

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624: LABORATORIO DE MICROCONTROLADORES

Laboratorio 1

Introducción a microcontroladores y manejo de GPIOs

Prof. Marco Villalta

Estudiantes:
Sofía Fonseca Muñoz, B42634

13 de enero de 2023

Índice

1. Introducción/Resumen	2
2. Nota teórica	3
2.1. Microcontrolador	3
2.1.1. Características eléctricas	4
2.1.2. Registros utilizados	4
2.2. Bibliotecas	5
2.3. Diseño del circuito	5
2.4. Componentes electrónicos	7
3. Desarrollo/Análisis de resultados	8
4. Conclusiones y recomendaciones	12
Referencias	13
5. Apéndice	14

1. Introducción/Resumen

Este laboratorio consiste en la creación de un dado digital con un microcontrolador PIC12F683. Para su creación se utilizaron siete LEDs para simular los lados y un botón que activa el dado.

Como parte de este laboratorio fue necesario aprender la programación de las entradas y salidas del microcontrolador. Además la creación de un generador de números aleatorios para que no tire los dados en orden.

La implementación de este proyecto se realizó por medio de Simulide. En el repositorio correspondiente a este laboratorio se puede encontrar tanto el código como la simulación, en el enlace https://github.com/sofifon/IE0624-2023/tree/main/Lab1_Die.

2. Nota teórica

2.1. Microcontrolador

El microcontrolador utilizado en este laboratorio corresponde a un PIC12f675. Este es un microcontrolador que permite ejecutar instrucciones rápidamente y es fácil de programar. Gracias a esto, este microcontrolador se ha adaptado a la industria automotriz, electrodomésticos y productos iniciales que requieren diferentes programaciones. Cuenta con 1.75 KB de memoria Flash y 128B de EEPROM. Además cuenta con 6 pines de propósito general de entrada y salida. Este microcontrolador opera entre 2 y 5.5 V.[1]

Este microcontrolador cuenta con 13 bits en el contador de programa. La memoria se divide en dos, por un lado se cuenta con los registros de propósito general mientras el otro contiene los registros especiales que son utilizados por el CPU para mantener un funcionamiento correcto de los periféricos.

En el siguiente diagrama de pines se encuentra las especificaciones de cada uno de ellos y las funcionalidades que pueden cumplir, por ejemplo los pines 3, 5, 6 y 7 pueden recibir información analógica porque cuentan con el canal de conversión analógico a digital. Los pines 7 y 6 son comparadores de entrada mientras que el 5 comparador de salida. El pin 3 puede dar el pulso de reloj interno del microcontrolador. [2]

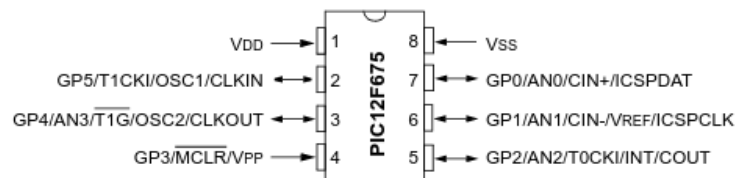


Figura 1: Diagrama de pines del procesador PIC12f675. [2]

En el diagrama de bloques de este microprocesador, se pueden observar otras características de este microprocesador, como que posee un oscilador interno de 4 MHz, la RAM de 64 x 8, el temporizador interno, entre otros. Los pines de propósito genral son bidireccionales, cuenta con un convertidor de analógico a digital de la entrada de datos. Además cuenta con la ALU que interpreta los registros para su funcionamiento programado. [2]

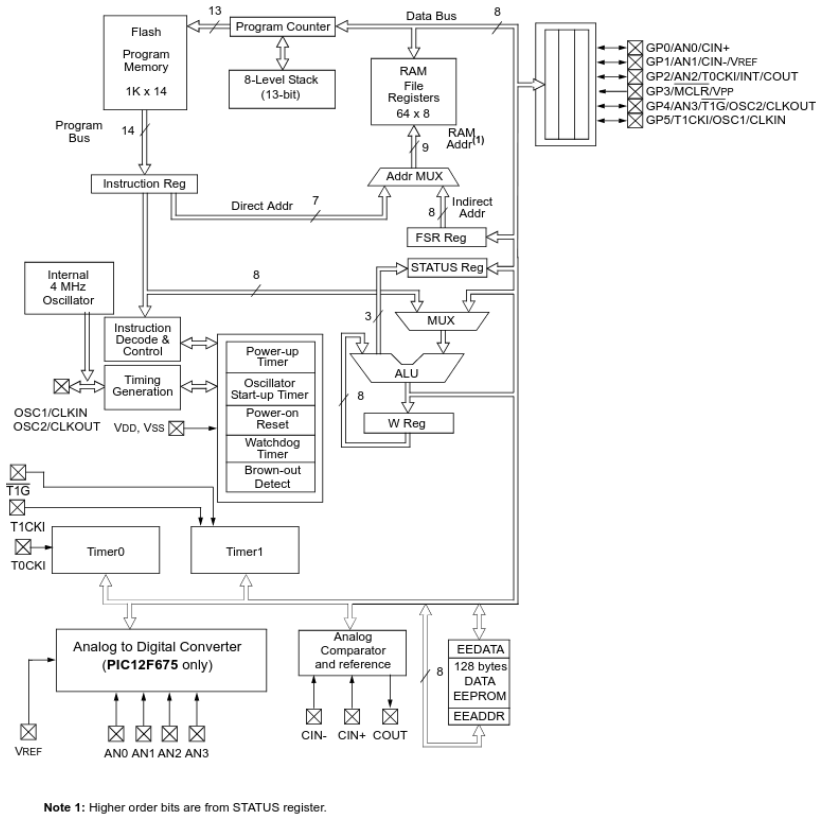


Figura 2: Diagrama de bloques del procesador PIC12f675. [2]

