Erriez DS1307 I2C RTC library for Arduino

1.0.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

DS1307 high precision I2C RTC library for Arduino

This is a DS1307 I2C RTC library for Arduino.

Library features

- libc `<time.h>` compatible
- Read/write date/time `struct tm`
- Set/get Unix epoch UTC `time_t`
- Set/get time (hour, min, sec)
- Set/get date and time (hour, min, sec, mday, mon, year, wday)
- Control `SQW` signal (disable / 1Hz / 4096Hz / 8192Hz / 32768Hz)
- Full RTC register access

Hardware

Any Arduino hardware with a TWI interface and `Wire.h` support.

Pins

Pins board - DS1307	VCC	GND	SDA	SCL	SQW
Arduino UNO (ATMega328 boards)	5V	GND	A4	A5	D2 (INT0)
Arduino Mega2560	5V	GND	D20	D21	D2 (INT4)
Arduino Leonardo	5V	GND	D2	D3	D7 (INT6)
Arduino DUE (ATSAM3X8E)	3V3	GND	20	21	2
ESP8266	3V3	GND	GPIO4 (D2)	GPIO5 (D1)	GPIO0 (D3)
ESP32	3V3	GND	GPIO21	GPIO22	GPIO0

Note: Tested ESP8266 / ESP32 boards:

- **ESP8266 boards**: ESP12F / WeMos D1 & R2 / Node MCU v2 / v3
- **ESP32 boards:** WeMos LOLIN32 / LOLIN D32

Other unlisted MCU's may work, but are not tested.

Examples

Arduino IDE | Examples | Erriez DS1307 RTC:

- Alarm Software alarm
- Bare Bare example without a library
- DumpRegisters Dump registers for debugging
- SetBuildDateTime Set build date/time
- SetGetDateTime Set/get date and time
- SetGetTime Set/get time
- SQWInterrupt 1Hz SQW interrupt pin
- Terminal Serial terminal example
- Test Regression test
- WriteRead Write/read `struct tm`

Documentation

- Doxygen online HTML
- Doxygen PDF
- DS1307 datasheet

Usage

Initialization

```
{c++}
#include <Wire.h>
#include <ErriezDS1307.h>

// Create RTC object
ErriezDS1307 rtc;

void setup()
{
    // Initialize TWI with a 100kHz (default)
    Wire.begin();
    Wire.setClock(100000);

    // Initialize RTC
    while (!rtc.begin()) {
        Serial.println(F("RTC not found"));
        delay(3000);
    }
}
```

Check oscillator status at startup

```
{c++}
// Check oscillator status
if (!rtc.isRunning()) {
    // Error: RTC oscillator stopped. Date/time cannot be trusted.
    // Set new date/time before reading date/time.

    // Enable oscillator
    rtc.clockEnable(true);
}
```

Set time

```
{c++}
// Write time to RTC
if (!rtc.setTime(12, 0, 0)) {
    // Error: Set time failed
}
```

Get time

```
{c++}
uint8_t hour;
uint8_t minute;
uint8_t second;

// Read time from RTC
if (!rtc.getTime(&hour, &minute, &second)) {
    // Error: RTC read failed
}
```

Set date and time

```
{c++}
// Write RTC date/time: 13:45:09  31 December 2019  2=Tuesday
if (!rtc.setDateTime(13, 45, 9,  31, 12, 2019,  2) {
    // Error: RTC write failed
}
```

Get date and time

```
{c++}
uint8_t hour;
uint8_t min;
uint8_t sec;
uint8_t mday;
uint8_t mon;
uint16_t year;
uint8_t wday;

// Read RTC date/time
if (!rtc.getDateTime(&hour, &min, &sec, &mday, &mon, &year, &wday) {
    // Error: RTC read failed
}

// hour: 0..23
// min: 0..59
// sec: 0..59
// mday: 1..31
// mon: 1..12
// year: 2000..2099
// wday: 0..6 (0=Sunday .. 6=Saturday)
```

Write date/time struct tm

```
{c++}
struct tm dt;

dt.tm_hour = 12;
dt.tm_min = 34;
dt.tm_sec = 56;
dt.tm_mday = 29;
dt.tm_mon = 1; // 0=January
dt.tm_year = 2020-1900;
dt.tm_wday = 6; // 0=Sunday

if (!rtc.write(&dt)) {
    // Error: RTC Read failed
}
```

