TEKNOFEST

AEROSPACE AND TECHNOLOGY FESTIVAL

DIGITAL TECHNOLOGIES IN INDUSTRY COMPETITION

PRE-DESIGN REPORT

TEAM NAME

Doha BOULOUHA
Omar SAHRAOUI
Nabil ABDELOUALI
Mohamed QERRAS
Aymane LAKHAL
Hamza EL FAHIM
Hicham LAACHA

PROJECT NAME

Digital Technologies Competition in Industry

APPLICATION ID

465677

table of contents

1. REPORT SUMMARY

Smart manufacturing is being driven by the advent and maturation of many technologies, including high-performance computing (HPC)-powered computer aided design (CAD) and engineering (CAE) software; cloud computing; the Internet of Things; advanced sensor technologies; 3D printing; industrial robotics; data analytics; machine learning; and wireless connectivity that better enables machine-to-machine (M2M) communications. Amongst the most important of these are the marriage of sensors and software into the Internet of Things (IoT).

Autonomous mobile robots or AMRs, just like humans, also have the ability to make their own decisions and then perform an action accordingly. A truly autonomous robot is one that can perceive its environment, make decisions based on what it perceives and/or has been programmed to recognize conditions and then actuate a movement or manipulation within that environment. With respect to robot mobility, for example, these decision-based actions include but are not limited to some basic tasks like starting, stopping, and maneuvering around obstacles that are in their way.

Technologies such as "artificial intelligence, autonomous robots, big data and advanced analytics, cloud computing, augmented and virtual reality, internet of things, additive manufacturing, next-generation smart sensor technologies and cyber security"

The robot must be 900 mm wide and 1000 mm long with a 500 mm height, able to lift Loads of 30cm x 30cm x 10cm dimensions and 25 kg (load weight). For automatic loading and discharge, the loads will be positioned on platforms, The weight of the cargo platform (defined with RFID tags) will be 25 kg. Along with the loads the bot should be finally able to lift up to 100 kg in total. The task is that there will be 4 loading and discharge points at points A, B, C and D on the track. It is necessary to leave the cargo platforms at the marked places while leaving them as they were taken from the same place. The guided robot should be able to go to the lading and discharge areas in accordance with the given directives. Necessary hurdles will be placed as well, in which case the robot should slow down, and if the hurdle is blocking the path, it should stop without crashing. While waiting, the robot should give a warning by playing a warning sound.

So in order to accomplish these tasks the machine should be equipped with the adequate type of wheels, encoders, motors that will provide the enough power to lift the heavy weight as well as assuring the movements, a battery that can accomplish the whole task supported by an ac/dc charging dock, and finally in order for it to be fully automated we need to equip multiple sensors, For the AMRs to navigate so well, they need to have highly advanced sensors installed into their unit. Different environmental sensors can be used, but most commonly, laser scanners are used or LiDAR sensors. LiDAR scanners work by emitting pulses of light that reach different surfaces, and they bounce the light back to the sensor where it is received again.

Another method to navigate is the updated technology of 3D cameras. they work in a similar fashion to the stereo cameras, by measuring the pixel distance of two images of the same scene. In addition to other sensors responsible for Object Detection, self-localization, mapping, how far it has traveled. etc.

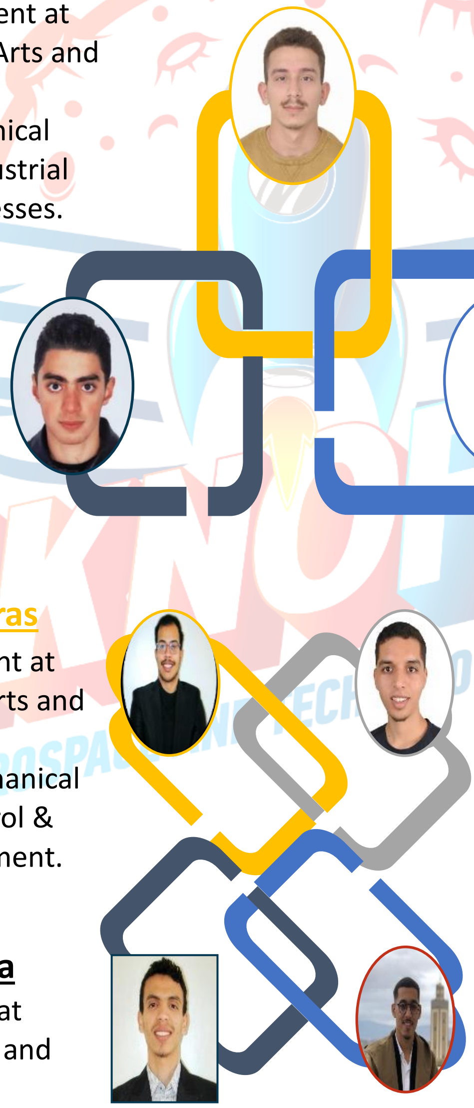
Automation is not a straightforward process. It requires significant planning and a deep understanding of your processes. From hardware to software, from functionality to operation, AMR is just more than navigation algorithms.

2. Team chart

2.1. Team Members

The team is made up of five young engineering students from the National School of Arts and Crafts in Meknes. Dynamic and ambitious, endowed with a versatile education, each committed to a different specialty (major), all determined to master our technical knowledge and develop

Aymane Lakhal
Captain
Engineering student at National School of Arts and Crafts
Branch: Mechanical engineering: industrial fabrication processes.



El fahim hamza
Engineering student at National School of Arts and Crafts
Branch: Mechanical engineering: Structures and product engineering.

Doha Boulouha
Engineering student at National School of Arts and Crafts
Branch: Mechanical engineering: Structures and product engineering.

Mohamed Qerras
Engineering student at National School of Arts and Crafts
Branch: Electromechanical Engineering: control & industrial management.

Nabil abdelouali
Engineering student at National School of Arts and Crafts
Branch: Industrial enginee ring: artificial Intelligence and Data science.

Hicham Laacha
Engineering student at National School of Arts and Crafts
Branch: Electromechanical Engineering: control & industrial management.

Omar Sahraoui
Engineering student at National School of Arts and Crafts
Branch: Electromechanical Engineering: control & industrial management.

our sense of innovation in the field of robotics.

Mechanical design

Catia V5

AutoCad

SolidWorks

Simulation tools

Matlab

Ansys

Programming languages

Java

C

Python

… courses

Automation of systems

Signal processing

Electrical machinery

Related to energies

Turbomachinery

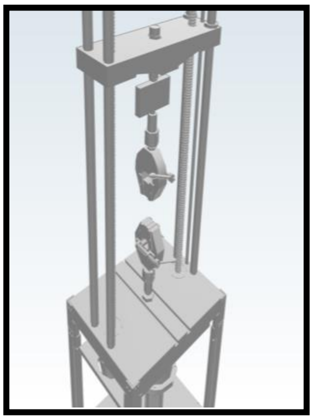
Thermal machines

Heat transfer

Completed Projects

- ☑ Bibliographic study of the mechanical system.
- ☑ Functional analysis and sizing of the various transmission elements.
- ☑ Simulation on CATIA V5 and graphic representation of the whole system

Design of a tensile testing machine



Conception of a concrete mixer

Design of a wind turbine

Interpersonal Skills

- Teamwork
- Ability to adapt
- Flexibility & Motivation
- Problem solving & Decision
- Organization & Rigor
- Conflict resolution &
- Creativity and sense of
- Sense of responsibility

2.2. Organization Chart and Task Distribution

Aymane Lakhal	Hamza EL FAHIM	Nabil ABDELOUALI
Team Organization and Analysis of Competition Rules and Design Study Objectives	Design of the Hardware part of the robot on Catia V5	Choice of materials and study of manufacturing processes

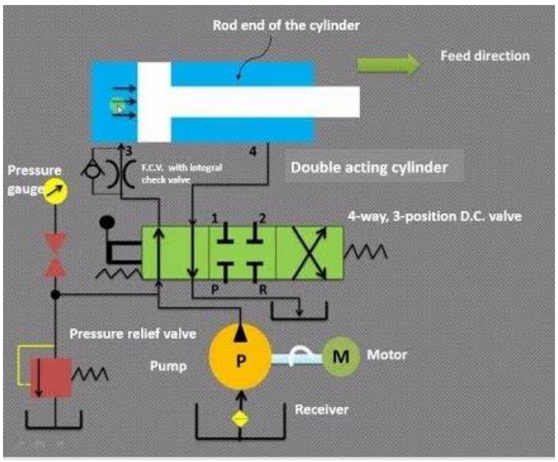
Doha BOULOUHA	Omar SAHRAOUI	Hicham LAACHA	Mohamed QERRAS
Preliminary design of the robot and choice of solutions	Design of the electronic part and development of algorithms	Design of the ele\ctronic part and development of algorithms	Design of the electronic part and development of algorithms

3. PROJECT CURRENT STATUS ASSESSMENT

This part is dedicated to talk about the myriad solutions that we took into consideration in our conception of the industrial robot. Having to take the load from a point A to a point B as the main function of our robot, we came up with a lot of ways to adapt our solution to your demands

such reducing its weight throw, optimizing its chassis geometry, the materials and the clutter between the different mechanical components, so that the robot's speed is at its maximum.
As mentioned in the preliminary report, we used a hydraulic cylinder as a way to lift the charge, this solution is working in a multitude of industrial projects, classical lifting systems or even a simple construction machines; this is a solution that permitted to lift a significant load.


But as we tried to apply this solution in our case, we found out that the hydraulic circuit for the cylinder is taking a lot of space. As seen in the picture bellow, for a cylinder to be useable, a whole bunch of other components is needed, we can mention: a motor, a pump, valves…. This is just the basic circuit for a hydraulic cylinder, the mechanism is working perfectly but the



necessary installation is equally imposing and needs a large space which makes our system more complex, and its components cluttered.

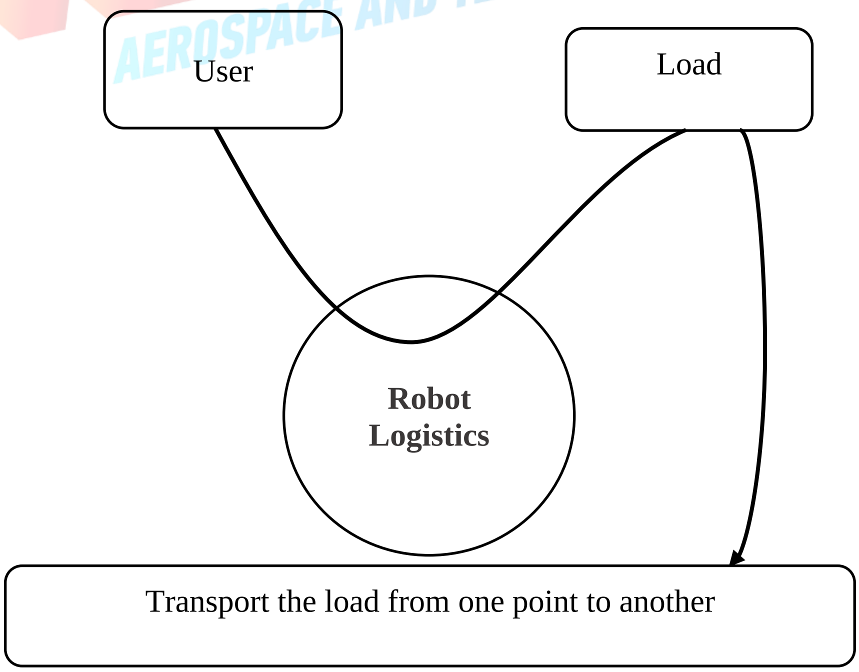
Our team of gathered as to add new aspects to a classical load moving robot.
From the mechanical to the electrical components, all was improved. And this will be detailed in the upcoming sections.

4. VEHICLE DESIGN

4.1. System Design

Functional analysis is an approach that "consists of researching and characterizing the functions offered by a product to meet the needs of its user. »
The approach is generally conducted in project mode and can be used to create (design) or improve (redesign) a product.
The object targeted by the approach can be an object, a material or a process, an organization, software, etc.
Needs are of all kinds and are expressed individually or collectively, objectively, or subjectively, with varying degrees of justification.
Where the studied functions are also diverse: service functions, evaluation functions, processing functions the framework of the study must also be considered constraints or variables deduced from the environment, regulations, uses, etc.

4.1.1. horned beast diagram



4.1.2. octopus diagram

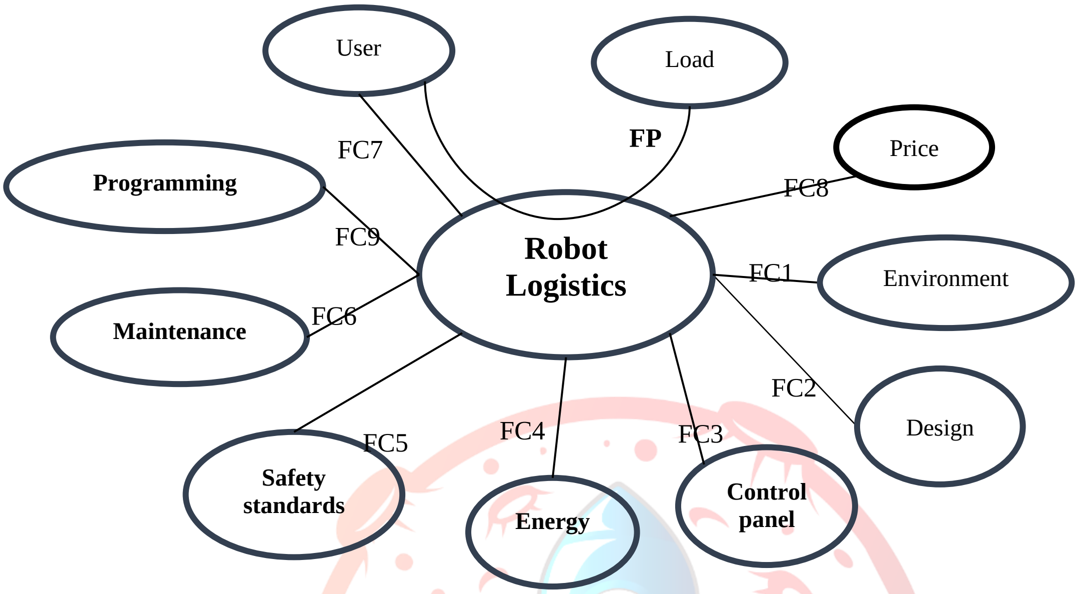
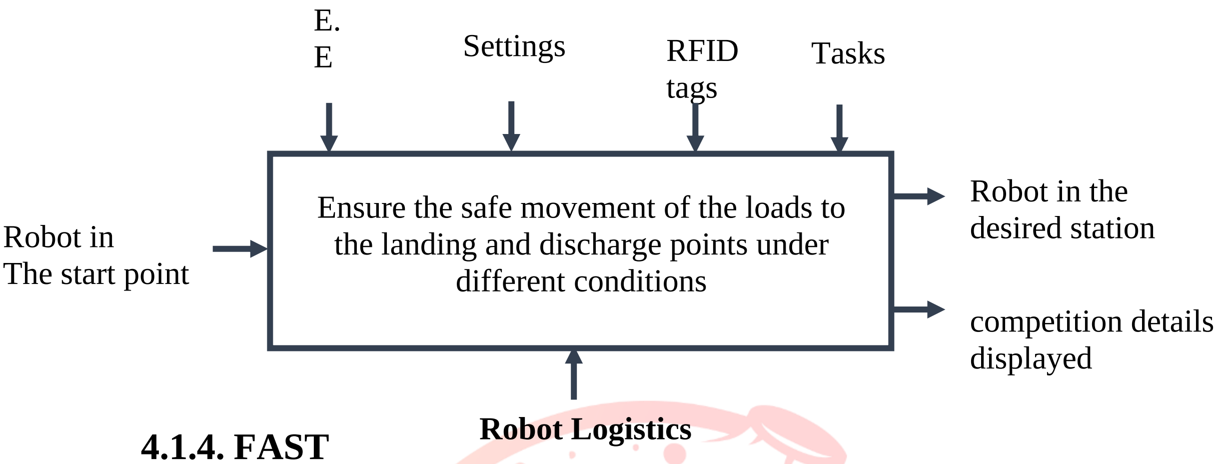


