ENGINEERING JOURNAL
WORLD ROBOT OLYMPIAD
FUTURE ENGINEERS

TEAM NAME: IOTRONIC POLIVIRTUAL - LEOTRONICS

TEAM CITY: QUERÉTARO, MÉXICO

TEAM ACADEMY: IOTRONIC ACADEMY

TEAM SCHOOL: CECYT 9 IPN / ANÁHUAC QRO.

COMPETITION PROGRAM: FUTURE ENGINEERS

CAR NAME: LOTUS ATOM X-BOW

MEMBERS: 3

ROSTER:

- **PROGRAMMER:** LUIS RICARDO CHAVARRÍA ARREOLA - IPN
- **MECHANIC:** LUIS LEONARDO CHAVARRÍA ARREOLA – IPN / ANÁHUAC
- **ELECTRONIC:** LUIS ENRIQUE RAMÍREZ CHAVARRÍA - IPN

TABLE OF CONTENTS

Note: Bold letters contain the most important information for the Engineering Journal Evaluation

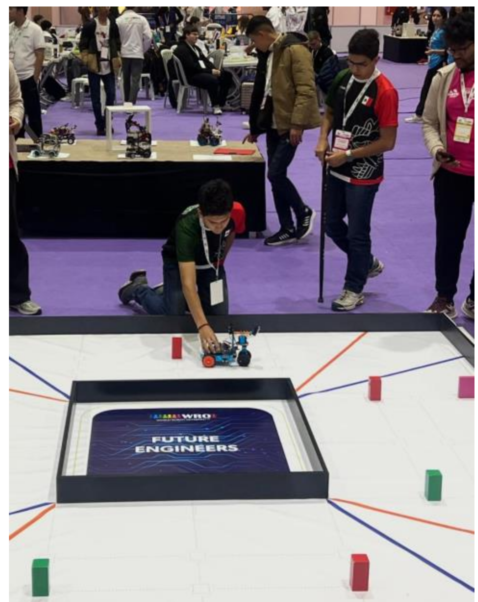
1. TEAM ORGANIZATION PLAN

1.1. SLOGAN (TEAM MOTTO)

Persist, lead the change and you will achieve the unimaginable!
The magic is believing in yourself

1.2. WHO ARE WE?

We are the **IoTronic Polivirtual - Leotronics team**, we are a team made up of 3 students, and as a fun fact, we are also family. Additionally, our team includes many more people, as our sponsors, teachers, coaches, mentors, and parents have helped us grow day by day, which makes us even stronger. This will be our second season in WRO. We have been champions in various categories around the world, playing a significant role at both national and international levels.



1.3. OBJECTIVES

Our objectives are:

- **Learn**: Learning is the **number one foundation for achieving any objective**, which is why we have studied, prepared ourselves, and acquired the tools to complement all our knowledge.
- **Teach**: We seek to teach others, especially about this world of **STEAM**, as it is essential to **continue growing** in what makes us happy.
- **Inspire**: We are not only seeking victory; just as we love to learn and teach, it goes hand in hand with inspiring others, **especially the new teams in the STEAM community**, showing them that it is possible. Everything you set your mind to can be achieved, that discipline and knowledge can be fun, and that **everyone has the potential to win**. **We want to inspire people, students, Querétaro, all of Mexico**, and make our team more than just a robotics team, but a movement of inspiration for all young people in the world.

2. (PROJECT PLANNING)

The Planning of this Project **has 3 phases**, which are: **Scope Plan, Team Plan** (Resources y Knowledge) and **Project Control Plan.**

2.1. SCOPE PLAN

In the Scope Plan, the following steps were determined: **1. Gather Requirements, 2. Define the Scope, 3. Create the Work Breakdown Structure**

Scope definition: Construction of a robot with mechanical-electrical systems and programmable electronic systems to perform autonomous tasks on a competition track measuring 3.2 m x 3.2 m.

Work Breakdown Structure required for the project:

1. **Empathize with the competition**
2. **Design**
3. **Build**
4. **Inspection and testing**
5. **Documentation**

2.2. TEAM PLAN

Within the Team Plan, **Resources and Knowledge Will me managed**.

2.2.1. RESOURCE MANAGEMENT

Resource Management is classified into Human Resources, Technological Resources, and Material Resources.

- **HUMAN RESOURCES– TEAM INTEGRATION**

Everything started in September 2021, when my brother and I (Leonardo and Ricardo) saw a robotics tournament on television and told our dad that we liked watching the event and that one day we wanted to go to that type of tournaments. After that my dad enrolled us in the Mexican School of Electricity so we could learn Basic Electricity and Electronics, and we trained every Saturday from September 4, 2021, to August 20, 2022. At the end of the course, my brother and I realized that we had developed technological skills, **so we decided to create a team to compete in robotics**.

My dad was searching for a **Technology Coach** and managed to find Fernando Juárez, who has experience in educational robotics tournaments. It's worth mentioning that my **dad** is our **Engineering Coach**, and my **mom** is the **Team Administrator**, who has taken care of everything necessary (uniforms, a practice space, managing trips, and logistics in general) to prepare us for the competition.

In August 2023 we decided to invite **our cousin Luis Enrique** so he could participate with us, since his skills in electronic topics complement us.

The team was organized as follows:

Robotics engineers:

- Luis Ricardo – Programming Technician
- Luis Leonardo – Mechanical Technician
- Luis Enrique – Electronics Technician

Coaches

- Fernando Juárez – Robotics
- Luis Chavarria – Engineering

Administration

- Deisy Arreola - Administration

Additionally, the participants will have the following administrative responsibilities:

- Luis Ricardo – Engineering Development
- Luis Leonardo – Project Planning and Control
- Luis Enrique – Development of the Testing Plan

- **TECHNOLOGICAL RESOURCES**

The following are the technological tools that were used and the associated activities:

Solid Works – Mechanical Design

Github – Code Control

Clickup – Project Control

MS Project – Project Timeline

MS Visio – Control Architecture

MS Excel – Calculation Reports

- **MATERIAL RESOURCES**

We have **facilities equipped** with the material resources for the development of robotics projects.

At **IOTRONIC Academy**, which has a 3D Resin Printer, a 3D PLA Printer, a Laser Cutter, a Lathe, a Milling Machine, an Electronics Laboratory, and a WRO Competition Track.

2.2.2. KNOWLEDGE MANAGEMENT

Knowledge Management covers the following activities: **1. Knowledge Assimilation** (Research, Training, Coaching, Mentoring), **2. Knowledge Transfer** (Knowledge Sharing, Lessons Learned), **3. Information Storage and Accessibility** (Recording and Consultation), 4. Innovation (Incorporating Knowledge)

As part of **Knowledge Assimilation**, the following activities were used as **Study Methods**:

a) Research on Robotics Trends, WRO Competitions, Mobility Technologies, and Computer Vision Technologies.
b) Training in SolidWorks, Digital Systems, and Computer Vision.
c) Technological coaching
d) Competition mentoring
e) Watch videos of WRO tournaments. (YouTube)
f) Use WRO Future Engineers Getting Started
g) Use Q&A Forum WRO

As part of **Knowledge transfer** we conducted:

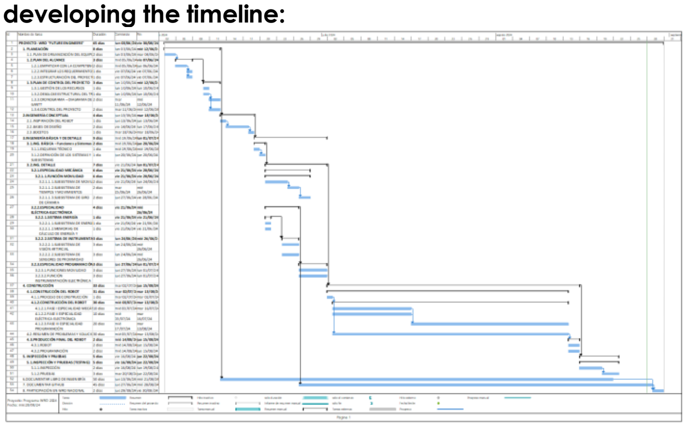
a) Interaction among team members
b) Work meetings
c) Lessons learned

As part of **Information Storage and Accessibility**, we use the IOTRONIC Drive.

2.3. PROJECT CONTROL PLAN

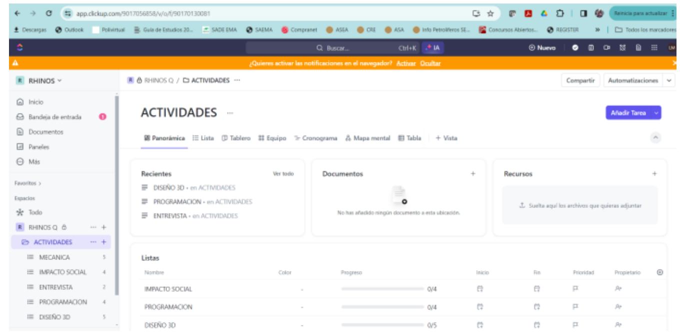
Within the Project Control Plan, the following phases are considered: **1. Define the Activities, 2. Sequence the Activities and Develop the Timeline, 3. Control the Project.**

Defining the activites, sequencing the activities and developing the timeline:



Project Control:

To ensure that a project is successful, it is important to have a comprehensive and detailed view of all project activities, especially the tasks and responsibilities of each person. For this, we use a software that helps us maintain proper Project Control.



Commitment dates are established based on the work schedule, and deliverables are defined, which can be uploaded as files or images.

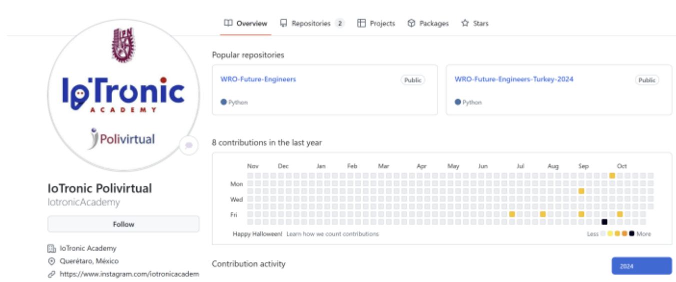
Project Documentation:

GitHub was used to document the project, and GitHub is an online platform that allows users to store, share, and collaborate on software projects. It is an open-source community that has become a hub for software development.

Among GitHub's features are:

- Store codes in repositories
- Share and present work
- Manage and track changes in the code
- Exchange code files
- Work on collaborative projects
- Stablish profesional relationships
- Crear un perfil y marca para promocionarse

GitHub is based on Git, a version control system that allows software developers to keep track of the changes made to the code.



