



**ROBOSTEM** – Μια Εργαλειοθήκη για την  
προώθηση της χρήσης  
Δεξιοτήτων STEM αξιοποιώντας  
Εφαρμογές Μικροελεγκτών  
2019-1-RO01-KA202-063965



Co-funded by the  
Erasmus+ Programme  
of the European Union

# ΚΑΤΑΣΚΕΥΗ ΡΟΜΠΟΤΙΚΟΥ ΒΡΑΧΙΟΝΑ ΜΕ ΑΙΣΘΗΤΗΡΑ ΘΕΡΜΟΚΡΑΣΙΑΣ



# Ρομποτικός Βραχίονας με Αισθητήρα Θερμοκρασίας

## Λίστα Εξοπλισμού:

Arduino Uno

Τροφοδοτικό 5V

Groove RGB LCD Οθόνη

Αντίσταση 220-ohm

Καλώδια Jumper

4x Μοτέρ

4x Περιστροφικό Ποτενσιόμετρο

Ψηφιακός αισθητήρας θερμοκρασίας DS18B20 1-καλωδίου

## Μεθοδολογία

Αυτό το έργο μπορεί να χωριστεί σε τρεις υποενότητες, όπου στη συνέχεια θα το συνδυάσουμε στο τέλος για να δείξουμε πώς μπορούν τα δομικά στοιχεία να αλληλεπιδράσουν. Για τους σκοπούς αυτής της εξήγησης, θα χρησιμοποιήσουμε ένα έργο ανοιχτού κώδικα για τον σχεδιασμό του ρομποτικού βραχίονα, που ονομάζεται uFactory, αλλά στην πραγματικότητα, μπορεί να χρησιμοποιηθεί οποιοδήποτε άλλο, καθώς αξιοποιούν την ίδια αρχή.

### 1. Υλικό για τον ρομποτικό βραχίονα

Το πιο κρίσιμο σημείο προκειμένου να πετύχει αυτό το έργο είναι η υλοποίηση ενός ρομποτικού βραχίονα. Για τους σκοπούς αυτού του έργου και για την απλούστευσή του, θα χρησιμοποιήσουμε ένα σχέδιο ανοιχτού κώδικα, το οποίο μπορεί να εκτυπωθεί σε 3D εκτυπωτή.

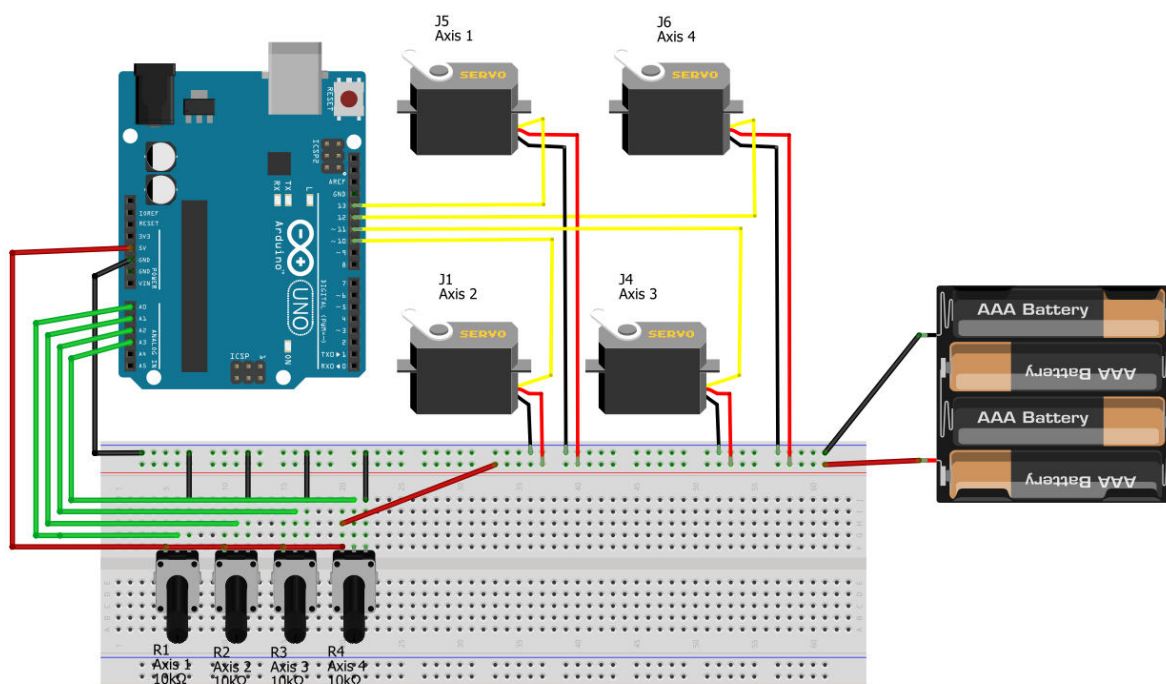
Παρακάτω είναι ένας σύνδεσμος προς ένα ανοιχτού κώδικα σχέδιο ρομποτικού βραχίονα. Με βάση αυτό μπορεί κανείς να καλύψει τη φυσική σχεδίαση του ρομποτικού βραχίονα.

