software field of autonomous car project

Michał Kiełkowski
Łukasz Kołodziej
Michał Bryl
Karol Korpecki

AGH Akademia Górniczo-Hutnicza im. Stanisława Staszica
Wydział Inżynierii Mechanicznej i Robotyki,
Katedra Robotyki i Mechatroniki,
Al. Mickiewicza 30, 30-059 Kraków
mkielkowski@student.agh.edu.pl
korpecki@student.agh.edu.pl
lukaszkolod@student.agh.edu.pl
mbryl@student.agh.edu.pl

1. INTRODUCTION

The purpose of our article it to get you familiar with our work on writing the code for the autonomous car, we are writing the code using github and we are testing it by arduino IDE. The car has two main functions: when connected by bluetooth we can decide if we want to control it remotely from smartphone or put it into the parking mode - that means it will auto detect a parking place and park in that spot. Our project car includes two types of parking and will prioritise the vertical parking over the parallel parking.

The field of our article is arduino IDE https://www.arduino.cc/en/software

Here are some basics programs for arduino with the electric connections
https://www.tinkercad.com/learn/designs

Our Project is based on Arduino Mega microcontroller. We started writing the code, which was based on some examples from instructables.
https://www.instructables.com/Autonomous-Parallel-Parking-Car-Making-Using-Ardui/

Our main inspiration was such article:
http://kth.diva-portal.org/smash/get/diva2:1200483/FULLTEXT01.pdf

2. PURPOSE AND SCOPE OF WORK

The main goal of our project is to achieve a real-life, working model of a car that is able to detect parking spots and park independently in two different parking modes.

Our report consists of essential information about the working principles of our project. It will guide You through our thinking process while designing, building and testing our car.

3. OVERVIEW OF EXISTING SOLUTIONS

Autonomous car concepts have been in use for some time now. Real-life autonomous cars are becoming more and more present in streets all around the world. The first concepts of such self-driving cars were constructed in the late 1920's and were based on radio control. Next big step for
autonomous cars was 1970, when the first self driving car was built by Hans Moravec, who constructed a car that was able to recognise and avoid obstacles on its way. Unfortunately, because of technological barriers, the car was able to move at speeds about 1 meter per 10 minutes (due to long and complicated calculations that exceeded computer's abilities) [8]

There are several existing autonomous car prototypes. Engineers from around the world are working on constructing consistent, safe and convenient solutions for autonomous transport. One of the most popular concepts is Tesla - electric-powered autonomous cars that are the concept of Elon's Musk company. Tesla's impressive crash-avoidance system has proven great during testing as well as in real-life cases.

