

Code documentation

1

Generated by Doxygen 1.8.11

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	car Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	doors	5
3.1.2.2	lights	6
3.1.2.3	r	6
3.1.2.4	seatbelts	6
3.1.2.5	tempEngine	6
3.1.2.6	templn	6
3.1.2.7	tempOut	6

4 File Documentation	7
4.1 FT800.cpp File Reference	7
4.1.1 Detailed Description	8
4.1.2 Function Documentation	8
4.1.2.1 delay_ms(int ms)	8
4.1.2.2 delay_us(int us)	8
4.1.2.3 ft800cmdWrite(unsigned char ftCommand)	9
4.1.2.4 ft800memRead16(unsigned long ftAddress)	9
4.1.2.5 ft800memRead32(unsigned long ftAddress)	10
4.1.2.6 ft800memRead8(unsigned long ftAddress)	11
4.1.2.7 ft800memWrite16(unsigned long ftAddress, unsigned int ftData16)	12
4.1.2.8 ft800memWrite32(unsigned long ftAddress, unsigned long ftData32)	12
4.1.2.9 ft800memWrite8(unsigned long ftAddress, unsigned char ftData8)	13
4.1.2.10 getData()	14
4.1.2.11 incCMDOffset(unsigned int currentOffset, unsigned char commandSize)	14
4.1.2.12 sendData(int data)	15
4.2 FT800.h File Reference	16
4.2.1 Detailed Description	23
4.2.2 Macro Definition Documentation	23
4.2.2.1 ABS	23
4.2.2.2 BLACK	23
4.2.2.3 BLUE	23
4.2.2.4 CLR_COL	23
4.2.2.5 CLR_STN	23
4.2.2.6 CLR_TAG	23
4.2.2.7 CMD_APPEND	23
4.2.2.8 CMD_BGCOLOR	24
4.2.2.9 CMD_BUTTON	24
4.2.2.10 CMD_CALIBRATE	24
4.2.2.11 CMD_CLOCK	24

4.2.2.12	CMD_COLDSTART	24
4.2.2.13	CMD_DIAL	24
4.2.2.14	CMD_DLSTART	24
4.2.2.15	CMD_FGCOLOR	24
4.2.2.16	CMD_GAUGE	24
4.2.2.17	CMD_GETMATRIX	24
4.2.2.18	CMD_GETPTR	25
4.2.2.19	CMD_GRADCOLOR	25
4.2.2.20	CMD_GRADIENT	25
4.2.2.21	CMD_INFLATE	25
4.2.2.22	CMD_INTERRUPT	25
4.2.2.23	CMD_KEYS	25
4.2.2.24	CMD_LOADIDENTITY	25
4.2.2.25	CMD_LOADIMAGE	25
4.2.2.26	CMD_LOGO	25
4.2.2.27	CMD_MEMCPY	25
4.2.2.28	CMD_MEMCRC	26
4.2.2.29	CMD_MEMSET	26
4.2.2.30	CMD_MEMWRITE	26
4.2.2.31	CMD_MEMZERO	26
4.2.2.32	CMD_NUMBER	26
4.2.2.33	CMD_PROGRESS	26
4.2.2.34	CMD_REGREAD	26
4.2.2.35	CMD_ROTATE	26
4.2.2.36	CMD_SCALE	26
4.2.2.37	CMD_SCREENSAVER	26
4.2.2.38	CMD_SCROLLBAR	27
4.2.2.39	CMD_SETFONT	27
4.2.2.40	CMD_SETMATRIX	27
4.2.2.41	CMD_SKETCH	27

4.2.2.42	CMD_SLIDER	27
4.2.2.43	CMD_SNAPSHOT	27
4.2.2.44	CMD_SPINNER	27
4.2.2.45	CMD_STOP	27
4.2.2.46	CMD_SWAP	27
4.2.2.47	CMD_TEXT	27
4.2.2.48	CMD_TOGGLE	28
4.2.2.49	CMD_TRACK	28
4.2.2.50	CMD_TRANSLATE	28
4.2.2.51	CMDBUF_SIZE	28
4.2.2.52	DECR	28
4.2.2.53	DECR_WRAP	28
4.2.2.54	DL_ALPHA_FUNC	28
4.2.2.55	DL_BEGIN	28
4.2.2.56	DL_BITMAP_HANDLE	28
4.2.2.57	DL_BITMAP_LAYOUT	28
4.2.2.58	DL_BITMAP_SIZE	29
4.2.2.59	DL_BITMAP_SOURCE	29
4.2.2.60	DL_BITMAP_TFORM_A	29
4.2.2.61	DL_BITMAP_TFORM_B	29
4.2.2.62	DL_BITMAP_TFORM_C	29
4.2.2.63	DL_BITMAP_TFORM_D	29
4.2.2.64	DL_BITMAP_TFORM_E	29
4.2.2.65	DL_BITMAP_TFORM_F	29
4.2.2.66	DL_BLEND_FUNC	29
4.2.2.67	DL_CALL	29
4.2.2.68	DL_CELL	30
4.2.2.69	DL_CLEAR	30
4.2.2.70	DL_CLEAR_RGB	30
4.2.2.71	DL_CLEAR_STENCIL	30

4.2.2.72	DL_CLEAR_TAG	30
4.2.2.73	DL_COLOR_A	30
4.2.2.74	DL_COLOR_MASK	30
4.2.2.75	DL_COLOR_RGB	30
4.2.2.76	DL_DISPLAY	30
4.2.2.77	DL_END	30
4.2.2.78	DL_JUMP	31
4.2.2.79	DL_LINE_WIDTH	31
4.2.2.80	DL_MACRO	31
4.2.2.81	DL_POINT_SIZE	31
4.2.2.82	DL_RESTORE_CONTEXT	31
4.2.2.83	DL_RETURN	31
4.2.2.84	DL_SAVE_CONTEXT	31
4.2.2.85	DL_SCISSOR_SIZE	31
4.2.2.86	DL_SCISSOR_XY	31
4.2.2.87	DL_STENCIL_FUNC	31
4.2.2.88	DL_STENCIL_MASK	32
4.2.2.89	DL_STENCIL_OP	32
4.2.2.90	DL_TAG	32
4.2.2.91	DL_TAG_MASK	32
4.2.2.92	DL_VERTEX2F	32
4.2.2.93	DL_VERTEX2I	32
4.2.2.94	DLSWAP_DONE	32
4.2.2.95	DLSWAP_FRAME	32
4.2.2.96	DLSWAP_LINE	32
4.2.2.97	DST_ALPHA	32
4.2.2.98	EDGE_STRIP_A	33
4.2.2.99	EDGE_STRIP_B	33
4.2.2.100	EDGE_STRIP_L	33
4.2.2.101	EDGE_STRIP_R	33

4.2.2.102 EQUAL	33
4.2.2.103 F16	33
4.2.2.104 FT800_ACTIVE	33
4.2.2.105 FT800_CLK36M	33
4.2.2.106 FT800_CLK48M	33
4.2.2.107 FT800_CLKEXT	34
4.2.2.108 FT800_CORERST	34
4.2.2.109 FT800_GPUACTIVE	34
4.2.2.110 FT800_PWRDOWN	34
4.2.2.111 FT800_SLEEP	34
4.2.2.112 FT800_STANDBY	34
4.2.2.113 FT800_VERSION	34
4.2.2.114 FT_CMD_FIFO_SIZE	34
4.2.2.115 FT_CMD_SIZE	35
4.2.2.116 FT_DL_SIZE	35
4.2.2.117 FTPOINTS	35
4.2.2.118 GEQUAL	35
4.2.2.119 GREATER	35
4.2.2.120 GREEN	35
4.2.2.121 INCR	35
4.2.2.122 INCR_WRAP	35
4.2.2.123 INT_CMDEEMPTY	35
4.2.2.124 INT_CMDFLAG	36
4.2.2.125 INT_CONVCOMPLETE	36
4.2.2.126 INT_PLAYBACK	36
4.2.2.127 INT_SOUND	36
4.2.2.128 INT_SWAP	36
4.2.2.129 INT_TAG	36
4.2.2.130 INT_TOUCH	36
4.2.2.131 INVALID_TOUCH_XY	36

4.2.2.132 INVERT	36
4.2.2.133 KEEP	36
4.2.2.134 L1	37
4.2.2.135 L4	37
4.2.2.136 L8	37
4.2.2.137 LCD_QVGA	37
4.2.2.138 LEQUAL	37
4.2.2.139 LESS	37
4.2.2.140 LINE_STRIP	37
4.2.2.141 LINEAR_SAMPLES	37
4.2.2.142 LINES	37
4.2.2.143 MAX	37
4.2.2.144 MEM_READ	38
4.2.2.145 MEM_WRITE	38
4.2.2.146 MIN	38
4.2.2.147 NEAREST	38
4.2.2.148 NEVER	38
4.2.2.149 NOTE	38
4.2.2.150 NOTEQUAL	38
4.2.2.151 ONE	38
4.2.2.152 ONE_MINUS_DST_ALPHA	38
4.2.2.153 ONE_MINUS_SRC_ALPHA	39
4.2.2.154 OPT_CENTER	39
4.2.2.155 OPT_CENTERX	39
4.2.2.156 OPT_CENTERY	39
4.2.2.157 OPT_FLAT	39
4.2.2.158 OPT_MONO	39
4.2.2.159 OPT_NOBACK	39
4.2.2.160 OPT_NODL	39
4.2.2.161 OPT_NOHANDS	39

4.2.2.162 OPT_NOHM	39
4.2.2.163 OPT_NOPOINTER	40
4.2.2.164 OPT_NOSECS	40
4.2.2.165 OPT_NOTICKS	40
4.2.2.166 OPT_RIGHTX	40
4.2.2.167 OPT_SIGNED	40
4.2.2.168 PALETTED	40
4.2.2.169 PLAYCOLOR	40
4.2.2.170 RAM_CMD	40
4.2.2.171 RAM_DL	40
4.2.2.172 RAM_G	40
4.2.2.173 RAM_PAL	41
4.2.2.174 RAM_REG	41
4.2.2.175 RECTS	41
4.2.2.176 RED	41
4.2.2.177 REG_CLOCK	41
4.2.2.178 REG_CMD_DL	41
4.2.2.179 REG_CMD_READ	41
4.2.2.180 REG_CMD_WRITE	41
4.2.2.181 REG_CPURESET	41
4.2.2.182 REG_CSPREAD	41
4.2.2.183 REG_DITHER	42
4.2.2.184 REG_DLSWAP	42
4.2.2.185 REG_FRAMES	42
4.2.2.186 REG_FREQUENCY	42
4.2.2.187 REG_GPIO	42
4.2.2.188 REG_GPIO_DIR	42
4.2.2.189 REG_HCYCLE	42
4.2.2.190 REG_HOFFSET	42
4.2.2.191 REG_HSIZE	42

4.2.2.192 REG_HSYNC0	42
4.2.2.193 REG_HSYNC1	43
4.2.2.194 REG_ID	43
4.2.2.195 REG_INT_EN	43
4.2.2.196 REG_INT_FLAGS	43
4.2.2.197 REG_INT_MASK	43
4.2.2.198 REG_MACRO_0	43
4.2.2.199 REG_MACRO_1	43
4.2.2.200 REG_OUTBITS	43
4.2.2.201 REG_PCLK	43
4.2.2.202 REG_PCLK_POL	43
4.2.2.203 REG_PLAY	44
4.2.2.204 REG_PLAYBACK_FORMAT	44
4.2.2.205 REG_PLAYBACK_FREQ	44
4.2.2.206 REG_PLAYBACK_LENGTH	44
4.2.2.207 REG_PLAYBACK_LOOP	44
4.2.2.208 REG_PLAYBACK_PLAY	44
4.2.2.209 REG_PLAYBACK_READPTR	44
4.2.2.210 REG_PLAYBACK_START	44
4.2.2.211 REG_PWM_DUTY	44
4.2.2.212 REG_PWM_HZ	44
4.2.2.213 REG_RENDERMODE	45
4.2.2.214 REG_ROTATE	45
4.2.2.215 REG_SNAPSHOT	45
4.2.2.216 REG_SNAPY	45
4.2.2.217 REG_SOUND	45
4.2.2.218 REG_SWIZZLE	45
4.2.2.219 REG_TAG	45
4.2.2.220 REG_TAG_X	45
4.2.2.221 REG_TAG_Y	45

4.2.2.222 REG_TAP_CRC	45
4.2.2.223 REG_TAP_MASK	46
4.2.2.224 REG_TOUCH_ADC_MODE	46
4.2.2.225 REG_TOUCH_CHARGE	46
4.2.2.226 REG_TOUCH_DIRECT_XY	46
4.2.2.227 REG_TOUCH_DIRECT_Z1Z2	46
4.2.2.228 REG_TOUCH_MODE	46
4.2.2.229 REG_TOUCH_OVERSAMPLE	46
4.2.2.230 REG_TOUCH_RAW_XY	46
4.2.2.231 REG_TOUCH_RZ	46
4.2.2.232 REG_TOUCH_RZTHRESH	46
4.2.2.233 REG_TOUCH_SCREEN_XY	47
4.2.2.234 REG_TOUCH_SETTLE	47
4.2.2.235 REG_TOUCH_TAG	47
4.2.2.236 REG_TOUCH_TAG_XY	47
4.2.2.237 REG_TOUCH_TRANSFORM_A	47
4.2.2.238 REG_TOUCH_TRANSFORM_B	47
4.2.2.239 REG_TOUCH_TRANSFORM_C	47
4.2.2.240 REG_TOUCH_TRANSFORM_D	47
4.2.2.241 REG_TOUCH_TRANSFORM_E	47
4.2.2.242 REG_TOUCH_TRANSFORM_F	47
4.2.2.243 REG_TRACKER	48
4.2.2.244 REG_VCYCLE	48
4.2.2.245 REG_VOFFSET	48
4.2.2.246 REG_VOL_PB	48
4.2.2.247 REG_VOL_SOUND	48
4.2.2.248 REG_VSIZE	48
4.2.2.249 REG_VSYNC0	48
4.2.2.250 REG_VSYNC1	48
4.2.2.251 REPEAT	48

4.2.2.252 REPLACE	48
4.2.2.253 RGB	49
4.2.2.254 RGB332	49
4.2.2.255 RGB565	49
4.2.2.256 SQ	49
4.2.2.257 SRC_ALPHA	49
4.2.2.258 TEXT8X8	49
4.2.2.259 TEXTVGA	49
4.2.2.260 TOUCHMODE_CONTINUOUS	49
4.2.2.261 TOUCHMODE_FRAME	49
4.2.2.262 TOUCHMODE_OFF	49
4.2.2.263 TOUCHMODE_ONESHOT	50
4.2.2.264 ULAW_SAMPLES	50
4.2.2.265 WHITE	50
4.2.2.266 xclock	50
4.2.2.267 xCS	50
4.2.2.268 xPD	50
4.2.2.269 xSDI	50
4.2.2.270 xSDO	50
4.2.2.271 ZERO	50
4.2.3 Function Documentation	50
4.2.3.1 delay_ms(int ms)	50
4.2.3.2 delay_us(int us)	51
4.2.3.3 ft800cmdWrite(unsigned char ftCommand)	52
4.2.3.4 ft800memRead16(unsigned long ftAddress)	52
4.2.3.5 ft800memRead32(unsigned long ftAddress)	53
4.2.3.6 ft800memRead8(unsigned long ftAddress)	53
4.2.3.7 ft800memWrite16(unsigned long ftAddress, unsigned int ftData16)	54
4.2.3.8 ft800memWrite32(unsigned long ftAddress, unsigned long ftData32)	55
4.2.3.9 ft800memWrite8(unsigned long ftAddress, unsigned char ftData8)	56

4.2.3.10	<code>getData()</code>	57
4.2.3.11	<code>incCMDOffset(unsigned int currentOffset, unsigned char commandSize)</code>	57
4.2.3.12	<code>sendData(int data)</code>	58
4.3	FT800api.cpp File Reference	59
4.3.1	Detailed Description	60
4.3.2	Function Documentation	61
4.3.2.1	<code>autko()</code>	61
4.3.2.2	<code>button(int16_t x, int16_t y, int16_t w, int16_t h, int16_t font, uint16_t options, const char *str)</code>	61
4.3.2.3	<code>calibrate()</code>	62
4.3.2.4	<code>dot(unsigned long color, unsigned int point_size, unsigned long point_x, unsigned long point_y)</code>	63
4.3.2.5	<code>initScreen()</code>	63
4.3.2.6	<code>line(unsigned long color, unsigned long line_x1, unsigned long line_y1, unsigned long line_x2, unsigned long line_y2, unsigned long width)</code>	64
4.3.2.7	<code>mainScreen()</code>	65
4.3.2.8	<code>number(int16_t x, int16_t y, int16_t font, uint16_t options, int value)</code>	66
4.3.2.9	<code>optionsScreen()</code>	66
4.3.2.10	<code>show()</code>	67
4.3.2.11	<code>smartMirrorScreen()</code>	67
4.3.2.12	<code>spinner(int16_t x, int16_t y, uint16_t style, uint16_t scale)</code>	68
4.3.2.13	<code>start(unsigned long color)</code>	69
4.3.2.14	<code>text(int16_t x, int16_t y, int16_t font, uint16_t options, const char *str)</code>	70
4.4	FT800api.h File Reference	71
4.4.1	Detailed Description	73
4.4.2	Function Documentation	73
4.4.2.1	<code>autko()</code>	73
4.4.2.2	<code>button(int16_t x, int16_t y, int16_t w, int16_t h, int16_t font, uint16_t options, const char *str)</code>	73
4.4.2.3	<code>calibrate()</code>	74
4.4.2.4	<code>dot(unsigned long color, unsigned int point_size, unsigned long point_x, unsigned long point_y)</code>	75
4.4.2.5	<code>initScreen()</code>	75

