MCP794xx Library Documentation

Chris Krasnichuk
Version
Sat Oct 13 2018

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Class Documentation

MCP794xx Class Reference

Container for the **MCP794xx** functions This class is used to contain all the interacting functions within an object.

`#include <MCP794xx.h>`

Public Member Functions

- void **start** ()
  Activate the RTC Clock.
- void **stop** ()
  Stop the RTC Clock.
- void **setBattOn** ()
  Turn the battery backup ON (on by default)
- void **setBattOff** ()
  Turn the battery backup OFF.
- void **setHours12** (int hour, bool _PM)
  Set the hour in 12 Hour format.
- void **setHours24** (int hour)
  Set the hour in 24 Hour format.
- void **setMinutes** (int minute)
  Set the minutes.
- void **setSeconds** (int second)
  Set the seconds.
- void **setTime12** (int hour, bool _PM, int minute, int second)
  Set the time in 12 Hour format.
- void **setTime24** (int hour, int minute, int second)
  Set the time in 24 Hour format.
- void **setYear** (int year)
  Set the year.
- void **setMonth** (int month)
  Set the month.
- void **setDate** (int date)
  Set the day of the month.

- void **setWeekday** (int weekday)
  Set the weekday (Monday, Tuesday, ...)
- void **setCalendar** (int year, int month, int date)
  Set the Date (Weekday must be set seperately)
- int **getHours** ()
  Returns the hour (returns hour, for 12 hour format check PM variable)
- int **getMinutes** ()
  Returns the minute of the hour.
- int **getSeconds** ()
  Returns the seconds of the minute.
- int **getYear** ()
  Returns the year (Last 2 digits)
- int **getMonth** ()
  Returns the month.
- int **getDate** ()
  Returns the day of the month.
- int **getWeekday** ()
  Returns the weekday.
- void **setAlarmHours12** (bool alarmSelect, int hours, bool _PM)
  Set alarm 0/1 to trigger on a match of hours 12 hour format.
- void **setAlarmHours24** (bool alarmSelect, int hours)
  Set alarm 0/1 to trigger on a match of hours 12 hour format.
- void **setAlarmMinutes** (bool alarmSelect, int minutes)
  Set alarm 0/1 to trigger on a match of minutes.
- void **setAlarmSeconds** (bool alarmSelect, int seconds)
  Set alarm 0/1 to trigger on a match of seconds.
- void **setAlarmWeekday** (bool alarmSelect, int weekday)
  Set alarm 0/1 to trigger on a match of weekday.
- void **setAlarmDate** (bool alarmSelect, int date)
  Set alarm 0/1 to trigger on a match of day of the month.
- void **setAlarmAll12** (bool alarmSelect, int month, int date, int weekday, int hours, bool _PM, int minutes, int seconds)
  Set alarm 0/1 to trigger on a match of seconds, minutes, hours, weekday, day, and month.
- void **setAlarmAll24** (bool alarmSelect, int month, int date, int weekday, int hours, int minutes, int seconds)
  Set alarm 0/1 to trigger on a match of seconds, minutes, hours, weekday, day, and month.
- void **enableAlarm** (bool alarmSelect)
  Enable alarm 0/1.
- void **clearFlag** (bool alarmSelect)
  Clears Alarm Interrupt Flag.
- void **disableAlarm** (bool alarmSelect)
  Disable alarm 0/1.
- byte **checkAlarm** ()
  Returns a bitwise flag byte indicating which alarm went off and why.
- int **getPwrDownHours** ()
- int **getPwrDownMinutes** ()
  Returns the minute when the Vin power was cut-off.
- int **getPwrDownMonth** ()
  Returns the month when the Vin power was cut-off.
- int **getPwrDownDate** ()

Returns the day of the month when Vin power was cut-off.

