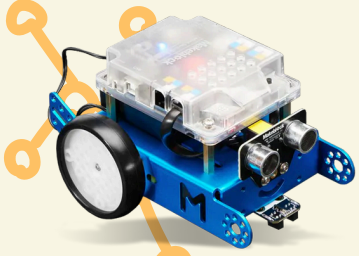
MERCEDES ORBANEJA

Eco-Robotics mBot - The Sweeper



Summary:

This code controls an mBot's behavior using inputs from **line follower** and **ultrasonic sensors**. Depending on the sensor readings, the robot, with its actuators, performs actions such as move forth in case **waste detected**, move back in case **white floor detected**, or turning right or left to **aling the robot with the waste found.**

Understanding these **sensors** and **functions** will help you grasp how the robot navigates its environment and **responds to different situations**.

Feel free to ask if you need further clarification or assistance with any part of the code! 😊

Our Mission:

Our mission is to **detect** all the trash from our **black area** and **clean** it, the sweeper robot **seeks** for any kind of waste and **push** and, as soon as detected, pushes the dirt away **wipin all the area without crossing the white line.**

Understanding the mBot:

The **mBot** is an educational robot designed to **introduce students to the world of robotics** and programming. It comes equipped with various **sensors** and **actuators**, allowing users to interact with and control the robot's behavior.

Key Features:

- **Modular Design**: The mBot features a modular design, allowing users to easily assemble and disassemble its components without the need for specialized tools.
- **Programmable**: Users can program the mBot using visual programming languages like mBlock or traditional text-based languages like Arduino.

- **Sensors**: The mBot comes with a variety of sensors, including line follower sensors, ultrasonic sensors, light sensors, and infrared sensors, enabling it to perceive and react to its environment.
- **Actuators**: It is equipped with motors, LED lights, buzzer...etc, allowing it to perform physical actions and provide feedback.
- **Wireless Connectivity**: Some versions of the mBot support wireless connectivity via Bluetooth or 2.4GHz wireless modules, enabling remote control and communication with other devices.

Let's break down some sensors used in this project:

Line Follower Sensor

Ultrasonic Sensor

Line Follower Sensor (port N2)

- This sensor is used to **detect lines on the ground**. It has two sensors, one on each side of the robot, to follow lines and **stay on track.**
- **In our project is used to keep the robot on the area inside the white line.**

Ultrasonic Sensors (port N4 and port N1):

- These sensors are used to **measure distance.** They emit ultrasonic waves and measure the time it takes for the waves to bounce back, allowing the robot to determine how far away objects are.
- **In our project is used to detect the waste, and place the robot exactly in front of the waste found so it can be properly pushed outside the white line.**

Understanding the code:

This code is written for an **mBot**, a robot that can be programmed using mBlock, either through **visual blocks or Arduino code**.

Functions:

- **move(int direction, int speed)**:

  This function is used to control the movement of the robot's motors based on the specified direction and speed (power).

- **delay(float seconds)**:

  This function creates a delay in the program for the specified number of seconds.

Setup:

- In the setup function, the program initializes various components and waits until a condition is met before proceeding.

Loop:

- The main logic of the robot's behavior is implemented in the loop function. It continuously checks sensor readings and executes corresponding actions based on the detected conditions.

:

