

## Code documentation

1

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b><a href="#">Class Index</a></b>	<b>1</b>
1.1	<a href="#">Class List</a> . . . . .	1
<b>2</b>	<b><a href="#">File Index</a></b>	<b>3</b>
2.1	<a href="#">File List</a> . . . . .	3
<b>3</b>	<b><a href="#">Class Documentation</a></b>	<b>5</b>
3.1	<a href="#">car Struct Reference</a> . . . . .	5
3.1.1	<a href="#">Detailed Description</a> . . . . .	5
3.1.2	<a href="#">Member Data Documentation</a> . . . . .	5
3.1.2.1	<a href="#">doors</a> . . . . .	5
3.1.2.2	<a href="#">lights</a> . . . . .	6
3.1.2.3	<a href="#">r</a> . . . . .	6
3.1.2.4	<a href="#">seatbelts</a> . . . . .	6
3.1.2.5	<a href="#">tempEngine</a> . . . . .	6
3.1.2.6	<a href="#">templn</a> . . . . .	6
3.1.2.7	<a href="#">tempOut</a> . . . . .	6

<b>4 File Documentation</b>	<b>7</b>
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)	10
4.1.2.5 ft800memRead32(unsigned long ftAddress)	11
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)	13
4.1.2.9 ft800memWrite8(unsigned long ftAddress, unsigned char ftData8)	14
4.1.2.10 getData()	15
4.1.2.11 incCMDOffset(unsigned int currentOffset, unsigned char commandSize)	15
4.1.2.12 sendData(int data)	16
4.2 FT800.h File Reference	17
4.2.1 Detailed Description	24
4.2.2 Macro Definition Documentation	24
4.2.2.1 ABS	24
4.2.2.2 BLACK	24
4.2.2.3 BLUE	24
4.2.2.4 CLR_COL	24
4.2.2.5 CLR_STN	24
4.2.2.6 CLR_TAG	24
4.2.2.7 CMD_APPEND	24
4.2.2.8 CMD_BGCOLOR	25
4.2.2.9 CMD_BUTTON	25
4.2.2.10 CMD_CALIBRATE	25
4.2.2.11 CMD_CLOCK	25

4.2.2.12	CMD_COLDSTART	25
4.2.2.13	CMD_DIAL	25
4.2.2.14	CMD_DLSTART	25
4.2.2.15	CMD_FGCOLOR	25
4.2.2.16	CMD_GAUGE	25
4.2.2.17	CMD_GETMATRIX	25
4.2.2.18	CMD_GETPTR	26
4.2.2.19	CMD_GRADCOLOR	26
4.2.2.20	CMD_GRADIENT	26
4.2.2.21	CMD_INFLATE	26
4.2.2.22	CMD_INTERRUPT	26
4.2.2.23	CMD_KEYS	26
4.2.2.24	CMD_LOADIDENTITY	26
4.2.2.25	CMD_LOADIMAGE	26
4.2.2.26	CMD_LOGO	26
4.2.2.27	CMD_MEMCPY	26
4.2.2.28	CMD_MEMCRC	27
4.2.2.29	CMD_MEMSET	27
4.2.2.30	CMD_MEMWRITE	27
4.2.2.31	CMD_MEMZERO	27
4.2.2.32	CMD_NUMBER	27
4.2.2.33	CMD_PROGRESS	27
4.2.2.34	CMD_REGREAD	27
4.2.2.35	CMD_ROTATE	27
4.2.2.36	CMD_SCALE	27
4.2.2.37	CMD_SCREENSAVER	27
4.2.2.38	CMD_SCROLLBAR	28
4.2.2.39	CMD_SETFONT	28
4.2.2.40	CMD_SETMATRIX	28
4.2.2.41	CMD_SKETCH	28

4.2.2.42	CMD_SLIDER	28
4.2.2.43	CMD_SNAPSHOT	28
4.2.2.44	CMD_SPINNER	28
4.2.2.45	CMD_STOP	28
4.2.2.46	CMD_SWAP	28
4.2.2.47	CMD_TEXT	28
4.2.2.48	CMD_TOGGLE	29
4.2.2.49	CMD_TRACK	29
4.2.2.50	CMD_TRANSLATE	29
4.2.2.51	CMDBUF_SIZE	29
4.2.2.52	DECR	29
4.2.2.53	DECR_WRAP	29
4.2.2.54	DL_ALPHA_FUNC	29
4.2.2.55	DL_BEGIN	29
4.2.2.56	DL_BITMAP_HANDLE	29
4.2.2.57	DL_BITMAP_LAYOUT	29
4.2.2.58	DL_BITMAP_SIZE	30
4.2.2.59	DL_BITMAP_SOURCE	30
4.2.2.60	DL_BITMAP_TFORM_A	30
4.2.2.61	DL_BITMAP_TFORM_B	30
4.2.2.62	DL_BITMAP_TFORM_C	30
4.2.2.63	DL_BITMAP_TFORM_D	30
4.2.2.64	DL_BITMAP_TFORM_E	30
4.2.2.65	DL_BITMAP_TFORM_F	30
4.2.2.66	DL_BLEND_FUNC	30
4.2.2.67	DL_CALL	30
4.2.2.68	DL_CELL	31
4.2.2.69	DL_CLEAR	31
4.2.2.70	DL_CLEAR_RGB	31
4.2.2.71	DL_CLEAR_STENCIL	31

4.2.2.72	DL_CLEAR_TAG	31
4.2.2.73	DL_COLOR_A	31
4.2.2.74	DL_COLOR_MASK	31
4.2.2.75	DL_COLOR_RGB	31
4.2.2.76	DL_DISPLAY	31
4.2.2.77	DL_END	31
4.2.2.78	DL_JUMP	32
4.2.2.79	DL_LINE_WIDTH	32
4.2.2.80	DL_MACRO	32
4.2.2.81	DL_POINT_SIZE	32
4.2.2.82	DL_RESTORE_CONTEXT	32
4.2.2.83	DL_RETURN	32
4.2.2.84	DL_SAVE_CONTEXT	32
4.2.2.85	DL_SCISSOR_SIZE	32
4.2.2.86	DL_SCISSOR_XY	32
4.2.2.87	DL_STENCIL_FUNC	32
4.2.2.88	DL_STENCIL_MASK	33
4.2.2.89	DL_STENCIL_OP	33
4.2.2.90	DL_TAG	33
4.2.2.91	DL_TAG_MASK	33
4.2.2.92	DL_VERTEX2F	33
4.2.2.93	DL_VERTEX2I	33
4.2.2.94	DLSWAP_DONE	33
4.2.2.95	DLSWAP_FRAME	33
4.2.2.96	DLSWAP_LINE	33
4.2.2.97	DST_ALPHA	33
4.2.2.98	EDGE_STRIP_A	34
4.2.2.99	EDGE_STRIP_B	34
4.2.2.100	EDGE_STRIP_L	34
4.2.2.101	EDGE_STRIP_R	34

4.2.2.102 EQUAL . . . . .	34
4.2.2.103 F16 . . . . .	34
4.2.2.104 FT800_ACTIVE . . . . .	34
4.2.2.105 FT800_CLK36M . . . . .	34
4.2.2.106 FT800_CLK48M . . . . .	34
4.2.2.107 FT800_CLKEXT . . . . .	35
4.2.2.108 FT800_CORERST . . . . .	35
4.2.2.109 FT800_GPUACTIVE . . . . .	35
4.2.2.110 FT800_PWRDOWN . . . . .	35
4.2.2.111 FT800_SLEEP . . . . .	35
4.2.2.112 FT800_STANDBY . . . . .	35
4.2.2.113 FT800_VERSION . . . . .	35
4.2.2.114 FT_CMD_FIFO_SIZE . . . . .	35
4.2.2.115 FT_CMD_SIZE . . . . .	36
4.2.2.116 FT_DL_SIZE . . . . .	36
4.2.2.117 FTPOINTS . . . . .	36
4.2.2.118 GEQUAL . . . . .	36
4.2.2.119 GREATER . . . . .	36
4.2.2.120 GREEN . . . . .	36
4.2.2.121 INCR . . . . .	36
4.2.2.122 INCR_WRAP . . . . .	36
4.2.2.123 INT_CMDEEMPTY . . . . .	36
4.2.2.124 INT_CMDFLAG . . . . .	37
4.2.2.125 INT_CONVCOMPLETE . . . . .	37
4.2.2.126 INT_PLAYBACK . . . . .	37
4.2.2.127 INT_SOUND . . . . .	37
4.2.2.128 INT_SWAP . . . . .	37
4.2.2.129 INT_TAG . . . . .	37
4.2.2.130 INT_TOUCH . . . . .	37
4.2.2.131 INVALID_TOUCH_XY . . . . .	37



4.2.2.132 INVERT . . . . .	37
4.2.2.133 KEEP . . . . .	37
4.2.2.134 L1 . . . . .	38
4.2.2.135 L4 . . . . .	38
4.2.2.136 L8 . . . . .	38
4.2.2.137 LCD_QVGA . . . . .	38
4.2.2.138 LEQUAL . . . . .	38
4.2.2.139 LESS . . . . .	38
4.2.2.140 LINE_STRIP . . . . .	38
4.2.2.141 LINEAR_SAMPLES . . . . .	38
4.2.2.142 LINES . . . . .	38
4.2.2.143 MAX . . . . .	38
4.2.2.144 MEM_READ . . . . .	39
4.2.2.145 MEM_WRITE . . . . .	39
4.2.2.146 MIN . . . . .	39
4.2.2.147 NEAREST . . . . .	39
4.2.2.148 NEVER . . . . .	39
4.2.2.149 NOTE . . . . .	39
4.2.2.150 NOTEQUAL . . . . .	39
4.2.2.151 ONE . . . . .	39
4.2.2.152 ONE_MINUS_DST_ALPHA . . . . .	39
4.2.2.153 ONE_MINUS_SRC_ALPHA . . . . .	40
4.2.2.154 OPT_CENTER . . . . .	40
4.2.2.155 OPT_CENTERX . . . . .	40
4.2.2.156 OPT_CENTERY . . . . .	40
4.2.2.157 OPT_FLAT . . . . .	40
4.2.2.158 OPT_MONO . . . . .	40
4.2.2.159 OPT_NOBACK . . . . .	40
4.2.2.160 OPT_NODL . . . . .	40
4.2.2.161 OPT_NOHANDS . . . . .	40

4.2.2.162 OPT_NOHM . . . . .	40
4.2.2.163 OPT_NOPOINTER . . . . .	41
4.2.2.164 OPT_NOSECS . . . . .	41
4.2.2.165 OPT_NOTICKS . . . . .	41
4.2.2.166 OPT_RIGHTX . . . . .	41
4.2.2.167 OPT_SIGNED . . . . .	41
4.2.2.168 PALETTED . . . . .	41
4.2.2.169 PLAYCOLOR . . . . .	41
4.2.2.170 RAM_CMD . . . . .	41
4.2.2.171 RAM_DL . . . . .	41
4.2.2.172 RAM_G . . . . .	41
4.2.2.173 RAM_PAL . . . . .	42
4.2.2.174 RAM_REG . . . . .	42
4.2.2.175 RECTS . . . . .	42
4.2.2.176 RED . . . . .	42
4.2.2.177 REG_CLOCK . . . . .	42
4.2.2.178 REG_CMD_DL . . . . .	42
4.2.2.179 REG_CMD_READ . . . . .	42
4.2.2.180 REG_CMD_WRITE . . . . .	42
4.2.2.181 REG_CPURESET . . . . .	42
4.2.2.182 REG_CSPREAD . . . . .	42
4.2.2.183 REG_DITHER . . . . .	43
4.2.2.184 REG_DLSWAP . . . . .	43
4.2.2.185 REG_FRAMES . . . . .	43
4.2.2.186 REG_FREQUENCY . . . . .	43
4.2.2.187 REG_GPIO . . . . .	43
4.2.2.188 REG_GPIO_DIR . . . . .	43
4.2.2.189 REG_HCYCLE . . . . .	43
4.2.2.190 REG_HOFFSET . . . . .	43
4.2.2.191 REG_HSIZE . . . . .	43

4.2.2.192 REG_HSYNC0 . . . . .	43
4.2.2.193 REG_HSYNC1 . . . . .	44
4.2.2.194 REG_ID . . . . .	44
4.2.2.195 REG_INT_EN . . . . .	44
4.2.2.196 REG_INT_FLAGS . . . . .	44
4.2.2.197 REG_INT_MASK . . . . .	44
4.2.2.198 REG_MACRO_0 . . . . .	44
4.2.2.199 REG_MACRO_1 . . . . .	44
4.2.2.200 REG_OUTBITS . . . . .	44
4.2.2.201 REG_PCLK . . . . .	44
4.2.2.202 REG_PCLK_POL . . . . .	44
4.2.2.203 REG_PLAY . . . . .	45
4.2.2.204 REG_PLAYBACK_FORMAT . . . . .	45
4.2.2.205 REG_PLAYBACK_FREQ . . . . .	45
4.2.2.206 REG_PLAYBACK_LENGTH . . . . .	45
4.2.2.207 REG_PLAYBACK_LOOP . . . . .	45
4.2.2.208 REG_PLAYBACK_PLAY . . . . .	45
4.2.2.209 REG_PLAYBACK_READPTR . . . . .	45
4.2.2.210 REG_PLAYBACK_START . . . . .	45
4.2.2.211 REG_PWM_DUTY . . . . .	45
4.2.2.212 REG_PWM_HZ . . . . .	45
4.2.2.213 REG_RENDERMODE . . . . .	46
4.2.2.214 REG_ROTATE . . . . .	46
4.2.2.215 REG_SNAPSHOT . . . . .	46
4.2.2.216 REG_SNAPY . . . . .	46
4.2.2.217 REG_SOUND . . . . .	46
4.2.2.218 REG_SWIZZLE . . . . .	46
4.2.2.219 REG_TAG . . . . .	46
4.2.2.220 REG_TAG_X . . . . .	46
4.2.2.221 REG_TAG_Y . . . . .	46

4.2.2.222 REG_TAP_CRC . . . . .	46
4.2.2.223 REG_TAP_MASK . . . . .	47
4.2.2.224 REG_TOUCH_ADC_MODE . . . . .	47
4.2.2.225 REG_TOUCH_CHARGE . . . . .	47
4.2.2.226 REG_TOUCH_DIRECT_XY . . . . .	47
4.2.2.227 REG_TOUCH_DIRECT_Z1Z2 . . . . .	47
4.2.2.228 REG_TOUCH_MODE . . . . .	47
4.2.2.229 REG_TOUCH_OVERSAMPLE . . . . .	47
4.2.2.230 REG_TOUCH_RAW_XY . . . . .	47
4.2.2.231 REG_TOUCH_RZ . . . . .	47
4.2.2.232 REG_TOUCH_RZTHRESH . . . . .	47
4.2.2.233 REG_TOUCH_SCREEN_XY . . . . .	48
4.2.2.234 REG_TOUCH_SETTLE . . . . .	48
4.2.2.235 REG_TOUCH_TAG . . . . .	48
4.2.2.236 REG_TOUCH_TAG_XY . . . . .	48
4.2.2.237 REG_TOUCH_TRANSFORM_A . . . . .	48
4.2.2.238 REG_TOUCH_TRANSFORM_B . . . . .	48
4.2.2.239 REG_TOUCH_TRANSFORM_C . . . . .	48
4.2.2.240 REG_TOUCH_TRANSFORM_D . . . . .	48
4.2.2.241 REG_TOUCH_TRANSFORM_E . . . . .	48
4.2.2.242 REG_TOUCH_TRANSFORM_F . . . . .	48
4.2.2.243 REG_TRACKER . . . . .	49
4.2.2.244 REG_VCYCLE . . . . .	49
4.2.2.245 REG_VOFFSET . . . . .	49
4.2.2.246 REG_VOL_PB . . . . .	49
4.2.2.247 REG_VOL_SOUND . . . . .	49
4.2.2.248 REG_VSIZE . . . . .	49
4.2.2.249 REG_VSYNC0 . . . . .	49
4.2.2.250 REG_VSYNC1 . . . . .	49
4.2.2.251 REPEAT . . . . .	49

4.2.2.252 REPLACE . . . . .	49
4.2.2.253 RGB . . . . .	50
4.2.2.254 RGB332 . . . . .	50
4.2.2.255 RGB565 . . . . .	50
4.2.2.256 SQ . . . . .	50
4.2.2.257 SRC_ALPHA . . . . .	50
4.2.2.258 TEXT8X8 . . . . .	50
4.2.2.259 TEXTVGA . . . . .	50
4.2.2.260 TOUCHMODE_CONTINUOUS . . . . .	50
4.2.2.261 TOUCHMODE_FRAME . . . . .	50
4.2.2.262 TOUCHMODE_OFF . . . . .	50
4.2.2.263 TOUCHMODE_ONESHOT . . . . .	51
4.2.2.264 ULAW_SAMPLES . . . . .	51
4.2.2.265 WHITE . . . . .	51
4.2.2.266 xclock . . . . .	51
4.2.2.267 xCS . . . . .	51
4.2.2.268 xPD . . . . .	51
4.2.2.269 xSDI . . . . .	51
4.2.2.270 xSDO . . . . .	51
4.2.2.271 ZERO . . . . .	51
4.2.3 Function Documentation . . . . .	51
4.2.3.1 delay_ms(int ms) . . . . .	51
4.2.3.2 delay_us(int us) . . . . .	52
4.2.3.3 ft800cmdWrite(unsigned char ftCommand) . . . . .	53
4.2.3.4 ft800memRead16(unsigned long ftAddress) . . . . .	54
4.2.3.5 ft800memRead32(unsigned long ftAddress) . . . . .	55
4.2.3.6 ft800memRead8(unsigned long ftAddress) . . . . .	55
4.2.3.7 ft800memWrite16(unsigned long ftAddress, unsigned int ftData16) . . . . .	56
4.2.3.8 ft800memWrite32(unsigned long ftAddress, unsigned long ftData32) . . . . .	57
4.2.3.9 ft800memWrite8(unsigned long ftAddress, unsigned char ftData8) . . . . .	58

4.2.3.10	<code>getData()</code>	59
4.2.3.11	<code>incCMDOffset(unsigned int currentOffset, unsigned char commandSize)</code>	59
4.2.3.12	<code>sendData(int data)</code>	60
4.3	FT800api.cpp File Reference	61
4.3.1	Detailed Description	62
4.3.2	Function Documentation	63
4.3.2.1	<code>autko()</code>	63
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>	63
4.3.2.3	<code>calibrate()</code>	64
4.3.2.4	<code>dot(unsigned long color, unsigned int point_size, unsigned long point_x, unsigned long point_y)</code>	65
4.3.2.5	<code>initScreen()</code>	65
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>	66
4.3.2.7	<code>mainScreen()</code>	67
4.3.2.8	<code>number(int16_t x, int16_t y, int16_t font, uint16_t options, int value)</code>	67
4.3.2.9	<code>optionsScreen()</code>	68
4.3.2.10	<code>show()</code>	68
4.3.2.11	<code>smartMirrorScreen()</code>	69
4.3.2.12	<code>spinner(int16_t x, int16_t y, uint16_t style, uint16_t scale)</code>	70
4.3.2.13	<code>start(unsigned long color)</code>	70
4.3.2.14	<code>text(int16_t x, int16_t y, int16_t font, uint16_t options, const char *str)</code>	71
4.4	FT800api.h File Reference	72
4.4.1	Detailed Description	74
4.4.2	Function Documentation	74
4.4.2.1	<code>autko()</code>	74
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>	75
4.4.2.3	<code>calibrate()</code>	76
4.4.2.4	<code>dot(unsigned long color, unsigned int point_size, unsigned long point_x, unsigned long point_y)</code>	76
4.4.2.5	<code>initScreen()</code>	77

