Reykjavík University

Mechatronics I

T-411-MECH

Laboratory 7

27/09/2020

Students:
Giacomo Menchi
Matteo Guerrini
William Paciaroni

Teacher:
Joseph T. Foley

Contents

1 Main Tasks

This below is the GitHub link with all the code inside.

Link: GitHub Repo

1.1 Main Task 1

Task: Get the GPS module to give you some data.

Screenshot: This task was automatically completed in the next one (and in all the others, to tell the truth), so check the main task 2 for proof.

1.2 Main Task 2

Task: Find your current location.

Screenshot: To achieve this, we got latitude and longitude from the GPS module, parsed the two values using the nmea crate (which converts data from Adafruit Ultimate GPS format to a more commonly used one) and then inserted the results in Google Maps to check that the position was correct.

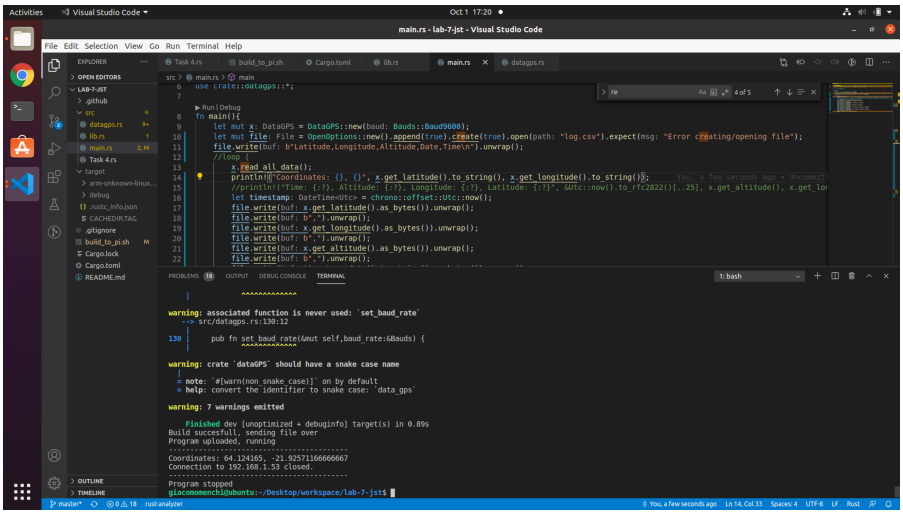

Figure 1: GPS coordinates obtained from the GPS module.

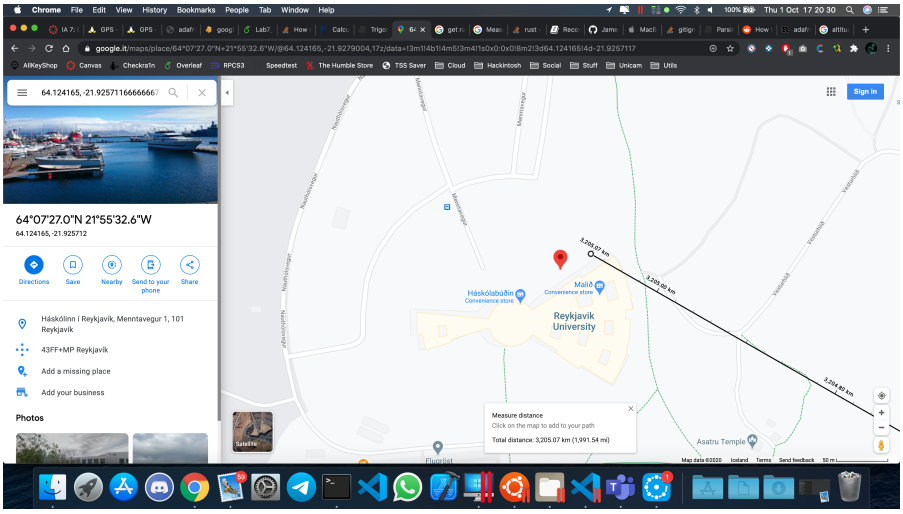

Figure 2: GPS coordinates when inserted in Google Maps.

1.3 Main Task 3

Task: Write out on the terminal where you currently are using some programming language with a fixed timer.

Screenshot: The screenshot shows how we wrote the current coordinates on the Rust console every 2 seconds.

