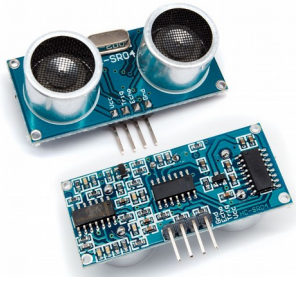
Starter Kit for Arduino Uno – Manual Vol.2

Exemple
Senzori

Exemplul 8 Folosirea senzorului de distanță cu ultrasunete HC-SR04



Piese necesare :

1 x placă Arduino UNO R3
1 x Senzor de distanţă cu ultrasunete HC-SR04
3 x LED (plus 3 rezistenţe de 1 kΩ pentru aceste leduri)
cabluri de legătură necesare între piese şi placă

Algoritm :

Iniţial ledurile sunt oprite.
La distanţa cea mai mare (eu am specificat în cod peste valoarea 3000) toate ledurile sunt de asemenea oprite.

Dacă se detectează un obstacol între valorile 2001 şi 3000 se aprinde 1 led (aproximativ între 30-40 cm distanţă).
Dacă obiectul se apropie între valorile 1000 si 2000 am setat să se aprindă 2 leduri.
Dacă obiectul e foarte aproape de senzor (sub 1000 respectiv sub 10-15 cm) am setat să se aprindă toate 3 ledurile.

Montare :

Senzorul are 4 pini respectiv:

VCC de la senzor se cupleaza la +5V de pe placa arduino.
TRIG de la senzor se cuplează la unul din pinii digitali de pe placa arduino (în cod am setat pinul 2)
ECHO de la senzor se cuplează la un alt pin digital de pe placa arduino (în cod am setat pinul 3)
GND de la senzor se cuplează la unul din pinii GND de la arduino

LED-urile se cuplează şi ele (prin intermediul unei rezistenţe de 1 kΩ) la pinii 8,9 şi 10 iar negativul de la leduri la GND de la arduino.

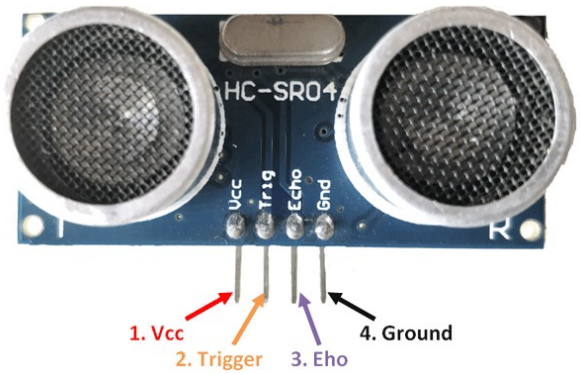
Pini :
* VCC la 5V
* TRIG la Digital pin 2
* ECHO la Digital pin 3
* GND la GND
* LED1 la Digital pin 8
* LED2 la Digital pin 9
* LED3 la Digital pin 10

Vom scrie un cod :

```
/*
* Ultrasound model: HC-SR04
* Senzor pins  | Arduino Board
* VCC la 5V
* TRIG la Digital pin 2
* ECHO la Digital pin 3
* GND la GND
* LED1 la Digital pin 8
* LED2 la Digital pin 9
* LED3 la Digital pin 10
*/

// initialize the necessary ports
int usTrigger = 2;
int usEcho = 3;
int led1 = 8;
int led2 = 9;
int led3 = 10;

void setup() {
// start ultrasound
pinMode(usTrigger, OUTPUT);
pinMode(usEcho, INPUT);
// start leds
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
}

void loop() {
// Initiate ultrasonic speaker
digitalWrite(usTrigger, HIGH);
delayMicroseconds(10);
digitalWrite(usTrigger, LOW);

// Reading out the signal strength
long timecount = pulseIn(usEcho, HIGH);
// if more than 3000 all LEDs stop.
if ( timecount > 3000 ) {
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
}
// If it’s between 2001 and 3000 start LED 1
if ( timecount > 2000 && timecount <= 3000 ) {
digitalWrite(led1, HIGH);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
}
// If it’s between 1000 and 2000 start LED 1 and 2
if ( timecount >= 1000 && timecount <= 2000 ) {
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
```

```
digitalWrite(led3, LOW);
}
// If it's under 1000 turn all LEDs
if ( timecount < 1000 ) {
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
}
// Wait 0.1 seconds before the next reading.
delay(100);
}
```

Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi", pe care trebuie să le corectați.



Exemplul 9 Folosirea senzorului de distanță cu ultrasunete HC-SR04 cu librăria „NewPing"

Parte din acest cod va apare în codul roboților cu 2 sau 4 roți cu ocolire obstacole

Piese necesare :

1 x placă Arduino UNO R3
1 x Senzor de distanţă cu ultrasunete HC-SR04
cabluri de legătură necesare între piese şi placă

Algoritm :

Distanța va fi afisată în „cm" pe Serial-Monitor.

Pini :
* VCC la 5V
* TRIG la Digital pin 2
* ECHO la Digital pin 3
* GND la GND

Atenție trebuie instalată librăria NewPing de aici
https://github.com/eliteio/Arduino_New_Ping

Vom scrie un cod :

```
#include <NewPing.h>

#define TRIGGER_PIN  2
#define ECHO_PIN     3
#define MAX_DISTANCE 400

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

unsigned int pingSpeed = 50; // 50ms
unsigned long pingTimer;

void echoCheck() {
  if (sonar.check_timer()) {
    Serial.print("Ping: ");
    Serial.print(sonar.ping_result / US_ROUNDTRIP_CM);
    Serial.println("cm");
  }
}

void setup() {
  Serial.begin(9600);
  pingTimer = millis(); // Start now.
}

void loop() {
  if (millis() >= pingTimer) {
    pingTimer += pingSpeed;
    sonar.ping_timer(echoCheck);
  }
}
```

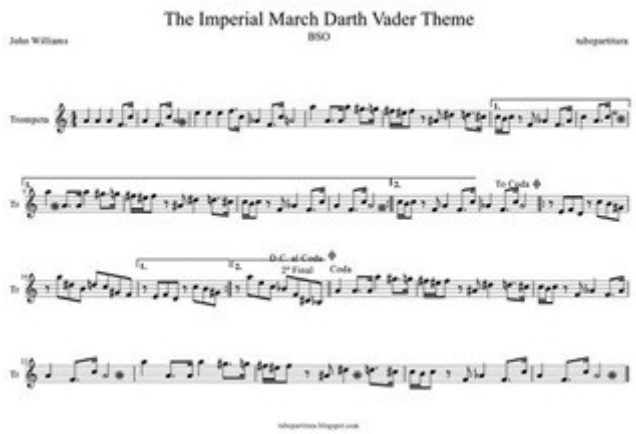
Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Pe Serial-Monitor apare: „Ping: xxx cm”

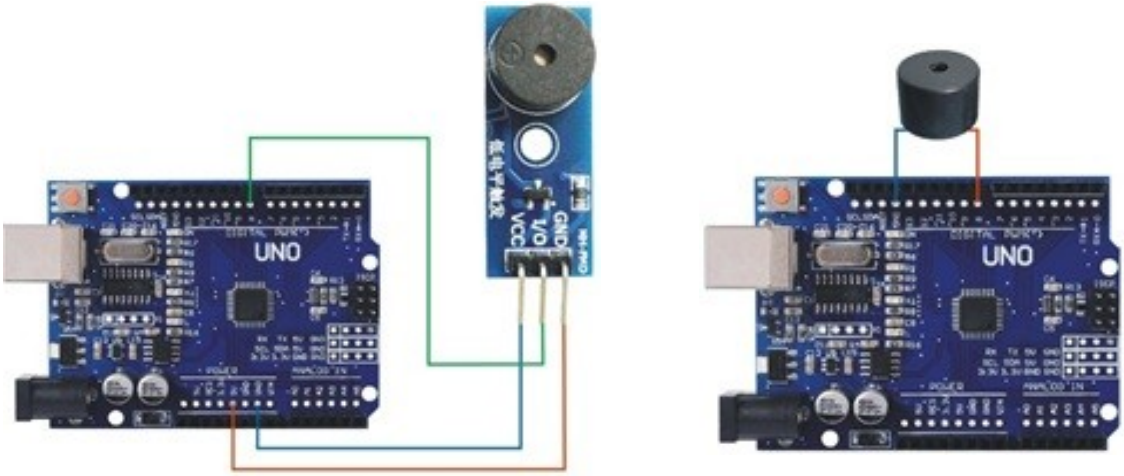
Exemplul 10 Să ne destindem puțin

Cum să facem un UNO să cânte !

