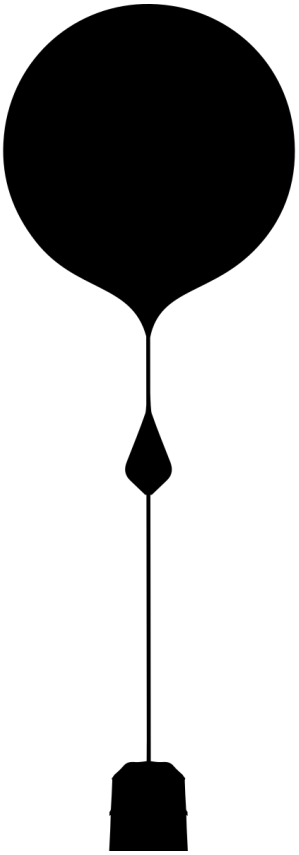
INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

# Innovative project report: RTK, you are not lost !



Tutor: G. Auriol

Yacine Benchehida
Hugo Le Belguet
Agathe Lièvre
Leila Meneceur
Assia Nguyen

ISS - 2021/2022

# Table of contents


**Introduction** **3**

**Background** **3**
- Glossary 3
- Problem 4
- Expectations and strategy 4
- Solution 4
- Diagram of the model 5

**GPS vs DGPS vs RTK** **6**
- GPS 6
- Measure on pseudo-random code 7
- Measure on carrier phase 9
- Different positioning modes 10
- DGPS 11
- RTK 13

**RTK architecture** **15**

**Types of frames** **16**
- NMEA 16
- RTCM 17

**Material** **18**

**Our work** **19**
- Tests 19
- Our solution 21

**Management method** **23**

**Security measures** **25**

**Challenges and lessons learned** **26**

**Conclusion** **27**

**Appendix** **27**
- Demo video 27
- Code 27
- Security code 28

**References** **29**

# 1. Introduction

Everyone is used to standard GPS geopositioning in their car, in their phone or in their computer. Standard GPS receivers have an accuracy between 2 to 10 meters, and only in outdoor conditions. As a group of 5 students, we teamed up to answer a problem raised by our tutor, professor Guillaume Auriol: to geolocate meteorological balloons, in association with a Paul Sabatier team led by professor Hassan Sabbah. As you can guess, the standard GPS is not accurate enough to precisely locate the balloon. So, our tutor was interested in a Real-Time Kinetics (RTK) solution since its accuracy is far more precise. The launch of the ArduSimple starter kit LR offered a relatively low-cost RTK module. Therefore, the aim of our project is to assess this solution to see if it is compatible with the needs of the project.

In this report, we will first describe the background to the project. Then, we will explain why we want to use RTK technology. Next, we will detail what RTK technology is and what kind of frames are used. We will also present the equipment we used and our tests and results. Finally, we will talk about the project management method we used over the course of this complex, semester-long project.

# 2. Background

In this first section, we will describe the context of the project. Our project is a subproject of a bigger project led by the Paul Sabatier University and the CNES (The National Center for Space Studies). This project is segmented into several subproject, each managed by a different Paul Sabatier team under the supervision of the CNES, which are:

- a subproject creating the structural design of the cube, called the Cubesat, transported by the helium balloon.
- a subproject dedicated to the choice and the assembling of electronic components like sensors and motors
- another subproject dedicated to the choice of the communication protocol and in the evaluation of the RTK module for a precise geopositioning of the cube transporting the electronic payload

The team managing the latter segment also deals with the evaluation of the RTK module. We kept in contact with them throughout the project to update each other on our progress and problems.

## A. Glossary

In the geo-positioning world, there are a lot of technical terms, we summarized a few of them here to make the text easier to read:

- **GNSS** - Global Navigation Satellite System (GPS being one of these, others include the Russian GLONASS, European GALILEO and Chinese BeiDou systems). [9]
- **Almanac** - Approximate orbits of all known satellites. [9]
- **Ephemeris** - Precise orbital information. [9]
- **Base Station** - Static unit used in RTK positioning.
- **Rover** - Moving unit used in RTK positioning.
- **L Band** - The GPS carriers are in the L band, centered at 1176.45 MHz (L5), 1227.60 MHz (L2), 1381.05 MHz (L3), and 1575.42 MHz (L1) frequencies.They are used for GPS units because they are able to penetrate clouds, fog, rain, storms, and vegetation. Only dense environments such as heavy forest canopies or concrete buildings can cause GPS units to receive data inaccurately. [7]

- **Geodetic/Geodesy** - A branch of applied mathematics concerned with the determination of the size and shape of the earth and the exact positions of points on its surface and with the description of variations of its gravity field.
- **Epoch** - A date and time from which a computer measures system time.

## B. Problem

Helium-filled weather balloons are often used to retrieve meteorological and environmental data. To collect this data, we have to send them flying up to 30 km from their launch location. The problem comes when the balloons deflate and land on the ground in unpredictable places. Researchers need to be able to locate them with precision so they can retrieve the payloads attached to the balloons. If the balloons fall in the middle of a forest, the precision of a few meters obtained from the usual GPS technology is not enough. Having a range of a few centimeters would be optimal to retrieve the balloons.

## C. Expectations and strategy

The Paul Sabatier team had some expectations for our project:

1. To send the balloon and set up the Base Station in an empty space, e.g. a field, near Toulouse.
2. To set up a first base station at the launching place and set up a second Base Station at the landing area predicted by the CNES. This second Base Station will locate the fallen balloon and its payload.
3. The Xbee modules provide communication between the Base Station and the Rover with a range up to 15 km.
4. When the communication is lost, the Rover's RTK board can store 50 000 position data points before reconnecting with the second Base Station and live-feeding it its position.
5. Because the Cubesat containing the RTK board is only 20x20 cm, the location system must be highly accurate.

## D. Solution

To try to solve this problem, we could use Differential GPS (DGPS) or RTK technology. DGPS is accurate down to less than a meter, while RTK can locate objects to within a few centimeters. Both systems use a fixed 'base' station and a mobile module called a 'Rover'. The difference between the two technologies is the data used to calculate the location. While DGPS uses the pseudorange, RTK uses the carrier phase, which we will discuss in greater detail in the next section. The drawback of RTK technology is that until recently, it had been very expensive.

With the launch of the RTK starter kit made by ArduSimple, a relatively low-cost solution was now available and the problem was solved. This kit is fully functional and ready to be tested. After our first meeting with our tutor, we defined our objectives and tasks:

1. To understand the RTK technology more deeply by doing some research before even using the kit. Our research can be combined in a state of the art about the architecture and the communication used by RTK systems. We can also add a comparison with the DGPS to learn what makes a technology more accurate than another in positioning.

2. To use and test the hardware kit. Since the boards are already flashed we can deploy it without touching the software.

3. The kit is sold with a Xbee module which has a medium range. We already know we can achieve a range of 1.5 km in an open field. We first need to measure the performance and range of this kit outside with obstacles.

4. To obtain an RTK accuracy of a few centimeters.

5. To display both the Base Station and the Rover's position.

Over the course of the project, our objectives evolved somewhat . We tested the Xbee module and found out that its range is largely sufficient for our project. So, we may not need to replace it with a LoRa module. We also met the Paul Sabatier team and they were able to present us with every aspect of the project from their point of view.

## E. Diagram of the model

Their final launch setup is presented here:

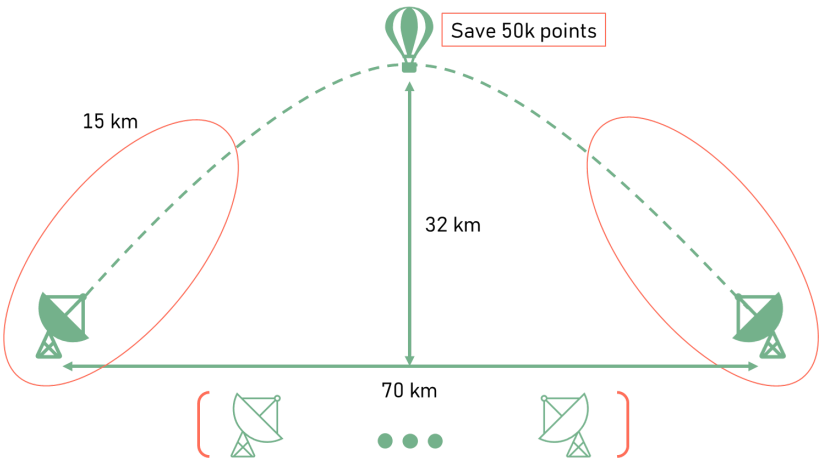



*Figure 1: diagram of the project*

They plan to have 2 stationary base stations that stay on the ground : one where they launch the balloon and one in the approximate area where it lands. In the balloon, they plan to place the rover. The bases and the rover communicate through Xbee modules. The rover will send its position to the first base up to a distance of 15km. After that, when the rover is out of range and cannot communicate with it, it will save 50 000 points of position until it can reach the second base station.

