```
 1
 2  #ifndef _FT800_H_
 3  #define _FT800_H_
 4
 5  /* FT800 Power Modes */
 6  #define CMD_ACTIVE  0x00
 7  #define CMD_STANDBY 0x41
 8  #define CMD_SLEEP   0x42
 9  #define CMD_PWRDOWN 0x50
10
11  /* FT800 Clock Switching */
12  #define CMD_CLKINT 0x48
13  #define CMD_CLKEXT 0x44
14  #define CMD_CLK48M 0x62
15  #define CMD_CLK36M 0x61
16
17  /* MISC */
18  #define CMD_CORERST 0x68
19
20
21  /* Definitions used for FT800 co processor command buffer */
22  #define FT_DL_SIZE           (8*1024)  //8KB Display List buffer size
23  #define FT_CMD_FIFO_SIZE     (4*1024)  //4KB coprocessor Fifo size
24  #define FT_CMD_SIZE          (4)       //4 byte per coprocessor command of ↵
       EVE
25
26  #define FT800_VERSION "1.9.0"
27  #define ADC_DIFFERENTIAL     1UL
28  #define ADC_SINGLE_ENDED     0UL
29  #define ADPCM_SAMPLES        2UL
30  #define ALWAYS               7UL
31  #define ARGB1555             0UL
32  #define ARGB2                5UL
33  #define ARGB4                6UL
34  #define BARGRAPH             11UL
35  #define BILINEAR             1UL
36  #define BITMAPS              1UL
37  #define BORDER               0UL
38
39  #define CMDBUF_SIZE          4096UL
40  #define CMD_APPEND           4294967070UL
41  #define CMD_BGCOLOR          4294967049UL
42  #define CMD_BITMAP_TRANSFORM 4294967073UL
43  #define CMD_BUTTON           4294967053UL
44  #define CMD_CALIBRATE        4294967061UL
45  #define CMD_CLOCK            4294967060UL
46  #define CMD_COLDSTART        4294967090UL
47  #define CMD_CRC              4294967043UL
48  #define CMD_DIAL             4294967085UL
49  #define CMD_DLSTART          4294967040UL
50  #define CMD_EXECUTE          4294967047UL
51  #define CMD_FGCOLOR          4294967050UL
52  #define CMD_GAUGE            4294967059UL
```

```
 53 #define CMD_GETMATRIX         4294967091UL
 54 #define CMD_GETPOINT          4294967048UL
 55 #define CMD_GETPROPS          4294967077UL
 56 #define CMD_GETPTR            4294967075UL
 57 #define CMD_GRADCOLOR         4294967092UL
 58 #define CMD_GRADIENT          4294967051UL
 59 #define CMD_HAMMERAUX         4294967044UL
 60 #define CMD_IDCT              4294967046UL
 61 #define CMD_INFLATE           4294967074UL
 62 #define CMD_INTERRUPT         4294967042UL
 63 #define CMD_KEYS              4294967054UL
 64 #define CMD_LOADIDENTITY      4294967078UL
 65 #define CMD_LOADIMAGE         4294967076UL
 66 #define CMD_LOGO              4294967089UL
 67 #define CMD_MARCH             4294967045UL
 68 #define CMD_MEMCPY            4294967069UL
 69 #define CMD_MEMCRC            4294967064UL
 70 #define CMD_MEMSET            4294967067UL
 71 #define CMD_MEMWRITE          4294967066UL
 72 #define CMD_MEMZERO           4294967068UL
 73 #define CMD_NUMBER            4294967086UL
 74 #define CMD_PROGRESS          4294967055UL
 75 #define CMD_REGREAD           4294967065UL
 76 #define CMD_ROTATE            4294967081UL
 77 #define CMD_SCALE             4294967080UL
 78 #define CMD_SCREENSAVER       4294967087UL
 79 #define CMD_SCROLLBAR         4294967057UL
 80 #define CMD_SETFONT           4294967083UL
 81 #define CMD_SETMATRIX         4294967082UL
 82 #define CMD_SKETCH            4294967088UL
 83 #define CMD_SLIDER            4294967056UL
 84 #define CMD_SNAPSHOT          4294967071UL
 85 #define CMD_SPINNER           4294967062UL
 86 #define CMD_STOP              4294967063UL
 87 #define CMD_SWAP              4294967041UL
 88 #define CMD_TEXT              4294967052UL
 89 #define CMD_TOGGLE            4294967058UL
 90 #define CMD_TOUCH_TRANSFORM   4294967072UL
