

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



Codul sursa din acest document este licentiat

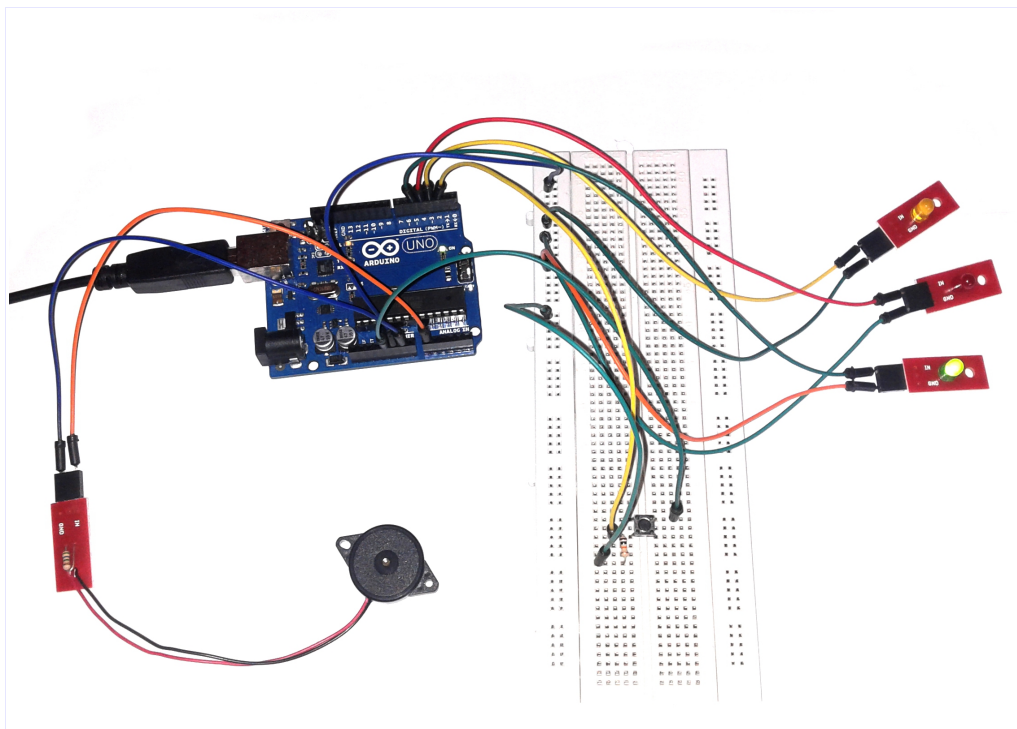
Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

## Arduino UNO si senzorul de vibratii brick

Senzorul de vibratii brick este o componenta care sesizeaza vibratiile mecanice (este capabil sa detecteze vibratiile de genul unei batai in usa). In acest tutorial, vei conecta senzorul de vibratii brick direct la placa Arduino si vei utiliza 3 led-uri brick (portocaliu, rosu, verde) pentru a evidentia comportamentul senzorului.

Al doilea exemplu pe care il vom discuta va fi cel al unei yale electronice care deschide usa atunci cand bati in usa conform unui anumit cod. Mai exact, daca bati in usa conform cu un anumit tipar, senzorul va detecta vibratiile iar Arduino le va compara cu un model deja existent in memoria lui. Daca tiparele se potrivesc, se va aprinde led-ul portocaliu, echivalentul deblocarii usii.



Proiectul complet il vei gasi la adresa de mai jos:

<http://www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/?ALLSTEPS>

<http://www.robofun.ro/forum>

Nu trebuie decat sa bati in usa dupa modelul explicat in link-ul de mai sus (in pagina, il vei gasi cu numele: "Shave and a Hair Cut, two bits.") iar led-ul portocaliu se va aprinde.

In imagine este prezent si un buton. Este necesar doar atunci cand incarci sketch-ul de la adresa anterioara. In cazul tau, poti conecta foarte usor un buton brick.

Vei avea nevoie de urmatoarele componente:

- Arduino UNO.
- Senzor de vibratii brick.
- 3 led-uri brick.
- Fire pentru conexiuni.
- Breadboard.

