

Mini Osciloscop Digital cu Generator de Semnal

Introducere

- Proiectul constă în realizarea unui **osciloscop digital standalone**, capabil să captureze și să afișeze semnale analogice în timp real, cu un generator intern de semnal și intrare externă prin jack 3.5mm.
- **Scopul lui:** vizualizarea și analiza semnalelor electrice la un cost accesibil, fără echipamente comerciale costisitoare.
- **Ideea de la care am pornit:** Sincer să fiu nici eu nu știu, pur și simplu am avut o iluminare
- **Utilitate:** pentru alții, un instrument funcțional sub 300 RON care înlocuiește un osciloscop entry-level. Pentru mine, o integrare completă a tuturor temelor PM — ADC, SPI, I2C, timere și întreruperi — într-un singur sistem hardware coerent.

Descriere generală



Osciloscopul este organizat în jurul a cinci module hardware și trei module software, toate coordonate de ATmega328P.

2.A Module hardware:

Semnalul analogic intră prin **mufa jack PJ392 3.5mm**, trece prin două **diode 1N4148** care îl protejează, apoi prin **op-amp-ul MCP6002** configurat ca buffer, și în final printr-un **filtru RC** care elimină zgomotul de înaltă frecvență înainte de ADC.

Placa ATmega328P Xplained Mini este centrul proiectului. Rulează la 16MHz, se programează direct prin USB și coordonează toate perifericele — ADC, SPI, I2C, timere și întreruperi.

Display-ul ILI9341 de 2.4" se conectează prin SPI și afișează forma de undă în timp real împreună cu scalele și măsurătorile automate. Deoarece operează la 3.3V, **level shifter-ul Pololu 2595** face conversia de tensiune față de MCU-ul de 5V.

DAC-ul MCP4921 se conectează pe același bus SPI și generează semnalul intern de test (sinus, triunghi, dreptunghi), care se întoarce la intrarea analogică formând o buclă de demo completă.

Modulul EEPROM AT24C256 se conectează prin I2C și salvează setările utilizatorului între sesiuni.

2.B Module software:

Sampling engine — Timer1 în mod CTC cu polling, fără ISR, fără trigger detection. Capturează 240 eșantioane cu perioadă fixă de 800μs/eșantion.

Display renderer — desenează coloană cu coloană grila 6×6, cursorul vertical și semnalul. Panou de statistici cu Vpp, Vavg, T/div și tensiunea la cursor.

Measurement engine — calculează Vpp și Vavg din buffer.

Hardware Design

Listă componente:

Componentă	Model complet	Cantitate	Rol	Preț estimat
Placă MCU	ATmega328P Xplained Mini	1	MCU central	De la PM lab
Display TFT	Ecran LCD ILI9341, cu slot pentru Card SD, 2.4"	1	Afișare semnal	~67 RON
DAC	MCP4921-E/P — DIP8 12-bit SPI	1	Generator semnal	~12 RON
Op-Amp	MCP6002-I/P — DIP8 dual rail-to-rail	1	Buffer intrare analogică	~2 RON
EEPROM	Modul AT24C256 I2C	1	Salvare setări	~7 RON
Level Shifter	Pololu 2595 — 4ch bidirecțional	1	5V↔3.3V SPI	~16 RON
Diode	1N4148 DO-35	2	Clamp tensiune 0-5V	~1 RON
Rezistori	1kΩ, 2kΩ, 3.3kΩ, 10kΩ — 1/4W	~25	Pull-up, filtru RC, divizori	~5 RON
Condensatori	100nF ceramic, 10nF ceramic, 10μF electrolitic	~15	Decoupling, filtru RC	~20 RON
Mufă jack	PJ392 3.5mm stereo mamă	1	Intrare semnal extern	~3 RON
Potențiometru	WH148 10kΩ liniar	1	Cursor de măsurare	~2 RON
Butoane	PCB 12×12×7.3mm + capace colorate	6+4	Interfață utilizator	~10 RON
Breadboard	830 puncte	1-2	Prototipare	Disponibil
Fire	Set jumper 140 fire + dupont M-F 40 fire	1+1	Conexiuni	~20 RON

Total estimat: ~165 RON

Modul EEPROM AT24C256 (I²C)

Pin modul	Pin ATmega328P	Funcție	Justificare
VCC	+5V	Alimentare	Aceeași tensiune ca MCU, evită level shifter pe liniile I ² C.
GND	GND	Masă	Referință comună obligatorie.
SDA	PC4	I ² C Data	Singurul pin hardware SDA al perifericului TWI din ATmega328P.