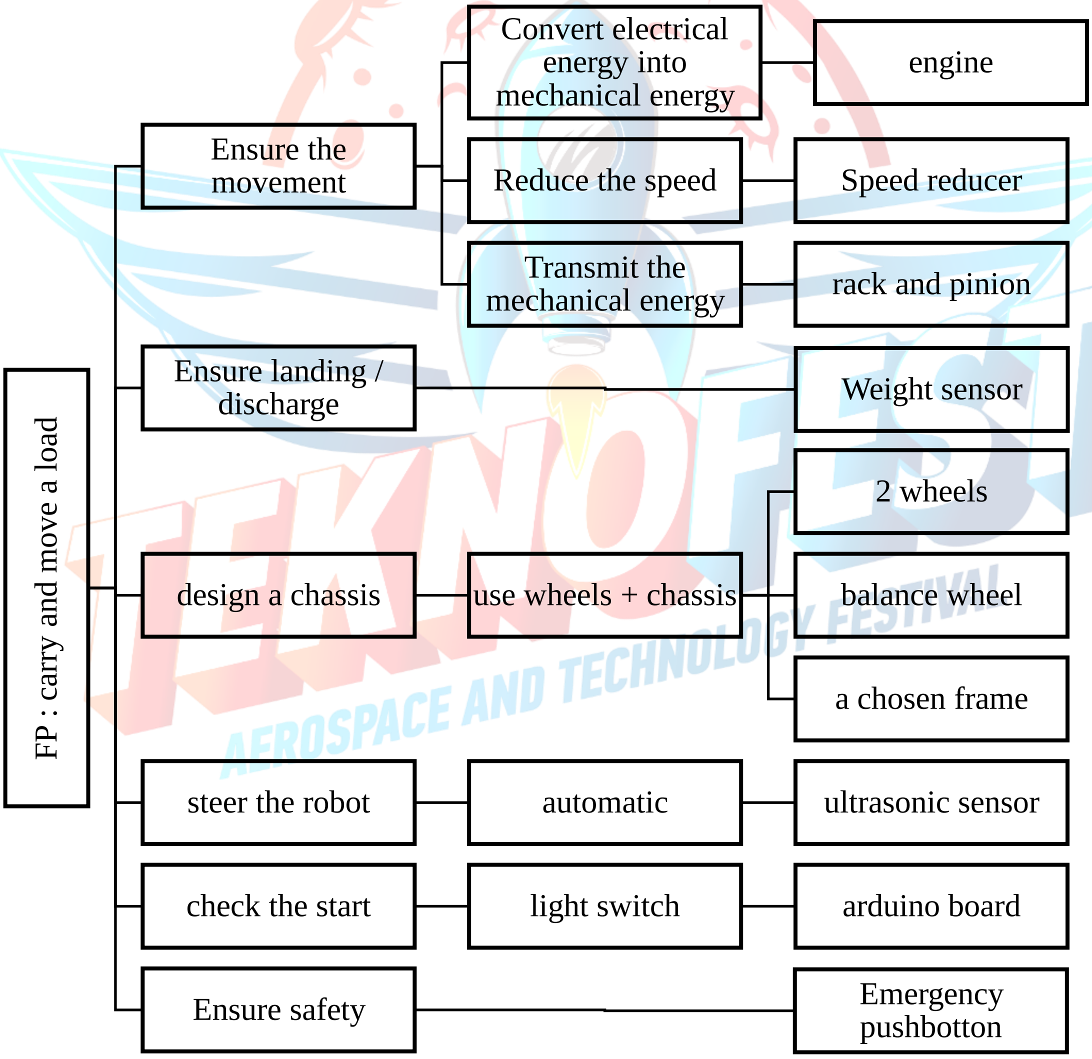
Figure 4.1:1:Diagram of interactors

FP	Transport the load from one point to another
FC1	Operate continuously and be compatible with its environment
FC2	Have a simple and esthetic design
FC3	Communicate it status with a monitoring interface that facilitate control
FC4	Use electrical energy to recharge it battery
FC5	Respect all the safety standards
FC6	Be easy to maintain
FC7	Be easy to use
FC8	Fit within the competition budget
FC9	Allow the user to easily program the robot

4.1.3. SADT A0 diagram



4.1.4. FAST



4.2. Mechanical Design of the Vehicle

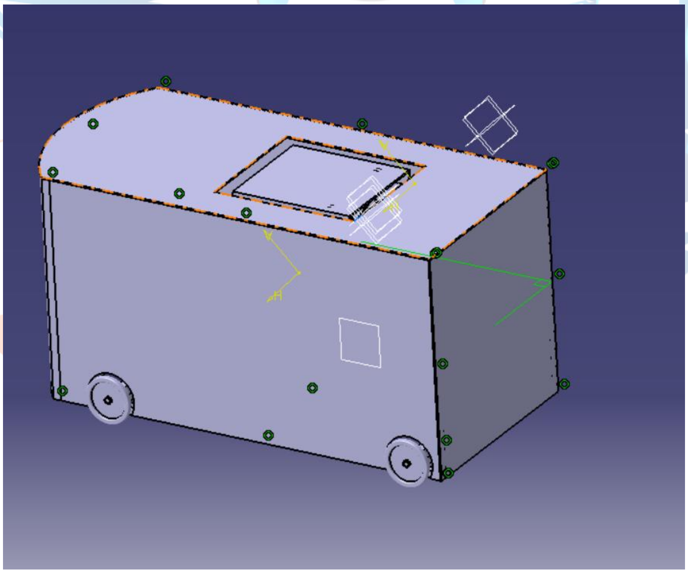
4.2.1. Mechanical Design Process

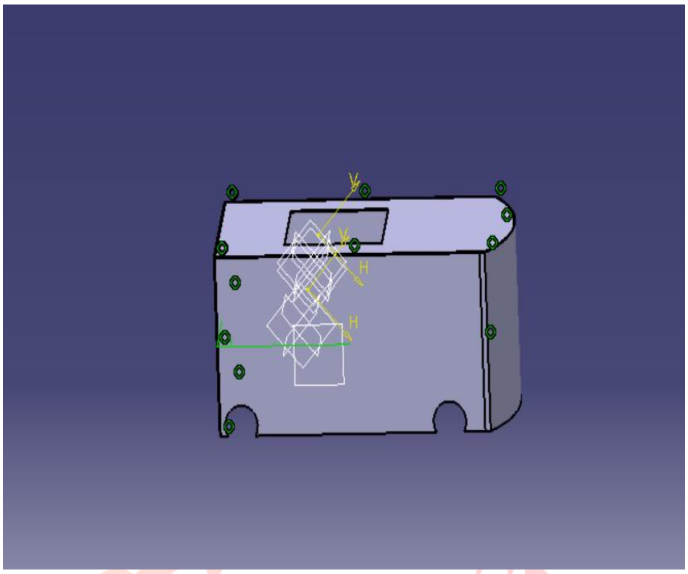
- **CATIA V5**

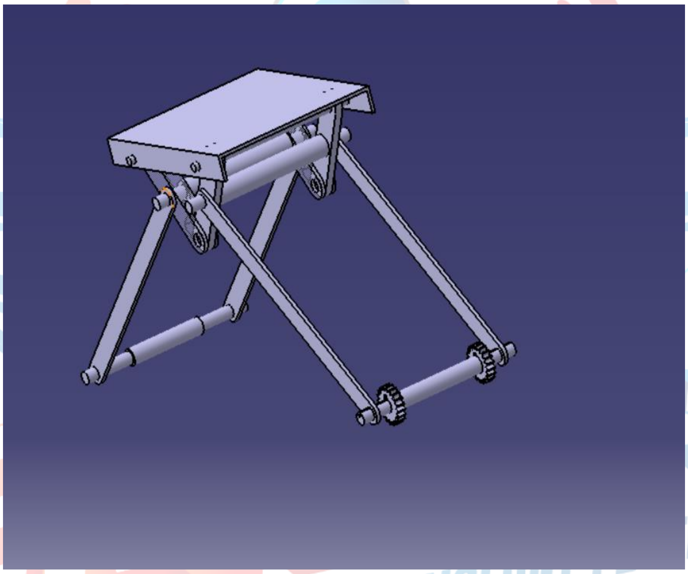
is a multi-platform software suite for computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided engineering (CAE), 3D modeling and Product lifecycle management (PLM), developed by the French company Dassault Systèmes.

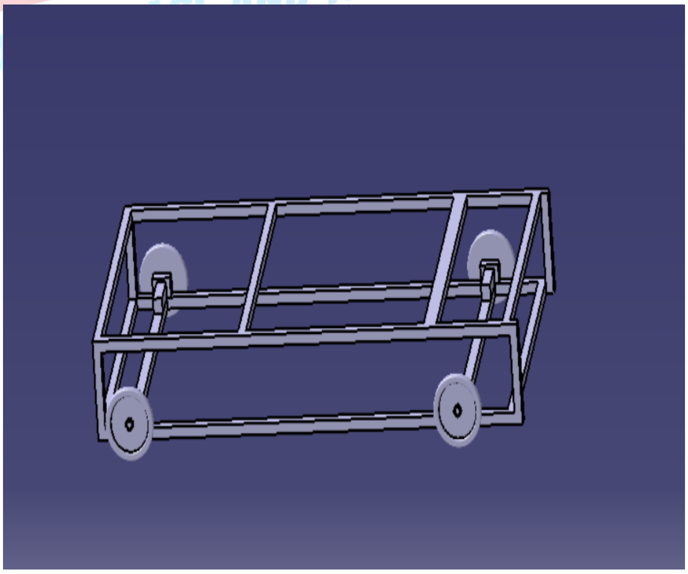
Since it supports multiple stages of product development from conceptualization, design and engineering to manufacturing, it is considered a CAx-software and is sometimes referred to as a 3D Product Lifecycle Management software suite. Like most of its competition it facilitates collaborative engineering through an integrated cloud service and have support to be used across disciplines including surfacing & shape design, electrical, fluid and electronic systems design, mechanical engineering and systems engineering.

In this part we go to the components of our robot.The design of the system is carried out on CATIA V5.



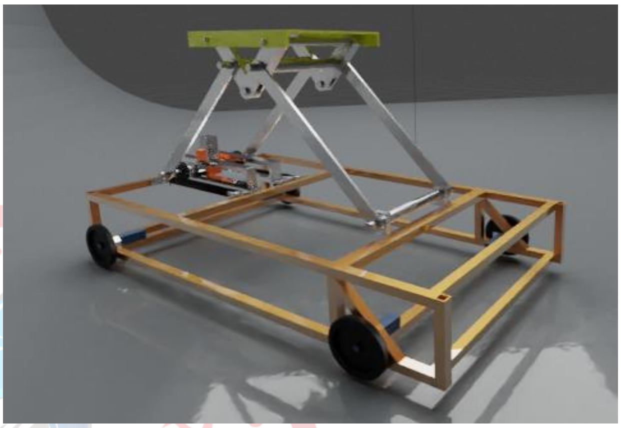


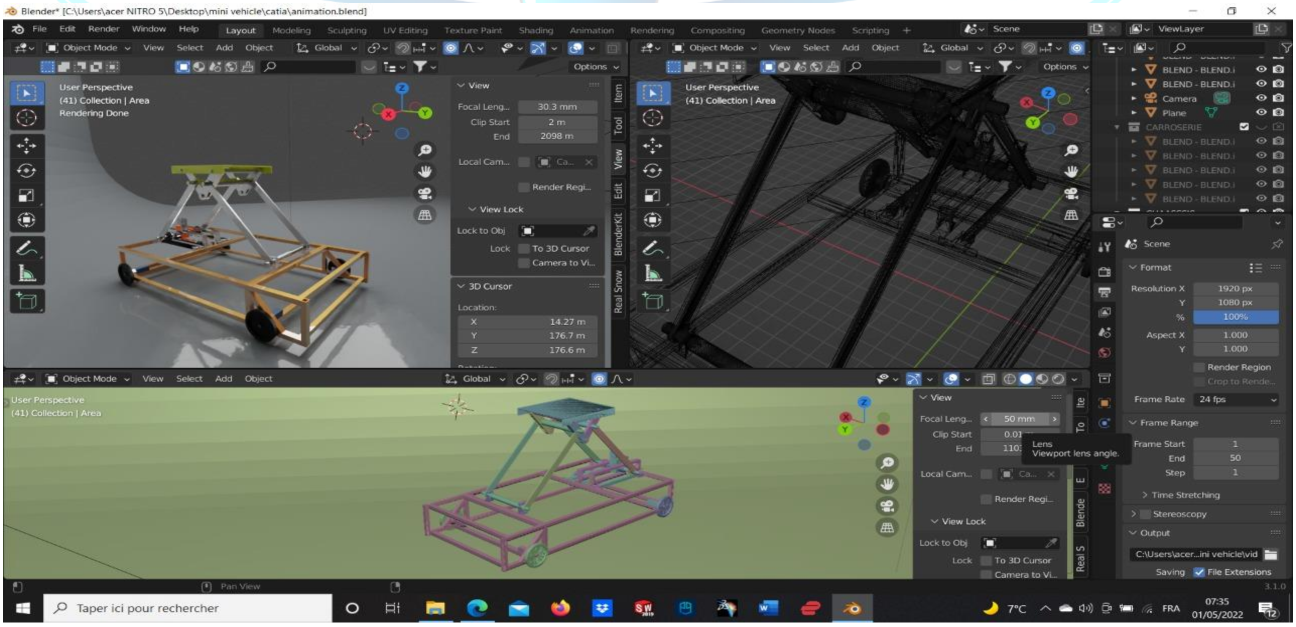




- **BLENDER**

We have also used blender software to create animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality





https://drive.google.com/file/d/1qKaLpk6oFHmqF0WkqhnoBYXF53ZXUGyD/view?usp=drivesdk

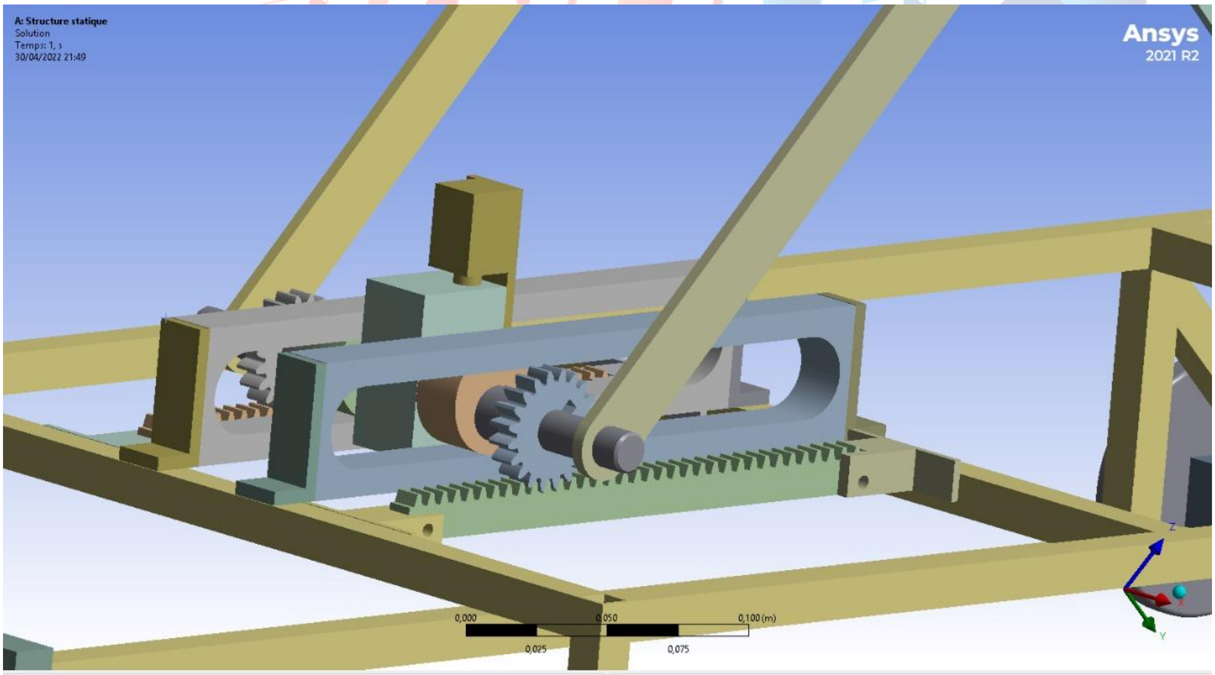
- **Ansys**

Ansys develops and markets engineering simulation software for use across the product life cycle. Ansys Mechanical finite element analysis software is used to simulate computer models of structures, electronics, or machine components for analyzing the strength, toughness, elasticity, temperature distribution, electromagnetism, fluid flow, and other attributes. Ansys is used to determine how a product will function with different specifications, without building test products or conducting crash tests. For example, Ansys software may simulate how a bridge will hold up after years of traffic, how to best process salmon in a cannery to reduce waste, or how to design a slide that uses less material without sacrificing safety.

