

UNIVERSIDAD DE COSTA RICA

Escuela de Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

Luis Javier Herrera Barrantes, B93840

MSc. Marco Villalta Fallas

4 de setiembre del 2022

Laboratorio 1: Introducción a microcontroladores y manejo de GPIOs

Resumen

El presente reporte de laboratorio es una forma introducción a los conocimientos básicos de microcontroladores. Se trabajó con el microcontrolador PIC12f675 en el software de simulación SimulIDE con el objetivo de crear un juego de bingo que debía desplegar un número aleatorio ubicado dentro del rango de 0 a 99. Para esto se tuvo que estudiar a fondo el funcionamiento básico del microcontrolador y saber como programarlo de la forma indicada. Además, se utilizaron diferentes componentes electrónicos como: un display doble de siete segmentos, un botón activador, un codificador y resistores. Esto con el fin de lograr cumplir con el diseño indicado. El funcionamiento del microcontrolador fue el esperado y se logró generar el número aleatorio de dos dígitos mostrado en el display al mantener el botón presionado.

1. Nota Teórica

1.1. Información del Microcontrolador

Para la elaboración del laboratorio 1 se utilizó el microcontrolador PIC12f675 del fabricante Microchip. Este cuenta con 8 pines en los cuales 6 pueden funcionar como entrada o salida y es basado en la arquitectura RISC de microprocesadores, además de operar a una frecuencia de 20 MHz [1]. También, este posee una célula de memoria FLASH para cargar el programa, opera en un rango de voltaje que va desde los -2.0 V a los 5.0 V y tiene los modos de direccionamiento directo, indirecto y relativo [1]. La organización de los pines en el microcontrolador se evidencia en la Figura 1.

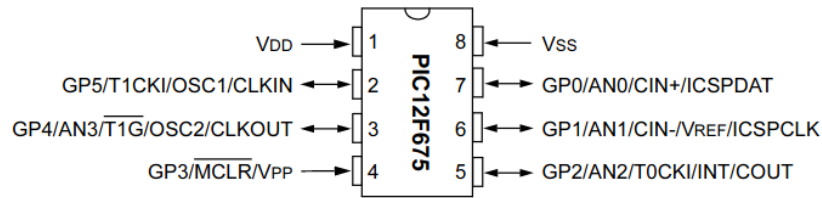


Figura 1: Distribución de pines del microcontrolador PIC12f675 [1].

Entre los periféricos con los que cuenta este tipo de microcontrolador están los 6 pines I/O o de propósito general GPIO, un comparador analógico, un módulo de conversión analógico a digital, un temporizador y una fuente de alta corriente para poder trabajar con LEDs.