# 3. GPS vs DGPS vs RTK

In this section, we will discuss why we chose RTK technology over DGPS. But first, we will see why DGPS (DGNSS using only GPS constellations) and RTK are more accurate than normal GPS (or more generally, GNSS).

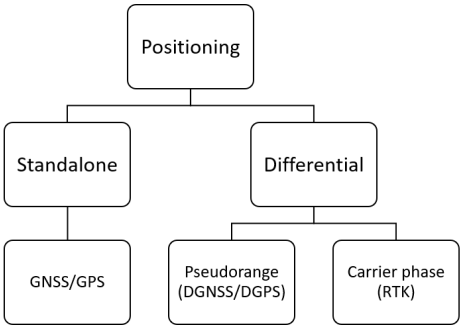



*Figure 2: classification of the technologies*

## A. GPS

[4] To determine the position of a user on Earth, satellite navigation systems use the principle of multilateration. This principle can calculate the geometrical distance between a receiver on the ground and an emitter in a satellite by measuring the propagation time of a signal emitted by the satellite. With the distances on a constellation of three satellites, we can determine the position of the receiver as the intersection of 3 spheres. To resolve the problem of the temporal unknown of desynchronization between the clocks on the satellites and the clock on the receiver, we need to consider the intersection of 4 spheres in a 4-dimensional space (space and time) instead of only 3 (only space). While using this method, no error corrections are made. The GNSS satellites just provide the best standard signals available. They have an accuracy of around ten meters.

To understand how a receiver can measure the propagation time of a radio signal, we have to understand the structure of a signal emitted by a satellite. A GNSS satellite emits a signal composed of a pure sinusoidal wave called a carrier wave. This wave possesses binary codes which contain information. The frequencies allocated to satellite navigation systems are in the upper part of the L band, between 1160 and 1610 MHz (with a wavelength of between 18 and 26 cm). On each frequency, there are two types of information:

- A time code based on the time of the signal's emission by the satellite (called pseudo-random code).
- A set of parameters needed by the receiver to calculate its position (almanach, ephemeris, constellation state…)

This information is transmitted by binary codes following a process called Binary Phase Shift Keying (BPSK). It consists of inverting the carrier phase according to the value of the binary code (0 or 1). A pseudo-random code is a binary sequence with a determined length and known values. The term 'pseudo-random' simply means that the binary sequence has the same stochastic behavior as a white noise. This property limits the problems created by interference and facilitates the encryption of the codes. To determine the travel time of the received signal, the receiver can use either the temporal information or the carrier phase. The process that measures the phase is more complex and more expensive to implement.

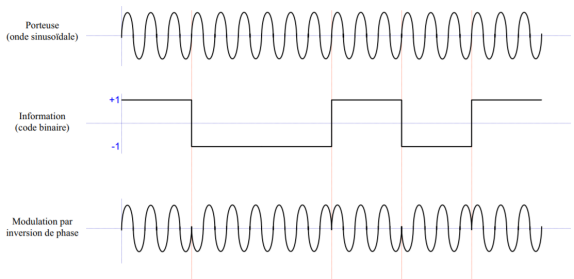



*Figure 3: binary information modulation on the carrier wave by phase inversion*

In short, a GNSS receiver is essentially composed of an antenna - which receives all the signals originating from the satellites - and a processing unit - which analyzes the signals and performs the measurements.

To carry out the measurements on the pseudo-random code and the carrier phase, the receiver must separate the signals originating from the different satellites. Since the received frequency is affected by the Doppler effect, the receiver uses long-term predictions about the satellites to calculate the received frequencies and to determine each satellite's natural frequency band at the time of the measurement. For each satellite, the measurements of the code and the phase are done independently by two lock loops:

- the code loop, Delay Lock Loop (DLL)
- the phase loop, Phase Lock Loop (PLL)

These loops match the satellite's signal (code or phase) to its replica generated by the receiver. To do this, the signal generated by the receiver is delayed until we have the maximum correlation between the two signals.

## B. Measure on pseudo-random code

[4] The pseudo-random code is a known temporal binary function. If $t$ is a value of a hypothetical reference clock, $code(t)$ is the value of the pseudo-random code at the moment $t$. The satellite and the receiver possess an internal clock that gives them their own time scale. We can consider:

- $t$, value of a reference clock
- $t_u(t)$, value of the receiver's clock
- $t^s(t)$, value of the satellite's clock
- $t - t_u(t)$ and $t - t^s(t)$, terms of desynchronization between the reference clock and those of the receiver and of the satellites
- $t_e$, signal's moment of emission from the satellite given by the reference clock (at this time, the satellite's clock has the value $t^s(t_e)$ and generates the value $code(t^s(t_e))$ of the pseudo-random code)
- $t_r$, the moment given by the reference clock when the receiver receipts the value $code(t^s(t_e))$ (at this time, the receiver's clock has the value $t_u(t_r)$ and generates the value $code(t_u(t_r))$ of the pseudo-random code)
- $\partial t(t_r)$, the delay of reception of the pseudo-random code by the receiver, it is the signal travel time and the desynchronization between the clocks

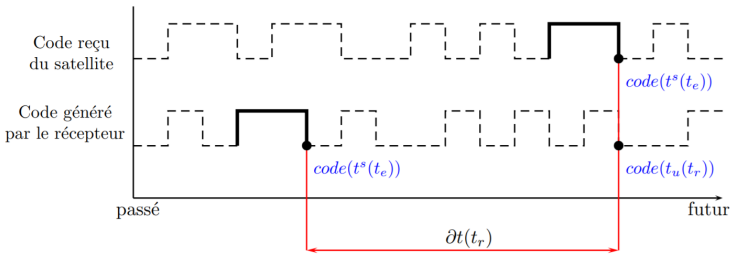



*Figure 4: pseudo-random code measurement*

The code loop matches the received code from the satellite and its replica internally generated by overlapping them until it obtains a maximal correlation peak. The offset value is simply the interval time $\partial t(t_r)$. So, the code loop calculates $\partial t(t_r)$ with the following expression:

$$\partial t(t_r) = t_u(t_r) - t^s(t_e)$$

In practice, the receiver does not give directly $\partial t(t_r)$ but its equivalent in distance called the pseudorange:

$$\rho(t_r) = c \cdot (t_u(t_r) - t^s(t_e)) + e(t_r)$$

with $e(t_r)$ the noise on the pseudorange measure

The pseudorange can be measured by the receiver in two different ways:

- simultaneity at the reception. At the moment of reception $t_r$, the receiver identifies the signal's moments of emission $(t_e)^n$ for all the $n$ apparent satellites. This is the most common way.
- simultaneity at the emission. For the same pseudo-random code transition, at the same moment $t_e$, the receiver identifies the moments of reception $(t_r)^n$ or all the $n$ apparent satellites.

Using the pseudo-random code, the receiver's accuracy to obtain its position depends essentially on the accuracy he can measure the pseudorange and on the satellites' geometry. The current GNSS receivers are able to measure with an accuracy of about a hundredth of a cycle. With the wavelength used for the pseudo-random code in current GNSS systems, the maximum accuracy a receiver can have is a few tens of centimeters.

## C. Measure on carrier phase

[4] The second method to calculate the signal propagation time emitted by GNSS satellites consists of using directly the carrier phase. A carrier wave is simply a temporal sinusoidal function. Its expression is the following: $onde(t) = A(t) \cdot sin(\varphi(t))$ with $A(t)$ the wave's amplitude and $\varphi(t)$ its phase depending on the time. With the carrier wave frequency and the original phase $\varphi_0$, we have:

$$\varphi(t) = f(t) \cdot t + \varphi_0$$

The phase loop follows the same principle as the code loop. It looks for the signals' correlation to determine the phase difference between the carrier wave received from the satellite and the one generated internally by the receiver.

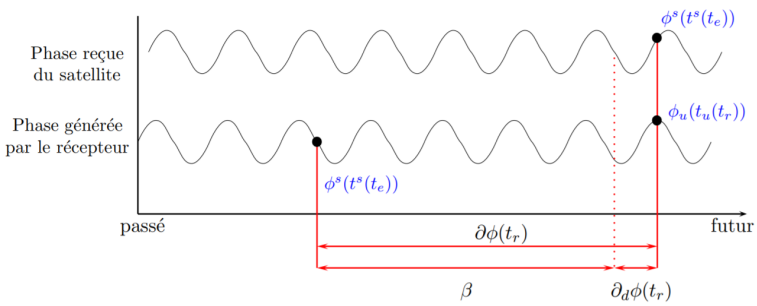



*Figure 5: principle of phase measurement on the carrier wave*

At the moment $t_e$ in the reference clock, the satellite's clock give the value $t^s(t_e)$ and generate the following phase (in cycles):

$$\varphi^s(t_e) = f^s(t_e) \cdot t^s(t_e) - \varphi^s(t_0)$$

with $f^s(t_e)$ the instant signal frequency emitted by the satellite and $\varphi^s(t_0)$ the original phase.

