DashDot Morse Coach

1. Introducere

DashDot Morse Coach este un sistem hardware interactiv dedicat învățării și perfecționării codului Morse, conceput pentru a transforma experiența tradițională de telegrafie într-un proces de instruire dinamic și modern. Sistemul funcționează ca un antrenor digital capabil să traducă în timp real semnalele introduse prin apăsări de buton în text alfanumeric, oferind utilizatorului un feedback complex: auditiv prin buzzer, vizual prin afișajul LCD și mecanic prin intermediul unui vitezometru analogic.

Scopul principal al proiectului este de a facilita memorarea alfabetului Morse prin combinarea sunetului, a textului digital și a mișcării mecanice, mutând accentul de la o învățare pasivă la o experiență practică, unde utilizatorul este motivat constant de indicatorii de performanță. Prin calcularea vitezei de scriere (WPM), sistemul permite monitorizarea progresului în timp real, oferind pasionaților un prim contact ideal cu telegrafia și o modalitate de perfecționare a ritmului de comunicare în codul Morse.

2. Descriere generală

Funcționarea sistemului **DashDot Morse Coach** se bazează pe preluarea semnalelor transmise manual de utilizator și transformarea acestora în informații digitale procesabile prin intermediul unei mașini de stări. Microcontrolerul monitorizează constant starea butoanelor de input, calculând cu precizie durata fiecărei apăsări pentru a distinge între elementele fundamentale ale codului (punctul și linia) și pentru a permite navigarea între cele două moduri principale de operare:

- **Modul PRACTICE:** Un mediu de antrenament liber unde utilizatorul poate exersa alfabetul Morse, primind feedback instantaneu pentru fiecare caracter. Se accesează din ecranul principal prin apăsarea butonului NEXT.
- **Modul EXAM:** Un mod de testare structurat unde sistemul solicită scrierea unui cuvânt selectat aleatoriu dintr-un dicționar de 10 cuvinte. Sistemul oferă un verdict final de tip Pass/Fail bazat pe acuratețea mesajului introdus orbește. Se accesează din ecranul principal prin apăsarea butonului DONE.

Interacțiunea dintre modulele proiectului este organizată astfel:

- **Modulul de Input (3x Butoane)**: Reprezintă calea de comunicare a utilizatorului cu sistemul.
  - Butonul MORSE (PD2): Utilizează întreruperi externe pentru captarea exactă a momentelor de apăsare și eliberare, asigurând o precizie ridicată în cronometrarea semnalelor.
  - Butonul NEXT (PD4): Utilizat pentru selecția modului Practice din meniul principal și pentru salvarea fiecărui caracter decodificat în timpul exersării.
  - Butonul DONE (PD7): Utilizat pentru selecția modului Exam din meniul principal și pentru transmiterea comenzii de finalizare a sesiunii.

- **Unitatea de Procesare (ATmega328P):** „Creierul” proiectului care măsoară intervalele de timp, gestionează algoritmul de decodificare și calculează în timp real viteza de scriere (WPM), determinând nivelul de performanță.
- **Feedback-ul Vizual (LCD 1602 I2C):** Afișează textul tradus, instrucțiunile de navigare pentru modurile de lucru și statisticile de performanță la finalul fiecărei sesiuni.
- **Feedback-ul Sonor (Buzzer):** Generează tonuri specifice codului Morse sincronizate perfect cu butonul, ajutând la formarea reflexului auditiv necesar unui telegrafist.
- **Indicatorul Mecanic (Servomotor):** Acționează ca un vitezometru analogic. Poziția brațului servomotorului indică vizual ritmul de scriere, oferind o confirmare fizică rapidă a progresului și constanței.
- **Sistemul de Semnalizare (3x LED)**: Oferă feedback de stare instantaneu: LED-ul Galben confirmă înregistrarea caracterelor, în timp ce LED-urile Verde și Roșu semnalizează verdictul „PASS” sau „FAIL” la finalul examinării.

3. Hardware Design

Componentele fizice utilizate pentru construirea antrenorului DashDot Morse Coach sunt următoarele:

Componentă	Specificații Tehnice	Rol în Proiect
Microcontroler	ATmega328P (XMINI)	Unitatea centrală de procesare; gestionează logica de decodificare și interfețele.
Display	LCD 1602 I2C	Feedback vizual principal: afișează textul tradus și statisticile de performanță.
Servomotor	Micro SG90	Indicator mecanic analogic pentru vizualizarea vitezei de scriere (WPM).
Buzzer	Pasiv 3.3V	Feedback sonor sincronizat cu apăsările, ajutând la formarea memoriei auditive.
Butoane(3x)	Push-button tactil	PD2 (Morse), PD4 (Next/Practice), PD7 (Done/Exam).
LED-uri(3x)	Verde, Roșu, Galben	Semnalizare vizuală pentru Pass (PB0), Fail (PB1) și înregistrare (PB3).
Rezistențe	220 Ohm / 10k Ohm	Protecția componentelor (LED/Buzzer) și limitarea curentului.
Cabluri	Jumper Wires	Realizarea conexiunilor electrice între periferice și placa de dezvoltare.

