INTRODUCTION

This application note is designed to take the design engineer through the steps to configure and set up the Microchip I2C™ MCP794XX RTCC family. Topics covered include setting the time and date, enabling external battery backup, enabling the CLKOUT function and configuring the alarm.

The information presented in this document is designed to be an example of possible configurations. The code supplied can be modified to change device functionality.

This application note should be read in conjunction with both the “Recommended Usage of Microchip Serial RTCC Devices” (AN1365), and the device data sheet (DS22266). The latest documentation can be found on the microchip web site: http://www.microchip.com/rtcc

MCP794XX FAMILY

The MCP794XX family of devices are stand-alone Real-Time Clocks. They operate independently of the host MCU once configured. The basic features of the RTCC are:

• Time and Date Registers
• Two Programmable Alarms
• Clock Out Feature
• EEPROM Memory
• 64 Bytes of Battery-Backed SRAM
• Battery Backup Pin

The minimum configuration required to correctly use the RTCC is:

• Correct crystal and matching load capacitors
• VCC within data sheet specification
• Required pull-up resistors
• Host MCU with functioning I2C master code

The following will be covered in this document. The required registers will be covered along with sample code and flowcharts where appropriate:

• Basic Time Configuration – including starting the oscillator and enabling battery backup
• MFP configuration
• Alarm configuration

SCHEMATIC DIAGRAM

The schematic shown in Figure 1 shows the minimum components required to operate the RTCC. The device requires pull-up resistors for the I2C data lines (see AN1028, “Recommended Usage of Microchip I2C™ Serial EEPROM Devices” for more information). Additionally, the MFP pin is an open-drain configuration, so if this functionality is going to be used, an additional pull-up is required.

The schematic also shows the required components for battery backup operation using a lithium coin cell (for other options refer to AN1365, “Recommended Usage of Microchip Serial RTCC Devices”). If the VBAT input and battery backup feature is not required, this pin should be tied to GND.

FIGURE 1: SCHEMATIC DIAGRAM
PCB Layout Considerations

The PCB layout is especially important for this device. Extra care should be taken around the oscillator pins on the device and the crystal. Please refer to AN1365, “Recommended Usage of Microchip Serial RTCC Devices”.

The best way to get started with the MCP794XX is to use the RTCC PICtail™ Plus daughter board module (AC164140). The module implements all the hardware suggestions in this document.

Crystal Selection Considerations

For any low-power oscillator-based system, including the RTCC devices, the crystal selection is critical for correct operation. The crystal must be matched with the load capacitors (please refer to the product data sheet for suggested crystal and load capacitors).

SOFTWARE OVERVIEW

The host MCU will need to communicate with the RTCC using the I²C protocol. This can either be done using the hardware module in the MCU or by bit-banging using discrete IO pins. Microchip provides application notes to support both hardware and software methods.

The code presented with this application note is designed to compile with the XC16 compiler for the following hardware:
• Explorer 16 Evaluation Board (DM240001)
• dsPIC33FJ256GP710A PIM Module (supplied with DM240001)
• RTCC I²C PICtail Plus Daughter Board (AC164140)

The code is presented in C and is portable with minimal effort to other PIC MCU devices. This code is designed to be a starting point for application development and is based around the I²C drivers in AN1079, “Using the C30 Compiler and the I²C™ Peripheral to Interface EEPROMs with the dsPIC33F”.

The application notes listed in Appendix B all support the same low-level commands, so the code can easily be ported to other PIC MCU platforms.

Basic Time Configuration

To enable the RTCC to count and maintain the time and date, the Time and Date registers must be loaded correctly and the oscillator started.

If the time and date is required to be maintained during the time when VCC is not available, the external VBAT pin can be supplied with a backup power source. To enable battery backup operation, an additional bit (VBATEN – VBAT Enable) needs to be set when loading the Time and Date registers.

