REX

REX EVOLUTION SERIES
SUPER STAR TRANSFORMERS
8 IN 1

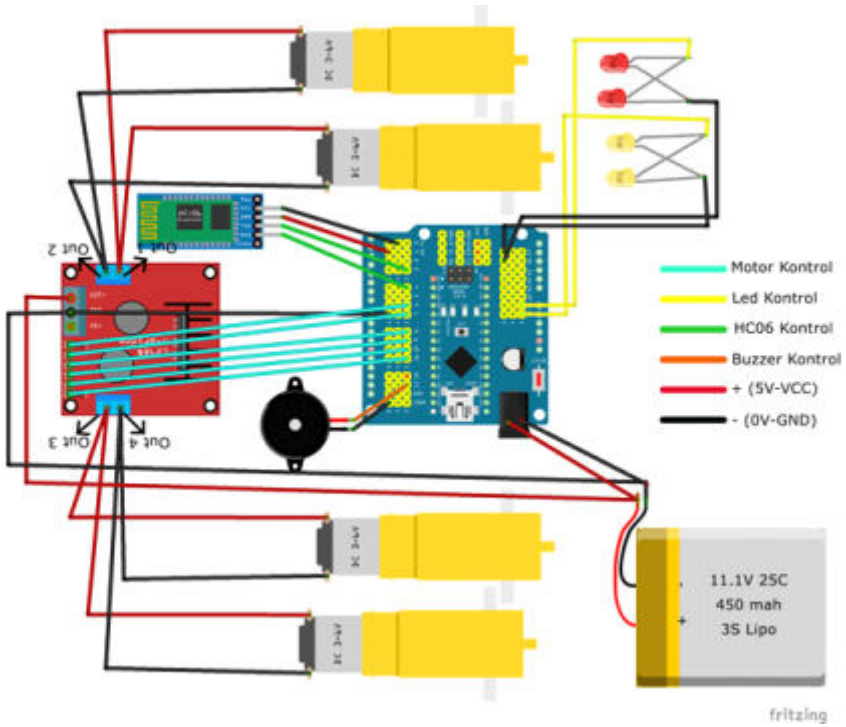
BTBot

Bluetooth Controlled Survivor Robot

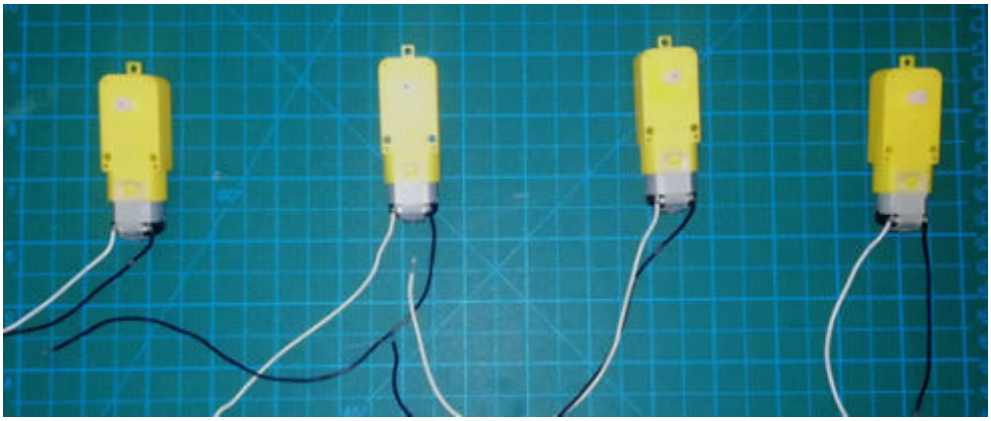
Author:**Mustafa Kemal AVCI**

Connection Diagram

After completing the installation of our Survivor robot and the assembly of its electronic components, we must make the cable connections according to the circuit diagram below.



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 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.

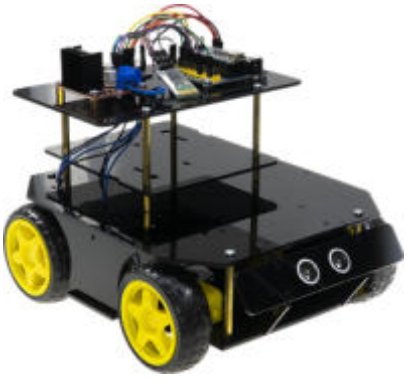


You should solder the motors with 25-30 cm cable as in the photo. We will use the white leads (+) and the black leads (-) as poles. In this case, when we connect the (+) black cable (-) pole to the white wires of the motors, the wheels rotate clockwise.

