Proyecto 1 – Reloj

Programación de microcontroladores
David Otoniel Carranza Méndez – 23979

Link: https://github.com/car23979/ProyectoNo1_Reloj.git

Link: https://youtu.be/YNLXwt6Rt64

El proyecto va de un reloj digital programable con capacidad para mostrar la hora, fecha y configurar una alarma, implementado en el microcontrolador **ATmega328P** (equivalente al Arduino Uno). El sistema integra múltiples subsistemas:

Temporización precisa mediante el uso de timers hardware.

Multiplexación de displays para reducir el uso de pines.

Gestión de interrupciones para manejar botones, actualizaciones de tiempo y alarmas.

Indicadores visuales (LEDs) para mostrar el modo de operación.

El código está escrito en ensamblador AVR, optimizado para aprovechar al máximo los recursos del microcontrolador. A continuación, se desglosan los componentes, cálculos y diagramas clave.

---

2. Metodología de Diseño

El sistema se divide en tres capas principales:

a. Capa de Hardware

Displays de 7 segmentos: Cuatro dígitos con ánodo común, controlados por transistores PNP.

Botones: Tres pulsadores para ajustar valores (incremento, decremento) y cambiar modos.

Buzzer: Zumbador activo controlado por un transistor NPN para generar la alarma.

LEDs indicadores: Cuatro LEDs para señalar el modo actual (reloj, fecha, configuración).

b. Capa de Firmware

Temporizadores:

Timer0: Gestiona la multiplexación de displays y el parpadeo de LEDs.

Timer1: Controla el incremento del tiempo (minutos, horas, días, meses).

Interrupciones:

PCINT0: Detecta cambios en los botones.

TIMER0_OVF y **TIMER1_OVF**: Manejan eventos periódicos.

c. Capa de Lógica de Control

Registros: Almacenan valores de tiempo actual (R19-R25) y alarma (R3, R5, R12, R13).

Máquina de estados: Nueve modos operativos gestionados por el registro R17.

---

3. Descripción Detallada del Hardware

a. Displays de 7 Segmentos

Configuración:

Segmentos (PORTD): PD0-PD7 controlan los segmentos A-G y el punto decimal.

Dígitos (PORTC): PC0-PC3 activan los transistores PNP que alimentan cada dígito.

Multiplexación:

Cada dígito se enciende durante 5 ms en secuencia, creando la ilusión de continuidad.

Frecuencia de refresco: 200 Hz (4 dígitos × 5 ms).

b. Circuito de Botones

Pines:

PB1 (Decremento): Pin físico 15.

PB2 (Incremento): Pin físico 16.

PB4 (Modo): Pin físico 18.

Pull-up interno: Habilitado para evitar estados flotantes.

Debouncing: Implementado por software con retardos de 50 ms.

c. Buzzer y Transistor

Configuración:

PB5 (D13): Controla la base del transistor NPN (2N2222).

Resistencia de base: 1kΩ para limitar la corriente.

Fuente externa: Utilizada si el buzzer requiere más de 5V.

---

4. Descripción Detallada del Firmware

a. Inicialización del Sistema

Stack Pointer (SP): Configurado en 0x03FF (final de la RAM).

Puertos I/O:

PORTD y PORTC: Salidas para displays.

PORTB: PB1, PB2 y PB4 como entradas con pull-up; PB0, PB3, PB5 como salidas.

Timers:

Timer0: Prescaler 64, recarga en 0xB2 para desbordamientos cada 5 ms.

Timer1: Prescaler 1024, recarga en 0x1B1E para desbordamientos cada 1 minuto.

b. Rutinas Clave

MULTIPLEX:

Alterna entre dígitos usando el contador R24.

Consulta la tabla TABLITA para convertir valores numéricos a segmentos.

CHECK_ALARM:

Compara los registros de tiempo actual (R19, R22, R23, R25) con los de la alarma (R3, R5, R12, R13).

Activa el buzzer si hay coincidencia.

