

MAXREFDES117# Code Documentation

V01.00

Generated by Doxygen 1.8.2

Wed Jun 22 2016 14:49:28

Contents

1	Main Page	1
1.1	Introduction	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	Adafruit_NeoPixel Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	Adafruit_NeoPixel	8
4.1.2.2	Adafruit_NeoPixel	8
4.1.2.3	~Adafruit_NeoPixel	8
4.1.3	Member Function Documentation	8
4.1.3.1	begin	8
4.1.3.2	canShow	8
4.1.3.3	clear	8
4.1.3.4	Color	8
4.1.3.5	Color	8
4.1.3.6	getBrightness	8
4.1.3.7	getPixelColor	8
4.1.3.8	getPixels	9
4.1.3.9	numPixels	9
4.1.3.10	setBrightness	9
4.1.3.11	setPin	9
4.1.3.12	setPixelColor	9
4.1.3.13	setPixelColor	9

4.1.3.14	setPixelColor	9
4.1.3.15	show	9
4.1.3.16	updateLength	9
4.1.3.17	updateType	9
5	File Documentation	11
5.1	RD117_ARDUINO/Adafruit_NeoPixel.cpp File Reference	11
5.2	RD117_ARDUINO/Adafruit_NeoPixel.h File Reference	11
5.2.1	Macro Definition Documentation	12
5.2.1.1	NEO_BGR	12
5.2.1.2	NEO_BGRW	12
5.2.1.3	NEO_BGWR	12
5.2.1.4	NEO_BRG	12
5.2.1.5	NEO_BRGW	12
5.2.1.6	NEO_BRWG	12
5.2.1.7	NEO_BWGR	13
5.2.1.8	NEO_BWRG	13
5.2.1.9	NEO_GBR	13
5.2.1.10	NEO_GBRW	13
5.2.1.11	NEO_GBWR	13
5.2.1.12	NEO_GRB	13
5.2.1.13	NEO_GRBW	13
5.2.1.14	NEO_GRWB	13
5.2.1.15	NEO_GWBR	13
5.2.1.16	NEO_GWRB	13
5.2.1.17	NEO_KHZ400	13
5.2.1.18	NEO_KHZ800	14
5.2.1.19	NEO_RBG	14
5.2.1.20	NEO_RBGW	14
5.2.1.21	NEO_RBWG	14
5.2.1.22	NEO_RGB	14
5.2.1.23	NEO_RGBW	14
5.2.1.24	NEO_RGWB	14
5.2.1.25	NEO_RWBG	14
5.2.1.26	NEO_RWGB	14
5.2.1.27	NEO_WBGR	14
5.2.1.28	NEO_WBRG	14

5.2.1.29	NEO_WGBR	15
5.2.1.30	NEO_WGRB	15
5.2.1.31	NEO_WRBG	15
5.2.1.32	NEO_WRGB	15
5.2.2	Typedef Documentation	15
5.2.2.1	neoPixelType	15
5.3	RD117_ARDUINO/algorithm.cpp File Reference	15
5.3.1	Detailed Description	16
5.3.2	Function Documentation	16
5.3.2.1	maxim_find_peaks	16
5.3.2.2	maxim_heart_rate_and_oxygen_saturation	16
5.3.2.3	maxim_peaks_above_min_height	17
5.3.2.4	maxim_remove_close_peaks	17
5.3.2.5	maxim_sort_ascend	17
5.3.2.6	maxim_sort_indices_descend	18
5.4	RD117_ARDUINO/algorithm.h File Reference	18
5.4.1	Detailed Description	19
5.4.2	Macro Definition Documentation	19
5.4.2.1	BUFFER_SIZE	19
5.4.2.2	false	19
5.4.2.3	FS	19
5.4.2.4	MA4_SIZE	19
5.4.2.5	min	20
5.4.2.6	true	20
5.4.3	Function Documentation	20
5.4.3.1	maxim_find_peaks	20
5.4.3.2	maxim_heart_rate_and_oxygen_saturation	20
5.4.3.3	maxim_peaks_above_min_height	21
5.4.3.4	maxim_remove_close_peaks	21
5.4.3.5	maxim_sort_ascend	21
5.4.3.6	maxim_sort_indices_descend	22
5.4.4	Variable Documentation	22
5.4.4.1	uch_spo2_table	22
5.5	RD117_ARDUINO/max30102.cpp File Reference	22
5.5.1	Detailed Description	23
5.5.2	Function Documentation	23
5.5.2.1	maxim_max30102_init	23

5.5.2.2	maxim_max30102_read_fifo	23
5.5.2.3	maxim_max30102_read_reg	24
5.5.2.4	maxim_max30102_reset	24
5.5.2.5	maxim_max30102_write_reg	24
5.6	RD117_ARDUINO/max30102.h File Reference	25
5.6.1	Detailed Description	26
5.6.2	Macro Definition Documentation	26
5.6.2.1	I2C_READ_ADDR	26
5.6.2.2	I2C_WRITE_ADDR	26
5.6.2.3	REG_FIFO_CONFIG	26
5.6.2.4	REG_FIFO_DATA	26
5.6.2.5	REG_FIFO_RD_PTR	27
5.6.2.6	REG_FIFO_WR_PTR	27
5.6.2.7	REG_INTR_ENABLE_1	27
5.6.2.8	REG_INTR_ENABLE_2	27
5.6.2.9	REG_INTR_STATUS_1	27
5.6.2.10	REG_INTR_STATUS_2	27
5.6.2.11	REG_LED1_PA	27
5.6.2.12	REG_LED2_PA	27
5.6.2.13	REG_MODE_CONFIG	27
5.6.2.14	REG_MULTI_LED_CTRL1	27
5.6.2.15	REG_MULTI_LED_CTRL2	27
5.6.2.16	REG_OVF_COUNTER	28
5.6.2.17	REG_PART_ID	28
5.6.2.18	REG_PILOT_PA	28
5.6.2.19	REG_PROX_INT_THRESH	28
5.6.2.20	REG_REV_ID	28
5.6.2.21	REG_SPO2_CONFIG	28
5.6.2.22	REG_TEMP_CONFIG	28
5.6.2.23	REG_TEMP_FRAC	28
5.6.2.24	REG_TEMP_INTR	28
5.6.3	Function Documentation	28
5.6.3.1	maxim_max30102_init	28
5.6.3.2	maxim_max30102_read_fifo	29
5.6.3.3	maxim_max30102_read_reg	29
5.6.3.4	maxim_max30102_reset	29
5.6.3.5	maxim_max30102_write_reg	30

5.7	RD117_ARDUINO/RD117_ARDUINO.ino File Reference	30
5.7.1	Macro Definition Documentation	31
5.7.1.1	MAX_BRIGHTNESS	31
5.7.2	Function Documentation	31
5.7.2.1	loop	31
5.7.2.2	setup	31
5.7.3	Variable Documentation	31
5.7.3.1	aun_ir_buffer	31
5.7.3.2	aun_red_buffer	31
5.7.3.3	ch_hr_valid	31
5.7.3.4	ch_spo2_valid	31
5.7.3.5	n_heart_rate	32
5.7.3.6	n_ir_buffer_length	32
5.7.3.7	n_spo2	32
5.7.3.8	uch_dummy	32
5.8	RD117_ARDUINO/SoftI2CMaster.h File Reference	32
5.8.1	Macro Definition Documentation	33
5.8.1.1	__tmp_reg__	33
5.8.1.2	_SOFTI2C_H	33
5.8.1.3	FAC	33
5.8.1.4	I2C_CPUFREQ	33
5.8.1.5	I2C_DELAY_COUNTER	33
5.8.1.6	I2C_FASTMODE	33
5.8.1.7	I2C_MAX_STRETCH	33
5.8.1.8	I2C_NOINTERRUPT	33
5.8.1.9	I2C_READ	34
5.8.1.10	I2C_SLOWMODE	34
5.8.1.11	I2C_TIMEOUT	34
5.8.1.12	I2C_TIMEOUT_DELAY_LOOPS	34
5.8.1.13	I2C_WRITE	34
5.8.1.14	SCL_DDR	34
5.8.1.15	SCL_IN	34
5.8.1.16	SCL_OUT	34
5.8.1.17	SDA_DDR	34
5.8.1.18	SDA_IN	34
5.8.1.19	SDA_OUT	34
5.8.1.20	SOFTI2DMASTER_H_	35

5.8.2	Function Documentation	35
5.8.2.1	__attribute__	35
5.8.2.2	i2c_delay_half	35
5.8.2.3	i2c_init	35
5.8.2.4	i2c_read	35
5.8.2.5	i2c_rep_start	35
5.8.2.6	i2c_start	35
5.8.2.7	i2c_start_wait	35
5.8.2.8	i2c_stop	35
5.8.2.9	i2c_wait_scl_high	35
5.8.2.10	i2c_write	35

Chapter 1

Main Page

1.1 Introduction

This is the code documentation for the MAXREFDES117# subsystem reference design.

