

Translator de cod Morse

Autor:

Găvenea Maria-Laurenția

Introducere

Proiectul presupune transmisia codificării Morse a unui cuvânt de la o plăcuță (ce primește informația de la utilizator prin intermediul unor butoane) la altă plăcuță, ce decodifică mesajul și îl afișează la un ecran LCD.

Scopul acestui proiect este acela de a învăța codul Morse într-un mod distractiv. Chiar dacă nu prea mai este folosit în ziua de azi, este un exercițiu bun de memorie și nu se știe niciodată când vei avea nevoie să știi să transmiți ... _ _ _ ...!

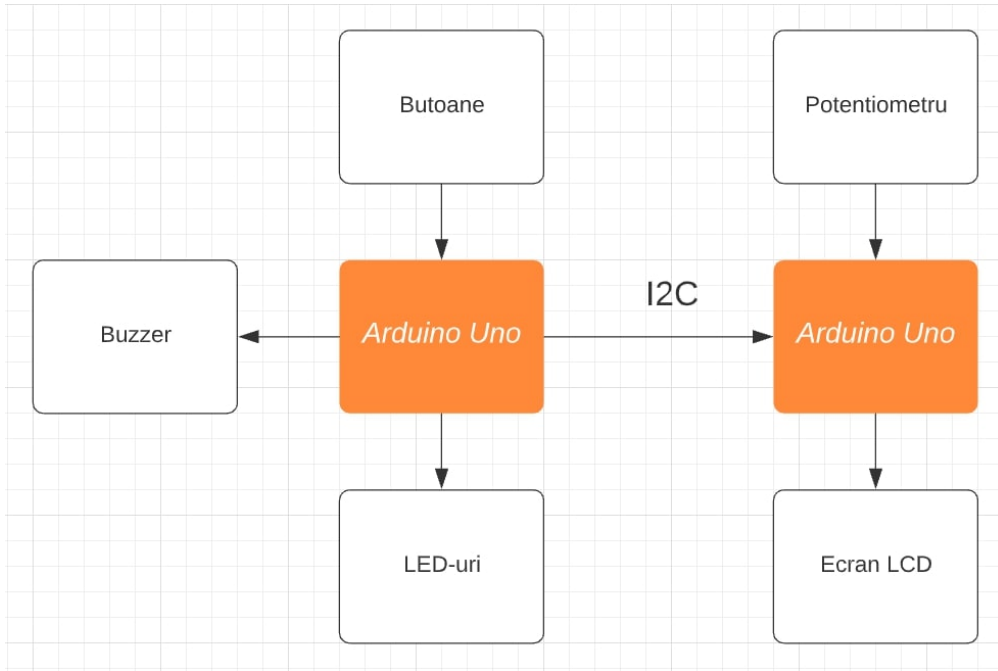
Descriere generală

Proiectul presupune comunicarea prin cod morse între două plăcuțe Arduino Uno, una având rol de transmițător și cealaltă de receptor.

Plăcuța transmițător are patru butoane cu următoarele roluri:

1. un buton pentru linie - la apăsarea acestuia se va declanșa buzzer-ul
2. un buton pentru punct - la apăsarea acestuia se va declanșa buzzer-ul
3. un buton care indică final de caracter - cât timp este apăsat va fi aprins LED-ul roșu
4. un buton care indică final de cuvânt
 1. cât timp este apăsat va fi aprins LED-ul verde
 2. în urma apăsării acestui buton va fi trimis cuvântul către cealaltă plăcuță

Plăcuța receptor va descifra mesajul primit și îl va afișa pe un ecran LCD. Potențiometrul reglează luminozitatea ecranului LCD.

