REX

REX EVOLUTION SERIES
SUPER STAR TRANSFORMERS
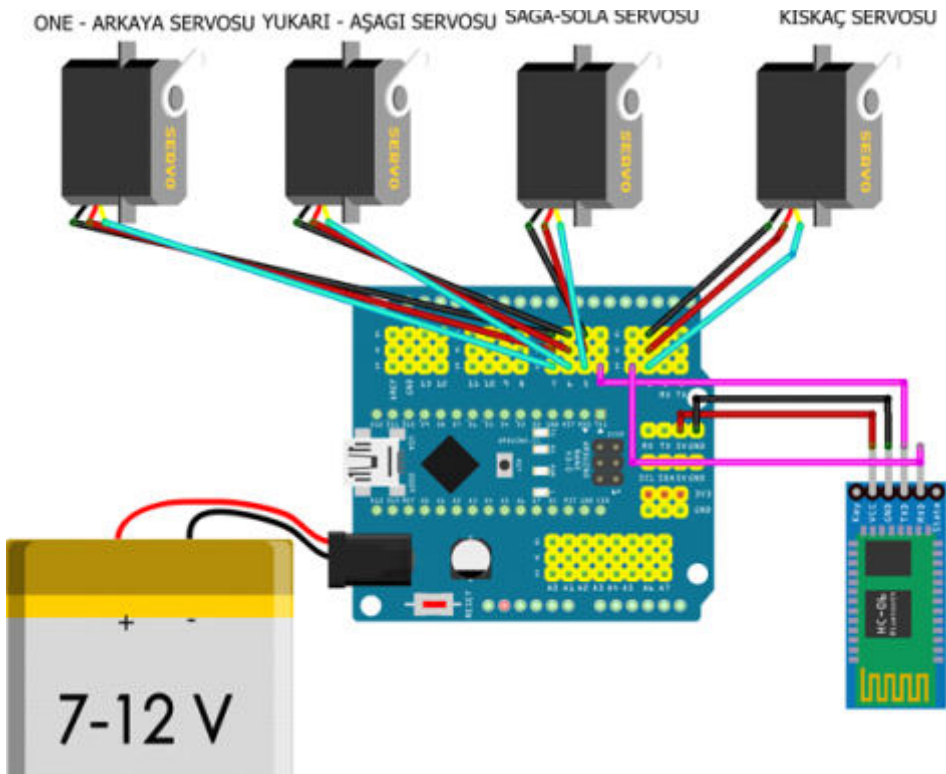**8 IN 1**

V-ArmBot

Voice Controlled Robot Arm Project

Author:**Mustafa Kemal AVCI**

After completing the installation of the armbot, we will control it with our conversation over bluetooth. First of all, we would like to remind you that the first indicator of a healthy operation of the robot arm is that when the robot arm is not connected to the battery or USB, you should be able to move securely and comfortably with your hand in all directions. The operation of the system may be disrupted because the servo motor will also be forced on an axis that you have difficulty with. Make sure that the calibration and assembly of the servo motors are in accordance with the manual. If you loosen the screws of the hard-rotating servos' heads, it will ease the rotation.

To control the robot arm via Bluetooth, install BT Voice Control For Arduino application on your Android device.

