EIE

Escuela de
Ingeniería Eléctrica

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

UNIVERSIDAD DE COSTA RICA

IE0624 - Laboratorio de Microcontroladores

LABORATORIO 1 : Introducción a microcontroladores y manejo de GPIOS

Oscar Fallas Cordero. B92861

Entrega: 3/Septiembre/2023

---

1. Introducción

El siguiente laboratorio permite un entendimiento inicial del uso de los microcontroladores, los cuáles en los últimos años han tomado gran valor debido al avance de la tecnología y la necesita de tener software diferentes dispositivos. Durante la práctica se inició estudiando el manejo de los GPIOS o salidas y entradas de propósito general, los cuáles tienen el fin de proporcionar entradas o salidas digitales manipuladas por medio de software. Por otra parte el laboratorio introduce al estudiante a la programación orientada al microcontrolador PIC realizando la lógica para la construcción de un dado de LED electrónico, para el cuál fue necesaria conocimientos de electrónica digital y analógica básica, y la investigación de como se generan los valores pseudo-aleatorios es los dispositivos incrustados. La dinámica del circuito corresponde a un botón de acción rompiendo la circulación de corriente, con el fin de enviar una indicación al microcontrolador de realizar el lanzamiento del dado y desplegando el valor por medio de sus salidas y compuertas lógicas para la manipulación de los LED.

2. Nota Teórica

2.1. Microcontrolador PIC

Los microcontroladores fundamentalmente son dispositivos concebidos fundamentalmente para realizar aplicaciones puntuales, debido a que los componentes que lo integran tienden a ser limitados. Parte de la caracterísiticas principales de un microcontrolador son las siguientes [1]:

- Capacidad de control de entrada y/o salida: El manejo individual y especifico de las líneas de entrada y salida permiten la versatilidad de programación basada en objetivos específicos.

- Registros de Estados: Es una unidad que almacena resultados de operaciones aritméticas y sus signos.
- Contador de programa: Este contador es un registro que almacena las direcciones de las instrucciones del CPU, con el fin de poder apuntar siempre a la instrucción que sigue para agilizar el proceso.
- Registro de Direccionamiento de Datos: Este registro almacena la dirección de los resultados guardados en memoria, con el fin de no realizar la búsqueda completa en toda la memoria y incrementar la velocidad de acceso.

Ahora, bien los microcontroladores PIC siguen una arquitectura desarrollada por la Universidad de Harvard, donde la memoria de instrucciones de programa y de datos están separadas. Además, la memoria de instrucciones es bastante más grande que la de datos, la primera está organizada por longitudes de bit desde 12 hasta los 16, mientras que la de datos únicamente se encuentra organizada por longitudes de 8 bits.
El microcontrolador PIC12F683 tiene un procesador basado en arquitectura RISC. Cuenta con seis GPIO descritos en el diagrama 1, mientras que sus dos pines restantes son las entradas de voltaje y tierra (V_{SS} y V_{DD} respectivamente). Su unidad de reloj trabaja a 32 kHz de frecuencia a 2V. La configuración de los GPIOS se realiza mediante la manipulación de registros: TRISIO, GPIO,ANSEL, etc.

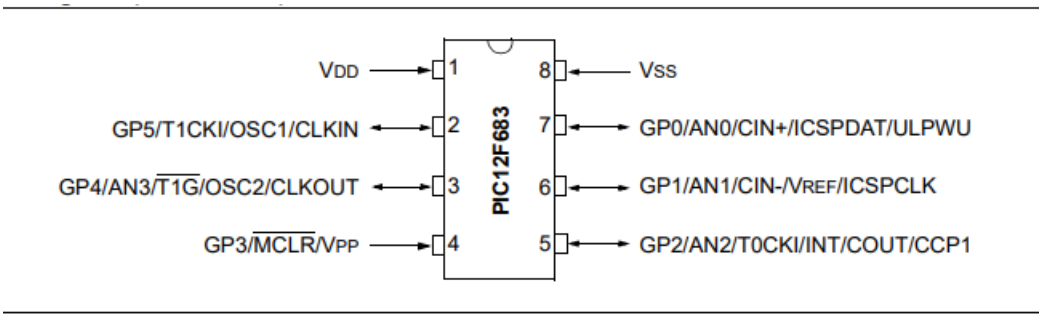

Figura 1: Diagrama de pines PICF6835. Tomado de [2]

Los registros TRISIO manipulan el modo de operación, si es entrada el registro se designará de manera lógica como un alto, es decir un uno, y por otro lado de salida se me como un bajo o un cero. Los registros GPIO son los que almacenan los datos habientes en los pines, estos corresponde a ocho bits. Los registros ANSEL determinan el uso del ADC para una selección analógica o digital.

