AI FARM ROBOTICS

RoboCar

With the

micro:bit

Let`s make any movement of RoboCar with your Micro:Bit.

AI farm robotic

1. DESCRIPTION

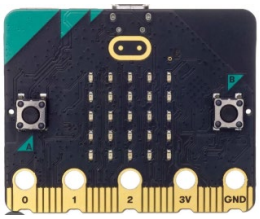
Micro:bit is significantly applied to STEM education, as a small microcontroller, which features small in size, easy to carry, and powerful function. At present, innovative technology products, like robots, wearable devices and interactive electronic games can be produced by programming and code. In this kit, we will guide you how to control and generate a Micro:bit **RoboCar** through programming in Makecode.

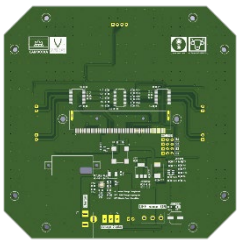
MakeCode is a framework for creating interactive and engaging programming experiences for those new to the world of programming. The platform provides the foundation for a tailored coding experience to create and run user programs on actual hardware or in a simulated target.

2. SPECIFICATION

- Application sector: Education Sector
- Type: Omni Wheel
- Voltage Requirement: 7.4 Volts (V) DC
- Battery life: 15 minutes
- Microcontroller: Micro:Bit
- Actuators: DC Motor 6 Volts (V)
- Communication: Radio Frequency
- Payload Capacity: Mac 0.5 kg
- Rechargeable: Charge controller system
- Additional Feature: Buzzer, RGB, Servos.

3. KITs

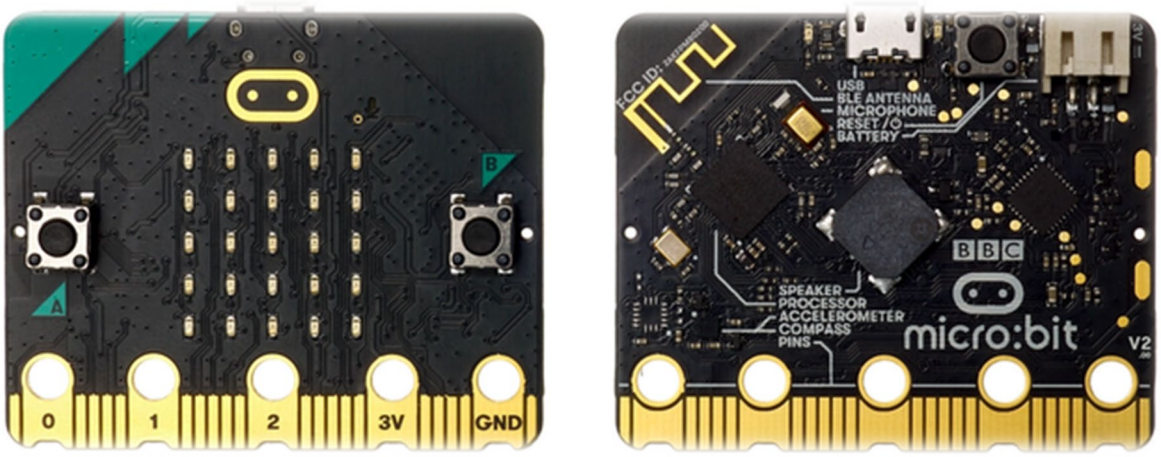
COMPONENTS			
#	Model	QTY	Picture
1	Micro:Bit main board	2	

2	Main board RoboCar	1	
3	DC motors	4	
4	Battery 7.4 V	1	
5	USB Cable Micro-USB	1	
6	Omni Wheels	4	

7	Remote Control (Joystick & Button)	1	
8	2-slot battery holder 1.5V AAA battery	1	
9	Battery 1.5V AAA	2	
10	Screwdriver	1	

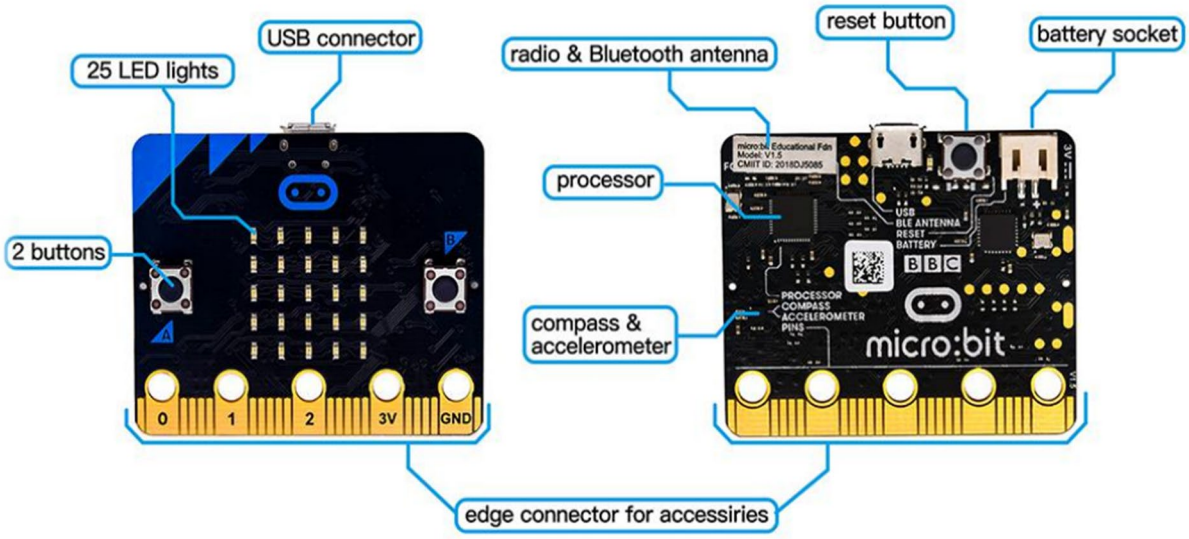
4. MICRO:BIT

4.1. Micro:Bit



- For more information, enter the Website: [Micro:Bit guide]

4.2. Functions



Features:

- NRF511822 processor (16Mhz 32bits, ARM cortex-M0, Bluetooth 4.0 low consumption/2.4GHz RF wireless, 16KkB RAM and 256kB flash)
- KL26Z microcontroller (48Mhz ARM Cortex-M0+ core, 128kB flash)
- 25 pcs programmable LEDs
- 2 programmable buttons
- Physical pins
- Light and Temperature sensor
- Accelerometer (MMA86532 and I2C get the data from accelerator sensor)

- Geomagnetic sensor/compass (MAG3110, I2C obtain three-axis geomagnetic data)
- Wireless communication, by radio and Bluetooth.

Micro USB port:

- For more detail, go to: https://microbit.org/getstarted/features/overview/

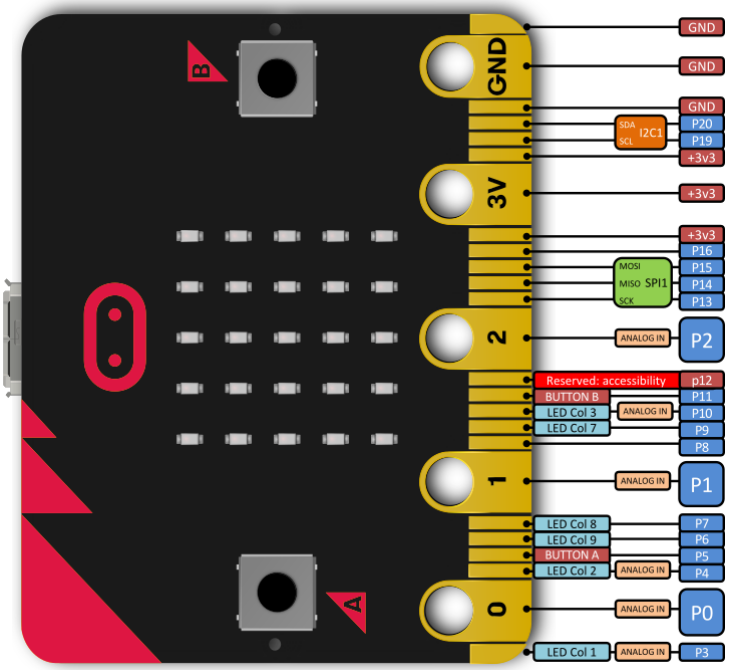
Hardware:

- Micro:bit board`s details: https://tech.microbit.org/hardware/

Micro:Bit`s pins:

Before getting started with the following projects, firstly need to figure out each pin of micro:bit main board. The BBC micro:bit has 25 external connections on the edge connector of the board, which we refer to as "pins". The edge connector is the gray area on the right side of the figure below. There are five large pins, that are also connected to holes in the board labeled: 0, 1, 2, 3V, and GND. And along the same edge, there are 20 small pins that you can use when plugging the BBC micro:bit into an edge connector.

