



**Universidade do Minho**  
Escola de Engenharia

Fábio Valter Inácio Martins

**Sistema de Aquisição e Teste de Sinais GNSS para  
Condução Autónoma**

Dissertação de Mestrado

Mestrado em Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação de  
Professor Doutor Luís Alexandre Rocha

Outubro de 2016

## DECLARAÇÃO

**Nome:** Fábio Valter Inácio Martins

**Endereço eletrónico:** a65313@uminho.pt      **Contacto:** +351918864505

**Cartão do Cidadão:** 14501414

**Título da dissertação:** Sistema de Aquisição e Teste de Sinais GNSS para Condução Autónoma

**Orientador:**

Professor Doutor Luís Alexandre Rocha

**Ano de conclusão:** 2016

Mestrado em Engenharia Eletrónica Industrial e Computadores

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, \_\_\_\_/\_\_\_\_/\_\_\_\_

Assinatura:

*“Take care of all your memories.*

*For you cannot relive them. “*

**Bob Dylan**



## AGRADECIMENTOS

Em primeiro queria agradecer aos meus pais, Inês Martins e Artur Martins, pelo apoio incondicional ao longo da minha vida e por, mesmo em tempos difíceis, sempre me possibilitarem as condições necessárias para finalizar o meu curso e ao meu irmão, Diogo Martins, pela boa disposição sempre presente.

Ao meu orientador, professor doutor Luís Alexandre Rocha, pelo apoio incansável, pela confiança e por toda a sabedoria transmitida no decorrer desta dissertação.

Aos meus amigos e colegas de trabalho, Carlos Ferreira, Carlos Silva e Fábio Leitão com quem partilhei o percurso académico, pelo espírito de companheirismo, pela boa disposição, pela ajuda sempre disponível e pela partilha constante de conhecimentos, sem eles o curso não teria sido o mesmo.

Ao Doutor Hélder Silva e ao Engenheiro Vasco Lima por toda a ajuda e disponibilidade que em muito contribuíram para a realização deste trabalho.

Por último, mas não menos importante, à minha namorada, Susana Costa, por todo o apoio nos momentos mais difíceis e pela paciência pelas muitas horas ocupadas pelo trabalho.

A todos vocês, o meu muito OBRIGADO.

Este trabalho foi financiado pelo projeto “INNOVCAR: Inovação para Veículos Inteligentes”, n°002797, cofinanciado pelo FEDER através do Portugal 2020 – Programa Operacional Competitividade e Internacionalização (COMPETE2020).



## RESUMO

Esta dissertação pretende desenvolver um sistema capaz de testar e adquirir sinais GNSS (*Global Navigation Satellite System*) a inserir em sistemas de condução autónoma.

Atualmente, tem existido um aumento significativo no desenvolvimento de sistemas de automatização baseados em câmaras de vídeo, radares, unidades de navegação inercial, navegação através de localização por satélites e sensores internos do veículo. A maioria destes sistemas é atualmente utilizada para assistir o condutor, prevenir acidentes rodoviários e tornar a condução mais confortável.

Estes sistemas avançados de assistência ao condutor estão atualmente implementados em sistemas como por exemplo *“adaptive cruise control”*, *“lane keeping”* e *“parking assistance”*. Alguns fabricantes incluíram já sistemas mais automatizados como *“Automatic Park Assist”*, *“Integrated Cruise assist”* e *“Traffic Jam assist”* [1].

Num futuro próximo, o objetivo é desenvolver um veículo totalmente automatizado que é capaz de se controlar sozinho e tomar decisões em todas as situações possíveis. Para o alcançar, todos estes sensores têm de ser otimizados, assegurando fiabilidade trabalhando em conjunto.

Todavia, os sistemas de navegação por satélite têm sérios problemas de fiabilidade, especialmente em ambientes urbanos, túneis e grandes obstáculos. Além disso, sistemas baseados em satélites contam também com falhas a nível de precisão devido a atrasos, interferências e reflexões.

Por isso, e para resolver estas dificuldades, têm sido desenvolvidas soluções multi-GNSS que podem combinar GPS, GLONASS, GALILEO, entre outros. Devido a esta evolução, surgiu a necessidade de sistemas capazes de testar recetores GNSS, para que seja possível caracterizá-los.

Assim, o sistema desenvolvido nesta dissertação é capaz de calcular a precisão, contar o tempo necessário para adquirir a primeira posição nos diferentes tipos de arranque e medir a intensidade do sinal necessária para adquirir a primeira posição e a intensidade para manter o recetor a adquirir posições.

Para fazer isto, é necessário gerar um sinal conhecido através de um simulador GNSS, configurar o recetor GNSS e calcular as métricas usando os dados recebidos do recetor.

Os resultados obtidos validam o método proposto, tendo sido possível caracterizar com sucesso um recetor de GNSS.

**Palavras- passe:** GNSS, condução autónoma, caracterização GNSS, fiabilidade





## ABSTRACT

This dissertation aims to develop a system capable of testing and acquiring GNSS (Global Navigation Satellite System) signals with the goal of autonomous driving.

Nowadays, there has been a sharp increase in works targeting the development of automated systems, based on video cameras, radars, INs (Inertial Navigation Systems), satellite navigation systems and in-vehicle sensors. The majority of these systems are currently used to assist the driver, preventing road accidents and making driving more comfortable. This kind of advanced driver assistance systems is present, for example, in systems such as “adaptive cruise control”, “lane keeping” and “parking assistance”. Some car manufacturers have already included more automated systems, as “Automatic Park assist”, “Integrated Cruise assist” and “Traffic Jam assist” [1].

In a near future, the objective is to achieve a fully automated vehicle, which is able to control itself and make all the decisions in every situation. To accomplish that, all these sensors have to be optimized, ensuring reliability while working together. However, the satellite navigation has serious problems with reliability, particularly in urban environments, tunnels and big obstacles. Furthermore, satellite based systems have also a lack of accuracy due to delays, interferences and multipath of the signal.

Therefore, to solve these difficulties, some multi-GNSS solutions have been developed which might combine GPS, GLONASS, GALILEO, among others. Due to this evolution, a need of systems capable of testing GNSS receptors have arisen, so it can be possible to characterize them.

Thus, the system developed in this dissertation is able to calculate the accuracy, to count the required time to acquire the first position in different start modes and measure the signal intensity needed to acquire the first position and the intensity to maintain the receptor obtaining positions. To do this, it is necessary to generate a known signal through a GNSS simulator, be able to configure the GNSS receptor and calculate the metrics using all the data received from the receptor.

The results obtained validate the proposed method and it was possible to characterize an existing GNSS receiver.

**Keywords:** GNSS, autonomous driving, GNSS characterization, reliability



# CONTEÚDO

1. Introdução.....	1
1.1. Enquadramento.....	2
1.2. Motivação.....	3
1.3. Objetivos da dissertação.....	4
1.4. Estrutura da dissertação.....	4
2. Estado da Arte e Enquadramento Teórico .....	7
2.1. GNSS (Global Navigation Satellite System) .....	7
2.1.1. História.....	7
2.1.2. Visão Geral do funcionamento .....	7
2.1.3. Segmentos GNSS .....	8
2.1.4. Espectro de frequências.....	9
2.1.5. Modelação sinais GNSS.....	9
2.1.6. Geração do código pseudoaleatório .....	10
2.1.7. Formato das mensagens de Navegação .....	11
2.2. Recetores GNSS.....	12
2.2.1. Calculo da distância recetor - satélite .....	12
2.2.2. Desmodulação da mensagem de navegação.....	14
2.2.3. Visão geral de um recetor GNSS.....	14
2.3. Condução Autónoma .....	15
2.4. Teste e Verificação de Recetores GNSS.....	17
2.5. Sistema de coordenadas .....	18
3. Sistema de teste para o APP ( <i>Automotive Precise Positioning</i> ) .....	21
3.1. Visão Geral do Sistema.....	21
3.2. Requisitos.....	22
3.3. Restrições.....	22
3.4. Hardware .....	22
3.4.1. Microcontrolador .....	23
3.4.2. Recetor GNSS.....	23

3.4.3.	Gerador GNSS .....	25
3.4.4.	Interface CAN-Bus para USB .....	26
3.4.5.	Transceiver CAN-Bus .....	26
3.5.	Protocolos de comunicação.....	27
3.5.1.	SPI .....	27
3.5.2.	NMEA .....	28
3.5.3.	CAN.....	30
3.5.4.	UBX.....	32
3.6.	Métricas de testes do recetor GNSS .....	34
3.6.1.	TTFF – <i>Time To First Fix</i> - Arranque de fábrica [22] .....	34
3.6.2.	TTFF – <i>Time To First Fix</i> - Arranque normal.....	34
3.6.3.	TTFF – <i>Time To First Fix</i> - Arranque em Standby.....	34
3.6.4.	Precisão de localização – Relativa .....	35
3.6.5.	Precisão de localização – Absoluta.....	35
3.6.6.	Precisão de localização – Dinâmica.....	36
3.6.7.	Tempo de Reaquisição .....	36
3.6.8.	Sensibilidade – para aquisição .....	36
3.6.9.	Sensibilidade – em funcionamento.....	36
3.6.10.	Interferências .....	37
4.	Software.....	39
4.1.	Microcontrolador .....	39
4.1.1.	Rotina principal .....	39
4.1.2.	Rotina de leitura do recetor GNSS .....	40
4.1.3.	Rotina de envio através de CAN.....	41
4.1.4.	Comunicação SPI (Recetor GNSS – Microcontrolador) .....	43
4.1.5.	Rotina de configuração do recetor GNSS .....	43
4.2.	Computador .....	44
4.2.1.	Interface .....	44
4.2.2.	Rotina Principal .....	45
4.2.3.	Processamento de dados NMEA.....	46

4.2.4.	Rotinas de início e terminação da recepção CAN-Bus.....	47
4.2.5.	Rotina de configuração do recetor GNSS.....	47
4.2.6.	Rotina de apresentação do mapa .....	49
4.2.7.	TTFF – Time To First Fix.....	50
4.2.8.	Precisão de localização .....	50
4.2.9.	Tempo de Reaquisição .....	52
4.2.10.	Sensibilidade .....	52
4.2.11.	Interferências .....	53
5.	Testes e Resultados .....	55
5.1.	Teste aos componentes do sistema .....	55
5.1.1.	Recetor GNSS.....	55
5.1.2.	Comunicação CAN-Bus.....	57
5.1.3.	Comunicação SPI.....	57
5.1.4.	Aquisição de posições .....	57
5.1.5.	Precisão .....	58
5.1.6.	Precisão Absoluta.....	59
5.1.7.	Posição Relativa .....	59
5.1.8.	TTFF – Tempo para a primeira aquisição.....	60
5.1.9.	Tempo de reaquisição .....	60
6.	Conclusões e Trabalho Futuro.....	61
7.	Referências Bibliográficas .....	63



## LISTA DE FIGURAS

Figura 1 - Visão geral do projeto APP .....	2
Figura 2 - Visão futurista de condução autônoma [4] .....	4
Figura 3 - Triangulação 3D [7] .....	8
<i>Figura 4 - Segmentos GNSS (adaptado de [8] )</i> .....	8
Figura 5 - Espectro de frequências [8] .....	9
Figura 6 - Sinais transmitidos pelos satélites [9] .....	10
Figura 7 - Exemplo de modulação BPSK [10].....	10
Figura 8 - Geração do código pseudoaleatório para código C/A .....	11
Figura 9 - Estrutura da mensagem de navegação GPS [9] .....	12
Figura 10 - Cálculo da distância através do atraso do sinal [11] .....	13
Figura 11 - Resultado da correlação entre o sinal do satélite e o sinal do recetor [8] .....	13
Figura 12 - Sinal contendo informação modulada em BPSK .....	14
Figura 13 - Informação extraída do sinal modulado da Figura 12.....	14
Figura 14 - Diagrama de blocos de um recetor GNSS [12] .....	15
Figura 15 - Evolução dos sistemas de condução autônoma e sensores utilizados [3] .....	17
Figura 16 - Visão Geral do sistema de Testes R&S [14] .....	17
Figura 17 - Representação Latitude e Longitude [15].....	19
Figura 18 - Visão geral do sistema de teste GNSS.....	21
Figura 19 - Montagem do hardware utilizado .....	23
Figura 20 - Kit de desenvolvimento uBlox M8T [16] .....	24
Figura 21 - Gerador GNSS LabSat3.....	25
Figura 22 - PEAK PCAN Pro .....	26
Figura 23 - Diagrama de conexões DB9 PEAK PCAN Pro.....	26
Figura 24 - Representação de uma mensagem CAN TTL e diferencial .....	26
Figura 25 - Conexões master - <i>slave</i> SPI [18].....	27
Figura 26 - Conexão entre vários dispositivos SPI [18] .....	27
Figura 27 - Simples comunicação SPI [18] .....	28
Figura 28 - Rede CAN-Bus [21] .....	30
Figura 29 - Transformação do sinal TTL para diferencial [21] .....	30

Figura 30 - Formato da trama CAN-Standard [21].....	31
Figura 31 - Formato da trama CAN-Extended [21] .....	32
Figura 32 - Exemplificação da medição da posição relativa do recetor GNSS. É considerada a distância entre as posições adquiridas (pontos verdes).....	35
Figura 33 - Exemplificação do teste à posição absoluta, sendo os pontos verdes as posições adquiridas, o vermelho a posição real do recetor e as setas as distâncias medidas entre eles.....	35
Figura 34 - Fontes de interferências do sinal GNSS (adaptado de [23]) .....	37
Figura 35 - Diagrama de blocos da sequência de software principal do microcontrolador.....	40
Figura 36 - Diagrama de blocos da sequência de software da receção da informação GNSS do recetor e posterior reenvio por CAN .....	41
Figura 37 - Diagrama de blocos da sequência de software do envio de informação .....	42
Figura 38 - Diagrama de blocos da rotina de configuração do recetor GNSS .....	44
Figura 39 - Interface gráfica do sistema de testes .....	45
Figura 40 - Sequencia da rotina principal do software no computador .....	45
Figura 41 - Rotina de processamento das mensagens NMEA recebidas por CAN-BUS .....	46
Figura 42 - Rotina de inicio de receção de dados NMEA recebidos por CAN .....	47
Figura 43 - Rotina de fim de receção de dados NMEA recebidos por CAN .....	47
Figura 44 - Rotina de geração e envio da mensagem UBX de configuração do recetor GNSS .....	48
Figura 45 - Rotina para apresentação do mapa com a ultima posição adquirida .....	49
Figura 46 - Rotina para calcular o tempo para primeira aquisição do recetor nos diferentes tipos de arranque .....	50
Figura 47 - Distância medida entre dois pontos do mapa (utilizando o Google Maps – 187,34m) .....	51
Figura 48 - Resultado do cálculo da formula Haversine no Matlab (187.2m) .....	52
Figura 49 -Diagrama de blocos da sequencia do teste de sensibilidade .....	53
Figura 50 - Interface gráfica implementada em C# com informação extraída das mensagens NMEA ..	55
Figura 51 - Interface gráfica implementada em C# de configuração do recetor GNSS através do protocolo UBX .....	56
Figura 52 - Exemplo de utilização da informação recebida do recetor GNSS.....	56
Figura 53 - Posição Gerada apresentada no Google Maps .....	58
Figura 54 - Mapa gerado no Matlab utilizando a posição adquirida pelo recetor GNSS .....	58
Figura 55 - Gráfico com a diferença entre a posição gerada e as posições adquiridas .....	59
Figura 56 - Gráfico de precisão absoluta, distância entre a posição gerada e as posições adquirida....	59



## LISTA DE TABELAS

Tabela 1 - Definição dos parâmetros de caracterização de um recetor GNSS .....	18
Tabela 2 - Características de funcionamento do recetor GNSS NEO-M8T [17] .....	24
Tabela 3 - Sensibilidades do recetor GNSS NEO-M8T [17] .....	25
Tabela 4 - Características técnicas do recetor GNSS NEO-M8T [17] .....	25
Tabela 5 - Formato da trama NMEA.....	29
Tabela 6 - Identificadores NMEA [20].....	29
Tabela 7 - Tabela de formato da mensagem 'ZDA', exemplo NMEA [20].....	29
Tabela 8 – Formato da trama UBX .....	32
Tabela 9 - Descrição das classes do protocolo UBX .....	33
Tabela 10 - Exemplificação da análise de uma mensagem UBX .....	33
Tabela 11 - Decomposição da mensagem NMEA em mensagens CAN.....	42
Tabela 12 - Exemplo de uma mensagem de configuração enviada por SPI para o recetor GNSS.....	44
Tabela 13 - Método de geração do Byte de sincronismo das mensagens CAN.....	49
Tabela 14 - Divisão da mensagem UBX em mensagens CAN-BUS .....	49



## LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

- ° - Grau;
- cm – centímetro;
- Hz – Hertz;
- s – segundo;
- ABS – *Anti-lock Break System*;
- ASIC - *Application Specific Integrated Circuits*;
- BPSK – *Binary Phase-shift keying*;
- CAN - *Controller Area Network*;
- CDMA – *Code Division Multiple Access*;
- CLL - *Code Lock Loop*;
- C/A - Coarse/acquisition;
- ESP – *Electronic Stability Program*
- GLONASS - *GLObalnaya NAVigatsionnaya Sputnikovaya Sistema*;
- GNSS – *Global Navigation Satellite System*;
- GPS – *Global Positioning System*;
- IRNSS - *Indian Regional Navigation Satellite System*;
- ITU - *The Internacional Telecommunications Union*;
- LFSR - *Linear Feedback Shift Registers.*;
- NMEA - *The National Marine Electronics Association*;
- PRN – *Pseudo Random Noise*;
- QZSS - *Quasi-Zenith Satellite System*;
- RF – Rádio Frequência;
- RISC - *Reduced Instruction Set Computer*;
- SPI – *Serial Peripheral Interface*;
- TTFF – *Time To First Fix*;
- TTL - *Transistor-Transistor Logic*;
- UART - *Universal Asynchronous Receiver/Transmitter*;
- UBX – *uBlox*;
- USB – *Universal Serial Bus*;



# 1. INTRODUÇÃO

Os sistemas de navegação por satélite começaram a ser desenvolvidos em 1973 pelos Estados Unidos como uma mais valia para os militares, permitindo saber a localização dos mesmos e o direcionamento de mísseis guiados por GPS – *Global Positioning System*. Consequentemente, outros países iniciaram o desenvolvimento dos seus próprios sistemas, como a Rússia e a China.

Em 1983 o sistema GPS disponibilizou o serviço para civis [2]. A partir deste momento, começaram a surgir inúmeras aplicações que utilizavam o sistema de navegação por satélite, como os sistemas de navegação para automóveis e sistemas de controlo de estruturas.

Estes acontecimentos, alteraram o mundo e o modo como as pessoas se deslocam, facilitando o trabalho das empresas de transporte, das viagens pessoais, do controlo aéreo, entre muitas outras, tornando possível o mundo como é atualmente.

Outra tecnologia que se desenvolveu igualmente em grande escala nos últimos anos foi a assistência ao condutor de automóveis [3]. Sistemas como o ABS, ESP, *AirBag*, travagem automática e outros, estes são responsáveis pela poupança de milhares de vidas todos os anos, pois evitam milhões de acidentes rodoviários e aumentam a segurança dos ocupantes. Com o crescente desenvolvimento da tecnologia e do aumento de veículos em circulação, surgiram novas necessidades de segurança rodoviária, então sistemas de condução autónoma começaram a ser desenvolvidos, como sistemas para seguir o veículo imediatamente á frente, para manter o carro entre as linhas, entre outros sistemas que visam reduzir a maior causa de acidentes rodoviários em todo o mundo: o condutor.

Alcançou-se agora o momento em que as duas tecnologias supracitadas se devem fundir, o sistema de navegação por satélite e os sistemas de assistência e automação dos veículos.

Sendo o objetivo desta fusão, num futuro próximo, o desenvolvimento de veículos completamente autónomos, isto é, sem a necessidade de intervenção do condutor.

Para isso é necessário que os sistemas de navegação por satélite atinjam precisões na ordem dos centímetros, para que, em conjunto com os sistemas atuais, permita a localização absoluta do veículo com o máximo de precisão possível.

Contudo, atualmente ainda não existem sistemas de navegação por satélite autónomos (sem estações de referência próximas) com este nível de precisão. Aqui se insere o âmbito desta dissertação, conseguir analisar e testar intensivamente os recetores GNSS – *Global Navigation Satellite System* em desenvolvimento, caracterizando-o com diferentes ambientes, constelações, sinais e outras variáveis,

procurando encontrar as melhores combinações e os melhores recetores para tornar possível o objetivo de num futuro próximo existirem veículos completamente autónomos.

## 1.1. Enquadramento

Devido ao crescente desenvolvimento de tecnologias relacionadas com condução autónoma surgiram requisitos relativamente à posição absoluta mais exigentes.

Assim, o objetivo principal desta dissertação, inserida no projeto Bosch-UMinho (*Innovative Car*), é a implementação de um sistema capaz de adquirir sinais GNSS e testar os recetores dos mesmos para posteriormente serem incorporados em sistemas de elevada precisão ligados ao mundo automóvel.

Para o teste, o sistema injeta um sinal GNSS no recetor e verifica a resposta do mesmo, caracterizando-o posteriormente quanto à sua precisão, sensibilidade, atraso, entre outros parâmetros, e assim é possível analisar que combinações de sinais otimizam as características do sistema.

Atualmente não existem recetores com capacidade para adquirir e processar sinais GNSS de diferentes bandas em simultâneo, por isso, o sistema está implementado com um recetor apto a adquirir sinais de diferentes constelações (GPS, GLONASS, BeiDou, QZSS) mas sempre na banda L1 que é aproximadamente 1500 MHz.

