POLITECNICO DI TORINO

Department of Electronics and Telecommunications

Master's Degree Thesis in Electronic Engineering

Study of Spoofing Threats Against GNSS Positioning, Navigation and Timing Units

Supervisors

Prof. Fabio DOVIS

Correlator

Ph.D Akmal RUSTAMOV

Candidate

Ten. Giorgio TRIPANO

Academic Year 2021-22

† A mio padre

Summary

A comparative analysis of the performance of different smartphones under intentional interference is performed and the effect on raw GNSS measurements is seen. In the thesis, we focus on the design and validation of different signal processing techniques, that aim at the detection and mitigation of the spoofing attack effects. These are standalone techniques, working at the receiver's level and providing discrimination of spoofing events without the need for external hardware or communication links. Four different techniques are explored each of them with its unique sets of advantages and disadvantages, and a unique approach to spoofing detection. For these techniques, a spoofing detection algorithm is designed and implemented, and its capabilities are validated by means of a set of datasets containing spoofing signals. The thesis focuses on two different aspects of the techniques, divided into detection and mitigation capabilities. Both detection techniques are complementary, their joint use is explored and experimental results are shown that demonstrate the advantages. In addition, each mitigation technique is analyzed separately as they require specialized receiver architecture in order to achieve spoofing detection and mitigation. These techniques are able to decrease the effects of the spoofing attacks, to the point of removing the spoofing signal from the receiver, and compute navigation solutions that are not controlled by the spoofer and lead to more accurate end results. The main contributions of this thesis are the description of a multidimensional ratio metric test for distinction between spoofing and multipath effects; the introduction of a cross-check between automatic gain control measurements and the carrier to noise density ratio, for the distinction between spoofing attacks and other interference events; the description of a novel signal processing method for detection and mitigation of spoofing effects, based on the use of linear regression algorithms; and the description of a spoofing detection algorithm based on a feedback tracking architecture.

Acknowledgements

Before I begin my paper, I would like to dedicate this space to those who have contributed to its realisation and to those who have supported me over the years.

My special thanks go to Professor Fabio DOVIS, who enabled me to carry out and develop this interesting topic, and for the knowledge he passed on to me throughout the process of writing the paper.

I would also like to thank Dr. Rustamov AKMAL, my thesis co-rapporteur, for his time, constant support, indispensable advice and for promptly suggesting the right changes to my thesis.

Finally, I would especially like to thank all my loved ones who have always been at my side during these years of training to encourage and spur me on to do my best.

Giorgio TRIPANO

Table of Contents

List of Tables

List of Figures

Acronyms

ADC	Analogue (to) Digital Converter
AGC	Automatic Gain Control
AI	Artificial Intelligence
AOA	Angle-of-Arrival
ARM	Advanced RISC Machine / Acorn RISC Machine
ARP	Address Resolution Protocol
BDS	Beidou
BPSK	Binary Phase Shift Keyinge
CAS	Commercial Authentication Service
CDMA	Code Division Multiple Access
COTS	Commercial Off-the-Shelf
CP	Cooperative Positioning
CPLD	Complex Programmable Logic Device
DDoS	Distributed Denial of Service
DKIM	Domain Keys Identified Mail
DMARC	Domain-based Message Authentication Reporting and Conformance
DAC	Digital (to) Analog Converter
DNS	Domain Name Server
DOP	Dilution of Precision
DoS	Denial-Of-Service
EGM	Electrical Grid Monitoring
EGNOS	European Geostationary Navigation Overlay Service
ERMES	Enhanced Radio Messaging System

EUSPA	European Global Navigation Satellite Systems Agency
FDD	Frequency Division Duplex
FHSS	Frequency Hopping Spread Spectrum
FLP	Fused Location Provider
FM	Frequency Modulation
FTP	File Transfer Protocol
GAL	Galileo
GLO	GLONASS
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input / Output
GPR/WPR	Ground Probing Radar / Wall probing radar
GPS	Global Positioning System
GRC	GNU Radio Companion
GSM	Global System for Mobile communications
GSM-R	GSM Railway
GUI	Graphical User Interfaces
HAPS	High Altitude Platform Stations
HEST	High Eirp Satellite Terminals
HIPERLAN	HIgh PErformance Radio Local Area Network
HRF1	HackRF One
I2C	Inter Integrated Circuit
ICD	Interface Control Document
ID	Identification
IF	Intermediate Frequency
ILS	Instrumental Landing System
IMT	International Mobile Telecommunications
IMU	Inertial Measurement Unit
IO	Input / Output
IP	Internet Protocol
ISM	Industrial Scientific and Medical
ITRF	International Terrestrial Reference Frame
ITS	Intelligent Transport Systems
LAN	Local Area Network
LCD	Liquid Crystal Display
LTE	Long Term Evolution

MAC	Media Access Control
MIME	Multipurpose Internet Mail Extensions
MitM	Man in the Middle
NLP	Network Location Provider
NMEA	National Marine Electronics Association
NRTK	Network Real Time Kinematic
OSNMA	Open Service Navigation Message Authentication Signal
PC	Personal Computer
PCB	Printed Circuit Board
PNT	Positioning, Navigation and Timing
PrM	Pseudorange Measurement
PRN	Pseudo-Random Noise
PVT	Position Velocity Time
REPL	Read Eval Print Loop
RF	Radio Frequency
RFI	Radio Frequency Interference
RFID	Radio Frequency Identification
R-LAN	Radio Local Area Network
S-PCS	Satellite-Personal Communications Systems
SAW	Surface Acoustic Wave Filters
SBAS	Satellite-Based Augmentation System
SD	Secure Digital
SDR	Standard Dimension Ratio
SET	Social Engineering Toolkit
SIS	Signal in Space
SIT	Satellite Interactive Terminal
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SPF	Sender Policy Framework
SPI	Serial Peripheral Interface
SRD	Short Range Devices
SRR	Short Range Radar
SSR	Secondary Surveillance Radar
SUT	Satellite User Terminal

TACAN	TACtical Air Navigation
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TOA	Time-Of-Arrival
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Aerial Vehicle
UMTS	Universal Mobile Telecommunications System
URL	Universal Resource Locator
USB	Universal Serial Bus
UTC	Universal Time Coordinated
UWB	Ultra Wide Band
V2V	Vehicle to Vehicle
WAS/LAN	Wireless Access Systems - Local Area Network
WBFM	Wide Band Frequency Modulation
WLL	Wireless Local Loop
WLS	Weighted Least Square
WWW	World Wide Web

Chapter 1

Introduction

1.1 Background Global Navigation Satellite System (GNSS)

Radio navigation is of utmost importance in several application fields. Nowadays, many civil and professional applications massively rely on the GNSS and related technologies to accurately estimate position and time. Existing GNSS-based systems are threatened by malicious attacks among which spoofing and meaconing constitute severe challenges to the receiver. Several of such GNSS systems constitute mass market applications and devices, and a threat to the GNSS receiver could have cascading effects at application levels and for interconnected systems. Networked GNSS receivers are in general ubiquitous because any receiver embedded in a complex system such as a smart device or smart connected cars can exploit network connectivity. This novel generation of valuable-performance GNSS receivers is prone both to standard RF spoofing attacks and to cyber-attacks conceived to hijack complex network-based services such as DGNSS-based cooperative positioning. By means of a set of experimental tests, this paper highlights possible metrics to be checked to identify malicious attacks to the positioning and navigation systems in mass market connected devices. The network-based exchange of GNSS data such as GNSS raw measurements recently disclosed in Android smart devices is conceived in this work to offer the possibility to compare or combine such metrics to better identifies spoofing and meaconing attacks. The presence of the GNSS technology in modern life has been constantly growing in recent years, supporting the use of GNSS receivers in diverse fields of applications involving civilian, military, leisurely, the safety of life and financial sectors. A constantly growing attention is being devoted to the security and safety of GNSS related technologies with one of the threats being malicious attacks such as spoofing and meaconing. The terms refer to an unauthorized transmission of locally generated Radio Frequency (RF) signals forcing receivers nearby to compute a fake Position Velocity Time (PVT) solution. From the perspective of GNSS signals, according to the current version of the Interface Control Documents (ICDs), GNSS signals (e.g. GPS L1 C/A and GLONASS) do not provide any means to the receivers by default in order to ensure the authenticity of the source of the satellite signals or to improve the robustness of the receiver against possible spoofing attacks. The European Galileo is on E1 and it is planning the use of an Open Service Navigation Message Authentication Signal (OSNMA) and the Commercial Authentication Service (CAS) on E6 for allowing users to calculate PVT solutions based on trusted signals in the near future. While these countermeasures are being implemented at system level to provide a minimum level of protection also on signals exploited by mass-market applications, there is a widespread debate if even by means of a simplistic spoofing attack it would be possible to threaten GNSS receivers with a cascading effect at application level and for interconnected systems. It is known that a lack of synchronization between spoofers and GNSS timescale can be theoretically used to detect such spoofing attacks with a small effort. On the other hand, specific dedicated detectors are not implemented in most positioning units in the consumer field such as simple unmanned aerial vehicles (UAVs), smartphones and other personal devices, making them vulnerable to spoofing attacks. From a general perspective, the GNSS receiver plays a core role as being usually the only one providing an absolute estimation of the position. However, in mass market devices, it is part of a positioning unit that is comprised of GNSS technology, along with Inertial Measurement Units (IMUs), electronic compasses, etc., which helps in aiding or refining the positioning solution. Furthermore, in several cases, the positioning unit is typically interfaced with an application layer (i.e. Location-Based Services), and the position information or related measurements are exchanged to other services or remotely processed in remote servers such as for snapshot positioning. Such an architecture is prone to any of common spoofing attacks at the signal and non-signal level (cyberattacks) especially if it is based on low cost commercially available off-the-shelf (COTS) products which use standard positioning services (GPS C/A, Galileo Open Service, GLONASS, etc.), and standard unencrypted communication services. Consequently, it is worth examining the impairment of intentional interferences to the low-cost GNSS units embedded in the mass market receivers and further assessing the resilience of the overall device or platform. Many researchers have utilized different methods to measure spoofing attacks in recent years. A number of techniques have been developed since then to mitigate spoofing for GNSS security. Despite these remarkable results, they require additional hardware or changes to the interface specification. We aim to develop an anti-spoofing detection and coping mechanism in connected COTS GNSS devices, presenting the assessment of parameters and metrics to understand the occurrence of a spoofing attack taking advantage of the cooperative paradigm of networked GNSS receivers. This approach is particularly interesting to explore because of the appealing application of Cooperative positioning (CP) in short-range communication networks such as Vehicle to Vehicle (V2V) in vehicular navigation field and distributed ad hoc infrastructures. In order to demonstrate the potential of CP techniques, a popular COTS GNSS receiver and different commercial Android smartphone devices were used. An advanced experiment was conducted considering a real-time-oriented collaborative framework using a combination of raw measurements to determine inter-agent distances and improve the position estimation accuracy. A low-cost portable spoofer platform able to place simplistic attacks is developed and experimental tests are performed on selected COTS devices.

1.2 Motivation

GNSS receivers are considered particularly vulnerable to spoofing attacks. The risk of jamming and spoofing attacks on GPS receivers is considered significant due to the recent increase in the threat of spoofing attacks. Since GNSS receivers are vulnerable to a variety of attacks, it is critical to establish defense mechanisms against GNSS spoofing attempts. It is well known that no COTS (Commercial off-the-shelf) GNSS receiver is capable of defending against a modern spoofing attack, although the threat of spoofing has recently attracted increasing attention. Given the spoofing problem, it is critical that receivers be equipped with methods to detect spoofing attempts. The purpose of spoofing detection is to warn the user not to trust the position and time determination. However, there is also a need to develop GNSS receivers that can recover a reliable position and time in the presence of spoofing signals.

1.3 Aim, Objectives and Contributions of the thesis

The aim of this thesis is to investigate the possibility of mitigating spoofing attacks against a GNSS receiver by sharing pseudorange or carrier phase measurements with other GNSS receivers. The case of two receivers exchanging these measurements will be investigated. The mitigation approach should identify measurements from spoofing signals and omit them from the computation of position, velocity and time (PVT). The focus is on the use of pseudorange measurements to perform spoofing mitigation, but it is compared to the use of carrier phases.

The research questions investigated in this thesis are:

- Is it possible to use pseudorange measurements from two receivers in order to mitigate a spoofing attack?
- How is the ability to perform spoofing mitigation using pseudoranges affected by different distances between the receivers and by different power levels on the spoofing signals?
- How does the performance of spoofing mitigation using carrier phases compare to using pseudorange measurements?

1.4 Limitations

The thesis considers only the civilian, open (unencrypted) GNSS signals. The developed algorithms are general and apply to different GNSS; however, evaluations have been performed exclusively with the GPS C/A signal on the L1 (1,575.42 MHz) carrier frequency. Furthermore, only spoofing attack types that have already been publicly described are considered. The spoofing device is assumed to have one transmission antenna. It is also assumed that the spoofing device neither transmits too much power, nor nulls the authentic signals. It would not be possible to recover the true signals if either of these two cases were true. Finally, it is assumed that all visible authentic satellite signals are spoofed during an attack and that receivers can acquire and track both the authentic and the spoofed signals in each case.

Chapter 2

Global Navigation Satellite System

2.1 Introduction to the GNSS

The European Global Navigation Satellite Systems Agency (EUSPA) defines GNSS as follows:

"Global Navigation Satellite System (GNSS) refers to a satellite constellation that provides signals from space capable of transmitting position and time data to GNSS receivers. The receivers then use this data to determine position. By definition, GNSS systems provide global coverage". [1]

This system provides a geo-spatial positioning service with global coverage that allows electronic receivers to determine their latitude, longitude and altitude with minimal error at any point on the earth's surface or in the atmosphere. Radio Frequency signals transmitted in line-of-sight from satellites are processed to obtain these values.

The performance of GNSS systems can be improved by regional Satellite-Based Augmentation Systems (SBAS), such as the European Geostationary Navigation Overlay Service (EGNOS). [2, 3, 4]

Examples of GNSS systems include:

- **NAVSTAR Global Positioning System (GPS)**; it is the first system to be realized, of US origin and still operational today. It was developed for military reasons and is still operated today by the US Army. It was born in 1993 with 24 satellites in orbit but only reached full functionality in 1995. Because of the way it is structured, it allows the free horizon view of at least 4 satellites at all times. It is currently the most advanced and widely used system; [5, 6, 7]

- **GLONASS**; a Russian system in operation since 1993. Today's state-of-the-art GPS receivers are able to work simultaneously with NAVSTAR and GLONASS; [5, 8, 9]

- **Galileo**; is a European civil system in operation since 2016; [10, 5, 11]

- **BeiDou**; is a Chinese positioning system currently only active at a regional level covering all of Asia, India, Japan and Australia; [5]

- **IRNSS**; is India's regional system covering the entire country and the surrounding 1500 km or so belt, which came into full operation in 2016 with the launch of the last three satellites needed. [3, 5]

SYSTEM	GPS	GLONASS	Galileo	BeiDou	IRNSS
Owner	USA	Russia	UE	China	India
Geodetic Datum	WGS84	PZ-60	WGS84	–	–
Minimum Error	5 m	5 m	1 m	10 m	20 m
Orbital Altitude	20.180 Km	19.130 Km	23.222 Km	21.150 Km	36.000 Km
Orbital Period	11 h 58 min	11 h 16 min	14 h 5 min	12 h 38 min	23 h 56 min
Orbiting Satellites	31	27	24	23	7
Planned Satellites	24	24	26	35	7
Status	Operative	Operative	Operative	Operative	Operative
Active Coverage	Global	Global	Global	Loc.(Global)	Local

Table 2.1: Comparative Table of GNSS Systems. [5]

2.2 Functional principles of GNSS

The objective of a GNSS system is to allow the operator who decides to use it to calculate the PVT solution (position, velocity and time). To achieve this, it is necessary to have a reference system for position and time; in particular, GNSS systems use the TOA (time-of-arrival concept) to arrive at the PVT solution. [12]

The TOA is the difference between the instant of departure from the satellite and the instant of arrival at Earth, measured with the receiver's clock 'synchronised' with that of the satellite. The signal propagation time from a satellite to the zenith is approximately 0.06 seconds.
Once the propagation time has been obtained, the receiver is able to calculate the distance between receiver and satellite. [13]

The operating principle of GNSS is based on multilateration, i.e. the spherical positioning method that starts by measuring the time it takes for a radio signal to travel the distance between the satellite and the receiver on the ground.
To achieve positioning, each satellite emits a continuous signal containing information about its position and the instant of time it left the satellite. The receiver also measures the instant at which the signal is received, but it will have a slight delay concerning the time at which the signal starts due to travel time. By exploiting this delay, the receiver can calculate the distance between it and the satellite. This time difference is present because the receiver and GPS are not in an ideal situation where their clocks are perfectly synchronised. [14]
To make an accurate measurement, the receiver and the satellite must have clocks that allow synchronisation with nanosecond precision. This level of precision can be achieved by using atomic clocks. Due to their high cost, atomic clocks are only found inside satellites, while receivers use a quartz clock that is reset periodically. To calculate the position in space, the receiver searches for the signal from the satellites and estimates its inaccuracy: the correct time value, corresponding to the time marked by all the atomic clocks of all the satellites, will cause all received signals to align to a single point in space. By doing so, the receiver can determine the correct value of time that will be used to calculate the position. Once the distance to a satellite has been determined, the operator's position must be on the surface of a sphere with a radius corresponding to this distance. To calculate the radius of the sphere it is assumed that the satellite sends the signal in the form of radio waves at time t_0 and that the user receives it at time $t_0 + \tau$ and therefore the radius of the sphere can be expressed as:

$$R_i = c\tau$$

Where c is the speed of light.

The receiver then calculates the correction required to make the spheres intersect at a single point, thus synchronising its clock with that of the satellites. This operation is repeated again and again by the receiver's clock, which will eventually possess an accuracy very similar to that of the satellites' atomic clocks. In conclusion, the position of the satellites is determined. Since the satellites follow very precise and well-defined orbits, this is done via information stored in the receiver's memory. It must be emphasised that in addition to the error due to travel time, it must be taken into account that the satellites move relative to the receiver, causing their clocks to run slower, that gravitational fields change both the speed of the clocks and the propagation of radio signals, and that the rotation of the Earth induces further negative effects. These factors can be taken into account through the use of the formulas proposed by the General and Special Theory of Relativity.

If we want to recapitulate these steps more simply, we could consider an operator who wants to obtain his position in a 3D system and to do so he generates a sphere that surrounds each satellite with a radius equal to the distance obtained. By observing the intersection of these spheres, he can obtain this information. Using the distance to four different satellites, four spheres can be obtained that intersect at a single point on the earth's surface, which corresponds precisely to the position where the receiver is located. Measurements could also be made with only three satellites, but the measurement would not be sufficiently reliable or precise. All these measurements are called pseudoranges. [15, 13, 12]

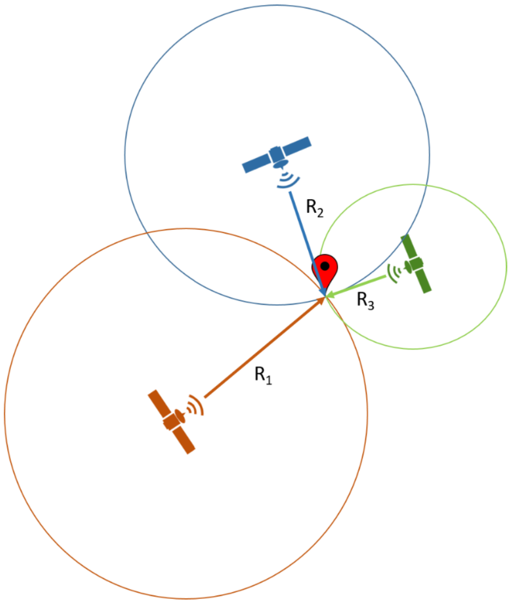


Figure 2.1: How GPS works. In this case is represented a trilateration with three different satellites.

2.3 Reference systems used by the GNSS

A position and time reference system must be defined when discussing GNSS systems. Due to the geoid shape of the earth, hence an irregular shape, the co-ordinate system used is the ellipsoid system in which the earth is approximated as an ellipsoid of revolution.

Therefore given a generic point P on the earth's surface an operator can define:

- **The geodetic latitude** φ, i.e. the angle on the meridian plane with respect to point P, between the equatorial plane and the line perpendicular to the surface of the ellipsoid passing through P;
- **The geodetic longitude** λ, i.e. the angle in the equatorial plane between the meridian of reference and the meridian plane passing through P;
- **The geodetic height** h, measured along the normal of the ellipsoid passing through P. [12]

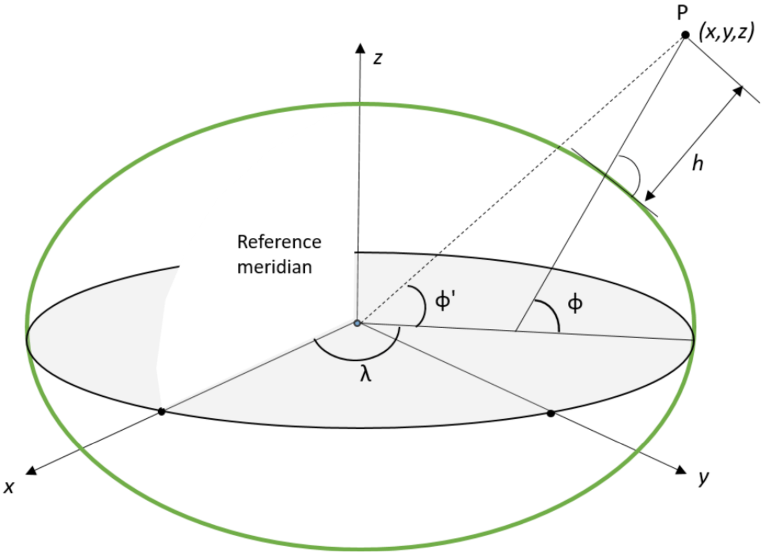


Figure 2.2: Ellipsoidal coordinates (φ,λ,h) and Cartesian coordinates (x,y,z).

In addition to a position reference system, a temporal reference system is also required because GNSSs need temporal measurements to obtain a given point on the earth's surface. These measurements, however, need synchronisation at the nanosecond level in order to have a calculation error of a few metres.
For example, if an operator were to multiply the time difference by the speed of light, even a microsecond difference in the satellite clock would result in a 300-metre error.

Once the user has defined these two reference systems, he must obviously define the station or stations he must rely on to perform the positioning.

In this case he will have the "reference system for stand-alone positioning", the "reference system for NRTK positioning" case and the "reference System for a Base-Rover positioning in post-processing" case. The case dealt with in this paper is the Reference system for stand-alone positioning, i.e. positioning without external aid. In this case, the Reference System is the one in which the coordinates of the satellites are available; to date, the International Terrestrial Reference Frame (ITRF14) system is used, and in particular a realisation of it called IGS14, which is aligned to it. This system has been active since 29 January 2017 and is materialised by a network of 252 stations spread across the globe. It is a slightly different system from the one commonly referred to for GNSS positioning, which is WGS84. The difference between the two can be considered to be in the centimetre range on average.

2.4 GNSS signals

The satellites transmit a signal called SIS (Signal In Space) transmitted via CDMA. This enables the receiver to identify and distinguish each satellite. In particular, each satellite transmits a combination of signals, depending on the frequency allocation and services provided. This combination varies according to its type; for example, military codes are encrypted to avoid spoofing attacks on this type of signal. [12]

In particular, satellite positioning systems can be broken down into 3 segments:

- **Control Segment**; here, a series of ground bases have the task of continuously monitoring the orbit of the individual satellites, setting any course corrections and transmitting the ephemeris (i.e. the list of the spatial coordinates of the satellites, and other time-varying elements, relating to the positions assumed at constant time intervals) [16] of the individual satellites for the next 24 hours. Once the bases have received the required signals from the satellites, they respond by sending commands and data.
- **Space Segment**; in this case the satellites navigating around the Earth constantly send signals to the users' GPS receivers and to the ground control bases.
- **User Segment**; i.e. all the GPS signal receivers that the user uses to locate his position. [5, 2]

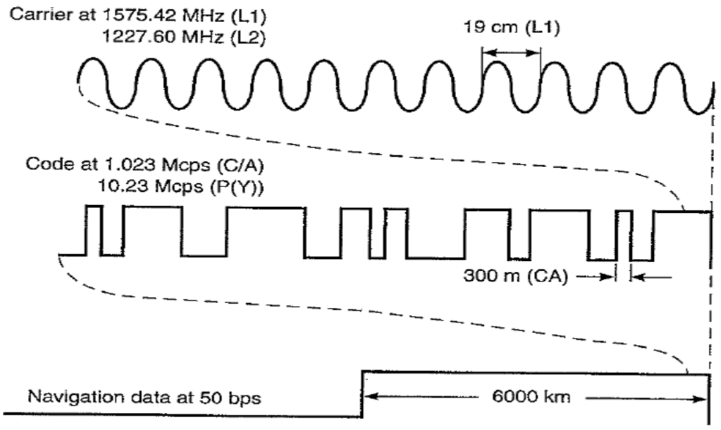


Figure 2.3: GPS satellite signals, containing the carrier signal (top), the code signal (middle) and the navigation data (bottom).

Obviously, the user will not always have all the information he or she wants, and in a correct manner, as there are factors that can adversely affect the quality of the signals he or she receives.