The Files page contains the File List page and the Globals page.

The Globals page contains the Functions, Variables, and Macros sub-pages.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Adafruit_NeoPixel	7
---------------------------------------------	---

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

RD117_ARDUINO/ Adafruit_NeoPixel.cpp	11
RD117_ARDUINO/ Adafruit_NeoPixel.h	11
RD117_ARDUINO/ algorithm.cpp	15
RD117_ARDUINO/ algorithm.h	18
RD117_ARDUINO/ max30102.cpp	22
RD117_ARDUINO/ max30102.h	25
RD117_ARDUINO/ RD117_ARDUINO.ino	30
RD117_ARDUINO/ SoftI2CMaster.h	32

Chapter 4

Data Structure Documentation

4.1 Adafruit_NeoPixel Class Reference

```
#include <Adafruit_NeoPixel.h>
```

Public Member Functions

- [Adafruit_NeoPixel](#) (uint16_t n, uint8_t p=6, [neoPixelType](#) t=NEO_GRB+NEO_KHZ800)
- [Adafruit_NeoPixel](#) (void)
- [~Adafruit_NeoPixel](#) ()
- void [begin](#) (void)
- void [show](#) (void)
- void [setPin](#) (uint8_t p)
- void [setPixelColor](#) (uint16_t n, uint8_t r, uint8_t g, uint8_t b)
- void [setPixelColor](#) (uint16_t n, uint8_t r, uint8_t g, uint8_t b, uint8_t w)
- void [setPixelColor](#) (uint16_t n, uint32_t c)
- void [setBrightness](#) (uint8_t)
- void [clear](#) ()
- void [updateLength](#) (uint16_t n)
- void [updateType](#) ([neoPixelType](#) t)
- uint8_t * [getPixels](#) (void) const
- uint8_t [getBrightness](#) (void) const
- uint16_t [numPixels](#) (void) const
- static uint32_t [Color](#) (uint8_t r, uint8_t g, uint8_t b, uint8_t w)
- uint32_t [getPixelColor](#) (uint16_t n) const
- bool [canShow](#) (void)

Static Public Member Functions

- static uint32_t [Color](#) (uint8_t r, uint8_t g, uint8_t b)

4.1.1 Detailed Description

Definition at line 116 of file Adafruit_NeoPixel.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Adafruit_NeoPixel (uint16_t *n*, uint8_t *p* = 6, neoPixelType *t* = NEO_GRB + NEO_KHZ800)

Definition at line 38 of file Adafruit_NeoPixel.cpp.

4.1.2.2 Adafruit_NeoPixel (void)

Definition at line 51 of file Adafruit_NeoPixel.cpp.

4.1.2.3 ~Adafruit_NeoPixel ()

Definition at line 60 of file Adafruit_NeoPixel.cpp.

4.1.3 Member Function Documentation

4.1.3.1 void begin (void)

Definition at line 65 of file Adafruit_NeoPixel.cpp.

4.1.3.2 bool canShow (void) [inline]

Definition at line 147 of file Adafruit_NeoPixel.h.

4.1.3.3 void clear ()

Definition at line 1198 of file Adafruit_NeoPixel.cpp.

4.1.3.4 uint32_t Color (uint8_t *r*, uint8_t *g*, uint8_t *b*) [static]

Definition at line 1096 of file Adafruit_NeoPixel.cpp.

4.1.3.5 uint32_t Color (uint8_t *r*, uint8_t *g*, uint8_t *b*, uint8_t *w*)

Definition at line 1102 of file Adafruit_NeoPixel.cpp.

4.1.3.6 uint8_t getBrightness (void) const

Definition at line 1194 of file Adafruit_NeoPixel.cpp.

4.1.3.7 uint32_t getPixelColor (uint16_t *n*) const

Definition at line 1107 of file Adafruit_NeoPixel.cpp.

4.1.3.8 uint8_t * getPixels (void) const

Definition at line 1148 of file Adafruit_NeoPixel.cpp.

4.1.3.9 uint16_t numPixels (void) const

Definition at line 1152 of file Adafruit_NeoPixel.cpp.

4.1.3.10 void setBrightness (uint8_t b)

Definition at line 1168 of file Adafruit_NeoPixel.cpp.

4.1.3.11 void setPin (uint8_t p)

Definition at line 1031 of file Adafruit_NeoPixel.cpp.

4.1.3.12 void setPixelColor (uint16_t n, uint8_t r, uint8_t g, uint8_t b)

Definition at line 1047 of file Adafruit_NeoPixel.cpp.

4.1.3.13 void setPixelColor (uint16_t n, uint8_t r, uint8_t g, uint8_t b, uint8_t w)**4.1.3.14 void setPixelColor (uint16_t n, uint32_t c)**

Definition at line 1070 of file Adafruit_NeoPixel.cpp.

4.1.3.15 void show (void)

Definition at line 111 of file Adafruit_NeoPixel.cpp.

4.1.3.16 void updateLength (uint16_t n)

Definition at line 73 of file Adafruit_NeoPixel.cpp.

4.1.3.17 void updateType (neoPixelType t)

Definition at line 86 of file Adafruit_NeoPixel.cpp.

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

- RD117_ARDUINO/[Adafruit_NeoPixel.h](#)
- RD117_ARDUINO/[Adafruit_NeoPixel.cpp](#)

Chapter 5

File Documentation

5.1 RD117_ARDUINO/Adafruit_NeoPixel.cpp File Reference

```
#include "Adafruit_NeoPixel.h"
```

5.2 RD117_ARDUINO/Adafruit_NeoPixel.h File Reference

```
#include <WProgram.h>
#include <pins_arduino.h>
```

Data Structures

- class [Adafruit_NeoPixel](#)

Macros

- #define [NEO_RGB](#) ((0 << 6) | (0 << 4) | (1 << 2) | (2))
- #define [NEO_RBG](#) ((0 << 6) | (0 << 4) | (2 << 2) | (1))
- #define [NEO_GRB](#) ((1 << 6) | (1 << 4) | (0 << 2) | (2))
- #define [NEO_GBR](#) ((2 << 6) | (2 << 4) | (0 << 2) | (1))
- #define [NEO_BRG](#) ((1 << 6) | (1 << 4) | (2 << 2) | (0))
- #define [NEO_BGR](#) ((2 << 6) | (2 << 4) | (1 << 2) | (0))
- #define [NEO_WRGB](#) ((0 << 6) | (1 << 4) | (2 << 2) | (3))
- #define [NEO_WRBG](#) ((0 << 6) | (1 << 4) | (3 << 2) | (2))
- #define [NEO_WGRB](#) ((0 << 6) | (2 << 4) | (1 << 2) | (3))
- #define [NEO_WGBR](#) ((0 << 6) | (3 << 4) | (1 << 2) | (2))
- #define [NEO_WBRG](#) ((0 << 6) | (2 << 4) | (3 << 2) | (1))
- #define [NEO_WBGR](#) ((0 << 6) | (3 << 4) | (2 << 2) | (1))
- #define [NEO_RWGB](#) ((1 << 6) | (0 << 4) | (2 << 2) | (3))
- #define [NEO_RWBG](#) ((1 << 6) | (0 << 4) | (3 << 2) | (2))
- #define [NEO_RGWB](#) ((2 << 6) | (0 << 4) | (1 << 2) | (3))
- #define [NEO_RGBW](#) ((3 << 6) | (0 << 4) | (1 << 2) | (2))