4.4.2.6	<code>line(unsigned long color, unsigned long line_x1, unsigned long line_y1, unsigned long line_x2, unsigned long line_y2, unsigned long width)</code>	76
4.4.2.7	<code>mainScreen()</code>	77
4.4.2.8	<code>number(int16_t x, int16_t y, int16_t font, uint16_t options, int32_t value)</code>	78
4.4.2.9	<code>optionsScreen()</code>	78
4.4.2.10	<code>show()</code>	78
4.4.2.11	<code>smartMirrorScreen()</code>	79
4.4.2.12	<code>spinner(int16_t x, int16_t y, uint16_t style, uint16_t scale)</code>	80
4.4.2.13	<code>start(unsigned long color)</code>	80
4.4.2.14	<code>text(int16_t x, int16_t y, int16_t font, uint16_t options, const char *str)</code>	81
4.4.2.15	<code>track(int16_t x, int16_t y, int16_t w, int16_t h, int16_t tag)</code>	82
4.4.3	Variable Documentation	82
4.4.3.1	<code>audi</code>	82
4.4.3.2	<code>cmdBufferRd</code>	82
4.4.3.3	<code>cmdBufferWr</code>	82
4.4.3.4	<code>cmdOffset</code>	82
4.4.3.5	<code>timeR</code>	82
4.5	I2C.cpp File Reference	82
4.5.1	Detailed Description	83
4.5.2	Function Documentation	83
4.5.2.1	<code>readPCF(char adres)</code>	83
4.5.3	Variable Documentation	84
4.5.3.1	<code>d</code>	84
4.6	I2C.h File Reference	84
4.6.1	Detailed Description	85
4.6.2	Macro Definition Documentation	86
4.6.2.1	<code>pinInt0</code>	86
4.6.2.2	<code>scl</code>	86
4.6.2.3	<code>sda</code>	86
4.6.3	Function Documentation	86
4.6.3.1	<code>readPCF(char adres)</code>	86

4.7	simulator.cpp File Reference	87
4.7.1	Detailed Description	87
4.7.2	Function Documentation	88
4.7.2.1	checkChangesAnalog()	88
4.7.2.2	checkChangesDigital()	88
4.7.2.3	printObj(struct car *obj, char *d)	88
4.7.2.4	readData()	89
4.7.2.5	readTemp(int portNumber)	89
4.7.2.6	save(struct car *audi, struct car *tmp)	90
4.7.2.7	sendData()	90
4.8	simulator.h File Reference	91
4.8.1	Detailed Description	92
4.8.2	Function Documentation	93
4.8.2.1	checkChangesAnalog()	93
4.8.2.2	checkChangesDigital()	93
4.8.2.3	printObj(struct car *obj, char *d)	93
4.8.2.4	readData()	94
4.8.2.5	readTemp(int portNumber)	94
4.8.2.6	save(struct car *audi, struct car *tmp)	95
4.8.2.7	sendData()	95
4.8.3	Variable Documentation	96
4.8.3.1	audi	96
4.8.3.2	dataFormat	96
4.8.3.3	saveData	96
4.8.3.4	screenNR	96
	Index	97

Chapter 1

Class Index

1.1 Class List

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

car	5
-------------------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

FT800.cpp	File containing declarations of all functions required to use with VM800	7
FT800.h	File containing declarations of all functions required to use with VM800	16
FT800api.cpp	File containing declarations of all API functions for VM800	59
FT800api.h	File containing declarations of all API functions for VM800	71
I2C.cpp	File containing declarations of function to read data with using I2C protocol	82
I2C.h	File containing declarations of function to read data with using I2C protocol	84
simulator.cpp	File containing declarations of all functions required to communication with car simulator . . .	87
simulator.h	File containing declarations of all functions required to communication with car simulator . . .	91

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::templn

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

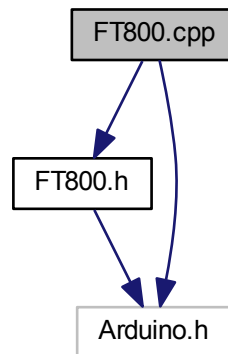
File Documentation

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*)

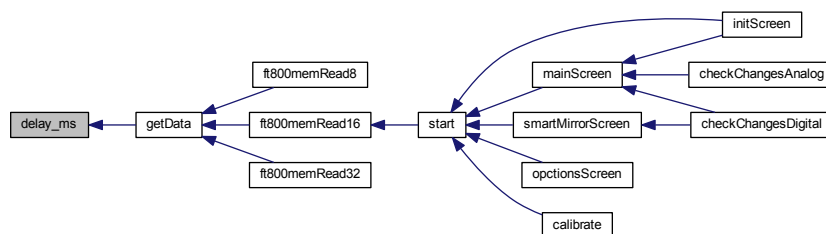
Pauses the program for the amount of time (in milisecond) specified as parameter *

Parameters

<i>ms</i>	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*)

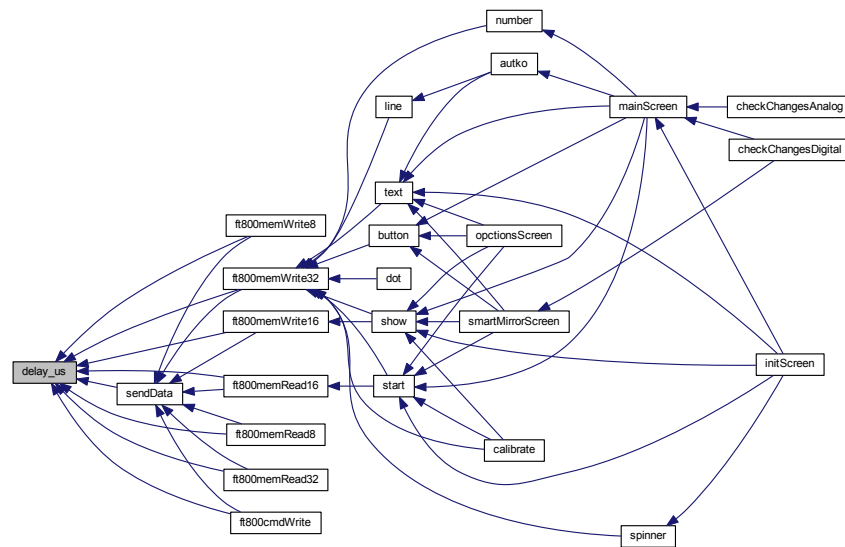
Pauses the program for the amount of time (in microsecond) specified as parameter *

Parameters

<i>us</i>	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*)

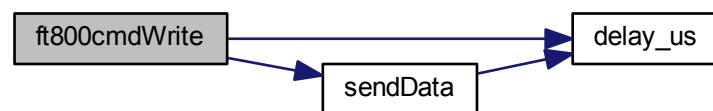
Sends FT800 command *

Parameters

<i>ftCommand</i>	command to send to device *
------------------	-----------------------------

Definition at line 304 of file FT800.cpp.

Here is the call 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

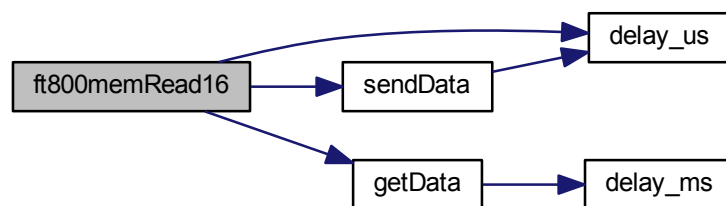
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

Returns

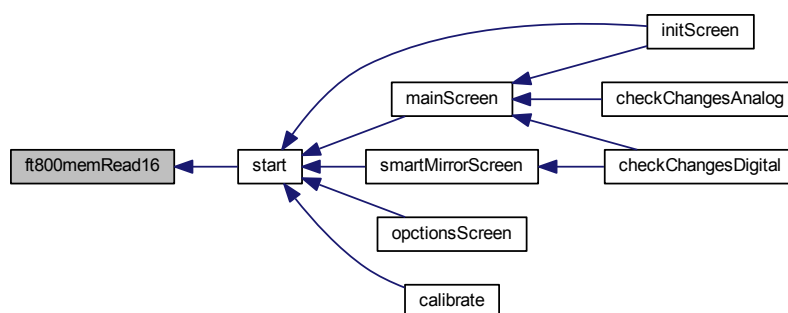
16 bit data obtained from device *

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

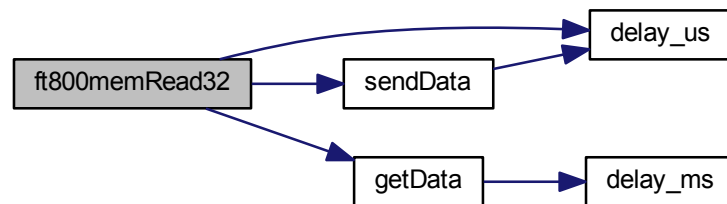
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

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*)**

Function to read 8 bit value from active device with using SPI interface *

Parameters

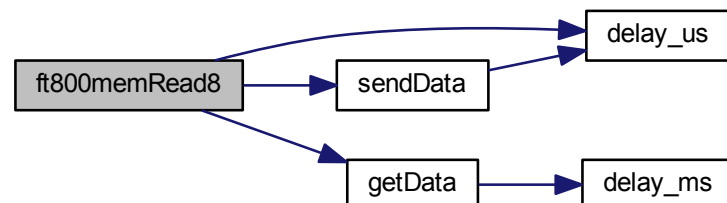
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

Returns

8 bit data obtained from device *

Definition at line 143 of file FT800.cpp.

Here is the call graph for this function:



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

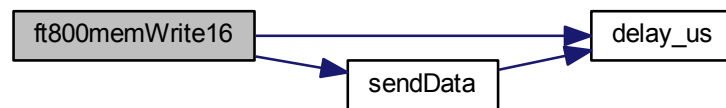
Funtion to send 16 bit value to active device with using SPI interface *

Parameters

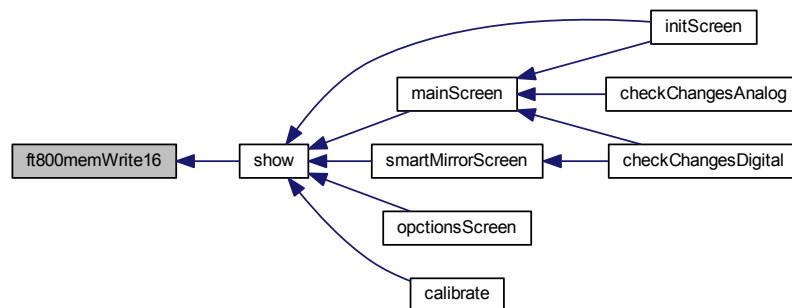
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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*)

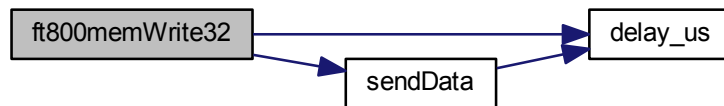
Funtion to send 32 bit value to active device with using SPI interface *

Parameters

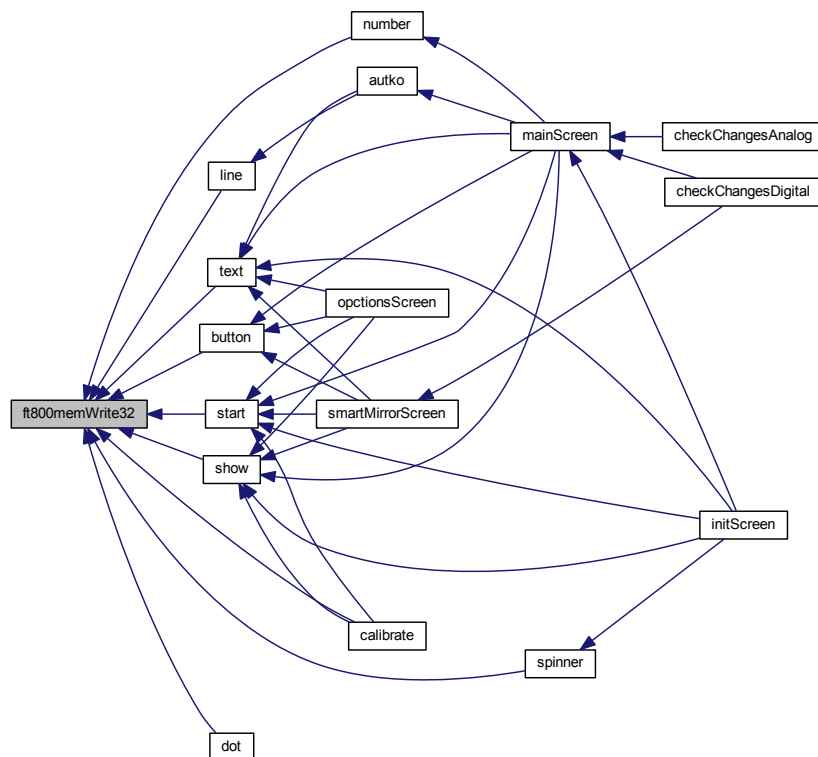
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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*)

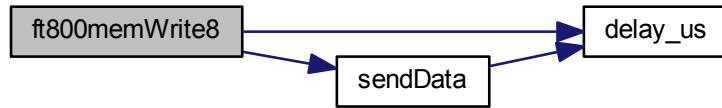
Function to send 8 bit value to active device with using SPI interface *

Parameters

<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

Definition at line 45 of file FT800.cpp.

Here is the call graph for this function:



4.1.2.10 unsigned char getData ()

Function getting data from active device with using SPI interface *

Returns

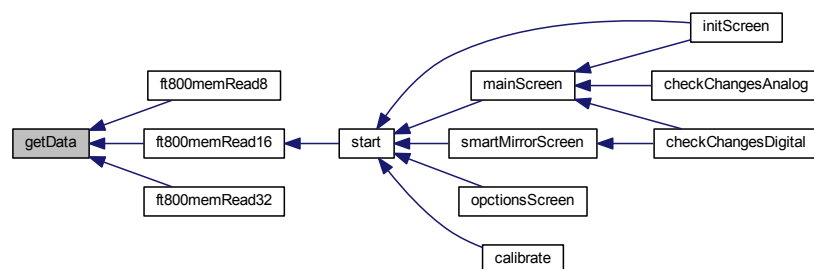
8 bit vcalue with obtained value *

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

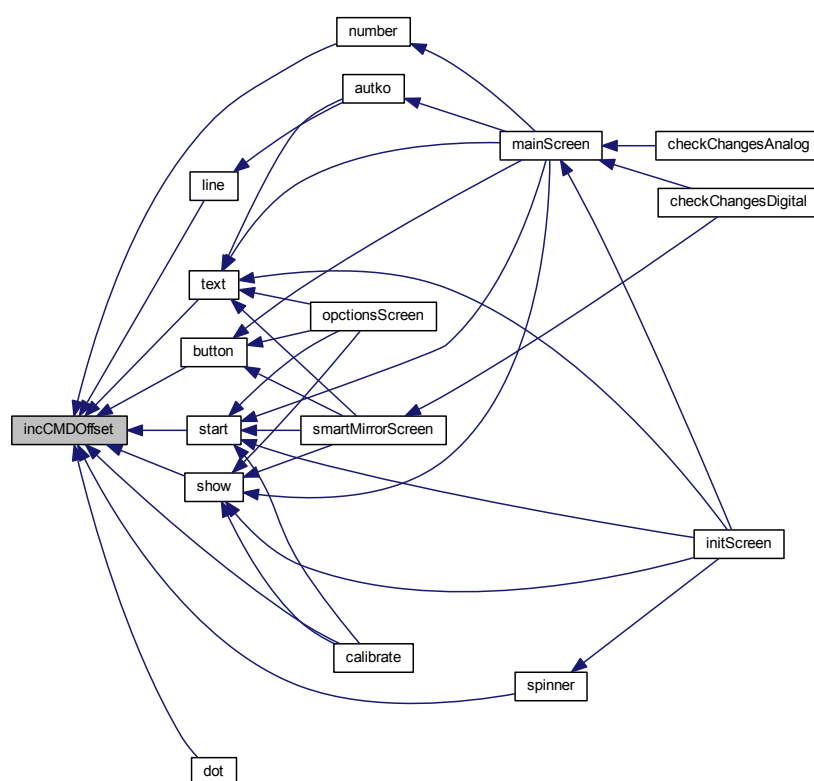
<i>currentOffset</i>	graphics processor command list pointer *
<i>commandSize</i>	number of bytes to increment the offset *