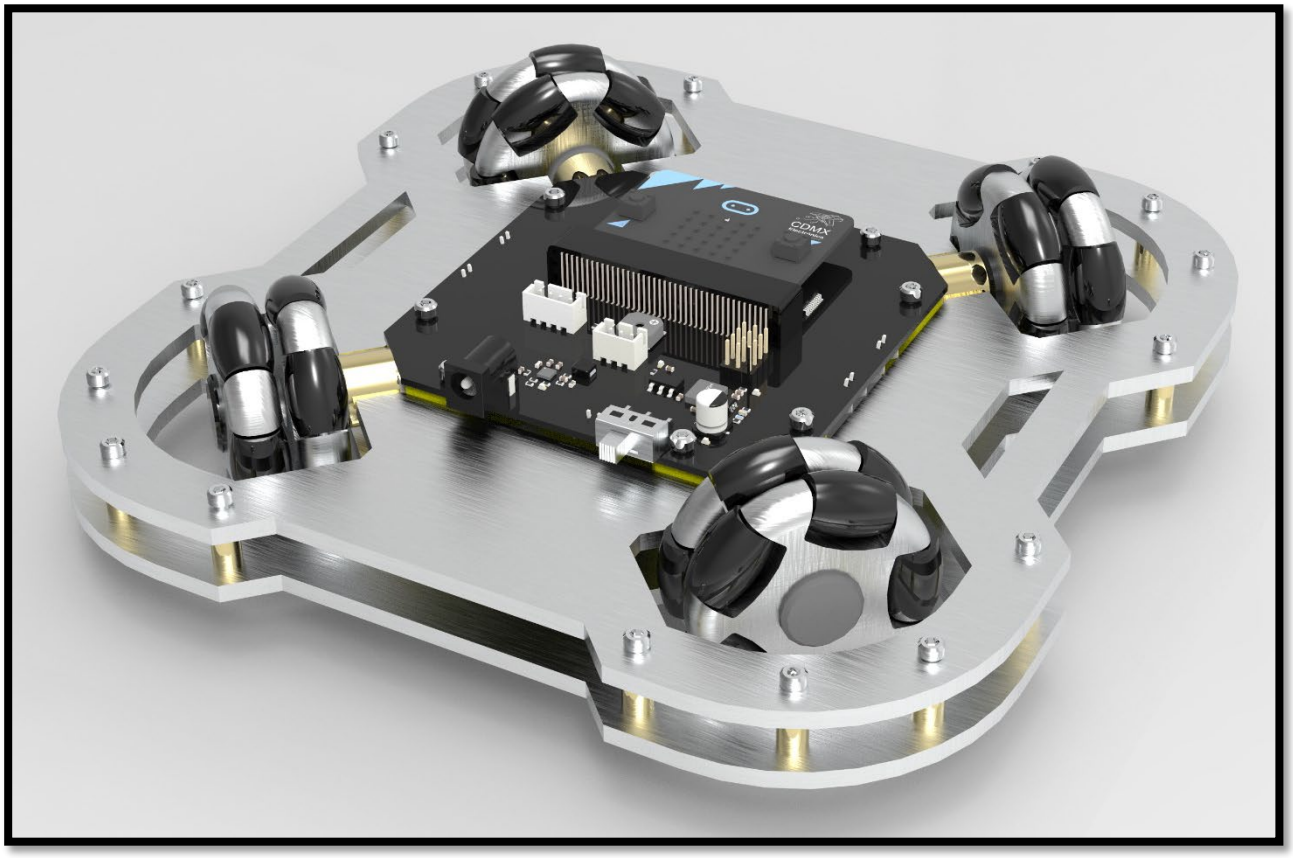
Please refer to the figure as shown below:



More details you can go to website: https://microbit.org/getstarted/features/overview/

5. RoboCar Connections to Micro:Bit

5.1. Connections



Connections of Micro:bit to RoboCar:

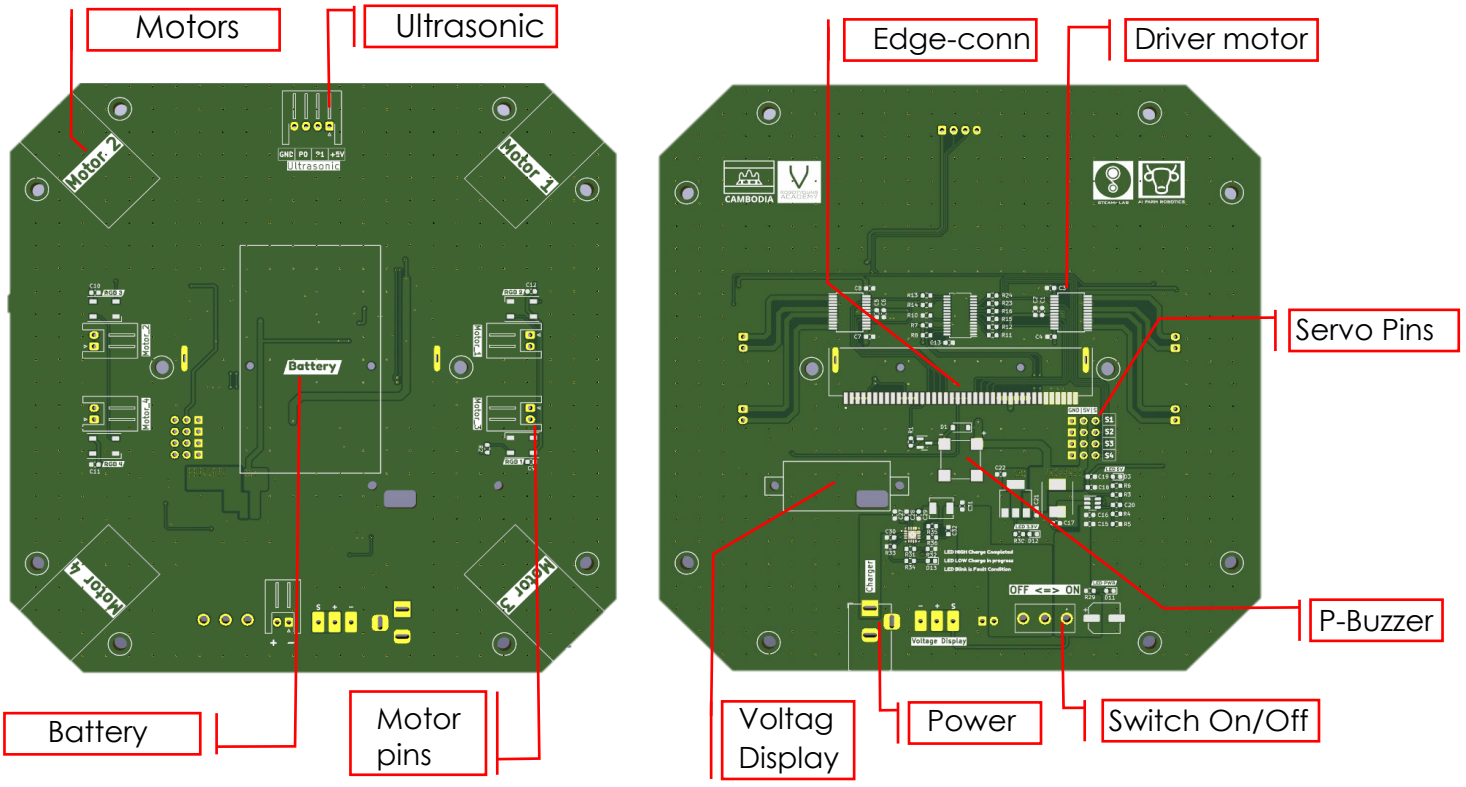
Micro:bit`s pin	Components of pin connected to
P13, P14	Motor 1
P15, P16	Motor 2
P9, P10	Motor 3
P11, P12	Motor 4
Driver Pins 2	Servo 1
Driver Pins 3	Servo 2

Driver Pins 4	Servo 3
Driver Pins 5	Servo 4
P1	Trig (T) of ultrasonic
P0	Echo (E) of ultrasonic
P8	Passive Buzzer
P2	RGB

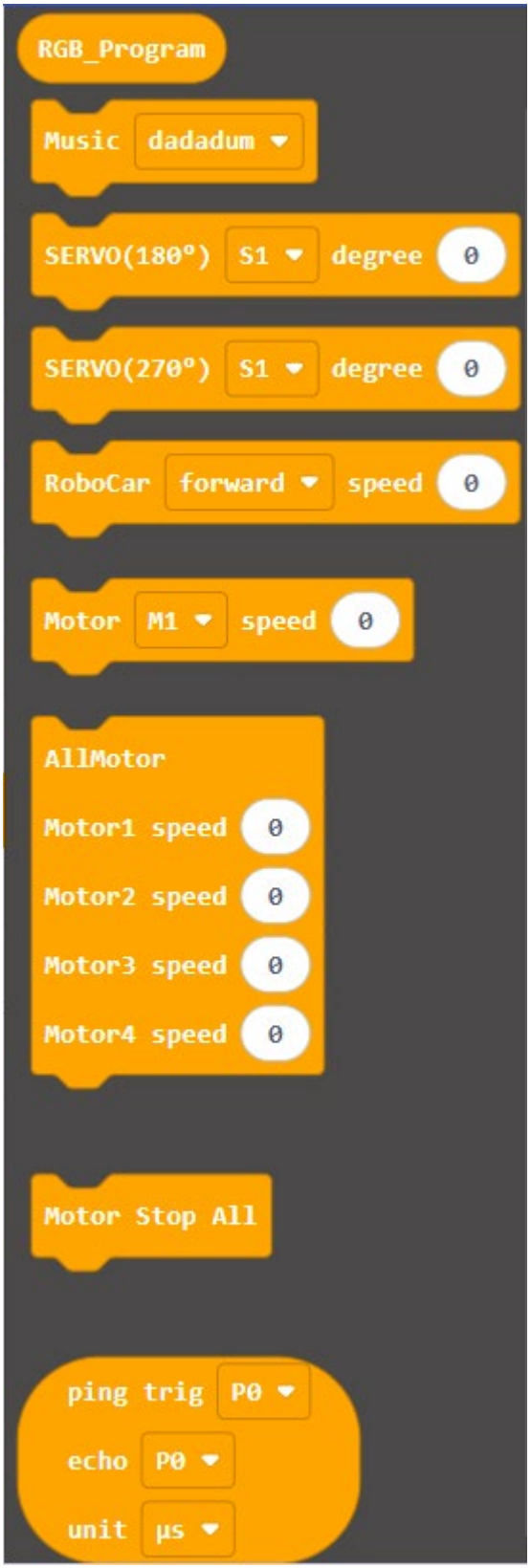
Power Supply

- This smart car is powered by battery (from 7.4V – 8.4V max).

5.2. PCB`s RoboCar function



5.3. RoboCar Library



1. **RGB_Program**: Control RGB light on RoboCar can be switch to many colors such as Red, Blue, Green…See example.

2. **Music(dadadum)**: this block can be control and choose many sounds in microbit.

3. **Servo (180 & 270) degree**: This block use to control servo motor which can turn to 180degree and 270 degrees.

4. **RoboCar (forward) speed**: This special one on RoboCar to control the movement forward, backward, leftside, or rightside… with speed (0- 255).

5. **Motor (M1) speed:** This block use to control the movement of each motor if you want to test the direction of motor with speed.

6. **AllMotor:** This block can be similar to Num 5, but this block can be added to all motor at the same time.