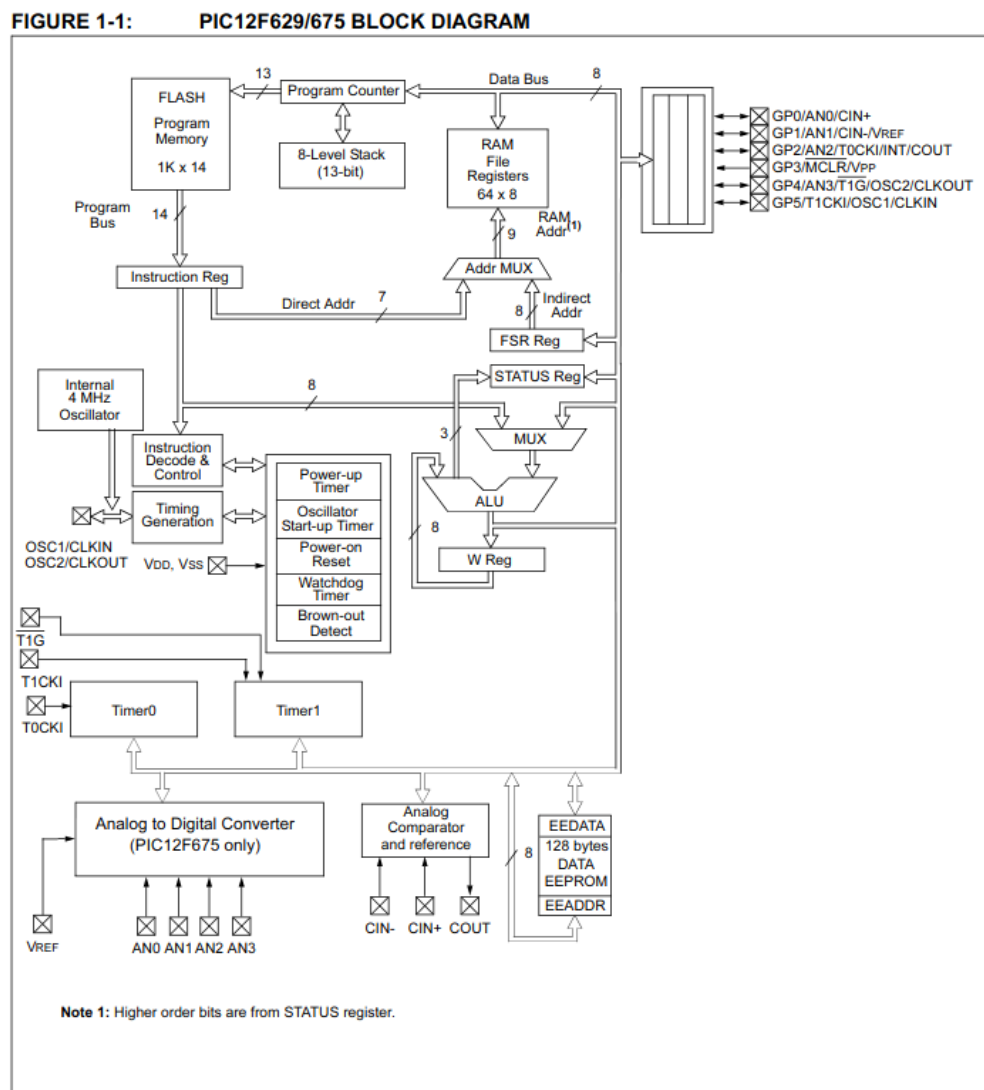


Figura 2: Diagrama de bloques de microcontrolador PIC12f675 [1].

De los periféricos con los que cuenta en el microcontrolador se modificó el registro GPIO

el cual tiene la utilidad de ser manipulado para seleccionar el estado de los pines de salida. Se escribe con un 0 si se quiere activar cierto pin con un 0 lógico y con un 1 si se quiere activar cierto pin con un 1 lógico. Al principio del programa se puso este registro en 0 y se fue modificando según fuera necesario para desarrollar la tarea principal. También, se utilizó en registro TRISIO que sirve para definir cuales de los pines del microcontrolador van a funcionar como salidas y cuales como entrada. Este registro se configuro con 5 pines en bajo (salidas) y el puerto 4 (GP3) en alto para que funcione como entrada. Por último, se cambió el registro CONFIG con la utilización de macros para así desactivar el RESET del puerto GP3 y que este pudiera funcionar como entrada además de deshabilitar el Watchdog Timer.

1.2. Componentes electrónicos complementarios

Unos de los complementos electrónicos utilizados en el diseño fue un decodificador de tipo BCD a siete segmentos. La función de este era codificar las señales que venían de los puertos de microcontrolador para mostrar el número correspondiente en la pantalla doble de siete segmentos. La codificación de estos números, como su nombre lo indica, es en formato BCD. Cuenta con cuatro conexiones de entrada por las cuales entran cuatro bits, estos se codifican y posteriormente salen las señales de salida por los puertos del a hasta el g correspondientes a cierto número para mostrar en la pantalla. Además, cuenta con un enable \overline{OE} para habilitar o deshabilitar el decodificador. Para efectos prácticos se encontró el decodificador BCD a 7 segmentos DM7446A de la marca Fairchild Semiconductors [2].

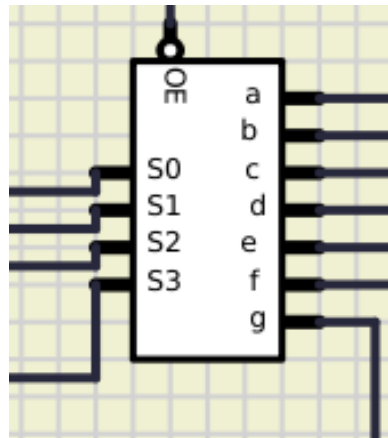


Figura 3: Codificador de 7 segmentos.

También, se utilizó un display doble de siete segmentos con ánodo común como el que se muestra en la Figura 4. Para efectos de diseño físico se tomo la hoja de datos del display doble de la marca Agilent Technologies, 14.2 mm (0.56 inch) General Purpose Two Digit Seven Segment Displays [3]. Para mostrar el color verde este puede soportar una potencia de 105 mW, corriente DC por segmento de 30mA y un voltaje en reversa de 5 V. Todo esto para un rango

de temperaturas que varía entre $-35\text{ }^{\circ}\text{C}$ a $+85\text{ }^{\circ}\text{C}$.

En el diseño se utilizó también un inversor para así poder seleccionar cual de dos display debía mostrar el número y desplegar números diferentes en ambos displays al mismo tiempo. Se agregó un switch que sirviera como botón pulsador para así dar la orden de generar los números correspondientes y resistores para proteger los componentes electrónicos de altas corrientes.

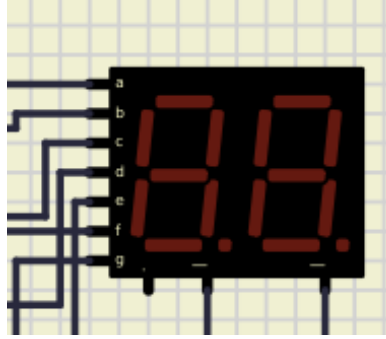


Figura 4: Codificador de 7 segmentos.

