**Ejemplo\_000\_BlinkingLed**

Objetivo: Implementar el circuito y código necesario para para hacer parpadear los leds del puerto D ……………….

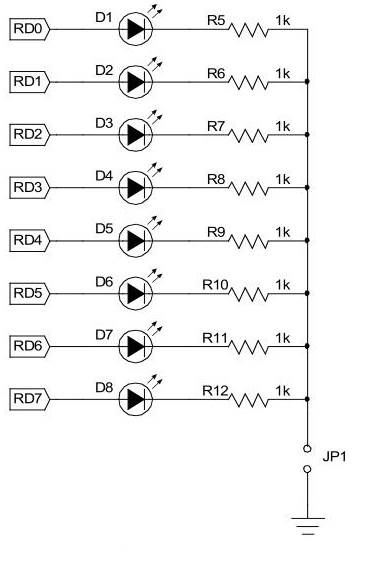
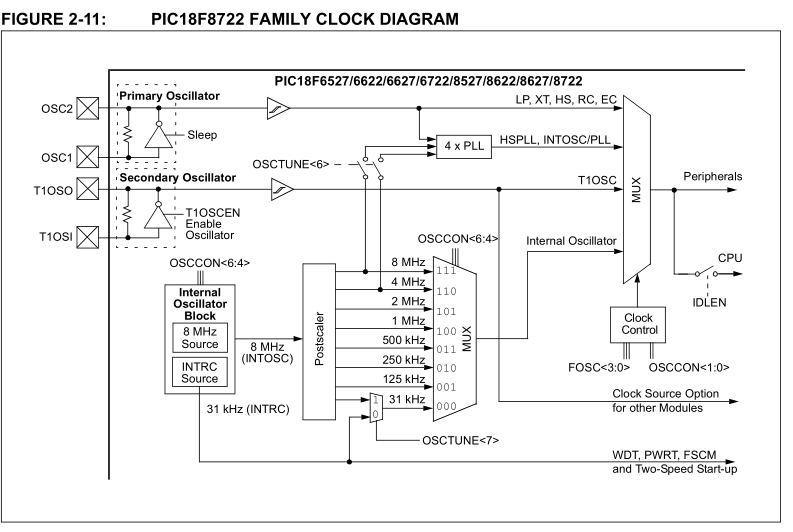


Figura 1. Circuito implementado en la tarjeta PICDEM PIC18

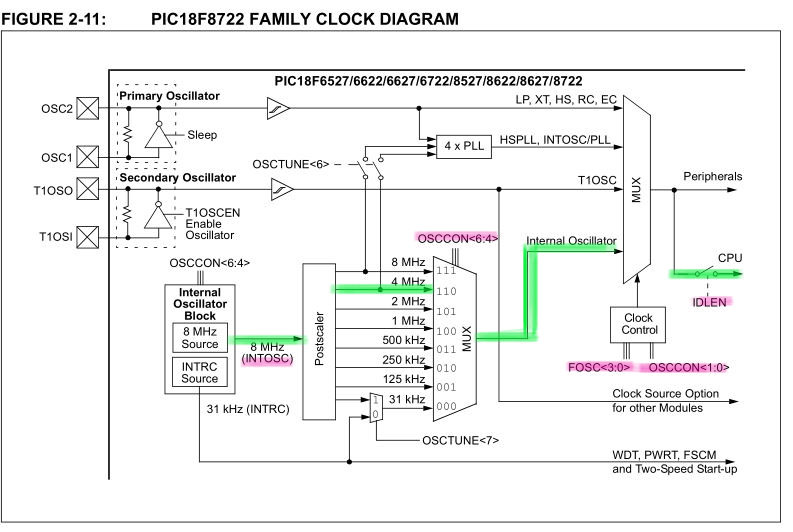
**Configuración del Oscilador**



**Figura 2** Diagrama del oscilador

Para el proyecto **Ejemplo\_000\_BlinkingLed** se seleccionara el oscilador interno a 4Mhz