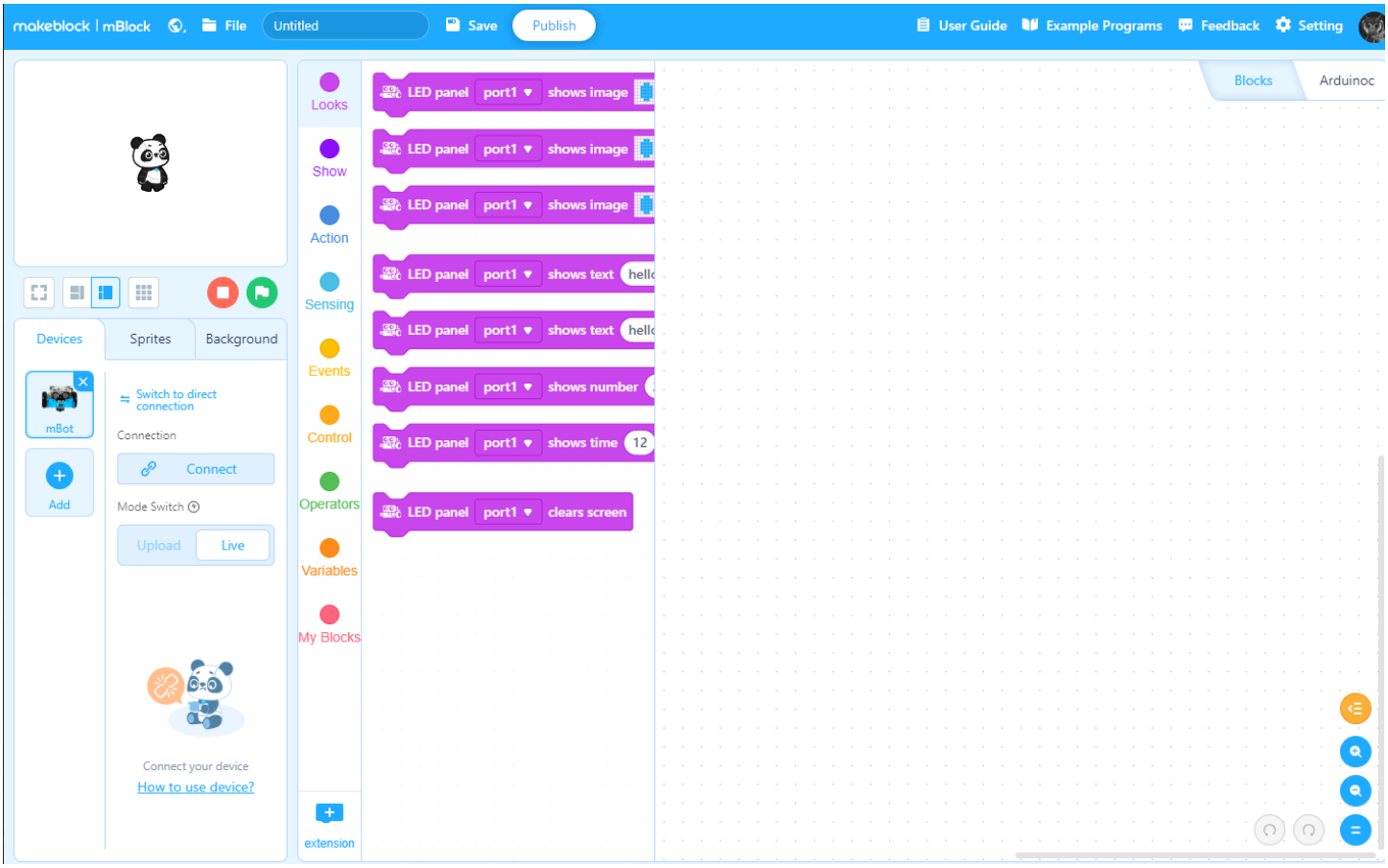
Block programming:

We are going to program through the mblock online platform:

https://ide.makeblock.com/

Interface

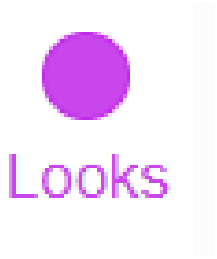
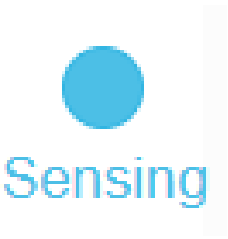
Let’s focus on Devices and the botton left, the blocks area at the middle and the programin area at the right.



Devices

select our robot mBot, delete any other device.

