creditwithout / -

Code Issues Pull requests Actions Projects Security Insights

라디오헤드 노래 목록 또는 계층 목록

0 stars 0 forks 0 watching 1 Branch 0 Tags Activity

Private repository

main 1 Branch 0 Tags Go to file t Go to file Add file Code

creditwithout Add files via upload 9d2f148 · 2 minutes ago

3D Models	Update readme.md	yesterday
ARA Versions	Update readme.md	yesterday
SCHEMES	Update readme.md	50 minutes ago
git hub printables	Add files via upload	2 minutes ago
media	Create readme.md	2 weeks ago
robot logic and analysis	Create readme.md	2 days ago
robot photos	Create readme.md	yesterday
src	Create readme.md	2 days ago
team photos	Create readme.md	2 days ago
video	Create readme.md	2 days ago
README.md	Update README.md	yesterday

README

Seteki® Team 🤖

Beyond the future! 😎

`Looking Up`

INSTAGRAM YOUTUBE GITHUB

This is the official repository of the Seteki® team that will represent Instituto Episcopal San José in the WRO category: Future Engineers 2025, Panama.

Overview of our repository folders 📜

```
📦 WRO2025 Seteki
├── 📁 3D-models              # Contains 3D design files for the robot's components
│   ├── 📁 1.1                # 1.1 Models
│   ├── 📁 2.14               # 2.14 Models
│   ├── 📁 3.1 (National)     # 3.1 Models
│   └── 📄 README.md          # 3D models explained
├── 📁 Ara Versions           # Contains an explanation of every ARA Robot
│   ├── 📄 ara.md             # A tiny graph of the models
│   └── 📄 README.md          # Explanation of every model
├── 📁 SCHEMES                # All of the circuit
│   ├── 📁 Cirkit designer    # Interactive circuit
│   ├── 📁 Fritzing           # Our Circuit in Fritzing
│   ├── 📁 PCB                # Every Of the realization of our PCB
│   └── 📄 README.md          # Explanation
├── 📁 media                  # All our social media
│   └──📄 README.md           # Explanation
├── 📁 Robotlogic & analysis # How our robot is working
│   ├── 📁 Flowcharts         # Flowcharts with explanation
│   ├── 📁 Graphics           # Seteki random photo xd
│   └──📄 README.md           # Explanation
├── 📁 Robot Photos           # Robot videos
│   ├── 📁 1.1                # 1.1 Photos and explanation
│   ├── 📁 2.14               # 2.14 Photos and exlplanation
│   ├── 📁 3.1 (National)     # 3.1 Photos and explanation
│   └──📄 README.md           # 3.1 Photos
├── 📁 src                    # Robot videos
│   ├── 📁 First Challenge    # Codes of the first round
│   ├── 📁 Second Challenge   # Codes of the second round
│   └──📄 README.md           # Explanation
├── 📁 team photos            # Our photos as participant
│   ├── 📁 Funny photos       # All the funny photos
│   ├── 📁 Photos with tutor # Our photos with our tutor
│   ├── 📁 formal with tutor # Our photos with our tutor
│   ├── 📁 members            # There´s a lot of individual photos
│   └──📄 README.md           # Explanation
├── 📁 video                  # All the robot videos
│   ├── 📁 First Challenge    # All videos of first round
│   ├── 📁 Other videos       # Other explicative videos
│   ├── 📁 Second Challenge   # All videos of the second round
│   └──📄 README.md           # Explanation
├── 📁 GitHub Printables      # All PDF Githubs ready to print
│   ├── 📁 1.1                # 1.1 pdf
│   ├── 📁 2.14               # 2.14 pdf
│   ├── 📁 3.1                # 3.1 pdf
│   └──📄 README.md           # Explanation
└── 📄 README.md              # Main documentation for the project
```

> **Caution**
>
> Due to GitHub problems some charts and GIFs can load a little bit slow, so we beg you to be patient while reading to not miss anything, THANKS!

