

Ceas multifuncțional

Proiect realizat de Ioniță Alexandru-Andrei, 332CA

Introducere

Despre ce este proiectul?

Ceas digital inteligent care combină afișarea timpului, măsurarea temperaturii, redarea posturilor FM și a fișierelor MP3 de pe card SD, plus configurare și control wireless prin Bluetooth, toate gestionate de un Arduino Mega 2560.

Care este scopul proiectului?

Să dezvolte un dispozitiv multifuncțional pentru noptieră, integrând protocoalele și modulele studiate la laborator (USART, timere/întreruperi, PWM, ADC, SPI, I²C) într-un sistem util și interactiv.

Ce a inspirat ideea?

Inspirat de ecosistemele smart home și de gadgeturile de tip „all-in-one” pentru casă, proiectul își propune să ofere informații de mediu și divertisment audio într-un singur dispozitiv compact.

De ce este util acest proiect?

Acest ceas multifuncțional:

afișează ora și temperatura în timp real;

redă radio FM și muzică de pe microSD;

permite setarea alarmelor și redarea audio prin aplicație mobilă.

Descriere generală



Descrierea componentelor

| Componentă | Funcționalitate principală |
|---------------------------|--|
| Arduino Mega 2560 (CH340) | MCU, gestionare protocoale I ² C/SPI/UART/PWM/ADC și ISR. |
| RTC DS3231 | Ceas în timp real I ² C, acuratețe ridicată. |

| | |
|-------------------------------------|---|
| LCD 16x2 I ² C (PCF8574) | Afișare temperatură, titlu melodie și mesaje. |
| Radio FM TEA5767 | Recepție FM I ² C, control frecvență și volum. |
| TM1637 4x7-seg | Afișare ora format HH:MM multiplexat. |
| Senzor temperat. LM35 | Citire analogică via ADC (10 mV/°C). |
| Modul MP3 YX5200 | Decodare MP3 de pe microSD, control serial UART. |
| Amplificator TPA3118 (12 V) | Amplificator Clasa D, până la 60 W. |
| Modul Bluetooth HC-05 | Comunicație UART master/slave pentru aplicație mobilă. |

Hardware Design

Lista componentelor și link-uri

| Nume componentă | Cantitate | Link achiziție | PIN |
|-------------------------------------|-----------|---|--|
| Arduino Mega 2560 | 1 | https://www.optimusdigital.ro/.../mega-2560-ch340.html | - |
| RTC DS3231 | 1 | https://www.optimusdigital.ro/.../modul-ceas-ds3231.html | SDA – pin 20, SCL – pin 21, VCC – 5V, GND – GND |
| Senzor de temperatură LM35 | 1 | https://www.optimusdigital.ro/.../senzor-lm35.html | OUT – A7, VCC – 5V, GND – GND |
| LCD 16x2 I ² C (PCF8574) | 1 | https://www.optimusdigital.ro/.../lcd-i2c.html | SDA – pin 20, SCL – pin 21, VCC – 5V, GND – GND |
| Modul display TM1637 4x7-seg | 1 | https://www.optimusdigital.ro/.../tm1637.html | DIO – D4, CLK – D5, VCC – 5V, GND – GND |
| Modul MP3 YX5200 | 1 | https://www.optimusdigital.ro/.../mp3-mini.html | VCC – 5V, GND – GND, SPK+ – la comutator audio, SPK- – GND |
| Amplificator TPA3118 60 W | 1 | https://www.optimusdigital.ro/.../tpa3118.html | IN+ – la ieșire comutator audio, IN- – GND, VCC – 5V/12V, GND – GND, OUT+/- – la difuzor |
| Modul Bluetooth HC-05 | 1 | https://www.optimusdigital.ro/.../hc05.html | VCC – 5V, GND – GND, TX (modul) – D8 (RX la Arduino), RX (modul) – D9 |
| Breadboard 750 puncte | 1 | (generic) | - |
| Jumper wires (Dupont) | set | (generic) | - |

