REX

REX EVOLUTION SERIES
SUPER STAR TRANSFORMERS
8 IN 1

ArmBot

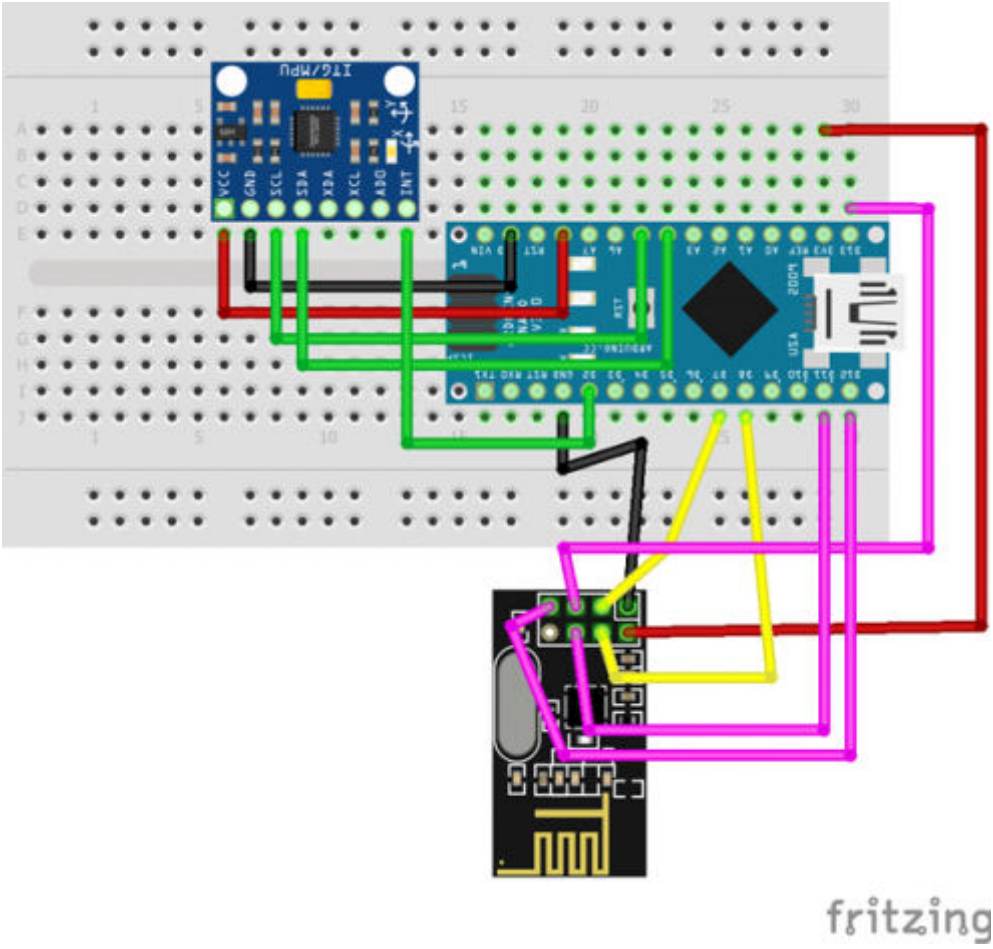
Glove Controlled Armbot

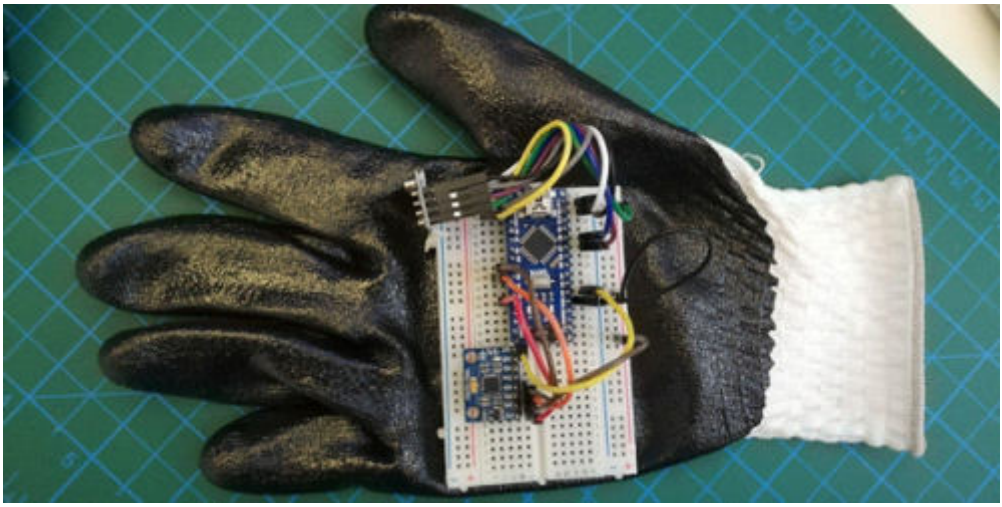
Author:**Mustafa Kemal AVCI**

With a glove on your hand, you can control the remote robot arm with very short delay times. When you bend your hand to the right and left, the Armbot will move to the right and left, when you bend your hand to the front and back, the robot arm will reach forward and pull back, the robot arm will go down when we suddenly slide our hand down, the robot arm will go up when you suddenly slide your hand up, and the robot arm will close when you suddenly slide your hand to the left. When you suddenly move your hand to the right, it will open the gripper.

Glove Circuit Diagram And Design

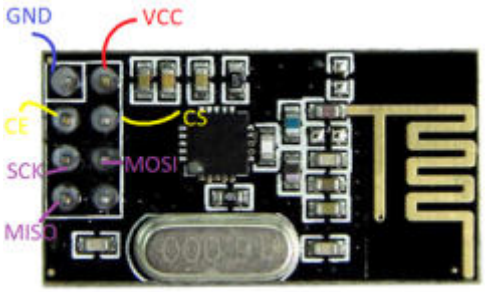
After completing the installation of your robot, make the connections on the gloves with 10 cm jumper cables using a breadboard, in accordance with the circuit design below. In order for the sticky surface on the bottom of the breadboard to stick to the glove, peel off the yellow foil and stick it to the top of the glove.





If you soldered the MPU 6050 gyro sensor's headers downwards as in this photo and used it directly by sticking it into the breadboard, use it with the pin pointing towards your wrist so that the sensor does not detect your hand movements, front-back, left-right reverse.

Since the pin names are not written on the NRF24L01 wireless module, you can use the image below.



MPU 6050 Gyro	INT	**D2**
	SDA	**A4**
	SCL	**A5**
	VCC	**5V**
	GND	**GND**

NRF24L01	SCK	D13
	MOSI	D11
	MISO	D12
	CS	D8
	CE	D7
	VCC	3.3V
	GND	GND

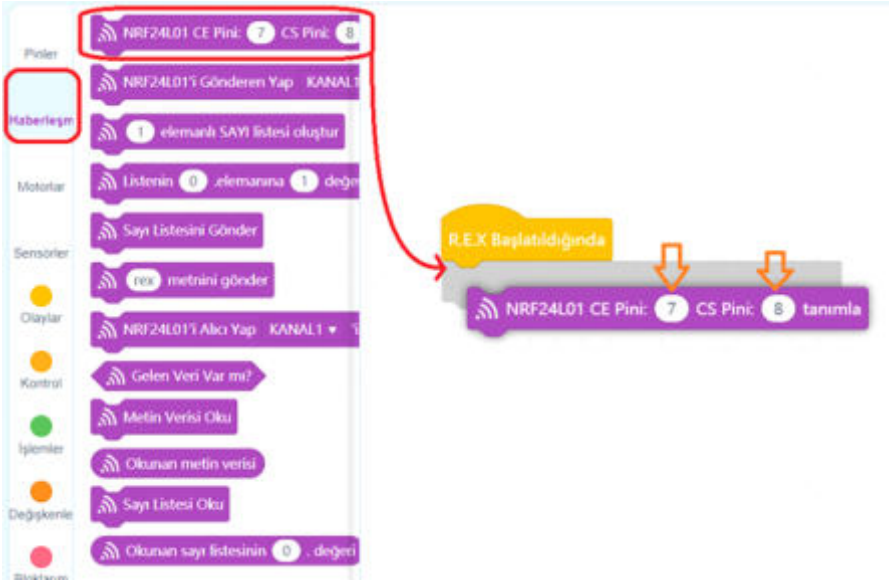
Glove System Coding
Let's move on to the coding phase by adding the mBlock 5 software R.E.X from the starting device library.