Tesla model S

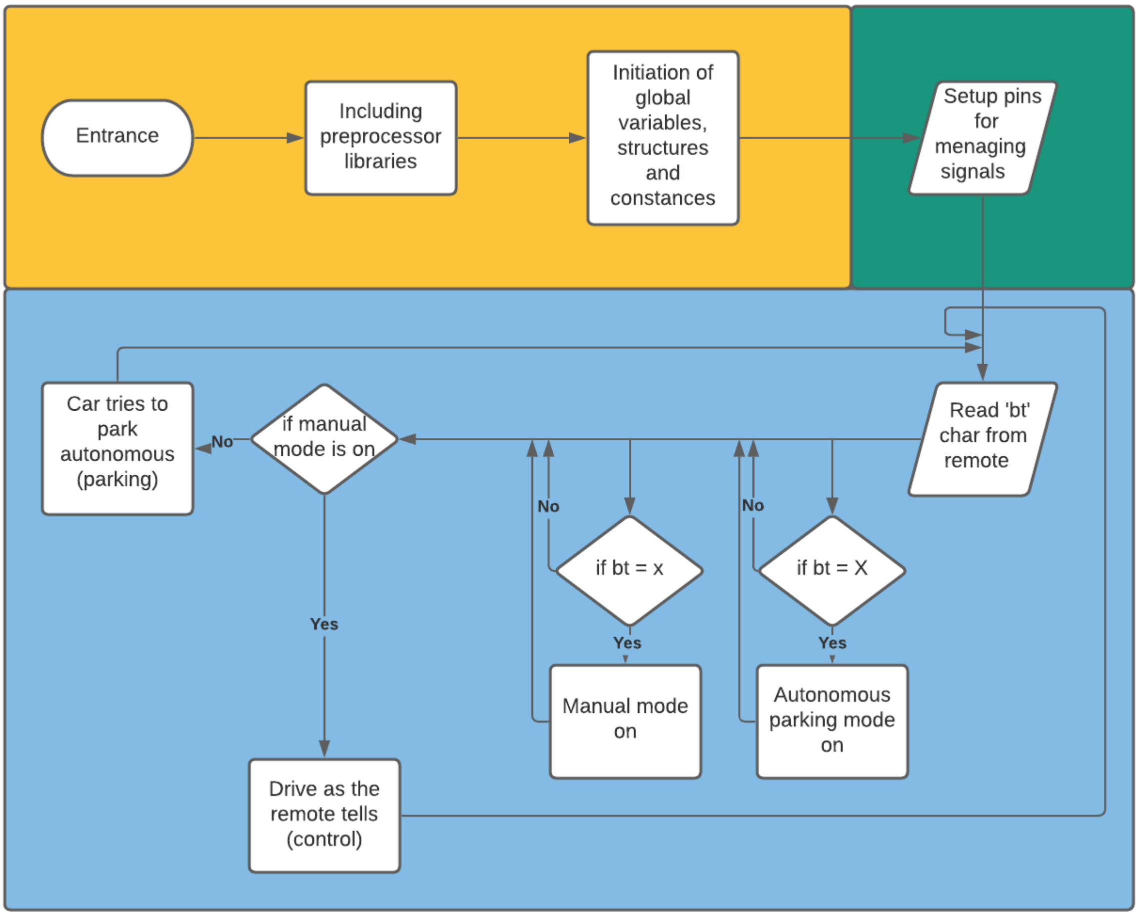
Tesla Roadster

4. MORPHOLOGICAL CHART (it is not necessary to put MC inside short articles)

MORPHOLOGICAL CHART FOR SELF PARKING CAR					
IN:	PROBLEM	SOLUTION 1	SOLUTION 2	SOLUTION 3	SOLUTION 4
MECHANICS					
1	How to turn the car?	Rack and pinion	4 wheels drive		
2	Type of motor?	DC motor	AC motor	Stepper motor	Servo
3	Gearbox?	No gearbox	planetary gears	belt gear	bevel gear
ELECTRONICS					
1	Microcontroller?	Arduino uno	Arduino mega	Raspberry PI	Husarian
2	Controlling motors?	Motorshield	L298N	H-bridge	
3	Power supply?	9 v battery for everything	9 v for controller, 3 times 18650 (12V)for motors	DC charger on cable	9 v for controller, LiPo (12V)battery for motors
Safety criteria		100%	75%	100%	90%
Power criteria		15%	100%	60%	100%
Price criteria		15pln	50pln	20pln	100pln
SOFTWARE					
1	data saving?	internal	PC	cloud	sd card
2	recognition system?	sensor analisis	AI	mapping	
3	Way of calculation?	processor	PC	cloud	mobile

The proper solutions were chosen by the help of Dr.Grabowski. We designed the concept of our car and tried to find the most suitable solutions for both mechanical and electrical parts. Apart from technical problems, we also had to consider the price of components and decide which solution would be the most suitable. We based our project mostly on parts we had, so that we wouldt spend so much money on additional components.

