

# Decoder cod morse

## Introducere

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

Proiectul meu constă într-un dispozitiv ce are rolul de a traduce codul morse transmis prin bătăi, deci prin vibrații (o lovitură normală reprezintă '.', iar cele mai puternice '-').

Ideea de decodare de cod morse a plecat de la un proiect pe care l-am găsit online ce detecta un cod secret bătut la ușă (o parolă secretă). Ideea principală mi s-a părut interesantă, însă am venit cu această dezvoltare deoarece mi s-a părut că fiind ceva mai complexă și cel puțin la fel de interesantă.

Cred că este util deoarece ar ajuta pe cineva ce nu cunoaște traducerile codului morse să traducă un mesaj.

## Descriere generală

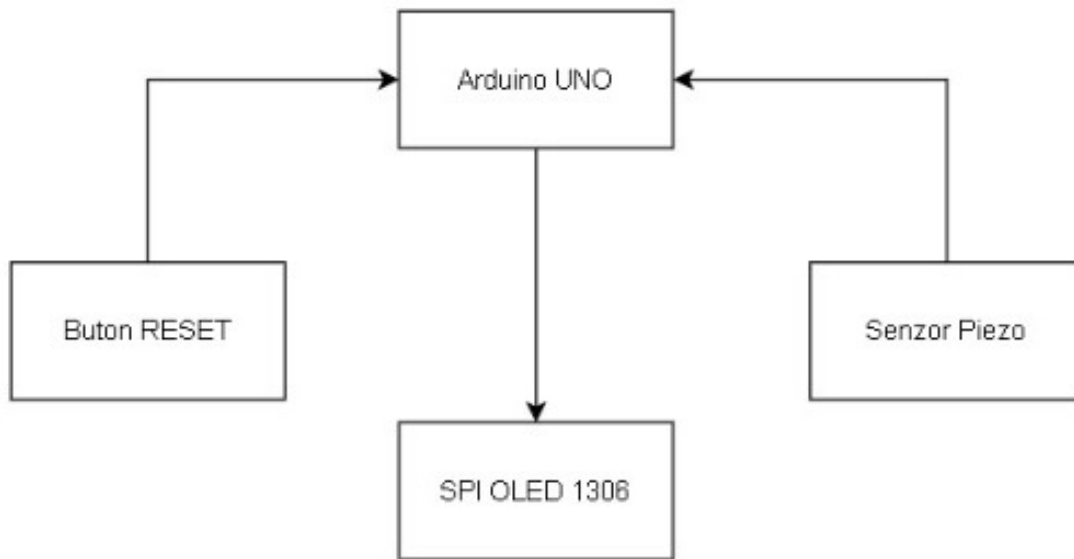
O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>

Dispozitivul se folosește de vibrațiile receptate de piezo prin bataile în cutie. Acestea sunt apoi filtrate în funcție de valoarea pe care o trimite mai departe senzorul și se obțin caractere din alfabetul morse. Acestea sunt apoi traduse și se afișează la ecran cuvântul rezultat.

Se poate folosi butonul pentru a șterge tot conținutul cuvântului și pentru a porni unul nou.

## Schema bloc



## Hardware Design

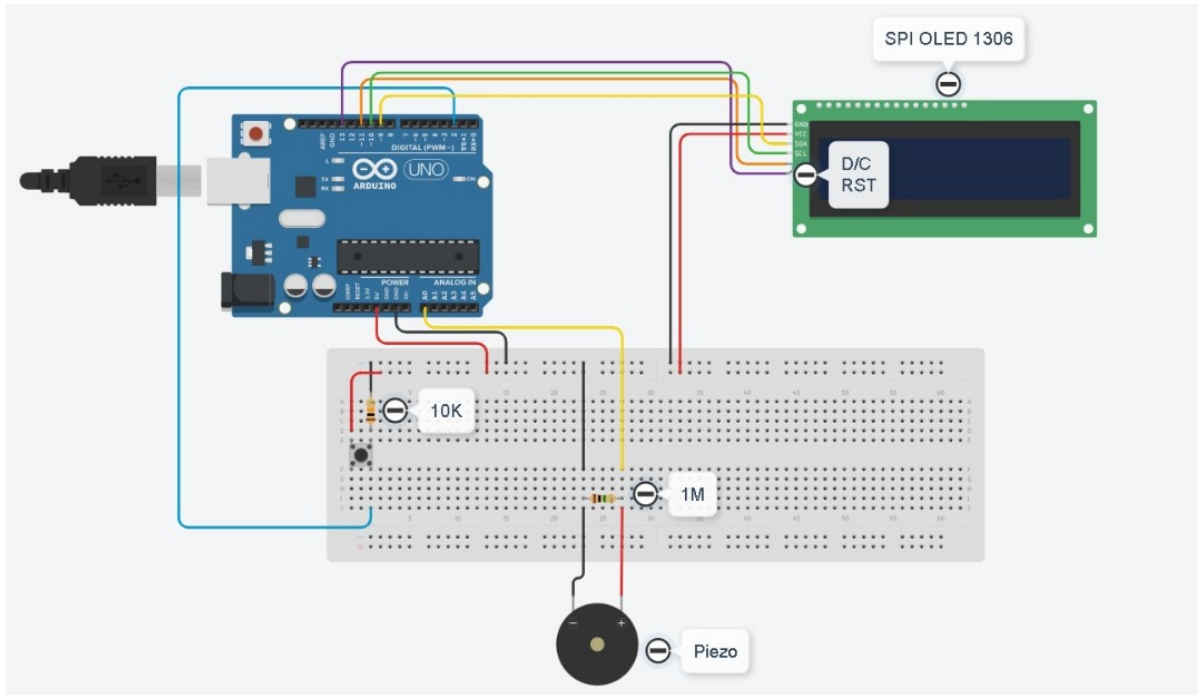
Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

