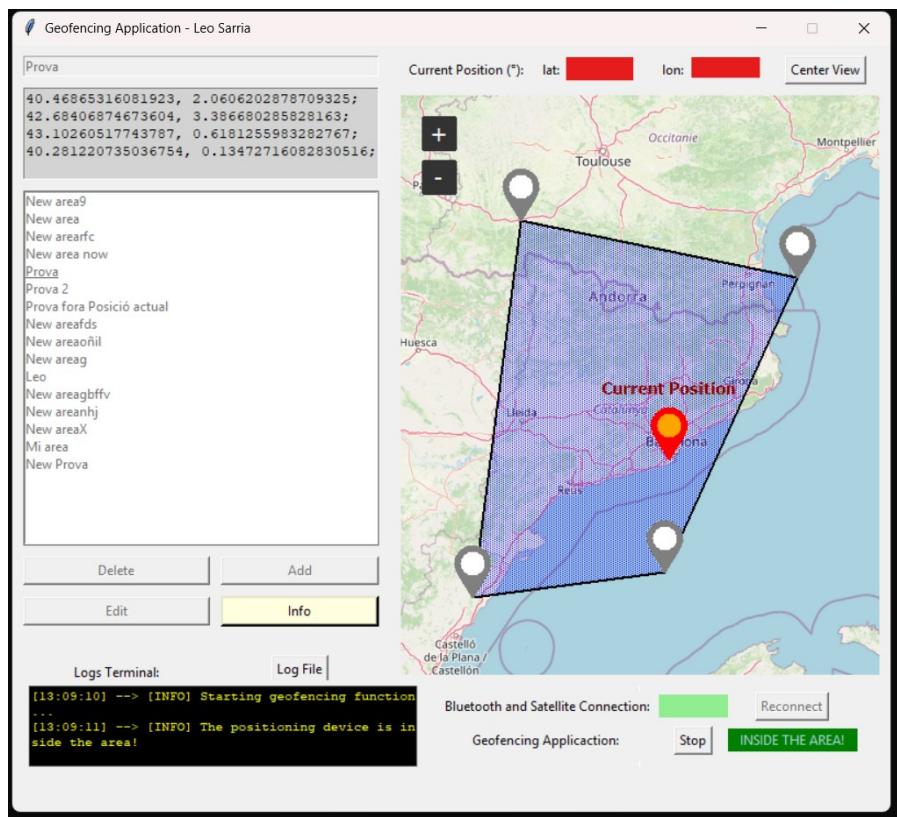


Pestanya 1

TREBALL DE RECERCA

Desenvolupament d'una aplicació de geofencing amb enfocament SDR



Autor: Leo Sarria Bertán
Tutor d'institut: Ramón Baena Iniesta
Tutor programa Argó: Gonzalo Seco Granados

INS Salvador Dalí
Octubre de 2025

Agraïments

Aquest projecte només ha estat possible gràcies al programa Argó, i per això agraeixo molt l'oportunitat que m'han ofert.

Aquest és un programa organitzat per la UAB, on part del professorat ofereix assessorament per al desenvolupament del treball de recerca, tema del qual assignen ells.

Dins d'aquest programa, agraeixo especialment l'ajuda i suport del meu tutor; Gonzalo Seco, qui m'ha presentat el tema. En fer-ho, m'ha obert les portes a tot el món de les telecomunicacions i els radioaficionats, fins ara desconegut per mi. Gràcies a ell he pogut experimentar amb aquestes tecnologies. També, la seva orientació i gran experiència han estat un gran suport.

Pel que fa al meu tutor del centre, Ramón Baena, no puc fer més que donar-li les gràcies. Ell va acceptar entrar en aquest programa, recolzant-me més del que se li demanava. El seu acompanyament i supervisió acadèmica han fet que aquesta proposta prengués la forma que ara té.

I finalment, la meva família. Aquesta sempre m'ha fet costat, i els seus punts de vista i suggerències m'han permès veure el veritable valor de l'esforç destinat, tant en el treball escrit com a l'aplicació final, i els seus aspectes de millora. Sense ells, aquest projecte no seria el mateix.

Resumen

Este es un proyecto que trata de comprender las bases de los sistemas GNSS (Global Navigation Satellite System) y del concepto SDR (Software Defined Radio), con la finalidad de aplicarlo para obtener un posicionamiento en tiempo real. A partir de ese marco teórico y de dicho posicionamiento, desarrollar una aplicación de delimitación geográfica virtual (*geofencing*) empleando el lenguaje de programación Python.

Aunque no se ha logrado emplear la metodología SDR, se ha optado por usar un módulo GPS convencional conectado a una placa microcontroladora. La aplicación se ha desarrollado progresivamente, aplicando diferentes versiones con el fin de mejorar la interfaz y sus funciones, hasta obtener la versión final. Ambos elementos se han unido mediante Bluetooth.

El proyecto se puede considerar exitoso, pese a que no se han cumplido todos los objetivos. Esta afirmación se debe a que se han asimilado todos los conceptos trabajados durante su desarrollo, e incluso elementos que no aparecen explícitamente como instalación y propiedades de las aplicaciones, así como diferentes entornos operativos sobre los que trabajar. Además de haber logrado desarrollar la aplicación de geofencing, que es funcional e intuitiva.

Abstract

This project focuses on understanding the GNSS (Global Navigation Satellite System) and SDR (Software Defined Radio) fundamentals, with the purpose of applying them to accomplish a real-time positioning. Starting from this theoretical framework and the mentioned positioning, search for the development of a geographic virtual delimitation (*geofencing*) application using the programming language Python.

Although it has not been possible to apply the SDR methodology, a conventional GPS module was used, connected to a microcontroller board. The application has been developed progressively, generating different variants in order to achieve a better user interface and functionality, until reaching its final version. Both elements have been connected via Bluetooth.

The project can be considered successful, even though not all the objectives have been completed. This is because most of the concepts worked in this project have been assimilated, even those that don't appear explicitly such as the installation and configuration of applications, as well as different operating systems to work with. Also because of the correct development of the geofencing application, which is functional and intuitive.

Taula de continguts:

INTRODUCCIÓ.....	1
MARC TEÒRIC.....	3
1. Definició de GNSS i introducció històrica.....	3
1.1. Definició.....	3
1.2. Història del primer sistema GNSS.....	3
1.3. Sistemes GNSS i les seves característiques.....	4
2. Concepte dels sistemes GNSS.....	5
2.1. Funcionament simplificat, triangulació bidimensional ideal.....	5
2.2. Triangulació tridimensional ideal.....	6
2.3. Triangulació bidimensional i tridimensional amb errors.....	7
3. Elements d'un sistema GNSS.....	8
3.1. Segment espacial.....	8
3.1.1. Nombre de satèl·lits.....	8
3.1.2. Òrbites i els seus efectes.....	9
3.1.3. Inclinació orbital.....	10
3.1.4. Satèl·lits.....	11
3.2. Segment de control o terrestre.....	12
3.2.1. Components del segment de control.....	12
3.3. Segment d'usuari.....	12
3.3.1. Components principals.....	13
3.4. Funcionament global dels sistemes GNSS.....	13
4. Senyals, fonts d'error i càcul de la posició.....	14
4.1. Senyals GPS.....	14
4.2. Fonts d'error i limitacions actual del sistema.....	15
4.2.1. Errors ionosfèrics.....	15
4.2.2. Errors troposfèrics.....	15
4.2.3. Errors d'efemèrides.....	15
4.2.4. Errors de multipath.....	16
4.2.5. Error de rellotge (satèl·lital i del receptor).....	16
4.2.6. Geometria dels satèl·lits.....	17
4.3. Resolució matemàtica de la posició i el temps.....	17
5. Mètodes per a millorar la precisió.....	19
6. Característiques del SDR.....	20
6.1. Concepte d'SDR.....	20
6.2. Avantatges i desavantatges de la implementació de la tecnologia SDR.....	20
6.3. Funcionament bàsic.....	21
MARC PRÀCTIC.....	22
7. Metodologia.....	22
8. Equipament del projecte (hardware).....	22
9. Programari del projecte i entorns de programació.....	25
10. Posicionament inicial.....	29
10.1. Configuració de hardware i software inicial, error “0x00007b”.....	29
10.2. Primera execució GNSS-SDR + RTKnavi.....	30

10.3. Aïllament de l'error i hipòtesis.....	31
11. Prova de l'entorn WSL2 (Windows Subsystem for Linux).....	33
11.1. Justificació del canvi d'entorn i explicació del programa utilitzat.....	33
11.2. Instal·lació de l'entorn, configuració de ports i instal·lació de GNSS-SDR.....	34
11.3. Prova de posicionament en temps real.....	35
11.4. Gravació de dades i posterior anàlisi.....	35
11.5. Prova amb rtl_test.....	36
11.6. Seguiment del tutorial “My First Fix” de gnss-sdr.org.....	36
11.7. Conclusió i tancament de l'experimentació amb SDR.....	37
12. Posicionament amb el mòdul GPS NEO-7M.....	38
12.1. Connexionat del mòdul amb el microcontrolador.....	38
12.2. Prova de funcionament amb STEAMakersBlocks.....	39
12.3. Codi final amb Arduino IDE, comunicació pel port o per Bluetooth.....	40
13. Desenvolupament de l'aplicació; idea inicial i selecció del mapa a utilitzar.....	40
13.1. Idea inicial.....	40
14. Desenvolupament de l'aplicació; construcció de l'UI.....	41
15. Desenvolupament de l'aplicació: evolució de les versions.....	42
15.1. Versió 1; UI inicial i gestió d'àrees.....	42
15.2. Versió 1.2; reestructuració del codi.....	43
15.3. Versió 2; visualització d'àrees.....	43
15.4. Versió 3; posicionament en temps real i comunicació Bluetooth.....	44
15.5. Versió 4; edició intuïtiva d'àrees.....	45
15.6. Versió 4.2; millora del sistema de connexió Bluetooth.....	46
15.7. Versió 4.3; gestió de la configuració i registre de logs.....	46
15.8. Versió 5; Versió final.....	47
16. Informació sobre l'aplicació.....	49
17. Pressupost global del projecte.....	49
CONCLUSIONS.....	51
ENLLAÇOS IMPORTANTS.....	52
BIBLIOGRAFIA.....	53
ANNEXES.....	55

INTRODUCCIÓ

Per tant, el treball es tracta d'un tema proporcionat pel tutor de la UAB, tot i que ha estat escollit entre d'altres. En aquest cas, el tema és "Desenvolupament d'un receptor de GPS o d'altres aplicacions amb la metodologia SDR (Software-Defined Radio)". No obstant això, des de l'inici s'ha deixat clar que aquesta era la base, i que es deixava, i fins i tot es pretenia, que s'implementés en algun context. En aquest punt entra en joc el segon element d'aquest pla de treball, i és la programació, que es volia incorporar com fos possible. Es van plantejar diverses idees, algunes objectivament inabastables com el desenvolupament d'un receptor propi. Però finalment es va arribar a la idea final, que permetia la implementació de la metodologia SDR per al posicionament en temps real.

Per tot això, la motivació abans de començar aquest treball no es pot justificar, ja que es desconeixien aquests sistemes. Quan va començar el projecte, aquest va suposar el descobriment personal dels sistemes de posicionament, i més encara, de les telecomunicacions com a concepte i tecnologia. És a dir, que l'atractiu d'aquest treball ha estat el descobriment de tot el que el constitueix, l'adquisició de coneixement i l'experimentació que ha ofert.

Els objectius del treball són abordar l'estudi i síntesi dels GNSS (Global Navigation Satellite System, és a dir, sistema global de navegació per satèl·lit), així com de la metodologia SDR, amb la finalitat d'aplicar-ho en el desenvolupament d'una aplicació de geofencing (delimitació geogràfica virtual). A més, s'han pensat possibles propostes per a ampliar la part pràctica, i poder comprendre millor la diferència entre un receptor GPS tradicional i un basat en SDR, la que es pretenia portar a terme era la recaptació de dades d'ambdós sistemes en el mateix entorn per tal de comparar-los objectivament, però finalment no ha estat possible.

Per descomptat, hi ha múltiples objectius secundaris, com l'experimentació amb el dispositiu SDR per tal d'avaluar el seu potencial en múltiples aplicacions, la millora de l'estil de programació, modularització i escalabilitat dels projectes, etc.

En resum, la hipòtesi del treball global és que és possible desenvolupar una aplicació de geofencing amb posicionament en temps real basat en la metodologia SDR, a partir dels conceptes que s'introdueixen a la part teòrica.

L'estructura del treball es caracteritza per trobar-se dividida en tres parts:

En referent a la secció teòrica, la seva finalitat és introduir tots els conceptes rellevants en sistemes GNSS (Global Navigation Satellite System), així com la metodologia SDR (Software Defined Radio). Es cerca brindar el marc teòric necessari per a poder comprendre tots els processos que constituiran la part pràctica d'aquest projecte.

Es tractarà, de manera general, la definició de les tecnologies GNSS i el seu origen.

S'introduirà de manera didàctica i visual la idea de posicionament, incorporant conceptes com la tridimensionalitat i l'error. A més a més, s'explicaran els elements intrínsecos de la mateixa infraestructura i les seves característiques. Finalment, es definirà una ona pròpia d'aquests sistemes a partir de les seves prestacions, els errors que poden afectar els seus resultats, i els termes que prenen part en la resolució matemàtica del posicionament final, a part dels mètodes per millorar la precisió d'aquests sistemes.

En últim lloc, s'exposarà la tecnologia SDR, definint-la i avaluant els seus avantatges i desavantatges enfront de tecnologies especialitzades.

Respecte a la part pràctica, serà dedicada a l'aplicació dels conceptes i tecnologies explorades fins ara. L'objectiu, el qual ja ha estat descrit, es tracta del desenvolupament d'una aplicació de geofencing. Però no es pot passar directament a aquest apartat, ja que prèviament s'ha fet molta experimentació, especialment en l'obtenció d'una posició en temps real que sigui fiable i estable. Prèviament s'explicarà la metodologia que el definirà, així com les eines emprades tant físiques com digitals.

Al llarg d'aquest apartat s'exposarà en detall la problemàtica instal·lació del material necessari. També com s'ha provat, de diverses maneres però sense èxit, a implementar la tecnologia SDR. I com després de proves i propostes sense cap resultat tangible, les quals s'han allargat per més d'un mes i mig, s'ha decidit de canviar la font del posicionament.

En aquest punt, s'explicarà com s'ha programat la placa microcontroladora per a rebre informació del mòdul GPS, i com aquesta envia la posició a través de Bluetooth. Així com el desenvolupament de l'aplicació.

Gran part del material el qual es referència en aquest escrit es pot trobar a una carpeta que s'ha penjat al Drive, l'enllaç de la qual es pot trobar al final del document. D'igual manera, s'ha publicat el codi del projecte a un repositori de GitHub, perquè es pugui consultar i descarregar, i l'enllaç es troba amb el del Drive.

MARC TEÒRIC

1. Definició de GNSS i introducció històrica

1.1. Definició

GNSS és una sigla que fa referència a “Global Navigation Satellite System”, o sistema global de navegació per satèl·lit. Aquest és el terme genèric amb què es considera qualsevol sistema satel·lital capaç de proporcionar un posicionament precís en l'espai amb cobertura global.

1.2. Història del primer sistema GNSS

Els sistemes de posicionament van sorgir de la necessitat militar de determinar la posició en temps real de qualsevol dispositiu. Van ser els Estats Units, durant la guerra freda (1947-1991), els primers a experimentar amb aquesta tecnologia, amb el resultat de la invenció del GPS. L'objectiu inicial era poder determinar la posició de submarins Polaris, els quals portaven míssils nuclears. Un dels primers sistemes de posicionament va ser el projecte OMEGA. Aquest era un conjunt de bases terrestres les quals emetien senyals de molt baixa freqüència, i la seva precisió es troava entre els 4.8 i els 8 kilòmetres. Tanmateix, era una infraestructura imprecisa, fràgil, i presentava dificultats per assolir la cobertura global pel seu elevat cost i limitat abast. Això va provocar que se cerqués un enfocament diferent. La inspiració va venir de Rússia:

La Unió Soviètica havia llençat a l'espai el Sputnik 1, el primer satèl·lit artificial de la història, el 4 d'octubre de 1957. Una vegada posat en òrbita, emetia senyals de ràdio d'aproximadament 20000 i 40000 MHz, i amb l'efecte Doppler d'aquests senyals segons el satèl·lit s'allunyava o aproximava, els estatunidencs van ser capaços de calcular la seva posició. Això va portar a la conclusió que seria possible fer servir un satèl·lit de posició coneguda per a localitzar qualsevol element a la terra, i com que es fa a l'espai, la cobertura global seria molt més senzilla d'implementar.

D'aquí va néixer el primer sistema de posicionament per satèl·lit; Transit. Es va realitzar el primer llançament amb el Transit 1A en setembre de 1959, tot i que no va estar exitós. Al

voltant del 1964 va ser operatiu per a la marina, i quatre anys més tard, va estar complet amb una constel·lació de vint-i-sis satèl·lits.

La primera versió d'aquest sistema era molt imprecisa, amb variacions de quilòmetres i visibilitat directa cada dues hores. A més, calia seguir el satèl·lit per almenys quinze minuts per obtenir una dada vàlida. Finalment, el projecte va resultar amb una precisió de 400 m.

Un dels avenços més rellevants després de Transit va estar el projecte Timation, també realitzat pels Estats Units. Aquest demostrava la viabilitat de col·locar rellotges de precisió als satèl·lits (conèixer el temps exacte és imprescindible per les aplicacions GPS), i juntament amb la invenció dels rellotges atòmics, va ser possible el desenvolupament d'una nova constel·lació.