5. MODEL AND SIMULATIONS



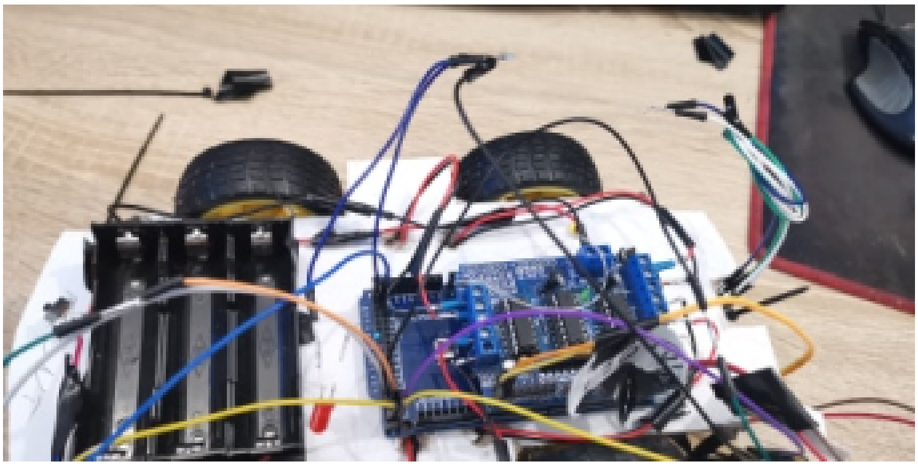
The general algorithm looks like this.

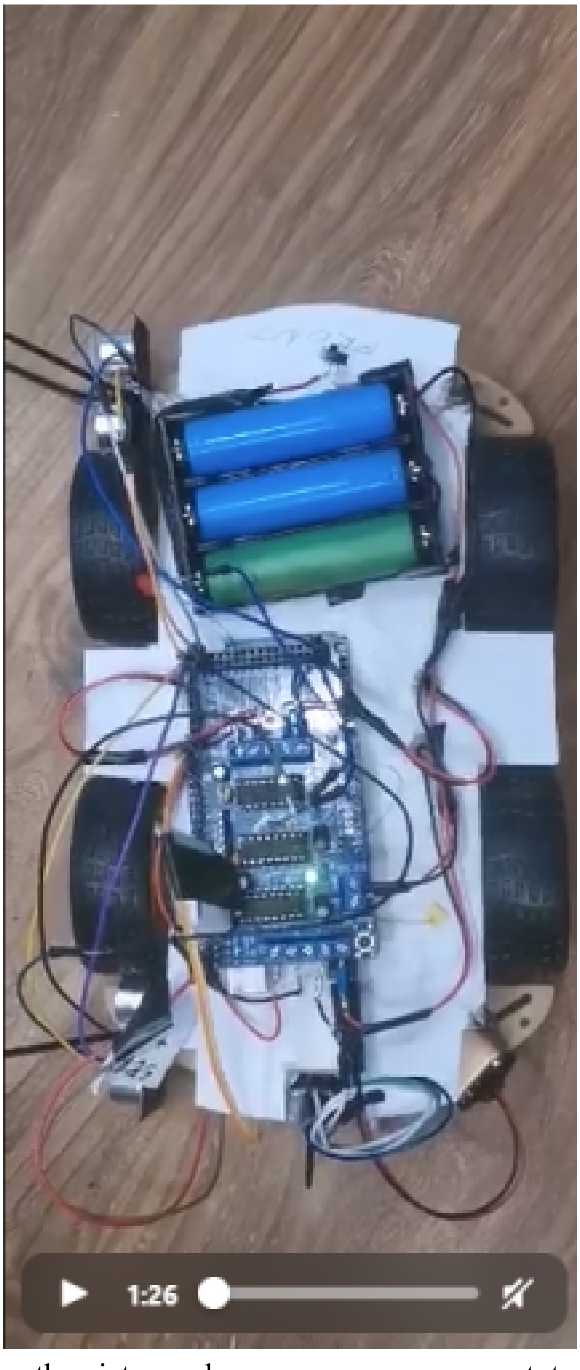
6. PROTOTYPE AND TEST

https://youtube.com/playlist?list=PLDGJH9v0cezfD7j6TbVhIkFeGWAXpRbrM

Link to our youtube videos from prototyping

We decided to build a real-life prototype as soon as possible to be able to regularly test our software. It consists of 4 dc motors, arduino mega, arduino motor shield, arduino bluetooth module, 3 ultrasonic distance sensors and one LED diode.