The imperial March – STAR WARS



Avem nevoie de un UNO și de un buzzer pasiv.



Vom scrie un cod :

Atenție mare sa nu omiteti ceva

```
//The imperial March – STAR WARS
// tone(8,LA3,Q)  delay(1+Q)
// toate notele definite in englezăcu exceptia,
//asta care se numeste in italiană “la” pentru că A0…A5 sunt pini “analog”.
// (Ab este definit  Ab, iar inversul(NOT) este LAb)
#define  C0 16.35
#define Db0 17.32
#define D0  18.35
```

```
#define Eb0 19.45
#define E0  20.60
#define F0  21.83
#define Gb0 23.12
#define G0  24.50
#define Ab0 25.96
#define LA0 27.50
#define Bb0 29.14
#define B0  30.87
#define C1  32.70
#define Db1 34.65
#define D1  36.71
#define Eb1 38.89
#define E1  41.20
#define F1  43.65
#define Gb1 46.25
#define G1  49.00
#define Ab1 51.91
#define LA1 55.00
#define Bb1 58.27
#define B1  61.74
#define C2  65.41
#define Db2 69.30
#define D2  73.42
#define Eb2 77.78
#define E2  82.41
#define F2  87.31
#define Gb2 92.50
#define G2  98.00
#define Ab2 103.83
#define LA2 110.00
#define Bb2 116.54
#define B2  123.47
#define C3  130.81
#define Db3 138.59
#define D3  146.83
#define Eb3 155.56
```

```
#define E3  164.81
#define F3  174.61
#define Gb3 185.00
#define G3  196.00
#define Ab3 207.65
#define LA3 220.00
#define Bb3 233.08
#define B3  246.94
#define C4  261.63
#define Db4 277.18
#define D4  293.66
#define Eb4 311.13
#define E4  329.63
#define F4  349.23
#define Gb4 369.99
#define G4  392.00
#define Ab4 415.30
#define LA4 440.00
#define Bb4 466.16
#define B4  493.88
#define C5  523.25
#define Db5 554.37
#define D5  587.33
#define Eb5 622.25
#define E5  659.26
#define F5  698.46
#define Gb5 739.99
#define G5  783.99
#define Ab5 830.61
#define LA5 880.00
#define Bb5 932.33
#define B5  987.77
#define C6  1046.50
#define Db6 1108.73
#define D6  1174.66
#define Eb6 1244.51
#define E6  1318.51
```

```
#define F6  1396.91
#define Gb6 1479.98
#define G6  1567.98
#define Ab6 1661.22
#define LA6 1760.00
#define Bb6 1864.66
#define B6  1975.53
#define C7  2093.00
#define Db7 2217.46
#define D7  2349.32
#define Eb7 2489.02
#define E7  2637.02
#define F7  2793.83
#define Gb7 2959.96
#define G7  3135.96
#define Ab7 3322.44
#define LA7 3520.01
#define Bb7 3729.31
#define B7  3951.07
#define C8  4186.01
#define Db8 4434.92
#define D8  4698.64
#define Eb8 4978.03
// DURATION OF THE NOTES
#define BPM 120    //  you can change this value changing all the others
#define H 2*Q //half 2/4
#define Q 60000/BPM //quarter 1/4
#define E Q/2   //eighth 1/8
#define S Q/4 // sixteenth 1/16
#define W 4*Q // whole 4/4
void setup() {
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
digitalWrite(9,LOW);
}
// the loop routine runs over and over again forever:
void loop() {
```

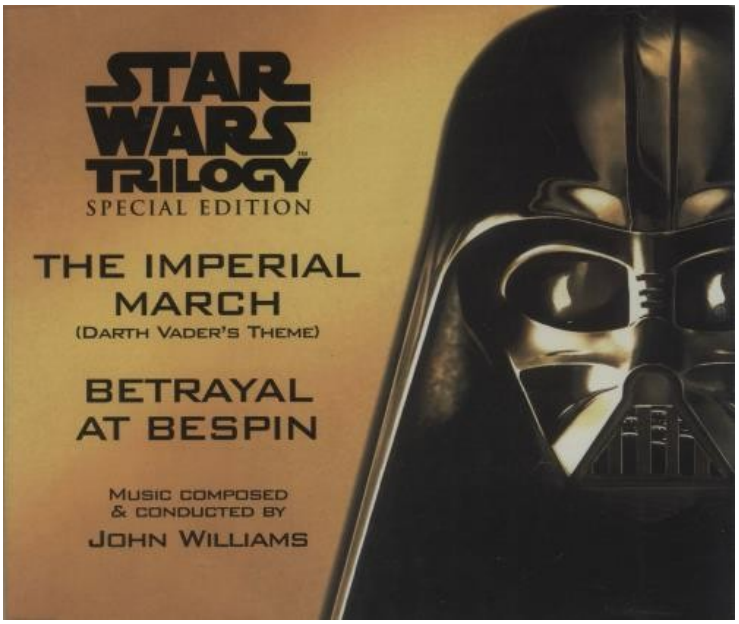
```
//tone(pin, note, duration)
tone(8,LA3,Q);
delay(1+Q); //delay duration should always be 1 ms more than the note in order to separate them.
tone(8,LA3,Q);
delay(1+Q);
tone(8,LA3,Q);
delay(1+Q);
tone(8,F3,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,LA3,Q);
delay(1+Q);
tone(8,F3,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,LA3,H);
delay(1+H);
tone(8,E4,Q);
delay(1+Q);
tone(8,E4,Q);
delay(1+Q);
tone(8,E4,Q);
delay(1+Q);
tone(8,F4,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,Ab3,Q);
delay(1+Q);
tone(8,F3,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,LA3,H);
```

```
delay(1+H);
tone(8,LA4,Q);
delay(1+Q);
tone(8,LA3,E+S);
delay(1+E+S);
tone(8,LA3,S);
delay(1+S);
tone(8,LA4,Q);
delay(1+Q);
tone(8,Ab4,E+S);
delay(1+E+S);
tone(8,G4,S);
delay(1+S);
tone(8,Gb4,S);
delay(1+S);
tone(8,E4,S);
delay(1+S);
tone(8,F4,E);
delay(1+E);
delay(1+E);//PAUSE
tone(8,Bb3,E);
delay(1+E);
tone(8,Eb4,Q);
delay(1+Q);
tone(8,D4,E+S);
delay(1+E+S);
tone(8,Db4,S);
delay(1+S);
tone(8,C4,S);
delay(1+S);
tone(8,B3,S);
delay(1+S);
tone(8,C4,E);
delay(1+E);
delay(1+E);//PAUSE QUASI FINE RIGA
tone(8,F3,E);
delay(1+E);
```

```
tone(8,Ab3,Q);
delay(1+Q);
tone(8,F3,E+S);
delay(1+E+S);
tone(8,LA3,S);
delay(1+S);
tone(8,C4,Q);
delay(1+Q);
tone(8,LA3,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,E4,H);
delay(1+H);
tone(8,LA4,Q);
delay(1+Q);
tone(8,LA3,E+S);
delay(1+E+S);
tone(8,LA3,S);
delay(1+S);
tone(8,LA4,Q);
delay(1+Q);
tone(8,Ab4,E+S);
delay(1+E+S);
tone(8,G4,S);
delay(1+S);
tone(8,Gb4,S);
delay(1+S);
tone(8,E4,S);
delay(1+S);
tone(8,F4,E);
delay(1+E);
delay(1+E);//PAUSE
tone(8,Bb3,E);
delay(1+E);
tone(8,Eb4,Q);
delay(1+Q);
```

```
tone(8,D4,E+S);
delay(1+E+S);
tone(8,Db4,S);
delay(1+S);
tone(8,C4,S);
delay(1+S);
tone(8,B3,S);
delay(1+S);
tone(8,C4,E);
delay(1+E);
delay(1+E);//PAUSE QUASI FINE RIGA
tone(8,F3,E);
delay(1+E);
tone(8,Ab3,Q);
delay(1+Q);
tone(8,F3,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,LA3,Q);
delay(1+Q);
tone(8,F3,E+S);
delay(1+E+S);
tone(8,C4,S);
delay(1+S);
tone(8,LA3,H);
delay(1+H);
delay(2*H);
}
```

Gata!