Connect the long leg of the buzzer that we will use for the horn (horn) to the D13 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 front and rear lights to A0 and A1 pins, respectively, and the short legs to any of the GND lines.

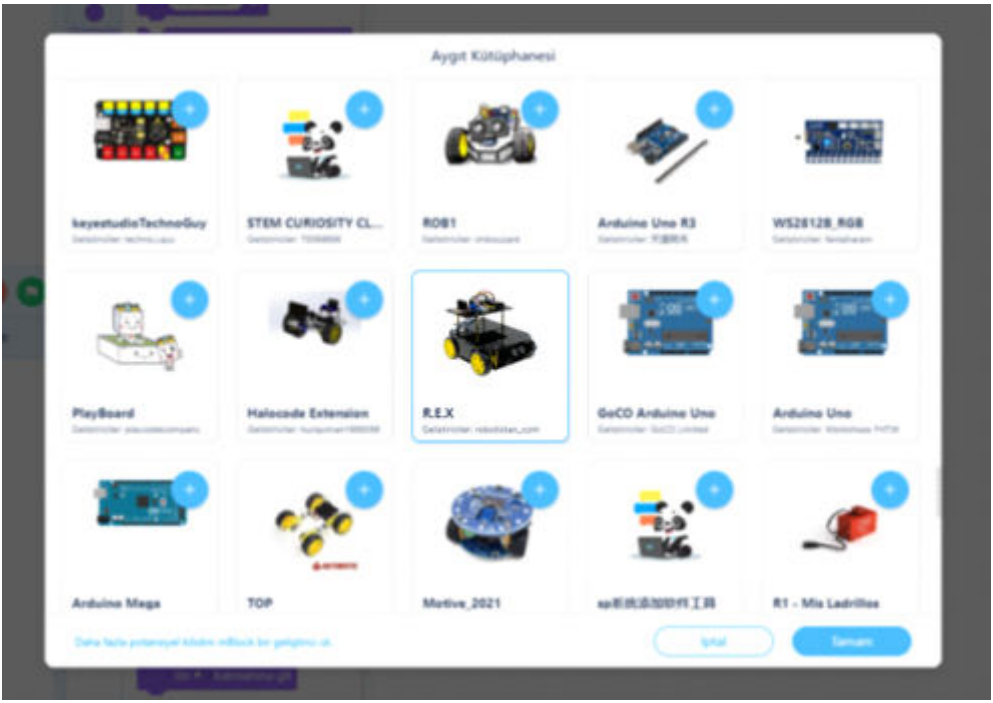
Our hc06 bluetooth module will communicate with the mobile "Arduino Bluetooth RC Car" application. Let's connect the RX pin of the Bluetooth module to the D4 pin and the TX pin to the D3 pin. Let's connect the VCC pin to the 5V output on the shield and the GND pin to any G pin on the shield.



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

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

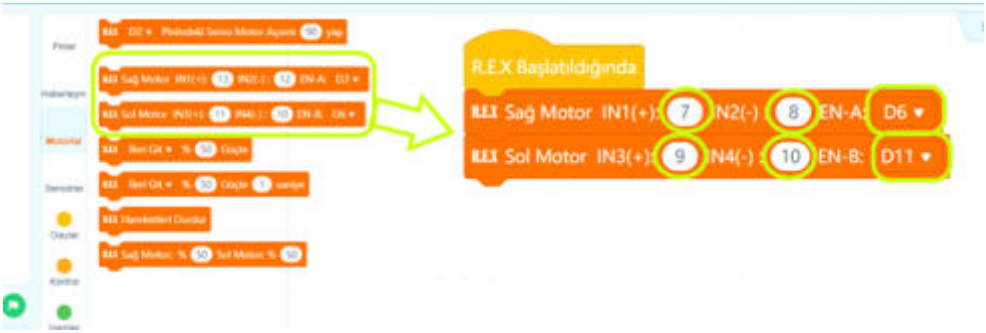


Our algorithm will be as follows;