When this value is received by the receiver, the reference clock is at the moment $t_r$. The receiver's clock has the value $t_u(t_r)$ and generate the following phase:

$$\varphi_u(t_r) = f_u(t_r) \cdot t_u(t_r) - \varphi_u(t_0)$$

Theoretically, the phase loop can calculate by correlation the value $\partial\varphi(t_r)$ (phase difference between the signal received from the satellite and the signal generated internally):

$$\partial\varphi(t_r) = \varphi_u(t_r) - \varphi^s(t_e)$$

Practically, only the decimal part (between 0 and 2π) is obtainable by correlation because this method cannot differentiate the different cycles of the carrier.

We have $t_1$ , the time when the receiver does its first measure on the signal's phase from the satellite. At this time, the receiver measures the decimal part $\partial_d\varphi(t_1)$ of the phase difference $\partial\varphi(t_1)$. The integer part β, called the integer ambiguity, of $\partial\varphi(t_1)$ is unknown. Thus, we have:

$$\partial\varphi(t_1) = \partial_d\varphi(t_1) + \beta + e(t_1)$$

with $e(t_1)$ the noise on the phase measure

Each loop possesses a lap counter. This counter records the number of whole phase cycles elapsed since the moment $t_1$. While there are no interruptions in the signal reception, the receiver measures at each moment $t_n > t_1$, the decimal part $\partial_d \varphi(t_n)$ and the number of whole phase cycles elapsed since the moment $t_1$ noted $N_{d\varphi}(t_n)$. At the moment $t_n$, we have :

$$\partial\varphi(t_n) = \partial_{d\varphi}(t_n) + N_{d\varphi}(t_n) + \beta + e(t_1)$$

The phase measure $\Phi(t_r)$ has the following expression:

$$\Phi(t_r) = \partial_{d\varphi}(t_r) + N_{d\varphi} + e(t_r)$$

Measuring the phase is far more precise than measuring the pseudorange because current receivers can measure $\partial_{d\varphi}(t_r)$ with an accuracy of about 3 thousandth of a cycle. It is less than 0.5mm on the carrier L1 of the GPS and less of a mm on the carrier L2. The phase measure $\Phi(t_r)$ on the carrier wave is by definition ambiguous. To determine the integer value of this ambiguity is the key step of a great accuracy position. Since the '80s, a lot of methods have been developed to process this problem. All of these methods use the ambiguity's integer part but it brings a greater complexity to process the data.

## D. Different positioning modes

[4] To calculate the position with a GNSS system, we can use two different calculation methods: the absolute mode and the differential mode.

The absolute mode - The initial role of a satellite navigation system is to allow every user with their only receiver to determine in real time their position and their speed in a global reference system. It is called the positioning in absolute mode. The receiver's position is calculated by measuring the pseudorange with a minimum of 4 satellites. It directly uses the multilateral principle. At the present time, the accuracy of the absolute mode is typically a few meters. This accuracy is limited because some variables of the observation model are not well known yet (satellites' orbits, clock terms, …).

The differential mode - The positioning in differential mode calculates the user's position relatively to a reference point with known coordinates. Thus, the result is a vector whose origin is the reference point and whose end is the user's position (fixed or moving). Emerged in the '80s, the differential mode enhanced significantly the level of accuracy given by the GNSS systems. This mode is based on a simple principle: by differentiating the observations made at the same time by two close receivers, we can eliminate the common errors of the 2 receivers. This greatly simplifies the expression of the geopositioning problem. There are many different differential modes which can be classified around 3 criteria:

- the observation used (code or phase)
- the calculation method used (real time or not)
- the user's trajectory (fixed or moving receiver)

Differential positioning with the code - The positioning mode using the measurements on the code are called DGPS (or more generally DGNSS) with no distinction between a moving user (kinematic case) or fixed user (static case) and a position calculation in real time or not. The level of accuracy ranges from 0.5 m for two close receivers to 2 m for 2 distant receivers.

Differential positioning with the phase - Measuring the phase is far more accurate than measuring the pseudorange but it involves a term of integer ambiguity. This term complicates the process. The level of accuracy is linked to the determination of the ambiguity. If the ambiguities are not fixed, the highest accuracy obtained in real time differential mode is a few tens of centimeters whereas it can be a few millimeters when the ambiguities are fixed at their exact integer value.

As we saw before, there are 2 different modes: a non-real time mode and a real time mode. The real time mode, called Real Time Kinematic (RTK), calculates the receiver's position (fixed or moving) with the ephemeris data diffused by the satellites. The accuracy, typically in centimeters, depends on many factors like the distance between the Rover and the base, the number of satellites used and their geometrical configuration in space. In RTK mode, after fixing the ambiguities at their exact integer value, the phase measurements, which are now unambiguous, are used to calculate the moving receiver's position.

## E. DGPS

DGPS uses a fixed, known position to adjust real time GPS signals to eliminate pseudorange errors.

[3] In this section, we will discuss DGPS Technology and explain the kind of distance measurement it uses. The classical DGNSS technique is an enhancement to a primary GNSS system that consists of the determination of the GNSS position for an accurately-surveyed position known as a reference station. The method takes advantage of the slow variation with time and user position of the errors due to ephemeris prediction, residual satellite clocks, ionospheric and tropospheric delays. Given that the position of the reference station is accurately known, the deviation of the measured position to the actual position and more importantly the corrections to the measured pseudoranges to each of the individual satellites can be calculated. Starting from the reference station, the system computes and broadcasts corrections to either the GNSS position or to the pseudorange measurements to the DGNSS users. In order to be able to apply these corrections, the receiver has to be enabled for DGNSS and stay in the vicinity of the reference station. This ensures that the two receivers (station and Rover) observe the same GNSS satellite. The measured positions of other GNSS user receivers are thus corrected. Other uncorrelated errors (e.g. multipath) cannot be corrected by this method and specific techniques have to be applied to mitigate them. When GPS is the only constellation used for the Differential GNSS technique, the system is called DGPS. DGPS accuracy is in the order of 1 m (1 sigma) for users in the range of a few tens of km from the reference station, growing at the rate of 1 m per 150 km of separation.

[5] Now, we will see what pseudorange is and how we can calculate it. The pseudorange is an approximation of the distance between a satellite and a GNSS receiver. A GNSS receiver will attempt to measure the ranges of (at least) four satellites as well as their positions when their positional data were transmitted. With the satellites' orbital parameters supplied in the almanac within the message, each position can be calculated for any point in time. When a GPS receiver measures the time offset it needs to apply to its replica of the code to reach maximum correlation with received signal, it is measuring the time difference between when a signal was transmitted (based on satellite clock) and when it was received (based on receiver clock). If the satellite and receiver clocks were synchronized, this would be a measure of range. Since they are not synchronized, it is called "pseudorange ". The problem is that the quartz oscillators used for timing in most GNSS receivers are only accurate to 1 ppm at best. This means that if the clock has not been corrected for a week the distance error will be massive. Even when the clock is corrected, a second later the error can be hundreds of meters. Fortunately, in a GNSS receiver the clock's time is used to

measure the ranges to several different satellites simultaneously, and so all the measured ranges have the same error. Ranges with the same error are called pseudoranges. By finding the pseudorange of an additional fourth satellite, the time error can also be estimated.

The pseudorange can be calculated with the following formula:

$$P_u^s = (t_u - t^s) \cdot c$$

where $P_u^s$ is the pseudorange between the receiver $u$ and the satellite $s$, $t_u$ is the receiver's clock time, $t^s$ is the satellite transmit time and $c$ is the speed of light

This expression can be related to the true range by introducing corrections to the clock times:

$$t_u = \tau_u + \Delta t_u \text{ and } t^s = \tau^s + \Delta t^s$$

where $\tau_u$ and $\tau^s$ are true times and $\Delta t_u$ and $\Delta t^s$ are clock corrections

We can then substitutes these into the equation of the pseudorange yields:

$$P_u^s = [(\tau_u - \tau^s) + (\Delta t_u - \Delta t^s)] \cdot c$$

$$P_u^s = \rho_u^s + (\Delta t_u - \Delta t^s) \cdot c + I_u^s + A_u^s$$

where $\rho_u^s$ is the true range, $I_u^s$ is the ionospheric delay and $A_u^s$ is the atmospheric delay. The delay terms are introduced because the propagation velocity is not $c$.

The equation for the pseudorange uses the true range and corrections applied for propagation delays because the propagation velocity is not the in-vacuum value $c = 2.99792458 \cdot 10^8\, m/s$. To convert times to distance, $c$ is used and then corrections applied for the actual velocity not equaling $c$. In RINEX data files, used internationally to exchange observations from GNSS receivers, pseudorange is given in distance units. The true range is related to the positions of the ground receiver and satellite. We also need to account for noise in the measurements.

The contributions to pseudorange noises can be random or not so random errors in measurements. It can be:

- the correlation function width, it can be inversely proportional to the bandwidth of the signal. Thus, the 1MHz bandwidth of the [12] Civilian Access code (C/A) produces a peak 1 second wide (300m) compared to the encrypted Precise code (P(Y)) 10MHz bandwidth which produces 0.1 second peak (30m). A rough rule is that the peak of the correlation function can be determined to 1% of width (with caution). Therefore, we have 3m for the C/A code and 0.3m for the P(Y) code.
- a thermal noise, it is the effects of other random radio noise in the GPS bands. Since the C/A code has a narrower bandwidth, tracking it, in theory, has 10 times less thermal noise power plus a factor of 2 more because of transmission power. But, thermal noise generally has the smallest effect.
- multipath, it is the reflected signals.

