HEROTECH

FTC #180

Technical Book

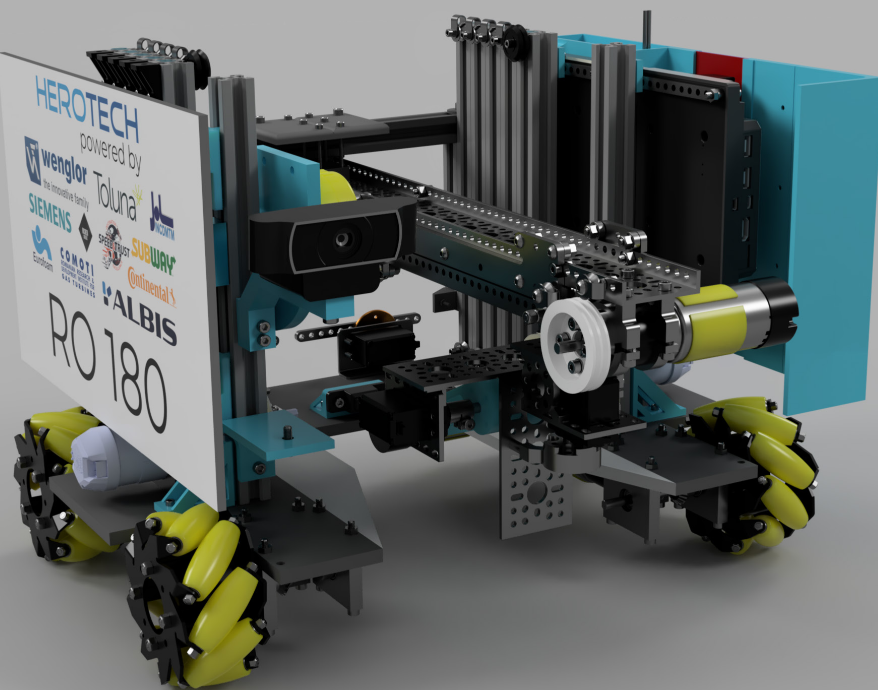

HEROTECH
powered by
wenglor
the innovative family
Toluna
SIEMENS
SUBWAY
Continental
ALBIS
RO 180

Table of contents

Technical
book

~~Prolologue~~

We thought of making a team to increase the STEM community in Romania, as such, in autumn of 2019 we choose the team members, people that fitted the FTC standards, a team that not only inspires gracious professionalism, a team that feels like home, like a group of friends, people that see robotics not as a means to an end, but as a passion, as a life style, a hobby that transcended into a way of thinking and living.

Our team started as three friends that found joy in coding and technology, the founders of HEROTECH, Marpozan Daniel, Muntean Codrin and Rădac Alex. Aces in their respective fields: Daniel in robotics, Codrin in coding and lastly Alex in mechatronics. At first there were a few small contests, anything from school to some regionals, even though they didn't bring prizes or recognition they played a crucial role in the future of HEROTECH, they bonded us.

One day, the small contests were suddenly overshadowed by something greater, a contest so huge none of us knew how to react to it, the Genius Olympiad. We found out about this contest from a university teacher in Oswego named Adrian Ieta. He challenged us to win and we challenged ourselves to participate. We didn't intend to win; our goal was to get there and get the experience from it. The journey was a difficult one filled with ups and downs. Firstly, the sponsoring and material part of the project imposed a great difficulty for us. We didn't know who could give us the amount of money we needed for the parts and other costs. Our first sponsor was Sysmep, we took every company in Sibiu and sent tens of mails to each one, and Sysmep answered first. They marked the true beginning of our story, from that point on the dream became a reality. we bought the kit and started working at Alex's house because we didn't have a workplace. For months we worked and the work payed off. In spring of 2019 we got an email saying that we got accepted.

The focus now was finding sponsors to found our flight and accommodations. We live in Romania; the contest was all the way in Oswego and our sponsor situation wasn't well put together. The media played the biggest and most important role. With their help our faces were on the first page of almost every newspaper in Sibiu. In a few weeks we had the money to fly to the U.S.A.

We arrived in New York at night on the 15th of June, from there we drove to Oswego. The contest had begun on the 17th of June with over 1400 participants. We were overwhelmed by the proportions, but we were determined to win. We presented our electricity saver project in the science and innovations category and we won the silver medal. On robotics on the other hand we competed with a robot that was meant to help firefighters in case of an intervention, our project was brilliant, our presentation was on point, as such, we got awarded the gold medal along with 2 other teams. Our dream finally was fulfilled. Besides the experience and journey as prize, we got to see New York in all its beauty.

Motivated by this achievement our next goal was to participate in other bigger and more challenging competitions, one of the greatest being the First Tech Challenge. The first step was making our team bigger. The three of us were overwhelmed by the quantity of work there was. Three members was just too few, as such our team got three new members: Fleșar Radu on coding, Vișa Daniel on mechatronics and Bleaja Sebastian on the marketing, advertising and PR side of things. We registered in the FTC community in Romania, after which we started to look once again for sponsors. Sadly, we didn't get the 5000 euro grant. After working with some of our past sponsors, we managed to get the minimum sum for getting the base kits. We built our own playground in our new base, the Wenglor company, which provided us a working space in their factory. Alex worked on projecting the robots base using Autodesk fusion360 program and our new journey began.

Chapter I

Activity Log

First

1st hour of working:

The first hour was challenging. Monday, the 11th day of October, the shipment came, sadly, some parts being custom made were delayed and couldn't arrive.

Second

2nd hour of working:

We looked over the parts and ordered what was missing.

Third

3rd hour of working:

We sorted all the parts into categories to make our work easier.

Fourth

4th hour of working:

We started the robot assembly.

Fifth

5th hour of working:
We had to use two different types of motors with different revs.

Sixth

6th hour of working:
We finished the assembly of the wheels and motors.

Seventh

7th hour of working:
We thought of the way the robot would grab on to the blocks and started designing the robot's arm in F360.

Eight

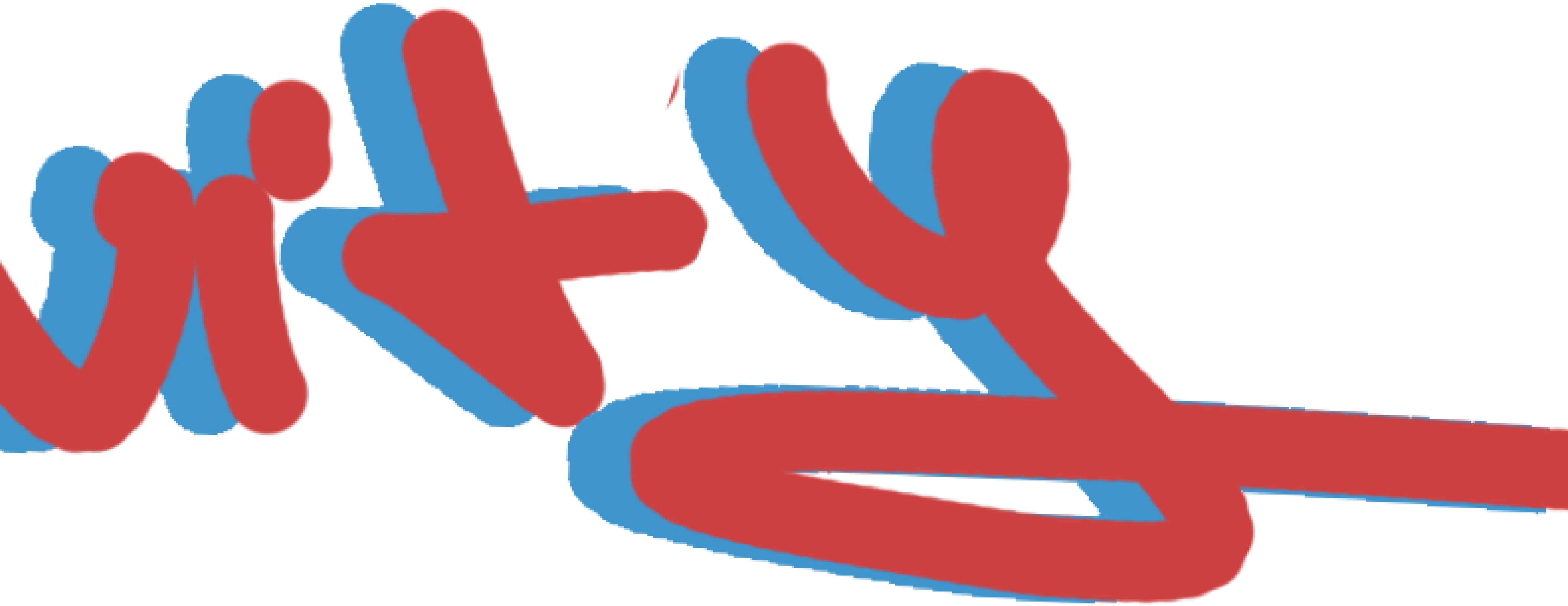
8th hour of working:
We finished designing the robot's arm.

9th hour of working:
We assembled the motors on the robot's arm.

