



# DISEÑO DE SISTEMAS BASADOS EN MICROPROCESADOR

## → Práctica 7 Laboratorio:

Esta práctica es continuación del anterior y su finalidad es mostrar el estado del semáforo por una pantalla LCD controlada por arduino y a su vez enviar información a un dispositivo móvil a través de un emisor bluetooth.

El envío y recepción de datos e información se realiza a través de una UART configurada en la placa STM32F411RE.

### ■ El funcionamiento del semáforo debe ser el siguiente:

El paso de vehículos siempre estará en verde, y solo se pondrá en rojo cuando se pulse el botón. En este estado se enviará periódicamente el mensaje *rojo* mediante Bluetooth al dispositivo móvil conectado al emisor y la pantalla LCD estará en rojo mostrando el mensaje *Pulse 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.
  - Se comprueba si existen vehículos cercanos al sensor ultrasónico (menos de 400 mm).
  - Si existen vehículos cercanos se esperan los 3 segundos.
  - Si no existen vehículos cercanos no se esperan los 3 segundos.
- Pasar el control de vehículos del verde al amarillo en la calle principal, se envía el mensaje *amarillo* al dispositivo móvil y la pantalla LCD se cambia a color azul y se muestra el mensaje *Espere Verde*.
- Esperar 3 segundos.
- Poner el semáforo de vehículos en rojo y el de peatones en verde, bajar la barrera, además se manda al dispositivo móvil la palabra *verde* y la pantalla LCD pasa a color verde mostrando el mensaje *Pase*.
- Espera 15 segundos.
- Poner semáforo de peatones en verde intermitente.
- Esperar 3 segundos.
- Poner semáforo de peatones en rojo y el principal en verde y subir la barrera.

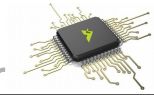


## → Resultados:

### → Código principal:

#### ➤ Código del while:

```
while (1){
    htim2.Instance->CCR1 = 75;
    HAL_Delay(1000);
    GPIOB->ODR |= GPIO_ODR_OD6_Msk; //Encender Verde Coches
    GPIOA->ODR |= GPIO_ODR_OD7_Msk; //Encender Rojo Peatones
    HAL_UART_Transmit(&huart1, "Rojo\n", 5, 1000); //Mandar Rojo
    if(pulsado == 1){
        modo = 1;
        control = 1;
        counter_parpadeo = 0;
        ultrasonidos();
        while(control==1){
            switch(modo){
                case 1:
                    counter_parpadeo=0;
                    if(cercanos == 1){ //Si existen vehículos <400 mm.
                        HAL_Delay(3000);
                    }
                    GPIOB->ODR&=~GPIO_ODR_OD6_Msk; //Apagar verde coches
                    GPIOC->ODR |= GPIO_ODR_OD7_Msk; //Encender amarillo coches
                    HAL_UART_Transmit(&huart1, "Amarillo\n", 9, 1000);
                    HAL_UART_Transmit(&huart1, 1, 1, 1000);
                    HAL_Delay(3000);
                    modo=2;
                    break;
                case 2:
                    GPIOC->ODR &=~ GPIO_ODR_OD7_Msk; //Apagar amarillo coches
                    GPIOA->ODR |= GPIO_ODR_OD9_Msk; //Encender rojo coches
                    htim2.Instance->CCR1 = 25;
                    GPIOA->ODR &=~ GPIO_ODR_OD7_Msk; //Apagamos rojo peatones
                    GPIOA->ODR |= GPIO_ODR_OD6_Msk; //Encender verde peatones
                    HAL_UART_Transmit(&huart1, "Verde\n", 6, 1000);
                    HAL_UART_Transmit(&huart1, 2, 1, 1000);
                    HAL_Delay(15000);
                    modo=3;
                    break;
```



```

case 3:
while(counter_parpadeo<15){
    GPIOA->ODR &=~ GPIO_ODR_OD6_Msk; //Apagar verde peatones
    HAL_Delay(100);
    GPIOA->ODR |= GPIO_ODR_OD6_Msk; //Encender verde peatones
    HAL_Delay(100);

    counter_parpadeo++;
}
modo=4;
break;
case 4:
GPIOA->ODR &=~ GPIO_ODR_OD6_Msk; //Apagar verde peatones
GPIOA->ODR &=~ GPIO_ODR_OD9_Msk; //Apagar rojo coches
GPIOB->ODR |= GPIO_ODR_OD6_Msk; //Encendemos verde coches
GPIOA->ODR |= GPIO_ODR_OD7_Msk; //Encendemos rojo peatones
modo=1;
control=0;
pulsado=0;
cercanos = 0;
times = 0;
break;
}
}
}

```

Con respecto a la anterior práctica, hemos marcado en azul las nuevas instrucciones que contiene el código. Todas ellas se corresponden con los envíos de mensajes tanto al emisor Bluetooth como a la placa arduino que controla el LCD.

