REX

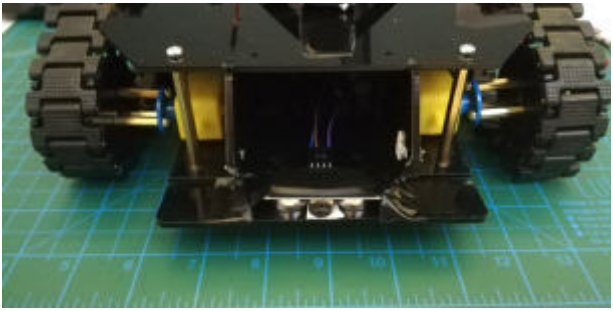
REX EVOLUTION SERIES
SUPER STAR TRANSFORMERS
8 IN 1

Destroyer

Non-falling Destroyer Application

Author: **Mustafa Kemal AVCI**

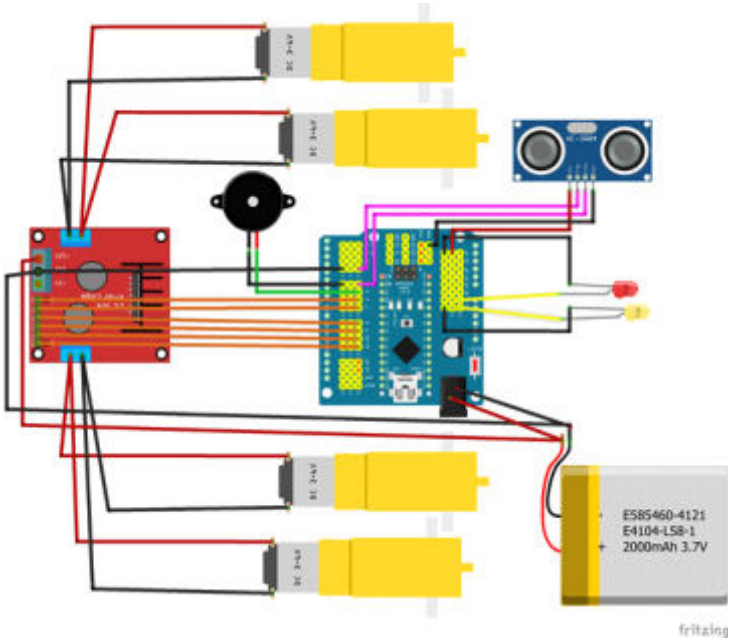
After the installation of our destroyer robot, we will make a small change in the design in order to make our application. Adhere the ultrasonic distance sensor face down as in the image below, using black electrical tape or a different tape.



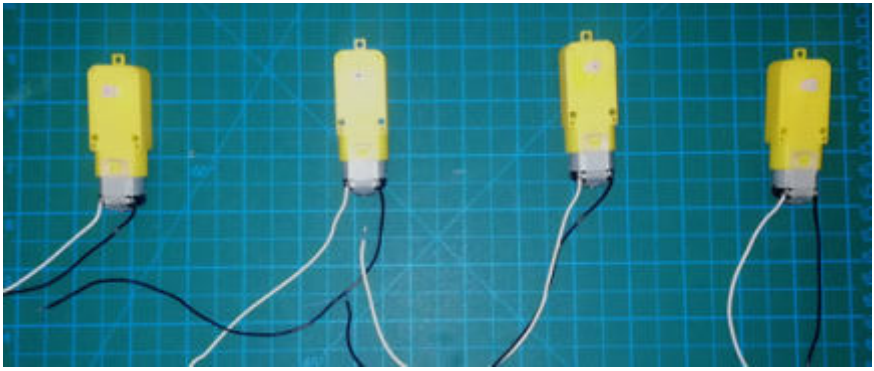
The reason the distance sensor stays that way is because it can detect the space in front of it. If it reads too much distance in front of it, it will mean a large gap. If it reads 0 cm or less, it will mean a small gap.

We will code the Buzzer and Leds so that the Robot can express the size of the sensed space and show the response stages. Thus, we will have information about the values coming from the distance sensor of the robot.