10th hour of working:
We mounted the motors on the robot's arm.

11th-15th hour of working:
We mounted the control hub and the expansion hub on the robot. We ted and organized all the cables and wires. We added a secondary motor to the robot's arm, as one motor wasn't powerful enough.

16th-20th hour of working:
We started to program the robot. We installed the drivers station and configured the motors ports. We looked up the connection for a ps4 joystick to the app. We learned how to program the motors to move backwards and forwards and to spin. We programed the power steering motors for the cube grip arm with SRS programming.



Team Interviews

Marpozan Daniel:

How do you manage your time?

"I'm very organized, each night I plan out the next day. I write everything on the note app on my phone, I set reminders and such, that's how I manage to do everything without forgetting something. I also leave sticky notes on my desk. I try to collaborate as much as I can with my team so that we don't sit around waiting for each other, to be efficient. If I have something to do alone for the electronics part, I set myself a deadline and respect it, even if sometimes it's hard to. All in all, I like what I do so I try to make time for everything."

Is it hard to learn for school and work for Herotech?

"No, because I know how to manage my time and I like what I'm doing. I'm not scared of the BAC exam; I want to focus on my projects this year."

How do you manage your time?

"I'm very organized, each night I plan out the next day. I write everything on the note app on my phone, I set reminders and such, that's how I manage to do everything without forgetting something. I also leave sticky notes on my desk. I try to collaborate as much as I can with my team so that we don't sit around waiting for each other, to be efficient. If I have something to do alone for the electronics part, I set myself a deadline and respect it, even if sometimes it's hard to. All in all, I like what I do so I try to make time for everything."

Is it hard to learn for school and work for Herotech?

"No, because I know how to manage my time and I like what I'm doing. I'm not scared of the BAC exam; I want to focus on my projects this year."

How does a usual day sound for you?

"I wake up at about 6 am, I take a shower and all that morning stuff and then I go to school. Usually I waste my time at school, I sometime miss some classes either because I'm not interested at all in them or because I have more important things to do with my time.

After school is over, if I have a large workload for the next day at school I go home and do it but there are some days in which I go to Wenglor and work on different stuff with the boys."

What are your hobbies and passions aside programing?

"I like cycling and fishing. Also, I like to walk, preferably in nature, I like to go out with my friends and I really enjoy coffee."

What do your friends and family think about what you do?

" I think that both my friends and my family are proud of what I do but unfortunately lately I haven't been able to spend as much time with them, Wenglor is like a second home to me, I spend a lot of time here. My family is supporting me but also worry constantly about how late I get home if we have a lot of work to do. My friends think that what we do is complicated, we learned everything step by step, so I don't see it like they do."

Do they support you?

"Yes, my family is helping me in what I do, going as far as giving me financial support when I need it."

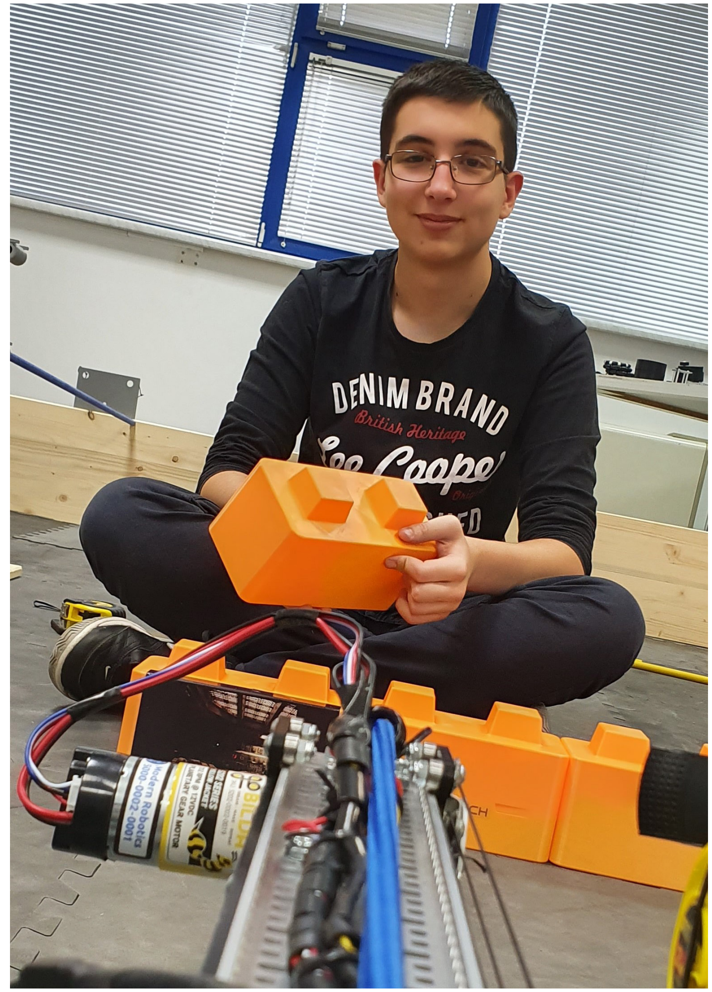
What do you do in the team?

"I'm responsible for the mechatronics part. Me and Alex make a great team. I think that we work pretty efficiently."

Codrin Muntean

How do you manage your time?
“I don’t really. I usually do anything but what I’m supposed to while panicking that I won’t have enough time to do what I know I have to, but when there’s little time left, I really work hard and I feel like I’m very productive. I start working at the beginning of an assignment, not being very invested in it and end up doing it in the last few days.”
Is it hard to learn for your finals and work for Herotech?
“Not really. For now, I don’t really stress about finals. Probably next semester I’m going to use more of my time to study. Obviously, there were times when I had to learn something for school and do stuff at our workspace and it was surely exhausting, but manageable.”

How does a usual day sound for you?
“Well, it depends. If there’s something we must work on, I’ll probably spend at least half of my day at Wenglor. When there’s nothing to code, because the team it’s working on the hardware, I usually assist them or do research for the autonomous programming.”



What are your hobbies and passions aside programing?
“I like technology and being in touch with the latest releases and researches. I love listening to music. There isn’t a day where I don’t open Spotify and shuffle one of my playlists, search for new music or listen to at least one podcast episode. I like intelligently crafted worlds, where mystery plays a big role and where you can uncover the secrets of it by paying attention to details, making connections and theorizing about it. I found these kinds of worlds in videogames (Dark Souls, Blasphemous) and manga’s (Attack on titan).”
What do your friends and family think about what you do?
“My family and friends 100% support me and I like to think they are proud of my achievements.”

What do you do in the team?
“In the team I am doing the programming part. Most of the code goes for the autonomous part, which also needs a lot of thinking for different strategies and for maximizing the outcome for every situation. So, I am responsible with researching for the autonomous task, thinking for a strategy and programming the robot along with Radu Fleșar. I also work on our websites and for different media that we need (logo, banners, posters, etc.) even though I’m not even close to an expert, but I’m trying to improve my design skills.”

Alex Radac

How do you manage your time?

"I usually procrastinate until I have to absolutely start working or I will miss a deadline. If I must do a lot of things in a day, I start with what is easier and finish with the harder tasks."

Is it hard to learn for your finals and work for Herotech?

"It is not really a big deal; I usually learn one hour a day for school, so it doesn't take too much of my time."

How does a usual day sound for you?

"Alarm rings. Snooze. Play a short music playlist. Make a new alarm in 10 minutes. Fall asleep. Alarm rings again. Get out of bed. Go to school. Leave while listening to music on the way back home. Eat. Play video games. Use my PC to design new parts for our robot or go to the place where we all work. Eat. Play the guitar/go to the gym/more video games. Learn for school. Browse Reddit. Eat. Go to bed. Browse Reddit while listening to music. Fall asleep."

What are your hobbies and passions aside programing?

"I love listening to music; I feel like it makes me more relaxed after a hard day. I like to play the guitar from time to time but nothing too serious and I play computer games with my friends."

What do your friends and family think about what you do?

"They are very proud and supportive because they know that what I am doing is not really easy and it will help me in the future."

What do you do in the team?

"I am in charge with designing and helping the assembly of the robot. I create different variants of the robot in Autodesk Fusion 360 and I choose the one which will be most easy to build and perform well. I also model parts that will be 3D printed."

AleX

Radu Fleșar.

How do you manage your time?

"I do a program at the beginning of every day and try to keep it as much as possible. I use my reminders on my smartphone a lot. After school I spend time with my family, we usually eat or go out in town, sometimes I go to the park with my dad."

Is it hard to learn for your finals and work for Herotech?

"No, I don't consider finals to be so hard, I have prep once a week for Romanian and the math I do at school is more than enough. I like learning to code on my own, I taught myself how to code because I don't like to have someone teach me programing."

How does a usual day sound for you?

"I wake up at 06:30 a.m. just about every day, I check my phone for about 10 minutes. I go to the bathroom; I only use cold water as it wakes me up fast and it became a habit. I eat breakfast, which is cereal most of the time, I can't eat anything else so early in the morning. After that I go to school. I walk to school and back listening to music. When I get home, I like to spend time with

my parents usually by going somewhere (to the park, "Big square" etc.).
After that I work together with my team at "Wenglor", a company that lets us use a large room to build our robot. After the work is done, I usually go home.
I like to spend Fridays by going out with my friends."

