

GPS

Generated by Doxygen 1.8.11

Contents

1	Hardware project	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	gps_data Struct Reference	7
4.1.1	Detailed Description	7
4.2	nmea_sentence_gga Struct Reference	7
4.2.1	Detailed Description	8
4.3	nmea_sentence_rmc Struct Reference	8
4.3.1	Detailed Description	8
5	File Documentation	9
5.1	software/gps.c File Reference	9
5.1.1	Detailed Description	10
5.1.2	Function Documentation	10
5.1.2.1	calcDistance(float lat1, float lon1, float lat2, float lon2)	10
5.1.2.2	deg2rad(float deg)	11
5.1.2.3	gpsSend(char *message)	12
5.1.2.4	toggleGPS(unsigned int state)	12
5.1.2.5	toggleGPSInterrupt(unsigned int state)	13

5.1.3	Variable Documentation	13
5.1.3.1	dataValid	13
5.1.3.2	GPSTData	13
5.2	software/gps.h File Reference	14
5.2.1	Detailed Description	15
5.2.2	Typedef Documentation	15
5.2.2.1	gps_data	15
5.2.3	Function Documentation	15
5.2.3.1	calcDistance(float lat1, float lon1, float lat2, float lon2)	15
5.2.3.2	deg2rad(float deg)	16
5.2.3.3	gpsSend(char *message)	16
5.2.3.4	toggleGPS(unsigned int state)	17
5.2.3.5	toggleGPSInterrupt(unsigned int state)	17
5.2.4	Variable Documentation	18
5.2.4.1	dataValid	18
5.2.4.2	GPSTData	18
5.3	software/led.c File Reference	18
5.3.1	Detailed Description	19
5.3.2	Function Documentation	19
5.3.2.1	initLED(void)	19
5.3.2.2	toggleLED(int n, unsigned int state, unsigned int duration)	19
5.4	software/led.h File Reference	20
5.4.1	Detailed Description	20
5.4.2	Function Documentation	21
5.4.2.1	initLED(void)	21
5.4.2.2	toggleLED(int n, unsigned int state, unsigned int duration)	21
5.5	software/main.c File Reference	21
5.5.1	Detailed Description	22
5.5.2	Function Documentation	22
5.5.2.1	delay(float x)	22

5.5.2.2	main(void)	23
5.5.2.3	toggleCommunication(unsigned int state)	24
5.5.3	Variable Documentation	25
5.5.3.1	modeSelected	25
5.6	software/main.h File Reference	25
5.6.1	Detailed Description	26
5.6.2	Function Documentation	26
5.6.2.1	delay(float x)	26
5.6.2.2	toggleCommunication(unsigned int state)	27
5.6.3	Variable Documentation	27
5.6.3.1	modeSelected	28
5.7	software/oled.c File Reference	28
5.7.1	Detailed Description	30
5.7.2	Function Documentation	30
5.7.2.1	calculateDirection()	30
5.7.2.2	displayMessage(char *string)	30
5.7.2.3	ftoa(char *p, float x)	31
5.7.2.4	gfx_BGcolour(int color)	32
5.7.2.5	gfx_CalculateOrbit(int angle, int distance, int *x, int *y)	32
5.7.2.6	gfx_DrawCircle(int x, int y, int radius, int color)	33
5.7.2.7	gfx_DrawLine(int x1, int y1, int x2, int y2, int color)	34
5.7.2.8	gfx_MoveOrigin(int x, int y)	34
5.7.2.9	gfx_PutString(char *string)	35
5.7.2.10	gfx_Rectangle(int x1, int y1, int x2, int y2, int color)	36
5.7.2.11	gfx_RectangleFilled(int x1, int y1, int x2, int y2, int color)	37
5.7.2.12	gfx_ScreenMode(int mode)	38
5.7.2.13	sendChar(int c)	38
5.7.2.14	SSTimeout(int t)	39
5.7.2.15	toggleOLEDInterrupt(unsigned int state)	40
5.7.2.16	txt_BGColor(int color)	41

5.7.2.17	<code>txt_FGColor(int color)</code>	41
5.7.2.18	<code>txt_Width(int multi)</code>	42
5.7.3	Variable Documentation	42
5.7.3.1	<code>displayHasBeenUpdated</code>	43
5.7.3.2	<code>modeDisplay</code>	43
5.7.3.3	<code>oldModeDisplay</code>	43
5.8	<code>software/oled.h</code> File Reference	43
5.8.1	Detailed Description	51
5.8.2	Function Documentation	51
5.8.2.1	<code>calculateDirection()</code>	51
5.8.2.2	<code>displayMessage(char *string)</code>	51
5.8.2.3	<code>ftoa(char *p, float x)</code>	52
5.8.2.4	<code>gfx_BGcolour(int color)</code>	53
5.8.2.5	<code>gfx_CalculateOrbit(int angle, int distance, int *x, int *y)</code>	53
5.8.2.6	<code>gfx_DrawCircle(int x, int y, int radius, int color)</code>	54
5.8.2.7	<code>gfx_DrawLine(int x1, int y1, int x2, int y2, int color)</code>	55
5.8.2.8	<code>gfx_MoveOrigin(int x, int y)</code>	55
5.8.2.9	<code>gfx_PutString(char *string)</code>	56
5.8.2.10	<code>gfx_Rectangle(int x1, int y1, int x2, int y2, int color)</code>	57
5.8.2.11	<code>gfx_RectangleFilled(int x1, int y1, int x2, int y2, int color)</code>	58
5.8.2.12	<code>gfx_ScreenMode(int mode)</code>	59
5.8.2.13	<code>sendChar(int c)</code>	59
5.8.2.14	<code>SSTimeout(int t)</code>	60
5.8.2.15	<code>toggleOLEDInterrupt(unsigned int state)</code>	61
5.8.2.16	<code>txt_BGColor(int color)</code>	62
5.8.2.17	<code>txt_FGColor(int color)</code>	62
5.8.2.18	<code>txt_Width(int multi)</code>	63
5.8.3	Variable Documentation	63
5.8.3.1	<code>displayHasBeenUpdated</code>	64
5.8.3.2	<code>modeDisplay</code>	64

5.8.3.3	oldModeDisplay	64
5.9	software/pad.c File Reference	64
5.9.1	Detailed Description	64
5.9.2	Function Documentation	65
5.9.2.1	initPAD(void)	65
5.10	software/pad.h File Reference	65
5.10.1	Detailed Description	66
5.10.2	Function Documentation	66
5.10.2.1	initPAD(void)	66
5.11	software/parser_nmea.c File Reference	66
5.11.1	Detailed Description	67
5.11.2	Function Documentation	67
5.11.2.1	hex2int(char c)	67
5.11.2.2	nmea_check(const char *sentence, int strict)	68
5.11.2.3	nmea_isfield(char c)	68
5.11.2.4	nmea_parse_gga(struct nmea_sentence_gga *frame, const char *sentence)	69
5.11.2.5	nmea_parse_rmc(struct nmea_sentence_rmc *frame, const char *sentence)	69
5.11.2.6	nmea_scan(const char *sentence, const char *format,...)	70
5.11.2.7	nmea_sentence_id(char *sentence)	71
5.12	software/parser_nmea.h File Reference	71
5.12.1	Detailed Description	73
5.12.2	Typedef Documentation	73
5.12.2.1	nmea_sentence_gga	73
5.12.2.2	nmea_sentence_rmc	73
5.12.3	Enumeration Type Documentation	73
5.12.3.1	nmea_sentence_id	73
5.12.4	Function Documentation	73
5.12.4.1	hex2int(char c)	73
5.12.4.2	nmea_check(const char *sentence, int strict)	74
5.12.4.3	nmea_isfield(char c)	75
5.12.4.4	nmea_parse_gga(nmea_sentence_gga *frame, const char *sentence)	75
5.12.4.5	nmea_parse_rmc(nmea_sentence_rmc *frame, const char *sentence)	76
5.12.4.6	nmea_scan(const char *sentence, const char *format,...)	76
5.12.4.7	nmea_sentence_id(char *sentence)	77

Chapter 1

Hardware project

This project is part of an engineering school programming project and aims to build a GPS software of an embedded system: compass, favorites places, waypoints, navigation, etc.

Project

Hardware used:

- Texas Instrument MSP430F169
- Adafruit Ultimate GPS Breakout
- 4D Systems uOLED-128-G2

A presentation of the project is available in [French](#).

Documentation

An [HTML](#) documentation is available, with graph to a better understanding of the process (see [Files](#) tabs).

An [PDF](#) (LaTeX) documentation is also available.

To update the documentation, make sure you have installed [Doxygen](#) and [Graphviz](#).

Then browse to the root of the project folder and run: `doxygen doxygen.cfg`

To update the LaTeX documentation, browse `doc/latex` and run `make`

Usage and options

To compile the project, you'll need [CrossWorks for MSP430](#).

Open `project.hzp` to launch project. See [CrossWorks for MSP430 Reference Manual](#) for further explanations.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

gps_data	7
nmea_sentence_gga	7
nmea_sentence_rmc	8

Chapter 3

File Index

3.1 File List

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

software/ gps.c	
File containing the GPS functions	9
software/ gps.h	
File containing the GPS functions	14
software/ led.c	
File containing the LED functions	18
software/ led.h	
File containing the LED functions	20
software/ main.c	
File containing the main functions	21
software/ main.h	
File containing the main functions	25
software/ oled.c	
File containing the OLED functions	28
software/ oled.h	
File containing the OLED functions	43
software/ pad.c	
File containing the PAD functions	64
software/ pad.h	
File containing the PAD functions	65
software/ parser_nmea.c	
File containing the NMEA parser functions	66
software/ parser_nmea.h	
File containing the NMEA parser functions	71

Chapter 4

Data Structure Documentation

4.1 `gps_data` Struct Reference

```
#include <gps.h>
```

Data Fields

- float `latitude`
The latitude.
- float `longitude`
The longitude.
- float `speed`
The speed.
- float `heading`
The heading.

4.1.1 Detailed Description

Structure that contains useful GPS data

The documentation for this struct was generated from the following file:

- `software/gps.h`

4.2 `nmea_sentence_gga` Struct Reference

```
#include <parser_nmea.h>
```

Data Fields

- float [latitude](#)
Latitude.
- float [longitude](#)
Longitude.
- int [fix_quality](#)
Quality.
- int [satellites_tracked](#)
Number of satellites.
- float [hdop](#)
HDOP.
- float [altitude](#)
Altitude.
- char [altitude_units](#)
Altitude unit.
- float [height](#)
Height.
- char [height_units](#)
Height unit.
- int [dgps_age](#)
Age.

4.2.1 Detailed Description

The structure that contains the data of GGA sentences

The documentation for this struct was generated from the following file:

- [software/parser_nmea.h](#)

4.3 nmea_sentence_rmc Struct Reference

```
#include <parser_nmea.h>
```

Data Fields

- int [valid](#)
Sentence validity.
- float [latitude](#)
Latitude.
- float [longitude](#)
Longitude.
- float [speed](#)
Speed.
- float [heading](#)
Heading.

4.3.1 Detailed Description

The structure that contains the data of RMC sentences

The documentation for this struct was generated from the following file:

- [software/parser_nmea.h](#)

Chapter 5

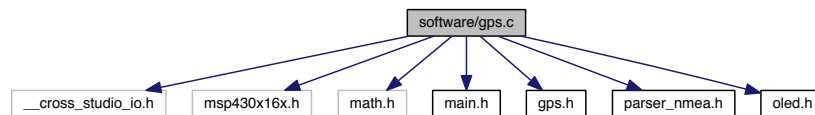
File Documentation

5.1 software/gps.c File Reference

File containing the GPS functions.

```
#include <__cross_studio_io.h>
#include <msp430x16x.h>
#include <math.h>
#include "main.h"
#include "gps.h"
#include "parser_nmea.h"
#include "oled.h"
```

Include dependency graph for gps.c:



Functions

- void [toggleGPS](#) (unsigned int state)
Toggle GPS (P4.0, ENABLE_GPS)
- void [toggleGPSInterrupt](#) (unsigned int state)
Toggle GPS interrupt.
- void [enableUSARTforGPS](#) (void)
Enable and config USART for GPS.
- void [gpsSend](#) (char *message)
Send sentences to GPS to configure it (interrupt mode)
- void [usart0_rx](#) (void)
Receive function for GPS data (USART0, interrupt mode)
- float [calcDistance](#) (float lat1, float lon1, float lat2, float lon2)
Calculate the distance between two points (Haversine formula)
- float [deg2rad](#) (float deg)
Degrees to radians converter.