Errors can be:

- **Atmospheric and Ionospheric in origin**; in this case, the speed of light in the atmosphere (used to calculate the radius of the sphere in which a satellite orbits) can be affected by the presence of electrical activity in the ionosphere and the presence of water vapour in the troposphere. These atmospheric disturbances change the speed of the signal, increasing the error in radius calculation; this error is not detected by normal receivers but by differential correction systems such as WAAS (USA) or EGNOS (Europe);
- **Multipath**; here the presence of extended flat surfaces (such as rock faces, extended walls or lakes) in the vicinity of the receiver can be a source of error as the signal may be reflected on these reaching the receiver after traveling a longer path than the actual distance to the satellites. To reduce the error, the user should distance himself from these areas;
- **Sky Cover**; thick forests, thick and dense cloud cover, heavy snowfall or enclosed environments (such as caves, buildings and gorges) can reduce or cancel out satellite signals that should reach the receiver, worsening the quality of positioning or making it impossible. In order to avoid this situation, the user should move to areas with greater visibility in order to improve the signal received and add more satellites to those visible;
- **Orbital or Ephemeris errors**; ephemeris is defined as the list of spatial coordinates of satellites and other time-varying elements, relating to positions assumed at particular times at constant time intervals. Therefore, orbital or ephemeris errors are errors in the estimation of satellite positions due to imperfect modelling of the forces acting on the GPS satellites. Usually an ephemeris error is of the order of 2 to 5 metres.
- **Incomplete or too old ephemerides**; this error occurs when the receiver has been switched off for a long time or was in a place far away from where it is switched on again. Once switched back on, it will take, in the best case, 12.5 minutes to reacquire the ephemerides of all the satellites. As long as the receiver does not have all the new ephemerides the positioning will be affected by errors that will tend to reduce as time passes and the new information is acquired. [5, 17, 18]

2.5 Global Positioning System (GPS)

GPS is a satellite navigation and positioning system originally developed for military purposes and later used in everyday electronic equipment that is becoming an essential part of the global information infrastructure and has endless applications in every sector of the world, from precision agriculture to disaster relief.

The GPS system consists of a series of artificial satellites that, while in orbit, transmit packets of timed information describing their position in space to a mobile terminal or GPS receiver.

Localization takes place through the transmission of a radio signal by each satellite and GPS receivers use these messages to determine the time it takes for the signals from each satellite to reach the receiver. Multiplying this time by the speed of light gives the distance between the receiver and each satellite. These measurements can be combined to determine the position of the receiver and the time of day. Its current degree of accuracy is of the order of a few metres and depends on weather conditions, the availability and position of satellites relative to the receiver, the quality and type of receiver, the effects of relativity and the effects of radio signal propagation in the ionosphere and troposphere. [19, 20, 6]

GPS, therefore, can be divided into three main parts that work in synergy to achieve PNT results:

- The first part relates to space and consists of a constellation of satellites operating in six separate orbits around the Earth. These satellites transmit signals to GPS receivers with the position and time of each satellite;
- The second part is related to control and consists of the ground monitoring/-control stations that keep the satellites in space;
- The third and final user-related part consists of a GPS receiver that picks up the signals from the satellites and calculates its position and 3D time.

In GPS, the satellite signal is a binary phase shift keying (BPSK) modulation of a bit sequence, rectangular in shape. The code sequence can be generated by a linear feedback shift register of maximum length. The need for efficient autocorrelation and low cross-correlation is imperative to distinguish transmissions from different satellites. To do this, GPS will use a family of codes dubbed 'Gold Codes'. Each GPS satellite transmits two classes of signals, one encrypted for the military and another for civilians. These signals are called legacy signals. [21, 22, 23]

Chapter 3

Spoofing

3.1 Smartphone vulnerability to simplistic spoofing attacks

With the rapid development of positioning and navigation technology based on the GNSS, mass market applications have significantly increased with a demand for more accurate and reliable services. According to Strategy Analytics, global smartphone shipments increased by 5% in 2021 compared to last year. Apple topped the global smartphone market with 22%, Samsung was second with 20%, and Xiaomi was third with 12%[24]. Moreover, the European Union Agency for the Space Programme (EUSPA) market report 2022 forecasts that by 2031, more than 10 billion GNSS devices will be in use across the world and currently global smartphone and wearable sales contribute to roughly 92% of global shipments. In May 2016, Google announced the availability of raw GNSS data starting with Android 7. For the first time, developers could access carrier and code measurements and decoded navigation messages from mass market devices. GNSS raw data support is mandatory for devices running Android 10 (API level 29) or higher. For Android 9 (API level 28) and below, GNSS raw data measurement support is mandatory for all Android devices with 2016 or later hardware. Currently, 84% of existing Android phones have raw data measurements[25]. There are several benefits to using raw GNSS data on smartphones. Their use can lead to higher GNSS performance by opening the door to more advanced GNSS processing techniques previously reserved for professional GNSS receivers. These benefits have been demonstrated by Code Based Positioning, Code Aided Positioning, Differential Positioning, and Precise Point Positioning [26], [27],[28]. Although the position computed from raw GNSS data may not be as optimal as a typical chipset output under normal conditions, the use of raw GNSS data can lead to improved solution accuracy in certain cases when external corrections are applied. However, due to the weakness of GNSS signals, GNSS receiver performance can be easily disrupted by anthropogenic interference, with jamming and spoofing activities being critical threats. Swept-frequency jamming is a typical intentional Radio Frequency Interference (RFI) that can be emitted by Personal Privacy Devices (PPDs) with a carrier frequency that varies across GNSS bands. Spoofing, on the other hand, refers to the transmission of spoofed GNSS signals with the intent of creating a false position at a victim's receiver without interfering with the operation of the GNSS receivers. The countermeasures for jamming and spoofing threats have been extensively discussed and proposed in the literature[29]. In recent years, several researchers have developed different techniques to measure spoofing attacks. A GNSS satellite simulator was utilized for the first time in a GNSS spoofing experiment in [30] and in [31] researchers investigated the practical aspects of a satellite lock takeover, where a victim receives spoofed signals after first being locked onto legitimate GPS signals. In [**Warner**], researchers at the University of Texas at Austin developed a portable low-cost GPS intermediate spoofer. Similarly in [32] a commercial super yacht was successfully spoofed with intermediate spoofing techniques, with it being one of the most well-known assessments of the threat of spoofing. A number of techniques have been developed since then to mitigate spoofing for GNSS security. The work in [33] developed advanced spoofing detection and mitigation techniques at various levels of signal processing in the receiver. In the near future, GPS Chips-Message Robust Authentication(Chimera) [34] and Galileo Open Service Navigation Message Authentication (OSNMA) [35] services will allow receivers to be resilient against counterfeit signals with both proposed to be fully operational by the end of 2022. At the present, it is worth examining the potential effects of intentional interference on the low-cost GNSS units embedded in Android smartphones as well as assessing the resilience of the receiver itself. In [36], Unicorn Team showed the spoofing technology using MATLAB® to Record GPS signal by a USRP™ B210 and Replay the signal by an SDR BladeRF™ to spoof PVT of a smartphone. [37] demonstrated a simple methodology to spoof the navigation solution in the phone using software radios and additional equipment. In [38, 39], the impact of spoofing attacks on mobile phones is analyzed and specific techniques are suggested to enhance security such as the use of cheap acceleration sensors. In [40], inertial navigation sensors such as magnetometer, accelerometer, and barometer are used for triggering possible spoofing event detection in smartphones. Despite the proposed solutions, there is as yet no inherent proven defense against GNSS spoofing in Android smartphones, nor is there an exhaustive study on the effects of just simplistic spoofing on them. To replicate a realistic spoofing attack in an everyday scenario, a portable low-cost spoofer has been developed, based on an open source signal generator and low-cost electronics and radio-frequency equipment. The authors made a general assessment of the vulnerability of GNSS receivers integrated with consumer devices to such a simplistic spoofing attack in

[41] but it is comprehensively seen that the Android smartphones under the test were resilient. This work is in continuation with those previous results, testing a wider variety of Android smartphones from relevant manufacturers and test campaigns are designed to define the limits of the smartphones against such a simplistic attack.

3.2 Spoofing

Spoofing techniques are diverse, the best known and most widely used being:

- IP spoofing;
- DNS spoofing;
- ARP spoofing;
- Web Spoofing;
- SMS Spoofing;
- E-Mail Spoofing;
- GNSS Spoofing; [42]
- GPS Spoofing. [43]

In this thesis, the last two items in the above list will be discussed in more detail. In this case, spoofing refers to the process in which someone (or something) tries to change the position reported by a device by providing an incorrect PVT to a local user or remote client, making them believe they are in a false position. The attacker will use an RF transmitter (or spoofer) near the receiver to make it intentionally calculate the previously defined wrong position. The radio interference produced by this device can overpower the weak signals coming from the satellites, causing them to be lost and potentially the loss of positioning. [44, 45, 46]

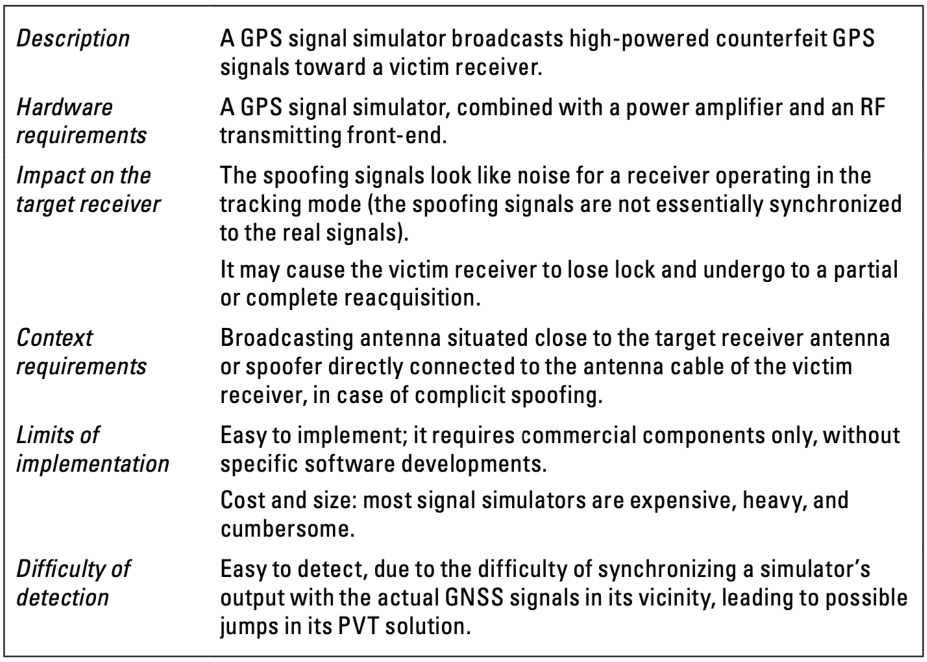
3.2.1 GNSS Spoofing

GNSS Spoofing is a type of radio interference that occurs when weak GNSS signals are overpowered by stronger radio signals on the same frequency causing a loss of accuracy and positioning. [47]

Spoofing attacks from a GNSS perspective can be divided into three types:

- Simplistic attacks;
- Intermediate (or shadowed) attacks;
- Sophisticated attacks. [48, 49, 42]

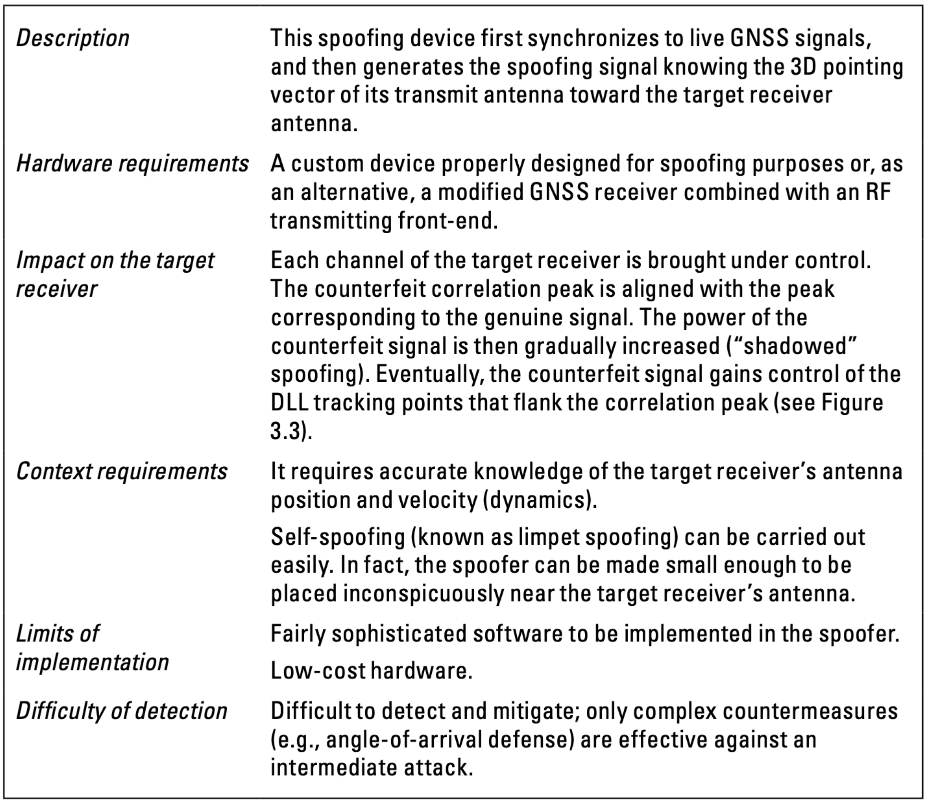
Regarding the first category, a GNSS signal simulator is typically coupled with an RF transmission front-end and is employed to imitate authentic signals. Here, the spoofer is able to generate GNSS signals that possess a higher power but are not perfectly coherent and synchronised with the authentic signals; in fact, the counterfeit signals could show satellites that are not visible in the target receiver's position, show discrepancies in the navigation messages and show non-negligible residual modulation due, precisely, to the lack of synchronisation. Therefore, this type of attack can be detected and addressed with simple countermeasures. [42]



Description	A GPS signal simulator broadcasts high-powered counterfeit GPS signals toward a victim receiver.
Hardware requirements	A GPS signal simulator, combined with a power amplifier and an RF transmitting front-end.
Impact on the target receiver	The spoofing signals look like noise for a receiver operating in the tracking mode (the spoofing signals are not essentially synchronized to the real signals). It may cause the victim receiver to lose lock and undergo to a partial or complete reacquisition.
Context requirements	Broadcasting antenna situated close to the target receiver antenna or spoofer directly connected to the antenna cable of the victim receiver, in case of complicit spoofing.
Limits of implementation	Easy to implement; it requires commercial components only, without specific software developments. Cost and size: most signal simulators are expensive, heavy, and cumbersome.
Difficulty of detection	Easy to detect, due to the difficulty of synchronizing a simulator's output with the actual GNSS signals in its vicinity, leading to possible jumps in its PVT solution.

Figure 3.1: Characteristics of a Simplistic Spoofing Attack. [42]

For the second category, the type of spoofing attack becomes more sophisticated than the previous one. In this case, the spoofer consists of a device capable of receiving GNSS signals and generating counterfeit signals. He is now able to extract satellite, time and position information using the signals received. Once this information is obtained, it generates counterfeit signals by exploiting the synchronisation of local codes and carriers. When an intermediate spoofer carries out an offensive action, its signals are able to align themselves in the code phase with the authentic ones, but to do so it must be synchronised within a half code chip and know the relative positions (3D vector) between the target antennas and the spoofer in order to carry out the attack successfully.
In this category, each tracking channel of the target GPS receiver is simultaneously attacked by the receiver-spoofer, which first performs code-phase alignment and then signal lifting. Furthermore, compared to simplistic attacks, intermediate attacks are usually synchronised with respect to authentic signals, so the frequency parameter is negligible. [48, 49, 42]



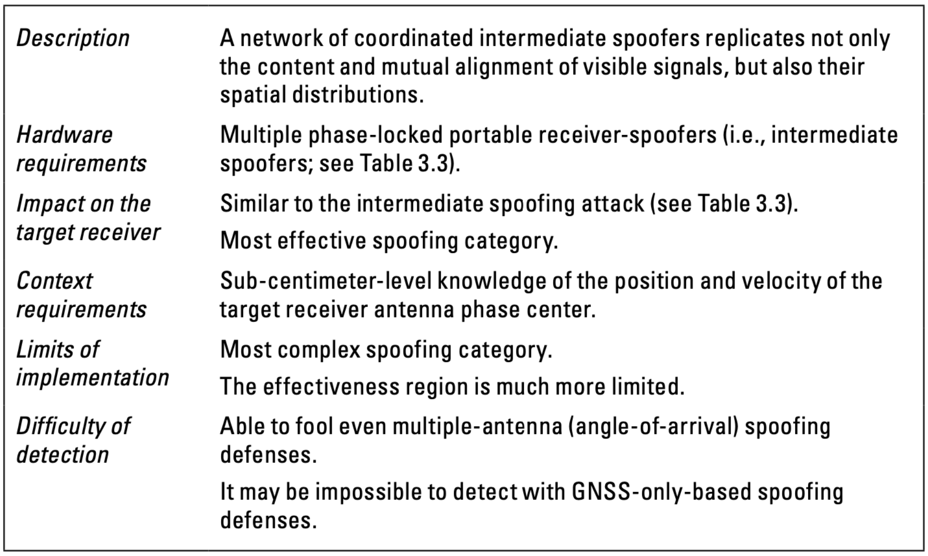
Description	This spoofing device first synchronizes to live GNSS signals, and then generates the spoofing signal knowing the 3D pointing vector of its transmit antenna toward the target receiver antenna.
Hardware requirements	A custom device properly designed for spoofing purposes or, as an alternative, a modified GNSS receiver combined with an RF transmitting front-end.
Impact on the target receiver	Each channel of the target receiver is brought under control. The counterfeit correlation peak is aligned with the peak corresponding to the genuine signal. The power of the counterfeit signal is then gradually increased ("shadowed" spoofing). Eventually, the counterfeit signal gains control of the DLL tracking points that flank the correlation peak (see Figure 3.3).
Context requirements	It requires accurate knowledge of the target receiver's antenna position and velocity (dynamics). Self-spoofing (known as limpet spoofing) can be carried out easily. In fact, the spoofer can be made small enough to be placed inconspicuously near the target receiver's antenna.
Limits of implementation	Fairly sophisticated software to be implemented in the spoofer. Low-cost hardware.
Difficulty of detection	Difficult to detect and mitigate; only complex countermeasures (e.g., angle-of-arrival defense) are effective against an intermediate attack.

Figure 3.2: Characteristics of a Sophisticated Spoofing Attack. [42]

For the third and last category, we have a more complex spoofing device consisting of multiple, synchronised and coordinated receivers. In this case, co-ordinated receivers are capable of generating and transmitting spoofed signals as in intermediate spoofers, but unlike these, they possess 3D position information at the sub-centimetre level about the phase centres of their antennas and the phase centre of the target antenna; therefore, they can easily defeat complex countermeasures (such as angle-of-arrival defence) by relying on the constructive properties of their RF signals and suppress authentic signals at the target receiver's antenna. Therefore, this type of attack could be effectively countered by using hybrid solutions (exploiting data from other sensors) or by using a cryptographic defence (such as signal authentication).

Unlike the two previous types, sophisticated spoofers have negligible modulation due to their complexity, allowing them to have accurate synchronisation between counterfeit and authentic signals.

Although they are very effective, their development and deployment becomes challenging due to the complexity associated with the development of this type of spoofer and the need for sub-centimetre-level knowledge of the target receiver's antenna. Therefore, the probability of a coordinated attack with sophisticated spoofers is particularly low. [48, 49, 42]



Description	A network of coordinated intermediate spoofers replicates not only the content and mutual alignment of visible signals, but also their spatial distributions.
Hardware requirements	Multiple phase-locked portable receiver-spoofers (i.e., intermediate spoofers; see Table 3.3).
Impact on the target receiver	Similar to the intermediate spoofing attack (see Table 3.3). Most effective spoofing category.
Context requirements	Sub-centimeter-level knowledge of the position and velocity of the target receiver antenna phase center.
Limits of implementation	Most complex spoofing category. The effectiveness region is much more limited.
Difficulty of detection	Able to fool even multiple-antenna (angle-of-arrival) spoofing defenses. It may be impossible to detect with GNSS-only-based spoofing defenses.

Figure 3.3: Characteristics of an Intermediate Spoofing Attack. [42]

GPS Spoofing

Today, GPS-dependent systems are ubiquitous in positioning and navigation applications and most vehicles and mobile phones are equipped with navigation and positioning systems using GPS.
In public transport, land, air and sea, rescue and police services, telecommunication systems, electronic system tracking or asset tracking, secure and safe GPS applications are receiving increasing attention. [50, 19, 51]

Although GPS bring many advantages, their wide use makes them attractive targets for attacks by hackers. GPS signals, due to a particular weakness in signals transmitted on wireless channels, are vulnerable to in-band interference; therefore, even low-power interference can easily disrupt or falsify GPS receivers within a radius of several kilometres.
Moreover, GPS is a technology whose signal structure is in the public domain and therefore more susceptible to interference. [50, 52, 53]

A spoofing attack, for instance, could effectively deceive a GPS receiver that monitors the activities of a particular vehicle or convoy into recording a different trajectory from the one it is/is actually taking. [52, 54]

GPS spoofing is based on intentional interference that aims to make GNSS receivers generate false positions or navigation routes. [55, 56]
This is done by imitating real GPS signals in order to mislead the target receiver without him knowing what is happening. Therefore, if a navigation system relies on GPS to determine its position, GPS spoofing offers a way to take control of it. [50, 53]

GPS spoofing does not attack the receiver itself, but aims to provide bogus inputs to induce the receiver to report incorrect information. [57, 21, 54]
This attack technique involves two steps: the acquisition of the legitimate GPS satellite signal and the subsequent transmission of the spoofing signal. [53, 21]

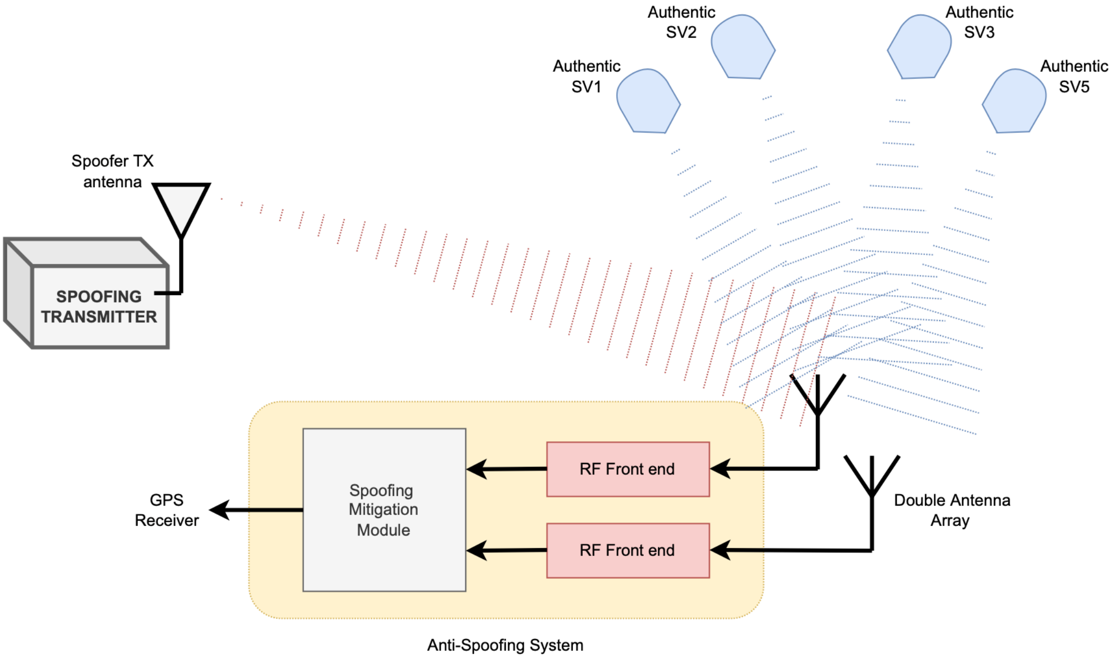


Figure 3.4: How GPS Spoofing works. [54]

GPS spoofing signals can be generated in three different ways:

- **GPS signal simulator**.
  Here, the simulator is tightly coupled with an RF front-end to imitate authentic GPS signals. Once the signals are generated, however, they will appear as noise to a receiver operating in tracking mode because they are not perfectly synchronised with the real GPS signals. Although there is this weakness, spoofing signals still have the potential to fool commercial GPS receivers, particularly if the strength of the counterfeit signal is higher than that of the authentic signals. Amplitude monitoring, inter-measurement consistency checks and inertial measurement unit (IMU) consistency checks would be sufficient to detect these types of signals.

- **Receiver-based spoofers**.
  In this category, a GPS receiver and a spoofing transmitter first synchronise with the current GPS signals by extracting all the most important and sensitive information (such as the satellite's position) and then, knowing the 3D pointing vector of its transmitter antenna towards the target receiver's antenna, generates a spoofing signal. Unlike the previous category, these spoofers are more difficult to detect, even if the receivers were in tracking mode, since they are synchronised with the real GPS satellites; it is also more

difficult to distinguish real signals from spoofed ones. The weaknesses of receiver-based spoofers are:

  - they need to project the spoofing signals to the intended victim receiver with the right delay and signal strength;
  - the spoofing power needs to be slightly higher than that of the authentic signal in order to deceive the target receiver;
  - they need to align the carrier frequency and phase to the authentic GPS signals;
  - they need to minimise the auto-jamming effect;
  - they need to suppress the relative latencies of the data bits.

