

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

IE2023 – Programación de microcontroladores.

Sección 20

Ing. Pedro Castillo



Proyecto 1

Reloj

Thomas Jabes David Solís Lopez- 21626

Guatemala, 12 de marzo de 2025

Proyecto 1: Reloj

En este proyecto, se diseñó e implementó un reloj digital con alarma utilizando un microcontrolador ATmega328P, el cual permite visualizar la hora en formato de 24 horas, configurar la fecha y establecer una alarma con un buzzer audible.

El sistema cuenta con cuatro displays de 7 segmentos multiplexados para mostrar la hora y la fecha. Además, se incorporaron botones de control que permiten ajustar la configuración del reloj y la alarma de manera intuitiva, con un sistema de overflow y underflow, lo que significa que los valores pueden avanzar o retroceder de manera cíclica (por ejemplo, 00 → 59 en minutos). Los dos puntos del display parpadean cada 500 ms, utilizando un Timer0 configurado con interrupciones.

Requisitos del Proyecto

- Visualización de la Hora y Minutos: Formato de 24 horas en 4 displays de 7 segmentos multiplexados.
- Alarma Configurable: Sistema de alarma audible que permite programar hora y minutos.
- Overflow y Underflow en Valores: Los valores de horas, minutos, días y meses deben cambiar de manera cíclica (Ejemplo: 03 → 02 → 01 → 00 → 59).
- Parpadeo de los Dos Puntos (:): Implementado con LEDs o en el display de 7 segmentos, usando un Timer0 para parpadear cada 500ms.
- Visualización de la Fecha (DD/MM): Debe mostrar la fecha considerando la cantidad de días de cada mes (Ejemplo: 28/02 para febrero).
- Indicador de Modo de Configuración: Se debe mostrar una señal al usuario (LED o parpadeo) cuando esté configurando hora, fecha o alarma.
- Anti-rebotes en Botones: Los botones deben funcionar con el mecanismo presionar y soltar, evitando rebotes eléctricos.
- Manejo de Interrupciones: Uso de Timer0 para controlar el tiempo y PCINT (Pin Change Interrupts) para los botones.

Esquemático del proyecto

En la siguiente figura 1 se presenta el esquemático eléctrico del reloj digital con alarma basado en el microcontrolador ATmega328P. Este circuito fue diseñado para gestionar

la visualización de la hora, la fecha y la alarma mediante displays de 7 segmentos multiplexados.

Descripción de los Elementos Utilizados

Para la implementación del circuito se utilizaron los siguientes componentes:

- Microcontrolador ATmega328P: Se encarga del procesamiento de datos, la multiplexación del display y el manejo de la alarma.
- 4 Displays de 7 segmentos (cátodo común): Permiten la visualización de la hora, fecha y alarma.
- Transistores NPN (2N3904 o 2N2222): Controlan la multiplexación de los displays activando y desactivando los dígitos de manera secuencial.
- Botones de entrada (Push-buttons):
 - Modo: Cambia entre la visualización de hora, fecha y alarma.
 - Incrementar/Decrementar: Ajustan los valores de hora, minutos, día, mes y alarma.
- Buzzer: Se activa cuando la alarma coincide con la hora del reloj.
- Resistencias de 330Ω a 1kΩ: Para limitar la corriente en los segmentos del display y proteger el circuito.
- LEDs indicadores: Señalan cuando el sistema está en modo configuración (hora, fecha o alarma).
- Fuente de alimentación de 5V: Suministra energía al sistema.

Configuración de los Pines

Pin	Función
PD0-PD6	Segmentos del display de 7 segmentos
PB0-PB3	Control de los dígitos del display (multiplexación)
PB4	LED indicador de configuración de fecha
PB5	LED indicador de configuración de hora
PD7	LED indicador de configuración de alarma
PC2-PC4	Botones de configuración (modo, incremento, decremento)
PC5	Control del buzzer (alarma)
PC0-PC1	Botones de incremento/decremento.

Tabla 1. Configuración de pines.

Cálculos

En este proyecto, el Timer0 del microcontrolador ATmega328p se configura para generar interrupciones cada 10ms.

Cada interrupción de 10ms es utilizada para:

- Contar 100 interrupciones y generar 1 segundo.
- Controlar el parpadeo de los dos puntos del reloj (cada 500ms).
- Verificar si ha llegado la hora de la alarma.
- Implementar un sistema antirrebote para los botones.