Tabelul de conexiuni este mai jos.

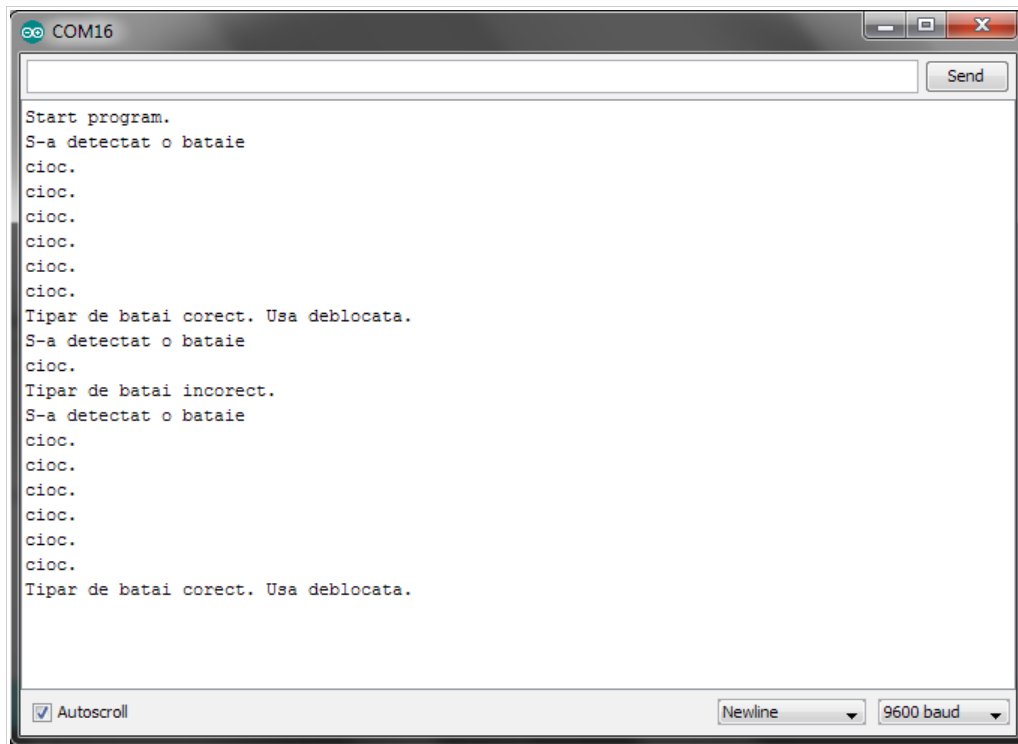
|                                     |                          |
|-------------------------------------|--------------------------|
| <b>Senzor vibratii pin IN</b>       | <b>Arduino analog A0</b> |
| <b>Senzor vibratii pin Gnd</b>      | <b>Arduino Gnd</b>       |
| <b>Led brick portocaliu pin IN</b>  | <b>Arduino digital 3</b> |
| <b>Led brick rosu pin IN</b>        | <b>Arduino digital 4</b> |
| <b>Led brick verde pin IN</b>       | <b>Arduino digital 5</b> |
| <b>Led brick portocaliu pin Gnd</b> | <b>Arduino Gnd</b>       |
| <b>Led brick rosu pin Gnd</b>       | <b>Arduino Gnd</b>       |
| <b>Led brick verde pin Gnd</b>      | <b>Arduino Gnd</b>       |

## Ce incarc in Arduino ?

Mai jos este listat sketch-ul (varianta simplista). Il copiezi (copy/paste) direct in mediul de programare Arduino si il incarci in placa Arduino UNO.

Incearca sa bati in apropierea senzorului ca si cum ai bate intr-o usa. Daca ai batut dupa modelul "Shave and a Hair Cut, two bits." (explicat in link-ul de mai sus - Instructables), led-ul verde se aprinde urmarind fiecare bataie. La sfarsit, daca modelul de batai este recunoscut cu cel din memoria interna, se aprinde led-ul portocaliu. In caz contrar, se aprinde led-ul rosu intr-o scurta secventa.

Daca vrei sa te asiguri ca bataile sunt detectate corect, deschide monitorul serial si vei obtine ceva ca in imaginea de mai jos.



