

SODA Arudino library

Generated by Doxygen 1.8.6

Sun Mar 15 2015 09:47:44

Contents

1	The SODA library for datalogging with Olympia Circuit's SODA HE board.	1
1.1	Introduction	1
1.2	Dependencies	1
1.3	Installation	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	SODA Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Function Documentation	8
4.1.2.1	adcRead	8
4.1.2.2	begin	8
4.1.2.3	blinks	8
4.1.2.4	bufferTime	8
4.1.2.5	checkTime	8
4.1.2.6	communicate	9
4.1.2.7	dataDownload	9
4.1.2.8	dataLineAdd	9
4.1.2.9	dataLineAdd	9
4.1.2.10	dataLineAdd	9
4.1.2.11	dataLineAddBytes	10
4.1.2.12	dataLineBegin	10
4.1.2.13	dataLineEnd	10
4.1.2.14	getClockTemp	11
4.1.2.15	getID	11
4.1.2.16	getStandby	11
4.1.2.17	getTime	11

4.1.2.18	printBuffer	11
4.1.2.19	serialSetTime	11
4.1.2.20	setID	11
4.1.2.21	setStandby	12
4.1.2.22	setTime	12
4.1.2.23	setWake	12
4.1.2.24	smoothAnalogRead	12
4.1.2.25	tcReadK	12
4.1.2.26	turnOff	13
4.1.2.27	updateTime	13
4.1.2.28	usbConnected	13
5	File Documentation	15
5.1	C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.cpp File Reference	15
5.1.1	Variable Documentation	15
5.1.1.1	buffer	15
5.1.1.2	bufferIndex	15
5.1.1.3	card	15
5.1.1.4	end_on_connect	15
5.1.1.5	file	15
5.1.1.6	single_file	15
5.1.1.7	timeArray	15
5.2	C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.h File Reference	15
5.2.1	Macro Definition Documentation	16
5.2.1.1	ADC_12BITS	16
5.2.1.2	ADC_14BITS	16
5.2.1.3	ADC_16BITS	16
5.2.1.4	ADC_18BITS	16
5.2.1.5	ADC_BASE	16
5.2.1.6	ADC_CH1	16
5.2.1.7	ADC_CH2	16
5.2.1.8	ADC_CH3	16
5.2.1.9	ADC_CH4	16
5.2.1.10	ADC_CONTROL	17
5.2.1.11	ADC_GAIN1	17
5.2.1.12	ADC_GAIN2	17
5.2.1.13	ADC_GAIN4	17
5.2.1.14	ADC_GAIN8	17
5.2.1.15	ADC_I2C_ADDR	17
5.2.1.16	CLOCK_ALARM1_ADDR	17

5.2.1.17	CLOCK_ALARM_STATUS	17
5.2.1.18	CLOCK_CONTROL_ADDR	17
5.2.1.19	CLOCK_I2C_ADDR	17
5.2.1.20	CLOCK_SETUP	17
5.2.1.21	CLOCK_TEMPERATURE_ADDR	17
5.2.1.22	CLOCK_TIME_CAL_ADDR	17
5.2.1.23	FILENAME	17
5.2.1.24	LEDPIN	17

Chapter 1

The SODA library for datalogging with Olympia Circuit's SODA HE board.

1.1 Introduction

SODA is an Arduino library for data logging. It provides functions to simplify datalogging tasks using Olympia Circuit's **SODA** HE 1.0 board. **SODA** stands for Simple Open Data Acquisition. The goal of the project is to provide simple, high quality tools for the collection and analysis of environmental data. The library was built by Peter Gould (peter@olympiacircuits.com). Some code was adapted from Petre Rodan's DS3231 library for Arduino. Additional thanks go to William Greiman for his SD_FAT library.

The **SODA** library consists of a single class **SODA**.

1.2 Dependencies

SdFat: library for SD card functions. This library needs to be added to your Arduino library along with **SODA**.

