RTC-Based Digital Alarm Clock Project Technical Documentation

Overview

Welcome to the documentation for the RTC-Based Digital Alarm Clock project. This comprehensive guide provides a detailed insight into the creation of a versatile alarm clock using the STM32F407VET6 microcontroller. This project combines hardware components such as a 4-digit 7-segment display, push buttons, a buzzer, and a DC-DC buck converter with a 9-volt battery. The clock, coded in C language using STM32CubeMX and VSCode with the PlatformIO extension, displays real-time information, sets alarms, and maintains accurate time even after power loss. This open-source project is distributed under the GNU General Public License v2 (GPL-2.0).

Table of Contents

- •Millis() Function
- •RTC Driver and Configuration
- •7-Segment Display Driver
- •Software Flowchart

4. **Functionality**

- •Time and Date Display
- •RTC Initialization
- •Time and Date Setting
- •Alarm Configuration and Triggering
- •Buzzer Activation

5. **Power Management**

- •RTC Chip for Timekeeping
- •Power Backup for RTC
- •Maintaining Accurate Time

6. **Usage Instructions**

- •Setting Time and Date
- •Configuring Alarms
- •Operating the Alarm Clock
- •Power-On and Power-Off Behavior

7. **Licensing**

- •GNU General Public License v2 (GPL-2.0)

8. **Future Enhancements**

- •User Interface Improvements
- •Multiple Alarms
- •Configurable Alarm Tones

9. **Acknowledgments**

- •References and Contributions

10. **Contact Information**

- • Contact Details

---

1. Introduction

Project Goals

The RTC-Based Digital Alarm Clock project aims to create an advanced alarm clock using the STM32F407VET6 microcontroller. This clock not only displays accurate time and date information but also incorporates alarm functionality and robust power management. The project's unique feature is the capability to maintain precise time even during power interruptions, thanks to the integrated RTC chip.

Key Components

- •STM32F407VET6 Microcontroller
- •4-Digit 7-Segment Display
- •Push Buttons for User Interaction
- •Buzzer for Audible Alerts
- •Pull-Up Resistors for Signal Stabilization
- •DC-DC Buck Converter for Voltage Regulation
- •9-Volt Battery for Main Power Supply

2. Hardware Setup

STM32F407VET6 Microcontroller

The STM32F407VET6 serves as the project's core, handling user inputs, display control, RTC communication, and more.

4-Digit 7-Segment Display

The 7-segment display visually conveys time, date, and alarm information using the microcontroller's GPIO pins.

Push Buttons

Four push buttons facilitate user interaction, allowing time and alarm configuration as well as navigation.

Buzzer

The buzzer provides audible alerts when alarms are triggered, ensuring user attention.

Pull-Up Resistors

Pull-up resistors stabilize signal levels and prevent undefined inputs from the push buttons.

DC-DC Buck Converter and 9-Volt Battery

A DC-DC buck converter regulates the 9-volt battery's voltage to a stable 5 volts, powering the microcontroller and components.

Schematic Diagram

Refer to the comprehensive schematic diagram that illustrates connections between the microcontroller, display, buttons, buzzer, converter, and battery, accurately depicting voltage conversion through the buck converter.

3. Software Architecture

STM32CubeMX Configuration

Project configuration is performed using STM32CubeMX to initialize peripherals such as GPIO, timers, and more.

Software Development Tools

Coding is executed using VSCode alongside the PlatformIO extension, providing a seamless environment for coding, debugging, and uploading to the microcontroller.

Millis() Function

The millis() function aids accurate time measurement intervals without blocking the microcontroller's execution.

RTC Driver and Configuration

The RTC module is configured to ensure precise time and date display, handling time synchronization, date configuration, and alarm settings.

7-Segment Display Driver

The display driver facilitates multiplexing of the 7-segment display, enabling dynamic updates for time, date, and alarm information.

Software Flowchart

Refer to the visual flowchart that provides an overview of the software's logic and operation sequence, enhancing code comprehension.

4. Functionality

Time and Date Display

The clock displays the current time and date in the [HH:MM] format, dynamically updating to reflect changes.

RTC Initialization

At startup, the RTC chip is initialized for accurate timekeeping, ensuring synchronization with the RTC for precise display.

Time and Date Setting

Users set time and date using the push buttons. The clock guides users through the process, updating the RTC accordingly.

Alarm Configuration and Triggering

Users configure alarms via the interface. When alarm times are reached, the buzzer sounds, and the display indicates triggered alarms.

Buzzer Activation

The buzzer sounds when alarms trigger, providing an audible alert in addition to the visual display.

5. Power Management

RTC Chip for Timekeeping

The RTC chip maintains accurate time during power interruptions, ensuring the clock displays the correct time upon power restoration.

Power Backup for RTC

The onboard lithium cell acts as a power backup, allowing the RTC chip to function during power outages and

preserving accurate timekeeping.

Maintaining Accurate Time

The combination of the RTC chip and power backup mechanism ensures the clock maintains precise time over extended periods.

6. Usage Instructions

Refer to the RTCAlarmClock_UserManual for this section.

7. Licensing

The RTC-Based Digital Alarm Clock project is distributed under the GNU General Public License v2 (GPL-2.0), encouraging open collaboration and usage.

8. Future Enhancements

User Interface Improvements

Enhance the user interface with more intuitive menu navigation and feedback mechanisms.

Multiple Alarms

Support configuring and managing multiple alarms to cater to various scheduling needs.

Configurable Alarm Tones

Allow users to select different alarm tones or melodies according to their preferences.

9. Acknowledgments

We extend our gratitude to the STM32 community, references, libraries, and contributors that have played a vital role in shaping this project.

10. Contact Information

For questions, suggestions, or collaboration opportunities, please feel free to contact :

subhadeepdhang1@gmail.com

linkedin.com/in/subhadeep-dhang

github.com/imrealsd