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>	78
4.4.2.7	<code>mainScreen()</code>	79
4.4.2.8	<code>number(int16_t x, int16_t y, int16_t font, uint16_t options, int32_t value)</code>	80
4.4.2.9	<code>optionsScreen()</code>	80
4.4.2.10	<code>show()</code>	81
4.4.2.11	<code>smartMirrorScreen()</code>	82
4.4.2.12	<code>spinner(int16_t x, int16_t y, uint16_t style, uint16_t scale)</code>	82
4.4.2.13	<code>start(unsigned long color)</code>	83
4.4.2.14	<code>text(int16_t x, int16_t y, int16_t font, uint16_t options, const char *str)</code>	84
4.4.3	Variable Documentation	85
4.4.3.1	<code>audi</code>	85
4.4.3.2	<code>cmdBufferRd</code>	85
4.4.3.3	<code>cmdBufferWr</code>	85
4.4.3.4	<code>cmdOffset</code>	85
4.4.3.5	<code>timeR</code>	85
4.5	I2C.cpp File Reference	86
4.5.1	Detailed Description	86
4.5.2	Function Documentation	86
4.5.2.1	<code>readPCF(char adres)</code>	86
4.5.3	Variable Documentation	87
4.5.3.1	<code>d</code>	87
4.6	I2C.h File Reference	87
4.6.1	Detailed Description	88
4.6.2	Macro Definition Documentation	88
4.6.2.1	<code>pinInt0</code>	88
4.6.2.2	<code>scl</code>	89
4.6.2.3	<code>sda</code>	89
4.6.3	Function Documentation	89
4.6.3.1	<code>readPCF(char adres)</code>	89
4.7	simulator.cpp File Reference	89

4.7.1	Detailed Description	90
4.7.2	Function Documentation	91
4.7.2.1	checkChangesAnalog()	91
4.7.2.2	checkChangesDigital()	91
4.7.2.3	printObj(struct car *obj, char *d)	92
4.7.2.4	readData()	92
4.7.2.5	readTemp(int portNumber)	93
4.7.2.6	save(struct car *audi, struct car *tmp)	94
4.7.2.7	sendData()	94
4.8	simulator.h File Reference	95
4.8.1	Detailed Description	96
4.8.2	Function Documentation	97
4.8.2.1	checkChangesAnalog()	97
4.8.2.2	checkChangesDigital()	97
4.8.2.3	printObj(struct car *obj, char *d)	98
4.8.2.4	readData()	98
4.8.2.5	readTemp(int portNumber)	99
4.8.2.6	save(struct car *audi, struct car *tmp)	100
4.8.2.7	sendData()	100
4.8.3	Variable Documentation	101
4.8.3.1	audi	101
4.8.3.2	dataFormat	101
4.8.3.3	saveData	101
4.8.3.4	screenNR	101
4.9	VM800Galileo.cpp File Reference	102
4.9.1	Function Documentation	103
4.9.1.1	loop()	103
4.9.1.2	setup(void)	103
4.9.2	Variable Documentation	104
4.9.2.1	audi	104



4.9.2.2	buf	104
4.9.2.3	cmdBufferRd	104
4.9.2.4	cmdBufferWr	105
4.9.2.5	cmdOffset	105
4.9.2.6	dataFormat	105
4.9.2.7	ft800Gpio	105
4.9.2.8	lcdHcycle	105
4.9.2.9	lcdHeight	105
4.9.2.10	lcdHoffset	105
4.9.2.11	lcdHsync0	105
4.9.2.12	lcdHsync1	106
4.9.2.13	lcdPclk	106
4.9.2.14	lcdPclkpol	106
4.9.2.15	lcdSwizzle	106
4.9.2.16	lcdVcycle	106
4.9.2.17	lcdVoffset	106
4.9.2.18	lcdVsync0	106
4.9.2.19	lcdVsync1	106
4.9.2.20	lcdWidth	107
4.9.2.21	ramCommandBuffer	107
4.9.2.22	ramDisplayList	107
4.9.2.23	saveData	107
4.9.2.24	screenNR	107
4.9.2.25	timeR	107



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">car</a> . . . . .	<a href="#">5</a>
-------------------------------	-------------------



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">FT800.cpp</a>	File containing declarations of all functions required to use with VM800 . . . . .	7
<a href="#">FT800.h</a>	File containing declarations of all functions required to use with VM800 . . . . .	17
<a href="#">FT800api.cpp</a>	File containing declarations of all API functions for VM800 . . . . .	61
<a href="#">FT800api.h</a>	File containing declarations of all API functions for VM800 . . . . .	72
<a href="#">I2C.cpp</a>	File containing declarations of function to read data with using I2C protocol . . . . .	86
<a href="#">I2C.h</a>	File containing declarations of function to read data with using I2C protocol . . . . .	87
<a href="#">simulator.cpp</a>	File containing declarations of all functions required to communication with car simulator . . .	89
<a href="#">simulator.h</a>	File containing declarations of all functions required to communication with car simulator . . .	95
<a href="#">VM800Galileo.cpp</a>	. . . . .	102



## Chapter 3

# Class Documentation

### 3.1 car Struct Reference

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

#### Public Attributes

- int [doors](#)
- int [seatbelts](#)
- int [lights](#)
- int [r](#)
- float [tempOut](#)
- float [tempIn](#)
- float [tempEngine](#)

#### 3.1.1 Detailed Description

Analog ports \* A0 - temp Out \* A1 - temp In \* A2 - temp Engine \*

A global car structure \*

Definition at line 34 of file simulator.h.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 int car::doors

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

Definition at line 35 of file simulator.h.

#### 3.1.2.2 int car::lights

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

Definition at line 37 of file simulator.h.

#### 3.1.2.3 int car::r

statu of reverse gear

Definition at line 38 of file simulator.h.

#### 3.1.2.4 int car::seatbelts

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

Definition at line 36 of file simulator.h.

#### 3.1.2.5 float car::tempEngine

temperature engine

Definition at line 41 of file simulator.h.

#### 3.1.2.6 float car::tempIn

temperature inside

Definition at line 40 of file simulator.h.

#### 3.1.2.7 float car::tempOut

temperature outside

Definition at line 39 of file simulator.h.

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

- [simulator.h](#)



## Chapter 4

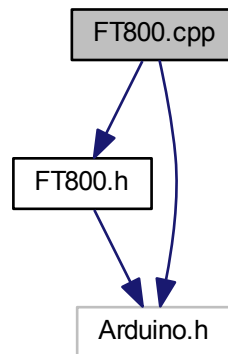
# File Documentation

### 4.1 FT800.cpp File Reference

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

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

Include dependency graph for FT800.cpp:



### Functions

- void [delay\\_us](#) (int us)
- void [delay\\_ms](#) (int ms)
- void [sendData](#) (int data)
- unsigned char [getData](#) ()
- void [ft800memWrite8](#) (unsigned long ftAddress, unsigned char ftData8)
- void [ft800memWrite16](#) (unsigned long ftAddress, unsigned int ftData16)
- void [ft800memWrite32](#) (unsigned long ftAddress, unsigned long ftData32)
- unsigned char [ft800memRead8](#) (unsigned long ftAddress)
- unsigned char [ft800memRead16](#) (unsigned long ftAddress)
- unsigned long [ft800memRead32](#) (unsigned long ftAddress)
- unsigned int [incCMDOffset](#) (unsigned int currentOffset, unsigned char commandSize)
- void [ft800cmdWrite](#) (unsigned char ftCommand)

### 4.1.1 Detailed Description

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

#### Author

Daniel Sienkiewicz

#### Date

28 February 2016

### 4.1.2 Function Documentation

#### 4.1.2.1 void delay\_ms ( int *ms* )

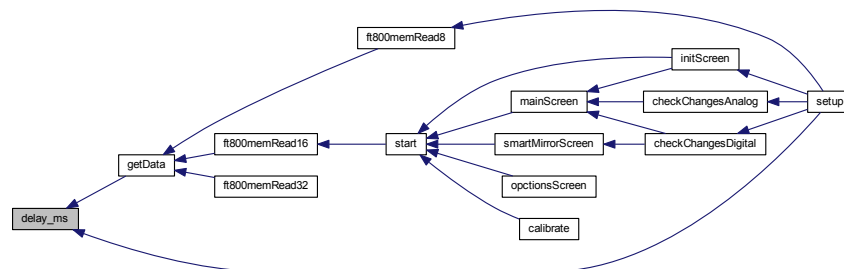
Pauses the program for the amount of time (in milisecond) specified as parameter \*

#### Parameters

<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* )

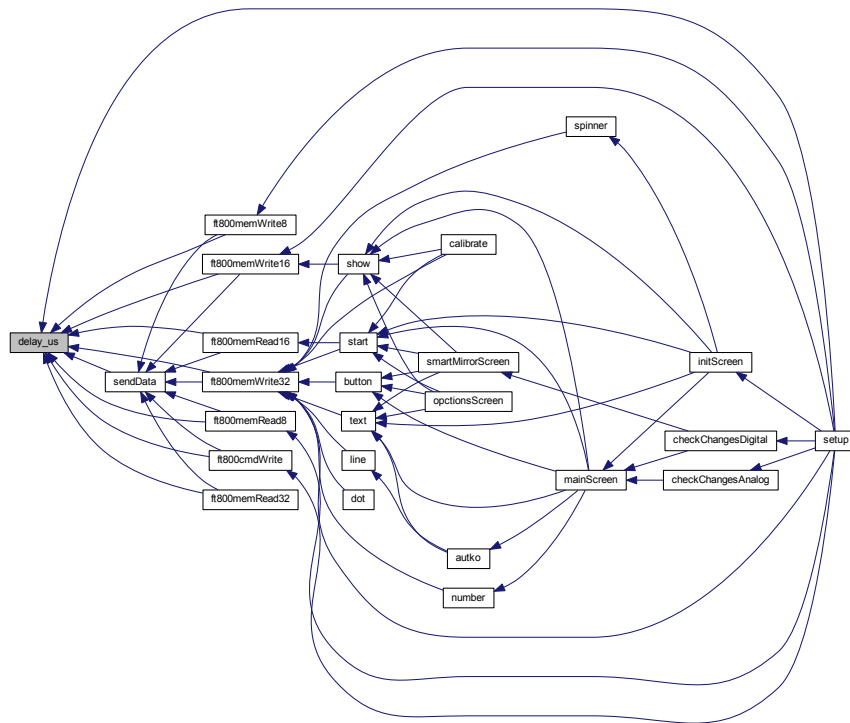
Pauses the program for the amount of time (in microsecond) specified as parameter \*

#### 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* )

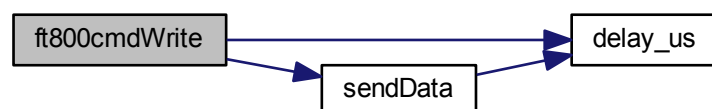
Sends FT800 command \*

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:



Here is the caller graph for this function:



#### 4.1.2.4 unsigned char ft800memRead16 ( unsigned long *ftAddress* )

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

##### Parameters

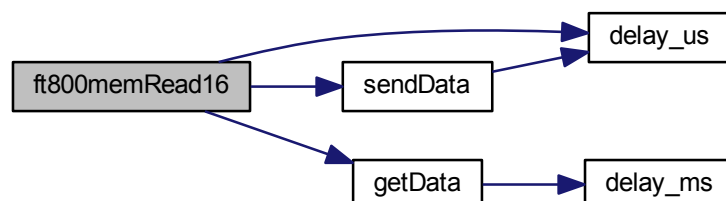
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

##### Returns

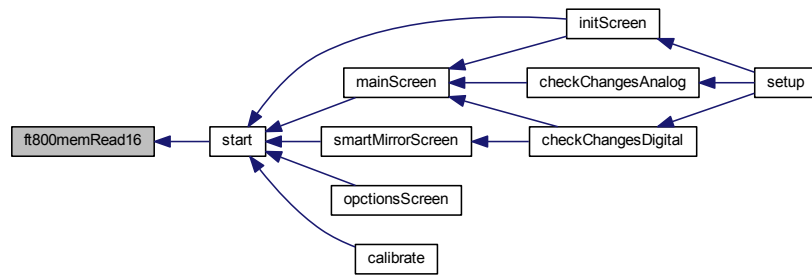
16 bit data obtained from device \*

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* )

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

##### Parameters

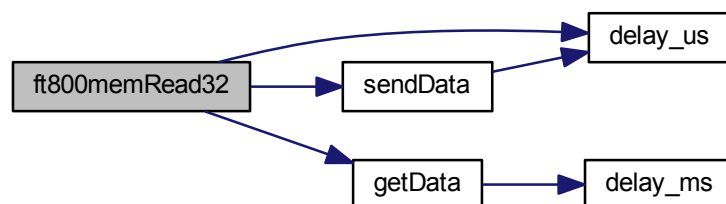
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

##### Returns

32 bit data obtained from device \*

Definition at line 226 of file FT800.cpp.

Here is the call graph for this function:



#### 4.1.2.6 unsigned char ft800memRead8 ( unsigned long *ftAddress* )

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

**Parameters**

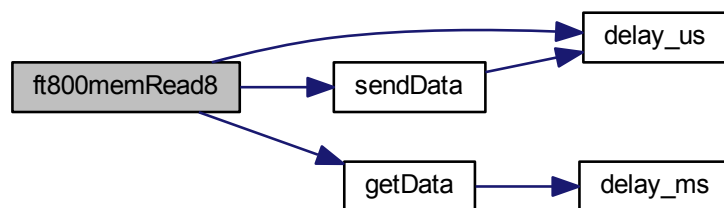
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

**Returns**

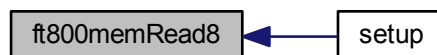
8 bit data obtained from device \*

Definition at line 143 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

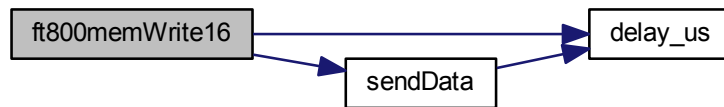
Funtion to send 16 bit value to active device with using SPI interface \*

**Parameters**

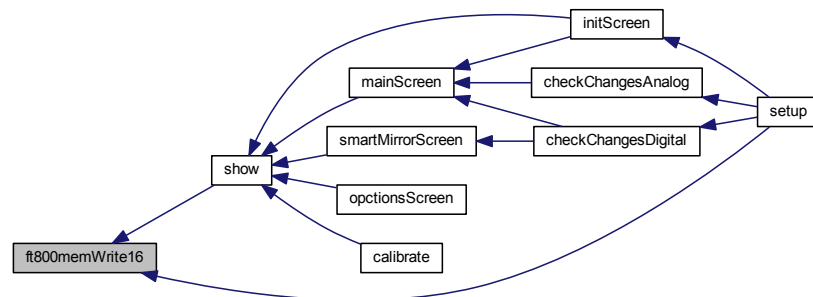
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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* )

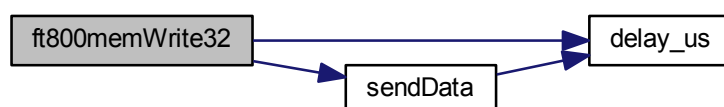
Function to send 32 bit value to active device with using SPI interface \*

##### Parameters

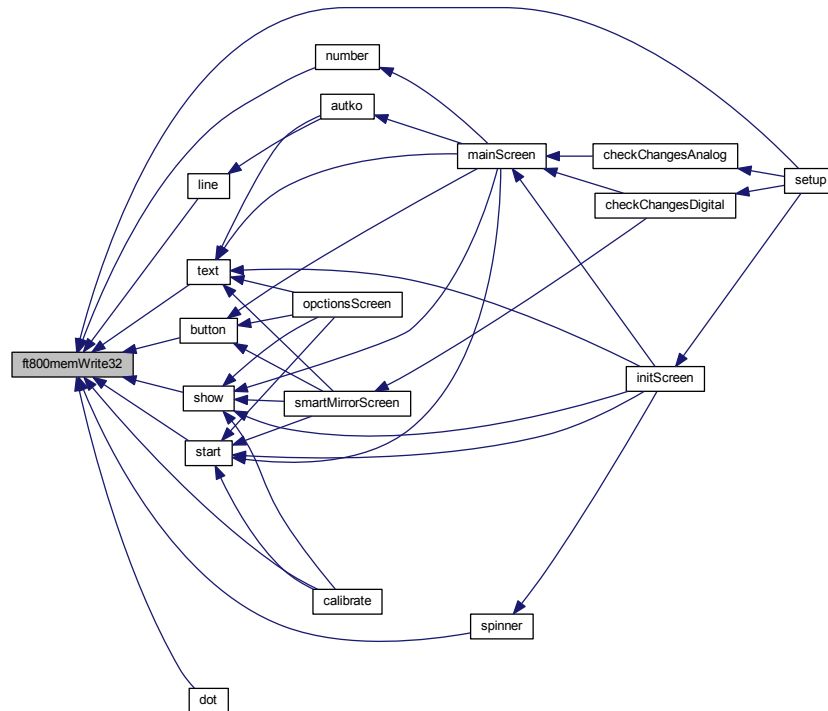
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

Definition at line 105 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

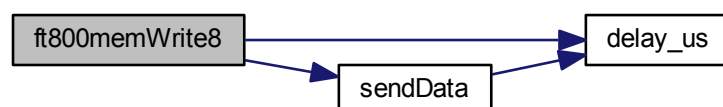
Funtion to send 8 bit value to active device with using SPI interface \*

##### Parameters

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

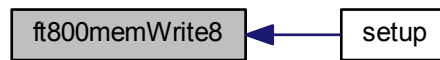
Definition at line 45 of file FT800.cpp.