1.3. Diseño del circuito

Primeramente se tuvo que diseñar el valor de resistencia para el resistor de pull-down. Este se utiliza como modo de complemento al botón pulsador para asegurar que cuando el botón se pulse, se propaguen los 5 V al pin de entrada del microcontrolador. Esta conexión se realizó en el software de simulación SimulIDE como se muestra en la Figura 5.

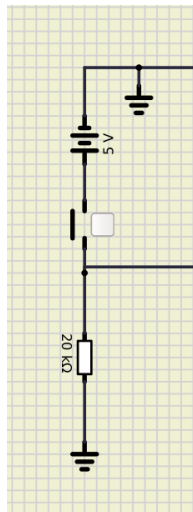


Figura 5: Conexión de resistor de pull-down.

Este resistor se va a encargar de mantener el pin de entrada bajo hasta el momento en que se active el respectivo botón y se desee poner la entrada en alto. Se elige entonces un valor grande

para este resistor de $R_p = 10\text{ k}\Omega$ con el fin de asegurarse por completo de que este valor va a ser mucho mayor que el valor de resistencia interna del botón R_{int} . Con esto presente, al presionar el botón:

$$V_{Rp} = \frac{5V \cdot 20k\Omega}{(20k + R_{int})} \rightarrow R_p \gg R_{int} \quad (1)$$

Pero R_{int} se toma como despreciable, entonces:

$$V_{Rp} \approx \frac{5V \cdot 20k\Omega}{20k\Omega} = 5V = V_{GP3} \quad (2)$$

Para proteger el display de siete segmentos de corrientes altas mayores a los 30mA, se colocaron resistores de $50\text{ }\Omega$ y se realizaron simulaciones con un amperímetro (como se muestra en la Figura 6) para asegurarse que la corriente que alimenta al display no supere la corriente máxima soportada.

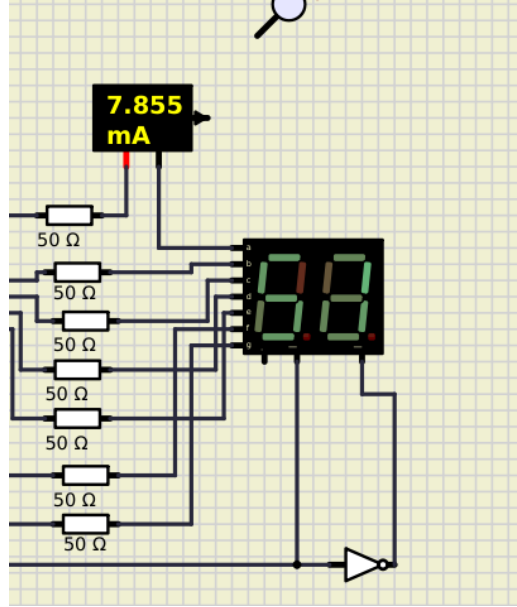


Figura 6: Medición de corriente de alimentación al display.

De tal forma, el esquemático del circuito completo propuesto fue el mostrado en la Figura 7.

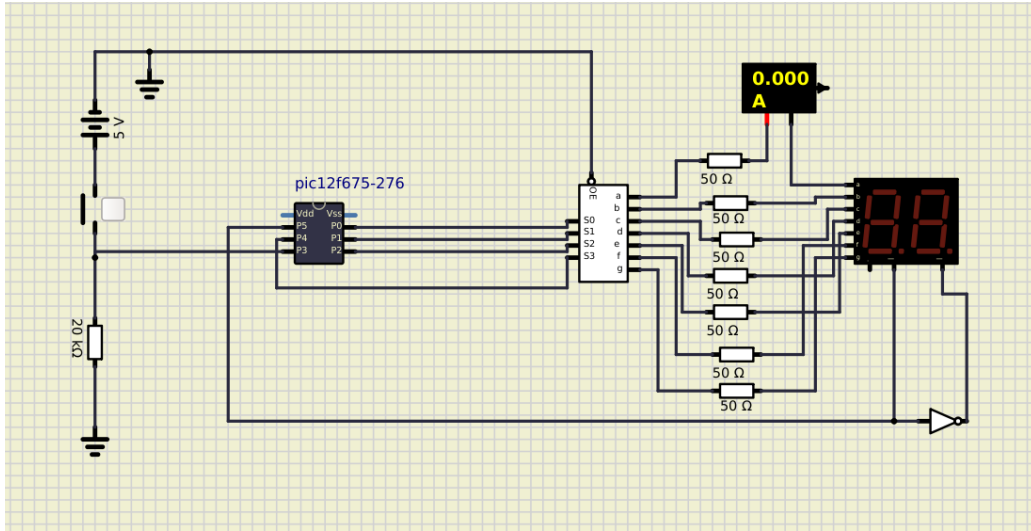


Figura 7: Esquemático del diseño completo.

