GPS

Generated by Doxygen 1.8.11

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

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

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

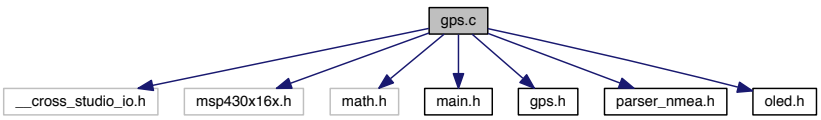
File Documentation

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 GPSData

  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

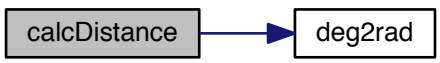
Parameters

lat1	The latitude of the first point
lon1	The longitude of the first point
lat2	The latitude of the second point
lon2	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

deg	The angle in degrees

Returns

The angle in radians

Here is the caller graph for this function:

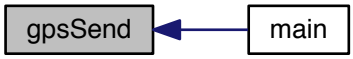
4.1.2.3 void gpsSend (char * *message*)

Send sentences to GPS to configure it (interrupt mode)

Parameters

message	Message to send

Here is the caller graph for this function:



4.1.2.4 void toggleGPS (unsigned int *state*)

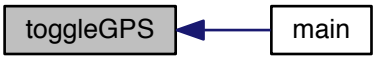
Toggle GPS (P4.0, ENABLE_GPS)

1 = enable, 0 = disable

Parameters

state	The new state

Here is the caller graph for this function:



4.1.2.5 void toggleGPSInterrupt (unsigned int *state*)

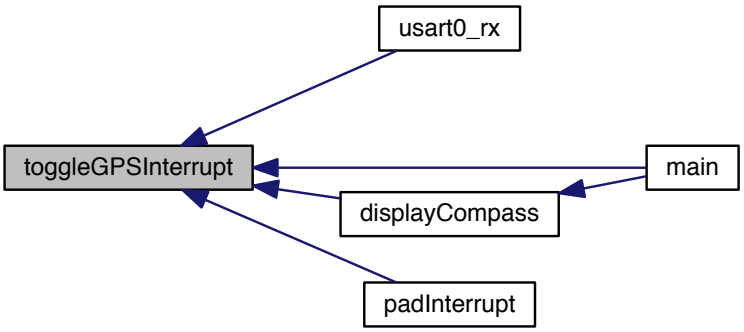
Toggle GPS interrupt.

1 = interrupt enable for GPS, 0 = disable

Parameters

state	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 GPSData

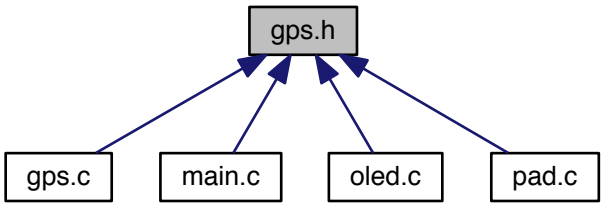
Store GPS valid data.

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 "$PMTK314,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0∗28\r\n"

  Tell the GPS to not send sentences.
- #define TURN_ALL "$PMTK314,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0∗28\r\n"

  Tell the GPS to send all sentences.
- #define GGA_RMC "$PMTK314,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0∗28\r\n"

  Tell the GPS to send GGA and RMC sentences only.
- #define RMC_ONLY "$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0∗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 GPSData

  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

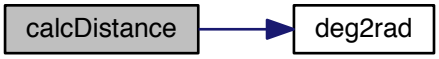
Parameters

lat1	The latitude of the first point
lon1	The longitude of the first point
lat2	The latitude of the second point
lon2	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.

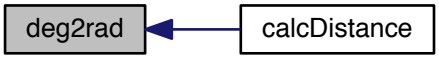
Parameters

deg	The angle in degrees

Returns

The angle in radians

Here is the caller graph for this function:



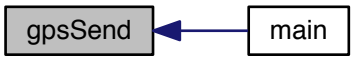
4.2.3.3 void gpsSend (char * *message*)

Send sentences to GPS to configure it (interrupt mode)

Parameters

message	Message to send

Here is the caller graph for this function:



4.2.3.4 void toggleGPS (unsigned int *state*)

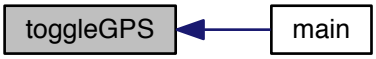
Toggle GPS (P4.0, ENABLE_GPS)

1 = enable, 0 = disable

Parameters

state	The new state

Here is the caller graph for this function:



4.2.3.5 void toggleGPSInterrupt (unsigned int *state*)

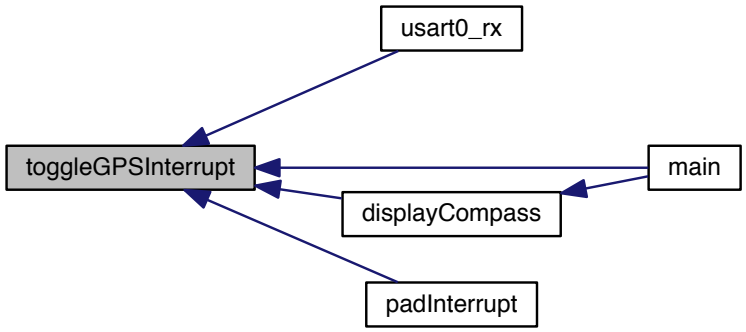
Toggle GPS interrupt.

1 = interrupt enable for GPS, 0 = disable

Parameters

state	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 GPSData

Useful data received and valid.