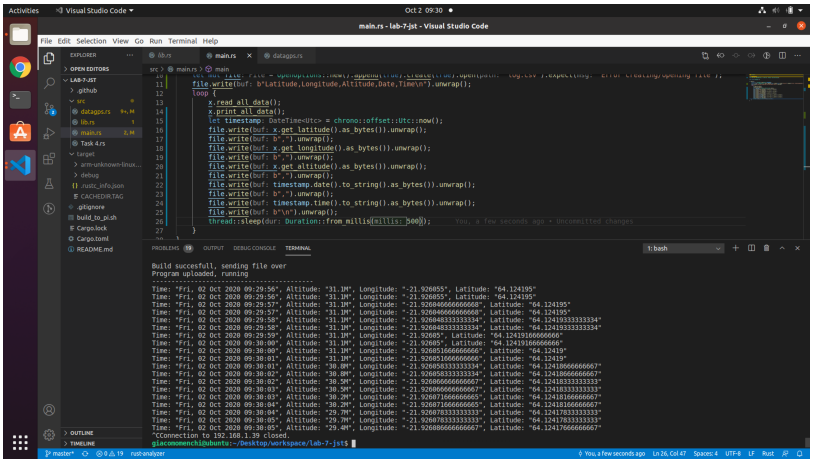

Figure 3: Coordinates with a two seconds delay.

1.4 Main Task 4

Task: Create a program that you can enter the desired coordinates and it tells you how close you are to it and which direction you need to go (north, south, west, east).

Screenshots: The screenshots below represent the distance from the GPS module location (Reykjavik University, Reykjavik, Iceland) and some coordinates (Polo Lodovici, Camerino, Italy).

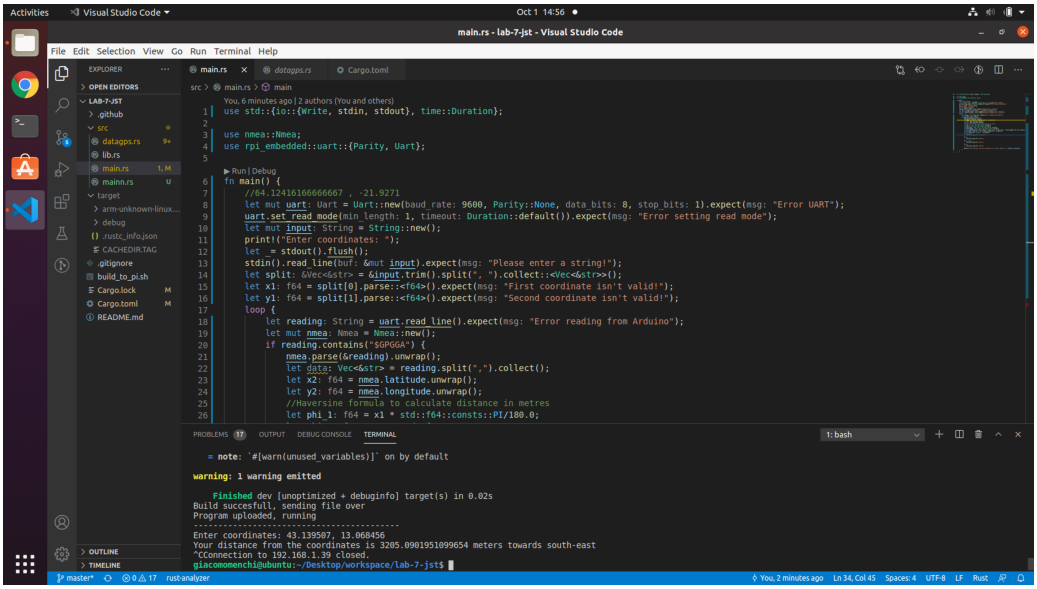

Figure 4: Coordinates distance and directions using GPS module.