The main noise sources are related to reflected signals and tracking approximations. If you have a high quality receiver, the noise will be about 10cm. For a low-cost receiver (~$200), the noise is a few meters but it depends on surroundings and antenna. In general, the C/A code pseudoranges are of similar quality to the P(Y) code ranges. C/A can use narrowband tracking which reduces the amount of thermal noise.

## F. RTK

RTK GPS employs a method of carrier-phase differential GPS positioning whereby users can obtain centimeter-level position accuracies in real time. The carrier-phase is the phase of the received carrier with respect to the phase of a carrier generated by an oscillator in the GPS receiver. That carrier has a nominally constant frequency, whereas the received carrier is changing in frequency because of the Doppler shift induced by the relative motion of the satellite and the receiver. The received carrier's phase is related to the phase of the carrier at the satellite through the time interval required for the signal to propagate from the satellite to the receiver. So, ideally, the carrier-phase observable would be the total number of full carrier cycles and fractional cycles between the antennas of a satellite and a receiver at any instant.

Unfortunately, a GPS receiver has no way of distinguishing one cycle of a carrier from another. The best it can do is to measure the fractional phase and then keep track of changes to the phase; the initial phase is undetermined, or ambiguous, by an integer number of cycles. To use the carrier phase as an observable for positioning, this unknown number of cycles, or ambiguity, must be estimated along with the other unknowns - the receiver's coordinates. If we convert the measured carrier phase in cycles to equivalent distance units by multiplying by the carrier's wavelength, we can express the carrier-phase observation equation for the pseudorange. The major difference (apart from the sign of the ionospheric propagation delay term) is the ambiguity term's presence. In fact, the carrier phase can be thought of as a biased range measurement just like pseudorange

[11] In the neighborhood of a clean-sky location, the main errors in the GNSS signal processing are constant, and hence they cancel out when differential processing is used. This includes the error in the satellite clock bias, the satellite orbital error, the ionospheric delay and the tropospheric delay. The noise of carrier measurements is much smaller than the one of the pseudo-code measurements. However, the processing of carrier measurements is subject to the so-called carrier phase ambiguity, an unknown integer number of times the carrier wave length, that needs to be fixed in order to rebuild full range measurements from carrier ones. The phase ambiguity can be fixed for dual-frequency differential measurements for two close receivers.

Although all GPS receivers must lock onto and track the signal's carrier to measure pseudoranges, they may not record carrier-phase observations for external use. Some, however, may use carrier-phase measurements internally to smooth - reduce the high-frequency noise of - the pseudorange measurements. The carrier-phase's rate-of-change is related to the Doppler shift, which is used to determine velocity. For high-accuracy position determination, carrier-phase measurements made by one receiver are typically combined with those made simultaneously by another receiver to form double differences in which the effects of satellite and receiver clock errors are essentially eliminated. The double differences are then processed using a least-squares filter of some kind to estimate the relative coordinates of one receiver (actually, its antenna) with respect to the other. If the coordinates of one of the receivers is well known in some coordinate frame, for example, it occupies a geodetic control point, then the coordinates of the second receiver can be determined in the same frame.

[5] Now, we will see what the carrier phase is and how we can calculate it. Carrier phase measurements are similar to pseudorange in that they are the difference in phase between the transmitting and receiving oscillators. The integration of the oscillator's frequency gives the clock time. A basic notion to have about carrier phase is the following formula: $\varphi = f\Delta t$ where $\varphi$ is the phase and $f$ is the frequency.

The carrier phase is the difference between phase of receiver oscillator and signal received plus the number of cycles at the initial start of tracking as you can see in the following equation:

$$\varphi_u^s(t_r) = \varphi_u(t_r) - \varphi_r^s(t_r) + N_u^s(1)$$

The received phase is related to the transmitted phase and propagation time by the next formula:

$$\varphi_r^s(t_r) = \varphi_t^s(t_t) = \varphi_r^s(t_r - \frac{\rho_u^s}{c}) = \varphi_t^s(t_t) - \dot{\varphi}^s(t_r) \cdot \frac{\rho_u^s}{c}$$

The rate of change of phase is frequency. The phase difference changes as $\frac{\rho}{c}$ changes. If the clock was perfect and nothing was moving, it would be constant. We also have to notice that the phase received at time $t$ is equivalent to the phase transmitted at $t - \tau$ .

When the phase is used, it is converted to distance using the standard L1 and L2 frequencies and vacuum speed of light. The clock terms are introduced to account for the difference between true frequencies and nominal frequencies. Like we had in the pseudorange, the ionospheric and atmospheric delays account for propagation velocity.

Nominally, phase can be measured with an accuracy of 1% of wavelength (~2 mm L1 and ~2.4 mm L2). Again, since it is affected by multipath, the ionospheric delays are ~30 m and the atmospheric delays are between 3 and 30 m.

Since the phase is more precise than the range, more effects need to be carefully accounted for with the phase. Precise and consistent definition of time of events is one the most critical areas. In general, phase can be treated like range measurement with unknown offset due to cycles and offsets of oscillator phases.

We can summarize all the information in the following table:

| | GPS | DGPS | RTK |
|---|---|---|---|
| Accuracy | ~ ten m | +/- 1 m | +/- 2 cm |
| Constellation | GPS<br>(4 satellites min) | GPS<br>(4 satellites min) | GPS, Galileo, Glonass, Beidou<br>(5 satellites min) |
| System | Receiver | Base + Rover | Base + Rover |
| Measurement method | Multilateration | Code phase/Corrected Pseudorange | Carrier phase |
| Positioning mode | Absolute | Differential (code) | Differential (phase) |
| Radio link | x | Less information<br>Slow transmission<br>Real-time/post-processed results | More information<br>Fast transmission<br>Real-time results |
| Range | x | 100 to 200 km | 10 to 20 km |

*Figure 6: comparison table of the different technologies*

# 4. RTK architecture

[8] In an RTK system, both the reference station and moving station consist of a single- or dual-frequency GPS receiver, the associated antenna, a data radio (sometimes called a radio modem), and its associated antenna. Typically, users employ identical GPS receivers and data radios at the base and Rover stations, although one must obviously use the reference station data radio for transmitting and the Rover station data radio for receiving. High-power transmitters and less costly receive-only radios are available. Often, the transmitting antenna has higher gain than the receiving antenna.

Some RTK systems integrate the GPS receiver and data radio into one package, with the GPS and radio link antennas even sometimes sharing a common enclosure. To achieve the best results, the reference station GPS antenna should be mounted in a location free, as much as possible, of multi-path. The radio link antennas should be as high as possible to maximize the link's coverage.

Some RTK installations use combined GPS/GLONASS receivers at the reference and Rover stations. Using GLONASS data in addition to GPS can provide faster ambiguity resolution and higher positioning accuracies.

The data link used to support RTK operations is usually a radio channel of some sort, although an optical data link could be used in some environments. For RTK operations that carry out double differencing using Message Types 18 and 19, the data must be updated every 0.5–2 seconds, rather than the more leisurely 10 seconds or more used with code differential operation. So, whereas the RTCM SC-104 messages for code DGPS are typically transmitted by marine radio navigation beacons at 200 bits per second (bps), the data links for RTK use need data rates of at least 2,400 bps and preferably 9,600 or even 19,200 bps. The bandwidths required to support such data rates can be found in the VHF and UHF part of the radio spectrum.

Because RTK data links operate at VHF/UHF frequencies, their use is limited, for the most part, to line of sight with the maximum distance $d$, in kilometers, that can be theoretically achieved given approximately by the following equation:

$$d = 3.57\sqrt{k} \cdot (\sqrt{h_t} + \sqrt{h_r})$$

where $h_t$ and $h_r$ are the heights in meters of the transmitting and receiving antennas above their common horizon.

In many cases, the heights may be approximated as heights above average terrain. The term $k$ is the effective earth radius factor that accounts for the fact that the distance to the radio horizon is normally longer than the distance to the geometric horizon because of atmospheric refraction. A typical value for $k$ in temperate climates is 1.33 ($k$ depends on the vertical gradient of refractivity near the earth's surface and normally varies from about 1.2 to 1.6, depending on weather conditions).

For a transmitting antenna at 30 meters above the terrain and a receiving antenna at 2 meters, the computed maximum propagation distance is 28 kilometers. It can be difficult to achieve such maximum distances in practice. Any obstructions along the propagation path will affect the signal's range. Signals might be blocked or reflected by buildings or other objects, diffracted over and around mountain peaks and ridges and the corners of structures, or even travel much longer distances than normal because of anomalous atmospheric ducting.

# 5. Types of frames

In this section, we will explain the different types of frames exchanged between all the different components (Base Station, Rover and satellite).