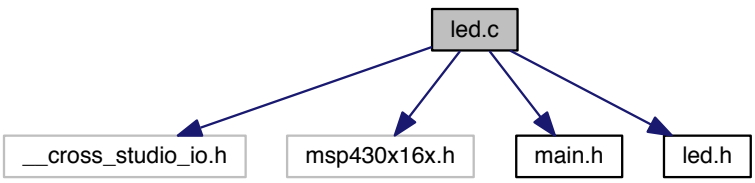
Useful data received and valid.

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*)

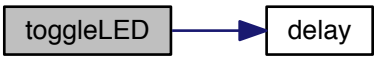
Toogle the state of the choosen LED for a choosen time.

duration = 0 -> stay in the state choosen

Parameters

n	The LED to toogle
state	The new state
duration	The time to toogle the state of the LED

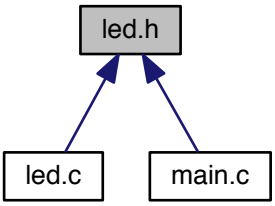
Here is the call graph for this function:



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*)

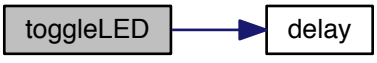
Toogle the state of the choosen LED for a choosen time.

duration = 0 -> stay in the state choosen

Parameters

n	The LED to toogle
state	The new state
duration	The time to toogle the state of the LED

Here is the call graph for this function:

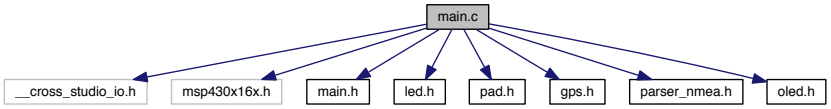


4.5 main.c File Reference

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

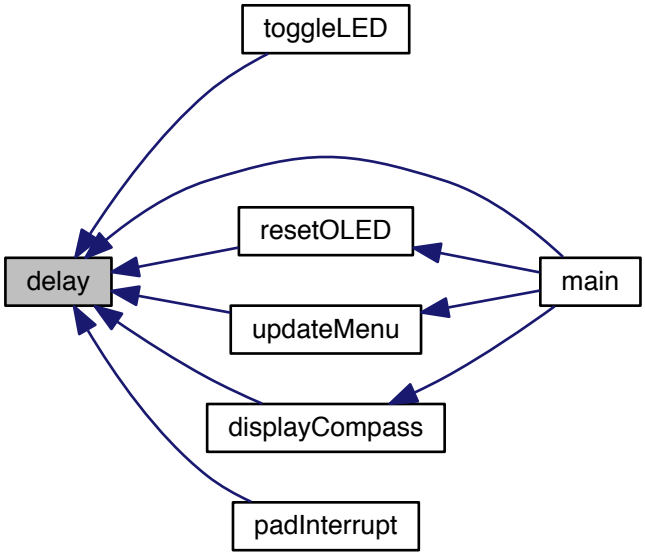
4.5.2.1 void delay (float *x*)

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

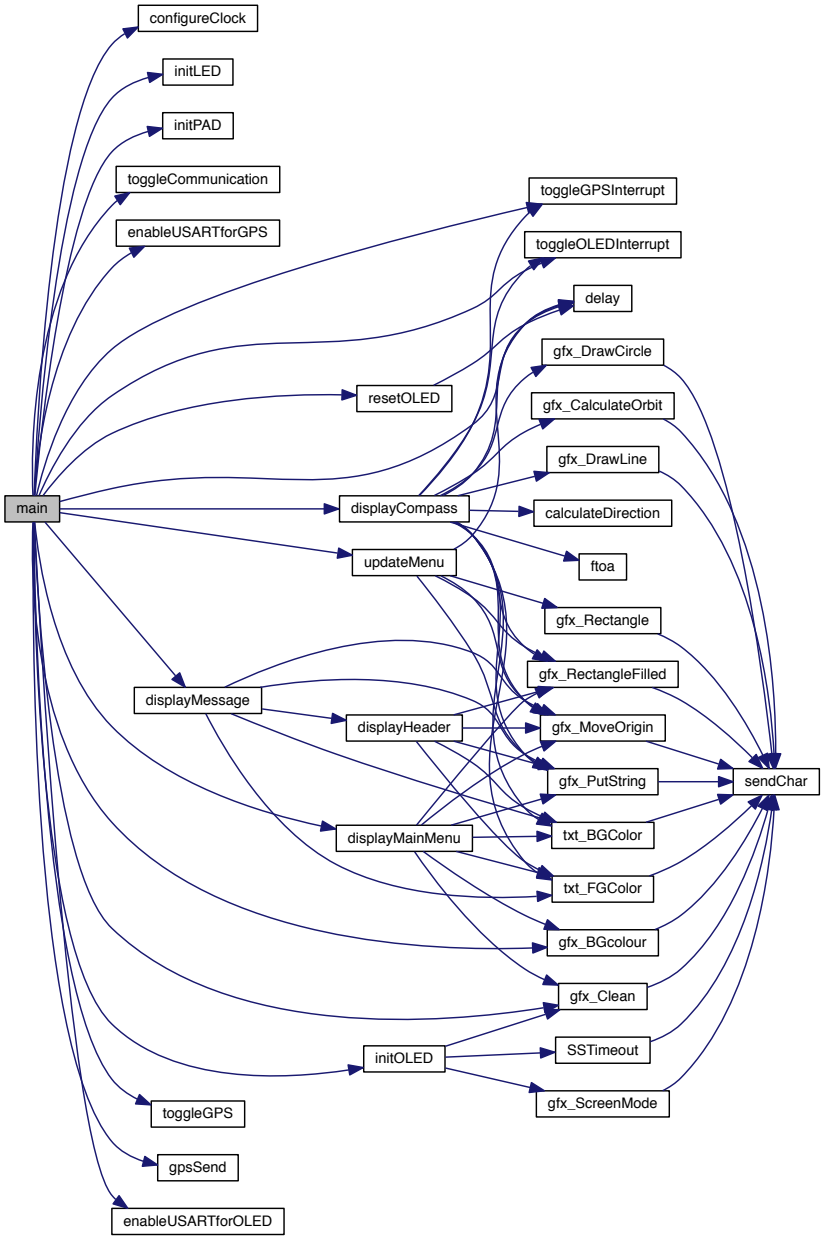
Compass entry point

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*)

