Control de un Reloj en Tiempo Real (RTC) programable

Alejandro González Quesada, Wayner Jiménez Vega, Yuliana Fernández Romero, Fernando Chacón Zúñiga

*Ingeniería en sistemas de computación, Universidad Fidélitas, San Pedro, Costa Rica*

[ngonzalez40383@ufide.ac.cr](mailto:ngonzalez40383@ufide.ac.cr)

wjimenez10058@ufide.ac.cr [yfernandez00458@ufide.ac.cr](mailto:yfernandez00458@ufide.ac.cr) fchacon20461@ufide.ac.cr

***Abstract*— El desarrollo de un sistema de control para un Reloj en Tiempo Real con Calendario (RTCC) programable responde a una necesidad real planteada por la empresa Cañada de Costa Rica S.A, que requiere una solución funcional, flexible y personalizable para la gestión de alarmas y eventos. Este proyecto no solo nos permite fortalecer el conocimiento técnico en el área de programación embebida, sino también aplicar de forma práctica los principios del Paradigma de Programación Orientada a Objetos (POO), integrando múltiples componentes de hardware y lógica de software.**

**La solución propuesta contempla el uso de una pantalla LCD1604 y botones pulsadores para lograr una interfaz de usuario interactiva que facilite la visualización y edición de la hora, el calendario y las alarmas. Asimismo, se incorpora una lógica robusta que permite alternar entre la visualización del reloj y las alarmas, así como configurar eventos futuros con notificaciones visuales y sonoras, adaptándose a las necesidades del usuario.**

1. INTRODUCCIÓN

Este proyecto nos pone en práctica conceptos fundamentales de diseño de sistemas embebidos, manejo de tiempo real, control de eventos, edición interactiva y uso eficiente de los recursos de un microcontrolador como Arduino Uno R3 o Arduino Mega. Adicionalmente, el desarrollo del calendario gregoriano completamente por software sin utilizar circuitos integrados adicionales representa un desafío técnico que fomenta el ingenio y el dominio de la lógica algorítmica de nosotros como estudiantes.

Con este sistema se garantiza no solo la funcionalidad básica de un reloj despertador digital, sino también una experiencia de usuario enriquecida y altamente personalizable. En este sentido, el proyecto contribuye al desarrollo de habilidades clave para el futuro profesional de nosotros, además de proveer a la empresa una herramienta útil, adaptable y diseñada a la medida de sus requerimientos.

1. OBJETIVO GENERAL

Desarrollar un sistema de control para un Reloj en Tiempo Real con Calendario (RTCC) programable utilizando el paradigma de programación orientada a objetos, que permita al usuario visualizar, configurar y gestionar la hora, fecha, alarmas y eventos a través de una interfaz implementada en una pantalla LCD1604 y controlada mediante unos botones pulsadores.

1. OBJETIVOS ESPECIFICOS

Diseñar e implementar una interfaz de usuario interactiva que permita visualizar en formato de 12 o 24 horas la hora actual, el calendario gregoriano, la hora de alarma y los mensajes correspondientes en la pantalla LCD1604.

Desarrollar una lógica de control por software que permita la edición del reloj, la fecha, la alarma y los eventos configurables, utilizando botones pulsadores para navegar, modificar y confirmar los distintos campos, sin utilizar circuitos integrados adicionales.

Integrar funciones de alarma personalizables que incluyan selección de tonos audibles, indicadores visuales mediante LED, y generación de alertas intermitentes programadas, garantizando una correcta activación y desactivación automática según el horario definido por el usuario.

1. EXPLICACION DE LA SOLUCIÓN

Este proyecto implementa un reloj digital con funciones avanzadas como:

Visualización de la hora actual (RTC).

Configuración de alarma programable.

Selección de tono de alarma personalizado.

Indicadores LED según el tono.

Interfaz mediante botones físicos y una pantalla LCD de 4 filas x 20 columnas.

Componentes Usados:

Pantalla LCD 20x4: Para mostrar hora, fecha, alarma, etc.

Botones físicos: Para navegar y editar valores.

LEDs (5): Indican el tono de alarma seleccionado.

Buzzer: Para emitir el sonido de alarma.

Arduino Mega (o Uno con suficientes pines analógicos/digitales).

Distribución de Pines

LCD conectada a pines: 23, 22, 24, 25, 26, 27.

Botones:

A15: ↑

A11: ↓

A14: ←

A12: →

A13: Centro (cambiar de vista)

A10: Seleccionar tono

A9: Detener alarma

LEDs: A0 a A3 + A7.

Buzzer: A8.

Selección de Tonos:

Hay 5 tonos disponibles (Do, Re, Mi, Fa, Sol).

Se pueden cambiar presionando el botón BTN\_SELECT\_TONE.

El tono seleccionado se reproduce durante 1 segundo.

Si no se selecciona otro en 3 segundos, el tono actual se confirma.

Alarma:

Se puede configurar una hora específica para activar la alarma.

Cuando se activa:

Suena el buzzer con el tono seleccionado.

Se enciende un LED específico relacionado al tono.

Se muestra un mensaje en la LCD: "ALARMA ACTIVADA".