Our algorithm will be as follows;

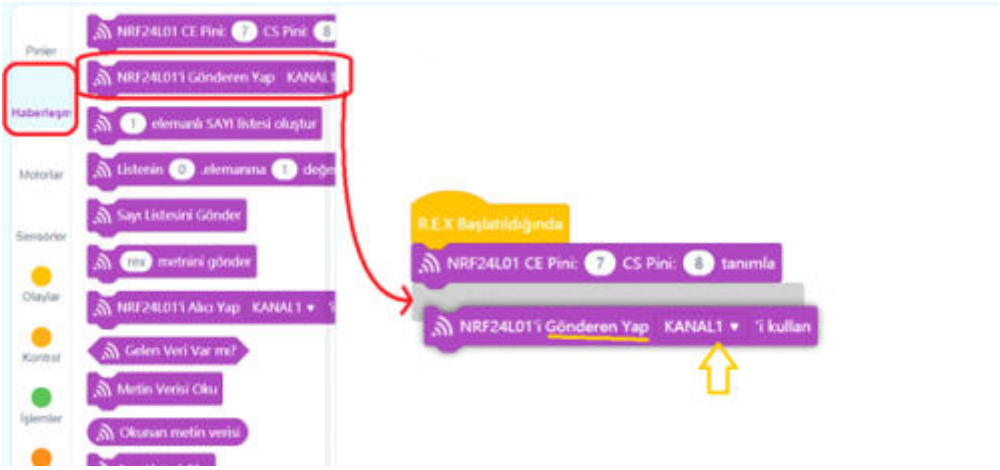
1. Start
2. Identify NRF24L01
3. Define NRF24L01 as SENDER
4. Initialize the accelerometer (MPU 6050)
5. Accelerometer Read
6. When scrolling down detects motion, send text "down" via wireless
7. When up slide detect motion, send text "up" via wireless
8. Send "turn off" text via wireless when motion to left slide is detected
9. Send "ac" text via wireless when right shift motion detected
10. Backbend Detected, send text "back" via wireless
11. When forward leaning detected, send text "on" via wireless
12. Left tilt When detected, send text "left" via wireless
13. Right tilt When detected, send text "right" via wireless
14. If no motion is detected, send the text "flat" via wireless
15. Go to step 5
16. Stop

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

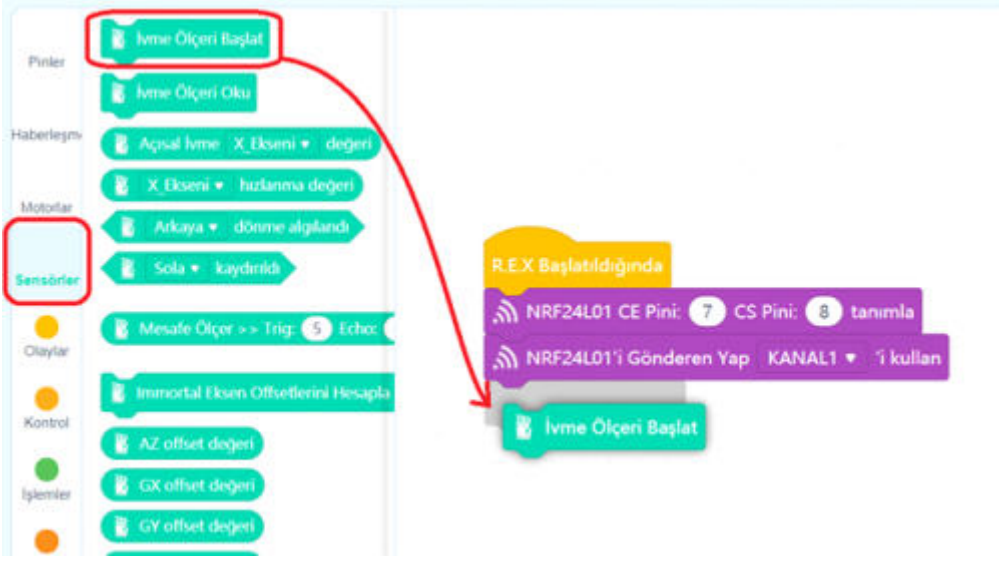
Let's drag and drop the block where we will define our NRF24L01 wireless module from the "Communication" category, as expressed in the image below. Let's check the output numbers that we connect the CS pin and CE pin.



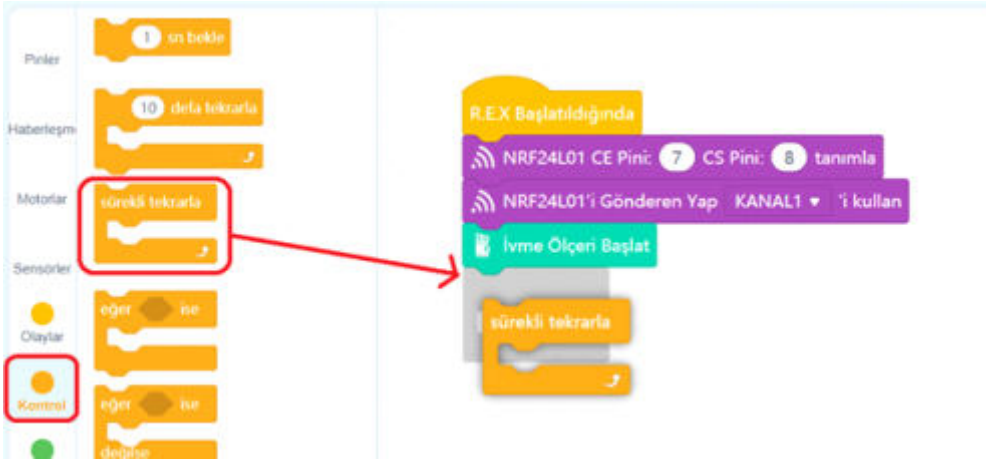
Since the NRF24L01 wireless communication module can be used as both a sender and a receiver, we need to set this feature. If we are going to use both the receiver and the sender in the same project, we must determine the Channel numbers separately. In this project, we are making our NRF24L01 module SENDER to use CHANNEL1, as it will only send the data produced by the gyro sensor to the robot as the SENDER in the glove system.



Now let's define the accelerometer. In this way, let's ensure that the MPU 6050 Gyro sensor is ready for use for our glove system. Let's drag and drop the Start Accelerometer block from the Sensors category as in the image below.