Així és com, el 1973, es va crear el programa nomenat NAVSTAR-GPS (Navigation Satellite with Time And Ranging - Global Positioning System), que posteriorment es denominaria GPS. Aquest programa va ser declarat com a operatiu el 1993.

Era designat per a ús exclusivament militar, fet que va canviar per la tragèdia succeïda l'any 1983, quan la Unió Soviètica va abatre un avió comercial coreà amb 269 víctimes mortals com a resultat d'una desviació accidental del seu trajecte. Això va demostrar la importància de disposar d'un sistema de navegació accessible per a l'àmbit civil.

Això va derivar en què l'abril de 1995 el sistema declarés la seva capacitat operativa plena i s'obrí oficialment a l'ús civil.

Arran d'aquesta invenció, es van crear més sistemes des d'altres països i organitzacions. Principalment per evitar la dependència a la infraestructura estatunidenca.

1.3. Sistemes GNSS i les seves característiques

En aquest apartat s'explicaran les diferents constel·lacions que han sorgit a base de l'aparició del GPS. Es pot trobar una taula de les característiques de cada un a l'annex 1.

- **GPS:** la constel·lació estatunidenca, la primera a ser oberta a l'ús civil. És important considerar que GPS, malgrat el seu nom, és un sistema GNSS més, i no el terme que els engloba tots, ja que aquesta és una confusió freqüent.
- **GLONASS:** és un terme amb el mateix significat que GNSS, tot i que en rus. Es tracta del segon sistema a aparèixer, després del GPS. Va ser funcional el gener de 1996, sota el comandament de la Unió Soviètica, però ara és controlat per la Federació Russa.
- **Galileo:** constel·lació creada per la Unió Europea per tal d'assegurar la independència tecnològica amb altres sistemes GNSS. Tanmateix, són compatibles

amb alguns d'ells, com GLONASS i GPS. Va ser operatiu per primer cop el desembre de 2016, amb la meitat dels satèl·lits actuals.

- **BeiDou:** sistema de posicionament xinès format per tres constel·lacions diferents: BeiDou-1, BeiDou-2 i BeiDou-3. El primer va ser una prova amb només 3 satèl·lits.
- **QZSS (Quasi-Zenith Satellite System):** un sistema de posicionament regional del Japó. Encara no pot funcionar per si mateix, pel fet que només consta de quatre dispositius totals, però sí complementàriament amb altres infraestructures.

2. Concepte dels sistemes GNSS

Durant aquest apartat es farà una introducció als diferents termes que constitueixen la base per als sistemes de posicionament, com també els múltiples conceptes que apareixen.

2.1. Funcionament simplificat, triangulació bidimensional ideal

Les dades de les quals es disposa a l'hora d'establir una posició són:

- Les posicions exactes dels satèl·lits.
- El temps en què el senyal va ser emès.
- El temps en què el senyal va ser rebut.

En el fons, el temps d'emissió i d'arribada del senyal no són importants per si mateixos pel que fa al posicionament. No obstant això, ens permeten calcular una dada que sí que ho és, la distància entre l'emissor i el receptor, amb aquesta fórmula:

$$D = \Delta t * c, \Delta t = t_f - t_0$$

On Δt és el temps de viatge del senyal, c la velocitat de la llum ($3 \times 10^8 \text{ m/s}$), t_f el temps final o de recepció i t_0 el temps inicial o d'emissió.

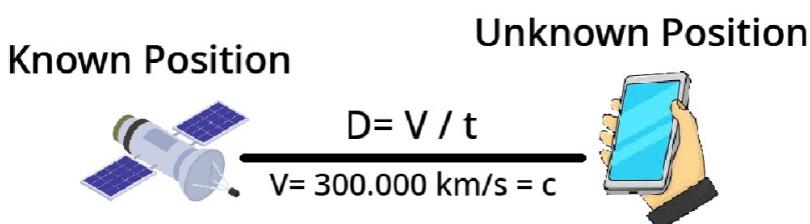
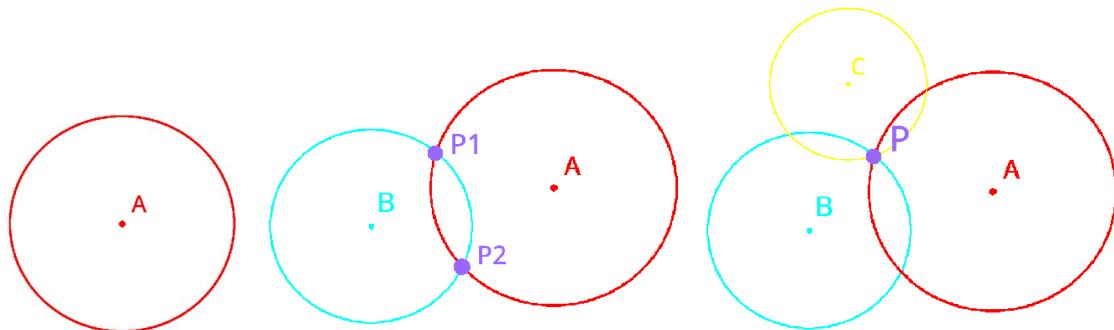


Fig. 1: Visualització del càlcul de la distància (m). Autoria pròpia amb el programa scratch.mit.edu.

Una forma de visualitzar el funcionament del sistema és d'una manera geomètrica en un pla bidimensional. En el cas en què es té una posició fixa coneguda, un punt, i una distància entre aquest punt i la posició que es tracta de trobar, el que resulta és una figura molt concreta. És una circumferència, on cada punt d'ella és una possible posició del receptor, i per tant hi ha infinites solucions (fig. 2).

Si, en canvi, es prenen dos satèl·lits, ara tenim dues circumferències diferents, i ambdues interseccionaran en dos punts diferents. En conseqüència, es passa de tenir infinites possibles solucions a només dues (fig. 3).

I si es pren un últim satèl·lit més, entre tots tres podran proporcionar una posició definitiva (fig. 4).



Figs. 2, 3, 4: Representació de la triangulació bidimensional. Autoria pròpia amb el programa scratch.mit.edu.

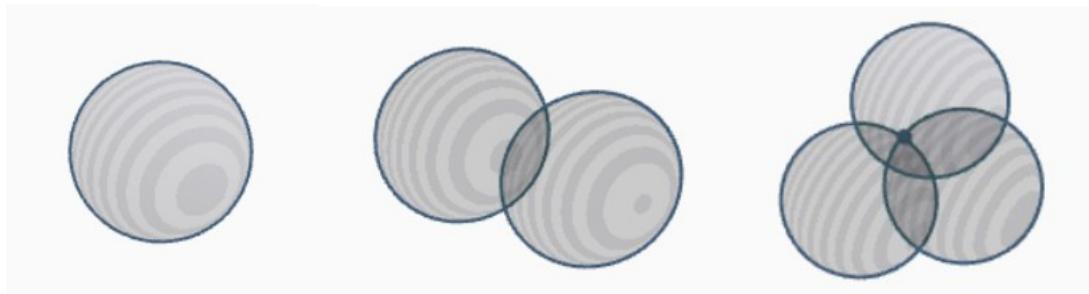
Aquest concepte és denominat triangulació, i és la base del funcionament de qualsevol sistema GNSS. Els satèl·lits no coneixen la posició de cap receptor, únicament envien un senyal amb la seva posició i temps exacte. I és el receptor que, en rebre múltiples senyals, calcula la seva posició.

2.2. Triangulació tridimensional ideal

L'aproximació prèvia pot ajudar a comprendre el funcionament darrere del sistema, però no és realista, ja que l'espai en què es troben tant els satèl·lits com el receptor és tridimensional. Tot i això, les dades de les quals es disposa no varien, per tant, cada satèl·lit representarà una esfera en comptes d'una circumferència, amb infinites possibles solucions (fig. 5).

Si, en canvi, s'utilitzen dos satèl·lits, el resultat tampoc és com en el cas bidimensional. Quan dues esferes interseccionen, la figura que es crea és similar a un anell (fig. 6).

En fer servir tres posicions fixes, en comptes d'obtenir una posició final com en el primer cas, el que s'obté són dos punts diferents (fig. 7).



Figs. 5, 6, 7: Representació de la triangulació tridimensional. Autoria pròpia amb el programa scratch.mit.edu.

Ara bé, tot i que en teoria caldria un quart satèl·lit per obtenir una solució geomètrica, com que els satèl·lits fan ús d'un sistema de coordenades centrat en la Terra, un dels punts sempre serà proper a la superfície terrestre, mentre que l'altre apareixerà fora de l'atmosfera, i per tant es pot descartar.

2.3. Triangulació bidimensional i tridimensional amb errors

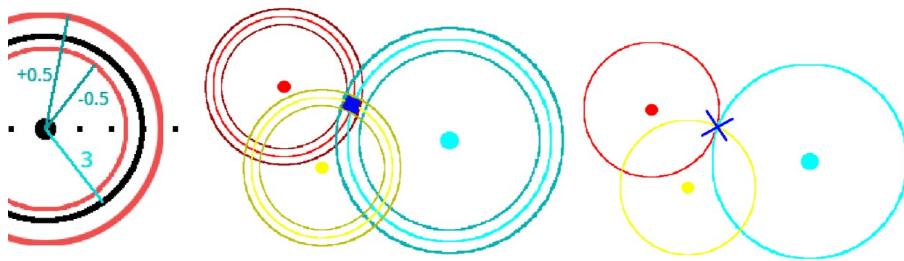
Ara bé, tots els exemples mostrats fins aleshores fan servir valors absoluts, amb una distància exacta i un resultat inequívoc. I malgrat que és l'objectiu idílic de tot sistema, i cada cop s'aproximen més, no és el cas real.

A les mesures reals sempre hi ha present un valor d'error, el qual pot provenir de múltiples fonts, i que acaba donant com a resultat una distància aproximada a la real, però en cap cas absoluta.

Aquest error es pot representar en última instància com la distància aproximada \pm valor d'error. Un exemple que ho representés en una circumferència seria aquest:

$$\text{radi} = 3 \pm 0,5 \text{ m} \text{ (fig. 8)}$$

Com es pot comprovar, en comptes d'una circumferència, ara hi ha una àrea definida per la circumferència més petita i la més gran. En conseqüència, en ajuntar-se amb dues posicions més, el resultat és una àrea (fig. 9).



Figs. 8, 9: Visualització de l'efecte de l'error a la triangulació. Autoria pròpria amb el programa scratch.mit.edu.

Fig. 10: Triangulació sense error. Autoria pròpria amb el programa scratch.mit.edu.

En un espai tridimensional, aquest concepte és molt més complex de representar, però el resultar final acaba sent un volum d'incertesa, dins del qual es troba la veritable posició.

3. Elements d'un sistema GNSS

Les constel·lacions de satèl·lits no són l'únic element que forma aquests complexos sistemes, ja que també són necessàries estacions terrestres per millorar la precisió, i per descomptat el receptor. En aquest apartat s'explicaran els segments que constitueixen aquesta infraestructura, descrivint les seves funcions principals. Finalment, es resumirà el funcionament global del sistema.

3.1. Segment espacial

El segment espacial està format per la constel·lació de satèl·lits que proporcionen la informació amb la qual després es calcularà la posició. Els elements que caracteritzen aquest segment són el nombre de satèl·lits, l'alçada de l'òrbita i la inclinació orbital.

3.1.1. Nombre de satèl·lits

El nombre de satèl·lits d'un sistema global operatiu sol estar entre els 20 i els 30. Tanmateix, no tots es troben en funcionament. Generalment, sempre hi ha 2 o 3 de reserva, i 1 o 2 en manteniment. Amb aquest mètode qualsevol organització pot garantir la cobertura global contínua, inclòs si fallen uns pocs satèl·lits.

En el cas del GPS, consta de 31 satèl·lits operatius (tot i que només requereix 24 per oferir cobertura global) i 5 en reserva.

Els satèl·lits addicionals milloren la precisió del sistema, és a dir, el fan redundant. El que és el mateix, disposa de més dispositius dels necessaris que reforcen la precisió i asseguren la continuïtat del sistema. Ara bé, la millora no és molt significativa.

3.1.2. Òrbites i els seus efectes

L'alçada de l'òrbita és un dels paràmetres més importants en qualsevol sistema satel·lital, no només els de posicionament. Afecta la cobertura, la precisió, la latència i la vida útil del satèl·lit. Per entendre-ho, és necessari introduir primer alguns conceptes bàsics sobre el moviment orbital.

L'òrbita pot ser el·líptica, amb el cos més gran situat a un dels focus, o circular, amb el cos al centre (que no deixa de ser un cas particular d'òrbita el·líptica). La velocitat que porta el satèl·lit per tal de mantenir-se en òrbita és major com més a prop del centre de massa. També, l'energia que requereix situar un satèl·lit augmenta molt amb l'alçada, tot i que no s'hagi d'invertir tanta a donar-li la velocitat necessària.

Per tot això, podria semblar que col·locar un satèl·lit més a prop de la Terra és més barat, ja que requereix molta menys energia per transportar-ho al lloc. Tanmateix, les òrbites altes (com les del GPS, a uns 20.200 km) ofereixen avantatges fonamentals:

- No hi ha fricció atmosfèrica. Els satèl·lits que sí que pateixen aquest efecte perdren velocitat de forma contínua, i necessiten recalibració constant per mantenir la seva òrbita, destinant energia per a propulsar-los. A més, tenen una vida útil més curta.
- Els senyals dels sistemes GNSS són afectats per l'atmosfera. Com més llunyana és la font d'una emissió, menys tangencial és l'emissió a l'atmosfera, i en conseqüència travessa menys part d'aquesta fins a arribar a la superfície.
- La cobertura, concepte que referència l'àrea que és capaç de cobrir cada satèl·lit, és molt major que en òrbites properes. Per tant, es redueix dràsticament el nombre de satèl·lits requerits per a assolir la cobertura global.

Contràriament, hi ha un problema inevitable que augmenta amb la distància, i és el temps que triga el missatge a arribar al receptor, o la latència. Com que aquest viatge a la velocitat de la llum, és quasi imperceptible en l'escala dels segons, però en serveis informàtics de transmissió d'informació pot ser un canvi vital. Especialment, si són bidireccionals (la informació viatge del satèl·lit a l'usuari, i a l'inrevés), com podria ser una connexió satel·lital a internet. La forma de calcular la latència és:

$$t = \frac{d}{c}$$

On d és la distància que ha de recórrer el senyal i c la velocitat de la llum, ≈ 300000 km/s

Aquest retard no és important als sistemes de posicionament en tant aquests calculen la diferència de temps d'emissió i de recepció. Però per a aplicacions com trucades, videotrucades, connexió a internet... seria un problema considerable.

Com que cada òrbita proporciona diferents avantatges i desavantatges, s'han designat diverses segons l'alçada. Aquestes tres són les generalment utilitzades per a aplicacions satel·litals:

- LEO (Low Earth Orbit): situada entre els 150 i els 2000 km. Proporciona una molt baixa latència perfecta per a comunicacions en temps real, al voltant de 22 ms. Presenta dos grans problemes: la necessitat de manteniment i propulsió constant pel fet de trobar-se a prop de l'atmosfera, una major congestió orbital i la necessitat de milers de satèl·lits per a assolir la cobertura global
- MEO (Medium Earth Orbit): situada entre els 2000 i 35000 km. Aquesta és l'òrbita a la qual pertanyen els sistemes GNSS. Un satèl·lit situat a 20000 quilòmetres de la superfície terrestre (com són els de GPS) experimenta una latència d'aproximadament 67,7 ms ($t = \frac{d}{c} = \frac{20000 * 10^3}{300000 * 10^3} = 0.067 \text{ s} = 67,7 \text{ ms}$).
- GEO (Geostationary Earth Orbit): aquests presenten uns avantatges molt específics. Se situen al voltant dels 35500 km d'alçada, i a aquesta alçada el període orbital¹ s'aproximen a les 24 hores. Per tant, es pot fer servir un satèl·lit que es mantingui en una posició fixa al cel. A més, amb un únic satèl·lit es pot proporcionar cobertura global. La contrapart és que requereix una major potència de senyal, i és molt més costós de posicionar que la resta de satèl·lits (tot i que no requerirà, possiblement, cap ajust orbital durant la seva vida útil, la qual serà més llarga). Aquestes característiques la fan perfecte per a aplicacions televisives o meteorològiques.

3.1.3. Inclinació orbital

La inclinació orbital quantifica què inclinat és un satèl·lit en relació amb l'equador. Si l'angle és 0 graus, el satèl·lit orbita únicament a l'altura de l'equador. Si, al contrari, és de 90 graus, llavors es tracta d'una òrbita polar, on el satèl·lit viatja de nord a sud i a l'inrevés

¹Període orbital: temps que triga el satèl·lit a donar una volta sencera al planeta.

passant pels pols. Però si es troba en qualsevol punt entre aquests dos, és una òrbita inclinada, la qual passa per tot el planeta, i si és un valor negatiu, llavors és una òrbita retrògrada².

Cada una d'aquestes òrbites tenen les seves característiques pròpies i aplicacions ideals, generalment combinant cobertura global, nombre de satèl·lits necessaris, visibilitat en zones habitades i geometria de recepció per al receptor.

3.1.4. Satèl·lits

En aquest subapartat s'explicarà breument l'evolució en tamany, pes i equipament dels satèl·lits des del seu inici. Es mostrerà tot element indispensable en un satèl·lit GNSS.