UFactory: [https://www.ufactory.cc/?gclid=CjwKCAjwsJ3ZBRBJEiwAtuvtIAy4GE087c-t4VmAOgyVVIxL7j3ZOg9c7-LNj1x\\_IcBRGOIYuTCQ\\_BoCHFoQAvD\\_BwE#/en/uarmswift](https://www.ufactory.cc/?gclid=CjwKCAjwsJ3ZBRBJEiwAtuvtIAy4GE087c-t4VmAOgyVVIxL7j3ZOg9c7-LNj1x_IcBRGOIYuTCQ_BoCHFoQAvD_BwE#/en/uarmswift)



## 2. Σύνδεση Μοτέρ

Τα μοτέρ πρόκειται για εξαρτήματα που καταναλώνουν αρκετή ενέργεια και ανάλογα με το μοτέρ που χρησιμοποιείται, η κατανάλωση ενέργειας είναι διαφορετική για το καθένα. Επομένως, χρειάζεται προσοχή να γνωρίζετε ποιο μοτέρ να χρησιμοποιήσετε. Η καλύτερη μέθοδος για να αποφύγετε προβλήματα είναι να χρησιμοποιήσετε μια ξεχωριστή τροφοδοσία ρεύματος 5-6V DC που είναι καλή (ως υπόδειξη, επειδή είναι προβληματική η εύρεση ή αρκετά ακριβή, μπορείτε να χρησιμοποιήσετε την τροφοδοσία από έναν υπολογιστή, καθώς τα περισσότερα είναι αρκετά ισχυρά για να τροφοδοτήσετε πολλά μοτέρ ταυτόχρονα). Το σχήμα 1 δείχνει τη διαμόρφωση για τον τρόπο σύνδεσης.



Σχήμα 1 – Μοτέρ και Ποτενσιόμετρα

Μετά τη διαμόρφωση του παραπάνω υλικού, τα μοτέρ ενεργοποιούνται, αλλά το Arduino πρέπει ακόμα να ρυθμιστεί με το λογισμικό. Για να ελέγξετε τη διαμόρφωση, φορτώστε το αρχείο "Servo Controllers Only".



