



# Construirea unei maini robotice cu o sonda de temperatura



# Mana robotica cu o sonda de temperatura

## Lista de echipament:

Arduino Uno

5 V Power Supply

Groove RGB LCD Screen

220-ohm resistor

Jumper Wires

4 x Servo

4 x Rotary Potentiometer

DS18B20 1-Wire Digital Temperature Sensor

## Metodologie

Acest proiect poate fi separat în trei subsecțiuni, unde apoi îl vom combina la sfârșit pentru a arăta modul în care elementele pot lucra împreună. În scopul acestei explicații, vom folosi un proiect open-source pentru proiectarea brațului robotizat, numit uFactory, dar în realitate, oricare altul poate fi folosit deoarece folosesc același principiu.

### 1. Hardware pentru bratul robotizat

Partea cea mai critică pentru ca acest proiect să aibă succes este implementarea unui braț robot. În scopul acestui proiect, vom folosi un design open-source, care poate fi imprimat 3D pentru a simplifica lucrurile.

Mai jos este un link către o sursă deschisă Robotic Arm Designs. Acesta este pentru a acoperi designul fizic al brațului robotic.

UFactory: [https://www.ufactory.cc/?gclid=CjwKCAjwsJ3ZBRBJEiwAtuvtIAy4GE087c-t4VmAOyyVVIXL7j3ZOg9c7-LNJ1x\\_lcBRGOIYuTCQ\\_BoCHFoQAvD\\_BwE#/en/uarmswift](https://www.ufactory.cc/?gclid=CjwKCAjwsJ3ZBRBJEiwAtuvtIAy4GE087c-t4VmAOyyVVIXL7j3ZOg9c7-LNJ1x_lcBRGOIYuTCQ_BoCHFoQAvD_BwE#/en/uarmswift)



## 2. Conectarea unui Servo

Servo-le sunt componente destul de consumatoare de energie și, în funcție de servo folosit, consumul de energie este diferit pentru fiecare, așa că în funcție de cel utilizat, trebuie să aveți grijă pentru a ști pe care să îl folosiți. Cea mai bună metodă de a evita problemele este să folosiți o sursă de alimentare separată 5-6V DC este bună (ca indiciu, deoarece acestea sunt problematice de găsit sau sunt destul de scumpe, puteți utiliza sursa de alimentare de la un computer, deoarece majoritatea sunt suficient de puternice pentru alimentează mai multe servo-uri în același timp). Figura 1 arată configurația cu privire la modul de conectare.