El pin cinco del microcontrolador se utilizó para ser conectado al cátodo del display de siete segmentos y así enviar la señal digital con un '1' o un '0' para elegir el display donde se quiere mostrar el número generado en el programa. Y, el pin 3 se utilizó como entrada para leer la señal generada por el botón. El resto de pines se utilizaron como salidas para codificar los números mostrados en el display. En el Cuadro 1 se muestran el costo monetario en el caso de que se quisiera implementar de forma física el diseño (esto sin contar el precio de protoboards, fuentes de alimentación y/o cables de conexión). El costo menor encontrado para el microcontrolador PIC12F675 fue de \$ 1.679 [4] y el costo total calculado del diseño contando todas sus partes fue de \$ 9.819.

1.4. Lista de componentes y costo

Lista de componentes		
Componente	Cantidad	Precio
PIC12f675	1	\$ 1.679
Resistores (50 Ω)	7	\$ 3.36
Resistor (20k Ω)	1	\$ 0.57
Inversor	1	\$ 0.16
Botón	1	\$ 0.1
Decodificador	1	\$ 3.95
Total		\$ 9.819

Cuadro 1: Componentes utilizados en el diseño y costo de implementación

2. Desarrollo/Análisis de Resultados

Para lograr resolver el problema de falta de hardware, ya que el microcontrolador PIC12f675 cuenta con una cantidad limitada de pines, se utilizó un decodificador BCD a 7 segmentos. Solo bastó entonces hacer uso de cuatro pines GPIO del microcontrolador para poder representar cualquier número de 0 al 9 en un display mandando la codificación binaria correspondiente para cada número. Adicionalmente, se tuvo que utilizar un pin GPIO como salida extra para enviar una señal lógica al ánodo común del display de 7 segmentos y elegir cual de las dos pantallas debía ser la que mostrara el número.

2.1. Funcionamiento de software

En el programa creado para describir el funcionamiento del microcontrolador se crearon dos funciones principales que realizan la mayor parte del trabajo al correr el ciclo *while* principal. La primera se identificó como *mostar_num* consta de dos argumentos de tipo entero que son: el número en sí a desplegar en el display y el otro se utiliza para saber en cuál de los displays se quiere desplegar dicho número. Dentro de la función lo que se hace es codificar cada número del 0 al 9 para poder mandar las señales correctas por los pines GPIO de salida que están conectados al decodificador. Dependiendo de el display que se elija se activa o no el pin 5 que corresponde al encargado de mandar la señal de selección para escoger el display donde se va mostrar el número.

Por otro lado, se creó la función con el nombre de *rando* la cuál se encarga de generar un número pseudoaleatorio entre un rango en específico. Para este caso se eligió el rango que va desde el 0 al 9 y cumplir así con los requerimientos mencionados en el enunciado del laboratorio. Debido a que no se puede utilizar la función *rand()* incluida en la librerías de C, se hizo uso de las 'static variables' para generar el número pseudoaleatorio. Las 'static variables' son un tipo de dato que cuentan con la característica que después de ser declaradas solamente se destruyen una vez la totalidad del programa haya corrido [5]. A partir de esto se creó un algoritmo utilizando una variable estática ingresada y los límites superior y inferior del rango para generar así un número pseudoaleatorio cada vez que se corre la función. Cabe resaltar que al no ser número completamente aleatorios los que se generan, cabe la posibilidad que se repitan números algunas veces. El funcionamiento general del programa se describe en el diagrama de la Figura 8.

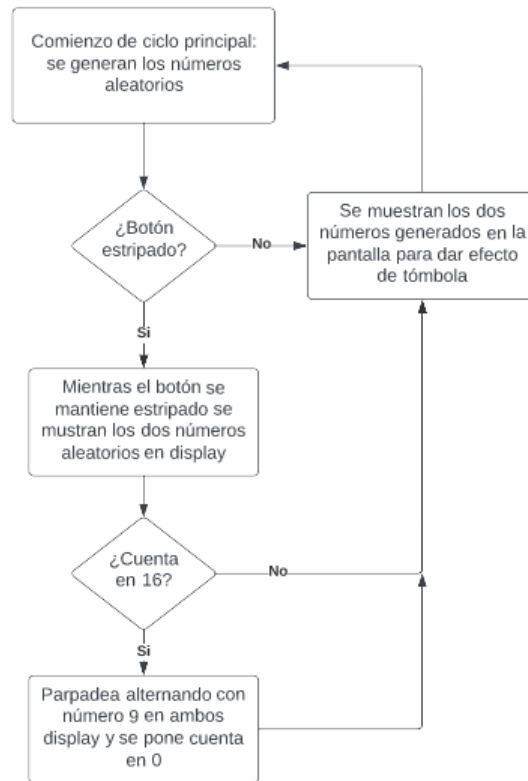


Figura 8: Diagrama de flujo que describe funcionamiento del programa creado.

