Manual de Usuario – Sistema de Control de LEDs con Botón, PWM y UART

Autor: Juan Esteban Mora Diaz
Plataforma: STM32 Nucleo-L476RG
Fecha: 22/05/2025

Descripción general del sistema

Este sistema embebido fue diseñado para correr en la placa Nucleo STM32L476RG y tiene como objetivo:

- Generar una señal de latido (heartbeat) en el LED LD2 de la placa.
- Encender un LED externo (LED1) por 3 segundos mediante la pulsación de un botón y recibir un comando para modificar el estado del mismo (LED ON/OFF toggle).
- Controlar un segundo LED externo (LED2) mediante señal PWM.
- Recibir comandos UART desde el PC para modificar el estado de los LEDs.
- Notificar por UART los eventos ocurridos en el sistema.

Todo el código está escrito en C puro, accediendo directamente a los registros del microcontrolador.

Componentes requeridos

Componente	**Cantidad**
STM32 Nucleo-L476RG	1
LEDs	2

Resistencias 220Ω-300Ω	2
Cables jumper (macho-hembra)	4
PC con terminal serial	1

Conexiones físicas

Funcion	**Pin STM32**	**Arduino Pin**	**Descripción**
LED1 (ON/OFF) Y toggle	PA7	D11	Controlado por botón (3 s) y por comando “t” y “T”
LED2 (PWM)	PA6	D12	Controlado vía UART
Botón B1	PC13	—	Botón azul en la placa
UART	PA2/PA3	—	Virtual COM (USB-STLink)
GND	—	GND	Común para ambos LEDs

Observación: Conecta cada LED en serie con una resistencia de entre 220Ω-300Ω entre su cátodo y GND.

Flujo del programa

Inicio

1. Se inicializan todos los periféricos: GPIO, SysTick, UART, TIM3.
2. Se apagan ambos LEDs externos.
3. Se envía por UART el mensaje: `Sistema iniciado.`

Heartbeat

- Cada 500 ms se alterna el estado del LED LD2 (PA5).
- Funciona como indicador de actividad del sistema.

Botón B1 (PC13)

- Al presionar el botón azul, se genera una interrupción EXTI13.
- La interrupción llama a la función `room_control_on_button_press()`.
- Esta enciende LED1 (PA7) por 3 segundos y envía por UART: `Botón B1: Presionado.`

Control de tiempo (timeout)

- Luego de 3 segundos, LED1 se apaga automáticamente.
- Se envía por UART: `LED externo apagado (timeout).`

Comunicación UART

El sistema acepta los siguientes comandos desde el PC:

Comando	Acción	Repuestos UART
`h / H`	PWM 100% en LED2 (PA6)	PWM LED: 100%.
`l / L`	PWM 0% (LED2 apagado)	PWM LED: 0%.
`t / T`	Alterna el LED1 ON/OFF del PWM	PWM LED: ON (toggle) ó OFF
`Otro`	Cualquier otro carácter	Comando desconocido.

Archivos y estructura

Archivo	Contenido principal

`main.c`	Inicialización del sistema y bucle principal
`room_control.c/h`	Lógica principal del sistema
`uart.c/h`	Comunicación UART (TX/RX, interrupciones)
`gpio.c/h`	Configuración de pines y operaciones básicas
`nvic.c/h`	Configuración de interrupciones NVIC/EXTI
`systick.c/h`	Temporizador para retardos y control de tiempo
`tim.c/h`	PWM por hardware usando TIM3_CH1 (PA6)

Diagrama de flujo

Para una mayor comprensión y profundización de la lógica del código, adjunto un link en donde puede observar un diagrama de flujo del código principal, bastante detallado hecho en Mermaid.
Haz click aquí:
https://www.mermaidchart.com/raw/5a97f637-c8e8-4000-b95b-f46ebcf91269?theme=light&version=v0.1&format=svg

Y si quieres interactuar con el código de Mermaid del diagrama de flujo aquí lo tienes:

```
flowchart TD
    A(["Inicio del sistema"])
    A --> B["room_control_app_init()"]
    B --> C["Apagar LED PA7"]
    C --> D["Set PWM PA6 = 0%"]
    D --> E["UART: Sistema iniciado"]
    E --> F(["Ciclo principal (room_control_app_update)"])

    F --> G{"¿Han pasado 500 ms?"}
    G -- Sí --> H["Alternar LED LD2 (PA5)"]
    H --> I["Actualizar last_heartbeat_tick"]
    G -- No --> I

    I --> J{"¿LED PA7 está encendido y pasaron 3 s?"}
    J -- Sí --> K["Apagar LED PA7"]
    K --> L["led_on = 0"]
    L --> M["UART: LED apagado (timeout)"]
    M --> F
    J -- No --> F

    %% Interrupción botón
    N(["Interrupción botón B1"])
    N --> O{"¿Anti-rebote OK?"}
    O -- Sí --> P["Encender LED PA7"]
    P --> Q["led_on = 1"]
    Q --> R["Guardar tick actual"]
    R --> S["UART: Botón presionado"]
    S --> F
    O -- No --> F

    %% Interrupción UART
    T(["Interrupción UART"])
    T --> U{"¿Comando recibido?"}
    U -- h/H --> V["PWM PA6 = 100%"]
    V --> W["UART: PWM 100%"]
    W --> F
```

```
U -- l/L --> X["PWM PA6 = 0%"]
X --> Y["UART: PWM 0%"]
Y --> F

U -- t/T --> Z{"¿LED PA7 apagado?"}
Z -- Sí --> AA["Encender LED PA7"]
AA --> AB["UART: Encendido (toggle)"]
AB --> F

Z -- No --> AC["Apagar LED PA7"]
AC --> AD["UART: Apagado (toggle)"]
AD --> F

U -- Otro --> AE["UART: Comando desconocido"]
AE --> F
```

Instrucciones de uso

1. Conecta la placa Nucleo a tu PC por USB.
2. Abre un monitor serial.
3. Configura:
    a. **Baudrate:** 115200
    b. **Bits de datos:** 8
    c. **Paridad:** Ninguna
    d. **Bits de parada:** 1
    e. **Control de flujo:** Ninguno
4. Observa el LED LD2 parpadeando.
5. Presiona el botón B1 para encender LED1.

6. Envía comandos por UART para controlar el LED2.

7. ¡Interactúa y Divierte!