3. Desarrollo/Análisis de Resultados

3.1. Construcción del circuito

En primer lugar, se construyó el circuito siguiendo las especificaciones eléctricas según la hoja del fabricante. Entre las especificaciones más importantes tenemos las siguientes:

Absolute Maximum Ratings(†)

Ambient temperature under bias	-40° to +125°C
Storage temperature	-65°C to +150°C
Voltage on V_{DD} with respect to V_{SS}	-0.3V to +6.5V
Voltage on $\overline{MCLR}$ with respect to Vss	-0.3V to +13.5V
Voltage on all other pins with respect to V_{SS}	-0.3V to (V_{DD} + 0.3V)
Total power dissipation(1)	800 mW
Maximum current out of V_{SS} pin	95 mA
Maximum current into V_{DD} pin	95 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD})	± 20 mA
Output clamp current, I_{OK} (V_o < 0 or V_o > V_{DD})	± 20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by GPIO	90 mA
Maximum current sourced GPIO	90 mA

Note 1: Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OI} \times I_{OL})$.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

Figura 2: Especificaciones eléctricas de los GPIO

Con el fin de proteger los GPIO se debe seguir la regla de la corriente máxima, por lo tanto para obtener una corriente de 20 mA se utiliza la ley de ohm para calcular las resistencia necesaria.

$$R = \frac{5V}{20mA} = 250\Omega$$

Ahora, según las instrucciones se necesita un interruptor capaz de accionar el lanzamiento del dado, para esto se procedió a conectar un interruptor que realizará la conexión directa a tierra con el fin de cortar el flujo de corriente en la entrada GP5 del microcontrolador que se configuró como una entrada.

Como salidas del PICF683 se utilizaron los pines GP0, GP1 y GP2, conectadas a un demultiplexor con el fin de realizar de introducir el numero resultante del dado de manera binaria con los tres GPIO y conseguir la salida de manera decimal. Uno de los inconveniente a la hora de hacerlo de esta forma es que un demultiplexor únicamente da salida a la entrada correspondiente la demultiplexación y la idea del dado es encender la cantidad de LED correspondiente al valor aleatorio obtenido. Por lo tanto, se utilizaron cinco compuertas OR para realizar la lógica si la salida siguiente se encuentra en alto o la salida actual para encender el LED correspondiente.

Finalmente, se utilizó una barra LED conectada las compuertas protegidas por una resistencia de 100Ω. A continuación se ve el diagrama completo construido para el circuito:

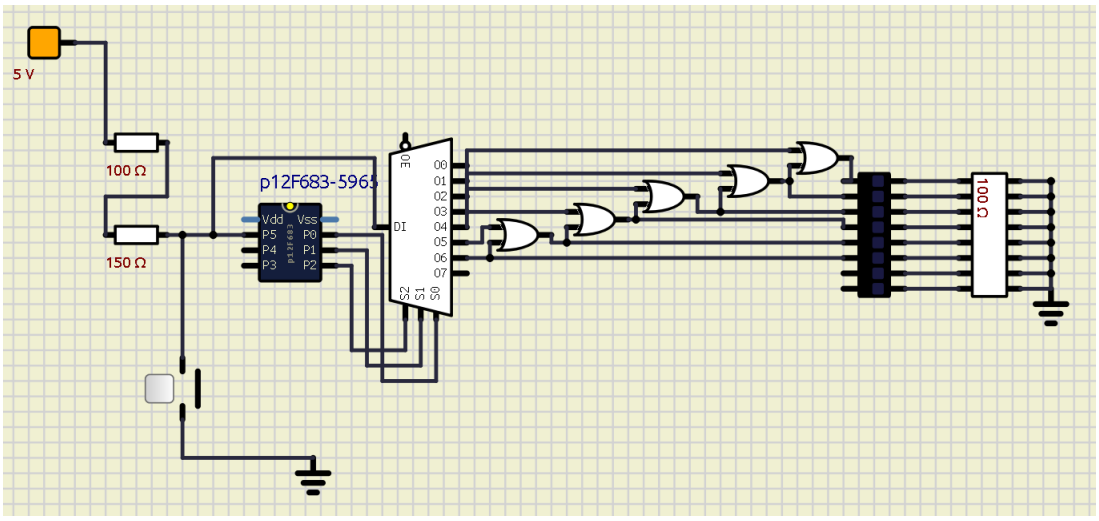

Figura 3: Diagrama dado eléctrico

El precio aproximado de la construcción del circuito es la siguiente:

Componente	Precio (¢)
PIC12F683	308
8 Resistencias	360
6 LED	1200
2 Integrados de 4 OR	1040
Total	2908

Cuadro 1: Tabla de valor de componentes. Referencia tomada link

3.2. Software Utilizado

Para programar la lógica interna de los microcontrolador se utilizaron funciones explicadas a continuación:

```
    #include <pic14/pic12f683.h>
    typedef unsigned int word;
    word __at 0x2007 __CONFIG = (_WDT_OFF & _WDTE_OFF);

```

Inicialmente, se incluyó las librerías del microcontrolador y se procedió a desactivar el *WatchdogTimer*, el cuál es un algoritmo contador de oscilaciones de un reloj ya sea independiente o el principal del microcontrolador, si el contador se desborda el RESET del dispositivo se activa no permitiendo mostrar la salida deseada [1]. Por lo cuál, se buscó el la documentación de las librerías para encontrar el macro que permite deshabilitar esta función.

```
int gen_random(){
  int p = 11;
  int q = 19;
  int M = p*q;

  xo = (xo*xo) % M;

  return xo;
```

```
}

```

Como lo que se quiere simular es el comportamiento de un dado, se necesita un generador de números pseudo aleatorios. En este caso se utilizó el algoritmo de Blumb Blumb Shub para generar esto valor.

```
void lanzar_dado(unsigned int numero){
  switch (numero)
  {
  case 1:
    GPIO = 0x01;
    break;
  case 2:
    GPIO = 0x02;
    break;
  case 3:
    GPIO = 0x03;
    break;
  case 4:
    GPIO = 0x04;
    break;
  case 5:
    GPIO = 0x05;
    break;
  case 6:
    GPIO = 0x06;
    break;

  default:
    GPIO = 0x00;
    break;
  }
}

```

Ahora, la función de lanzar es bastante simple se utiliza una estructura switch para codificar el numero resultante de la función de generación de números aleatorios y los codifica de manera binaria para enviarlos por medio de los GPIO al demultiplexor.

```
void delay(unsigned int tiempo)
{
  unsigned int i;
  unsigned int j;

  for(i=0;i<tiempo;i++)
    for(j=0;j<1275;j++);
}

```

Una simple función de retraso con el fin de poder darle suficiente tiempo al microcontrolador de desplegar el dato.

```
void main(void)

{
    TRISIO = 0b00100000; //Poner todos los pines como salidas
  GPIO = 0x00;

    unsigned int time = 100;
```

```
8
9       //Loop forever
10      while ( 1 )
11      {
12      if(GP5 == 0){
13        unsigned int rand = gen_random();
14        if(rand > 6){
15          rand = (rand % 6) + 1;
16          lanzar_dado(rand);
17        } else {
18          lanzar_dado(rand);
19        }
20        delay(100);
21      }}
22  }
23
```

Finalmente, la función principal en primera instancia indica por medio de la sobreescritura a los registros TRISIO que el GPIO cinco funcionará como entrada y los demás como salida de datos. Por otra parte, también con el uso de los registros destinados a guardas los datos de GPIO se ponen todos incialmente en cero, con el fin de no añadir ruido. El lazo principal se activará cada vez que se rompa el flujo de corriente por el GPIO 5 donde llamará a la función de generar random y determina o no si este número es mayor al número de caras del dado, si lo es realiza la conversión mediante el uso de el resto de la división. Finalmente, envía el valor generado a la función generar dato y añade un retraso para asegurar el despliegue correcto del valor en los LED.
Para visualizar la funcionabilidad completa del circuito puede dirigirse al siguiente link.

4. Observaciones y Recomendaciones

- Es importante realizar un diseño antes de construir una implementación como esta, debido a que brinda una idea más clara de la relación de la entrada y salida y que complementos pueden llegar a surgir.
- El estudio de la hoja del fabricante es de vital importancia, dado que nos da las referencias de parámetros adecuados de manipulación de las entradas y salidas del microcontrolador, así como características importantes como funcionamiento de los registros.
- El uso de simuladores para realizar el aprendizaje y primeros diseños da una perspectiva real de como se construye un sistema, sin poner el riesgo el equipo.
- Para acceder al repositorio del laboratorio puede utilizar el siguiente enlace a oscarfc164/IE0624

Referencias

[1] F. Valdés and R. P. Areny, *Microcontroladores fundamentos y aplicaciones con PIC*, vol. 1149. Marcombo, 2007.

[2] Microchip, *PIC12F683 Datasheet*. 2007.