- `#define NEO_RBWG ((2 << 6) | (0 << 4) | (3 << 2) | (1))`
- `#define NEO_RBGW ((3 << 6) | (0 << 4) | (2 << 2) | (1))`
- `#define NEO_GWRB ((1 << 6) | (2 << 4) | (0 << 2) | (3))`
- `#define NEO_GWBR ((1 << 6) | (3 << 4) | (0 << 2) | (2))`
- `#define NEO_GRWB ((2 << 6) | (1 << 4) | (0 << 2) | (3))`
- `#define NEO_GRBW ((3 << 6) | (1 << 4) | (0 << 2) | (2))`
- `#define NEO_GBWR ((2 << 6) | (3 << 4) | (0 << 2) | (1))`
- `#define NEO_GBRW ((3 << 6) | (2 << 4) | (0 << 2) | (1))`
- `#define NEO_BWRG ((1 << 6) | (2 << 4) | (3 << 2) | (0))`
- `#define NEO_BWGR ((1 << 6) | (3 << 4) | (2 << 2) | (0))`
- `#define NEO_BRWG ((2 << 6) | (1 << 4) | (3 << 2) | (0))`
- `#define NEO_BRGW ((3 << 6) | (1 << 4) | (2 << 2) | (0))`
- `#define NEO_BGWR ((2 << 6) | (3 << 4) | (1 << 2) | (0))`
- `#define NEO_BGRW ((3 << 6) | (2 << 4) | (1 << 2) | (0))`
- `#define NEO_KHZ800 0x0000`
- `#define NEO_KHZ400 0x0100`

Typedefs

- `typedef uint8_t neoPixelType`

5.2.1 Macro Definition Documentation

5.2.1.1 `#define NEO_BGR ((2 << 6) | (2 << 4) | (1 << 2) | (0))`

Definition at line 59 of file Adafruit_NeoPixel.h.

5.2.1.2 `#define NEO_BGRW ((3 << 6) | (2 << 4) | (1 << 2) | (0))`

Definition at line 89 of file Adafruit_NeoPixel.h.

5.2.1.3 `#define NEO_BGWR ((2 << 6) | (3 << 4) | (1 << 2) | (0))`

Definition at line 88 of file Adafruit_NeoPixel.h.

5.2.1.4 `#define NEO_BRG ((1 << 6) | (1 << 4) | (2 << 2) | (0))`

Definition at line 58 of file Adafruit_NeoPixel.h.

5.2.1.5 `#define NEO_BRGW ((3 << 6) | (1 << 4) | (2 << 2) | (0))`

Definition at line 87 of file Adafruit_NeoPixel.h.

5.2.1.6 `#define NEO_BRWG ((2 << 6) | (1 << 4) | (3 << 2) | (0))`

Definition at line 86 of file Adafruit_NeoPixel.h.

5.2.1.7 #define NEO_BWGR ((1 << 6) | (3 << 4) | (2 << 2) | (0))

Definition at line 85 of file Adafruit_NeoPixel.h.

5.2.1.8 #define NEO_BWRG ((1 << 6) | (2 << 4) | (3 << 2) | (0))

Definition at line 84 of file Adafruit_NeoPixel.h.

5.2.1.9 #define NEO_GBR ((2 << 6) | (2 << 4) | (0 << 2) | (1))

Definition at line 57 of file Adafruit_NeoPixel.h.

5.2.1.10 #define NEO_GBRW ((3 << 6) | (2 << 4) | (0 << 2) | (1))

Definition at line 82 of file Adafruit_NeoPixel.h.

5.2.1.11 #define NEO_GBWR ((2 << 6) | (3 << 4) | (0 << 2) | (1))

Definition at line 81 of file Adafruit_NeoPixel.h.

5.2.1.12 #define NEO_GRB ((1 << 6) | (1 << 4) | (0 << 2) | (2))

Definition at line 56 of file Adafruit_NeoPixel.h.

5.2.1.13 #define NEO_GRBW ((3 << 6) | (1 << 4) | (0 << 2) | (2))

Definition at line 80 of file Adafruit_NeoPixel.h.

5.2.1.14 #define NEO_GRWB ((2 << 6) | (1 << 4) | (0 << 2) | (3))

Definition at line 79 of file Adafruit_NeoPixel.h.

5.2.1.15 #define NEO_GWBR ((1 << 6) | (3 << 4) | (0 << 2) | (2))

Definition at line 78 of file Adafruit_NeoPixel.h.

5.2.1.16 #define NEO_GWRB ((1 << 6) | (2 << 4) | (0 << 2) | (3))

Definition at line 77 of file Adafruit_NeoPixel.h.

5.2.1.17 #define NEO_KHZ400 0x0100

Definition at line 102 of file Adafruit_NeoPixel.h.

5.2.1.18 #define NEO_KHZ800 0x0000

Definition at line 100 of file Adafruit_NeoPixel.h.

5.2.1.19 #define NEO_RGB ((0 << 6) | (0 << 4) | (2 << 2) | (1))

Definition at line 55 of file Adafruit_NeoPixel.h.

5.2.1.20 #define NEO_RGBW ((3 << 6) | (0 << 4) | (2 << 2) | (1))

Definition at line 75 of file Adafruit_NeoPixel.h.

5.2.1.21 #define NEO_RBWG ((2 << 6) | (0 << 4) | (3 << 2) | (1))

Definition at line 74 of file Adafruit_NeoPixel.h.

5.2.1.22 #define NEO_RGB ((0 << 6) | (0 << 4) | (1 << 2) | (2))

Definition at line 54 of file Adafruit_NeoPixel.h.

5.2.1.23 #define NEO_RGBW ((3 << 6) | (0 << 4) | (1 << 2) | (2))

Definition at line 73 of file Adafruit_NeoPixel.h.

5.2.1.24 #define NEO_RGWB ((2 << 6) | (0 << 4) | (1 << 2) | (3))

Definition at line 72 of file Adafruit_NeoPixel.h.

5.2.1.25 #define NEO_RWBG ((1 << 6) | (0 << 4) | (3 << 2) | (2))

Definition at line 71 of file Adafruit_NeoPixel.h.

5.2.1.26 #define NEO_RWGB ((1 << 6) | (0 << 4) | (2 << 2) | (3))

Definition at line 70 of file Adafruit_NeoPixel.h.

5.2.1.27 #define NEO_WBGR ((0 << 6) | (3 << 4) | (2 << 2) | (1))

Definition at line 68 of file Adafruit_NeoPixel.h.

5.2.1.28 #define NEO_WBRG ((0 << 6) | (2 << 4) | (3 << 2) | (1))

Definition at line 67 of file Adafruit_NeoPixel.h.

5.2.1.29 #define NEO_WGBR ((0 << 6) | (3 << 4) | (1 << 2) | (2))

Definition at line 66 of file Adafruit_NeoPixel.h.

5.2.1.30 #define NEO_WGRB ((0 << 6) | (2 << 4) | (1 << 2) | (3))

Definition at line 65 of file Adafruit_NeoPixel.h.

5.2.1.31 #define NEO_WRBG ((0 << 6) | (1 << 4) | (3 << 2) | (2))

Definition at line 64 of file Adafruit_NeoPixel.h.

5.2.1.32 #define NEO_WRGB ((0 << 6) | (1 << 4) | (2 << 2) | (3))

Definition at line 63 of file Adafruit_NeoPixel.h.

5.2.2 Typedef Documentation**5.2.2.1 typedef uint8_t neoPixelType**

Definition at line 111 of file Adafruit_NeoPixel.h.

5.3 RD117_ARDUINO/algorithm.cpp File Reference

```
#include "algorithm.h"
#include "arduino.h"
```

Functions

- void [maxim_heart_rate_and_oxygen_saturation](#) (uint32_t *pun_ir_buffer, int32_t [n_ir_buffer_length](#), uint32_t *pun_red_buffer, int32_t *pn_spo2, int8_t *pch_spo2_valid, int32_t *pn_heart_rate, int8_t *pch_hr_valid)
Calculate the heart rate and SpO2 level.
- void [maxim_find_peaks](#) (int32_t *pn_locs, int32_t *n_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)
Find peaks.
- void [maxim_peaks_above_min_height](#) (int32_t *pn_locs, int32_t *n_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height)
Find peaks above n_min_height.
- void [maxim_remove_close_peaks](#) (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_min_distance)
Remove peaks.
- void [maxim_sort_ascend](#) (int32_t *pn_x, int32_t n_size)
Sort array.
- void [maxim_sort_indices_descend](#) (int32_t *pn_x, int32_t *pn_indx, int32_t n_size)
Sort indices.

5.3.1 Detailed Description

Project: MAXREFDES117# Filename: [algorithm.cpp](#) Description: This module calculates the heart rate/SpO2 level

This code follows the following naming conventions:

```
char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an-
_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array)
auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t *
pn_pmod_value
```

Definition in file [algorithm.cpp](#).