- **Sophisticated receiver-based spoofers**.
  These types of spoofers are among the most complex because manifold array synchronisation and carrier phase alignment can only be achieved for a very small region where the target receiver antennas are located. To operate, they need to know the centimetre position of the phase centre of the target receiver antenna in order to perfectly synchronise the code and carrier phase of the spoofing signal with those of the authentic signals at the receiver [7]. In addition to being among the most complex, they are among the most effective in that they are resistant to arrival direction antispoofing techniques; this is due to the fact that they are able to exploit different transmit antennas by synthesising an array manifold consistent with the array manifold of the authentic signal, thus defeating an angle-of-arrival (AOA) discriminating GPS receiver. Although very effective, its realisation is very difficult if not impossible due to the geometry and movement of the target receiver's antennas, which imposes physical limitations on the positioning of the spoofer's antenna with respect to the antennas of the receiver to be deceived. [50, 55, 58, 59]

Not all GPSs are the same; in fact, GPS receivers work with three different levels of operation concerning signal processing, data bits and the navigation/positioning solution. These three levels, of course, have vulnerabilities:

- **Vulnerability at the signal processing level**.
  The factors favouring this weakness are:

  - the backwards-compatible GPS technology whose signal characteristics do not particularly change in different generations of GPS satellites;
  - the knowledge of GPS signal type, modulation type, pseudo-random noise signals (PRN), transmission frequency, Doppler range, signal power and signal bandwidth are known;
  - the presence in most commercially available GPS receivers of an automatic gain control (AGC) block, which can increase the vulnerability of GPS receivers to higher power spoofings signals, as it automatically adjusts the receiver input gain according to the stronger spoofing signals. [60]

  Therefore, knowing the operating basis and general structure of a civil GPS receiver, an attacker can generate counterfeit signals that are very similar to authentic GPS signals in order to deceive GPS receivers. [50]

- **Vulnerabilities at the data bit level**.
  The framing structure of GPS signals and the navigation frame are in the public domain and do not change rapidly; for example, satellite ephemeris information can be acquired in less than a minute, but remain unchanged for 12.5 minutes. [52]
  During this time, the spoofer can produce the GPS data frame; furthermore, it is possible to induce the receiver to reject valid satellite signals due to a manipulation of the satellite health bits. [50, 61]

- **Vulnerabilities in navigation and position resolution**.
  Counterfeit pseudorange measurements can be introduced into the receiver through spoofing.
  These will lead to an incorrect position, velocity and time (PVT) solution. [50]

Chapter 4

Devices Used in Experimental Spoofing Campaign

For the experimental phase of this project, various materials and equipment provided by Politecnico di Torino and the Italian Army were used, which will be described in the following chapter.

4.1 Raspberry Pi

Raspberry Pi is a complete hardware device packed into a single board combined with a programming platform. It is a miniature computer, an entire hardware ecosystem collected in a single board that has been on the market since 2012, produced to carry out robotics projects and research at an amateur and professional level. It is a complete device, costs little, is small (the size of a non-scientific pocket calculator) and is easy to configure. The board is pre-configured, but modules can be added to increase and customise its functionality according to the type of use. The Raspberry Pi's main configuration has a high level of connectivity; it includes a Wi-Fi LAN port, USB and HDMI inputs, 1 SD card slot (which acts as a hard disk), a stereo jack audio output, with a micro USB port for power, and GPIO (i.e. a long black base with little teeth that completes the connectivity picture by offering the possibility of connecting to multiple electrical devices). Depending on the model, there are many other connectors, to connect external devices such as LCD displays and even webcams. The board supports the Linux operating system, and of course its own Raspberry Pi OS platform. With Raspberry Pi, one can realise all kinds of projects. With this mini PC, one can create video games, sound systems, computers and 3D printers, but also media centres and smartphones. It is also possible to use the board for smart home and robot management. Various versions of Raspberry Pi can be found on the market, each of which offers specific functionalities and predispositions. For this paper, the device was used with the following features : There are various versions of Raspberry Pi on the market and for this work the 'Raspberry Pi 4 Model B' was used, which has:

- 2 USB 2.0 ports and 2 USB 3.0 ports;
- 1.5 GHz quad core processor (CPU speed);
- Gigabit Ethernet port (1000Base-T);
- 4 GB RAM memory;
- Dual-band wireless LAN (802.11ac/n);
- Bluetooth 5.0;
- MicroSD 32GB;
- Micro HDMI ports (Supporting 2 x 4K displays);
- USB-C Power Supply port;
- Card reader;
- Cooling fan. [62, 63, 64, 65]

Raspberry Pi was used to 'build' the spoofing signal and in particular to generate the different satellites with the fake coordinates that will replace the real coordinates where the GPS-equipped device is located.

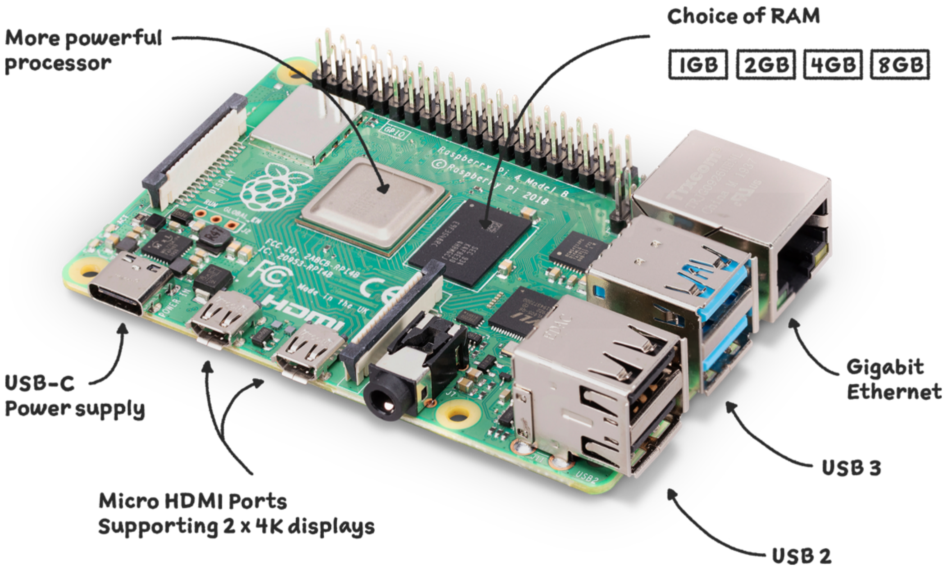


Figure 4.1: Raspberry Pi 4 Model B. [62]

4.1.1 General Purpose Input Output (GPIO) Lines

Raspberry Pi boards offer a very wide variety of uses, partly due to the presence of 40-pin male connectors (Raspberry Pi 1, 2, 3, zero) that connect the board to different types of electronic circuits. This possibility is provided by a series of GPIO lines, which can also be programmed in Python. GPIO (General Purpose Input Output) lines are the connections between the processor and the board's connector pins. These lines can be programmed to act as input ports or output ports. Some of these lines have additional functionality; they can be configured as UART, SPI or I2C lines. To understand the location and naming of the GPIO lines in the Raspberry Pi's 40-pin connector, the pinout command must be used. Using the LxTerminal and entering the command $pinout head in a few moments, you will have the board's peripherals displayed on the terminal, specifically the 40-pin connector. Programming the IO ports can be done using various programming languages, but it is advisable to use Python language as it is the most widely used language on this platform. [66]

4.2 FileZilla

FileZilla Client is a free cross-platform software that allows file transfer over the network via the FTP (File Transfer Protocol) on GNU/Linux, Microsoft Windows, and Mac OS X. Among the various protocols supported, in addition to FTP are SFTP, and FTP over SSL/TLS.
FTP is a protocol for the transmission of data between hosts based on TCP and with a client-server architecture.
In telecommunications and computing, the Transmission Control Protocol / Transfer Control Protocol (TCP) is a transport layer packet network protocol that deals with transmission control, making data communication on the network between sender and receiver reliable. The main features of this programme are:

- the site manager, which allows the user to create a list of FTP sites and select one of them with a drop-down menu. Once the desired site has been selected, the programme will connect to the site itself, allowing files to be uploaded and downloaded;
- the message log, which is present at the top and contains the list of all messages sent to the servers by the programme and the corresponding replies received from the FTP servers;
- the file and folder view, which is the graphic interface associated with the file transfer engine. Located below the message log and composed of two windows of equal size, it allows the user to drag and drop, i.e. navigate through the system's folders (left-hand side) and drag them to the other side (right-hand side) to transfer the selected files or folders to the desired server;
- the transfer queue, located at the bottom of the screen, consists of a red and a green light and indicates the download or upload transmission speed. [67, 68]

In this project, the programme is used to obtain the daily satellite data to be used to generate the spoofing signal.

4.3 MATLAB

MATLAB is a programming and numerical calculation platform that combines a desktop environment optimised for iterative analysis and design processes with a programming language that expresses mathematical operations with matrices and arrays in a straightforward manner. MATLAB applications allow users to see how different algorithms work synergistically with data until they achieve the desired results, and then generate a MATLAB program to reproduce or automate the desired work. The MATLAB platform is optimised for solving technical and scientific problems, MATLAB's matrix-based language is the most natural way to express computational mathematics, and the built-in graphing capabilities simplify data visualisation and analysis. In addition, MATLAB's code can be integrated into other languages to allow algorithms and applications to be implemented directly from web, enterprise and production systems. Today, this program is used by millions of engineers and scientists worldwide for data analysis, algorithm development and modelling.MATLAB underpins active safety systems in cars, space vehicles, health monitoring devices, smart power grids and LTE cellular networks. It is used for machine learning, signal and image processing, computer vision, communications, computational finance, control design, robotics and much more.

Key features of the program include:

- High-level language for technical and scientific processing;
- Customised desktop environment for exploration, design and troubleshooting;
- Graphs for visualising data and tools in order to create customised representations;
- Applications for curve fitting, data classification, signal analysis, control system synchronisation and many other tasks;
- Complementary toolboxes for a wide range of technical and scientific applications;
- Tools for creating applications with customised user interfaces;
- Interfaces for C/C++, Java, .NET, Python, SQL, Hadoop and Microsoft Excel;
- Royalty-free implementation options for sharing MATLAB programmes with end users. [69, 70]

For this elaborate MATLAB is used to generate the code that is able to automatically load the spoofing code onto the HackRF One device without any input from the operators.

4.4 Python

Python is a high-level object-oriented programming language known for its clarity, power and flexibility and is widely used for developing distributed applications, scripting, numerical computation and system testing.
It is a language that is quick to learn, understand and use, with a clean and uniform syntax; in fact, the philosophy behind its creation focuses primarily on code readability and maintainability.

It is a multi-paradigm language that aims to be dynamic, simple and flexible; moreover, it is an interpreted language, which means that an interpreter reads and executes the code directly, without compilation. Although Python is considered an interpreted language, the source code is not actually converted directly into machine language. In fact, it first passes through a pre-compilation phase in bytecode, which is almost always reused after the first execution of the programme, thus avoiding reinterpreting the source each time and improving performance. The Python interpreter also supports an interactive mode of use (REPL) through which code can be entered directly from a terminal and the result immediately displayed.

Python supports the object-oriented paradigm, structured programming and many functional programming and reflection features. Among Python's most immediately recognizable features are untyped variables and the use of indentation for specification syntax instead of the more common parentheses, the overloading of operators and functions via delegates, the presence of a rich assortment of basic types and functions and standard libraries, advanced syntaxes such as slicing and list comprehension. Python also has a graphics section, the Python Turtle Graphics module, which allows lines of code to be applied to graphics, and it is possible to enhance and extend its functionality through third-party extensions. [71, 72]

The most important qualities, therefore, of this programme are:

- It is multi-paradigm;
- It is portable;
- It is easy to use;
- It has many libraries;
- It is very performant;
- It manages memory automatically;
- Can be integrated with other languages. [73]

4.5 Xiaomi Mi 8

Xiaomi Mi 8 smartphone is used for satellite data collection. The Xiaomi Mi 8 is used because it is the first dual-band GPS smartphone capable of supporting dual-frequency positioning and dual-band signal cooperation, improving location accuracy in urban and non-urban environments. Thanks to these two factors, the device is able to reduce the positioning error from 5 metres to 30 centimetres. Dual-frequency positioning consists of a dual-frequency GPS that uses two frequency bands instead of one; in particular, this smartphone uses the L1 / E1 and L5 / E5 bands (generally the bands used for civil GPS are L1 / E1, L2 / E2 and L5 / E5). The L and E preceding the band number represent GPS (of US origin) and GLONASS (of European origin) respectively. In order to use a dual-frequency GPS, the device must have supporting hardware, and to do so, the Xiaomi company used the Broadcom BCM4775 chipset, making its dual-frequency GPS compatible. This chip uses a microcontroller-sensor hub combination with integrated GNSS that will allow it to receive GPS L1 C/A, GLONASS L1, BeiDou (BDS) B1 - QZSS L1, Galileo (GAL) E1 - GPS L5, Galileo E5a and QZSS L5 signals. The BCM47755 chipset also allows for greater multipath resistance, greater resistance to reflections, greater immunity to interference and noise, and reduces errors that can be caused by other sources (such as electrons in the ionosphere). All these factors result in greater positional accuracy, due to a reduction in accuracy deviation, and improved navigation capability. [74]

Figure 4.2: BCM47755 Chipset. [75]

4.6 GnssLogger

One of the latest frontiers in GNSS is precision positioning from Android devices. One of the first apps developed with this in mind was released by Google in May 2016, which allowed developers access to GNSS raw data for smartphones running Android 7 or later.
This paved the way for the development of several apps to record and process this data, allowing the user to be able to analyse its quality and understand what the major sources of error might be.
Currently, Google's GnssLogger allows in-depth analysis and logging of all types of position and sensor data such as GPS (Global Positioning System), network position and other sensor data. [76]

This application has the following features:

- **Home Tab**, which aims to control various data records such as raw GNSS measures;
- **LOG Tab**, which:
  - displays all position and raw measurement data;
  - allows offline logging control using 'Start Log', 'Stop and Send' and 'Time Log';
  - allows access to specific items using the corresponding switches on the Home Tab;
  - Deletes existing log files from the disk;
- **Map Tab**, which:
  - allows you to display on GoogleMap, the position provided by the GPS chipset, Network Location Provider (NLP), Fused Location Provider (FLP) and calculated Weighted Least Square (WLS) position;
  - Toggles between different map views and location types;
- **Plot Tab**, displaying CN0 (signal strength), residual PR (pseudorange) and residual PRR (pseudorange) versus time;
- **Status Tab**, which displays detailed information on all visible GNSS satellites such as GPS, Beidou (BDS), QZSS, GAL (Galileo), GLO (GLONASS) and IRNSS;

- **Skyplot Tab**, which:
  - displays data from all visible GNSS satellites using a sky-plot;
  - displays the average CN0 of all satellites in view and those used in fixes;
- **AGNSS Card**, which aims to test the functionality of the assisted GNSS;
- **WLS Analysis Tab**, which shall;
  - display the weighted least square position, velocity and associated uncertainties calculated from the raw GNSS measurements processed;
  - compare the WLS results with the values reported by the GNSS chipset. [77]

4.7 GPSLogger

GPSLogger is an app from early 2009 that uses the GPS capabilities of the Android phone to record coordinates in GPS format files at regular intervals. This app is used to record the user's position and route, record the operator's journeys, note landmarks along the way, display position, speed, orthometric height (altitude above sea level), direction and many other statistics on a given route.
It is, therefore, a basic and lightweight GPS tracker that focuses on accuracy, with a focus on energy saving and that can also work offline (which is why it has no built-in maps). [78]
Once the measurements have been taken, the user has the option of accessing the recorded tracks at any time using the in-app Tracklist, viewing them in their preferred external viewer (which must be installed), saving them and sharing them in GPX, KML and TXT format in many ways.

This application has several real-time settings of the recording process including:

- GPS update interval: specifies the time interval between corrections in order to save battery power. In many Android devices, this setting is a suggested (and not mandatory) interval due to operating system specifications, so the interval between fixes may be different from the one set. If an external Bluetooth GPS receiver is used and it is desired to record at the maximum speed supported by the hardware, the user has the option of setting the update interval to 'Minimum Available'. However, it is not possible to set a GPS update interval of more than 3 seconds, as the GPS must remain active and update frequently to return good accuracy;

- Interval Filter: If enabled, the application starts recording a track point after the specified time interval from the previousl recorded track point. This setting is intended to limit the number of track points recorded in a track. If the user also sets the Distance Filter, the two filters are in OR: the application activates the recording of a track point after the set amount of time or distance travelled. Regardless of the value of this setting, GPS Logger continues to record a track point when the operator stops or starts moving: these points are important for statistics and track display;

- Distance Filter: If enabled, the application starts recording a track point when the distance to the previous recorded point is greater than (or equal to) the set value. If the Interval filter is also set, the two filters are in OR. [79]

GPSLogger also has a system for altitude correction that is applied in real time to all altitudes when viewing and exporting. During acquisition, the App always records the raw data. Therefore, the user has the possibility to activate/deactivate/change the altitude corrections at any time, without corrupting or modifying the recorded data; in particular, an EGM96 correction is used in which the GPS sensor provides an altitude referred to an approximate earth surface (called Ellipsoid). By activating this setting, the application enables an automatic altitude correction based on the NGA Earth Geoid model to calculate the real altitude referred to sea level (Real Earth Geoid). The first time the user activates this option, the application downloads the EGM grid from OSMGeo.org and saves it to a private folder. This is a one-time operation; no further connections are required to keep the correction active. If the device is unable to download the EGM file (e.g. if the smartphone has no Internet connectivity), it is possible to manually activate the altitude correction in which the operator specifies an offset of the overall altitude. The offset is also applied to the flight. [79, 80]

Chapter 5

Work Development and Analysis on Spoofing Vulnerability

This chapter illustrates the simulations performed, the results obtained in a system subject to spoofing interference, and analyses the methodologies for detecting and understanding when the operator is subject to such an attack. The entire simulation and data collection campaign took place within a controlled area, i.e. in the military facility "Ex Military Hospital Alessandro Riberi" (N 45°02’49.3" E 7°39’00.5"). Figure 5.1 shows the view of the complex from above.

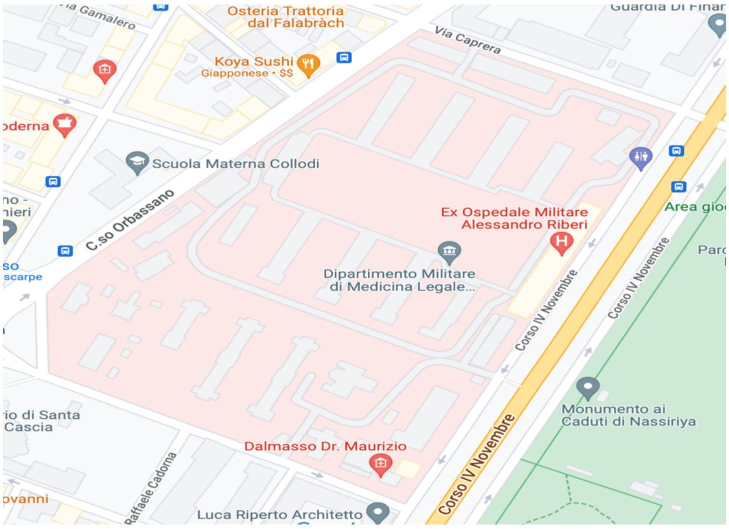


Figure 5.1: Caption

5.1 Data acquisition with spoofing signal in a static situation

To make the user lose his position on his device, the attacker must generate a spoofing signal. In this way, he will be able to disrupt and corrupt the correct reception of the signal by the operator's device. As a first case, a situation will be simulated in which both the attacker and the victim are stationary in their respective positions at a maximum distance of 5 metres. The device illustrated in Figure 5.2 will be used to generate the spoofing signal.

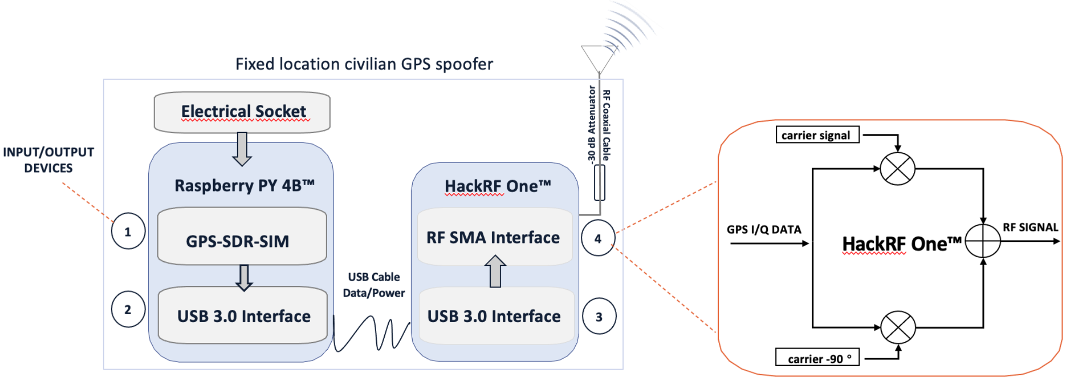


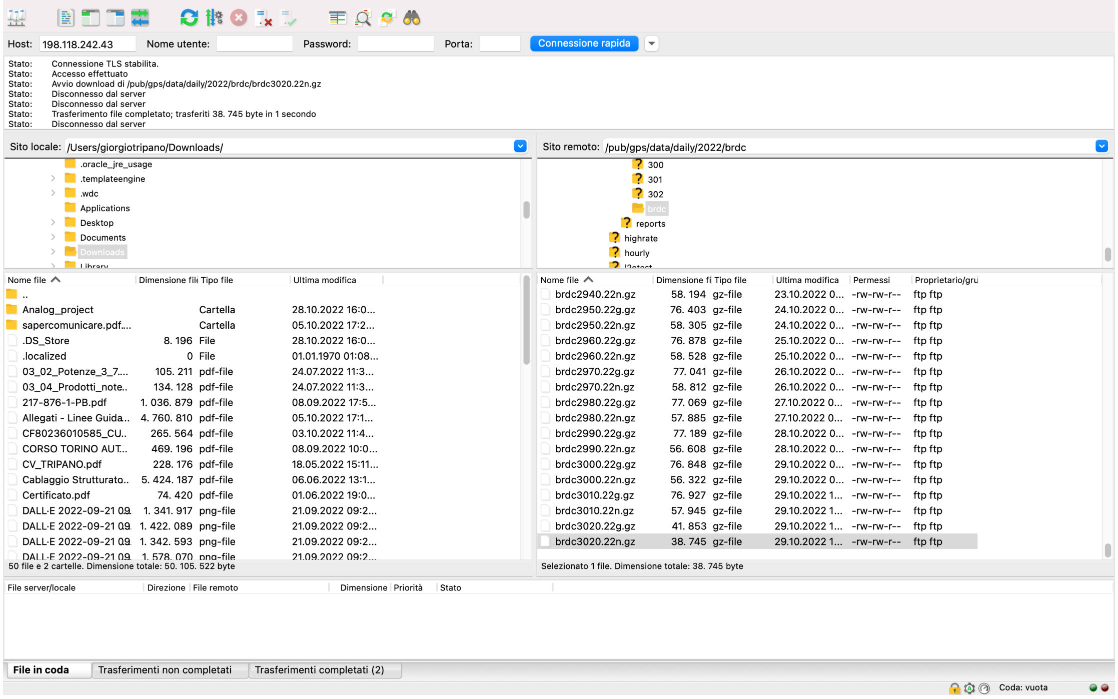
Figure 5.2: High-level schematic of the low-cost fixed location spoofer.

As shown in Figure 5.2, the following equipment was used to emit the spoofing signal

- Raspberry Pi 4 Model B hardware device, connected to a simple electrical outlet;
- HackRF One Device;
- Input/Output Devices (Keyboard and Monitor);
- Antenna;
- Two Xiaomi RedMI 8 Phones.

To emit this signal, it will first have to be 'built'; the Raspberry Pi 4 Model B hardware device is used to do this. The signal is constructed in the following way:

- Access the FileZilla program by downloading the compressed archive gzip brdc(daily.date)n.gz as shown in Figure 5.3;



Host: 198.118.242.43 Nome utente: Password: Porta:

```
Stato:    Connessione TLS stabilita.
Stato:    Accesso effettuato
Stato:    Avvio download di /pub/gps/data/daily/2022/brdc/brdc3020.22n.gz
Stato:    Disconnesso dal server
Stato:    Disconnesso dal server
Stato:    Trasferimento file completato; trasferiti 38. 745 byte in 1 secondo
Stato:    Disconnesso dal server
```

Sito locale: /Users/giorgiotripano/Downloads/

Nome file	Dimensione file	Tipo file	Ultima modifica
..			
Analog_project		Cartella	28.10.2022 16:0...
sapercomunicare.pdf....		Cartella	05.10.2022 17:2...
.DS_Store	8. 196	File	28.10.2022 16:0...
.localized	0	File	01.01.1970 01:08...
03_02_Potenze_3_7....	105. 211	pdf-file	24.07.2022 11:3...
03_04_Prodotti_note..	134. 128	pdf-file	24.07.2022 11:3...
217-876-1-PB.pdf	1. 036. 879	pdf-file	08.09.2022 17:5...
Allegati - Linee Guida..	4. 760. 810	pdf-file	05.10.2022 17:1...
CF80236010585_CU..	265. 564	pdf-file	03.10.2022 11:4...
CORSO TORINO AUT...	469. 196	pdf-file	08.09.2022 10:0...
CV_TRIPANO.pdf	228. 176	pdf-file	18.05.2022 15:11...
Cablaggio Strutturato..	5. 424. 187	pdf-file	06.06.2022 13:1...
Certificato.pdf	74. 420	pdf-file	01.06.2022 19:0...
DALL·E 2022-09-21 09.	1. 341. 917	png-file	21.09.2022 09:2...
DALL·E 2022-09-21 09.	1. 422. 089	png-file	21.09.2022 09:2...
DALL·E 2022-09-21 09.	1. 342. 593	png-file	21.09.2022 09:2...
DALL·E 2022-09-21 09.	1. 578. 070	png-file	21.09.2022 09:2...