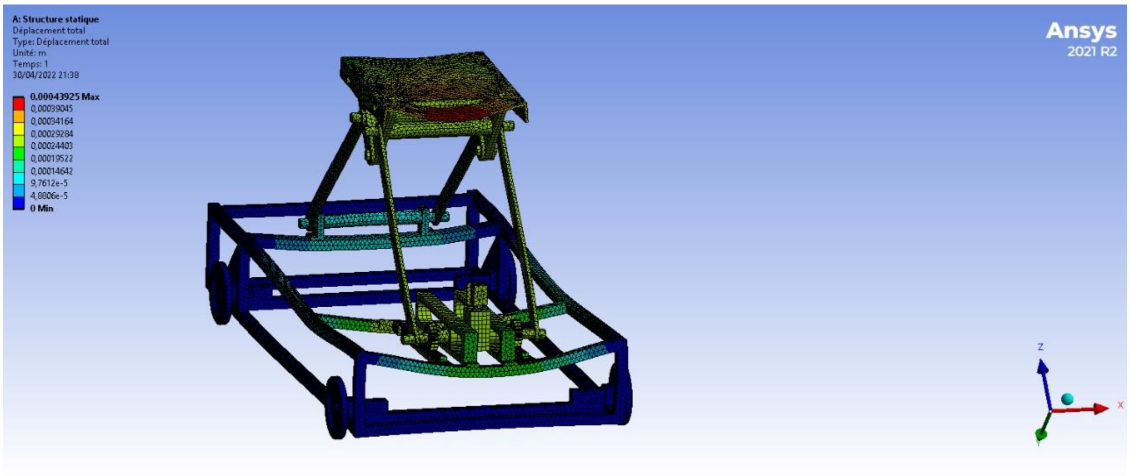
For this part, and to test our vehicle, we did a simulation using ANSYS, the first simulation-driven design tool to combine instant physics simulation, proven Ansys high-fidelity simulation and interactive geometry modeling in a single user experience.

our system is composed of a chassis which is in turn composed of prismatic tubes with a 2 mm thickness and it has not undergone a remarkable deformation apart from the place where there is a translational guidance and a pivot guidance with the upper table that will carry the loads.

- **Our mechanical system**

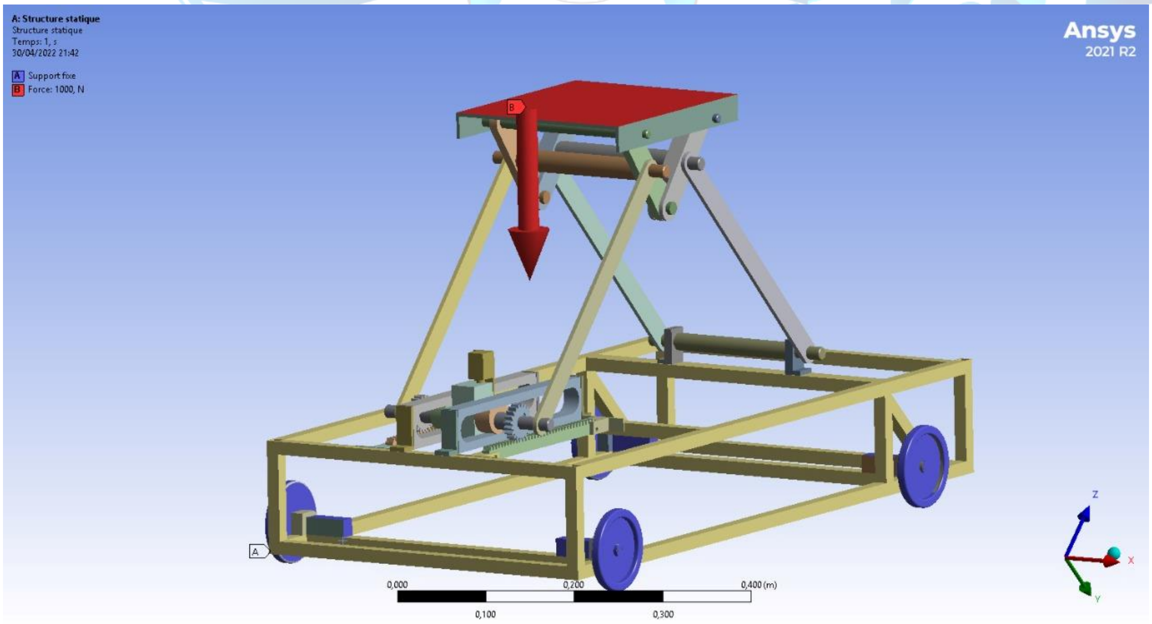


- **For our first test, we found the following results**

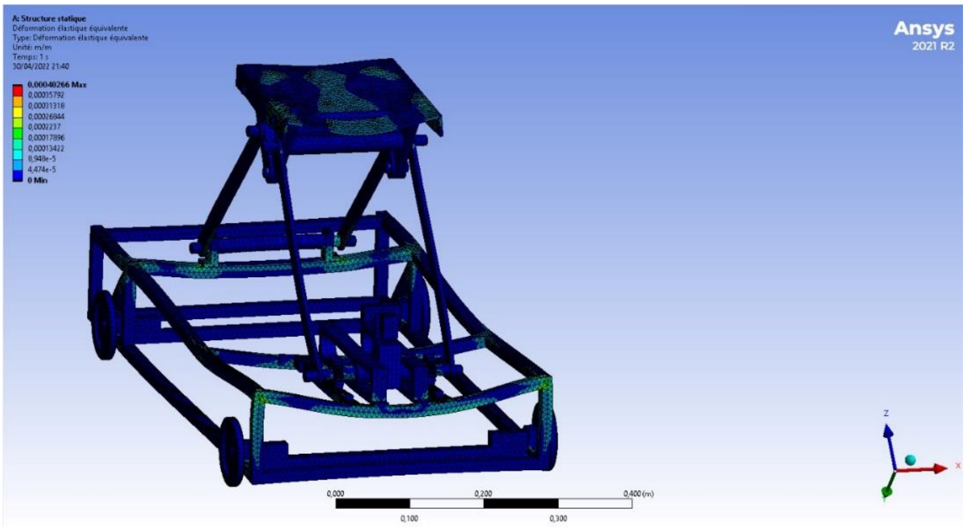


this result represents the displacement of each component in our system; we note that the displacement is negligible (order of 10^{-3}), it should also be noted that the displacement affects the operation of our mechanical system; and since it is negligible, therefore there will be no failure during the operation phase.

- **For our second test**

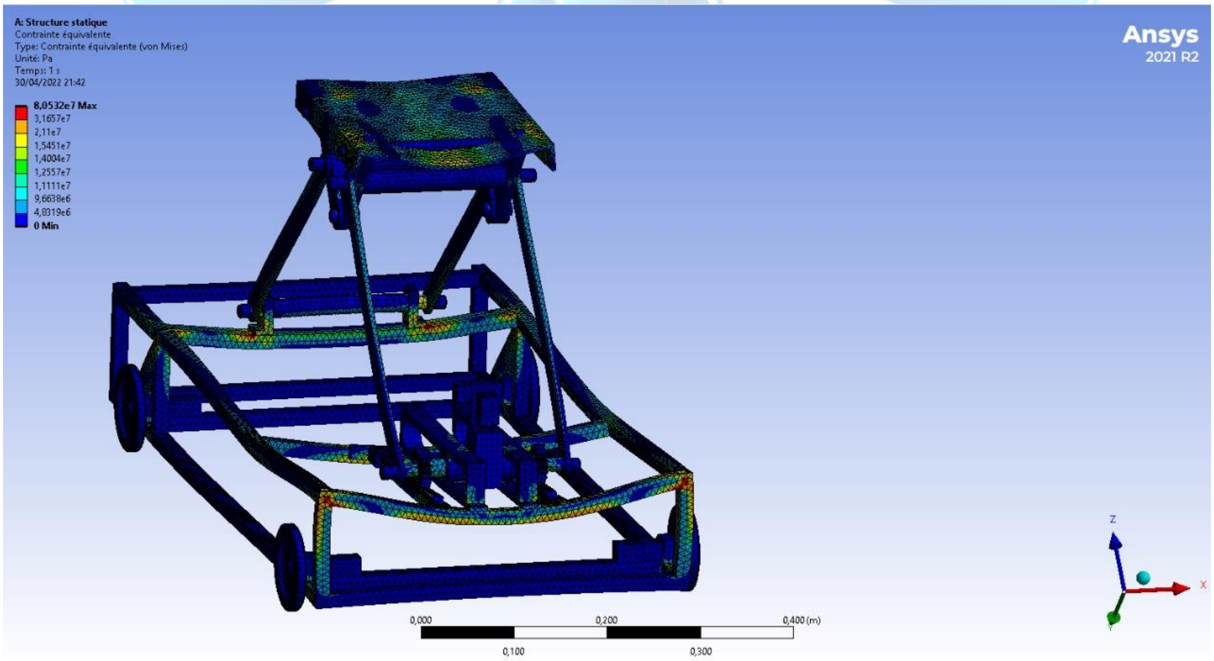


this result represents the boundary conditions of our system and the charges applied to it, we note that the wheels (colored blue) are the fixed elements in our simulation because they carry the support, and the table (colored red) is the element in which all the charges are applied (the masses of the loads)



to check if we have plastic, elastic deformation or a rupture, we made the simulation below: this simulation shows us the percentage of displacement of each component compared to their initial position. we note that the displacements are negligible, and this is due to the positioning of the components in the preliminary design phase.

- **stress distribution in our vehicle**



we can see with this simulation the places in our system which undergoes stress concentration, which implies the places that needs more support.

https://drive.google.com/file/d/1qjY0vZCA57ksAVfiYSrMOMZSX16d9e7j/view?usp=drivesdk

4.2.2. matériaux

The choice of materials is a fundamental and very complex task. Indeed, in many cases, it does not only concern a purely technical aspect meeting functional requirements, but also expectations relating to user preferences within the framework of a specific market. Thus, the choice of materials must be analyzed from the angle of materials engineering but also from that of industrial design and consider all the information associated with it.

It is imperative to make the choice of the material-process pair based on the criteria of the specifications and in a context of sustainable development.

the product demands certain performance from the material

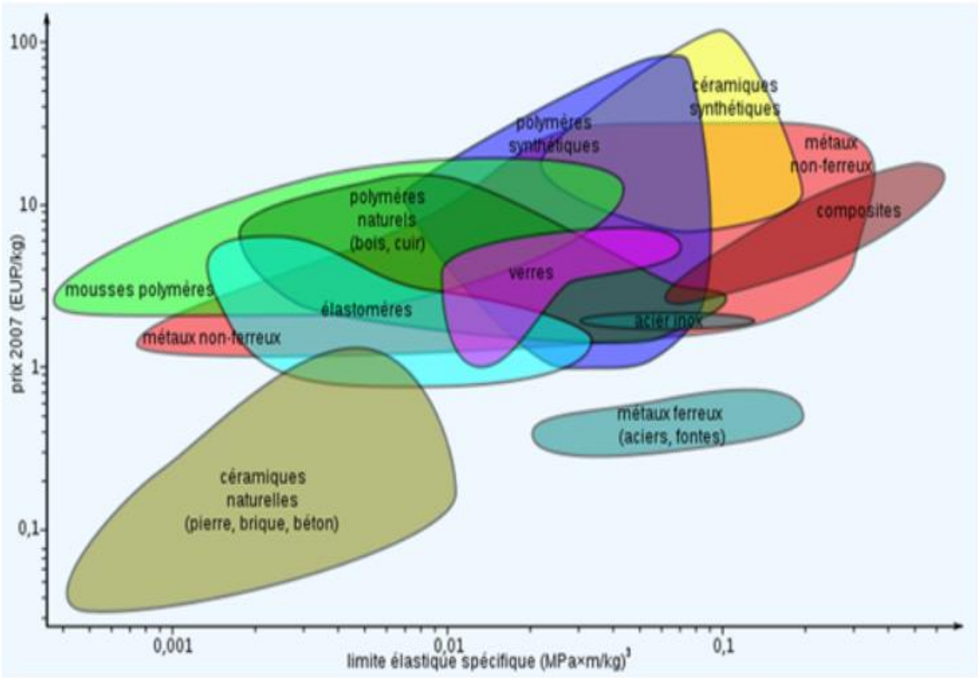
the process imposes manufacturing or construction constraints.

	R_m (Mpa)	R_e (Mpa)	E (Gpa)	ν	densité	A%
Acier ordinaire	300/1100	200/900	210	0,3	7,85	>17
Acier hautes caractéristiques	1100/1800	1000/1700	210	0,3	7,85	
Aciers inoxydables austénitiques		180/240	195	0,3	7,85	40
Alliages aluminium	200/600	100/500	70	0,34	2,8	5 / 30
Titane	650	500	110	0,34	4,5	35 / 55

pour une résistance égale à un effort axial (indice 1 pour une tôle en acier) selon les référentiels Dunod

	indice de prix	indice de poids
Acier inox	3,5	0,7
Alliage Al	4,4	0,5
Allage Ti	23	0,25

To optimize the choice of material, it is necessary to determine the desired performance, which will result in a combination of material properties. For example, it is possible to express the performance of a material by the relationship between its price and its mass (Price/M).



After our numerical analysis on Ansys we can choose a material that will resist the load applied on it.

For this purpose, the robot can be made from ordinary steel.

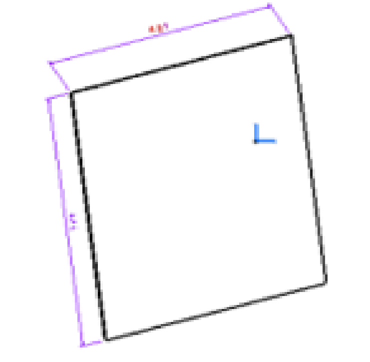
4.2.3. Production méthodes

there are several production methods to build our robot. in this part we will focus on the manufacturing range of the frames.