Amb l'avanç de la tecnologia, els satèl·lits han anat augmentant en pes, mida i complexitat, a causa de l'augment en instruments i tecnologies que incorporen.

- Primeres generacions (Transit, 1960–1970): d'aproximadament 1 m de diàmetre i uns centenars de kg, amb rellotges limitats i cobertura discontínua. Vida útil molt curta.
- Generació NAVSTAR-GPS (1978–1995): Pes mitjà d'entre 800 i 1000 kg una dimensió de 5 m. Introduïen rellotges atòmics, els transmissors L-band i control tèrmic i orbital.
- Satèl·lits moderns (GPS Block III, Galileo, BeiDou): Pes aproximat d'entre 1.5 i 2.5 t, dimensions grans per allotjar panells solars i antenes múltiples. Disposen de rellotges atòmics redundants, transmissió en múltiples bandes i tecnologia per resistir interferències i falles. La vida útil supera els 15 anys amb actualitzacions de programari estant en servei.

L'equipament de què disposen els satèl·lits actuals, generalment, es tracta del següent: un rellotge atòmic que brinda exactitud en la mesura del temps, transmissors de senyal que envien la informació adient a través de l'antena de radiació, i els sistemes que mantenen el satèl·lit amb energia, en l'òrbita correcta, i monitorejat en cas de qualsevol error intern.

²Òrbita retrògrada: de sentit de rotació contrari al de la Terra, en aquest cas.

3.2. Segment de control o terrestre

El segment terrestre està format per diferents tipus d'estacions terrestres, amb funcions específiques. L'objectiu global d'aquesta part és recollir dades del segment espacial i mantenir-lo actualitzat i operatiu, assegurant la precisió de les dades.

Tot i que aquí s'explicarà de què és compost, s'ha de tenir en compte que aquest sistema és propi de cada país o constel·lació per a la qual sigui dissenyada.

3.2.1. Components del segment de control

- Estacions de monitoreig: múltiples estacions independents monitorean el sistema assignat. La informació que acostuma a prendre importància són les dessincronitzacions dels rellotges atòmics, dades de l'òrbita (ephemerides) i falles del mateix satèl·lit.
- Estació de control: generalment hi ha una principal i una de reforç per sistema. Rep totes les dades de les estacions de monitoreig, i aplica les correccions corresponents. Envia correccions al rellotge atòmic d'un satèl·lit, o a la seva posició orbital. Són les encarregades de fer que una constel·lació sigui sempre funcional gestionant els inconvenients.
- Antenes de comandament i control: envien les correccions corresponents als satèl·lits. Són incorporades a les estacions esmentades anteriorment.

L'exemple més senzill per a un d'aquests sistemes és OCS (Operational Control Segment), que és dissenyat per a GPS. Actualment, disposa d'una estació de control (MCS, Master Control Station), 16 estacions de monitoreig i 11 antenes de comandament i control.

3.3. Segment d'usuari

Aquesta part és la que, finalment, vincula l'usuari amb tota la resta de sistemes GNSS. En última instància, són els receptors i processadors dels senyals GNSS. En aquest apartat s'exposarà els components principals que el componen i el seu funcionament. També es tractaran els possibles mètodes per obtenir un senyal i descodificar-lo, pel fet que en aquest subapartat apareix la metodologia SDR, i és un dels conceptes més importants d'aquest projecte.

Un receptor GPS és aquell que rep el senyal dels satèl·lits GPS, el descodifica, i extreu el PVT (Position, Velocity and Time).

3.3.1. Components principals

- Antena GNSS: hi ha diferents tipus d'antenes, cada una amb les seves prestacions. Les tres que acostumen a ser les més presents als receptors GNSS són l'antena de "patch o antena plana, l'antena helicoidal i l'antena xip o lineal (consultar annex 2).
- Hardware o software especialitzat: en el cas del hardware, es tracta dels processadors, els quals reben el senyal dels satèl·lits sincronitzant-se amb el seu rellotge. Són els que calculen el PVT. Tanmateix, no és necessari un processador especialitzat per descodificar un senyal GNSS, ja que es pot fer mitjançant SDR (Software Defined Radio). El seu nom significa literalment ràdio definida per software, i bàsicament permet el processament de senyals de diferents característiques a partir de programari.
- Rellotge del receptor: generalment, el receptor no té un rellotge, o no prou precís. És per això que no és un component essencial del receptor, pel fet que no és realista incorporar un rellotge atòmic a cada receptor. En canvi, el temps de recepció es soluciona com a una variable matemàtica més, tot i que això provoca que es requereixin 4 satèl·lits per a proporcionar un resultat. Aquesta variable mostra una de les funcionalitats més essencials dels sistemes de posicionament, ja que també proporcionen el temps actual amb extrema precisió.

3.4. Funcionament global dels sistemes GNSS

Amb tot el que s'ha vist, ja es pot definir el funcionament global de qualsevol sistema de posicionament satel·lital normatiu (fig. 11).

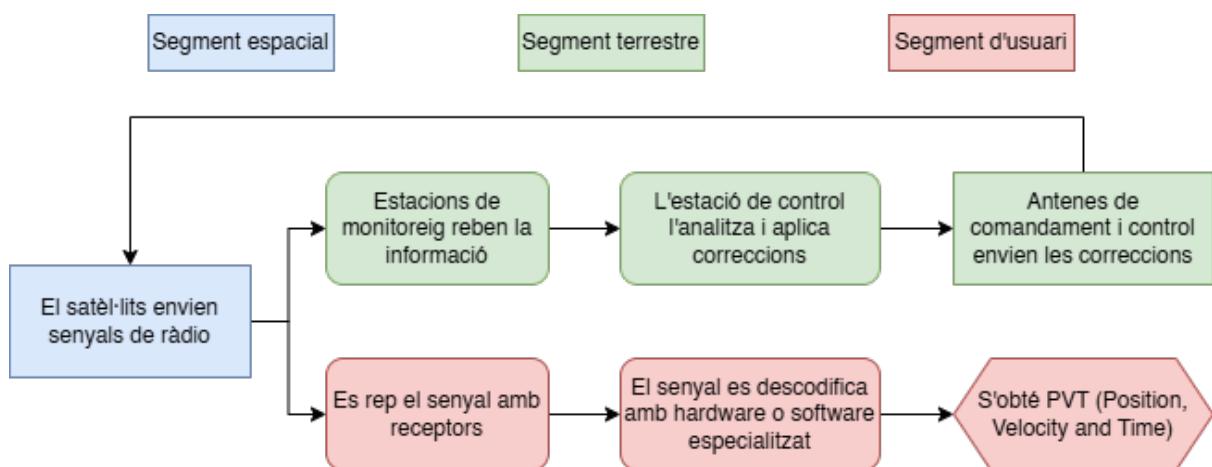


Fig. 11: Esquema del funcionament dels segments GNSS. Autoria pròpia amb el programa [draw.io](#).

4. Senyals, fonts d'error i càlcul de la posició

En aquest apartat s'explicarà els diferents senyals que s'utilitzen en GPS, com també els errors que poden afectar-les i com es solucionen. Es farà una introducció a les funcions matemàtiques que ens permeten obtenir una posició, les variables que es busquen resoldre i totes les dades que apareixen:

4.1. Senyals GPS

Els senyals transmesos pels satèl·lits GNSS es troben dins la banda L (entre aproximadament 1,1 i 1,6 GHz). Són ones electromagnètiques modulades, i combinen tres components per poder transmetre la informació necessària:

- Ona portadora (carrier): ona sinusoidal d'una alta freqüència que transporta la informació i permet el càlcul de la fase del senyal.
- Codis PRN (*Pseudo-Random Noise sequences, o ranging codes*): seqüències binàries pseudoaleatòries que permeten mesurar el temps de viatge del senyal. Com que cada satèl·lit emet una pròpia, els permet identificar. Un dels avantatges d'aquest element és que permet a diversos satèl·lits comunicar-se amb el receptor amb la mateixa freqüència sense deixar de ser identifiables.
- Missatge de navegació (Navigation data): codi binari que conté les efemèrides (posició i velocitat del satèl·lit), paràmetres de rellotge, l'*Almanac* (posicionament aproximat de tota la constel·lació) i informació complementària com l'estat de salut del satèl·lit. És el que porta tota la informació necessària per a l'obtenció de la posició o la sincronització temporal.

La gran majoria d'ones d'aquest sistema tenen aquestes propietats en comú:

- Freqüència i longitud d'ona: les bandes L1 (1575,42 MHz), L2 (1227,60 MHz) i L5 (1176,45 MHz) són les més utilitzades; la longitud d'ona associada és d'uns 19 a 25cm.
- Nivell de senyal (C/N0): relació entre la potència del senyal i el soroll (dB-Hz). Amb valors típics de 25–50 dB-Hz; per sota de 30, la recepció és feble.
- Polarització circular dreta (RHCP): assegura una millor recepció independentment de l'orientació de l'antena. Aquesta característica referència el sentit del gir que efectua l'ona sobre si mateixa.
- Potència molt baixa: en arribar a la superfície terrestre, els senyals tenen potències entre -125 dBm i -130 dBm, per això requereixen antenes i receptors molt sensibles,

i no és possible obtenir un posicionament fiable a dintre d'edificis o túnels.

- Sincronització temporal: cada senyal incorpora marques de temps molt precises provinents del rellotge atòmic del satèl·lit. Per aquest motiu els sistemes GNSS es poden emprar per a aplicacions que requereixin un temps molt estable i precís, i no només per a posicionament.
- Determinació d'observables: a partir del senyal, el receptor pot obtenir:
 - Pseudorang (pseudorange): distància apparent derivada del temps de propagació.
 - Fase de portadora: mesura de fase amb precisió centimètrica.
 - Desplaçament Doppler: desplaçament de l'energia, la freqüència, del senyal. Apareix a causa del desplaçament del satèl·lit amb la perspectiva del receptor.

4.2. Fonts d'error i limitacions actual del sistema

4.2.1. Errors ionosfèrics

La ionosfera és la regió atmosfèrica entre els 60 i els 1000 quilòmetres d'alçada (tot i que la major part dels efectes tenen lloc especialment entre els 200 i els 400 km). Rep el seu nom perquè presenta gran quantitat d'ions carregats, els quals poden interferir amb la propagació d'ones electromagnètiques com les dels sistemes GNSS.

En situacions normatives, l'error varia entre els 15 metres de dia, i els 3 de nit. Però pot arribar a un màxim de 50 m depenent de l'angle en què el senyal penetra l'atmosfera.

4.2.2. Errors troposfèrics

Per troposfera s'entén la capa inferior de l'atmosfera, entre els 0 i els 12 km d'alçada. La capa afecta el pseudorang per la seva humitat i pels canvis de pressió atmosfèrica.

Aquest efecte no depèn de la freqüència del senyal, a diferència del primer. En canvi, afecta la seva modulació i fase. Això és perquè la troposfera és un medi no dispersiu.

L'error que provoca la troposfera en casos normals es calcula entre els 2.5 m quan el cos orbitant es troba al zenit respecte al receptor, i fins a 30 m quan es troba sobre l'horitzó. Per a evitar aquest efecte, s'utilitzen models troposfèrics.

4.2.3. Errors d'efemèrides

Les efemèrides són les dades orbitals dels satèl·lits, i com que la posició inicial de

l'emissor es pren com a dada a l'hora de calcular la posició, errors en aquestes es tradueixen directament en errors de posicionament.

Els satèl·lits reben correccions periòdiques que minimitzen aquest error. Per tant, aquest error només apareixerà amb un problema dins de la mateixa estructura del sistema.

4.2.4. Errors de multipath

Aquest efecte es dona quan els senyals d'un mateix satèl·lit arriba al receptor amb diferents temps i angles. El seu motiu és que aquests es poden reflectir en superfícies properes, com edificis o el fins i tot el terra (per tant afecta especialment entorns urbans).

L'error total pot ascendir als 30 m en entorns urbans, i a 3 m en els rurals.

En la seva resolució es troba el motiu pel qual els sistemes GNSS empren diferents bandes (L1, L2 i L5), ja que en rebre senyals de més d'una, el receptor pot diferenciar les reals de les duplicades. Fins i tot, proporciona més pes a les bandes més resistents al multipath, com l'L5, quan detecta un entorn on aquest efecte pren més importància.

4.2.5. Error de rellotge (satèl·lital i del receptor)

Els satèl·lits estan equipats amb rellotges atòmics, els quals són extremadament precisos, i a penes es desfasen per ells mateixos. Tot i això, el constant monitoreig des del segment terrestre corregeix qualsevol dessincronització.

No obstant això, la relativitat d'Einstein demostra que els rellotges dels satèl·lits acaben sumant $38 \mu\text{s}$ d'error per dia, el que equival a 11,4 km ($\frac{300 \text{ m}}{1 \mu\text{s}} * 38 \mu\text{s} = 11400 \text{ m}$).

Això es deu a aquests dos efectes relativistes:

- Relativitat especial: explica que el temps d'un cos amb una diferència de velocitat molt major a la d'un altre, s'alenteix des del punt de vista del segon cos. Aplicat als satèl·lits, des de la Terra es considera que la rapidesa amb què passa el temps per al dispositiu és menor a la terrestre.
- Relativitat general: explica que la gravetat deforma l'espai-temps, amb el resultat de què en presència d'un camp gravitatori major, el temps passa més lent. Per tant, un satèl·lit que és més lluny del centre de la Terra que algú a la seva superfície, experimentarà un temps més ràpid.

En el cas dels satèl·lits, la primera provoca $-7 \mu\text{s/dia}$, i la segona, $+45 \mu\text{s/dia}$. Amb la

suma dels dos s'obté la xifra total d'error per aquests efectes: +38 µs/dia.

Per tal de mitigar completament aquest efecte, en tant es coneixen les fórmules, els rellotges atòmics s'ajusten directament a la Terra per tal de contrarestar-lo

4.2.6. Geometria dels satèl·lits

La precisió del posicionament final no depèn només de la qualitat del senyal, sinó també de com els satèl·lits estan redistribuïts al cel respecte del receptor, el que es coneix com a geometria dels satèl·lits. Per una banda, una geometria favorable, amb bona separació entre els satèl·lits redueix els errors, però per l'altra una desfavorable els amplifica.

DOP (Dilution Of Precision) és el valor que quantifica quant afecta la precisió de la posició calculada, sent el valor 0 el valor idílic. Es classifiquen segons el que mesuren: com HDOP que quantifica la precisió horitzontal, VDOP la vertical, GDOP la geomètrica global...

4.3. Resolució matemàtica de la posició i el temps

Com ja ha estat esmentat, la base per a obtenir la posició és el càlcul de la diferència de temps o el temps que ha trigat el senyal des de la seva emissió a la seva recepció. Es pot obtenir una distància amb aquesta fórmula:

$$D=c\Delta t \quad (1)$$

On c és la velocitat de la llum (3×10^8 m/s) i Δt la diferència de temps ($\Delta t=t_f-t_0$)

Però el valor obtingut no és la distància geomètrica real, perquè tots els errors esmentats anteriorment afecten el senyal i alteren el resultat. És per això que a aquesta mesura se l'anomena pseudodistància.

Per mesurar la distància vàlida en qualsevol sistema GNSS, s'aplica aquesta fórmula:

$$R_p = p + c(dt_r - dt^s) + T + \alpha_f STEC + K_{p,r} - K_p^S + M_p + \varepsilon_p \quad (2)$$

On:

- R_p és la pseudodistància calculada
- p és la distància geomètrica real entre el satèl·lit i el receptor
- dt_r i dt^s són els errors dels rellotges del satèl·lit i del receptor.
- T és el retard troposfèric
- $\alpha_f STEC$ és un terme sobre el retard ionosfèric, que depèn de la freqüència
- $K_{p,r}$ i K_p^S són els retards instrumentals tant del satèl·lit com del receptor
- M_p és l'efecte multipath
- ε_p és el soroll del receptor

Per tant, tal com es pot apreciar a l'equació (2), la pseudodistància equival la distància geomètrica més els errors. La següent fórmula és per a definir la p d'un satèl·lit específic:

$$p_i = \sqrt{\square}$$

On p_i és la distància real entre receptor i el satèl·lit "i", X_i, Y_i, Z_i és la posició coneguda del satèl·lit "i". I X, Y, Z són les posicions desconegudes del receptor.

Com que al final tenim quatre (com a mínim) equacions no lineals amb quatre incògnites, no es poden resoldre directament. Per això, s'utilitza una aproximació linealitzant³ les equacions al voltant d'una posició inicial del receptor i s'itera⁴ fins que les solucions convergeixen a valors estables per a les coordenades i el rellotge.

³Linearització: procés pel qual una equació no lineal s'aproxima per una equació lineal en un entorn proper a una solució inicial. S'utilitza perquè els sistemes de distàncies entre receptor i satèl·lits són no lineals gràcies a la presència de l'arrel quadrada en la fórmula de la distància.

⁴Iteracions: es refereix a repetir el càcul diverses vegades, ajustant progressivament les coordenades del receptor, fins que els canvis entre dues solucions consecutives siguin mínims.

5. Mètodes per a millorar la precisió

Com s'ha pogut comprovar, els sistemes GNSS no són perfectes. Tot i que errors de 3 a 5 metres siguin admissibles per a la majoria d'aplicacions civils, hi ha certs casos en què una precisió de menys d'un metre és essencial, com en la cartografia o la navegació autònoma. Per això existeix el mètode d'ampliació d'aquestes infraestructures, que proporciona millores al sistema sense canviar la seva essència.