Part 1 / About Us 🗣️

We are Team Seteki, proudly representing Instituto Episcopal San José in Panama’s World Robot Olympiad (WRO) Future Engineers 2025 category. Our mission is to blend technology and innovation, knowing that innovation can go hand-in-hand with responsibility.

Our Philosophy 😎

We are a team in which there are no hierarchies in the team, we each develop our parts and we are all involved and we `KNOW` every part of the realization of this whole project that would be our robor ARA, we have values of humility and perseverance as a team, both when we work as a team as well as when we go to compete, we are a team in which two already have experience in what is robotics and specifically the WRO competition and we have a rookie in this world of what robotics is.

With our project we want to innovate and bring something that is well worked to this competition, we want the effort put into this robot to be noticed.

The truth is that our team is in a balance that we feel comfortable and perfect, each one develops a part that is essential for the other part, so we need each other and are values that apart from robotics and technological competition, are values that the WRO is also seeking to establish in all its competitors.

Without further ado, let's continue btw :)

Team Name Explication 🥸

The name of our Team (Seteki), comes from an animal called " ***Atelopus zeteki*** ". It is a small species of anuran amphibian of the Bufonidae family.

When we chose this name, it was because we are people who are using the possible resources to develop these robots and technology, but sometimes we forget that all this comes from natural resources and we are taking advantage of all this and we do not take care of it, and as a small annex we wanted to give this message that we also have a responsibility to the world and in addition to this we also want to have our country represented in our team.

> ⚠️ **Warning**
>
> This species is in danger of extinction, more information here.

Team presentation 😟

Jean Sosa

Name	Description & Rol	GIF	Age	IG
Jean Paul Sosa Cruz	Robot logic developer, 3D designer and 3D printing, Robot documentation as well as team workflow. for him all is music, and playing instruments, and has a good trayectory in robotics.		17	INSTAGRAM

Jonathan Sosa

Name	Description & Rol	GIF	Age	IG
Jonathan Michael Sosa Cruz	He is the mechanical genius of the team, he repairs and builds anything and he is in charge of all the robot's circuitry, he only thinks about soccer and getting home to play the bass guitar.		15	INSTAGRAM

Anthony Gomez

Name	Description & Rol	GIF	Age	IG
Anthony Bruce Gomez Arjona	He is the nerdiest of the three, he is the scientist and physicist of the team, he is the developer of the logic and he is in charge of the programming, talk to him about jurasic park and you will have all his attention.		17	INSTAGRAM

Edgar Ruiz

Name	Description & Rol	GIF	Age	IG
Edgar Antonio Ruiz Caballero	Active participant in robotics competitions such as RoboCup Jr , WRO, and others since 2015. - National Winner of RoboCup Jr 2017 and team coachrepresenting Panama in Nagoya, Japan. - Third Place National Winner in WRO 2018 as team coach for The Oxford School. - National Winner of RoboCup Jr 2018 and team coach representing Panama in Sydney, Australia in 2019. - Participant in 2021 STEAM training hosted by Grupo Editorial EDELVIVES in Madrid, Spain. - National Winner of WRO 2022 and team coach representing Panama in Dortmund, Germany. - National Winner of WRO 2024 (World Robot Olympiad) and team coach representing Panama in Izmir, Turkey.		26	