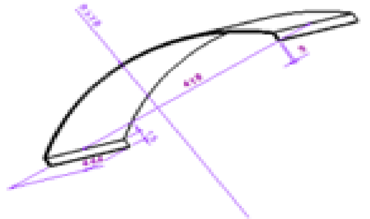
so we have:

- **Lift table: component machining, bolted assembly ….**
- **Frame: profile wading.**

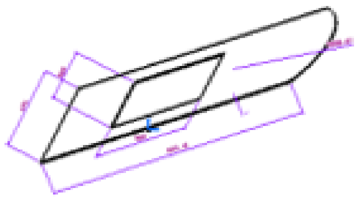
- **A sheet metal shaping techniques for the body**



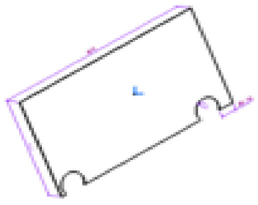
Phase	Operation	Tools
10	Cutting	shear



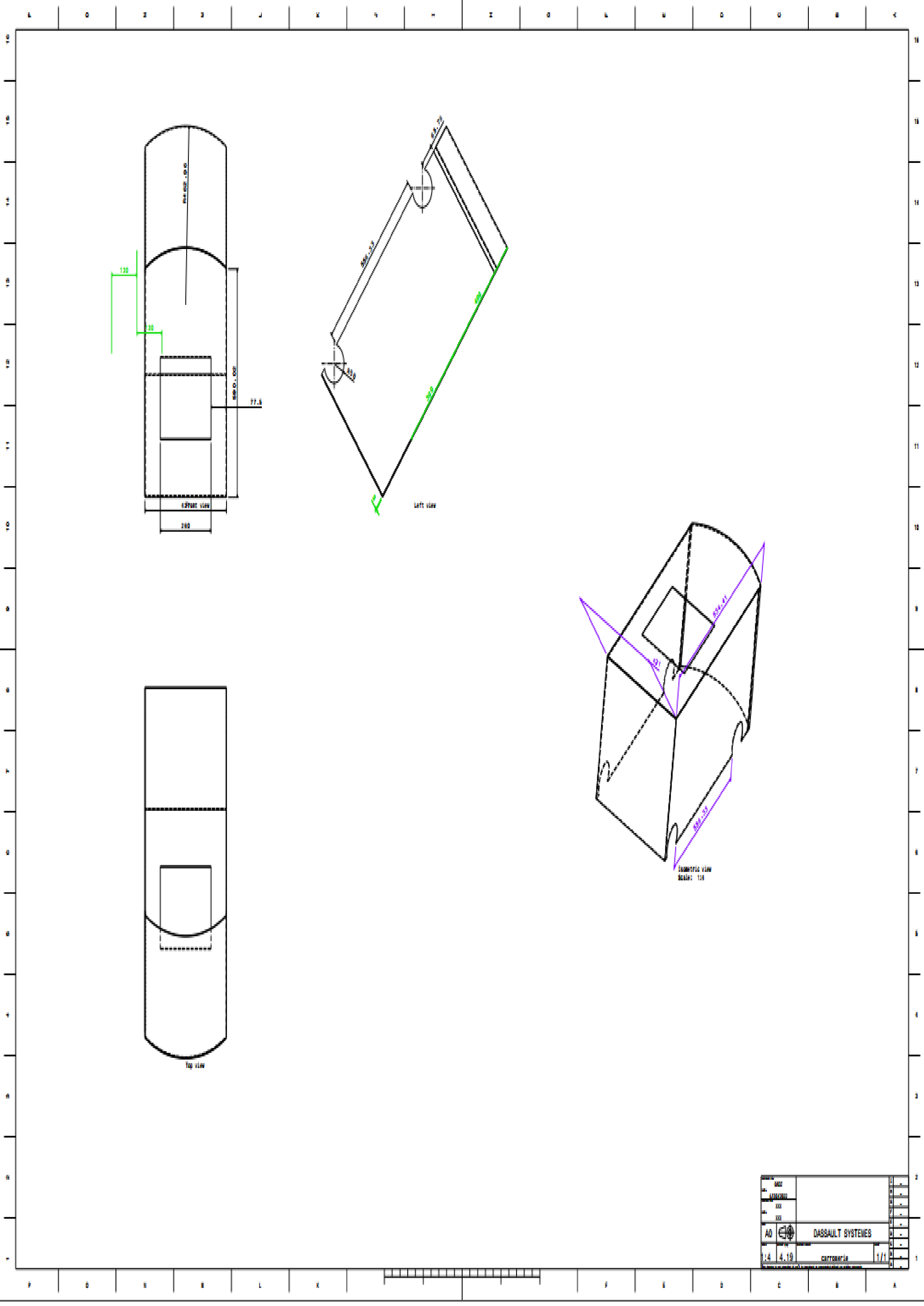
Phase	Operation	Tools
10	Cutting	shear
20	Bending	Bender
30	Folding	Folding machine



Phase	Operation	Tools
10	Cutting	shear
20	Puncture	shear



Phase	Operation	Tools
10	Cutting	shear
20	Punching	shear



130
130
77.5
260
Left view
Top view
Isometric view
Scale: 1:4
A0
DASSAULT SYSTEMES
1:4
4.19
carrosserie
1/1

4.2.4. engine choice

- **mathematical calculation**

- **Data needed for the calculation**

- m is the total mass of the robot, in Kg
- v(t) the speed at a time t, in Meters / Second
- d a distance, in Meters
- g gravitational constant, in Meters / s^2 (it can be rounded to 10)
- r diameter of the wheel, in Meters
- alpha the slope that the robot must climb, (preferably in radians)

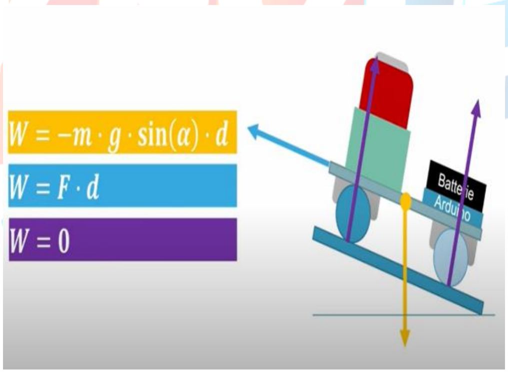
- **Theorem to apply**

To calculate the power of the motor, we will use the theorem of kinetic energy. Here is what it says:

The variation of the kinetic energy between instant 0 and instant t is equal to the sum of the works of the external forces. The work of a force is equal to the multiplication of the force vector by the displacement vector.

So, we have:

The motor force that makes the robot move forward (force forward and parallel to the ground: the robot climbs the slope). The work is positive because the force is in the same direction as the movement of the robot.



we have

m	140kg
v(t)	0.3m/s
d	15m
g	$9.81m.s^{-2}$
r	5cm
α	30°

Kinetic energy theorem

$$\Delta E_c = \sum W\, F_{extern}$$

With the force of the engine

$$W = F.d$$

Weight work

$$W = -m.g.\sin(\alpha).d$$

It is assumed that the rolling without slip
so

$$E_c = \frac{1}{2} m\, v^2(t)$$

$$=> \frac{1}{2} m\, v^2 = F_{mot}.d - m.g \times \sin(\alpha).d$$

$$=> F_{mot} = m.g.\sin(\alpha) + \frac{1}{2}.\frac{m}{d}.v^2$$

$$=> F_{mot} = 140.9.81.\sin(30) + \frac{1}{2}.\frac{140}{15}.0.09$$

$$F_{mot} = 701N$$

This force F is such that the slope is compensated and we have the desired movement! The force compensates the terrestrial attraction and gives the desired acceleration.
A motor provides torque. To know this torque of each wheel, simply we divided the force by the number of wheels and multiply it by the radius of the robot wheel:

$$C_m = \frac{F}{4}.r$$

$$C_m = \frac{701}{4}.0.05$$

$$C_m = 8.77\, N.m$$

the robot must have a torque Cm to reach a speed v(t)! With these values, we will be able to calculate the power that the engine must provide! And with the previous results, we can easily calculate the power of the motor:

$$P_{motor} = C_m.\omega$$

$$P_{motor} = 52.62\, W$$

The result of the calculation gives us the engine power!
Engine acceleration:

$$m.a = F - m.g.\sin(\alpha)$$

so

$$a = 0.07\ m.s^{-1}$$

The motor chosen: Servomoteurs in kit - K series for our robot.

- **Description**

Servomoteurs kits are the ideal solution for designing machines that require high performance in a small footprint. Kit motors allow direct integration into the mechanics, eliminating transmission elements and reducing the complexity of the machine. The use of kit motors makes it possible to reduce size and increase reliability.

- **Advantages**

Low tension

Compactness and robustness

Large hollow shaft

Direct drive: precise and dynamic movement

Optional Hall effect sensor

Improved system rigidity

Simple, lightweight and compact machine design

No need for coupling system

Reduced overall costs

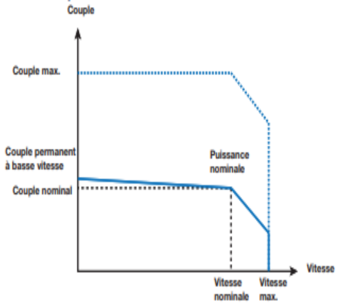
Increases reliability and reduces maintenance

- **Integration assistance**

Caractéristiques techniques

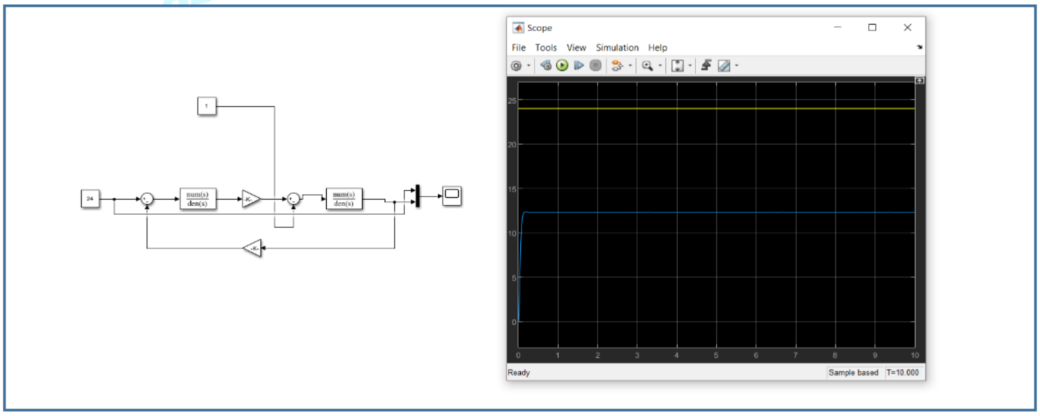
Refroidissement	Convection naturelle
Puissance	0,06...2 kW
Couple	0,07...23 Nm
Vitesse max.	10 000 min^{-1}
Nombre de pôles	4 (K32) / 6 (K44) / 8 (K64) / 12 (K89) / 18 (K178)
Diamètre	32 – 44 – 64 – 89 - 178 mm
Alimentation	12 - 24 - 48 - 96 VDC 240 VAC seulement pour la taille K178

Données techniques



Moteur	Puissance nominale Pn	Couple nominal Mn	Vitesse Nominale Nn	Courant nominal In	Couple à basse vitesse Mo	Courant de rotation lente Io	Couple max. M max	Courant max. I max	Max. Vitesse N max	Fréquence à vitesse max.	Moment d'inertie J
	[kW]	[Nm]	[min⁻¹]	[Arms]	[Nm]	[Arms]	[Nm]	[Arms]	[min⁻¹]	[Hz]	[kgmm²]
Alimentation 24 VDC											
K064050-8Y_	0,058	0,527	1053	3,954	0,530	4,0	1,86	14,0	1830	122	9
K064050-9Y_	0,039	0,526	700	3,156	0,527	3,2	1,85	11,1	1464	98	9
K064050-EY_	0,022	0,530	396	2,489	0,531	2,5	1,87	8,8	1144	76	9
K064100-8Y_	0,047	0,933	484	3,502	0,937	3,5	3,29	12,3	915	61	18
K064100-9Y_	0,030	0,931	305	2,794	0,933	2,8	3,28	9,8	732	49	18
K064100-EY_	0,015	0,939	152	2,202	0,940	2,2	3,30	7,7	572	38	18
K064200-8Y_	0,033	1,560	204	2,928	1,565	2,9	5,50	10,3	457	30	36
K064200-9Y_	0,018	1,556	113	2,336	1,559	2,3	5,47	8,2	366	24	36
K064200-EY_	0,006	1,569	36	1,841	1,570	1,8	5,51	6,5	286	19	36
K089050-6Y_	0,117	1,332	839	7,501	1,343	7,6	4,72	26,6	1373	137	37
K089050-7Y_	0,082	1,320	597	6,038	1,327	6,1	4.66	21,3	1115	112	37
K089050-9Y_	0,030	1,313	216	3,845	1,315	3,9	4.62	13,5	714	71	37
K089100-6Y_	0,098	2,353	396	6,626	2,369	6,7	8,32	23,4	686	69	78
K089100-7Y_	0,066	2,330	272	5,331	2,341	5,4	8,22	18,8	558	56	78
K089100-9Y_	0,019	2,318	77	3,394	2,320	3,4	8,15	11.9	357	36	78
K089200-4Y_	0,153	3,850	379	8,809	3,901	8,9	13,7	31.3	558	56	150
K089200-7Y_	0,045	3,883	111	4,441	3,896	4,5	13,7	15,6	279	28	150
K089200-9Y_	0,004	3,861	11	2,827	3,862	2,8	13,6	9,9	178	18	150
K178050-6Y_	0,217	6,969	297	13,885	7,030	14,0	26,8	53,3	486	73	470
K178050-8Y_	0,100	6,991	137	8,797	7,016	8,8	26,7	33,6	307	46	470
K178050-EY	0,024	6,986	33	5,567	6,991	5,6	26,6	21,2	194	29	470
K178100-8Y_	0,077	13,639	54	8,583	13,668	8,6	48,0	30.2	153	23	920
K178100-9Y_	0,035	13,667	24	6,809	13,680	6,8	48,0	23,9	121	18	920

This allows us to model our engine on SIMULINK



The Step on the far left is used to put the tension in the system.

The circle with the + / - is normally known! What does it do? It regulates the system: it will reduce the relative voltage at the motor terminals, which will have the effect of reducing the acceleration and tending towards a constant speed of rotation.

The resistivity block translates the resistance of the motor and therefore converts voltage to current. We found the normal resistance ®, but also the resistivity (L) with the term "s" which is the variable in the frequency domain (homogeneous to the inverse of a time, i.e., a frequency). We therefore obtain an intensity at the output of this block.

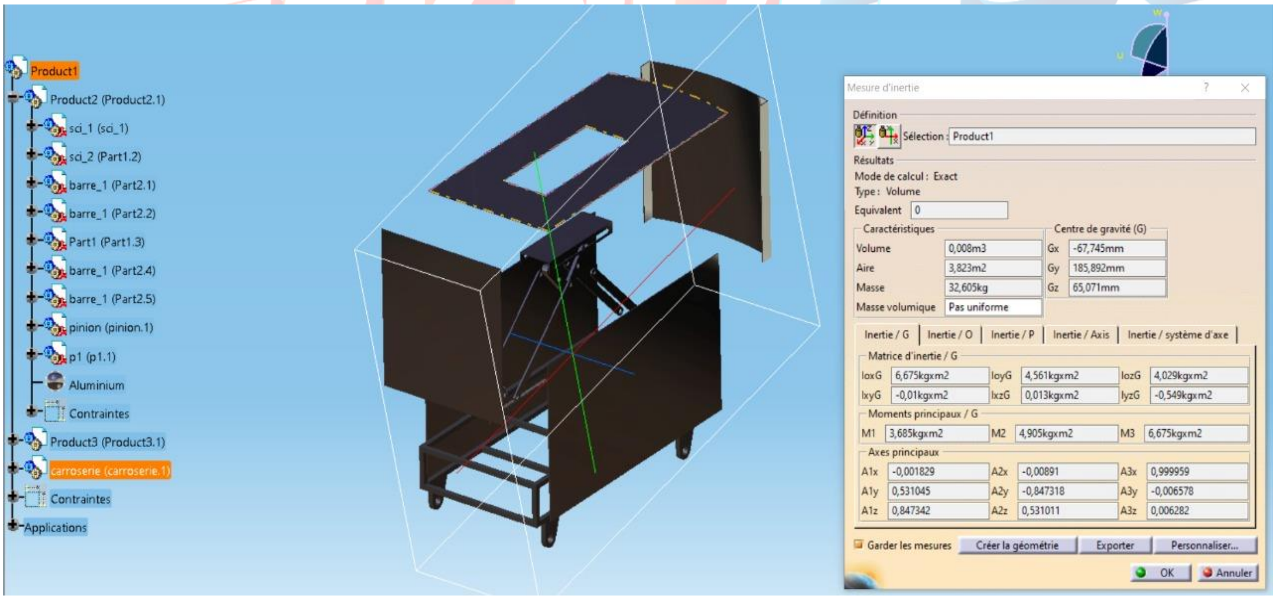
Torque constancy converts current into torque.

From the torque provided by the motor, the disturbances (friction, slope, etc.) are removed in the form of torque.

We therefore obtain the total torque on the axis of the motor. This torque will make it possible to set the robot in motion, which has a certain inertia (J) and which is subject to certain viscous friction (f) at the level of the axis. We therefore have a rotation speed that can be displayed thanks to the scope.