2.2. Funcionamiento electrónico

Con el fin de asegurarse de que el botón y el resistor de pull-down estén funcionando de la manera indicada y pasando al pin GPIO 3 buenos '1's y '0's lógicos se colocó un voltímetro para medir el voltaje en ese nodo como se muestra en la Figura 9.

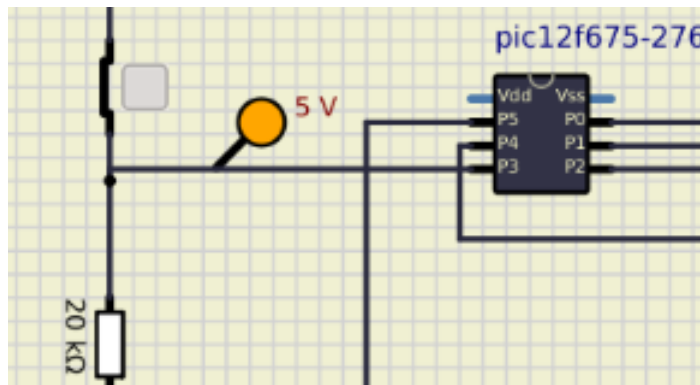


Figura 9: Funcionamiento electrónico al presionar el botón.

Además, se colocaron voltímetros en la sección del esquemático donde se encuentra el decoder para saber con certeza el voltaje que se está transmitiendo al decoder en sí y al display de 7 segmentos. De esta forma se garantiza que se está trabajando con los valores de voltaje correctos.

Además, se colocó un amperímetro entre el decoder y el display para así tener en cuenta que este valor de corriente no puede superar los 30mA (valor máximo que soporta el display). La Figura 10 resume el funcionamiento electrónico de esta parte.

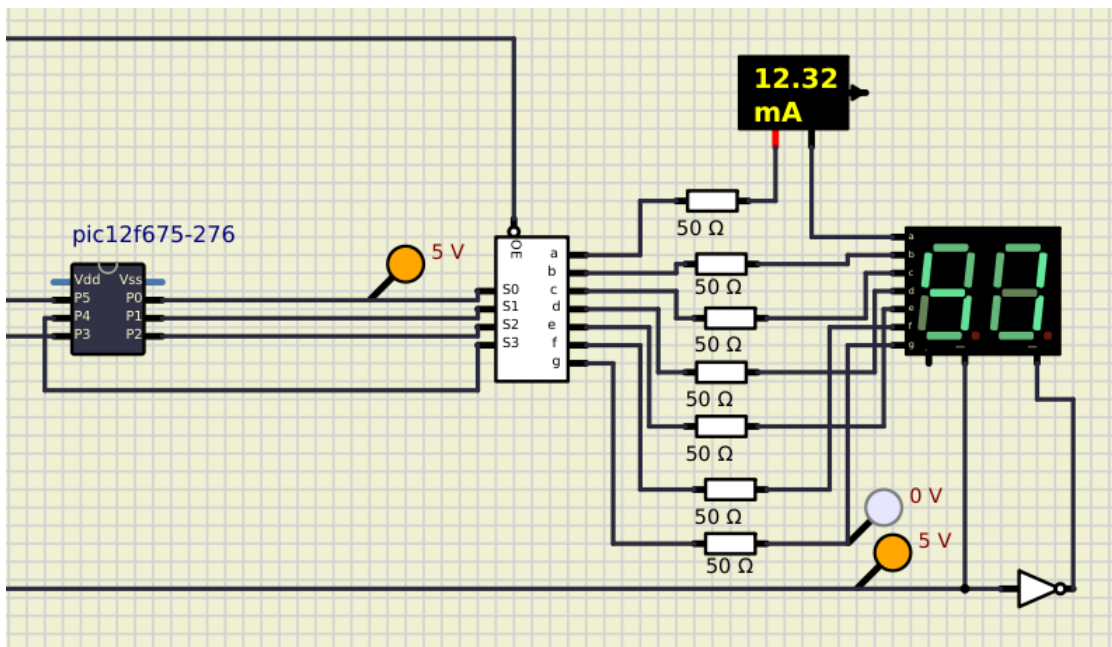


Figura 10: Funcionamiento electrónico de sección con decoder y display.

Cabe resaltar en la Figura 10 no se puede ver un número de dos dígitos claramente en el display debido a que el botón no se encuentra presionado. En estos casos el programa va a empezar a generar números y mostrarlos en la pantalla. Lo hace tan rápido que da el efecto que aparece en dicha Figura. Si se presiona el botón, un número se apreciará claramente en el display, como es el caso de la Figura 11.