50 file e 2 cartelle. Dimensione totale: 50. 105. 522 byte

Sito remoto: /pub/gps/data/daily/2022/brdc

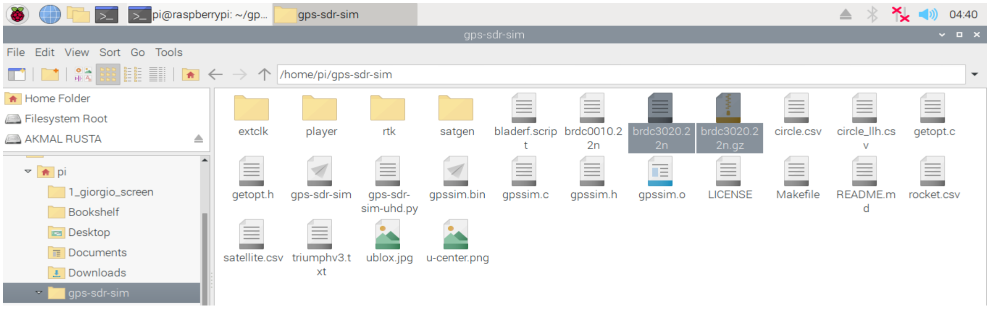
Nome file	Dimensione file	Tipo file	Ultima modifica	Permessi	Proprietario/gru
brdc2940.22n.gz	58. 194	gz-file	23.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2950.22g.gz	76. 403	gz-file	24.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2950.22n.gz	58. 305	gz-file	24.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2960.22g.gz	76. 878	gz-file	25.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2960.22n.gz	58. 528	gz-file	25.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2970.22g.gz	77. 041	gz-file	26.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2970.22n.gz	58. 812	gz-file	26.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2980.22g.gz	77. 069	gz-file	27.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2980.22n.gz	57. 885	gz-file	27.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2990.22g.gz	77. 189	gz-file	28.10.2022 0...	-rw-rw-r--	ftp ftp
brdc2990.22n.gz	56. 608	gz-file	28.10.2022 0...	-rw-rw-r--	ftp ftp
brdc3000.22g.gz	76. 848	gz-file	29.10.2022 0...	-rw-rw-r--	ftp ftp
brdc3000.22n.gz	56. 322	gz-file	29.10.2022 0...	-rw-rw-r--	ftp ftp
brdc3010.22g.gz	76. 927	gz-file	29.10.2022 1...	-rw-rw-r--	ftp ftp
brdc3010.22n.gz	57. 945	gz-file	29.10.2022 1...	-rw-rw-r--	ftp ftp
brdc3020.22g.gz	41. 853	gz-file	29.10.2022 1...	-rw-rw-r--	ftp ftp
brdc3020.22n.gz	38. 745	gz-file	29.10.2022 1...	-rw-rw-r--	ftp ftp

Selezionato 1 file. Dimensione totale: 38. 745 byte

File server/locale	Direzione	File remoto	Dimensione	Priorità	Stato

Figure 5.3: FileZilla program.

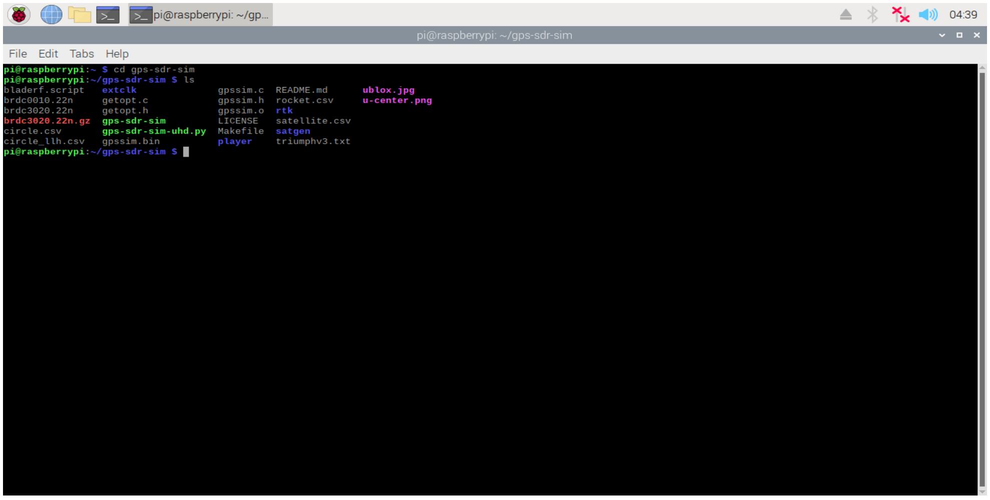
- Download the compressed gzip archive and extract it inside the gps-sdr-sim folder (specifically/home/pi/gps-sdr-sim) as shown in Figure 5.4;



/home/pi/gps-sdr-sim

Figure 5.4: Extraction of the compressed gzip downloaded from FileZilla program.

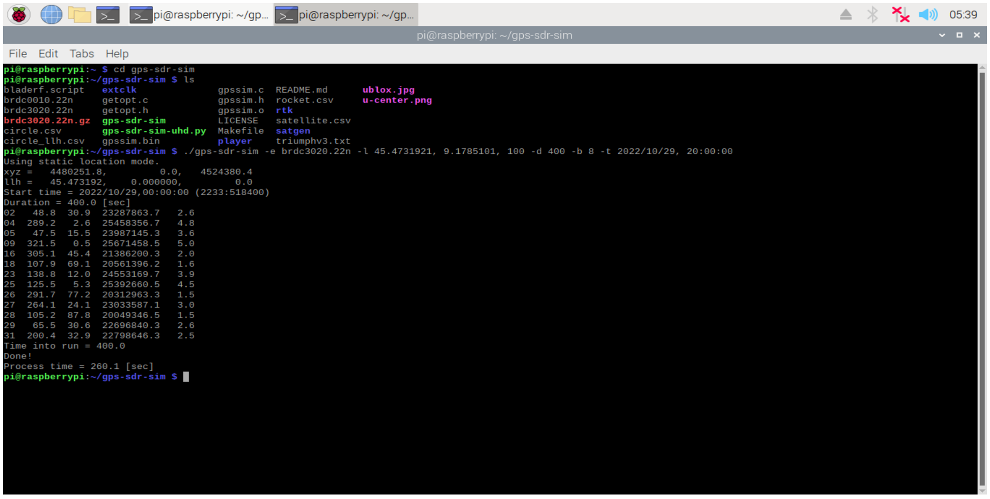
- Once these two steps are done, open the pi@raspberrypi terminal from which you will enter the gps-sdr-sim folder and check the files inside (see Figure 5.5);



```
pi@raspberrypi:~ $ cd gps-sdr-sim
pi@raspberrypi:~/gps-sdr-sim $ ls
bladerf.script   extclk              gpssim.c  README.md      ublox.jpg
brdc0010.22n     getopt.c            gpssim.h  rocket.csv     u-center.png
brdc3020.22n     getopt.h            gpssim.o  rtk
brdc3020.22n.gz  gps-sdr-sim         LICENSE   satellite.csv
circle.csv       gps-sdr-sim-uhd.py  Makefile  satgen
circle_llh.csv   gpssim.bin          player    triumphv3.txt
pi@raspberrypi:~/gps-sdr-sim $ 
```

Figure 5.5: Terminal of pi@raspberrypi.

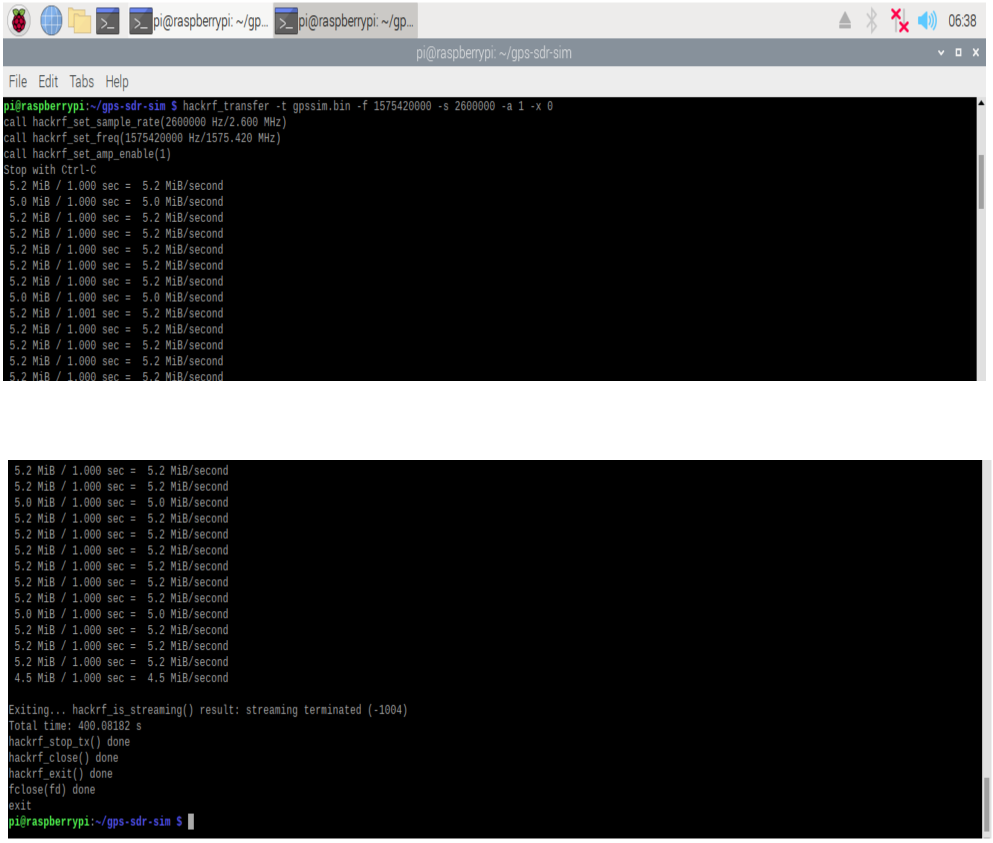
- After the check, the various bogus satellites are generated using the command "./gps-sdr-sim -e (file downloaded from FileZilla) -l (latitude, longitude, altitude) -d (duration) -b (bit) -t (date, time)". Specifically in this case the command will be "./gps-sdr-sim -e brdc.3020.22n -l 45.4731921, 9.1785101, 100 -d 400 -b 8 -t 2022/10/29, 20:00:00". Figure 5.6 shows the different satellites generated by the above command and the duration of the spoofing signal in seconds;



```
pi@raspberrypi:~ $ cd gps-sdr-sim
pi@raspberrypi:~/gps-sdr-sim $ ls
bladerf.script   extclk              gpssim.c  README.md      ublox.jpg
brdc0010.22n     getopt.c            gpssim.h  rocket.csv     u-center.png
brdc3020.22n     getopt.h            gpssim.o  rtk
brdc3020.22n.gz  gps-sdr-sim         LICENSE   satellite.csv
circle.csv       gps-sdr-sim-uhd.py  Makefile  satgen
circle_llh.csv   gpssim.bin          player    triumphv3.txt
pi@raspberrypi:~/gps-sdr-sim $ ./gps-sdr-sim -e brdc3020.22n -l 45.4731921, 9.1785101, 100 -d 400 -b 8 -t 2022/10/29, 20:00:00
Using static location mode.
xyz =   4480251.8,         0.0,   4524380.4
llh =   45.473192,    0.000000,         0.0
Start time = 2022/10/29,00:00:00 (2233:518400)
Duration = 400.0 [sec]
02   48.8  30.9  23287863.7   2.6
04  289.2   2.6  25458356.7   4.8
05   47.5  15.5  23987145.3   3.6
09  321.5   0.5  25671458.5   5.0
16  305.1  45.4  21386200.3   2.0
18  107.9  69.1  20561396.2   1.6
23  138.8  12.0  24553169.7   3.9
25  125.5   5.3  25392660.5   4.5
26  291.7  77.2  20312963.3   1.5
27  264.1  24.1  23033587.1   3.0
28  105.2  87.8  20049346.5   1.5
29   65.5  30.6  22696840.3   2.6
31  200.4  32.9  22798646.3   2.5
Time into run = 400.0
Done!
Process time = 260.1 [sec]
pi@raspberrypi:~/gps-sdr-sim $ 
```

Figure 5.6: Generation of bogus satellites.

- Once the satellite signals have been generated, it is now possible to generate the spoofing signal by connecting the Raspberry Pi 4 Model B hardware device with the HackRF One device and using the command hackrf transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 0. In Figure 5.7, we can see how the signal is subsequently emitted by the device.



```
pi@raspberrypi:~/gps-sdr-sim $ hackrf_transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 0
call hackrf_set_sample_rate(2600000 Hz/2.600 MHz)
call hackrf_set_freq(1575420000 Hz/1575.420 MHz)
call hackrf_set_amp_enable(1)
Stop with Ctrl-C
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.0 MiB / 1.000 sec =  5.0 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.0 MiB / 1.000 sec =  5.0 MiB/second
 5.2 MiB / 1.001 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
```

```
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.0 MiB / 1.000 sec =  5.0 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.0 MiB / 1.000 sec =  5.0 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 5.2 MiB / 1.000 sec =  5.2 MiB/second
 4.5 MiB / 1.000 sec =  4.5 MiB/second

Exiting... hackrf_is_streaming() result: streaming terminated (-1004)
Total time: 400.08182 s
hackrf_stop_tx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
pi@raspberrypi:~/gps-sdr-sim $ 
```

Figure 5.7: Generation of Spoofing signals.

During the spoofing signal, data can be collected using the application introduced in Section 4.6. Tables 5.1, 5.2, 5.3 and 5.4 show the data collected and the differences between the spoofed coordinates and those identified by the application in the spoofed environment. Specifically, 100 measurements lasting 150 seconds each were taken with the following coordinates [45°28'23.5" N 9°10'42.6" E] (Milan, Castello Sforzesco).

n	POSITION SET	DETECTED POSITION	T_M	D_{IC}
1	45°28'23.5" N 9°10'42.6" E	45°28'25.3" N 9°10'42.2" E	150 s	56.2 m
2	45°28'23.5" N 9°10'42.6" E	45°28'24.6" N 9°10'42.4" E	150 s	33.0 m
3	45°28'23.5" N 9°10'42.6" E	45°28'25.2" N 9°10'43.5" E	150 s	56.3 m
4	45°28'23.5" N 9°10'42.6" E	45°28'24.2" N 9°10'41.0" E	150 s	42.1 m
5	45°28'23.5" N 9°10'42.6" E	45°28'24.1" N 9°10'41.8" E	150 s	25.3 m
6	45°28'23.5" N 9°10'42.6" E	45°28'25.6" N 9°10'42.0" E	150 s	66.0 m
7	45°28'23.5" N 9°10'42.6" E	45°28'23.3" N 9°10'43.7" E	150 s	23.3 m
8	45°28'23.5" N 9°10'42.6" E	45°28'23.7" N 9°10'44.0" E	150 s	29.4 m
9	45°28'23.5" N 9°10'42.6" E	45°28'25.2" N 9°10'41.0" E	150 s	62.9 m
10	45°28'23.5" N 9°10'42.6" E	45°28'23.3" N 9°10'41.9" E	150 s	18.2 m
11	45°28'23.5" N 9°10'42.6" E	45°28'23.9" N 9°10'44.0" E	150 s	32.5 m
12	45°28'23.5" N 9°10'42.6" E	45°28'23.9" N 9°10'40.9" E	150 s	40.0 m
13	45°28'23.5" N 9°10'42.6" E	45°28'24.1" N 9°10'44.1" E	150 s	35.8 m
14	45°28'23.5" N 9°10'42.6" E	45°28'23.1" N 9°10'41.3" E	150 s	31.7 m
15	45°28'23.5" N 9°10'42.6" E	45°28'22.9" N 9°10'44.2" E	150 s	38.5 m
16	45°28'23.5" N 9°10'42.6" E	45°28'25.4" N 9°10'41.6" E	150 s	63.7 m
17	45°28'23.5" N 9°10'42.6" E	45°28'24.2" N 9°10'43.0" E	150 s	22.4 m
18	45°28'23.5" N 9°10'42.6" E	45°28'25.0" N 9°10'41.3" E	150 s	55.5 m
19	45°28'23.5" N 9°10'42.6" E	45°28'23.0" N 9°10'44.0" E	150 s	33.4 m
20	45°28'23.5" N 9°10'42.6" E	45°28'23.6" N 9°10'43.1" E	150 s	10.4 m
21	45°28'23.5" N 9°10'42.6" E	45°28'25.4" N 9°10'42.1" E	150 s	60.6 m
22	45°28'23.5" N 9°10'42.6" E	45°28'24.9" N 9°10'42.6" E	150 s	44.2 m
23	45°28'23.5" N 9°10'42.6" E	45°28'25.2" N 9°10'42.7" E	150 s	52.4 m
24	45°28'23.5" N 9°10'42.6" E	45°28'25.1" N 9°10'41.1" E	150 s	60.2 m
25	45°28'23.5" N 9°10'42.6" E	45°28'22.8" N 9°10'42.8" E	150 s	20.7 m

Table 5.1: Table 1 of data collected with spoofing signal in a static situation. T_M represents the Measurement Time and D_{IC} represents the Deviation from Imposed Coordinates.

n	POSITION SET	DETECTED POSITION	T_M	D_{IC}
26	45°28’23.5" N 9°10’42.6" E	45°28’24.2" N 9°10’44.2" E	150 s	41.0 m
27	45°28’23.5" N 9°10’42.6" E	45°28’24.1" N 9°10’41.6" E	150 s	28.7 m
28	45°28’23.5" N 9°10’42.6" E	45°28’24.9" N 9°10’43.2" E	150 s	44.4 m
29	45°28’23.5" N 9°10’42.6" E	45°28’22.9" N 9°10’44.1" E	150 s	36.8 m
30	45°28’23.5" N 9°10’42.6" E	45°28’24.4" N 9°10’43.9" E	150 s	38.3 m
31	45°28’23.5" N 9°10’42.6" E	45°28’23.1" N 9°10’41.7" E	150 s	23.6 m
32	45°28’23.5" N 9°10’42.6" E	45°28’23.6" N 9°10’42.2" E	150 s	10.5 m
33	45°28’23.5" N 9°10’42.6" E	45°28’25.6" N 9°10’41.5" E	150 s	68.4 m
34	45°28’23.5" N 9°10’42.6" E	45°28’24.3" N 9°10’41.5" E	150 s	35.2 m
35	45°28’23.5" N 9°10’42.6" E	45°28’25.2" N 9°10’42.3" E	150 s	52.3 m
36	45°28’23.5" N 9°10’42.6" E	45°28’24.9" N 9°10’41.1" E	150 s	55.5 m
37	45°28’23.5" N 9°10’42.6" E	45°28’24.4" N 9°10’44.3" E	150 s	46.1 m
38	45°28’23.5" N 9°10’42.6" E	45°28’24.0" N 9°10’42.4" E	150 s	15.6 m
39	45°28’23.5" N 9°10’42.6" E	45°28’24.5" N 9°10’41.4" E	150 s	39.7 m
40	45°28’23.5" N 9°10’42.6" E	45°28’25.6" N 9°10’43.7" E	150 s	69.3 m
41	45°28’23.5" N 9°10’42.6" E	45°28’24.3" N 9°10’43.4" E	150 s	29.6 m
42	45°28’23.5" N 9°10’42.6" E	45°28’25.4" N 9°10’41.4" E	150 s	64.9 m
43	45°28’23.5" N 9°10’42.6" E	45°28’24.1" N 9°10’41.1" E	150 s	38.4 m
44	45°28’23.5" N 9°10’42.6" E	45°28’24.6" N 9°10’41.7" E	150 s	39.1 m
45	45°28’23.5" N 9°10’42.6" E	45°28’26.3" N 9°10’44.1" E	150 s	91.9 m
46	45°28’23.5" N 9°10’42.6" E	45°28’25.7" N 9°10’43.1" E	150 s	69.9 m
47	45°28’23.5" N 9°10’42.6" E	45°28’23.4" N 9°10’42.5" E	150 s	4.90 m
48	45°28’23.5" N 9°10’42.6" E	45°28’23.8" N 9°10’41.4" E	150 s	29.6 m
49	45°28’23.5" N 9°10’42.6" E	45°28’25.1" N 9°10’41.1" E	150 s	60.3 m
50	45°28’23.5" N 9°10’42.6" E	45°28’23.9" N 9°10’40.9" E	150 s	40.4 m

Table 5.2: Table 2 of data collected with spoofing signal in a static situation. T_M represents the Measurement Time and D_{IC} represents the Deviation from Imposed Coordinates.

n	POSITION SET	DETECTED POSITION	T_M	D_{IC}
51	45°28’23.5" N 9°10’42.6" E	45°28’24.0" N 9°10’43.8" E	150 s	28.0 m
52	45°28’23.5" N 9°10’42.6" E	45°28’24.4" N 9°10’44.2" E	150 s	44.4 m
53	45°28’23.5" N 9°10’42.6" E	45°28’23.9" N 9°10’42.9" E	150 s	12.8 m
54	45°28’23.5" N 9°10’42.6" E	45°28’24.2" N 9°10’43.1" E	150 s	24.8 m
55	45°28’23.5" N 9°10’42.6" E	45°28’24.2" N 9°10’42.7" E	150 s	21.7 m
56	45°28’23.5" N 9°10’42.6" E	45°28’24.1" N 9°10’43.8" E	150 s	31.4 m
57	45°28’23.5" N 9°10’42.6" E	45°28’23.6" N 9°10’44.0" E	150 s	29.6 m
58	45°28’23.5" N 9°10’42.6" E	45°28’25.0" N 9°10’41.9" E	150 s	48.8 m
59	45°28’23.5" N 9°10’42.6" E	45°28’23.1" N 9°10’43.0" E	150 s	14.8 m
60	45°28’23.5" N 9°10’42.6" E	45°28’24.7" N 9°10’41.8" E	150 s	41.9 m
61	45°28’23.5" N 9°10’42.6" E	45°28’24.6" N 9°10’43.9" E	150 s	43.3 m
62	45°28’23.5" N 9°10’42.6" E	45°28’23.7" N 9°10’41.0" E	150 s	36.0 m
63	45°28’23.5" N 9°10’42.6" E	45°28’25.1" N 9°10’41.3" E	150 s	58.6 m
64	45°28’23.5" N 9°10’42.6" E	45°28’23.0" N 9°10’42.1" E	150 s	20.2 m
65	45°28’23.5" N 9°10’42.6" E	45°28’23.4" N 9°10’43.9" E	150 s	27.3 m
66	45°28’23.5" N 9°10’42.6" E	45°28’24.8" N 9°10’43.8" E	150 s	47.6 m
67	45°28’23.5" N 9°10’42.6" E	45°28’25.0" N 9°10’44.4" E	150 s	60.0 m
68	45°28’23.5" N 9°10’42.6" E	45°28’24.9" N 9°10’41.3" E	150 s	50.9 m
69	45°28’23.5" N 9°10’42.6" E	45°28’24.5" N 9°10’41.8" E	150 s	34.8 m
70	45°28’23.5" N 9°10’42.6" E	45°28’25.0" N 9°10’44.0" E	150 s	54.4 m
71	45°28’23.5" N 9°10’42.6" E	45°28’25.0" N 9°10’41.7" E	150 s	51.7 m
72	45°28’23.5" N 9°10’42.6" E	45°28’23.2" N 9°10’41.0" E	150 s	36.6 m
73	45°28’23.5" N 9°10’42.6" E	45°28’25.2" N 9°10’41.1" E	150 s	63.0 m
74	45°28’23.5" N 9°10’42.6" E	45°28’23.2" N 9°10’41.8" E	150 s	20.5 m
75	45°28’23.5" N 9°10’42.6" E	45°28’23.7" N 9°10’43.8" E	150 s	26.2 m

Table 5.3: Table 3 of data collected with spoofing signal in a static situation. T_M represents the Measurement Time and D_{IC} represents the Deviation from Imposed Coordinates.

n	POSITION SET	DETECTED POSITION	T_M	D_{IC}
76	45°28’23.5" N 9°10’42.6" E	45°28’23.1" N 9°10’40.9" E	150 s	39.6 m
77	45°28’23.5" N 9°10’42.6" E	45°28’22.9" N 9°10’41.7" E	150 s	27.1 m
78	45°28’23.5" N 9°10’42.6" E	45°28’24.8" N 9°10’42.9" E	150 s	41.3 m
79	45°28’23.5" N 9°10’42.6" E	45°28’24.6" N 9°10’44.3" E	150 s	49.1 m
80	45°28’23.5" N 9°10’42.6" E	45°28’23.8" N 9°10’42.9" E	150 s	10.0 m
81	45°28’23.5" N 9°10’42.6" E	45°28’24.5" N 9°10’42.8" E	150 s	31.8 m
82	45°28’23.5" N 9°10’42.6" E	45°28’23.1" N 9°10’41.6" E	150 s	24.9 m
83	45°28’23.5" N 9°10’42.6" E	45°28’25.7" N 9°10’42.5" E	150 s	68.8 m
84	45°28’23.5" N 9°10’42.6" E	45°28’25.5" N 9°10’43.8" E	150 s	67.6 m
85	45°28’23.5" N 9°10’42.6" E	45°28’25.4" N 9°10’41.9" E	150 s	61.8 m
86	45°28’23.5" N 9°10’42.6" E	45°28’25.5" N 9°10’43.9" E	150 s	67.3 m
87	45°28’23.5" N 9°10’42.6" E	45°28’25.6" N 9°10’42.6" E	150 s	66.3 m
88	45°28’23.5" N 9°10’42.6" E	45°28’23.0" N 9°10’42.0" E	150 s	22.0 m
89	45°28’23.5" N 9°10’42.6" E	45°28’23.2" N 9°10’41.4" E	150 s	27.8 m
90	45°28’23.5" N 9°10’42.6" E	45°28’22.9" N 9°10’41.4" E	150 s	33.3 m
91	45°28’23.5" N 9°10’42.6" E	45°28’24.1" N 9°10’41.2" E	150 s	36.4 m
92	45°28’23.5" N 9°10’42.6" E	45°28’23.0" N 9°10’44.1" E	150 s	35.4 m
93	45°28’23.5" N 9°10’42.6" E	45°28’23.7" N 9°10’42.4" E	150 s	8.90 m
94	45°28’23.5" N 9°10’42.6" E	45°28’24.9" N 9°10’41.0" E	150 s	56.2 m
95	45°28’23.5" N 9°10’42.6" E	45°28’25.5" N 9°10’42.9" E	150 s	61.4 m
96	45°28’23.5" N 9°10’42.6" E	45°28’24.2" N 9°10’43.9" E	150 s	35.3 m
97	45°28’23.5" N 9°10’42.6" E	45°28’23.3" N 9°10’43.5" E	150 s	19.9 m
98	45°28’23.5" N 9°10’42.6" E	45°28’24.3" N 9°10’41.4" E	150 s	37.7 m
99	45°28’23.5" N 9°10’42.6" E	45°28’24.3" N 9°10’43.5" E	150 s	32.5 m
100	45°28’23.5" N 9°10’42.6" E	45°28’23.4" N 9°10’44.3" E	150 s	35.9 m

Table 5.4: Table 4 of data collected with spoofing signal in a static situation. T_M represents the Measurement Time and D_{IC} represents the Deviation from Imposed Coordinates.