SCL	PC5	I ² C Clock	Singurul pin hardware SCL al perifericului TWI din ATmega328P.
-----	-----	------------------------	--

Display LCD ILI9341 2.4" cu slot SD (SPI)

Pin modul	Conectat la	Funcție	Justificare
VCC	+5V	Alimentare modul	Modulul are regulator on-board, acceptă 3.3-5V direct.
GND	GND	Masă	Referință comună obligatorie.
CS	Pololu 2595 — L3 (← H3 ← PD7)	Chip Select display	GPIO liber, ales pentru a păstra PB2 (SS hardware) pentru CS_DAC.
RESET	+3.3V cu pull-up 10kΩ	Reset hardware display	Reset-ul se face software prin comandă SPI 0x01, economisind un canal de level shifter.
DC/RS	Pololu 2595 — L4 (← H4 ← PD6)	Data/Command select	GPIO liber pe PORTD, comutat la fiecare transfer SPI între comandă și date.
SDI (MOSI)	Pololu 2595 — L2 (← H2 ← PB3)	SPI Master Out	Pin hardware MOSI al perifericului SPI din ATmega328P, fixat de arhitectură.
SCK	Pololu 2595 — L1 (← H1 ← PB5)	SPI Clock	Pin hardware SCK al perifericului SPI, fixat de arhitectură.
LED	+3.3V	Backlight	Alimentare directă pentru backlight, mereu pornit (poate fi controlat ulterior prin PWM).
SDO (MISO)	neconectat	SPI Master In	Display-ul nu este citit în acest proiect, economisind un canal de level shifter.
SD_CS	neconectat	Chip Select card SD	Slot-ul SD nu este folosit în acest proiect.
SD_MOSI	neconectat	SPI Master Out (SD)	Slot-ul SD nu este folosit în acest proiect.
SD_MISO	neconectat	SPI Master In (SD)	Slot-ul SD nu este folosit în acest proiect.
SD_SCK	neconectat	SPI Clock (SD)	Slot-ul SD nu este folosit în acest proiect.

Level Shifter Pololu 2595 (4 canale bidirecțional)

Pin shifter	Conectat la	Funcție	Justificare
HV	+5V	Alimentare partea înaltă	Tensiunea logică a MCU-ului (ATmega328P la 5V).
H1	PB5 (SCK MCU)	Canal 1 — intrare 5V SCK	Linia SCK SPI vine de la MCU la 5V și trebuie shiftată la 3.3V pentru display.
H2	PB3 (MOSI MCU)	Canal 2 — intrare 5V MOSI	Linia MOSI SPI vine de la MCU la 5V și trebuie shiftată la 3.3V pentru display.
H3	PD7 (CS_TFT MCU)	Canal 3 — intrare 5V CS	Selectarea display-ului trebuie shiftată la 3.3V pentru a nu deteriora ILI9341.
H4	PD6 (DC MCU)	Canal 4 — intrare 5V DC	Linia data/command trebuie shiftată la 3.3V, altfel se depășește tensiunea maximă a pinilor ILI9341.
LV	+3.3V	Alimentare partea joasă	Tensiunea logică a display-ului ILI9341 (3.3V).
L1	SCK display	Canal 1 — ieșire 3.3V SCK	SCK shiftat ajunge la display la nivel logic sigur.
L2	MOSI display	Canal 2 — ieșire 3.3V MOSI	MOSI shiftat ajunge la display la nivel logic sigur.

L3	CS display	Canal 3 — ieșire 3.3V CS	CS shiftat ajunge la display la nivel logic sigur.
L4	DC display	Canal 4 — ieșire 3.3V DC	DC shiftat ajunge la display la nivel logic sigur.