BT Voice Control for Ardu...
SimpleLabsIN
Aç

Connection Diagram

Let's connect the Sg 90 servo motors to the pins on the Arduino Nano shield, corresponding to the pins shown in the image below. Connect the servo motor to pin 2 to be used for holding and releasing, the servo to be used for right and left rotation to pin 5, the servo to be used for up and down movements to pin 6, and the servo to move forward and backward to pin 7.
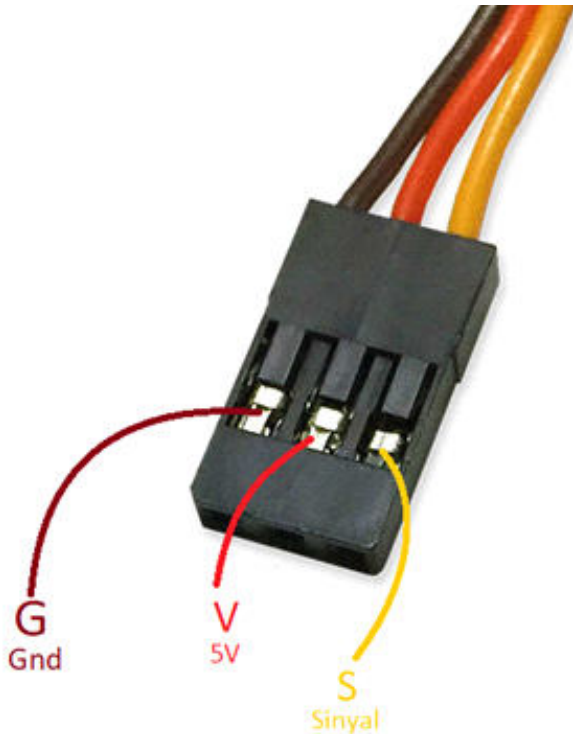
Connect the VCC pin of the HC06 bluetooth module to the 5V pin and the GND pin to the G pin. Connect the Tx pin to pin 4 and the Rx pin to pin 3.

Coding the Robot

After completing the installation of the robot, we must first run the servos one by one and check their starting positions. Then we will determine the widest and narrowest angles of each servo and make it act within this range. If we force the servo to an angle where it cannot rotate without determining the range, it will draw too much current and cause it to break down or cause other servo motors to malfunction.
Let's start the mBlock 5 software and add R.E.X from the device library and start tuning the servo motors.

We drag and drop the block when the rex starts from the events category.

We must have set it up so that the fully open state of the servo on the D2 pin is 0 angle. But sometimes we don't pay much attention to it during installation and there may be different angles. As in the image below, we should test the angles of each motion servo by changing our servos and their angles and uploading the code.

When we connect the Arduino nano to the computer and upload this code, let's observe the action of the servo. 75 degrees is the angle we will use to close the clamp. To open the clamp, it is enough to be at 15 degrees. Try this out to determine the most suitable angle range and make a note of it. We will use it later in the coding of the Armbot.

We must determine the ranges of all servo movements by trying the code, pin number and angle value in the image above.
The following table gives the angle ranges of the servos used in this setup. Your servos may differ from these ranges. Build this angle chart for your own robot by trying only one servo at a time.

Movement Name	Servo Pin No	ANGLE RANGE	starting angle
On - Off	D2	15-75	75
Down-Up	D6	0-40	40
Forward-Backward	D7	40-0	0
Right-Left	D5	15-85	45

Now as soon as Rex starts, we have to get all of our servos to their starting position. So, let's drag and drop the Servo block in the motors category and write the starting positions of the servos as in the image below.

Let's take the 1 second wait block from the control blocks and let the servos come to their starting position. Right after that, let's define our Bluetooth connection. For this, let's use the Bluetooth identification block in the communication blocks as in the image below.
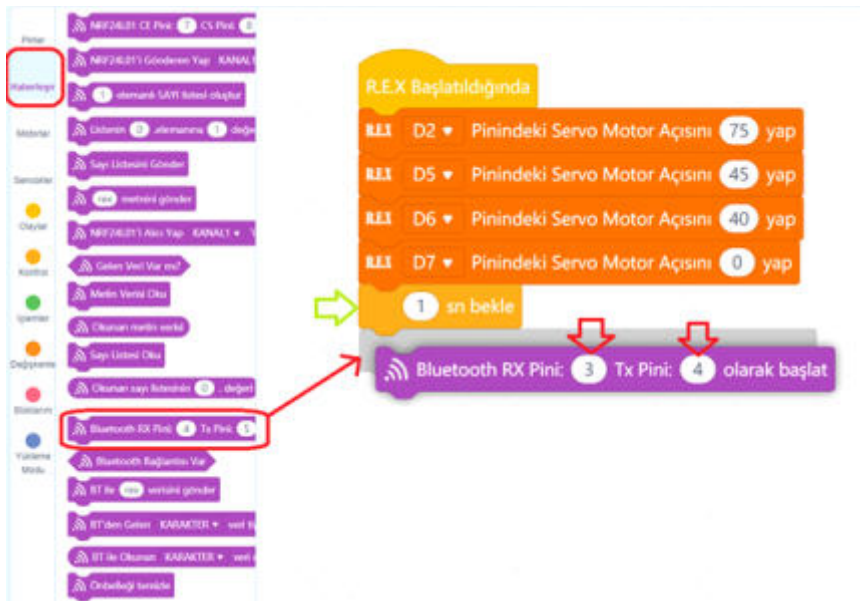

