

## 1 Introducción:

Para esta práctica de laboratorio se hizo una especie de dado electrónico el cual por medio de un microcontrolador PIC12f675 genera una especie de números pseudo aleatorios, y cuando un usuario externo toca un botón de entrada, se simula que se lanza el dado y el microcontrolador da una salida, enciende los leds según corresponda. Este micro controlador es sumamente básico y con poca memoria, por lo que no es posible utilizar algoritmos muy complejos para generar los números aleatorios, por lo que se tomó otra alternativa que se discute más adelante. Además con el fin de eliminar el factor del “bouncing” del botón de entrada se utiliza un filtro pasivo hecho con una resistencia y un capacitor.

## 2 Nota Teórica:

### 2.1 PIC12f675

El PIC12f675 es un microcontrolador el cual cuenta con un procesador de tipo RISC (Reduced instruction set), además cuenta con un reloj interno de hasta 4MHz. La característica que diferencia a este microcontrolador de los demás, es su poco consumo de corriente, el cual anda alrededor de  $100 \mu A$  según la hoja de datos. También cuenta con un conversor analógico/digital, y un comparador con schmitt trigger.

En cuanto a los periféricos, este microcontrolador cuenta con 6 pines en entrada/salida completamente programables (GPIOs). Se debe tener cuidado, ya que como se muestra en la siguiente figura, el GP3/MCLR, se puede utilizar únicamente como entrada, y además funciona como un clear activo en estado bajo.

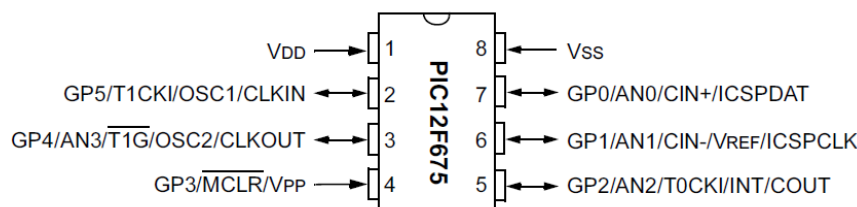


Abbildung 1: Esquema general del PIC12F675

#### 2.1.1 Organización de memoria en el PIC12F675

Este microcontrolador cuenta con un contador de programa (PC) de 13 bits. La memoria del PIC12F675 está partida en dos bancos, los cuales una partición contiene los registros de propósito general y la otra contiene los registros especiales. (Los registros especiales son utilizados por el CPU y los periféricos para controlar la correcta funcionalidad del equipo)

#### 2.1.2 Registro TRISIO

Con este registro se le dice al microcontrolador cual va a ser la configuración de entrada/salida. Es decir mediante este registro se le puede decir al microcontrolador cuales pines van a ser entradas y cuales van a ser salidas.

TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111
--------	---	---	---------	---------	---------	---------	---------	---------	-----------

Abbildung 2: Configuración general del registro TRISIO

Como se muestra en la figura anterior, es un registro de 8bits, sin embargo solo 6 son utilizables por el programador para designar entradas y salidas. Un ejemplo de esto se muestra en seguida:

---

```

void main(){
    TRISIO = 0b00100000; // Se designa GP5 como entrada y el resto como salidas
}

//Otra forma de hacer lo mismo para evitar bugs

void main(){
    TRISIO0 = 0; //Se designa GP0 como salida
    TRISIO1 = 0; //Se designa GP1 como salida
    TRISIO2 = 0; //Se designa GP2 como salida
    TRISIO3 = 0; //Se designa GP3 como salida
    TRISIO4 = 0; //Se designa GP4 como salida
    TRISIO5 = 1; //Se designa GP5 como entrada
}

```

---

### 2.1.3 Registro GPIO

Este registro sirve para decirle al microcontrolador el estado por default de los pines, es decir si es alto o bajo.

GPIO	—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
------	---	---	-------	-------	-------	-------	-------	-------

Abbildung 3: Configuración general del registro GPIO

Al igual que con el registro TRISIO, el GPIO cuenta con 8 bits; sin embargo, solo 6 de esos bits son utilizables por el programador para designar el estado por defecto de los pines de entrada/salida.

