

GPS

Generated by Doxygen 1.8.11

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	gps_data Struct Reference	5
3.1.1	Detailed Description	5
3.2	nmea_sentence_gga Struct Reference	5
3.2.1	Detailed Description	6
3.3	nmea_sentence_rmc Struct Reference	6
3.3.1	Detailed Description	6
4	File Documentation	7
4.1	gps.c File Reference	7
4.1.1	Detailed Description	8
4.1.2	Function Documentation	8
4.1.2.1	calcDistance(float lat1, float lon1, float lat2, float lon2)	8
4.1.2.2	deg2rad(float deg)	9
4.1.2.3	gpsSend(char *message)	10
4.1.2.4	toggleGPS(unsigned int state)	10
4.1.2.5	toggleGPSInterrupt(unsigned int state)	11
4.1.3	Variable Documentation	11
4.1.3.1	dataValid	11

4.1.3.2	GPSData	11
4.2	gps.h File Reference	12
4.2.1	Detailed Description	13
4.2.2	Typedef Documentation	13
4.2.2.1	gps_data	13
4.2.3	Function Documentation	13
4.2.3.1	calcDistance(float lat1, float lon1, float lat2, float lon2)	13
4.2.3.2	deg2rad(float deg)	14
4.2.3.3	gpsSend(char *message)	14
4.2.3.4	toggleGPS(unsigned int state)	15
4.2.3.5	toggleGPSInterrupt(unsigned int state)	15
4.2.4	Variable Documentation	16
4.2.4.1	dataValid	16
4.2.4.2	GPSData	16
4.3	led.c File Reference	16
4.3.1	Detailed Description	17
4.3.2	Function Documentation	17
4.3.2.1	initLED(void)	17
4.3.2.2	toggleLED(int n, unsigned int state, unsigned int duration)	17
4.4	led.h File Reference	18
4.4.1	Detailed Description	18
4.4.2	Function Documentation	19
4.4.2.1	initLED(void)	19
4.4.2.2	toggleLED(int n, unsigned int state, unsigned int duration)	19
4.5	main.c File Reference	19
4.5.1	Detailed Description	20
4.5.2	Function Documentation	20
4.5.2.1	delay(float x)	20
4.5.2.2	main(void)	21
4.5.2.3	toggleCommunication(unsigned int state)	22

4.5.3	Variable Documentation	23
4.5.3.1	modeSelected	23
4.6	main.h File Reference	23
4.6.1	Detailed Description	24
4.6.2	Function Documentation	24
4.6.2.1	delay(float x)	24
4.6.2.2	toggleCommunication(unsigned int state)	25
4.6.3	Variable Documentation	25
4.6.3.1	modeSelected	26
4.7	oled.c File Reference	26
4.7.1	Detailed Description	28
4.7.2	Function Documentation	28
4.7.2.1	calculateDirection()	28
4.7.2.2	displayMessage(char *string)	28
4.7.2.3	ftoa(char *p, float x)	29
4.7.2.4	gfx_BGcolour(int color)	30
4.7.2.5	gfx_CalculateOrbit(int angle, int distance, int *x, int *y)	30
4.7.2.6	gfx_DrawCircle(int x, int y, int radius, int color)	31
4.7.2.7	gfx_DrawLine(int x1, int y1, int x2, int y2, int color)	32
4.7.2.8	gfx_MoveOrigin(int x, int y)	32
4.7.2.9	gfx_PutString(char *string)	33
4.7.2.10	gfx_Rectangle(int x1, int y1, int x2, int y2, int color)	34
4.7.2.11	gfx_RectangleFilled(int x1, int y1, int x2, int y2, int color)	35
4.7.2.12	gfx_ScreenMode(int mode)	36
4.7.2.13	sendChar(int c)	36
4.7.2.14	SSTimeout(int t)	37
4.7.2.15	toggleOLEDInterrupt(unsigned int state)	38
4.7.2.16	txt_BGColor(int color)	39
4.7.2.17	txt_FGColor(int color)	39
4.7.2.18	txt_Width(int multi)	40

4.7.3	Variable Documentation	40
4.7.3.1	displayHasBeenUpdated	41
4.7.3.2	modeDisplay	41
4.7.3.3	oldModeDisplay	41
4.8	oled.h File Reference	41
4.8.1	Detailed Description	49
4.8.2	Function Documentation	49
4.8.2.1	calculateDirection()	49
4.8.2.2	displayMessage(char *string)	49
4.8.2.3	ftoa(char *p, float x)	50
4.8.2.4	gfx_BGcolour(int color)	51
4.8.2.5	gfx_CalculateOrbit(int angle, int distance, int *x, int *y)	51
4.8.2.6	gfx_DrawCircle(int x, int y, int radius, int color)	52
4.8.2.7	gfx_DrawLine(int x1, int y1, int x2, int y2, int color)	53
4.8.2.8	gfx_MoveOrigin(int x, int y)	54
4.8.2.9	gfx_PutString(char *string)	55
4.8.2.10	gfx_Rectangle(int x1, int y1, int x2, int y2, int color)	55
4.8.2.11	gfx_RectangleFilled(int x1, int y1, int x2, int y2, int color)	56
4.8.2.12	gfx_ScreenMode(int mode)	57
4.8.2.13	sendChar(int c)	57
4.8.2.14	SSTimeout(int t)	58
4.8.2.15	toggleOLEDInterrupt(unsigned int state)	59
4.8.2.16	txt_BGColor(int color)	60
4.8.2.17	txt_FGColor(int color)	60
4.8.2.18	txt_Width(int multi)	61
4.8.3	Variable Documentation	61
4.8.3.1	displayHasBeenUpdated	62
4.8.3.2	modeDisplay	62
4.8.3.3	oldModeDisplay	62
4.9	pad.c File Reference	62

4.9.1	Detailed Description	62
4.9.2	Function Documentation	63
4.9.2.1	initPAD(void)	63
4.10	pad.h File Reference	63
4.10.1	Detailed Description	64
4.10.2	Function Documentation	64
4.10.2.1	initPAD(void)	64
4.11	parser_nmea.c File Reference	64
4.11.1	Detailed Description	65
4.11.2	Function Documentation	65
4.11.2.1	hex2int(char c)	65
4.11.2.2	nmea_check(const char *sentence, int strict)	66
4.11.2.3	nmea_isfield(char c)	66
4.11.2.4	nmea_parse_gga(struct nmea_sentence_gga *frame, const char *sentence)	67
4.11.2.5	nmea_parse_rmc(struct nmea_sentence_rmc *frame, const char *sentence)	67
4.11.2.6	nmea_scan(const char *sentence, const char *format,...)	68
4.11.2.7	nmea_sentence_id(char *sentence)	69
4.12	parser_nmea.h File Reference	69
4.12.1	Detailed Description	71
4.12.2	Typedef Documentation	71
4.12.2.1	nmea_sentence_gga	71
4.12.2.2	nmea_sentence_rmc	71
4.12.3	Enumeration Type Documentation	71
4.12.3.1	nmea_sentence_id	71
4.12.4	Function Documentation	71
4.12.4.1	hex2int(char c)	71
4.12.4.2	nmea_check(const char *sentence, int strict)	72
4.12.4.3	nmea_isfield(char c)	73
4.12.4.4	nmea_parse_gga(nmea_sentence_gga *frame, const char *sentence)	73
4.12.4.5	nmea_parse_rmc(nmea_sentence_rmc *frame, const char *sentence)	74
4.12.4.6	nmea_scan(const char *sentence, const char *format,...)	74
4.12.4.7	nmea_sentence_id(char *sentence)	75

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

gps_data	5
nmea_sentence_gga	5
nmea_sentence_rmc	6

Chapter 2

File Index

2.1 File List

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

gps.c	File containing the GPS functions	7
gps.h	File containing the GPS functions	12
led.c	File containing the LED functions	16
led.h	File containing the LED functions	18
main.c	File containing the main functions	19
main.h	File containing the main functions	23
oled.c	File containing the OLED functions	26
oled.h	File containing the OLED functions	41
pad.c	File containing the PAD functions	62
pad.h	File containing the PAD functions	63
parser_nmea.c	File containing the NMEA parser functions	64
parser_nmea.h	File containing the NMEA parser functions	69

Chapter 3

Data Structure Documentation

3.1 `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.

3.1.1 Detailed Description

Structure that contains useful GPS data

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

- `gps.h`

3.2 `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.

3.2.1 Detailed Description

The structure that contains the data of GGA sentences

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

- [parser_nmea.h](#)

3.3 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.

3.3.1 Detailed Description

The structure that contains the data of RMC sentences

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

- [parser_nmea.h](#)

Chapter 4

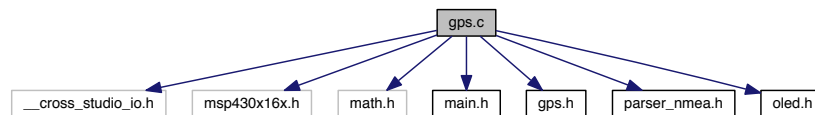
File Documentation

4.1 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.

4.1.1 Detailed Description

File containing the GPS functions.

Author

Gaël Foppolo (gaelfoppolo)

4.1.2 Function Documentation

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

Calculate the distance between two points (Haversine formula)

See also

[Wikipedia](#)

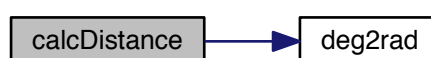
Parameters

<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:



4.1.2.2 `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:

**4.1.2.3 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:

**4.1.2.4 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:



4.1.2.5 void toggleGPSInterrupt (unsigned int *state*)

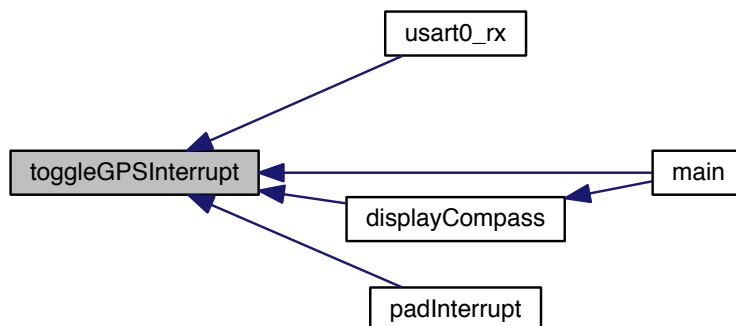
Toggle GPS interrupt.

1 = interrupt enable for GPS, 0 = disable

Parameters

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

Here is the caller graph for this function:



4.1.3 Variable Documentation

4.1.3.1 unsigned int dataValid

Data are valid or not?

4.1.3.2 gps_data GPSTData

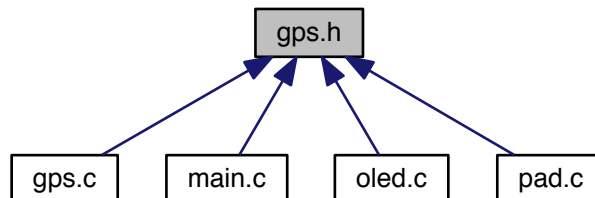
Store GPS valid data.

Useful data received and valid.

4.2 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 "$PMTK314,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*28\r\n"`
Tell the GPS to not send sentences.
- `#define TURN_ALL "$PMTK314,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0*28\r\n"`
Tell the GPS to send all sentences.
- `#define GGA_RMC "$PMTK314,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0*28\r\n"`
Tell the GPS to send GGA and RMC sentences only.
- `#define RMC_ONLY "$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0*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.

4.2.1 Detailed Description

File containing the GPS functions.

Author

Gaël Foppolo (gaelfoppolo)

4.2.2 Typedef Documentation

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

Structure that contains useful GPS data

4.2.3 Function Documentation

4.2.3.1 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:

4.2.3.2 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:

4.2.3.3 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:



4.2.3.4 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:



4.2.3.5 void toggleGPSInterrupt (unsigned int *state*)

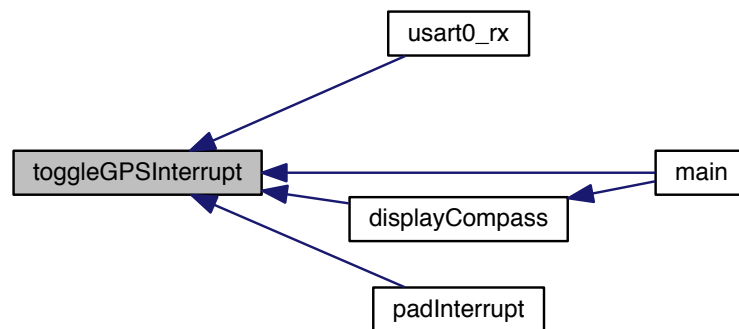
Toggle GPS interrupt.

1 = interrupt enable for GPS, 0 = disable

Parameters

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

Here is the caller graph for this function:



4.2.4 Variable Documentation

4.2.4.1 unsigned int dataValid

Data are valid or not?

4.2.4.2 struct gps_data GPSPData

Useful data received and valid.

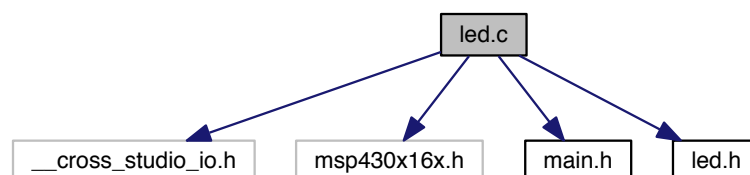
Useful data received and valid.

4.3 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.

4.3.1 Detailed Description

File containing the LED functions.

Author

Gaël Foppolo (gaelfoppolo)

4.3.2 Function Documentation

4.3.2.1 void `initLED` (void)

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

All ready to use and state cleared

Here is the caller graph for this function:



4.3.2.2 void `toggleLED` (int n, unsigned int state, unsigned int duration)

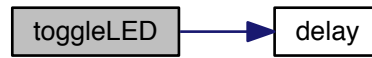
Toogle the state of the choosen LED for a choosen time.

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

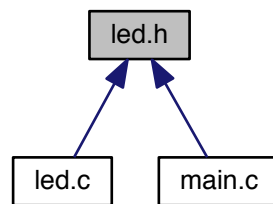
Here is the call graph for this function:



4.4 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.

4.4.1 Detailed Description

File containing the LED functions.

Author

Gaël Foppolo (gaelfoppolo)

4.4.2 Function Documentation

4.4.2.1 void initLED (void)

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

All ready to use and state cleared

Here is the caller graph for this function:



4.4.2.2 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:

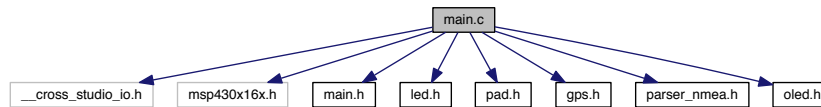


4.5 main.c File Reference

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](#)

4.5.1 Detailed Description

File containing the main functions.

Author

Gaël Foppolo (gaelfoppolo)

4.5.2 Function Documentation

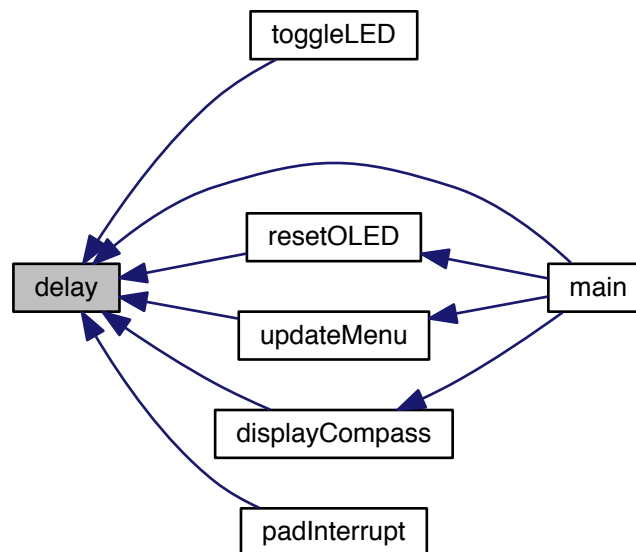
4.5.2.1 void delay (float x)

Wait for x sec.

Parameters

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

Here is the caller graph for this function:



4.5.2.2 void main (void)

Menu entry point

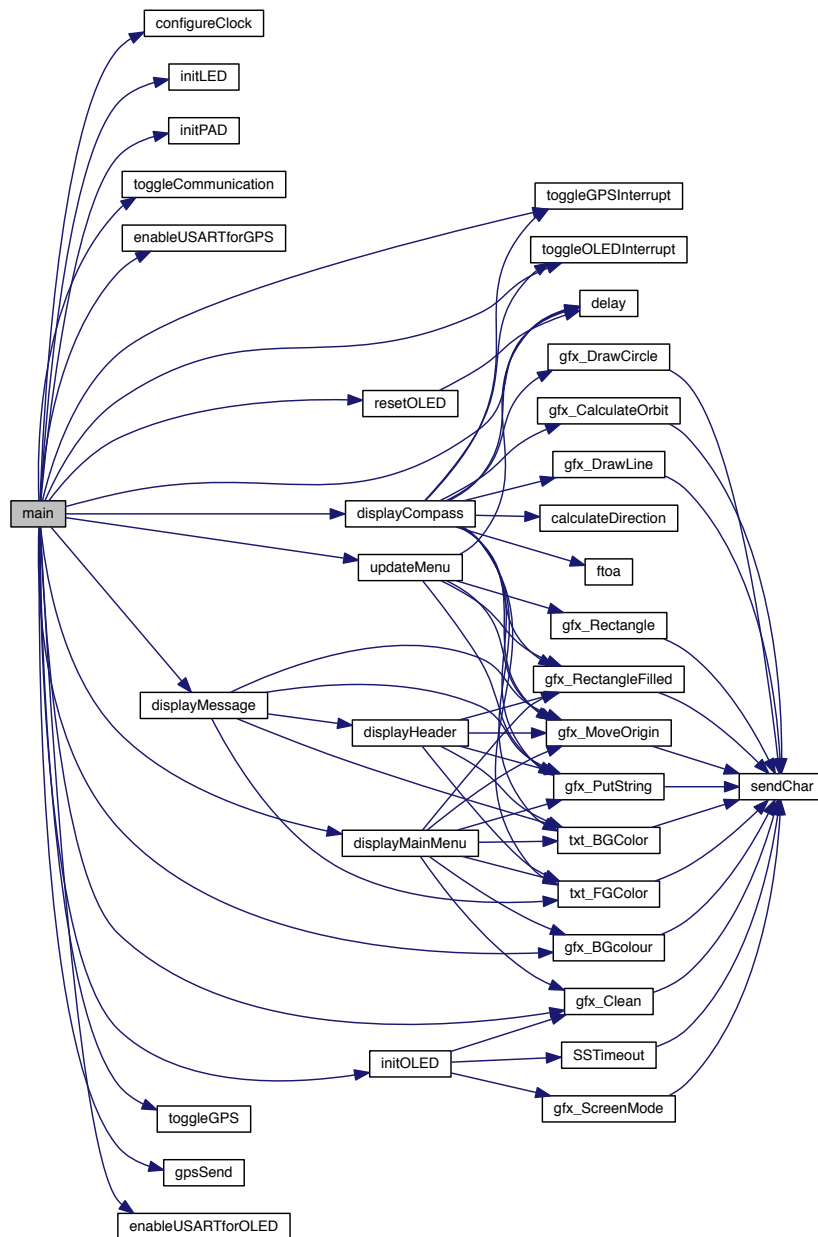
Compass entry point

Navigation entry point

Record entry point

Shutdown entry point

Here is the call graph for this function:



4.5.2.3 void toggleCommunication (unsigned int state)

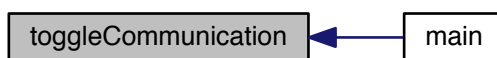
Toggle the communication (P4.2, CMD_SWITCH)

1 = USB, 0 = GPS

Parameters

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

Here is the caller graph for this function:



4.5.3 Variable Documentation

4.5.3.1 unsigned int modeSelected

Mode selected by the user,

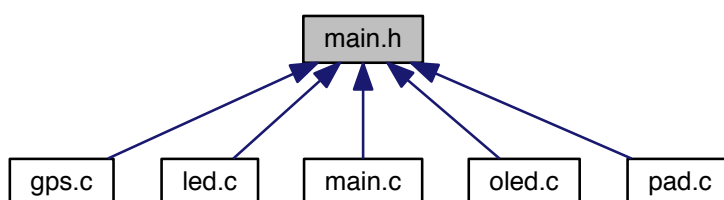
See also

[M_MENU](#), etc.

4.6 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`

4.6.1 Detailed Description

File containing the main functions.

Author

Gaël Foppolo (gaelfoppolo)

4.6.2 Function Documentation

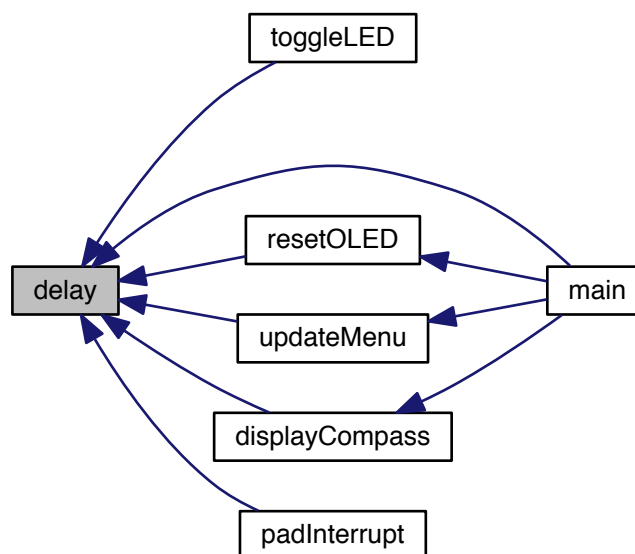
4.6.2.1 void delay (float x)

Wait for x sec.

Parameters

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

Here is the caller graph for this function:



4.6.2.2 void toggleCommunication (unsigned int *state*)

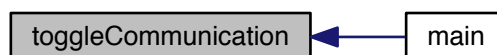
Toogle the communication (P4.2, CMD_SWITCH)

1 = USB, 0 = GPS

Parameters

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

Here is the caller graph for this function:



4.6.3 Variable Documentation

4.6.3.1 unsigned int modeSelected

Mode selected by the user,

See also

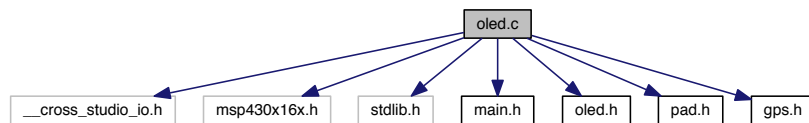
[M_MENU](#), etc.

4.7 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?

4.7.1 Detailed Description

File containing the OLED functions.

Author

Gaël Foppolo (gaelfoppolo)

4.7.2 Function Documentation

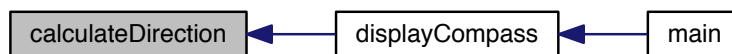
4.7.2.1 `char* calculateDirection ()`

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

Returns

The direction.

Here is the caller graph for this function:



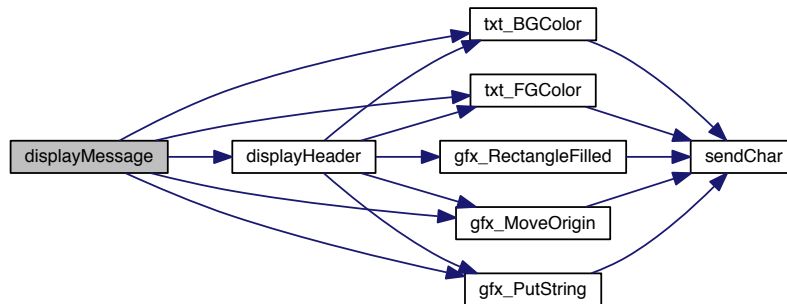
4.7.2.2 `void displayMessage (char * string)`

Display a string in the center of the screen.

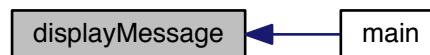
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



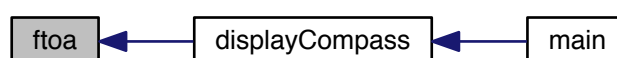
4.7.2.3 void ftoa (char * p, float x)

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:



4.7.2.4 void gfx_BGcolour (int *color*)

Set the background color.

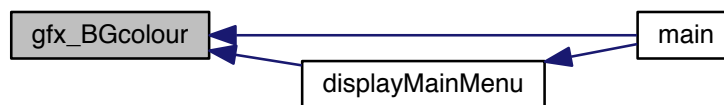
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



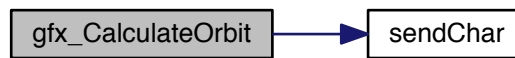
4.7.2.5 void gfx_CalculateOrbit (int *angle*, int *distance*, int * *x*, int * *y*)

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

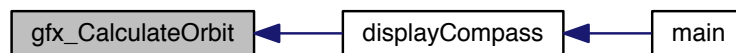
Parameters

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:



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

Draw a circle.

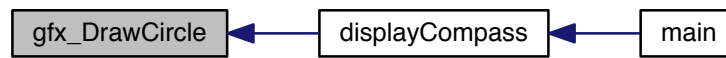
Parameters

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:



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

Draw a line.

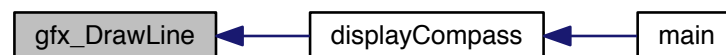
Parameters

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:



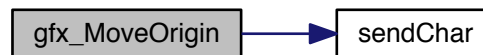
4.7.2.8 void gfx_MoveOrigin (int x, int y)

Move to origin to a position.

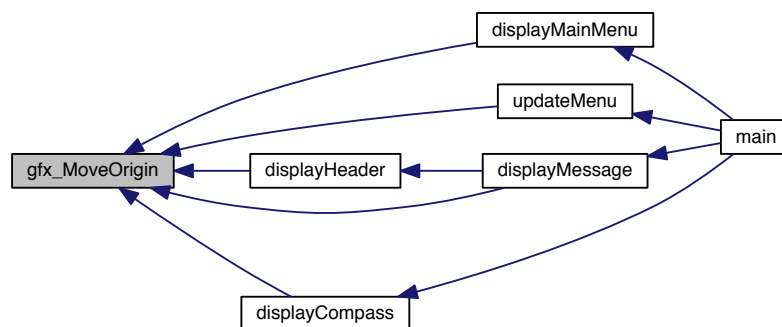
Parameters

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:



4.7.2.9 void gfx_PutString (char * *string*)

Put a string on the screen.

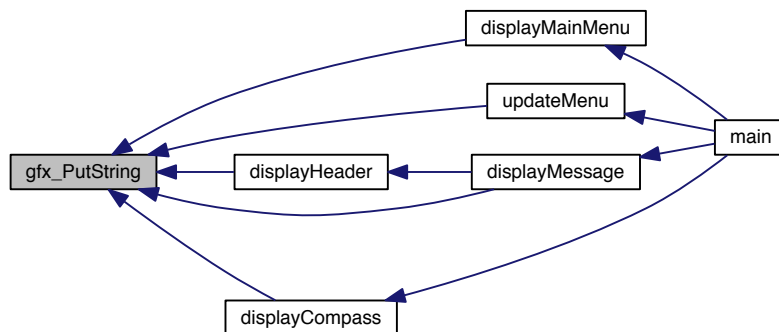
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Draw a rectangle.

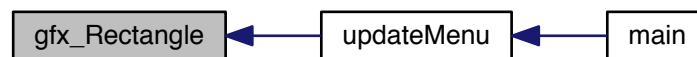
Parameters

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:



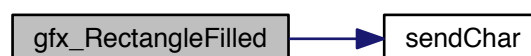
4.7.2.11 void gfx_RectangleFilled (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle filled with a color.

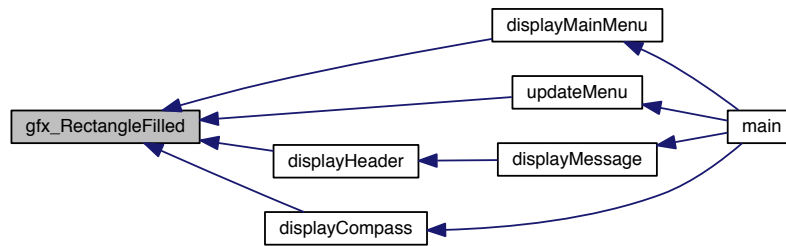
Parameters

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:



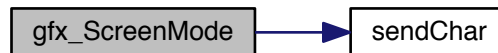
4.7.2.12 void gfx_ScreenMode (int mode)

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:



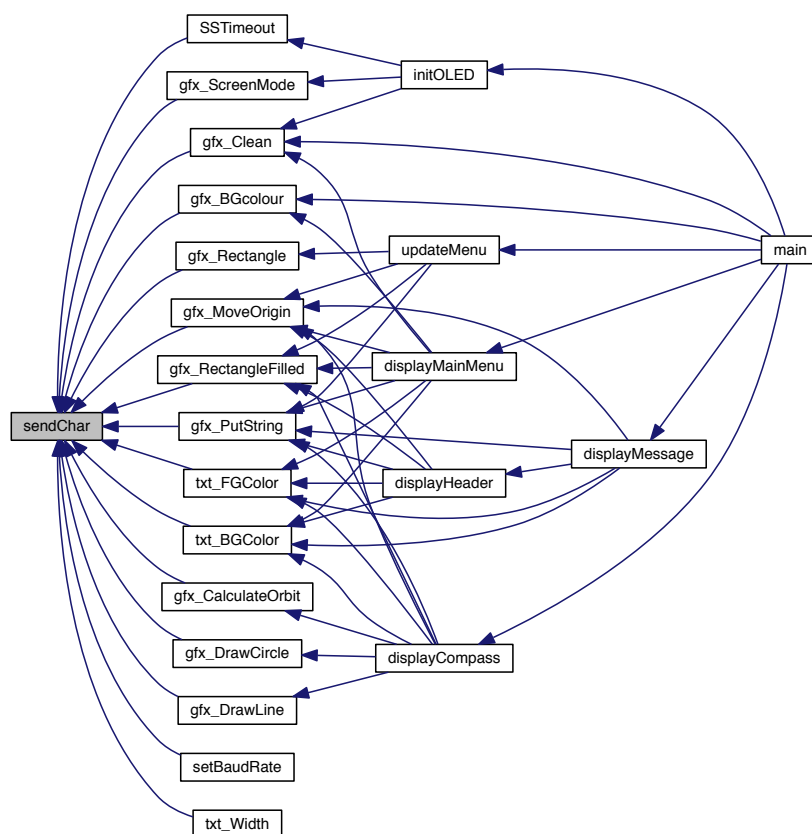
4.7.2.13 void sendChar (int c)

Send char.

Parameters

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

Here is the caller graph for this function:



4.7.2.14 void SStimeout (int t)

Screensave mode.

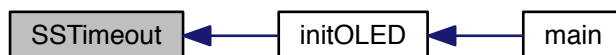
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.15 void toggleOLEDInterrupt (unsigned int *state*)

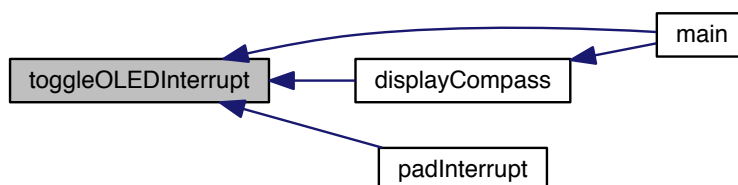
Toggle OLED interrupt.

1 = interrupt enable for OLED, 0 = disable

Parameters

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

Here is the caller graph for this function:



4.7.2.16 void txt_BGColor (int *color*)

Set the text background color.

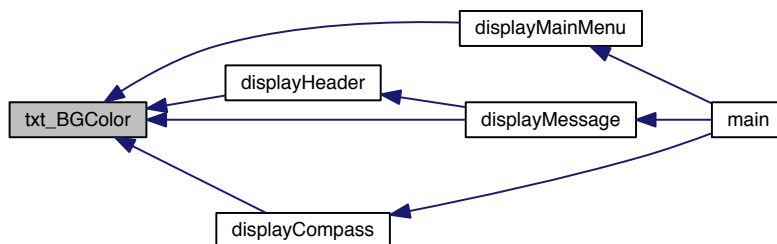
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.17 void txt_FGColor (int *color*)

Set the text color.

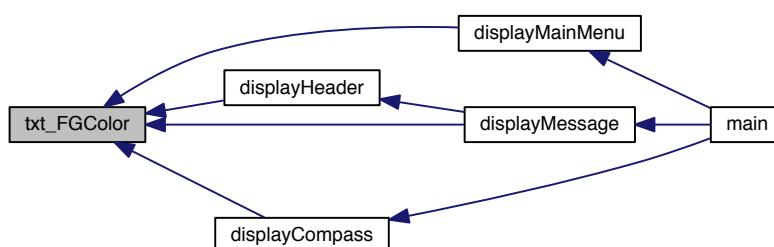
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.18 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:



4.7.3 Variable Documentation

4.7.3.1 int displayHasBeenUpdated

Is the display has been updated, aka needed

4.7.3.2 unsigned int modeDisplay = MD_SHUTDOWN

What mode is selected right now

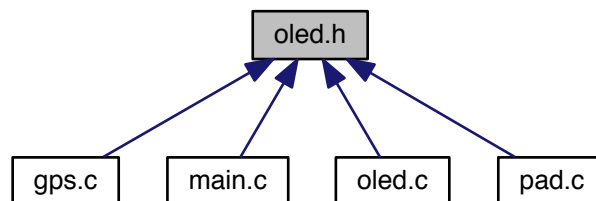
4.7.3.3 unsigned int oldModeDisplay

What mode was selected just before

4.8 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 CADETBLUE 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 0xFEAO
- 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 0xFFFE
- 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** 0xFE5
- 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`

4.8.1 Detailed Description

File containing the OLED functions.

Author

Gaël Foppolo (gaelfoppolo)

4.8.2 Function Documentation

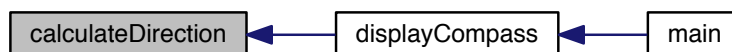
4.8.2.1 char* calculateDirection ()

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

Returns

The direction.

Here is the caller graph for this function:



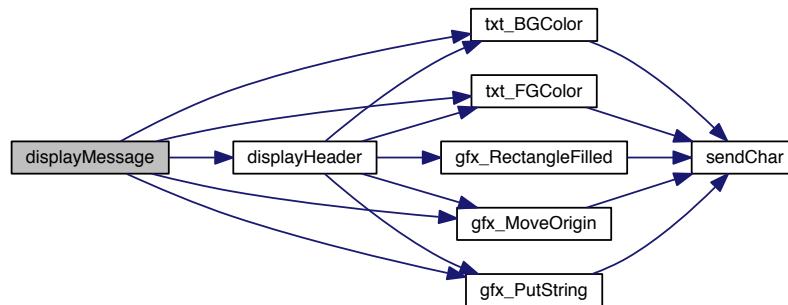
4.8.2.2 void displayMessage (char * string)

Display a string in the center of the screen.

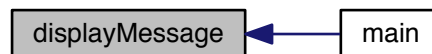
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



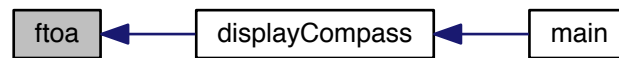
4.8.2.3 void ftoa (char * p, float x)

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:



4.8.2.4 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:



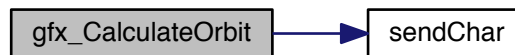
4.8.2.5 void gfx_CalculateOrbit (int *angle*, int *distance*, int * *x*, int * *y*)

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

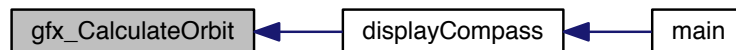
Parameters

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:



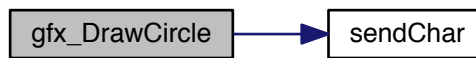
4.8.2.6 void gfx_DrawCircle (int x, int y, int radius, int color)

Draw a circle.

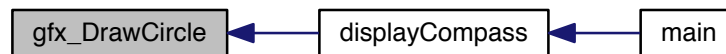
Parameters

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:



4.8.2.7 void gfx_DrawLine (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a line.

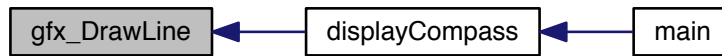
Parameters

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:



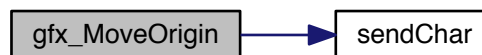
4.8.2.8 void gfx_MoveOrigin (int x, int y)

Move to origin to a position.

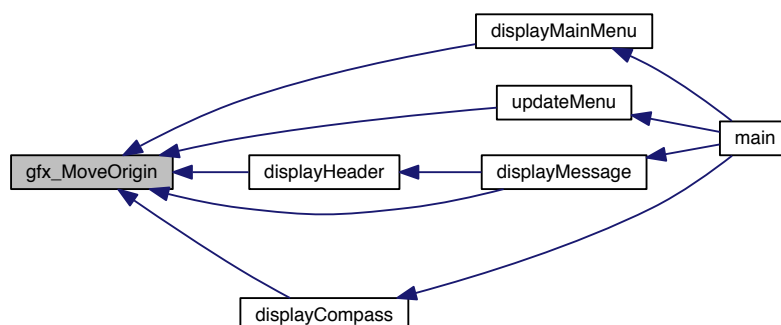
Parameters

in	x	The new x pos
in	y	The new y pos

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.9 void gfx_PutString (char * *string*)

Put a string on the screen.

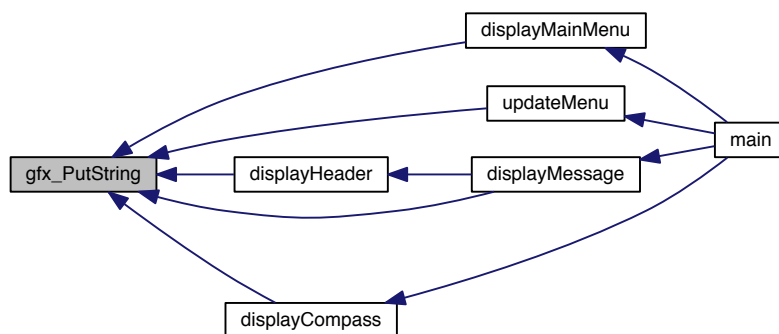
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:

4.8.2.10 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:



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

Draw a rectangle filled with a color.

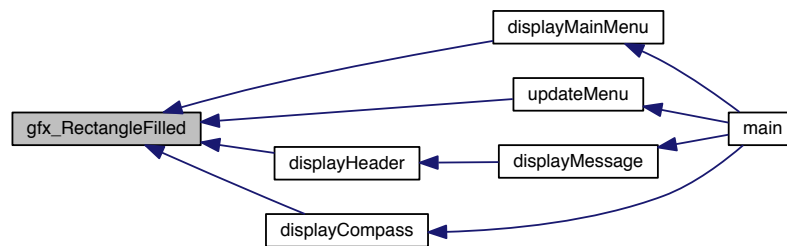
Parameters

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:



4.8.2.12 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:



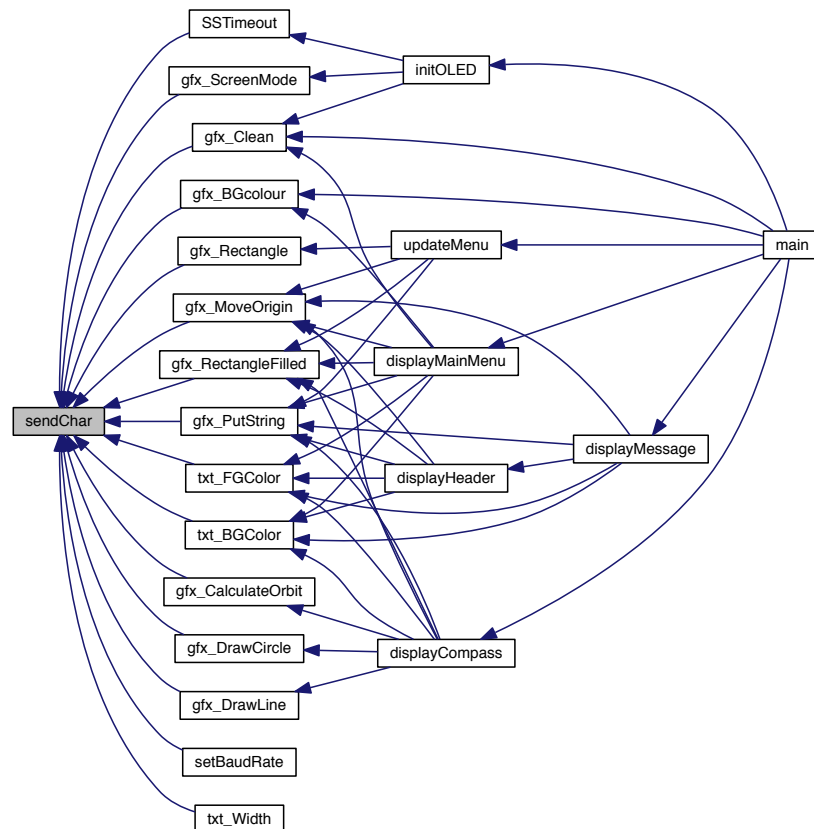
4.8.2.13 void sendChar (int c)

Send char.

Parameters

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

Here is the caller graph for this function:



4.8.2.14 void SSTimeout (int *t*)

Screensave mode.

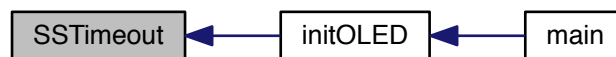
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.15 void toggleOLEDInterrupt (unsigned int *state*)

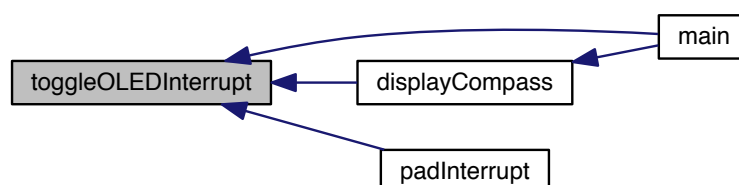
Toggle OLED interrupt.

1 = interrupt enable for OLED, 0 = disable

Parameters

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

Here is the caller graph for this function:



4.8.2.16 void txt_BGColor (int *color*)

Set the text background color.

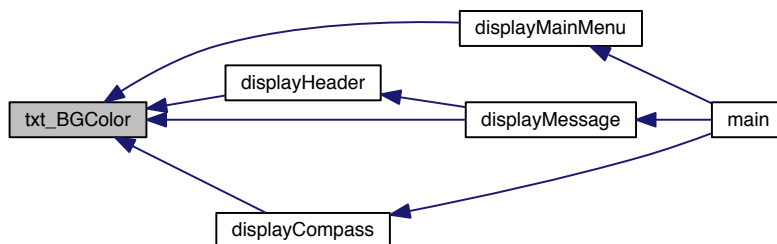
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.17 void txt_FGColor (int *color*)

Set the text color.

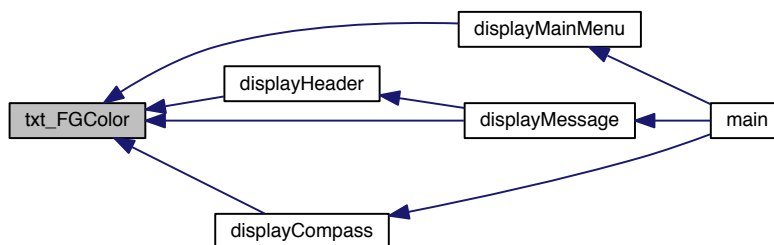
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.18 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:



4.8.3 Variable Documentation

4.8.3.1 int displayHasBeenUpdated

Is the display has been updated, aka needed

4.8.3.2 unsigned int modeDisplay

What mode is selected right now

4.8.3.3 unsigned int oldModeDisplay

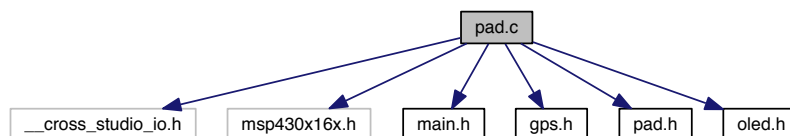
What mode was selected just before

4.9 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.

4.9.1 Detailed Description

File containing the PAD functions.

Author

Gaël Foppolo (gaelfoppolo)

4.9.2 Function Documentation

4.9.2.1 void initPAD (void)

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

All ready to use and state cleared

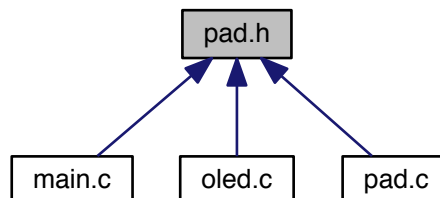
Here is the caller graph for this function:



4.10 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.

4.10.1 Detailed Description

File containing the PAD functions.

Author

Gaël Foppolo (gaelfoppolo)

4.10.2 Function Documentation

4.10.2.1 void `initPAD` (void)

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

All ready to use and state cleared

Here is the caller graph for this function:



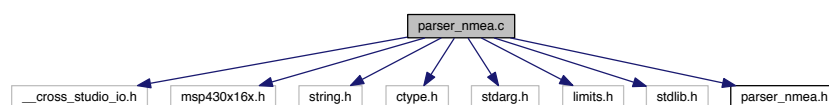
4.11 parser_nmea.c File Reference

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.

4.11.1 Detailed Description

File containing the NMEA parser functions.

Author

Gaël Foppolo (gaelfoppolo)

4.11.2 Function Documentation

4.11.2.1 int hex2int (char c)

Transform hexa to integer.

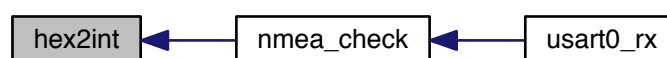
Parameters

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

Returns

An integer (hex)

Here is the caller graph for this function:



4.11.2.2 `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:



4.11.2.3 `int nmea_isfield (char c)`

Check if the char is part of the field.

Aka char isn't a comma or star

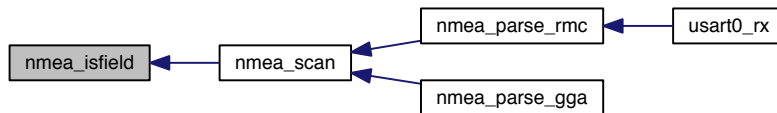
Parameters

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

Returns

1 is valid, 0 if not

Here is the caller graph for this function:



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

Parse a GGA sentence.

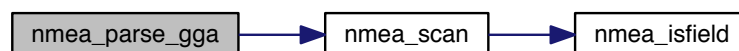
Parameters

<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:



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

Parse a RMC sentence.

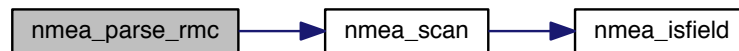
Parameters

<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:



4.11.2.6 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

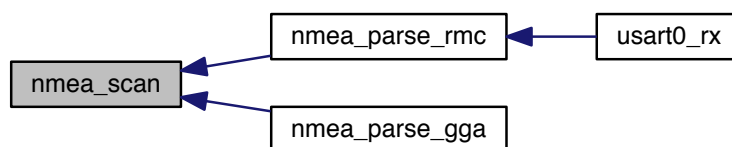
Returns

1 on success, 1 if not

Here is the call graph for this function:



Here is the caller graph for this function:

**4.11.2.7 enum nmea_sentence_id nmea_sentence_id (char * sentence)**

Determine sentence identifier.

Parameters

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

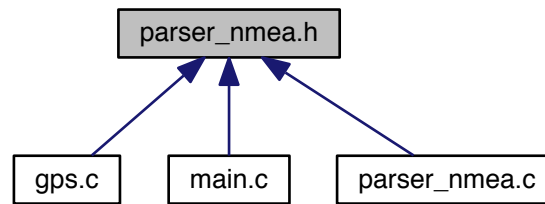
Returns

The type of sentence

4.12 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.

4.12.1 Detailed Description

File containing the NMEA parser functions.

Author

Gaël Foppolo (gaelfoppolo)

4.12.2 Typedef Documentation

4.12.2.1 typedef struct nmea_sentence_gga nmea_sentence_gga

The structure that contains the data of GGA sentences

4.12.2.2 typedef struct nmea_sentence_rmc nmea_sentence_rmc

The structure that contains the data of RMC sentences

4.12.3 Enumeration Type Documentation

4.12.3.1 enum nmea_sentence_id

The sentence identifier

4.12.4 Function Documentation

4.12.4.1 int hex2int (char c)

Transform hexa to integer.

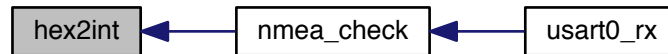
Parameters

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

Returns

An integer (hex)

Here is the caller graph for this function:



4.12.4.2 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:



4.12.4.3 int nmea_isfield (char c)

Check if the char is part of the field.

Aka char isn't a comma or star

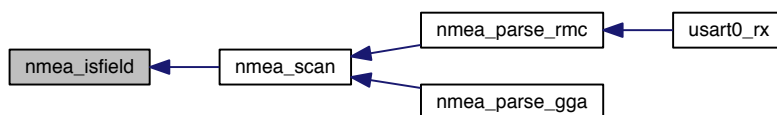
Parameters

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

Returns

1 is valid, 0 if not

Here is the caller graph for this function:



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

Parse a GGA sentence.

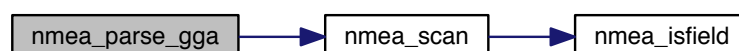
Parameters

<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:



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

Parse a RMC sentence.

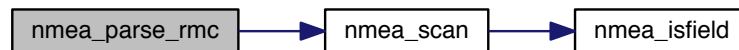
Parameters

<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:



4.12.4.6 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

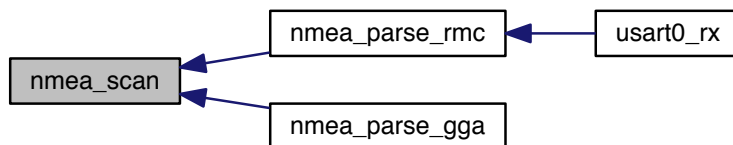
Returns

1 on success, 1 if not

Here is the call graph for this function:



Here is the caller graph for this function:



4.12.4.7 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, 8
 gps.h, 13
calculateDirection
 oled.c, 28
 oled.h, 49

dataValid
 gps.c, 11
 gps.h, 16
deg2rad
 gps.c, 8
 gps.h, 14
delay
 main.c, 20
 main.h, 24
displayHasBeenUpdated
 oled.c, 40
 oled.h, 61
displayMessage
 oled.c, 28
 oled.h, 49

ftoa
 oled.c, 29
 oled.h, 50

GPSPData
 gps.c, 11
 gps.h, 16
gfx_BGcolour
 oled.c, 29
 oled.h, 51
gfx_CalculateOrbit
 oled.c, 30
 oled.h, 51
gfx_DrawCircle
 oled.c, 31
 oled.h, 52
gfx_DrawLine
 oled.c, 32
 oled.h, 53
gfx_MoveOrigin
 oled.c, 32
 oled.h, 54
gfx_PutString
 oled.c, 33
 oled.h, 54
gfx_Rectangle
 oled.c, 34

 oled.h, 55
gfx_RectangleFilled
 oled.c, 35
 oled.h, 56
gfx_ScreenMode
 oled.c, 36
 oled.h, 57
gps.c, 7
 calcDistance, 8
 dataValid, 11
 deg2rad, 8
 GPSPData, 11
 gpsSend, 10
 toggleGPSInterrupt, 11
 toggleGPS, 10
gps.h, 12
 calcDistance, 13
 dataValid, 16
 deg2rad, 14
 GPSPData, 16
 gps_data, 13
 gpsSend, 14
 toggleGPSInterrupt, 15
 toggleGPS, 15
gps_data, 5
 gps.h, 13
gpsSend
 gps.c, 10
 gps.h, 14

hex2int
 parser_nmea.c, 65
 parser_nmea.h, 71

initLED
 led.c, 17
 led.h, 19
initPAD
 pad.c, 63
 pad.h, 64

led.c, 16
 initLED, 17
 toggleLED, 17
led.h, 18
 initLED, 19
 toggleLED, 19

main
 main.c, 21

- main.c, 19
 - delay, 20
 - main, 21
 - modeSelected, 23
 - toggleCommunication, 22
- main.h, 23
 - delay, 24
 - modeSelected, 25
 - toggleCommunication, 25
- modeDisplay
 - oled.c, 41
 - oled.h, 62
- modeSelected
 - main.c, 23
 - main.h, 25
- nmea_check
 - parser_nmea.c, 65
 - parser_nmea.h, 72
- nmea_isfield
 - parser_nmea.c, 66
 - parser_nmea.h, 72
- nmea_parse_gga
 - parser_nmea.c, 67
 - parser_nmea.h, 73
- nmea_parse_rmc
 - parser_nmea.c, 67
 - parser_nmea.h, 73
- nmea_scan
 - parser_nmea.c, 68
 - parser_nmea.h, 74
- nmea_sentence_gga, 5
 - parser_nmea.h, 71
- nmea_sentence_id
 - parser_nmea.c, 69
 - parser_nmea.h, 71, 75
- nmea_sentence_rmc, 6
 - parser_nmea.h, 71
- oldModeDisplay
 - oled.c, 41
 - oled.h, 62
- oled.c, 26
 - calculateDirection, 28
 - displayHasBeenUpdated, 40
 - displayMessage, 28
 - ftoa, 29
 - gfx_BGcolour, 29
 - gfx_CalculateOrbit, 30
 - gfx_DrawCircle, 31
 - gfx_DrawLine, 32
 - gfx_MoveOrigin, 32
 - gfx_PutString, 33
 - gfx_Rectangle, 34
 - gfx_RectangleFilled, 35
 - gfx_ScreenMode, 36
 - modeDisplay, 41
 - oldModeDisplay, 41
 - SSTimeout, 37
 - sendChar, 36
 - toggleOLEDInterrupt, 38
 - txt_BGColor, 38
 - txt_FGColor, 39
 - txt_Width, 40
- oled.h, 41
 - calculateDirection, 49
 - displayHasBeenUpdated, 61
 - displayMessage, 49
 - ftoa, 50
 - gfx_BGcolour, 51
 - gfx_CalculateOrbit, 51
 - gfx_DrawCircle, 52
 - gfx_DrawLine, 53
 - gfx_MoveOrigin, 54
 - gfx_PutString, 54
 - gfx_Rectangle, 55
 - gfx_RectangleFilled, 56
 - gfx_ScreenMode, 57
 - modeDisplay, 62
 - oldModeDisplay, 62
 - SSTimeout, 58
 - sendChar, 57
 - toggleOLEDInterrupt, 59
 - txt_BGColor, 59
 - txt_FGColor, 60
 - txt_Width, 61
- pad.c, 62
 - initPAD, 63
- pad.h, 63
 - initPAD, 64
- parser_nmea.c, 64
 - hex2int, 65
 - nmea_check, 65
 - nmea_isfield, 66
 - nmea_parse_gga, 67
 - nmea_parse_rmc, 67
 - nmea_scan, 68
 - nmea_sentence_id, 69
- parser_nmea.h, 69
 - hex2int, 71
 - nmea_check, 72
 - nmea_isfield, 72
 - nmea_parse_gga, 73
 - nmea_parse_rmc, 73
 - nmea_scan, 74
 - nmea_sentence_gga, 71
 - nmea_sentence_id, 71, 75
 - nmea_sentence_rmc, 71
- SSTimeout
 - oled.c, 37
 - oled.h, 58
- sendChar
 - oled.c, 36
 - oled.h, 57
- toggleCommunication

- main.c, [22](#)
 - main.h, [25](#)
- toggleGPSInterrupt
 - gps.c, [11](#)
 - gps.h, [15](#)
- toggleGPS
 - gps.c, [10](#)
 - gps.h, [15](#)
- toggleLED
 - led.c, [17](#)
 - led.h, [19](#)
- toggleOLEDInterrupt
 - oled.c, [38](#)
 - oled.h, [59](#)
- txt_BGColor
 - oled.c, [38](#)
 - oled.h, [59](#)
- txt_FGColor
 - oled.c, [39](#)
 - oled.h, [60](#)
- txt_Width
 - oled.c, [40](#)
 - oled.h, [61](#)