Here is the call graph for this function:





Here is the caller graph for this function:



#### 4.1.2.10 unsigned char getData ( )

Function getting data from active device with using SPI interface \*

##### Returns

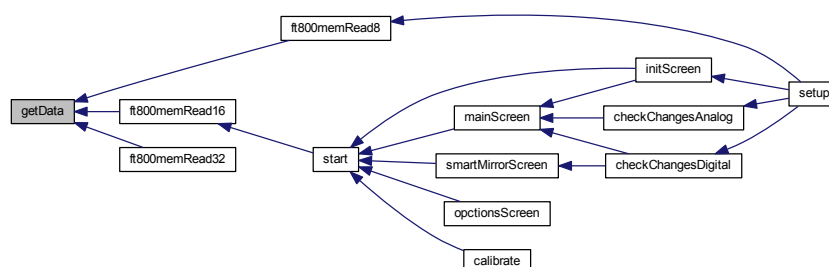
8 bit vvalue with obtained value \*

Definition at line 31 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

## Parameters

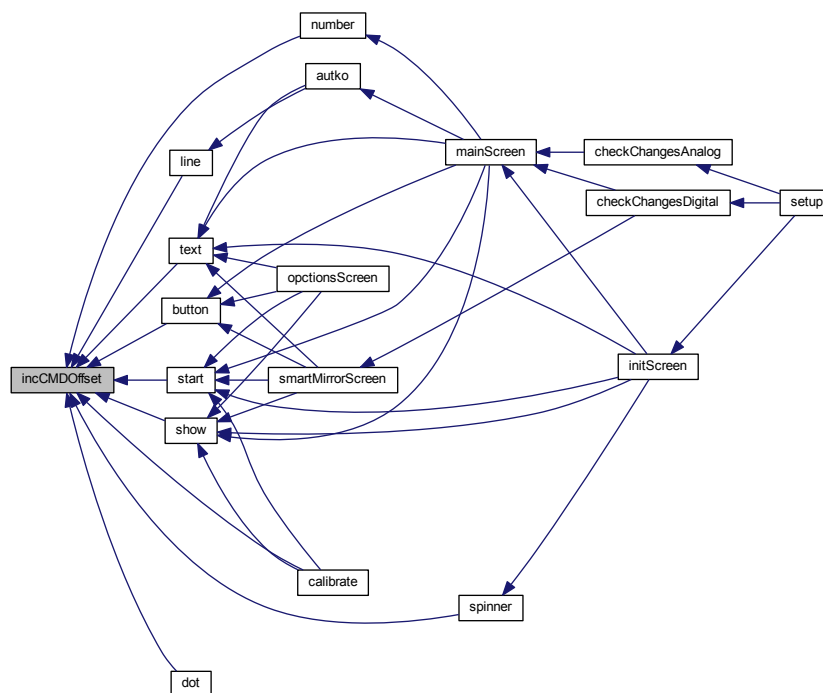
<i>currentOffset</i>	graphics processor command list pointer *
<i>commandSize</i>	number of bytes to increment the offset *

## 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

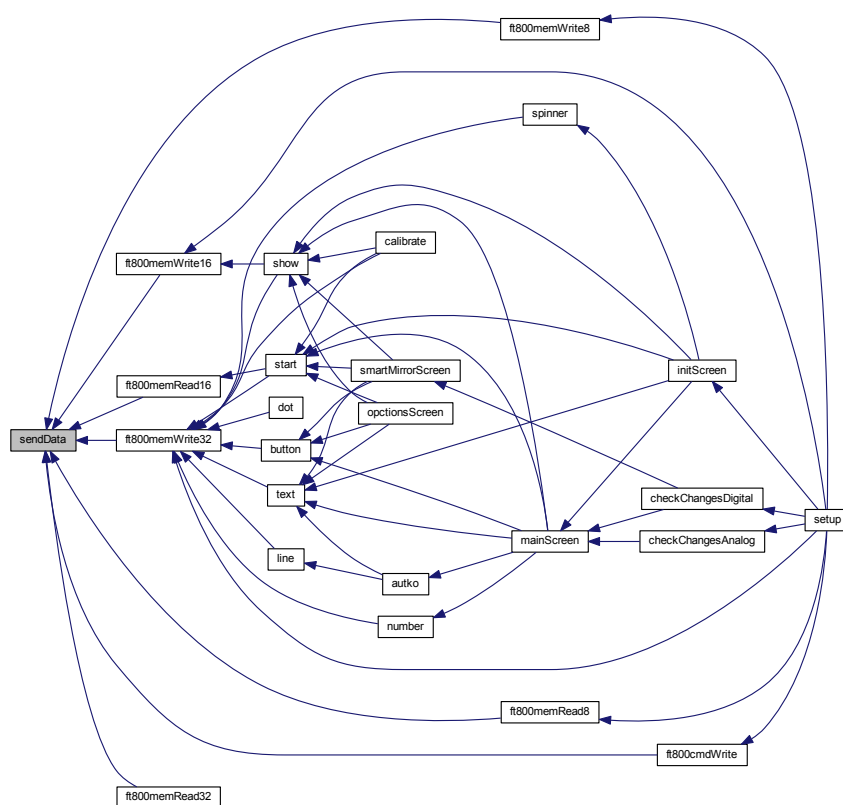
<i>data</i>	8 bit value to send to device *
-------------	---------------------------------