Contudo, está preparado para futuramente ser facilmente implementada a receção e testes com recetores capazes de adquirir diferentes bandas.

Como pode ser observado na Figura 1, este sistema está integrado no elemento “Sensores de posição” do projeto APP - *Automotive Precise Positioning*, que em conjunto com outros sensores visa determinar a posição relativa e absoluta de um veículo com precisões na ordem das dezenas de centímetros.

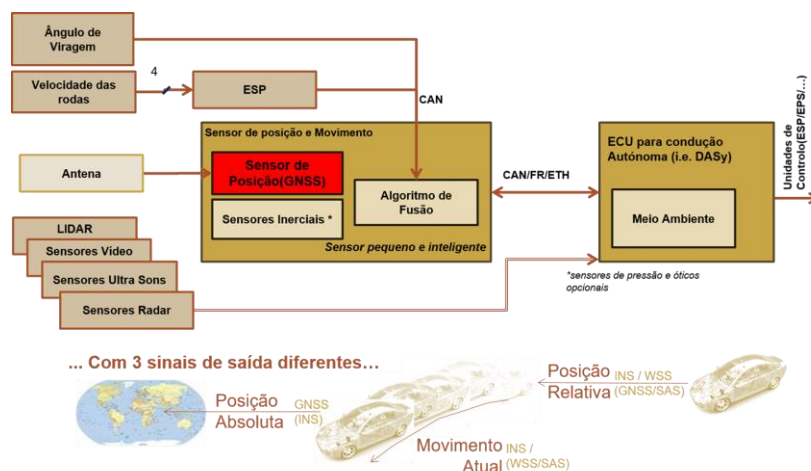


Figura 1 - Visão geral do projeto APP

## 1.2. Motivação

Nos últimos anos o crescimento de sistemas de assistência ao condutor tem revelado um crescimento exponencial, com resultados encorajadores e que em muito têm contribuído no combate aos acidentes rodoviários devido a erro humano.

As soluções atuais, como ABS, ESP e outros, ocupam um papel fundamental na prevenção rodoviária, contudo a tendência é a automatização da condução com sistemas que seguem o carro da frente em filas de trânsito, travagem automática aquando da deteção de um obstáculo, entre outros sistemas.

Assim, o objetivo deste desenvolvimento é a condução de veículos, em todos os ambientes, completamente automatizada, sem necessidade de qualquer intervenção humana, à semelhança da visão futurista apresentada na Figura 2. Contudo, nos dias que correm ainda não existe nenhum sistema completamente fiável de localização absoluta e relativa do veículo, devido a vários tipos de erros como a perda de sinal, atrasos, entre outros, nem um sistema com precisão na ordem dos centímetros a baixo custo, que é um fator essencial quando o objetivo é a massificação na indústria automóvel. Nesse sentido, a motivação desta dissertação será o desenvolvimento do sistema para aquisição de sinais GNSS e o teste dos recetores dos mesmos com vista a descobrir a melhor opção com precisão na ordem das dezenas de centímetros, baixo custo de produção e completamente fiável.

O projeto APP da Bosch, onde alguns elementos são desenvolvidos em parceria com a Universidade do Minho, no qual esta dissertação se insere pretende obter a localização relativa e absoluta através de um vasto conjunto de sensores, ilustrados na Figura 1, e para isso, é necessário o uso de um microprocessador desenvolvido para a indústria automóvel, fiável, resistente e com elevado poder de processamento. O microprocessador escolhido pela Bosch é da empresa Renesas, da família RH850 e cumpre os requisitos enumerados.

Os dados relativos à posição absoluta adquiridos no sistema através dos sinais de localização por satélite serão fundidos com os sinais provenientes de uma unidade inercial, com o ângulo de viragem das rodas e com a velocidade de cada uma delas o que permite obter a posição relativa, possibilitando assim a aproximação da posição exata do veículo na ordem das dezenas de centímetros.



Figura 2 - Visão futurista de condução autónoma [4]

### 1.3. Objetivos da dissertação

O objetivo geral da dissertação é desenvolver um sistema capaz de adquirir sinais GNSS, testar e validar os recetores dos mesmos. O sistema que se pretende obter visa garantir a fiabilidade dos recetores a serem posteriormente utilizados no projeto APP, que pretende alcançar a posição mais exata possível de veículos para que, futuramente seja possível a condução de veículos sem necessidade de intervenção humana.

O objetivo global da dissertação pode ser subdividido em objetivos mais pequenos, apresentados de seguida:

- Obtenção da posição absoluta utilizando diferentes constelações (GPS, GLONASS, BeiDou, IRNSS);
- Testar e caraterizar o recetor de sinais GNSS;
- Exportar dados da caraterização do recetor GNSS;
- Transmitir informação, ao algoritmo de fusão com a unidade inercial, com a maior rapidez e fiabilidade possível;

### 1.4. Estrutura da dissertação

O capítulo um desta dissertação é de caráter introdutório. Este enquadra o tema ao leitor expondo a motivação e os objetivos que se pretendem alcançar. No segundo capítulo é apresentado o estado da arte relativo ao desenvolvimento de sistemas de condução autónoma, recetores e métodos de caraterização dos mesmos, e o enquadramento teórico referente ao sistema global de localização por satélite (GNSS).

Por sua vez, no capítulo terceiro é explicado o sistema APP, onde é detalhado o *hardware* escolhido para o desenvolvimento do projeto, os protocolos utilizados para a comunicação entre o



microcontrolador, recetor e computador e as métricas sobre as quais o recetor GNSS deve ser caracterizado.

De seguida, é esmiuçado o *software* desenvolvido para a implementação do sistema de testes, este é apresentado separadamente entre o software do microcontrolador e o do computador.

No quinto capítulo, são exibidos os resultados obtidos com o sistema desenvolvido, onde é exposto o teste efetuado a um recetor GNSS, caracterizando-o.

Por fim, o capítulo seis é dedicado às conclusões relativas ao desenvolvimento da dissertação, apresentado as maiores dificuldades e o que pode se melhorado no futuro.



## 2. ESTADO DA ARTE E ENQUADRAMENTO TEÓRICO

### 2.1. GNSS (Global Navigation Satellite System)

#### 2.1.1. História

Os sistemas de localização global por satélite foram inicialmente desenvolvidos em 1973 pelos Estados Unidos através do projeto *US Defense Navigation Satellite* sendo na data exclusivamente para uso militar, projeto este que foi mais tarde renomeado como *NAVigation Satellites providing Time and Range* (NAVSTAR) e finalmente *Global Positioning System* (GPS).

O primeiro satélite foi lançado em 1978 e, em 1993, alcançaram os 24 satélites necessários para a cobertura total do planeta, atualmente são utilizados 27 para melhor precisão, tendo ainda disponíveis 4 satélites suplentes [5].

O sistema foi inicialmente desenvolvido para guiar mísseis, estando disponível para uso civil em 1983. Seguiu-se mais tarde, em 1982, o lançamento do primeiro satélite do sistema GLONASS [2], contando atualmente com 24 dos 29 satélites previstos. Atualmente, está em desenvolvimento o sistema de navegação europeu, onde estão em órbita 11 dos 30 satélites previstos para o funcionamento nominal, assim como o sistema Chinês (BeiDou/COMPASS) que já conta com 20 dos 35 satélites necessários para a cobertura global.

#### 2.1.2. Visão Geral do funcionamento

O funcionamento base dos sistemas que constituem o Sistema de Navegação por Satélite Global é bastante semelhante, e consiste na colocação dos satélites em órbitas estratégicas para que qualquer ponto do globo tenha sempre ao seu alcance pelo menos os quatro satélites necessários para cálculo da posição.

A posição é calculada através da descodificação dos sinais emitidos pelos quatro satélites, de onde é possível retirar a distância do recetor a cada um dos satélites, assim como a posição exata dos satélites. Posteriormente, recorrendo a um algoritmo de triangulação é possível obter a posição exata do recetor, explicado graficamente na Figura 3.

Este seria o funcionamento ideal, contudo, não é ainda possível a civis obter a sua posição exata, devido a fatores como os atrasos na ionosfera, reflexões do sinal, erros nos recetores, interferências no

sinal, entre outros. Assim, nos últimos anos tem-se acentuado o desenvolvimento de soluções para esta falta de precisão e fiabilidade [6], como a fusão entre os sinais de diferentes constelações, sistemas para correção de erro e criação de mais sinais disponíveis para civis.

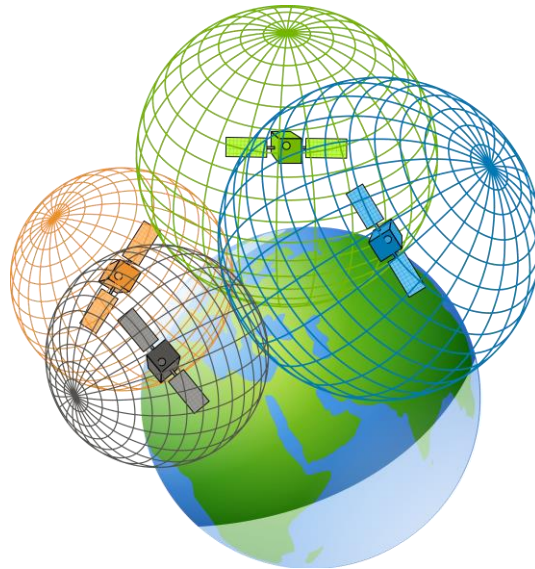


Figura 3 - Triangulação 3D [7]

### 2.1.3. Segmentos GNSS

O sistema GNSS está dividido em três segmentos principais: espacial, controlo e utilizador. Como pode ser observado na Figura 4, o segmento espacial inclui os satélites em órbita a transmitir sinais controlados por relógios atómicos (rubídio e hidrogénio) com estabilidades na ordem de 1 a 10 nS por dia. O segmento de controlo contém um conjunto de estações de monitorização espalhadas pelo globo que fornecem correções para o relógio dos satélites, mensagens de navegação e órbita. O segmento do utilizador é composto pelos recetores que fornecem posição e navegação aos utilizadores.

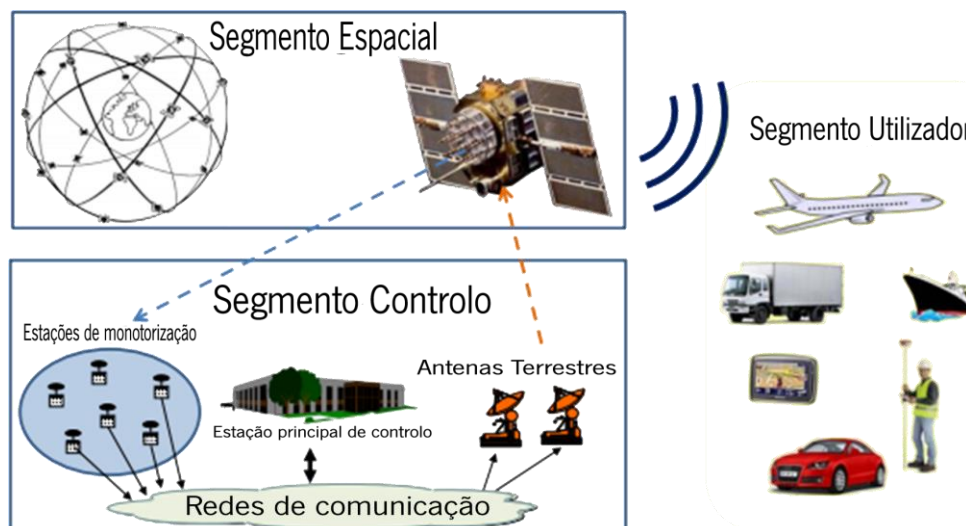


Figura 4 - Segmentos GNSS (adaptado de [8])

## 2.1.4. Espectro de frequências

Os satélites transmitem sinais na banda L em diferentes frequências, estes sinais contêm o código PRN – *Pseudo Random Noise* e dados de navegação. O código PRN é uma sequência de bits que permitem a determinação da distância do satélite ao recetor. A mensagem de navegação contém informação sobre a posição dos satélites, *Almanac*, *Ephemeris*, desvios do relógio, estado do sistema, entre outros. O espectro de frequências é regulado pela ITU – *The International Telecommunications Union* e os sistemas de navegação por satélite estão alocados na banda L, como pode ser observado na Figura 5. Para ser possível as diferentes constelações enviarem sinais na mesma frequência é utilizada a técnica CDMA – *Code Division Multiple Access*, onde cada satélite tem o seu próprio código pseudoaleatório. Desta forma, é possível ao recetor identificar de que satélite o sinal é proveniente.

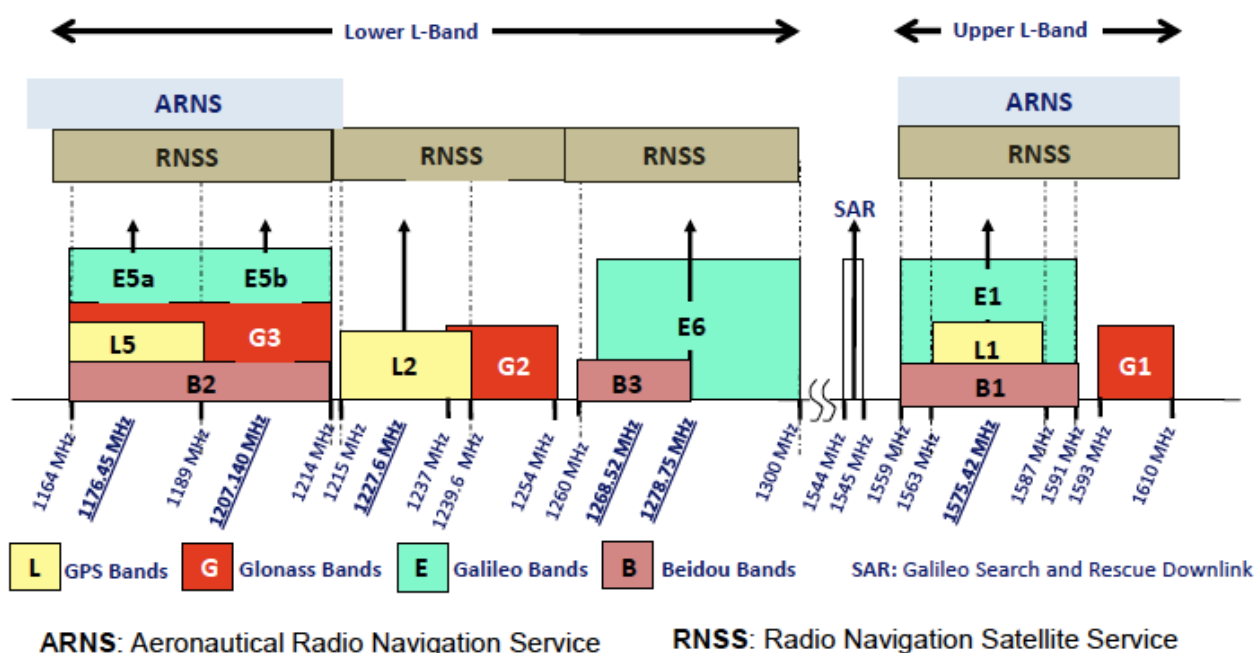


Figura 5 - Espectro de frequências [8]

## 2.1.5. Modelação sinais GNSS

A informação a ser transmitida dos satélites para os recetores tem uma frequência de 50 Hz, sendo que a frequência portadora está situada na banda L, entre os 1164 MHz e 1610 MHz, excluindo as frequências entre 1300 MHz e 1559 MHz, como pode ser observado na Figura 5. Para além da mensagem de navegação, é ainda transmitido um código pseudoaleatório que é utilizado para a medição da distância do satélite ao recetor.

O código pseudoaleatório é somado com a mensagem de navegação, e posteriormente este sinal é modulado em BPSK para ser transmitido. A mesma mensagem de navegação é enviada através de

diferentes sinais utilizando códigos pseudoaleatórios diferentes e em frequências portadoras diferentes, criando assim as diferentes bandas, como pode ser observado graficamente na Figura 6.

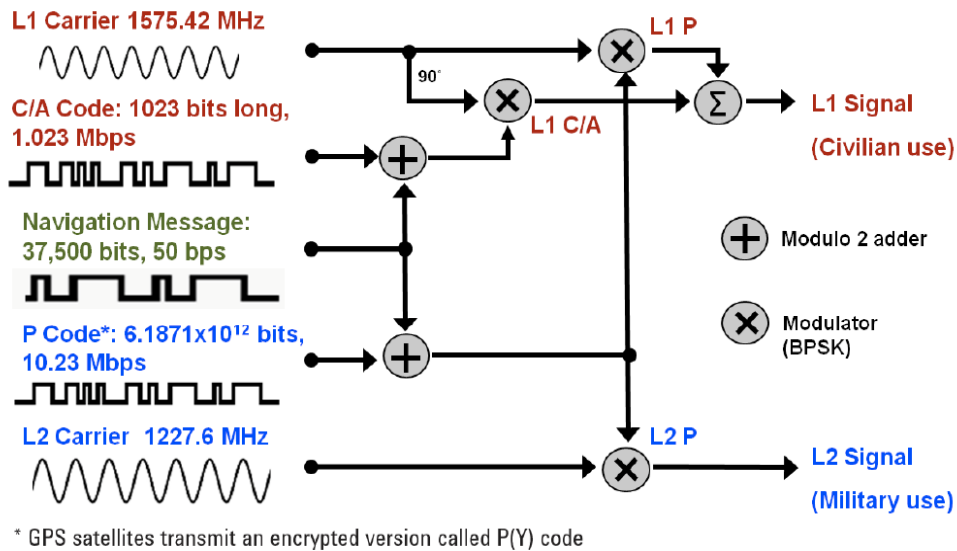


Figura 6 - Sinais transmitidos pelos satélites [9]

Um exemplo de modulação BPSK é apresentado na Figura 7, onde o sinal código pseudoaleatório C/A 'C' é somado com a mensagem de navegação 'D', de seguida o sinal é modulado em BPSK resultando no sinal final em que aquando da alteração do estado do bit o sinal altera a fase 180°.

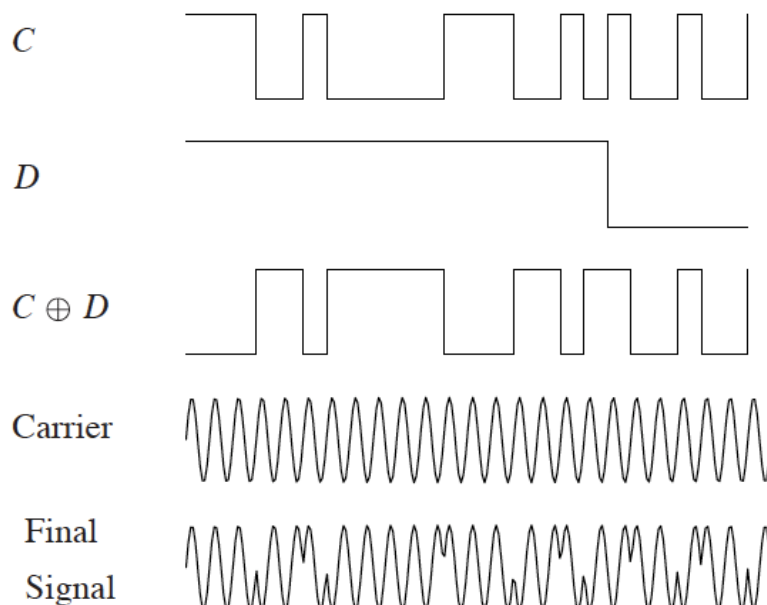


Figura 7 - Exemplo de modulação BPSK [10]

### 2.1.6. Geração do código pseudoaleatório

São gerados dois códigos pseudoaleatórios, o C/A – *Coarse/Acquisition* e o P – *Precision* (também chamado código Y quando está ativo o sistema *anti-spoofing*). A geração destes é efetuada através de um LFSR – *Linear Feedback Shift Registers*. A geração do código C/A para cada satélite é ilustrada na Figura 8.

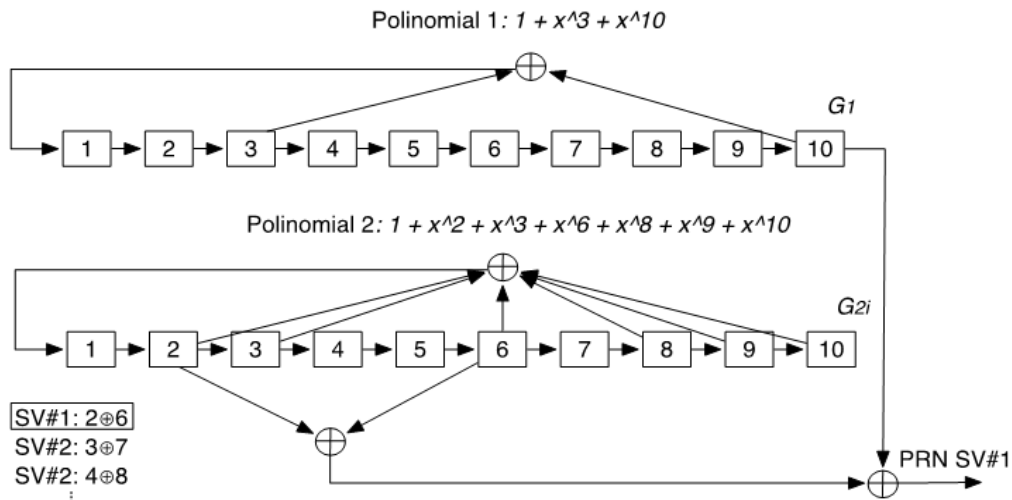


Figura 8 - Geração do código pseudoaleatório para código C/A

### 2.1.7. Formato das mensagens de Navegação

As mensagens de navegação contêm informação sobre o estado dos satélites, correções para os relógios, modelos da ionosfera, posição e trajetória dos satélites *Ephemeris*, *Almanac*, entre outros dados.