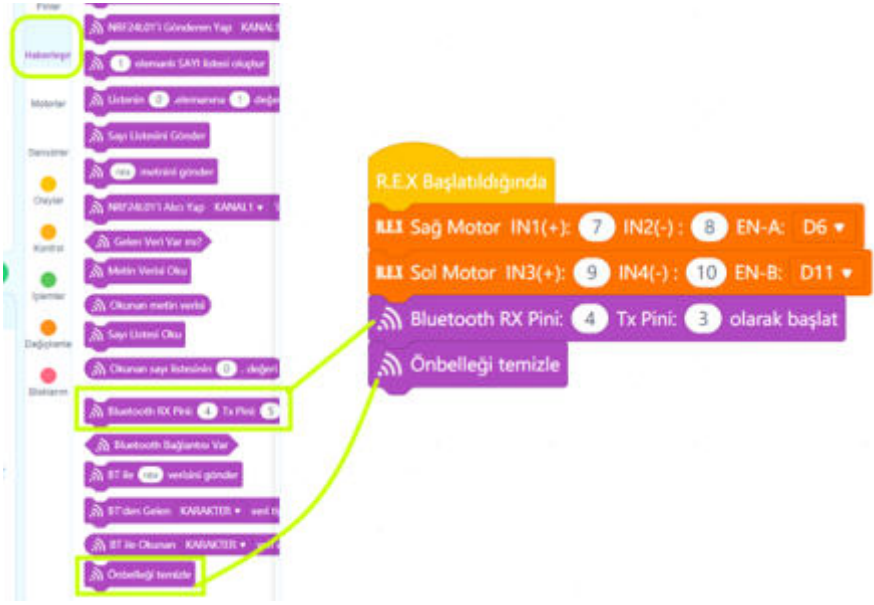
1. Start
2. Identify engines
3. Identify Bluetooth
4. Define variable for Speed Control
5. Read with if there is a Bluetooth connection
6. React by with
7. Stop

Drag and drop the "When REX Starts" block from the event blocks into the coding area.

Drag and drop it under the "When REX Starts" block as in the image. Write the pin definitions in the circuit diagram to the necessary places as in the image.



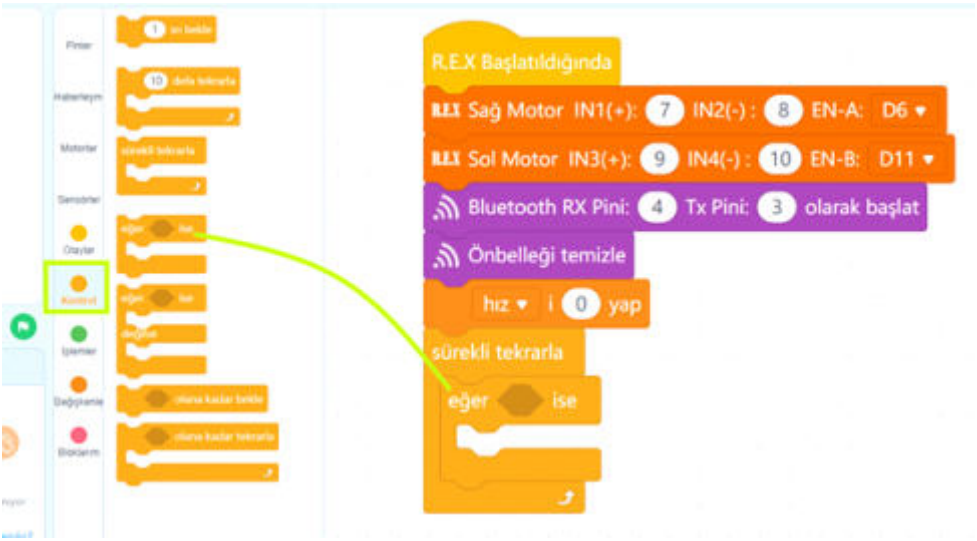
We find the Bluetooth identification block in the "Communication" category as in the image and place it under the engine identification blocks. Then we change the pin numbers to be the same as the numbers on the circuit diagram, as in the image. With the "clear cache" block, we remove possible cache problems in our connection.



