

Signal Corps - Old School Morse Code Trainer

Introducere

Signal Corps Trainer este un dispozitiv embedded de antrenament si decodare a codului Morse, construit pe placa ATmega328P-XMINI si o cheie telegrafică vintage sovietică ca interfata de intrare.

Ce face proiectul:

- Utilizatorul trimite cod Morse apasand cheia telegrafică, iar sistemul decodează caracterele in timp real si le afisează pe LCD
- In modul Trainer, sistemul propune o litera aleatorie si verifica daca a fost transmisa corect
- In modul Decoder, orice semnal primit este interpretat si afişat fara restrictii
- Pe ecranul OLED se vizualizeaza forma de unda a semnalului
- Buzzerul si LED-urile ofera feedback audio-vizual imediat

Scopul proiectului este sa transforme o cheie telegrafică intr-un instrument functional de invatare. Codul Morse nu tine doar de istorie: este folosit in radioamatorism, aviatie civila si comunicatii de urgenta.

Fata de aplicatiile software de invatare Morse, acest dispozitiv ofera senzatia mecanica reala a cheii si sunetul pe care il auzeau operatorii acum cateva decenii. Combinatia dintre feedback fizic si audio face memorarea mai rapida decat invatatul vizual.

Descriere Generala

Schema bloc:



Descrierea modulelor:

- **ATmega328P-XMINI** ruleaza intreaga logica: cronometreaza durata semnalelor audio captate de microfon, decodifica caracterele Morse, actualizeaza afisajele si controleaza buzzerul si LED-urile. Programarea se face direct prin USB via mEDBG, fara programator extern.
- **LCD 1602 (I2C, 0x27)** afiseaza caracterele decodate si starea curenta a sistemului (modul activ, scor, erori).
- **OLED 0.96" SSD1306 (I2C, 0x3C)** arata forma de unda a semnalului in timp real, cu scroll orizontal, si animatii la fiecare raspuns.

- **Senzorul KY-037** este sursa principala de input in modul Trainer. Ascuta tonul emis de buzzer si converteste sunetul intr-un semnal analogic pe A0, pe baza caruia se face decodarea Morse.
- **Cheia telegrafică** este conectata la D2 si actioneaza buzzerul cat timp e apasata. Sistemul nu citeste cheia direct pentru decodare, ci o foloseste doar pentru a porni sursa audio captata apoi de microfon.
- **Butonul SW2 pe D4** comuta intre modul Trainer si modul Decoder la fiecare apasare.
- **Buzzerul piezo pasiv pe D3** scoate un ton de 2500 Hz cat timp cheia e apasata, folosind functia tone(). Frecventa de 2500 Hz a fost aleasa pentru a coincide cu rezonanta naturala a buzzerului piezo, ceea ce maximizeaza volumul si permite captarea sigura prin microfon.
- **LED verde** confirma un raspuns corect in modul Trainer, **LED rosu** semnalizeaza o greseala sau zgomot detectat.

Hardware Design

Stadiul Actual al Implementarii

In acest moment hardware-ul este complet asamblat pe breadboard. Toate componentele sunt fizic conectate la placa ATmega328P-XMINI, alimentarea functioneaza la 5V prin USB, iar magistrala I2C raspunde corect la ambele dispozitive (LCD si OLED). Schema electrica a fost desenata in Autodesk Fusion Electronics si verificarea ERC a trecut fara erori critice.

Pana acum au fost testate individual:

- Cheia telegrafică, verificata cu multimetrul si conectata la D2
- Buzzerul piezo pe D3, cu generare de ton de 2500 Hz prin functia tone()
- Senzorul KY-037 pe A0, calibrat pentru detectia tonului emis de buzzer
- Magistrala I2C cu LCD-ul 1602 raspunzand la adresa 0x27 si OLED-ul SSD1306 la 0x3C
- Butonul de mod pe D4, citit cu INPUT_PULLUP
- LED-urile verde si rosu pe D5 si D6, prin rezistori de 220Ω

In primele iteratii software, decodarea Morse se baza pe citirea directa a cheii pe pinul digital D2 cu intrerupere INT0. Pe parcurs s-a renuntat la aceasta abordare in favoarea captarii audio prin microfon, ceea ce face proiectul mai apropiat de un decodor real CW si valorifica mult mai bine senzorul KY-037 (ADC). Cheia ramane conectata, dar rolul ei este acum doar acela de a actiona buzzerul.

