



**Instituto Politécnico
Nacional
Escuela Superior de
Computo**



Unidad de Aprendizaje:

Instrumentación y control

Grupo: 5CV4

Practica

Introducción a PIC

Integrantes de equipo: Arzate Salazar Emiliano

Bautista Ríos Alfredo

Galicia Rodríguez Andrés

Profesor: Cervantes De Anda Ismael

Fecha de entrega:

3/Junio/2024

Contenido

Objetivo.....	1
Introducción teórica	1
1. PIC 16F628A	1
Características principales	2
Aplicaciones	2
2. OSC (Oscilador)	2
Tipos de osciladores.....	3
3. RAM (Memoria de acceso aleatorio).....	3
Características de la RAM del PIC 16F628A	3
4. EEPROM (Memoria de solo lectura programable y borrrable eléctricamente)	3
Características de la EEPROM del PIC 16F628A.....	3
5. Timer (Temporizador).....	4
6. UART (Universal Asynchronous Receiver Transmitter)	4
Características del UART del PIC 16F628A	4
7. Voltajes y corriente de trabajo	4
8. Pulldown	5
Desarrollo	5
Componentes y Conexiones.....	8
Armado del Circuito	8
Funcionamiento del Circuito	9
Simulaciones.....	9
1. Estado 0 lógico	9
2. Estado 1 lógico	10
Conclusión	12
Bibliografía	13

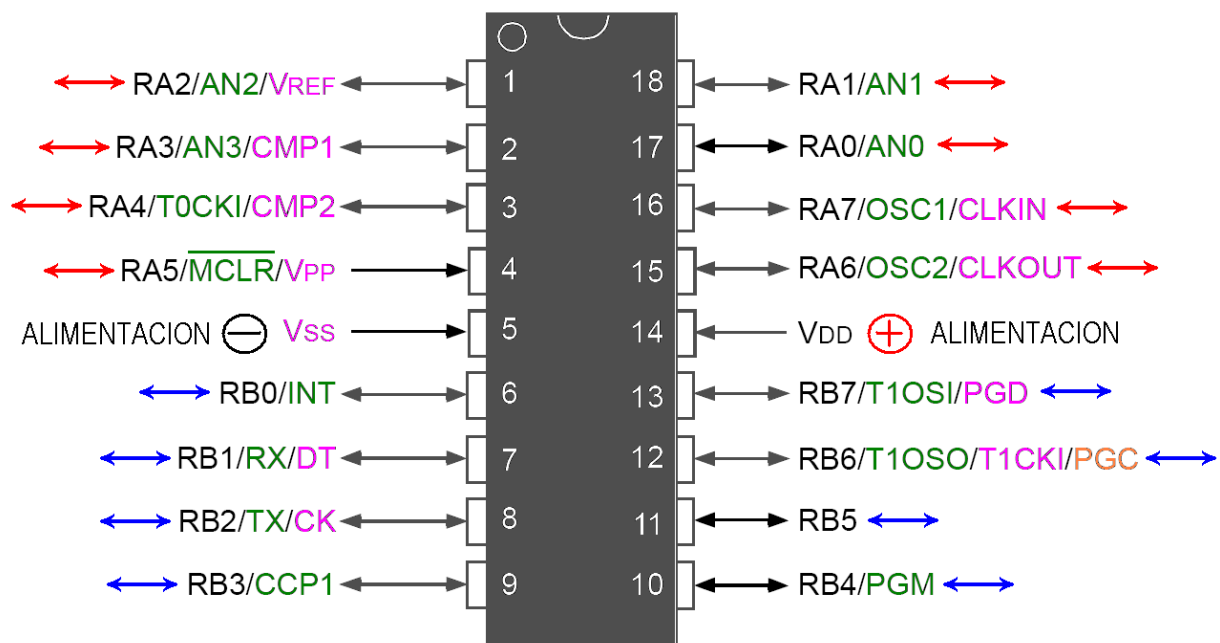
Objetivo

Se trabajará con un circuito simple utilizando el PIC 16F628A asignando pines de entrada y salida con la asignación de proceso necesario para su uso. Además, exploraremos cómo configurar los registros del PIC para controlar las salidas necesarias utilizando interrupciones o timers.