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

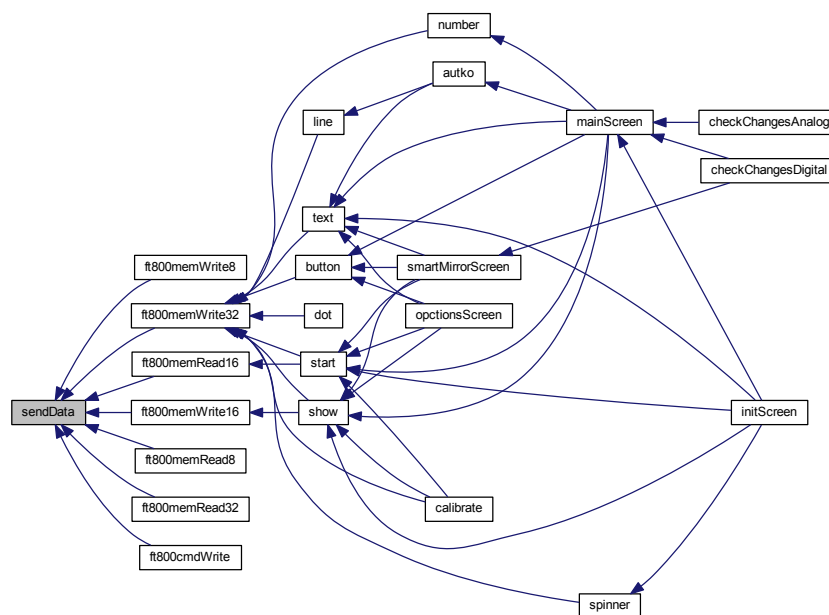
<i>data</i>	8 bit value to send to device *
-------------	---------------------------------