 91 #define CMD_TRACK             4294967084UL
 92 #define CMD_TRANSLATE         4294967079UL
 93
 94 #define DECR                  4UL
 95 #define DECR_WRAP             7UL
 96 #define DLSWAP_DONE           0UL
 97 #define DLSWAP_FRAME          2UL
 98 #define DLSWAP_LINE           1UL
 99 #define DST_ALPHA             3UL
100 #define EDGE_STRIP_A          7UL
101 #define EDGE_STRIP_B          8UL
102 #define EDGE_STRIP_L          6UL
103 #define EDGE_STRIP_R          5UL
104 #define EQUAL                 5UL
105 #define GEQUAL                4UL
```

```
106 #define GREATER               3UL
107 #define INCR                  3UL
108 #define INCR_WRAP             6UL
109 #define INT_CMDEMPTY          32UL
110 #define INT_CMDFLAG           64UL
111 #define INT_CONVCOMPLETE      128UL
112 #define INT_PLAYBACK          16UL
113 #define INT_SOUND             8UL
114 #define INT_SWAP              1UL
115 #define INT_TAG               4UL
116 #define INT_TOUCH             2UL
117 #define INVERT                5UL
118
119 #define KEEP                  1UL
120 #define L1                    1UL
121 #define L4                    2UL
122 #define L8                    3UL
123 #define LEQUAL                2UL
124 #define LESS                  1UL
125 #define LINEAR_SAMPLES        0UL
126 #define LINES                 3UL
127 #define LINE_STRIP            4UL
128 #define NEAREST               0UL
129 #define NEVER                 0UL
130 #define NOTEQUAL              6UL
131 #define ONE                   1UL
132 #define ONE_MINUS_DST_ALPHA   5UL
133 #define ONE_MINUS_SRC_ALPHA   4UL
134 #define OPT_CENTER            1536UL
135 #define OPT_CENTERX           512UL
136 #define OPT_CENTERY           1024UL
137 #define OPT_FLAT              256UL
138 #define OPT_MONO              1UL
139 #define OPT_NOBACK            4096UL
140 #define OPT_NODL              2UL
141 #define OPT_NOHANDS           49152UL
142 #define OPT_NOHM              16384UL
143 #define OPT_NOPOINTER         16384UL
144 #define OPT_NOSECS            32768UL
145 #define OPT_NOTICKS           8192UL
146 #define OPT_RIGHTX            2048UL
147 #define OPT_SIGNED            256UL
148 #define PALETTED              8UL
149 #define FTPOINTS              2UL
150 #define RECTS                 9UL
151
152 #define RAM_CMD               1081344UL
153 #define RAM_DL                1048576UL
154 #define RAM_G                 0UL
155 #define RAM_PAL               1056768UL
156 #define RAM_REG               1057792UL
157
158
```

```
159
160 #define REG_ANALOG            1058104UL
161 #define REG_ANA_COMP          1058160UL
162 #define REG_CLOCK             1057800UL
163 #define REG_CMD_DL            1058028UL
164 #define REG_CMD_READ          1058020UL
165 #define REG_CMD_WRITE         1058024UL
166 #define REG_CPURESET          1057820UL
167 #define REG_CRC               1058152UL
168 #define REG_CSPREAD           1057892UL
169 #define REG_CYA0              1058000UL
170 #define REG_CYA1              1058004UL
171 #define REG_CYA_TOUCH         1058100UL
172 #define REG_DATESTAMP         1058108UL
173 #define REG_DITHER            1057884UL
174 #define REG_DLSWAP            1057872UL
175 #define REG_FRAMES            1057796UL
176 #define REG_FREQUENCY         1057804UL
177 #define REG_GPIO              1057936UL
178 #define REG_GPIO_DIR          1057932UL
179 #define REG_HCYCLE            1057832UL
180 #define REG_HOFFSET           1057836UL
181 #define REG_HSIZE             1057840UL
182 #define REG_HSYNC0            1057844UL