Esto nos permitirá aprender sobre la programación de microcontroladores, la configuración de pines de entrada/salida, y cómo interactuar con componentes externos. Una vez que dominemos esto, podremos avanzar hacia proyectos más complejos, como la adquisición de datos desde un ADC o la comunicación con otros dispositivos.

Introducción teórica

1. PIC 16F628A



PIC16F627A / **PIC16F628A** / PIC16F648A

GNU licence by JimyXT

Figura 1. Datasheet del PIC16F628A

El PIC 16F628A es un microcontrolador de 8 bits de la familia PIC de Microchip Technology. Es un dispositivo económico, versátil y fácil de programar, lo que lo convierte en una opción popular para aficionados y profesionales en electrónica.

Características principales

- Arquitectura RISC avanzada
- Conjunto de instrucciones reducido de 35 instrucciones
- Oscilador interno de 4 MHz
- 128 bytes de memoria EEPROM
- 22 pines de E/S
- 3 temporizadores
- Un módulo USART
- Dos comparadores
- Una referencia de voltaje programable
- Bajo consumo de energía
- Amplio rango de voltaje de operación (2.0 - 5.5 V)

Aplicaciones

- Control de motores
- Sensores
- Adquisición de datos
- Interfaz hombre-máquina
- Domótica
- Proyectos educativos

2. OSC (Oscilador)

El oscilador es un circuito electrónico que genera una señal de reloj de frecuencia estable. Esta señal de reloj es esencial para el funcionamiento del microcontrolador, ya

que sincroniza todas las operaciones internas. El PIC 16F628A puede utilizar un oscilador interno o externo.

Tipos de osciladores

- **Oscilador interno:** El PIC 16F628A tiene un oscilador interno de 4 MHz. Este oscilador es conveniente para aplicaciones que no requieren alta precisión de temporización.
- **Oscilador externo:** Un oscilador externo puede proporcionar mayor precisión de temporización que un oscilador interno. Se puede utilizar un cristal de cuarzo o un circuito RC como oscilador externo.

3. RAM (Memoria de acceso aleatorio)

La RAM es una memoria volátil que almacena datos que el microcontrolador puede leer y escribir. La RAM se utiliza para almacenar variables, datos temporales y el programa en ejecución. Cuando se apaga el microcontrolador, se borra el contenido de la RAM.

Características de la RAM del PIC 16F628A

- 224 bytes de RAM

4. EEPROM (Memoria de solo lectura programable y borrrable eléctricamente)

La EEPROM es una memoria no volátil que almacena datos que el microcontrolador puede leer y escribir. La EEPROM se utiliza para almacenar datos permanentes, como configuraciones del sistema y datos de calibración. El contenido de la EEPROM no se borra cuando se apaga el microcontrolador.

Características de la EEPROM del PIC 16F628A

- 128 bytes de EEPROM

5. Timer (Temporizador)

Un temporizador es un circuito electrónico que cuenta eventos o mide intervalos de tiempo. Los temporizadores se utilizan en una amplia variedad de aplicaciones, como generar señales de temporización, controlar motores y medir el tiempo.

El PIC 16F628A tiene tres temporizadores

- **Timer0:** Un temporizador de 8 bits que puede funcionar en varios modos, como modo contador, modo temporizador y modo generador de PWM.
- **Timer1:** Un temporizador de 16 bits que puede funcionar en varios modos, como modo contador, modo temporizador y modo captura/comparación.
- **Timer2:** Un temporizador de 8 bits que se utiliza para generar señales de temporización para el módulo USART.

6. UART (Universal Asynchronous Receiver Transmitter)

El UART es un módulo de comunicación serie que permite al microcontrolador enviar y recibir datos a otros dispositivos. El UART se utiliza en una amplia variedad de aplicaciones, como la comunicación con computadoras, sensores y otros microcontroladores.

Características del UART del PIC 16F628A

- Tasa de baudios configurable de 50 bps a 921.6 kpbs
- Soporte para bits de parada, paridad y control de flujo

7. Voltajes y corriente de trabajo

- **Voltaje de alimentación:** El PIC 16F628A funciona con un voltaje de alimentación de 2.0 V a 5.5 V.

- **Corriente de alimentación:** La corriente de alimentación del PIC 16F628A depende de la velocidad de reloj y de la cantidad de pines que estén en uso. La corriente típica de alimentación es de 12 μ A a 32 kHz y de 120 μ A a 1 MHz.