| | | | |
|----------------------------------|-----|-------------------------|--|
| Condensator 47 nF/50 V | 2 | (ceramic, cuplaj audio) | Între ieșirea audio și IN+ amplificator (opțional, pentru filtrare zgomot/impulsuri) |
| Cablu audio AUX 3.5 mm M-M | 1 | (generic) | Între amplificator și difuzor sau între surse audio externe (după caz) |
| Cablu boxă twin-core AWG20 | 1 m | (speaker wire) | Între OUT+/- amplificator și difuzor |
| Adaptor 12 V 2 A | 1 | (generic) | Alimentare pentru amplificator (și/sau Arduino, dacă e nevoie) |
| Difuzor 4-8 Ω, 3-5 W | 1 | (generic) | La OUT+ și OUT- de la amplificator |



Cod

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#define LED_PIN PB7
#define BUTTON_PIN PD2
#define BUZZER_PIN PH4
void uart_init() {
  UBRR0H = 0;
  UBRR0L = 103;
  UCSR0B = (1 << RXEN0) | (1 << TXEN0);
  UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);
}
void uart_tx(char c) {
  while (!(UCSR0A & (1 << UDRE0)));
  UDR0 = c;
}
void uart_print(const char *s) {
  while (*s) uart_tx(*s++);
}
volatile uint8_t btn_flag = 0;
ISR(INT0_vect) { btn_flag = 1; }
void pwm_init() {
  DDRB |= (1 << PB4);
```

```
TCCR2A = (1 << COM2A1) | (1 << WGM20);
TCCR2B = (1 << CS21);
OCR2A = 0;
}
void adc_init() {
ADMUX = (1 << REFS0);
ADCSRA = (1 << ADEN) | (1 << ADPS2);
}
uint16_t adc_read(uint8_t ch) {
ADMUX = (ADMUX & 0xF0) | (ch & 0x0F);
ADCSRA |= (1 << ADSC);
while (ADCSRA & (1 << ADSC));
return ADC;
}
void spi_init() {
DDRB |= (1 << PB2) | (1 << PB1) | (1 << PB0);
SPCR = (1 << SPE) | (1 << MSTR);
PORTB |= (1 << PB0);
}
uint8_t spi_transfer(uint8_t data) {
SPDR = data;
while (!(SPSR & (1 << SPIF)));
return SPDR;
}
void twi_init() {
TWSR = 0;
TWBR = 72;
TWCR = (1 << TWEN);
}
void twi_start() {
TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN);
while (!(TWCR & (1 << TWINT)));
}
void twi_stop() {
TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);
}
uint8_t twi_write(uint8_t data) {
TWDR = data;
TWCR = (1 << TWINT) | (1 << TWEN);
while (!(TWCR & (1 << TWINT)));
return (TWSR & 0xF8);
}
void setup() {
DDRB |= (1 << LED_PIN);
DDRH |= (1 << BUZZER_PIN);
DDRD &= ~(1 << BUTTON_PIN);
PORTD |= (1 << BUTTON_PIN);
uart_init();
EICRA |= (1 << ISC01);
EIMSK |= (1 << INT0);
sei();
}
```