183 #define REG_HSYNC1            1057848UL
184 #define REG_ID                1057792UL
185 #define REG_INT_EN            1057948UL
186 #define REG_INT_FLAGS         1057944UL
187 #define REG_INT_MASK          1057952UL
188 #define REG_MACRO_0           1057992UL
189 #define REG_MACRO_1           1057996UL
190 #define REG_OUTBITS           1057880UL
191 #define REG_PCLK              1057900UL
192 #define REG_PCLK_POL          1057896UL
193 #define REG_PLAY              1057928UL
194 #define REG_PLAYBACK_FORMAT   1057972UL
195 #define REG_PLAYBACK_FREQ     1057968UL
196 #define REG_PLAYBACK_LENGTH   1057960UL
197 #define REG_PLAYBACK_LOOP     1057976UL
198 #define REG_PLAYBACK_PLAY     1057980UL
199 #define REG_PLAYBACK_READPTR  1057964UL
200 #define REG_PLAYBACK_START    1057956UL
201 #define REG_PWM_DUTY          1057988UL
202 #define REG_PWM_HZ            1057984UL
203 #define REG_RENDERMODE        1057808UL
204 #define REG_ROMSUB_SEL        1058016UL
205 #define REG_ROTATE            1057876UL
206 #define REG_SNAPSHOT          1057816UL
207 #define REG_SNAPY             1057812UL
208 #define REG_SOUND             1057924UL
209 #define REG_SWIZZLE           1057888UL
210 #define REG_TAG               1057912UL
211 #define REG_TAG_X             1057904UL
```

```
212 #define REG_TAG_Y            1057908UL
213 #define REG_TAP_CRC          1057824UL
214 #define REG_TAP_MASK         1057828UL
215 #define REG_TOUCH_ADC_MODE   1058036UL
216 #define REG_TOUCH_CHARGE     1058040UL
217 #define REG_TOUCH_DIRECT_XY  1058164UL
218 #define REG_TOUCH_DIRECT_Z1Z2 1058168UL
219 #define REG_TOUCH_MODE       1058032UL
220 #define REG_TOUCH_OVERSAMPLE 1058048UL
221 #define REG_TOUCH_RAW_XY     1058056UL
222 #define REG_TOUCH_RZ         1058060UL
223 #define REG_TOUCH_RZTHRESH   1058052UL
224 #define REG_TOUCH_SCREEN_XY  1058064UL
225 #define REG_TOUCH_SETTLE     1058044UL
226 #define REG_TOUCH_TAG        1058072UL
227 #define REG_TOUCH_TAG_XY     1058068UL
228 #define REG_TOUCH_TRANSFORM_A 1058076UL
229 #define REG_TOUCH_TRANSFORM_B 1058080UL
230 #define REG_TOUCH_TRANSFORM_C 1058084UL
231 #define REG_TOUCH_TRANSFORM_D 1058088UL
232 #define REG_TOUCH_TRANSFORM_E 1058092UL
233 #define REG_TOUCH_TRANSFORM_F 1058096UL
234 #define REG_TRACKER          1085440UL
235 #define REG_TRIM             1058156UL
236 #define REG_VCYCLE           1057852UL
237 #define REG_VOFFSET          1057856UL
238 #define REG_VOL_PB           1057916UL
239 #define REG_VOL_SOUND        1057920UL
240 #define REG_VSIZE            1057860UL
241 #define REG_VSYNC0           1057864UL
242 #define REG_VSYNC1           1057868UL
243
244
245 #define REPEAT               1UL
246 #define REPLACE              2UL
247 #define RGB332               4UL
248 #define RGB565               7UL
249 #define SRC_ALPHA            2UL
250 #define TEXT8X8              9UL
251 #define TEXTVGA              10UL
252 #define TOUCHMODE_CONTINUOUS 3UL
253 #define TOUCHMODE_FRAME      2UL
254 #define TOUCHMODE_OFF        0UL
255 #define TOUCHMODE_ONESHOT    1UL
256 #define ULAW_SAMPLES         1UL
257 #define ZERO                 0UL
258
259 #define DEV_ID_LOC 0x102400
260
261 #define VERTEX2F(x,y) ((1UL<<30)|(((x)&32767UL)<<15)|(((y)&32767UL)<<0))
262 #define VERTEX2II(x,y,handle,cell) ((2UL<<30)|(((x)&511UL)<<21)|(((y)
      &511UL)<<12)|(((handle)&31UL)<<7)|(((cell)&127UL)<<0))
263 #define BITMAP_SOURCE(addr) ((1UL<<24)|(((addr)&1048575UL)<<0))
```

```
264 #define CLEAR_COLOR_RGB(red,green,blue) ((2UL<<24)|(((red)&255UL)<<16)|
      (((green)&255UL)<<8)|(((blue)&255UL)<<0))