2.1.1. Características eléctricas

En la siguiente tabla se pueden encontrar las características físicas de este microcontrolador, las cuales se deben tomar en cuenta para la implemetación en la que se desea utilizar el mococontrolador.[2]

Especificaciones eléctricas	
Temperatura ambiente	-40 to +125°C
Temperatura de almacenamiento	-65°C to +150°C
Voltaje entre Vdd y Vss	-0.3 to +6.5V
Voltaje entre los otros pines y Vss	-0.3V a Vdd
Disipación de potencia total	800 mW
Máxima corriente de salida por cada pin	25 mA
Máxina corriente de entrada por cada pin	25 mA
Máxima corriente de salida total	125 mA
Máxima corriente de entrada total	125 mA

2.1.2. Registros utilizados

Para la implementación de este laboratorio, se utilizaron dos registros correspondientes a las funcionalidades de los pines utilizados. Estos corresponden a GPIO y TRSIO y se encuentran configurados como las imágenes a continuación. Estos registros responden al puerto bidireccional GPIO

de 6 bits, por lo que se puede ver como los dos bits más significativos no están implementados y se leen como 0.

El registro TRISIO es utilizado para determinar si el pin es una entrada, en alto o una salida en bajo, de manera que activa el modo de alta impedancia o lleva la información al pin correspondientemente. El registro GPIO permite asignar un valor a cada uno de los pines, si no se especifica el pin, se asigna el valor a todos los pines, pero se puede agregar el número para modificar el valor de uno solo de los pines.

REGISTER 3-2: TRISIO: GPIO TRI-STATE REGISTER (ADDRESS: 85h)							
U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0
bit 7		bit 0					

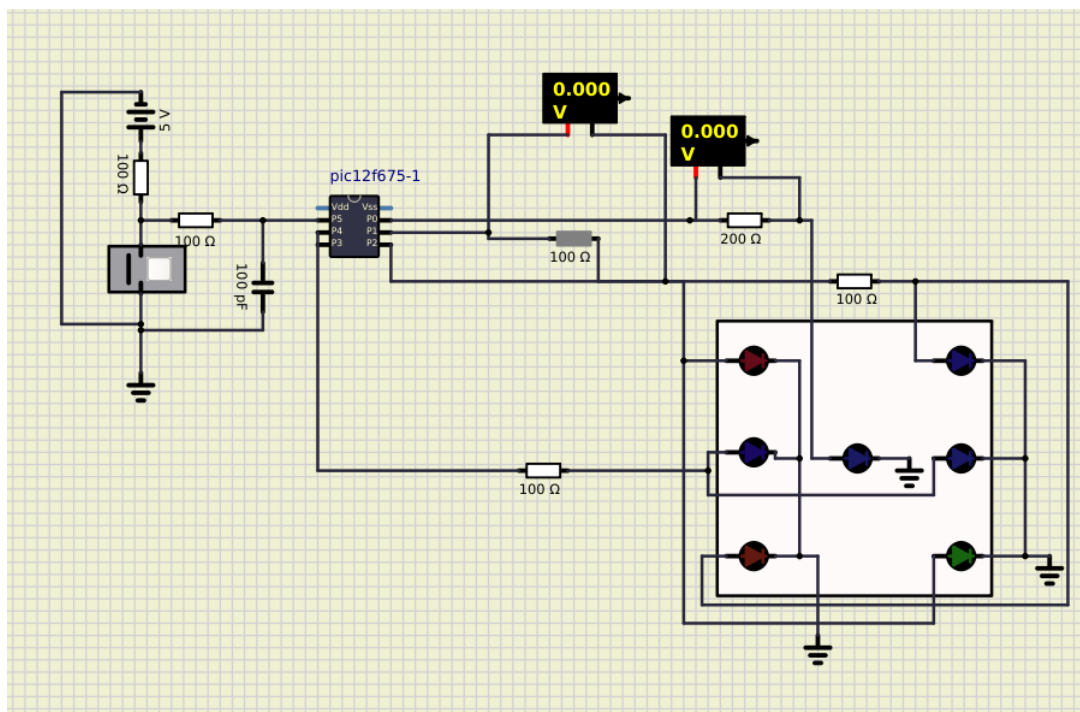


Figura 5: Diseño completo del circuito del dado. Imagen propia.