Interrupciones:

TIMER0_OVF: Alterna dígitos y parpadea LEDs.

TIMER1_OVF: Incrementa minutos, horas, días y meses.

c. Máquina de Estados (Modos)

Modo	Descripción	LEDs Activos
0	Reloj Normal	Ninguno
1	Fecha Normal	PC4
2	Configurar Minutos Reloj	PB0
3	Configurar Horas Reloj	PB0 + PB3
4	Configurar Mes	PC5
5	Configurar Día	PC4 + PC5
6	Configurar Minutos Alarma	PB3
7	Configurar Horas Alarma	PC4 + PB3
8	Alarma Apagada	Ninguno

---

5. Cálculos Técnicos

a. Temporización del Timer0

Objetivo: Refrescar displays cada 5 ms.

Frecuencia del reloj: 1 MHz (prescaler global de 16).

Fórmula:

Ticks=FrecuenciaPrescaler×Tiempo=1,000,00064×0.005=78.125Ticks=PrescalerFrecuencia ×Tiempo=641,000,000×0.005=78.125

Valor de recarga:

TCNT0=256−78=178=0xB2TCNT0=256−78=178=0xB2

b. Temporización del Timer1

Objetivo: Incrementar minutos cada 1 minuto lógico.

Fórmula:

Ticks=1,000,0001024×60≈58,594Ticks=10241,000,000×60≈58,594

Valor de recarga:

c. Consumo Eléctrico

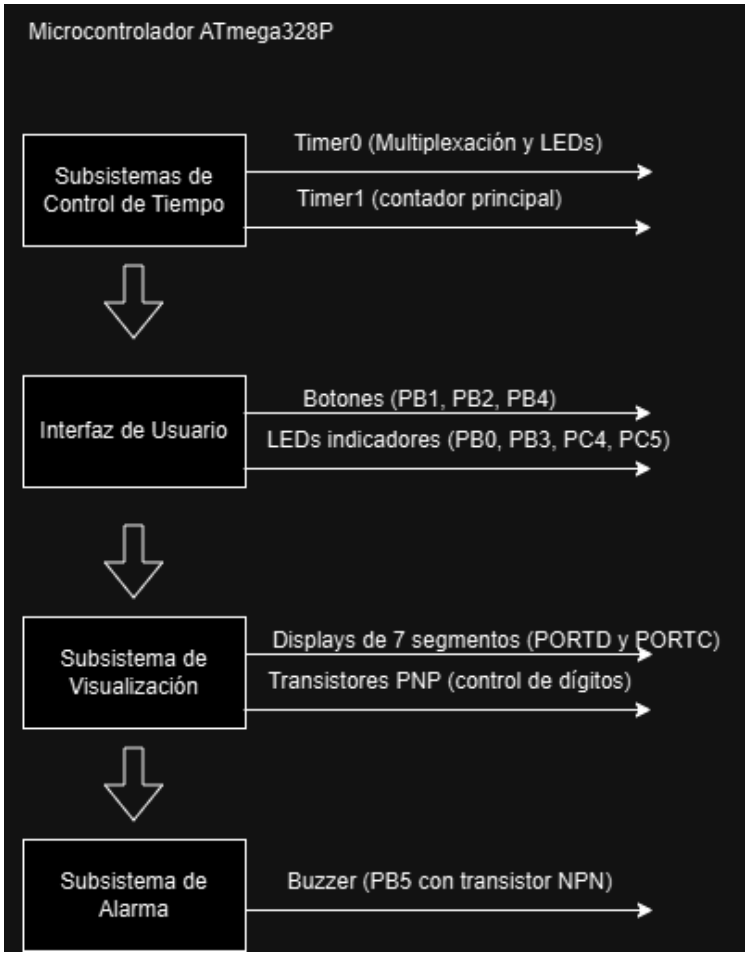
Corriente por segmento: 20 mA (típico para displays de 7 segmentos).

