RAT

Remote Autonomous Transporter

(formerly CAT, "Coche Autónomo Teledirigido")

A remotely controlled robot that travels to a specified location while avoiding obstacles.

Daniel Azemar	1390451
Jialuo Chen	1390203
Sergi Pous	1390450
Adrià Rico	1391488

PROJECT SPRINT #2.
DATE: 03 May 2017

Table of Contents

RAT

A remotely controlled robot that travels to a specified location while avoiding obstacles.

Note: amendments from sprint 4 are written in blue.

Project description

The sole purpose of RAT is to transport an object from one point to another. To accomplish this, RAT will wait until it remotely receives a coordinate to go to, then it'll travel there while avoiding any obstacle in the way. It'll also periodically send its current position to the remote controller, so that the robot can be localized at any time. Alternatively, RAT will also be able to be manually controlled.

Below is a detailed description on how we plan to carry out these functionalities:

- Movement:
  RAT will be able to move using two DC motors (with wheels attached) controlled by a motor driver board. This setup makes it possible for RAT to move forwards and backwards with an adjustable speed. Rotation will be accomplished by activating only one of the two motors.
- Remotely sending and receiving data:
  We'll use two transceivers controlled by two Arduino UNO boards.
  One of them will be located in the robot, and the other will be connected to the computer used to send the coordinates and receive the current robot position.
- Route tracing:
  When RAT receives an objective coordinate, it'll get its current position and orientation using a GPS and a magnetometer respectively. It'll then trace a

straight line from its current position to the objective coordinate, calculate the direction, and rotate accordingly. When it determines it's facing the correct direction, it'll start moving forward until an obstacle is found or the objective coordinates have been reached. Periodical checks will be made to make sure it's not going offtrack.

- Obstacle avoiding:
  RAT will be able to detect nearby obstacles using a sonar. Once an obstacle is detected, it'll try to overcome it and then resume its original path. We still don't know a lot about how to code this, but we've been promised we're going to be taught.
- Manual remote control:
  RAT will be able to be remotely controlled via a Telegram bot with a simple button interface. The computer running the bot must also be connected via USB to the Arduino used to send signals to RAT. Alternatively, it'll also be possible to manually control RAT directly from the computer.

Note on the limitations of the distance traveled:

The precision of the GPS used is around 3m, so RAT's duty is to land inside an area of 3m around the specified point. Specifying points too close to RAT would make it The transceiver used has a range of about 100m, so making RAT go towards points farther than that leads to losing all kind of communication with it. It could still, in principle, autonomously get to its objective, and then receive new orders from a different computer nearby.

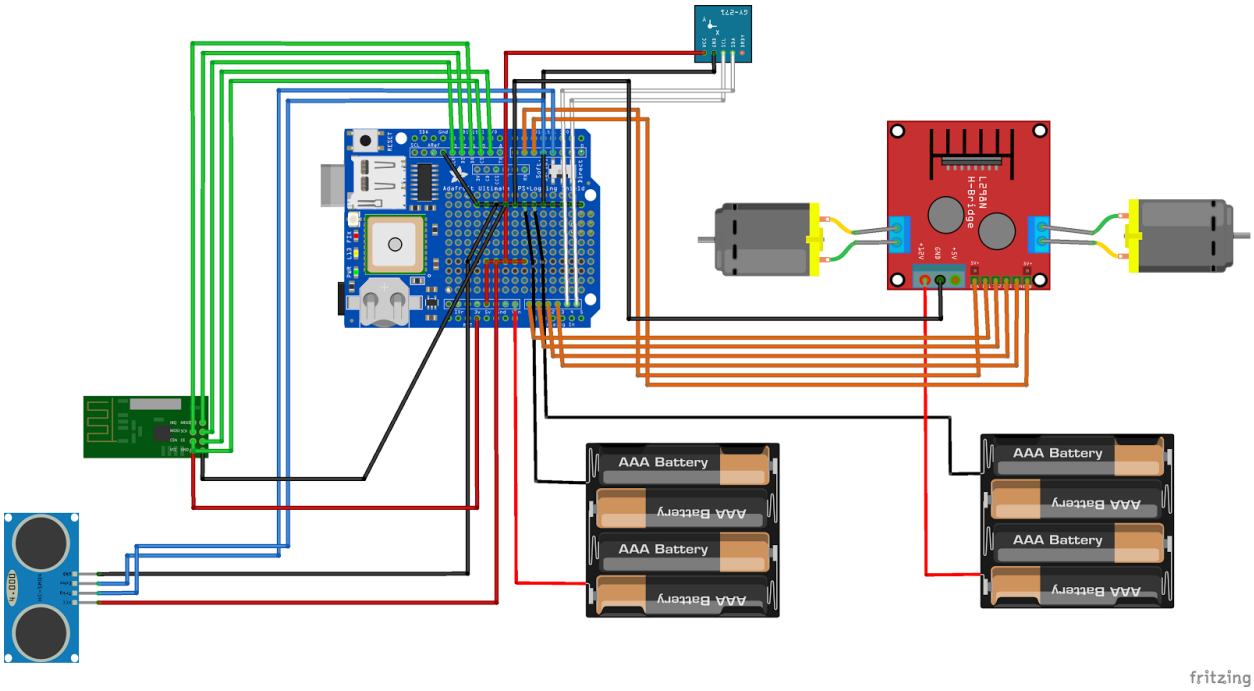
Electronic components

- Arduino UNO (2x) (one of them is of our own, it'll be used to send data from the computer to the robot using the transceiver)
- nRF24L01P+ (Transceiver) (2x)
- L293D (Motor Driver Board)
- HC-SR04 (Sonar)
- Adafruit Ultimate GPS Logger Shield

- DC motor (2x)
- GY-271 (Magnetometer)

Scheme

RAT



Notes:

- All red wires go to Vcc (5V for all the components except the transceiver which goes to the 3.3V pin).

- All black wires go to GND.

- All the AAA batteries shown in the scheme are 1.2V AA batteries in actuality.

- The GPS shield uses digital pins 0, 1 (for the direct connection), 7 and 8 (for the soft serial connection).