Variables

- unsigned int [dataValid](#)
- [gps_data](#) GPSTData

Store GPS valid data.

5.1.1 Detailed Description

File containing the GPS functions.

Author

Gaël Foppolo (gaelfoppolo)

5.1.2 Function Documentation

5.1.2.1 float calcDistance (float *lat1*, float *lon1*, float *lat2*, float *lon2*)

Calculate the distance between two points (Haversine formula)

See also

[Wikipedia](#)

Parameters

<i>lat1</i>	The latitude of the first point
<i>lon1</i>	The longitude of the first point
<i>lat2</i>	The latitude of the second point
<i>lon2</i>	The longitude of the second point

Returns

The distance (in km)

Here is the call graph for this function:



5.1.2.2 float deg2rad (float *deg*)

Degrees to radians converter.

Parameters

<i>deg</i>	The angle in degrees
------------	----------------------

Returns

The angle in radians

Here is the caller graph for this function:

**5.1.2.3 void gpsSend (char * *message*)**

Send sentences to GPS to configure it (interrupt mode)

Parameters

<i>message</i>	Message to send
----------------	-----------------

Here is the caller graph for this function:

**5.1.2.4 void toggleGPS (unsigned int *state*)**

Toggle GPS (P4.0, ENABLE_GPS)

1 = enable, 0 = disable

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.1.2.5 void toggleGPSInterrupt (unsigned int *state*)

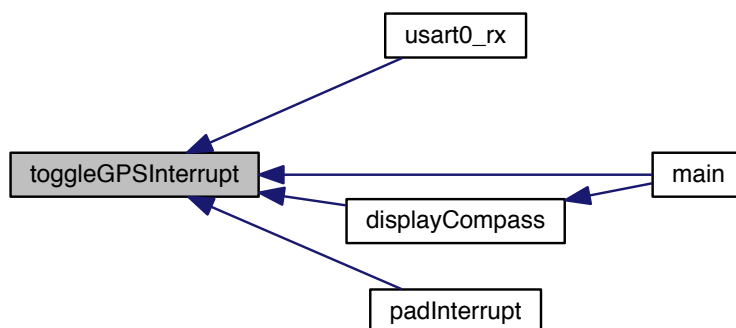
Toggle GPS interrupt.

1 = interrupt enable for GPS, 0 = disable

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.1.3 Variable Documentation

5.1.3.1 unsigned int dataValid

Data are valid or not?

5.1.3.2 gps_data GPSTData

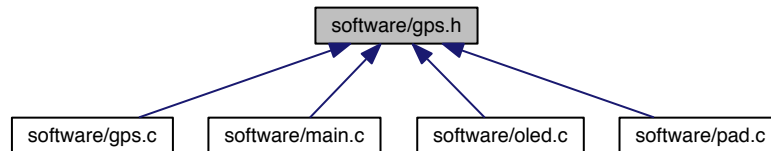
Store GPS valid data.

Useful data received and valid.

5.2 software/gps.h File Reference

File containing the GPS functions.

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



Data Structures

- struct [gps_data](#)

Macros

- #define [NUMBERS_OF_SENTENCE](#) 4
How many sentences do we want to receive in interrupt mode.
- #define [NUMBERS_OF_SENTENCE_MAX](#) 10
How many sentences do we want to receive in interrupt mode (MAX)
- #define [RATE_1SEC](#) "\$PMTK220,1000*1F\r\n"
GPS emits NMEA sentences every 1 sec.
- #define [RATE_2SEC](#) "\$PMTK220,2000*1C\r\n"
GPS emits NMEA sentences every 2 sec.
- #define [RATE_5SEC](#) "\$PMTK220,5000*1B\r\n"
GPS emits NMEA sentences every 5 sec.
- #define [RATE_10SEC](#) "\$PMTK220,10000*2F\r\n"
GPS emits NMEA sentences every 10 sec.
- #define [DISABLE_ALL](#) "\$PMTK314,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*28\r\n"
Tell the GPS to not send sentences.
- #define [TURN_ALL](#) "\$PMTK314,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0*28\r\n"
Tell the GPS to send all sentences.
- #define [GGA_RMC](#) "\$PMTK314,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0*28\r\n"
Tell the GPS to send GGA and RMC sentences only.
- #define [RMC_ONLY](#) "\$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0*29\r\n"
Tell the GPS to send RMC sentences only.

Typedefs

- typedef struct [gps_data](#) [gps_data](#)

Functions

- void [toggleGPS](#) (unsigned int state)
Toggle GPS (P4.0, ENABLE_GPS)
- void [toggleGPSInterrupt](#) (unsigned int state)
Toggle GPS interrupt.
- void [enableUSARTforGPS](#) (void)
Enable and config USART for GPS.
- void [gpsSend](#) (char *message)
Send sentences to GPS to configure it (interrupt mode)
- void [usart0_rx](#) (void)
Receive function for GPS data (USART0, interrupt mode)
- float [calcDistance](#) (float lat1, float lon1, float lat2, float lon2)
Calculate the distance between two points (Haversine formula)
- float [deg2rad](#) (float deg)
Degrees to radians converter.

Variables

- unsigned int [dataValid](#)
- struct [gps_data](#) GPSTData
Useful data received and valid.

5.2.1 Detailed Description

File containing the GPS functions.

Author

Gaël Foppolo (gaelfoppolo)

5.2.2 Typedef Documentation

5.2.2.1 typedef struct [gps_data](#) [gps_data](#)

Structure that contains useful GPS data

5.2.3 Function Documentation

5.2.3.1 float [calcDistance](#) (float *lat1*, float *lon1*, float *lat2*, float *lon2*)

Calculate the distance between two points (Haversine formula)

See also

Wikipedia

Parameters

<i>lat1</i>	The latitude of the first point
<i>lon1</i>	The longitude of the first point
<i>lat2</i>	The latitude of the second point
<i>lon2</i>	The longitude of the second point

Returns

The distance (in km)

Here is the call graph for this function:

5.2.3.2 float deg2rad (float *deg*)

Degrees to radians converter.

Parameters

<i>deg</i>	The angle in degrees
------------	----------------------

Returns

The angle in radians

Here is the caller graph for this function:

5.2.3.3 void gpsSend (char * *message*)

Send sentences to GPS to configure it (interrupt mode)

Parameters

<i>message</i>	Message to send
----------------	-----------------

Here is the caller graph for this function:



5.2.3.4 void toggleGPS (unsigned int *state*)

Toggle GPS (P4.0, ENABLE_GPS)

1 = enable, 0 = disable

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.2.3.5 void toggleGPSInterrupt (unsigned int *state*)

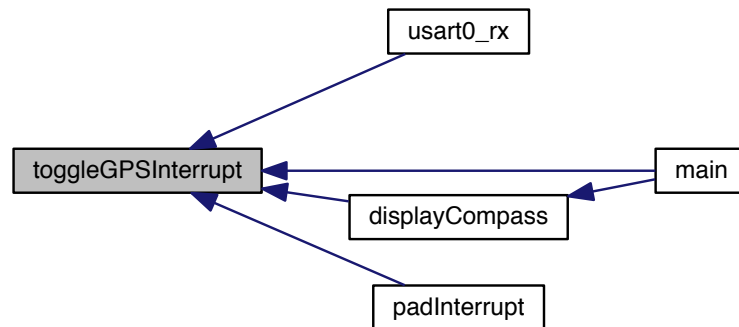
Toggle GPS interrupt.

1 = interrupt enable for GPS, 0 = disable

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.2.4 Variable Documentation

5.2.4.1 unsigned int dataValid

Data are valid or not?

5.2.4.2 struct gps_data GPSData

Useful data received and valid.

Useful data received and valid.

5.3 software/led.c File Reference

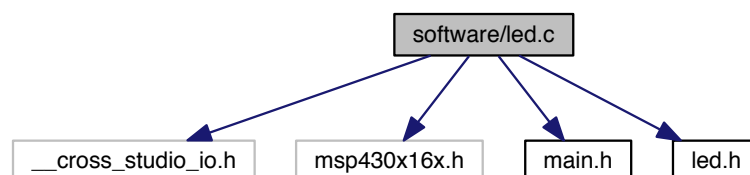
File containing the LED functions.

```

#include <__cross_studio_io.h>
#include <msp430x16x.h>
#include "main.h"
#include "led.h"

```

Include dependency graph for led.c:



Functions

- void `initLED` (void)
Init LED (P1.0 -> P1.4)
- void `toggleLED` (int *n*, unsigned int *state*, unsigned int *duration*)
Toogle the state of the choosen LED for a choosen time.

5.3.1 Detailed Description

File containing the LED functions.

Author

Gaël Foppolo (gaelfoppolo)

5.3.2 Function Documentation

5.3.2.1 void `initLED` (void)

Init LED (P1.0 -> P1.4)

All ready to use and state cleared

Here is the caller graph for this function:



5.3.2.2 void `toggleLED` (int *n*, unsigned int *state*, unsigned int *duration*)

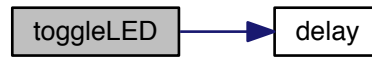
Toogle the state of the choosen LED for a choosen time.

duration = 0 -> stay in the state choosen

Parameters

<i>n</i>	The LED to toogle
<i>state</i>	The new state
<i>duration</i>	The time to toogle the state of the LED

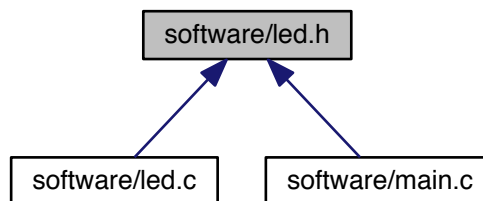
Here is the call graph for this function:



5.4 software/led.h File Reference

File containing the LED functions.

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



Functions

- void `initLED` (void)
Init LED (P1.0 -> P1.4)
- void `toggleLED` (int n, unsigned int state, unsigned int duration)
Toogle the state of the choosen LED for a choosen time.

5.4.1 Detailed Description

File containing the LED functions.

Author

Gaël Foppolo (gaelfoppolo)

5.4.2 Function Documentation

5.4.2.1 void initLED (void)

Init LED (P1.0 -> P1.4)

All ready to use and state cleared

Here is the caller graph for this function:



5.4.2.2 void toggleLED (int *n*, unsigned int *state*, unsigned int *duration*)

Toogle the state of the choosen LED for a choosen time.

duration = 0 -> stay in the state choosen

Parameters

<i>n</i>	The LED to toogle
<i>state</i>	The new state
<i>duration</i>	The time to toogle the state of the LED

Here is the call graph for this function:

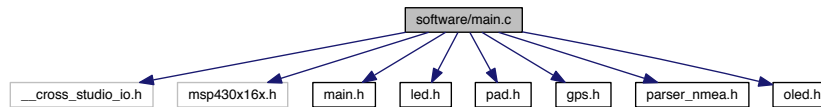


5.5 software/main.c File Reference

File containing the main functions.

```
#include <__cross_studio_io.h>
#include <msp430x16x.h>
#include "main.h"
#include "led.h"
#include "pad.h"
#include "gps.h"
#include "parser_nmea.h"
#include "oled.h"
```

Include dependency graph for main.c:



Functions

- void [main](#) (void)
- void [toggleCommunication](#) (unsigned int state)
Toogle the communication (P4.2, CMD_SWITCH)
- void [configureClock](#) (void)
Configure the external clock.
- void [delay](#) (float x)
Wait for x sec.

