise en marche du module, il doit effectuer un auto- test de :

- L'état des périphériques BLE, GPS, *Motion Sensors* et Wifi (fonctionnement ou dysfonctionnement)
- L'état de la connectivité avec l'application serveur.
- Compatibilité logiciel/matériel
- Vérification du binaire (calcul de CRC)

Par la suite le composant «GEO» doit envoyer le résultat du test à l'application centrale, ainsi il doit signaler son état de fonctionnement à son porteur (deux LED s'allument sur le module).

e. Fonctions de maintenance et mise à jour

Le module GEO doit être configurable et peut être mis à jour.

Le module doit être maintenable :

- On doit pouvoir faire mettre à jour notre système.
- On doit pouvoir changer des pièces défectueuses.
- Lors de l'initiation du composant s'il détecte que sa version SW n'est pas correcte il doit rentrer dans le mode maintenance et attendre le téléchargement d'un nouvel exécutable.
- Fonction debug/test : cette fonction doit offrir la possibilité de contrôler l'exécution du programme, en lui permettant par divers moyens de stopper (mettre en pause l'exécution du programme) et d'observer par exemple le contenu des différentes variables en mémoire. L'état d'exécution peut alors être observé afin, par exemple, de déterminer la cause d'une défaillance.
- Fonction de calibration : La fonction calibration permet de connaître la précision de la fonction de géolocalisation du module. L'utilisateur ne peut pas passer à ce mode, c'est au fabricant d'entrer à ce mode avant le mode opérationnel pour avoir une précision exacte du module.

4.3 Conception Matériel

4.3.1 Architecture globale du composant

Cette partie sera consacrée à l'architecture globale du module à savoir la description des blocs qui compose le module GEO et une description des flux externes échangés

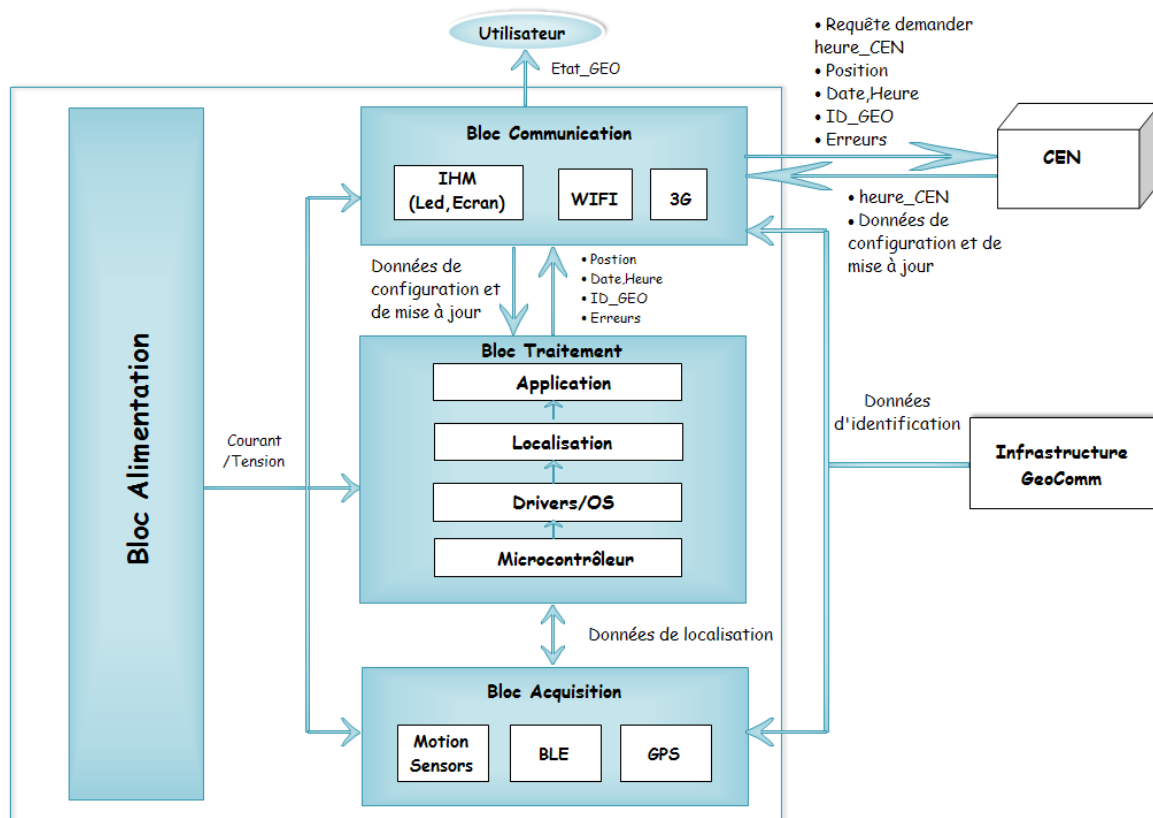


Figure 24. Architecture globale du composant GEO

L'architecture globale du module GEO se compose de trois blocs :

- **Bloc acquisition** : Ce bloc se charge de la collection des données. Il repose sur des composants électroniques assemblés en système d'acquisition de données (l'instrumentation), et sur le traitement du signal (théorie et traitements).
- **Bloc traitement** : C'est une unité qui contient le logiciel de traitement et d'exécution.
- **Bloc communication** : Ce bloc permet de communiquer les données de géolocalisation afin d'être exploité.
- **Bloc alimentation** : ce bloc permet de fournir l'énergie nécessaire pour faire fonctionner le système.

4.3.2 Spécification des flux externes :

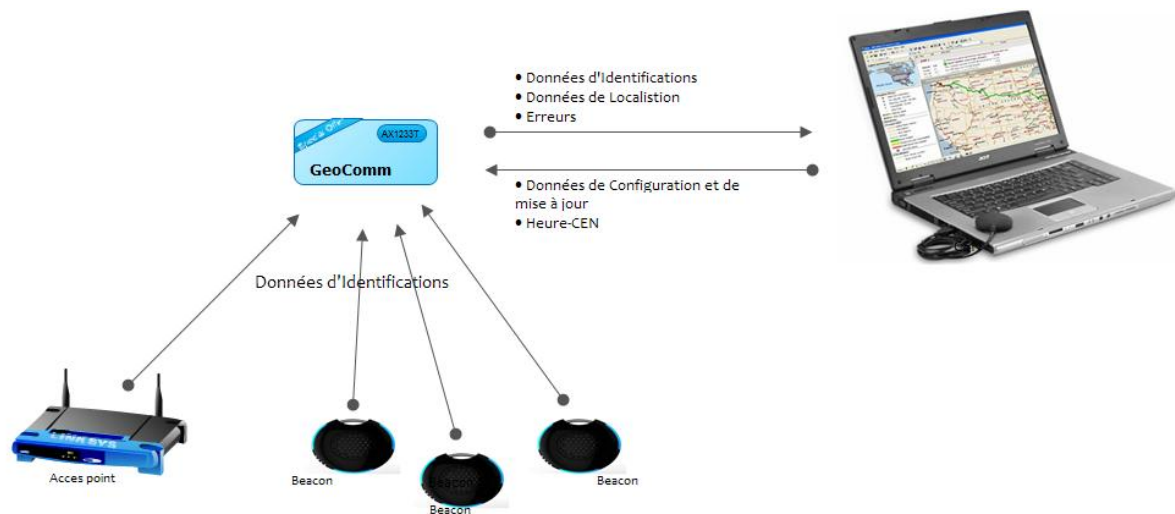


Figure 25. Flux externes échangés

Flux	Données	Source	Destination
Données d'identification	L'ID_GEO du module	GEO	CEN
Données de localisation	Latitude, Longitude, Altitude, date, heure, accélération, vitesse.	GEO	CEN
Données de configuration et de mise à jour	Périodicité de GEO, ..	CEN	GEO
Erreurs	Erreur lié à la défaillance de l'un des composants	GEO	CEN

Tableau 6. Flux échangés entre GEO et CEN

Flux	Données	Source	Destination
Données d'identification	Adresse MAC BLE fixe Adresse MAC des bornes d'accès Wifi	Infrastructure GeoComm	GEO

Tableau 7. Flux échangés entre GEO et Infrastructure GeoComm

4.3.3 Description des Blocs

Afin de comprendre la structure interne de GEO, il est utile de proposer, dans un premier temps, une décomposition du bloc [GEO] sous la forme d'un diagramme de définition de blocs, puis dans un deuxième temps, étudier les flux entre les différentes parties.

Les principaux Bloc sont :

a. Bloc d'acquisition

Fonctions acquisition données outdoor	
Entrée	Ondes électromagnétiques
Fonction	Localiser le porteur de GEO en outdoor par le biais du GPS
Sortie	Une trame NMEA contenant les données de localisations
Fonctions acquisition données indoor	
Entrée	Ondes électromagnétiques
Fonction	Localisation en utilisant les signaux des bornes BLE fixe combinés avec les données des montions sensors
Sortie	Coordonnées de localisations

Tableau 8. Les Fonctions du Bloc d'acquisition

b. Bloc de traitement

Ce Bloc est à base du microcontrôleur sur lequel est embarqué un système d'exploitation. Ce bloc doit traiter les données provenant du bloc Acquisition pour déterminer la position du porteur du système GEO.

Ce bloc réalise les fonctions suivantes :

Fonction Acquisition	
Entrée	<ul style="list-style-type: none"> ▪ Données GPS ▪ Données <i>Motion sensors</i> ▪ Données BLE
Fonction	<ul style="list-style-type: none"> ▪ Fait cycliquement l'acquisition des données de localisation. ▪ vérifie leur état (intervalle fonctionnel, taux de rafraichissement, état du capteur) ▪ met à disposition ces données avec un état de validation (valid data, data out of range, incorrect refresh rate, invalidsensor)
Sortie	<ul style="list-style-type: none"> ▪ Données GPS avec état de validation. ▪ Données Motion sensors avec état de validation. ▪ Données BLE avec état de validation.

Fonction Localisation	
Entrée	<ul style="list-style-type: none"> ▪ Données GPS avec état de validation ▪ Données <i>Motion sensors</i> avec état de validation ▪ Données BLE avec état de validation ▪ Données de configuration
Fonction	<ul style="list-style-type: none"> ▪ Calcule la position géographique de l'utilisateur avec les données GPS ou les données BLE/<i>Motion sensors</i>. ▪ La position est calculée à partir des données BLE/<i>Motion sensors</i> s'ils sont valides sinon la dernière position est retournée avec l'état non rafraîchie. ▪ les coordonnées de l'utilisateur (latitude, longitude, altitude, date) sont mis à disposition avec leur état de validation
Sortie	<ul style="list-style-type: none"> ▪ Coordonnés de l'utilisateur avec état de validation
Attribuer un identifiant statique ID_Geo pour chaque module	

Tableau 9. Les Fonctions Du bloc de traitement

c. Bloc communication

Ce bloc doit assurer les échanges des données avec l'application centrale (par Wifi /3G) et de communiquer l'état de fonctionnement du système à l'utilisateur via un terminal (signal lumineux ou message sur écran LCD)

Ce bloc réalise les fonctions suivantes :

Transmission/réception Wifi/3G	
Entrée	<ul style="list-style-type: none"> ▪ Coordonnés de l'utilisateur avec état de validation ▪ ID_GEO ▪ Erreurs ▪ Trame contenant l'heure de la centrale
Fonction	<ul style="list-style-type: none"> ▪ Une fois GEO est mis en marche une requête sera envoyé à l'application centrale pour demander l'heure de cette dernier pour une synchronisation éventuelle. ▪ Recevoir l'heure de la centrale. ▪ Transférer périodiquement les données de localisation à l'application centrale après les avoir encapsulé dans une trame. ▪ Informer la centrale sur l'état des composants.
Sortie	<ul style="list-style-type: none"> ▪ Des trames constituées d'un en-tête, d'un corps (coordonnées de localisation, ID_GEO, état des composants) et d'un FCS (Frame Sequence Check) permettant la correction d'erreur. ▪ L'heure de la centrale
Interaction avec GeoComm	
Entrée	<ul style="list-style-type: none"> ▪ Booléens indiquent l'état de chaque composant (wifi/3G,GPS , BLE, <i>Motion sensors</i>). ▪ Coordonnés de l'utilisateur avec état de validation

Fonction	<ul style="list-style-type: none"> ▪ Traiter les booléens - Chaque composant et présenter par une Led sur GEO si booléen est true la Led sera éteinte si false la Led s'allumera pour indiquer le dysfonctionnement du composant. - Si l'un des composants backbone (BLE, <i>Motion sensors</i>, GPS, Wifi/3G) est défectueux une Led représentant le fonctionnement globale du GEO sera allumer. ▪ Afficher les coordonnées de localisation ainsi que l'état du GEO sur l'écran.
Sortie	<ul style="list-style-type: none"> ▪ Signal lumineux ▪ Messages affichés sur l'écran LCD

Tableau 10. Fonctions du Bloc de Communication

Parmi les fonctions importantes de ce bloc est de Buffériser les coordonnées de localisation dans une mémoire dans le cas d'un échec de connexion wifi/3G.

d. Bloc d'alimentation

Ce bloc doit fournir de l'énergie électrique aux périphériques BLE, GPS, *Motion Sensors* et Wifi. Il possède des caractéristiques particulières à prendre en considération en fonction du matériel alimenté : courant (I, en Ampères), tension (U, en Volts), puissance (P, en Watts), type alternatif ou continu, fréquence.

4.3.4 Diagramme des Bloc Interne du GEO

On donne les principales caractéristiques des liaisons entre les instances de blocs :

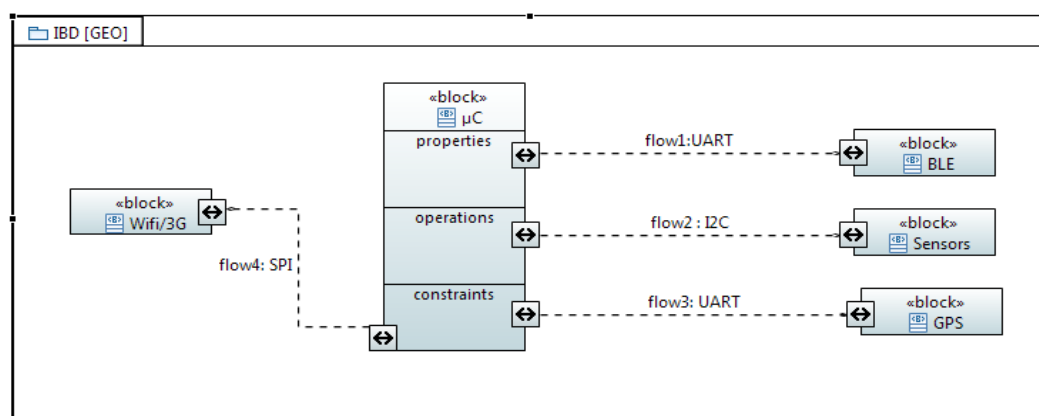


Figure 26. Bloc interne du GEO

Le diagramme de bloc interne du système met en jeu les instances des blocs qui composent le système ainsi que les flux qui sont échangés entre les instances de bloc. Les échanges entre les éléments se font par l'intermédiaire de ports.

On donne les principales caractéristiques des liaisons entre les instances de blocs :

- Liaison « Microcontrôleur » - « BLE » : Composée du port flow1 : UART c'est une liaison série de type RS232 servant de communiquer les données acquis par le BLE au Microcontrôleur selon le protocole UART.
- Liaison « Microcontrôleur » - « Motion Sensors » : composée du port flow2 : I2C c'est une liaison servant à remonter les données du motion sensor au Microcontrôleur afin d'être traité selon le protocole I2C.
- Liaison « Microcontrôleur » - « WIFI » : Composée du port flow4 : SPI c'est liaison série RS232 servant de faire la communication entre le module GEO et la centrale selon le protocole SPI.
- Liaison « Microcontrôleur » - « GPS » : Composée du port flow3 : UART c'est une liaison série de type RS232 servant de communiquer la trame NMEA émis par le GPS au Microcontrôleur pour déterminer la localisation.

Un deuxième diagramme de bloc interne orienté « Energie » montre les différentes liaisons pour l'alimentation en énergie des blocs.

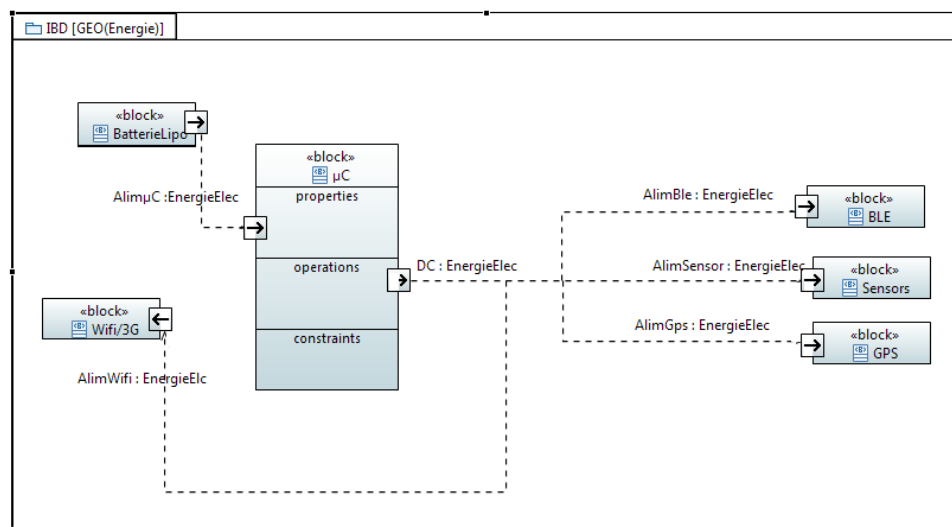


Figure 27. Bloc interne orienté énergie

Conclusion

Dans ce chapitre, nous avons présenté les différents diagrammes élaborés qui nous ont permis de cerner les différentes fonctionnalités de notre système avant de passer à la phase de réalisation. Dans le chapitre suivant, nous présenterons les différents outils matériels et logiciels utilisés pour la réalisation du projet.

Chapitre 5

Outils, développement et Réalisation

Dans ce chapitre, nous présentons l'environnement matériel et logiciel du projet. Ensuite, nous nous intéressons à la description des différentes étapes suivies pour la réalisation du projet.

5. Outils, développement et réalisation

5.1 Outils matériels

5.1.1 Carte Arduino

Pour la réalisation de notre projet on a opté pour la carte *Arduino* car d'une part elle dispose de tous les bus dont on a besoin notamment le SPI, l'I2C et l'UART et d'autre part elle est facile d'intégrer dans différents projets.

La carte Arduino Mega 2560, est une carte à microcontrôleur basée sur un Atmega2560.