Componente Folosite si Rolul Lor

Referinta	Componenta	Rol in proiect
U1	ATmega328P-XMINI	Microcontroler principal, ruleaza logica de decodare Morse si controleaza toate perifericele
U2	LCD 1602 cu interfata I2C	Afisaj text pentru caracterele decodate si mesaje de stare

U3	OLED 0.96" SSD1306	Afisaj grafic pentru forma de unda a semnalului si animatii de feedback
SEN1	Senzor sunet KY-037	Sursa principala de input audio, capteaza tonul buzzerului pentru decodarea Morse
SW1	Cheie telegrafică vintage	Actioneaza buzzerul, declansand emisia tonului audio
SW2	Push-button	Comuta intre modul Trainer si modul Decoder
BZ1	Buzzer Piezo Pasiv	Genereaza tonul de 2500 Hz captat apoi de microfon
LED1	LED Verde 5mm	Confirma vizual raspunsurile corecte in modul Trainer
LED2	LED Rosu 5mm	Indica vizual erorile sau prezenta zgomotului ambiental
R1, R2	Rezistor 220Ω	Limiteaza curentul prin LED-uri, protejand pinii MCU si diodele

Pinii Folositi si Justificarea Alegerii

Toti pinii se refera la placa ATmega328P-XMINI. Alimentarea la 5V vine direct prin USB, nu e nevoie de sursa externa.

Pin MCU	Tip	Conectat la	De ce acest pin
PD2	Digital	Cheie telegrafică	Citit cu INPUT_PULLUP pentru a actiona buzzerul atunci cand cheia e apasata. Initial folosit pentru intrerupere INT0, dar dupa schimbarea logicii software s-a renuntat la intrerupere
PD3	PWM	Buzzer Piezo Pasiv	Pin cu PWM hardware, ideal pentru functia tone() care genereaza forma de unda audio la 2500 Hz
PD4	Digital	Buton selector mod	Pin digital simplu, butonul nu are nevoie de PWM sau intrerupere, foloseste INPUT_PULLUP
PD5	Digital	LED Verde prin R1 (220Ω)	Pin digital pentru control on/off al LED-ului, lipit langa PD4 pentru cablaj simplu
PD6	Digital	LED Rosu prin R2 (220Ω)	Acelasi rationament ca la PD5, grupare logica a iesirilor digitale
PC0	ADC	KY-037 iesire analogica (AO)	Singurul tip de pin care poate citi valoarea continua a semnalului acustic, prin analogRead(). Esential pentru decodarea Morse din semnalul captat
PC4	SDA (I2C)	LCD + OLED (magistrala partajata)	Pin dedicat hardware pentru SDA pe ATmega328P, suporta TWI nativ
PC5	SCL (I2C)	LCD + OLED (magistrala partajata)	Pin dedicat hardware pentru SCL, pereche cu PC4 pentru I2C
5V	Power	Alimentare toate componentele	Tensiune standard pentru toate modulele alese
GND	Power	Masa comuna	Referinta de zero pentru intregul circuit

Alegerile de pini au tinut cont de trei criterii: functionalitatea hardware speciala (PWM pentru buzzer, ADC pentru senzor, TWI pentru afisaje), gruparea logica pentru cablaj curat pe breadboard (PD3-PD6 in zona switch-urilor si LED-urilor), si compatibilitatea cu magistrala I2C dedicata (PC4/PC5). Decizia de a folosi microfonul ca sursa principala de input face ca pinul ADC (PC0) sa devina critic pentru functionarea sistemului, nu doar auxiliar.

Note Tehnice

Magistrala I2C este partajata intre LCD si OLED. Coexistenta functioneaza fara probleme deoarece cele doua dispozitive au adrese diferite: LCD = 0x27, OLED = 0x3C.

- SW1 si SW2 folosesc pull-up intern (INPUT_PULLUP). Rezistenta de 10kΩ externa este optionala si nu a fost montata
- Intreg sistemul e alimentat la 5V prin USB de pe placa X-MINI
- Pini AREF, ADC6 si ADC7 au fost marcati explicit ca No-Connect in schema Fusion pentru a evita erori ERC
- Buzzerul si microfonul sunt plasate aproape unul de altul pe breadboard (la 2-3 cm distanta) pentru a asigura captarea fiabila a tonului
- Frecventa de 2500 Hz pentru buzzer a fost aleasa dupa testare experimentală: la 700 Hz volumul era prea mic pentru captarea sigura prin microfon, iar 2500 Hz corespunde rezonantei naturale a piezo-ului si ofera volum maxim

Schema Electrica

Schema electrica a fost realizata in Autodesk Fusion si include toate componentele cu conexiunile corespunzatoare. Verificarea regulilor electrice (ERC) a fost trecuta fara erori critice.