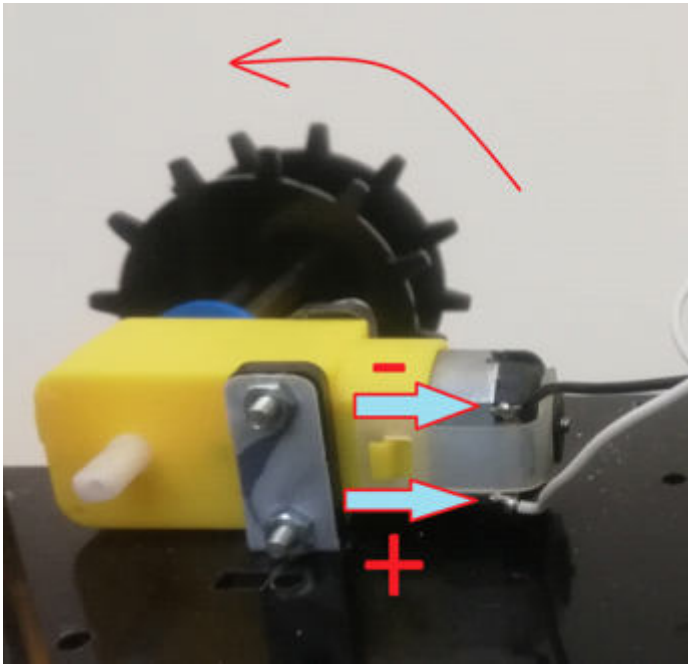
Connection Diagram



Connect Right Motor (+) poles to Out 1 (-) poles to Out 2, left motor (+) poles to Out 3 and (-) poles to Out 4. We will control the Out1, out2, out3 and out4 outputs on the motor driver (L298N) with the IN1, IN2, IN3 and IN4 pins, respectively. For right motor direction control, connect IN1 and IN2 pins to digital pins D7 and D8 on the sensor shield, respectively. Connect IN3 and IN4 pins to D9 and D10 digital pins for left motor direction control. Connect ENA pin to D6 pin for right motor speed control and ENB pin to D11 pin for left motor speed control.



You should solder the motors with 25-30 cm cable as in the photo. We will use the white leads as (+) and the black leads as (-) poles. In this case, when we connect the (+) black cable (-) pole to the white wires of the motors, the pallets rotate clockwise.



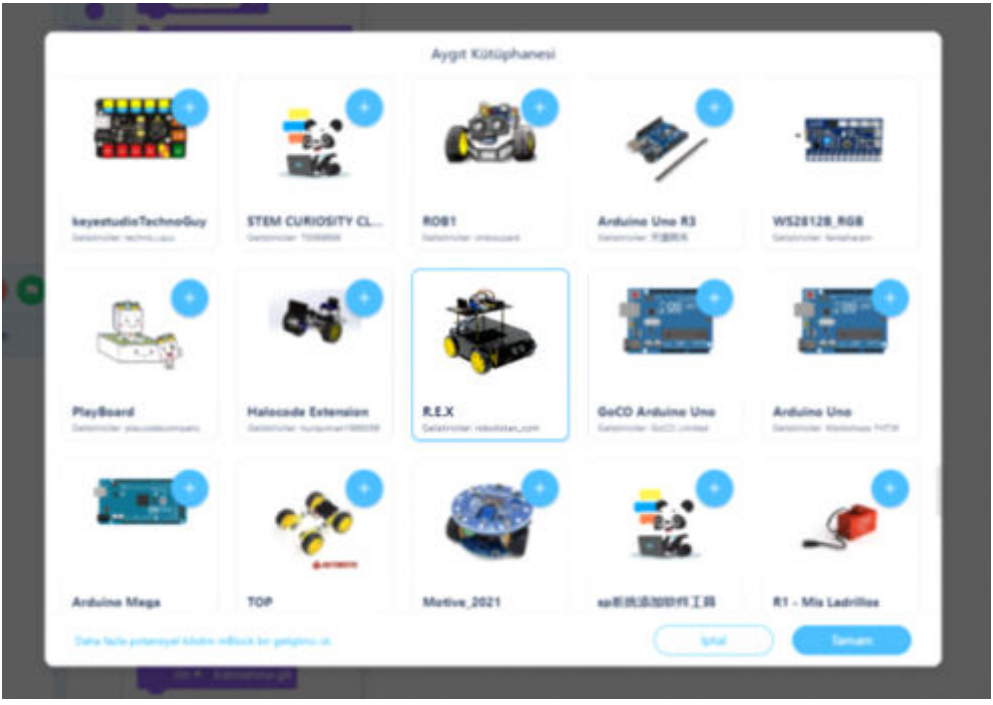
Connect the long leg of the buzzer that we will use for the audible warning to the D5 pin and the other leg to any GND pin designated as G on the sensor shield.