The registers required for basic time configuration are shown in the memory map in the product data sheet. Please refer to the device data sheet (DS22266) for register definitions. All time and date is stored as Binary Coded Decimal (BCD), examples of BCD are shown in Table 1:

<table>
<thead>
<tr>
<th>Decimal Digit</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0x01</td>
</tr>
<tr>
<td>10</td>
<td>0x10</td>
</tr>
<tr>
<td>12</td>
<td>0x12</td>
</tr>
<tr>
<td>23</td>
<td>0x23</td>
</tr>
</tbody>
</table>

Functions to convert between BCD and HEX are provided in Appendix A.

It should be noted that the RTCC device does not perform any error checking on the values loaded into the registers. Care should be taken to ensure that the ranges detailed in the data sheet are respected.

Additional bits located in the time/date registers used as part of the time and date configuration are discussed below:

To load the Time and Date registers, the RTCC oscillator should be stopped or all the registers must be written before the rollover of the seconds counter (the oscillator is stopped by default after a Reset or power-up). The recommended practice is to load the Time and Date registers from 0x01 through 0x06 and then load the Seconds register 0x00 with the ST (oscillator Start bit set, enabling the oscillator).

This is outlined below in the sequence to set the following time and date:

11:45:23am 08/14/2012 Tuesday – 12-hour format

The example shows each register being written separately. An alternative is to use a software array and load addresses 0x01 through 0x06 in one I²C write sequence. This should be followed by writing the Seconds register 0x00. It should be noted that the RTCC does not assign a value to the day of the week, this can be set by the user. The RTCC will increment this counter at the midnight rollover.

1. Stop the oscillator.
2. Write 0x45 to the Minutes register [0x01].
3. Write 0x71 to the Hours register [0x02] (11 hours + 12-hour format).
4. Write 0x02 to the Day register [0x03] (Day register assignment is arbitrary, this register is just a counter). If enabling the external battery backup then the VBATEN bit must be set (VBATEN = 1).
5. Write 0x14 to the Date register [0x04].
6. Write 0x08 to the Month register [0x05].
7. Write 0x12 to the Year register [0x06].
With all the Time and Date registers loaded, the oscillator can be started by loading the Seconds register and setting the ST bit (ST = 1).
8. Write 0xA3 to the Seconds register [0x00] – 23 Seconds + ST bit set.
The code required to perform this function is shown below.

```c
void WriteRTCCTime(void)
{
IdleI2C(); //Ensure that I2C Module is Idle
StartI2C(); //Generate I2C Start Condition
WriteI2C(0xDE); //Control byte for RTCC
IdleI2C(); //Wait for I2C Module
WriteI2C(0x00); //Address for Minutes
IdleI2C();
WriteI2C(0x45); //45 Minutes
IdleI2C();
WriteI2C(0x11|12HOUR|AM); //11 Hours 12 Hour Format
IdleI2C();
//WriteI2C(0x02);//Tuesday
WriteI2C(0x02|VBATEN);//Tuesday + VBATEN set
IdleI2C();
WriteI2C(0x14); //14th of the Month
IdleI2C();
WriteI2C(0x08); //August
IdleI2C();
WriteI2C(0x12); //2012
IdleI2C();
StopI2C(); //Generate Stop Condition
//Now write the seconds and Start the Oscillator
StartI2C(); //Generate I2C Start Condition
WriteI2C(0xDE); //Control byte for RTCC
IdleI2C();
WriteI2C(0x00); //Address for Seconds Register
IdleI2C();
WriteI2C(0x23|ST); //Load seconds + ST=1
IdleI2C();
StopI2C(); //Send Stop Bit
}
```

MFP Configuration
The RTCC features a multi-function pin (MFP) that supports the following functions:
- CLKOUT function (1Hz-32.768 kHz)
- Alarm function
- Open Drain output controlled by the OUT bit

It should be noted that the MFP pin is an open drain configuration that requires a pull-up resistor to operate correctly. Additionally, the MFP clockout is derived from the main oscillator, so the oscillator has to be running for this function to work (ST = 1).

