

## SODA Arudino library

Generated by Doxygen 1.8.6

Mon Jun 9 2014 10:40:09



# Contents

<b>1</b>	<b>The SODA library for datalogging with Olympia Circuit's SODA HE board.</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Dependencies . . . . .	1
1.3	Installation . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
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 . . . . .	9
4.1.2.3	blinks . . . . .	9
4.1.2.4	bufferTime . . . . .	9
4.1.2.5	checkTime . . . . .	9
4.1.2.6	communicate . . . . .	9
4.1.2.7	dataDownload . . . . .	10
4.1.2.8	dataLineAdd . . . . .	10
4.1.2.9	dataLineAdd . . . . .	10
4.1.2.10	dataLineAdd . . . . .	10
4.1.2.11	dataLineBegin . . . . .	11
4.1.2.12	dataLineEnd . . . . .	11
4.1.2.13	getClockTemp . . . . .	11
4.1.2.14	getID . . . . .	11
4.1.2.15	getStandby . . . . .	11
4.1.2.16	getTime . . . . .	12
4.1.2.17	printBuffer . . . . .	12

4.1.2.18	<a href="#">serialSetTime</a>	12
4.1.2.19	<a href="#">setID</a>	12
4.1.2.20	<a href="#">setStandby</a>	12
4.1.2.21	<a href="#">setTime</a>	13
4.1.2.22	<a href="#">setWake</a>	13
4.1.2.23	<a href="#">smoothAnalogRead</a>	13
4.1.2.24	<a href="#">turnOff</a>	13
4.1.2.25	<a href="#">updateTime</a>	13
4.1.2.26	<a href="#">usbConnected</a>	14
<b>5</b>	<b>File Documentation</b>	<b>15</b>
5.1	<a href="#">C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.cpp File Reference</a>	15
5.1.1	Variable Documentation	15
5.1.1.1	<a href="#">buffer</a>	15
5.1.1.2	<a href="#">bufferIndex</a>	15
5.1.1.3	<a href="#">card</a>	15
5.1.1.4	<a href="#">file</a>	15
5.1.1.5	<a href="#">timeArray</a>	15
5.2	<a href="#">C:/Users/Peter/Documents/Arduino/libraries/SODA/SODA.h File Reference</a>	15
5.2.1	Macro Definition Documentation	16
5.2.1.1	<a href="#">ADC_12BITS</a>	16
5.2.1.2	<a href="#">ADC_14BITS</a>	16
5.2.1.3	<a href="#">ADC_16BITS</a>	16
5.2.1.4	<a href="#">ADC_18BITS</a>	16
5.2.1.5	<a href="#">ADC_BASE</a>	16
5.2.1.6	<a href="#">ADC_CH1</a>	16
5.2.1.7	<a href="#">ADC_CH2</a>	16
5.2.1.8	<a href="#">ADC_CH3</a>	16
5.2.1.9	<a href="#">ADC_CH4</a>	16
5.2.1.10	<a href="#">ADC_CONTROL</a>	16
5.2.1.11	<a href="#">ADC_GAIN1</a>	16
5.2.1.12	<a href="#">ADC_GAIN2</a>	16
5.2.1.13	<a href="#">ADC_GAIN4</a>	16
5.2.1.14	<a href="#">ADC_GAIN8</a>	17
5.2.1.15	<a href="#">ADC_I2C_ADDR</a>	17
5.2.1.16	<a href="#">CLOCK_ALARM1_ADDR</a>	17
5.2.1.17	<a href="#">CLOCK_ALARM_STATUS</a>	17
5.2.1.18	<a href="#">CLOCK_CONTROL_ADDR</a>	17
5.2.1.19	<a href="#">CLOCK_I2C_ADDR</a>	17
5.2.1.20	<a href="#">CLOCK_SETUP</a>	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
<b>Index</b>		<b>18</b>



## 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](mailto: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:

<a href="#">SODA</a> . . . . .	<a href="#">7</a>
--------------------------------	-------------------



## 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

- long [adcRead](#) (int ch, int bit, int gain)
- void [begin](#) ()
- void [blinks](#) (int n)
- void [bufferTime](#) ()
- int [checkTime](#) (int place)
- void [communicate](#) ()
- void [dataDownload](#) ()
- void [dataLineAdd](#) (int value)
- void [dataLineAdd](#) (long value)
- void [dataLineAdd](#) (float value)
- void [dataLineBegin](#) ()
- void [dataLineEnd](#) ()
- float [getClockTemp](#) ()
- long [getID](#) ()
- int [getStandby](#) ()
- void [getTime](#) ()
- void [printBuffer](#) ()
- void [serialSetTime](#) ()
- void [setID](#) (long ID)
- void [setStandby](#) (unsigned char val)
- void [setTime](#) ()
- void [setWake](#) (int val, int valType)
- int [smoothAnalogRead](#) (int pin1)
- int [tcReadK](#) (int ch)
- void [turnOff](#) ()
- void [updateTime](#) (int val, int place)
- bool [usbConnected](#) ()

#### 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::dataLineBegin ( )**

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\(\)](#).