8. Pulldown

Una resistencia pulldown es una resistencia conectada a tierra que se utiliza para evitar que un pin de entrada flote en un estado indeterminado.

Desarrollo

Para el desarrollo de la practica se utilizó el código en lenguaje C proporcionado en clase, en el cual se hace la selección de puertos y asignación de entradas del PIC para poder interactuar con este mediante la lógica del código, este código es el siguiente:

```
/*
 * File:    LecturaPuertos.c
 * Author:  Ismael Cervantes de Anda
 * Código para configurar los puertos de un PIC16F628A
 * Created on 20 de mayo de 2022, 09:27 AM
 */
// PIC16F628A Configuration Bit Settings

// 'C' source line config statements

// CONFIG
#pragma config FOSC = INTOSCIO // Oscillator Selection bits (INTOSC oscillator:
I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)
#pragma config WDTE = OFF      // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = ON      // Power-up Timer Enable bit (PWRT enabled)
#pragma config MCLRE = OFF     // RA5/MCLR/VPP Pin Function Select bit
(RA5/MCLR/VPP pin function is digital input, MCLR internally tied to VDD)
#pragma config BOREN = OFF     // Brown-out Detect Enable bit (BOD disabled)
#pragma config LVP = OFF      // Low-Voltage Programming Enable bit (RB4/PGM
pin has digital I/O function, HV on MCLR must be used for programming)
#pragma config CPD = OFF      // Data EE Memory Code Protection bit (Data
memory code protection off)
#pragma config CP = OFF       // Flash Program Memory Code Protection bit (Code
protection off)

// #pragma config statements should precede project file includes.
```

```

// Use project enums instead of #define for ON and OFF.

#include <xc.h>
#include <pic16f628a.h>

#define _XTAL_FREQ 4000000

#define Ent0 PORTAbits.RA0
#define Ent1 PORTAbits.RA1
#define Ent2 PORTAbits.RA2
#define Ent3 PORTAbits.RA3
#define Ent4 PORTAbits.RA4
#define Ent5 PORTAbits.RA5
#define Ent6 PORTAbits.RA6
#define Ent7 PORTAbits.RA7

#define Sal0 PORTBbits.RB0
#define Sal1 PORTBbits.RB1
#define Sal2 PORTBbits.RB2
#define Sal3 PORTBbits.RB3
#define Sal4 PORTBbits.RB4
#define Sal5 PORTBbits.RB5
#define Sal6 PORTBbits.RB6
#define Sal7 PORTBbits.RB7

#define GIE INTCONbits.GIE
#define TOIE INTCONbits.T0IE
#define TOIF INTCONbits.T0IF
#define PEIE INTCONbits.PEIE

char    var7seg;
char    continter;

void __interrupt() VectorInterrupcion(void)           // Funciones de
interrupción.
{
    if (EEIF == 1) {EEIF = 0;}                       //interrupción por fin de
escritura en EEPROM
}

void ConfigPIC(void)
{
    TRISA = 0B11111111;                               //Puerto A como entradas
    TRISB = 0x00;                                       //Puerto B como salidas
}

```



```
CMCON = 0x07; //Terminales del puerto A como entradas digitales
PORTA = 0;
PORTB = 0;
}

void main(void)
{
    ConfigPIC();

    while (1)
    {
        if (Ent3 == 0) { Sal2 = 0;}
        if (Ent3 == 1) { Sal2 = 1;}
    }
    return;
}
```

El circuito que se muestra en la Figura 2 es un circuito sencillo utilizando un PIC16F628A como componente central. A continuación, se explica su armado y funcionamiento:

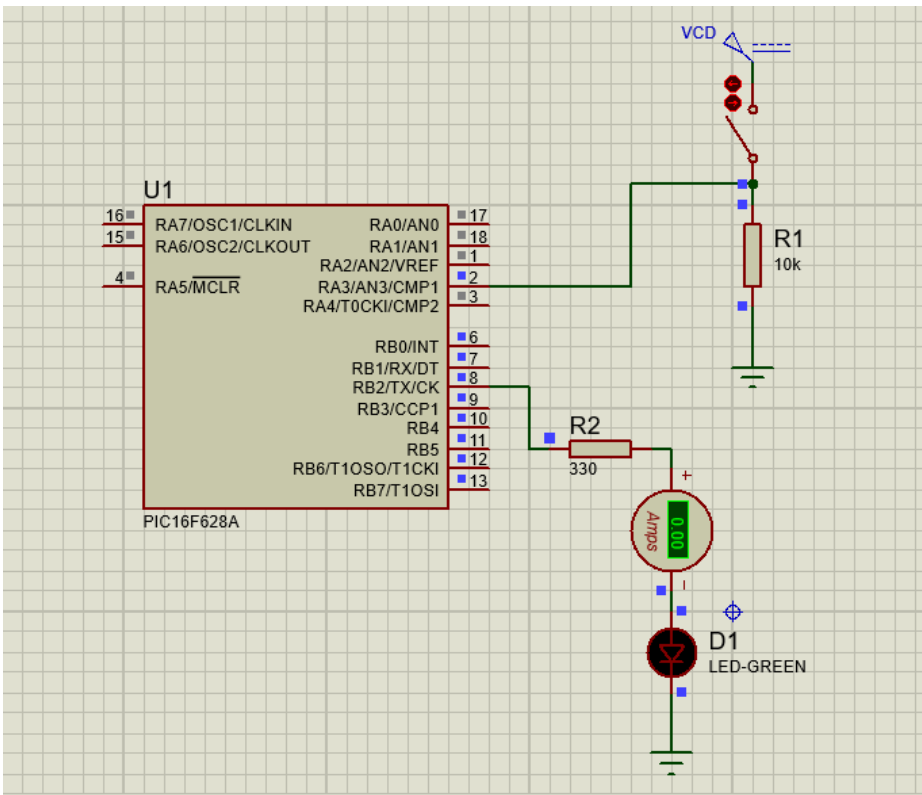


Figura 2. Estado 0 lógico

Componentes y Conexiones

1. Microcontrolador PIC16F628A: Es el corazón del circuito y se encarga de controlar las operaciones. Sus pines están configurados de la siguiente manera:

- Pines de alimentación:

- Vdd (Pin 14): Se conecta a +5V.
- Vss (Pin 5): Se conecta a tierra (GND).

- Pines de entrada/salida:

- PORTA (Pines RA3 - RA7): Configurados como entradas ("1").
- PORTB (Pines RB0 - RB7): Configurados como salidas ("0").

2. Resistencia de Pull-up:

- Resistencia de 10 k Ω conectada entre +5V y el pin RA3. Esto asegura que el pin RA3 esté en un estado alto por defecto.

3. LED y Resistencia de Limitación de Corriente:

- LED: Se conecta con su ánodo al pin RB2 a través de una resistencia.
- Resistencia de 220 Ω a 470 Ω : Se utiliza para limitar la corriente que pasa por el LED, protegiendo así tanto el LED como el microcontrolador.

Armado del Circuito

1. Alimentación:

- Conectar el pin Vdd del PIC16F628A a una fuente de +5V.
- Conectar el pin Vss del PIC16F628A a tierra.

2. Configuración de Pines:

- Conectar una resistencia de 10 k Ω entre el pin RA3 y +5V.
- Configurar los pines RA3 a RA7 como entradas (TRISA = 0b11111111).
- Configurar los pines RB0 a RB7 como salidas (TRISB = 0b00000000).

3. Conexión del LED:

- Conectar una resistencia de 220Ω a 470Ω entre el pin RB2 y el ánodo del LED.
- Conectar el cátodo del LED a tierra.

Funcionamiento del Circuito

1. Estado Inicial:

- Cuando el circuito se alimenta, el pin RA3 se mantiene en un estado alto debido a la resistencia de pull-up.

2. Detección de Entrada:

- El microcontrolador puede detectar cambios en el estado de RA3. Si se pulsa un interruptor conectado entre RA3 y tierra, RA3 pasará a un estado bajo.

3. Control del LED:

- Dependiendo del estado del pin RA3, el microcontrolador puede encender o apagar el LED conectado al pin RB2. Por ejemplo, si RA3 está en alto, el programa puede hacer que RB2 esté en bajo, apagando el LED. Si RA3 está en bajo, RB2 puede estar en alto, encendiendo el LED.

4. Programa de Control:

- El programa cargado en el PIC16F628A debe leer el estado del pin RA3 y, en función de ese estado, cambiar el estado del pin RB2 para controlar el LED.