Blocks used for this project:

	1. **RGB LED Panel (not used in this project but applicable in case you feel so 😉)**: • **Description**: Controls the RGB LED panel on the mBot, allowing you to set its color, writte a text, a number or the time. • **Application**: Provides visual feedback, indicating the robot's status or actions, such as changing colors or showing messages..etc when detecting obstacles or following lines
	1. **RGB LED Block**: • **Description**: Controls the RGB LEDs (all, left or right) on the mBot, allowing you to set its color or plays a note • **Application**: Provides visual feedback, indicating the robot's status or actions, such as changing colors when detecting obstacles or following lines...etc
	1. **Move Block**: • **Description**: Controls the movement of the robot's motors based on the specified direction and speed (power). • **Application**: Allows the robot to move forward, backward, turn left, or turn right as per the detected conditions. Both motors are independent from each other.
	1. **Line Follower Sensor Block**: • **Description**: Represents the line follower sensor used to detect lines on the ground. • **Application**: Crucial for the robot to follow lines accurately, ensuring it stays on the desired path. 2. **Ultrasonic Sensor Block**: • **Description**: Represents the ultrasonic sensor used to measure distance. • **Application**: The robot uses these sensors to detect objects and determine their distance.
	1. **When mBot (mcore) starts up Block**: • **Description**: Initiates the program in the mBlock environment. • **Application**: Starts the robot's operation, allowing it to respond to sensor inputs and execute commands.
	1. **Forever Block**: • **Description**: Creates a loop that continuously executes its contents. • **Application**: Ensures that the robot continuously monitors its sensors and performs actions based on the detected conditions.

	2. **If Block**: • **Description**: Checks a condition and executes the enclosed code if the condition is true. • **Application**: Enables the robot to make decisions based on sensor readings, such as changing direction when detecting an obstacle. 3. **Wait until:** • **Description**: Stops to program to run until a sensor is receiving the commanded signal. • **Application**: Allows to control the robot until a signal is manually given, in this project you have to press the on-board button and any sensor is to detect something (our hand).
	1. **Comparison Operators** (e.g., >, <, ==): • **Description**: Compares two values and returns true or false based on the comparison result. • **Application**: Used in conditional statements to compare sensor readings or variable values, determining the robot's behavior accordingly.
	1. **Variable Blocks** (e.g., LeftEyeP1, RightEyeP4): • **Description**: Represents variables used to store and manipulate data within the program. • **Application**: Stores sensor readings or intermediate values calculated during program execution, allowing for complex decision-making and control flow.

By **categorizing** the blocks and including the usage of variables, this guide provides a comprehensive overview of the mBlock components and their applications within the provided code.
If you have any questions or need further clarification on any block or concept, feel free to ask!

Programming:

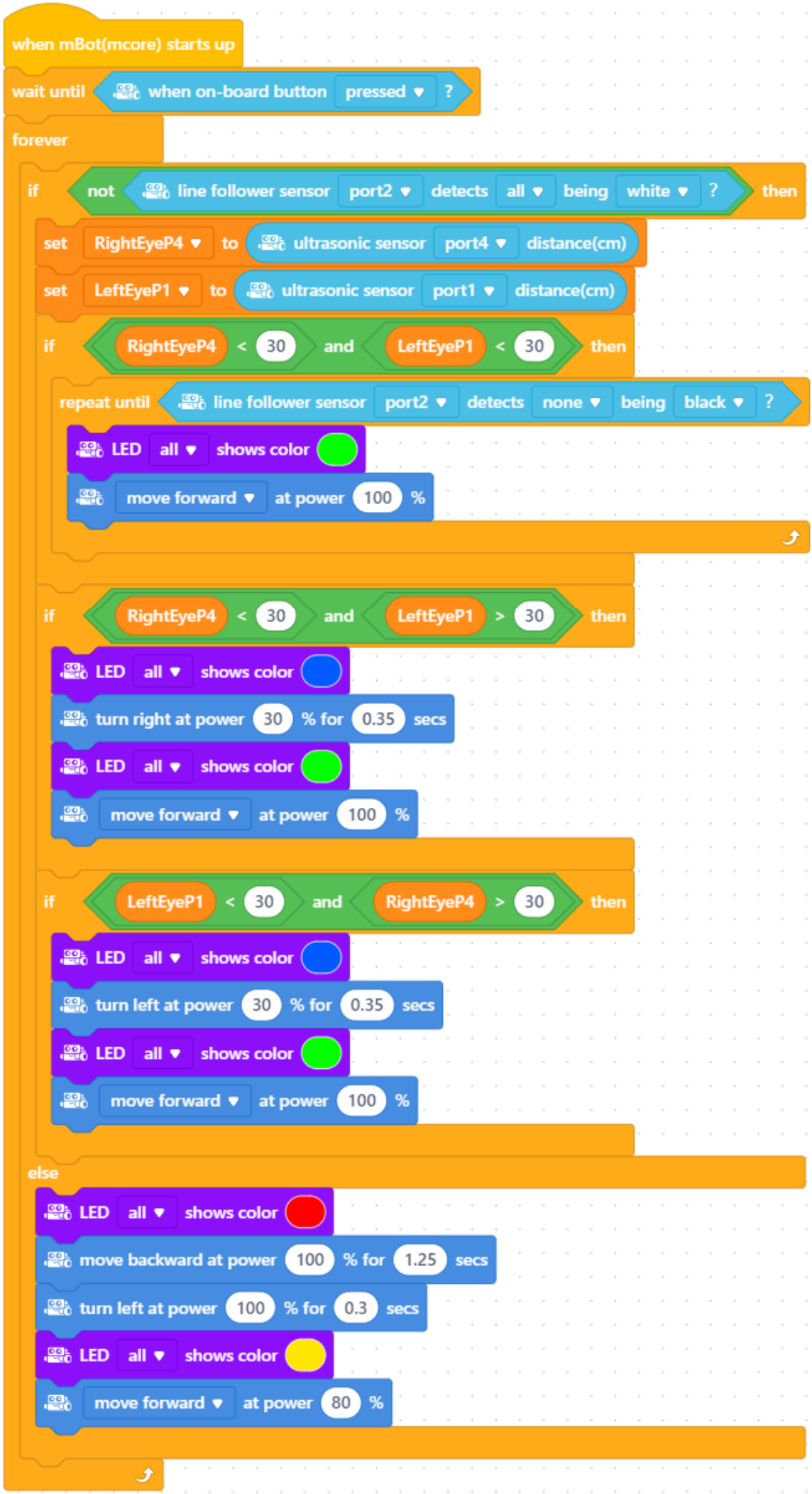
To implement this code using mBlock's visual programming interface:

1. Follow the steps given by the trainner, don't worry, you have the complete block program below in case you get lost
2. Drag and drop the appropriate blocks onto the workspace.
3. Use control blocks to create conditional statements based on sensor readings.
4. Combine movement blocks with these conditions to control the robot's behavior.
5. Utilize the delay block to introduce pauses or timing constraints as required.
6. Optionally, incorporate RGB LED blocks to provide visual feedback on the robot's actions

By assembling these blocks and configuring them according to the logic outlined in the code, you can replicate the functionality in mBlock's visual programming environment.

Feel free to reach out if you need assistance with specific blocks or their configurations!

Block program:



That's all:

I hope you enjoyed the workshop 💚

Mercedes Orbaneja Fernández
Secondary Education Science Teacher
Electric Engineer

Further informacion and tutorials:

https://www.youtube.com/watch?v=VYBofYoZttI
https://www.youtube.com/watch?v=TdiT4OPUvcc
https://www.youtube.com/watch?v=O5krkiwYK6A
https://www.youtube.com/watch?v=1jdigNfGXOQ