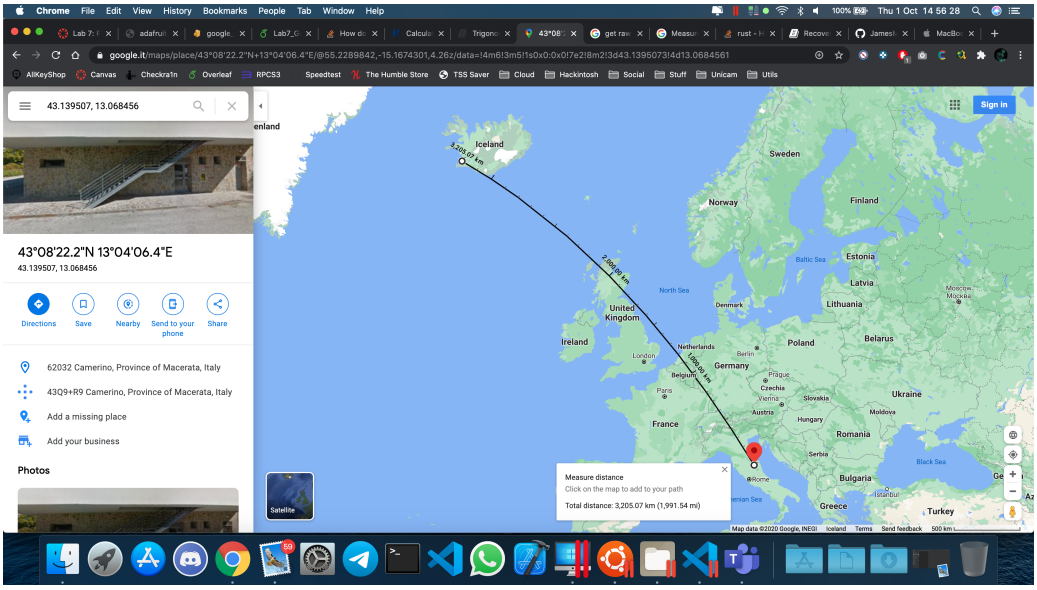

Figure 5: Coordinates distance using Google Maps (proof of correctness).

1.5 Main Task 5

Task: Find the beacon and record yourself there as proof. In the case of Covid lab shutdown choose a specific point yourself and get to it. You can use your phone as power supply for your RasPi.

NOTE +- 5m is acceptable

Pictures: As visible in the pictures, we reached the designed location, obtained the data from the GPS module, and then compared it to the actual coordinates we had to reach using Google Maps (we could have also used the code from the task before, but we didn't think about it in that moment since we were freezing), obtaining a 3.7 meters difference (which is below the acceptance threshold).

Figure 6: Photo proof containing the three of us us in the spot.

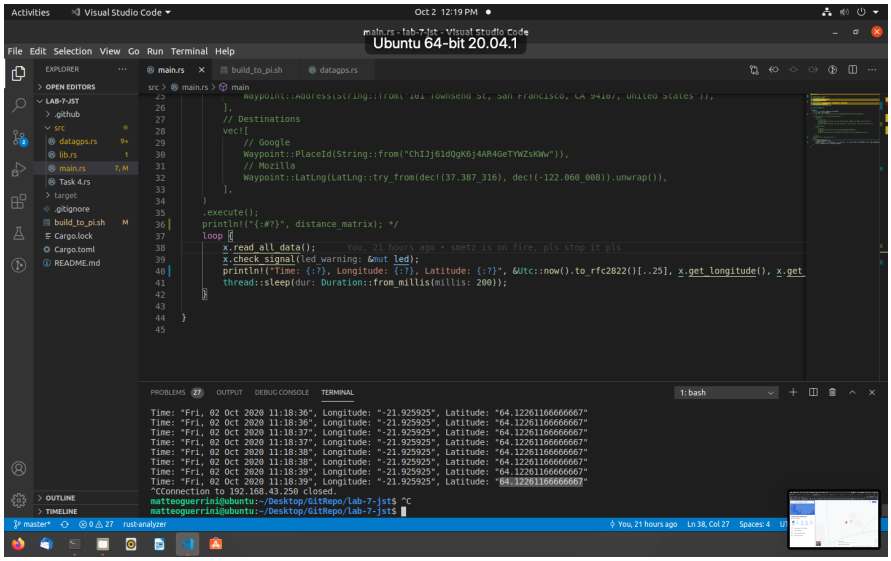

Figure 7: Screenshot of the data got from the GPS module.

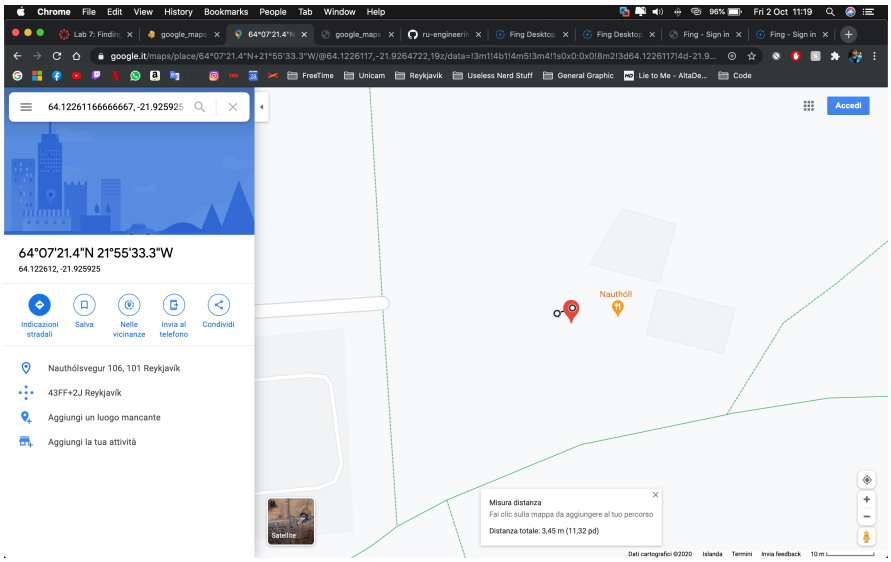

Figure 8: Coordinates distance using Google Maps (proof of correctness).

