

Práctica 5

Manejo de la sección de E/S del microcontrolador ESP32 (II)

Objetivo: El alumno se familiarizará con la configuración y uso de puertos mediante el uso de los registros de bajo nivel del periférico de entrada y salida.

Equipo: - Computadora Personal y tarjeta de desarrollo del ESP32.

Teoría: - Técnicas de Charlieplexing

Desarrollo:

Utilizar la implementación de las funciones de la Práctica # 4:

1. *gpioInitError_t* **gpio_init**(*gpioNum_t* gpioNum, *gpioMode_t* gpioMode)
2. *void* **gpio_write**(*gpioNum_t* gpioNum, *gpioLevel_t* gpioLevel)
3. *gpioLevel_t* **gpio_read**(*gpioNum_t* gpioNum)

Implementar el juego del “gato encerrado” (tic tac toe). Para esto es necesario implementar las siguientes funciones:

1. *eButtonState_t* **checkButton**(*void*)

Función que retorna el estado del boton, con las siguientes posibles acciones:

- *eBtnLongKeyPress*: cuando se presione por mas de 1000 ms.
- *eBtnDoubleKeyPress*: cuando se presione dos veces en menos de 500 ms después de soltarlo la primera vez.
- *eBtnShortKeyPress*: cualquier otra acción sobre el botón que no sea los previos.
- *eBtnUndefined*: valor por defecto del estado del boton cuando aun no se clasifica la accion.

2. *bool* **playSequence**(*eGameState_t* gameState)

Función que desplegará las siguientes animaciones:

- *eStalemate*: Desplegar un patrón de 'X' alternando entre los dos colores (1000 ms encendido y 500 ms apagado).
- *eRedPlayerWin*: Parpadear todo el tablero de color rojo (1000 ms encendido y 500 ms apagado)
- *eGreenPlayerWin*: Parpadear todo el tablero de color verde (1000 ms encendido y 500 ms apagado)

Estas animaciones se deberán desplegar sobre el tablero dado en Fig. 1

3. *eGameState_t* **checkBoard**(*sBoardState_t* *boardState, *eButtonState_t* buttonState)

Función que concentra la lógica del juego, contiendo la responsabilidad de posicionar el cursor en una casilla disponible y evaluar el estado actual del tablero para determinar si aun continua el juego, ya hay un ganador o hubo un empate (*eOngoingGame*, *eStalemate*, *eRedPlayerWin*, *eGreenPlayerWin*).

4. void **displayBoard**(sBoardState_t *boardState)

Despliega el estado actual del tablero en base a la matriz lógica contenida en *boardState* \rightarrow *gameBoard*[][]; y así mismo parpadear (500 ms encendido y 100 ms apagado) el LED correspondiente a la posición actual del jugador. Desplegándolo sobre el esquemático de la Fig. 1.

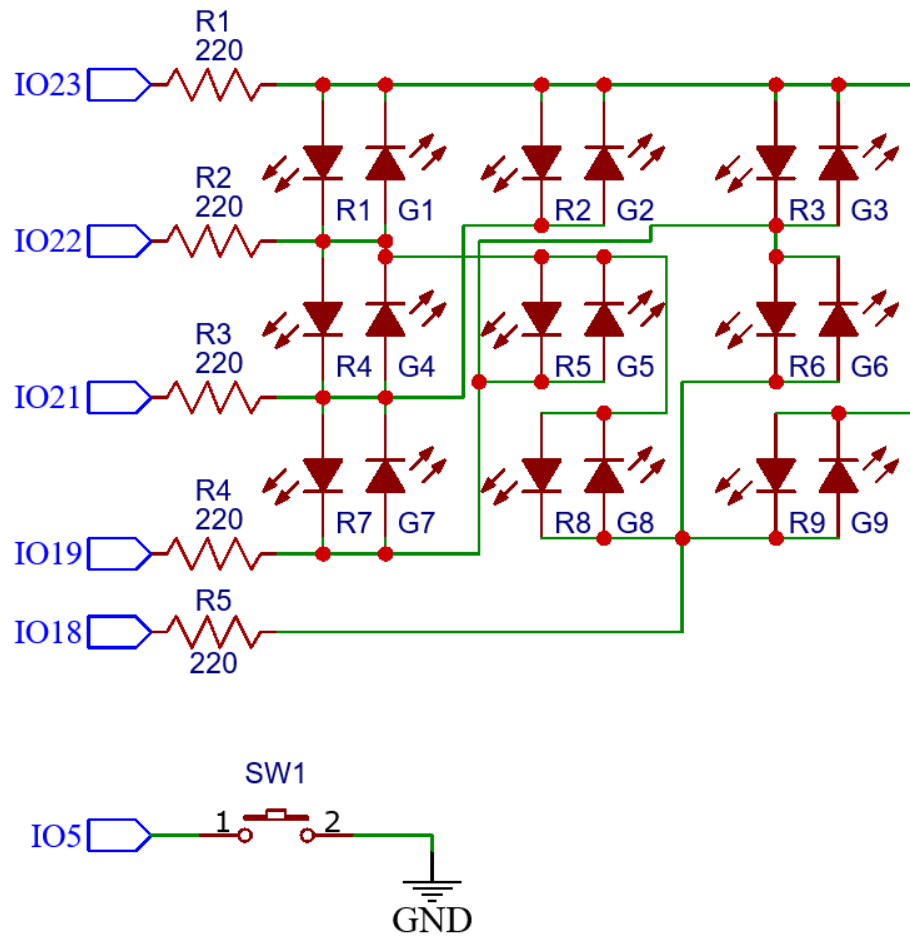


Figure 1: Esquemático

Apoyarse con el siguiente circuito simulado para verificar su funcionalidad:

<https://wokwi.com/projects/344015998533239378>

Comentarios y Conclusiones.

Bibliografía y Referencias.