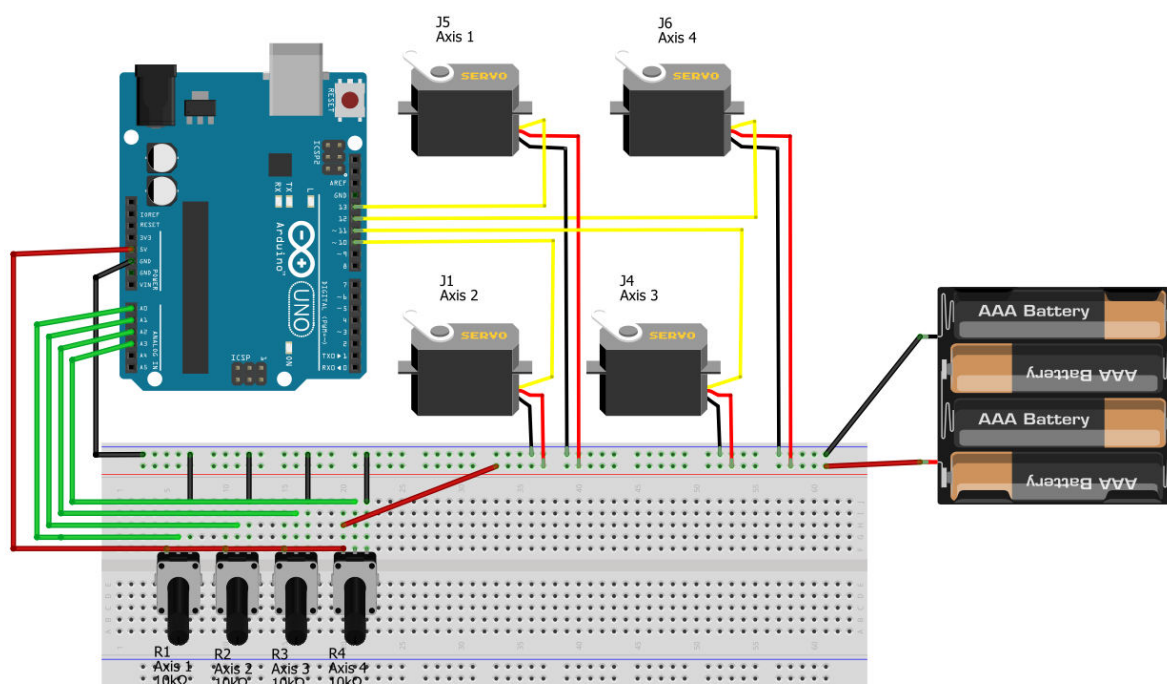


Figure 1 - Servo and Potentiometers

După configurarea hardware-ului de mai sus, servo-urile sunt pornite, dar Arduino trebuie încă configurat cu software-ul. Pentru a verifica configurația, încărcați fișierul „Servo Controllers Only”.



Mai jos este un comentariu care explica codul:

```
#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. Conectarea LCD-ului si senzorului de temperatura

Senzorul de temperatură este utilizat pentru a oferi feedback în timp real despre mediul care este sondat. În acest scenariu vom folosi un senzor de temperatură DS18B20. Pentru acest senzor, vom folosi un rezistor pull-up (rezistor care este conectat în paralel cu linia senzorului la VDD, acesta este pentru a stabili semnalul).

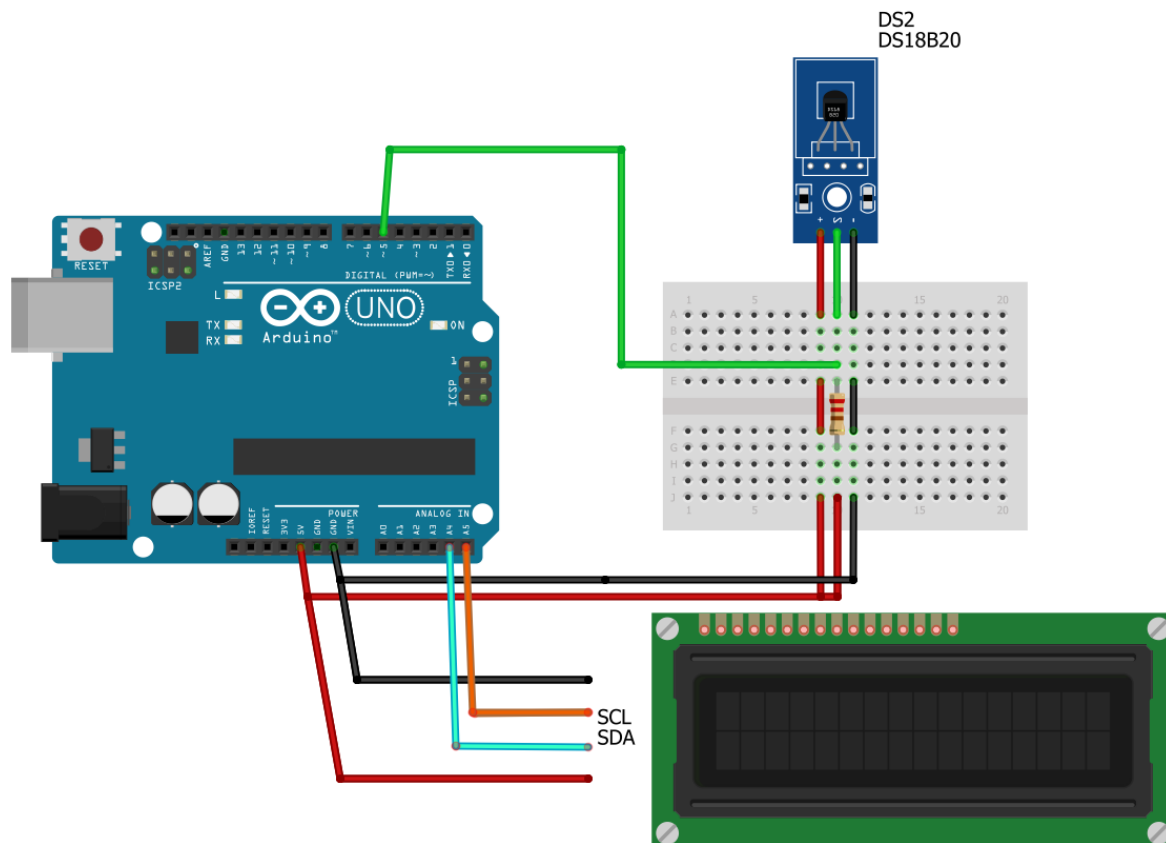


Figure 2 LCD & Temperature Sensor

Deoarece configuratia este diferita, vom folosi urmatorul cod:

```
#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);

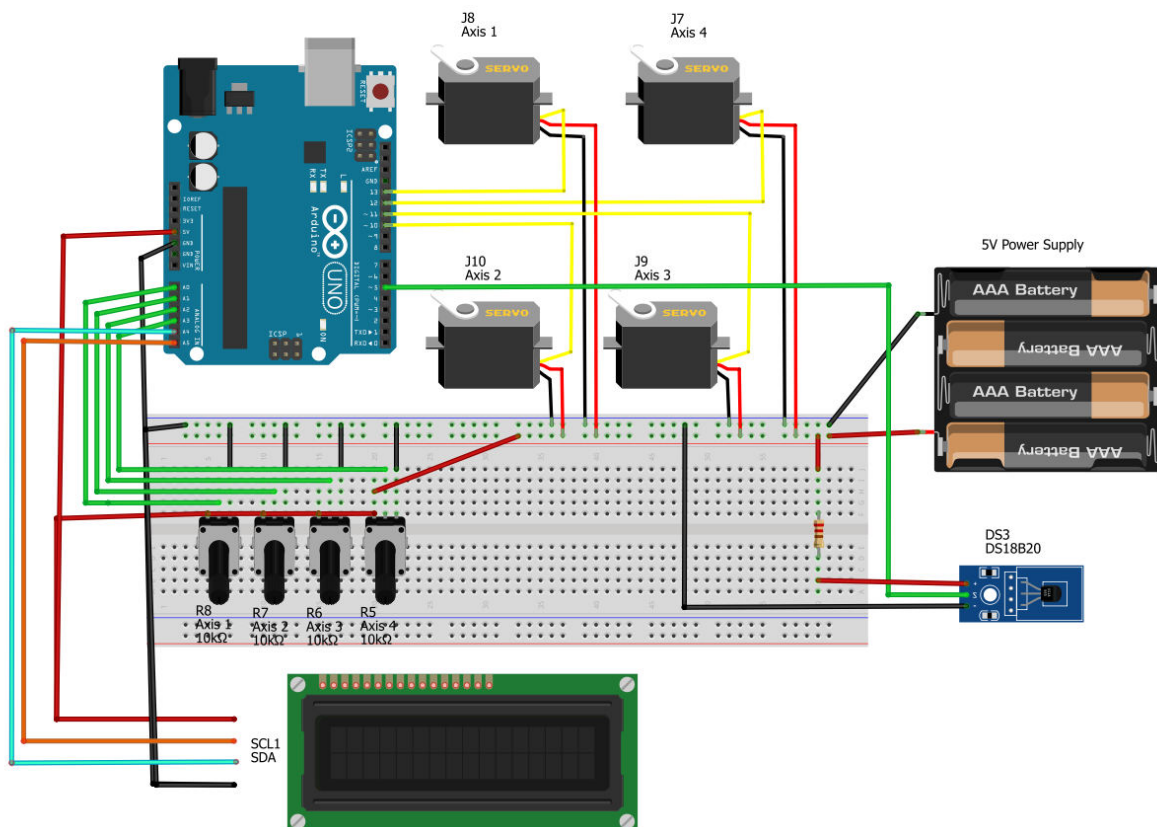
  delay(100);
}
```

---



#### 4. Punand totul cap la cap

Secțiunile de mai sus au fost afișate separat, astfel încât să fie mai ușor pentru găsirea defectăunilor, deoarece acestea sunt circuite destul de complexe și se vor întâmpla greșeli. Secțiunile de mai sus folosesc aceeași configurație ca cea prezentată în imaginea de mai jos. Imaginea de mai jos este arătată doar pentru a arăta că poate fi combinată într-un singur circuit.



Codul este, de asemenea, același cu celelalte secțiuni, dar trebuie combinat într-una singură. Următorul este codul care trebuie utilizat, acesta se află și în fișierul „Studiu de caz 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);  
}
```