3. PROJECT EXECUTION

The Execution of this Project has **4 phases**, which are: **1. Empathize with the** Competition (Rules and Missions), **2. Robot Design** (Conceptual Engineering, Basic Engineering, and Detail Engineering), **3. Robot Construction** (Evolution of the Robot, Identifying Problems and Solutions, Final Production of the Robot), and **4. Inspection and Testing** (Inspection and Testing)

3.1. EMPATHIZE WITH THE COMPETITION

In this phase, we must understand correctly the project's scope, the competition's objective, and its rules. The competition will be described next:

Misions/Challenges

The challenges for autonomous vehicles this season are Time Attack races: there will not be multiple cars on the track at the same time. Instead, one car will attempt to achieve the best time by completing several laps fully autonomously.

The two challenges are the following:

Open challenge: The vehicle has to complete three (3) laps on the track with random locations in the inner walls of the track.

Obstacle challenge: The vehicle has to complete three (3) laps in the track with randomly placed red and green traffic signals. The traffic signals indicate which side of the lane the vehicle should follow. The red pillar is the traffic signal to stay in the right side of the lane while the green pillar is the traffic signal to stay on the left side of the lane. The continuation of the vehicle to the third lap is indicated by the last traffic signal of the second lap. A green traffic signal indicates that the robot must proceed and continue the third lap in the same direction. A red traffic signal indicates that the vehicle must turn around and complete the third lap in the opposite direction. The vehicle must not move any of the traffic signals. After the robot completes the three laps, it must find the parking area and perform a parallel parking maneuver.

The initial direction in which the car must drive on the track (clockwise or counterclockwise) will vary in different challenge rounds. The initial position of the car, as well as the number and location of the traffic signals, are randomly defined before the round (after the verification time).

The competition is based on 2 Challenge Rounds: the Open Challenge will last 3 minutes, and the Obstacle Challenge will also last 3 minutes.

Track dimension

The competition track has the following dimensions (3.2m x 3.2m) and specifications:

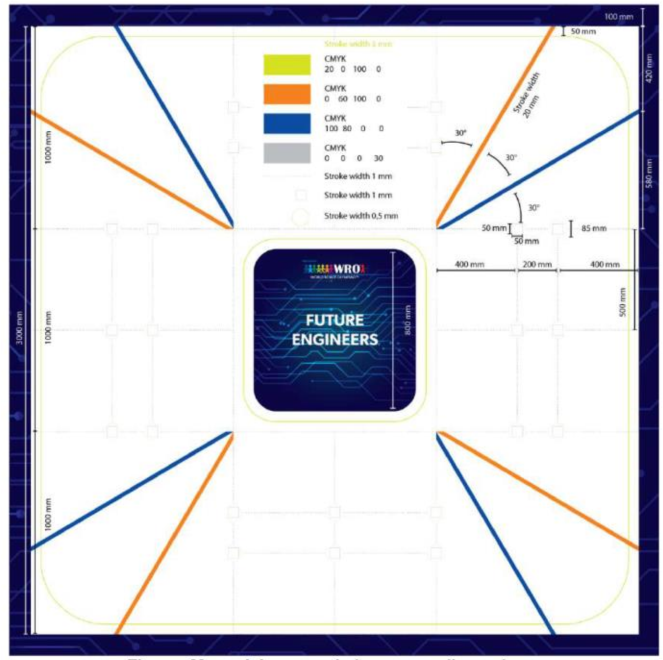


Figura: Mapa del campo de juego con dimensiones

Signaling

The track will also include the following traffic signals:

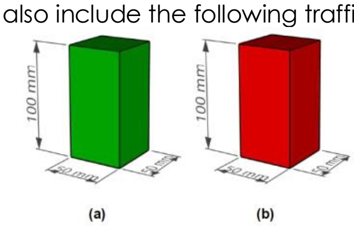
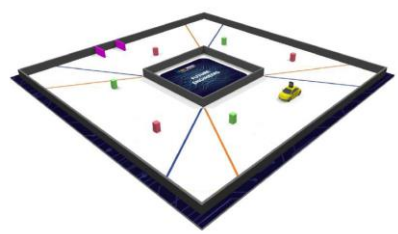


Figura: Dimensiones de las señales de tráfico

Playing field

The following figure shows the playing field with the game objects



The playing field represents a racetrack where traffic signals (represented by colored obstacles - pillars) are placed. The track consists of eight sections: four corner sections and four straight sections.

The corner sections are marked with red dashed lines in the following figure.

The straight sections are marked with blue dashed lines.

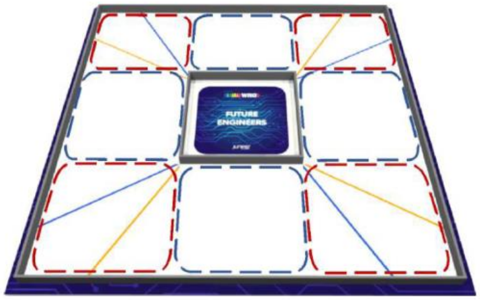


Figura: Diferentes tipos de secciones en el campo de juego

Each straight section is divided into 6 zones. The six internal zones within the section are for the starting position of the car. Four T-intersections and two X-intersections are used to position the traffic signals. The places where traffic signals can be placed are called traffic signal seats.

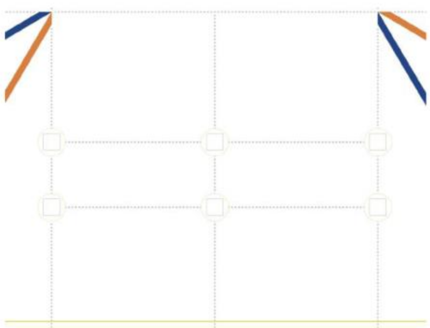


Figura: Zonas y asientos de señales de tráfico en la sección recta

It is possible to place a parking area in one of the straight sections. The width of the parking area is always 20 cm. The length is variable and is calculated as 1.25 times the length of the robot.
The parking area is bounded by two wooden elements with dimensions of 20 cm x 2 cm x 10 cm in magenta. The right element is placed right next to the dashed line. The position of the left element is defined as described above..

Figura: Definición del tamaño del estacionamiento

Scoring

	Requirements	Point value	Total available
1.	**Driving Open and Obstacle Challenge**		
1.1.	The vehicle drives from a section in the challenge driving direction. This is applicable for the starting section, but not applicable for the finish section and other section next after it.	1	24
1.2.	The vehicle drives a full lap. 8 sections were passed successfully in the challenge driving direction. The starting section is included in the eight sections for the first lap. The lap is considered as completed if the vehicle completely drives out of the last (corner) section in the lap. So, the vehicle can start moving in the opposite direction after this and the lap will be still considered.	1	3
1.3.	After the completion of three laps the vehicle stopped in the finish section.	3	3
	Additional points for Obstacle Challenge rounds:		
	Not completed three laps		
1.4	One or more traffic signs were moved. Vehicle must complete at least one round to qualify for score.	2	2
1.5.	The traffic signs were not moved. Vehicle must complete at least one round to qualify for score.	4	4
	After the completion of three laps		
1.6	One or more traffic signs were moved.	8	8
1.7	**No** traffic signs were moved.	10	10
1.8	Final lap completed in the correct direction	15	15
1.9.1	Parking successfully (completely in the parking area)	15	15
1.9.2	Parking partly in the parking area	7	7
2.	The team performed repairing actions by taking the vehicle out of the field even if the actions were not successful.	Total round points divided by factor 2	
3.	**Engineering journal and vehicle documentation** Refer to appendix C for a breakdown of the engineering journal scoring.		30

Practice and Verification Area

Within the competition, there will be practice time that can take place between rounds

After each practice time, there will be a vehicle verification period to check the requirements.

Each team must work during the practice time in their designated place until the verification time, at which point the team's vehicle must be placed in a designated area (verification area).

3.2. ROBOT DESIGN

To develop the design, a sequencing of activities must be carried out, starting with **Robot Inspiration**, followed by **Conceptual Engineering**, then **Basic Engineering**, and finally **Detail Engineering**.

In the **Conceptual Engineering** the objective of the Project is defined and the **Design Bases** are established (which includes requirements, regulations, and the creation of sketches).

In the **Basic Engineering** the design of the **Mechanical-Electrical-Electronic System** and the **Programming System** is developed.

In the **Detail Engineering** the **System Integration Architecture**, **Calculation Reports**, and **Programming** are developed.

Both the Basic Engineering phase and the Detail Engineering phase are divided into the following functions:

- **Mobility Management**
- **Power and Sensor Management**
- **Obstacle Management**

Technical principles

The technical principles on which we develop our Robot are:

a) **Mechanical:** It is responsible for the accessories, parts, or mechanical equipment with which our main structure (chassis) will be built, as well as the complementary infrastructure that will serve the purpose of mobility.
b) **Electrical:** It is responsible for the dynamic equipment or motors that will enable the mechanical mobility functions. It also considers the type of electrical power supply and batteries.
c) **Electronics:** It is based on the sensors, cameras, or actuators that will be used to detect or activate mechanisms. It also considers the connections to the brain of the Robot (the processor) through the input and output ports.
d) **Programming:** It is based on the programming routine necessary for the Robot to perform specific tasks automatically to overcome obstacles.

3.2.1. ROBOT INSPIRATION

The name of our robot is **"LOTUS ATOM X-BOW"** and this name was inspired by our favorite cars.

3.2.2. CONCEPTUAL ENGINEERING

The objective of the project is to build a Competition Robot for the FUTURE ENGINEERS Program.

3.2.2.1. Design bases

In this section, all regulatory constraints must be identified, which will serve as a basis for defining the Vehicle Design, taking into account the requirements of the missions. The identification of the constraints will be carried out in 2 Systems: 1. Mechanical-Electrical-Electronic Design and 2. Programming Design.

1. Mechanical-Electrical-Electronic System Design: only 3D printed elements, elements prepared with a CNC machine, elements cut from acrylic/wood/metal, or any element made of any material may be used. The following must be met:"

- The dimensions of the vehicle must not exceed 300x200 mm and 300 mm in height.
- The weight of the vehicle must not exceed 1.5 kilograms.
- The vehicle must be a 4-wheeled vehicle with a drive axle and a steering actuator of any type. It must be front-wheel drive, rear-wheel drive, or all-wheel drive. Vehicles with a differential wheel base are not allowed.
- The use of omnidirectional wheels or spherical wheels is not allowed
- The vehicle must be autonomous and complete the 'missions' on its own. Radio communication systems, remote control, or wired control systems are not allowed while the vehicle is in operation.
- DC electric motors and/or servos from any brand may be used.
- A maximum of two motors is allowed to make the vehicle move forward or backward (i.e., to drive the robot; these are the drive motors). The drive motors must be connected directly to the axle that rotates the wheels or indirectly through a gear system. The two drive motors cannot be independently connected to the drive wheels.
- Teams may use any battery of their choice; there are no restrictions on the brand, function, or number of batteries used.