Robot Name Explication

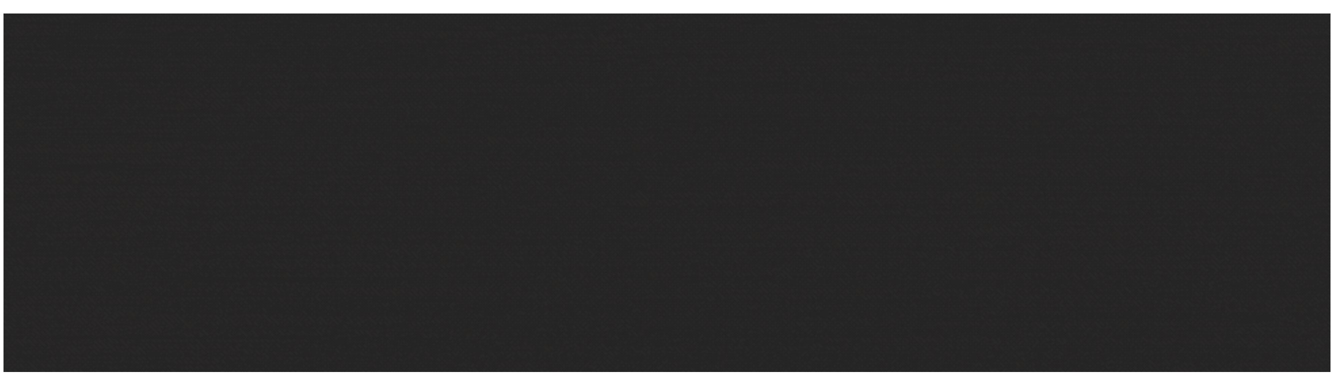
The name of all versions of our robot (ARA) comes from ***Ara macao***, practically the name of the macaw family. Macaws are large parrots measuring 46 to 51 cm in length and weighing 285 to 287 g in the case of the great macaw, and 90 to 95 cm and 1,708 g in the case of the green-winged macaw.

> ⓘ **Note**
>
> More information here.

Robot Versions 🚘

1.0	1.1	2.12	2.13	2.14
It has 3 ultrasonics, No differential	It's the same configuration than 1.1 version, but de 3D designs had a big improvement.	Has only camera, no ultrasonics, has a differential integrated.	This one has 2 ultrasonics and the camera, we quit differential to keed developing it	this one has 3 ultrasonics and a great improvement in 3D designs.

Chasis & 3D Designs 🚖



Platforms we use 🛹

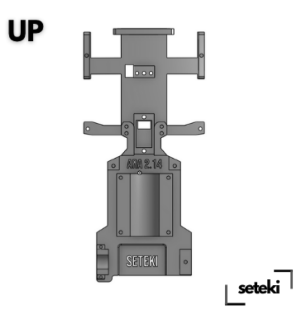
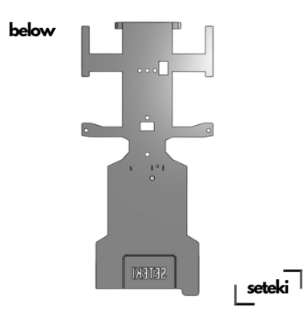
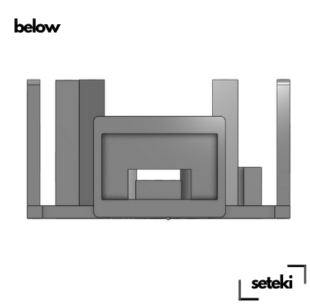
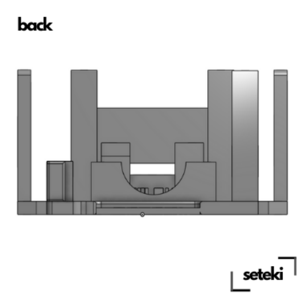
OnShape	Autodesk FUSION 360

Main chassis 🚄

Our chassis is fully printed with CREALITY PETG-CR filament material, as we want our chassis to be resistant and not wear out so quickly when implementing gear mechanisms.