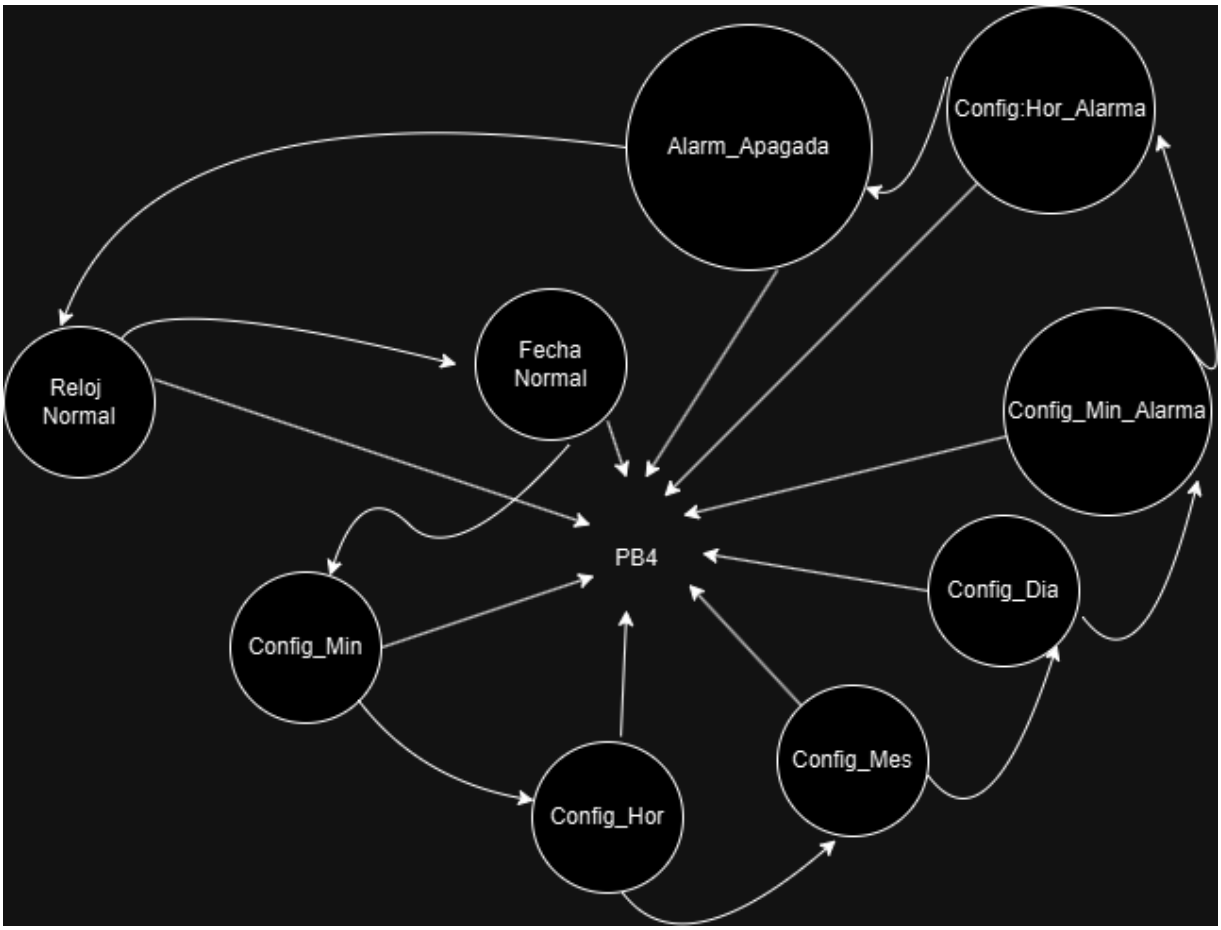
Corriente promedio:

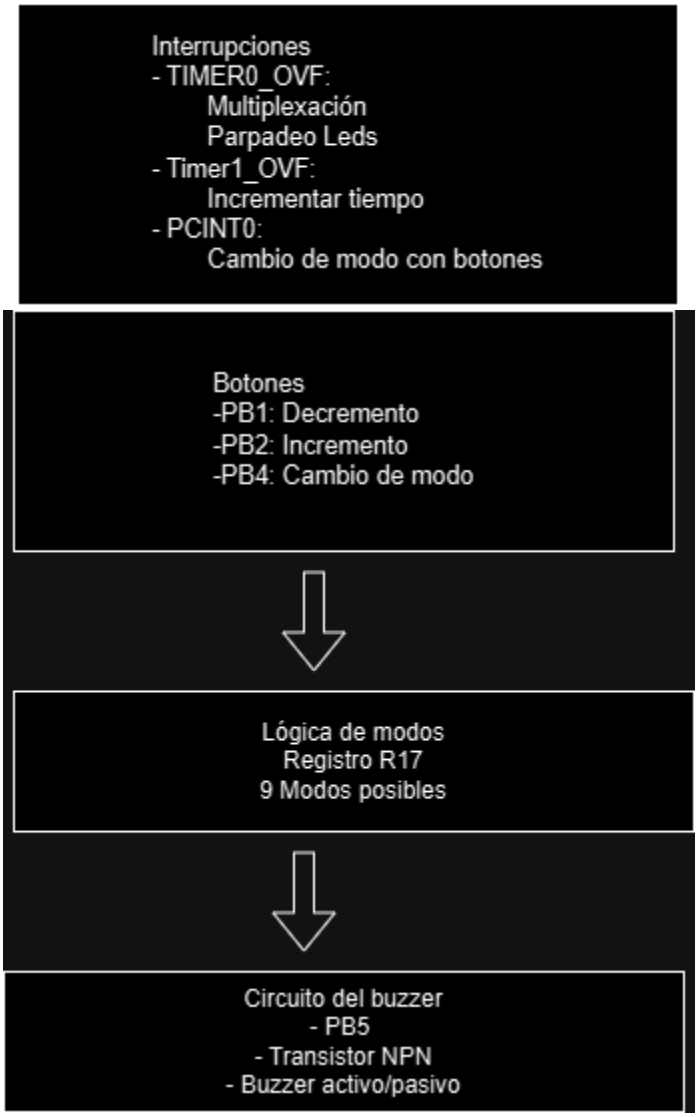
Iavg=20 mA×7×0.25=35 mA (por dı´gito)Iavg=20mA×7×0.25=35mA(por digito)

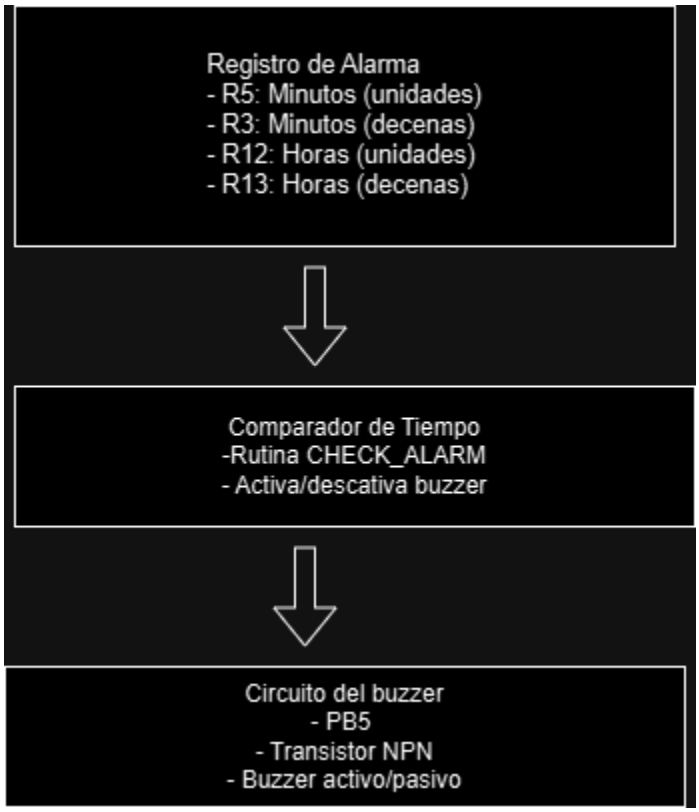
Transistor PNP: BC327 con β≥100β≥100, capaz de manejar hasta 500 mA.