La ecuación general para el **tiempo de desbordamiento** del Timer0 es:

$$TNCT0 = 256 - \frac{f_{clk} \cdot t_{deseado}}{prescalar}$$

Se necesita que el Timer0 tarde exactamente 10ms en llegar a 256, lo cual usamos los siguientes valores:

- Frecuencia del reloj del sistema: $f_{clk}=16,000,000$ Hz
- Prescaler usado: 1024
- Tiempo deseado de interrupción: 10ms = 0.01s

Sustituyendo los valores:

$$TNCT0 = 256 - \frac{0.01 \cdot 16,000,000}{1024}$$

$$TNCT0 \approx 99$$

Cada vez que ocurre la interrupción del Timer0, se recarga TCNT0 con 99 para que siempre tarde 10ms en desbordarse.

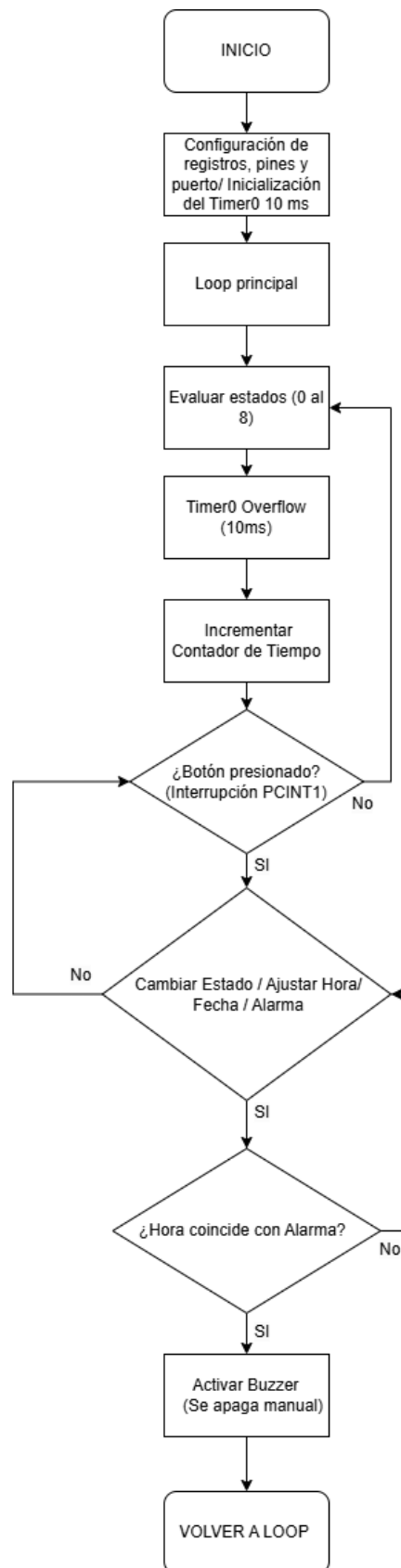


Figura 2. Diagrama de flujo del reloj.

Programación del Proyecto

La programación del reloj digital con alarma se realizó en Assembly para ATmega328P. Tal como se observa en la figura 2, en esta sección se explicará la programación por bloques de código.

Estructura General del Código:

1. Encabezado y Configuración del Hardware
2. Inicialización del Sistema
3. Bucle Principal (LOOP)
4. Manejo de Estados
5. Interrupciones
6. Multiplexación del Display
7. Configuración de la Alarma
8. Manejo del Timer0

A continuación, se explica el código siguiendo su estructura:

1. Encabezado y Configuración del Hardware

Se incluyen las definiciones del hardware y la configuración de variables.

Se incluye el archivo de definiciones del ATmega328P.

```
.include "M328PDEF.inc"
```

```
.cseg
```

Variables de Temporización

Estas variables controlan la actualización de la hora y el parpadeo de los dos puntos (:) en el display

.def cont500ms = R20 ; Contador para 500ms

.def cont1s = R19 ; Contador de 1 segundo

.def cont60s = R18 ; Contador de 60 segundos

Variables del Estado del Reloj

.def estado = R16 ; Estados del sistema

Estado controla la función actual del reloj, puede representar los siguientes modos:

- 0000 → Visualización de la hora.
- 0001 → Configuración de horas.
- 0002 → Configuración de minutos.
- 0003 → Visualización de la fecha.
- 0004 → Configuración del mes.
- 0005 → Configuración del día.
- 0006 → Visualización de la alarma.
- 0007 → Configuración de minutos de la alarma.
- 0008 → Configuración de horas de la alarma.

