

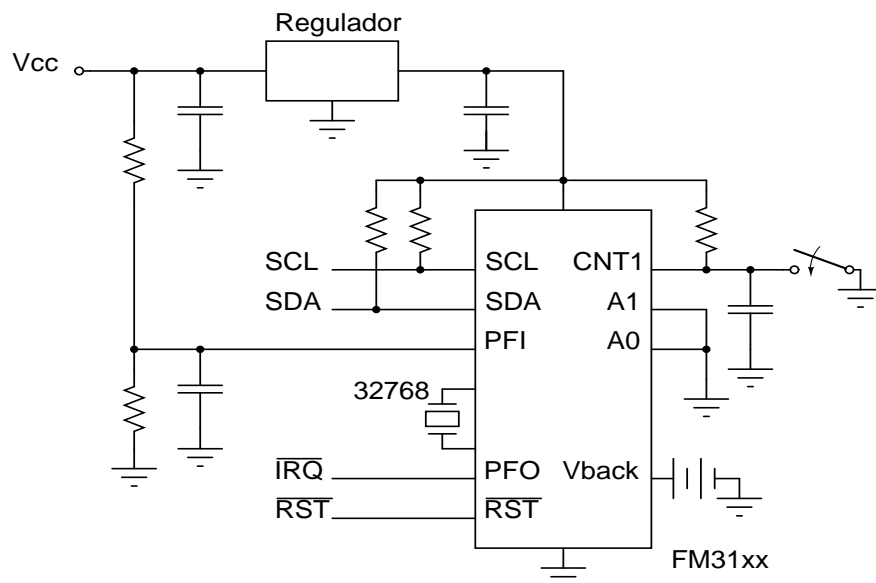
Revisiones	Fecha	Comentarios
0	08/02/07	

En el comentario técnico CTC-045 vimos la forma de utilizar la interfaz I²C en los Ramtron VRS51L3074. En esta oportunidad, accederemos al RTC de un Processor Companion FM31xx utilizando dichas rutinas. El código fue tomado de lo provisto por el fabricante, con modificaciones menores. Para compilarlo es necesario disponer del código que acompaña a CTC-045

Hardware

En el kit de desarrollo del VRS51L3074 disponemos de un FM31256 y una FM24CL64, con un pequeño bus I²C. todo lo que tenemos que hacer es conectarlos a los pines SCL y SDA del micro. El FM31256 no tiene alimentación de backup, la cual deberá proveerse por un conector adicional. el circuito esquemático figura en el manual del usuario del kit.

Para aquellos lectores que no disponen de un kit y desean experimentar, el siguiente es un circuito típico de aplicación del Processor Companion:



Software

Escritura de un registro del Processor Companion

El primer parámetro es el ID del Processor Companion, el siguiente el número de registro y el último el valor a escribir:

```
void RTCWrite(unsigned char idw,unsigned char address,unsigned char value)
{
    I2CRXTX = idw;                               // START + ID + W

    WaitTXEMP();
    I2CRXTX = address;                           // registro
    WaitTXEMP();
}
```

CAN-062, Uso del RTC de un Processor Companion FM31xx con Ramtron VRS51L3074

```
I2CRXTX = value; // dato
WaitI2CIDLE(); // STOP
}
```

Lectura de un registro del Processor Companion

```
unsigned char RTCRead(unsigned char idw, unsigned char address)
{
    I2CRXTX = idw; // START + ID + W

    WaitTXEMP();
    I2CRXTX = address; // registro

    WaitI2CIDLE(); // STOP

    I2CRXTX = idw+1; // START + ID + R
    WaitTXEMP();
    I2CCONFIG |= 0x02; // No Ack en byte siguiente (sólo uno)
    WaitRXAV(); // Espera byte
    I2CCONFIG &= ~0x02; // Ack (para la siguiente operación)
    return I2CRXTX; // Devuelve dato
}
```