A informação contida no *Almanac* e na *Ephemeris* são semelhantes, informam o recetor sobre a posição dos satélites e suas trajetórias. O *Almanac* é uma informação menos precisa, tendo validade de vários meses e é transmitida por todos os satélites informando a posição e trajetórias de todos. Por sua vez, a *Ephemeris* é uma informação mais precisa e necessária para aplicações de elevada precisão, tem a validade de cerca de trinta minutos e cada satélite transmite apenas informação sobre a sua posição e trajetória.

A estrutura da mensagem enviada pelos satélites do sistema GPS é apresentada na Figura 9.

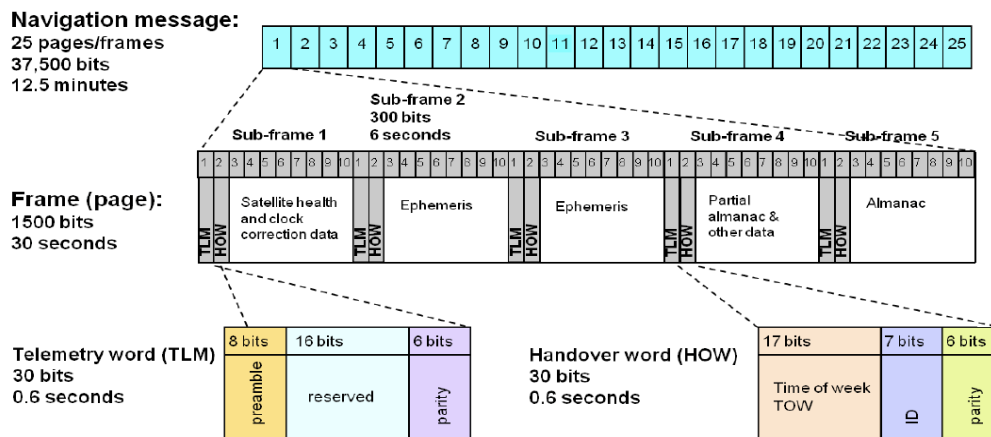


Figura 9 - Estrutura da mensagem de navegação GPS [9]

Cada mensagem de navegação completa contém 37000 bits e demora exatamente 12.5 minutos a ser completamente transmitida. Esta pode ser subdividida em 25 páginas de 1500 bits enviadas a cada 30 segundos, estas por sua vez podem ser divididas em 5 subpáginas.

Cada subpágina contém informação específica sobre um determinado parâmetro como o estado dos satélites, *Ephemeris*, *Almanac* e outros. Para além disso, cada uma das subpáginas incluem informação sobre a telemetria e tempo da semana o que permite obter uma nova posição a cada 0,6 segundos.

A informação exposta é relativa à criação de mensagens para o sistema GPS, as restantes constelações têm sistemas de mensagens e modulação idênticas, para mais detalhe sobre os mesmos consultar[9].

## 2.2. Recetores GNSS

Os recetores pertencem ao segmento do utilizador e são responsáveis pela receção e processamento dos sinais GNSS enviados pelos satélites, possibilitando assim calcular a posição em que se encontra.

A receção do sinal tem duas componentes principais, demodular os dados da mensagem de navegação e calcular a distância até ao satélite.

### 2.2.1. Cálculo da distância recetor - satélite

O cálculo da distância do recetor ao satélite é efetuado através do algoritmo CLL - *Code Lock Loop*. A implementação deste algoritmo inicia-se pela geração sincronizada do código pseudoaleatório



(PRN) pelo satélite e pelo recetor. De seguida, o recetor aplica o algoritmo CLL onde calcula o tempo da viagem do sinal desde o satélite.

Este algoritmo atrasa o sinal (PRN) gerado pelo recetor até ficar sincronizado com o sinal gerado pelo satélite, o atraso necessário para que os sinais coincidam corresponde ao tempo de viagem do sinal. Com este tempo, utiliza-se a equação (1), e é possível obter a distância do recetor ao satélite. O conceito pode ser mais facilmente entendido observando a Figura 10.

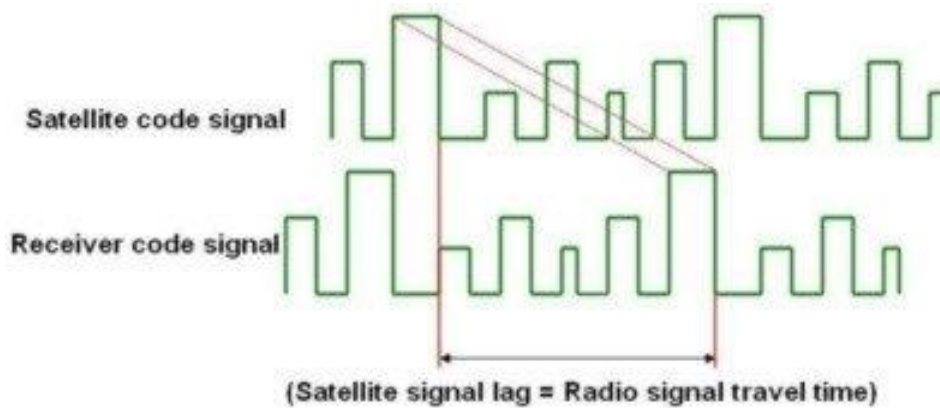


Figura 10 - Cálculo da distância através do atraso do sinal [11]

$$\text{Distância} = \text{Velocidade da Luz} * \text{Tempo} \quad (1)$$

O recetor atrasa o seu sinal até que este coincida com o sinal recebido do satélite e recorre à correlação para encontrar o exato momento em que os sinais coincidem, momento em que a correlação atinge o seu pico, como se observa no exemplo da Figura 11.

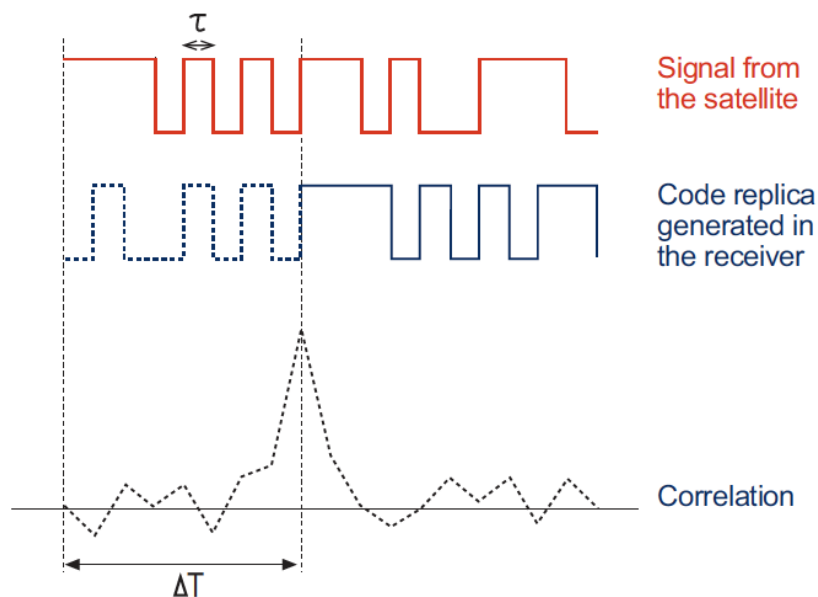


Figura 11 - Resultado da correlação entre o sinal do satélite e o sinal do recetor [8]

### 2.2.2. Desmodulação da mensagem de navegação

A informação GNSS está modulada em BPSK, o recetor para extrair a informação do sinal tem de desmodular a mesma. Esta desmodulação é efetuada através da deteção da mudança de fase no sinal. A mudança de fase indica a inversão do sinal (0 ou 1), por isso é necessária a utilização de um preâmbulo para sincronizar a receção da mensagem.

Na Figura 12, pode-se observar o exemplo de uma mensagem modulada em BPSK, de seguida, na Figura 13 observa-se a informação já demodulada.

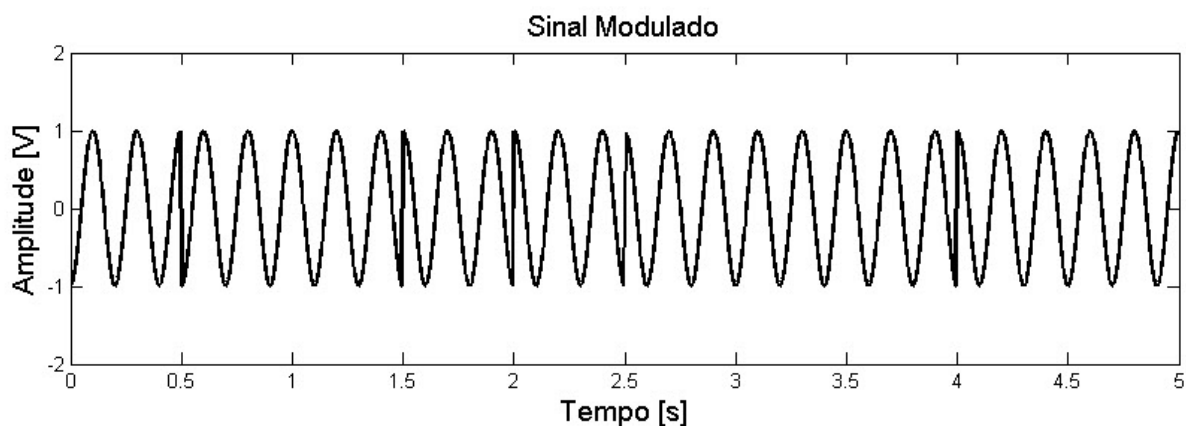


Figura 12 - Sinal contendo informação modulada em BPSK

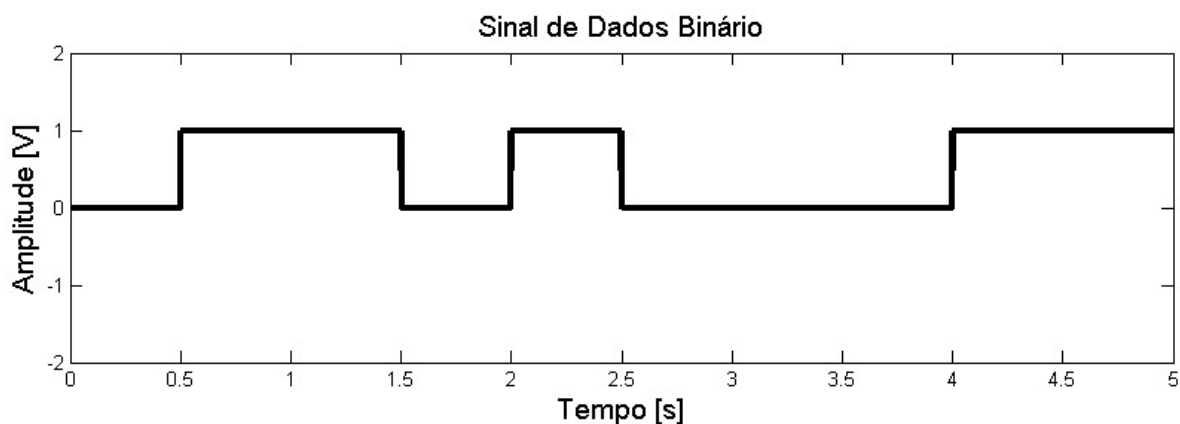


Figura 13 - Informação extraída do sinal modulado da Figura 12

### 2.2.3. Visão geral de um recetor GNSS

Como pode ser observado na Figura 14, o recetor é composto por diferentes blocos: a antena ativa que é necessária devido à baixa potência do sinal GNSS; o RF *front-end* que faz a ligação entre o sinal recebido da antena e o restante circuito. É neste bloco que é efetuada a desmodulação do sinal para que só chegue ao circuito de processamento a informação relativa à mensagem de navegação e ao código pseudoaleatório, estes com uma frequência consideravelmente inferior, considerando que a

frequência portadora é de 1575.42 MHz, para a banda L1, e os sinais que chegam ao circuito de processamento são de 1.023 MHz (consultar 2.1.5); o circuito que processa os sinais provenientes do RF *front-end* obtendo a distância aos satélites, através do método descrito no subcapítulo Cálculo da distância recetor - satélite, e da descriptação dos dados recebidos na mensagem de navegação; o elemento responsável pelo cálculo da posição atual do recetor assim como pela análise dos dados recebidos através da mensagem de navegação, que podem conter informações relativas a desvios de relógio, correções atmosféricas e outras informações relevantes para o cálculo da posição.

Os recetores calculam a posição em que se encontram e de seguida transmitem a informação para os sistemas que a vão utilizar através do protocolo NMEA.

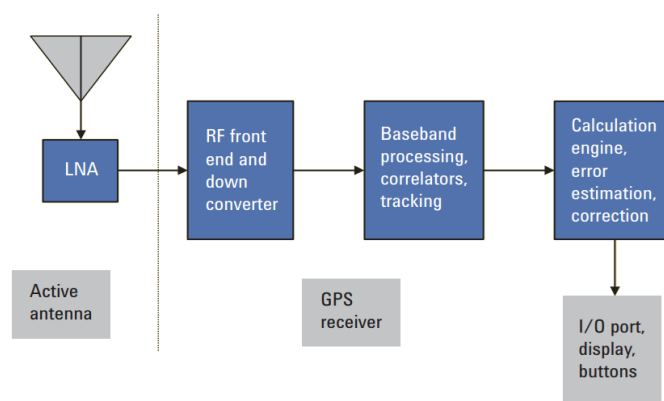


Figura 14 - Diagrama de blocos de um recetor GNSS [12]

### 2.3. Condução Autónoma

Nos últimos anos os sistemas de assistência ao condutor revelaram-se de extrema importância, uma vez que conseguem evitar milhares de acidentes rodoviários. Por isso, os desenvolvimentos nesta área têm cada vez mais impacto na indústria automóvel.

Os sistemas de assistência ao condutor iniciaram como extras para veículos de marcas e modelos topo de gama, todavia, devido à sua eficácia e aumento de segurança rodoviária eles tornaram-se uma realidade em praticamente todos os veículos, como exemplo: o sistema ABS - *Antilock Breaking System* que evita que as rodas derrapem quando ocorre uma travagem brusca reduzindo o tempo de paragem; o sistema ESP - *Electronic Stability Program* que ajuda o veículo em situações de subviragem, sob viragem, acelerações e reduções bruscas bloqueando as rodas para compensar a trajetória; o sistema Airbag que previne que os ocupantes de um veículo se aleijem com gravidade e em caso de acidente, este enche um saco com gás para amortecer o impacto dos ocupantes.

Mais tarde, com o desenvolvimento de novas tecnologias e aperfeiçoamento das existentes foram surgindo sistemas de assistência ao condutor mais complexos e completos, como: sistema de proteção de pedestres em que o veículo trava automaticamente quando deteta um pedestre em frente ao carro; sistema que trava automaticamente quando deteta que o carro imediatamente à frente está a travar ou mesmo parado e o condutor não se tenha apercebido; sistema que ajuda o condutor a manter o veículo dentro das linhas do pavimento utilizando um sinal sonoro quando este se distrai; sistema que deteta o ângulo morto do veículo e outros sistemas.

Posteriormente, com o sucesso destes sistemas de assistência ao condutor, iniciou-se o desenvolvimento de sistemas parcialmente autónomos que já assumem algumas funções do veículo como a aceleração e a direção, mas, sempre com monitorização do condutor. Como exemplos: o sistema de estacionamento automático; o sistema de assistência em filas de trânsito que conduz o veículo quando este se encontra numa fila de trânsito a baixa velocidade; o sistema que iguala a velocidade do veículo com o veículo imediatamente à frente.

Dando continuidade ao desenvolvimento de sistemas cada vez mais autónomos surgiram sistemas com um elevado grau de autonomia em que o veículo assume todas as funções e não é necessária a monitorização do condutor, caso o veículo verifique alguma condição em que não esteja garantida a segurança dos ocupantes, este avisa o condutor com tempo suficiente para este assumir o controlo do veículo. Como exemplo de sistemas deste género: condução em filas de trânsito condução em autoestradas, que ainda não estão completamente desenvolvidos [13].

O objetivo do desenvolvimento da condução autónoma são sistemas em que não é necessário qualquer tipo de condução por parte do condutor, sistemas totalmente autónomos, sendo que na data de redação desta dissertação ainda não existe nenhum sistema com estas características.

Na Figura 15 é possível observar a evolução esperada dos sistemas de condução autónoma ao longo dos anos, assim como os sensores necessários para o alcançar.

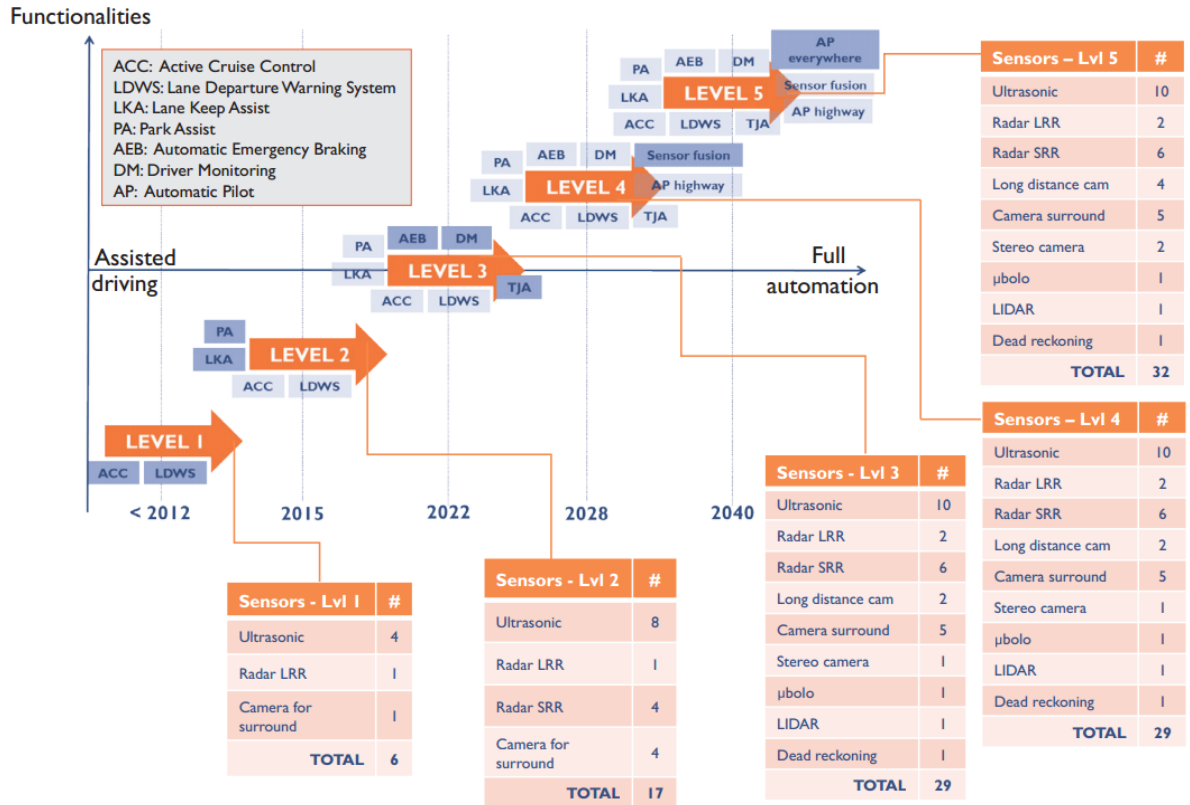


Figura 15 - Evolução dos sistemas de condução autónoma e sensores utilizados [3]

## 2.4. Teste e Verificação de Recetores GNSS

Os sistemas de teste de recetores GNSS utilizam um gerador ou simulador de sinais para criar um sinal GNSS válido. Conseguindo obter um sinal válido, é injetado no recetor e é verificada a resposta do mesmo, este é o padrão de testes que normalmente se utiliza, como ilustrado na Figura 16

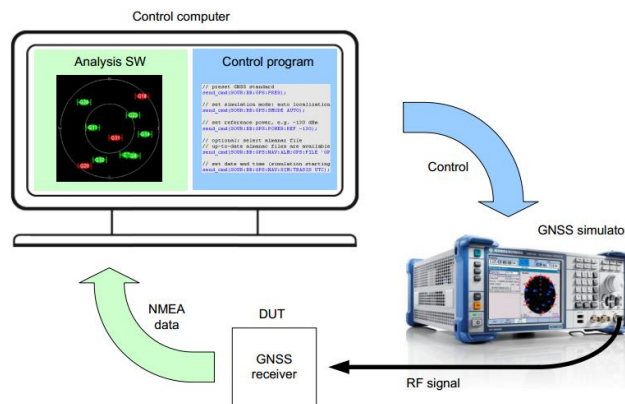


Figura 16 - Visão Geral do sistema de Testes R&S [14]

Uma vez recebida a posição no sistema de análise, esta é comparada com a posição gerada e são efetuadas várias medições com o intuito de caracterizar o recetor segundo um conjunto de métricas. A Tabela 1 apresenta uma visão geral dos parâmetros e uma breve descrição sobre cada um.