Read date/time struct tm

```
{c++}
struct tm dt;

// Read RTC date/time
if (!rtc.read(&dt)) {
    // Error: RTC read failed
}
```

Read Unix Epoch UTC

```
{c++}
time_t t;

// Read Unix epoch UTC from RTC
if (!rtc.getEpoch(&t)) {
    // Error: RTC read failed
}
```

Write Unix Epoch UTC

```
{c++}
// Write Unix epoch UTC to RTC
if (!rtc.setEpoch(1599416430UL)) {
    // Error: Set epoch failed
}
```

Square Wave Out (SQW)

```
{c++}
rtc.setSquareWave(SquareWaveDisable);   // Disable
rtc.setSquareWave(SquareWave1024Hz);    // 1024Hz
rtc.setSquareWave(SquareWave4096Hz);    // 4096Hz
rtc.setSquareWave(SquareWave8192Hz);    // 8192Hz
rtc.setSquareWave(SquareWave32768Hz);   // 32768Hz
```

Library dependencies

- `Wire.h`
- `Terminal.ino` requires `ErriezSerialTerminal` library.

Library installation

Please refer to the `Wiki` page.

More Arduino Libraries from Erriez

- `Erriez Libraries`

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 ErriezDS1307 Class Reference

DS1307 RTC class.

```
#include <ErriezDS1307.h>
```

Public Member Functions

- bool begin ()

  Initialize and detect DS1307 RTC.
- bool isRunning ()

  Read RTC CH (Clock Halt) from seconds register.
- bool clockEnable (bool enable=true)

  Enable or disable oscillator.
- time_t getEpoch ()

  Read Unix UTC epoch time_t.
- bool setEpoch (time_t t)

  Write Unix epoch UTC time to RTC.
- bool read (struct tm ∗dt)

  Read date and time from RTC.
- bool write (const struct tm ∗dt)

  Write date and time to RTC.
- bool setTime (uint8_t hour, uint8_t min, uint8_t sec)

  Write time to RTC.
- bool getTime (uint8_t ∗hour, uint8_t ∗min, uint8_t ∗sec)

  Read time from RTC.
- bool setDateTime (uint8_t hour, uint8_t min, uint8_t sec, uint8_t mday, uint8_t mon, uint16_t year, uint8_t wday)

  Set date time.
- bool getDateTime (uint8_t ∗hour, uint8_t ∗min, uint8_t ∗sec, uint8_t ∗mday, uint8_t ∗mon, uint16_t ∗year, uint8_t ∗wday)

  Get date time.
- bool setSquareWave (SquareWave squareWave)

  Configure SQW (Square Wave) output pin.

- uint8_t bcdToDec (uint8_t bcd)

  BCD to decimal conversion.
- uint8_t decToBcd (uint8_t dec)

  Decimal to BCD conversion.
- uint8_t readRegister (uint8_t reg)

  Read register.
- bool writeRegister (uint8_t reg, uint8_t value)

  Write register.
- bool readBuffer (uint8_t reg, void ∗buffer, uint8_t len)

  Read buffer from RTC.
- bool writeBuffer (uint8_t reg, void ∗buffer, uint8_t len)

  Write buffer to RTC.

4.1.1 Detailed Description

DS1307 RTC class.

Definition at line 80 of file ErriezDS1307.h.

4.1.2 Member Function Documentation

4.1.2.1 bcdToDec()

```
uint8_t ErriezDS1307::bcdToDec (
            uint8_t bcd )
```

BCD to decimal conversion.

Parameters

bcd	BCD encoded value.

Returns

Decimal value.

Definition at line 443 of file ErriezDS1307.cpp.

4.1.2.2 begin()

```
bool ErriezDS1307::begin ( )
```

Initialize and detect DS1307 RTC.

Call this function from setup().

Return values

true	RTC detected.
false	RTC not detected.

Definition at line 52 of file ErriezDS1307.cpp.

4.1.2.3 clockEnable()

```
bool ErriezDS1307::clockEnable (
            bool enable = true )
```

Enable or disable oscillator.

