



Comentario técnico: CTC-011

Título: **Migración de PIC 16F62x a 16F62xA**

Autor: Sergio R. Caprile, Senior Engineer

Revisiones	Fecha	Comentarios
0	23/10/03	

Comentamos acerca de cómo migrar aplicaciones basadas en PIC16F62x a PIC16F62xA; incluimos diferencias de software, hardware y soporte de programación.

Hardware

El PIC 16F62xA es compatible pin a pin con los 16F62x, no debería ser necesario realizar modificaciones en el hardware que hoy usa 16F62x para que funcione con 16F62xA. No obstante, incorpora una serie de características diferentes que deberán ser tenidas en cuenta, dado que algunas opciones no son directamente equivalentes.

Las nuevas características y diferencias son:

- Oscilador ER pasa a RC: deberá agregarse un capacitor externo si usa este oscilador.
- El modo dual speed (que afectaba al oscilador interno y el ER) solo afecta al oscilador interno
- El oscilador interno está calibrado 4MHz 1%
- La memoria flash cambia su modo de protección, si se habilita se protege toda la memoria.
- El oscilador del Timer1 funciona hasta 32,768 KHz y no hasta 200KHz como en el 16F62x
- Resuelve el problema del comparador descrito en la errata del 16F62x

Programación

Al momento de escribir este documento, el PIC 16F62xA no es soportado por la última revisión de firmware conocida del PICStart Plus. Sí es soportado por el MPLAB ICD2 versión 1.31 y ProMate versión 6.2.

Diferencias eléctricas

Las especificaciones eléctricas resultan en la mayoría de los casos funcionalmente equivalentes. Se recomienda la lectura de las correspondientes hojas de datos para detalles más específicos.

Errata

Existen erratas documentadas, se sugiere la lectura de la errata correspondiente a este dispositivo. Podemos adelantar que la revisión A tiene un defecto en la escritura de la EEPROM, que puede ocasionar problemas de ejecución de código. Según recomienda Microchip, puede evitarse el efecto ejecutando una instrucción *SLEEP* inmediatamente después de setear el bit *EECON1 WR*, y permitir que el procesador salga del modo sleep por la interrupción de finalización de escritura. Esto se realiza seteando los bits *PEIE* en el registro *INTCON* y *EEIE* en el registro *PIE1*. Las demás interrupciones deberán estar inhabilitadas, de modo que solamente la interrupción de finalización de escritura en EEPROM sea la que pueda despertar al procesador. Se deberá tener en cuenta que muchos periféricos suspenden su operación durante el modo sleep, por lo que deberán tomarse las precauciones correspondientes.

El siguiente es un fragmento de código recomendado por Microchip para evitar este posible inconveniente, no obstante, el mismo no chequea el estado previo de las demás interrupciones o periféricos:

```
BANKSEL 0X00          ;select Bank0
BCF PIR1, EEIF       ;ensure write complete
                     ;flag is clear
BANKSEL 0x80         ;change to Bank1
```

CTC-011, Migración de PIC 16F62x a 16F62xA

```
MOVLW 1 << PEIE      ;enable only
                      ;peripheral interrupt
MOVWF INTCON         ;
MOVLW 1 << EEIE      ;enable only EE write
                      ;complete interrupt
MOVWF PIE1           ;
BSF EECON1, WREN     ;enable EE write
MOVLW 0x55           ;required write
                      ;protect squence
MOVWF EECON2         ;
MOVLW 0xAA           ;second part of
                      ;sequence
MOVWF EECON2         ;
BSF EECON1, WR       ;initiate write
SLEEP                ;suspend operation
                      ;during write
BCF EECON1,WREN      ;disable EE write
                      ;program execution
                      ;resumes with this
                      ;instruction upon EE
                      ;write completion
```