

## Cuprins

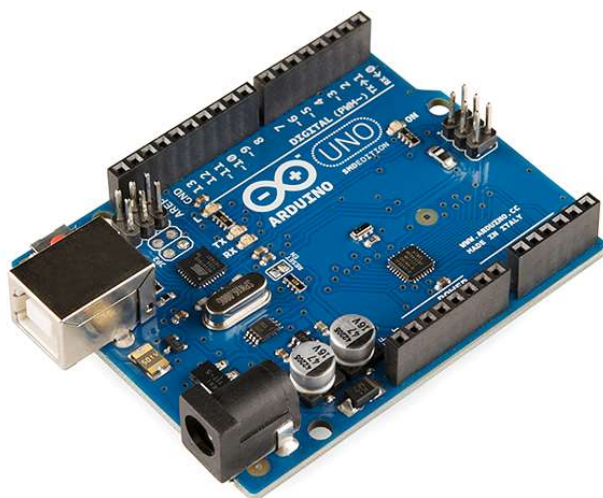
<b>Introducere .....</b>	<b>2</b>
<b>Componente electronice .....</b>	<b>3</b>
<b>Concluzii .....</b>	<b>16</b>
<b>Bibliografie .....</b>	<b>17</b>

## Introducere

Robotica este știința care se ocupă cu tehnologia, proiectarea și fabricarea roboților. Robotica educativă este un mediu de învățare în care persoanele implicate sunt motivate de proiectare și construcție de creații proprii. Aceste creații sunt prezentate în primă instanță într-o formă mentală și mai târziu în formă fizică, care sunt construite cu diverse tipuri de materiale și controlate de un sistem informatic, așa numitele prototipuri sau simulări. (preluare Wikipedia)

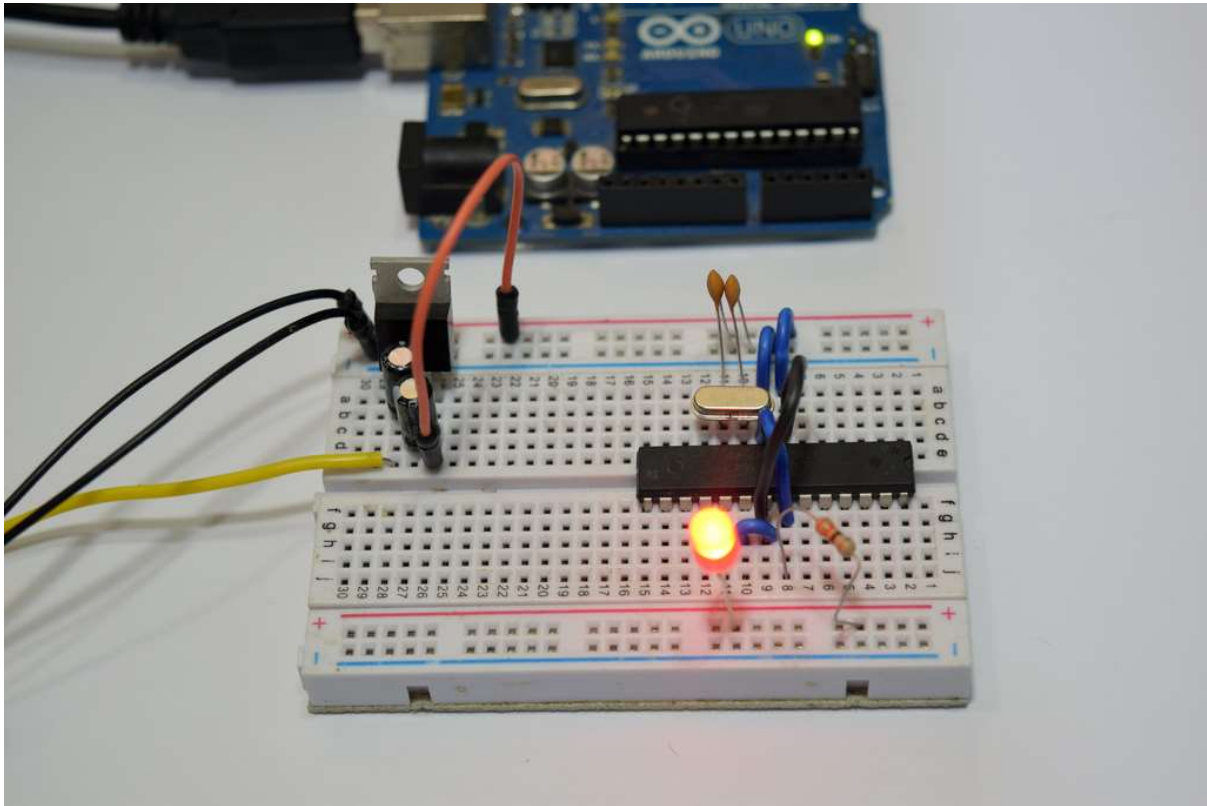
Ca urmare a apariției pe piață a unor diverse plăci de dezvoltare, senzori, motoare, ș.m.d.p. la prețuri deosebit de atractive acest auxiliar a fost elaborat din dorința de a dezvoltui elevilor un domeniu nou și anume robotica.