on the pictures above you can see our prototype

test we conducted for the first prototype:
-if the bluetooth model connected
-if the car can be easily driven by your phone
-if the speed of the car is good
-if by clicking the “x” button on your phone the car goes to park mode and cannot be moved by the operator

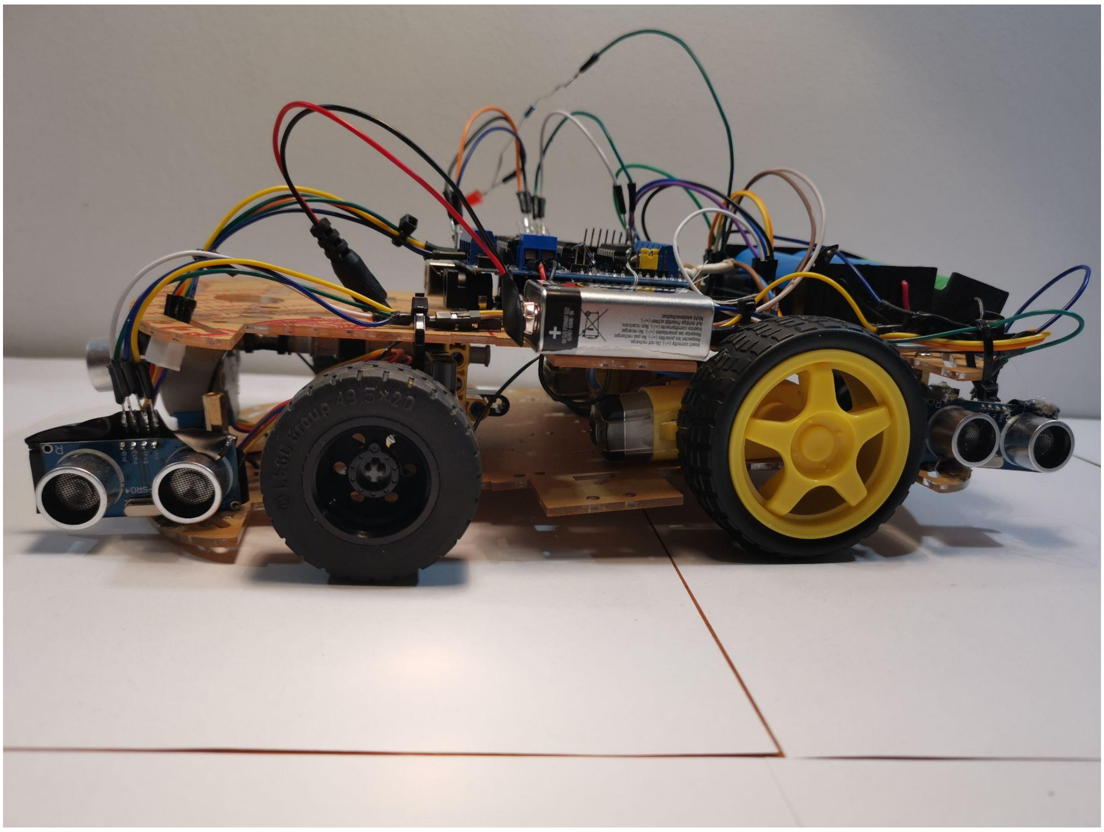
-if the LED lights up so we can see that the arduino detected the park mode
-if when park mode is on it will go forward
-if the speed of the car going forward is good enough
-if the 3 sensors that we have so far are working properly
-if it can park vertically
-if it can park parallel

Each of the tests that we concluded above was successful beside the parallel parking because the sensor cannot read the values if they are not parallel to the wall.