Connections:

Sonar (blue)

Trigger Pin -> Digital Pin 3 (PWM)

Echo Pin -> Digital Pin 4

Motor Driver (orange)

ENA -> Digital Pin 5 (PWM)

ENB -> Digital Pin 6 (PWM)

IN1 -> Analog Pin A0 (Digital Pin 15)

IN2 -> Analog Pin A1 (Digital Pin 15)

IN3 -> Analog Pin A2 (Digital Pin 16)

IN4 -> Analog Pin A3 (Digital Pin 17)

Transceiver (slave, green)

CE Pin -> Digital Pin 9 (PWM)

CSN Pin -> Digital Pin 10 (PWM)

SCK Pin -> Digital Pin 13

MOSI Pin -> Digital Pin 11 (PWM)

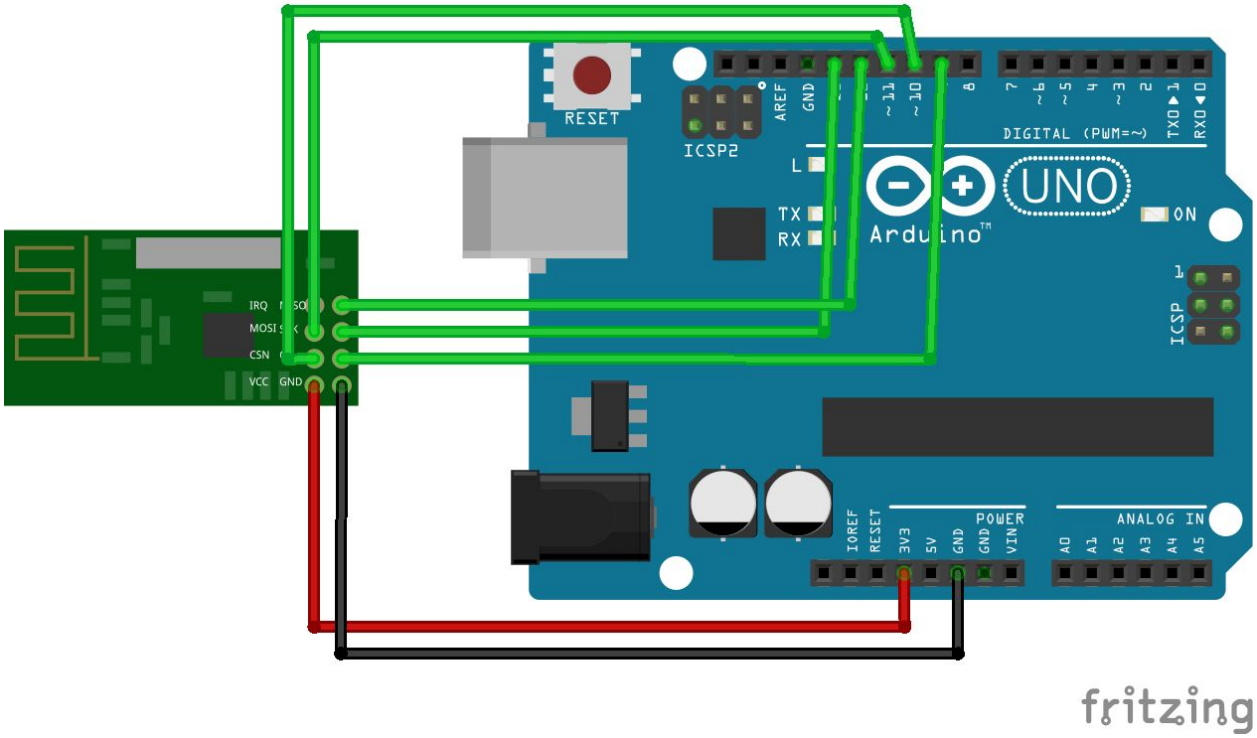
MISO Pin -> Digital Pin 12

Magnetometer (white)

SCL Pin -> Analog Pin A5 (I2C Clock)

SDA Pin -> Analog Pin A4 (I2C Data)

Master (computer)



Connections:

USB alimentation

Transceiver (master, green)

CE Pin -> Digital Pin 9 (PWM)

CSN Pin -> Digital Pin 10 (PWM)

SCK Pin -> Digital Pin 13

MOSI Pin -> Digital Pin 11 (PWM)

MISO Pin -> Digital Pin 12

Extra components and 3D pieces

- Wheel for DC motor (2x)
  They'll be connected to the motors to allow RAT to move on ground.
- Caster wheel
  A support wheel which will be placed on the back of the robot.
- 4 AA Battery Holder (2x)
  Used to conveniently hold 4 AA batteries in series so that they produce 5V of voltage.
- 1.2V AA Rechargeable Battery (8x)
  4 of them them are to power the Arduino board, the other 4 to power the motors.

List of 3D pieces:

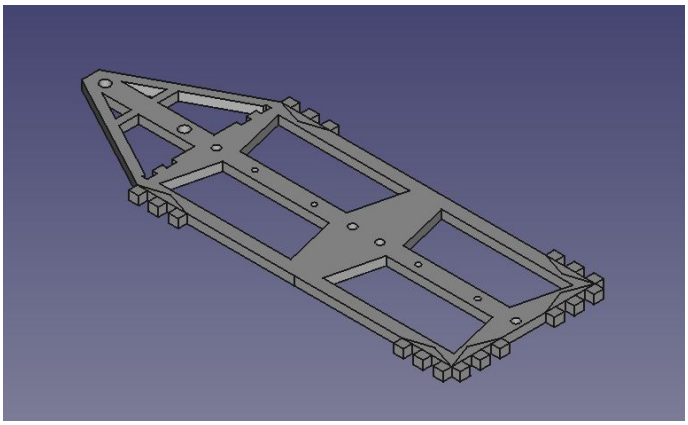
V1 problems and announcement of V2:

After having seen the errors that V1 had, we are encouraged by a new design and a second print, this time with the aim to be more consistent.

The main purpose of this reprint isn't to fix the errors that we've detected, but to generate fewer pieces (for example, the box holding the batteries is now only one piece, not 5 like before, one per wall).