Definition at line 19 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

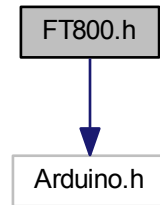


4.2 FT800.h File Reference

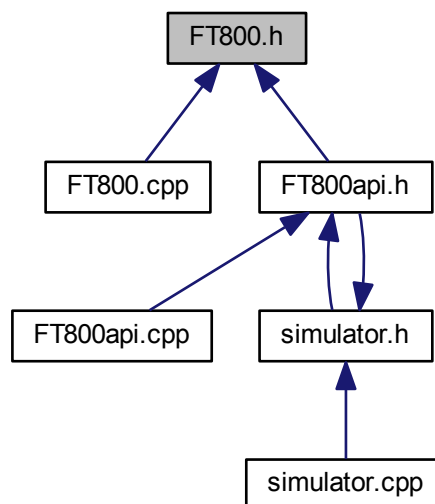
File containing declarations of all functions required to use with VM800.


```
#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` 0xfffff1eUL
- #define `CMD_BGCOLOR` 0xfffff09UL
- #define `CMD_BUTTON` 0xfffff0dUL
- #define `CMD_CALIBRATE` 0xfffff15UL
- #define `CMD_CLOCK` 0xfffff14UL
- #define `CMD_COLDSTART` 0xfffff32UL
- #define `CMD_DIAL` 0xfffff2dUL
- #define `CMD_DLSTART` 0xfffff00UL
- #define `CMD_FGCOLOR` 0xfffff0aUL
- #define `CMD_GAUGE` 0xfffff13UL
- #define `CMD_GETMATRIX` 0xfffff33UL
- #define `CMD_GETPTR` 0xfffff23UL
- #define `CMD_GRADCOLOR` 0xfffff34UL
- #define `CMD_GRADIENT` 0xfffff0bUL
- #define `CMD_INFLATE` 0xfffff22UL
- #define `CMD_INTERRUPT` 0xfffff02UL
- #define `CMD_KEYS` 0xfffff0eUL
- #define `CMD_LOADIDENTITY` 0xfffff26UL
- #define `CMD_LOADIMAGE` 0xfffff24UL
- #define `CMD_LOGO` 0xfffff31UL
- #define `CMD_MEMCPY` 0xfffff1dUL
- #define `CMD_MEMCRC` 0xfffff18UL
- #define `CMD_MEMSET` 0xfffff1bUL
- #define `CMD_MEMWRITE` 0xfffff1aUL
- #define `CMD_MEMZERO` 0xfffff1cUL
- #define `CMD_NUMBER` 0xfffff2eUL
- #define `CMD_PROGRESS` 0xfffff0fUL
- #define `CMD_REGREAD` 0xfffff19UL
- #define `CMD_ROTATE` 0xfffff29UL
- #define `CMD_SCALE` 0xfffff28UL
- #define `CMD_SCREENSAVER` 0xfffff2fUL
- #define `CMD_SCROLLBAR` 0xfffff11UL
- #define `CMD_SETFONT` 0xfffff2bUL
- #define `CMD_SETMATRIX` 0xfffff2aUL
- #define `CMD_SKETCH` 0xfffff30UL

- #define `CMD_SLIDER` 0xfffff10UL
- #define `CMD_SNAPSHOT` 0xfffff1fUL
- #define `CMD_SPINNER` 0xfffff16UL
- #define `CMD_STOP` 0xfffff17UL
- #define `CMD_SWAP` 0xfffff01UL
- #define `CMD_TEXT` 0xfffff0cUL
- #define `CMD_TOGGLE` 0xfffff12UL
- #define `CMD_TRACK` 0xfffff2cUL
- #define `CMD_TRANSLATE` 0xfffff27UL
- #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_VERTEX2I` 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_CMDEEMPTY](#) 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 0xfffff1eUL`

Definition at line 105 of file FT800.h.

4.2.2.8 `#define CMD_BGCOLOR 0xfffff09UL`

Definition at line 106 of file FT800.h.

4.2.2.9 `#define CMD_BUTTON 0xfffff0dUL`

Definition at line 107 of file FT800.h.

4.2.2.10 `#define CMD_CALIBRATE 0xfffff15UL`

Definition at line 108 of file FT800.h.

4.2.2.11 `#define CMD_CLOCK 0xfffff14UL`

Definition at line 109 of file FT800.h.

4.2.2.12 `#define CMD_COLDSTART 0xfffff32UL`

Definition at line 110 of file FT800.h.

4.2.2.13 `#define CMD_DIAL 0xfffff2dUL`

Definition at line 111 of file FT800.h.

4.2.2.14 `#define CMD_DLSTART 0xfffff00UL`

Definition at line 112 of file FT800.h.

4.2.2.15 `#define CMD_FG_COLOR 0xfffff0aUL`

Definition at line 113 of file FT800.h.

4.2.2.16 `#define CMD_GAUGE 0xfffff13UL`

Definition at line 114 of file FT800.h.

4.2.2.17 `#define CMD_GETMATRIX 0xfffff33UL`

Definition at line 115 of file FT800.h.

4.2.2.18 #define CMD_GETPTR 0xfffff23UL

Definition at line 116 of file FT800.h.

4.2.2.19 #define CMD_GRADCOLOR 0xfffff34UL

Definition at line 117 of file FT800.h.

4.2.2.20 #define CMD_GRADIENT 0xfffff0bUL

Definition at line 118 of file FT800.h.

4.2.2.21 #define CMD_INFLATE 0xfffff22UL

Definition at line 119 of file FT800.h.

4.2.2.22 #define CMD_INTERRUPT 0xfffff02UL

Definition at line 120 of file FT800.h.

4.2.2.23 #define CMD_KEYS 0xfffff0eUL

Definition at line 121 of file FT800.h.

4.2.2.24 #define CMD_LOADIDENTITY 0xfffff26UL

Definition at line 122 of file FT800.h.

4.2.2.25 #define CMD_LOADIMAGE 0xfffff24UL

Definition at line 123 of file FT800.h.

4.2.2.26 #define CMD_LOGO 0xfffff31UL

Definition at line 124 of file FT800.h.

4.2.2.27 #define CMD_MEMCPY 0xfffff1dUL

Definition at line 125 of file FT800.h.

4.2.2.28 `#define CMD_MEMCRC 0xfffff18UL`

Definition at line 126 of file FT800.h.

4.2.2.29 `#define CMD_MEMSET 0xfffff1bUL`

Definition at line 127 of file FT800.h.

4.2.2.30 `#define CMD_MEMWRITE 0xfffff1aUL`

Definition at line 128 of file FT800.h.

4.2.2.31 `#define CMD_MEMZERO 0xfffff1cUL`

Definition at line 129 of file FT800.h.

4.2.2.32 `#define CMD_NUMBER 0xfffff2eUL`

Definition at line 130 of file FT800.h.

4.2.2.33 `#define CMD_PROGRESS 0xfffff0fUL`

Definition at line 131 of file FT800.h.

4.2.2.34 `#define CMD_REGREAD 0xfffff19UL`

Definition at line 132 of file FT800.h.

4.2.2.35 `#define CMD_ROTATE 0xfffff29UL`

Definition at line 133 of file FT800.h.

4.2.2.36 `#define CMD_SCALE 0xfffff28UL`

Definition at line 134 of file FT800.h.

4.2.2.37 `#define CMD_SCREENSAVER 0xfffff2fUL`

Definition at line 135 of file FT800.h.

4.2.2.38 #define CMD_SCROLLBAR 0xfffff11UL

Definition at line 136 of file FT800.h.

4.2.2.39 #define CMD_SETFONT 0xfffff2bUL

Definition at line 137 of file FT800.h.

4.2.2.40 #define CMD_SETMATRIX 0xfffff2aUL

Definition at line 138 of file FT800.h.

4.2.2.41 #define CMD_SKETCH 0xfffff30UL

Definition at line 139 of file FT800.h.

4.2.2.42 #define CMD_SLIDER 0xfffff10UL

Definition at line 140 of file FT800.h.

4.2.2.43 #define CMD_SNAPSHOT 0xfffff1fUL

Definition at line 141 of file FT800.h.

4.2.2.44 #define CMD_SPINNER 0xfffff16UL

Definition at line 142 of file FT800.h.

4.2.2.45 #define CMD_STOP 0xfffff17UL

Definition at line 143 of file FT800.h.

4.2.2.46 #define CMD_SWAP 0xfffff01UL

Definition at line 144 of file FT800.h.

4.2.2.47 #define CMD_TEXT 0xfffff0cUL

Definition at line 145 of file FT800.h.

4.2.2.48 `#define CMD_TOGGLE 0xfffff12UL`

Definition at line 146 of file FT800.h.

4.2.2.49 `#define CMD_TRACK 0xfffff2cUL`

Definition at line 147 of file FT800.h.

4.2.2.50 `#define CMD_TRANSLATE 0xfffff27UL`

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_VERTEX2I 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_CMDEEMPTY 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)**

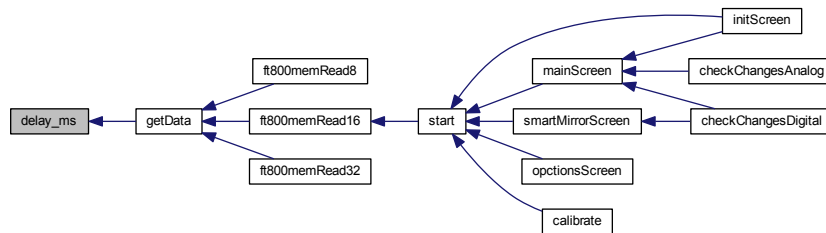
Pauses the program for the amount of time (in milisecond) specified as parameter *

Parameters

<i>ms</i>	millisecond 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)

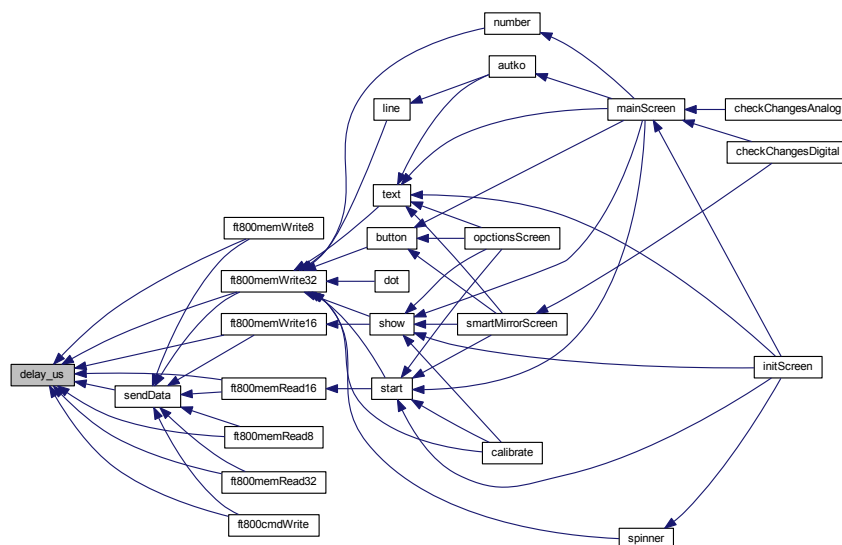
Pauses the program for the amount of time (in microsecond) specified as parameter *

Parameters

<i>us</i>	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*)

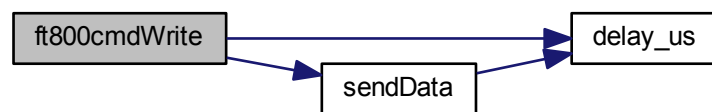
Sends FT800 command *

Parameters

<i>ftCommand</i>	command to send to device *
------------------	-----------------------------

Definition at line 304 of file FT800.cpp.

Here is the call 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

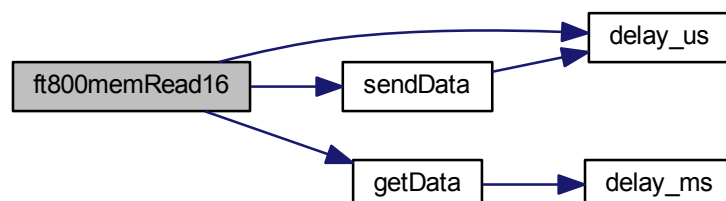
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

Returns

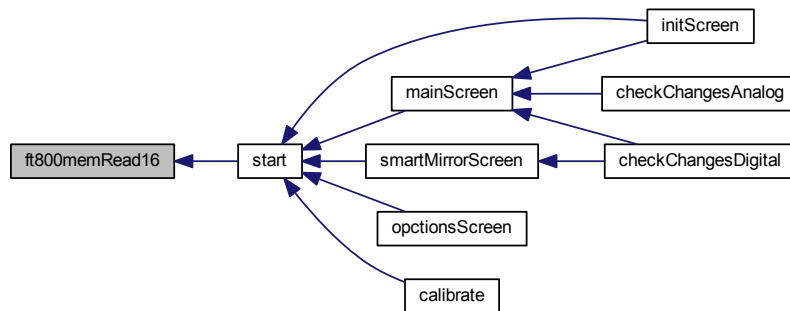
16 bit data obtained from device *

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*)

Function to read 32 bit value from active device with using SPI interface *

Parameters

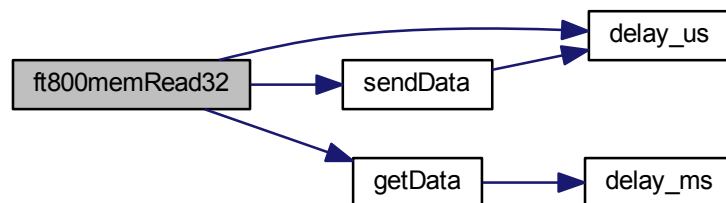
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

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*)

Function to read 8 bit value from active device with using SPI interface *

Parameters

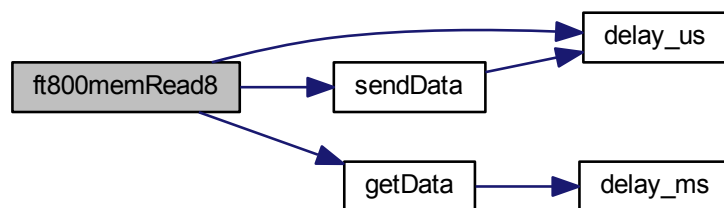
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

Returns

8 bit data obtained from device *

Definition at line 143 of file FT800.cpp.

Here is the call graph for this function:



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

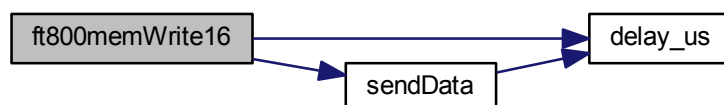
Funtion to send 16 bit value to active device with using SPI interface *

Parameters

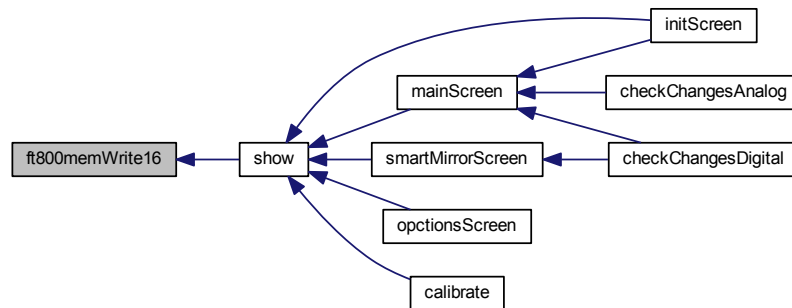
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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*)

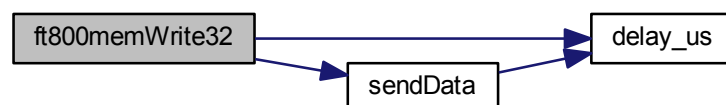
Function to send 32 bit value to active device with using SPI interface *

Parameters

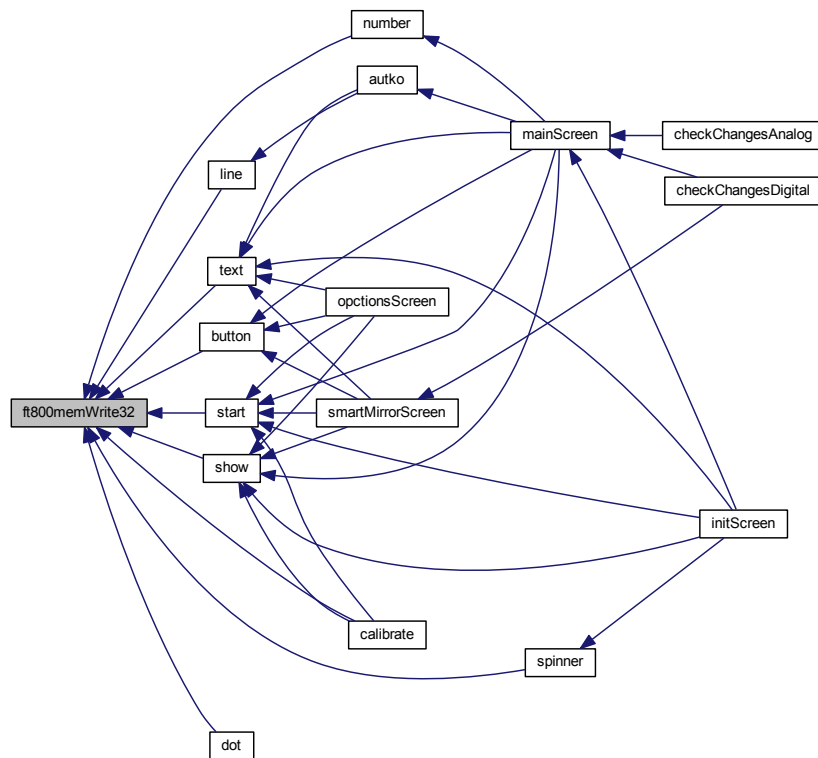
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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*)

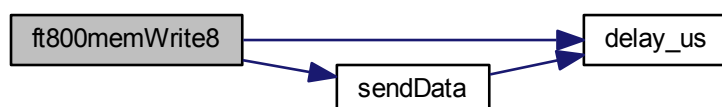
Funtion to send 8 bit value to active device with using SPI interface *

Parameters

<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

Definition at line 45 of file FT800.cpp.

Here is the call graph for this function:



4.2.3.10 unsigned char getData ()

Function getting data from active device with using SPI interface *

Returns

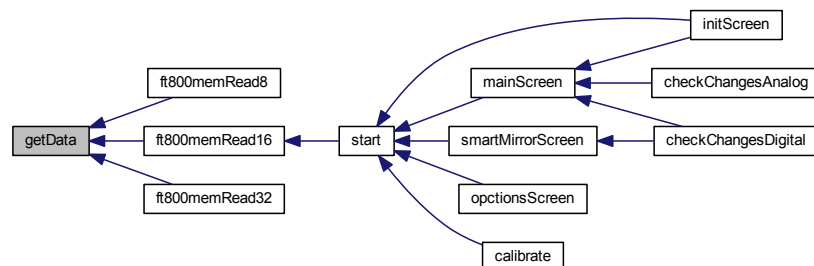
8 bit vvalue with obtained value *

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

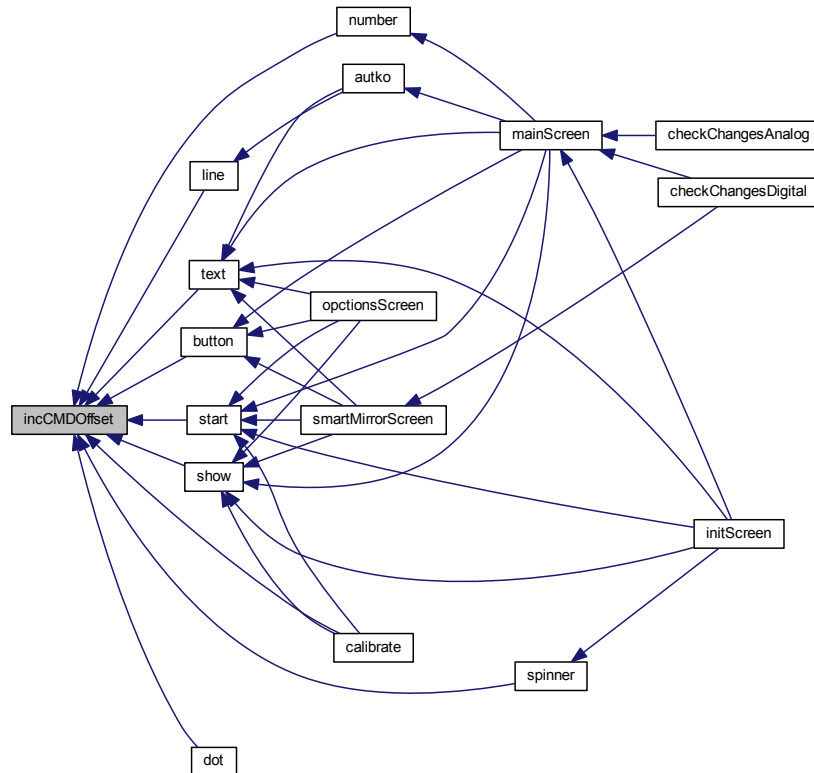
<i>currentOffset</i>	graphics processor command list pointer *
<i>commandSize</i>	number of bytes to increment the offset *

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

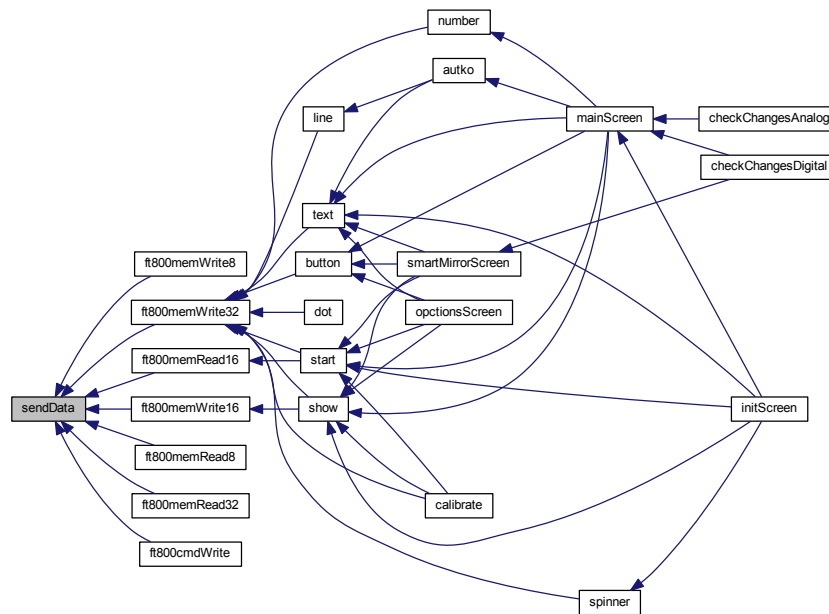
<i>data</i>	8 bit value to send to device *
-------------	---------------------------------

Definition at line 19 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

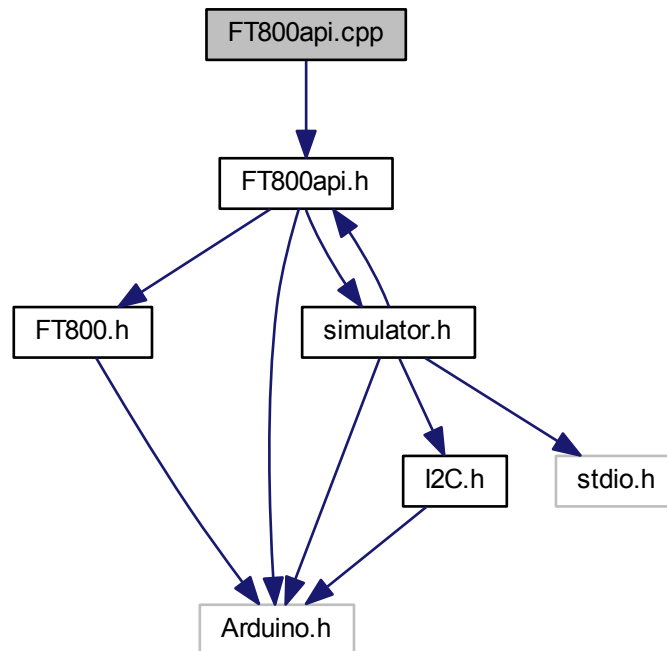


4.3 FT800api.cpp File Reference

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 [optionsScreen](#) ()