CLKOUT Configuration
In this example, we will configure the MFP pin as a square wave output with a frequency of 1Hz. The sequence required to do this is below:

It is assumed that the oscillator is running.
1. Write 0x40 to the Control register [0x07] – SQWE=1 + RS2=0 + RS1=0 + RS0=0 + OUT=0

The code to perform this is shown below:

```c
void SetMFP_1Hz(void)
{
IdleI2C(); //Ensure that I2C Module is Idle
StartI2C(); //Generate I2C Start Condition
WriteI2C(0xDE); //Control byte for RTCC
IdleI2C();
WriteI2C(0x07); //Address for Control Register
IdleI2C();
WriteI2C(0x40);
IdleI2C();
StopI2C(); //Generate Stop Condition
}
```

Alarm Configuration
The RTCC features two independent alarms. Both alarms can function independently of each other, however, it should be noted that the interrupt from the alarm modules is tied to the same MFP pin.

The example shown here shows the configuration of ALARM0. ALARM1 is identical and can be configured in the same manner.

In this example, the time is already loaded as shown in the basic configuration example:

```c
11:45:23am 08/14/2012 Tuesday – 12-hour format
```
The Alarm0 registers are loaded for this time + 2 Minutes.

```c
11:47:23am 08/14/2012 Tuesday – 12-hour format
```
The alarm configuration will be set with the following configuration:
- Alarm 0 Active
- Alarm 0 Polarity Low (MFP asserts low on alarm)
• Full Alarm match (Seconds, Minutes, Hour, Day, Date and Month) – all conditions must match for the alarm to be triggered.

The following steps show how the Alarm 0 is configured. Alarm 1 can be configured in the same manner.

1. Write 0x23 to the Alarm0 Seconds register [0x0A].
2. Write 0x47 to the Alarm0 Minutes register [0x0B].
3. Write 0x71 to the Alarm0 Hours register [0x0C] – 11 hours in 12-hour format.
4. Write 0x72 to the Alarm0 Day register [0x0D] – Tuesday + Alarm Polarity Low + Match on all. The Alarm0 Interrupt Flag is also cleared.
5. Write 0x14 to the Alarm0 Date register [0x0E].
6. Write 0x08 to the Alarm0 Month register [0x0F].

With all the Alarm0 registers set we can now activate the Alarm0 on the Control register.

7. Write 0x10 to the Control register [0x07] – Alarm0 enabled no CLKOUT, Alarm1 disabled.

Care should be taken to ensure that the ALMxIF (Alarm Interrupt Flag is cleared by the user software). The AlarmX Day register contains this bit. The register should be read before being modified and written. This will prevent changing the alarm configuration. Also, it is recommended that the alarm registers be loaded before the alarm is enabled.

The code required to configure the Alarm0 for this condition is shown below.

```c
void ConfigureAlarm0(void)
{
    IdleI2C(); //Ensure that I2C Module is Idle
    StartI2C(); //Generate I2C Start Condition
    WriteI2C(0xDE); //Control byte for RTCC
    IdleI2C();
    WriteI2C(0x0A); //Address for Seconds Alarm0
    IdleI2C();
    WriteI2C(0x23); //23 Seconds Alarm0
    IdleI2C();
    WriteI2C(0x47); //47 Minutes Alarm0
    IdleI2C();
    WriteI2C(0x11 | 12HOUR | AM); //11 Hours -12 Hour Format
    Alarm0
    IdleI2C();
    WriteI2C(0x02 | MATCHALL); //Tuesday + Match on all Conditions Alarm0
    IdleI2C();
    WriteI2C(0x14); //14th of the Month Alarm0
    IdleI2C();
    WriteI2C(0x08); //August Alarm0
    IdleI2C();
    StopI2C(); //Generate Stop Condition
    //Now write the seconds and Start the Oscillator
    StartI2C(); //Generate I2C Start Condition
    WriteI2C(0xde); //Control byte for RTCC
    IdleI2C();
    WriteI2C(0x07); //Address for Control Register
    IdleI2C();
    WriteI2C(0x10); //Alarm 0 enabled, No CLKOUT
    IdleI2C();
    StopI2C(); //Send Stop Bit
}
```