Puede ser detenida con el botón BTN\_STOP\_ALARM.

Edición:

Se puede alternar entre editar la hora actual o la alarma con el botón central.

Los botones ↑ y ↓ incrementan o decrementan campos.

El botón → cambia de campo (hora, minuto, día, mes, etc.).

Se permite configurar:

Hora, minuto, segundo

Día, mes, año

Día de la semana (editable manualmente)

Interfaz LCD:

La pantalla LCD muestra:

Primera fila: Modo RTC o Alarma y si está en "Edición".

Segunda fila: La hora actual o de la alarma.

Tercera fila: La fecha (dd/mm/yyyy).

Cuarta fila: Día de la semana.

1. COMPONENTES UTILIZADOS
2. *Arduino Mega 2560*

Tiene una gran cantidad de pines digitales y analógicos, ideal para manejar múltiples entradas y salidas como en este proyecto.

Es el microcontrolador principal que controla todo el sistema. Aquí se ejecuta el programa que gestiona el RTC, la visualización en pantalla, la configuración de alarmas, el control de botones, el buzzer y los LEDs.

1. *Pantalla LCD1604 (LCD2)*

Muestra la información principal del sistema: la hora, la fecha, el día, el modo activo (RTCC o Alarma), y los mensajes de ayuda/interacción. Tiene 4 filas por 16 columnas de caracteres. Está conectada al Arduino mediante los pines digitales.

1. *Botones Pulsadores*

Identificados como:

BTN\_IZQUIERDO: Entra/sale del modo de edición.

BTN\_CENTRAL: Cambia entre visualización de RTC y alarma.

BTN\_DERECHO: Cambia entre campos a editar (hora, minutos, segundos, día).

BTN\_ARRIBA: Incrementa el valor del campo seleccionado.

BTN\_ABAJO: Decrementa el valor del campo seleccionado.

BTN\_TONO: Permite escuchar/seleccionar uno de los tonos para la alarma.

BTN\_STOP\_ALARM: Detiene manualmente la alarma si está activa.

Cada botón tiene una resistencia de pull-down de 10kΩ para asegurar lecturas estables.

1. *Resistencias (R6-R12, R13)*

10kΩ: Usadas como pull-down en cada botón, garantizan que el estado del pin sea bajo (LOW) cuando el botón no está presionado.

220Ω: Se encuentran en serie con los LEDs y el buzzer para limitar la corriente y evitar dañar los componentes.

1. *LEDs (D1-D5)*

Colores: BRIGHT, PURPLE, RED, WHITE, YELLOW.

Emulan señales visuales de la alarma.

Se pueden usar también para indicar cuál tono está seleccionado.

Parpadean a 3 Hz por 120 segundos cuando se activa un evento programado.

También pueden activarse con la alarma durante 60 segundos.

1. *Speaker / Buzzer LS1*

Emite el tono sonoro de la alarma. Puede reproducir 5 tipos diferentes de tono que el usuario puede escuchar y seleccionar.

Controlado directamente por un pin del Arduino a través de una resistencia limitadora de 220Ω.

1. *Conexiones generales*

Todos los componentes están conectados al Arduino Mega de manera lógica:

Las salidas digitales controlan los LEDs y el buzzer.

Las entradas digitales leen los estados de los botones.

La LCD utiliza pines digitales para mostrar la información en tiempo real.

Funcionamiento Principal (loop())

Cada vez que se ejecuta el loop:

Lee el estado de los botones.

Permite editar valores si está en modo edición.

Actualiza la hora cada segundo.

Verifica si la hora coincide con la de la alarma → activa alarma.

Parpadea los campos en edición cada 500 ms.

Actualiza la pantalla LCD cuando sea necesario.

Ventajas de la Solución

Interfaz intuitiva con botones físicos.

Totalmente configurable: tanto el reloj como la alarma.

Selección de tono personalizada.

Indicadores visuales (LED) y auditivos (buzzer).

Código modular y bien estructurado.

1. CONCLUSION

Este proyecto nos permitió poner en práctica lo aprendido durante el curso, aplicando conceptos de programación para crear un sistema funcional y útil como un reloj despertador digital. A través del uso del Arduino Mega y componentes básicos como una pantalla LCD, botones y LEDs, logramos desarrollar una solución que no solo muestra la hora y la fecha en tiempo real, sino que también permite al usuario configurar alarmas personalizadas, cambiar tonos y recibir alertas visuales y sonoras.

Fue una experiencia enriquecedora, ya que nos enfrentamos al reto de diseñar todo desde cero, incluyendo la lógica para el calendario y el manejo de los botones. Nos ayudó a entender mejor cómo se pueden construir sistemas interactivos combinando software y hardware, y cómo hacer que un proyecto sea amigable para el usuario.

En resumen, este trabajo nos permitió reforzar los conocimientos, mejorar nuestras habilidades de programación y dar un paso más en el camino para convertirnos en ingenieros.

1. REFERENCIAS
2. [1] L. del Valle Hernández, "Reloj con Arduino, cómo controlar los tiempos con un RTC," *Programar Fácil*, [Online]. Available: <https://programarfacil.com/blog/arduino-blog/reloj-con-arduino-rtc/>.