Amplificator Operațional MCP6002 (buffer intrare analogică)

Pin op-amp	Conectat la	Funcție	Justificare
1 (VOUTA)	R2 → A0 (PC0)	Ieșire canal A	Semnalul buffer-at iese către filtrul RC anti-aliasing și apoi către ADC-ul MCU.
2 (VINA-)	VOUTA (feedback)	Intrare inversoare canal A	Conectată direct la ieșire pentru configurație voltage follower (gain = 1).
3 (VINA+)	Semnal de la jack (după clamp diode)	Intrare neinversoare canal A	Primește semnalul de măsurat, izolând sursa de impedanța joasă a ADC.
4 (VSS)	GND	Masă	Alimentare negativă (single supply, deci GND).
5 (VINB+)	GND	Intrare neinversoare canal B	Legat la GND pentru a termina corect canalul B neutilizat și a preveni oscilații.
6 (VINB-)	VOUTB (pin 7)	Intrare inversoare canal B	Legat la propria ieșire pentru a închide bucla canalului B neutilizat.
7 (VOUTB)	VINB- (pin 6)	Ieșire canal B	Neutilizată, închisă în feedback unity gain pentru stabilitate.
8 (VDD)	+5V (cu 100nF decoupling)	Alimentare pozitivă	Alimentare single supply 5V, comună cu MCU-ul, cu condensator de decuplare lângă pin.

DAC MCP4921 (generator semnal SPI 12-bit)

Pin DAC	Conectat la	Funcție	Justificare
1 (VDD)	+5V (cu 100nF decoupling)	Alimentare	Aceeași tensiune cu MCU, permite comunicație SPI directă fără level shifter.
2 (CS)	PB2	Chip Select	Pin SS hardware al ATmega328P, dedicat selectării DAC-ului pe bus-ul SPI.
3 (SCK)	PB5	SPI Clock	Pin hardware SCK al perifericului SPI, partajat cu display-ul (CS-uri diferite separă device-urile).
4 (SDI)	PB3	SPI Master Out	Pin hardware MOSI al perifericului SPI, primește comanda de 16 biți de la MCU.
5 (LDAC)	GND	Latch DAC	Legat la GND pentru actualizare imediată a ieșirii după fiecare scriere SPI.
6 (VREFA)	+5V	Referință tensiune	Permite ieșirii DAC să atingă tot domeniul 0-5V (rezoluție ~1.22 mV/bit).
7 (AVSS)	GND	Masă analogică	Referință comună pentru partea analogică a DAC-ului.
8 (VOUTA)	DAC_OUT → jack intrare	Ieșire analogică	Semnalul generat (sinus, triunghi, dreptunghi) se întoarce către intrarea osciloscopului pentru bucla de demo.

Butoane PCB (interfață utilizator)

Toate butoanele folosesc aceeași topologie: un terminal la GND, celălalt la pinul MCU prin rezistor de pull-up extern către VCC. Pin-ul MCU citește LOW când butonul este apăsat.

Buton	Pin ATmega328P	Funcție	Justificare
HOLD	PD2	Run/Stop captură	Pin INT0 — întrerupere externă dedicată pentru reacție instantă la oprirea capturii.
MODE	PD3	Ciclare formă de undă	Pin INT1 — întrerupere externă dedicată pentru schimbare rapidă a modului.
TIME_P	PD4	Mărire time/div	Pin GPIO cu suport PCINT20 pentru debouncing prin întrerupere.
TIME_M	PD5	Micșorare time/div	Pin GPIO cu suport PCINT21, alăturat lui TIME_P pentru rutare ușoară.
VOLT_P	PB0	Mărire volt/div	GPIO liber pe PORTB, neutilizat de SPI sau ADC.
VOLT_M	PB1	Micșorare volt/div	GPIO liber pe PORTB, alăturat lui VOLT_P pentru rutare ușoară.

Potențiometrul WH148 10kΩ (cursor de măsurare)

Pin pot.	Conectat la	Funcție	Justificare				
1	GND	Capăt minim	Stabilește limita inferioară a domeniului de tensiune (0V).				
2 (wiper)	PC1 (A1)	Tensiune variabilă	Pin ADC1 al MCU citește tensiunea wiper-ului ca cursor de măsurare pe axa X.	3	+5V	Capăt maxim	Stabilește limita superioară a domeniului (5V), egală cu VREF al ADC.