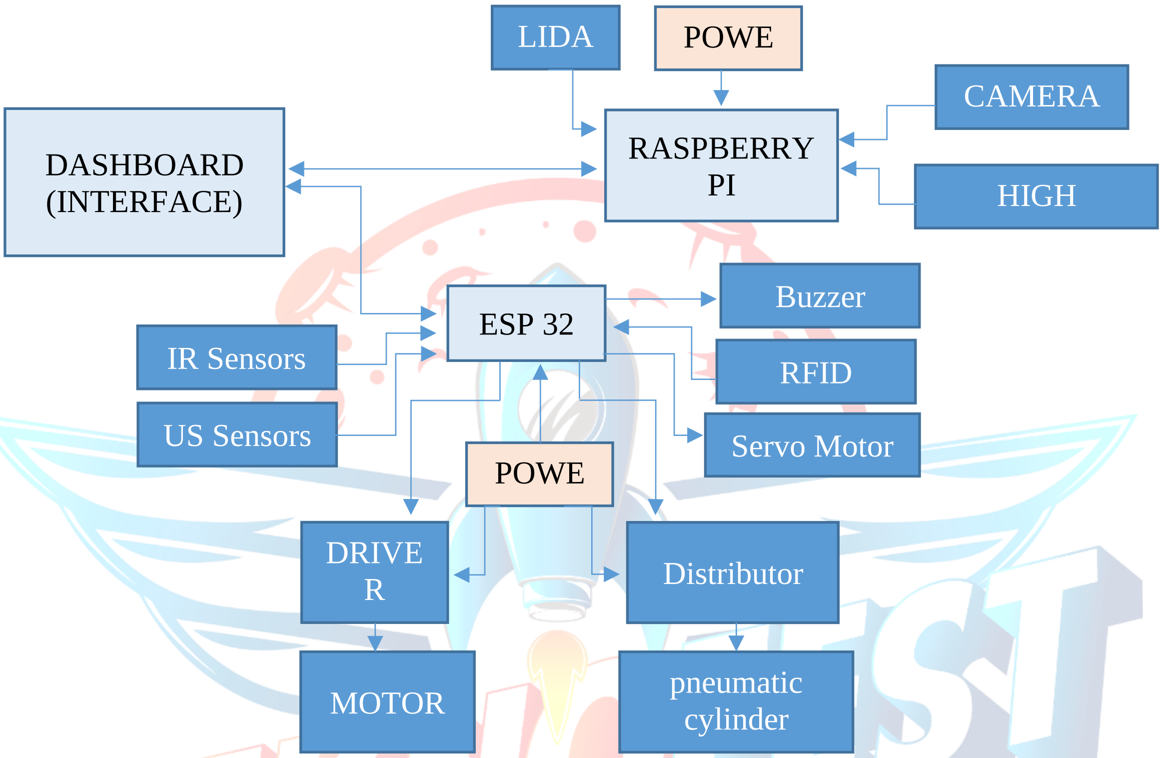
Finally, this rotational speed is multiplied by the speed constant which will regulate the voltage in the motor. Physically speaking, it is the electromotive force, generated by the rotation of the rotor in the stator, which will create a voltage opposite to that of the supply.: tare: In short... it's the influence of the rotational speed of the motor on the relative voltage of this motor.

4.2.5. Physical Propertie



4.3. Electronic Design, Algorithm and Software Design

4.3.1. Electronic Design Process



Raspberry Pi is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in countries. The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices.

The Raspberry Pi Camera Module v2 is a custom designed add-on board for the Raspberry Pi, with a fixed lens. It is capable of 3280 x 2464-pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

ESP32 is a series of system-on-chip microcontrollers from Expressive Systems, based on Tensilica's Xtensa LX6 architecture, integrating dual-mode Wi-Fi and Bluetooth management, and a DSP. It is an evolution of ESP8266.

An infrared sensor (IR sensor) is a radiation-sensitive optoelectronic component with a spectral sensitivity in the infrared wavelength range 780 nm … 50 µm. IR sensors are now widely used in motion detectors, which are used in building services to switch on lamps or in alarm systems to detect unwelcome guests. In a defined angle range, the sensor elements detect the heat radiation (infrared radiation) that changes over time and space due to the movement of people. Such infrared sensors only have to meet relatively low requirements and are low-cost mass-produced items. InfraTec does not supply such products, InfraTec develops, produces and sells pyroelectric detectors.

An ultrasonic sensor emits short high-frequency sound pulses at regular intervals. These pulses propagate through the air at the speed of sound. When they encounter an object, they reflect and return as an echo to the sensor. This then calculates the distance separating it from the target based on the time elapsed between the transmission of the signal and the reception of the echo.

LiDAR is a remote sensing and ranging method similar to radar, but which emits pulses of infrared light, instead of radio waves, and then measures the time it returns after being reflected off nearby objects.

RFID: Radio-identification, most often referred to by the acronym RFID (from English radio frequency identification), is a method for storing and retrieving data remotely using markers called "radio-tags".

A pneumatic cylinder is a linear actuator in which the energy of compressed air is transformed into mechanical work.

Electronic hardware needs to transfer the data from the motor movement directions and sensors to the software correctly through physical integration, hardware and devices.

- **Master electronic development hardware**

Raspberry Pi 4B 8Gb ram.   1500 DH

Side Electronic equipment:

1.Magnetic Sensor	500 DH
2. Intel RealSense 435i	250 DH
3. Intel T265	200 DH
4. IMX 219	150 DH
5. QR Barcode reader	100 DH
6. Arduino Mega 2560	200 DH
7. Motor Drivers	200 DH+100DH/U
8. Encoder	90 DH

9. HMI Display 250 DH

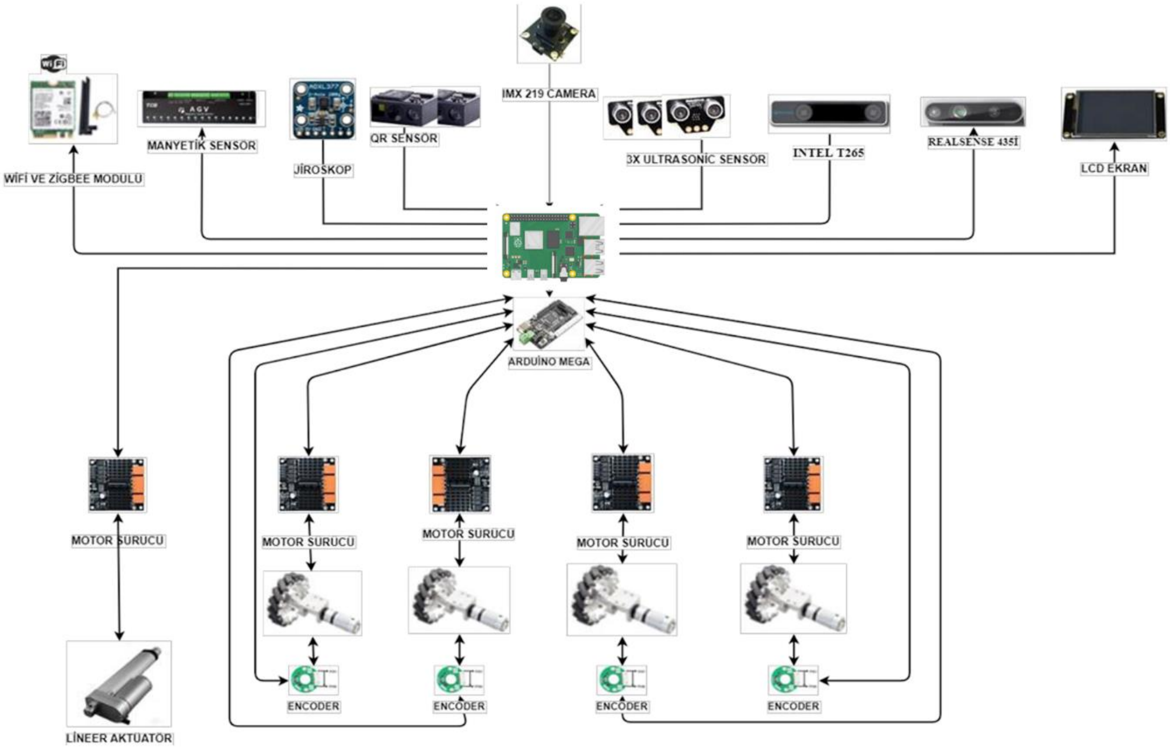
10. Gyroscope 100 DH

The job descriptions of the electronic systems within the system have been determined and communication will be provided over the ROS system by using the Raspberry pi 4b Master device according to the working styles. Hardware operating priorities have been determined in the system, and algorithms and flowchart will be created in this way. Considering the sample scenario, the operation of the hardware and mathematical calculations will be made accordingly. The duties and working mechanism of peripheral equipment and electronic equipment will be as follows.

- **Hardware working scenario**

16-bit data will be analyzed over the RS-232 communication protocol from magnetic sensors. The magnetic stripe will scan the information and assign lane tracking to our autonomous vehicle. Using Intel RealSense 435i depth camera and Intel T265 IMU device, 2D or 3D map of the track will be drawn and objects on the track will be marked. The QR barcode reader will detect the location and which direction the vehicle should go. It will be used to use the Arduino mega 2560 Slave as a device and to provide the movement commands of the wheels to the motors with the PWM driving method. Motor drivers will provide speed control by the voltage values that will be directed to the motors according to the command signals coming from the Arduino mega. The encoder is a magnetic type sensor that will read the rotation of the motors in the system. The encoder, which reads incrementally, will provide the speed information for the PID control system by reading the motor speed information with mathematical calculations. The gyroscope displays the X, Y, Z position information of the autonomous vehicle.

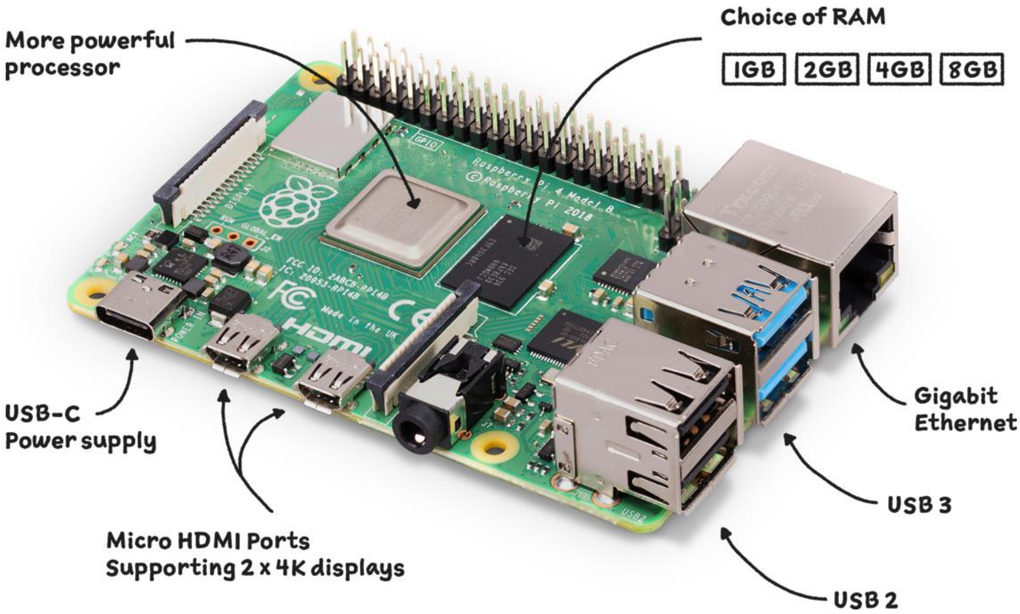
and will take part in autonomous movements and provide the vehicle's movement and direction capability in line with this information. The HMI screen, on the other hand, will be a screen design that will include the vehicle's battery status information and some parameters.



.

The Raspberry Pi is a credit card-sized, ARM-based, unibody nanocomputer designed by professors from the University of Cambridge's Computer Science Department as part of the Raspberry Pi3 Foundation.

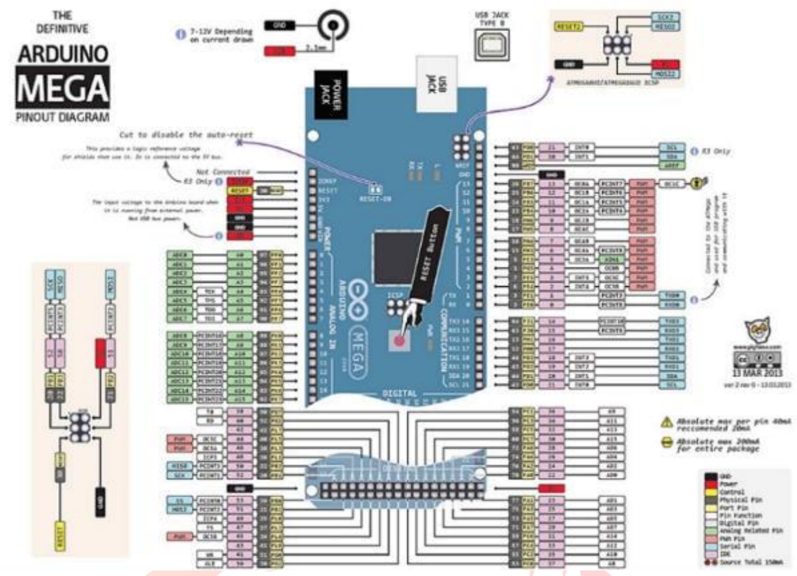
The Raspberry Pi was created in order to democratize access to computers and to digital making4 (an English term designating both problem-solving capacity and technical and computer skills)5. This democratization is possible due to the reduced cost of the Raspberry Pi, but also thanks to free software4. The Raspberry Pi allows the execution of several variants of the free operating system GNU/Linux, in particular Debian, and compatible software. It also works with the Microsoft Windows operating system: Windows 10 IoT Core6, Windows 10 on ARM (for the moment[When?] relatively unstable), that of Google Android Pi7 and even a version of the OS/MVT d 'IBM accompanied by the APL\3602 system.

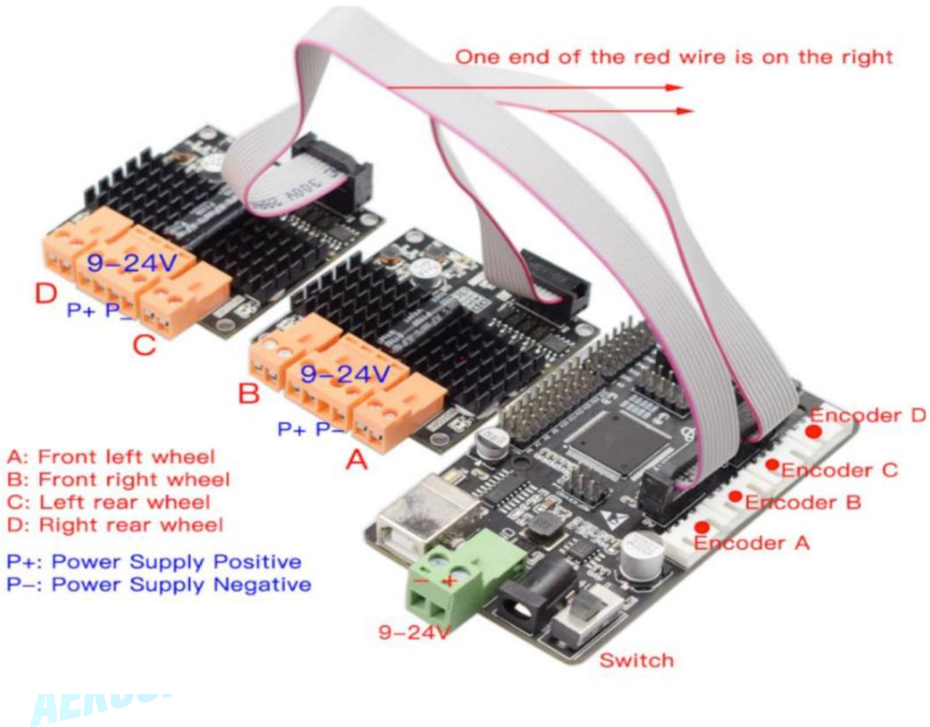


- **Arduino Mega 2560**

Considering the operational characteristics of the autonomous vehicle, a second microcontroller (Slave device) Arduino Mega 2560 will be used for the motor control system. The decision commands from the master device Xavier NX will be transmitted to Arduino Mega via serial communication and the motor motion system will be triggered. Arduino Mega 2560 will provide the mobility of four motors connected to 4 Mecanum wheels with PWM driving method. Arduino Mega 2560 has 54 digital input / output pins. 15 of them can be used as PWM outputs. In addition, 16 analog inputs, 4 UART (hardware serial ports), 1 16MHz crystal ICSP header and reset button are available. Engine.

A total of 24 pins will be used for the motion system. 4 of them are power pins, 4 pins are PWM driving, 4 digital pins forward capability, 4 digital pins reverse direction and 8 special pins will be used to read encoder information.