## Codul sursa.

Codul sursa de mai jos este o varianta mult simplificata a celui prezentat in Instructables (spre exemplu, nu poti inregistra un nou tipar).

```
const int senzorVibratii = 0;
const int ledPortocaliu = 3;
const int ledRosu = 4;
const int ledVerde = 5;
const int pragDetectie = 3;
const int gradRespingere = 25;
const int gradMediuRespingere = 15;
const int perioadaOFF = 150;
const int perioadaON = 650;
const int bataiMax = 20;
const int perioadaTotala = 1200;

int modelBatai[bataiMax] = {50, 25, 25, 50, 100, 50, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int bufferCitiri[bataiMax];
int nivelSenzor = 0;
```

```

void setup() {
    pinMode(ledPortocaliu, OUTPUT);
    pinMode(ledRosu, OUTPUT);
    pinMode(ledVerde, OUTPUT);
    Serial.begin(9600);
    Serial.println("Start program.");
    digitalWrite(ledVerde, HIGH);
}

void loop() {
    nivelSenzor = analogRead(senzorVibratii);
    if (nivelSenzor >= pragDetectie) {
        capturaTipar();
    }
}

void capturaTipar() {
    Serial.println("S-a detectat o bataie");

    int i = 0;
    for (i=0; i<bataiMax; i++) {
        bufferCitiri[i]=0;
    }

    int indexBatai=0;
    int perioadaStart=millis();
    int perioadaCurenta=millis();

    digitalWrite(ledVerde, LOW);
    delay(perioadaOFF);
    digitalWrite(ledVerde, HIGH);

    do {
        nivelSenzor = analogRead(senzorVibratii);
        if (nivelSenzor >= pragDetectie) {
            Serial.println("cioc.");
            perioadaCurenta=millis();
            bufferCitiri[indexBatai] = perioadaCurenta-perioadaStart;
            indexBatai ++;
            perioadaStart=perioadaCurenta;
            digitalWrite(ledVerde, LOW);
            delay(perioadaOFF);
            digitalWrite(ledVerde, HIGH);
        }

        perioadaCurenta=millis();
    }
}

```

```

    } while ((perioadaCurenta-perioadaStart < perioadaTotala) &&
(indexBatai < bataiMax));

    if (modelValid() == true){
        activeazaLED();
    } else {
        Serial.println("Tipar de batai incorect.");
        digitalWrite(ledVerde, LOW);
        for (i=0;i<4;i++){
            digitalWrite(ledRosu, HIGH);
            delay(100);
            digitalWrite(ledRosu, LOW);
            delay(100);
        }
        digitalWrite(ledVerde, HIGH);
    }
}

void activeazaLED(){
    Serial.println("Tipar de batai corect. Usa deblocata.");
    int i=0;
    digitalWrite(ledPortocaliu, HIGH);
    digitalWrite(ledVerde, HIGH);
    delay (perioadaON);
    digitalWrite(ledPortocaliu, LOW);
    digitalWrite(ledVerde, LOW);
}

boolean modelValid(){
    int i=0;
    int indexBataieCurenta = 0;
    int indexBataieModel = 0;
    int intervalMaximBatai = 0;

    for (i=0;i<bataiMax;i++){
        if (bufferCitiri[i] > 0){
            indexBataieCurenta++;
        }
        if (modelBatai[i] > 0){
            indexBataieModel++;
        }
        if (bufferCitiri[i] > intervalMaximBatai){
            intervalMaximBatai = bufferCitiri[i];
        }
    }
    if (indexBataieCurenta != indexBataieModel){

```

```

        return false;
    }

    int diferentaTotala=0;
    int diferentaTimp=0;
    for (i=0;i<bataiMax;i++){
        bufferCitiri[i]= map(bufferCitiri[i],0, intervalMaximBatai, 0,
100);
        diferentaTimp = abs(bufferCitiri[i]-modelBatai[i]);
        if (diferentaTimp > gradRespingere){
            return false;
        }
        diferentaTotala += diferentaTimp;
    }
    if (diferentaTotala/indexBataieModel>gradMediuRespingere){
        return false;
    }
    return true;
}

```