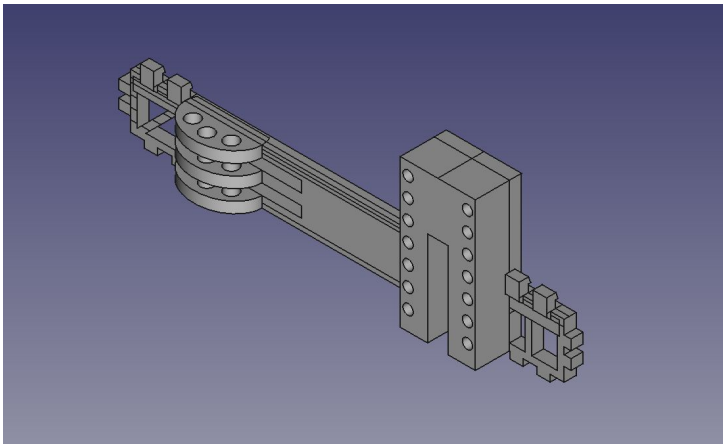
We think that'll fix many problems when joining the pieces (we've struggled a lot with V1) and it also makes the modelling is easier.

Below, we'll show the errors in V1 within the table of pieces of the last sprint:

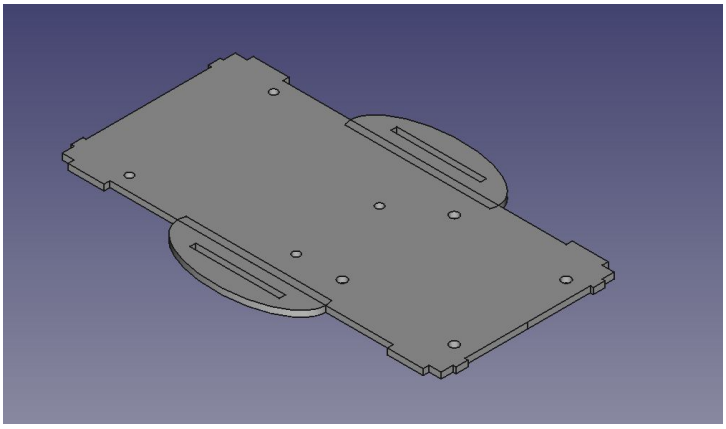
No.	Description	Image
1	**Base** of the robot. Here we'll attach both the Battery Holder and the Caster Wheel.	
	Errors detected: 1. We put one hole of the caster wheel in the wrong place. We had to drill it. 2. We realized that maybe we needed 10 batteries instead of 8 to supply enough power to the Arduino and all sensors. We had to expand the base.	

<table>
<tr><td rowspan="2">2</td><td>Laterals of the robot.

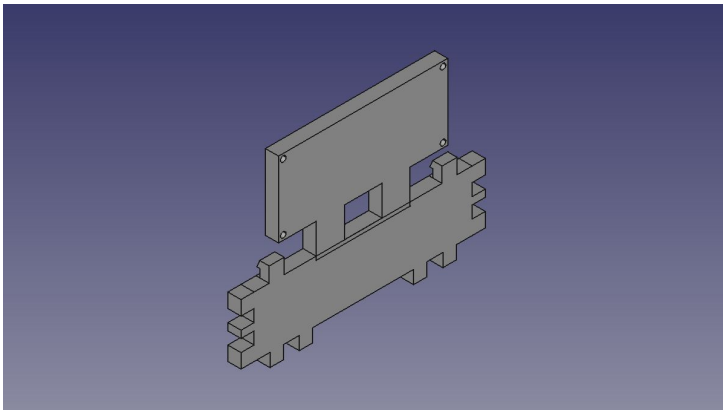
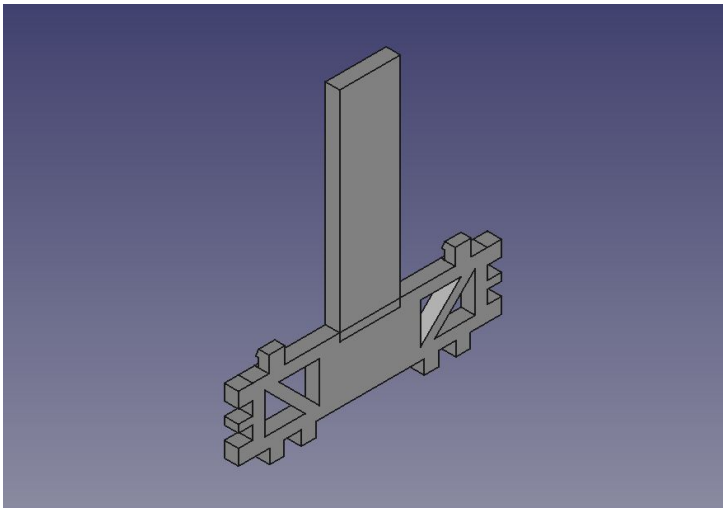
Here we'll attach the motors.

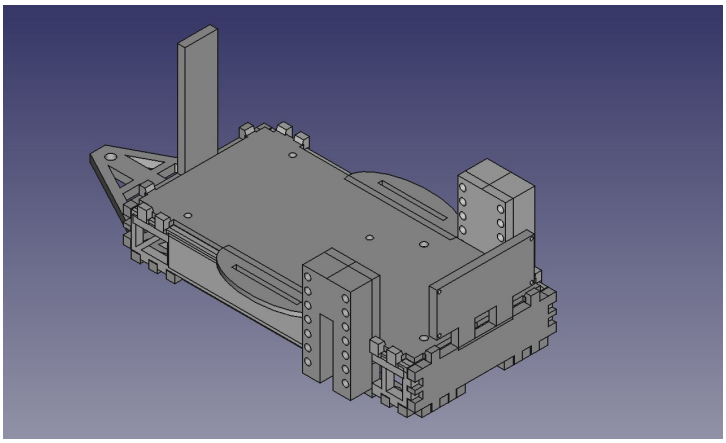
...and a mystery piece, coming up next sprint!</td><td>  </td></tr>
<tr><td colspan="2">Errors detected:

1. The batteries didn’t have the holes to pass the wires in the correct place. Had to drill it.
2. We missed a protrusion in the motors so we had to melt the piece to make them fit.
3. The “fit-in” piece to hold the cover wasn’t printed precisely enough; the cover did not fit.</td></tr>
<tr><td rowspan="2">3</td><td>Cover of the robot.