Els més comuns són:

- DGNSS (Differential Global Navigation Satellite System): una estació terrestre de coordenades conegeudes compara la posició obtinguda mitjançant un sistema GNSS amb la real. D'aquesta diferència extreu correccions que s'envien als receptors propers. Pot arribar a disminuir entre 1 i 3 metres l'error total, però només pot oferir una cobertura local, generalment aplicada a estructures aeroportuàries.
- SBAS (Satellite-Based Augmentation System): aquest sistema de correcció utilitza satèl·lits geoestacionaris, que ofereixen una cobertura molt major a la del DGNSS, però amb una precisió menor. Els seus errors oscil·len entre 1 i 2 m, i les seves aplicacions solen ser de navegació civil i aviació.
- RTK (Real Time Kinematics): el mètode que més precisió aporta, amb errors centimètrics. Compara en temps real, via ràdio o internet, les dades obtingudes per una base amb les del receptor mòbil. Això el fa extremadament precís, però paral·lelament, també molt dependent en la qualitat de la comunicació entre els dos elements.
- PPP (Precise Point Positioning): similar a RTK, però aquest sistema implementa models precisos sobre les òrbites dels satèl·lits i els seus rellotges, fent innecessària una estació base. La contrapart és que requereix postprocessar les dades durant un temps que pot variar entre els pocs minuts, i unes poques hores. També té una gran precisió.

6. Característiques del SDR

6.1. Concepte d'SDR

SDR (Software Defined Radio) és un concepte que es va introduir als anys noranta, i que va marcar un abans i un després en les telecomunicacions sense fil. Fins al moment, tot el processament d'un senyal s'havia de fer analògicament amb maquinari dedicat (com desmoduladors, filtratges, descodificadors...) i, per tant, si es volia canviar algun aspecte del processament de l'ona, o passar a una altra diferent, s'havia de canviar l'equipament físic.

Aquesta tecnologia, com indica el seu nom, va demostrar que es podia realitzar el processament d'un senyal a partir de programari, el que és el mateix, per a fer un canvi en el processament, aquest es podia fer al codi en comptes de l'equipament.

6.2. Avantatges i desavantatges de la implementació de la tecnologia SDR

La tecnologia en qüestió va suposar una gran innovació en el camp de la telecomunicació, ja que aviat es va descobrir les seves grans qualitats, com:

- Flexibilitat: un mateix dispositiu, tant receptor com processador, pot gestionar múltiples protocols i bandes de freqüència.
- Actualizable: permet incorporar noves funcions simplement amb programari.
- Cost reduït en entorns experimentals i d'educació, pel fet que evita múltiples dispositius específics.
- Facilitat per a simulacions i experiments amb diferents tecnologies de comunicació.
- Tanmateix, presenta certes deficiències comparada amb receptors específics;
- Necessitat de processadors potents per a senyals amb amplada de banda elevat o amb múltiples canals oberts.
- Pot aparèixer latència i pèrdua de precisió si el software no està optimitzat.
- Limitacions en la gestió de senyals molt complexos o d'alta freqüència, on el hardware dedicat encara és més eficient.

Com es pot comprovar, la seva utilitat depèn molt del context en què s'utilitza. Però, tot i això, ha sigut especialment útil en camps com la simulació del processament dels senyals, o l'educatiu, que permet l'experimentació sense haver de manejar elements físics.

6.3. Funcionament bàsic

Els receptors GNSS tradicionals disposen de circuits dedicats específicament als senyals per a les quals estan dissenyats. Per tant, la seva tasca només consisteix a rebre el senyal amb una antena, i processar-lo amb els seus circuits.

Un dispositiu SDR, en canvi, no pot dur a terme aquesta tasca. Generalment, el seu funcionament consisteix únicament a rebre el senyal amb una antena, la qual sol ser externa, perquè el punt d'aquests sistemes és poder rebre tota mena de senyals. I processar-la per tal que un ordinador, placa, o el processador adient pugui interpretar-la. Això mostra una altra característica d'aquesta tecnologia, perquè és obsoleta sense cap programa que la faci servir. Per això, aquelles aplicacions que són molt específiques com podrien ser els receptors GNSS, els receptors de ràdio comercials... tots aquests incorporin hardware especialitzat en comptes d'això.

No obstant això, la funcionalitat d'un sistema SDR serà determinat per l'antena que es faci servir, i l'amplada de banda i freqüència que sigui capaç de processar el dispositiu. El rang de freqüència depèn del model, alguns econòmics com el RTL-SDR cobreixen entre els 24 i 1766 MHz, mentre que d'altres professionals arriben a desenes de megahertz.

MARC PRÀCTIC

7. Metodologia

Per tal d'assolir tots els objectius, s'ha seguit una metodologia basada en la recerca teòrica i l'experimentació. Tot problema trobat s'ha provat d'aïllar a base de proves repetides i meticuloses. Un cop reunides suficients dades, s'han fet hipòtesis del possible error i els seus orígens, i s'ha intentat descartar fonts d'error o solucionar-los si no es podien descartar.

El desenvolupament de l'aplicació, en canvi, ha estat definit per un procés iteratiu. S'ha començat des d'una versió mínima (la interfície gràfica), i s'han anat afegint funcionalitats amb noves versions fins a arribar a la definitiva.

8. Equipament del projecte (hardware)

En aquest apartat s'explicarà detalladament tot el maquinari emprat durant aquest treball. Es definirà el concepte que representa cada element, s'explicaran breument de què es componen i la seva funcionalitat, i s'exploraran les seves limitacions, possibles aplicacions i avantatges d'utilitzar-lo en comptes d'un altre aparell.

RTL-SDR V3:

És un dispositiu receptor de baix cost que fa servir el xip Realtek RTL2832U¹ per tal de transformar senyals analògics en digitals. Tal com el seu nom indica, es basa en la metodologia SDR (Software Defined Radio). Aquest concepte representa el fet de poder emprar programari per processar i analitzar senyals, en comptes de necessitar elements físics específics per a cada tipus d'aplicació.



Fig. 12: Dispositiu RTL-SDR V3. Autoria pròpia.

El RTL-SDR té tantes aplicacions com tipus de senyals pot rebre dins de les seves limitacions. Això vol dir que pot processar senyals de ràdios comercials (FM, AM), televisió digital terrestre (DVB-T), bandes d'aviació (com ADS-B), GNSS (L1)...

El seu principal avantatge i característica, com ja ha estat mencionat, és el fet que fa servir la metodologia SDR. A més, és relativament econòmic ja que amb un dispositiu entre 30 i 60 euros reemplaça molts altres receptors dedicats. Alhora, és d'una mida reduïda.

Encara més, pot adaptar-se a les noves tecnologies de telecomunicacions sense necessitat de modificar els seus components, en tant que sempre que es mantinguin dins els límits de recepció del RTL-SDR, només caldrà actualitzar els programes o l'antena emprats.

No obstant això, també presenta alguns desavantatges, tal com la seva limitada sensibilitat (sobretot sense ús d'un LNA extern), freqüència i amplada de banda. Tot i que és comprensible pel seu preu i multifuncionalitat. Encara més, per si sol no pot fer res, en tant que necessita tant una antena com un programa que processi els senyals.

Prosseguint, és un dispositiu que requereix software complex i molta configuració per tal de funcionar de manera eficient. Per tant, no és tan immediat i senzill d'utilitzar com pot ser un receptor específic, que proporciona el senyal totalment processat.

Les seves limitacions són la resolució de 8 bits, que es tradueix en menys precisió que receptors professionals, l'amplada de banda limitada, i especialment la menor estabilitat i sensibilitat en freqüències majors a 1766 MHz, a causa que no cobreix senyals com el wifi.

Antena activa GPS-001:

Una antena activa és una antena amb LNA (Low Noise Amplifier) integrat, en conseqüència, necessiten rebre corrent per funcionar correctament.

La diferència entre la passiva i l'activa és bàsicament que la primera transmet el senyal tal com el rep, i la segona l'amplifica injectant corrent. És per això que és crucial fer servir l'antena activa per a aquesta aplicació, a causa que el senyal dels satèl·lits és molt débil.

El model usat en aquest cas és DollaTek - Antena activa GPS impermeable de 28 dB, 3 metros, puerto SMA para NEO-6M U-BLOX



Fig. 13: Antena activa GPS. Autoria pròpia.

Les seves aplicacions són molt limitades, ja que només és dissenyat per a rebre senyals GPS. Per tant, no té sentit realitzar cap comparació amb altres tecnologies. Tampoc és raonable comparar-la amb altres tipus d'antenes perquè tenen funcions diferents.

Mòdul GPS NEO-7M:

El KS0319 és un mòdul GPS basat en el xip u-blox NEO-7M, dissenyat per obtenir posicions a partir de senyals dels satèl·lits GPS. Inclou una antena ceràmica integrada i connector SMA per connectar una antena externa millorant la recepció. Envia les dades en format NMEA estàndard.



Fig. 14: Mòdul NEO-7M. Autoria pròpia.

Únicament es pot emprar per obtenir la posició amb GPS o per a sincronització precisa gràcies al seu port PPS (pulsacions per segon). Però la seva versatilitat es troba en el fet que es pot utilitzar amb un microcontrolador, sense requerir programes per funcionar.

Presenta grans avantatges enfront del RTL-SDR, a raó que és molt més senzill de fer servir. A més, per si sol pot funcionar perfectament, en tant que no requereix cap antena externa, però alhora admet l'ús d'una per millorar la precisió. A més a més, és realment econòmic.

Però té certes limitacions, com per exemple la baixa precisió i sensibilitat en ser un mòdul econòmic, molt inferior a receptors professionals. A més, el seu funcionament en interiors és quasi nul sense antena externa, i el temps per poder establir la primera posició (TTFF, time-to-first-fix) és major que molts altres sistemes.

9. Programari del projecte i entorns de programació

En aquest apartat s'exposarà en detall tot el programari que s'usarà al llarg del projecte. Indicant la seva funcionalitat i característiques més rellevants. No tots els programes s'utilitzaran en la mostra final, sinó que o bé serveixen per configuracions rellevants, o per proves i depuració.

Connexionat del RTL-SDR:

Per fer ús d'aquest dispositiu, no n'hi ha prou amb connectar-lo a l'ordinador, ja que aquest el mostra com a dispositiu desconegut. És imprescindible instal·lar un

driver específic per al RTL-SDR. Per això es farà servir Zadig:

És un software de llicència pública. Serveix per accedir a dispositius que l'ordinador no reconeix de manera automàtica, assignant-los un driver específic predeterminat perquè sigui reconegut immediatament. El que reconeix és el xip que incorpora el dispositiu connectat, per tant, no ens mostrarà RTL-SDR, sinó RTL2832U.

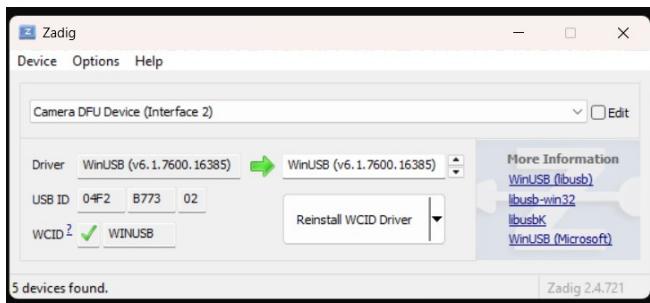


Fig. 15: Imatge del programa Zadig. Autoria pròpia.

També és necessari un mètode per a injectar corrent al dispositiu, i això es fa amb el bias tee. Com ja ha estat comentat, el bias tee és un circuit que permet el pas del corrent a través de dispositius connectats (com pot ser una antena activa). En aquest cas cal descarregar-ho independentment. El que s'acabarà fent servir d'aquest programa són dos executables nomenats bias_tee_on i bias_tee_off.

És molt important activar el bias tee només quan el dispositiu amb l'antena activa són connectats. I retirar el dispositiu una vegada apagat el bias tee. Això és per evitar danys als aparells.

GNSS-SDRlib i GNSS-SDR:

GNSS-SDR (Global Navigation Satellite Systems - Software Defined Radio) és un receptor de senyals de satèl·lit (GPS, Galileo, GLONASS, BeiDou...) totalment definit per software. Funciona amb qualsevol receptor SDR compatible (com el RTL-SDR v3), i pot llegir les dades en temps real o postprocessar-les, retornant posició, velocitat, nombre de satèl·lits als quals fa seguiment... però existeixen dos programes, independents entre ells.

El programa GNSS-SDR ha estat desenvolupat pel Centre Tecnològic de Telecomunicacions de Catalunya, i la seva primera versió va ser publicada el març de 2011. Es tracta d'un programari de codi obert escrit en C++. Té gran versatilitat

pel que fa a la configuració, ja que s'utilitza amb arxius de configuració (.conf) amb les seves comandes específiques. I és una eina orientada a ús professional, pel fet que no disposa d'interfície gràfica.

En canvi, l'eina que es provarà d'usar en un principi és GNSS-SDRlib, una aplicació desenvolupada únicament per un programador, Taro Suzuki. Es tracta d'un codi, com l'anterior, de processament de senyals GNSS en temps real o gravades per proporcionar informació i una posició. Aquest sí que disposa d'interfície d'usuari, i incorpora una funcionalitat (GNSS Radar) que permet millorar la selecció dels satèl·lits, perquè mostra en temps real els més propers.

Ambdós codis són de codi obert (que es pot consultar i editar, si es pot compartir o comercialitzar depèn de la llicència en cada cas) i es poden trobar a GitHub. I tots dos són compatibles amb el rtl-sdr.

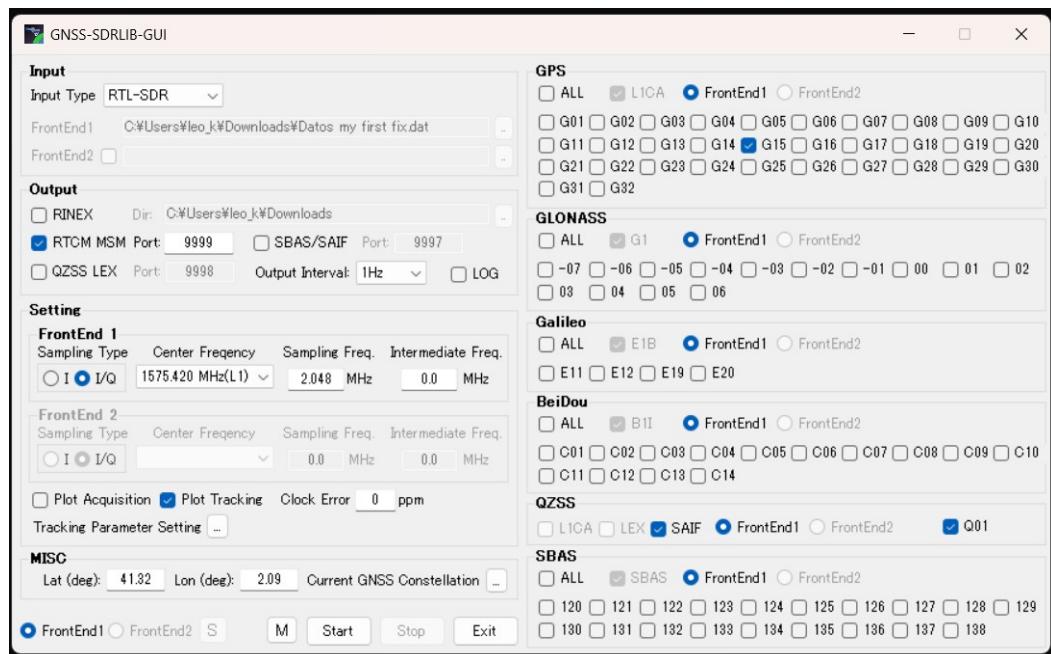


Fig. 16: Imatge del programa GNSS-SDRLIB-GUI de GNSS-SDRLIB. Autoria pròpia.

RTKlib:

En aquest projecte s'utilitzaran eines d'un programa de codi obert, RTKlib, que a més és compatible amb GNSS-SDRlib. Aquest codi aplica el concepte RTK (Real-Time Kinematics).

Les aplicacions més rellevants que incorpora el programa són:

- RTKnavi → Processament de les dades de GNSS-SDRlib en temps real
- RTKpost → Postprocessat de les dades, per exemple, en arxius RINEX.
- RTKplot → Gràfiques sobre errors, posicionament...

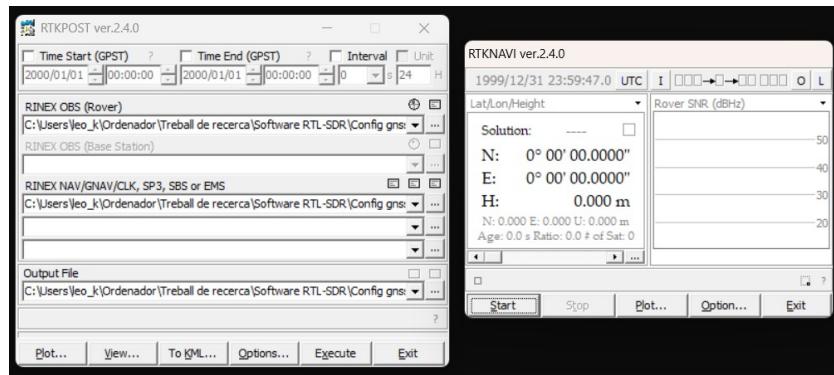


Fig. 17: Imatge del programa RTKPOST i RTKNAVI de RTKLIB. Autoria pròpia.