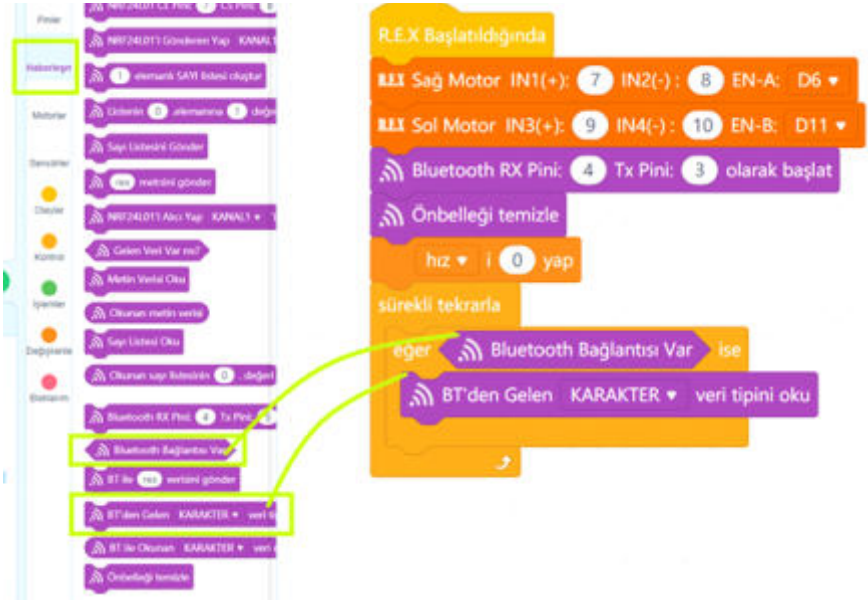
In order to create the speed variable that we will use to adjust the motor speeds according to the speed adjustment character that will come from the Arduino Bluetooth RC Car mobile application, we create our variable by entering the "Variables" category and following the path in the image below.

Now, it's time to constantly check the bluetooth connection, compare it with certain characters when there is incoming data, and make our robot give the reactions we want. We will do these operations in the "Control" category in the continuous repeat block.

The Bluetooth connection is always on and as soon as the connection is established, the Arduino Bluetooth RC Car application starts sending characters (S) via bluetooth. Since we want to do all the operations when the connection is established, we move the "If" block from the Control blocks category into the "Repeat continuously" block that we placed in the block field before.



The condition of our "if" condition block is the status of the bluetooth connection. If there is a connection, it is necessary to read the incoming character message. Because Arduino sends forwards as characters with the Bluetooth RC Car application. Let's place the status block and the character reading block from the communication blocks in the "If" block, as in the image below, for connection control and character reading.



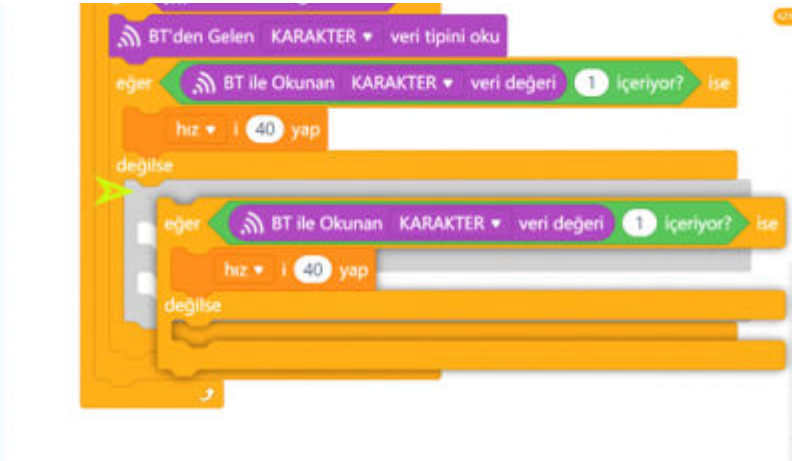
As soon as we read good with the one coming from Bluetooth, we will start the comparisons. While making these comparisons, we will proceed by making them inside the "If not" blocks. When Arduino Bluetooth RC Car application sends numbers 1-9, we will use them as speed settings. We will assign the lowest speed to 40% power and the highest speed to 100% power to our speed variable. Let's take the "If not" block from the Control category and place it just below the "Read CHARACTER data pi from IT" block.

Then let's drag and drop the "does apple contains a" status block from the operations category into the condition field of the "if not" block.

