EE513 - Connected Embedded Systems: Assignment 1

Generated by Doxygen 1.8.13

Contents

Chapter 1

EE513 - Connected Embedded Systems: Assignment 1

This assignment focuses on interfacing a DS3231 RTC with an embedded linux device - in my case, a Raspberry Pi 3 B+.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

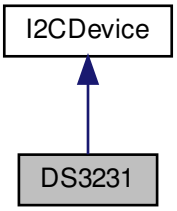
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

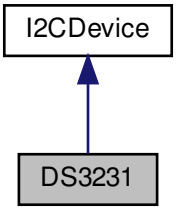
Class Documentation

4.1 DS3231 Class Reference

Inheritance diagram for DS3231:



Collaboration diagram for DS3231:



Public Member Functions

- **DS3231** (char)
- **DS3231** (bool)
- void sysTimeRTCInit ()
- void readTimeAndDate ()
- void writeTime (char *)
- void writeDate (char *)
- void readAlarms ()
- void setAlarms (char *)
- void readTemp ()
- void setInterrupt (bool, bool)
- void sqWaveGen (int, bool)
- void clearAlarms ()

Additional Inherited Members

4.1.1 Member Function Documentation

4.1.1.1 clearAlarms()

```
void DS3231::clearAlarms ( )
```

Clear the RTC Alarm Flags

Clears the Alarm 1 and Alarm 2 Flag bits of register 0x0F.

4.1.1.2 readAlarms()

```
void DS3231::readAlarms ( )
```

Print the RTC Alarms

Prints the values stored in the Alarm 1 and Alarm 2 registers (0x07 - 0x0D).

4.1.1.3 readTemp()

```
void DS3231::readTemp ( )
```

Print the RTC temperature

Temperature has a precision of 0.25 degrees. The upper part of the temperature is in reg 1 The lower part of the temperature is in tbe upper two bits of reg 2

4.1.1.4 readTimeAndDate()

```
void DS3231::readTimeAndDate ( )
```

Print the RTC Time and Date

Prints the values stored in registers 2, 1, and 0 (Hours, Minutes, Seconds), followed by registers 3, 4, 5, and 6 (Day, Date, Month, Year).

4.1.1.5 setAlarms()

```
void DS3231::setAlarms (
            char * times )
```

Set the RTC Alarm Times/Dates

Writes to the Alarm registers (0x07 - 0x0D). Sets all "Alarm Mask Bits" to 0 - Alarm will trigger when date, hours, minutes and seconds match

4.1.1.6 setInterrupt()

```
void DS3231::setInterrupt (
            bool alarm1,
            bool alarm2 )
```

Set the RTC Alarm Interrupt

Takes two boolean values representing the status of alarms 1 and 2. If either are to be set, the interrupt is set, along with the corresponding alarm enable bit. If neither are set, the interrupt bit is cleared, i.e. the Square Wave Generator is enabled.

4.1.1.7 sqWaveGen()

```
void DS3231::sqWaveGen (
            int freqOption,
            bool stopInterrupt )
```

Enable the Square Wave Generator at the desired frequency

Takes an integer value representing one of the four available frequencies of the square wave generator. A boolean represents the toggling of the interrupt bit, which enables the generator is set low. Each case of the switch statement performs the required bit modifications for the "Rate Select" bits of register 0x0E.

4.1.1.8 sysTimeRTCInit()

```
void DS3231::sysTimeRTCInit ( )
```

Sets the RTC time based on the output of the Linux "date" utility.

This function creates a temporary file, runs the command line "date" utility - formatting to give time (sec min hr) and date (day, date, month, year), and stores the output in the temporary file. This file is read into a string vector and the temporary file is closed and deleted. The time values and date values are added to separate char arrays for writing to the RTC.

4.1.1.9 writeDate()

```
void DS3231::writeDate (
            char * date )
```

Write the input dates to the RTC date registers

Takes a character array containing "date" values to be written. Calls the "writeToReg" function to write the required value to the required registers. Defined values are used to reference the day, date, month. and year registers (0x03, 0x04, 0x05 and 0x06). A check is performed to ensure the date is within the correct range

4.1.1.10 writeTime()

```
void DS3231::writeTime (
            char * times )
```

Write the input times to the RTC time registers

Takes a character array containing "time" values to be written. Calls the "writeToReg" function to write the required value to the required registers. Defined values are used to reference the second, minute and hour registers (0x00, 0x01, and 0x02).

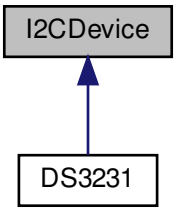
The documentation for this class was generated from the following files:

- DS3231.h
- DS3231.cpp

4.2 I2CDevice Class Reference

```
#include <I2CDevice.h>
```

Inheritance diagram for I2CDevice:



Public Member Functions

- **I2CDevice** (int)
- **I2CDevice** (int, char)

Protected Member Functions

- int openBus ()
- void connectDevice ()
- void resetReadAddr ()
- char * readBuffer ()
- void writeToReg (unsigned int, char)
- void setupProc ()

Protected Attributes

- int **file**
- std::vector< char > **buf**
- char **addr**

4.2.1 Detailed Description

I2CDevice Class

4.2.2 Member Function Documentation

4.2.2.1 connectDevice()

```
void I2CDevice::connectDevice ( )  [protected]
```

Open the connection to the I2C Device

4.2.2.2 openBus()

```
int I2CDevice::openBus ( )  [protected]
```

Open the Bus connection to the I2C Device

4.2.2.3 readBuffer()

```
char * I2CDevice::readBuffer ( )  [protected]
```

Reads from the I2C devices registers

4.2.2.4 resetReadAddr()

```
void I2CDevice::resetReadAddr ( )  [protected]
```

Set the initial read address for the I2C Device

4.2.2.5 setupProc()

```
void I2CDevice::setupProc ( )  [protected]
```

Setup Procedure for I2C Devices

Calls the functions required to open an I2C connection to a device, and read the buffers of this device.

4.2.2.6 writeToReg()

```
void I2CDevice::writeToReg (
            unsigned int reg,
            char val )  [protected]
```

Write to the I2C Devices registers

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

- I2CDevice.h
- I2CDevice.cpp

Index