Para este laboratorio, existía la necesidad de colocar un botón como activador del dado y los LEDs que simulaban los lados del dado. Por lo tanto, se necesita un GPIO para el botón y como el microcontrolador cuenta con 5, no alcanza cada uno para programar un LED del dado. Por lo que se escogió una configuración de 7 LEDs conectados estratégicamente de manera que el LED del centro esté conectado a un puerto y los LEDs de los lados compartan una conexión de dos en dos. De manera que para que se puedan poner los números del 1 al 6 se puede escoger que al menos uno de los pines esté activado, combinando la cantidad hasta sumar el número deseado.

Cabe destacar que este laboratorio se pudo resolver con 6 LEDs pero para alcanzar un representación estética del dado se utilizaron 7 para que cuando se enciendan los diferentes números realmente parezca cada uno de los lados de un dado real. Además el costo de agregar un LED demás es muy bajo. Junto con esto, los LEDs que comparten conexión están colocados en extremos contrarios del dado para alcanzar los efectos deseados.

Junto con los elementos anteriormente mencionados, se agregaron resistencias que evitan que tanto los LEDs como el microcontrolador se encuentren en un rango de corriente que no sobre pase sus límites. Dentro de las características del microcontrolador anteriormente mencionadas es que la corriente máxima de entrada como de salida para cada uno de los pines es de 25 mA.

En el caso del botón se utiliza una fuente de 5V, por lo que se debe encontrar un valor de resistencia para que la corriente no sobrepase los 25mA. Utilizando la ecuación a continuación se puede encontrar que se necesita una resistencia de $200\ \Omega$, en este caso se escoge dividirla en dos para que en el momento en que el botón se presione existe un paso de corriente hacia el microcontrolador. El capacitor que se agrega para que la corriente inicial no sobrepase los límites y pueda causar problemas con el microprocesador.

$$V = IR \Rightarrow R = \frac{V}{I} = \frac{5V}{25mA} = 200 \, \Omega \quad (1)$$

Por lo que la conexión del botón queda así:

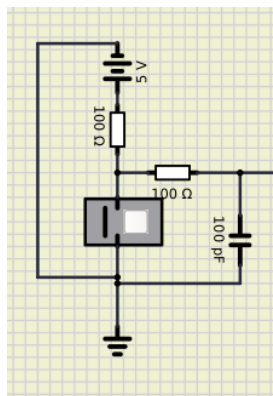


Figura 6: Conexión del botón. Imagen propia.

En el caso de las resistencias, estas también no soportan una corriente mayor a los 25 mA, por lo que utilizando de nuevo la ecuación 1, se puede determinar que se necesitan resistencias de $200\ \Omega$. En el caso de las conexiones que tienen dos LEDs, la corriente a cada uno va a ser la mitad porque la corriente se divide en dos en paralelo para cargas iguales. Por lo que se escoge una resistencia de $100\ \Omega$ para la carga que es el doble.

2.4. Componentes electrónicos

Para la creación del semáforo, se utilizó la lista de componentes que se encuentra en la siguiente tabla. Los precios de cada componente fueron tomados de <https://www.crcibernetica.com/> y de <https://www.microchip.com/> tomando en cuenta un tipo de cambio de 595 colones equivalente a 1 USD. Cabe destacar que Microchip no es una empresa que trabaja en Costa Rica por lo que el costo se ve afectado por costos de envío. Además los precios de las resistencias y los LEDs se calcularon a partir de paquetes grandes dividiendo su valor entre la cantidad, ya que por su valor no se consiguen individualmente.

Componente	Precio (colones)
PIC12F675	780
Mini Push Button Switch	210
50 x Resistor 100 ohm	150
1 x Resistor 200 ohm	75
7 x LED rojo	140
Fuente 5 V	4518
Precio total	5873

El precio total de todos los componentes necesarios para construir el dado sería 5873 colones.

3. Desarrollo/Análisis de resultados

Para el desarrollo de este proyecto, se siguió la siguiente línea de pensamiento con respecto a su funcionamiento:

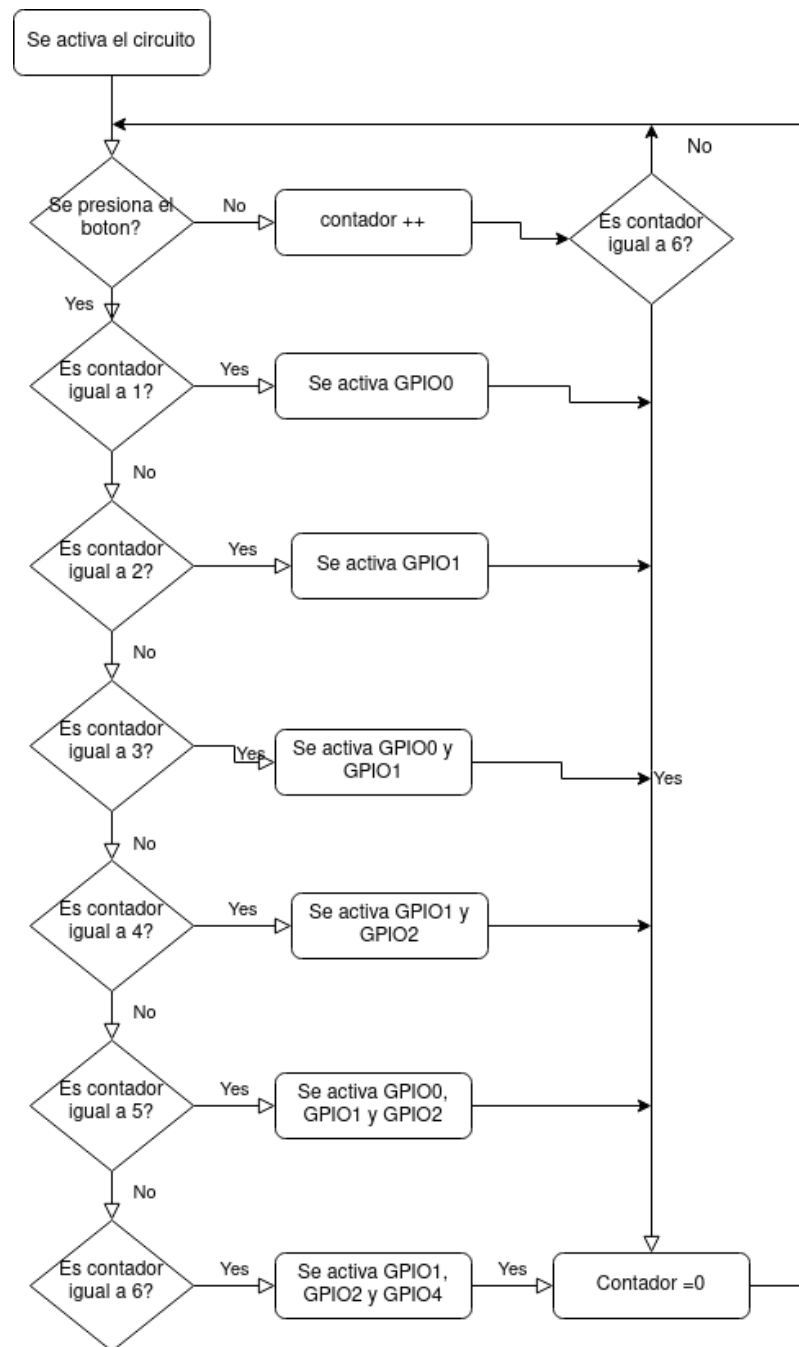


Figura 7: Diagrama de bloques del algoritmo utilizado para el dado. Creación propia.

Para hacer una elección de un número aleatorio, se escogió implementar un contador en la máquina que va de 0 a 6, como la frecuencia de ejecución es muy alta en el microcontrolador, los números van pasando muy rápido por lo que es muy difícil para un ser humano obtener el mismo número seguido al presionar el botón. Una de las decisiones que se tomó al hacer el código fue devolver el

contador a cero cada vez que se tocaba el botón para que cambiara la frecuencia de cada número en ese momento.

Otra de las preocupaciones es que se eligiera cuando el número estaba en cero y que no hubiera una opción. Sin embargo, se implementó de manera que el contador vuelve a cero al final mientras que se suma un nuevo valor al contador al principio del ciclo, por lo que nunca va a ser cero en el momento de hacer la elección.

Las siguientes imágenes muestran el resultado donde se puede ver los valores de 1 a 6 en el dado. Y que al presionarlo aleatoriamente se puede obtener todas las opciones.

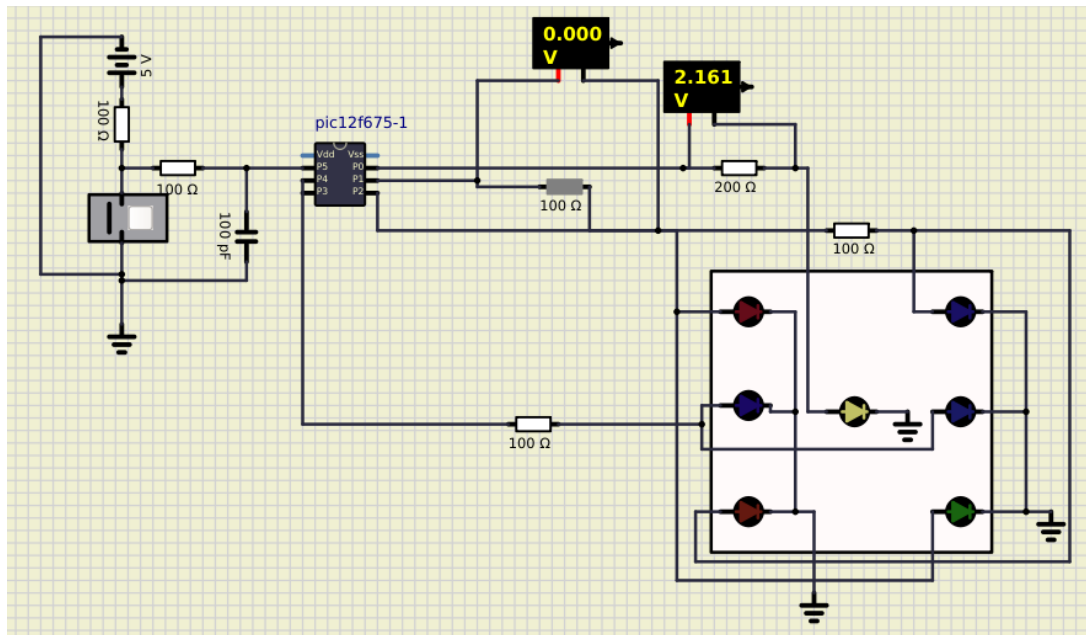


Figura 8: Demo del dado mostrando el lado con valor 1.

