

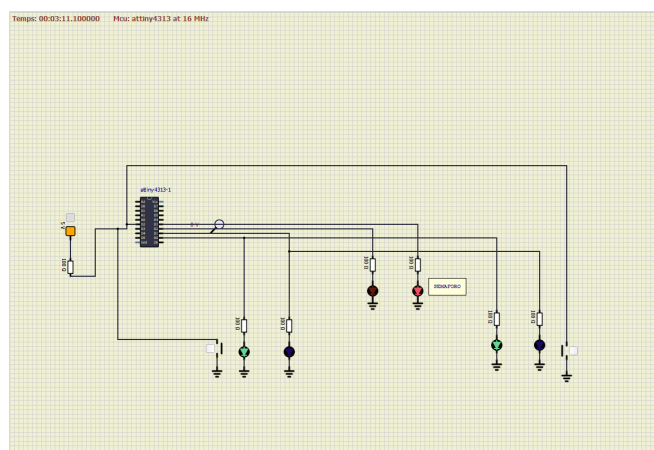
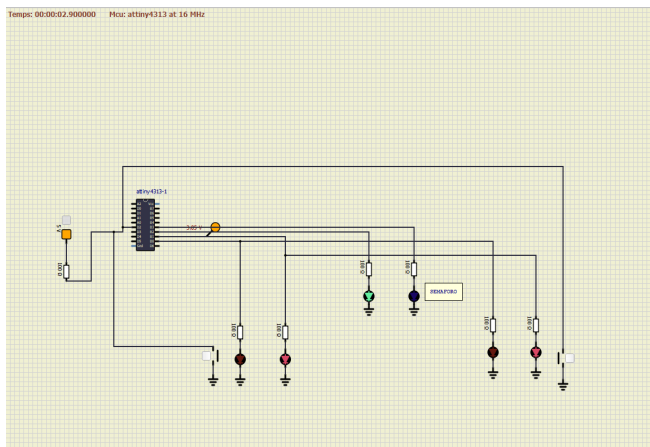
### Introducción:

Este código implementa un controlador de semáforos utilizando el microcontrolador ATtiny85. El proyecto tiene como objetivo demostrar la funcionalidad del microcontrolador, utilizar interrupciones y administrar una máquina de estados para controlar los semáforos. En el siguiente informe, proporcionaremos una visión general del desarrollo, análisis y hallazgos clave.

El ATtiny85 es un microcontrolador versátil que ofrece un conjunto de características ideales para aplicaciones de control de semáforos. El diagrama de bloques muestra cómo se conectan los periféricos esenciales, como los LED de los semáforos, al microcontrolador. La configuración de pines y las características eléctricas son críticas para garantizar un funcionamiento seguro y eficiente.