We have to constantly check if it comes with bluetooth. We will make the movements of the robot arm when it comes with Bluetooth. If there is no incoming, no action will be taken. We take the repeat block from the control blocks and use it as in the image below. Let's drag and drop the block, if any, right into it.
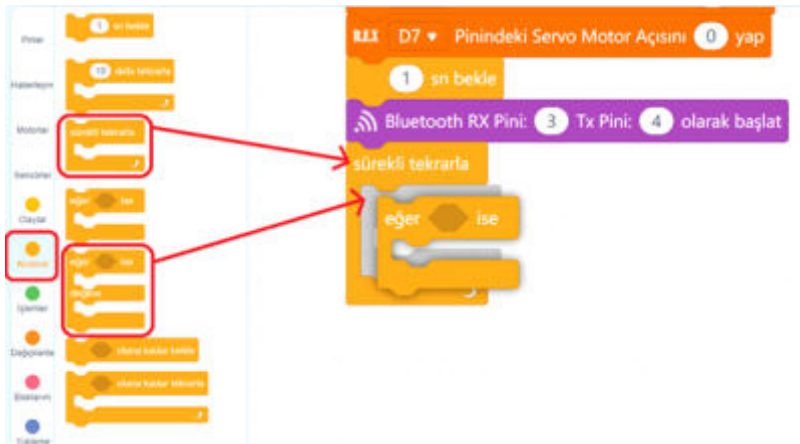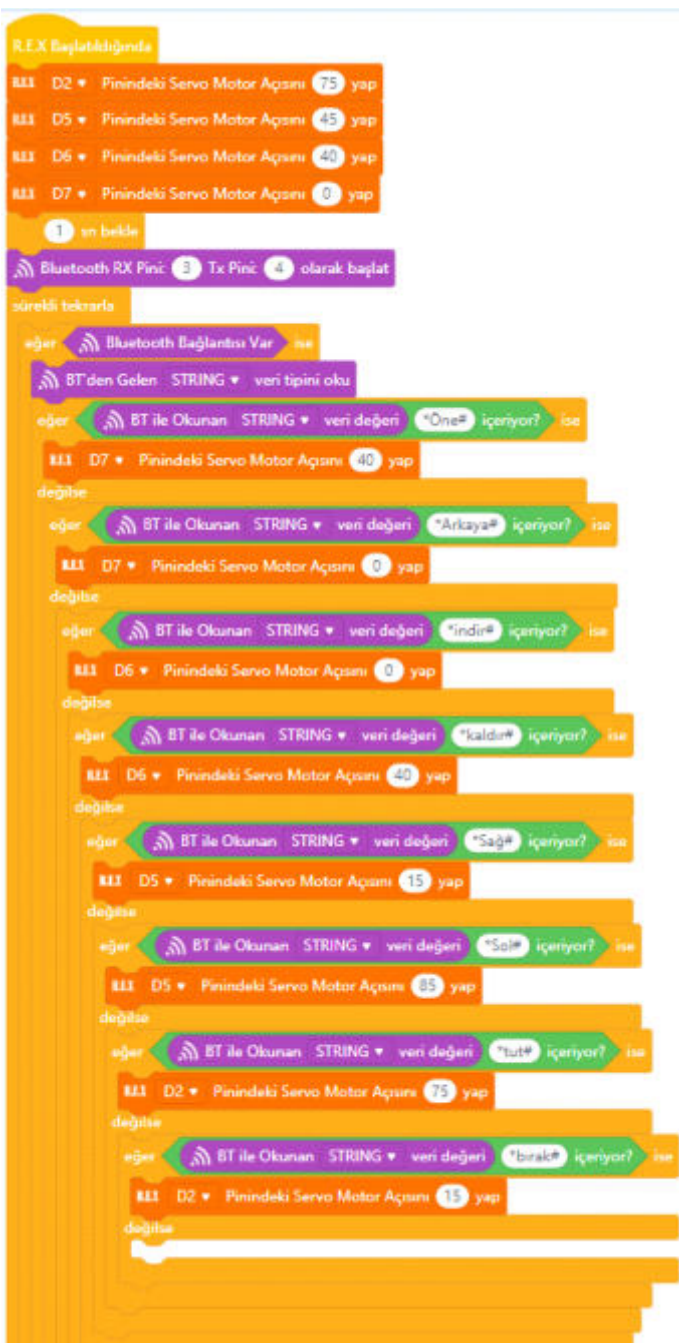
Our conditional expression will be: If there is a Bluetooth connection, the STRING (character string) data pin coming from Bluetooth is school. String data pi is a data pi with more than one character. We are making such an adjustment on the block, as word will come from Bluetooth. Let's use our bluetooth blocks as in the image below.
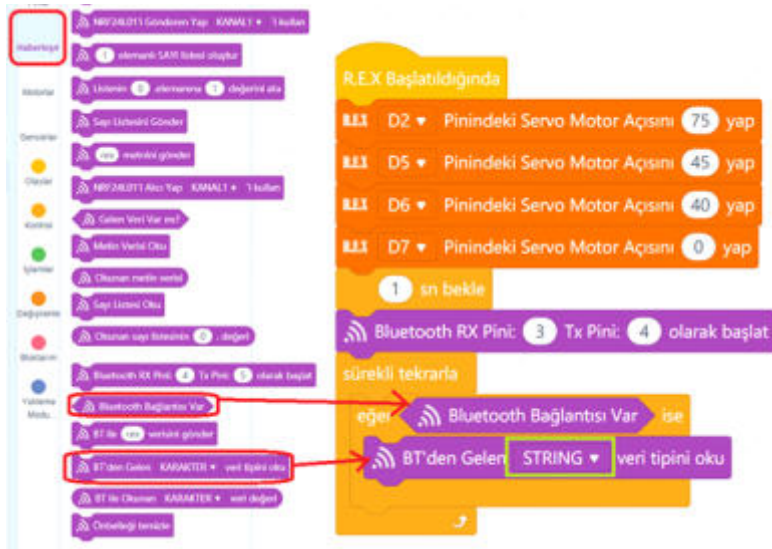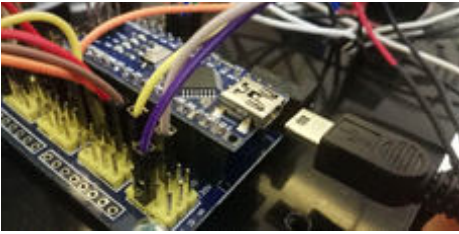