Elevii vor fi puși la curent cu cele mai noi tehnologii utilizate în crearea și programarea roboților în scop demonstrativ. În plus, elevii pot observa asemănări, deosebiri cu alte probleme, algoritmi, întâlniți la informatică în clasele IX-XII. De menționat aici ar fi faptul că nu se pune accentul pe componentele electronice din punct de vedere al fizicii ci pe maniera de a îmbina pe acestea cu informatica cu scopul de a descoperii această frumoasă ramură. Din acest motiv principalele componente electronice sunt prezentate pe scurt.



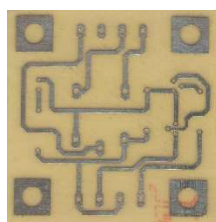
# Componente electronice

## 1. Breadboard



Un **breadboard** este o placă de test sau de prototipuri, cu anumite terminale și contacte în sistem grilă, unde se pot introduce diferite componente electronice, pentru a realiza circuite temporare. Nu avem nevoie de pistol de lipit, pur și simplu componentele se introduc în găurile aferente, de regulă contactele sunt legate pe verticală, iar în partea de sus a plăcii se găsesc două linii de contacte legate pe orizontală. Acestea din urmă de folosesc pentru alimentarea cu curent electric.

După testarea funcționalității unui proiect circuitul se poate realiza ulterior pe un PCB (Printed Circuit Board).



## 2. Fire

Rolul firelor este simplu, realizează conexiuni între diverse componente și placa Arduino.



## 3. Rezistorul

Rezistorul este o componentă electrică care reduce curentul electric. Capacitatea rezistorului de a reduce curentul se numește rezistență și se măsoară în unități de ohmi (simbol:  $\Omega$ ).



## 4. Condensatorul

Condensatorul este o componentă electronică care stochează sarcina electrică. Condensatorul este format din 2 conductori apropiați (de obicei plăci) care sunt separați de un material dielectric. Plăcile acumulează sarcină electrică atunci când sunt conectate la sursa de alimentare. O placă acumulează sarcină pozitivă iar cealaltă placă acumulează sarcină negativă.

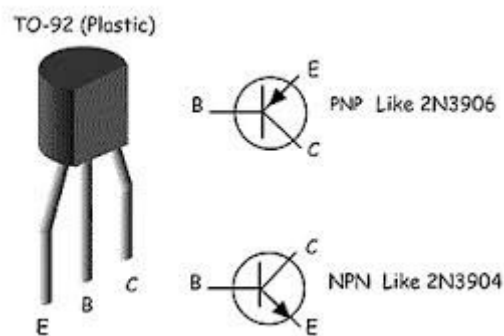
Capacitatea este cantitatea de încărcare electrică stocată în condensator la tensiunea de 1 Volt. Capacitatea este măsurată în unități de Farad (F).

Condensatorul deconectează curentul în circuitele de curent continuu (DC) și scurtcircuitul în circuitele de curent alternativ (AC).



## 5. Tranzistorul

Tranzistorul este un dispozitiv electronic din categoria semiconductoarelor care are cel puțin trei terminale (borne sau electrozi), care fac legătura la regiuni diferite ale cristalului semiconductor. Este folosit mai ales pentru a amplifica și a comuta semnale electronice și putere electrică



## 6. Dioda

Dioda este o componentă electronică cu două terminale având conductanță asimetrică. Dioda are rezistență mică (ideal zero) la trecerea curentului într-o direcție și rezistență mare (ideal infinită) la trecerea în cealaltă direcție.



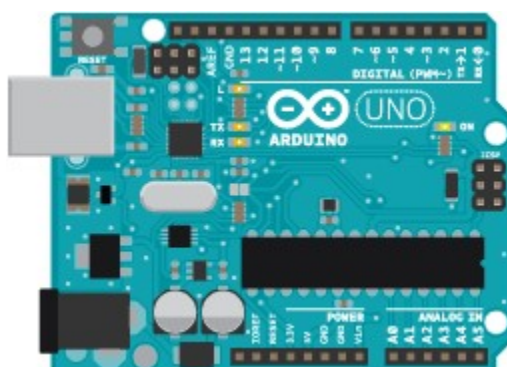
## 7. Instrumente utile

În afara de un calculator de cele mai multe ori avem nevoie de câteva instrumente utile în construcția și realizarea micilor proiecte de introducere în acest interesant domeniu. Așa cum se poate observa, nu ar trebui să lipsească șurubelnița, cleștele, patentul, aparatul de măsură, ciocanul de lipit...