Definition at line 19 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

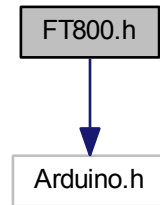


## 4.2 FT800.h File Reference

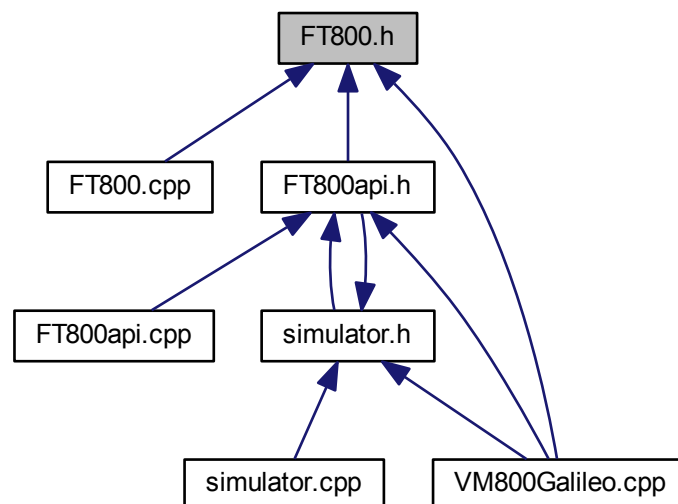
File containing declarations of all functions required to use with VM800.

```
#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 0xffffffff2UL`

Definition at line 146 of file FT800.h.

4.2.2.49 `#define CMD_TRACK 0xffffffff2cUL`

Definition at line 147 of file FT800.h.

4.2.2.50 `#define CMD_TRANSLATE 0xffffffff27UL`

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 )**

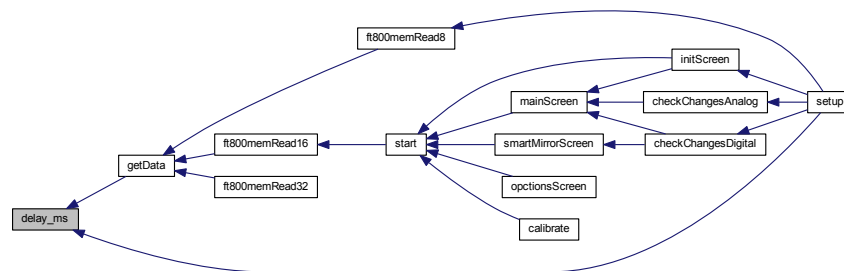
Pauses the program for the amount of time (in milisecond) specified as parameter \*

**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 )**

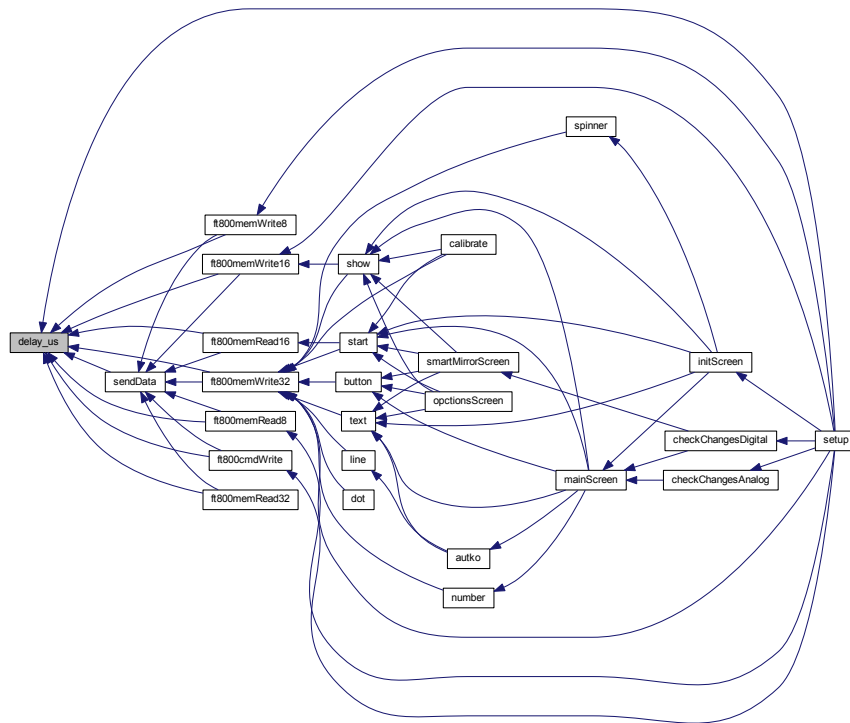
Pauses the program for the amount of time (in microsecond) specified as parameter \*

**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 )

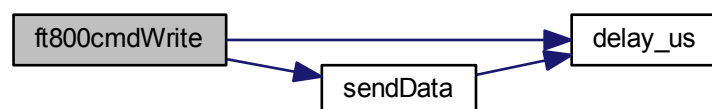
Sends FT800 command \*

##### 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:



Here is the caller graph for this function:



#### 4.2.3.4 unsigned char ft800memRead16 ( unsigned long *ftAddress* )

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

##### Parameters

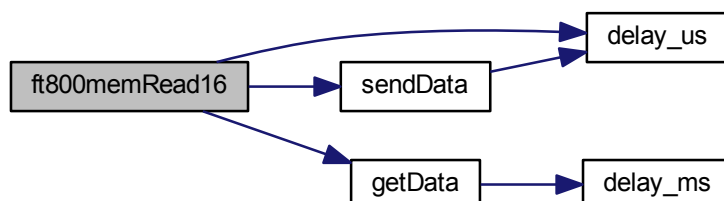
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

##### Returns

16 bit data obtained from device \*

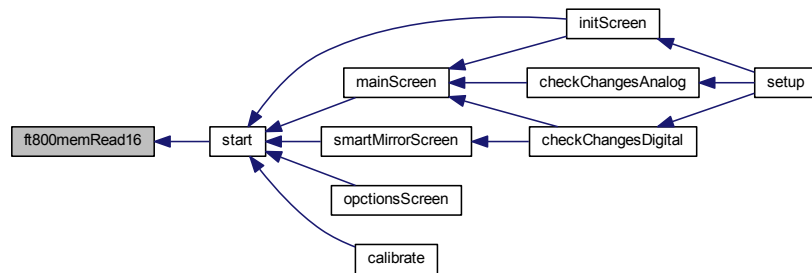
Definition at line 177 of file FT800.cpp.

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

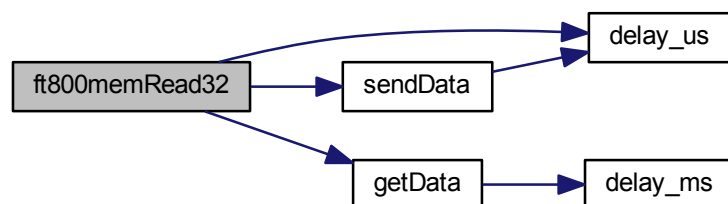
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

##### 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**

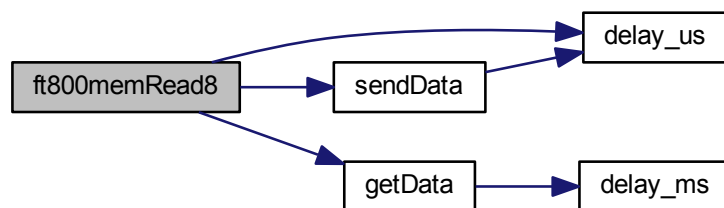
<i>ftAddress</i>	FT800 memory space address (24 bits) *
------------------	--

**Returns**

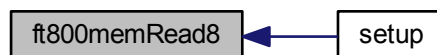
8 bit data obtained from device \*

Definition at line 143 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

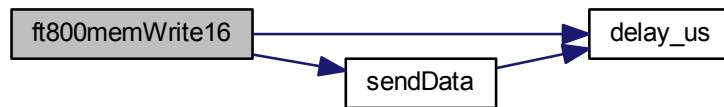
Funtion to send 16 bit value to active device with using SPI interface \*

**Parameters**

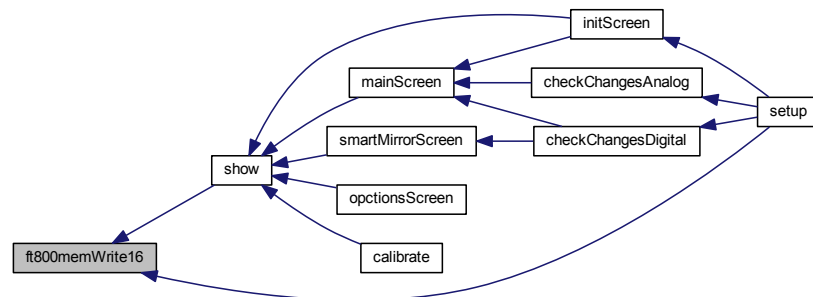
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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* )

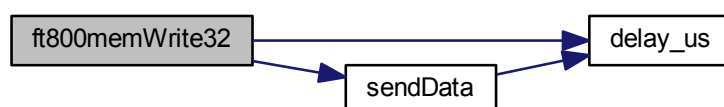
Function to send 32 bit value to active device with using SPI interface \*

##### Parameters

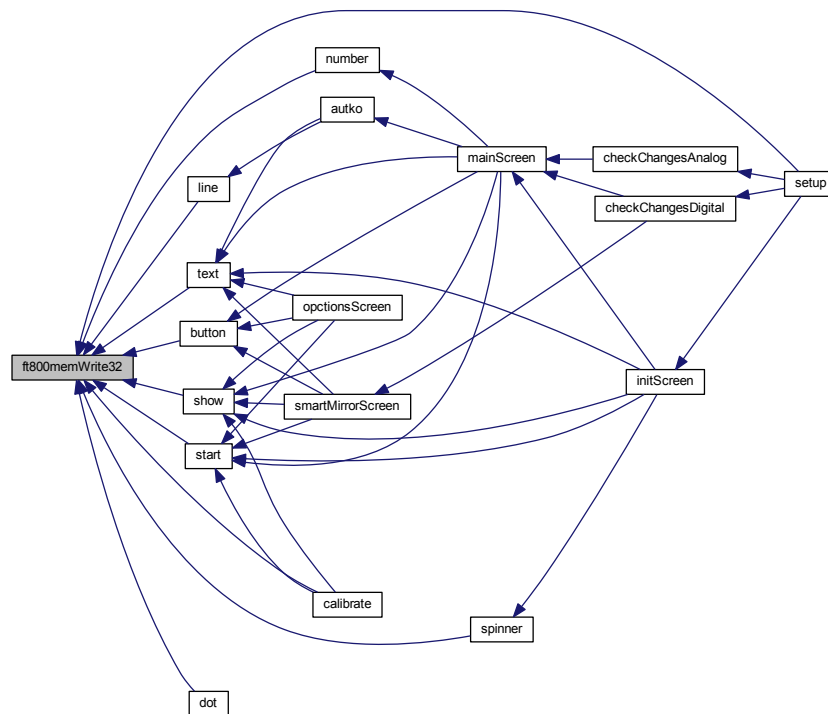
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

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* )

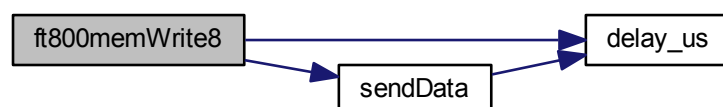
Funtion to send 8 bit value to active device with using SPI interface \*

##### Parameters

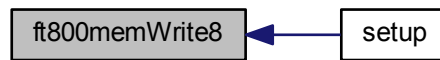
<i>ftAddress</i>	FT800 memory space address (24 bits) *
<i>ftData8</i>	a byte to send *

Definition at line 45 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.2.3.10 unsigned char getData ( )

Function getting data from active device with using SPI interface \*

##### Returns

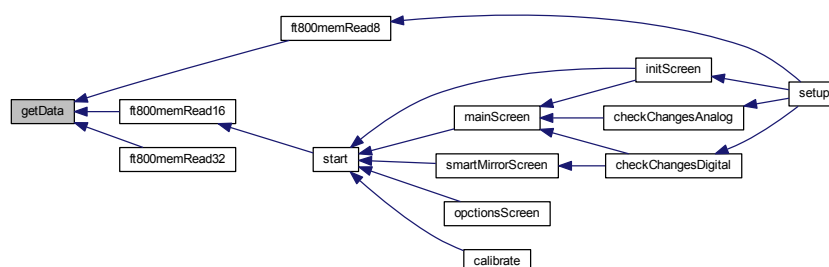
8 bit vvalue with obtained value \*

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

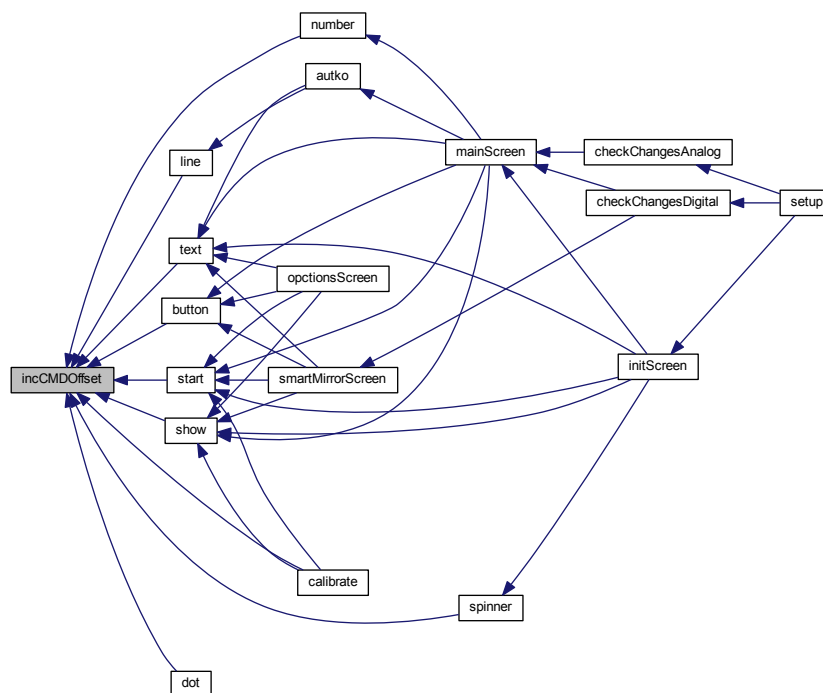
<i>currentOffset</i>	graphics processor command list pointer *
<i>commandSize</i>	number of bytes to increment the offset *

## 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

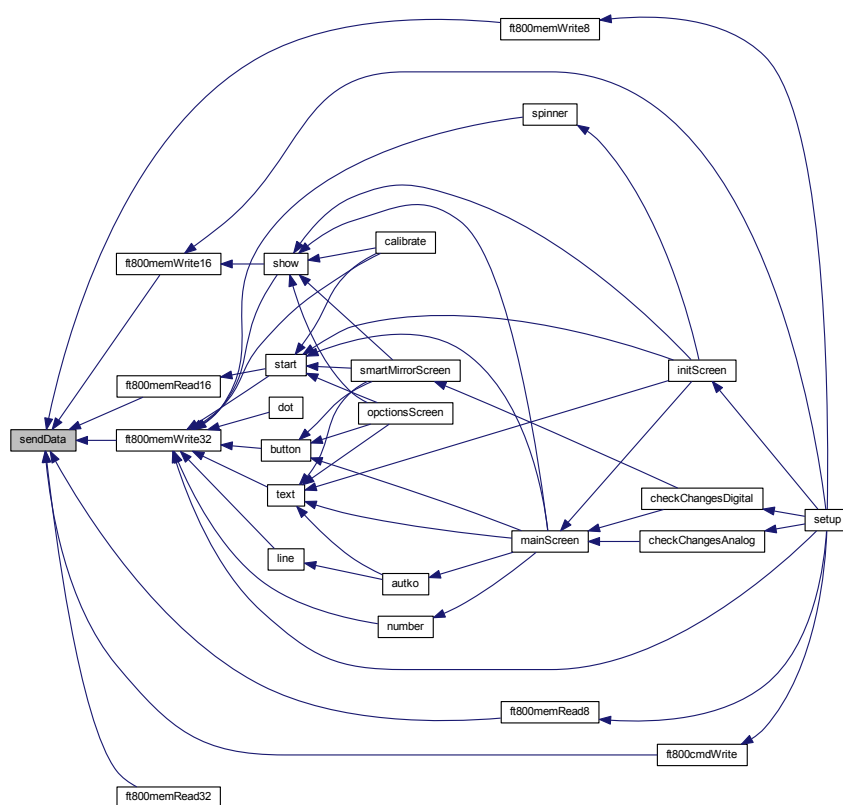
<i>data</i>	8 bit value to send to device *
-------------	---------------------------------

Definition at line 19 of file FT800.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

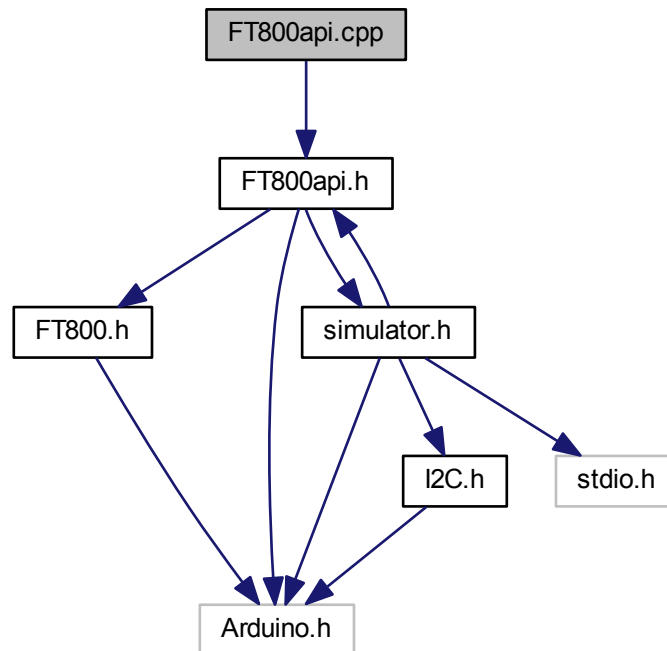


## 4.3 FT800api.cpp File Reference

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

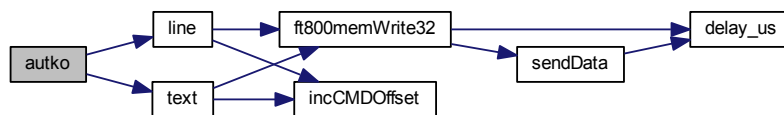
## 4.3.2 Function Documentation

## 4.3.2.1 void autko ( )

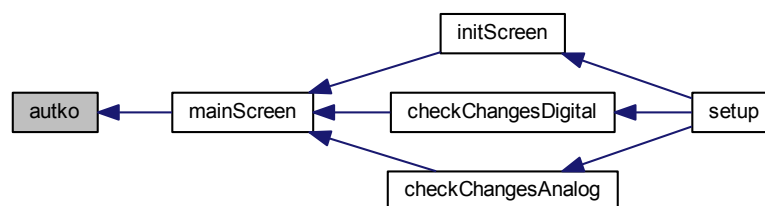
Function drawing car with proper attributes \*

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 )

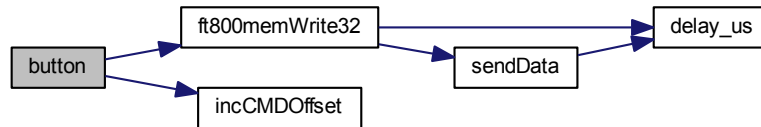
Function which draw a button on the screen \*

## Parameters

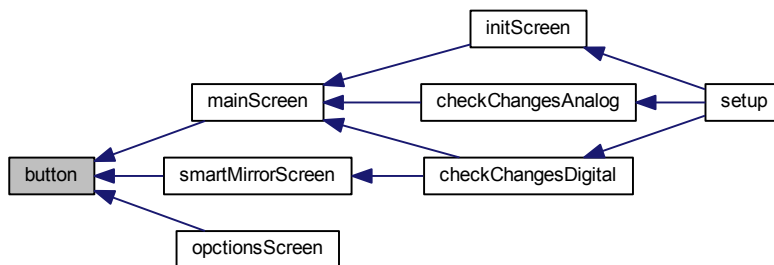
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>w</i>	width for the button *
<i>h</i>	height for the button *
<i>font</i>	font fort the button text *
<i>options</i>	options for the button *
<i>str</i>	text to draw inside button *

Definition at line 127 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

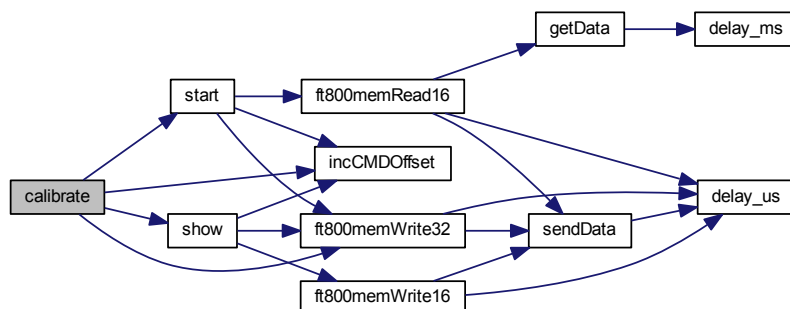


#### 4.3.2.3 void calibrate ( )

Function which calibrate screen \*

Definition at line 248 of file FT800api.cpp.

Here is the call graph for this function:



#### 4.3.2.4 void dot ( unsigned long *color*, unsigned int *point\_size*, unsigned long *point\_x*, unsigned long *point\_y* )

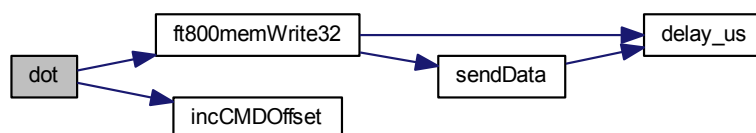
Function which draw a dot on the screen \*

##### Parameters

<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 230 of file FT800api.cpp.

Here is the call graph for this function:

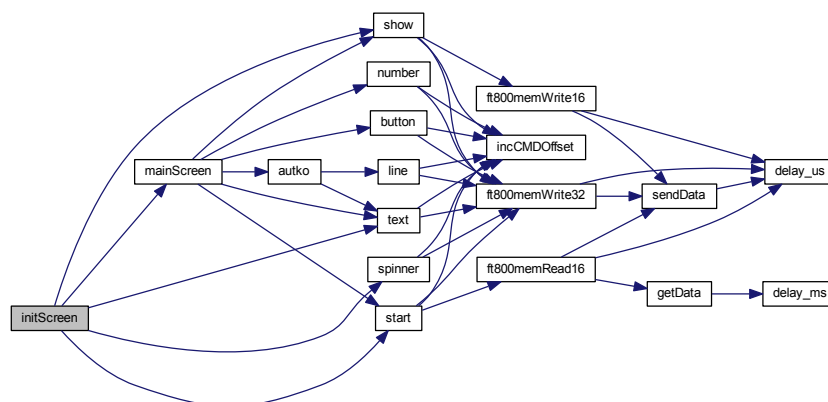


#### 4.3.2.5 void initScreen ( )

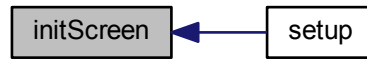
Function showing init screen durig main screen is loading \*

Definition at line 10 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.3.2.6 void line ( unsigned long *color*, unsigned long *line\_x1*, unsigned long *line\_y1*, unsigned long *line\_x2*, unsigned long *line\_y2*, unsigned long *width* )

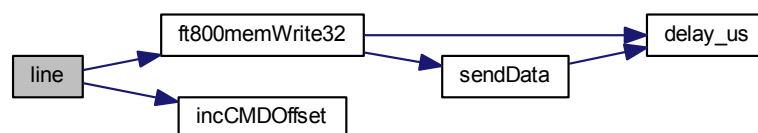
Function which draw a line on the screen \*

#### Parameters

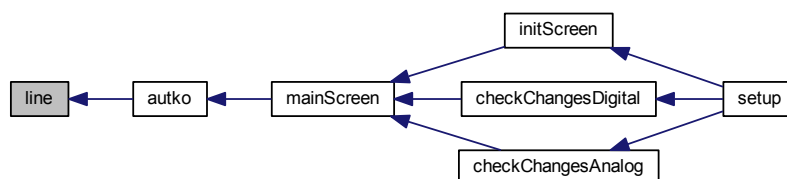
<i>color</i>	line color *
<i>line_x1</i>	x-coordinate for the beginning of the first end *
<i>line_y1</i>	y-coordinate for the beginning of the first end *
<i>line_x2</i>	x-coordinate for the beginning of the second end *
<i>line_y2</i>	y-coordinate for the beginning of the second end *
<i>width</i>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

Definition at line 209 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

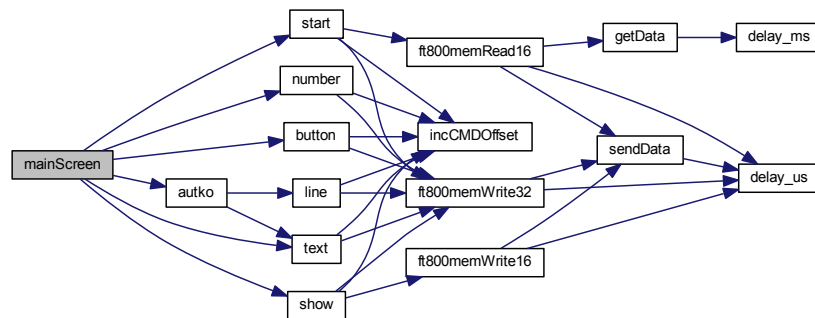


## 4.3.2.7 void mainScreen ( )

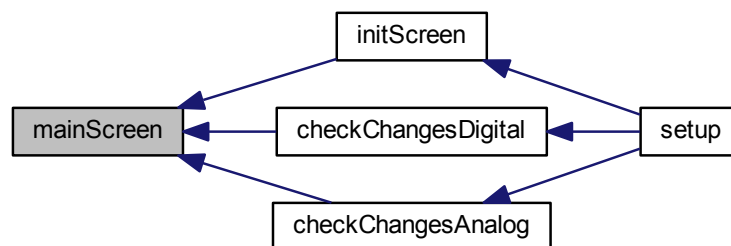
Function showing main screen \*

Definition at line 49 of file FT800api.cpp.

Here is the call graph for this function:



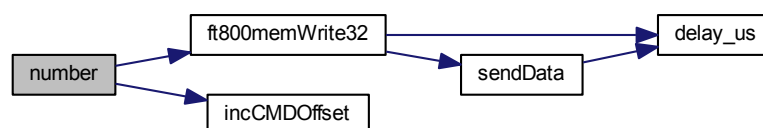
Here is the caller graph for this function:



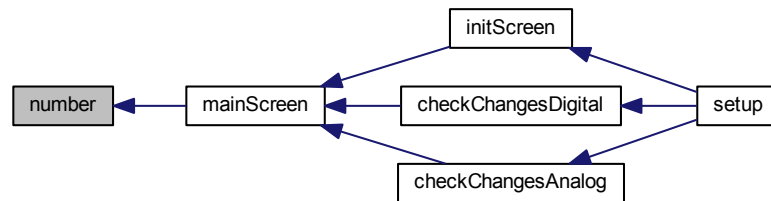
## 4.3.2.8 void number ( int16\_t x, int16\_t y, int16\_t font, uint16\_t options, int value )

Definition at line 194 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

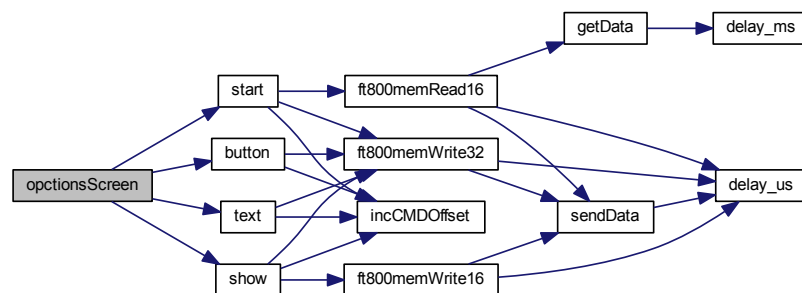


#### 4.3.2.9 void optionsScreen ( )

Function showing options screen \*

Definition at line 77 of file FT800api.cpp.

Here is the call graph for this function:

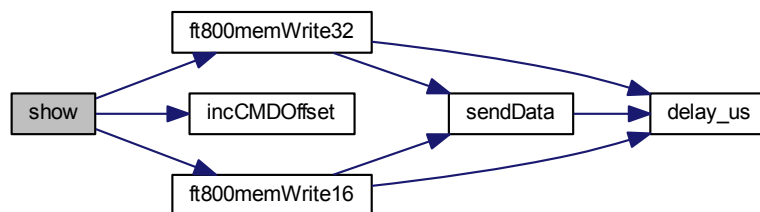


#### 4.3.2.10 void show ( )

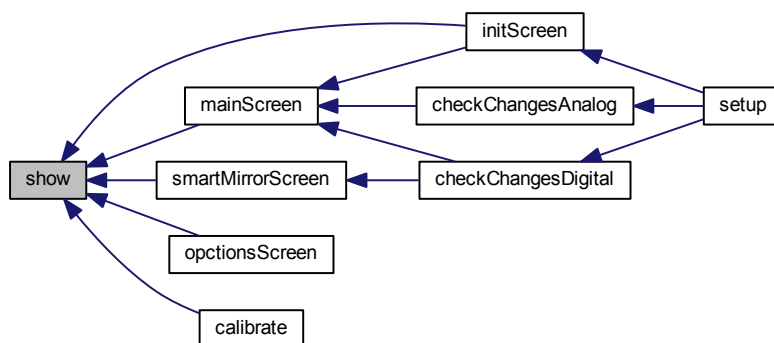
Function showing prepared screen from buffer \*

Definition at line 280 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

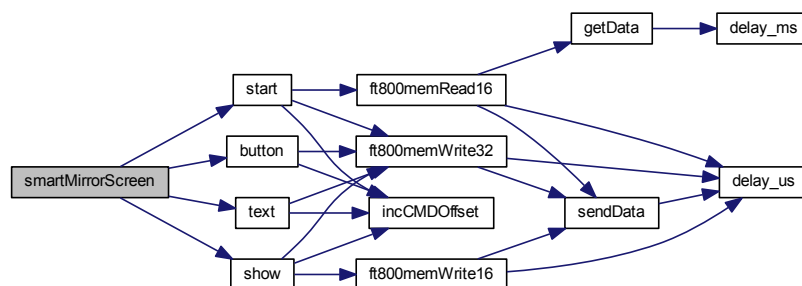


#### 4.3.2.11 void smartMirrorScreen ( )

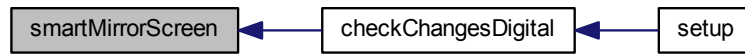
Function showing smart mirror screen \*

Definition at line 68 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.3.2.12 void spinner ( int16\_t x, int16\_t y, uint16\_t style, uint16\_t scale )

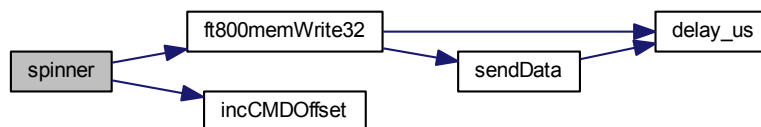
Function which draw a spinner on the screen \*

##### Parameters

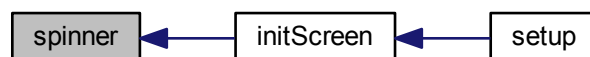
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>style</i>	look swcreeen *
<i>scale</i>	size of spinner *

Definition at line 116 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.3.2.13 void start ( unsigned long color )

Function which start inicjalize new screen \*

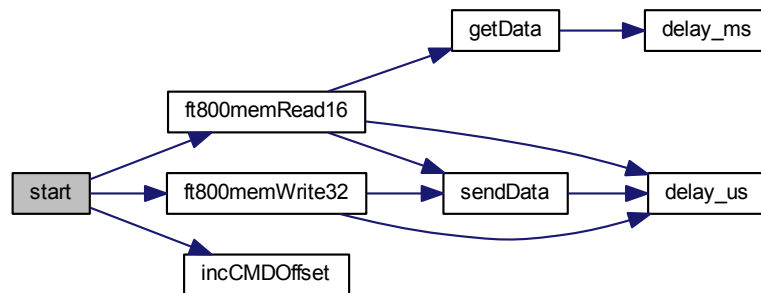


## Parameters

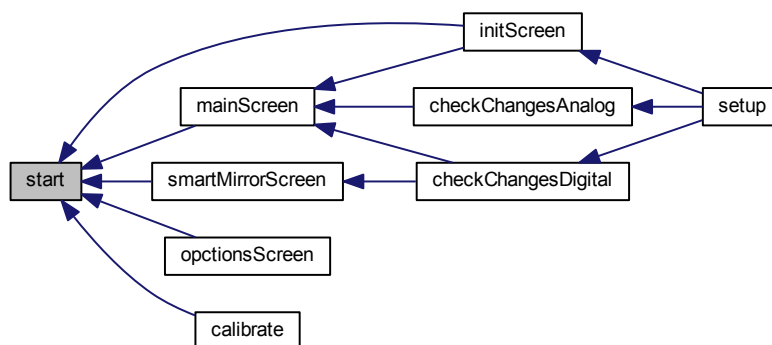
<i>color</i>	background color *
--------------	--------------------

Definition at line 259 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.3.2.14 void text ( int16\_t x, int16\_t y, int16\_t font, uint16\_t options, const char \* str )

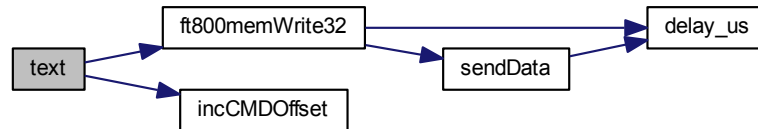
Function which draw a text on the screen \*

## Parameters

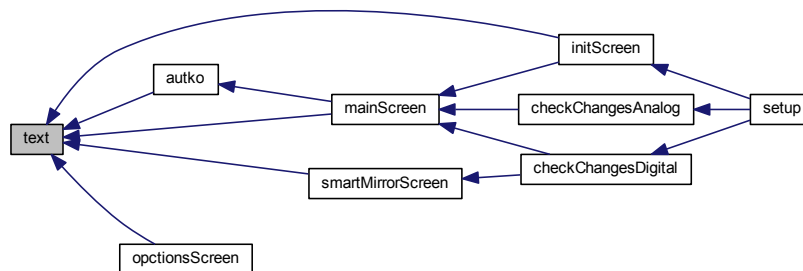
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>font</i>	font for the text *
<i>options</i>	options to set for the text *
<i>str</i>	text to draw on the screen *

Definition at line 162 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



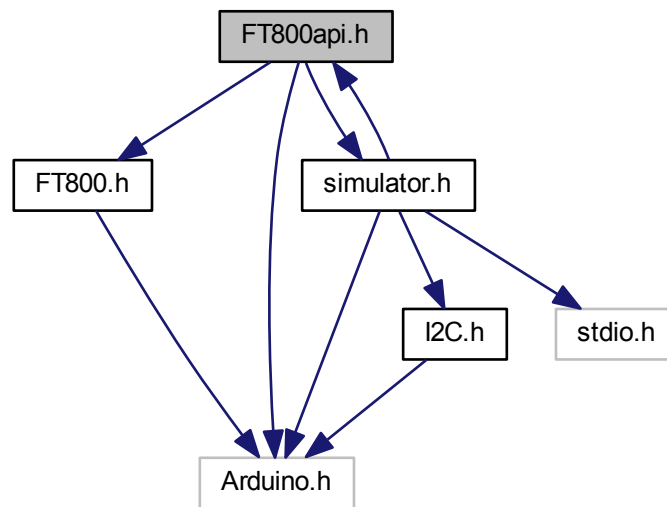
## 4.4 FT800api.h File Reference

File containing declarations of all API functions for VM800.

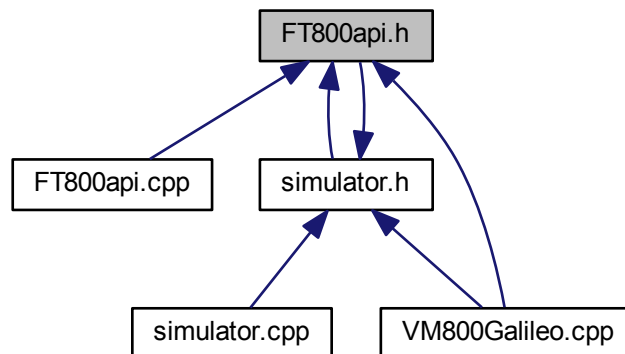
```

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

Include dependency graph for FT800api.h:



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



## Functions

- void [initScreen](#) ()
- void [optionsScreen](#) ()
- void [mainScreen](#) ()
- void [smartMirrorScreen](#) ()
- void [spinner](#) (int16\_t x, int16\_t y, uint16\_t style, uint16\_t scale)
- void [button](#) (int16\_t x, int16\_t y, int16\_t w, int16\_t h, int16\_t font, uint16\_t options, const char \*str)

- void [text](#) (int16\_t x, int16\_t y, int16\_t font, uint16\_t options, const char \*str)
- void [line](#) (unsigned long color, unsigned long line\_x1, unsigned long line\_y1, unsigned long line\_x2, unsigned long line\_y2, unsigned long width)
- void [dot](#) (unsigned long color, unsigned int point\_size, unsigned long point\_x, unsigned long point\_y)
- void [start](#) (unsigned long color)
- void [number](#) (int16\_t x, int16\_t y, int16\_t font, uint16\_t options, int32\_t value)
- void [show](#) ()
- void [calibrate](#) ()
- void [autko](#) ()

## Variables

- unsigned int [cmdOffset](#)
- unsigned int [cmdBufferRd](#)
- unsigned int [cmdBufferWr](#)
- struct [car](#) \* [audi](#)
- int [timeR](#)

### 4.4.1 Detailed Description

File containing declarations of all API functions for VM800.