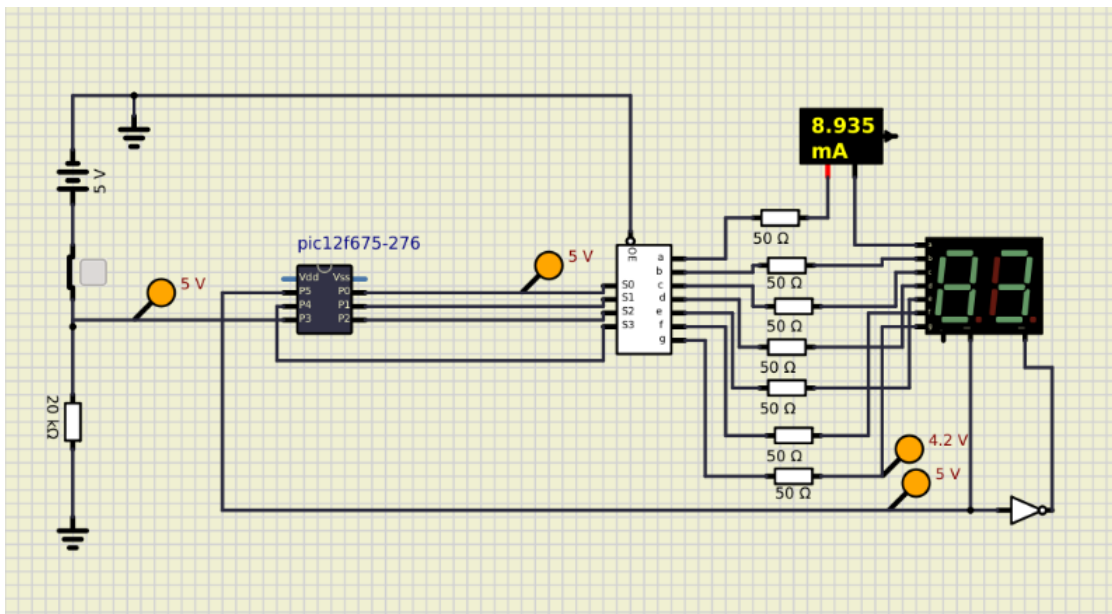


Figura 11: Funcionamiento a la hora de mostrar número aleatorio.

3. Conclusiones y recomendaciones

Para poder empezar a aprender de manera indicada el funcionamiento de los microcontroladores es de suma importancia primero estudiar el funcionamiento específico del microcontrolador que se esté utilizando. Esto para evitar posibles confusiones a la hora de empezar con el diseño que pueden resultar en perdidas de tiempo considerable, además de que cada microcontrolador tiene sus características específicas de operación. Cabe resaltar entonces que se logró elaborar un diseño funcional utilizando los ocho pines del microcontrolador PIC12f675 para simular un juego de bingo. Esto con la ayuda de diversos componentes electrónicos y utilizando un botón como medio indicador para generar el número pseudoaleatorio. Además, se cumplió con el requerimiento donde al llegar al dieciseisavo número ingresado, el juego se termina y la pantalla parpadea repetidas veces con el número 99. Se cumplió entonces con todos los requerimientos solicitados en el diseño para este laboratorio.

El link al repositorio de Github es el siguiente:

https://github.com/luisja01/Laboratorio_Microcontroladores.git

Referencias

- [1] Microchip, “PIC12F629/675 - Data Sheet,” n.º 1, 2003.
- [2] D. D. B. to 7-Segment Decoders/Drivers, Fairchild Semiconductor, mar. de 1998. dirección: <https://www.farnell.com/datasheets/59373.pdf>.
- [3] 1. mm (0.56 inch) General Purpose Two Digit Seven Segment Displays, Agilent Technologies, Berkeley, California, jul. de 2004. dirección: <https://www.farnell.com/datasheets/95204.pdf>.
- [4] Octopart. “Microchip PIC12F675-I/SN.” (2022), dirección: <https://octopart.com/pic12f675-i%5C%2Fsn-microchip-106501>.
- [5] S. Arya. “Static Vatiabile in C.” (2022), dirección: <https://www.scaler.com/topics/static-variables-in-c/>.

Apéndice A - Hojas de Fabricante



PIC12F629/675

Data Sheet

8-Pin FLASH-Based 8-Bit
CMOS Microcontrollers

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

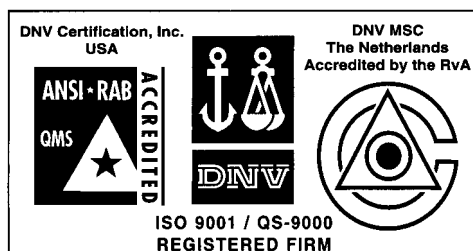
Accuron, dsPIC, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICkit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerTool, rPIC, rLAB, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.