Toogle the communication (P4.2, CMD_SWITCH)

1 = USB, 0 = GPS

Parameters

state	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,

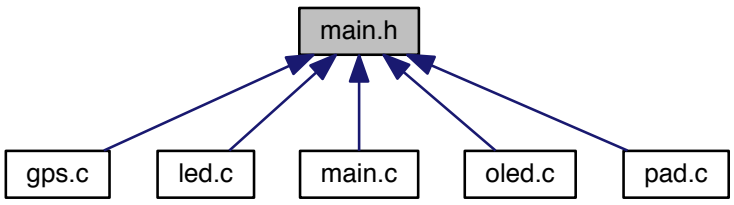
See also

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

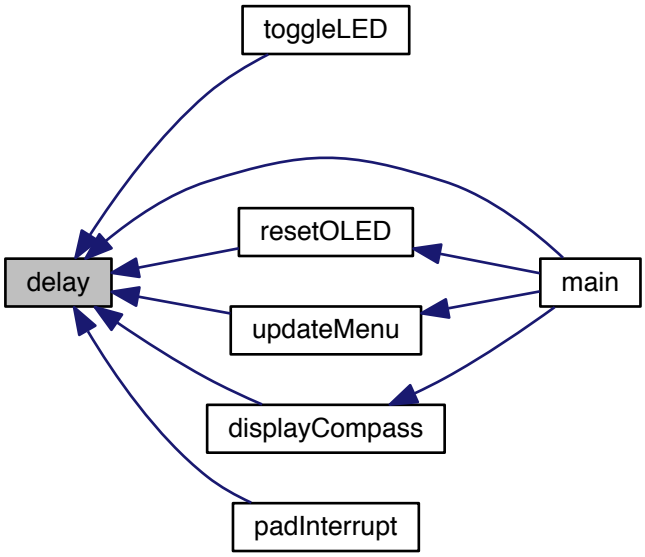
4.6.2.1 void delay (float *x*)

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

state	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

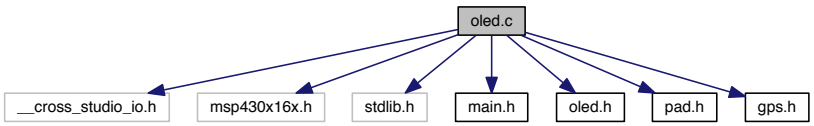
M_MENU, etc.

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

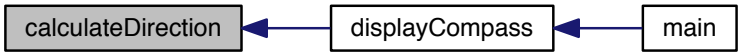
4.7.2.1 char* calculateDirection ( )

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

Returns

The direction.

Here is the caller graph for this function:



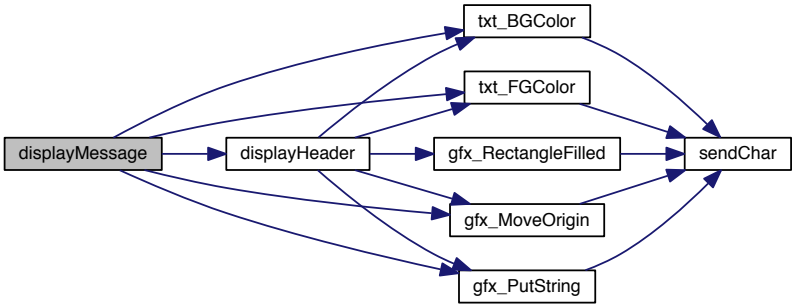
4.7.2.2 void displayMessage (char * *string*)

Display a string in the center of the screen.

Parameters

string	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

	p	The buffer (string)
in	*x*	The float

Here is the caller graph for this function:

4.7.2.4 void gfx_BGcolour (int *color*)

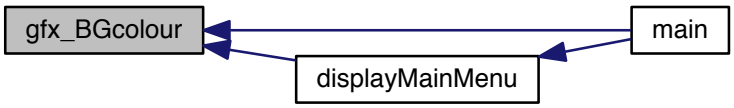
Set the background color.

Parameters

in	*color*	The color

Here is the call graph for this function:

Here is the caller graph for this function:



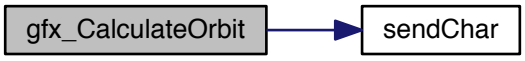
4.7.2.5 void gfx_CalculateOrbit (int *angle*, int *distance*, int * *x*, int * *y*)

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

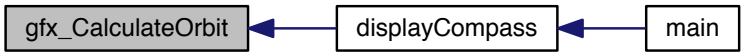
Parameters

in	*angle*	The angle
in	*distance*	The distance
	x	The x pos computed
	y	The y pos computed

Here is the call graph for this function:



Here is the caller graph for this function:



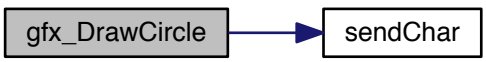
4.7.2.6 void gfx_DrawCircle (int *x*, int *y*, int *radius*, int *color*)

Draw a circle.

Parameters

in	*x*	x pos of center of the circle
in	*y*	y pos of center of the circle
in	*radius*	The radius
in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:

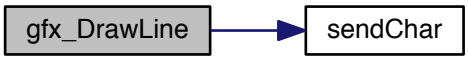
4.7.2.7 void gfx_DrawLine (int *x1,* int *y1,* int *x2,* int *y2,* int *color*)

Draw a line.