Cette carte dispose :

- de 54 broches numériques I/O
- de 16 entrées analogiques,
- de 4 UART (port série matériel),
- d'un quartz 16Mhz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP,
- et d'un bouton de réinitialisation

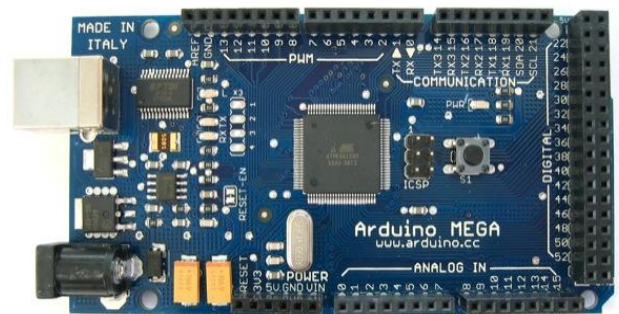


Figure 28. Carte Arduino MEGA

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur. Elle dispose de toute une série de facilités pour communiquer avec un ordinateur à l'aide d'un câble USB, l'alimentation étant fournie par le port USB, une autre carte Arduino, ou avec d'autres microcontrôleurs.

Un circuit intégré Atmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'Atmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur.

La communication en série de niveau TTL (5V) est disponible sur les broches 0 (RX) et 1 (TX). Le code utilisé pour programmer l'Atmega8U2 utilise le driver standard USB COM.

5.1.2 CC debugger

Le CC Debugger est un petit programmeur et un débogueur pour la TI Low Power RF System-on-Chips. Il a été utilisé pour configurer, programmer et déboguer les systèmes basés sur la puce de Texas Instruments notamment le module BLE112. [13]



Figure 29. CC Debugger

5.2 Outils logiciels

5.2.1 ISIS Proteus Professional Version 8



Figure 30. Logo du logiciel ISIS

ISIS est un éditeur de schémas électronique qui intègre un simulateur analogique, logique ou mixte. Il est simple à utiliser, et dispose d'une large librairie de composants. Une fois le schéma réalisé, il est possible de l'envoyer vers le logiciel Ares pour en effectuer le routage. Grâce à des modules additionnels, ISIS est également capable de simuler le comportement d'un microcontrôleur (PIC, AVR, Atmel, 8051, ARM, HC11...) et son interaction avec les composants qui l'entourent. Les résultats de la simulation sont très proches de la réalité. De plus, il dispose d'éléments de mesure virtuels : un oscilloscope, un voltmètre, un analyseur logique...

5.2.2 Eagle



Figure 31. Logo du logiciel Eagle

Eagle (Easily Applicable Graphical Layout Editor) est un logiciel de conception assistée par ordinateur de circuits imprimés. Il comprend un éditeur de schémas, un logiciel de routage de circuit imprimé avec une fonction d'auto routage, et un éditeur de bibliothèques. Le logiciel est fourni avec une série de bibliothèques de composants de base. C'est un logiciel multiplateforme.

5.2.3 Logiciel Arduino

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le *firmware* et le programme au travers de la liaison série. Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande. Le langage de programmation utilisé est le C++, compilé avec `avr-g++3`, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties.

Le logiciel Arduino intègre également un **TERMINAL SERIE** (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino (Figure 32). Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs : un élément essentiel pour améliorer, tester et corriger ses programmes.

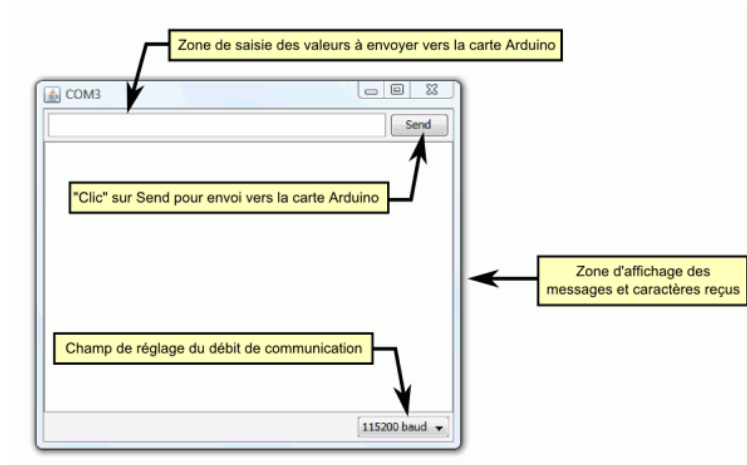


Figure 32. Moniteur série du logiciel Arduino

5.2.4 BLE SW Update Tool

Un logiciel fourni par Bluegiga, possède un compilateur BGbuild pour compiler et charger le firmware pour le configurer et programmer les modules BLE de Bluegiga.

5.2.5 Skytraq GPS Viewer

Une simple interface d'utilisateur qui permet au développeur de tester et évaluer complètement le module GPS ST32 Skytraq en temps réel. Le GPS Viewer démarre sans première installation. Juste quelques clic permettent de sauvegarder toutes les données de positionnement pour une analyse plus approfondie de du module GPS.

5.3 Langages utilisés

5.3.1 BGScript Scripting langage

BGScript est un simple langage librement disponible, développé par la société Bluegiga pour l'utilisation avec leurs modules BLE. Il nous permet de développer des programmes sans utiliser des microcontrôleurs externes à l'aide de simples commandes.

Le code s'exécute seulement en répondant à des événements BGScript, chaque ligne de code est exécutée en ordre, commençant par la définition et finissant par la déclaration de l'événement. Chaque ligne représente une seule commande.

5.3.2 Le langage XML

Le langage XML (eXtended Markup Language) est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. Sa simplicité, sa flexibilité et ses possibilités d'extension ont permis de l'adapter à de multiples domaines allant des données géographiques au dessin vectoriel en passant par les échanges commerciaux. De nombreuses technologies se sont développées autour de XML et enrichissent ainsi son environnement.

Le langage XML dérive de SGML (Standard Generalized Markup Language) et de HTML (HyperText Markup Language). Comme ces derniers, il s'agit d'un langage orienté texte et formé de balises qui permettent d'organiser les données de manière structurée.

5.3.3 Langage de programmation Arduino

Tout en permettant l'utilisation des fonctions classiques disponibles en langage C et C++, le langage Arduino (open source) est un langage orienté pour la programmation microcontrôleur qui offre des fonctions de syntaxe très simple mais très puissantes. La plupart des bibliothèques utiles sont également disponibles pour la communication série avec le PC, l'utilisation d'afficheur LCD standard, de clavier matriciel, de servomoteurs ou encore de moteurs pas-à-pas.

5.3.4 Serveur web embarqué

Un serveur web embarqué est un serveur web destiné à être mis en fonction sur un système embarqué et permettant de consulter et d'agir sur ce dernier comme sur n'importe quel élément du web. En effet le serveur web développé consiste à passer des informations de localisation

au serveur (l'application centrale) en utilisant le protocole HTTP via le module Wifi cc3000 qui constitue notre Interface Réseau. Le serveur web est développé par le langage PHP et les balises HTML.

5.4 Développement et réalisation

Notre projet consiste à concevoir et à réaliser le module GEO offrant une solution de géolocalisation complète indoor/outdoor, elle intègre tous les modules étudiés et choisis. La figure 33 représente l'architecture à mettre en œuvre

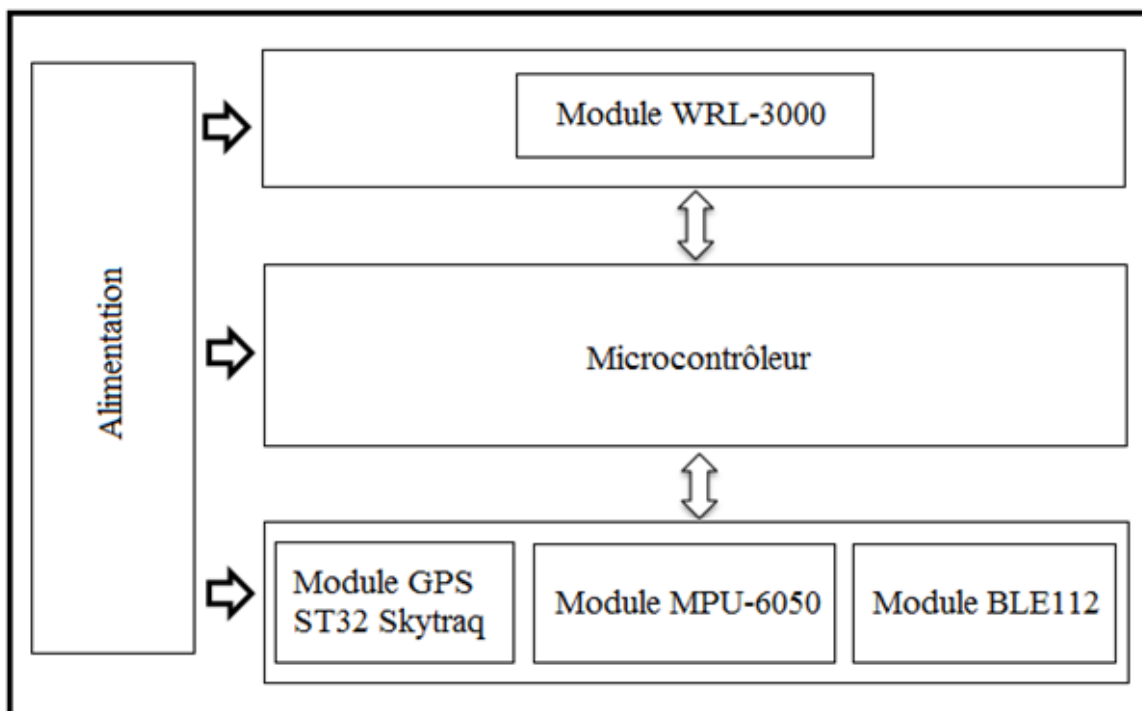


Figure 33. Architecture du module GEO

5.4.1 GPS ST32 Skytraq

a. Soudage du module GPS ST32

Le connecteur du GPS ST32 est de type : *pin header* 1.27mm x 8 pins. Donc une soudure des fils sur les pins est nécessaire. Pour se faire nous avons utilisé :

- Un fer à souder avec une forme pointue
- Etain/plomb assez mince 0.46 mm.
- Fils de bobines recouverts d'une couche supplémentaire de vernis thermo-adhérent dont ils se touchent sans provoquer de court-circuit, cette couche devenant adhérente sous l'effet de la chaleur ou d'un dissolvant.
- Pin Connectors

Après soudage des fils sur les pins et les connecteurs sur l'autre côté des fils :

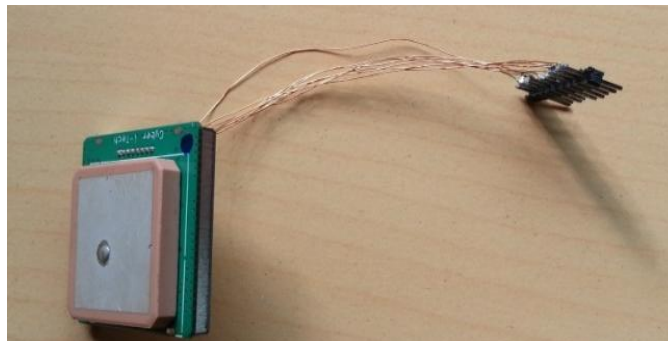


Figure 34. Soudage du module GPS ST32 Skytraq

b. Programmation du module GPS ST32

Une série de tests ont été effectués sur le module GPS ST32 Skytraq mais jusqu'à présent le module GPS ST32 ne retourne ni les réponses ACK ni les NACK ni aucune information à propos son fonctionnement, malgré qu'on a suivi la datasheet et les recommandations du support techniques. Comme le module GPS est un composant indispensable pour la réalisation d'une solution de géolocalisation outdoor, nous étions amenés à commander un autre module GPS qui est le module ublox LEA-6S. Les tests effectués seront présentés dans l'annexe 2.

5.4.2 SmartGPS

Cette partie a pour but de programmer le récepteur GPS afin de se géolocaliser grâce à la réception de signaux émis par des satellites géostationnaires. Le récepteur GPS détermine par calcul sa position et peut la transmettre sous forme d'une trame NMEA.

Le module GPS ublox LEA-6S est embarqué sur la carte SmartGPS de mikroElectronica dont on présente les caractéristiques dans l'annexe 3.



Figure 35. SmartGPS

Afin de tester le fonctionnement du récepteur SmartGPS (le module GPS ublox LEA-6S) nous l'avons branché avec PC via USB et s'assurer que l'antenne est sous le ciel et nous avons visualisé les résultats sur le logiciel GPS Viewer (Figure 36)

→ Résultat

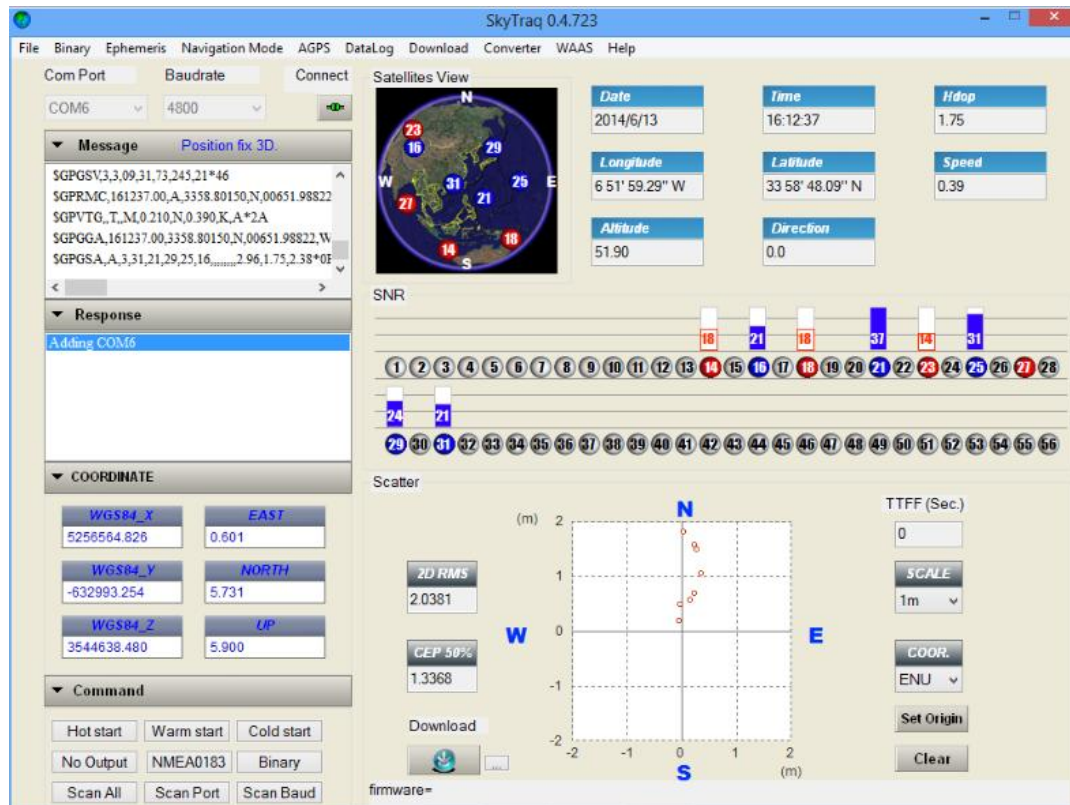


Figure 36. Résultat de la localisation avec GPS sous GPS viewer

En bleu : les satellites utilisés pour la localisation

En rouge : les satellites dont le signal n'est pas suffisant pour déterminer la position

Données de localisation :

Longitude : 6 51'59.29'' W

Latitude : 33 58'48.09'' N

Attitude : 51.90

Date : 2014/06/13

Heure : 16 :12 :37

→ SmartGPS et Arduino

Le module GPS ublox LEA-6S communique via une interface UART.

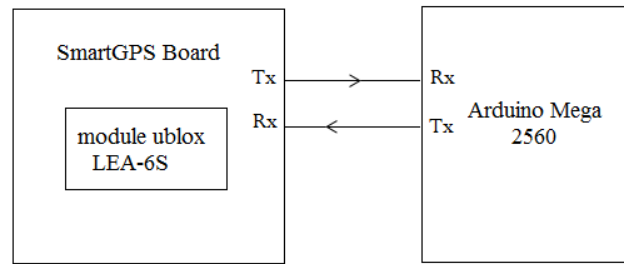


Figure 37. Communication SmartGPS avec Arduino

Afin de faciliter l'exploitation des modules GPS, une bibliothèque de GPS pour Arduino, nommée TinyGPS, a été développée. TinyGPS fournit la plupart des fonctionnalités GPS NMEA. Il suffit juste de faire la configuration de la liaison série et un bon schéma de branchement.

Ainsi notre software pour le module GPS ublox LEA-6S s'est basé sur cette librairie. Les principales fonctions assurant la récupération des données GPS sont les suivantes :

- `gps.f_get_position`: pour obtenir la position (la latitude, la longitude)
- `gps.stats` : pour donner indice si les données sont correctes ou pas. Il fournit des statistiques qui aident à la résolution des problèmes.
- `gps.crack_datetime` : Pour un accès plus facile à la date / heure
- Utiliser la librairie 'SoftSerial' disponible sur les librairies Arduino afin d'assurer une communication série entre le GPS et l'Arduino

→ Résultat

Sats	HDOP	Latitude (deg)	Longitude (deg)	Fix	Date	Time	Date	Alt (m)	Card	Chars	Sentences	Checksum
----	----	----	----	----	----	----	----	----	----	----	----	----
****	****	*****	*****	****	*****	*****	****	*****	***	3	0	0
5	173	33.980014	-6.866400	88	06/13/2014	16:10:05	118	47.00	NNE	243	2	0
5	173	33.980014	-6.866396	212	06/13/2014	16:10:18	243	47.20	NNE	3714	28	0

Figure 38. Résultat retourné par le SmartGPS

La figure 38 présente les résultats retournés par le SmartGPS GPS à savoir la latitude, la longitude, la date... En comparant avec les résultats obtenus avec le logiciel GPS viewer, nous avons trouvé les mêmes résultats.

5.4.3 Motion Sensors MPU6050

Le rôle du *Motion Sensor* est très important puisqu'il va faire le suivi du mouvement à chaque fois qu'une position est actualisée par le BLE112.

Cette section sera consacrée à la programmation et au calibrage de notre module avec Arduino.

a. Montage du module MPU-6050

Le module MPU-6050 possède un *Breakout Board* qui contient des trous ce qui facilite le soudage d'un pin Connector.

La figure 39 illustre le brochage de notre module avec l'Arduino puisque notre module est lié via I2C donc la connexion est assez simple.