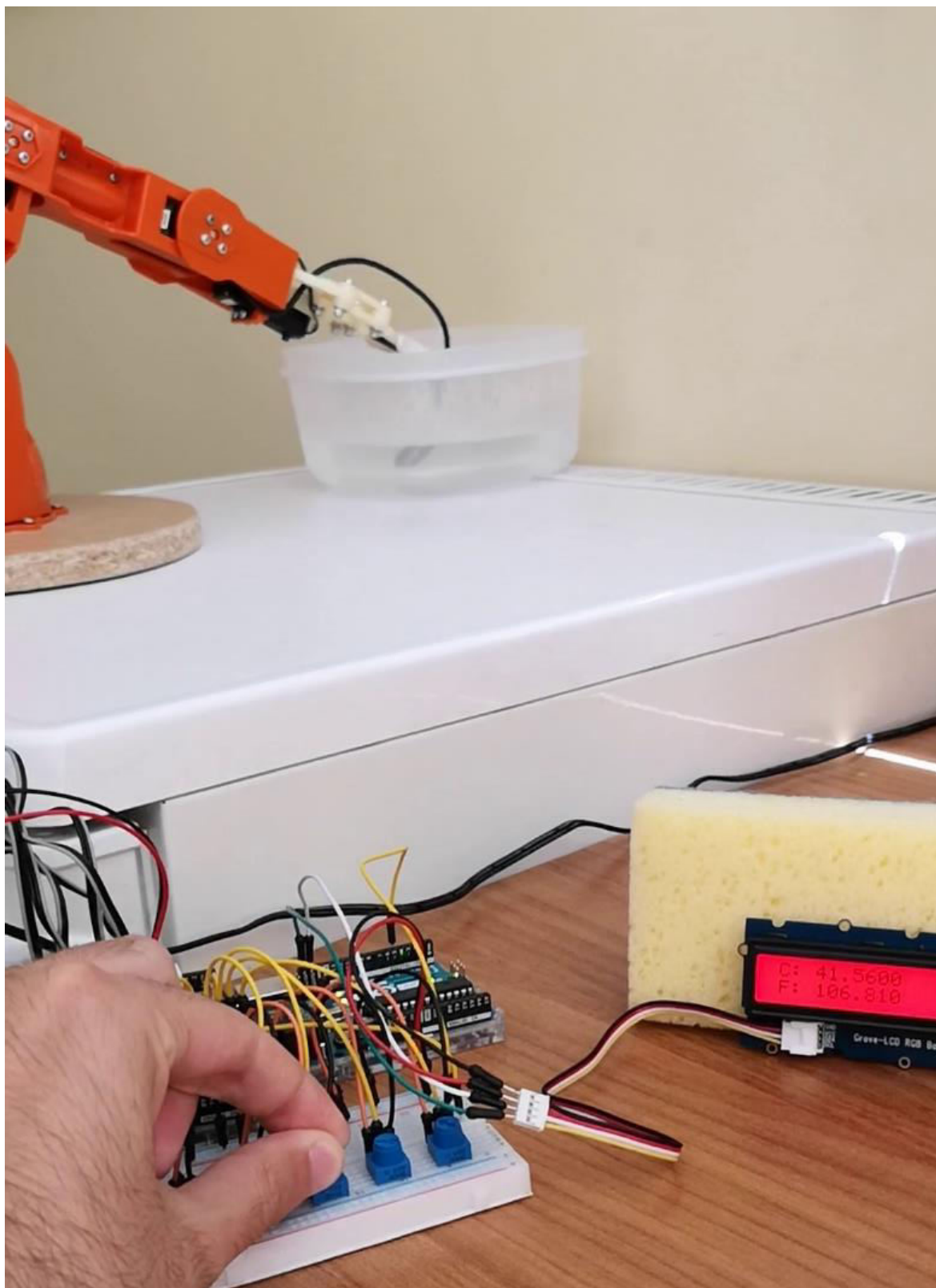
## 5. Finalul proiectarii

Mai jos sunt imagini cu proiectul terminat si diferite modalitati in care acesta functioneaza:





Imaginea de mai jos arată brațul robotizat în funcțiune. Brațul robotului ține o sondă de temperatură și o coboară în apă, unde este apoi afișată pe LCD





**ROBOSTEM** – A Trainers Toolkit To  
Foster STEM Skills Using Microcontroller  
Applications  
2019-1-RO01-KA202-063965



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