5.2 Data acquisition with spoofing signal in a dynamic situation

In Section 5.1, measurements were made in which both the attacker and the victim were in a static situation. In this section, unlike the previous one, measurements will be made in which both the spoofer and the victim are in a dynamic situation. The aim of the proposed case study is to simulate a real situation in which the attacker was able to place a spoofer on the same vehicle as the tracking device. The measurements were carried out in an open, flat environment with normal weather conditions. To carry them out the portable spoofer shown in Figures 5.8 and 5.9 was used:

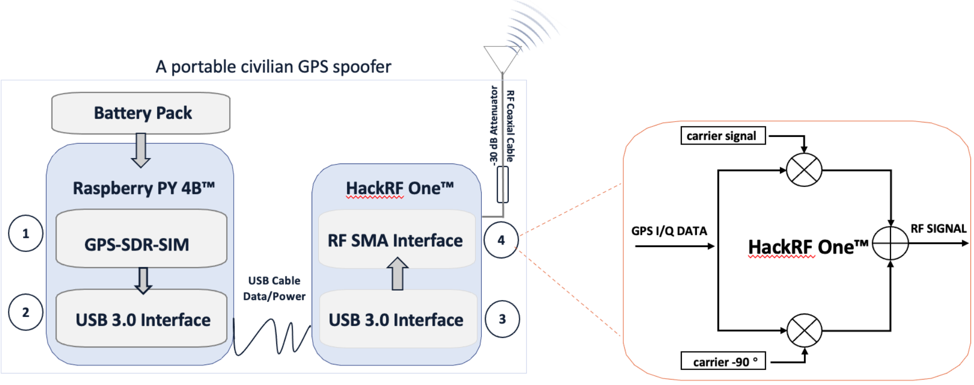


Figure 5.8: High-level schematic of the low-cost portable spoofer.

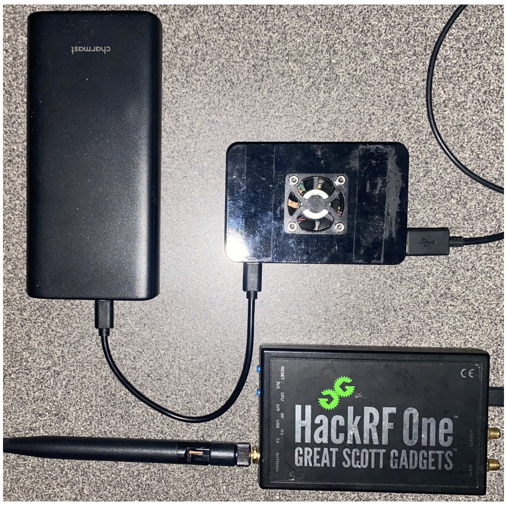


Figure 5.9: Low-cost Portable Spoofer.

Compared to the device used previously and described in Figure 5.1, there are no more input/output devices and instead of a power socket there is now a battery pack that will supply power to the Raspberry Pi hardware device. Therefore in the end the device will consist of:

- Raspberry Pi 4 Model B Hardware Device;
- HackRF One Device;
- Antenna;
- Battery Pack;
- Two Xiaomi RedMI 8 Phones.

Again, the spoofing signal will have to be generated, but unlike the previous procedure, the file type "gpssim.bin" will be created and subsequently loaded into the hardware. This hardware, thanks to a Matlab programme, will be able to automatically emit the spoofing signal via Hackrf One for the duration of time previously set by the attacker. The measurements were carried out using the Italian Army's VM 90 multi-role vehicle travelling at a speed of between 10 km/h and 50km/h inside the military facility mentioned at the beginning of the chapter.

Figure 5.10: Multirole Vehicle VM90 in Italian Army Service.

A moving data set was collected during the spoofing signal. Table 5.5 shows the data collected. As in the case of 5.1, the differences between the spoofed coordinates set and those identified by the GnssLogger application were expressed. In this case, 30 measurements of 150 seconds each were taken with coordinates set at [45°28’23.5" N 9°10’42.6" E] (Milan, Castello Sforzesco).

n	POSITION SET	DETECTED POSITION	T_M	D_{IC}
1	45°28’23.5" N 9°10’42.6" E	45°28’24.6" N 9°10’43.5" E	150 s	39.8 m
2	45°28’23.5" N 9°10’42.6" E	45°28’23.4" N 9°10’42.5" E	150 s	5.0 m
3	45°28’23.5" N 9°10’42.6" E	45°28’23.3" N 9°10’43.8" E	150 s	25.2 m
4	45°28’23.5" N 9°10’42.6" E	45°28’24.9" N 9°10’41.9" E	150 s	45.6 m
5	45°28’23.5" N 9°10’42.6" E	45°28’22.9" N 9°10’42.5" E	150 s	18.4 m
6	45°28’23.5" N 9°10’42.6" E	45°28’25.3" N 9°10’43.2" E	150 s	56.3 m
7	45°28’23.5" N 9°10’42.6" E	45°28’24.2" N 9°10’41.3" E	150 s	36.2 m
8	45°28’23.5" N 9°10’42.6" E	45°28’25.0" N 9°10’41.9" E	150 s	49.6 m
9	45°28’23.5" N 9°10’42.6" E	45°28’25.6" N 9°10’41.8" E	150 s	66.6 m
10	45°28’23.5" N 9°10’42.6" E	45°28’24.6" N 9°10’41.5" E	150 s	42.4 m
11	45°28’23.5" N 9°10’42.6" E	45°28’23.9" N 9°10’42.9" E	150 s	12.5 m
12	45°28’23.5" N 9°10’42.6" E	45°28’25.1" N 9°10’43.3" E	150 s	52.3 m
13	45°28’23.5" N 9°10’42.6" E	45°28’24.7" N 9°10’41.3" E	150 s	48.5 m
14	45°28’23.5" N 9°10’42.6" E	45°28’25.3" N 9°10’42.4" E	150 s	55.8 m
15	45°28’23.5" N 9°10’42.6" E	45°28’25.5" N 9°10’42.0" E	150 s	63.9 m
16	45°28’23.5" N 9°10’42.6" E	45°28’25.1" N 9°10’43.3" E	150 s	50.4 m
17	45°28’23.5" N 9°10’42.6" E	45°28’24.7" N 9°10’41.9" E	150 s	41.2 m
18	45°28’23.5" N 9°10’42.6" E	45°28’24.7" N 9°10’42.7" E	150 s	36.9 m
19	45°28’23.5" N 9°10’42.6" E	45°28’22.9" N 9°10’40.8" E	150 s	42.9 m
20	45°28’23.5" N 9°10’42.6" E	45°28’24.5" N 9°10’41.1" E	150 s	45.9 m
21	45°28’23.5" N 9°10’42.6" E	45°28’23.9" N 9°10’41.8" E	150 s	22.2 m
22	45°28’23.5" N 9°10’42.6" E	45°28’25.1" N 9°10’43.7" E	150 s	55.1 m
23	45°28’23.5" N 9°10’42.6" E	45°28’24.3" N 9°10’43.4" E	150 s	30.1 m
24	45°28’23.5" N 9°10’42.6" E	45°28’25.4" N 9°10’42.3" E	150 s	58.5 m
25	45°28’23.5" N 9°10’42.6" E	45°28’25.3" N 9°10’42.4" E	150 s	56.2 m
26	45°28’23.5" N 9°10’42.6" E	45°28’24.4" N 9°10’42.8" E	150 s	29.4 m
27	45°28’23.5" N 9°10’42.6" E	45°28’24.9" N 9°10’43.8" E	150 s	50.4 m
28	45°28’23.5" N 9°10’42.6" E	45°28’25.2" N 9°10’41.5" E	150 s	58.4 m
29	45°28’23.5" N 9°10’42.6" E	45°28’24.2" N 9°10’42.6" E	150 s	22.6 m
30	45°28’23.5" N 9°10’42.6" E	45°28’23.5" N 9°10’41.2" E	150 s	32.1 m

Table 5.5: Table 1 of data collected with spoofing signal in a dynamic situation. T_M is the Measurement Time and D_{IC} is the Deviation from Imposed Coordinates.

5.3 Results of data acquisition in the presence of the spoofing signal

This section shows some of the graphs of the collected measurements produced by Matlab (see Section 4.3).

In the case of signals collected in a normal situation, i.e. without the presence of the spoofing signal, output graphs such as the one depicted in Figures 5.11 and 5.12 can be obtained.

The situation changes dramatically when the user is operating in scenarios 2 and 3, i.e. when data is collected with the presence of the spoofing signal. In this case, the collected data will produce output graphs like those shown in Figures 5.13, 5.14, 5.15 and 5.16.

The graphs in the following Section are just some of those produced during the spoofing experimental campaigns and thanks to them in the following Sections it will be possible to carry out a study to recognise when the user is subject to a spoofing attack.

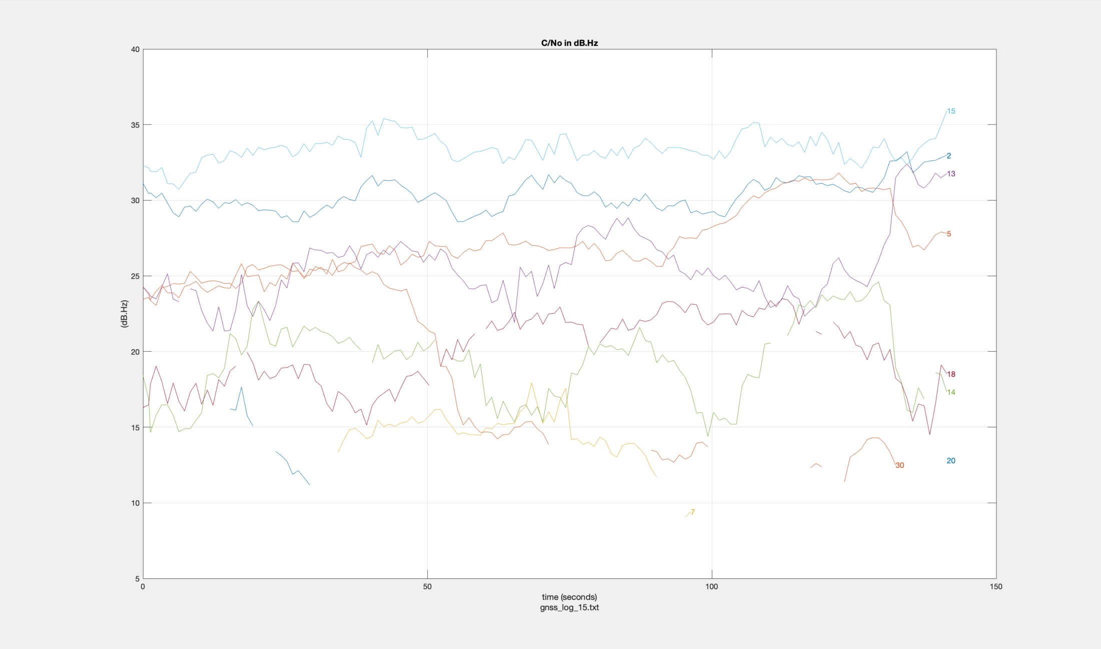


Figure 5.11: Graph of a signal collected in a situation where the user is not a victim of a spoofing attack.

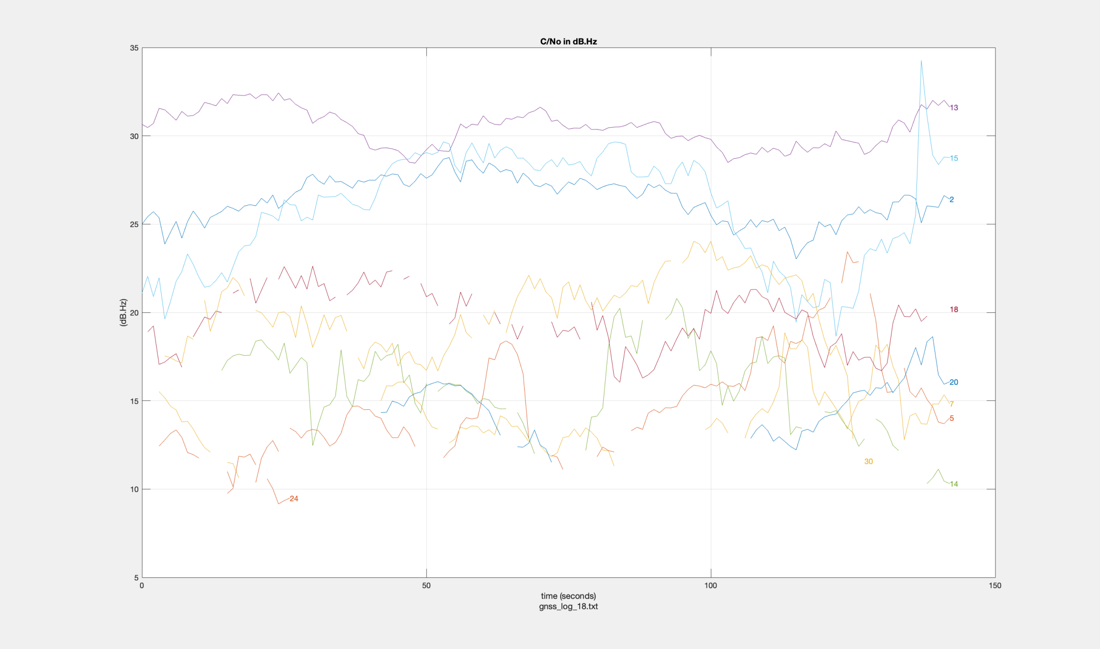


Figure 5.12: Graph of a signal collected in a situation where the user is not a victim of a spoofing attack.

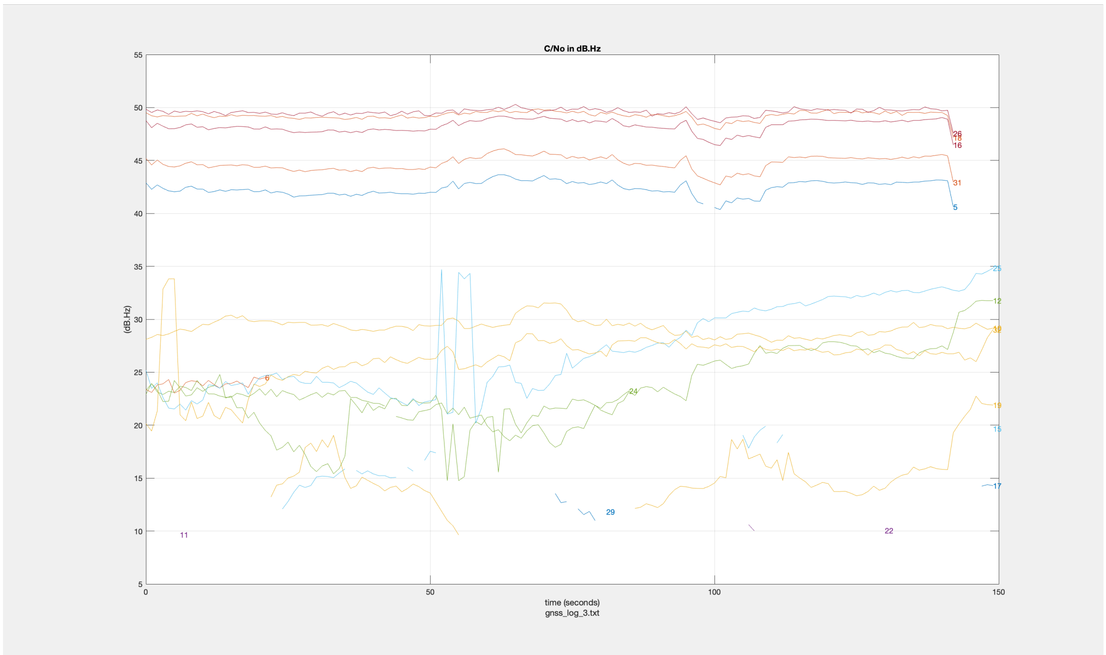


Figure 5.13: Graph of a signal collected in a situation where the user is the victim of a spoofing attack. Scenario 2, static situation.

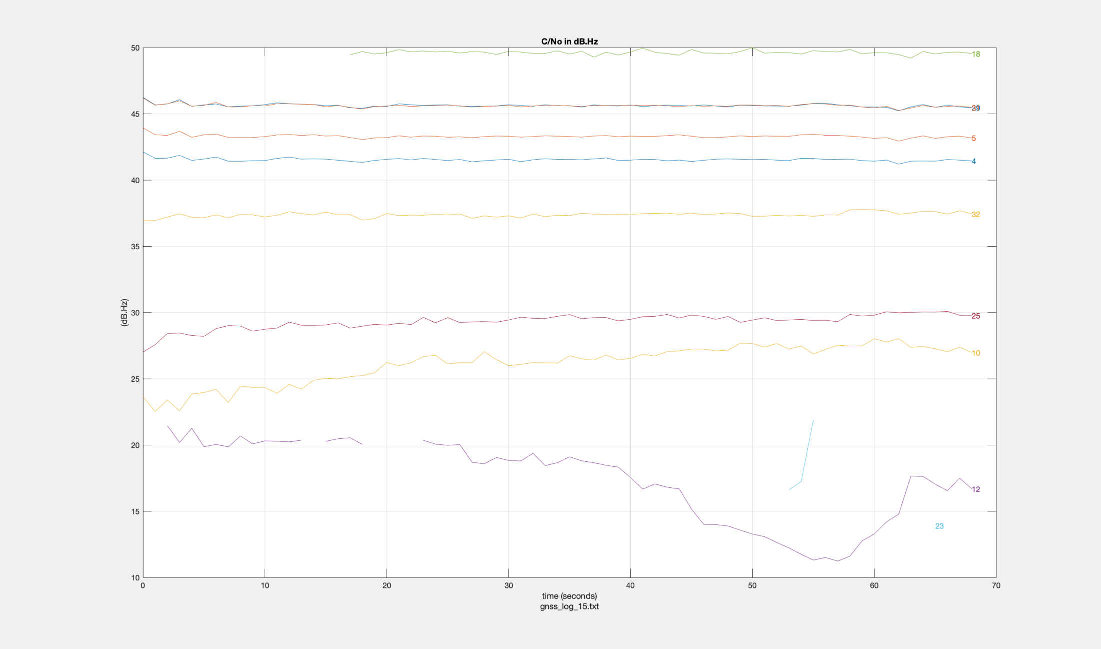


Figure 5.14: Graph of a signal collected in a situation where the user is the victim of a spoofing attack. Scenario 2, static situation.

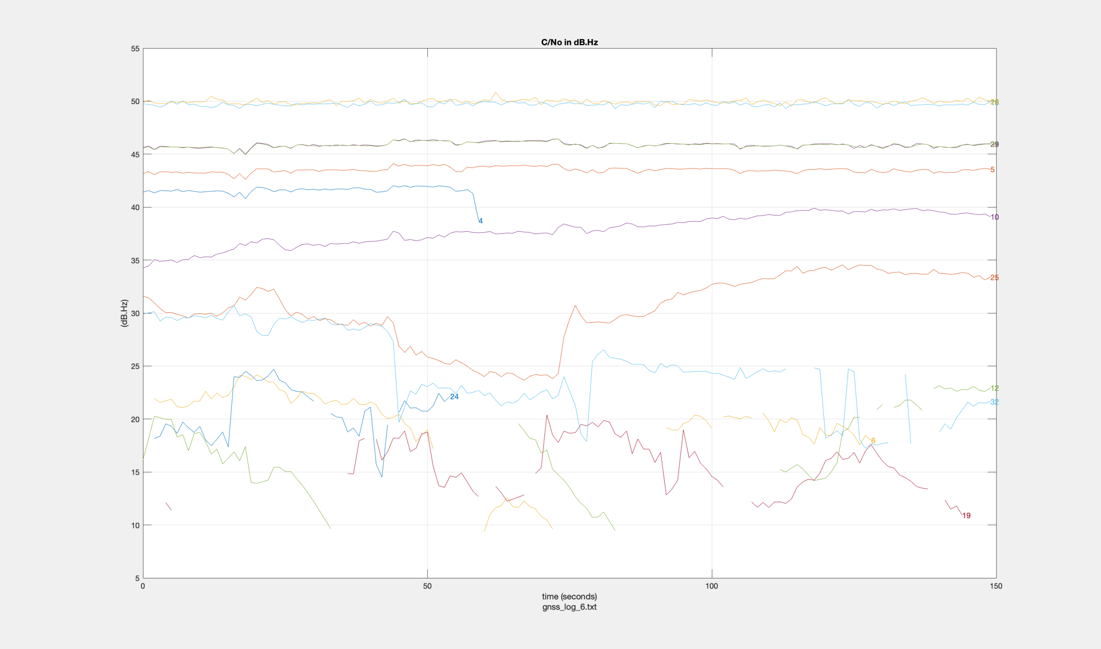


Figure 5.15: Graph of a signal collected in a situation where the user is the victim of a spoofing attack. Scenario 2, dynamic situation.

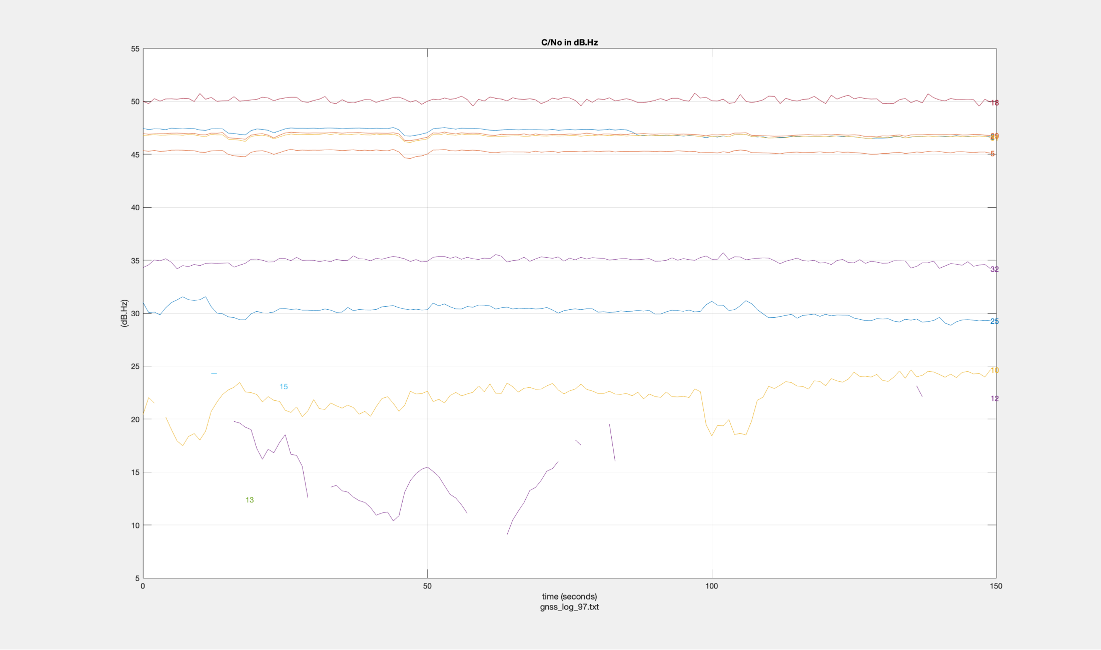


Figure 5.16: Graph of a signal collected in a situation where the user is the victim of a spoofing attack. Scenario 2, dynamic situation.