What are your hobbies and passions aside programing?

"I sing, I sing a lot, I can sing to anything other than the saxophone and harmonica. I also play basketball for 4 years. As a passion I really like Formula 1."

What do your friends and family think about what you do?

"They like that I'm doing what I like even though sometimes they get mad because I can't spend more time with them."

Do they support you?

"Yes, emotionally and financially. My parents are involved in First."

What do you do in the team?

"I do the programming part and I love it. Code always looked easy to me so I usually finish my work and help my teammates do theirs."

Daniel Vișa

How do you manage your time?

"First I think about everything I have to do for the day, after that I try to find a way to solve all the problems and to do all the stuff I have to do."

Is it hard to learn for your finals and work for Herotech?

"No, it's a pleasure, I learn things and meet new people. Also, I can follow my hobby or passion, I don't know what to call it."

How does a usual day sound for you?

"I wake up at 06:45 a.m. I eat breakfast After that I go to school. I drive to school and back. Usually I spend 6/7 hours in school. When I get home, I do my homework or go to private lessons. After I finish my homework I like to work with my team."

What are your hobbies and passions aside programing?

"My main passion is basketball, a sport which I played for 8 years. Sometimes I like to play games online

like CS: GO or NBA with my friends or drive throug town."

What do your friends and family think about what you do?

"They always agreed with doing what I like, my friends on the other hand get mad sometimes that I can't play with them so much anymore or because I can't go out with them so much because of the workload."

Do they support you?

"Yes, of course."

What do you do in the team?

"I'm responsible for the mechatronics part. Me and Alex make a great team. I think that we work pretty efficiently."

Bleaja Sebastian

How do you manage your time?

“It would be an overstatement to say that I “manage” my time, I just try to be as productive as possible, sometimes I just procrastinate all day and get nothing done, all in all they balance out. I’m used to having a lot of activities at once and it sometimes feels just overwhelming but starting is always the hardest, it gets easier as you go on.”

Is it hard to learn for your finals and work for Herotech?

“I would say that it is indeed scary to have finals and important tests on top of extracurricular activities, that being said, I don’t think it’s going to be that hard keeping things in control. All in all, looking at my past performances I don’t need to worry a lot.”

How does a usual day sound for you?

“I usually wake up at 6 am on school days, 10 am on days in which I have something to do, and 12 am in rest days. I have my first meal after one or two hours after I wake up. I like to plan out my day in a way that lets me interact as much as I can with people, socializing energizes

me, if I stay indoors for more than two days I start to get really unmotivated and gloomy. I finish school at about 2 pm, it is also worth mentioning that my morning commute is done by foot even if I live far from school (about 4km). I don't usually go straight home after school, if I don't have any meetings to attend to or business to do, I go out with my friends.
I usually get home by 8 pm and dine with my family. School related work is usually done after 9 pm, I get that done at about 10 or 11 pm and watch some Netflix until I doze off."

What are your hobbies and passions?

"Until lately I loved representing people, all kinds of people, speaking in order to build a greater good. Now I think that I should focus more on myself. I was present in many social circles, from the student council to the county council, from volunteering to a festival to preparing to take the interview of a great pianist and one of my idols. I consider myself outgoing and creative, if nothing else I have my imagination.
I always enjoyed writing, but I've only recently started to do it a professional or competitive level. Writing is rewriting, exercising words, scrambling them, in such way that, at some point, after many hours, it will beget both clarity and style."

What do your friends and family think about what you do?

"I try to keep my friends out of this life, I'm not in any way embarrassed, I just feel like combining this with the fun of being carefree when I'm with them wouldn't be such a great idea. However, they are proud of me."

What do you do in the team?

"I do the writing, literally putting it in any other way wouldn't work, I write stuff. Also, I speak all the times in conversations that are held in English. I also oversee the book you are very reading, and I must say that I'm very much stressed about how it will turn out, if you are reading this then it did."

Sponsors

In autumn of 2019, after we decided to participate in the FTC, we soon realized that we cannot benefit from the grant offered by "Nație prin Educație" because there are two teams participating in the contest from our school. As such we had to buy the parts and kit from our own money. We started a fund raising campaign that succeeded because of the mass media influence, we would like to thank Izabela Păulescu in that matter. We had the support of the vice-mayor and the Sibiu county council president. We obtained the financial support from some companies from Sibiu, Brașov, Bucharest and Timișoara. The sponsors are Siemens, Continental Automotive, Speed & Trust, Toluna, Incorso Consult, Wenglor, Albis, Subway and Eurofoam.

We got support from the authorities too. We wrote a letter to the presidential administration bureau which redirected it to the Minister of research and innovation that put us in contact with National Institute of Research and Development for Mechatronics and Measuring Technique (INCD-MTM Bucharest) and Romanian Research & Development Institute for Gas Turbines. All our sponsorships were hard to get because we knocked on the door of every company and sent a lot of emails. We would like to thank the people at "Asociatia Maini Unite" and "Fundatia Comunitara" for the support with the paperwork and contracts.

Before Christmas the company ALDIS made us a surprise by sponsoring us yet again, by accident, in the end calling it "a gift from Santa".

A friend of ours suggested that we get in the "Ajut eu" campaign and we were selected to be backed up by them in the last week of November. We worked hard to make a robot presentable to the news and in the end, we managed to do it. Our little robot could take a cube and put it on top of another one. In the 29th of November we went to Bucharest where our team was presented to the public. A whole country had just seen us on the news. Thanks to the "Ajut eu" campaign that's supported by the "Mereu Aproape" foundation and the news source and channel "Antena 1" we managed to get some equipment, a 3d printer, a laptop, a gluing station, a drilling station and some desk equipment also most importantly parts for the robot that we are going to present in Japan at the WRS 2020. The robot for WRS 2020 needs to be able to help visually impaired people and communicate with the people in a home. We would also like to thank CSU Sibiu for letting us introduce ourselves to the basketball community by inviting us to one of the most important matches of this season. We entered the court with the teams and had a few minutes to present ourselves and our robot.