Let's connect the long legs of the leds that we will use for light notification to the A0 and A1 pins, respectively, and the short legs to any of the GND lines.

Connect the echo pin of our HC SR-04 sensor, which we will use to measure the distance, to the D4 pin, the trig pin to the D3 pin, the VCC pin to one of the 5V outputs on the shield, and the GND pin to one of the G pins on the shield.

Finally, you must connect power to the system. You should multiply the (+) and (-) ends of your battery power in the 7-12V range. As you can see in the diagram, one of the (+) and (-) terminals must be connected to the motor driver module and one to the Sensor Shield. After making the battery connection, you should make the connection to the common GND pin by pulling one more cable from any GND pin on the sensor shield to the GND terminal on the motor driver.

Coding the Robot
Let's start the mBlock 5 software and add R.E.X from the device library and move on to the coding phase.

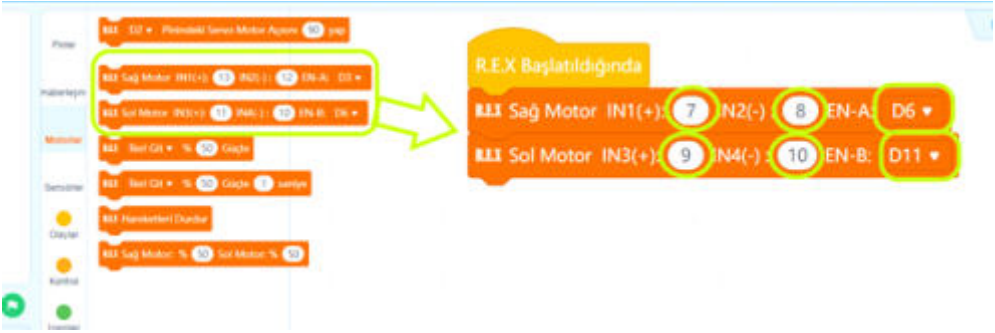


Our algorithm will be as follows;

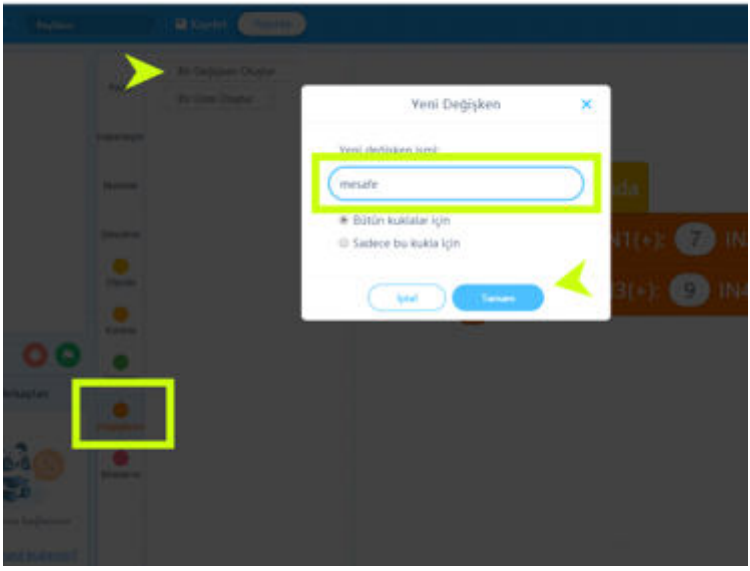
1. Start
2. Identify motors
3. Define variable for Distance Controller
4. Read distance
5. Advance if distance is less than 10 cm
6. Stop and react if the distance is less than 10 cm
7. Go to step 4
8. Stop

Drag and drop the "When REX Starts" block from the Events category into the coding area.

Drag and drop the blocks that we have defined the Right and Left Motor control pins in the "Motors" category, under the "When REX Starts" block, respectively, as in the image. Write the pin definitions in the circuit diagram to the necessary places as in the image.



We will create a distance variable to store the distance value that the ultrasonic distance sensor will read. For this, we enter the "Variables" category and click the "Create a Variable" button. In the window that opens, we write the variable name as distance and press the OK button. Thus, we have defined our variable and set its initial value to 0.