Regarding the Controllers, Sensors, Cameras, Encoders, and Actuators allowed in the FUTURE ENGINEERS regulations.

- Sensors of any brand, function, or quantity may be used.
- Teams may use any electronic component; there are no restrictions on type, brand, quantity, or purpose.

2. Programming Design: any programming language may be used. The following must be met:

- An SBC or SBM may be used without brand restrictions
- There may be more than one SBC/SBM in the vehicle.

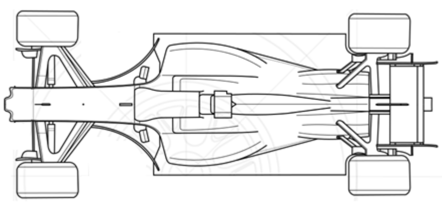
3.2.2.2. Regulations – Robot Specifications

Size	Requirements
Maximum	30 cm long 20 cm wide 30 cm high

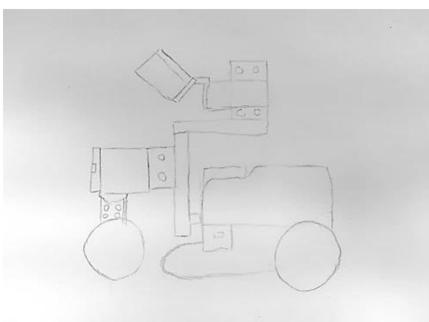
Weight	Requirements
Maximum	1.5 kilograms

3.2.2.3. Conceptual Design with SKETCHES:

Sketch 1



Sketch 2



3.2.3. BASIC ENGINEERING

As part of Basic Engineering, the specifications of materials and equipment will be integrated, both in the Mechanical-Electrical-Electronic System and in the Programming System.

The design intention of the Robot consists of **7 subsystems grouped into 3 Management Functions and 2** Systems, which are:

1. Mechanical-Electrical-Electronic System

1) Mobility Management Function

- Mechanical Structure Subsystem
- Mobility Subsystem with Electric Motors

2) Power and Sensor Management Function

- Power Supply Subsystem
- Electronic Sensors Subsystem
- Camera Subsystem
- Control Subsystem

2. Programming System

3) Obstacle Management Function

- Programming Subsystem
  - Open Challenge Strategy
  - Obstacle Challenge Strategy

Describing the Subsystems

- **Mechanical Structure Subsystem:** The objective is to develop the Robot's chassis, complementary infrastructure, and the implementation of 4 wheels.
- **Mobility Subsystem with Electric Motors:** The objective is to program movements based on activity times.
- **Power Supply Subsystem:** The objective is to provide electrical power to all motors, sensors, and the camera.
- **Proximity Sensors Subsystem:** The objective is to have sensors for line detection and obstacle proximity.
- **Camera Subsystem:** The objective is to have a camera for object and color identification.
- **Control Subsystem:** The objective is to have an Embedded Control System where the programming codes will be stored.
- **Programming Subsystem:** The objective is to have programming to execute the missions and strategies for the Open Challenge and the Obstacle Challenge.

3.2.4. DETAIL ENGINEERING

As part of Detail Engineering, calculation reports and designs were generated for the Mechanical-Electrical-Electronic System and the Programming System.

Introductively, we can say that our philosophy for building the Robot took into account 3 essential points:

1. Considering different technologies in the market to achieve a rapid integration of the robot's structure. For this, MakeBlock and Weemake technologies were taken as references.
2. Innovate the design by modifying motors (removing and adding gears to adjust speed), designing mechanical elements, and printing them in 3D.
3. Do not use complete kits with pre-designed mechanisms or codes.

3.2.4.1. MECHANICAL-ELECTRICAL-ELECTRONIC SYSTEM

This system will detail the design of Mobility Management and Energy and Sensor Management.

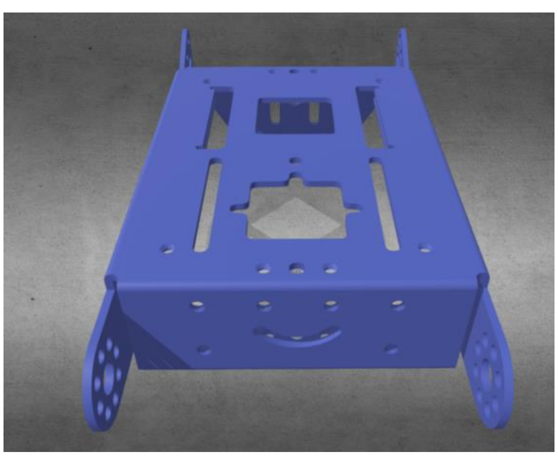
3.2.4.1.1. MOBILITY MANAGEMENT

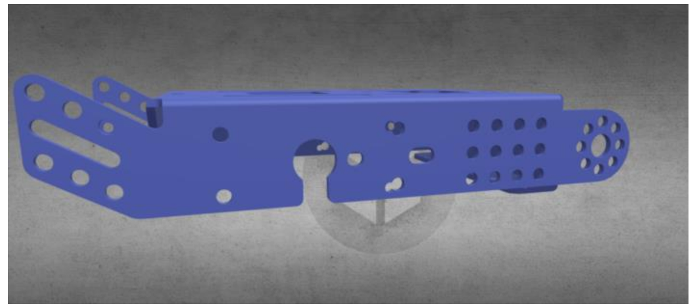
In this section, we will describe how the robot's structure was designed, the design of the wheels, how motors were selected, the list of materials, and calculation reports to determine speed, distance, and torque.

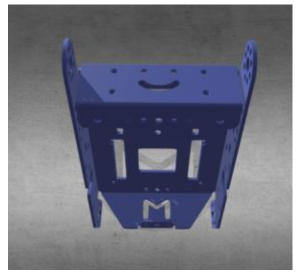
3.2.4.1.1.1. Mechanical Structure Subsystem (Chassis and Wheels)

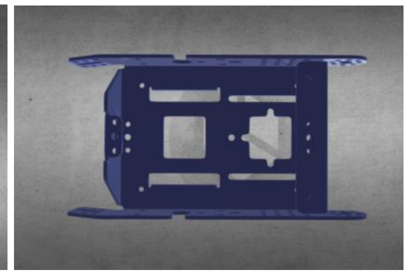
The Robot's structure was integrated based on what was described in the Robot Inspiration section, using the chassis and wheels from the MakeBlock Robot as a reference.

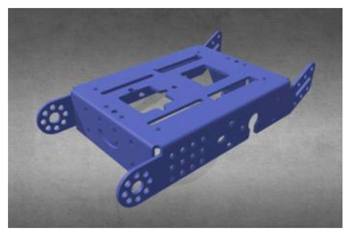
Designo of the Chassis: The chassis is made of aluminum and was designed with dimensions of 13 cm wide x 17 cm long x 3 cm high, with a thickness of 2 mm. This chassis will house the Controller, Sensors, battery, and rear wheels.

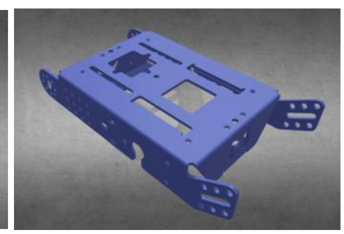












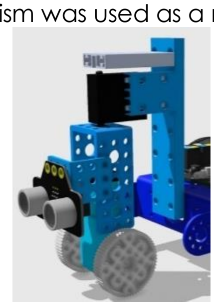
Design of the wheels: The wheels are designed with a plastic rim and a polyurethane tire, and the size varies between front and rear wheels due to the speed and torque required.

Wheels are a fundamental component in mobile robots, as they provide the necessary traction for movement. The size of the wheels is an important factor that affects the speed, energy efficiency, ground clearance, and traction of the robot.
Larger wheels require more torque to turn, but can move faster and overcome more obstacles. On the other hand, smaller wheels need less torque, but have lower speed and ground clearance.
The rear wheels are designed with a 7 cm plastic rim and a tire that is 3 mm thick.

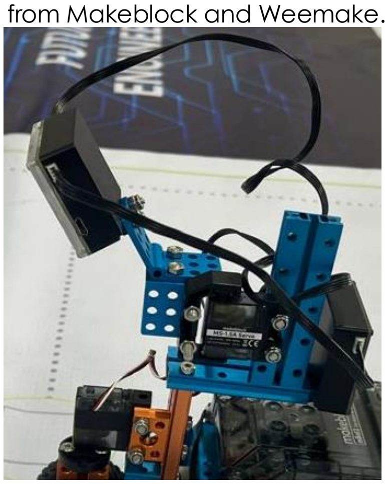
The front wheels are designed with a 4 cm plastic rim and a tire that is 2 mm thick.

The size of the front wheels is considered to be smaller in diameter than the rear wheels because the robot has front-wheel drive and requires more precision in its movements.

Design of Complementary Infrastructure: Since infrastructure is required to support the front wheels, a Weemake mechanism was used as a reference.



And for the infrastructure that supports the camera and sensors, a mechanism was designed integrating components from Makeblock and Weemake.



3.2.4.1.1.2. Mobility Subsystem with Electric Motor

As part of the design, it was necessary to evaluate the different types of motors available for educational robotics applications in order to make an appropriate selection.

Selecting the appropriate motors for a robot is crucial to achieving precise and efficient movements.

Below are **six key criteria for selecting motors** for this type of robot.

1. Torque
   - The maximum weight that the robot must support (which is 1.5 kg) is determined. Then, a motor is chosen that can handle at least this load capacity, with a certain safety margin.

2. Maximum acceleration
   - The higher the gear ratio of the reducer, the greater the maximum acceleration of the motor. Therefore, a target speed must be defined, and the motor with the appropriate characteristics should be selected.

3.Speed
   - The speed of the motors will affect the movement capabilities of the robot. Depending on the application, you may need to balance high-speed movements for quick actions or slower speeds for more precise tasks.
   - The motor must also be able to maintain a constant speed throughout its range of motion to ensure accurate and precise movements.

4. Precision and accuracy
   - Precise and accurate movements are essential, and motors with high precision will ensure that the robot can reliably repeat tasks. However, precision should not be overestimated, as it will impact the cost.