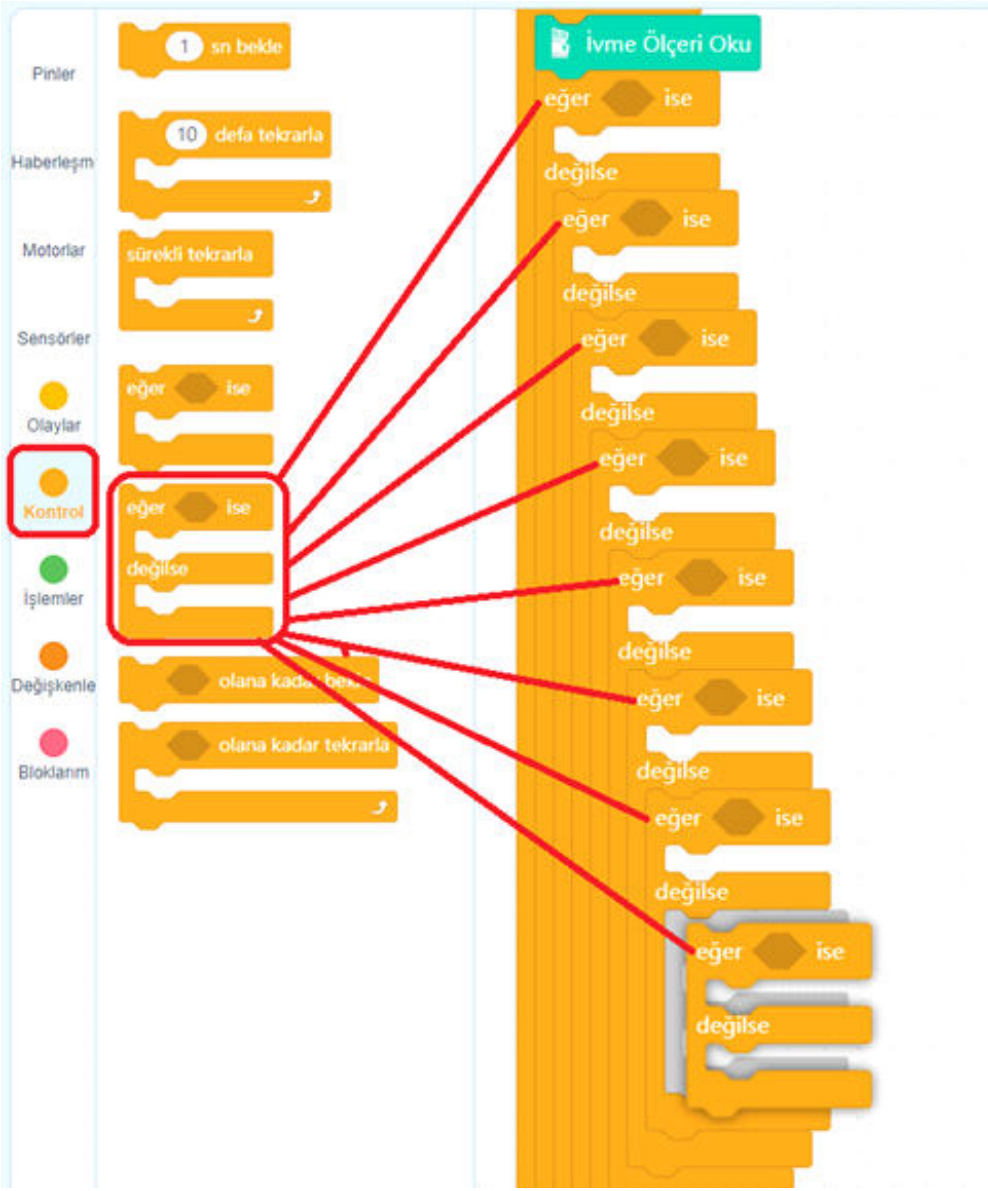


By continuously reading the accelerometer, we will set up an endless loop so that the change in our hand movements can be transmitted instantaneously wirelessly. For this, let's drag and drop the Repeat Continuously block from the Control blocks.

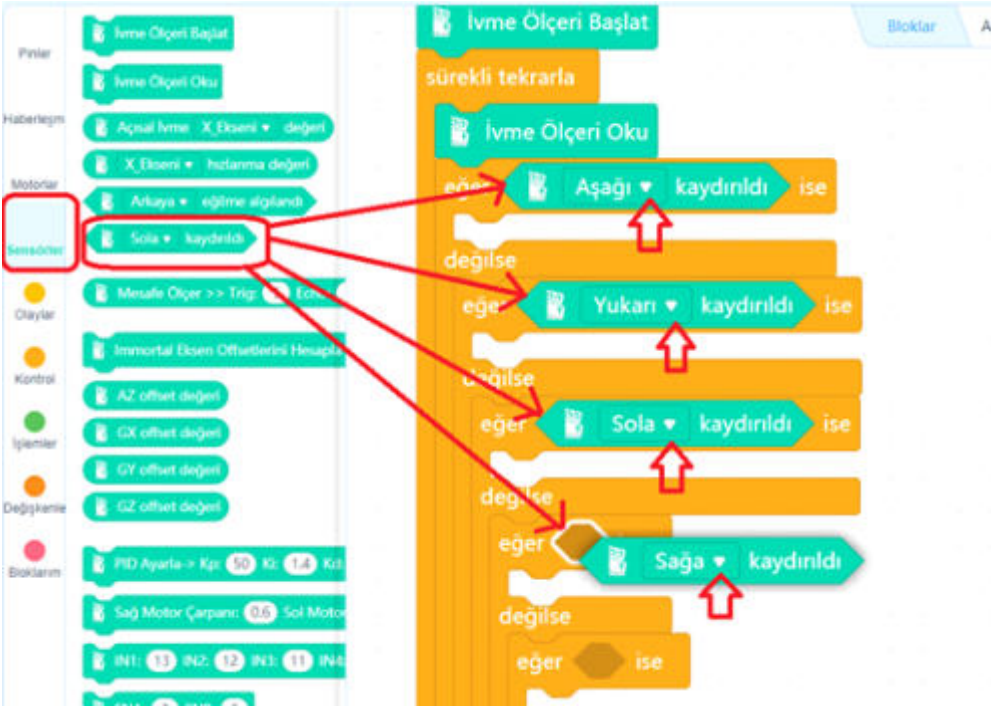


Each time the loop returns to the beginning, it must first read the values of the axes from the accelerometer. Because the comparisons we will make in the following steps will be based on the values we get as a result of this reading. Let's take the Accelerometer Read block from the Sensors category and place it inside the repeat continuously block.

Now we can create our conditional statements and start making our comparisons. Each of our conditions will be tested if the previous condition is not met. So, let's take the If If Otherwise block from the control blocks and place it just below the Accelerometer Read block. If not, let's place them in the areas of each other as in the image below.