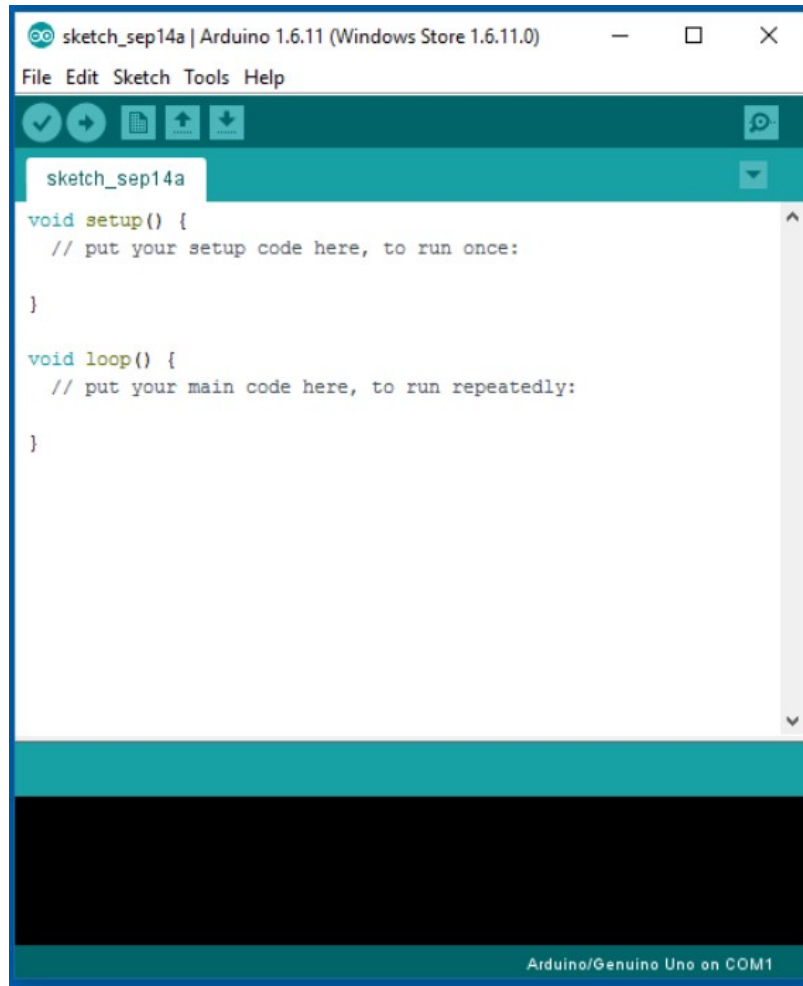
## 8. Arduino

Arduino este o platformă open-source utilizată pentru construirea de proiecte electronice. Arduino constă atât într-o placă de circuite programabile fizic (microcontroler), cât și într-o parte de software sau pe scurt IDE (Integrated Development Environment) care rulează pe computerul dvs. Acesta este folosit pentru scrie și încărca codul computerului pe un board fizic.



Utilizarea unei astfel de platforme este simplă, conexiunea cu pc-ul este realizată printr-un cablu usb, instalarea drivere-lor se face simplu dacă această placă nu este totuși recunoscută automat de sistemul de operare.

## 9. Software



Ca și orice alt mediu gratuit de programare după descărcare și instalare, sunt puse la dispoziția utilizatorului un editor de text pentru editarea codului sursă, un meniu pentru salvare, încărcare aplicații, diverse setări, ajutor, etc.

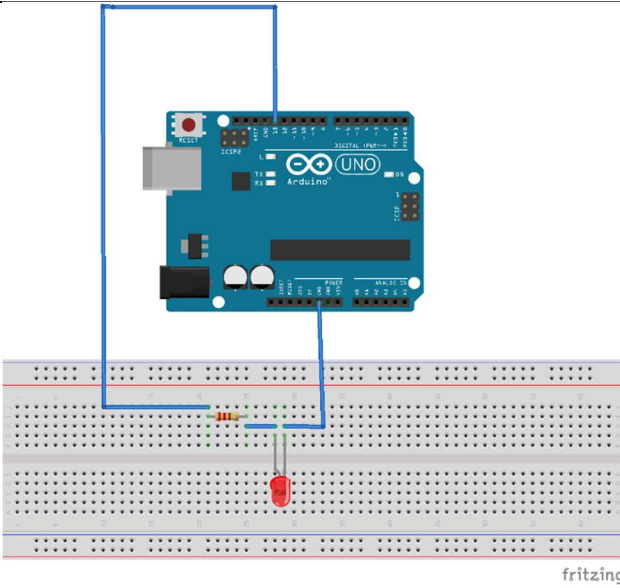
Arduino Este probabil unul dintre cele mai de succes proiecte sau platforme de software și hardware gratuit și cel care a avut cel mai mare impact în lumea DIY. Comunitatea a creat atât software open source pentru programarea microcontrolerului plăcilor, cât și diferite plăci hardware cu care sunt, de asemenea, libere să lucreze. Toate sunt licențiate sub licența GNU GPL, astfel