5.3.2 Function Documentation

5.3.2.1 void maxim_find_peaks (int32_t * pn_locs, int32_t * n_npks, int32_t * pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)

Find peaks.

Details

Find at most MAX_NUM peaks above MIN_HEIGHT separated by at least MIN_DISTANCE

Return values

None

Definition at line 209 of file [algorithm.cpp](#).

5.3.2.2 void maxim_heart_rate_and_oxygen_saturation (uint32_t * pun_ir_buffer, int32_t n_ir_buffer_length, uint32_t * pun_red_buffer, int32_t * pn_spo2, int8_t * pch_spo2_valid, int32_t * pn_heart_rate, int8_t * pch_hr_valid)

Calculate the heart rate and SpO2 level.

Details

By detecting peaks of PPG cycle and corresponding AC/DC of red/infra-red signal, the an_ratio for the SPO2 is computed. Since this algorithm is aiming for Arm M0/M3. formula for SPO2 did not achieve the accuracy due to register overflow. Thus, accurate SPO2 is precalculated and save longo uch_spo2_table[] per each an_ratio.

Parameters

in	*pun_ir_buffer	- IR sensor data buffer
in	n_ir_buffer_length	- IR sensor data buffer length
in	*pun_red_buffer	- Red sensor data buffer
out	*pn_spo2	- Calculated SpO2 value
out	*pch_spo2_valid	- 1 if the calculated SpO2 value is valid
out	*pn_heart_rate	- Calculated heart rate value
out	*pch_hr_valid	- 1 if the calculated heart rate value is valid

Return values

None

Definition at line 69 of file algorithm.cpp.

5.3.2.3 void maxim_peaks_above_min_height (int32_t * *pn_locs*, int32_t * *n_npk*s, int32_t * *pn_x*, int32_t *n_size*, int32_t *n_min_height*)

Find peaks above *n_min_height*.

Details

Find all peaks above MIN_HEIGHT

Return values

None

Definition at line 223 of file algorithm.cpp.

5.3.2.4 void maxim_remove_close_peaks (int32_t * *pn_locs*, int32_t * *pn_npk*s, int32_t * *pn_x*, int32_t *n_min_distance*)

Remove peaks.

Details

Remove peaks separated by less than MIN_DISTANCE

Return values

None

Definition at line 253 of file algorithm.cpp.

5.3.2.5 void maxim_sort_ascend (int32_t * *pn_x*, int32_t *n_size*)

Sort array.

Details

Sort array in ascending order (insertion sort algorithm)

Return values

None

Definition at line 282 of file algorithm.cpp.

5.3.2.6 void maxim_sort_indices_descend (int32_t * pn_x, int32_t * pn_indx, int32_t n_size)

Sort indices.

Details

Sort indices according to descending order (insertion sort algorithm)

Return values

None

Definition at line 300 of file algorithm.cpp.

5.4 RD117_ARDUINO/algorithm.h File Reference

```
#include <arduino.h>
```

Macros

- #define [true](#) 1
- #define [false](#) 0
- #define [FS](#) 25
- #define [BUFFER_SIZE](#) ([FS](#)* 4)
- #define [MA4_SIZE](#) 4
- #define [min](#)(x, y) ((x) < (y) ? (x) : (y))

Functions

- void [maxim_heart_rate_and_oxygen_saturation](#) (uint32_t *pun_ir_buffer, int32_t [n_ir_buffer_length](#), uint32_t *pun_red_buffer, int32_t *pn_spo2, int8_t *pch_spo2_valid, int32_t *pn_heart_rate, int8_t *pch_hr_valid)
Calculate the heart rate and SpO2 level.
- void [maxim_find_peaks](#) (int32_t *pn_locs, int32_t *n_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)
Find peaks.
- void [maxim_peaks_above_min_height](#) (int32_t *pn_locs, int32_t *n_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height)
Find peaks above n_min_height.
- void [maxim_remove_close_peaks](#) (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_min_distance)
Remove peaks.
- void [maxim_sort_ascend](#) (int32_t *pn_x, int32_t n_size)
Sort array.
- void [maxim_sort_indices_descend](#) (int32_t *pn_x, int32_t *pn_indx, int32_t n_size)
Sort indices.

Variables

- const uint8_t [uch_spo2_table](#) [184]

5.4.1 Detailed Description

Project: MAXREFDES117# Filename: [algorithm.h](#) Description: This module is the heart rate/SpO2 calculation algorithm header file

Revision History:

1-18-2016 Rev 01.00 SK Initial release.

This code follows the following naming conventions:

char ch_pmod_value

char (array) s_pmod_s_string[16]

float f_pmod_value

int32_t n_pmod_value

int32_t (array) an_pmod_value[16]

int16_t w_pmod_value

int16_t (array) aw_pmod_value[16]

uint16_t uw_pmod_value

uint16_t (array) auw_pmod_value[16]

uint8_t uch_pmod_value

uint8_t (array) auch_pmod_buffer[16]

uint32_t un_pmod_value

int32_t * pn_pmod_value

Definition in file [algorithm.h](#).

5.4.2 Macro Definition Documentation

5.4.2.1 #define BUFFER_SIZE (FS* 4)

Definition at line 69 of file [algorithm.h](#).

5.4.2.2 #define false 0

Definition at line 67 of file [algorithm.h](#).

5.4.2.3 #define FS 25

Definition at line 68 of file [algorithm.h](#).

5.4.2.4 #define MA4_SIZE 4

Definition at line 70 of file [algorithm.h](#).

5.4.2.5 **#define min(x, y) ((x) < (y) ? (x) : (y))**

Definition at line 71 of file algorithm.h.

5.4.2.6 **#define true 1**

Definition at line 66 of file algorithm.h.

5.4.3 Function Documentation

5.4.3.1 **void maxim_find_peaks (int32_t * pn_locs, int32_t * n_npks, int32_t * pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)**

Find peaks.

Details

Find at most MAX_NUM peaks above MIN_HEIGHT separated by at least MIN_DISTANCE

Return values

None

Definition at line 209 of file algorithm.cpp.

5.4.3.2 **void maxim_heart_rate_and_oxygen_saturation (uint32_t * pun_ir_buffer, int32_t n_ir_buffer_length, uint32_t * pun_red_buffer, int32_t * pn_spo2, int8_t * pch_spo2_valid, int32_t * pn_heart_rate, int8_t * pch_hr_valid)**

Calculate the heart rate and SpO2 level.

Details

By detecting peaks of PPG cycle and corresponding AC/DC of red/infra-red signal, the an_ratio for the SPO2 is computed. Since this algorithm is aiming for Arm M0/M3. formula for SPO2 did not achieve the accuracy due to register overflow. Thus, accurate SPO2 is precalculated and save longo uch_spo2_table[] per each an_ratio.

Parameters

in	*pun_ir_buffer	- IR sensor data buffer
in	n_ir_buffer_length	- IR sensor data buffer length
in	*pun_red_buffer	- Red sensor data buffer
out	*pn_spo2	- Calculated SpO2 value
out	*pch_spo2_valid	- 1 if the calculated SpO2 value is valid
out	*pn_heart_rate	- Calculated heart rate value
out	*pch_hr_valid	- 1 if the calculated heart rate value is valid

Return values

None

Definition at line 69 of file algorithm.cpp.

5.4.3.3 void maxim_peaks_above_min_height (int32_t * *pn_locs*, int32_t * *n_npk*s, int32_t * *pn_x*, int32_t *n_size*, int32_t *n_min_height*)

Find peaks above *n_min_height*.

Details

Find all peaks above MIN_HEIGHT

Return values

None

Definition at line 223 of file algorithm.cpp.

5.4.3.4 void maxim_remove_close_peaks (int32_t * *pn_locs*, int32_t * *pn_npk*s, int32_t * *pn_x*, int32_t *n_min_distance*)

Remove peaks.

Details

Remove peaks separated by less than MIN_DISTANCE

Return values

None

Definition at line 253 of file algorithm.cpp.

5.4.3.5 void maxim_sort_ascend (int32_t * *pn_x*, int32_t *n_size*)

Sort array.

Details

Sort array in ascending order (insertion sort algorithm)

Return values

None

Definition at line 282 of file algorithm.cpp.

5.4.3.6 void maxim_sort_indices_descend (int32_t * pn_x, int32_t * pn_idx, int32_t n_size)

Sort indices.