Here we'll attach both the Arduino UNO board and the Motor Driver Board.</td><td>  </td></tr>
<tr><td colspan="2">Errors detected:

1. The holes where the chips and boards had to fit were not precisely placed. We had to struggle to pass the screws through the board’s</td></tr>
</table>

	holes and as a result they were under a lot of pressure due to the tilt of the screws.	
4	**Front** of the robot. Here we'll attach the Sonar.	
	Errors detected: 1. The holes where the sonar had to fit in were so close to the borders that the printer skipped the thin wall that was separating them. The sonar is now hold by a rubber band.	
5	**Back** of the robot. Here we'll attach the Transceiver.	
	Errors detected:	

	1. The transceiver holder is way too small.	
ALL	Overall view of the structure.	
	Errors detected: 1. The batteries in the inside were too tall. The cover couldn't fit at all.	

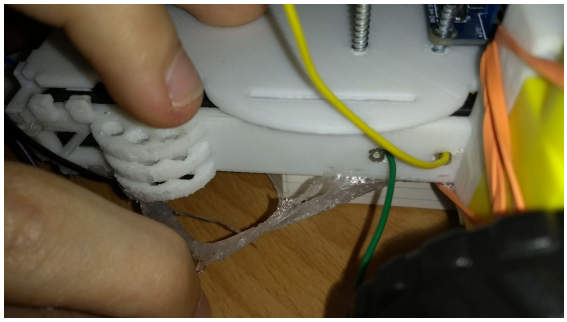
Note:

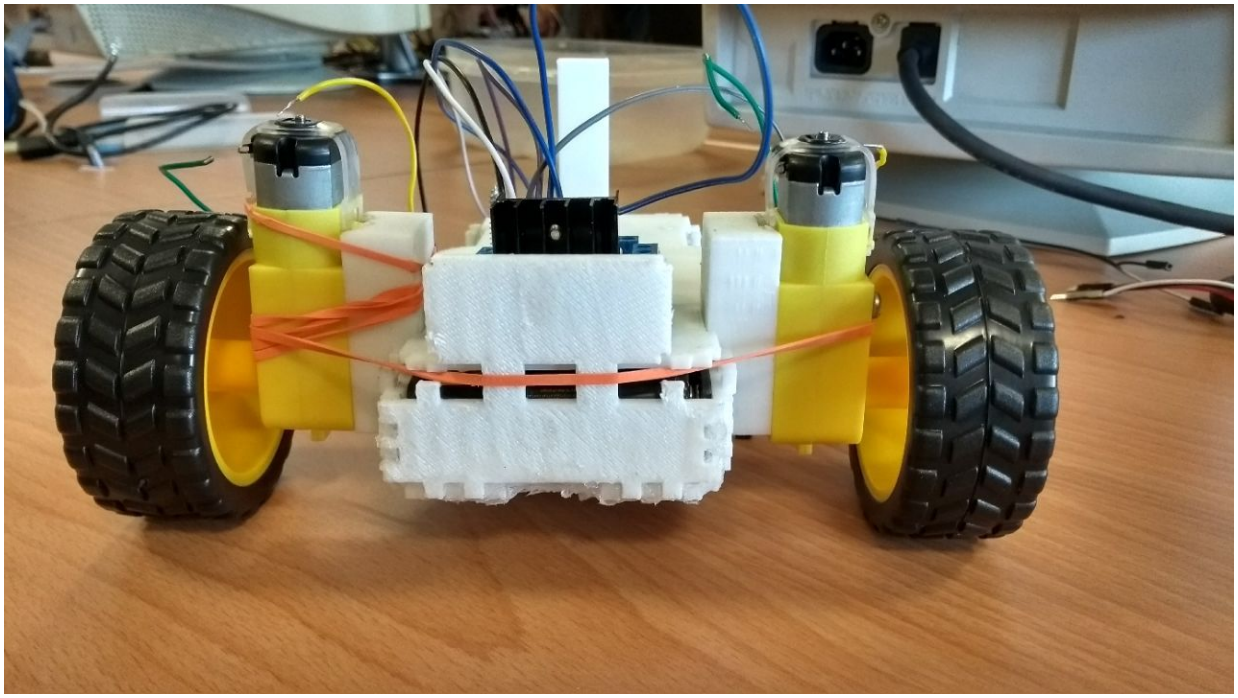
We'll join all the pieces together by sticking their respective hooks and ledges with superglue. However, the cover will use Snap-fit to make it easy to put it in and out, since the Arduino Board and the batteries are inside the robot, and we'll want to have easy access to them.

For future works!

If you're gonna use silicone to stick the pieces together,

DO NOT USE **KITCHEN** SILICONE! The results are AWFUL.

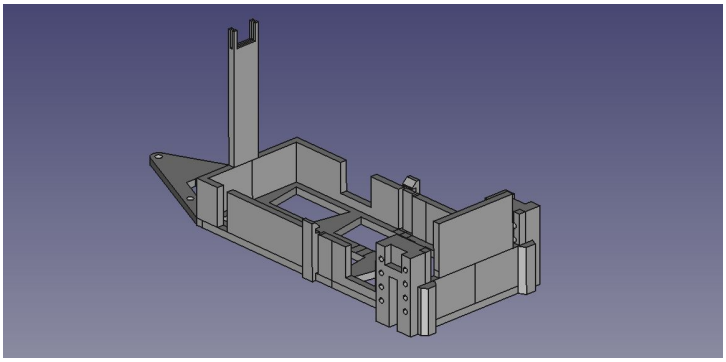
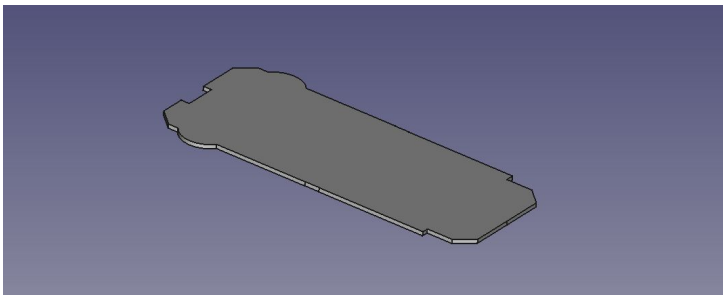