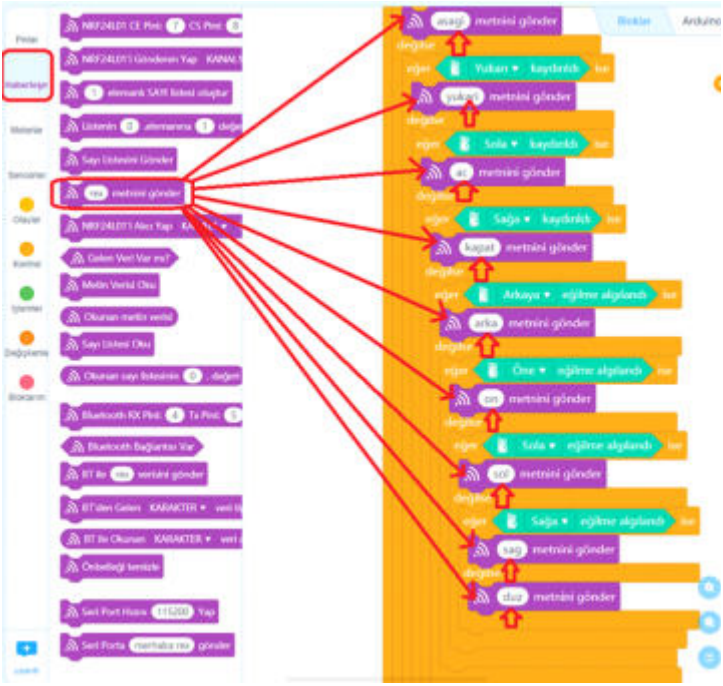


Our first four conditions will be the sliding movements of our hands. We are using the Left shift detected block in the Sensors category. We change the direction information and set it as in the image below and drag and drop it.

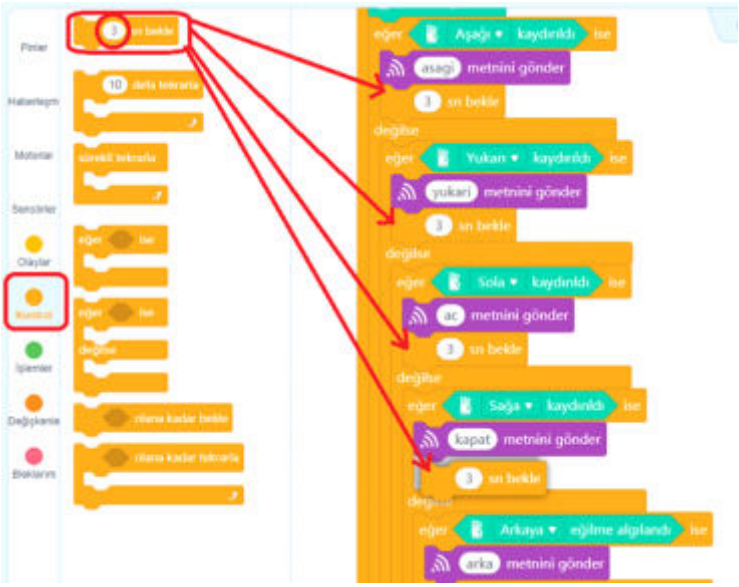


The next four conditions are the perception of the forward, backward, right and left bending motion that our hand will make. Let's change the direction of the Back bend detected block in the Sensors category and drag and drop the remaining four into the condition area of the If not block, as in the image below.

Now, when these conditions are met, let's send the messages to be sent wirelessly. For this, we will use the send rex text block in the communication category. We will delete the rex text and write the following with "I will send" and drag the blocks to their appropriate places and leave them as in the image below. If we did not bend our hand in any direction, we should send this situation as an as we do not want the robot arm to move. For this, we write "flat" in the menu field, which will be sent to the if not field of the last if-if-not block.

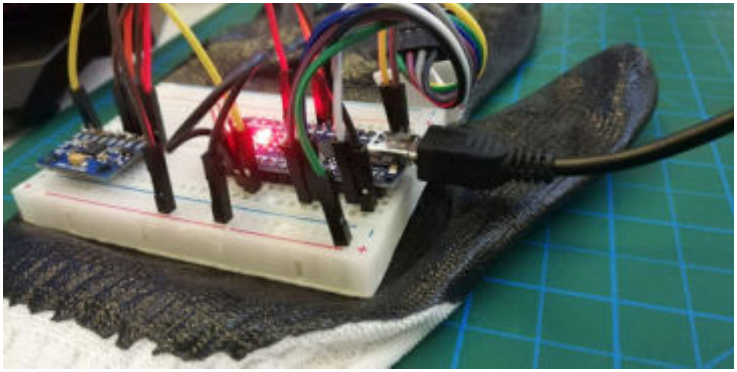


Let's drag the 3-second waiting blocks into the slide movements and let them drop the block in the Control category, as in the image below, in order to perceive the sliding movements of our hands in a healthy way.

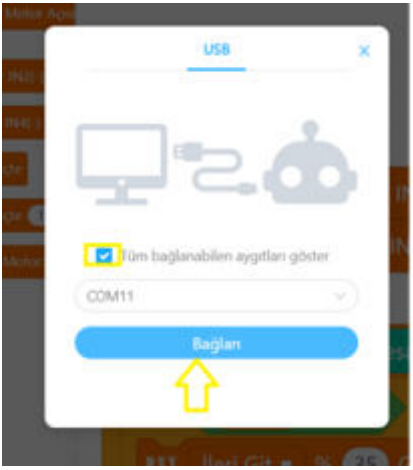


when we have completed the codes it should be as follows.

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 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.

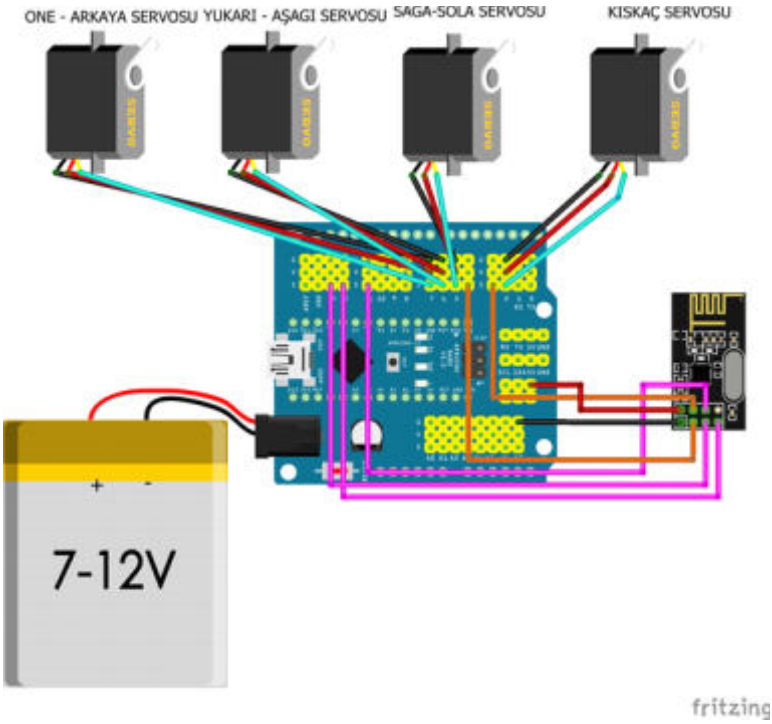


When the connection is established, the "disconnect", "settings" and "Install" buttons will be active. Let's upload the code to our robot by clicking the upload button.

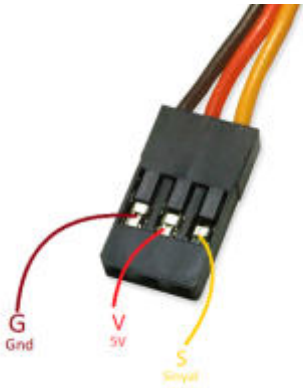
We have completed the codes of the glove system. At this stage, the robot arm will not move. Because we did not upload the necessary codes to the robot arm for the robot arm to understand the wireless sent by the glove. Now let's prepare the software for my robot arm.

Armbot Connection Diagram

Armbot will instantly read the messages coming from the glove system and change the angle of the servo motors according to the incoming data, and make forward, backward, right-left, up, down and open and close movements. Let's make the connections of our servos in accordance with the connection diagram below.