Details

Sort indices according to descending order (insertion sort algorithm)

Return values

None

Definition at line 300 of file algorithm.cpp.

5.4.4 Variable Documentation

5.4.4.1 const uint8_t uch_spo2_table[184]

Initial value:

```
= { 95, 95, 95, 96, 96, 96, 97, 97, 97, 97, 97, 98, 98, 98, 98, 98, 99, 99, 99,
    99,
    99, 99, 99, 99, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100,
    100, 100, 100, 100, 100, 100,
    100, 100, 100, 100, 99, 99, 99, 99, 99, 99, 99, 99, 98, 98, 98,
    98, 98, 98, 97, 97,
    97, 97, 96, 96, 96, 96, 95, 95, 95, 94, 94, 94, 93, 93, 93, 92,
    92, 92, 91, 91,
    90, 90, 89, 89, 89, 88, 88, 87, 87, 86, 86, 85, 85, 84, 84, 83,
    82, 82, 81, 81,
    80, 80, 79, 78, 78, 77, 76, 76, 75, 74, 74, 73, 72, 72, 71, 70,
    69, 69, 68, 67,
    66, 66, 65, 64, 63, 62, 62, 61, 60, 59, 58, 57, 56, 56, 55, 54,
    53, 52, 51, 50,
    49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34,
    33, 31, 30, 29,
    28, 27, 26, 25, 23, 22, 21, 20, 19, 17, 16, 15, 14, 12, 11, 10, 9
    , 7, 6, 5,
    3, 2, 1 }
```

Definition at line 74 of file algorithm.h.

5.5 RD117_ARDUINO/max30102.cpp File Reference

```
#include "max30102.h"
#include "SoftI2CMaster.h"
#include "algorithm.h"
```

Functions

- bool [maxim_max30102_write_reg](#) (uint8_t uch_addr, uint8_t uch_data)
Write a value to a MAX30102 register.
- bool [maxim_max30102_read_reg](#) (uint8_t uch_addr, uint8_t *puch_data)
Read a MAX30102 register.
- bool [maxim_max30102_init](#) ()

Initialize the MAX30102.

- bool [maxim_max30102_read_fifo](#) (uint32_t *pun_red_led, uint32_t *pun_ir_led)

Read a set of samples from the MAX30102 FIFO register.

- bool [maxim_max30102_reset](#) ()

Reset the MAX30102.

5.5.1 Detailed Description

Project: MAXREFDES117# Filename: [max30102.cpp](#) Description: This module is an embedded controller driver for the MAX30102

Revision History:

1-18-2016 Rev 01.00 GL Initial release.

This code follows the following naming conventions:

char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array) auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t * pn_pmod_value

Definition in file [max30102.cpp](#).

5.5.2 Function Documentation

5.5.2.1 bool maxim_max30102_init ()

Initialize the MAX30102.

Details

This function initializes the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 111 of file max30102.cpp.

5.5.2.2 bool maxim_max30102_read_fifo (uint32_t * pun_red_led, uint32_t * pun_ir_led)

Read a set of samples from the MAX30102 FIFO register.

Details

This function reads a set of samples from the MAX30102 FIFO register

Parameters

out	<i>*pun_red_led</i>	- pointer that stores the red LED reading data
out	<i>*pun_ir_led</i>	- pointer that stores the IR LED reading data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 154 of file max30102.cpp.

5.5.2.3 bool maxim_max30102_read_reg (uint8_t uch_addr, uint8_t * puch_data)

Read a MAX30102 register.

Details

This function reads a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
out	<i>puch_data</i>	- pointer that stores the register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 88 of file max30102.cpp.

5.5.2.4 bool maxim_max30102_reset (void)

Reset the MAX30102.

Details

This function resets the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 202 of file max30102.cpp.

5.5.2.5 bool maxim_max30102_write_reg (uint8_t uch_addr, uint8_t uch_data)

Write a value to a MAX30102 register.

Details

This function writes a value to a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
in	<i>uch_data</i>	- register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 66 of file max30102.cpp.

5.6 RD117_ARDUINO/max30102.h File Reference

```
#include <arduino.h>
```

Macros

- #define I2C_WRITE_ADDR 0xAE
- #define I2C_READ_ADDR 0xAF
- #define REG_INTR_STATUS_1 0x00
- #define REG_INTR_STATUS_2 0x01
- #define REG_INTR_ENABLE_1 0x02
- #define REG_INTR_ENABLE_2 0x03
- #define REG_FIFO_WR_PTR 0x04
- #define REG_OVF_COUNTER 0x05
- #define REG_FIFO_RD_PTR 0x06
- #define REG_FIFO_DATA 0x07
- #define REG_FIFO_CONFIG 0x08
- #define REG_MODE_CONFIG 0x09
- #define REG_SPO2_CONFIG 0x0A
- #define REG_LED1_PA 0x0C
- #define REG_LED2_PA 0x0D
- #define REG_PILOT_PA 0x10
- #define REG_MULTI_LED_CTRL1 0x11
- #define REG_MULTI_LED_CTRL2 0x12
- #define REG_TEMP_INTR 0x1F
- #define REG_TEMP_FRAC 0x20
- #define REG_TEMP_CONFIG 0x21
- #define REG_PROX_INT_THRESH 0x30
- #define REG_REV_ID 0xFE
- #define REG_PART_ID 0xFF

Functions

- bool [maxim_max30102_init](#) ()
Initialize the MAX30102.
- bool [maxim_max30102_read_fifo](#) (uint32_t *pun_red_led, uint32_t *pun_ir_led)
Read a set of samples from the MAX30102 FIFO register.
- bool [maxim_max30102_write_reg](#) (uint8_t uch_addr, uint8_t uch_data)
Write a value to a MAX30102 register.
- bool [maxim_max30102_read_reg](#) (uint8_t uch_addr, uint8_t *puch_data)
Read a MAX30102 register.
- bool [maxim_max30102_reset](#) (void)
Reset the MAX30102.

5.6.1 Detailed Description

Project: MAXREFDES117# Filename: [max30102.h](#) Description: This module is an embedded controller driver header file for MAX30102

Revision History:

1-18-2016 Rev 01.00 GL Initial release.

This code follows the following naming conventions:

char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array) auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t * pn_pmod_value

Definition in file [max30102.h](#).

5.6.2 Macro Definition Documentation

5.6.2.1 #define I2C_READ_ADDR 0xAF

Definition at line 67 of file max30102.h.

5.6.2.2 #define I2C_WRITE_ADDR 0xAE

Definition at line 66 of file max30102.h.

5.6.2.3 #define REG_FIFO_CONFIG 0x08

Definition at line 78 of file max30102.h.

5.6.2.4 #define REG_FIFO_DATA 0x07

Definition at line 77 of file max30102.h.

5.6.2.5 #define REG_FIFO_RD_PTR 0x06

Definition at line 76 of file max30102.h.

5.6.2.6 #define REG_FIFO_WR_PTR 0x04

Definition at line 74 of file max30102.h.

5.6.2.7 #define REG_INTR_ENABLE_1 0x02

Definition at line 72 of file max30102.h.

5.6.2.8 #define REG_INTR_ENABLE_2 0x03

Definition at line 73 of file max30102.h.

5.6.2.9 #define REG_INTR_STATUS_1 0x00

Definition at line 70 of file max30102.h.

5.6.2.10 #define REG_INTR_STATUS_2 0x01

Definition at line 71 of file max30102.h.

5.6.2.11 #define REG_LED1_PA 0x0C

Definition at line 81 of file max30102.h.

5.6.2.12 #define REG_LED2_PA 0x0D

Definition at line 82 of file max30102.h.

5.6.2.13 #define REG_MODE_CONFIG 0x09

Definition at line 79 of file max30102.h.

5.6.2.14 #define REG_MULTI_LED_CTRL1 0x11

Definition at line 84 of file max30102.h.

5.6.2.15 #define REG_MULTI_LED_CTRL2 0x12

Definition at line 85 of file max30102.h.

5.6.2.16 #define REG_OVF_COUNTER 0x05

Definition at line 75 of file max30102.h.

5.6.2.17 #define REG_PART_ID 0xFF

Definition at line 91 of file max30102.h.

5.6.2.18 #define REG_PILOT_PA 0x10

Definition at line 83 of file max30102.h.