5.4 Methodology

We developed a low-cost portable spoofer 5.17 based on a Great Scott Gadgets™ HackRF One™ platform [81] and a Raspberry™ PI 4B. The used front-end HackRF One™ is a low-cost, open-source Software Defined Radio allowing fast and accurate RF signal transmission from binary files. This front-end can receive and transmit signals from 1 MHz to 6 GHz with adjustable power and channel capacity. The software used to numerically generate the spoofed GPS signal is GPS-SDR-SIM [82], an open GPS L1 C/A signal generator toolbox distributed with a MIT license [83]. The attack was planned to simulate a static position and all the visible satellites belonging to GNSS constellations and their signals were transmitted to the SDR equipment. An optional reference clock/oscillator can be used to discipline the signal generation at an increased cost of the overall equipment. For the scope of the paper, a reference oscillator was not connected to the front end. The power supply can be provided through a mass-market, 10000 mAh battery pack according to the supply specification of the Raspberry™ PI 4B. The HackRF One™can be then supplied by the Raspberry PI itself through the USB 3.0 interface.

The spoofing attack can be performed through the portable spoofer according to the following steps:

1. *Trajectory generation.* The fake trajectory was generated in Linux OS implementing the daily GPS broadcast ephemeris Receiver Independent Exchange Format (RINEX) file, a National Marine Electronics Association (NMEA) GGA stream, and a `.csv` file containing the Earth-centered Earth-fixed (ECEF) position with a 10 Hz sampling rate. The file is transmitted through the USB interface of the Raspberry™ PI 4B. The Spoofer takes a RINEX formatted GPS ephemeris archive and location as input and generates a GPS baseband signal for the SDR platform to playback.

```
$ cd gps-sdr-sim
$ gcc gpssim.c -lm -fopenmp -o gps-sdr-sim
$ ./gps-sdr-sim -h
$ ./gps-sdr-sim -e brdc0640.22n -l
```

2. *Numerical signal generation.* The trajectory is then injected into the GPS-SDR-SIM. The software generates the simulated pseudorange and Doppler for the GPS satellites in view. This simulated range data was used to produce a file with In-phase / Quadrature (I / Q) samples of the complex baseband signal envelope ready to be inserted into the front end of the SDR (i.e. HackRF One™) given in Figure 5.17.

```
$ ./gps-sdr-sim -e brdc0640.22n
-l 41.347041,68.950321,100
```

3. *Digital to Analog Conversion and RF Signal Transmission.* The front-end (HackRF One™) is in charge to perform the digital-to-analog conversion by mixing the baseband signal provided at step 2 with the carrier frequency (i.e. GPS L1 C/A), thus, offering quadrature modulation in the L1 band.

4. *RF signal transmission.* After baseband signal samples are generated, we can transmit them through an antenna of the SDR platform at L1 frequencies. Specifically, the command line below is used to transmit the samples using HackRF, at 1575.42MHz, repeatedly:

```
$ hackrf_transfer -t gpssim.bin
-f 1575420000 -s 2600000 -a 1 -x 0
```

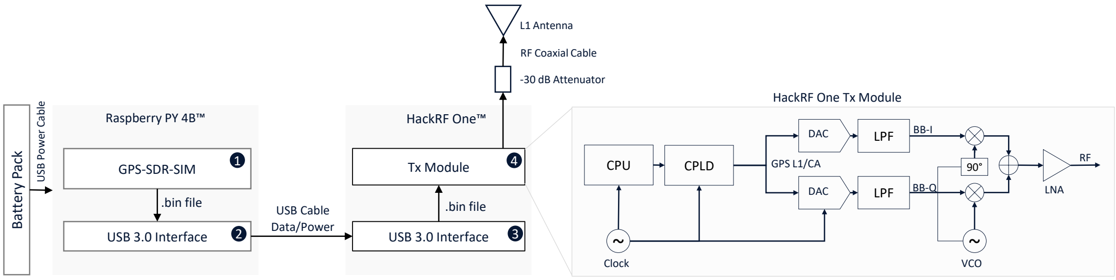


Figure 5.17: High-level schematic of the low-cost portable spoofer (left). The transmitter architecture (right) produces I/Q modulated GPS L1/CA signals transmitted at 1575.42 Mhz through the Digital-to-Analog conversion and up-mixing of baseband signal samples (BB-I, BB-Q).

In absence of interference, the GNSS signal received at the antenna can be modelled as the sum of N_s independent satellites' signals 5.1

$$x_{f_c}(t) = \sum_{i=1}^{N_s-1} [\sqrt{P_{R,i}} D_i(t-\tau_i) C_i(t-\tau_i) \times \tag{5.1}$$

$$cos\left(2\pi\left(f_c + f_{d,i}(t)\right)t + \triangle\theta_i\right)] + n(t), \tag{5.2}$$

where $P_{R,i}$ is the received signal power, $D_i(t)$ is the navigation data bit stream, $C_i(t)$ is the pseudo-random code sequence, f_c is the carrier frequency shifted by the observed Doppler shift $f_{d,i}$, τ_i is the propagation delay and $\triangle\theta_i$ is the phase offset. Eventually, $n(t)$ is the thermal noise contribution. The received power of each signal, P_{R_i}, reflects the unique propagation path it covered between transmitting and receiving antennas. From a geometrical and physical standpoint, legitimate GNSS signals propagate through different channels, and the free space path loss mainly contributes to the differences in the received power observed for each satellite. Multipath-related constructive and destructive interferences are also responsible for fluctuations in the received power, being conditioned by the elevation at which each satellite is observed. In order to cheat a GNSS receiver, a portable spoofer must replicate all the components of the navigation signals defined in 5.1, such as satellite spreading code, RF carrier, and navigation data bits of the selected constellation. A simplistic GNSS spoofer generates and transmits GNSS-like signals, but cannot keep phase and time coherence w.r.t. to the legitimate signals without external synchronization sources. The generated fake signals have a similar structure to the legitimate signals, however, they may differ in terms of Doppler and phase shifts of both code and carrier. Furthermore, different power levels are usually observed at the receiver location for spoofing and legitimate signals respectively. Advanced attacks may calibrate the signal power to be similar enough to the received power of each legitimate GNSS signal. However, such a calibration would require accurate knowledge of the attacker-to-victim range, thus of the victim's location, as typically addressed by sophisticated spoofing actions. In the following, spoofing GNSS signals will be identified through the apex $(\cdot)^{(S)}$. A simplistic spoofer will generate N_{sp} fake signals characterized by code delay $\tau_i^{(S)}$, carrier phase $\triangle\theta_i^{(S)}$, Doppler shift $f_{d,i}^{(S)}(t)$ and data bit stream $\hat{D}_i$ for $i = 1{,}2{,}3, ...N_{sp}$. A further Doppler shift, $f_d^{(S)}(t)$ may be introduced by the relative kinematics of transmitting and receiving antennas and is assumed equal to zero when both are static or they both are carried on the same rigid body. The expression for the sum of N_{sp} single frequency, single constellation spoofed signals is 5.3

$$x_{f_c}^{(S)}(t) = \sum_{i=1}^{N_{sp}-1} [\sqrt{P_{R,i}^{(S)}}\hat{D}_i(t-\tau_i^{(S)})C_i(t-\tau_i^{(S)}) \times \tag{5.3}$$

$$cos(2\pi(f_c + f_{d,i}^{(S)}(t) + f_d^{(S)}(t))t + \triangle\theta_i^{(S)})] + n(t) \tag{5.4}$$

where the received power at the antenna, $P_{R,i}^{(S)}$, reflects the different amplitude attributed by the spoofer to each signal to simulate different path losses and the $\hat{D}_i$ highlights possible differences w.r.t. the legitimate data stream foresaw in 5.1. The value of $P_{R,i}^{(S)}$, as for the real signals, may actually change over time, but differently from the case of (5.1), such variations depend on

- the misalignment of transmitting and receiving antennas as well as any changes in their relative heading during the spoofing attack
- the fading effects introduced by the terrestrial channel and mostly due to multipath which is very relevant when the attacker is at the same altitude as the victim receiver.

Furthermore, differently from the authentic GNSS signals, all the spoofing signals travel through the same propagation path from the spoofer to the receiver, thus experiencing a common, yet single physical channel. The aforementioned power variations will then reflect in a similar way on each generated satellite's signal by introducing strong spatial and temporal correlation.

When a GNSS receiver is under a spoofing attack, it receives both authentic and spoofed signals, and additive thermal noise affects their sum. Therefore, the total signal at the victim receiver's front end is modelled as

$$x_{tot}(t) = \bar{x}_{f_c}(t) + \bar{x}_{f_c}^{(S)}(t) + n(t) \tag{5.5}$$

where the notation $\bar{x}$ indicates the noiseless, legitimate and spoofing signals derived from 5.1 and 5.3 by neglecting the respective noise terms. Without any lack of generality, in the following f_c will be referred to as GPS L1 centre frequency, i.e., 1575.42 MHz.

5.4.1 Devices under test

A variety of Xiaomi Redmi, Samsung and Huawei Android smartphones with multi-frequency GNSS chipsets were chosen to test the effect of the simplistic spoofing attack performed through the portable spoofer described above. The list of devices under test is reported in Table 5.6. These devices are all equipped with Google Android™ OS and the GNSS Logger Android application provided by Google™ was installed for the procurement of GNSS raw measurements. Additionally, the devices' PVT solutions were logged through the Android™ NMEA Tools application, which provides the GNSS standalone position of the smartphone in standard NMEA format [84].

Model	**System on cheap (SOC)**	**GPS Bands**
Xiaomi Redmi 6 Pro	Qualcomm Snapdragon 636	L1
Xiaomi Redmi 8	Qualcomm Snapdragon 845	L1+L5
Xiaomi Redmi 8 Pro	Qualcomm Snapdragon 845	L1+L5

Table 5.6: Android™ Devices under test and embedded GNSS chipsets.

ID	SCENARIOS	TARGET
S1	No spoofing signal	Xiaomi Redmi 8, Redmi 8 Pro
S2	Spoofing signal in a static situation	Xiaomi Redmi 8, Redmi 8 Pro
S3	Spoofing signal in a dynamic situation	Xiaomi Redmi 8, Redmi 8 Pro

Table 5.7: Test scenarios on different Android™ smartphones reporting spoofing vulnerability through their raw GNSS measurements.

5.4.2 Test metodology

Experiments on smartphones were carried out in a dedicated test campaign. Each test foresaw a 150 s data collection for all the scenarios listed in Table 5.7, in controlled environmental conditions. The range of the spoofer was kept at about 5 m and in order to prevent any RFI disturbances beyond the range of the controlled environment a 30 dB attenuator was used and inserted at the coaxial cable to reduce transmitting signal power levels and limit the spoofer coverage. The actual locations of the smartphones, i.e. the test site, were at N 45°02'49.3" E 7°39'00.5", 2022-10-15 UTC +00:00, while the portable spoofer broadcast spoofing signals over GPS L1 band with a fake location at N 45°28'23.5" E 9°10'42.6"(Milan, Castello Sforzesco), 2022-10-15 UTC +00:00 which was approximately 120 km away from the test location when switched on. Based on the results of the previous test campaigns [85], [41], we modified the test settings to achieve the most vulnerable conditions under which an Android™ smartphone could be spoofed by a simplistic spoofer. To this aim, prior to the test, the smartphones were kept in pilot/airplane mode for at least 2 hours with the smartphones devoid of any network and GNSS signals. This could enable the internal GNSS receiver to discard any ephemeris downloaded through the network or demodulated through GNSS signals. We performed three different simplistic spoofing scenarios, shown in Table 5.7. In the first scenario, the devices received real GNSS signals for 150 s. During the 150 s, the devices kept tracking true GNSS signals. In the second scenario, the spoofer transmitted spoofed signals for 350 s and then the devices kept tracking live GNSS signals. The third scenario was the same as the first but more receivers were concurrently under attack. In all three cases, two Xiaomi RedMI 8s were used to include multi-frequency support for GPS L1 and L5 navigation signals.

The set of visible real satellite signals during the test campaign is reported in Table 5.8. During the spoofing attacks, a set of fake signals were generated to force the GNSS receiver to estimate a wrong PVT solution and they are also reported in Table 5.8. As it can be seen, such a set includes:

- satellite signals that would be broadcasted by satellites that are not actually visible to the receivers at the time and location of the tests;
- satellite signals that are already being tracked by the receivers, for which the spoofing signal has to overcome the on-going tracking of the real signal.

Subset	GPS L1/CA PRN Number
Real	1, 3, 8, 10, 14, 21, 22, 32
Fake	1, 3, 8, 10, 14, 21, 22, 27, 32
Common	1,3, 8, 10, 14, 21, 22, 32

Table 5.8: GPS PRNs identifiers showing available satellite signals during the spoofed and non-spoofed scenarios.

5.4.3 Raw observables of interest

Among all the available raw data, the fields of interest for the investigations pursued within this study are the following:

- *Automatic Gain Control (AGC).* The AGC implementation in a smartphone acts as a variable gain amplifier adjusting the power of the incoming signal. Changes in the value are typical indicators of power fluctuations of the input signal in a given frequency band. AGC value and its variations affect all the received GNSS signals. Independent effects on each signal cannot be inferred from such data.

- *Carrier-to-noise Density Ratio* (C/N_0): The carrier-to-noise density ratio C/N_0 measures the strength of the useful GNSS signal w.r.t. the noise floor and has a direct relationship with the signal intensity. Strong received signals result in high C/N_0 values, typically leading to better signal tracking and PVT determination. It is also a basic indicator of the quality of a received satellite signal. Abrupt variations to it can indicate the presence of interference while an unnaturally high value could also indicate the presence of a fake satellite signal. C/N_0 is estimated by each tracking channel independently, therefore data is available for each received signal.

- *Pseudorange and Pseudorange rate* (PrM): The pseudorange is the well-known measure of the distance between the user and the satellite, affected by the bias of the user clock. The observation of the behaviour of the pseudorange and of its variation over time (pseudorange rate), allows us to directly see the effect of the spoofing signals (when they are tracked by the receiver) and to motivate any impact of the interference on the subsequent PVT solution.

For the sake of completeness, the output position estimates from the computed PVT solution will also be investigated to provide evidence of the vulnerability of the devices under test to the simplistic spoofing attack. It is worth recalling that misleading PVT solutions are indeed the actual objective of such malicious actions.

5.5 Spoofing effects on smartphones raw measurements

In this section, we analyze the effects of the designed spoofing test on raw data and position estimation provided by the devices under test scenarios S1, S2 and S3. The effects of the spoofing signals are hereafter reported following the signal processing flow of a conventional GNSS receiver architecture, i.e., from the AGC to the PVT computation.

5.5.1 AGC

Automatic Gain Control (AGC) is an audio pre-processor which automatically normalizes the output of the captured signal by boosting or lowering input from the microphone to match a preset level so that the output signal level is virtually constant. AGC can be used by applications where the input signal dynamic range is not important but where a constant strong capture level is desired. An application creates an Automatic Gain Control object to instantiate and control an AGC engine in the audio framework. To attach the AutomaticGainControl to a particular AudioRecord, specify the audio session ID of this AudioRecord when creating the AutomaticGainControl. The audio session is retrieved by calling AudioRecord.getAudioSessionId() on the AudioRecord instance. On some devices, an AGC can be inserted by default in the capture path by the platform according to the MediaRecorder.AudioSource used. The application should call AutomaticGainControl.getEnable() after creating the AGC to check the default AGC activation state on a particular AudioRecord session.
It is observed that the effect of turning on the spoofer is similar to what in-band jamming or interference would do. Due to the presence of powerful spoofing signals, the receiver reduces the amplification of the incoming signal which, while disturbing real signals, allows fake signals to be easily acquired. It is seen as technically possible to counteract easily since the time flag of the spoofed signal is not strictly synchronized with real-time GNSS signals. For AGC, depending on the front-end quantization of a receiver it affects the C/N_0 with different sensitivities and COTS receivers generally have lower bit quantizations. Hence building on the correlation, a parameter equating to the AGC to C/N_0 ratio of its absolute values is observed. Across different Android smartphones, slightly different levels of AGC and C/N_0 are observed depending on the front-end and digital signal processing blocks on similar test conditions. Hence this parameter standardizes the power of a signal at

the receiver to an extent taking into account the only variable available in Android devices currently to consider the front-end stage. By identifying the AGC to C/N_0 response of the receiver's front-end to RFI events, we could be able to draw a threshold that will allow us to discriminate between jamming events and spoofing attacks. While spoofing attacks lead to a drop of the AGC when they appear within the band, the way they are generated is different because of their respective nature. For a non-intentional RFI attack, the signal is not consistent with the satellite and noise is added to the targeted GPS band, which leads to a drop of the C/N_0 of the tracked signal. Conversely, during a spoofing attack, the signal is generated to look like a GPS signal. Thus, it increases the power of the carrier signal and so it leads to a raise of the C/N_0 value.

5.5.2 Carrier-to-Noise Density Ratio

The carrier-to-noise ratio, often known as CNR or C/N_0, is a measurement used in communications that compares the strength of the received carrier to that of the received noise. In comparison to low C/N_0 ratios, high C/N_0 ratios offer superior reception quality and typically improved accuracy and reliability in communications. If a signal contains too much noise, the receiving device may not be able to distinguish between the noise and the signal itself. High C/N_0 ratios are a sign of low bit error rates from one end of the digital communication to the other because C/N_0 is calculated as a form of digital transmission. The decibel difference between the desired signal's carrier power and the sum of its received noise power is known as the C/N_0 ratio. The carrier-to-noise ratio in dB is calculated using the following formula if the incoming carrier strength is Pc and the noise level is Pn, both of which are measured in microwatts:

$$C/N_0 = 10log_{10}(P_c/P_n) \tag{5.6}$$

The C/N_0 ratio and signal-to-noise ratio (S/N) are mathematically related measurements that both describe the effectiveness of a communication channel. Most of the time, real circumstances, such as assessing the signal-to-relative noise of twisted-pair copper cable, Wi-Fi, and LTE/5G, give more meaning to the S/N ratio standard. In satellite communications, the C/N_0 ratio is frequently used to gauge the amount of noise received. Figures 5.18 and 5.19 represent GNSS receiver C/N_0 and PrM values of GPS signals for the entire test duration. As can be seen in Figure 5.18 on the left, under non-spoofing conditions C/N_0 spreads approximately 10-45 dB-Hz between the good and the poorly received satellites up to 150 seconds. This is usually due to the fact that signals from high and low-elevation satellites and atmospheric effects. Satellite elevation has a direct relationship with the received signal strength and pseudorange distance as seen in Figure 5.18 on the right. It is clear that the spoofer acts as a source of interference over the L1 frequency band disturbing the healthy satellites during the spoofing time span and tracking of low elevation satellites being lost. Thus, it increases the power of the carrier signal and so it leads to a raise of the C/N_0 value. It is important to highlight the fact that spoofed signals from fake satellites are steadily synchronized from 150 seconds to 500 seconds, while C/N_0 changes by 35-45 dB-Hz. A similar conclusion is reached in Figure 5.19 which also shows a jump in the carrier-to-noise ratio when simplistic spoofing begins. From the graph, it is seen that C/N_0 of all PRNs share a common behaviour throughout the spoofing attack because they come from the same source. When we calculate the correlation coefficient between all pairs before and after spoofing, the correlation coefficient is much smaller during the spoofing period than in real time. This could also be an indication that they come from the same transmitter. Our results shed new light on the development of methods to detect simplistic spoofing based on the anomalies in the raw data measurements. A simplistic spoofer is a cheap low complexity spoofer and it is not jamming and spoofing L5 but if smartphones only use L1 because of low complexity, to synchronization also L5 having multi-frequency smartphones are not provide robustness to vulnerability attacks.

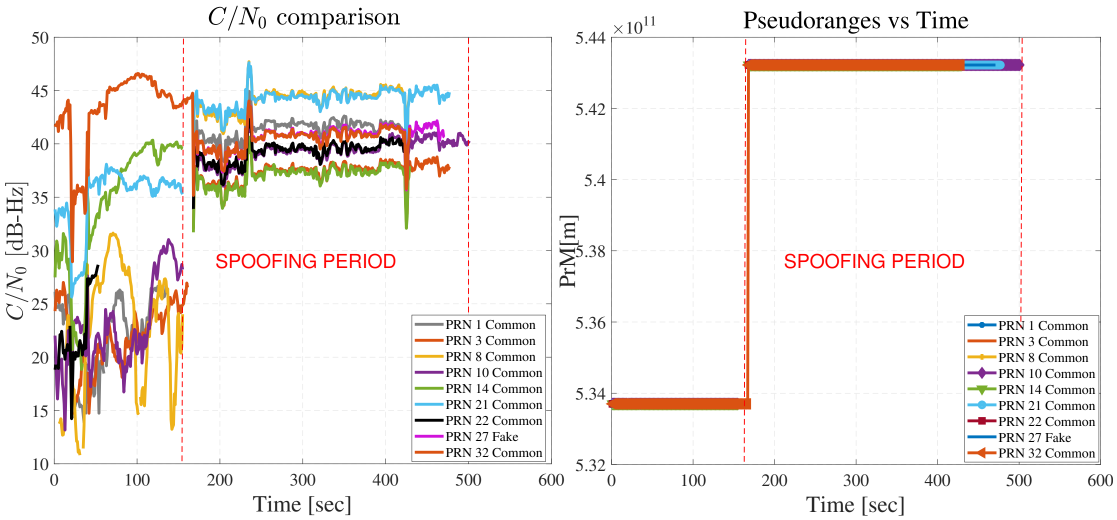


Figure 5.18: GPS spoofing Scenario 1. C/N_0 Comparison (left), The Pseudorange Ratio Comparison (right).

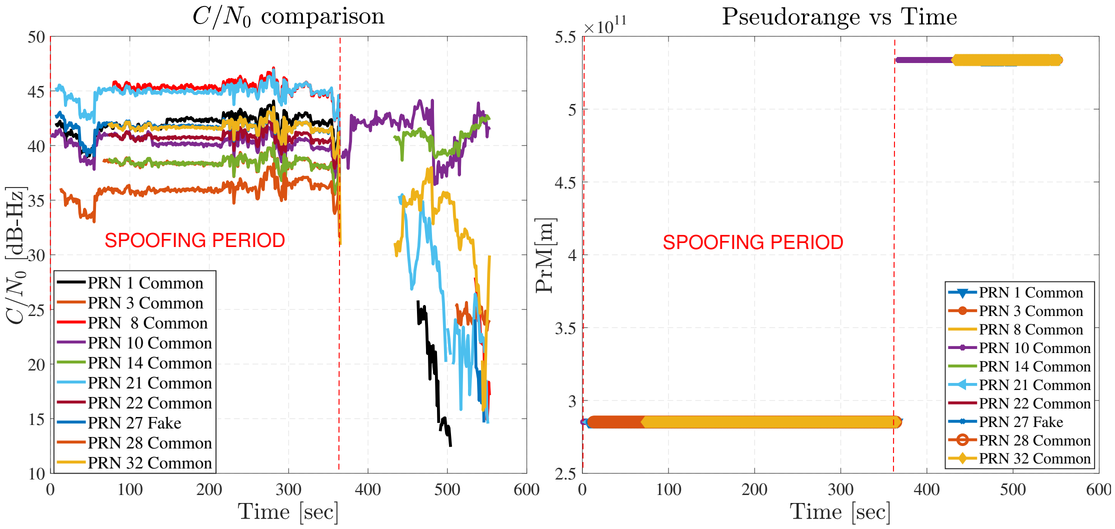


Figure 5.19: GPS spoofing Scenario 2. C/N_0 Comparison (left), The Pseudorange Ratio Comparison (right).

5.5.3 Pseudorange Measurements (PrM)

The given Figures 5.18 and 5.19 right side represent the comparison of the pseudoranges between all PRNs during the entire test period. The right plot in Figure 5.18 provides information about the general behaviour of pseudoranges. When the simplistic spoofer is turned on at 150 seconds, all common PRNs' pseudoranges started to increase dramatically, leading to the smartphone deviating position from actual to fake coordinates. Further, it can be seen that the stabilization of PrM remains for 350 seconds during the spoofing period. An interesting finding is that when the signal is switched from real to spoofing, there is a high jump in the pseudoranges that is not due to the change in position since the change in position would not justify such a high jump. The reason for this is the different time biases. In fact, we can see the dynamics of different pseudoranges. The important thing here is that the smartphone does nothing to detect this and that it actually indicates a fake position.

