Universidad de Costa Rica

Escuela de Ingeniería Eléctrica

IE0624 - Laboratorio de Microcontroladores

Informe del Laboratorio 1

Introducción a microcontroladores y manejo de GPIOS

Profesor

Marco Villalta Fallas

Jose Daniel Blanco Solis, B71137

III 2024

17 de Enero de 2025

Índice

Resumen

Este proyecto describe la implementación de un sistema basado en el microcontrolador PIC12F683, un botón y una matriz de LEDs, el cual se diseño para mostrar números aleatorios utilizando un registro de desplazamiento con retroalimentación lineal, cuando se presiona el botón, el microcontrolador genera un número aleatorio entre 1 y 6 y enciende el conjunto correspondiente de LEDs para representar el número.

1. Introducción

El objetivo de este laboratorio fue desarrollar un sistema de dado digital utilizando el microcontrolador PIC12F683, varios LEDs para la visualización de números y un botón como entrada para simular el lanzamiento del dado, con esta práctica, se introdujo el manejo del simulador, los pines GPIO y la metodología de desarrollo en el curso, se implementó un algoritmo basado en un registro de desplazamiento de retroalimentación lineal para la generación de números pseudoaleatorios, obteniendo un sistema capaz de mostrar un número aleatorio entre 1 y 6 con el conjunto de LEDs al presionar el botón y apagar los LEDs después de un breve retraso.

2. Nota teórica

2.1. Microcontrolador PIC12f683

El PIC12f683 es un microcontrolador de 8 bits de la familia PIC de microchip, conocido por su simplicidad, ofrece una memoria flash de 1.75 KB, 128 bytes de EEPROM y 128 bytes de RAM, posee una arquitectura harvard lo que le permite una operación eficiente, también incluye un oscilador interno de 4 MHz, simplificando el diseño del circuito.

2.2. Configuración de periféricos

Se desactivaron el watchdog timer (WDT) y la función de master clear reset (MCLR) mediante la configuración del registro CONFIG, como se muestra en el siguiente bloque de código:

```
1 typedef unsigned int word;
2 word __at 0x2007 __CONFIG = (_WDTE_OFF & _MCLRE_OFF);
```

El WDT es un temporizador que reinicia el microcontrolador si este deja de responder, mientras que MCLR proporciona una opción de reinicio manual, ambas funciones se deshabilitaron para evitar reinicios no deseados durante el desarrollo y prueba del sistema.

2.3. Generación de números pseudoaleatorios

La generación de números pseudoaleatorios se implementó utilizando un registro de desplazamiento de retroalimentación lineal (LFSR), dado que se considero el método mas eficiente, el LFSR opera sobre 16 bits y utiliza operaciones de desplazamiento y XOR para producir nuevas secuencias cada vez que se presiona el botón, el resultado se limita para el rango entre 1 y 6.

2.4. Diseño del circuito

El circuito incluye un botón conectado a GPIO4 para proporcionar la entrada de usuario y 7 LEDs conectados a los pines GPIO para mostrar los números generados, las resistencias limitadoras de corriente protegen los LEDs y el microcontrolador de daños por sobrecorriente.

Debido a las limitaciones en la cantidad de pines GPIO disponibles, se optó por una configuración de LEDs en la que el central está conectado a un puerto, mientras que los de los bordes están agrupados de manera que dos LEDs comparten la misma conexión, lo que permitio permitio mostrar los números del 1 al 6 al activar combinaciones específicas de pines GPIO.

El diseño contempla el uso de 7 LEDs, a pesar de que el sistema podría funcionar con 6 LEDs, el uso de un séptimo LED en el centro mejora la apariencia visual, dado que el costo de agregar un LED adicional es mínimo, esta decisión fue justificada para mejorar la calidad del proyecto, los LEDs que comparten conexiones están ubicados en posiciones opuestas del dado, lo que permite una mejor distribución de la luz y una apariencia más convincente de los números representados.