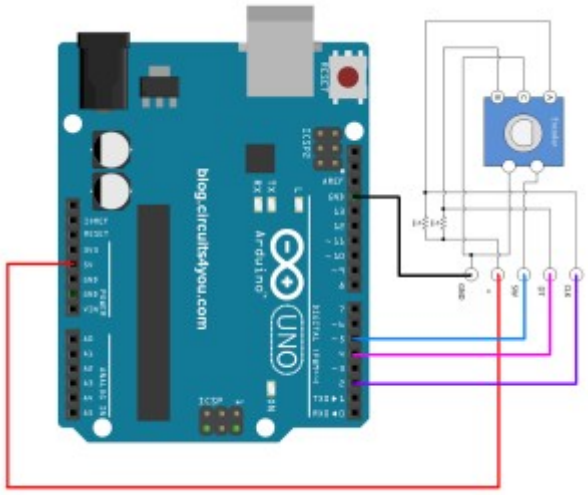


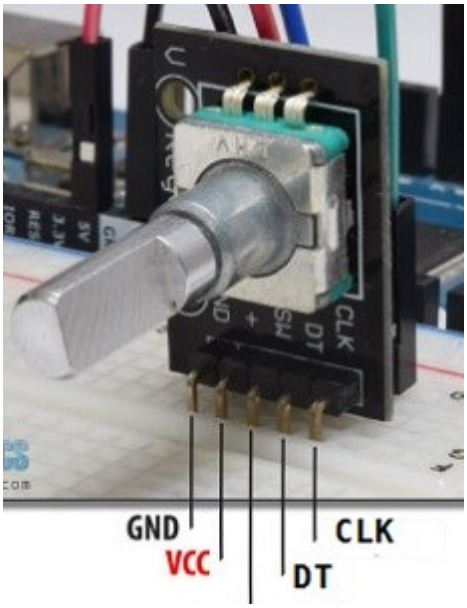
Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Exemplul 11 Folosirea senzorului Rotary Encoder

O schema simpla :

Afișează pe serial poziția encoderului





Piese necesare :

1 x placă Arduino UNO R3
1 x Modul encoder rotativ compatibil Arduino (143)
cabluri de legătură necesare între piese şi placă

Pini :
* VCC la 5V
* CLK la Digital pin 2
* DT la Digital pin 4
* SW la Digital pin 5 se comporta ca button
* GND la GND

Vom scrie un cod :

```
// Arduino Rotary Encoder Tutorial
int encoderCLK = 2;
int encoderDT = 4;

volatile int lastEncoded = 0;
volatile long encoderValue = 0;

long lastencoderValue = 0;

int lastMSB = 0;
int lastLSB = 0;

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

pinMode(encoderCLK, INPUT);
pinMode(encoderDT, INPUT);

digitalWrite(encoderCLK, HIGH); //turn pullup resistor on
digitalWrite(encoderDT, HIGH); //turn pullup resistor on

//call updateEncoder() when any high/low changed seen
```

```
//on interrupt 0 (pin 2), or interrupt 1 (pin 3)
attachInterrupt(0, updateEncoder, CHANGE);
attachInterrupt(1, updateEncoder, CHANGE);
}

void loop(){
Serial.println(encoderValue);
delay(10);
}

void updateEncoder(){
int MSB = digitalRead(encoderCLK); //MSB = most significant bit
int LSB = digitalRead(encoderDT); //LSB = least significant bit

int encoded = (MSB << 1) |LSB; //converting the 2 pin value to single number
int sum = (lastEncoded << 2) | encoded; //adding it to the previous encoded value

if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) encoderValue ++;
if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) encoderValue –;

lastEncoded = encoded; //store this value for next time
}

// Arduino Rotary Encoder Tutorial

Pentru butonul SW folosim exemplu clasic de la Arduino :

// Button
// set pin numbers:
const int buttonPin = 5; // the number of the pushbutton pin
const int ledPin = 13; // the number of the LED pin

// variables will change:
int buttonState = 0; // variable for reading the pushbutton status

void setup() {
// initialize the LED pin as an output:
pinMode(ledPin, OUTPUT);
// initialize the pushbutton pin as an input:
pinMode(buttonPin, INPUT);
}

void loop() {
// read the state of the pushbutton value:
buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.
// if it is, the buttonState is HIGH:
if (buttonState == HIGH) {
// turn LED on:
digitalWrite(ledPin, HIGH);
} else {
// turn LED off:
digitalWrite(ledPin, LOW);
}
}
// Button
```

Afiseaza pe Serial-Monitor poziția rotirii. Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Exemplul 12 Folosirea Display 16x2 LCD varianta simplă

Necesar la afisarea valorilor date de senzori

Librăria LiquidCrystal vă permite să controlați afișajele LCD care sunt compatibile cu driverul Hitachi HD44780. Puteți scrie de obicei prin interfața cu 16 pini.

Acest exemplu de cod imprimă „Hello World!” pe ecranul LCD și arată timpul în secunde de la resetarea Arduino.

LCD au o interfață paralelă, ceea ce înseamnă că microcontrolerul trebuie să manipuleze mai mulți pini de interfață simultan pentru a controla afișajul. Interfața constă din următorii pini :

Un pin de selectare a registrului (**RS**) care controlează locul în care scrieți datele din memoria LCD. Puteți selecta fie registrul de date, care conține ceea ce apare pe ecran, fie un registru de instrucțiuni, care este locul în care controlerul LCD caută instrucțiuni despre ce trebuie să faceți în continuare.

Un pin de citire / scriere (**R/W**) care selectează modul de citire sau modul de scriere

Un pin de activare care permite scrierea în registre 8 pini de date (**D0-D7**). Stările acestor pini (ridicat sau scăzut) sunt biții pe care îi scrieți într-un registru când scrieți sau valorile pe care le citiți când citiți.

Un pin de afișare constrast (**Vo**), pinii de alimentare (+ 5V)

Un pin (**Gnd**)

Pinii de iluminare cu fundal LED (**BkIt+** și **BKIt-**) pe care îi puteți utiliza pentru a alimenta LCD-ul, pentru a controla contrastul afișajului și pentru a porni și a opri LED-ul. lumină de fundal, respectiv. Procesul de control al afișajului implică introducerea datelor care formează imaginea a ceea ce doriți să afișați în registrele de date, apoi introducerea instrucțiunilor în registrul de instrucțiuni. Biblioteca LiquidCrystal simplifică acest lucru pentru dvs., astfel încât să nu aveți nevoie să cunoașteți instrucțiunile de nivel scăzut.

Ecranele LCD compatibile Hitachi pot fi controlate în două moduri: 4 biți sau 8 biți. Modul pe 4 biți necesită șapte pini I / O de la Arduino, în timp ce modul pe 8 biți necesită 11 pini. Pentru afișarea textului pe ecran, puteți face totul în modul pe 4 biți, astfel încât exemplul arată cum să controlați un LCD 16x2 în modul pe 4 biți.

Piese necesare :