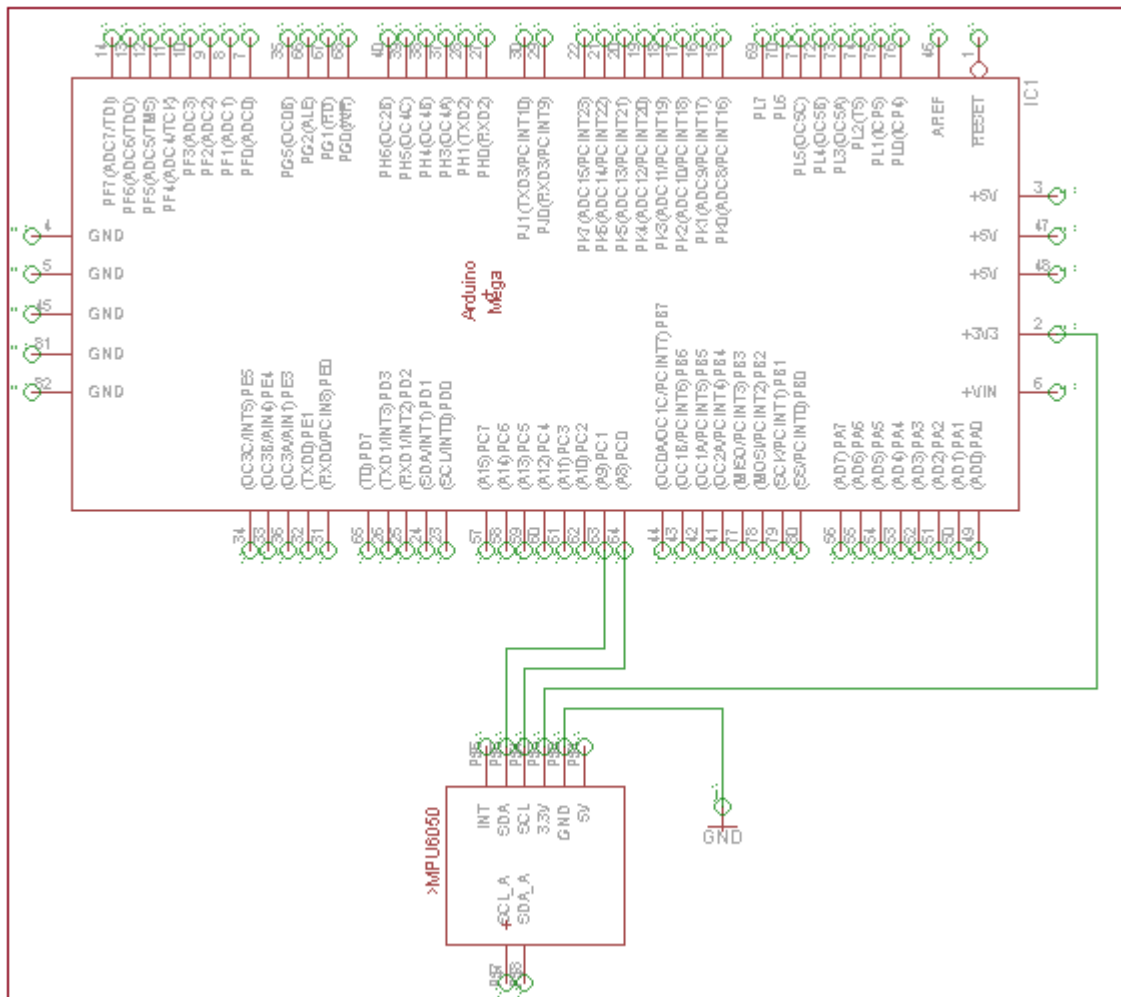


Figure 39. Branchement du module MPU6050 avec l'Arduino

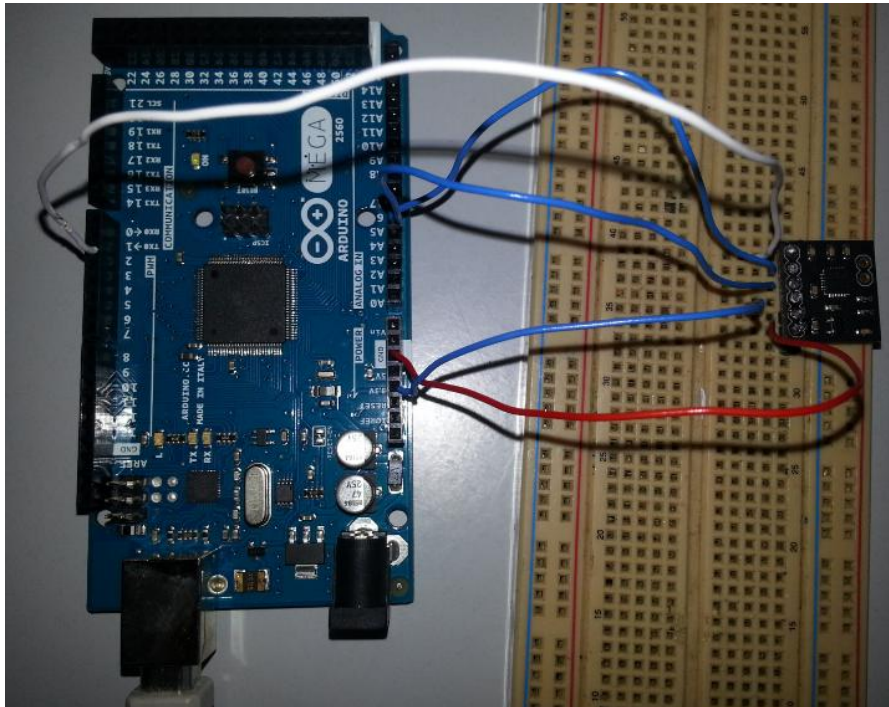


Figure 40. Réalisation du branchement de MPU6050 avec Arduino

b. Programmation du module MPU-6050

Après avoir branché notre module MPU6050 à la carte Arduino on passe à la programmation du module, pour ce faire on va exploiter la bibliothèque de Jeff Rowberg en l'installant sur Arduino IDE.

L'organigramme suivant présente les étapes logicielles suivies afin d'exploiter le composant, en commençant par initialiser le bus I2C et configurer les registres du module MPU-6050, après on s'assure que la liaison est établie en lisant la valeur de registre MPU6050_WHO_IAM_I.

En effet ce registre contient par défaut la valeur 0x69 d'après le Datasheet [10], ensuite le processeur DMP traite les valeurs brutes de l'accéléromètre et le gyroscope pour en sortir les angles d'Euler, le quaternion et les valeurs de l'accéléromètre et le gyroscope en éliminant la gravité.

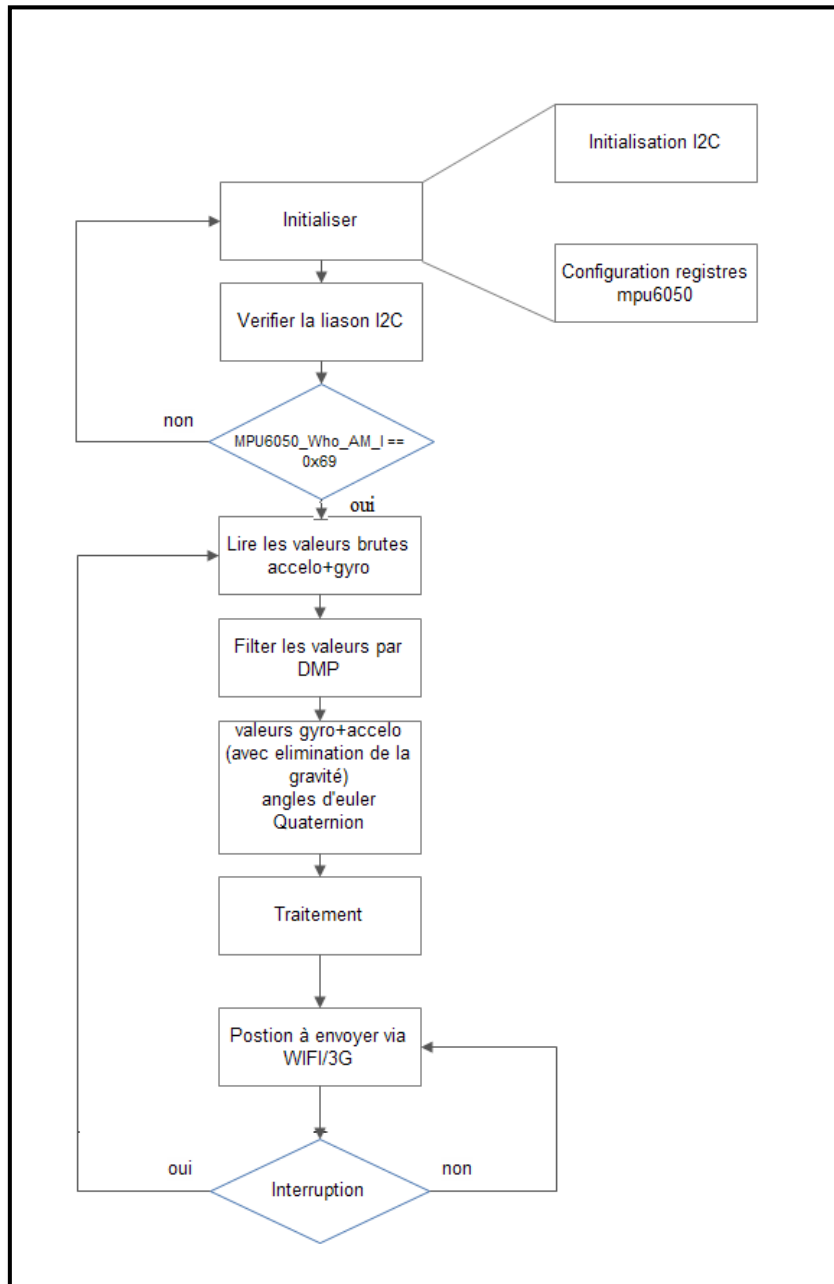


Figure 41. Acquisition des données du MPU-6050

→ Calibration du MPU-6050

La calibration est une étape indispensable avant le démarrage du calcul il sert à fixer les références (0,0,0) du module. Pour calibrer notre module nous avons appliqué l'algorithme suivant :

- Mettre le module sur une plate horizontale.
- Mettre tous les offsets à 0
("SetXGyroOffset/setYGyroOffset/setZGyroOffset/setZAccelOffset" =0).

- Passer le programme à Arduino et nous ouvrons le moniteur en série pour voir les valeurs retournées.
- Laisser le module opérant pour quelques minutes (5-10) donc la température est stabilisée
- Vérifier les valeurs dans le moniteur en série et on note-les.
- Mettre à jours nos offsets et on passe le programme à nouveau à la carte.
- Répéter ce processus jusqu'à ce que notre programme retourne 0 pour chaque gyro, 0 pour X et Y accel et +16384 pour Z accel. [15]

```

a/g: -18 20 16346 0 4 -4
a/g: 2 -2 16378 -3 3 -4
a/g: -8 -8 16424 7 -4 8
a/g: 4 18 16390 6 -11 3
a/g: 14 24 16348 8 -3 5
a/g: -16 -38 16366 5 -3 2
a/g: 16 18 16428 2 -1 4
a/g: 16 36 16400 4 1 0
a/g: 50 72 16400 -5 -1 -1
a/g: 50 34 16400 -3 -12 -6
a/g: 22 20 16380 0 -3 -4
*****
*****
0.21 -0.20 16378.23 0.78 -0.31 1.82 *****
*****

```

Figure 42. Résultat de la calibration

→ Résultat

Une fois le composant est branché correctement et programmé on peut visualiser les valeurs retournées sur la console Arduino.

```

Édition Croquis Outils Aide
J8050_DMP6 $
Serial.print("\t");
Serial.print(aaReal.y);
Serial.print("\t");
Serial.println(aaReal.z);
#endif
#ifdef OUTPUT_READABLE_WORLD
// display initial world
// and rotated based on
mpu.dmpGetQuaternion(&q,
mpu.dmpGetAccel(&a, f);
mpu.dmpGetGravity(&grav);
mpu.dmpGetLinearAccel(&a, f);
mpu.dmpGetLinearAccelInInt(&a, f);
Serial.print("accelerati");
Serial.print(aaWorld.x);
Serial.print("\t");
Serial.print(aaWorld.y);
Serial.print("\t");
Serial.println(aaWorld.z);
#endif
#ifdef OUTPUT_TEAPOT
// display quaternion vs
teapotPacket[2] = fifoBu
teapotPacket[3] = fifoBu
teapotPacket[4] = fifoBu
teapotPacket[5] = fifoBu
rsement terminé
Défilement automatique
Pas de fin de ligne 115200 baud

```

```

quaternion(XYZ) -0.20 -0.06 0.14 0.97
angle d'euler(XYZ) 158.02 10.30 14.89
acceleration(XYZ) -1146 1513 -1501
quaternion(XYZ) -0.22 -0.07 0.14 0.96
angle d'euler(XYZ) 156.12 10.80 14.54
acceleration(XYZ) -1197 1477 -1492
quaternion(XYZ) -0.23 -0.07 0.14 0.96
angle d'euler(XYZ) 154.22 11.27 14.18
acceleration(XYZ) -1248 1436 -1492
quaternion(XYZ) -0.25 -0.07 0.14 0.96
angle d'euler(XYZ) 152.31 11.74 13.80
acceleration(XYZ) -1296 1393 -1492
quaternion(XYZ) -0.26 -0.07 0.14 0.95
angle d'euler(XYZ) 150.40 12.19 13.40
acceleration(XYZ) -1336 1348 -1493
quaternion(XYZ) -0.28 -0.07 0.14 0.95
angle d'euler(XYZ) 148.48 12.63 12.99
acceleration(XYZ) -1380 1302 -1494
quaternion(XYZ) -0.30 -0.08 0.14 0.94
angle d'euler(XYZ) 146.55 13.06 12.56
acceleration(XYZ) -1422 1256 -1494
quaternion(XYZ) -0

```

Figure 43. Valeurs retournées par le composant MPU6050

5.4.4 Bluetooth Low Energy BLE 112

Cette partie a pour but la programmation du module BLE112 afin qu'il récupère l'identifiant de la balise la plus proche du module.

Il existe deux manières pour programmer le BLE112 :

- **Programmer avec le langage BGScript et exécuter une application directement sur le module**
- **Contrôle externe avec le protocole BGLib**

a. Soudage du module BLE112

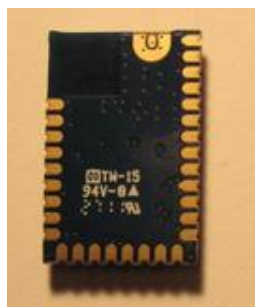


Figure 44. BLE112 avant soudage

Vu les petites dimensions du modules BLE112 et l'absence d'un support qui facilite l'exploitation de ce dernier, une soudure a été faite directement des petits fils sur les pins du module. Les figures 45 et 46 illustrent le module BLE112 après soudage des fils sur les pins et les connecteurs sur l'autre côté des fils.

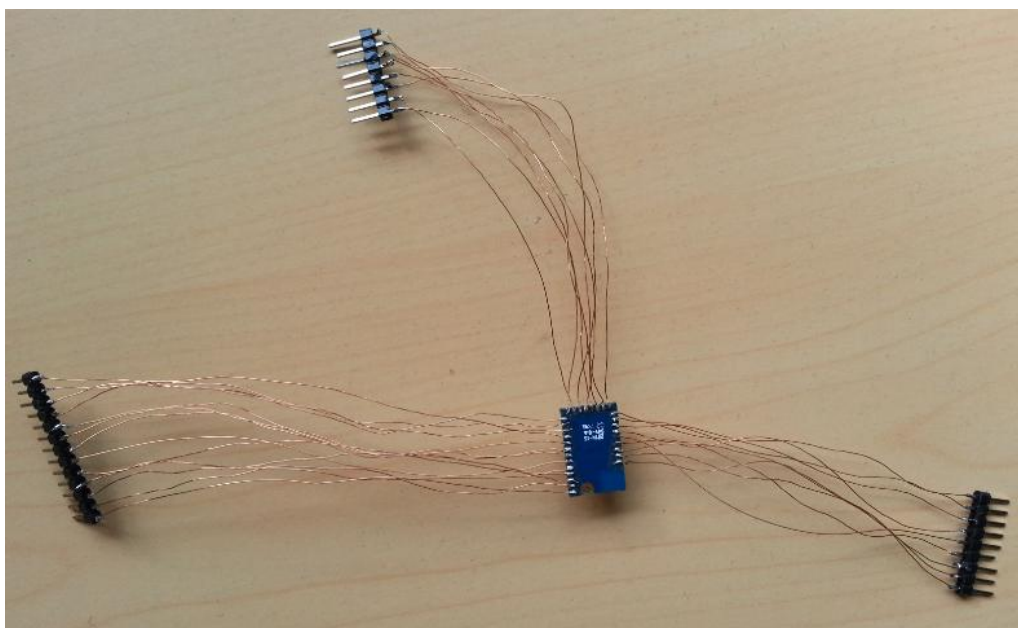


Figure 45. Face arrière du module Bluegiga BLE112 soudé

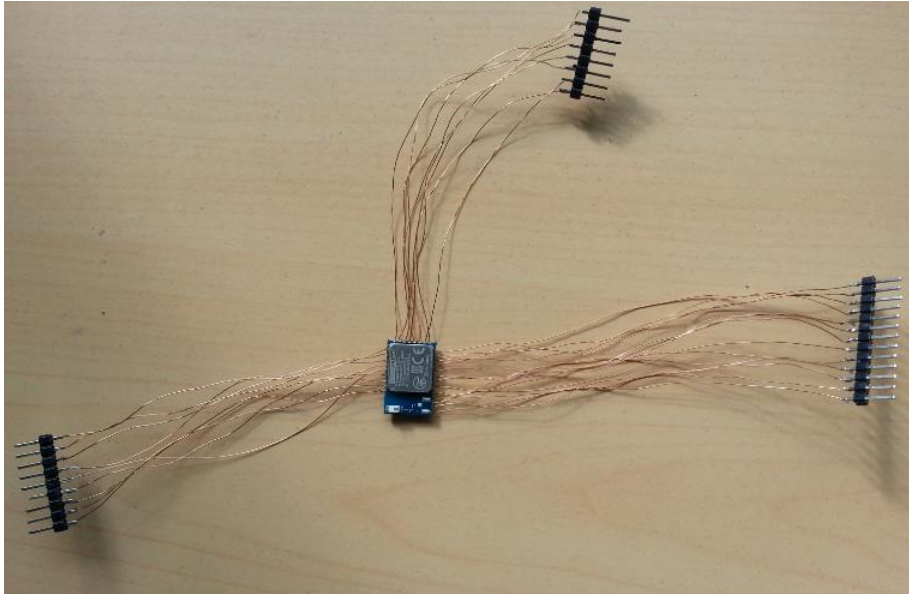


Figure 46. Face avant du module Bluegiga BLE112 soudé

b. Programmation du BLE112 avec BGScript

Afin d'utiliser le BLE112. Il doit être configuré et programmé selon les étapes suivantes :