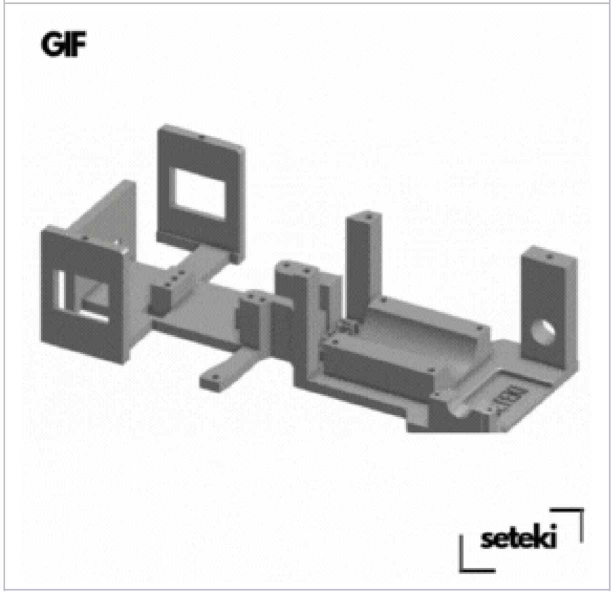
> ⓘ **Note**
>
> Click here To see the 3D Chasis part.