Python amb VSCode:

Tota la part de codi del projecte, a excepció del codi del microcontrolador, s'executarà amb Python. Aquest és un llenguatge de programació molt intuïtiu, que s'ha escollit per diverses raons:

- Es caracteritza per una sintaxi molt clara i lleigible
- Disposa d'innumerables llibreries que permeten crear diferents tipus d'aplicacions
- Ja s'ha treballat molt anteriorment, com també amb Tkinter (una llibreria per a desenvolupar la interfície gràfica), i per tant és un entorn familiar.

El desenvolupament del codi es durà a terme a l'entorn de *Visual Studio Code* (VSCode), que és un editor de codi obert. És amb el que s'havia treballat el llenguatge amb anterioritat.

Arduino IDE i STEAMakersBlocks:

Aquest és el llenguatge de programació que s'ha emprat per a controlar la sortida del mòdul GPS i enviar-la a l'ordinador.

STEAMakersBlocks és una plataforma que permet programar plaques a partir de blocs, sense necessitat de codi escrit. L'avantatge que presenta és que després permet traduir aquest codi a codi escrit, cosa que permet aprendre (com ha estat aquest cas) codi nou o l'estructura d'aquest, igual que compartir-lo i depurar-lo.

D'altra banda, Arduino IDE és una interfície que et permet escriure codi i pujar-lo a la placa, directament.

És per tot això que s'ha emprat STEAMakersBlocks en primera instància per a comprovar el correcte funcionament del mòdul GPS i alhora familiaritzar-se amb el mateix llenguatge i les funcions específiques.

No obstant això, aquesta no havia estat la proposta inicial. Ja que per la familiaritat al llenguatge de Python, s'havia provat de desenvolupar aquesta aplicació amb MicroPython, una versió adaptada per a microcontroladors amb la que l'ESP32 és compatible. Fent servir Thonny, similar a Arduino IDE però per a aquest llenguatge de programació. Tanmateix, veient la dificultat que implicava (a causa que Arduino és molt més pràtic per a microcontroladors, amb molta més compatibilitat per a llibreries, com la de GPS) i la falta de documentació, es va acabar abandonant la idea.

10. Posicionament inicial

Com que tota la pràctica depèn de la posició en temps real, primer es cerca aconseguir-la de la forma més senzilla, per tal de conèixer el procediment. A partir d'aquí es podrà automatitzar aquest procés, i assegurar la base del projecte.

10.1. Configuració de hardware i software inicial, error “0x00007b”

Primer, s'ha de descarregar tots els programes per a configurar el RTL-SDR i processar els senyals captats per aquest. Les aplicacions descarregades inicialment i els seus enllaços són:

- Zadig: <https://zadig.akeo.ie/>
- bias tee: <https://github.com/rtlsdrblog/rtl-sdr/releases/tag/v1.1>
- GNSS-SDRlib: <https://github.com/taroz/GNSS-SDRLIB>
- RTKlib: <https://github.com/rtklibexplorer/RTKLIB/releases/tag/v2.5.0>

Com ja s'ha explicat a l'apartat de metodologia, Zadig s'utilitza per tal que l'ordinador reconegui el dispositiu RTL-SDR, perquè, en cas contrari, el marca com a dispositiu desconegut. La instal·lació i execució ha anat com cal esperar.

Una vegada instal·lat el controlador, s'ha executat amb normalitat (consultar annex 3)

GNSS-SDRlib s'ha instal·lat des de GitHub. En un inici tot semblava correcte, però en iniciar GNSS-SDRgui (la interfície d'usuari) ha aparegut un error de Windows.

L'error ha estat “**0x00007b**”, que generalment fa referència a un error amb les dependències⁵, ja sigui per:

- Incompatibilitat entre mòduls de dependències: que una llibreria (dependència, .dll) era feta per a sistemes de 32 bits, i l'ordinador són en base a 64 bits, o a l'inrevés.
- Combinació de llibreries de 32 i 64 bits al mateix programa.
- Llibreria corrupta o inexistent

Per trobar l'error es va fer servir Dependency Walker, un programa que, tal com el seu nom indica, simula un executable (.exe) i recorre totes les dependències d'aquest notificant possibles errors. Després, et proporciona un llistat d'aquests.

Una vegada analitzada l'execució de GNSS-SDRgui, va mostrar l'absència de múltiples llibreries, entre elles algunes que s'havien provat d'inserir manualment prèviament.

Es va trobar que totes aquestes llibreries formaven part del redistribuible **Microsoft Visual C++ Redistributable for Visual Studio 2012**⁶, el qual es va instal·lar.

Havent reiniciat l'ordinador, GNSS-SDRgui es va iniciar amb normalitat, mostrant la interfície.

RTKlib, per la seva banda, no ha mostrat problemes en la seva instal·lació ni execució.

10.2. Primera execució GNSS-SDR + RTKnavi

Una vegada establert tot el sistema, es van posar en marxa els preparatius per obtenir el primer posicionament.

Es va connectar el RTL-SDR a l'ordinador, amb l'antena activa connectada per SMA. El programa del bias tee es va executar per terminal, mostrant el missatge corresponent. L'antena es trobava en interior, al costat d'una finestra, que hauria de ser suficient per a obtenir una posició.

Però una vegada encès GNSS-SDRgui i RTKnavi, ambdós connectats entre ells,

5Dependència o llibreria: conjunt de recursos reutilitzables que els programes poden importar per evitar redefinir funcionalitats.

6Un redistribuible és un conjunt de dependències que molts programes utilitzen, fins al punt en què és més factible instal·lar aquestes al propi ordinador, que a cada programa.

va aparèixer un altre error. Aquest no era explícit, perquè els programes mostraven un funcionament normatiu. Sinó que RTKnavi mostrava “timeout” al cap de pocs segons. Es va assumir que seria un problema de sincronització, perquè el missatge del segon programa possiblement indicava que no arribaven dades, o no de la manera correcta.

Es va provar amb diferents configuracions, totes amb el mateix resultat.

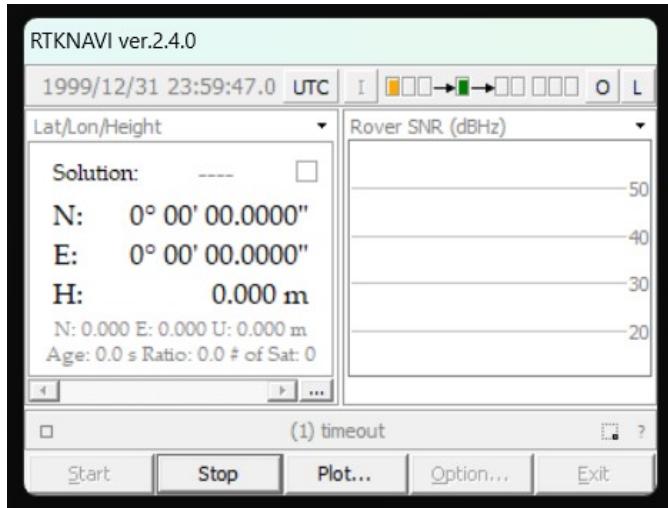


Fig. 18: Imatge de RTKNAVI amb connexió establerta però sense rebre dades. Autoria pròpia.

10.3. Aïllament de l'error i hipòtesis

En un punt, es va decidir postprocessar la informació amb RTKpost.

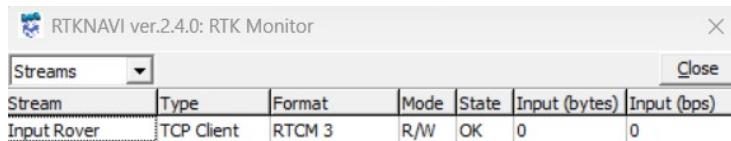
Com ha estat explicat a l'apartat de metodologia, RTKpost és un programa de postprocessament de dades que treballa conjuntament amb GNSS-SDRLIB. Pot fer ús de diversos formats, tot i que per conveniència s'ha decidit utilitzar arxius RINEX⁷ (Receiver Independent Exchange Format).

Per poder realitzar aquest processament, primer s'han d'obtenir els arxius RINEX. Per tal de fer-ho, només s'ha de canviar la configuració de la sortida d'informació (*output*) de GNSS-SDRLib, assignant-li una carpeta on guardar les dades.

L'execució ha semblat correcta i s'ha mantingut per dos minuts, però en comprovar els fitxers ambdós eren buits. Per tant, el problema no es troba a RTKlib o la seva connexió amb GNSS-SDRLib, sinó que es troba al segon programa, perquè **no genera dades útils**. Això també s'ha pogut comprovar des de la terminal de

⁷Els arxius RINEX són dades en forma de text que permeten intercanviar informació sobre sistemes GNSS independentment del receptor usat. Es componen per dos arxius complementaris, un d'observacions (.obs) i un de navegació (.nav), el primer amb informació sobre dades generals amb les quals es calcula la posició, i el segon amb dades sobre les efemèrides i correccions del rellotge dels satèl·lits.

RTKnavi, als paràmetres del flux de dades, on ha mostrat 0 bytes d'entrada en totes les proves realitzades fins al moment.



Streams							Close
Stream	Type	Format	Mode	State	Input (bytes)	Input (bps)	
Input Rover	TCP Client	RTCM 3	R/W	OK	0	0	

Fig. 19: Monitor de RTKNAVI amb les dades del flux de dades. Autoria pròpia.

El programa GNSS-SDRlib ha de mostrar per terminal, en un màxim de dos minuts, una posició fixada. A més, també ha de mostrar el moment en què comença a descodificar les dades dels satèl·lits per obtenir les efemèrides o el temps del satèl·lit. No obstant això, independentment del temps d'execució, la configuració del rtl-sdr, la qualitat dels senyals GNSS i la correcte aplicació del bias tee, no mostra en cap moment dades processades (consultar annex 4 per a veure una mostra de la terminal d'aquesta execució incompleta).

Les dades que s'observen són del mateix senyal. En canvi, les dades que haurien d'aparèixer al cap d'uns segons haurien de mostrar més informació (exemple a l'annex 5).

Per tota aquesta evidència, és indiscutible que GNSS-SDRlib no és capaç de proporcionar un fix de posició. Pot haver-hi diversos motius que provoquin aquest problema. Els més probables que s'han trobat són:

- Problema de configuració de la interfície gràfica: descartat, pel fet que s'ha provat amb diferents configuracions, totes amb el mateix resultat. També s'ha prosseguit amb rigor els exemples dels vídeos “Decoding GPS using an RTL SDR Receiver” i “GPS Decoding And Plotting Using a RTL SDR And GPS Antenna | GNSS-SDR | RTKLIB”, que es poden trobar al llistat de fonts.
- Exposició massa breu per a l'obtenció de dades vàlides, o mal senyal: és pràcticament impossible, perquè s'han fet proves de diverses llargades, arribant als deu minuts. A més, gran part d'elles s'han realitzat a un terrat, sense cap edifici que sobrepassi l'alçada de l'antena, i per tant amb plena exposició al cel. Les condicions meteorològiques han variat d'una prova a una altra, tot i que no es considera un possible factor condicionant.
- Flux de dades irregular o inestable: aquest no es pot descartar. Els sistemes GNSS requereixen una constància i precisió excepcionals, i un senyal inconstant podria provocar un error als càlculs que derivés en què mai es proporcionés una resposta.
- Problema a l'antena o al rtl-sdr (al hardware): tot i que possible, és poc probable. El

material s'ha tractat amb cura rigorosa, i no presenta cap senyal visual de maltractament. Tot i això, no s'ha pogut comprovar el funcionament de l'antena fins ara. El rtl-sdr s'ha testejat amb diverses aplicacions de ràdio, amb el programa SDRSharp i una antena passiva bipolar, amb un funcionament normatiu.

Com que després d'un gran número de proves i diferents configuracions, que s'han portat a terme durant un mes, no s'ha arribat a cap resultat vàlid, s'ha hagut de canviar de programa. S'ha fet tot el que s'ha pogut per tal d'aïllar i solucionar l'error, però el funcionament del programa GNSS-SDRlib més enllà de la interfície és desconegut, i comprendre'l s'allunya massa de l'objectiu i context d'aquest projecte.

11. Prova de l'entorn WSL2 (Windows Subsystem for Linux)

11.1. Justificació del canvi d'entorn i explicació del programa utilitzat

Finalment, s'ha optat per provar d'usar GNSS-SDR. A la metodologia s'ha explicat que existeixen dos codis amb un nom similar, GNSS-SDRLIB i l'aplicació esmentada. La primera era la que s'ha provat d'utilitzar prèviament, però amb els errors que ha presentat, es provarà de fer un canvi en l'eina emprada.

També s'ha de canviar de sistema operatiu, o almenys fer servir un subsistema (sistema que s'executa sobre un sistema operatiu, en aquest cas, interpreta les ordres com a Linux i les “tradueix” a Windows) com ha estat el cas. És imprescindible perquè de tota la documentació existent per al programa, a penes es fa referència a Windows, en canvi, Linux és present a tota font consultada. Així doncs, s'utilitzarà WSL2, que és un subsistema que permet executar l'entorn complet de Linux dins de Windows. A més, pot representar una millora ja que GNSS-SDRlib ha donat problemes a Windows des de la seva instal·lació.

Una de les opcions més prometedores contemplades va ser l'ús d'una VM⁸ (Virtual Machine), però s'ha descartat per la dificultat d'accés als ports des de l'entorn i temor a una sincronització i capacitat de processament insuficient.

⁸Un entorn o màquina virtual simula un ordinador complet dins del propi, on normalment ambdós sistemes es troben aïllats l'un de l'altre.

11.2. Instal·lació de l'entorn, configuració de ports i instal·lació de GNSS-SDR

WSL2 no s'ha de descarregar com un programa més, sinó que ve incorporat dins de Windows, però s'ha d'habilitar. Per fer això, es pot anar a l'apartat “Activar o Desactivar les Característiques de Windows”, però fer-ho per terminal (fent servir PowerShell amb privilegis d'administrador) és més directe (consultar annex 6).

Una vegada habilitades les característiques de WSL2, només cal instal·lar la distribució desitjada. En aquest cas s'ha descarregat Ubuntu, sense cap raó específica. De nou, es pot fer des de la terminal, tot i que si es preferís també existeix l'opció de descarregar-ho en línia o a la Microsoft Store.

Un cop realitzat aquest procés, Ubuntu ja està instal·lat a l'ordinador, i es pot accedir directament a través del cercador d'aplicacions o amb la comanda “wsl.exe -d Ubuntu”.

Però encara es necessita un programa més abans de passar a instal·lar GNSS-SDR. Com que WSL2 no pot accedir als ports de l'ordinador encara, és a dir, que no pot reconèixer el rtl-sdr. Perquè ho pugui fer s'ha d'instal·lar usbipd-win. Aquest procés es pot realitzar des de la terminal de Windows PowerShell, amb privilegis d'administrador. Si no es fa des d'aquest rol, demanarà que es doni consentiment al programa com a tal, com ha passat en aquest cas (consultar annex 7).

Un cop fet això, s'ha d'instal·lar des de la terminal de Linux aquests paquets, que incorporen dependències per a sistemes com WSL2, i la comanda usbipd entre d'altres:

```
sudo apt update  
sudo apt install linux-tools-$(uname -r) linux-cloud-tools-$(uname -r)
```

Per tal de comprovar si ha funcionat, primer es llisten els dispositius connectats actualment a l'ordinador. Després, s'identifica el dispositiu rtl-sdr (normalment sota el nom de “RTL2838UHIDIR”, el seu xip) i es guarda el “busid”, l'identificador que Windows assigna al xip (consultar annex 8).

En aquesta ocasió el rtl-sdr es troba sota l'identificador 2-1. Ara només cal introduir les últimes comandes, indicant el “busid” trobat anteriorment (consultar

annex 9).

```
PS C:\WINDOWS\system32> usbpd attach --wsl --busid 2-5
>>
usbpd: info: Using WSL distribution 'Ubuntu-22.04' to attach; the device will be available in all WSL 2 distributions.
usbpd: info: Detected networking mode 'net'.
usbpd: info: Using IP address [REDACTED] to reach the host.
PS C:\WINDOWS\system32> =
```

Fig. 20: Fragment de CMD on es connecta el dispositiu rtl-sdr a usbpd. Autoria pròpia.

11.3. Prova de posicionament en temps real

Una vegada instal·lat tot l'entorn i les funcions necessàries, ja només quedava la prova de posicionament. Per dur-la a terme, els preparatius van estar els mateixos d'abans, tot i que sumant la connexió del dispositiu dins l'entorn WSL2. Una vegada fet això, es va procedir amb la prova.

Aquesta es va realitzar, com ha estat mencionat, amb el programa GNSS-SDR, amb un arxiu de configuració el qual ha estat redactat amb l'ajuda de diversos exemples trobats a internet. Tot i que no és la mateixa, es pot trobar l'última versió d'aquest arxiu (Configuració lectura dades rtl-sdr 10 min.conf) utilitzada per fer proves a la carpeta amb material penjada al Drive (enllaç al final del document). Aquesta no fa exactament la mateixa funció, ja que analitza una gravació en comptes de dades en temps real, però la resta de funcions són similars.

El resultat va estar inesperat, perquè, per una banda, va funcionar correctament, mostrant per la mateixa terminal totes les dades dels satèl·lits, però per l'altra, mostrava exactament la mateixa informació que amb l'altre programa, GNSS-SDRlib. Per tant, aquest mètode tampoc proporciona la posició, tot i semblar executar-se amb normalitat.

11.4. Gravació de dades i posterior anàlisi

Com que el mètode inicialment emprat no havia donat resultat, es va provar amb aquesta estratègia. No s'esperava cap classe de millora respecte a la situació inicial, però suposava certs avantatges per a la depuració del codi: es faria una gravació de llarga durada a més d'una normal, per a poder descartar la influència de la llargada de l'exposició als senyals com a font d'error de manera definitiva. A més, es podria

accedir als senyals de manera immediata sense necessitat de buscar exposició a cel obert, i així facilitar la prova de noves configuracions.