- int **getPwrDownWeekday** ()
  Returns the day of the week (1-7) when Vin power was cut-off.
- int **getPwrUpHours** ()
- int **getPwrUpMinutes** ()
  Returns the minute when the Vin power was applied.
- int **getPwrUpMonth** ()
  Returns the month when the Vin power was applied.
- int **getPwrUpDate** ()
  Returns the day of the month when Vin power was applied.
- int **getPwrUpWeekday** ()
  Returns the day of the week (1-7) when Vin power was applied.
- void **setMFPin** (bool value)
  Sets the value of the Multifunction pin, disables alarms.
- void **setMFPinSquareWave** (int selectOut)
  Configures the Multifunction pin to output a sqaure wave, disables alarms.
- void **writeData** (byte reg, byte *buffer, int numBytes)
  Write a byte of data to SRAM.
- byte **readData** (byte reg, byte *buffer, int numBytes)
  read a byte of data from SRAM
- void **standbyMode** ()
  Currently Unused.

Public Attributes

- bool **PM** = NULL
  *Signifies if the last **getHours()** call had a 12h result in the PM.*
- bool **LPYR** = NULL
  *Signifies if the last **getYear()** resulted in a leap year.*

Detailed Description

Container for the **MCP794xx** functions This class is used to contain all the interacting functions within an object.

Member Function Documentation

MCP794xx::checkAlarm ()

Returns a bitwise flag byte indicating which alarm went off and why.

The returned byte is formatted as follows:

|ALM1IF|3 BITS FOR ALM1 MATCH CONFIG|ALM0IF|3 BITS FOR ALM0 MATCH CONFIG|

performing a bitwise AND (&) operation using the following masks:

_0statusIF _0statusMask _0matchSec _0matchMin _0matchHours _0matchWeekday

_0matchDate

_0matchAll

_1statusIF

_1statusMask

_1matchSec

_1matchMin

_1matchHours

_1matchWeekday

_1matchDate

_1matchAll

Will indicate which alarm went off and the match conditions.

void MCP794xx::clearFlag (bool *alarmSelect*)

Clears Alarm Interrupt Flag.

Parameters:

alarmSelect	used to select which alarm's interrupt flag to clear

void MCP794xx::disableAlarm (bool *alarmSelect*)

Disable alarm 0/1.

Parameters:

alarmSelect	used to select which alarm to disable

void MCP794xx::enableAlarm (bool *alarmSelect*)

Enable alarm 0/1.

Parameters:

alarmSelect	used to select which alarm to enable

MCP794xx::getPwrDownHours ()

Returns the hour when the Vin power was cut-off Works for both 12/24h formats 12h format uses the _PM class variable to indicate if the hour is in the PM

MCP794xx::getPwrUpHours ()

Returns the hour when the Vin power was applied Works for both 12/24h formats 12h format uses the _PM class variable to indicate if the hour is in the PM

void MCP794xx::setAlarmAll12 (bool *alarmSelect*, int *month*, int *date*, int *weekday*, int *hours*, bool *_PM*, int *minutes*, int *seconds*)

Set alarm 0/1 to trigger on a match of seconds, minutes, hours, weekday, day, and month.

Parameters:

alarmSelect	used to select which alarm to set
month	the month which must match for alarm to trigger
date	the day of the motnh which must match for the alarm to trigger
weekday	the day of the week which must match for the alarm to trigger
hours	the hour which must match for the alarm to trigger (12h format)
_PM	indicates whether the hour is in the AM or PM (TRUE if in PM)
minutes	the minute which must match for the alarm to trigger
seconds	the second which must match for the alarm to trigger

void MCP794xx::setAlarmAll24 (bool *alarmSelect*, int *month*, int *date*, int *weekday*, int *hours*, int *minutes*, int *seconds*)

Set alarm 0/1 to trigger on a match of seconds, minutes, hours, weekday, day, and month.