Variables

- unsigned int [modeSelected](#)

5.5.1 Detailed Description

File containing the main functions.

Author

Gaël Foppolo (gaelfoppolo)

5.5.2 Function Documentation

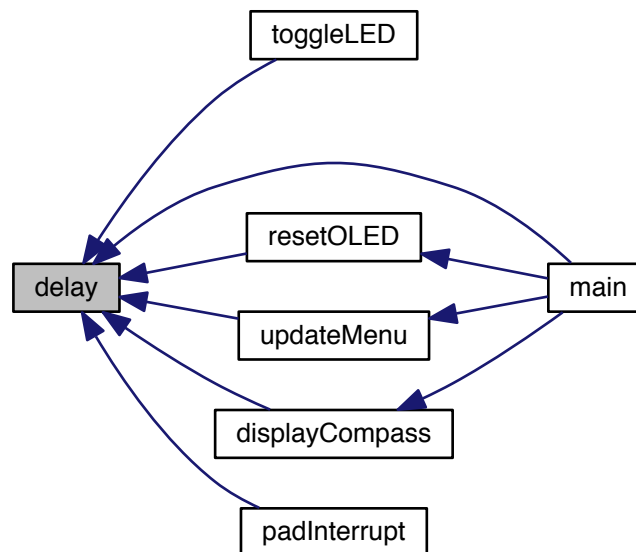
5.5.2.1 void delay (float x)

Wait for x sec.

Parameters

<i>x</i>	The time to wait (in sec ~)
----------	-----------------------------

Here is the caller graph for this function:



5.5.2.2 void main (void)

Menu entry point

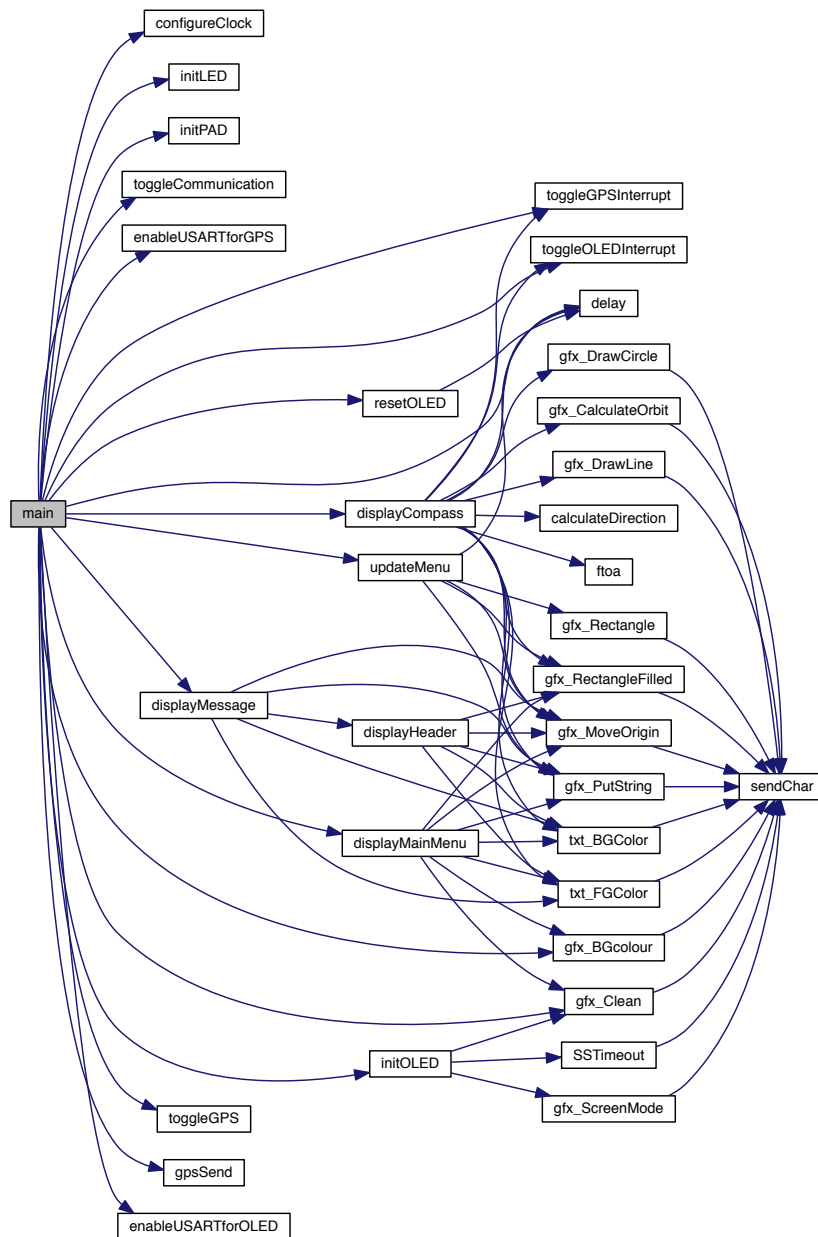
Compass entry point

Navigation entry point

Record entry point

Shutdown entry point

Here is the call graph for this function:



5.5.2.3 void toggleCommunication (unsigned int state)

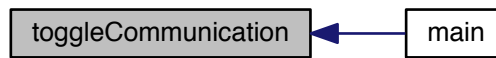
Toggle the communication (P4.2, CMD_SWITCH)

1 = USB, 0 = GPS

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.5.3 Variable Documentation

5.5.3.1 unsigned int modeSelected

Mode selected by the user,

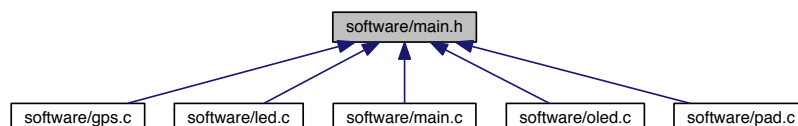
See also

[M_MENU](#), etc.

5.6 software/main.h File Reference

File containing the main functions.

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



Macros

- `#define M_MENU 0`
Mode menu.
- `#define M_COMPASS 1`
Mode compass.
- `#define M_NAVIG 2`
Mode navigation.
- `#define M_RECORD 3`
Mode record.
- `#define M_SHUTDOWN 4`
Mode shutdown.

- `#define COMM_GPS 0`
Communication with GPS module.
- `#define COMM_USB 1`
Communication with USB.
- `#define YES 1`
YES.
- `#define NO 0`
NO.

Functions

- void `configureClock` (void)
Configure the external clock.
- void `toggleCommunication` (unsigned int state)
Toogle the communication (P4.2, CMD_SWITCH)
- void `delay` (float x)
Wait for x sec.

Variables

- unsigned int `modeSelected`

5.6.1 Detailed Description

File containing the main functions.

Author

Gaël Foppolo (gaelfoppolo)

5.6.2 Function Documentation

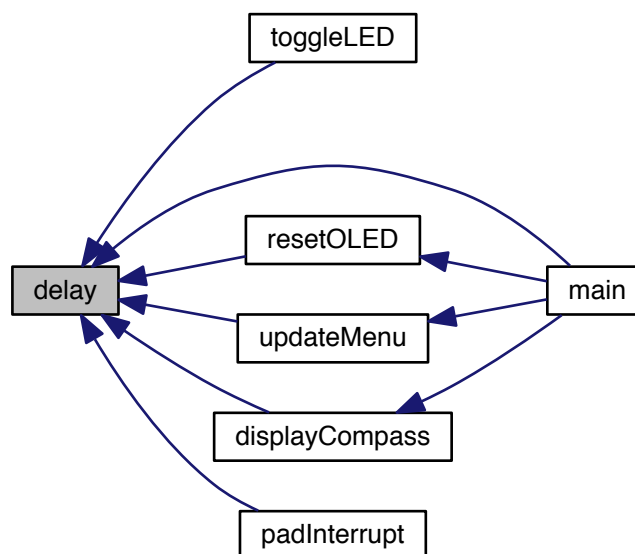
5.6.2.1 void delay (float x)

Wait for x sec.

Parameters

x	The time to wait (in sec ~)
---	-----------------------------

Here is the caller graph for this function:



5.6.2.2 void toggleCommunication (unsigned int *state*)

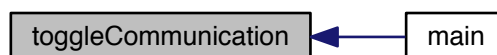
Toogle the communication (P4.2, CMD_SWITCH)

1 = USB, 0 = GPS

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.6.3 Variable Documentation

5.6.3.1 unsigned int modeSelected

Mode selected by the user,

See also

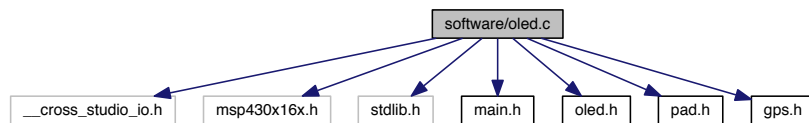
[M_MENU](#), etc.

5.7 software/oled.c File Reference

File containing the OLED functions.

```
#include <__cross_studio_io.h>
#include <msp430x16x.h>
#include <stdlib.h>
#include "main.h"
#include "oled.h"
#include "pad.h"
#include "gps.h"
```

Include dependency graph for oled.c:



Functions

- void [enableUSARTforOLED](#) (void)
Enable and config USART for OLED.
- void [resetOLED](#) ()
Reset OLED.
- void [toggleOLEDInterrupt](#) (unsigned int state)
Toggle OLED interrupt.
- void [sendChar](#) (int c)
Send char.
- void [usart1_rx](#) (void)
Receive function for OLED data (USART1, interrupt mode)
- void [gfx_Clean](#) ()
Clean the screen.
- void [gfx_BGcolour](#) (int color)
Set the background color.
- void [gfx_PutString](#) (char *string)
Put a string on the screen.
- void [gfx_RectangleFilled](#) (int x1, int y1, int x2, int y2, int color)
Draw a rectangle filled with a color.

- void `SSTimeout` (int t)
Screensave mode.
- void `gfx_CalculateOrbit` (int angle, int distance, int *x, int *y)
Calculate the (x,y) pos (orbit) from angle and distance.
- void `gfx_DrawCircle` (int x, int y, int radius, int color)
Draw a circle.
- void `gfx_DrawLine` (int x1, int y1, int x2, int y2, int color)
Draw a line.
- void `gfx_MoveOrigin` (int x, int y)
Move to origin to a position.
- void `gfx_ScreenMode` (int mode)
Screen mode (portrait/landscape)
- void `txt_FGColor` (int color)
Set the text color.
- void `txt_BGColor` (int color)
Set the text background color.
- void `setBaudRate` ()
Set the baud rate.
- void `gfx_Rectangle` (int x1, int y1, int x2, int y2, int color)
Draw a rectangle.
- void `txt_Width` (int multi)
Set the width of the text.
- void `initOLED` ()
Configure OLED for proper using.
- void `displayMainMenu` ()
Display menu.
- void `updateMenu` ()
Update the menu with currently selected.
- void `displayHeader` ()
Display message header.
- void `displayMessage` (char *string)
Display a string in the center of the screen.
- void `displayCompass` ()
Display the compass.
- void `ftoa` (char *p, float x)
Float to string conversion.
- char * `calculateDirection` ()
Calculate the direction (N, S, NE, etc.)

Variables

- int `displayHasBeenUpdated`
- unsigned int `modeDisplay` = `MD_SHUTDOWN`
- unsigned int `oldModeDisplay`
- int `answer` = 0
The answer received by the OLED.
- int `flagReceive` = 0
Answer received?

5.7.1 Detailed Description

File containing the OLED functions.