```
pwm_init();
adc_init();
spi_init();
twi_init();
uart_print("Ceas multi-lab\r\n");
}
void delay_ms(uint16_t ms) {
while(ms-- > 0) _delay_ms(1);
}
void loop() {
PORTB |= (1 << LED_PIN);
delay_ms(300);
PORTB &= ~(1 << LED_PIN);
delay_ms(300);
uart_print("Salut!\r\n");
if (btn_flag) {
uart_print("Buton INTO!\r\n");
PORTH |= (1 << BUZZER_PIN);
delay_ms(100);
PORTH &= ~(1 << BUZZER_PIN);
btn_flag = 0;
}
static uint8_t duty = 0;
OCR2A = duty;
duty += 32;
uint16_t val = adc_read(0);
char buf[32];
float temp = val * (5.0 / 1023.0) * 20;
snprintf(buf, sizeof(buf), "Temp: %.1f C\r\n", temp);
uart_print(buf);
PORTB &= ~(1 << PB0);
uint8_t resp = spi_transfer(0xAA);
PORTB |= (1 << PB0);
twi_start();
uint8_t status = twi_write(0xD0);
if (status == 0x18) uart_print("RTC gasit!\r\n");
else uart_print("RTC lipsa!\r\n");
twi_stop();
delay_ms(1000);
}
int main(void) {
setup();
while (1) {
loop();
}
}
```

Lab 0 (GPIO): pinii pentru buzzer, display 8 segmente

Lab 1 (UART): Serial Monitor + SoftwareSerial pentru Bluetooth HC-05, primești/setezi alarmă/radio de la PC sau app BT.

Lab 3 (Timere, PWM): PWM pe buzzer, toggling cu millis().

Lab 5 (SPI): Citire fisiere SD Card

Lab 6 (I2C): RTC DS3231, LCD, Radio RDA5807M – toate pe I2C.

Rezultate Obținute

În urma dezvoltării proiectului „Ceas multifuncțional”, am obținut următoarele rezultate:

Funcționalitate completă hardware: Toate modulele hardware au fost conectate și testate — ceas RTC, afișaj LCD I2C, display TM1637, buzzer de alarmă, modul radio FM și MP3 player cu amplificator.

Afișaj ora și dată: Ceasul afișează în timp real ora și data atât pe LCD I2C, cât și pe display-ul TM1637 cu 4 cifre.

Alarmă sonoră: Sistemul permite programarea unei alarme, care activează buzzer-ul la ora setată și afișează un mesaj pe LCD.

Redare audio: Am reușit să comut între două surse audio (radio FM sau MP3 player), semnalul fiind amplificat și, opțional, redat pe difuzor.

Interfață Bluetooth: Utilizatorul poate seta ora alarmei, poate schimba frecvența radio sau melodia de pe MP3 player direct dintr-o aplicație pe telefon, folosind modulul Bluetooth HC-05.

Cod modular, utilizare registre: Pentru testare și validare, am implementat funcționalitățile atât cu biblioteci, cât și direct pe registre AVR, pentru a ilustra cunoștințele teoretice și practice dobândite la fiecare laborator.

Testare și prezentare: Proiectul a fost testat pe breadboard și poate fi demonstrat live, atât cu, cât și fără unele componente (ex: difuzor), arătând fluxul semnalului și interacțiunea dintre module.

Concluzii

Proiectul a validat atât integrarea hardware a componentelor electronice diverse, cât și implementarea software pe platforma Arduino/AVR, acoperind temele principale din laboratoarele de microcontrolere: GPIO, UART, întreruperi, ADC, PWM, SPI și I2C, precum și comunicarea wireless și interfața cu utilizatorul.

Bibliografie/Resurse

Software

Arduino Documentation & Tutorials <https://www.arduino.cc/en/Guide/HomePage>

PlatformIO Documentation <https://docs.platformio.org/>

Librăria "avr-libc" și manualul de programare AVR
<https://www.nongnu.org/avr-libc/user-manual/index.html>

Hardware:

ATmega2560 Datasheet
<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega2560-Data-Sheet-DS40002211A.pdf>

Optimus Digital - Fise tehnice module

Modul RTC DS3231 -
<https://www.optimusdigital.ro/ro/altele/1102-modul-cu-ceas-in-timp-real-ds3231.html>

Modul MP3 YX5200 -
<https://www.optimusdigital.ro/ro/audio-altele/2146-modul-de-redare-mp3-in-miniatura.html>

Modul Radio RDA5807M -
<https://www.optimusdigital.ro/ro/wireless-radio-fm/745-modul-radio-fm-rda5807m.html>

Modul Bluetooth HC-05 -
<https://www.optimusdigital.ro/ro/wireless-bluetooth/153-modul-bluetooth-master-slave-hc-05-cu-adaptor.html>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2025/eradu/alexandru.ionita03>



Last update: **2025/05/25 18:15**