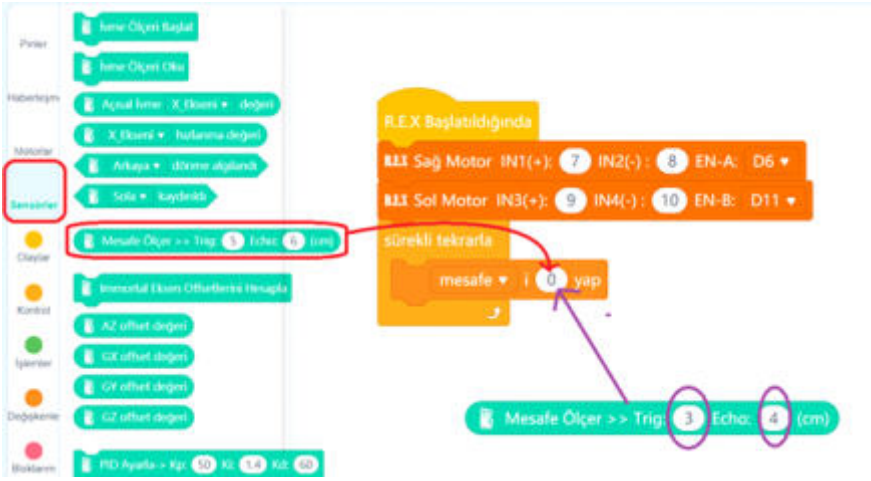


When we write the variable name as "distance" and press the OK button, the "Variables" category blocks should appear as follows.

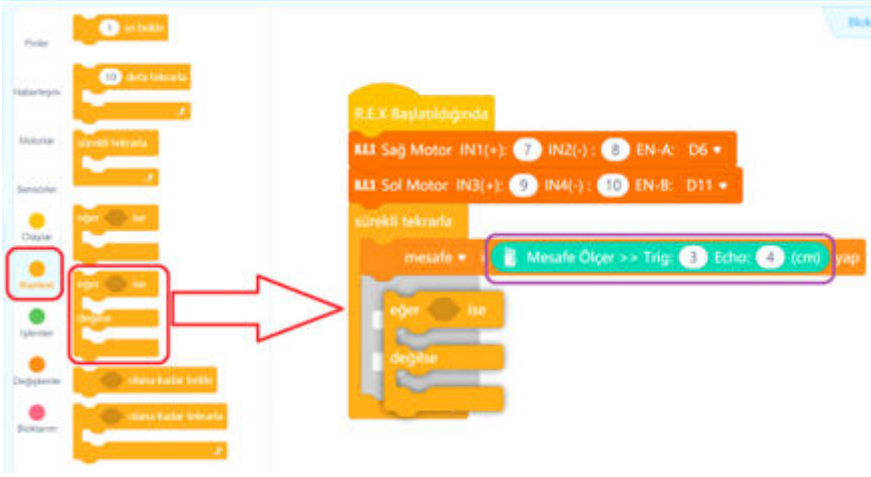
Since the distance will be read continuously and the direction and speed will be given to the motors according to the distance, we place the "Repeat continuously" cycle block from the category of "Control" blocks, just below the motor identification blocks.

We will assign the value of the distance variable to the value read from the sensor continuously. Whenever we need the distance value in the program flow, we will use that variable. Let's place the "set distance to 0" block in the "variables" category inside the repeat constantly block as in the image.

The distance measuring sensor (HC-SR04) gives us the accuracy of one hundredth of a cm. For example, 36.42 cm. Let's move the distance meter block from the "Sensors" category to our work area as in the image below. As we have determined in our circuit diagram, let's change the pin number that we connect the Trig pin to 3, and the pin number to which we connect the Echo pin to 4.