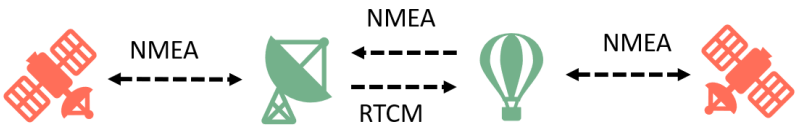



*Figure 7: types of frames exchanged*

## A. NMEA

[1] The National Marine & Electronics Association is a non-profit association founded by professionals from the electronics industry, manufacturers, distributors, sellers and professors. Their goal is to standardize marine equipment. They created many standards, notably the NMEA-0183 standard. It is defined as a data transmission protocol between GPS instruments and electronic equipment.

[10] With this standard, the data are transmitted in ASCII characters with the [CR] (carriage return <0x0D>), [LF] (line feed <0x0A>) and a baud rate of 4800. The data are encased in frames. Each frame starts with a '$' followed by 2 letters to identify the receiver, the talker id:

- GP for Global Positioning System
- BD ou GB for Beidou
- GA for Galileo
- GL for GLONASS
- GN for GPS+GLONASS
- etc.

This is followed by 3 letters to identify the frame:

- GGA for Global Positioning System Fix Data
- GLL for Geographic Position - Latitude/Longitude
- GSA for GPS DOP and Active Satellites
- GSV for Satellites in View
- VTG for Track made good and Ground speed
- RMC for Recommended Minimum Navigation Information
- etc.

Then, there are a variable number of fields with a comma separating them. The comma is a field separator to un-concatenate the data with a data analysis program. After the data, there is an optional field called checksum preceded by a * (Exclusive OR of all the characters encased between the $ and the *). Some frames require the checksum. Finally, there is the closing sequence composed of the [CR][RF]. A frame can possess a maximum total of 82 characters.

We can summarize the frame with the following expression:

```
$<talker ID><Trame type>[,<data>,<data>]*<checksum><0x0D><0x0A>
```

[2] When we recovered the data from the Base Station, we found $GPGGA frames like the following example:

```
$GNGGA,092535.00,4334.23269,N,00128.01143,E,2,12,1.93,237.8,M,48.5,M,1.0,0000*69
```

| Name | Data | Description |
|---|---|---|
| Sentence identifier | `$GNGGA` | Global Positioning System Fix Data |
| Time<br>*UTC of position* | `092535.00` | 09:25:35 Z |
| Latitude<br>*N or S* | `4334.23269`<br>`N` | 43° 34’ 23.26”N<br>43.57054483333 |
| Longitude<br>*E or W* | `00128.01143`<br>`E` | 1° 46’ 68.57”E<br>1.46685716666 |
| GPS Quality indicator:<br>● 0 = No fix<br>● 1 = GPS fix<br>● 2 = DGPS fix | `2` | Data is from a DGPS fix |
| Number of satellites in use (not in view) | `12` | 12 satellites are used |
| Horizontal Dilution of Precision (HDOP) | `1.93` | Relative accuracy of horizontal position |
| Altitude above mean-sea-level<br>*Meters* | `237.8`<br>`M` | 237.8 m above mean-sea-level |
| Geoidal separation<br>*Height of geoid above WGS84 ellipsoid* | `48.5,M` | 48.5 m |
| Time since last DGPS update<br>*Seconds* | `1.0` | 1 second since last update from differential reference station |
| DGPS reference station ID | blank | No station ID |
| Checksum | `0000*69` | Used by program to check for transmission errors |

*Figure 8: example of frame parsing*

## B. RTCM

[8] In 1985, the Radio Technical Commission for Maritime Services (RTCM) suggested a standard format for coding and transmitting such corrections. Although other proprietary formats have been devised, the RTCM format remains the most widely used. For example, member agencies of the International Association of Lighthouse Authorities, such as the U.S. and Canadian Coast Guards, employ the RTCM format for DGPS correction transmissions broadcast by selected marine radio navigation beacons.

Researchers had shown by the early 1990s that it was possible to improve on the accuracies afforded by DGPS by transmitting carrier-phase data to user receivers, permitting real-time, high-accuracy positioning even if the receivers were moving. They christened the technique real-time kinematic, or RTK.

In response, RTCM Special Committee 104 (SC-104), which devised the DGPS message format, added four new message types to Version 2.1 (published in January 1994) to handle RTK requirements. The 4 message types consist of 2 message pairs. Message Types 18 and 19 contain the raw carrier-phase and pseudorange measurements, respectively, made at the reference station. The measurements may be L1 or L2 and are accompanied by a high-precision time stamp. Message Types 20 and 21 contain corrections to the corresponding measurements based on the reference station's known position, the satellite's position, and the behavior of its clock as determined from the broadcast navigation message. The corrections are adjusted for reference receiver clock offset but not for ionospheric or tropospheric delay. In other words, the pseudorange correction of Message Type 21 is very similar to a Message Type 1 or 9 correction used for DGPS but has additional measurement quality information and can be used to support dual-frequency receivers.

To avoid large biases, the integer whole cycle carrier-phase ambiguity at the initial epoch is reduced to a small value in the Message Type 20 carrier-phase corrections. The Message Types 18 and 19 pair is intended for use in double-differencing algorithms, whereas the 20 and 21 pair is meant for processing undifferenced data. Although there are several advantages to its use, this latter pair of message types seems not to be used as much as the former.

RTCM SC-104 published Version 2.2 of its standard in January 1998. Message Type 22 was added to further support RTK. This message type provides additional corrections to the reference station antenna information provided in Message Types 3 and 32 used for pseudorange differential GPS and GLONASS operations respectively. Some GPS receiver manufacturers have also devised their own proprietary formats for RTK use as an alternative to the RTCM SC-104 standard.

# 6. Material

We were in possession of an ArduSimple starter kit LR composed of :

- 2 simpleRTK2B boards (Rover and Base Station)
- 2 Radio Modules LR (Long Range) with Xbee + 2 radio antennas
- 2 u-blox ANN-MB-00 Antenna for GNSS Dual Band with cable (IP67)
- Base and Rover preconfiguration

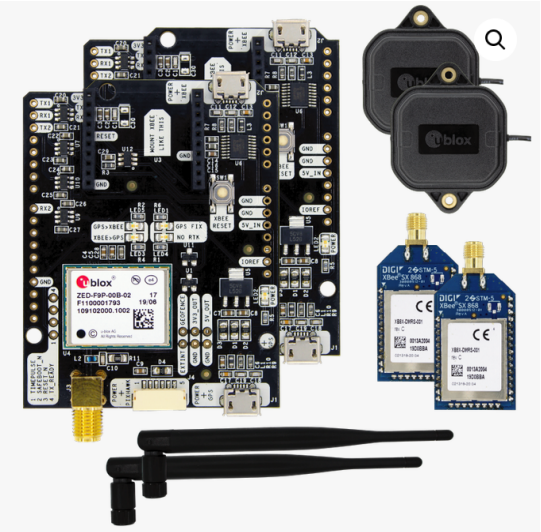


*Figure 9: simpleRTK2B – Starter Kit LR*

Mid-project, our tutor bought another simpleRTK2B board with a Xbee module so we could try testing new configurations: one base with two rovers or two bases with one Rover for example. We also used an ESP32 and the Arduino IDE. We used them to see the frames exchanged between the base and the Rover. We also coded a program to parse the frames and send it to a database. Moreover, we had two frame sniffers made by our tutor. We could attach the Xbee modules on them and, with the XCTU software, we could see the frames exchanged between the two modules. We used this system to test the range of the Xbee modules. Finally, we used u-center and u-center 2 which are software developed by uBlox to manage and display the simpleRTK2B data.

# 7. Our work

In this section, we will see the work we did during this project.

## A. Tests

To understand and demonstrate the full potential of our solution, we thought of a lot of ways to test it.

Testing the system - The first test we ever conducted was to turn on the RTK solution we bought and follow the quick plug and play guide provided by the official datasheet. While doing this we had our first encounter with the U-center software where all the data about the devices can be visualized. During this very first launch we noticed that the precision was far away from what we wanted to achieve and that we had no RTK correction on.

Sniffing the communication - Our first objective was to have a clear idea of the packet that each device was sending and receiving. In order to see that we started by using one of the Xbee modules as a sniffer while the base or the Rover were activated on their own. Basically we plugged one device at a time, while the other was off, and we launched our communication module meanwhile to listen to what was being transmitted and received. We did that thanks to the XCTU software that offers a graphical view of the transmitted data. This was the first step toward understanding the frame we were dealing with, unfortunately at this time we had only two communication modules and thus we couldn't listen while both the base and Rover were on. Because of that we were not able to identify the exchange that was going on.