HEROTECH
powered by

wenglor
the innovative family

Toluna

INCDMTM

SIEMENS

INCORSO
CONSULT

DESTINATION
SPEED&TRUST
EXCELLENCE

SUBWAY

Continental

Eurofoam

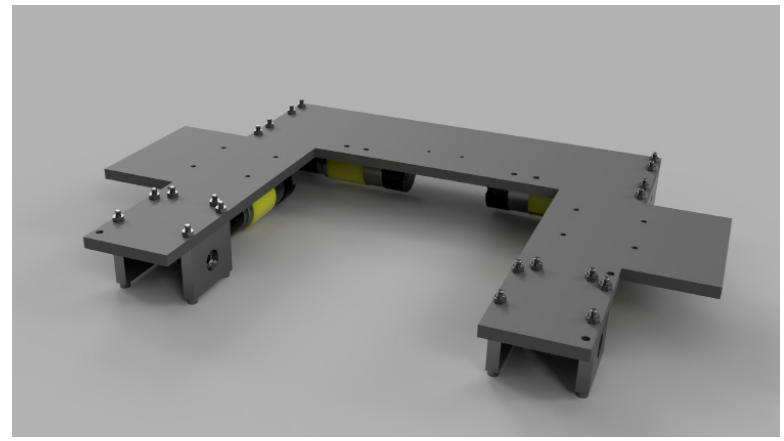
COMOTI
ROMANIAN RESEARCH & DEVELOPMENT INSTITUTE FOR GAS TURBINES

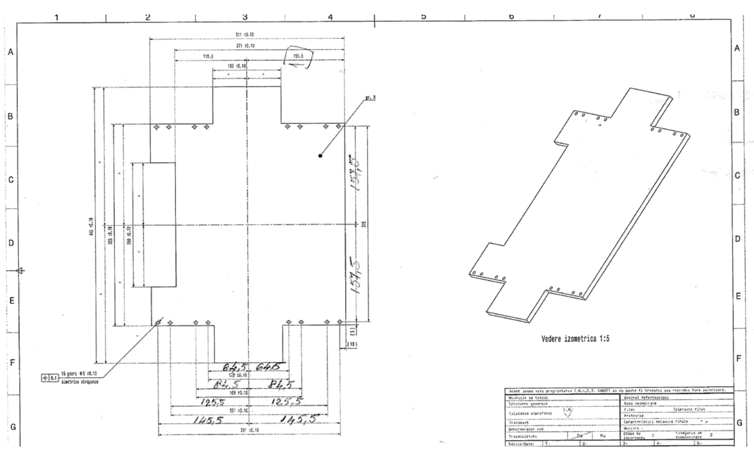
ALBIS

Technical systems

1. The robots base

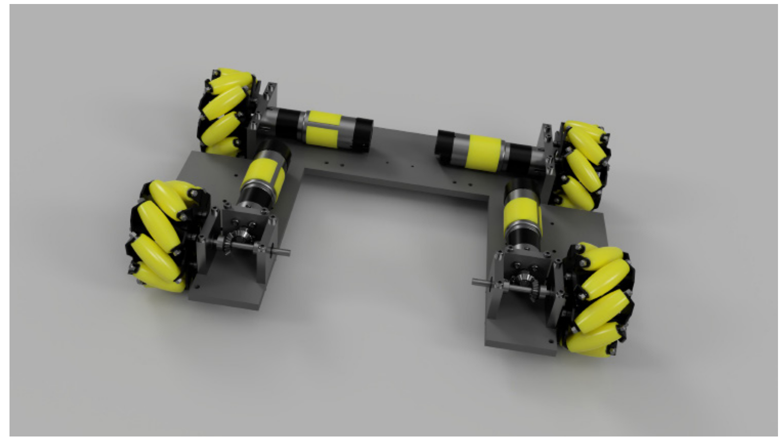
The robots base was projected by Alex so that it is rigid and meets all the requirements necessary for the subsequent construction that will be located on it. Therefore, we choose to make the base using 8 mm aluminum, size 445x310, making a U-shaped cutout in which a stone could easily enter. Also, the base has a cutout for each wheel, so that all the ensemble is as compact as possible. For the making of this we were helped by the Romanian Research and Development institute for Gas Turbines COMOTI Bucharest. They CNC cut it according to the technical details we sent them. 2.





2. Wheels and movement

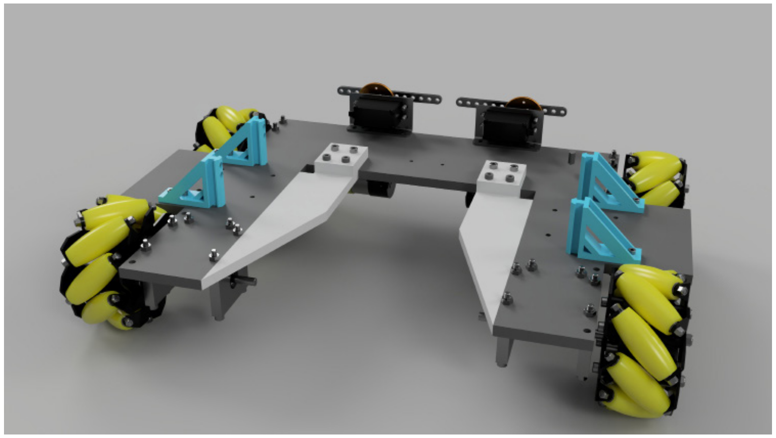
We opted for mecanum wheels which have the advantage of moving in any direction, the wheels are put into motion by four Gobilda 223 RPM motors, positioned under the robot's base. Each motor is equipped with an encoder so that we can precisely determine the robots on the field. For the wheels in the back we use direct coupling with the motor, while for the ones in the front we use brass bevel wheels with a 1:1 ratio. We choose this alternative because our intake system for the stones wouldn't allow us the positioning of the two motors in the front in a perpendicular position with the wheels.

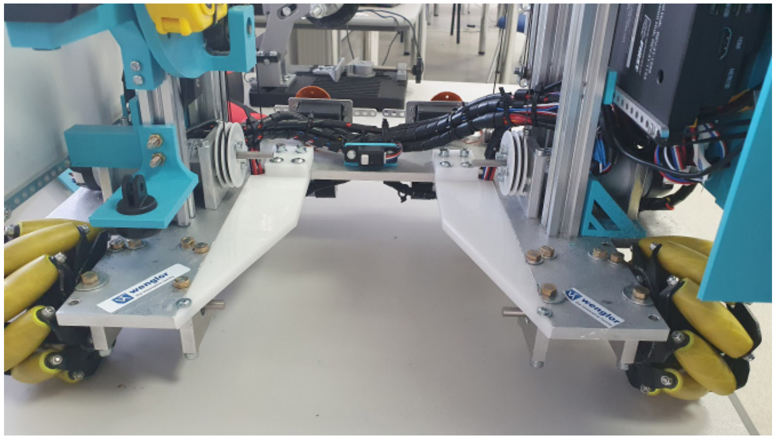


The motors are placed on the base using some rectangular aluminum pieces. As for the wheels in the front we used an axis fixed on two aluminum walls via a couple of bearings. On this axis we placed a bevel gear on each side which is then connected to another one from the engine's axis. Initially, because of some late arriving's, we used different RPM motors, the ones in the front having 117 RPM and the ones in the back 223 RPM. This forced us, for some weeks, to set all the motors at 117 RPM, which inevitably created problems in the training for the demos and the actual performance at the demos.

3. Intake system

We adopted a “not-so-complicated” stone intake system that proved itself highly efficient. To grip the game elements, we simply “go through” it with the robot, guiding it with some triangular pieces that allow the stone to enter in any position. The purpose of these pieces is to facilitate the entry of the game element in the U-shaped cutout. Initially we 3D printed these pieces, but they proved themselves to be too frail for competition use because when a game element fell on them, they would fall out from where we attached them on the base. Thus, we remade these pieces in Wenglor’s workshop and choose to make them out of polyoxymethylene. The two pieces from the guiding system held

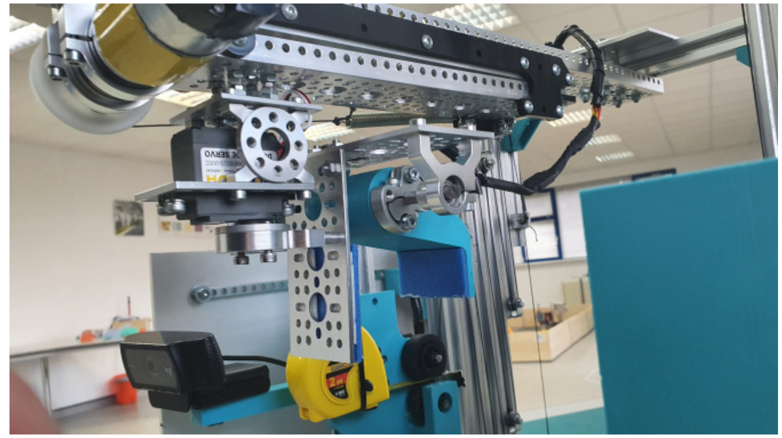


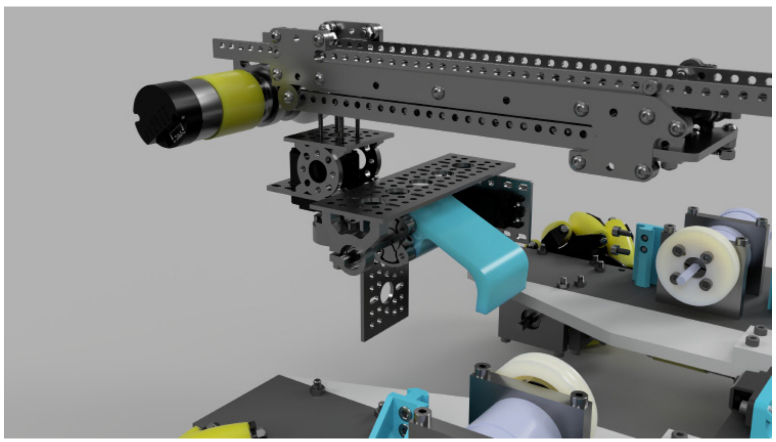


on to the base using two screws, each having a stopper that prevents buckling during the game. . At the end of the stone intake zone we mounted a touch sensor that signals if the game element is or is not introduced. Initially we foreseen the robot with an intake system based on rubber wheels which help the stones enter the U-shaped cutout. This system intake idea is advantageous because it facilitates the griping of the otherwise hardly accessible stones. Because we used the maximum of eight DC motors, we implemented this system using two servomotors connected to an ensemble of gears. In the end, we choose to only use the simple intake variant, without any auxiliary mechanisms.