1 x placă Arduino UNO R3

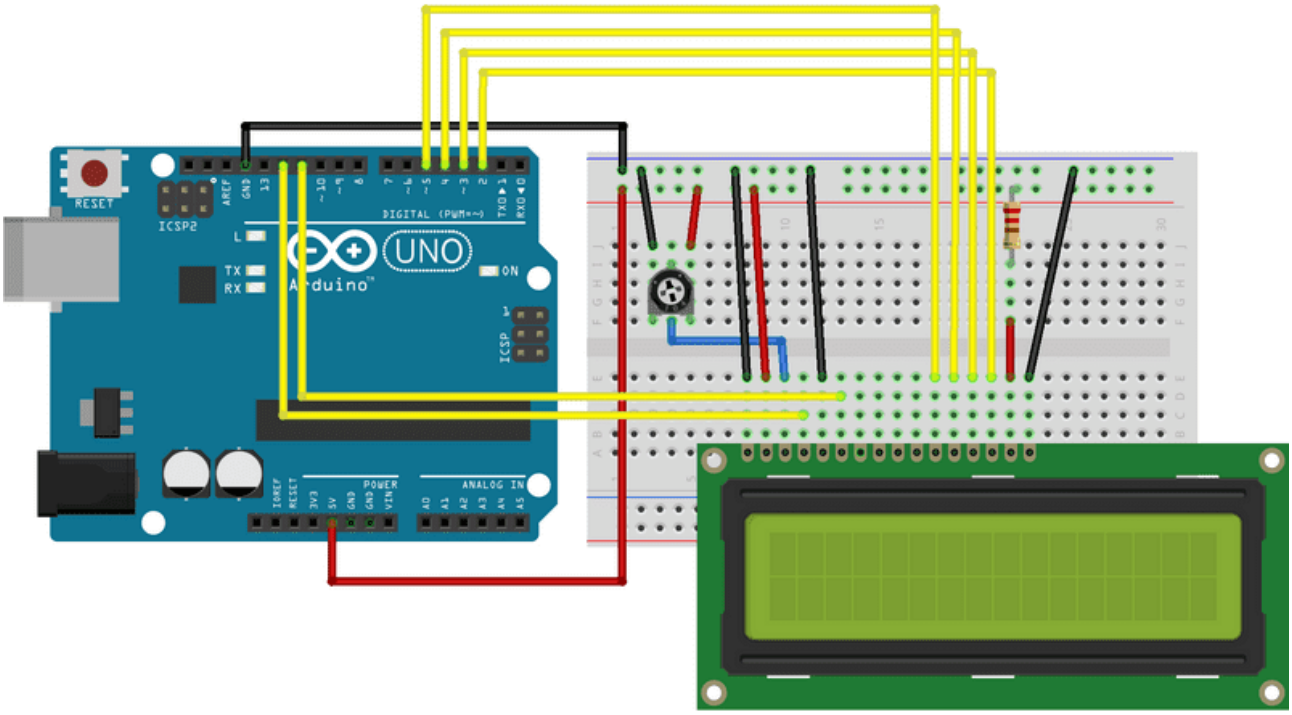
1 x LCD 2×16 caractere LED Blue Display

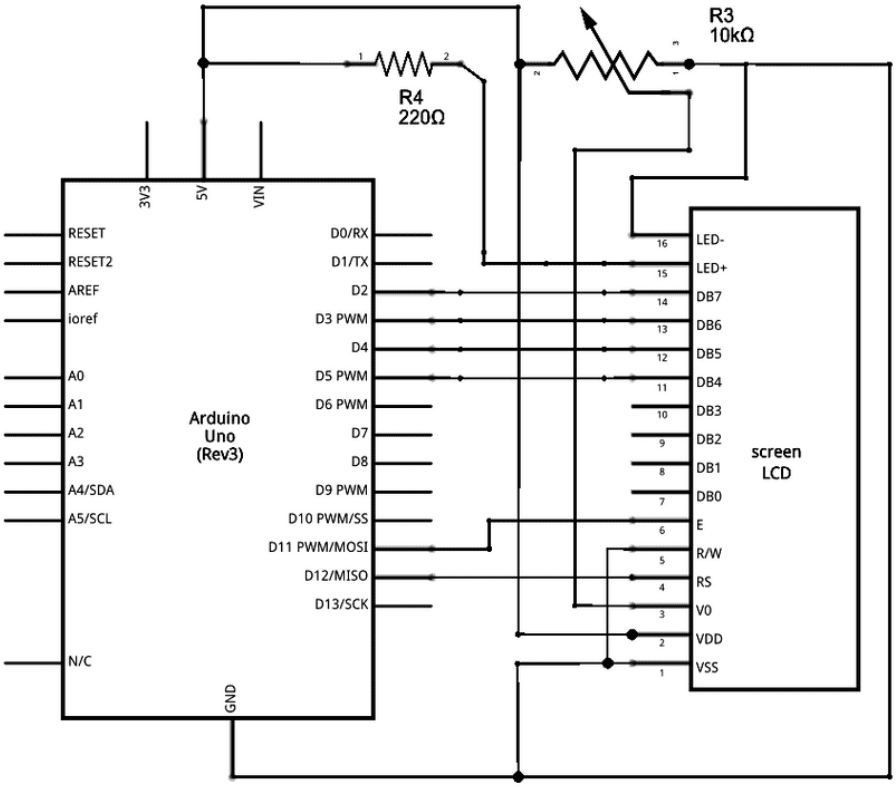
1 x Conector 40 pini tata in linie

1 x Potentiometru trimmer 10kΩ

1 X Rezistor 220Ω

cabluri de legătură necesare între piese şi placă





Înainte de a conecta ecranul LCD la placa Arduino, vă sugerăm să cositoriti conectorul de 16 pini pe display LCD

LCD RS pin la digital pin 12

LCD Enable pin la digital pin 11

LCD D4 pin la digital pin 5

LCD D5 pin la digital pin 4

LCD D6 pin la digital pin 3

LCD D7 pin la digital pin 2

LCD R/W pin la GND

LCD VSS pin la GND

LCD VCC pin la 5V

LCD LED+ la 5V prin 220 ohm resistor

LCD LED- la GND

În plus, conectați un potentiometru de 10k la + 5V și GND, cu pin mijloc la LCD VO (pin3).

Vom scrie un cod :

```
/*
  LiquidCrystal Library - Hello World
16x2 LCD display.
  The circuit:
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * LCD VSS pin to ground
 * LCD VCC pin to 5V
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)
*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
```

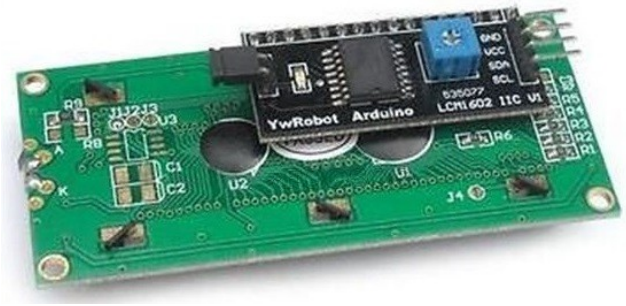
```
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

Afiseaza pe display "hello, world!". Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Exemplul 13 Folosirea Display 16x2 LCD varianta i2C

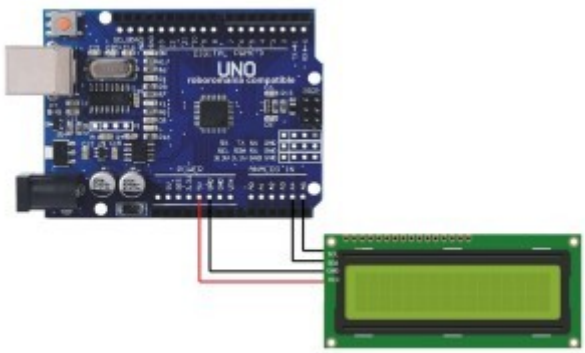
Necesar la afisarea valorilor date de senzori

Dacă vreți sa scăpați de așa multe fire



Piese necesare :

1 x placă Arduino UNO R3
1 x LCD 2x16 caractere LED Blue Display cu interfață i2C
cabluri de legătură necesare între piese şi placă (4 fire tata-mama)



Librăria LiquidCrystal_I2C vă permite să controlați afișajele LCD i2C

Pini :
* VCC la 5V
* SDA la SDA sau pin Analog 04
* SCL la SCL sau pin Analog 05
* GND la GND

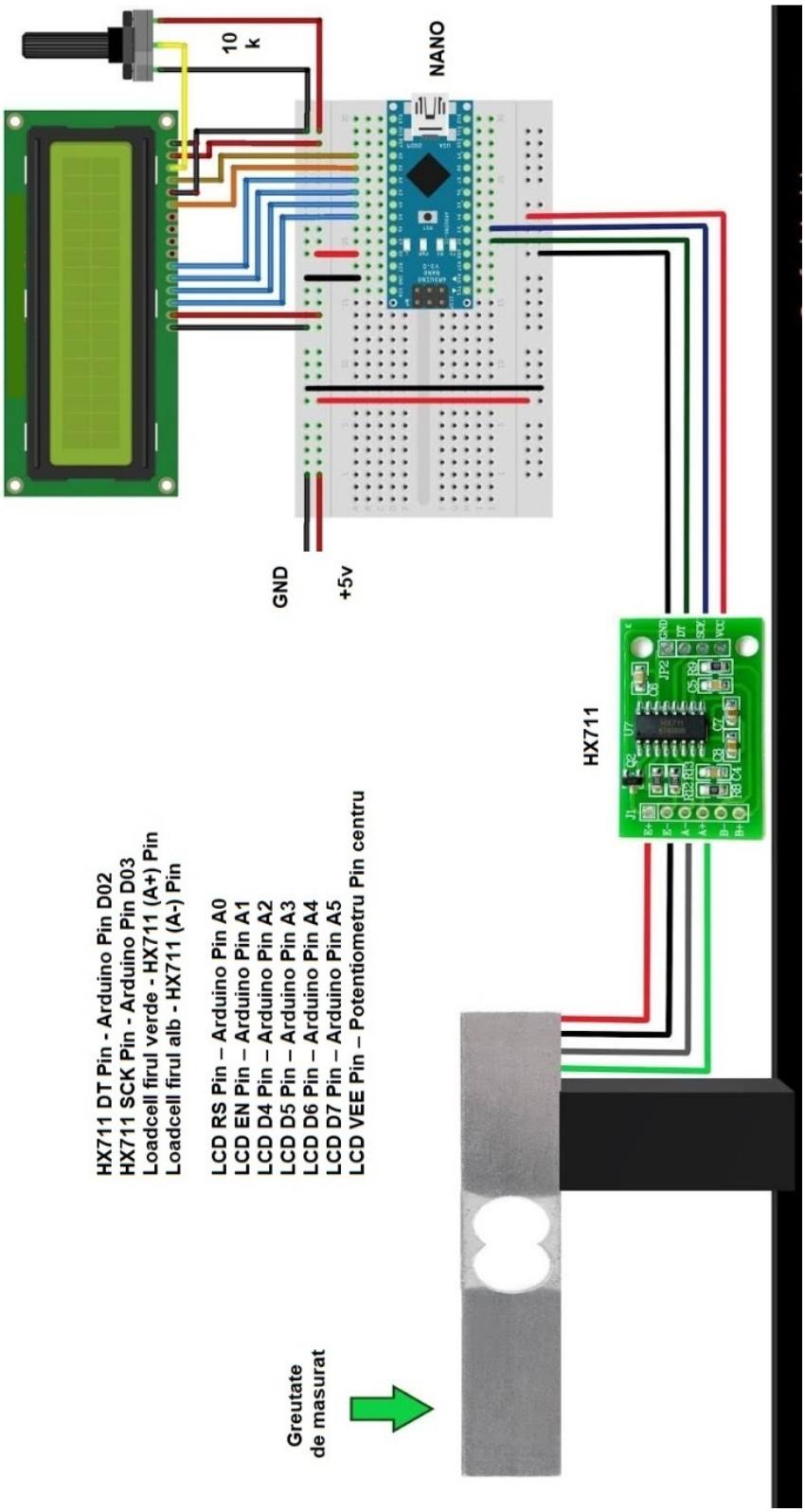
Vom scrie un cod :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);     //adresa 1 sau
//LiquidCrystal_I2C lcd(0x3f, 16, 2);    //adresa 2

void setup()
{
        lcd.begin();
        lcd.backlight();
        lcd.print("Roboromania");
}

void loop()
{
  lcd.setCursor ( 0, 0 );
  lcd.print ("Roboromania");
  lcd.setCursor ( 0, 1 );
  lcd.print ("Roboromania");
}
```