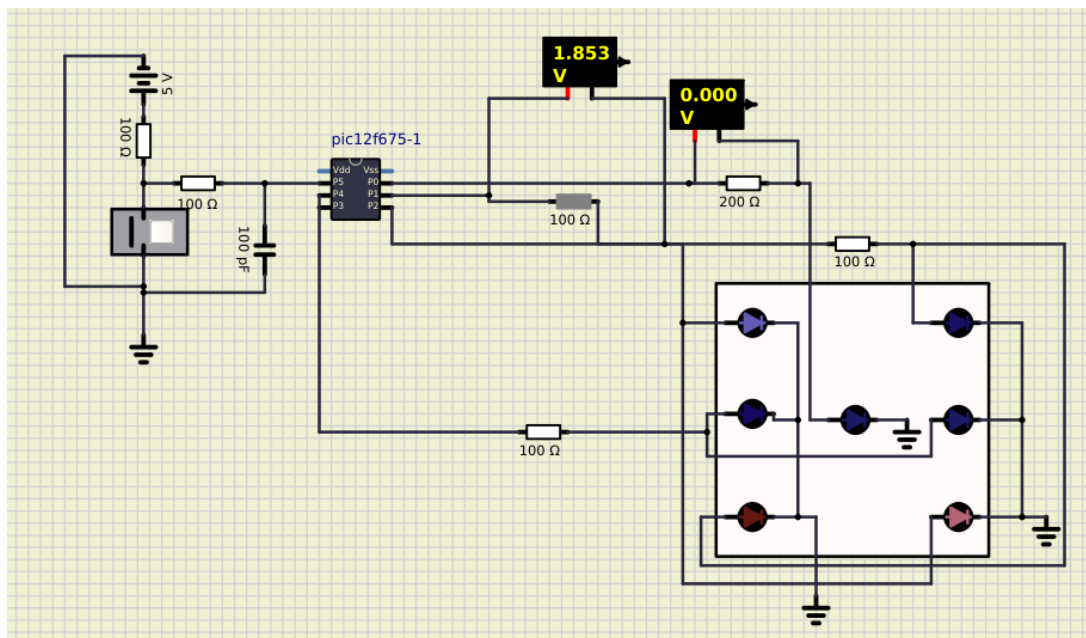


Figura 9: Demo del dado mostrando el lado con valor 2.

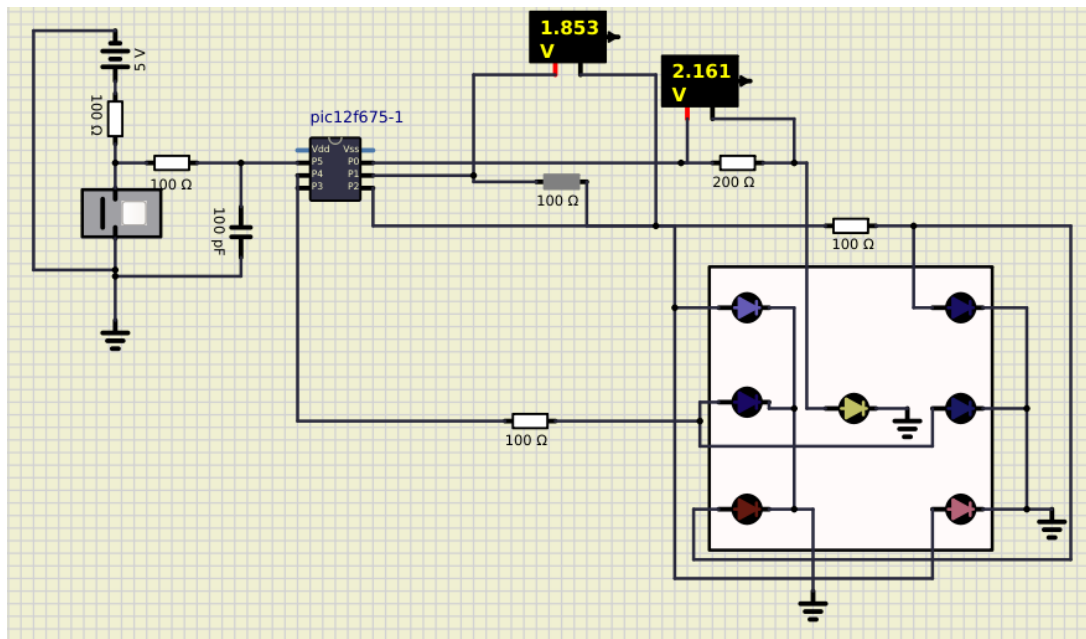


Figura 10: Demo del dado mostrando el lado con valor 3.