265 #define TAG(s) ((3UL<<24)|(((s)&255UL)<<0))
266 #define COLOR_RGB(red,green,blue) ((4UL<<24)|(((red)&255UL)<<16)|(((green)
      &255UL)<<8)|(((blue)&255UL)<<0))
267 #define BITMAP_HANDLE(handle) ((5UL<<24)|(((handle)&31UL)<<0))
268 #define CELL(cell) ((6UL<<24)|(((cell)&127UL)<<0))
269 #define BITMAP_LAYOUT(format,linestride,height) ((7UL<<24)|(((format)
      &31UL)<<19)|(((linestride)&1023UL)<<9)|(((height)&511UL)<<0))
270 #define BITMAP_SIZE(filter,wrapx,wrapy,width,height) ((8UL<<24)|(((filter)
      &1UL)<<20)|(((wrapx)&1UL)<<19)|(((wrapy)&1UL)<<18)|(((width)&511UL)<<9)|
      (((height)&511UL)<<0))
271 #define ALPHA_FUNC(func,ref) ((9UL<<24)|(((func)&7UL)<<8)|(((ref)&255UL)
      <<0))
272 #define STENCIL_FUNC(func,ref,mask) ((10UL<<24)|(((func)&7UL)<<16)|(((ref)
      &255UL)<<8)|(((mask)&255UL)<<0))
273 #define BLEND_FUNC(src,dst) ((11UL<<24)|(((src)&7UL)<<3)|(((dst)&7UL)<<0))
274 #define STENCIL_OP(sfail,spass) ((12UL<<24)|(((sfail)&7UL)<<3)|(((spass)
      &7UL)<<0))
275 #define POINT_SIZE(size) ((13UL<<24)|(((size)&8191UL)<<0))
276 #define LINE_WIDTH(width) ((14UL<<24)|(((width)&4095UL)<<0))
277 #define CLEAR_COLOR_A(alpha) ((15UL<<24)|(((alpha)&255UL)<<0))
278 #define COLOR_A(alpha) ((16UL<<24)|(((alpha)&255UL)<<0))
279 #define CLEAR_STENCIL(s) ((17UL<<24)|(((s)&255UL)<<0))
280 #define CLEAR_TAG(s) ((18UL<<24)|(((s)&255UL)<<0))
281 #define STENCIL_MASK(mask) ((19UL<<24)|(((mask)&255UL)<<0))
282 #define TAG_MASK(mask) ((20UL<<24)|(((mask)&1UL)<<0))
283 #define BITMAP_TRANSFORM_A(a) ((21UL<<24)|(((a)&131071UL)<<0))
284 #define BITMAP_TRANSFORM_B(b) ((22UL<<24)|(((b)&131071UL)<<0))
285 #define BITMAP_TRANSFORM_C(c) ((23UL<<24)|(((c)&16777215UL)<<0))
286 #define BITMAP_TRANSFORM_D(d) ((24UL<<24)|(((d)&131071UL)<<0))
287 #define BITMAP_TRANSFORM_E(e) ((25UL<<24)|(((e)&131071UL)<<0))
288 #define BITMAP_TRANSFORM_F(f) ((26UL<<24)|(((f)&16777215UL)<<0))
289 #define SCISSOR_XY(x,y) ((27UL<<24)|(((x)&511UL)<<9)|(((y)&511UL)<<0))
290 #define SCISSOR_SIZE(width,height) ((28UL<<24)|(((width)&1023UL)<<10)|
      (((height)&1023UL)<<0))
291 #define CALL(dest) ((29UL<<24)|(((dest)&65535UL)<<0))
292 #define JUMP(dest) ((30UL<<24)|(((dest)&65535UL)<<0))
293 #define BEGIN(prim) ((31UL<<24)|(((prim)&15UL)<<0))
294 #define COLOR_MASK(r,g,b,a) ((32UL<<24)|(((r)&1UL)<<3)|(((g)&1UL)<<2)|
      (((b)&1UL)<<1)|(((a)&1UL)<<0))
295 #define CLEAR(c,s,t) ((38UL<<24)|(((c)&1UL)<<2)|(((s)&1UL)<<1)|(((t)&1UL)
      <<0))
296 #define END() ((33UL<<24))
297 #define SAVE_CONTEXT() ((34UL<<24))
298 #define RESTORE_CONTEXT() ((35UL<<24))
299 #define RETURN() ((36UL<<24))
300 #define MACRO(m) ((37UL<<24)|(((m)&1UL)<<0))
301 #define DISPLAY() ((0UL<<24))
302
303 #define FT_GPU_NUMCHAR_PERFONT (128)
304 #define FT_GPU_FONT_TABLE_SIZE (148)
305
```

```
306
307
308 /* FT800 font table structure */
309 /* Font table address in ROM can be found by reading the address from
      0xFFFFC location. */
310 /* 16 font tables are present at the address read from location 0xFFFFC */
311 typedef struct FT_Gpu_Fonts
312 {
313     /* All the values are in bytes */
314     /* Width of each character font from 0 to 127 */
315     uint8_t FontWidth[FT_GPU_NUMCHAR_PERFONT];
316     /* Bitmap format of font wrt bitmap formats supported by FT800 - L1,
        L4, L8 */
317     unsigned int    FontBitmapFormat;
318     /* Font line stride in FT800 ROM */
319     unsigned int    FontLineStride;
320     /* Font width in pixels */
321     unsigned int    FontWidthInPixels;
322     /* Font height in pixels */
323     unsigned int    FontHeightInPixels;