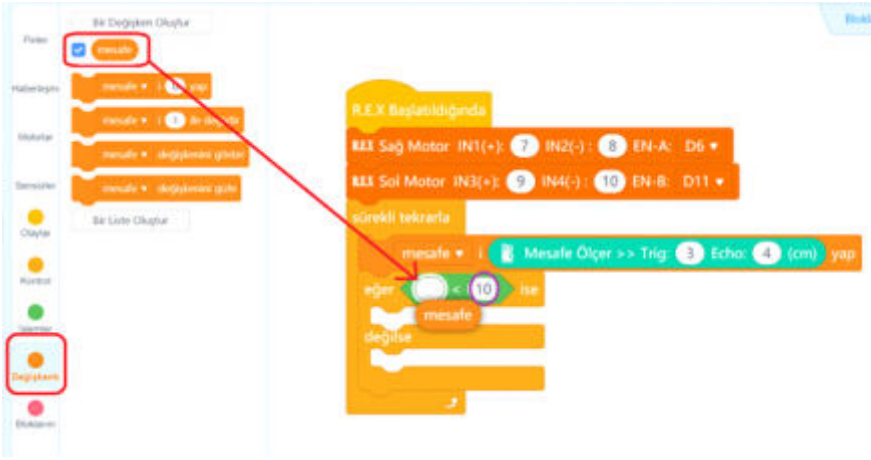


After changing the Trig and Echo pin numbers, we place it in the block where we assign a value to the distance variable as in the image below. Now we can prepare the blocks on which we will perform the mathematical test. If the value of the distance variable is greater than 10.00 cm, we want the motors to stop and react, if not, to exhibit forward behavior. In other words, if the condition we are testing is true, a command will be executed, and if it is false, a different command will be executed, drag and drop the "If If Otherwise" block under the distance variable block in the "Control" blocks category.



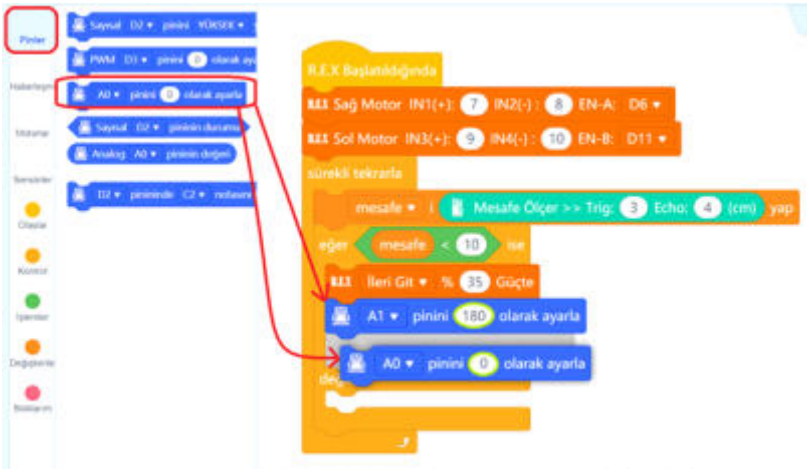
Our condition will be if the distance is less than 10. To express the condition, let's take the "<" less than operator from the operations category and drag and drop it into the condition field.

Since we will check whether the value of our distance variable is less than 10, let's change the number 50 in the operator block to 10 and drag and drop the distance variable value block into the space to the left of the green operator block.

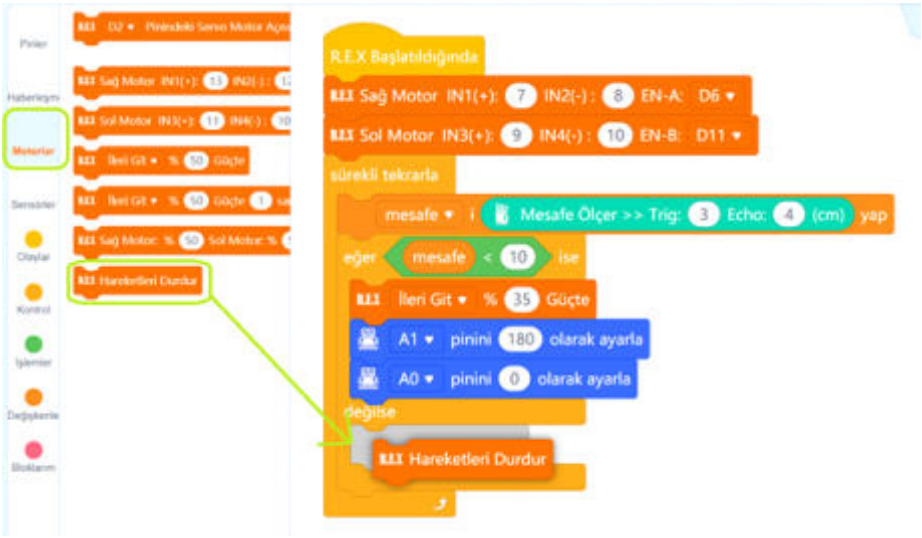


Since our distance sensor faces the ground, the value it reads will mostly be between 0-3 cm. Due to its technology, the sensor can read distances less than 4 cm as 0. This means that there is no space in front of you, you can move forward. Therefore, as long as this condition is met, the robot must move forward. For this, we drag the forward block in the Engines category and leave it to the command field to be executed if the condition is true. Let's set the engine speed to 35%. If we do it faster, our robot may fall off the table until the distance is checked and decided.