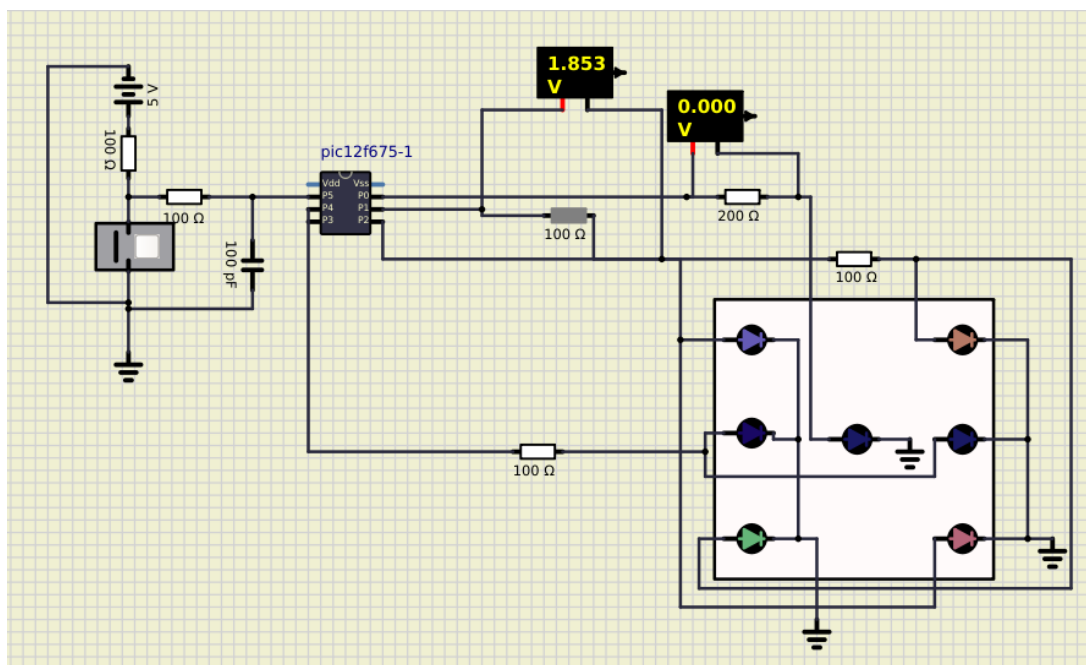


Figura 11: Demo del dado mostrando el lado con valor 4.

4. Conclusiones y recomendaciones

- Al programar un microcontrolador, una hoja de datos puede brindar mucha información tanto como para la parte electrónica como de programación.
- Es importante estudiar el mercado de microcontroladores para encontrar uno que se ajuste a las necesidades que requiere la tarea. Esto puede ayudar a reducir costos y energía así como evitar que lo escogida pueda cubrir las funciones que se busquen.
- Al programar un microcontrolador se debe tomar en cuenta las características físicas tanto como su capacidad de memoria para analizar alternativas para las funciones deseadas.

Referencias

1. Microchip (2022) PIC12F675. Recuperado de: <https://www.microchip.com/en-us/product/PIC12F675>
2. Microchip Technology Inc. (2010) PIC12F629/675 Data Sheet. Recuperado de: <https://ww1.microchip.com/downloads/aemDocuments/documents/MCU08/ProductDocuments/DataSheets/41190G.pdf>

5. Apéndice



PIC12F629/675

Data Sheet

8-Pin, Flash-Based 8-Bit
CMOS Microcontrollers

8-Pin Flash-Based 8-Bit CMOS Microcontroller

High-Performance RISC CPU:

- Only 35 Instructions to Learn
 - All single-cycle instructions except branches
- Operating Speed:
 - DC – 20 MHz oscillator/clock input
 - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect, and Relative Addressing modes

Special Microcontroller Features:

- Internal and External Oscillator Options
 - Precision Internal 4 MHz oscillator factory calibrated to $\pm 1\%$
 - External Oscillator support for crystals and resonators
 - 5 μ s wake-up from Sleep, 3.0V, typical
- Power-Saving Sleep mode
- Wide Operating Voltage Range – 2.0V to 5.5V
- Industrial and Extended Temperature Range
- Low-Power Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Detect (BOD)
- Watchdog Timer (WDT) with Independent Oscillator for Reliable Operation
- Multiplexed MCLR/Input Pin
- Interrupt-on-Pin Change
- Individual Programmable Weak Pull-ups
- Programmable Code Protection
- High Endurance Flash/EEPROM Cell
 - 100,000 write Flash endurance
 - 1,000,000 write EEPROM endurance
 - Flash/Data EEPROM Retention: > 40 years

Low-Power Features:

- Standby Current:
 - 1 nA @ 2.0V, typical
- Operating Current:
 - 8.5 μ A @ 32 kHz, 2.0V, typical
 - 100 μ A @ 1 MHz, 2.0V, typical
- Watchdog Timer Current
 - 300 nA @ 2.0V, typical
- Timer1 Oscillator Current:
 - 4 μ A @ 32 kHz, 2.0V, typical

Peripheral Features:

- 6 I/O Pins with Individual Direction Control
- High Current Sink/Source for Direct LED Drive
- Analog Comparator module with:
 - One analog comparator
 - Programmable on-chip comparator voltage reference (CVREF) module
 - Programmable input multiplexing from device inputs
 - Comparator output is externally accessible
- Analog-to-Digital Converter module (PIC12F675):
 - 10-bit resolution
 - Programmable 4-channel input
 - Voltage reference input
- Timer0: 8-Bit Timer/Counter with 8-Bit Programmable Prescaler
- Enhanced Timer1:
 - 16-bit timer/counter with prescaler
 - External Gate Input mode
 - Option to use OSC1 and OSC2 in LP mode as Timer1 oscillator, if INTOSC mode selected
- In-Circuit Serial Programming™ (ICSP™) via two pins

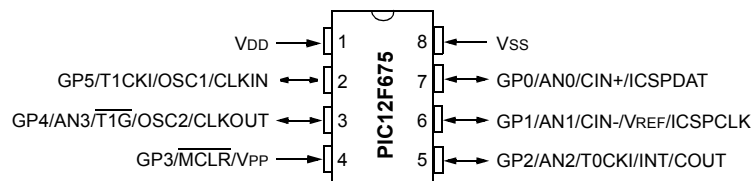
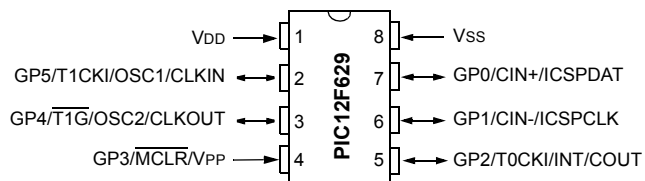
Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)				
PIC12F629	1024	64	128	6	—	1	1/1
PIC12F675	1024	64	128	6	4	1	1/1

* 8-bit, 8-pin devices protected by Microchip's Low Pin Count Patent: U.S. Patent No. 5,847,450. Additional U.S. and foreign patents and applications may be issued or pending.

PIC12F629/675

Pin Diagrams

8-pin PDIP, SOIC, DFN-S, DFN



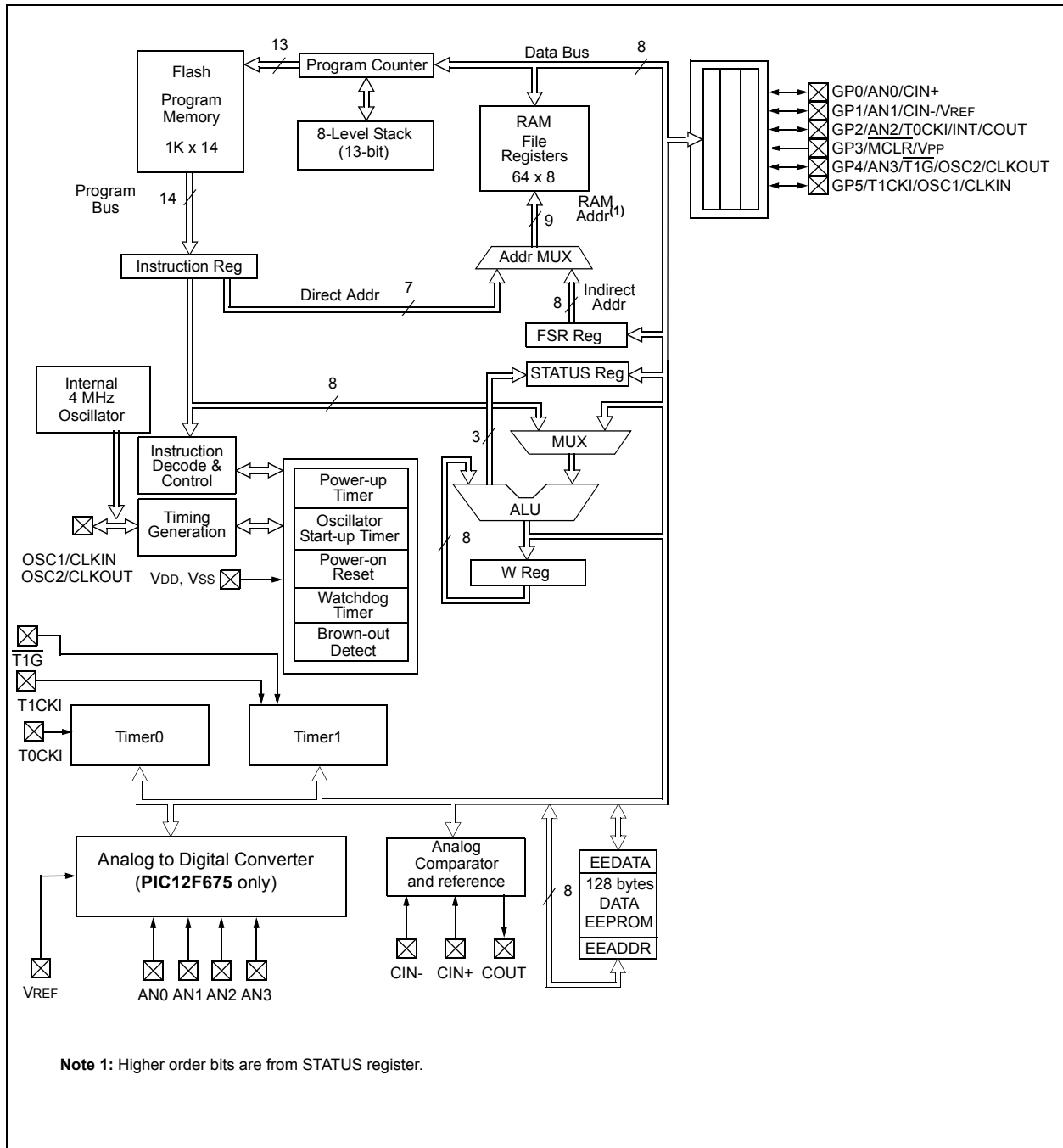
1.0 DEVICE OVERVIEW

This document contains device specific information for the PIC12F629/675. Additional information may be found in the PIC® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this Data

Sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC12F629 and PIC12F675 devices are covered by this Data Sheet. They are identical, except the PIC12F675 has a 10-bit A/D converter. They come in 8-pin PDIP, SOIC, MLF-S and DFN packages. Figure 1-1 shows a block diagram of the PIC12F629/675 devices. Table 1-1 shows the pinout description.

FIGURE 1-1: PIC12F629/675 BLOCK DIAGRAM



3.0 GPIO PORT

There are as many as six general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

Note: Additional information on I/O ports may be found in the PIC® Mid-Range Reference Manual, (DS33023).

3.1 GPIO and the TRISIO Registers

GPIO is an 6-bit wide, bidirectional port. The corresponding data direction register is TRISIO. Setting a TRISIO bit (= 1) will make the corresponding GPIO pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISIO bit (= 0) will make the corresponding GPIO pin an output (i.e., put the contents of the output latch on the selected pin). The exception is GP3, which is input-only and its TRISIO bit will always read as '1'. Example 3-1 shows how to initialize GPIO.

Reading the GPIO register reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified, and then written to the PORT data latch. GP3 reads '0' when MCLREN = 1.

The TRISIO register controls the direction of the GP pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISIO

register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

Note: The ANSEL (9Fh) and CMCON (19h) registers (9Fh) must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'. The ANSEL register is defined for the PIC12F675.

EXAMPLE 3-1: INITIALIZING GPIO

```
BCF    STATUS,RP0    ;Bank 0
CLRF   GPIO          ;Init GPIO
MOVLW  07h           ;Set GP<2:0> to
MOVWF  CMCON         ;digital IO
BSF    STATUS,RP0    ;Bank 1
CLRF   ANSEL         ;Digital I/O
MOVLW  0Ch           ;Set GP<3:2> as inputs
MOVWF  TRISIO        ;and set GP<5:4,1:0>
                           ;as outputs
```

3.2 Additional Pin Functions

Every GPIO pin on the PIC12F629/675 has an interrupt-on-change option and every GPIO pin, except GP3, has a weak pull-up option. The next two sections describe these functions.

3.2.1 WEAK PULL-UP

Each of the GPIO pins, except GP3, has an individually configurable weak internal pull-up. Control bits WPUx enable or disable each pull-up. Refer to Register 3-3. Each weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset by the GPPU bit (OPTION<7>).

REGISTER 3-1: GPIO: GPIO REGISTER (ADDRESS: 05h)

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **GPIO<5:0>:** General Purpose I/O pin

1 = Port pin is >V_{IH}

0 = Port pin is <V_{IL}

PIC12F629/675

REGISTER 3-2: TRISIO: GPIO TRI-STATE REGISTER (ADDRESS: 85h)

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0
bit 7		bit 0					

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **TRISIO<5:0>:** General Purpose I/O Tri-State Control bit

1 = GPIO pin configured as an input (tri-stated)

0 = GPIO pin configured as an output

Note: TRISIO<3> always reads '1'.

REGISTER 3-3: WPU: WEAK PULL-UP REGISTER (ADDRESS: 95h)

U-0		U-0		R/W-1		R/W-1		U-0		R/W-1		R/W-1		R/W-1	
—		—		WPU5		WPU4		—		WPU2		WPU1		WPU0	
bit 7														bit 0	

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **WPU<5:4>:** Weak Pull-up Register bit

1 = Pull-up enabled

0 = Pull-up disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **WPU<2:0>:** Weak Pull-up Register bit

1 = Pull-up enabled

0 = Pull-up disabled

Note 1: Global $\overline{\text{GPPU}}$ must be enabled for individual pull-ups to be enabled.

2: The weak pull-up device is automatically disabled if the pin is in Output mode (TRISIO = 0).

12.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings†

Ambient temperature under bias	-40 to +125°C
Storage temperature	-65°C to +150°C
Voltage on VDD with respect to VSS	-0.3 to +6.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS	-0.3 to +13.5V
Voltage on all other pins with respect to VSS	-0.3V to (VDD + 0.3V)
Total power dissipation ⁽¹⁾	800 mW
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > VDD)	± 20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > VDD)	± 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

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 to maximum rating conditions for extended periods may affect device reliability.

Note: Voltage spikes below VSS at the $\overline{\text{MCLR}}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100 Ω should be used when applying a “low” level to the $\overline{\text{MCLR}}$ pin, rather than pulling this pin directly to VSS.