Clear or set CH (Clock Halt) bit to seconds register

Parameters

enable	true: Enable RTC clock. false: Stop RTC clock.

Return values

true	Success.
false	Oscillator enable failed.

Definition at line 105 of file ErriezDS1307.cpp.

4.1.2.4 decToBcd()

```
uint8_t ErriezDS1307::decToBcd (
            uint8_t dec )
```

Decimal to BCD conversion.

Parameters

dec	Decimal value.

Returns

BCD encoded value.

Definition at line 455 of file ErriezDS1307.cpp.

4.1.2.5 getDateTime()

```
bool ErriezDS1307::getDateTime (
            uint8_t * hour,
            uint8_t * min,
            uint8_t * sec,
            uint8_t * mday,
            uint8_t * mon,
            uint16_t * year,
            uint8_t * wday )
```

Get date time.

Parameters

hour	Hours 0..23
min	Minutes 0..59
sec	Seconds 0..59
mday	Day of the month 1..31
mon	Month 1..12 (1=January)
year	Year 2000..2099
wday	Day of the week 0..6 (0=Sunday, .. 6=Saturday)

Return values

true	Success.
false	Get date/time failed.

Definition at line 391 of file ErriezDS1307.cpp.

4.1.2.6 getEpoch()

```
time_t ErriezDS1307::getEpoch ( )
```

Read Unix UTC epoch time_t.

Returns

Unix epoch time_t seconds since 1970.

Definition at line 130 of file ErriezDS1307.cpp.

4.1.2.7 getTime()

```
bool ErriezDS1307::getTime (
            uint8_t * hour,
            uint8_t * min,
            uint8_t * sec )
```

Read time from RTC.

Read hour, minute and second registers from RTC.

Parameters

hour	Hours 0..23.
min	Minutes 0..59.
sec	Seconds 0..59.

Return values

true	Success.
false	Invalid second, minute or hour read from RTC. The time is set to zero.

Definition at line 303 of file ErriezDS1307.cpp.

4.1.2.8 isRunning()

```
bool ErriezDS1307::isRunning ( )
```

Read RTC CH (Clock Halt) from seconds register.

The application is responsible for checking the CH (Clock Halt) bit before reading date/time date. This function may be used to judge the validity of the date/time registers.

Return values

true	RTC clock is running.
false	The date/time data is invalid when the CH bit is set. The application should enable the oscillator, or program a new date/time.

Definition at line 81 of file ErriezDS1307.cpp.

4.1.2.9 read()

```
bool ErriezDS1307::read (
             struct tm * dt )
```

Read date and time from RTC.

Read all RTC registers at once to prevent a time/date register change in the middle of the register read operation.

Parameters

dt	Date and time struct tm.

Return values

true	Success
false	Read failed.

Definition at line 190 of file ErriezDS1307.cpp.

4.1.2.10 readBuffer()

```
bool ErriezDS1307::readBuffer (
             uint8_t reg,
             void * buffer,
             uint8_t readLen )
```

Read buffer from RTC.

Parameters

reg	RTC register number 0x00..0x07.
buffer	Buffer.
readLen	Buffer length. Reading is only allowed within valid RTC registers.

Return values

true	Success
false	I2C read failed.

Definition at line 541 of file ErriezDS1307.cpp.

4.1.2.11 readRegister()

```
uint8_t ErriezDS1307::readRegister (
             uint8_t reg )
```

Read register.

Please refer to the RTC datasheet.

Parameters

reg	RTC register number 0x00..0x12.

Returns

value 8-bit unsigned register value.

Definition at line 469 of file ErriezDS1307.cpp.

4.1.2.12 setDateTime()

```
bool ErriezDS1307::setDateTime (
             uint8_t hour,
             uint8_t min,
             uint8_t sec,
             uint8_t mday,
             uint8_t mon,
             uint16_t year,
             uint8_t wday )
```

Set date time.

Parameters

hour	Hours 0..23
min	Minutes 0..59
sec	Seconds 0..59
mday	Day of the month 1..31
mon	Month 1..12 (1=January)
year	Year 2000..2099
wday	Day of the week 0..6 (0=Sunday, .. 6=Saturday)

Return values

true	Success.
false	Set date/time failed.

Definition at line 351 of file ErriezDS1307.cpp.