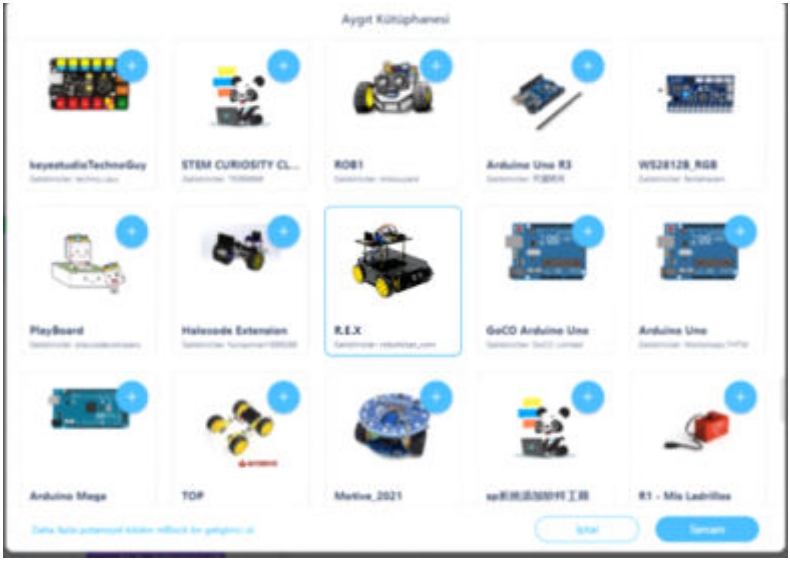
Let's connect the Sg 90 servo motors to the pins on the Arduino Nano shield, corresponding to the pins shown in the image below.



Coding the Armbot

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.



I drag and drop the block when 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.

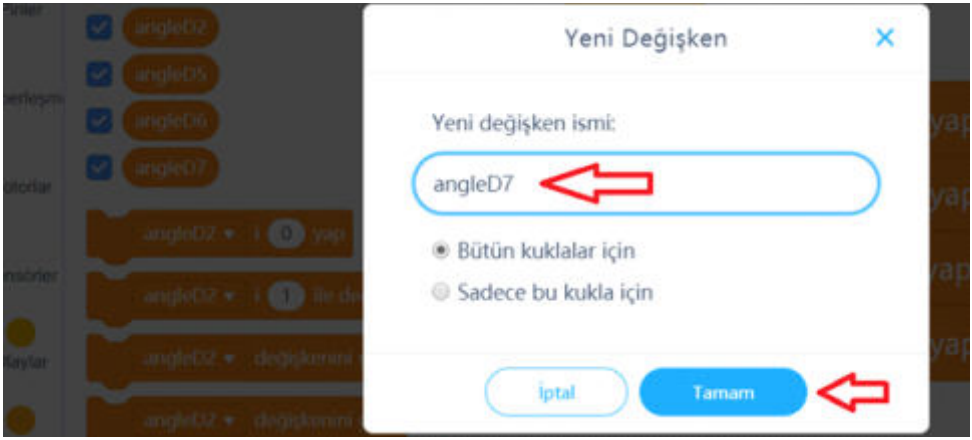
When we connect the Arduino nano to the computer and upload this code, let's observe the action of the servo. 75 degrees is the angle we will use to close the clamp. To open the clamp, it is enough to be at 15 degrees. Try this out to determine the most suitable angle range and make a note of it. We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

Hareket Adı	Servo P n No	Açı Aralığı	Başl angıç açısı
Aç-Kapat	D2	15-75	75
Aşağı -Yukarı	D6	0-40	40
Öne-Arkaya	D7	40-0	0
Sağa-Sola	D5	15-85	45

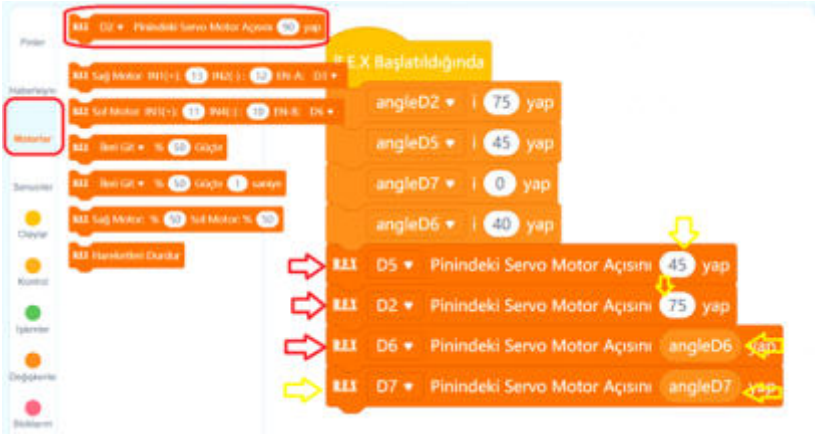
As you adjust the servo angles, you may have noticed that the servos turn very quickly. Moving so hard and fast sometimes makes remote control difficult and also causes the screws of the robot to loosen more quickly. Therefore, we will ensure that our robot arm moves smoothly by changing the angles of the servos by 1 or 2 degrees in 0.05 seconds.

In order to change the angle values by slowly increasing or decreasing, we will store the angle values of the servos in the variables and perform mathematical operations on them. Therefore, we create variables named angleD2, angleD5, angleD6 and angleD7 and drag and drop them as the starting positions under the block when the rex starts, as in the image below.
Let's create our variables in order by clicking the Create a variable button from the Variables category.



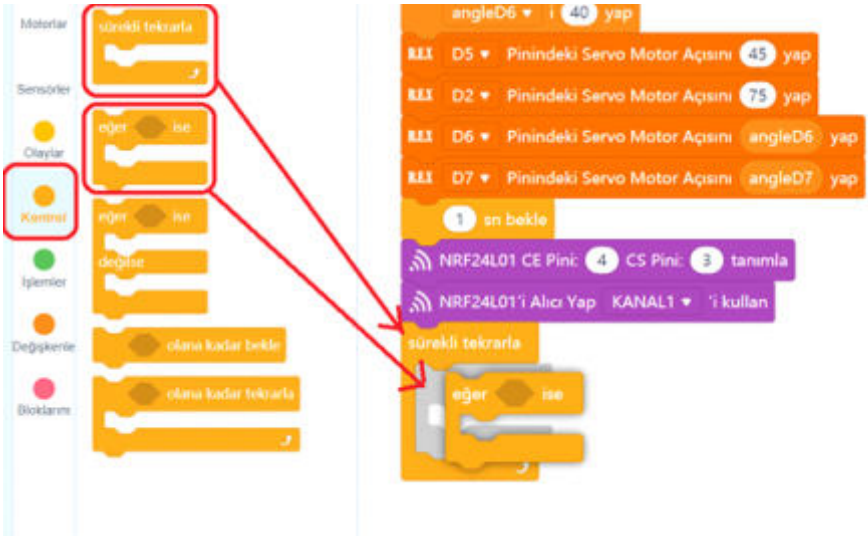
Now as soon as Rex starts, we have to get all of our servos to their starting position. So, let's drag and drop the Servo block in the motors category and write the starting positions of the servos as in the image below. You can set the servo angles either from the keyboard or by using the value blocks of the variables you have created. It is done in two different ways below.