#### Author

Daniel Sienkiewicz

#### Date

28 February 2016

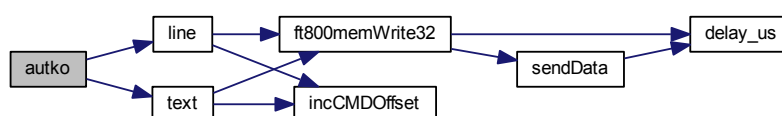
### 4.4.2 Function Documentation

#### 4.4.2.1 void autko ( )

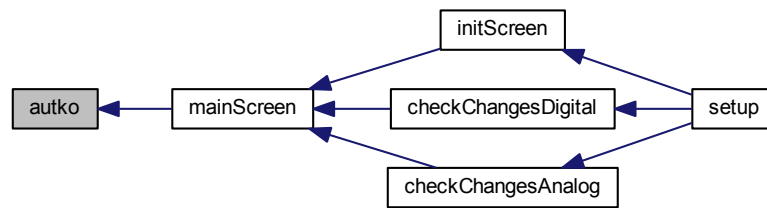
Function drawing car with proper attributes \*

Definition at line 18 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.4.2.2 void button ( int16\_t x, int16\_t y, int16\_t w, int16\_t h, int16\_t font, uint16\_t options, const char \* str )

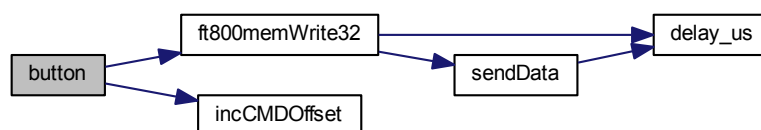
Function which draw a button on the screen \*

##### Parameters

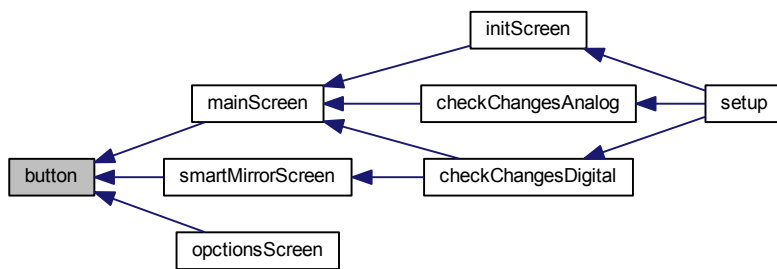
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>w</i>	width for the button *
<i>h</i>	height for the button *
<i>font</i>	font fort the button text *
<i>options</i>	options for the button *
<i>str</i>	text to draw inside button *

Definition at line 127 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

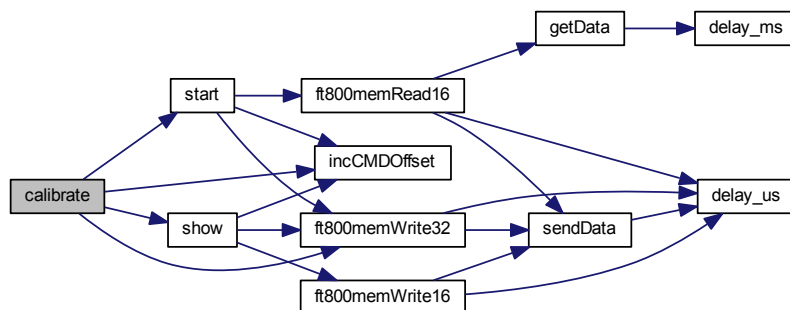


#### 4.4.2.3 void calibrate ( )

Function which calibrate screen \*

Definition at line 248 of file FT800api.cpp.

Here is the call graph for this function:



#### 4.4.2.4 void dot ( unsigned long color, unsigned int point\_size, unsigned long point\_x, unsigned long point\_y )

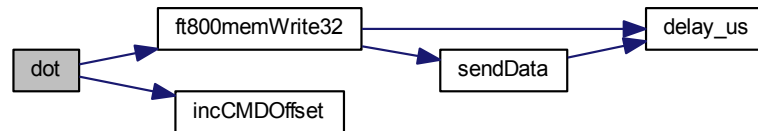
Function which draw a dot on the screen \*

Parameters

<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 230 of file FT800api.cpp.

Here is the call graph for this function:

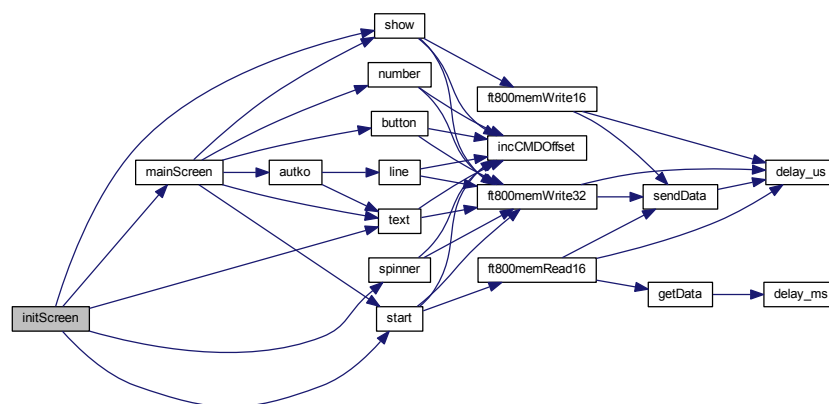


#### 4.4.2.5 void initScreen ( )

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:



Here is the caller graph for this function:



4.4.2.6 void line ( unsigned long *color*, unsigned long *line\_x1*, unsigned long *line\_y1*, unsigned long *line\_x2*, unsigned long *line\_y2*, unsigned long *width* )

Function which draw a line on the screen \*

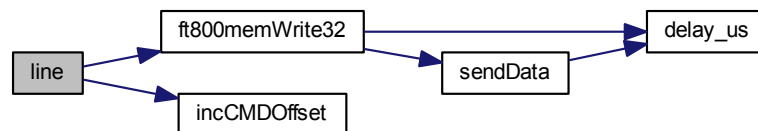


## Parameters

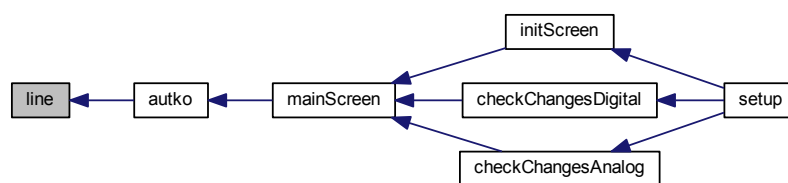
<i>color</i>	line color *
<i>line_x1</i>	x-coordinate for the beginning of the first end *
<i>line_y1</i>	y-coordinate for the beginning of the first end *
<i>line_x2</i>	x-coordinate for the beginning of the second end *
<i>line_y2</i>	y-coordinate for the beginning of the second end *
<i>width</i>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

Definition at line 209 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

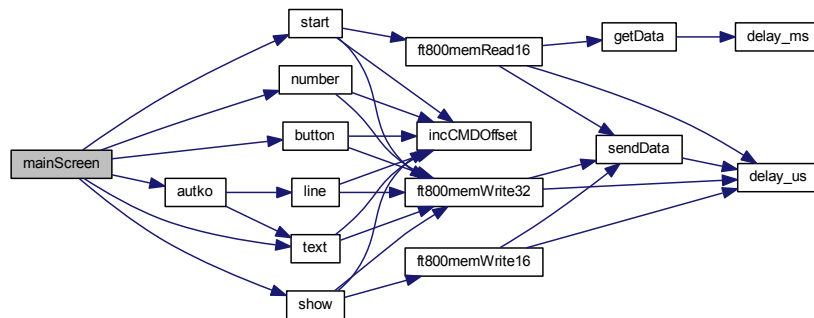


## 4.4.2.7 void mainScreen ( )

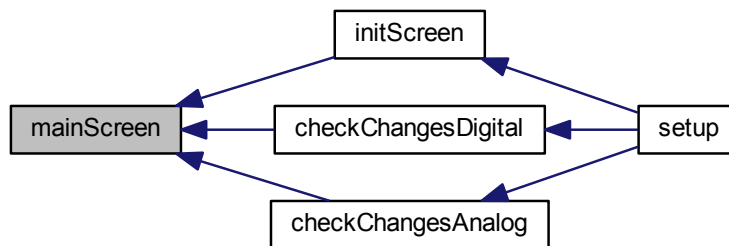
Function showing main screen \*

Definition at line 49 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



**4.4.2.8** void number ( int16\_t x, int16\_t y, int16\_t font, uint16\_t options, int32\_t value )

Function which draw a number on the screen \*

#### Parameters

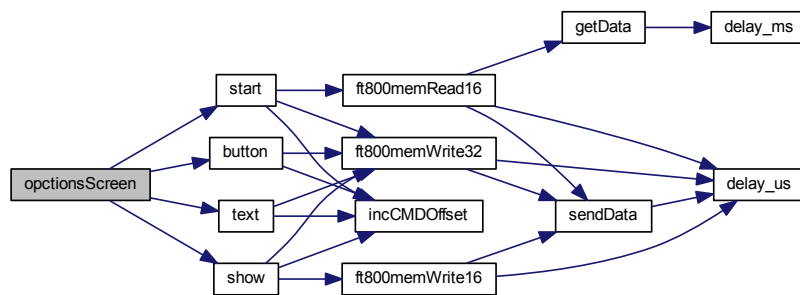
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>font</i>	font for the number *
<i>options</i>	options to set for the number *
<i>value</i>	value to draw on the screen *

**4.4.2.9** void optionsScreen ( )

Function showing options screen \*

Definition at line 77 of file FT800api.cpp.

Here is the call graph for this function:

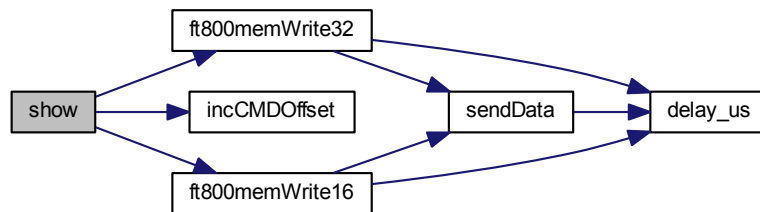


#### 4.4.2.10 void show ( )

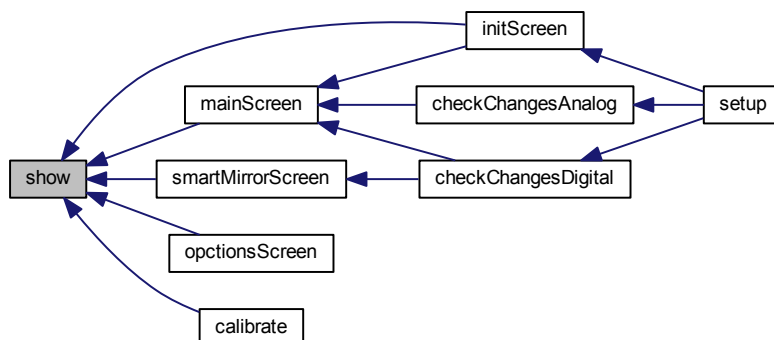
Function showing prepared screen from buffer \*

Definition at line 280 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

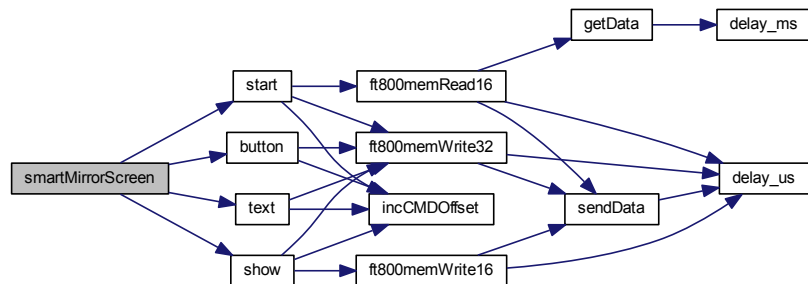


#### 4.4.2.11 void smartMirrorScreen ( )

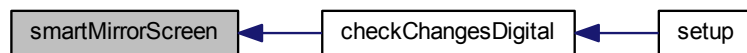
Function showing smart mirror screen \*

Definition at line 68 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.4.2.12 void spinner ( int16\_t x, int16\_t y, uint16\_t style, uint16\_t scale )

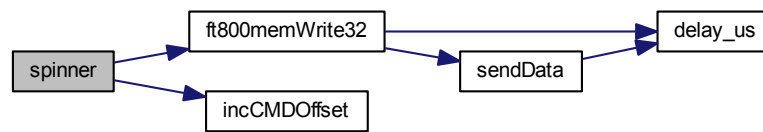
Function which draw a spinner on the screen \*

##### Parameters

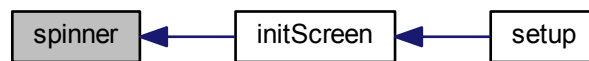
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>style</i>	look swscreen *
<i>scale</i>	size of spinner *

Definition at line 116 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.4.2.13 void start ( unsigned long *color* )

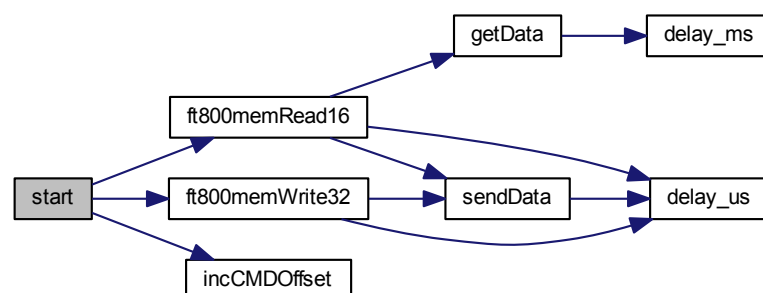
Function which start inicjalize new screen \*

##### Parameters

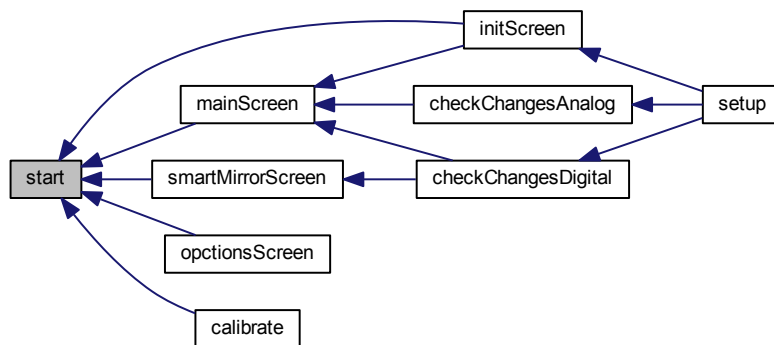
<i>color</i>	backgroud color *
--------------	-------------------

Definition at line 259 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.4.2.14 void text ( int16\_t x, int16\_t y, int16\_t font, uint16\_t options, const char \* str )

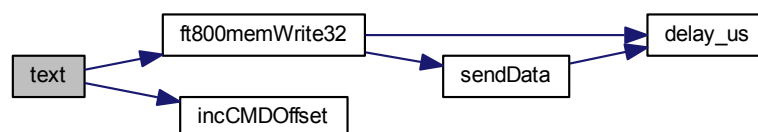
Function which draw a text on the screen \*

##### Parameters

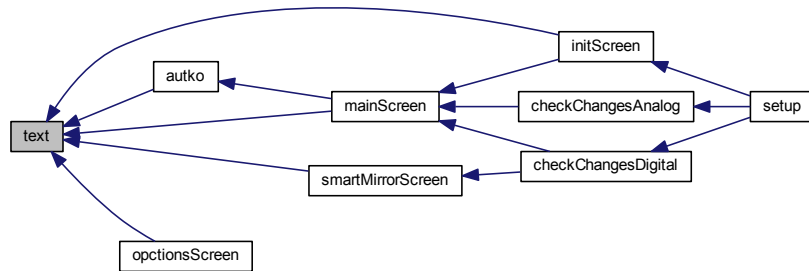
<i>x</i>	x-coordinate on the screen *
<i>y</i>	y-coordinate on the screen *
<i>font</i>	font for the text *
<i>options</i>	options to set for the text *
<i>str</i>	text to draw on the screen *

Definition at line 162 of file FT800api.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



### 4.4.3 Variable Documentation

#### 4.4.3.1 struct car\* audi

Main car structure with data from sensors

Definition at line 41 of file VM800Galileo.cpp.

#### 4.4.3.2 unsigned int cmdBufferRd

Used to navigate command ring buffer

Definition at line 35 of file VM800Galileo.cpp.

#### 4.4.3.3 unsigned int cmdBufferWr

Used to navigate command ring buffer

Definition at line 36 of file VM800Galileo.cpp.

#### 4.4.3.4 unsigned int cmdOffset

Used to navigate command ring buffer

Definition at line 37 of file VM800Galileo.cpp.

#### 4.4.3.5 int timeR

Data refresh time to save to file

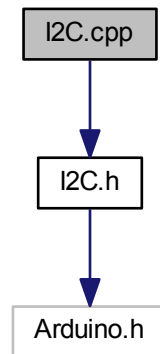
Definition at line 44 of file VM800Galileo.cpp.

## 4.5 I2C.cpp File Reference

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

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

Include dependency graph for I2C.cpp:



### Functions

- int `readPCF` (char adres)

### Variables

- int `d` = 1

#### 4.5.1 Detailed Description

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

##### Author

Daniel Sienkiewicz

##### Date

28 February 2016

#### 4.5.2 Function Documentation

##### 4.5.2.1 int readPCF ( char *adres* )

Reading value from PCF8574N I/O Expander \*



## Parameters

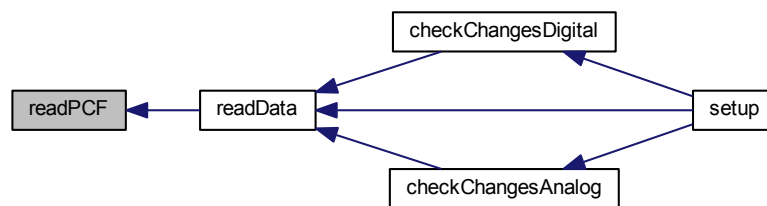
<i>adres</i>	The address of PCF8574N I/O Expander *
--------------	--

## 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

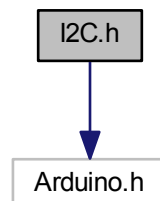
Definition at line 10 of file I2C.cpp.