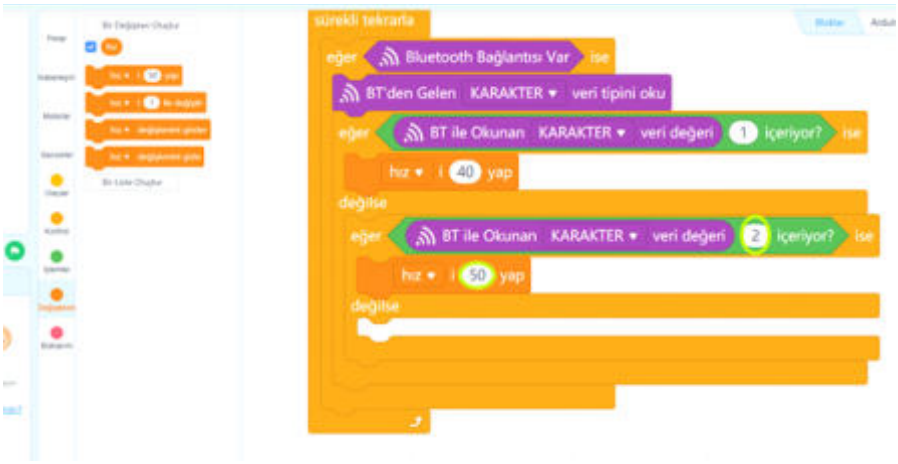
The "character data value" whose content will be controlled is written in the "apple" field of the green status block that we place in the condition field, and the possible and content in the "a" field. If the incoming data value contains 1, we will set its variable to 40.

We change the speed variable from the variables category to “set the speed to 40” and drag and drop it to the command field that will be processed if the condition is true, as follows.

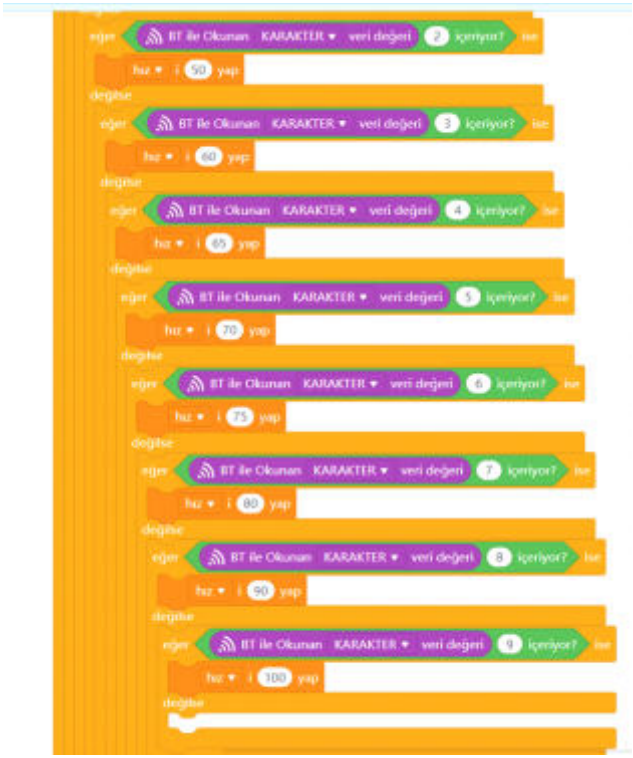
If our condition is not true, we will compare the contents of the ”character data value” to 2. Pra k to do this, as soon as we right-click on the “If not” block and select the “Uncopy” option, we insert the copied block group into the otherwise section.



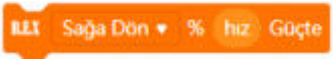
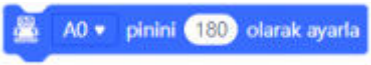
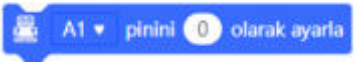
Then, we update the content of the with in the second comparison to 2 and code the speed to be set to 50 if the condition is true.