- 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

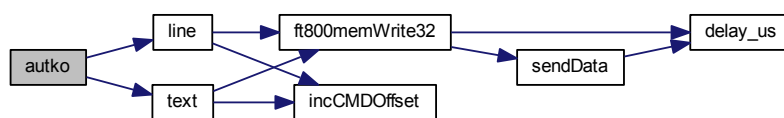
4.3.2 Function Documentation

4.3.2.1 void autko ()

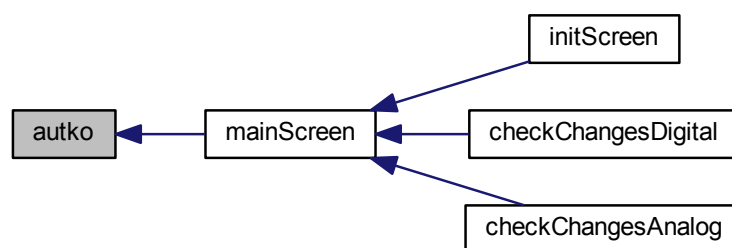
Function drawing car with proper attributes *

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

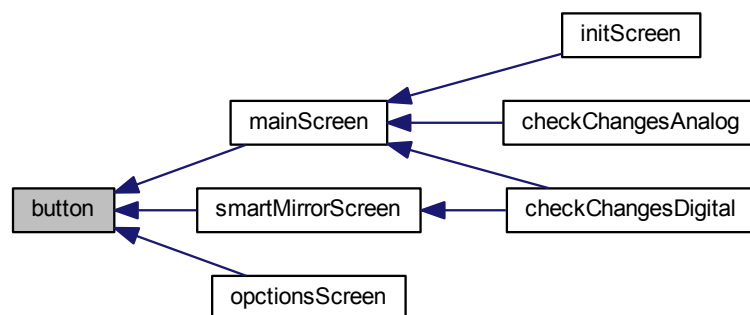
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>w</i>	width for the button *
<i>h</i>	height for the button *
<i>font</i>	font fort the button text *
<i>options</i>	options for the button *
<i>str</i>	text to draw inside button *

Definition at line 128 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

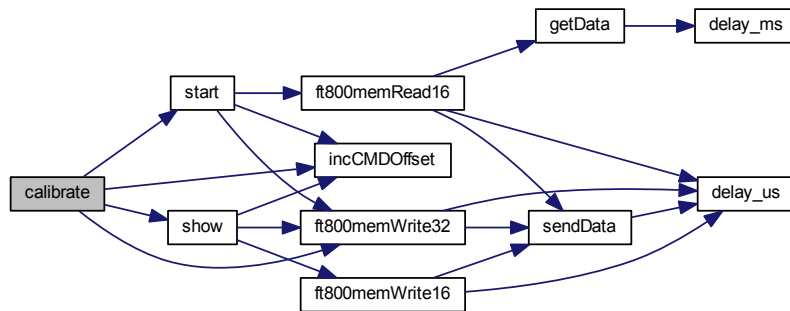


4.3.2.3 void calibrate ()

Function which calibrate screen *

Definition at line 249 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*)

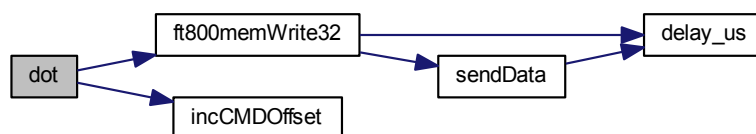
Function which draw a dot on the screen *

Parameters

<i>color</i>	dot color *
<i>point_size</i>	size for the dot *
<i>point_x</i>	x-coordinate for the dot *
<i>point_y</i>	y-coordinate for the dot *

Definition at line 231 of file FT800api.cpp.

Here is the call graph for this function:

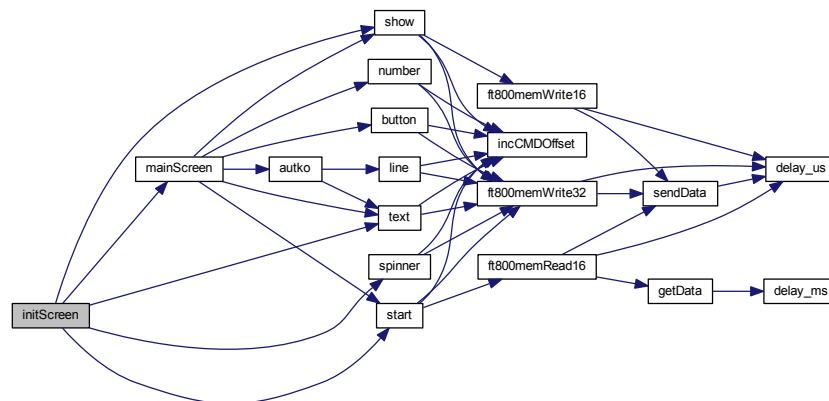


4.3.2.5 void initScreen ()

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:



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*)

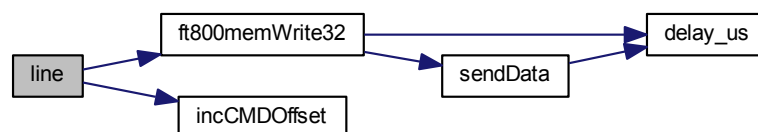
Function which draw a line on the screen *

Parameters

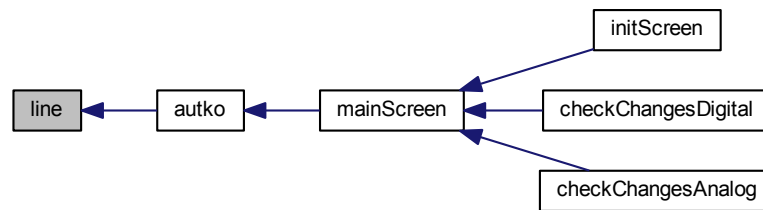
<i>color</i>	line color *
<i>line_x1</i>	x-coordinate for the beginning of the first end *
<i>line_y1</i>	y-coordinate for the beginning of the first end *
<i>line_x2</i>	x-coordinate for the beginning of the second end *
<i>line_y2</i>	y-coordinate for the beginning of the second end *
<i>width</i>	line width *

Definition at line 210 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

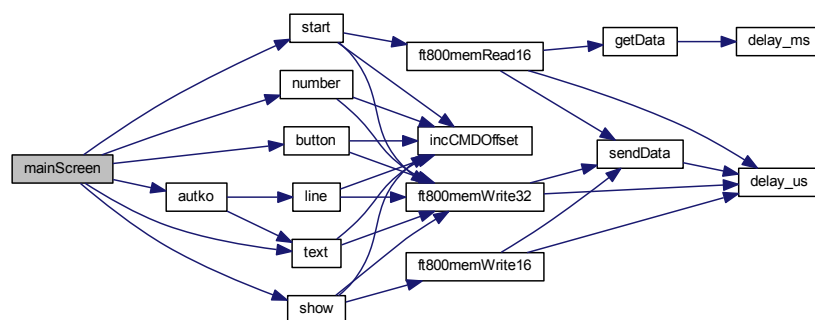


4.3.2.7 void mainScreen ()

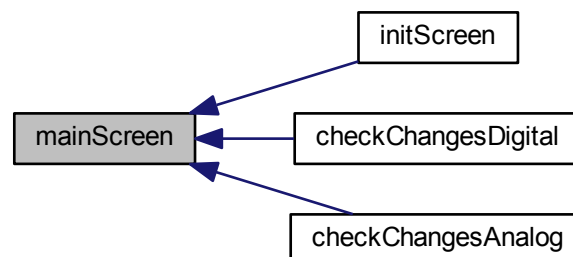
Function showing main screen *

Definition at line 49 of file FT800api.cpp.

Here is the call graph for this function:



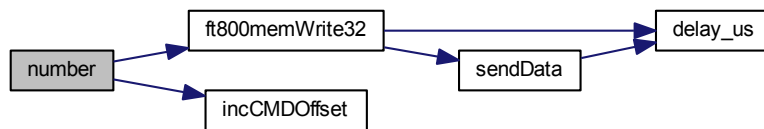
Here is the caller graph for this function:



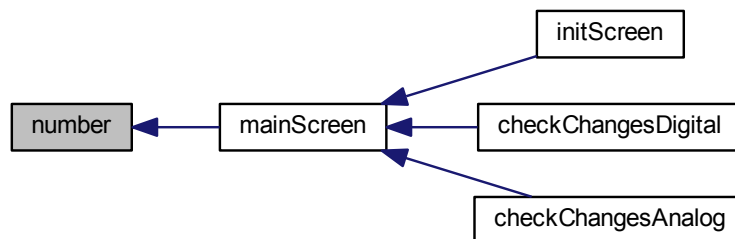
4.3.2.8 void number (int16_t x, int16_t y, int16_t font, uint16_t options, int value)

Definition at line 195 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

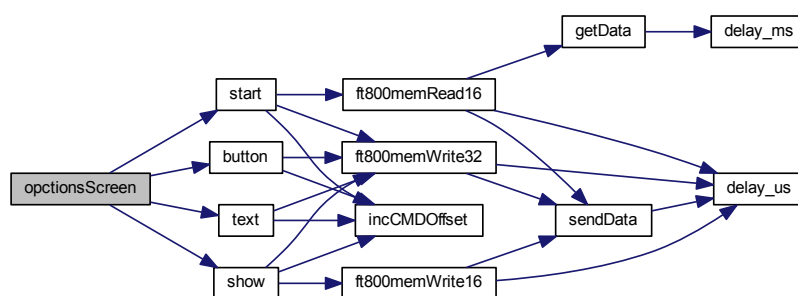


4.3.2.9 void optionsScreen ()

Function showing options screen *

Definition at line 78 of file FT800api.cpp.

Here is the call graph for this function:

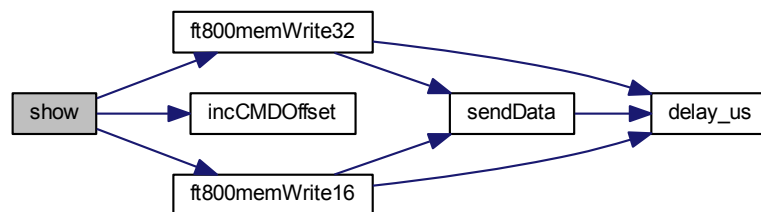


4.3.2.10 void show ()

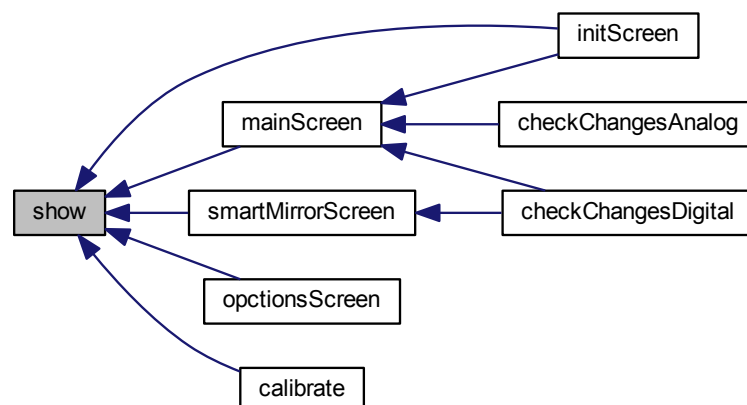
Function showing prepared screen from buffer *

Definition at line 281 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

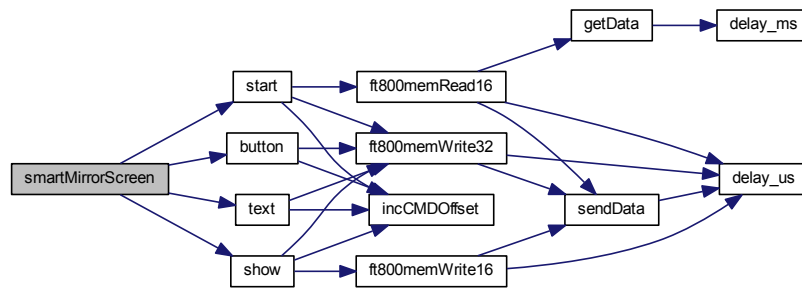


4.3.2.11 void smartMirrorScreen ()

Function showing smart mirror screen *

Definition at line 69 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)

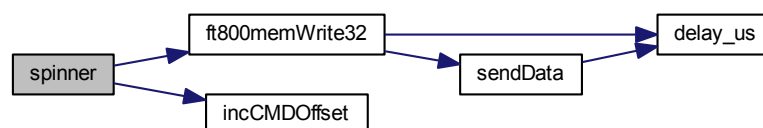
Function which draw a spinner on the screen *

Parameters

<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>style</i>	look swscreen *
<i>scale</i>	size of spinner *

Definition at line 117 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*)

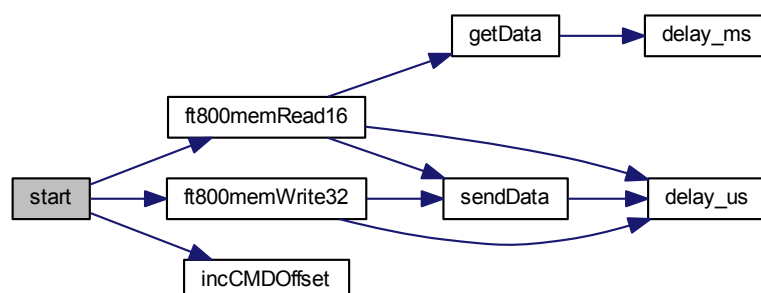
Function which start inicjalize new screen *

Parameters

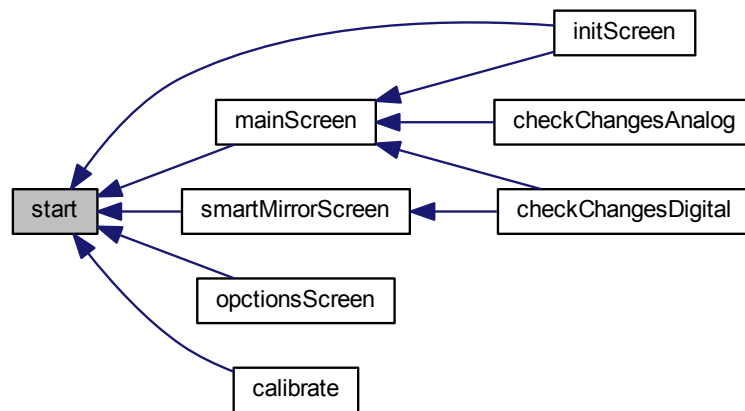
<i>color</i>	background color *
--------------	--------------------

Definition at line 260 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)

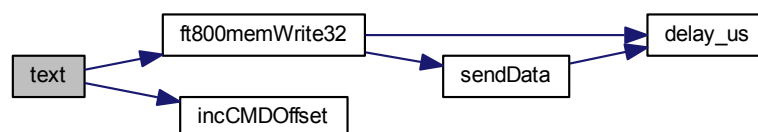
Function which draw a text on the screen *

Parameters

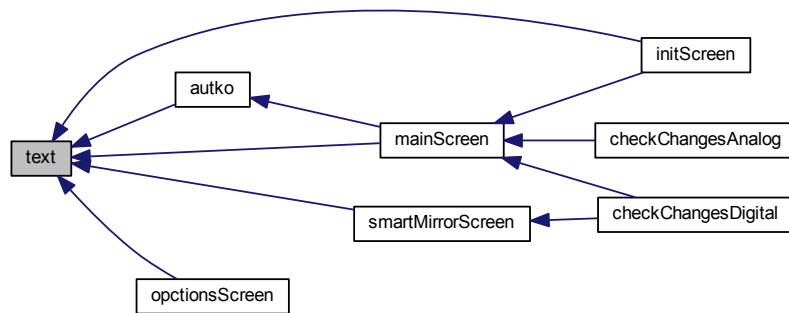
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>font</i>	font for the text *
<i>options</i>	options to set for the text *
<i>str</i>	text to draw on the screen *

Definition at line 163 of file FT800api.cpp.

Here is the call graph for this function:



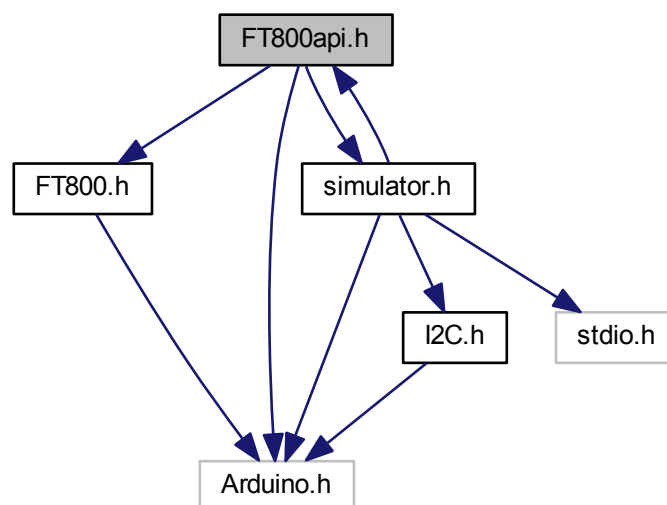
Here is the caller graph for this function:



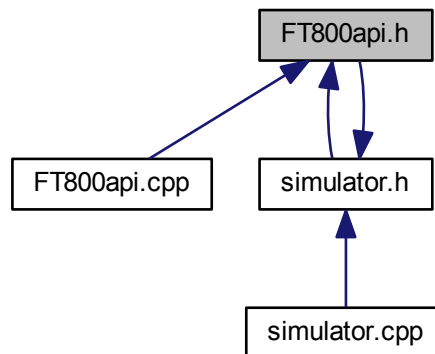
4.4 FT800api.h File Reference

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](#) ()
- void [track](#) (int16_t x, int16_t y, int16_t w, int16_t h, int16_t tag)

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

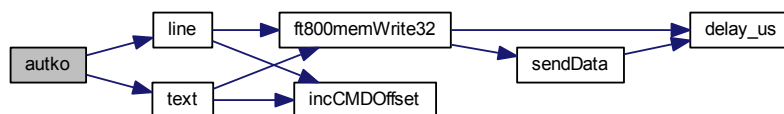
4.4.2 Function Documentation

4.4.2.1 void autko ()

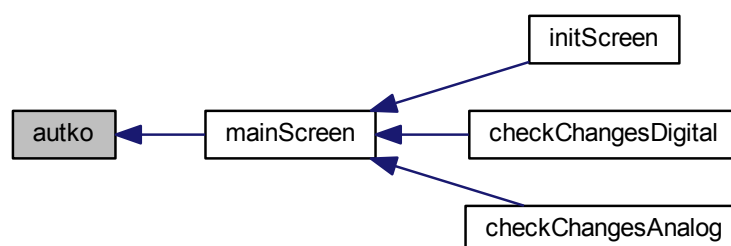
Function drawing car with proper attributes *

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)

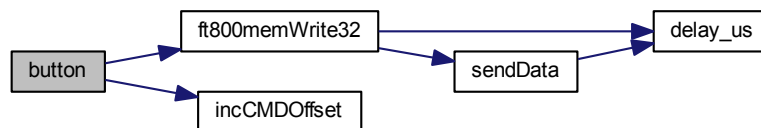
Function which draw a button on the screen *

Parameters

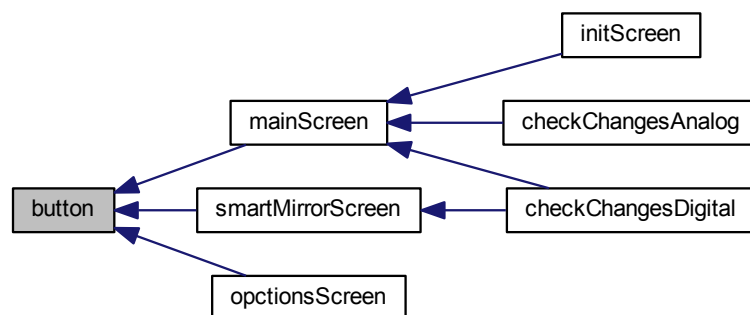
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>w</i>	width for the button *
<i>h</i>	height for the button *
<i>font</i>	font fort the button text *
<i>options</i>	options for the button *
<i>str</i>	text to draw inside button *

Definition at line 128 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

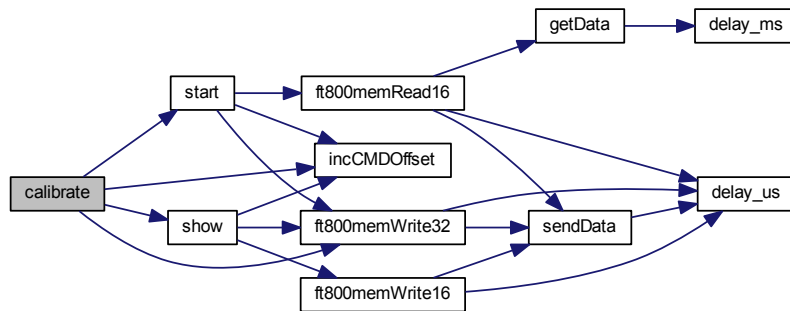


4.4.2.3 void calibrate ()

Function which calibrate screen *

Definition at line 249 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*)

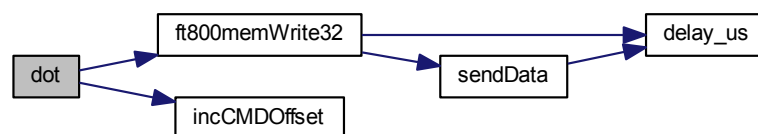
Function which draw a dot on the screen *

Parameters

<i>color</i>	dot color *
<i>point_size</i>	size for the dot *
<i>point_x</i>	x-coordinate for the dot *
<i>point_y</i>	y-coordinate for the dot *

Definition at line 231 of file FT800api.cpp.

Here is the call graph for this function:

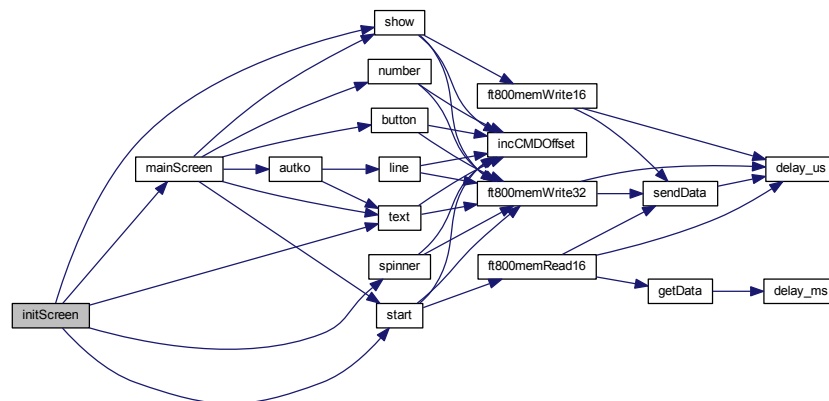


4.4.2.5 void initScreen ()

Function showing init screen during main screen is loading *

Definition at line 10 of file FT800api.cpp.

Here is the call 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*)

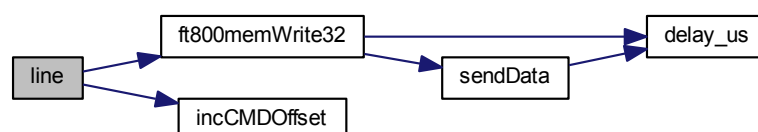
Function which draw a line on the screen *

Parameters

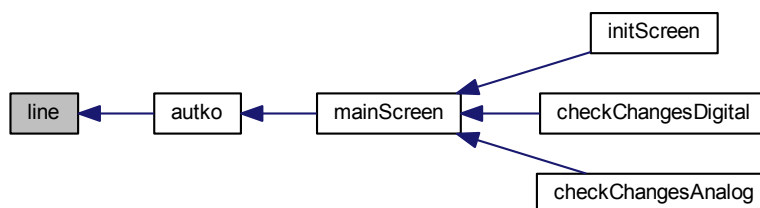
<i>color</i>	line color *
<i>line_x1</i>	x-coordinate for the beginning of the first end *
<i>line_y1</i>	y-coordinate for the beginning of the first end *
<i>line_x2</i>	x-coordinate for the beginning of the second end *
<i>line_y2</i>	y-coordinate for the beginning of the second end *
<i>width</i>	line width *

Definition at line 210 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

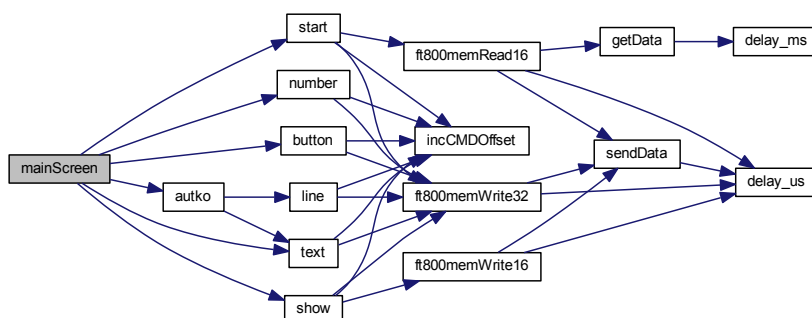


4.4.2.7 void mainScreen ()

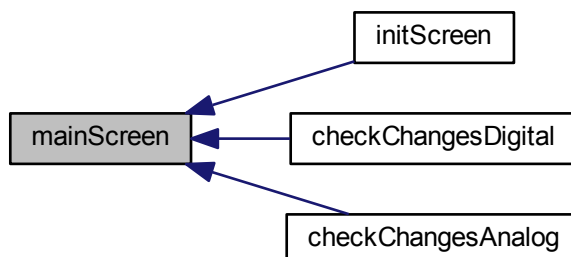
Function showing main screen *

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

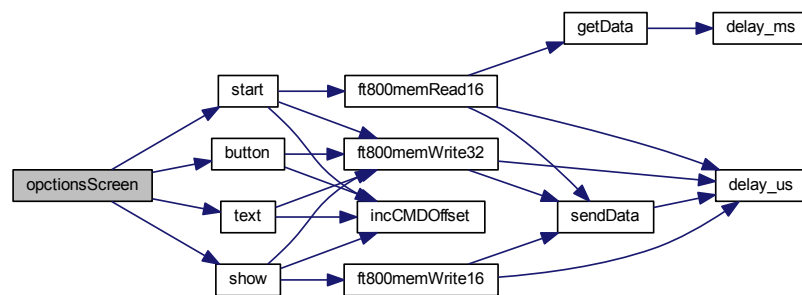
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>font</i>	font for the number *
<i>options</i>	options to set for the number *
<i>value</i>	value to draw on the screen *

4.4.2.9 void optionsScreen ()

Function showing options screen *

Definition at line 78 of file FT800api.cpp.

Here is the call graph for this function:

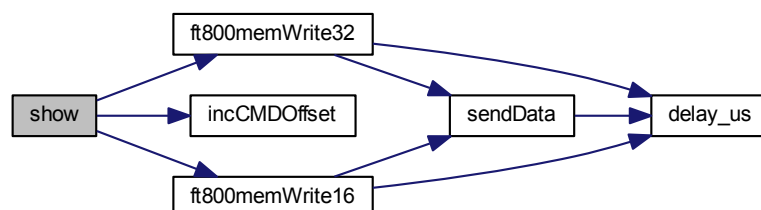


4.4.2.10 void show ()

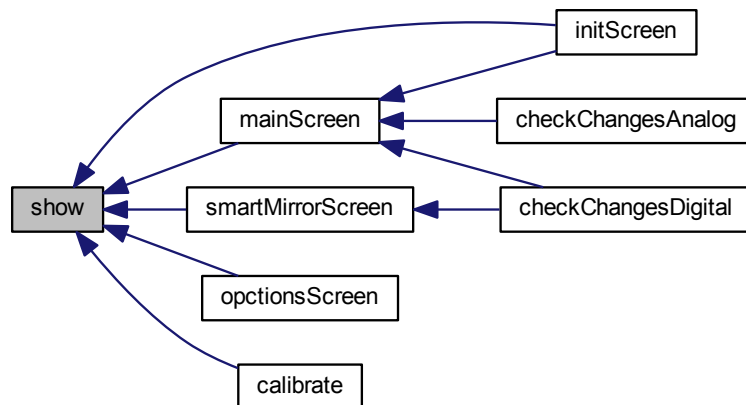
Function showing prepared screen from buffer *

Definition at line 281 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

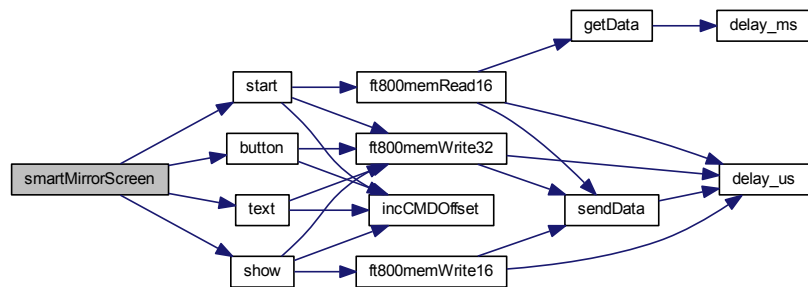


4.4.2.11 void smartMirrorScreen ()

Function showing smart mirror screen *

Definition at line 69 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)

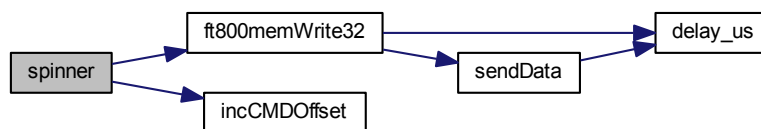
Function which draw a spinner on the screen *

Parameters

<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>style</i>	look swscreen *
<i>scale</i>	size of spinner *

Definition at line 117 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)

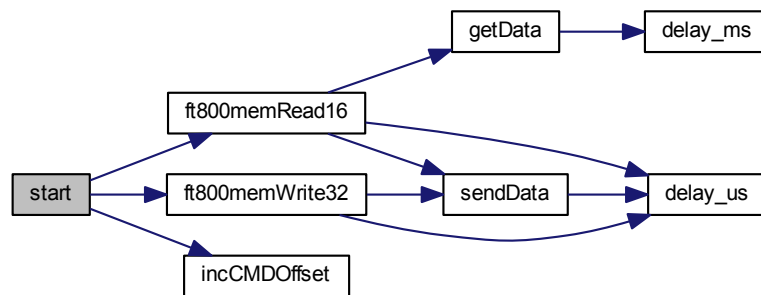
Function which start inicjalize new screen *

Parameters

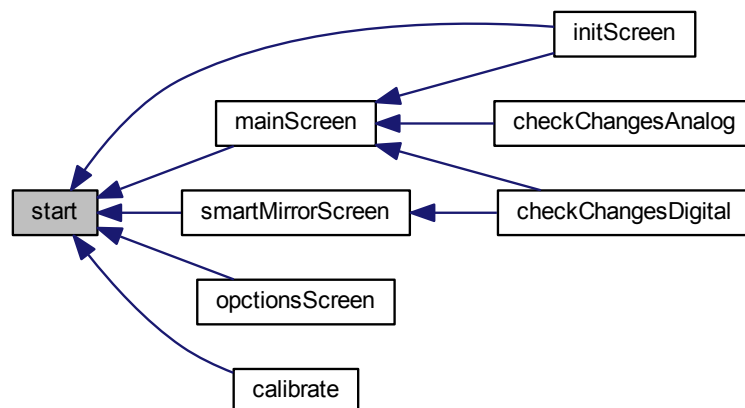
<i>color</i>	backgroud color *
--------------	-------------------

Definition at line 260 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)

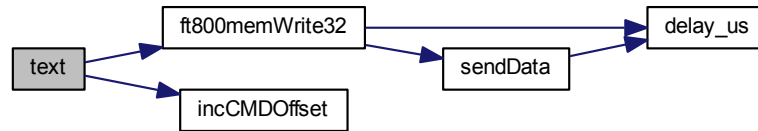
Function which draw a text on the screen *

Parameters

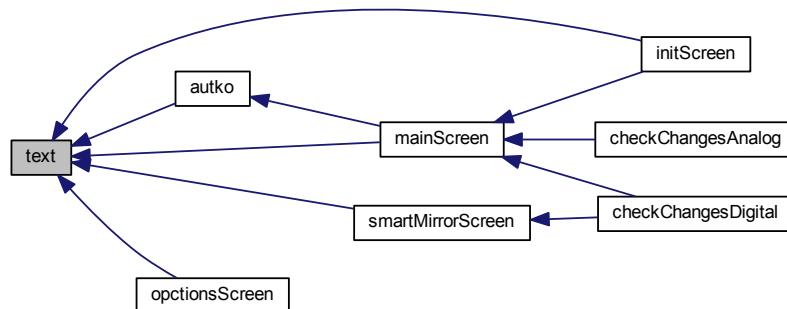
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>font</i>	font for the text *
<i>options</i>	options to set for the text *
<i>str</i>	text to draw on the screen *

Definition at line 163 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.2.15 void track (int16_t x, int16_t y, int16_t w, int16_t h, int16_t tag)

4.4.3 Variable Documentation

4.4.3.1 struct car* audi

4.4.3.2 unsigned int cmdBufferRd

4.4.3.3 unsigned int cmdBufferWr

4.4.3.4 unsigned int cmdOffset

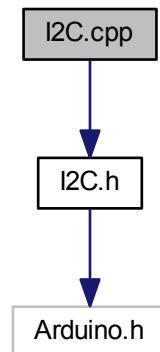
4.4.3.5 int timeR

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

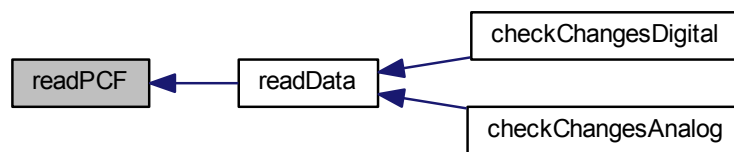
<i>adres</i>	The address of PCF8574N I/O Expander *
--------------	--

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 handling

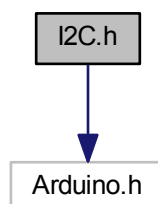
Definition at line 10 of file I2C.cpp.

4.6 I2C.h File Reference

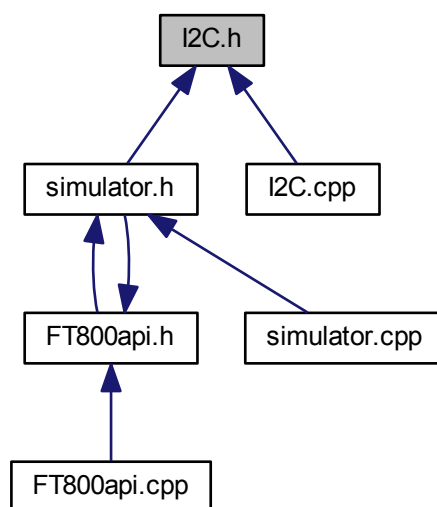
File containing declarations of function to read data with using I2C protocol.