3.1. Schema Electrică

4. Software Design

Descrierea codului aplicaţiei (firmware)

Firmware-ul aplicației gestionează în timp real achiziția de semnale, decodificarea caracterelor Morse, controlul perifericelor de feedback și navigarea într-o structură ierarhică de meniuri. Sistemul este asincron, bazându-se pe mecanisme de întreruperi pentru a elimina latențele de procesare și pentru a asigura o precizie maximă în cronometrarea impulsurilor.

Din punct de vedere didactic, structura software-ului integrează conceptele fundamentale din trei laboratoare distincte, aplicate simultan pe lângă funcțiile de bază de I/O digital (GPIO):

- **Laboratorul de Întreruperi Externe:** Folosit pentru citirea asincronă a butonului Morse pe pinul PD2, eliminând tehnica ineficientă de polling.
- **Laboratorul de Timere și PWM:** Utilizat dublu — Timer 0 controlează baza de timp în milisecunde și generează semnalul acustic (PWM) pe PD5, iar Timer 1 generează un semnal Fast PWM de 50 Hz pentru poziționarea servomotorului.
- **Laboratorul de Protocoale de Comunicație Serie (I2C):** Implementează controlul ecranului LCD 1602 prin magistrala TWI (Hardware I2C) la adresa 0x27, reducând numărul de conexiuni fizice la doar două fire (SDA/SCL).

Mediu de dezvoltare

- **Microchip Studio / PlatformIO:** Utilizat pentru scrierea codului în limbaj C pur (fără framework-ul Arduino), compilare cu setul de unelte avr-gcc și încărcarea binarului pe microcontroler prin programatorul integrat mEDBG al plăcii Xplained Mini.

Biblioteci şi surse 3rd-party folosite

Pentru a asigura o eficiență maximă și un consum minim de memorie, proiectul este dezvoltat nativ, fără utilizarea unor biblioteci externe sau framework-uri complexe. Controlul perifericelor se realizează prin configurarea directă a regiștrilor microcontrolerului, fiind utilizate doar bibliotecile standard incluse în compilatorul avr-gcc:

- **<avr/io.h>:** Pentru maparea și controlul direct al regiștrilor hardware ai ATmega328P.
- **<avr/interrupt.h>:** Pentru gestionarea vectorilor de întrerupere hardware.
- **<util/delay.h>:** Pentru generarea întârzierilor calibrate necesare protocolului de inițializare a ecranului.
- **<string.h> și <stdio.h>:** Utilizate pentru manipularea bufferelor de text și formatarea indicatorilor pe LCD.

Algoritmi şi structuri implementate

- **Mașină de Stări Finită (Finite State Machine - FSM):** Logica aplicației este ghidată de o structură cu 4 stări distincte:

- **WELCOME:** Starea inițială de repaus ce afișează meniul principal și așteaptă selecția utilizatorului.
- **PRACTICE:** Modul de antrenament liber în care caracterele sunt decodificate și concatenate la mesaj în urma confirmării manuale (Butonul NEXT).
- **EXAM:** Modul de testare cu cerință extrasă dinamic și evaluare finală.
- **RESULT:** Starea de calcul și afișare a statisticilor de performanță.

- **Algoritmul de Decodificare (String Matching):** Semnalele (puncte și lines) sunt colectate într-un buffer de caractere (morse_buffer). La apăsarea butonului NEXT, acest buffer este comparat prin funcția strcmp cu șabloanele standard din alfabetul internațional Morse.
- **Calculul Vitezei WPM (Words Per Minute):** Implementat eficient prin aritmetică pe numere întregi pentru a evita utilizarea resurselor de tip float:

$$WPM = \frac{char_count \times 12}{t_total}$$

(Notă: Formula derivă din standardul internațional Paris (standardul Morse), unde un cuvânt standard are o lungime medie de 5 caractere. Constanta 12 reprezintă simplificarea matematică a raportului dintre cele 60 de secunde dintr-un minut și cele 5 caractere per cuvânt ($\frac{60}{5} = 12$), asigurând un calcul rapid pe 8 biți).