EEPROM: standard Arduino library for EEPROM functions (comes with your Arduino installation).

Wire: standard Arduino library for I2C communication (comes with your Arduino installation).

1.3 Installation

The contents of the **SODA** folder should be added to the library folder of your Arduino installation (e.g., C:\Program Files (x86)\Arduino\libraries). Arduino must be restarted after the library has been added.

Chapter 2

Class Index

2.1 Class List

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

SODA	7
--------------------------------	-------------------

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.cpp	15
C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.h	15

Chapter 4

Class Documentation

4.1 SODA Class Reference

```
#include <SODA.h>
```

Public Member Functions

- void [begin](#) ()
- void [updateTime](#) (int val, int place)
- int [checkTime](#) (int place)
- void [setTime](#) ()
- void [getTime](#) ()
- void [bufferTime](#) ()
- void [serialSetTime](#) ()
- float [getClockTemp](#) ()
- void [setWake](#) (int val, int valType)
- void [turnOff](#) ()
- void [setStandby](#) (unsigned char val)
- int [getStandby](#) ()
- long [adcRead](#) (int ch, int bit, int gain)
- int [tcReadK](#) (int ch)
- int [smoothAnalogRead](#) (int pin1)
- void [dataLineBegin](#) (boolean binary=false, boolean set_end_on_connect=false, boolean set_single_file=true, int sd_cs_pin=17)
- void [dataLineAdd](#) (int value)
- void [dataLineAdd](#) (long value)
- void [dataLineAdd](#) (float value)
- void [dataLineAddBytes](#) (const void *[buffer](#), int nbytes)
- void [dataLineEnd](#) ()
- void [dataDownload](#) ()
- void [communicate](#) ()
- void [setID](#) (long ID)
- long [getID](#) ()
- void [printBuffer](#) ()
- bool [usbConnected](#) ()
- void [blinks](#) (int n)

4.1.1 Detailed Description

A class to handle basic datalogging functions using Olympia Circuit's [SODA](#) HE 1.0 Arduino-compatible board.

4.1.2 Member Function Documentation

4.1.2.1 `long SODA::adcRead (int ch, int bit, int gain)`

Returns a reading from the MCP3424 18-bit analog-digital converter.

Parameters

<i>ch</i>	an int argument specifying channel 1, 2, 3, or 4.
<i>bit</i>	an int argument specifying the bit encoding 1 = 12, 2=14, 3=16, 4=18 bits.
<i>gain</i>	as in argument specifying the level of gain from the programmable gain amplifier 1 = x1, 2 = x2, 3 = x4, 4 = x8. The return value is adjusted for gain so that a signal of 100 nV with gain = 4 will return a reading of 100 nV.

Returns

long ADC value in nanoVolts ($1 * 10^{-9}$ volts).

4.1.2.2 `void SODA::begin ()`

Initializes an instance of the [SODA](#) class. Should be called in each sketch before any other [SODA](#) functions.

4.1.2.3 `void SODA::blinks (int n)`

Blinks the led connected to pin 13. Used for simple communications such as to show when a process is under way or finished.

Parameters

<i>n</i>	an int that sets the number of times to blink. Each blink = 100 ms on, 100 ms off.
----------	--

4.1.2.4 `void SODA::bufferTime ()`

Loads the timeArray into a formatted character buffer. Format = YYYY-MM-DD HH:MM:SS

See Also

[getTime\(\)](#)

4.1.2.5 `int SODA::checkTime (int place)`

Returns an int value from the timeArray.

Parameters

<i>place</i>	an int specifying the place in the time array (0 = year, 1 = month, 2 = day, 3 = hour, 4 = minute, 5 = second).
--------------	---

See Also

[setTime\(\)](#)

[updateTime\(\)](#)

4.1.2.6 void SODA::communicate ()

Handles communication between the [SODA](#) and a computer/tablet through the serial monitor. Commands are sent through a serial connection using the format [XMORE_INFO] where X is a one-character command and MORE_INFO are optional, additional characters used to complete some commands such as setting the clock. current commands are: [D] downloads the logger file on the sd card

[I] return the logger_id

[R] runs through the sketch and normally outputs a line of current readings to the serial connection.

