DS3234

DS3234 (SPI Bus RTC) Arduino and chipKit library

Manual

PREFACE:

This library has been made to easily interface and use the DS3234 RTC with the Arduino and chipKit development boards.
This library makes use of the built-in hardware SPI port of the microcontroller so there are some pin connections that are required (see below).

You can always find the latest version of the library at
http://electronics.henningkarlsen.com/

If you make any modifications or improvements to the code, I would appreciate that you share the code with me so that I might include it in the next release. I can be contacted through **http://electronics.henningkarlsen.com/contact.php**.

For version information, please refer to **version.txt**.

REQUIRED PINS:

	Arduino					**chipKit**			
DS3234	**Uno**	**Mega**	**Due**	**Leonardo**	**Bobuino**	**Uno32**	**uC32**	**Max32**	
DIN	D11	D51	D75	D16	D11	D11	D11	D51	> MOSI
DOUT	D12	D50	D74	D14	D12	D12	D12	D50	> MISO
SCLK	D13	D52	D76	D15	D13	D13	D13	D52	
CS	User selectable (Set to D8 by default in the supplied demos)								

- Note that the SPI pins are only available on the ICSP header on the Arduino Due and Arduino Leonardo.
- Boards with SPI Master/Slave Select jumpers should be set to the Master position.

ICSP HEADER PINOUT:

Pin #	**Signal**			**Signal**	**Pin #**
1	MISO	●	●	Vcc	2
3	SCLK	●	●	MOSI	4
5	Reset	●	●	GND	6

OTHER IMPORTANT INFORMATION:

The library has only been tested with Vcc connected to 3.3v. While the chip can tolerate up to 5.5v Vcc supply it is recommended to use 3.3v.
All inputs does tolerate up to 5.5v even if Vcc is connected to 3.3v.

STRUCTURES:

Time;	
Structure to manipulate time- and date-data.	

Variables:	`hour, min, sec:  For holding time-data`
	`date, mon, year: For holding date-data`
	`dow:             Day-of-the-week with monday being the first day`

Usage:	`Time t; // Define a structure named t of the Time-class`

DEFINED LITERALS:

Weekdays

For use with setDOW() and Time.dow

MONDAY:	1
TUESDAY:	2
WEDNESDAY:	3
THURSDAY:	4
FRIDAY:	5
SATURDAY:	6
SUNDAY:	7

Select length

For use with getTimeStr(), getDateStr(), getDOWStr() and getMonthStr()

FORMAT_SHORT:	1
FORMAT_LONG:	2

Select date format

For use with getDateStr()

FORMAT_LITTLEENDIAN:	1
FORMAT_BIGENDIAN:	2
FORMAT_MIDDLEENDIAN:	3

FUNCTIONS:

DS3234(CE);	
The main class of the interface.	
Parameters:	`CE:   CE-pin of the DS3234`
Usage:	`DS3234 rtc(8); // Start an instance of the DS3234 class`

setTime(hour, min, sec);	
Set the time.	

Parameters:	`hour: Hour to store in the DS3234 (0-23)`
	`min:  Minute to store in the DS3234 (0-59)`
	`sec:  Second to store in the DS3234 (0-59)`

Returns:	`Nothing`
Usage:	`rtc.setTime(23, 59, 59); // Set the time to 23:59:59`

setDate(date, mon, year);	
Set the date.	

Parameters:	`date: Date of the month to store in the DS3234 (1-31)`
	`mon:  Month to store in the DS3234 (1-12)`
	`year: Year to store in the DS3234 (2000-2099)`

Returns:	`Nothing`
Usage:	`rtc.setDate(26, 1, 2014); // Set the date to January 26th, 2014.`
Notes:	No checking for illegal dates will be done so Feb 31th is possible to input. *The effect of doing this is unknown.*

setDOW(dow);	
Set the day-of-the-week.	
Parameters:	`dow: Day of the week to store in the DS3234 (1-7)`
Returns:	`Nothing`
Usage:	`rtc.setDOW(FRIDAY); // Set the day-of-the-week to be Friday`
Notes:	Monday is 1, and through to Sunday being 7

getTime();

Get current data from the DS3234.

Parameters:	None
Returns:	Time-structure
Usage:	`t = rtc.getTime(); // Read current time and date.`

getTimeStr([format]);

Get current time as a string.

Parameters:	format: **<Optional>** FORMAT_LONG "hh:mm:ss" (default) FORMAT_SHORT "hh:mm"
Returns:	(char array) containing the current time with or without seconds.
Usage:	`Serial.print(rtc.getTimeStr()); // Send the current time over a serial connection`

getDateStr([slformat[, eformat[, divider]]]);

Get current date as a string.

Parameters:	slformat: **<Optional>** FORMAT_LONG Year with 4 digits (yyyy) (default) FORMAT_SHORT Year with 2 digits (yy) eformat: **<Optional>** FORMAT_LITTLEENDIAN "dd.mm.yyyy" (default) FORMAT_BIGENDIAN "yyyy.mm.dd" FORMAT_MIDDLEENDIAN "mm.dd.yyyy" divider: **<Optional>** Single character to use as divider. Default is '.'
Returns:	(char array) containing the current date in the specified format.
Usage:	`Serial.print(rtc.getDateStr()); // Send the current date over a serial connection`
Notes:	More information on date formats can be found on Wikipedia: http://en.wikipedia.org/wiki/Date_format#Date_format

getDOWStr([format]);

Get current day-of-the-week as a string.

Parameters:	format: **<Optional>** FORMAT_LONG Day-of-the-week in English (default) FORMAT_SHORT Abbreviated Day-of-the-week in English (3 letters)
Returns:	(char array) containing the current day-of-the-week in full or abbreviated format.
Usage:	`Serial.print(rtc.getDOWStr(FORMAT_SHORT)); // Send the current day in abbreviated format over a serial connection`

getMonthStr([format]);

Get current month as a string.

Parameters:	format: **<Optional>** FORMAT_LONG Month in English (default) FORMAT_SHORT Abbreviated month in English (3 letters)
Returns:	(char array) containing the current month in full or abbreviated format.
Usage:	`Serial.print(rtc.getMonthStr()); // Send the current month over a serial connection`

getTemp();

Get the current internal temperature from the DS3234.

Parameters:	none
Returns:	(float) containing the current temperature in °C.
Usage:	`temp = rtc.getTemp(); // Read the current temperature and put the result in temp`
Notes:	The temperature sensor has an accuracy of ±3°C and a precision of 0.25°C. The temperature is updated every 64 seconds.

poke(address, value);	
Write one single byte to on-chip RAM.	

Parameters:	`address: address of byte to write (0-255)`
	`value  : value to write to <address> (0-255)`

Returns:	`Nothing`
Usage:	`rtc.poke(15, 160); // Write 160 to address 15`

peek(address);	
Read one single byte from on-chip RAM.	
Parameters:	`address: address of byte to read (0-255)`
Returns:	`(byte) containing data read from on-chip RAM`
Usage:	`b=rtc.peek(18); // Read a single byte from address 18 and put the result in b`