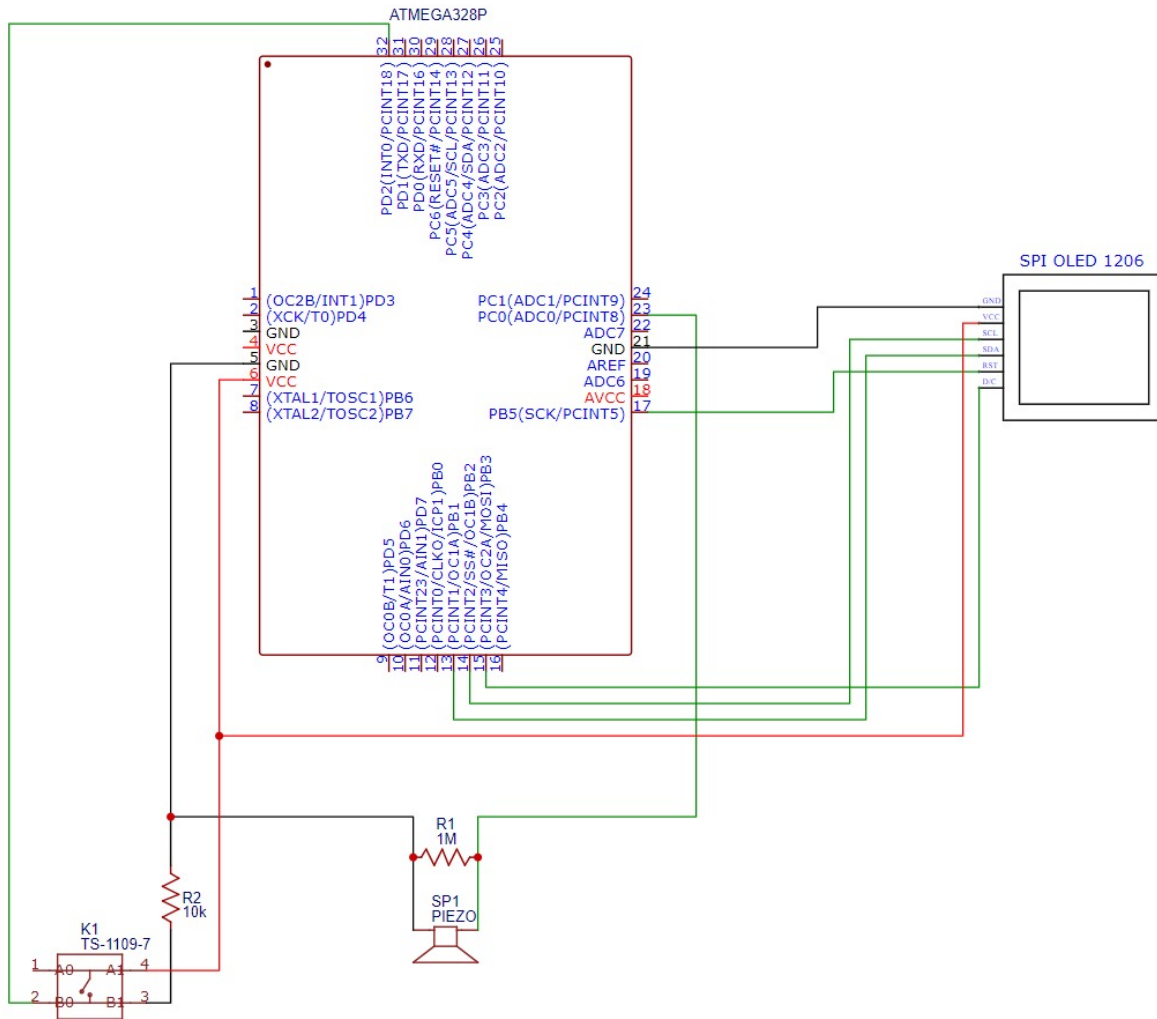
## Lista de piese

- 1 x plăcuță de dezvoltare Arduino Uno
- 1 x breadboard
- 1 x ecran SPI OLED 1306
- 1 x rezistență 10K
- 1 x buton
- 1 x piezo
- 1 x rezistență 1M
- fire de conectare