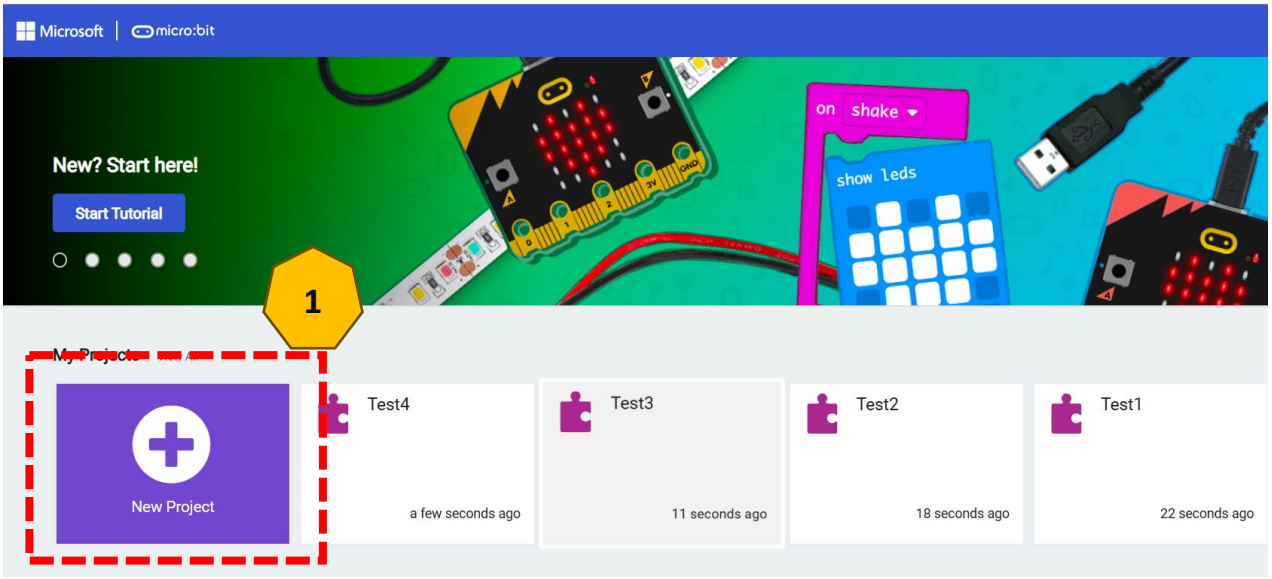
7. **Motor Stop All**: To stop all the movement of motor we can use this block.

8. **Pin trig [P0]**
   Echo [P0]
   Unit [cm]: Is the block that we use to control with module ultrasonic that we have built on RoboCar.

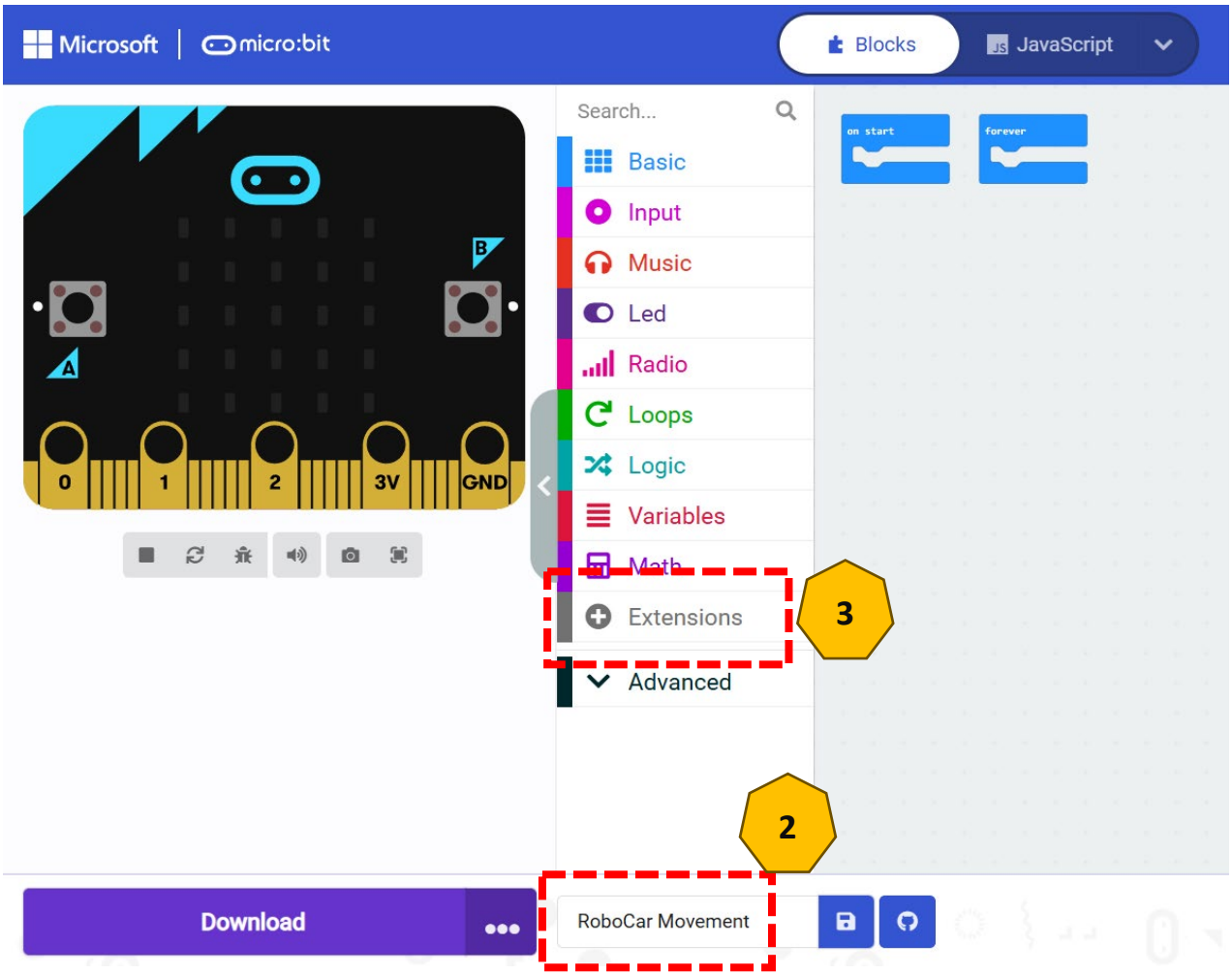
- RoboCar`s library, this library provides a set of tools easily program and control RoboCar using MakeCode for Microbit. This library was built to support such as Music, Servo 180 and 270degree, 4 wheels control movement such as (Forward, backward, leftside, rightside, rotateright, and rotateleft…), Each motor control with speed, all motor control with speed, and stop functions, Sensor Integrations, and Support wireless communication and advanced robotic_feature.

➢ Installation library of RoboCar extension

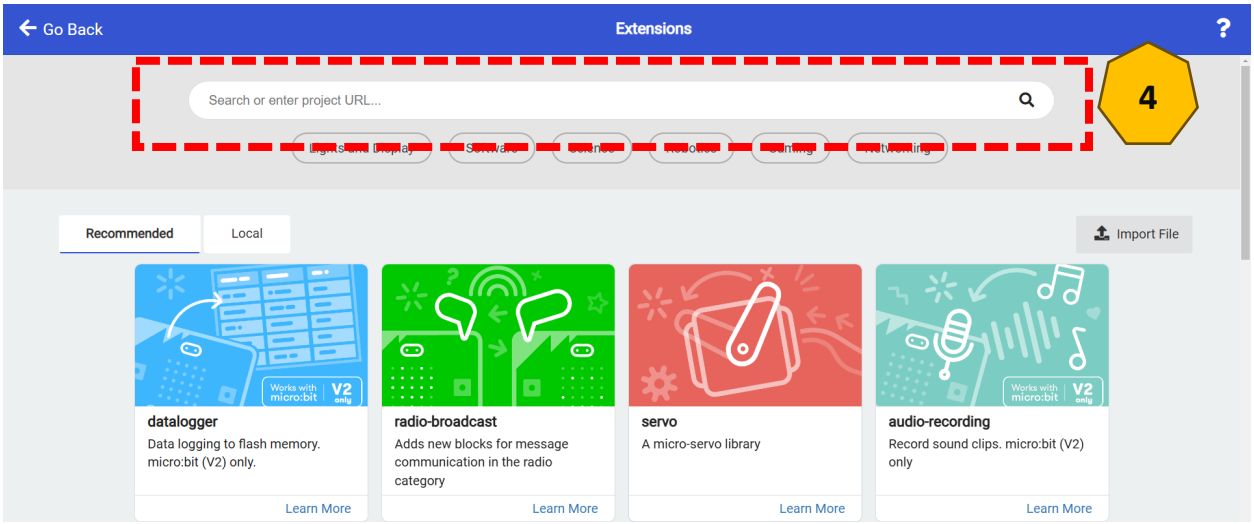
- Open the platform such Google Chrome, Safari, firefox, Opera, Brave (recommended Google Chrome) and go to MakeCode Editor.



