3.11 Memoria EEPROM

El microcontrolador PIC16F887 dispone de 256 localidades de memoria EEPROM controlados por los bits de los siguientes registros:

- EECON1 (registro de control);
- EECON2 (registro de control);
- EEDAT (almacena los datos listos para escritura y lectura); y
- EEADR (almacena la dirección de la EEPROM a la que se accede).

Además, el registro EECON2 no es un registro verdadero, no existe físicamente en el chip. Se utiliza sólo durante la escritura de los datos en la memoria. Los registros EEDATH y EEADRH se utilizan durante la escritura y lectura de la EEPROM. Los dos se utilizan también durante la escritura y lectura de la memoria de programa (FLASH). Por considerar esto una zona de riesgo (por supuesto usted no quiere que el microcontrolador borre su propio programa por casualidad), no vamos a discutirlo aquí, no obstante le avisamos que tenga cuidado.

Registro EECON1

	R/W (x)				R/W (x)	R/W (0)	R/S (0)	R/S (0)	Características
EECON1	EEPGD	-	-	-	WRERR	WREN	WR	RD	Nombre de bit
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Leyenda

-	Bit no implementado
R/W	Bit de lectura/escritura
R	Bit de lectura
S	Bit se puede poner sólo a uno
(0)	Después del reinicio, el bit se pone a cero
(x)	Después del reinicio, el estado de bit es desconocido

EEPGD - Program/Data EEPROM Select bit (bit de selección de memorias)

- 1 - Acceso a la memoria Flash de programa.
- 0 - Acceso a la memoria de datos EEPROM.

WRERR - EEPROM Error Flag bit (bit de error de escritura)

- 1 - Se produce un error de escritura de forma prematura y ha ocurrido un error.
- 0 - Se ha completado la operación de escritura.

WREN - EEPROM Write Enable bit (bit de habilitación de escritura)

- 1 - Escritura de datos en la EEPROM habilitada.
- 0 - Escritura de datos en la EEPROM deshabilitada.

WR - Write Control bit (bit de control de escritura)

- 1 - Se ha iniciado una operación de escritura de datos en la EEPROM.
- 0 - Se ha completado una operación de escritura de datos en la EEPROM.

RD - Read Control bit (bit de control de lectura)

- 1 - Inicia una lectura de la memoria EEPROM.
- 0 - Lectura de la memoria EEPROM deshabilitada.

LECTURA DE LA MEMORIA EEPROM

Para leer los datos de la memoria EEMPROM, siga los siguientes pasos:

- **Paso 1**: Escribir la dirección (00h - FFh) en el registro EEADR.
- **Paso 2**: Seleccionar el bloque de memoria EEPROM al poner a cero el bit EEPGD del registro EECON1.
- **Paso 3**: Poner a uno el bit RD del mismo registro para leer el contenido de la localidad.
- **Paso 4**: El dato se almacena en el registro EEDAT y está listo para su uso.

El siguiente ejemplo muestra el procedimiento anteriormente descrito al escribir un programa en lenguaje ensamblador:

```
BSF STATUS,RP1   ;
BCF STATUS,RP0   ; Acceder al banco 2
MOVF ADDRESS,W   ; Mover la dirección al registro W
MOVWF EEADR      ; Escribir la dirección
BSF STATUS,RP0   ; Acceder al banco 3
BCF EECON1,EEPGD ; Seleccionar la EEPROM
BSF EECON1,RD    ; Leer los datos
BCF STATUS,RP0   ; Acceder al banco 2
MOVF EEDATA,W    ; Dato se almacena en el registro W
```

La misma secuencia de programa escrita en C se parece a lo siguiente:

```
W = EEPROM_Read(ADDRESS);
```

Las ventajas del uso del lenguaje C se han hecho más obvias, no lo cree?

ESCRITURA EN LA MEMORIA EEPROM

Antes de escribir los datos en la memoria EEPROM es necesario escribir la dirección en el registro EESADR y los datos en el registro EESAT. Sólo ha quedado seguir a una secuencia especial para iniciar la escritura para cada byte. Durante el proceso de escritura las interrupciones deben estar deshabilitadas. El ejemplo que sigue muestra el procedimiento anteriormente descrito al escribir un programa en lenguaje ensamblador:

```
BSF STATUS,RP1
BSF STATUS,RP0
BTFSC EECON,WR1  ; Esperar a que se complete la escritura anterior
GOTO $-1 ;
BCF STATUS,RP0   ; Banco 2
MOVF ADDRESS,W   ; Mover la dirección a W
MOVWF EEADR      ; Escribir la dirección
MOVF DATA,W      ; Mover los datos a W

MOVWF EEDATA     ; Escribir los datos
BSF STATUS,RP0   ; Banco 3
BCF EECON1,EEPGD ; Seleccionar la EEPROM
BSF EECON1,WREN  ; Escritura a la EEPROM habilitada
BCF INCON,GIE    ; Todas las interrupciones deshabilitadas

MOVLW 55h
MOVWF EECON2
MOVLW AAh
MOVWF EECON2
BSF EECON1,WR

BSF INTCON,GIE   ; Interrupciones habilitadas
BCF EECON1,WREN  ; Escritura a la EEPROM deshabilitada
```

La misma secuencia de programa escrita en C se parece a lo siguiente:

```
W = EEPROM_Write(ADDRESS, W);
```

No hace falta comentar nada. **Vamos a hacerlo en mikroC...**

```
// El ejemplo muestra cómo utilizar la librería EEPROM en el compilador mikroC PRO for PIC.

char ii; // La variable ii utilizada en el bucle

void main(){
  ANSEL = 0;                    // Configuración de los pines AN como E/S digitales
  ANSELH = 0;
  PORTB = 0;
  PORTC = 0;
  PORTD = 0;
  TRISB = 0;
  TRISC = 0;
  TRISD = 0;
  for(ii = 0; ii < 32; ii++)      // Llenar el búfer con los datos
    EEPROM_Write(0x80+ii, ii);    // Escribir los datos en la dirección 0x80+ii

  EEPROM_Write(0x02,0xAA);        // Escribir un dato en la dirección 2 de la EEMPROM
  EEPROM_Write(0x50,0x55);        // Escribir un dato en la dirección 0x50

  // de la EEMPROM
  Delay_ms(1000);                 // Diodos en los puertos PORTB y PORTC
  PORTB = 0xFF;                   // para indicar el comienzo de la lectura
  PORTC = 0xFF;
  Delay_ms(1000);
  PORTB = 0x00;
  PORTC = 0x00;
  Delay_ms(1000);
  PORTB = EEPROM_Read(0x02);      // Leer los datos de la dirección 2 de la EEPROM y

  // visualizarla en el puerto PORB
  PORTC = EEPROM_Read(0x50);      // Leer los datos de la dirección 0x50 de la EEPROM y

  // visualizarla en el puerto PORC
  Delay_ms(1000);

  for(ii = 0; ii < 32; ii++) {    // Leer el bloque de 32 bytes de la dirección
    PORTD = EEPROM_Read(0x80+ii); // 0x80 y visualizarla en el puerto PORTD

  Delay_ms(250);
  }
}
```

Aprimera vista, basta con encender una fuente de alimentación para hacer funcionar un microcontrolador. A primera vista, basta con apagar una fuente de alimentación para detenerlo. Sólo a primera vista. En realidad, el arranque y el final del funcionamiento son las fases críticas de las que se encarga una señal especial denominada RESET.