5.Energy efficiency
   - Low-power motors will help extend the robot's battery life, reducing the frequency of recharges during operation.

6.Motor type and drive
   - There are DC motors, DC geared motors, DC geared motors with encoders, servo motors, brushless motors, and stepper motors. Each of these has unique characteristics that make them suitable for specific applications.

Below is a **comparative evaluation matrix for motors** used in educational robotics applications:

		CARACTERISTICAS DE MOTORES PARA PROYECTOS				
	Tipo	Motor reductor				Mini Motor DC
	Modelo	MOT-125	MOT-120	MOT-145	MOT-150	MOT-050
Característica	Descripción					
Alimentación	Mínimo y máximo voltaje de operación.	6 Vcc	3 - 6 Vcc	6 Vcc	12 Vcc	3 - 5 Vcc
Rotación	Capacidad de girar alrededor de su propio eje.	360°	360°	360°	360°	360°
Ángulo de paso	Avance angular producido al motor por cada impulso de excitación. Medido en grados.	N/A	N/A	N/A	N/A	N/A
Relación de reducción	Cantidad de pasos efectuados por el rotor para completar una revolución.	N/A	N/A	N/A	N/A	N/A
Torque	Fuerza que es capaz de hacer en su eje. Llamado también par. Expresado en Kg*cm. A mayor par, mayor corriente de consumo.	2 kgf*cm	1 Kgf*cm a 6 Vcc	2 kgf*cm	2.7 Kgf*cm	0.11 Kgf*cm
Relación de transmisión	Cantidad de vueltas que tiene que dar el primer engrane para que el último gire una vez.	100:1	1:48 Motor biaxial (que tiene dos ejes)	N/A	N/A	N/A
Temperatura de operación	Temperatura mínima y máxima en la que puede trabajar.	-30° a 60°	-30° a 60°	-30° a 60°	-30° a 60°	-30° a 60°
Tipo de engranaje		Metálico	ABS	Metálico	Metálico	Metálico
Eje		Tipo "D"	Doble Recto	Tipo "D"	Tipo "I"	Circular
Conector		Para soldar	Para soldar	Para soldar	Cables	Para soldar
Corriente de consumo sin carga	Consumo.	120mA	**3 Vcc ≤ 150mA, 6 Vcc ≤ 200mA**	6 Vcc 322mA	12 Vcc a 80mA	5 Vcc 100mA
Velocidad	Revoluciones por minuto (rpm).	6 Vcc 310 rpm	3 Vcc 90 +- 10% rpm 6 Vcc 200 +- 10% rpm	6 Vcc 47.5 rpm	12 Vcc 4.2 rpm	5 Vcc 13,000 rpm
Peso		9gr	28gr	80gr	55gr	19gr
Dimensiones	Medidas de motor.	35.5x12x10mm	70×22.5×23mm	25x25x65mm	46x32x35mm	15x20x40mm
Ruido	Expresado en decibeles.	<35dB	<65dB	<45dB	<45dB	<35dB
Accesorios		N/A	N/A	N/A	N/A	N/A
Ideal para		Dar movimiento a llantas, poleas, engranes, entre otros.	Dar movimiento a llantas, poleas, engranes, entre otros.	Dar movimiento a llantas, poleas, engranes, entre otros.	Dar movimiento a llantas, poleas, engranes, entre otros.	Dar movimiento a llantas, poleas, engranes, entre otros.

*N/A: No Aplica

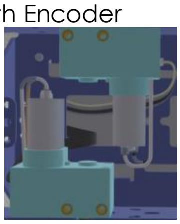
		CARACTERÍSTICAS DE MOTORES PARA PROYECTOS				
	Tipo	Motor a pasos	Servo motor			
	Modelo	MOT-130	MOT-100	MOT-110	MOT-135	MOT-140
Característica	Descripción					
Alimentación	Mínimo y máximo voltaje de operación.	5 Vcc	3.5 - 6 Vcc	3.5 - 6 Vcc	3.5 - 6 Vcc	4.8 - 6 Vcc
Rotación	Capacidad de girar alrededor de su propio eje.	360°	180°	180°	180°	360°
Ángulo de paso	Avance angular producido al motor por cada impulso de excitación. Medido en grados.	5.625°	N/A	N/A	N/A	N/A
Relación de reducción	Cantidad de pasos efectuados por el rotor para completar una revolución.	1/64	N/A	N/A	N/A	N/A
Torque	Fuerza que es capaz de hacer en su eje. Llamado también par. Expresado en Kg*cm. A mayor par, mayor corriente de consumo.	0.3 Kgf*cm	1.8 kgf*cm	4.8 Vcc - 1.8 Kgf*cm 6 Vcc - 2.2 Kgf*cm	11 Kgf*cm a 6 Vcc	6.5 Kgf*cm a 6 Vcc
Relación de transmisión	Cantidad de vueltas que tiene que dar el primer engrane para que el último gire una vez.	N/A	N/A	N/A	N/A	N/A
Temperatura de operación	Temperatura mínima y máxima en la que puede trabajar.	-30° a 60°	-30° a 60°	-30° a 60°	-30° a 60°	-30° a 60°
Tipo de engranaje		Metálico	Nylon	Metálico	Metálico	Nylon
Eje		Tipo "I"	Tipo piñón	Tipo piñón	Tipo piñón	Tipo piñón
Conector		Universal 5 pines	Tipo "S"	Tipo "S"	Tipo "S"	Tipo "S"
Corriente de consumo sin carga	Consumo.	55mA	20mA a 6 Vcc	40mA a 6Vcc	100mA a 6 Vcc	100mA a 6 Vcc
Velocidad	Revoluciones por minuto (rpm).	Depende de la programación de 1 a 30rpm	0.10s/60° a 6Vcc	0.08s/60° a 6 Vcc	<0.16s/60° a 6 Vcc	<0.15s/60° a 6 Vcc
Peso		35 grms	9gr	15gr	60gr	55gr
Dimensiones	Medidas de motor.	Longitud de eje 9mm, diámetro 28mm, alto, 20mm.	23x12.2x29mm	35x32x12mm	40.7x19.7x42.9mm	40x20x39 mm
Ruido	Expresado en decibeles.	<35dB	<35 dB	<35dB	<50 dB	<55dB
Accesorios		N/A	3 brazos de material termoplástico acabados en color blanco, 1 tornillo de sujeción, 2 tornillos para montaje.	3 brazos de material termoplástico acabados en color negro, 1 tornillo de sujeción, 2 tornillos para montaje.	3 brazos de material termoplástico, 1 tornillo de sujeción, 2 tornillos para montaje.	3 brazos de material termoplástic 1 tornillo de sujeción, 2 tornillos para montaje.
Ideal para		Movimientos precisos. Ejemplo: Impresora 3D	Utilizado para ala fija de avión, coche radio control, robótica articulaciones. Conector compatible con receptores Futaba y JR.	Utilizado para ala fija de avión, coche radio control, robótica articulaciones. Conector compatible con receptores Futaba y JR.	Utilizado para ala fija de avión, coche radio control, robótica articulaciones. Conector compatible con receptores Futaba y JR.	Utilizado para ala fija de avión, coche radio control, robótica articulaciones. Conector compatible con receptores Futab y JR.

*N/A: No Aplica

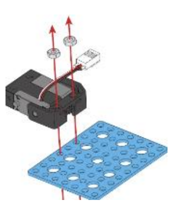
MOTOR SELECTION

After analyzing and evaluating the motors, it was determined that a **DC geared motor** will be used for the mobility of the wheels and a **servo motor** for the steering.

DC Geared Motor with Encoder



SERVOMOTOR

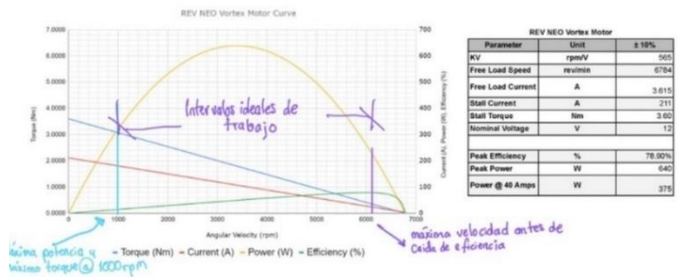


CALCULATION RECORDS

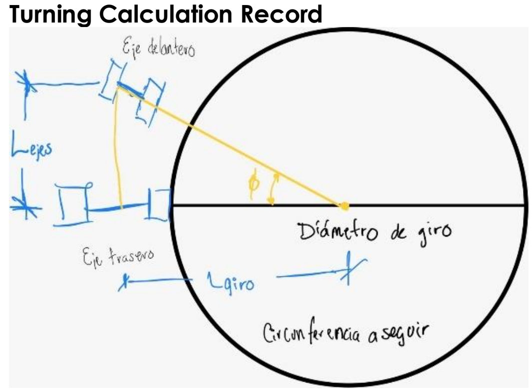
Distance Traveled per Wheel

	Cálculo deDistancia recorrida de la Llanta			
	DATOS			
	diametro de la llanta	d=	40	mm
	Pi	π=	3.1416	
Variable de decisión de distancia	# vueltas de la llanta	=	1	
	Perimetro	P=	π*d	
		P=	3.1416 * 60 mm	
		P=	125.664	mm
	Distancia = # de vueltas * Perimetro de la llanta			
	1 vuelta completa =		125.664	mm
	Distancia =		15 * 188.496 mm	
Resultado	Distancia =		126	mm

Torque



Turning Calculation Record



3.2.4.1.2. POWER AND SENSOR MANAGEMENT

In this section, the battery, sensors, camera to be used, and embedded system will be selected. Additionally, a calculation record for electrical loads, a materials list, a wiring diagram, and a control architecture will be created.

3.2.4.1.2.1. Power Supply Subsystem

In this section, the battery will be selected, and the energy consumption and power will be calculated.

Battery Selection: The pros and cons of the batteries will be stated, along with their characteristics, and the appropriate battery for the robot will be selected.

La siguiente tabla detalla: batería de polímero de litio versus batería de iones de litio:

Feature	Iones de litio (Li-ion)	De polímero de litio (Li-Po)
Electrolito	Liquido	Estado sólido, tipo gel o polímero
Estructura	Rígido, rectangular	Se puede moldear en varias formas.
Seguridad	Menos seguro debido a la posibilidad de fugas y fugas térmicas	Menos seguro debido a la posibilidad de fugas y fugas térmicas.
Densidad de energía	Más Bajo	Más alto
Ciclo de vida	Más	Shorter
Costo	Menos costoso	Más caro
Menos seguro debido a la posibilidad de fugas y fugas térmicas.	Autodescarga	1-2% por mes
Aplicaciones típicas	Vehículos eléctricos, Médicos, Sistema UPS	Drones, vehículos RC, electrónica portátil.