After reading the incoming one, we will make a comparison. Since it will only come with one at a time, we will make each comparison with the previous one when it is not. For this, let's drag and drop 8 of the control blocks, if not, to the areas of each other if they are not, as in the image below.

Since we will make a textual comparison with the incoming, apple a is included in the transactions category. Let's place it in each of the condition fields of the block as in the image below.

We will place the STRING data read from bluetooth in the apple written area. In the a part, we will write the contents of the text that will come from the bluetooth. BT Voice Control for Arduino app converts your speech into text using google voice recognition. When sending as a *, it is placed at the beginning and the # character at the end. Therefore, when we say forward, it will come to us as *Feature#. Let's fill in our sexual comparisons by writing as in the image below.

At the end of the coding process, we will determine the angles of our appropriate servo motors according to the letter. We will use servo blocks from the Motors category. According to the table on page 4, let's change the pin numbers and angle values of our servos as in the image.


The coding phase is atti. The full image of the codes 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 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 that your CH340 chip card is connected to. If connection is not possible, you can select other COM port numbers from the drop-down list.
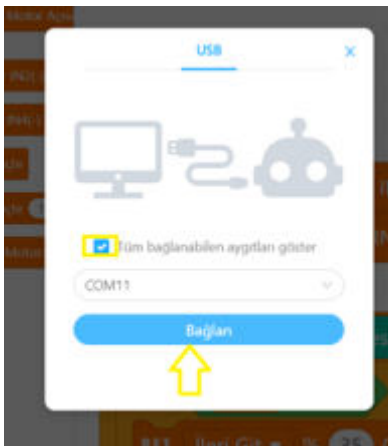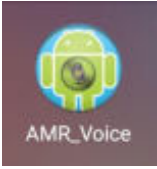

Once connected, "disconnect", "settings" and "Install" buttons will be active. Let's upload the code to our robot by clicking the upload button.