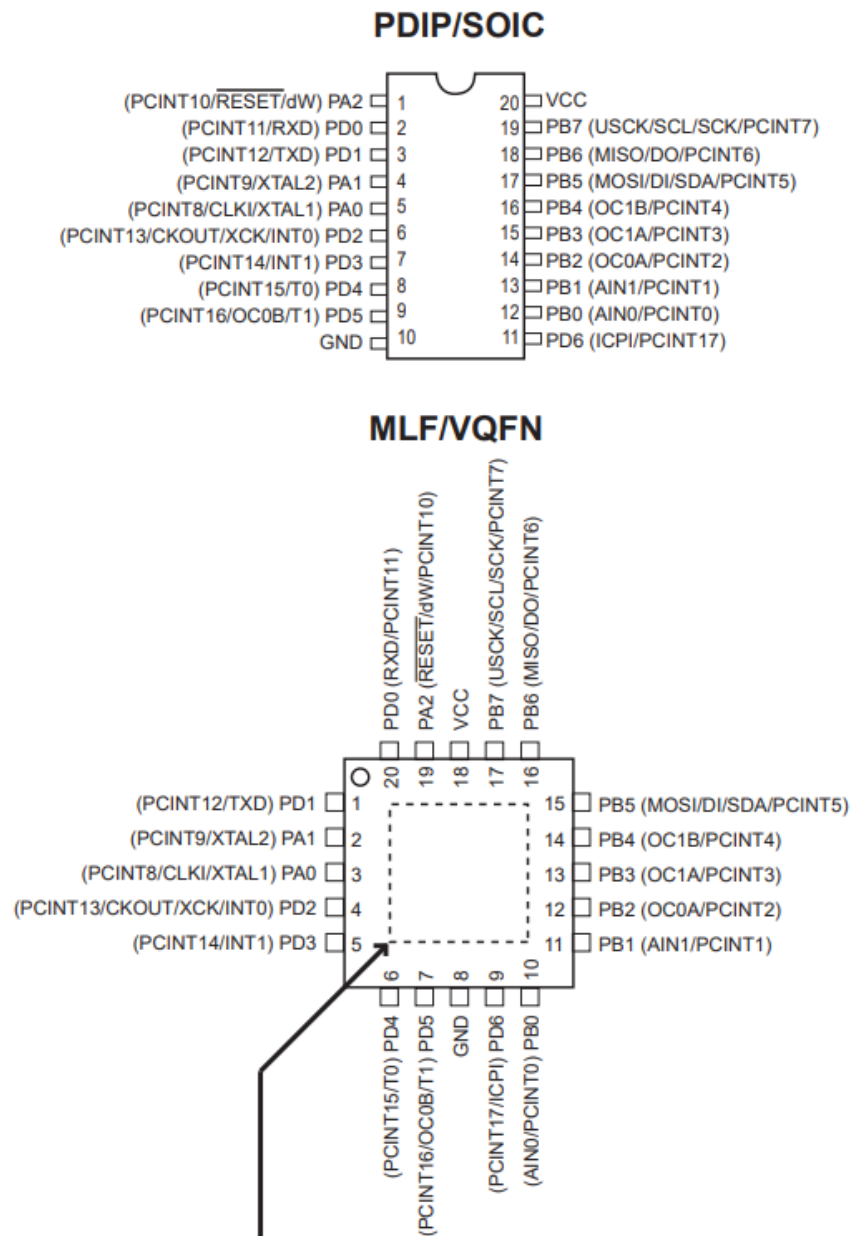
### Desarrollo y Análisis:

#### Capturas de Pantalla



## 1. Pin Configurations

Figure 1-1. Pinout ATtiny2313A/4313



## Laboratorio # 2

### GPIOs, Timers y FSM

#### 2.1 Block Diagram

Figure 2-1. Block Diagram

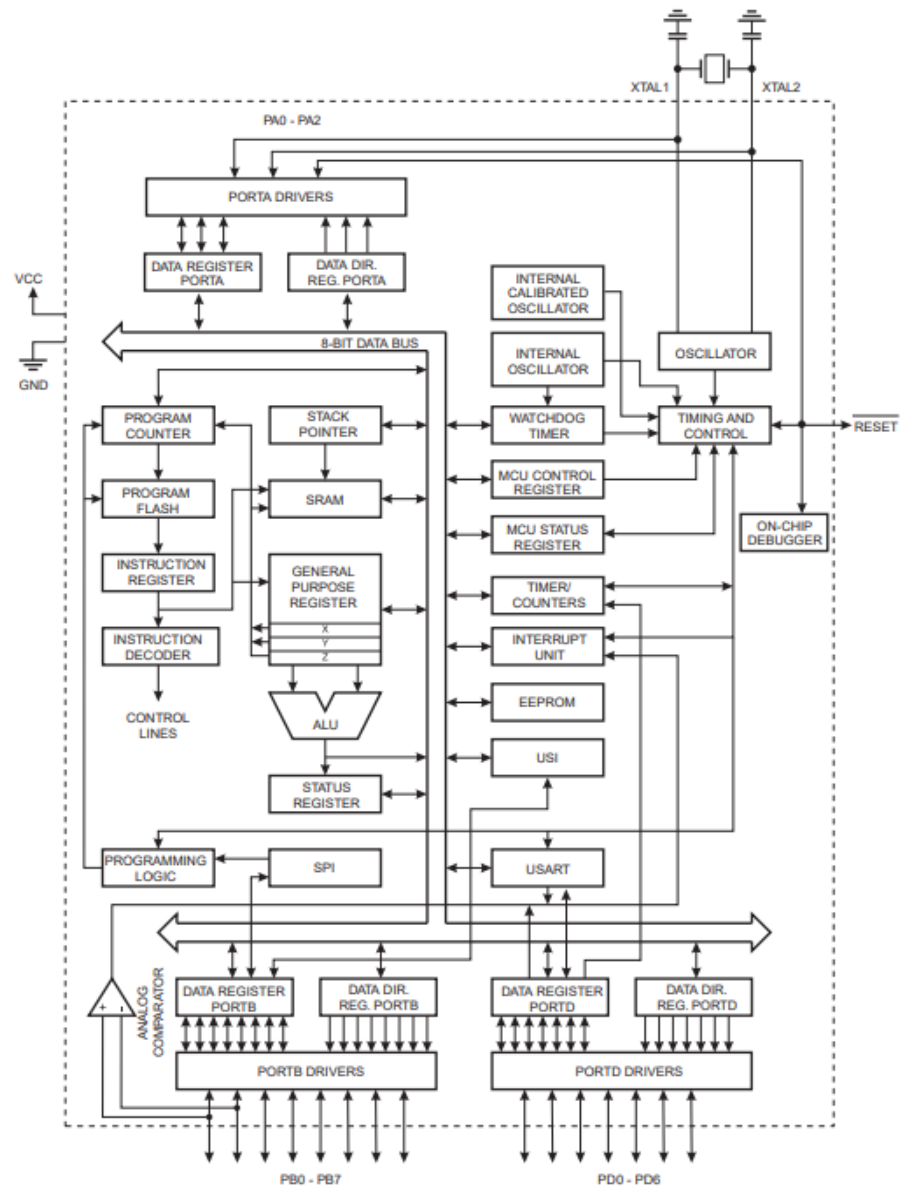
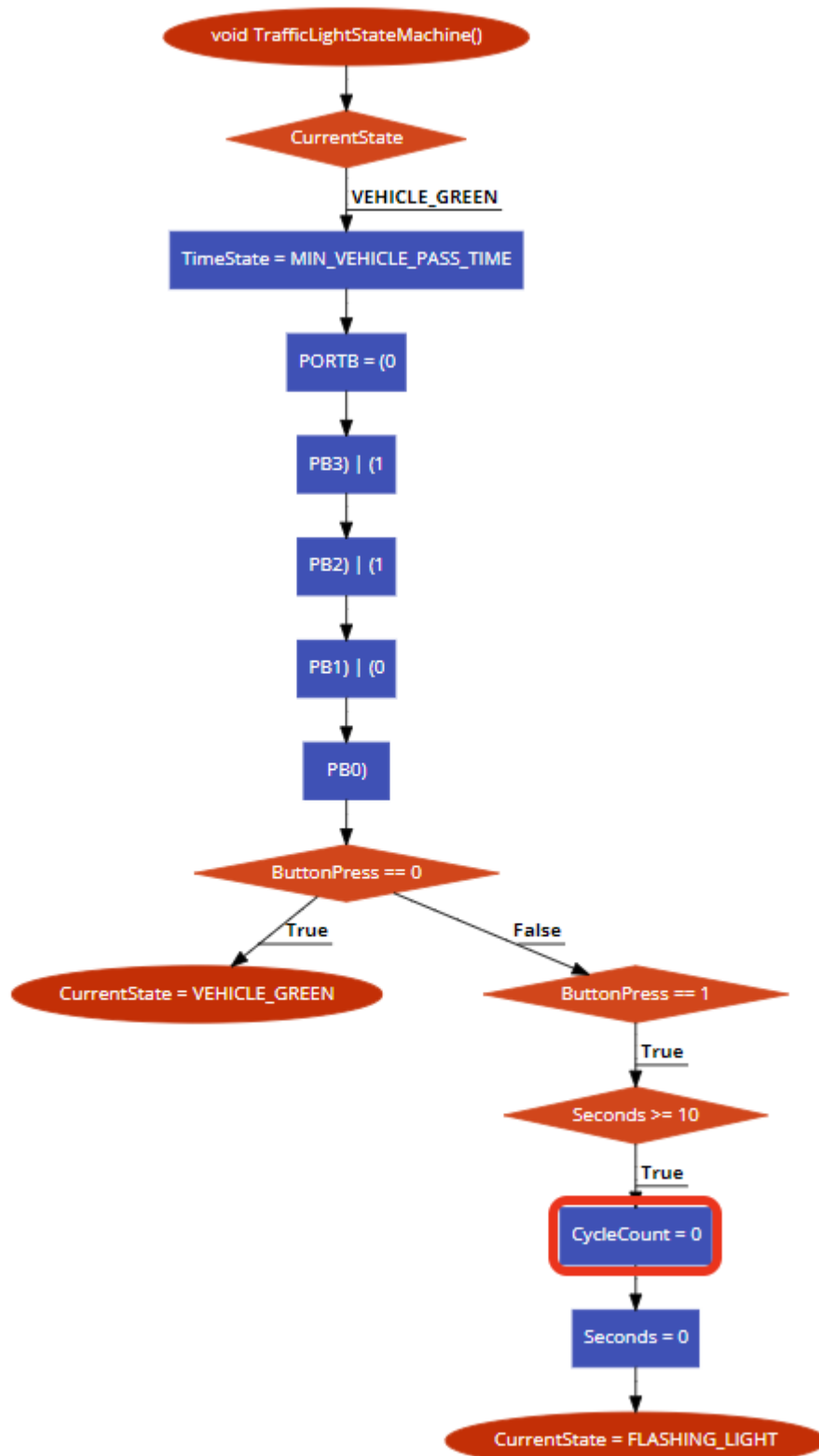
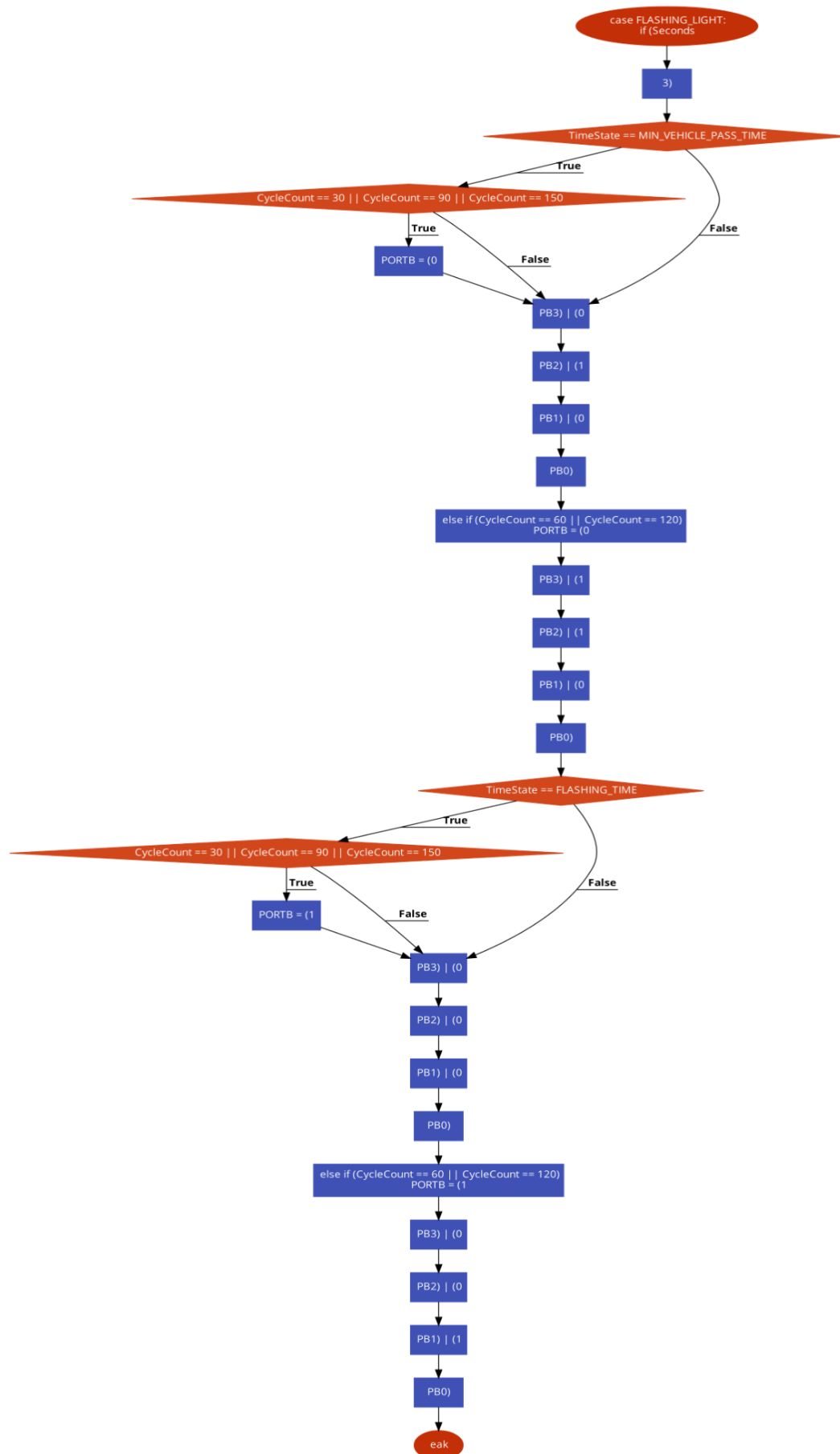