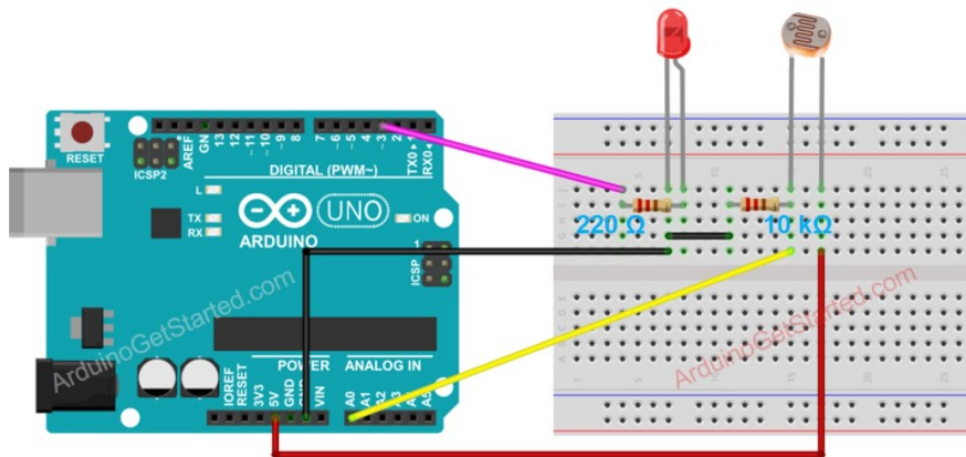
## 10.Ledul

 <p>void loop() – programul principal, va rula atâta timp cât placa primește curent, pornește ledul (HIGH), la digitalWrite(LED, LOW) acesta va fi oprit</p>	<pre>#define LED 13 void setup() {   pinMode(LED, OUTPUT); } void loop() {   digitalWrite(LED, HIGH); }</pre> <p>Ledul + este conectat la pinul 13, se definește LED , masa (-) la GND, avem nevoie de o rezistență (220 ohm, altfel riscam să ardem această componentă). Observăm #define – definirea pinului 13 void setup() – se declară aici ledul ca ieșire</p>
---	--

Pentru a varia intensitatea acestui led, a simula o aprindere lentă, stingere lentă trebuie să folosim un alt pin, de această dată analogic, fără a intra în alte detalii de ordin tehnic, iată codul asociat acestui experiment practic.

<pre>void setup() {   pinMode(10, OUTPUT); } void loop() {   for (int i = 0; i &lt; 255; i++)   {     analogWrite(10, i);     delay(50);   }   for (int i = 255; i &gt; 0; i--)   {     analogWrite(10, i);     delay(50);   } }</pre>	<p>Fără a utiliza define, putem seta direct pinul de ieșire.</p> <p>Iată un for simplu care va trimite spre led (pinul 10) valoarea lui i în creștere apoi în descreștere.</p> <p>Desigur trebuie asigurat un delay (o mică pauză) că altfel viteza de schimbare a intensității va fi prea mare și aproape insesizabilă pentru ochiul uman.</p> <p>Această întrerupere (delay), pauză o să o mai folosim pentru a da posibilitatea anumitor senzori de a trimite / primii date corespunzătoare din mediul înconjurător.</p>
--	---

## 11.Senzor de lumină



```

const int senzor = A0;
const int led = 3;
int citire;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  citire = analogRead(senzor);
  if(citire < 400)
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);
}

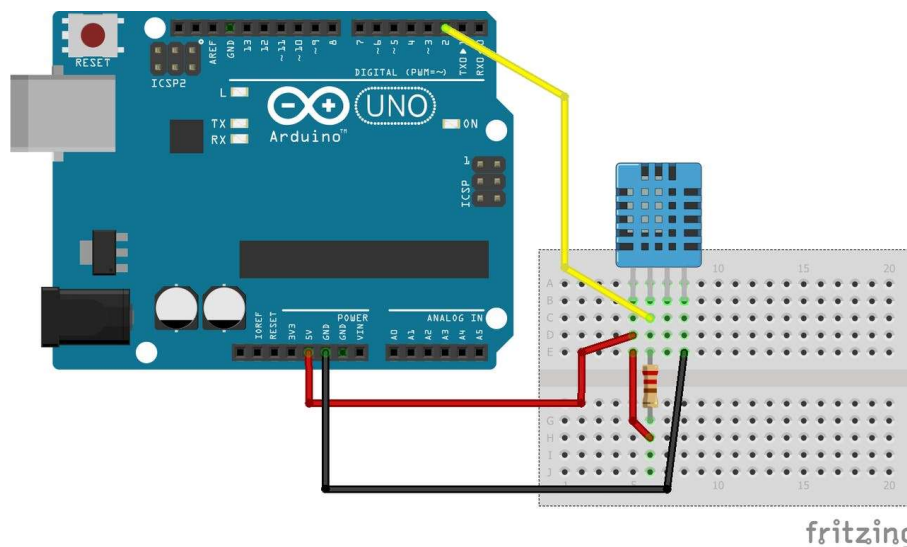
```

Senzorul de lumina, un pin se leagă la A0, analog, celălalt la masă printr-un rezistor, ledul aici la pinul 3.

Se primește, prin citire valoarea transmisă de senzorului de lumină, analog, dacă valoarea e sub 400, se pornește automat ledul, altfel se oprește.

Iată o simulare simplă de automatizare, aprindere / stingere a unui bec într-o încăpere.