- **Création du projet** : créer un fichier `project.bgproj` qui contient une organisation de tous les autres fichiers inclus dans le projet (Annexe 4)
- **Configuration hardware** : `hardware.xml` est un fichier qui contient la configuration du hardware par exemple : activer/désactiver des fonctionnalités selon notre besoin.
 - Définir le niveau de Tx power : `<txpower power="13" bias="5" />` cela correspond à 0 dBm
 - Activer le pin P1_6 du BLE112 comme wake-up pin cela permet d'affirmer à réveiller le module lors de l'envoi des données sur UART
`<wakeup_pin enable="true" port="1" pin="6" state="up" />`
 - Définir l'interface de communication utilisée : USART 1 UART Alt.1
 - Fixer le taux de transmission de données : 34800 Kbps (Annexe 4)
- **Création du GATT database** : décrire dans un fichier `gatt.xml` les profils (services) du Bluetooth utilisés et leurs caractéristiques
- Ecrire un simple BGScript (un fichier `.bgs`) qui répond à la fonctionnalité « Advertisement »
- Compiler le fichier `project.bgproj`
- Embarquer le fichier généré après compilation (`out.hex`) dans le module BLE112 à l'aide de CC Debugger

Afin de tester le fonctionnement du module BLE112. Il a été utilisé dans un premier temps comme « advertiser ». Pour ce faire le projet créé a été compilé et implémenté sur le module selon les étapes suivantes :

- Connecter le CC Debugger avec le PC via USB
- Connecter le CC Debugger avec l'interface du module BLE112 selon le schéma suivant :

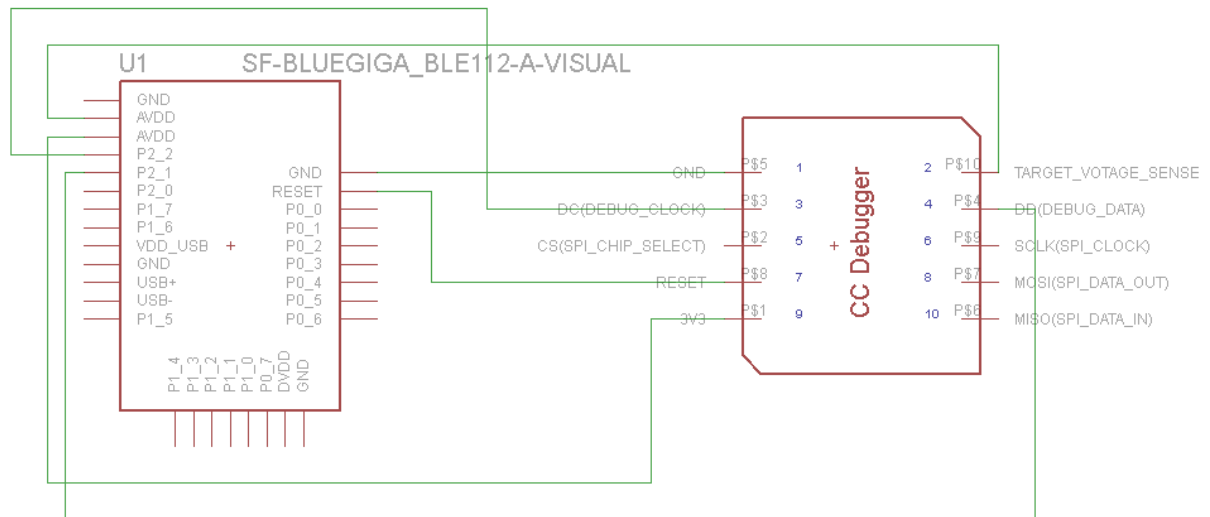


Figure 47. Branchement du BLE112 avec le CC Debugger sous Eagle

A l'aide de la figure 47, le BLE112 est connecté au CC Debugger.

- Appuyer sur le bouton Reset du CC debugger et s'assurer que sa LED est changée du rouge en vert car cela signifie qu'il détecte un dispositif à l'autre extrémité

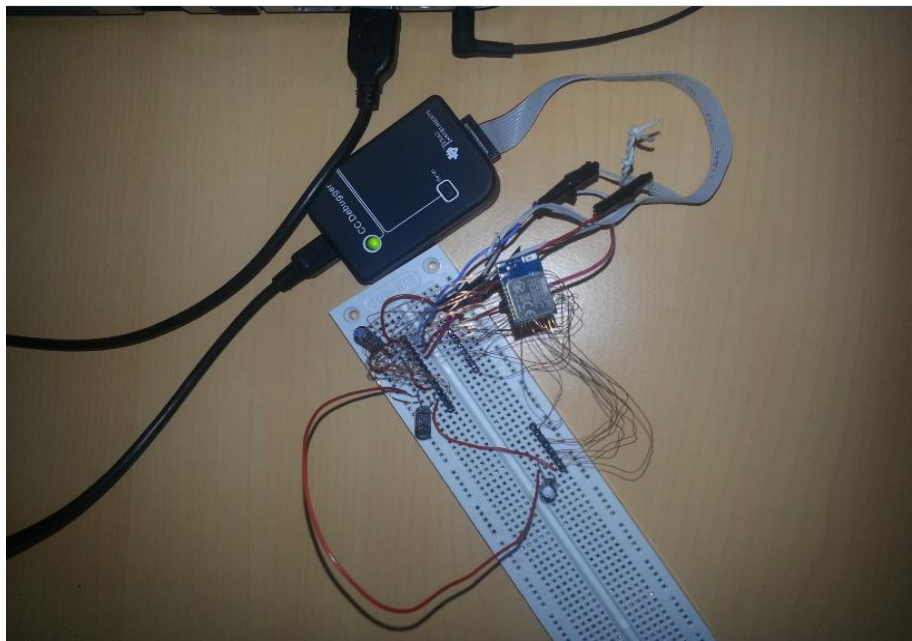


Figure 48. Réalisation du schéma de branchement BLE112 avec CC Debugger

- Lancer l’outil “Bluegiga BLE SW Update tool” et s’assurer que le CC Debugger est affiché dans la liste des ports
- cliquer sur Info afin d’obtenir les informations du module BLE112 : Nom du module, Adresse MAC, Numéro de série et Clé de licence. Cela permet de vérifier la connectivité du module BLE112 avec le CC Debugger.

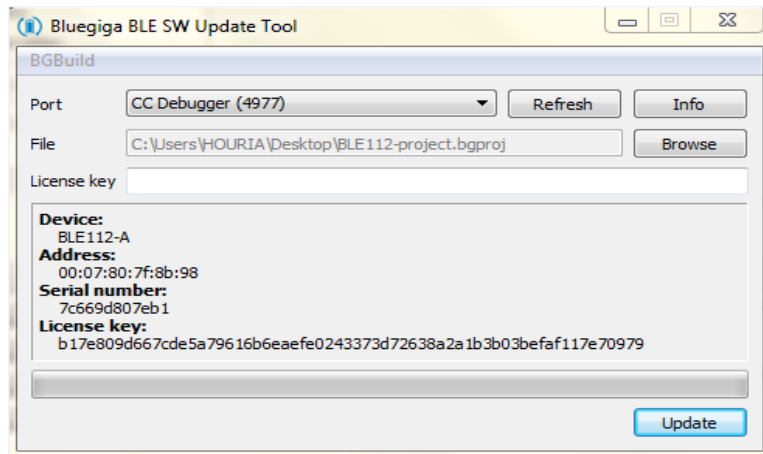


Figure 49. Interface BLE SW Update Tool

- Cliquer sur **Browse** et localiser le fichier project.bgproj et BGBuild => Select manually pour le compiler avec le compilateur BGbuild.exe de Bluegiga

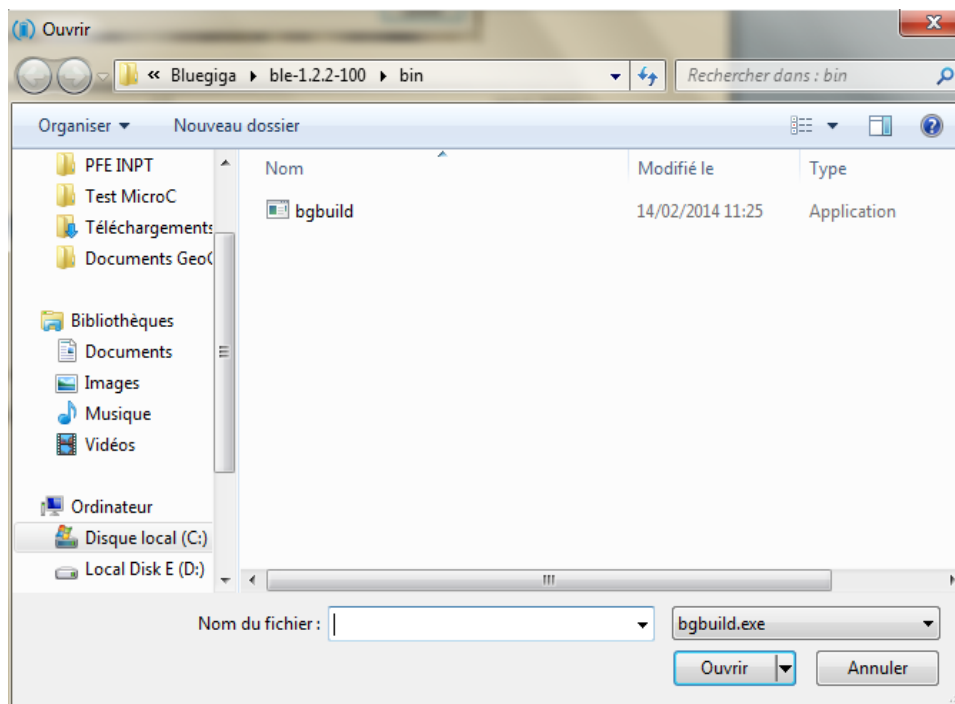


Figure 50. Compilateur BGbuild

Cette étape sert à compiler le projet et le module sera flashé avec un fichier.hex généré par le compilateur.

- Implémenter le software dans le module en cliquant sur update et en s'assurant qu'un message « Update completed » s'affiche sur un fond vert clair.

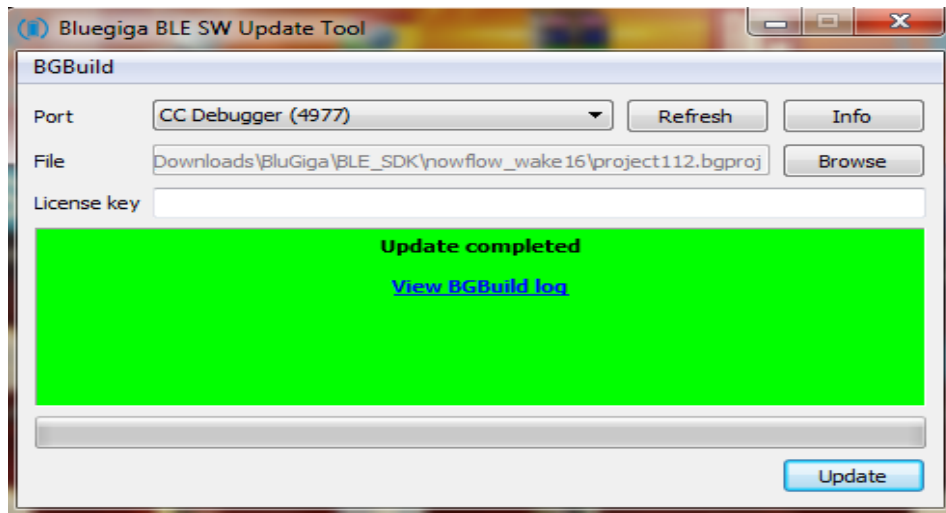


Figure 51. Résultat de la compilation

Un simple clic sur **View BGBuild log** pour plus de détails

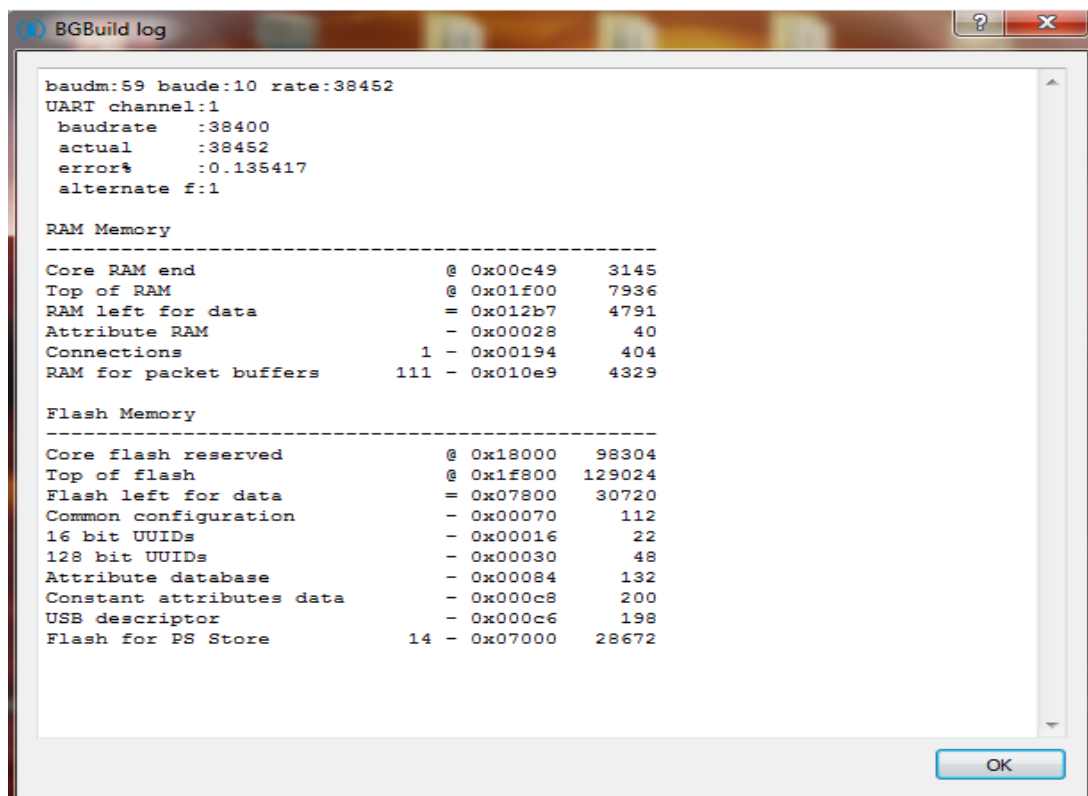


Figure 52. Affichage du BGBuild log

- Visualiser les résultats de la programmation sur l'application Android « BLE Scanner » qui permet de Scanner les autres périphériques BLE. La figure 53 montre le résultat de détection de notre module BLE112 et en changeant sa position le RSSI change.

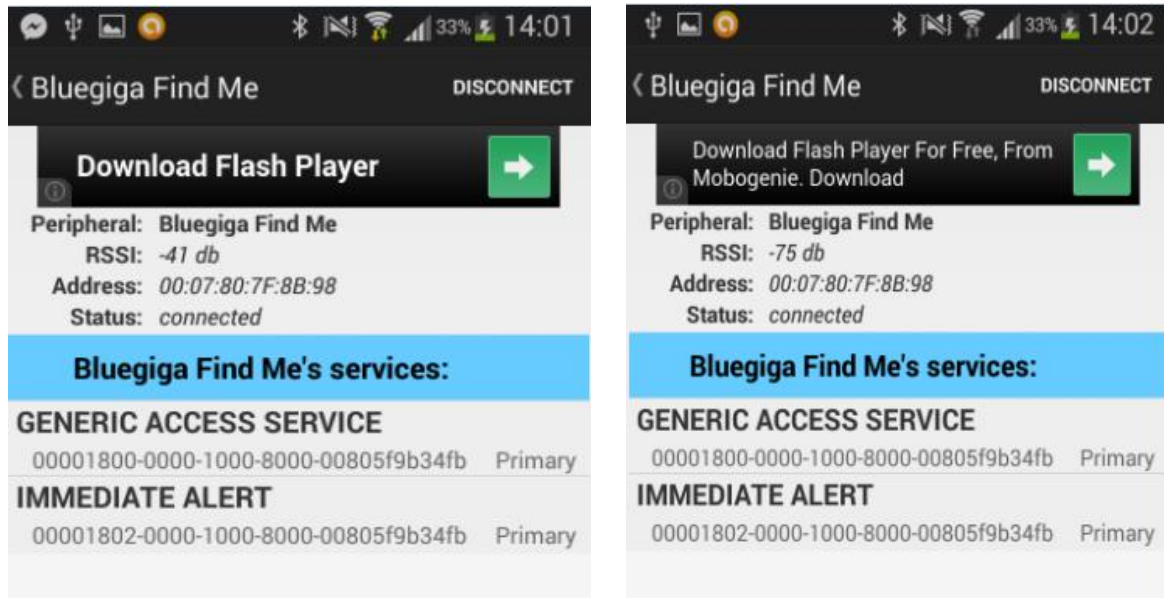


Figure 53. Détection du module BLE112 par l'application Android « BLE Scanner »

c. Identification des Beacons par le module BLE112

Après avoir testé notre module en le programmant en tant qu'advertiser, il est temps de le programmer en tant que Scanner pour qu'il puisse détecter les balises fixe. Cette fois-ci la deuxième méthode a été adoptée en utilisant notre carte Arduino.

→ Présentation des Beacons



Figure 54. Beacons avec leur pile CR2032 3V

Un Beacon est un tout petit appareil qui émet via Bluetooth un signal, qui peut être utilisé comme une balise pour faire du géo-positionnement. (Figure 54)

Cette balise émet à intervalle régulier un message, composé d'un triplet de valeurs (UUID, major, minor). Ce triplet permet d'identifier de manière unique un iBeacon.

En utilisant juste une ou deux des trois valeurs du triplet, vous pourrez déterminer l'appartenance d'un iBeacon à une zone précise.

Le module BLE112 doit scanner donc régulièrement la présence d'iBeacon répondant à la région définie précédemment.

Sous iOS 7 ou Android 4.3, l'interaction avec un iBeacon se fait avec l'application **easiBeacon**, cette plateforme de test permet :

- La configuration de l'intervalle de diffusion des messages : 2s
- Changer le nom de l'identifiant UUID
- La détection de l'iBeacon le plus proche
- La visualisation du niveau de sa batterie : 100%
- L'estimation de la distance séparée entre l'iBeacon et le smartphone : 0 m (Figure 55).