- Click [+New Project] and then name the project to "e.g. RoboCar Movement" and you will see the Tab below

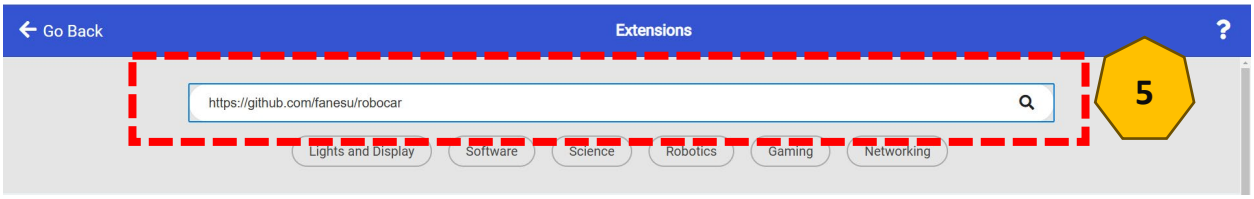


-After create and named it already, go to **"EXTENSION"** and you will see the next slide below, go to search bar "Search or enter project URL…"

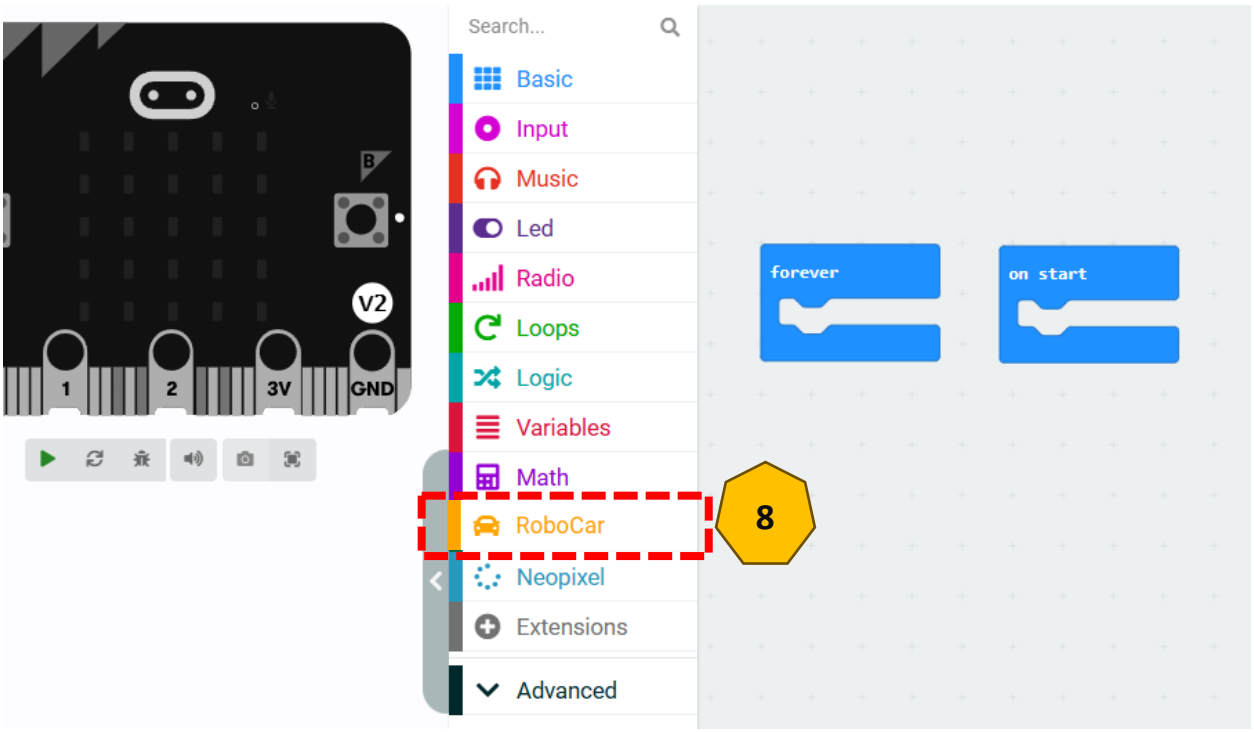


- click on search bar, Copy and Paste this extension below:

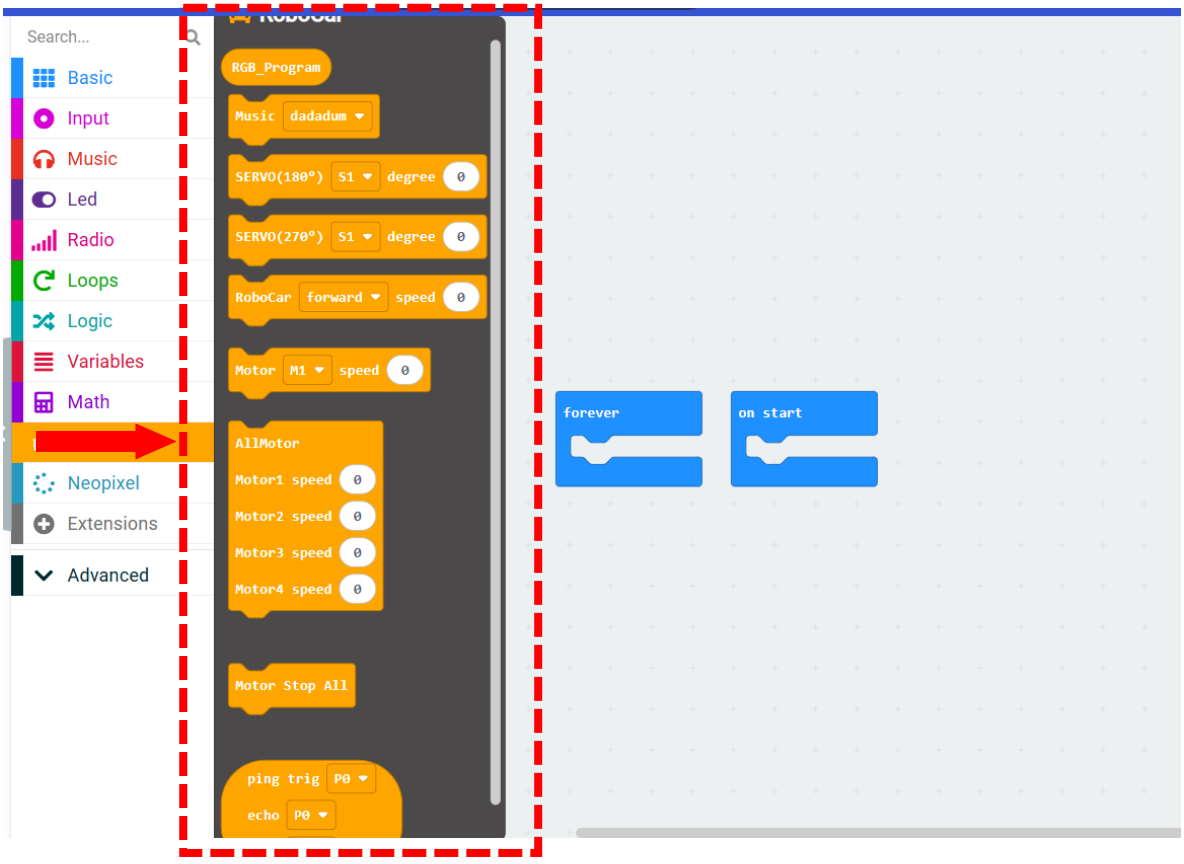
----------https://github.com/MENG-Vathanak/ROBOCAR_CONTROL.git----------



- Click "Search" or this sign  then you will see the extension, then click on the extension again.



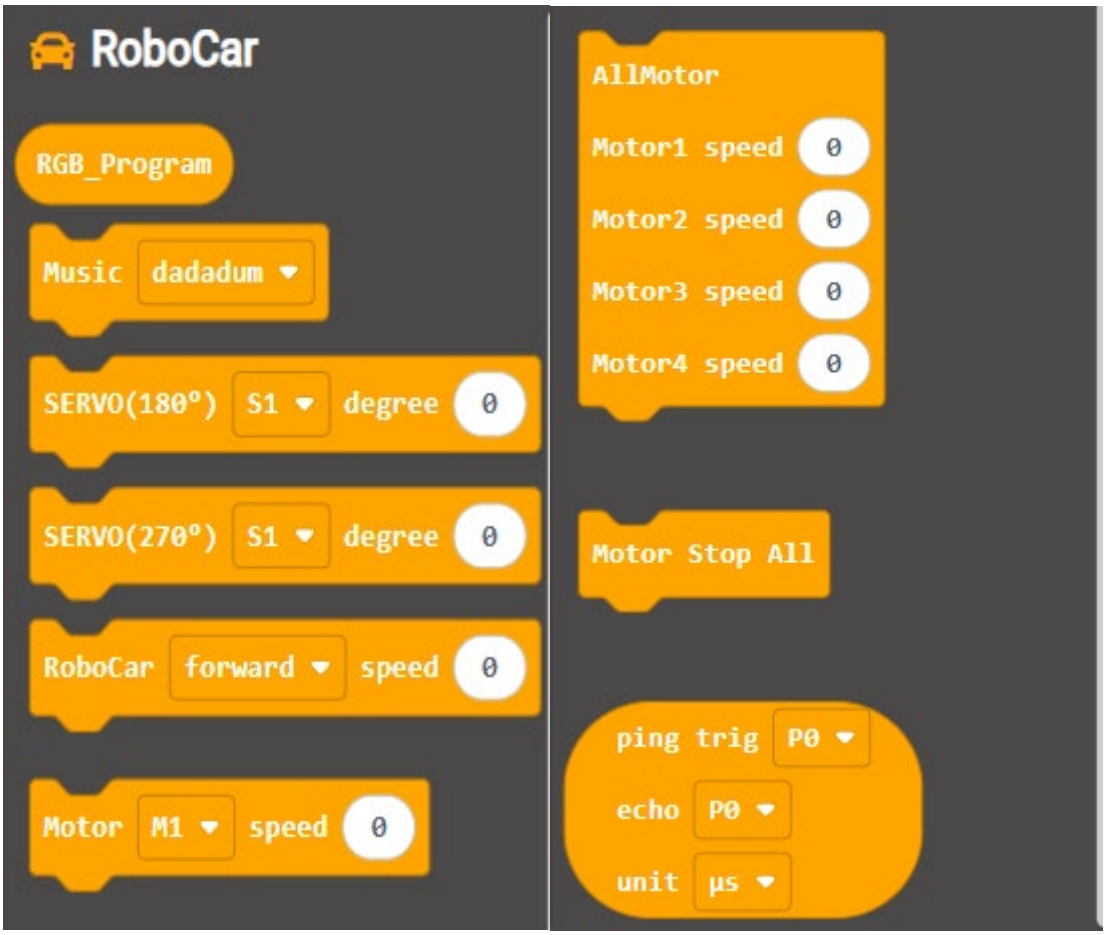
- Click on this "RoboCar" and you will get this



Note: The extension added into the project can be used for only one project, next time you create the new project add the extension as the process agai

5.4. Overview of Library RoboCar

- In the library, RoboCar has 9 block which is having different function such as:

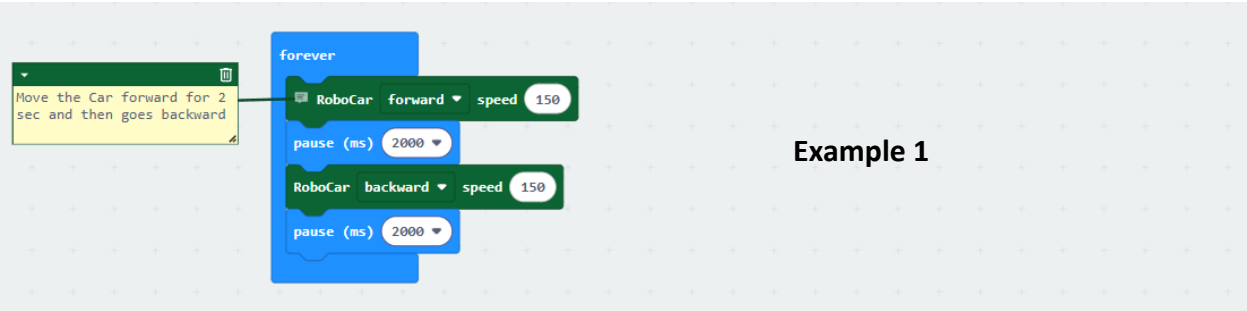


- RGB Car program: Control the RGB light built in RoboCar
- Music: play the "Music list" with passive buzzer on Microbit.
- Servo (180) and (270): control the servo motor with up to 4 servo motors.
- RoboCar [Movement] speed (0): Control the movement of 4 wheels omni with already set the direction such as (Forward, backward, leftside, rightside, rotateleft, rotateright, forwardleft, forwardright, backleft, and backright)
- Motor [M1] speed (0): this block was built to control the direction of each motor by changing the motor pins [M1, M2, M3, and M4] with speed (-255 to 255).
- All motor: This block was built to control all 4 motors with different speed at the same time.
- Motor Stop all: this block was use to stop all the motor movement.