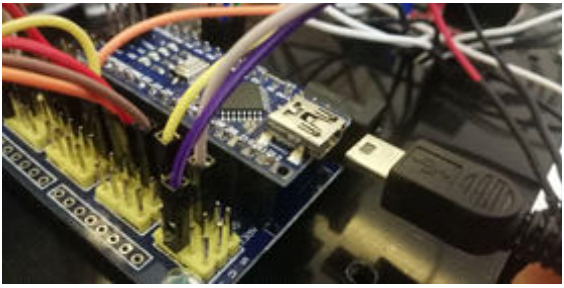
In this way, we make all comparisons up to 9 by duplicating the last "if if not" blocks and inserting them into the "if not" part, as follows. We determine the speed variable as 6 if 3, 65 if 4, 70 if 5, 75 if 6, 80 if 7, 90 if 8 and 100 if 9.



With this technique, we set the code to work for each character that will come from the Arduino Bluetooth RC Car application. In the table below, other character data values, their meanings and blocks to be worked on are given. In this way, complete your coding code.

Variable comes from BT	Meaning	Block to Work	Block Category
F	Forward		Motors, Variables
B	Backwars		Motors, Variables
L	Turn left		Motors, Variables
R	Turn right		Motors, Variables
G	go left		Motors, Variables
I	go right		Motors, Variables
H	go back left		Motors, Variables
J	go back right		Motors, Variables
S	stop		Motors
W	enable front headlight		Pins
w	disable front headlight		Pins
U	enable rear headlight		Pins
u	disable rear headlight		Pins
V	horn		Pins

A0 and A1 pins are analog pins, they can also send voltage to the pins in the 0-255 value range, such as PWM. Since most LEDs reach maximum brightness around 3V, we have not shortened the life of the LEDs by providing a voltage of around 3V with a value of 180, rather than giving the A0 and A1 pins a value of 255, which means 5V.

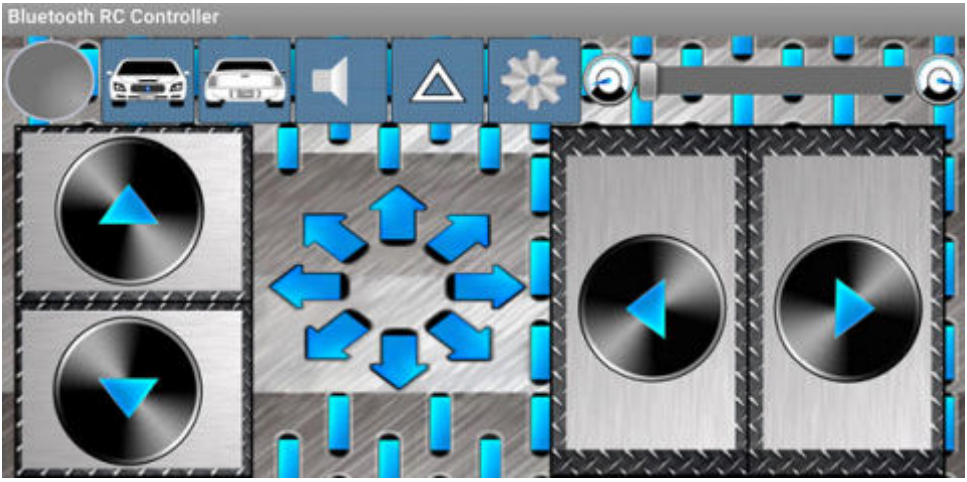


We can now upload the codes we have prepared to our robot. We connect one end of the USB cable to Arduino nano and the other end to our computer.

Click on the now 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.

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

Execution of Codes
To control our robot, we download the "Arduino Bluetooth RC Car" application from our app store.