5.5.4 Effect on smartphone PVT estimation

The previous tests carried out in [85], [41], by the authors where the HackRF One™ spoofer was simply switched on within range of the smartphones without considering the network connections of the smartphones nor the real and fake satellite skyplot commonalities. It represented a very simplistic spoofing test to find out the vulnerability of smartphones. Observing an NMEA log, which takes the position of the smartphone directly from the Android Fused Location Provider (FLP) API, it was seen that none of the smartphones were spoofed in this test and they only acted as a jamming type of RF interference, degrading the quality of the PVT solution as seen before in [85], [41]. Some of the smartphones did not acquire the spoofed signals at all and for the rest, the spoofed signals were filtered out due to different considerations, intended or unintended. The 'State filter' indicates the GNSS signal measurement state, which is a value provided with each satellite signal measurement for every epoch. For the value to be passed by the filter, two conditions must be met, i.e. the signal's code should be locked by the receiver and the Time of Week (TOW) should be decoded. As reported the spoofed signal passed this filter in some of the phones. The next inherent defence mechanism of the smartphone is the 'TimeNanos' parameter provided by the GNSS receiver front-end of the smartphone. It is the internal hardware clock value in nanoseconds and one of the smartphones fail to quantify it under the effects of spoofing. This causes errors in all time computations towards the PVT algorithm and acts as an unintentional defence mechanism against simplistic spoofing. In some phones even if the spoofed signal passes through these stages and the pseudoranges are acquired, a Weighted Least Squares (WLS) solution cannot be acquired by combining both fake and real satellite measurements since there is an enormous difference in the magnitude of the pseudoranges. This is due to the internal GNSS hardware clock time being used to compute the pseudoranges and since simplistic spoofing does not account for real-time transmission, indiscriminate values of the pseudoranges are seen for the fake satellites.

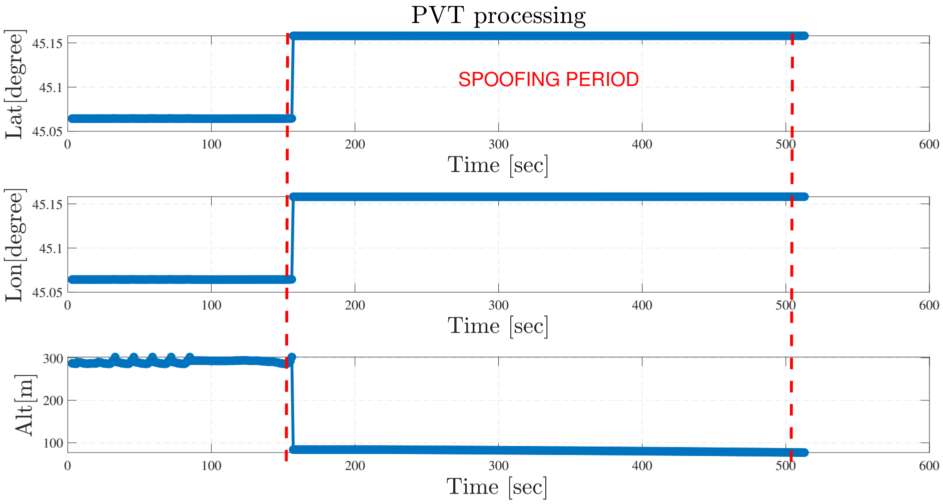


Figure 5.20: Effect of spoofing on Geodetic coordinates. Scenario 1.

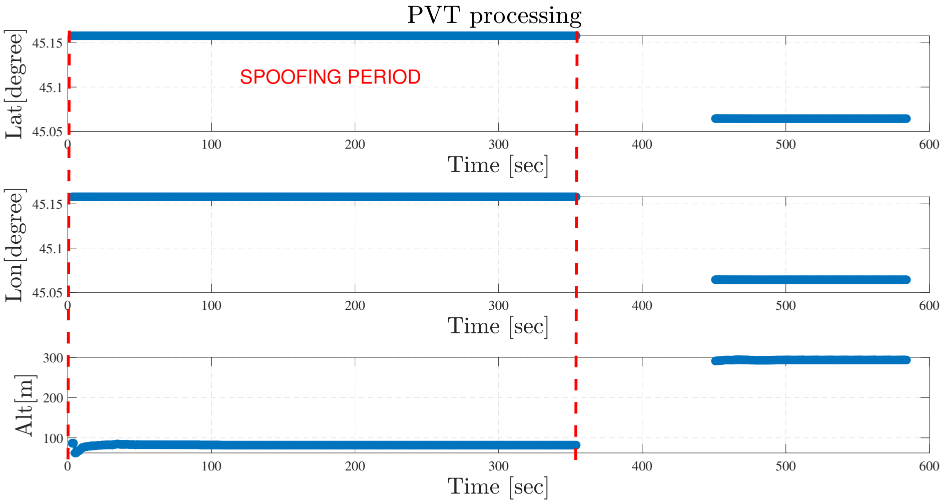


Figure 5.21: Effect of spoofing on Geodetic coordinates. Scenario 2.

Based on the results of the previous test campaign [85], [41], we decided to modify the test conditions in order to reach the optimal scenario where an Android smartphone could be spoofed by a simplistic spoofer and perform a new test campaign. For this, prior to the test, the smartphones were kept in pilot/airplane mode for at least 2 hours with the smartphones devoid of any network and GNSS signals. This could enable the internal GNSS receiver to discard any ephemeris downloaded by its network or demodulated through GNSS signals. Android FLP API works as a hybrid method between the GPS Location Provider (GLP) and Network Location Provider (NLP) sources [86]. The network-based location uses either cell towers or wireless network information to determine the location of a device. Algorithms could be available in the FLP API which might detect the anomaly in a large difference in the time and position information from both GLP and NLP. Furthermore, the spoofed signal broadcasted was of the satellite sky condition available at the time the airplane mode was switched on for the smartphones. The test was started with the spoofer turned on first followed by switching on only the GNSS receiver function of the smartphones. The phone does not have access to any type of network connection, i.e. airplane mode, and the only source for positioning and timing comes from the GPS engine. It is seen that both the time and locations of all the smartphones under investigation were successfully spoofed on observing the NMEA stream. Figures 5.20 and 5.21 represents the change in latitude, longitude and altitude in the NMEA log files of the smartphones both for Scenario 1 and 2. It can be seen that there are sudden jumps from the real location to the simulated position when spoofing starts. The vertical dotted line in the figure corresponds to the start and end of the spoofing period. In both cases, we observed that smartphone moved to the fake position. In Scenario 1 missing geodetic coordinates are noticed at 300 seconds when simplistic spoofing end. In contrast, Figure 5.21 also showed discontinuities between 350 and 450 seconds on the estimated latitude, longitude and altitude coordinates in Scenario 2. These are due to the misalignment between real and simulated timescales which also depends on the internal logic and the smartphone position algorithms. Overall, it has been seen that the receiver has no defence against the simplistic spoofing attack with the latitude, longitude and altitude changing to that of the spoofed coordinates hence validating the spoofing mechanism employed on a smartphone.

Appendix A

HackRF One

A.1 Introduction to HackRF One

The HackRF One is a device used for RF-related experiments and measurements covering a frequency range from 1 to 6000 MHz and numerous radio bands. It is used for experiments and measurement setups with open source programmes for SDR, for the development of radio communication software or for radio field measurements.
HackRF provides a maximum sampling rate of 20 MS/s and is also capable of measuring broadband signals such as WFM, DECT and Wi-fi. The analogue-to-digital converter (ADC) operates at 8 bits offering a dynamic range of 48 dB. The digitised I/Q data is handled by a Xilinx Complex Programmable Logic Device (CPLD) and an integrated ARM Cortex processor.
Due to the type of design and components adopted, the device only supports half-duplex operations.

The Printed Circuit Board (PCB) is made of 4 layers and all components are SMD; in addition, jumper sockets provide General Purpose Input/Output (GPIO) signals as well as input and output parameters of the CPLD. A JTAG connector can be used to programme the ARM Cortex processor.

The HackRF One board is very flexible in the choice of the signal path and the RF switch at all critical connection points allows the selection of various components according to user programming. After the antenna input, there are two amplifiers, one for input and the other for output (GaAs MMIC MGA-81).
The amplifier block is followed by a low-pass filter and a high-pass filter, which can be used to limit the signal in the input and output path. After the filter, the signal arrives at an RFFC 5072 RF mixer, which can be used up to 6 GHz.

The signal is mixed up or down, depending on the user's programming, and finally transferred to a baseband circuit. The mixer and filters can be bypassed by additional RF switches, allowing the IF signals (Intermediate Frequency) to be switched directly to the amplifiers or directly to the antenna. A Maxim MAX2837 component is used as the baseband chip, covering a frequency range from 2.3 to 2.7 GHz. The chip uses monolithic filters that provide a very linear signal and a low noise figure. The IQ data is then passed to a Maxim MAX5864 ADC/DAC chip. This ADC as well as the DAC have an 8-bit resolution; the converters support a maximum sampling rate of 20MS/s. Finally, the digital signals are passed to the XC2C CPLD. The entire system and all interfaces are controlled by an ARM Dual Core Cortex processor (NXP LPC4320). [87, 88]

In this thesis, HackRF One is used to output the spoofing signal, which is generated and subsequently uploaded to this device by a Raspberry Pi 4 board (a mini PC for hardware projects).

Figure A.1: HackRF One device.

Figures A.2 and A.3 show, respectively, the structure of the HackRF One device without casing and a block diagram of it, and some aspects of HRF1 will be discussed in more detail in the following subsections

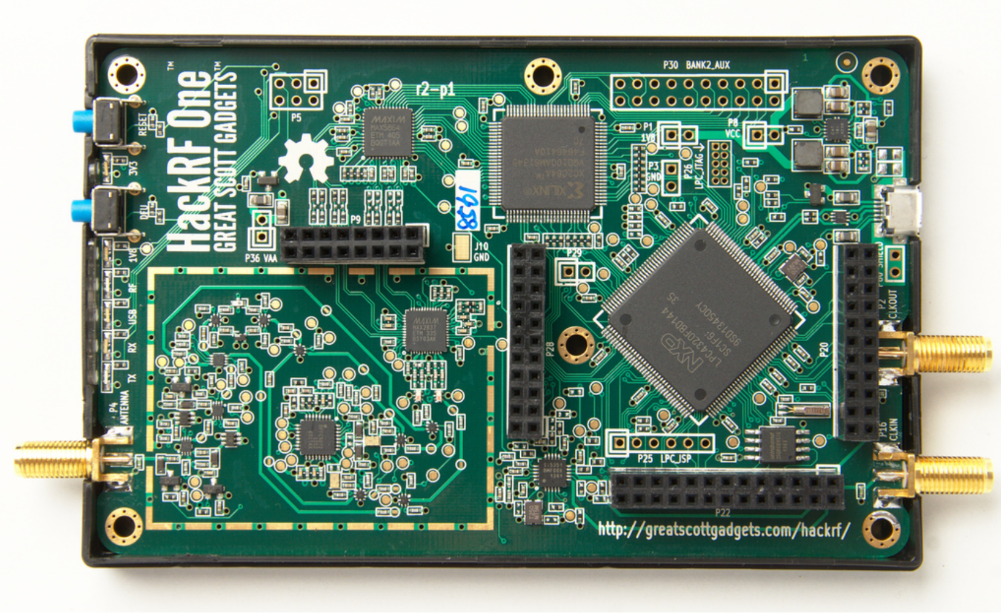


Figure A.2: HackRF One device without outer casing.

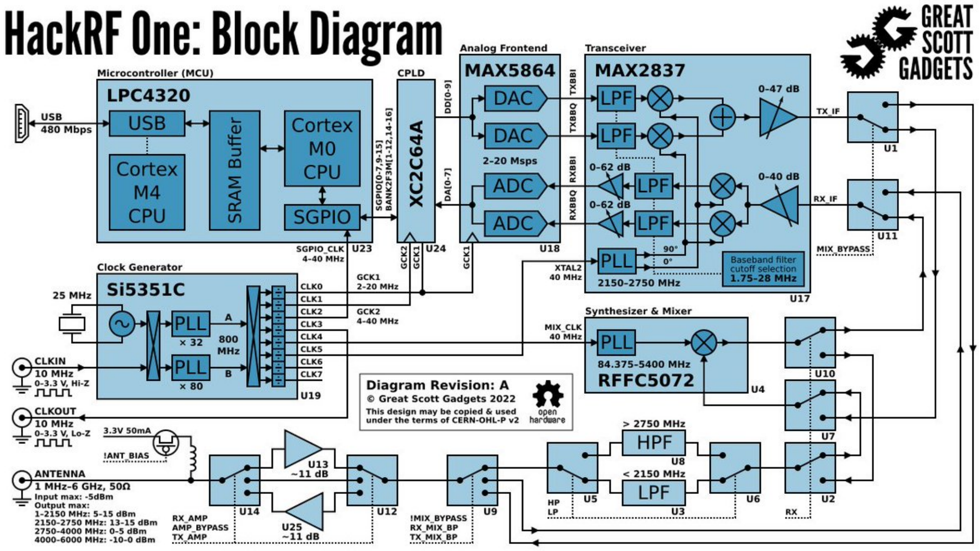


Figure A.3: HackRF One block diagram. [89]

Now its features and functionality will be discussed in more detail. The HackRF One (HRF1) device is a Software Defined Radio (SDR) device that is capable of receiving or transmitting signals only one at a time instead of receiving and transmitting at the same time and therefore operates in half duplex mode. HRF1 has the ability to digitise radio signals received or transmitted by the device and is used to receive and transmit signals between the frequencies 1 MHz and 6 GHz. Within this frequency range are most devices operating with Bluetooth, near-field communication (NFC), cellular technology and FM radio. The device is used in conjunction with a computer and software capable of processing SDR such as GNU Radio Companion (GRC). The HackRF One runs mainly on the Ubuntu operating system, which the GNU Radio Companion SDR uses as its frequency management system. Companion SDR is used as a means of capturing, analysing and transmitting radio frequencies generated by another hardware device.

In order to use the hardware, it is necessary to:

- Computer, required to run the software to analyse the captured data;
- Radio device, which is any device capable of receiving transmissions from HackRF One;
- HackRF One, i.e. the device used to receive and transmit signals.

In Figures A.4 and A.5 the meaning of the various external elements of the HRF1 device are specified.

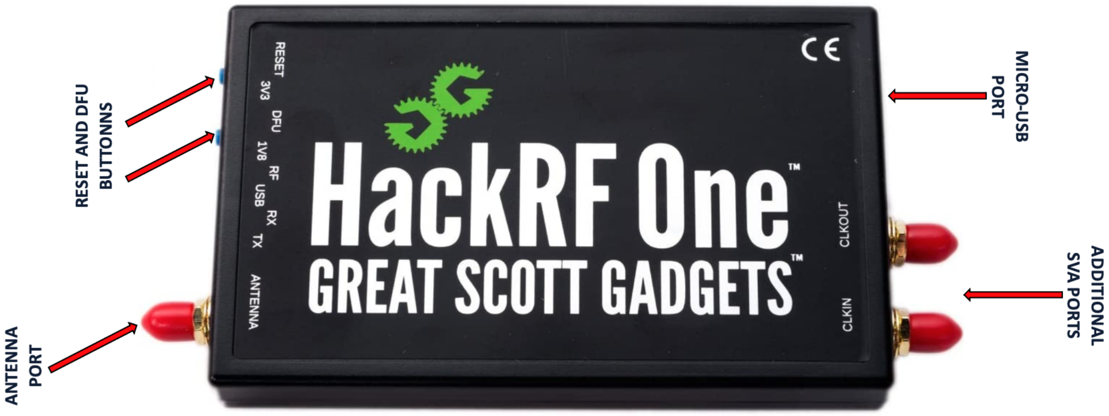


Figure A.4: HackRF One Top View.

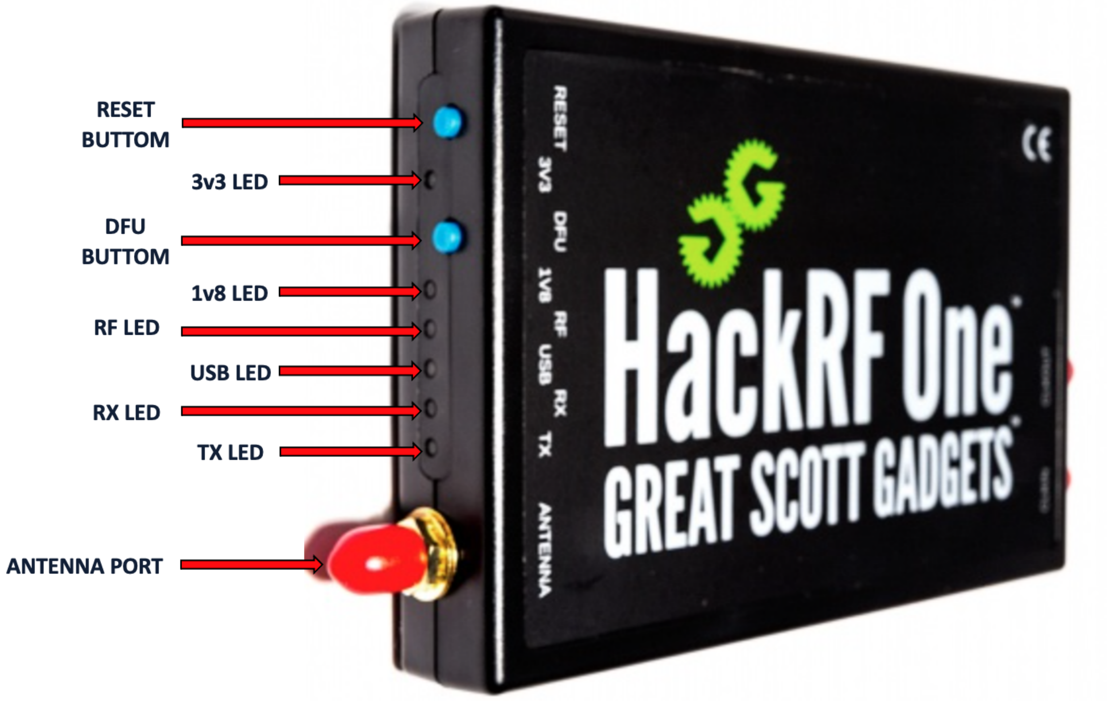


Figure A.5: HackRF One Left View.

The Table A.1, on the other hand, describes and explains when and why the LEDs outside and visible to the device are activated.

BUTTON/LIGHT	FUNCTION
Reset Button	Used to reboot the HackRF One, equivalent to unplugging the device and plugging it back in (“HackRF One", n.d.).
3v3 LED	All three of these LEDs are used to indicate power and should be lit when the HackRF One is plugged in. The various colors are used to distinguish between the multiple LEDs on the side of the HackRF One (“FAQ", n.d.).
1V8 LED	All three of these LEDs are used to indicate power and should be lit when the HackRF One is plugged in. The various colors are used to distinguish between the multiple LEDs on the side of the HackRF One (“FAQ", n.d.).
RF LED	All three of these LEDs are used to indicate power and should be lit when the HackRF One is plugged in. The various colors are used to distinguish between the multiple LEDs on the side of the HackRF One (“FAQ", n.d.).
USB LED	Indicates that the HackRF One is communicating over USB (“FAQ", n.d.).
DFU Button	Used to install or update the firmware if it is not working properly or has never been installed (“HackRF One", n.d.).
RX LED	An orange light that indicates that the device is receiving information (“FAQ", n.d.).
TX LED	A red light that indicates that the device is transmitting information (“FAQ", n.d.).

Table A.1: HackRF One LED Lights Meaning. [90]

A.1.1 Sampling Frequency

The sampling rate is defined as the number of samples per second that is measured and stored in digital form of an analogue signal, it is described in hertz and the following calculation must be carried out to obtain it:

$$\text{Sample per Second} = \frac{\text{Sampling Period}}{\text{Sampling Rate}}$$

During the acquisition phase of an audio signal, it is necessary to identify the ideal sampling frequency since there is no fixed correct value, in fact, it depends on the type of information to be acquired. The Figure A.6 shows the wave of an analogue signal and the sampling of the signal itself. [91]

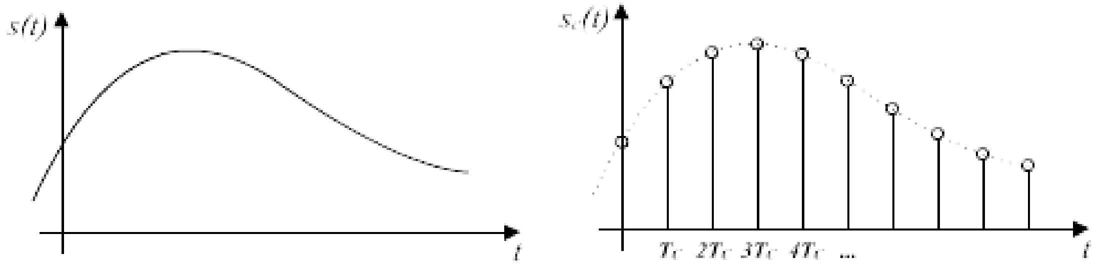


Figure A.6: The wave of an analogue signal and signal sampling. [92]

A.1.2 Half-Duplex Operations

Duplex refers to the ability of two devices (or points) to communicate with each other. In a duplex communication system, you, therefore, have two devices that are able to transmit and receive information bidirectionally. There are two types of duplex devices: Full-Duplex and Half-Duplex. HackRF One belongs to the second category. In Half-Duplex operations, communication is provided in both directions, but the transmission and reception of information must occur alternately; while one point is transmitting, the other only needs to receive, waiting for the transmitter to stop transmitting before it can respond. [93]

A.1.3 MAX5864 Analog Front End

On the next page the ADC and DAC type converters will be introduced since both are used within the MAX5864 analogue front-end integrated into the HRR1 device. The MAX5864 analogue front-end is ideal for portable communication equipment and has two 8-bit receive ADCs and two 8-bit transmit DACs. Both categories of converters operate simultaneously or independently in Frequency-Division Duplex (FDD) and Time-Division Duplex (TDD) modes, and a 3-wire serial interface controls the power-down and transceiver operating modes. The structure of the MAX5864 Analog Front End is shown in Figure A.7. [94]

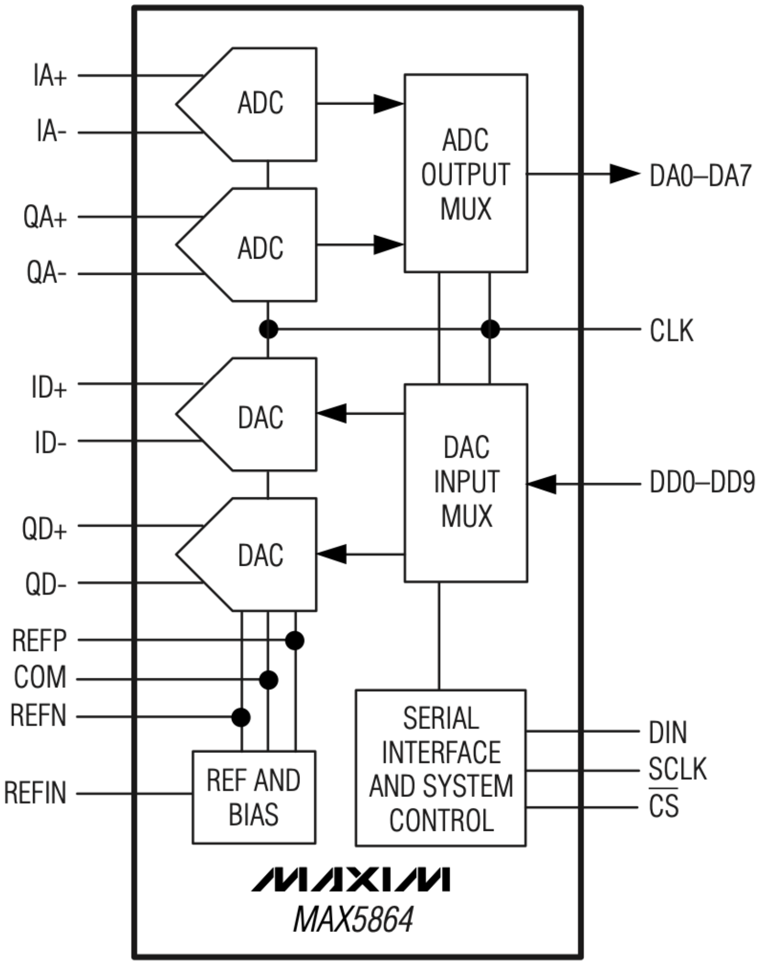


Figure A.7: Functional diagram of MAX5864 Analog Front End. [94]

Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC)

ADCs are electronic circuits that convert incoming analogue signals into the corresponding number expressed in binary so that the acquisition system can process, display, store and analyse them. The Resolution of an ADC is expressed in bits and indicates the number of discrete values the converter is capable of producing. In the case of the HRF1 device, the resolution is 8 bits so the ADC is capable of encoding an analogue input in 256 discrete levels ($2^8 = 256$). [95, 96]

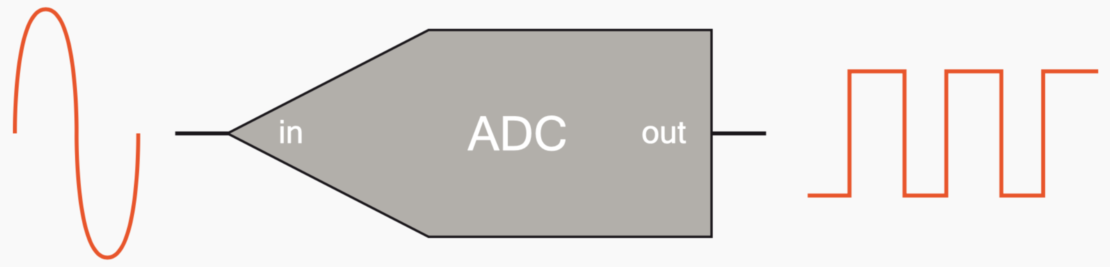


Figure A.8: How an ADC works. [97]

A comparison of the main ADC types is given in the table A.2.

