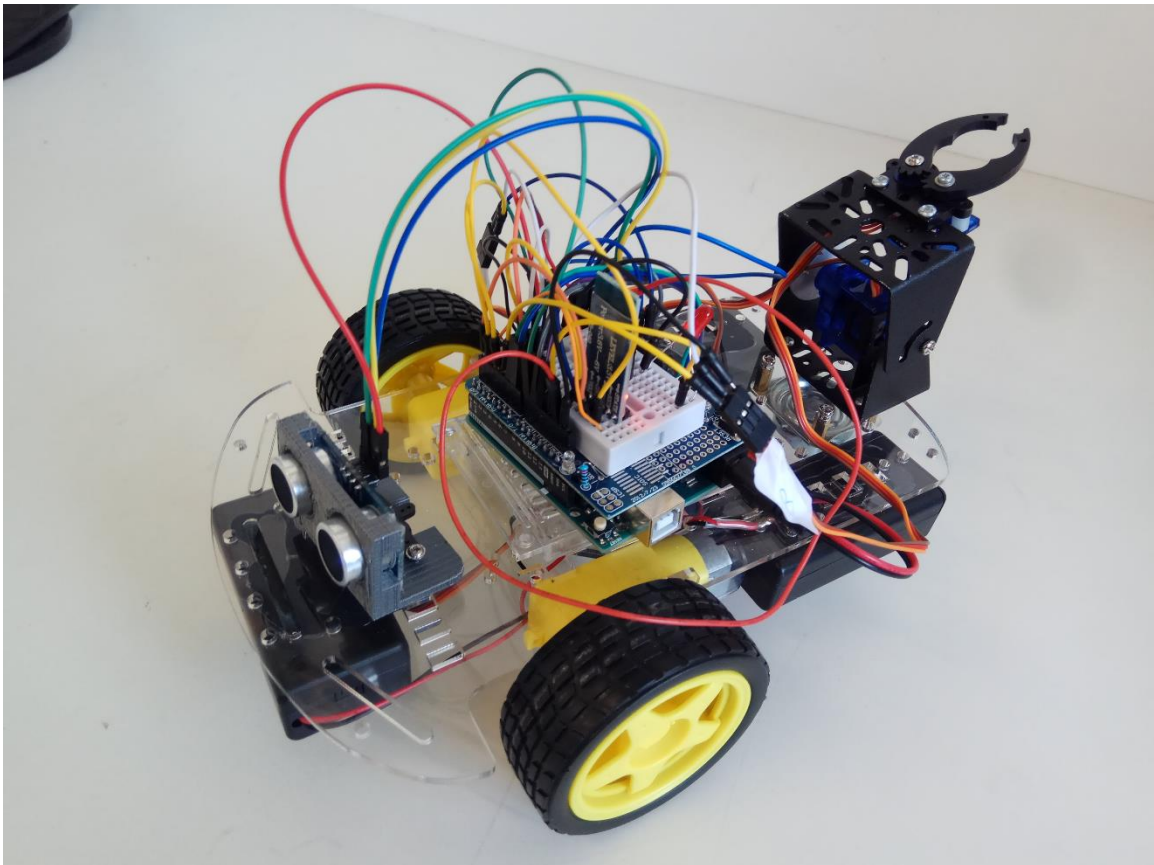


ΡΟΜΠΟΤ ΜΕ ΠΡΟΣΘΗΚΗ ΒΡΑΧΙΟΝΑ



Σε αυτό το βήμα, θέλοντας να δώσουμε στο ρομπότ μας την ικανότητα της μεταφοράς αντικειμένων, προσθέσαμε έναν **βραχίονα**. Ο πρώτος βραχίονας που χρησιμοποιήσαμε έχει δύο αρθρώσεις: καρπό και δαγκάνα, ώστε να μπορεί να κρατά ή να αφήνει ένα αντικείμενο και ταυτόχρονα να το ανασηκώνει ή να το αποθέτει κάπου. Οι δύο αρθρώσεις του βραχίονα απαιτούν τη χρήση δύο servo motors αντίστοιχα. Συνεπώς πρέπει να χρησιμοποιήσουμε άλλα δύο PWM pins του Arduino.