## 4.6 I2C.h File Reference

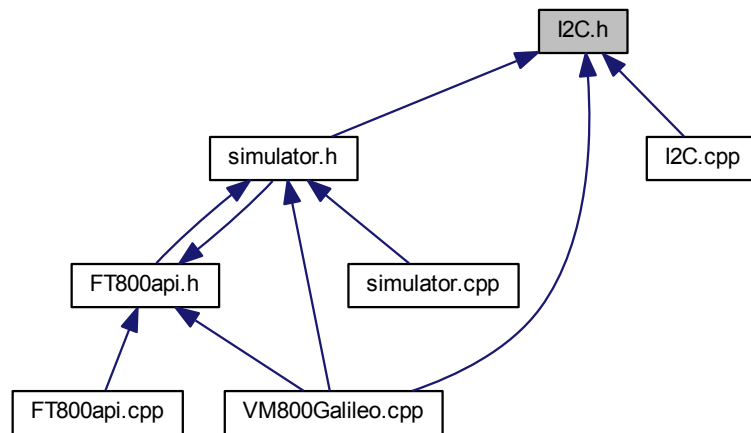
File containing declarations of function to read data with using I2C protocol.

```
#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

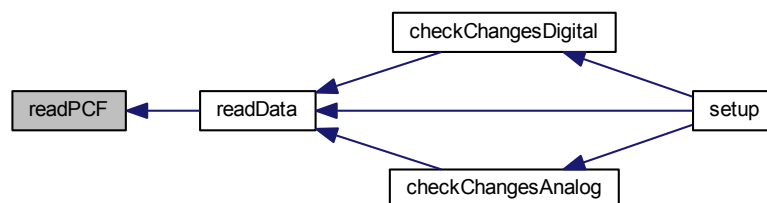
<i>adres</i>	The address of PCF8574N I/O Expander *
--------------	--

##### 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:

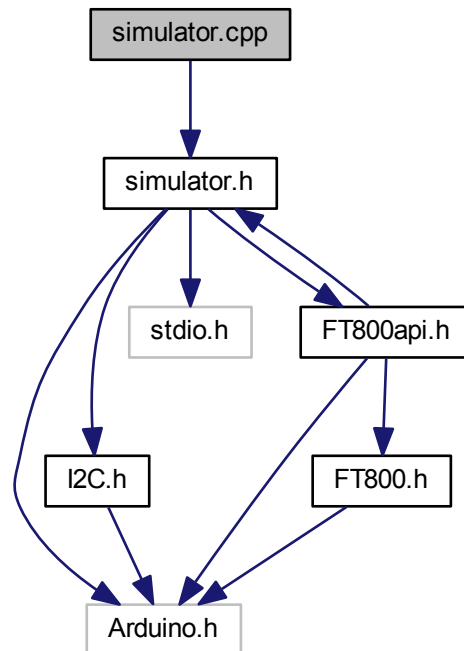


## 4.7 simulator.cpp File Reference

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

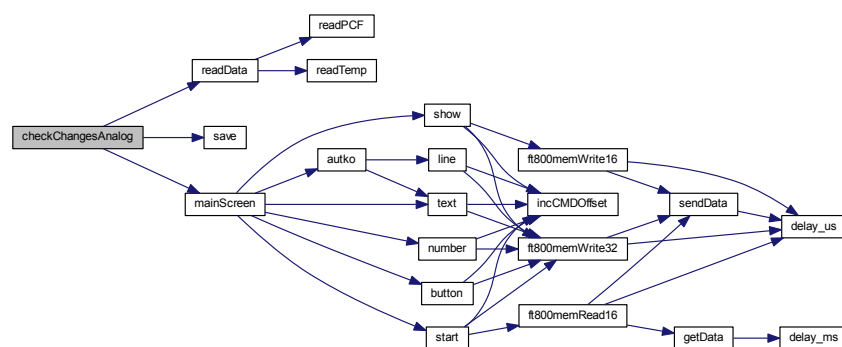
## 4.7.2 Function Documentation

### 4.7.2.1 void checkChangesAnalog ( )

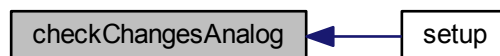
Check if sth on analog ports was changed \*

Definition at line 138 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

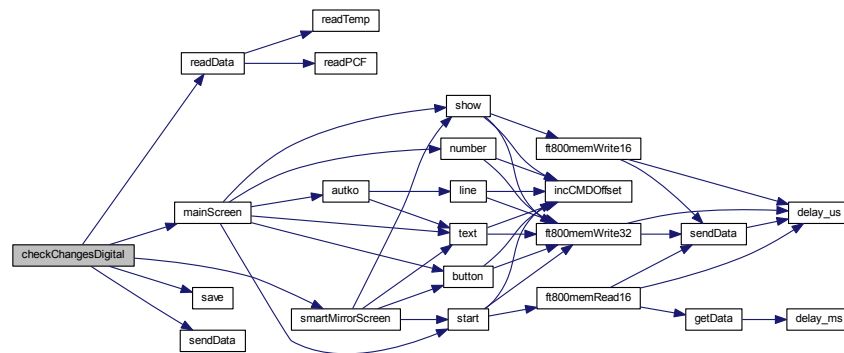


### 4.7.2.2 void checkChangesDigital ( )

Check if sth on digital ports was changed \*

Definition at line 96 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

##### Parameters

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

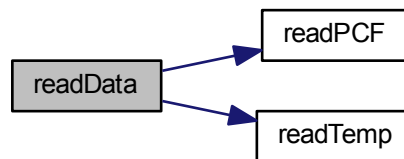
Definition at line 10 of file simulator.cpp.

#### 4.7.2.4 struct car\* readData ( )

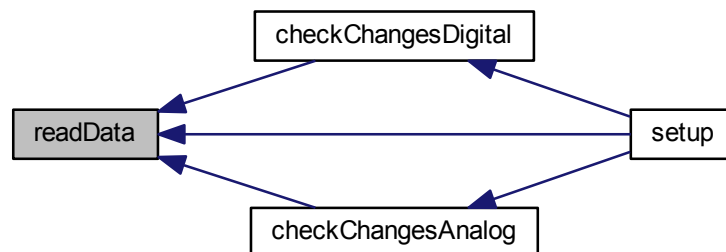
Reading data about car status \*

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

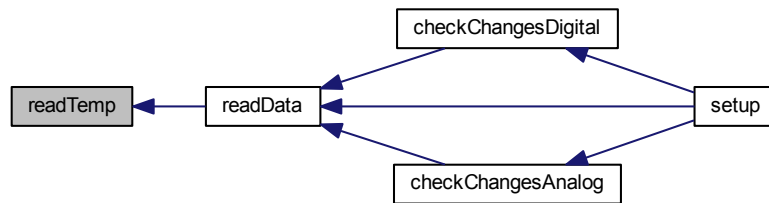
<i>portNumber</i>	The number of the analog input pin to read *
-------------------	--

##### 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 )

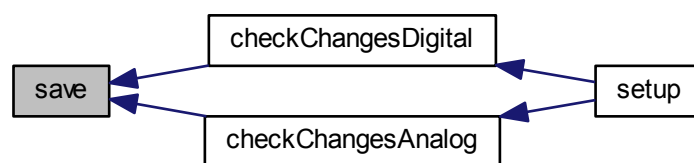
Copying data function from temporary to main struct \*

##### 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.



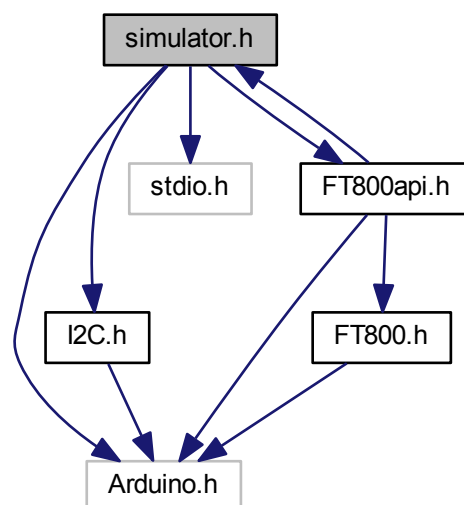
Here is the caller graph for this function:



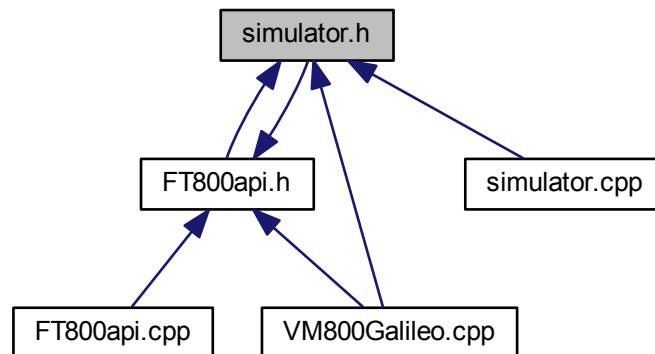
## 4.8 simulator.h File Reference

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

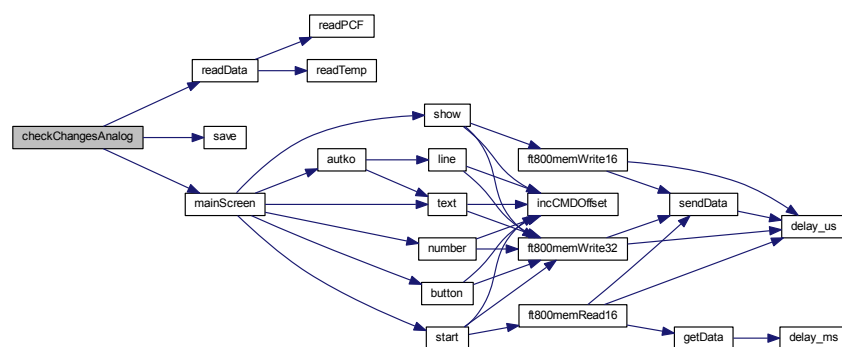
## 4.8.2 Function Documentation

### 4.8.2.1 void checkChangesAnalog ( )

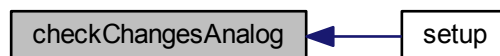
Check if sth on analog ports was changed \*

Definition at line 138 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

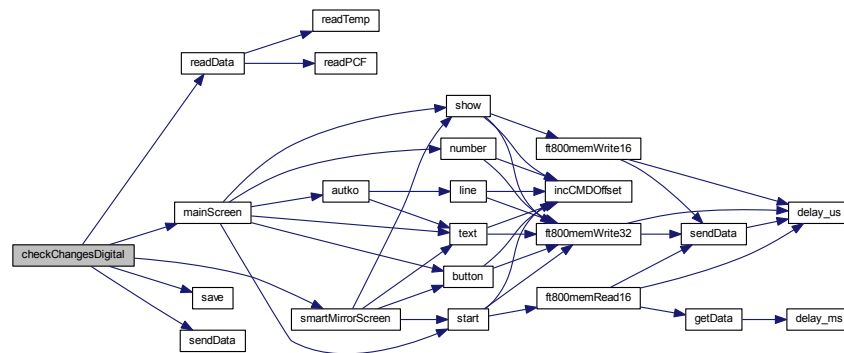


### 4.8.2.2 void checkChangesDigital ( )

Check if sth on digital ports was changed \*

Definition at line 96 of file simulator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

##### Parameters

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

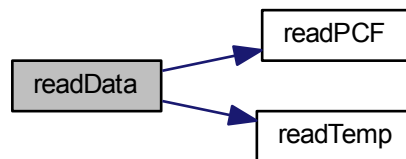
Definition at line 10 of file simulator.cpp.

#### 4.8.2.4 struct car\* readData ( )

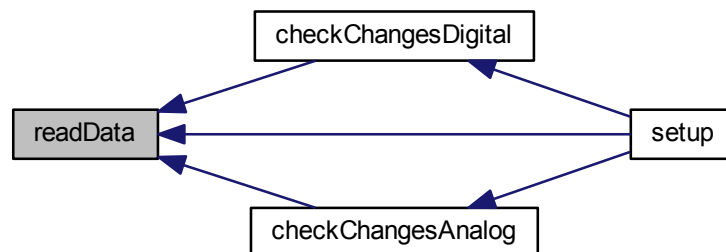
Reading data about car status \*

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

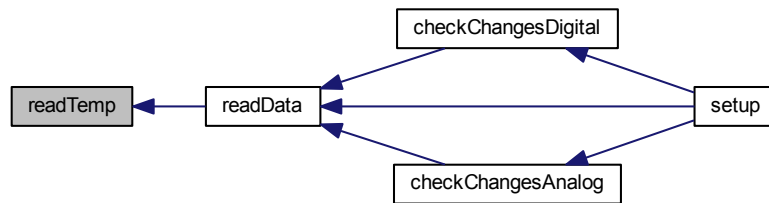
<i>portNumber</i>	The number of the analog input pin to read *
-------------------	--

##### 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 )

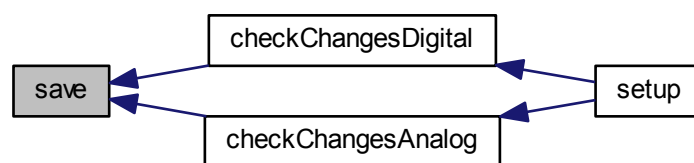
Copying data function from temporary to main struct \*

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`

Main car structure with data from sensors

Definition at line 41 of file VM800Galileo.cpp.

#### 4.8.3.2 `int dataFormat`

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

Definition at line 42 of file VM800Galileo.cpp.

#### 4.8.3.3 `int saveData`

If data will be saving on SD card

Definition at line 43 of file VM800Galileo.cpp.

#### 4.8.3.4 `short int screenNR`

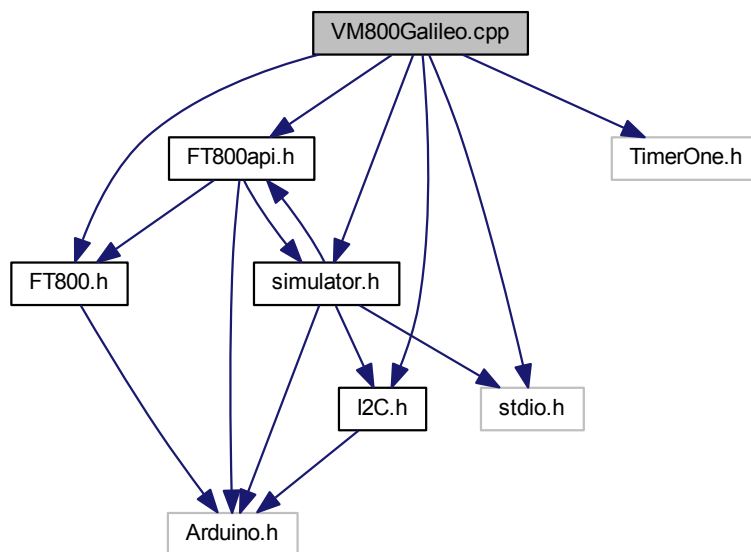
Selected screen - 1 main screen, 2 - smart mirror, 3 - options

Definition at line 40 of file VM800Galileo.cpp.

## 4.9 VM800Galileo.cpp File Reference

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

Include dependency graph for VM800Galileo.cpp:



### Functions

- void [setup](#) (void)
- void [loop](#) ()

### Variables

- unsigned int [lcdWidth](#)
- unsigned int [lcdHeight](#)
- unsigned int [lcdHcycle](#)
- unsigned int [lcdHoffset](#)
- unsigned int [lcdHsync0](#)
- unsigned int [lcdHsync1](#)
- unsigned int [lcdVcycle](#)
- unsigned int [lcdVoffset](#)
- unsigned int [lcdVsync0](#)
- unsigned int [lcdVsync1](#)
- unsigned char [lcdPclk](#)
- unsigned char [lcdSwizzle](#)



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

### 4.9.1 Function Documentation

#### 4.9.1.1 void loop ( )

function executed in infinity loop after finished executing setup function \*

Definition at line 230 of file VM800Galileo.cpp.

#### 4.9.1.2 void setup ( void )

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

< Active width of LCD display

< Active height of LCD display

< Total number of clocks per line

< Start of active line

< Start of horizontal sync pulse

< End of horizontal sync pulse

< Total number of lines per screen

< Start of active screen

< Start of vertical sync pulse

< End of vertical sync pulse

< Pixel Clock

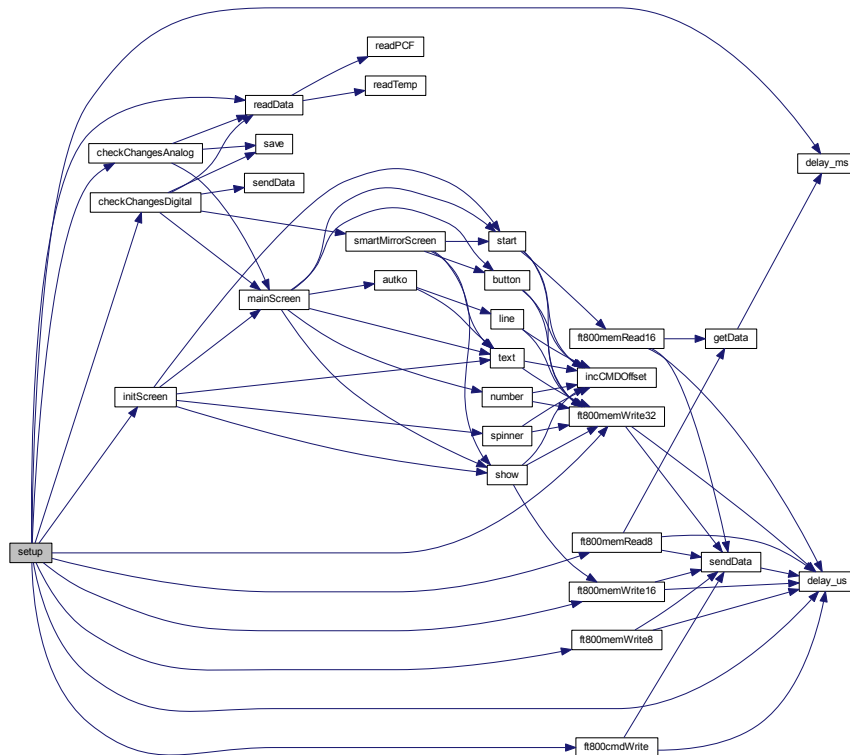
< Define RGB output pins

< Define active edge of PCLK

< WQVGA display parameters

Definition at line 51 of file VM800Galileo.cpp.

Here is the call graph for this function:



## 4.9.2 Variable Documentation

### 4.9.2.1 struct car\* audi

Main car structure with data from sensors

Definition at line 41 of file VM800Galileo.cpp.

### 4.9.2.2 char buf[9]

Actual date

Definition at line 45 of file VM800Galileo.cpp.

### 4.9.2.3 unsigned int cmdBufferRd

Used to navigate command ring buffer

Definition at line 35 of file VM800Galileo.cpp.

#### 4.9.2.4 unsigned int cmdBufferWr =0x0000

Used to navigate command ring buffer

Definition at line 36 of file VM800Galileo.cpp.

#### 4.9.2.5 unsigned int cmdOffset =0x0000

Used to navigate command rung buffer

Definition at line 37 of file VM800Galileo.cpp.

#### 4.9.2.6 int dataFormat = 3

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

Definition at line 42 of file VM800Galileo.cpp.

#### 4.9.2.7 unsigned char ft800Gpio

Used for FT800 GPIO register

Definition at line 38 of file VM800Galileo.cpp.

#### 4.9.2.8 unsigned int lcdHcycle

Total number of clocks per line

Definition at line 20 of file VM800Galileo.cpp.

#### 4.9.2.9 unsigned int lcdHeight

Active height of LCD display

Definition at line 19 of file VM800Galileo.cpp.

#### 4.9.2.10 unsigned int lcdHoffset

Start of active line

Definition at line 21 of file VM800Galileo.cpp.

#### 4.9.2.11 unsigned int lcdHsync0

Start of horizontal sync pulse

Definition at line 22 of file VM800Galileo.cpp.

#### 4.9.2.12 unsigned int lcdHsync1

End of horizontal sync pulse

Definition at line 23 of file VM800Galileo.cpp.

#### 4.9.2.13 unsigned char lcdPclk

Pixel Clock

Definition at line 28 of file VM800Galileo.cpp.

#### 4.9.2.14 unsigned char lcdPclkpol

Define active edge of PCLK

Definition at line 30 of file VM800Galileo.cpp.

#### 4.9.2.15 unsigned char lcdSwizzle

Define RGB output pins

Definition at line 29 of file VM800Galileo.cpp.

#### 4.9.2.16 unsigned int lcdVcycle

Total number of lines per screen

Definition at line 24 of file VM800Galileo.cpp.

#### 4.9.2.17 unsigned int lcdVoffset

Start of active screen

Definition at line 25 of file VM800Galileo.cpp.

#### 4.9.2.18 unsigned int lcdVsync0

Start of vertical sync pulse

Definition at line 26 of file VM800Galileo.cpp.

#### 4.9.2.19 unsigned int lcdVsync1

End of vertical sync pulse

Definition at line 27 of file VM800Galileo.cpp.

#### 4.9.2.20 unsigned int lcdWidth

Active width of LCD display

Definition at line 18 of file VM800Galileo.cpp.

#### 4.9.2.21 unsigned long ramCommandBuffer =RAM\_CMD

Set beginning of graphics command memory

Definition at line 33 of file VM800Galileo.cpp.

#### 4.9.2.22 unsigned long ramDisplayList =RAM\_DL

Set beginning of display list memory

Definition at line 32 of file VM800Galileo.cpp.

#### 4.9.2.23 int saveData = 0

If data will be saving on SD card

Definition at line 43 of file VM800Galileo.cpp.

#### 4.9.2.24 short int screenNR = 1

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

Definition at line 40 of file VM800Galileo.cpp.

#### 4.9.2.25 int timeR = 1

Data refresh time to save to file

Definition at line 44 of file VM800Galileo.cpp.



# Index

- ABS
  - FT800.h, [24](#)
- audi
  - FT800api.h, [85](#)
  - simulator.h, [101](#)
  - VM800Galileo.cpp, [104](#)
- autko
  - FT800api.cpp, [63](#)
  - FT800api.h, [74](#)
- BLACK
  - FT800.h, [24](#)
- BLUE
  - FT800.h, [24](#)
- buf
  - VM800Galileo.cpp, [104](#)
- button
  - FT800api.cpp, [63](#)
  - FT800api.h, [75](#)
- CLR\_COL
  - FT800.h, [24](#)
- CLR\_STN
  - FT800.h, [24](#)
- CLR\_TAG
  - FT800.h, [24](#)
- CMD\_APPEND
  - FT800.h, [24](#)
- CMD\_BGCOLOR
  - FT800.h, [24](#)
- CMD\_BUTTON
  - FT800.h, [25](#)
- CMD\_CALIBRATE
  - FT800.h, [25](#)
- CMD\_CLOCK
  - FT800.h, [25](#)
- CMD\_COLDSTART
  - FT800.h, [25](#)
- CMD\_DIAL
  - FT800.h, [25](#)
- CMD\_DLSTART
  - FT800.h, [25](#)
- CMD\_FGCOLOR
  - FT800.h, [25](#)
- CMD\_GAUGE
  - FT800.h, [25](#)
- CMD\_GETMATRIX
  - FT800.h, [25](#)
- CMD\_GETPTR
  - FT800.h, [25](#)
- CMD\_GRADCOLOR
  - FT800.h, [26](#)
- CMD\_GRADIENT
  - FT800.h, [26](#)
- CMD\_INFLATE
  - FT800.h, [26](#)
- CMD\_INTERRUPT
  - FT800.h, [26](#)
- CMD\_KEYS
  - FT800.h, [26](#)
- CMD\_LOADIDENTITY
  - FT800.h, [26](#)
- CMD\_LOADIMAGE
  - FT800.h, [26](#)
- CMD\_LOGO
  - FT800.h, [26](#)
- CMD\_MEMCPY
  - FT800.h, [26](#)
- CMD\_MEMCRC
  - FT800.h, [26](#)
- CMD\_MEMSET
  - FT800.h, [27](#)
- CMD\_MEMWRITE
  - FT800.h, [27](#)
- CMD\_MEMZERO
  - FT800.h, [27](#)
- CMD\_NUMBER
  - FT800.h, [27](#)
- CMD\_PROGRESS
  - FT800.h, [27](#)
- CMD\_REGREAD
  - FT800.h, [27](#)
- CMD\_ROTATE
  - FT800.h, [27](#)
- CMD\_SCALE
  - FT800.h, [27](#)
- CMD\_SCREENSAVER
  - FT800.h, [27](#)
- CMD\_SCROLLBAR
  - FT800.h, [27](#)
- CMD\_SETFONT
  - FT800.h, [28](#)
- CMD\_SETMATRIX
  - FT800.h, [28](#)
- CMD\_SKETCH
  - FT800.h, [28](#)
- CMD\_SLIDER
  - FT800.h, [28](#)
- CMD\_SNAPSHOT

FT800.h, [28](#)  
 CMD\_SPINNER  
     FT800.h, [28](#)  
 CMD\_STOP  
     FT800.h, [28](#)  
 CMD\_SWAP  
     FT800.h, [28](#)  
 CMD\_TEXT  
     FT800.h, [28](#)  
 CMD\_TOGGLE  
     FT800.h, [28](#)  
 CMD\_TRACK  
     FT800.h, [29](#)  
 CMD\_TRANSLATE  
     FT800.h, [29](#)  
 CMDBUF\_SIZE  
     FT800.h, [29](#)  
 calibrate  
     FT800api.cpp, [64](#)  
     FT800api.h, [76](#)  
 car, [5](#)  
     doors, [5](#)  
     lights, [5](#)  
     r, [6](#)  
     seatbelts, [6](#)  
     tempEngine, [6](#)  
     tempIn, [6](#)  
     tempOut, [6](#)  
 checkChangesAnalog  