Variables del Tiempo (Horas, Minutos y Segundos)

.def useg = R21 ; Unidades de segundos

.def dseg = R22 ; Decenas de segundos

.def umin = R23 ; Unidades de minutos

.def dmin = R24 ; Decenas de minutos

.def uhor = R25 ; Unidad de hora

.def dhor = R26 ; Decena de horas

Variables de Fecha (Día y Mes)

.def mes_u = R0 ; Unidades del mes

.def mes_d = R1 ; Decenas del mes

.def dia_u = R2 ; Unidades del día

.def dia_d = R3 ; Decenas del día

Variables de la Alarma

.def alar_hor_d = R4 ; Decenas de horas de la alarma

.def alar_hor_u = R5 ; Unidades de horas de la alarma

.def alar_min_d = R6 ; Decenas de minutos de la alarma

.def alar_min_u = R7 ; Unidades de minutos de la alarma

.def alarma_activa = R8 ; 0 = OFF, 1 = ON

El ensamblador AVR no permite asignar valores inmediatos (LDI) a registros de propósito general de R0 a R15 directamente, por lo que se usa MOV para transferir valores desde otro registro, lo cual se usó el R28.

2. Inicialización del Sistema

En la subrutina MAIN, se configura el stack y se inicializan las variables.

MAIN:

; Configuración del Stack

LDI R16, LOW(RAMEND)

OUT SPL, R16

LDI R17, HIGH(RAMEND)

OUT SPH, R17

Se inicializa la pila de memoria para llamadas y retorno de funciones.

Configuración de pines

Puertos de salida:

- PD0-PD6 para los segmentos del display.
- PB0-PB3 para el control de los dígitos del display.
- PB4, PB5 y PD7 para los LEDs indicadores de configuración.

- PC5 para el buzzer de la alarma.

Puertos de entrada (con Pull-up activado):

- PC0, PC1, PC2 y PC3 para los botones de control del reloj.

Habilitación de interrupciones

SEI

3. Bucle Principal (LOOP)

El bucle principal revisa el estado actual del sistema y llama a la función correspondiente.

LOOP:

CPI estado, 0

BREQ ESTADO_0000 ; Estado de visualización de la hora

CPI estado, 1

BREQ ESTADO_0001 ; Configuración de horas

CPI estado, 2

BREQ ESTADO_0002 ; Configuración de minutos

RJMP LOOP

Cada estado tiene su propia rutina para mostrar información en el display o aceptar entrada.

Controlar los modos de operación:

- Visualización de hora (0000).
- Configuración de horas/minutos (0001, 0002).
- Visualización de fecha (0003).
- Configuración de fecha (0004, 0005).
- Visualización y configuración de la alarma (0006 - 0008).

4. Manejo de Estados

Cada estado se encarga de visualizar una parte diferente de la información en el display de 7 segmentos.

ESTADO_0000:

CALL DHOR_DISPLAY

CALL UHOR_DISPLAY

CALL DMIN_DISPLAY

CALL UMIN_DISPLAY

RJMP LOOP

Como se mencionó anteriormente hay 8 estados lo cual se configuran cada uno.

5. Interrupciones

Las interrupciones permiten cambiar estados y modificar la hora sin afectar el flujo del programa.

ISR_BOTONES: IN R17, PINC ; Leer los botones

SBIS PINC, PC2

CALL Cambiar_Estado_Principal

SBIS PINC, PC3

CALL Cambiar_Estado

RETI

- PC2 cambia entre Hora, Fecha y Alarma.
- PC3 cambia entre Configuración de Hora y Minutos.
- PC0/PC1 Incrementa y decrementa.

6. Multiplexación del Display

Los displays se controlan con una única línea de datos, encendiendo cada dígito de forma secuencial, se carga el valor del estado en el display.

7. Configuración de la Alarma

CHECK_ALARM: CP dhor, alar_hor_d

```

BRNE END_CHECK CP uhor, alar_hor_u
BRNE END_CHECK CP dmin, alar_min_d
BRNE END_CHECK CP umin, alar_min_u
BRNE END_CHECK
LDI R28, 1
MOV alarma_activa, R28
SBI PORTC, PC5

END_CHECK:
RET

```

Si la hora actual coincide con la alarma, se activa el buzzer.

8. Manejo del Timer0

El Timer0 se usa para controlar la actualización del reloj y el parpadeo de los dos puntos.

```

ISR_TIMER0_OVF:

INC cont500ms

CPI cont500ms, 50

BRNE END_ISR CLR cont500ms

SBI PINC, PC4

END_ISR: RETI

```

Cada 500ms, el Timer0 hace parpadear los dos puntos

Resumen de la programación.

Bloque	Función
Configuración del hardware	Define pines y registros
Inicialización del sistema	Configura el Stack y habilita interrupciones
Manejo de estados	Controla si se muestra la hora, fecha o alarma
Interrupciones	Detecta eventos en botones y Timer0

Multiplexación del display	Activa los dígitos en secuencia rápida
Manejo de la alarma	Compara la hora con la alarma y activa el buzzer

Tabla 2. Programación de bloques.