Parameters

in	*x1*	The x pos of the beginning of the line
in	*y1*	The y pos of the beginning of the line
in	*x2*	The x pos of the ending of the line
in	*y2*	The y pos of the ending of the line
in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:

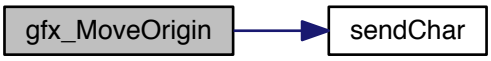
4.7.2.8 void gfx_MoveOrigin (int *x,* int *y*)

Move to origin to a position.

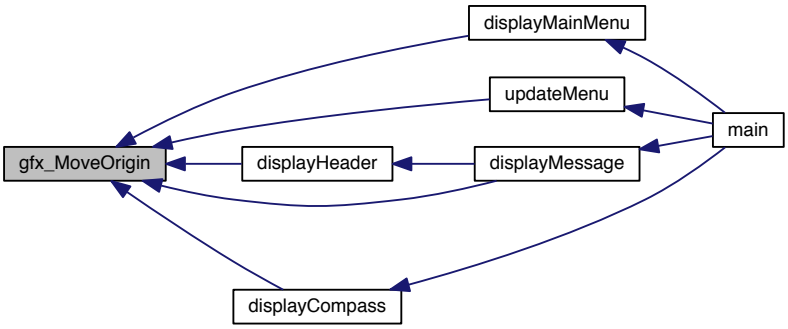
Parameters

in	*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:



4.7.2.9 void gfx_PutString (char * *string*)

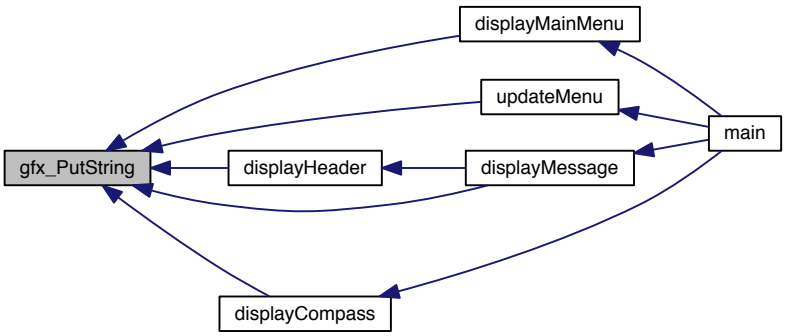
Put a string on the screen.

Parameters

string	The string

Here is the call graph for this function:

Here is the caller graph for this function:



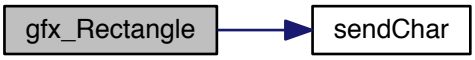
4.7.2.10 void gfx_Rectangle (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle.

Parameters

in	*x1*	The x pos of the top left corner
in	*y1*	The y pos of the top left corner
in	*x2*	The x pos of the bottom right corner
in	*y2*	The y pos of the bottom right corner
in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:

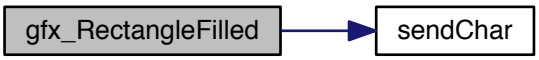
4.7.2.11 void gfx_RectangleFilled (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle filled with a color.

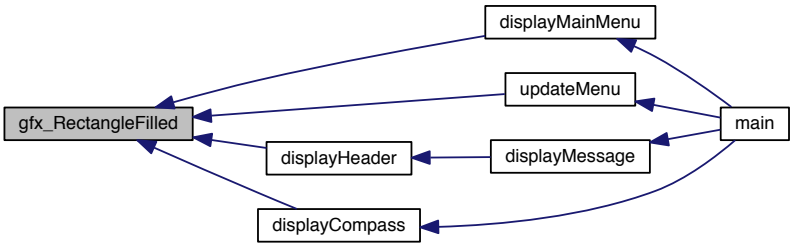
Parameters

in	*x1*	The x pos of the top left corner
in	*y1*	The y pos of the top left corner
in	*x2*	The x pos of the bottom right corner
in	*y2*	The y pos of the bottom right corner
in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:



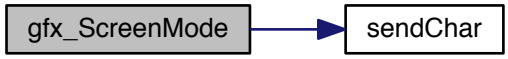
4.7.2.12 void gfx_ScreenMode (int *mode*)

Screen mode (portrait/landscape)

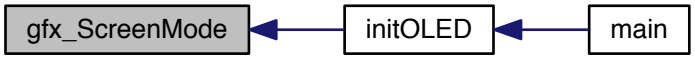
Parameters

in	*mode*	The mode

Here is the call graph for this function:



Here is the caller graph for this function:



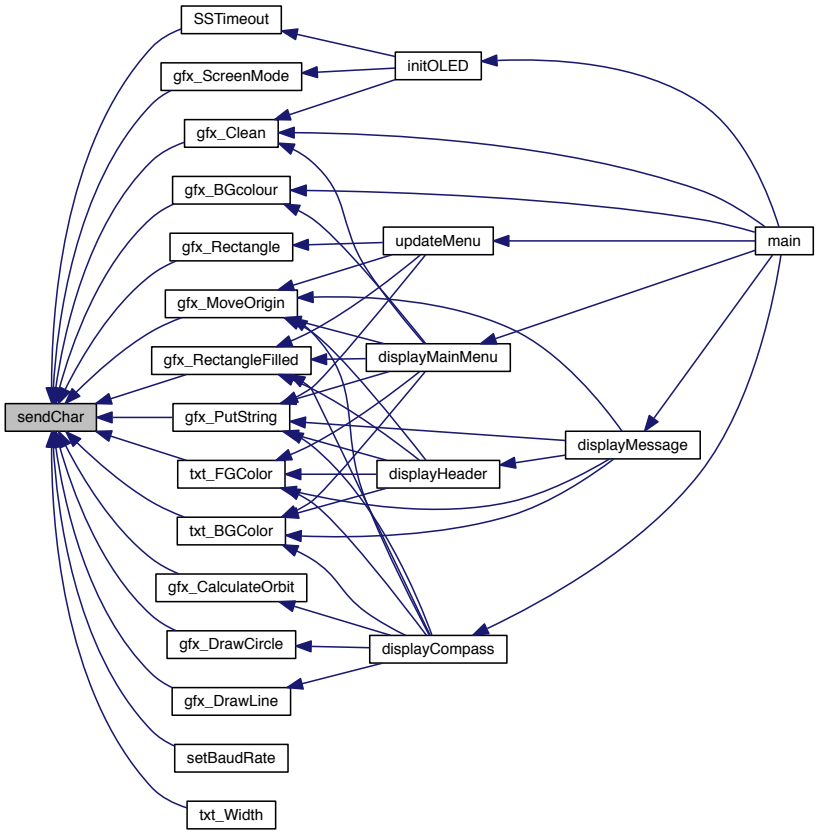
4.7.2.13 void sendChar (int *c*)

Send char.

Parameters

c	The int to send

Here is the caller graph for this function:



4.7.2.14 void SSTimeout (int *t*)

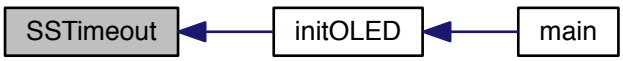
Screensave mode.

Parameters

in	*t*	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*)

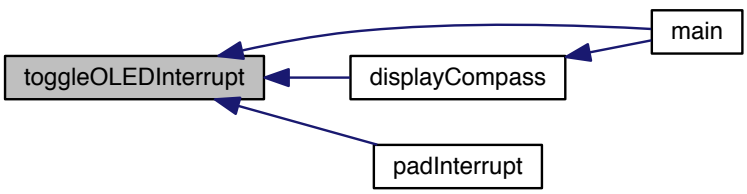
Toggle OLED interrupt.

1 = interrupt enable for OLED, 0 = disable

Parameters

state	The new state

Here is the caller graph for this function:



4.7.2.16 void txt_BGColor (int *color*)

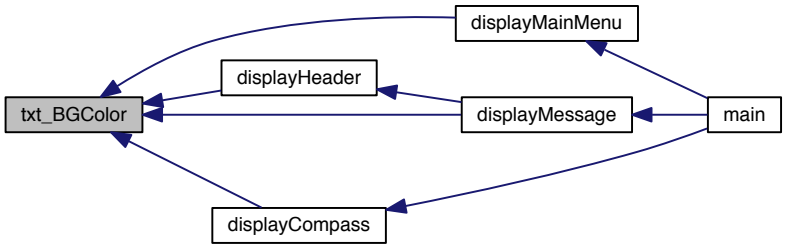
Set the text background color.

Parameters

in	*color*	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*)

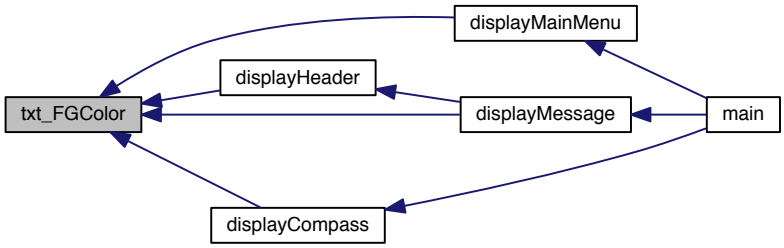
Set the text color.

Parameters

in	*color*	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	*multi*	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

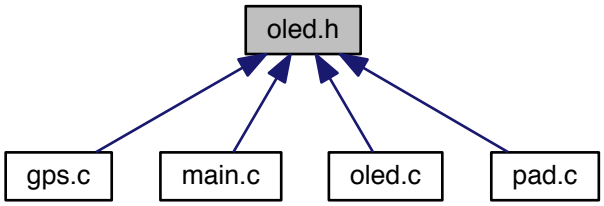
4.7.3.3 unsigned int oldModeDisplay

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 naviguation.
- #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 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 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 0xFEF5

  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:

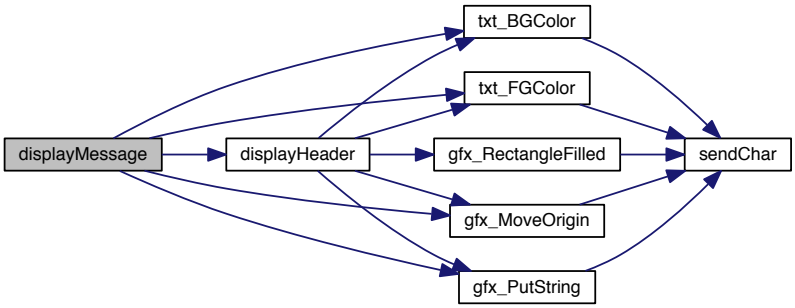
4.8.2.2 void displayMessage (char * *string*)

Display a string in the center of the screen.

Parameters

string	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

	p	The buffer (string)
in	*x*	The float

Here is the caller graph for this function:

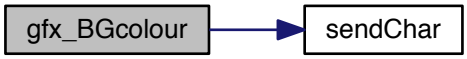
4.8.2.4 void gfx_BGcolour (int *color*)

Set the background color.