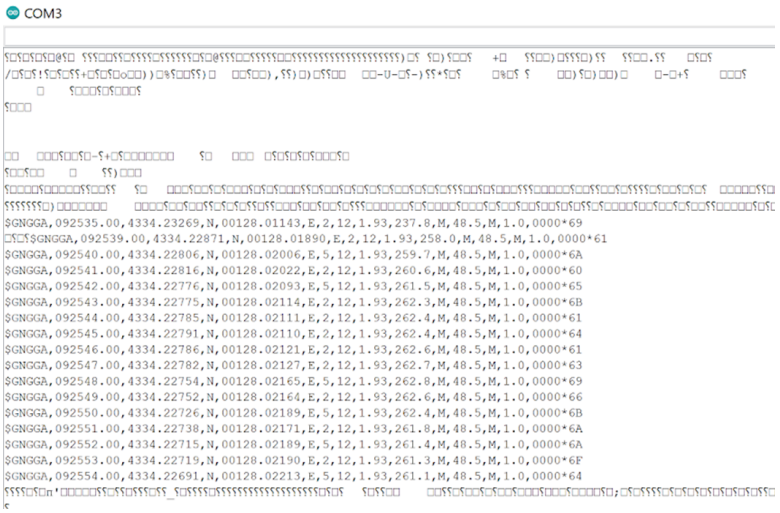



*Figure 10: frame interception with an ESP32*

Limit test of the range - With the same process, we also organized a way to test the range capacity of our module. This test was motivated by the need to know if we had to change the existing communication protocol for a LoRa protocol that is supposed to have a large range. According to the data and the information we received, the communication ships equipped on our devices are supposed to work up to 15 km. This is the max value obtained in a perfect situation which involves a clear view with no obstacle between the devices. To verify this we configured one of the Xbee modules to transmit basic messages every 1 second in broadcast, and the other module to a listening state, using the XCTU software once again. In practice, we went away from the city in the countryside. Once there, we had a team with the transmitter that stayed immobile while the other was moving away from it. While doing that we would check if we were still receiving the transmitted message. The result we got was far away from the theory but because of everything that can stand in the way when on the ground the signal can be seriously impacted, we have to keep in mind that the balloon will be up in the air and therefore nothing will impact the signal.

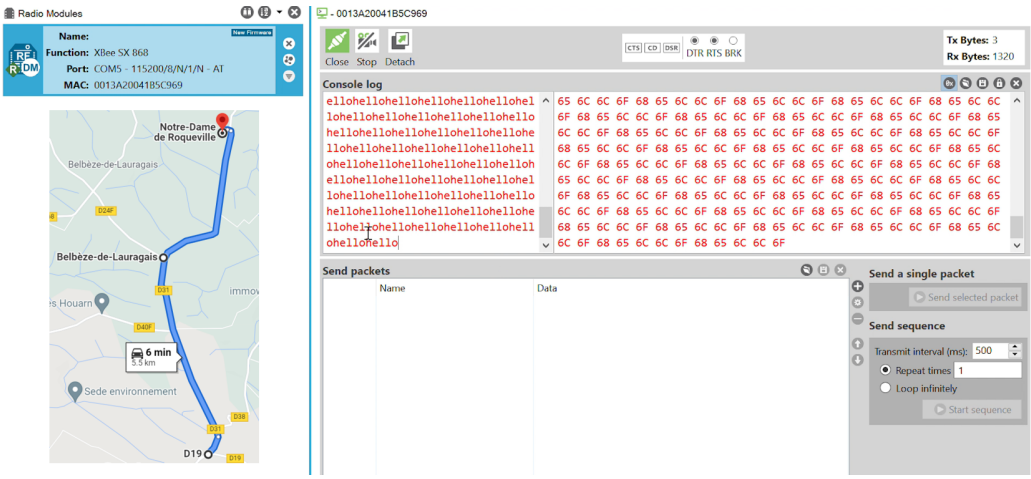



*Figure 11: Xbee range test with XCTU*

Getting the best possible accuracy - Once we had a better idea of what was truly going on and after a lot of different configuration ideas, we decided to test our RTK solution once again. Contrary to the first time, we planned it better with the knowledge we gathered from our research. The main difference was with the base, we decided to fix its position with coordinates entered manually, which means we found an open space outside, chose a place to get its latitude and longitude coordinates via satellite image, and that's the place where the base will be installed. This method is opposed to the "survey in" method in which the base calculates its position on itself, unfortunately when we tried this solution, even after hours of being on, the position was still not accurate enough. Knowing that once the base is set it must not move at all, we decided that choosing the place where it would be in advance was the best way to go, especially since we won't have so much time for the base to calibrate in the real situation of our project. We also chose the correct frames that were to be sent and received by each of the devices. This was our biggest breakthrough, with this configuration we achieved a 2 cm accuracy which is the best this technology can do.

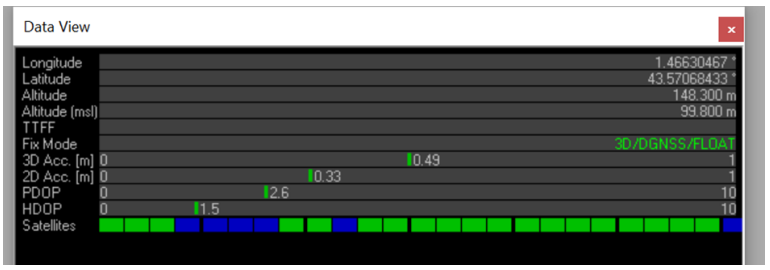



*Figure 12: RTK accuracy*

To conclude with the testing part, we finally set our system in its last version to observe that we were able to get the best possible accuracy.

## B. Our solution

Our solution is presented in the schematics below:

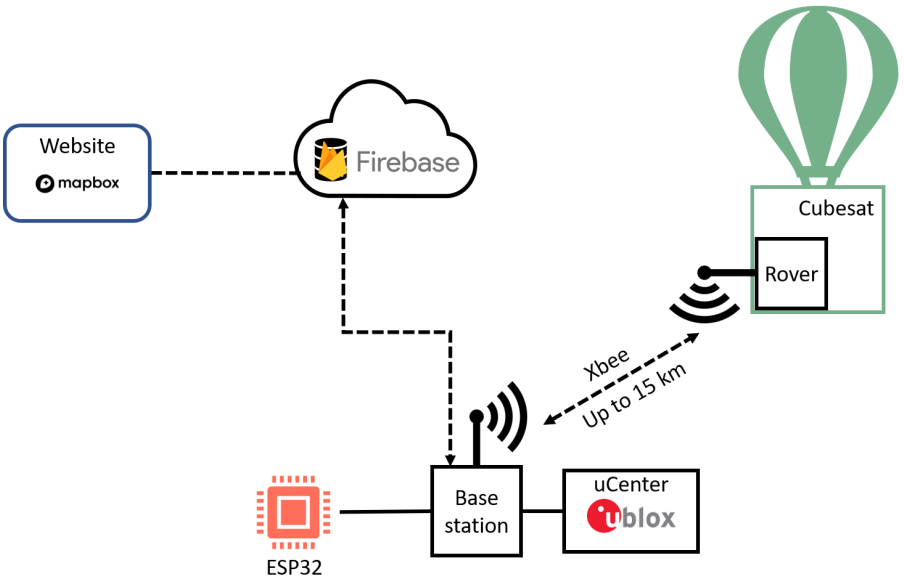



*Figure 13: architecture of our solution*

Retrieving the position frame - The last test we had to deal with was about retrieving the geopositioning data of the Rover to view it on a map on the ground, because so far the Rover position was only displayed on the computer directly linked to it, however as the Rover will be 30 km up in the sky it is not viable. Therefore we had to develop this map, but before we could do that we had to know the exact structure of the frame containing the data we wanted to retrieve so that we could parse it correctly. In order to do that we used an ESP32 that we linked to the receiving port of the base. Thanks to that we were able to see the frame being received by the base, identify the one corresponding to the Rover's position, and parse it so that we would only exploit the longitude/latitude information.

Parsing the frames

By plugging the ESP32 to the transmission port of UART of the base station, we can retrieve the corrected position in raw NMEA frames .With an arduino code, we then parse them to extract only the latitude and longitude. We then send this information via Wifi to a database every second (that can be changed).

```
COM3

$GNGGA,092535.00,4334.23269,N,00128.01143,E,2,12,1.93,237.8,M,48.5,M,1.0,0000*69
Latitude : 43.570545
Longitude : 1.466857
Altitude : 237.800003
```

*Figure 14: frame parsing with an ESP32*

## Database

We saved the location of the Rover in the Firebase Realtime database so that we can view this information remotely. The Firebase real-time database is a NoSQL cloud-hosted database that uses data synchronization to automatically receive new information in real time from each connected client.