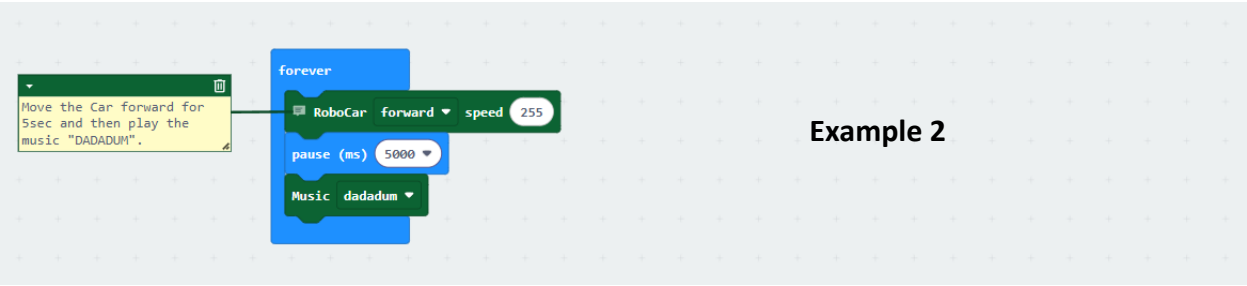
6. Projects

Let`s dive into the example of "RoboCar"

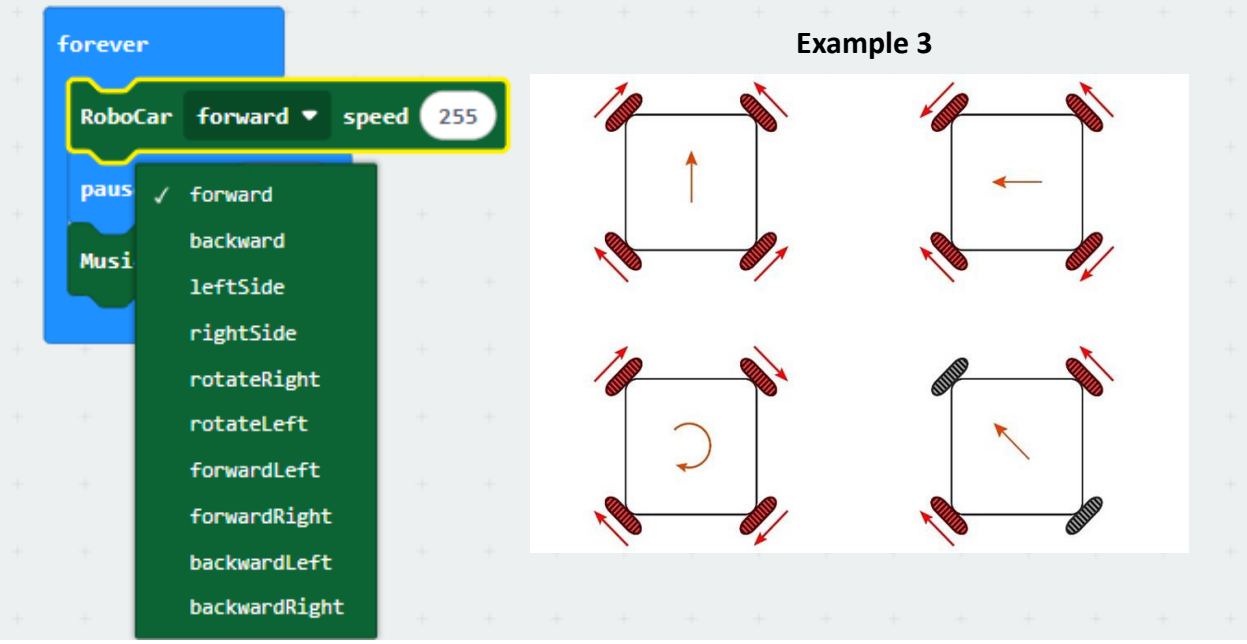
Example: **Basic** with DC motor Control



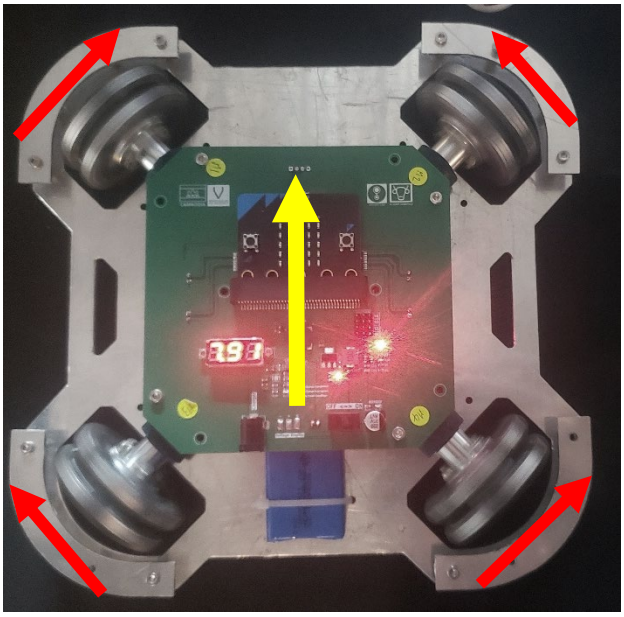
- Move the Car Forward and Backward automatically:

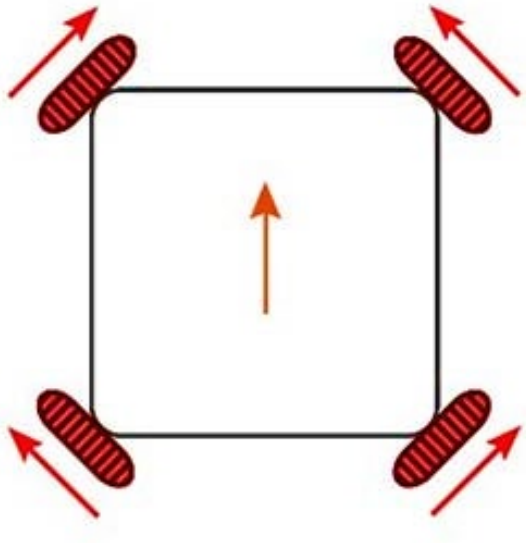


- Move the car Forward for 5 second with 255 speed (maximum) and then play the music name "Dadadum".
-

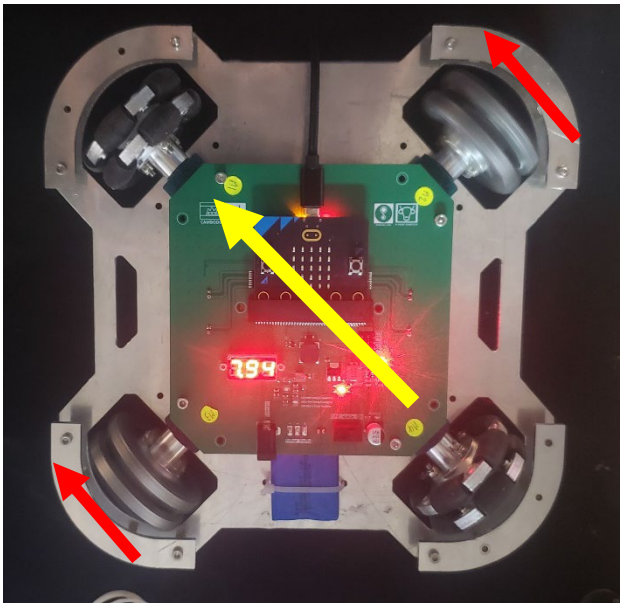


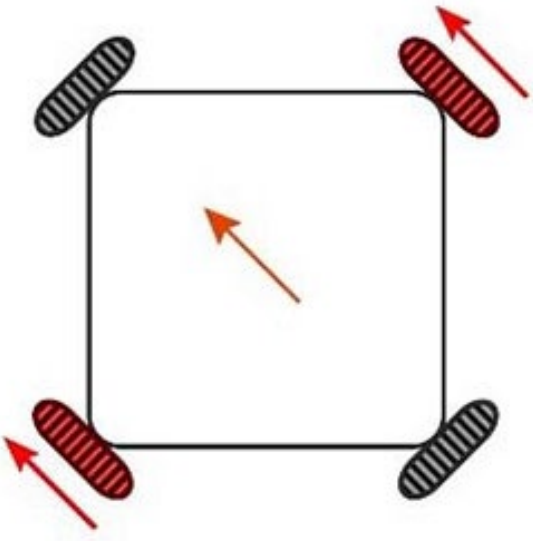
- Switching mode of RoboCar, there 10 movements in RoboCar. If you want to control it you have to set the front of RoboCar. So, you can easily set the direction and control the RoboCar. Example in this point I have set the Microbit as front or face of the car and I have to control the forward of the RoboCar