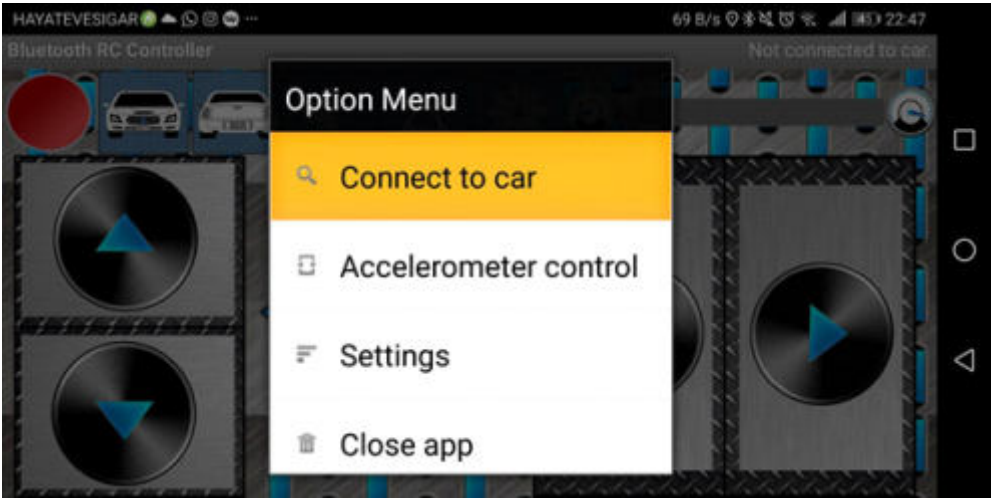


Every button you interact with in the screenshot is programmed to send a character to our robot via bluetooth. You can see what these characters are by clicking the gear symbol and clicking the Settings option from the menu that opens.

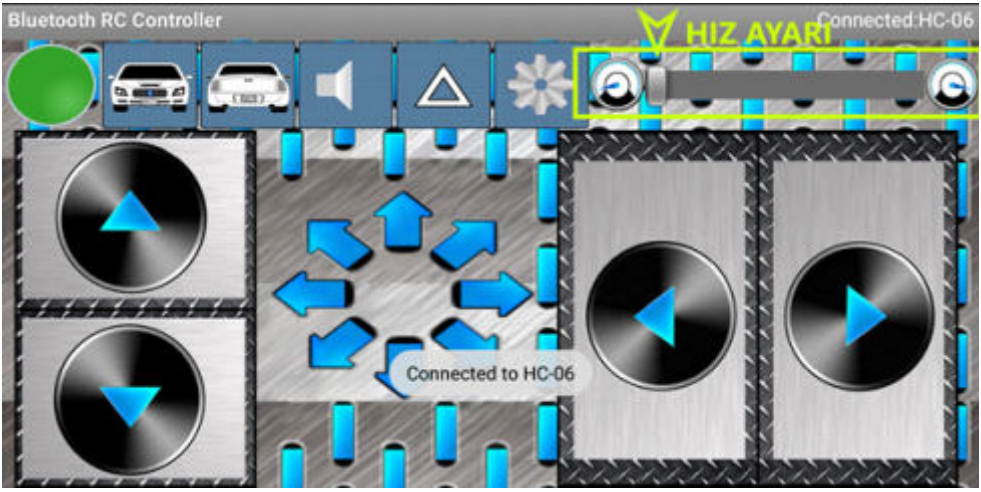
We are ready to control our robot with the Arduino Bluetooth RC Car application. First we have to check that our power connections are complete and error free. If the red leds on the Arduino Nano, sensor shield and motor driver are on, it means that our power connection is ready. The led of the HC06 bluetooth module is blinking.

We search for a device by entering the bluetooth settings of our mobile device. Let's wait for him to see our robot as HC-06. After seeing HC-06 on our device, when we click on it, we enter 1234 as the password and complete the pairing process.

We open the Arduino Bluetooth RC car application and click on the gear symbol. When there is no connection, the red circle will be flashing. Let's choose "Connect to Car" from the drop-down menu and select "HC-06" from the list of bluetooth devices that opens. Here it will ask us for permission to control Bluetooth devices. Let us continue.



3-5 seconds after selecting the HC-06, you can get the notification that the connection has been made and observe that the flashing circle turns green. In addition, the red led on our HC06 module will turn on continuously, so we can control our r.Artık robot. You can set our speed in the speed setting section and enjoy your robot.



Possible Problems and Solutions

If a Bluetooth connection cannot be established,
Uninstall and reinstall the app and don't forget to accept the request for permission to use bluetooth connection during the first connection.

If the bluetooth connection is lost during use, especially when you want to move the motors, Your battery power is weak or the battery you are using cannot feed the robot momentarily. Charge your battery.

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 turns back when you give the forward command and turns left when you give the right command;
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.

YouTube

youtube.com/robotistan

FORUM

robotistan

forum.robotistan.com

BLOG

robotistan

maker.robotistan.com

Robotistan Electronics Company

Mustafa Kemal AVCI (Conent) - Fadıl PALA - Mehmet AKÇALI (Editor) - (Mehmet Nasır KARAER (Graphics) - Mert ALTUNTAS (Translation)

info@robotistan.com - www.robotistan.com
Phone: 0850 766 0 425