All Sides

ABOVE	BELOW	FRONT	BACK
			

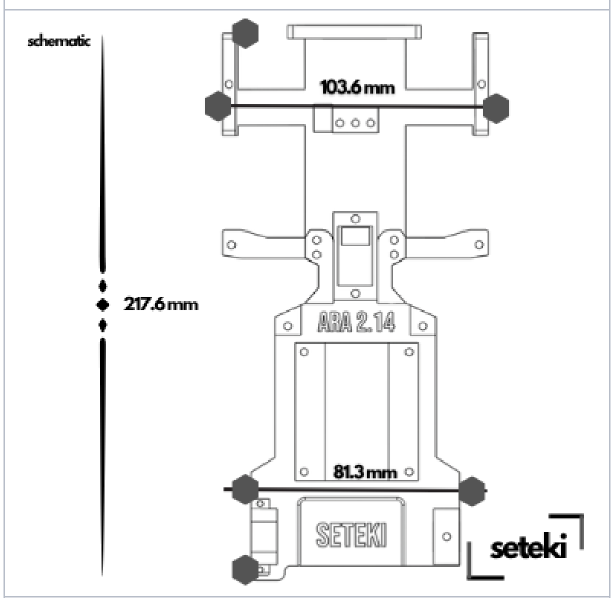
GIF View 3D

Animation



SCHEMATIC VIEW

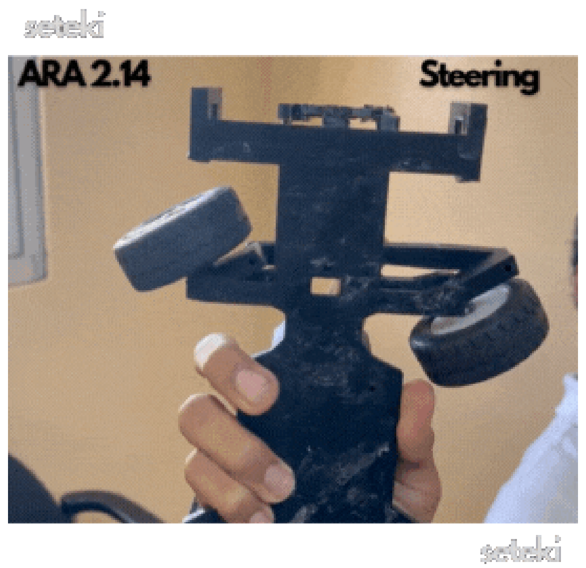
Schematic



Steering

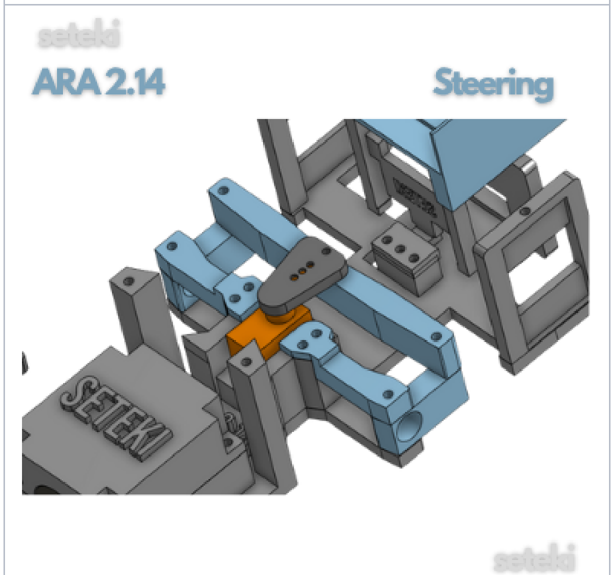
real view

physical



3D VIEW

3D

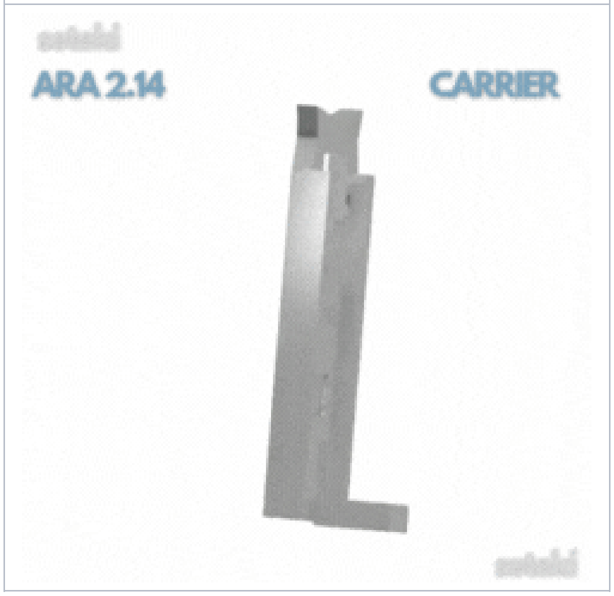


Camara Carrier

This is the support for the OPEN MV H7 Plus camera.

3D VIEW

3D



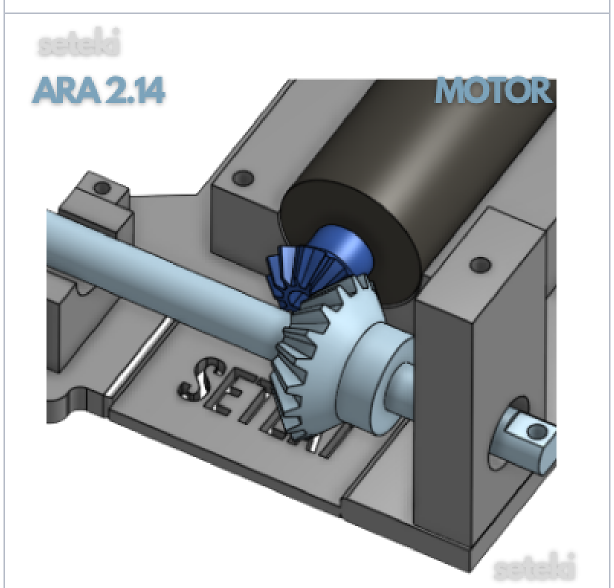
> ⓘ **Note**
>
> We use CREALITY PETG

Mobility management

We use a plñon and shaft that connects to a Pololu motor that consumes 12 volts, and we control it through a motor driver.