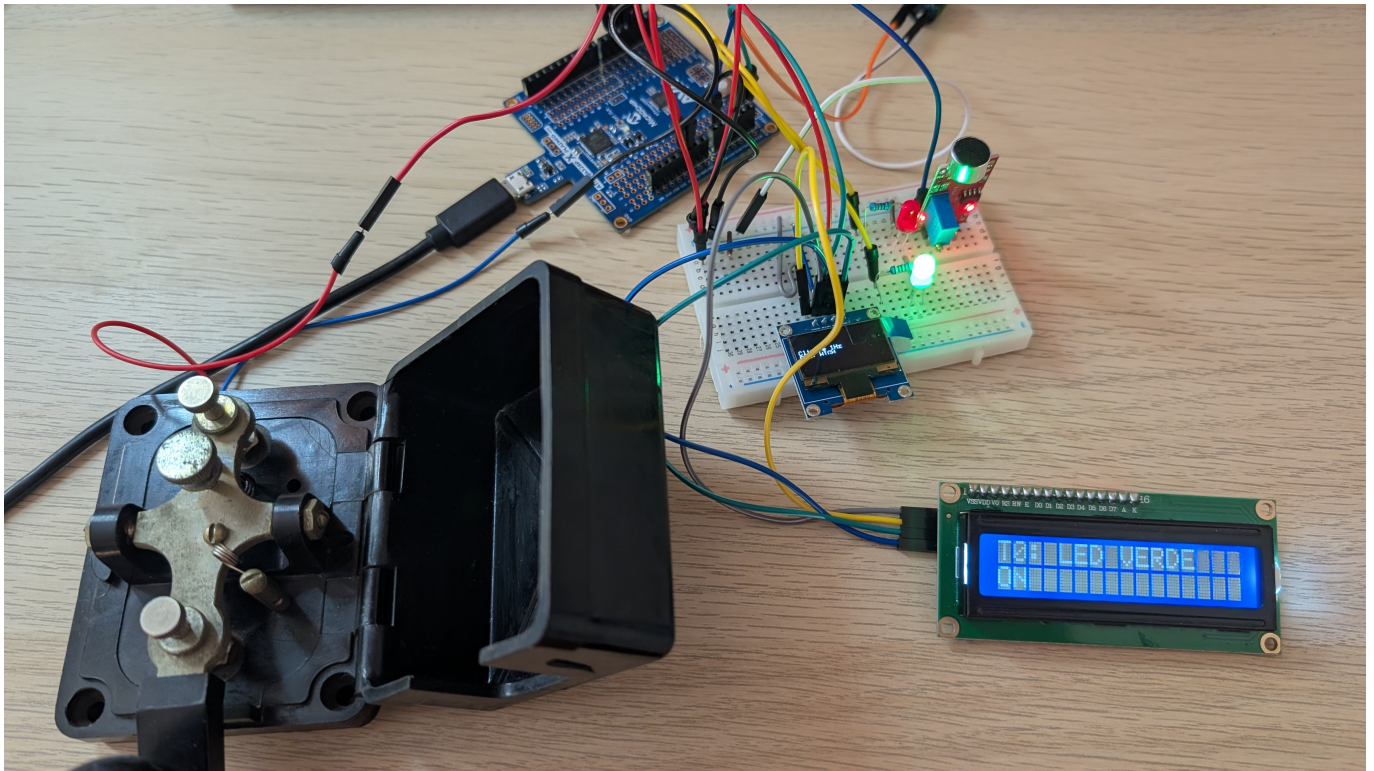


Explicatii ale schemei:

- Blocul central este placa ATmega328P-XMINI cu cei trei pini de alimentare (VCC, VCC_2, AVCC) legati la +5V, plus pini GND legati la masa comuna
- AREF, ADC6, ADC7 sunt marcati ca No-Connect deoarece nu sunt utilizati in proiect
- Magistrala I2C (PC4 SDA si PC5 SCL) iese din MCU si se ramifica in paralel catre LCD si OLED, ambele alimentate la 5V
- Sensorul KY-037 este reprezentat printr-un header cu 4 pini (VCC, GND, DO, AO) cu doar AO conectat la PC0
- Cheia telegrafică (S1) si butonul de mod (S2) sunt switch-uri momentar normal-deschise, cu un terminal la pinul MCU si celalalt la GND
- LED-urile sunt conectate prin rezistori de 220Ω in serie, anozii la (PD5, PD6) si catodii la GND
- Buzzerul piezo este conectat intre PD3 si GND, fara componente suplimentare necesare

Imagini cu Implementarea

Asamblarea pe breadboard cu placa ATmega328P-XMINI:



Sistemul montat in carcasa finala:





Demo video al modului Trainer in functiune:

Software Design

Stadiul Actual al Implementarii Software

Software-ul ruleaza pe ATmega328P-XMINI cu Arduino IDE si este complet functional in modul Trainer. Codul implementeaza si modul Decoder, comutarea intre cele doua moduri se face din butonul SW2 conectat la PD4. Cele doua afisaje sunt folosite cu roluri distincte: LCD-ul 1602 prezinta modul curent si scorul intr-un format stabil, iar OLED-ul afiseaza continutul dinamic (litera curenta, secventa Morse transmisa, osciloscop al semnalului audio, animatii de feedback).

Logica decodarii Morse functioneaza pe baza captarii audio prin microfonul KY-037. Sistemul masoara durata fiecarui impuls sonor, clasifica semnalul ca dot (sub 250 ms) sau dash (250-1500 ms), iar dupa o pauza de 800 ms decodeaza secventa intr-un caracter. Validarea raspunsului in modul Trainer se face prin comparare directa cu codul Morse al literei cerute.

Mediu de Dezvoltare

- Arduino IDE cu board package "ATmega328P Xplained Mini" (Microchip)
- Programare prin interfata mEDBG (debugger on-board) via USB, fara programator extern
- Monitor serial la 9600 baud pentru depanare in timp real
- Schema electrica realizata in **Autodesk Fusion Electronics**

Biblioteci Folosite si Motivatia Alegerii

Biblioteca	Versiune	Rol	Motivul alegerii
Wire.h	nativa	Comunicatie I2C cu LCD si OLED	Biblioteca standard Arduino, suporta hardware TWI al ATmega328P fara configurare suplimentara
Adafruit_SSD1306	2.5.x	Driver pentru ecranul OLED	Suport oficial pentru chip-ul SSD1306, are functii de desenare grafica si gestionare buffer
Adafruit_GFX	1.11.x	Grafica primitiva (linii, cercuri, text)	Dependentă a Adafruit_SSD1306, ofera primitive grafice (cercuri, dreptunghiuri) pentru animatii
LiquidCrystal_I2C	1.1.2	Driver pentru LCD 1602 cu modul I2C	Compatibila cu modulul PCF8574 lipit pe LCD, are functii simple (print, setCursor, backlight)

Toate bibliotecile sunt instalate prin Library Manager din Arduino IDE. Motivul principal al alegerii: sunt cele mai stabile si testate biblioteci pentru hardware-ul ales, cu documentatie buna si comunitate activa care raspunde la probleme.

Elementul de Noutate

Proiectul are doua elemente care il diferentiaza de un trainer Morse clasic:

- Captarea semnalului prin microfon, nu prin citire digitala a cheii. Aceasta abordare face dispozitivul sa functioneze ca un decodor CW real, capabil sa interpreteze orice sursa audio de tonuri Morse (radio CW, alta cheie, telefon care reda sunete), nu doar propria cheie. Senzorul KY-037 nu mai este auxiliar, ci sursa principala de input.
- Integrarea unei chei telegrafice vintage sovietice ca interfata haptica. Spre deosebire de aplicatiile software de invatare Morse care folosesc tastatura sau ecran tactil, dispozitivul ofera senzatiile mecanice reale a tastarii pe o cheie istorica, ceea ce schimba complet experienta de invatare.

In plus, cele doua ecrane (LCD si OLED) sunt folosite pentru roluri distincte si complementare, nu duplicate, ceea ce ofera o experienta de utilizare mai bogata.