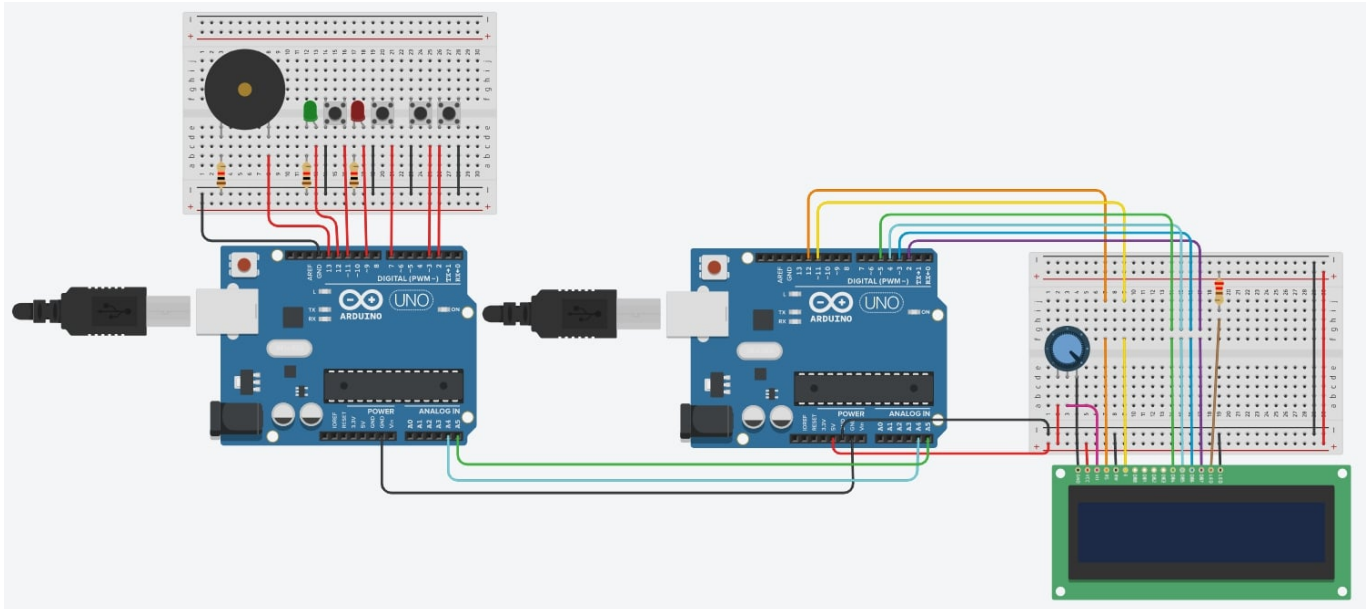


Hardware Design

Componente necesare:

Componentă	Număr
Arduino Uno	2
Breadboard	2
LED roșu	1
LED verde	1
Buzzer	1
Butoane	4
Rezistență 1k	3
Rezistență 220	1
Potențiomtru	1
LCD 1602	1

Schema electrică:



Software Design

Am folosit ca mediu de dezvoltare Arduino IDE.

Biblioteci utilizate: Wire.h (I2C), LiquidCrystal.h (LCD)

Plăcuța transmițător

1. Butoanele

Fiecare apăsare de buton generează o întrerupere:

- butonul pentru linie → întrerupere INT0
- butonul pentru punct → întrerupere INT1
- butonul pentru final de caracter → întrerupere PCINT2
- butonul pentru final de cuvânt → întrerupere PCINT0

Am făcut debouncing pentru fiecare buton folosind funcția millis() și acceptând intervalul minim între două apăsări succesive ca fiind de 300 ms.

2. Buzzer-ul

Există trei situații în care buzzer-ul sună:

- a) a fost apăsat butonul pentru linie → frecvență: 525 HZ și timp: 800 ms
- b) a fost apăsat butonul pentru punct → frecvență: 525 HZ și timp: 400 ms
- c) se încearcă depășirea dimensiunii buffer-ului de transmisie → frecvență: 1500 HZ și timp: 500 ms (în acest caz tot buffer-ul este invalidat, nu se trimite mesajul și o scriere următoare se va face de la începutul buffer-ului)

Timpul și frecvența sunt setate în ISR-uri, iar buzzer-ul este activat cu funcția tone() în bucla principală.

3. LED-urile

LED-urile sunt aprinse cât timp este apăsat butonul corespunzător fiecăruia.

- a) LED-ul roșu → este aprins și stins în rutina de tratare a întreruperii PCINT2
- b) LED-ul verde → este aprins și stins în rutina de tratare a întreruperii PCINT0

4. Comunicarea cu plăcuța receptor

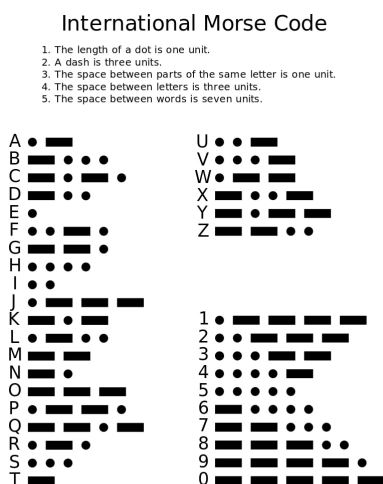
Pentru comunicarea prin I2C m-am folosit de biblioteca Wire.h.
După ce sunt transmise datele către cealaltă plăcuță, se afișează mesajul transmis la monitorul serial.