## 12.Senzor de temperatură și umiditate



O schemă simplă de măsurare a temperaturii și umidității mediului înconjurător.

```

#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600);
}
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  return;
}
float hif = dht.computeHeatIndex(f, h);
float hic = dht.computeHeatIndex(t, h, false);
Serial.print(F(" Humidity: "));
Serial.print(h);
Serial.print(F("% Temperature: "));
Serial.print(t);
Serial.print(F("C "));
Serial.print(f);
Serial.print(F("F Heat index: "));
Serial.print(hic);
Serial.print(F("C "));
Serial.print(hif);
Serial.println(F("F"));
}

```

#include "DHT.h" – unele componente beneficiază de propriile librării, propriile funcții pentru a ușura utilizarea acestora. Aceste fișiere ori sunt deja prezente kit-ul arduino ori se pot atașa prin copiere sau instalare. Serial.begin se poate utiliza, și se recomanda utilizarea acesteia, pentru a afișa anumite mesaje, valori, în fereastra arduino pentru testarea proiectului.

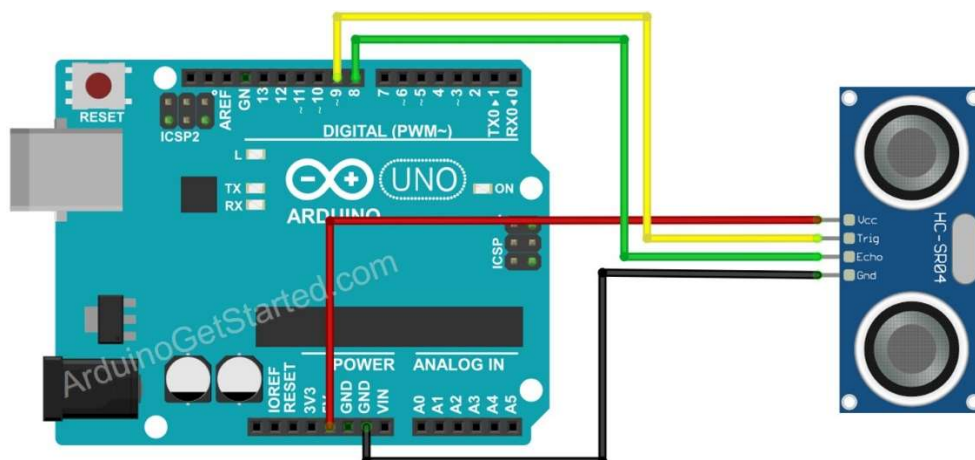
Citire, calcul și afișarea datelor.

Multe componente, "primesc" din partea producătorilor librării, exemple de utilizare a lor, foarte simplu și ușor de implementat în proiect.

Combinarea acestui senzor cu orice modul (ecran, display) poate duce la o afișare mult mai comoda a datelor.

De obicei un proiect care trebuie să prezinte utilizatorului date utile, informații, utilizează o astfel de componentă.

### 13.Senzor ultrasunete

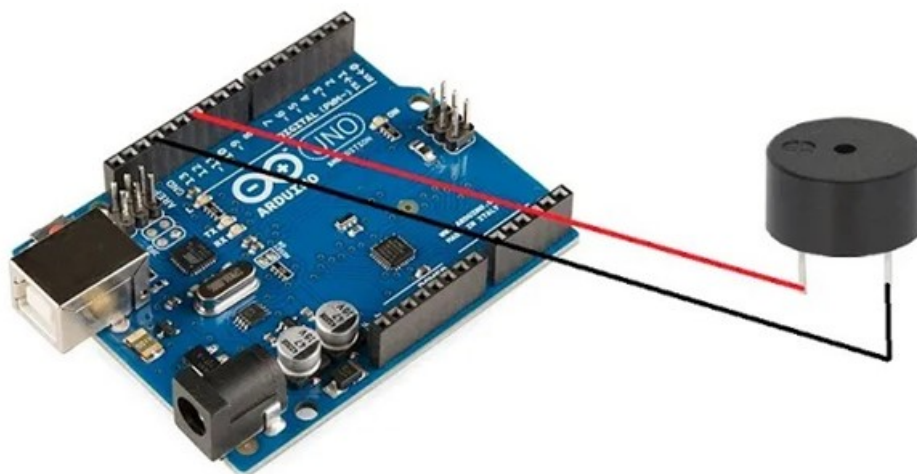


Senzorul de ultrasunete este o altă componentă utilă, interesantă în diverse proiecte. După cum îi spune și numele, acesta trimite o undă sonoră care, se reflectă dintr-un obstacol, revine înapoi și astfel se poate calcula distanța față de un obiect.