     simulator.cpp, [91](#)  
     simulator.h, [97](#)  
 checkChangesDigital  
     simulator.cpp, [91](#)  
     simulator.h, [97](#)  
 cmdBufferRd  
     FT800api.h, [85](#)  
     VM800Galileo.cpp, [104](#)  
 cmdBufferWr  
     FT800api.h, [85](#)  
     VM800Galileo.cpp, [104](#)  
 cmdOffset  
     FT800api.h, [85](#)  
     VM800Galileo.cpp, [105](#)  
 d  
     I2C.cpp, [87](#)  
 DECR\_WRAP  
     FT800.h, [29](#)  
 DECR  
     FT800.h, [29](#)  
 DL\_ALPHA\_FUNC  
     FT800.h, [29](#)  
 DL\_BEGIN  
     FT800.h, [29](#)  
 DL\_BITMAP\_HANDLE  
     FT800.h, [29](#)  
 DL\_BITMAP\_LAYOUT  
     FT800.h, [29](#)  
 DL\_BITMAP\_SIZE  
     FT800.h, [29](#)  
 DL\_BITMAP\_SOURCE  
     FT800.h, [30](#)  
 DL\_BITMAP\_TFORM\_A  
     FT800.h, [30](#)  
 DL\_BITMAP\_TFORM\_B  
     FT800.h, [30](#)  
 DL\_BITMAP\_TFORM\_C  
     FT800.h, [30](#)  
 DL\_BITMAP\_TFORM\_D  
     FT800.h, [30](#)  
 DL\_BITMAP\_TFORM\_E  
     FT800.h, [30](#)  
 DL\_BITMAP\_TFORM\_F  
     FT800.h, [30](#)  
 DL\_BLEND\_FUNC  
     FT800.h, [30](#)  
 DL\_CALL  
     FT800.h, [30](#)  
 DL\_CELL  
     FT800.h, [30](#)  
 DL\_CLEAR\_RGB  
     FT800.h, [31](#)  
 DL\_CLEAR\_STENCIL  
     FT800.h, [31](#)  
 DL\_CLEAR\_TAG  
     FT800.h, [31](#)  
 DL\_CLEAR  
     FT800.h, [31](#)  
 DL\_COLOR\_MASK  
     FT800.h, [31](#)  
 DL\_COLOR\_RGB  
     FT800.h, [31](#)  
 DL\_COLOR\_A  
     FT800.h, [31](#)  
 DL\_DISPLAY  
     FT800.h, [31](#)  
 DL\_END  
     FT800.h, [31](#)  
 DL\_JUMP  
     FT800.h, [31](#)  
 DL\_LINE\_WIDTH  
     FT800.h, [32](#)  
 DL\_MACRO  
     FT800.h, [32](#)  
 DL\_POINT\_SIZE  
     FT800.h, [32](#)  
 DL\_RESTORE\_CONTEXT  
     FT800.h, [32](#)  
 DL\_RETURN  
     FT800.h, [32](#)  
 DL\_SAVE\_CONTEXT  
     FT800.h, [32](#)  
 DL\_SCISSOR\_SIZE  
     FT800.h, [32](#)  
 DL\_SCISSOR\_XY  
     FT800.h, [32](#)  
 DL\_STENCIL\_FUNC



- FT800.h, [32](#)
- DL\_STENCIL\_MASK
  - FT800.h, [32](#)
- DL\_STENCIL\_OP
  - FT800.h, [33](#)
- DL\_TAG\_MASK
  - FT800.h, [33](#)
- DL\_TAG
  - FT800.h, [33](#)
- DL\_VERTEX2II
  - FT800.h, [33](#)
- DL\_VERTEX2F
  - FT800.h, [33](#)
- DLSWAP\_DONE
  - FT800.h, [33](#)
- DLSWAP\_FRAME
  - FT800.h, [33](#)
- DLSWAP\_LINE
  - FT800.h, [33](#)
- DST\_ALPHA
  - FT800.h, [33](#)
- dataFormat
  - simulator.h, [101](#)
  - VM800Galileo.cpp, [105](#)
- delay\_ms
  - FT800.cpp, [8](#)
  - FT800.h, [51](#)
- delay\_us
  - FT800.cpp, [8](#)
  - FT800.h, [52](#)
- doors
  - car, [5](#)
- dot
  - FT800api.cpp, [64](#)
  - FT800api.h, [76](#)
- EDGE\_STRIP\_A
  - FT800.h, [33](#)
- EDGE\_STRIP\_B
  - FT800.h, [34](#)
- EDGE\_STRIP\_L
  - FT800.h, [34](#)
- EDGE\_STRIP\_R
  - FT800.h, [34](#)
- EQUAL
  - FT800.h, [34](#)
- F16
  - FT800.h, [34](#)
- FT800.cpp, [7](#)
  - delay\_ms, [8](#)
  - delay\_us, [8](#)
  - ft800cmdWrite, [9](#)
  - ft800memRead16, [10](#)
  - ft800memRead32, [11](#)
  - ft800memRead8, [11](#)
  - ft800memWrite16, [12](#)
  - ft800memWrite32, [13](#)
  - ft800memWrite8, [14](#)
  - getData, [15](#)
  - incCMDOffset, [15](#)
  - sendData, [16](#)
- FT800.h, [17](#)
  - ABS, [24](#)
  - BLACK, [24](#)
  - BLUE, [24](#)
  - CLR\_COL, [24](#)
  - CLR\_STN, [24](#)
  - CLR\_TAG, [24](#)
  - CMD\_APPEND, [24](#)
  - CMD\_BGCOLOR, [24](#)
  - CMD\_BUTTON, [25](#)
  - CMD\_CALIBRATE, [25](#)
  - CMD\_CLOCK, [25](#)
  - CMD\_COLDSTART, [25](#)
  - CMD\_DIAL, [25](#)
  - CMD\_DLSTART, [25](#)
  - CMD\_FGCOLOR, [25](#)
  - CMD\_GAUGE, [25](#)
  - CMD\_GETMATRIX, [25](#)
  - CMD\_GETPTR, [25](#)
  - CMD\_GRADCOLOR, [26](#)
  - CMD\_GRADIENT, [26](#)
  - CMD\_INFLATE, [26](#)
  - CMD\_INTERRUPT, [26](#)
  - CMD\_KEYS, [26](#)
  - CMD\_LOADIDENTITY, [26](#)
  - CMD\_LOADIMAGE, [26](#)
  - CMD\_LOGO, [26](#)
  - CMD\_MEMCPY, [26](#)
  - CMD\_MEMCRC, [26](#)
  - CMD\_MEMSET, [27](#)
  - CMD\_MEMWRITE, [27](#)
  - CMD\_MEMZERO, [27](#)
  - CMD\_NUMBER, [27](#)
  - CMD\_PROGRESS, [27](#)
  - CMD\_REGREAD, [27](#)
  - CMD\_ROTATE, [27](#)
  - CMD\_SCALE, [27](#)
  - CMD\_SCREENSAVER, [27](#)
  - CMD\_SCROLLBAR, [27](#)
  - CMD\_SETFONT, [28](#)
  - CMD\_SETMATRIX, [28](#)
  - CMD\_SKETCH, [28](#)
  - CMD\_SLIDER, [28](#)
  - CMD\_SNAPSHOT, [28](#)
  - CMD\_SPINNER, [28](#)
  - CMD\_STOP, [28](#)
  - CMD\_SWAP, [28](#)
  - CMD\_TEXT, [28](#)
  - CMD\_TOGGLE, [28](#)
  - CMD\_TRACK, [29](#)
  - CMD\_TRANSLATE, [29](#)
  - CMDBUF\_SIZE, [29](#)
  - DECR\_WRAP, [29](#)
  - DECR, [29](#)
  - DL\_ALPHA\_FUNC, [29](#)

DL\_BEGIN, [29](#)  
DL\_BITMAP\_HANDLE, [29](#)  
DL\_BITMAP\_LAYOUT, [29](#)  
DL\_BITMAP\_SIZE, [29](#)  
DL\_BITMAP\_SOURCE, [30](#)  
DL\_BITMAP\_TFORM\_A, [30](#)  
DL\_BITMAP\_TFORM\_B, [30](#)  
DL\_BITMAP\_TFORM\_C, [30](#)  
DL\_BITMAP\_TFORM\_D, [30](#)  
DL\_BITMAP\_TFORM\_E, [30](#)  
DL\_BITMAP\_TFORM\_F, [30](#)  
DL\_BLEND\_FUNC, [30](#)  
DL\_CALL, [30](#)  
DL\_CELL, [30](#)  
DL\_CLEAR\_RGB, [31](#)  
DL\_CLEAR\_STENCIL, [31](#)  
DL\_CLEAR\_TAG, [31](#)  
DL\_CLEAR, [31](#)  
DL\_COLOR\_MASK, [31](#)  
DL\_COLOR\_RGB, [31](#)  
DL\_COLOR\_A, [31](#)  
DL\_DISPLAY, [31](#)  
DL\_END, [31](#)  
DL\_JUMP, [31](#)  
DL\_LINE\_WIDTH, [32](#)  
DL\_MACRO, [32](#)  
DL\_POINT\_SIZE, [32](#)  
DL\_RESTORE\_CONTEXT, [32](#)  
DL\_RETURN, [32](#)  
DL\_SAVE\_CONTEXT, [32](#)  
DL\_SCISSOR\_SIZE, [32](#)  
DL\_SCISSOR\_XY, [32](#)  
DL\_STENCIL\_FUNC, [32](#)  
DL\_STENCIL\_MASK, [32](#)  
DL\_STENCIL\_OP, [33](#)  
DL\_TAG\_MASK, [33](#)  
DL\_TAG, [33](#)  
DL\_VERTEX2I, [33](#)  
DL\_VERTEX2F, [33](#)  
DLSWAP\_DONE, [33](#)  
DLSWAP\_FRAME, [33](#)  
DLSWAP\_LINE, [33](#)  
DST\_ALPHA, [33](#)  
delay\_ms, [51](#)  
delay\_us, [52](#)  
EDGE\_STRIP\_A, [33](#)  
EDGE\_STRIP\_B, [34](#)  
EDGE\_STRIP\_L, [34](#)  
EDGE\_STRIP\_R, [34](#)  
EQUAL, [34](#)  
F16, [34](#)  
FT800\_ACTIVE, [34](#)  
FT800\_CLK36M, [34](#)  
FT800\_CLK48M, [34](#)  
FT800\_CLKEXT, [34](#)  
FT800\_CORERST, [35](#)  
FT800\_GPUACTIVE, [35](#)  
FT800\_PWRDOWN, [35](#)  
FT800\_SLEEP, [35](#)  
FT800\_STANDBY, [35](#)  
FT800\_VERSION, [35](#)  
FT\_CMD\_FIFO\_SIZE, [35](#)  
FT\_CMD\_SIZE, [35](#)  
FT\_DL\_SIZE, [36](#)  
FTPOINTS, [36](#)  
ft800cmdWrite, [53](#)  
ft800memRead16, [54](#)  
ft800memRead32, [55](#)  
ft800memRead8, [55](#)  
ft800memWrite16, [56](#)  
ft800memWrite32, [57](#)  
ft800memWrite8, [58](#)  
GEQUAL, [36](#)  
GREATER, [36](#)  
GREEN, [36](#)  
getData, [59](#)  
INCR\_WRAP, [36](#)  
INCR, [36](#)  
INT\_CMDEMPTY, [36](#)  
INT\_CMDFLAG, [36](#)  
INT\_CONVCOMPLETE, [37](#)  
INT\_PLAYBACK, [37](#)  
INT\_SOUND, [37](#)  
INT\_SWAP, [37](#)  
INT\_TAG, [37](#)  
INT\_TOUCH, [37](#)  
INVALID\_TOUCH\_XY, [37](#)  
INVERT, [37](#)  
incCMDOffset, [59](#)  
KEEP, [37](#)  
L1, [37](#)  
L4, [38](#)  
L8, [38](#)  
LCD\_QVGA, [38](#)  
LEQUAL, [38](#)  
LESS, [38](#)  
LINE\_STRIP, [38](#)  
LINEAR\_SAMPLES, [38](#)  
LINES, [38](#)  
MAX, [38](#)  
MEM\_READ, [38](#)  
MEM\_WRITE, [39](#)  
MIN, [39](#)  
NEAREST, [39](#)  
NEVER, [39](#)  
NOTEQUAL, [39](#)  
NOTE, [39](#)  
ONE\_MINUS\_DST\_ALPHA, [39](#)  
ONE\_MINUS\_SRC\_ALPHA, [39](#)  
ONE, [39](#)  
OPT\_CENTERX, [40](#)  
OPT\_CENTERY, [40](#)  
OPT\_CENTER, [40](#)  
OPT\_FLAT, [40](#)  
OPT\_MONO, [40](#)  
OPT\_NOBACK, [40](#)

OPT\_NODL, [40](#)  
OPT\_NOHANDS, [40](#)  
OPT\_NOHM, [40](#)  
OPT\_NOPOINTER, [40](#)  
OPT\_NOSECS, [41](#)  
OPT\_NOTICKS, [41](#)  
OPT\_RIGHTX, [41](#)  
OPT\_SIGNED, [41](#)  
PALETTERED, [41](#)  
PLAYCOLOR, [41](#)  
RAM\_CMD, [41](#)  
RAM\_DL, [41](#)  
RAM\_PAL, [41](#)  
RAM\_REG, [42](#)  
RAM\_G, [41](#)  
RECTS, [42](#)  
REG\_CLOCK, [42](#)  
REG\_CMD\_DL, [42](#)  
REG\_CMD\_READ, [42](#)  
REG\_CMD\_WRITE, [42](#)  
REG\_CPURESET, [42](#)  
REG\_CSPREAD, [42](#)  
REG\_DITHER, [42](#)  
REG\_DLSWAP, [43](#)  
REG\_FRAMES, [43](#)  
REG\_FREQUENCY, [43](#)  
REG\_GPIO\_DIR, [43](#)  
REG\_GPIO, [43](#)  
REG\_HCYCLE, [43](#)  
REG\_HOFFSET, [43](#)  
REG\_HSIZE, [43](#)  
REG\_HSYNC0, [43](#)  
REG\_HSYNC1, [43](#)  
REG\_INT\_EN, [44](#)  
REG\_INT\_FLAGS, [44](#)  
REG\_INT\_MASK, [44](#)  
REG\_ID, [44](#)  
REG\_MACRO\_0, [44](#)  
REG\_MACRO\_1, [44](#)  
REG\_OUTBITS, [44](#)  
REG\_PCLK\_POL, [44](#)  
REG\_PCLK, [44](#)  
REG\_PLAYBACK\_FORMAT, [45](#)  
REG\_PLAYBACK\_FREQ, [45](#)  
REG\_PLAYBACK\_LENGTH, [45](#)  
REG\_PLAYBACK\_LOOP, [45](#)  
REG\_PLAYBACK\_PLAY, [45](#)  
REG\_PLAYBACK\_READPTR, [45](#)  
REG\_PLAYBACK\_START, [45](#)  
REG\_PLAY, [44](#)  
REG\_PWM\_DUTY, [45](#)  
REG\_PWM\_HZ, [45](#)  
REG\_RENDERMODE, [45](#)  
REG\_ROTATE, [46](#)  
REG\_SNAPSHOT, [46](#)  
REG\_SNAPY, [46](#)  
REG\_SOUND, [46](#)  
REG\_SWIZZLE, [46](#)  
REG\_TAG\_X, [46](#)  
REG\_TAG\_Y, [46](#)  
REG\_TAP\_CRC, [46](#)  
REG\_TAP\_MASK, [46](#)  
REG\_TAG, [46](#)  
REG\_TOUCH\_ADC\_MODE, [47](#)  
REG\_TOUCH\_CHARGE, [47](#)  
REG\_TOUCH\_DIRECT\_XY, [47](#)  
REG\_TOUCH\_DIRECT\_Z1Z2, [47](#)  
REG\_TOUCH\_MODE, [47](#)  
REG\_TOUCH\_OVERSAMPLE, [47](#)  
REG\_TOUCH\_RAW\_XY, [47](#)  
REG\_TOUCH\_RZTHRESH, [47](#)  
REG\_TOUCH\_RZ, [47](#)  
REG\_TOUCH\_SCREEN\_XY, [47](#)  
REG\_TOUCH\_SETTLE, [48](#)  
REG\_TOUCH\_TAG\_XY, [48](#)  
REG\_TOUCH\_TAG, [48](#)  
REG\_TOUCH\_TRANSFORM\_A, [48](#)  
REG\_TOUCH\_TRANSFORM\_B, [48](#)  
REG\_TOUCH\_TRANSFORM\_C, [48](#)  
REG\_TOUCH\_TRANSFORM\_D, [48](#)  
REG\_TOUCH\_TRANSFORM\_E, [48](#)  
REG\_TOUCH\_TRANSFORM\_F, [48](#)  
REG\_TRACKER, [48](#)  
REG\_VCYCLE, [49](#)  
REG\_VOFFSET, [49](#)  
REG\_VOL\_PB, [49](#)  
REG\_VOL\_SOUND, [49](#)  
REG\_VSIZE, [49](#)  
REG\_VSYNC0, [49](#)  
REG\_VSYNC1, [49](#)  
REPEAT, [49](#)  
REPLACE, [49](#)  
RED, [42](#)  
RGB332, [50](#)  
RGB565, [50](#)  
RGB, [49](#)  
SRC\_ALPHA, [50](#)  
sendData, [60](#)  
SQ, [50](#)  
TEXT8X8, [50](#)  
TEXTVGA, [50](#)  
TOUCHMODE\_CONTINUOUS, [50](#)  
TOUCHMODE\_FRAME, [50](#)  
TOUCHMODE\_OFF, [50](#)  
TOUCHMODE\_ONESHOT, [50](#)  
ULAW\_SAMPLES, [51](#)  
WHITE, [51](#)  
xCS, [51](#)  
xPD, [51](#)  
xSDI, [51](#)  
xSDO, [51](#)  
xclock, [51](#)  
ZERO, [51](#)  
FT800\_ACTIVE  
    FT800.h, [34](#)  
FT800\_CLK36M

- FT800.h, [34](#)
- FT800\_CLK48M
  - FT800.h, [34](#)
- FT800\_CLKEXT
  - FT800.h, [34](#)
- FT800\_CORERST
  - FT800.h, [35](#)
- FT800\_GPUACTIVE
  - FT800.h, [35](#)
- FT800\_PWRDOWN
  - FT800.h, [35](#)
- FT800\_SLEEP
  - FT800.h, [35](#)
- FT800\_STANDBY
  - FT800.h, [35](#)
- FT800\_VERSION
  - FT800.h, [35](#)
- FT800api.cpp, [61](#)
  - autko, [63](#)
  - button, [63](#)
  - calibrate, [64](#)
  - dot, [64](#)