Es va provar amb diferents arxius de configuració, i aquells que sí que semblaven analitzar la gravació, no mostraven més que la duració d'aquesta. Es pot trobar un exemple (Processament del senyal 10 min a WSL2.mp4) a la carpeta amb recursos de Drive. Dins de la carpeta no es proporciona la mateixa gravació de dades, ja que en teoria incorpora la posició en què va ser presa. Els arxius RINEX sí que es fa perquè són buits.

```
Tracking of GPS L1 C/A signal started on channel 1 for satellite GPS PRN 15 (Block IIR-M)
Current receiver time: 1 min 51 s
Loss of lock in channel 7!
```

Fig. 21: Fragment de la descodificació de dades amb GNSS-SDR. Autoria pròpia.

11.5. Prova amb rtl_test

Aquesta funció executable a WSL2 realitza una gravació en temps real amb el dispositiu rtl-sdr, però no analitza les dades sinó el funcionament del dispositiu i si hi ha pèrdua d'informació. Aquesta funció ajudaria a descartar una de les principals sospites; que la inestable sincronització de Windows amb el subsistema era la causant de la falta de posicionament.

Hi ha un exemple present (Comprovació RTL_TEST.mp4) a la carpeta del Drive.

El resultat de la prova va estar que només hi havia una pèrdua mínima de 20 bytes en iniciar el programa, la qual és normal perquè s'està sincronitzant. Per tant, aquesta possible font d'error també queda descartada.

```
Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
lost at least 20 bytes
^CSignal caught, exiting!

User cancel, exiting...
Samples per million lost (minimum): 0
leo_k@PClen:~$
```

Fig. 22: Comprovació exitosa amb rtl-test. Autoria pròpia.

11.6. Seguiment del tutorial “My First Fix” de gnss-sdr.org

Aquest tutorial es pot trobar a l'enllaç [gnss-sdr.org /my-first-fix/](http://gnss-sdr.org/my-first-fix/)

Com que fins ara no hi ha hagut cap resultat tangible, s'ha decidit realitzar de manera ràpida aquest tutorial per descartar el mal funcionament del programa. Consisteix en el processament d'un arxiu amb gravacions de dades GNSS.

Segons la pàgina web referenciada anteriorment, els missatges mostrats a l'annex 10 representen un processament correcte.

El resultat ha estat l'esperat, amb aquest mateix log, i també l'aparició de diferents arxius, entre ells els d'observació i navegació RINEX. Per tant, el programa és funcional i ha estat instal·lat correctament.

```
New GPS NAV message received in channel 3: subframe 5 from satellite GPS PRN 17 (Block IIR-M) with CN0=45.03 dB-Hz
Position at 2013-Apr-04 06:24:44.000000 UTC using 5 observations is Lat = 41.274854 [deg], Long = 1.987693 [deg], Height = 75.02 [m]
Velocity: East: 0.46 [m/s], North: 0.63 [m/s], Up = -0.40 [m/s]
Position at 2013-Apr-04 06:24:44.500000 UTC using 5 observations is Lat = 41.274858 [deg], Long = 1.987703 [deg], Height = 82.95 [m]
Velocity: East: -0.28 [m/s], North: 0.72 [m/s], Up = 1.02 [m/s]
Current receiver time: 1 min 39 s
```

Fig. 23: Log de la terminal amb l'execució exitosa del tutorial "My First Fix". Autoria pròpia.

11.7. Conclusió i tancament de l'experimentació amb SDR

Amb totes les dades recollides fins ara, no es pot extreure cap possible solució. El que s'ha observat ha estat que el hardware és funcional, igual que el software. Però hi ha almenys un error present en tots els processos que constitueixen la recepció i decodificació de les dades GNSS que no és possible identificar. Ja s'han descartat els més plausibles, com una instal·lació incorrecta, una mala sincronització, un senyal de mala qualitat o amb poc temps de gravació... i, per tant, no és possible solucionar aquest problema.

Cal recalcar que aquesta tecnologia és molt tècnica, i que l'objectiu d'aquest treball és entendre el procés general que realitza cada part i posar-ho en pràctica. Però entendre l'estructura dels programes o els possibles errors que els afecten no formen part d'aquest. Així mateix, no és realista pretendre fer-ho, a causa de la forta limitació que representa el nivell tècnic i el temps disponible.

Com a últim argument, és important comprendre que aquest punt del treball és l'inicial, ja que l'objectiu principal de la part pràctica és el desenvolupament d'una aplicació de geofencing, i l'ús del rtl-sdr només és per obtenir la posició en temps real, la base més important de l'aplicació.

Per totes aquestes raons, no queda altra opció que deixar de banda la idea

d'implementar el rtl-sdr, així com la tecnologia sdr, en aquest projecte. Aquest canvi no modifica l'objectiu del projecte, sinó que el fa viable dins dels límits de temps i recursos disponibles.

La millor alternativa que s'ha considerat ha sigut l'ús d'un mòdul GPS amb un microcontrolador ESP32 STEAMakers. Es preveia utilitzar-lo com a comparació amb el rtl-sdr, per veure si la precisió canviava, en cas que hi hagués temps, però finalment serà la font del posicionament en temps real. A més, això permetrà realitzar l'aplicació, i si s'aconseguís fer funcionar el rtl-sdr, només s'hauria de canviar l'entrada de dades, però l'aplicació romandria igual.

12. Posicionament amb el mòdul GPS NEO-7M

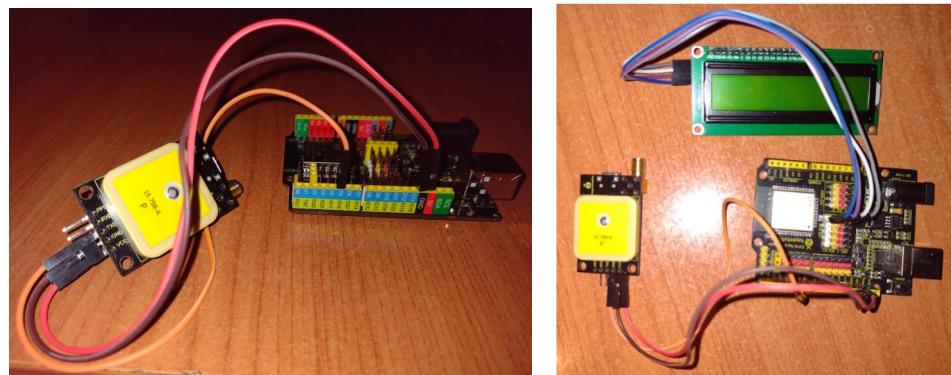
En aquest apartat s'explicarà com s'ha desenvolupat el programa del controlador que s'encarregarà d'obtenir la posició en temps real i enviar-la a l'ordinador. Es tractarà el connexionat entre els elements i la placa, així com les diferents versions del programa.

12.1. Connexionat del mòdul amb el microcontrolador

El mòdul consta de 5 pins, però només es faran servir 3:

- VCC: entrada de voltatge, es connectarà a la sortida de voltatge de la placa, ja sigui 3.3V o 5V.
- GND: terra, directament en contacte amb GND del controlador
- TXD: sortida de dades. S'envia la informació en format NMEA a través d'aquest pin, que es connecta a l'entrada RXD (pin D5) de l'ESP32, perquè aquest rep les dades.

A més, s'ha fet ús d'un display LCD 2x16 per a mostrar informació. S'ha incorporat a la placa com mostra la figura X



Figs. 24, 25: Mostra del connexionat del material emprat. Autoria pròpia.

12.2. Prova de funcionament amb STEAMakersBlocks

Una vegada establert tot el connexionat, es va passar directament a la prova de funcionament. Es va dur a terme directament a l'aplicació STEAMakersBlocks, per la simplicitat que representa ajuntar quatre blocs per a fer funcionar el mòdul.

Aquest va ser el codi que es va fer:

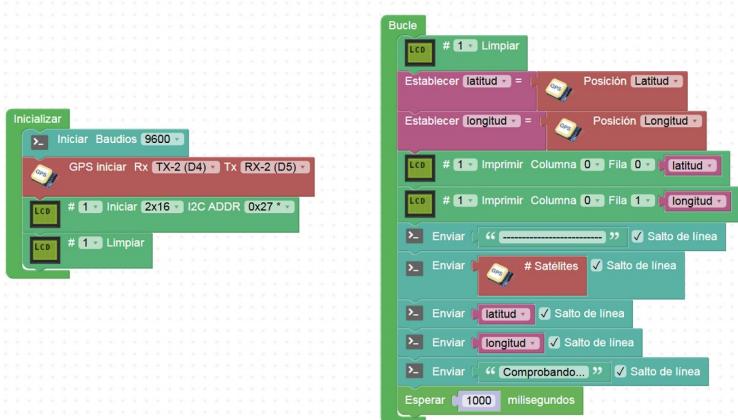


Fig. 26: Codi de prova de funcionament del mòdul GPS. Autoria pròpia amb l'aplicació STEAMakersBlocks

El programa va funcionar a la primera. En aquest punt es va provar de comparar el funcionament del mòdul independentment, o amb l'antena connectada (disposa d'un port SMA). Tot i que no hi ha dades, la diferència era evident, ja que en el cas de l'antena connectada el temps des que s'encenia fins que proporcionava la primera posició canvia radicalment (especialment en interior), i la precisió també era molt superior.

12.3. Codi final amb Arduino IDE, comunicació pel port o per Bluetooth

Com que era necessària una complexitat més elevada, es va optar per fer servir Arduino IDE per tal de desenvolupar el codi que proporcionés la posició a l'ordinador, i pujar-ho a la placa. S'ha fet partint del codi de STEAMakersBlocks passat a codi escrit com a base.

Primer es va realitzar mitjançant una comunicació bàsica per cable, però ràpidament es van considerar els grans avantatges que proporcionaria una comunicació per Bluetooth, i conseqüentment es va implementar. El codi final es troba a la carpeta penjada a Drive sota el nom de ESP32_main.ino.

El programa incorpora dues llibreries, que es poden incloure des de Arduino IDE:

- LiquidCrystal_I2C: controla el display LCD 2 (rows) x 16 (columns)
- TinyGPSPlus: extreure informació com latitud, longitud, hora... de les dades NMEA del mòdul GPS.

Envia missatges en format JSON per Bluetooth d'aquest estil:

```
{"ts": 6, "estat": "SEARCHING", "lat": 0.0, "lon": 0.0, "alt": 0.0, "vel_kmh": 0.0, "sats": -1, "hdop": -1.0}
```

On es pot veure l'estat del posicionament, la latitud, longitud i altitud, la velocitat instantània i l'hdop (horizontal dilution of precision). L'estat pot ser “SEARCHING”, “UNSURE” o “FIXED”.

13. Desenvolupament de l'aplicació; idea inicial i selecció del mapa a utilitzar

En aquest apartat s'explicarà a fons l'aplicació que es vol desenvolupar. S'explicarà l'objectiu a assolir, les característiques que es preveu que tindrà, i com es vol fer. A més, s'explicaran totes les opcions considerades a l'hora d'integrar un mapa amb l'aplicació.

13.1. Idea inicial

Com ja s'ha dit algunes vegades, l'objectiu final d'aquest projecte és el desenvolupament d'una aplicació de geofencing. Però, a què fa referència això?

El geofencing és una eina que té pràcticament infinites aplicacions, però en aquest cas no es vol posar el focus en cap d'elles. Contràriament, es vol donar pes al mateix concepte, ser capaç de recrear-ho, i crear una aplicació completa, amb les funcionalitats que pugui necessitar tan intuïtiva com sigui possible.

El que es pretén és fer aquesta aplicació amb Python, amb la llibreria **tkinter**, que permet crear finestres, botons, camps d'entrada i altres objectes interactius de forma senzilla, i és ideal per al desenvolupament d'interfícies intuïtives per a programadors sense experiència al front-end. Les característiques que definiran aquesta aplicació seran:

- Modular: no és viable construir res en un únic document. Es crearà un codi amb lògica pròpia per a cada element diferenciat de l'aplicació, per tal que el projecte sigui fàcil de mantenir i actualitzar, que sigui escalable. L'exemple més visual és separar tot el que representarà la lògica de l'aplicació de la construcció de la interfície.
- Interfície independent de la connexió Bluetooth: que si l'aplicació no rep dades, no es quedi penjada.
- Tolerant a errors: que notifiqui de tot error que apareix
- Preparada davant d'entrades incorrectes: que sàpiga identificar quan una informació, sigui introduïda pel mateix programa o per l'usuari, és impossible (exemple: una àrea de zero a dos punts)

Es pot veure un diagrama de com és el funcionament global de l'aplicació a l'annex 11.

14. Desenvolupament de l'aplicació; construcció de l'UI

Per a la construcció de l'UI (User Interface) es va decidir dividir-la en dues parts. L'esquerre incorporaria tota la part de gestió de les àrees, mentre que la dreta mostraria el mapa.

Aquests són els elements que es van incorporar al marc esquerre:

- Nom de l'àrea: una caixa de text que mostra o et permet editar el nom de l'àrea.
- Punts de les àrees: quadre on introduir els punts que constitueixen una àrea.
- Llista d'àrees: una llista que mostra totes les àrees que s'han anat creant. Les guarda dins d'un arxiu JSON que carrega en iniciar-se
- Botons: funcionen a “modes”. Si la interfície està normal, mostren el d'esborrar, afegir i editar una àrea. En fer clic en esborrar passen a tornar endarrere, o esborrar-

ho tot. En canvi, mentreafegeixes o edites una àrea et permet cancel·lar l'acció o salvar els canvis.

Per la part dreta, només hi ha el mapa, i al fons el botó “Start” que inicia l'aplicació de geofencing.

El mapa es farà a partir de la llibreria **tkintermapview**, la qual va ser escollida després de provar diverses llibreries i configuracions, com folium, webbrowser, o webview. Finalment es va trobar aquesta, que com el seu nom indica, és absolutament compatible amb tkinter. Mentre es cercava la millor llibreria, es va veure el que possiblement serà un dels majors problemes que apareixerà en el desenvolupament d'aquesta aplicació, i és la gestió de fils d'execució. Aquest concepte s'explicarà més endavant.

15. Desenvolupament de l'aplicació: evolució de les versions

En aquest apartat es posarà en ordre els mòduls que s'han anat creant. Com que una mateixa aplicació consta, quasi cada vegada, de diferents mòduls, es posaran conjuntament a l'inici de l'apartat. S'explicarà en cada cas els canvis que s'han realitzat i les característiques més rellevants.

15.1. Versió 1; UI inicial i gestió d'àrees

- `geofencing_ui_V1.py`

La primera versió, en què es va crear l'UI amb l'estructura ideada prèviament. Únicament incorporava tota la gestió d'àrees, amb les seves funcions d'afegir, esborrar, editar àrees. L'edició es realitzava mitjançant la caixa de text, on s'introduïen punts seguint aquesta estructura:

`latitud1,longitud1;latitud2,longitud2;... → 4.53523,6.65433;5.54224,4.234562;...`

Era una edició tosca i lenta, i es va veure clar la necessitat de l'edició a través del mapa. No obstant això, la lògica dels botons va funcionar de manera ideal, i no es preveu canviar el seu funcionament en les pròximes millores.

Paral·lelament, aquest programa és definit dins del mateix document, i no és una manera viable de treballar-lo. Incorpora elements de lògica i interfície al mateix lloc.

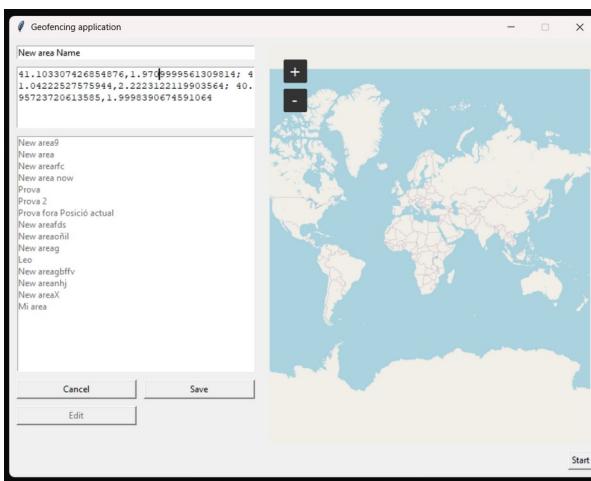


Fig. 27: Disseny inicial de l'aplicació de geofencing amb tkiter. Autoria pròpia.

15.2. Versió 1.2; reestructuració del codi

- geofencing_ui_V2.py
- geofencing_logic_V1.py

Pràcticament, la mateixa versió que abans, però amb la lògica separada a un mòdul independent, i definida com a classe.

15.3. Versió 2; visualització d'àrees

- geofencing_ui_V2.py
- geofencing_logic_V2.py
- is_inside_area_function.py

S'ha implementat la visualització de les àrees en el mapa i la lògica (simulada) de geofencing amb la llibreria **shapely**, que crea i fa operacions amb figures geomètriques.

Aquest és l'exemple ideal de la utilitat de la separació de lògiques; com en aquest cas l'UI quasi no s'ha hagut de modificar, el codi roman intacte.

S'ha identificat un problema menor; les àrees es defineixen segons l'ordre de la

llista de punts que les constitueixen, el que pot donar lloc a estructures incorrectes. Tanmateix, encara hi ha problemes seriosos com la validació de les dades d'entrada (no acceptar un nom repetit, una àrea de menys de tres punts...) o funcionalitats a implementar.