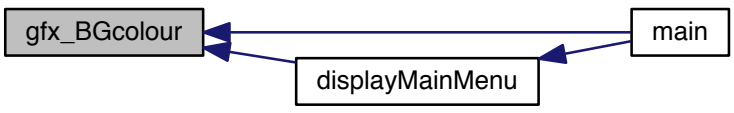
Parameters

in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:



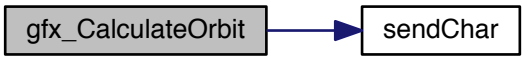
4.8.2.5 void gfx_CalculateOrbit (int *angle*, int *distance*, int * *x*, int * *y*)

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

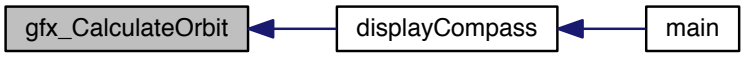
Parameters

in	*angle*	The angle
in	*distance*	The distance
	x	The x pos computed
	y	The y pos computed

Here is the call graph for this function:



Here is the caller graph for this function:



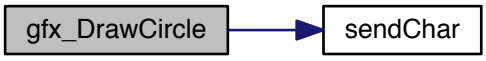
4.8.2.6 void gfx_DrawCircle (int *x*, int *y*, int *radius*, int *color*)

Draw a circle.

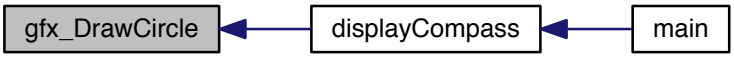
Parameters

in	*x*	x pos of center of the circle
in	*y*	y pos of center of the circle
in	*radius*	The radius
in	*color*	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	*x1*	The x pos of the beginning of the line
in	*y1*	The y pos of the beginning of the line
in	*x2*	The x pos of the ending of the line
in	*y2*	The y pos of the ending of the line
in	*color*	The color

Here is the call graph for this function:

Here is the caller graph for this function:

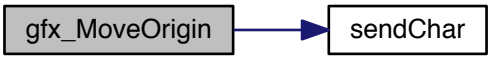
4.8.2.8 void gfx_MoveOrigin (int *x*, int *y*)

Move to origin to a position.

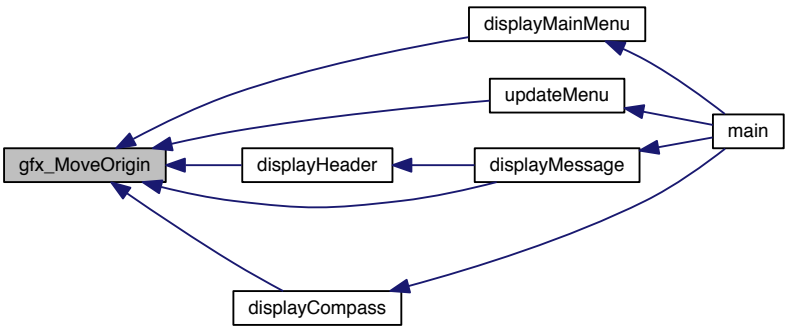
Parameters

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*)

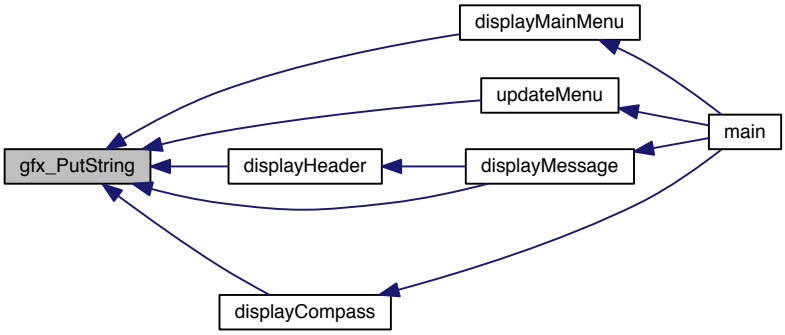
Put a string on the screen.

Parameters

string	The string

Here is the call graph for this function:

Here is the caller graph for this function:



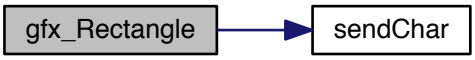
4.8.2.10 void gfx_Rectangle (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle.

Parameters

in	*x1*	The x pos of the top left corner
in	*y1*	The y pos of the top left corner
in	*x2*	The x pos of the bottom right corner
in	*y2*	The y pos of the bottom right corner
in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:

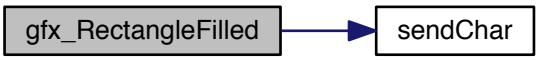
4.8.2.11 void gfx_RectangleFilled (int *x1*, int *y1*, int *x2*, int *y2*, int *color*)

Draw a rectangle filled with a color.

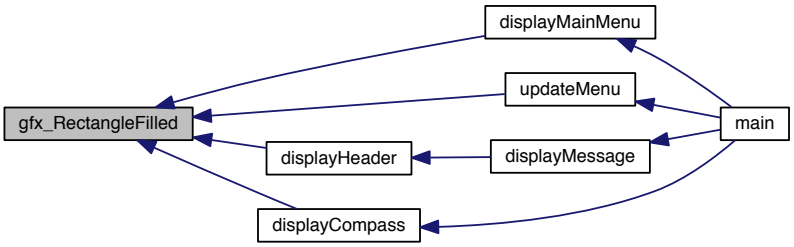
Parameters

in	*x1*	The x pos of the top left corner
in	*y1*	The y pos of the top left corner
in	*x2*	The x pos of the bottom right corner
in	*y2*	The y pos of the bottom right corner
in	*color*	The color

Here is the call graph for this function:



Here is the caller graph for this function:



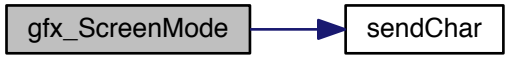
4.8.2.12 void gfx_ScreenMode (int *mode*)

Screen mode (portrait/landscape)

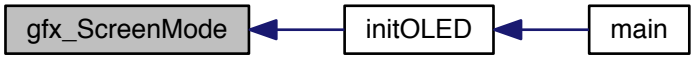
Parameters

in	*mode*	The mode

Here is the call graph for this function:



Here is the caller graph for this function:



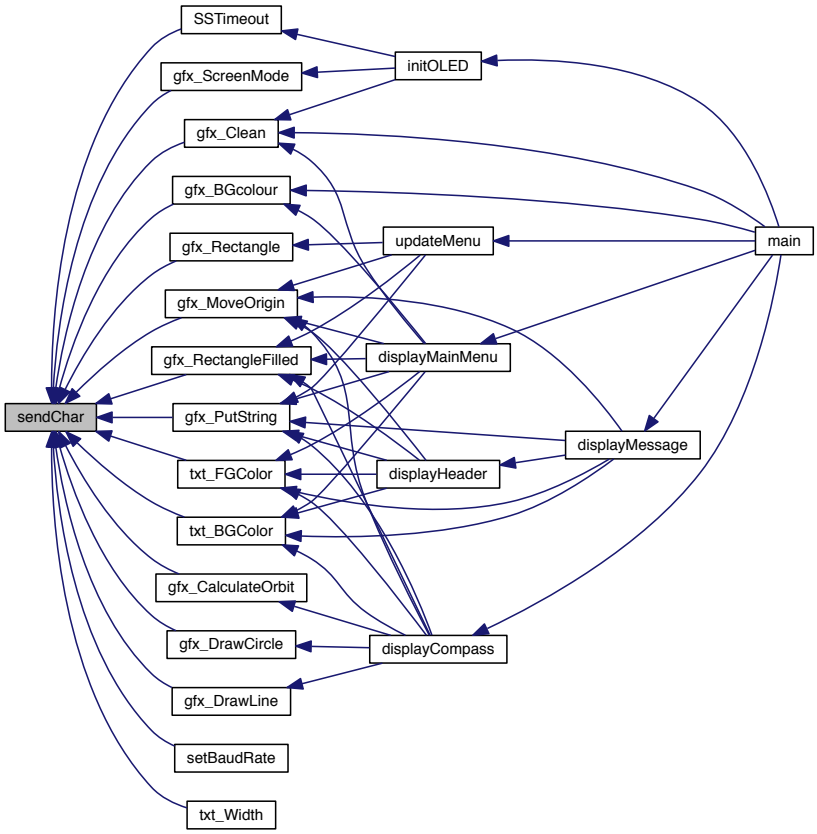
4.8.2.13 void sendChar (int *c*)

Send char.

Parameters

c	The int to send

Here is the caller graph for this function:



4.8.2.14 void SSTimeout (int *t*)

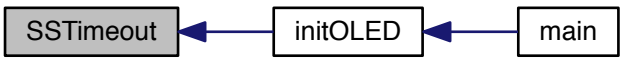
Screensave mode.

Parameters

in	*t*	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*)

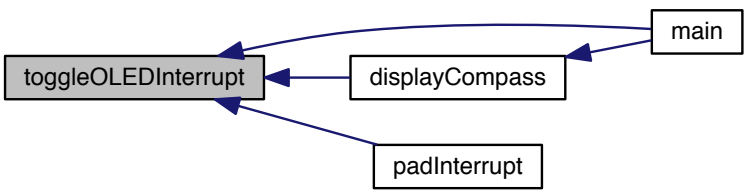
Toggle OLED interrupt.

1 = interrupt enable for OLED, 0 = disable

Parameters

state	The new state

Here is the caller graph for this function:



4.8.2.16 void txt_BGColor (int *color*)

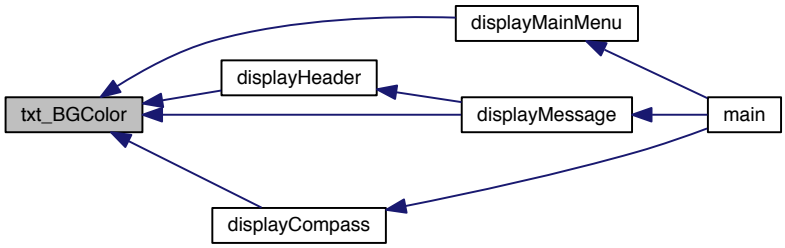
Set the text background color.

Parameters

in	*color*	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*)

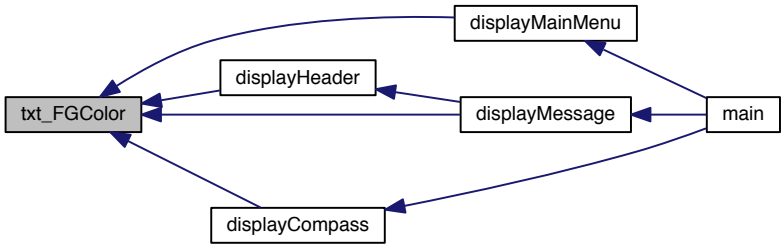
Set the text color.