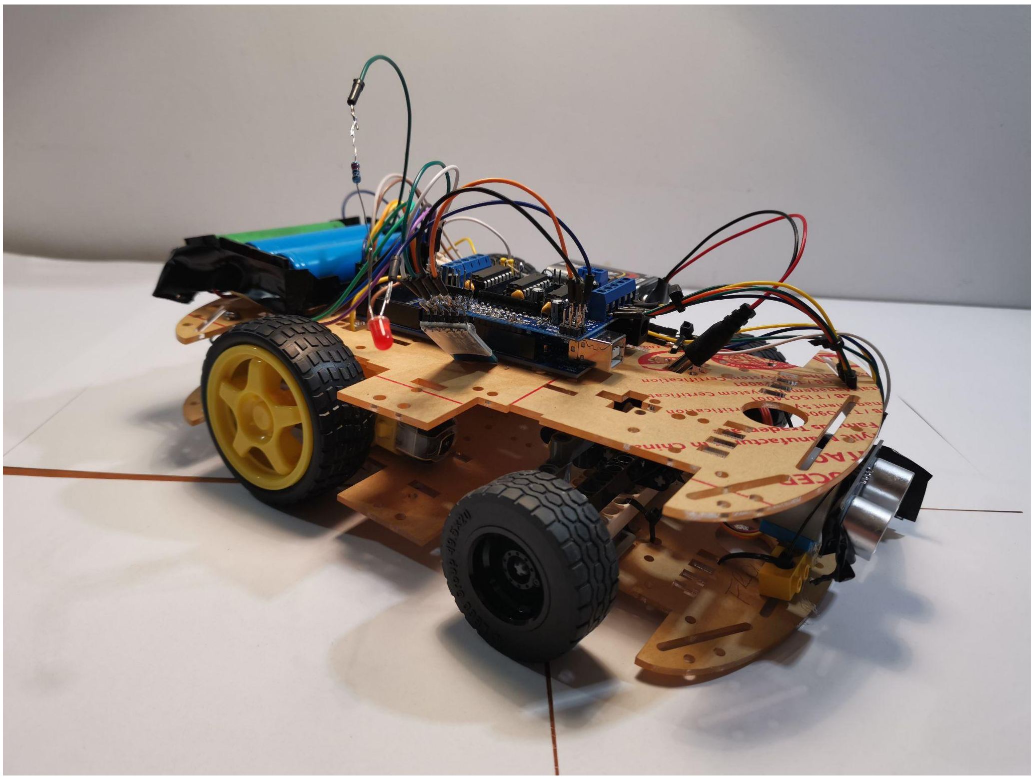
Based on such a setup, we were able to achieve a prototype that helped us test each function of our code. The greatest weakness of such a solution was that it behaved tank-like so it was capable of performing turns while standing which is a huge simplification of the whole parking. Next thing we learnt using this protohype is that the ultrasonic sensors do not work when the reflected surface is not parallel to the

Second prototype:

We decided to build a real-life prototype as soon as possible to be able to regularly test our software. It consists of 2 dc motors, car-axle, arduino mega, arduino motor shield, arduino bluetooth module, 4 ultrasonic distance sensors and LED diode. Based on such a setup, we were able to achieve a decently working prototype that helped us test each function of our code.

