Advanced Real-Time Clock System

Project Overview

A comprehensive real-time embedded system using ATmega32 microcontroller (8 MHz) that functions as:
- Real-time clock (RTC) and calendar
- Alarm system with buzzer notification
- Stopwatch with start/stop/reset functionality
- Countdown timer with configurable time and buzzer alert

All features are user-controlled via push buttons and visually displayed through a 16x2 character LCD.
Hardware modules include an RTC module (DS1307/DS3231) and a buzzer for alerts.

Hardware Components

- Microcontroller: ATmega32 (8 MHz clock)
- RTC Module: DS1307 or DS3231 (I2C communication)
- Character LCD: 16x2 HD44780-based (parallel interface)
- Push Buttons: 6 buttons for various functions
- Buzzer: For alarm and countdown notifications
- Power Source: Stable 5V (with optional backup battery for RTC)

System Architecture

The system is built using modular programming principles with modules:
1. Main Program Loop
2. RTC Interface
3. LCD Display
4. Button Interface
5. Stopwatch
6. Countdown Timer
7. Alarm System
8. Buzzer Control
9. Time Utilities

System Modes

Mode 0: Clock Mode – Display current time/date
Mode 1: Time Setting Mode – Edit time/date
Mode 2: Alarm Setting Mode – Configure alarm
Mode 3: Stopwatch Mode – Measure elapsed time
Mode 4: Countdown Mode – Timer with buzzer alert

Button Guide

MODE – Cycle modes
SET – Change fields
INC – Increment value

DEC – Decrement value
START/STOP – Start/Stop timing
RESET – Reset stopwatch/timer

Pin Connections

LCD: RS-PD0, RW-PD1, EN-PD2, D4-PD4, D5-PD5, D6-PD6, D7-PD7
RTC: SDA-PC1, SCL-PC0
Buttons: MODE-PC0, INC-PC1, DEC-PC2, SET-PC3, START/STOP-PC4, RESET-PC5
Buzzer: PB0

Technical Specifications

Clock Speed: 8 MHz
Timer1: CTC mode, 1 Hz interrupt
RTC Protocol: I2C
Display: 16x2 LCD, 4-bit mode

Build & Flash

Requirements: AVR-GCC, AVRDUDE, USBasp
Commands:
make
make flash
make clean
make size

Memory Usage

Flash: ~8-12 KB
RAM: ~200-300 bytes
EEPROM: Not used

Features

Implemented:
- Real-time clock & date
- Alarm
- Stopwatch
- Countdown timer
- Button debouncing
- LCD management
- I2C RTC communication
- Modular architecture
- Interrupt-based timing

Future:
- EEPROM storage
- Temperature display
- Multiple alarms
- Brightness adjustment

- Melody buzzer
- 12/24h toggle
- Day of week display

Troubleshooting

LCD issues – check connections, contrast
RTC not responding – check I2C wiring
Buttons not working – check pull-ups
Buzzer not sounding – test direct pin

License

MIT License – Open source

Contributing

Pull requests and issues are welcome