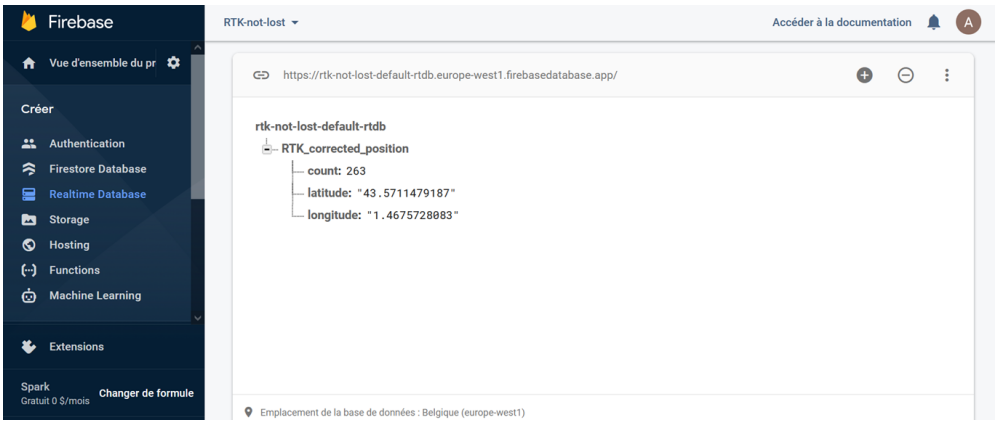



*Figure 15: our Firebase database*

## Website

We created a web application accessible with this QR code. To do so, we wrote a HTML and JS Code where we connect to the firebase to access the database, and to an interface called mapbox that uses the data stored in the database to draw the position of the rover on a map. Since the base station is stationary we can enter it on the map interface to display it on the same map. You can see the website on the following link: https://rtk-not-lost.web.app/.

*Figure 16: QR code access to our website*

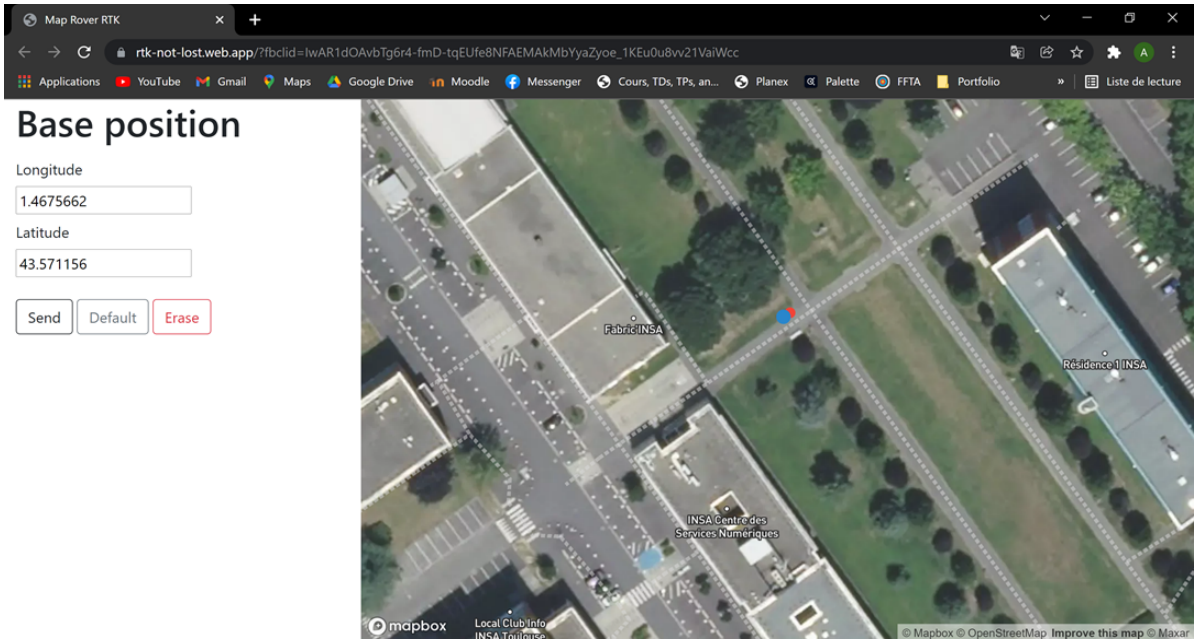



*Figure 17: our website*

To conclude, we were able to display the position of the Rover and the Base Station on the same map that will be accessible with no physical connection to the Rover.

Comparison

For you to have a clearer idea of the difference there is between GPS, DGPS and RTK, we conducted a last test that you can see on this slide. Those three pictures have been taken at the exact same place, the center of this intersection, the differences speak for itself.

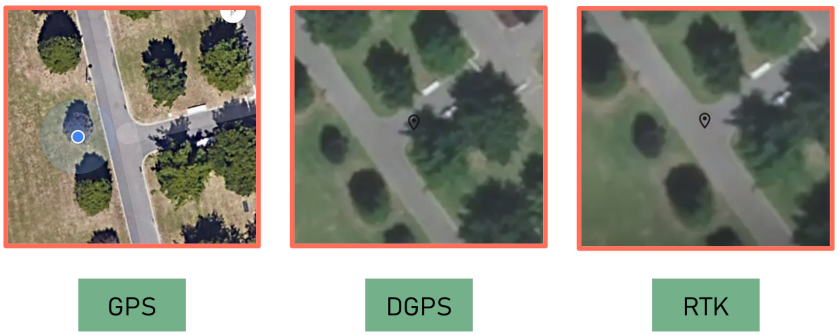



*Figure 18: comparison of accuracies*

# 8. Management method

Now that we explained all the technical details of the project, we will present how we managed the project. We used the agile method, which is the most appropriate management method for our project and the easiest to use. It consists of 2-weeks sprints, more or less, during which we will work to achieve one objective. At the end of this sprint, the team has to meet up to talk about the advancement and the problems encountered. Here is the calendar of the different sprints we led in this project.

| | | |
|---|---|---|
| **1st meeting**<br>11/10/21 | • Explanations about the project<br>• Goals setting | |
| **Sprint #1**<br>11/10/21 - 25/10/21 | • Research about the technologies (GPS, DGPS, RTK…)<br>• Testing the RTK kit | **2nd meeting**<br>28/10/21 |
| **Sprint #2**<br>25/10/21 - 08/11/21 | • Research about the RTK architecture<br>• Research bout the types of frames<br>• Preparing the first project review | **3rd meeting**<br>04/11/21 |
| **Sprint #3**<br>08/11/21 - 22/11/21 | • Writing the report's theorical sections<br>• Studying the different existing configurations for the base/rover<br>• Sniffing the communication | **4th meeting**<br>23/11/21 |
| **Sprint #4**<br>22/11/21 - 06/12/21 | • Meeting with the Paul Sabatier team<br>• Testing the Xbee modules' range<br>• Writing the report | **5th meeting**<br>02/12/21 |
| **Sprint #5**<br>06/12/21 - 20/12/21 | • Preparing the second project review<br>• Attending the Paul Sabatier team's presentation<br>• Trying to get the best possible accuracy<br>• Retrieving the position frames with an ESP32<br>• Writing the report | **6th meeting**<br>06/01/22 |
| **Sprint #6**<br>03/01/22 - 17/01/21 | • Parsing the frames via an ESP32<br>• Collecting them in a database<br>• Displaying the position on a website<br>• Writing the report<br>• Preparing the last project review | |
| **Final meeting**<br>17/01/21 | • Demonstration<br>• Feedbacks | |

*Figure 19: Agile method sprint time table*

In the following table, you can see the workload repartition between all the members of the team:

| | Research | | | Tests | | | Code | | | | | Report / Deliverable | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Names | GPS | DGPS | RTK | Xbee range | Frame sniffing | Frame analysis | Read raw data | Frame parsing | Receive the rover's position | Database | Display on a map | Context | GPS vs DGPS vs RTK | RTK architecture | Types of frames | Material | Our work | Management method | Security measures (Proverif) | Challenges and lessons | Slides | Demo video |
| Yacine | | | | X | X | X | X | X | X | X | X | | | X | | X | X | X | X | | | X |
| Hugo | | | | X | X | X | X | X | X | X | X | | | | | X | X | X | X | | | X |
| Agathe | X | X | X | X | | | X | X | X | | | X | X | X | X | X | X | X | | | X | X |
| Leila | | X | X | X | | | | | | | | | X | | | | | X | | X | | |
| Assia | | | | X | X | X | X | X | X | X | X | X | X | | X | X | X | X | | X | | X |

*Figure 20: workload repartition*

# 9. Security measures

Our project consists in setting up a RTK localization solution, the goal is to recover the payload of a weather balloon. This balloon will be sent to 32km of altitude with measuring devices to make readings. The particularity of RTK (Real Time Kinetic) is that, unlike a traditional GPS, it offers an accuracy to the nearest cm. This is made possible by a correction of the position between a Base Station (immobile) on the ground and a Rover (mobile) which is part of the balloon payload.

We have two types of telecommunication, the first between the base and the Rover in LoRa and the second between the base/Rover and the satellites. The base and the Rover calculate their positions independently of each other, then the Rover sends its position to the base, which will correct it and send it back to the Rover.

In our case, we could imagine that an attacker would want to recover the payload before us to exploit the data. To do this he would not have physical access because the Rover is located in a balloon several kilometers from the ground, and the base is permanently with the team or in a secure place. The attacker will thus have access to the medium only by listening to the communications.

Our project does not include any measures for communication security because it is not really interesting, as it is unlikely that an attacker would want to steal weather data for research purposes. So all the location data is in the clear and can be accessed by anyone listening on the right frequency.