324     /* Pointer to font graphics raw data */
325     unsigned int    PointerToFontGraphicsData;
326 } FT_Gpu_Fonts_t;
327
328
329 /* FT800 FUNCTIONS                                                    
    ***************************************************************/
330 void HOST_CMD_ACTIVE(void);         /* send host command activate (wake-up
     command */
331 void HOST_CMD_WRITE(uint8_t CMD);   /* send host command */
332
333 void HOST_MEM_READ_STR(uint32_t addr, uint8_t *pnt, uint8_t len);   /*
    read len bytes of data from memory */
334 void HOST_MEM_WR_STR(uint32_t addr, uint8_t *pnt, uint8_t len);     /*
    write len bytes of data into memory */
335
336 void HOST_MEM_WR8(uint32_t addr, uint8_t data);     /* write  8bit (1byte)
     data to memory */
337 void HOST_MEM_WR16(uint32_t addr, uint32_t data);   /* write 16bit
    (2bytes) data to memory */
338 void HOST_MEM_WR32(uint32_t addr, uint32_t data);   /* write 32bit
    (4bytes) data to memory */
339 uint8_t HOST_MEM_RD8(uint32_t addr);                /* read  8bit  (1byte)
     data from memory */
340 uint32_t HOST_MEM_RD16(uint32_t addr);              /* read  16bit
    (2bytes) data from memory */
341 uint32_t HOST_MEM_RD32(uint32_t addr);              /* read  32bit
    (4bytes) data from memory */
342
343 /*** CO-PROCESSOR                                                     
    ***************************************************************/
344 uint8_t cmd_ready(void);                /* check if co-processor is ready
    */
```

```
345 uint8_t cmd(uint32_t data);             /* command function (tries to
      execute command max. 255 times) */
346 uint8_t cmd_execute(uint32_t data);     /* execute function (returns 0:
      when failed to execute command, ie. co-p. is busy) */
347
348 void cmd_track(int16_t x, int16_t y, int16_t w, int16_t h, int16_t tag);
                                            /* set touch engine for tracking
      */
349 void cmd_spinner(int16_t x, int16_t y, uint16_t style, uint16_t scale);
                                            /* draw spinner */
350 void cmd_slider(int16_t x, int16_t y, int16_t w, int16_t h, uint16_t
      options, uint16_t val, uint16_t range); /* draw slider */
351
352 void cmd_text(int16_t x, int16_t y, int16_t font, uint16_t options, const
      char* str);                           /* draw text */
353 void cmd_button(int16_t x, int16_t y, int16_t w, int16_t h, int16_t font,
      uint16_t options, const char* str); /* draw button */
354 void cmd_keys(int16_t x, int16_t y, int16_t w, int16_t h, int16_t font,
      uint16_t options, const char* str);     /* draw keyboard */
355
356 void cmd_memzero(uint32_t ptr, uint32_t num);   /* write zero to a block
      of memory */
357
358 void cmd_fgcolor(uint32_t c);           /* set widget foreground color */
359 void cmd_bgcolor(uint32_t c);           /* set widget background color */
360 void cmd_gradcolor(uint32_t c);         /* set 3d button highlight color
      */
361 void cmd_gradient(int16_t x0, int16_t y0, uint32_t rgb0, int16_t x1,
      int16_t y1, uint32_t rgb1); /* draw gradient */
362
363 void cmd_loadidentity(void);                /* set current matrix to the
      identity matrix */
364 void cmd_setmatrix(void);                   /* write current matrix to the
       display list */
365 void cmd_rotate(int32_t angle);             /* apply rotation to the
      current matrix */
366 void cmd_translate(int32_t tx, int32_t ty); /* apply translation to the
      current matrix */
367
368 #endif
369
```