Plăcuța receptor

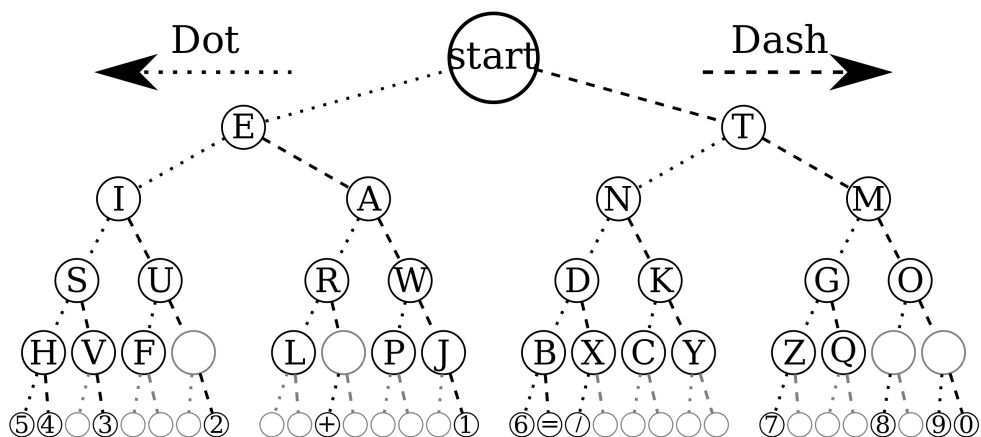
După ce a primit tot cuvântul, începe să îl decodifice.

1. Decodificare mesaj primit

Codurile acceptate sunt doar cele pentru litere și cifre. Acestea pot fi regăsite în imaginea de mai jos:



Pentru a realiza decodificarea mesajului am creat un arbore Morse pornind de la imaginea de mai jos:



Am reprezentat arborele printr-un array în care am pus fiecare caracter pe poziția sa, iar pe pozițiile libere am pus caracterul '?':

```
char morseTree[] = "?ETIANMSURWDKGOHV?L?PJBXCYZQ??54?3???2????????16????????7???8?90\0";
```

Funcția de căutare în arbore am realizat-o în următorul fel:

```
char decode_character(char *ch) {
    int len = strlen(ch);

    if (len <= 5) {
        int index = 0;

        for (int i = 0; i < len; ++i) {
            if (ch[i] == '.') {
                // deplasare in subarborele stang
                index = 2 * index + 1;
            } else if (ch[i] == '-') {
                // deplasare in subarborele drept
                index = 2 * index + 2;
            } else {
                // codificare gresita
                return '?';
            }
        }

        return morseTree[index];
    }

    // codifica gresita
    return '?';
}
```

În șirul de intrare:

- când este găsit un punct, se continuă căutarea în subarborele stâng
- când este găsită o linie, se continuă căutarea în subarborele drept
- dacă este găsit orice alt caracter sau dimensiunea șirului primit este mai mare ca 5 (înălțimea arborelui) atunci codificarea este greșită și funcția întoarce caracterul '?'