4.1.2.13 setEpoch()

```
bool ErriezDS1307::setEpoch (
             time_t t )
```

Write Unix epoch UTC time to RTC.

Parameters

t	time_t time

Return values

true	Success.

Return values

false	Set epoch failed.

Definition at line 162 of file ErriezDS1307.cpp.

4.1.2.14 setSquareWave()

```
bool ErriezDS1307::setSquareWave (
             SquareWave squareWave )
```

Configure SQW (Square Wave) output pin.

This will disable or initialize the SQW clock pin.

Parameters

squareWave	SquareWave configuration: Disable: SquareWaveDisable 1Hz: SquareWave1Hz 4096Hz: SquareWave4096Hz 8192Hz: SquareWave8192Hz 32748Hz: SquareWave32768Hz

Return values

true	Success
false	Set squareWave failed.

Definition at line 430 of file ErriezDS1307.cpp.

4.1.2.15 setTime()

```
bool ErriezDS1307::setTime (
             uint8_t hour,
             uint8_t min,
             uint8_t sec )
```

Write time to RTC.

Write hour, minute and second registers to RTC.

Parameters

hour	Hours 0..23.
min	Minutes 0..59.
sec	Seconds 0..59.

Return values

true	Success.
false	Set time failed.

Definition at line 277 of file ErriezDS1307.cpp.

4.1.2.16 write()

```
bool ErriezDS1307::write (
            const struct tm * dt )
```

Write date and time to RTC.

Write all RTC registers at once to prevent a time/date register change in the middle of the register write operation. This function enables the oscillator.

Parameters

dt	Date/time struct tm. Providing invalid date/time data may result in unpredictable behavior.

Return values

true	Success.
false	Write failed.

Definition at line 245 of file ErriezDS1307.cpp.

4.1.2.17 writeBuffer()

```
bool ErriezDS1307::writeBuffer (
            uint8_t reg,
            void * buffer,
            uint8_t writeLen )
```

Write buffer to RTC.

Please refer to the RTC datasheet.

Parameters

reg	RTC register number 0x00..0x07.
buffer	Buffer.
writeLen	Buffer length. Writing is only allowed within valid RTC registers.

Return values

true	Success
false	I2C write failed.

Definition at line 513 of file ErriezDS1307.cpp.

4.1.2.18 writeRegister()

```
bool ErriezDS1307::writeRegister (
            uint8_t reg,
            uint8_t value )
```

Write register.

Please refer to the RTC datasheet.

Parameters

reg	RTC register number 0x00..0x12.
value	8-bit unsigned register value.

Return values

true	Success
false	Write register failed

Definition at line 492 of file ErriezDS1307.cpp.

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

- src/ErriezDS1307.h
- src/ErriezDS1307.cpp

Chapter 5

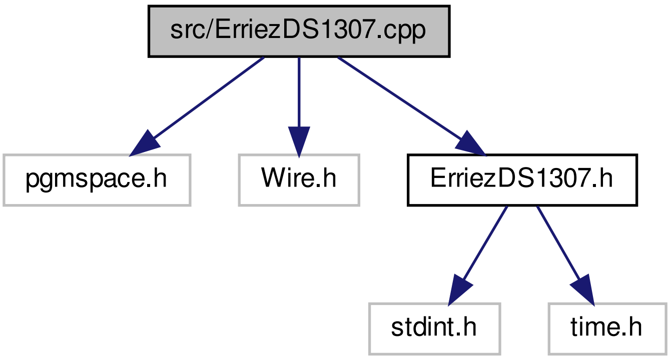
File Documentation

5.1 src/ErriezDS1307.cpp File Reference

DS1307 RTC library for Arduino.

```
#include <pgmspace.h>
#include <Wire.h>
#include "ErriezDS1307.h"
```

Include dependency graph for ErriezDS1307.cpp:



5.1.1 Detailed Description

DS1307 RTC library for Arduino.