Author

Gaël Foppolo (gaelfoppolo)

5.7.2 Function Documentation

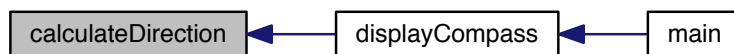
5.7.2.1 `char* calculateDirection ()`

Calculate the direction (N, S, NE, etc.)

Returns

The direction.

Here is the caller graph for this function:



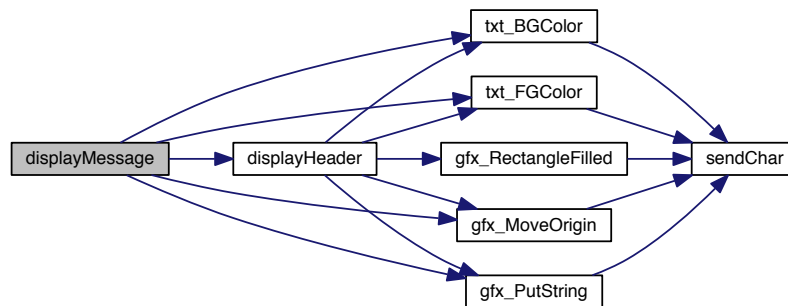
5.7.2.2 `void displayMessage (char * string)`

Display a string in the center of the screen.

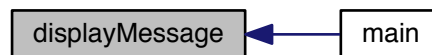
Parameters

<i>string</i>	The string to display
---------------	-----------------------

Here is the call graph for this function:



Here is the caller graph for this function:



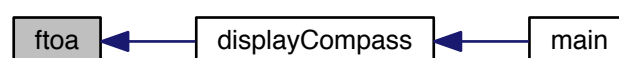
5.7.2.3 void ftoa (char * p, float x)

Float to string conversion.

Parameters

	<i>p</i>	The buffer (string)
in	<i>x</i>	The float

Here is the caller graph for this function:



5.7.2.4 void gfx_BGcolour (int *color*)

Set the background color.

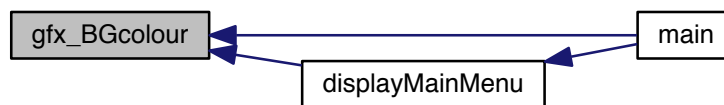
Parameters

in	<i>color</i>	The color
----	--------------	-----------

Here is the call graph for this function:



Here is the caller graph for this function:



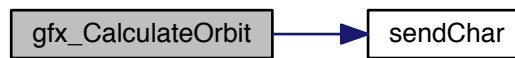
5.7.2.5 void gfx_CalculateOrbit (int *angle*, int *distance*, int * *x*, int * *y*)

Calculate the (x,y) pos (orbit) from angle and distance.

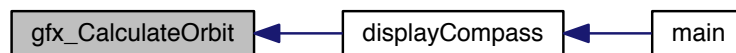
Parameters

in	<i>angle</i>	The angle
in	<i>distance</i>	The distance
	<i>x</i>	The x pos computed
	<i>y</i>	The y pos computed

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.6 void gfx_DrawCircle (int *x*, int *y*, int *radius*, int *color*)

Draw a circle.

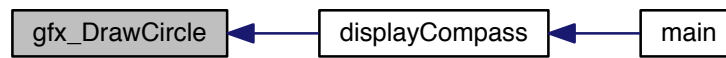
Parameters

in	<i>x</i>	x pos of center of the circle
in	<i>y</i>	y pos of center of the circle
in	<i>radius</i>	The radius
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.7 void gfx_DrawLine (int x1, int y1, int x2, int y2, int color)

Draw a line.

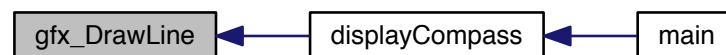
Parameters

in	<i>x1</i>	The x pos of the beginning of the line
in	<i>y1</i>	The y pos of the beginning of the line
in	<i>x2</i>	The x pos of the ending of the line
in	<i>y2</i>	The y pos of the ending of the line
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



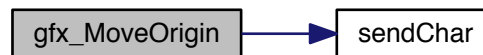
5.7.2.8 void gfx_MoveOrigin (int x, int y)

Move to origin to a position.

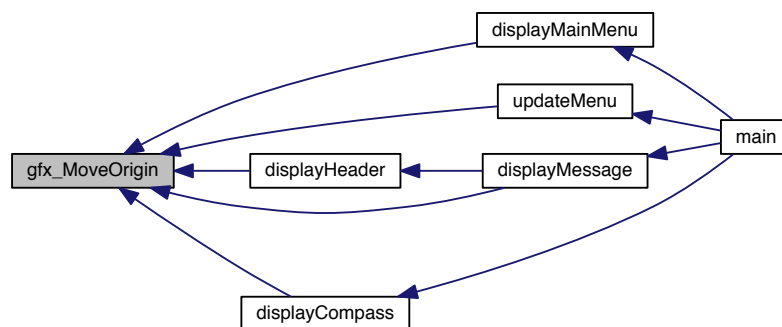
Parameters

in	<i>x</i>	The new x pos
in	<i>y</i>	The new y pos

Here is the call graph for this function:



Here is the caller graph for this function:



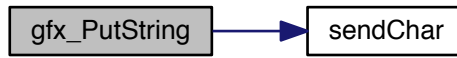
5.7.2.9 void gfx_PutString (char * *string*)

Put a string on the screen.

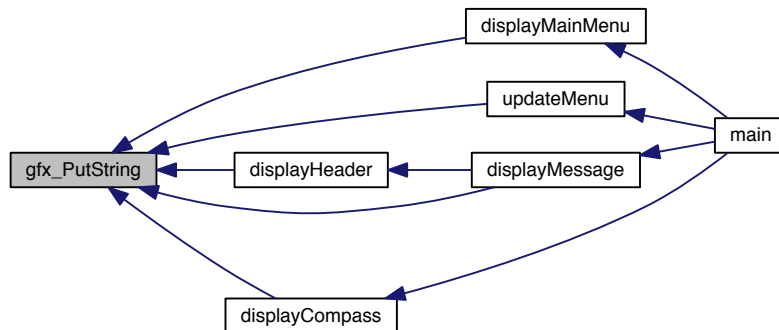
Parameters

<i>string</i>	The string
---------------	------------

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.10 void gfx_Rectangle (int x1, int y1, int x2, int y2, int color)

Draw a rectangle.

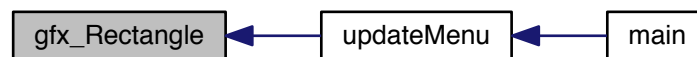
Parameters

in	<i>x1</i>	The x pos of the top left corner
in	<i>y1</i>	The y pos of the top left corner
in	<i>x2</i>	The x pos of the bottom right corner
in	<i>y2</i>	The y pos of the bottom right corner
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



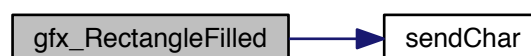
5.7.2.11 void gfx_RectangleFilled (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle filled with a color.

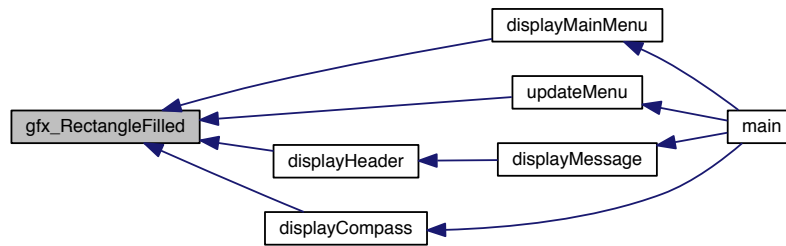
Parameters

in	<i>x1</i>	The x pos of the top left corner
in	<i>y1</i>	The y pos of the top left corner
in	<i>x2</i>	The x pos of the bottom right corner
in	<i>y2</i>	The y pos of the bottom right corner
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



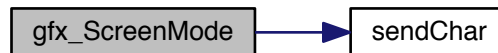
5.7.2.12 void gfx_ScreenMode (int mode)

Screen mode (portrait/landscape)

Parameters

in	<i>mode</i>	The mode
----	-------------	----------

Here is the call graph for this function:



Here is the caller graph for this function:



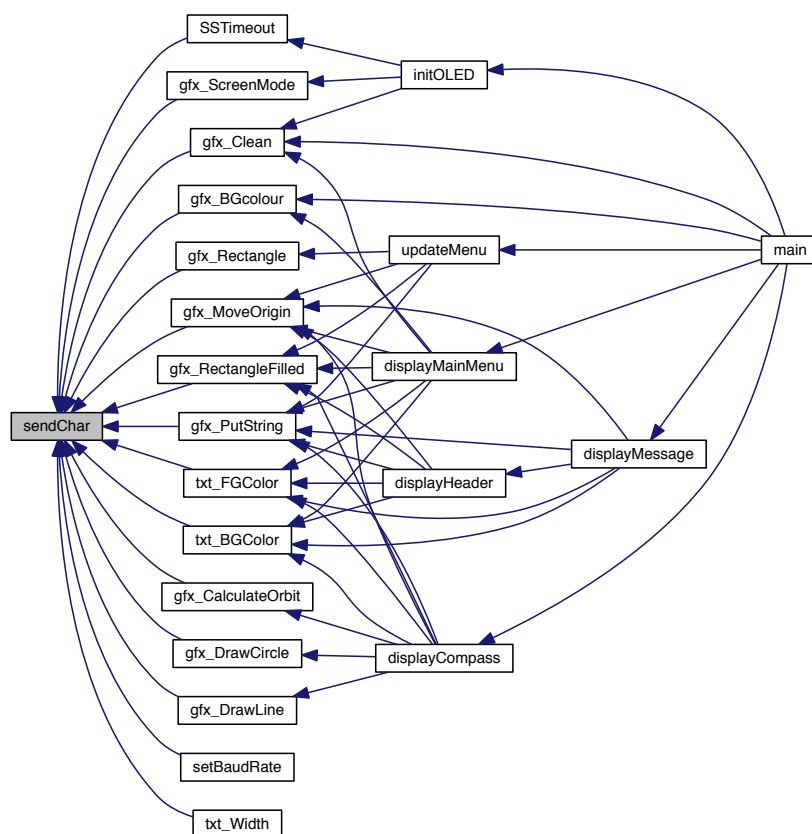
5.7.2.13 void sendChar (int c)

Send char.

Parameters

<code>c</code>	The int to send
----------------	-----------------

Here is the caller graph for this function:



5.7.2.14 void SStimeout (int t)

Screensave mode.

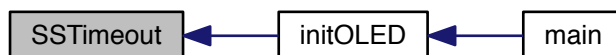
Parameters

<code>in</code>	<code>t</code>	The mode
-----------------	----------------	----------

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.15 void toggleOLEDInterrupt (unsigned int *state*)

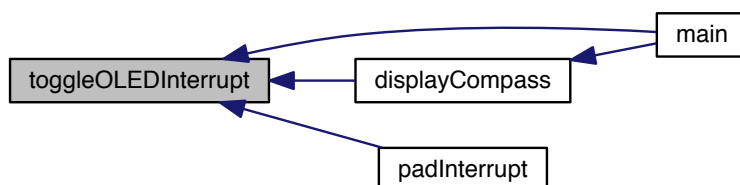
Toggle OLED interrupt.

1 = interrupt enable for OLED, 0 = disable

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.7.2.16 void txt_BGColor (int *color*)

Set the text background color.

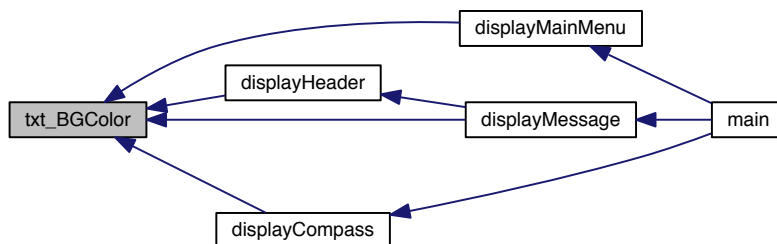
Parameters

in	<i>color</i>	The color
----	--------------	-----------

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.17 void txt_FGColor (int *color*)

Set the text color.