Ακολουθεί μια εξήγηση του τι κάνει ο κώδικας με τα σχόλια που προστέθηκαν εξηγώντας τι κάνει κάθε μέρος.

```
#include <Servo.h> //Library for Servo

Servo Axis1; //Define a Servo
Servo Axis2;
Servo Axis3;
Servo Axis4;

//Define Register for Integer. This is used to contain the value of the potentiometers
int Axis1Pot;
int Axis1val;
int Axis2Pot;
int Axis2val;
int Axis3Pot;
int Axis3val;
int Axis4Pot;
int Axis4val;

void setup() {
  Axis1.attach(8); //Define which servo is attached to which pin upon the Arduino Board
  Axis2.attach(9);
  Axis3.attach(10);
  Axis4.attach(11);
}

void loop() {
  Axis1val = analogRead(A0); //Read values from the Arduino Pin
  Axis2val = analogRead(A1);
  Axis3val = analogRead(A2);
  Axis4val = analogRead(A3);

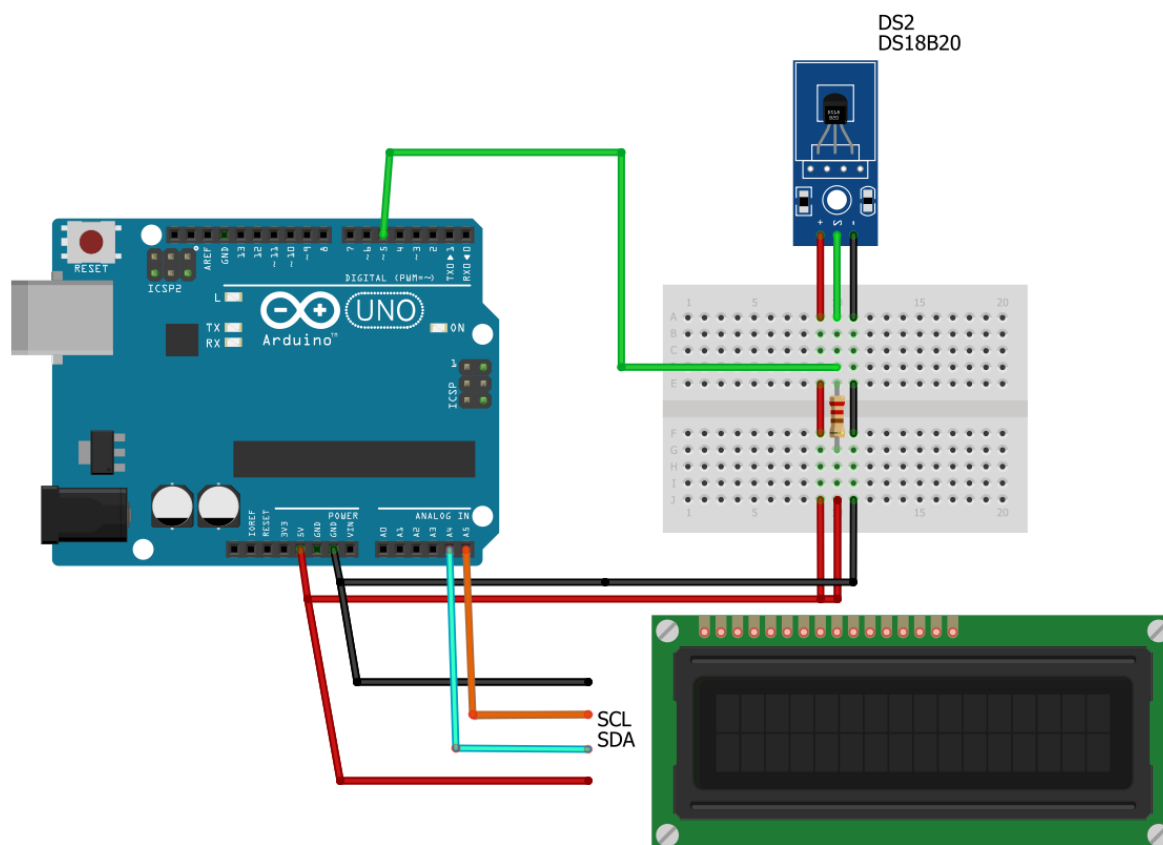
  Axis1Pot = map(Axis1val,0,1023,0,180); //Map values from "0to1024" to "0to180"
  Axis2Pot = map(Axis2val,0,1023,15,165);
  Axis3Pot = map(Axis3val,0,1023,0,180);
  Axis4Pot = map(Axis4val,0,1023,0,180);

  Axis1.write(Axis1Pot); //Output value upon Servos
  Axis2.write(Axis2Pot);
  Axis3.write(Axis3Pot);
  Axis4.write(Axis4Pot);
}
```



### 3. Σύνδεση της οθόνης LCD και του Αισθητήρα Θερμοκρασίας

Ο αισθητήρας θερμοκρασίας χρησιμοποιείται για να δώσει ανάδραση σε πραγματικό χρόνο για το υπό εξέταση μέσο. Σε αυτό το σενάριο θα χρησιμοποιήσουμε έναν αισθητήρα θερμοκρασίας DS18B20. Για αυτόν τον αισθητήρα, θα χρησιμοποιήσουμε μια αντίσταση έλξης (αντίσταση που συνδέεται παράλληλα με τη γραμμή του αισθητήρα στο VDD, για να σταθεροποιηθεί το σήμα).