Para proteger tanto los LEDs como el microcontrolador, se integraron resistencias en serie con cada LED, estas limitan la corriente que pasa a través de los LEDs, asegurando que no exceda los

límites máximos de corriente del microcontrolador, que es de 25 mA por pin, además, se utilizó una resistencia en la conexión del botón para limitar la corriente cuando este se activa, evitando daños potenciales al microcontrolador.

El valor de la resistencia se calculó para garantizar que la corriente no supere los 25 mA cuando se utiliza una fuente de 5V. Mediante la ecuación $V = IR$, se determinó que se necesita una resistencia de 200 . Para una mayor seguridad, esta resistencia fue dividida en dos, de manera que se distribuyera mejor la carga y se minimizara el riesgo de picos de corriente, el esquema del circuito se muestra en la siguiente imagen.

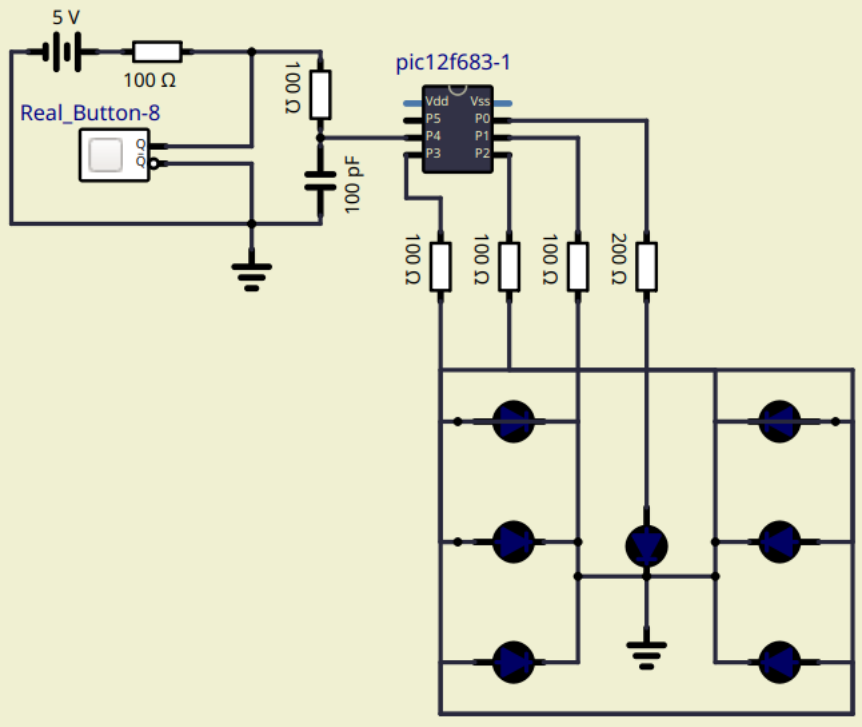

Figura 1: Circuito eléctrico.

3. Desarrollo/Análisis

3.1. Diagrama de flujo

El flujo del programa incluye la inicialización del microcontrolador, donde se configuran los pines GPIO y se desactivan las funciones no esenciales, el sistema espera a que el botón sea presionado, genera un número aleatorio mediante LFSR y enciende los LEDs correspondientes, después de un retraso, los LEDs se apagan y el programa vuelve a esperar una nueva entrada.