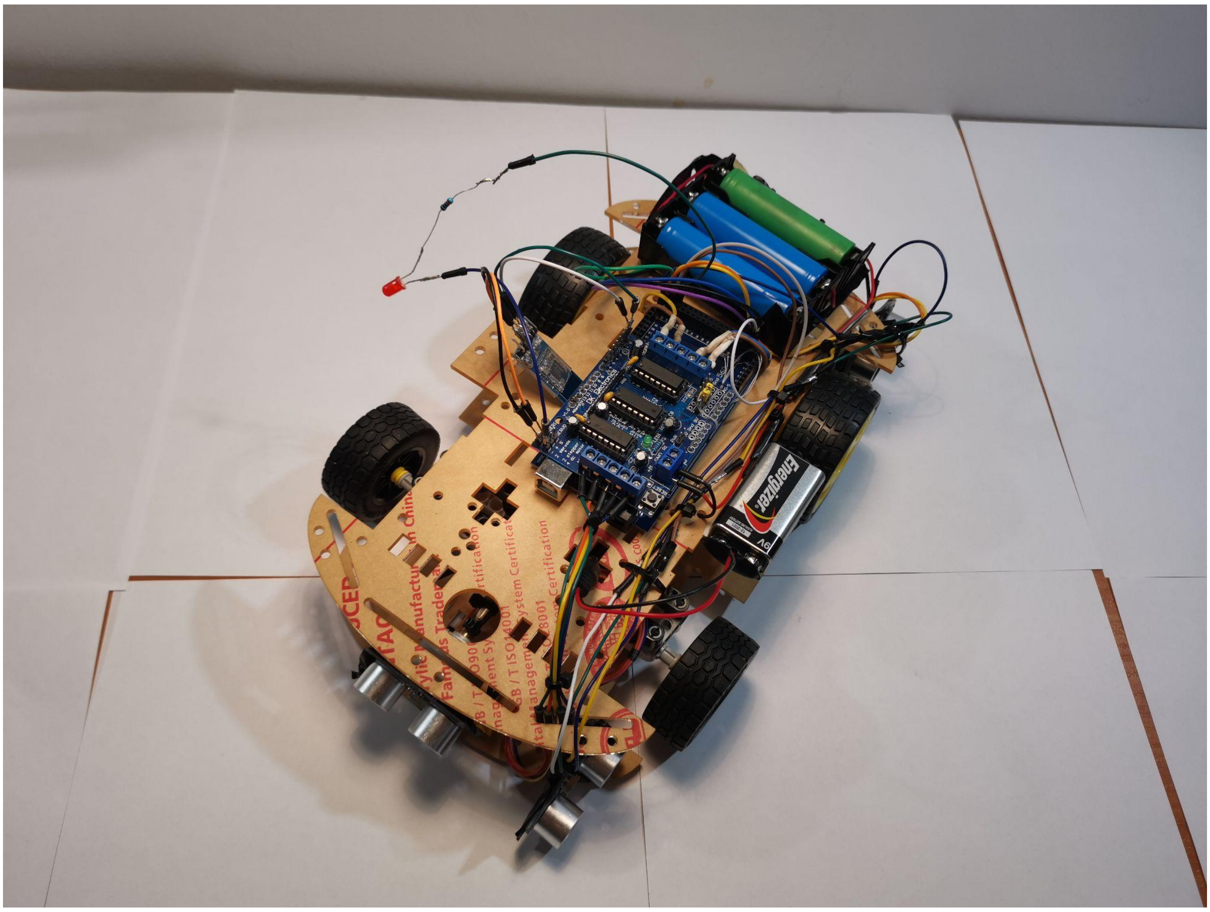


Within the tests conducted on the second prototype we managed to determine why the first approach we have chosen was wrong. The mathematical methods we used to park could not work due to unpredictable frictions inside the DC motors brushes and in front wheels themselves.



The results made us change the approach, so we have decided to determine parking conditions and parking algorithms using the experimental way.

The method was successful and allowed us to find proper values.



7. MEASUREMENTS AND ANALYSIS

During our prototyping we encountered numerous problems connected with sensors and measurements.Tank version of our prototype had a problem with parallel parking, strictly when going back he would hit the wall every time. To check what was wrong, we conducted a number of tests using a serial monitor to read values from the sensors. Tank version of our prototype showed that ultrasonic distance meters are not accurate while measuring distances to objects that are set at an angle.