Mult mai simplu!

Afiseaza pe display "Roboromania". Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Exemplul 14 Folosirea senzorului de greutate de 1kg



Piese necesare varianta NANO :

1 x placă Arduino NANO

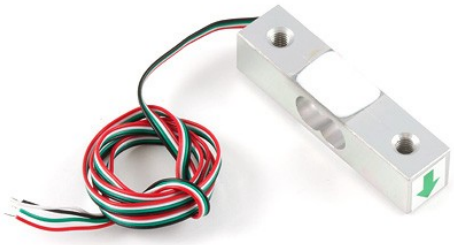
1 x LCD 2×16 caractere LED Blue Display

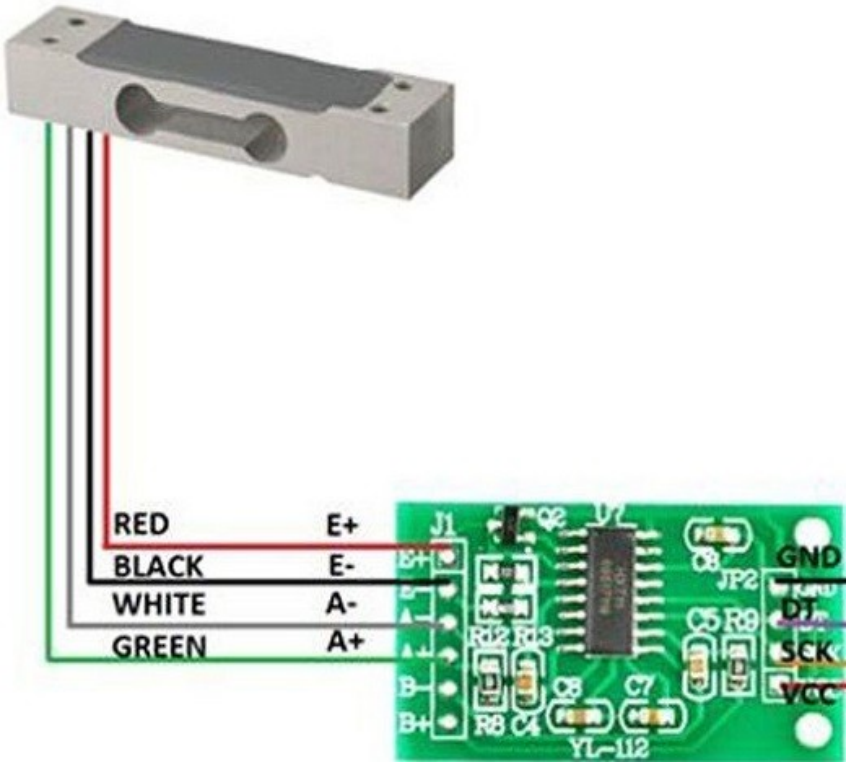
1 x Senzor Greutate 1kg

1 x Potentiometru trimmer 10kΩ

1 X Breadboard 420 (121)

cabluri de legătură necesare între piese şi placă





Necesar de instalat Librăria HX711

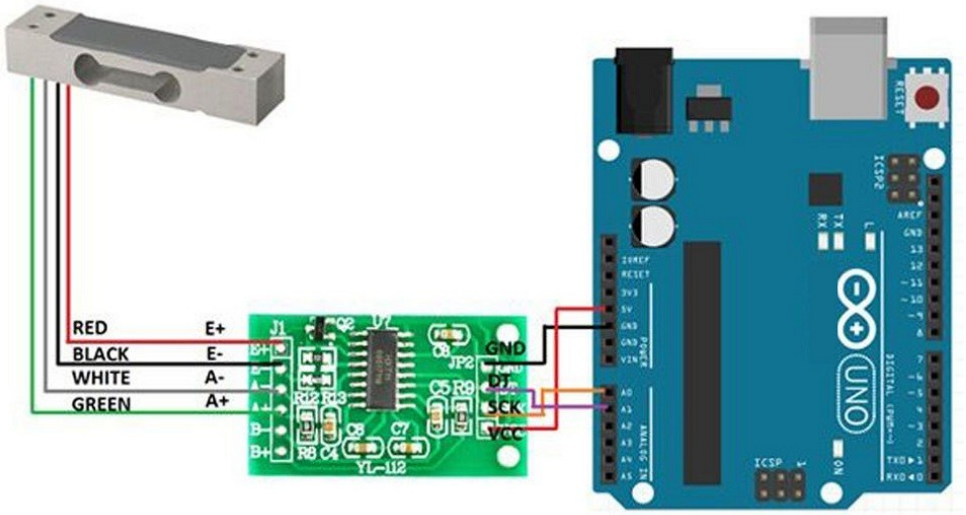
Vom scrie un cod :

```
//----------------
#include "HX711.h"
const int LOADCELL_DOUT_PIN = 2;
const int LOADCELL_SCK_PIN = 3;
HX711 scale;

#include <LiquidCrystal.h>
const int rs = A0, en = A1, d4 = A2, d5 = A3, d6 = A4, d7 = A5;
LiquidCrystal lcd(A0, A1, A2, A3, A4 ,A5);

void setup()
{
  lcd.begin(16, 2);
  Serial.begin(9600);
  delay(100);

  Serial.println("Weight ");
  Serial.println("Measuring...");
```

```
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  scale.set_scale(2280.f);
  scale.tare();

  lcd.print("Insert Weight");
  delay(100);
  lcd.clear();
 }

void loop()
{
  Serial.print("one reading:\t");
  Serial.print(scale.get_units(), 1);
  Serial.print("\t| average:\t");
  Serial.println(scale.get_units(10), 1);
  scale.power_down();
  delay(100);
  scale.power_up();

  lcd.print("Weight :");
  delay(100);
  lcd.clear();
  delay(1);

  lcd.print(scale.get_units());
  lcd.print("g");
  delay(100);
  lcd.clear();
  delay(1);
}

//-----------------
```

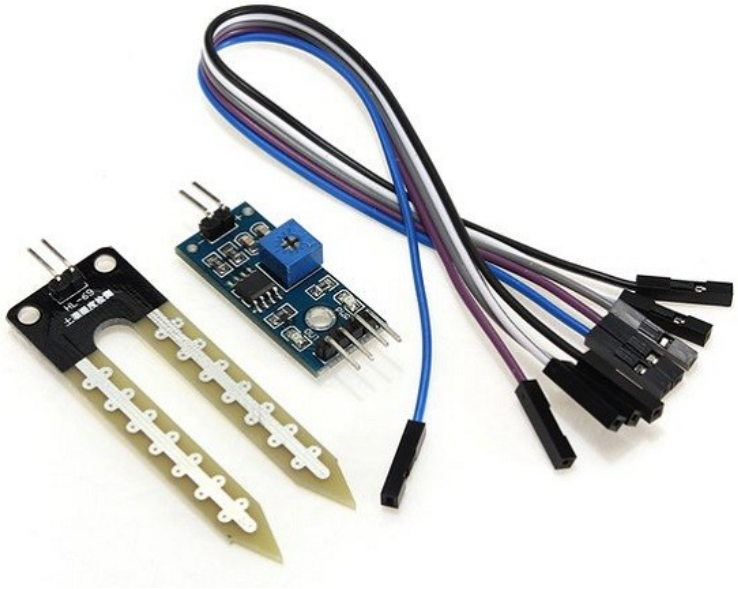
Atenție la fire și la conectare.