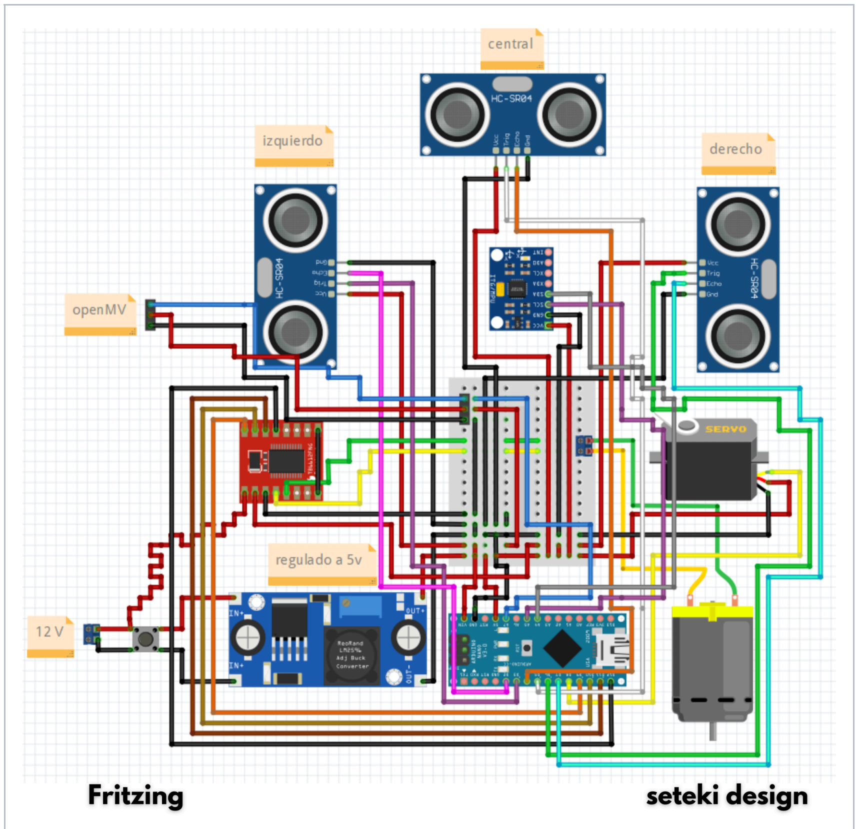
3D VIEW

3D



We control all of this by code.

Schemes



Scheme

Vehicle Components

```
Every robot electrical component
```

- `Ultrasonic Sensors` [3] HC-SR04
  - Buy Here
- `Motor Driver` [1] Puente H TB6612FNG 1.2A
  - Buy Here
- `Servo motor` [1] INJORA 7KG
  - Buy Here
- `Motor` [1] Metal Gearmotor 25Dx65L mm HP 12V with 48 CPR Encoder
  - Buy Here
- `Voltage Regulator` [1] LM2596
  - Buy Here
- `Interruptor Latching` [1] Tact Switch
  - Buy Here
- `Capacitors` [2] capacitors
  - Buy Here
- `Male female conectors` [12] XH-4Y
  - Buy Here
- `Gyroscope` [1] MPU-6050 (IMU Sensor)
  - Buy Here
- `Micro-controller` [1] Arduino NANO ESP32
  - Buy Here
- `Terminal` [1] PCB Terminal
  - Buy Here
- `Prototype wires` [32] Prototype wires

- Buy Here
- `Normal Wires` [11] Normal Wires
  - Buy Here
- `Camera` [1] Open MV H7 Plus
  - Buy Here
- `Lipo Battery` [1] Airsoft LIPPO Battery 11.1V
  - Buy Here
- `Ceramic Capacitors` [2] Ceramic Capacitors
  - Buy Here

Motor Code / Driver:

Outputs

```
const int pwma = 9;  // PWM pin for speed control
const int ain2 = 10; // Motor direction pin AIN2
const int ain1 = 11; // Motor direction pin AIN1
const int stby = 12; // Standby pin (if applicable)
```

These pins interface with an H-bridge motor driver, we do it with the `TB6612FNG` :

- pwma controls motor speed via PWM (Pulse Width Modulation).
- ain1 and ain2 determine the rotation direction:
- AIN1 = HIGH and AIN2 = LOW → Forward
- AIN1 = LOW and AIN2 = HIGH → Reverse
- Both LOW → Stop

Forward Movement:

```
void moverAdelante(int velocidad) {
    digitalWrite(ain2, LOW);
    digitalWrite(ain1, HIGH);
    analogWrite(pwma, constrain(velocidad, 0, 255));
}
```

- Sets direction pins for forward movement.
- analogWrite(pwma, velocidad) modulates motor power.
- constrain() ensures the speed remains within [0, 255].