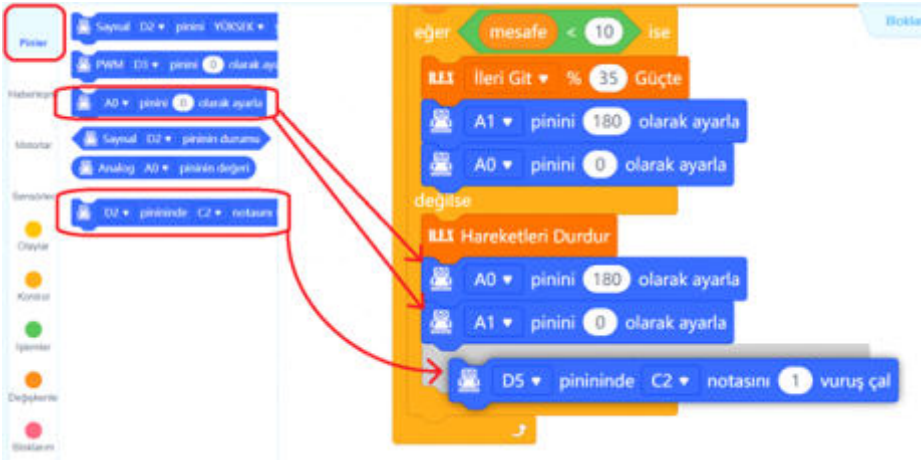
While our robot continues its forward movement, let one of its leds turn on and the other turn off. As in the image below, with the "Set A0 pin to 0" block in the Pins category just below the go forward block, we light our A1 led and turn our A0 pin off. With this block, where you can set a value between 0 and 255, our LED's maximum brightness will already reach between 120-180. Therefore, we use the value of 180 to turn the LED on at full brightness and the value of 0 to turn the LED off.



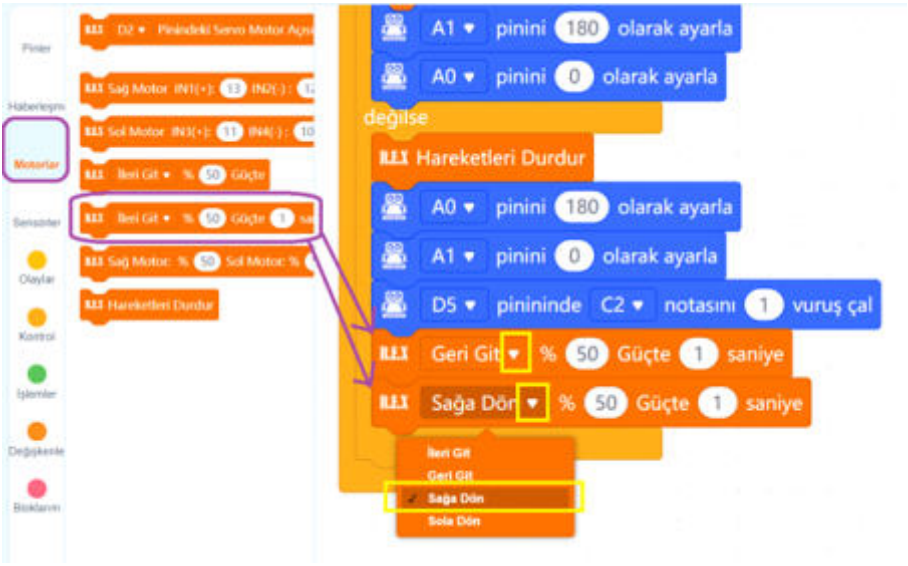
If the condition is not met, which commands will our robot run? For this, we will stop the motors and make them react as we mentioned in step 6 of the algorithm. Let's drag and drop the Stop motions block in the Motors category to the commands to be executed when the condition to stop the motors is not met.