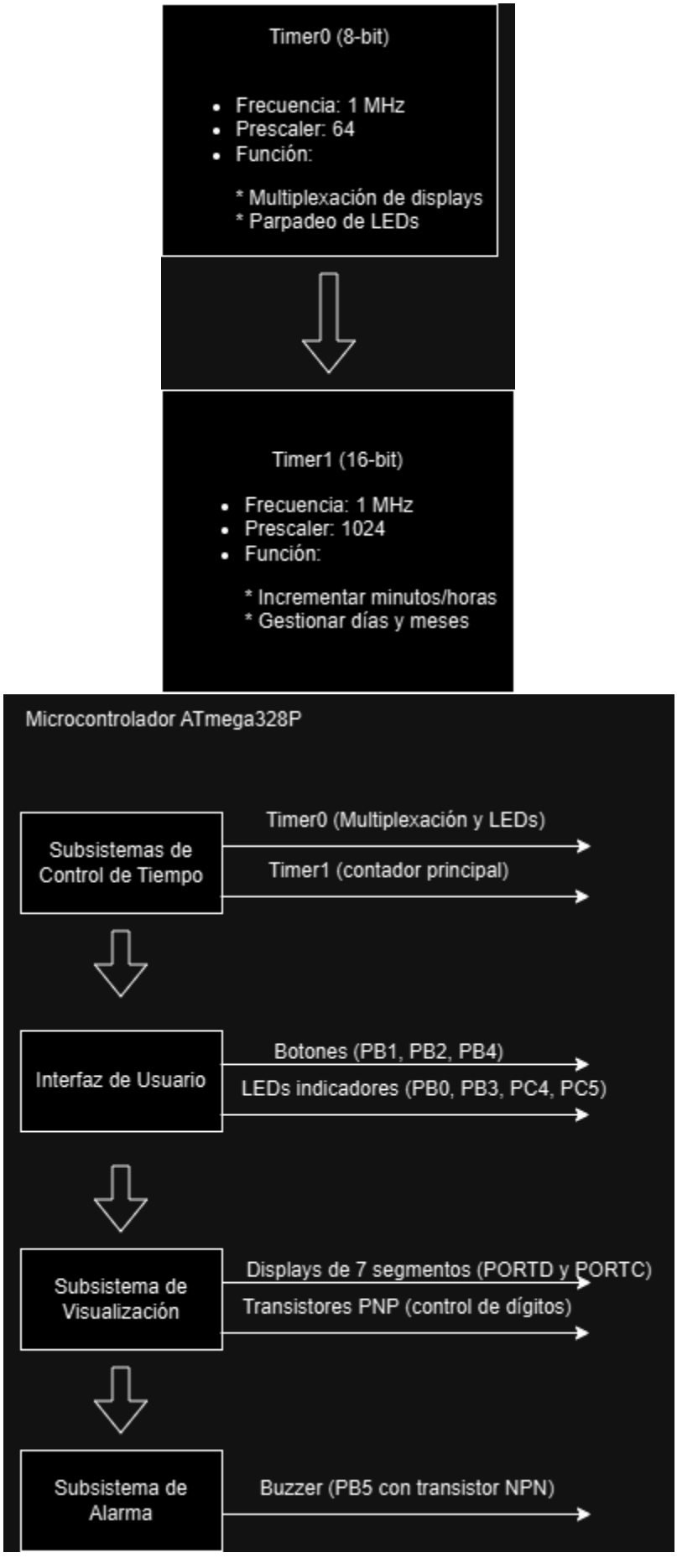
Diagramas:

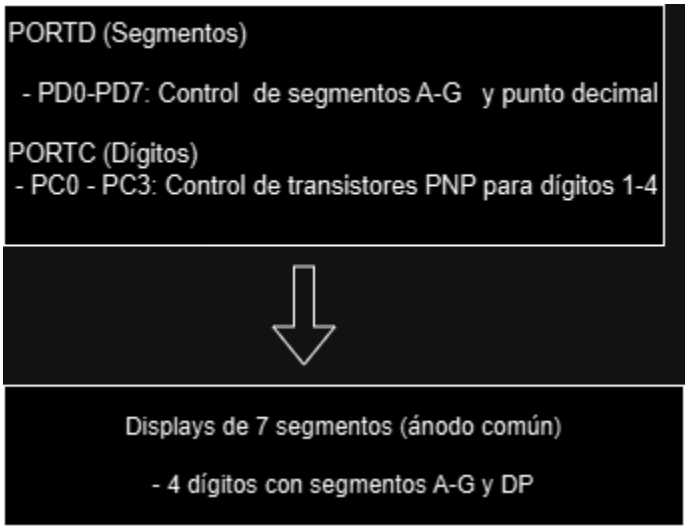

Microcontrolador ATmega328P
Subsistemas de Control de Tiempo
Timer0 (Multiplexación y LEDs)
Timer1 (contador principal)
Interfaz de Usuario
Botones (PB1, PB2, PB4)
LEDs indicadores (PB0, PB3, PC4, PC5)
Subsistema de Visualización
Displays de 7 segmentos (PORTD y PORTC)
Transistores PNP (control de dígitos)
Subsistema de Alarma
Buzzer (PB5 con transistor NPN)


Alarm_Apagada
Config:Hor_Alarma
Reloj
Normal
Fecha
Normal
Config_Min_Alarma
PB4
Config_Dia
Config_Min
Config_Mes
Config_Hor


Interrupciones
- TIMER0_OVF:
Multiplexación
Parpadeo Leds
- Timer1_OVF:
Incrementar tiempo
- PCINT0:
Cambio de modo con botones
Botones
-PB1: Decremento
-PB2: Incremento
-PB4: Cambio de modo
Lógica de modos
Registro R17
9 Modos posibles
Circuito del buzzer
- PB5
- Transistor NPN
- Buzzer activo/pasivo


Registro de Alarma
- R5: Minutos (unidades)
- R3: Minutos (decenas)
- R12: Horas (unidades)
- R13: Horas (decenas)
Comparador de Tiempo
-Rutina CHECK_ALARM
- Activa/descativa buzzer
Circuito del buzzer
- PB5
- Transistor NPN
- Buzzer activo/pasivo


Timer0 (8-bit)
Frecuencia: 1 MHz
Prescaler: 64
Función:
* Multiplexación de displays
* Parpadeo de LEDs
Timer1 (16-bit)
Frecuencia: 1 MHz
Prescaler: 1024
Función:
* Incrementar minutos/horas
* Gestionar días y meses
Microcontrolador ATmega328P
Subsistemas de Control de Tiempo
Timer0 (Multiplexación y LEDs)
Timer1 (contador principal)
Interfaz de Usuario
Botones (PB1, PB2, PB4)
LEDs indicadores (PB0, PB3, PC4, PC5)
Subsistema de Visualización
Displays de 7 segmentos (PORTD y PORTC)
Transistores PNP (control de dígitos)
Subsistema de Alarma
Buzzer (PB5 con transistor NPN)


PORTD (Segmentos)
- PD0-PD7: Control de segmentos A-G y punto decimal
PORTC (Dígitos)
- PC0 - PC3: Control de transistores PNP para dígitos 1-4
Displays de 7 segmentos (ánodo común)
- 4 dígitos con segmentos A-G y DP