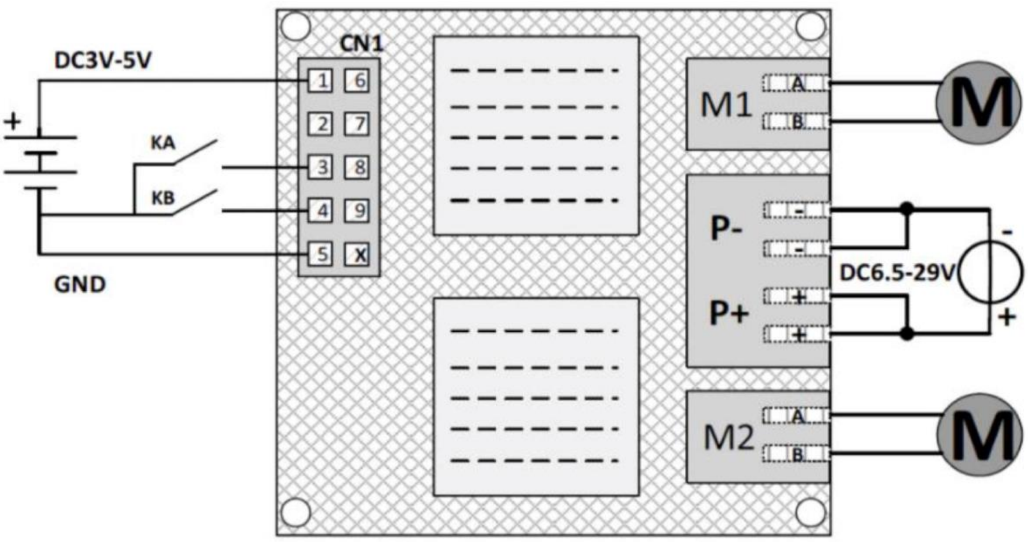




Motor Driver ESC is Electronic Speed Control's (Electronic Speed Control) is an abbreviation. ESC's control According to the information they get from the card, they pass the electrical energy over it and turn the engine at the desired speed. that allows us to are speed controllers. In short, ESCs control the motors. ESCs control the engine speed by reducing or increasing the electrical energy going to the engine.

Apart from the motor speed control, the excess current of the motor and the energy source used are determined by a certain When it falls below the limit, it provides protection by cutting the current to the motor windings. Same time It prevents the battery from falling below the critical limit. RPM speed information is taken from motors with encoder, PWM driving method and PID algorithm command will be sent to the drivers. 4 units using 2 motor drivers' direction and speed capability will be provided to the engine. 5-28V voltage according to the driver support technical catalog range, 12A multi-channel current output and 290W electrical power, these drivers are preferred.

has been done. In terms of safety, it is seen that the instantaneous peak current is based on 70A. Same time, The minimum pulse width of the PWM input in the driver is as low as 2us, precisely indicates that adjustments can be made. This dynamic PWM sensitivity plays a vital role in its control. Below is an example working circuit:



- **Encoder**

Encoder: While rotating or translational motion of a shaft, digital electricity is generated during this process.

It is the name given to the electromechanical device that produces the signal. All precision-controlled machines movements and position information are provided by encoders of different types and models. Well The encoder records the angular position change and movement of a shaft into an analog or digital code. transforms. Gyroscope and Accelerometer Sensor The gyroscope sensor is a system that can detect angular velocity. That is, a stationary object has three verticals The rotation direction and speed are determined by comparing the angular ratios on the axis. Perceived It processes the data by the processor and converts it into electrical signals. gyroscope, direction measurement or It works with the principle of maintaining angular balance, which is used in adjustment. gyroscopic movement It is based on the laws of physics and the principle of centrifugation. Gyroscope sensor like an accelerometer, but with a big difference; single accelerometer While the acceleration is measured on the coordinate, the rotation speed is measured according to the three coordinates in the gyroscope.

- **Ultrasonic Distance Sensor**

The autonomous vehicle needs to detect distances at different points. Ultrasonic distance sensors are preferred at different points in order for the vehicle to stop when it sees an obstacle and to detect that it is under load during load pick-up. The sensor is based on the constant sound velocity of sound waves. By receiving the sound waves sent from the T (Trigger) region in the R (Reciever) region, the principle of finding the distance works by multiplying the sound velocity of 343.2 m/s of the elapsed time.

4.3.2. Algorithm design process

- **Algorithms included in the project**

Main system algorithm.

tape tracking.

Positioning.

RFID and Navigation Determination.

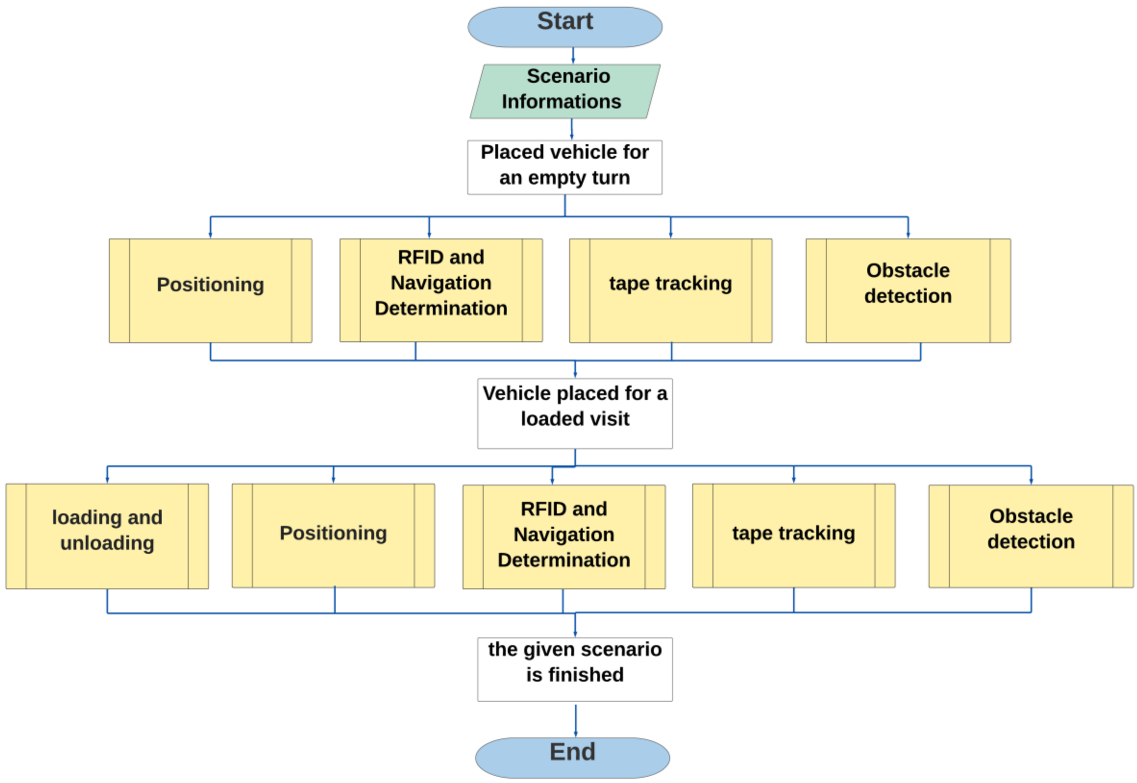
loading and unloading.

Security system algorithm

- **Main system algorithm.**

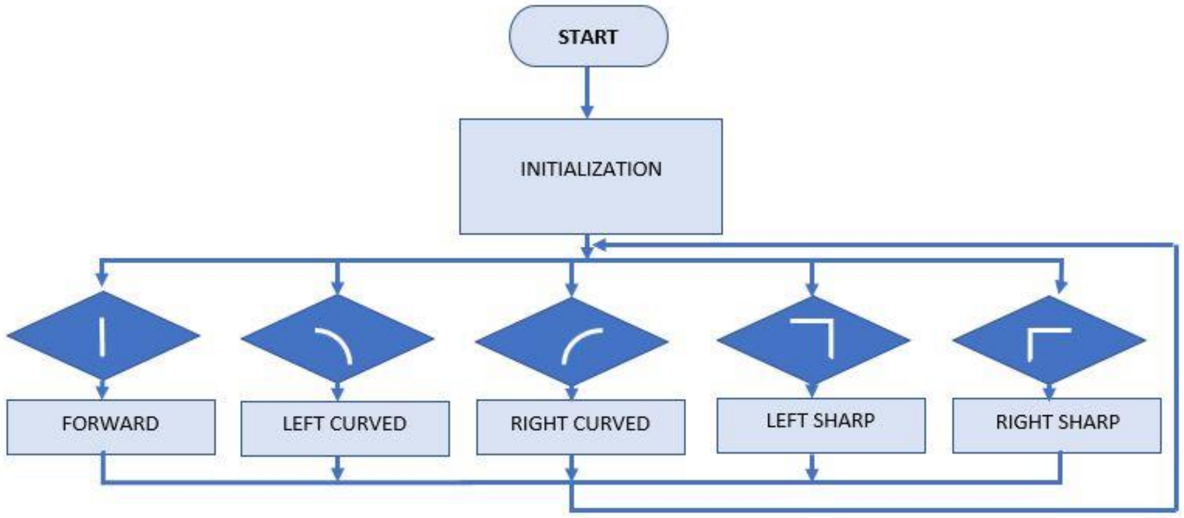
the main algorithm is encompassed by all the operating sub-algorithms.

First, the robot will position itself in the model, then it will read the RFID to know the scenario requested, then it will follow the trajectory using a third algorithm, in parallel it will detect the existing obstacles, and the fifth algorithm is just for loading and unloading in precise positions, in the end a security algorithm.



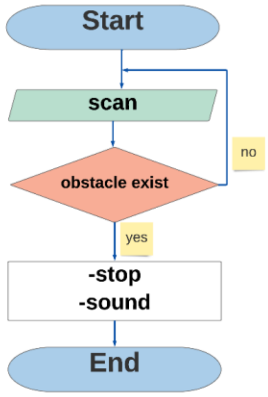
- **tape tracking algorithm.**

using the camera, the robot will follow the trajectory given by RFID using the algorithm below.



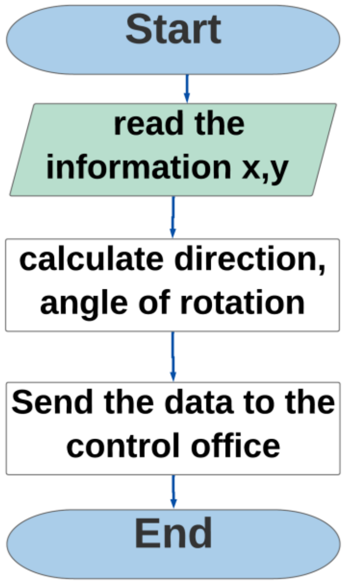
- **Obstacle detection algorithm**

Obstacle detection algorithm is for the safety of the robot, after each obstacle detection the robot stops and triggers a sound.



- **Positioning algorithm.**

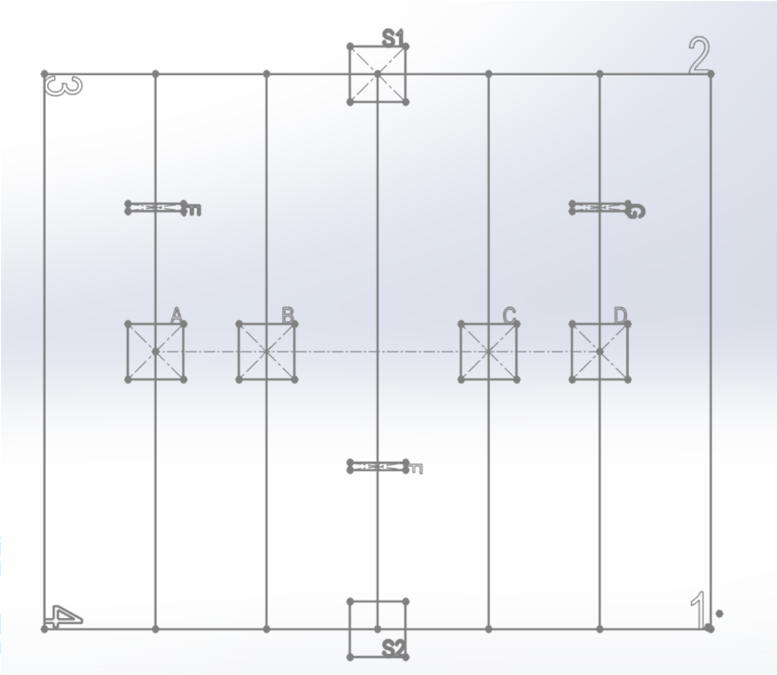
the positioning algorithm has the role of supervising the position of the robot from the control office



- **RFID and Navigation Determination.**

we propose the following model to detail the choice of scenario by RFID either in the empty tour or in the loaded tour.

Empty turn:



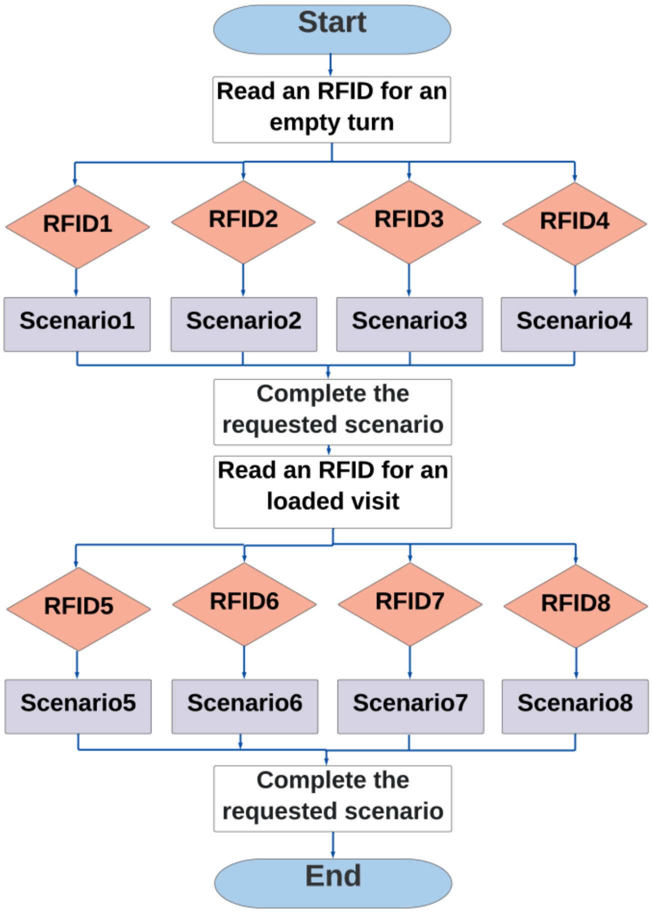
S1-3-4-S2-1-2-S1 scenario1
S1-2-1-S2-4-3-S1 scenario2

S2-1-2-S1-3-4-S2 scenario3
S2-4-3-S1-2-1-S2 scenario4

Loaded visit:

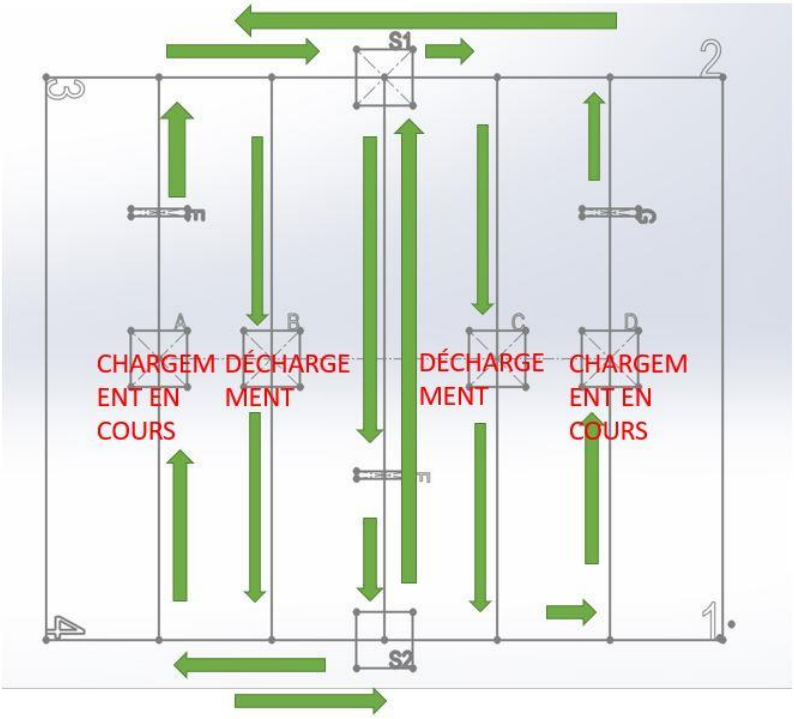
S1-F-S2-AE-S1-CDG-S1-B-S2-F-S1 scenario5
S1-F-S2-B-S1-GDC-S1-EA-S2-F-S1 scenario6

