

Proyecto No. 1 - Explicación del Código

José de Jesus Valenzuela Velásquez

17/03/2025

1. Configuración Inicial

La rutina **SETUP** se encarga de la configuración inicial del microcontrolador. En este bloque, se establece el puntero de pila, se desactiva la comunicación UART, y se configuran los puertos de entrada/salida (I/O) para los botones y LEDs. También se habilitan las interrupciones para los cambios en los pines de los botones y se inicializan las variables en la RAM. Esto asegura que el sistema esté listo para funcionar antes de entrar en el bucle principal.

2. Bucle Principal

El bucle principal se ejecuta en la rutina **MAIN**. En cada iteración, se apagan los displays mediante la subrutina **APAGAR_DISPLAYS**, se selecciona qué dígito mostrar utilizando la subrutina **SELECCIONAR_DISPLAY**, se muestra el dígito correspondiente a través de la subrutina **MOSTRAR_DIGITO**, y se introduce un pequeño retraso para la multiplexación mediante la subrutina **DELAY_MUX**. Este bucle se repite indefinidamente, permitiendo que el reloj funcione continuamente.

3. Manejo de Displays

Las rutinas **SELECCIONAR_DISPLAY** y **MOSTRAR_DIGITO** son responsables de determinar qué información se debe mostrar en el display en función del modo actual (hora, fecha, configuración). **SELECCIONAR_DISPLAY** verifica el modo y selecciona el dígito correspondiente, mientras que **MOSTRAR_DIGITO** obtiene el patrón para el dígito actual y lo muestra en el display de 7 segmentos. Esto se logra mediante un sistema de multiplexión que controla múltiples displays, asegurando que solo uno esté activo en un momento dado.

4. Interrupciones

El código incluye dos rutinas de interrupción: **PCINT0_ISR** para el cambio de pines y **TMR0_ISR** para el desbordamiento del temporizador. La rutina **PCINT0_ISR** maneja la entrada de los botones, permitiendo al usuario cambiar

entre modos y ajustar la hora y la fecha. La rutina **TMR0_ISR** se encarga de incrementar el tiempo, controlar el parpadeo de los LEDs y activar la alarma si corresponde. Esto permite que el reloj funcione de manera precisa y responda a las entradas del usuario de forma eficiente.

5. Manejo del Tiempo

Las subrutinas de manejo de tiempo, como **INCREMENTAR_SEGUNDOS**, **INCREMENTAR_MINUTOS**, **INCREMENTAR_HORAS**, **INCREMENTAR_MIN_DECENAS** y **INCREMENTAR_DÍA**, son responsables de la lógica de incremento de tiempo. Estas subrutinas manejan el conteo de segundos, minutos y horas, asegurando que se realicen las transiciones adecuadas (por ejemplo, de 59 segundos a 0 y 1 minuto, o de 23:59 a 00:00). También se incluyen subrutinas para manejar el cambio de días y meses, teniendo en cuenta los límites de cada mes y el formato de 24 horas.

6. Configuración de Hora y Fecha

El código permite la configuración manual de la hora y la fecha a través de botones. Las subrutinas como **INCREMENTAR_HORAS_MANUAL**, **DECREMENTAR_HORAS_MANUAL**, **INCREMENTAR_DIAS_MANUAL** y **DECREMENTAR_DIAS_MANUAL** permiten al usuario ajustar la hora y la fecha según sea necesario. Se implementan verificaciones para asegurarse de que los valores se mantengan dentro de los límites válidos (por ejemplo, no permitir que las horas excedan 23 o que los días excedan el número de días en el mes).

7. Alarma

El sistema de alarma se gestiona mediante variables que almacenan la hora y los minutos de la alarma, así como su estado (activa o inactiva). Las subrutinas **INCREMENTAR_HORAS_ALARMA**, **DECREMENTAR_HORAS_ALARMA**, **INCREMENTAR_MINUTOS_ALARMA** y **DECREMENTAR_MINUTOS_ALARMA** permiten al usuario configurar la alarma. La alarma se activa cuando la hora actual coincide con la hora configurada, y se emite un sonido a través de un buzzer conectado al microcontrolador.