Reverse Movement:

```
void moverAtras(int velocidad) {
    digitalWrite(ain1, LOW);
    digitalWrite(ain2, HIGH);
    analogWrite(pwma, constrain(velocidad, 0, 255));
}
```

```
 Inverts the direction pins to drive the motor backwards.
```

Motor stop:

```
void detenerMotor() {
    digitalWrite(ain1, LOW);
    digitalWrite(ain2, LOW);
```

```
    analogWrite(pwma, 0);
}
```

- Both control pins set to LOW to disable the H-bridge path.
- PWM set to zero ensures no motion.

Speed Profiles and Motion States:

```
const int velocidadNormal = 255;
const int velocidadGiro = 255;
const int velocidadEstabilizacion = 200;
const int velocidadCorreccionMin = 225;
const int velocidadCorreccionMax = 255;
```

These values determine PWM duty cycles in various robot states:

```
- velocidadNormal: cruising speed in NORMAL state.

- velocidadGiro: used during turning (EN_MANIOBRA).

- velocidadEstabilizacion: reduced speed post-turn.

- velocidadCorreccionMin/Max: used during CORRIGIENDO for ramping velocity.
```

Servo Code

Configuration:

```
const int servoPin = 8;
Servo steeringServo;
```

- The servo motor is connected to digital pin 8 on the microcontroller.
- Servo is an instance of the ESP32Servo library (compatible with ESP32 architecture).
- It allows PWM signal generation to control angle-based rotation (typically 0° to 180°) via hardware timer abstraction.

Servo Angle Constraints and Calibration:

```
const int servoNeutral = 90;      // Angle to go straight
const int servoMinAngle = 60;     // Minimum limit to avoid damage
const int servoMaxAngle = 180;    // Maximum limit to avoid damage
```

- `servoNeutral` defines the center (straight) orientation of the steering — critical for path alignment.
- `servoMinAngle` and servoMaxAngle serve as safety constraints to prevent mechanical overtravel or hardware strain.
- These bounds are enforced when sending `servo angles` in PID control logic

PID-Based Servo Control:

```
float setpoint = 0.0;          // Desired heading
float error = 0.0, prevError = 0.0;
float pidOutput = 0.0;         // Will be mapped to steering angle
```

- The error between current yaw and desired heading is calculated.
- A PID controller generates an output (`pidOutput`) proportional to that error.
- This output is mapped to a servo angle, like:

```
int servoAngle = constrain(servoNeutral + pidOutput, servoMinAngle, servoMaxAngle);
steeringServo.write(servoAngle);
```

- This allows real-time adjustments to robot steering using closed-loop feedback from IMU yaw data.

Ultrasonics Code

Confguration:

```
#define TRIG_IZQUIERDO 3
#define ECHO_IZQUIERDO 2
#define TRIG_CENTRO 5
#define ECHO_CENTRO 4
#define TRIG_DERECHA 6
#define ECHO_DERECHA 7
```

Each ultrasonic sensor uses two pins: one for triggering a sound pulse (TRIG) and one for receiving the echo (ECHO).

The robot has three sensors:

- Left (IZQUIERDO)
- Center (CENTRO)
- Right (DERECHA)

These are used to sense obstacles, corridor walls, or available openings on each side.

Reading Function:

```
float medirDistancia(int trigPin, int echoPin) {
    digitalWrite(trigPin, LOW); delayMicroseconds(2);
    digitalWrite(trigPin, HIGH); delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long duration = pulseIn(echoPin, HIGH, TIMEOUT_ULTRASONICO);
    if (duration == 0) return 999.9;
    float dist = (duration * 0.0343) / 2.0;
    if (dist < DISTANCIA_MIN_VALIDA || dist > DISTANCIA_MAX_VALIDA) return 999.9;
    return dist;
}
```

Filtering: Median Filter (Noise Reduction):

```
#define FILTER_SIZE 3
float lecturasIzquierdo[FILTER_SIZE];
float lecturasCentro[FILTER_SIZE];
float lecturasDerecho[FILTER_SIZE];
```

Each sensor uses a sliding window filter of size 3.