Σχήμα 2: Οθόνη LCD & Αισθητήρας Θερμοκρασίας



Δεδομένου ότι η διαμόρφωση είναι διαφορετική, θα χρησιμοποιήσουμε τον ακόλουθο κώδικα για να την κάνουμε λειτουργική.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include "rgb_lcd.h"

#define ONE_WIRE_BUS 5
//define that temperature sensor will use pinout 5

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

float Celcius=0;
float Fahrenheit=0;

rgb_lcd lcd;

const int colorR = 255;
const int colorG = 0;
const int colorB = 0;

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

  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);

  lcd.setRGB(colorR, colorG, colorB);

  delay(1000);
}

void loop(void)
{
  sensors.requestTemperatures();
  Celcius=sensors.getTempCByIndex(0);
  Fahrenheit=sensors.toFahrenheit(Celcius);
  Serial.print(" C ");
  Serial.print(Celcius);
  Serial.print(" F ");
  Serial.println(Fahrenheit);

  // set the cursor to column 0, line 0
  lcd.setCursor(0, 0);
  lcd.print(" C: ");
  lcd.print(Celcius);

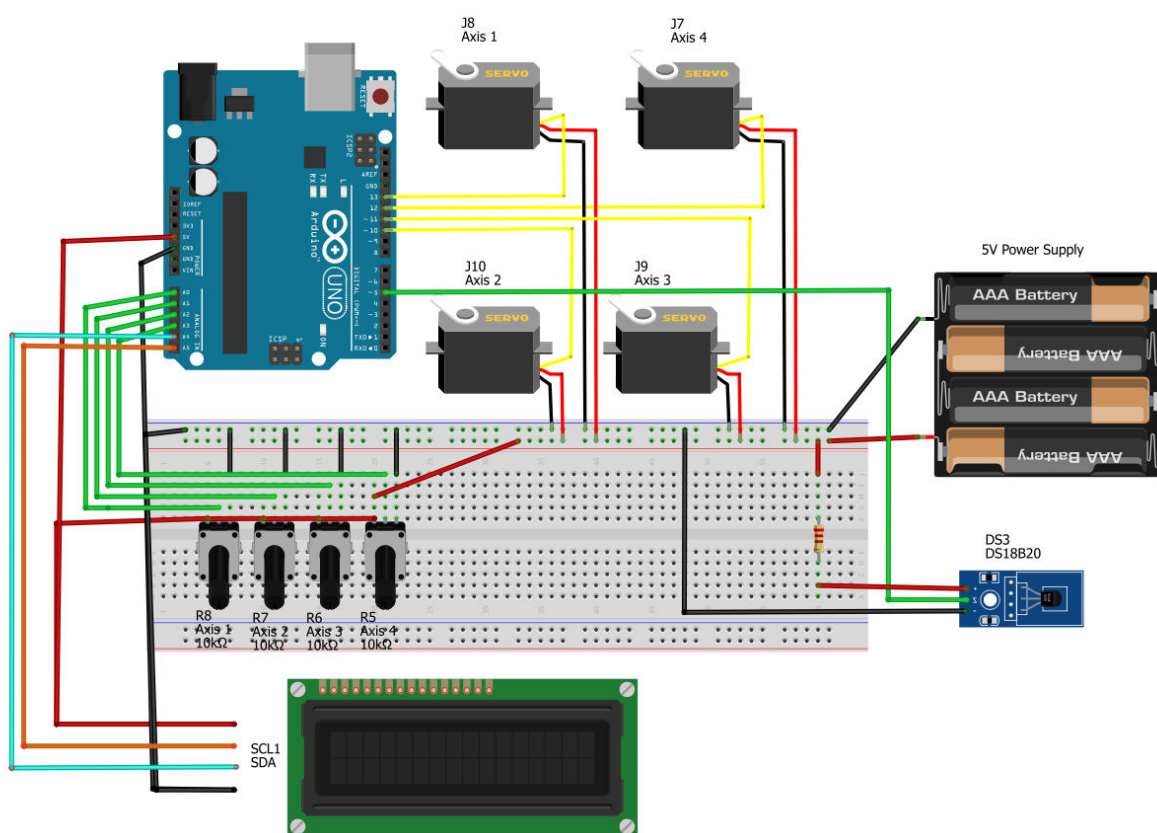
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  lcd.print(" F: ");
  lcd.print(Fahrenheit);