(Etapa 3) Surse şi funcţii implementate

Listă funcții critice

Funcție	Tip Returnat	Descriere și Rol Hardware
`I2C_init()`, `I2C_start()`, `I2C_stop()`, `I2C_write()`	`void`	Driverul software nativ pentru protocolul TWI (I2C) la frecvența de 100 kHz.
`LCD_init()`, `LCD_command()`, `LCD_print()`	`void`	Inițializează ecranul în mod de control pe 4 biți și gestionează afișarea textului.
`set_servo_angle(uint8_t angle)`	`void`	Modifică registrul `OCR1B` pentru a varia lățimea pulsului PWM (Timer 1) între 1 ms și 2 ms.
`decode_morse(char* cod)`	`char`	Returnează litera corespunzătoare șirului de puncte și linii sau `?` pentru cod invalid.
`hardware_init()`	`void`	Configurează direcțiile pinilor (DDR), rezistențele pull-up și timerele hardware.

Gestionarea Întreruperilor (ISR)

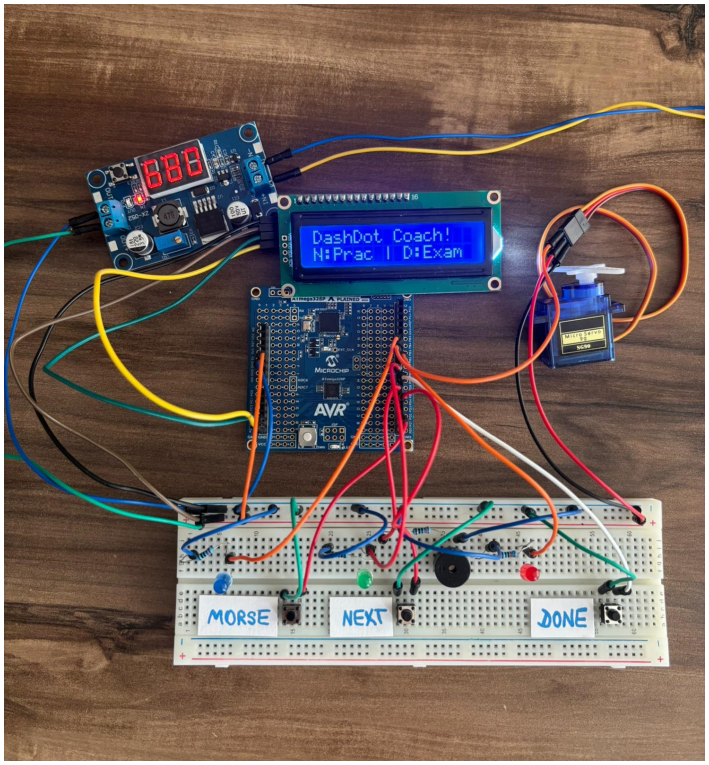
- **`ISR(TIMER0_OVF_vect)`:** Incrementat la fiecare revărsare a Timerului 0, generând o bază de timp precisă în milisecunde (`ms_ticks`) folosită la debouncing și cronometrare.
- **`ISR(INT0_vect)`:** Declanșat de butonul MORSE pe pinul PD2. La apăsare (Transition Low), activează starea pinului de ieșire și cuplează pinul PD5 la Timer 0 (`COM0B0`) pentru a genera sunetul pe buzzer, salvând totodată timestamp-ul actual. La eliberare (Transition High), decuplează modulul de sunet și calculează durata totală a impulsului. Dacă valoarea depășește pragul de

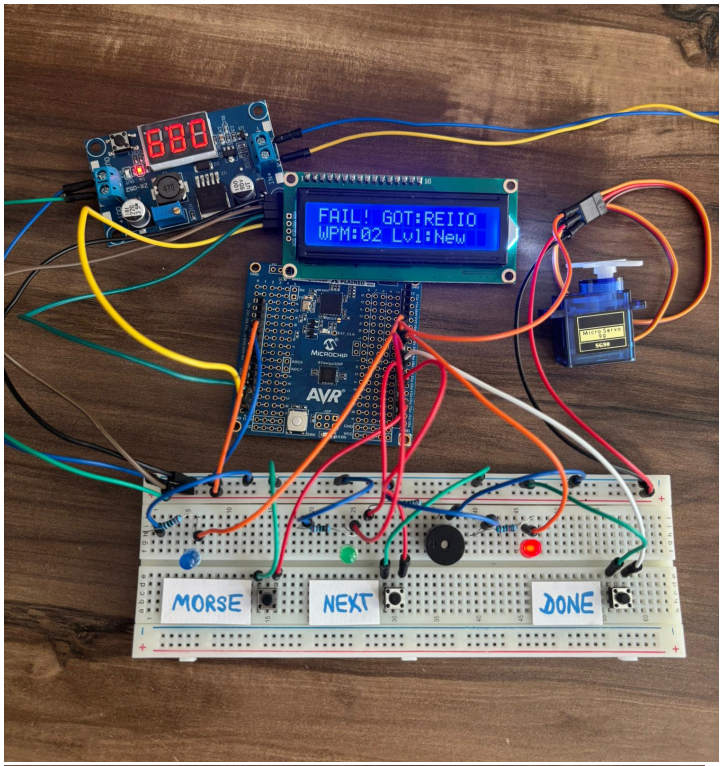
debouncing de 50 ms, decide dacă semnalul a fost un punct (`.`) sau o linie (`-`) pe baza pragului critic de 250 ms.