Energy and Power Calculation Record

DISPOSITIVO	VOLTAJE		CONSUMO DE CORRIENTE		TIEMPO ACTIVO		CICLO DE TRABAJO	CONSUMO DE ENERGÍA		# DE EQUIPOS	CONSUMO DE ENERGÍA TOTAL	
SISTEMA EMBEBIDO	3.7	V	500	mA	7	min	11.67%	1.8500	Wh	1	1.8500	Wh
SENSOR ULTRASONICO	5	V	15	mA	7	min	11.67%	0.0750	Wh	3	0.2250	Wh
RANGE SENSOR	5	V	40	mA	7	min	11.67%	0.2000	Wh	1	0.2000	Wh
CÁMARA	5	V	250	mA	7	min	11.67%	1.2500	Wh	1	1.2500	Wh
MOTOR DC	5	V	800	mA	7	min	11.67%	4.0000	Wh	1	4.0000	Wh
SERVOMOTOR	5	V	500	mA	7	min	11.67%	2.5000	Wh	1	2.5000	Wh
											10.0250	Wh

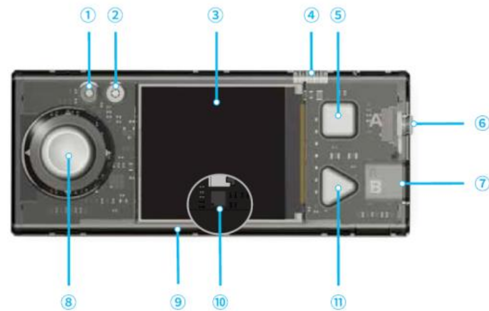
	TIEMPO DE TRABAJO DE BATERÍA			
BATERIA	6.66	Wh	1,800	mAh
CONSUMO DE ENERGÍA	10.0250	Wh		
	0.6643	HORAS		

3.2.4.1.2.2. Electronic Sensors Subsystem

The structure of the robot was integrated based on what is described in the Robot Inspiration section, using the chassis and wheels from the MakeBlock robot as a reference.

Sensor selection: the sensors used in MakeBlock will be listed, and a selection will be made.

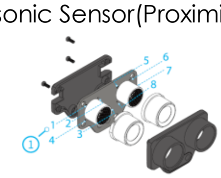
The sensors on the **CyberPi board of the mBot2** include an embedded **gyroscope-accelerometer (IMU) and a light** sensor, which can be used to map the robot's location and the rate of speed change.

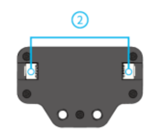


① Light sensor ⑩ Gyroscope, accelerometer

Additionally, the sensor kit that can be used includes the Ultrasonic Sensor, RGB Sensor, IR Sensor, and PIR Sensor.

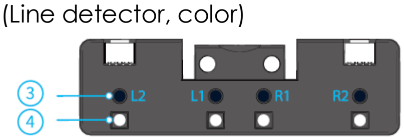
Ultrasonic Sensor(Proximity)





IR Sensor(Proximity)

RGB Sensor (Line detector, color)



PIR Sensor (People detector)

The selected sensors are:

- Gyroscope
- Ultrasonic Sensor
- IR Sensor
- RGB Sensor

3.2.4.1.2.3. Camera Subsystem

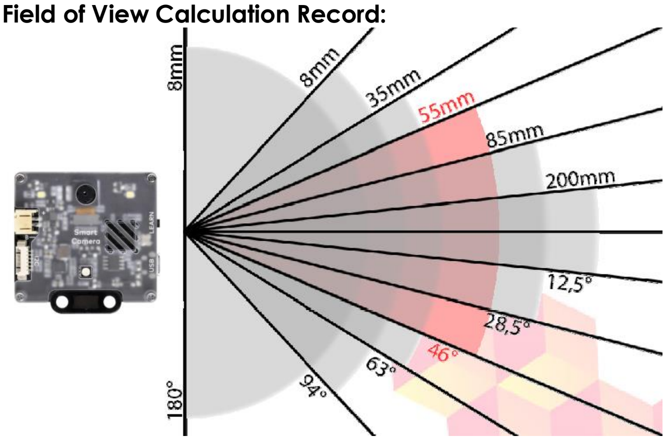
In this section, the camera to be used will be selected, and the field of view to be used with the camera will be determined.

Camera Selection: There are cameras for computer vision applications with an educational focus that can be installed on Raspberry Pi, Arduino, LEGO, and mBot.

MBot Camera

Since the **mBot camera** has local processing for basic computer vision applications, **it was decided to use this camera** to reduce the learning curve, and with application commands, it can be used for image and color recognition

Field of View Calculation Record:



The maximum field of view is shown in the previous figure, but a shorter range can be configured, which can improve detection accuracy. Therefore, a range of 30% of the maximum value was defined

3.2.4.1.2.4. Control Subsystem

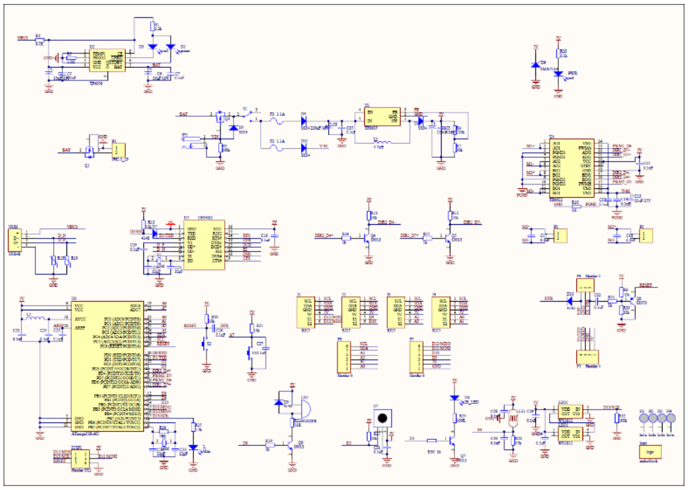
In this section, the embedded system to be used will be selected, the wiring diagram of the embedded system will be created, the board design will be shown, and the control system architecture will be developed

Selection of the Embedded System (Controller): Although there are various embedded systems, we will only analyze 3 which are Raspberry, Arduino and mBot (ESP32).

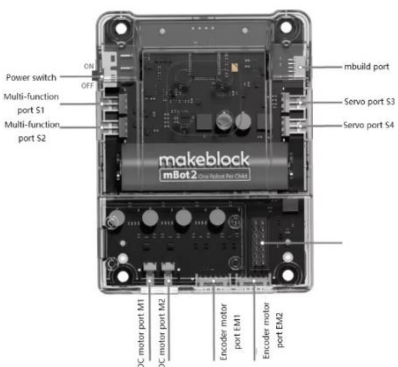
Features/Specs	Raspberry Pi Pico	Arduino Uno R3	ESP32 Xtensa
Microcontroller	RP2040	Atmega 328 P	ESP32
Cores	Dual-Core	Single Core	2
Core Architecture	32-bit ARM Cortex-M0+	8-bit RISC	32BIT
CPU Clock	48MHz, upto 133	16MHz	160Mhz
RAM Size	264 Kbyte SRAM	2KByte	520 KB
Flash Size	2 Mbyte Q-SPI Flash	32 Kbyte	16Mb
EEPROM	NO	1 Kbyte	NO
Programming Language	MicroPython, C, C++	C alike, Arduino IDE	Python
Board Power Input	5VDC vi VSYS Pin (Pin 39)	5VDC via USB B	5V (via USB) or 7-12V (via Vin)
Alternative Board Power	2- 5VDC via VSYS Pin (Pin 39)	7-12VDC via DC Barrel	7-12V (via Vin)
MCU Voltage	3.3VDC	5VDC	5VDC
GPIO Voltage	3.3VDC	5VDC	5VDC
USB Interface	USB 1.1 DEVICE AND HOST	External USB Serial IC	Micro USB C
Program Loading	USB micro B USB Mass storange	USB B, Virtual serial port	Micro USB C, Bluetooth
GPIO Voltage	26 X Digital input/output (Total)	20 x Digital Input/output (Total)	36
ADC	3 x 12-bit	6 x 10-bit	18
UART	2	1 (shared with USB serial)	2
I2C	2	1	2
SPI	2	1	1
PWM	16	6	8
On Boad LED	1x programmable LED (GP25)	1 x programmable LED (D13)	1
Pines	NO	Female Header pre-soldered	26

After analyzing the table, it can be seen that the 3 embedded systems have the capacity to face the challenges of the challenges, but... one of the characteristics of the mBot2 is that it can be programmed in blocks (which through commands can have access to simplify routines) and in the case of using the mbot camera, then the preconfigured functions of artificial vision can be maximized. For this reason, mBot2 was selected as the Controller. **For this reason, mBot2 was selected as the controller.**

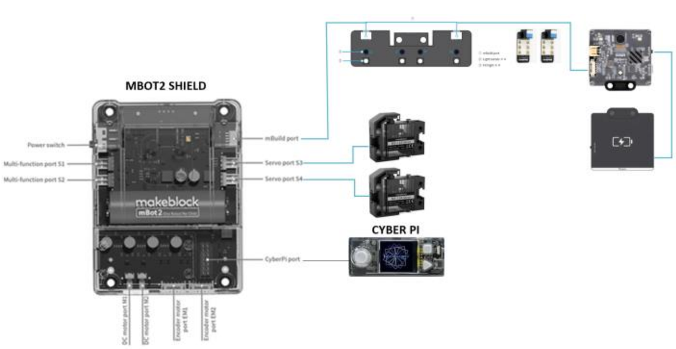
The Wiring Diagram for the mBot2 Embedded System was developed in Visio:



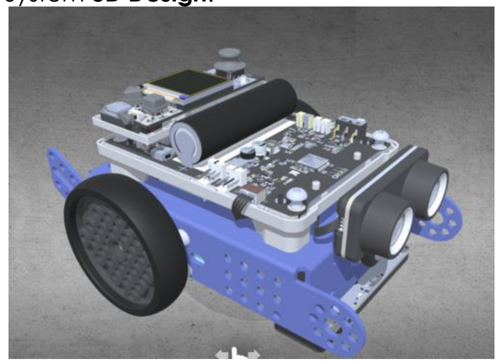
the board design of the mBot2 from MakeBlock is shown below:



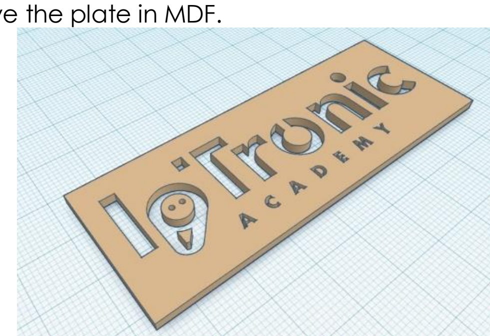
The architecture of the control system, which integrates the embedded system with the sensors, motors, and batteries, was developed in Visio and is shown below:



Control System **3D Design**:



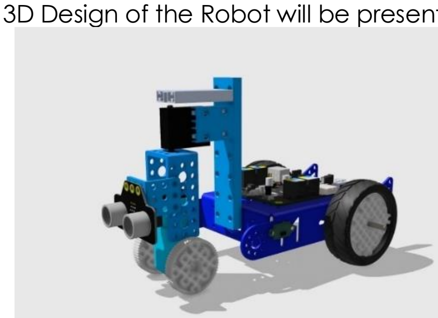
Aceesories **3D Design**:
The robot nameplate was designed in 3D to cut and engrave the plate in MDF.