Έτσι στον **καινούργιο κώδικα**, πρέπει να ονομάσουμε αρχικά τους δύο νέους servo. Για κάθε άρθρωση(servo), έχουμε δύο κινήσεις. Για την δαγκάνα: ανοιχτή και κλειστή. Για τον καρπό: πάνω- κάτω. Αυτό γίνεται εύκολα δίνοντας σε κάθε servo δύο γωνίες περιστροφής. Στην περίπτωση μας, μετά από δοκιμές καταλήξαμε για τη δαγκάνα: ανοιχτή(90°)-κλειστή(0°), ενώ για τον servo καρπό: πάνω(90°)-κάτω(170°). Οι γωνίες που επιλέγονται εξαρτώνται προφανώς από τον προσανατολισμό των σερβοκινητήρων στην κατασκευή αλλά και από τα ιδιαίτερα μηχανολογικά χαρακτηριστικά της κάθε κατασκευής.

Για να γίνει πιο κατανοητή η λειτουργία του βραχίονα παραθέτουμε το κομμάτι του κώδικα που προστέθηκε:

```
else if(state=='O')    // Open, για να ανοίγει η δαγκάνα.

{myservo1.write(90);}

else if(state=='C')    // Close, για να κλείνει η δαγκάνα.

{myservo1.write(30);}

else if(state=='D')    // Down, ο καρπός κάτω.

{myservo2.write(170);}

else if(state=='U')    //Up, ο καρπός πάνω.

{myservo2.write(90);}
```

Κάθε φορά που στέλνεται ο αντίστοιχος χαρακτήρας, μέσω Bluetooth, ένας από τους servo κινητήρες, στρέφεται σε συγκεκριμένη γωνία και έτσι πραγματοποιούνται οι κινήσεις του βραχίονα.

Στην android εφαρμογή προστέθηκαν τέσσερα κουμπιά(Buttons) με τις ονομασίες open-close-up-down ώστε πατώντας τα να πραγματοποιούνται οι αντίστοιχες λειτουργίες.

Το βίντεο της λειτουργίας του συγκεκριμένου βήματος, φαίνεται στον παρακάτω σύνδεσμο:
<https://youtu.be/bVy3IUawqbQ>

Ο κώδικας για το ρομπότ με τον βραχίονα

```
#include <Ultrasonic.h>

//Ultrasonic    (Trig,Echo)

Ultrasonic sensor (A0,A1);

int distance=100;

//BLUETOOTH

char state='M';
```

```
char mode='m';

int i=0;
int j=0;

//L298 pins
const int motorR1 = 4; // ΔΕΞΙΟ ΜΟΤΕΡ ΜΠΡΟΣΤΑ
const int motorR2 = 3; // ΔΕΞΙΟ ΜΟΤΕΡ ΠΙΣΩ
const int motorL1 = 8; // ΑΡΙΣΤΕΡΟ ΜΟΤΕΡ ΜΠΡΟΣΤΑ
const int motorL2 = 7; // ΑΡΙΣΤΕΡΟ ΜΟΤΕΡ ΠΙΣΩ

#include <Servo.h>

Servo myservo1;
Servo myservo2;
Servo myservo3;
int pos;
int leftDistance;
int rightDistance;
//Led
const int led = 12;
//Speaker
const int speaker = 13;

void setup(){
  Serial.begin(9600);

  // Set pins as outputs, inputs:
```

```
pinMode(motorR1, OUTPUT);  
pinMode(motorR2, OUTPUT);  
pinMode(motorL1, OUTPUT);  
pinMode(motorL2, OUTPUT);  
pinMode(led, OUTPUT);  
pinMode(speaker,OUTPUT);
```

```
noTone(speaker);
```

```
myservo1.attach(5);//δαγκάνα  
myservo2.attach(6);//πανω κατω  
myservo3.attach(9);//λαιμός
```

```
myservo1.write(30);  
myservo2.write(90);  
myservo3.write(90);
```

```
delay(5000);
```

```
//ο servo1 είναι η δαγκάνα. Είναι κλειστή στις 20 μοίρες περίπου και οριζόντια  
ανοιχτή στις 90. Δεν χρειάζεται να ανοίξει παραπάνω.
```

```
//ο servo2 είναι το πάνω-κάτω(μεσαίος servo). κάτω 180 μοίρες, οριζόντια θέση-90  
μοίρες.
```

```
// ο servo3 είναι ο λαιμός, 90 μοίρες κοιτα μπροστά
```

```
}
```

```
void loop(){
```

```
if(Serial.available() > 0){
```

```
state = Serial.read();
```

```

if(state=='A'){ mode='a';}
if(state=='M'){mode='m';}

stop();

myservo1.write(30);
myservo2.write(90);
}