[t] prints the current clock time to the serial connection

[YYYY-MM-DD HH:MM:SS] sets the clock time

See Also

[dataLineBegin\(\)](#)
[dataLineAdd\(\)](#)
[dataLineEnd\(\)](#)

set standby so the clock will be reset properly

4.1.2.7 void SODA::dataDownload ()

Reads the contents of the data file from the SD card and streams it through the serial connection. The name of the file is set using #define filename definition at the top of [SODA.h](#).

See Also

[dataLineBegin\(\)](#)
[dataLineAdd\(\)](#)
[dataLineEnd\(\)](#)

4.1.2.8 void SODA::dataLineAdd (int value)

Adds an int value to the current data line. A comma is placed before the value. param value an int value

See Also

[dataLineBegin\(\)](#)
[dataLineEnd\(\)](#)

4.1.2.9 void SODA::dataLineAdd (long value)

Adds a long value to the current data line. A comma is placed before the value.

Parameters

<i>value</i>	an int value
--------------	--------------

See Also

[dataLineBegin\(\)](#)
[dataLineEnd\(\)](#)

4.1.2.10 void SODA::dataLineAdd (float value)

Adds a float value to the current data line. A comma is placed before the value.

Parameters

<i>value</i>	an int value
--------------	--------------

See Also

[dataLineBegin\(\)](#)
[dataLineEnd\(\)](#)

4.1.2.11 void SODA::dataLineAddBytes (const void * *buffer*, int *nbytes*)

Adds a series of bytes located at *buffer* and of length *nbytes* to the current data line. No comma is placed before the value.

Parameters

<i>buffer</i>	as const void*
<i>nbytes</i>	as int value

See Also

[dataLineAdd\(\)](#)

4.1.2.12 void SODA::dataLineBegin (boolean *binary* = false, boolean *set_end_on_connect* = false, boolean *set_single_file* = true, int *sd_cs_pin* = 17)

Begins a new dataline and writes the loggerid and current time separated by a comma. The clock is read by the functions so there's no need to make a separate call to [getTime\(\)](#). The function typically begins writing the line to the SD card. If the USB cable is connected it instead writes to the serial monitor. If writing to the SD card, the file is opened and left open until a call to [dataLineEnd\(\)](#).

Parameters

<i>binary</i>	a boolean indicating whether to write file in binary mode
<i>set_end_on_connect</i>	a boolean indicating whether the file should be closed when a USB connection is detected
<i>set_single_file</i>	a boolean indicating whether to save data to a single file or to create a new file each time.
<i>sd_cs_pin</i>	an int value for the chip-select pin for the SD card. Pin 17 is the default for the normal build.

See Also

[dataLineAdd\(\)](#)
[dataLineEnd\(\)](#)
[getID\(\)](#)
[getTime\(\)](#)
[setID\(\)](#)

4.1.2.13 void SODA::dataLineEnd ()

Terminates a data line. Adds a carriage return/line feed to the end of the data line and, if writing to the SD card, then closes the file.

See Also

[dataLineBegin\(\)](#)
[dataLineAdd\(\)](#)

4.1.2.14 float SODA::getClockTemp ()

Returns the value from the internal temperature sensor in the DS3231 real time clock.

Returns

temperature in Celsius as float

4.1.2.15 long SODA::getID ()

Returns the logger ID stored in microcontroller's EEPROM.

Returns

ID a long integer.

4.1.2.16 int SODA::getStandby ()

Retrieves the standby variable that's used to indicate whether the logger is in logging or communication mode.

Returns

standby as unsigned char

See Also

[setStandby](#)

4.1.2.17 void SODA::getTime ()

Loads the time from the clock to the timeArray. (0 = year, 1 = month, 2 = day, 3 = hour, 4 = minute, 5 = second).