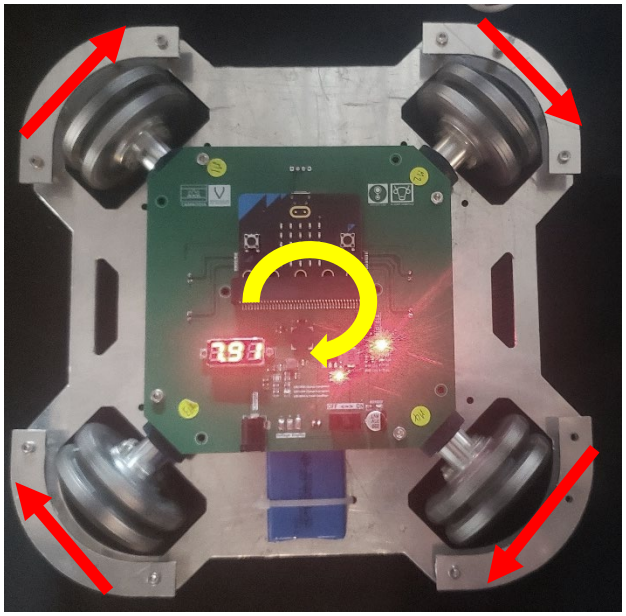


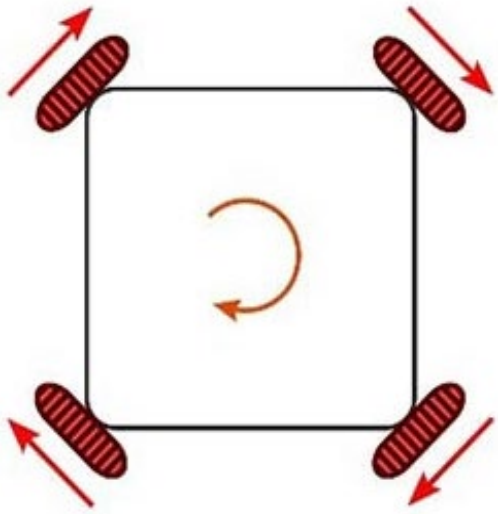
Forward direction





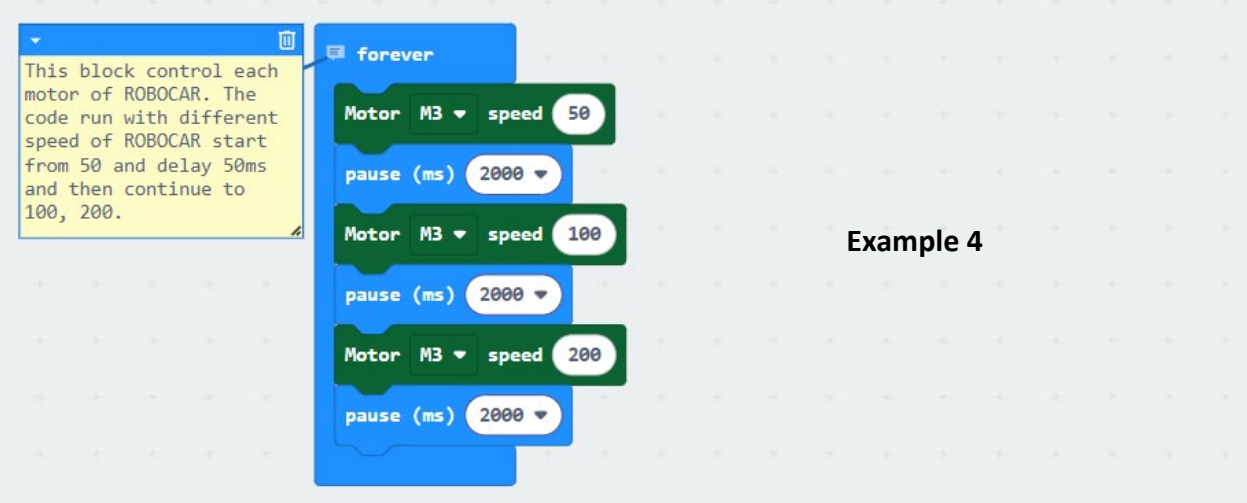
Forwardleft direction





RotateRight direction

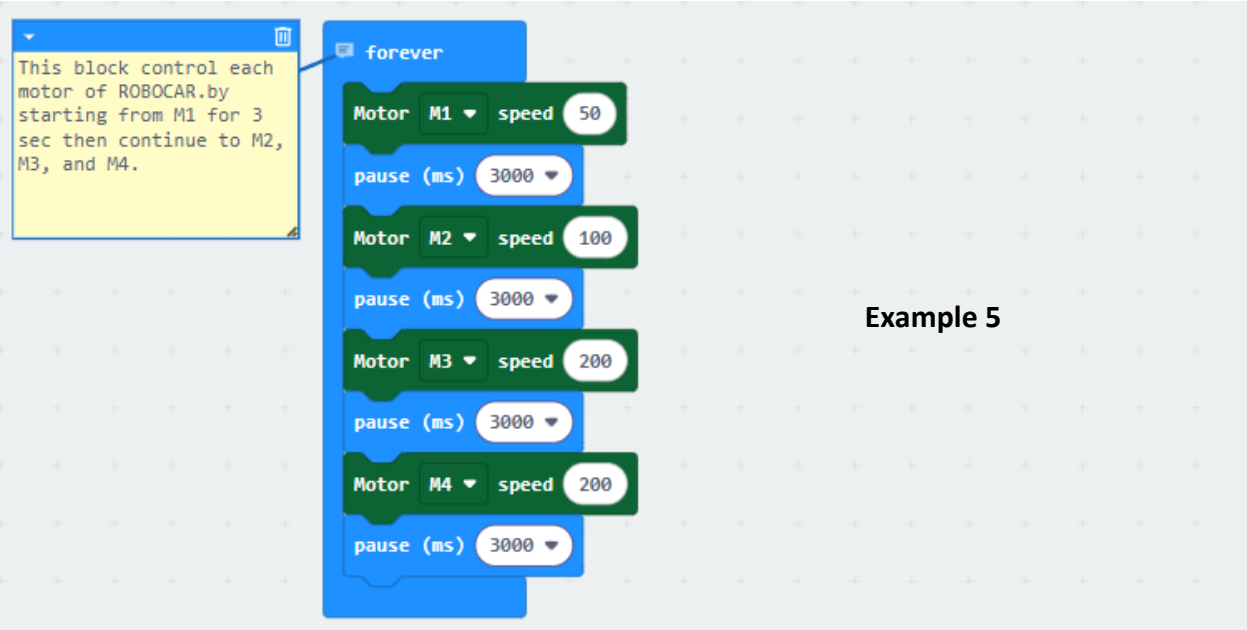
- Each motor control:



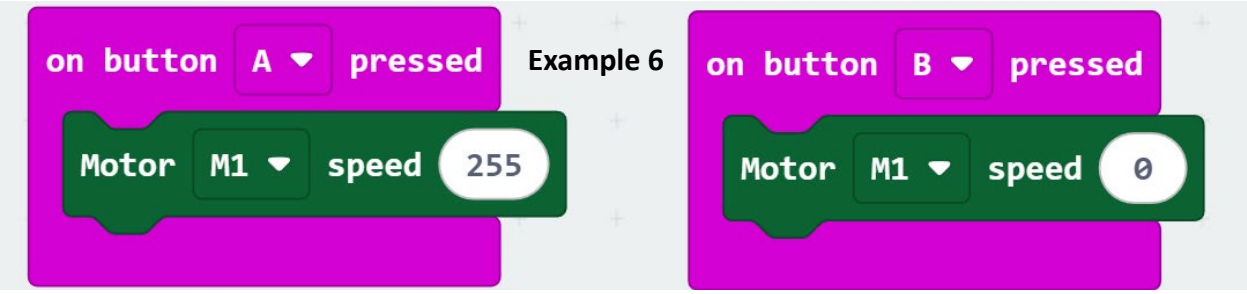
Example 4

- In this code, show the M3 motor control by changing the speed with delay 2000ms.its start from speed 50 for 2 sec then continue to 100 then 200. We can change this Motor number select the [M1, M2, M3 and M4] of each motor.
  In this block you can control all 4 motors with different direction and speed to any movement and speed by yourself.

- Motor 1, 2, 3, 4 control:



- **Button (A, B)** control with RoboCar:

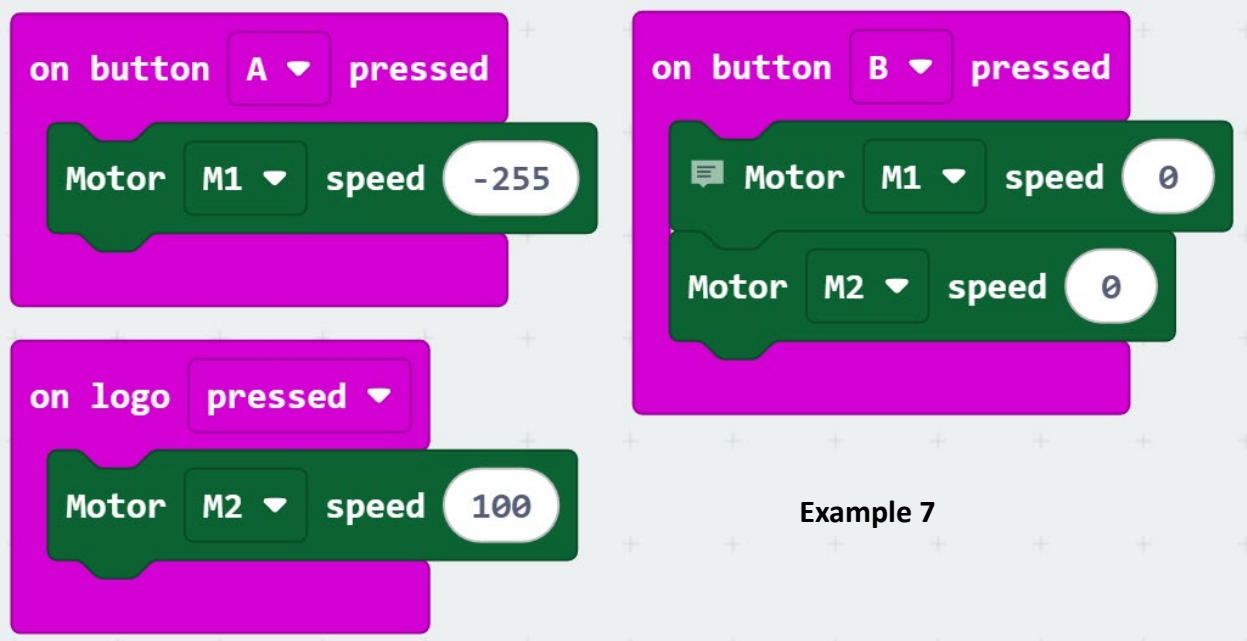


When A pressed, the motor start to run with speed 255. When the B pressed, the motor stop. We also can adjust the speed to (-255) to make motor into another direction as below.