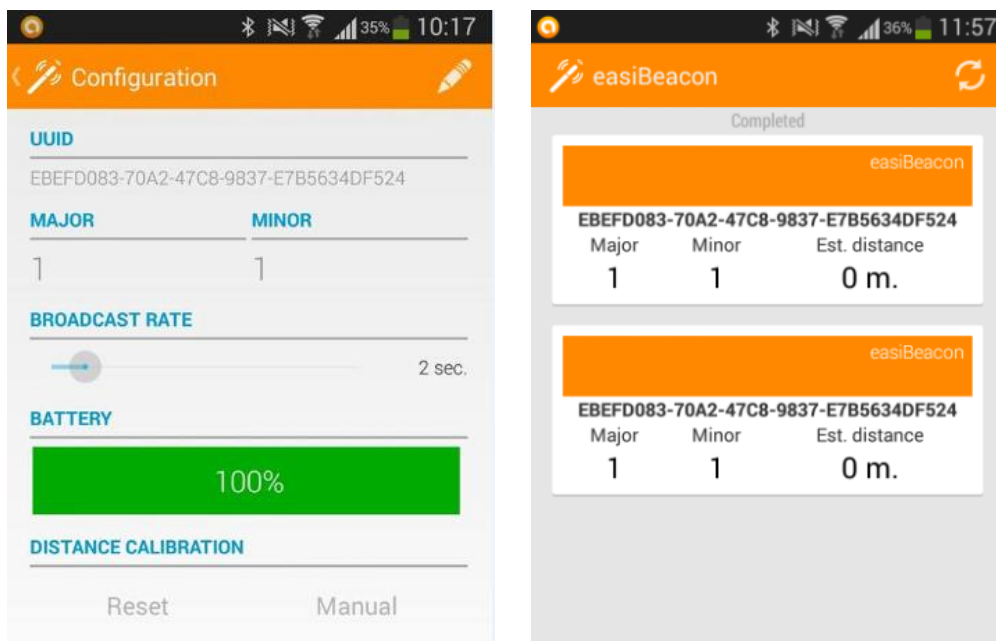


Figure 55. Résultat détection des Beacons par l'application Android « easiBeacon »

Le but principal de la localisation indoor est l'identification des balises BLE fixes, cette méthode consiste à récupérer l'identifiant du Beacon à laquelle le module GEO est plus proche.

Le module BLE112 a été programmé comme « Scanner » en utilisant le protocole BGAPI (Figure 56) qui utilise l'API binaire pour envoyer des commandes et recevoir des événements à partir d'un périphérique externe - généralement un microcontrôleur, mais tout ce qui peut communiquer sur UART directement ou un port série virtuel USB. [16]

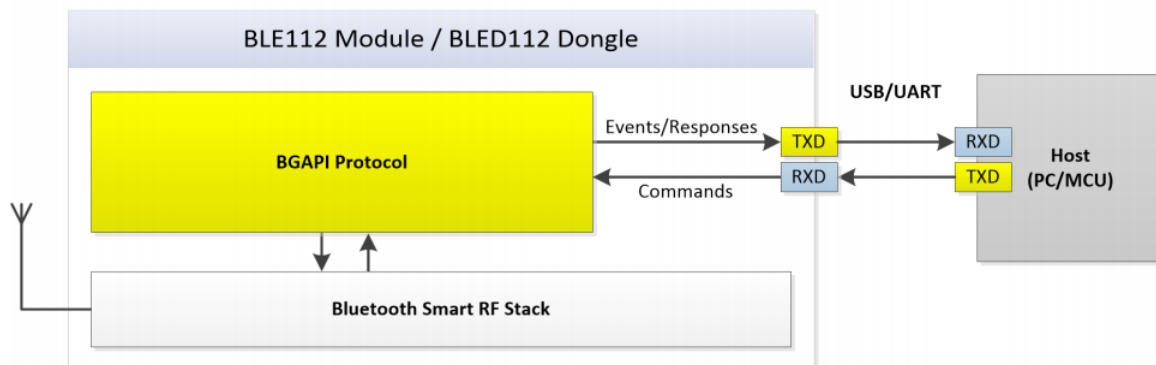


Figure 56. Protocol BGAPI

Toutes les commandes de BGAPI sont également disponibles en tant que fonctions C ANSI organisées dans une bibliothèque Arduino appelée BGLib. Nous avons utilisé la carte Arduino qui possède l'AVR comme microcontrôleur pour contrôler le module BLE112 :

- Envoyer des commandes (flux : MCU=>module) :
 Initialisation du wake-up pin P1_6
 Paramètres de scan : intervalle de scan 200ms
 Demande de scan des autres périphériques BLE
- Recevoir des réponses (flux : module=>MCU)
 Résultats de scan contenant l'adresse MAC des Beacons et le RSSI.
- Recevoir des paquets événement lorsqu'un événement survient sur le module (flux : module=>MCU)

Cela exige que le BLE112 doit être programmé avec UART connecté: `usart mode="packet"`
 Donc le module BLE112 a été reprogrammé à nouveau avec un fichier **.bgproj** qui contient que les deux fichiers de configuration **hardware.xml** et le **gatt.xml** et non pas le fichier **.bgs**
 Après la reconfiguration du BLE112, nous l'avons branché avec la carte Arduino (Figure 57).

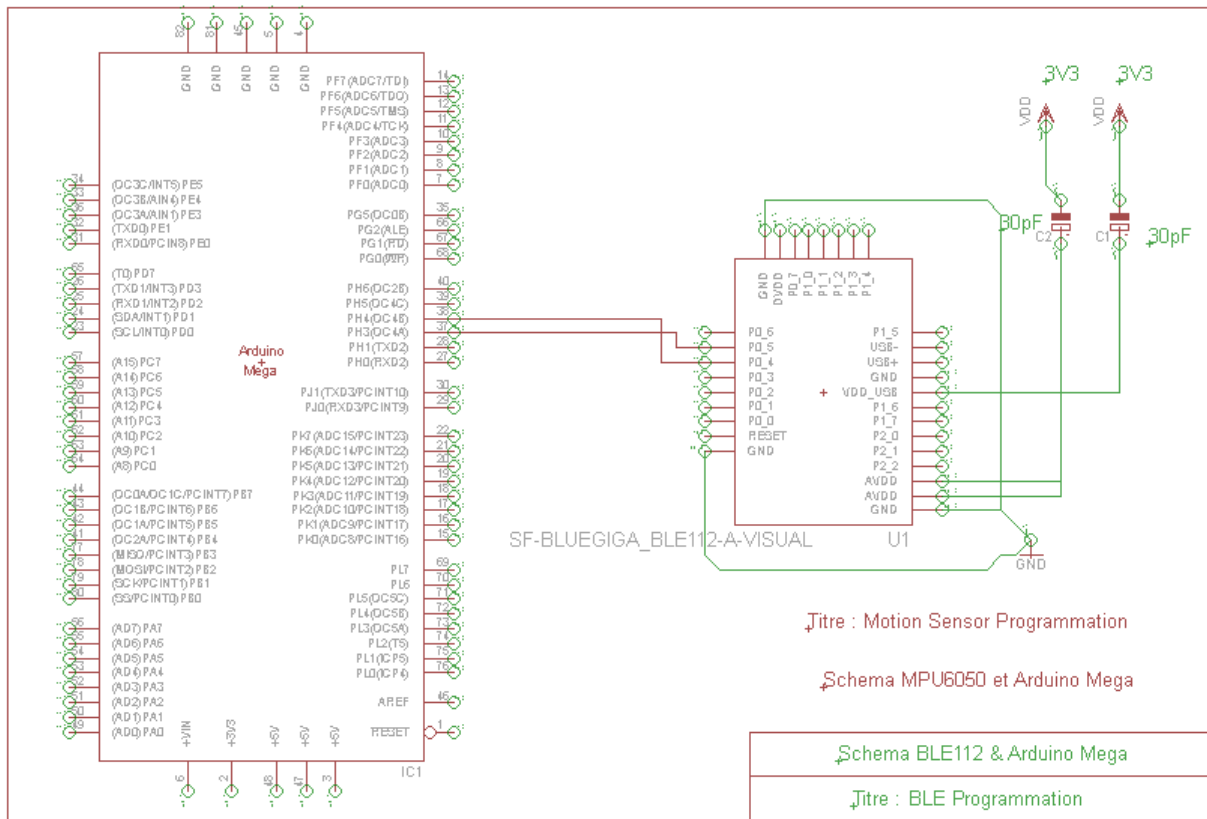


Figure 57. Schéma branchement BLE112 avec la carte Arduino sous Eagle

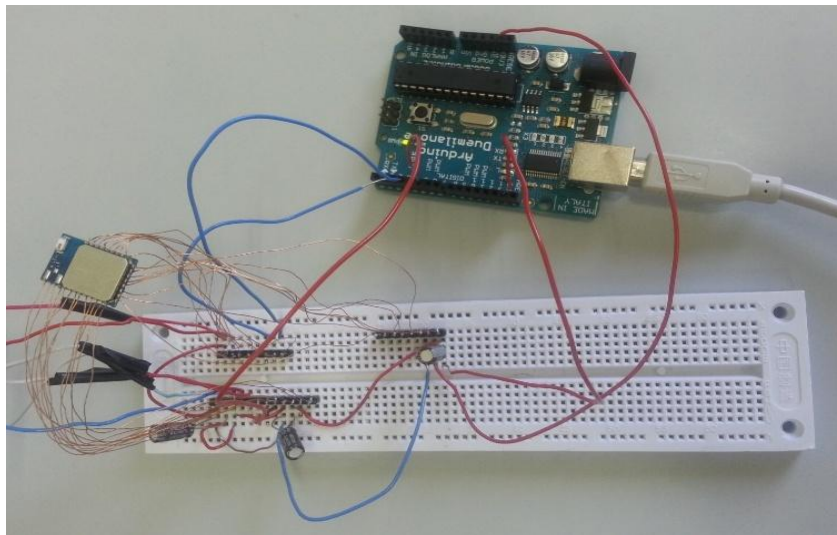
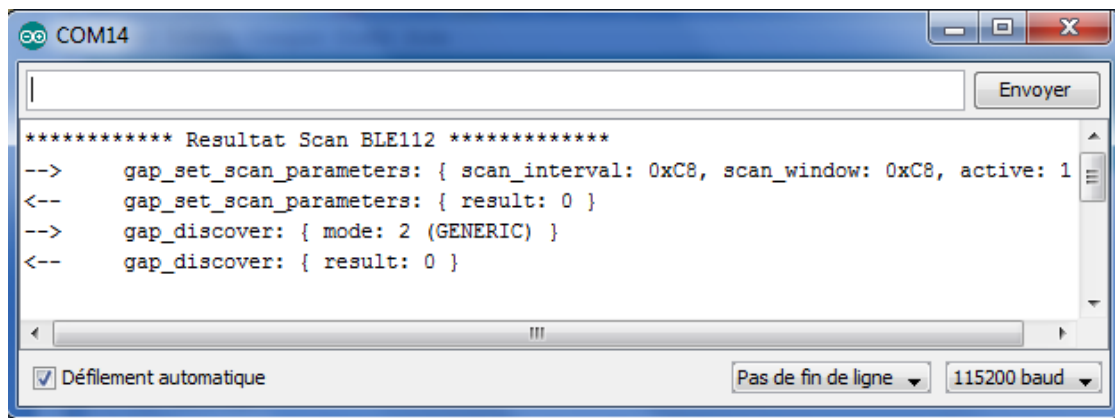


Figure 58. Réalisation du schéma branchement BLE112 avec la carte Arduino

→ **Résultat**

Le résultat du scan s’affiche dans la console lorsque notre module ne détecte aucun Beacon le résultat trouvé est 0 (Figure 59)



```

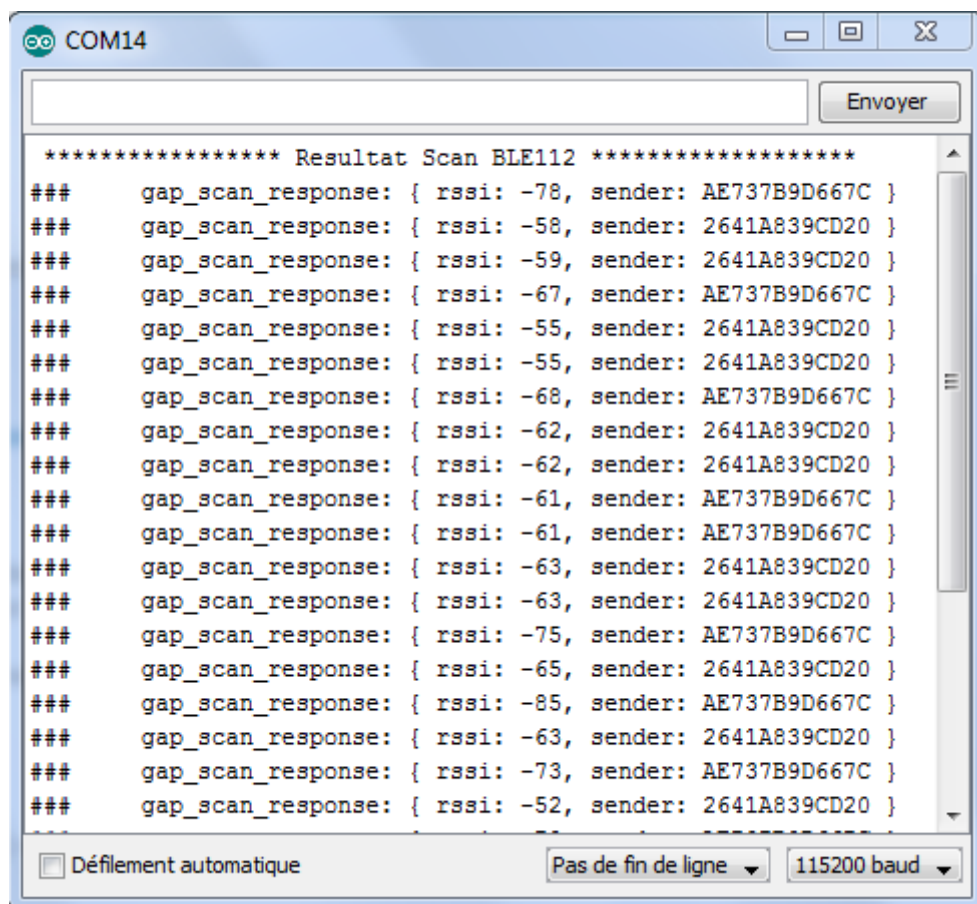
***** Resultat Scan BLE112 *****
-->   gap_set_scan_parameters: { scan_interval: 0xC8, scan_window: 0xC8, active: 1 }
<--   gap_set_scan_parameters: { result: 0 }
-->   gap_discover: { mode: 2 (GENERIC) }
<--   gap_discover: { result: 0 }

```

Figure 59. Résultat du scan

Lorsque le BLE112 détecte les 2 Beacons les résultats suivants s'affichent sur la console (figure 60) :

- Adresse MAC des deux Beacons : « AE737B9D667C » et « 2641A839CD20 »
- RSSI de chaque Beacons : en faisant varier la distance entre le Beacon et le BLE112 le RSSI change.



```

***** Resultat Scan BLE112 *****
###   gap_scan_response: { rssi: -78, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -58, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -59, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -67, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -55, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -55, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -68, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -62, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -62, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -61, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -61, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -63, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -63, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -75, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -65, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -85, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -63, sender: 2641A839CD20 }
###   gap_scan_response: { rssi: -73, sender: AE737B9D667C }
###   gap_scan_response: { rssi: -52, sender: 2641A839CD20 }

```

Figure 60. Résultats de détection des deux Beacons

d. Calibrage et estimation de la distance

En faisant des tests de calibration il est possible d'estimer des informations sur la distance entre un Beacon et le module BLE112 en fonction de la puissance émise par le Beacon.

Il ya trois zones statiques qui entourent chaque Beacon.

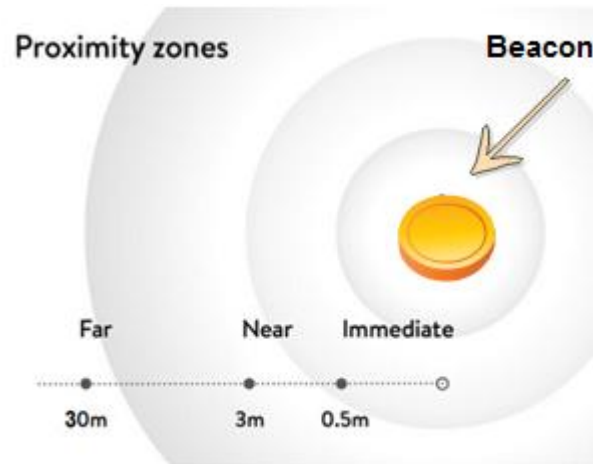


Figure 61. Zones de proximité d'un Beacon

Zone immédiate (0-0.5m) : Quand un dispositif est maintenu en place à proximité d'un Beacon on obtient une marge de puissance qui varie entre -45 et -55 dBm

Près de la zone (0.5-3m) : En quelques mètres du Beacon. La puissance est de -64 à 69 dBm

Zone lointaine (3-30m) : Plus que quelques mètres. La puissance est de -70 à -91 dBm

5.4.5 Wifi WRL-3000

a. Montage du module WRL-3000

Le module wifi « WRL-3000 » communique via le protocole SPI.

Le port de connexion de ce module wifi est décrit dans le tableau suivant:

VCC	Power connection. The WRL-3000 requires up to 300mA. Voltage should be between 2.7 and 4.8V
EN	Pull high to enable module to disable module and enter sleep mode
CLK	Data Clock
DIN	Data IN to the module
IRQ	Interrupt request line
DOUT	Data OUT of the module
CS	Chip select (active low)
GND	Ground connection

Tableau 11. Pins WRL-3000

La carte Arduino joue le rôle du maitre. La communication avec le module wifi se fait principalement via les trois lignes suivantes :

- **MISO** (Master In Slave Out) -> pin 50 : La ligne pour la réception de données du module wifi.
- **MOSI** (Master Out Slave In) -> pin 51: La ligne pour l’envoi de données au module wifi.
- **SCK** (Serial Clock) -> pin 52: l’horloge qui synchronise la transmission de données généré par la carte Méga.

D’autres pins sont utilisés pour sélectionner le module (chip select), l’activer (Enable) et pour envoyer des interruptions (Irq).

La figure 62 décrit le branchement de la carte Arduino Mega et le module Wifi WRL-3000.