4. Stone gripping

The stone gripping system was made multiple times because we were not pleased with the way this worked. Initially we realized a system that grips the stones from the back of the robot with a fixed part and a claw that grips the stone, the fixed part grips the stone from its back while the moving claw grasps it from the front. The claw is moved by two servomotors and the entirety of the system was caught on the slider which throughput the robot to move the cube front to back. This system proved itself to be inefficient and so we modified it to grip the body of the cube with a fixed part and to

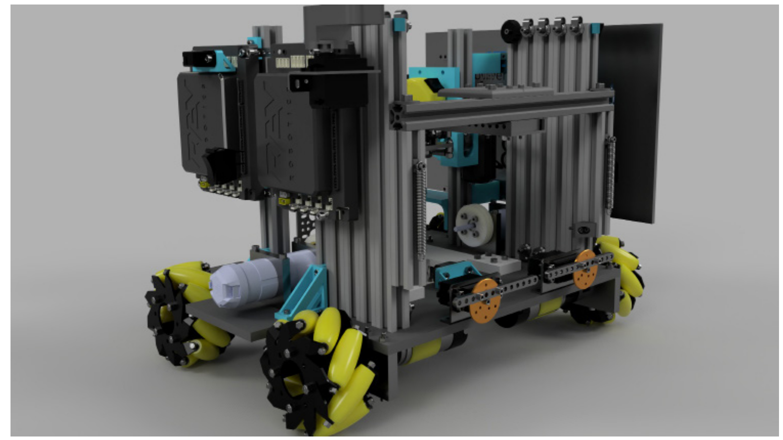




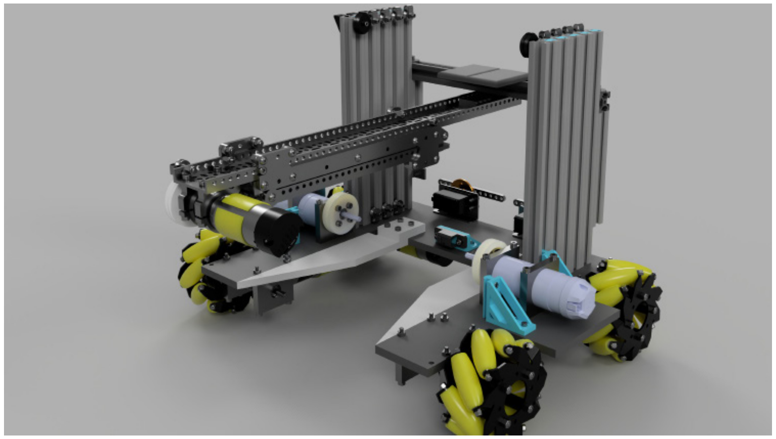
grip the interlocked top part of the cube with a mobile piece. The piece that grips the cube was projected in such a way that it can be mounted on a 6 mm axis moved by a servomotor. The claw was 3D printed and coated by a anti adherent material so that we were sure that the stone doesn't budge from its position. This servomotor system is connected to another servomotor that allows the movement of the cube in any position. This fact facilitates the capstones gripping and the stacking of the cubes in more than one way.

5. The lift system

To raise the game elements at the desired height we are using a sliding system made by using multiple 20 mm V slot extruded aluminum profiles. We choose to build this system by ourselves, as such, we projected some plastic insertions that were 3D printed that allow the aluminum profiles to slide. On each profile we mounted 3 insertions with screws, 2 for sliding and one that doesn't allow the unintentional sliding of the profiles lower than the desired level. For the mounting of the insertions in the interior of the V profiles we introduced a nut, heating it, so that the screw is secure. We repeated this process many times so that in the end we can opt for the best choice in which the distance between the aluminum profiles is the smallest. This diminishes the horizontal movement of the profiles. Initially we designed and built the robot with 2 rows of profiles in the back and two in the front, but this proved to be inefficient because of the friction. In the end we only used two rows of profiles positioned in the back of the robot. The first profile form each row is fixed on the base with screws, but also with some 3D printed parts that do not allow unwanted movement of the profiles. Moreover, for the complete elimination of friction related problems we used a white grease spray that proved to be essential. On each profile we mounted two V groove bearings that are fixed on with special nuts that can be inserted in the profile's interior. The sliding motion is realized using a highly resistant fishing string. To prevent it from falling off the bearings we mounted Gobilda defenders. The movement of the two lifts is realized by two REV motors that are fixed on the base with 2 aluminum parts designed by us. We choose to block the movement of these two pieces because the force they handle is quite big and they tend to move. Initially we 3D printed support pieces for them, but while being at Timisoara for the demo they cracked, as such we choose a much more resistant alternative. On each one of the motors axes there is a wheel on which the string is wrapped on. Although the two lifts can move up and down due to gravity, we choose to mount another string that pulls the whole system down. This is tied to the last profile and is connected to the same motor. We use encoders to determine the position of the game element at every moment.

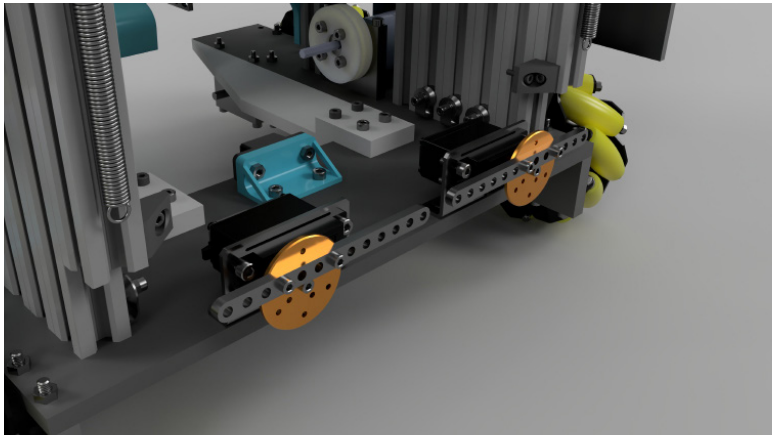


6. Slider



The horizontal slider was the system that caused us the most trouble, switching from one system to another until we found an optimal solution. Initially we conceived this system in a similar way to the lifting system, using the aluminum profiles and plastic insertions. For the bidirectional movement we used two REV Core Hex motors. This system was slow, unprecise and posed a lot of string related issues. Moreover, this was extending only forwards, thing that posed problems because we were losing a lot of time spinning around 180 degrees to position the cube on the foundation. We rethought the system so that the slider extends from the front to the back of the robot and built it using aluminum U channel from Gobilda that slides with the help of a couple of V groove bearings. Furthermore, we used a single motor for movement. This system had its problems too because we had to use a lot of bearings for a linear movement without experiencing level differences. In the end we opted for a much more efficient solution, the Goblida Cascading Kit. We used this system with only 2 aluminum profiles and a motor that moves the whole slider in both ways.

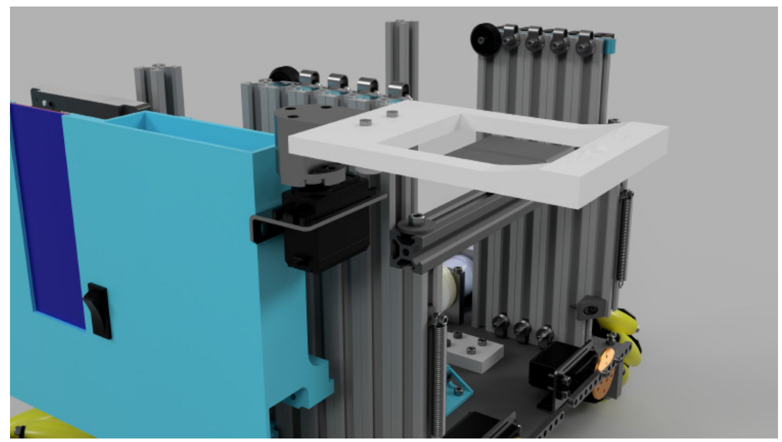
7. Foundation movement system



To move the foundation in the autonomous and controlled stages of the game we use two servomotors positioned on the back of our robot. We positioned them there to facilitate moving the foundation as fast as possible. We mounted these two servomotors using special aluminum holders that are fixed on the robot's base using four screws. To grip the foundation, we used two aluminum pieces from the Gobilda kit that we later cut at the desired measurements, all so that we could move the foundation as easily as possible. These pieces are coupled with the servomotors using the aluminum Servo Horn from REV. to pull the foundation we position the robot perpendicularly to it, driving the robot to its edge and then, using the servomotors that grip the foundation, we pull it by simply moving our robot.

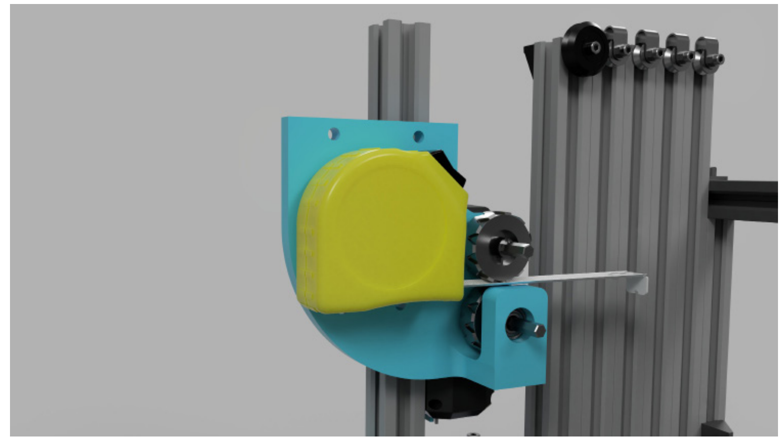
8. Capstone gripping system

Initially we 3D printed a capstone with the dimensions of a game element, so that we can effortlessly take it from the field. This was not efficient because we were losing a lot of time and couldn't stack that many cubes because its dimensions were too big. Therefore, we thought of a smaller capstone that can be included in the robot. On the last cube placed we positioned the capstone and our plan is to put them together on the stack. The stone is directly brought in the special place on the robot, that will be raised and transported to the back part of the robot with the help of the slider. Here, we are going to rotate the stone 90 degrees using the servomotor on the slider and raise it up. At the same time, from the corner of the robot, a servomotor-based system moves the capstone 180 degrees so that the robot can lift it up with the cube when we action the lift. We opted for the servomotor variant to not exceed the imposed size. The servomotor is attached to the battery holder using a special support. The capstone is rectangular, and it is 3D printed. These pieces present an interior hole that allows positioning in the designated area, the interlock area of the stone. On the servomotor we mounted a Servo Hub from Gobilda, on which we placed a 3D printed piece. These pieces present a hole in which the capstone is set using a plastic cylinder. To restrict the capstones movement when we rotate the servomotor, we designed two guiding holes in which the heads of the screws with which we fastened the piece fixed on the Servo Hub fit.