See Also

[setTime\(\)](#)

4.1.2.18 void SODA::printBuffer ()

Prints the contents of the buffer[] array, usually a formatted time stamp.

4.1.2.19 void SODA::serialSetTime ()

Set the clock based on input from the Serial connection. Serial data are first saved to the buffer[] array and then loaded to the timeArray before being sent to the clock. Serial data format = 'YYYY-MM-DD HH:MM:SS'.

See Also

[setTime\(\)](#)

4.1.2.20 void SODA::setID (long ID)

Writes an logger ID number as a long value to the microcontroller's EEPROM (address 0 to 3).

Parameters

<i>ID</i>	a long value to be used as the logger ID
-----------	--

4.1.2.21 void SODA::setStandby (unsigned char *val*)

Sets the standby variable to indicate whether the logger is in logging or communication mode.

Parameters

<i>val</i>	unsigned char
------------	---------------

See Also

[getStandby](#)

4.1.2.22 void SODA::setTime ()

Resets the time in the clock to the values from timeArray.

See Also

[checkTime\(\)](#)
[getTime\(\)](#)
[updateTime\(\)](#)

4.1.2.23 void SODA::setWake (int *val*, int *valType*)

Sets the clock alarm. Used to wake up the logger and begin a new measurement. Example: setWake(10,2); sets the alarm to the next 10 minute interval.

Parameters

<i>val</i>	an int time value.
<i>valType</i>	an int indicating the units of time 1= secs, 2 = mins, 3=hours.

See Also

[turnOff](#)

4.1.2.24 int SODA::smoothAnalogRead (int *pin1*)

An improved version of analogRead that reduces noise in the measurement

Parameters

<i>pin1</i>	pin number to make reading
-------------	----------------------------

Returns

an int value of the average reading (between 0 and 1023).

4.1.2.25 int SODA::tcReadK (int *ch*)

Returns a temperature reading from a type K themocouple.

Parameters

<i>ch</i>	an int argument specifying ADC channel 1,2,3, or 4
-----------	--

Returns

a int value of temperature in degrees C. Int is used instead of a float since the precision of the measurement cannot realistically support decimal numbers.

4.1.2.26 void SODA::turnOff ()

Turns of the datalogger board by resetting the clock alarm pin, thereby shutting off the voltage regulator.

See Also

[setWake\(\)](#)

4.1.2.27 void SODA::updateTime (int *val*, int *place*)

Updates the time array. Need to run setTime to send time array to clock.

Parameters

<i>val</i>	an int time values.
<i>place</i>	an int specifying the place in the time array (0 = year, 1 = month, 2 = day, 3 = hour, 4 = minute, 5 = second).

See Also

[setTime\(\)](#)
[getTime\(\)](#)

4.1.2.28 bool SODA::usbConnected ()

Tests to see if the USB is connected. A USB connection causes pin 0 of the microcontroller to read as a digital high.

Returns

boolean values where connected = true, not connected = false.

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

- C:/Users/Peter/Documents/Arduino/libraries/SODA/[SODA.h](#)
- C:/Users/Peter/Documents/Arduino/libraries/SODA/[SODA.cpp](#)

Chapter 5

File Documentation

5.1 C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.cpp File Reference

```
#include "Arduino.h"  
#include <EEPROM.h>  
#include "SODA.h"  
#include <SdFat.h>
```

Variables

- SdFat [card](#)
- SdFile [file](#)
- int [timeArray](#) [6] = {2010,1,1,12,0,0}
- char [buffer](#) [30]
- int [bufferIndex](#) = 0
- boolean [end_on_connect](#) = false
- boolean [single_file](#) = false

5.1.1 Variable Documentation

5.1.1.1 char [buffer](#)[30]