Operating the System
You can feed the Armbot from the Arduino Nano shield with a voltage between 7-12 V, or you can feed it with a 5V powerbank from the Usb port.

We start our application, which is downloaded and installed on our device.

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.
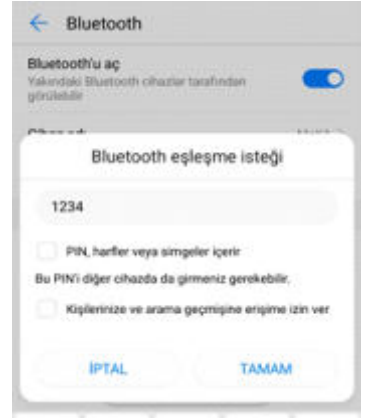
We can now open our app. If the Bluetooth connection of our phone is not active, the application will automatically stream your Bluetooth connection when opening. When the application is opened, you should see the screenshot below.

Now we will connect our robot to our application. We have to make the power connections of our robot. After making the power connections to the jag input of the sensor shield and the motor driver, the red led on the HC06 module will start to flash. A flashing response means that the bluetooth device is not connected. In order to connect, we click on the 3 dots icon in the upper right corner and give the "Connect Robot" command.
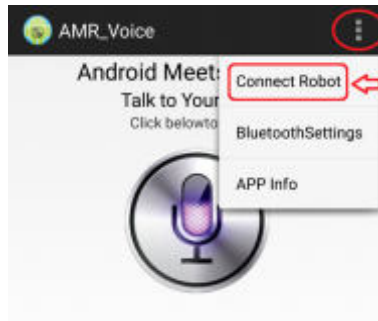
A list of bluetooth devices paired with our device will appear. Scroll through the list and find the HC-06 and click on it.

A list of bluetooth devices paired with our device will appear. Scroll through the list and find the HC-06 and click on it.

After receiving the notification that our application has been connected to our robot, the LED of the HC06 module on our robot should turn solid red. We can now give voice commands from our application. We click on the microphone icon and say the words we determined during coding.
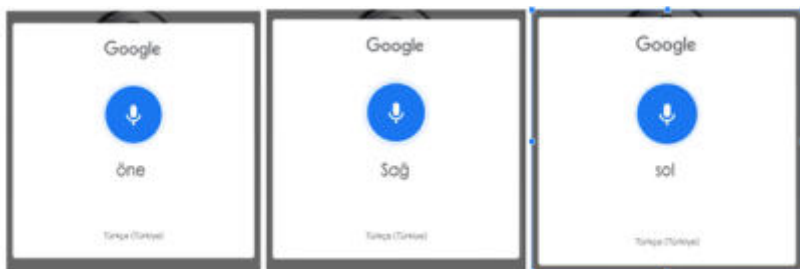
When the voice command is detected and converted to text, we will hear a short beep from our device. 1-2 seconds after this sound, it is sent to our robot with our speech and it gives a response as soon as it receives food with the robot.

You can code your robot to react differently to different words.

POSSIBLE PROBLEMS AND SOLUTIONS
If the armbot is constantly moaning and trembling, it does not detect the turn commands; Make sure that the starting positions and moving angles of the servos are set correctly. If the servos are set to angles where they cannot rotate, they may tremble. This also affects the movements of other servos.

If the Bluetooth connection gets disconnected after a while, this may be a problem when the batteries get weak in systems powered by 6-pin batteries. Another reason is that when servo motors are forced to the direction that they cannot rotate, they can draw a lot of current and create fluctuations in the system and the bluetooth connection can be broken. The last point is that your Arduino Nano shield or HC-06 module is broken.

If you hear crackling noises from the armbot when it is not working, this is normal. Some servos carry the weight of the arm. This causes the servo motor to consume power and sizzle.
If the Armbot does not move even though your voice identification is made and the bluetooth connection is provided; Pay attention to the capitalization of incoming messages. Left is capitalized while Right is capitalized.

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