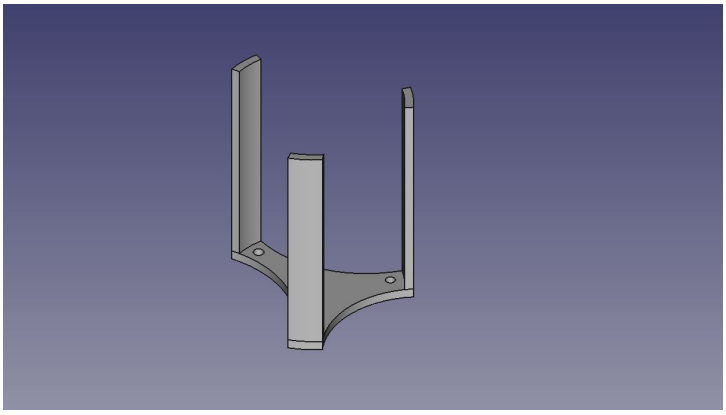


V2 pieces:

With all that we've mentioned before, here are the new pieces that will hopefully fix all of those problems at once.

No.	Description	Image

1	**Base** of the robot. Here we'll attach both the Battery Holder and the Caster Wheel, and sonar, and transceiver, and motors. ...and a mystery piece, coming up next page!	
	Errors detected: 1. Well, the truth is that this piece is already printed. And it has errors... We wanted to print it all in one piece so that we didn't have to join the pieces (see the sticking problems we had before). The problem came with the "fit-in" piece to hold the cover: Just bending it a little was enough to make it break.	
2	**Cover** of the robot. Here we'll attach both the Arduino UNO board and the Motor Driver Board.	

3.	And… Ladies and Gentlemen: With all of you… the **MYSTERY PIECE.** What is its purpose? What was it made for? Will it be useful for the robot at all? SEE **RAT WAITER**!	

Foreseen risks and contingency plan

No.	Description	Probability	Impact	Contingency plan
~~1~~	~~We don't receive the magnetometer~~ We've already received the magnetometer!	~~Low~~	~~High~~	~~Use trial and error to determine the current orientation.~~
2	The magnetometer hasn't got enough precision	Medium	Medium	Direction checks should be done very frequently (stopping the robot).
3	The motors are not firmly fixed (and they make the robot tilt to the right or left instead of going in a straight line) We are currently facing this problem.	Medium	Medium	Redesign motor attachment to the robot and/or use superglue. Contingency plan can't be applied because the problem actually originates from the motors themselves: one seems to have more power than the other. We're trying to compensate for the difference of power by making them move at slightly different speeds.
4	~~Misalignment or rupture of the snap-fit joints (between the cover and lateral 3D pieces).~~ Actually happened, but we've redesigned the 3D pieces.	~~High~~	~~Low~~	~~1. Redesign the joints and reprint the pieces.~~ ~~2. (In case 1 keeps failing) Design and print a hinge which will be used to attach the cover with the laterals.~~ ~~3. (In case 2 were to fail too)~~ ~~Just come up with a different way of accessing~~

				~~the insides of the robot.~~
5	~~Misalignment or rupture of the joints to be sticked with superglue.~~ Also happened, but again, we redesigned the 3D pieces.	~~Medium~~	~~Medium~~	~~Redesign the joints and reprint the pieces.~~

RAT WAITER

MYSTERY PIECE

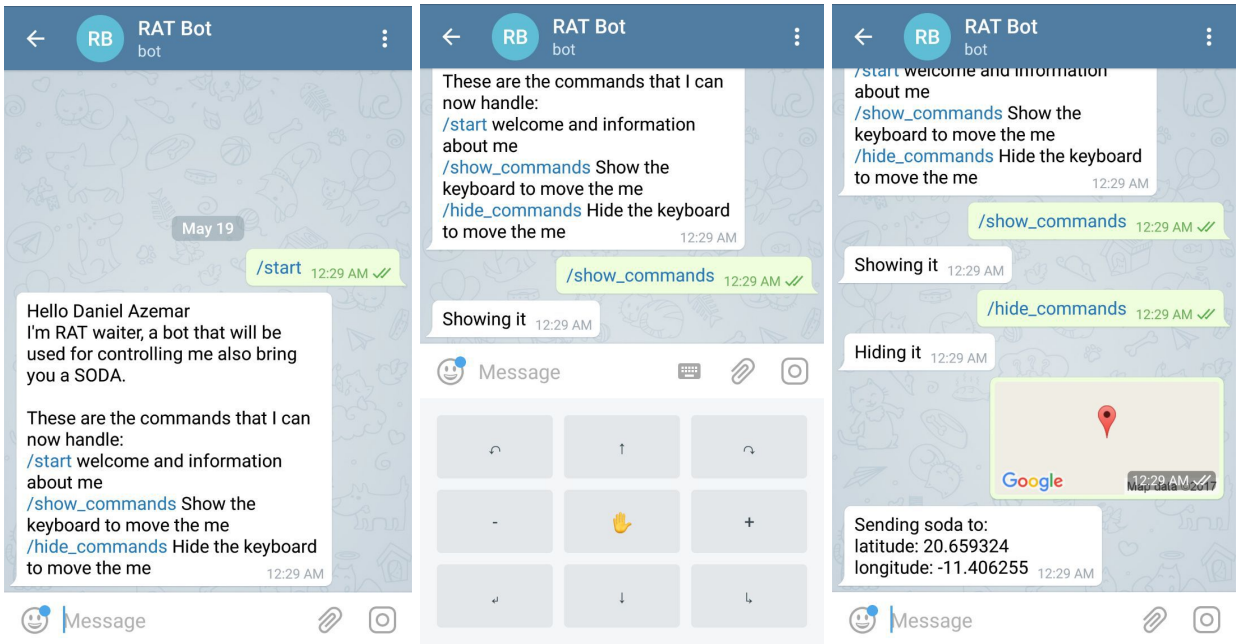
Presentation:

Have you ever thought it's too damn work going down stairs for a soda when you are thirsty at your office correcting exams?