Mufă Jack PJ-392 3.5mm Stereo (intrare semnal extern)

Pin mufă	Conectat la	Funcție	Justificare
1 (Sleeve)	GND	Masă comună	Referință de tensiune comună între sursa semnalului și circuit.
2 (Ring)	GND	Canal stereo secundar	Legat la masă pentru a evita zgomot indus din canalul drept neutilizat.
3 (Tip)	Diode clamp 1N4148 → VINA+ (MCP6002 pin 3)	Semnal de măsurat	Intră semnalul analogic care trece prin protecție și buffer înainte de ADC.

Pin MCU	Funcție	Periferic
PB0	VOLT_P (buton)	GPIO
PB1	VOLT_M (buton)	GPIO
PB2	CS_DAC	SPI SS
PB3	MOSI	SPI
PB5	SCK	SPI
PC0	Semnal intrare	ADC0

PC1	Cursor de măsurare	ADC1
PC4	SDA	I ² C
PC5	SCL	I ² C
PD2	HOLD	INT0
PD3	MODE	INT1
PD4	TIME_P	GPIO/PCINT
PD5	TIME_M	GPIO/PCINT
PD6	DC display	GPIO
PD7	CS_TFT	GPIO



Descrierea schemei electrice

Schema este organizată în jurul plăcii ATmega328P Xplained Mini, reprezentată ca header 1×20 cu toate semnalele etichetate funcțional.

Lanțul de intrare analogică începe în mufa jack PJ-392, unde semnalul trece prin două diode 1N4148 montate ca dublu clamp (protecție 0–5V), apoi prin rezistorul R1 care limitează curentul. Op-amp-ul MCP6002 este configurat ca voltage follower pe canalul A, buffer-ând semnalul către filtrul RC anti-aliasing format din R2 și C2 (10nF). Ieșirea filtrului ajunge la pinul ADC A0 (PC0) al MCU.

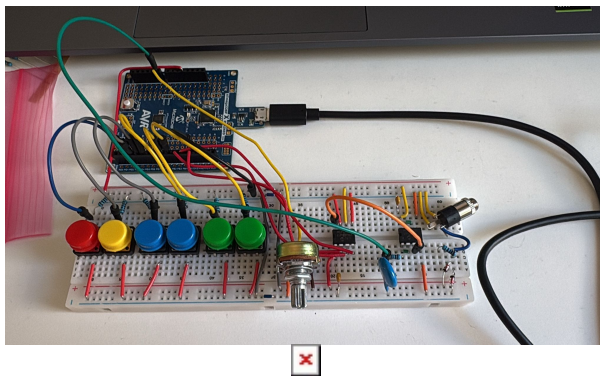
Bus-ul SPI este partajat între display-ul ILI9341 și DAC-ul MCP4921. SCK și MOSI provin de la MCU, iar selectarea device-ului activ se face prin Chip Select separat (CS_DAC pe PB2, CS_TFT pe PD7). DAC-ul, alimentat la 5V, comunică direct; display-ul, cu logică 3.3V, primește semnalele prin level shifter-ul Pololu 2595 (conector 1×8 dedicat).

Bus-ul I²C conectează modulul EEPROM AT24C256 (header 1×4 cu pull-up-uri integrate pe modul) la pinii SDA/SCL ai MCU.

Interfața utilizator conține șase butoane (HOLD, MODE, TIME±, VOLT±) cu pull-up extern (R5–R10), și un potențiomtru WH148 de 10kΩ (R11) conectat ca divizor între VCC și GND, cu wiper-ul ducând la ADC1 (PC1) pentru controlul cursorului de măsurare pe axa X a formei de undă.

Decuplarea alimentării este asigurată de condensatoarele C1 și C3 (100nF) montate fizic lângă pinii VDD ai op-amp-ului și DAC-ului.