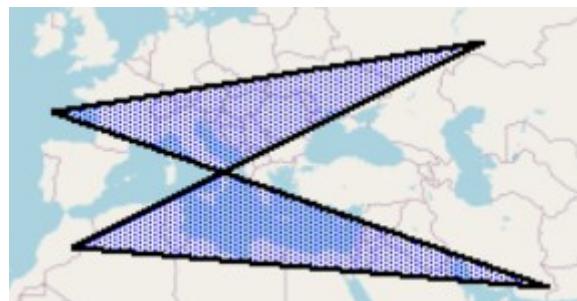


Fig. 28: Visualització d'un àrea amb punts sense ordenar. Autoria pròpia.

15.4. Versió 3; posicionament en temps real i comunicació

Bluetooth

- geofencing_ui_V3.py
- geofencing_logic_V3.py
- is_inside_area_function.py
- geofencing_read_bt.py

Codi complet amb posicionament en temps real via Bluetooth. Del mòdul de lògica només ha canviat la funció de geofencing, que prèviament utilitzava una posició simulada però ara està implementada..

Encara queda molt a incorporar i millorar. Especialment, la gestió de la connexió Bluetooth, ja que s'ha de reiniciar l'aplicació quan aquesta es perd.

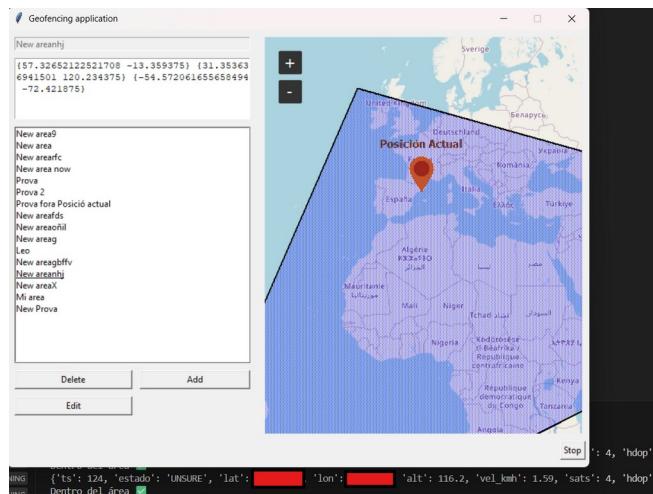


Fig. 29: Mostra de l'aplicació funcionant paral·lelament amb el receptor Bluetooth. Autoria pròpia.

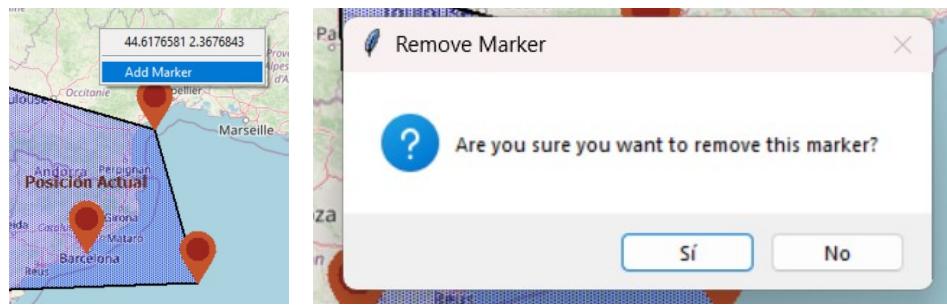
15.5. Versió 4; edició intuïtiva d'àrees

- geofencing_ui_V4.py
- geofencing_logic_V4.py
- is_inside_area_function_2.py
- geofencing_read_bt.py

Implementació de l'edició intuïtiva a través del mapa interactiu, en comptes de per la terminal. Actualment, els dos mètodes conviuen simultàniament, encara està per decidir si les àrees només es podran editar amb el mapa, o si la terminal s'actualitzarà amb les dades introduïdes al mapa i a l'inrevés, i per tant es podran emprar ambdós sistemes.

L'edició a partir del mapa funciona d'aquesta manera: amb el clic dretà sobre ell et mostra l'opció d'afegir un marcador, i si cliques amb l'esquerra a sobre d'un, et permet esborrar-lo.

Al mòdul amb la funció de geofencing (is_inside_area) se l'ha afegit una funció per ordenar els punts segons distància al centre geomètric entre tots els punts. Abans les figures es creaven a partir de l'ordre d'indexació dels punts, el que donava lloc a figures estranyes.



Figs. 30, 31: Mostra de l'edició a través del mapa. Autoria pròpia.

15.6. Versió 4.2; millora del sistema de connexió Bluetooth

- geofencing_ui_V4.py
- geofencing_logic_V4.py
- is_inside_area_function_2.py
- geofencing_read_bt_2.py

Modificació del codi de Bluetooth. Ara gestiona reconnexió, estats... també incorpora una parada automàtica si no s'aconsegueix connectar-se passat un nombre determinat d'intents.

La reconnexió funciona a partir d'un sistema de temps. Si fa cert temps que no es rep un missatge (timeout), torna a provar de connectar-se, ja que possiblement estava "escoltant" un canal que ja no era obert.

Es pot comprovar tot el funcionament d'aquesta funcionalitat a l'annex 12.

Com era previst, la gestió dels fils d'execució està resultant un repte, però s'ha pogut portar endavant amb la llibreria **Threading**.

15.7. Versió 4.3; gestió de la configuració i registre de logs

- geofencing_ui_V4.py
- geofencing_logic_V4.py
- is_inside_area_function_2.py
- geofencing_read_bt_2.py
- config_manager.py
- debug_logger.py

Com es pot comprovar pel nom dels nous mòduls, s'han implementat dos elements nous:

- Gestió de configuració: tots els valors incorporats al codi a la força, com el temps d'espera del Bluetooth abans de llençar un "timeout" o el nom de l'arxiu on es guarden les àrees es guarden a un arxiu a part. Cada document accedeix a aquest document per a importar les dades que necessita.
- Log d'esdeveniments: es guarden logs a un document per a facilitar el seguiment i la depuració de l'aplicació. Són jerarquitzats, amb la seva classe entre claudàtors a l'inici del missatge, i poden ser: info, warning, error o Bluetooth. Mostren la data si no s'ha registrat res aquell dia, i cada missatge mostra el temps exacte en què es va emetre.

Tot i que el nom no ha canviat, el codi tant de l'UI com de la seva lògica sí que ha canviat.

15.8. Versió 5; Versió final

- geofencing_ui_V5.py
- geofencing_logic_V5.py
- is_inside_area_function_2.py
- geofencing_read_bt_2.py
- config_manager.py
- debug_logger_2.py

En aquesta última modificació s'han aplicat canvis radicals. Primer de tot, a l'UI, ja que s'han incorporat elements nous a ambdós marcs (dretà i esquerrà), i s'ha afegit un marc nou (fons) amb encara més:

- Marc esquerre:
 - Botó d'informació: obre un arxiu amb una guia d'ús de l'aplicació.
 - La terminal amb els punts de les àrees ja no és editable, només informa.
- Marc dret:
 - Botó de centrar vista a la posició actual
- Marc fons:
 - Terminal de logs: llegeix el document de logs incorporat a la darrera modificació, i mostra els missatges per aquesta terminal incorporada. Ha suposat una nova gestió de fils d'execució. Incorpora un codi de colors per a

cada classe de missatge.

- Botó “Log File”: obre l’arxiu on es guarden tots els logs.
- Indicador de l’estat de la connexió Bluetooth i satel·lital segons colors. El color verd fosc indica connexió amb la placa i posició estable, mentre que el verd clar indica posició imprecisa. El color groc indica que no hi ha un posicionament, però si connexió amb la placa, alhora que el taronja indica que no n’hi ha, però que es cerca activament. Finalment, el vermell mostra que ja no es cerca establir la connexió perquè s’ha superat cert nombre d’intents.
- Botó de reconnexió: quan l’indicador del senyal Bluetooth és vermell, aquest botó s’activa, i et permet reiniciar la cerca.
- Indicador visual per a l’aplicació de geofencing: fins ara la funció indicava si la posició era o no dins de l’àrea per la terminal on s’executés el programa, un mètode unacceptable. Ara es mostra a través d’un text i a la terminal de logs.

El funcionament del registre de logs i la terminal integrada es pot consultar al diagrama de l’annex 13.

Aquesta és la versió definitiva de l’aplicació “Geofencing Application - Leo Sarria”.

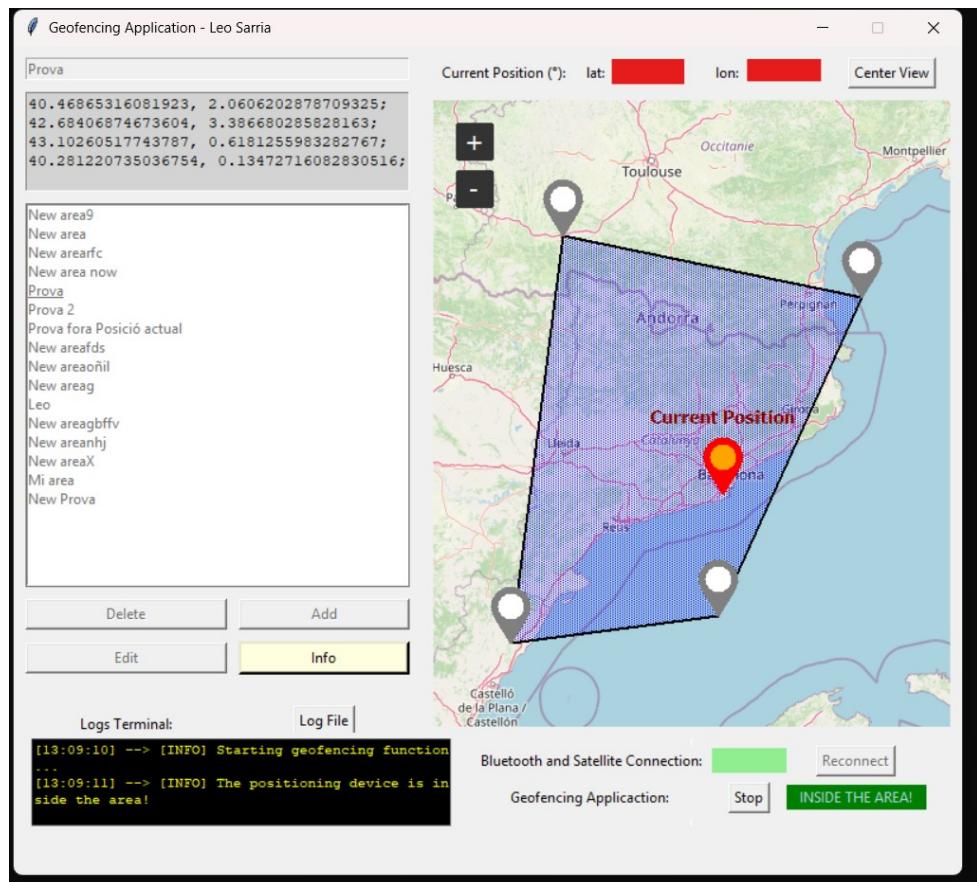


Fig. 32: Aplicació funcional final, amb el programa de geofencing en marxa. Autoria pròpia.

16. Informació sobre l'aplicació

L'aplicació es pot trobar a GitHub, a l'enllaç proporcionat més endavant. Té una llicència MIT, per tant, es pot descarregar, modificar, i emprar per a qualsevol fi, inclòs el comercial, sempre que s'indici l'autor original i es mantingui la llicència. Disposa d'un arxiu README que explica aspectes sobre l'aplicació, i una guia d'ús que es pot accedir també des de la mateixa aplicació. Es trobarà una versió en anglès i traduïda al català d'aquesta guia al repositori de GitHub.

S'ha decidit fer-la en anglès perquè, en publicar-la en un repositori a GitHub, és més accessible internacionalment.

Totes les llibreries de les quals fa ús l'aplicació s'han plasmat a l'annex 14.

17. Pressupost global del projecte

En aquest apartat es farà una petita incisió pel que fa a l'aspecte econòmic del projecte. Aquests són els elements que s'han fet servir i el seu preu aproximat, hi ha

enllaços a un lloc de compra fiable al annex 15:

- Placa ESP32 STEAMakers: 15 €
- rtl-sdr V3 (en aquest cas es va adquirir juntament amb un kit, amb una antena dipol i trípode per a aquesta): 50 €
- Mòdul GPS NEO-7M: 16 €
- Antena activa: 12 €

Com es pot comprovar, el rtl-sdr, tot i anar amb un kit, és l'objecte més car. Però s'ha de considerar el fet que permet rebre tota mena de senyals, a diferència del mòdul GPS, que només té una funció. Tot i això, al final és prescindible, ja que no s'ha pogut utilitzar. L'antena activa es recomana perquè la velocitat i precisió del mòdul GPS millora considerablement amb aquesta connectada.

CONCLUSIONS

Aquest projecte ha estat molt enriquidor, principalment, perquè s'ha gaudit de l'oportunitat de poder investigar el funcionament de diferents tecnologies amb gran influència en el món actual. Entre elles, els sistemes GNSS, i els receptors definits per programari. Aquests conceptes s'han comprès completament.

Respecte al desenvolupament de l'aplicació, s'han extret certes conclusions que seran de gran utilitat. Com la importància de treballar amb una estructura de codi modular amb lògiques separades, els elements imprescindibles d'una aplicació, o la documentació que l'ha d'acompanyar.

Cal destacar, tot i que no s'hagi pogut aplicar, tota l'experiència que s'ha adquirit amb la metodologia SDR, i aplicacions relacionades a aquesta. D'igual manera, s'ha testejat en altres contextos de manera exitosa, com escoltar emissores de ràdio, o comunicacions entre avions i la torre de control.

Tanmateix, aquesta recapitulació només referència aprenentatges derivats del mateix temari del treball, i no proporciona una visió completa de totes les conclusions extretes.

Per començar, amb aquest projecte s'ha après que el resultat no sempre serà correcte. Aquest ha estat el cas, ja que, independentment de la recerca teòrica que s'ha realitzat prèviament, la quantitat de recursos i documentació, i el temps invertit, no s'ha pogut fer funcionar el dispositiu rtl-sdr per a una aplicació de posicionament. No obstant això, aquesta ha estat una oportunitat immensa, pel fet que en comptes de prosseguir unes instruccions i aconseguir allò que es cerca, s'ha hagut de dedicar un esforç major, i com a resultat s'ha pogut comprendre molt més a fons les tecnologies amb les quals s'ha treballat. Altrament, s'ha hagut de decidir deixar de banda l'objectiu inicial i buscar una alternativa, amb la finalitat de poder avançar amb el projecte, i per tant s'ha après a actuar no només segons la planificació inicial, sinó amb el material real i la situació actual.

Paral·lelament, sí que s'han assolit la resta d'objectius, ja que s'ha pogut aconseguir el posicionament en temps real amb el mòdul GPS, i integrar-la amb l'aplicació, la qual és funcional, amb totes les utilitats que es pretenia des d'un inici, i amb una interfície intuïtiva.

En definitiva, aquest projecte ha permès comprendre i aplicar els principis fonamentals de posicionament per satèl·lit, desenvolupant alhora una aplicació que reflecteix tot el procés d'aprenentatge. Per tot això, aquest projecte es considera exitós.

ENLLAÇOS IMPORTANTS

Carpeta de Drive amb material sobre el projecte:

[https://drive.google.com/drive/folders/1t67lw22AGPEhrx539AiQmShMc4wpZvie?
usp=sharing](https://drive.google.com/drive/folders/1t67lw22AGPEhrx539AiQmShMc4wpZvie?usp=sharing)

O a través del codi QR següent:



Repositori de GitHub amb el codi del projecte:

https://github.com/LeoNotAvailable/Geofencing-Application_Research-Project

O a través del codi QR següent:



BIBLIOGRAFIA

Seco, G; López, J. (2014). *Introducción a los sistemas de radionavegación y radiodeterminación*. Universitat Oberta de Catalunya.

<https://openaccess.uoc.edu/server/api/core/bitstreams/59a18fff-13ba-4051-8de0-ad8f55ba9544/content>

Abelardo, D. (2008). *TFG Sistema GNSS (GLOBAL NAVIGATION SATELLITE SYSTEM)*. Arantxa.

<http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20080125DavidGarcia.pdf>

Machado, J. (2014). *Software Defined Radio: Basic Principles and Applications*. Semantic scholar.

<https://pdfs.semanticscholar.org/775e/0c10f8c2a375b81366cbb8770f298130a900.pdf>

GPS. 27 de septiembre de 2025. Wikipedia. <https://es.wikipedia.org/w/index.php?title=GPS&oldid=169691647>

Transit (Satellite). 23 de octubre de 2025. Wikipedia.
[https://en.wikipedia.org/w/index.php?title=Transit_\(satellite\)&oldid=1318326431](https://en.wikipedia.org/w/index.php?title=Transit_(satellite)&oldid=1318326431)

User Guides. (14 de juliol de 2025). Navipedia.
https://gssc.esa.int/navipedia/index.php/User_Guides

Dana, P. (10 de desembre de 2018). Electronic Navigation, Precise Positioning, and Geographic Information Systems Research and Development. pdana.