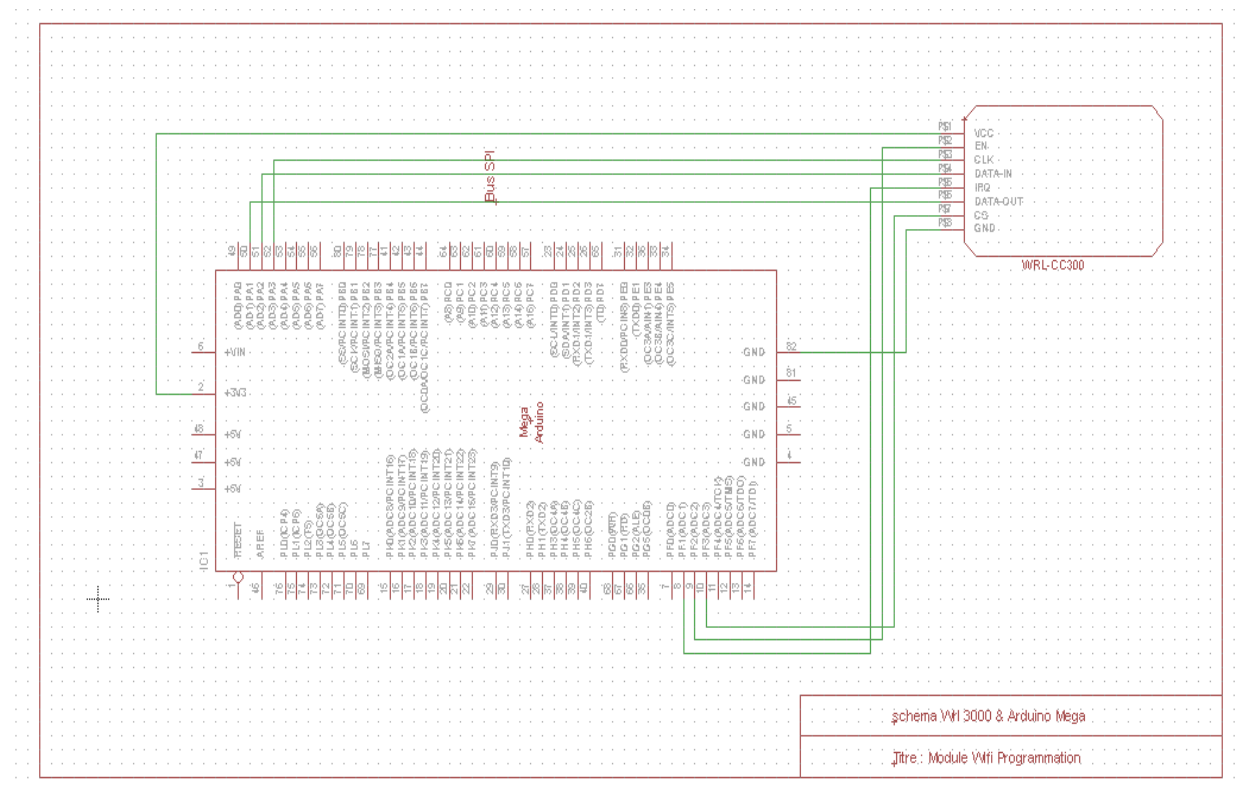


Figure 62. Branchement du module Wifi avec la carte Arduino sous Eagle

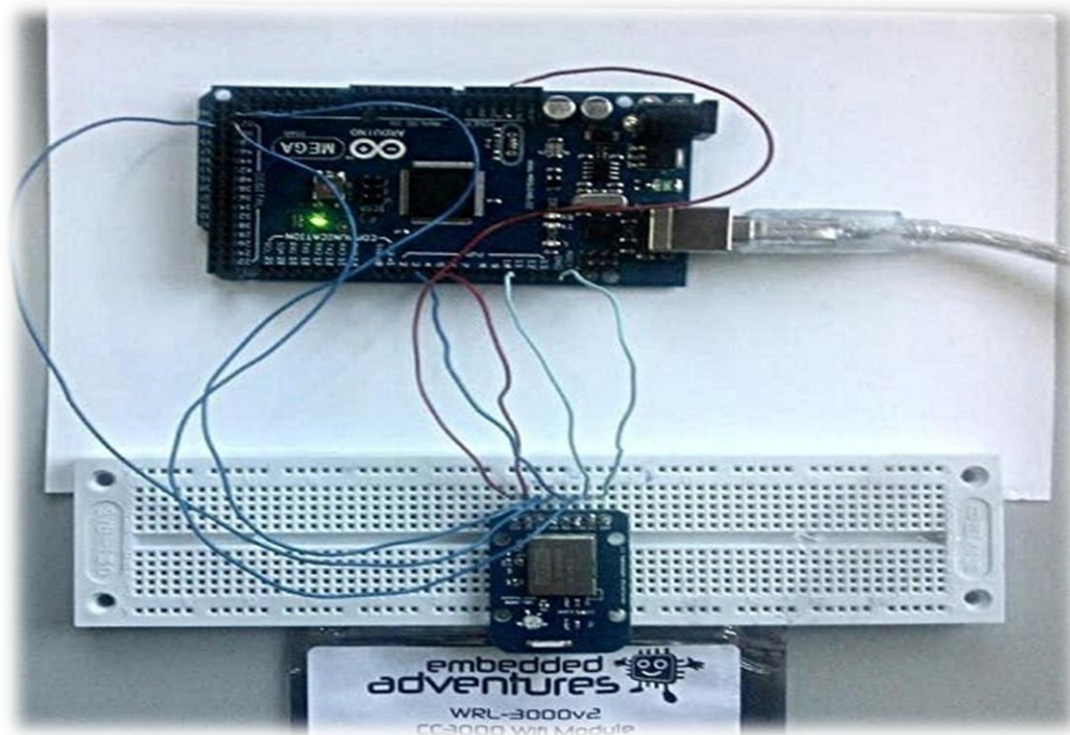


Figure 63. Réalisation du schéma du branchement Wifi avec Arduino

b. Programmation du module WRL-3000

Le module wifi est utilisé pour envoyer les données de localisation issues du module BLE112, vers l'application centrale afin de positionner le porteur du module GEO sur une Map.

Il est prévu aussi pour envoyer les données issues des autres modules (GPS, MPU-6050)

Pour la programmation du module wifi cc3000 on a exploité son driver fourni par le constructeur Texas Instrument (TI).

L'exploitation du WRL 3000 passe par les opérations suivantes :

→ **Test, Configuration, Connexion**

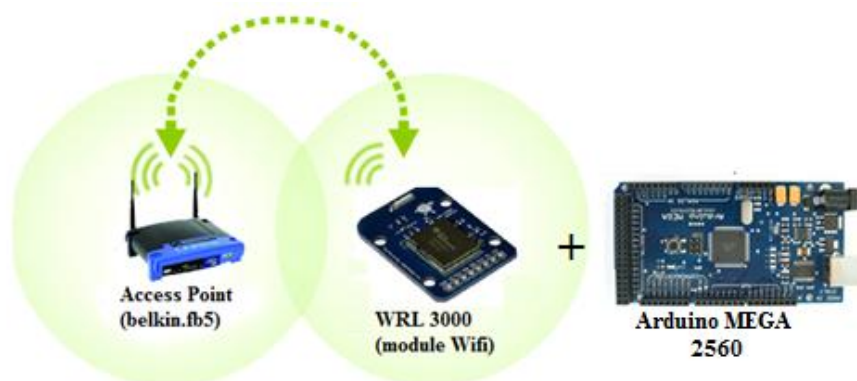


Figure 64. Connexion du module WRL3000 avec le point d'accès

- Dans cette phase, on teste si module est fonctionnel ou non tout en récupérant son adresse MAC.
- Après nous le configurons en précisant les paramètres de connexion : Wifi name (WLAN_SSI), le mot de passe (WLAN_PASS), type sécurité réseau (WLAN_SECURITY), adresse IP de serveur (IPAddr), et le numéro de port (PORT).
- Ensuite on le connecte le à notre réseau en faisant appel à la méthode « ConnectToAP() » qui prend en paramètre les données suivante : WLAN_SSID, WLAN_PASS, WLAN_SECURITY
- On observe les résultats d'établissement de la connexion (Adresse MAC, Adresse IP..) via le terminal série d'Arduino Mega (Figure 65).

```
COM34
Initialising the CC3000 ...
Hello, CC3000!

Firmware V. : 1.10
MAC Address : 0x08 0x00 0x28 0x58 0xB0 0x9E
Networks found: 1
=====
SSID Name   : belkin.fb5
RSSI        : 58
Security Mode: 3
=====
Deleting old connection profiles
Attempting to connect to belkin.fb5
Connected!
Request DHCP

IP Addr: 192.168.2.29
Netmask: 255.255.255.0
Gateway: 192.168.2.1
DHCPsrv: 192.168.2.1
DNSserv: 192.168.2.1

Défilement automatique
```

Affichage @MAC du module

Scanning et affichage des reseaux Wifi trouvés

Une adresse IP est attribuée au module par le protocole DHP

Figure 65. Résultat de l'établissement de la connexion

→ Envoi des données de localisation au serveur : Client/serveur avec sockets

Les coordonnées du porteur du module GEO(le client) sont remontées à l'application centrale (serveur distant) en utilisant les sockets en mode non connecté (UDP).

Les Sockets UDP en Mode datagramme consistent à envoyer de paquets de données (datagrammes) sans avoir établi la connexion entre parties client et serveur.

La communication entre le Client GEO et le serveur Distant (application centrale) suit la procédure suivant :

- La partie serveur crée une socket et la lie à un port UDP particulier
- La partie client crée une socket pour accéder à la couche UDP et la lie sur un port quelconque
- Le serveur se met en attente de réception de paquet sur sa socket
- Le client envoie un paquet via sa socket en précisant l'adresse du destinataire
- Le paquet est reçu par le serveur

La figure 66 décrit le principe de communication entre le module wifi WRL-3000 (client GEO) et l'application centrale:

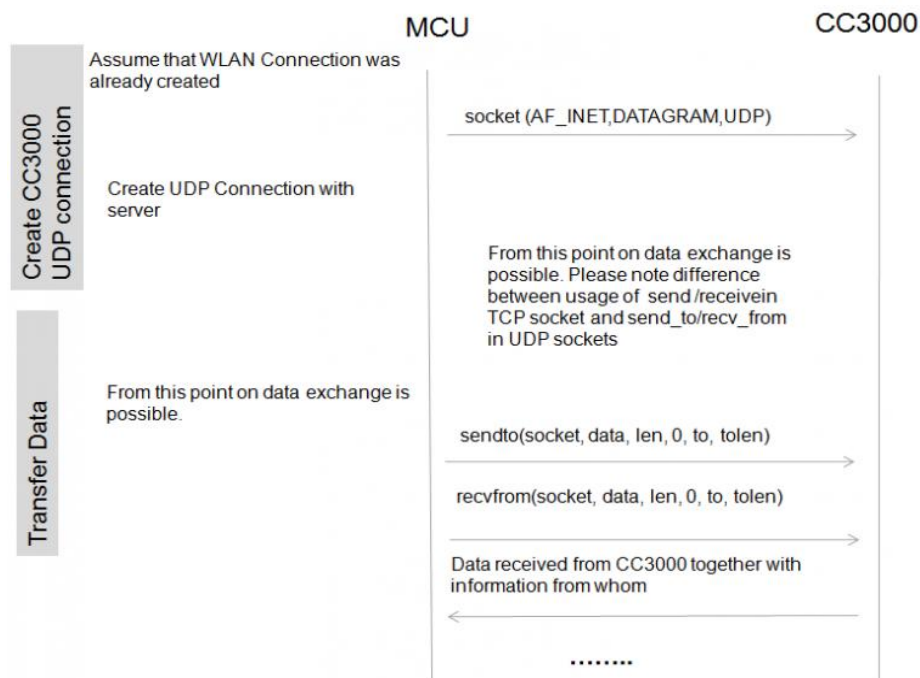


Figure 66. Procédure d'envoi et réception par socket UDP

→ Résultat

La figure 67 illustre un exemple de trame de localisation reçue au niveau serveur.

Cette donnée est traitée pour afficher par la suite la position du localisé sur un fond cartographique.

```

Microsoft Windows [version 6.2.9200]
(c) 2012 Microsoft Corporation. Tous droits réservés.
C:\Users\Ufact>cd C:\wampserver\www\UDP
C:\wampserver\www\UDP>php server.php
Message recu : $ae737b9d667c-6 51'59.18 W-33 58'48.23 N$

```

Figure 67. Exemple d'une trame de localisation reçue par le serveur

→ Sécurité des données : Implémentation du Cryptage de César

Pour assurer la sécurité des données de localisation nous avons implémenté l'algorithme de César [17] qui est une méthode de cryptage, considérée comme le plus ancien des algorithmes de chiffrement par substitution.

Cette méthode consiste à remplacer chaque lettre du texte à chiffrer par la lettre qui se situe n places plus loin de l'alphabet.

Nous avons donc défini notre clé de décalage et la figure 68 illustre une donnée de localisation avant et après l'implémentation du Cryptage de César.

```

beacon found :
ae737b9d667c
Locating data to send :
$ae737b9d667c-6 51'59.18 W-33 58'48.23 N$
Locating data encrypted :
aPTZVZQ\SYYZRPY]XTJX\QT [ ] `PVV]X[JW[QUV]WaW
data is sent, End.

```

Figure 68. Donnée avant et après cryptage

La donnée est déchiffré au niveau serveur tout en implémentant la fonction inverse du chiffrement, ensuite il est traité pour afficher la position du localisé sur un fond cartographique.

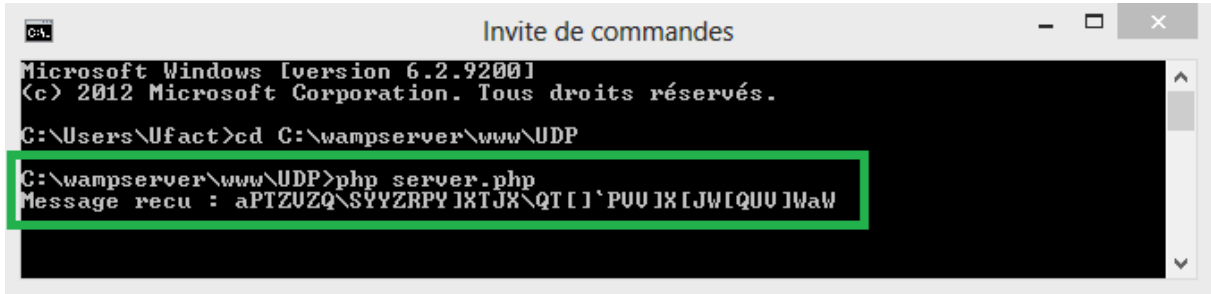


Figure 69. Donnée cryptée

5.4.6 Version 0 du GEO

Dans cette partie nous présentons la version 0 du module GEO réalisé, cette version permet la localisation indoor en utilisant le module BLE112 intégré sur la carte Arduino, les Beacons (balises BLE fixe) et le module wifi WRL-3000. Ainsi que la localisation outdoor en utilisant le Smart GPS. La figure suivante illustre les différents composants de la version 0 du GEO

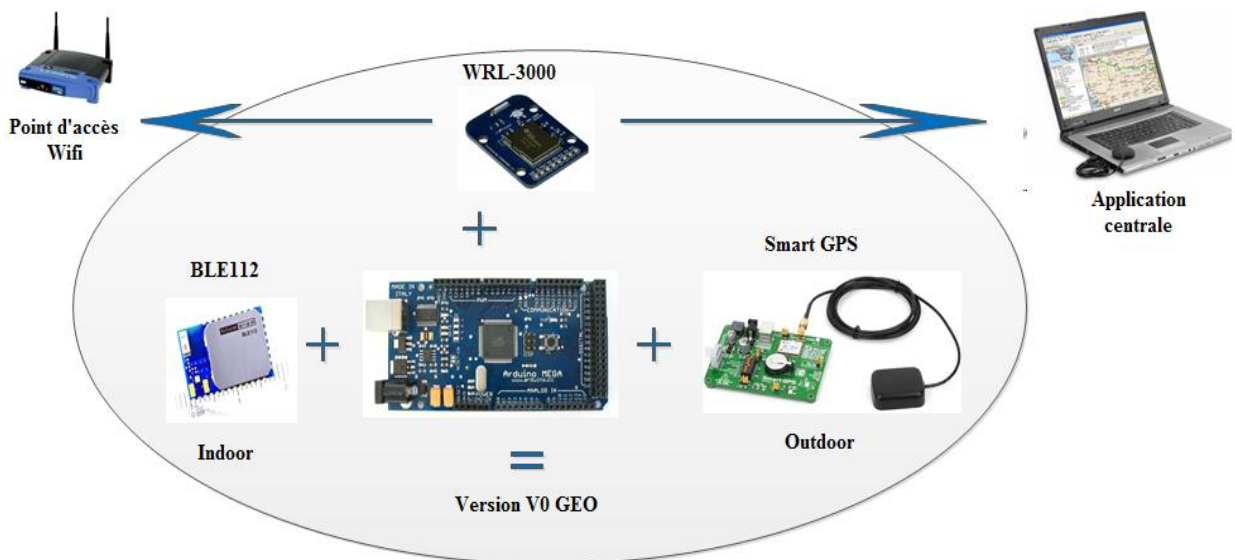


Figure 70. Schéma illustratif de la version 0 du GEO

→ Montage

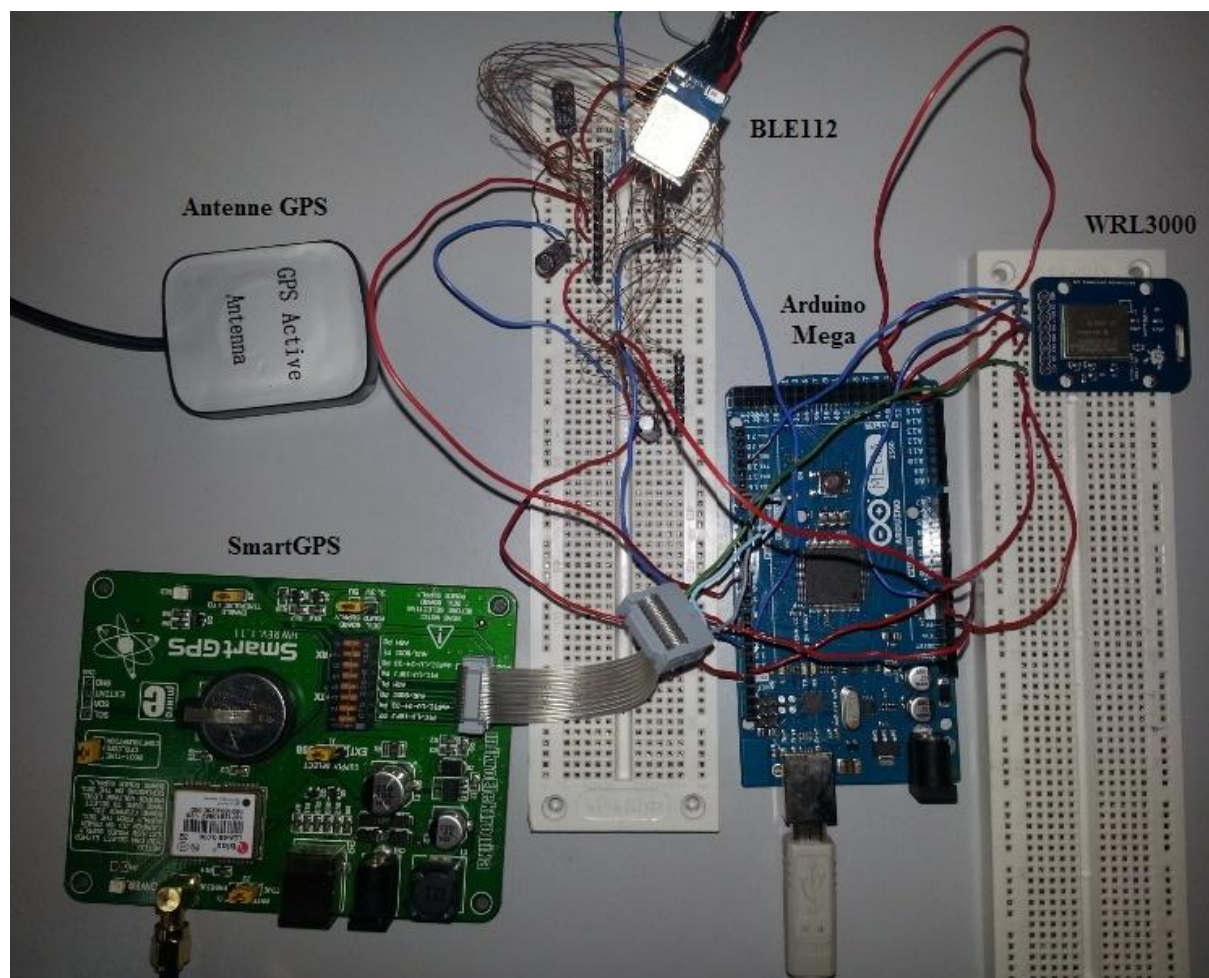


Figure 71. Montage du prototype V0

Après l'établissement de la connexion au réseau wifi auquel notre serveur est connecté, on commence par tester la localisation outdoor en récupérant les trames du récepteur GPS. Ces derniers contiennent toutes les données permettant de localiser le porteur du module en temps réel (altitude, longitude, date, time..).