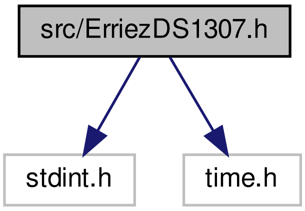
Source: https://github.com/Erriez/ErriezDS1307 Documentation: https://erriez.↵ github.io/ErriezDS1307

5.2 src/ErriezDS1307.h File Reference

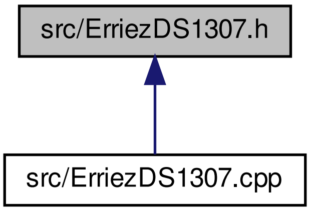
DS1307 RTC library for Arduino.

```
#include <stdint.h>
#include <time.h>
```

Include dependency graph for ErriezDS1307.h:



This graph shows which files directly or indirectly include this file:



Classes

- class ErriezDS1307

  DS1307 RTC class.

Macros

- #define DS1307_REG_SECONDS 0x00

  DS1307 registers.
- #define DS1307_REG_MINUTES 0x01

  Minutes register.
- #define DS1307_REG_HOURS 0x02

  Hours register.

- #define DS1307_REG_DAY_WEEK 0x03

  Day of the week register.
- #define DS1307_REG_DAY_MONTH 0x04

  Day of the month register.
- #define DS1307_REG_MONTH 0x05

  Month register.
- #define DS1307_REG_YEAR 0x06

  Year register.
- #define DS1307_REG_CONTROL 0x07

  Control register.
- #define DS1307_NUM_REGS 8

  DS1307 number of registers.
- #define DS1307_SEC_CH 7

  DS1307 register bit defines.
- #define DS1307_HOUR_12H_24H 6

  12 or 24 hour mode
- #define DS1307_HOUR_AM_PM 5

  AM/PM.
- #define DS1307_CTRL_OUT 7

  Enable oscillator.
- #define DS1307_CTRL_SQWE 4

  Square-Wave Enable.
- #define DS1307_CTRL_RS1 1

  Rate Select 1.
- #define DS1307_CTRL_RS0 0

  Rate Select 0.
- #define DS1307_ADDR (0xD0 >> 1)

  DS1307 I2C 7-bit address.

Enumerations

- enum SquareWave {
  SquareWaveDisable = ((0 << DS1307_CTRL_SQWE) | (0 << DS1307_CTRL_RS1) | (0 << DS1307_CTRL_RS0)), SquareWave1Hz = ((1 << DS1307_CTRL_SQWE) | (0 << DS1307_CTRL_RS1) | (0 << DS1307_CTRL_RS0)), SquareWave4096Hz = ((1 << DS1307_CTRL_SQWE) | (0 << DS1307_CTRL_RS1) | (1 << DS1307_CTRL_RS0)), SquareWave8192Hz = ((1 << DS1307_CTRL_SQWE) | (1 << DS1307_CTRL_RS1) | (0 << DS1307_CTRL_RS0)),
  SquareWave32768Hz = ((1 << DS1307_CTRL_SQWE) | (1 << DS1307_CTRL_RS1) | (1 << DS1307_CTRL_RS0)) }

  Squarewave enum.

5.2.1 Detailed Description

DS1307 RTC library for Arduino.

Source: `https://github.com/Erriez/ErriezDS1307` Documentation: `https://erriez.github.io/ErriezDS1307`

5.2.2 Macro Definition Documentation

5.2.2.1 DS1307_NUM_REGS

```
#define DS1307_NUM_REGS 8
```

DS1307 number of registers.

8 RTC register: 0x00..0x07

Definition at line 50 of file ErriezDS1307.h.

5.2.2.2 DS1307_REG_SECONDS

```
#define DS1307_REG_SECONDS 0x00
```

DS1307 registers.

Seconds register

Definition at line 40 of file ErriezDS1307.h.

5.2.2.3 DS1307_SEC_CH

```
#define DS1307_SEC_CH 7
```

DS1307 register bit defines.

Clock halt

Definition at line 53 of file ErriezDS1307.h.

5.2.3 Enumeration Type Documentation

5.2.3.1 SquareWave

```
enum SquareWave
```

Squarewave enum.

Enumerator

SquareWaveDisable	SQW disable.
SquareWave1Hz	SQW 1Hz.
SquareWave4096Hz	SQW 4096Hz.
SquareWave8192Hz	SQW 8192Hz.
SquareWave32768Hz	SQW 32768Hz.

Definition at line 68 of file ErriezDS1307.h.

Index