  delay(100);
}
```



#### 4. Τοποθετώντας όλα τα υλικά μαζί

Οι παραπάνω ενότητες παρουσιάστηκαν ξεχωριστά, ώστε να είναι ευκολότερος ο εντοπισμός σφαλμάτων, καθώς πρόκειται για αρκετά πολύπλοκα κυκλώματα και είναι βέβαιο ότι θα συμβούν λάθη. Οι παραπάνω ενότητες χρησιμοποιούν την ίδια διαμόρφωση όπως φαίνεται στην παρακάτω εικόνα. Η παρακάτω εικόνα δείχνει απλώς το πλήρες συνδυαστικό κύκλωμα.



Ο κώδικας είναι επίσης ίδιος με τις άλλες ενότητες, αλλά πρέπει να συνδυαστεί σε έναν συνολικό. Ακολουθεί ο κώδικας που πρέπει να χρησιμοποιηθεί, βρίσκεται επίσης στο αρχείο «Μελέτη Περίπτωσης 1».





```
#include <Servo.h> //Library for Servo
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include "rgb_lcd.h"

#define ONE_WIRE_BUS 5

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

float Celcius=0;
float Fahrenheit=0;

rgb_lcd lcd;

const int colorR = 138;
const int colorG = 43;
const int colorB = 226;

Servo Axis1; //Define a Servo
Servo Axis2;
Servo Axis3;
Servo Axis4;

int Axis1Pot; //Define Register for Integer
int Axis1val;
int Axis2Pot;
int Axis2val;
int Axis3Pot;
int Axis3val;
int Axis4Pot;
int Axis4val;

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

  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);

  lcd.setRGB(colorR, colorG, colorB);

  Axis1.attach(8); //Define which servo is attached to which pin upon the Arduino Board
  Axis2.attach(9);
  Axis3.attach(10);
  Axis4.attach(11);
}
```



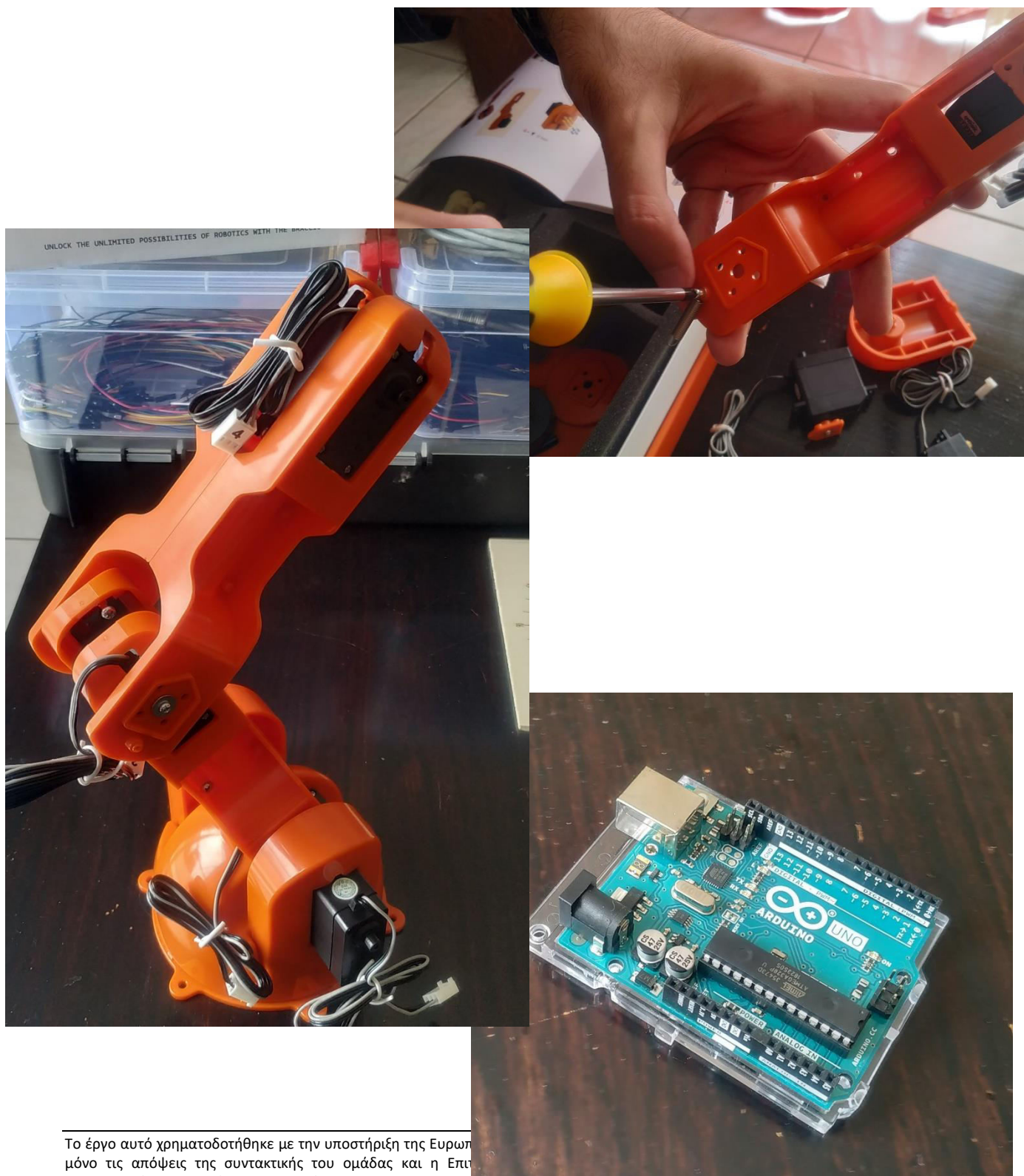