The value of motor 1 run between (255 and 0), its call Counter Clock Wise, and the value between -255 and 0 it`s call Clock wise.

Clockwise

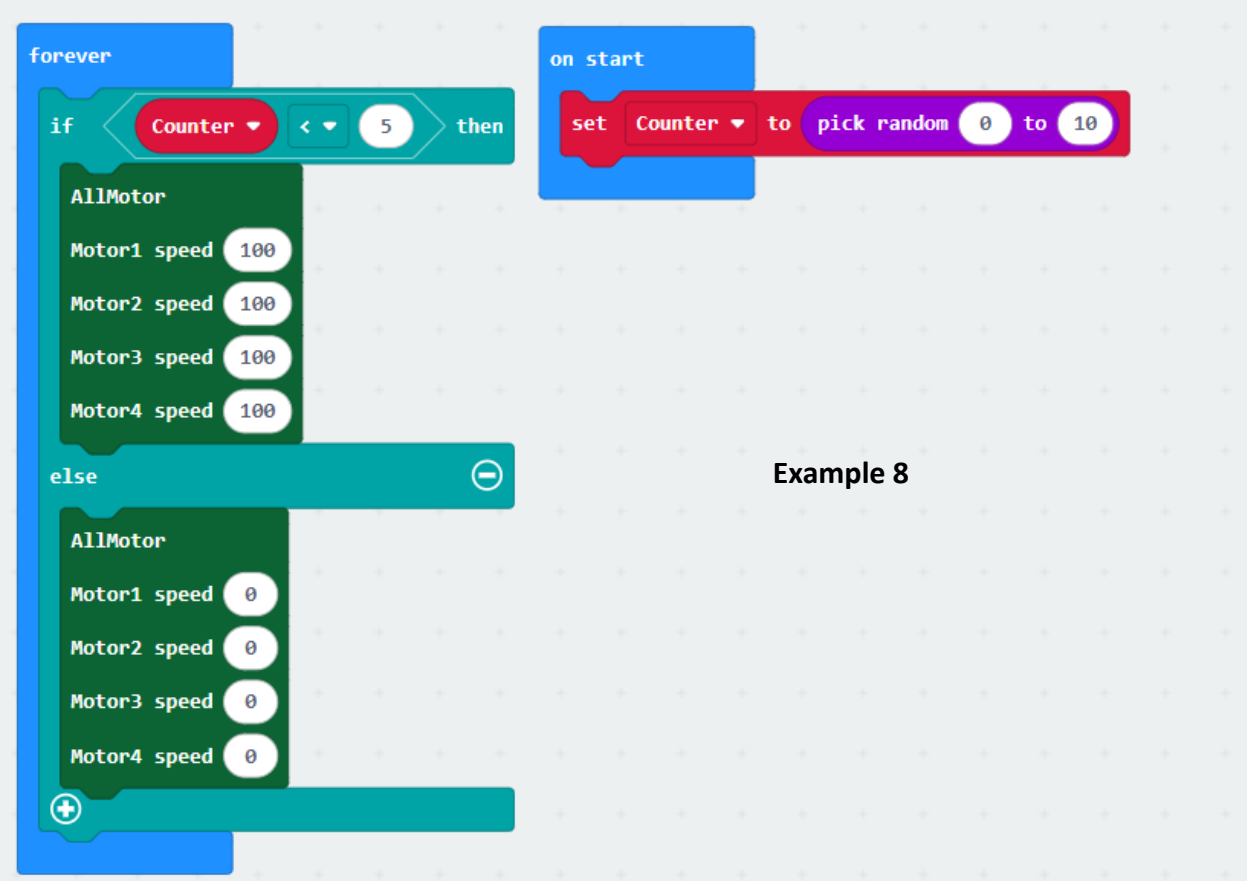
Counter-Clockwise



Example 7

- Button and Logo controlling motor, the button A pressed start the motor M1 and, on logo pressed the motor M2 start, on button B Is to stop all the motor. We can do this for all four motors by adjust the direction and speed by using inputs such as button A, B, logo, loud sound to control any movements.
  Note: some function such Loud and Logo support Micro:Bit V2.

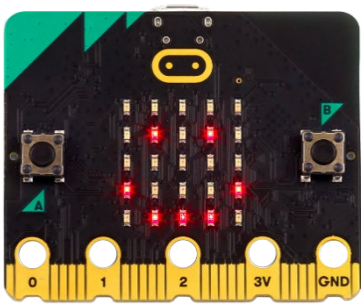
- **Logic** with Motors control



Example 8

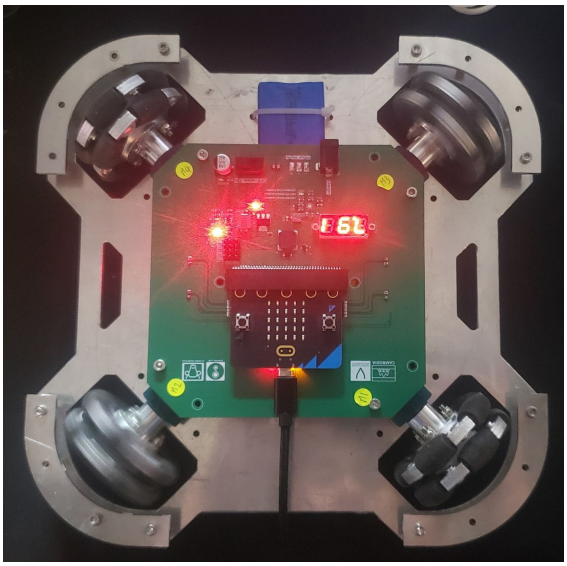
- Control the motor speed with condition by picking numbers (Counter) from 0 to 10. If the counter is lower than 5 its going to start all four motors and if it`s higher than 5 it`s stop all the motor.

- **Radio** control with RoboCar
  Controlling 2 microbit which is working as transmitter and another one is receiver.



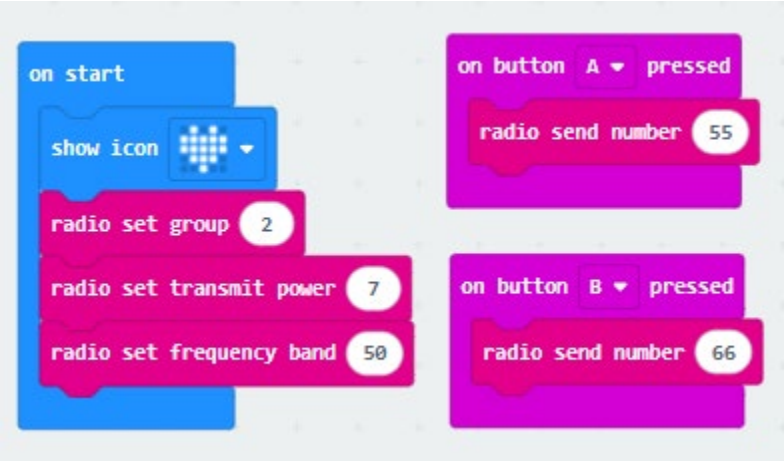
Transmit

Radio Control



Receiver

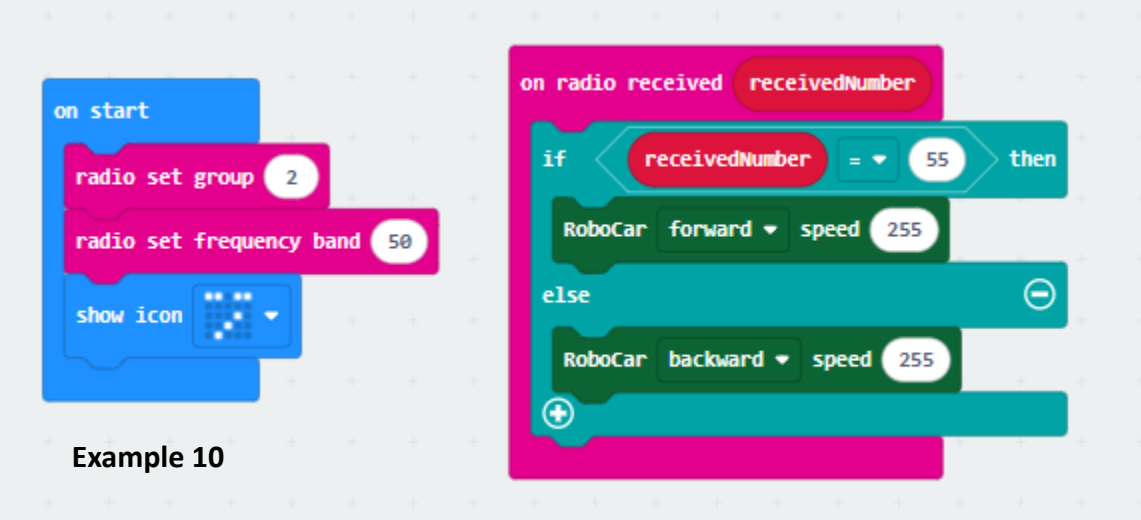
Code: download the code below to transmit



Example 9

- Set the group to 2, with transmit power 7, and frequency 50. The button A and B pressed set the radio send number 55 and 66.