5.6.2.19 #define REG_PROX_INT_THRESH 0x30

Definition at line 89 of file max30102.h.

5.6.2.20 #define REG_REV_ID 0xFE

Definition at line 90 of file max30102.h.

5.6.2.21 #define REG_SPO2_CONFIG 0x0A

Definition at line 80 of file max30102.h.

5.6.2.22 #define REG_TEMP_CONFIG 0x21

Definition at line 88 of file max30102.h.

5.6.2.23 #define REG_TEMP_FRAC 0x20

Definition at line 87 of file max30102.h.

5.6.2.24 #define REG_TEMP_INTR 0x1F

Definition at line 86 of file max30102.h.

5.6.3 Function Documentation**5.6.3.1 bool maxim_max30102_init ()**

Initialize the MAX30102.

Details

This function initializes the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 111 of file max30102.cpp.

5.6.3.2 bool maxim_max30102_read_fifo (uint32_t * *pun_red_led*, uint32_t * *pun_ir_led*)

Read a set of samples from the MAX30102 FIFO register.

Details

This function reads a set of samples from the MAX30102 FIFO register

Parameters

out	<i>*pun_red_led</i>	- pointer that stores the red LED reading data
out	<i>*pun_ir_led</i>	- pointer that stores the IR LED reading data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 154 of file max30102.cpp.

5.6.3.3 bool maxim_max30102_read_reg (uint8_t *uch_addr*, uint8_t * *puch_data*)

Read a MAX30102 register.

Details

This function reads a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
out	<i>puch_data</i>	- pointer that stores the register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 88 of file max30102.cpp.

5.6.3.4 bool maxim_max30102_reset (void)

Reset the MAX30102.

Details

This function resets the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 202 of file max30102.cpp.

5.6.3.5 bool maxim_max30102_write_reg (uint8_t uch_addr, uint8_t uch_data)

Write a value to a MAX30102 register.

Details

This function writes a value to a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
in	<i>uch_data</i>	- register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 66 of file max30102.cpp.

5.7 RD117_ARDUINO/RD117_ARDUINO.ino File Reference

```
#include <Arduino.h>
#include "algorithm.h"
#include "max30102.h"
```

Macros

- #define `MAX_BRIGHTNESS` 255

Functions

- void `setup` ()
- void `loop` ()

Variables

- uint32_t [aun_ir_buffer](#) [100]
- uint32_t [aun_red_buffer](#) [100]
- int32_t [n_ir_buffer_length](#)
- int32_t [n_spo2](#)
- int8_t [ch_spo2_valid](#)
- int32_t [n_heart_rate](#)
- int8_t [ch_hr_valid](#)
- uint8_t [uch_dummy](#)

5.7.1 Macro Definition Documentation

5.7.1.1 #define MAX_BRIGHTNESS 255

Definition at line 88 of file RD117_ARDUINO.ino.

5.7.2 Function Documentation

5.7.2.1 void loop ()

Definition at line 144 of file RD117_ARDUINO.ino.

5.7.2.2 void setup ()

Definition at line 108 of file RD117_ARDUINO.ino.

5.7.3 Variable Documentation

5.7.3.1 uint32_t aun_ir_buffer[100]

Definition at line 96 of file RD117_ARDUINO.ino.

5.7.3.2 uint32_t aun_red_buffer[100]

Definition at line 97 of file RD117_ARDUINO.ino.

5.7.3.3 int8_t ch_hr_valid

Definition at line 103 of file RD117_ARDUINO.ino.

5.7.3.4 int8_t ch_spo2_valid

Definition at line 101 of file RD117_ARDUINO.ino.

5.7.3.5 int32_t n_heart_rate

Definition at line 102 of file RD117_ARDUINO.ino.

5.7.3.6 int32_t n_ir_buffer_length

Definition at line 99 of file RD117_ARDUINO.ino.

5.7.3.7 int32_t n_spo2

Definition at line 100 of file RD117_ARDUINO.ino.

5.7.3.8 uint8_t uch_dummy

Definition at line 104 of file RD117_ARDUINO.ino.

5.8 RD117_ARDUINO/SoftI2CMaster.h File Reference

```
#include <avr/io.h>
#include <Arduino.h>
```

Macros

- #define [SOFTI2DMASTER_H_](#)
- #define [I2C_TIMEOUT](#) 100
- #define [I2C_NOINTERRUPT](#) 0
- #define [I2C_SLOWMODE](#) 1
- #define [FAC](#) 1
- #define [I2C_CPUFREQ](#) (F_CPU/[FAC](#))
- #define [_SOFTI2C_H](#) 1
- #define [I2C_FASTMODE](#) 0
- #define [I2C_TIMEOUT_DELAY_LOOPS](#) ([I2C_CPUFREQ](#)/1000UL)*[I2C_TIMEOUT](#)/4000UL
- #define [I2C_MAX_STRETCH](#) 1
- #define [I2C_DELAY_COUNTER](#) ((([I2C_CPUFREQ](#)/25000L)/2-19)/3)
- #define [I2C_READ](#) 1
- #define [I2C_WRITE](#) 0
- #define [SDA_DDR](#) (_SFR_IO_ADDR([SDA_PORT](#)) - 1)
- #define [SCL_DDR](#) (_SFR_IO_ADDR([SCL_PORT](#)) - 1)
- #define [SDA_OUT](#) _SFR_IO_ADDR([SDA_PORT](#))
- #define [SCL_OUT](#) _SFR_IO_ADDR([SCL_PORT](#))
- #define [SDA_IN](#) (_SFR_IO_ADDR([SDA_PORT](#)) - 2)
- #define [SCL_IN](#) (_SFR_IO_ADDR([SCL_PORT](#)) - 2)
- #define [__tmp_reg__](#) 0

Functions

- boolean `__attribute__((noinline)) i2c_init(void)`
- void `i2c_delay_half(void)`
- void `i2c_wait_scl_high(void)`
- boolean `i2c_init(void)`
- bool `i2c_start(uint8_t addr)`
- bool `i2c_rep_start(uint8_t addr)`
- void `i2c_start_wait(uint8_t addr)`
- void `i2c_stop(void)`
- bool `i2c_write(uint8_t value)`
- uint8_t `i2c_read(bool last)`

5.8.1 Macro Definition Documentation

5.8.1.1 `#define __tmp_reg__ 0`

Definition at line 199 of file SoftI2CMaster.h.

5.8.1.2 `#define _SOFTI2C_H 1`

Definition at line 85 of file SoftI2CMaster.h.

5.8.1.3 `#define FAC 1`

Definition at line 81 of file SoftI2CMaster.h.

5.8.1.4 `#define I2C_CPUFREQ (F_CPU/FAC)`

Definition at line 82 of file SoftI2CMaster.h.

5.8.1.5 `#define I2C_DELAY_COUNTER (((I2C_CPUFREQ/25000L)/2-19)/3)`

Definition at line 174 of file SoftI2CMaster.h.

5.8.1.6 `#define I2C_FASTMODE 0`

Definition at line 129 of file SoftI2CMaster.h.

5.8.1.7 `#define I2C_MAX_STRETCH 1`

Definition at line 161 of file SoftI2CMaster.h.

5.8.1.8 `#define I2C_NOINTERRUPT 0`

Definition at line 79 of file SoftI2CMaster.h.

5.8.1.9 #define I2C_READ 1

Definition at line 187 of file SoftI2CMaster.h.

5.8.1.10 #define I2C_SLOWMODE 1

Definition at line 80 of file SoftI2CMaster.h.

5.8.1.11 #define I2C_TIMEOUT 100

Definition at line 78 of file SoftI2CMaster.h.

5.8.1.12 #define I2C_TIMEOUT_DELAY_LOOPS (I2C_CPUFREQ/1000UL)*I2C_TIMEOUT/4000UL

Definition at line 159 of file SoftI2CMaster.h.

5.8.1.13 #define I2C_WRITE 0

Definition at line 188 of file SoftI2CMaster.h.

5.8.1.14 #define SCL_DDR (_SFR_IO_ADDR(SCL_PORT) - 1)

Definition at line 192 of file SoftI2CMaster.h.

5.8.1.15 #define SCL_IN (_SFR_IO_ADDR(SCL_PORT) - 2)

Definition at line 196 of file SoftI2CMaster.h.

5.8.1.16 #define SCL_OUT _SFR_IO_ADDR(SCL_PORT)

Definition at line 194 of file SoftI2CMaster.h.