- ➔ En la primera instrucción solo mandamos al emisor Bluetooth ya que la pantalla LCD está en estado rojo por defecto y no es necesario mandarle el mensaje.
- ➔ En las siguientes instrucciones:
  - Las que mandan mensajes con cadenas de texto se corresponden con el emisor Bluetooth.
  - Las que mandan mensajes con un valor se corresponden con la placa Arduino.
    - Una vez que la placa recibe estos valores, actúa en consecuencia de los mismos para cambiar el estado de la placa LCD.



■ Código de Arduino para controlar el LCD:

```
#include <Wire.h>
#include "rgb_lcd.h"
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

rgb_lcd lcd;
int opcion;
void setup()
{
    lcd.begin(16, 2);
    mySerial.begin(9600);
    lcd.clear();
    lcd.print("Pulse boton");
    lcd.setRGB(255, 0, 0);
}

void loop()
{
    opcion = -1;
    if(mySerial.available()>0){
        opcion = mySerial.read();
        switch(opcion){
            case 0:
                lcd.clear();
                lcd.print("Espere verde");
                lcd.setRGB(0, 0, 255);
                break;
            case 2:
                lcd.clear();
                lcd.print("Pase");
                lcd.setRGB(0, 255, 0);
                break;
```

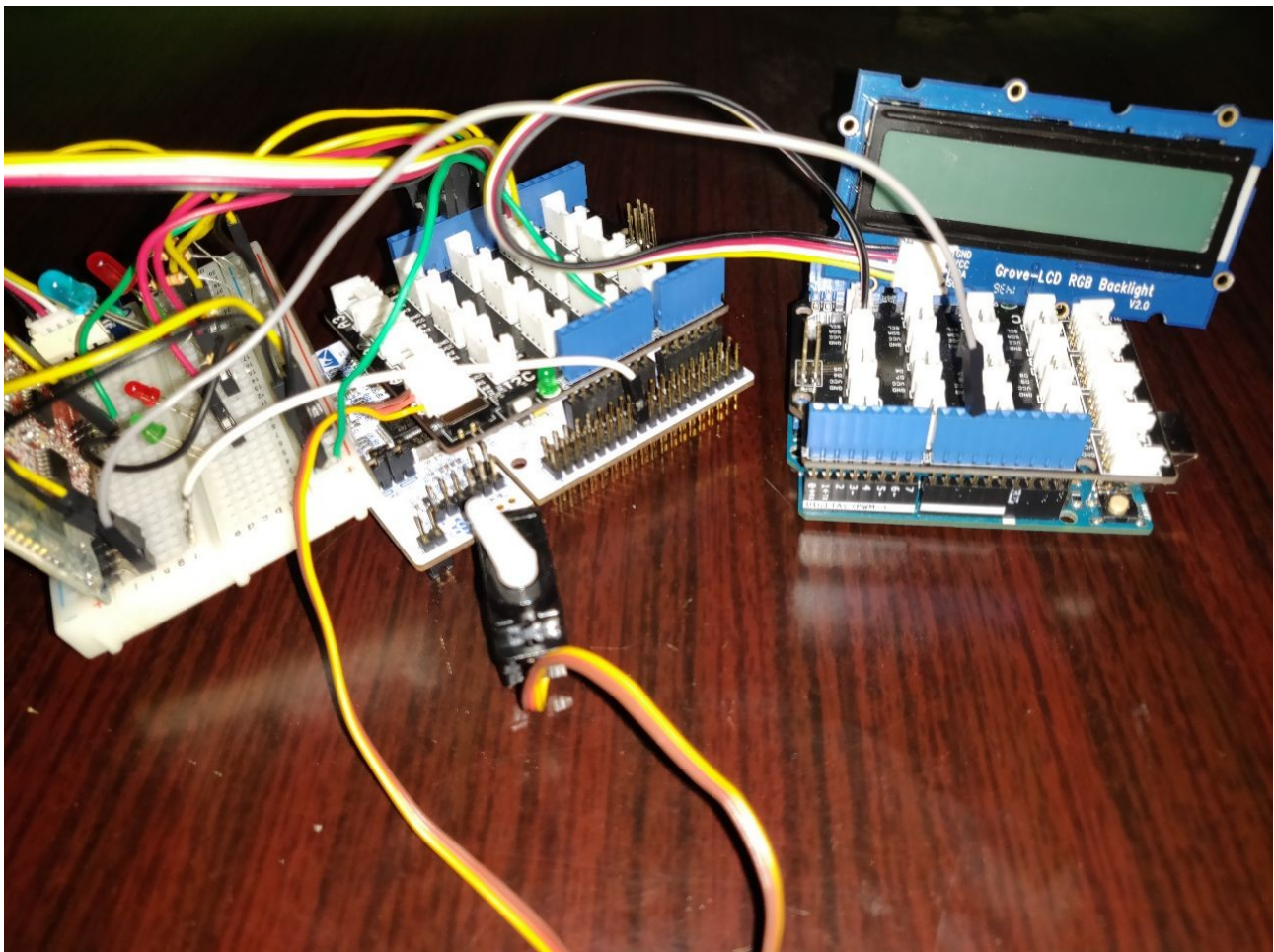


```
default:
    lcd.clear();
    lcd.print("Pulse boton");
    lcd.setRGB(255, 0, 0);
    break;
}
}
```