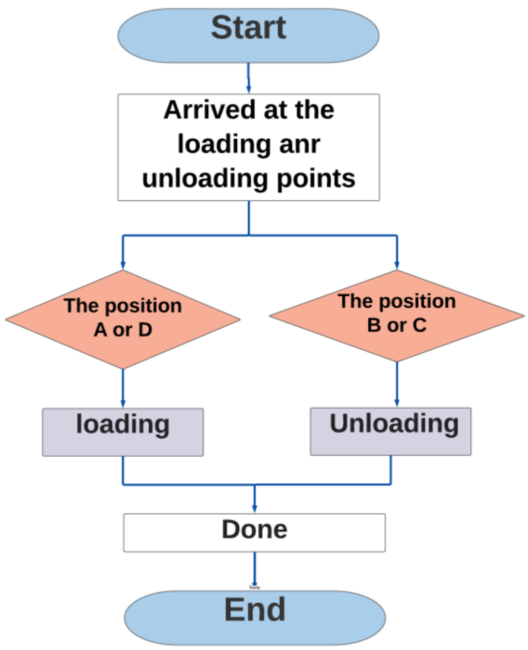
S2-F-S1-GD-S2-BEA-S2-C-S1-F-S2 scenario7
S2-F-S1-C-S2-AEB-S2-DG-S1-E-S2 scenario8



- **loading and unloading.**

We assume that we have points A and D for loading and points C and D for unloading, the robot must stop in these positions to do the task requested and then continue


S1
2
3
4
1
S2
A
B
C
D
CHARGEM
ENT EN
COURS
DÉCHARGE
MENT
DÉCHARGE
MENT
CHARGEM
ENT EN
COURS


Start
Arrived at the loading anr unloading points
The position A or D
The position B or C
loading
Unloading
Done
End

4.3.3. Software Design Process

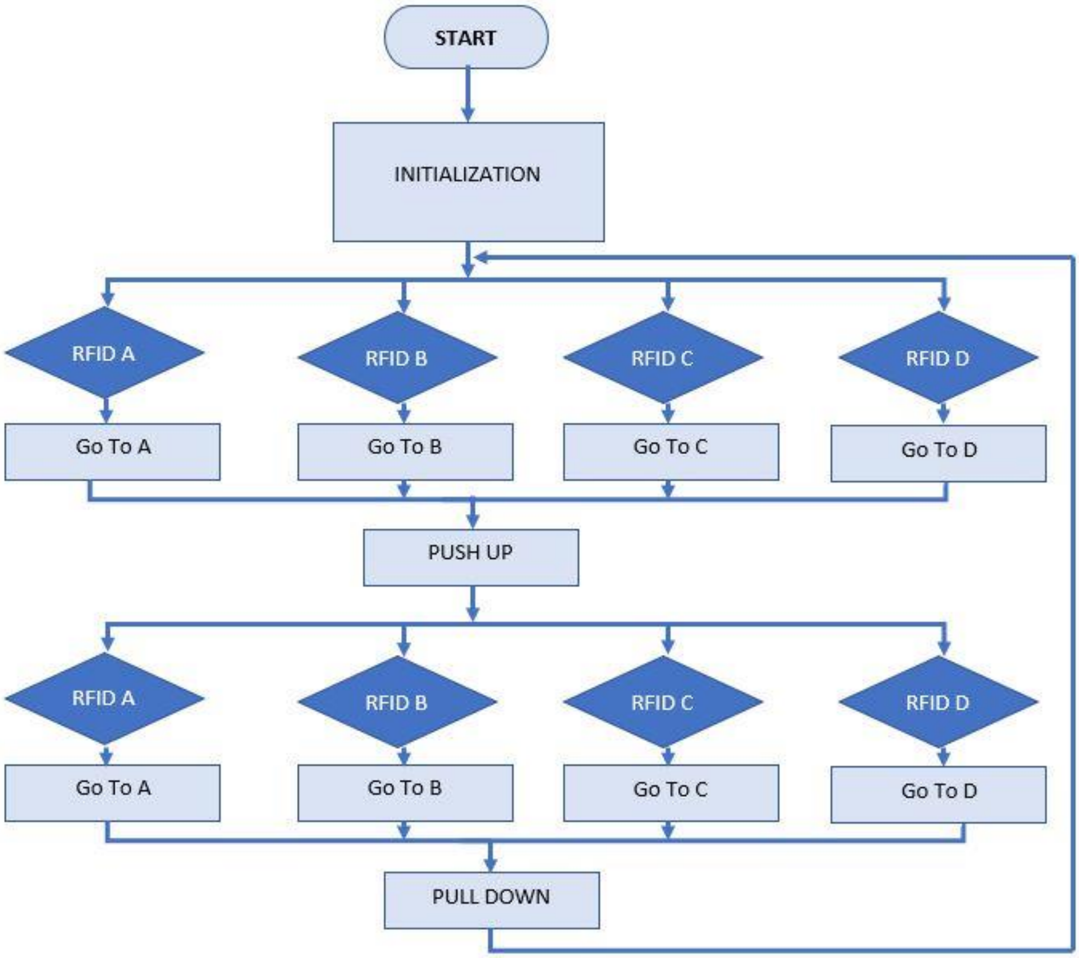


Figure 4.3:1 : Normal operation algorithm

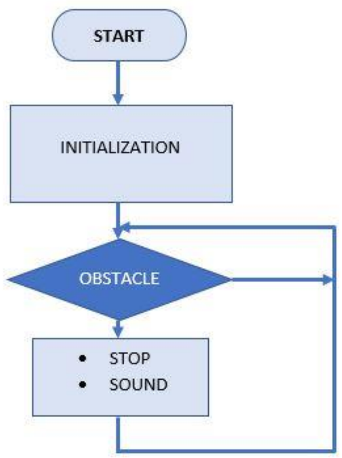


Figure** 4.3:2 **: Obstacle detection algorithm

Figure** 4.3:3 **: Line follower algorithm

the code is written in the ARDUINO software and here some part of our code

- **Main**

```
from __future__ import print_function
from flask import Flask, Response
from pyzbar import pyzbar
from picamera.array import PiRGBArray
from picamera import PiCamera
from datetime import datetime

import numpy as np
import cv2
import time


camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))
time.sleep(0.1)

app = Flask(__name__)


@app.route('/stream')
def stream():
    return Response(gen(),
                    mimetype='multipart/x-mixed-replace;
boundary=frame')

def gen():
    while True:
        frame = get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame +
b'\r\n\r\n')

def get_frame():
    camera.capture(rawCapture, format="bgr",
use_video_port=True)
    frame = rawCapture.array
    process_frame(frame)
    ret, jpeg = cv2.imencode('.jpg', frame)
    rawCapture.truncate(0)

    return jpeg.tobytes()

def process_frame(frame):
    decoded_objs = decode(frame)
    draw_positions(frame, decoded_objs)
```

```
def decode(frame):
    decoded_objs = pyzbar.decode(frame, scan_locations=True)
    for decoded_obj in decoded_objs:
        print(datetime.now().strftime('%H:%M:%S.%f'))
        print('Type: ', decoded_obj.type)
        print('Data: ', decoded_obj.data)

    return decoded_objs

def draw_positions(frame, decoded_objs):
    for decoded_obj in decoded_objs:
        left, top, width, height = decoded_obj.rect
        frame = cv2.rectangle(frame,
                              (left, top),
                              (left + width, height + top),
                              (0, 255, 0), 2)


if __name__ == '__main__':
    app.run(host="0.0.0.0", debug=False, threaded=True)
```

- **comms**

```
import os
import sys
import socket

class Fifo_Comms(object):
    def __init__(self, read_file, write_file,
read_block_size=50):
        self.read_block_size = read_block_size
     self.read_fifo = os.open(read_file, os.O_RDONLY |
os.O_NONBLOCK)
        self.write_fifo = os.open(write_file, os.O_WRONLY |
os.O_NONBLOCK)

    def __enter__(self):
        return self

    def __exit__(self ,type, value, traceback):
        os.close(self.read_fifo)
        os.close(self.write_fifo)

    def write(self, data):
        os.write(self.write_fifo, data)

    def read(self):
        bytes = []
        while True:
            try:
```

```
bytes.append(os.read(self.read_fifo,read_block_size))
            except OSERROR as exc:
                if exc.errno == errno.EAGAIN:
                    return bytes

class Socket_Comms(object):
    def __init__(self, socket_path, block_size=64):
        self.socket_path = socket_path
        self.socket_blocking = False
        self.block_size = block_size

    def __enter__(self):
        try:

            print("Opening Socket")
            # Create UDS Socket
            self.socket = socket.socket(socket.AF_UNIX,
socket.SOCK_STREAM)
            self.socket.connect(self.socket_path)
            self.socket.setblocking(0)
            print("Socket Open")
        except socket.error as err:
            raise err
        return self

    def __exit__(self ,type, value, traceback):
        print("Closing socket")
        self.socket.close()

    def write(self, data):

print("{:x}{:x}{:x}{:x}{:x}".format(data[0],data[1],data[2],dat
a[3],data[4]))
        self.socket.sendall(data)

    def read(self):
        data = []
        try:
            while True:
                data.append(self.socket.recv(self.block_size))
        except socket.error as err:
            raise err
        return data
```

control

```
import sys

import line_analysis
import fysom

from datetime import datetime
```

```
from compiler.ast import flatten

class Control_Parameters(object):
    def __init__(self,
                 forward_steps,
                 forward_speed,
                 reversing_steps,
                 reversing_speed,
                 turning_steps,
                 turning_speed,
                 finding_bend_steps,
                 finding_bend_speed,
                 finding_bend_threshold=10):
        def check_param(param):
            assert 320 >= param > 0, 'Param can only be in the
range 0-320'

        check_param(forward_steps)
        check_param(forward_speed)
        check_param(reversing_steps)
        check_param(reversing_speed)
        check_param(turning_steps)
        check_param(turning_speed)
        check_param(finding_bend_steps)
        check_param(finding_bend_speed)
        assert 0 < finding_bend_threshold < 30

        self.forward_steps       = forward_steps
        self.forward_speed       = forward_speed
        self.reversing_steps     = reversing_steps
        self.reversing_speed     = reversing_speed
        self.turning_steps       = turning_steps
        self.turning_speed       = turning_speed
        self.finding_bend_steps  = finding_bend_steps
        self.finding_bend_speed  = finding_bend_speed
        self.finding_bend_thresh = finding_bend_threshold

class Control(object):

    LEFT = 1
    RIGHT = 2

    def on_init(self, e):
        pass

    def on_out(self, e):
        direction = e.args[0]
        lines = e.args[1]
        # Can't see any lines
        if direction is None:
            direction = self.last_turn
```

```
            # self.backwards_turn(direction)
            # self.current_order = (self.motion.backward,
            #
(self.params.reversing_steps,
            #
self.params.reversing_speed))
            self.fsm.ev_reversing()
        elif direction is Control.LEFT:
            self.current_order = (self.motion.rotate_left,
                                  (self.params.turning_steps,
                                   self.params.turning_speed))
            self.last_turn = direction
        else:
            self.current_order = (self.motion.rotate_right,
                                  (self.params.turning_steps,
                                   self.params.turning_speed))
            self.last_turn = direction

    def on_reversing(self, e):
        self.current_order = (self.motion.backward,
                              (self.params.reversing_steps,
                               self.params.reversing_speed))

    def on_perturbing(self, e):
        if self.last_turn == Control.LEFT:
            self.current_order = (self.motion.rotate_left,
                                  (self.params.turning_steps,
                                   self.params.turning_speed))
        else:
            self.current_order = (self.motion.rotate_right,
                                  (self.params.turning_steps,
                                   self.params.turning_speed))

    def on_in_top(self, e):
        self.current_order = (self.motion.forward,
                              (self.params.forward_steps,
                               self.params.forward_speed))

    def backwards_turn(self, direction):
        if direction is Control.LEFT:
            self.current_order = (self.motion.specific_move,
                                  (self.motion.BACKWARD,

self.params.finding_bend_steps,

self.params.finding_bend_speed,
                                   self.motion.FORWARD,

self.params.finding_bend_steps,

int(self.params.finding_bend_speed*0.2)))
```

```
                self.last_turn = direction
            else:
                self.current_order = (self.motion.specific_move,
                                      (self.motion.FORWARD,

self.params.finding_bend_steps,

int(self.params.finding_bend_speed*0.2),
                                       self.motion.BACKWARD,

self.params.finding_bend_steps,

self.params.finding_bend_speed))
                self.last_turn = direction

    def on_in_bottom(self, e):
        direction = e.args[0]
        print("Moving in old direction {0}".format(direction))
        self.backwards_turn(direction)

    def __init__(self, motion, parameters):
        #self.log_file = open('/root/log.file', 'w')
        self.motion = motion
        self.params = parameters
        self.current_order = (None, (None, None))
        self.last_turn = Control.RIGHT
        self.fsm = \
            fysom.Fysom(
                {'initial': 'init',
                 'events':
                     [{'name': 'ev_in_top',        'src':
'init',         'dst': 'in_top'},
                      {'name': 'ev_in_bottom',     'src':
'init',         'dst': 'in_bottom'},
                      {'name': 'ev_out',           'src':
'init',         'dst': 'out'},
                      {'name': 'ev_in_top',        'src':
'out',          'dst': 'in_top'},
                      {'name': 'ev_in_bottom',     'src':
'out',          'dst': 'in_bottom'},
                      {'name': 'ev_out',           'src':
'out',          'dst': 'out'},
                      {'name': 'ev_reversing',     'src':
'out',          'dst': 'reversing'},
                      {'name': 'ev_in_top',        'src':
'in_top',       'dst': 'in_top'},
                      {'name': 'ev_in_bottom',     'src':
'in_top',       'dst': 'in_bottom'},
                      {'name': 'ev_out',           'src':
'in_top',       'dst': 'out'},
```

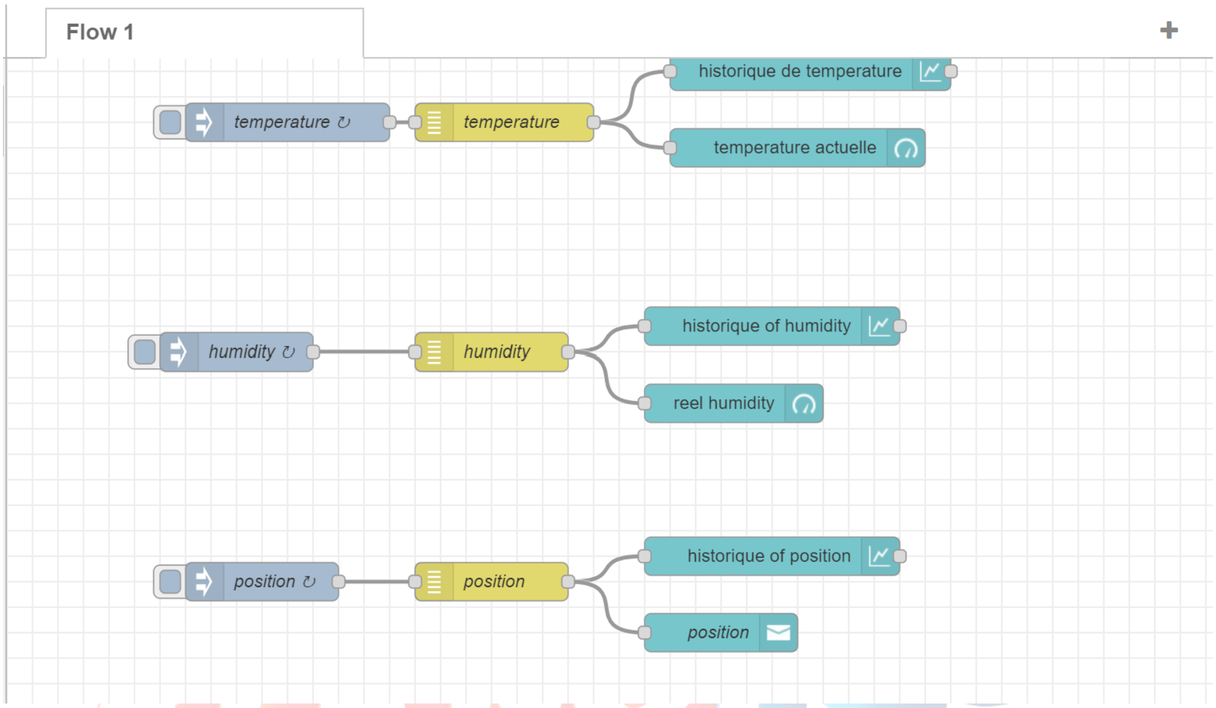
```
                      {'name': 'ev_in_top',        'src':
'reversing',    'dst': 'perturbing'},
                      {'name': 'ev_in_bottom',     'src':
'reversing',    'dst': 'perturbing'},
                      {'name': 'ev_out',           'src':
'reversing',    'dst': 'reversing'},
                      {'name': 'ev_in_top',        'src':
'perturbing',   'dst': 'in_top'},
                      {'name': 'ev_in_bottom',     'src':
'perturbing',   'dst': 'in_bottom'},
                      {'name': 'ev_out',           'src':
'perturbing',   'dst': 'out'},
                      {'name': 'ev_in_top',        'src':
'in_bottom',    'dst': 'in_top'},
                      {'name': 'ev_in_bottom',     'src':
'in_bottom',    'dst': 'in_bottom'},
                      {'name': 'ev_out',           'src':
'in_bottom',    'dst': 'out'}],
                  'callbacks':
                      {'onperturbing'   : self.on_perturbing,
                       'onreversing'    : self.on_reversing,
                       'oninit'         : self.on_init,
                       'onout'          : self.on_out,
                       'onin_top'       : self.on_in_top,
                       'onin_bottom'    : self.on_in_top}})

        def log(e):
            #self.log_file.write('event: %s, src: %s, dst: %s,
time:%s\n' % (e.event, e.src, e.dst, datetime.now()))
            #self.log_file.flush()
            print 'event: %s, src: %s, dst: %s, motion:%s' %
(e.event, e.src, e.dst, self.current_order[0])
        self.fsm.onchangestate = log

    def find_centre_line(self, lines):
        intersections = [intersect for intersect in [[line for
line in row if line[0] <= 52 and line[1] >= 48] for row in
lines.values()] if len(intersect) > 0]
        if(len(intersections) > 0):
            return intersections[0]
        return None

    def find_closest_line(self, lines):
        closest_lefts =  flatten([intersect for intersect in
[[line[1] for line in row if line[1] <= 50] for row in
lines.values()] if len(intersect) > 0])
        closest_rights = flatten([intersect for intersect in
[[line[0] for line in row if line[0] >= 50] for row in
lines.values()] if len(intersect) > 0])

        len_left = len(closest_lefts)
```

