Real-Time Clock and Calendar Library

Majenko Technologies

November 2011

Contents

1 Description 3

2 Public Methods 4

2.1 begin(); 4

2.2 unsigned char hours(); 4

2.3 unsigned char minutes(); 4

2.4 unsigned char seconds(); 4

2.5 unsigned char year(); 4

2.6 unsigned char month(); 4

2.7 unsigned char day(); 4

2.8 unsigned char dayOfWeek(); 4

2.9 void hours(unsigned char); 4

2.10 void minutes(unsigned char); 4

2.11 void seconds(unsigned char); 4

2.12 void year(unsigned char); 4

2.13 void month(unsigned char); 4

2.14 void day(unsigned char); 4

2.15 void dayOfWeek(unsigned char); 4

2.16 void outputEnable(); 5

2.17 void outputEnable(unsigned char); 5

2.18 void outputDisable(); 5

2.19 void alarmEnable(); 5

2.20 void alarmDisable(); 5

2.21 void chimeEnable(); 5

2.22 void chimeDisable(); 5

2.23 void alarmMask(unsigned char); 5

2.24 void alarmRepeat(unsigned char); 6

2.25 void alarmHours(unsigned char); 6

2.26 void alarmMinutes(unsigned char); 6

2.27 void alarmSeconds(unsigned char); 6

2.28 void alarmMonth(unsigned char); 6

2.29 void alarmDay(unsigned char); 6

2.30 void alarmDayOfWeek(unsigned char); 6

2.31 void attachInterrupt(void (*)()); 6

3 Public Properties 7

4 Examples 8

4.1 Use the alarm to send the time through the serial port every second 8

5 Copyright 10

1 Description

The RTCC library provides a simple interface to the internal Real-Time Clock and Calendar module of the PIC32 used in the chipKIT™ Uno32™. The interface is exposed through a RTCC class and includes facilities for setting / retrieving the time and date, and configuring the alarm. An interrupt is provided to attach your own interrupt routine to which is triggered when the alarm is activated.

2 Public Methods

2.1 begin();

Enable and configure the RTCC module.

2.2 unsigned char hours();

2.3 unsigned char minutes();

2.4 unsigned char seconds();

2.5 unsigned char year();

2.6 unsigned char month();

2.7 unsigned char day();

2.8 unsigned char dayOfWeek();

These methods all return a part of the date or time from the RTCC module. All values are returned in an unsigned char and vary between 0 and 99. The actual return range depends on the part of the date or time requested.

Hours - 0 to 23
Minutes and Seconds - 0 to 59
Year - 0 to 99
Month - 1 to 12
Day - 1 to 31
Day of Week - 0 to 6

2.9 void hours(unsigned char);

2.10 void minutes(unsigned char);

2.11 void seconds(unsigned char);

2.12 void year(unsigned char);

2.13 void month(unsigned char);

2.14 void day(unsigned char);

2.15 void dayOfWeek(unsigned char);

These methods are the opposite to the methods 2.2 to 2.8 in that they are used to set the relevent portion of the date or time to a specific value.

2.16 void outputEnable();

2.17 void outputEnable(unsigned char);

Pin 2 of the chipKIT™ Uno32™ can be configured to be an output from the RTCC module. This can be either a regular 1-second, 50% duty cycle square wave, or be triggered by the internal alarm facility. If no parameter is specified it defaults to the output of the alarm state. Parameter can be one of:
SECONDS - Output the 1 second 50% duty cycle square wave.
ALARM - Toggle the output on alarm trigger.

2.18 void outputDisable();

This method disables the RTCC output on Pin 2 and returns it back to normal digital IO usage.

2.19 void alarmEnable();

Enable and configure the alarm system.

2.20 void alarmDisable();

Disable the alarm system. Note, this does not clear any alarm settings, it just turns it off.

2.21 void chimeEnable();

Enable "Chime" mode for the alarm. In Chime mode the alarm will keep triggering at the predefined regular alarm interval until the alarm is turned off.

2.22 void chimeDisable();

Disable "Chime" mode for the alarm and return it to normal repeat-n or single-shot mode.

2.23 void alarmMask(unsigned char);

Configure which part of the alarm time and date is used to trigger the alarm. Parameter is one of:

AL_HALF_SECOND The alarm will trigger every half second regardless of the configuration of any alarm time or date.

AL_SECOND The alarm will trigger every second regardless of the configuration of any alarm time or date.

AL_10_SECOND Only the seconds 0 to 9 are used to determine the alarm trigger time.

AL_MINUTE The alarm will trigger once a minute determined by the number of seconds 0 to 59.

AL_10_MINUTE The alarm will trigger sometime within a 10 minute window. The minutes 0 to 9 and the seconds 0 to 59 are used to determine the alarm time.

AL_HOUR The alarm will trigger hourly on the time specified by the minutes and seconds.

AL_DAY The full time is used to trigger the alarm, but the date is ignored. The alarm will trigger every day at the same time until the repeat runs out.

AL_WEEK The time and the day of the week are used to determine the alarm trigger time. The alarm will trigger on the same day at the same time each week.

AL_MONTH The day of the month and the time are used to determine the alarm trigger time. The alarm will trigger at the same time on the same day of each month.

AL_YEAR The alarm will trigger once a year determined by the month/day and full time combination of the alarm time.

2.24 void alarmRepeat(unsigned char);

Specify how many times an alarm should repeat until it is automatically disabled.

2.25 void alarmHours(unsigned char);

2.26 void alarmMinutes(unsigned char);

2.27 void alarmSeconds(unsigned char);

2.28 void alarmMonth(unsigned char);

2.29 void alarmDay(unsigned char);

2.30 void alarmDayOfWeek(unsigned char);

Set the alarm time and date. The values are in the same format as the normal time and date functions.

2.31 void attachInterrupt(void (*)());

Attach a function to the alarm interrupt. The function is in the form:

> void myInterruptFunction(void) { ... }

3 Public Properties

There are no public properties defined for this library.

4 Examples

4.1 Use the alarm to send the time through the serial port every second

```
#include <RTCC.h>

void setup()
{
  Serial.begin(9600);

  // Initialize the RTCC module
  RTCC.begin();

  // Set the time to something sensible
  RTCC.hours(9);
  RTCC.minutes(59);
  RTCC.seconds(0);
  RTCC.year(11);
  RTCC.month(05);
  RTCC.day(9);

  // Set the alarm to trigger every second
  RTCC.alarmMask(AL_SECOND);
  RTCC.chimeEnable();
  RTCC.alarmEnable();

  // Attach our routine to send the time through the serial port
  RTCC.attachInterrupt(&outputTime);
}

void loop()
{
}

void outputTime()
{
  char time[50];

  // Format the time and print it.
  sprintf(time,"%02d/%02d/%02d %02d:%02d:%02d\n",
    RTCC.day(),
    RTCC.month(),
    RTCC.year(),
    RTCC.hours(),
    RTCC.minutes(),
    RTCC.seconds()
  );
  Serial.print(time);
}
```

5 Copyright

All code is copyright 2011 Majenko Technologies. The code is provided freely with no warranty either implicitly or explicity implied. You may freely use the code for whatever you wish, but Majenko Technologies will not be liable for any damage of any form howsoever caused by the use or misuse of this code.