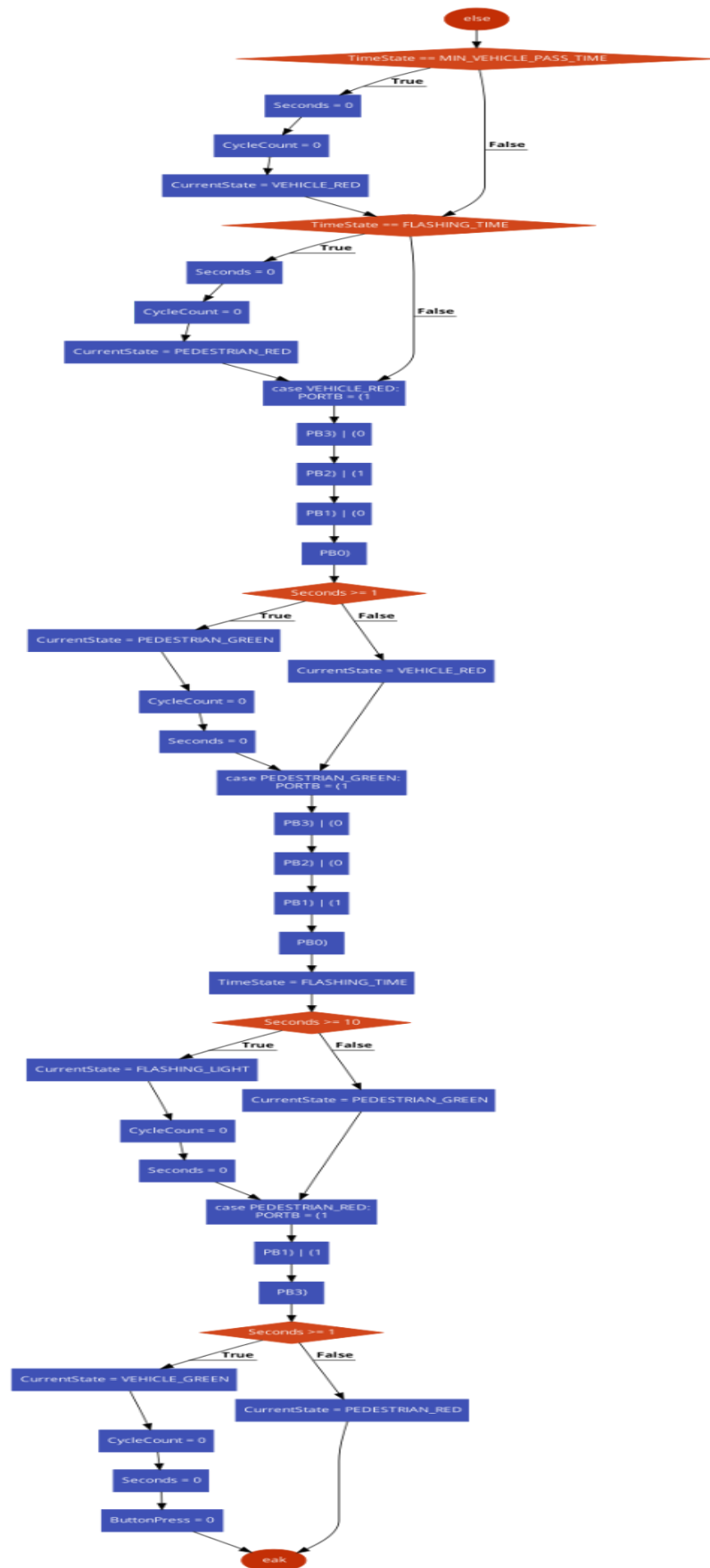
Diagram de flujo del case :