2 Hard Tasks

2.1 Hard Task 1

Task: If the signal is not strong enough, make the circuit blink a Red LED to indicate a weak signal.

Video: The link below shows the circuit in action. It takes a bit of time for the GPS module to lose signal, but as soon as it does, the red led starts blinking. The method we used to check if the signal is still present or not is based on two data fields:

- **Number of satellites**, which are the satellites that can be reached, has to be >=3 (three is the minimum number usable to triangulate the position, as the "triangulate" term itself suggests).
- **Horizontal Dilution of Precision (HDOP)**, which is used to evaluate precision of measurement, has to be <=5 (above five the precision starts to get worse, as explained in the table below, and we wanted to only have good measurements).

Link: Video Link

Meaning of DOP Values[*citation needed*] [edit]

DOP Value	Rating	Description
1	Ideal	Highest possible confidence level to be used for applications demanding the highest possible precision at all times.
1-2	Excellent	At this confidence level, positional measurements are considered accurate enough to meet all but the most sensitive applications.
2-5	Good	Represents a level that marks the minimum appropriate for making accurate decisions. Positional measurements could be used to make reliable in-route navigation suggestions to the user.
5-10	Moderate	Positional measurements could be used for calculations, but the fix quality could still be improved. A more open view of the sky is recommended.
10-20	Fair	Represents a low confidence level. Positional measurements should be discarded or used only to indicate a very rough estimate of the current location.
>20	Poor	At this level, measurements are inaccurate by as much as 300 meters with a 6-meter accurate device (50 DOP × 6 meters) and should be discarded.

Figure 9: HDOP scale explained.

2.2 Hard Task 2

Task: Write out onto a file using the CSV format you current location, height over sea level and current time.

log.csv: Inside the GitHub repo is the log.csv file with the data requested, also accessible at this link below.

Link: File Link

2.3 Hard Task 3

Task: Map out your trip. Make a circle. You do not have to use google maps but we suggest using the crate.

Picture: The picture shows the trip we mapped using the GPS module. By logging the coordinates visited to the log.csv file from the previous task and uploading it to the website below, the entire path followed was printed directly to the satellite map.

Link: Map Website

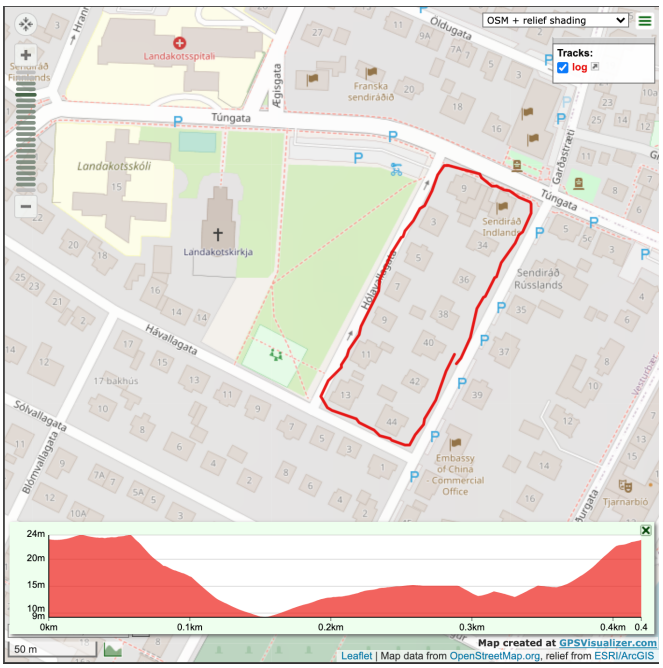

Figure 10: Trip map.

3 Advanced Tasks

3.1 Advanced Task 1

Task: Add the height to the map for the trip.

Picture: From the previous task, we just had to change a few of the website settings to also include altitude in the trip map. By setting "Add DEM elevation data" to "best available source" in the homepage of the website before uploading and then clicking "Draw an elevation profile" after the map has been plotted, the result is the image above. As visible from the graph on the bottom, we started from a discretely high point, descended to 9 meters, then got back up to the starting position again, still at the same height.

3.2 Advanced Task 2

Task: Figure out a physical feedback system to know if you are going the right direction

Video and explanation: The video below shows the completed task, where we used a led to signal if the direction we were going to was right or wrong. The underlying implementation uses two types of distances:

- **Current distance from destination**, obtained by subtracting current coordinates from destination coordinates and calculating the absolute value of the result.
- **Previous distance from destination**, obtained by subtracting previous coordinates from destination coordinates and calculating the absolute value of the result.

By checking if the current distance from destination is less than the previous one, the program can understand if we are going in the right direction or not, and thus turn on or off the led.

Link: Video Link