Type ADC	Advantages	Disadvantages	Resolution	Sampling
Dual Slope	Cheap	Slow	20 bits	100 Hz
Flash	Very Fast	Low Resolution	12 bits	10 GHz
Pipeline	Very Fast	Limited Resolution	16 bits	1 GHz
SAR	Good S/R Ratio	No Anti-Aliasing Filter	18 bits	10 MHz
Sigma-Delta	High Dynamics	Hysteresis	32 bits	1 MHz

Table A.2: Comparison of the Main ADC Types. The S/R represents the Speed/Resolution Ratio and the Resolution and Sampling columns show the maximum value that can be obtained. [97]

As far as DAC devices are concerned, they are converters that receive as input a decimal N number encoded in binary via n inputs and convert it into an analogue signal; it, therefore, has the opposite task of an ADC type converter. Within the HRF1 device, 10-bit DACs are used, i.e. more particulate converters used for precision control. 12-bit DACs are used because the manufacturers did not want to slow down the device's processing speed. [98]

A.2 Setting up the HackRF One Device and work environment

To set up the HackRF One device, the operator should:

- Connect the antenna to the antenna port to avoid any damage as using the HackRF One without an antenna may damage the hardware of the device;
- Download Ubuntu Linux or GNU Radio operating system;
- Connect the HackRF One to your computer with the Micro-USB to USB cable. Once connected the first three LED lights will come on ensuring the device has been installed correctly.

After setting up the device, the operator will need to set up the working environment as follows:

- Install the HackRF One libraries on the computer with the terminal command "sudo apt-get install hackrf" in order to start using the device;
- After installation, type the command "hackrf_info" to check that the HRF1 device is connected correctly; if it is installed correctly, the terminal should respond with "Found HackRF board".

A.3 Installing the operating system

This section will explain the steps to be able to install the operating system correctly. As mentioned in Section A.2, the software that can be used are Ubuntu Linux and GNU Radio. For this elaboration, the second operating system of the two just mentioned will be used, and to do this the operator will have to:

- Open a terminal window;
- Type the command: 'apt-get install gnuradio'. If the user is not logged in, he will have to type "sudo" in front of this command;
- Open GNU Radio Companion using the command: "gnuradio-companion";
  - After initializing GNU Radio Companion, an untitled project will automatically open, showing a blank white space in the middle where the user can build his flowchart.

  - Once the user has drawn the flowchart he can run it using the play button located on the toolbar. If the user wishes to terminate the programme during execution, he must use the red "X" in the pop-up window which will appear during execution of the diagram.

A generic flowchart can be divided into 4 parts:

- Options, available by default when starting GNU Radio Companion (GRC);
- Variable, also available by default at GRC start-up and is used to set the sampling rate for HRF1;
- Functions, which provide various commands to perform specific tasks and comprise the most essential parts of the software;
- Arrows, which connect the function blocks to create the flowchart.

Bibliography

[1] European Global Navigation Satellite Systems Agency (EUSPA). *What is GNSS?* Available at `https://www.euspa.europa.eu/european-space/eu-space-programme/what-gnss`. 2021 (cit. on p. 5).

[2] Gary Johnston, Anna Riddell, and Grant Hausler. «The international GNSS service». In: (2017), pp. 967–982 (cit. on pp. 5, 11).

[3] Wikipedia. *Sistema satellitare globale di navigazione — Wikipedia, L'enciclopedia libera.* Available at `https://it.wikipedia.org/wiki/Sistema_satellitare_globale_di_navigazione`. 2020 (cit. on pp. 5, 6).

[4] Mobatime. *QUAL È LA DIFFERENZA TRA GNSS E GPS?* Available at `https://www.mobatime.com/it/article/difference-between-gnss-gps/`. 2021 (cit. on p. 5).

[5] Mondogeo. *I sistemi di posizionamento satellitare GNSS.* Available at `http://www.mondogeo.it/Pagine_200/pagina_00.html`. 2018 (cit. on pp. 6, 11, 12).

[6] Wikipedia. *Sistema di posizionamento globale — Wikipedia, L'enciclopedia libera.* Available at `https://it.wikipedia.org/wiki/Sistema_di_posizionamento_globale`. 2022 (cit. on pp. 6, 13).

[7] GPS.gov. *GPS: The Global Positioning System, A global public service brought to you by the U.S. government.* Available at `https://www.gps.gov`. 2022 (cit. on p. 6).

[8] Wikipedia. *GLONASS — Wikipedia, L'enciclopedia libera.* Available at `https://it.wikipedia.org/wiki/GLONASS`. 2022 (cit. on p. 6).

[9] glonass-iac. *About GLONASS.* Available at `https://www.glonass-iac.ru/en/about_glonass/`. 2022 (cit. on p. 6).

[10] Wikipedia. *Sistema di posizionamento Galileo — Wikipedia, L'enciclopedia libera.* Available at `https://it.wikipedia.org/wiki/Sistema_di_posizionamento_Galileo`. 2022 (cit. on p. 6).

[11] ESA. *What is Galileo?* Available at `https://www.esa.int/Applications/Navigation/Galileo/What_is_Galileo`. 2022 (cit. on p. 6).

[12] Esteban Garbin Manfredini. «Signal processing techniques for GNSS anti-spoofing algorithms». Doctoral Dissertation. Politecnico di Torino, 2017 (cit. on pp. 7–9, 11).

[13] Ing. S. Ponte. *GNSS (Global Navigation Satellite System).* Dip. Ingegneria Industriale e dell'Informazione (DIII), Seconda Università degli Studi di Napoli, 2015 (cit. on pp. 7, 8).

[14] Sabine Hemmer. *Come funziona un GPS?* Available at `https://scienzapertutti.infn.it/chiedi-allesperto/tutte-le-risposte/3226-0481-gps-global-positioning-system-2`. 2019 (cit. on p. 7).

[15] Wikipedia contributors. *Pseudorange — Wikipedia, The Free Encyclopedia.* Available at `https://en.wikipedia.org/wiki/Pseudorange`. 2022 (cit. on p. 8).

[16] Topogram. *Effemeridi trasmesse ed effemeridi precise.* Available at `https://www.topoprogram.it/normativa-catasto/terreni/html/Effemeridi_trasmesse_ed_effemeridi_precise.htm`. 2021 (cit. on p. 11).

[17] Wikipedia contributors. *Global Positioning System — Wikipedia, The Free Encyclopedia.* Available at `https://en.wikipedia.org/wiki/Global_Positioning_System`. 2022 (cit. on p. 12).

[18] Francois D. Cote, Ioannis N. Psaromiligkos, and Warren J. Gross. «GNSS Modulation: A Unified Statistical Description». In: *IEEE Transactions on Aerospace and Electronic Systems* 47.3 (2011), pp. 1814–1836. DOI: `10.1109/TAES.2011.5937267` (cit. on p. 12).

[19] Adam Chapman. *GPS Spoofing.* Available at `https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/Red_Chapman.pdf`. 2017 (cit. on pp. 13, 21).

[20] Seok Been Im, Stefan Hurlebaus, and Young Jong Kang. «Summary review of GPS technology for structural health monitoring». In: *Journal of Structural Engineering* 139.10 (2013), pp. 1653–1664 (cit. on p. 13).

[21] Jonathan A. Larcom and Hong Liu. «Modeling and characterization of GPS spoofing». In: (2013), pp. 729–734. DOI: `10.1109/THS.2013.6699094` (cit. on pp. 14, 21).

[22] Navigation National Coordination Office for Space-Based Positioning and Timing. *What is GPS.* Available at `https://www.ceotech.it/gps-spoofing-lhackeraggio-della-posizione-gps/`. 2013 (cit. on p. 14).

[23] Li Shen and Peter R Stopher. «Review of GPS travel survey and GPS data-processing methods». In: *Transport reviews* 34.3 (2014), pp. 316–334 (cit. on p. 14).

[24] *Counterpoint Technology Market Research*. Accessed: 2022-03-10. URL: https://www.counterpointresearch.com/global-smartphone-share/ (cit. on p. 15).

[25] Google LLC. *Raw GNSS Measurements*. Accessed: 2022-03-10. URL: https://developer.android.com/ (cit. on p. 15).

[26] Neil Gogoi, Alex Minetto, and Fabio Dovis. «On the Cooperative Ranging between Android Smartphones Sharing Raw GNSS Measurements». In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)* (2019), pp. 1–5. DOI: 10.1109/VTCFall.2019.8891320 (cit. on p. 15).

[27] Mieczysław Bakuła, Marcin Uradziński, and Kamil Krasuski. «Performance of GPS Smartphone Positioning with the Use of P(L1) vs. P(L5) Pseudorange Measurements». In: *Remote Sensing* 14.4 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14040929. URL: https://www.mdpi.com/2072-4292/14/4/929 (cit. on p. 15).

[28] Gérard Lachapelle and Paul Gratton. «GNSS Precise Point Positioning with Android Smartphones and Comparison with High Performance Receivers». In: *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*. Dec. 2019, pp. 1–9. DOI: 10.1109/ICSIDP47821.2019.9173062 (cit. on p. 15).

[29] Fabio Dovis. *GNSS interference threats and countermeasures*. Artech House, 2015 (cit. on p. 16).

[30] Todd Humphreys, Brent Ledvina, Mark Psiaki, Brady O'Hanlon, Paul Kintner, and Jr. «Assessing the Spoofing threat: Development of a Portable GPS Civilian Spoofer». In: *Proceedings of the Proceedings of the ION GNSS International Techniqueal Meeting of the Satellite Division, (ION GNSS+ 208) Savannah, GA, USA*. 2008, pp. 2314–2325 (cit. on p. 16).

[31] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. «On the Requirements for Successful GPS Spoofing Attacks». In: *Proceedings of the 18th ACM Conference on Computer and Communications Security*. CCS '11. Chicago, Illinois, USA: Association for Computing Machinery, 2011, pp. 75–86. ISBN: 9781450309486. DOI: 10.1145/2046707.2046719. URL: https://doi.org/10.1145/2046707.2046719 (cit. on p. 16).

[32] Jahshan Bhatti and Todd Humphreys. «Hostile Control of Ships via False GPS Signals: Demonstration and Detection». In: *Navigation*. Vol. 64. Mar. 2017, pp. 51–66. DOI: 10.1002/navi.183 (cit. on p. 16).

[33] Komal Songala, Supraja Ammana, Hari Ramachandruni, and Dattatreya Achanta. «Simplistic Spoofing of GPS Enabled Smartphone». In: *IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*. Dec. 2020, pp. 460–463. DOI: `10.1109/WIECON-ECE52138.2020.9397980` (cit. on p. 16).

[34] Jon M Anderson, Katherine L Carroll, Nathan P DeVilbiss, James T Gillis, Joanna C Hinks, Brady W O'Hanlon, Joseph J Rushanan, Logan Scott, and Renee A Yazdi. «Chips-message robust authentication (Chimera) for GPS civilian signals». In: *Proceedings of the 30th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*. 2017, pp. 2388–2416 (cit. on p. 16).

[35] I Fernandez-Hernandez, G Vecchione, and F Dıaz-Pulido. «Galileo authentication: a programme and policy perspective». In: *69th International Astronautical Congress*. 2018 (cit. on p. 16).

[36] Ling Huang and Qing Yang. *GPS Spoofing: low-cost GPS emulator, DEF CON 23*. Accessed: 2022-06-10. URL: `https://defcon.org/html/defcon-23/dc-23-index.html` (cit. on p. 16).

[37] Sherman C. Lo, Yu-Hsuan Chen, Tyler G. R. Reid, A. E. Perkins, Todd Walter, and Per K. Enge. «Keynote: The Benefits of Low Cost Accelerometers for GNSS Anti-Spoofing». In: *Proceedings of the ION 2017 Pacific PNT Meeting*. 2017, pp. 775–796. DOI: `10.33012/2017.15109` (cit. on p. 16).

[38] Damian Miralles, Nathan Levigne, Dennis M. Akos, Juan Blanch, and Sherman C. Lo. «Android Raw GNSS Measurements as the New Anti-Spoofing and Anti-Jamming Solution». In: *Proceedings of the 31st International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2018)* (2018) (cit. on p. 16).

[39] Silvia Ceccato, Francesco Formaggio, Gianluca Caparra, Nicola Laurenti, and Stefano Tomasin. «Exploiting side-information for resilient GNSS positioning in mobile phones». In: *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE. 2018, pp. 1515–1524 (cit. on p. 16).

[40] Dong-Kyeong Lee, Matthias Petit, Damian Miralles, Sherman Lo, and Dennis Akos. «Analysis of Raw GNSS Measurements Derived Navigation Solutions from Mobile Devices with Inertial Sensors». In: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, Florida*. 2019, pp. 3812–3831 (cit. on p. 16).

[41] Akmal Rustamov, Neil Gogoi, Alex Minetto, and Fabio Dovis. «Assessment of the Vulnerability to Spoofing Attacks of GNSS Receivers Integrated in Consumer Devices». In: *2020 International Conference on Localization and GNSS (ICL-GNSS)*. 2020, pp. 1–6. DOI: 10.1109/ICL-GNSS49876.2020.9115489 (cit. on pp. 17, 55, 61, 63).

[42] Fabio Dovis. *GNSS Interference Threats and Countermeasures*. Artech House; Unabridged edizione, 2015 (cit. on pp. 17–20).

[43] P. Babu, Lalitha Bhaskari, and CH. Satyanarayana. «A Comprehensive Analysis of Spoofing». In: *International Journal of Advanced Computer Sciences and Applications* (Jan. 2011). DOI: 10.14569/IJACSA.2010.010623 (cit. on p. 17).

[44] Editorial Team - Everything RF. *What is GPS Spoofing?* Available at https://www.everythingrf.com/community/what-is-gps-spoofing. 2022 (cit. on p. 17).

[45] Logan Scott, LS Consulting. *Spoofing*. Available at https://insidegnss.com/spoofing/. 2013 (cit. on p. 17).

[46] Septentrio. *What is spoofing and how to ensure GPS security?* Available at https://www.septentrio.com/en/learn-more/insights/what-spoofing-and-how-ensure-gps-security. 2020 (cit. on p. 17).

[47] G. Lopez, M. Simsky. *What is GNSS Spoofing?* Available at https://www.gim-international.com/content/article/what-is-gnss-spoofing. 2021 (cit. on p. 18).

[48] M Pini, M Fantino, A Cavaleri, S Ugazio, and L Lo Presti. «Signal quality monitoring applied to spoofing detection». In: *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*. 2011, pp. 1888–1896 (cit. on pp. 18–20).

[49] Steven M. Kay. *Fundamentals of statistical signal processing. Vol. 2., Detection theory*. Prentice-Hall PTR, Upper Saddle River, NJ, 1998 (cit. on pp. 18–20).

[50] Ali Jafarnia-Jahromi, Ali Broumandan, J. Nielsen, and Gérard Lachapelle. «GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques». In: *International Journal of Navigation and Observation* 2012 (July 2012). DOI: 10.1155/2012/127072 (cit. on pp. 21, 23, 24).

[51] Francesco Palmieri. *GPS Spoofing: l'hackeraggio della posizione GPS*. Available at https://www.ceotech.it/gps-spoofing-lhackeraggio-della-posizione-gps/. 2021 (cit. on p. 21).

[52] Xi-jun Cheng, Ke-jin Cao, Jiang-ning Xu, and Bao Li. «Analysis on Forgery Patterns for GPS Civil Spoofing Signals». In: (2009), pp. 353–356. DOI: 10.1109/ICCIT.2009.88 (cit. on pp. 21, 24).

[53] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. «On the Requirements for Successful GPS Spoofing Attacks». In: (2011). DOI: `10.1145/2046707.2046719`. URL: `https://doi.org/10.1145/2046707.2046719` (cit. on p. 21).

[54] Significato delle cose. *GPS Spoofing*. Available at `https://significa.it/gps-spoofing/`. 2021 (cit. on pp. 21, 22).

[55] Paul Montgomery, Todd Humphreys, and B. Ledvina. «Receiver-Autonomous Spoofing Detection: Experimental Results of a Multi-Antenna Receiver Defense Against a Portable Civil GPS Spoofer». In: *Proceedings of the Institute of Navigation, National Technical Meeting* 1 (Jan. 2009), pp. 124–130 (cit. on pp. 21, 23).

[56] Jon S Warner and Roger G Johnston. «GPS spoofing countermeasures». In: *Homeland Security Journal* 25.2 (2003), pp. 19–27 (cit. on p. 21).

[57] Tyler Nighswander, Brent Ledvina, Jonathan Diamond, Robert Brumley, and David Brumley. «GPS Software Attacks». In: (2012). DOI: `10.1145/2382196.2382245`. URL: `https://doi.org/10.1145/2382196.2382245` (cit. on p. 21).

[58] T. Humphreys, B. Ledvina, Mark Psiaki, B. O'Hanlon, and Jr Kintner. «Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer». In: (Jan. 2008), pp. 2314–2325 (cit. on p. 23).

[59] B.M. Ledvina, William Bencze, B. Galusha, and I. Miller. «An in-line anti-spoofing device for legacy civil GPS receivers». In: *Proceedings of the 2010 International Technical Meeting of the Institute of Navigation* (Jan. 2010), pp. 698–712 (cit. on p. 23).

[60] Hengqing Wen, Peter Yih-Ru Huang, John Dyer, Andy Archinal, and John Fagan. «Countermeasures for GPS signal spoofing». In: (2005), pp. 1285–1290 (cit. on p. 24).

[61] Xi-jun Cheng, Jiang-ning Xu, Ke-jin Cao, and Jie Wang. «An Authenticity Verification Scheme Based on Hidden Messages for Current Civilian GPS Signals». In: (2009), pp. 345–352. DOI: `10.1109/ICCIT.2009.91` (cit. on p. 24).

[62] Raspberry Pi. *Raspberry Pi 4*. Available at `https://www.raspberrypi.com/products/raspberry-pi-4-model-b/` (cit. on pp. 26, 27).

[63] Libero Tecnologia. *Cos'è Raspberry Pi e come si utilizza*. Available at `https://tecnologia.libero.it/raspberry-pi-cose-a-cosa-serve-49292` (cit. on p. 26).

[64] devACADEMY. *Raspberry Pi: cos'è, a cosa serve e versioni disponibili*. Available at `https://devacademy.it/raspberry-pi/` (cit. on p. 26).

[65] Wikipedia. *Raspberry Pi — Wikipedia, L'enciclopedia libera*. Available at https://it.wikipedia.org/wiki/Raspberry_Pi. 2022. URL: http://it.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=129564492 (cit. on p. 26).

[66] Marco Lai. *Raspberry Pi come usare linee GPIO*. Available at https://logicaprogrammabile.it/raspberry-pi-come-usare-linee-gpio/. 2018 (cit. on p. 27).

[67] Angelo Casarcia. *Cosa è FileZilla?* Available at https://www.angelocasarcia.it/cosa-e-filezilla/ (cit. on p. 28).

[68] Wikipedia. *FileZilla — Wikipedia, L'enciclopedia libera*. Available at https://it.wikipedia.org/wiki/FileZilla. 2022 (cit. on p. 28).

[69] Mathworks. *Matlab*. Available at https://it.mathworks.com/products/matlab.html. 2022 (cit. on p. 29).

[70] Mathworks. *Descrizione del prodotto MATLAB*. Available at https://it.mathworks.com/help/matlab/learn_matlab/product-description.html. 2022 (cit. on p. 29).

[71] Geekandjob. *Python, Cos'è Python*. Available at https://www.geekandjob.com/wiki/python. 2020 (cit. on p. 30).

[72] Wikipedia. *Python — Wikipedia, L'enciclopedia libera*. Available at https://it.wikipedia.org/wiki/Python. 2022 (cit. on p. 30).

[73] Ezio Melotti. *Perché usare Python,Quali sono le caratteristiche che fanno di Python un linguaggio da imparare e approfondire*. Available at https://www.html.it/pag/15608/perche-usare-python/. 2016 (cit. on p. 30).

[74] Michele Perrone. *Xiaomi Mi 8 e navigazione Dual-frequency GPS: ecco come funziona*. Available at https://gizchina.it/2018/05/xiaomi-mi-8-dual-gps/. 2018 (cit. on p. 31).

[75] EUSPA, European Union Agency for the Space Programme. *Xiaomi Mi 8 e navigazione Dual-frequency GPS: ecco come funzionaBroadcom announces world's first dual frequency GNSS receiver for smartphones*. Available at https://www.euspa.europa.eu/newsroom/news/broadcom-announces-world-s-first-dual-frequency-gnss-receiver-smartphones. 2017 (cit. on p. 31).

[76] REDAZIONE GEOMEDIA. *Posizionamento di precisione GNSS con dispositivi Android*. Available at https://rivistageomedia.it/2019070416495/Rilievo-e-localizzazione/posizionamento-di-precisione-gnss-con-dispositivi-android. 2019 (cit. on p. 32).

[77] Google. *GnssLogger App.* Available at https://play.google.com/store/apps/details?id=com.google.android.apps.location.gps.gnsslogger&hl=it&gl=US. 2022 (cit. on p. 33).

[78] Google Play. *GPS Logger, BasicAirData.* Available at https://play.google.com/store/apps/details?id=eu.basicairdata.graziano.gpslogger&hl=it&gl=US. 2022 (cit. on p. 33).

[79] Basic Air Data. *BasicAirData GPS Logger – Getting started guide.* Available at https://www.basicairdata.eu/projects/android/android-gps-logger/getting-started-guide-for-gps-logger/. 2022 (cit. on p. 34).

[80] GPSLogger. *GPSLogger for Android, A battery efficient GPS logging application.* Available at https://gpslogger.app. 2022 (cit. on p. 34).

[81] Great Scott Gadgets. *HackRF One.* Accessed: 2022-03-10. URL: https://greatscottgadgets.com/hackrf/ (cit. on p. 50).

[82] *Software-Defined GPS Signal Simulator.* Accessed: 2022-03-10. URL: https://github.com/osqzss/%7BGPS%7D-sdr-sim (cit. on p. 50).

[83] *MIT Licence.* Accessed: 2022-03-10. URL: https://opensource.org/licenses/mit-license.php (cit. on p. 50).

[84] Peter Ho. *NMEA Tools. (Version 2.7.35) [Mobile app].* Accessed: 2022-03-10. 2013. URL: https://play.google.com/store/apps/details?id=com.peterhohsy.nmeatools (cit. on p. 54).

[85] Akmal Rustamov, Neil Gogoi, Alex Minetto, and Fabio Dovis. «GNSS Anti-Spoofing Defense Based on Cooperative Positioning». In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020).* Oct. 2020, pp. 3326–3337. DOI: 10.33012/2020.17565 (cit. on pp. 55, 61, 63).

[86] Damian Miralles, Dennis M. Akos, Dong-Kyeong Lee, Andriy Konovaltsev, Lothar Kurz, and Sherman Lo. «Robust Satellite Navigation in the Android Operating System using the Android Raw GNSS Measurements Engine and Location Providers». In: *2020 European Navigation Conference (ENC).* 2020, pp. 1–12. DOI: 10.23919/ENC48637.2020.9317434 (cit. on p. 63).

[87] GREAT SCOTT GADGETS. *HackRF One.* Available at https://greatscottgadgets.com/hackrf/one/. 2021 (cit. on p. 65).

[88] Wimo. *Great Scott Gadgets HackRF One SDR TRx.* Available at https://www.wimo.com/it/hackrf-one (cit. on p. 65).

[89] Great Scott Gadgets. *Hardware Components.* Available at https://hackrf.readthedocs.io/en/latest/hardware_components.html. 2022 (cit. on p. 66).

[90] LCDI, Leahy Center of Digital Investigation. *HackRF One.* Available at https://www.champlain.edu. 2017 (cit. on p. 69).

[91] Prof. V. Fionda, University of Calabria. *Tecnologie Multimediali.* Available at https://www.mat.unical.it/fionda/didattica/tm1718/LEZIONE5.pdf. 2018 (cit. on p. 70).

[92] Wikipedia. *Frequenza di campionamento — Wikipedia, L'enciclopedia libera.* Available at https://it.wikipedia.org/wiki/Frequenza_di_campionamento. 2022 (cit. on p. 70).

[93] Ephesos Software. *Che cos'è il funzionamento half-duplex e full-duplex e come influisce sul router?* Available at https://it.ephesossoftware.com/articles/technology-explained/what-is-half-duplex-and-full-duplex-operation-and-how-does-it-affect-your-router.html. 2022 (cit. on p. 70).

[94] Maxim Integrated. *Maxim,Ultra-Low-Power, High-Dynamic- Performance, 22Msps Analog Front End.* Available at https://www.maximintegrated.com/en/products/analog/data-converters/analog-front-end-ics/MAX5864.html. 2003 (cit. on p. 71).

[95] Wikipedia. *Convertitore analogico-digitale — Wikipedia, L'enciclopedia libera.* Available at https://it.wikipedia.org/wiki/Convertitore_analogico-digitale. 2022 (cit. on p. 72).

[96] Edutecnica. *ADC conversione analogico-digitale.* Available at http://www.edutecnica.it/elettronica/adc/adc.htm. 2020 (cit. on p. 72).

[97] Grant Maloy Smith. *Cosa è un convertitore ADC (Convertitore Analogico-Digitale)?* Available at https://dewesoft.com/it/daq/convertitore-adc-guida-completa. 2020 (cit. on p. 72).

[98] Edutecnica. *DAC: conversione digitale - analogica.* Available at http://www.edutecnica.it/elettronica/dac/dac.htm. 2020 (cit. on p. 72).