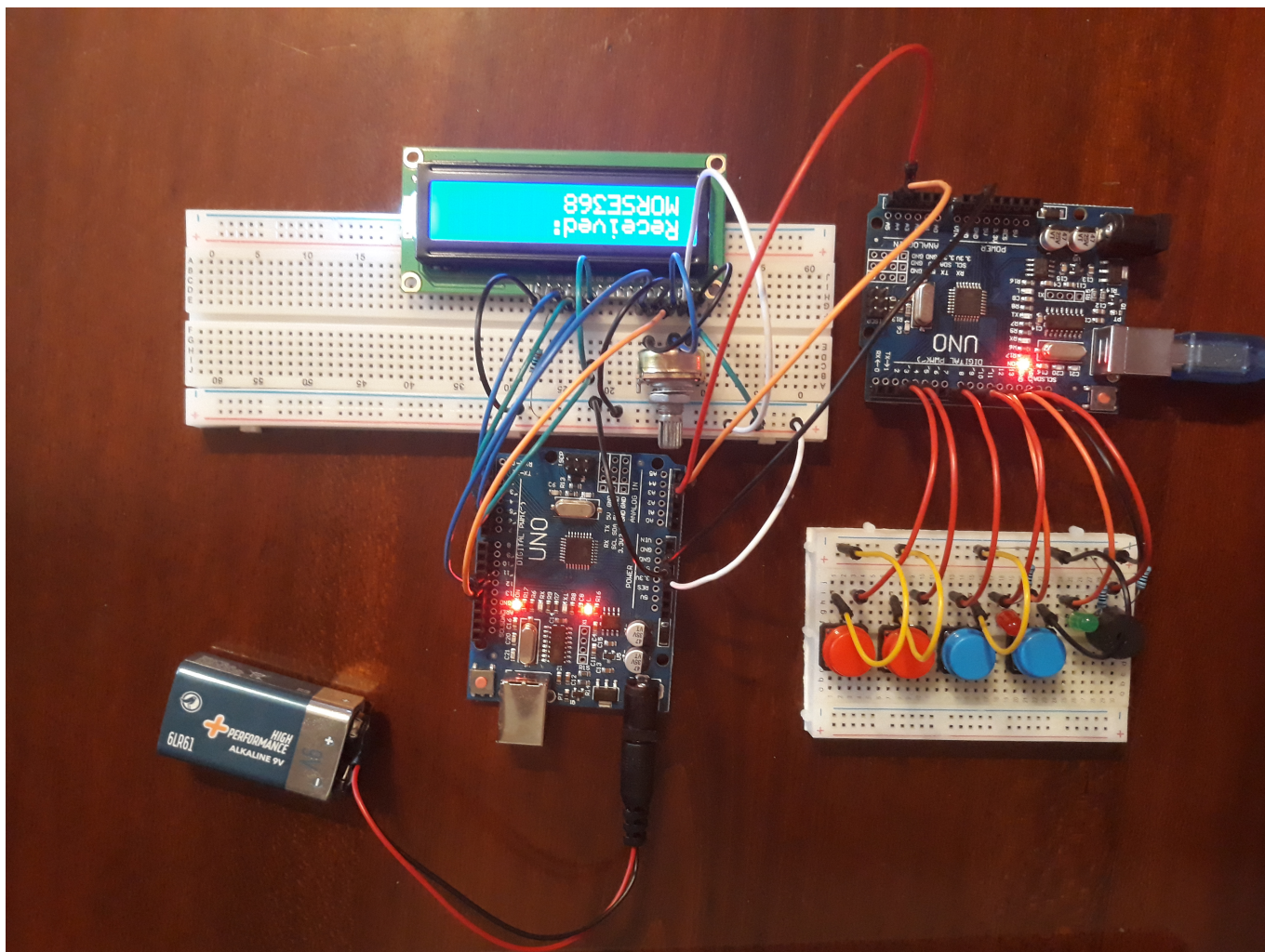
În caz de succes, funcția întoarce ultimul caracter la care a ajuns în arbore după ce a terminat de parcurs șirul primit ca parametru.

2. Afișare mesaj decodificat

Pentru afișarea la LCD a mesajului final m-am folosit de biblioteca LiquidCrystal.h. Pe măsură ce sunt decodificate caracterele, sunt afișate la LCD. De fiecare dată când este primit un cuvânt, acesta se afișează în locul celui vechi.

Rezultate Obținute

Circuit final



Demo

Link YouTube: <https://youtu.be/RQwSEu8Wrus>

Concluzii

Faptul că am realizat proiectul fizic și nu într-un simulator m-a făcut să mă lovesc și de alte probleme (de exemplu: în cazul butoanelor a trebuit să fac debouncing, la LCD trebuiau lipiți pinii etc).

Per total, proiectul a fost o experiență plăcută și interesantă în care am aprofundat mai bine niște noțiuni, am dat de probleme noi și am învățat și puțin din codul Morse.

Download

[translator_morse_surse.zip](#)

Jurnal

25.04.2021 - Publicare pagină, realizare schemă electrică și schemă bloc

07.05.2021 - Configurare plăcuță transmițător

26.05.2021 - Configurare plăcuță receptor și realizare comunicare între plăcuțe

Bibliografie/Resurse

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Debounce>

<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>

<https://electrosome.com/interfacing-lcd-arduino-uno/>

<https://www.electronicsforu.com/technology-trends/learn-electronics/16x2-lcd-pinout-diagram>

<https://www.arduino.cc/en/reference/wire>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/abasoc/translatorcodmorse>



Last update: **2021/06/02 12:26**