---

```

void main(){
    GPIO = 0; // Se designan todos los pines con un cero a la salida.
}

//Otra forma de hacer lo mismo para evitar bugs

void main(){
    GP0 = 0; //Se designa GP0 como un cero por defecto.
    GP1 = 0; //Se designa GP0 como un cero por defecto.
    GP2 = 0; //Se designa GP0 como un cero por defecto.
    GP3 = 0; //Se designa GP0 como un cero por defecto.
    GP4 = 0; //Se designa GP0 como un cero por defecto.
    GP5 = 0; //Se designa GP0 como un cero por defecto.
}

```

---

Esto es especialmente útil, ya que se sabe que la corriente viaja de mayor potencial a menor potencial, y por ende se puede pensar en alguna configuración en que la corriente vaya de los pines hasta alguna tierra, pero esto ya depende del diseño físico.

### 2.1.4 Registro ANSEL

Este registro controla el conversor analógico/digital; sin embargo, para esta implementación, no se utilizó y por ende se dejó apagado.

ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0
-------	---	-------	-------	-------	------	------	------	------

Abbildung 4: Configuración general del registro ANSEL

### 2.1.5 Registro CMCON

Este registro controla el comparador dentro del microcontrolador; pero al igual que con el registro ANSEL, el comparador no fue necesario para la implementación de dado y por ende se dejó apagado.

CMCON	—	COUT	—	CINV	CIS	CM2	CM1	CM0
-------	---	------	---	------	-----	-----	-----	-----

Abbildung 5: Configuración general del registro CMCON

## 2.2 Características eléctricas del PIC12F675

Tomando la hoja de datos del microcontrolador, se obtienen las características eléctricas físicas del mismo. Esto es importante ya que permiten dimensionar resistencias, y componentes periféricos. Todo lo anterior con el fin de proteger los componentes y asegurar el buen funcionamiento del diseño.

Como se ve en la siguiente figura, cada pin del PIC12F675 es capaz de entregar 25mA, Con el fin de proteger tanto los LEDs del diseño, y el microcontrolador, se utilizan resistencias de protección, las cuales se dimensionan en la sección siguiente. Además la corriente total que pueden entregar en total los pines juntos es de 125mA según la hoja de datos, por lo que se debe cuidar que la suma de las corrientes cuando varios pines estén dando una salida no sobrepase los 125mA.

#### 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 $V_{SS}$ .....	-0.3 to +6.5V
Voltage on $\overline{MCLR}$ with respect to $V_{SS}$ .....	-0.3 to +13.5V
Voltage on all other pins with respect to $V_{SS}$ .....	-0.3V to ( $V_{DD} + 0.3V$ )
Total power dissipation <sup>(1)</sup> .....	800 mW
Maximum current out of $V_{SS}$ pin .....	300 mA
Maximum current into $V_{DD}$ pin .....	250 mA
Input clamp current, $I_{IK}$ ( $V_I < 0$ or $V_I > V_{DD}$ ).....	$\pm 20$ mA
Output clamp current, $I_{OK}$ ( $V_O < 0$ or $V_O > V_{DD}$ ).....	$\pm 20$ 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 all GPIO .....	125 mA
Maximum current sourced all GPIO .....	125 mA

Abbildung 6: Características eléctricas del microcontrolador en cuestión

## 2.3 Dimensionamiento de las resistencias de protección de los LED

Para todo ingeniero eléctrico, técnico en electrónica o entusiasta, es bien sabido que los LED son elementos eléctricos sumamente frágiles y tienden a quemarse por sobre corrientes muy fácilmente. Es por esto que siempre que se tiene un led, el mismo debe ir acompañado de una resistencia en serie. El valor de esta resistencia va a estar dado por el resto de componentes del circuito.

Como en este caso se conoce que la corriente máxima que da el microcontrolador es de 25mA, se puede usar este dato para el dimensionamiento de la resistencia. Sin embargo por temas de protección del mismo

microcontrolador, nunca se debe apuntar a utilizar el máximo, sino un poco menos, por lo que para este diseño se procurará estar por debajo de los 20mA por pin en todo momento.