5.8.1.17 #define SDA_DDR (_SFR_IO_ADDR(SDA_PORT) - 1)

Definition at line 191 of file SoftI2CMaster.h.

5.8.1.18 #define SDA_IN (_SFR_IO_ADDR(SDA_PORT) - 2)

Definition at line 195 of file SoftI2CMaster.h.

5.8.1.19 #define SDA_OUT _SFR_IO_ADDR(SDA_PORT)

Definition at line 193 of file SoftI2CMaster.h.

5.8.1.20 #define SOFTI2DMASTER_H_

Definition at line 59 of file SoftI2CMaster.h.

5.8.2 Function Documentation**5.8.2.1 void __attribute__ ((noinline))****5.8.2.2 void i2c_delay_half (void)**

Definition at line 207 of file SoftI2CMaster.h.

5.8.2.3 boolean i2c_init (void)

Definition at line 272 of file SoftI2CMaster.h.

5.8.2.4 uint8_t i2c_read (bool *last*)

Definition at line 462 of file SoftI2CMaster.h.

5.8.2.5 bool i2c_rep_start (uint8_t *addr*)

Definition at line 313 of file SoftI2CMaster.h.

5.8.2.6 bool i2c_start (uint8_t *addr*)

Definition at line 295 of file SoftI2CMaster.h.

5.8.2.7 void i2c_start_wait (uint8_t *addr*)

Definition at line 338 of file SoftI2CMaster.h.

5.8.2.8 void i2c_stop (void)

Definition at line 365 of file SoftI2CMaster.h.

5.8.2.9 void i2c_wait_scl_high (void)

Definition at line 225 of file SoftI2CMaster.h.

5.8.2.10 bool i2c_write (uint8_t *value*)

Definition at line 385 of file SoftI2CMaster.h.

Index

- ~Adafruit_NeoPixel
 - Adafruit_NeoPixel, [8](#)
- _SOFTI2C_H
 - SoftI2CMaster.h, [33](#)
- __attribute__
 - SoftI2CMaster.h, [35](#)
- __tmp_reg__
 - SoftI2CMaster.h, [33](#)
- Adafruit_NeoPixel, [7](#)
 - ~Adafruit_NeoPixel, [8](#)
 - Adafruit_NeoPixel, [8](#)
 - Adafruit_NeoPixel, [8](#)
 - begin, [8](#)
 - canShow, [8](#)
 - clear, [8](#)
 - Color, [8](#)
 - getBrightness, [8](#)
 - getPixelColor, [8](#)
 - getPixels, [8](#)
 - numPixels, [9](#)
 - setBrightness, [9](#)
 - setPin, [9](#)
 - setPixelColor, [9](#)
 - show, [9](#)
 - updateLength, [9](#)
 - updateType, [9](#)
- Adafruit_NeoPixel.h
 - NEO_BGR, [12](#)
 - NEO_BGRW, [12](#)
 - NEO_BGWR, [12](#)
 - NEO_BRG, [12](#)
 - NEO_BRGW, [12](#)
 - NEO_BRWG, [12](#)
 - NEO_BWGR, [12](#)
 - NEO_BWRG, [13](#)
 - NEO_GBR, [13](#)
 - NEO_GBRW, [13](#)
 - NEO_GBWR, [13](#)
 - NEO_GRB, [13](#)
 - NEO_GRBW, [13](#)
 - NEO_GRWB, [13](#)
 - NEO_GWBR, [13](#)
 - NEO_GWRB, [13](#)
 - NEO_KHZ400, [13](#)
 - NEO_KHZ800, [13](#)
 - NEO_RGB, [14](#)
 - NEO_RGBW, [14](#)
 - NEO_RBGW, [14](#)
 - NEO_RGBW, [14](#)
 - NEO_RGBW, [14](#)
 - NEO_RGWB, [14](#)
 - NEO_RWBG, [14](#)
 - NEO_RWGB, [14](#)
 - NEO_WBGR, [14](#)
 - NEO_WBRG, [14](#)
 - NEO_WGBR, [14](#)
 - NEO_WGRB, [15](#)
 - NEO_WRBG, [15](#)
 - NEO_WRGB, [15](#)
 - neoPixelType, [15](#)
- algorithm.cpp
 - maxim_find_peaks, [16](#)
 - maxim_heart_rate_and_oxygen_saturation, [16](#)
 - maxim_peaks_above_min_height, [17](#)
 - maxim_remove_close_peaks, [17](#)
 - maxim_sort_ascend, [17](#)
 - maxim_sort_indices_descend, [17](#)
- algorithm.h
 - BUFFER_SIZE, [19](#)
 - FS, [19](#)
 - false, [19](#)
 - MA4_SIZE, [19](#)
 - maxim_find_peaks, [20](#)
 - maxim_heart_rate_and_oxygen_saturation, [20](#)
 - maxim_peaks_above_min_height, [21](#)
 - maxim_remove_close_peaks, [21](#)
 - maxim_sort_ascend, [21](#)
 - maxim_sort_indices_descend, [21](#)
 - min, [19](#)
 - true, [20](#)
 - uch_spo2_table, [22](#)
- aun_ir_buffer
 - RD117_ARDUINO.ino, [31](#)
- aun_red_buffer
 - RD117_ARDUINO.ino, [31](#)
- BUFFER_SIZE
 - algorithm.h, [19](#)
- begin
 - Adafruit_NeoPixel, [8](#)

- canShow
 - Adafruit_NeoPixel, 8
- ch_hr_valid
 - RD117_ARDUINO.ino, 31
- ch_spo2_valid
 - RD117_ARDUINO.ino, 31
- clear
 - Adafruit_NeoPixel, 8
- Color
 - Adafruit_NeoPixel, 8
- FAC
 - SoftI2CMaster.h, 33
- FS
 - algorithm.h, 19
- false
 - algorithm.h, 19
- getBrightness
 - Adafruit_NeoPixel, 8
- getPixelColor
 - Adafruit_NeoPixel, 8
- getPixels
 - Adafruit_NeoPixel, 8
- I2C_CPUFREQ
 - SoftI2CMaster.h, 33
- I2C_DELAY_COUNTER
 - SoftI2CMaster.h, 33
- I2C_FASTMODE
 - SoftI2CMaster.h, 33
- I2C_MAX_STRETCH
 - SoftI2CMaster.h, 33
- I2C_NOINTERRUPT
 - SoftI2CMaster.h, 33
- I2C_READ
 - SoftI2CMaster.h, 33
- I2C_READ_ADDR
 - max30102.h, 26
- I2C_SLOWMODE
 - SoftI2CMaster.h, 34
- I2C_TIMEOUT
 - SoftI2CMaster.h, 34
- I2C_WRITE
 - SoftI2CMaster.h, 34
- I2C_WRITE_ADDR
 - max30102.h, 26
- i2c_delay_half
 - SoftI2CMaster.h, 35
- i2c_init
 - SoftI2CMaster.h, 35
- i2c_read
 - SoftI2CMaster.h, 35
- i2c_rep_start
 - SoftI2CMaster.h, 35
- i2c_start
 - SoftI2CMaster.h, 35
- i2c_start_wait
 - SoftI2CMaster.h, 35
- i2c_stop
 - SoftI2CMaster.h, 35
- i2c_wait_scl_high
 - SoftI2CMaster.h, 35
- i2c_write
 - SoftI2CMaster.h, 35
- loop
 - RD117_ARDUINO.ino, 31
- MA4_SIZE
 - algorithm.h, 19
- MAX_BRIGHTNESS
 - RD117_ARDUINO.ino, 31
- max30102.cpp
 - maxim_max30102_init, 23
 - maxim_max30102_read_fifo, 23
 - maxim_max30102_read_reg, 24
 - maxim_max30102_reset, 24
 - maxim_max30102_write_reg, 24
- max30102.h
 - I2C_READ_ADDR, 26
 - I2C_WRITE_ADDR, 26
 - maxim_max30102_init, 28
 - maxim_max30102_read_fifo, 29
 - maxim_max30102_read_reg, 29
 - maxim_max30102_reset, 29
 - maxim_max30102_write_reg, 30
 - REG_FIFO_CONFIG, 26
 - REG_FIFO_DATA, 26
 - REG_FIFO_RD_PTR, 26
 - REG_FIFO_WR_PTR, 27
 - REG_INTR_ENABLE_1, 27
 - REG_INTR_ENABLE_2, 27
 - REG_INTR_STATUS_1, 27
 - REG_INTR_STATUS_2, 27
 - REG_LED1_PA, 27
 - REG_LED2_PA, 27
 - REG_MODE_CONFIG, 27
 - REG_MULTI_LED_CTRL1, 27
 - REG_MULTI_LED_CTRL2, 27
 - REG_OVF_COUNTER, 27
 - REG_PART_ID, 28
 - REG_PILOT_PA, 28
 - REG_REV_ID, 28
 - REG_SPO2_CONFIG, 28
 - REG_TEMP_CONFIG, 28
 - REG_TEMP_FRAC, 28
 - REG_TEMP_INTR, 28
- maxim_find_peaks
 - algorithm.cpp, 16