Progres Actual(pentru Milestone 2)



Dovada a functionalitatii componentelor

Codul rulează simultan 5 module hardware care comunică între ele. DAC-ul (prin SPI) generează un semnal digital, care iese pe pinul 8 ca tensiune analogică. Această tensiune trece prin op-amp ca buffer, ajunge la pinul ADC, este eșantionată și raportată pe UART. În paralel, butoanele (GPIO) schimbă forma de undă generată, iar potențiometrul (alt canal ADC) reglează amplitudinea. Faptul că valorile DAC și ADC afișate pe Serial Monitor sunt aproape identice (diferență sub 20mV) confirmă că întregul lanț — SPI → DAC → op-amp → ADC → UART — funcționează corect.

Aceasta este “bucă de demo” a osciloscopului — DAC-ul produce semnal de test, iar restul lanțului îl procesează exact ca pentru un semnal real venit din jack. Înlocuind sursa (DAC → jack cu intrare externă), arhitectura rămâne identică. Practic, am validat hardware-ul de procesare a semnalului înainte să-l alimentez cu semnal real, ceea ce înseamnă că proiectul are deja partea cea mai critică funcțională: achiziția de date analogice prin op-amp + ADC, coordonată de MCU prin SPI și raportată prin UART.

O problemă pe care am întâmpinat-o este faptul că display-ul pe care l-am luat inițial era cu interfață paralelă, nu SPI, și nu mi-am dat seama de la început. Când am ajuns să îl conectez, am realizat că nu îl pot folosi și a trebuit să dau comandă pentru alt display compatibil SPI. Acesta este motivul pentru care la momentul de față nu îl am integrat în proiect. Problema se va rezolva în curând — display-ul ajunge pe 18 mai, abia atunci voi putea finaliza proiectul complet din punct de vedere hardware.

Software Design

Stadiul actual al implementării software

Firmware-ul este complet funcțional și rulează pe ATmega328P Xplained Mini la 16 MHz, scris în C pur (fără Arduino HAL), compilat cu avr-gcc prin PlatformIO.

Funcționalități implementate:

- Generator de semnal intern (DAC MCP4921 pe SPI): sinusoidă, triunghi, dreptunghi
- Captură de semnal analogic prin ADC intern (10-bit, canal 0, $V_{ref} = 5V$)

- Afişare în timp real pe display ILI9341 2.4" TFT (SPI) la ~1 fps
- Grilă osciloscop 6x6 (240x216 px zona de semnal, 0.83V/div)
- Cursor de măsurare controlat cu potențiometrul (poziție continuă pe ecran)
- Panou de statistici: Vpp, Vavg, T/div, tensiune la cursor
- 6 butoane cu întreruperi hardware (fără polling):
 - **MODE** (INT1): ciclează forma de undă SIN → TRI → SQR
 - **HOLD** (INT0): îngheață / reia captura
 - **VOLT+** / **VOLT-** (PCINT): crește / scade amplitudinea (5 niveluri: 20%–100%)
 - **TIME+** / **TIME-** (PCINT): crește / scade numărul de perioade vizibile (1–4)
- Potențiometrul (ADC canal 1): controlează cursorul de măsurare pe axa X
- Persistență setări în EEPROM AT24C256 (I²C): mod, amplitudine, timebase — restaurate la repornire

Structura modulelor software:

Modul	Fișiere	Rol
UART	uart.c / uart.h	Debug serial 57600 baud
ADC	adc.c / adc.h	Citire semnal + potențiometrul
SPI	spi.c / spi.h	Bus comun display + DAC
DAC	dac.c / dac.h	Generator semnal MCP4921
Display	display.c / display.h	Driver ILI9341, font 5x7, statistici
Timer1	timer1.c / timer1.h	Cadență captura (CTC, prescaler 64)
Buttons	buttons.c / buttons.h	INT0/INT1/PCINT, flaguri volatile
I ² C	i2c.c / i2c.h	Comunicație EEPROM AT24C256
EEPROM	eeprom.c / eeprom.h	Persistență setări

Proiectul nu folosește biblioteci externe — tot codul este scris de la zero folosind doar headere AVR-libc standard (avr/io.h, avr/interrupt.h, util/delay.h, avr/pgmspace.h)