9. The tape measure parking system

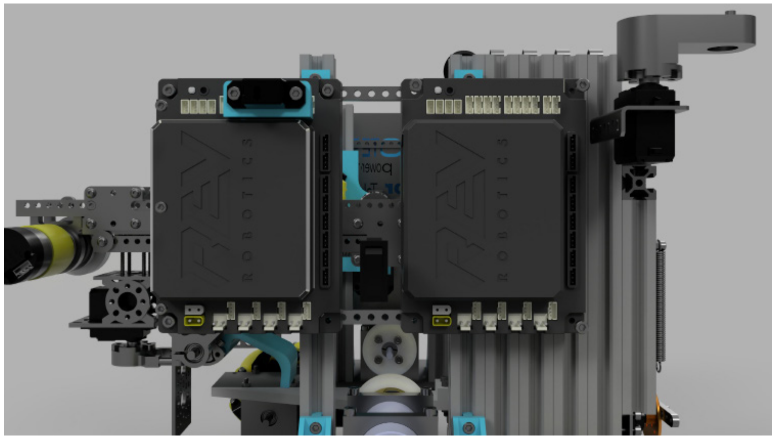
Although we didn't plan to use it at first, we choose to add this feature to our robot because we thought that it would be useful in the situation that we don't have time to park. The tape measure is mounted on a 3D printed piece designed by us. This piece is mounted on an aluminum profile. On the same profile we mounted a REV Core Hex motor that operates the tape measure. The extensible part of the tape measure goes through two rubber wheels. One of these



is tied to the motors axis while the other one is freely rotating on an ax fixed with two bearings. The tape measure extends backwards because at the end of the match, when we need to park, the robot is positioned with its back headed to the parking area.

10. Electronics

To control all the electric part of the robot we use a REV expansion hub and a REV control hub that communicate between each other by a serial cable. To the expansion hub are connected: two REV motors with encoders for the lift, the Gobilda motor with his encoder for the horizontal slider and the REV motor that operates the tape measure. Also connected here are two Gobilda servomotors from the stone gripping system. In the autonomous period we use a color sensor that detects the alliance of which we belong so that we can run the corresponding program. This color sensor is positioned near the expansion hub. The sensor connects to the hub through the I2C port. The touch sensor detects if the stone entered the robots U-cutout and is also mounted on the expansion hub at one of the analogue ports. The control hub operates four motors with their respective encoders used to move the chassis. There are also two servomotors that move the foundation



connected to the control hub and the servomotor used to move the capstone. The battery is mounted on the robot so that we can easily replace it. The battery cable is connected to a switch that turns off the robot in case of an emergency. From the switch the cable enters the expansion hub from which a power cable is connected to the control hub. We mounted a switch on this cable too in the event of needing to resort to the separate supply of power to the two controllers. During a match from Timisoara a robot from the other alliance bumped unintentionally into our robot, destroying one of the USB ports from the control hub. Thus, we were forced to use a USB Hub in which a serial cable that communicates with the expansion hub and the cable for the Logitech c920 camera used to detect the skystones in the autonomous timeframe. We choose this camera for its performance this camera is placed on the frontal side of the robot so that it has a large field of view, being mounted on an aluminum profile using a 3D printed support. Also, on this support we mounted a holder for the GoPro camera used to film the game. Each cable is covered in a protection spiral to avoid shearing and destroying them. The cable for the lift that extends vertically represented a big issue because they tend to cling to other components. For an efficient cable management, we choose to mount some plastic support pieces on the exterior side of the slider. We mounted all the vertically moving cables on these supports. All the cables were extended using other cables of the same dimensions, being welded and isolated using a heat shrinkable tube to assure their resistance and protection. To prevent the deterioration of the control hub and the expansion hub and to assure that all the cables are well secured in their ports we covered all that part with two 3D printed pieces. The emergency switch and the alliance

Hardware encountered problems

• The bevel gears that we ordered from Amazon didn't arrive on time from China so we were forced to find somebody that could make them for us in a short period of time to assure that we can attend the Timisoara demo.

• The order from Gobilda didn't arrive on time

• Our game area is improvised, as such our foundation can't be used, and the game elements are heavier.

• Initially the base wasn't made with a U-cutout in which the cube should enter so we had to find a company that could help us with that.

• In Timisoara at the demo our 3D printed supports for the lift cracked, so we had to remake them from aluminum.

• The guiding system that we printed broke, so we had to make it using a harder material.

• The encoders aren't exact.

• A motor heated up a lot during the game.

• At the demo in Timisoara a robot bumped into us and broke a USB port from the control hub.

• The cables that sheared snapped.

• The motors axis is short thus the wheel is poorly fixed.

• One of the axes of the motors isn't fixed in place so it influences the movement.

• The initial design was inefficient, the friction was too high, and we had two rows of sliders in the front and in the back of the robot.

• Initially we didn't have a system that permitted the downwards movement and we placed a weight on the slider to make it easier.

• We didn't have any protections for the bearings, so the string was falling frequently.

• The lift cables were initially mounted on the interior so the slider was moving a stone, the stone would bump into them.

• The initial gripping system for the capstone wasn't efficient, a part of the capstone was breaking off during the movement.

• Our first capstone couldn't be easily lifted as it was big and heavy.

• The mecanum wheels weren't properly mounted, so the movement of the robot wasn't precise.

7.1.1. An overview of the program

The tele-op is controlled by 2 drivers. One driver is controlling the chassis and the other one is controlling the stone-stacking system.
The whole tele-op program is made in a single file due to lack of complexity. After initializing the robot, we set the positions of all the servos to default for making sure they are in position for fulfilling the tasks. The program extends the LinearOpMode, so all the logic happens inside the runOpMode function. After pressing start, the program flow enters a while loop that executes until pressing stop. In this loop we check for input from both the gamepads and then move the motors accordingly to the input. We automatized features like rising the lift to the desired level by the number of stones in the skyscraper and extending the arm so that we can position the stone on the foundation if we are lined up with the foundation, all at the press of a button.

7.1.2. Input

For getting the input we had a problem with multiple registrations of a button press, because the while loop executes way faster than we are capable of pressing and releasing a button. For this problem we came with the solution of creating a class inside our DriverControl program called ButtonToggle. This class together with an enum called ButtonStatus, helps us determine when a button was pressed and then released, therefore making the action that should execute happen only once. The ButtonStatus enum contains three elements: NOT_BEGUN, IN_PROGRESS and COMPLETE. These elements describe the state of a button press. The class has a Status method that takes a boolean as a parameter that will be true – if the button is pressed and false contrary. The function returns an object of type ButtonStatus. For each button that we use on our controller we declared a variable of type ButtonToggle on which we call the Status function exactly once per loop execution. By doing this, the function can compare the current parameter with the parameter of the call before. The table below clarifies the return value of the function based on the parameter of this call and the previous one.

Previous call parameter	Current call parameter	Return value of current call
False	False	NOT_BEGUN
False	True	IN_PROGRESS
True	True	IN_PROGRESS
True	False	COMPLETE

(1)- X axis for rotate and Y axis	is for chassis movement
(2)- X axis is for chassis strafe	(A)- SLOW MODE
(Y)- Foundation Servo Grab	(B)- Foundation Servo Release

(1)- Tape measure	(Y)- Arm servo2
(2)- Manual lift movement	(A)- Cube Grab
(X)- Lift state switch	(B)- Arm servo2

(1)- Increase Skyscraper level	(2)- Decrease Skyscraper level

(1)- Auto Arm Retraction	(3)- Arm Retraction
(2)- Auto Arm Extension	(4)- Arm Extension

7.1.3 Chassis Movement

Chassis movement is done by the PerformChassisMovement method. Basically, the program is reading the gamepad input from the left stick X, Y axis and right stick X axis.

Each input variable has a range between -1 and 1.

We are calculating the power that we should apply for each motor by doing a set of calculations different for each variable

The Y axis joystick represents the forward and backward movement, by adding Y to each motor power, the movement of each motor will be in the same direction.

The X axis joystick is responsible for the robot rotation movement. We apply this value, positive to the left side motors and negative to the right-side motor. The result is similar to the tank steering system.