<http://www.pdana.com/PHDWWW.html>

GNSS. (35 de octubre de 2017). Raege. <https://raege.eu/es/gnss-es/>

Vilajosana, X. (2 de marzo de 2023). ¿Qué es el GNSS?. Universitat Oberta de Catalunya. <https://blogs.uoc.edu/informatica/es/auge-del-gnss/>

Tipos De Satélites: Diferencias En Sus Funciones Y Utilidad. (26 de abril de 2023). eos.
<https://eos.com/es/blog/tipos-de-satelite/>

Space Segment. GPS.gov. <https://www.gps.gov/space-segment?>

GPS Ground Segment. (5 Novembre 2020). Navipedia.
https://gssc.esa.int/navipedia/index.php/GPS_Ground_Segment

GPS User Segment. (4 Novembre 2020). Navipedia.
https://gssc.esa.int/navipedia/index.php/GPS_User_Segment

Antenas GPS y Sistemas GPS mejorados. (18 de octubre de 2019). Leedeo.
<https://www.leedeo.es/l/antena-de-gps-y-sistemas-gps-mejorados/>

McAlister, Chris. (septiembre de 2023). GNSS Signals. pressbooks.
<https://usq.pressbooks.pub/gpsandgnss/chapter/3-4-gnss-signals/>

Garrido, N. (24 de noviembre de 2016). Errores atmosféricos en GNSS (GPS). nagarvil.
<https://nagarvil.webs.upv.es/errores-atmosfericos-gnss-gps/>

¿Cómo mitigar el impacto de la ionosfera en el posicionamiento GNSS?. (22 d'agost de 2022). comnavtech. <https://www.comnavtech.com/sp/about/blogs/383.html>

What Is GNSS Multipath Mitigation?. (13 de septiembre de 2023).U-Blox. <https://www.u-blox.com/en/technologies/multipath-mitigation>

Tech Minds. (28 d'agost 2020). *Decoding GPS using an RTL SDR Receiver*. Youtube.
<https://www.youtube.com/watch?v=YG2fJRTAoHA>

Fuzz the Pi Guy. (4 de desembre de 2021). *GPS Decoding And Plotting Using a RTL SDR And GPS Antenna | GNSS-SDR | RTKLIB*. Youtube. <https://www.youtube.com/watch?v=hksUORPVNDM&t=14s>

ANNEXES

Annex 1:

Taula amb característiques de cada sistema GNSS. Es tracta de dades extretes a mitjans de l'any 2025 que poden no ser vàlides en dates futures, ja que varien segons actualitzacions, satèl·lits llençats en òrbita, posats en servei o retirats.

	Nº de satèl·lits	Alçada	Inclinació orb.	Període orb.
GPS	31 (necessita 24 mínim)	20200 km	55°	11 h, 58'
GLONASS	29 (24 actius)	19100 km	64.8°	11 h, 15'
Galileo	30 (24 actius)	23222 km	56°	14 h
BeiDou-3	30	22000 km	55°	12 h, 53'
QZSS*	4 (incomplet)	32 000–39 000 km	75.4°	23 h, 56'

* El sistema QZSS, que només disposa de 4 satèl·lits perquè encara no es troba terminat (es preveu que llencin 7 més), té un dels seus satèl·lits en òrbita geoestacionària a uns 35700 km d'alçada, però els altres tres compleixen les característiques mostrades.

Annex 2:

Taula amb les característiques generals dels tipus d'antenes més freqüents en receptors GNSS.

Tipus d'antena →	De "patch" o plana	Helicoidal	Xip o lineal
Característiques	Incorpora un dielèctric ceràmic	Forma de tub amb el material receptor envoltant-lo	Integrable com a micro component elèctrica a circuits
Tamany mínim	70x70 mm	100x100 mm	3,2x1,6x1,1 mm
Avantatges	Baix cost i guany acceptable de 5dB	Similars característiques a la de patch però amb una amplada de banda major	Tamany reduït i fàcil integració en PCB
Desavantatges	Amplada de banda estreta i soroll considerable	Dimensions i cost de producció superiors a la de patch	Guany entre 3 i 6 dB inferior a les altres dues

Annex 3:

Missatge mostrat per terminal de Windows una vegada executat el programa de bias tee:

```
C:\Users\leo_k\Ordenador\Treball de recerca\Software RTL-SDR\bt_driver>bias_tee_on.bat →
S'executa el programa
```

```
C:\Users\leo_k\Ordenador\Treball de recerca\Software RTL-SDR\bt_driver>rtl_biast -d 0 -b 1
Found Rafael Micro R820T tuner → El programa identifica el dispositiu
```

Annex 4:

Aquesta és una mostra de la terminal d'una execució no exitosa de GNSS-SDRlib, on només es veuen les dades del satèl·lit:

```
R3, C/N0=34,5, peak=1,1, codei= 238, freq= -4000,0
R4, C/N0=34,8, peak=1,2, codei= 1189, freq= 3000,0
G03, C/N0=34,9, peak=1,2, codei= 251, freq= 6400,0
R5, C/N0=34,8, peak=1,4, codei= 1822, freq= -3400,0
G16, C/N0=34,5, peak=1,1, codei= 990, freq= 6000,0
R6, C/N0=34,6, peak=1,2, codei= 946, freq= 1600,0
G08, C/N0=35,1, peak=1,2, codei= 257, freq= 2800,0
```

Annex 5:

Mostra d'una execució exitosa del programa GNSS-SDRlib, realitzada per Taro Suzuki (el creador d'aquest mateix programa) amb un dispositiu rtl-sdr.

```
G03 ID= 2 tow:311322.0 week= 2120 cnt=232619
G19 ID= 2 tow:311322.0 week= 2120 cnt= 221278
```

En aquest fragment es poden observar diferents dades. La primera és el canal intern que usa l'aplicació per gestionar els satèl·lits, irrelevat per al posicionament. La segona és tow (time of the week), que representa el temps en què es va emetre el senyal amb relació a una setmana, en segons. En aquest cas, un dimecres passat el migdia. La tercera dada és week= 2120, que és el nombre de setmanes que han passat des de l'inici del temps en GNSS. En aquest cas, va ser a inicis de 1980, per tant, han passat 2120 setmanes des de llavors. És a dir, 40,65 anys des de llavors.

Vídeo de l'execució: <https://www.youtube.com/watch?v=BpbCi9Ou1mo>

Annex 6:

Habilitació de WSL2 (Windows Subsystem for Linux) des de Windows Powershell amb privilegis d'administrador.

```
PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-  
Subsystem-Linux /all /norestart  
>> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart  
>> wsl --set-default-version 2  
>> wsl --install -d Ubuntu
```

Annex 7:

Instal·lació de usbipd-win;

```
PS C:\Users\leo_k> winget install --id=dorssel.usbipd-win -e  
Encontrado usbipd-win [dorssel.usbipd-win] Versión 5.2.0  
El propietario de esta aplicación le concede una licencia.  
Microsoft no es responsable, ni tampoco concede ninguna licencia de paquetes de terceros.  
Descargando https://github.com/dorssel/usbipd-win/releases/download/v5.2.0/usbipd-  
win_5.2.0_x64.msi
```

 4.10 MB / 4.10 MB

El hash del instalador se verificó correctamente

Iniciando instalación de paquete...

El instalador solicitará que se ejecute como administrador. Espere una indicación.

Instalado correctamente

Annex 8:

Llistat dels dispositius connectats a l'ordinador:

```
PS C:\Users\leo_k> usbipd list  
Connected:  
BUSID VID:PID DEVICE STATE  
2-1 0bda:2838 RTL2838UHIDIR Not shared  
2-6 25a7:fa23 Dispositivo de entrada USB Not shared  
2-7 04f2:b773 ACER HD User Facing, Camera DFU Device Not shared  
2-10 13d3:3594 MediaTek Bluetooth Adapter Not shared  
Persisted:  
GUID DEVICE
```

Annex 9:

```
PS C:\WINDOWS\system32> usbipd bind --busid 2-1  
PS C:\WINDOWS\system32> usbipd attach --wsl --busid 2-1  
usbipd: info: Using WSL distribution 'Ubuntu' to attach; the device will be available in all WSL 2
```

distributions.

```
usbipd: info: Loading vhci_hcd module.  
usbipd: info: Detected networking mode 'nat'.  
usbipd: info: Using IP address 192.168.144.1 to reach the host.
```

Annex 10:

Processament exitós amb fix de posició segons l'exemple mostrat a: [gnss-sdr.org /my-first-fix/](http://gnss-sdr.org/my-first-fix/)

New GPS NAV message received in channel 2: subframe 3 from satellite GPS PRN 17 (Block IIR-M)

New GPS NAV message received in channel 1: subframe 3 from satellite GPS PRN 11 (Block IIR)

First position fix at 2013-Apr-04 06:23:31.740000 UTC is Lat = 41.2749 [deg], Long = 1.98754 [deg], Height= 100.795 [m]

Position at 2013-Apr-04 06:23:32.000000 UTC using 4 observations is Lat = 41.274888307 [deg], Long = 1.987581872 [deg], Height = 86.928 [m]

Position at 2013-Apr-04 06:23:32.500000 UTC using 4 observations is Lat = 41.274964746 [deg], Long = 1.987510141 [deg], Height = 90.557 [m]

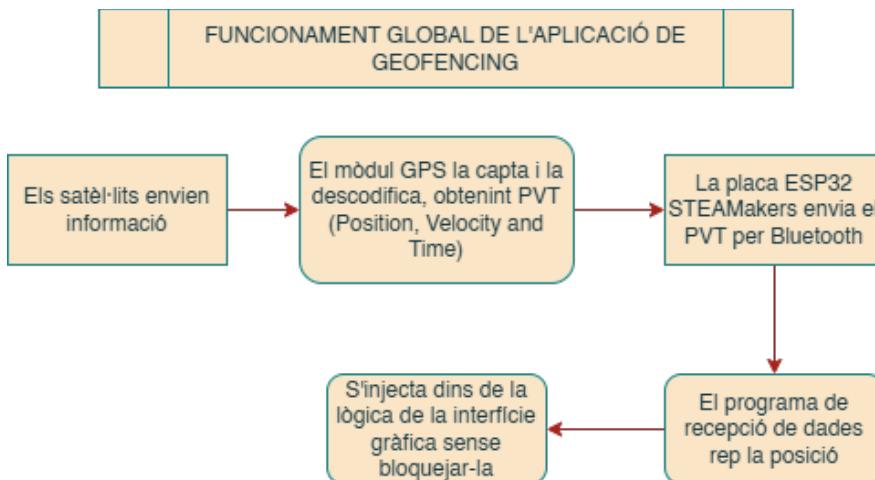
Current receiver time: 27 s

Position at 2013-Apr-04 06:23:33.000000 UTC using 4 observations is Lat = 41.274921885 [deg], Long = 1.987605767 [deg], Height = 73.365 [m]

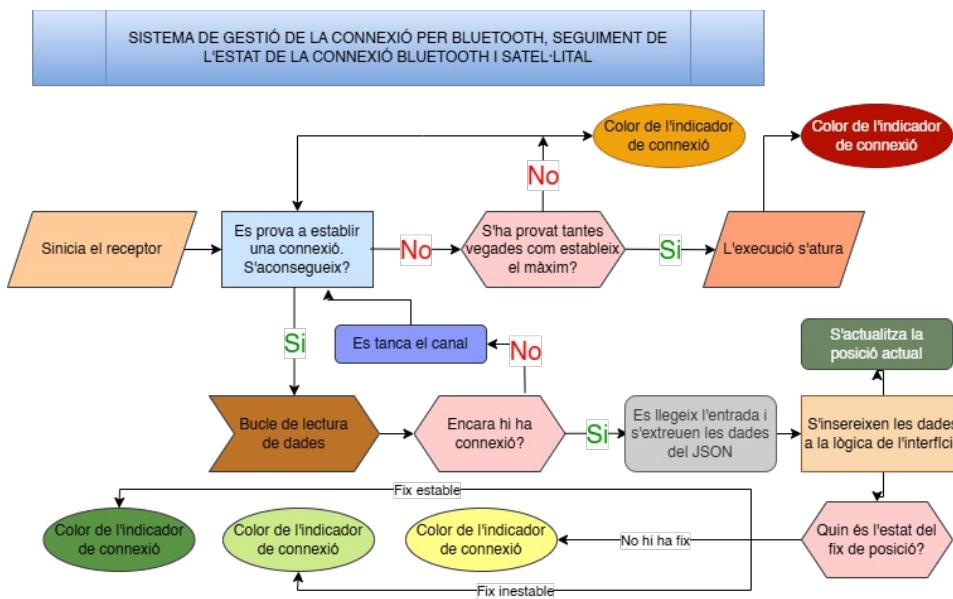
Position at 2013-Apr-04 06:23:33.500000 UTC using 4 observations is Lat = 41.274866502 [deg], Long = 1.987553835 [deg], Height = 83.313 [m]

Current receiver time: 28 s

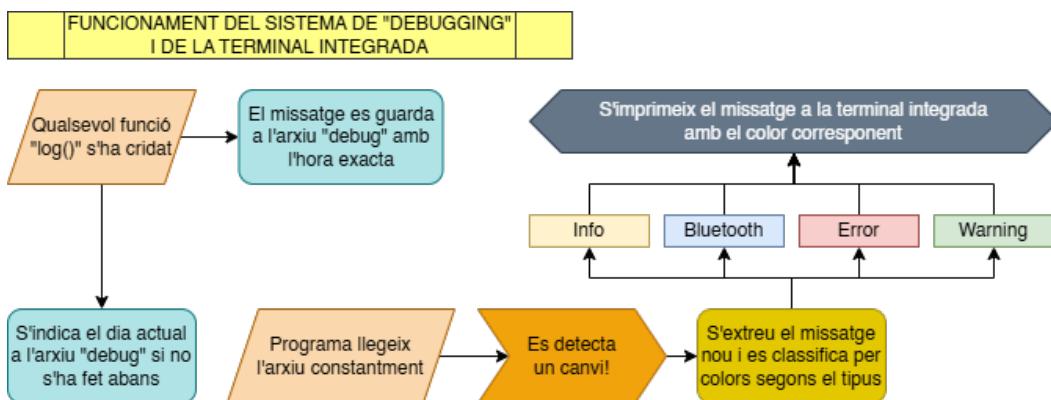
Annex 11:



Annex 12:



Annex 13:



Annex 14:

Totes les llibreries utilitzades per l'aplicació “Geofencing Application” de Python:

- tkinter: permet crear interfícies d'usuari senzilles amb diferents objectes
- tkintermapview: incorpora lògica de mapes a tkinter, amb aquest objecte
- threading: gestió dels fils d'execució
- time: gestiona la funció de parar durant uns segons dins del programa
- datetime: proporciona l'hora acurada, utilitzat aquí per a la gestió de logs
- json: llegeix, converteix informació a aquest format específic.
- os: obrir arxius, llegir-los, editar-los, obrir la seva finestra per l'usuari...
- serial: maneja l'entrada de dades, l'entrada Bluetooth es gestiona com una entrada física més.
- shapely.geometry: per crear els polígons i comprovar si la posició és a dins.
- math: funcions matemàtiques diverses

Annex 15:

Enllaços on s'ha comprat el hardware d'aquest projecte:

Placa ESP32 STEAMakers: https://www.amazon.es/KEYESTUDIO-Desarrollo-Compatible-Bluetooth-Arduino/dp/B0BHZ8H6LM/ref=sr_1_21?adgrpid=120677549978&dib=eyJ2ljojMSJ9.TRGsqlzMMOgAW_6Mij09wX0t9Uar8oow_vypqoOsbTEvpGZEjtdbNZqTeeak6xwmvHGkp4Et6Zynk6Oq2Kvq3zr0Z4JmeXt732LO0piTwmih0gh8n0IVUZ0Yj_GtHGuultOoqMKEqCZjBzHkNQzCNe0YG4IKF7s9Ci2WAnQ0_k-eSC0Q1nKqjB51Q_IIIEt4cMRM1bBoo-R7-AVsoxe4mG8oNh-GJ--pcgaTNWjlwzi4E1M6IYEu65QDYRREhOPQXt1C0Y4_NRzPo9LQXdtvhVIY9i-F5FWXr5fjlduKO8.ZNJimoilKUI24UaZbJvPI35zuARTTYQdv_PrvmSAPDM&dib_tag=se&hvadid=601382925032&hvdev=c&hvexpln=0&hvlocphy=9197910&hvnetw=g&hvocijid=8731253270532127222--&hvqmt=b&hvrand=8731253270532127222&hvtargid=wd-1210216474947&hydadcr=19377_2264613&keywords=esp32+placa&mcid=bb09e21e3bc833759225ae9ee730441f&qid=1761477968&sr=8-21

Kit rtl-sdr V3: https://www.astroradio.com/p/receptor_rtlsdr_blog_r820t2 rtl2832u_antena_dipolo/?srstid=AfmBOop1uytSr3VI2U5E8LZ-9NxivkWTYnb0wP2Yoqbt3plzTPI37elSN

Mòdul GPS NEO-7M: https://www.amazon.es/dp/B08R5ZQX4L/ref=sspa_dk_detail_4?psc=1&pd_rd_i=B08R5ZQX4L&pd_rd_w=kqe4J&content-id=amzn1.sym.0c640cbd-b6e0-461d-8cfc-a8934b5122df&pf_rd_p=0c640cbd-b6e0-461d-8cfc-a8934b5122df&pf_rd_r=VWBPA24V0C5GWJ02RX2X&pd_rd_wg=EJJIM&pd_rd_r=7fd61737-02bd-4caf-9fb9-4a3ac9bbe27c&s=electronics&sp_csd=d2lkZ2V0TmFtZT1zcF9kZXZhawWw

Antena activa: https://www.amazon.es/Othmro-Antena-Activa-Ganancia-Fuerte/dp/B0838YP81R?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&psc=1&smid=A3LT74XL5YCZAN