In order to change the angle values by slowly increasing or decreasing, we will store the angle values of the servos in the variables and perform mathematical operations on them. Therefore, we create variables named angleD2, angleD5, angleD6 and angleD7 and drag and drop them as the starting positions under the block when the rex starts, as in the image below.
Let's create our variables in order by clicking the Create a variable button from the Variables category.

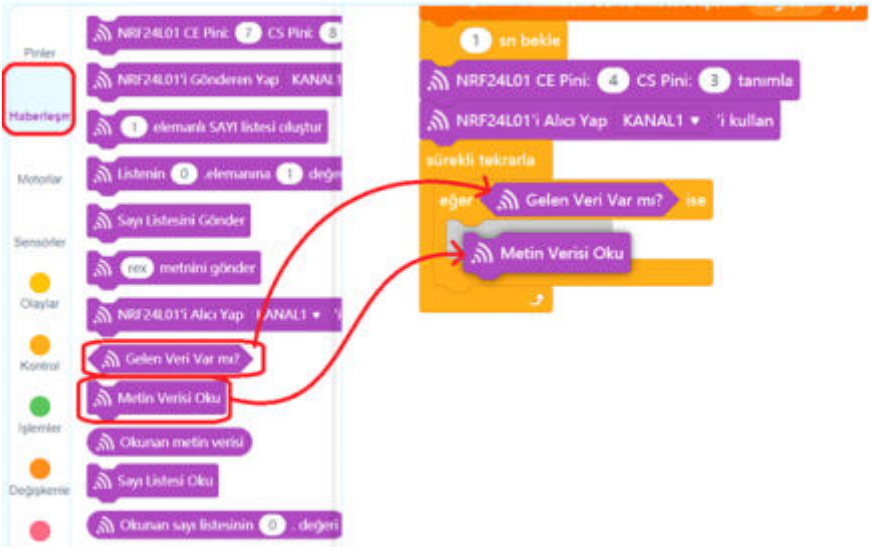


Let's wait 1 second and let the servos come to their starting position. Then, let's define the NRF24L01 module and set it as a receiver in CHANNEL1. These commands are shown in the image below. Because NRF24L01 in the glove system will send data from CHANNEL1.

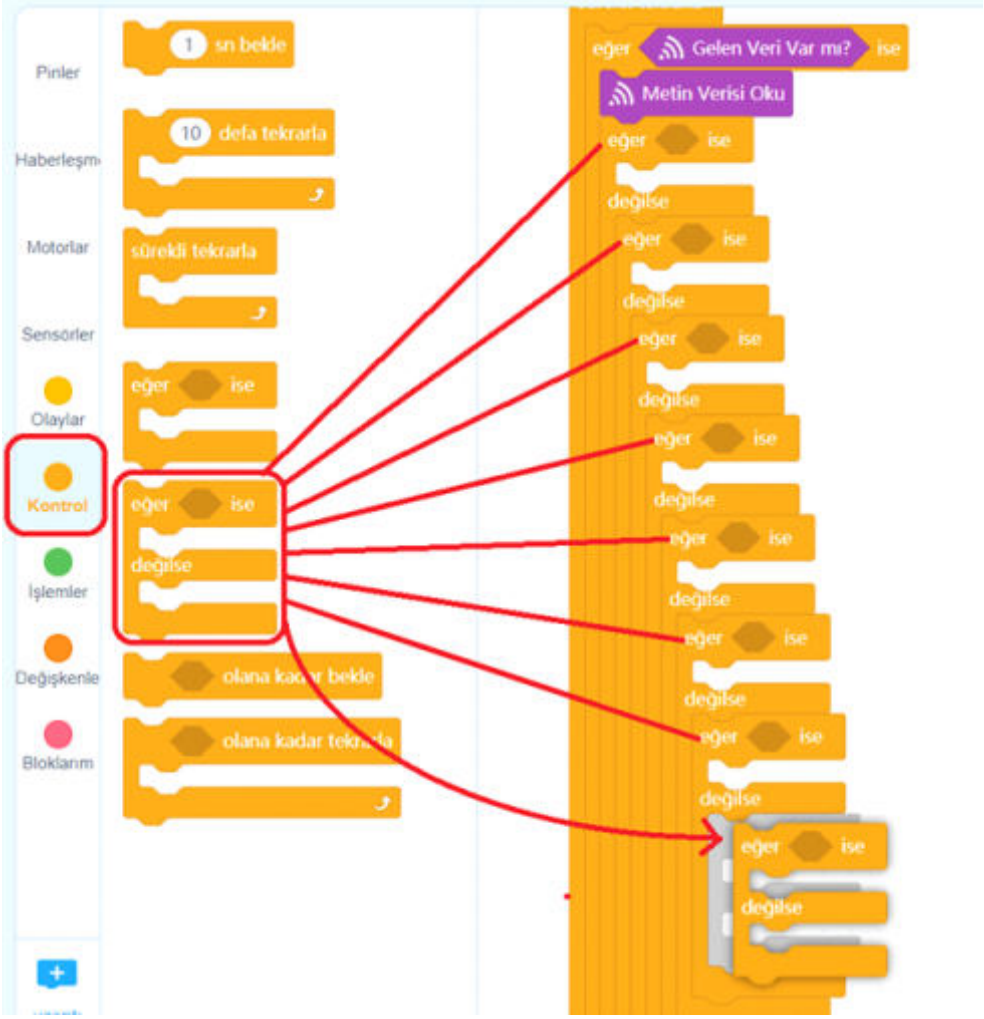
Now we can create our infinite loop. In the endless loop, if wireless data is detected, let this data be read and let's prepare the servo motor movements by checking the word in the read data. Let's drag and drop the Repeat Continuously from the control blocks. Right after that, let's drag and drop the If block into the Repeat forever block.



If there is incoming data, Men will be read and reacted according to the incoming text, so let's drag and drop our code blocks shown below from the communication category.

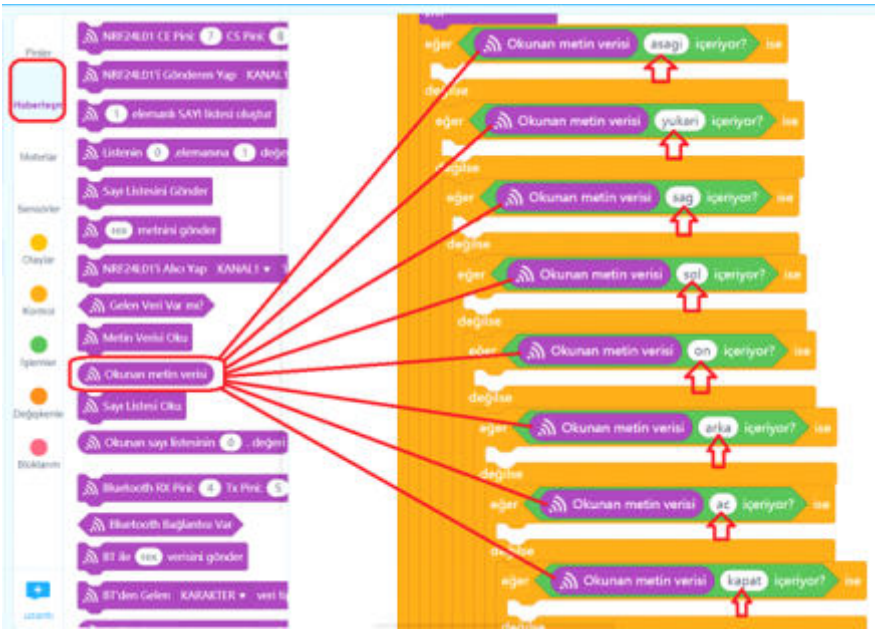


It will come to our robot wirelessly with a total of 8 different options: front, back, left, right, up, down, open, close. Since only one message will be reacted in a single read, we will make all our conditions in the if not block. Let's place each condition in the field if the previous condition is not. Let's drag 8 of the If If Not Blocks in the control blocks and leave them to the fields if they are not, as in the image below.