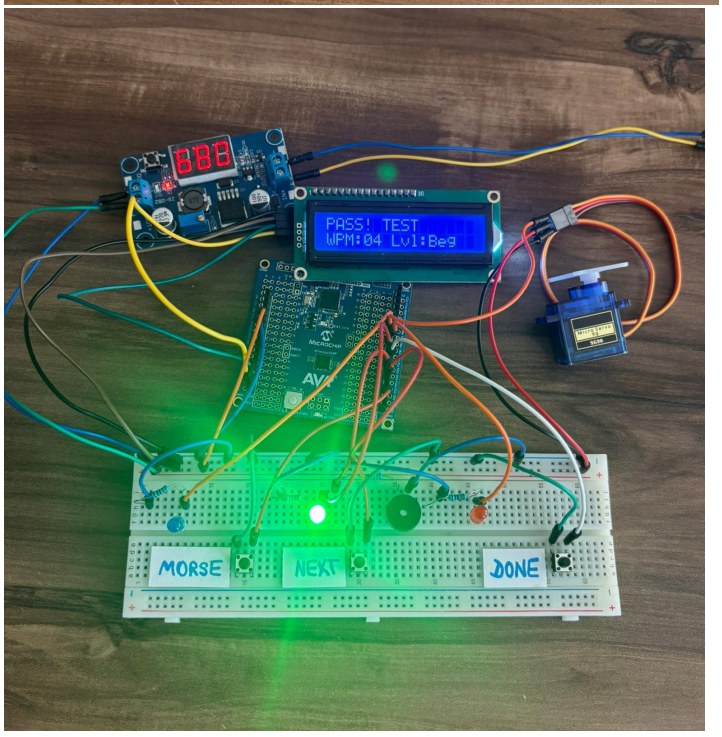
5. Rezultate Obţinute

În urma implementării hardware și software, sistemul DashDot Morse Coach este complet funcțional și rulează conform specificațiilor, oferind o experiență interactivă și fluidă.

Mai jos sunt prezentate imaginile cu cele trei stări principale ale proiectului (Meniul de pornire, Afișarea erorilor în caz de FAIL și Validarea succesului în caz de PASS):







6. Concluzii

A fost un proiect foarte tare de realizat și chiar mi-a plăcut mult să lucrez la el și să mă joc cu el. Pe parcursul dezvoltării, am aprofundat implementarea unei mașini de stări (FSM) și lucrul direct cu timerele și întreruperile pe microcontrolerul ATmega328P.

7. Jurnal

- **09.05.2026** - Milestone 1: Finalizarea conceptului “DashDot Morse Coach”, stabilirea listei de hardware necesar și structurarea documentației.
- **16.05.2026** - Milestone 2: Asamblarea circuitului pe breadboard, testarea individuală a perifericelor (comunicația I2C pentru LCD, semnalul PWM pentru Servo) și implementarea primei versiuni a decodificatorului Morse.
- **24.05.2026** - Milestone 3: Optimizarea majoră a software-ului prin eliminarea codului blocant de debounce la butoanele de control. Adaugarea dicționarului de examen cu 10 cuvinte dinamice și adăugarea diagnosticului de erori direct pe ecranul LCD.
- **25.05.2026** - Milestone 4: Calibrarea finală a formulei pentru WPM și a pragurilor nivelurilor de viteză, testarea integrată a tuturor funcționalităților și finalizarea paginii de documentație pe OCW.

8. Bibliografie/Resurse

- Datasheet ATmega328P.
- Specificații protocol I2C pentru modulele PCF8574.
- Standardul internațional pentru timpii în Codul Morse.

Export to PDF

From:
https://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
https://ocw.cs.pub.ro/courses/pm/prj2026/bianca.popa1106/alexandra.misloschi

Last update: **2026/05/25 09:08**