Justificarea Functionalitatilor din Laborator

Proiectul integreaza in mod natural elemente din mai multe laboratoare:

- **Laboratorul 0 (GPIO):** LED-urile verde (PD5) si rosu (PD6) sunt comandate prin digitalWrite() pentru feedback vizual. Butoanele (PD2 cheia, PD4 selector mod) sunt citite prin digitalRead() cu INPUT_PULLUP pentru a folosi rezistentele de pull-up interne ale MCU.
- **Laboratorul 1 (UART):** comunicatia seriala la 9600 baud este folosita pentru debugging pe parcursul dezvoltarii: afisarea valorilor ADC din microfon, log-uri pentru fiecare dot/dash detectat, mesaje de stare la schimbarea modului, raportarea rezultatelor de validare (CORECT/GRESIT).
- **Laboratorul 3 (Timere si PWM):** functia tone() folosita pentru buzzer se bazeaza pe Timer2 al

ATmega328P, care genereaza un semnal de frecventa specifica (2500 Hz) prin PWM pe PD3. Acelasi mecanism de timing software (millis()) este folosit pentru a cronometra durata fiecarui semnal Morse si a clasifica dot/dash.

- **Laboratorul 4 (ADC):** senzorul KY-037 este citit prin analogRead() pe pinul A0 (PC0). Conversia analog-digital de 10 biti ofera o rezolutie suficienta pentru a detecta varfurile de amplitudine ale tonului audio fata de zgomotul ambiental. Citirea se face in burst (16 esantioane in 4 ms) pentru a prinde varful unei sonore.
- **Laboratorul 6 (I2C):** magistrala I2C este partajata intre LCD (0x27) si OLED (0x3C). Cele doua dispozitive coexista pe acelasi bus SDA/SCL fara conflicte, deoarece au adrese hardware diferite. Wire.h gestioneaza automat protocolul TWI.

Scheletul Proiectului

Codul este organizat in jurul unei masini de stare cu doua niveluri:

- Nivel exterior (Mod aplicatie): MOD_TRAINER si MOD_DECODER, comutate de butonul SW2 cu debounce software
- Nivel interior (Stare in cadrul Trainer): STARE_LLECTIE, STARE_FEEDBACK_CORECT, STARE_FEEDBACK_GRESIT, STARE_TIMEOUT

Funcția principală loop() apelează, în ordine:

1. Citirea cheii pe PD2 pentru a acționa buzzerul (tone/noTone pe PD3)
2. Verificarea butonului de mod pe PD4 cu debounce de 250 ms
3. Rutarea către logica modului curent (Trainer sau Decoder)
4. Animația LCD la fiecare 800 ms (puncte și linii rotative pe margini)

Interacțiunea dintre componente:

1. **Cheia - Buzzer:** legatura directa software, fara intarziere
2. **Buzzer - Microfon:** legatura acustica prin aer, captata cu KY-037
3. **Microfon - MCU:** prin ADC pe PC0 cu citire in burst
4. **MCU - Afisaje:** prin I2C, fiecare ecran are propria functie de update
5. **MCU - LED-uri:** prin GPIO digital pentru feedback de rezultat

Validarea funcționării a fost făcută în trei etape: testare individuală (fiecare componentă separat), testare integrată (toate componentele împreună fără logica Morse) și testare end-to-end (modul Trainer complet cu cheie, buzzer, microfon, decodare și verificare). Fiecare etapă a folosit Serial Monitor pentru a verifica ca valorile transmise corespund cu cele așteptate.

Calibrarea Senzorului KY-037

Calibrarea microfonului este critică pentru funcționarea decodării, deoarece nivelul de baseline (zgomot ambiental) și pragul de detecție variază de la o sesiune la alta în funcție de mediu.

Procesul de calibrare rulează automat la pornire și durează 3 secunde:

1. In aceasta perioada, codul colecteaza 100 de esantioane din microfon (citire in burst de 16 esantioane fiecare pentru a prinde varful)
2. Calculeaza baseline-ul ca medie a celor 100 de varfuri
3. Identifica varful maxim observat in liniste
4. Seteaza pragul de detectie ca: $\text{pragVariatie} = (\text{varfMax} - \text{baselineMic}) + 10$

Pragul este astfel garantat sa fie peste orice zgomot ambiental existent in momentul calibrarii, plus o marja de siguranta. Adaptarea continua a baseline-ului in timpul rularii (filtru exponential cu 5% pondere pentru valori noi) ajuta sistemul sa reziste la zgomote graduale (ventilator, conversatii in fundal).