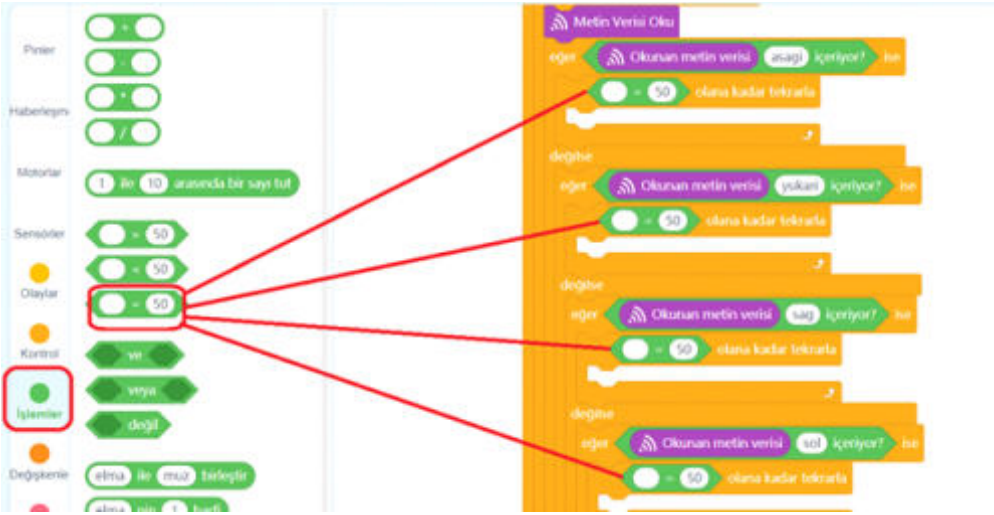
To compare the read menu data, the apple in the Transactions category contains a? Let's drag the block to all condition fields and leave it as in the image below.

We will compare the read me n data with down, up, ten, back, right, left, open, close. After dragging the menu data block read from the communication blocks, let's write the "a" statements and their contents as in the image below.



Now it's time to change the servo angles. Each servo will act in two directions, that is, its angle will increase or decrease. For example, we will adjust the servo on pin D6 in response to the up and down commands. For downward movement, the angle must decrease to 0 (si r), and for upward movement, the angle must increase to 40 degrees. We take the Repeat Until Happening block from the control blocks and fill in the fields of all our conditions, if any, as in the image below.

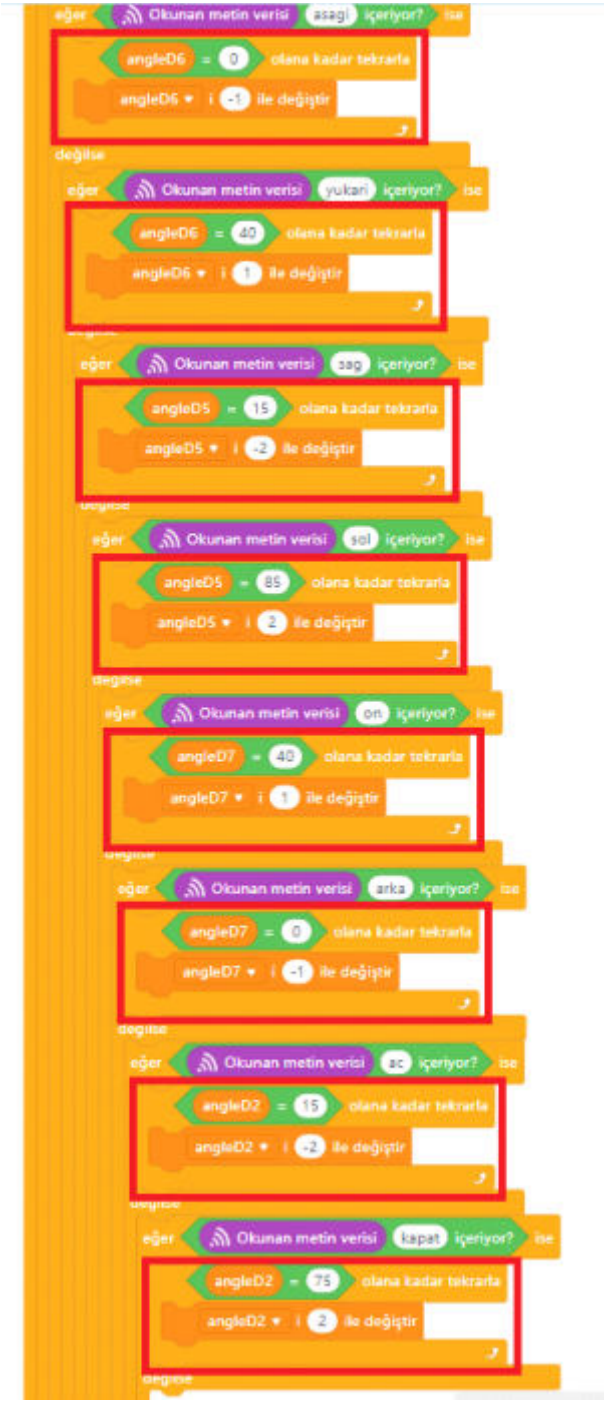
Let's drag the equal block that we will use to compare the repeat blocks until it happens from the transactions category and leave it as in the image below.



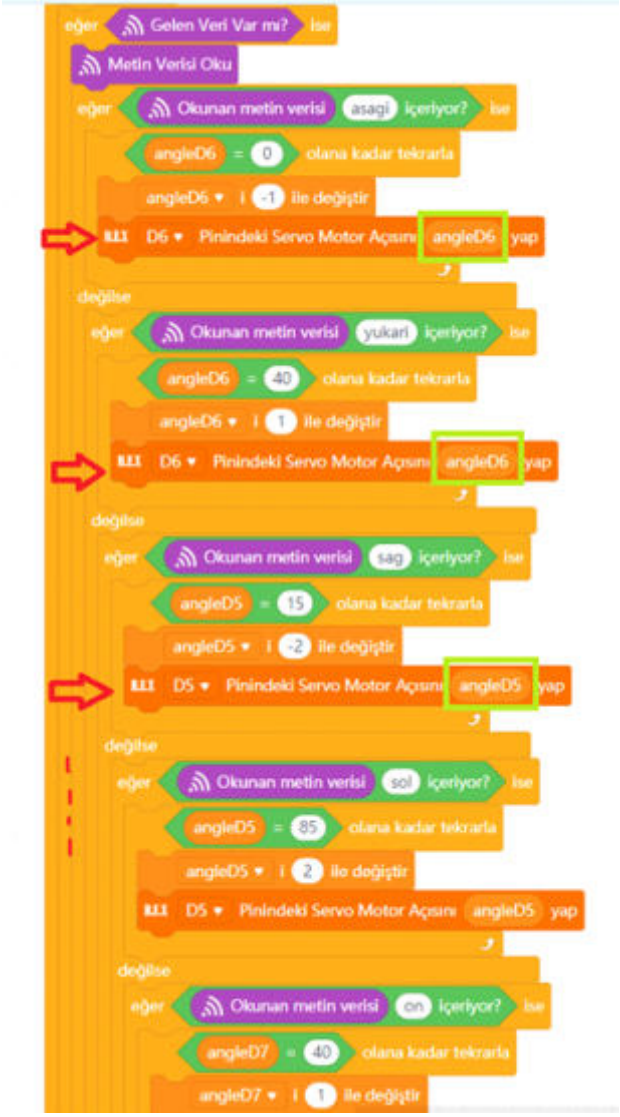
Let's fill in our condition fields according to the table below. The angle values here are made according to the calibration settings of our robot arm. You should determine the servo angles of your robot and determine the angle values accordingly.