Thus a malicious person would only have to launch a "man in the middle" attack in order to retrieve the position of the Rover, redirect it to his own pirate base to obtain the correction and send a false position to the legitimate base in parallel in this way the team would be sent to the wrong place and he could retrieve the package without difficulty.

To remedy this, we can imagine setting up an encryption of the radio link using a public and private encryption key. We could for example use the ElGamal algorithm. It allows an encryption between the base and the Rover directly on a public channel which does not distort the channel and ensures the integrity and confidentiality of data. The keys are generated asynchronously. Without going into details here is how this algorithm works:

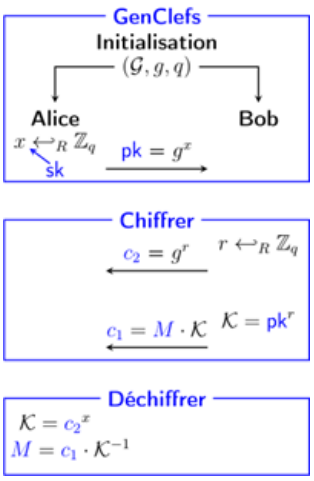



*Figure 21: pseudo-code ElGamal algorithm*

Here are the different steps:

- We generate a very large prime integer $A$
- We select an integer x included in $[1; A - 1]$
- We select $G$, a generating root of $A$
- We calculate the public key : $y = G^x modulo(A)$ and the private one: $x$
- To encrypt : we select R an integer included in $[1; A - 1]$
- For $c1 = (G^T) modulo(A)$
- For $c2 = (M\, x\, y^R) modulo(A)$
- To decrypt : $M = (c1(c2^x)^{-1}) modulo(A)$

We will now study the current case and the case with countermeasures on ProVerif. You can find the code in the appendix. With it, we instaure an asynchronous ElGamal encryption with a public and private key for the server and the client. Then, we want to prove that, if an attacker listens to this connection, he would not be able to recover the private keys and so, he would not be able to decipher the message. The following test results prove it:

```
-- Query not attacker(privatekeyy[]) in process 0
Translating the process into Horn clauses...
Completing...
Starting query not attacker(privatekeyy[])
RESULT not attacker(privatekeyy[]) is true.

--------------------------------------------------------------
Verification summary:

Query not attacker(privatekeyy[]) is true.

--------------------------------------------------------------
```

*Figure 22: ProVerif results*

# 10. Challenges and lessons learned

The first challenge we encountered was to assimilate the new notions. We had to do a lot of research to understand the RTK technology and architecture. These notions are truly complex and took us a lot of time to comprehend. Moreover, we used some new technologies to build a database and display a map on our website. After all, we still managed to do everything we wanted and learned a lot of new skills.

Even though the kit in our possession was supposed to be “plug & play”, we had a hard time finding the right configuration to obtain an RTK positioning. First, we had difficulties reading the raw frames obtained with the XCTU software. So, we created a program using an ESP32 to recover the frames and parse them. Then, to find the root of our problem, we linked the base and the rover’s board together. In doing so, we realized that it was the Xbee module that was not properly configured. So then, we configured the Xbee module the right way, with the same baud rate. This solution was found thanks to research and strategic thinking. It truly was a team effort.

One of the main challenges was to find the right time and place to do the experiments. As a matter of fact, to have a RTK positioning, you need an open-sky view. If we do not have that, it is impossible to have an accuracy of a few centimeters. Moreover, since we are in winter, it was difficult to test outside because our computers would not hold long enough to really test our solution.

# 11. Conclusion

Even though we achieved what was required, we can still think of ways to improve our system. Implementing the roaming solution which would allow the use of more than one base and cover more ground. We also could imagine saving the position point instead of just writing over it in the database to have the full course of the rover, and use this path for other purposes. Finally the design of our website could be more attractive, and the database more efficient with a better solution, to reach real-time for example.

To conclude, we learned a lot during this project. We encountered obstacles and challenges but we overcame them all at the end. It was frustrating at the beginning but very satisfying when we found the answers. We developed a lot of skills with this project, technical but also management wise. We discovered new technologies and we are very happy about how our project turned out in the end. We really thank our tutor Mr Auriol for proposing this subject and working with us, and Mr Scanlan for helping us with the report and the presentation.

# 12. Appendix

## A. Demo video

You can find the video on the following link: https://www.youtube.com/watch?v=SnIFUUjsiBk

## B. Code

You can find our code on our GitHub repository at the following link:
https://github.com/ALievre/RTK_project

## C. Security code

Study of the counter-measure on ProVerif:

```
type publickey.  -> public key

type privatekey.  -> privete key

free publickeyy: publickey.  -> we specify what everybody knows

free privatekeyy: privatekey [private]. -> we specify what is private

free c : channel.  -> clear channel


fun aenc1(bitstring,publickey) : bitstring. ->asynchronous encryption of c1

fun aenc2(bitstring,publickey) : bitstring. ->asynchronous encryption of c1

reduc forall message1:bitstring,pk1:publickey,z1:privatekey;
dec1(aenc1(message1,pk1),z1) = message1. -> decryption variable declaration
of transmission 1

reduc forall message2:bitstring,pk2:publickey,z2:privatekey;
dec2(aenc2(message2,pk2),z2) = message2. -> decryption variable declaration
of transmission 2


query attacker(privatekeyy). -> we define what the attacker wants to know


process

new c1:bitstring; -> variable creation (messages to transmit = GPS
coordinates)

new c2:bitstring; -> variable creation (messages to transmit = GPS
coordinates)

out (c,aenc1(c1,publickeyy)); -> we send both encrypted messages on channel
c

out (c,aenc2(c2,publickeyy)); -> we send both encrypted messages on channel
c

in (c,z:privatekey) [precise]; -> we create a variable z that is the
private key of the user who receives the message

out (c,dec1(aenc1(c1,publickeyy),z)); -> we decrypte the received message

out (c,dec2(aenc2(c2,publickeyy),z)) -> we decrypte the received message
```

# 13. References

In alphabetical order:


[1] C. Aoun, *Les Trames NMEA*. Gien: LP du Giennois, pp. 1-2. Available: http://www.cedricaoun.net/eie/trames%20NMEA183.pdf

[2] G. Baddeley, "GPS - NMEA sentence information", *Aprs.gids.nl*, 2001. [Online]. Available: http://aprs.gids.nl/nmea/. [Accessed: Jan- 2022].

[3] "Differential GNSS - Navipedia", *Gssc.esa.int*, 2011. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Differential_GNSS. [Accessed: Oct- 2021].

[4] S. Durand, *Amélioration de la précision de la localisation différentielle temps réel par mesure de phase des systèmes GNSS : étude détaillée des équations d'observation et du problème de résolution des ambiguïtés entières*. Paris: Observatoire de Paris, 2003, pp. 5-20. Available: http://recherche.ign.fr/labos/lareg/pdf/Theses/these_stephane_durand_2003.pdf

[5] T. Herring, *Principles of the Global Positioning System Lecture 08*. Cambridge, Massachusetts: MIT, 2012, pp. 3-14. Available: https://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-540-principles-of-the-global-positioning-system-spring-2012/lecture-notes/MIT12_540S12_lec8.pdf

[6] "How Does DGPS (Differential GPS) Work?", *RACELOGIC Support Centre*, 2018. [Online]. Available: https://en.racelogic.support/VBOX_Automotive/01General_Information/Knowledge_Base/How_Does_DGPS_(Differential_GPS)_Work%3F. [Accessed: Dec- 2021].

[7] R. J, "L1, L2, and L5 GPS Signals: What Do They Mean? – Equator", *Equator*, 2021. [Online]. Available: https://equatorstudios.com/l1-l2-and-l5-gps-signals-what-do-they-mean/. [Accessed: Jan- 2022].

[8] R. Langley, "RTK GPS", *GPS World*, pp. 1-5, 1998. Available: http://www2.unb.ca/gge/Resources/gpsworld.september98.pdf

[9] C. McGillivray, "How Does GPS Work? - RACELOGIC Support Centre", *En.racelogic.support*, 2020. [Online]. Available: https://en.racelogic.support/VBOX_Automotive/01General_Information/Knowledge_Base/How_Does_GPS_Work%3F. [Accessed: Dec- 2021].

[10] E. Raymond, "NMEA Revealed", *https://gpsd.gitlab.io/*, 2022. [Online]. Available: https://gpsd.gitlab.io/gpsd/NMEA.html. [Accessed: Jan- 2022].

[11] "Real Time Kinematics - Navipedia", *Gssc.esa.int*, 2011. [Online]. Available: https://gssc.esa.int/navipedia/index.php/Real_Time_Kinematics. [Accessed: Oct- 2021].

[12] J. Van Sickle, "The P and C/A Codes | GEOG 862: GPS and GNSS for Geospatial Professionals", *E-education.psu.edu*, 2020. [Online]. Available: https://www.e-education.psu.edu/geog862/node/1741. [Accessed: Jan- 2022].