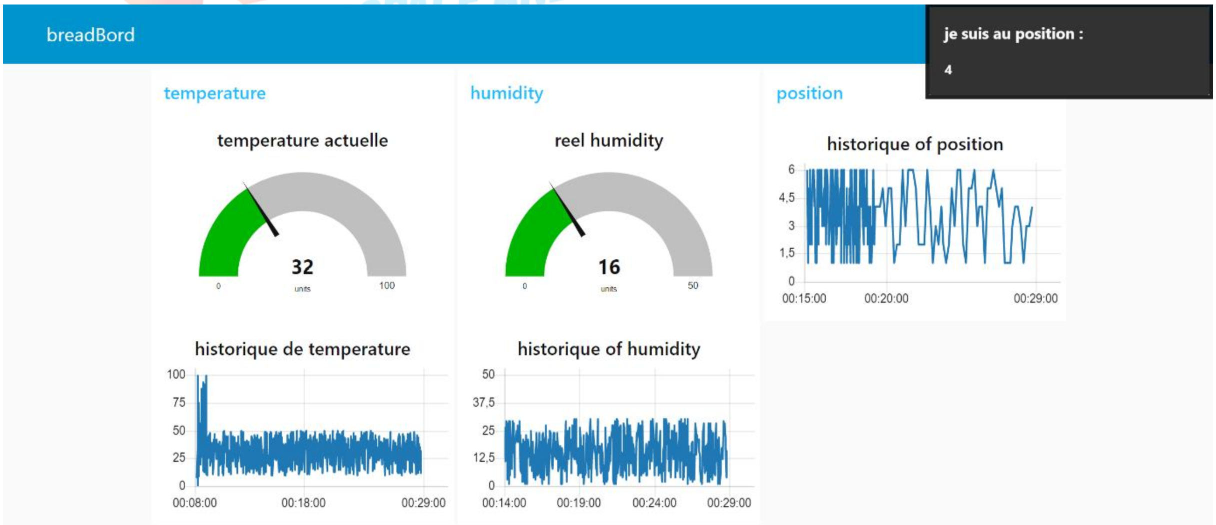
```
        len_right = len(closest_rights)

        if len_left == 0 and len_right == 0:
            return None
        if len_left > 0 and len_right == 0:
            return Control.LEFT
        if len_left == 0 and len_right > 0:
            return Control.RIGHT
        print 'closest lefts then rights'
        print closest_lefts
        print closest_rights
        if((50 - max(closest_lefts)) > (min(closest_rights) -
50)):
            return Control.RIGHT
        else:
            return Control.LEFT

    def progress(self, lines):
        centre_line = self.find_centre_line({item:val for
(item,val) in lines.items() if item >= 0})
        if centre_line is None:
            print {item:val for (item,val) in lines.items() if
item < 0}
            centre_line = self.find_centre_line({item:val for
(item,val) in lines.items() if item < 0})
            if centre_line is None:
                print("OUT")
                closest_line = self.find_closest_line(lines)
                self.fsm.ev_out(closest_line, lines)
            else:
                print("IN BOTTOM")
                self.fsm.ev_in_bottom(self.last_turn, lines)
        else:
            print("IN TOP")
            self.fsm.ev_in_top(centre_line, lines)

        self.current_order[0](*self.current_order[1])
```

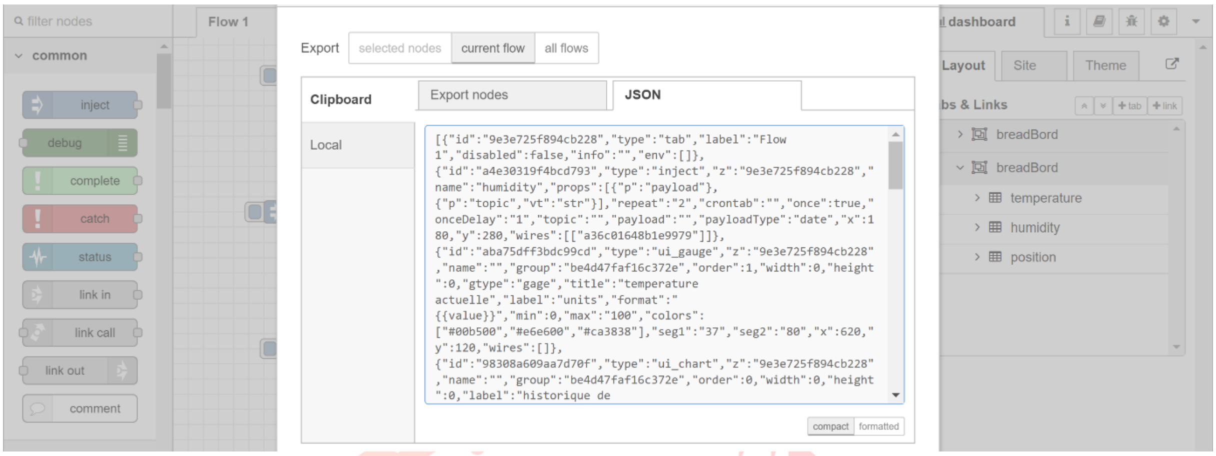
4.4. Dental Interfaces

Our breadboard represent all of information we need about our car like the actual position in the map, his tempter and humidity. The dashboard is developed by using JavaScript and Node-Red.

This breadboard is this first part of the final breadboard. This last one continent the roadmap of the car and illustrate the actual position in 3D and we hope to achieve the final result







5. SAFETY

Effective and reliable workplace security is very important to any business because it reduces insurance, compensation, liabilities, and other expenses that the company must pay to its stakeholders, ultimately leading to increased business revenue and a reduction in operational charges incurred.

For safety, and to respect the norms of security, our device is going to have an emergency stop pushbutton, located in a very clear and visible location, and also the exterior design and materiel is not going to be dangerous or sharp in case the robot bumped into a worker. And also, we are going to include a weight sensor, so the vehicle doesn’t carry more that its capacity; and an obstacles sensor so it does not bump in someone or something, and finally a heat sensor to alarm any concerning changes of temperature in the robot or near it. All of that plus an alarm system to notify the workers. It's also very important to take some safety measures while the devices are working like:

- Remove all obstructions from the robot work area.
- Make sure that the robot is not carrying anything at the start point.
- Check for signs of damage to the robot.
– Remove loose fitting clothing (ties, scarves, sleeves, etc.)
- Locate the emergency stop pushbutton.
- Be Aware of Your Surroundings

Also, to make sure that those Precautions are respected, we are going to write a list of instruction, print it, and put it in a place were everybody can see it.

For safety, and to respect the norms of security, our device is going to have an emergency stop pushbutton, located in a very clear and visible location, and also the exterior design and materiel is not going to be dangerous or sharp in case the robot bumped into a worker.

And also, we are going to include a weight sensor, so the vehicle doesn't carry more that its capacity; and an obstacles sensor so it does not bump in someone or something, and finally a heat sensor to alarm any concerning changes of temperature in the robot or near it. All of that plus an alarm system to notify the workers.

It's also very important to take some safety measures while the devices are working like:

- **Remove all obstructions from the robot work area.**
- **Make sure that the robot is not carrying anything at the start point.**
- **Check for signs of damage to the robot.**
- **Remove loose fitting clothing (ties, scarves, sleeves, etc.)**
- **Locate the emergency stop pushbutton.**

6. TEST

here are the links of the code tests on a mini robot and the reaction against accidents

also, there is a small animation concerning the components of our robot.

- **animation in blender:**

https://drive.google.com/file/d/1s-OuTNW4yUSEsvar6zQheOS6g7CDBpTG/view?usp=drivesdk

- **simulation on ANSYS:**

https://drive.google.com/file/d/1qjY0vZCA57ksAVfiYSrMOMZSX16d9e7j/view?usp=drivesdk

- **code testing:**

https://drive.google.com/file/d/1raSO8aDHDiio1ApCjt6By1IOv03Z3_GR/view?usp=drivesdk

https://drive.google.com/file/d/1qy5Vy-u2-fM_lafC3izYRcI8KiA1Kflq/view?usp=drivesdk

- **platform testing:**

https://drive.google.com/file/d/1rBJ0C8pNAP9H_hmxOYqUXXs5qDmKG1ci/view?usp=drivesdk

7. EXPERIENCE

While doing the numeric simulation we come to the conclusion that some parts are more likely going to get ruptured as a consequence of the charges applied by the weight of the platform and the loads, so added more support to the longitudinal chassis.

8. TIME, BUDGET AND RISK PLANNING

Manufacturing and raw material budget:

To manufacture the robot in the workshop of our establishment we need several steel sheets, assembly accessories and other materials

- **for the aesthetic part.**

The estimate of the maximum load: 100$

- **For the electronic part:**

The estimate of the maximum load: 500$

The maximum budget to build the robot is 600$

- **Time**

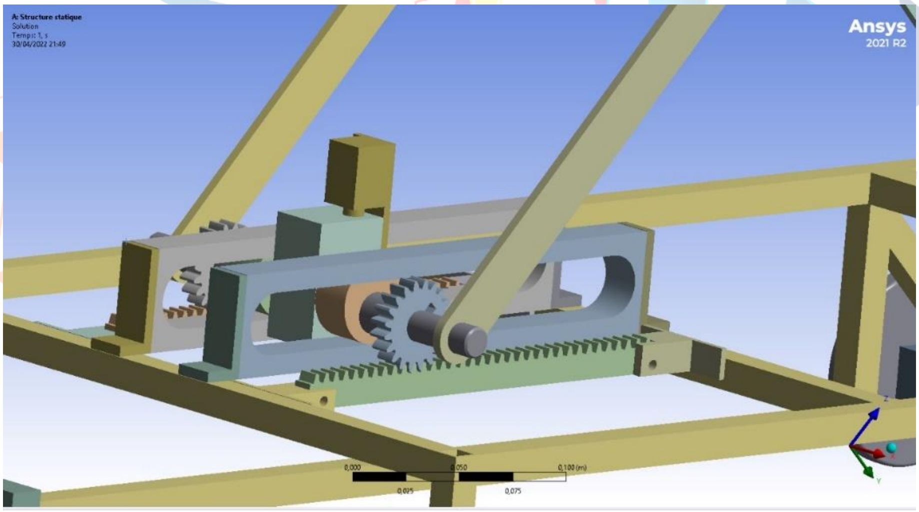
The maximum time to manufacture and manage the code of our robot is estimated in 45 days The work will have to be separated on 2 teams.

1st team will have the task of manufacturing the chassis, bodywork, and frames

2nd will be responsible for mounting electrical components.

9. AUTHENTICITY

robots are widely used nowadays in factories, with different purposes, and also there is a variety of solutions and configurations that goes in the making of them. so, it took our team some time and research to come up with a unique and an authentic design and aspect to differentiate our project from other projects.
First of all is the external design, our robot has unique design, that make it movement flexible and it give it the access to difficult areas while lifting the platform.
Also, for translation system, that makes the table go up to lift the platform and the loads, we did come up with a new solution that uses totally a mechanical system (explained in details in other sections) to reduce the consumption of the energy



10. NATIONALITY

Some of the materials in our project are of national production. The main reason we prefer this is the ease of sourcing our domestic products and the fact that we have taken care to ensure that the materials to be used in our vehicle are domestic. Products like Arduino (Clone), strain relief, motor driver are made in Turkey. We will also make our own special motor driver. This motor driver will give each motor a maximum of 24 volts and 6 amps by connecting 6 motors. Also, not all the battery power can be given to some materials. For this reason, we are going to make a DC/DC converter. We will print the PCB designs that we will make on CNC and solder the necessary components. The goal here is to increase the rate of locality in the vehicle with the parts that we produce.

11. References

https://www.noor-book.com/en/ebook-Guide-du-dessinateur-industriel-Chevalier-pdf
https://hal.archives-ouvertes.fr/cel-01636674/file/POLY-SAPH113-Mat%C3%A9riaux.pdf
https://www.thierry-lequeu.fr/data/RAP-LALLEMAND-VIDAL.pdf
https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/techniques/644/644-121-p17.pdf
https://www.youtube.com/channel/UCfgTefz2tJkvGHDXbW9cKcw/playlists
https://www.youtube.com/c/IncroyablesExp%C3%A9riencesvik95170/videos
https://www.youtube.com/user/lesbonsprofs/featured
https://www.youtube.com/c/SketchUp/playlists
https://www.youtube.com/channel/UCG96dhvGgkeTdAW1DKb9iJg/videos
https://www.youtube.com/channel/UCVC0zvnvYEX2ae7yL7EZFtQ/videos GMP
https://www.youtube.com/watch?v=A0UFqBF5gFU
moteur https://www.youtube.com/watch?v=S_j1sSFJqdM
https://www.pdfprof.com/PDF_Documents_Telecharger_Gratuits_PDF.php?PDF=23631-analyse+fonctionnelle+d%27un+robot+suiveur+de+ligne
https://www.pdfprof.com/PDF_Documents_Telecharger_Gratuits_PDF.php?PDF=23631-analyse+fonctionnelle+d%27un+robot+suiveur+de+ligne
https://inis.iaea.org/collection/NCLCollectionStore/_Public/39/082/39082365.pdf
https://www.pdfprof.com/PDF_Documents_Telecharger_Gratuits_PDF.php?PDF=23631-analyse+fonctionnelle+d%27un+robot+suiveur+de+ligne
https://slideplayer.fr/slide/1296233/
https://docplayer.fr/38414869-Mini-projet-isi-robot-de-piscine-analyse-fonctionnelle.html
https://fr.slideshare.net/azizshadowstokes/2743557-dossierpperobotsuiveurdeligne
https://www.robot-maker.com/forum/tutorials/article/50-choisir-et-simuler-un-moteur-pour-votre-robot/
https://www.parkem.ch/produkte/Parker%20servo_control/193_062001_Kit_motors_catalog_FR.pdf
https://ph.parker.com/us/en/frameless-low-voltage-servo-motors-k-series-ce-certified#
https://en.wikipedia.org/wiki/CATIA
https://en.wikipedia.org/wiki/Ansys

https://ferabeton.com/prix/tole-acier-plaque-acier-tole-plane/

https://nadaf.ma/fr/46-roues-assemblage