Hardware-ul a fost calibrat in paralel: potentiometrul albastru de pe placa KY-037 a fost rotit pana cand LED-ul Status integrat reactioneaza la buzzer dar nu la vorbire normala. Aceasta ajusteaza sensibilitatea analogica a microfonului inainte de procesarea ADC.

Optimizari Realizate

Pe parcursul dezvoltarii au fost necesare cateva optimizari pentru a face sistemul stabil:

1. Optimizare RAM prin char[] in loc de String

ATmega328P are doar 2 KB de RAM. Versiunea initiala folosea obiecte String din Arduino, care alocau dinamic memorie pentru fiecare litera primita. Combinatia cu buffer-ul OLED (1024 bytes pentru 128x64 pixeli) ducea la stack overflow si initializarea OLED-ului esua silentios.

Solutia: inlocuirea tuturor String-urilor cu char[] de dimensiune fixa (secventaMorse[9], textDecodat[25], morseAsteptat[8]) si folosirea functiilor C standard (strcmp, strcpy_P, strlen). Economie estimata: ~400 bytes RAM.

2. Tabel Morse in PROGMEM

Tabelul cu cele 26 de coduri Morse era stocat in RAM. Mutarea lui in Flash memory (PROGMEM, 32 KB disponibili) elibereaza inca ~150 bytes RAM. Citirea se face cu `pgm_read_word()` si `strcpy_P()` la nevoie.

3. Esantionare in burst pentru ADC

In loc de un singur `analogRead()` pe iteratie, codul citeste 16 esantioane in 4 ms si pastreaza maximul. Pentru un ton de 2500 Hz cu perioada de 400 microsecunde, asta inseamna ca se acopera cel putin 10 cicluri complete ale undei sonore, garantand detectia varfului. Fara aceasta optimizare, detectia era nesigura.

4. Histerezis pe detectia sunetului

Detectia incepe la `prag_ON` (variatie peste prag) si se opreste la `prag_OFF` (`prag/2`). Aceasta marja evita oscilatii rapide la limita pragului, care altfel ar genera dot-uri si dash-uri false la inceputul si sfarsitul fiecarui sunet.

5. Verificare stare inainte de actualizare OLED

Bug subtil care a aparut: in switch-ul de stare al modului Trainer, dupa `procesezaSemnalMicrofon()` si

ruleaza_lectie() (care poate schimba starea la STARE_FEEDBACK_CORECT), codul continua si apeleaza actualizeazaOLED_Trainer(). Aceasta suprascrisa imediat mesajul de feedback (bifa pentru corect, X pentru gresit) cu litera lectiei. Mesajul de feedback aparea doar cateva milisecunde inainte sa fie inlocuit.

Solutia: verificarea starii curente inainte de a desena ecranul de lectie, pentru a nu suprascrisa ecranele de feedback:

```
if (stareTrainer == STARE_LECTIE) {  
    actualizeazaOLED_Trainer();  
}
```

6. Initializare I2C cu retry si viteza redusa

OLED-urile SSD1306 ieftine uneori esueaza la prima incercare de initializare cu viteza I2C standard (400 kHz). Codul reduce explicit viteza la 100 kHz prin `Wire.setClock(100000)` si reincearca initializarea de pana la 5 ori cu pauze de 300 ms intre incercari. Fara aceasta optimizare, dispozitivul nu pornea consistent la fiecare alimentare.

Rezultate Obtinate

- Cheia telegrafică acționează buzzerul fără latență perceptibilă
- Buzzerul generează 2500 Hz sincronizat cu apăsarea cheii, la volum suficient pentru captare prin microfon
- Senzorul KY-037 detectează tonul buzzerului prin ADC și permite decodarea Morse pe baza duratei apăsărilor
- OLED-ul afișează litera curentă și secvența Morse transmisă în timp real
- LCD-ul arată scorul și starea sistemului
- Magistrala I2C cu două dispozitive la adrese 0x27 și 0x3C funcționează stabil
- LED-urile verde și roșu răspund corect pe D5 și D6 pentru feedback de răspuns corect sau greșit
- Comutarea între modul Trainer și modul Decoder prin butonul de pe PD4 funcționează cu debounce software
- Sistemul rulează stabil ore întregi fără reinitializări sau pierderi de conexiune I2C

Concluzii

Proiectul a pornit de la o cheie telegrafică autentică și a ajuns un dispozitiv funcțional de învățare. Ce e interesant e că obiectul fizic schimbă complet experiența față de o simplă aplicație: senzația mecanică și sunetul buzzerului fac învățatul mai intuitiv.