5.1.1.2 int [bufferIndex](#) = 0

5.1.1.3 SdFat [card](#)

5.1.1.4 boolean [end_on_connect](#) = false

5.1.1.5 SdFile [file](#)

5.1.1.6 boolean [single_file](#) = false

5.1.1.7 int [timeArray](#)[6] = {2010,1,1,12,0,0}

5.2 C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.h File Reference

```
#include <Arduino.h>  
#include <Wire.h>
```

Classes

- class [SODA](#)

Macros

- #define [FILENAME](#) "DATA.CSV"
- #define [CLOCK_I2C_ADDR](#) 0x68
- #define [CLOCK_CONTROL_ADDR](#) 0x0E
- #define [CLOCK_TEMPERATURE_ADDR](#) 0x11
- #define [CLOCK_TIME_CAL_ADDR](#) 0x00
- #define [CLOCK_SETUP](#) 0x5
- #define [CLOCK_ALARM1_ADDR](#) 0x07
- #define [CLOCK_ALARM_STATUS](#) 0x0F
- #define [ADC_I2C_ADDR](#) 0x6E
- #define [ADC_CONTROL](#) 0x00
- #define [ADC_BASE](#) 0x80
- #define [ADC_CH1](#) 0x00
- #define [ADC_CH2](#) 0x20
- #define [ADC_CH3](#) 0x40
- #define [ADC_CH4](#) 0x60
- #define [ADC_18BITS](#) 0x0C
- #define [ADC_16BITS](#) 0x08
- #define [ADC_14BITS](#) 0x04
- #define [ADC_12BITS](#) 0x00
- #define [ADC_GAIN1](#) 0x00
- #define [ADC_GAIN2](#) 0x01
- #define [ADC_GAIN4](#) 0x02
- #define [ADC_GAIN8](#) 0x03
- #define [LEDPIN](#) 13