Ensuite on teste la localisation indoor en lançant BLE112 en mode scan pour détecter les balises BLE fixes -Beacon- (à position préalablement connue et sauvegardé)

Une fois un Beacon est détecté on récupère son adresse MAC et sa position, pour estimer la position du porteur du module GEO.

Pour les deux type de localisation (indoor /outdoor) on construit notre trame sous une forme interprété par le serveur, puis on la crypte en implémentant l'algorithme de César, enfin la donnée de localisation est envoyée vers le serveur.

La figure 72 expose les résultats de test sur la Version 0 du module GEO.

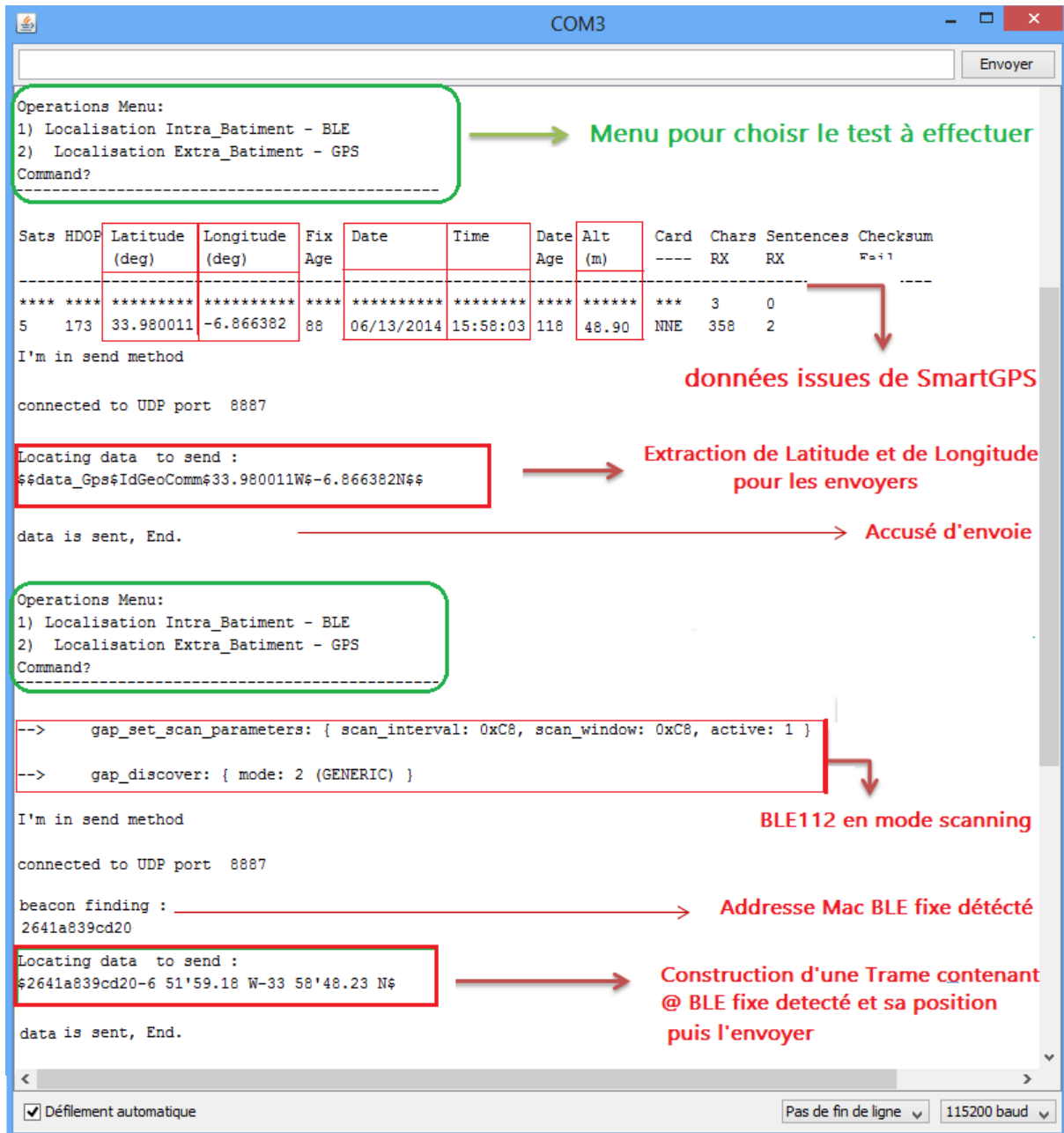


Figure 72. Résultat du scan et l'envoi de la localisation vers le serveur

Au niveau serveur distant la donnée reçue est décryptée et traitée pour afficher sur un plan géo-référencé (Google Map) la position du porteur du module GEO.

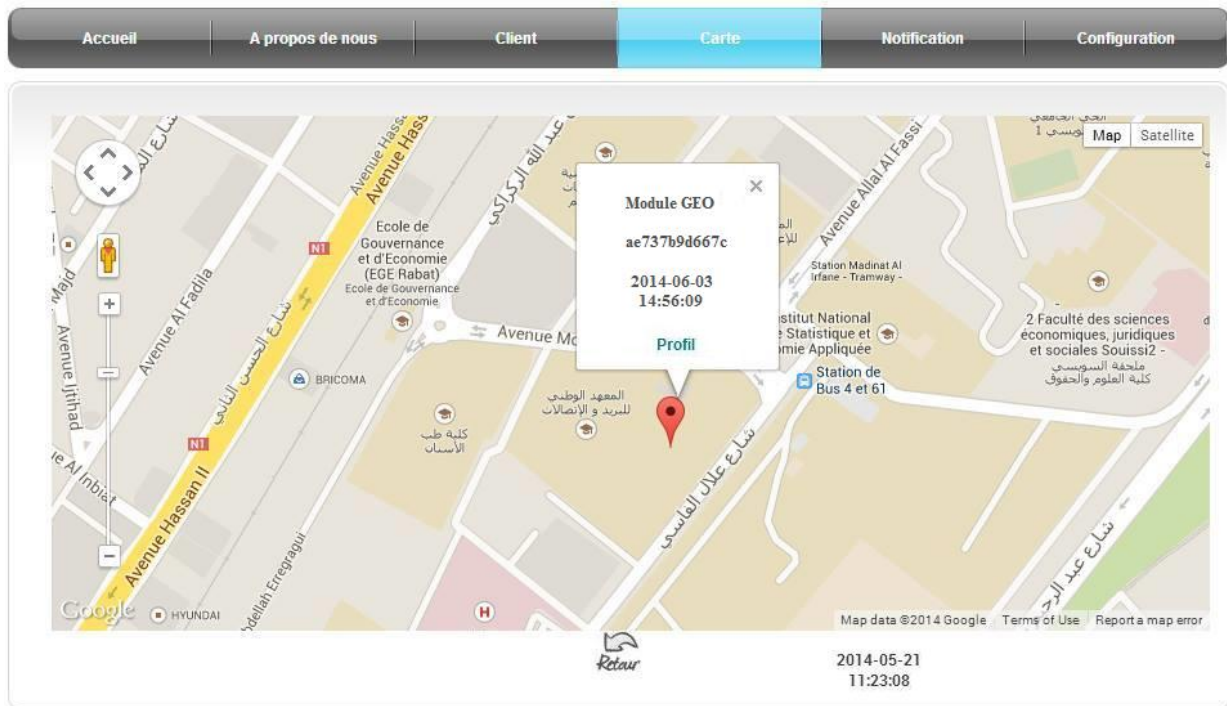


Figure 73. Affichage de la localisation du module GEO sur l'application centrale

De la même façon lorsqu'on est en outdoor le récepteur GPS détermine la position du localisé et envoie la trame cryptée vers l'application centrale

Conclusion

Dans ce chapitre nous avons détaillé les outils matériels et logiciels utilisés pour la réalisation de notre projet ainsi que les résultats obtenus et à travers un ensemble de captures d'écran nous avons illustré les résultats de la programmation des modules BLE112, GPS ST32, MPU-6050, WRL3000 et le SmartGPS, qui nous a menés à répondre au cahier de charge en réalisant une première version du module GEO.

Conclusion générale et perspectives

L'objectif de ce travail était la conception et la réalisation d'un système de géolocalisation permettant l'intégration des services nécessitant une géolocalisation à l'intérieur de bâtiments (Indoor) ou dans des situations où la localisation par GPS est inopérante. Les méthodes existantes fournissent des performances acceptables, mais nécessitent, pour la plupart, l'installation d'infrastructures spéciales ou des modifications importantes des infrastructures existantes. Le coût de ces modifications a empêché le déploiement de ce type de solutions. Il convenait donc d'étudier la faisabilité d'un système de géolocalisation, ne nécessitant pas un coût élevé et qui possède une bonne précision.

Le travail effectué a montré, la faisabilité d'un système, reposant sur la nouvelle technologie Bluetooth Low Energy (BLE) combinée avec les *Motions Sensors* pour la géolocalisation Indoor, et sur le récepteur GPS pour la géolocalisation Outdoor.

Cette approche nécessite une première phase d'acquisition de données de localisation à partir du GPS, BLE et *Motion Sensors*.

La seconde phase du système de localisation consiste en l'extraction de l'information pertinente des mesures effectuées et l'envoyer via Wifi vers une base de donnée distante de l'application centrale pour l'afficher sur un fond cartographique adapté.

Ce travail décrit les démarches suivies pour la mise en œuvre d'une première version prototype du système (V0). Il présente aussi les résultats obtenus, notamment les valeurs retournées par le *Motion Sensor* après calibration, que nous visons les exploiter afin de déterminer le déplacement pour obtenir une localisation précise.

Ainsi que l'identification des balises BLE par le module BLE112 comme une solution de la localisation indoor, la récupération de la trame NMEA par le récepteur GPS SmartGPS pour la localisation outdoor et par la suite une communication des données de localisation avec l'application centrale via Wifi est nécessaire pour les afficher sur un fond cartographique.

Plusieurs difficultés ont été rencontrées, à savoir le retard de la livraison des composants commandé, réception de quelques modules sans support, réception du chip GPS muet et les difficultés de soudage des composants.

Pour ce qui est perspective, le projet poursuivra son développement en ouvrant la voie à l'implémentation des fonctionnalités non atteintes mentionnées dans le cahier de charges, notamment le dimensionnement du bloc d'alimentation et la mise en place d'un prototype en intégrant les différents modules, comme une seconde version prototype du système (V1).

D'autres améliorations peuvent être apportées au projet. En ce qui concerne l'optimisation de la consommation de l'énergie et la réalisation d'un circuit imprimé miniature.

Le stage effectué au sein du centre ATD m'a donné l'occasion de faire le lien entre les connaissances académiques, notamment en matière de systèmes embarqués et le monde professionnel.

D'une part, il m'a permis de développer mes compétences techniques, d'approfondir mes connaissances théoriques et pratiques, de stimuler un esprit d'initiative et de créativité, et d'apprendre le métier du secteur de géolocalisation. D'autre part, l'environnement de travail, au sein d'une équipe, m'a donné l'occasion d'améliorer mon savoir-faire, de travailler avec rigueur et d'affermir un esprit d'équipe et de professionnalisme. Enfin, cette expérience a aiguisé mes capacités d'analyse, de conception et de synthèse et a surtout fortifié ma motivation, détermination et ambition.

Références

- [1] **[BENAMOUR,Avril 2011]**M.Fouad BENAMOUR, *Géolocalisation outdoor*,2011,[en ligne], date de dernière mise à jour : Avril 2011, Disponible sur : http://www.aripicardie.org/fileadmin/res/PDF/08_WiDeHouse.pdf , 1 Mars2014
- [2] **[Wikipedia]**, Disponible sur :<http://fr.wikipedia.org/wiki>, 30 Mars2014,
- [3] **[Fauconnier, 2014]** Mme. Fauconnier, *Cinq solutions françaises de géolocalisation indoor*, 2014,[en ligne], date de dernière mise à jour :9 Janvier 2014, Disponible sur : <http://www.journaldunet.com/ebusiness/commerce/geolocalisation-en-magasin/> , 11 Mars2014
- [4] **[Fillipino, 2014]** M. Dominique Fillipino, *Géolocalisation indoor, comment ça marche* ,2014 , [en ligne], date de dernière mise à jour :20 Janvier 2014, Disponible sur :<http://www.journaldunet.com/solutions/mobilite/geolocalisation-indoor.shtml>,10 Mars 2014.
- [5] **[Estimote, 2014]** Disponible sur : <http://estimote.com/estimote-for-retail.html>, 20 Mai 2014
- [6] **[Mouser, 2014]** MouserElectronics, *Composants électroniques*, 2014, [en ligne], date de dernière mise à jour : 16 juin 2014, Disponible sur : <http://www.mouser.fr/Electronic-Components/> , 20 MARS 2014.
- [7] **[Skytraq Venus 16,2012]**Skytraq Venus 16, *GPS module ST32-AM Datasheet*, 2012 , [en ligne], date de dernière mise à jour : 16 juin 2014, Disponible sur : <http://www.perthold.de/BINARY/gps-st32.pdf> , 1 juin 2014
- [8] **[Bluegiga, 2011]** Bluegiga technologies, *BLE112 Bluetooth Smart Module*, 2013, [en ligne], date de dernière mise à jour : avril 2014, Disponible sur : <https://www.bluegiga.com/en-US/products/bluetooth-4.0-modules/ble112-bluetooth--smart-module/> , 10 juin 2014.
- [9] **[Bluegiga, 2011]** Bluegiga technologies, *BLE112 Datasheet*, 2011, [en ligne], date de dernière mise à jour : 20 décembre 2011, Disponible sur : https://www.bluetooth.org/tpg/RefNotes/BLE112_Datasheet1.pdf , 10 juin 2014.
- [10] **[Invense, 2013]** Invenses technologies, *MPU-6000 and MPU-6050 Product Specification*, 2013, [en ligne], date de dernière mise à jour : 19 août 2011, Disponible sur : <http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A-00v3.4.pdf> , 13 juin 2014.

- [11] **[Embeddeb Adventures, 2013]** Embedded Adventures, *WRL-3000 V2*, 2013, [en ligne], date de dernière mise à jour : 27 juin 2013, Disponible sur : http://www.embeddedadventures.com/datasheets/WRL-3000_hw_v2_doc_v1.pdf, 13 juin 2014.
- [12] **[FIROUZ, 2011]** Mme. Ilham FIROUZ, *Les microcontrôleurs dans les systèmes embarqués*, 2013, [en ligne], date de dernière mise à jour : 30 mars 2014, Disponible sur : <http://embarque.developpez.com/cours-tutoriels/programmation/systeme-embarque/introduction-microcontroleur/>, 2 Mai 2014.
- [13] **[Texas Instruments, 2014]** Texas Instruments, *CC Debugger*, 2013, [en ligne], date de dernière mise à jour : 23 mai 2014, Disponible sur : <http://www.ti.com/tool/cc-debugger>, 28 Mai 2014.
- [14] **[Skytraq Venus, 2013]** Texas Instruments, *Binary Messages of Skytraq Venus 8 GNSS Receiver*, 2013, [en ligne], date de dernière mise à jour : 10 juillet 2013, Disponible sur : http://www.smokingresistor.com/wp-content/uploads/2013/11/AN0028_1.4.25.pdf, 8 Mai 2014.
- [15] **[I2CdevLib, 2013]** *How to decide Gyro and accelerometer offset*, 2013, [en ligne], date de dernière mise à jour : 14 décembre 2013, Disponible sur : <http://www.i2cdevlib.com/forums/topic/91-how-to-decide-gyro-and-accelerometer-offset/>, 12 juin 2014.
- [16] **[Jef Rowberg, 2014]** *BGAPI/BGLib Implementation on BLE devices*, 2014, [en ligne], date de dernière mise à jour : 27 Mars 2014, Disponible sur : <https://bluegiga.zendesk.com/entries/45890933--REFERENCE-BGAPI-BGLib-Implementation-on-BLE-devices>, 2 juin 2014.
- [17] **[Lifchetz, 2011]** M.LifchetzRenaud, *Le cryptage de Cesar*, 2014, [en ligne], date de dernière mise à jour : 6 Mai 2012, Disponible sur : <http://www.primenumbers.net/Renaud/fr/crypto/Cesar.htm>, 10 juin 2014.
- [18] **[Arduino, 2011]** *Libraries for Arduino*, 2014, [en ligne], Disponible sur : <http://playground.arduino.cc/Main/LibraryList#>, 9 juin 2014.

Annexe 1: Circuit d'adaptation MAX3232

a. Définition

La norme RS232 définit des caractéristiques électriques (niveaux des signaux), mécaniques (connecteurs) et fonctionnelles (nature des signaux). Les valeurs des tensions que les interfaces séries doivent fournir aux matériels connectés sont les suivantes (logique négative) :

- Un 0 logique est reconnu pour une tension 12V
- Un 1 logique est reconnu pour une tension -12V

Le module GPS ST32 fonctionne avec des niveaux TTL soit 0-5V, 0V pour le niveau 0 et 5V pour le niveau 1. Brancher donc directement une ligne RS232 sur le GPS ST32 n'aurait donc aucun sens et pourrait aussi endommager le système en imposant des tensions de 12V. Pour rendre compatible une ligne RS232 le GPS ST32 il existe un circuit intégré : max3232 qui est doté d'un convertisseur de tension, au travers des capacités et qui génère une tension de 12V depuis les 5V (doubleur de tension), et une tension de -12V depuis la tension 0.

b. Schéma électrique du brochage RS232

²Le brochage du composant MAX3232 avec le connecteur DB9 est décrit par la figure 74.

