|  |  |
| --- | --- |
| **MECH 471 / MECH 6621 Microcontrollers for Mechatronics**  **Final Report Project**   |  | | --- | | **BY**  **(UNDERGRADUATE)**  **Astik Patel……………………………....……………………………………..… 27525745**  **Irwin Lopez…………………………………………………………………….... 29372431**  **Mina Khela ……………………………………………………………………… 26270263**  **Tong Yu………………………………………………………………………….. 26586538** | |

|  |
| --- |
|  |

**INTRODUCTION**

This course introduces the fundamentals of computing with microcontrollers using an Atmel Atmega328P. The project consist of design, building, programming an semi-autonomous racing/ drifting car using arduino programming to develop the software of the microcontroller. The car was design using off the shelf and 3D printed parts.

The features that were developed and implemented on the semi-autonomous car are the following:

**CAPABILITIES AND FEATURES**

**Software Capabilities**

**list:**

1. motor speed / car speed

2. traction control

3. breaking control

4. steering control

5. cruise control

6. Collision detection (Lidar)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. motor speed / car speed

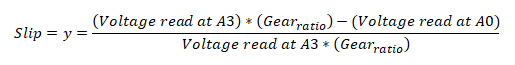
We used PWM control for the esc for more accuracy compared to esc.Write( ). In order to have the car accelerated smoothly, we started off with the minimum PWM to make the car overcome static friction and then added an exponentially growing component to it so that it would accelerated faster to the maximum safe velocity.

2. traction control

Two tachometers are mounted at the front wheels and one at the main motor. The voltage generated by the tachometers at the front wheels is read by pins A0 and A5, while the voltage generated by the tachometer at the main motor is read at A3.



A P-controller is used to control traction. The program uses a reference slip (r) of 0.6. Slip is measured according to the following formula:



The gear ratio is 16/40.

The error (e) is given by the following equation



The assumption is that slip will only occur when back wheels (A3) spin faster than front wheels (A0). The compensation will activate if the slip is greater than 0.6 (ie; e is negative).

The error e is multiplied by a Kp (10).



And after checking for saturation (maximum), the u value is used to control the input to the esc:

Limit of 100 used so that input swings are not too large. Loop continues until slip falls below 0.6.

****

3. braking control

Since the esc did not come with a pre-made breaking function, we could not just set the PWM to 1500 since it the remaining momentum of the car will carry it forward for quite a while after we set the motor to neutral.

To get breaking, we set the esc to 1350 PWM to spin the wheels backwards for less than 1 second in order to stop.

4. steering control

For steering, we are mapping 0-255 values from the left joystick of a PS2 controller to PWMs that corresponds to approximately 35 degrees to each side, akin to a potentiometer. We have also implemented a deadzone on the joystick of approximately 7 degrees to each side to help with user control and to avoid unneeded inputs from the slight twitching of the user’s thumb .

One we originally had was the inability to have the car go straight ahead when not giving steering commands. This was due to various physical factors such as improperly aligned front wheels or difference in weight distribution etc. We resolved this issue by implementing a calibration function for steering into our car, through which we can control the default that the arduino considers to be straight. This was achieved through the use of a calibration value “calib” and then modifying it with user input before sending it off to be mapped to the joystick.

5. cruise control

Using a flag, we have the ability to toggle on-and-off cruise control while retaining steering capabilities. The cruising speed can be adjusted by the user via the up/down on the d-pad on the of the PS2 controller prior to toggling it.

**Optional Capabilities**

6. Collision detection

Lidar is used to prevent collision of the car with objects. The minimum distance to slow down to a complete stop, while at maximum speed was found to be 3 meters, including a factor of safety of 1.5.