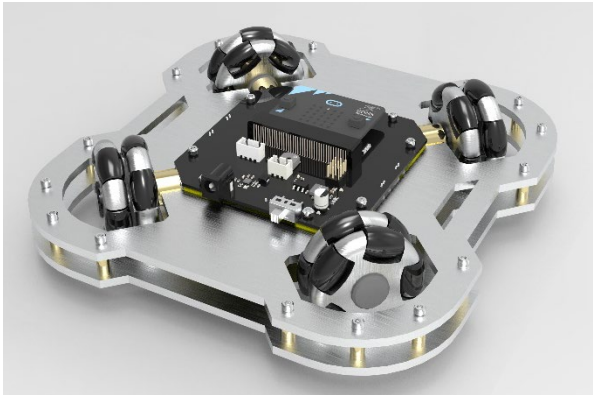
Receiver code:



Example 10

- Set group and frequency the same as transmit, use on radio received (received number) because if we used radio send string the received radio should be Received string too. But at this point we used radio send numbers (55 and 66) So, the we used "On radio received number".

+ Project 2: RoboCar with yahboom remote control

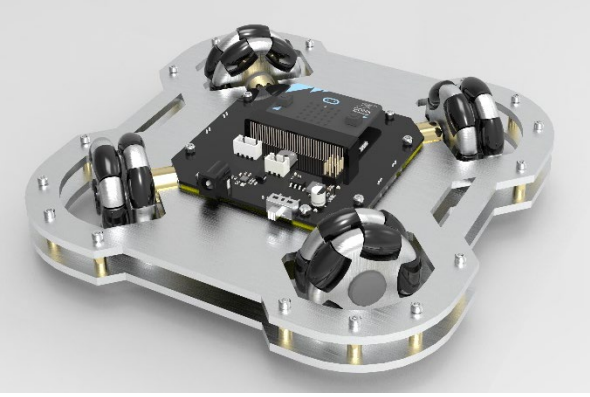


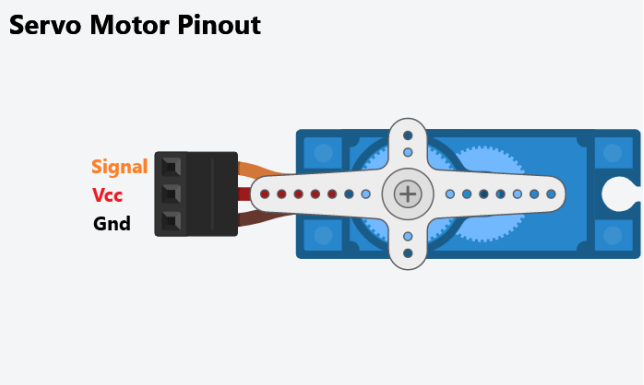
- This was aim to control the RoboCar with Joystcik by radio function. The joystick control the movements of RoboCar such as Forward, backward, leftside, rightside, BWright, BWleft, FW right, and FWleft. The joystick also control the speed of RoboCar by mapping from joystick value.

- The Transmit code (TX): https://makecode.microbit.org/_Y5Fd6ydWFDhX
- The Receive code (RX): https://makecode.microbit.org/_9Yk3Mk5YoEVu

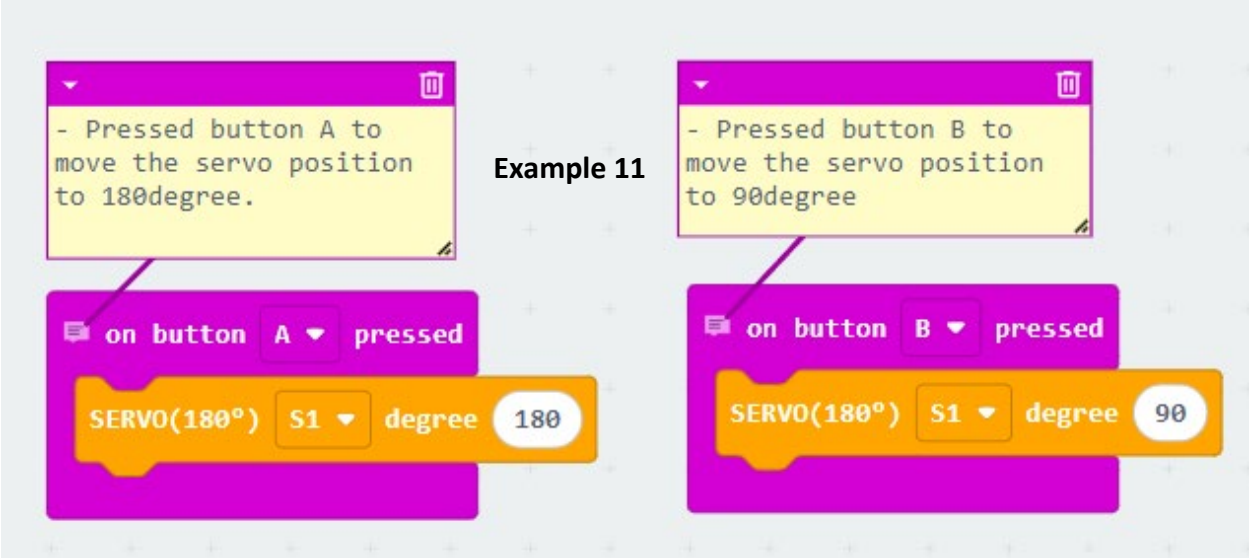
TX RX

- **Servo Motor** (we have built-in the driver servo motor which allowed us to use 5 servo motors on **RoboCar**).



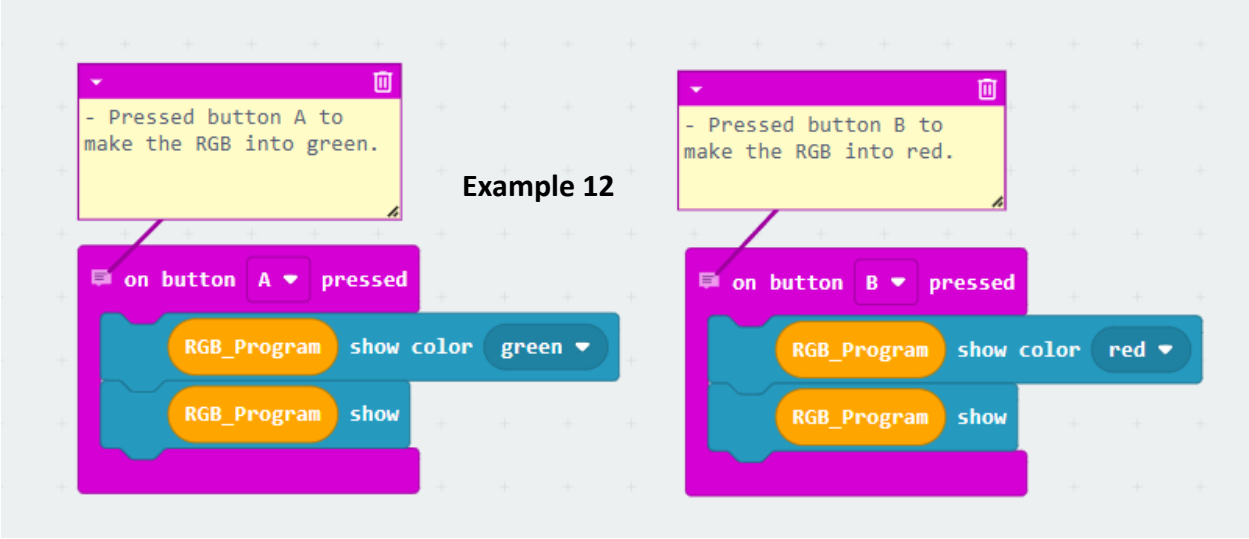


- On the RoboCar`PCB hase provided 5 pins of servo motors, connected the Signal, VCC, and GND follow the instructions
- Connect:
  + Signal to S_Pins on RoboCar
  + Vcc to (+5V)
  + GND to GND
- By following this code below, you will:



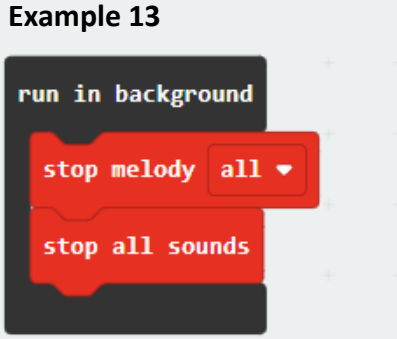
- By following this code, u can move the servo between 0 – 180 degrees with 5 servo motor at the same time. Furthermore, we also have an extension which is supported by servo motor (270degree). This extension can be used with a servo motor that is 270 degrees.

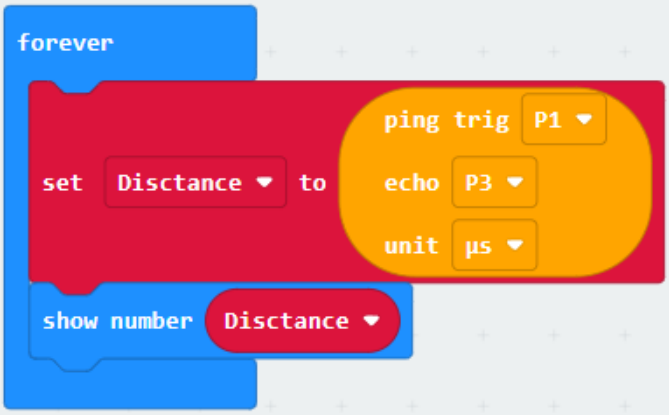
- **RGB Control:** On Robocar we also built the RGB on PCB. So, to control the RGB on Robocar let`s start with the code below:
  The code is controlled by button A and B



- **Ultrasonic control:** We have built ultrasonic pins for using ultrasonic. Anyway, the ultrasonic has the same pin with the P0 of speaker on Micro:bit. To avoid conflict the code should be:
  Before using ultrasonic we have to make sure that is not to the same pins as speaker or anything else to avoid conflicts.

Example 13





- Run in background in **example 13** is to immediately stop all music and sound effects without blocking the main program execution.

References

- **Find the Doc here:** https://github.com/MENG-Vathanak/ROBOCAR_CONTROL.git
- **Toshiba. (2023). *TB6612FNG Motor Driver Datasheet*. Retrieved from** https://www.toshiba.com
- **BBC. (2024). *Micro:bit Technical Reference*. Retrieved from** https://microbit.org
- **Microsoft MakeCode. (2024). *Micro:bit MakeCode Guide*. Retrieved from** https://makecode.microbit.org