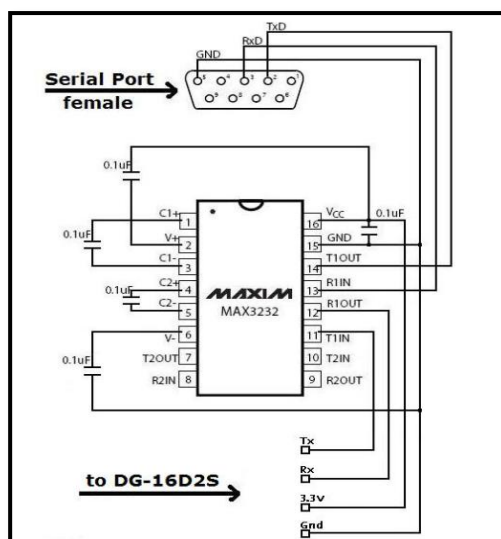


Figure 74. Schéma brochage du MAX3232

c. Schéma sous ISIS

L'implémentation du schéma fonctionnel décrit sous ISIS est le suivant :

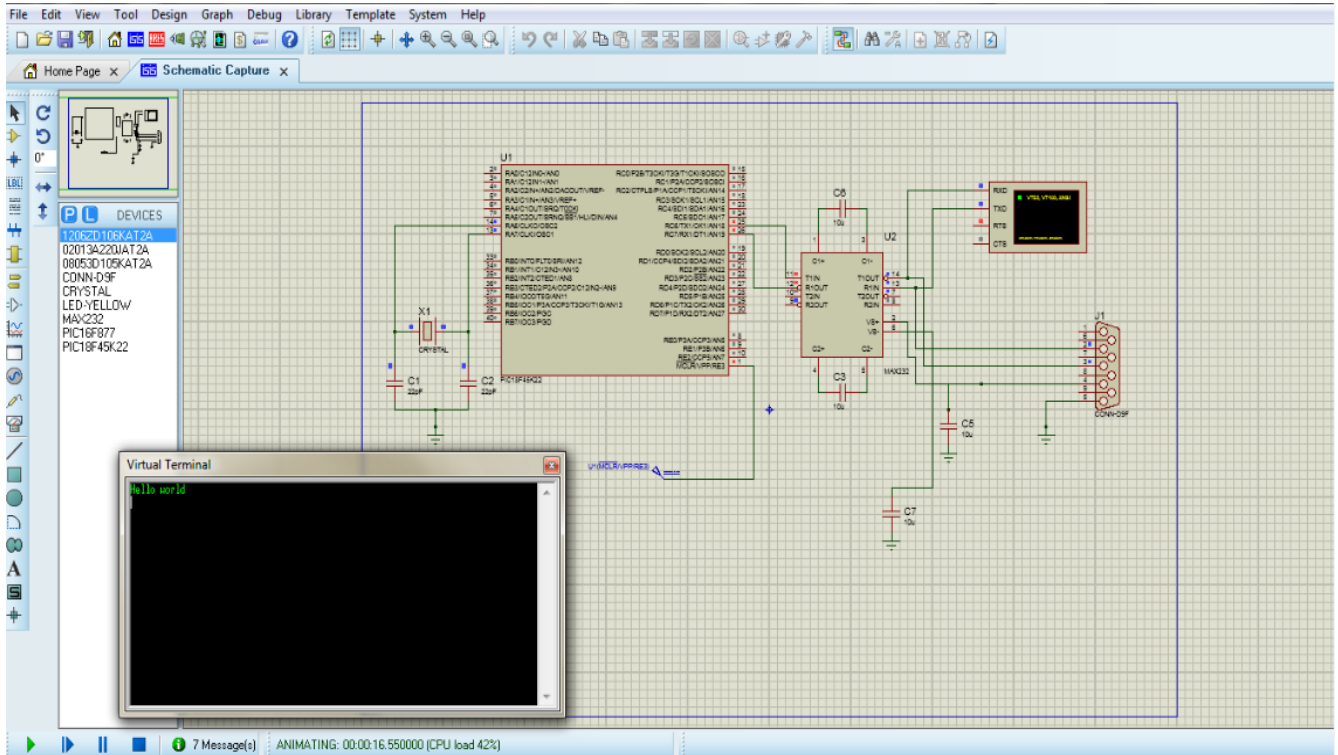


Figure 75. Simulation du Schéma MAX3232 sous ISIS

Pour vérifier le fonctionnement de l'adaptateur MAX3232 nous avons fait une simulation sur ISIS en envoyant un message 'Hello world' via le RS232 et visualisant le résultat dans un terminal virtuel.

d. Réalisation

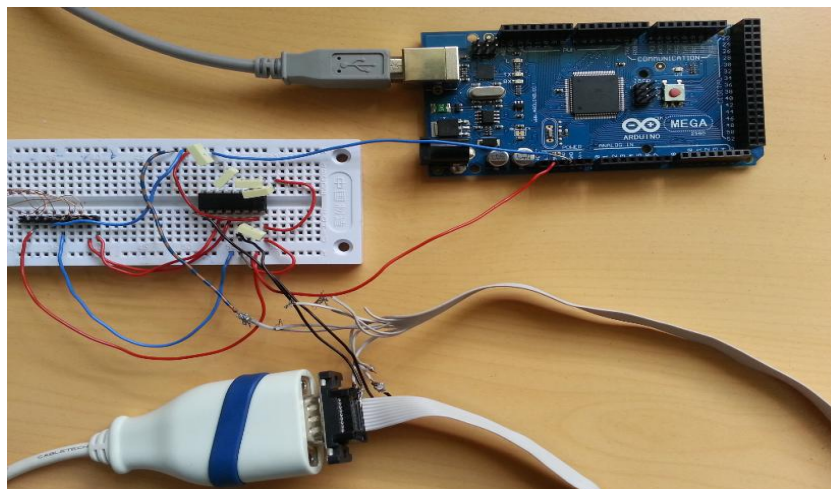


Figure 76. Branchement de l'adaptateur RS232

L'adaptateur réalisé a pour but de connecter le GPS ST32 avec le PC.

Annexe 2 : Programmation GPS ST32

→ Test avec GPS Viewer

Afin de tester le fonctionnement du récepteur GPS ST32 Skytraq, le logiciel GPS Viewer a été utilisé pour visualiser les résultats du positionnement en temps réel.

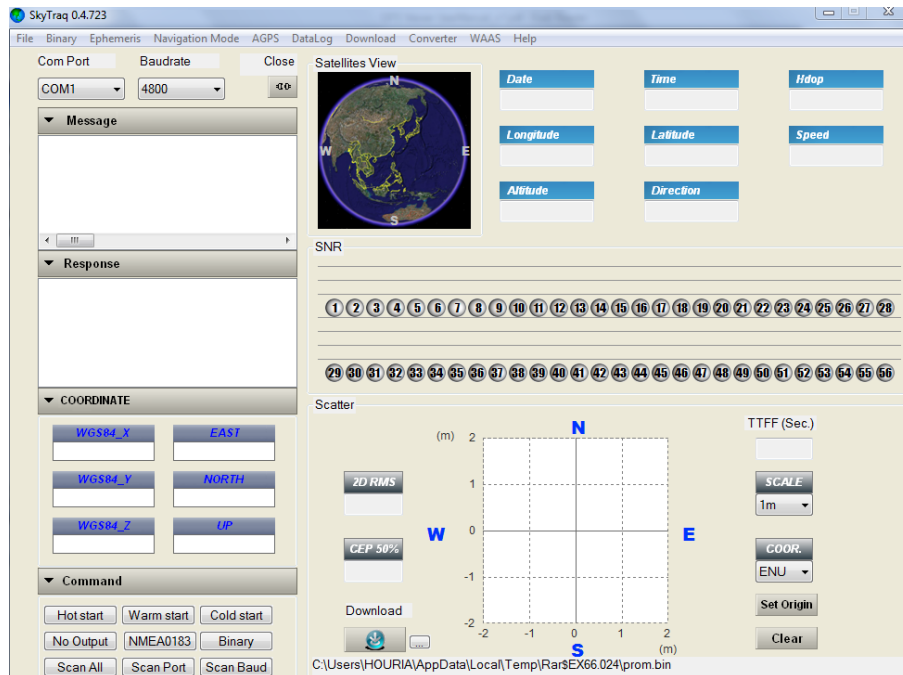


Figure 77. Interface GPS Viewer

Pour ce faire, nous avons suivi les étapes ci-dessous :

- Alimenter le GPS ST32
- S'assurer qu'il soit clair par rapport au ciel pour qu'il puisse détecter les satellites
- Connecter le GPS ST32 avec le PC via le circuit d'adaptation MAX3232 réalisé (Annexe 1) (Figure 78)

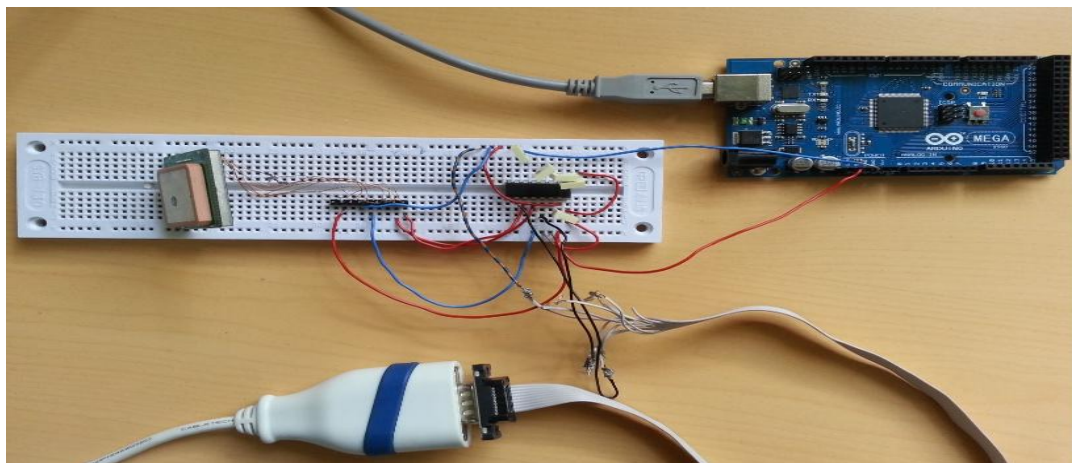


Figure 78. Branchement du ST-32 au PC à travers RS232

- Ajuster le *Baud Rate* par 38400 (proposé dans la *Datasheet*) [7]
- Sélectionner le port Com utilisé Com 9
- Cliquer sur l'icône Connect

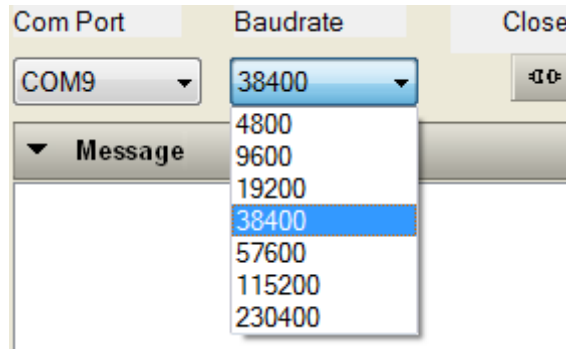


Figure 79. Configuration du Baudrate et le port Com

→ **Résultat :**

Après plusieurs essais effectués, le GPS reste inopérant et il n'a pas pu communiquer avec le PC via l'interface GPS Viewer.

→ **Test avec le protocole Skytraq Binary**

Ensuite nous sommes passés au test du module directement avec la carte Arduino (Figure 80), en utilisant le protocole de communication Skytraq Binary proposé par le manuel « Application Note AN0003 Binary Messages » [14]

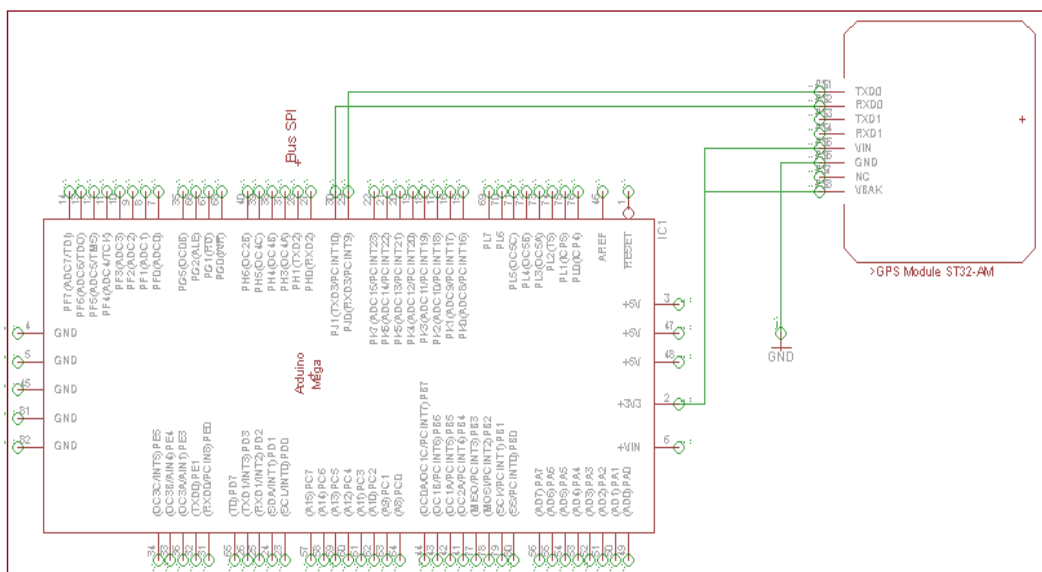


Figure 80. Branchement du GPS ST32 avec l'Arduino sous Eagle

Ce protocole sert à commander le module GPS ST32 à partir de la carte Arduino en envoyant des commandes de configuration et recevant des réponses ACK en cas de réussite de l'opération et des réponses NACK en cas d'échec.

Structure du message binaire :

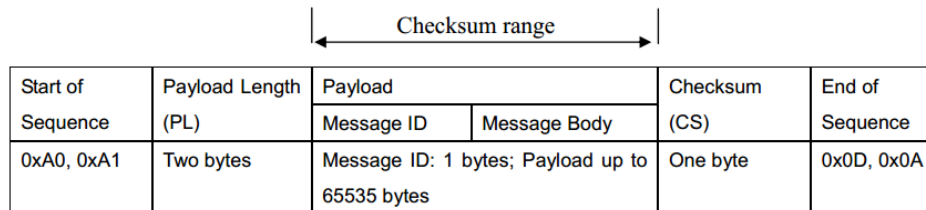


Figure 81. Structure du message binaire

Pour ce faire nous avons utilisé les commandes en hexadécimal suivantes :

- SYSTEM RESTART – Force System to restart (0x1): cette commande sert à forcer le système de redémarrer :

A0 A1 00 0F 01 01 07 D8 0B 0E 08 2E 03 09 C4 30 70 00 64 16 0D 0A

- CONFIGURE SERIAL PORT – Set up serial port property (0x5): cette commande permet de configurer le port *Serial Com* (COM1) et le *Bauderate* (34800) utilisés

A0 A1 00 04 05 00 00 00 05 0D 0A

- CONFIGURE MESSAGE TYPE – Configure and select output message type (0x9) : Cette commande permet de sélectionner le type de sortie : NMEA

A0 A1 00 03 09 00 01 09 0D 0A

Annexe 3 : SmartGPS Board

Le SmartGPS est une solution qui possède le module GPS ublox LEA-6S qui permet d'acquérir facilement des données GPS avec des réglages minimum et avec beaucoup de flexibilité



Figure 82. SmartGPS Board

Spécifications techniques du module ublox LEA-6S

- Alimentation 2.7V-3.6V
- Haut niveau d'immunité aux interférences
- Suivre simultanément jusqu'à 16 satellites
- Acquisition accélérée des signaux faibles.
- Antenne active
- Interface de communication : UART, I2C et USB
- Sensibilité : Tracking & Navigation 160 dBm
Reacquisition -160 dBm
Cold Start -147 dBm
- Protocol NMEA : GSV, RMC, GSA, GGA, GLL, VTG, TXT
- Baudrate: 9600

Annexe 4: Configuration BLE112

1. Fichier : project.bgproj

```
<?xml version="1.0" encoding="UTF-8" ?>
<project>
  <gatt in="gatt.xml" />
  <hardware in="hardware.xml" />
  <image out="out.hex" />
  <device type="ble112" />
</project>
```

2. Fichier: hardware.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<hardware>
  <sleeposc enable="true" ppm="30" />
  <sleep enable="true" />
  <script enable="false" />
  <txpower power="13" bias="5" />
  <port index="0" pull="down" tristatemask="0" />
  <port index="1" pull="down" tristatemask="0" />
  <port index="2" pull="down" tristatemask="0" />
  <wakeup_pin enable="true" port="1" pin="6" state="up" />
  <usb enable="false" endpoint="api" />
  <usart mode="packet" channel="1" alternate="1" baud="38400" endpoint="api" flow="false" />
  <!--
    UART PIN ASSIGNMENT TABLE:
    +-----+-----+-----+-----+-----+-----+
    | channel | alternate | RX | TX | RTS | CTS |
    +-----+-----+-----+-----+-----+-----+
    | 0 | 1 | P0_2 | P0_3 | P0_5 | P0_4 |
    | 0 | 2 | P1_4 | P1_5 | P1_3 | P1_2 |
    | 1 | 1 | P0_5 | P0_4 | P0_3 | P0_2 | (SELECTED)
    | 1 | 2 | P1_7 | P1_6 | P1_5 | P1_4 |
    +-----+-----+-----+-----+-----+
  -->
</hardware>
```

3. Code BGScript : BLE112 Advertiser

```
dim addr(6)

# Boot Event listener
event system_boot(major ,minor ,patch ,build ,ll_version ,protocol_version ,hw )

    #Get local BT address
    call system_address_get( ) (addr(0:6))

    # Write BT address to DI service serial number string
    call attributes_write(xgatt_dis_2a25,0,6,addr(0:5))

    #set to advertising mode
    call gap_set_mode(gap_general_discoverable,gap_undirected_connectable)

    #set bondable mode
    call sm_set_bondable_mode(1)

end

# Disconnection event listener
event connection_disconnected(handle,result)
    #connection disconnected, continue advertising
    call gap_set_mode(gap_general_discoverable,gap_undirected_connectable)
end
```