3.2.4.1.3. LIST OF MATERIALS:

Next, the list of **Mechanical, Electrical, and Electronic** materials will be presented.

3.2.4.1.4. 3D Design of the Robot
Next, the 3D Design of the Robot will be presented:



3.2.4.2. PROGRAMMING SYSTEM

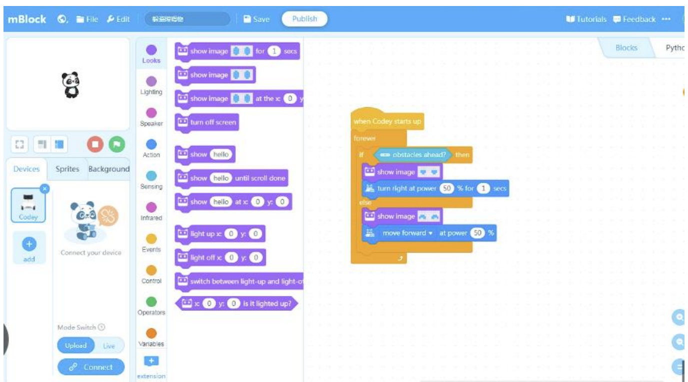
In this section, the programming language will be selected, the programming environment will be defined, and the programming algorithm will be developed.

3.2.4.2.1. PROGRAMMING LANGUAGE SELECTION:

The options for programming the mbot 2 are:

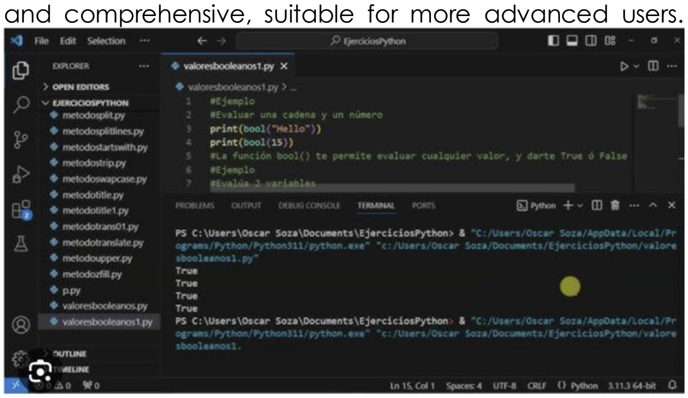
1. mBlock

A graphical programming software based on Scratch 2.0 that can be installed on a computer. The free app for Chrome, as well as the app for Android or iOS, or the Bluetooth module that attaches to the robot can also be used.



2. Python

A more demanding programming language, but flexible and comprehensive, suitable for more advanced users.



It is worth mentioning that, **although programming is done in mBlock**, the coding **can be viewed in Python**. Since mBlock has commands that combine coding sequences in Python, it was decided to select mBlock for efficiency, but the documentation on GitHub was created in Python

3.2.4.2.2. OBSTACLE MANAGEMENT

In this section, Artificial Vision will be described as a programming tool, and the programming algorithms for the challenges will be presented.

Artificial vision is an interdisciplinary scientific field that studies how computers can gain high-level understanding from digital images or videos. From an engineering perspective, it seeks to understand and automate tasks that the human visual system can perform.

3.2.4.2.2.1. Challenge Programming Subsystem

In this subsystem, the Philosophy of the Challenge Strategy, the Programming Algorithms, and the Programming Tools used are integrated

CHALLENGE STRATEGY

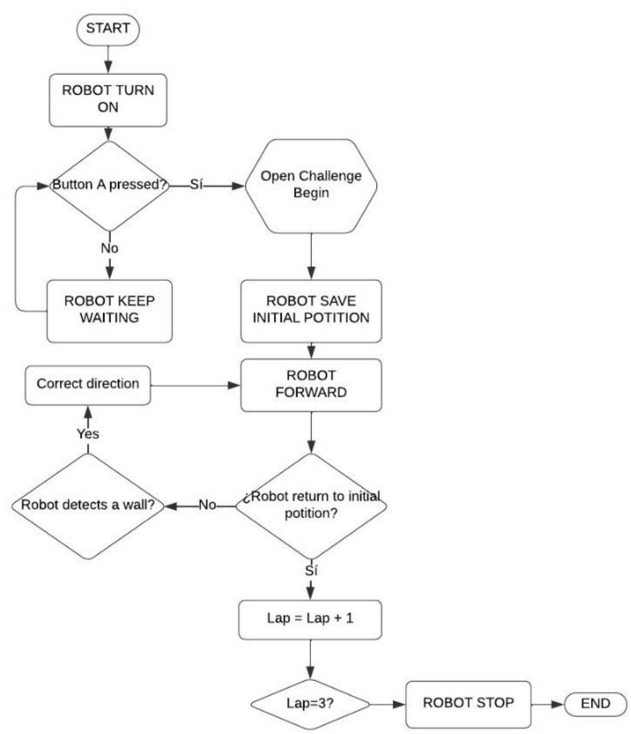
Open Challenge Strategy: The strategy involves using proximity sensors to locate the limiting walls and using line sensors to control the robot's turning.

Obstacle Challenge Strategy: The strategy involves using proximity sensors to locate the limiting walls and using a camera for obstacle and color recognition to enable the robot to change direction and identify parking spots to park itself.

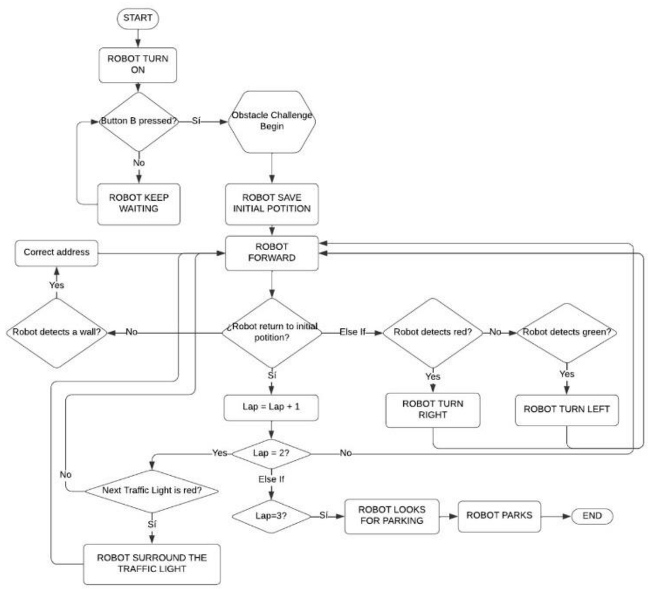
PROGRAMMING ALGORITHMS

Programming Algorithms:

Next, the algorithm for the **Open Challenge** Strategy will be presented:



Now, the algorithm for the **Obstacle Challenge** Strategy will be presented:"



PROGRAMMING TOOLS USED

The following programming technologies were used: **PID, Artificial Vision, Discrete Control, Times and movements.**

3.3. ROBOT CONSTRUCTION

In this phase of the project, the robot must be constructed according to the established design. This involves assembling the robot, coding, problem detection, finding solutions, and refining the robot to achieve the Final Production of the Robot.

3.3.1. CONSTRUCTION PROCESS

Assembling the robot was very exciting, as my cousin, my brother, and I learned a lot and had fun. We spent many Saturdays eating pizza, sandwiches, tacos, and soda; on a couple of occasions, we stayed up late working on the assembly because we didn't even notice the passage of time.

Our robot went through **4 phases of construction evolution**, and in each phase, we encountered errors or challenges that we had to address, which allowed us to continuously improve the robot.

The **assembly process** was carried out in the following **4 phases**:

Phase I – Assembly of the Mobility Management System - Mechanical

Base Mechanical Structure (Chassis)

- Assembly of the robot's base structure
- Assembly of the wheels

Complementary Mechanical Structure

- Structure to support the camera and sensors

Phase II – Assembly of the Energy Management System - Electrical

- Installation of the battery
- Installation of the motors

Phase III – Assembly of the Sensor and Actuator Management System - Electronics

- Innstallation of the Controller
- Installation of the Cameras and Sensors

Phase IV – Obstacle Management - Robot Programming

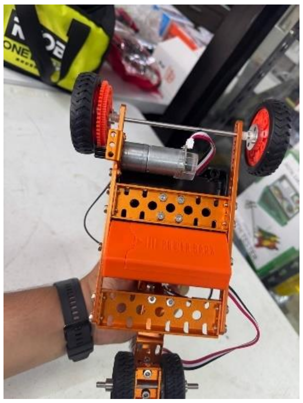
- 1st Round
- 2nd Round

3.3.2. EVOLUTION OF THE ROBOT

It starts with prototypes and is refined through constant evolution.

MECHANICAL PROTOTYPES

We started with various prototypes that didn't always work, or at least not as we desired, which led us to build different prototypes. We firmly believe that trial and error are the best elements for achieving great things.



Some parts were made using the 3D printer, the lathe, and the cutter.



EVOLUTION OF THE ROBOT

Our robot went through **3 phases of design and construction evolution**, and in each phase, we encountered errors or challenges that we had to address, which allowed us to continuously improve the robot.