algorithm.h, 20
 maxim_heart_rate_and_oxygen_saturation
 algorithm.cpp, 16
 algorithm.h, 20
 maxim_max30102_init
 max30102.cpp, 23
 max30102.h, 28
 maxim_max30102_read_fifo
 max30102.cpp, 23
 max30102.h, 29
 maxim_max30102_read_reg
 max30102.cpp, 24
 max30102.h, 29
 maxim_max30102_reset
 max30102.cpp, 24
 max30102.h, 29
 maxim_max30102_write_reg
 max30102.cpp, 24
 max30102.h, 30
 maxim_peaks_above_min_height
 algorithm.cpp, 17
 algorithm.h, 21
 maxim_remove_close_peaks
 algorithm.cpp, 17
 algorithm.h, 21
 maxim_sort_ascend
 algorithm.cpp, 17
 algorithm.h, 21
 maxim_sort_indices_descend
 algorithm.cpp, 17
 algorithm.h, 21
 min
 algorithm.h, 19

 n_heart_rate
 RD117_ARDUINO.ino, 31
 n_ir_buffer_length
 RD117_ARDUINO.ino, 32
 n_spo2
 RD117_ARDUINO.ino, 32
 NEO_BGR
 Adafruit_NeoPixel.h, 12
 NEO_BGRW
 Adafruit_NeoPixel.h, 12
 NEO_BGWR
 Adafruit_NeoPixel.h, 12
 NEO_BRG
 Adafruit_NeoPixel.h, 12
 NEO_BRGW
 Adafruit_NeoPixel.h, 12
 NEO_BRWG
 Adafruit_NeoPixel.h, 12
 NEO_BWGR
 Adafruit_NeoPixel.h, 12

NEO_BWRG
 Adafruit_NeoPixel.h, 13
 NEO_GBR
 Adafruit_NeoPixel.h, 13
 NEO_GBRW
 Adafruit_NeoPixel.h, 13
 NEO_GBWR
 Adafruit_NeoPixel.h, 13
 NEO_GRB
 Adafruit_NeoPixel.h, 13
 NEO_GRBW
 Adafruit_NeoPixel.h, 13
 NEO_GRWB
 Adafruit_NeoPixel.h, 13
 NEO_GWBR
 Adafruit_NeoPixel.h, 13
 NEO_GWRB
 Adafruit_NeoPixel.h, 13
 NEO_KHZ400
 Adafruit_NeoPixel.h, 13
 NEO_KHZ800
 Adafruit_NeoPixel.h, 13
 NEO_RBG
 Adafruit_NeoPixel.h, 14
 NEO_RBGW
 Adafruit_NeoPixel.h, 14
 NEO_RBWG
 Adafruit_NeoPixel.h, 14
 NEO_RGB
 Adafruit_NeoPixel.h, 14
 NEO_RGBW
 Adafruit_NeoPixel.h, 14
 NEO_RGWB
 Adafruit_NeoPixel.h, 14
 NEO_RWBG
 Adafruit_NeoPixel.h, 14
 NEO_RWGB
 Adafruit_NeoPixel.h, 14
 NEO_WBGR
 Adafruit_NeoPixel.h, 14
 NEO_WBRG
 Adafruit_NeoPixel.h, 14
 NEO_WGBR
 Adafruit_NeoPixel.h, 14
 NEO_WGRB
 Adafruit_NeoPixel.h, 15
 NEO_WRBG
 Adafruit_NeoPixel.h, 15
 NEO_WRGB
 Adafruit_NeoPixel.h, 15
 neoPixelType
 Adafruit_NeoPixel.h, 15
 numPixels
 Adafruit_NeoPixel, 9

RD117_ARDUINO.ino
 aun_ir_buffer, 31
 aun_red_buffer, 31
 ch_hr_valid, 31
 ch_spo2_valid, 31
 loop, 31
 n_heart_rate, 31
 n_ir_buffer_length, 32
 n_spo2, 32
 setup, 31
 uch_dummy, 32
 RD117_ARDUINO/Adafruit_NeoPixel.cpp, 11
 RD117_ARDUINO/Adafruit_NeoPixel.h, 11
 RD117_ARDUINO/SoftI2CMaster.h, 32
 RD117_ARDUINO/algorithm.cpp, 15
 RD117_ARDUINO/algorithm.h, 18
 RD117_ARDUINO/max30102.cpp, 22
 RD117_ARDUINO/max30102.h, 25
 REG_FIFO_CONFIG
 max30102.h, 26
 REG_FIFO_DATA
 max30102.h, 26
 REG_FIFO_RD_PTR
 max30102.h, 26
 REG_FIFO_WR_PTR
 max30102.h, 27
 REG_INTR_ENABLE_1
 max30102.h, 27
 REG_INTR_ENABLE_2
 max30102.h, 27
 REG_INTR_STATUS_1
 max30102.h, 27
 REG_INTR_STATUS_2
 max30102.h, 27
 REG_LED1_PA
 max30102.h, 27
 REG_LED2_PA
 max30102.h, 27
 REG_MODE_CONFIG
 max30102.h, 27
 REG_MULTI_LED_CTRL1
 max30102.h, 27
 REG_MULTI_LED_CTRL2
 max30102.h, 27
 REG_OVF_COUNTER
 max30102.h, 27
 REG_PART_ID
 max30102.h, 28
 REG_PILOT_PA
 max30102.h, 28
 REG_REV_ID
 max30102.h, 28
 REG_SPO2_CONFIG
 max30102.h, 28
 REG_TEMP_CONFIG
 max30102.h, 28
 REG_TEMP_FRAC
 max30102.h, 28
 REG_TEMP_INTR
 max30102.h, 28
 SCL_DDR
 SoftI2CMaster.h, 34
 SCL_IN
 SoftI2CMaster.h, 34
 SCL_OUT
 SoftI2CMaster.h, 34
 SDA_DDR
 SoftI2CMaster.h, 34
 SDA_IN
 SoftI2CMaster.h, 34
 SDA_OUT
 SoftI2CMaster.h, 34
 SOFTI2CMASTER_H_
 SoftI2CMaster.h, 34
 setBrightness
 Adafruit_NeoPixel, 9
 setPin
 Adafruit_NeoPixel, 9
 setPixelColor
 Adafruit_NeoPixel, 9
 setup
 RD117_ARDUINO.ino, 31
 show
 Adafruit_NeoPixel, 9
 SoftI2CMaster.h
 _SOFTI2C_H, 33
 __attribute__, 35
 __tmp_reg__, 33
 FAC, 33
 I2C_CPUFREQ, 33
 I2C_DELAY_COUNTER, 33
 I2C_FASTMODE, 33
 I2C_MAX_STRETCH, 33
 I2C_NOINTERRUPT, 33
 I2C_READ, 33
 I2C_SLOWMODE, 34
 I2C_TIMEOUT, 34
 I2C_WRITE, 34
 i2c_delay_half, 35
 i2c_init, 35
 i2c_read, 35
 i2c_rep_start, 35
 i2c_start, 35
 i2c_start_wait, 35
 i2c_stop, 35
 i2c_wait_scl_high, 35
 i2c_write, 35

- SCL_DDR, [34](#)
- SCL_IN, [34](#)
- SCL_OUT, [34](#)
- SDA_DDR, [34](#)
- SDA_IN, [34](#)
- SDA_OUT, [34](#)
- SOFTI2DMASTER_H_, [34](#)

true

- algorithm.h, [20](#)

uch_dummy

- RD117_ARDUINO.ino, [32](#)

uch_spo2_table

- algorithm.h, [22](#)

updateLength

- Adafruit_NeoPixel, [9](#)

updateType

- Adafruit_NeoPixel, [9](#)