PIC12F629/675

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 $\overline{\text{MCLR}}$ /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

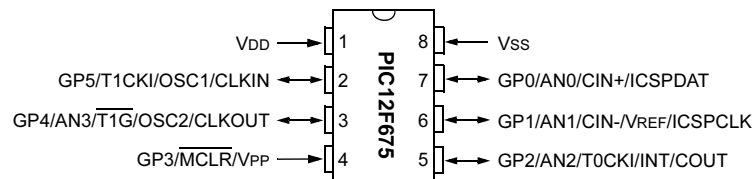
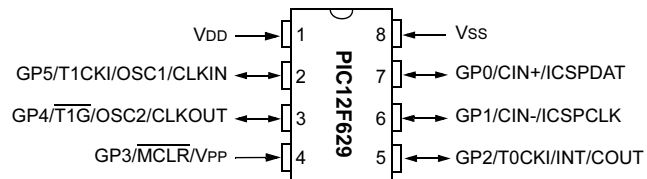
Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	FLASH (words)	SRAM (bytes)	EEPROM (bytes)				
PIC12F629	1024	64	128	6	—	1	1/1
PIC12F675	1024	64	128	6	4	1	1/1

* 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



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 PICmicro™ Mid-Range Reference Manual, (DS33023)

3.1 GPIO and the TRISIO Registers

GPIO is an 6-bit wide, bi-directional 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 Hi-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

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}

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

PIC12F629/675

REGISTER 3-2: TRISIO — GPIO TRISTATE REGISTER (ADDRESS: 85h)

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

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.

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

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			

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.

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

3.2.2 INTERRUPT-ON-CHANGE

Each of the GPIO pins is individually configurable as an interrupt-on-change pin. Control bits IOC enable or disable the interrupt function for each pin. Refer to Register 3-4. The interrupt-on-change is disabled on a Power-on Reset.

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of GPIO. The 'mismatch' outputs of the last read are OR'd together to set, the GP Port Change Interrupt flag bit (GPIF) in the INTCON register.

This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of GPIO. This will end the mismatch condition.
- Clear the flag bit GPIF.

A mismatch condition will continue to set flag bit GPIF. Reading GPIO will end the mismatch condition and allow flag bit GPIF to be cleared.

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the GPIF interrupt flag may not get set.

REGISTER 3-4: IOC — INTERRUPT-ON-CHANGE GPIO REGISTER (ADDRESS: 96h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
bit 7		bit 0					

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

bit 5-0 **IOC<5:0>:** Interrupt-on-Change GPIO Control bit
 1 = Interrupt-on-change enabled
 0 = Interrupt-on-change disabled

Note 1: Global interrupt enable (GIE) must be enabled for individual interrupts to be recognized.

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

14.2 mm (0.56 inch) General Purpose Two Digit Seven Segment Displays

Technical Data

HDSP-52xE Series
HDSP-52xG Series
HDSP-52xY Series

Features

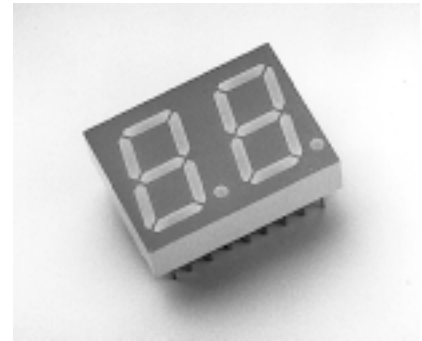
- **Industry Standard Size**
- **Industry Standard Pin-Out**
15.24 mm (0.6 in.) DIP Leads
on 2.54 mm (0.1 in.) Centers
- **Choice of Colors**
Red, Green, Yellow
- **Mitered Font**
Mitered Corners on Segments
- **Gray Face Paint**
Gray Package Gives Optimum
Contrast
- **± 50° Viewing Angle**
- **Design Flexibility**
Common Anode or Common
Cathode
- **Categorized for Luminous
Intensity**
- **Green and Yellow
Categorized for Color**

Applications

- **Suitable for Indoor Use**
- **Not Recommended for
Industrial Applications, i.e.
Operating Temperatures
Requirements Exceeding
85°C or Below -35°C^[1]**
- **Extreme Temperature
Cycling Not
Recommended^[2]**

Description

These 14.2 mm (0.56 inch) two digit displays use industry standard size and pin-out. The devices are available as either common anode or common cathode. These gray-faced displays are available in a choice of high efficiency red (HER), green, or yellow colors. The



HDSP-521x and HDSP-523x series are suitable for indoor use.

These parts are subjected to Outgoing Quality Assurance (OQA) inspection with AQL of 0.065% for functional and visual/cosmetic rejects.

Devices

HER HDSP-	Green HDSP-	Yellow HDSP-	Description
521E	521G	521Y	Common Anode Right Hand Decimal
523E	523G	523Y	Common Cathode Right Hand Decimal

Notes:

1. For industrial applications, it is recommended to use HDSP-5521/5523/5621/5623/5721/5723.
2. For details, please contact your local Agilent sales office or an authorized distributor.