Tabela 1 - Definição dos parâmetros de caracterização de um recetor GNSS

Definição dos termos técnicos	
Termos técnicos	Definição
Tempo para a primeira aquisição	O tempo que o recetor necessita para adquirir a primeira posição a partir dos sinais recebidos dos satélites. É distinguido entre arranque a frio, morno e quente.
Tempo de reaquisição	O tempo total que o recetor demora a determinar uma posição depois de uma perda total de sinal, como ao passar num túnel.
Sensibilidade de aquisição	A potência mínima do sinal necessária para que o recetor o interprete como sinal GNSS e conseqüentemente alcance uma posição válida.
Sensibilidade em funcionamento	A potência mínima do sinal necessária para que o recetor continue a descodificar os sinais recebidos pelos satélites e conseqüentemente manter a posição válida. Normalmente esta potência é inferior á sensibilidade de aquisição.
Precisão de localização	A precisão espacial com que o recetor consegue determinar a sua posição em relação é posição real. É distinguida entre absoluta, repetitiva e dinâmica. Absoluta: a precisão da posição relativa a uma posição fixa conhecida. Repetitiva: a variação entre as posições obtidas na mesma posição. Dinâmica: a precisão da posição com o recetor em movimento.
Reflexões	Os sinais de satélite reais são refletidos em edificios e objetos de grande dimensão, assim o recetor não recebe apenas o sinal diretamente do satélite, mas também reflexões do mesmo com atrasos, atenuações e deslocamentos <i>doppler</i> . É avaliada a imunidade do recetor a estes sinais.

## 2.5. Sistema de coordenadas

De uma perspetiva genérica, os sinais transmitidos pelos satélites permitem ao recetor GNSS calcular a distância a cada um dos mesmos e através de um algoritmo de triangulação conseguem obter a sua posição. O formato desta posição é representado por coordenadas geográficas com três componentes, latitude, longitude e altitude.

Como se pode observar na Figura 17, a latitude é medida em graus e é o ângulo correspondente entre o equador e os polos, sendo o equador 0° e os polos 90° Norte ou Sul e a longitude é o angulo entre um ponto e o Meridiano de Greenwich, sendo os seus valores entre 0° e 180° Este ou Oeste.

A altitude é o a distância em metros desde um ponto até ao nível médio das águas do mar.

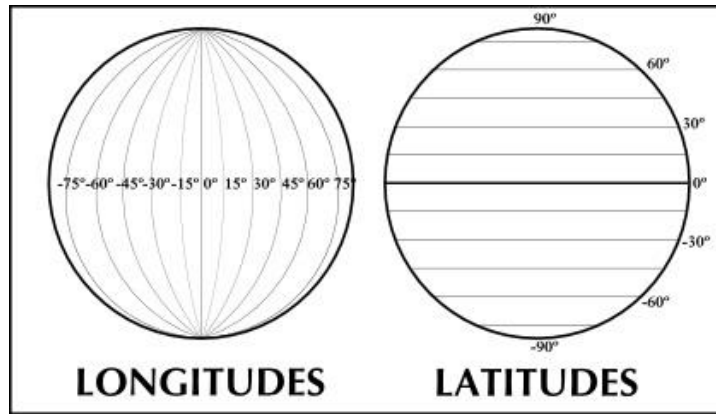


Figura 17 - Representação Latitude e Longitude [15]

Como estes ângulos não seriam suficientes para determinar posições com precisão, cada ângulo é dividido em 60 minutos e por sua vez cada minuto é dividido em 60 segundos.

Assim uma coordenada pode ser apresentada em vários formatos, sendo o mais utilizado o formato  $gg^{\circ}mm'ss''$ , em que 'gg' são graus, 'mm' minutos e 'ss' os segundos. De modo a facilitar as comunicações existem algumas compactações deste sistema, onde é enviado o grau, os minutos e os segundos num só valor decimal, ficando  $gg^{\circ}dd$  em que 'gg' são graus e 'dd' o valor decimal.





### 3. SISTEMA DE TESTE PARA O APP (*AUTOMOTIVE PRECISE POSITIONING*)

#### 3.1. Visão Geral do Sistema

Na Figura 18 é apresentada a visão geral do sistema de teste, verificação e aquisição de sinais GNSS.



Figura 18 - Visão geral do sistema de teste GNSS

O processo de teste e aquisição inicia-se no computador onde está centrado o algoritmo de testes, aqui é enviada a informação para o gerador sobre o sinal a gerar e para o receptor sobre que constelações está apto para adquirir. O sinal gerado é injetado no receptor, e este por sua vez decodifica a mensagem enviada e calcula a posição em que se encontra através do código pseudoaleatório. A informação NMEA que o receptor consegue decifrar é enviada, através de SPI, para o microcontrolador. Posteriormente a informação é reenviada através de CAN-Bus para o computador.

No computador é efetuado o algoritmo de teste onde se compara o sinal gerado com o sinal recebido e se caracteriza o receptor segundo um conjunto de parâmetros.

### 3.2. Requisitos

Os requisitos necessários para que o sistema seja considerado funcional foram decididos em conjunto com a Bosch e são os seguintes:

- Adquirir informação transmitida pelo recetor GNSS;
- Configurar recetor para adquirir diferentes bandas e constelações;
- Configurar gerador de sinais GNSS em diferentes bandas e constelações;
- Testar recetor de sinais GNSS segundo um conjunto de parâmetros;
- Transmitir informações de navegação (ex. posição) do recetor para o computador.

### 3.3. Restrições

Todos os requisitos enumerados em 3.2 têm de ser cumpridos tendo em conta um conjunto de restrições definidos no planeamento do sistema. Estes são na sua maioria impostos pelo facto do projeto ter a finalidade de testar sensores passíveis de serem incorporados em sistemas ligados à indústria automóvel. As restrições da dissertação são apresentadas de seguida:

- Implementação do sistema no microcontrolador definido pela Bosch;
- Inexistência de recetores de diferentes bandas em simultâneo;
- Software de compilação e depuração definido pela Bosch (Multi IDE);
- Plataforma de desenvolvimento definida pela Bosch;
- Fiabilidade e consistência.

### 3.4. Hardware

Os componentes de *hardware* que se utilizaram foram na sua maioria restrições impostas pelo projeto APP, no qual esta dissertação se insere, uma vez que o mesmo hardware é utilizado em diferentes elementos do mesmo. Como a sua finalidade é a indústria automóvel, a escolha do hardware teve de ser extremamente cuidadosa, devido às elevadas exigências de segurança, robustez e fiabilidade que esta indústria impõe.

Na Figura 19 pode-se observar a montagem do hardware utilizado para o desenvolvimento do sistema, e os seus componentes são explicados em mais detalhe nas secções seguintes.

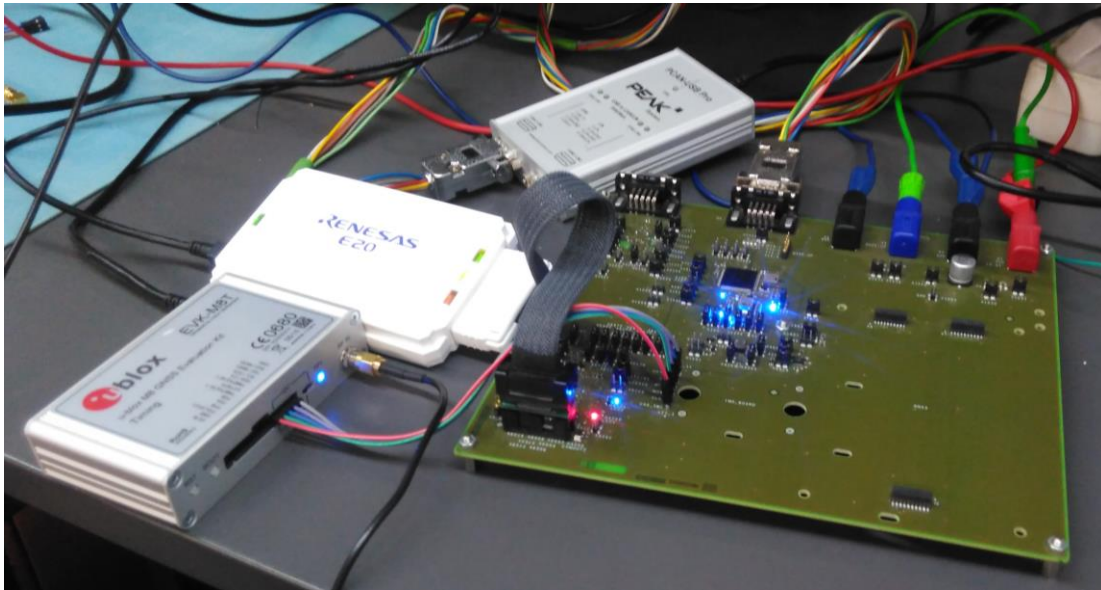


Figura 19 - Montagem do hardware utilizado

#### 3.4.1. Microcontrolador

O microcontrolador a utilizar no projeto tem de obrigatoriamente ser desenhado tendo em conta as exigências da indústria automóvel. Assim, foi usado o RH850 da empresa Renesas.

Este microcontrolador, possui dois núcleos principais, cada um com um microcontrolador de segurança a realizar mecanismos de redundância e ainda um microcontrolador a garantir a segurança do microcontrolador. Conta ainda com mecanismos de proteção de corrupção da memória, picos de tensão, entre outros mecanismos que visam a segurança e robustez de funcionamento do microcontrolador.

O RH850 possui uma arquitetura de 32 bits e o seu conjunto de instruções é RISC – *Reduced Instruction Set Computer*. Este conta com um vasto conjunto de periféricos entre os quais os que se utilizaram nesta dissertação: CAN e SPI.

#### 3.4.2. Recetor GNSS

O recetor de sinais GNSS que se utilizou no desenvolvimento do projeto é o modelo NEO-M8T da empresa uBlox. Este foi utilizado apenas para validação do sistema, pois o recetor final ainda se encontra em desenvolvimento. Assim, utilizou-se o *kit* de desenvolvimento EVKIT- M8T.



Figura 20 - Kit de desenvolvimento uBlox M8T [16]

Este recetor é capaz de adquirir diferentes constelações simultaneamente (GPS, GALILEO, GLONASS, BeiDou) e ainda os serviços de correção (SBAS). Todavia, não possui capacidade para aquisição de diferentes bandas, adquirindo apenas as bandas centradas em 1575.42 MHz. A frequência de atualização com diferentes constelações é de 4 Hz.

O recetor M8T pertence ao segmento profissional da uBlox, por isso, não poderá ser utilizado no projeto final APP uma vez que seria necessária a certificação para a indústria automóvel. Contudo, como a interface será a mesma (NMEA), foi possível a validação do sistema com este recetor.

Os detalhes relativos ao funcionamento do recetor são apresentados na Tabela 2.

Tabela 2 - Características de funcionamento do recetor GNSS NEO-M8T [17]

		Características de funcionamento do recetor GNSS NEO-M8T					
		GPS GLONASS	GPS BeiDou	GPS	GLONASS	BeiDou	GALILEO
Tempo para primeira aquisição (s)	Frio	25	28	29	30	34	45
	Morno	2	2	2	2	3	7
	Quente	1	1	1	1	1	1
Precisão de posição (m)	Autónomo	2.5	2.5	2.5	4.0	3.0	TBC
	SBAS	2.0	2.0	2.0	N/A	N/A	N/A
Precisão de velocidade (m/s)		0.05	0.05	0.05	0.1	0.1	0.1
Frequência de atualização (Hz)		4	4	10	10	10	10
Precisão da direção (graus)		0.3	0.3	0.3	0.4	0.5	0.5

Na Tabela 3 pode-se observar a informação relativa às sensibilidades mínimas requeridas pelo recetor para a aquisição de uma posição válida nas diferentes situações possíveis.

Tabela 3 - Sensibilidades do recetor GNSS NEO-M8T [17]

Sensibilidade do recetor GNSS NEO-M8T							
Sensibilidade (dBm)	Navegação	-167	-166	-166	-166	-159	-159
	Reaquisição	-160	-160	-160	-156	-156	-153
	Arranque frio	-148	-148	-148	-145	-143	-138
	Arranque morno	-157	-157	-157	-151	-146	-142
	Arranque quente	-160	-160	-160	-156	-155	-151

Na Tabela 4 pode-se observar as características técnicas do recetor, assim como os limites de operação.

Tabela 4 - Características técnicas do recetor GNSS NEO-M8T [17]

Características técnicas do recetor GNSS NEO-M8T		
Tensão de alimentação	2.7 V ~ 3.6 V	
Interfaces	UART, USB, SPI, DDC(I2C)	
Sinais GNSS em simultâneo	3	
Limites Operação	Dinâmico (g)	≤ 4
	Altitude (m)	50000
	Velocidade(m/s)	500

### 3.4.3. Gerador GNSS

Na Figura 21 pode observar-se o gerador de sinais GNSS LabSat3 utilizado para os testes. Este é capaz de gravar, reproduzir sinais GNSS em 3 constelações e gerar trajetos definidos pelo utilizador.



Figura 21 - Gerador GNSS LabSat3

### 3.4.4. Interface CAN-Bus para USB

O adaptador utilizado foi PEAK PCAN-USB Pro (Figura 22) pois cumpre os requisitos necessários para a comunicação. A sua função é fazer a interface entre a rede CAN e o computador, através de USB 2.0. Este permite comunicações entre 5 Kbit/s até 1 Mbit/s e é compatível com CAN *standard* e CAN *extended*.

A ligação entre o adaptador e o microcontrolador utiliza as fichas DB9, cujo diagrama de ligação pode ser observado na Figura 23.



Figura 22 - PEAK PCAN Pro

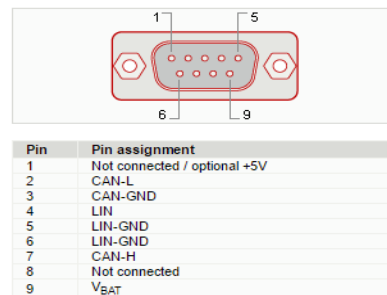


Figura 23 - Diagrama de conexões DB9 PEAK PCAN Pro

### 3.4.5. Transceiver CAN-Bus

Para a função de traduzir os sinais TTL do CAN para a rede em diferencial utilizou-se um ASIC da Bosch que permite esta função com taxa de transferência máxima de 5 MHz. Contudo, optou-se pela utilização de 1 MHz de taxa de transferência, devido a ser a máxima taxa permitida no CAN-Standard

Na Figura 24 pode-se observar um exemplo de uma mensagem antes e depois de ser traduzida.

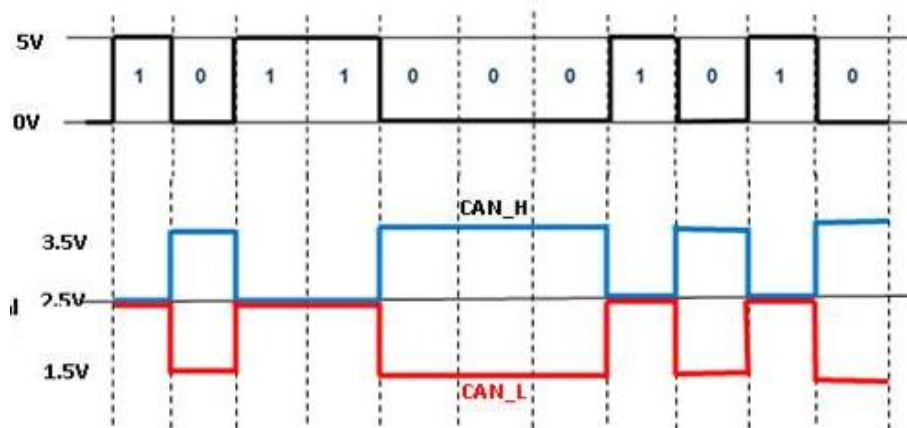


Figura 24 - Representação de uma mensagem CAN TTL e diferencial

### 3.5. Protocolos de comunicação

#### 3.5.1. SPI

O protocolo SPI – *Serial Peripheral Interface* utilizou-se na comunicação entre o microcontrolador e o recetor GNSS. No microcontrolador utilizado a velocidade de transmissão máxima é de 20 Mbps e com diferentes dispositivos.

A interface SPI utiliza quatro fios, MOSI - *Master Out Slave In*, MISO – *Master In Slave Out*, CLK – *Clock* e SS – *Slave select*, e a ligação entre o dispositivo *Master* e o *Slave* pode ser observada na Figura 25.

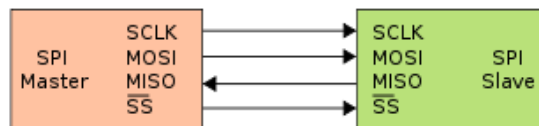


Figura 25 - Conexões master - slave SPI [18]

Este protocolo permite a ligação entre vários dispositivos, um *master* e vários *slaves*, e a conexão entre estes é apresentada na Figura 26, onde a seleção entre os dispositivos *slave* é efetuada através do SS, sinal que ativa apenas o *slave* com o qual o master pretende comunicar.

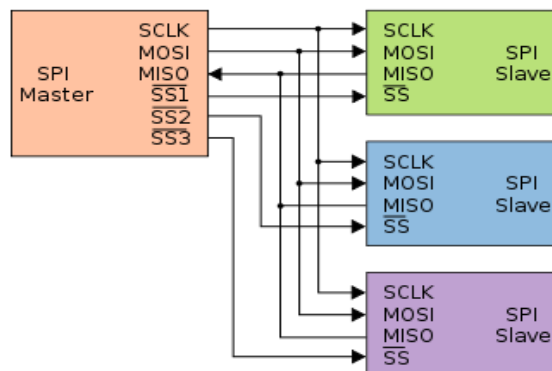


Figura 26 - Conexão entre vários dispositivos SPI [18]

Na Figura 27 observa-se o comportamento de cada uma das linhas da comunicação SPI. É possível observar que a comunicação é bidirecional, isto é, quando o *master* está a transmitir através da linha MOSI, este está simultaneamente a receber dados através da linha MISO. Os pulsos de relógio são sempre gerados pelo dispositivo *master*, e deste modo, um dispositivo *slave* não tem a capacidade para iniciar uma comunicação, pois apenas envia informação quando o *master* gera pulsos de relógio. As comunicações são sempre iniciadas pelo dispositivo *master* através da ativação da linha SS

correspondente ao dispositivo *slave* com o qual pretende comunicar. Neste momento, o dispositivo *slave* fica ativo e aquando da receção dos pulsos de relógios este transmite dados na linha MISO e recebe os dados enviados pelo *master* através da linha MOSI.

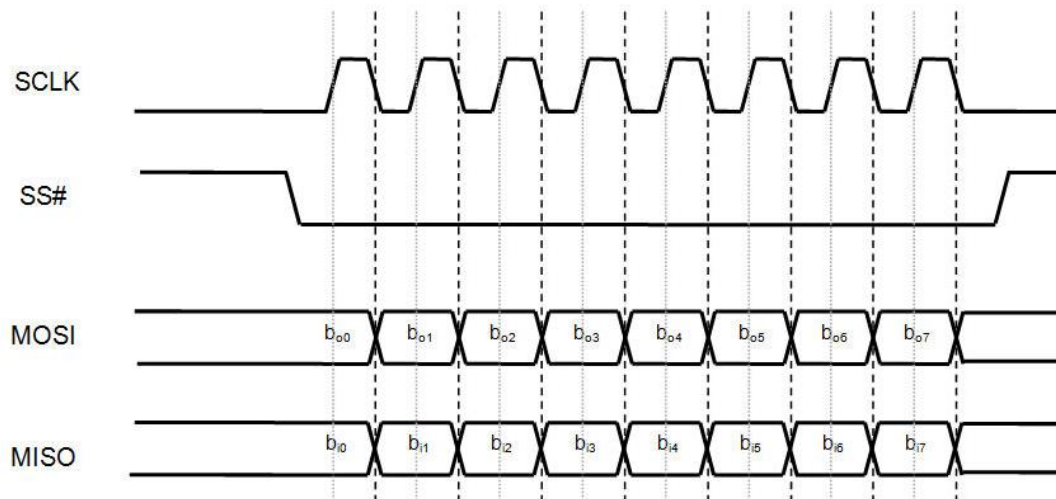


Figura 27 - Simples comunicação SPI [18]

### 3.5.2. NMEA

O protocolo NMEA - *National Marine Electronics Association* é o protocolo padrão para as comunicações entre os recetores GNSS e os dispositivos que utilizam a informação provenientes dos mesmos, neste caso com um microcontrolador. Este protocolo foi desenvolvido inicialmente para ser a interface entre os dispositivos eletrónicos marítimos. Na Tabela 5 observa-se o formato de uma trama NMEA, esta inicia sempre com o carácter '\$', termina com os caracteres <CR><LF> e os dados a ser transmitidos são separados por virgula. Excluindo os caracteres de início e terminação a trama contém ainda endereço, dados e *checksum*, e a divisão entre os dados e o *checksum* é um '\*' [19].

O endereço indica a proveniência da mensagem (GPS, GNSS, GLONASS, entre outros) e ainda o formato da mensagem, possibilitando a correta análise da mensagem recebida. Os dados são a informação que se pretende transmitir e o *checksum* é um mecanismo de controlo de erros, que faz um calculo (Exclusive OR) com os caracteres entre '\$' e '\*' e o resultado é acrescentado à mensagem, e assim o recetor da mensagem faz o mesmo cálculo e compara o valor com o recebido na mensagem, e caso sejam iguais a mensagem foi transmitida com sucesso, caso contrário significa que ocorreram erros na transmissão e a mensagem é descartada.



Tabela 5 - Formato da trama NMEA

\$	Endereço		Dados	*Checksum	<CR><LF>
Exemplo:					
\$	GP	ZDA	,141644.00,22,03,2002,00,00	*67	<CR><LF>

Os primeiros dois bytes do Endereço são o identificador da proveniência da mensagem e as possibilidades são apresentadas na Tabela 6.