Parameters

in	*color*	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	*multi*	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

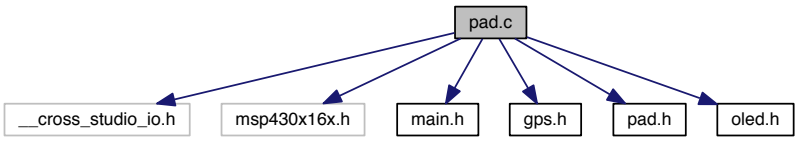
What mode was selected just before

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)

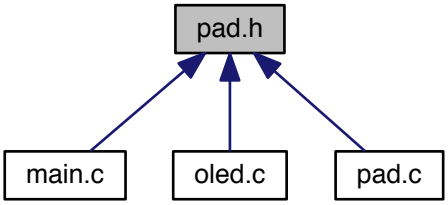
All ready to use and state cleared

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

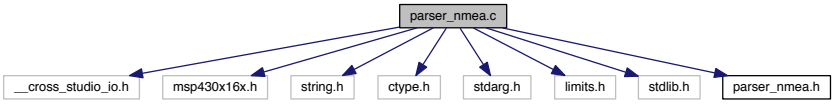
Here is the caller graph for this function:

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.

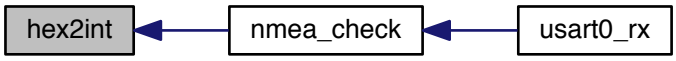
Parameters

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

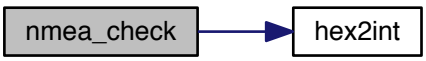
Parameters

sentence	The sentence to test
strict	Accept or not sentence with checksum

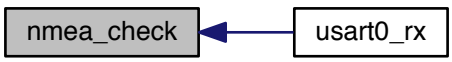
Returns

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

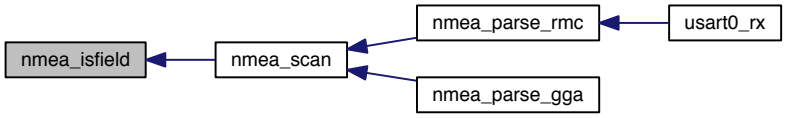
Parameters

c	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.

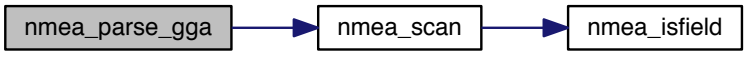
Parameters

frame	The struct where to put the parsed data
sentence	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.

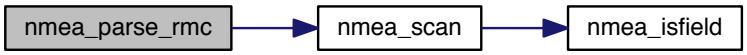
Parameters

frame	The struct where to put the parsed data
sentence	The sentence to parse

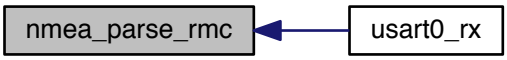
Returns

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

sentence	The sentence to parse
format	The format of the sentence

See also

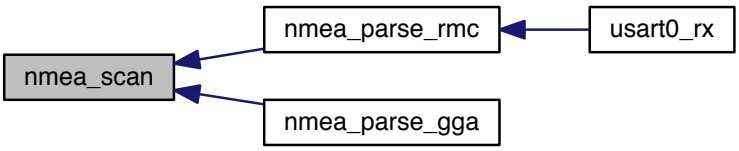
nmea_parse_*** functions for further explanations

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

sentence	Then sentence to test

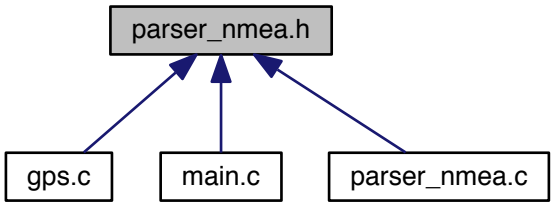
Returns

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_G↩SV**,
  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.

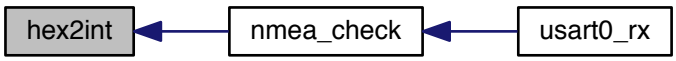
Parameters

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

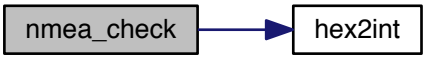
Parameters

sentence	The sentence to test
strict	Accept or not sentence with checksum

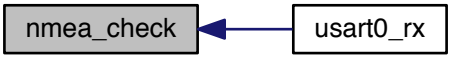
Returns

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

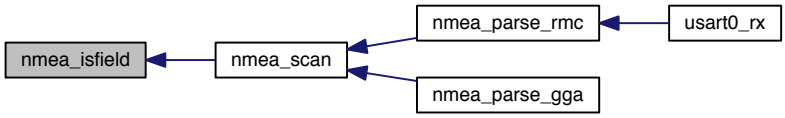
Parameters

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.

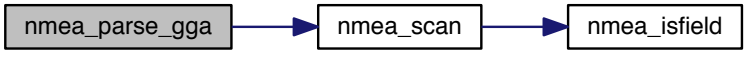
Parameters

frame	The struct where to put the parsed data
sentence	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.

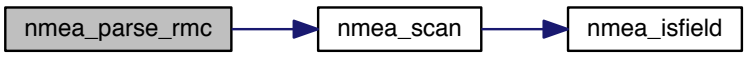
Parameters

frame	The struct where to put the parsed data
sentence	The sentence to parse

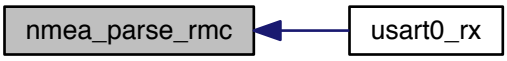
Returns

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

sentence	The sentence to parse
format	The format of the sentence

See also

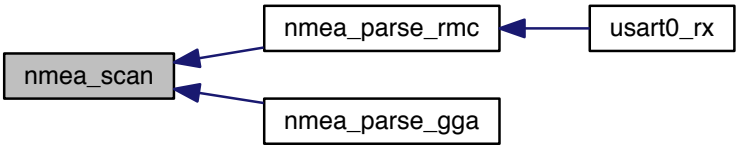
nmea_parse_∗∗∗ functions for further explanations

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

sentence	Then sentence to test

Returns

The type of sentence

Index