



DISEÑO DE SISTEMAS BASADOS EN MICROPROCESADOR

→ Práctica 3 Laboratorio:

Esta práctica se divide en dos partes; la primera consiste en el diseño de un circuito referente a un semáforo sobre la protoboard y la placa Nucleo F411RE, compuesto por tres leds para el control de paso de vehículos, dos leds para el control de peatones y un botón que permita el control del semáforo.

La segunda parte consiste en controlar el funcionamiento del semáforo mediante una máquina de estados finita (FSM), haciendo uso del lenguaje de programación C y el software STM32CubeMX junto con un IDE de desarrollo (Visual Studio Code + PlatformIO, TrueStudio Atollic, Keil, ...) con el fin de obtener el siguiente comportamiento.

El paso de vehículos siempre estará en verde, y solo se pondrá en rojo cuando se pulse el botón. En este caso, tras 3 segundos, se pondrá en verde el paso de peatones y rojo el de vehículos, 15 segundos después parpadeará y, 3 segundos después, el sistema volverá a su estado inicial. La secuencia será:

- Pulsar botón del semáforo
- Esperar 3 segundos
- Pasar el control de vehículos del verde al amarillo en la calle principal
- Esperar 3 segundos
- Poner el semáforo de vehículos en rojo y el de peatones en verde.
- Espera 15 segundos
- Poner semáforo de peatones en verde intermitente
- Esperar 3 segundos
- Poner semáforo de peatones en rojo y el principal en verde

Se deberá entregar el código desarrollado junto al circuito desarrollado explicándolo brevemente.



→ Resultados:

■ Código principal:

```
switch (modo) {
    default:
        //Encender Verde Coches
        HAL_GPIO_WritePin(D10_Led_GPIO_Port, D10_Led_Pin, GPIO_PIN_SET);
        //Encender Rojo Peatones
        HAL_GPIO_WritePin(D11_Led_GPIO_Port, D11_Led_Pin, GPIO_PIN_SET);
        //Si se pulsa el botón
        if(GPIOA->IDR&GPIO_IDR_ID0_Msk){
            modo = 1;
        }
        break;
    case 1:
        counter_parpadeo = 0;
        //Esperamos 3 segundos
        HAL_Delay(3000);
        // Apagamos verde coches
        HAL_GPIO_WritePin(D10_Led_GPIO_Port, D10_Led_Pin, GPIO_PIN_RESET);
        //Encendemos amarillo coches
        HAL_GPIO_WritePin(D9_Led_GPIO_Port, D9_Led_Pin, GPIO_PIN_SET);
        //Esperamos 3 segundos
        HAL_Delay(3000);
        modo = 2;
        break;
    case 2:
        //Apagar amarillo coches
        HAL_GPIO_WritePin(D9_Led_GPIO_Port, D9_Led_Pin, GPIO_PIN_RESET);
        //Encendemos rojo coches
        HAL_GPIO_WritePin(D8_Led_GPIO_Port, D8_Led_Pin, GPIO_PIN_SET);
        //Apagamos rojo peatones
        HAL_GPIO_WritePin(D11_Led_GPIO_Port, D11_Led_Pin, GPIO_PIN_RESET);
        //Encendemos verde peatones
        HAL_GPIO_WritePin(D12_Led_GPIO_Port, D12_Led_Pin, GPIO_PIN_SET);
        //Esperamos 15 segundos
        HAL_Delay(15000);
        modo = 3;
        break;
```



```

case 3:
//while para el parpadeo
while(counter_parpadeo<15){

    //Apagar verde peatones
    HAL_GPIO_WritePin(D12_Led_GPIO_Port, D12_Led_Pin, GPIO_PIN_RESET);
    HAL_Delay(100);
    //Encendemos verde peatones
    HAL_GPIO_WritePin(D12_Led_GPIO_Port, D12_Led_Pin, GPIO_PIN_SET);
    HAL_Delay(100);
    counter_parpadeo++;
}
modo = 4;
break;
case 4:
//Apagar verde peatones
HAL_GPIO_WritePin(D12_Led_GPIO_Port, D12_Led_Pin, GPIO_PIN_RESET);
//Apagar rojo coches
HAL_GPIO_WritePin(D8_Led_GPIO_Port, D8_Led_Pin, GPIO_PIN_RESET);
//Encendemos Verde Coches
HAL_GPIO_WritePin(D10_Led_GPIO_Port, D10_Led_Pin, GPIO_PIN_SET);
//Encendemos Rojo Peatones
HAL_GPIO_WritePin(D11_Led_GPIO_Port, D11_Led_Pin, GPIO_PIN_SET);
modo = 0;
break;
}

```



■ Imagen del circuito:

Hemos utilizado el botón que viene con la shell debido a que con el botón directamente en la protoboard no hemos conseguido que funcione, quizá por problemas eléctricos.