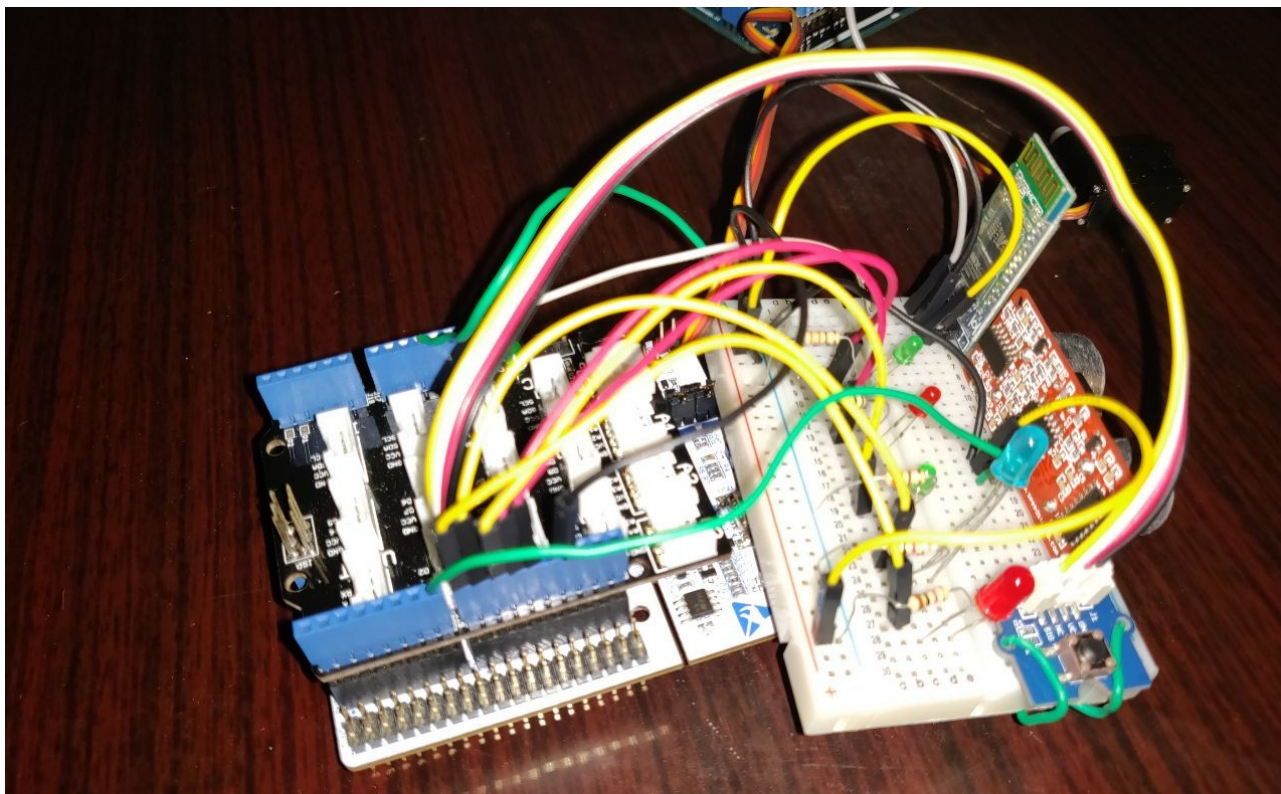
Aquí podemos observar como se gestiona la recepción de los mensajes en la placa arduino.

Con el switch controlamos el estado de la pantalla LCD según el valor que se reciba como hemos visto antes en el código principal del programa.

### → Imagen del Circuito:







Este es el circuito que hemos diseñado para la práctica. Consta de 5 LEDs y un botón que representan la estructura del semáforo. El LED azul representa la luz ámbar y el botón que hemos utilizado ha sido el de la shell.

La configuración de los cables es la siguiente:

- Los cables amarillos son de alimentación para el semáforo de coches.
- Los cables rojos son de alimentación para el semáforo de peatones y para el sensor de ultrasonidos.
- El cable negro es el cable de tierra al que se conectan las resistencias de los LEDs y el sensor de ultrasonidos.
- El cable verde es el que controla la emisión y recepción de señales del sensor de ultrasonidos y otro da corriente al sensor de ultrasonido y al emisor Bluetooth.
- El cable blanco conecta el puerto TX de la placa principal con el RX del emisor Bluetooth.
- El cable gris conecta el RX del emisor Bluetooth con el RX de la placa Arduino (Como un puente para repartir desde el TX de la placa principal a los 2).