Justificarea funcționalităților din laborator

Concept laborator	Unde e folosit în proiect	Justificare
GPIO	Pini CS/DC display, butoane cu pull-up intern	Controlul chip-select pe busul SPI partajat
UART	Debug serial la 57600 baud	Monitorizare valori ADC în timp real
Întreruperi externe	INT0 (HOLD), INT1 (MODE), PCINT (VOLT/TIME)	Butoanele răspund instant indiferent de durata draw_frame
Timer1 CTC	Funcțiile timer1_init/reset/wait în capture_frame	Cadență precisă: OCR1A × 4μs = perioadă fixă per eșantion
ADC	Canal 0 (ieșire op-amp), Canal 1 (potențiometrul)	Conversie analog-digital pentru ambele intrări
SPI	Display ILI9341 + DAC MCP4921 pe același bus	Transfer rapid 8 MHz, chip-select separat per periferic
I ² C	EEPROM AT24C256 la adresa 0x50	Stocare persistentă; potrivit pentru memorii externe

Scheletul proiectului și interacțiunea dintre funcționalități

Bucloa principală (main.c):

```
while (1) {
    // 1. Citire potențiomtru → cursor_x (mapare liniară cu calibrare hardware)
    // 2. Procesare flaguri ISR (mode/hold/volt/time) → update stare + EEPROM
    // 3. dacă !hold:
    //     capture_frame() → warm-up 128 iter. (OCR=35) + 240 eșantioane (OCR=200)
    //     draw_frame() → grilă + cursor + semnal coloană cu coloană
    //     draw_cursor_label() → tensiune la cursor, lângă linie
    //     draw_status_bar() → mod, T/div, V/div, HOLD
    //     draw_stats_panel() → Vpp, Vavg, T/div, CUR
    // 4. dacă hold:
    //     detectare schimbare cursor_x → redesenare frame din buffer (fără captură)
}
```

Fluxul de date:

```
DAC (MCP4921) -> semnal analogic -> condensator cuplaj AC 10μF
-> op-amp MCP6002 (buffer unitar, protecție diode) -> ADC canal 0
-> sample_buffer[240] -> calcul Vpp/Vavg/CUR -> display ILI9341
```

Sincronizarea DAC-ADC în capture_frame():

1. Timer1 CTC resetat la începutul fiecărei perioade
2. DAC scrie eșantionul → `_delay_us(5)` settling → ADC citit → așteptare OCR1A
3. Perioadă fixă: $200 \times 4\mu\text{s} = 800\mu\text{s}$ /eșantion → rata de captură ~1.25 kHz

Validare funcționalități:

- **ADC + DAC:** semnal generat intern trimis și citit înapoi prin ADC; varianță < 20mV confirmată prin UART (`uart_send_uint16` în faza de debug)
- **Butoane:** testate prin apăsare individuală — flagurile `pending_*` setate din ISR și procesate în bucla principală; debounce prin condiția `if (!pending_x) pending_x = 1`
- **EEPROM:** verificat prin cicluri de scoatere/rebranșare USB — setările (mod, amplitudine, timebase) restaurate corect; magic byte 0xAB la adresa 0x0000 previne încărcarea de date corupte
- **Timer1 CTC:** confirmat vizual — forma de undă stabilă la rata corectă de eșantionare
- **Cursor:** validat în modul HOLD — potențiometrul mută cursorul lin, CUR în statistici se actualizează în timp real

Calibrarea elementelor de senzoristica

Potențiometrul (ADC canal 1):

Potențiometrul fizic nu acoperă întregul rang teoretic 0-1023 al ADC-ului. Valorile reale au fost măsurate prin UART în faza de debug:

- Capăt minim fizic: ADC \approx 173
- Capăt maxim fizic: ADC \approx 954

Aceste valori sunt definite ca constante POT_MIN = 173 și POT_MAX = 954. Formula de calibrare cu clamp:

```
pot_clamped = (pot_raw <= POT_MIN) ? 0 :
               (pot_raw >= POT_MAX) ? (POT_MAX - POT_MIN) :
               (pot_raw - POT_MIN);
cursor_x = 1 + (uint32_t)pot_clamped * (SAMPLE_COUNT - 2) / (POT_MAX -
POT_MIN);
```

Rezultat: la capătul minim fizic → cursor la x=1, la capătul maxim → cursor la x=239. Fără calibrare, ultimul sfert al cursei rămânea fără efect vizibil.