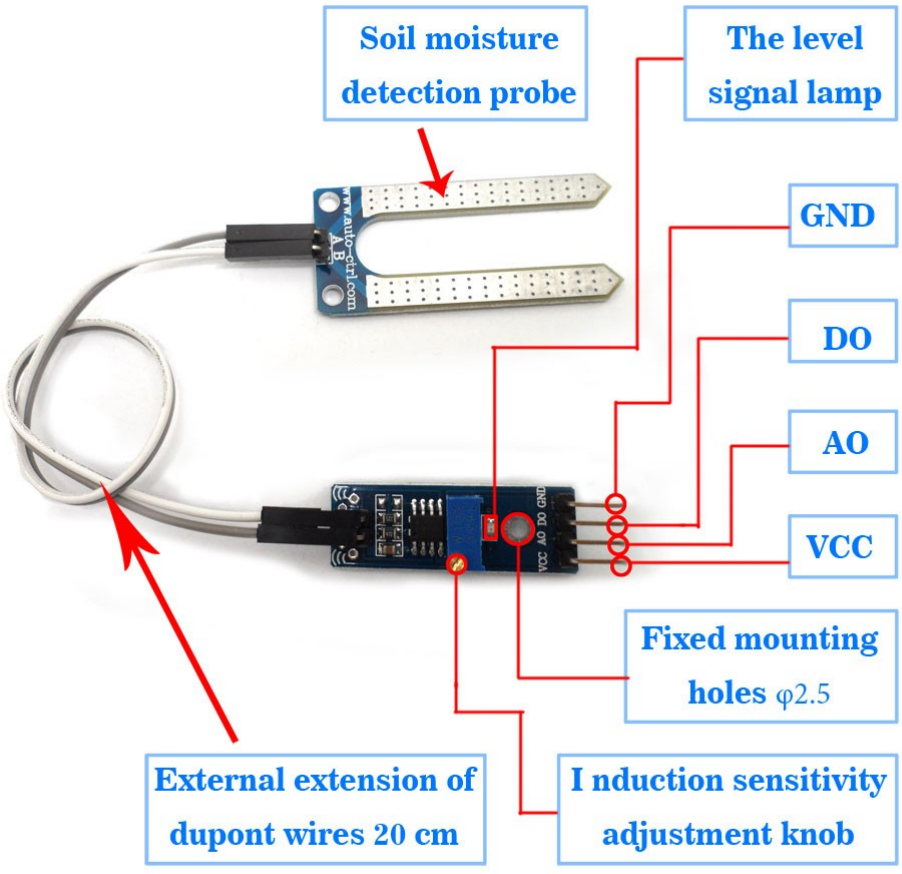
Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.



Exemplul 15 Folosirea Senzorului de umiditate sol

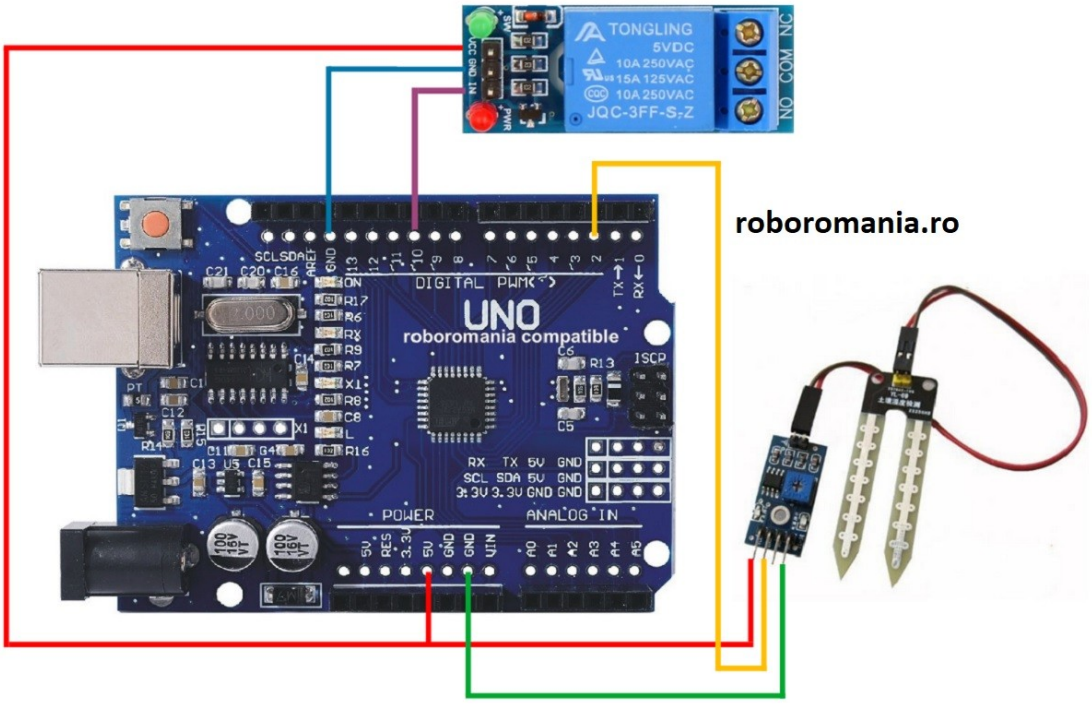
(Soil Hygrometer Humidity)





Senzor umiditate sol (Soil Hygrometer Humidity) compatibil Arduino detectează cantitatea apei din sol.
Electrozii senzorului acționează ca un rezistor variabil care va schimba de la 100kohmi (atunci când sunt ude) la 2Mohmi în stare uscată.
Sensibilitatea se poate regla cu un potențiometru.
Utilizează un comparator de tensiuni LM393.
Tensiunea de lucru de la 3,3 V la 5 V.

Pini :
A0 – Analog output
D0 – Digital output
GND – Ground
VCC – Positive voltage 3,3V – 5V

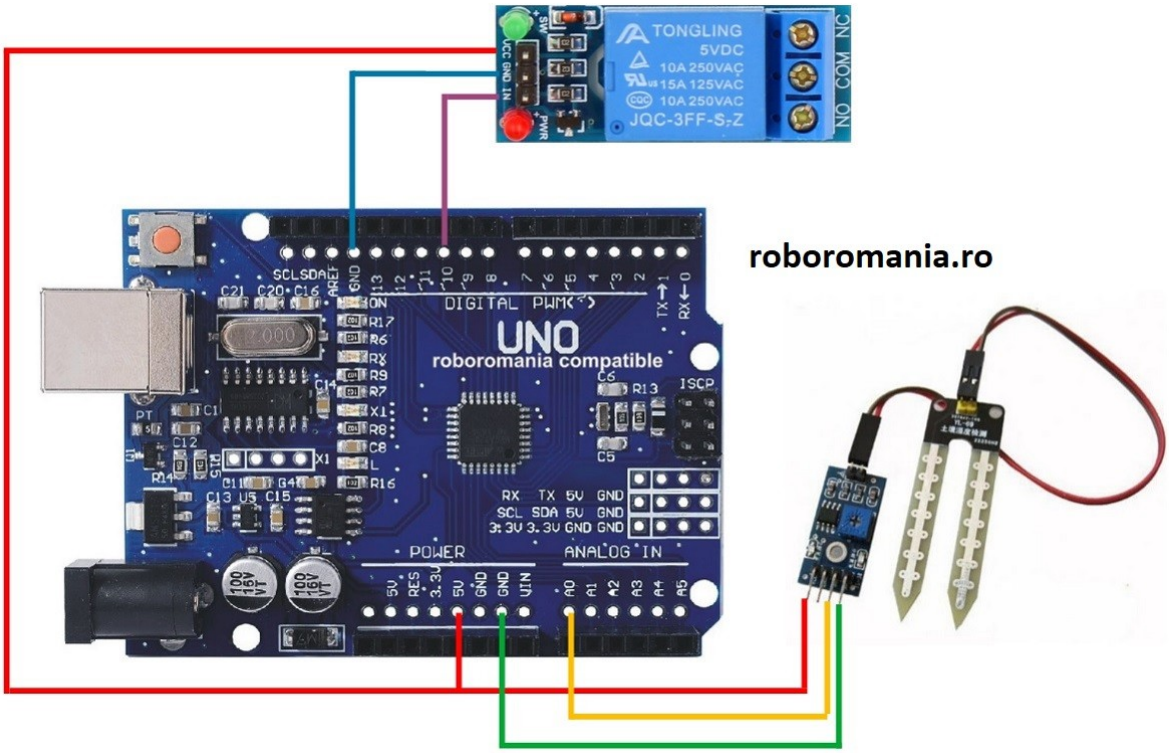


Vom scrie un cod pentru varianta iesire digitala :