  - initScreen, [65](#)
  - line, [66](#)
  - mainScreen, [67](#)
  - number, [67](#)
  - optionsScreen, [68](#)
  - show, [68](#)
  - smartMirrorScreen, [69](#)
  - spinner, [70](#)
  - start, [70](#)
  - text, [71](#)
- FT800api.h, [72](#)
  - audi, [85](#)
  - autko, [74](#)
  - button, [75](#)
  - calibrate, [76](#)
  - cmdBufferRd, [85](#)
  - cmdBufferWr, [85](#)
  - cmdOffset, [85](#)
  - dot, [76](#)
  - initScreen, [77](#)
  - line, [77](#)
  - mainScreen, [79](#)
  - number, [80](#)
  - optionsScreen, [80](#)
  - show, [81](#)
  - smartMirrorScreen, [82](#)
  - spinner, [82](#)
  - start, [83](#)
  - text, [84](#)
  - timeR, [85](#)
- FT\_CMD\_FIFO\_SIZE
  - FT800.h, [35](#)
- FT\_CMD\_SIZE
  - FT800.h, [35](#)
- FT\_DL\_SIZE
  - FT800.h, [36](#)
- FTPOINTS
  - FT800.h, [36](#)
- ft800Gpio
  - VM800Galileo.cpp, [105](#)
- ft800cmdWrite
  - FT800.cpp, [9](#)
  - FT800.h, [53](#)
- ft800memRead16
  - FT800.cpp, [10](#)
  - FT800.h, [54](#)
- ft800memRead32
  - FT800.cpp, [11](#)
  - FT800.h, [55](#)
- ft800memRead8
  - FT800.cpp, [11](#)
  - FT800.h, [55](#)
- ft800memWrite16
  - FT800.cpp, [12](#)
  - FT800.h, [56](#)
- ft800memWrite32
  - FT800.cpp, [13](#)
  - FT800.h, [57](#)
- ft800memWrite8
  - FT800.cpp, [14](#)
  - FT800.h, [58](#)
- GEQUAL
  - FT800.h, [36](#)
- GREATER
  - FT800.h, [36](#)
- GREEN
  - FT800.h, [36](#)
- getData
  - FT800.cpp, [15](#)
  - FT800.h, [59](#)
- I2C.cpp, [86](#)
  - d, [87](#)
  - readPCF, [86](#)
- I2C.h, [87](#)
  - pinInt0, [88](#)
  - readPCF, [89](#)
  - scl, [88](#)
  - sda, [89](#)
- INCR\_WRAP
  - FT800.h, [36](#)
- INCR
  - FT800.h, [36](#)
- INT\_CMDEMPTY
  - FT800.h, [36](#)
- INT\_CMDFLAG
  - FT800.h, [36](#)
- INT\_CONVCOMPLETE
  - FT800.h, [37](#)
- INT\_PLAYBACK
  - FT800.h, [37](#)
- INT\_SOUND
  - FT800.h, [37](#)
- INT\_SWAP

- FT800.h, [37](#)
- INT\_TAG
  - FT800.h, [37](#)
- INT\_TOUCH
  - FT800.h, [37](#)
- INVALID\_TOUCH\_XY
  - FT800.h, [37](#)
- INVERT
  - FT800.h, [37](#)
- incCMDOffset
  - FT800.cpp, [15](#)
  - FT800.h, [59](#)
- initScreen
  - FT800api.cpp, [65](#)
  - FT800api.h, [77](#)
- KEEP
  - FT800.h, [37](#)
- L1
  - FT800.h, [37](#)
- L4
  - FT800.h, [38](#)
- L8
  - FT800.h, [38](#)
- LCD\_QVGA
  - FT800.h, [38](#)
- LEQUAL
  - FT800.h, [38](#)
- LESS
  - FT800.h, [38](#)
- LINE\_STRIP
  - FT800.h, [38](#)
- LINEAR\_SAMPLES
  - FT800.h, [38](#)
- LINES
  - FT800.h, [38](#)
- lcdHcycle
  - VM800Galileo.cpp, [105](#)
- lcdHeight
  - VM800Galileo.cpp, [105](#)
- lcdHoffset
  - VM800Galileo.cpp, [105](#)
- lcdHsync0
  - VM800Galileo.cpp, [105](#)
- lcdHsync1
  - VM800Galileo.cpp, [105](#)
- lcdPclk
  - VM800Galileo.cpp, [106](#)
- lcdPclkpol
  - VM800Galileo.cpp, [106](#)
- lcdSwizzle
  - VM800Galileo.cpp, [106](#)
- lcdVcycle
  - VM800Galileo.cpp, [106](#)
- lcdVoffset
  - VM800Galileo.cpp, [106](#)
- lcdVsync0
  - VM800Galileo.cpp, [106](#)
- lcdVsync1
  - VM800Galileo.cpp, [106](#)
- lcdWidth
  - VM800Galileo.cpp, [106](#)
- lights
  - car, [5](#)
- line
  - FT800api.cpp, [66](#)
  - FT800api.h, [77](#)
- loop
  - VM800Galileo.cpp, [103](#)
- MAX
  - FT800.h, [38](#)
- MEM\_READ
  - FT800.h, [38](#)
- MEM\_WRITE
  - FT800.h, [39](#)
- MIN
  - FT800.h, [39](#)
- mainScreen
  - FT800api.cpp, [67](#)
  - FT800api.h, [79](#)
- NEAREST
  - FT800.h, [39](#)
- NEVER
  - FT800.h, [39](#)
- NOTEQUAL
  - FT800.h, [39](#)
- NOTE
  - FT800.h, [39](#)
- number
  - FT800api.cpp, [67](#)
  - FT800api.h, [80](#)
- ONE\_MINUS\_DST\_ALPHA
  - FT800.h, [39](#)
- ONE\_MINUS\_SRC\_ALPHA
  - FT800.h, [39](#)
- ONE
  - FT800.h, [39](#)
- OPT\_CENTERX
  - FT800.h, [40](#)
- OPT\_CENTERY
  - FT800.h, [40](#)
- OPT\_CENTER
  - FT800.h, [40](#)
- OPT\_FLAT
  - FT800.h, [40](#)
- OPT\_MONO
  - FT800.h, [40](#)
- OPT\_NOBACK
  - FT800.h, [40](#)
- OPT\_NODL
  - FT800.h, [40](#)
- OPT\_NOHANDS
  - FT800.h, [40](#)
- OPT\_NOHM

- FT800.h, [40](#)
- OPT\_NOPOINTER
  - FT800.h, [40](#)
- OPT\_NOSECS
  - FT800.h, [41](#)
- OPT\_NOTICKS
  - FT800.h, [41](#)
- OPT\_RIGHTX
  - FT800.h, [41](#)
- OPT\_SIGNED
  - FT800.h, [41](#)
- optionsScreen
  - FT800api.cpp, [68](#)
  - FT800api.h, [80](#)
- PALETTED
  - FT800.h, [41](#)
- PLAYCOLOR
  - FT800.h, [41](#)
- pinInt0
  - I2C.h, [88](#)
- printObj
  - simulator.cpp, [92](#)
  - simulator.h, [98](#)
- r
  - car, [6](#)
- RAM\_CMD
  - FT800.h, [41](#)
- RAM\_DL
  - FT800.h, [41](#)
- RAM\_PAL
  - FT800.h, [41](#)
- RAM\_REG
  - FT800.h, [42](#)
- RAM\_G
  - FT800.h, [41](#)
- RECTS
  - FT800.h, [42](#)
- REG\_CLOCK
  - FT800.h, [42](#)
- REG\_CMD\_DL
  - FT800.h, [42](#)
- REG\_CMD\_READ
  - FT800.h, [42](#)
- REG\_CMD\_WRITE
  - FT800.h, [42](#)
- REG\_CPURESET
  - FT800.h, [42](#)
- REG\_CSPREAD
  - FT800.h, [42](#)
- REG\_DITHER
  - FT800.h, [42](#)
- REG\_DLSWAP
  - FT800.h, [43](#)
- REG\_FRAMES
  - FT800.h, [43](#)
- REG\_FREQUENCY
  - FT800.h, [43](#)
- REG\_GPIO\_DIR
  - FT800.h, [43](#)
- REG\_GPIO
  - FT800.h, [43](#)
- REG\_HCYCLE
  - FT800.h, [43](#)
- REG\_HOFFSET
  - FT800.h, [43](#)
- REG\_HSIZE
  - FT800.h, [43](#)
- REG\_HSYNC0
  - FT800.h, [43](#)
- REG\_HSYNC1
  - FT800.h, [43](#)
- REG\_INT\_EN
  - FT800.h, [44](#)
- REG\_INT\_FLAGS
  - FT800.h, [44](#)
- REG\_INT\_MASK
  - FT800.h, [44](#)
- REG\_ID
  - FT800.h, [44](#)
- REG\_MACRO\_0
  - FT800.h, [44](#)
- REG\_MACRO\_1
  - FT800.h, [44](#)
- REG\_OUTBITS
  - FT800.h, [44](#)
- REG\_PCLK\_POL
  - FT800.h, [44](#)
- REG\_PCLK
  - FT800.h, [44](#)
- REG\_PLAYBACK\_FORMAT
  - FT800.h, [45](#)
- REG\_PLAYBACK\_FREQ
  - FT800.h, [45](#)
- REG\_PLAYBACK\_LENGTH
  - FT800.h, [45](#)
- REG\_PLAYBACK\_LOOP
  - FT800.h, [45](#)
- REG\_PLAYBACK\_PLAY
  - FT800.h, [45](#)
- REG\_PLAYBACK\_READPTR
  - FT800.h, [45](#)
- REG\_PLAYBACK\_START
  - FT800.h, [45](#)
- REG\_PLAY
  - FT800.h, [44](#)
- REG\_PWM\_DUTY
  - FT800.h, [45](#)
- REG\_PWM\_HZ
  - FT800.h, [45](#)
- REG\_RENDERMODE
  - FT800.h, [45](#)
- REG\_ROTATE
  - FT800.h, [46](#)
- REG\_SNAPSHOT
  - FT800.h, [46](#)

REG\_SNAPY  
    FT800.h, [46](#)  
REG\_SOUND  
    FT800.h, [46](#)  
REG\_SWIZZLE  
    FT800.h, [46](#)  
REG\_TAG\_X  
    FT800.h, [46](#)  
REG\_TAG\_Y  
    FT800.h, [46](#)  
REG\_TAP\_CRC  
    FT800.h, [46](#)  
REG\_TAP\_MASK  
    FT800.h, [46](#)  
REG\_TAG  
    FT800.h, [46](#)  
REG\_TOUCH\_ADC\_MODE  
    FT800.h, [47](#)  
REG\_TOUCH\_CHARGE  
    FT800.h, [47](#)  
REG\_TOUCH\_DIRECT\_XY  
    FT800.h, [47](#)  
REG\_TOUCH\_DIRECT\_Z1Z2  
    FT800.h, [47](#)  
REG\_TOUCH\_MODE  
    FT800.h, [47](#)  
REG\_TOUCH\_OVERSAMPLE  
    FT800.h, [47](#)  
REG\_TOUCH\_RAW\_XY  
    FT800.h, [47](#)  
REG\_TOUCH\_RZTHRESH  
    FT800.h, [47](#)  
REG\_TOUCH\_RZ  
    FT800.h, [47](#)  
REG\_TOUCH\_SCREEN\_XY  
    FT800.h, [47](#)  
REG\_TOUCH\_SETTLE  
    FT800.h, [48](#)  
REG\_TOUCH\_TAG\_XY  
    FT800.h, [48](#)  
REG\_TOUCH\_TAG  
    FT800.h, [48](#)  
REG\_TOUCH\_TRANSFORM\_A  
    FT800.h, [48](#)  
REG\_TOUCH\_TRANSFORM\_B  
    FT800.h, [48](#)  
REG\_TOUCH\_TRANSFORM\_C  
    FT800.h, [48](#)  
REG\_TOUCH\_TRANSFORM\_D  
    FT800.h, [48](#)  
REG\_TOUCH\_TRANSFORM\_E  
    FT800.h, [48](#)  
REG\_TOUCH\_TRANSFORM\_F  
    FT800.h, [48](#)  
REG\_TRACKER  
    FT800.h, [48](#)  
REG\_VCYCLE  
    FT800.h, [49](#)  
REG\_VOFFSET  
    FT800.h, [49](#)  
REG\_VOL\_PB  
    FT800.h, [49](#)  
REG\_VOL\_SOUND  
    FT800.h, [49](#)  
REG\_VSIZE  
    FT800.h, [49](#)  
REG\_VSYNC0  
    FT800.h, [49](#)  
REG\_VSYNC1  
    FT800.h, [49](#)  
REPEAT  
    FT800.h, [49](#)  
REPLACE  
    FT800.h, [49](#)  
RED  
    FT800.h, [42](#)  
RGB332  
    FT800.h, [50](#)  
RGB565  
    FT800.h, [50](#)  
RGB  
    FT800.h, [49](#)  
ramCommandBuffer  
    VM800Galileo.cpp, [107](#)  
ramDisplayList  
    VM800Galileo.cpp, [107](#)  
readData  
    simulator.cpp, [92](#)  
    simulator.h, [98](#)  
readPCF  
    I2C.cpp, [86](#)  
    I2C.h, [89](#)  
readTemp  
    simulator.cpp, [93](#)  
    simulator.h, [99](#)  
SRC\_ALPHA  
    FT800.h, [50](#)  
save  
    simulator.cpp, [94](#)  
    simulator.h, [100](#)  
saveData  
    simulator.h, [101](#)  
    VM800Galileo.cpp, [107](#)  
scl  
    I2C.h, [88](#)  
screenNR  
    simulator.h, [101](#)  
    VM800Galileo.cpp, [107](#)  
sda  
    I2C.h, [89](#)  
seatbelts  
    car, [6](#)  
sendData  
    FT800.cpp, [16](#)  
    FT800.h, [60](#)  
    simulator.cpp, [94](#)

- simulator.h, [100](#)
- setup
  - VM800Galileo.cpp, [103](#)
- show
  - FT800api.cpp, [68](#)
  - FT800api.h, [81](#)
- simulator.cpp, [89](#)
  - checkChangesAnalog, [91](#)
  - checkChangesDigital, [91](#)
  - printObj, [92](#)
  - readData, [92](#)
  - readTemp, [93](#)
  - save, [94](#)
  - sendData, [94](#)
- simulator.h, [95](#)
  - audi, [101](#)
  - checkChangesAnalog, [97](#)
  - checkChangesDigital, [97](#)
  - dataFormat, [101](#)
  - printObj, [98](#)
  - readData, [98](#)
  - readTemp, [99](#)
  - save, [100](#)
  - saveData, [101](#)
  - screenNR, [101](#)
  - sendData, [100](#)
- smartMirrorScreen
  - FT800api.cpp, [69](#)
  - FT800api.h, [82](#)
- spinner
  - FT800api.cpp, [70](#)
  - FT800api.h, [82](#)
- SQ
  - FT800.h, [50](#)
- start
  - FT800api.cpp, [70](#)
  - FT800api.h, [83](#)
- TEXT8X8
  - FT800.h, [50](#)
- TEXTVGA
  - FT800.h, [50](#)
- TOUCHMODE\_CONTINUOUS
  - FT800.h, [50](#)
- TOUCHMODE\_FRAME
  - FT800.h, [50](#)
- TOUCHMODE\_OFF
  - FT800.h, [50](#)
- TOUCHMODE\_ONESHOT
  - FT800.h, [50](#)
- tempEngine
  - car, [6](#)
- templn
  - car, [6](#)
- tempOut
  - car, [6](#)
- text
  - FT800api.cpp, [71](#)
  - FT800api.h, [84](#)
- timeR
  - FT800api.h, [85](#)
  - VM800Galileo.cpp, [107](#)
- ULAW\_SAMPLES
  - FT800.h, [51](#)
- VM800Galileo.cpp, [102](#)
  - audi, [104](#)
  - buf, [104](#)
  - cmdBufferRd, [104](#)
  - cmdBufferWr, [104](#)
  - cmdOffset, [105](#)
  - dataFormat, [105](#)
  - ft800Gpio, [105](#)
  - lcdHcycle, [105](#)
  - lcdHeight, [105](#)
  - lcdHoffset, [105](#)
  - lcdHsync0, [105](#)
  - lcdHsync1, [105](#)
  - lcdPclk, [106](#)
  - lcdPclkpol, [106](#)
  - lcdSwizzle, [106](#)
  - lcdVcycle, [106](#)
  - lcdVoffset, [106](#)
  - lcdVsync0, [106](#)
  - lcdVsync1, [106](#)
  - lcdWidth, [106](#)
  - loop, [103](#)
  - ramCommandBuffer, [107](#)
  - ramDisplayList, [107](#)
  - saveData, [107](#)
  - screenNR, [107](#)
  - setup, [103](#)
  - timeR, [107](#)
- WHITE
  - FT800.h, [51](#)
- xCS
  - FT800.h, [51](#)
- xPD
  - FT800.h, [51](#)
- xSDI
  - FT800.h, [51](#)
- xSDO
  - FT800.h, [51](#)
- xclock
  - FT800.h, [51](#)
- ZERO
  - FT800.h, [51](#)