Scalarea tensiunii (ADC → Volți):

ADC 10-bit cu Vref = 5V → 1 LSB = 5V/1023 \approx 4.89 mV. Vpp și Vavg sunt calculate direct din valorile ADC brute:

```
vpp_x10 = (uint32_t)(vmax - vmin) * 50 / 1023; // zecimi de volt
vavg_x10 = (uint32_t)vavg * 50 / 1023;
```

Consistența afișaj-calcul:

Funcția adc_to_y() mapează ADC 0→fundul grilei, ADC 1023→vârful grilei, fără margini de compresie — grila de 6 diviziuni \times 36px corespunde exact cu 5V / 6 \approx 0.83V/div, consistent cu valorile numerice afișate.

Optimizări

1. Font în PROGMEM (Flash, nu RAM)

Tabelul font 5 \times 7 (59 caractere \times 5 bytes = 295 bytes) este declarat cu atributul PROGMEM și citit cu pgm_read_byte(). ATmega328P are doar 2KB SRAM — fără această optimizare, fontul ar ocupa ~15% din SRAM-ul disponibil.

2. Warm-up cu 128 iterații (artefact triunghi)

Problema: condensatorul de cuplaj AC (10 μ F, RC \approx 50ms) nu se poate stabili rapid dacă DAC-ul face un salt brusc de la warm-up la captură. Soluția: 128 iterații de warm-up cu OCR=35 (rapid, 140 μ s/pas) garantează că phase = 0 la final pentru orice phase_step din intervalul 1-4, deoarece 128 \times k

$\text{mod } 64 = 0 \forall k$. Astfel nu există salt DAC la tranziția warm-up → captură.

3. Desenare coloană cu coloană (fără buffer de ecran)

ATmega328P nu are suficient RAM pentru un framebuffer complet ($240 \times 320 \times 2 = 150\text{KB}$). Soluția: `draw_frame()` procesează ecranul coloană cu coloană — șterge coloana, desenează grila, desenează semnalul. Se evită flickerul global și se elimină necesitatea unui buffer secundar.

4. ISR minim + flaguri volatile

ISR-urile de butoane nu fac nimic altceva decât să seteze un flag: `if (!pending_x) pending_x = 1`. Procesarea efectivă (UART, display, EEPROM) se face în bucla principală. Astfel ISR-ul durează < 10 cicli, minimizând latența pentru alte întreruperi.

5. SPI bus partajat DAC + Display

Ambele periferice SPI (ILI9341 și MCP4921) partajează MOSI/SCK; chip-select separat (CS_DAC pe PB2, CS_DISPLAY pe PD7) permite comunicație fără conflicte și elimină necesitatea unui al doilea controler SPI.

Rezultate Obținute

Filmare care demonstrează modul de funcționare al Osciloscopului:

<https://youtube.com/shorts/OEbLZLtn7qA>

Bibliografie/Resurse

Resurse Hardware

- ATmega328P Xplained Mini User Guide — Microchip Technology: <https://ww1.microchip.com/downloads/en/DeviceDoc/50002659A.pdf>
- ATmega328P Xplained Mini — OCW PM UPB: <https://ocw.cs.pub.ro/courses/pm/proiect/xplainedmini>
- ILI9341 TFT LCD Driver Datasheet — ILI Technology: <https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf>
- MCP4921 DAC Datasheet — Microchip Technology: <https://ww1.microchip.com/downloads/en/DeviceDoc/21897B.pdf>
- MCP6002 Op-Amp Datasheet — Microchip Technology: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP6001-1R-1U-2-4-1-MHz-Low-Power-Op-Amp-DS20001733L.pdf>
- AT24C256 EEPROM Datasheet — Microchip Technology: <https://ww1.microchip.com/downloads/en/DeviceDoc/doc0670.pdf>
- Pololu 2595 Level Shifter — Pololu: <https://www.pololu.com/product/2595>

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/vlad.radulescu2901/sebastian.barbu2707>



Last update: **2026/05/24 12:43**