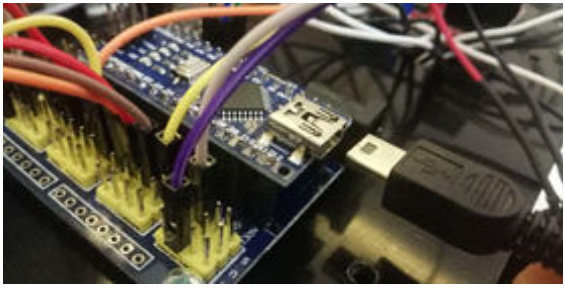
Now it's time to make our robot react. For this, we will turn on the led on the A0 pin and turn off the led on the A1 pin. We will also play a 1-second note from the buzzer we connect to the D5 pin. For this, let's drag and drop the blocks shown in the image below in the Pins category, just below the stop motion block.



As the final reaction of our robot, we will give the command to go backwards for 1 second and turn right for 1 second. In this way, we will move away from the pit and make it turn in a different direction. We are making the changes in the image below by taking two of the "go forward 1 second at 50% power" block in the Engines category.



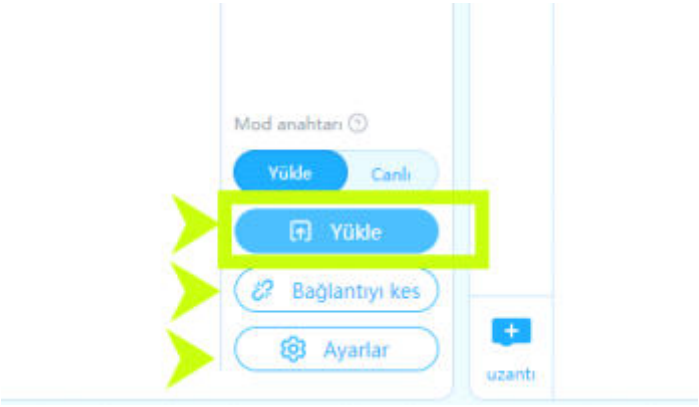
The final version of our code should be as follows.



We can now upload the codes we have prepared to our robot. We connect one end of the USB cable to Arduino nano or the other end to our computer.

Click on the connect button in the installation mode and tick the show all connectable devices option. The mBlock software will automatically insert the COM port number to which our CH340 chip card is connected. If connection is not possible, you can select other COM port numbers from the drop-down list.

When the connection is established, the “disconnect”, “settings” and “Install” buttons will be active. Let’s upload the code to our robot by clicking the upload button.



Execution of Codes
As soon as the code is loaded, our robot will try to move by reading the data from the sensor. After loading the codes, you can make the robot run the codes by removing the usb cable and connecting the cables of the battery providing power to the sensor shield and the motor driver, or by turning on the switch, if any. You can achieve the precision you want by adjusting the rotation times of your robot according to the battery power, the forward speed and the rotation speed.

Possible Problems and Solutions
If your robot is only going back and forth, not turning left or right or vice versa,
Make sure to check the locations of IN1,IN2,IN3 and IN4 pins on the shield and on the motor driver. They may be confused.

If your robot is going backwards in the Forward code, and turning left in the turn right code;
Make sure that OUT1 and OUT 3 on the motor driver are connected to the (+) pole of the motors, and OUT 2 and OUT 4 are connected to the (-) pole of the motors. Also, make sure that the motors on the right of the robot are connected to OUT1 and OUT2 on the driver, and the motors on the left are connected to OUT3 and OUT4.

If your robot encounters an obstacle, the backlight does not turn on, and it always continues to go forward;
Make sure that the jumper cables connected to the Trig and Echo pins of the HC-SR04 (distance sensor) module are more stable. With regard to the stance angle of the sensor, a very small movement in the jumper cables on these two pins causes the sensor to read 0 (si) value, which in turn causes the obstacle detection system to fail.

If your robot detects the pit but stops late and falls;
Decrease the speed in the forward command. At 30% the robot will move quite slowly. Lowering it further may prevent your robot from moving. Because sending low power to the motors may not be enough for their rotation according to the load they carry.

YouTube

youtube.com/robotistan

FORUM

robotistan

forum.robotistan.com

BLOG

robotistan

maker.robotistan.com

Robotistan Electronics Company

Mustafa Kemal AVCI (Content) - Fadıl PALA - Mehmet AKÇALI (Editor) - (Mehmet Nasır KARAER (Graphics) - Mert ALTUNTAS (Translation)
info@robotistan.com - www.robotistan.com
Phone: 0850 766 0 425