Inicialización del RTC

El valor a escribir en el registro Companion Control dependerá del hardware, dado que el trickle charger deberá activarse sólo para respaldo con capacitor (bit VBC).

```
void Init_RTC (void){
    // Capacitor : Activa el trickle charger (setea VBC en registro 0x0B)
    //RTCWrite(0xD0,0x0B,0x04);

    // Pila de litio: Desconecta el trickle charger (resetea VBC en registro 0x0B)
    //RTCWrite(0xD0,0x0B,0x00);
}
```

Lectura de fecha y hora

Antes de poder leer el valor del RTC, debemos setear el bit R en el registro RTC Control, lo cual produce la transferencia del valor del RTC a los registros que serán leídos. Luego de leer el RTC, procederemos a resetear este bit, para que pueda volver a ser seteado la próxima vez que vayamos a leer el RTC.

Los datos de fecha y hora se entregan en el formato del RTC (BCD), en el array apuntado por el segundo parámetro. El primer parámetro es el ID del Processor Companion.

```
void RTCReadAll(unsigned char idw, unsigned char *rtcvalue){
register unsigned char loop=6;

    I2CRXTX = idw; // START + ID + W
    WaitTXEMP();
    I2CRXTX = 0x00; // registro RTC Control

    WaitTXEMP();

    I2CRXTX = 0x01; // RTC Control = 0x01 (R bit = 1)

    WaitI2CIDLE(); // STOP

    I2CRXTX = (idw+1); // START + ID + R
    WaitTXEMP();
    WaitRXAV(); // lee CAL/CTRL (ignora)
    *rtcvalue = I2CRXTX;

    while(loop--){
        WaitRXAV(); // lee registro siguiente
        *(rtcvalue++) = I2CRXTX; // guarda en array
    }
}
```

CAN-062, Uso del RTC de un Processor Companion FM31xx con Ramtron VRS51L3074

```
I2CONFIG |= 0x02; // no ACK último registro (STOP al final)

WaitRXAV(); // lee
*(rtcvalue++) = I2CRXTX; // guarda
I2CONFIG &= ~0x02; // ACK (siguiente operación)

I2CRXTX = idw; // START + ID + W

WaitTXEMP();
I2CRXTX = 0x00; // registro RTC Control

WaitTXEMP();
I2CRXTX = 0x00; // RTC Control = 0x00 (R bit = 0)

WaitI2CIDLE(); // STOP

}
```

Puesta en hora

Para actualizar el RTC, deberemos setear el bit W en el registro de control, y luego escribiremos todos los registros relacionados con el mismo. Luego, reseteamos el bit W. La función es similar a la de lectura, le pasamos el ID del Processor Companion y un array con los valores a escribir

```
void RTCSet(unsigned char idw, unsigned char *rtcparam)
{
register unsigned char loop=7;

    I2CRXTX = idw; // START + ID + W

    WaitTXEMP();
    I2CRXTX = 0x00; // registro RTC Control
    WaitTXEMP();
    I2CRXTX = 0x02; // Setea bit W

    WaitTXEMP();
    I2CRXTX = 0x00; // Resetea OSCEN en registro CAL/Control
    // CAL no afecta, arranca el oscilador

    while(loop--){
        WaitTXEMP(); // registro siguiente
        I2CRXTX = *(rtcparam++); // escribe datos
    }

    WaitI2CIDLE(); // STOP

    I2CRXTX = idw; // START + ID + W

    WaitTXEMP();
    I2CRXTX = 0x00; // registro RTC Control
    WaitTXEMP();
    I2CRXTX = 0x00; // Resetea bit W

    WaitI2CIDLE(); // STOP
}
```

Ejemplo de uso

El siguiente es un ejemplo muy simple de inicialización y uso del RTC:

```
void main (void){

unsigned char rtcval[7];
const static __code unsigned char rtcstart[]={0,0,0,0,0,0,0}; // simula fecha y hora

    Init_I2C();
    Init_RTC();

    RTCSet(0xD0, rtcstart); // ponemos el RTC en hora
    RTCWrite(0xD0,0x01,0x00); // si no lo ponemos en hora y no hay Vbackup,
    // debemos arrancar el oscilador en forma manual

    while(1)
        RTCReadAll(0xD0,rtcval); // Lee el RTC
}
```