Para ello se deben configurar los bits correspondientes a **Figura 3.**



**Figura 3**, Configuración para oscilador interno igual a 4MHZ

Registro **OSCCON** : Registro de configuración del oscilador (byte de dirección FD3h )

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BIT7** | **BIT6** | **BIT5** | **BIT4** | **BIT3** | **BIT2** | **BIT1** | **BIT0** |
| IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | IOFS | SCS1 | SCS0 |

**IDLEN**: IDLEN=1, El dispositivo entra en modo IDLE cuando la instrucción SLEEP es ejecutada (La CPU mantiene encendida pero no ejecuta ninguna instrucción);IDLEN=0, El dispositivo entra en modo SLEEP cuando la instrucción SLEEP es ejecutada, En modo SLEEP el dispositivo se apaga y desconecta el reloj de la CPU, en este modo la única forma de despertarlo es con el WDT o con una interrupción externa.

Para el proyecto **Ejemplo\_000\_BlinkingLed** se seleccionara

IDLEN=0;

**IRCF<2:0>**: Selección interna de la frecuencia del oscilador, Si es 8 MHz la conexión es directa desde el oscilador interno al multiplexor de control de reloj, de lo contrario, pasa antes por divisor de frecuencias (prescalar), en el cual 8 frecuencias pueden ser seleccionadas,

Para el proyecto **Ejemplo\_000\_BlinkingLed** se seleccionara

IRCF=0x06;

#define IRC4Mhz 0x60

**OSTS:** Estado del timer oscilador de arranque, **OSCTS**=1, el oscilador primario está listo y corriendo, **OSTS=0**, el oscilador primario no está listo. Este bit es de solo lectura y no se usara para el proyecto **Ejemplo\_000\_BlinkingLed** se seleccionara

**IOFS:** Estado del oscilador interno, IOFS=0, INTOSC no es estable; IOFS=1, INTOSC es estable.(bit de solo lectura)

**SCS<1:0>:** Selección de reloj del sistema, con estos bits se seleccionan tres fuente de oscilador, el primario (conectado a OSC1 y OSC2, el secundario (timer1) conectado a T1OSO y T1OSI, y el bloque oscilador interno.

Para el proyecto **Ejemplo\_000\_BlinkingLed** se seleccionara

SCS=0x02

#define InternalOscillator 0x02

Con lo anterior el valor del registro OSCON es,

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BIT7** | **BIT6** | **BIT5** | **BIT4** | **BIT3** | **BIT2** | **BIT1** | **BIT0** |
| IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | IOFS | SCS1 | SCS0 |
| 0 | 1 | 1 | 0 | x | x | 1 | 0 |

**OSCCON=**IRC4Mhz | InternalOscillator;

Para el proyecto **Ejemplo\_000\_BlinkingLed** se seleccionara no se usara **PLL**  por lo tanto, **OSCTUNE=0x00.**

OSCCON=IRC4Mhz|InternalOscillator;

OSCTUNE = 0x00;

A este punto el MCU, está en la capacidad de ejecutar instrucciones, ahora se configurara el puerto D ( PORTD) para que todos sus pines sean salidas digitales.

Los puertos I/O cuentan con tres registros principales,

**TRIS**=Establece la dirección, entrada (igualar a uno) o salida (igualar a cero),

**LAT=**Escribe cero o uno lógico al puerto.

**PORT**=Para lectura del puerto.

**I/O PORTS**

Los pines del MCU comparter diferentes funciones , por ello algunos puertos requieren configuración extra para ser manipulados como entradas o salidas digitales. Para mayor informacio ver la sección 11 I/O Ports del [datasheet](http://ww1.microchip.com/downloads/en/DeviceDoc/39646c.pdf)

#define LedsOnBoardOn 0xFF

LATD=0;

PORTD=0;

TRISD=0;

TRISD&=~(LedsOnBoardOn);