FIRST EVOLUTION PHASE

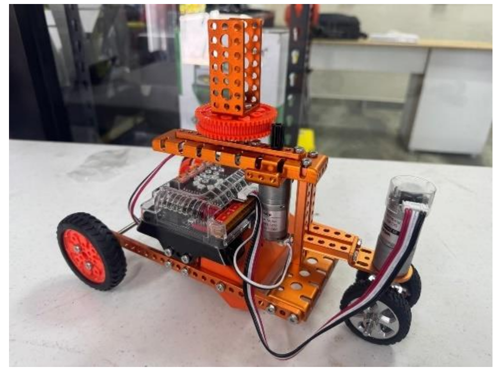
Assembly of the "Competition Track":

Although it was difficult to find someone who could print the competition track, we were able to get it printed.

Chassis Design– Mobility Function

Development Features: In this first design version, we used 1 DC motor to control 2 rear wheels and 1 DC motor to control the steering with 2 front wheels. This design resulted in slow speed and poor turning control, so this option was discarded

Version A



SECOND PHASE OF EVOLUTION

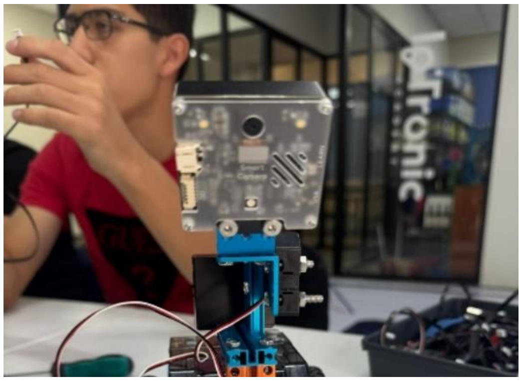
Chassis Modification – Mobility Function

Development Features: In this second design version, we used 1 DC motor to control 2 rear wheels and a gear transmission system to improve speed. Additionally, a servo motor was used to control the 2 front wheels. This design improved speed (but with deficiencies in speed control) and enhanced turning control.

Version B

Design of the Structure for the Camera Turning Control Function.

Development Features: A support that holds a gear controlled by a servo motor was integrated. The main idea is that, based on detection, the camera's position can be rotated for greater reach.

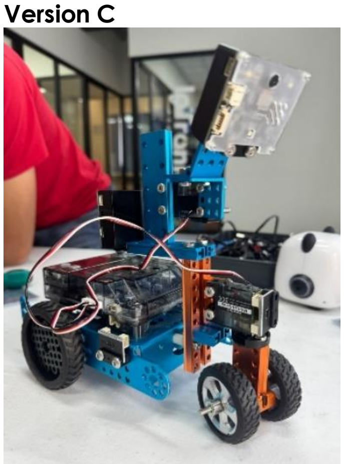


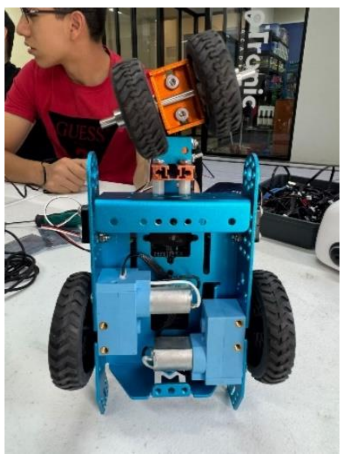
THIRD PHASE OF EVOLUTION

Chassis Modification– Mobility Funcion

Development Features: In this new version, we used 2 DC motors to control the 2 rear wheels and 1 servo motor to control the front wheels. This design improved speed control.

Version C



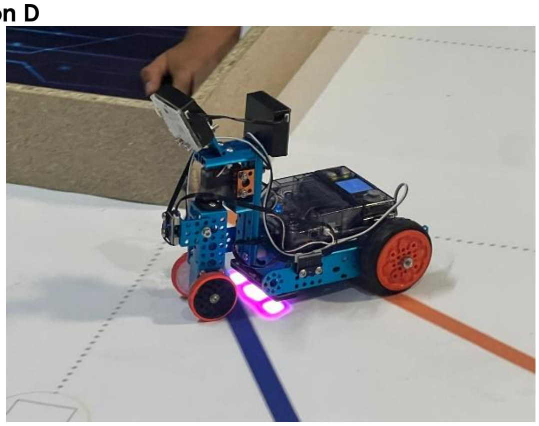


FOURTH PHASE OF EVOLUTION

Chassis Modification – Mobility Function

Development Features: In this new version, we used 1 DC motor to control the mobility of the 2 front wheels and 1 servo motor to control the turning of the front wheels. This design improved control and ensured compliance with the regulations.

Version D



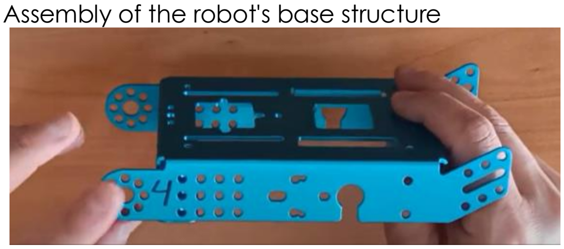
3.3.3. FINAL ASSEMBLY OF THE ROBOT

There are 4 phases for the assembly of the robot.

Phase I – Assembly of the Mobility Management System - Mechanica

Base Mechanical Structure (Chassis)

- Assembly of the robot's base structure



- Assembly of the wheels

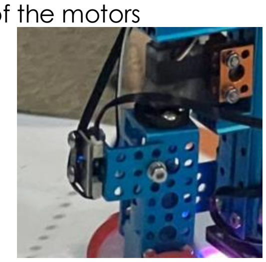
Complementary Mechanical Structure

- Structure to support the camera, sensors, and front traction.

Phase II – Assembly of the Energy Management System - Electrical

- Installation of the battery

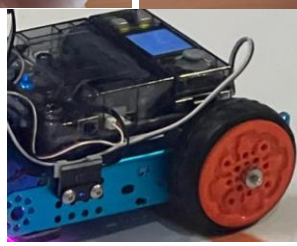
- Installation of the motors



Phase III - Assembly of the Sensor and Actuator Management System - Electronics

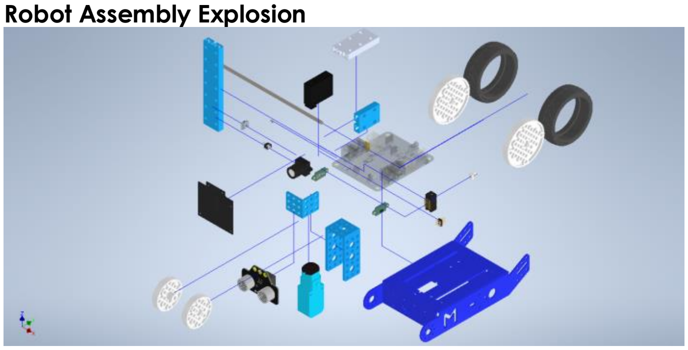
- Installation of the Controller





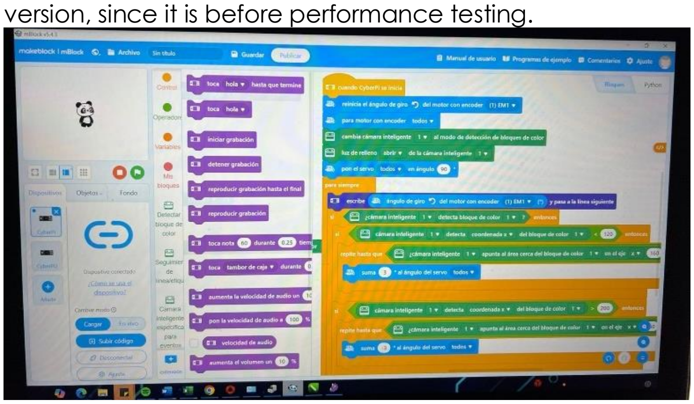
- Installation of the Sensors y Cameras


Robot Assembly Explosion



Phase IV – Obstacle Management - Robot Programming

In this phase, the code was developed almost in its final version, since it is before performance testing.



3.3.4. INNOVATION

As a technological innovation, we had to disassemble a motor and change the last gear of the mechanism for a larger gear because we needed to adjust the torque and speed of the robot to improve performance on the competition track.

Below is the motor with 3 gears, and with the last gear larger (38 teeth instead of 24 teeth).

3.3.5. SUMMARY OF PROBLEMS AND SOLUTIONS

During the construction of the Robot we encountered many problems, among which are the following:

Problem: Incorrect detection of traffic signs.
Description: The robot's vision system detected multiple traffic signs simultaneously, which caused confusion in decision making, since it was not able to correctly differentiate between duplicate signs.
Solution: Improve the recognition algorithm so that it identifies and dismisses duplicate signs in the same vision frame. This was achieved by creating a sectorization with pixels and thus avoiding redundancies in decision making.

Problem: Poor color detection due to lighting conditions.
Description: The robot's camera did not have sufficient lighting, which affected its ability to detect and differentiate colors, especially in low-light or shadow environments.
Solution: Install an LED lighting system that provides direct light to the camera detection area. This will ensure constant and adequate lighting, significantly improving the accuracy in color detection and allowing the robot to make more reliable visual decisions in different environments.

Problem: Inaccurate measurements on range sensors.
Description: The range sensors were sometimes providing incorrect or inconsistent readings, affecting the overall performance of the robot and its ability to avoid obstacles.
Solution: Implement a regular calibration process and adjustment of sensor positions. This will reduce inaccuracies and increase the reliability of measurements under various conditions.

Problem: Front drive and steering tire coming loose.
Description: The front tire, which also provides the robot's drive and steering, would occasionally come loose, disrupting its operation and compromising control.
Solution: Reinforce the tire mount using better quality fasteners or a fastening system that prevents loosening. Additionally, industrial adhesives or security screws may be

considered to ensure a stronger and longer lasting attachment.

Problem: Excessive power consumption by the camera.
Description: The robot's camera consumed a large amount of power, which decreased battery life and limited the robot's operating time.
Solution: Integrate an additional battery dedicated exclusively to the vision components, thus improving the overall autonomy of the robot.

Problem: Insufficient robot travel speed.
Description: The robot had a slow travel speed, which affected its performance in tasks that required agile mobility and short response times.
Solution: Improve the robot's speed by increasing the power of the motors or adjusting the transmission ratio on the gearmotors. Additionally, optimizing the control software can help the robot respond more quickly to travel instructions.

Problem: Limitation in turning due to rear-wheel drive.
Description: The rear-wheel drive design caused the robot to make turns that were too wide, making it difficult to maneuver in tight spaces.
Solution: Evaluate the possibility of implementing a front-wheel drive system to improve turning capacity. Alternatively, the turning angle can be adjusted, which will allow a more agile range of movement in small areas.