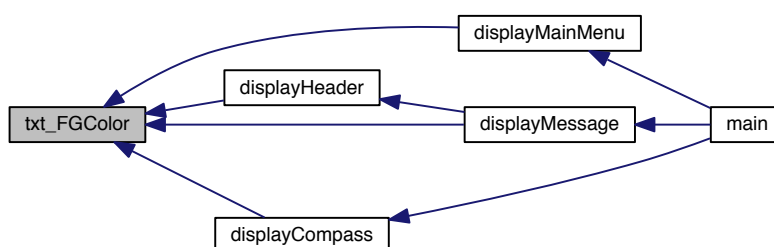
Parameters

in	<i>color</i>	The color
----	--------------	-----------

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.2.18 void txt_Width (int *multi*)

Set the width of the text.

Parameters

in	<i>multi</i>	The multi
----	--------------	-----------

Here is the call graph for this function:



5.7.3 Variable Documentation

5.7.3.1 int displayHasBeenUpdated

Is the display has been updated, aka needed

5.7.3.2 unsigned int modeDisplay = MD_SHUTDOWN

What mode is selected right now

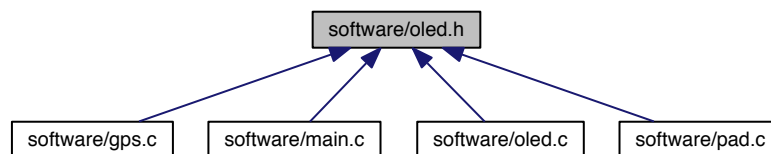
5.7.3.3 unsigned int oldModeDisplay

What mode was selected just before

5.8 software/oled.h File Reference

File containing the OLED functions.

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



Macros

- #define **CURRENT** 1
Current.
- #define **OLD** 0
OLD.
- #define **MD_COMPASS** 0
Mode selected on display is compass.
- #define **MD_NAVIG** 1
Mode selected on display is navigation.
- #define **MD_RECORD** 2
Mode selected on display is record.
- #define **MD_SHUTDOWN** 3
Mode selected on display is shutdown.
- #define **NONE** ""
Direction is none.
- #define **NORTH** "N"
Direction is north.

- #define **NE** "NE"
Direction is north-east.
- #define **EAST** "E"
Direction is east.
- #define **SE** "SE"
Direction is south-east.
- #define **SOUTH** "S"
Direction is south.
- #define **SW** "SO"
Direction is south-west.
- #define **WEST** "O"
Direction is west.
- #define **NW** "NO"
Direction is north-west.
- #define **OLED_ANSWER_ACK** 6
OLED respond is ACK.
- #define **ALICEBLUE** 0xF7DF
Color.
- #define **ANTIQUEWHITE** 0xFF5A
Color.
- #define **AQUA** 0x07FF
Color.
- #define **AQUAMARINE** 0x7FFA
Color.
- #define **AZURE** 0xF7FF
Color.
- #define **BEIGE** 0xF7BB
Color.
- #define **BISQUE** 0xFF38
Color.
- #define **BLACK** 0x0000
Color.
- #define **BLANCHEDALMOND** 0xFF59
Color.
- #define **BLUE** 0x001F
Color.
- #define **BLUEVIOLET** 0x895C
Color.
- #define **BROWN** 0xA145
Color.
- #define **BURLYWOOD** 0xDDD0
Color.
- #define **CADETBBLUE** 0x5CF4
Color.
- #define **CHARTREUSE** 0x7FE0
Color.
- #define **CHOCOLATE** 0xD343
Color.
- #define **CORAL** 0xFBEA
Color.
- #define **CORNFLOWERBLUE** 0x64BD

- Color.*
 - #define CORNSILK 0xFFDB
- Color.*
 - #define CRIMSON 0xD8A7
- Color.*
 - #define CYAN 0x07FF
- Color.*
 - #define DARKBLUE 0x0011
- Color.*
 - #define DARKCYAN 0x0451
- Color.*
 - #define DARKGOLDENROD 0xBC21
- Color.*
 - #define DARKGRAY 0xAD55
- Color.*
 - #define DARKGREEN 0x0320
- Color.*
 - #define DARKKHAKI 0xBDAD
- Color.*
 - #define DARKMAGENTA 0x8811
- Color.*
 - #define DARKOLIVEGREEN 0x5345
- Color.*
 - #define DARKORANGE 0xFC60
- Color.*
 - #define DARKORCHID 0x9999
- Color.*
 - #define DARKRED 0x8800
- Color.*
 - #define DARKSALMON 0xECAF
- Color.*
 - #define DARKSEAGREEN 0x8DF1
- Color.*
 - #define DARKSLATEBLUE 0x49F1
- Color.*
 - #define DARKSLATEGRAY 0x2A69
- Color.*
 - #define DARKTURQUOISE 0x067A
- Color.*
 - #define DARKVIOLET 0x901A
- Color.*
 - #define DEEPPINK 0xF8B2
- Color.*
 - #define DEEPSKYBLUE 0x05FF
- Color.*
 - #define DIMGRAY 0x6B4D
- Color.*
 - #define DODGERBLUE 0x1C9F
- Color.*
 - #define FIREBRICK 0xB104
- Color.*

- #define FLORALWHITE 0xFFDE
Color.
- #define FORESTGREEN 0x2444
Color.
- #define FUCHSIA 0xF81F
Color.
- #define GAINSBORO 0xDEFB
Color.
- #define GHOSTWHITE 0xFFDF
Color.
- #define GOLD 0xFEA0
Color.
- #define GOLDENROD 0xDD24
Color.
- #define GRAY 0x8410
Color.
- #define GREEN 0x0400
Color.
- #define GREENYELLOW 0xAFE5
Color.
- #define HONEYDEW 0xF7FE
Color.
- #define HOTPINK 0xFB56
Color.
- #define INDIANRED 0xCAEB
Color.
- #define INDIGO 0x4810
Color.
- #define IVORY 0xFFFF
Color.
- #define KHAKI 0xF731
Color.
- #define LAVENDER 0xE73F
Color.
- #define LAVENDERBLUSH 0xFF9E
Color.
- #define LAWNGREEN 0x7FE0
Color.
- #define LEMONCHIFFON 0xFFD9
Color.
- #define LIGHTBLUE 0xAEDC
Color.
- #define LIGHTCORAL 0xF410
Color.
- #define LIGHTCYAN 0xE7FF
Color.
- #define LIGHTGOLD 0xFFDA
Color.
- #define LIGHTGREEN 0x9772
Color.
- #define LIGHTGREY 0xD69A

- Color.*
 - #define [LIGHTPINK](#) 0xFDB8
- Color.*
 - #define [LIGHTSALMON](#) 0xFD0F
- Color.*
 - #define [LIGHTSEAGREEN](#) 0x2595
- Color.*
 - #define [LIGHTSKYBLUE](#) 0x867F
- Color.*
 - #define [LIGHTSLATEGRAY](#) 0x7453
- Color.*
 - #define [LIGHTSTEELBLUE](#) 0xB63B
- Color.*
 - #define [LIGHTYELLOW](#) 0xFFFC
- Color.*
 - #define [LIME](#) 0x07E0
- Color.*
 - #define [LIMEGREEN](#) 0x3666
- Color.*
 - #define [LINEN](#) 0xFF9C
- Color.*
 - #define [MAGENTA](#) 0xF81F
- Color.*
 - #define [MAROON](#) 0x8000
- Color.*
 - #define [MEDIUMAQUAMARINE](#) 0x6675
- Color.*
 - #define [MEDIUMBLUE](#) 0x0019
- Color.*
 - #define [MEDIUMORCHID](#) 0xBABA
- Color.*
 - #define [MEDIUMPURPLE](#) 0x939B
- Color.*
 - #define [MEDIUMSEAGREEN](#) 0x3D8E
- Color.*
 - #define [MEDIUMSLATEBLUE](#) 0x7B5D
- Color.*
 - #define [MEDIUMSPRINGGREEN](#) 0x07D3
- Color.*
 - #define [MEDIUMTURQUOISE](#) 0x4E99
- Color.*
 - #define [MEDIUMVIOLETRED](#) 0xC0B0
- Color.*
 - #define [MIDNIGHTBLUE](#) 0x18CE
- Color.*
 - #define [MINTCREAM](#) 0xF7FF
- Color.*
 - #define [MISTYROSE](#) 0xFF3C
- Color.*
 - #define [MOCCASIN](#) 0xFF36
- Color.*

- #define **NAVAJOWHITE** 0xFE5F5
Color.
- #define **NAVY** 0x0010
Color.
- #define **OLDLACE** 0xFFBC
Color.
- #define **OLIVE** 0x8400
Color.
- #define **OLIVEDRAB** 0x6C64
Color.
- #define **ORANGE** 0xFD20
Color.
- #define **ORANGERED** 0xFA20
Color.
- #define **ORCHID** 0xDB9A
Color.
- #define **PALEGOLDENROD** 0xEF55
Color.
- #define **PALEGREEN** 0x9FD3
Color.
- #define **PALETURQUOISE** 0xAF7D
Color.
- #define **PALEVIOLETRED** 0xDB92
Color.
- #define **PAPAYAWHIP** 0xFF7A
Color.
- #define **PEACHPUFF** 0xFED7
Color.
- #define **PERU** 0xCC27
Color.
- #define **PINK** 0xFE19
Color.
- #define **PLUM** 0xDD1B
Color.
- #define **POWDERBLUE** 0xB71C
Color.
- #define **PURPLE** 0x8010
Color.
- #define **RED** 0xF800
Color.
- #define **ROSYBROWN** 0xBC71
Color.
- #define **ROYALBLUE** 0x435C
Color.
- #define **SADDLEBROWN** 0x8A22
Color.
- #define **SALMON** 0xFC0E
Color.
- #define **SANDYBROWN** 0xF52C
Color.
- #define **SEAGREEN** 0x2C4A

- Color.*
 - #define [SEASHELL](#) 0xFFBD
- Color.*
 - #define [SIENNA](#) 0xA285
- Color.*
 - #define [SILVER](#) 0xC618
- Color.*
 - #define [SKYBLUE](#) 0x867D
- Color.*
 - #define [SLATEBLUE](#) 0x6AD9
- Color.*
 - #define [SLATEGRAY](#) 0x7412
- Color.*
 - #define [SNOW](#) 0xFFDF
- Color.*
 - #define [SPRINGGREEN](#) 0x07EF
- Color.*
 - #define [STEELBLUE](#) 0x4416
- Color.*
 - #define [TAN](#) 0xD5B1
- Color.*
 - #define [TEAL](#) 0x0410
- Color.*
 - #define [THISTLE](#) 0xDDFB
- Color.*
 - #define [TOMATO](#) 0xFB08
- Color.*
 - #define [TURQUOISE](#) 0x471A
- Color.*
 - #define [VIOLET](#) 0xEC1D
- Color.*
 - #define [WHEAT](#) 0xF6F6
- Color.*
 - #define [WHITE](#) 0xFFFF
- Color.*
 - #define [WHITESMOKE](#) 0xF7BE
- Color.*
 - #define [YELLOW](#) 0xFFE0
- Color.*
 - #define [YELLOWGREEN](#) 0x9E66
- Color.*

Functions

- void [enableUSARTforOLED](#) ()
Enable and config USART for OLED.
- void [resetOLED](#) ()
Reset OLED.
- void [toggleOLEDInterrupt](#) (unsigned int state)
Toggle OLED interrupt.