It is toggled On/Off using the circle (Red button) in the PS2 controller. The anti-collision functionality is only functional when the button is pressed, which is clearly indicated using an led, attached at the back bumper of the car.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Hardware features include;**

1 X Servo-motor 996R

1 X arduino uno, fasteners

1 X wireless PS2 controller

1 X wireless receiver

1 X idler gear

1 X idler gear support bracket (3D printed)

1 X Front table (3D printed)

2 X Front motor holders (3D printed)

Rear transmission with custom casing (3D printed),

Custom built rear passenger axle

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**RESULTS**

Version 1:

The first version of the RC Drift car gave us an inside of what issues we were going to have. The main issue we had was the front parts breaking due to collisions and weak front components, despite the protective foam. The most error was the rc car only turning in donut due to misalignment, we were able to correct it with the code.

Version 2:

For this new version. An old chassis was used to convert it to car within the specifications.

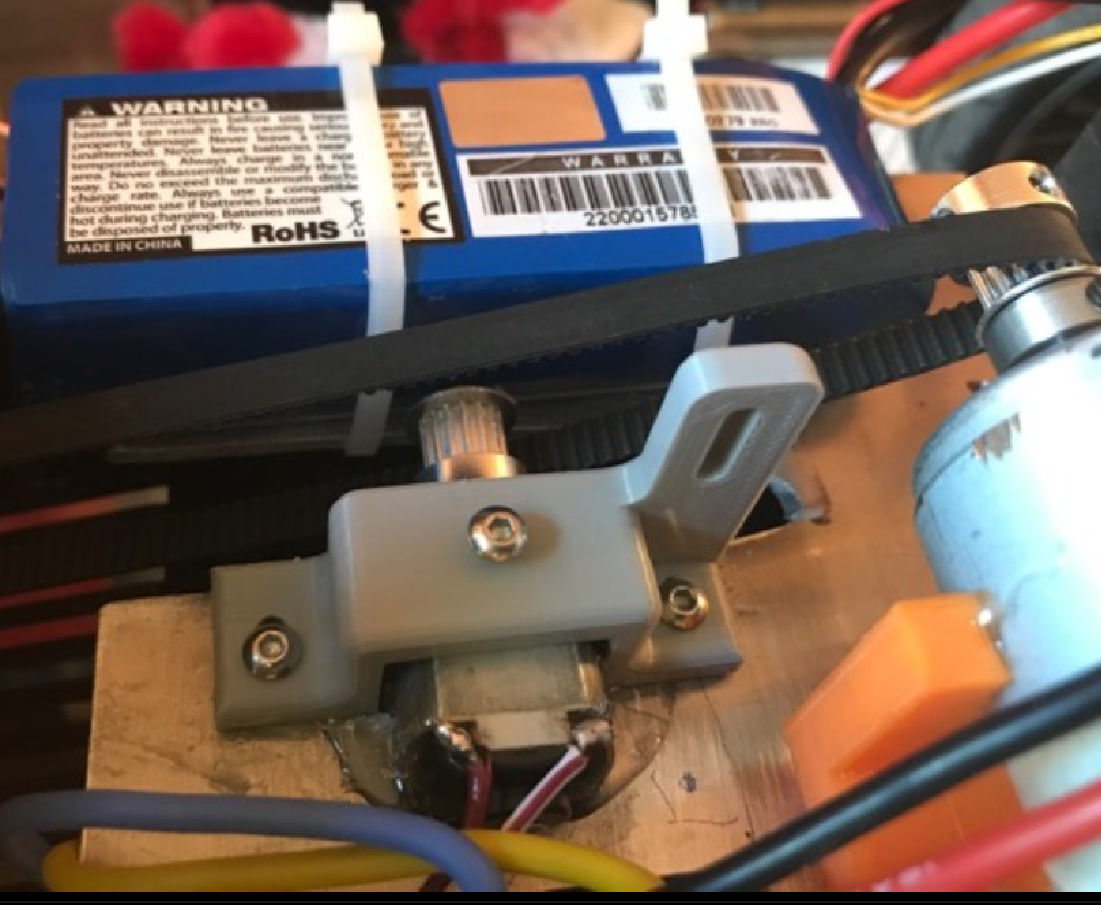
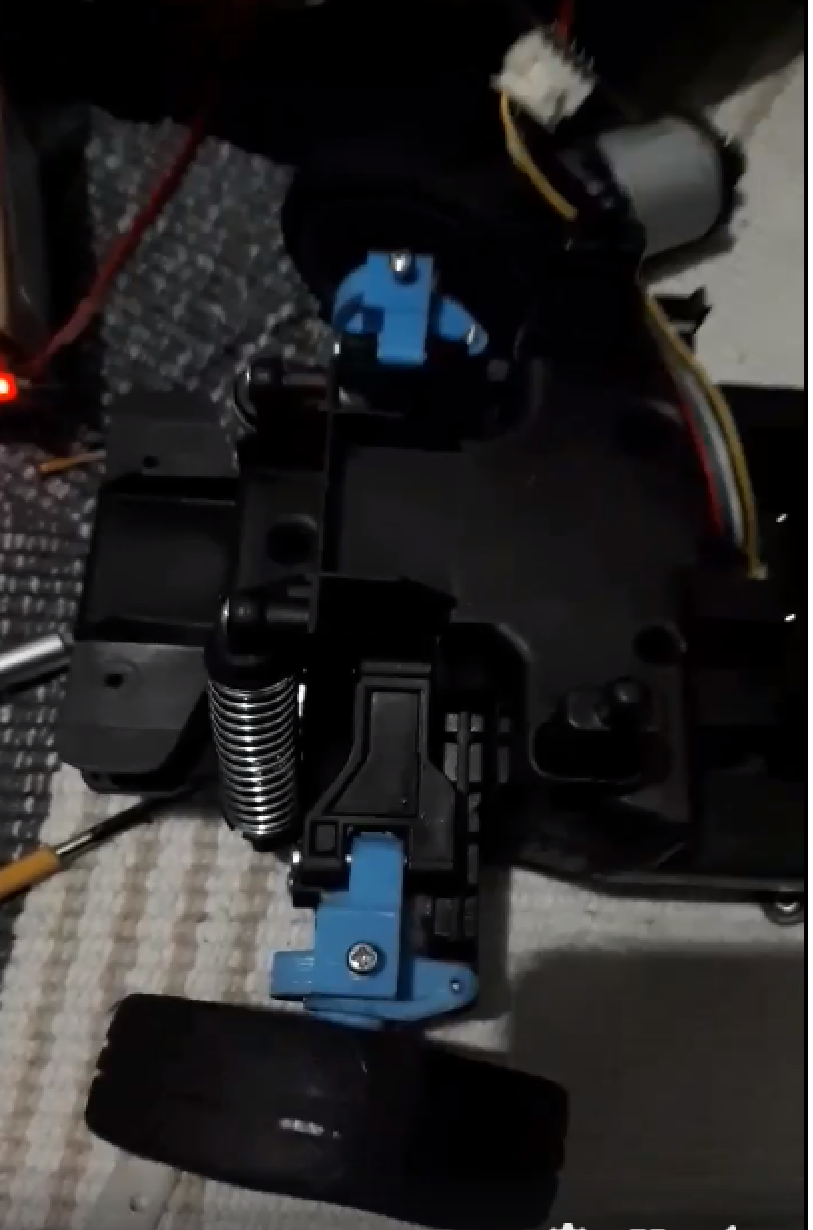
The front axle and front motor support and rear transmission were redesign to custom fit our components, as shown in the appendix A.

The designed arduino programs were debugged mutiple times

**REFERENCES**

-<https://users.encs.concordia.ca/~bwgordon/mech471.html> , Dr. Brandon W. Gordon.