**CONCLUSION**

Following the steps in this application note, along with the included MPLAB® project, show the basic configuration of the Microchip I²C RTCC devices. By using available off-the-shelf development tools, any hardware issues will be mitigated allowing the engineer to concentrate on the firmware development. The code is presented in C and can easily be ported to other PIC MCU platforms.
APPENDIX A:
Functions to convert between Hexadecimal and BCD format. The code examples shown here are for reference, additional code is available on the Microchip web site: www.microchip.com.

```c
//This function converts a BCD to DEC
//Input: BCD Value
//Returns: Hex Value
char BcdToDec(char val)
{
    return ( (val/16*10) + (val%16) );
}

//This function converts HEX to BCD
//Input: Hex Value
//Returns: BCD Value
char DecToBcd(char val)
{
    return ( (val/10*16) + (val%10) );
}
```

APPENDIX B:
The following application notes support the low-level I2C commands used in this application note:

PIC10/12 – AN982, “Interfacing I2C™ Serial EEPROMs to PIC10 and PIC12 Devices” (DS00982) (ASM Bit Bang)

PIC16 – AN974, “Interfacing I2C™ Serial EEPROMs to PICmicro® Microcontroller’s” (DS00974) (ASM Bit Bang)

PIC16 – AN976, “Using the MSSP Module to Interface I2C™ EEPROMs with PIC16 Devices” (DS00976) (ASM Hardware Module)

PIC18 – AN979, “Interfacing I2C™ Serial EEPROMs to PIC18 Devices” (DS00979) (ASM Bit Bang)

PIC18 – AN989, “Using the MSSP Module to Interface I2C™ Serial EEPROMs with PIC18 Devices” (DS00989) (ASM Hardware Module)

PIC18 – AN997, “Using the C18 Compiler to Interface I2C™ Serial EEPROMs with PIC18 Devices” (DS00997) (C Bit Bang)

PIC18 – AN991, “Using the C18 Compiler and the MSSP to Interface I2C™ EEPROMs with PIC18 Devices” (DS00991) (C Hardware Module)

PIC24/dsPIC – AN1100, “Using the C30 Compiler to Interface Serial EEPROMs with dsPIC33” (DS01100) (C Bit Bang)

PIC24/dsPIC – AN1079, “Interfacing dsPIC33F with I2C™ Serial EEPROMs” (DS01079) (C Hardware Module)
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELoo, KEELoo logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, Hi-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MFP, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rFLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
Printed on recycled paper.
ISBN: 9781620769614

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, Keeloo® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
### AMERICAS

Corporation Office  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
http://www.microchip.com/support  
Web Address:  
www.microchip.com

**Atlanta**  
Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

**Boston**  
Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

**Chicago**  
Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

**Cleveland**  
Independence, OH  
Tel: 216-447-0464  
Fax: 216-447-0643

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

**Detroit**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

**Indianapolis**  
Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453

**Los Angeles**  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

**Santa Clara**  
Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

**Toronto**  
Mississauga, Ontario, Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

**Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**China - Beijing**  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

**China - Chengdu**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**China - Chongqing**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

**China - Hangzhou**  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

**China - Hong Kong SAR**  
Tel: 852-2943-5100  
Fax: 852-2401-3431

**China - Nanjing**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

**China - Qingdao**  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

**China - Shanghai**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**China - Shenyang**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**China - Shenzhen**  
Tel: 86-755-8864-2200  
Fax: 86-755-8203-1760

**China - Wuhan**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**China - Xian**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**China - Xiamen**  
Tel: 86-592-2388138  
Fax: 86-592-2388130

**China - Zhuhai**  
Tel: 86-756-3210040  
Fax: 86-756-3210049

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**India - Pune**  
Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

**Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

**Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or 82-2-558-5934

**Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-8850

**Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

**Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**Taiwan - Hsin Chu**  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**  
Tel: 886-7-213-7628  
Fax: 886-7-330-9305

**Taiwan - Taipei**  
Tel: 886-2-2508-8600  
Fax: 886-2-2508-0102

**Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

**Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

Worldwide Sales and Service

11/29/12