Isn't it tedious meet students at the corridor when you just want to go to the coffee for a beer?

WORRY NO MORE!

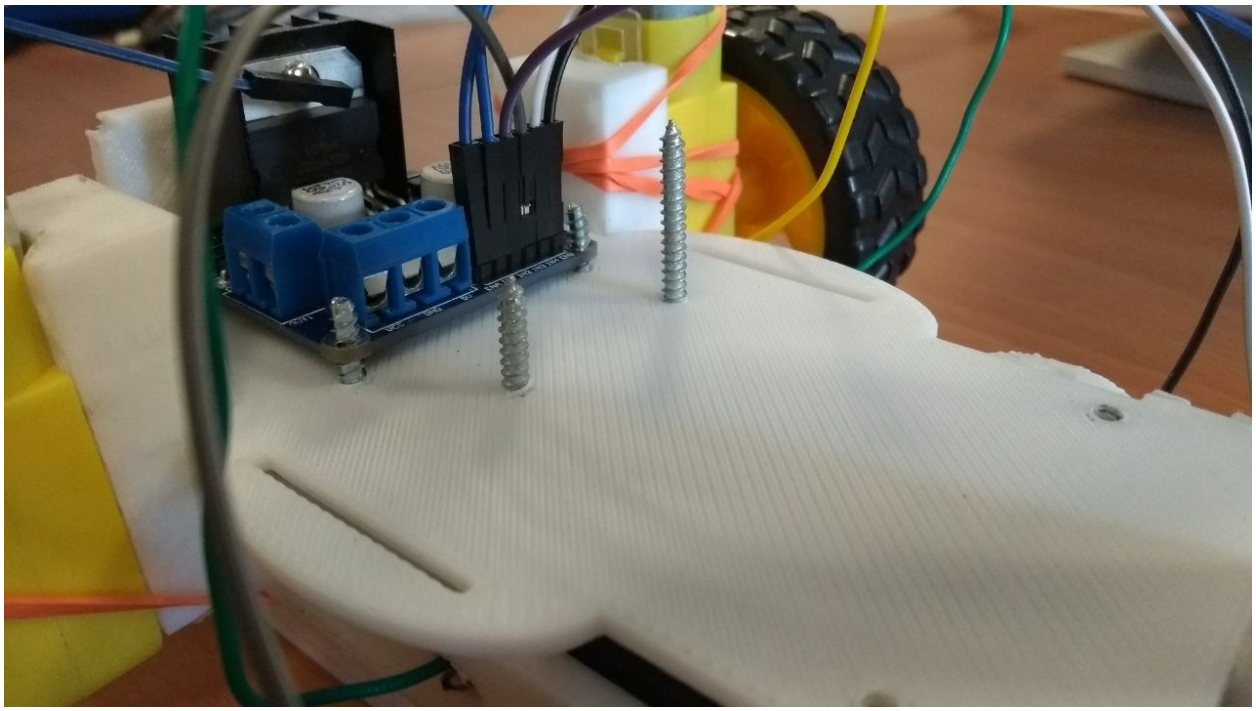
INTRODUCING **RAT WAITER**



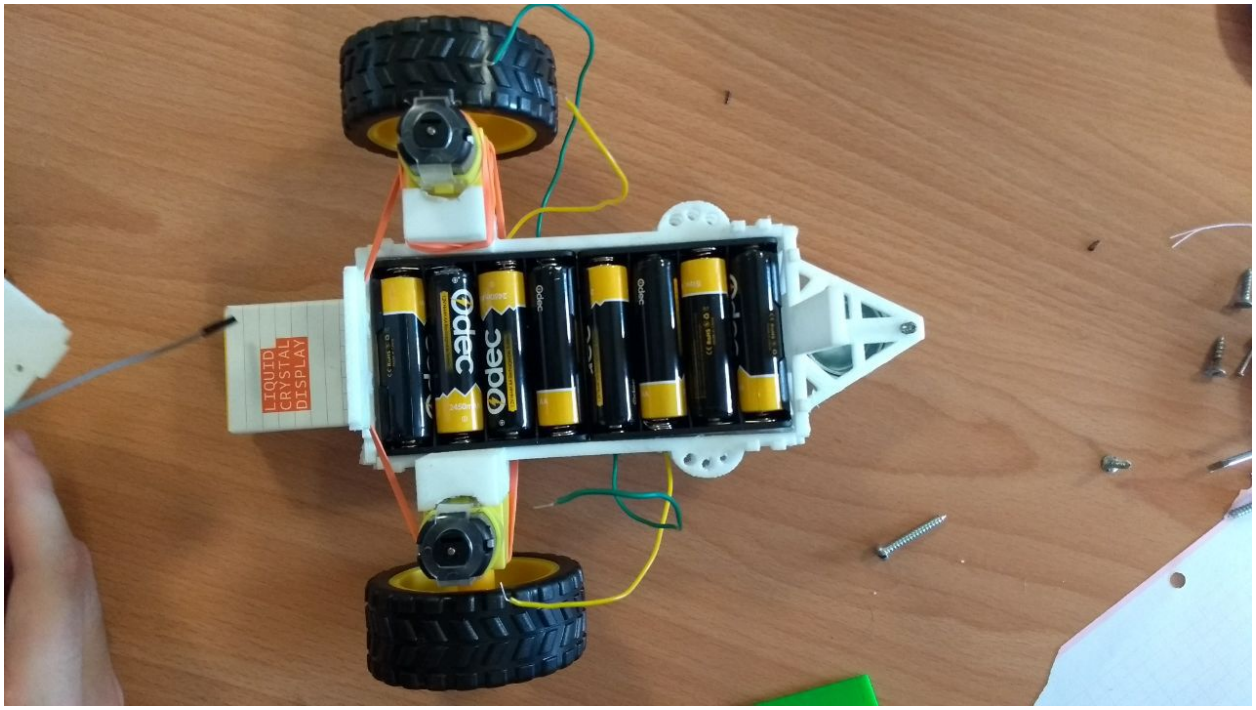
With the new bot we can control RAT in two ways: Manual or Autonomous. The latter is not available yet but you can see the concept of what it's going to be.