Diagrama de los estados

Tal como se observa en la siguiente figura 3, se muestra el diagrama general del proyecto, en el cual se destacan los estados más importantes: 0, 3 y 6. En estos estados se visualizan respectivamente la hora, la fecha y la alarma.

El proyecto está programado para funcionar como un reloj digital, permitiendo la configuración según los días del mes. Además, incluye una alarma programable, la cual debe apagarse de manera manual.

Diagrama de estados finitos

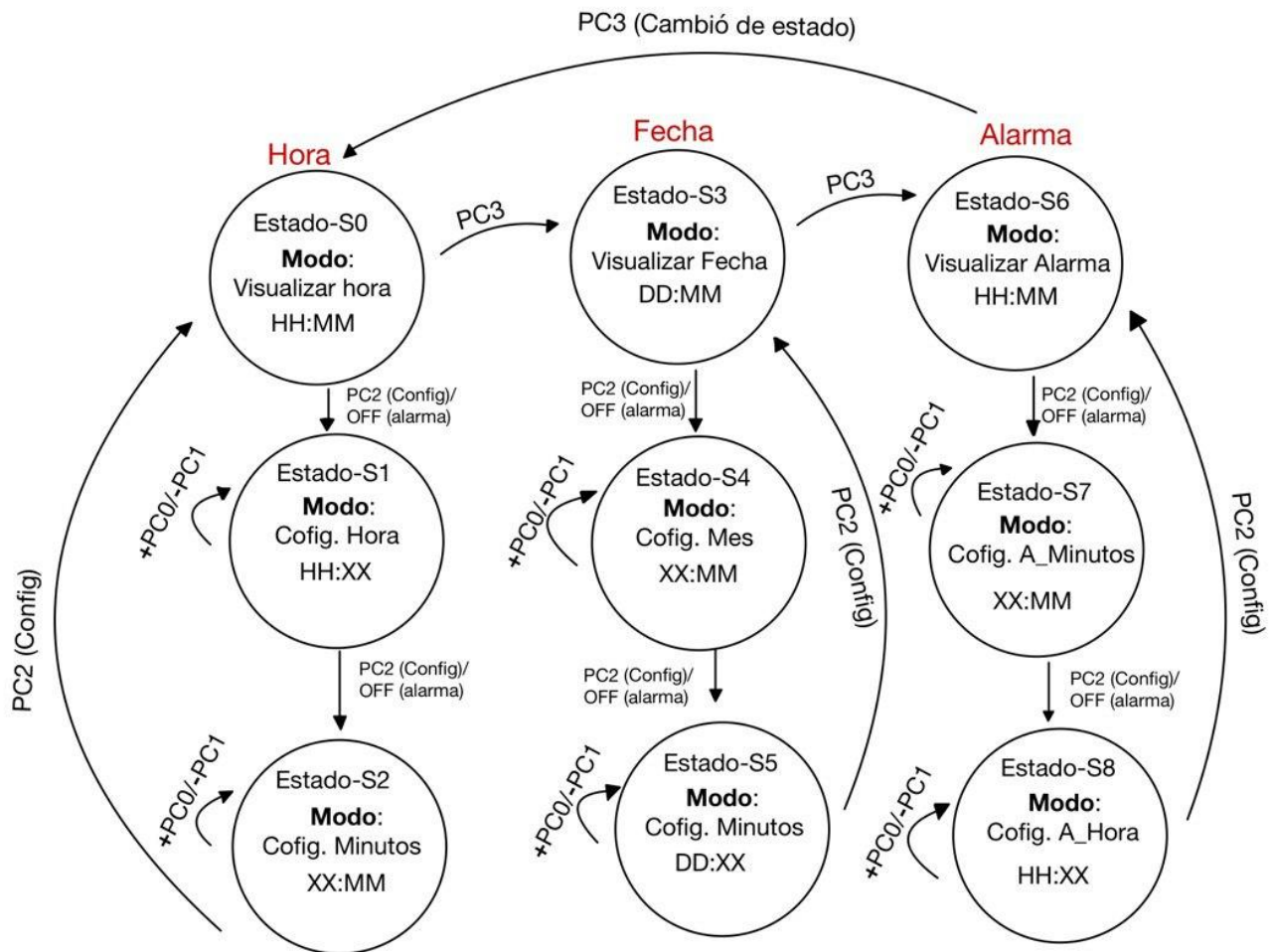


Figura 3. Diagrama de estados finitos.