Tabela 6 - Identificadores NMEA [20]

GNSS configurado	Identificador
GPS, SBAS, QZSS	GP
GLONASS	GL
GALILEO	GA
BeiDou	GB
Combinações GNSS	GN

Os últimos três *bytes* do endereço indicam ao recetor o formato dos dados, indicando a que corresponde cada valor. Contudo, o recetor tem de ter acesso às tabelas com esta informação.

Utilizando o exemplo da Tabela 5 'ZDA', a Tabela 7 é a correspondente.

Tabela 7 - Tabela de formato da mensagem 'ZDA', exemplo NMEA [20]

Nº Campo	Nome	Unidade	Formato	Exemplo	Descrição
0	xxZDA	-	String	\$GPZDA	Mensagem ZDA e ID (xx)
1	Tempo	-	Hhmmss.ss	141644.00	Tempo UTC
2	Dia	Day	dd	22	Dia UTC (1-31)
3	Mês	Mês	mm	03	Mês UTC (1-12)
4	Ano	Ano	aaaa	2002	Ano UTC
5	Ltzh	-	-xx	00	<i>Local time zone hours</i> (fixado em 00)
6	Ltzm	-	zz	00	<i>Local time zone minutes</i> (fixado em 00)
7	Cs	-	Hexadecimal	*67	Checksum

### 3.5.3. CAN

CAN-Bus foi o protocolo utilizado na comunicação entre o microcontrolador e o computador. Este foi inicialmente desenvolvido para o mundo automóvel pela BOSCH. Ao contrário de protocolos como USB ou Ethernet, o CAN não foi desenvolvido para enviar mensagens de grandes dimensões controladas por um dispositivo *master*, foi desenvolvido para enviar mensagens de pequenas dimensões (1 a 8 bytes) para toda a rede CAN simultaneamente. Sendo utilizado um mecanismo de prioridades para determinar o acesso à rede onde os nós com identificador inferior têm prioridade mais elevada e garantem o acesso à rede.

A taxa de transmissão máxima em funcionamento standard é de 1 Mbps, e a informação é transmitida para toda a rede, garantindo que existe sempre sincronismo entre todos os nós.

Na Figura 28 observa-se uma rede CAN-Bus, que é formada por dois cabos interlaçados com uma resistência de 120  $\Omega$  em cada ponta para evitar reflexões do sinal.

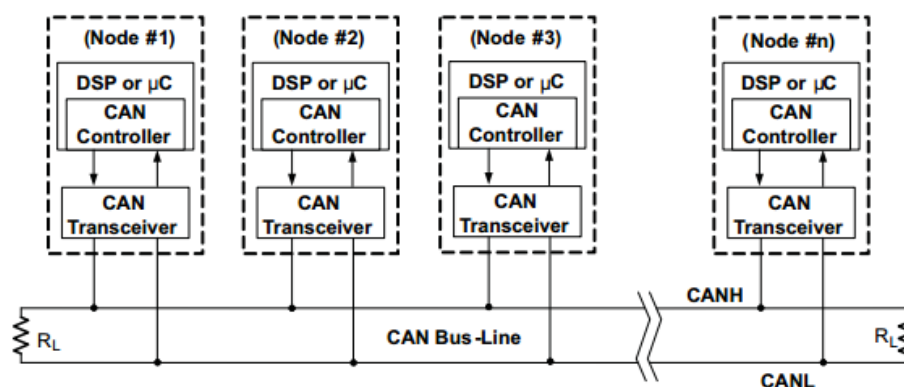


Figura 28 - Rede CAN-Bus [21]

Todos os nós necessitam de um *transceiver* CAN-Bus para transformar o sinal TTL entre 0 V e Tensão de alimentação (3.3 V ou 5 V) em diferencial, como é apresentado na Figura 29. Esta transformação resulta numa redução do ruído na rede CAN, uma vez que o sinal diferencial é bastante menos suscetível a ruídos e robusto.

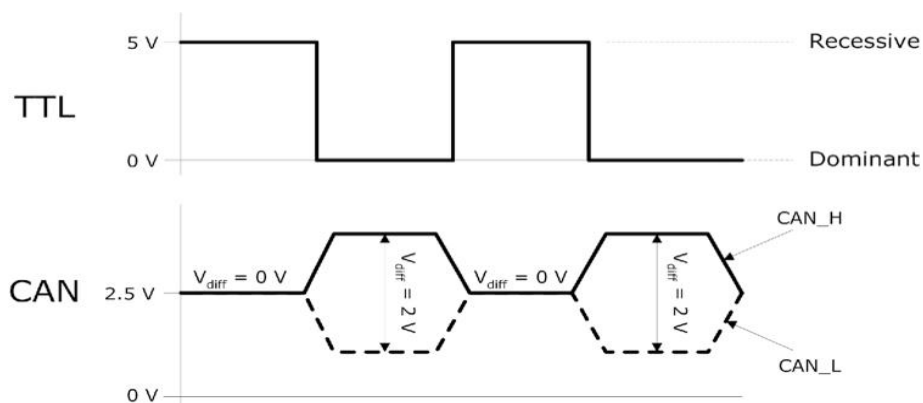


Figura 29 - Transformação do sinal TTL para diferencial [21]

O formato da trama enviada para a rede CAN, apresentado na Figura 30, é relativo ao CAN-Standard (utilizado nesta dissertação) e é composto pelo campo dos dados e alguns mecanismos de verificação de erros.

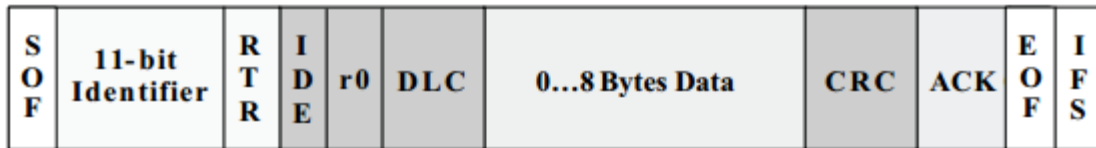


Figura 30 - Formato da trama CAN-Standard [21]

O significado dos campos da mensagem são:

- SOF – este bit é o sinal de início de mensagem, e é usado também para o sincronismo dos nós depois destes estarem num estado de espera;
- Identificador de 11 bits, estabelece a prioridade da mensagem, quanto mais baixo maior a prioridade da mensagem, e maior a possibilidade de esta ser transmitida;
- RTR – é utilizado para efetuar um pedido de informação a um determinado nó, contudo a informação é transmitida para toda a rede, o identificador é o que especifica o nó que vai responder, a informação de resposta é também transmitida para toda a rede;
- IDE – identifica que a mensagem é do formato standard e que não contem extensão para o ID;
- R0 – bit reservado;
- DLC – campo responsável por identificar quantos bytes o campo Dados contem;
- Dados – é o campo onde é transmitida a informação relevante para o recetor da mensagem, este pode ter entre 8 e 64 bits.
- CRC – contém um cálculo efetuado com os bytes a serem transmitidos, deste modo o recetor da mensagem efetua o mesmo calculo com a mensagem recebida, se o valor for igual a mensagem é valida, caso contrário a mensagem é descartada;
- ACK – cada nó que receber a mensagem corretamente força este sinal recessivo (5V/3V) com um sinal dominante (0V), indicando assim que a mensagem foi transmitida com sucesso, caso a mensagem tenha chegado ao recetor com erros, este deixa o sinal recessivo. Deste modo o emissor vai reenviar a mensagem até esta ser transmitida com sucesso.
- EOF – indica o fim da mensagem CAN;
- IFS - é um campo com 7 bits que contem o tempo necessário para um recetor mover uma mensagem recebida com sucesso para a sua memória, deste modo é garantido que todos os nós estão prontos a receber a próxima mensagem.

Na Figura 31, é apresentado o CAN-extended, contudo este não foi utilizado nesta dissertação pois o numero de nós não o exigia.

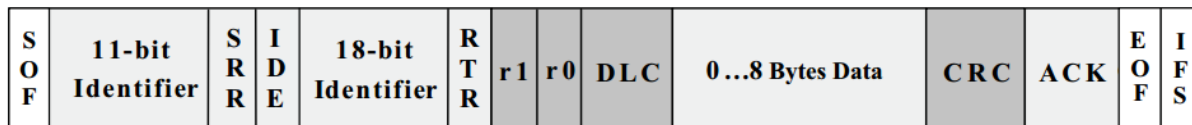


Figura 31 - Formato da trama CAN-Extended [21]

### 3.5.4. UBX

O protocolo UBX é um protocolo proprietário da U-Blox [17] (fabricante do recetor GNSS utilizado) e foi desenvolvido para a comunicação entre o recetor e o GNSS. Embora permita a transmissão de toda a informação relativa à aquisição GNSS, este é mais utilizado para configurações específicas do recetor.

Na Tabela 8 pode-se observar o formato da trama. Esta inicia-se com dois bytes de sincronização ‘μ’ e ‘b’, de seguida estão reservados dois bytes onde está a informação relativa à classe da mensagem e ao identificador da mensagem dentro dessa classe. O campo seguinte indica o tamanho do campo que contem os dados a transmitir, e os últimos dois campos de informação contêm os dois bytes do controlo de erros, o Checksum. As mensagens são divididas em 12 classes, apresentadas na Tabela 9, com cada uma destas com vários Identificadores, e é assim que o recetor da mensagem sabe como identificar os valores transmitidos no campo dados.

Tabela 8 – Formato da trama UBX

Caracter de sincronização 1 (Hex)	Caracter de sincronização 2 (Hex)	Classe (Hex)	ID (Hex)	Tamanho (Hex)	Dados (Hex)	CK_A (Hex)	CK_B (Hex)
Exemplo:							
B5	62	06	3E	C0 00	00 00 20 01 00 08 10 00 01 00 00 01	8B	90

Tabela 9 - Descrição das classes do protocolo UBX

Nome	Classe (Hex)	Descrição
NAV	01	Resultados de navegação: Posição, Velocidade, Tempo, Aceleração e outros
RXM	02	Mensagens de gestão do recetor
INF	04	Mensagens de informação: estilos de impressão, erros e outros
ACK	05	Mensagens de resposta ACK/NACK
CFG	06	Mensagens de configuração do recetor
UPD	09	Mensagens para atualizar o <i>firmware</i>
MON	0A	Mensagens de monitorização: Estados das comunicações e carga do CPU e outros
AID	0B	Mensagens para transmitir informação de assistência ao recetor
TIM	0D	Mensagens de tempo
ESF	10	Mensagens de fusão com sensores inerciais exteriores
MGA	13	Mensagens de assistência para Multi-GNSS
LOG	21	Mensagens para guardar dados

O recetor da mensagem tem de ter acesso às tabelas que indicam o que corresponde aos valores recebidos no campo dados para cada identificador, como exemplo é apresentado na Tabela 10 o significado dos valores recebidos.

Tabela 10 - Exemplificação da análise de uma mensagem UBX

Nº Campo	Nome	Exemplo (Hex)	Descrição
0	msgVer	00	Versão da mensagem (0 para esta versão)
1	numTrkChHw	00	Numero de canais disponíveis (apenas leitura)
2	numTrkChUse	20	Numero de canais a utilizar
3	numConfigBlocs	01	Numero de blocos seguintes
4	gnssID	00	Identificador GNSS (Tabela 6)
5	resTrkCh	08	Numero mínimo de canais para este identificador GNSS
6	masTrkCh	10	Numero máximo de canais para este identificador GNSS
7	Reserved	00	Reservado
8	Flags	01 00 00 01	Flags de configuração

### 3.6. Métricas de testes do recetor GNSS

Os testes aos recetores GNSS tem como função a caracterização do recetor atendendo a um conjunto de parâmetros, sendo os mais comuns o TTFF – *Time to first fix* onde se verifica quanto tempo demora até ao recetor calcular a primeira posição válida e a precisão da posição calculada. No entanto, outros testes têm-se tornado populares, como a sensibilidade e a imunidade a interferências.

Uma nota geral a ter em conta para a realização destes testes é que os resultados podem variar consoante a posição dos satélites, clima e outros fatores, e devido a isso, é necessária a geração de sinais GNSS em ambientes controlados, de modo a possibilitar a repetibilidade dos resultados.

Os parâmetros sobre os quais os recetores GNSS são descritos nos subcapítulos seguintes.

#### 3.6.1. TTFF – *Time To First Fix* - Arranque de fábrica [22]

Neste teste, a memória do recetor é limpa, eliminando os dados relativos ao *Almanac* e *Ephemeris*, como se o recetor não estivesse ligado nos últimos meses (o *Almanac* tem validade de vários meses), de seguida o recetor é ligado e é contado o tempo até à primeira aquisição de posição válida.

Para a validação dos resultados este teste é efetuado algumas vezes e é calculada a média dos resultados.

#### 3.6.2. TTFF – *Time To First Fix* - Arranque normal

Este teste tem o seu procedimento semelhante ao arranque a frio, contudo, este teste contabiliza o tempo que o recetor demora a adquirir uma posição válida partindo de um ponto em que ainda possui dados relativamente recentes quanto à sua posição [tempo (+/-20s), velocidade (+/-25 m/s) e posição (+/-100kms)] e os dados *Almanac* válidos, o que permite uma aquisição mais rápida. Estes dados demoram 12.5 minutos a serem transmitidos do satélite até ao recetor. Mas podem também ser descarregados pela rede móvel (*Assisted-GPS*) ou pela internet para diminuir o tempo necessário para os testes.

Este teste é efetuado várias vezes e o valor a considerar é a média dos mesmos.

#### 3.6.3. TTFF – *Time To First Fix* - Arranque em Standby

Para a realização deste teste é necessário que o recetor tenha ainda tempo, velocidade, distancia, Almanac e Ephemeris válidos, permitindo assim uma aquisição de posição rápida.

O valor considerado é a média de algumas repetições do teste.

### 3.6.4. Precisão de localização – Relativa

Este parâmetro calcula-se verificando a posição relativa do recetor, sendo que este cálculo tem em consideração a repetibilidade da posição entre elas, desprezando a posição absoluta.

Na Figura 32 observa-se uma exemplificação da medição da precisão relativa, onde os pontos verdes são as posições adquiridas e é considerada a distância entre elas, sendo melhor quanto menor for a distância média entre as medições. O valor final considerado como resultado é a média das distâncias entre as posições adquiridas.

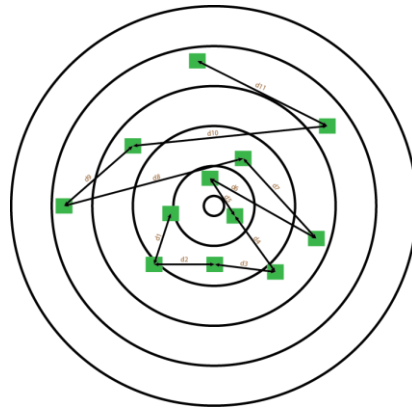


Figura 32 - Exemplificação da medição da posição relativa do recetor GNSS. É considerada a distância entre as posições adquiridas (pontos verdes)

### 3.6.5. Precisão de localização – Absoluta

Este parâmetro avalia a precisão da posição que o recetor adquire comparada com a posição real do mesmo, como exemplificado na Figura 33, onde os pontos verdes são as posições adquiridas pelo recetor e o ponto vermelho a posição real do mesmo.

É considerada média das diferenças entre as posições adquiridas e a posição real do recetor, e o recetor é melhor quanto menor for este valor.

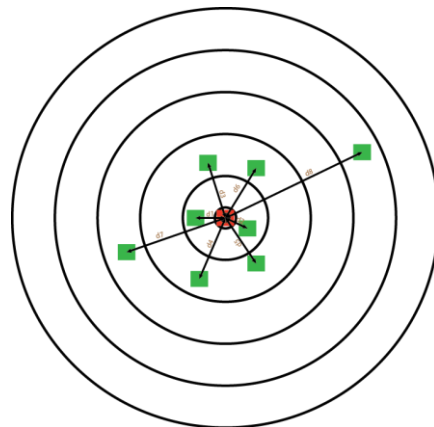


Figura 33 - Exemplificação do teste à posição absoluta, sendo os pontos verdes as posições adquiridas, o vermelho a posição real do recetor e as setas as distâncias medidas entre eles

### 3. Sistema de Teste para o APP (*Automotive Precise Positioning*)

#### 3.6.6. Precisão de localização – Dinâmica

Este teste compara a velocidade de movimento obtida pelo recetor, comparada com a velocidade real do mesmo. Como os testes serão efetuados a partir de um sinal gerado, é conhecida a velocidade de referência à qual o recetor se deve aproximar.

É considerada a média das diferenças entre a velocidade adquirida e a velocidade real, sendo o recetor melhor quanto menor for este valor.

#### 3.6.7. Tempo de Reaquisição

Este parâmetro avalia a rapidez com que o recetor readquire o sinal depois de uma perda total do mesmo (ex. passagem em túnel). É contabilizado o tempo desde que o recetor encontra novamente o sinal até ao momento em que este adquire uma posição válida.

São efetuadas várias repetições do teste e é considerada a média dos tempos, sendo o recetor tanto melhor quanto menor for este valor.

#### 3.6.8. Sensibilidade – para aquisição

A sensibilidade para aquisição é referente ao nível mínimo de potência que o sinal tem de ter para permitir uma aquisição a partir de um arranque do zero (sem informação nenhuma no recetor) dentro de um determinado tempo.

O nível de potência do sinal necessária para a aquisição é em norma inferior ao nível necessário para manter o funcionamento.

Um teste pode ser efetuado considerando o nível do sinal necessário para que o recetor adquira uma posição a partir de um arranque do zero em 100 segundos ou menos.

São efetuadas várias repetições deste teste e é considerada a média da potência do sinal necessária.

#### 3.6.9. Sensibilidade – em funcionamento

A sensibilidade em funcionamento verifica o nível do sinal necessário para que o recetor continue em funcionamento, depois de já conter toda a informação necessária para o funcionamento normal (Almanac, Ephemeris, tempo, entre outros). Normalmente este é significativamente inferior ao necessário para a primeira aquisição.



Uma das formas de realizar este teste é conseguir ter um recetor em pleno funcionamento e reduzir iterativamente o nível do sinal gerado até o recetor perder a capacidade de adquirir posições.

Realizam-se várias vezes este teste e é considerada a média do nível de sinal necessário.

### 3.6.10. Interferências

As interferências são um dos grandes problemas dos recetores de sistemas de navegação por satélite, como se pode observar na Figura 34, estas são provenientes de diferentes fontes, sinais intencionais, frequências fora de banda terrestres, reflexões em objetos de grandes dimensões e outras.

Para realizar este teste injetam-se interferências no sinal gerado e verifica-se a resposta do recetor a estas. O valor a considerar é o nível das diferentes interferências necessárias para que o recetor deixe de conseguir adquirir posições. São efetuados vários testes e é considerada a média do valor.

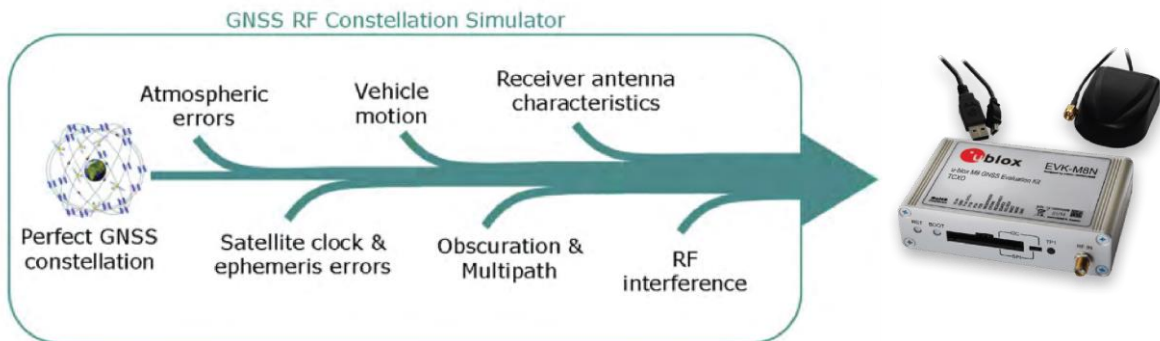


Figura 34 - Fontes de interferências do sinal GNSS (adaptado de [23])



## 4. SOFTWARE

### 4.1. Microcontrolador

O *software* do microcontrolador é essencialmente responsável pela leitura da informação proveniente do recetor, da análise, validação e transmissão da mesma para o computador, da análise das mensagens de configuração recebidas do computador e consequentemente pela configuração do recetor.

O computador por sua vez tem a responsabilidade de adquirir e analisar a informação recebida pelo microcontrolador relativa ao recetor GNSS e efetuar o algoritmo de teste da mesma. Este sistema de teste consiste, numa visão geral, na configuração do gerador GNSS para gerar uma posição (fixa ou em movimento) com diferentes combinações de constelações e bandas, de seguida aguarda pela informação do recetor relativa à posição previamente gerada e efetua as comparações e cálculos necessários para a caracterização do recetor.

#### 4.1.1. Rotina principal

Na Figura 35 é apresentado o diagrama de blocos da sequência que o software implementado no microcontrolador executa.

O *software* inicia-se configurando os periféricos necessários para o funcionamento do sistema (SPI, CAN e os relógios dos periféricos e do sistema), de seguida o sistema entra num ciclo infinito onde espera por mensagem CAN provenientes do computador, aquando da receção de uma mensagem este analisa os dois primeiros caracteres da mensagem, estes são os identificadores da mensagem e informam o microcontrolador sobre que ação efetuar com a restante mensagem.