```
void loop() {  
  sensors.requestTemperatures();  
  Celcius=sensors.getTempCByIndex(0);  
  Fahrenheit=sensors.toFahrenheit(Celcius);  
  Serial.print(" C ");  
  Serial.print(Celcius);  
  Serial.print(" F ");  
  Serial.println(Fahrenheit);  
  
  // set the cursor to column 0, line 0  
  lcd.setCursor(0, 0);  
  lcd.print(" C: ");  
  lcd.print(Celcius);  
  
  // (note: line 1 is the second row, since counting begins with 0):  
  lcd.setCursor(0, 1);  
  lcd.print(" F: ");  
  lcd.print(Fahrenheit);  
  
  Axis1val = analogRead(A0); //Read values from the Arduino Pin  
  Axis2val = analogRead(A1);  
  Axis3val = analogRead(A2);  
  Axis4val = analogRead(A3);  
  
  Axis1Pot = map(Axis1val,0,1023,0,180); //Map values from "0to1024" to "0to180"  
  Axis2Pot = map(Axis2val,0,1023,15,165);  
  Axis3Pot = map(Axis3val,0,1023,0,180);  
  Axis4Pot = map(Axis4val,0,1023,0,180);  
  
  Axis1.write(Axis1Pot); //Output value upon Servos  
  Axis2.write(Axis2Pot);  
  Axis3.write(Axis3Pot);  
  Axis4.write(Axis4Pot);  
}
```



## 5. Ολοκλήρωση Πρότζεκτ

Παρακάτω είναι μερικές εικόνες που δείχνουν πώς λειτουργεί το έργο αλλά και τη σχετική εργασία που χρειάστηκε.



Το έργο αυτό χρηματοδοτήθηκε με την υποστήριξη της Ευρωπαϊκής Ένωσης, μόνο τις απόψεις της συντακτικής του ομάδας και η Ευρωπαϊκή Ένωση δεν φέρει ευθύνη για τις πληροφορίες που περιέχονται σε αυτό.



Η παρακάτω εικόνα δείχνει τον ρομποτικό βραχίονα να λειτουργεί. Ο ρομποτικός βραχίονας κρατά έναν αισθητήρα θερμοκρασίας και τον κατεβάζει στο νερό όπου στη συνέχεια εμφανίζεται στην οθόνη LCD.

