Code documentation

1

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 car Struct Reference

```
#include <simulator.h>
```

Public Attributes

- int doors
- int seatbelts
- int lights
- int r
- float tempOut
- float tempIn
- float tempEngine

3.1.1 Detailed Description

Analog ports ∗ A0 - temp Out ∗ A1 - temp In ∗ A2 - temp Engine ∗

A global car structure ∗

Definition at line 34 of file simulator.h.

3.1.2 Member Data Documentation

3.1.2.1 int car::doors

status of doors in car. 1 - open, 0 closed

Definition at line 35 of file simulator.h.

3.1.2.2 int car::lights

status of lights. 1 -turn on, 0 - turn off

Definition at line 37 of file simulator.h.

3.1.2.3 int car::r

statu of reverse gear

Definition at line 38 of file simulator.h.

3.1.2.4 int car::seatbelts

status of seatbelts in car. 1 - open, 0 - closed

Definition at line 36 of file simulator.h.

3.1.2.5 float car::tempEngine

temperature engine

Definition at line 41 of file simulator.h.

3.1.2.6 float car::tempIn

temperature inside

Definition at line 40 of file simulator.h.

3.1.2.7 float car::tempOut

temperature outside

Definition at line 39 of file simulator.h.

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

- simulator.h

Chapter 4

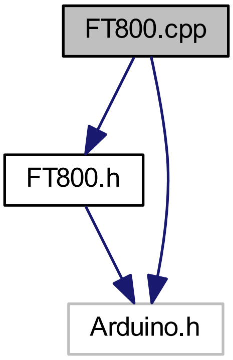
File Documentation

4.1 FT800.cpp File Reference

File containing declarations of all functions required to use with VM800.

```
#include "FT800.h"
#import <Arduino.h>
```

Include dependency graph for FT800.cpp:



Functions

- void delay_us (int us)
- void delay_ms (int ms)
- void sendData (int data)
- unsigned char getData ()
- void ft800memWrite8 (unsigned long ftAddress, unsigned char ftData8)
- void ft800memWrite16 (unsigned long ftAddress, unsigned int ftData16)
- void ft800memWrite32 (unsigned long ftAddress, unsigned long ftData32)
- unsigned char ft800memRead8 (unsigned long ftAddress)
- unsigned char ft800memRead16 (unsigned long ftAddress)
- unsigned long ft800memRead32 (unsigned long ftAddress)
- unsigned int incCMDOffset (unsigned int currentOffset, unsigned char commandSize)
- void ft800cmdWrite (unsigned char ftCommand)

4.1.1 Detailed Description

File containing declarations of all functions required to use with VM800.

Author

Daniel Sienkiewicz

Date

28 February 2016

4.1.2 Function Documentation

4.1.2.1 void delay_ms (int *ms*)

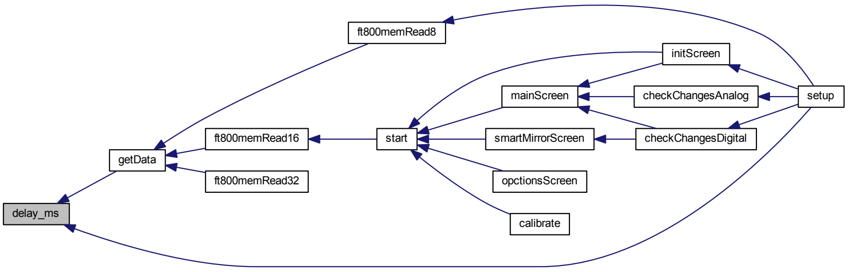
Pauses the program for the amount of time (in milisecond) specified as parameter ∗

Parameters

ms	milisecond to delay ∗

Definition at line 15 of file FT800.cpp.

Here is the caller graph for this function:



4.1.2.2 void delay_us (int *us*)

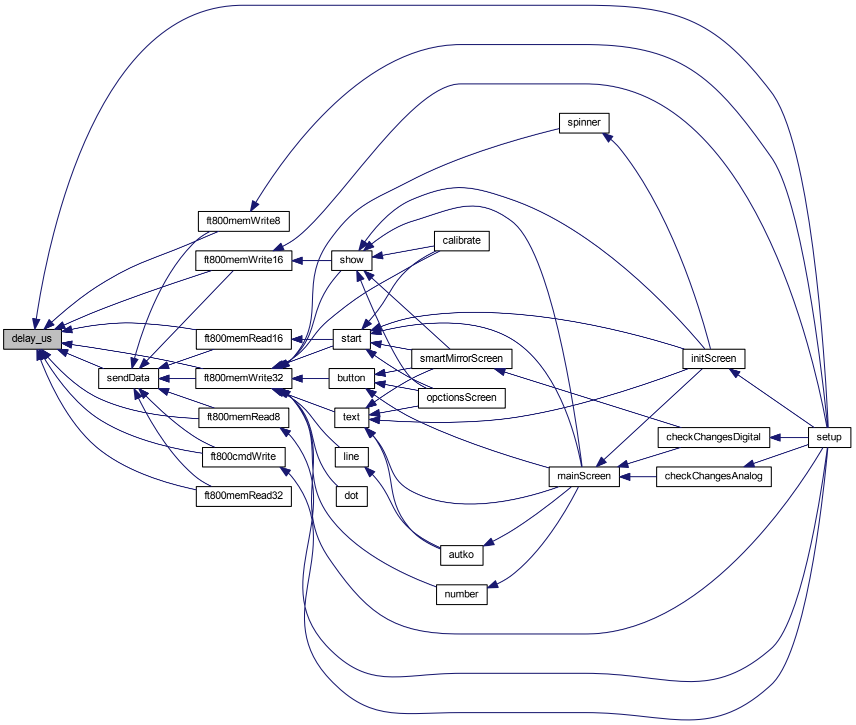
Pauses the program for the amount of time (in microsecond) specified as parameter ∗

Parameters

us	microseconds to delay ∗

Definition at line 11 of file FT800.cpp.

Here is the caller graph for this function:



4.1.2.3 void ft800cmdWrite (unsigned char *ftCommand*)

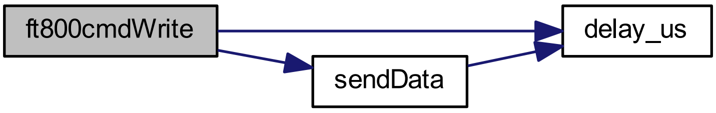
Sends FT800 command ∗

Parameters

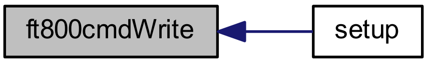
ftCommand	command to send to device ∗

Definition at line 304 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.4 unsigned char ft800memRead16 (unsigned long *ftAddress*)

Funtion to read 16 bit value from active device with using SPI interface ∗

Parameters

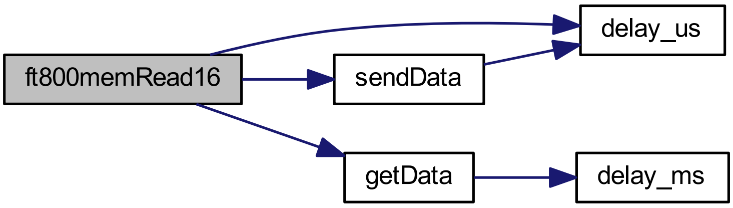
ftAddress	FT800 memory space address (24 bits) ∗

Returns

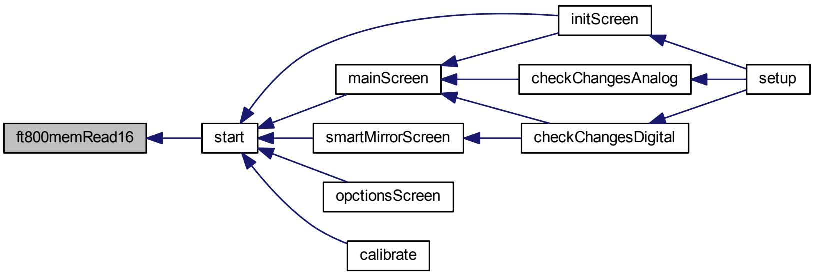
16 bit data obtained from device ∗

Definition at line 177 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.5 unsigned long ft800memRead32 (unsigned long *ftAddress*)

Funtion to read 32 bit value from active device with using SPI interface ∗

Parameters

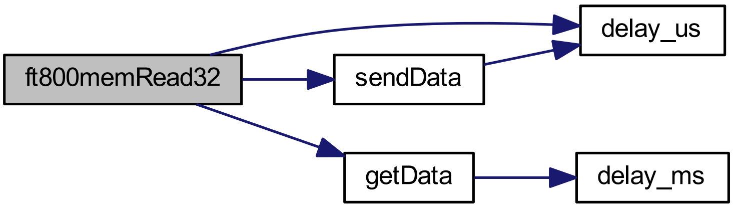
ftAddress	FT800 memory space address (24 bits) ∗

Returns

32 bit data obtained from device ∗

Definition at line 226 of file FT800.cpp.

Here is the call graph for this function:



4.1.2.6 unsigned char ft800memRead8 (unsigned long *ftAddress*)

Funtion to read 8 bit value from active device with using SPI interface ∗

Parameters

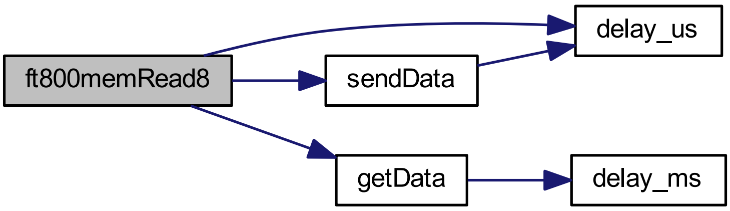
ftAddress	FT800 memory space address (24 bits) ∗

Returns

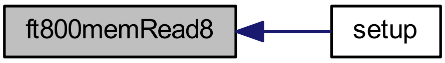
8 bit data obtained from device ∗

Definition at line 143 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.7 void ft800memWrite16 (unsigned long *ftAddress,* unsigned int *ftData16*)

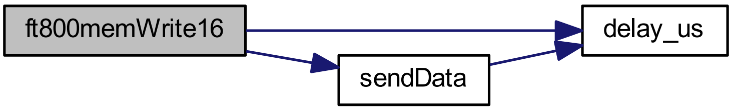
Funtion to send 16 bit value to active device with using SPI interface ∗

Parameters

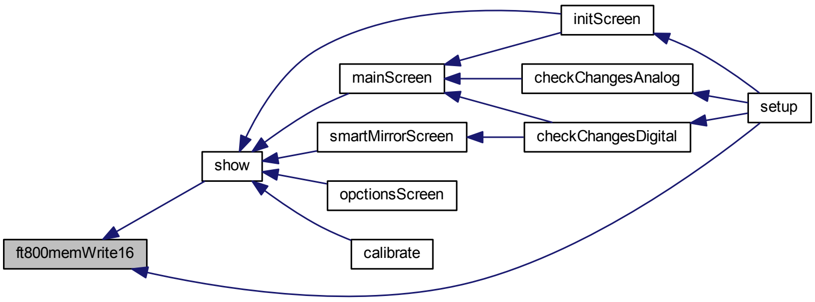
ftAddress	FT800 memory space address (24 bits) ∗
ftData8	a byte to send ∗

Definition at line 73 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.8 void ft800memWrite32 (unsigned long *ftAddress,* unsigned long *ftData32*)

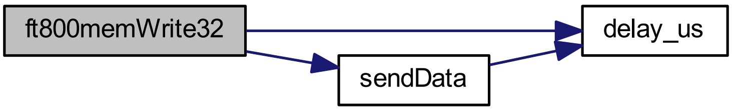
Funtion to send 32 bit value to active device with using SPI interface ∗

Parameters

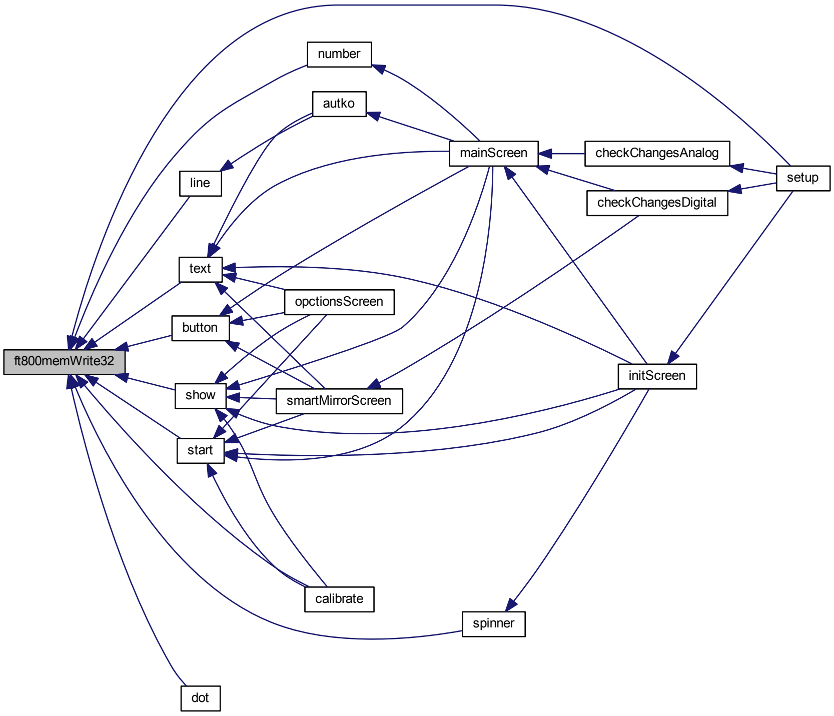
ftAddress	FT800 memory space address (24 bits) ∗
ftData8	a byte to send ∗

Definition at line 105 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.9 void ft800memWrite8 (unsigned long *ftAddress*, unsigned char *ftData8*)

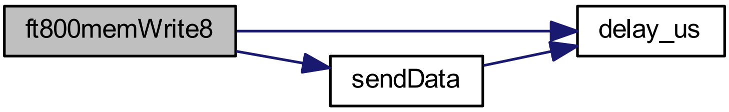
Funtion to send 8 bit value to active device with using SPI interface ∗

Parameters

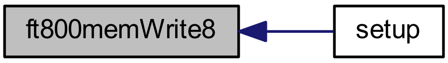
ftAddress	FT800 memory space address (24 bits) ∗
ftData8	a byte to send ∗

Definition at line 45 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.10 unsigned char getData ( )

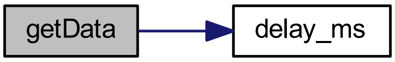
Function getting data from active device with using SPI interface ∗

Returns

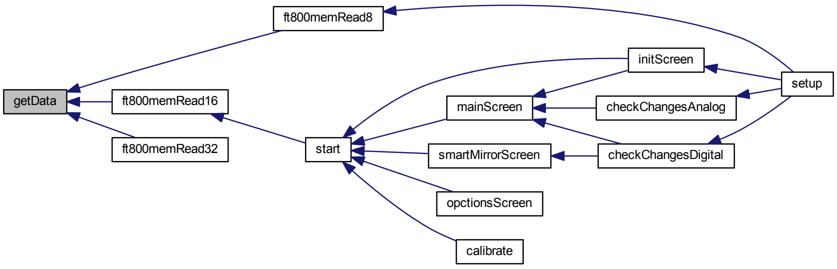
8 bit vcalue with obtained value ∗

Definition at line 31 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.11 unsigned int incCMDOffset (unsigned int *currentOffset,* unsigned char *commandSize*)

Adds commandSize to the currentOffset. Checks for 4K ring-buffer offset roll-over ∗

Parameters

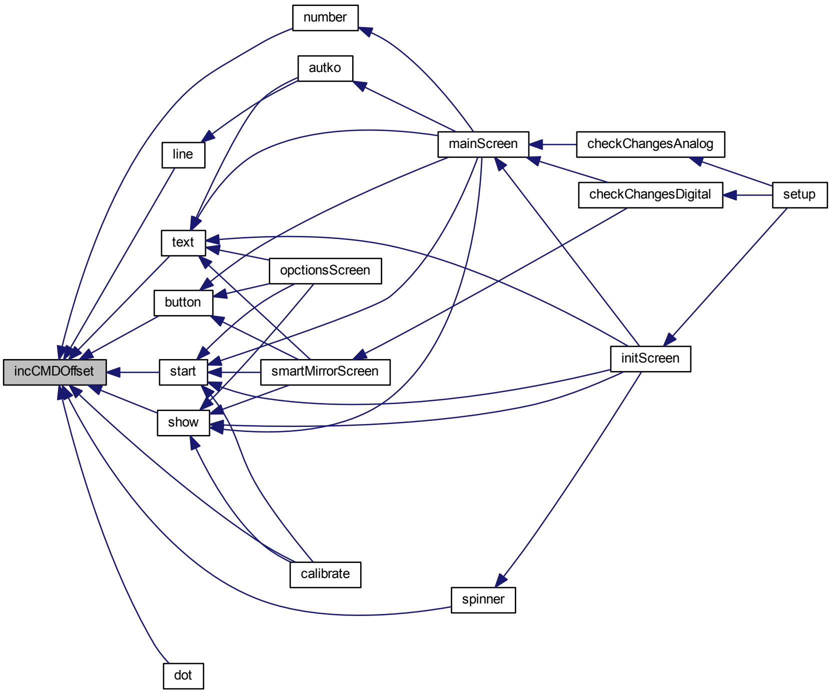
currentOffset	graphics processor command list pointer ∗
commandSize	number of bytes to increment the offset ∗

Returns

the new ring buffer pointer after adding the command ∗

Definition at line 294 of file FT800.cpp.

Here is the caller graph for this function:



4.1.2.12 void sendData (int *data*)

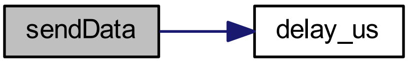
Function sending data to active device with using SPI interface ∗

Parameters

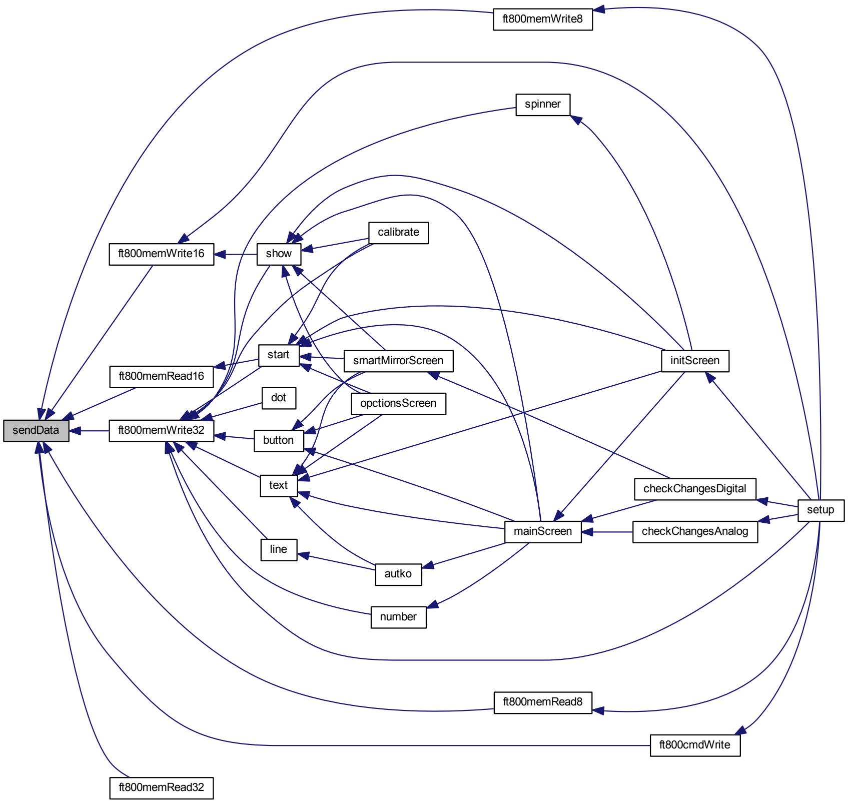
data	8 bit value to send to device ∗

Definition at line 19 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

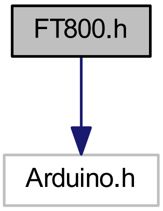


4.2 FT800.h File Reference

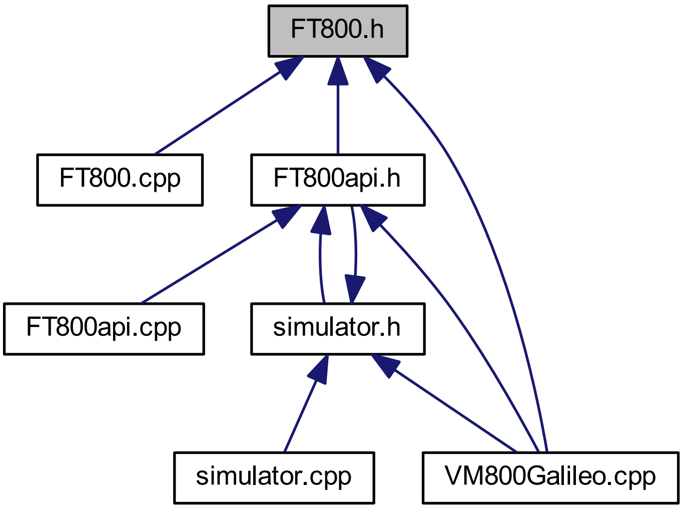
File containing declarations of all functions required to use with VM800.

```
#import <Arduino.h>
```

Include dependency graph for FT800.h:



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



Macros

- #define FT_DL_SIZE (8∗1024)
- #define FT_CMD_FIFO_SIZE (4∗1024)
- #define FT_CMD_SIZE (4)
- #define FT800_VERSION "1.9.0"
- #define RAM_CMD 0x108000UL
- #define RAM_DL 0x100000UL
- #define RAM_G 0x000000UL
- #define RAM_PAL 0x102000UL
- #define RAM_REG 0x102400UL

- #define REG_CLOCK 0x102408UL
- #define REG_CMD_DL 0x1024ecUL
- #define REG_CMD_READ 0x1024e4UL
- #define REG_CMD_WRITE 0x1024e8UL
- #define REG_CPURESET 0x10241cUL
- #define REG_CSPREAD 0x102464UL
- #define REG_DITHER 0x10245cUL
- #define REG_DLSWAP 0x102450UL
- #define REG_FRAMES 0x102404UL
- #define REG_FREQUENCY 0x10240cUL
- #define REG_GPIO 0x102490UL
- #define REG_GPIO_DIR 0x10248cUL
- #define REG_HCYCLE 0x102428UL
- #define REG_HOFFSET 0x10242cUL
- #define REG_HSIZE 0x102430UL
- #define REG_HSYNC0 0x102434UL
- #define REG_HSYNC1 0x102438UL
- #define REG_ID 0x102400UL
- #define REG_INT_EN 0x10249cUL
- #define REG_INT_FLAGS 0x102498UL
- #define REG_INT_MASK 0x1024a0UL
- #define REG_MACRO_0 0x1024c8UL
- #define REG_MACRO_1 0x1024ccUL
- #define REG_OUTBITS 0x102458UL
- #define REG_PCLK 0x10246cUL
- #define REG_PCLK_POL 0x102468UL
- #define REG_PLAY 0x102488UL
- #define REG_PLAYBACK_FORMAT 0x1024b4UL
- #define REG_PLAYBACK_FREQ 0x1024b0UL
- #define REG_PLAYBACK_LENGTH 0x1024a8UL
- #define REG_PLAYBACK_LOOP 0x1024b8UL
- #define REG_PLAYBACK_PLAY 0x1024bcUL
- #define REG_PLAYBACK_READPTR 0x1024acUL
- #define REG_PLAYBACK_START 0x1024a4UL
- #define REG_PWM_DUTY 0x1024c4UL
- #define REG_PWM_HZ 0x1024c0UL
- #define REG_RENDERMODE 0x102410UL
- #define REG_ROTATE 0x102454UL
- #define REG_SNAPSHOT 0x102418UL
- #define REG_SNAPY 0x102414UL
- #define REG_SOUND 0x102484UL
- #define REG_SWIZZLE 0x102460UL
- #define REG_TAG 0x102478UL
- #define REG_TAG_X 0x102470UL
- #define REG_TAG_Y 0x102474UL
- #define REG_TAP_CRC 0x102420UL
- #define REG_TAP_MASK 0x102424UL
- #define REG_TOUCH_ADC_MODE 0x1024f4UL
- #define REG_TOUCH_CHARGE 0x1024f8UL
- #define REG_TOUCH_DIRECT_XY 0x102574UL
- #define REG_TOUCH_DIRECT_Z1Z2 0x102578UL
- #define REG_TOUCH_MODE 0x1024f0UL
- #define REG_TOUCH_OVERSAMPLE 0x102500UL
- #define REG_TOUCH_RAW_XY 0x102508UL
- #define REG_TOUCH_RZ 0x10250cUL

- #define REG_TOUCH_RZTHRESH 0x102504UL
- #define REG_TOUCH_SCREEN_XY 0x102510UL
- #define REG_TOUCH_SETTLE 0x1024fcUL
- #define REG_TOUCH_TAG 0x102518UL
- #define REG_TOUCH_TAG_XY 0x102514UL
- #define REG_TOUCH_TRANSFORM_A 0x10251cUL
- #define REG_TOUCH_TRANSFORM_B 0x102520UL
- #define REG_TOUCH_TRANSFORM_C 0x102524UL
- #define REG_TOUCH_TRANSFORM_D 0x102528UL
- #define REG_TOUCH_TRANSFORM_E 0x10252cUL
- #define REG_TOUCH_TRANSFORM_F 0x102530UL
- #define REG_TRACKER 0x109000UL
- #define REG_VCYCLE 0x10243cUL
- #define REG_VOFFSET 0x102440UL
- #define REG_VOL_PB 0x10247cUL
- #define REG_VOL_SOUND 0x102480UL
- #define REG_VSIZE 0x102444UL
- #define REG_VSYNC0 0x102448UL
- #define REG_VSYNC1 0x10244cUL
- #define CMDBUF_SIZE 4096UL
- #define CMD_APPEND 0xffffff1eUL
- #define CMD_BGCOLOR 0xffffff09UL
- #define CMD_BUTTON 0xffffff0dUL
- #define CMD_CALIBRATE 0xffffff15UL
- #define CMD_CLOCK 0xffffff14UL
- #define CMD_COLDSTART 0xffffff32UL
- #define CMD_DIAL 0xffffff2dUL
- #define CMD_DLSTART 0xffffff00UL
- #define CMD_FGCOLOR 0xffffff0aUL
- #define CMD_GAUGE 0xffffff13UL
- #define CMD_GETMATRIX 0xffffff33UL
- #define CMD_GETPTR 0xffffff23UL
- #define CMD_GRADCOLOR 0xffffff34UL
- #define CMD_GRADIENT 0xffffff0bUL
- #define CMD_INFLATE 0xffffff22UL
- #define CMD_INTERRUPT 0xffffff02UL
- #define CMD_KEYS 0xffffff0eUL
- #define CMD_LOADIDENTITY 0xffffff26UL
- #define CMD_LOADIMAGE 0xffffff24UL
- #define CMD_LOGO 0xffffff31UL
- #define CMD_MEMCPY 0xffffff1dUL
- #define CMD_MEMCRC 0xffffff18UL
- #define CMD_MEMSET 0xffffff1bUL
- #define CMD_MEMWRITE 0xffffff1aUL
- #define CMD_MEMZERO 0xffffff1cUL
- #define CMD_NUMBER 0xffffff2eUL
- #define CMD_PROGRESS 0xffffff0fUL
- #define CMD_REGREAD 0xffffff19UL
- #define CMD_ROTATE 0xffffff29UL
- #define CMD_SCALE 0xffffff28UL
- #define CMD_SCREENSAVER 0xffffff2fUL
- #define CMD_SCROLLBAR 0xffffff11UL
- #define CMD_SETFONT 0xffffff2bUL
- #define CMD_SETMATRIX 0xffffff2aUL
- #define CMD_SKETCH 0xffffff30UL

- #define CMD_SLIDER 0xffffff10UL
- #define CMD_SNAPSHOT 0xffffff1fUL
- #define CMD_SPINNER 0xffffff16UL
- #define CMD_STOP 0xffffff17UL
- #define CMD_SWAP 0xffffff01UL
- #define CMD_TEXT 0xffffff0cUL
- #define CMD_TOGGLE 0xffffff12UL
- #define CMD_TRACK 0xffffff2cUL
- #define CMD_TRANSLATE 0xffffff27UL
- #define DL_ALPHA_FUNC 0x09000000UL
- #define DL_BITMAP_HANDLE 0x05000000UL
- #define DL_BITMAP_LAYOUT 0x07000000UL
- #define DL_BITMAP_SIZE 0x08000000UL
- #define DL_BITMAP_SOURCE 0x01000000UL
- #define DL_BITMAP_TFORM_A 0x15000000UL
- #define DL_BITMAP_TFORM_B 0x16000000UL
- #define DL_BITMAP_TFORM_C 0x17000000UL
- #define DL_BITMAP_TFORM_D 0x18000000UL
- #define DL_BITMAP_TFORM_E 0x19000000UL
- #define DL_BITMAP_TFORM_F 0x1A000000UL
- #define DL_BLEND_FUNC 0x0B000000UL
- #define DL_BEGIN 0x1F000000UL
- #define DL_CALL 0x1D000000UL
- #define DL_CLEAR 0x26000000UL
- #define DL_CELL 0x06000000UL
- #define DL_CLEAR_RGB 0x02000000UL
- #define DL_CLEAR_STENCIL 0x11000000UL
- #define DL_CLEAR_TAG 0x12000000UL
- #define DL_COLOR_A 0x0F000000UL
- #define DL_COLOR_MASK 0x20000000UL
- #define DL_COLOR_RGB 0x04000000UL
- #define DL_DISPLAY 0x00000000UL
- #define DL_END 0x21000000UL
- #define DL_JUMP 0x1E000000UL
- #define DL_LINE_WIDTH 0x0E000000UL
- #define DL_MACRO 0x25000000UL
- #define DL_POINT_SIZE 0x0D000000UL
- #define DL_RESTORE_CONTEXT 0x23000000UL
- #define DL_RETURN 0x24000000UL
- #define DL_SAVE_CONTEXT 0x22000000UL
- #define DL_SCISSOR_SIZE 0x1C000000UL
- #define DL_SCISSOR_XY 0x1B000000UL
- #define DL_STENCIL_FUNC 0x0A000000UL
- #define DL_STENCIL_MASK 0x13000000UL
- #define DL_STENCIL_OP 0x0C000000UL
- #define DL_TAG 0x03000000UL
- #define DL_TAG_MASK 0x14000000UL
- #define DL_VERTEX2F 0x40000000UL
- #define DL_VERTEX2II 0x02000000UL
- #define CLR_COL 0x4
- #define CLR_STN 0x2
- #define CLR_TAG 0x1
- #define DECR 4UL
- #define DECR_WRAP 7UL
- #define DLSWAP_DONE 0UL

- #define DLSWAP_FRAME 2UL
- #define DLSWAP_LINE 1UL
- #define DST_ALPHA 3UL
- #define EDGE_STRIP_A 7UL
- #define EDGE_STRIP_B 8UL
- #define EDGE_STRIP_L 6UL
- #define EDGE_STRIP_R 5UL
- #define EQUAL 5UL
- #define GEQUAL 4UL
- #define GREATER 3UL
- #define INCR 3UL
- #define INCR_WRAP 6UL
- #define INT_CMDEMPTY 32UL
- #define INT_CMDFLAG 64UL
- #define INT_CONVCOMPLETE 128UL
- #define INT_PLAYBACK 16UL
- #define INT_SOUND 8UL
- #define INT_SWAP 1UL
- #define INT_TAG 4UL
- #define INT_TOUCH 2UL
- #define INVERT 5UL
- #define KEEP 1UL
- #define L1 1UL
- #define L4 2UL
- #define L8 3UL
- #define LEQUAL 2UL
- #define LESS 1UL
- #define LINEAR_SAMPLES 0UL
- #define LINES 3UL
- #define LINE_STRIP 4UL
- #define NEAREST 0UL
- #define NEVER 0UL
- #define NOTEQUAL 6UL
- #define ONE 1UL
- #define ONE_MINUS_DST_ALPHA 5UL
- #define ONE_MINUS_SRC_ALPHA 4UL
- #define OPT_CENTER 1536UL
- #define OPT_CENTERX 512UL
- #define OPT_CENTERY 1024UL
- #define OPT_FLAT 256UL
- #define OPT_MONO 1UL
- #define OPT_NOBACK 4096UL
- #define OPT_NODL 2UL
- #define OPT_NOHANDS 49152UL
- #define OPT_NOHM 16384UL
- #define OPT_NOPOINTER 16384UL
- #define OPT_NOSECS 32768UL
- #define OPT_NOTICKS 8192UL
- #define OPT_RIGHTX 2048UL
- #define OPT_SIGNED 256UL
- #define PALETTED 8UL
- #define PLAYCOLOR 0x00a0a080
- #define FTPOINTS 2UL
- #define RECTS 9UL
- #define REPEAT 1UL

- #define REPLACE 2UL
- #define RGB332 4UL
- #define RGB565 7UL
- #define SRC_ALPHA 2UL
- #define TEXT8X8 9UL
- #define TEXTVGA 10UL
- #define TOUCHMODE_CONTINUOUS 3UL
- #define TOUCHMODE_FRAME 2UL
- #define TOUCHMODE_OFF 0UL
- #define TOUCHMODE_ONESHOT 1UL
- #define ULAW_SAMPLES 1UL
- #define ZERO 0UL
- #define RGB(r, g, b) ((((r) << 16) | (g) << 8) | (b))
- #define SQ(v) ((v) * (v))
- #define MIN(x, y) ((x) > (y) ? (y) : (x))
- #define MAX(x, y) ((x) > (y) ? (x) : (y))
- #define NOTE(n, sharp) (((n) - 'C') + ((sharp) * 128))
- #define F16(s) (((s) * 65536))
- #define INVALID_TOUCH_XY 0x8000
- #define ABS(x) ((x) > (0) ? (x) : (-x))
- #define LCD_QVGA
- #define xSDI 8
- #define xSDO 9
- #define xclock 10
- #define xPD 11
- #define xCS 12
- #define FT800_ACTIVE 0x00
- #define FT800_STANDBY 0x41
- #define FT800_SLEEP 0x42
- #define FT800_PWRDOWN 0x50
- #define FT800_CLKEXT 0x44
- #define FT800_CLK48M 0x62
- #define FT800_CLK36M 0x61
- #define FT800_CORERST 0x68
- #define FT800_GPUACTIVE 0x40
- #define MEM_WRITE 0x80
- #define MEM_READ 0x00
- #define RED 0xFF0000
- #define GREEN 0x00FF00
- #define BLUE 0x0000FF
- #define WHITE 0xFFFFFF
- #define BLACK 0x000000

Functions

- void delay_us (int us)
- void delay_ms (int ms)
- void sendData (int data)
- unsigned char getData ()
- void ft800memWrite8 (unsigned long ftAddress, unsigned char ftData8)
- void ft800memWrite16 (unsigned long ftAddress, unsigned int ftData16)
- void ft800memWrite32 (unsigned long ftAddress, unsigned long ftData32)
- unsigned char ft800memRead8 (unsigned long ftAddress)
- unsigned char ft800memRead16 (unsigned long ftAddress)
- unsigned long ft800memRead32 (unsigned long ftAddress)
- unsigned int incCMDOffset (unsigned int currentOffset, unsigned char commandSize)
- void ft800cmdWrite (unsigned char ftCommand)

4.2.1 Detailed Description

File containing declarations of all functions required to use with VM800.

Author

Daniel Sienkiewicz

Date

28 February 2016

4.2.2 Macro Definition Documentation

4.2.2.1 #define ABS(*x*) ((x) $>$ (0) ? (x) : (-x))

Definition at line 275 of file FT800.h.

4.2.2.2 #define BLACK 0x000000

Black colour

Definition at line 322 of file FT800.h.

4.2.2.3 #define BLUE 0x0000FF

Blue colour

Definition at line 320 of file FT800.h.

4.2.2.4 #define CLR_COL 0x4

Definition at line 193 of file FT800.h.

4.2.2.5 #define CLR_STN 0x2

Definition at line 194 of file FT800.h.

4.2.2.6 #define CLR_TAG 0x1

Definition at line 195 of file FT800.h.

4.2.2.7 #define CMD_APPEND 0xffffff1eUL

Definition at line 105 of file FT800.h.

4.2.2.8 #define CMD_BGCOLOR 0xffffff09UL

Definition at line 106 of file FT800.h.

4.2.2.9 #define CMD_BUTTON 0xffffff0dUL

Definition at line 107 of file FT800.h.

4.2.2.10 #define CMD_CALIBRATE 0xffffff15UL

Definition at line 108 of file FT800.h.

4.2.2.11 #define CMD_CLOCK 0xffffff14UL

Definition at line 109 of file FT800.h.

4.2.2.12 #define CMD_COLDSTART 0xffffff32UL

Definition at line 110 of file FT800.h.

4.2.2.13 #define CMD_DIAL 0xffffff2dUL

Definition at line 111 of file FT800.h.

4.2.2.14 #define CMD_DLSTART 0xffffff00UL

Definition at line 112 of file FT800.h.

4.2.2.15 #define CMD_FGCOLOR 0xffffff0aUL

Definition at line 113 of file FT800.h.

4.2.2.16 #define CMD_GAUGE 0xffffff13UL

Definition at line 114 of file FT800.h.

4.2.2.17 #define CMD_GETMATRIX 0xffffff33UL

Definition at line 115 of file FT800.h.

4.2.2.18 #define CMD_GETPTR 0xffffff23UL

Definition at line 116 of file FT800.h.

4.2.2.19 #define CMD_GRADCOLOR 0xffffff34UL

Definition at line 117 of file FT800.h.

4.2.2.20 #define CMD_GRADIENT 0xffffff0bUL

Definition at line 118 of file FT800.h.

4.2.2.21 #define CMD_INFLATE 0xffffff22UL

Definition at line 119 of file FT800.h.

4.2.2.22 #define CMD_INTERRUPT 0xffffff02UL

Definition at line 120 of file FT800.h.

4.2.2.23 #define CMD_KEYS 0xffffff0eUL

Definition at line 121 of file FT800.h.

4.2.2.24 #define CMD_LOADIDENTITY 0xffffff26UL

Definition at line 122 of file FT800.h.

4.2.2.25 #define CMD_LOADIMAGE 0xffffff24UL

Definition at line 123 of file FT800.h.

4.2.2.26 #define CMD_LOGO 0xffffff31UL

Definition at line 124 of file FT800.h.

4.2.2.27 #define CMD_MEMCPY 0xffffff1dUL

Definition at line 125 of file FT800.h.

4.2.2.28 #define CMD_MEMCRC 0xffffff18UL

Definition at line 126 of file FT800.h.

4.2.2.29 #define CMD_MEMSET 0xffffff1bUL

Definition at line 127 of file FT800.h.

4.2.2.30 #define CMD_MEMWRITE 0xffffff1aUL

Definition at line 128 of file FT800.h.

4.2.2.31 #define CMD_MEMZERO 0xffffff1cUL

Definition at line 129 of file FT800.h.

4.2.2.32 #define CMD_NUMBER 0xffffff2eUL

Definition at line 130 of file FT800.h.

4.2.2.33 #define CMD_PROGRESS 0xffffff0fUL

Definition at line 131 of file FT800.h.

4.2.2.34 #define CMD_REGREAD 0xffffff19UL

Definition at line 132 of file FT800.h.

4.2.2.35 #define CMD_ROTATE 0xffffff29UL

Definition at line 133 of file FT800.h.

4.2.2.36 #define CMD_SCALE 0xffffff28UL

Definition at line 134 of file FT800.h.

4.2.2.37 #define CMD_SCREENSAVER 0xffffff2fUL

Definition at line 135 of file FT800.h.

4.2.2.38 #define CMD_SCROLLBAR 0xffffff11UL

Definition at line 136 of file FT800.h.

4.2.2.39 #define CMD_SETFONT 0xffffff2bUL

Definition at line 137 of file FT800.h.

4.2.2.40 #define CMD_SETMATRIX 0xffffff2aUL

Definition at line 138 of file FT800.h.

4.2.2.41 #define CMD_SKETCH 0xffffff30UL

Definition at line 139 of file FT800.h.

4.2.2.42 #define CMD_SLIDER 0xffffff10UL

Definition at line 140 of file FT800.h.

4.2.2.43 #define CMD_SNAPSHOT 0xffffff1fUL

Definition at line 141 of file FT800.h.

4.2.2.44 #define CMD_SPINNER 0xffffff16UL

Definition at line 142 of file FT800.h.

4.2.2.45 #define CMD_STOP 0xffffff17UL

Definition at line 143 of file FT800.h.

4.2.2.46 #define CMD_SWAP 0xffffff01UL

Definition at line 144 of file FT800.h.

4.2.2.47 #define CMD_TEXT 0xffffff0cUL

Definition at line 145 of file FT800.h.

4.2.2.48 #define CMD_TOGGLE 0xffffff12UL

Definition at line 146 of file FT800.h.

4.2.2.49 #define CMD_TRACK 0xffffff2cUL

Definition at line 147 of file FT800.h.

4.2.2.50 #define CMD_TRANSLATE 0xffffff27UL

Definition at line 148 of file FT800.h.

4.2.2.51 #define CMDBUF_SIZE 4096UL

Definition at line 104 of file FT800.h.

4.2.2.52 #define DECR 4UL

Definition at line 196 of file FT800.h.

4.2.2.53 #define DECR_WRAP 7UL

Definition at line 197 of file FT800.h.

4.2.2.54 #define DL_ALPHA_FUNC 0x09000000UL

Definition at line 151 of file FT800.h.

4.2.2.55 #define DL_BEGIN 0x1F000000UL

Definition at line 163 of file FT800.h.

4.2.2.56 #define DL_BITMAP_HANDLE 0x05000000UL

Definition at line 152 of file FT800.h.

4.2.2.57 #define DL_BITMAP_LAYOUT 0x07000000UL

Definition at line 153 of file FT800.h.

4.2.2.58 #define DL_BITMAP_SIZE 0x08000000UL

Definition at line 154 of file FT800.h.

4.2.2.59 #define DL_BITMAP_SOURCE 0x01000000UL

Definition at line 155 of file FT800.h.

4.2.2.60 #define DL_BITMAP_TFORM_A 0x15000000UL

Definition at line 156 of file FT800.h.

4.2.2.61 #define DL_BITMAP_TFORM_B 0x16000000UL

Definition at line 157 of file FT800.h.

4.2.2.62 #define DL_BITMAP_TFORM_C 0x17000000UL

Definition at line 158 of file FT800.h.

4.2.2.63 #define DL_BITMAP_TFORM_D 0x18000000UL

Definition at line 159 of file FT800.h.

4.2.2.64 #define DL_BITMAP_TFORM_E 0x19000000UL

Definition at line 160 of file FT800.h.

4.2.2.65 #define DL_BITMAP_TFORM_F 0x1A000000UL

Definition at line 161 of file FT800.h.

4.2.2.66 #define DL_BLEND_FUNC 0x0B000000UL

Definition at line 162 of file FT800.h.

4.2.2.67 #define DL_CALL 0x1D000000UL

Definition at line 164 of file FT800.h.

4.2.2.68 #define DL_CELL 0x06000000UL

Definition at line 166 of file FT800.h.

4.2.2.69 #define DL_CLEAR 0x26000000UL

Definition at line 165 of file FT800.h.

4.2.2.70 #define DL_CLEAR_RGB 0x02000000UL

Definition at line 167 of file FT800.h.

4.2.2.71 #define DL_CLEAR_STENCIL 0x11000000UL

Definition at line 168 of file FT800.h.

4.2.2.72 #define DL_CLEAR_TAG 0x12000000UL

Definition at line 169 of file FT800.h.

4.2.2.73 #define DL_COLOR_A 0x0F000000UL

Definition at line 170 of file FT800.h.

4.2.2.74 #define DL_COLOR_MASK 0x20000000UL

Definition at line 171 of file FT800.h.

4.2.2.75 #define DL_COLOR_RGB 0x04000000UL

Definition at line 172 of file FT800.h.

4.2.2.76 #define DL_DISPLAY 0x00000000UL

Definition at line 173 of file FT800.h.

4.2.2.77 #define DL_END 0x21000000UL

Definition at line 174 of file FT800.h.

4.2.2.78 #define DL_JUMP 0x1E000000UL

Definition at line 175 of file FT800.h.

4.2.2.79 #define DL_LINE_WIDTH 0x0E000000UL

Definition at line 176 of file FT800.h.

4.2.2.80 #define DL_MACRO 0x25000000UL

Definition at line 177 of file FT800.h.

4.2.2.81 #define DL_POINT_SIZE 0x0D000000UL

Definition at line 178 of file FT800.h.

4.2.2.82 #define DL_RESTORE_CONTEXT 0x23000000UL

Definition at line 179 of file FT800.h.

4.2.2.83 #define DL_RETURN 0x24000000UL

Definition at line 180 of file FT800.h.

4.2.2.84 #define DL_SAVE_CONTEXT 0x22000000UL

Definition at line 181 of file FT800.h.

4.2.2.85 #define DL_SCISSOR_SIZE 0x1C000000UL

Definition at line 182 of file FT800.h.

4.2.2.86 #define DL_SCISSOR_XY 0x1B000000UL

Definition at line 183 of file FT800.h.

4.2.2.87 #define DL_STENCIL_FUNC 0x0A000000UL

Definition at line 184 of file FT800.h.

4.2.2.88 #define DL_STENCIL_MASK 0x13000000UL

Definition at line 185 of file FT800.h.

4.2.2.89 #define DL_STENCIL_OP 0x0C000000UL

Definition at line 186 of file FT800.h.

4.2.2.90 #define DL_TAG 0x03000000UL

Definition at line 187 of file FT800.h.

4.2.2.91 #define DL_TAG_MASK 0x14000000UL

Definition at line 188 of file FT800.h.

4.2.2.92 #define DL_VERTEX2F 0x40000000UL

Definition at line 189 of file FT800.h.

4.2.2.93 #define DL_VERTEX2II 0x02000000UL

Definition at line 190 of file FT800.h.

4.2.2.94 #define DLSWAP_DONE 0UL

Definition at line 198 of file FT800.h.

4.2.2.95 #define DLSWAP_FRAME 2UL

Definition at line 199 of file FT800.h.

4.2.2.96 #define DLSWAP_LINE 1UL

Definition at line 200 of file FT800.h.

4.2.2.97 #define DST_ALPHA 3UL

Definition at line 201 of file FT800.h.

4.2.2.98 #define EDGE_STRIP_A 7UL

Definition at line 202 of file FT800.h.

4.2.2.99 #define EDGE_STRIP_B 8UL

Definition at line 203 of file FT800.h.

4.2.2.100 #define EDGE_STRIP_L 6UL

Definition at line 204 of file FT800.h.

4.2.2.101 #define EDGE_STRIP_R 5UL

Definition at line 205 of file FT800.h.

4.2.2.102 #define EQUAL 5UL

Definition at line 206 of file FT800.h.

4.2.2.103 #define F16(*s*) (((s) ∗ 65536))

Definition at line 273 of file FT800.h.

4.2.2.104 #define FT800_ACTIVE 0x00

Initializes FT800

Definition at line 303 of file FT800.h.

4.2.2.105 #define FT800_CLK36M 0x61

Select 36MHz PLL

Definition at line 309 of file FT800.h.

4.2.2.106 #define FT800_CLK48M 0x62

Select 48MHz PLL

Definition at line 308 of file FT800.h.

4.2.2.107 #define FT800_CLKEXT 0x44

Select external clock source

Definition at line 307 of file FT800.h.

4.2.2.108 #define FT800_CORERST 0x68

Reset core - all registers default

Definition at line 310 of file FT800.h.

4.2.2.109 #define FT800_GPUACTIVE 0x40

Definition at line 311 of file FT800.h.

4.2.2.110 #define FT800_PWRDOWN 0x50

Place FT800 in Power Down (core off)

Definition at line 306 of file FT800.h.

4.2.2.111 #define FT800_SLEEP 0x42

Place FT800 in Sleep (clk off)

Definition at line 305 of file FT800.h.

4.2.2.112 #define FT800_STANDBY 0x41

Place FT800 in Standby (clk running)

Definition at line 304 of file FT800.h.

4.2.2.113 #define FT800_VERSION "1.9.0"

Definition at line 16 of file FT800.h.

4.2.2.114 #define FT_CMD_FIFO_SIZE (4∗1024)

4KB coprocessor Fifo size

Definition at line 13 of file FT800.h.

4.2.2.115 #define FT_CMD_SIZE (4)

4 byte per coprocessor command of EVE

Definition at line 14 of file FT800.h.

4.2.2.116 #define FT_DL_SIZE (8∗1024)

8KB Display List buffer size

Definition at line 12 of file FT800.h.

4.2.2.117 #define FTPOINTS 2UL

"POINTS" is a reserved word

Definition at line 251 of file FT800.h.

4.2.2.118 #define GEQUAL 4UL

Definition at line 207 of file FT800.h.

4.2.2.119 #define GREATER 3UL

Definition at line 208 of file FT800.h.

4.2.2.120 #define GREEN 0x00FF00

Green colour

Definition at line 319 of file FT800.h.

4.2.2.121 #define INCR 3UL

Definition at line 209 of file FT800.h.

4.2.2.122 #define INCR_WRAP 6UL

Definition at line 210 of file FT800.h.

4.2.2.123 #define INT_CMDEMPTY 32UL

Definition at line 211 of file FT800.h.

4.2.2.124 #define INT_CMDFLAG 64UL

Definition at line 212 of file FT800.h.

4.2.2.125 #define INT_CONVCOMPLETE 128UL

Definition at line 213 of file FT800.h.

4.2.2.126 #define INT_PLAYBACK 16UL

Definition at line 214 of file FT800.h.

4.2.2.127 #define INT_SOUND 8UL

Definition at line 215 of file FT800.h.

4.2.2.128 #define INT_SWAP 1UL

Definition at line 216 of file FT800.h.

4.2.2.129 #define INT_TAG 4UL

Definition at line 217 of file FT800.h.

4.2.2.130 #define INT_TOUCH 2UL

Definition at line 218 of file FT800.h.

4.2.2.131 #define INVALID_TOUCH_XY 0x8000

Definition at line 274 of file FT800.h.

4.2.2.132 #define INVERT 5UL

Definition at line 219 of file FT800.h.

4.2.2.133 #define KEEP 1UL

Definition at line 220 of file FT800.h.

4.2.2.134 #define L1 1UL

Definition at line 221 of file FT800.h.

4.2.2.135 #define L4 2UL

Definition at line 222 of file FT800.h.

4.2.2.136 #define L8 3UL

Definition at line 223 of file FT800.h.

4.2.2.137 #define LCD_QVGA

QVGA = 320 x 240 (VM800B/C 3.5")

Definition at line 280 of file FT800.h.

4.2.2.138 #define LEQUAL 2UL

Definition at line 224 of file FT800.h.

4.2.2.139 #define LESS 1UL

Definition at line 225 of file FT800.h.

4.2.2.140 #define LINE_STRIP 4UL

Definition at line 228 of file FT800.h.

4.2.2.141 #define LINEAR_SAMPLES 0UL

Definition at line 226 of file FT800.h.

4.2.2.142 #define LINES 3UL

Definition at line 227 of file FT800.h.

4.2.2.143 #define MAX(*x, y*) ((x) > (y) ? (x) : (y))

Definition at line 271 of file FT800.h.

4.2.2.144 #define MEM_READ 0x00

FT800 Host Memory Read

Definition at line 315 of file FT800.h.

4.2.2.145 #define MEM_WRITE 0x80

FT800 Host Memory Write

Definition at line 314 of file FT800.h.

4.2.2.146 #define MIN(*x*, *y*) ((x) > (y) ? (y) : (x))

Definition at line 270 of file FT800.h.

4.2.2.147 #define NEAREST 0UL

Definition at line 229 of file FT800.h.

4.2.2.148 #define NEVER 0UL

Definition at line 230 of file FT800.h.

4.2.2.149 #define NOTE(*n*, *sharp*) (((n) - 'C') + ((sharp) * 128))

Definition at line 272 of file FT800.h.

4.2.2.150 #define NOTEQUAL 6UL

Definition at line 231 of file FT800.h.

4.2.2.151 #define ONE 1UL

Definition at line 232 of file FT800.h.

4.2.2.152 #define ONE_MINUS_DST_ALPHA 5UL

Definition at line 233 of file FT800.h.

4.2.2.153 #define ONE_MINUS_SRC_ALPHA 4UL

Definition at line 234 of file FT800.h.

4.2.2.154 #define OPT_CENTER 1536UL

Definition at line 235 of file FT800.h.

4.2.2.155 #define OPT_CENTERX 512UL

Definition at line 236 of file FT800.h.

4.2.2.156 #define OPT_CENTERY 1024UL

Definition at line 237 of file FT800.h.

4.2.2.157 #define OPT_FLAT 256UL

Definition at line 238 of file FT800.h.

4.2.2.158 #define OPT_MONO 1UL

Definition at line 239 of file FT800.h.

4.2.2.159 #define OPT_NOBACK 4096UL

Definition at line 240 of file FT800.h.

4.2.2.160 #define OPT_NODL 2UL

Definition at line 241 of file FT800.h.

4.2.2.161 #define OPT_NOHANDS 49152UL

Definition at line 242 of file FT800.h.

4.2.2.162 #define OPT_NOHM 16384UL

Definition at line 243 of file FT800.h.

4.2.2.163 #define OPT_NOPOINTER 16384UL

Definition at line 244 of file FT800.h.

4.2.2.164 #define OPT_NOSECS 32768UL

Definition at line 245 of file FT800.h.

4.2.2.165 #define OPT_NOTICKS 8192UL

Definition at line 246 of file FT800.h.

4.2.2.166 #define OPT_RIGHTX 2048UL

Definition at line 247 of file FT800.h.

4.2.2.167 #define OPT_SIGNED 256UL

Definition at line 248 of file FT800.h.

4.2.2.168 #define PALETTED 8UL

Definition at line 249 of file FT800.h.

4.2.2.169 #define PLAYCOLOR 0x00a0a080

Definition at line 250 of file FT800.h.

4.2.2.170 #define RAM_CMD 0x108000UL

Definition at line 20 of file FT800.h.

4.2.2.171 #define RAM_DL 0x100000UL

Definition at line 21 of file FT800.h.

4.2.2.172 #define RAM_G 0x000000UL

Definition at line 22 of file FT800.h.

4.2.2.173 #define RAM_PAL 0x102000UL

Definition at line 23 of file FT800.h.

4.2.2.174 #define RAM_REG 0x102400UL

Definition at line 24 of file FT800.h.

4.2.2.175 #define RECTS 9UL

Definition at line 252 of file FT800.h.

4.2.2.176 #define RED 0xFF0000

Red colour

Definition at line 318 of file FT800.h.

4.2.2.177 #define REG_CLOCK 0x102408UL

Definition at line 27 of file FT800.h.

4.2.2.178 #define REG_CMD_DL 0x1024ecUL

Definition at line 28 of file FT800.h.

4.2.2.179 #define REG_CMD_READ 0x1024e4UL

Definition at line 29 of file FT800.h.

4.2.2.180 #define REG_CMD_WRITE 0x1024e8UL

Definition at line 30 of file FT800.h.

4.2.2.181 #define REG_CPURESET 0x10241cUL

Definition at line 31 of file FT800.h.

4.2.2.182 #define REG_CSPREAD 0x102464UL

Definition at line 32 of file FT800.h.

4.2.2.183 #define REG_DITHER 0x10245cUL

Definition at line 33 of file FT800.h.

4.2.2.184 #define REG_DLSWAP 0x102450UL

Definition at line 34 of file FT800.h.

4.2.2.185 #define REG_FRAMES 0x102404UL

Definition at line 35 of file FT800.h.

4.2.2.186 #define REG_FREQUENCY 0x10240cUL

Definition at line 36 of file FT800.h.

4.2.2.187 #define REG_GPIO 0x102490UL

Definition at line 37 of file FT800.h.

4.2.2.188 #define REG_GPIO_DIR 0x10248cUL

Definition at line 38 of file FT800.h.

4.2.2.189 #define REG_HCYCLE 0x102428UL

Definition at line 39 of file FT800.h.

4.2.2.190 #define REG_HOFFSET 0x10242cUL

Definition at line 40 of file FT800.h.

4.2.2.191 #define REG_HSIZE 0x102430UL

Definition at line 41 of file FT800.h.

4.2.2.192 #define REG_HSYNC0 0x102434UL

Definition at line 42 of file FT800.h.

4.2.2.193 #define REG_HSYNC1 0x102438UL

Definition at line 43 of file FT800.h.

4.2.2.194 #define REG_ID 0x102400UL

Definition at line 44 of file FT800.h.

4.2.2.195 #define REG_INT_EN 0x10249cUL

Definition at line 45 of file FT800.h.

4.2.2.196 #define REG_INT_FLAGS 0x102498UL

Definition at line 46 of file FT800.h.

4.2.2.197 #define REG_INT_MASK 0x1024a0UL

Definition at line 47 of file FT800.h.

4.2.2.198 #define REG_MACRO_0 0x1024c8UL

Definition at line 48 of file FT800.h.

4.2.2.199 #define REG_MACRO_1 0x1024ccUL

Definition at line 49 of file FT800.h.

4.2.2.200 #define REG_OUTBITS 0x102458UL

Definition at line 50 of file FT800.h.

4.2.2.201 #define REG_PCLK 0x10246cUL

Definition at line 51 of file FT800.h.

4.2.2.202 #define REG_PCLK_POL 0x102468UL

Definition at line 52 of file FT800.h.

4.2.2.203 #define REG_PLAY 0x102488UL

Definition at line 53 of file FT800.h.

4.2.2.204 #define REG_PLAYBACK_FORMAT 0x1024b4UL

Definition at line 54 of file FT800.h.

4.2.2.205 #define REG_PLAYBACK_FREQ 0x1024b0UL

Definition at line 55 of file FT800.h.

4.2.2.206 #define REG_PLAYBACK_LENGTH 0x1024a8UL

Definition at line 56 of file FT800.h.

4.2.2.207 #define REG_PLAYBACK_LOOP 0x1024b8UL

Definition at line 57 of file FT800.h.

4.2.2.208 #define REG_PLAYBACK_PLAY 0x1024bcUL

Definition at line 58 of file FT800.h.

4.2.2.209 #define REG_PLAYBACK_READPTR 0x1024acUL

Definition at line 59 of file FT800.h.

4.2.2.210 #define REG_PLAYBACK_START 0x1024a4UL

Definition at line 60 of file FT800.h.

4.2.2.211 #define REG_PWM_DUTY 0x1024c4UL

Definition at line 61 of file FT800.h.

4.2.2.212 #define REG_PWM_HZ 0x1024c0UL

Definition at line 62 of file FT800.h.

4.2.2.213 #define REG_RENDERMODE 0x102410UL

Definition at line 63 of file FT800.h.

4.2.2.214 #define REG_ROTATE 0x102454UL

Definition at line 64 of file FT800.h.

4.2.2.215 #define REG_SNAPSHOT 0x102418UL

Definition at line 65 of file FT800.h.

4.2.2.216 #define REG_SNAPY 0x102414UL

Definition at line 66 of file FT800.h.

4.2.2.217 #define REG_SOUND 0x102484UL

Definition at line 67 of file FT800.h.

4.2.2.218 #define REG_SWIZZLE 0x102460UL

Definition at line 68 of file FT800.h.

4.2.2.219 #define REG_TAG 0x102478UL

Definition at line 69 of file FT800.h.

4.2.2.220 #define REG_TAG_X 0x102470UL

Definition at line 70 of file FT800.h.

4.2.2.221 #define REG_TAG_Y 0x102474UL

Definition at line 71 of file FT800.h.

4.2.2.222 #define REG_TAP_CRC 0x102420UL

Definition at line 72 of file FT800.h.

4.2.2.223 #define REG_TAP_MASK 0x102424UL

Definition at line 73 of file FT800.h.

4.2.2.224 #define REG_TOUCH_ADC_MODE 0x1024f4UL

Definition at line 74 of file FT800.h.

4.2.2.225 #define REG_TOUCH_CHARGE 0x1024f8UL

Definition at line 75 of file FT800.h.

4.2.2.226 #define REG_TOUCH_DIRECT_XY 0x102574UL

Definition at line 76 of file FT800.h.

4.2.2.227 #define REG_TOUCH_DIRECT_Z1Z2 0x102578UL

Definition at line 77 of file FT800.h.

4.2.2.228 #define REG_TOUCH_MODE 0x1024f0UL

Definition at line 78 of file FT800.h.

4.2.2.229 #define REG_TOUCH_OVERSAMPLE 0x102500UL

Definition at line 79 of file FT800.h.

4.2.2.230 #define REG_TOUCH_RAW_XY 0x102508UL

Definition at line 80 of file FT800.h.

4.2.2.231 #define REG_TOUCH_RZ 0x10250cUL

Definition at line 81 of file FT800.h.

4.2.2.232 #define REG_TOUCH_RZTHRESH 0x102504UL

Definition at line 82 of file FT800.h.

4.2.2.233 #define REG_TOUCH_SCREEN_XY 0x102510UL

Definition at line 83 of file FT800.h.

4.2.2.234 #define REG_TOUCH_SETTLE 0x1024fcUL

Definition at line 84 of file FT800.h.

4.2.2.235 #define REG_TOUCH_TAG 0x102518UL

Definition at line 85 of file FT800.h.

4.2.2.236 #define REG_TOUCH_TAG_XY 0x102514UL

Definition at line 86 of file FT800.h.

4.2.2.237 #define REG_TOUCH_TRANSFORM_A 0x10251cUL

Definition at line 87 of file FT800.h.

4.2.2.238 #define REG_TOUCH_TRANSFORM_B 0x102520UL

Definition at line 88 of file FT800.h.

4.2.2.239 #define REG_TOUCH_TRANSFORM_C 0x102524UL

Definition at line 89 of file FT800.h.

4.2.2.240 #define REG_TOUCH_TRANSFORM_D 0x102528UL

Definition at line 90 of file FT800.h.

4.2.2.241 #define REG_TOUCH_TRANSFORM_E 0x10252cUL

Definition at line 91 of file FT800.h.

4.2.2.242 #define REG_TOUCH_TRANSFORM_F 0x102530UL

Definition at line 92 of file FT800.h.

4.2.2.243 #define REG_TRACKER 0x109000UL

Definition at line 93 of file FT800.h.

4.2.2.244 #define REG_VCYCLE 0x10243cUL

Definition at line 94 of file FT800.h.

4.2.2.245 #define REG_VOFFSET 0x102440UL

Definition at line 95 of file FT800.h.

4.2.2.246 #define REG_VOL_PB 0x10247cUL

Definition at line 96 of file FT800.h.

4.2.2.247 #define REG_VOL_SOUND 0x102480UL

Definition at line 97 of file FT800.h.

4.2.2.248 #define REG_VSIZE 0x102444UL

Definition at line 98 of file FT800.h.

4.2.2.249 #define REG_VSYNC0 0x102448UL

Definition at line 99 of file FT800.h.

4.2.2.250 #define REG_VSYNC1 0x10244cUL

Definition at line 100 of file FT800.h.

4.2.2.251 #define REPEAT 1UL

Definition at line 253 of file FT800.h.

4.2.2.252 #define REPLACE 2UL

Definition at line 254 of file FT800.h.

4.2.2.253 #define RGB(*r,* *g,* *b*) ((((r) $<<$ 16) | (g) $<<$ 8) | (b))

Definition at line 268 of file FT800.h.

4.2.2.254 #define RGB332 4UL

Definition at line 255 of file FT800.h.

4.2.2.255 #define RGB565 7UL

Definition at line 256 of file FT800.h.

4.2.2.256 #define SQ(*v*) ((v) $*$ (v))

Definition at line 269 of file FT800.h.

4.2.2.257 #define SRC_ALPHA 2UL

Definition at line 257 of file FT800.h.

4.2.2.258 #define TEXT8X8 9UL

Definition at line 258 of file FT800.h.

4.2.2.259 #define TEXTVGA 10UL

Definition at line 259 of file FT800.h.

4.2.2.260 #define TOUCHMODE_CONTINUOUS 3UL

Definition at line 260 of file FT800.h.

4.2.2.261 #define TOUCHMODE_FRAME 2UL

Definition at line 261 of file FT800.h.

4.2.2.262 #define TOUCHMODE_OFF 0UL

Definition at line 262 of file FT800.h.

4.2.2.263 #define TOUCHMODE_ONESHOT 1UL

Definition at line 263 of file FT800.h.

4.2.2.264 #define ULAW_SAMPLES 1UL

Definition at line 264 of file FT800.h.

4.2.2.265 #define WHITE 0xFFFFFF

White colour

Definition at line 321 of file FT800.h.

4.2.2.266 #define xclock 10

Clock line - output for Galileo

Definition at line 285 of file FT800.h.

4.2.2.267 #define xCS 12

Chip Select line for screen - output for Galileo

Definition at line 287 of file FT800.h.

4.2.2.268 #define xPD 11

PD line for screen - output for Galileo

Definition at line 286 of file FT800.h.

4.2.2.269 #define xSDI 8

SDI line for SPI interface - input for Galileo

Definition at line 283 of file FT800.h.

4.2.2.270 #define xSDO 9

SDO line for SPI interface - output for Galileo

Definition at line 284 of file FT800.h.

4.2.2.271 #define ZERO 0UL

Definition at line 265 of file FT800.h.

4.2.3 Function Documentation

4.2.3.1 void delay_ms (int *ms*)

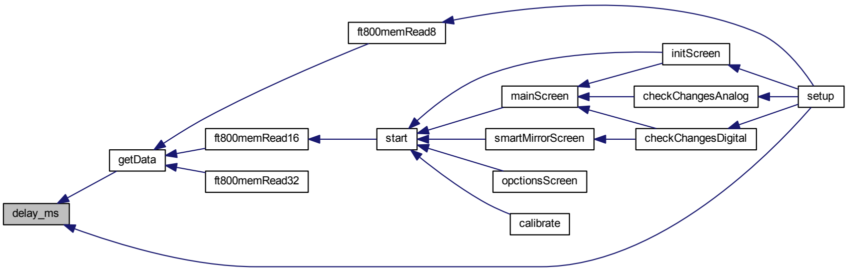
Pauses the program for the amount of time (in milisecond) specified as parameter ∗

Parameters

ms	milisecond to delay ∗

Definition at line 15 of file FT800.cpp.

Here is the caller graph for this function:



4.2.3.2 void delay_us (int *us*)

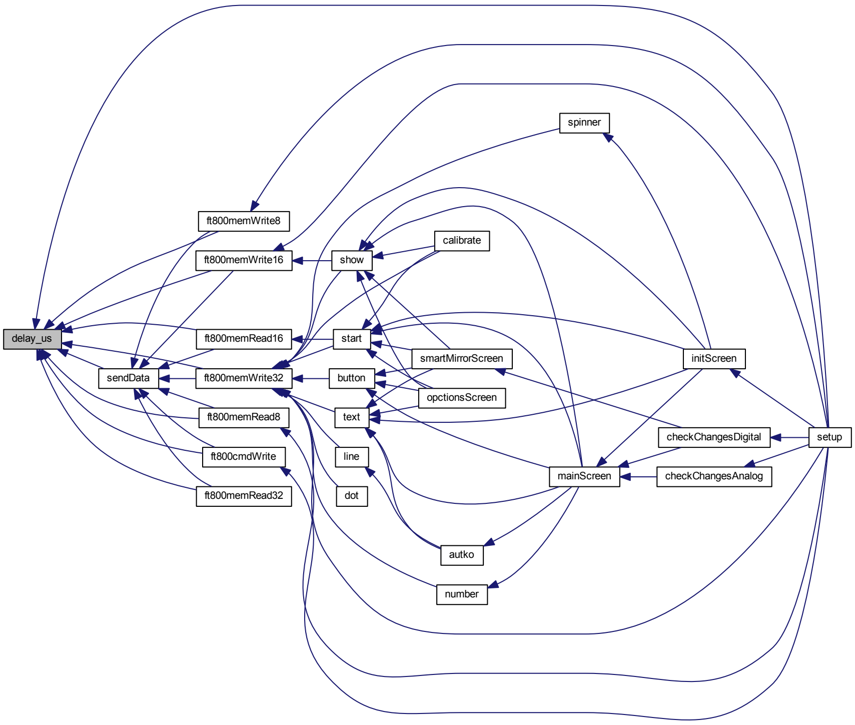
Pauses the program for the amount of time (in microsecond) specified as parameter ∗

Parameters

us	microseconds to delay ∗

Definition at line 11 of file FT800.cpp.

Here is the caller graph for this function:



4.2.3.3 void ft800cmdWrite (unsigned char *ftCommand*)

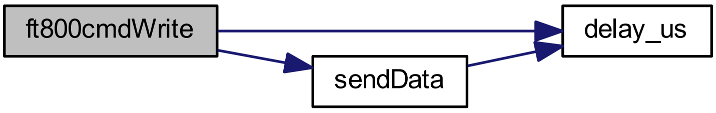
Sends FT800 command ∗

Parameters

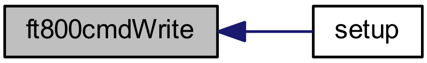
ftCommand	command to send to device ∗

Definition at line 304 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.4 unsigned char ft800memRead16 (unsigned long *ftAddress*)

Funtion to read 16 bit value from active device with using SPI interface ∗

Parameters

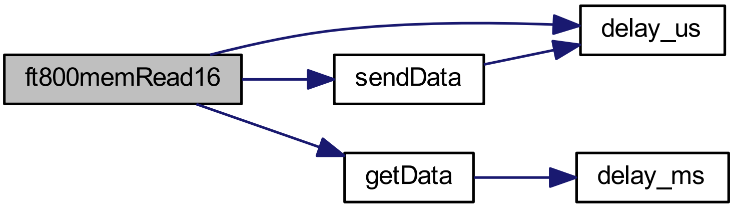
ftAddress	FT800 memory space address (24 bits) ∗

Returns

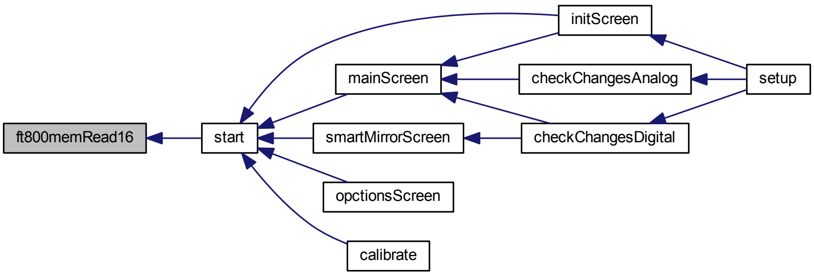
16 bit data obtained from device ∗

Definition at line 177 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.5 unsigned long ft800memRead32 (unsigned long *ftAddress*)

Funtion to read 32 bit value from active device with using SPI interface ∗

Parameters

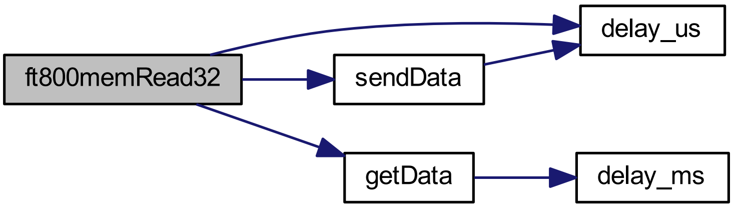
ftAddress	FT800 memory space address (24 bits) ∗

Returns

32 bit data obtained from device ∗

Definition at line 226 of file FT800.cpp.

Here is the call graph for this function:



4.2.3.6 unsigned char ft800memRead8 (unsigned long *ftAddress*)

Funtion to read 8 bit value from active device with using SPI interface ∗

Parameters

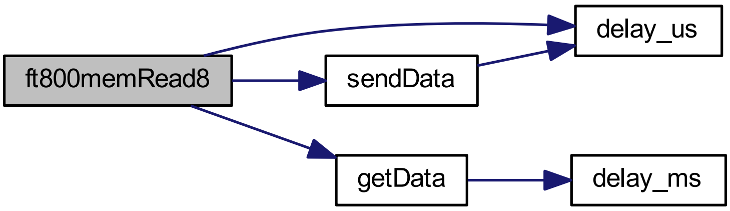
ftAddress	FT800 memory space address (24 bits) ∗

Returns

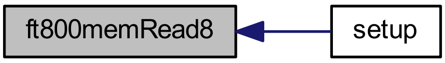
8 bit data obtained from device ∗

Definition at line 143 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.7 void ft800memWrite16 (unsigned long *ftAddress*, unsigned int *ftData16*)

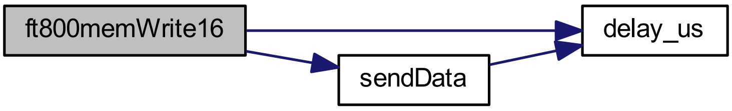
Funtion to send 16 bit value to active device with using SPI interface ∗

Parameters

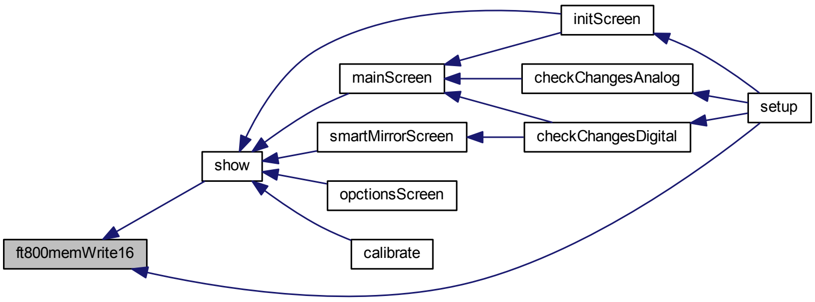
ftAddress	FT800 memory space address (24 bits) ∗
ftData8	a byte to send ∗

Definition at line 73 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.8 void ft800memWrite32 (unsigned long *ftAddress,* unsigned long *ftData32*)

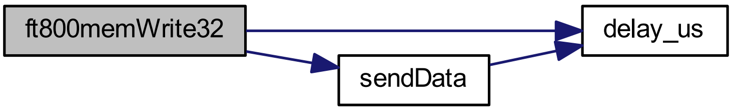
Funtion to send 32 bit value to active device with using SPI interface ∗

Parameters

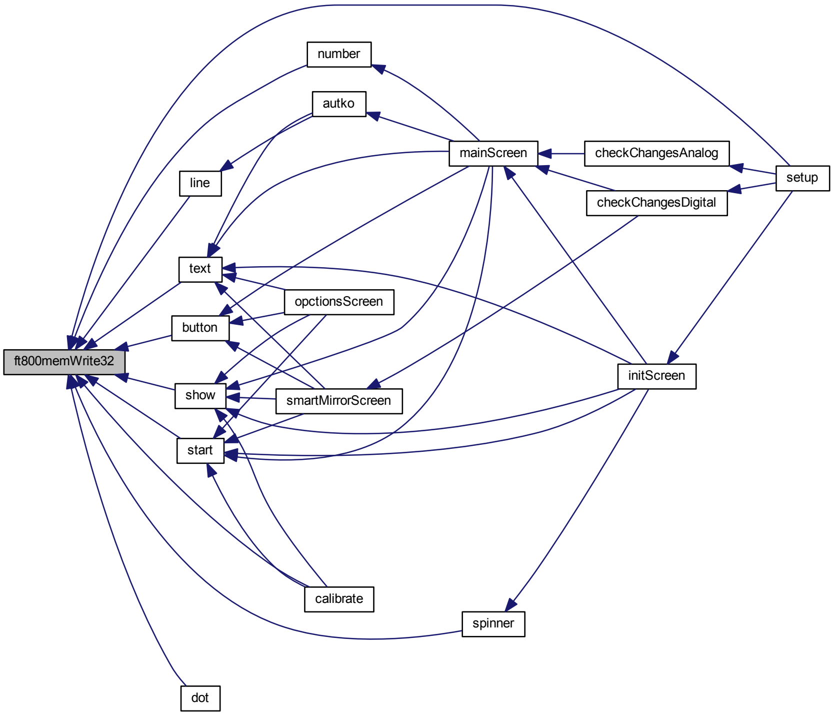
ftAddress	FT800 memory space address (24 bits) ∗
ftData8	a byte to send ∗

Definition at line 105 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.9 void ft800memWrite8 (unsigned long *ftAddress*, unsigned char *ftData8*)

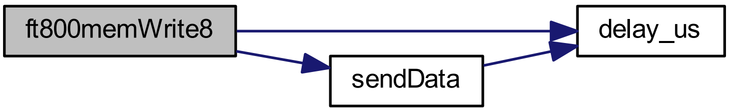
Funtion to send 8 bit value to active device with using SPI interface ∗

Parameters

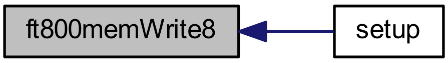
ftAddress	FT800 memory space address (24 bits) ∗
ftData8	a byte to send ∗

Definition at line 45 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.10 unsigned char getData ( )

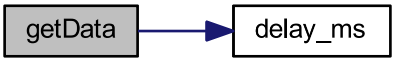
Function getting data from active device with using SPI interface ∗

Returns

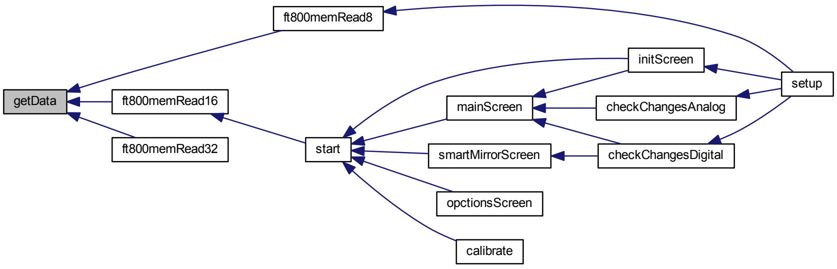
8 bit vcalue with obtained value ∗

Definition at line 31 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.2.3.11 unsigned int incCMDOffset (unsigned int *currentOffset*, unsigned char *commandSize*)

Adds commandSize to the currentOffset. Checks for 4K ring-buffer offset roll-over ∗

Parameters

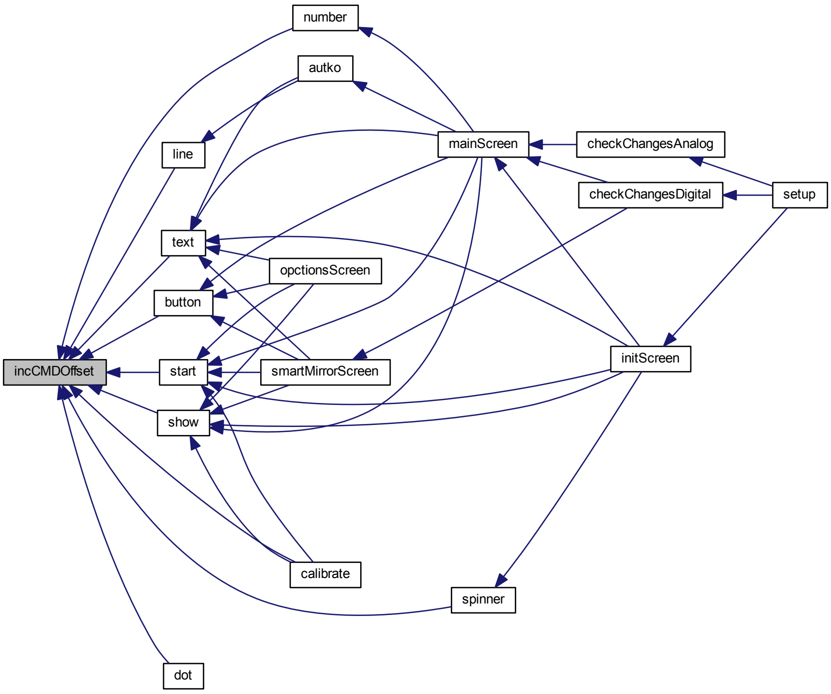
currentOffset	graphics processor command list pointer ∗
commandSize	number of bytes to increment the offset ∗

Returns

the new ring buffer pointer after adding the command ∗

Definition at line 294 of file FT800.cpp.

Here is the caller graph for this function:



4.2.3.12 void sendData (int *data*)

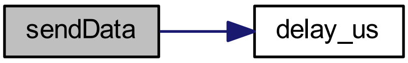
Function sending data to active device with using SPI interface ∗

Parameters

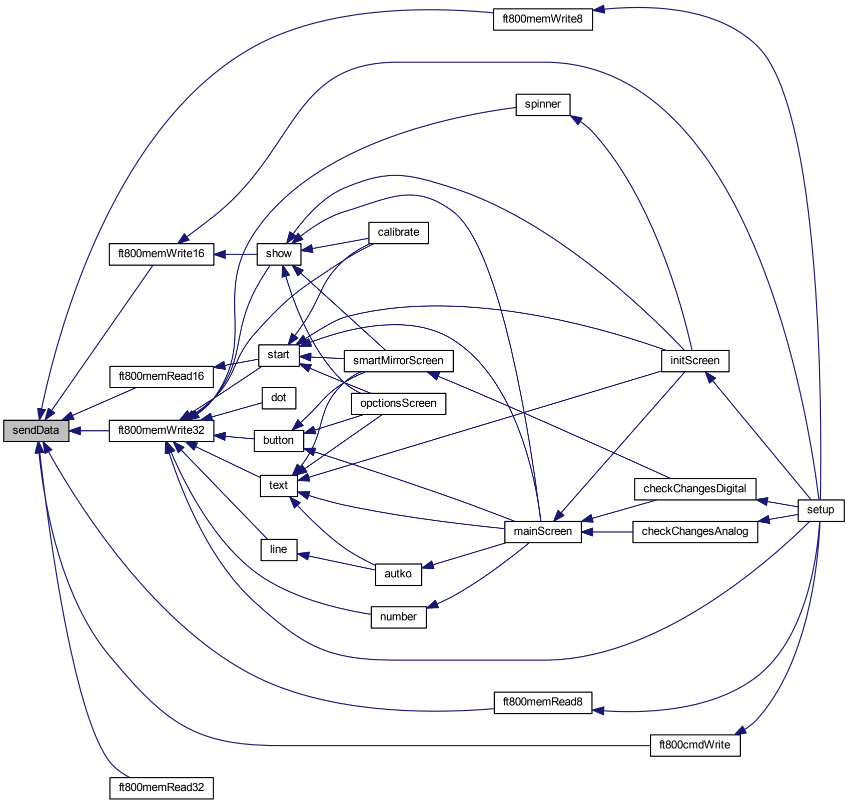
data	8 bit value to send to device ∗

Definition at line 19 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

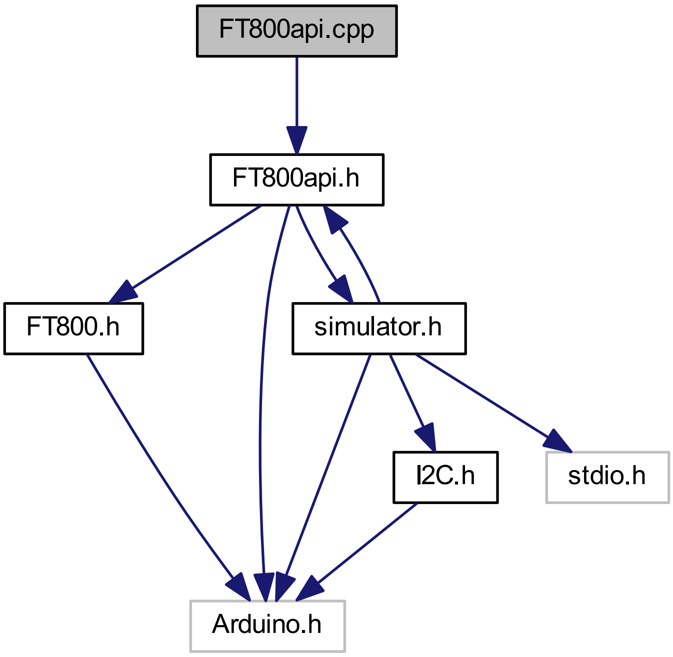


4.3 FT800api.cpp File Reference

File containing declarations of all API functions for VM800.

```
#include "FT800api.h"
```

Include dependency graph for FT800api.cpp:



Functions

- void initScreen ()
- void autko ()
- void mainScreen ()
- void smartMirrorScreen ()
- void opctionsScreen ()
- void spinner (int16_t x, int16_t y, uint16_t style, uint16_t scale)
- void button (int16_t x, int16_t y, int16_t w, int16_t h, int16_t font, uint16_t options, const char ∗str)
- void text (int16_t x, int16_t y, int16_t font, uint16_t options, const char ∗str)
- void number (int16_t x, int16_t y, int16_t font, uint16_t options, int value)
- void line (unsigned long color, unsigned long line_x1, unsigned long line_y1, unsigned long line_x2, unsigned long line_y2, unsigned long width)
- void dot (unsigned long color, unsigned int point_size, unsigned long point_x, unsigned long point_y)
- void calibrate ()
- void start (unsigned long color)
- void show ()

4.3.1 Detailed Description

File containing declarations of all API functions for VM800.

Author

Daniel Sienkiewicz

Date

28 February 2016

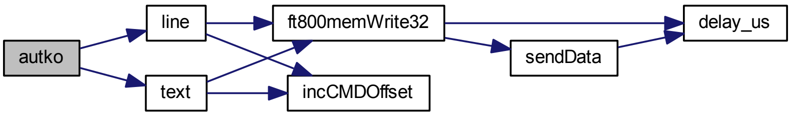
4.3.2 Function Documentation

4.3.2.1 void autko ( )

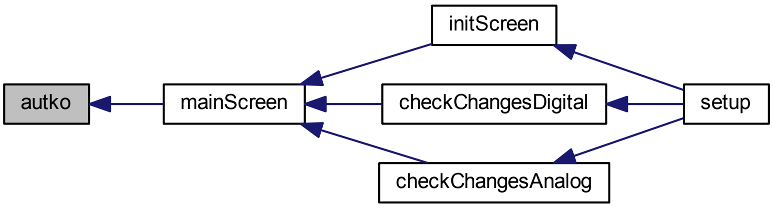
Function drawing car with proper attributes ∗

Definition at line 18 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.2.2 void button (int16_t *x*, int16_t *y*, int16_t *w*, int16_t *h*, int16_t *font*, uint16_t *options*, const char ∗ *str*)

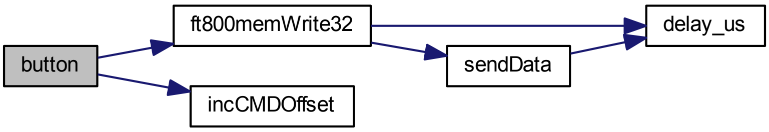
Function which draw a button on the screen ∗

Parameters

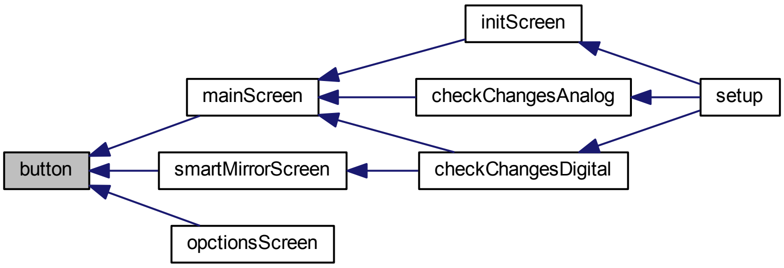
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
w	width for the button ∗
h	height for the button ∗
font	font fort the button text ∗
options	options for the button ∗
str	text to draw inside button ∗

Definition at line 127 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

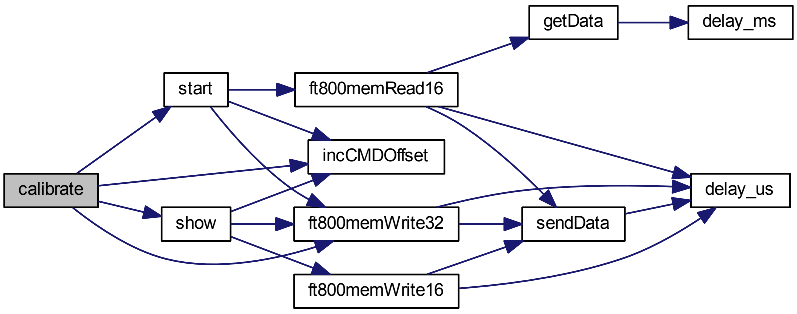


4.3.2.3 void calibrate ( )

Function which calibrate screen ∗

Definition at line 248 of file FT800api.cpp.

Here is the call graph for this function:



4.3.2.4 void dot (unsigned long *color,* unsigned int *point_size,* unsigned long *point_x,* unsigned long *point_y*)

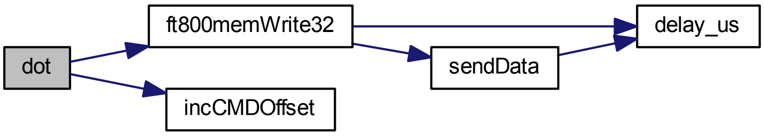
Function which draw a dot on the screen ∗

Parameters

color	dot color ∗
point_size	size for the dot ∗
point_x	x-coordinate for the dot ∗
point_y	y-coordinate for the dot ∗

Definition at line 230 of file FT800api.cpp.

Here is the call graph for this function:

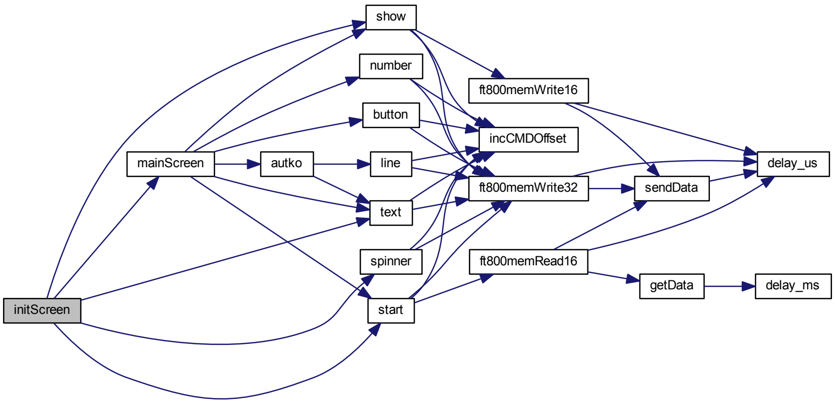


4.3.2.5 void initScreen ( )

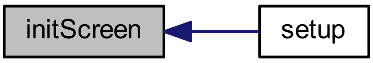
Function showing init screen durig main screen is loading ∗

Definition at line 10 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.2.6 void line (unsigned long *color,* unsigned long *line_x1,* unsigned long *line_y1,* unsigned long *line_x2,* unsigned long *line_y2,* unsigned long *width*)

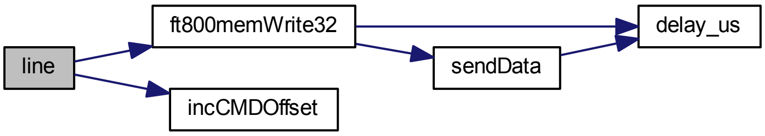
Function which draw a line on the screen ∗

Parameters

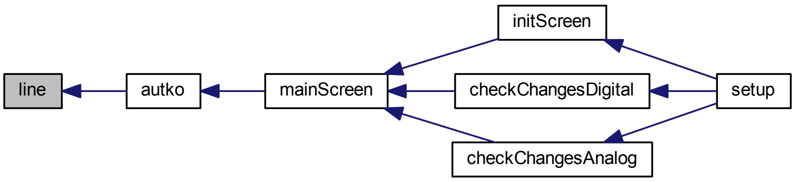
color	line color ∗
line_x1	x-coordinate for the beginning of the first end ∗
line_y1	y-coordinate for the beginning of the first end ∗
line_x2	x-coordinate for the beginning of the second end ∗
line_y2	y-coordinate for the beginning of the second end ∗
width	line width ∗

Definition at line 209 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

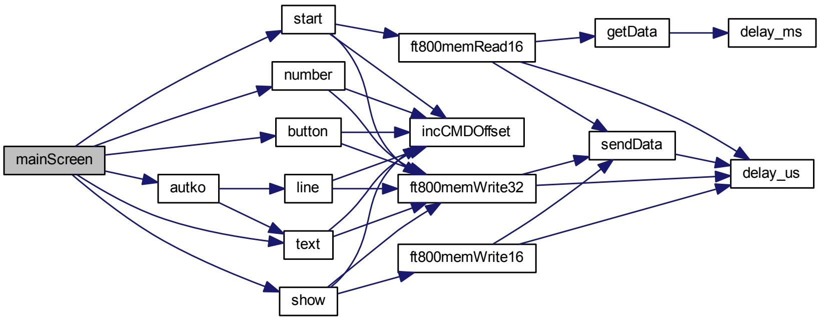


4.3.2.7 void mainScreen ( )

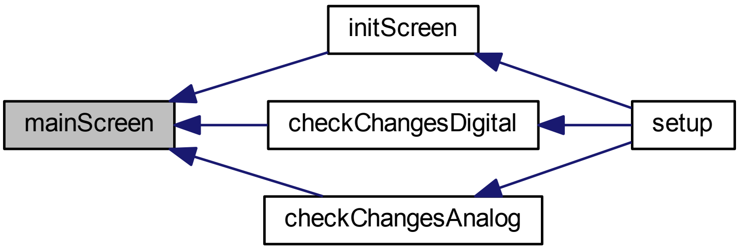
Function showing main screen ∗

Definition at line 49 of file FT800api.cpp.

Here is the call graph for this function:



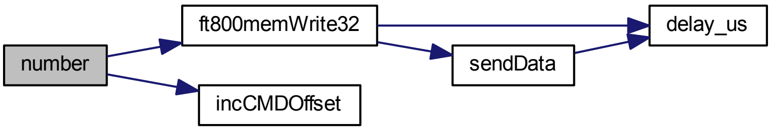
Here is the caller graph for this function:



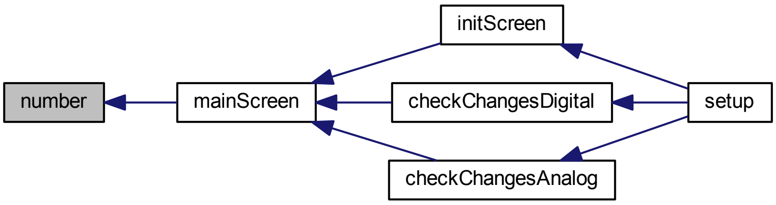
4.3.2.8 void number (int16_t *x,* int16_t *y,* int16_t *font,* uint16_t *options,* int *value*)

Definition at line 194 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

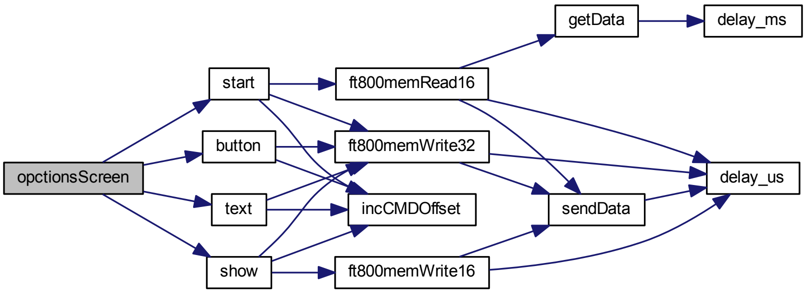


4.3.2.9 void opctionsScreen ( )

Function showing options screen ∗

Definition at line 77 of file FT800api.cpp.

Here is the call graph for this function:

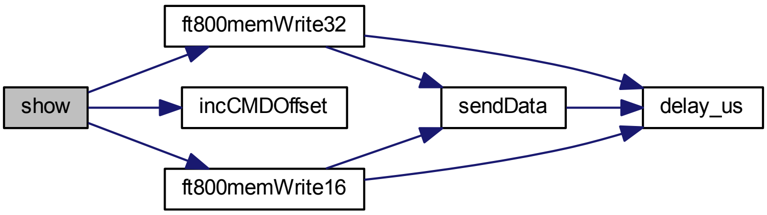


4.3.2.10 void show ( )

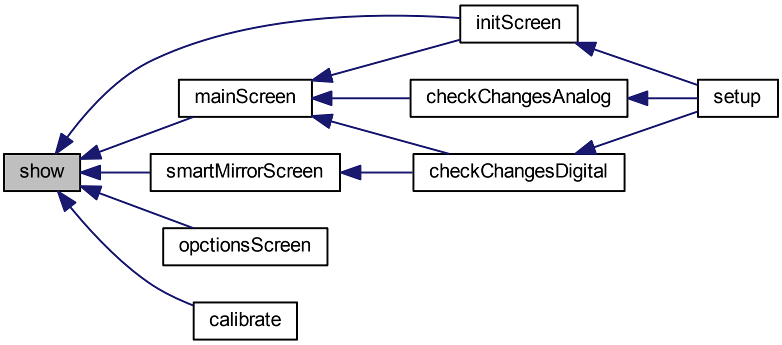
Function showing prerared screen from buffor ∗

Definition at line 280 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

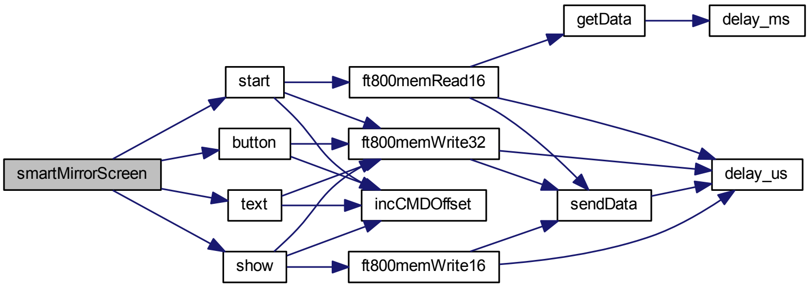


4.3.2.11 void smartMirrorScreen ( )

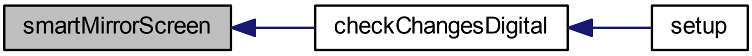
Function showing smart mirror screen ∗

Definition at line 68 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.2.12 void spinner (int16_t *x*, int16_t *y*, uint16_t *style*, uint16_t *scale*)

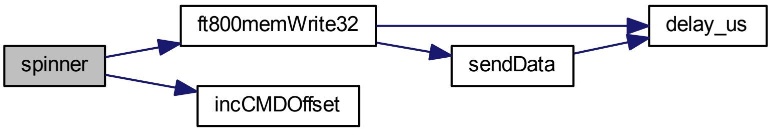
Function which draw a spinner on the screen ∗

Parameters

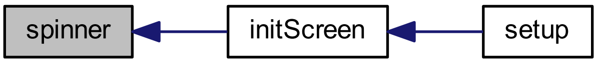
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
style	look swcreen ∗
scale	size of spinner ∗

Definition at line 116 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.2.13 void start (unsigned long *color*)

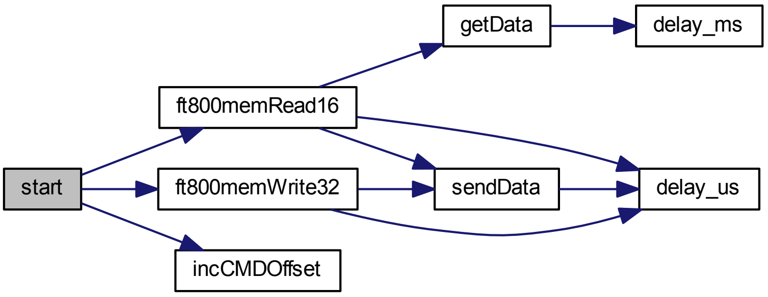
Function which start inicjalize new screen ∗

Parameters

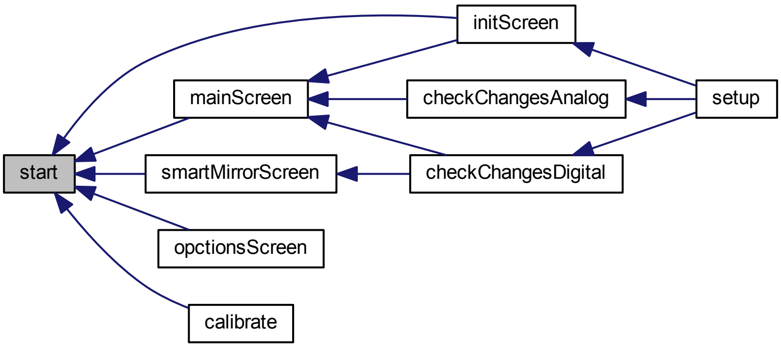
color	backgroud color ∗

Definition at line 259 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.2.14 void text (int16_t *x,* int16_t *y,* int16_t *font,* uint16_t *options,* const char ∗ *str*)

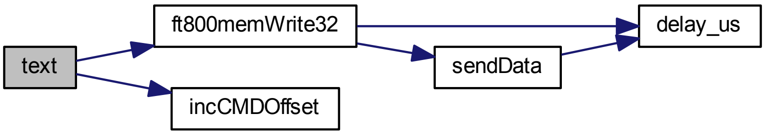
Function which draw a text on the screen ∗

Parameters

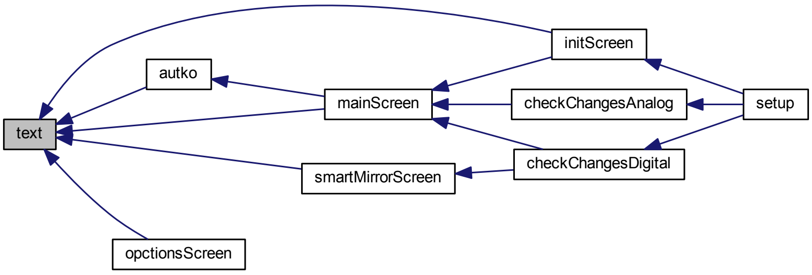
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
font	font for the text ∗
options	options to set for the text ∗
str	text to draw on the screen ∗

Definition at line 162 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

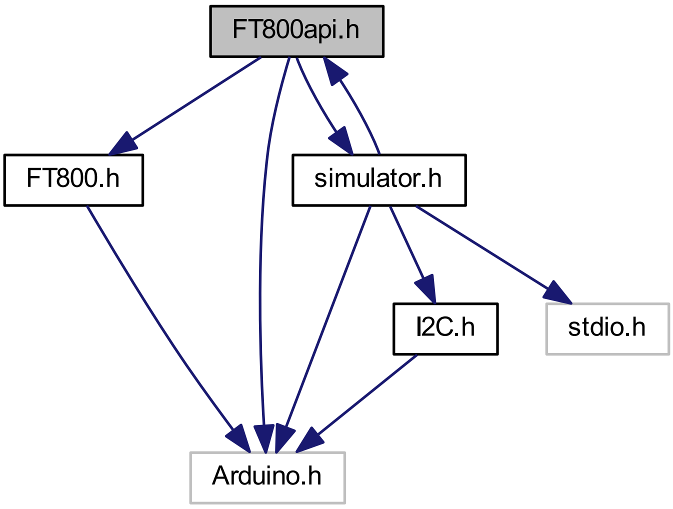


4.4 FT800api.h File Reference

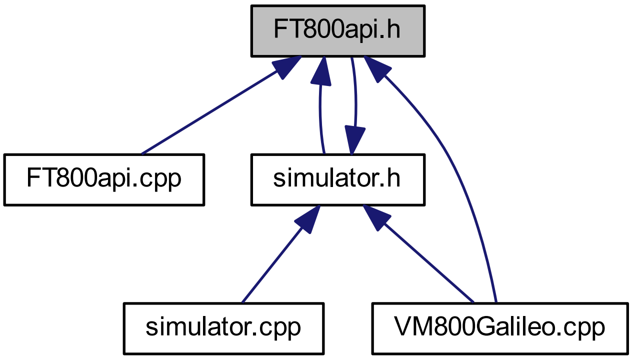
File containing declarations of all API functions for VM800.

```
#include "FT800.h"
#include "simulator.h"
#import <Arduino.h>
```

Include dependency graph for FT800api.h:



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



Functions

- void initScreen ()
- void opctionsScreen ()
- void mainScreen ()
- void smartMirrorScreen ()
- void spinner (int16_t x, int16_t y, uint16_t style, uint16_t scale)
- void button (int16_t x, int16_t y, int16_t w, int16_t h, int16_t font, uint16_t options, const char ∗str)

- void text (int16_t x, int16_t y, int16_t font, uint16_t options, const char ∗str)
- void line (unsigned long color, unsigned long line_x1, unsigned long line_y1, unsigned long line_x2, unsigned long line_y2, unsigned long width)
- void dot (unsigned long color, unsigned int point_size, unsigned long point_x, unsigned long point_y)
- void start (unsigned long color)
- void number (int16_t x, int16_t y, int16_t font, uint16_t options, int32_t value)
- void show ()
- void calibrate ()
- void autko ()

Variables

- unsigned int cmdOffset
- unsigned int cmdBufferRd
- unsigned int cmdBufferWr
- struct car ∗ audi
- int timeR

4.4.1 Detailed Description

File containing declarations of all API functions for VM800.

Author

Daniel Sienkiewicz

Date

28 February 2016

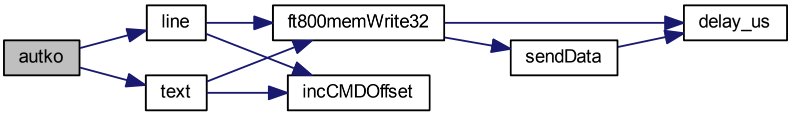
4.4.2 Function Documentation

4.4.2.1 void autko ( )

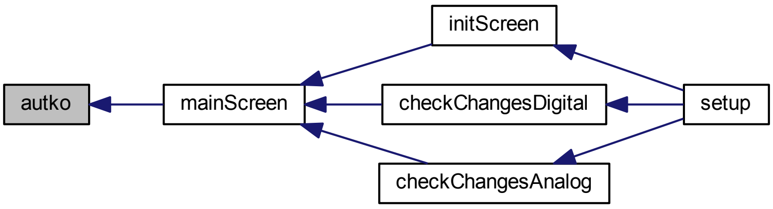
Function drawing car with proper attributes ∗

Definition at line 18 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.2 void button (int16_t *x,* int16_t *y,* int16_t *w,* int16_t *h,* int16_t *font,* uint16_t *options,* const char ∗ *str*)

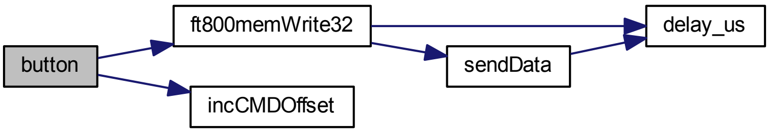
Function which draw a button on the screen ∗

Parameters

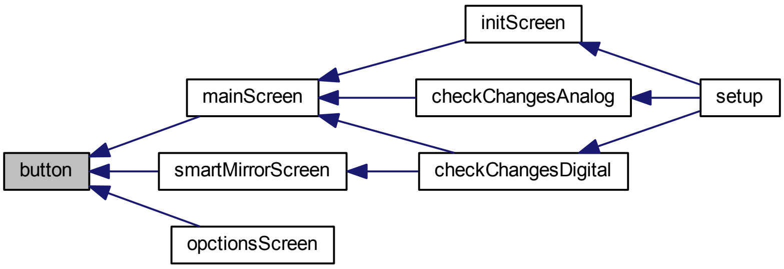
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
w	width for the button ∗
h	height for the button ∗
font	font fort the button text ∗
options	options for the button ∗
str	text to draw inside button ∗

Definition at line 127 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

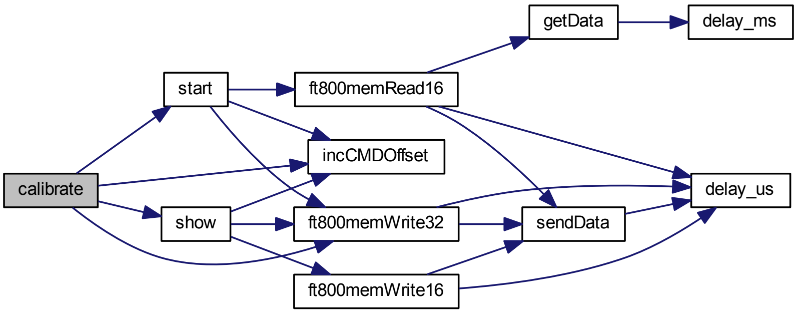


4.4.2.3 void calibrate ( )

Function which calibrate screen ∗

Definition at line 248 of file FT800api.cpp.

Here is the call graph for this function:



4.4.2.4 void dot (unsigned long *color,* unsigned int *point_size,* unsigned long *point_x,* unsigned long *point_y*)

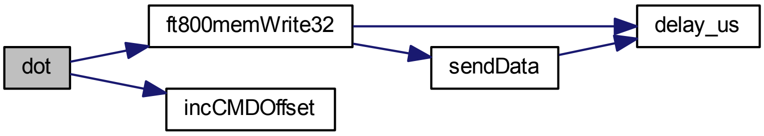
Function which draw a dot on the screen ∗

Parameters

color	dot color ∗
point_size	size for the dot ∗
point_x	x-coordinate for the dot ∗
point_y	y-coordinate for the dot ∗

Definition at line 230 of file FT800api.cpp.

Here is the call graph for this function:

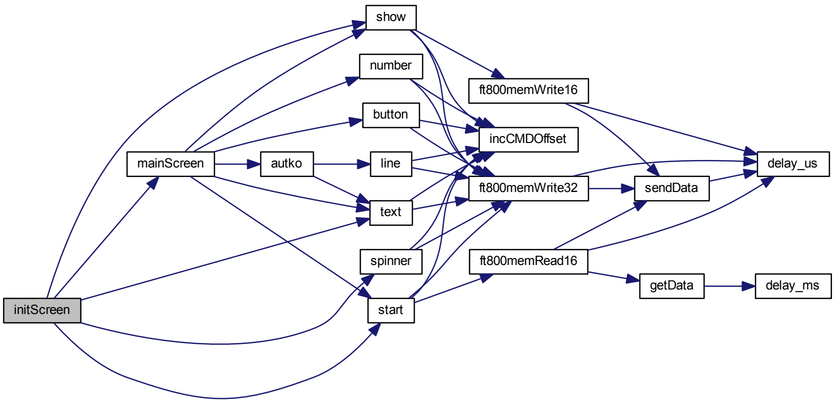


4.4.2.5 void initScreen ( )

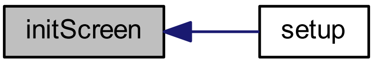
Function showing init screen durig main screen is loading ∗

Definition at line 10 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.6 void line (unsigned long *color,* unsigned long *line_x1,* unsigned long *line_y1,* unsigned long *line_x2,* unsigned long *line_y2,* unsigned long *width*)

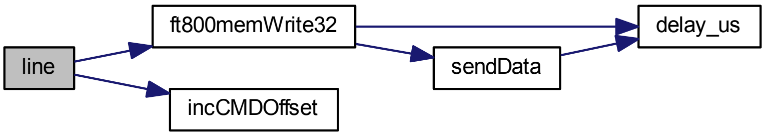
Function which draw a line on the screen ∗

Parameters

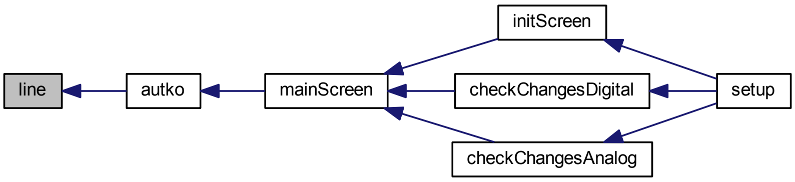
color	line color ∗
line_x1	x-coordinate for the beginning of the first end ∗
line_y1	y-coordinate for the beginning of the first end ∗
line_x2	x-coordinate for the beginning of the second end ∗
line_y2	y-coordinate for the beginning of the second end ∗
width	line width ∗

Definition at line 209 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

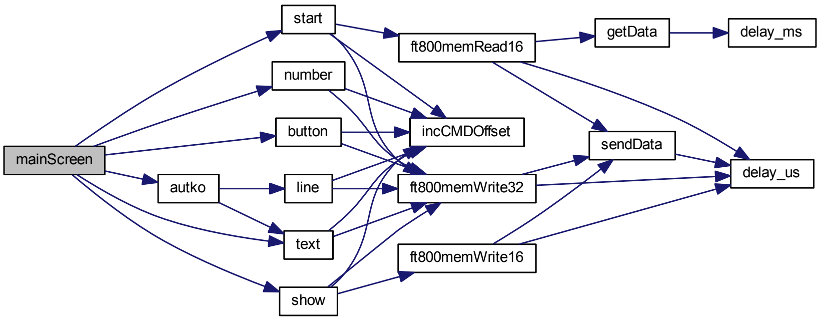


4.4.2.7 void mainScreen ( )

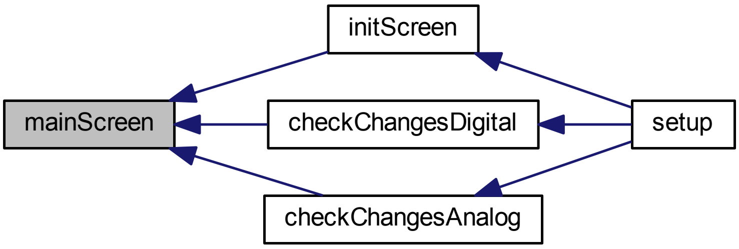
Function showing main screen ∗

Definition at line 49 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.8 void number (int16_t *x,* int16_t *y,* int16_t *font,* uint16_t *options,* int32_t *value*)

Function which draw a number on the screen ∗

Parameters

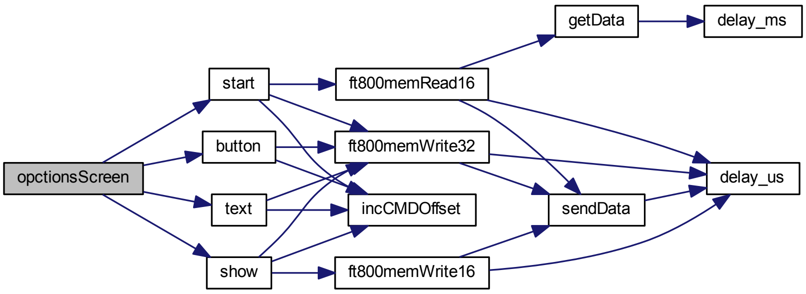
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
font	font for the number ∗
options	options to set for the number ∗
value	value to draw on the screen ∗

4.4.2.9 void opctionsScreen ( )

Function showing options screen ∗

Definition at line 77 of file FT800api.cpp.

Here is the call graph for this function:

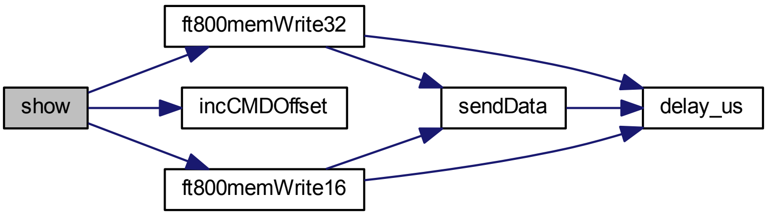


4.4.2.10 void show ( )

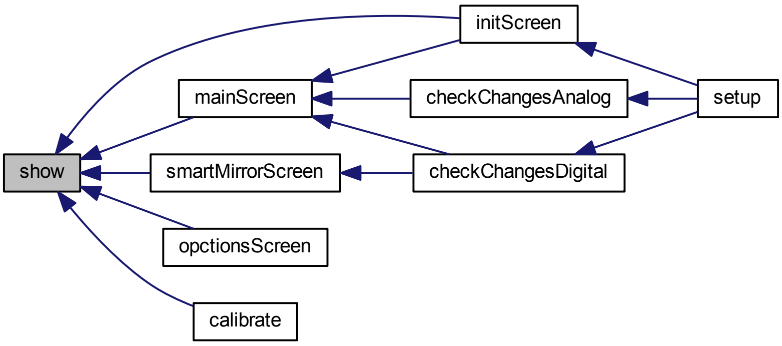
Function showing prerared screen from buffor ∗

Definition at line 280 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

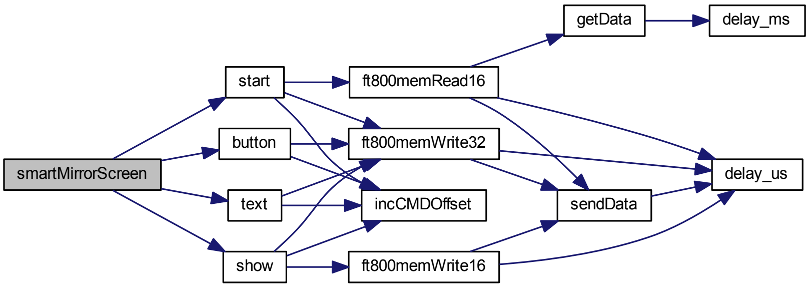


4.4.2.11 void smartMirrorScreen ( )

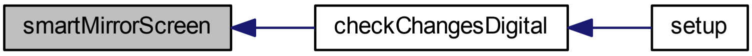
Function showing smart mirror screen ∗

Definition at line 68 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.12 void spinner (int16_t *x*, int16_t *y*, uint16_t *style*, uint16_t *scale*)

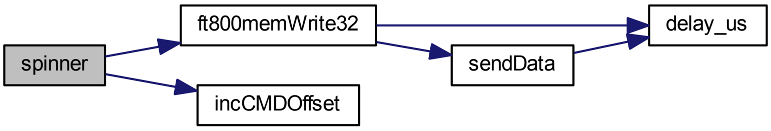
Function which draw a spinner on the screen ∗

Parameters

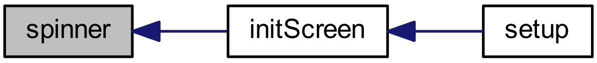
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
style	look swcreen ∗
scale	size of spinner ∗

Definition at line 116 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.13 void start (unsigned long *color*)

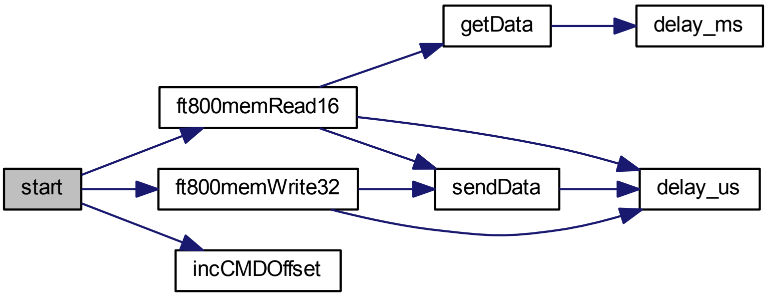
Function which start inicjalize new screen ∗

Parameters

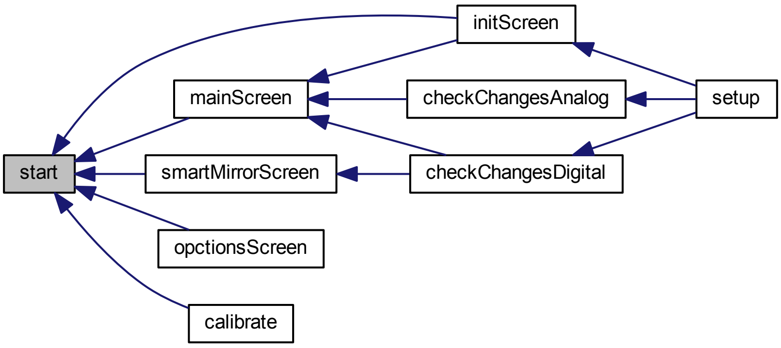
color	backgroud color ∗

Definition at line 259 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.14 void text (int16_t *x,* int16_t *y,* int16_t *font,* uint16_t *options,* const char ∗ *str*)

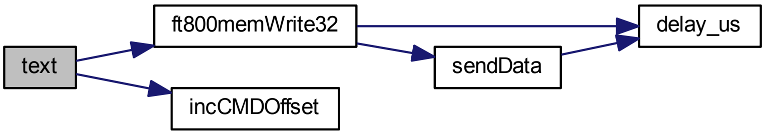
Function which draw a text on the screen ∗

Parameters

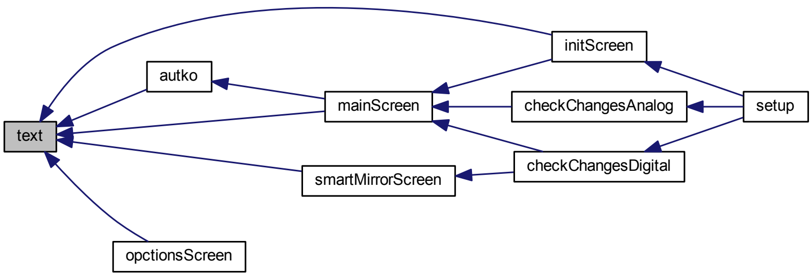
x	x-coordinate on the screen ∗
y	y-coordinate on the screen ∗
font	font for the text ∗
options	options to set for the text ∗
str	text to draw on the screen ∗

Definition at line 162 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.3 Variable Documentation

4.4.3.1 struct car∗ audi

Main car structure with data from sensors

Definition at line 41 of file VM800Galileo.cpp.

4.4.3.2 unsigned int cmdBufferRd

Used to navigate command ring buffer

Definition at line 35 of file VM800Galileo.cpp.

4.4.3.3 unsigned int cmdBufferWr

Used to navigate command ring buffer

Definition at line 36 of file VM800Galileo.cpp.

4.4.3.4 unsigned int cmdOffset

Used to navigate command rung buffer

Definition at line 37 of file VM800Galileo.cpp.

4.4.3.5 int timeR

Data refresh time to save to file

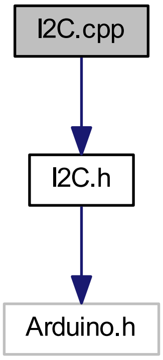
Definition at line 44 of file VM800Galileo.cpp.

4.5 I2C.cpp File Reference

File containing declarations of function to read data with using I2C protocol.

```
#include "I2C.h"
```

Include dependency graph for I2C.cpp:



Functions

- int readPCF (char adres)

Variables

- int d = 1

4.5.1 Detailed Description

File containing declarations of function to read data with using I2C protocol.

Author

Daniel Sienkiewicz

Date

28 February 2016

4.5.2 Function Documentation

4.5.2.1 int readPCF (char *adres*)

Reading value from PCF8574N I/O Expander ∗

Parameters

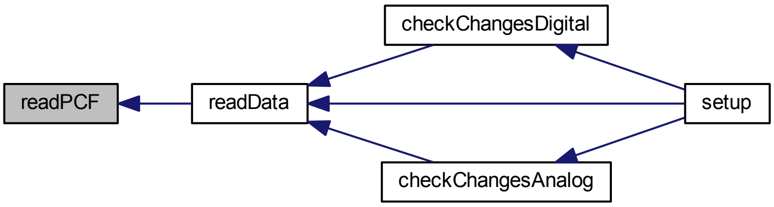
adres	The address of PCF8574N I/O Expander ∗

Returns

Value from the specified PCF8574N I/O Expander ∗

Definition at line 12 of file I2C.cpp.

Here is the caller graph for this function:



4.5.3 Variable Documentation

4.5.3.1 int d = 1

Delay time - for PCF handing

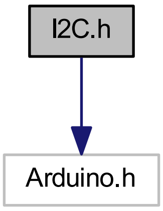
Definition at line 10 of file I2C.cpp.

4.6 I2C.h File Reference

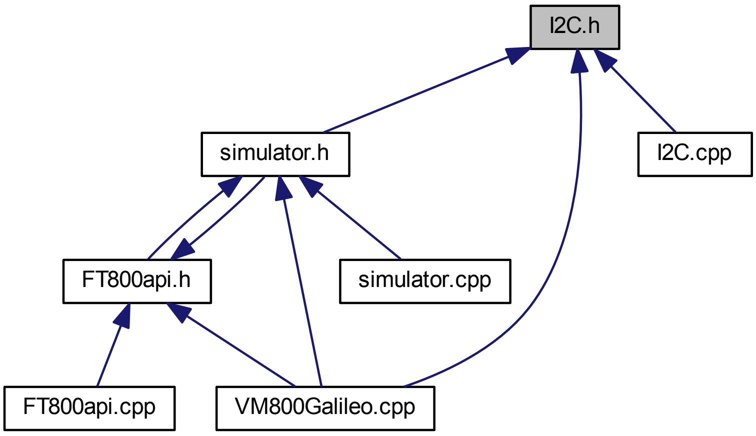
File containing declarations of function to read data with using I2C protocol.

```
#import <Arduino.h>
```

Include dependency graph for I2C.h:



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



Macros

- #define sda 7
- #define scl 6
- #define pinInt0 2

Functions

- int readPCF (char adres)

4.6.1 Detailed Description

File containing declarations of function to read data with using I2C protocol.

Author

Daniel Sienkiewicz

Date

28 February 2016

4.6.2 Macro Definition Documentation

4.6.2.1 #define pinInt0 2

Interrput port number

Definition at line 14 of file I2C.h.

4.6.2.2 #define scl 6

SCL port number

Definition at line 13 of file I2C.h.

4.6.2.3 #define sda 7

SDA port number

Definition at line 12 of file I2C.h.

4.6.3 Function Documentation

4.6.3.1 int readPCF (char *adres*)

Reading value from PCF8574N I/O Expander ∗

Parameters

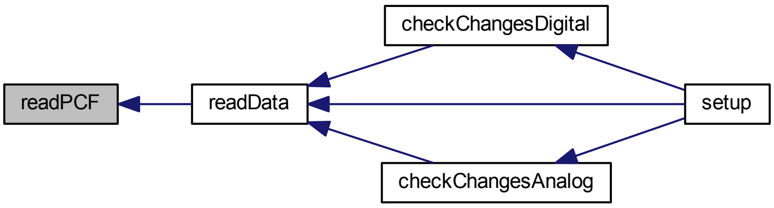
adres	The address of PCF8574N I/O Expander ∗

Returns

Value from the specified PCF8574N I/O Expander ∗

Definition at line 12 of file I2C.cpp.

Here is the caller graph for this function:

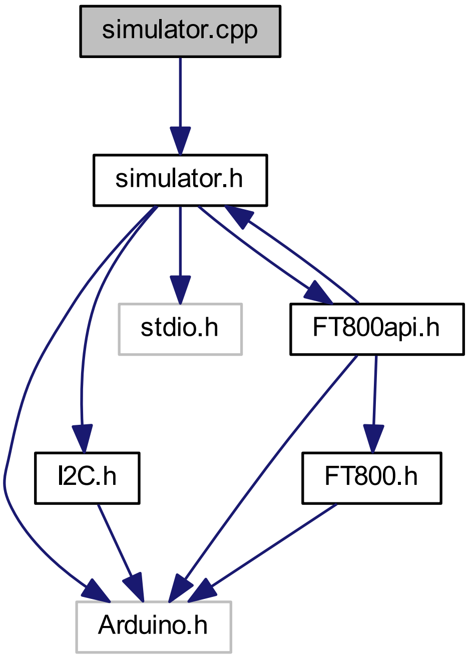


4.7 simulator.cpp File Reference

File containing declarations of all functions required to communication with car simulator.

```
#include "simulator.h"
```

Include dependency graph for simulator.cpp:



Functions

- void printObj (struct car ∗obj, char ∗d)
- int readTemp (int portNumber)
- void save (struct car ∗audi, struct car ∗tmp)
- struct car ∗ readData ()
- void checkChangesDigital ()
- void sendData ()
- void checkChangesAnalog ()

4.7.1 Detailed Description

File containing declarations of all functions required to communication with car simulator.

Author

Daniel Sienkiewicz

Date

28 February 2016

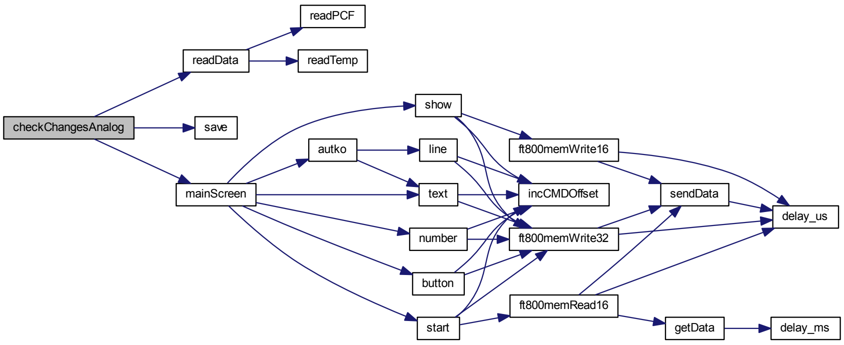
4.7.2 Function Documentation

4.7.2.1 void checkChangesAnalog ( )

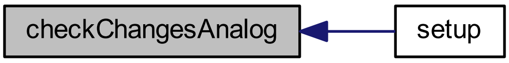
Check if sth on analog ports was changed ∗

Definition at line 138 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

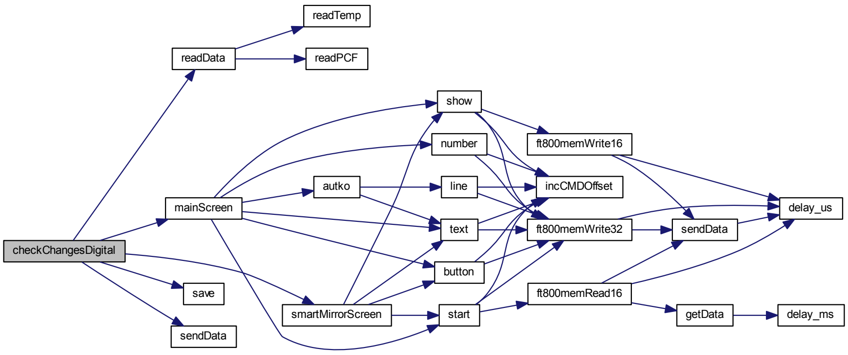


4.7.2.2 void checkChangesDigital ( )

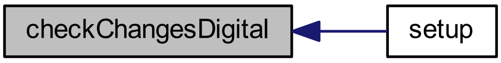
Check if sth on digital ports was changed ∗

Definition at line 96 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.3 void printObj (struct car ∗ *obj,* char ∗ *d*)

Debug function to print car structure on a serial monitor ∗ console and to log file on SD car ∗

Parameters

Car	struct to print and save with selected format into file∗
d	actual date ∗

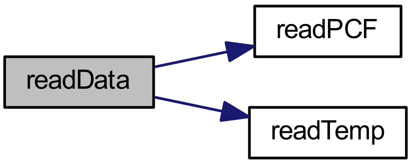
Definition at line 10 of file simulator.cpp.

4.7.2.4 struct car∗ readData ( )

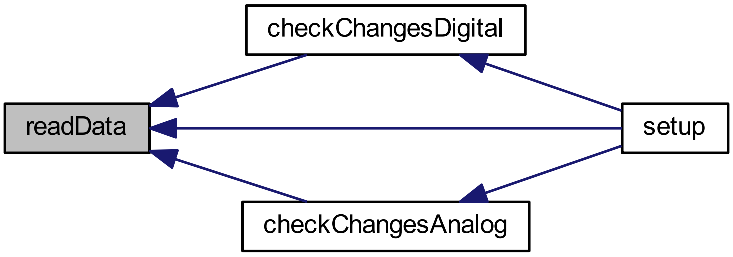
Reading data about car status ∗

Definition at line 77 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.5 int readTemp (int *portNumber*)

Reading value from analog ports (temperatures) $*$

Parameters

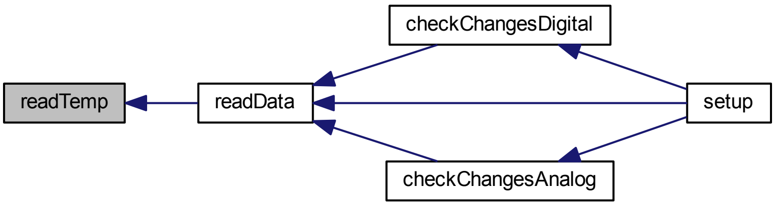
portNumber	The number of the analog input pin to read $*$

Returns

Value from the specified analog pin $*$

Definition at line 63 of file simulator.cpp.

Here is the caller graph for this function:



4.7.2.6 void save (struct car ∗ *audi,* struct car ∗ *tmp*)

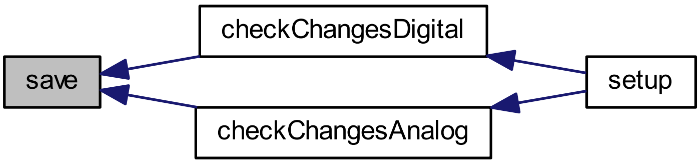
Copying data function from temporary to main struct ∗

Parameters

∗*audi,*∗*tmp*	Structures to and from which data are copied ∗

Definition at line 67 of file simulator.cpp.

Here is the caller graph for this function:

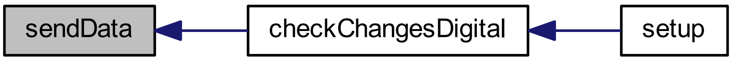


4.7.2.7 void sendData ( )

Sending actial data to web server ∗

Definition at line 132 of file simulator.cpp.

Here is the caller graph for this function:

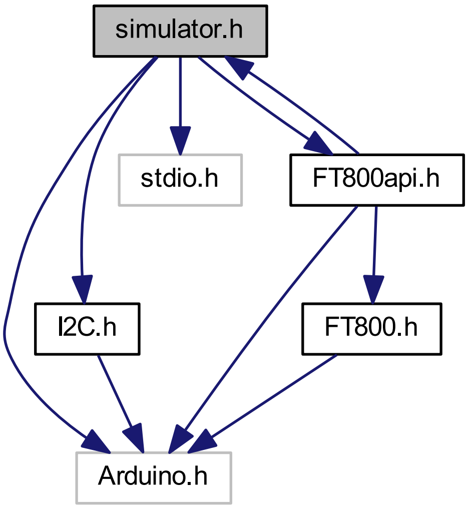


4.8 simulator.h File Reference

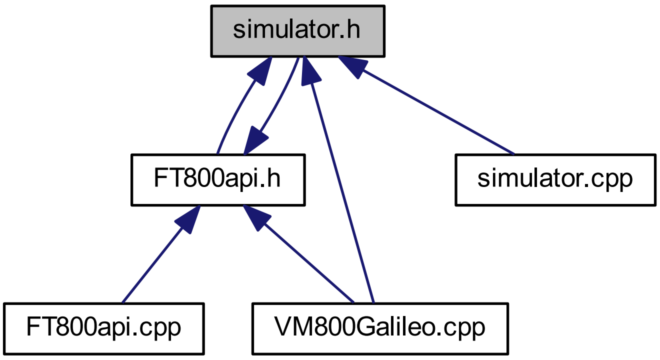
File containing declarations of all functions required to communication with car simulator.

```
#import <Arduino.h>
#include "I2C.h"
#include <stdio.h>
#include "FT800api.h"
```

Include dependency graph for simulator.h:



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



Classes

- struct car

Functions

- void printObj (struct car ∗obj, char ∗d)
- void checkChangesAnalog ()
- void checkChangesDigital ()
- struct car ∗ readData ()
- void save (struct car ∗audi, struct car ∗tmp)
- int readTemp (int portNumber)
- void sendData ()

Variables

- struct car ∗ audi
- int dataFormat
- int saveData
- short int screenNR

4.8.1 Detailed Description

File containing declarations of all functions required to communication with car simulator.

Author

Daniel Sienkiewicz

Date

28 February 2016

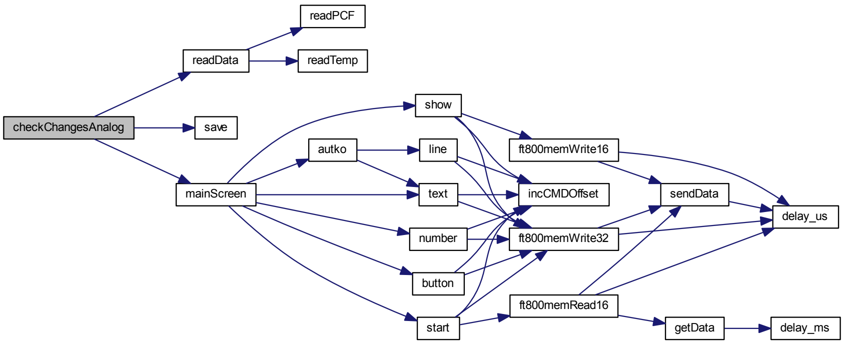
4.8.2 Function Documentation

4.8.2.1 void checkChangesAnalog ( )

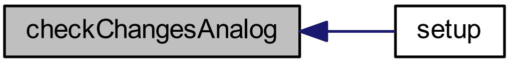
Check if sth on analog ports was changed ∗

Definition at line 138 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

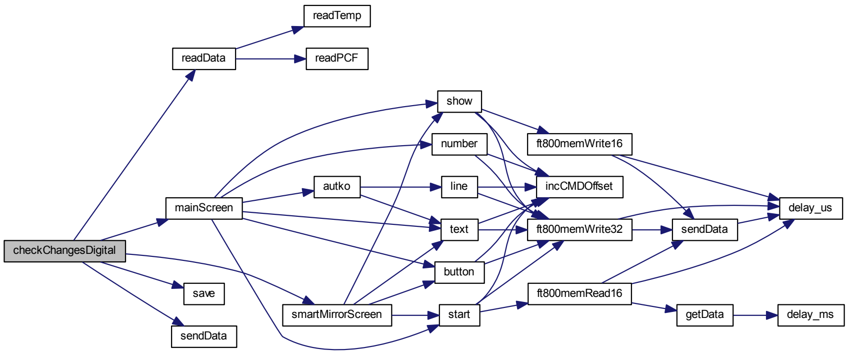


4.8.2.2 void checkChangesDigital ( )

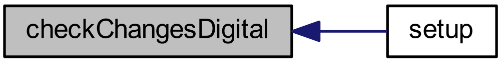
Check if sth on digital ports was changed ∗

Definition at line 96 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.3 void printObj (struct car ∗ *obj*, char ∗ *d*)

Debug function to print car structure on a serial monitor ∗ console and to log file on SD car ∗

Parameters

Car	struct to print and save with selected format into file∗
d	actual date ∗

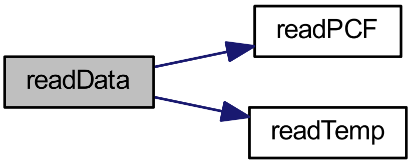
Definition at line 10 of file simulator.cpp.

4.8.2.4 struct car∗ readData ( )

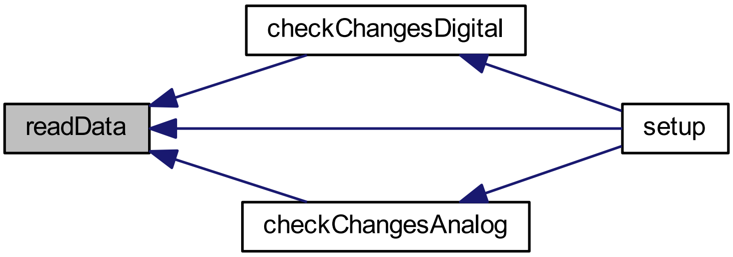
Reading data about car status ∗

Definition at line 77 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.2.5 int readTemp (int *portNumber*)

Reading value from analog ports (temperatures) ∗

Parameters

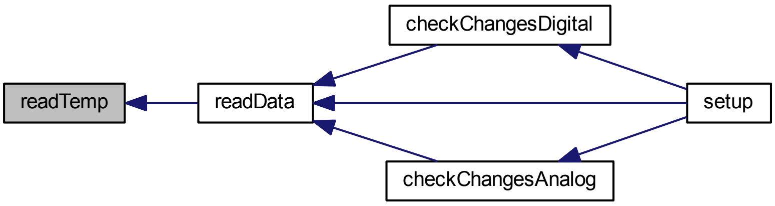
portNumber	The number of the analog input pin to read ∗

Returns

Value from the specified analog pin ∗

Definition at line 63 of file simulator.cpp.

Here is the caller graph for this function:



4.8.2.6 void save (struct car ∗ *audi,* struct car ∗ *tmp*)

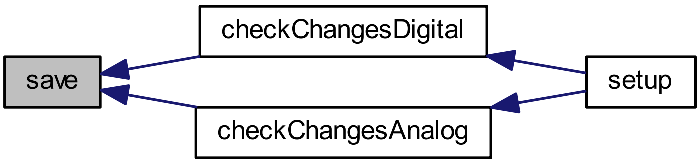
Copying data function from temporary to main struct ∗

Parameters

∗*audi,*∗*tmp*	Structures to and from which data are copied ∗

Definition at line 67 of file simulator.cpp.

Here is the caller graph for this function:

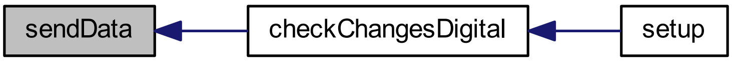


4.8.2.7 void sendData ( )

Sending actial data to web server ∗

Definition at line 132 of file simulator.cpp.

Here is the caller graph for this function:



4.8.3 Variable Documentation

4.8.3.1 struct car∗ audi

Main car structure with data from sensors

Definition at line 41 of file VM800Galileo.cpp.

4.8.3.2 int dataFormat

Selected format to save in file 1 - CSV, 2 - XML, 3 - JSON

Definition at line 42 of file VM800Galileo.cpp.

4.8.3.3 int saveData

If data will be saving on SD card

Definition at line 43 of file VM800Galileo.cpp.

4.8.3.4 short int screenNR

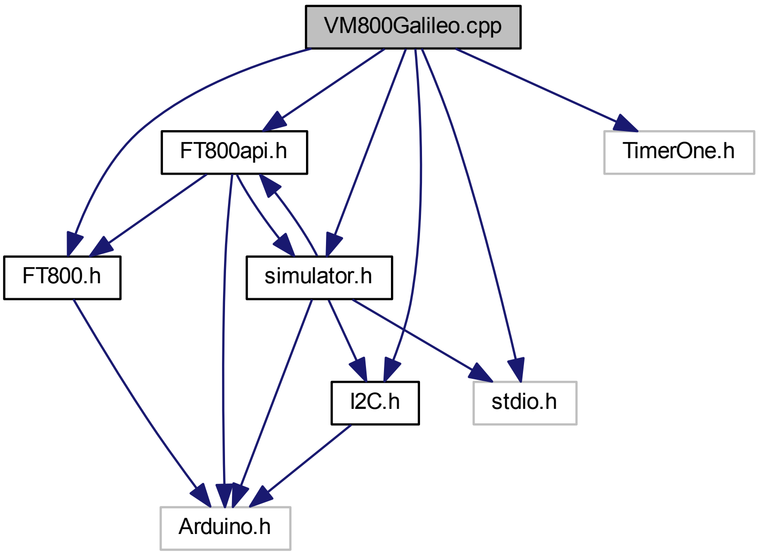
Selected screen - 1 main screen, 2 - smart mirror, 3 - options

Definition at line 40 of file VM800Galileo.cpp.

4.9 VM800Galileo.cpp File Reference

```
#include "FT800.h"
#include "FT800api.h"
#include "I2C.h"
#include "simulator.h"
#include <stdio.h>
#include <TimerOne.h>
```

Include dependency graph for VM800Galileo.cpp:



Functions

- void setup (void)
- void loop ()

Variables

- unsigned int lcdWidth
- unsigned int lcdHeight
- unsigned int lcdHcycle
- unsigned int lcdHoffset
- unsigned int lcdHsync0
- unsigned int lcdHsync1
- unsigned int lcdVcycle
- unsigned int lcdVoffset
- unsigned int lcdVsync0
- unsigned int lcdVsync1
- unsigned char lcdPclk
- unsigned char lcdSwizzle

- unsigned char lcdPclkpol
- unsigned long ramDisplayList =RAM_DL
- unsigned long ramCommandBuffer =RAM_CMD
- unsigned int cmdBufferRd
- unsigned int cmdBufferWr =0x0000
- unsigned int cmdOffset =0x0000
- unsigned char ft800Gpio
- short int screenNR = 1
- struct car ∗ audi
- int dataFormat = 3
- int saveData = 0
- int timeR = 1
- char buf [9]

4.9.1 Function Documentation

4.9.1.1 void loop ( )

function executed in infinity loop after finished executing setup function ∗

Definition at line 230 of file VM800Galileo.cpp.

4.9.1.2 void setup (void)

setup function for Intel Galileo executed only once with first start of program ∗ < QVGA display parameters

< Active width of LCD display

< Active height of LCD display

< Total number of clocks per line

< Start of active line

< Start of horizontal sync pulse

< End of horizontal sync pulse

< Total number of lines per screen

< Start of active screen

< Start of vertical sync pulse

< End of vertical sync pulse

< Pixel Clock

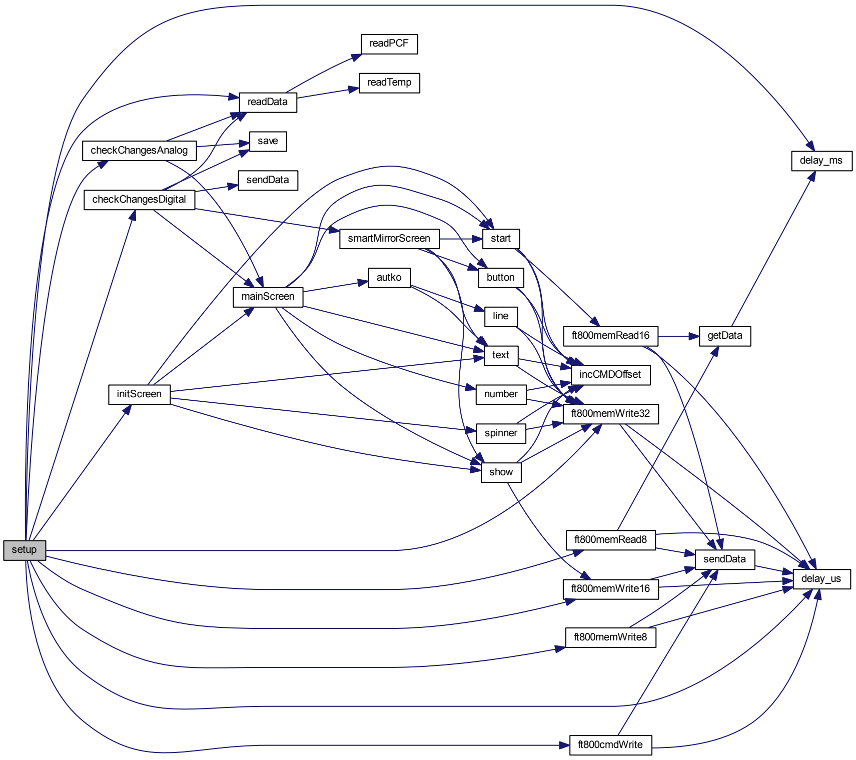
< Define RGB output pins

< Define active edge of PCLK

< WQVGA display parameters

Definition at line 51 of file VM800Galileo.cpp.

Here is the call graph for this function:



4.9.2 Variable Documentation

4.9.2.1 struct car∗ audi

Main car structure with data from sensors

Definition at line 41 of file VM800Galileo.cpp.

4.9.2.2 char buf[9]

Actual date

Definition at line 45 of file VM800Galileo.cpp.

4.9.2.3 unsigned int cmdBufferRd

Used to navigate command ring buffer

Definition at line 35 of file VM800Galileo.cpp.

4.9.2.4 unsigned int cmdBufferWr =0x0000

Used to navigate command ring buffer

Definition at line 36 of file VM800Galileo.cpp.

4.9.2.5 unsigned int cmdOffset =0x0000

Used to navigate command rung buffer

Definition at line 37 of file VM800Galileo.cpp.

4.9.2.6 int dataFormat = 3

Selected format to save in file 1 - CSV, 2 - XML, 3 - JSON

Definition at line 42 of file VM800Galileo.cpp.

4.9.2.7 unsigned char ft800Gpio

Used for FT800 GPIO register

Definition at line 38 of file VM800Galileo.cpp.

4.9.2.8 unsigned int lcdHcycle

Total number of clocks per line

Definition at line 20 of file VM800Galileo.cpp.

4.9.2.9 unsigned int lcdHeight

Active height of LCD display

Definition at line 19 of file VM800Galileo.cpp.

4.9.2.10 unsigned int lcdHoffset

Start of active line

Definition at line 21 of file VM800Galileo.cpp.

4.9.2.11 unsigned int lcdHsync0

Start of horizontal sync pulse

Definition at line 22 of file VM800Galileo.cpp.

4.9.2.12 unsigned int lcdHsync1

End of horizontal sync pulse

Definition at line 23 of file VM800Galileo.cpp.

4.9.2.13 unsigned char lcdPclk

Pixel Clock

Definition at line 28 of file VM800Galileo.cpp.

4.9.2.14 unsigned char lcdPclkpol

Define active edge of PCLK

Definition at line 30 of file VM800Galileo.cpp.

4.9.2.15 unsigned char lcdSwizzle

Define RGB output pins

Definition at line 29 of file VM800Galileo.cpp.

4.9.2.16 unsigned int lcdVcycle

Total number of lines per screen

Definition at line 24 of file VM800Galileo.cpp.

4.9.2.17 unsigned int lcdVoffset

Start of active screen

Definition at line 25 of file VM800Galileo.cpp.

4.9.2.18 unsigned int lcdVsync0

Start of vertical sync pulse

Definition at line 26 of file VM800Galileo.cpp.

4.9.2.19 unsigned int lcdVsync1

End of vertical sync pulse

Definition at line 27 of file VM800Galileo.cpp.

4.9.2.20 unsigned int lcdWidth

Active width of LCD display

Definition at line 18 of file VM800Galileo.cpp.

4.9.2.21 unsigned long ramCommandBuffer =RAM_CMD

Set beginning of graphics command memory

Definition at line 33 of file VM800Galileo.cpp.

4.9.2.22 unsigned long ramDisplayList =RAM_DL

Set beginning of display list memory

Definition at line 32 of file VM800Galileo.cpp.

4.9.2.23 int saveData = 0

If data will be saving on SD card

Definition at line 43 of file VM800Galileo.cpp.

4.9.2.24 short int screenNR = 1

Selected screen - 1 main screen, 2 - smart mirror, 3 - options

Definition at line 40 of file VM800Galileo.cpp.

4.9.2.25 int timeR = 1

Data refresh time to save to file

Definition at line 44 of file VM800Galileo.cpp.

Index