command	variable	condution
down	angleD6	until 0/ change to -1
up	angleD6	up to 40/1 change
right	angleD5	until 15/ change to -2
left	angleD5	until 85/ 2 change
front	angleD7	until 40 / 1 change
back	angleD7	change to / -1 until 0
on	angleD2	until 15/ change to -2
off	angleD2	until 75/ change 2

Let's prepare our blocks according to the table as follows.


eğer Okunan metin verisi asagi içeriyor? ise
angleD6 = 0 olana kadar tekrarla
angleD6 i -1 ile değiştir
değilse
eğer Okunan metin verisi yukari içeriyor? ise
angleD6 = 40 olana kadar tekrarla
angleD6 i 1 ile değiştir
eğer Okunan metin verisi sag içeriyor? ise
angleD5 = 15 olana kadar tekrarla
angleD5 i -2 ile değiştir
eğer Okunan metin verisi sol içeriyor? ise
angleD5 = 85 olana kadar tekrarla
angleD5 i 2 ile değiştir
eğer Okunan metin verisi on içeriyor? ise
angleD7 = 40 olana kadar tekrarla
angleD7 i 1 ile değiştir
eğer Okunan metin verisi arka içeriyor? ise
angleD7 = 0 olana kadar tekrarla
angleD7 i -1 ile değiştir
değilse
eğer Okunan metin verisi ac içeriyor? ise
angleD2 = 15 olana kadar tekrarla
angleD2 i -2 ile değiştir
eğer Okunan metin verisi kapat içeriyor? ise
angleD2 = 75 olana kadar tekrarla
angleD2 i 2 ile değiştir
değilse

After calculating the values of our variables, we need to transfer these values to the servo motors. Let's place the servo blocks from the motor blocks inside the loops and drag and drop the values of the variables into the angle fields of the servos.



Let's place a 0.05-second wait block just below the servo blocks so that the changes in the variables are reflected softly on the movements in the servos, as in the image below. We drag and drop a total of 8 0.05 second waiting blocks.

That's it for Armbot's codes. The final state of all blocks should be as in the image below.

Operating the System
Run the glove system by feeding the Arduino Nano from the USB port with a 5V powerbank. You should use it with the breadboard on your hand while the powerbank is in the palm of your hand. You can understand that data is being sent from the very fast flashing of the leds on the Arduino nano.

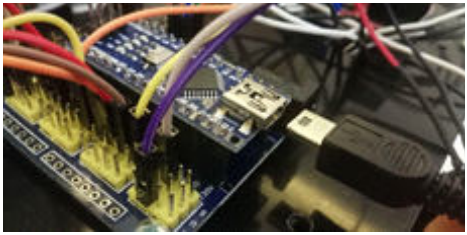
You can feed the Armbot from the Arduino Nano shield with a voltage between 7-12 V or you can feed it with a 5V powerbank from the Usb port. Feeding it with 5V from the USB port provides a more stable operation of the robot arm.

Slowly tilt your hand back and forth, try to learn the motion detection points. Try to do the tilts slowly, as very fast tilt movements may be perceived as panning movements.

You will make the robot arm move up and down and open and close movements by sliding your hand. Putting your hand down abruptly without any shaking will cause the arm to go down, and if you do it up quickly without any shaking, the robot arm will go up. Moving your hand to the right abruptly without trembling will open the grip of the robot arm, and sliding it to the left will close the grip of the robot arm.

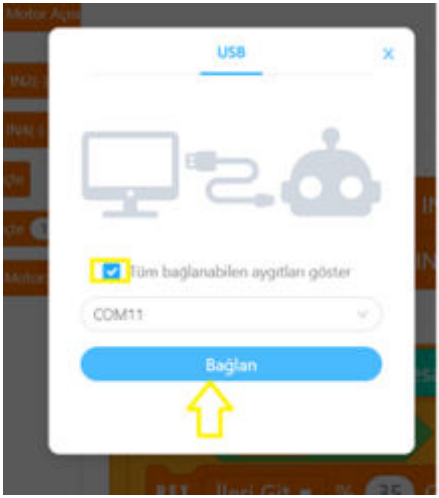
In the parts where scrolling operations are used, 3-second pauses are placed. This is because the accelerometer sensor can accurately detect the next movement. It takes approximately 2-3 seconds for the sensor to reach a fixed position after the sliding operations. That's why you should use a 3-second wait inside the detected state when checking for scrolling.

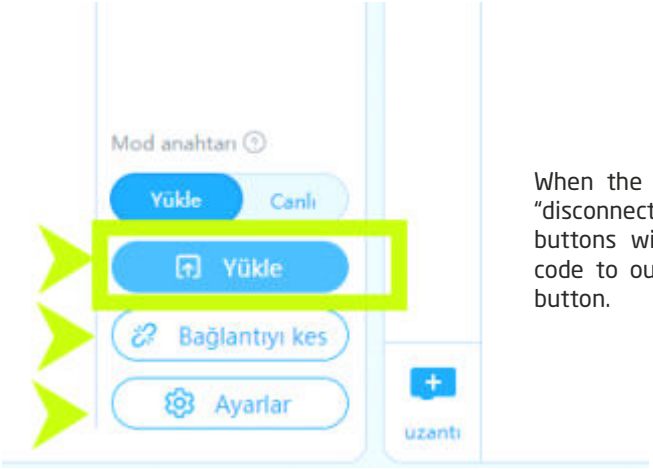
In our trials, Gloves and Armbot successfully communicated at a distance of 7-10 meters in the open area, and Armbot was able to process commands.



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 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.





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.

Possible Problems and Solution Suggestions
If there is no movement in the robot arm even though you do your glove movements; Check if there is data transfer. The LED on the Arduino nano should be blinking very quickly. Otherwise, check the connections of the cables.
If the glove's right-to-left, back-to-back bending movements are perceived in reverse; You must rotate the sensor and attach it to the breadboard, or you must swap the right, left, front, back statements of the code blocks with each other.

If right left front back open close up and down reverse is detected; You can change the directions on the gyro sensor blocks in the glove system.
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 Armbot misunderstands commands; Especially in swiping movements, the movement must be done suddenly and quickly. While the breadboard is on the glove, it can move left and right during the sliding movement and the movement can be perceived that way. At the same time, if we make the tilting movements too fast, they can be perceived as a scrolling movement. If we do the bending movements in a slow and controlled way, and the sliding movements in a short, smooth and straight way, the commands are given more stable.

If the Armbot sometimes stops detecting commands, fluctuations in the current may affect the wireless communication. Make sure the motors are running smoothly and your battery is fully charged. You may have a faulty servo motor. Replace your servo motor, which groans at every angle and makes noise even when there is no load, with a spare one.

If the Armbot reacts very slowly to the movements, the next movement following the slide will be detected after 3 seconds, since there is a 3-second delay after the slide movements. If you want to speed up the movements of the arm, you can increase the amount of change in the value or shorten the time while the servo angle of the arm is changing.

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