- void `sendChar` (int c)
Send char.
- void `usart1_rx` ()
Receive function for OLED data (USART1, interrupt mode)
- void `gfx_Clean` ()
Clean the screen.
- void `gfx_BGcolour` (int color)
Set the background color.
- void `gfx_PutString` (char *string)
Put a string on the screen.
- void `gfx_RectangleFilled` (int x1, int y1, int x2, int y2, int color)
Draw a rectangle filled with a color.
- void `SSTimeout` (int t)
Screensave mode.
- void `setBaudRate` ()
Set the baud rate.
- void `gfx_CalculateOrbit` (int angle, int distance, int *x, int *y)
Calculate the (x,y) pos (orbit) from angle and distance.
- void `gfx_DrawCircle` (int x, int y, int radius, int color)
Draw a circle.
- void `gfx_DrawLine` (int x1, int y1, int x2, int y2, int color)
Draw a line.
- void `gfx_ScreenMode` (int mode)
Screen mode (portrait/landscape)
- void `gfx_MoveOrigin` (int x, int y)
Move to origin to a position.
- void `gfx_Rectangle` (int x1, int y1, int x2, int y2, int color)
Draw a rectangle.
- void `txt_Width` (int multi)
Set the width of the text.
- void `txt_FGColor` (int color)
Set the text color.
- void `txt_BGColor` (int color)
Set the text background color.
- void `initOLED` ()
Configure OLED for proper using.
- void `displayMainMenu` ()
Display menu.
- void `updateMenu` ()
Update the menu with currently selected.
- void `displayMessage` (char *string)
Display a string in the center of the screen.
- void `displayCompass` ()
Display the compass.
- void `displayHeader` ()
Display message header.
- char * `calculateDirection` ()
Calculate the direction (N, S, NE, etc.)
- void `ftoa` (char *p, float x)
Float to string conversion.

Variables

- unsigned int [modeDisplay](#)
- unsigned int [oldModeDisplay](#)
- int [displayHasBeenUpdated](#)

5.8.1 Detailed Description

File containing the OLED functions.

Author

Gaël Foppolo (gaelfoppolo)

5.8.2 Function Documentation

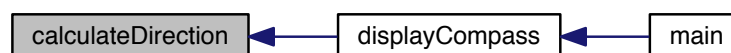
5.8.2.1 `char* calculateDirection ()`

Calculate the direction (N, S, NE, etc.)

Returns

The direction.

Here is the caller graph for this function:



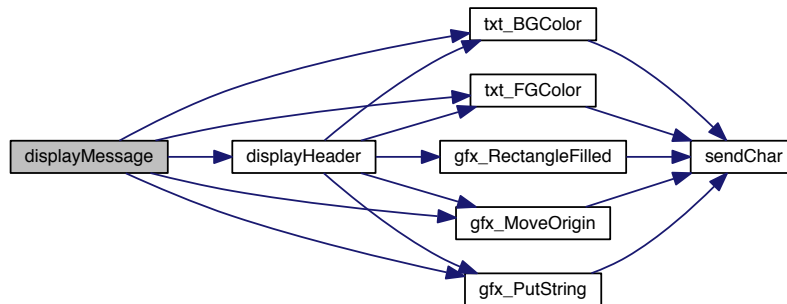
5.8.2.2 `void displayMessage (char * string)`

Display a string in the center of the screen.

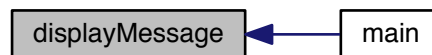
Parameters

<i>string</i>	The string to display
---------------	-----------------------

Here is the call graph for this function:



Here is the caller graph for this function:



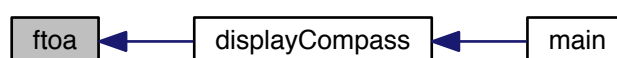
5.8.2.3 void ftoa (char * p, float x)

Float to string conversion.

Parameters

	<i>p</i>	The buffer (string)
in	<i>x</i>	The float

Here is the caller graph for this function:



5.8.2.4 void gfx_BGcolour (int *color*)

Set the background color.

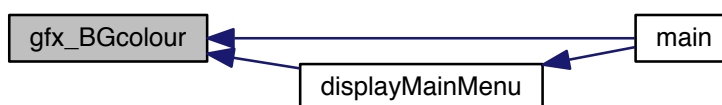
Parameters

in	<i>color</i>	The color
----	--------------	-----------

Here is the call graph for this function:



Here is the caller graph for this function:



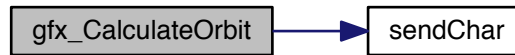
5.8.2.5 void gfx_CalculateOrbit (int *angle*, int *distance*, int * *x*, int * *y*)

Calculate the (x,y) pos (orbit) from angle and distance.

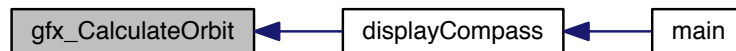
Parameters

in	<i>angle</i>	The angle
in	<i>distance</i>	The distance
	<i>x</i>	The x pos computed
	<i>y</i>	The y pos computed

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.6 void gfx_DrawCircle (int *x*, int *y*, int *radius*, int *color*)

Draw a circle.

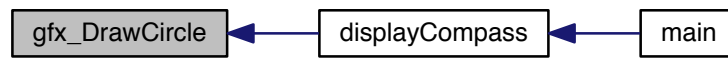
Parameters

in	<i>x</i>	x pos of center of the circle
in	<i>y</i>	y pos of center of the circle
in	<i>radius</i>	The radius
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.7 void gfx_DrawLine (int x1, int y1, int x2, int y2, int color)

Draw a line.

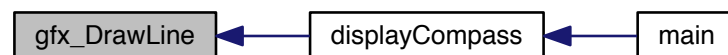
Parameters

in	<i>x1</i>	The x pos of the beginning of the line
in	<i>y1</i>	The y pos of the beginning of the line
in	<i>x2</i>	The x pos of the ending of the line
in	<i>y2</i>	The y pos of the ending of the line
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.8 void gfx_MoveOrigin (int x, int y)

Move to origin to a position.

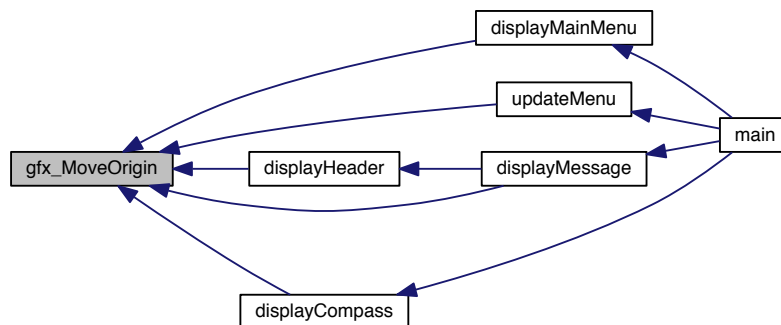
Parameters

in	<i>x</i>	The new x pos
in	<i>y</i>	The new y pos

Here is the call graph for this function:



Here is the caller graph for this function:

**5.8.2.9 void gfx_PutString (char * *string*)**

Put a string on the screen.

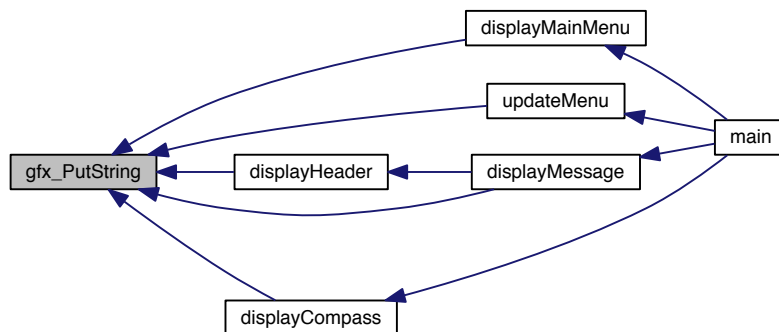
Parameters

<i>string</i>	The string
---------------	------------

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.10 void gfx_Rectangle (int x1, int y1, int x2, int y2, int color)

Draw a rectangle.

Parameters

in	<i>x1</i>	The x pos of the top left corner
in	<i>y1</i>	The y pos of the top left corner
in	<i>x2</i>	The x pos of the bottom right corner
in	<i>y2</i>	The y pos of the bottom right corner
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



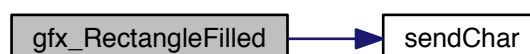
5.8.2.11 void gfx_RectangleFilled (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle filled with a color.

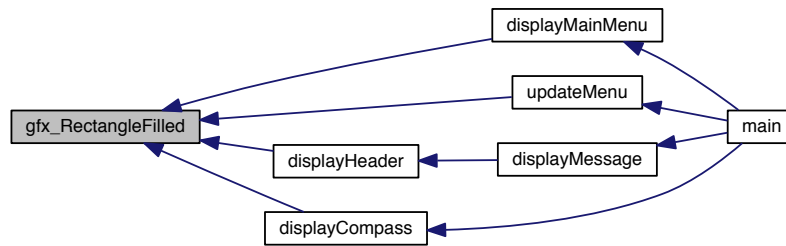
Parameters

in	<i>x1</i>	The x pos of the top left corner
in	<i>y1</i>	The y pos of the top left corner
in	<i>x2</i>	The x pos of the bottom right corner
in	<i>y2</i>	The y pos of the bottom right corner
in	<i>color</i>	The color

Here is the call graph for this function:



Here is the caller graph for this function:



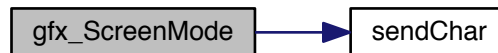
5.8.2.12 void gfx_ScreenMode (int mode)

Screen mode (portrait/landscape)

Parameters

in	<i>mode</i>	The mode
----	-------------	----------

Here is the call graph for this function:



Here is the caller graph for this function:



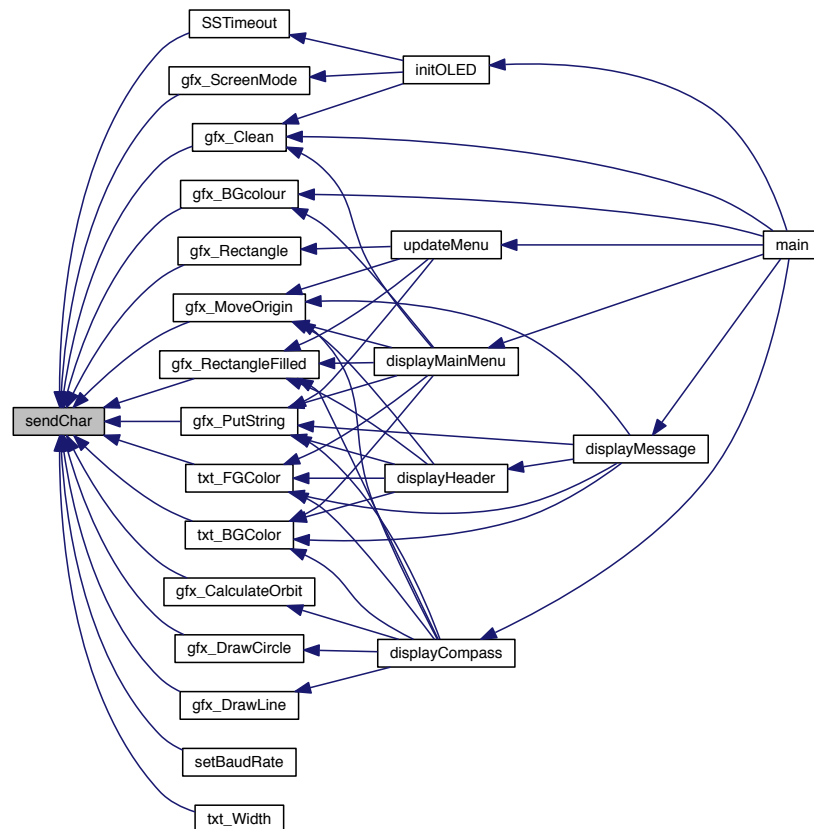
5.8.2.13 void sendChar (int c)

Send char.

Parameters

<code>c</code>	The int to send
----------------	-----------------

Here is the caller graph for this function:



5.8.2.14 void SSTimeout (int t)

Screensave mode.

Parameters

<code>in</code>	<code>t</code>	The mode
-----------------	----------------	----------

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.15 void toggleOLEDInterrupt (unsigned int *state*)

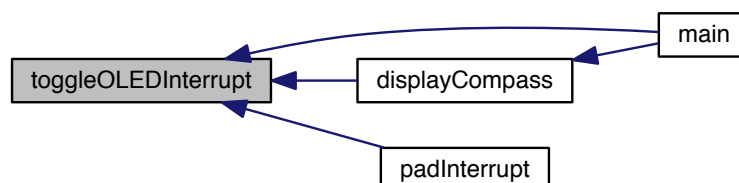
Toggle OLED interrupt.

1 = interrupt enable for OLED, 0 = disable

Parameters

<i>state</i>	The new state
--------------	---------------

Here is the caller graph for this function:



5.8.2.16 void txt_BGColor (int *color*)

Set the text background color.

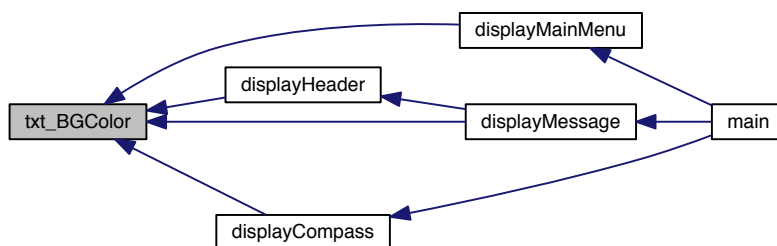
Parameters

in	<i>color</i>	The color
----	--------------	-----------

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.17 void txt_FGColor (int *color*)

Set the text color.

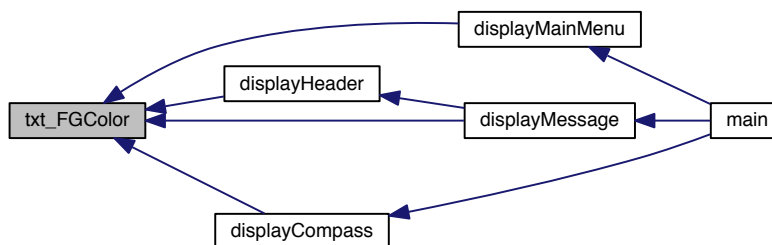
Parameters

in	<i>color</i>	The color
----	--------------	-----------

Here is the call graph for this function:



Here is the caller graph for this function:



5.8.2.18 void txt_Width (int *multi*)

Set the width of the text.

Parameters

in	<i>multi</i>	The multi
----	--------------	-----------

Here is the call graph for this function:



5.8.3 Variable Documentation

5.8.3.1 int displayHasBeenUpdated

Is the display has been updated, aka needed

5.8.3.2 unsigned int modeDisplay

What mode is selected right now

5.8.3.3 unsigned int oldModeDisplay

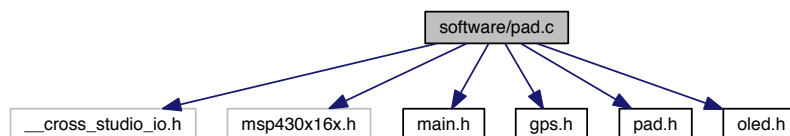
What mode was selected just before

5.9 software/pad.c File Reference

File containing the PAD functions.

```
#include <__cross_studio_io.h>
#include <msp430x16x.h>
#include "main.h"
#include "gps.h"
#include "pad.h"
#include "oled.h"
```

Include dependency graph for pad.c:



Functions

- void [initPAD](#) (void)
Init LED (P2.0 -> P2.4)
- void [padInterrupt](#) (void)
Interrupt function for PAD.

5.9.1 Detailed Description

File containing the PAD functions.

Author

Gaël Foppolo (gaelfoppolo)

5.9.2 Function Documentation

5.9.2.1 void initPAD (void)

Init LED (P2.0 -> P2.4)

All ready to use and state cleared

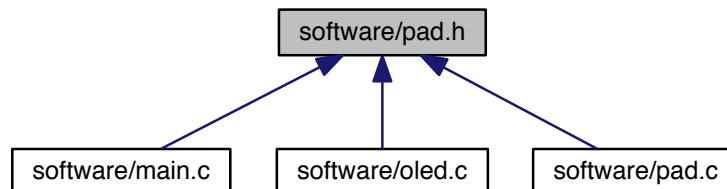
Here is the caller graph for this function:



5.10 software/pad.h File Reference

File containing the PAD functions.

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



Macros

- `#define` `PUSH` `0x1E`
PUSH position.
- `#define` `TOP` `0x1D`
TOP position.
- `#define` `RIGHT` `0x0F`
RIGHT position.
- `#define` `BOTTOM` `0x1B`
BOTTOM position.
- `#define` `LEFT` `0x17`
LEFT position.

Functions

- void `initPAD` (void)
Init LED (P2.0 -> P2.4)
- void `padInterrupt` (void)
Interrupt function for PAD.

5.10.1 Detailed Description

File containing the PAD functions.

Author

Gaël Foppolo (gaelfoppolo)

5.10.2 Function Documentation

5.10.2.1 void `initPAD` (void)

Init LED (P2.0 -> P2.4)

All ready to use and state cleared

Here is the caller graph for this function:



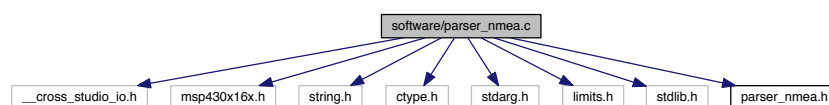
5.11 software/parser_nmea.c File Reference

File containing the NMEA parser functions.

```

#include <__cross_studio_io.h>
#include <msp430x16x.h>
#include <string.h>
#include <ctype.h>
#include <stdarg.h>
#include <limits.h>
#include <stdlib.h>
#include "parser_nmea.h"
  
```

Include dependency graph for `parser_nmea.c`:



Functions

- int [hex2int](#) (char c)
Transform hexa to integer.
- enum [nmea_sentence_id](#) [nmea_sentence_id](#) (char *sentence)
Determine sentence identifier.
- int [nmea_isfield](#) (char c)
Check if the char is part of the field.
- int [nmea_scan](#) (const char *sentence, const char *format,...)
Scanf-like processor for NMEA sentences.
- int [nmea_parse_rmc](#) (struct [nmea_sentence_rmc](#) *frame, const char *sentence)
Parse a RMC sentence.
- int [nmea_parse_gga](#) (struct [nmea_sentence_gga](#) *frame, const char *sentence)
Parse a GGA sentence.
- int [nmea_check](#) (const char *sentence, int strict)
Check sentence validity and checksum.

5.11.1 Detailed Description

File containing the NMEA parser functions.

Author

Gaël Foppolo (gaelfoppolo)

5.11.2 Function Documentation

5.11.2.1 int hex2int (char c)

Transform hexa to integer.

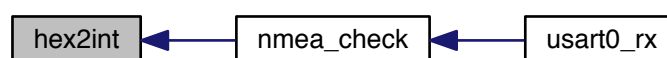
Parameters

in	c	An integer (char)
----	---	-------------------

Returns

An integer (hex)

Here is the caller graph for this function:



5.11.2.2 int nmea_check (const char * *sentence*, int *strict*)

Check sentence validity and checksum.

Calculate checksum and compare it

Parameters

<i>sentence</i>	The sentence to test
<i>strict</i>	Accept or not sentence with checksum

Returns

1 for valid sentences.

Here is the call graph for this function:



Here is the caller graph for this function:



5.11.2.3 int nmea_isfield (char *c*)

Check if the char is part of the field.

Aka char isn't a comma or star

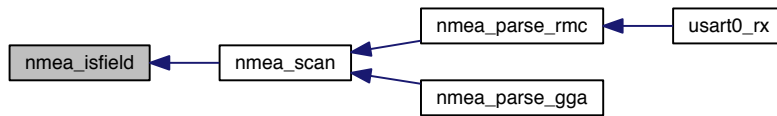
Parameters

<i>c</i>	The char to test
----------	------------------

Returns

1 is valid, 0 if not

Here is the caller graph for this function:



5.11.2.4 int nmea_parse_gga (nmea_sentence_gga * *frame*, const char * *sentence*)

Parse a GGA sentence.

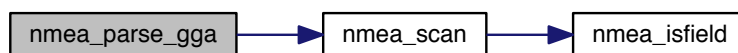
Parameters

<i>frame</i>	The struct where to put the parsed data
<i>sentence</i>	The sentence to parse

Returns

1 on success

Here is the call graph for this function:



5.11.2.5 int nmea_parse_rmc (nmea_sentence_rmc * *frame*, const char * *sentence*)

Parse a RMC sentence.

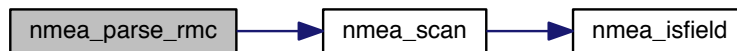
Parameters

<i>frame</i>	The struct where to put the parsed data
<i>sentence</i>	The sentence to parse

Returns

1 on success

Here is the call graph for this function:



Here is the caller graph for this function:

**5.11.2.6 int nmea_scan (const char * *sentence*, const char * *format*, ...)**

Scanf-like processor for NMEA sentences.

Supports the following formats: c - single character (char *) d - direction, returned as 1/-1, default 0 (int *) f - float (float *) o - longitude value, transform all in degrees (float *) a - latitude value, transform all in degrees (float *) i - decimal (integer), default zero (int *) s - string (char *) t - talker identifier and type (char *)

Parameters

<i>sentence</i>	The sentence to parse
<i>format</i>	The format of the sentence

See also

nmea_parse_*** functions for further explanations

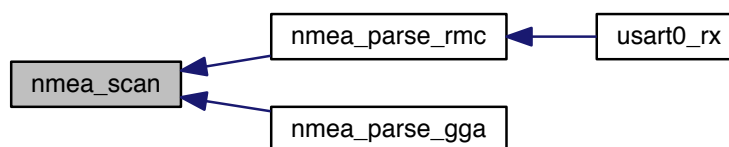
Returns

1 on success, 1 if not

Here is the call graph for this function:



Here is the caller graph for this function:

**5.11.2.7 enum nmea_sentence_id nmea_sentence_id (char * sentence)**

Determine sentence identifier.

Parameters

<i>sentence</i>	Then sentence to test
-----------------	-----------------------

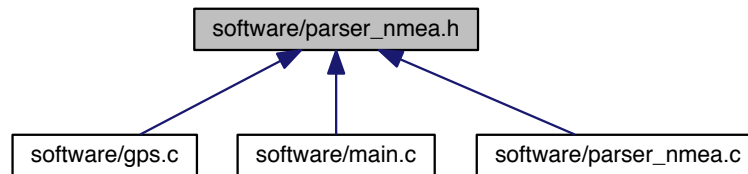
Returns

The type of sentence

5.12 software/parser_nmea.h File Reference

File containing the NMEA parser functions.

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



Data Structures

- struct [nmea_sentence_rmc](#)
- struct [nmea_sentence_gga](#)

Macros

- `#define NMEA_MAX_LENGTH 100`
Maximum length for NMEA sentences.

Typedefs

- typedef struct [nmea_sentence_rmc](#) [nmea_sentence_rmc](#)
- typedef struct [nmea_sentence_gga](#) [nmea_sentence_gga](#)

Enumerations

- enum [nmea_sentence_id](#) {
NMEA_INVALID = -1, **NMEA_UNKNOWN** = 0, **NMEA_SENTENCE_RMC**, **NMEA_SENTENCE_GGA**,
NMEA_SENTENCE_GSA, **NMEA_SENTENCE_GLL**, **NMEA_SENTENCE_GST**, **NMEA_SENTENCE_GSV**,
NMEA_SENTENCE_VTG }

Functions

- int [hex2int](#) (char c)
Transform hexa to integer.
- int [nmea_scan](#) (const char *sentence, const char *format,...)
Scanf-like processor for NMEA sentences.
- int [nmea_isfield](#) (char c)
Check if the char is part of the field.
- enum [nmea_sentence_id](#) [nmea_sentence_id](#) (char *sentence)
Determine sentence identifier.
- int [nmea_check](#) (const char *sentence, int strict)
Check sentence validity and checksum.
- int [nmea_parse_rmc](#) ([nmea_sentence_rmc](#) *frame, const char *sentence)
Parse a RMC sentence.
- int [nmea_parse_gga](#) ([nmea_sentence_gga](#) *frame, const char *sentence)
Parse a GGA sentence.

5.12.1 Detailed Description

File containing the NMEA parser functions.

Author

Gaël Foppolo (gaelfoppolo)

5.12.2 Typedef Documentation

5.12.2.1 typedef struct nmea_sentence_gga nmea_sentence_gga

The structure that contains the data of GGA sentences

5.12.2.2 typedef struct nmea_sentence_rmc nmea_sentence_rmc

The structure that contains the data of RMC sentences

5.12.3 Enumeration Type Documentation

5.12.3.1 enum nmea_sentence_id

The sentence identifier

5.12.4 Function Documentation

5.12.4.1 int hex2int (char c)

Transform hexa to integer.

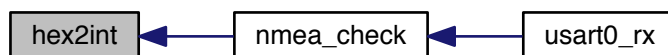
Parameters

in	c	An integer (char)
----	---	-------------------

Returns

An integer (hex)

Here is the caller graph for this function:



5.12.4.2 int nmea_check (const char * *sentence*, int *strict*)

Check sentence validity and checksum.

Calculate checksum and compare it

Parameters

<i>sentence</i>	The sentence to test
<i>strict</i>	Accept or not sentence with checksum

Returns

1 for valid sentences.

Here is the call graph for this function:



Here is the caller graph for this function:



5.12.4.3 int nmea_isfield (char c)

Check if the char is part of the field.

Aka char isn't a comma or star

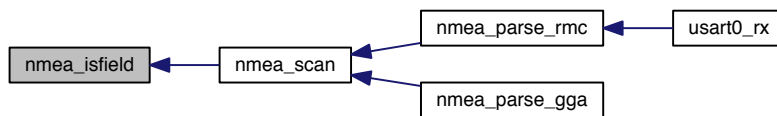
Parameters

c	The char to test
---	------------------

Returns

1 is valid, 0 if not

Here is the caller graph for this function:



5.12.4.4 int nmea_parse_gga (nmea_sentence_gga * frame, const char * sentence)

Parse a GGA sentence.

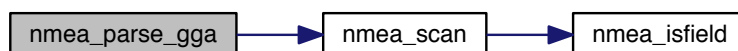
Parameters

<i>frame</i>	The struct where to put the parsed data
<i>sentence</i>	The sentence to parse

Returns

1 on success

Here is the call graph for this function:



5.12.4.5 int nmea_parse_rmc (nmea_sentence_rmc * *frame*, const char * *sentence*)

Parse a RMC sentence.

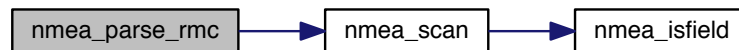
Parameters

<i>frame</i>	The struct where to put the parsed data
<i>sentence</i>	The sentence to parse

Returns

1 on success

Here is the call graph for this function:



Here is the caller graph for this function:



5.12.4.6 int nmea_scan (const char * *sentence*, const char * *format*, ...)

Scanf-like processor for NMEA sentences.

Supports the following formats: c - single character (char *) d - direction, returned as 1/-1, default 0 (int *) f - float (float *) o - longitude value, transform all in degrees (float *) a - latitude value, transform all in degrees (float *) i - decimal (integer), default zero (int *) s - string (char *) t - talker identifier and type (char *)

Parameters

<i>sentence</i>	The sentence to parse
<i>format</i>	The format of the sentence

See also

nmea_parse_*** functions for further explanations

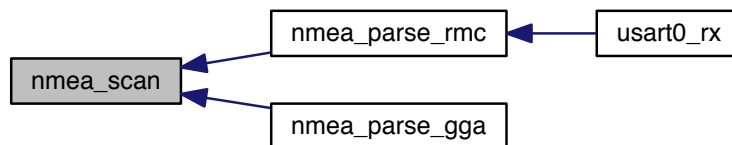
Returns

1 on success, 1 if not

Here is the call graph for this function:



Here is the caller graph for this function:

**5.12.4.7 enum nmea_sentence_id nmea_sentence_id (char * sentence)**

Determine sentence identifier.

Parameters

<i>sentence</i>	Then sentence to test
-----------------	-----------------------

Returns

The type of sentence

Index

calcDistance
 gps.c, 10
 gps.h, 15
calculateDirection
 oled.c, 30
 oled.h, 51

dataValid
 gps.c, 13
 gps.h, 18
deg2rad
 gps.c, 10
 gps.h, 16
delay
 main.c, 22
 main.h, 26
displayHasBeenUpdated
 oled.c, 42
 oled.h, 63
displayMessage
 oled.c, 30
 oled.h, 51

ftoa
 oled.c, 31
 oled.h, 52

GPSPData
 gps.c, 13
 gps.h, 18
gfx_BGcolour
 oled.c, 31
 oled.h, 52
gfx_CalculateOrbit
 oled.c, 32
 oled.h, 53
gfx_DrawCircle
 oled.c, 33
 oled.h, 54
gfx_DrawLine
 oled.c, 34
 oled.h, 55
gfx_MoveOrigin
 oled.c, 34
 oled.h, 55
gfx_PutString
 oled.c, 35
 oled.h, 56
gfx_Rectangle
 oled.c, 36

 oled.h, 57
gfx_RectangleFilled
 oled.c, 37
 oled.h, 58
gfx_ScreenMode
 oled.c, 38
 oled.h, 59
gps.c
 calcDistance, 10
 dataValid, 13
 deg2rad, 10
 GPSPData, 13
 gpsSend, 12
 toggleGPSInterrupt, 13
 toggleGPS, 12
gps.h
 calcDistance, 15
 dataValid, 18
 deg2rad, 16
 GPSPData, 18
 gps_data, 15
 gpsSend, 16
 toggleGPSInterrupt, 17
 toggleGPS, 17
gps_data, 7
 gps.h, 15
gpsSend
 gps.c, 12
 gps.h, 16

hex2int
 parser_nmea.c, 67
 parser_nmea.h, 73

initLED
 led.c, 19
 led.h, 21
initPAD
 pad.c, 65
 pad.h, 66

led.c
 initLED, 19
 toggleLED, 19
led.h
 initLED, 21
 toggleLED, 21

main
 main.c, 23

- main.c
 - delay, [22](#)
 - main, [23](#)
 - modeSelected, [25](#)
 - toggleCommunication, [24](#)
- main.h
 - delay, [26](#)
 - modeSelected, [27](#)
 - toggleCommunication, [27](#)
- modeDisplay
 - oled.c, [43](#)
 - oled.h, [64](#)
- modeSelected
 - main.c, [25](#)
 - main.h, [27](#)
- nmea_check
 - parser_nmea.c, [67](#)
 - parser_nmea.h, [74](#)
- nmea_isfield
 - parser_nmea.c, [68](#)
 - parser_nmea.h, [74](#)
- nmea_parse_gga
 - parser_nmea.c, [69](#)
 - parser_nmea.h, [75](#)
- nmea_parse_rmc
 - parser_nmea.c, [69](#)
 - parser_nmea.h, [75](#)
- nmea_scan
 - parser_nmea.c, [70](#)
 - parser_nmea.h, [76](#)
- nmea_sentence_gga, [7](#)
 - parser_nmea.h, [73](#)
- nmea_sentence_id
 - parser_nmea.c, [71](#)
 - parser_nmea.h, [73](#), [77](#)
- nmea_sentence_rmc, [8](#)
 - parser_nmea.h, [73](#)
- oldModeDisplay
 - oled.c, [43](#)
 - oled.h, [64](#)
- oled.c
 - calculateDirection, [30](#)
 - displayHasBeenUpdated, [42](#)
 - displayMessage, [30](#)
 - ftoa, [31](#)
 - gfx_BGcolour, [31](#)
 - gfx_CalculateOrbit, [32](#)
 - gfx_DrawCircle, [33](#)
 - gfx_DrawLine, [34](#)
 - gfx_MoveOrigin, [34](#)
 - gfx_PutString, [35](#)
 - gfx_Rectangle, [36](#)
 - gfx_RectangleFilled, [37](#)
 - gfx_ScreenMode, [38](#)
 - modeDisplay, [43](#)
 - oldModeDisplay, [43](#)
 - SSTimeout, [39](#)
 - sendChar, [38](#)
 - toggleOLEDInterrupt, [40](#)
 - txt_BGColor, [40](#)
 - txt_FGColor, [41](#)
 - txt_Width, [42](#)
- oled.h
 - calculateDirection, [51](#)
 - displayHasBeenUpdated, [63](#)
 - displayMessage, [51](#)
 - ftoa, [52](#)
 - gfx_BGcolour, [52](#)
 - gfx_CalculateOrbit, [53](#)
 - gfx_DrawCircle, [54](#)
 - gfx_DrawLine, [55](#)
 - gfx_MoveOrigin, [55](#)
 - gfx_PutString, [56](#)
 - gfx_Rectangle, [57](#)
 - gfx_RectangleFilled, [58](#)
 - gfx_ScreenMode, [59](#)
 - modeDisplay, [64](#)
 - oldModeDisplay, [64](#)
 - SSTimeout, [60](#)
 - sendChar, [59](#)
 - toggleOLEDInterrupt, [61](#)
 - txt_BGColor, [61](#)
 - txt_FGColor, [62](#)
 - txt_Width, [63](#)
- pad.c
 - initPAD, [65](#)
- pad.h
 - initPAD, [66](#)
- parser_nmea.c
 - hex2int, [67](#)
 - nmea_check, [67](#)
 - nmea_isfield, [68](#)
 - nmea_parse_gga, [69](#)
 - nmea_parse_rmc, [69](#)
 - nmea_scan, [70](#)
 - nmea_sentence_id, [71](#)
- parser_nmea.h
 - hex2int, [73](#)
 - nmea_check, [74](#)
 - nmea_isfield, [74](#)
 - nmea_parse_gga, [75](#)
 - nmea_parse_rmc, [75](#)
 - nmea_scan, [76](#)
 - nmea_sentence_gga, [73](#)
 - nmea_sentence_id, [73](#), [77](#)
 - nmea_sentence_rmc, [73](#)
- SSTimeout
 - oled.c, [39](#)
 - oled.h, [60](#)
- sendChar
 - oled.c, [38](#)
 - oled.h, [59](#)
- software/gps.c, [9](#)
- software/gps.h, [14](#)

- software/led.c, [18](#)
- software/led.h, [20](#)
- software/main.c, [21](#)
- software/main.h, [25](#)
- software/oled.c, [28](#)
- software/oled.h, [43](#)
- software/pad.c, [64](#)
- software/pad.h, [65](#)
- software/parser_nmea.c, [66](#)
- software/parser_nmea.h, [71](#)

- toggleCommunication
 - main.c, [24](#)
 - main.h, [27](#)
- toggleGPSInterrupt
 - gps.c, [13](#)
 - gps.h, [17](#)
- toggleGPS
 - gps.c, [12](#)
 - gps.h, [17](#)
- toggleLED
 - led.c, [19](#)
 - led.h, [21](#)
- toggleOLEDInterrupt
 - oled.c, [40](#)
 - oled.h, [61](#)
- txt_BGColor
 - oled.c, [40](#)
 - oled.h, [61](#)
- txt_FGColor
 - oled.c, [41](#)
 - oled.h, [62](#)
- txt_Width
 - oled.c, [42](#)
 - oled.h, [63](#)