## Laboratorio # 2

### GPIOs, Timers y FSM





## **Detalle del código :**

Claro, voy a explicar en detalle el código proporcionado, centrándome en los aspectos clave y la lógica de programación.

El código implementa un controlador de semáforos con un microcontrolador ATtiny85. Aquí están los detalles:

### **\*\*Interrupciones:\*\***

El código utiliza dos interrupciones importantes: una para el botón (INT0\_vect) y otra para manejar el tiempo (TIMER0\_OVF\_vect). La interrupción del botón se utiliza para detectar cuándo se presiona el botón de control del semáforo. La interrupción de tiempo se utiliza para generar retardos y contar segundos.

### **\*\*Máquina de Estados:\*\***

El programa utiliza una máquina de estados para controlar las luces del semáforo en diferentes situaciones. Los estados principales son:

- **\*\*VEHICLE\_GREEN (LDPV):\*\*** En este estado, la luz del semáforo para vehículos está verde. Si se presiona el botón durante este estado y ha pasado un tiempo mínimo, se pasa al estado FLASHING\_LIGHT.
- **\*\*FLASHING\_LIGHT (LDP):\*\*** En este estado, las luces parpadean antes de cambiar a otros estados. Después de un tiempo, vuelven al estado VEHICLE\_RED o PEDESTRIAN\_RED según la situación.
- **\*\*VEHICLE\_RED (LDVD):\*\*** En este estado, la luz del semáforo para vehículos está en rojo, permitiendo que los peatones pasen.
- **\*\*PEDESTRIAN\_GREEN (LDPP):\*\*** En este estado, la luz del semáforo para peatones está en verde. Después de un tiempo, se vuelve al estado FLASHING\_LIGHT.
- **\*\*PEDESTRIAN\_RED (LDPD):\*\*** En este estado, la luz del semáforo para peatones está en rojo.

### **\*\*Temporización:\*\***

El código utiliza el contador de ciclos para contar el tiempo, y se basa en el hecho de que, con la configuración del temporizador, aproximadamente cada 63 ciclos se pasa un segundo. Esto se utiliza para controlar las transiciones de estado y las luces parpadeantes.

### **\*\*Configuración de Periféricos:\*\***

La función `TimerConfiguration` configura el temporizador (Timer0) para generar interrupciones periódicas que se utilizan para controlar el tiempo. La función `ExternalSwitchConfiguration` establece la configuración para el botón.

En resumen, el código implementa un controlador de semáforos con una máquina de estados que cambia las luces del semáforo en función del tiempo y las interrupciones del botón. Esto se logra mediante la gestión de estados y temporizadores controlados por interrupciones. El código es una representación práctica de cómo se pueden utilizar microcontroladores como el ATtiny85 para controlar dispositivos en el mundo real.

## **Conclusión:**

En resumen, este proyecto de controlador de semáforos utiliza el microcontrolador ATtiny85 de manera efectiva para gestionar secuencias de semáforos. La máquina de estados programada permite transiciones precisas entre fases de vehículos y peatones, incluido un estado de luz intermitente.

El análisis detallado confirma que el circuito funciona de manera segura y eficiente. Este proyecto demuestra la aplicación práctica de microcontroladores y diseño de circuitos, y cómo estos componentes pueden utilizarse en aplicaciones del mundo real.

Los apéndices proporcionan hojas de datos para referencia, lo que enriquece la comprensión de los componentes utilizados en el proyecto.

## **Bibliografía :**

<https://www.gadgetronicx.com/attiny85-timer-tutorial-generating-time-delay-interrupts/>

<https://www.gadgetronicx.com/attiny85-external-pin-change-interrupt/>