The filtering process involves:

- Storing new readings in an array.
- Using qsort() to sort the window.
- Rejecting invalid values (999.9).
- Returning the median valid value.

```
float calcularMediana(float arr[], int n);
```

- This eliminates transient spikes and false echoes.

First Challenge:

Explanation

IMU Sensor, configuration:

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
```

- The `MPU6050` is interfaced using the Adafruit unified sensor driver, abstracting raw register-level communication.
- The mpu object is an instance of `Adafruit_MPU6050`, which allows access to accelerometer and gyroscope data.

> Caution
>
> Due to GitHub problems some charts and GIFs can load a little bit slow, so we beg you to be patient while reading to not miss anything, THANKS!'

Gyroscope Offset Calibration:

```
void calibrarGiroscopio() {
    float sum = 0.0;
    Serial.println("Calibrando giroscopio...");
    delay(1500);
    for (int i = 0; i < numCalibSamples; i++) {
        sensors_event_t a, g, temp;
        mpu.getEvent(&a, &g, &temp);
        sum += g.gyro.z;
        delay(5);
    }
    gyroOffset = sum / numCalibSamples;
    Serial.print("Offset Giroscopio (Z rad/s): ");
    Serial.println(gyroOffset, 6);
}
```

- Purpose: Calculates a bias term for the Z-axis gyroscope by averaging static readings.
- `gyroOffset` represents the baseline drift due to sensor noise, temperature, or manufacturing variance.
- Result is stored in radians per second (rad/s).
- Critical to ensure accurate yaw integration in motion states.

Gyroscope Readings:

```
sensors_event_t a, g, temp;
mpu.getEvent(&a, &g, &temp);
```

This provides:

- `g.gyro.z` → Angular velocity about Z-axis (yaw rate)
- Output is in rad/s, which is converted to deg/s for most internal calculations.

Yaw Angle Estimation via Gyroscope Filter:

```
void calcularKalmanYaw(float gyroZ_rad, float dt) {
    float gyroRateCorrected_rad = gyroZ_rad - gyroOffset;
    float rate_deg = gyroRateCorrected_rad * 180.0f / PI - kalmanBias;
    kalmanYaw += rate_deg * dt;
    // ... Filter update math omitted ...
}
```

Logic / Explanation

The robot logic for the WRO Future Engineers first round is structured as a robust `finite state machine` (FSM) governing all motion and behavior through distinct states: `NORMAL`, `EN_MANIOBRA`, `ESTABILIZANDO`, `CORRIGIENDO`, `PARKING`, `ESQUIVAR_BLOQUES`, and DETENIDO. The system begins in the NORMAL state, driving forward using PWM signals `(analogWrite(pwma, velocidadNormal))` while continuously monitoring three ultrasonic sensors on the left, center, and right. These sensors provide noisy distance data, which is filtered using a median filter (e.g., `calcularMediana(` lecturasCentro, 3)) to improve measurement reliability. The robot uses these filtered distances to detect obstacles or openings; for instance, if the left distance suddenly increases compared to distancialzquierdoInicial multiplied by umbralRatioLateral, the robot interprets it as an available turn and enters the EN_MANIOBRA state.

Camera Code

```
Open mv h7 plus with open mv software
```

Configuration

```
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
```

- The camera sensor is initialized with RGB565 pixel format and QVGA resolution (320x240).
- Auto-gain and auto-whitebalance are disabled to ensure stable and consistent color tracking, as these automatic adjustments can interfere with blob detection in color space.

LAB Color Thresholds:

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package