<pre>int trigPin = 9; int echoPin = 8; float duration_us, distance_cm; void setup() {   Serial.begin(9600);   pinMode(trigPin, OUTPUT);   pinMode(echoPin, INPUT); } void loop() {   digitalWrite(trigPin, HIGH);   delayMicroseconds(10);   digitalWrite(trigPin, LOW);   duration_us = pulseIn(echoPin, HIGH);   distance_cm = 0.017 * duration_us;   Serial.print("distance: ");   Serial.print(distance_cm);   Serial.println(" cm");   delay(500); }</pre>	<p>Setarea pinilor</p> <p>Variabile pentru calculul distanței măsurate</p> <p>Citiri, scrieri de date, calcule pentru măsurarea distanței folosind ultrasunetele.</p>
---	---

## 14. Buzzer

Este un difuzor de mici dimensiuni, pasiv, care poate genera tonuri între anumite frecvențe.



```

int buzzer = 10;
void setup() {
  pinMode(buzzer, OUTPUT);
}
void loop() {
  for (int i = 0; i < 80; i++) {
    digitalWrite(buzzer,
HIGH);
    delay(1);
    digitalWrite(buzzer, LOW);
    delay(1);
  }
  delay(50);
  for (int j = 0; j < 100; j++) {
    digitalWrite(buzzer,
HIGH);
    delay(2);
    digitalWrite(buzzer, LOW);
    delay(2);
  }
  delay(100);
}
}

```

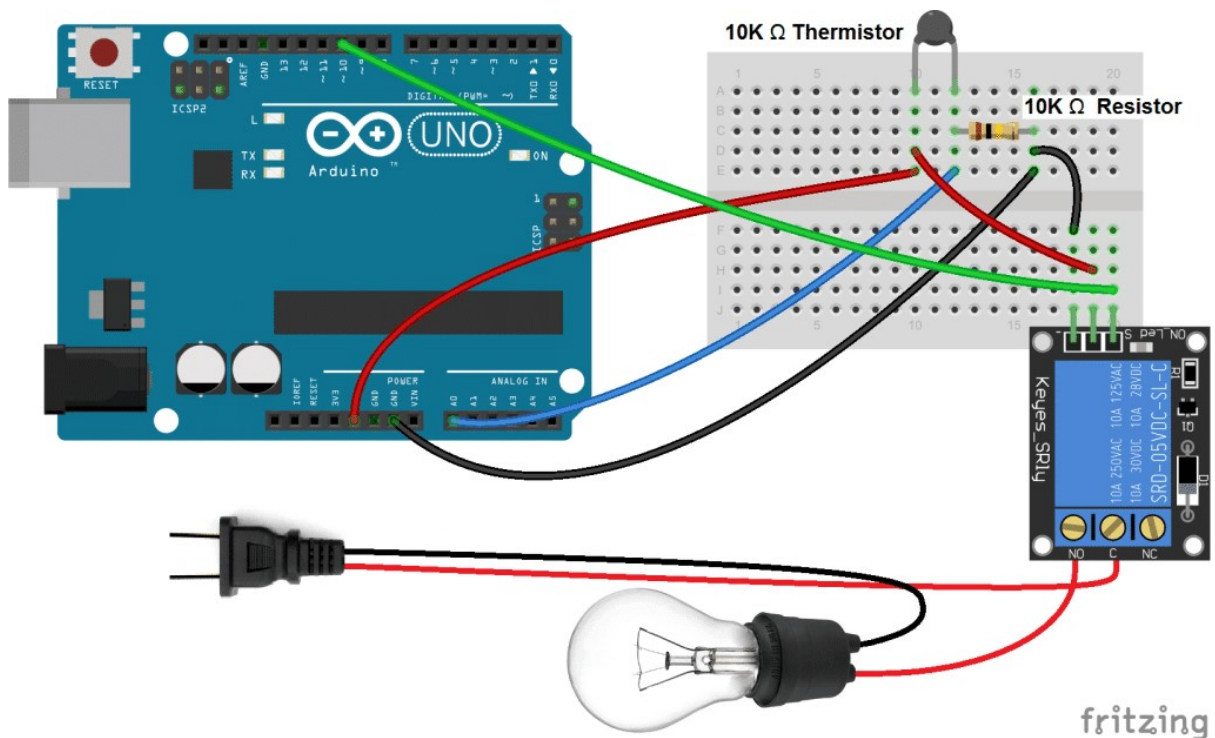
Setarea pin buzzer, ca output

Parcurgeri de valori diferite, cu delay diferit duce la sunete diferite

La o căutare simplă pe internet puteți descoperii o serie de aplicații capabile să redea o serie de cântecele, toate bazate pe astfel de utilizari, ex:  
<https://create.arduino.cc/project-hub/jrance/super-mario-theme-song-w-piezo-buzzer-and-arduino-1cc2e4>

## 15. Releu

Acest modul este deosebit de util dacă dorim să controlăm componente de putere și nu numai. Pe baza anumitor parametrii acest dispozitiv poate să deschidă sau să închidă un circuit.



```

#include <math.h>
int pinOut = 10;
double Thermistor(int RawADC) {
  double Temp;
  Temp = log(10000.0*((1024.0/RawADC-1)));
  Temp = 1 / (0.001129148 + (0.000234125 +
(0.000000876741 * Temp * Temp ))* Temp );
  Temp = Temp - 273.15;
  Temp = (Temp * 9.0) / 5.0 + 32.0;
  return Temp;
}
void setup() {
  Serial.begin(9600);
  pinMode(10, OUTPUT);
}
void loop() {
  int val;
  double temp;
  val=analogRead(0);
  temp=Thermistor(val);
  Serial.print("Temperature = ");
  Serial.print(temp);
  if (temp >= 100){
    digitalWrite(pinOut, LOW);
  }
  else {
    digitalWrite(pinOut, HIGH);
  }
  delay(500);
} (pinOut, HIGH); } delay(500); }

```

Exact ca in C++ arduino accepta includerea de librării diverse.