No caso da receção de um “Iniciar Receção GNSS” (0xAA) o microcontrolador ativa a *flag* de comunicação com recetor GNSS e a restante mensagem será ignorada, caso o valor seja (0xBB) “Terminar Receção GNSS” a *flag* será desativa e a restante mensagem não será analisada, caso a receção seja “Configuração recetor GNSS” (0xCC) o microcontrolador invoca a rotina de configuração e analisa a restante mensagem para determinar a configuração a executar, como discriminado em 4.1.5.

O ciclo infinito supracitado efetua duas ações principais, a resposta a uma nova mensagem CAN e caso a *flag* de comunicação com o recetor GNSS esteja ativa este invoca a rotina de receção da informação do recetor através de SPI e posteriormente o envio da mesma por CAN para o computador, esta rotina é explicada em detalhe em 4.1.4.

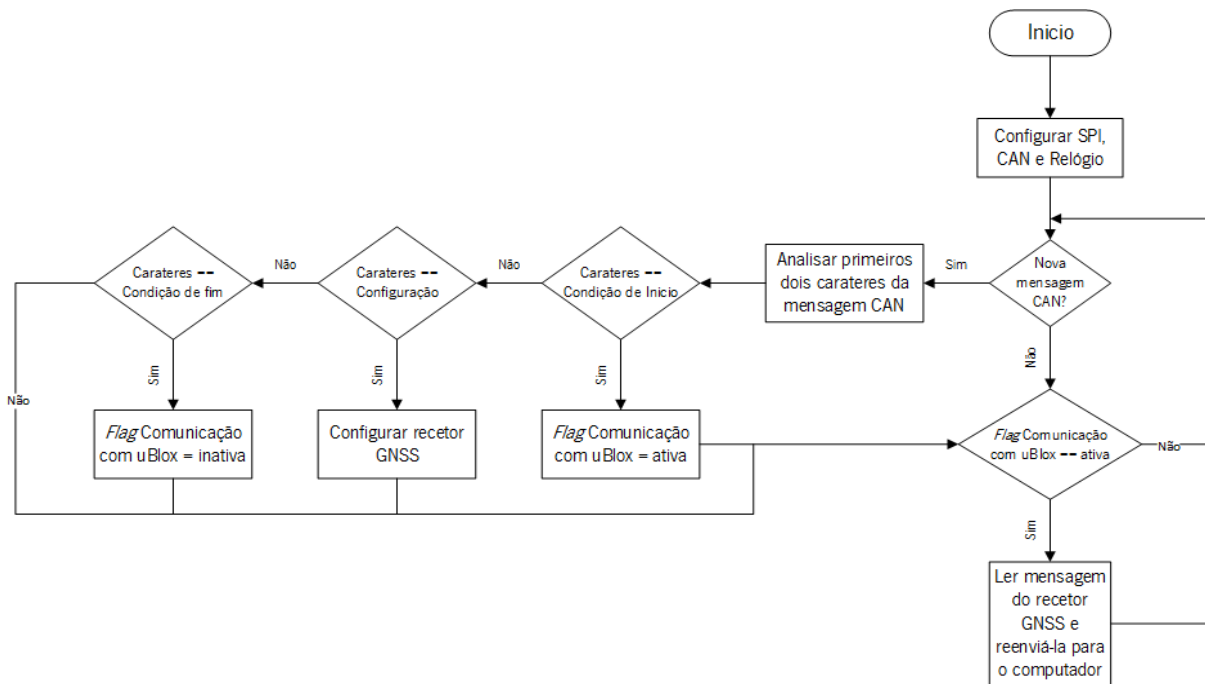


Figura 35 - Diagrama de blocos da sequência de software principal do microcontrolador

#### 4.1.2. Rotina de leitura do recetor GNSS

Na Figura 36 pode-se observar a sequência que o software de receção da informação proveniente do recetor efetua.

O diagrama corresponde à rotina “Ler mensagem do recetor GNSS e reenvia-la para o computador” da Figura 35.

Esta rotina inicia-se com a leitura de um *byte* do recetor GNSS, de seguida o microcontrolador verifica se a *flag* de início de trama já se encontra ativa (já recebido o início de trama) ou que o *byte* é diferente de 0xFF (*byte* correspondente à ausência de informação, para mais detalhe consultar capítulo 3.5.1), caso uma destas condições se verifiquem o *byte* é analisado e é efetuada a ação correspondente.

Se o *byte* recebido for “Início de trama” ('\$' no caso do protocolo NMEA) a *flag inicio* de trama é ativada, caso seja “fim de trama” (<CR><LF> no protocolo NMEA) a mesma é desativada, o contador da mensagem é reiniciado e o vetor contendo a mensagem completa é enviado para o computador por CAN através da rotina “Enviar mensagem completa através de CAN”.

Caso não seja nenhum destes o *byte* é considerado parte da mensagem e é adicionado ao vetor “DadosRecebidosSPIBuffer”.

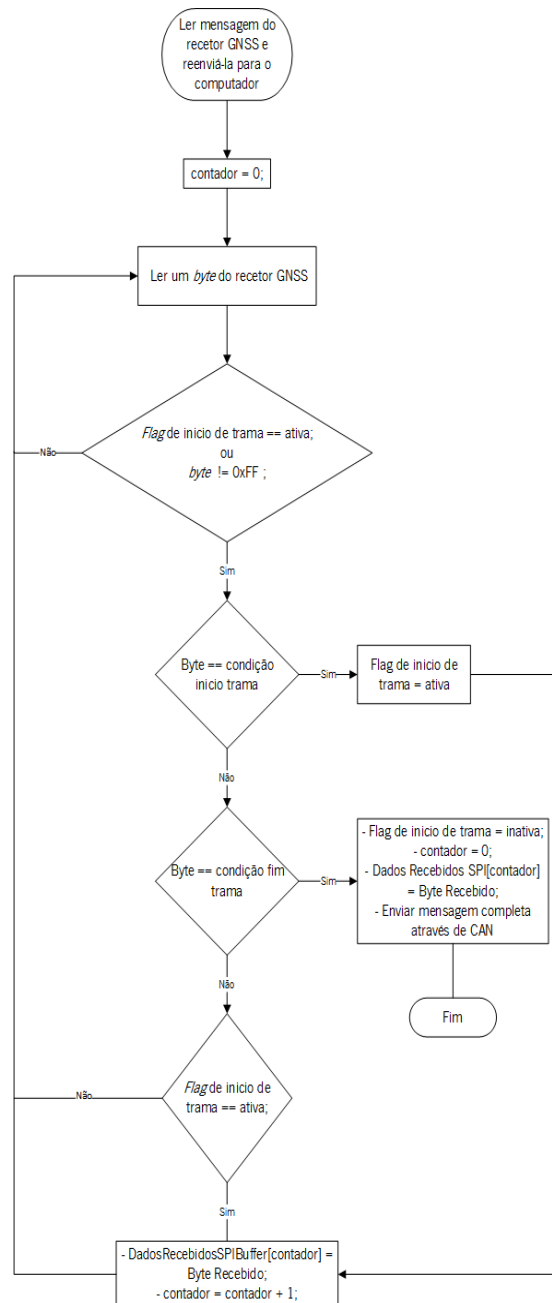


Figura 36 - Diagrama de blocos da sequência de software da recepção da informação GNSS do recetor e posterior reenvio por CAN

#### 4.1.3. Rotina de envio através de CAN

Esta rotina, apresentada na Figura 37, corresponde à rotina “Enviar mensagem completa através de CAN” do diagrama da Figura 35 e é responsável pela separação da mensagem recebida por SPI do recetor em mensagens de oito *bytes* (*payload* da mensagem CAN) e posterior envio da mesma.

A sequência inicia-se pela inicialização dos contadores e da *flag* fim de trama, de seguida entra num ciclo até o valor do contador ‘i’ ser igual ao valor inteiro imediatamente inferior à divisão do tamanho da mensagem a enviar por oito. Deste modo, o microcontrolador envia todas as mensagens enquanto

estas tiverem pelo menos oito *bytes*. Dentro de cada iteração do ciclo referido existe outro ciclo que copia os oito *bytes* para o vetor “mensagem a enviar” e quando este terminar enviar a mensagem através de CAN. De seguida, o microcontrolador verifica se ainda existem bytes que não foram enviados calculando o resto da divisão do tamanho da mensagem por oito, se o resultado for diferente de zero significa que ainda existem dados a enviar, contudo os restantes são menos que oito bytes, então o software entra num ciclo de oito repetições, insere no vetor “mensagem a enviar” os bytes (inclusive o byte “fim de trama”) preenchendo os bytes restante com o valor 0xFF e de seguida envia a mensagem através de CAN.

A divisão da mensagem NMEA em mensagens CAN pode ser observada na Tabela 11.

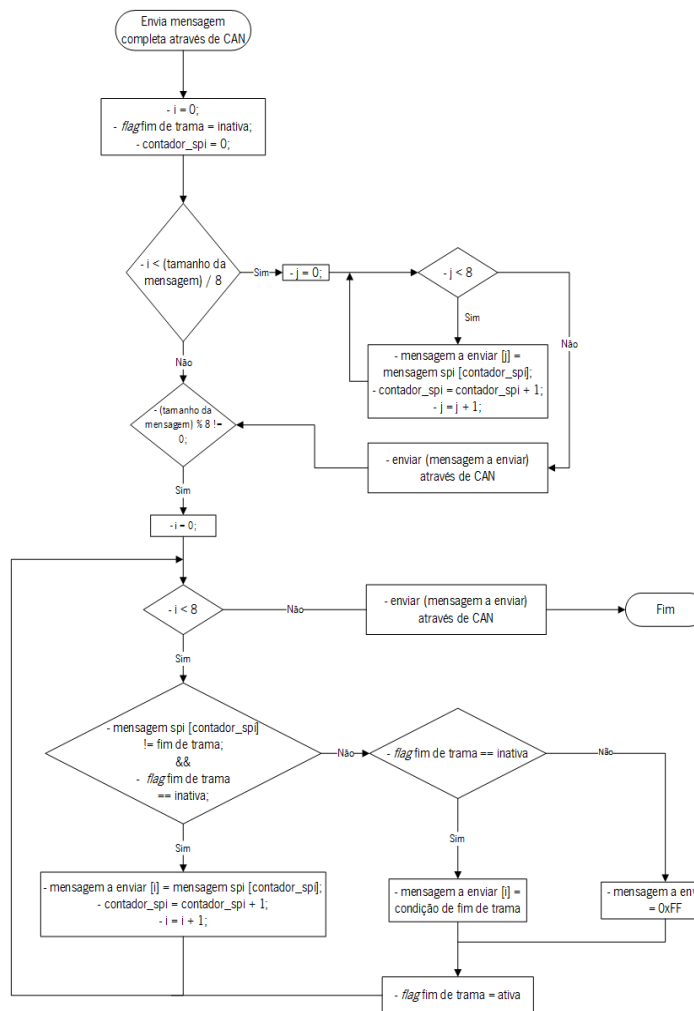


Figura 37 - Diagrama de blocos da sequência de software do envio de informação

Tabela 11 - Decomposição da mensagem NMEA em mensagens CAN

\$	G	P	Z	D	A	,	1	4	1	6	4	4	.	0	0	,	2	2	,	0	3	,	2	0	0	2	,	0	0	,	0	0	*	6	7	CR	LF	FF	FF
Mensagem CAN 1								Mensagem CAN 2								Mensagem CAN 3								Mensagem CAN 4								Mensagem CAN 5							

#### 4.1.4. Comunicação SPI (Recetor GNSS – Microcontrolador)

Através da interface SPI os dados são enviados seguindo o protocolo NMEA (consultar 3.5.2). A interface SPI é bastante versátil em termos de número de bits enviados por comunicação e velocidade, sendo a escolha dos mesmos sempre função do dispositivo *master*.

Na comunicação implementada no microcontrolador (dispositivo *master*) optou-se por 8 bits em cada comunicação, por motivos de simplicidade e redução do processamento necessário para a análise da mensagem, e uma taxa de transferência de 500 Kbit/s sendo esta a que revelou melhor relação entre erros de transmissão e velocidade.

Sendo o recetor GNSS o dispositivo *slave*, este não tem capacidade para enviar dados no momento em que os tem disponíveis, só envia quando o microcontrolador enviar alguma mensagem (gerando assim os pulsos de relógio necessários).

O software implementado no microcontrolador seguiu as indicações do fabricante do recetor GNSS, assim sendo, quando o microcontrolador pretende receber dados do recetor este envia mensagens sem informação relevante (0xFF), gerando assim os pulsos de relógio necessários para a recepção de um byte. A razão do envio da mensagem (0xFF) é que o recetor GNSS analisa todas as mensagens enviadas pelo microcontrolador e quando recebe 50 destas mensagens, o recetor desliga a análise das mensagens até à recepção de uma mensagem diferente. Deste modo reduz a análise desnecessária das mensagens provenientes do microcontrolador, ficando com mais tempo de processamento disponível para a aquisição e descriptação da informação recebida dos satélites.

#### 4.1.5. Rotina de configuração do recetor GNSS

Esta rotina, apresentada na Figura 38, é responsável pela configuração do recetor GNSS atendendo aos parâmetros recebidos, através de CAN, do computador.

Para que o microcontrolador obtenha todos os parâmetros necessários para a configuração, este tem de juntar as diferentes mensagens CAN com a informação que o utilizador escolheu na interface, uma vez que a mensagem de configuração tem sempre mais que 8 bytes, esta teve de ser dividida para poder ser transmitida por CAN-Bus (consultar 4.2.5). Um exemplo de uma mensagem de configuração enviada por SPI pode ser observada na Tabela 12.

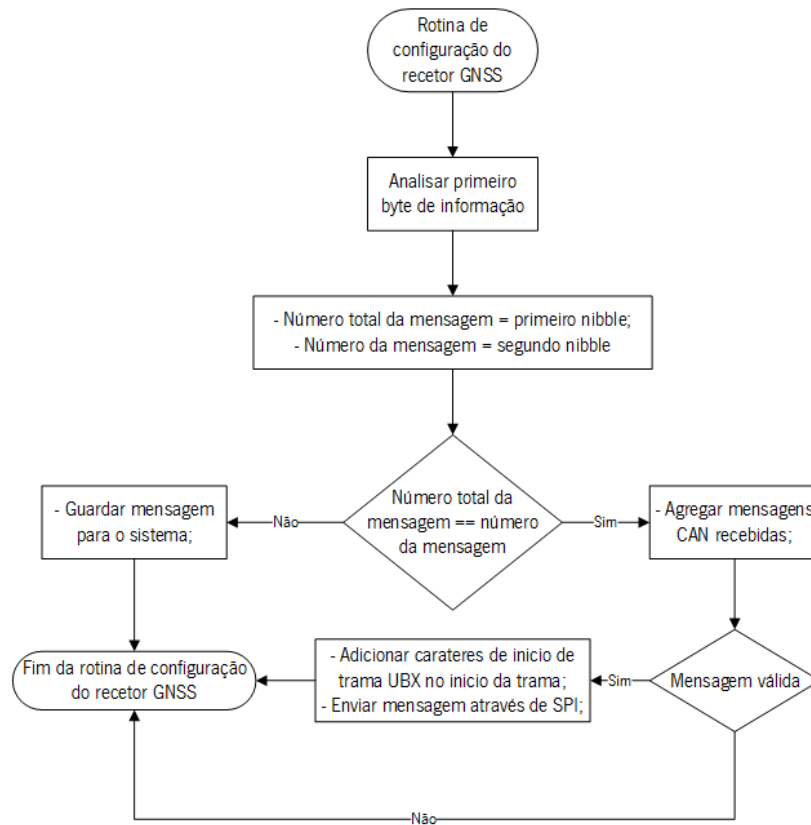


Figura 38 - Diagrama de blocos da rotina de configuração do recetor GNSS

Tabela 12 - Exemplo de uma mensagem de configuração enviada por SPI para o recetor GNSS

Byte num.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Valor (hex)	B5	62	06	3E	0C	00	00	00	20	01	00	08	10	08	10	00	01	00	00	01	8B	90

## 4.2. Computador

### 4.2.1. Interface

A interface apresentada na Figura 39 foi desenvolvida em MATLAB e permite ao utilizador a configuração manual do recetor e do gerador, possibilitando o teste ao recetor GNSS com diferentes combinações de sinais. Além disto, a interface permite iniciar e terminar a aquisição de sinais, apresentar um mapa com a localização obtida pelo recetor GNSS, exibir a última posição adquirida e efetuar os diferentes testes, expostos no subcapítulo 3.6.



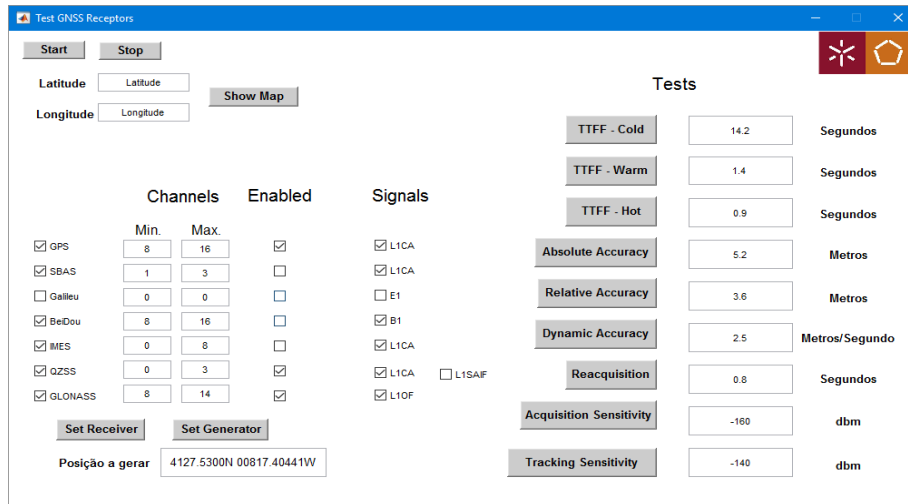


Figura 39 - Interface gráfica do sistema de testes

#### 4.2.2. Rotina Principal

Esta é a rotina efetuada no arranque do programa no computador, e inicia-se pela inicialização da comunicação CAN-BUS, configurando-a de acordo com a inicialização efetuado no microcontrolador. De seguida é inicializado o temporizador com o tempo de atualização da posição desejado, que pode ser configurado pelo utilizador na interface. Este temporizador invoca uma rotina cada vez que é acionado.

De seguida o sistema entra num estado de espera, onde todas as ações são desencadeadas pelo utilizador, através de um botão, ou pelo timer caso este já tenha sido iniciado.

As opções disponíveis para o utilizador podem ser observadas na Figura 40 e cada uma destas será apresentada em detalhe nos subcapítulos seguintes bem como os procedimentos para efetuar os cálculos das métricas de caracterização que são apresentados no capítulo 2.4.

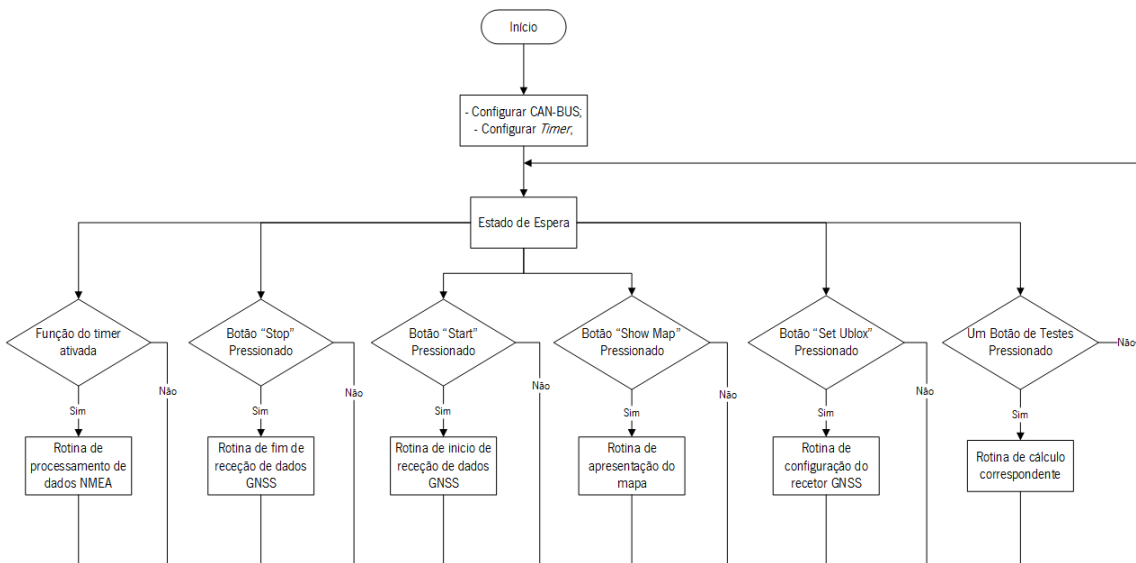


Figura 40 - Sequencia da rotina principal do software no computador

## 4.2.3. Processamento de dados NMEA

Esta é a rotina invocada pelo temporizador e é responsável pela recepção das mensagens CAN, processá-las e extrair a informação relevante das mesmas, como a posição.

Esta inicia-se pela verificação da existência de alguma mensagem CAN recebida, caso não exista a rotina é terminada. No caso de existirem mensagens, esta são todas juntas numa única mensagem (como o *payload* do CAN-Bus é 8 *Bytes* e a mensagem NMEA é sempre mais que 8, a mensagem foi dividida em parcelas de 8 *Bytes*, para mais detalhe consultar subcapítulo 'Rotina de envio através de CAN'. Posteriormente, a mensagem é dividida pelos caracteres de início de trama NMEA '\$', e pelos caracteres de terminação '<CR><LF>'. Cada uma das mensagens NMEA é analisada e são verificados os primeiros cinco caracteres, pois estes contêm o identificador e o formato da mensagem recebida, possibilitando assim ao computador entender o que indicam os restantes valores da mensagem.