ANNEX

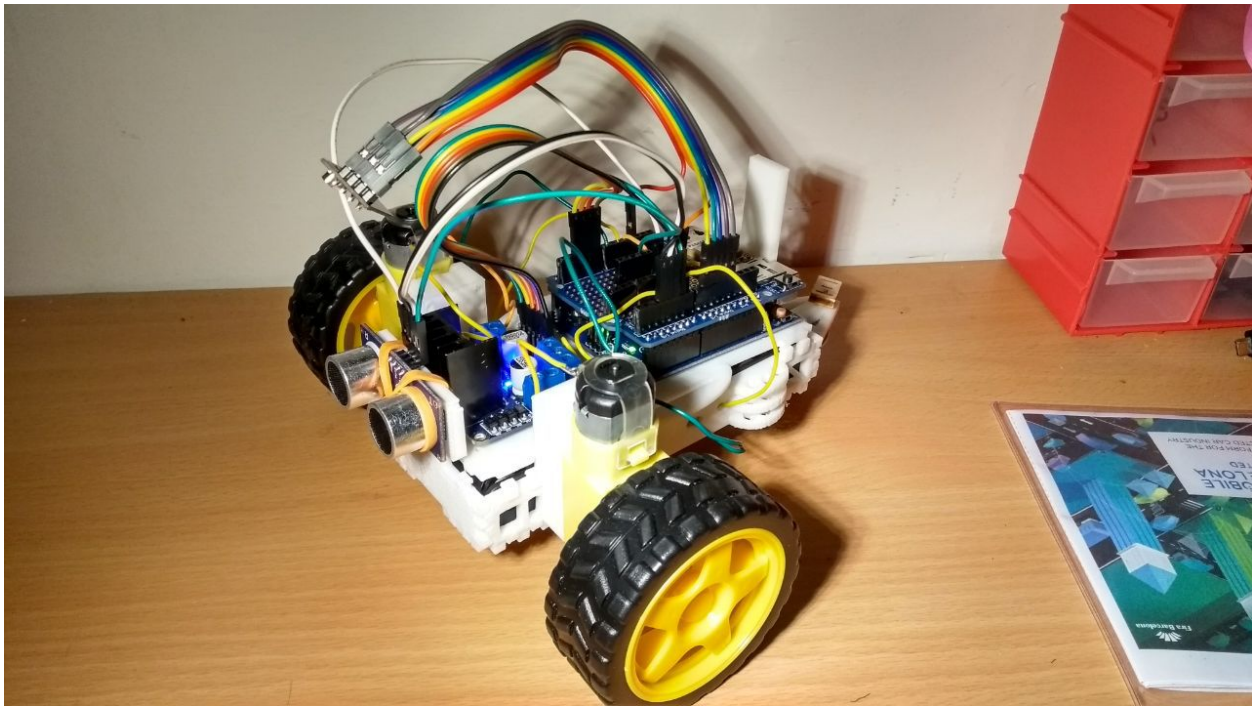
Here we'll show some pictures of our RAT.



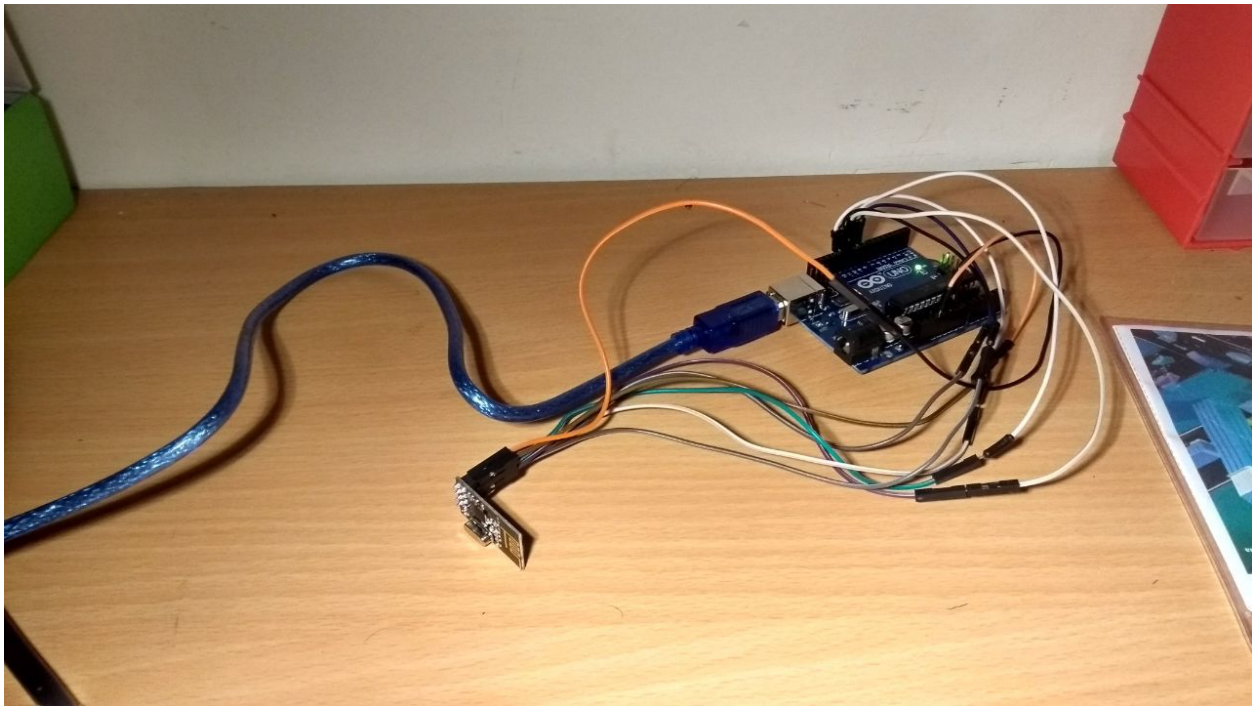
The problems with the screws. It's not appreciable yet, but the chips could eventually break if they diverge too much.