Pe parcurs au apărut câteva probleme neașteptate, în special cu configurarea driver-ului mEDBG în Arduino IDE și cu calibrarea timpilor pentru dot/dash la viteze diferite de tastare. Schimbarea de la

decodare digitala prin cheie la decodare prin microfon a fost o decizie luata in timpul implementarii, dupa testare: solutia audio s-a dovedit mai aproape de un decodor CW real si valorifica mai bine senzorul KY-037. Partajarea magistralei I2C intre doua dispozitive a necesitat atentie la adrese, dar a functionat fara modificari hardware.

Cele mai mari provocari software au fost gestionarea memoriei RAM limitate de 2 KB pe ATmega328P si descoperirea bug-ului cu suprascrierea OLED-ului la feedback (care arata corect pentru cateva milisecunde si dispare). Ambele au fost rezolvate prin optimizari descrise in sectiunea Software Design.

Dificultati intampinate:

- Configurarea board-ului ATmega328P-XMINI in Arduino IDE si instalarea driver-ului mEDBG
- Partajarea magistralei I2C fara conflicte de adrese
- Calibrarea pragurilor de timing dot/dash in functie de viteza de tastare
- Reglarea potentiometrului de pe KY-037 si gasirea frecventei optime pentru buzzer (2500 Hz in loc de 700 Hz initial) pentru a asigura captare audio fiabila
- Optimizarea consumului de RAM prin inlocuirea String cu char[] si mutarea tabelului Morse in PROGMEM
- Identificarea si rezolvarea bug-ului de race condition la afisarea feedback-ului pe OLED

Download

Arhiva proiectului(github) contine:

- Codul sursa complet (.ino)
- Schema electrica
- Fisier README.md cu instructiuni de instalare si utilizare
- Link github <https://github.com/rodion-bl/Signal-Corps-Morse-Code-Trainer-ATmega328p>

Jurnal

- **25.04.2026** - Comanda componentelor
- **26-30.04.2026** - Saptamana de asteptare a livrarii, citire datasheet-uri
- **02.05.2026** - Primire componente, verificare individuala
- **03.05.2026** - Test cheie telegrafică cu multimetru si lipirea a 2 cabluri cu ciocan de lipit
- **04.05.2026** - Setup Arduino IDE, driver mEDBG, primul upload functional
- **05.05.2026** - Asamblare breadboard, test cheie + buzzer + OLED
- **06.05.2026** - Integrare LCD pe I2C, verificare coexistenta cu OLED

- **07.05.2026** - Documentatie pe wiki, schema bloc draw.io
- **13.05.2026** - Realizare schema electrica completa in Fusion Electronics, verificare ERC
- **15.05.2026** - Trecere de la decodare digitala prin cheie la decodare prin microfon, ajustare frecventa buzzer la 2500 Hz
- **16.05.2026** - Filmare demo Trainer si actualizare documentatie cu noua logica audio
- **20.05.2026** - Implementare mod Decoder si comutare prin butonul SW2 pe PD4
- **22.05.2026** - Optimizare RAM: inlocuire String cu char[], tabel Morse in PROGMEM
- **23.05.2026** - Rezolvare bug suprascriere OLED la feedback, marire timp de afisare a rezultatului
- **24.05.2026** - Montare in carcasa finala, filmare demo complet

Bibliografie / Resurse

Resurse Hardware

- [ATmega328P-XMINI User Guide - Microchip Technology](#)
- [SSD1306 OLED Driver Datasheet - Solomon Systech](#)
- [KY-037 Sound Sensor Module - Arduino Modules Info](#)

Resurse Software

- [Adafruit SSD1306 Library - GitHub](#)
- [Adafruit GFX Library - GitHub](#)
- [LiquidCrystal_I2C Library - GitHub](#)
- [Arduino Language Reference - arduino.cc](#)
- [Morse Code Timing Standard - ITU-R M.1677-1](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/cezar.zlatea/rodion.balaniuc>



Last update: **2026/05/25 14:55**