```

```

if( mode=='m'){
//Bluetooth Mode

```

```

/*****Forward*****/

```

```

// If the "state" is 'F', robot will go forward

```

```

if (state == 'F') { forward(); }

```

```

/*****Backward*****/

```

```

// If the "state" is 'B', robot will go backward

```

```

else if (state == 'B') {

```

```

    backward();

```

```

}

```

```

/*****Left*****/

```

```

// If the "state" is 'L', robot will turn left

```

```

else if (state == 'L') {

```

```

    left();

```

```

}

```

```

/*****Right*****/

```

```

// If the "state" is 'R', robot will turn right

```

```

else if (state == 'R') {

```

```

    right();

```

```

}

```

```

/*****Stop*****/
// If the "state" is 'S', robot will stop
else if (state == 'S') {
    stop();
}

// if the state is 'O', gripper open

else if(state=='O')
{myservo1.write(90);
}

// if the state is 'C', gripper closed
else if(state=='C')
{myservo1.write(30);
}

// if the state is 'D', arm down)
else if(state=='D')
{myservo2.write(170);
}

// if the state is 'U', arm up)
else if(state=='U')
{myservo2.write(90);
}

/*****Lights*****/
//If state is equal with letter 'W', turn leds on or of off
else if (state == 'W') {
    if (i==0){

```

```

        digitalWrite(led, HIGH);
        i=1;
    }
    else if (i==1){
        digitalWrite(led, LOW);
        i=0;
    }
    state='n';
}

/*****Horn sound*****/

//If state is equal with letter 'V', play (or stop) speaker sound
else if (state == 'V'){
    if (j==0){
        tone(speaker, 1000); //Speaker on
        j=1;
    }
    else if (j==1){
        noTone(speaker); //Speaker off
        j=0;
    }
    state='n';
}

}

//*****AYTOMATH ΛΕΙΤΟΥΡΓΙΑ*****

else if(mode=='a'){

```

```

distance=sensor .Ranging(CM);
Serial.print("Distance ");
Serial.println(distance);
delay(10);
if(distance>=20){
    forward();
}
if(distance<20){
    //If an object detected at 20cm, stop the robot and find a way out
    stop();
tone(speaker,500);
delay(1000);
noTone(speaker);
for (pos = 90; pos >= 0; pos -= 1) { // goes from 90 degrees to 0 degrees
    myservo3.write(pos);           // tell servo to go to position in variable 'pos'
    delay(5);                      // waits 5ms for the servo to reach the position
}
//Read new distance from the right side
rightDistance = sensor.Ranging(CM);
delay(10);
//Now look left
for (pos =0; pos <= 180; pos+= 1) { // goes from 0 degrees to 0 degrees 180
    myservo3.write(pos);           // tell servo to go to position in variable 'pos'
    delay(5);                      // waits 5ms for the servo to reach the position
}
//Read new distance from the left side
leftDistance = sensor.Ranging(CM);
delay(10);

```



```
pos = 90; // look forward again  
myservo3.write(pos);
```

```
//Finally compare left and right distances and make the best turn decision  
if (leftDistance > rightDistance){  
  left();  
  delay(150); // Change the time to make 90 deg. turn  
}
```

```
else if (leftDistance < rightDistance){  
  right();  
  delay(150);  
}  
else{ //that means that two distances are equal
```

```
//back for 0.5sec  
digitalWrite(led,HIGH); // turn the led on  
backward();  
delay(500);
```

```
left(); // and turn left
```

```
delay(150); } } }
```

```
//Movement functions
```

```
void forward(){  
  digitalWrite(motorR1, HIGH);
```

```
digitalWrite(motorR2, LOW);  
digitalWrite(motorL1, HIGH);  
digitalWrite(motorL2, LOW);  
}
```

```
void backward(){  
    digitalWrite(motorR1, LOW);  
    digitalWrite(motorR2, HIGH);  
    digitalWrite(motorL1, LOW);  
    digitalWrite(motorL2, HIGH);  
}
```

```
void right(){  
    digitalWrite(motorR1, LOW);  
    digitalWrite(motorR2, HIGH);  
    digitalWrite(motorL1, HIGH);  
    digitalWrite(motorL2, LOW);  
}
```

```
void left(){  
    digitalWrite(motorR1, HIGH);  
    digitalWrite(motorR2, LOW);  
    digitalWrite(motorL1, LOW );  
    digitalWrite(motorL2, HIGH);  
  
}
```

```
void stop(){
```

```
digitalWrite(motorR1, LOW);  
digitalWrite(motorR2, LOW);  
digitalWrite(motorL1, LOW);  
digitalWrite(motorL2, LOW);  
}
```