

MikroElektronika books

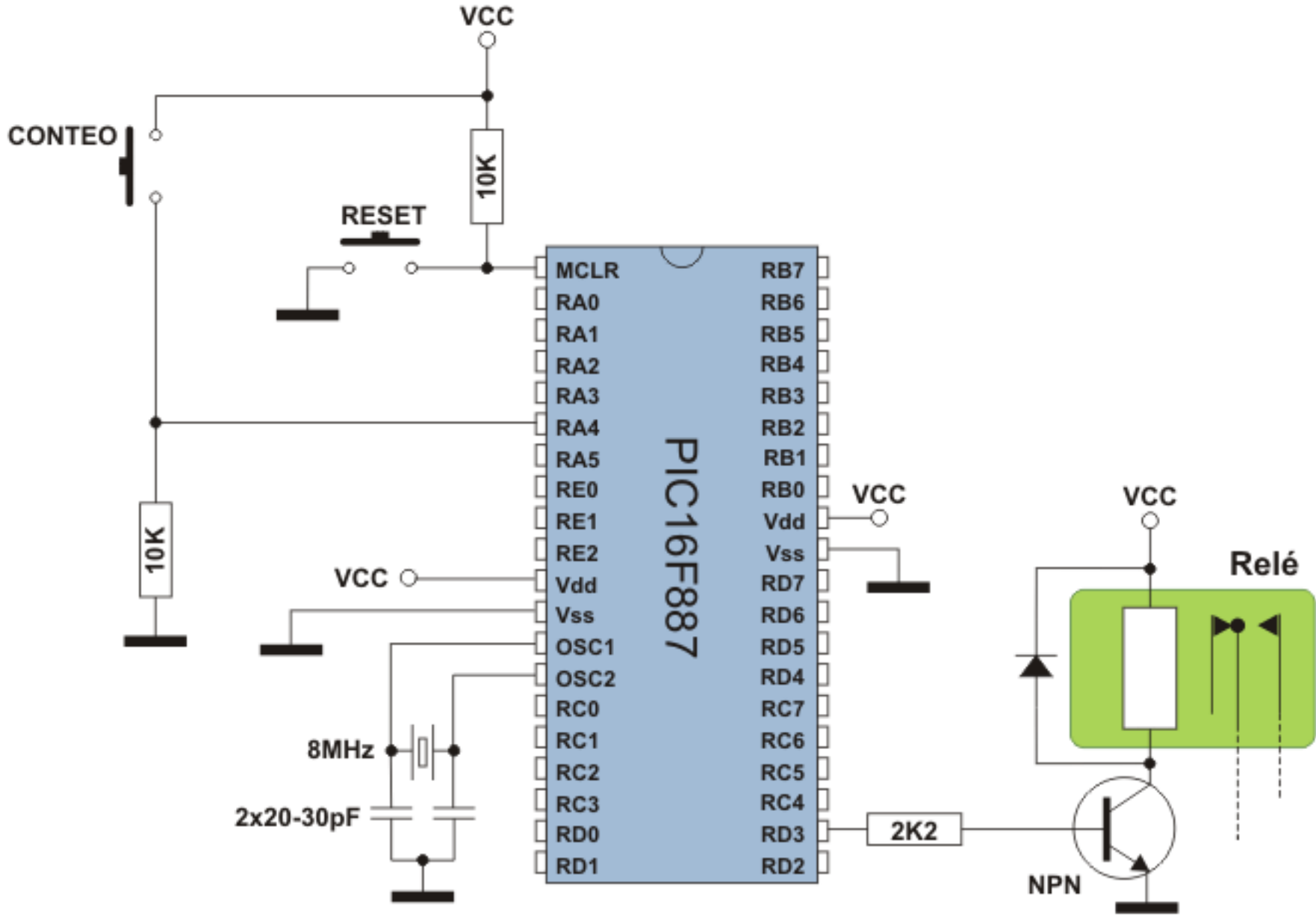
Book: [Microcontroladores PIC – Programación en C con ejemplos](#)

☰
Table of Contents

4.5 Ejemplo 3

Timer0 como un contador, declarar variables nuevas, constantes de enumeración, utilizar relés...

En cuanto a los ejemplos anteriores, el microcontrolador ha ejecutado el programa sin haber sido afectado de ninguna forma por su entorno. En la práctica, hay pocos dispositivos que funcionen de esta manera (por ejemplo, un simple controlador de luz de neón). Los pines de entrada se utilizan también en este ejemplo. En la siguiente figura se muestra un esquema, mientras que el programa está en la siguiente página. Todo sigue siendo muy simple. El temporizador Timer0 se utiliza como un contador. La entrada del contador está conectada a un botón de presión, así que cada vez que se presiona el botón, el temporizador Timer0 cuenta un pulso. Cuando el número de pulsos coincida con el número almacenado en el registro TEST, un uno lógico (5V) aparecerá en el pin PORTD.3. Este voltaje activa un relé electromecánico, y por eso este bit se le denomina ‘RELÉ’ en el programa.



```
/*Cabecera******/

void main() {
    char TEST = 5;           // Constante TEST = 5
    enum salidas {RELÉ = 3}; // Constante RELAY = 3

    ANSEL = 0;               // Todos los pines de E/S se configuran como digitales
    ANSELH = 0;
    PORTA = 0;               // Reiniciar el puerto PORTA
    TRISA = 0xFF;            // Todos los pines del puerto PORTA se configuran como entradas
    PORTD = 0;               // Reiniciar el puerto PORTD
    TRISD = 0b11110111;      // Pin RD3 se configura como salida, mientras que los demás

    // se configuran como entradas
    OPTION_REG.F5 = 1;       // Contador TMR0 recibe los pulsos por el pin RA4
    OPTION_REG.F3 = 1;       // Valor del pre-escalador 1:1

    TMR0 = 0;               // Reiniciar el temporizador/contador TMR0

    do {
        if (TMR0 == TEST)    // ¿Coincide el número en el temporizador con la
                               // constante TEST?
        {
            (PORTD.RELAY = 1); // Números coinciden. Poner el bit RD3 a uno (salida RELÉ)
        }
        while (1);           // Quedarse en el bucle infinito
    }
}
```

Sólo una constante de enumeración RELÉ se utiliza en este ejemplo. Se le asigna un valor mediante la declaración.

```
enum salidas {RELÉ = 3}; // Constante RELÉ = 3
```

Si varios pines del puerto PORTD están conectados a los relés, la expresión anterior se puede escribir de la siguiente manera también:

```
enum salidas {RELÉ=3, CALENTADOR, MOTOR=6, SURTIDOR};
```

A todas las constantes, precedidas por las constantes con valores asignados (RELÉ=3 y MOTOR=6), se les asignan automáticamente los valores de las constantes precedentes, incrementados en 1. En este ejemplo, a las constantes CALENTADOR y SURTIDOR se les asignan los valores 4 y 7, es decir (CALENTADOR=4 y SURTIDOR=7), respectivamente.

SUBSCRIBE TO

- 📧 newsletter
- 🐙 github
- in linkedin
- f facebook
- 📺 youtube
- 📷 instagram
- 🐦 twitter

JOIN US Careers Make a Click Internship

COMPANY

- About us
- Contact
- Leadership
- PressKit
- Distributors
- Timeline
- Terms

TOOLCHAINS

- PIC
- dsPIC
- PIC32
- ARM
- AVR
- FT90X
- 8051
- PSOC
- CEC

RESOURCES

- mikroBUS™
- mikroSDK
- Click Cloud
- Premium TS
- Hexiwear™
- Libstock™
- eBooks
- Outlet
- Legacy