Si se tiene que un uno lógico es aproximadamente 5V, y que el máximo de corriente que se desea es de 20mA, se puede utilizar ley de Ohm para obtener el valor del diseño para la resistencia.

$$V = IR \rightarrow R = \frac{5}{20mA} \approx 250\Omega \quad (1)$$

Como eventualmente este diseño se quiere hacer de forma física, el valor de resistencia de 250Ω, es un poco idealista, y un tanto complicado de conseguir. Por lo que para esto se utiliza una tabla de componentes disponibles en la bodega de la escuela de ingeniería eléctrica y así diseñar una resistencia aproximadamente igual a la del diseño. Según la foto que se muestra en seguida, una resistencia de 250Ω, se puede obtener conectando una resistencia de 220Ω en serie con una de 27Ω. Ya si se quiere más precisión, se pueden conectar resistencias de 1Ω para llegar a los 250Ω, pero con una resistencia de 247Ω, debería ser más que suficiente para proteger todo el sistema.

1 Ω	4.7 Ω	8.2 Ω	10 Ω	12 Ω
15 Ω	18 Ω	22 Ω	27 Ω	33 Ω
39 Ω	47 Ω	56 Ω	68 Ω	82 Ω
100 Ω	120 Ω	150 Ω	180 Ω	220 Ω
270 Ω	330 Ω	390 Ω	470 Ω	510 Ω
560 Ω	680 Ω	820 Ω	1 KΩ	1.2 KΩ
1.5 KΩ	1.8 KΩ	2.2 KΩ	2.7 KΩ	3.3 KΩ
3.9 KΩ	4.7 KΩ	5.6 KΩ	6.8 KΩ	8.2 KΩ
10 KΩ	12 KΩ	15 KΩ	18 KΩ	22 KΩ
27 KΩ	33 KΩ	39 KΩ	47 KΩ	56 KΩ
68 KΩ	82 KΩ	100 KΩ	110 KΩ	120 KΩ
150 KΩ	180 KΩ	220 KΩ	270 KΩ	330 KΩ
390 KΩ	470 KΩ	560 KΩ	680 KΩ	820 KΩ
910 KΩ	1 MΩ	1.2 MΩ	1.5 MΩ	1.8 MΩ
2.2 MΩ	2.7 MΩ	3.3 MΩ	3.9 MΩ	4.7 MΩ
5.6 MΩ	6.8 MΩ	8.2 MΩ	10 MΩ	12 MΩ
15 MΩ	18 MΩ	22 MΩ	27 MΩ	

Abbildung 7: Resistencias Disponibles en bodega

## 2.4 Tratando el rebote del botón de entrada

Cuando se enciende o apaga un interruptor, ya sea un switch, o un botón, la salida del mismo no es totalmente limpia, sino que presenta rebotes debido que los materiales físicos rebotan entre ellos. Esto es un problema ya que los rebotes pueden inevitablemente introducir falsas lecturas al sistema.

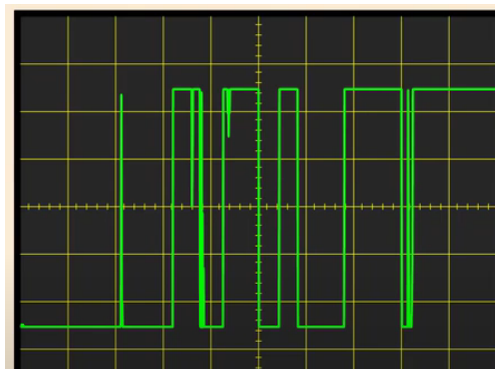


Abbildung 8: Efecto de rebote de un botón

Existen diversas formas para contrarrestar este efecto, ya sea por hardware o por software. Para este diseño específico se decidió ir por la solución implementada en hardware, ya que es fácil, eficiente y barata de implementar.

Para esto se aplica un filtro con el fin de eliminar los rebotes del switch.