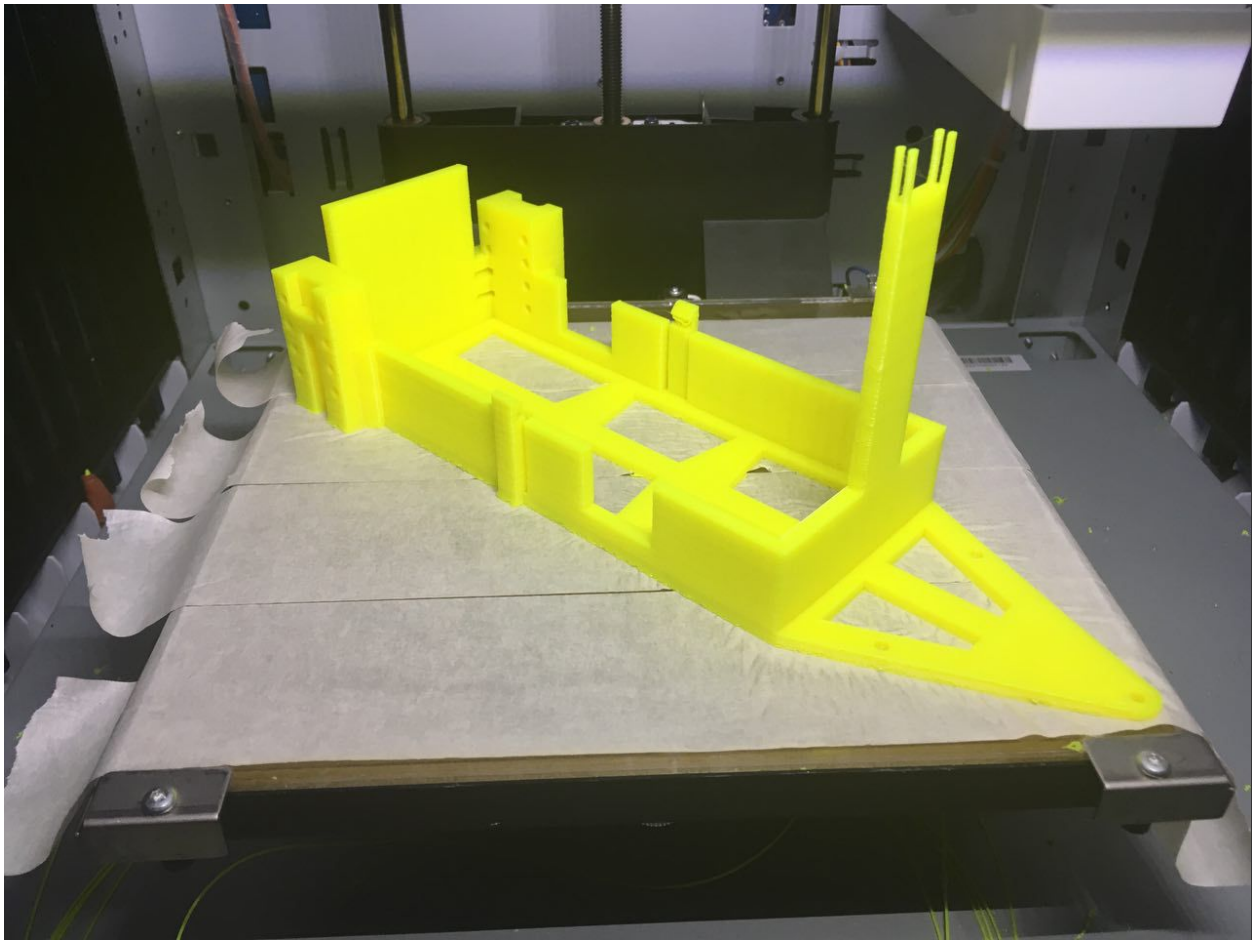
The batteries of the robot. They fit perfectly!



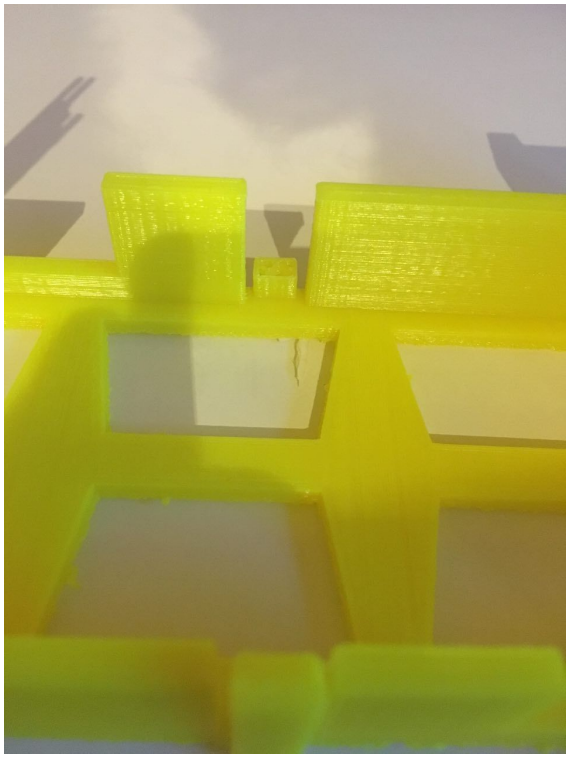
Our robot once (almost) fully assembled. The magnetometer just came yesterday and we've not been able to test it!



And here is the other transceiver, which is connected to the computer running the Telegram bot.



Here is the V2 base piece just coming out from the oven.



And here is the broken part, just coming out from the oven.

Here are some videos demonstrating what RAT is capable of doing at the moment:

First communication test with Telegram bot: https://youtu.be/EcW7nlJUPmk
Remote controlling + obstacle detecting: https://youtu.be/hFyf0Mygbd4

References

This project has been inspired by the following Internet projects:

http://www.instructables.com/id/Arduino-Powered-Autonomous-Vehicle/

http://www.robotshop.com/letsmakerobots/fundamentals-a-gps-guided-vehicle