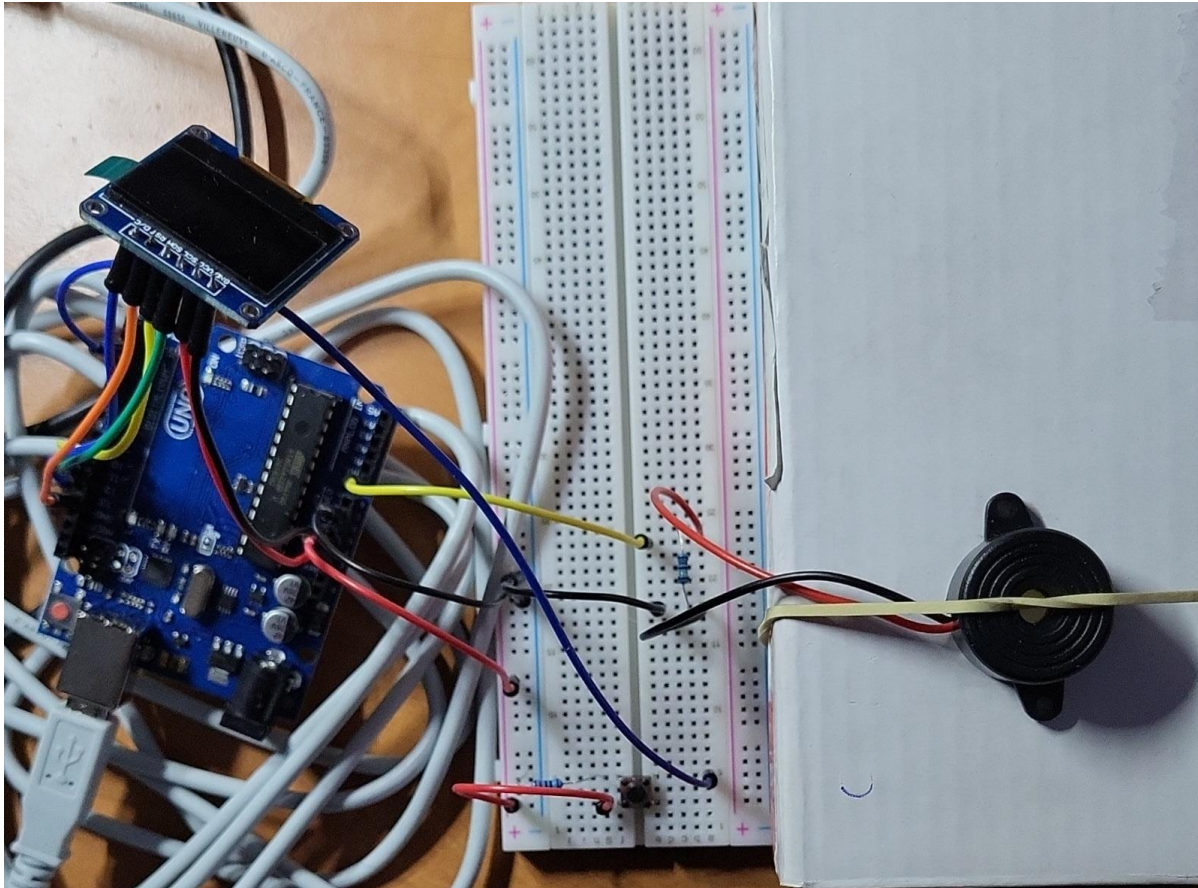
### Schema hardware



### Schema electrica



## Poze componente hardware



## Software Design

Descrierea codului aplicației (firmware):

- medii de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Descrierea codului aplicației (firmware):

1. mediu de dezvoltare: Arduino IDE 1.8.19
2. librării și surse 3rd-party:
  - Adafruit\_GFX.h (librarie 3rd-party pentru controlul unui display SPI OLED)
  - Adafruit\_SSD1306.h (librarie 3rd-party speciala pentru acest tip de display)
3. funcții:
  - funcție de decodificare a literelor după inputul primit sub formă de cod morse
  - filtrare a intensității batailor pentru a obține caractere din alfabetul morse ('.' sau '-')


## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

## Concluzii

În urma realizării acestui proiect pot spune că înțeleg mai bine felul în care funcționează microprocesoarele și componentele care intră în alcatuirea dispozitivelor, majoritatea învățăturilor trase fiind datorate erorilor care mi-au dat batai de cap. În ciuda acestora, mi-a făcut plăcere să ajung la acest rezultat. }

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC creează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

[cristian\\_apostol\\_decoder\\_morse.zip](#)

## Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

## Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[How to use Adafruit displays](#)

[Secret Knock Detector](#)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/amocanu/cristian.apostol>



Last update: **2022/06/01 08:52**