## See Also

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

**4.1.2.12 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\(\)](#)

**4.1.2.13 float SODA::getClockTemp ( )**

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

## Returns

temperature in Celsius as float

**4.1.2.14 long SODA::getID ( )**

Returns the logger ID stored in microcontroller's EEPROM.

## Returns

ID a long integer.



**4.1.2.15** `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](#)

**4.1.2.16** `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\(\)](#)

**4.1.2.17** `void SODA::printBuffer ( )`

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

**4.1.2.18** `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\(\)](#)

**4.1.2.19** `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
-----------	--

**4.1.2.20** `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](#)

#### 4.1.2.21 void SODA::setTime ( )

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

See Also

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

#### 4.1.2.22 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](#)

#### 4.1.2.23 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).

#### 4.1.2.24 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.

#### 4.1.2.25 void SODA::turnOff ( )

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

See Also

[setWake\(\)](#)

4.1.2.26 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\(\)](#)

## 4.1.2.27 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

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

#### 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 SdFile [file](#)

5.1.1.5 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 ADC_12BITS 0x00`
- `#define ADC_14BITS 0x04`
- `#define ADC_16BITS 0x08`
- `#define ADC_18BITS 0x0C`
- `#define ADC_BASE 0x80`
- `#define ADC_CH1 0x00`
- `#define ADC_CH2 0x20`
- `#define ADC_CH3 0x40`
- `#define ADC_CH4 0x60`
- `#define ADC_CONTROL 0x00`
- `#define ADC_GAIN1 0x00`
- `#define ADC_GAIN2 0x01`
- `#define ADC_GAIN4 0x02`
- `#define ADC_GAIN8 0x03`
- `#define ADC_I2C_ADDR 0x6E`
- `#define CLOCK_ALARM1_ADDR 0x07`
- `#define CLOCK_ALARM_STATUS 0x0F`
- `#define CLOCK_CONTROL_ADDR 0x0E`
- `#define CLOCK_I2C_ADDR 0x68`
- `#define CLOCK_SETUP 0x5`
- `#define CLOCK_TEMPERATURE_ADDR 0x11`
- `#define CLOCK_TIME_CAL_ADDR 0x00`
- `#define FILENAME "DATA.CSV"`
- `#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, [16](#)

ADC\_GAIN2  
SODA.h, [16](#)

ADC\_GAIN4  
SODA.h, [16](#)

ADC\_GAIN8  
SODA.h, [16](#)

ADC\_I2C\_ADDR  
SODA.h, [17](#)

adcRead  
SODA, [8](#)

begin  
SODA, [9](#)

blinks  
SODA, [9](#)

buffer  
SODA.cpp, [15](#)

bufferIndex  
SODA.cpp, [15](#)

bufferTime  
SODA, [9](#)

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, [9](#)

communicate  
SODA, [9](#)

dataDownload  
SODA, [10](#)

dataLineAdd  
SODA, [10](#)

dataLineBegin  
SODA, [11](#)

dataLineEnd  
SODA, [11](#)

FILENAME  
SODA.h, [17](#)

file  
SODA.cpp, [15](#)

getClockTemp  
SODA, [11](#)

getID  
SODA, [11](#)

getStandby  
SODA, [11](#)

getTime  
SODA, [12](#)

LEDPIN  
SODA.h, [17](#)

printBuffer  
SODA, [12](#)

SODA, [7](#)

adcRead, [8](#)

begin, [9](#)

blinks, [9](#)

bufferTime, [9](#)

checkTime, [9](#)

communicate, [9](#)



- dataDownload, [10](#)
- dataLineAdd, [10](#)
- dataLineBegin, [11](#)
- dataLineEnd, [11](#)
- getClockTemp, [11](#)
- getID, [11](#)
- getStandby, [11](#)
- getTime, [12](#)
- printBuffer, [12](#)
- serialSetTime, [12](#)
- setID, [12](#)
- setStandby, [12](#)
- setTime, [12](#)
- setWake, [13](#)
- smoothAnalogRead, [13](#)
- turnOff, [13](#)
- updateTime, [13](#)
- usbConnected, [14](#)
- SODA.cpp
  - buffer, [15](#)
  - bufferIndex, [15](#)
  - card, [15](#)
  - 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, [16](#)
  - ADC\_GAIN2, [16](#)
  - ADC\_GAIN4, [16](#)
  - ADC\_GAIN8, [16](#)
  - ADC\_I2C\_ADDR, [17](#)
  - CLOCK\_ALARM1\_ADDR, [17](#)
  - CLOCK\_I2C\_ADDR, [17](#)
  - CLOCK\_SETUP, [17](#)
  - FILENAME, [17](#)
  - LEDPIN, [17](#)
- serialSetTime
  - SODA, [12](#)
- setID
  - SODA, [12](#)
- setStandby
  - SODA, [12](#)
- setTime
  - SODA, [12](#)
- setWake
  - SODA, [13](#)
- smoothAnalogRead
  - SODA, [13](#)
- timeArray
  - SODA.cpp, [15](#)
  - turnOff
    - SODA, [13](#)
  - updateTime
    - SODA, [13](#)
  - usbConnected
    - SODA, [14](#)