5.2.1 Macro Definition Documentation

5.2.1.1 #define [ADC_12BITS](#) 0x00

5.2.1.2 #define [ADC_14BITS](#) 0x04

5.2.1.3 #define [ADC_16BITS](#) 0x08

5.2.1.4 #define [ADC_18BITS](#) 0x0C

5.2.1.5 #define [ADC_BASE](#) 0x80

5.2.1.6 #define [ADC_CH1](#) 0x00

5.2.1.7 #define [ADC_CH2](#) 0x20

5.2.1.8 #define [ADC_CH3](#) 0x40

5.2.1.9 #define [ADC_CH4](#) 0x60

5.2.1.10 `#define ADC_CONTROL 0x00`

5.2.1.11 `#define ADC_GAIN1 0x00`

5.2.1.12 `#define ADC_GAIN2 0x01`

5.2.1.13 `#define ADC_GAIN4 0x02`

5.2.1.14 `#define ADC_GAIN8 0x03`

5.2.1.15 `#define ADC_I2C_ADDR 0x6E`

5.2.1.16 `#define CLOCK_ALARM1_ADDR 0x07`

5.2.1.17 `#define CLOCK_ALARM_STATUS 0x0F`

5.2.1.18 `#define CLOCK_CONTROL_ADDR 0x0E`

5.2.1.19 `#define CLOCK_I2C_ADDR 0x68`

5.2.1.20 `#define CLOCK_SETUP 0x5`

5.2.1.21 `#define CLOCK_TEMPERATURE_ADDR 0x11`

5.2.1.22 `#define CLOCK_TIME_CAL_ADDR 0x00`

5.2.1.23 `#define FILENAME "DATA.CSV"`

5.2.1.24 `#define LEDPIN 13`

Index

ADC_12BITS
SODA.h, 16

ADC_14BITS
SODA.h, 16

ADC_16BITS
SODA.h, 16

ADC_18BITS
SODA.h, 16

ADC_BASE
SODA.h, 16

ADC_CH1
SODA.h, 16

ADC_CH2
SODA.h, 16

ADC_CH3
SODA.h, 16

ADC_CH4
SODA.h, 16

ADC_CONTROL
SODA.h, 16

ADC_GAIN1
SODA.h, 17

ADC_GAIN2
SODA.h, 17

ADC_GAIN4
SODA.h, 17

ADC_GAIN8
SODA.h, 17

ADC_I2C_ADDR
SODA.h, 17

adcRead
SODA, 8

begin
SODA, 8

blinks
SODA, 8

buffer
SODA.cpp, 15

bufferIndex
SODA.cpp, 15

bufferTime
SODA, 8

C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.cpp, 15

C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.h, 15

CLOCK_ALARM1_ADDR
SODA.h, 17

CLOCK_ALARM_STATUS
SODA.h, 17

CLOCK_CONTROL_ADDR
SODA.h, 17

CLOCK_I2C_ADDR
SODA.h, 17

CLOCK_SETUP
SODA.h, 17

card
SODA.cpp, 15

checkTime
SODA, 8

communicate
SODA, 8

dataDownload
SODA, 9

dataLineAdd
SODA, 9

dataLineAddBytes
SODA, 10

dataLineBegin
SODA, 10

dataLineEnd
SODA, 10

end_on_connect
SODA.cpp, 15

FILENAME
SODA.h, 17

file
SODA.cpp, 15

getClockTemp
SODA, 10

getID
SODA, 11

getStandby
SODA, 11

getTime
SODA, 11

LEDPIN
SODA.h, 17

printBuffer
SODA, 11

SODA, 7

adcRead, 8

- begin, [8](#)
- blinks, [8](#)
- bufferTime, [8](#)
- checkTime, [8](#)
- communicate, [8](#)
- dataDownload, [9](#)
- dataLineAdd, [9](#)
- dataLineAddBytes, [10](#)
- dataLineBegin, [10](#)
- dataLineEnd, [10](#)
- getClockTemp, [10](#)
- getID, [11](#)
- getStandby, [11](#)
- getTime, [11](#)
- printBuffer, [11](#)
- serialSetTime, [11](#)
- setID, [11](#)
- setStandby, [12](#)
- setTime, [12](#)
- setWake, [12](#)
- smoothAnalogRead, [12](#)
- tcReadK, [12](#)
- turnOff, [13](#)
- updateTime, [13](#)
- usbConnected, [13](#)
- SODA.cpp
 - buffer, [15](#)
 - bufferIndex, [15](#)
 - card, [15](#)
 - end_on_connect, [15](#)
 - file, [15](#)
 - single_file, [15](#)
 - timeArray, [15](#)
- SODA.h
 - ADC_12BITS, [16](#)
 - ADC_14BITS, [16](#)
 - ADC_16BITS, [16](#)
 - ADC_18BITS, [16](#)
 - ADC_BASE, [16](#)
 - ADC_CH1, [16](#)
 - ADC_CH2, [16](#)
 - ADC_CH3, [16](#)
 - ADC_CH4, [16](#)
 - ADC_CONTROL, [16](#)
 - ADC_GAIN1, [17](#)
 - ADC_GAIN2, [17](#)
 - ADC_GAIN4, [17](#)
 - ADC_GAIN8, [17](#)
 - ADC_I2C_ADDR, [17](#)
 - CLOCK_ALARM1_ADDR, [17](#)
 - CLOCK_I2C_ADDR, [17](#)
 - CLOCK_SETUP, [17](#)
 - FILENAME, [17](#)
 - LEDPIN, [17](#)
- serialSetTime
 - SODA, [11](#)
- setID
 - SODA, [11](#)
- setStandby
 - SODA, [12](#)
- setTime
 - SODA, [12](#)
- setWake
 - SODA, [12](#)
- single_file
 - SODA.cpp, [15](#)
- smoothAnalogRead
 - SODA, [12](#)
- tcReadK
 - SODA, [12](#)
- timeArray
 - SODA.cpp, [15](#)
- turnOff
 - SODA, [13](#)
- updateTime
 - SODA, [13](#)
- usbConnected
 - SODA, [13](#)