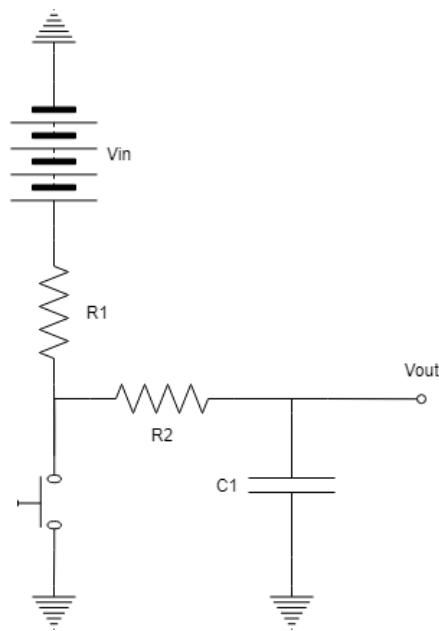


Abbildung 9: Efecto de rebote de un botón

Como no se tiene una entrada sinusoidal, y además no se pretende filtrar ciertas frecuencias, no hay parámetros importantes para dimensionar el capacitor, con solo que se cargue y se descargue en un tiempo prudente es suficiente. Por esto mismo se escoge de los capacitores en bodega, uno de 100pF, lo cual se va a cargar sumamente rápido y lo mismo para la descarga, cumpliendo así el objetivo de filtrar la señal del switch. Teniendo esto se pueden dimensionar las resistencias teniendo en cuenta el tema del tiempo.

$$\tau = RC \rightarrow 200\Omega \cdot 100pF \approx 20nS \quad (2)$$

Este valor es lo suficientemente pequeño como para que sea imperceptible por el usuario, y parezca que se está tirando el dado en tiempo real.

## 2.5 Diseño Final del dado electrónico

Tomando en consideración todas las secciones anteriores, se puede crear un diseño funcional para lo solicitado en el enunciado del laboratorio. Tomando en cuenta que se quiere simular un dado, es necesario utilizar 7 LEDs, y 4 resistencias para los mismos. Esto debido a que los extremos del dado tendrán dos leds. Es decir cada la esquina superior izquierda está conectada a la esquina inferior derecha. La esquina superior derecha está conectada a la esquina inferior izquierda y por último los laterales están conectados entre ellos como se muestra en la siguiente figura.

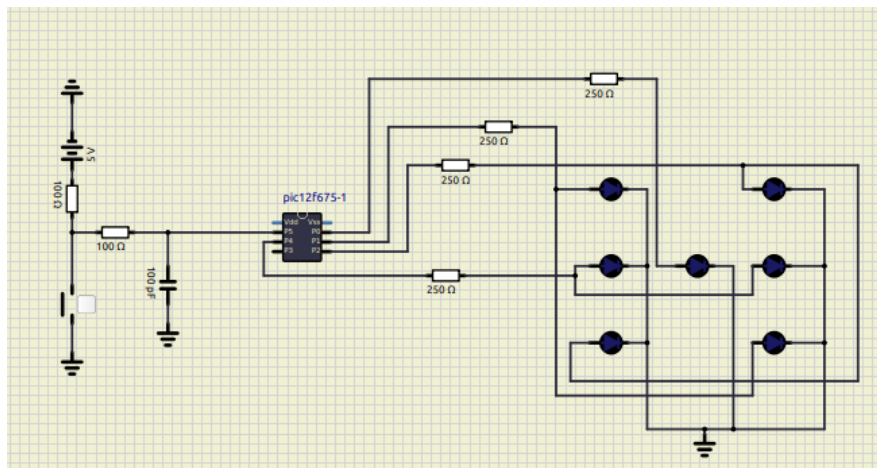


Abbildung 10: Esquemático del circuito montado. (Creación Propia)

Los componentes del filtro de entrada son los mismos discutidos en la sección anterior, dos resistencias, un botón, la batería y el capacitor.

Buscando vendedores de los componentes anteriormente mencionados, se tiene el problema de que no venden la cantidad exacta de componentes; sin embargo esto no presenta un problema debido a que un paquete con 100 resistencias cuesta alrededor de 6 dólares, lo que sí sube considerablemente el precio del proyecto, pero sigue siendo un proyecto que sale por aproximadamente 10 mil colones. (Las capturas de los precios se presentan en la sección de anexos)

Componente	Cantidad	Precio
PIC12F675	1	\$ 3.5
100 $\Omega$ Resistor	2	\$1.20
220 $\Omega$ Resistor	4	\$ 0.6
27 $\Omega$ Resistor	4	\$ 6.5
LED	7	\$ 5.7
<b>TOTAL</b>	<b>\$ 17.5</b>	

Tabelle 1: Tabla resumen de la cantidad de componentes y sus precios.

### 3 Análisis de resultados:

### 4 Conclusiones y recomendaciones