Parameters:

alarmSelect	used to select which alarm to set
month	the month which must match for alarm to trigger
date	the day of the motnh which must match for the alarm to trigger
weekday	the day of the week which must match for the alarm to trigger
hours	the hour which must match for the alarm to trigger (24h format)
minutes	the minute which must match for the alarm to trigger
seconds	the second which must match for the alarm to trigger

void MCP794xx::setAlarmDate (bool *alarmSelect*, int *date*)

Set alarm 0/1 to trigger on a match of day of the month.

Parameters:

alarmSelect	used to select which alarm to set
date	the day of the month which triggers the alarm (1-31)

void MCP794xx::setAlarmHours12 (bool *alarmSelect*, int *hours*, bool *_PM*)

Set alarm 0/1 to trigger on a match of hours 12 hour format.

Parameters:

alarmSelect	used to select which alarm to set
hours	the hour which triggers the alarm (12h format)
_PM	PM indicator for the hours, should be TRUE if the hour is in the PM

void MCP794xx::setAlarmHours24 (bool *alarmSelect*, int *hours*)

Set alarm 0/1 to trigger on a match of hours 12 hour format.

Parameters:

alarmSelect	used to select which alarm to set
hours	the hour which triggers the alarm (24h format)

void MCP794xx::setAlarmMinutes (bool *alarmSelect*, int *minutes*)

Set alarm 0/1 to trigger on a match of minutes.

Parameters:

alarmSelect	used to select which alarm to set
minutes	the minute which triggers the alarm

void MCP794xx::setAlarmSeconds (bool *alarmSelect*, int *seconds*)

Set alarm 0/1 to trigger on a match of seconds.

Parameters:

alarmSelect	used to select which alarm to set
seconds	the second which triggers the alarm

void MCP794xx::setAlarmWeekday (bool *alarmSelect*, int *weekday*)

Set alarm 0/1 to trigger on a match of weekday.

Parameters:

alarmSelect	used to select which alarm to set
weekday	the day of the week which triggers the alarm. Values from 1-7 (or use the enumeration)

void MCP794xx::setCalendar (int *year*, int *month*, int *date*)

Set the Date (Weekday must be set seperately)

Parameters:

year	the last two digits of the year to be set.
month	the month to be set
date	the day of the month to be set

void MCP794xx::setDate (int *date*)

Set the day of the month.

Parameters:

date	the day of the month to be set

void MCP794xx::setHours12 (int *hour*, bool *_PM*)

Set the hour in 12 Hour format.

Parameters:

hour	the hour to be set, in 12 hour format.

_PM	indicates if the hour to be set is AM (false) or PM (true)

void MCP794xx::setHours24 (int *hour*)

Set the hour in 24 Hour format.

Parameters:

hour	the hour to be set, in 24 hour format.

void MCP794xx::setMinutes (int *minute*)

Set the minutes.

Parameters:

minute	the minute to be set.

void MCP794xx::setMonth (int *month*)

Set the month.

Parameters:

month	the month to be set.

void MCP794xx::setSeconds (int *second*)

Set the seconds.

Parameters:

second	the second to be set.

void MCP794xx::setTime12 (int *hour*, bool *_PM*, int *minute*, int *second*)

Set the time in 12 Hour format.

Parameters:

hour	the hour to be set, in 12 hour format.
_PM	indicates if the hour to be set in AM (false) or PM (true)
minute	the minute to be set.
second	the second to be set.

void MCP794xx::setTime24 (int *hour*, int *minute*, int *second*)

Set the time in 24 Hour format.

Parameters:

hour	the hour to be set, in 24 hour format.
minute	the minute to be set.

second	the second to be set.

void MCP794xx::setWeekday (int *weekday*)

Set the weekday (Monday, Tuesday, ...)

Parameters:

weekday	the day of the week to be set. Values from 1-7 (or use the enumeration)

void MCP794xx::setYear (int *year*)

Set the year.

Parameters:

year	the last two digits of the year to be set.

The documentation for this class was generated from the following files:

- MCP794xx.h
- MCP794xx.cpp