Absolute Maximum Ratings at $T_A=25^{\circ}\text{C}$

Parameter	HER HDSP-521E HDSP-523E	Green HDSP-521G HDSP-523G	Yellow HDSP-521Y HDSP-523Y	Units
Average Power per Segment or DP	62.5	105	45	mW
Peak Forward Current per Segment or DP (1/10 Duty Cycle, 0.1 ms Pulse Width)	90	90	60	mA
DC Forward Current per Segment or DP ^[1]	25 ^[1]	30 ^[3]	20 ^[2]	mA
Reverse Voltage per Segment or DP	5	5	5	V
Operating Temperature	-35 to +85	-35 to +85	-35 to +85	$^{\circ}\text{C}$
Storage Temperature	-35 to +85	-35 to +85	-35 to +85	$^{\circ}\text{C}$
Wave Soldering Temperature for 3 seconds ^[4] (2 mm [0.063 in.] below Body)	250	250	250	$^{\circ}\text{C}$

Notes:

- Derate above 25°C at 0.33 mA/ $^{\circ}\text{C}$.
- Derate above 25°C at 0.27 mA/ $^{\circ}\text{C}$.
- Derate above 40 celcius at 0.35 mA/celcius.
- Not recommended to be soldered more than 2 times. Minimum interval between solderings is 15 minutes. Total soldering time not to exceed 5 seconds.

Optical/Electrical Characteristics at $T_A=25^{\circ}\text{C}$

High Efficiency Red

Devices HDSP-	Parameter	Symbol	Min.	Typ.	Max.	Units	Test Conditions
521E 523E	Luminous Intensity/Segment (Segment Average) ^[1,2]	I_V	2.28	4.00	7.69	mcd	$I_F = 10\text{ mA}$
	Forward Voltage/Segment or DP	V_F		2.05	2.60	V	$I_F = 20\text{ mA}$
	Peak Wavelength	λ_{PEAK}		640		nm	
	Dominant Wavelength ^[3]	λ_d		628		nm	$I_F = 10\text{ mA}$
	Reverse Voltage/Segment or DP ^[4]	V_R	5.0			V	$I_R = 100\text{ }\mu\text{A}$
	Temperature Coefficient of V_F /Segment or DP	$\Delta V_F/^{\circ}\text{C}$		-2		mV/ $^{\circ}\text{C}$	

Green

Devices HDSP-	Parameter	Symbol	Min.	Typ.	Max.	Units	Test Conditions
521G 523G	Luminous Intensity/Segment (Segment Average) ^[1,2]	I_V	2.28	3.50	5.13	mcd	$I_F = 10\text{ mA}$
	Forward Voltage/Segment or DP	V_F		2.0	2.4	V	$I_F = 10\text{ mA}$
	Peak Wavelength	λ_{PEAK}		568		nm	
	Dominant Wavelength ^[3]	λ_d	564.5	570	576.5	nm	$I_F = 10\text{ mA}$
	Reverse Voltage/Segment or DP ^[4]	V_R	5.0			V	$I_R = 100\text{ }\mu\text{A}$
	Temperature Coefficient of V_F /Segment or DP	$\Delta V_F/^{\circ}\text{C}$		-2		mV/ $^{\circ}\text{C}$	

DM7446A, DM7447A BCD to 7-Segment Decoders/Drivers

General Description

The 46A and 47A feature active-low outputs designed for driving common-anode LEDs or incandescent indicators directly. All of the circuits have full ripple-blanking input/output controls and a lamp test input. Segment identification and resultant displays are shown on a following page. Display patterns for BCD input counts above nine are unique symbols to authenticate input conditions.

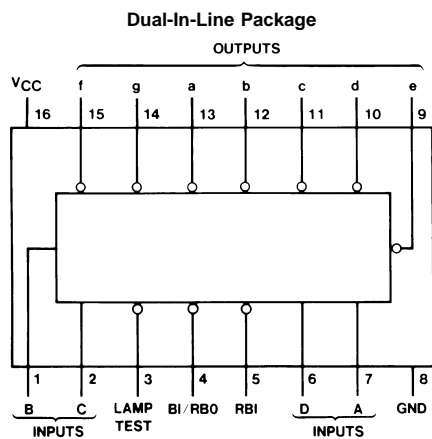
All of the circuits incorporate automatic leading and/or trailing-edge, zero-blanking control (RBI and RBO). Lamp test (LT) of these devices may be performed at any time

when the BI/RBO node is at a high logic level. All types contain an overriding blanking input (BI) which can be used to control the lamp intensity (by pulsing) or to inhibit the outputs.

Features

- All circuit types feature lamp intensity modulation capability
- Open-collector outputs drive indicators directly
- Lamp-test provision
- Leading/trailing zero suppression

Connection Diagram



Order Number DM5447AJ, DM7446AN or DM7447AN
See Package Number J16A or N16E