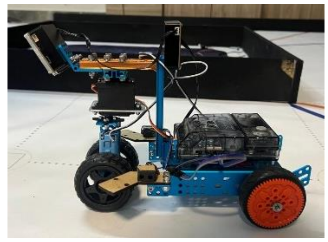
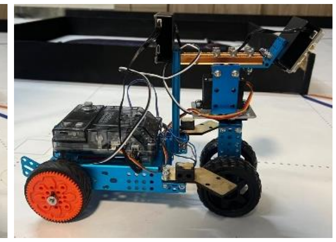
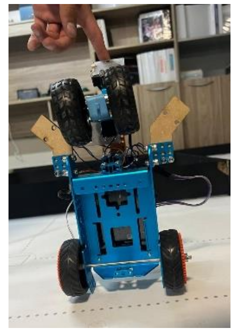
3.3.6. FINAL PRODUCTION OF THE ROBOT

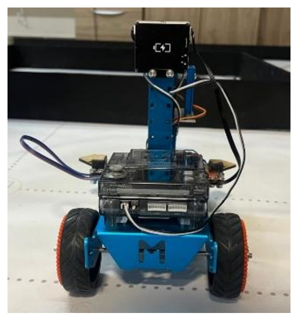
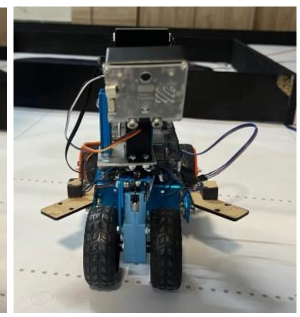
Finally, after all this great challenge, we have successfully completed our FUTURE ENGINEERS Competition Robot project and its programming.

3.3.6.1. ROBOT

The GOAT L3 Robot in its Version 0.0 for the National WRO Future Engineers Competition.

Pictures Vehicule:

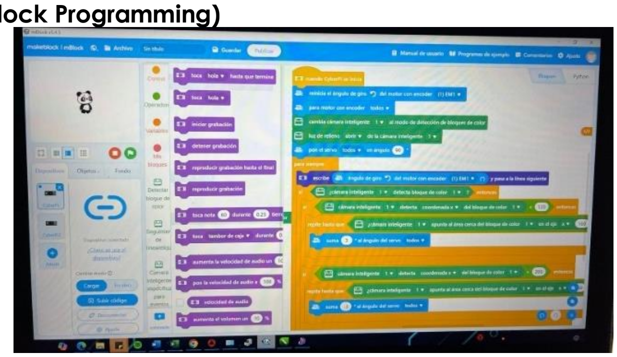
3.3.6.2. PROGRAMMING

In programming, we decided to use block-based programming
This programming is segmented by each of the 3 Challenges:

1. Open Challenge
2. Obstacle Challenge– 3 Laps
3. Obstacle Challenge – Parking in a Parking space

In this phase, the code is developed and is in its final version, although we will be improving the code up to 2 weeks before the competition and it will be uploaded to Github in its final version on November 10.

Next, the program for the Open Challenge is presented: (Block Programming)



Next, the program for the Obstacle Challenge is presented: (Programming in Python)

```python
# generated by mBlock5 for CyberPi
# codes make you happy

import event, time, cyberpi, mbot2, mbuild
import time
# initialize variables
vuelta = 0

@event.is_press('b')
def is_btn_press():
    global vuelta
    mbot2.servo_set(90,"S3")
    vuelta = 0
    mbuild.smart_camera.set_mode("color", 1)
    mbuild.quad_rgb_sensor.set_led_color("blue", 1)
    cyberpi.reset_rotation('all')
    mbot2.servo_set(90,"all")
    while not vuelta == 4:
      cyberpi.console.print(mbuild.quad_rgb_sensor.get_ground_sta("all", 1))
      if mbuild.quad_rgb_sensor.get_ground_sta("all", 1) > 0:
        mbot2.straight(15)
        mbot2.servo_set(135,"S3")
        while not cyberpi.get_rotation('z') < -89:
          mbot2.drive_power(50, -25)

        mbot2.servo_set(0,"S3")
        mbot2.turn(-13)
        time.sleep(0.3)
        mbot2.servo_set(90,"S3")
        mbot2.straight(5)
        mbot2.EM_stop("ALL")
        time.sleep(0)
        mbot2.servo_set(90,"S3")
        vuelta = vuelta + 1
        cyberpi.reset_rotation('all')

      else:
        mbot2.forward(50)

@event.is_press('middle')
def is_joy_press():
    global vuelta
    # DO SOMETHING
    pass

def correcci_C3_B3n():
    global vuelta
    if cyberpi.get_rotation('z') > 0:
      mbot2.EM_stop("ALL")
      cyberpi.console.println(cyberpi.get_rotation('z'))
      while not cyberpi.get_rotation('z') < 1:
        mbot2.servo_set((mbot2.servo_get("S3") + 1),"S3")
        mbot2.forward(120)
        cyberpi.console.println(cyberpi.get_rotation('z'))

    if cyberpi.get_rotation('z') < 0:
      mbot2.EM_stop("ALL")
      cyberpi.console.println(cyberpi.get_rotation('z'))
      while not cyberpi.get_rotation('z') > -1:
        mbot2.servo_set((mbot2.servo_get("S3") - 1),"S3")
        mbot2.forward(120)
        cyberpi.console.println(cyberpi.get_rotation('z'))

@event.is_press('a')
def is_btn_press1():
    global vuelta
    cyberpi.reset_rotation('all')
    for count2 in range(3):
      for count in range(4):
        mbot2.servo_set(90,"all")
        time.sleep(1)
        mbot2.forward(60, 0.5)
        mbot2.forward(120, 0.5)
        while not mbuild.ranging_sensor.get_distance(3) < 60:
          mbot2.forward(180)

        mbot2.EM_stop("ALL")
        time.sleep(0.1)
        mbot2.servo_set(135,"S3")
        while not cyberpi.get_rotation('z') < -90:
          mbot2.EM_set_speed(40,"EM1")
          cyberpi.console.println(cyberpi.get_rotation('z'))
          cyberpi.console.clear()

        while not cyberpi.get_rotation('z') < 0:
          mbot2.EM_set_speed(-40,"EM1")
          cyberpi.console.println(cyberpi.get_rotation('z'))
          cyberpi.console.clear()

        cyberpi.reset_rotation('all')
        mbot2.EM_stop("ALL")

      mbot2.servo_set(0,"S3")
      mbot2.turn(-15)
      cyberpi.reset_rotation('all')

    mbot2.servo_set(90,"all")
    mbot2.forward(50, 2)

@event.start
def on_start():
    global vuelta
    # DO SOMETHING
    pass
    end
```

Note: Coding can be done in blocks, but can be visualized in python, or vice versa, programming in python and visualizing in blocks.

3.4. INSPECCIÓN Y PRUEBAS (TESTING)

En este apartado se realizan las Inspecciones Físicas del Robot y las Pruebas.

3.4.1. INSPECTION

In the inspection process, a checklist was used that includes the dimensions of the robot, the materials to be used, and the design restrictions. After the inspection, **it was deemed that the inspection was satisfactory.**

3.4.2. TESTING

As a testing plan, it was defined that the following types of tests would be conducted:

- Unit Tests: After completing the construction of each subsystem.
- Operational Tests or Calibration of the Robot: These are functionality tests for each subsystem with the fully finished robot.
- Performance Tests: These are tournament simulation tests, consisting of 3 runs per simulation, and to evaluate performance, 2 simulations are conducted.

It was defined as the Scoring Objective the following:

Missions	Target Time (s)	Target Score
Abierta	45	27
Obstaculos - Vueltas	80	8
Obstaculos - Parqueada	15	7
Libro de Ingeniería	0	30
	140	72

The first evaluation is that the average results of a simulation should be equal to or greater than 70% of the Scoring Objective.

The second evaluation is that the average of the second simulation should be at least 10% higher than that of the first simulation.

SIMULACIÓN 1

Missions	Target Time (s)	Target Score	S1 (s)	Score	S2 (s)	Score	S3 (s)	Score	Average (s)	Average Score
Abierta	45	27	43	24	45	20	42	23	43	22
Obstaculos - Vueltas	80	8	75	2	74	4	72	4	74	3
Obstaculos - Parqueada	15	7	17	0	15	7	14	0	15	2
Libro de Ingeniería	0	30	0	30	0	30	0	30	0	30
	140	72	135	56	134	61	128	57	132	58
				78%		85%		79%		81%

SIMULACIÓN 2

Missions	Target Time (s)	Target Score	S1 (s)	Score	S2 (s)	Score	S3 (s)	Score	Average (s)	Average Score
Abierta	45	27	43	24	45	24	42	27	43	25
Obstaculos - Vueltas	80	8	75	4	74	8	72	8	74	7
Obstaculos - Parqueada	15	7	17	0	15	7	14	7	15	5
Libro de Ingeniería	0	30	0	30	0	30	0	30	0	30
	140	72	135	58	134	69	128	72	132	66
				81%		96%		100%		92%
Improvement (%)				3%		11%		21%		12%

The performance tests were satisfactory!

4. STRATEGIES

The team defined 3 types of strategies to approach the tournament:

After a brainstorming meeting among the team members, we discussed **1. The Competition Rules, 2. Experience in Regional Tournaments, 3. The Experience of the Coaches**, and established the following gameplay strategies:

1. Open Challenge Strategies
   a) Starting Position
   b) Starting Direction
2. Obstacle Challenge Strategies – 3 laps
   a) 3 laps in the same direction
   b) 2 laps in the same direction and 1 in the opposite direction
   c) Parking detection

Development of the Open Challenge Strategies

1 a): Start in a clockwise direction at low speed to ensure compliance with the Challenge. In the final phase, implement the route at high speed.

1 b): Pay attention to the configuration of the program's decision variables in order to execute the route in a counterclockwise direction.

Development of the Obstacle Challenge Strategies

2 a): Start in a clockwise direction at low speed to ensure compliance with the Challenge. In the final phase, implement the route at high speed.

2 b): Pay attention to the configuration of the program's decision variables so that when detecting the last signal, the robot can correctly decide whether the final lap is in the same direction or involves a change of direction.

2 c): After completing the 3 laps, I was able to identify the parking spot and park.

5. ACKNOWLEDGEMENTS

To our parents, uncles, and grandparents who supported us and continue to support us in this project, as well as to our Coach and Mentor, and of course to our sponsors, without whom none of this would be possible

We especially thank God for blessing us at all times and for giving us these experiences.

6. PHOTOS, VIDEOS AND GITHUB

The photographic report, videos, and GitHub management can be found at the following link.
https://github.com/IotronicAcademy
https://github.com/IotronicAcademy/WRO-Future-Engineers-Mex-2025