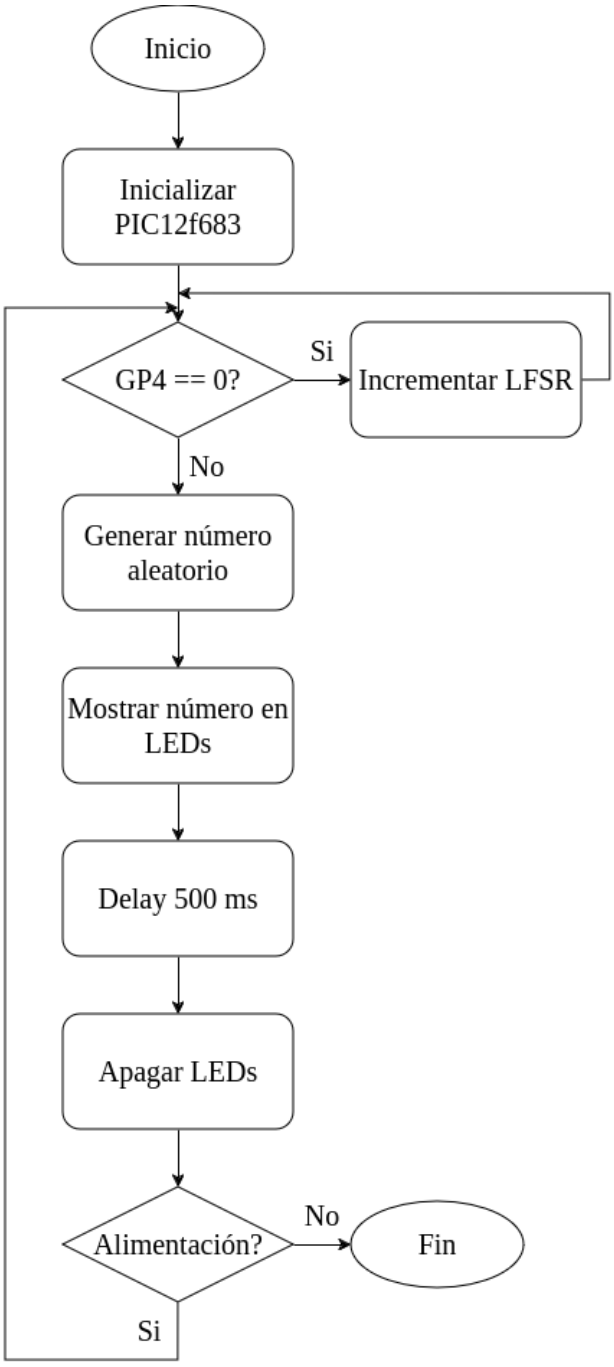

Figura 2: Diagrama de flujo del proceso.

3.2. Código

El código se estructuró con varias funciones, `initPIC12f683` configura el microcontrolador, asegurando que todos los pines estén en el estado adecuado antes de la operación, `generarNumeroAleatorio` usa un LFSR para generar números pseudoaleatorios, mientras que `mostrarNumero` controla los LEDs para mostrar el número generad, estas implementaciones se muestran a continuacion:

```
void initPIC12f683() {
    TRISIO = 0x10;
    ANSEL = 0x00;
    GPIO = 0x00;
}
```

```
uint8_t generarNumeroAleatorio() {
    unsigned lsb = lfsr & 0x01;  // Se obtiene el bit menos
       significativo
    lfsr >>= 1;                  // Se desplaza el LFSR a la derecha
    if (lsb) {
        lfsr ^= 0xEF01u;         // Se aplica XOR con la m scara si
           el bit menos significativo es 1
    }
    return (lfsr % 6) + 1;       // Se retorna un valor entre 1 y 6
}
```

```
void mostrarNumero(int numero) {
    GPIO &= 0b11100000;
    switch (numero) {
        case 1: GPIO |= 0b11100001; break;
        case 2: GPIO |= 0b11100010; break;
        case 3: GPIO |= 0b11100011; break;
        case 4: GPIO |= 0b11100110; break;
        case 5: GPIO |= 0b11100111; break;
        case 6: GPIO |= 0b11101110; break;
    }
}
```

El sistema respondió según lo esperado, generando y mostrando números aleatorios, cada vez que se presionaba el botón, los LEDs correspondientes se encendieron y apagaron después de un retraso, proporcionando una visualización clara del número generado, se observó que la distribución de números era adecuada.

4. Conclusiones y Recomendaciones

La implementación del LFSR para la generación de números pseudoaleatorios fue eficaz ante la falta de funciones definidas para este fin, para futuras mejoras, se podría implementar un debounce más robusto utilizando un temporizador en lugar de un simple retardo, también se recomienda investigar técnicas de inicialización de la semilla para mejorar la aleatoriedad de los números generados.

Referencias

[1] Microchip (2007) PIC12F683. Recuperado de: https://www.microchip.com/en-us/product/PIC12F683