Axel prototype

Parallel:

To make a function that was responsible for parallel parking in our axel prototype, we conducted a series of tests that showed the amount of time that the car should drive backwards when the wheels were turned. We knew that the maximum angle of our axle is about 35 degrees, so the turn radius of our car was a bit more than we expected. Overall, we combined the knowledge about the turn radius and the time for the car to drive backwards, to achieve a working, self parking model. Also, due to the correct final formula for the parallel parking, we could derive minimal assumptions for the parking place that our prototype car is able to park in.

Vertical:
Vertical parking mode was based on the experience for parallel parking. We tested the time for which the car was supposed to drive on turned wheels before straightening them into neutral position and driving slowly towards the back wall. We chose time values so that the car would be as perpendicular to the back wall as possible.

8. SUMMARY AND CONCLUSIONS (1-6pages)
- it is the MOST DIFFICULT PART

PART 1	-We have done two prototypes. First gave us the basic concept of the entire project. In this prototype we made a basic code for measuring the distance of the parking spot,also basing on that prototype we prepared our first working algorithm for detecting the parking spot and checking if the conditions for parking are sufficient. Using the knowledge gained by building the first prototype we prepared the second one with a steering axle. While creating the second prototype we came across multiple problems,to solve them we created the console which showed us measurements made by sensors. Finally we determined the missing data ,which are necessary to perform parallel and vertical parking, experimentally -we have achieved to usage of two main functions so the car can be controlled remotely and the parking mode can be activated, so far when the parking mode is activated the car moves forward at a certain speed and scans the surrounding with it's three ultra sensors, we have already written the code for the parking below but we haven't tested if it works yet
PART 2	-Most interesting part of our project was to observe and be the part of a workflow of people that have never seen each other before and have never worked together before. It made it a bit hard to start, but as some time passed we achieved a great system for teamwork and moved on with good pace. -The hardest thing to overcome in our project was to overcome the lack of mechanical parts and possibilities of building a real-life prototype due to the current world situation and COVID. Despite that, we managed to move on with the project and build a prototype for testing. -we analyzed the results by the simple testing process, we made some changes in the code then we powered the car and observed how it behaves. Creating a real-life prototype took hours of work and effort, but made our software more predictable and better due to the possibility of live testing. -Observations: having the right code is not sufficient for the project to work. You always have to keep in mind that pins can be connected wrongly, that the batteries could not deliver the power needed, the correct parts are also very important, some arduinos are not powerful enough to make the calculations fast enough. There are many more things that can go wrong and we had to be aware of everything that was being done in order to maintain a working prototype.
PART 3	- future work We are not planning to stop our work, and continue to develop our knowledge and abilities for future projects. - what will I be doing next? We are planning to make an automatic machine which can make variety of cocktails. The project has to properly measure the level of liquid located in containers.

9. BIBLIOGRAPHY

[1] https://www.arduino.cc/en/software
[2]https://www.tinkercad.com/users/46VKyXijIJE-mkielkowski?category=circuits&sort=likes&view_ mode=default
[3]https://www.instructables.com/Autonomous-Parallel-Parking-Car-Making-Using-Ardui/
[4] http://kth.diva-portal.org/smash/get/diva2:1200483/FULLTEXT01.pdf
[5] https://pl.wikipedia.org/wiki/Historia_samochodu
[6] AGH University www.agh.edu.pl

10. APPENDIX (ANNEXES)

- Autodesk Fusion 360: https://www.autodesk.com/products/fusion-360/overview - Eagle Free: https://www.autodesk.com/products/eagle/free-download
- SciLab: https://www.scilab.org
- FreeDyn: http://www.freedyn.at/download/freedyn/
- Libre Office: https://pl.libreoffice.org
- Blender: https://www.blender.org
- Inkscape: https://inkscape.org
- Gimp: https://www.gimp.org
- KiCAD EDA: https://kicad-pcb.org
- Shotcut: https://shotcut.org/download/