// varianta iesire digitala

```
int rel_pin = 10;
int sensor_pin = 2;

void setup() {
pinMode(rel_pin, OUTPUT);
pinMode(sensor_pin, INPUT);

}
```

```
void loop() {
if(digitalRead(sensor_pin) == HIGH){
digitalWrite(rel_pin, HIGH);
} else {
digitalWrite(rel_pin, LOW);
delay(1000);
}
}
// ———-
```



Vom scrie un cod pentru varianta iesire digitala :

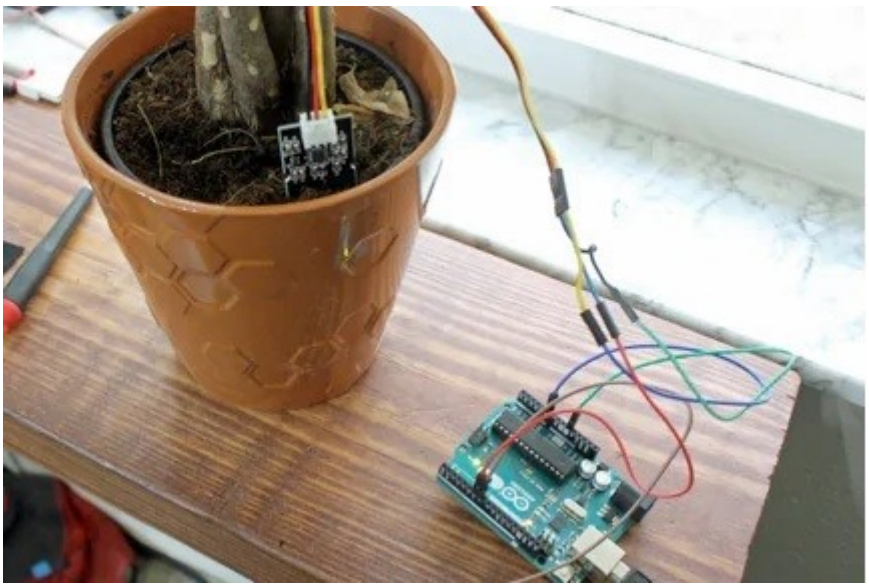
```
// varianta iesire analogica
int rel_pin = 10;
int sensor_pin = A0;

int output_value = 0;
int setup_value = 50; // se seteaza in cod intre 0% si 100%

void setup() {
Serial.begin(9600);
pinMode(rel_pin, OUTPUT);
digitalWrite(rel_pin, LOW);
}
```

```
void loop() {
output_value= analogRead(sensor_pin);
output_value = map(output_value,550,0,0,100);
Serial.print(„Mositure : „);
Serial.print(output_value);
Serial.println(„%”);
delay(1000);
if(output_value < setup_value){
digitalWrite(rel_pin, HIGH);
} else {
digitalWrite(rel_pin, LOW);
delay(1000);
}
}
//————-
```

E simplu, trebuie doar să fim atenți la conectarea pinilor, corespunzător sketch-ului Arduino folosit.

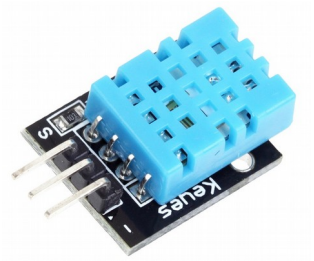
Atenție la declararea pinilor.

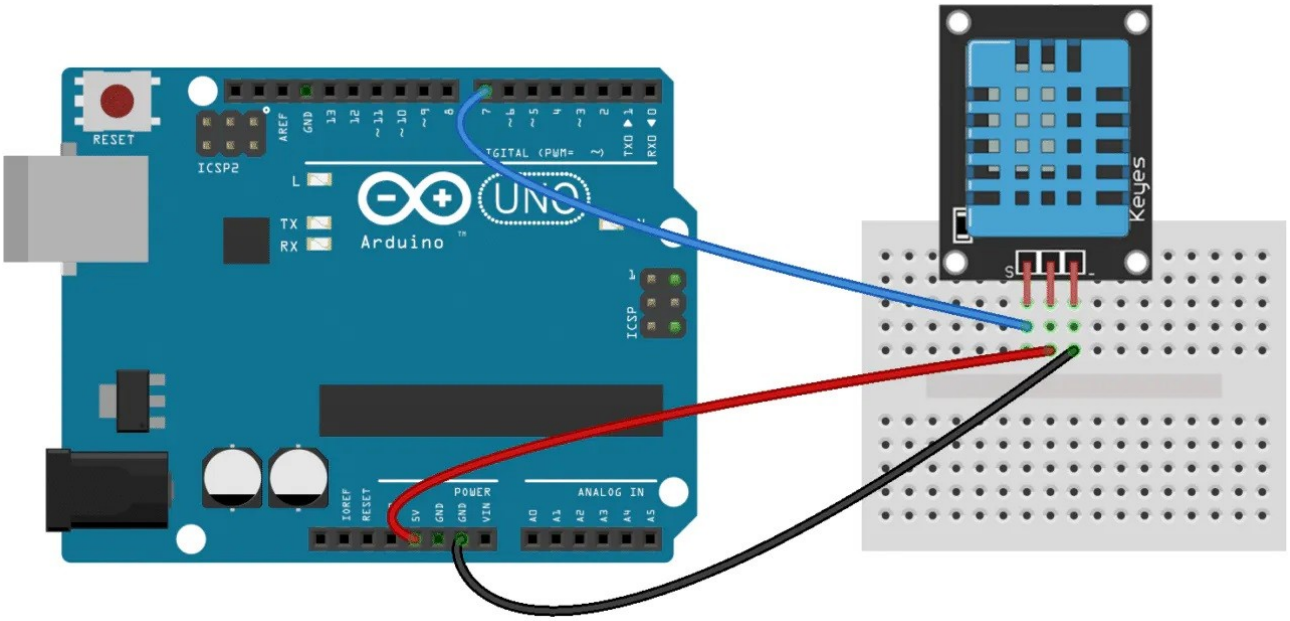
Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.



Exemplul 16 Folosirea Senzorului temperatură şi umiditate DHT11

DHT11 este un senzor de temperatură şi de umiditate ușor de folosit având o buna fiabilitate și stabilitate.
Temperatura este măsurată cu un NTC ,iar umiditatea relativă este măsurată folosind un senzor capacitiv .
Semnalul de ieșire este digital.