As mensagens são analisadas uma a uma até encontrar a mensagem pretendida, no exemplo da Figura 41 procura-se pela mensagem 'GNGGA' que contém nos seus campos de informação a posição calculada pelo recetor GNSS a partir da análise de diferentes constelações.

Uma vez encontrada a mensagem, são extraídos os dados relativos à posição, é atualizada a interface e os dados são gravados no sistema para serem interpretados.

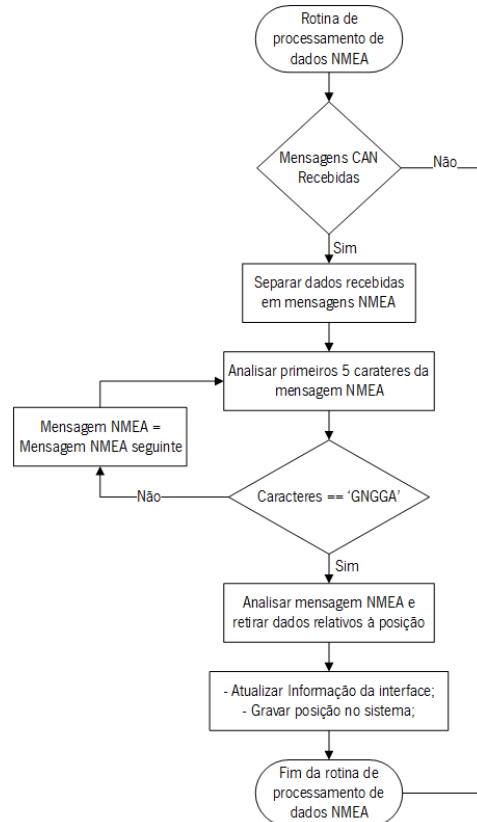


Figura 41 - Rotina de processamento das mensagens NMEA recebidas por CAN-BUS

#### 4.2.4. Rotinas de início e terminação da recepção CAN-Bus

Estas duas rotinas são responsáveis, respetivamente, pelo início e terminação da aquisição de sinais GNSS.

A rotina de início, que ocorre pela ação do utilizador, verifica se o canal CAN e o temporizador já se encontram ativos, no caso de algum deles não estar, a rotina apresentada na Figura 42 ativa-os. Sempre que ambos estão ativos, é enviado o comando 'AA' para o microcontrolador, indicando que este deve iniciar a recepção de sinais GNSS provenientes do recetor.

A rotina de terminação, ilustrada na Figura 43, por sua vez, verifica se alguns dos canais se encontra ativo e caso se verifique, desativa-o. De seguida, envia o comando 'BB' para o microcontrolador terminar a recepção de dados GNSS do Recetor.

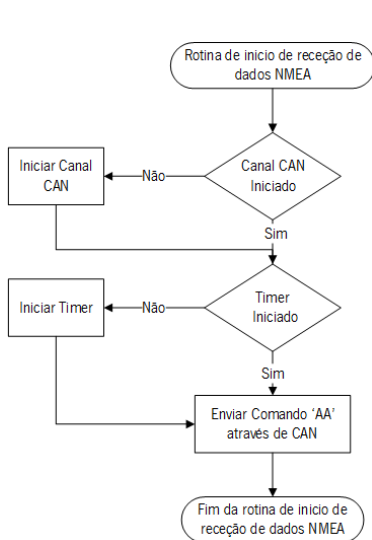


Figura 42 - Rotina de início de recepção de dados NMEA recebidos por CAN

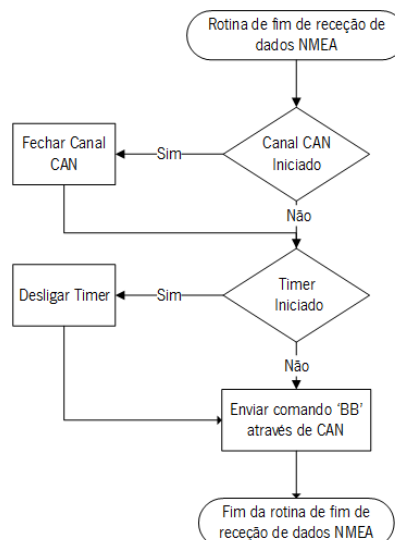


Figura 43 - Rotina de fim de recepção de dados NMEA recebidos por CAN

#### 4.2.5. Rotina de configuração do recetor GNSS

Esta rotina, apresentada na Figura 44, é responsável pela geração da mensagem de configuração a enviar para o microcontrolador configurar o recetor GNSS.

A primeira ação desta rotina é gerar a mensagem tendo em consideração as opções do utilizador na interface, pois estas configurações permitem escolher que constelações o recetor deve adquirir, quantos canais disponibilizar para cada constelação e outras opções.

A mensagem é gerada seguindo o protocolo UBX, detalhado no subcapítulo 3.5.4, e como este protocolo contém mecanismo de controlo de erros de transmissão, é efetuado o cálculo do Checksum e é adicionado à mensagem. Além das opções disponíveis para o utilizador, a mensagem contém também

dois bytes relativos á classe a qual pertence a mensagem, e o Identificador da mesma. Estes são adicionados no inicio da mensagem.

De seguida, como a mensagem UBX de configuração GNSS contem sempre mais de 8 Bytes, a mensagem gerada é dividida em parcelas de 8 bytes para ser enviada através de CAN-Bus para o microcontrolador. Como o protocolo UBX não conta com caracteres de terminação e o seu tamanho é variável, implementou-se um mecanismo de controlo para que o microcontrolador consiga refazer a mensagem recebida por CAN. O mecanismo implementado, e descrito na Tabela 13, consiste em adicionar um byte de controlo no início da mensagem em que o primeiro nibble é o total de mensagens e o segundo é numero da própria mensagem. Como pode ser observado no subcapítulo 4.1.1 para que o microcontrolador identifique estas mensagens como mensagens de configuração, o primeiro Byte deve ter o valor 'CC'. Por isso é adicionado no inicio da mensagem este valor.

Um exemplo das mensagens de configuração enviadas para o microcontrolador pode ser analisado na Tabela 14.

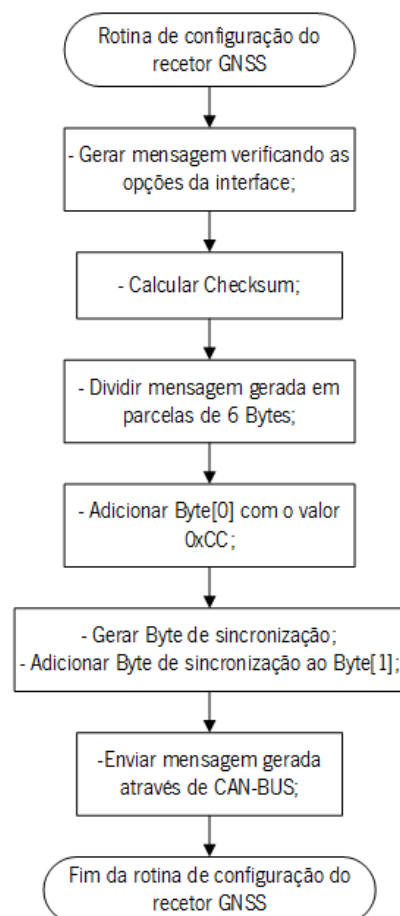


Figura 44 - Rotina de geração e envio da mensagem UBX de configuração do recetor GNSS

Tabela 13 - Método de geração do Byte de sincronismo das mensagens CAN

Passo	Descrição do passo	Total de mensagens (primeiro <i>Nibble</i> )	Numero da mensagem (segundo <i>Nibble</i> )
1º	Valor inicial	0011 (0x03)	0001 (0x01)
2º	Rodar 1º <i>Nibble</i> 4 vezes á esquerda	00110000 (0x30)	0x01 (0x01)
3º	Soma lógica entre o 1º e 2º <i>Nibble</i>	00110001 (0x31)	

Tabela 14 - Divisão da mensagem UBX em mensagens CAN-BUS

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0xCC	0x31	0x06	0x3E	12	0	0	0	0xCC	0x32	32	1	0	8	16	0	0xCC	0x33	1	0	0	1	0x8B	0x90
Mensagem CAN 1								Mensagem CAN 2								Mensagem CAN 3							

#### 4.2.6. Rotina de apresentação do mapa

Esta rotina, apresentada na Figura 45 é responsável por mostrar a imagem de um mapa com a posição adquirida pelo recetor GNSS marcada. Quando é ativada, por ação do utilizador, a rotina verifica a ultima posição gravada no sistema pela rotina ‘Processamento de dados NMEA’. Depois de verificar a validade da mesma, esta rotina apresenta uma imagem centrada na posição adquirida, com um marcador na mesma. No caso de a posição ser inválida ou inexistente é apresentada uma mensagem de erro.

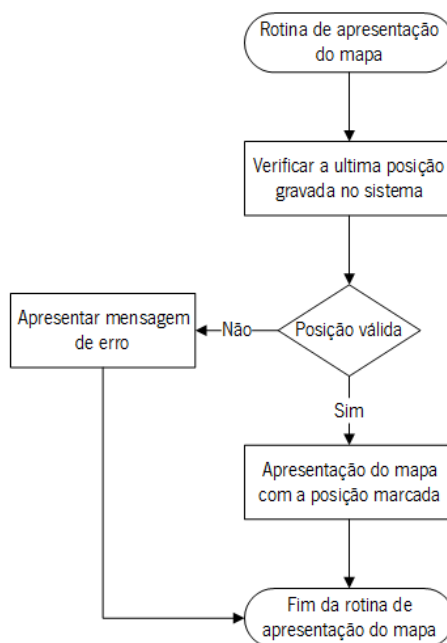


Figura 45 - Rotina para apresentação do mapa com a ultima posição adquirida

## 4.2.7. TTFF – Time To First Fix

O diagrama de software ilustrado na Figura 46, apresenta a sequência para testar o tempo necessário para a primeira aquisição com os diferentes tipos de arranque (de fábrica, normal e em *standby*) inicia-se com o envio para o recetor do comando necessário para este efetuar o arranque pretendido: para o arranque de fábrica envia-se o comando para limpar toda a memória do recetor; no arranque normal apagam-se apenas a *Ephemeris* do sistema; no arranque em *standby* nenhuma informação é limpa, e é necessário que o recetor contenha os dados *Ephemeris* e *Almanac* ainda válidos e ainda que tenha adquirido posições válida nos últimos minutos.

De seguida, inicia-se um temporizador e este conta o tempo até que o recetor seja capaz de transmitir uma posição válida. A este tempo é subtraído o tempo necessário para a comunicação Computador -> Microcontrolador -> Recetor e vice-versa e o resultado é o valor considerado como o tempo necessário para a primeira aquisição.

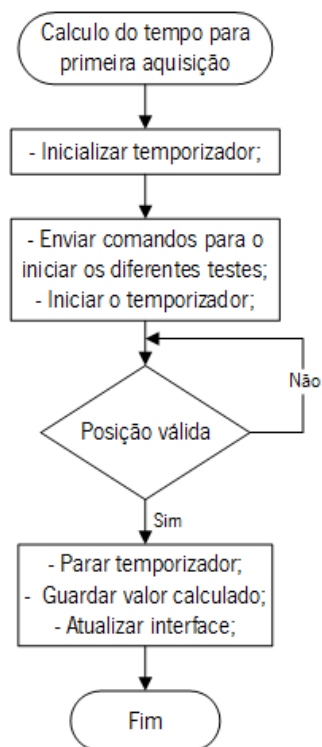


Figura 46 - Rotina para calcular o tempo para primeira aquisição do recetor nos diferentes tipos de arranque

## 4.2.8. Precisão de localização

O *software* para calcular a precisão do recetor utiliza as coordenadas das posições adquiridas para efetuar os cálculos. Existem dois testes a realizar com posições fixas: a posição relativa e absoluta e um teste para movimento.

Para o cálculo da posição absoluta é medida a distância entre a posição adquirida e a posição real do recetor, para o calculo da posição relativa a distância é medida entre a posição adquirida e a posição adquirida anteriormente e para o cálculo da posição em movimento (velocidade), é medida a distância entre a posição adquirida e a posição adquirida anteriormente e divide-se esse valor pelo tempo entre as aquisições e é obtida a velocidade.

Na Figura 47, apresenta-se um exemplo de dois pontos no mapa e a medição distância entre eles. Estes são os pontos utilizados como exemplo para demonstrar o método implementado para o calculo da distância. Deste modo é possível a comparação do resultado obtido pelo *software* implementado com a medida efetuada pela ferramenta do Google Maps.



Figura 47 - Distância medida entre dois pontos do mapa (utilizando o Google Maps – 187,34m)

Uma vez adquiridas as coordenadas, recorre-se à formula *haversine* [24] apresentada em (2), (3) e (4) que calcula a distância entre dois pontos numa esfera utilizando as latitudes e longitudes desses pontos.

$$LatitudePonto1_{grausDecimal} = 41.45260788N$$

$$LongitudePonto1_{grausDecimal} = 8.2905136W$$

$$LatitudePonto2_{grausDecimal} = 41.45328736N$$

$$LongitudePonto2_{grausDecimal} = 8.28845903W$$

*Haversine:*

$$a = \sin\left(\frac{\Delta\varphi}{2}\right)^2 + \cos\varphi_1 * \cos\varphi_2 * \sin\left(\frac{\Delta\lambda}{2}\right)^2 \quad (2)$$

$$c = 2 * atan2(\sqrt{a}, \sqrt{1-a}) \quad (3)$$

$$distancia = R * c \quad (4)$$

Onde:

- $\varphi$  é a latitude em radianos;
- $\lambda$  é a longitude em radianos;
- $R$  é o raio da terra aproximado (6371 Kms);

De seguida, utilizando o MATLAB para efetuar os cálculos, obteve-se a mesma distância que a ferramenta do Google, como é possível observar na Figura 48, provando que o software implementado está a calcular corretamente a distância.

```
>> haversine([41.45328736 -8.28845903], [41.45260788 -8.2905136])  
  
ans =  
  
0.1872|
```

Figura 48 - Resultado do cálculo da formula Haversine no Matlab (187.2m)

#### 4.2.9. Tempo de Reaquisição

O software implementado para calcular o tempo de reaquisição do sinal, inicia-se pelo envio de um sinal para o gerador indicando ao mesmo para não gerar sinal durante alguns segundos, de seguida envia-se um comando indicando a retoma da geração e é iniciado um temporizador que contabiliza o tempo necessário até o recetor calcular uma posição válida novamente.

#### 4.2.10. Sensibilidade

Como se pode observar no diagrama da Figura 49, o *software* responsável pelo cálculo deste parâmetro inicia-se com a geração de um sinal GNSS com um nível de potência elevado, de modo a que o recetor adquira uma posição válida, de seguida é diminuída iterativamente a potencia do sinal até que o recetor não consiga adquirir uma posição. É efetuado este teste tanto no momento da primeira aquisição, como em funcionamento.



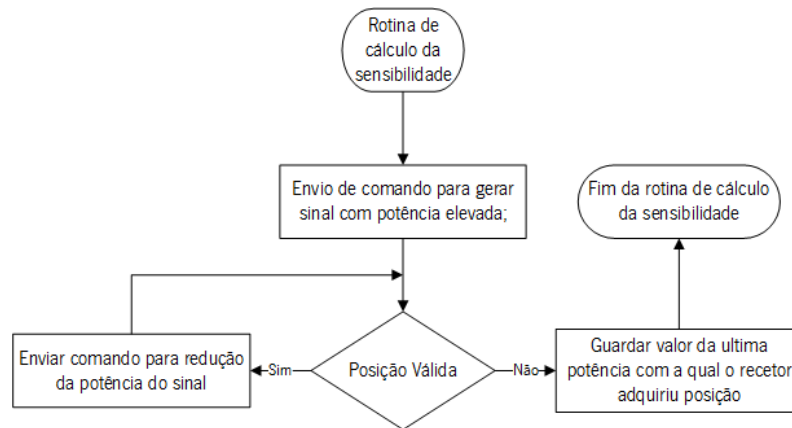


Figura 49 -Diagrama de blocos da sequência do teste de sensibilidade

#### 4.2.11. Interferências

O *software* de teste da imunidade do recetor GNSS a interferências, dos mais variados tipos, tem o mesmo princípio do teste de sensibilidade, todavia, neste é enviado um comando iterativamente para o gerador, para elevar a potência das interferências simuladas e é guardado o valor da última potência de interferências com a qual o recetor é capaz de adquirir uma posição.



## 5. TESTES E RESULTADOS

Para alcançar um sistema final funcional foi necessário o teste individual de cada um dos elementos que constituem o sistema. Assim foi possível validar e verificar as limitações de cada um deles, permitindo obter o sistema final caracterizado e com todos os componentes implementados a operar conforme o planeado.

### 5.1. Teste aos componentes do sistema

#### 5.1.1. Recetor GNSS

Este foi um dos elementos mais testados individualmente, pois foi necessária a implementação de dois protocolos de comunicação (NMEA e UBX). Por isso, para um teste mais completo, foi desenvolvido um pequeno programa de testes em C#, onde se implementou a comunicação com o recetor utilizando ambos os protocolos. Para acelerar o processo de teste foi utilizada a UART como meio de comunicação.

Na Figura 50, pode-se observar a interface que apresenta os resultados de algumas mensagens recebidas utilizando o protocolo NMEA.

The screenshot shows a Windows application window titled 'Form1' with a configuration section at the top and a data display section below. The configuration section includes dropdown menus for Port (COM9), Baud Rate (9600), Stop Bits (One), Parity (None), Data Bits (8), and Flow Control (None), along with 'Check Ports', 'Disconnect', and 'Connect' buttons. The data display section has tabs for 'Debug', 'Config GNSS', 'NMEA Data', and 'Available SV'. The 'NMEA Data' tab is active, showing the following data:

<b>RMC</b>						
UTC Time	Data Status	Latitude	Longitude	Speed Over Ground (knots)	Course Over Ground (Deg)	UTC Date
15:05:29	Valid	4127.14357 N	00817.40382 W	0.064	Unavailable	31/05/16

<b>VTG</b>		
Course Over Ground (Deg)	Speed Over Ground (knots)	Speed Over Ground (km/h)
Unavailable	0.064	0.118

<b>GGA</b>							
UTC Time	Latitude	Longitude	Quality	N° Satellites	HDOP	Altitude	Geoid Separation
15:05:29	4127.14357 N	00817.40382 W	Standard	12	0.87	216.9	50.3

<b>GSA</b>				
Operation Mode	Navigation Mode	PDOP	HDOP	VDOP
atically (2D/3D)	3D	1.52	0.87	1.25

Below the tables, there are several empty text input fields. The first one contains the text: GNGLL,4127.14357,N,00817.40382,W,150529.00,A,A\*68. The second one contains: GNZDA,150529.00,31,05,2016,00,00\*70. The third one contains: GNGST,150529.00,70,...29.1.4.4.6\*48. To the right of these fields is a 'Position Mode' label and a dropdown menu.

Figura 50 - Interface gráfica implementada em C# com informação extraída das mensagens NMEA

Na Figura 51 observa-se a interface de configuração para o recetor GNSS utilizando o protocolo UBX.

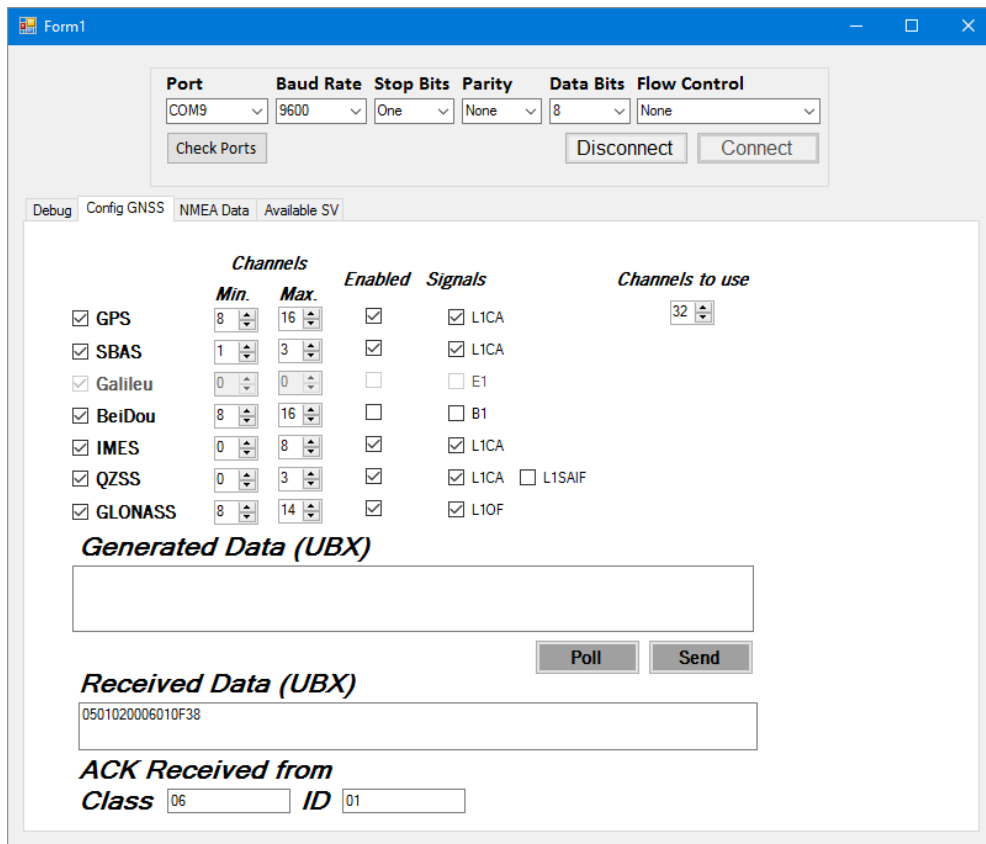


Figura 51 - Interface gráfica implementada em C# de configuração do recetor GNSS através do protocolo UBX

Depois de implementados os dois protocolos testaram-se algumas das funcionalidades possíveis com os dados recebidos do recetor GNSS Na Figura 52 observa-se um exemplo onde a verde estão os satélites disponíveis e a vermelho os que não estão visíveis.