Simulaciones

1. Estado 0 lógico

La Figura 2 también es la representación del estado '0' lógico, pues se muestra la resistencia en configuración pull-up con el switch abierto.

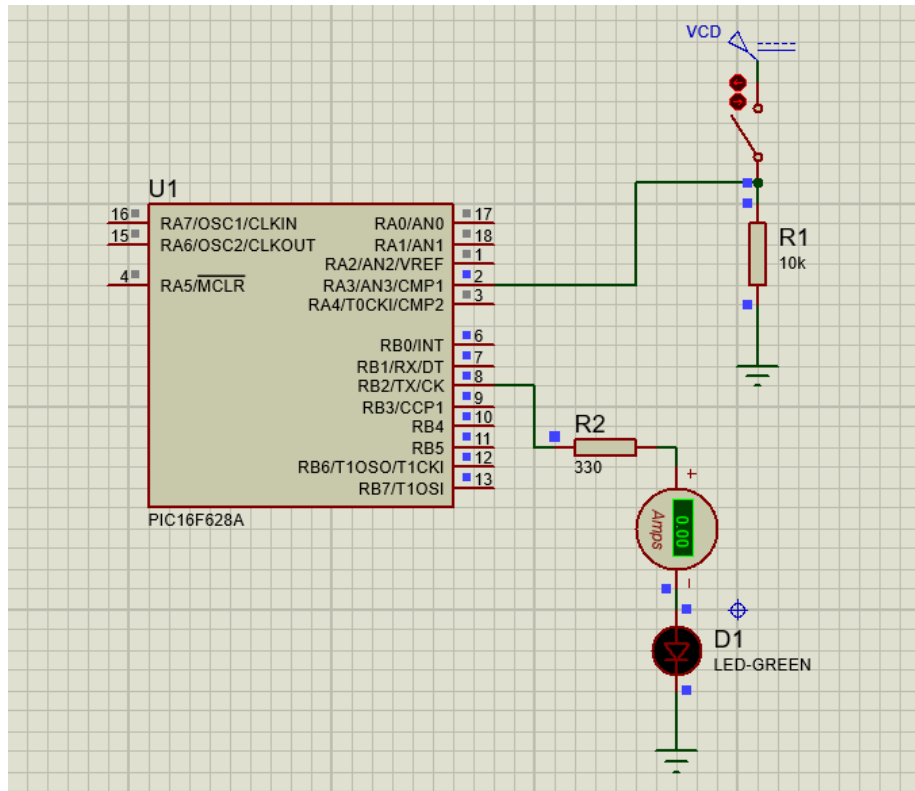


Figura 2. Estado 0 lógico

2. Estado 1 lógico

En la Figura 3 se muestra el '1' lógico al estar el switch cerrado en la resistencia pull-up.



Conclusión

El circuito divisor de voltaje presentado demuestra la funcionalidad de este circuito electrónico y sirve como una plataforma educativa valiosa para comprender la interacción entre el microcontrolador PIC16F628A y el exterior. Al combinar la simplicidad del divisor de voltaje con las capacidades de programación del PIC, este ejercicio proporciona una introducción práctica a los conceptos fundamentales de la electrónica digital y los sistemas embebidos.

No solo adquirimos conocimientos sobre la configuración de entradas y salidas del microcontrolador, sino que también profundizan en la comprensión de:

- **Manejo de periféricos externos:** El control mediante el PIC16F628A ilustra la interacción del microcontrolador con dispositivos externos, sentando las bases para el control de sensores, actuadores y otros componentes electrónicos.
- **Configuración de pines:** La selección y configuración adecuadas de los pines del microcontrolador son cruciales para la correcta operación del circuito. Este proceso refuerza la comprensión de la función de cada pin y su papel en la comunicación con el mundo exterior.
- **Programación básica:** La programación del PIC16F628A para controlar demuestra la aplicación práctica de los conceptos de programación en el contexto del control de hardware. Los estudiantes aprenden a traducir instrucciones de alto nivel en código máquina que el microcontrolador puede ejecutar.

Bibliografia

Microchip Technology PIC16F628A Datasheet:

<https://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>

"PIC16F628A Assembly Language Programming" por Newbie Circuit:

https://www.youtube.com/watch?v=srp0_9OGXdl

"PIC16F628A Programming Guide" por Elektor Labs:

<https://www.elektormagazine.com/news/new-book-pic-microcontroller-programming>