Piese necesare :

1 x placă Arduino UNO R3
1 x Senzor temperatură şi umiditate DHT11
cabluri de legătură necesare între piese şi placă

Necesar de instalat Librăria DHT

Vom scrie un cod :

```
#include <dht.h>
#define dht_dpin 2 //no ; here. Set equal to channel sensor is on
dht DHT;

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

```
  delay(300);//Let system settle
  Serial.println("Humidity and temperature\n\n");
  delay(700);//Wait rest of 1000ms recommended delay before
  //accessing sensor
}//end "setup()"

void loop(){
  //This is the "heart" of the program.
  DHT.read11(dht_dpin);

    Serial.print("Current humidity = ");
    Serial.print(DHT.humidity);
    Serial.print("%  ");
    Serial.print("temperature = ");
    Serial.print(DHT.temperature);
    Serial.println("C  ");
  delay(800);//Don't try to access too frequently... in theory
  //should be once per two seconds, fastest,
  //but seems to work after 0.8 second.
}// end loop()
```

Afiseaza pe Serial-Monitor temperatura și umiditatea aerului.

Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Exemplul 17 Folosirea Senzorului temperatură şi umiditate DHT11 Afisarea pe display LCD 2x16 i2C

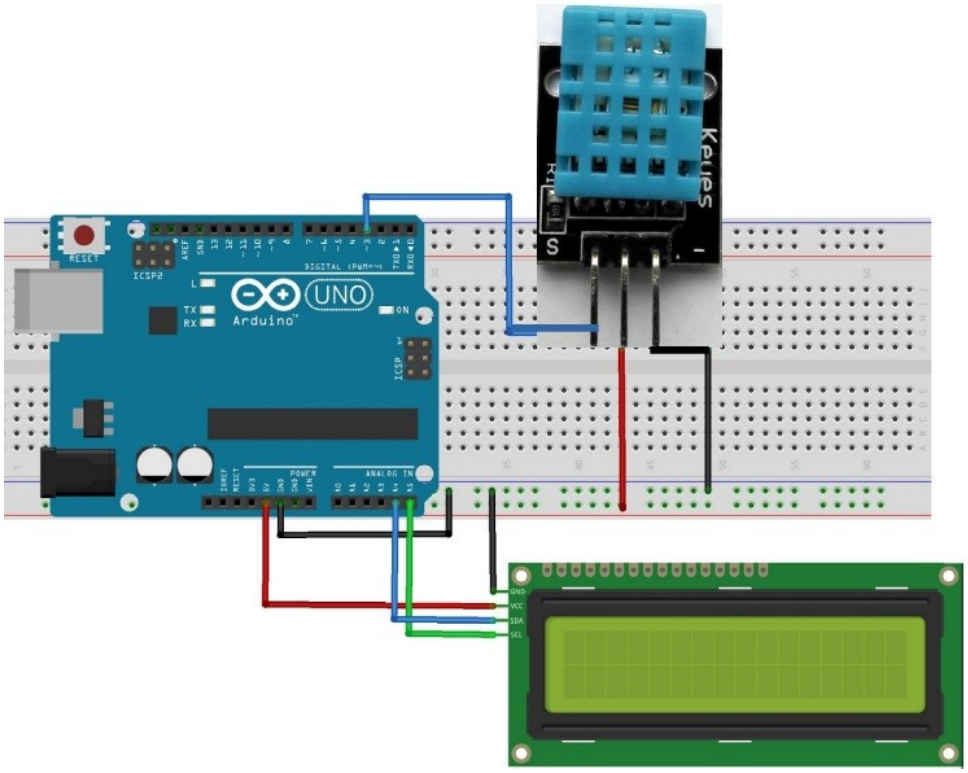
Piese necesare :

1 x placă Arduino UNO R3

1 x Senzor temperatură şi umiditate DHT11

1 x LCD 2x16 caractere LED Blue Display cu interfață i2C

cabluri de legătură necesare între piese şi placă



Pini LCD :
* VCC la 5V
* SDA la SDA sau pin Analog 04
* SCL la SCL sau pin Analog 05
* GND la GND

Pini DHT:
* (+)VCC la 5V
* (data)S la pin D3
* (-)GND la GND

Vom scrie un cod :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <dht.h>
#define dht_dpin 2 //no ; here. Set equal to channel sensor is on
dht DHT;

LiquidCrystal_I2C lcd(0x3f, 16, 2);

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

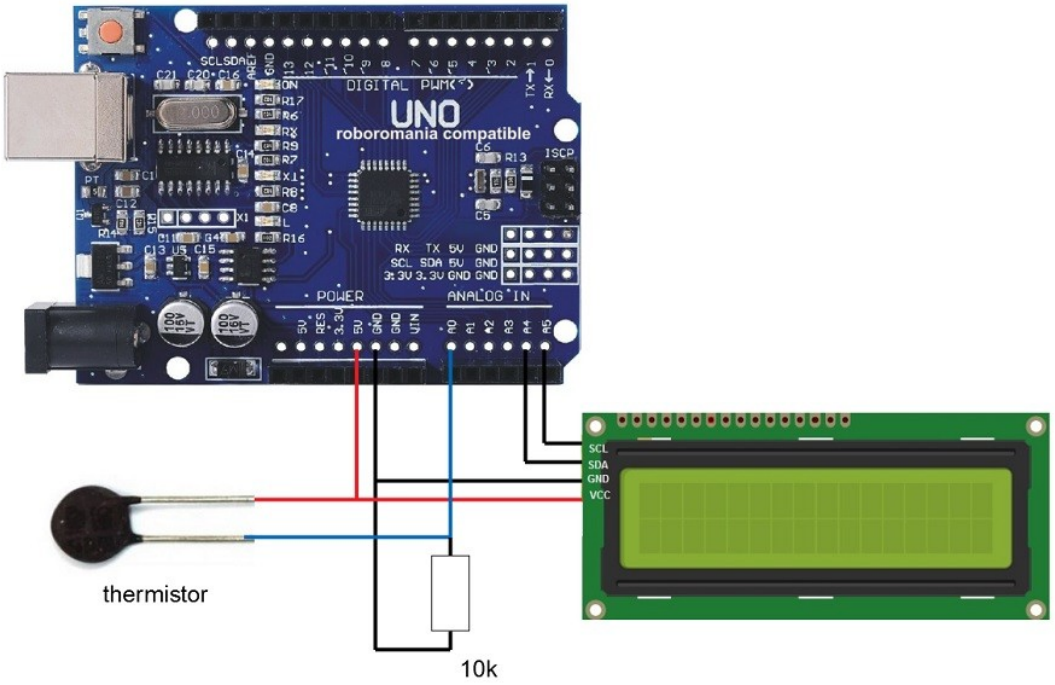
```
          lcd.begin();
          lcd.backlight();

 delay(500);
}

void loop()
{
  DHT.read11(dht_dpin);
    Serial.print("Current humidity = ");
    Serial.print(DHT.humidity);
    Serial.print("%  ");
    Serial.print("temperature = ");
    Serial.print(DHT.temperature);
    Serial.println("C  ");
  delay(800);

  lcd.setCursor ( 0, 0 );
  lcd.print ("humid = ");
  lcd.setCursor ( 9, 0 );
  lcd.print(DHT.humidity,0);
  lcd.setCursor ( 12, 0 );
  lcd.print("%  ");
  lcd.setCursor ( 0, 1 );
  lcd.print("temp. = ");
  lcd.setCursor (9, 1 );
  lcd.print(DHT.temperature,0);
  lcd.setCursor (12, 1 );
  lcd.println("C   ");
}
```

Afiseaza pe LCD temperatura și umiditatea aerului.

Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Exemplul 18 Măsurarea temperaturii cu un termistor de 10k și afișare pe LCD i2C

Termistorul este un rezistor cu valoare variabila în funcție de temperatura



Piese necesare :

1 x placă Arduino UNO R3

1 x Termistor 10kΩ

1 x LCD 2x16 caractere LED Blue Display cu interfață i2C

cabluri de legătură necesare între piese şi placă

1 x Rezistor 10kΩ

Termistor 10kΩ :

Rezistență la 25^{o}C = 10k;
Acuratețe: 1%;
Interval de temperatură masurabil: -40^{o}C … 200^{o}C;

E simplu, trebuie doar să fim atenți la conectarea pinilor, corespunzător sketch-ului Arduino folosit.

Pini LCD :
* VCC la 5V
* SDA la SDA sau pin Analog 04
* SCL la SCL sau pin Analog 05
* GND la GND

Vom scrie un cod :

```
#include <math.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3f, 16, 2);
float A = 1.009249522e-03, B = 2.378405444e-04, C = 2.019202697e-07;
float T,logRt,Tf,Tc;
float Thermistor(int Vo) {
 logRt = log(10000.0*((1024.0/Vo-1)));
 T = (1.0 / (A + B*logRt + C*logRt*logRt*logRt));  // Temperature value in Kelvin
 Tc = (T - 273.15) / 10U;          // Convert Kelvin to Celcius
 Tf = (Tc * 1.8) + 32.0;           // Convert Kelvin to Fahrenheit
 return T;
}
void setup(){
 lcd.begin();
lcd.backlight();
}
void loop()
{
 lcd.setCursor(0,0);
  lcd.print("Temp:");
  lcd.print(((Thermistor(analogRead(0)))) - 273.15);
  lcd.print("k ");
 lcd.setCursor(0,1);
  lcd.print((Tc));
  lcd.print(" C ;");
 lcd.setCursor(9,1);
  lcd.print((Tf));
  lcd.print(" F");
 delay(800);
}
```

Afiseaza pe LCD temperatura aerului.

Atenție dacă copiati codul din acest manual în ID poate sa apara erori de la „fonturi”, pe care trebuie să le corectați.

Urmează Vol.2