```
#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`

Interrupt 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

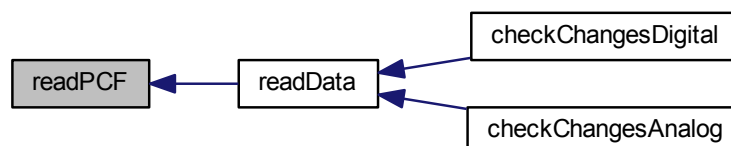
<i>adres</i>	The address of PCF8574N I/O Expander *
--------------	--

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:

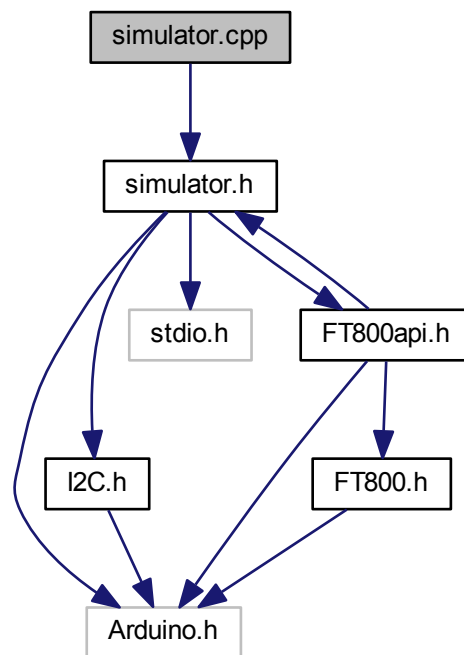


4.7 simulator.cpp File Reference

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

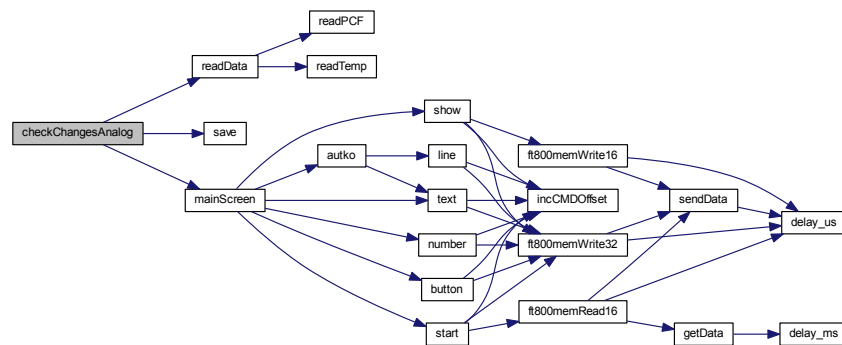
4.7.2 Function Documentation

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:

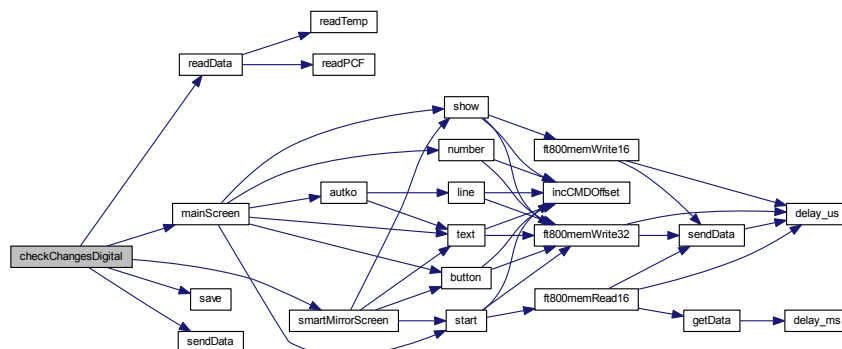


4.7.2.2 void checkChangesDigital ()

Check if sth on digital ports was changed *

Definition at line 96 of file simulator.cpp.

Here is the call graph for this function:



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

<i>Car</i>	struct to print and save with selected format into file*
<i>d</i>	actual date *

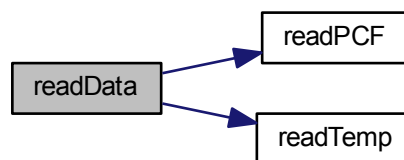
Definition at line 10 of file simulator.cpp.

4.7.2.4 struct car* readData ()

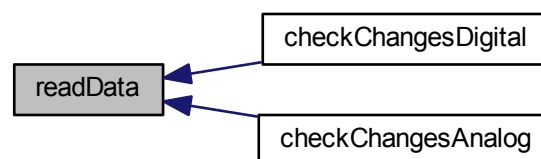
Reading data about car status *

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

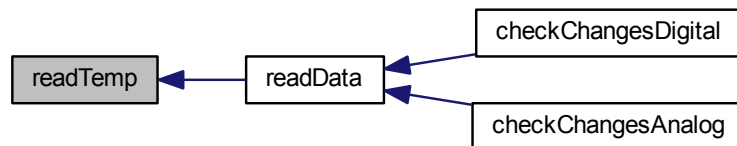
<i>portNumber</i>	The number of the analog input pin to read *
-------------------	--

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)

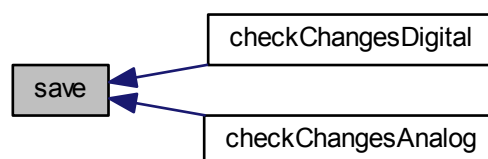
Copying data function from temporary to main struct *

Parameters

<i>*audi,*tmp</i>	Structures to and from which data are copied *
-------------------	--

Definition at line 67 of file simulator.cpp.

Here is the caller graph for this function:

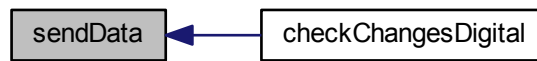


4.7.2.7 void sendData ()

Sending actual data to web server *

Definition at line 132 of file simulator.cpp.

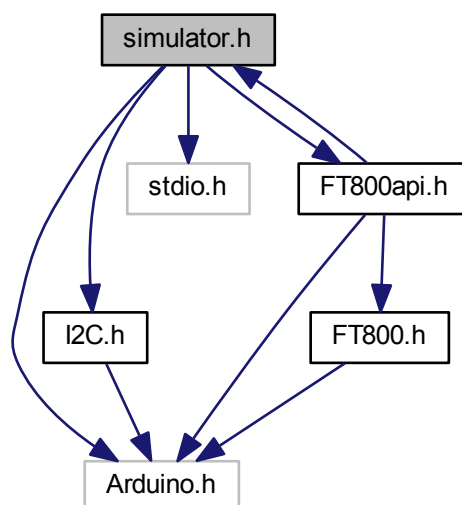
Here is the caller graph for this function:



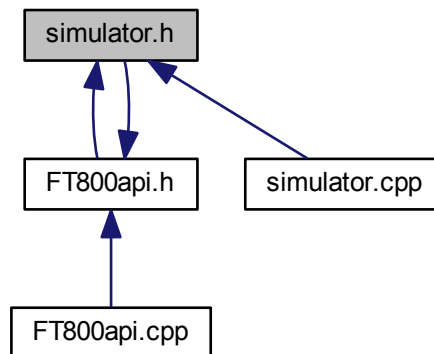
4.8 simulator.h File Reference

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

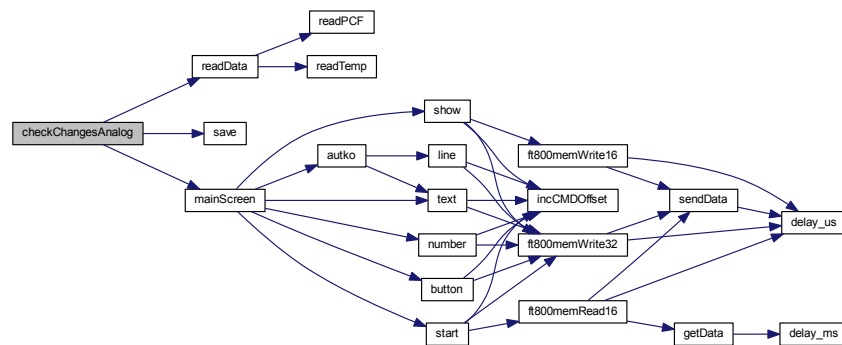
4.8.2 Function Documentation

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:

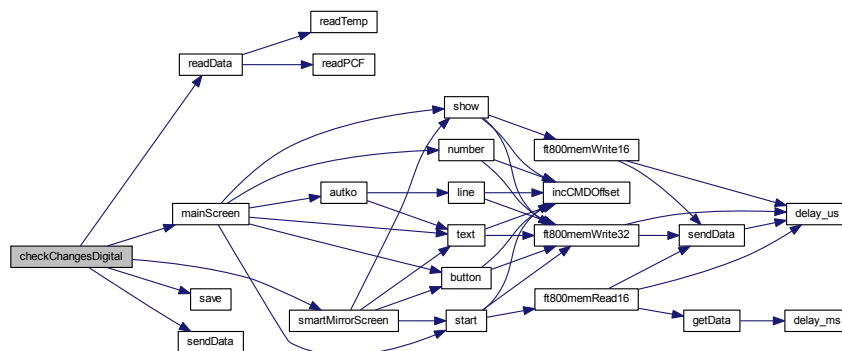


4.8.2.2 void checkChangesDigital ()

Check if sth on digital ports was changed *

Definition at line 96 of file simulator.cpp.