**APPENDIX A**

**1-Front drive-train 2- Tensioner bracket / tachometer holder**

**3- New Transmission / motor bracket**

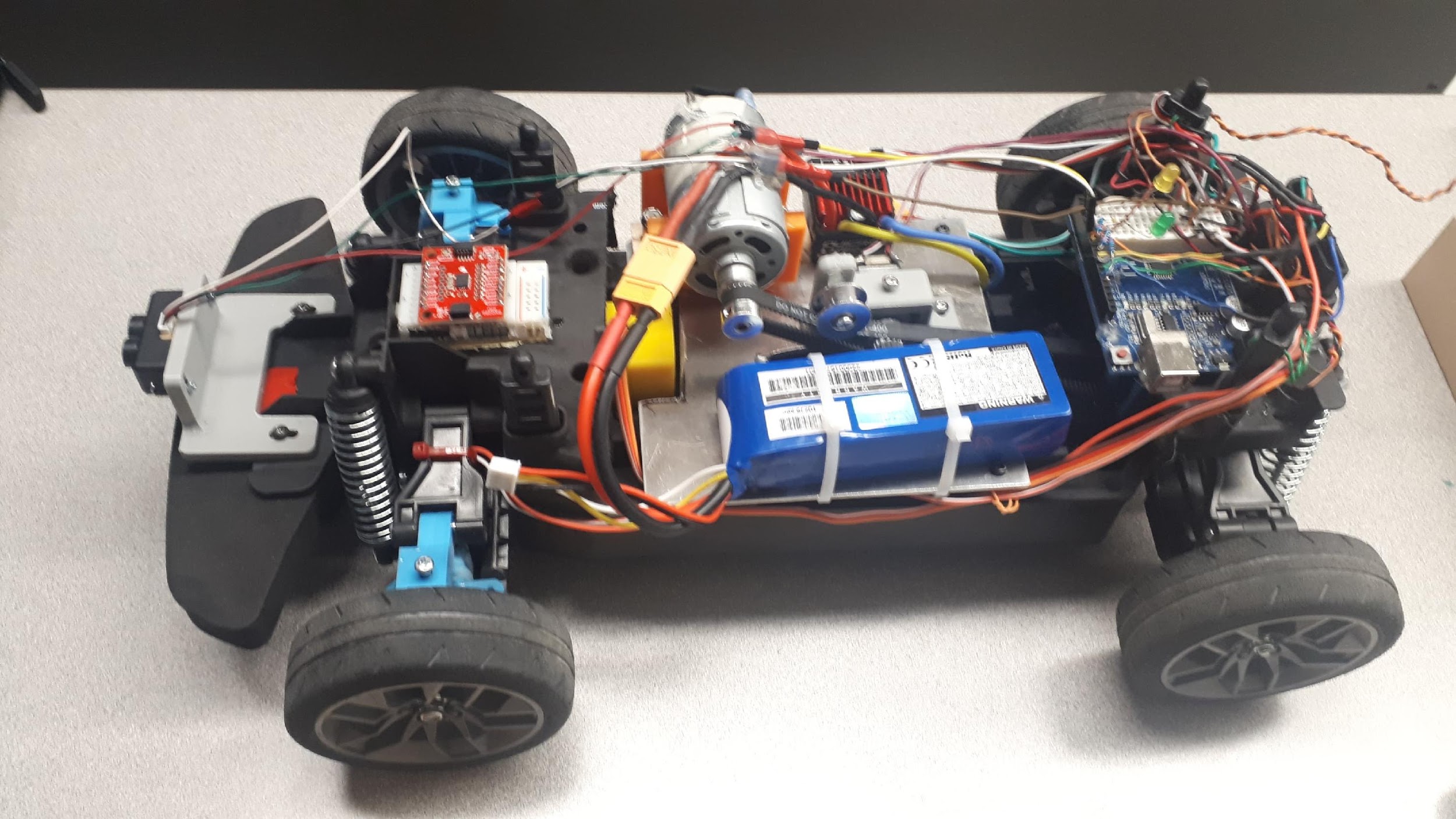




**4-Front holder bracket + wheel holder**



**5. Overview of “under-the-hood”**



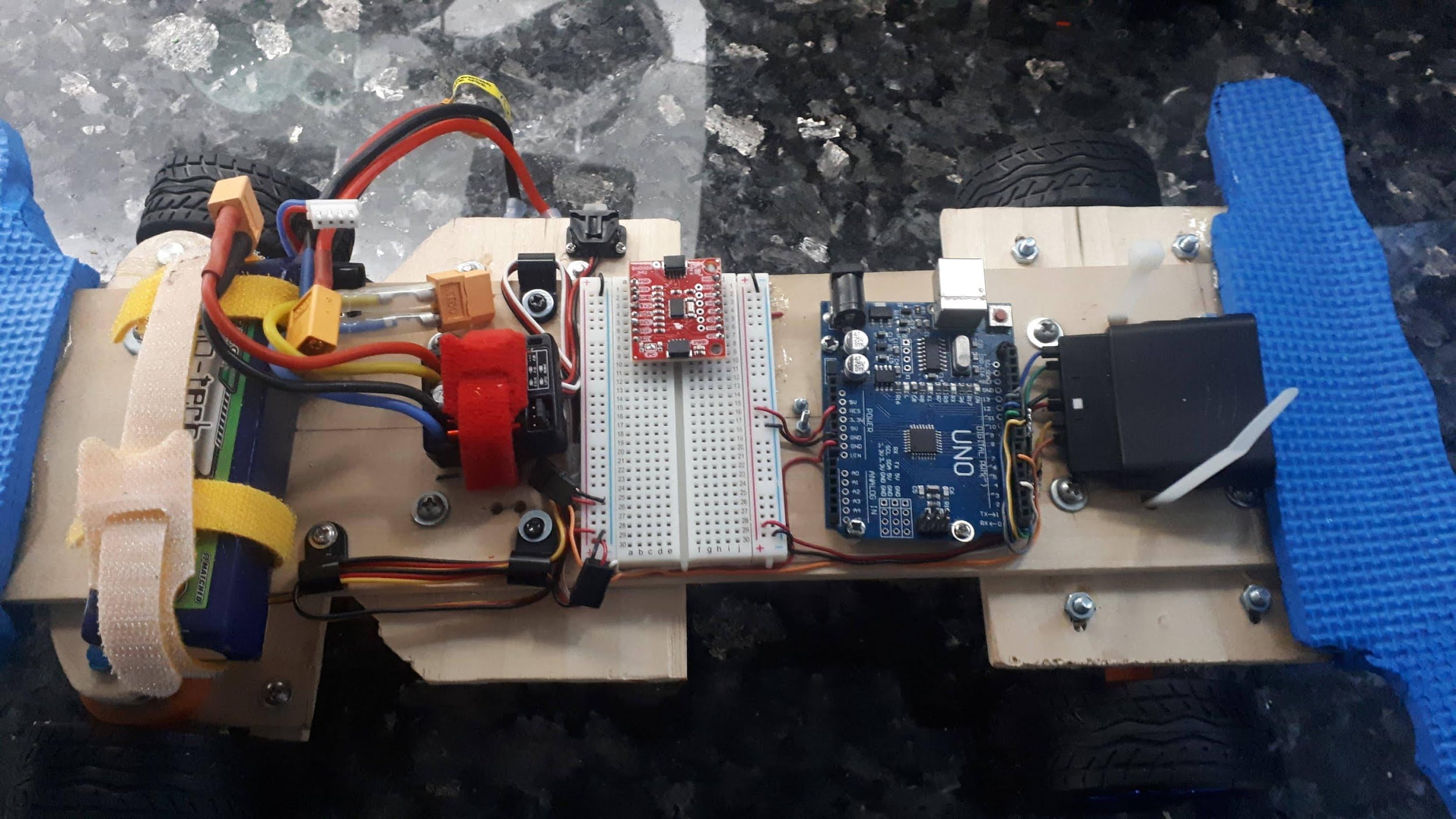
**6. Top cover installed**



**7 - front holes for the lidar sensors**



**8- Version 1 – prototype 1**





**9- Version 1 – broken front steering shaft after testing**