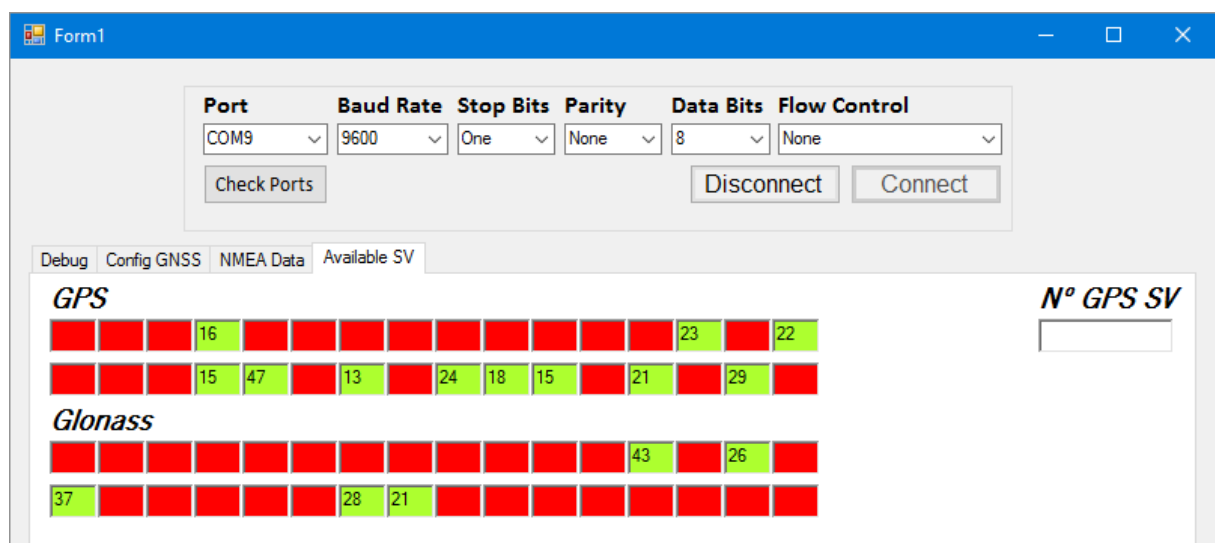


Figura 52 - Exemplo de utilização da informação recebida do recetor GNSS

### 5.1.2. Comunicação CAN-Bus

A comunicação CAN-Bus fez parte dos requisitos do sistema e foi um dos elementos que mais dificuldades ofereceu devido à complexidade da configuração do protocolo no microcontrolador. Esta complexidade é causada pelas muitas opções disponíveis pelo microcontrolador para o protocolo.

Uma vez configurado o protocolo CAN no microcontrolador procedeu-se à configuração do ASIC desenvolvido pela Bosch para funcionar como *transceiver* CAN-BUS. Para finalizar os testes ao protocolo, utilizou-se o adaptador PEAK CAN Pro para comunicar com o computador. Aqui, recorreu-se ao programa (PCAN-View) disponibilizado pela empresa do adaptador para validar a comunicação com o microcontrolador e testar os comandos a enviar para o mesmo.

### 5.1.3. Comunicação SPI

A implementação deste protocolo de comunicação não é trivial devido aos extensivos parâmetros de configuração oferecidos pelo microcontrolador para as comunicações. Por isso, para validar a fiabilidade da comunicação do recetor GNSS com o microcontrolador efetuaram-se um conjunto de testes onde se verificou a taxa de transmissão máxima permitida por ambos os dispositivos e foi testada a melhor opção entre enviar mensagens (com um *Byte* de cada vez ou agrupá-los na mesma mensagem). Verificou-se também se o microcontrolador estava a transmitir corretamente para o recetor através do envio de uma mensagem de configuração GNSS e posterior leitura da mesma utilizando um *software* proprietário do fabricante do recetor.

### 5.1.4. Aquisição de posições

O teste inicial para o verificar se o recetor estava a adquirir a posição correta foi efetuado colocando o recetor a adquirir posições e a transmitir as mesmas para o computador. Posteriormente colocaram-se as coordenadas no mapa e verificou-se se a posição estava correta.

O teste final à aquisição foi efetuado com a utilização de um gerador GNSS e verificou-se se a posição adquirida era compatível com a gerada.

## Resultados funcionais

### 5.1.5. Precisão

O teste realizado relativo à precisão foi efetuado através da geração de uma posição fixa utilizando o gerador GNSS LabSat3 [25]. Gerou-se o sinal GNSS utilizando a constelação GPS e a posição gerada foi '51.98954050 -0.99137767' (edifício Bosch em Inglaterra). Na Figura 53, observa-se a coordenada gerada no Google Maps.

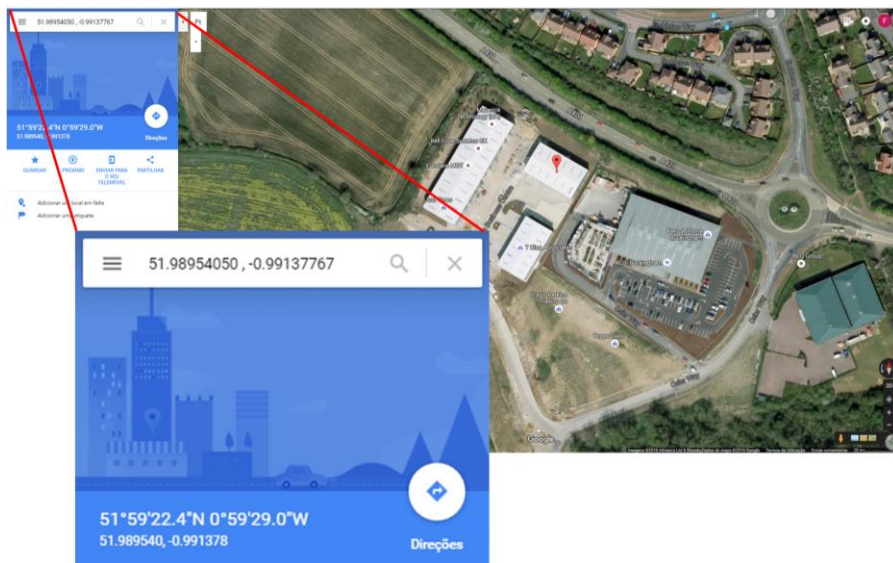


Figura 53 - Posição Gerada apresentada no Google Maps

De seguida, utilizando o sistema implementado verificou-se se a posição adquirida pelo recetor é próxima da posição gerada. Como forma de análise de alto nível, gerou-se em matlab um mapa centrado na posição adquirida. Analisando a Figura 54, observa-se que a posição adquirida é muito próxima da gerada.

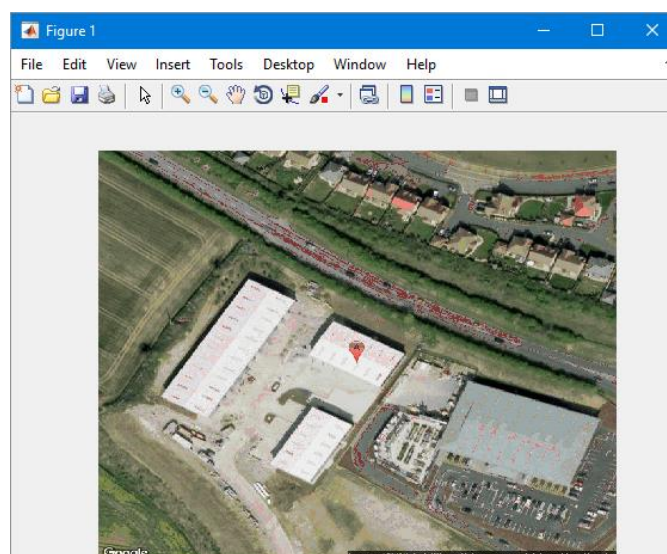


Figura 54 - Mapa gerado no Matlab utilizando a posição adquirida pelo recetor GNSS

### 5.1.6. Precisão Absoluta

Para o cálculo da posição absoluta gerou-se continuamente a mesma coordenada e adquiriram-se posições durante alguns segundos. De seguida, calculou-se a distância entre a posição gerada e a posição adquirida para todas as aquisições e posteriormente efetuou-se a média desses valores. O gráfico apresentado na Figura 55 exibe a distância entre a posição gerada e as 95 posições adquiridas.

A média das distâncias absolutas medidas neste teste foi de 10.36 cm.

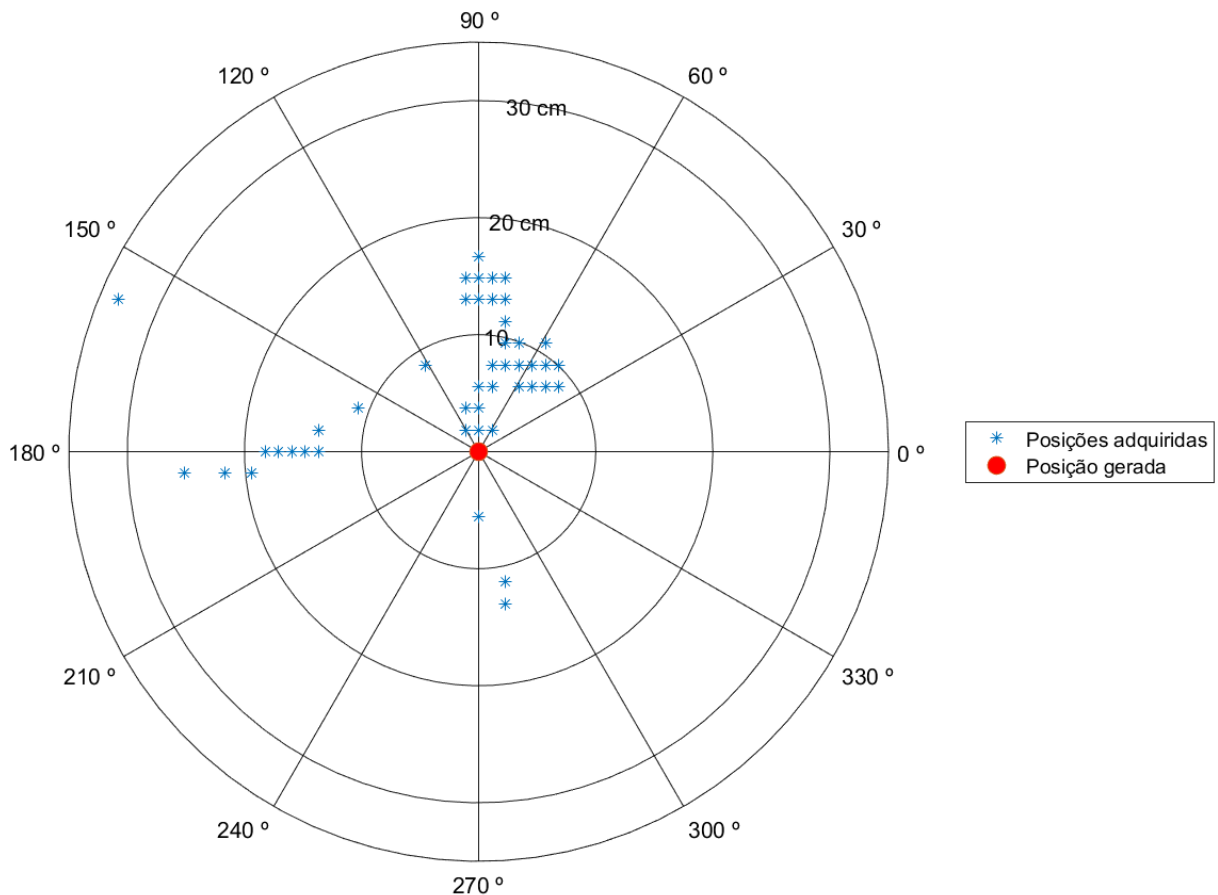


Figura 55 - Gráfico com a diferença entre a posição gerada e as posições adquiridas

### 5.1.7. Posição Relativa

Para testar o sistema quanto à caracterização da posição relativa foram utilizadas as mesmas amostras utilizadas para o cálculo da posição absoluta (5.1.6), contudo, a distância é calculada relativamente à posição adquirida imediatamente anterior. A média das distâncias relativas medidas é 1.59 cm;

## 5. Testes e Resultados

### 5.1.8. TTFF – Tempo para a primeira aquisição

O teste efetuado para caracterizar o recetor quanto ao tempo necessário para a aquisição da primeira posição nos diferentes tipos de arranque foi realizado sem o gerador, utilizando os sinais reais de GNSS.

O software enviou o comando para fazer reset ao recetor através dos três tipos de arranque e contou o tempo até que o recetor adquiriu a primeira posição:

- O tempo necessário para a primeira aquisição a partir do arranque de fábrica (*cold start*) foi 26.7 segundos, com o recetor a adquirir sinais GPS e GLONASS;
- Para a primeira aquisição utilizando o método de arranque normal (*warm start*) o tempo despendido pelo recetor foi 19 segundos.
- Utilizando o arranque em *standby* (hot start) o tempo que o recetor necessitou foi 5 segundos.

Os valores obtidos foram resultado de múltiplas tentativas e o valor apresentado é a média dos mesmos.

### 5.1.9. Tempo de reaquisição

Para a realização deste teste ligou-se o recetor GNSS durante alguns minutos, de seguida, desligou-se o gerador durante 10 segundos. Voltando a liga-lo contou-se o tempo necessário até o recetor readquirir uma posição.

Foram efetuadas várias repetições e a média de todos foi de 0.5 segundos.



## 6. CONCLUSÕES E TRABALHO FUTURO

No decorrer desta dissertação desenvolveu-se um sistema que permite a aquisição de sinais GNSS e a caracterização do recetor dos mesmos. Este consiste na geração de sinais GNSS perfeitamente conhecidos e na posterior aquisição dos mesmos utilizando um recetor GNSS. Com os dados adquiridos, são efetuados um conjunto de cálculos com o intuito de caracterizar o recetor seguindo um conjunto de métricas.

O principal objetivo de implementar um sistema capaz de adquirir e caracterizar um recetor GNSS foi cumprido. Contudo, para uma validação mais aprofundada do sistema, são necessários mais testes, inclusive o cálculo da precisão dinâmica, sensibilidades e interferências.

Ainda assim, os resultados foram promissores alcançando-se um sistema funcional com capacidade para caracterizar diferentes recetores GNSS implementado com um microcontrolador certificado para a indústria automóvel, possibilitando a caracterização dos sensores a incorporar no projeto APP.

Tratando-se de um microcontrolador recente, de gama profissional e com pouca informação disponível, as maiores dificuldades no desenvolvimento do software do mesmo foram as configurações iniciais dos protocolos de comunicação CAN-Bus e SPI.

No hardware, devido ao facto de esta dissertação se inserir no projeto APP e o *hardware* ser escolhido em conjunto com a Bosch, as maiores dificuldades foram a aquisição dos componentes com a celeridade necessária. Devido a isso, não foi possível efetuar todos os testes propostos inicialmente pois não foi possível implementar a comunicação entre o computador e o gerador atempadamente, e por isso os testes efetuados utilizaram o *software* proprietário do recetor.

Futuramente, deverá ser implementada a comunicação entre o gerador e computador permitindo assim alargar o conjunto de testes efetuados ao recetor, permitindo uma caracterização mais completa.

Devem ainda ser testados algoritmos mais completos para o cálculo da distância entre posições, completar a interface de testes consoante os novos testes e otimizar a interligação de todos os componentes criando um sistema mais fiável.



## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Bosch - Automated driving. Available: [http://products.bosch-mobility-solutions.com/en/de/specials/specials\\_for\\_more\\_driving\\_safety/automated\\_driving/motivation\\_and\\_conditions/levels\\_of\\_automation\\_1/levels\\_of\\_automation.html](http://products.bosch-mobility-solutions.com/en/de/specials/specials_for_more_driving_safety/automated_driving/motivation_and_conditions/levels_of_automation_1/levels_of_automation.html)  
Available:[http://products.bosch-mobility-solutions.com/en/de/driving\\_safety/driving\\_safety\\_systems\\_for\\_passenger\\_cars\\_1/driver\\_assistance\\_systems/driver\\_assistance\\_systems\\_2.html](http://products.bosch-mobility-solutions.com/en/de/driving_safety/driving_safety_systems_for_passenger_cars_1/driver_assistance_systems/driver_assistance_systems_2.html)
- [2] M. B. Bokov, A. Edelkina, M. Klubova, T. Thurner, N. P. Velikanova, and K. Vishnevskiy, "“After 300 meters turn right” –the future of Russia’s GLONASS and the development of global satellite navigation systems," *foresight*, vol. 16, no. 5, pp. 448-461, 2014.
- [3] G. Girardin and E. Mounier. (2015). Sensors and Data Management for Autonomous Vehicles report 2015. Available: [http://www.i-micronews.com/images/Flyers/MEMS/Yole\\_Sensors\\_and\\_Data\\_Management\\_for\\_Autonomous\\_Vehicles\\_Report\\_October\\_2015\\_web.pdf](http://www.i-micronews.com/images/Flyers/MEMS/Yole_Sensors_and_Data_Management_for_Autonomous_Vehicles_Report_October_2015_web.pdf)
- [4] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game-theoretical formulation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 584-596, 2007.
- [5] (2016). Novatel - GPS. Available: <http://www.novatel.com/an-introduction-to-gnss/chapter-3-satellite-systems/gps/>
- [6] Imar. (2015). iTraceRT-F402-E. Available: [http://gnss.co.jp/wp-content/uploads/2015/10/TraceRT-F402-E\\_en.pdf](http://gnss.co.jp/wp-content/uploads/2015/10/TraceRT-F402-E_en.pdf)
- [7] (2015). Triangulação 3D. Available: [https://openclipart.org/image/2400px/svg\\_to\\_png/191659/GPS-3D-trilateration.png](https://openclipart.org/image/2400px/svg_to_png/191659/GPS-3D-trilateration.png)
- [8] J. S. Subirana, J. M. J. Zornoza, M. Hernández-Pajares, E. S. Agency, and K. Fletcher, *GNSS Data Processing* (no. vol. 1). ESA Communications, 2013.
- [9] A. Technologies, "GNSS Technologies and Receiver Testing," ed, 2013.
- [10] K. Borre, "The E1 Galileo Signal," Aalborg Univ., Denmark [Online]. Available: [http://waas.stanford.edu/papers/Borre/galileo\\_sig.pdf](http://waas.stanford.edu/papers/Borre/galileo_sig.pdf), accessed on Mar, vol. 9, p. 2015, 2009.
- [11] G. Seepersad. (2015). How is the distance between a GPS device and a GPS satellite calculated? Available: <https://www.quora.com/How-is-the-distance-between-a-GPS-device-and-a-GPS-satellite-calculated>
- [12] A. Technologies, "Application Note 'GPS Receiver Testing'," ed, 2010.
- [13] Bosch. (2015). Degrees of Automations. Available: [http://products.bosch-mobility-solutions.com/en/de/specials/specials\\_safety/automated\\_driving/motivation\\_and\\_conditions/levels\\_of\\_automation\\_1/levels\\_of\\_automation.html](http://products.bosch-mobility-solutions.com/en/de/specials/specials_safety/automated_driving/motivation_and_conditions/levels_of_automation_1/levels_of_automation.html)
- [14] Rohde&Schwarz. GPS, Glonass, Galileo Receiver Testing Using a GNSS Signal Simulator. Available: [https://cdn.rohde-schwarz.com/pws/dl\\_downloads/dl\\_application/application\\_notes/1gp86/1GP86\\_2E\\_GNSS\\_Receiver\\_Testing.pdf](https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1gp86/1GP86_2E_GNSS_Receiver_Testing.pdf)
- [15] R. Alves Pena. Latitudes e Longitudes. Available: <http://escolakids.uol.com.br/latitudes-e-longitudes.htm>
- [16] U-Blox, "GNSS Receptor M8T-Evk," ed, 2015.
- [17] uBlox, "NEO/LEA-M8T, u-blox M8 concurrent GNSS timing modules," ed, 2016.
- [18] Wikipédia. Spi - Serial Peripheral Interface. Available: [https://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface\\_Bus](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)

- [19] gpsinformation. NMEA data. Available: <http://www.gpsinformation.org/dale/nmea.htm#nmea>
- [20] u-Blox, "u-blox M8 concurrent GNSS timing modules - Data Sheet," R03 ed, 2016.
- [21] T. Instruments, "Introduction to the Controller Area Network (CAN)," ed, 2008.
- [22] A. Technologies. GPS Receiver Test. Available: <http://cp.literature.agilent.com/litweb/pdf/5990-4943EN.pdf>
- [23] Spirent, "The fundamentals of GNSS Simulation," ed.
- [24] C. d. a. m. b. L. L. points. Available: <http://www.movable-type.co.uk/scripts/latlong.html>
- [25] LabSat, "LabSat3 - Datasheet," ed, 2015.