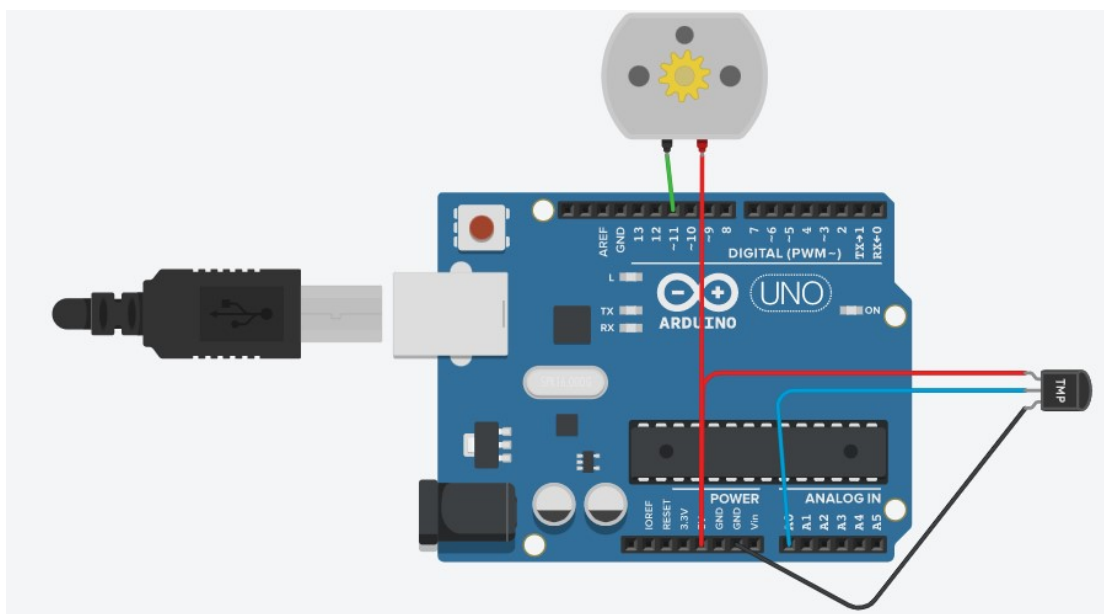
Funcție pentru calcul și returnare temperatură.

Așa cum reiasă din imaginea de mai sus, analizând acest cod, putem observa că la o temperatură mai mare de 100 grade releul va *deschide* circuitul din care face parte becul => oprire a lui automată și, va comuta pentru închiderea circuitului permițând becului să fie alimentat pentru funcționare.

Observăm utilitatea acestui dispozitiv care deși e alimentat la 5 volți poate fi utilizat în anumite proiecte care implică 220 volți.

## 16.Motor

Un motor este un dispozitiv care transformă energia electrică în energie mecanică. Aceste dispozitive sunt din ce în ce mai folosite în diverse proiecte deoarece oferă mișcare, mobilitate.





```

int sensorPin = 0;
int motorPin = 11;
void setup()
{
  Serial.begin(9600);
  pinMode(motorPin, OUTPUT);
}

void loop()
{
  int reading = analogRead(sensorPin);
  float voltage = reading * 5.0;
  voltage /= 1024.0;
  float temperatureC = (voltage - 0.5) * 100 ;
  Serial.print(temperatureC); Serial.println(" grade C");
  if(temperatureC<=50)digitalWrite(motorPin, HIGH);
  else digitalWrite(motorPin, LOW);
  delay(1000);
}