The second X axis joystick is applying this value to the motors in diagonal and the opposite value for the other two motors. By doing this, we permit the mecanum wheels to work their magic by making the robot strafe.

```
private void PerformChassisMovement(){
    double X = gamepad1.left_stick_x / 1.5;
    double Y = -gamepad1.left_stick_y;
    double Y2 = gamepad1.right_stick_x;

    double FrontLeft  = Y + X + X2;
    double RearLeft   = Y + X - X2;
    double FrontRight = Y - X - X2;
    double RearRight  = Y - X + X2;

    front_right.setPower(FrontRight*speedModifier);
    front_left.setPower(FrontLeft*speedModifier);
    back_right.setPower(RearRight*speedModifier);
    back_left.setPower(RearLeft*speedModifier);
}
```

7.1.4 Servo movements

For the rotation of the servos we use a method called PerformServoMovement. Firstly, this method checks for input from the gamepads using ButtonToggle class.

Because the servos are fixed on different sides of the robot the drivers cannot see them every time. That was a problem that forced us to assign a button for each servo rotation so the drivers know even if they cannot see the servo that the it has rotated to the desired position.

We use servos for the foundation-move system, for the stone place system we have 2 servos one for the mechanism that grabs the stone and one to turn the stone to 90 degrees (for safer stacking and for the capstone). The capstone servo is moving the capstone from the initial position to the rear of the robot whre it will be mounted on the stone directly from the robot.

7.1.5 Lift movement method

The lift movement is theoretical based on 2 different movements. The default one is the autonomous movement and the second one is the driver-controlled movement.

The autonomous movement helps the stacking driver by setting the lift level to the real stacking level just with a button press. The whole operation is controlled by the AutoSliderMovement method. Basically, we use a variable that saves the current stacked stones + 1. That represents the level our lift needs to go to stack the stone. When the driver presses the DPAD-UP button the lift will go to that level and the arm will move forward putting the stone in the right position above the stacked stones. The driver needs to set manually the lift position to stack the stone. When the stone is in perfect position the driver will release the stone using the grab servo. After the release has been done the arm will automatically retract and the lift will be set to the driver selected position(*) by pressing DPAD-DOWN.

(*) – The stacking driver has 3 default options for the lift state movement. The first option is CUBE_GRAB that represents the level the lift needs to be to perfectly grab the stone. The second option is CUBE_TRANSPORT that represents the level the lift needs to be to stop the stone from rubbing with the soft tiles. The robot can move under the skybridge even the lift level is set to CUBE_TRANSPORT. The third option is CUBE_ENTER that represents the perfectly lift level to take the stone in the robot.

The lift motors are actioned using their built-in magnetic encoders. Because we had some problems with the RUN TO POSITION motor mode, we wrote a method that simulates that mode using the encoders but it is more efficient. The method is constantly reading the encoder value and it calculates the speed for each motor. Theoretical the speed is proportional to the remaining distance but in some cases like the lift-up movement the power is 100% because by our calculations the error is minimal, and we gain some precious time by doing that.

Each stacked stone level has a corresponding encoder value. The first value is 1400 and it's increasing by 1000 ticks per level. The first value needs to be a little bit higher because we cannot move the arm higher below 1300 lift motor ticks. That would seriously damage our robot and it's code protected.

2.Autonomous

7.2.1 Our first try

The first time when we tried doing an autonomous program, we wrote some functions for moving different parts of the robot that took the number of ticks and the speed at which the specific motor should rotate. When we tested a simple combination of these function, so that the robot would move forward to a stone and take it, we realized there are quite a few problems to our attempt. Firstly, we realized that giving a constant speed to a motor until it reaches the target destination would not be accurate because of overshooting. This was mostly visible on chassis movement because of the mecanum wheels. For example, on a 90 degrees rotation, even if the motors stop when the robot reaches 90 degrees, they will still slide because of inertia, making the robot turn way more than planned. The second problem was that we could not do two things at once. Our functions for moving the robot contained a while loop, that would finish executing just when the specific part reached its destination. We realized this was not a good approach because our robot should be able to do more things at once (e.g. moving forward while rising the lift)

7.2.2 Researching and Solutions

After reading on the internet about robots programming and checking forums to see if others had the same problems, we found solutions to the problems described earlier.

7.2.2.1 PID Controller

PID Control stands for Proportional-Integral-Derivative feedback control. The idea behind this concept is that it takes the desired value and it compares it with the current value (called feedback) and deducts an output value. As an example, we used PID Control on rotating the robot. The desired value is the number of degrees we want to rotate. The feedback is the current gyroscope angle and the output value is the power we should apply to chassis motors. We tuned the scalars Ki and Kp values so that our robot starts rotating at higher speed and after that, decreasing this value, so that the slowing rate of a turn will eliminate the overshoot.

7.2.2.2 Multitasking

We mean multitasking not in running different code on different threads but doing multiple things at the same time without being stuck in a while loop. Without using this system, we would have to wait for one action to finish before starting the next one. We designed an interface that will be extended by each class that is responsible with moving motors on our robot. This interface called HTAction has 1 public field: a boolean called Finished, and two public void methods named Setup and Execute. So how does this work? It is easier to explain on an example:

```
HTAction moveForward = new MoveForward(20);
HTAction raiseLift = new MoveLift(LiftState.ENCODER_ZERO);
HTAction rotateChassis = new RotateRobot(90);

int state = 1;

waitForStart();

moveForward.Setup();
raiseLift.Setup();

while(opModeIsActive() && !isStopRequired())
{
    switch(state):
    {
        case 1:
        {
            moveForward.Execute();
            raiseLift.Execute();
            if(moveForward.Finished && raiseLift.Finished)
            {
                state = 2;
                rotateChassis.Setup();
            }
        }
        case 2:
        {
            rotateChassis.Execute();
            if(rotateChassis.Finished)
            {
                state = 3;
            }
        }

    }
}
```

We declared 3 variables of type HTAction and an integer that represents the current state of the program. After initializing the robot, we call the Setup function on the first actions that we want to execute simultaneously. In the main loop we have a switch case for the value of the state variable. So firstly, it will fall into case 1. There it will call Execute on both variables and the condition of the if statement will be true just when both the actions are Finished. After that we set the state to be 2 and call Setup on the HTAction that we want to execute for state 2.

An example of implementing the interface on a class:

```
public void Execute()
{
    CheckIfFinished();
    if(Finished)
        return;

    double power = CalculatePower(maxSpeed);

    Orientation currentAngle = HardwareConfiguration.imu.getAngularOrientation(AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES);

    double degreesOff;
    if(!useAngleZero)
    {
        degreesOff = Math.abs(currentAngle.firstAngle - (startAngle.firstAngle));
    }
    else
    {
        degreesOff = Math.abs(currentAngle.firstAngle);
    }

    double correction = 0;
    if(degreesOff > 0.2)
    {
        correction = (Math.pow((degreesOff + 2) / 5, 2) + 2) / 100;
    }

    if(currentAngle.firstAngle != startAngle.firstAngle)
    {
        if(currentAngle.firstAngle>startAngle.firstAngle)
        {
            correction = correction;
        }
        else
        {
            correction = -correction;
        }
    }
    HardwareConfiguration.front_right.setPower(power-correction);
    HardwareConfiguration.front_left.setPower(power+correction);
    HardwareConfiguration.back_right.setPower(power-correction);
    HardwareConfiguration.back_left.setPower(power+correction);
```

```
public void Setup()
{
    startAngle = HardwareConfiguration.imu.getAngularOrientation(AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES);
    HardwareConfiguration.front_right.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    HardwareConfiguration.front_left.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    HardwareConfiguration.back_right.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
    HardwareConfiguration.back_left.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);

    HardwareConfiguration.front_right.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
    HardwareConfiguration.front_left.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
    HardwareConfiguration.back_right.setMode(DcMotor.RunMode.RUN_USING_ENCODER);
    HardwareConfiguration.back_left.setMode(DcMotor.RunMode.RUN_USING_ENCODER);

    HardwareConfiguration.front_right.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
    HardwareConfiguration.front_left.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
    HardwareConfiguration.back_right.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
    HardwareConfiguration.back_left.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);

    trTargetTicks = ticksToMove;
}
```

What we want to do next, to further improve this system is declare a class that should handle all the execution of the actions on their corresponding states, so that we improve code readability of our autonomous programs.

7.2.3 Hardware Configuration Class

Because we have many classes that interact with our hardware components, we decided that it is a better idea to have a class that is going to manage them, for easier accessibility. Before coming to this realization, we read our hardwareMap object inside the Autonomous program class. The problem with this approach was that we had to pass the hardwareMap object to each constructor of classes that would interact with the hardware, which was slightly inconvenient. Here intervenes the HardwareConfiguration class. It is declared public and all its members are static, so we can interact with them without getting a reference of it. It consists of variables for all our hardware components, a Setup function, which takes as parameter the hardwareMap and gets the references to our hardware. It also has a Reset method because, if the class is declared static, the fields won't be automatically managed by the garbage collector and their values will carry on with the next execution of a program (after pressing the stop and start button), so in this method we just set them to null.