Here is the call graph for this function:



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

<i>Car</i>	struct to print and save with selected format into file*
<i>d</i>	actual date *

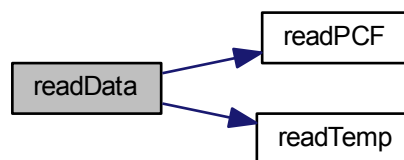
Definition at line 10 of file simulator.cpp.

4.8.2.4 struct car* readData ()

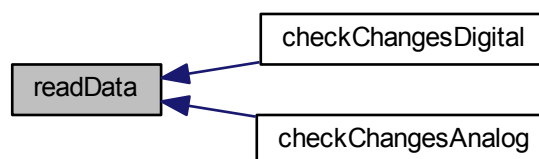
Reading data about car status *

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

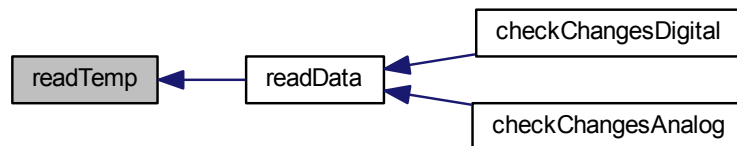
<i>portNumber</i>	The number of the analog input pin to read *
-------------------	--

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)

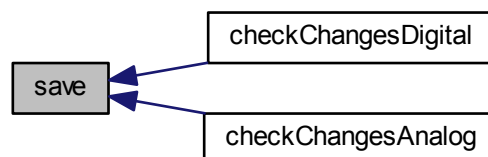
Copying data function from temporary to main struct *

Parameters

<i>*audi,*tmp</i>	Structures to and from which data are copied *
-------------------	--

Definition at line 67 of file simulator.cpp.

Here is the caller graph for this function:

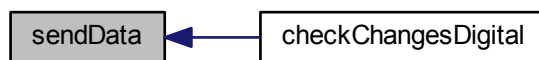


4.8.2.7 void sendData ()

Sending actual 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`

4.8.3.2 `int` `dataFormat`

4.8.3.3 `int` `saveData`

4.8.3.4 `short int` `screenNR`

Index

ABS
 FT800.h, [23](#)

audi
 FT800api.h, [82](#)
 simulator.h, [96](#)

autko
 FT800api.cpp, [61](#)
 FT800api.h, [73](#)

BLACK
 FT800.h, [23](#)

BLUE
 FT800.h, [23](#)

button
 FT800api.cpp, [61](#)
 FT800api.h, [73](#)

CLR_COL
 FT800.h, [23](#)

CLR_STN
 FT800.h, [23](#)

CLR_TAG
 FT800.h, [23](#)

CMD_APPEND
 FT800.h, [23](#)

CMD_BGCOLOR
 FT800.h, [23](#)

CMD_BUTTON
 FT800.h, [24](#)

CMD_CALIBRATE
 FT800.h, [24](#)

CMD_CLOCK
 FT800.h, [24](#)

CMD_COLDSTART
 FT800.h, [24](#)

CMD_DIAL
 FT800.h, [24](#)

CMD_DLSTART
 FT800.h, [24](#)

CMD_FGCOLOR
 FT800.h, [24](#)

CMD_GAUGE
 FT800.h, [24](#)

CMD_GETMATRIX
 FT800.h, [24](#)

CMD_GETPTR
 FT800.h, [24](#)

CMD_GRADCOLOR
 FT800.h, [25](#)

CMD_GRADIENT
 FT800.h, [25](#)

CMD_INFLATE
 FT800.h, [25](#)

CMD_INTERRUPT
 FT800.h, [25](#)

CMD_KEYS
 FT800.h, [25](#)

CMD_LOADIDENTITY
 FT800.h, [25](#)

CMD_LOADIMAGE
 FT800.h, [25](#)

CMD_LOGO
 FT800.h, [25](#)

CMD_MEMCPY
 FT800.h, [25](#)

CMD_MEMCRC
 FT800.h, [25](#)

CMD_MEMSET
 FT800.h, [26](#)

CMD_MEMWRITE
 FT800.h, [26](#)

CMD_MEMZERO
 FT800.h, [26](#)

CMD_NUMBER
 FT800.h, [26](#)

CMD_PROGRESS
 FT800.h, [26](#)

CMD_REGREAD
 FT800.h, [26](#)

CMD_ROTATE
 FT800.h, [26](#)

CMD_SCALE
 FT800.h, [26](#)

CMD_SCREENSAVER
 FT800.h, [26](#)

CMD_SCROLLBAR
 FT800.h, [26](#)

CMD_SETFONT
 FT800.h, [27](#)

CMD_SETMATRIX
 FT800.h, [27](#)

CMD_SKETCH
 FT800.h, [27](#)

CMD_SLIDER
 FT800.h, [27](#)

CMD_SNAPSHOT
 FT800.h, [27](#)

CMD_SPINNER
 FT800.h, [27](#)

CMD_STOP
 FT800.h, [27](#)
 CMD_SWAP
 FT800.h, [27](#)
 CMD_TEXT
 FT800.h, [27](#)
 CMD_TOGGLE
 FT800.h, [27](#)
 CMD_TRACK
 FT800.h, [28](#)
 CMD_TRANSLATE
 FT800.h, [28](#)
 CMDBUF_SIZE
 FT800.h, [28](#)
 calibrate
 FT800api.cpp, [62](#)
 FT800api.h, [74](#)
 car, [5](#)
 doors, [5](#)
 lights, [5](#)
 r, [6](#)
 seatbelts, [6](#)
 tempEngine, [6](#)
 tempIn, [6](#)
 tempOut, [6](#)
 checkChangesAnalog
 simulator.cpp, [88](#)
 simulator.h, [93](#)
 checkChangesDigital
 simulator.cpp, [88](#)
 simulator.h, [93](#)
 cmdBufferRd
 FT800api.h, [82](#)
 cmdBufferWr
 FT800api.h, [82](#)
 cmdOffset
 FT800api.h, [82](#)

 d
 I2C.cpp, [84](#)
 DECR_WRAP
 FT800.h, [28](#)
 DECR
 FT800.h, [28](#)
 DL_ALPHA_FUNC
 FT800.h, [28](#)
 DL_BEGIN
 FT800.h, [28](#)
 DL_BITMAP_HANDLE
 FT800.h, [28](#)
 DL_BITMAP_LAYOUT
 FT800.h, [28](#)
 DL_BITMAP_SIZE
 FT800.h, [28](#)
 DL_BITMAP_SOURCE
 FT800.h, [29](#)
 DL_BITMAP_TFORM_A
 FT800.h, [29](#)
 DL_BITMAP_TFORM_B
 FT800.h, [29](#)
 DL_BITMAP_TFORM_C
 FT800.h, [29](#)
 DL_BITMAP_TFORM_D
 FT800.h, [29](#)
 DL_BITMAP_TFORM_E
 FT800.h, [29](#)
 DL_BITMAP_TFORM_F
 FT800.h, [29](#)
 DL_BLEND_FUNC
 FT800.h, [29](#)
 DL_CALL
 FT800.h, [29](#)
 DL_CELL
 FT800.h, [29](#)
 DL_CLEAR_RGB
 FT800.h, [30](#)
 DL_CLEAR_STENCIL
 FT800.h, [30](#)
 DL_CLEAR_TAG
 FT800.h, [30](#)
 DL_CLEAR
 FT800.h, [30](#)
 DL_COLOR_MASK
 FT800.h, [30](#)
 DL_COLOR_RGB
 FT800.h, [30](#)
 DL_COLOR_A
 FT800.h, [30](#)
 DL_DISPLAY
 FT800.h, [30](#)
 DL_END
 FT800.h, [30](#)
 DL_JUMP
 FT800.h, [30](#)
 DL_LINE_WIDTH
 FT800.h, [31](#)
 DL_MACRO
 FT800.h, [31](#)
 DL_POINT_SIZE
 FT800.h, [31](#)
 DL_RESTORE_CONTEXT
 FT800.h, [31](#)
 DL_RETURN
 FT800.h, [31](#)
 DL_SAVE_CONTEXT
 FT800.h, [31](#)
 DL_SCISSOR_SIZE
 FT800.h, [31](#)
 DL_SCISSOR_XY
 FT800.h, [31](#)
 DL_STENCIL_FUNC
 FT800.h, [31](#)
 DL_STENCIL_MASK
 FT800.h, [31](#)
 DL_STENCIL_OP
 FT800.h, [32](#)
 DL_TAG_MASK

- FT800.h, [32](#)
- DL_TAG
 - FT800.h, [32](#)
- DL_VERTEX2II
 - FT800.h, [32](#)
- DL_VERTEX2F
 - FT800.h, [32](#)
- DLSWAP_DONE
 - FT800.h, [32](#)
- DLSWAP_FRAME
 - FT800.h, [32](#)
- DLSWAP_LINE
 - FT800.h, [32](#)
- DST_ALPHA
 - FT800.h, [32](#)
- dataFormat
 - simulator.h, [96](#)
- delay_ms
 - FT800.cpp, [8](#)
 - FT800.h, [50](#)
- delay_us
 - FT800.cpp, [8](#)
 - FT800.h, [51](#)
- doors
 - car, [5](#)
- dot
 - FT800api.cpp, [63](#)
 - FT800api.h, [75](#)
- EDGE_STRIP_A
 - FT800.h, [32](#)
- EDGE_STRIP_B
 - FT800.h, [33](#)
- EDGE_STRIP_L
 - FT800.h, [33](#)
- EDGE_STRIP_R
 - FT800.h, [33](#)
- EQUAL
 - FT800.h, [33](#)
- F16
 - FT800.h, [33](#)
- FT800.cpp, [7](#)
 - delay_ms, [8](#)
 - delay_us, [8](#)
 - ft800cmdWrite, [9](#)
 - ft800memRead16, [9](#)
 - ft800memRead32, [10](#)
 - ft800memRead8, [11](#)
 - ft800memWrite16, [11](#)
 - ft800memWrite32, [12](#)
 - ft800memWrite8, [13](#)
 - getData, [14](#)
 - incCMDOffset, [14](#)
 - sendData, [15](#)
- FT800.h, [16](#)
 - ABS, [23](#)
 - BLACK, [23](#)
 - BLUE, [23](#)
 - CLR_COL, [23](#)
 - CLR_STN, [23](#)
 - CLR_TAG, [23](#)
 - CMD_APPEND, [23](#)
 - CMD_BGCOLOR, [23](#)
 - CMD_BUTTON, [24](#)
 - CMD_CALIBRATE, [24](#)
 - CMD_CLOCK, [24](#)
 - CMD_COLDSTART, [24](#)
 - CMD_DIAL, [24](#)
 - CMD_DLSTART, [24](#)
 - CMD_FGCOLOR, [24](#)
 - CMD_GAUGE, [24](#)
 - CMD_GETMATRIX, [24](#)
 - CMD_GETPTR, [24](#)
 - CMD_GRADCOLOR, [25](#)
 - CMD_GRADIENT, [25](#)
 - CMD_INFLATE, [25](#)
 - CMD_INTERRUPT, [25](#)
 - CMD_KEYS, [25](#)
 - CMD_LOADIDENTITY, [25](#)
 - CMD_LOADIMAGE, [25](#)
 - CMD_LOGO, [25](#)
 - CMD_MEMCPY, [25](#)
 - CMD_MEMCRC, [25](#)
 - CMD_MEMSET, [26](#)
 - CMD_MEMWRITE, [26](#)
 - CMD_MEMZERO, [26](#)
 - CMD_NUMBER, [26](#)
 - CMD_PROGRESS, [26](#)
 - CMD_REGREAD, [26](#)
 - CMD_ROTATE, [26](#)
 - CMD_SCALE, [26](#)
 - CMD_SCREENSAVER, [26](#)
 - CMD_SCROLLBAR, [26](#)
 - CMD_SETFONT, [27](#)
 - CMD_SETMATRIX, [27](#)
 - CMD_SKETCH, [27](#)
 - CMD_SLIDER, [27](#)
 - CMD_SNAPSHOT, [27](#)
 - CMD_SPINNER, [27](#)
 - CMD_STOP, [27](#)
 - CMD_SWAP, [27](#)
 - CMD_TEXT, [27](#)
 - CMD_TOGGLE, [27](#)
 - CMD_TRACK, [28](#)
 - CMD_TRANSLATE, [28](#)
 - CMDBUF_SIZE, [28](#)
 - DECR_WRAP, [28](#)
 - DECR, [28](#)
 - DL_ALPHA_FUNC, [28](#)
 - DL_BEGIN, [28](#)
 - DL_BITMAP_HANDLE, [28](#)
 - DL_BITMAP_LAYOUT, [28](#)
 - DL_BITMAP_SIZE, [28](#)
 - DL_BITMAP_SOURCE, [29](#)
 - DL_BITMAP_TFORM_A, [29](#)
 - DL_BITMAP_TFORM_B, [29](#)

DL_BITMAP_TFORM_C, 29
DL_BITMAP_TFORM_D, 29
DL_BITMAP_TFORM_E, 29
DL_BITMAP_TFORM_F, 29
DL_BLEND_FUNC, 29
DL_CALL, 29
DL_CELL, 29
DL_CLEAR_RGB, 30
DL_CLEAR_STENCIL, 30
DL_CLEAR_TAG, 30
DL_CLEAR, 30
DL_COLOR_MASK, 30
DL_COLOR_RGB, 30
DL_COLOR_A, 30
DL_DISPLAY, 30
DL_END, 30
DL_JUMP, 30
DL_LINE_WIDTH, 31
DL_MACRO, 31
DL_POINT_SIZE, 31
DL_RESTORE_CONTEXT, 31
DL_RETURN, 31
DL_SAVE_CONTEXT, 31
DL_SCISSOR_SIZE, 31
DL_SCISSOR_XY, 31
DL_STENCIL_FUNC, 31
DL_STENCIL_MASK, 31
DL_STENCIL_OP, 32
DL_TAG_MASK, 32
DL_TAG, 32
DL_VERTEX2I, 32
DL_VERTEX2F, 32
DLSWAP_DONE, 32
DLSWAP_FRAME, 32
DLSWAP_LINE, 32
DST_ALPHA, 32
delay_ms, 50
delay_us, 51
EDGE_STRIP_A, 32
EDGE_STRIP_B, 33
EDGE_STRIP_L, 33
EDGE_STRIP_R, 33
EQUAL, 33
F16, 33
FT800_ACTIVE, 33
FT800_CLK36M, 33
FT800_CLK48M, 33
FT800_CLKEXT, 33
FT800_CORERST, 34
FT800_GPUACTIVE, 34
FT800_PWRDOWN, 34
FT800_SLEEP, 34
FT800_STANDBY, 34
FT800_VERSION, 34
FT_CMD_FIFO_SIZE, 34
FT_CMD_SIZE, 34
FT_DL_SIZE, 35
FTPOINTS, 35
ft800cmdWrite, 51
ft800memRead16, 52
ft800memRead32, 53
ft800memRead8, 53
ft800memWrite16, 54
ft800memWrite32, 55
ft800memWrite8, 56
GEQUAL, 35
GREATER, 35
GREEN, 35
getData, 57
INCR_WRAP, 35
INCR, 35
INT_CMDEEMPTY, 35
INT_CMDFLAG, 35
INT_CONVCOMPLETE, 36
INT_PLAYBACK, 36
INT_SOUND, 36
INT_SWAP, 36
INT_TAG, 36
INT_TOUCH, 36
INVALID_TOUCH_XY, 36
INVERT, 36
incCMDOffset, 57
KEEP, 36
L1, 36
L4, 37
L8, 37
LCD_QVGA, 37
LEQUAL, 37
LESS, 37
LINE_STRIP, 37
LINEAR_SAMPLES, 37
LINES, 37
MAX, 37
MEM_READ, 37
MEM_WRITE, 38
MIN, 38
NEAREST, 38
NEVER, 38
NOTEQUAL, 38
NOTE, 38
ONE_MINUS_DST_ALPHA, 38
ONE_MINUS_SRC_ALPHA, 38
ONE, 38
OPT_CENTERX, 39
OPT_CENTERY, 39
OPT_CENTER, 39
OPT_FLAT, 39
OPT_MONO, 39
OPT_NOBACK, 39
OPT_NODL, 39
OPT_NOHANDS, 39
OPT_NOHM, 39
OPT_NOPOINTER, 39
OPT_NOSECS, 40
OPT_NOTICKS, 40
OPT_RIGHTX, 40

OPT_SIGNED, [40](#)
PALETTERED, [40](#)
PLAYCOLOR, [40](#)
RAM_CMD, [40](#)
RAM_DL, [40](#)
RAM_PAL, [40](#)
RAM_REG, [41](#)
RAM_G, [40](#)
RECTS, [41](#)
REG_CLOCK, [41](#)
REG_CMD_DL, [41](#)
REG_CMD_READ, [41](#)
REG_CMD_WRITE, [41](#)
REG_CPURESET, [41](#)
REG_CSPREAD, [41](#)
REG_DITHER, [41](#)
REG_DLSWAP, [42](#)
REG_FRAMES, [42](#)
REG_FREQUENCY, [42](#)
REG_GPIO_DIR, [42](#)
REG_GPIO, [42](#)
REG_HCYCLE, [42](#)
REG_HOFFSET, [42](#)
REG_HSIZE, [42](#)
REG_HSYNC0, [42](#)
REG_HSYNC1, [42](#)
REG_INT_EN, [43](#)
REG_INT_FLAGS, [43](#)
REG_INT_MASK, [43](#)
REG_ID, [43](#)
REG_MACRO_0, [43](#)
REG_MACRO_1, [43](#)
REG_OUTBITS, [43](#)
REG_PCLK_POL, [43](#)
REG_PCLK, [43](#)
REG_PLAYBACK_FORMAT, [44](#)
REG_PLAYBACK_FREQ, [44](#)
REG_PLAYBACK_LENGTH, [44](#)
REG_PLAYBACK_LOOP, [44](#)
REG_PLAYBACK_PLAY, [44](#)
REG_PLAYBACK_READPTR, [44](#)
REG_PLAYBACK_START, [44](#)
REG_PLAY, [43](#)
REG_PWM_DUTY, [44](#)
REG_PWM_HZ, [44](#)
REG_RENDERMODE, [44](#)
REG_ROTATE, [45](#)
REG_SNAPSHOT, [45](#)
REG_SNAPY, [45](#)
REG_SOUND, [45](#)
REG_SWIZZLE, [45](#)
REG_TAG_X, [45](#)
REG_TAG_Y, [45](#)
REG_TAP_CRC, [45](#)
REG_TAP_MASK, [45](#)
REG_TAG, [45](#)
REG_TOUCH_ADC_MODE, [46](#)
REG_TOUCH_CHARGE, [46](#)
REG_TOUCH_DIRECT_XY, [46](#)
REG_TOUCH_DIRECT_Z1Z2, [46](#)
REG_TOUCH_MODE, [46](#)
REG_TOUCH_OVERSAMPLE, [46](#)
REG_TOUCH_RAW_XY, [46](#)
REG_TOUCH_RZTHRESH, [46](#)
REG_TOUCH_RZ, [46](#)
REG_TOUCH_SCREEN_XY, [46](#)
REG_TOUCH_SETTLE, [47](#)
REG_TOUCH_TAG_XY, [47](#)
REG_TOUCH_TAG, [47](#)
REG_TOUCH_TRANSFORM_A, [47](#)
REG_TOUCH_TRANSFORM_B, [47](#)
REG_TOUCH_TRANSFORM_C, [47](#)
REG_TOUCH_TRANSFORM_D, [47](#)
REG_TOUCH_TRANSFORM_E, [47](#)
REG_TOUCH_TRANSFORM_F, [47](#)
REG_TRACKER, [47](#)
REG_VCYCLE, [48](#)
REG_VOFFSET, [48](#)
REG_VOL_PB, [48](#)
REG_VOL_SOUND, [48](#)
REG_VSIZE, [48](#)
REG_VSYNC0, [48](#)
REG_VSYNC1, [48](#)
REPEAT, [48](#)
REPLACE, [48](#)
RED, [41](#)
RGB332, [49](#)
RGB565, [49](#)
RGB, [48](#)
SRC_ALPHA, [49](#)
sendData, [58](#)
SQ, [49](#)
TEXT8X8, [49](#)
TEXTVGA, [49](#)
TOUCHMODE_CONTINUOUS, [49](#)
TOUCHMODE_FRAME, [49](#)
TOUCHMODE_OFF, [49](#)
TOUCHMODE_ONESHOT, [49](#)
ULAW_SAMPLES, [50](#)
WHITE, [50](#)
xCS, [50](#)
xPD, [50](#)
xSDI, [50](#)
xSDO, [50](#)
xclock, [50](#)
ZERO, [50](#)
FT800_ACTIVE
 FT800.h, [33](#)
FT800_CLK36M
 FT800.h, [33](#)
FT800_CLK48M
 FT800.h, [33](#)
FT800_CLKEXT
 FT800.h, [33](#)
FT800_CORERST
 FT800.h, [34](#)

FT800_GPUACTIVE
 FT800.h, 34
 FT800_PWRDOWN
 FT800.h, 34
 FT800_SLEEP
 FT800.h, 34
 FT800_STANDBY
 FT800.h, 34
 FT800_VERSION
 FT800.h, 34
 FT800api.cpp, 59
 autko, 61
 button, 61
 calibrate, 62
 dot, 63
 initScreen, 63
 line, 64
 mainScreen, 65
 number, 65
 optionsScreen, 66
 show, 66
 smartMirrorScreen, 67
 spinner, 68
 start, 69
 text, 70
 FT800api.h, 71
 audi, 82
 autko, 73
 button, 73
 calibrate, 74
 cmdBufferRd, 82
 cmdBufferWr, 82
 cmdOffset, 82
 dot, 75
 initScreen, 75
 line, 76
 mainScreen, 77
 number, 77
 optionsScreen, 78
 show, 78
 smartMirrorScreen, 79
 spinner, 79
 start, 80
 text, 81
 timeR, 82
 track, 82
 FT_CMD_FIFO_SIZE
 FT800.h, 34
 FT_CMD_SIZE
 FT800.h, 34
 FT_DL_SIZE
 FT800.h, 35
 FTPOINTS
 FT800.h, 35
 ft800cmdWrite
 FT800.cpp, 9
 FT800.h, 51
 ft800memRead16
 FT800.cpp, 9
 FT800.h, 52
 ft800memRead32
 FT800.cpp, 10
 FT800.h, 53
 ft800memRead8
 FT800.cpp, 11
 FT800.h, 53
 ft800memWrite16
 FT800.cpp, 11
 FT800.h, 54
 ft800memWrite32
 FT800.cpp, 12
 FT800.h, 55
 ft800memWrite8
 FT800.cpp, 13
 FT800.h, 56

 GEQUAL
 FT800.h, 35
 GREATER
 FT800.h, 35
 GREEN
 FT800.h, 35
 getData
 FT800.cpp, 14
 FT800.h, 57

 I2C.cpp, 82
 d, 84
 readPCF, 83
 I2C.h, 84
 pinInt0, 86
 readPCF, 86
 scl, 86
 sda, 86
 INCR_WRAP
 FT800.h, 35
 INCR
 FT800.h, 35
 INT_CMDEMPTY
 FT800.h, 35
 INT_CMDFLAG
 FT800.h, 35
 INT_CONVCOMPLETE
 FT800.h, 36
 INT_PLAYBACK
 FT800.h, 36
 INT_SOUND
 FT800.h, 36
 INT_SWAP
 FT800.h, 36
 INT_TAG
 FT800.h, 36
 INT_TOUCH
 FT800.h, 36
 INVALID_TOUCH_XY
 FT800.h, 36
 INVERT

- FT800.h, [36](#)
- incCMDOffset
 - FT800.cpp, [14](#)
 - FT800.h, [57](#)
- initScreen
 - FT800api.cpp, [63](#)
 - FT800api.h, [75](#)
- KEEP
 - FT800.h, [36](#)
- L1
 - FT800.h, [36](#)
- L4
 - FT800.h, [37](#)
- L8
 - FT800.h, [37](#)
- LCD_QVGA
 - FT800.h, [37](#)
- LEQUAL
 - FT800.h, [37](#)
- LESS
 - FT800.h, [37](#)
- LINE_STRIP
 - FT800.h, [37](#)
- LINEAR_SAMPLES
 - FT800.h, [37](#)
- LINES
 - FT800.h, [37](#)
- lights
 - car, [5](#)
- line
 - FT800api.cpp, [64](#)
 - FT800api.h, [76](#)
- MAX
 - FT800.h, [37](#)
- MEM_READ
 - FT800.h, [37](#)
- MEM_WRITE
 - FT800.h, [38](#)
- MIN
 - FT800.h, [38](#)
- mainScreen
 - FT800api.cpp, [65](#)
 - FT800api.h, [77](#)
- NEAREST
 - FT800.h, [38](#)
- NEVER
 - FT800.h, [38](#)
- NOTEQUAL
 - FT800.h, [38](#)
- NOTE
 - FT800.h, [38](#)
- number
 - FT800api.cpp, [65](#)
 - FT800api.h, [77](#)
- ONE_MINUS_DST_ALPHA
 - FT800.h, [38](#)
- ONE_MINUS_SRC_ALPHA
 - FT800.h, [38](#)
- ONE
 - FT800.h, [38](#)
- OPT_CENTERX
 - FT800.h, [39](#)
- OPT_CENTERY
 - FT800.h, [39](#)
- OPT_CENTER
 - FT800.h, [39](#)
- OPT_FLAT
 - FT800.h, [39](#)
- OPT_MONO
 - FT800.h, [39](#)
- OPT_NOBACK
 - FT800.h, [39](#)
- OPT_NODL
 - FT800.h, [39](#)
- OPT_NOHANDS
 - FT800.h, [39](#)
- OPT_NOHM
 - FT800.h, [39](#)
- OPT_NOPOINTER
 - FT800.h, [39](#)
- OPT_NOSECS
 - FT800.h, [40](#)
- OPT_NOTICKS
 - FT800.h, [40](#)
- OPT_RIGHTX
 - FT800.h, [40](#)
- OPT_SIGNED
 - FT800.h, [40](#)
- optionsScreen
 - FT800api.cpp, [66](#)
 - FT800api.h, [78](#)
- PALETTED
 - FT800.h, [40](#)
- PLAYCOLOR
 - FT800.h, [40](#)
- pinInt0
 - I2C.h, [86](#)
- printObj
 - simulator.cpp, [88](#)
 - simulator.h, [93](#)
- r
 - car, [6](#)
- RAM_CMD
 - FT800.h, [40](#)
- RAM_DL
 - FT800.h, [40](#)
- RAM_PAL
 - FT800.h, [40](#)
- RAM_REG
 - FT800.h, [41](#)
- RAM_G
 - FT800.h, [40](#)

RECTS
 FT800.h, [41](#)
REG_CLOCK
 FT800.h, [41](#)
REG_CMD_DL
 FT800.h, [41](#)
REG_CMD_READ
 FT800.h, [41](#)
REG_CMD_WRITE
 FT800.h, [41](#)
REG_CPURESET
 FT800.h, [41](#)
REG_CSPREAD
 FT800.h, [41](#)
REG_DITHER
 FT800.h, [41](#)
REG_DLSWAP
 FT800.h, [42](#)
REG_FRAMES
 FT800.h, [42](#)
REG_FREQUENCY
 FT800.h, [42](#)
REG_GPIO_DIR
 FT800.h, [42](#)
REG_GPIO
 FT800.h, [42](#)
REG_HCYCLE
 FT800.h, [42](#)
REG_HOFFSET
 FT800.h, [42](#)
REG_HSIZE
 FT800.h, [42](#)
REG_HSYNC0
 FT800.h, [42](#)
REG_HSYNC1
 FT800.h, [42](#)
REG_INT_EN
 FT800.h, [43](#)
REG_INT_FLAGS
 FT800.h, [43](#)
REG_INT_MASK
 FT800.h, [43](#)
REG_ID
 FT800.h, [43](#)
REG_MACRO_0
 FT800.h, [43](#)
REG_MACRO_1
 FT800.h, [43](#)
REG_OUTBITS
 FT800.h, [43](#)
REG_PCLK_POL
 FT800.h, [43](#)
REG_PCLK
 FT800.h, [43](#)
REG_PLAYBACK_FORMAT
 FT800.h, [44](#)
REG_PLAYBACK_FREQ
 FT800.h, [44](#)
REG_PLAYBACK_LENGTH
 FT800.h, [44](#)
REG_PLAYBACK_LOOP
 FT800.h, [44](#)
REG_PLAYBACK_PLAY
 FT800.h, [44](#)
REG_PLAYBACK_READPTR
 FT800.h, [44](#)
REG_PLAYBACK_START
 FT800.h, [44](#)
REG_PLAY
 FT800.h, [43](#)
REG_PWM_DUTY
 FT800.h, [44](#)
REG_PWM_HZ
 FT800.h, [44](#)
REG_RENDERMODE
 FT800.h, [44](#)
REG_ROTATE
 FT800.h, [45](#)
REG_SNAPSHOT
 FT800.h, [45](#)
REG_SNAPY
 FT800.h, [45](#)
REG_SOUND
 FT800.h, [45](#)
REG_SWIZZLE
 FT800.h, [45](#)
REG_TAG_X
 FT800.h, [45](#)
REG_TAG_Y
 FT800.h, [45](#)
REG_TAP_CRC
 FT800.h, [45](#)
REG_TAP_MASK
 FT800.h, [45](#)
REG_TAG
 FT800.h, [45](#)
REG_TOUCH_ADC_MODE
 FT800.h, [46](#)
REG_TOUCH_CHARGE
 FT800.h, [46](#)
REG_TOUCH_DIRECT_XY
 FT800.h, [46](#)
REG_TOUCH_DIRECT_Z1Z2
 FT800.h, [46](#)
REG_TOUCH_MODE
 FT800.h, [46](#)
REG_TOUCH_OVERSAMPLE
 FT800.h, [46](#)
REG_TOUCH_RAW_XY
 FT800.h, [46](#)
REG_TOUCH_RZTHRESH
 FT800.h, [46](#)
REG_TOUCH_RZ
 FT800.h, [46](#)
REG_TOUCH_SCREEN_XY
 FT800.h, [46](#)

REG_TOUCH_SETTLE
 FT800.h, [47](#)
REG_TOUCH_TAG_XY
 FT800.h, [47](#)
REG_TOUCH_TAG
 FT800.h, [47](#)
REG_TOUCH_TRANSFORM_A
 FT800.h, [47](#)
REG_TOUCH_TRANSFORM_B
 FT800.h, [47](#)
REG_TOUCH_TRANSFORM_C
 FT800.h, [47](#)
REG_TOUCH_TRANSFORM_D
 FT800.h, [47](#)
REG_TOUCH_TRANSFORM_E
 FT800.h, [47](#)
REG_TOUCH_TRANSFORM_F
 FT800.h, [47](#)
REG_TRACKER
 FT800.h, [47](#)
REG_VCYCLE
 FT800.h, [48](#)
REG_VOFFSET
 FT800.h, [48](#)
REG_VOL_PB
 FT800.h, [48](#)
REG_VOL_SOUND
 FT800.h, [48](#)
REG_VSIZE
 FT800.h, [48](#)
REG_VSYNC0
 FT800.h, [48](#)
REG_VSYNC1
 FT800.h, [48](#)
REPEAT
 FT800.h, [48](#)
REPLACE
 FT800.h, [48](#)
RED
 FT800.h, [41](#)
RGB332
 FT800.h, [49](#)
RGB565
 FT800.h, [49](#)
RGB
 FT800.h, [48](#)
readData
 simulator.cpp, [89](#)
 simulator.h, [94](#)
readPCF
 I2C.cpp, [83](#)
 I2C.h, [86](#)
readTemp
 simulator.cpp, [89](#)
 simulator.h, [94](#)
SRC_ALPHA
 FT800.h, [49](#)
save
 simulator.cpp, [90](#)
 simulator.h, [95](#)
saveData
 simulator.h, [96](#)
scl
 I2C.h, [86](#)
screenNR
 simulator.h, [96](#)
sda
 I2C.h, [86](#)
seatbelts
 car, [6](#)
sendData
 FT800.cpp, [15](#)
 FT800.h, [58](#)
 simulator.cpp, [90](#)
 simulator.h, [95](#)
show
 FT800api.cpp, [66](#)
 FT800api.h, [78](#)
simulator.cpp, [87](#)
 checkChangesAnalog, [88](#)
 checkChangesDigital, [88](#)
 printObj, [88](#)
 readData, [89](#)
 readTemp, [89](#)
 save, [90](#)
 sendData, [90](#)
simulator.h, [91](#)
 audi, [96](#)
 checkChangesAnalog, [93](#)
 checkChangesDigital, [93](#)
 dataFormat, [96](#)
 printObj, [93](#)
 readData, [94](#)
 readTemp, [94](#)
 save, [95](#)
 saveData, [96](#)
 screenNR, [96](#)
 sendData, [95](#)
smartMirrorScreen
 FT800api.cpp, [67](#)
 FT800api.h, [79](#)
spinner
 FT800api.cpp, [68](#)
 FT800api.h, [79](#)
SQ
 FT800.h, [49](#)
start
 FT800api.cpp, [69](#)
 FT800api.h, [80](#)
TEXT8X8
 FT800.h, [49](#)
TEXTVGA
 FT800.h, [49](#)
TOUCHMODE_CONTINUOUS
 FT800.h, [49](#)
TOUCHMODE_FRAME

- FT800.h, [49](#)
- TOUCHMODE_OFF
 - FT800.h, [49](#)
- TOUCHMODE_ONESHOT
 - FT800.h, [49](#)
- tempEngine
 - car, [6](#)
- tempIn
 - car, [6](#)
- tempOut
 - car, [6](#)
- text
 - FT800api.cpp, [70](#)
 - FT800api.h, [81](#)
- timeR
 - FT800api.h, [82](#)
- track
 - FT800api.h, [82](#)
- ULAW_SAMPLES
 - FT800.h, [50](#)
- WHITE
 - FT800.h, [50](#)
- xCS
 - FT800.h, [50](#)
- xPD
 - FT800.h, [50](#)
- xSDI
 - FT800.h, [50](#)
- xSDO
 - FT800.h, [50](#)
- xclock
 - FT800.h, [50](#)
- ZERO
 - FT800.h, [50](#)