```

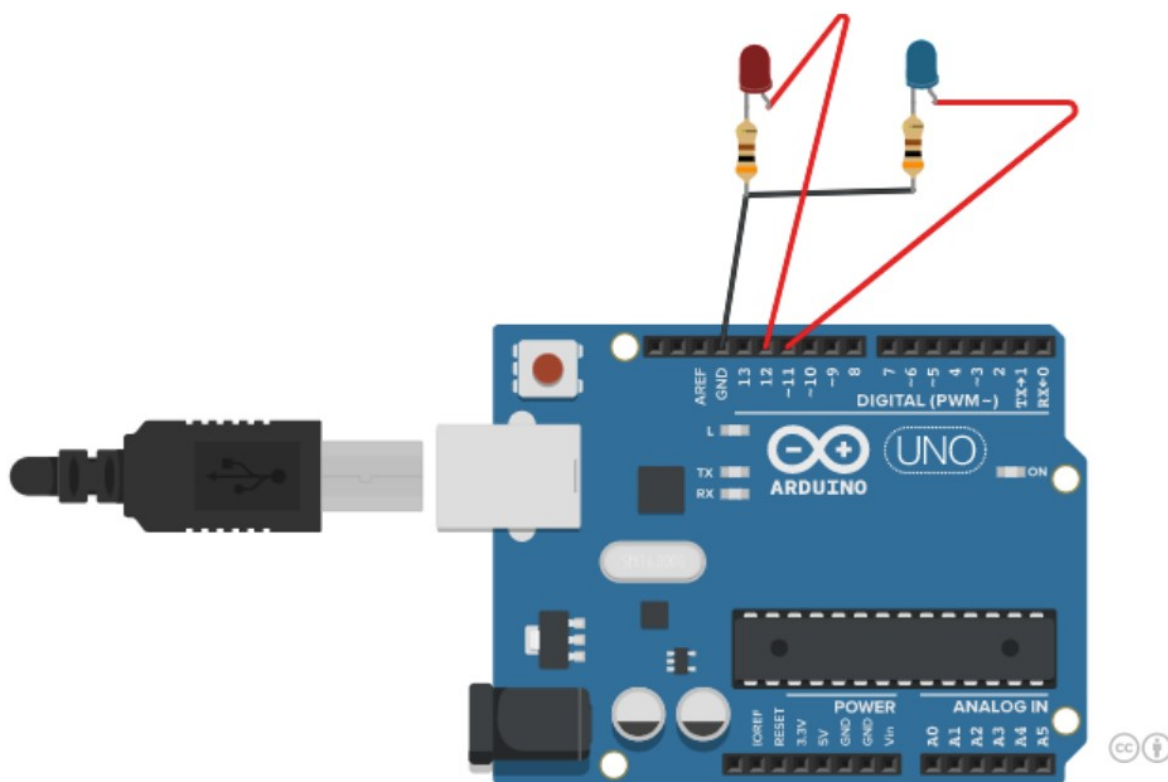
Avem aici un simulator ce poate fi folosit ca răricea unei componente care generează căldură (ex. procesor), desigur prin atașarea unei elice motorului.

Funcționarea este simplă, se convertesc semnalele primite de la senzorul de temperatură în grade celsius, și dacă temperatura depășește un prag minim, setat în aplicație, motorul începe să învârtă elicea, în caz contra desigur motorul se oprește

## Concluzii

Au fost prezentate mai sus câteva scheme simple pentru a evidenția ușurința folosirii unei plăci arduino împreună cu o serie de dispozitive atât din punct de vedere hardware cât și software. Înaintea utilizării componentelor fizic, trebuie să subliniez aici că, există publicate cu acces gratuit o serie de aplicații online care permit simularea unor astfel de proiecte, desigur multe limitate din punct de vedere al componentelor disponibile. O astfel de aplicație este tinkercad accesibila la <https://www.tinkercad.com/>.

Există desigur mult mai multe dispozitive la care avem acces și împreună cu documentația acestora, oricine poate să realizeze o serie de aplicații, circuite, dispozitive mai mult sau mai puțin complexe ce pot fi utilizate în viața reală.



## Bibliografie

1. <https://www.instructables.com/How-to-Build-an-Arduino-Uno-on-a-BreadBoard/>
2. [https://www.rapidtables.org/ro/electric/Electronic\\_components.html](https://www.rapidtables.org/ro/electric/Electronic_components.html)
3. <https://www.elforum.info/topic/135632-poze-cu-tot-felul-de-componente-electronice/page/48/>
4. <https://electroniclight.ro/2n3904-tranzistor-npn-40v/1349.htm>
5. <https://ro.wikipedia.org/wiki/Diod%C4%83>
6. <https://conectica.ro/retelistica/scule-si-truse-pentru-retelistica/trusa-de-scule-electronica-28-piese-gembird-tk-solder>
7. <https://cleste.ro/atelier/placa-de-dezvoltare-arduino/>
8. <https://www.arduino.cc/>
9. <https://arduino.stackexchange.com/questions/816/c-vs-the-arduino-language>
10. <https://www.hwlibre.com/ro/>
11. <https://arduinogetstarted.com/tutorials/arduino-light-sensor>
12. <https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239>
13. <https://ardushop.ro/ro/home/47-modul-senzor-ultrasonic-detector-distanta.html>
14. <https://arduinogetstarted.com/tutorials/arduino-ultrasonic-sensor>
15. <https://create.arduino.cc/projecthub/munir03125344286/add-buzzer-to-arduino-bf010b>
16. <https://cleste.ro/modul-buzzer-pasiv.html>
17. <https://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/>
18. <https://sites.google.com/site/arduinoelectronicsiprogramare/roboti/motoare-si-miscare>
19. <https://www.hwlibre.com/ro/motor-paso-a-paso/>
20. [https://www.tutorialspoint.com/arduino/arduino\\_dc\\_motor.htm](https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm)
21. <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>
22. <https://roboromania.ro/2016/11/15/uno-led-cu-intensitate-variabila/>
23. <https://blog.robofun.ro/2021/09/24/ce-este-un-breadboard-si-care-este-scopul-in-proiecte-de-electronica/>