7.2.4 Detecting skystones

For detecting skystones we have installed a camera on the robot that can be repositioned so that the 3 stones facing the bridge are in its field of view. We tested the Vuforia code examples from the FTC repository, but we weren't satisfied with the results because they were not accurate. Two or three stones were sometimes grouped and recognized as a single one, so we weren't able to determine the position of the skystone. We ended up finding a library called EasyOpenCV that takes the video input of the camera, makes it in grayscale, converts it to negative and draws 3 rectangles at a specific position. We aligned or camera so that in the initial position of the robot, each rectangle covers about ¾ out of a stone. Then, a custom class checks for the medium color on each of those rectangles. The skystone is going to be the rectangle colored white.

7.2.5 Touch sensor

In autonomous, it is very important to know when a stone is in position to be grabbed, because it won't always fit perfectly. By using touch sensor, after going forward to pick a stone, we can confirm is the stone is pressing the sensor button, if it is, then we can safely go on with our program. Moving forward and checking for input at the same time is possible using our multitasking system described at (7.2.2.2)

7.2.6 Angle correction

While moving the chassis with a constant speed given to each motor, we realized that the robot is not going in a straight line and its final position can be off with about 2 degrees. This may seem like a small error, but it accumulates for all the movements it does in autonomous. The solution we came with was calculating the offset and adding a correction to the motor powers so that the robot will get back on track if it happens to deviate. Each time we calculate the power we will give to the motors, we first determine the offset value, which is the current angle minus the start angle. If this angle is less than 0.2 degrees, we don't want to apply correction. Applying a bigger value for correction, can make the robot move jerky or oscillate too much. We played with this value, until we got something that is working: correction = (((degreesOff + 2) / 5)^2 + 2) / 100. We use a square function because if the integer is smaller than 3 degrees, squaring it will output a value lower than 1 and contrary, it will output a value bigger than 1. After that we add 2 and divide by 100 so we get a smaller value better suited for a

7.2.7 Color sensor

For the autonomous program we are using a color sensor installed just in the back of our alliance marker. The color sensor is reading the color of the back the alliance marker that represents the opposite value of the current alliance color. Basically, if the sensor reads BLUE, we are RED and opposite. During the autonomous initialization the color sensor starts reading and we determine the alliance color. There are 2 options RED or BLUE. Based on the value of the alliance color variable the robot must pick the right autonomous program to run. If the color is RED the BLUE autonomous will run and if the color is BLUE the RED autonomous program will run. The only difference between the BLUE and RED autonomous is that all strafes movements are turned opposite. This was only a test to see how much automatization can we include in our program.

8. Bill of materials

ITEM	QTY	PRICE/PIECE ($)	TOTAL PRICE ($)
Aluminum chassis	1	donation from COMOTI Bucharest	
Aluminum motor holders (for chassis)	4	donation from COMOTI Bucharest	
Aluminum motor holders (for lifts)	4	donation from Wenglor Romania	
Aluminum shaft holders (for bevel gears shaft holders)	4	donation from COMOTI Bucharest	
POM intake parts	2	donation from Wenglor Romania	
Aluminum slider holder	2	donation from Wenglor Romania	
Mecanum wheels set - 4	1	139	139
223 RPM Gobilda motor for chassis	4	40	160
Core Hex Motor REV	1	21	21
HD Hex Motor REV	2	28	56
REV Control HUB	1	300	300
REV Expansion HUB	1	175	175
312 RPM Gobilda Motor	1	40	40
Gobilda Cascading kit	1	100	100
Gobilda 2000 Series Dual Mode Servo	5	28	140
REV Touch Sensor	1	6	6
REV color sensor	12	14	14

V slot aluminum profile	12	2	24
Fishing line (50 meters)	1	40	40
V groove bearings	22	0.9	19.8
1601 Series Flanged Ball Bearing	2	3	6
V-Groove Bearing Shield	13	2.5	32.5
Gobilda 1309 Series Sonic Hub	4	6	24
Gobilda 3407 Series Hub Mount Winch Pul-ley	3	5	15
Gobilda 1908 Series Servo Hub	3	3.5	10.5
REV aluminum servo horn	2	4	8
REV 12 V sllim battery	2	50	100
REV battery charger	1	32	32
REV 15MM METAL BENT SERVO BRACKET V2 - 4 PACK	1	5	5
Gobilda 1801 Series Servo Plate	2	3	6
Spring	2	1	2
Gobilda 1106 Series Square Beam	2	2	4
Gobilda 1102 Series Flat Beam	1	4.2	4.2
Gobilda 1204 Series Gusseted Angle Mount	1	5	5
Gobilda 2101 Series Stainless Steel D-Shaft (6mm Diameter, 100mm Length)	2	2.4	4.8
Bevel gear	4	40	160
Gobilda 1200 Series 1-Side, 2-Post Pat-tern Mount	2	4	8

2906 Series Aluminum Set Screw Collar (6mm Bore, 8mm Length)	2	2.5	5
1123 Series Pattern Plate	1	2.5	2.5
Measurement tape	1	3	3
Screws, washers, nuts	1	50	50
Antiadherent material	1	5	5
White grease	1	7	7
PETG filament spool from PRUSA	4	25	100
Stickers	1	15	15
PVC foam board	1	5	5
Logitech c920 camera	1	80	80
Wires	1	100	100
protection for wires	1	15	15
Switch	2	6	12
			TOTAL 2061.3

9. Events

9.1 Our trip to Timisoara

On the 18th of January, our team took a trip to Timisoara to participate in our first demo.

We left Sibiu on Friday morning at 8.30 am after getting a counterweight for our robot. The road took about 5 hours, after which we arrived in Timisoara and went to the mall to grab a coffee. A day earlier most of us worked until 2 am to make sure the robot will work properly. After our check in we spent time together as a team. As the only writer for this book, having no experience in robots whatsoever or coding, I think that the purpose of this contest is not only making something revolutionary, something that will change the world, it's the moments that bond us, that shows what's like having a second family, its these moments that help some of us to feel accepted, after all bright minds can be sad too, right? Saturday in the morning We went to "Universitatea de Vest din Timisoara: "where the competition took place. We prepared our robot for the technical inspection then we played some friendly matches with other teams. While playing we noticed that we had some technical problems so we tried to solve them. Also, Radu and Codrin worked on the autonomous part of the robot. They managed to write the code in a very short time so we were capable to park the robot and to move the foundation.

9.2 Sibiu Demo

On the 14th of February we participated at the Valentines Robotics Games. The event was organized by GearManiacsTeam and we played some friendly matches with them and BytesTeam. We told the audience about the FTC and about the contest in which we participate in and we also prepared for the regionals. We had a lot of fun and met a lot of new and interesting people.

9.3 CSU Sibiu

We had the pleasure to be invited by CSU Sibiu at the basketball match between them and UVT Cluj, where they let us present ourselves before the match. We controlled our robot in the game field, where every spectator could see it. Even if the robot wasn't in its final state, we were happy for such an opportunity. Needless to say, it was a good opportunity to find new sponsors for our team.

Control Award, Sponsored by Arm, Inc. Submission Form

Updated 10.21.2019

****Please turn in this sheet during your Judge Interview along with your Engineering Notebook****

Team # RO 180	**Team Name:** Herotech

Autonomous objectives:

Our first autonomous objectives is detecing and delivering 2 skystone after the bridge. If the skystone is near the plexi glass wall, we can't easily take it and we will go for 1 skystone and 1 stone. We will keep the last stone in our robot, so that we can easily place it on the foundation in teleop mode. After the delivery we park under the bridge.

Sensors used:

During the autonomous period we are using the Control Hub built-in gyroscope to correct all the chassis movements (7.2.6). To detect the skystone position we use a Logitech C920 camera (7.2.4), and for knowing when a stone is in position for grabbing we use a touch sensor (7.2.5). A color sensor is used to determine the alliance color (7.2.7)

Key algorithms:

During the autonomous period the chassis movements (Stafing and moving forward/backward) are corrected using an algorithm that takes the offset degrees as input and returns a correction value (7.2.6)

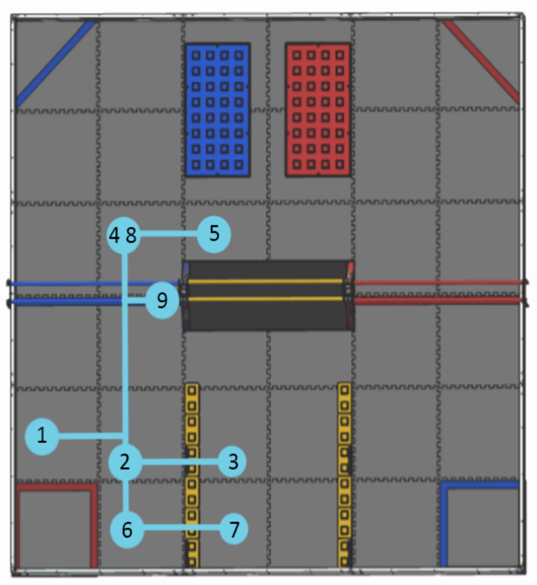
Driver controlled enhancements:

In the driver controlled period the lift system is working in the auto mode. The lift is moving to the right stacking position just at the pressing of one button (7.1.5)

Engineering notebook references:

You can find all the detailed information you need about the programming in the 7th chapter from our notebook.

Autonomous program diagrams:



Our CAD: