



# adStar-L

*- SDK Reference Manual -*

**32bits EISC Microprocessor *adStar-L-L***

**Ver 1.0**  
**June 23, 2015**

**Advanced Digital Chips Inc.**

**History**

2015-06-23

Created Preliminary Specification

---

## **adStar-L SDK Reference Manual**

---

***©Advanced Digital Chips Inc.***

All right reserved.

No part of this document may be reproduced in any form without written permission from Advanced Digital Chips Inc.

Advanced Digital Chips Inc. reserves the right to change in its products or product specification to improve function or design at any time, without notice.

### ***Office***

22F, Bldg A, KeumkangPenterium IT Tower,  
810, Gwanyang\_dong, Dongan-Gu, Anyang-si, Gyeonggi-do, 431-060, Korea  
Tel : +82-31-463-7500  
Fax : +82-31-463-7588  
URL : <http://www.adc.co.kr>

## — Table of Contents —

<b>1. SOFTWARE 개발환경 .....</b>	<b>13</b>
<b>2. ADSTAR-L SDK 활용 .....</b>	<b>19</b>
2.1 adStar-L SDK 구성 및 기본 정의.....	19
2.2 adStar-L SDK library build.....	21
2.3 Demo Program 실행.....	22
2.3.1 Project build & download.....	23
2.3.2 Nand Flash(SD Card)에 파일 복사하기 (Mass Storage Mode 사용하기).....	25
2.3.3 Demo Program build & download.....	25
2.4 adStar-L 세프로젝트만들기.....	26
<b>3. BOOTLOADER .....</b>	<b>31</b>
3.1 Bootloader.....	31
3.1.1 Bootloader 사용하기.....	32
3.1.2 Bootloader Mode.....	33
가. Remote Communication Mode.....	33
나. Mass Storage Mode.....	36
다. Execute Mode.....	36
라. User Define Mode (execute_fat).....	38
3.2 Nand Boot Code .....	39
3.2.1 Nand Boot Code 사용하기.....	39
3.3 BootLoader 없이 사용하기.....	41
<b>4. LIB_CONFIG.H .....</b>	<b>43</b>
<b>5. UART .....</b>	<b>46</b>
5.1 uart_config.....	46
5.2 uart_putch.....	47
5.3 uart_putdata.....	47
5.4 uart_putstr.....	47
5.5 uart_getch.....	48
5.6 uart_getdata.....	48
5.7 uart_rx_flush.....	48
5.8 uart_tx_flush.....	49
5.9 set_debug_channel.....	49
5.10 get_debug_channel.....	49
5.11 debugprintf.....	50
5.12 debugstring.....	50
5.13 PRINTLINE.....	50
5.14 PRINTVAR( A ) .....	51
5.15 UART Example .....	51

< UART Interrupt >.....	52
<b>6. INTERRUPT .....</b>	<b>53</b>
6.1 <i>init_interrupt</i> .....	53
6.2 <i>set_interrupt</i> .....	53
6.3 <i>enable_interrupt</i> .....	53
6.4 <i>Interrupt Example</i> .....	55
<b>7. TIMER .....</b>	<b>56</b>
7.1 <i>set_timer</i> .....	56
7.2 <i>stop_timer</i> .....	56
7.3 <i>delayms</i> .....	56
7.4 <i>TIMER Example</i> .....	57
<b>8. GRAPHIC.....</b>	<b>58</b>
8.1 <i>setscreen</i> .....	58
8.2 <i>createframe</i> .....	58
8.3 <i>setframebuffer</i> .....	59
8.4 <i>setdoubleframebuffer</i> .....	60
8.5 <i>setframebufferxy</i> .....	60
8.6 <i>set_draw_target</i> .....	61
8.7 <i>get_draw_target</i> .....	61
8.8 <i>getbackframe</i> .....	61
8.9 <i>getfrontframe</i> .....	62
8.10 <i>flip</i> .....	62
8.11 <i>getscreenwidth</i> .....	62
8.12 <i>getscreenheight</i> .....	63
8.13 <i>getscreenpitch</i> .....	63
8.14 <i>getscreenbpp</i> .....	63
8.15 <i>drawputpixel</i> .....	64
8.16 <i>draw_line</i> .....	64
8.17 <i>draw_rect</i> .....	64
8.18 <i>draw_rectfill</i> .....	65
8.19 <i>draw_roundrect</i> .....	65
8.20 <i>draw_roundrectfill</i> .....	66
8.21 <i>draw_circle</i> .....	66
8.22 <i>draw_circleshell</i> .....	67
8.23 <i>draw_ellipse</i> .....	67
8.24 <i>draw_ellipselfill</i> .....	67
8.25 <i>loadbmp</i> .....	68
8.26 <i>loadbmpp</i> .....	68
8.27 <i>loadjpg</i> .....	68

8.28 loadjpgp.....	69
8.29 loadtga.....	69
8.30 loadtgap.....	70
8.31 loadpng.....	70
8.32 loadpngp.....	70
8.33 loadsurf.....	71
8.34 loadjpg_hw.....	71
8.35 draw_surface.....	71
8.36 draw_surface_rect.....	72
8.37 draw_set_clip_winodw.....	72
8.38 draw_surface_scale.....	73
8.39 draw_surface_scalerect.....	73
8.40 release_surface.....	74
8.41 savebmp.....	74
8.42 savejpg.....	75
8.43 createsurface_from.....	75
8.44 single frame & double frame 사용 예.....	77
8.45 set frame buffer xy 사용 예.....	80
8.46 Graphic Example.....	81
<b>9. MOVIE FILE PLAY.....</b>	<b>83</b>
9.1 loadmpvie.....	83
9.2 release_movie.....	83
9.3 movie_play.....	84
9.4 movie_drawnext.....	84
9.5 movie_drawnextex.....	85
9.6 movie_seek.....	85
9.7 movie_current_time.....	85
9.8 Example.....	86
<b>10. SOUND .....</b>	<b>87</b>
10.1 sound_init().....	87
10.2 sound_loadwav.....	87
10.3 sound_loadwavp.....	87
10.4 sound_loadmp3.....	88
10.5 sound_loadmp3p.....	88
10.6 sound_release.....	88
10.7 sound_play.....	89
10.8 sound_stop.....	89
10.9 sound_vol.....	89
10.10 sound_vol_wav.....	90
10.11 sound_pause.....	90

10.12 sound_resume.....	90
10.13 sound_isplay.....	91
10.14 sound_ispause.....	91
10.15 Sound Example.....	92
<b>11. FILE SYSTEM.....</b>	<b>94</b>
11.1 f_mount.....	94
11.2 f_chdrive.....	94
11.3 f_chdir.....	95
11.4 FILE System Example.....	95
<b>12. FONT .....</b>	<b>96</b>
12.1 create_bmpfont.....	96
12.2 release_bmpfont.....	96
12.3 bmpfont_draw.....	97
12.4 bmpfont_draw_vleft.....	97
12.5 bmpfont_draw_vright.....	97
12.6 egl_font_set_color.....	98
12.7 bmpfont_makesurface.....	98
12.8 bmpfont_setkerning.....	98
12.9 bmpfont_setautokerning.....	99
12.10 create_bitfont.....	99
12.11 release_bitfont.....	99
12.12 bitfont_draw.....	100
12.13 bitfont_draw_vleft.....	100
12.14 bitfont_draw_vright.....	100
12.15 bitfont_makesurface.....	101
12.16 FONT Example.....	101
< font image 제작 방법 >.....	104
<b>13. SPI.....</b>	<b>107</b>
13.1 spi_master_init.....	107
13.2 spi_set_freq.....	107
13.3 spi_master_xfer.....	107
13.4 spi_wait_empty_fifo.....	108
13.5 SPI Example.....	108
<b>14. EGL LIBRARY .....</b>	<b>117</b>
WINDOW .....	120
►egl_create_window function.....	121
►egl_window_show function.....	122
►egl_window_isshow function.....	123

▶ <i>egl_window_invalidate function</i>	124
▶ <i>egl_window_invalidate_rect function</i>	125
▶ <i>egl_window_redraw_rect function</i>	126
▶ <i>egl_window_set_bg function</i>	127
▶ <i>egl_window_get_bg function</i>	128
▶ <i>egl_window_add_object function</i>	129
▶ <i>egl_window_set_callback function</i>	130
▶ <i>egl_window_get_active function</i>	131
▶ <i>egl_user_touch_input function</i>	132
▶ <i>egl_draw function</i>	133
▶ <i>egl_visible_object function</i>	134
▶ <i>egl_window_delete_object function</i>	135
▶ <i>egl_release_window function</i>	136
▶ <i>Window Example</i>	137
BUTTON.....	139
▶ <i>egl_create_button function</i>	140
▶ <i>egl_button_callback function</i>	141
▶ <i>egl_release_button function</i>	142
▶ <i>Button Example</i>	143
IMAGE BUTTON.....	144
▶ <i>egl_create_image_button function</i>	145
▶ <i>egl_image_button_callback function</i>	146
▶ <i>egl_release_image_button function</i>	147
▶ <i>Image Button Example</i>	148
TOGGLE IMAGE BUTTON.....	149
▶ <i>egl_create_toggle_image function</i>	150
▶ <i>egl_toggle_image_callback function</i>	151
▶ <i>egl_toggle_image_set_on function</i>	152
▶ <i>egl_release_toggle_image function</i>	153
▶ <i>Toggle Image Button Example</i>	154
LABEL .....	155
▶ <i>egl_create_label function</i>	156
▶ <i>egl_label_callback function</i>	157
▶ <i>egl_label_set_text function</i>	158
▶ <i>egl_label_set_redraw_bg function</i>	159
▶ <i>egl_label_set_color function</i>	160
▶ <i>egl_release_label function</i>	161
▶ <i>Label Example</i>	162
CHECK BUTTON / RADIO BUTTON .....	163
▶ <i>egl_create_checkbutton function</i>	164
▶ <i>egl_checkbutton_callback function</i>	165

▶ <i>egl_checkbutton_set_check function</i> .....	166
▶ <i>egl_checkbutton_get_check function</i> .....	167
▶ <i>egl_checkbutton_set_style function</i> .....	167
▶ <i>egl_release_checkbutton function</i> .....	168
▶ <i>Check / Radio button Example</i> .....	170
PROGRESS BAR .....	172
▶ <i>egl_create_progressbar function</i> .....	173
▶ <i>egl_progressbar_set_barcolor function</i> .....	174
▶ <i>egl_progressbar_set_bgcolor function</i> .....	175
▶ <i>egl_progressbar_set_textcolor function</i> .....	176
▶ <i>egl_progressbar_set_text function</i> .....	177
▶ <i>egl_progressbar_set_pos function</i> .....	178
▶ <i>egl_progressbar_get_pos function</i> .....	179
▶ <i>egl_release_progressbar function</i> .....	180
▶ <i>Progress Bar Example</i> .....	181
SCROLL BAR.....	182
▶ <i>egl_create_scrollbar function</i> .....	184
▶ <i>egl_scrollbar_callback function</i> .....	185
▶ <i>egl_scroll_set_position function</i> .....	186
▶ <i>egl_scroll_get_position function</i> .....	187
▶ <i>egl_scroll_set_totalcount function</i> .....	188
▶ <i>egl_scroll_get_totalcount function</i> .....	189
▶ <i>egl_scroll_set_viewcount function</i> .....	190
▶ <i>egl_scroll_get_viewcount function</i> .....	191
▶ <i>egl_scroll_set_upcount function</i> .....	192
▶ <i>egl_scroll_get_upcount function</i> .....	193
▶ <i>egl_scroll_set_bgcolor function</i> .....	194
▶ <i>egl_scroll_set_size function</i> .....	195
▶ <i>egl_release_scrollbar function</i> .....	196
▶ <i>Scroll Bar Example</i> .....	197
SLIDER .....	198
▶ <i>egl_create_slider function</i> .....	200
▶ <i>egl_slider_callback function</i> .....	202
▶ <i>egl_slider_set_pos function</i> .....	203
▶ <i>egl_slider_get_pos function</i> .....	204
▶ <i>egl_slider_set_range function</i> .....	205
▶ <i>egl_slider_get_range function</i> .....	206
▶ <i>egl_slider_stepit function</i> .....	207
▶ <i>egl_slider_set_tick_frequency function</i> .....	208
▶ <i>egl_slider_set_tick_style function</i> .....	209
▶ <i>egl_slider_set_thumb_size function</i> .....	210

▶ <i>egl_slider_get_thumb_size function</i> .....	211
▶ <i>egl_slider_set_barcolor function</i> .....	212
▶ <i>egl_slider_set_tickcolor function</i> .....	213
▶ <i>egl_slider_set_transparent function</i> .....	214
▶ <i>egl_release_slider function</i> .....	215
▶ <i>Slider Example</i> .....	216
<b>LIST BOX .....</b>	<b>217</b>
▶ <i>egl_create_listbox function</i> .....	219
▶ <i>egl_listbox_callback function</i> .....	220
▶ <i>egl_listbox_additem function</i> .....	220
▶ <i>egl_listbox_delitem function</i> .....	222
▶ <i>egl_listbox_delitem_text function</i> .....	223
▶ <i>egl_listbox_delitem_num function</i> .....	224
▶ <i>egl_listbox_alldelitem function</i> .....	225
▶ <i>egl_listbox_get_all_itemlist function</i> .....	226
▶ <i>egl_listbox_get_sel_item function</i> .....	227
▶ <i>egl_listbox_get_multiple_sel_itemlist function</i> .....	228
▶ <i>egl_listbox_set_bgcolor function</i> .....	229
▶ <i>egl_listbox_set_selbgcolor function</i> .....	230
▶ <i>egl_listbox_set_textcolor function</i> .....	231
▶ <i>egl_listbox_set_seltxtcolor function</i> .....	232
▶ <i>egl_listbox_set_textalign function</i> .....	233
▶ <i>egl_listbox_set_scrollwidth function</i> .....	234
▶ <i>egl_listbox_change_item_text function</i> .....	235
▶ <i>egl_listbox_change_item_num function</i> .....	236
▶ <i>egl_release_listbox function</i> .....	237
▶ <i>List box Example</i> .....	238
<b>CIRCLE GAUGE.....</b>	<b>241</b>
▶ <i>egl_create_circle_gauge function</i> .....	242
▶ <i>egl_circle_gauge_set_value function</i> .....	244
▶ <i>egl_circle_gauge_get_value function</i> .....	245
▶ <i>egl_release_circle_gauge function</i> .....	246
▶ <i>Circle Gauge Example</i> .....	247
<b>BAR GAUGE.....</b>	<b>249</b>
▶ <i>egl_create_bar_gauge function</i> .....	250
▶ <i>egl_bar_gauge_set_value function</i> .....	252
▶ <i>egl_bar_gauge_get_value function</i> .....	253
▶ <i>egl_release_bar_gauge function</i> .....	254
▶ <i>Bar Gauge Example</i> .....	255
<b>PICTURE.....</b>	<b>257</b>
▶ <i>egl_create_picture function</i> .....	258

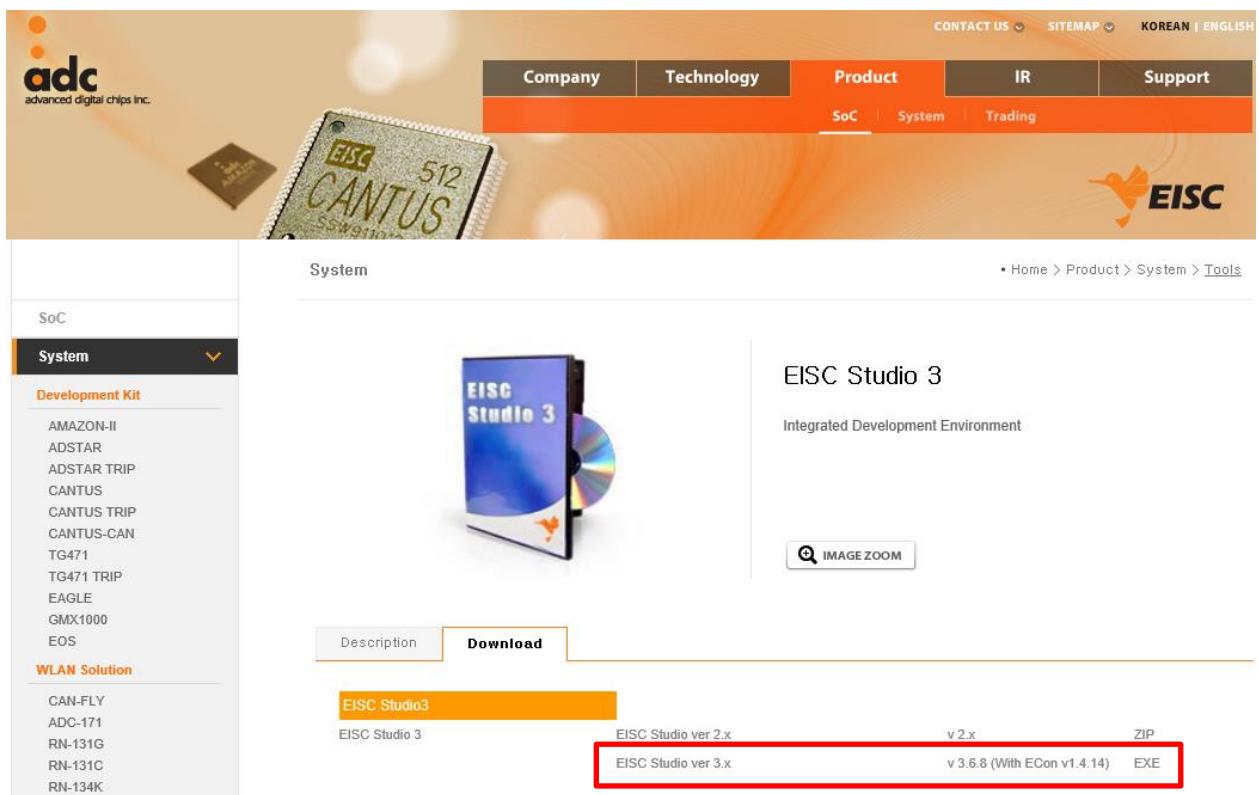
▶ <i>egl_picture_callback function</i> .....	259
▶ <i>egl_picture_set function</i> .....	260
▶ <i>egl_release_picture function</i> .....	261
▶ <i>Picture Example</i> .....	262
ANIMATION.....	263
▶ <i>egl_create_animation function</i> .....	264
▶ <i>egl_release_animation function</i> .....	266
▶ <i>Animation Example</i> .....	267
CUSTOM OBJECT.....	268
▶ <i>egl_create_custom_object function</i> .....	269
▶ <i>Custom Example</i> .....	271
MESSAGEBOX.....	273
▶ <i>egl_show_messagebox function</i> .....	274
▶ <i>MessageBox Example</i> .....	276
EGL FONT .....	278
▶ <i>egl_get_font function</i> .....	279
▶ <i>egl_set_font function</i> .....	280
▶ <i>egl_font_set_bkmode function</i> .....	281
▶ <i>egl_font_get_bk_color function</i> .....	282
▶ <i>egl_font_set_bk_color function</i> .....	283
▶ <i>egl_font_get_color function</i> .....	284
▶ <i>egl_font_set_color function</i> .....	285
▶ <i>create_bitfont function</i> .....	286
▶ <i>release_bitfont function</i> .....	287
▶ <i>create_bmpfont function</i> .....	288
▶ <i>bmfont_release function</i> .....	289
▶ <i>draw_text function</i> .....	290
▶ <i>draw_text_pivot function</i> .....	291
▶ <i>draw_text_len function</i> .....	292
▶ <i>draw_text_in_box function</i> .....	293
▶ <i>text_width function</i> .....	295
EGL PRIMITIVES .....	296
▶ <i>draw_line function</i> .....	298
▶ <i>draw_hline function</i> .....	299
▶ <i>draw_vline function</i> .....	300
▶ <i>draw_thickline function</i> .....	301
▶ <i>draw_rect function</i> .....	302
▶ <i>draw_rectfill function</i> .....	303
▶ <i>draw_rectfill_gradient function</i> .....	304
▶ <i>draw_rectfill_h_gradient function</i> .....	305
▶ <i>draw_rectfill_v_gradient function</i> .....	306

▶ <i>draw_roundrect function</i> .....	307
▶ <i>draw_roundrectfill function</i> .....	308
▶ <i>draw_arc function</i> .....	309
▶ <i>draw_pie function</i> .....	310
▶ <i>draw_piefill function</i> .....	311
▶ <i>draw_ellipse function</i> .....	312
▶ <i>draw_ellipselfill function</i> .....	313
▶ <i>draw_circle function</i> .....	314
▶ <i>draw_circlefill function</i> .....	315
▶ <i>draw_bezier function</i> .....	316
▶ <i>draw_polyline function</i> .....	318
▶ <i>draw_polygon function</i> .....	319
▶ <i>draw_polygonfill function</i> .....	320
EGL ETC. ....	321
▶ <i>egl_init function</i> .....	322
▶ <i>egl_show_object function</i> .....	323
▶ <i>egl_object_set_redraw function</i> .....	324
<b>15. DEBUGGING</b> .....	<b>325</b>
<i>15.1 Debugging 헤더</i> .....	329
<i>15.2 Debugging</i> .....	329

# 1. Software 개발환경

첫 번째 장으로 adStar-L 을 사용하기 위한 개발환경을 구축하는 방법부터 알아보도록 하겠다. 에이디칩스에서는 adStar-L Chip 을 위해 프로그램 코드 편집, compile, download, debugging 을 하나의 프로그램에서 수행할 수 있는 통합개발환경(IDE) EISC Studio3 를 제공하기 때문에 자료실에서 EISC Studio3 설치 파일을 내려 받아 설치만 해주면 간단하게 개발환경 구축을 마칠 수 있다. 참고로 EISC Studio3 는 Windows OS 만 지원하면 XP 이상에서 동작한다.

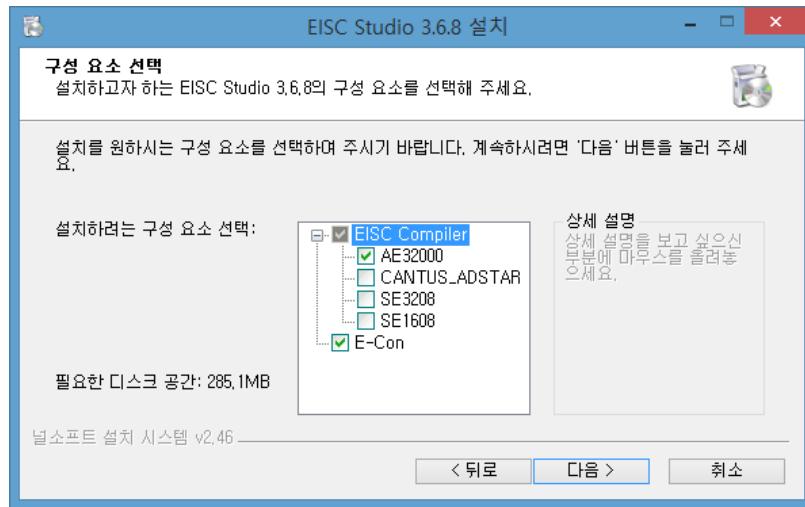
1. 에이디칩스 홈페이지(<http://www.adc.co.kr>)에 접속해서, Product → System → EISC Studio3 → Download 에서 EISC Studio ver 3.x 의 v3.6.8 을 내려 받는다.  
(설치 및 사용자 매뉴얼도 있으니 참고 하도록 한다.)



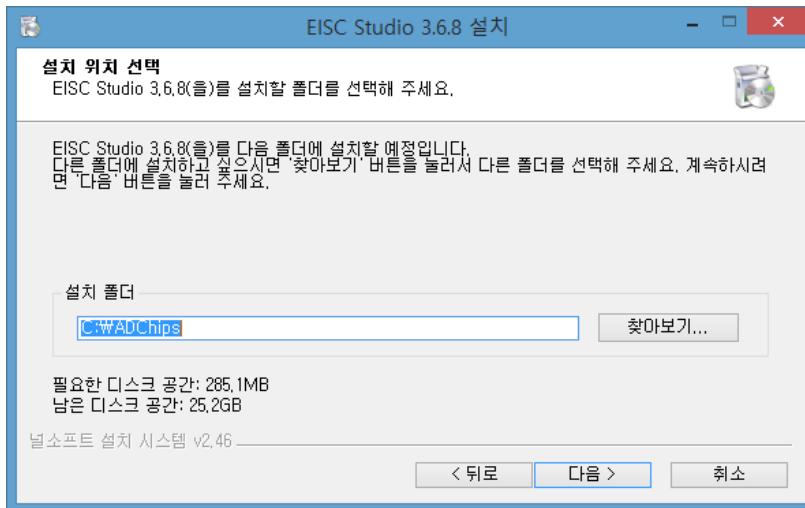
2. 내려 받은 ES3\_setup\_v3.6.8.exe 를 실행하여 설치를 시작하면, 설치를 시작한다는 창이 뜨는데 '다음'을 클릭하여 설치를 시작한다. '다음'을 클릭하면 아래와 같이 설치할 Compiler 와 개발장비인 E-CON 의 Driver 설치를 선택하는 창이 뜬다. adStar-L 은 AE32000 processor<sup>1</sup>를 내장한 Chip 으로 AE32000 을 선택하면 된다.  
그리고 E-CON<sup>2</sup>장비를 처음 사용하는 사용자라면 E-CON 을 선택하여 E-Con Driver 도 설치한다.

<sup>1</sup>에이디칩스에서 개발한 32bit Processor. (EISC Architecture)

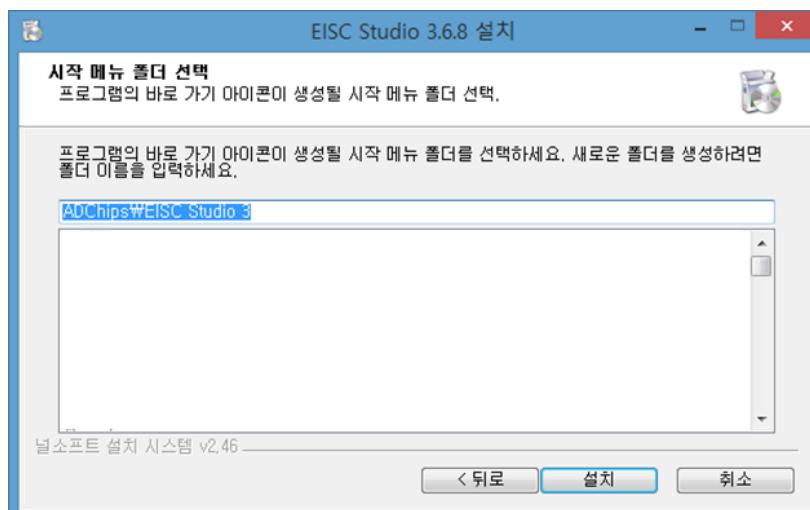
<sup>2</sup>Adstar-L 과 연결하여 program download 및 debugging 을 수행 할 때 필요 한 장비.

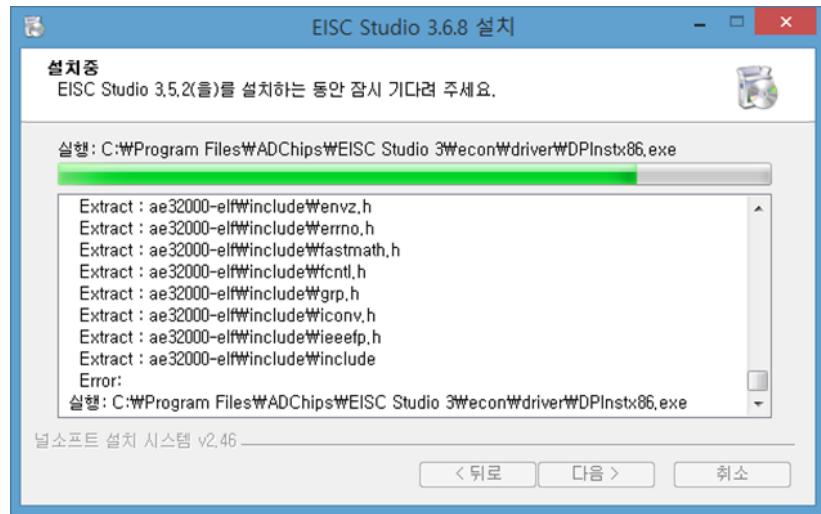


3. compiler 와 E-CON Driver 선택을 하고 다음을 클릭하면, 설치할 폴더를 선택하는 창이 나온다.  
 특별한 경우가 아니라면 default 로 되어있는 경로에 설치하면 된다.  
 (주의사항으로는 경로상에 괄호 ( ) 가 들어가면 안 된다.)

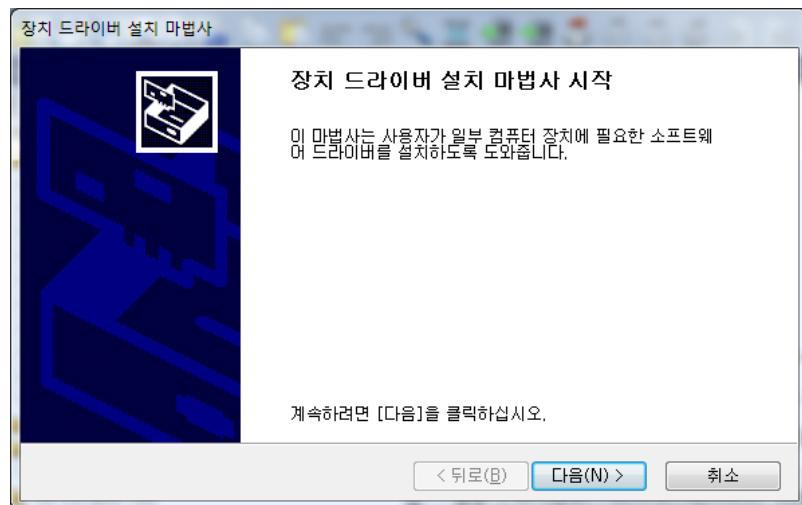


4. 다음을 클릭하면 아래와 같은 시작 메뉴 폴더를 선택하는 창이 뜨고, 설치를 누르면 설치가 시작된다.

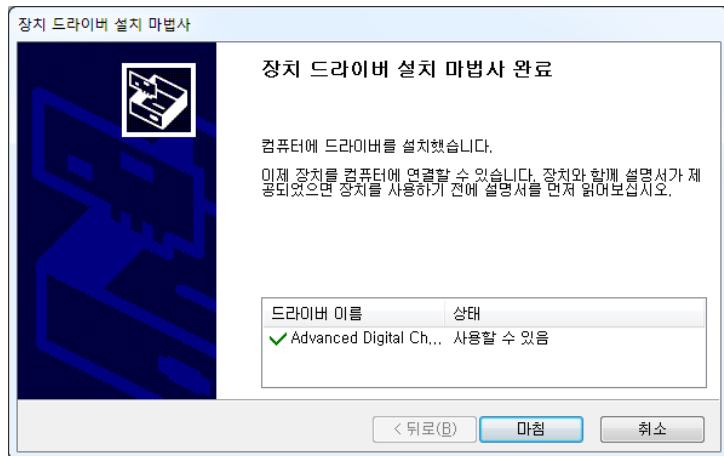




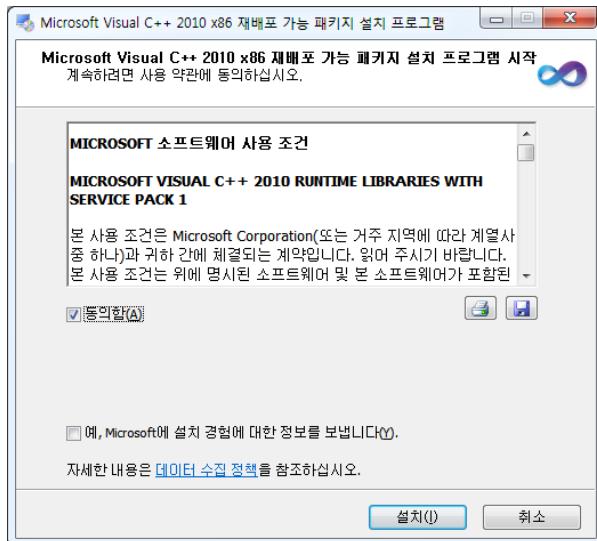
- 구성요소 선택에서 E-CON 을 체크하였으면 설치가 끝날 무렵 E-Con Driver 설치를 위한 다음과 같은 창이 뜨고, 다음을 클릭하면 Driver 설치를 시작한다. 드라이버 설치 중 Windows 보안 경고 창이 뜨면 "이 드라이버 소프트웨어를 설치합니다."를 선택하도록 한다.  
(주의사항으로는 컴퓨터와 E-CON 이 연결되어 있을 경우 설치에 문제가 생길 수도 있으므로, 컴퓨터와 E-CON 이 연결되지 않은 상태에서 설치하기 바란다.)



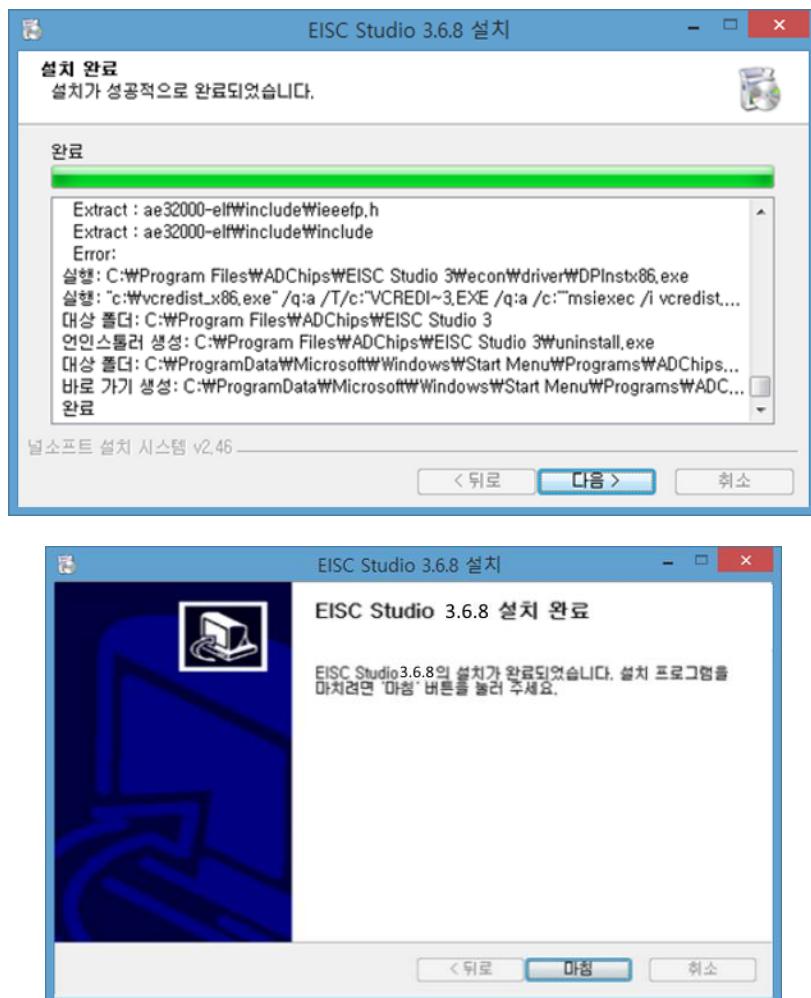
6. 장치 드라이버 설치가 완료되면 다음과 같은 창이 나오고, 마침을 눌러 EISC Studio3 의 남은 설치과정을 계속 진행한다.



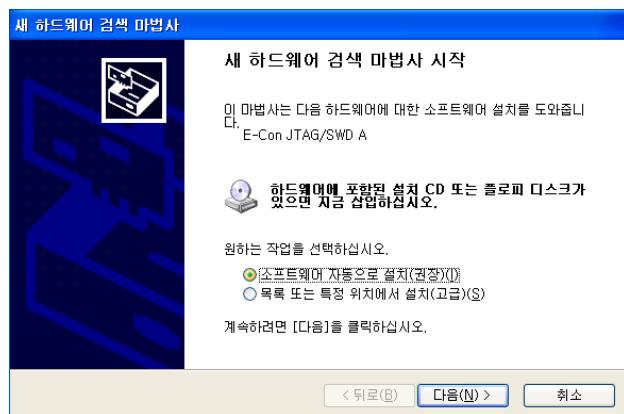
7. 다음과 같이 Microsoft Visual C++ 2010 재배포 가능 패키지를 설치하라는 창이 뜨면 동의함에 체크한 후 설치하면 된다. EISC Studio3 를 실행하는데 필요하므로 설치하고 넘어가도록 한다.



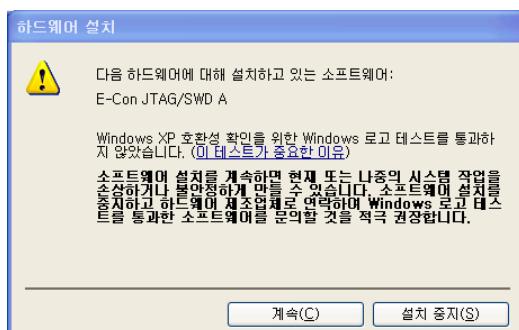
8. 설치가 완료되면 다음과 같은 창이 출력되고, 다음을 누르면 설치가 완료된다.



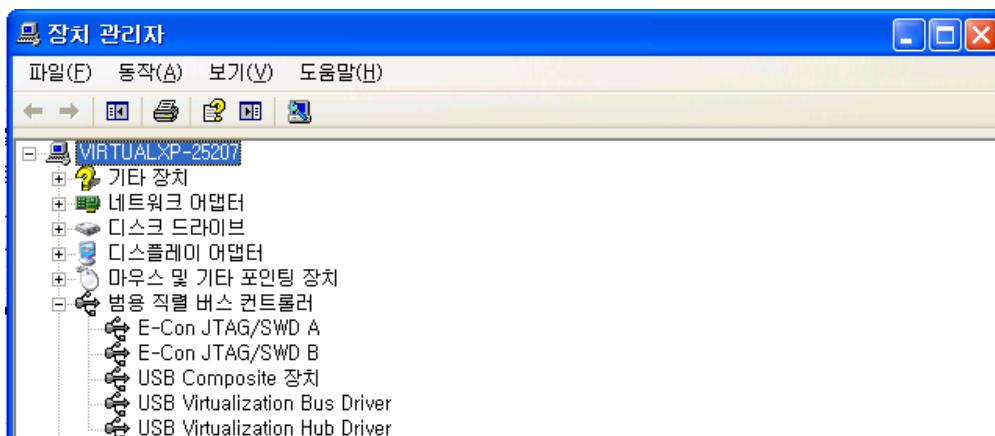
9. EISC Studio 3 설치가 완료 되면 다음으로 Download 장비인 E-Con driver 를 설치하여야 한다.  
EISC Studio 3 를 설치하면서 E-Con driver 파일을 복사하였다면 E-Con 을 컴퓨터와 연결하여  
driver 설치를 진행하면 되고, E-Con driver 파일을 복사 하지 않았을 경우에는 EISC Studio  
3 를 설치한 폴더에 econ\driver 라는 폴더 안에 DPIInstx86.exe(32bit) 또는 DPIInstx64(64bit)  
를 실행하여 드라이버 파일을 복사 하고, 설치를 진행하면 된다.  
**(DPIInstx86.exe 또는 DPIInstx64.exe 를 실행해서 설치 할 경우에도, 컴퓨터와 E-CON 을  
연결하지 않은 상태에서 진행하도록 한다.)**
10. E-Con driver 파일 복사가 되었다면, 컴퓨터와 E-Con 을 연결하였을 때 Windows7 의 경우에는  
자동으로 드라이버를 찾아 설치를 진행하게 된다. Windows XP 이하 버전에서는 다음과 같이  
새 하드웨어 검색 마법사 창이 뜨게 되는데, "소프트웨어 자동으로 설치"를 체크하고 확인을  
클릭하면 driver 설치가 진행된다.



설치 중 다음과 같은 경고창이 뜨게 되는데, 계속을 클릭하면 된다.



- E-Con은 A,B 두 개의 채널로 되어 있어 위와 같은 방식으로 B 채널도 설치해주면 된다. 설치가 모두 완료되면 장치관리자에서 다음과 같이 드라이버 설치가 완료 된 것을 확인할 수 있다. E-Con driver 설치와 관련된 좀 더 자세한 내용은 adchips 홈페이지([www.adc.co.kr](http://www.adc.co.kr)) Product→System→E-CON→Download 의 "E-CON Driver Install Guide"를 참고하기 바란다.

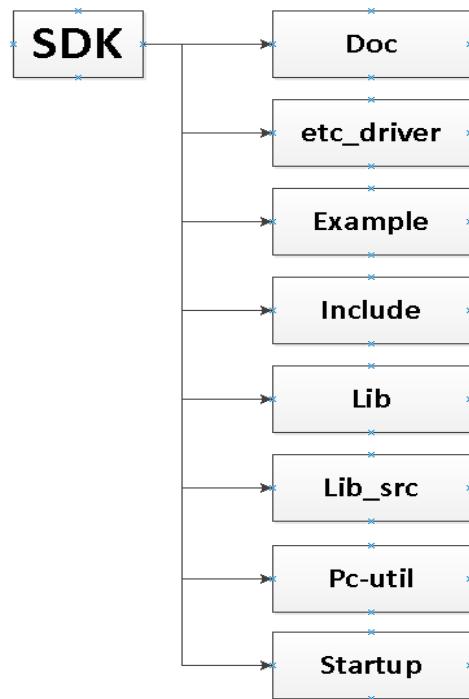


## 2. ADSTAR-L SDK 활용

개발에 편의를 위해 adchips에서는 adStar-L SDK를 제공하고 있다. adStar-L SDK는 adchips의 홈페이지([www.adc.co.kr](http://www.adc.co.kr))의 Product → SoC → ADSTAR-L → Tools & Software에서 내려 받을 수 있다.

### 2.1 adStar-L SDK 구성 및 기본 정의

adStar-L SDK는 다음과 같이 구성되어 있다.



Doc → 현재 보고 있는 매뉴얼을 포함, adStar-L 와 관련된 문서가 들어있는 폴더.

Example → adStar-L STK Board에서 동작하는 예제 프로그램 폴더.

( Example 폴더의 flash\_data에는 SDK 예제 프로그램이 동작하는데 있어 필요한 image, sound 파일이 들어있다. )

Include → adStar-L SDK의 header file 폴더.

lib → adStar-L SDK의 library file 폴더.

lib\_src → adStar-L SDK의 library source file 폴더.

Pc-util → adStar-L 을 사용하는데 있어 필요한 Utility 폴더. adStar-L USB Driver 가 들어있다.

Startup → adStar-L 개발에 필요한 startup 코드와 link script 코드가 들어있는 폴더. 이 외에 STK board 기본 설정 코드도 들어있다.

adStar-L SDK에는 사용하기 편하고 구분이 쉽게 변수형을 아래와 같이 정의해놓았다.  
SDK Source를 참조하는데 아래 표를 참고하기 바란다.

Signed char	S8, s8
Signed short	S16, s16
Signed int	S32, s32
Unsigned char	U8, u8, __u8, BYTE, uchar
Unsigned short	U16, u16, __u16, WORD, ushort
Unsigned int	U32, u32, __u32
Unsigned long	DWORD, ulong
Unsigned long long	U64, u64, __u64
Volatile unsigned char	vU8
Volatile unsigned short	vU16
Volatile unsigned int	vU32
Volatile unsigned long long	vU64
1	TRUE, true
0	FALSE, false

그리고 Register의 경우 'R\_' 접두어를 붙여 정의해놓았으므로 SDK Source에서 R\_이 붙은 명칭은 Register로 생각하면된다.

예를 들어 R\_TM0CON은 채널 0 번 timer control register이다.

## 2.2 adStar-L SDK library build

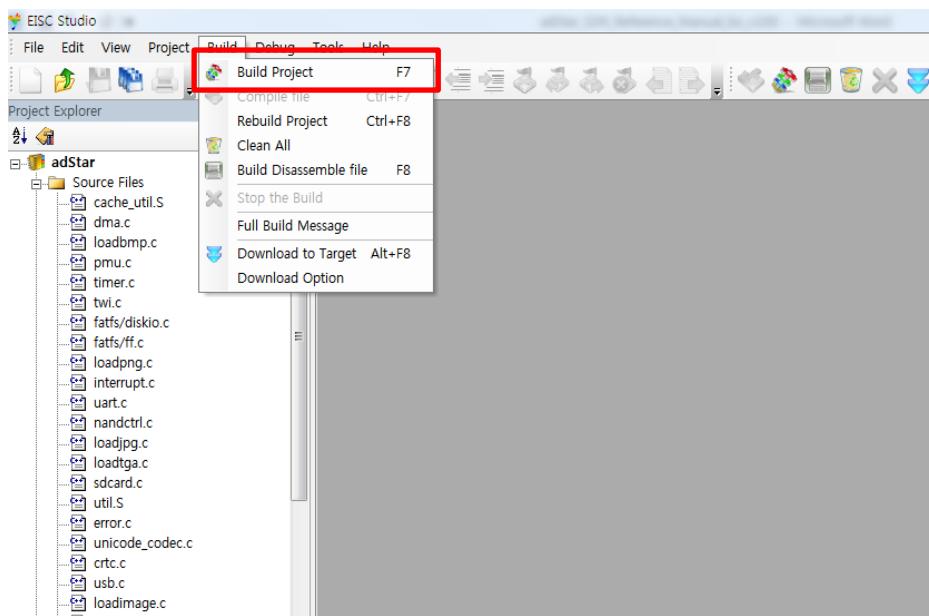
adStar-L SDK 를 홈페이지에서 내려 받았으면, 제일 먼저 adStar-L SDK Library 를 build 해주어야 한다. adStar-L SDK Library source 는 lib\_src 폴더에 있으므로, lib\_src 폴더의 adStar-L.epx 파일을 열어서 build menu 의 build project 를 실행하거나, F7 을 눌러서 project build 를 진행하면 된다. Build 가정상적으로 완료 되면, lib 폴더안에 libadStar-L.a 파일이 생성 된다.  
(adStar-L SDK 의 모든 예제들이 library 를 참조하므로 제일 먼저 해주어야 한다.)

이름	수정한 날짜	유형	크기
doc	2015-06-18 오전...	파일 폴더	
etc_driver	2015-06-18 오전...	파일 폴더	
example	2015-06-18 오후...	파일 폴더	
include	2015-06-18 오전...	파일 폴더	
lib	2015-06-18 오전...	파일 폴더	
lib_src	2015-06-18 오전...	파일 폴더	
pc-util	2015-06-14 오후...	파일 폴더	
startup	2015-06-18 오전...	파일 폴더	
changelog.txt	2015-06-18 오후...	텍스트 문서	0KB

[ SDK library ]

이름	수정한 날짜	유형	크기
egl	2015-06-14 오후...	파일 폴더	
fatfs	2015-06-14 오후...	파일 폴더	
obj	2015-06-18 오후...	파일 폴더	
output	2015-06-14 오후...	파일 폴더	
usb_host	2015-06-14 오후...	파일 폴더	
adStar-L.epx	2015-06-18 오후...	EIS STUDIO 3 Pro...	10KB
cache_util.S	2015-06-01 오전...	S 파일	4KB
cache_util_2.S	2015-05-29 오후...	S 파일	4KB
crtc.c	2015-05-29 오후...	C 파일	5KB
dma.c	2015-04-07 오후...	C 파일	6KB
error.c	2015-02-12 오후...	C 파일	2KB
interrupt.c	2015-03-23 오전...	C 파일	18KB
Makefile.mk	2015-06-18 오후...	MK 파일	26KB
mp3play.c	2015-02-26 오후...	C 파일	25KB
nandctrl.c	2015-06-10 오전...	C 파일	46KB
pmu.c	2015-02-28 오후...	C 파일	2KB
rtc.c	2015-02-16 오후...	C 파일	4KB
sdcard.c	2015-04-07 오후...	C 파일	14KB
serialflash.c	2015-02-12 오후...	C 파일	2KB
soundmixer.c	2015-04-07 오후...	C 파일	27KB

[ lib\_src 폴더의 adStar-L.epx 파일 ]



[ Build Project ]



[ Build Project 정상 완료 ]

이름	수정한 날짜	유형	크기
libadStar-L.a		A 파일	2,347KB
libjpeg.a		A 파일	1,965KB
libmad.a		A 파일	335KB
libpng.a		A 파일	1,232KB
libz.a		A 파일	410KB

[ libadStar-L.a 파일 생성 ]

만약에 adStar-L library 를 수정하게 되면 위처럼 다시 library 를 build 해주어야 한다.

그리고 adStar-L library 를 사용하는 program 도 다시 build 해주어야 수정 된 library 가 적용된다.

### 2.3 Demo Program 실행

adStar-L SDK example 폴더에는 adStar-L STK Board 에서 동작하는 예제들이 준비 되어 있어서, STK Board 에서 adStar-L 동작을 쉽게 확인 해 볼 수 있다.

이번 장에서는 SDK 의 example 중 Demo Project Source 를 build, download, 실행하는 방법에 대해 알아보도록 하겠다.

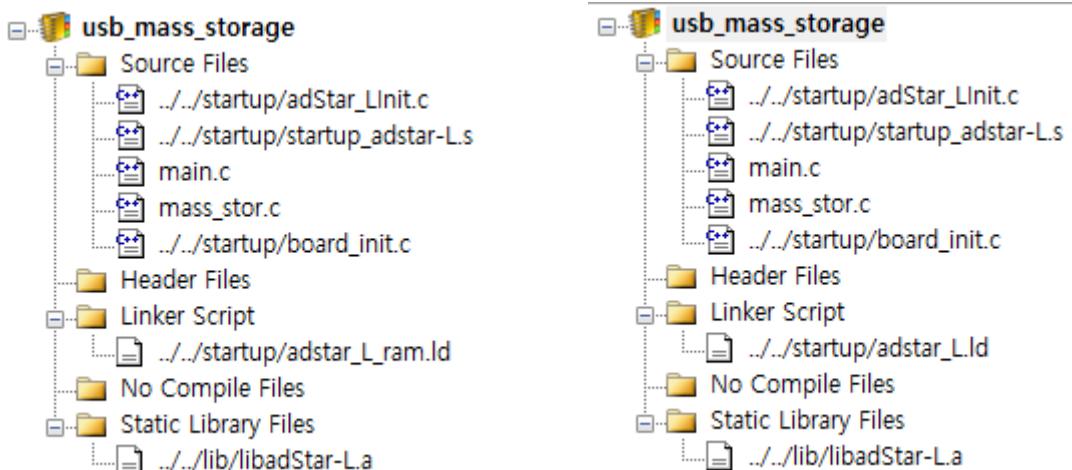
우선 Demo 예제를 동작 시키기 위해서는 Nand Flash에 출력될 이미지 및 사운드 파일이 있어야 한다. Nand Flash에 파일을 복사하기 위해서는 Example 폴더의 `usb_mass_storage` project를 실행하거나 bootloader의 mass storage mode를 사용하면 되는데, bootloader는 다음 장에서 설명하도록 하고, 이번 장에서는 `usb_mass_storage` project를 사용하여 진행하는 방법에 대해 설명하도록 하겠다. `usb_mass_storage`를 실행하기 위해 우선 project를 build하고, download하는 방법에 대해서 알아보자.

### 2.3.1 Project build & download

example 폴더의 `usb_mass_storage` 폴더를 열어 `usb_mass_storage.epx` 파일을 마우스로 더블클릭하여 project를 open 한다.

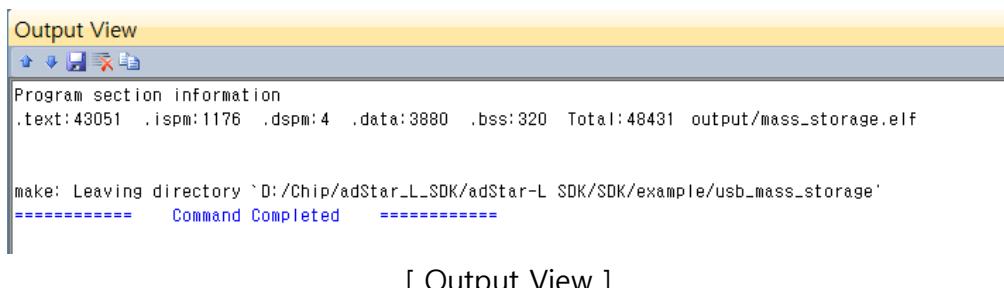
Project를 open한 후 build하여 board에서 실행되는 .bin(binary)파일을 만들기 전에 좌측 Project Explorer 창의 Linker Script에 있는 파일을 변경해 주어야 한다. SDK에서 제공되는 Linker Script 파일은 두 가지가 있는데, 하나는 `adStar-L.ld` 파일이고, 다른 하나는 `adStar-L_ram.ld` 파일이다. SDK에 example들은 기본적으로 `adStar-L_ram.ld` 파일을 사용하도록 되어 있는데, `adStar-L_ram.ld` 파일은 프로그램을 SDRAM에서 동작시킬 때 사용하는 Linker Script 파일로 bootloader가 올라간 상황에서 프로그램을 동작시킬 때 사용할 수 있다. 이번 장에서는 bootloader 없이 프로그램을 동작시키는 방법에 대한 설명으로, `adStar-L_ram.ld` 파일 대신 `adStar-L.ld` 파일을 사용한다.

따라서 Linker Script에서 마우스 오른쪽을 클릭하여 Set Linker Script File을 선택하여, startup 폴더의 `adStar-L.ld` 파일을 선택해 준다.



< `adStar-L_ram.ld` 파일을 `adStar-L.ld` 파일로 변경 >

Linker Script를 변경하였으면, build → Build Project를 실행하거나 F7 키를 눌러서 project를 build 한다. build가 완료되면 Output View에 다음과 같이 출력된다.

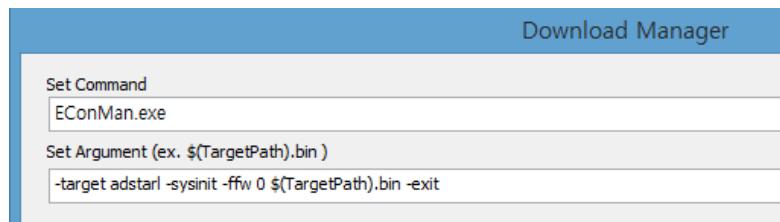


[ Output View ]

Build 가 정상적으로 완료 되면, output 폴더에 usb\_mass\_storage.elf.bin 파일이 생성 된 것을 확인할 수 있다.

다음으로 build 로 생성된 usb\_mass\_storage.elf.bin 파일을 STK board 로 download 하는 방법에 대해 알아보겠다. Download 는 download 할 project 를 EISC Studio3 에 열어 놓은 상태에서 E-CON 을 사용하여 다음의 순서대로 진행하면 된다.

1. Download 를 하기 위해 adStar-L board 와 E-CON 장비를 준비한다.  
(EISC Studio 3 가 설치되어 있어야 하고, E-CON Driver 가 설치되어 있어야 한다. 설치 방법은 1 장의 Software 개발환경을 참고하기 바란다.)
2. adStar-L board 와 E-CON 장비를 연결한 후 board 의 전원을 켠다.
3. Build 메뉴의 Download Option 을 클릭한다. 그러면 다음과 같이 Download Manager 창이 뜨는데, Set Command 에는 EISC Studio3 를 설치한 폴더에서 econ 이라는 폴더 안의 EConMan.exe 라는 E-CON 용 다운로드 프로그램을 선택해준다. (default 값인 EconMan.exe 로 하여도 된다. 그리고 Set Argument 에는 "-target adStarl -sysinit -ffw 0x0 \$(TargetPath).bin -exit" 이라고 입력하고 OK 를 클릭한다.



Argument 에 대해 설명하면,

**-target** 은 말 그대로 download 할 target 을 말하며 adStarl 로 적어주면 된다.

**-sysinit** 는 download 를 위한 초기화를 해주는 command 이다.

**-ffw** 는 build 결과물로 나온 bin 파일을 다운로드 하는 command 로 0x0 은 download 할 주소이고, \$(TargetPath).bin 은 download 할 파일명으로 \$(TargetPath).bin 은 현재 열려있는 project 의 bin 파일을 뜻한다. Bootloader 를 사용하지 않고 프로그램을 실행시킬 때는, linker script 를 adstar-L.ld 로 사용하여 build 하고, 0 번지에 download 하면 된다.

**-exit** 는 download 가 완료되면 자동으로 종료하라는 command 이다.

(Set Argument에 대한 자세한 내용은 [www.adc.co.kr](http://www.adc.co.kr)→ Product → System → E-CON → Download → E-CON.pdf 파일을 참고하기 바란다.)

- Build 메뉴의 Download to Target 을 클릭하면 download 가 진행이 된다.

### 2.3.2 Nand Flash(SD Card)에 파일 복사하기 (Mass Storage Mode 사용하기)

Download 가 완료 된 후 usb cable 을 컴퓨터와 STK board 의 device 단자에(STK board 의 con2) 연결을 한 후 board 에 전원을 켠다. 전원을 켜면 다음과 같이 새로운 저장 장치가 나타난다. 이 저장장치가 board 의 Nand Flash 가 잡힌 것으로 이곳에 파일을 복사하면 NandFlash 에 저장되어 adStar-L 프로그램에서 사용할 수 있다.



Demo Project 에서 사용하는 이미지 및 사운드 파일은 example 폴더의 flash\_data 폴더 안의 내용을 복사하면 된다.

만약 Nand Flash 대신 SD Card 에 파일을 복사하기 위해서는 usb\_mass\_storage project 의 mass\_stor.c 파일을 열어 상단에 주석으로 처리되어 있는 `#define SDCARD_STORAGE` 를 주석을 풀 다음 build 한 후 board 에 download 하면 위와 같은 방법으로 SD Card 에 파일을 복사할 수 있다.

### 2.3.3 Demo Program build & download

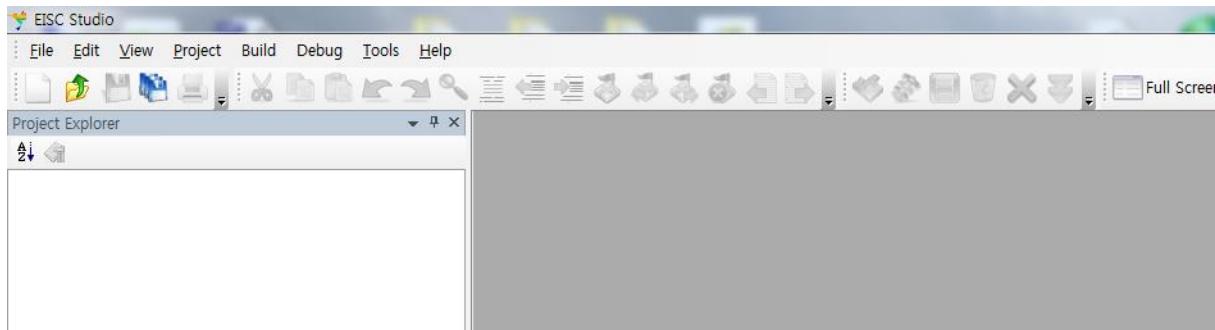
Nand Flash에 Demo Project에서 사용하는 이미지 및 사운드를 복사하였으면 example 폴더의 Demo Project 를 열어서 이전 usb\_mass\_storage project 때와 마찬가지로 Linker Script 를 adStar-L.ld 파일로 변경한 후 build 하여 E-Con 으로 download 하면 Demo 동작을 확인할 수 있다.  
( 이번 역시 bootloader 를 사용하지 않고 실행하는 것으로 내부 serial flash 0 번지에 download 하면 된다.)

## 2.4 adStar-L 새프로젝트만들기

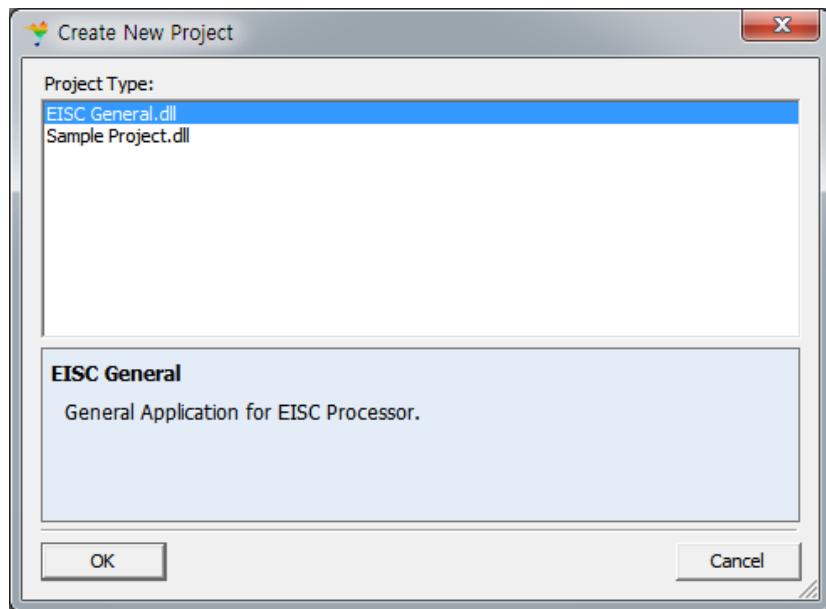
이번 장에서는 “Hello adchips!”라는 문장을

UART를 통해 출력하는 프로그램을 작성하면서 새프로젝트를 만드는 법에 대해 알아보겠다.

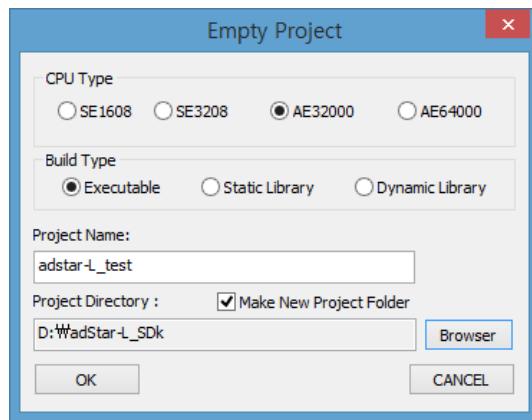
- 먼저 EISC Studio3를 실행한다.



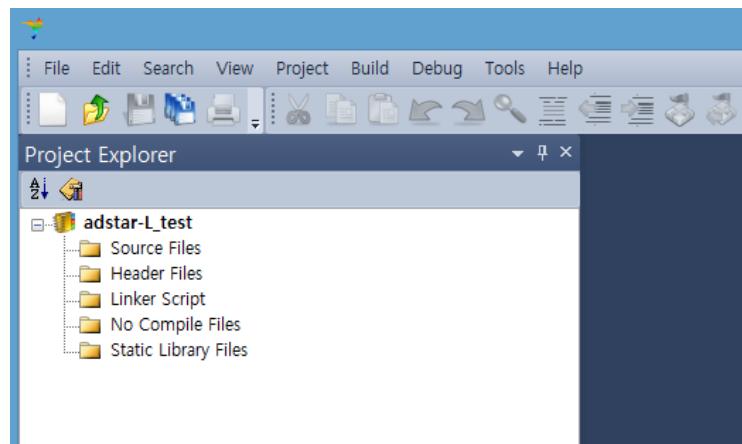
- 메뉴의 File → New → Project를 클릭하면 다음과 같은 창이 나오는데, 아래 그림과 같이 EISC General.dll을 선택한 후 OK 버튼을 누른다.



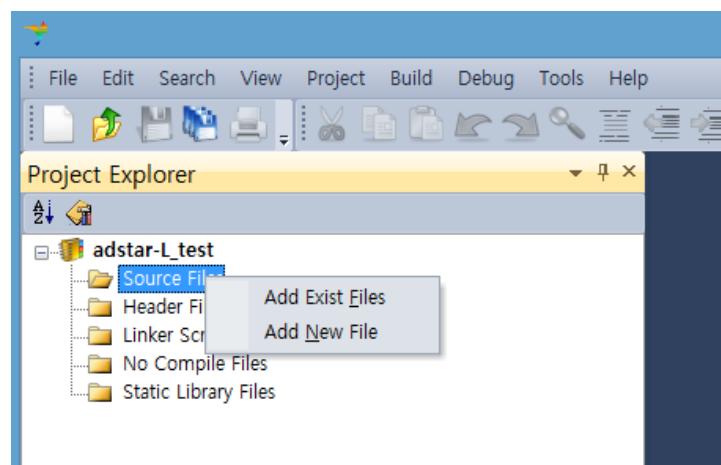
- 다음과 같이, project 설정창이 출력되는데, CPU Type, Build Type, Project Name, Project Directory를 결정해주어야 한다.  
adStar-L은 CPU Type으로 AE32000을 선택해주면되고, 동작프로그램을 작성할 것이기에 Build Type으로 Executable을 선택해준다. 다음으로 Project Name과 Project를 생성할 위치를 정해주면된다. Project Directory 옆에 위치한 체크박스(Make New Project Folder)는 Project Name과 같은 이름의 폴더를 생성 유무를 결정하는 것으로, 체크하게 되면 Project Directory로 지정해놓은 폴더에 Project Name과 같은 이름의 폴더를 생성한 후 그 안에 Project file을 생성해준다. 체크하지 않을 경우에는 Project Directory로 지정해놓은 폴더에 Project file을 생성한다. 모든 설정이 끝나면 OK를 클릭하여 새프로젝트 생성을 완료한다.



- 새로운 프로젝트가 생성되면 다음과 같이 Project Explorer 창에 Project 명과 Project tree 를 확인할 수 있는데, 이곳에 adStar-L 용 project 에 기본적으로 필요한 파일들을 추가해 주어야 한다.

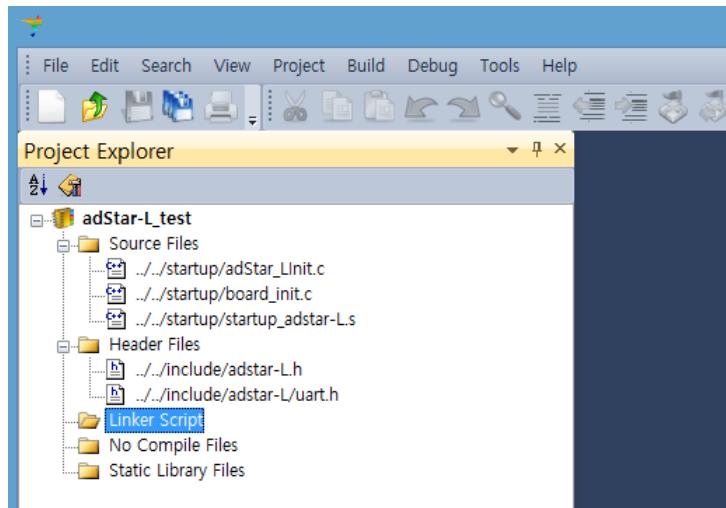


- 먼저 Source Files에서 마우스 오른쪽 키를 클릭하여 Add Exist Files를 선택하여 파일을 추가한다. 추가할 파일은 SDK의 startup 폴더에 있는 startup\_adStar-L.s, adStar-LInit.c, board\_init.c이다. (board\_init.c 파일은 adStar-L stk board를 사용할 경우 추가해주면 된다.)

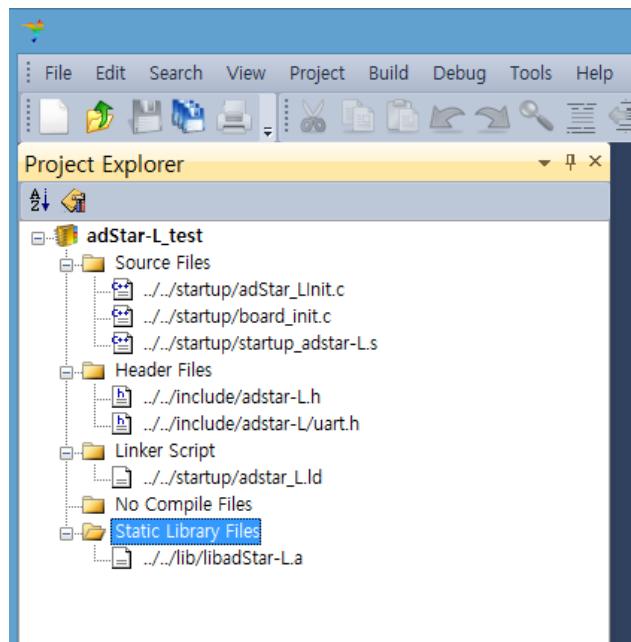


- 다음으로 Source Files 와 마찬가지로 Header Files에서도 마우스 오른쪽 키를 클릭하여 include 폴더에 있는 adStar-L.h 파일을 추가해준다. adStar-L.h 파일만 추가해줘도 되지만

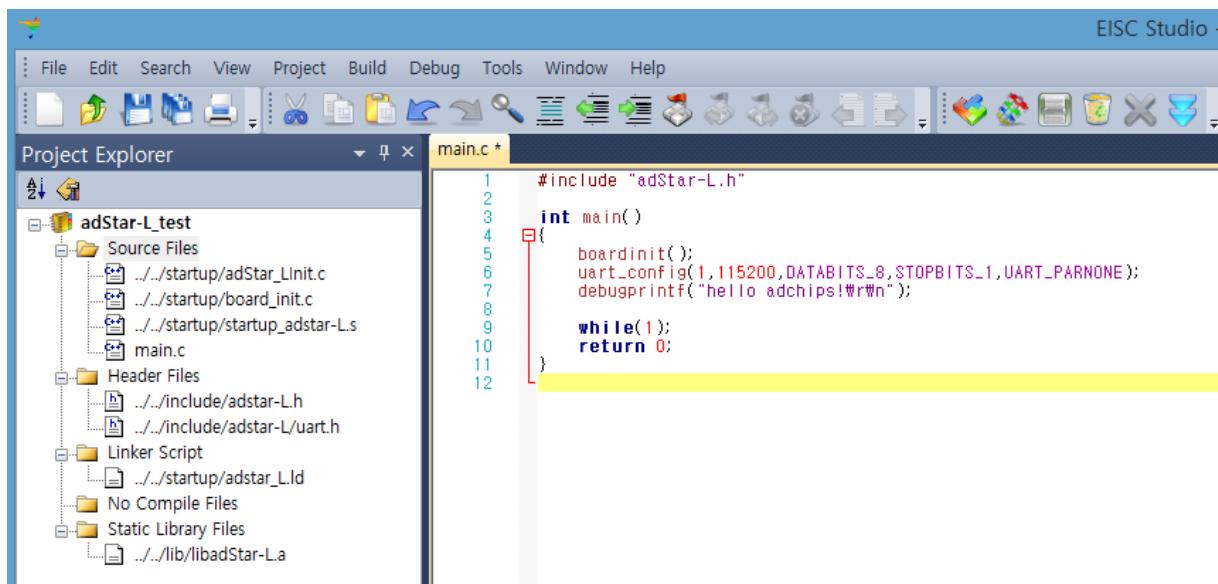
프로그램 작성 시 함수 사용에 도움을 받기 위해서는 include/adStar-L 폴더에 있는 관련 header 파일도 같이 추가해주면 좋다. 여기서는 UART 를 사용하는 프로그램을 작성 해 볼것이므로, UART.h 파일도 추가하도록 하겠다.



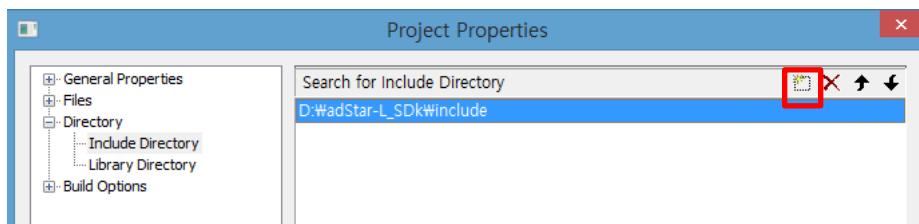
7. 다음으로 Linker Script 에 startup 폴더에 있는 adStar-L.ld 파일을 추가하고, Static Library Files 에 lib 폴더에 있는 libadStar-L.a 파일을 추가한다. 그리고 지금 작성하고 있는 UART 예제프로그램에는 필요없지만, MP3 재생이나, JPG 이미지 파일을 출력하는 프로그램 작성시에는 libmad.a 파일과 libjpeg.a 파일도 추가해주어야 한다.



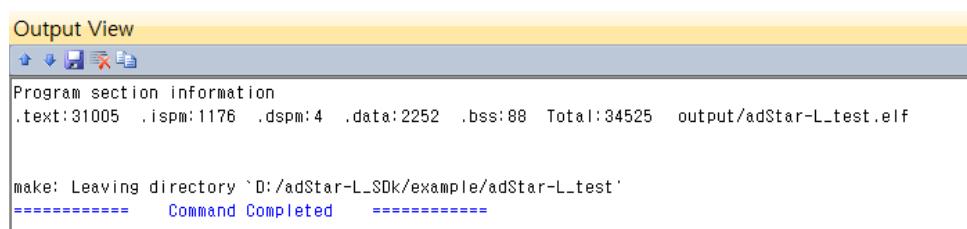
8. 위처럼 추가를 다 했으면 Source Files 에서 마우스 오른쪽 키를 누르고 Add New File 을 클릭 main 프로그램을 작성할 main.c 를 생성해주면 된다. 다음은 main.c 를 생성하고, UART 를 통해 Hello adchips!라는 문자열을 출력하는 코드를 작성 한 것이다.



9. 코드를 살펴보면, 제일 먼저 boardinit() 함수를 호출하여 사용하는 board에 맞게 pin 설정을 한다. (boardinit()함수는 사용하는 board마다 다르므로 board에 맞게 pin 설정을 해주어야 한다.) 다음으로 uart\_config()함수를 호출하여 UART를 설정한 값대로 초기화 해준다. 그리고 debugprintf()함수를 사용하여 "hello adchips!"를 출력한다.
10. 코드작성을 마쳤으면, project를 build해야하는데, 그 전에 먼저 project 설정을 해주어야한다. Project Explorer 창에서 project명에서 마우스오른쪽을 클릭하여 Properties를 클릭한다. 그러면 다음과 같은 창이 나오는데, Directory에서 Include Directory에 SDK의 include 폴더를 추가해준다. (우측상단의 점선박스를 클릭하면 폴더를 추가할 수 있다.)



11. include 폴더까지 추가를 했으면 Build 메뉴에서 Build Project를 클릭하거나, F7을 누르면 Project Build가 이루어진다. Build가 정상적으로 완료되었다면, Output View 창에 다음과 같이 출력되고, output 폴더에 project name.elf.bin 파일이 생성된다.



12. 생성된 bin 파일을 E-Con 을 사용하여 STK board 에 download 하면 Uart 1 번 채널을 통해 hello adchips!가 출력되는 것을 확인 할 수 있다. (STK board 의 con3)

## 3. Bootloader

이번 장에서는 adStar-L Bootloader에 대해 설명한다.

adStar-L 내부에 Flash를 포함한 것과, Flash를 포함하지 않은 칩을 위해, 내부 Flash Booting과 Nand Booting이 가능하다. 이에 맞춰 Bootloader도 Flash Booting을 위한 bootloader와 Nand Booting을 위한 Nand Boot Code를 지원한다.

### 3.1 Bootloader

Bootloader는 Serial Flash를 사용한 Flash Boot Mode<sup>3</sup>일 경우에 사용할 수 있도록 제작된 booting 프로그램으로, SDK Example의 bootloader 폴더 안에 있으며, adStar-L Booting 기능 외에 다음의 기능을 제공한다.

- 가. Remote Communication Mode.
- 나. Mass Storage Mode.
- 다. Execute Mode.
- 라. User Define Mode

Remote Communication Mode는 RemoteManCLI 프로그램을 사용하여 adStar-L와 Communication 할 수 있는 mode로, 주로 USB를 사용하여 application 프로그램을 DRAM으로 download하여 실행할 때 사용한다. application 프로그램의 동작을 빠르게 확인해 볼 수 있는 장점이 있다. 단 DRAM에 download하는 것이기 때문에, 동작을 확인할 때마다 download 해주어야 한다.

Mass Storage Mode는 example의 usb\_mass\_storage와 같은 동작을 수행하는 Mode로 adStar-L board의 Nand Flash를 usb memory와 같이 컴퓨터에서 저장 장치로 인식되도록 해준다. 따라서 bootloader가 adStar-L 칩에 download되어 있으면, Nand Flash에 file을 복사하기 위해 usb\_mass\_storage 예제를 download하지 않고도, bootloader의 Mode만 변경하여 file을 복사할 수 있다.

Execute Mode는 실행 모드로, Flash의 0x14000(sector number : 20)번지에 저장된 실행 프로그램을 Dram으로 복사한 후 복사한 프로그램을 실행하는 Mode이다.

마지막으로 User Define Mode는 사용자가 원하는 동작을 코딩하여 사용하도록 해 놓은 Mode인데 Bootloader에서는 기본적으로 Execute Mode와 비슷한 동작을 하도록 프로그램 되어 있다. Execute Mode와 다른 점은 내부 Flash에서 실행 프로그램을 복사하는 것이 아니라, Nand Flash(SD Card)에 boot.bin이라는 파일을 Dram으로 복사한 후 실행하도록 되어 있다.

자세한 동작 및 사용법에 대해서는 뒤에 설명하도록 하겠다.

<sup>3</sup> adStarL의 Boot Mode 중 하나로 Serial Flash를 사용하여 booting하는 mode이다. 자세한 사항은 datasheet를 참고하기 바란다.

### 3.1.1 Bootloader 사용하기

Bootloader 는 SDK example 안에 들어있다. 다른 application 과 마찬가지로 project 를 열어서 build 후에 adStar-L 에 download 하면 동작을 확인할 수 있다.

Bootloader project 를 열었으면 build → Build Project 를 실행하여 프로그램을 build 한다.

Build 가 완료되면, E-Con 을 board 와 연결한 후 전원을 켜고, build → Download to Target 을 실행하면 download 가 진행된다.

(build 및 download 의 좀더 자세한 설명은 본 매뉴얼의 2.3.1 장을 참고한다.)

Bootloader 에 몇 가지의 모드가 있는데, 코드상에서 다음 부분이 모드를 설정하는 부분이다.

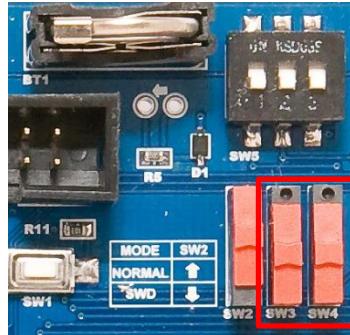
```
*R_GPIODIR(2)=0x1<<7;//mode switch
*R_GPIODIR(3)=0x1<<0; //mode switch
int mode = ((*R_GPILEV(2)>>6)&(1<<1)) | ((*R_GPILEV(3))&1);

switch(mode)
{
case 0:
    bin_execute();
    break;
case 1:
    RSP_Run();
    break;
case 2:
    usb_clock_init();
    mass_storage_main();
    break;
case 3:
    bin_execute_fat();
    break;
}
```

< bootloader project 의 main.c >

코드를 보면 P2.6 과 P3.1 PIO 를 입력으로 받아 mode 를 선택하게 되어 있는데, STK 보드상에서는 SW3 번과 SW4 번에 연결되어 있다.

( STK board 와 다른 PIO 를 사용하고 싶다면, 위 코드를 수정하여 사용자가 원하는 PIO 를 사용하면 된다.)



<Mode select switch SW3, SW4>

위의 사진에서 보듯이 booting mode 별 SW 설정은 다음과 같다.

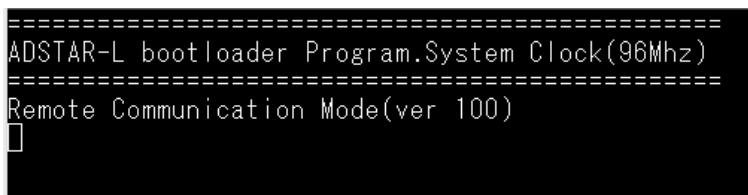
(SW2 는 SWD mode 설정으로 SW2 를 down 으로 한 후 전원을 켜면, SWD mode 로 동작하여, 프로그램이 시작되지 않고 시작점에서 멈춰 있다. Debugging 을 할 때 이 모드를 사용한다.

MODE	SW3	SW4
Execute Mode	down	down
USB Communication Mode	down	up
Mass Storage Mode	up	down
Excute Fat Mode	up	up

### 3.1.2 Bootloader Mode

#### 가. Remote Communication Mode

Remote Communication Mode 는 RemoteManCLI 프로그램을 사용하여 adStar-L 와 Communication 할 수 있는 mode 로, 주로 USB 를 사용하여 application 프로그램을 DRAM 으로 download 한 후 바로 실행하여, 동작을 빠르게 확인해 볼 때 사용한다. Remote Communication Mode 로 부팅을 하면 다음과 같은 debug message 가 출력된다.

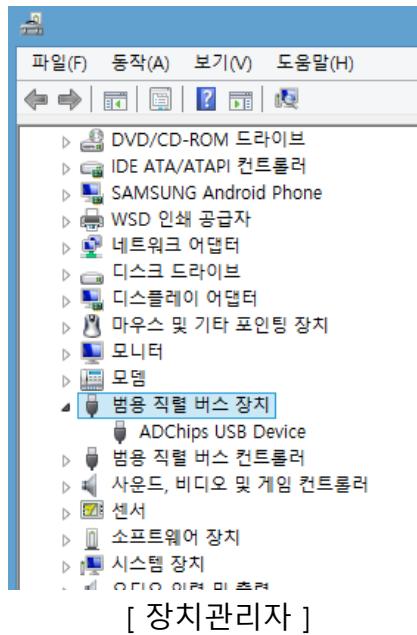


[ Remote Communication Mode ]

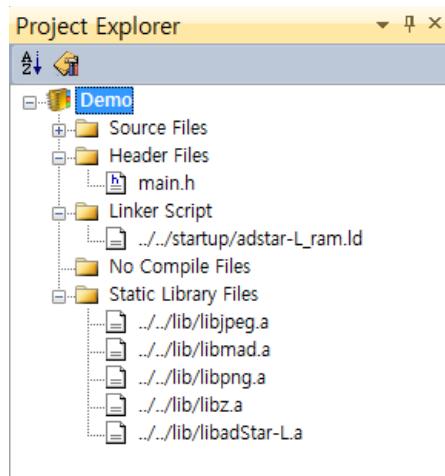
Remote Communication Mode 는 adStar-L USB device 를 사용하기 때문에 adStar-L USB Driver 를 설치해주어야 된다. USB Driver 는 pc-util/usb\_driver 폴더에 들어있다.

설치 방법은 usb\_driver 폴더 안에 있는 하나는 usb\_driver 폴더에 있는 ADChips USB Driver Install.exe 파일을 실행하여 설치를 진행하면 된다.

1. Usb\_driver 폴더의 ADChips USB Driver Install.exe 파일을 실행하여 설치를 진행한다.
2. 설치가 정상적으로 완료되면, usb 를 연결하고, bootloader 의 Remote Communication Mode 로 부팅을 한다.
3. 새로운 장치를 찾았다고 화면의 우측 하단에 표시되면서, driver 설치를 시작하게 된다. Windows7 을 사용하는 사용자라면 위 설치 동작으로 driver 가 복사되었기 때문에 자동으로 설치가 이루어진다. Windows xp 를 사용하는 사용자라면 “새 하드웨어 검색 마법사”창이 나타나는데, “소프트웨어 자동으로 설치(권장)”을 선택한 후 다음을 누르면 역시 driver 파일을 찾아 자동으로 설치가 이루어진다.
4. USB driver 설치가 완료되면, 장치관리자에서 adStar-L USB driver 가 설치된 것을 확인할 수 있다.



USB Driver 설치를 완료했으면 Remote Communication Mode에서 example 폴더에 있는 application 프로그램을 Dram에 download 하여 바로 확인을 해 볼 수 있는데, Dram에서 application을 동작시키기 전에 application project를 열어 link script 파일을 확인하도록 한다.

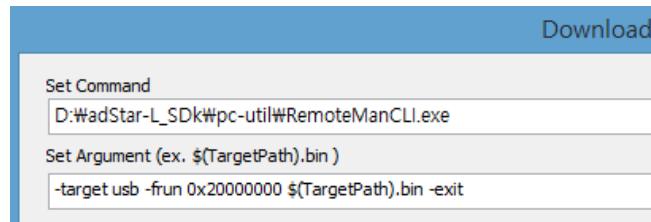


Demo Project는 기본적으로 Linker Script에 `../startup/adStar-L_ram.ld` 파일로 되어 있는데, 확인하였을 때 `adStar-L_ram.ld`로 되어 있으면 프로그램이 Dram에서 동작하도록 되어 있는 것이기 때문에 그대로 build하면 된다. 만약 `adStar-L.ld` 파일로 되어 있다면, 이는 Dram이 아닌 flash에서 동작하도록 되어 있는 것으로 startup 폴더의 `adStar-L_ram.ld` 파일로 변경해야 한다.

정리하면 Flash에서 동작하는 프로그램은 link script에 `adStar-L.ld` 파일을 load하면 되고, Dram에서 동작하는 프로그램의 경우에는 link script에 `adStar-L_ram.ld` 파일을 load하면 된다.

같은 프로그램에서 link script 파일만 변경하여 Flash 또는 Dram 에서 동작하는 binary 를 생성할 수 있다. (bootloader 를 사용할 경우 Dram 에서 application 을 동작시키므로 Bootloader 를 사용할 경우 특별한 경우가 아니라면 application 의 link script 는 adStar-L\_ram.ld 파일을 사용하면 된다.)

Link script 를 확인하였으면 다음으로 Dram 에 download 하여 실행하는 방법에 대해 알아보겠다. build 한 후에 adStar-L 를 Remote Communication Mode 로 부팅한다. 그런 후에 ECon 으로 download 할 때와 마찬가지로 build→download option 을 실행한다.



[ build -> download option ]

Dram 에 download 하기 위해서는 EConMan 프로그램이 아닌 RemoteMenCLI 라는 프로그램을 사용해야 하는데, pc\_util 폴더에 있으므로 찾아서 선택하면 된다.

Argument 는 EConMan 과 비슷한데, 설명하면 다음과 같다.

#### **-target usb**

➔ target 명을 명시하는 부분으로 usb 라고 쓰면된다.

#### **-frun 0x20000000 \$(TargetPath).bin**

➔ binary file 을 downlaod 하고 실행하는 명령어로, Dram 영역인 0x20000000 에 현재 project 에서 생성된 binary 파일을 download 한 후 실행한다.

#### **-exit**

➔ 모든 동작이 정상적으로 이루어 졌으면 프로그램을 종료한다.

위처럼 download option 을 설정 한 후에, Remote Communication Mode 상태에서 build→download to target 을 실행하면 현재 열려져 있는 project 의 binary 파일 download 가 진행되고, 진행이 완료되면 바로 실행한다.

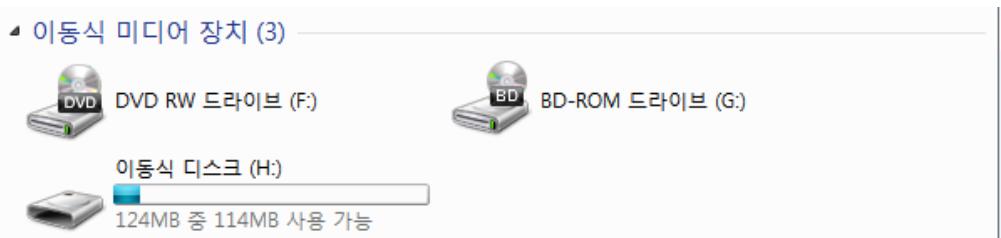
처음에 설명 했듯이, Remote Communication Mode 상태에서 application 을 빨리 확인을 해 볼 수 있지만, Dram 에 download 하여 실행하는 것으로, reset 시 다시 download 해야 한다는 점을 알고 있어야 한다. reset 시에도 항상 같은 동작을 하도록 하고 싶은 경우에는 Flash 에서 application 을 동작하도록 하거나, 뒤에 설명할 Execute Mode 를 사용하면 된다.

## 나. Mass Storage Mode

Mass Storage Mode 는 example 의 usb\_mass\_storage 와 같은 동작을 수행하는 Mode 로 adStar-L board 의 Nand Flash 를 usb memory 와 같이 컴퓨터에서 저장 장치로 인식되도록 해준다. 따라서 Mass Storage Mode 를 이용하여 usb\_mass\_storage 프로그램을 새로 download 할 필요 없이 Nand Flash 를 저장 장치로 사용할 수 있다. Mass Storage Mode 로 부팅을 하면 다음과 같이 debug message 를 출력하고, usb\_mass\_storage 와 같이 이동식 디스크 장치라는 저장장치가 생성된다.

```
=====
ADSTAR-L bootloader Program.System Clock(96Mhz)
=====
NAND Manufacturer : Samsung, 128MiB 3.3V 8-bit
(..../lib_src/nandctrl.c:1282) make bad block information done(bad-block(0))
USB Mass-Storage Mode Running(Interrupt Mode)
```

[ Mass Storage Mode ]



[ 이동식 디스크 저장 장치 ]

usb\_mass\_storage 와 마찬가지로 이 저장장치에 이미지 및 사운드 파일을 복사할 수 있다.

## 다. Execute Mode

Execute Mode 는 booting 과 동시에 application 이 동작하는 모드로, 주로 application 개발이 완료 되었을 때 사용하는 모드이다. Execute Mode 로 booting 을 하면 다음과 같이 터미널 창에 출력되는 것을 확인할 수 있다.

```
=====
ADSTAR-L bootloader Program.System Clock(96Mhz)
=====
Execute Mode, Run boot.bin from FLASH
startfp : 0x200002fc
=====
ADSTARLOW Drawing View.System Clock(96Mhz)
=====
(crtc.c:170) CRTC 480 x 272 Setting Done
RGB565 Mode
```

[ Execute Mode ]

Execute Mode 는 내부 Flash 의 data 를 Dram 으로 복사하여 실행하는 형태로 build 하여 생성된 project\_name.elf.bin 파일을 지정된 내부 Flash 영역에 download 한 후에 Execute mode 로 부팅을 하면 application 이 동작을 한다. Dram 으로 복사되어 실행될 프로그램이기에 link script 는 adStar-L\_ram.ld 파일을 사용한다.

지정된 내부 Flash 영역은 20 번째 섹터로 address 로 하면 0x14000 번지이다.  
bootloader 에서 0x14000 번지에서부터 data 를 dram 에 복사하여 실행하게 된다

```

46     #define FLASH_APP_OFFSET (1024*4*20)
47     static void bin_execute()
48     {
49         void (*entryfp)();
50         debugstring("Execute Mode, Run boot.bin from FLASH #r\n");
51         memcpy((void*)0x20000000,(void*)FLASH_APP_OFFSET,(512*1024)-FLASH_APP_OFFSET);
52         dcache_invalidate_way();
53         entryfp = (void(*)())*(U32*)0x20000000;
54         debugprintf("startfp : %#xr\n",entryfp);
55         entryfp();
56     }

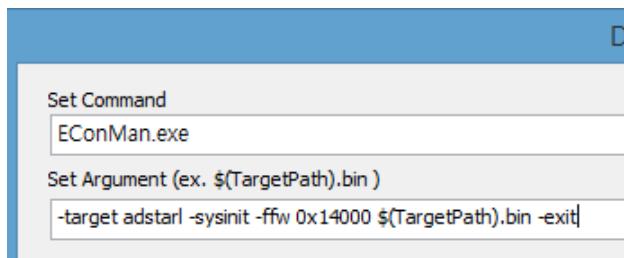
```

[ bootloader main.c ]

Main.c 의 65 번재 줄을 보면 Flash 영역의 시작 주소가 정의 되어 있는데, bootloader 의 size 가 커져서 영역이 겹치게 되면 정의 값을 수정해주어야 한다.

내부 Flash 의 0x14000 번지에 download 하는 방법은 다음의 두 가지 방법 중 하나를 사용하면 된다.

### 1) EConMan 을 사용

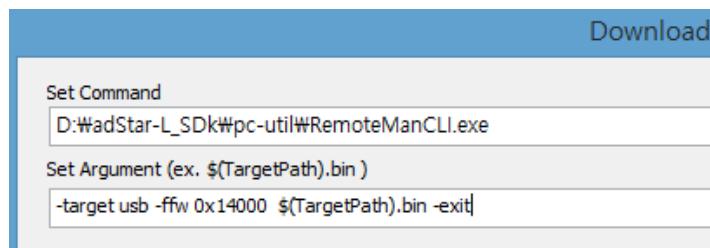


[ download option ]

기존에 EConMan 프로그램을 사용하여 download 하는 방법에서 address 를 0x14000 으로만 해주면 된다. 이런 식으로 원하는 address 에 binary 파일을 download 할 수 있다.

### 2) RemoteManager 사용

Bootloader 의 Remote Communication Mode 로 booting 하여 RemoteManCLI 를 사용하여 0x14000 에 download 할 수 있다. 방법은 Download Option 을 실행하여 다음과 같이 적어주면 된다.



-ffw 라는 command 를 사용하여 Flash 의 지정한 0x14000 번지에 application 을 download 한다.

EConMan 또는 RemoteManCLI 를 사용하여 download 를 완료한 후에 Execute mode 로 booting 하면 application 이 동작하는 것을 확인할 수 있다.

#### 라. User Define Mode (execute\_fat)

User Define Mode 는 사용자가 원하는 동작을 코딩하여 사용할 수 있도록 만들어 놓은 Mode 이지만, bootloader 에서 기본적으로 Execute Mode 와 비슷한 동작을 하도록 프로그램 되어 있다. Booting 을 하면 Execute Mode 와 같이 바로 application 이 동작을 한다. 다만 Execute Mode 가 내부 Flash 에 저장되어 있는 프로그램을 Dram 에 복사하여 실행하는 것이었다면, User Define Mode(execute\_fat)에서는 Nand Flash(SD Card)의 boot.bin 파일을 Dram 으로 복사한 후에 실행한다는 점이 다르다.

Nand Flash(SD Card)의 boot.bin 파일을 Dram 으로 복사하여 실행하는 형태에 대해 설명하면, build 하여 생성된 project\_name.elf.bin 파일을 boot.bin 파일로 이름을 변경하여 Mass Storage 모드로 부팅, 저장장치 root 에 복사를 한 후, User Define Mode 로 부팅을 하면 저장한 application 이 동작을 한다.

여기서 주의할 점은 application 을 Dram 에서 동작시키기 때문에 link script 를 Dram 에서 동작하도록 설정 된 adStar-L\_dram.ld 파일을 사용하여야 한다.

이름	수정한 날짜	유형	크기
font	2012-02-16 오후...	파일 폴더	
icon	2012-02-16 오후...	파일 폴더	
mp3	2012-02-16 오후...	파일 폴더	
png	2012-02-16 오후...	파일 폴더	
wav	2012-02-16 오후...	파일 폴더	
weather	2012-02-16 오후...	파일 폴더	
bg.bmp	2011-11-09 오후...	비트맵 이미지	1,126KB
bg.jpg	2011-11-09 오후...	JPEG 이미지	82KB
boot.bin	2012-02-16 오후...	BIN 파일	444KB
FreeRTOS.bmp	2011-11-09 오후...	비트맵 이미지	72KB
Microphone-bg.jpg	2011-11-09 오후...	JPEG 이미지	32KB
mp3bg.bmp	2011-11-09 오후...	비트맵 이미지	1,126KB
Penguins.jpg	2011-11-09 오후...	JPEG 이미지	760KB
t1.bmp	2011-11-09 오후...	비트맵 이미지	77KB
t2.bmp	2011-11-09 오후...	비트맵 이미지	77KB
t3.bmp	2011-11-09 오후...	비트맵 이미지	77KB

[ Nand Flash 의 boot.bin 파일 ]

SD Card 에서 boot.bin 파일을 Dram 으로 복사하여 실행 할 수도 있는데, main.c 의

4 번째줄의 #define FAT\_APP\_TYPE DRIVE\_NAND 대신 #define FAT\_APP\_TYPE DRIVE \_SDCARD 를 사용하면, SD Card 의 boot.bin 파일을 Dram 에 복사하여 실행한다.

### 3.2 Nand Boot Code

Nand Boot Code 는 내부 Flash 가 없는 adStar-L 를 사용할 경우 사용하는 Nand Booting 용 bootloader 로 Nand Flash Boot Mode<sup>4</sup>로 설정이 되어 있어야 한다.

Nand Flash Boot Mode 는 Nand Flash 0 번 블록의 2Kbyte data 를 내부 sram 에 복사하여 booting 하는 mode 이기 때문에, Nand Boot Code 의 크기는 2Kbyte 로 제한된다. 따라서 Nand Boot Code 는 수정 없이 제공되는 것을 그대로 사용하기를 권장한다.

Nand Boot Code 가 하는 역할은 Nand Flash 1 번 블록부터 7 개 블록의 data(application program)를 Dram 으로 복사하여 실행하는 역할을 수행한다.

```
#define BOOTLOADER_STARTADDR 0x20000000
#define COPY_BLOCK 7//128K+7Mbyte
int main()
{
    void (*startfp)();
    boardinit();
    myuart_init();
    *R_NFMCFG = 0x2222;
    nand_reset();
    mydebugstring("Nand Booting.\r\n");
    //U16 id = nand_id_read();
    copydata_from_nand((U8*)BOOTLOADER_STARTADDR,1,COPY_BLOCK);
    startfp = (void (*)(*))(*(volatile unsigned int*)BOOTLOADER_STARTADDR);
    startfp();
    while(1);
    return 0;
}
```

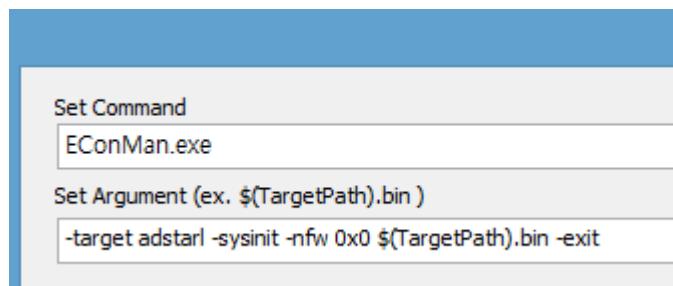
[ nand\_boot\_code 의 main.c ]

#### 3.2.1 Nand Boot Code 사용하기

Nand Boot Code 는 SDK example 안에 들어있다. 다른 application 과 마찬가지로 project 를 열어서 build 후에 Nand Flash 의 0 번 블록에 download 하면 된다.

Build 는 이전에 설명한 방법대로 진행을 하면 되는데, download 는 기존 내부 Flash 에 download 하는 것과는 command 가 다르다. 다음은 Nand Flash 에 download 시 download option 이다. 참고하기 바란다.

<sup>4</sup> adStar 의 Boot Mode 중 하나로 Nand Flash 를 사용하여 booting 하는 mode 이다. 자세한 사항은 datasheet 를 참고하기 바란다.

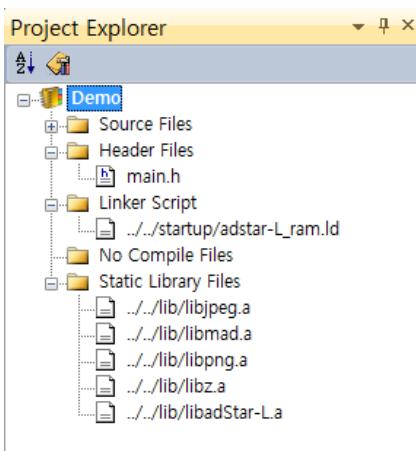


[ Download Option ]

'-nfw' command 를 사용하여 Nand Flash 에 download 한다.

다음으로 Nand Boot Code 에서 Dram 에 복사하여 실행할 코드를 download 하는 방법에 대해 알아보겠다.

먼저 Nand Boot Code 에 의해 Dram 에 복사될 코드는 Dram 에서 수행되기 때문에 link script 가 adStar-L\_ram.ld 인지 확인을 한다.

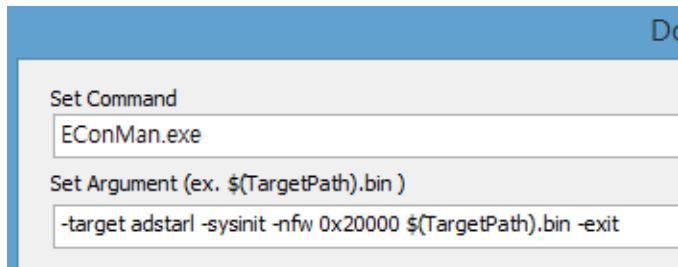


[ Linker Script ]

확인을 한 후 build 과정을 거쳐 download 를 진행하면 된다. 이 역시 Nand Flash 에 download 하므로 Nand Boot Code 를 download 할 때와 같은 command 를 사용한다. 다른 점은 download 하는 위치가 달라진다.

Nand Boot Code 에 의해 Dram 에 복사되어 실행할 코드는 Nand Flash 의 1 번 블록에 download 한다. Nand Flash 1 번 블록의 address 는 Nand Flash 마다 다르므로 주의하기 바란다.

(기본적으로 small page Flash 의 경우 1 번 블록은 0x4000 이고, large page Flash 의 경우 1 번 블록은 0x20000 이다.)



[ Download Option ]

Download 완료 후 실행하면 다음과 같이 Nand Booting 라고 출력되고, 1 번 블록에 download 한 application 이 실행되는 것을 확인할 수 있다.

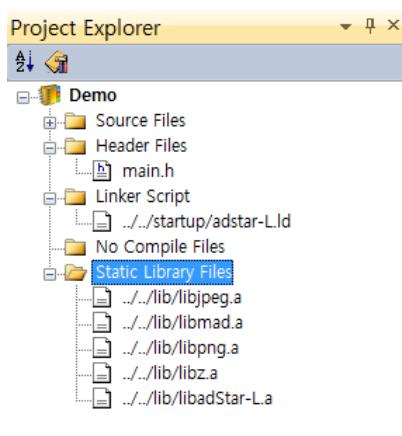
1 번 블록에 Application 대신 Bootloader 를 download 하여 실행할 수도 있다. 이때는 Bootloader 에서 Linker script 파일로 bootloader.ld 파일이 아닌, bootloader\_ram.ld 파일을 사용해야 한다.

Nand Booting.  
START

### 3.3 BootLoader 없이 사용하기

지금까지 Bootloader 와 Nand Boot Code 에 대해서 알아 보았는데, 이번에는 Bootloader 를 사용하지 않고 프로그램을 실행하는 방법에 대해 설명하도록 하겠다.

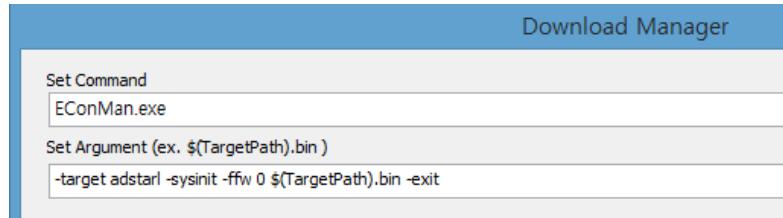
Bootloader 를 사용하지 않으면 Linker Script 파일과, 프로그램을 download 하는 위치가 달라지게 된다. Linker Script 는 adStar-L\_ram.ld 파일 대신 adStar-L.ld 파일을 사용한다. 앞에서 몇 차례 설명하였지만, adStar-L\_ram.ld 파일은 Dram 에서 프로그램이 동작하도록 설정되어 있는 파일이고, adStar-L.ld 파일은 flash 에서 프로그램이 동작하도록 설정되어 있는 파일이다. Bootloader 를 사용하지 않으면 실행 프로그램을 flash 에서 동작시켜야 하므로 adStar-L.ld 파일을 사용해야 한다. adStar-L.ld 파일은 startup 폴더에 있으므로 변경해주면 된다.



[ Linker Script ]

Linker Script 를 변경하였으면, build 후에 download 해주어야 하는데, bootloader 를 사용하지 않기 때문에, RemoteCLI 프로그램을 사용할 수 없고, EConMan 프로그램을 사용해서 download 해야 한다.

Build → Download Option 에 다음과 같이 적고, Download to Target 을 실행하면 된다.



Bootloader 를 사용하지 않기 때문에 0 번지에 바로 download 하면 된다. Download 가 완료된 후 reset 을 하면 동작을 확인할 수 있다. 정상적으로 download 하였는데도 아무런 동작을 하지 않는다면, Linker Script 가 adStar-L.ld 로 되어있는지 확인하기 바란다.

## 4. lib\_config.h

adStar-L SDK 의 lib\_src 를 보면 Header Files 에 lib\_config.h 파일이 있는데, 파일명에서도 알 수 있듯이 lib\_config.h 파일은 adStar-L SDK library 의 설정 파일이라고 보면 된다. 이 장에서는 lib\_config.h 파일에 대해 설명하도록 하겠다.

```

1  /******
2   Copyright (C) 2011      Advanced Digital Chips Inc.
3   |           http://www.adc.co.kr
4   Author : Software Team.
5
6   이 파일을 수정 함으로써 전체 실행파일 사이즈와 메모리 사용량을 줄일 수도 있다.
7   하지만 줄어든 만큼 성능의 저하가 발생 할 경우도 있다.
8   제거 하고자 하는 기능은 0 으로 설정 함으로써 제거된다.
9
10  ******/
```

[ lib\_config.h 요약 ]

lib\_config.h 는 Nested Interrupt 설정에 대한 내용으로 시작한다.

```

/******
Nested Interrupt
특정 인터럽트 서비스 함수(IRQ)내에서 또 다른 인터럽트를 받아 들일 것인지에 대한 설정
특정 인터럽트를 1 로 설정 할 경우 해당 인터럽트 핸들러내에서 모든 다른 인터럽트를 받아 들인다.
단, 인터럽트 진입 속도가 그렇지 않은 경우 보다 느린다.
ex)
SDK 내의 MP3 decoding 은 DMA 인터럽트내에서 frmae 단위로 이루어 진다.
따라서 이 구간내에서도 다른 인터럽트를 받아 들이고자 한다면
모든 DMA 인터럽트를 1 로 설정 하면 된다.
******/
#define SUPPORT_NESTED_INTPIN_EIROUT_0
#define SUPPORT_NESTED_INTPIN_SYSTIMER_0
#define SUPPORT_NESTED_INTPIN_TIMER0_0
#define SUPPORT_NESTED_INTPIN_SOUND_MIXER_0

#define SUPPORT_NESTED_INTPIN_EIROUT1_0
#define SUPPORT_NESTED_INTPIN_FRAMESYNC_0
#define SUPPORT_NESTED_INTPIN_DMA0_1          //DMA is used by sound , 1 is recommended
#define SUPPORT_NESTED_INTPIN_GPIOA_0

#define SUPPORT_NESTED_INTPIN_UART0_0
#define SUPPORT_NESTED_INTPIN_DMA1_1
#define SUPPORT_NESTED_INTPIN_TIMER1_0
#define SUPPORT_NESTED_INTPIN_PMC_0

#define SUPPORT_NESTED_INTPIN_DMA2_1
#define SUPPORT_NESTED_INTPIN_ADC_0
#define SUPPORT_NESTED_INTPIN_USB_0
#define SUPPORT_NESTED_INTPIN_GPIOB_0

#define SUPPORT_NESTED_INTPIN_UART1_0
#define SUPPORT_NESTED_INTPIN_DMA3_1
#define SUPPORT_NESTED_INTPIN_SPI0_0
#define SUPPORT_NESTED_INTPIN_MJPEG0_0

#define SUPPORT_NESTED_INTPIN_GPIOC_0
#define SUPPORT_NESTED_INTPIN_SDHC_0
#define SUPPORT_NESTED_INTPIN_GPIOD_0
#define SUPPORT_NESTED_INTPIN_WATCHDOG_0

#define SUPPORT_NESTED_INTPIN_NAND_0
#define SUPPORT_NESTED_INTPIN_TWI_0
#define SUPPORT_NESTED_INTPIN_RTC_0
#define SUPPORT_NESTED_INTPIN_RTC_ALARM_0

#define SUPPORT_NESTED_INTPIN_SPI1_0
#define SUPPORT_NESTED_INTPIN_CAP_OVERFLOW_0
#define SUPPORT_NESTED_INTPIN_MJPEG1_0
#define SUPPORT_NESTED_INTPIN_CLOCK_CTRL_R_0
```

Nested Interrupt 사용을 설정하는 부분으로 1로 설정하면 해당 Interrupt routine에서 Nested Interrupt를 허용한다는 의미이다. Nested Interrupt 설명에 적혀 있듯이, MP3 decoding 이 DMA Interrupt 내에서 frame 단위로 이루어지는데, 그 사이에 다른 Interrupt를 받아 들여야 하므로 DMA 쪽 Interrupt는 default로 Nested Interrupt를 허용해 놓는다. 이 외에도 Nested Interrupt를 허용하고 싶은 경우에는 1로 설정하면 된다. 중첩 허용은 최대 14개까지 가능하므로, 꼭 필요한 경우만 사용하기 바란다.

```

/* **** INTERNAL TIME OUT ****
 * TWI Function에서 Status를 기다리는 최대 시간을 설정한다.
 */
#define TWI_RESP_TIME_OUT_COUNT (7200*100) // About 100ms @ AHB 101MHz

/* **** External NAND/SD-Card/USB Memory Supports ****
 * CONFIG_NAND 1
 * CONFIG_SDCARD 1
 * CONFIG_USBHOST 0

```

다음은 TWI Timeout count 값을 설정하는 부분이 있고, 사용할 저장 장치를 설정하는 부분이 있다. 이곳에서 사용할 저장장치를 설정하여 사용하지 않는 저장장치에 관련된 부분의 size를 줄일 수 있다. 기본 설정은 NAND Flash와 SD Card를 사용하도록 되어있다.

```

/* **** FAT File System ****
 * //drive number
 * #define DRIVE_NAND 0
 * #define DRIVE_SDCARD 1
 *
 * #define CONFIG_FAT_READONLY 0
 * #define CONFIG_FAT_VOLUMES 3
 *
 * /*
 * 아래 설정 값은 include/fatfs/ffconf.h 파일을 참조.
 * 영문만을 사용 할 경우
 * CONFIG_CODE_PAGE 437 로 설정
 * library 프로젝트에 lib_src/fatfs/option/ccsbc5.c 파일을 추가 한 후 rebuild 한다
 * 한글을 사용 할 경우
 * 약 136,408 byte 가 늘어 난다.
 * CONFIG_CODE_PAGE 949 로 설정
 * library 프로젝트에 lib_src/fatfs/option/cc949.c 파일을 추가 한 후 rebuild 한다
 */
#define CONFIG_CODE_PAGE 949

```

사용하는 저장장치의 driver number를 설정하는 부분과 driver의 개수를 설정하는 부분이다. 그리고 filesystem에서 영문만을 사용할지 한글도 사용할지에 따라 설정을 해줄 수 있는데, 적힌 내용처럼 영문만 사용할 경우 변경을 해주면 filesystem 관련 코드 사이즈가 줄어든다.

```

/* *****
 *      UART Config
 *****/
/*
 * 인터럽트 모드일 경우      uart 송/수신 data 를 임시로 저장 할 버퍼의 사이즈이다.
 */
#define UART_BUF_SIZE 512
#define CONFIG_UART_RX_INTERRUPT
//#define CONFIG_UART_TX_INTERRUPT

/*
 * debugstring, debugprintf 할수의 최종 출력 채널
 */
#define DEBUG_CHANNEL 1

```

Uart Config 에서는 Uart library 에서 기본적으로 RX interrupt 를 사용하고, Ring buf 를 사용하여 uart 로 전송되는 data 를 저장을 하는데, data 를 저장하는 Ring buf 의 사이즈를 결정하고, 사용할 Interrupt 를 설정한다. 기본적으로 library 에서 RX interrupt 사용하도록 설정 되어 있고, Uart Interrupt Routine 도 작성되어 있다. CONFIG\_UART\_TX\_INTERRUPT 정의를 사용하면 UART Tx 전송을 interrupt 를 사용하여 전송되는데, debugstring( ), debugprintf( )함수는 poling 방식으로 동작한다. ( uart\_putch, uart\_putdata, uart\_putstring 함수는 interrupt 방식으로 동작한다. )

```

/* *****
 *      SYSTEM CLOCK
 *****/
/*
 * 외부 OSC 입력 값
 * STK 보드의 경우 10Mhz OSC 를 사용한다.
 * 10Mhz 가 아닌 경우 adStarInit.c 의 PMUInit() 함수에서 PLL 설정 값을 수정 해야 한다.
 * 사용 가능한 주파수는 doc/adStar_PLL_Setting.xls 파일을 이용.
 */
#define OSC_CLOCK 2000000

/* *****
 *      SOUND Mixer
 *****/
/*
 * 버퍼 사이즈를 줄이면 메모리 사용량은 줄어 들지만
 * 인터럽트 발생 횟수가 늘어 난다.
 * 만약 wav 파일을 사용하지 않고 MP3 만 재생 한다면
 * 1 프레임씩 디코딩 하므로 이 사이즈를 약 5kbyte 만 설정 해도 충분하다.
 */
#define WAVE_BUF_MAX (512*1024) //if you use MP3, > 5kbyte
#define WAVE_RELOAD_BUF_MAX (1024*128)
#define DEFAULT_VOLUME 255//max 255

/*
 * 최종 출력 sample rate
 */
#define SND_OUTPUT_HZ 48000

```

lib\_config.h 의 마지막으로 System clock 설정과 Sound mixer 에 대한 설정이 있다. Sound mixer 와 관련하여 buffer size, volume, output channel, sample rate 를 설정할 수 있다.

그리고 System clock 에서는 입력 clock 을 설정하는 부분이 있는데, 입력 clock 으로 2Mhz 가 아닌 다른 Clock 를 사용할 경우 이 부분을 수정 해주면 된다. 그리고 PMUInit()함수에서 입력 Clock 에 대한 PLL setting 을 해주어야 원하는 System clock 으로 adStar-L 를 동작시킬 수 있다.

lib\_config.h 파일의 모든 설정이 완료 되었으면 library 를 다시 build 하여 libadStar-L.a 파일을 새로 생성하여 사용하여야 한다.

# 5. UART

## 5.1 uart\_config

```
BOOL uart_config(int ch, int baud, UART_DATABITS databits, UART_STOPBITS stopbit,
UART_PARITY parity )
```

UART 사용을 위한 초기 설정을 하는 함수이다.

lib\_config.h 파일에 #define CONFIG\_UART\_RX\_INTERRUPT 가 정의 되어 있으면, Uart Rx Interrupt 설정도 같이 한다. (기본적으로 CONFIG\_UART\_RX\_INTERRUPT 가 정의 되어 있다.)

### Parameter

- |          |  |
|----------|--|
| ch       | UART channel 값. 초기 설정할 channel 을 입력한다.                             |
| baud     | UART baud rate 값. Baud rate 값을 설정한다.<br>SDK 는 기본적으로 115200 을 사용한다. |
| databits | UART 전송 data bit 을 설정한다. SDK 에서는 data bit 를 다음과 같이 정의하고 있다.        |

```
typedefenum{
    DATABITS_5 = 5,
    DATABITS_6 = 6,
    DATABITS_7 = 7,
    DATABITS_8 = 8
}UART_DATABITS;
```

- |         |  |
|---------|--|
| stopbit | UART 의 stop bit 를 설정한다. SDK 에서는 stop bit 를 다음과 같이 정의하고 있다. |
|---------|--|

```
Typedefenum{
    STOPBITS_1 = 1,
    STOPBITS_2 = 2
}UART_STOPBITS;
```

- |        |  |
|--------|--|
| parity | UART 의 parity bit 를 설정한다. SDK 에서는 parity bit 를 다음과 같이 정의하고 있다. |
|--------|--|

```
Typedefenum{
    UART_PARNONE = 0,
    UART_PARODD,
    UART_PAREVEN
}UART_PARITY;
```

### Return Value

TRUE( 1 ) or FALSE( 0 )

## 5.2 uart\_putchar

**BOOL** uart\_putchar(**int** n, **char** ch )

UART n 채널을 통해 ch 값(한 문자)을 출력한다.

### Parameter

- n                   값을 출력할 UART channel 값. adStar-L 는 UART 5 개 channel 을 가지고 있다.  
Ch                 UART 를 통해 출력할 값.

### Return Value

TRUE( 1 ) or FALSE( 0 )

## 5.3 uart\_putdata

**BOOL** uart\_putdata(**int** n, **U8\*** buf, **int** len )

UART n 채널을 통해 buf 에 저장되어 있는 문자를 len 만큼 출력한다..

### Parameter

- n                   값을 출력할 UART channel 값. adStar-L 는 UART 5 개 channel 을 가지고 있다.  
buf                출력할 문자열이 저장되어 있는 buffer.  
len                출력할 문자열의 개수.

### Return Value

TRUE( 1 ) or FALSE( 0 )

## 5.4 uart\_putstr

**BOOL** uart\_putstr(**int** n, **U8\*** buf)

UART n 채널을 통해 buf 에 저장되어 있는 문자열을 출력한다.

### Parameter

- n                   값을 출력할 UART channel 값. adStar-L 는 UART 5 개 channel 을 가지고 있다.  
buf                출력할 문자열이 저장되어 있는 buffer.

### Return Value

TRUE( 1 ) or FALSE( 0 )

## 5.5 uart\_getch

**BOOL** uart\_getch(**int** n, **char\*** ch )

UART n 채널을 통해 1Byte 값(한 문자)을 받아 ch에 저장한다.

### Parameter

n              값을 입력 받을 UART channel 값. adStar-L는 UART 5 개의 channel을 가지고 있다.  
 ch             UART를 통해 값을 입력 받을 변수.

### Return Value

TRUE( 1 ) or FALSE( 0 )

## 5.6 uart\_getdata

**int** uart\_getdata( **int** n, **U8\*** buf, **int** bufmax )

UART n 채널을 통해 bufmax 만큼 값(문자)을 읽어서 buf에 저장한다.

### Parameter

n              값을 입력 받을 UART channel 값.  
 Buf            UART를 통해 값을 입력 받을 buffer.  
 Bufmax        값(문자)을 읽을 개수. (Byte 단위)

### Return Value

읽은 data byte 수. Bufmax 값과 비교하여 같으면 함수가 정상적으로 처리 한 것이고, 그렇지 않을 경우에는 정상 처리하지 못한 것이다.

## 5.7 uart\_rx\_flush

**void** uart\_rx\_flush( **int** ch )

UART n 채널의 rxfifo를 초기화 시킨다.

### Parameter

ch             초기화 시킬 UART channel 값.

### Return value

없음.

## 5.8 uart\_tx\_flush

**void** uart\_tx\_flush(**int** ch )

UART n 채널의 txfifo 를 초기화 시킨다.

### Parameter

ch                  초기화 시킬 Uart channel 값.

### Return value

없음.

## 5.9 set\_debug\_channel

**Void** set\_debug\_channel(**int** ch )

디버깅용으로 사용하는 **debugprintf**, **debugstring**, **PRINTVAR**, **PRINTLINE** 함수에 의해 디버깅 메시지가 출력 될 UART 채널을 결정한다.

### Parameter

ch                  디버깅용으로 사용할 UART 채널.

### Return value

없음.

## 5.10 get\_debug\_channel

**int** get\_debug\_channel( )

디버깅용으로 사용하는 UART 채널 값을 return 한다.

### Parameter

없음

### Return Value

현재 설정된 디버깅용 UART 채널 값.

## 5.11 debugprintf

**void** debugprintf(**const char\***const format, . . . )

c 언어의 printf와 같은 역할을 하는 함수로, UART를 통해, 숫자, 문자, 변수의 내용을 출력할 때 사용한다. 출력되는 UART 채널은 set\_debug\_channel 함수를 통해 설정해 놓은 디버깅 채널을 사용한다. 기본으로 0 번 채널이 설정되어 있다. 참고로 lib\_config.h 파일에서 Tx interrupt를 사용하도록 설정하여도, 이 함수는 polling 방식으로 동작한다.

### Usage

```
debugprintf("result number : %dWrWn",result);
```

→UART를 통해 "result number :"라는 문자열과 result 변수 값을 10 진수로 출력한다.

그리고 줄 바꿈을 한다. printf 함수와 사용법이 같다. 단 줄 바꿈 시 WrWn을 모두 써주어야 한다.

## 5.12 debugstring

**void** debugstring(**const char\*** str)

문자열을 출력하는 함수로, 문자열만 출력하기를 원할 때 사용하는 함수이다. 출력되는 UART 채널은 set\_debug\_channel 함수를 통해 설정해 놓은 디버깅 채널을 사용한다. 기본으로 0 번 채널이 설정되어 있다. 참고로 debugprintf 함수를 사용하여도 문자열만 출력 가능하며, lib\_config.h 파일에서 Tx interrupt를 사용하도록 설정하여도, 이 함수는 polling 방식으로 동작한다.

### Usage

```
debugstring("==== adStar-L Start ===WrWn");
```

→""안의 문자열을 UART를 통해 출력한다.

## 5.13 PRINTLINE

매크로 함수로 PRINTLINE을 호출한 곳의 줄(line)값을 UART 통해 출력한다. 채널은 설정해 놓은 디버깅 채널을 사용한다. 출력되는 UART 채널은 set\_debug\_channel 함수를 통해 설정해 놓은 디버깅 채널을 사용한다. 기본으로 0 번 채널이 설정되어 있다.

### Usage

```
PRINTLINE;
```

→ 함수를 호출한 곳의 line 값을 출력한다..

## 5.14 PRINTVAR( A )

매크로 함수로 PRINTVAR 를 호출한 곳의 줄(line)값과 A 값을 UART 를 통해 출력한다. 출력되는 UART 채널은 set\_debug\_channel 함수를 통해 설정해 놓은 디버깅 채널을 사용한다. 기본으로 0 번 채널이 설정되어 있다.

### Usage

```
int a = 10;
PRINTVAR(a);;

→ 변수 a 의 값을 출력한다. 레지스터 값을 출력 할 수도 있다.
```

## 5.15 UART Example

```
#include "adStar-L.h"

Int main()
{
    boardinit();
    uart_config(1, 115200, DATABITS_8, STOPBITS_1, UART_PARNONE );
        // UART 1 번 채널을 다음과 같이 설정한다.
        // Baud rate = 115200
        // Data bit = 8bit
        // Stop bit = 1
        // Parity = none
    debugstring("=====WrWn");
    debugprintf("ADSTAR-L UART Example. System clock(%dMhz)WrWn",get_ahb_clock()/1000000);
    debugstring("=====WrWn");

    U8 ch;
    While(1)
    {
        if(uart_getch(1, &ch))      // UART 1 번에서 1Byte 데이터를 읽어 ch 에 저장한다.
        {
            uart_putch(1, ch); // UART 1 번에서 입력 받은 데이터를 UART 0 번으로 출력한다.
        }
    }
}
```

## < UART Interrupt >

adStar-L SDK library 는 lib\_config.h 파일의 정의에 따라 UART Interrupt 를 사용하도록 설정되어 있다.

Default 값은 UART Rx Interrupt 를 설정하도록 되어 있다.

```
148
149  */
150  */
151  */
152  */
153  */
154  */
155 #define UART_BUF_SIZE 512
156 #define CONFIG_UART_RX_INTERRUPT
157 // #define CONFIG_UART_TX_INTERRUPT
158
```

< lib\_config.h >

기본적으로 UART Rx Interrupt 를 사용하도록 설정 되어 있기 때문에, uart\_config( )함수를 호출하면 UART 초기화를 진행하고, Interrupt 를 설정한다. Interrupt 가 설정된 상태에서 UART Interrupt 가 발생하면 uart.c 의 UARTxISR()가 호출되어 Interrupt Routine 을 수행한다.  
(uart.c, uart.h 참조)

# 6. Interrupt

## 6.1 init\_interrupt

```
void init_interrupt(void);
```

interrupt 를 초기화 하는 함수이다. 인터럽트 관련 함수를 사용하기 위해서 반드시 호출해주어야 하는 함수로 adStar-L SDK 의 startup\_adStar-L.s 에서 호출하여 인터럽트를 초기화한다.

## 6.2 set\_interrupt

```
BOOL set_interrupt(INTERRUPT_TYPE intnum, void (*fp)());
```

해당 interrupt 가 발생하였을 때 호출할 함수를 등록하는 함수이다.  
Uart, Sound mixer 등의 Interrupt 함수는 SDK 에서 생성 및 설정 되어 있으므로 중복 등록하지 않도록 주의하기 바란다.

### Parameter

intnum	interrupt type (interrupt type 에 대한 정보는 다음 장 참고)
(*fp)()	interrupt 가 발생했을 때 호출할 함수

### Return value

TRUE or FALSE

## 6.3 enable\_interrupt

```
void enable_interrupt(INTERRUPT_TYPE intnum, BOOL b);
```

해당 interrupt 를 enable 하는 함수이다.set\_interrupt 함수로 interrupt 함수를 등록하고, enable\_interrupt 함수로 활성화 시킨다.

### Parameter

intnum	interrupt type (interrupt type 에 대한 정보는 다음 장 참고)
b	1 or TRUE => interrupt enable 0 or FALSE => interrupt disable

### Return value

TRUE or FALSE

**INTERRUPT\_TYPE**

INTERRUPT_TYPE		INTERRUPT_TYPE	
INTNUM_EIRQ0	External interrupt 0	INTNUM_PMC	PMC interrupt
INTNUM_EIRQ1	External interrupt 1	INTNUM_ADC	ADC interrupt
INTNUM_SYSTIMER	System timer interrupt	INTNUM_USB	USB interrupt
INTNUM_TIMER0	Timer interrupt 0	INTNUM_SPIO	SPI 0 interrupt
INTNUM_TIMER1	Timer interrupt 1	INTNUM_SPI1	SPI 1 interrupt
INTNUM_SOUND_MIXER	Sound mixer interrupt	INTNUM_JPEG0	JPEG0 interrupt
INTNUM_FRAME_SYNC	Frame sync interrupt	INTNUM_JPEG1	JPEG1 interrupt
INTNUM_DMA0	DMA interrupt 0	INTNUM_SDHC	SDHC interrupt
INTNUM_DMA1	DMA interrupt 1	INTNUM_NAND	NAND Flash interrupt
INTNUM_DMA2	DMA interrupt 2	INTNUM_WATCHDOG	Watch dog interrupt
INTNUM_DMA3	DMA interrupt 3	INTNUM_TWI	TWI interrupt
INTNUM_GPIOA	GPIO A interrupt	INTNUM_RTC	RTC interrupt
INTNUM_GPIOB	GPIO B interrupt	INTNUM_RTC_ALARM	RTC Alarm interrupt
INTNUM_GPIOC	GPIO C interrupt	INTNUM_CAPTURE_OVERFLOW	Capture overflow interrupt
INTNUM_GPIOD	GPIO D interrupt		
INTNUM_UART0	UART 0 interrupt	INTNUM_CLOCK_CTRL_R	Clock control interrupt
INTNUM_UART1	UART 1 interrupt		

## 6.4 Interrupt Example

```
#include "adStar-L.h"

void EIRQ0ISR( )
{
    debugprintf("EIRQ0 Interrupt\r\n");
}

int main()
{
    boardinit();
    uart_config(1, 115200, DATABITS_8, STOPBITS_1, UART_PARNONE );
    set_interrupt( INTNUM_EIRQ0, EIRQ0ISR );
        //External Interrupt 0 번이 발생하였을 때 호출 될 interrupt 함수를 등록한다.
    enable_interrupt( INTNUM_EIRQ0, TRUE );
        // External Interrupt 0 번을 enable 시킨다.

    while(1)
    return 0;
}
```

## 7. TIMER

### 7.1 set\_timer

**BOOL** set\_timer(**int** nCh, **U32** ms)

nCh 채널 timer 를 설정 및 동작시키는 함수이다. 사용할 timer 채널과 주기를 결정하고 timer control register 에서 timer 동작을 enable 한다. 그리고 마지막으로 timer interrupt 를 enable 한다.

set\_interrupt 함수로 timer interrupt 등록한 후 이 함수를 호출하면 설정한 주기만큼 timer interrupt 가 발생한다.

#### Parameter

<b>nCh</b>	설정할 timer 채널 값.
<b>ms</b>	timer interrupt 주기. (ms 단위)

#### Return Value

TRUE( 1 ) or FALSE ( 0 )

### 7.2 stop\_timer

**BOOL** stop\_timer(**int** nCh)

nCh 채널 timer 를 정지시키는 함수이다. nCh 채널의 timer 동작을 disable 시킨다.

#### Parameter

<b>nCh</b>	정지시킬 timer 채널 값.
------------	------------------

#### Return Value

TRUE( 1 ) or FALSE ( 0 )

### 7.3 delayms

**BOOL** delayms(**U32** ms)

ms 만큼 delay 가 걸린다. (단위는 ms 이다)

#### Parameter

<b>ms</b>	delay 걸리는 시간 단위는 ms.
-----------	----------------------

**Return Value**

TRUE( 1 ) or FALSE ( 0 )

**7.4 TIMER Example**

```
#include "adStar-L.h"

void TIMER0ISR( )
{
    debugprintf("==TIMER0ISR==\r\n");
}

int main()
{
    boardinit();
    uart_config(1, 115200, DATABITS_8, STOPBITS_1, UART_PARNONE );
    set_interrupt( INTNUM_TIMER0, TIMER0ISR );
        //Timer 0 번 Interrupt 가 발생하였을 때 호출 될 interrupt 함수를 등록한다.
    set_timer( 0, 1000);
        // 1 초마다 Timer0 Interrupt 가 발생한다.
    delayms(5000);
        // 5 초간 delay 가 발생.
    stop_timer( 0 );
        // Timer 0 번 Interrupt 를 disable 시킨다. 더 이상 Timer 가 발생하지 않음.
    while(1)
        return 0;
}
```

# 8. Graphic

## 8.1 setscreen

**void** setscreen(**SCREENRES** size, **U32** screenmode)

LCD 해상도와 RGB 모드를 설정하는 함수. LCD 출력을 위해 처음으로 설정해주어야 한다.  
설정한 해상도에 맞게 CRT Controller 를 설정한다.

### Parameter

size      LCD 의 해상도 값으로, 사용하려는 LCD 의 해상도로 설정을 해주면 해상도에  
              맞게 CRT Controller 를 설정한다. SCREENRES 는 다음과 같이 정의되어 있다.  
              (LCD 마다 특성이 다르기 때문에 해상도와 맞게 설정하였는데도 LCD 가  
              정상적으로 나오지 않을 경우에는 crt.c 의 setscreen( ) 함수에서 LCD 특성에 맞게  
              값을 수정해주어야 한다.)

```
Typedefenum {
    SCREEN_480x272 = 0,
    SCREEN_640x480,
    SCREEN_800x480,
} SCREENRES;
```

Screenmode    RGB 모드를 설정하는 값으로, 다음과 같이 정의 되어 있다.

**SCREENMODE\_RGB888**  
**SCREENMODE\_RGB565**

### Return value

없음

## 8.2 createframe

**SURFACE\*** createframe(**U32** w, **U32** h, **U32** bpp)

화면에 출력 될 frame(메모리 영역)을 생성하는 함수. 가로크기가 w 이고, 세로크기가 h 인 이미지  
또는 도형을 그릴 수 있는 영역을 생성한다. 화면 출력을 위해서 한 개 이상의 frame 이 필요하다.

### Parameter

w      생성될 frame 의 가로크기를 설정한다.

- h                    생성될 frame 의 세로 크기를 설정한다.  
 bpp                생성될 frame 의 bit per pixel 값을 설정한다.

**Return value**

frame 의 정보를 가지고 있는 SURFACE 구조체를 return 한다. SURFACE 는 다음과 같이 정의되어 있다.

```
typedefstruct {
    U32 w;
    U32 h;
    U32 bpp;
    void* pixels;
    U32 pixtype;
    PALETTE* pal;
    U8 ashiftbit;
    U8 rshiftbit;
    U8 gshiftbit;
    U8 bshiftbit;
    void* reserve;
} SURFACE;
```

**8.3 setframebuffer**

**void setframebuffer(SURFACE\* surf)**

화면에 보여지는 frame 을선택하는 함수. createframe()으로 생성한 frame 을 선택하면 그 frame 이 화면에 보이게 된다. ( single frame )

화면 출력을 위해서 반드시 setframebuffer(), setdoubleframebuffer() 중 한가지를 호출하여 frame 을 선택해야 한다.

**Parameter**

- surf                createframe() 에 의해 생성된 SURFACE 구조체.

**Return value**

없음.

## 8.4 setdoubleframebuffer

**void** setdoubleframebuffer(**SURFACE\*** surf, **SURFACE\*** surf2)

화면에 보여지는 frame 을 두 개 설정하는 함수. frame 을 두 개 설정한 후, getbackframe()함수와, flip()함수를 사용하여 두 개의 frame 을 번갈아 가며 화면에 보여지게 할 수 있다. ( double frame ) 두 개의 frame 은 편의상 현재 보이는 frame 을 front frame, 현재 보여지지 않고 있는 frame 을 back frame 이라고 한다.

(setdoubleframebuffer() 함수를 사용하기 위해서는 frame 을 두 개 생성해야 한다.)

화면 출력을 위해서 반드시 setframebuffer(), setdoubleframebuffer() 중 한가지를 호출하여 frame 을 선택해야 한다.

### Parameter

surf	createframe() 에 의해 생성된 SURFACE 구조체 화면 출력에 사용 될 첫 번째 frame 포인터이다.
surf2	createframe() 에 의해 생성된 SURFACE 구조체 화면 출력에 사용 될 두 번째 frame 포인터.

### Return value

없음.

## 8.5 setframebufferxy

**void** setframebufferxy(**SURFACE\*** surf, **U32** x, **U32** y);

화면에 보여지는 frame 의 시작 위치를 조정하여, frame 의 특정 지점부터 화면에 보여지도록 하는 함수이다. frame 크기가 화면보다 클 경우에 적용할 수 있으며, x 와 y 값은 frame 의 크기에서 화면 해상도의 차이 값을 Max 로 가질 수 있다.

(예: 800\*480 해상도, frame size : 1024\*512 면, Max x = 224, Max y = 32 )

setframebufferxy() 함수를 사용하기 전에 setframebuffer() 또는 setdoublebuffer()함수로 화면에 출력할 frame 선택을 반드시 해주어야 한다.

### Parameter

surf	createframe() 에 의해 생성된 SURFACE 구조체.
x	전체 frame 중 출력 할 부분의 시작 x 좌표.
y	전체 frame 중 출력 할 부분의 시작 y 좌표.

### Return value

없음.

## 8.6 set\_draw\_target

**Void** set\_draw\_target(**SURFACE\*** surf)

이미지를 출력하거나, 도형을 그릴 frame 을 선택하는 함수. createframe() 으로 생성된 frame 을 인자로 적어주면, draw 관련 함수를 호출했을 때 해당 frame 에 draw 하게 된다.

### Parameter

surf              frame 포인터. createframe()로 생성된 SURFACE 구조체

### Return value

없음.

## 8.7 get\_draw\_target

**SURFACE\*** get\_draw\_target( );

현재 draw target 으로 지정된 frame 의 포인터를 불러온다.

### Parameter

없음.

### Return value

frame 포인터. createframe()로 생성된 SURFACE 구조체

## 8.8 getbackframe

**SURFACE\*** getbackframe()

setdoubleframebuffer()함수를 사용하여 frame 두 개를 사용할 경우, 현재 화면에 보여지지 않고 있는 frame 의 포인터를 반환하는 함수이다. 주로 setdrawtarget()함수와 함께 사용하여 현재 보여지지 않고 있는 frame 에 draw 하는 경우에 사용한다.

### Parameter

없음.

### Return value

createframe()에 의해 생성된 SURFACE 구조체를 return 한다. frame 의 정보를 가지고 있다.

## 8.9 getfrontframe

**SURFACE\***getfrontframe()

setdoubleframebuffer()함수를 사용하여 frame 두 개를 사용할 경우, 현재 화면에 보여지고 있는 frame의 포인터를 반환하는 함수이다.

### Parameter

없음.

### Return value

createframe()에 의해 생성된 SURFACE 구조체를 return 한다. frame의 정보를 가지고 있다.

## 8.10 flip

**void** flip()

setdoubleframebuffer()함수를 사용하여 frame 두 개를 사용할 경우, frame 간에 전환을 할 때 사용하는 함수이다.

### Parameter

없음.

### Return value

없음.

## 8.11 getscreenwidth

**U32** getscreenwidth( )

현재 화면의 가로 크기 값을 반환하는 함수. setscreen()함수를 통해 설정 된 width 값이 반환된다.

### Parameter

없음.

### Return value

화면의 가로크기 값을 반환한다.

## 8.12 getscreenheight

**U32** getscreenheight( )

현재 화면의 세로 크기 값을 반환하는 함수. setscreen()함수를 통해 설정 된 height 값이 반환된다.

### Parameter

없음.

### Return value

화면의 세로크기 값을 반환한다.

## 8.13 getscreencpitch

**U32** getscreencpitch()

화면의 pitch 값을 반환하는 함수. Setscreen() 함수를 통해서 설정 된 pitch 값이 반환된다.  
pitch 값은 screen width × bpp ÷ 8 이다.

### Parameter

없음.

### Return value

화면의 pitch 값을 반환한다.

## 8.14 getscreenbpp

**U32** getscreenbpp()

화면의 bpp(bit per pixel)값을 반환하는 함수. setscreen() 함수를 통해 설정 된 bpp 값이 반환된다.

### Parameter

없음.

### Return value

화면의 bpp 값을 반환한다.

## 8.15 drawputpixel

```
void drawputpixel(int x, int y, U8 r, U8 g, U8 b);
```

한 점을 찍는 함수. 지정한 위치에 지정된 색의 점을 찍는다. RGB 값을 직접 주는 함수로 drawsetrgb( )함수의 영향을 받지 않는다.

### Parameter

x	점을 찍은 x 좌표
y	점을 찍은 y 좌표
r	점의 색상 중 r 값. ( 0 ~ 255 )
g	점의 색상 중 g 값. ( 0 ~ 255 )
b	점의 색상 중 b 값. ( 0 ~ 255 )

### Return value

없음.

## 8.16 draw\_line

```
void draw_line( int x1, int y1, int x2, int y2 , EGL_COLOR color)
```

x1, y1 위치에서 x2, y2 위치로 선을 그린다. 선의 색은 MAKE\_COLORREF( ) MACRO 에 의해 색을 설정하여 사용한다.

### Parameter

x1	선의 시작점의 x 좌표.
y1	선의 시작점의 y 좌표.
x2	선의 끝점의 x 좌표.
y2	선의 끝점의 y 좌표.
color	선의 color. MAKE_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

### Return Value

없음.

## 8.17 draw\_rect

```
void draw_rect(int x, int y, int w, int h, EGL_COLOR c)
```

x, y 좌표를 시작점으로 가로가 w, 세로가 h 인 사각형을 그린다.

**Parameter**

- x              사각형의 시작점의 x 좌표
- y              사각형의 시작점의 y 좌표
- w              사각형의 가로 길이.
- h              사각형의 세로 길이.
- c              사각형의 color. MAKE\_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

**Return value**

없음.

**8.18 draw\_rectfill**

**void draw\_rectfill(int x, int y, int w, int h, EGL\_COLOR c)**

x, y 좌표를 시작점으로 가로가 w, 세로가 h 인 사각형을 그린다. draw\_rect 와 다른 점은 안을 채운 사각형을 그린다.

**Parameter**

- x              사각형의 시작점의 x 좌표
- y              사각형의 시작점의 y 좌표
- w              사각형의 가로 길이.
- h              사각형의 세로 길이.
- c              사각형의 color. MAKE\_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

**Return value**

없음.

**8.19 draw\_roundrect**

**void draw\_roundrect(int x0, int y0, int w, int h, int corner, EGL\_COLOR c)**

x, y 좌표를 시작점으로 가로가 w, 세로가 h 인 모서리가 부드러운 사각형을 그린다. corner 값에 따라 모서리의 부드러운 형태가 조정된다.

**Parameter**

- x0              사각형의 시작점의 x 좌표
- y0              사각형의 시작점의 y 좌표
- w              사각형의 가로 길이.
- h              사각형의 세로 길이.

- corner 모서리의 부드러운 형태 조정 값. 이 값이 클수록 부드러운 부분이 넓어진다.  
 c 사각형의 color. MAKE\_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

**Return value**

없음.

**8.20 draw\_rectfill**

**void draw\_rectfill(int x0, int y0, int w, int h, int corner, EGL\_COLOR c)**

x, y 좌표를 시작점으로 가로가 w, 세로가 h 인 모서리가 부드러운 사각형을 그린다. corner 값에 따라 모서리의 부드러운 형태가 조정된다. draw\_rect 와 다르게 안을 채운 사각형을 그린다.

**Parameter**

- x0 사각형의 시작점의 x 좌표  
 y0 사각형의 시작점의 y 좌표  
 w 사각형의 가로 길이.  
 h 사각형의 세로 길이.  
 corner 모서리의 부드러운 형태 조정 값. 이 값이 클수록 부드러운 부분이 넓어진다.  
 c 사각형의 color. MAKE\_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

**Return value**

없음.

**8.21 draw\_circle**

**void draw\_circle(int x, int y, int r, EGL\_COLOR color)**

x, y 좌표를 원점으로 하고 반지름이 r 인 원을 그린다.

**Parameter**

- x 원점의 x 좌표  
 y 원점의 y 좌표  
 r 원의 반지름  
 color 원의 color. MAKE\_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

**Return value**

없음.

## 8.22 draw\_circlefill

```
void draw_circlefill(int x, int y, int r, EGL_COLOR c)
```

x, y 좌표를 원점으로 하고 반지름이 r 인 안을 채운 원을 그린다.

### Parameter

x	원점의 x 좌표
y	원점의 y 좌표
r	원의 반지름
c	원의 color. MAKE_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

### Return value

없음.

## 8.23 draw\_ellipse

```
void draw_ellipse (int x, int y, int rx, int ry, EGL_COLOR c );
```

x, y 좌표를 원점으로 하고 x 축 반지름이 rx이고 y 축 반지름이 ry인 타원을 그린다.

### Parameter

x	원점의 x 좌표
y	원점의 y 좌표
rx	타원의 x 축 반지름
ry	타원의 y 축 반지름
c	타원의 color. MAKE_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

### Return value

없음.

## 8.24 draw\_ellipselfill

```
void draw_ellipselfill (int x, int y, int rx, U16 ry, EGL_COLOR c);
```

x, y 좌표를 원점으로 하고 x 축 반지름이 rx이고 y 축 반지름이 ry인 안을 채운 타원을 그린다.

### Parameter

x	원점의 x 좌표
y	원점의 y 좌표

rx	타원의 x 축 반지름
ry	타원의 y 축 반지름
c	타원의 color. MAKE_COLORREF(r,g,b) 매크로를 사용하여 값을 설정한다.

**Return value**

없음.

**8.25 loadbmp**

**SURFACE\*** loadbmp(**char\*** fname);

Bmp 파일을 load 하는 함수. bmp 파일을 load 하여 메모리 영역을 생성 drawsurface( )함수를 사용하여 이미지를 출력할 수 있도록 SURFACE 구조체를 생성 후 저장을 한다.

**Parameter**

fname            bmp 파일 이름

**Return value**

이미지 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

**8.26 loadbmpp**

**SURFACE\*** loadbmpp(**U8\*** startaddr)

BMP 이미지를 메모리에서 load 하는 함수. Loadbmp 와 다른 점은 파일을 load 하는 것이 아니고, 메모리에서 load 한다는 점이다.

**Parameter**

startaddr        이미지가 저장되어 있는 메모리 주소.

**Return value**

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

**8.27 loadjpg**

**SURFACE\*** loadjpg(**char\*** fname);

jpg 파일을 load 하는 함수. jpg 파일을 load 하여 메모리 영역을 생성 drawsurface( )함수를 사용하여 이미지를 출력할 수 있도록 SURFACE 구조체를 생성 후 저장을 한다.

( 참고로 jpg image 출력을 위해서는 libjpeg.a 파일을 project에 추가해주어야 한다.)

#### Parameter

fname            jpg 파일 이름

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.28 loadjpg

**SURFACE\*** loadjpg(**U8\*** databuf, **U32** len)

JPG 이미지를 메모리에서 load하는 함수. Loadjpg 와 다른 점은 파일을 load하는 것이 아니고, 메모리에서 load 한다는 점이다.

#### Parameter

databuf            이미지가 저장되어 있는 메모리 주소.  
len                이미지 데이터의 길이

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.29 loadtga

**SURFACE\*** loadtga(**char\*** fname)

tga 파일을 load하는 함수. tga 파일을 load하여 메모리 영역을 생성 drawsurface( )함수를 사용하여 이미지를 출력할 수 있도록 SURFACE 구조체를 생성 후 저장을 한다.

#### Parameter

fname            tga 파일 이름

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.30 loadtgap

**SURFACE\*** loadtgap(**U8\*** startaddr)

tga 이미지를 메모리에서 load 하는 함수. loadtga 와 다른 점은 파일을 load 하는 것이 아니고, 메모리에서 load 한다는 점이다.

#### Parameter

Startaddr 이미지가 저장되어 있는 메모리 주소.

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.31 loadpng

**SURFACE\*** loadpng(**char\*** filename)

png 파일을 load 하는 함수. png 파일을 load 하여 메모리 영역을 생성 drawsurface( )함수를 사용하여 이미지를 출력할 수 있도록 SURFACE 구조체를 생성 후 저장을 한다.

( 참고로 png image 출력을 위해서는 libpng.a 와 libz.a 파일을 project 에 추가해주어야 한다.)

#### Parameter

filename png 파일 이름

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.32 loadpngp

**SURFACE\*** loadpngp(**U8\*** pngbuf, **U32** datalen)

png 이미지를 메모리에서 load 하는 함수. loadpng 와 다른 점은 파일을 load 하는 것이 아니고, 메모리에서 load 한다는 점이다.

#### Parameter

Pngbuf 이미지가 저장되어 있는 메모리 주소  
Datalen 이미지 데이터 길이

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.33 loadsurf

**SURFACE\* loadsurf(char\* fname)**

Suf 이미지 파일을 load 하는 함수. suf 파일을 load 하여 메모리 영역을 생성 drawsurface( ) 함수를 사용하여 이미지를 출력할 수 있도록 SURFACE 구조체를 생성 후 저장한다.

Suf 파일은 adStar-L RGB 모드에 맞게 변환한 파일로 load 시간이 지원하는 이미지 중 가장 적게 소요된다. 사용방법은 다른 이미지 load 함수와 동일하며, 이미지 변환은 adStar-L SDK pc-util 의 MakeSurface2.exe 프로그램을 사용하면 된다.

#### Parameter

filename                  surf 파일 이름

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.34 loadjpg\_hw

**SURFACE\*loadjpg\_hw(char\* fname)**

JPEG H/W decoder 를 사용하여 JPG 이미지 파일을 load 한다. S/W 로 load 하는 것보다 decoding 속도가 빠르고, H/W decoder 는 standard format 만 지원하기 때문에, pc-util 의 makesurface2.exe 파일을 사용하여 standard format jpg 이미지로 변환해 주어야 한다.

#### Parameter

fname                  JPG 파일 이름

#### Return value

이미지가 정보와 이미지 데이터가 저장된 메모리 영역의 포인터.

### 8.35 draw\_surface

**BOOL draw\_surface(SURFACE\* src\_surf, int dx, int dy)**

이미지를 출력할 때 사용하는 함수.

#### Parameter

src_surf	load 함수에서 return 받은 이미지 데이터가 저장된 메모리 영역 포인터.
dx	이미지가 위치할 x 좌표.
dy	이미지가 위치할 y 좌표.

**Return value**

TRUE or FALSE

**8.36 draw\_surface\_rect****BOOL draw\_surface\_rect(SURFACE\* src\_surf, int dx, int dy, int sx, int sy, int w, int h)**

이미지 중 지정된 영역만 출력할 때 사용하는 함수. 이미지를 지정한 위치부터 지정한 사이즈만큼 dx, dy 에 출력한다. 이미지의 일부분만 출력할 때 사용하면 된다.

**Parameter**

src_surf	load 함수에서 return 받은 이미지 데이터가 저장된 메모리 영역 포인터.
dx	이미지가 위치할 x 좌표.
dy	이미지가 위치할 y 좌표.
sx	이미지 중 출력할 부분의 시작 x 좌표.
sy	이미지 중 출력할 부분의 시작 y 좌표.
w	출력할 이미지의 가로 길이. 시작 sx 값부터 w 만큼 출력한다.
h	출력할 이미지의 세로 길이. 시작 sy 값부터 h 만큼 출력한다.

**Return value**

TRUE or FALSE

**8.37 draw\_set\_clip\_window****void draw\_set\_clip\_window(SURFACE\* dst, CLIP\_RECT\* pRect)**

이미지가 출력될 수 있는 영역을 제한한다. pRect 안의 영역에만 이미지가 출력된다.

**Parameter**

dst	대상 frame.
pRect	이미지 출력을 허용할 영역.
	typedef struct
	{
	int x;
	int y;
	int endx;
	int endy;
	}
	CLIP_RECT;

**Return value**

없음.

**8.38 draw\_surface\_scale**

**BOOL** draw\_surface\_scale(**SURFACE\*** src\_surf, **int** dx, **int** dy, **int** dw, **int** dh)

이미지를 확대 또는 축소해서 출력한다. 원본 이미지의 가로 세로 길이보다 dw, dh 값이 크면, 이미지가 확대 돼서 출력되고, dw, dh 값이 작으면, 축소 돼서 출력된다.

**Parameter**

src_surf	load 함수에서 return 받은 이미지 데이터가 저장된 메모리 영역 포인터.
dx	이미지가 출력 될 x 좌표.
dy	이미지가 출력 될 y 좌표.
dw	출력될 이미지의 가로 길이, 원본 이미지 가로길이보다 이 값이 크면 가로길이가 확대되어 출력되고, 작으면 가로길이가 축소되어 출력된다.
dh	출력될 이미지의 세로 길이, 원본 이미지 세로길이보다 이 값이 크면 세로길이가 확대되고, 작으면 세로길이가 축소되어 출력된다.

**Return value**

TRUE or FALSE

**8.39 draw\_surface\_scalerect**

**BOOL** draw\_surface\_scalerect(**SURFACE\*** src\_surf, **int** dx, **int** dy, **int** dw, **int** dh,  
**int** sx, **int** sy, **int** sw, **int** sh)

drawsurface() 함수와 drawrect()함수를 합쳐 놓은 함수이다. 이미지의 특정 부분을 확대 또는 축소하여 출력한다. 이미지의 (sx,sy)부터 sw,sh 영역의 이미지를 dx,dy 좌표에 dw, dh 크기로 출력한다. sw,sh 값보다 dw, dh 값이 크면 확대, 작으면 축소이다.

**Parameter**

src_surf	load 함수에서 return 받은 이미지 데이터가 저장된 메모리 영역 포인터.
dx	이미지가 출력될 x 좌표
dy	이미지가 출력 될 y 좌표
dw	출력될 이미지의 가로 길이, 출력할 부분 이미지의 가로길이보다 크면 가로로 확대되고, 작으면 가로로 축소된다.
dh	출력될 이미지의 세로 길이, 출력할 부분 이미지의 세로길이보다 크면 세로로 확대되고, 작으면 세로로 축소된다.

sx	원본 이미지 중 출력할 부분의 시작 x 좌표.
sy	원본 이미지 중 출력할 부분의 시작 y 좌표
sw	출력할 이미지의 가로 길이. 시작 sx 값부터 sw 만큼 출력한다
sh	출력할 이미지의 세로 길이. 시작 sy 값부터 sh 만큼 출력한다

**Return value**

TRUE or FALSE

**8.40 release\_surface****void release\_surface(SURFACE\* surf);**

이미지 load 함수에 의해 생성된 메모리 영역을 해제한다. 메모리가 제한적이기 때문에 사용하지 않는 이미지에 대해서 release\_surface( )함수로 메모리 해제해주는 것이 좋다.

**Parameter**

surf            load 함수에서 return 받은 이미지 데이터가 저장된 메모리 영역 포인터..

**Return value**

없음.

**8.41 savebmp****BOOL savebmp(char\* savefname, SURFACE\* src);**

SURFACE 의 내용을 BMP 이미지로 저장한다. 현재화면을 저장하거나, 특정 SURFACE 를 이미지로 저장할 때 사용한다.

**Parameter**

savefname	저장하려는 BMP 파일 이름
src	저장하려는 화면의 SURFACE

**Return value**

TRUE or FALSE

## 8.42 savejpg

**BOOL** savejpg(**char\*** savefname, **SURFACE\*** src);

SURFACE 의 내용을 JPG 이미지로 저장한다. 현재화면을 이미지 파일로 저장하거나, 특정 SURFACE 를 이미지로 저장할 때 사용한다.

### Parameter

savefname	저장하려는 JPG 파일 이름
Src	저장하려는 화면의 SURFACE

### Return value

TRUE or FALSE

<savebmp/savejpg example>

```
SURFACE* current_frame = getfrontframe();
res = savebmp("image1.bmp",current_frame); // 현재 보여지는 화면을 image1.bmp 파일로 저장.
res = savejpg("image2.jpg",current_frame); // 현재 보여지는 화면을 image2.jpg 파일로 저장.
```

## 8.43 createsurface\_from

**SURFACE\*** createsurface\_from(**SURFACE\*** src, **U32** option);

이미지의 복사본을 만든다. option 을 통해 좌/우 90 도 회전, 180 도 회전, 좌우/상하 뒤집은 복사본을 만들 수 있다. 이미지를 회전하여 출력할 경우 이미지를 load 한 후에 createsurface\_from 함수를 사용하여 회전한 복사본을 만들어 출력하면 된다. 회전과 관련된 option 은 다음과 같으며, 원본 그대로의 복사본을 원하면, option 값으로 0 을 적어주면 된다. 복사본이 생성 되는 것으로 원본이미지는 그대로 사용할 수 있다.

```
#define PIVOT_RIGHT (1)
#define PIVOT_LEFT (1<<1)
#define PIVOT_180 (1<<2)
#define PIVOT_VFLIP (1<<4)
#define PIVOT_HFLIP (1<<5)
```

### Parameter

src	복사할 surface 의 원본 source.
option	복사본을 만들면서 적용할 option.

**Return value**

복사본 이미지 정보와 데이터가 저장된 메모리 영역의 포인터.

**Example**

```
SURFACE* surface = loadbmp("test.bmp");
SURFACE* surface_right =createsurface_from(surface, PIVOT_RIGHT)
    // 원본이미지를 오른쪽 90 도 회전한 복사본 이미지를 만든다.
SURFACE* surface_left =createsurface_from(surface, PIVOT_LEFT)
    // 원본이미지를 왼쪽 90 도 회전한 복사본 이미지를 만든다.
SURFACE* surface_180 =createsurface_from(surface, PIVOT_180)
    // 원본이미지를 180 도 회전한 복사본 이미지를 만든다.
SURFACE* surface_vflip =createsurface_from(surface, PIVOT_VFLIP)
    // 원본이미지를 상하 뒤집은 복사본 이미지를 만든다.
SURFACE* surface_hflip =createsurface_from(surface, PIVOT_HFLIP)
    // 원본이미지를 좌우 뒤집은 복사본 이미지를 만든다.

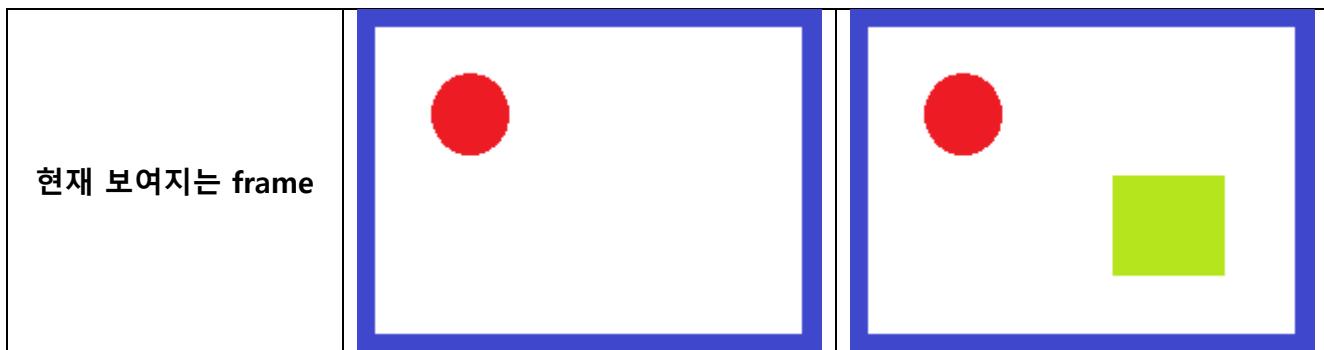
drawsurface(surface,0,0);
drawsurface(surface_right,200,0);
drawsurface(surface_left,0,200);
```

## 8.44 single frame & double frame 사용 예

### < SINGLE FRAME >

Single frame 은 화면에 보여지는 frame 을 하나만 생성한다.

그래서 현재 보여지는 화면에 이미지 및 도형을 그리는 동작을 수행한다.



위 그림처럼 frame 이 하나만 생성이 되고, 이미지 및 도형은 하나의 frame 에 계속 그려지게 된다.

Single frame 설정 및 사용법은 다음과 같다. ( 위 그림처럼 동작하는 코드 )

```
setscreen(SCREEN_800x480, SCREENMODE_RGB565 )
SURFACE* frame = createframe(800, 480, 16);
setframebuffer(frame);
setdrawtarget(frame);

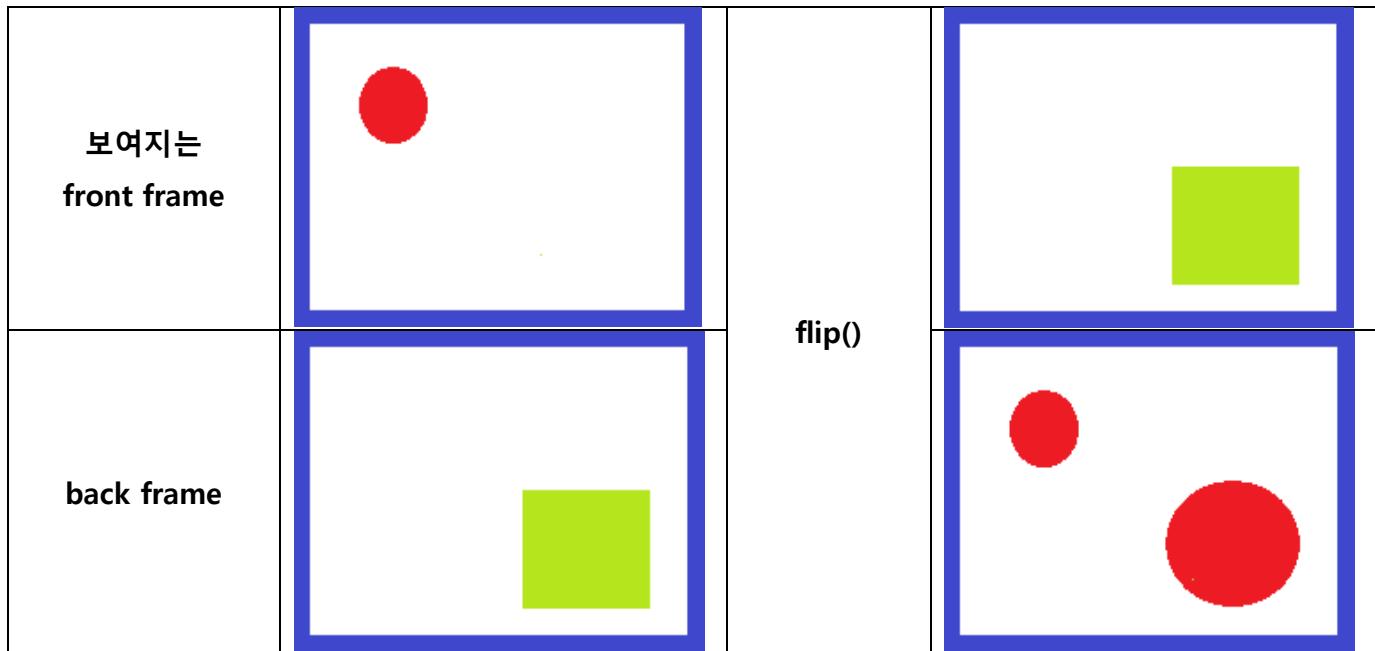
draw_circlefill(50,50,10,MAKE_COLORREF(255,0,0));
draw_rect(100,100,100,100,MAKE_COLORREF(0,255,0));
```

frame 하나를 생성하고, setframebuffer() 함수와 setdrawtarget() 함수로 생성한 하나의 frame 을 설정한 후, 이미지 또는 도형을 그리면 된다.

## <DOUBLE FRAME >

Double frame 은 frame 을 두 개 생성한다. 그래서 한 개의 frame 은 화면에 보여지고, 다른 한 개의 frame 은 화면에 보여지지 않는다. 편의상 화면에 보여지는 frame 을 front frame 이라 하고, 화면에 보이지 않는 frame 을 back frame 라 하겠다.

Double frame 을 사용하게 되면 back frame 에 이미지 및 도형을 그려서 현재 보여지는 front frame 에 영향을 주지 않고, 이미지 및 도형을 그릴 수 있다.



위 그림에서 보듯이 back frame 에 도형을 그려도 front frame 에는 영향을 미치지 않아, 화면에 변화를 주지 않는다. Back frame 에 도형을 그린 후 flip()함수를 호출하면 frame 간에 전환이 이루어져 back frame 이 front frame 이 되면서 화면에 보여지게 되고, front frame 이 back frame 이 되면서 화면에 보이지 않게 된다.

double frame 의 사용법은 다음과 같다. (위 그림 처럼 동작하는 코드)

```

setscreen(SCREEN_800x480, SCREENMODE_RGB565 )
SURFACE* frame1 = createframe(800, 480, 16);
SURFACE* frame2 = createframe(800, 480, 16);

setdoubleframebuffer(frame1, frame2);

set_draw_target(getbackframe());
draw_circlefill(50,50,10,MAKE_COLORREF(255,0,0));
flip();

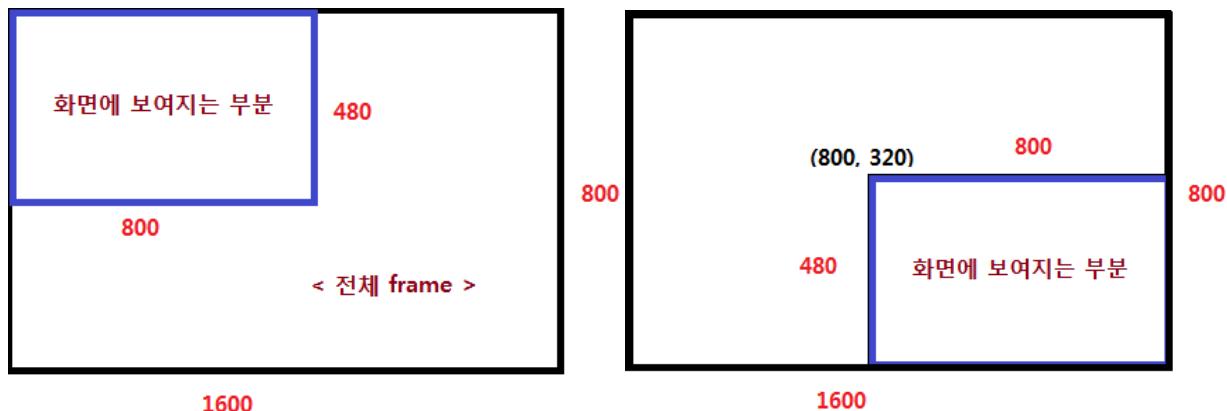
```

```
set_draw_target(getbackframe());
draw_rectfill(600,300,100,100,MAKE_COLORREF(0,255,0));
flip();
```

```
set_draw_target(getbackframe());
draw_circlefill(600,300,20,MAKE_COLORREF(255,0,0));
```

frame 을 두 개를 생성하고, setdrawtarget() 함수를 사용하여 backframe 을 draw target 으로 설정 한 후 이미지 및 도형을 그린다. 다 그린 후에 flip()함수를 호출하여 back frame 을 front frame 으로 만들어 화면에 보이게 한다. Front frame 은 back frame 이 되고, setdrawtarget() 함수로 draw target 이 되어 이미지 및 도형이 그려지게 된다.

## 8.45 set frame buffer xy 사용 예



setframebufferxy 함수를 사용하여, 화면 크기보다 큰 frame 을 생성하였을 때 위 그림처럼 특정 위치만 화면에 보여지게 할 수 있다.

```

setscreen(SCREEN_800x480, SCREENMODE_RGB565);
SURFACE *frame = createframe(1600,800,16);
drawsetclipwindow(0,0,1600,800) // 표시할 화면 사이즈 변경
setframebuffer(frame);
set_draw_target(frame);

.....
.....
setframebufferxy(frame,800,320); // 위의 두 번째 그림처럼 전체 frame 중 800,320 부터
                                화면에 보여지게 된다.

```

위 코드에서처럼 frame 의 크기를 화면 해상도보다 크게 설정하고, setframebuffer( )함수로 화면 해상도보다 큰 frame 을 선택한다. 그런 후에 setframebufferxt()함수로 화면에 보여지는 부분의 시작점을 설정하면 frame 의 그 지점부터 화면에 보여지게 된다.

drawsetclipwindow(0,0,1600,800) 함수를 호출하여 표시할 화면 사이즈를 frame 사이즈에 맞게 변경해주어야 한다. 기본 값은 setscreen( )함수를 호출 할 때, 화면 해상도 사이즈 값으로 설정된다.

## 8.46 Graphic Example

### < SINGLE FRAME >

```
#include "adStar-L.h"

int main()
{
    boardinit();
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);
    debugstring("=====Wr\n");
    debugprintf("ADSTAR-L Graphic.System Clock(%dMhz)\r\n",get_ahb_clock()/1000000);
    debugstring("=====Wr\r\n");

    crtcclock_init(); // CRT clock enable.
    setscreen(SCREEN_800x480,SCREENMODE_RGB565); // 해상도 및 draw mode 를 설정한다.
    SURFACE *frame = createframe(800,480,16); // frame 을 생성한다.
    setframebuffer(frame); // 화면에 보여질 frame 을 선택한다.
    setfrawtarget(frame); // 이미지 및 도형이그려질 frame 을 선택한다.

    lcdon();
    draw_rectfill(0,0,getscreentwidth(),getscreentheight(),MAKE_COLORREF(255,255,255));
        // 0,0 부터 800,480 크기의 흰색 채운 사각형을 기린다. 화면 clear.
    draw_line(30,30,100,100,MAKE_COLORREF(255,0,0)); // 30,30 에서 100,100 까지 선을 그린다.
    draw_rect(10,10,100,100, MAKE_COLORREF(255,0,0));
        // 10,10 부터 가로 세로 크기가 100 인 사각형을 그린다.
    draw_circle(100,100,10, MAKE_COLORREF(255,0,0));
        // 원점이 100,100 이고 반지름이 10 인 원을 그린다.
    draw_circlefill(200,200,20, MAKE_COLORREF(255,0,0));
        // 원점이 200,200 이고 반지름이 20 인 채운 원을 그린다.
    draw_roundrect(50,50,100,100,5, MAKE_COLORREF(255,0,0));
        // 50,50 부터 가로 세로 크기가 100 인 모서리가 부드러운
        // 사각형을 그린다.
    draw_ellipse (300, 100, 20, 40, MAKE_COLORREF(255,0,0));
        // 원점이 300,100 이고, x 축 반지름이 20, y 축 반지름이 40 인 타원을 그린다.

    while(1);
    return 0;
}
```

## &lt; DOUBLE FRAME &gt;

```
#include "adStar-L.h"

int main()
{
    boardinit();
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);
    debugstring("=====Wr\n");
    debugprintf("ADSTAR-L Graphic.System Clock(%dMhz)\r\n",GetAHBclock()/1000000);
    debugstring("=====Wr\n");

    FATFS fs;
    F_mount(DRIVE_NAND, &fs);
    crtc_clock_init(); // CRT clock enable.

    setscreen(SCREEN_800x480,SCREENMODE_RGB565); // 해상도 및 draw mode 를 설정한다.

    SURFACE *frame1 = createframe(800,480,16); // frame1 을 생성한다.
    SURFACE *frame2 = createframe(800,480,16); // frame2 을 생성한다.

    //double frame 사용을 위해 frame 을 2 개 생성한다.

    setdoubleframebuffer(frame1,frame2); //double frame 으로 설정을 한다..

    lcdon();
    set_draw_target(getbackframe()); // 이미지 및 도형이 그려질 frame 을
                                    // back frame 으로 선택한다.

    SURFACE* image1 = loadbmp("test1.bmp"); // Nand 에서 test1.bmp 파일을 load 한다.
    draw_surface(image1,0,0); // bmp 이미지를 0,0 에 출력한다. (back frame)
    flip(); // frame 을 전환하여 back frame 을 화면에 보여준다.

    set_draw_target(getbackframe()); // 이미지 및 도형이 그려질 frame 을
                                    // back frame 으로 선택한다.

    SURFACE* image2 = loadjpg("test2.jpg"); // Nand 에서 test2.jpg 파일을 load 한다.
    draw_surface_rect(image2,0,0,100,100,200,200); // jpg 이미지를 0,0 에 출력하는데,
                                                // jpg 이미지 중 100,100 부터 가로 세로 200 사이즈만큼 출력한다.

    flip(); // frame 을 전환하여 back frame 을 화면에 보여준다. (jpg image).
    release_surface(image1); // 사용하지 않는 image1 을 메모리에서 제거한다.

    while(1);

    return 0;
}
```

## 9. Movie file play

Movie 파일은 SDK 의 pc-util 폴더 내에 있는 MovieGenerator™을 통해서 생성할 수 있다.  
MovieGenerator™에 대한 상세한 내용은 별도의 문서가 있으므로 본 장에서는 다루지 않는다.

### 9.1 loadmpvie

**MOVIE\*** loadmovie(**char\*** fname, **BOOL** bAll )

Movie 파일을 저장장치에서 load 하여 필요한 정보를 얻어온다.

이때 audio data 는 전체가 주 메모리에 저장된다. 따라서 audio data 가 있는 경우 충분한  
메모리가 확보되어야 한다.

11025Hz \* 2 (16bit mono) \* 30 sec = 661500 bytes

Loadmovie 에 의해서 사용된 메모리는 release\_movie 를 이용하여 메모리 해제할 수 있다.

#### Parameter

fname      Movie 파일 이름.

bAll      Video data 도 주 메모리에 로딩할 것인지 여부.

false 인 경우 video 를 매 frame 마다 저장 장치에서 읽어 오게 된다.

#### Return value

Pointer to Movie.

### 9.2 release\_movie

**void** release\_movie(**MOVIE\*** mov);

loadmovie 함수에서 생성된 메모리를 해제한다.

#### Parameter

mov      Pointer to MOVIE.

#### Return value

없음.

### 9.3 movie\_play

**BOOL** movie\_play(**MOVIE\*** mov, **int** x, **int** y, **BOOL** bAudio);

Video 의 첫 번째 frame 을 해당 영역에 출력한다. bAudio 가 true 이고 movie 파일에 audio 가 존재할 경우 audio play 를 시작한다.

#### Parameter

mov	Pointer to MOVIE.
x	x 좌표.
y	y 좌표.
bAudio	audio 출력 여부, false 일 경우 audio 를 출력하지 않는다.

#### Return value

True or False.

### 9.4 movie\_drawnext

**MOVIE\_DRAW\_RESULT** movie\_drawnext(**MOVIE\*** mov, **int** x, **int** y, **BOOL** b\_audiosync);

Mov 의 다음 video frame 을 특정 좌표에 출력한다. b\_audiosync 가 disable 될 경우 video 를 audio 와 동기화 하지 않고 차례대로 출력하게 된다.

#### Parameter

mov	Pointer to MOVIE
x	x 좌표
y	y 좌표
b_audiosync	출력할 video frame 과 audio 에 동기화 여부. True 일 경우 현재 출력 중인 audio 에 동기화를 맞춘 video 를 출력한다.

#### Return value

**MOVIE\_DRAW\_RESULT\_FAIL** – if mov is null, or mov is not running

**MOVIE\_DRAW\_RESULT\_OK** – if success.

**MOVIE\_DRAW\_RESULT\_EOF** – if end of file.

## 9.5 movie\_drawnextex

**MOVIE\_DRAW\_RESULT** movie\_drawnextex(**MOVIE\*** mov, **int** x, **int** y, **int** w, **int** h,  
**BOOL** b\_audiosync);

Mov 의 다음 video frame 을 특정 좌표에 설정한 크기로 출력한다. b\_audiosync 가 disable 될 경우 video 를 audio 와 동기화 하지 않고 차례대로 출력하게 된다.

### Parameter

mov	Pointer to MOVIE.
x	x 좌표.
y	y 좌표.
w	Video frame 의 가로크기 (video 크기보다 작을 경우 축소, 클 경우 확대)
h	Video frame 의 세로크기 (video 크기보다 작을 경우 축소, 클 경우 확대)
b_audiosync	출력할 video frame 과 audio 에 동기화 여부. True 일 경우 현재 출력 중인 audio 에 동기화를 맞춘 video 를 출력한다.

### Return value

MOVIE\_DRAW\_RESULT\_FAIL – if mov is null, or mov is not running

MOVIE\_DRAW\_RESULT\_OK – if success.

MOVIE\_DRAW\_RESULT\_EOF – if end of file.

## 9.6 movie\_seek

**U32** movie\_seek(**MOVIE\*** mov, **U32** ms);

Mov 파일의 현재 위치를 특정 시간으로 변경한다.

### Parameter

mov	Pointer to MOVIE
ms	Milliseconds

### Return value

Number of video frame

## 9.7 movie\_current\_time

**U32** movie\_current\_time(**MOVIE\*** mov);

현재 출력중인 frame 의 시간을 반환한다.

**Parameter**

mov      Pointer to Movie.

**Return value**

Current milliseconds.

**9.8 Example**

```
int main()
{
    ...
    ...
    MOVIE* mov = loadmovie(fname, FALSE);
    while(1)
    {
        If(movie_drawnext(mov, x, y, true) == MOVIE_DRAW_RESULT_EOF)
            Break;
        Flip();
    }
    ...
    ...
}
```

# 10. Sound

이번 장에서는 adStar-L sound 출력에 대해 알아보겠다.

adStar-L 는 8bit/16bit, signed/unsigned, mono/stereo WAV 와 MP3 를 지원한다.

## 10.1 sount\_init()

**BOOL** sount\_init()

Sound 출력을 위해 Soundmixer 를 초기화 하는 함수.

### Parameter

없음.

### Return value

없음.

## 10.2 sound\_loadwav

**WAVE\*** sound\_loadwav(**char\*** filename);

WAV 파일을 load 하는 함수. WAV 파일을 load 하여 WAVE 구조체를 생성 메모리에 저장한다.

### Parameter

filename           WAV 파일

### Return value

WAV 파일의 정보와 데이터가 저장된 WAVE 구조체.

## 10.3 sound\_loadwavp

**WAVE\*** sound\_loadwavp(**U8\*** buf, **U32** len);

WAV data 를 메모리에서 load 하는 함수.

### Parameter

buf               WAV data 가 저장되어 있는 buffer.

len               WAV data 의 길이.

**Return value**

WAV 파일의 정보와 데이터가 저장된 WAVE 구조체.

**10.4 sound\_loadmp3**

**WAVE\*** sound\_loadmp3(**char\*** filename);

MP3 파일을 load 하는 함수. MP3 파일을 load 하여 WAVE 구조체를 생성 메모리에 저장한다.

**Parameter**

filename            MP3 파일

**Return value**

MP3 파일의 정보와 데이터가 저장된 WAVE 구조체.

**10.5 sound\_loadmp3p**

**WAVE\*** sound\_loadmp3p(**void\*** buf, **U32** mp3databufsize);

MP3 data 를 메모리에서 load 하는 함수.

**Parameter**

buf                MP3 data 가 저장되어 있는 buffer.

Mp3databufsize    mp3 data 의 길이.

**Return value**

MP3 파일의 정보와 데이터가 저장된 WAVE 구조체.

**10.6 sound\_release**

**BOOL** sound\_release(**WAVE\*** pWave);

load\_soundwav() 또는 load\_soundmp3() 함수에 의해 생성된 메모리를 해제한다. 사용할 수 있는 메모리가 제한적이기 때문에 사용하지 않는 sound 대해 sound\_release( )함수를 호출하여 사용하지 않는 메모리를 반환하는 것이 좋다.

**Parameter**

pWavesound\_loadwav( )함수 또는 sound\_loadmp3( )함수에 의해 생성된 WAVE 구조체.

**Return value**

TRUE(1) or FALSE(0)

**10.7 sound\_play**

**BOOL** sound\_play(**WAVE\*** pWave);

load\_soundwav() 또는 load\_soundmp3() 함수로 load 한 sound 를 play 한다.

**Parameter**

pWavesound\_loadwav( )함수 또는 sound\_loadmp3( )함수에 의해 생성된 WAVE 구조체.

**Return value**

TRUE(1) or FALSE(0)

→adStar-L 는 sound 출력을 4 개채널 중 하나를 선택해서 할 수 있다. 0 번 1 번 채널은 I2S 를 통한 출력이고, 2 번 3 번 채널은 digital modulator 를 통한 출력이다. SDK 에서는 2 번 채널을 default 출력으로 정하고 있는데, 이를 변경하기 위해서는 lib\_src/adStar-L.epx 파일 열어 libconfig.h 파일의 "#define SND\_OUTPUT\_CHANNEL 2"에서 2 가 채널을 나타내는 값으로 이 값을 수정한 후 rebuild 하면 다른 채널로 출력이 가능하다.

**10.8 sound\_stop**

**BOOL** sound\_stop(**WAVE\*** pWave)

play 중인 sound 를 stop 시킨다.

**Parameter**

pWave                          현재 play 중인 sound 의 WAVE 구조체.

**Return value**

TRUE(1) or FALSE(0)

**10.9 sound\_vol**

**void** sound\_vol(**U8** vol);

sound volume 을 조정한다.

**Parameter**

vol            volume 값. 범위는 0 부터 255 이다.

**Return value**

없음

**10.10 sound\_vol\_wav**

**void** sound\_vol\_wav(**WAVE\*** pWav, **U8** vol);

음원 파일 각각의 volume 을 조정한다. sound\_vol() 함수와 별도로 volume 을 설정한다. 두 개의 음원을 동시에 출력할 때 음원마다 volume 크기를 다르게 하고 싶을 경우에 사용하면 된다.

**Parameter**

pWave        volume 을 조정할 sound 구조체  
vol            volume 값. 범위는 0 부터 255 이다.

**10.11 sound\_pause**

**BOOL** sound\_pause(**WAVE\*** pWave);

Play 중인 sound 를 pause 시킨다.

**Parameter**

pWave        현재 play 중인 sound 의 WAVE 구조체.

**Return value**

TRUE(1) or FALSE(0)

**10.12 sound\_resume**

**BOOL** sound\_resume(**WAVE\*** pWave);

Pause 상태의 sound 를 resume 시킨다.

**Parameter**

pWave        현재 pause 중인 sound 의 WAVE 구조체.

**Return value**

TRUE(1) or FALSE(0)

**10.13 sound\_isplay**

**BOOL** sound\_isplay(**WAVE\*** pWave);

현재 pWave sound 가 play 중인지 확인할 수 있다.

**Parameter**

pWave                          현재 play 중인지 확인하려는 sound 의 WAVE 구조체.

**Return value**

TRUE(1 = play) or FALSE(0 = stop)

**10.14 sound\_ispause**

**BOOL** sound\_ispause(**WAVE\*** pWave);

현재 pWave sound 가 pause 중인지 확인할 수 있다.

**Parameter**

pWave                          현재 pause 중인지 확인하려는 sound 의 WAVE 구조체.

**Return value**

TRUE(1 = pause) or FALSE(0 = none pause)

## 10.15 Sound Example

### < WAV File Play >

```
#include "adStar-L.h"

Intmain()
{
    boardinit();
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);
    debugstring("=====WrWn");
    debugprintf("ADSTAR-L Wave Play.System Clock(%dMhz)WrWn",get_ahb_clock()/1000000);
    debugstring("=====WrWn");

    FATFS fs;
    f_mount(DRIVE_NAND, &fs);

    sound_init();           // Sound 출력을 위한 sound mixer 초기화
    WAVE* sound = sound_loadwav("test.wav"); // wav 파일 load

    sound_play(sound);     // wav 파일 재생

    delayms(1000);

    if(soundisplay(sound)) // 1초후에 wav 파일 재생 중인지 확인
    {
        sound_stop(sound); // 재생 중이면 재생 중지
    }

    sound_release(sound); // 더 이상 사용하지 않을 sound 라 메모리에서 제거.

    ...
    ...

    while(1);
    return 0;
}
```

## &lt; MP3 File Play &gt;

```
#include "adStar-L.h"

Intmain()
{
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);
    debugstring("=====Wr\n");
    debugprintf("ADSTAR-L MP3Play.System Clock(%dMhz)\r\n",get_ahb_clock()/1000000);
    debugstring("=====Wr\r\n");

    FATFS fs;
    f_mount(DRIVE_NAND, &fs);

    sound_init();           // Sound 출력을 위한 sound mixer 초기화
    WAVE* sound = sound_loadmp3("test.mp3"); // mp3 파일 load

    sound_play(sound);     // mp3 파일 재생

    delayms(1000);

    sound_pause(sound);   // 1초 후에 Sound pause

    delayms(1000);

    If(sound_ispause(sound)) // 1초 후에 Sound 가 pause 상태인지 확인
    {
        sound_resume(sound); // pasuse 중이면 resume
    }

    ...
    ...

    while(1);
    return 0;
}
```

# 11. File System

adStar-L SDK 는 Nand Flash 와 SD Card 에 file system 을 적용하여 file 저장 장치로 사용할 수 있다. 이번 장에서는 Nand Flash 와 SD Card 를 file 저장장치로 사용하기 위해 초기 설정하는 법에 대해 설명을 하고, 자세한 file system 사용에 관한 내용은 adStar-L SDK 중 doc 폴더안의 fatfs 매뉴얼을 참고하기 바란다.( SDK/doc/fatfs/doc/00index\_e.html)

## 11.1 f\_mount

**FRESULT f\_mount(BYTE, FATFS\*);**

Nand Flash 또는 SD Card 를 mount 하여 file 저장 장치로 사용하도록 설정한다.

### Parameter

BYTE            Nand Flash 또는 SD Card 의 장치 번호. 이에 대해서는 다음과 같이 정의 되어 있다.

```
#define DRIVE_NAND 0
#define DRIVE_SDCARD 1
```

FATFS          장치 정보를 가지고 있는 구조체

### Result value

동작의 결과.

0 인 경우 정상 동작 한 것이다. 이 외에 결과값은 fatfs 매뉴얼을 참조하기 바란다.

## 11.2 f\_chdrive

**FRESULT f\_chdrive(BYTE);**

Mount 된 driver 간에 drive change 할 때 사용한다.

### Parameter

BYTE          Nand Flash 또는 SD Card 의 장치 번호.  
#define DRIVE\_NAND 0  
#define DRIVE\_SDCARD 1

### Return value

동작의 결과.

0 인 경우 정상 동작 한 것이다. 이 외에 결과값은 fatfs 매뉴얼을 참조하기 바란다.

## 11.3 f\_chdir

**FRESULT f\_chdir(const TCHAR\* );**

## Parameter

TCHAR\* 이동 할 directory 명

## Return value

## 동작의 결과.

0인 경우 정상 동작한 것이다. 이 외에 결과값은 fatfs 매뉴얼을 참조하기 바란다.

## 11.4 FILE System Example

```
int main()
{
    init_interrupt(); // Interrupt 초기화
    uart_config(1, 115200, DATABITS_8, STOPBITS_1, UART_PARNONE );

    FATFS nand_fs, sdcard_fs;
    f_mount(DRIVE_NAND, &nand_fs); // Nand Flash mount
    f_mount(DRIVE_SDCARD, &sdcard_fs); // SD Card mount

    SURFACE* bmp = loadbmp("test.bmp"); // Nand Flash에서 이미지 파일 Load

    f_chdrive(DRIVE_SDCARD); // SD Card로 drive change

    WAVE* wav = sound_loadwav("test.wav"); // SD Card에서 사운드 파일 Load

    f_chdir("font"); // SD Card의 font directory로 이동.

    ...
    ...

    while(1)
        return 0;
}
```

## 12. Font

adStar-L SDK에서는 두 종류의 font를 사용할 수 있다. 이미지 폰트(편의상 bmp font라 하겠다)와 bit 타입의 폰트(편의상 bit font라 하겠다)가 있는데, 이번 장에서는 이 두 폰트를 사용하기 위해 필요한 함수에 대해 설명하도록 하겠다.

bmp font는 프로그램을 사용하여 폰트 이미지를 제작해야 하지만 여러 종류의 폰트를 쉽게 사용할 수 있다. bit font는 따로 이미지 제작 없이 제공되는 파일을 가지고 사용할 수 있지만 폰트의 종류는 변경할 수 없다.

bmp font의 제작 방법은 추가로 제공되는 문서를 참고하기 바란다.

### << bmp font >>

#### 12.1 create\_bmpfont

**EGL\_FONT\*** create\_bmpfont(**const char** \*fname)

폰트 파일을 load하여 bmp font를 사용할 수 있도록 준비한다.

##### Parameter

fname 폰트 파일 이름

##### Result value

bmp font 포인터.

#### 12.2 release\_bmpfont

**void** release\_bmpfont(**EGL\_FONT\*** pFont);

생성한 bmp font 중 더 이상 사용하지 않을 font를 release 한다.

##### Parameter

pFont bmp font 포인터

##### Return value

없음.

### 12.3 bmpfont\_draw

```
int bmpfont_draw(EGL_FONT* pFont, int x, int y, const char *str)
```

bmp font 를 사용하여 특정 위치에 문자를 출력한다.

#### Parameter

pFont	출력에 사용할 bmp font 포인터.
x	문자가 출력될 x 좌표
y	문자가 출력될 y 좌표
str	출력될 문자(열)

#### Return value

출력된 문자열의 width.

### 12.4 bmpfont\_draw\_vleft

```
int bmpfont_draw_vleft(EGL_FONT* pFont, int x, int y, const char *str)
```

bmp font 를 사용하여 특정 위치에 문자를 출력하는데, 좌측으로 90 도 회전한 문자를 출력한다.

#### Parameter

pFont	출력에 사용할 bmp font 포인터.
x	문자가 출력될 x 좌표
y	문자가 출력될 y 좌표
str	출력될 문자(열)

#### Return value

출력된 문자열의 width.

### 12.5 bmfont\_draw\_vright

```
int bmfont_draw_vright(EGL_FONT* pFont, int x, int y, const char *str)
```

bmp font 를 사용하여 특정 위치에 문자를 출력하는데, 우측으로 90 도 회전한 문자를 출력한다.

#### Parameter

pFont	출력에 사용할 bmp font 포인터.
x	문자가 출력될 x 좌표
y	문자가 출력될 y 좌표

str 출력될 문자(열)

## Return value

출력된 문자열의 width.

## 12.6 egl\_font\_set\_color

**EGL\_COLOR** egl\_font\_set\_color(**EGL\_FONT\*** font, **EGL\_COLOR** clr)

폰트의 색을 변경한다.

## Parameter

font 색을 변경하려는 bmp font 포인터.

clr font color. MAKE\_COLORREF(r,g,b) 매크로를 사용한다.

## Return value

이전에 설정되어 있던 color.

## 12.7 bmpfont makesurface

**SURFACE\*** bmpfont makesurface(**EGL FONT\*** pFont, **char\*** text)

자주 사용하는 문자를 이미지화 시켜서 필요할 때마다 draw 함수를 사용하여 출력할 수 있도록 만들어 준다. surface 를 생성하기 때문에 더 이상 사용하지 않을 경우에는 release\_surface() 함수로 release 해주어야 한다.

## Parameter

pFont 출력에 사용할 bmp font 포인터.

text 이미지로 만들 문자(열)

## Return value

## SURFACE 구조체

## 12.8 bmpfont setkerning

**bool** bmpfont\_setkerning(**EGL FONT\*** pFont, **int** k)

**bmfont** 의 자간을 조정한다.

**Parameter**

- |       |                       |
|-------|-----------------------|
| pFont | 출력에 사용할 bmp font 포인터. |
| k     | 자간 간격                 |

**Return value**

TRUE(1) or FALSE(0)

**12.9 bmpfont\_setautokerning**

**bool** bmpfont\_setautokerning(**EGL\_FONT\*** pFont, **bool** b)

bmp font 의 글자의 간격을 동일하게 할지를 결정한다.

**Parameter**

- |   |   |
|---|---|
| b | TRUE : 글자의 간격을 동일하게 맞춘다. 각 글자마다 width 가 다르기 때문에,<br>문자열의 전체 width 가 글자에 따라 다르다.<br>FALSE : 글자의 간격을 글자의 width 에 따라 다르게 맞춘다.<br>문자열의 전체 Width 는 동일하다. |
|---|---|

**Return value**

TRUE(1) or FALSE(0)

**<< bit font >>****12.10 create\_bitfont**

**EGL\_FONT\*** create\_bitfont( )

bit font 를 사용할 수 있도록 생성한다.

**Return value**

bit font 포인터.

**12.11 release\_bitfont**

**void** release\_bitfont(**EGL\_FONT\*** pFont)

생성한 bit font 중 더 이상 사용하지 않을 font 를 release 한다.

**Parameter**

pFont              bit font 포인터

**Return value**

없음.

**12.12 bitfont\_draw**

**int bitfont\_draw(EGL\_FONT\* pFont, int x, int y, const char\* str)**

bit font 를 사용하여 문자를 출력한다.

**Parameter**

pFont	출력에 사용 할 bit font 포인터.
x	문자를 출력할 x 좌표
y	문자를 출력할 y 좌표
str	출력할 문자(열)

**12.13 bitfont\_draw\_vleft**

**int bitfont\_draw\_vleft(EGL\_FONT\* pFont, int x, int y, const char\* str)**

bit font 를 사용하여 문자를 출력하는데, 좌측으로 90 도 회전하여 출력한다.

**Parameter**

pFont	출력에 사용 할 bit font 포인터.
x	문자를 출력할 x 좌표
y	문자를 출력할 y 좌표
str	출력할 문자(열)

**12.14 bit\_draw\_vright**

**int bit\_draw\_vright(EGL\_FONT\* pFont, int x, int y, const char\* str)**

bit font 를 사용하여 문자를 출력하는데, 우측으로 90 도 회전하여 출력한다.

**Parameter**

pFont	출력에 사용 할 bit font 포인터.
x	문자를 출력할 x 좌표

y	문자를 출력할 y 좌표
str	출력할 문자(열)

### 12.15 bitfont\_makesurface

**SURFACE\*** bitfont\_makesurface(**EGL\_FONT\*** pFont, **char\*** str)

자주 사용하는 문자를 이미지화 시켜서 필요할 때마다 draw 함수를 사용하여 출력할 수 있도록 만들어 준다. surface 를 생성하기 때문에 더 이상 사용하지 않을 경우에는 release\_surface() 함수로 release 해주어야 한다.

#### Parameter

pFont	출력에 사용 할 bit font 포인터.
str	이미지로 만들 문자(열)

#### Return value

SURFACE 구조체

### 12.16 FONT Example

#### < bm font example >

```
#include "adStar-L.h"

int main()
{
    boardinit();
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);
    debugstring("=====WrWn");
    debugprintf("ADSTAR-L FONT Example. System Clock(%dMhz)WrWn",get_ahb_clock()/1000000);
    debugstring("=====WrWn");

    FATFS fs;
    f_mount(DRIVE_NAND,&fs);
    f_chdrive(DRIVE_NAND);

    crtc_clock_init();
```

```
SURFACE* frame = createframe(800,480,16);
setscreen(SCREEN_800x480,SCREENMODE_RGB565);
lcdon();

setframebuffer(frame);
setdrawtarget(frame);
drawsetrgb(255,255,255);
drawrectfill(0,0,getscreentwidth(),getscreenheight());

f_chdir("font");           // font 파일이 있는 font 폴더로 이동.
EGL_FONT* pFont = create_bmpfont("nanum_8bit_16px_han.fnt");
                         // nanum_8bit_16px_han.fnt 파일을 load 하여, 사용할 수 있도록 setting.
egl_font_set_color(pFont, MAKE_COLORREF(0, 0, 255));
                         // nanum 체의 font 색을 파란색으로 변경.
bmpfont_draw(pFont, 100,100,"나눔체 테스트");
                         // nanum 체를 사용하여 문자열을 100,100 위치에 출력.

SURFACE* fontsurf = bmpfont_makesurface(pFont, "bmfont_makesurface 로 만든 surface");
                         // 문자열을 하나의 image 로 만듦. drawsurface 함수를 사용하여 image 처럼 사용할 수 있다.
drawsurface(fontsurf,100,300);
                         // image 로 만든 font 를 100,300 위치에 출력.

while(1);
return 0;
}
```

## < bit font example >

```
#include "adStar-L.h"

int main()
{
    boardinit();
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);
    debugstring("=====Wr\n");
    debugprintf("ADSTAR-L FONT Example. System Clock(%dMhz)Wr\n",get_ahb_clock()/1000000);
    debugstring("=====Wr\n");

    FATFS fs;
    f_mount(DRIVE_NAND,&fs);
    f_chdrive(DRIVE_NAND);

    crtcclock_init();
    SURFACE* frame = createframe(800,480,16);
    setscreen(SCREEN_800x480,SCREENMODE_RGB565);
    lcdon();

    setframebuffer(frame);
    setdrawtarget(frame);
    drawsetrgb(255,255,255);
    drawrectfill(0,0,getscreentwidth(),getscreentheight());

    EGL_FONT* pFont = create_bitfont();           // bit font 를 사용하도록 setting.
    egl_font_set_color(pFont, MAKE_COLORREF(255,0,0)); // bit type font 색을 빨간색으로 변경.
    bitfont_draw(pFont, 100,100,"bit type font test"); // bit type font 를 사용하여 100,100 위치에 문자열을 출력.

    SURFACE* fontsurf = bitfont_makesurface(pFont, "bf_makesurface 로 만들어진 surface");
    // 문자열을 하나의 image 로 만듦. drawsurface 함수를 사용하여 image 처럼 사용할 수 있다.
    drawsurface(fontsurf,100,300); // image 로 만든 font 를 100,300 위치에 출력.

    while(1);
    return 0;
}
```

## < font image 제작 방법 >

1. <http://www.angelcode.com/products/bmfont>에서 Bitmap Font Generator v1.12를 다운받아 설치한다.

### Bitmap Font Generator

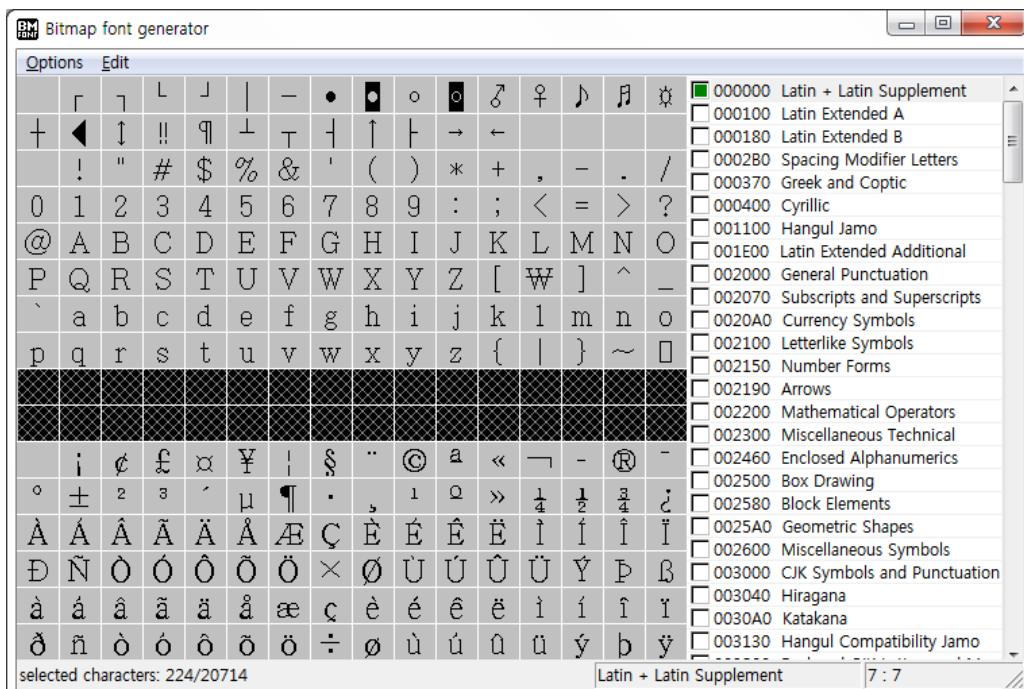
This program will allow you to generate bitmap fonts from TrueType fonts. The application generates both image files and character descriptions that can be read by a game for easy rendering of fonts.

[download installer for v1.12 \(344KB\)](#)

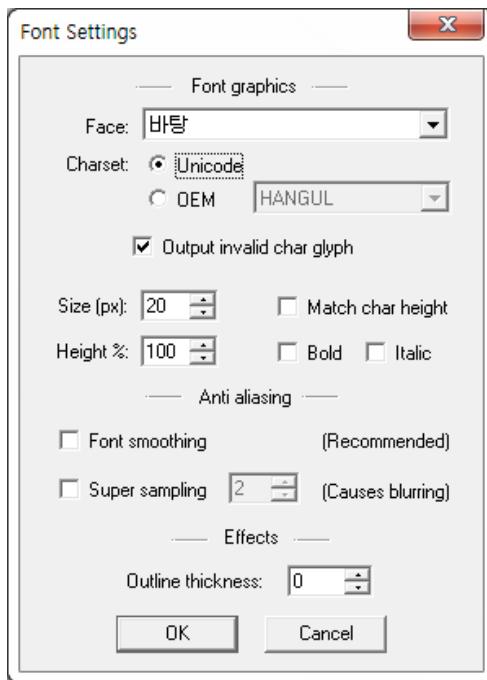
[download installer for v1.12a beta \(426KB\)](#)



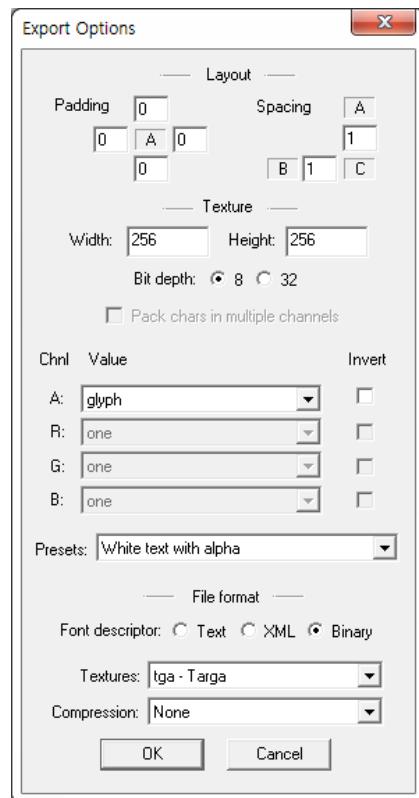
2. 설치한 후 실행하면 아래 그림과 같은 창이 나온다.



3. Options → Font settings 을 실행하여 원하는 Font 와 Font 의 size 를 결정한다.

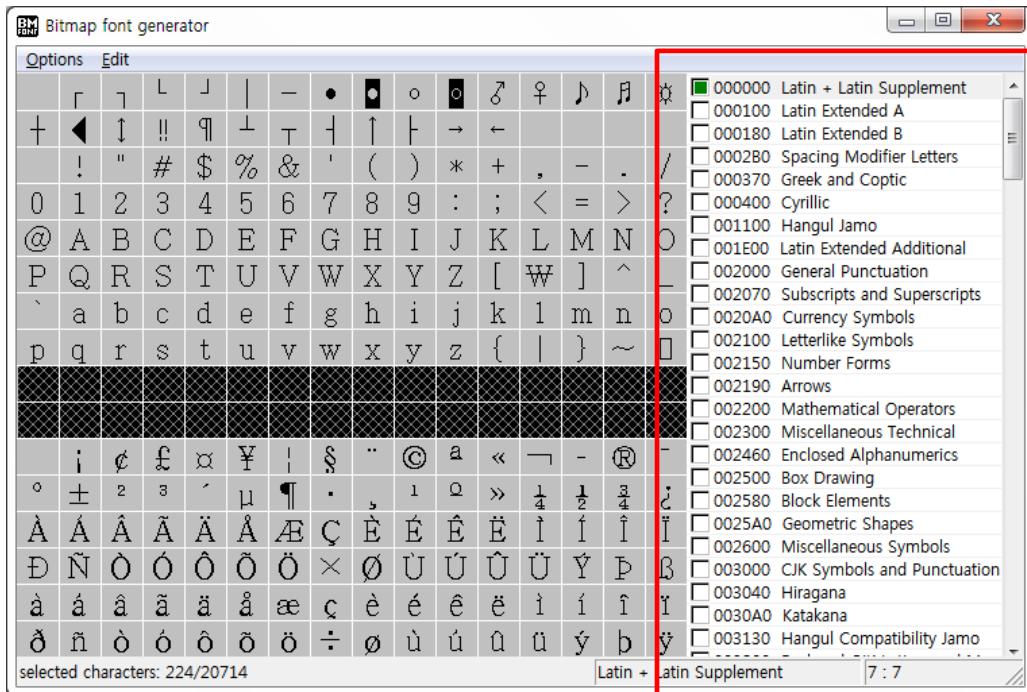


4. Options → Export options 를 실행하여 아래 그림과 같이 설정한다.



Width 나 Height 는 하나의 bitmap 파일의 가로 세로 크기로, 크기가 커지면 총 bitmap 파일의 수는 줄어드나 낭비되는 영역이 존재할 수 있다.

5. 처음 실행했을 때 나온 main 창에서 사용하고 싶은 font 를 체크한다. 좌측 화면에서 font 를 마우스로 선택하여 선택된 font 는 image 생성이 되지 않게도 할 수 있다.



6. Option → Save bitmap font as 를 실행하여 저장하면, FNAME.fnt 파일과 FNAME\_xx.tga 파일이 생성된다. 생성된 파일 중 FNAME.fnt 파일을 bmfont\_init 함수로 load 하면 된다.
7. fnt → font 의 위치 정보와 출력에 필요한 기타 정보를 저장  
tga → 실제 font image



# 13. SPI

## 13.1 spi\_master\_init

```
void spi_master_init(int ch)
```

SPI master 초기 설정을 한다. SPI 2 개 channel 을 지원하는데, 원하는 channel 을 선택하여 사용할 수 있다.

### Parameter

ch      channel 값.

## 13.2 spi\_set\_freq

```
int spi_set_freq(int ch, int mode, U32 freq)
```

SPI baud rate 을 설정한다.

### Parameter

ch      channel 값.

mode     master/slave 값. (SPI\_MASTER / SPI\_SLAVE)

freq     frequency 값.

### Return value

Baud rate 값.

## 13.3 spi\_master\_xfer

```
void spi_master_xfer(int ch, U8 *wbuf, int wlength, U8 *rbuf, int rlength, int continue_xfer)
```

wlength 만큼 wbuf 의 데이터를 Slave 로 전송하고, rlength 만큼 rbuf 에 데이터를 Slave 로부터 읽어온다.

### Parameter

ch      channel 값.

wbuf     전송할 data buffer.

wlength   전송할 data 길이.

Rbuf     읽어올 data buffer.

rlength   읽어올 data 길이.

Continue\_xfer 계속 전송 여부 결정.

### 13.4 spi\_wait\_empty\_fifo

**void** spi\_wait\_empty\_fifo(**int** ch)

SPI Tx FIFO 가 비었는지를 확인한다.

### Parameter

ch channel 값.

### 13.5 SPI Example

<SPI EEPROM Example>

==== w25xxx\_flash.h ===

```
#ifndef __W25Xxx_H__
#define __W25Xxx_H__

int w25xxx_write_buffer (U32 addr, U8 *buffer, int length);
int w25xxx_read_buffer (U32 addr, U8 *buffer, int length);
int w25xxx_erase (U32 start_addr, int length);
int w25xxx_block_erase (U32 addr, int length);
int w25xxx_wait_ready (void);
int w25xxx_check_id (void);

#define W25XXX_PAGE_SIZE      (256)
#define W25XXX_SECTOR_SIZE    (4096)
#define W25XXX_BLOCK_SIZE     (16*W25XXX_SECTOR_SIZE)

#define W25X16_TOTAL_SIZE     (2*1024*1024)
#define W25X32_TOTAL_SIZE     (64*W25XXX_BLOCK_SIZE)
#define W25X64_TOTAL_SIZE     (128*W25XXX_BLOCK_SIZE)
#define W25X32_FLASH_MAXADDR  W25X32_TOTAL_SIZE

// W25Xxx: 3.3V -> Max ??MHz
#define W25Xxx_SPI_FREQ   23000000 // 23MHz

// W25Xxx Flash instruction
#define W25Xxx_WREN        0x06 // Write Enable
#define W25Xxx_WRDI        0x04 // Write Disable
```

```
#define W25Xxx_RDSR      0x05 // Read Status Register
//#define W25Xxx_WRSR      0x01 // Write Status Register
#define W25Xxx_READ        0x03 // Read Data Bytes
#define W25Xxx_FAST_READ   0x0B // Read Data Bytes at Higher Speed
#define W25Xxx_DUAL_READ   0x3B // Read Data Bytes at Higher Speed
#define W25Xxx_Page_Program 0x02 // Page Program
#define W25Xxx_BERASE      0xD8 // Block Erase
#define W25Xxx_SERASE      0x20 // Sector Erase
#define W25Xxx_CERASE      0xC7 // Sector Erase
#define W25Xxx_READID      0x90 // Read ID /* ID: 0x00, 0xEF, (0x14, 0x15, 0x16) */
#define W25Xxx_READJEDECID 0x9F // Read ID /* ID: 0x00, 0xEF, (0x14, 0x15, 0x16) */

#define W25XXX_MAKERID     0xEF
#define W25X16_DEVID       0x14
#define W25X32_DEVID       0x15

#define W25Xxx_PDOWN       0xB9 // Power-down
#define W25Xxx_PON          0xAB // Release Power-down / Device ID

#define NORMAL             W25Xxx_READ
#define FAST               W25Xxx_FAST_READ
#define BLOCK              W25Xxx_BERASE
#define SECTOR             W25Xxx_SERASE
#define ALL                W25Xxx_CERASE

#define W25Xxx_SR_WIP      (1 << 0) // Wait In Progress bit
#define W25Xxx_SR_WEL      (1 << 1) // Wait Enable Latch bit

#define W25Xxx_ID1          0x00 // Manufacture ID
#define W25Xxx_ID2          0xEF // Memory Type
#define W25X16_ID3          0x14 // Memory Capacity
#define W25X32_ID3          0x15 // Memory Capacity
#define W25X64_ID3          0x16 // Memory Capacity

#define W25Xxx_JEDECID1    0xEF // Manufacture ID
#define W25Xxx_JEDECID2    0x30 // Memory Type
#define W25X16_JEDECID3    0x15 // Memory Capacity
#define W25X32_JEDECID3    0x16 // Memory Capacity
#define W25X64_JEDECID3    0x17 // Memory Capacity

#endif //__W25Xxx_H__
```

**==== w25xxx\_flash.c ====**

```
#include "adStar-L.h"
#include "w25xxx_flash.h"
#include "spi.h"

int w25xxx_write_buffer (U32 addr, U8 *buffer, int length)
{
    int wlen;
    U8 wbuf[4];
    int ret = 0;
    int i;

    U32 startaddr = addr;
    U32 endaddr = addr+length-1;
    U32 startpage;
    U32 endpage;

    startpage = startaddr/W25XXX_PAGE_SIZE;
    endpage = endaddr/W25XXX_PAGE_SIZE;

    for(i=startpage;i<endpage+1;i++)
    {
        U32 offset = addr%W25XXX_PAGE_SIZE;
        if(offset + length > W25XXX_PAGE_SIZE)
        {
            wlen = W25XXX_PAGE_SIZE - offset;
        }
        else
        {
            wlen = length;
        }

        asm("clr 13");

        wbuf[0] = W25Xxx_WREN;
        spi_master_xfer(SPI_CH1,wbuf, 1, NULL, 0, 0);
        ret = w25xxx_wait_ready();
        if (ret != 0)
            break;
    }
}
```

```
wbuf[0] = W25Xxx_Page_Program;

wbuf[1] = (addr>>16);
wbuf[2] = (addr>>8);
wbuf[3] = addr;

spi_master_xfer(SPI_CH1,wbuf, 4, NULL, 0, 1); // Write Command & Address
                                                // Continue to write data next
spi_master_xfer(SPI_CH1,buffer, wlen, NULL, 0, 0); // Write Data

/* Write Cycle Time of W25Xxx FLASH: 3.3V -> 11ms, 1.2ms */
ret = w25xxx_wait_ready();
if (ret != 0)
    break;

buffer += wlen;
addr += wlen;
length -=wlen;

wbuf[0] = W25Xxx_WRDI;
spi_master_xfer(SPI_CH1,wbuf, 1, NULL, 0, 0);
ret = w25xxx_wait_ready();
if (ret != 0)
    break;

asm("set 13");
}

return ret;
}

/*
 * mode: NORMAL, FAST
 */
int w25xxx_read_buffer (U32 addr, U8 *buffer, int length)
{
    int ret = 0;
    U8 wbuf[4];

    wbuf[0] = W25Xxx_READ;
    wbuf[1] = (addr>>16);
    wbuf[2] = (addr>>8);
```

```
wbuf[3] = addr;
asm("clr 13");
spi_master_xfer(SPI_CH1,wbuf, 4, buffer, length, 0);

ret = w25xxx_wait_ready();
asm("set 13");

return ret;
}

/*
 * SECTOR erase
 */
int w25xxx_erase (U32 addr, int length)
{
    U8 wbuf[4];
    int ret = 0;
    unsigned int esize = 0;
    unsigned int sector_size = W25XXX_SECTOR_SIZE;
    unsigned int block_size = W25XXX_BLOCK_SIZE;

    length += sector_size - 1;
    length &= ~(sector_size - 1);

    addr &= ~(sector_size - 1);

    while (length) {

        if (length >= block_size) {
            esize = block_size;
        }
        else if (length >= sector_size) {
            esize = sector_size;
        }

        wbuf[0] = W25Xxx_WREN;
        spi_master_xfer(SPI_CH1, wbuf, 1, NULL, 0, 0);
        ret = w25xxx_wait_ready();
        if (ret != 0)
            break;
    }
}
```

```
if (esize == block_size) {
    wbuf[0] = W25Xxx_BERASE;
}
else if (esize == sector_size) {
    wbuf[0] = W25Xxx_SERASE;
}

wbuf[1] = (addr>>16);
wbuf[2] = (addr>>8);
wbuf[3] = addr;

spi_master_xfer(SPI_CH1, wbuf, 4, NULL, 0, 0);

ret = w25xxx_wait_ready();
if (ret != 0)
    break;

addr += esize;
length -= esize;

wbuf[0] = W25Xxx_WRDI;
spi_master_xfer(SPI_CH1,wbuf, 1, NULL, 0, 0);
ret = w25xxx_wait_ready();
if (ret != 0)
    break;
}

return 0;
}

int w25xxx_wait_ready (void)
{
U8 inst = W25Xxx_RDSR;
U8 status;
int delay = 0;
int ret = 0;

do {
    spi_master_xfer (SPI_CH1, &inst, 1, &status, 1, 0);
    delay++;
    if (delay > 0x80000) {
        ret = -1;
    }
}
```

```
        break;
    }
} while (status & W25Xxx_SR_WIP);
return ret;
}

int w25xxx_check_id (void)
{
    U8 buf[4];

    buf[0] = W25Xxx_READID;
    buf[1] = 0;
    buf[2] = 0;
    buf[3] = 0;

    spi_master_xfer(SPI_CH1, buf, 4, buf, 2, 0);

    w25xxx_wait_ready();

    if ((buf[0] != W25XXX_MAKERID) || (buf[1] != W25X32_DEVID)) {
        debugprintf(" 0x%x ", buf[0] );
        debugprintf("0x%x %r%w", buf[1] ); // W25X32_ID3
        return -1;
    }
    return 0;
}
```

**==== main.c ====**

```
#include "adStar-L.h"
#include "w25xxx_flash.h"

extern void boardinit();

void SPI_pin_init()
{
    *R_PAF4 &= ~(0xf<<12);
    *R_PAF4 |= 0x5<<12;      //spi_sck1, spi_cs1

    *R_PAF5 &= ~(0xf<<0);
    *R_PAF5 |= 0x5<<0;      //spi_miso1, spi_mosi1
}

int main()
{
    boardinit();
    uart_config(1,115200,DATABITS_8,STOPBITS_1,UART_PARNONE);

    debugstring("=====Wr\n");
    debugprintf("ADSTAR-L SPI_Flash.System Clock(%dMhz)\r\n",get_ahb_clock()/1000000);
    debugstring("=====Wr\r\n");

    SPI_pin_init();

    int ret;

    spi_master_init(SPI_CH1);
    int baud = spi_set_freq(SPI_CH1,SPI_MASTER, W25Xxx_SPI_FREQ);
    debugprintf("SPI SCK : %d KHz\r\n", SPI_SCK(baud)/1000 );

    ret = w25xxx_check_id();
    if (ret != 0) {
        debugstring("W25Xxx ID Check Fail\r\n!");
        while(1);
    }

    U8 rbuf[256];
    U8 wbuf[256];
    int i;
```

```
for(i=0;i<256;i++)
    wbuf[i]=i;
for(i=0;i<256;i++)
    rbuf[i]=0;

w25xxx_erase(0,W25XXX_SECTOR_SIZE);
w25xxx_write_buffer(0,wbuf,256);
w25xxx_read_buffer(0,rbuf,256);
for(i=0;i<256;i++)
{
    if(wbuf[i]!=rbuf[i])
    {
        debugprintf("wbuf[%d] = %d != rbuf[%d] = %d \r\n",i,wbuf[i],i,rbuf[i]);
        debugstring("Test Error\r\n");
        while(1);
    }
    else
    {
        debugprintf("wbuf[%d] = %d == rbuf[%d] = %d \r\n",i,wbuf[i],i,rbuf[i]);
    }
}
while(1);
return 0;
}
```

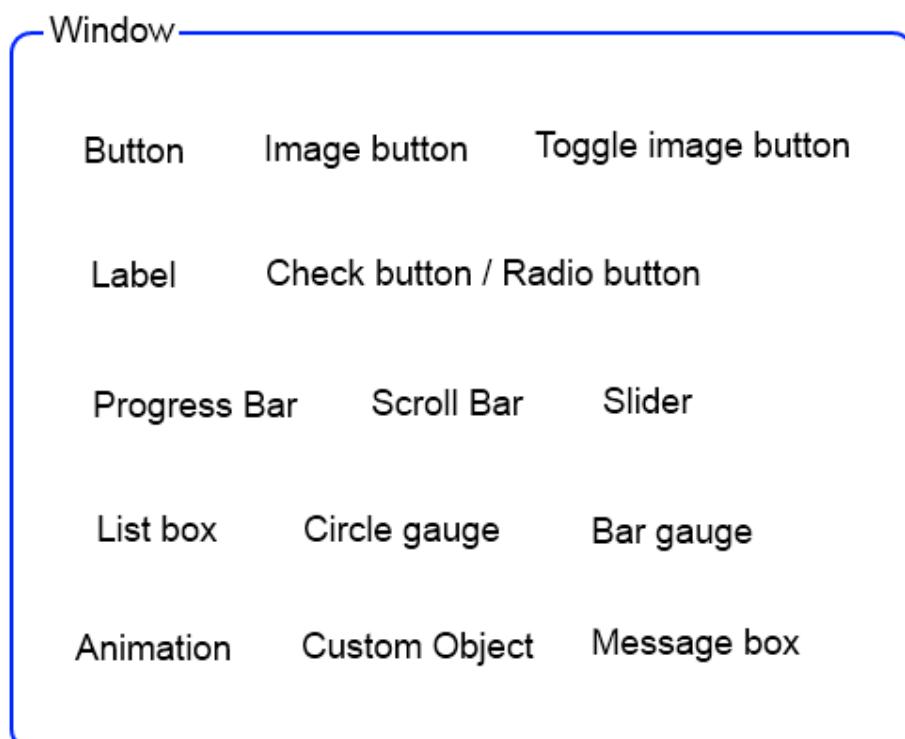
# 14. EGL Library

## 1) 소개

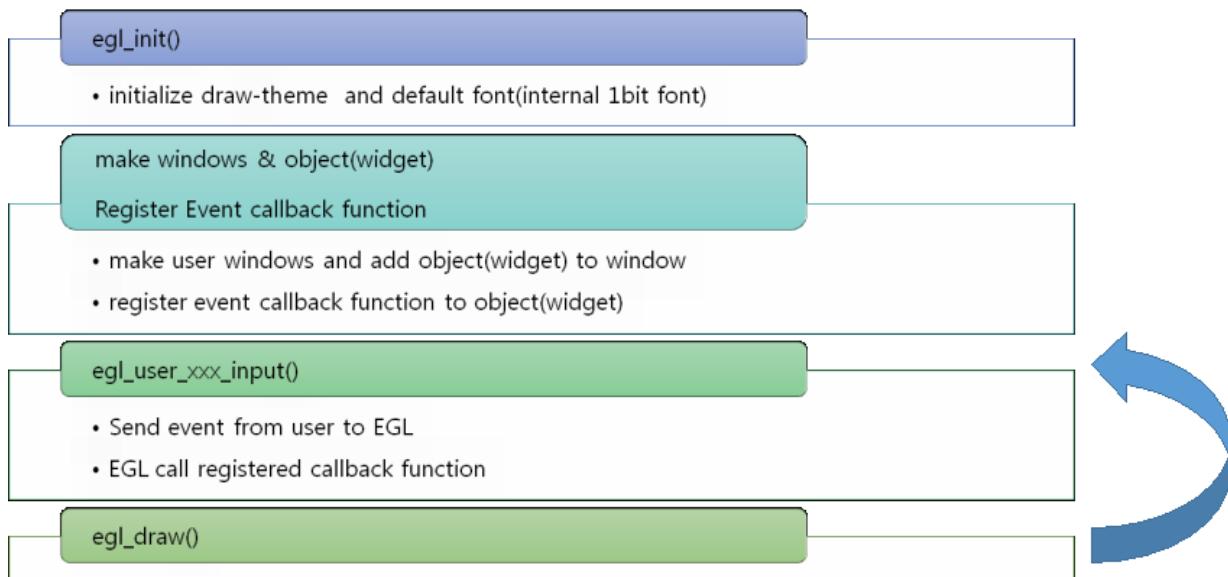
EGL(Embedded Graphic Library)은 graphic application 개발시간을 단축하고 다양한 object를 편리하게 사용할 수 있도록 제작 되었다. EGL은 다양한 object를 간편하게 사용할 수 있도록 함수를 제공하며, event에 대한 처리와 object 관리를 library 내에서 하도록 되어 있다. Event에 대한 추가 동작은 각 object별 사용자 callback 함수를 두어, 개발자가 원하는 동작의 callback 함수를 작성하여 사용할 수 있다.

## 2) 구성

EGL은 다음의 object들로 구성되어 있다.



### 3) Program 기본 구조



< Source Code >

```

#include "adStar-L.h"

static void btn_callback(EGL_HANDLE h, int event) // button callback function.
{
    /* 사용자 call back 함수 */
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn;
    egl_init();
    hWin = egl_create_window("Main Window");      // create window
    btn = egl_create_button(100,100,200,50,"Button 1"); // create button obj
    egl_btn_callback(btn, btn_callback);
    egl_window_add_object(hWin, btn);           // button obj add.
    egl_window_show(hWin);                     // window show

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) ) // touch 입력.
    }
}
  
```

```
egl_user_touch_input( touchdown, &touch_pt ); // touch 처리, msg 처리.
```

```
egl_draw();
```

```
}
```

```
...
```

```
}
```

# Window

Function	Description
<a href="#"><u>egl_create_window</u></a>	Window 를 생성한다.
<a href="#"><u>egl_window_show</u></a>	해당 window 를 화면에 출력할지를 결정한다.
<a href="#"><u>egl_window_isshow</u></a>	해당 window 가 현재 보여지고 있는지를 확인한다.
<a href="#"><u>egl_window_invalidate</u></a>	현재 window 안의 object 를 다시 draw 한다.
<a href="#"><u>egl_window_invalidate_rect</u></a>	현재 window 의 지정한 영역의 object 를 다시 draw 한다.
<a href="#"><u>egl_window_redraw_rect</u></a>	현재 window 의 지정한 영역을 다시 draw 한다.
<a href="#"><u>egl_window_set_bg</u></a>	Window 의 배경 이미지를 설정한다.
<a href="#"><u>egl_window_get_bg</u></a>	Window 의 배경 이미지를 가져온다.
<a href="#"><u>egl_window_add_object</u></a>	Window 에 object 를 추가한다.
<a href="#"><u>egl_window_set_callback</u></a>	Window 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#"><u>egl_window_get_active</u></a>	현재 보이는 window 의 handle 값을 반환한다.
<a href="#"><u>egl_user_touch_input</u></a>	Msg_handler 로 touch message 를 전달한다.
<a href="#"><u>egl_draw</u></a>	현재 windows 의 object 를 draw 한다.
<a href="#"><u>egl_visible_object</u></a>	Object 의 visible 상태를 설정한다.
<a href="#"><u>egl_window_delete_object</u></a>	해당 window 에서 object 를 제거한다.
<a href="#"><u>egl_release_window</u></a>	Window 를 제거한다.

## ► egl\_create\_window function

```
EGL_HANDLE egl_create_window(  
    const char* title  
);
```

## Overview

Window 를 생성한다. Window 의 크기는 사용하는 해상도와 같다.

## Parameter

## Return Value

생성된 Window 의 handle.

## Example

```
EGL_HANDLE hWin;  
  
hWin = egl_create_window("Main Window");
```

## ► egl\_window\_show function

```
BOOL egl_window_show(  
    EGL_HANDLE hWin,  
    BOOL bShow  
)
```

### Overview

해당 window 를 화면에 출력할지를 결정한다.

### Parameter

EGL_HANDLE hWin	출력을 결정 할 window handle.
BOOL bShow	출력 여부 결정.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
hWin = egl_create_window("Main Window");  
...  
egl_window_show(hWin, TRUE);
```

## ► egl\_window\_isshow function

```
BOOL egl_window_isshow(  
    EGL_HANDLE hWin  
)
```

### Overview

해당 window 가 현재 보여지고 있는지를 확인한다.

### Parameter

EGL_HANDLE hWin	출력을 상태를 확인 할 window handle.
-----------------	-----------------------------

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
hWin = egl_create_window("Main Window");  
...  
if(egl_window_isshow(hWin))  
{  
    ...  
}
```

## ► egl\_window\_invalidate function

```
void egl_window_invalidate( void );
```

### Overview

현재 window 안의 object 를 다시 draw 한다.

### Parameter

없음.

### Return Value

없음.

### Example

```
EGL_HANDLE hWin;  
hWin = egl_create_window("Main Window");  
...  
egl_window_invalidate( );
```

## ► `egl_window_invalidate_rect` function

```
void egl_window_invalidate_rect(  
    EGL_RECT* pRect  
) ;
```

## Overview

현재 window 의 지정한 영역의 object 를 다시 draw 한다.

## Parameter

EGL\_RECT\* pRect 지정 영역 value.

```
typedef Struct _tag_RECT{  
    int x;  
    int y;  
    int w;  
    int h;  
}EGL_RECT
```

## Return Value

없음.

## Example

```
EGL_HANDLE hWin;  
EGL_RECT pRect;  
hWin = egl_create_window("Main Window");  
...  
pRect.x = 100;  
pRect.y = 200;  
pRect.w = 150;  
pRect.h = 150;  
egl_window_invalidate_rect(&pRect );
```

## ► **egl\_window\_redraw\_rect** function

```
Void egl_window_redraw_rect(  
    EGL_RECT* pRect  
)
```

## Overview

현재 window 의 지정한 영역을 다시 draw 한다.

## Parameter

EGL\_RECT\* pRect 지정 영역 value.

```
typedef Struct _tag_RECT{  
    int x;  
    int y;  
    int w;  
    int h;  
}EGL_RECT
```

## Return Value

없음.

## Example

```
EGL_HANDLE hWin;  
EGL_RECT pRect;  
hWin = egl_create_window("Main Window");  
...  
pRect.x = 100;  
pRect.y = 200;  
pRect.w = 150;  
pRect.h = 150;  
egl_window_redraw_rect(&pRect);
```

## ► egl\_window\_set\_bg function

```
BOOL egl_window_set_bg(  
    EGL_HANDLE hWin;  
    SURFACE* Img;  
)
```

### Overview

Window 의 배경 이미지를 설정한다.

### Parameter

EGL_HANDLE hWin	배경 이미지를 설정할 Window handle.
SURFACE* Img	배경 이미지 SURFACE

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
SURFACE* surface = loadbmp("windowbg.bmp");  
hWin = egl_create_window("Main Window");  
egl_window_set_bg(hWin, surface);
```

## ► `egl_window_get_bg` function

```
SURFACE* egl_window_get_bg(  
    EGL_HANDLE hWin;  
);
```

## Overview

Window 의 배경 이미지를 가져온다.

## Parameter

EGL\_HANDLE hWin 배경 이미지를 가져 올 Window handle.

## Return Value

해당 window 의 image.

## Example

```
EGL_HANDLE hWin;  
SURFACE* getimage;  
hWin = egl_create_window("Main Window");  
...  
getimage = egl_window_get_bg(hWin);
```

## ► egl\_window\_add\_object function

```
BOOL egl_window_add_object(  
    EGL_HANDLE hWin,  
    EGL_HANDLE hObj  
)
```

### Overview

Window 에 object 를 추가한다.

### Parameter

EGL_HANDLE hWin	Object 를 추가 할 window handle.
EGL_HANDLE hObj	Window 에 추가 할 object handle.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
EGL_HANDLE btn;  
hWin = egl_create_window("Main Window");  
btn = egl_create_button(100, 100, 100, 50, "Button");  
egl_window_add_object(hWin, btn);
```

## ► egl\_window\_set\_callback function

```
void egl_window_set_callback(  
    EGL_HANDLE hWin,  
    EVENT_CALLBACK cb  
)
```

### Overview

Window 이벤트가 발생했을 때 호출 되는 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hWin	Callback 함수가 등록되는 Window handle.
EVENT_CALLBACK cb	Event 발생 시 호출되는 callback 함수.

### Return Value

없음.

### Example

```
Void window_callback(EGL_HANDLE h, int event)  
{  
    ...  
}  
  
int main()  
{  
    ...  
  
    EGL_HANDLE hWin;  
    hWin = egl_create_window("Main Window");  
    egl_window_callback(hWin, window_callback);  
    ...  
}
```

## ► egl\_window\_get\_active function

```
EGL_HANDLE egl_window_get_active( void );
```

### Overview

현재 출력되고 있는 window 의 handle 값을 반환한다.

### Parameter

없음.

### Return Value

현재 출력되고 있는 window handle.

### Example

```
EGL_HANDLE hWin;  
EGL_HANDLE active_win;  
hWin = egl_create_window("Main Window");  
...  
active_win = egl_window_get_active( );
```

## ► egl\_user\_touch\_input function

```
void egl_user_touch_input(
    BOOL bPressed,
    EGL_POINT* pt
);
```

### Overview

Msg\_handler 로 touch message 를 전달한다.

### Parameter

BOOL bPressed	Touch 가 눌린 상태.
EGL_POINT* pt	Touch 된 좌표.

```
typedef struct _tagPOINT
{
    int x;
    int y;
} EGL_POINT;
```

### Return Value

없음.

### Example

```
EGL_POINT touch_pt;
BOOL touchdown = FALSE;
...
if(process_touch(&touchdown, &touch_pt))
    egl_user_touch_input(touchdown, &touch_pt);
```

## ► egl\_draw function

```
void egl_draw( void );
```

### Overview

현재 windows 의 object 를 draw 한다.

### Parameter

없음.

### Return Value

없음.

### Example

```
if(process_touch(&touchdown, &touch_pt))
    egl_user_touch_input(touchdown, &touch_pt);
egl_draw();
```

## ► egl\_visible\_object function

```
void egl_visible_object (
    EGL_HANDLE h,
    bool b
);
```

### Overview

Object 의 visible 상태를 설정한다. 설정 후 egl\_window\_invalidate( ) 함수를 호출해야 화면에 적용된다.  
(또는 egl\_window\_invalidate\_rect( ), egl\_window\_redraw\_rect() 함수를 사용해도 된다. )

### Parameter

EGL_HANDLE h	visible 상태를 변경할 object handle.
bool b	visible 상태, TRUE : visible, FALSE : invisible

### Return Value

없음.

### Example

```
egl_visible_object( button, FALSE );
egl_window_invalidate( );
...
egl_draw( );
```

## ► egl\_window\_delete\_object function

```
bool egl_window_delete_object (
    EGL_HANDLE hWin,
    EGL_HANDLE hObj
);
```

### Overview

해당 window 의 object 를 제거한다. Window object list 에서만 제거된다.

제거 후 egl\_window\_invalidate( ) 함수를 호출해야 화면에 적용된다.

(또는 egl\_window\_invalidate\_rect( ), egl\_window\_redraw\_rect() 함수를 사용해도 된다. )

### Parameter

EGL_HANDLE hWin	Object 가 포함 된 window handle.
EGL_HANDLE hObj	제거 할 object handle.

### Return Value

TRUE or FALSE.

### Example

```
egl_window_delete_object( hWin, button );
egl_window_invalidate( );
...
egl_draw( );
```

## ► **egl\_release\_window** function

```
    bool egl_release_window (
```

## Overview

해당 window 를 제거한다.

## Parameter

## Return Value

**TRUE or FALSE.**

## Example

```
If(hWin != NULL)
{
    egl_release_window( hWin );
    hWin = NULL;
}
```

## ► Window Example.

### Example

```
#include "adStar-L.h"

static void btn1_callback(EGL_HANDLE h, int event)
{
    if( event == BTN_CLICKED )
    {
        debugprintf("btn1 clicked\r\n");
        egl_window_show( hWin2, TRUE );
    }
}

static void btn2_callback(EGL_HANDLE h, int event)
{
    if( event == BTN_CLICKED )
    {
        debugprintf("btn2 clicked\r\n");
        egl_window_show( hWin1, TRUE );
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin1;
    EGL_HANDLE hWin2;
    EGL_HANDLE btn1;
    EGL_HANDLE btn2;
    SURFACE* bg = loadbmp("bg.bmp");
    egl_init();
    hWin1 = egl_create_window("First Window");
    egl_windows_set_bg(hWin1, bg);
    hWin2 = egl_create_window("Second Window");
    btn1 = egl_create_button(100, 100, 200, 50, "Second Window");
```

```
egl_btn_callback(btn1, btn1_callback);
btn2 = egl_create_button(100, 200, 200, 50, "First Window");
egl_btn_callback(btn2, btn2_callback);
egl_window_add_object(hWin1, btn1);
egl_window_add_object(hWin2, btn2);
egl_window_show(hWin1);
egl_draw();

while(1)
{
    if( process_touch(&touchdown, &touch_pt ) )
        egl_user_touch_input(touchdown, &touch_pt );

    egl_draw();
}
```

# Button



Function	Description
<a href="#"><u>egl_create_button</u></a>	Button 을 생성한다.
<a href="#"><u>egl_button_callback</u></a>	Button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#"><u>egl_release_button</u></a>	Button 을 제거한다.

## Define

```
BUTTON_EVENT           typedef enum
{                      BTN_CLICKED = 0,
                      BTN_PRESSED,
                      BTN_MAX,
}BUTTON_EVENT
```

## ► egl\_create\_button function

```
EGL_HANDLE egl_create_button(  
    int x,  
    int y,  
    int w,  
    int h,  
    const char* text  
)
```

### Overview

Button 을 생성한다.

### Parameter

int x	생성될 Button 의 x 좌표.
int y	생성될 Button 의 y 좌표.
int w	생성될 Button 의 가로 크기.
int h	생성될 Button 의 세로 크기.
const char* text	Button 에 출력될 text.

### Return Value

생성된 Button 의 handle.

### Example

```
EGL_HANDLE btn[2];  
  
btn[0] = egl_create_button(100, 100, 100, 50, "Button1");  
  
btn[1] = egl_create_button(300, 100, 100, 50, "Button2");
```

## ► egl\_button\_callback function

```
BOOL egl_button_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 Button handle.
EVENT_CALLBACK cb	Event 가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void btn_callback(EGL_HANDLE h, int event)  
{  
    If(event == BTN_CLICKED)  
    {  
        ...  
    }  
}  
  
int main()  
{  
    ...  
    EGL_HANDLE btn;  
    btn = egl_create_button(100, 100, 100, 50, "Button1");  
    egl_button_callback(btn,btn_callback);  
    ...  
}
```

## ► egl\_release\_button function

```
    BOOL egl_release_button(  
        EGL_HANDLE hObj  
    );
```

## Overview

Button 을 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE

## Example

```
if(button != NULL)
{
    egl_window_delete_object( hWin, button);
    egl_release_button( button );
    button = NULL;
}
```

## ► Button Example.

### Example

```
void btn_callback(EGL_HANDLE h, int event)
{
    if(event == BTN_CLICKED)
    {
        debugprintf("button clicked.");
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn;
    egl_init();
    btn = egl_create_button(100, 100, 100, 50, "Button Ex");
    egl_button_callback(btn,btn_callback);
    egl_window_add_object(hWin, btn);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

## Image button



&lt; img &gt;

&lt; pressed\_img &gt;

Function	Description
<a href="#">egl_create_image_button</a>	Image button 을 생성한다.
<a href="#">egl_image_button_callback</a>	Image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다
<a href="#">egl_release_image_button</a>	Image button 을 제거한다.

## ► egl\_create\_image\_button function

```
EGL_HANDLE egl_create_image_button(  
    SURFACE* img,  
    SURFACE* pressed_img,  
    int x,  
    int y,  
    int w,  
    int h  
) ;
```

### Overview

Image button 을 생성한다.

### Parameter

SURFACE* img	Image button 의 표시 될 image.
SURFACE* pressed_img	눌렸을 때 표시 될 image.
int x	Image button 의 x 좌표.
int y	Image button 의 y 좌표.
int w	Image button 의 가로 크기.
int h	Image button 의 세로 크기.

### Return Value

생성 된 Image button handle.

### Example

```
EGL_HANDLE imagebtn;  
SURFACE* img = loadbmp("btning.bmp");  
SURFACE* pressed_img = loadbmp("pressedimg.bmp");  
imagebtn = egl_create_image_button(img, pressed_img, 100, 100, 150, 50);
```

## ► egl\_image\_button\_callback function

```
BOOL egl_image_button_callback(
    EGL_HANDLE hObj,
    EVENT_CALLBACK cb
);
```

### Overview

Image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 Image button handle.
EVENT_CALLBACK cb	Event 가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void imgbtn_callback(EGL_HANDLE h, int event)
{
    ...
}

int main()
{
    ...
    EGL_HANDLE imagebtn;
    SURFACE* img = loadbmp("btning.bmp");
    SURFACE* pressed_img = loadbmp("pressedimg.bmp");
    imagebtn = egl_create_image_button(img, pressed_img, 100, 100, 150, 50);
    egl_image_button_callback(imagebtn, imgbtn_callback);
    ...
}
```

## ► **egl\_release\_image\_button** function

```
    BOOL egl_release_image_button(
```

## Overview

Image button 을 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE

## Example

```
if(imgbtn != NULL)
{
    egl_window_delete_object( hWin, imgbtn );
    egl_release_image_button(imgbtn);
    imgbtn = NULL;
}
```

## ► Image Button Example.

### Example

```
void imgbtn_callback(EGL_HANDLE h, int event)
{
    if(event == BTN_CLICKED)
    {
        debugprintf("image button clicked.");
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE imbtn;
    SURFACE* img = loadbmp("btning.bmp");
    SURFACE* pressed_img = loadbmp("pressedimg.bmp");
    egl_init();
    imbtn = egl_create_image_button(img, pressed_img, 100, 100, 150, 50);
    egl_button_callback(imbtn, imbtn_callback);
    egl_window_add_object(hWin, imbtn);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

## Toggle image button



&lt; surf\_on &gt;



&lt; surf\_off &gt;

Function	Description
<a href="#">egl_create_toggle_image</a>	Toggle image button 을 생성한다.
<a href="#">egl_toggle_image_callback</a>	Toggle image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_toggle_image_set_on</a>	Toggle image button 의 상태를 설정한다.
<a href="#">egl_release_toggle_image</a>	Toggle image 를 제거한다.

### Define

```
TOGGLE_IMAGE_EVENT           typedef enum
{                           TOGGLE_IMAGE_ON = 0,
                           TOGGLE_IMAGE_OFF,
} TOGGLE_IMAGE_EVENT;
```

## ► egl\_create\_toggle\_image function

```
EGL_HANDLE egl_create_toggle_image(
    SURFACE* surf_off,
    SURFACE* surf_on,
    int x,
    int y,
    int w,
    int h
);
```

### Overview

Toggle image button 을 생성한다.

### Parameter

SURFACE* surf_off	Toggle imgae 중 off 시에 출력 할 image. ( default 상태 : Off )
SURFACE* surf_on	Toggle image 중 on 시에 출력 할 image.
int x	Toggle image 의 x 좌표.
int y	Toggle image 의 y 좌표.
int w	Toggle image 의 가로 크기.
int h	Toggle image 의 세로 크기.

### Return Value

생성 된 toggle image handle.

### Example

```
EGL_HADLE toggleimg;
SURFACE* surf_off = loadpng("off.png");
SURFACE* surf_on = loadpng("on.png");
toggleimg = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);
```

## ► egl\_toggle\_image\_callback function

```
BOOL egl_toggle_image_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Toggle image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 toggle image handle.
EVENT_CALLBACK cb	Event 가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void toggleimg_callback(EGL_HANDLE h, int event)  
{  
    if(event == TOGGLE_IMAGE_ON)  
        ...  
    else  
        ...  
}  
int main()  
{  
    EGL_HANDLE toggleimage;  
    ...  
    toggleimage = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);  
    egl_toggle_image_callback(toggleimage, toggleimg_callback);  
    ...  
}
```

## ► egl\_toggle\_image\_set\_on function

```
BOOL egl_toggle_image_set_on(  
    EGL_HANDLE hObj,  
    BOOL b  
)
```

### Overview

Toggle image button의 상태를 설정한다.

### Parameter

EGL_HANDLE hObj	상태를 설정 할 toggle image handle.
BOOL b	설정 값. TRUE => On FALSE => Off

### Return Value

TRUE or FALSE

### Example

```
EGL_HADLE toggleimg;  
SURFACE* surf_off = loadpng("off.png");  
SURFACE* surf_on = loadpng("on.png");  
toggleimg = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);  
egl_toggle_image_set_on(toggleimg, TRUE);
```

## ► **egl\_release\_toggle\_image** function

```
    BOOL egl_release_toggle_image(  
        EGL_HANDLE hObj,  
    );
```

## Overview

Toggle image button 을 제거한다. 해당 object 가 window 에 add 되어 있는 경우,  
egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE

## Example

```
if( toggle != NULL )
{
    egl_window_delete_object( hWin, toggle );
    egl_release_toggle_image( toggle );
    toggle = NULL;
}
```

## ► Toggle Image Button Example.

### Example

```
void toggleimg_callback(EGL_HANDLE h, int event)
{
    if(event == TOGGLE_IMAGE_ON)
        debugprintf(" toggle image on \r\n");
    else
        debugprintf(" toggle image off \r\n");
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE toggleimg;
    SURFACE* surf_off = loadpng("off.png");
    SURFACE* surf_on = loadpng("on.png");
    egl_init();
    toggleimg = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);
    egl_toggle_image_callback(toggleimg, toggleimg_callback);
    egl_toggle_image_set_on(toggleimg, TRUE);
    egl_window_add_object(hWin, toggleimg);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Label

Function	Description
<a href="#"><u>egl_create_label</u></a>	Label object 를 생성한다.
<a href="#"><u>egl_label_callback</u></a>	Label 이벤트가 발생했을 때 호출 될 callback 함수를 등록한다.
<a href="#"><u>egl_label_set_text</u></a>	Label text 를 설정한다.
<a href="#"><u>egl_label_set_redraw_bg</u></a>	Label 배경을 draw 할지를 결정한다.
<a href="#"><u>egl_label_set_color</u></a>	Label text color 를 설정한다.
<a href="#"><u>egl_release_label</u></a>	Label 을 제거한다.

## ► egl\_create\_label function

```
EGL_HANDLE egl_create_label(  
    int x,  
    int y,  
    int w,  
    int h,  
    const char* text  
)
```

### Overview

Label object 를 생성한다.

### Parameter

int x	label 의 x 좌표.
int y	label 의 y 좌표.
int w	label 의 가로 크기.
int h	label 의 세로 크기.
const char* text	label 의 text.

### Return Value

생성 된 label handle.

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test");
```

## ► egl\_label\_callback function

```
BOOL egl_label_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Label 이벤트가 발생했을 때 호출 될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 label handle.
EVENT_CALLBACK cb	Event 가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void label_callback(EGL_HANDLE h, int event)  
{  
    if(event == LABEL_CLICKED)  
    {  
        ....  
    }  
  
    int main()  
    {  
        EGL_HANDLE label;  
        ....  
        label = egl_create_label(100,100,150,50,"label");  
        egl_label_callback(label, label_callback);  
        ...  
    }
```

## ► egl\_label\_set\_text function

```
BOOL egl_label_set_text(  
    EGL_HANDLE h,  
    char* text  
)
```

### Overview

Label 의 text 를 변경한다.

### Parameter

EGL_HANDLE h	Text 를 변경 할 label handle.
char text	변경 할 text.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test" );  
egl_label_set_text( label, "change text");
```

## ► egl\_label\_set\_redraw\_bg function

```
void egl_label_set_redraw_bg(  
    EGL_HANDLE h,  
    BOOL b  
)
```

### Overview

Label 배경을 draw 할지를 결정한다. label 을 단독 object 로 사용할 경우 TRUE 로 설정해야 한다.

### Parameter

EGL_HANDLE hObj	상태를 설정 할 label handle.
BOOL b	설정 값. TRUE or FALSE

### Return Value

없음.

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test" );  
egl_label_set_redraw_bg(label, TRUE);
```

## ► egl\_label\_set\_color function

```
void egl_label_set_color(  
    EGL_HANDLE h,  
    EGL_COLOR clr  
)
```

### Overview

Label text color 를 설정한다.

### Parameter

EGL_HANDLE h	Color 를 변경 할 label handle.
EGL_COLOR clr	Color value. MAKE_COLORREF(r, g, b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HADLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test" );  
egl_label_set_color( label, MAKE_COLORREF( 0, 255, 0 ) );
```

## ► egl\_release\_label function

```
    BOOL egl_release_label(  
        EGL_HANDLE hObj  
    );
```

## Overview

Label 을 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE

## Example

```
if( label != NULL )  
{  
    egl_window_delete_object( hWin, label );  
    egl_release_label( label );  
    label = NULL;  
}
```

## ► Label Example.

### Example

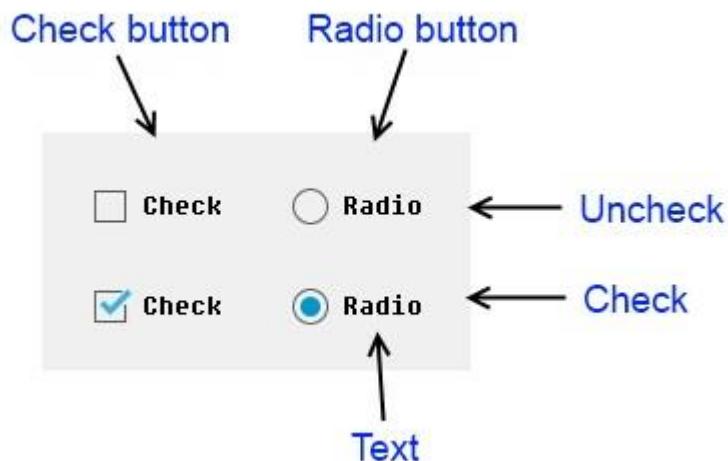
```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE label;
    egl_init();
    label = egl_create_label(50, 50, 100, 30, "label");
    egl_label_set_redraw_bg(label, TRUE);
    egl_label_set_color(label, MAKE_COLORREF(0, 0, 255));
    egl_window_add_object(hWin, label);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Check button / Radio button



Function	Description
<a href="#">egl_create_checkbutton</a>	Check / Radio button 을 생성한다.
<a href="#">egl_checkbutton_callback</a>	Check / Radio button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_checkbutton_set_check</a>	Check / Radio button 을 check / uncheck 상태를 설정한다.
<a href="#">egl_checkbutton_get_check</a>	Check / Radio button 의 현재 check 상태 값을 가져온다.
<a href="#">egl_checkbutton_set_style</a>	Check / Radio button 의 style 을 변경한다.
<a href="#">egl_release_checkbutton</a>	Check / Radio button 을 제거한다.

## Define

```
CHECK_EVENT
typedef enum
{
    CHECK_CHECKED = 0,
    CHECK_UNCHECKED,
} CHECK_EVENT;
```

## ► egl\_create\_checkbutton function

```
EGL_HANDLE egl_create_checkbutton(
    int x,
    int y,
    int w,
    int h,
    const char* text,
    CHECK_STYLE style
);
```

### Overview

Check / Radio button 을 생성한다.

### Parameter

int x	생성될 Check / Radio button 의 x 좌표.
int y	생성될 Check / Radio button 의 y 좌표.
int w	생성될 Check / Radio button 의 가로 크기.
int h	생성될 Check / Radio button 의 세로 크기.
const char* text	Check / Radio button 에 출력될 text.
CHECK_STLE style	Check / Radio button 의 style. Check button ->CHECK_STYLE_CHECKBUTTON. Radio button ->CHECK_STYLE_RADIOBUTTON.

### Return Value

생성된 Check / Radio button 의 handle.

### Example

```
EGL_HANDLE check;
EGL_HANDLE radio;
check = egl_create_checkbutton(100,100, 100, 32, "Check", CHECK_STYLE_CHECKBUTTON);
radio = egl_create_checkbutton(250, 100, 100, 32, "Radio", CHECK_STYLE_RADIOBUTTON);
```

## ► egl\_checkbutton\_callback function

```
BOOL egl_button_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Check / Radio button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 Check / Radio button handle.
EVENT_CALLBACK cb	Event 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
Void check_callback(EGL_HANDLE h, int event)  
{  
    if(event == CHECK_CHECKED)  
    ...  
    else if(event == CHECK_UNCHECKED)  
    ...  
}  
int main()  
{  
    ...  
    EGL_HANDLE check;  
    check = egl_create_checkbutton(100, 100, 100, 32, "Check", STYLE_CHECKBUTTON);  
    egl_checkbutton_callback(check, check_callback);  
    ...  
}
```

## ► egl\_checkbutton\_set\_check function

```
void egl_checkbutton_set_check(  
    EGL_HANDLE hObj,  
    BOOL b  
)
```

### Overview

Check / Radio button 을 check / uncheck 상태를 설정한다.

### Parameter

EGL_HANDLE hObj	Check 상태를 설정할 Check / Radio button handle.
BOOL b	TRUE = Check. FALSE = Uncheck.

### Return Value

없음.

### Example

```
EGL_HANDLE check;  
  
EGL_HANDLE radio;  
  
check = egl_create_checkbutton(100, 100, 100, 32, "Check", STYLE_CHECKBUTTON);  
  
radio = egl_create_checkbutton(200, 100, 100, 32, "Radio", STYLE_RADIOBUTTON);  
  
egl_checkbutton_set_check(check, TRUE);  
  
egl_checkbutton_set_check(radio, FALSE);
```

## ► egl\_checkbutton\_get\_check function

```
BOOL egl_checkbutton_get_check(  
    EGL_HANDLE hObj,  
);
```

### Overview

Check / Radio button 의 현재 check 상태 값을 가져온다.

### Parameter

EGL_HANDLE hObj	Check 상태를 가지고 올 Check / Radio button handle.
-----------------	--

### Return Value

Check 상태를 반환. TRUE = check.

FALSE = uncheck.

### Example

```
EGL_HANDLE check;  
  
EGL_HANDLE radio;  
  
BOOL check_status = 0;  
  
BOOL radio_status = 0;  
  
check = egl_create_checkbutton(100, 100, 100, 32, "Check", STYLE_CHECKBUTTON);  
  
radio = egl_create_checkbutton(200, 100, 100, 32, "Radio", STYLE_RADIOBUTTON);  
  
check_status = egl_checkbutton_get_check(check);  
  
radio_status = egl_checkbutton_get_check(radio);
```

## ► egl\_checkbutton\_set\_style function

```
void egl_checkbutton_set_style(  
);
```

```
EGL_HANDLE hObj,
CHECK_STYLE style
);
```

## Overview

Check / Radio button 의 style 을 변경한다.

## Parameter

EGL_HANDLE hObj	Style 을 변경할 Check / Radio button handle.
CHECK_STYLE style	Check / Radio button 의 style. Check button ->CHECK_STYLE_CHECKBUTTON. Radio button ->CHECK_STYLE_RADIOBUTTON.

## Return Value

없음

## Example

```
EGL_HANDLE check;

EGL_HANDLE radio;

check = egl_create_checkbutton(100, 100, 100, 32, "Check", CHECK_STYLE_CHECKBUTTON);

radio = egl_create_checkbutton(200, 100, 100, 32, "Radio",
CHECK_STYLE_RADIOBUTTON);

egl_checkbutton_set_style(check, CHECK_STYLE_RADIOBUTTON);

egl_checkbutton_set_style(radio, CHECK_STYLE_CHECKBUTTON);
```

## ► egl\_release\_checkbutton function

```
BOOL egl_release_checkbutton(
    EGL_HANDLE hObj
);
```

## Overview

Check / Radio button 을 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE

## Example

```
if( check != NULL)
{
    egl_window_delete_object( hWin, check );
    egl_release_checkbutton( check );
    check = NULL;
}
```

## ► Check / Radio button Example.

### Example

```
EGL_HANDLE check;
EGL_HANDLE radio1;
EGL_HANDLE radio2;

void check_callback(EGL_HANDLE h, int event)
{
    if(event == CHECK_CHECKED)
    {
        if(egl_checkbutton_get_check())
        {
            debugprintf("status : checked\r\n");
        }
        else
        {
            debugprintf("status : unchecked\r\n");
        }
    }
}

void radio1_callback(EGL_HANDLE h, int event)
{
    if(event == CHECK_CHECKED)
    {
        debugprintf("radio1 select\r\n");
        egl_checkbutton_set_check(radio2, FALSE);
    }
}

void radio2_callback(EGL_HANDLE h, int event)
{
    if(event == CHECK_CHECKED)
    {
        debugprintf("radio2 select\r\n");
        egl_checkbutton_set_check(radio1, FALSE);
    }
}
```

```
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...

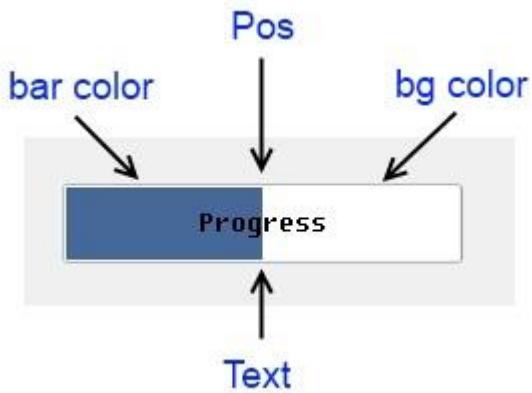
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;

    egl_init();
    check = egl_create_checkbutton(100, 100, 100, 32, "Check", CHECK_STYLE_CHECKBUTTON);
    egl_checkbutton_callback(check, check_callback);
    radio1 = egl_create_checkbutton(250, 100, 100, 32, "Radio1", CHECK_STYLE_RADIOBUTTON);
    egl_checkbutton_callback(radio1, radio1_callback);
    radio2 = egl_create_checkbutton(250, 160, 100, 32, "Radio2", CHECK_STYLE_RADIOBUTTON);
    egl_checkbutton_callback(radio2, radio2_callback);
    egl_window_add_object(hWin, check);
    egl_window_add_object(hWin, radio1);
    egl_window_add_object(hWin, radio2);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Progress Bar



Function	Description
<a href="#"><u>egl_create_progressbar</u></a>	Progress bar 를 생성한다.
<a href="#"><u>egl_progressbar_set_barcolor</u></a>	Progress bar 의 bar color 를 설정한다.
<a href="#"><u>egl_progressbar_set_bgcolor</u></a>	Progress bar 의 back ground color 를 설정한다.
<a href="#"><u>egl_progressbar_set_textcolor</u></a>	Progress bar 의 text color 를 설정한다.
<a href="#"><u>egl_progressbar_set_text</u></a>	Progress bar 의 text 를 설정한다.
<a href="#"><u>egl_progressbar_set_pos</u></a>	Progress bar 의 bar position 을 설정한다.
<a href="#"><u>egl_progressbar_get_pos</u></a>	Progress bar 의 현재 bar position value 를 가져온다.
<a href="#"><u>egl_release_progressbar</u></a>	Progress bar 를 제거한다.

## ► egl\_create\_progressbar function

```
EGL_HANDLE egl_create_progressbar(
    int x,
    int y,
    int w,
    int h,
    const char* text,
    BOOL style
    BOOL bVertical
);
```

### Overview

Progress bar 를 생성한다.

### Parameter

int x	생성될 Progress bar 의 x 좌표.
int y	생성될 Progress bar 의 y 좌표
int w	생성될 Progress bar 의 가로 크기
int h	생성될 Progress bar 의 세로 크기
const char* text	Progress bar 에 출력 될 text.
BOOL style	Progress bar 의 style. Nomal = STYLE_PGBAR_NOMAL. Divide bar = STYLE_PGVAR_DIV.
BOOL bVertical	Progress 의 출력 형태. TRUE = Vertical. FALSE = Horizontal.

### Return Value

생성 된 Progress bar 의 handle.

### Example

```
EGL_HANDLE pgbar;
pgbar = egl_create_progressbar(100, 100, 200, 40, "Progress", STYLE_PGBAR_NOMAL, FALSE);
```

## ► egl\_progressbar\_set\_barcolor function

```
void egl_progressbar_set_barcolor(  
    EGL_HANDLE hObj,  
    EGL_COLOR clr  
)
```

### Overview

Progress bar 의 bar color 를 설정한다.

### Parameter

EGL_HANDLE hObj	Bar color 를 설정할 Progress bar handle.
EGL_COLOR clr	Color value.
	MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL,  
FALSE);  
  
egl_progressbar_set_barcolor(pgbar, MAKE_COLORREF(0, 0, 0xff));
```

## ► egl\_progressbar\_set\_bgcolor function

```
void egl_progressbar_set_bgcolor(  
    EGL_HANDLE hObj,  
    EGL_COLOR clr  
)
```

### Overview

Progress bar 의 back ground color 를 설정한다.

### Parameter

EGL_HANDLE hObj	Back ground color 를 설정 할 Progress bar handle.
EGL_COLOR clr	Color value. MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL,  
FALSE);  
  
egl_progressbar_set_bgcolor(pgbar, MAKE_COLORREF(0, 0xff, 0));
```

## ► egl\_progressbar\_set\_textcolor function

```
void egl_progressbar_set_textcolor(  
    EGL_HANDLE hObj,  
    EGL_COLOR clr  
)
```

### Overview

Progress bar 의 text color 를 설정한다.

### Parameter

EGL_HANDLE hObj	Text color 를 설정 할 Progress bar handle.
EGL_COLOR clr	Color value.
	MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL,  
FALSE);  
  
egl_progressbar_set_textcolor(pgbar, MAKE_COLORREF(0xff, 0, 0xff));
```

## ► egl\_progressbar\_set\_text function

```
void egl_progressbar_set_text(  
    EGL_HANDLE hObj,  
    const char* text  
)
```

### Overview

Progress bar 의 text 를 설정한다.

### Parameter

EGL_HANDLE hObj	Text 를 설정할 Progress bar handle.
const char* text	설정 할 Text.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL,  
FALSE);  
  
egl_progressbar_set_text(pgbar, "text1");
```

## ► egl\_progressbar\_set\_pos function

```
void egl_progressbar_set_pos(  
    EGL_HANDLE hObj,  
    int value  
)
```

### Overview

Progress bar 의 bar position 을 설정한다.

### Parameter

EGL_HANDLE hObj	Bar position 을 변경 할 Progress bar handle.
int value	Bar position 값 ( 0 ~ 100% )

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL,  
FALSE);  
  
egl_progressbar_set_pos(pgbar, 50);
```

## ► egl\_progressbar\_get\_pos function

```
int egl_progressbar_get_pos(  
    EGL_HANDLE hObj  
)
```

### Overview

Progress bar 의 현재 bar position value 를 가져온다.

### Parameter

EGL\_HANDLE hObj                          Bar position value 를 가져 올 Progress bar handle.

### Return Value

Bar position value. ( 0 ~ 100%)

### Example

```
EGL_HANDLE pgbar;  
  
int pgbar_pos;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL,  
FALSE);  
  
pgbar_pos = egl_progressbar_get_pos(pgbar);
```

## ► [egl\\_release\\_progressbar](#) function

```
    BOOL egl_release_progressbar(  
        EGL_HANDLE hObj  
    );
```

## Overview

Progress bar 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE

## Example

```
if(pgbar != NULL)
{
    egl_window_delete_object( hWin, pgbar );
    egl_release_progressbar(pgbar);
    pgbar = NULL;
}
```

## ► Progress Bar Example.

### Example

```
EGL_HANDLE pgbar;
int value = 0;

void pgbar_callback(EGL_HANDLE h, int event)
{
    value++;
    if(value > 100)
        value = 0;
    egl_progressbar_set_pos(pgbar, value);
}

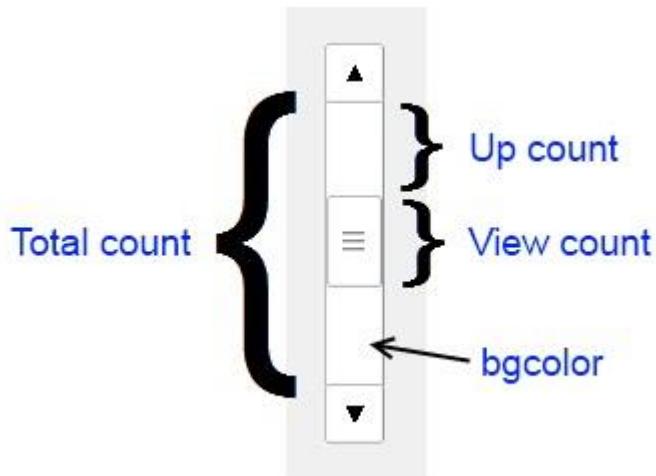
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    egl_init();
    pgbar = egl_create_progressbar(100, 100, 200, 40, "progress", STYLE_PGBAR_NOMAL, FALSE);
    egl_progress_callback(pgbar, pgbar_callback);
    egl_window_add_object(hWin, pgbar);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

## Scroll Bar



Function	Description
<a href="#"><u>egl_create_scrollbar</u></a>	Scroll bar 를 생성한다.
<a href="#"><u>egl_scrollbar_callback</u></a>	Scroll bar 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#"><u>egl_scroll_set_position</u></a>	Scroll bar 의 thumb position 을 설정한다.
<a href="#"><u>egl_scroll_get_position</u></a>	Scroll bar 의 thumb position 을 가져온다.
<a href="#"><u>egl_scroll_set_totalcount</u></a>	Scroll bar 의 total count 를 설정한다.
<a href="#"><u>egl_scroll_get_totalcount</u></a>	Scroll bar 의 total count 를 가져온다.
<a href="#"><u>egl_scroll_set_view_count</u></a>	Scroll bar 의 view count 를 설정한다.
<a href="#"><u>egl_scroll_get_view_count</u></a>	Scroll bar 의 view count 를 가져온다.
<a href="#"><u>egl_scroll_set_upcount</u></a>	Scroll bar 의 up count 를 설정한다.
<a href="#"><u>egl_scroll_get_upcount</u></a>	Scroll bar 의 up count 를 가져온다.
<a href="#"><u>egl_scroll_set_bgcolor</u></a>	Scroll bar 의 back ground color 를 설정한다.
<a href="#"><u>egl_scroll_set_size</u></a>	Scrollbar 의 크기를 설정한다.

[egl\\_release\\_scrollbar](#)

Scrollbar 를 제거한다.

**Define**

SCROLLBAR\_EVENT

```
typedef enum
{
    SCBAR_CLICKED = 0,
} SCROLLBAR_EVENT;
```

## ► egl\_create\_scrollbar function

```
EGL_HANDLE egl_create_scrollbar(
    int x,
    int y,
    int w,
    int h,
    int totalcount,
    int viewcount,
    BOOL bVertical
);
```

### Overview

Scroll bar 를 생성한다.

### Parameter

int x	생성 될 Scroll bar 의 x 좌표.
int y	생성 될 Scroll bar 의 y 좌표.
int w	생성 될 Scroll bar 의 가로 크기.
int h	생성 될 Scroll bar 의 세로 크기.
int totalcount	Scroll bar 가 관리할 항목의 전체 크기.
int viewcount	Scroll bar 의 thumb 가 표시할 항목의 크기
BOOL bVertical	Scroll bar 의 출력 형태. TRUE = Vertical. FALSE = Horizontal.

### Return Value

생성 된 Scroll bar handle.

### Example

```
EGL_HANDLE scroll;
scroll = egl_create_scrollbar(100, 100, 30, 200, 30, 10, TRUE);
```

## ► egl\_scrollbar\_callback function

```
BOOL egl_scrollbar_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Scroll bar 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

ELG_HANDLE hObj	Callback 함수를 등록 할 Scroll bar handle.
EVENT_CALLBACK cb	호출 될 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void scroll_callback(EGL_HANDLE h, int event)  
{  
  
}  
int main()  
{  
  
...  
    EGL_HANDLE scroll;  
    scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
    egl_scroll_callback(scroll, scroll_callback);  
    ...  
}
```

## ► egl\_scroll\_set\_position function

```
void egl_scroll_set_position(  
    EGL_HANDLE hObj,  
    int totalcount,  
    int viewcount,  
    int upcount  
)
```

### Overview

Scroll bar 의 thumb position 을 설정한다.

### Parameter

EGL_HANDLE hObj	Thumb position 을 설정 할 Scroll bar handle.
int totalcount	Scroll bar 가 관리 할 항목의 전체 크기.
int viewcount	Scroll bar 의 thumb 가 표시할 항목의 크기.
int upcount	Scroll bar 의 thumb 이전 항목의 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_position(scroll, 25, 5, 10);
```

## ► egl\_scroll\_get\_position function

```
void egl_scroll_get_position(  
    EGL_HANDLE hObj,  
    int* totalcount,  
    int* viewcount,  
    int* upcount  
) ;
```

### Overview

Scroll bar 의 thumb position 을 가져온다.

### Parameter

EGL_HANDLE hObj	Thumb position 을 가져 올 Scroll bar handle.
int* totalcount	Total count 를 저장 할 포인터.
int* viewcount	View count 를 저장 할 포인터.
int* upcount	Up count 를 저장 할 포인터.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
int totalcount = 0;  
int viewcount = 0;  
int upcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
egl_scroll_get_position(scroll, &totalcount, &viewcount, &upcount);
```

## ► egl\_scroll\_set\_totalcount function

```
void egl_scroll_set_totalcount(  
    EGL_HANDLE hObj,  
    int totalcount  
)
```

### Overview

Scroll bar 의 total count 를 설정한다.

### Parameter

EGL_HANDLE hObj	Total count 를 설정 할 Scroll bar handle.
int totalcount	Scroll bar 가 관리 할 항목의 전체 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_totalcount(scroll, 45);
```

## ► egl\_scroll\_get\_totalcount function

```
int egl_scroll_get_totalcount(  
    EGL_HANDLE hObj  
)
```

### Overview

Scroll bar 의 total count 를 가져온다.

### Parameter

EGL\_HANDLE hObj                      Total count 를 가져 올 Scroll bar handle.

### Return Value

Total count value.

### Example

```
EGL_HANDLE scroll;  
int totalcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
totalcount = egl_scroll_get_totalcount(scroll);
```

## ► egl\_scroll\_set\_viewcount function

```
void egl_scroll_set_viewcount(  
    EGL_HANDLE hObj,  
    int viewcount  
)
```

### Overview

Scroll bar 의 view count 를 설정한다.

### Parameter

EGL_HANDLE hObj	View count 를 설정 할 Scroll bar handle.
int viewcount	Scroll bar 의 thumb 가 표시할 항목의 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_viewcount(scroll, 5);
```

## ► egl\_scroll\_get\_viewcount function

```
int egl_scroll_get_viewcount(  
    EGL_HANDLE hObj  
)
```

### Overview

Scroll bar 의 view count 를 가져온다.

### Parameter

EGL_HANDLE hObj	View count 를 가져 올 Scroll bar handle.
-----------------	--------------------------------------

### Return Value

View count value.

### Example

```
EGL_HANDLE scroll;  
int viewcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
viewcount = egl_scroll_get_viewcount(scroll);
```

## ► egl\_scroll\_set\_upcount function

```
void egl_scroll_set_upcount(  
    EGL_HANDLE hObj,  
    int upcount  
)
```

### Overview

Scroll bar 의 up count 를 설정한다.

### Parameter

EGL_HANDLE hObj	Up count 를 설정 할 Scroll bar handle.
int upcount	Scroll bar 의 thumb 이전 항목의 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_upcount(scroll, 10);
```

## ► egl\_scroll\_get\_upcount function

```
int egl_scroll_get_upcount(  
    EGL_HANDLE hObj  
)
```

### Overview

Scroll bar 의 up count 를 가져온다.

### Parameter

EGL\_HANDLE hObj                      Up count 를 가져 올 Scroll bar handle.

### Return Value

Up count value.

### Example

```
EGL_HANDLE scroll;  
int upcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
upcount = egl_scroll_get_upcount(scroll);
```

## ► egl\_scroll\_set\_bgcolor function

```
void egl_scroll_set_bgcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
)
```

### Overview

Scroll bar 의 back ground color 를 설정한다.

### Parameter

EGL_HANDLE hObj	Back ground color 를 설정 할 Scroll bar handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_bgcolor(scroll, 0, 0, 0xff);
```

## ► egl\_scroll\_set\_size function

```
void egl_scroll_set_size(  
    EGL_HANDLE hObj,  
    int w,  
    int h  
)
```

### Overview

Scroll bar 의 크기를 설정한다.

### Parameter

EGL_HANDLE hObj	크기를 설정 할 Scroll bar handle.
int w	가로 크기.
int h	세로 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
...  
egl_scroll_set_size(scroll, 40, 200);
```

## ► egl\_release\_scrollbar function

```
    BOOL egl_release_scrollbar(  
        EGL_HANDLE hObj  
)
```

## Overview

Scroll bar 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

### TRUE or FALSE.

## Example

```
If( scroll != NULL )  
{  
    egl_window_delete_object( hWin, scroll );  
    egl_release_scrollbar( scroll );  
    scroll = NULL;  
}
```

## ► Scroll Bar Example.

### Example

```
EGL_HANDLE scroll;
void scroll_callback(EGL_HANDLE h, int event)
{
    int totalcount = 0;
    int viewcount = 0;
    int upcount = 0;

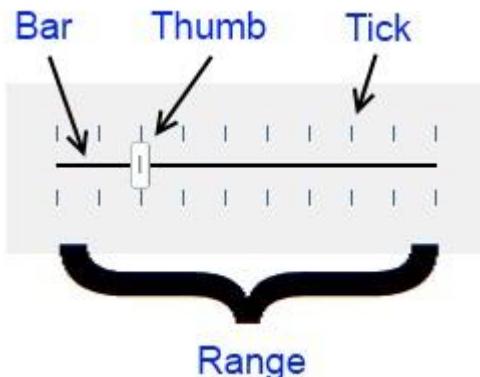
    if(event == SCBAR_CLICKED)
    {
        egl_scroll_get_position(scroll, &totalcount, &viewcount, &upcount);
        debugprintf("%d, %d, %d \r\n", totalcount, viewcount, upcount);
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    egl_init();
    scroll = egl_create_scrollbar(100, 100, 30, 200, 30, 10, TRUE);
    egl_scroll_set_totalcount(scroll, 40);
    egl_scroll_callback(scroll, scroll_callback);
    egl_window_add_object(hWin, scroll);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Slider



Function	Description
<a href="#">egl_create_slider</a>	Slider 를 생성한다.
<a href="#">egl_slider_callback</a>	Slider 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_slider_set_pos</a>	Slider 의 thumb position 을 설정한다.
<a href="#">egl_slider_get_pos</a>	Slider 의 thumb position 값을 가져온다.
<a href="#">egl_slider_set_range</a>	Slider 의 range 를 설정한다.
<a href="#">egl_slider_get_range</a>	Slider 의 range 를 가져온다.
<a href="#">egl_slider_stepit</a>	Slider 를 전후로 한 step 이동한다.
<a href="#">egl_slider_set_tick_frequency</a>	Slider 의 tick 빈도 수를 설정한다.
<a href="#">egl_slider_set_tick_style</a>	Slider 의 tick style 을 설정한다.
<a href="#">egl_slider_set_thumb_size</a>	Slider 의 thumb size 를 설정한다.
<a href="#">egl_slider_get_thumb_size</a>	Slider 의 thumb size 를 가져온다.
<a href="#">egl_slider_set_barcolor</a>	Slider 의 bar color 를 설정한다.
<a href="#">egl_slider_set_tickcolor</a>	Slider 의 tick color 를 설정한다.

[egl\\_slider\\_set\\_transparent](#)

Slider 의 배경을 투명하게 할지를 결정한다.

[egl\\_release\\_slider](#)

Slider 를 제거한다.

**define**

SLIDER\_EVENT

```
typedef enum
{
    SLR_CLICKED = 0,
} SLIDER_EVENT;
```

## ► egl\_create\_slider function

```
EGL_HANDLE egl_create_slider(
    int x,
    int y,
    int w,
    int h,
    int range,
    TICKSTYLE style,
    BOOL bVertical
);
```

### Overview

Slider 를 생성한다.

### Parameter

int x	생성 될 Slider 의 x 좌표.
int y	생성 될 Slider 의 y 좌표.
int w	생성 될 Slider 의 가로 크기.
int h	생성 될 Slider 의 세로 크기.
int range	생성 될 Slider 의 Range.
TICKSTYLE style	생성 될 Slider 의 Tick style TICK_NONE : 눈금 표시 없음. TICK_TOPLEFT : 눈금 표시가 위쪽 또는 좌측에 표시. TICK_BOTTOMRIGHT : 눈금 표시가 아래쪽 또는 우측에 표시. TICK_BOTH : 눈금 표시가 양쪽에 표시.
BOOL bVertical	Slider 의 출력 형태. TRUE = Vertical. FALSE = Horizontal.

### Return Value

생성 된 Slider handle.

### Example

```
EGL_HANDLE slider;  
  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);
```

## ► egl\_slider\_callback function

```
BOOL egl_slider_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Slider 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수를 등록 할 Slider handle.
EVENT_CALLBACK cb	호출 될 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
Void slider_callback(EGL_HANDLE h, int event)  
{  
  
}  
  
int main()  
{  
  
    ...  
    EGL_HANDLE slider;  
    slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
    egl_slider_callback(slider, slider_callback);  
    ...  
}
```

## ► egl\_slider\_set\_pos function

```
void egl_slider_set_pos(  
    EGL_HANDLE hObj,  
    int nPos  
)
```

### Overview

Slider 의 thumb position 을 설정한다.

### Parameter

EGL_HANDLE hObj	Position 을 설정 할 Slider handle.
int nPos	Position 위치. (nPos 는 정수로 0 ~ range)

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_pos(slider, 2);
```

## ► egl\_slider\_get\_pos function

```
int egl_slider_get_pos(  
    EGL_HANDLE hObj  
)
```

### Overview

Slider 의 thumb position 값을 가져온다.

### Parameter

EGL\_HANDLE hObj                          Position 을 가져 올 Slider handle.

### Return Value

Position value.

### Example

```
EGL_HANDLE slider;  
int slider_pos = 0;  
slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
slider_pos = egl_slider_get_pos(slider);
```

## ► egl\_slider\_set\_range function

```
void egl_slider_set_range(  
    EGL_HANDLE hObj,  
    int nMinPos,  
    int nMaxPos  
) ;
```

### Overview

Slider 의 range 를 설정한다.

### Parameter

EGL_HANDLE hObj	Range 를 설정 할 Slider handle.
int nMinPos	Min position 값. ( range = Max position – Min position ).
int nMaxPos	Max position 값. ( range = Max position – Min position).

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_range(slider, 0, 15);
```

## ► egl\_slider\_get\_range function

```
void egl_slider_get_range(  
    EGL_HANDLE hObj,  
    int* lpMinPos,  
    int* lpMaxPos  
)
```

### Overview

Slider 의 range 를 가져온다.

### Parameter

EGL_HANDLE hObj	Range 값을 가져 올 Slider handle.
int* lpMinPos	Min position 값을 저장 할 포인터.
int* lpMaxPos	Max position 값을 저장 할 포인터.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
int max_pos = 0;  
int min_pos = 0;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_get_range(slider, &min_pos, &max_pos);
```

## ► egl\_slider\_stepit function

```
void egl_slider_stepit(  
    EGL_HANDLE hObj,  
    BOOL inc  
)
```

### Overview

Slider 를 전후로 한 step 이동한다.

### Parameter

EGL_HANDLE hObj	Step 이동 할 Slider handle.
BOOL inc	Step 의 증가/감소를 결정. TRUE = 증가. FALSE = 감소.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_stepit(slider, TRUE);  
//egl_slider_stepit(slider, FALSE);
```

## ► egl\_slider\_set\_tick\_frequency function

```
void egl_slider_set_tick_frequency(  
    EGL_HANDLE hObj,  
    int freq  
)
```

### Overview

Slider 의 tick 빈도 수를 설정한다.

### Parameter

EGL_HANDLE hObj	Tick 빈도 수를 설정 할 Slider handle.
int freq	Tick 출력 빈도 수.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_tick_frequency(slider, 2);
```

## ► egl\_slider\_set\_tick\_style function

```
void egl_slider_set_tick_frequency(  
    EGL_HANDLE hObj,  
    TICKSTYLE style  
)
```

### Overview

Slider 의 tick style 을 설정한다.

### Parameter

EGL_HANDLE hObj	Tick style 을 설정 할 Slider handle.
TICKSTYLE style	<p>TICK style.</p> <p>TICK_NONE : 눈금 표시 없음.</p> <p>TICK_TOPLEFT : 눈금 표시가 위쪽 또는 좌측에 표시.</p> <p>TICK_BOTTOMRIGHT : 눈금 표시가 아래쪽 또는 우측에 표시.</p> <p>TICK_BOTH : 눈금 표시가 양쪽에 표시.</p>

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_tick_style(slider, TICK_BOTTOMRIGHT);
```

## ► egl\_slider\_set\_thumb\_size function

```
void egl_slider_set_thumb_size(  
    EGL_HANDLE hObj,  
    int width,  
    int height  
)
```

### Overview

Slider 의 thumb size 를 설정한다.

### Parameter

EGL_HANDLE hObj	Thumb size 를 설정 할 Slider handle.
int width	Thumb 의 가로 크기.
int height	Thumb 의 세로 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_thumb_size(slider, 10, 30);
```

## ► egl\_slider\_get\_thumb\_size function

```
void egl_slider_get_thumb_size(  
    EGL_HANDLE hObj,  
    int* w,  
    int* h  
)
```

### Overview

Slider 의 thumb size 를 가져온다.

### Parameter

EGL_HANDLE hObj	Thumb size 를 가져 올 Slider handle.
int* w	Thumb 가로 크기를 저장할 포인터.
int* h	Thumb 세로 크기를 저장할 포인터.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
int w = 0;  
int h = 0;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_get_thumb_size(slider, &w, &h);
```

## ► egl\_slider\_set\_barcolor function

```
void egl_slider_set_barcolor(  
    EGL_HANDLE hObj,  
    EGL_COLOR clr  
)
```

### Overview

Slider 의 bar color 를 설정한다.

### Parameter

EGL_HANDLE hObj	Bar color 를 설정 할 Slider handle.
EGL_COLOR clr	Color Value.
	MAKE_COLORREF(r,g,b) 매크로를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_barcolor(slider, MAKE_COLORREF( 0, 0xff, 0));
```

## ► egl\_slider\_set\_tickcolor function

```
void egl_slider_set_tickcolor(  
    EGL_HANDLE hObj,  
    EGL_COLOR clr  
)
```

### Overview

Slider 의 tick color 를 설정한다.

### Parameter

EGL_HANDLE hObj	Tick color 를 설정 할 Slider handle.
EGL_COLOR clr	Color Value. MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_tickcolor(slider, MAKE_COLORREF(0xff, 0, 0));
```

## ► egl\_slider\_set\_transparent function

```
void egl_slider_set_transparent(  
    EGL_HANDLE hObj,  
    BOOL bflag  
)
```

### Overview

Slider 의 배경을 투명하게 할지를 결정한다.

### Parameter

EGL_HANDLE hObj	배경을 투명하게 설정 할 Slider handle.
BOOL bflag	TRUE => 배경 투명. FALSE => 배경 불투명.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_transparent(slider, TRUE);
```

## ► egl\_release\_slider function

```
    BOOL egl_release_slider(  
        EGL_HANDLE hObj  
    );
```

## Overview

Slider 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

**TRUE or FALSE.**

## Example

```
if( slider != NULL)
{
    egl_window_delete_object( hWin, slider );
    egl_release_slider( slider );
    slider = NULL;
}
```

## ► Slider Example.

### Example

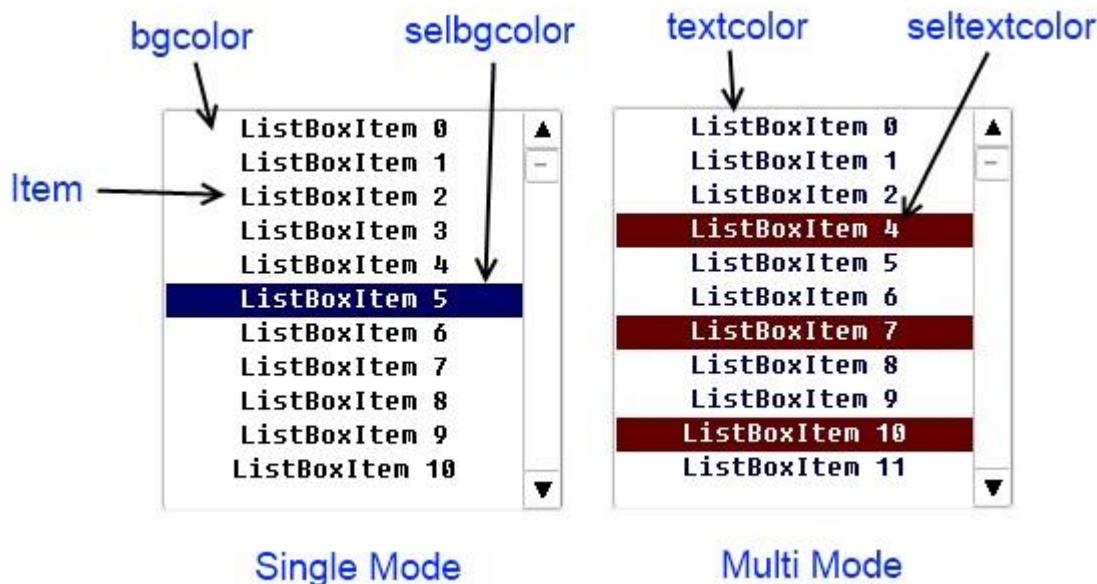
```
EGL_HANDLE slider;
void slider_callback(EGL_HANDLE h, int event)
{
    if(event == SLR_CLICKED)
    {
        debugprintf("position = %d\n", egl_slider_get_pos());
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    egl_init();
    slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);
    egl_sliderl_callback(slider, slider_callback);
    egl_window_add_object(hWin, slider);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# List Box



Function	Description
<a href="#">egl_create_listbox</a>	List box 를 생성한다.
<a href="#">egl_listbox_callback</a>	List box 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_listbox_additem</a>	List box 에 Item 을 추가한다.
<a href="#">egl_listbox_delitem</a>	List box 의 마지막 Item 을 제거한다.
<a href="#">egl_listbox_delitem_text</a>	List box 에서 text 에 해당되는 Item 을 제거한다.
<a href="#">egl_listbox_delitem_num</a>	List box 에서 num 번째 Item 을 제거한다.
<a href="#">egl_listbox_alldelitem</a>	List box 의 모든 Item 을 제거한다.
<a href="#">egl_listbox_get_all_itemlist</a>	List box 의 모든 Item 들의 text 목록을 반환한다.
<a href="#">egl_listbox_get_sel_item</a>	List box 가 Single Mode 일 때 선택된 Item 의 text 를 반환한다.
<a href="#">egl_listbox_get_multiple_sel_itemlist</a>	List box 가 Multiple Mode 일 때 선택된 Item 들의 text 목록을 반환한다.
<a href="#">egl_listbox_set_bgcolor</a>	List box 의 배경색을 지정한다.

[egl\\_listbox\\_set\\_selbgcolor](#) List box에서 선택된 Item의 배경색을 지정한다.

[egl\\_listbox\\_set\\_textcolor](#) List box의 글자 색을 지정한다.

[egl\\_listbox\\_set\\_selextcolor](#) List box에서 선택된 item의 글자 색을 지정한다.

[egl\\_listbox\\_set\\_textalign](#) List box의 글자 align을 설정한다.

[egl\\_listbox\\_set\\_scrollwidth](#) List box의 scrollbar의 가로 크기를 설정한다.

[egl\\_listbox\\_change\\_item\\_text](#) List box에서 text에 해당하는 item명을 변경한다.

[egl\\_listbox\\_change\\_item\\_num](#) List box에서 num번째 item명을 변경한다.

[egl\\_release\\_listbox](#) List box를 제거한다.

### Define

```
LIST_EVENT           typedef enum
{
    LIST_CHANGED = 0,
} LIST_EVENT;
```

## ► egl\_create\_listbox function

```
EGL_HANDLE egl_create_listbox(  
    int x,  
    int y,  
    int w,  
    int h,  
    bool bMultiple  
)
```

### Overview

List box 를 생성한다.

### Parameter

int x	생성될 List box 의 x 좌표.
int y	생성될 List box 의 y 좌표.
int w	생성될 List box 의 가로 크기.
int h	생성될 List box 의 세로 크기.
Bool bMultiple	생성될 List box 에서 복수 item 선택 여부를 설정한다.

### Return Value

생성된 List box 의 pointer

### Example

```
EGL_HANDLE listbox[2];  
  
listbox [0] = egl_create_button (100, 100, 200, 200, TRUE);  
  
listbox [1] = egl_create_button (400, 100, 200, 200, FALSE);
```

## ► egl\_listbox\_callback function

```
void egl_listbox_evnet_callback (
    EGL_HANDLE hObj,
    EVENT_CALLBACK cb
);
```

### Overview

List box 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	이벤트가 발생되는 List box 의 handle
EVENT_CALLBACK cb	이벤트가 발생했을 때 호출될 callback 함수

### Return Value

없음.

### Example

```
static void listbox_callback(EGL_HANDLE h, int event)
{
    if(event == LIST_CHANGED)
        debugprintf("LIST_CHANGED\r\n");
}

void main(void)
{
    EGL_HANDLE listbox;
    ...
    egl_listbox_callback(listbox, listbox_callback);
}
```

## ► egl\_listbox\_additem function

```
void egl_listbox_additem (
    EGL_HANDLE hObj,
    const char* text
```

```
 );
```

## Overview

List box 에 Item 을 추가한다.

## Parameter

EGL_HANDLE hObj	Item 을 추가할 List box 의 handle
const char* text	추가할 Item 의 text

## Return Value

없음.

## Example

```
EGL_HANDLE listbox;  
  
egl_listbox_additem(listbox, "ListboxItem 1");
```

## ► egl\_listbox\_delitem function

```
void egl_listbox_delitem (
    EGL_HANDLE hObj
);
```

### Overview

List box 의 마지막 Item 을 제거한다.

### Parameter

EGL\_HANDLE hObj                      Item 을 제거할 List box 의 handle

### Return Value

없음.

### Example

```
EGL_HANDLE listbox;

egl_listbox_delitem(listbox);
```

## ► egl\_listbox\_delitem\_text function

```
BOOL egl_listbox_delitem_text (
    EGL_HANDLE hObj,
    const char* text
);
```

### Overview

List box에서 text에 해당되는 Item을 제거한다.

### Parameter

EGL_HANDLE hObj	Item을 제거할 List box의 handle
const char* text	제거할 item의 text.

### Return Value

TRUE or FALSE.

### Example

```
EGL_HANDLE listbox;

egl_listbox_delitem_text(listbox, "item1");
```

## ► egl\_listbox\_delitem\_num function

```
BOOL egl_listbox_delitem_num (
    EGL_HANDLE hObj,
    int num
);
```

### Overview

List box에서 num 번째 Item 을 제거한다.

### Parameter

EGL_HANDLE hObj	Item 을 제거할 List box 의 handle
int num	제거할 item number.

### Return Value

TRUE or FALSE.

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_delitem_num(listbox, 3);
```

## ► egl\_listbox\_alldelitem function

```
void egl_listbox_alldelitem (
    EGL_HANDLE hObj,
);
```

### Overview

List box 의 모든 Item 을 제거한다.

### Parameter

EGL_HANDLE hObj	Item 을 제거할 List box 의 handle
-----------------	------------------------------

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_alldelitem(listbox);
```

## ► egl\_listbox\_get\_all\_itemlist function

```
const char** egl_listbox_get_all_itemlist (
    EGL_HANDLE hObj,
    int* itemcnt
);
```

### Overview

List box 의 모든 Item 들의 text 목록을 반환한다.

### Parameter

EGL_HANDLE hObj	Item 의 text 목록을 반환 받을 List box 의 handle
int* itemcnt	List box 의 Item 개수를 반환 받을 변수의 pointer

### Return Value

모든 Item 의 text list pointer

※ 주의 : 반드시 return 받은 list pointer 의 메모리를 해제하여야 한다.

### Example

```
EGL_HANDLE listbox;
int ItemCnt;
const char** ItemTextList;

ItemTextList = egl_listbox_all_itemlist(listbox, &ItemCnt);

free(ItemTextList);
```

## ► egl\_listbox\_get\_sel\_item function

```
const char* egl_listbox_get_sel_item (
    EGL_HANDLE hObj,
    int* index
);
```

### Overview

List box 가 Single Mode 일 때 선택된 Item 의 Text 를 반환한다

### Parameter

EGL_HANDLE hObj	선택된 Item 의 text 를 반환 받을 List box 의 handle
Int* index	선택된 Item 의 순번을 반환 받을 변수의 pointer

### Return Value

선택된 Item 의 text pointer

### Example

```
EGL_HANDLE listbox;
int selectItemIndex;
const char* selectItemText;

selectItemText = egl_listbox_get_sel_item(listbox, &selectItem);
```

## ► egl\_listbox\_get\_multiple\_sel\_itemlist function

```
const char** egl_listbox_get_multiple_sel_itemlist (
    EGL_HANDLE hObj,
    int* selcnt
);
```

### Overview

List box 가 Multiple Mode 일 때 선택된 Item 의 text 들의 목록을 반환한다.

### Parameter

EGL_HANDLE hObj	선택된 Item 들의 text 를 반환 받을 List box 의 handle
int* selcnt	선택된 Item 의 개수를 반환 받을 변수의 pointer

### Return Value

선택된 Item 들의 text list pointer

※ 주의 : 반드시 return 받은 list pointer 의 메모리를 해제하여야 한다.

### Example

```
EGL_HANDLE listbox;
int selectItemCnt;
const char** selectItemTextList;

selectItemTextList = egl_listbox_get_sel_itemlist (listbox, & selectItemCnt);

free(selectItemTextList);
```

## ► egl\_listbox\_set\_bgcolor function

```
void egl_listbox_set_bgcolor (
    EGL_HANDLE hObj,
    EGL_COLOR clr
);
```

### Overview

List box 의 배경색을 지정한다.

### Parameter

EGL_HANDLE hObj	배경색을 지정할 List box 의 handle
EGL_COLOR clr	Color value. MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음

### Example

```
EGL_HANDLE listbox;

egl_listbox_set_bgcolor(listbox, MAKE_COLORREF(0x0, 0x0, 0xFF)); // Blue
```

## ► egl\_listbox\_set\_selbgcolor function

```
void egl_listbox_set_selbgcolor (
    EGL_HANDLE hObj,
    EGL_COLOR clr
);
```

### Overview

List box에서 선택된 Item의 배경색을 지정한다.

### Parameter

EGL_HANDLE hObj	선택된 Item의 배경색을 지정할 List box의 handle
EGL_COLOR clr	Color value. MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음

### Example

```
EGL_HANDLE listbox;

egl_listbox_set_selbgcolor(listbox, MAKE_COLORREF(0xFF, 0x0, 0x0)); // Red
```

## ► egl\_listbox\_set\_textcolor function

```
void egl_listbox_set_textcolor (
    EGL_HANDLE hObj,
    EGL_COLOR clr
);
```

### Overview

List box 의 글자 색을 지정한다.

### Parameter

EGL_HANDLE hObj	글자 색을 지정할 List box 의 handle
EGL_COLOR clr	Color value. MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음

### Example

```
EGL_HANDLE listbox;

egl_listbox_set_textcolor(listbox, MAKE_COLORREF(0x0, 0x0, 0x0)); // Black
```

## ► egl\_listbox\_set\_seltextcolor function

```
void egl_listbox_set_seltextcolor (
    EGL_HANDLE hObj,
    EGL_COLOR clr
);
```

### Overview

List box에서 선택된 Item의 글자 색을 지정한다.

### Parameter

EGL_HANDLE hObj	선택된 Item의 글자 색을 지정할 List box의 handle
EGL_COLOR clr	Color value. MAKE_COLORREF(r,g,b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음

### Example

```
EGL_HANDLE listbox;

egl_listbox_set_seltextcolor(listbox, MAKE_COLORREF(0xFF, 0xFF, 0xFF)); // White
```

## ► egl\_listbox\_set\_textalign function

```
void egl_listbox_set_textalign (
    EGL_HANDLE hObj,
    int align
);
```

### Overview

List box 의 item align 을 설정한다.

### Parameter

EGL_HANDLE hObj	Align 을 설정 할 List box 의 handle
int align	Align value. EGL_ALIGN_LEFT / EGL_ALIGN_RIGHT / EGL_ALIGN_CENTER EGL_ALIGN_TOP/ EGL_ALIGN_BOTTOM

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_textalign(listbox, EGL_ALIGN_LEFT );
```

## ► egl\_listbox\_set\_scrollwidth function

```
void egl_listbox_set_scrollwidth (
    EGL_HANDLE hObj,
    int width
);
```

### Overview

List box 의 scrollbar 의 가로 크기를 설정한다.

### Parameter

EGL_HANDLE hObj	Scrollbar 의 가로크기를 설정 할 List box 의 handle
int width	Scrollbar 의 가로 크기.

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_scrollwidth(listbox, 40);
```

## ► egl\_listbox\_change\_item\_text function

```
BOOL egl_listbox_change_item_text (
    EGL_HANDLE hObj,
    const char* text,
    const char* changetext
);
```

### Overview

List box에서 text에 해당하는 item 명을 변경한다.

### Parameter

EGL_HANDLE hObj	Item 명을 변경 할 List box 의 handle
const char* text	Item text.
const char* changetext	변경 할 Item text.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE listbox;

egl_listbox_change_item_text(listbox, "test1", "test2");
```

## ► egl\_listbox\_change\_item\_num function

```
BOOL egl_listbox_change_item_num (
    EGL_HANDLE hObj,
    int num,
    const char* changetext
);
```

### Overview

List box에서 num 번째 item 명을 변경한다.

### Parameter

EGL_HANDLE hObj	Item 명을 변경 할 List box 의 handle
int num	Item number.
const char* changetext	변경 할 Item text.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE listbox;

egl_listbox_change_item_num(listbox, 3, "test2");
```

## ► egl\_release\_listbox function

```
BOOL egl_listbox_set_scrollwidth (
    EGL_HANDLE hObj
);
```

### Overview

List box 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

### Parameter

EGL_HANDLE hObj	제거 할 List box 의 handle
-----------------	------------------------

### Return Value

TRUE or FALSE

### Example

```
if( listbox != NULL)
{
    egl_window_delete_object( hWin, listbox );
    egl_release_listbox( listbox );
    listbox = NULL;
}
```

## ► List box Example.

### Example

```
#include "adStar-L.h"
extern BOOL process_touch(BOOL* touchdown,EGL_POINT* pPoint);

static void listbox_cb1(EGL_HANDLE h, int event)
{
    int itemcnt, index;
    const char **itemlist;
    const char *selitem;
    int i;

    if(event == LIST_CHANGED) {
        debugprintf("LIST_CHANGED\r\n");
        // Get all item list
        itemlist = egl_listbox_get_all_itemlist(h, &itemcnt);
        debugprintf("All Item %d\r\n", itemcnt);
        for(i = 0; i < itemcnt; i++)
            debugprintf("Item : [%s]\r\n", itemlist[i]);
        // Get selected item list
        selitem = egl_listbox_get_sel_item(h, &index);
        debugprintf("Selected Item [%d] : [%s]\r\n", index, selitem);
    }
    free(itemlist);
}

static void listbox_cb2(EGL_HANDLE h, int event)
{
    int selcnt;
    const char **selitemlist;
    int i;

    if(event == LIST_CHANGED) {
        debugprintf("LIST_CHANGED\r\n");
        selitemlist = egl_listbox_get_multiple_sel_itemlist(h, &selcnt);
        debugprintf("Selected Items %d \r\n", selcnt);
        for(i = 0; i < selcnt; i++)
            debugprintf("Selected Item : [%s]\r\n", selitemlist[i]);
    }
}
```

```
free(selitemlist);

}

void user_main()
{
    float addpos=1.0f;
    EGL_POINT touch_pt;
    BOOL touchdown=FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE listbox[2];

    egl_init();

    hWin = egl_create_window("Main Window");

    listbox[0] = egl_create_listbox(100, 100, 200, 200, FALSE);
    listbox[1] = egl_create_listbox(400, 100, 200, 200, TRUE);

    egl_listbox_callback(listbox[0], listbox_cb1);
    egl_listbox_callback(listbox[1], listbox_cb2);

    {
        int i;
        char str_text[16];
        for(i=0;i<100;i++)
        {
            sprintf(str_text,"ListBoxItem %d",i);
            egl_listbox_additem(listbox[0],str_text);
            egl_listbox_additem(listbox[1],str_text);
        }
    }

    // Delete last item
    egl_listbox_delitem(listbox[0]);
    // Delete "ListBoxItem 3" item
    egl_listbox_delitem_text(listbox[1], "ListBoxItem 3");

    // Set background color
    egl_listbox_set_bgcolor(listbox[1], 0xff, 0xff, 0xff);
    // Set selected background color
    egl_listbox_set_selbgcolor(listbox[1], 0x66, 0x0, 0x0);
    // Set text color
    egl_listbox_set_textcolor(listbox[1], 0, 0, 0x66);
```

```
// Set selected text color
egl_listbox_set_seltextrcolor(listbox[1], 0xff, 0xff, 0xff);

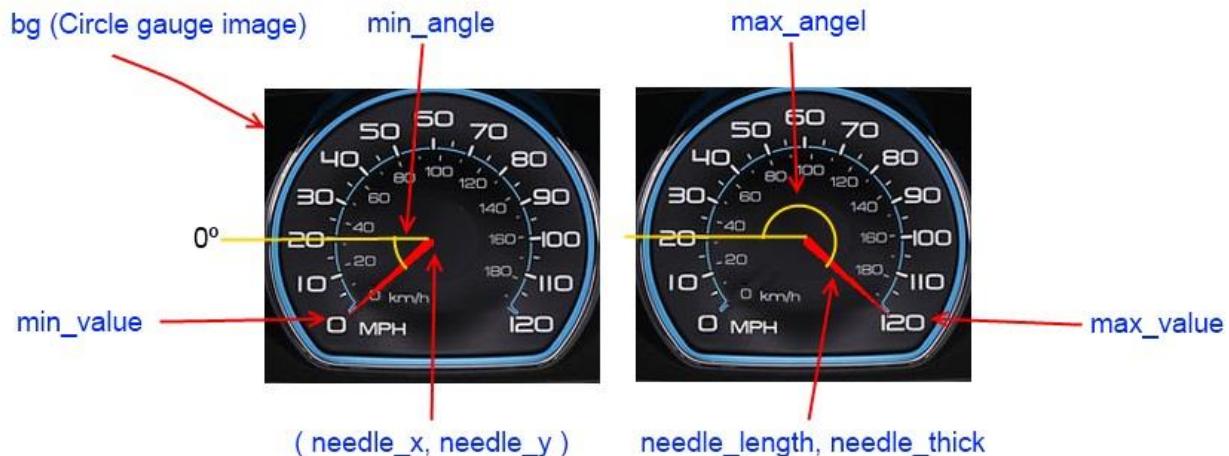
egl_window_add_object(hWin, listbox[0]);
egl_window_add_object(hWin, listbox[1]);

egl_window_show(hWin, TRUE);
egl_draw();

while(1)
{
    BOOL bEvent = TRUE;

    if(process_touch(&touchdown, &touch_pt))
        egl_user_touch_input(touchdown, &touch_pt);
    else
        bEvent = FALSE;
    egl_draw();
}
}
```

## Circle Gauge



Function	Description
<a href="#"><u>egl_create_circle_gauge</u></a>	Circle gauge 를 생성한다.
<a href="#"><u>egl_circle_gauge_set_value</u></a>	Circle gauge 값을 설정한다.
<a href="#"><u>egl_circle_gauge_get_value</u></a>	Circle gauge 값을 가져온다.
<a href="#"><u>egl_release_circle_gauge</u></a>	Circle gauge 를 제거한다.

## ► egl\_create\_circle\_gauge function

```
EGL_HANDLE egl_create_circle_gauge(
    SURFACE* bg,
    int x,
    int y,
    EGL_CIRCLE_GAUGE_INFO* pInfo
);
```

### Overview

Circle gauge 를 생성한다.

### Parameter

SURFACE* bg	Circle gauge image.
int x	Circle gauge x 좌표.
int y	Circle gauge y 좌표.
EGL_CIRCLE_GAUGE_INFO* pInfo	Circle gauge 정보.

```
typedef struct _tagCIRCLE_GAUGE_INFO
{
    int needle_x;      // gauge needle x 좌표.
    int needle_y;      // gauge needle y 좌표.
    int needle_length; // gauge needle 길이.
    int needle_thick;  // gauge needle 두께.
    int min_value;     // gauge 의 최소 값.
    int max_value;     // gauge 의 최대 값.
    int min_angle;     // gauge 의 최소 각.
    int max_angle;     // gauge 의 최대 각.
}EGL_CIRCLE_GAUGE_INFO;
```

### Return Value

생성 된 circle gauge handle

### Example

```
SURFACE* gauge_bg;  
EGL_HANDLE c_gauge;  
EGL_CIRCLE_GAUGE_INFO CGInfo;  
CGInfo.needle_x = 113;  
CGInfo.needle_y = 101;  
CGInfo.min_angle = -42;  
CGInfo.max_angle = 222;  
CGInfo.min_valie = 0;  
CGInfo.max_value = 120;  
CGInfo.needle_length = 80;  
CGInfo.needle_thick = 3;  
gauge_bg = loadbmp("gauge.bmp");  
c_gauge = egl_create_circle_gauge(gauge_bg, 286, 37, &CGInfo);
```

## ► egl\_circle\_gauge\_set\_value function

```
BOOL egl_circle_gauge_set_value(  
    EGL_HANDLE h,  
    int value  
)
```

### Overview

Circle gauge 값을 설정한다.

### Parameter

EGL_HANDLE h	Gauge 값을 설정 할 circle gauge handle.
int value	설정 gauge 값.

### Return Value

TRUE or FALSE

### Example

```
...  
c_gauge = egl_create_circle_gauge(gauge_bg, 286,37,&CGInfo);  
egl_circle_gauge_set_value(c_gauge, 80);  
...
```

## ► egl\_circle\_gauge\_get\_value function

```
int egl_circle_gauge_get_value(  
    EGL_HANDLE h  
)
```

### Overview

Circle gauge 값을 가져온다.

### Parameter

EGL\_HANDLE h                      Gauge 값을 가져 올 circle gauge handle.

### Return Value

Gauge 값.

### Example

```
...  
c_gauge = egl_create_circle_gauge(gauge_bg, 286,37,&CGInfo);  
int gauge_value = egl_circle_gauge_get_value(c_gauge);  
...
```

## ► **egl\_release\_circle\_gauge** function

```
BOOL egl_release_circle_gauge(  
    EGL_HANDLE hObj  
) ;
```

## Overview

Circle gauge 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

## Parameter

## Return Value

## TRUE or FALSE.

## Example

```
If( circle != NULL)
{
    egl_window_delete_object( hWin, circle );
    egl_release_circle( circle );
    circle = NULL;
}
```

## ► Circle Gauge Example.

### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

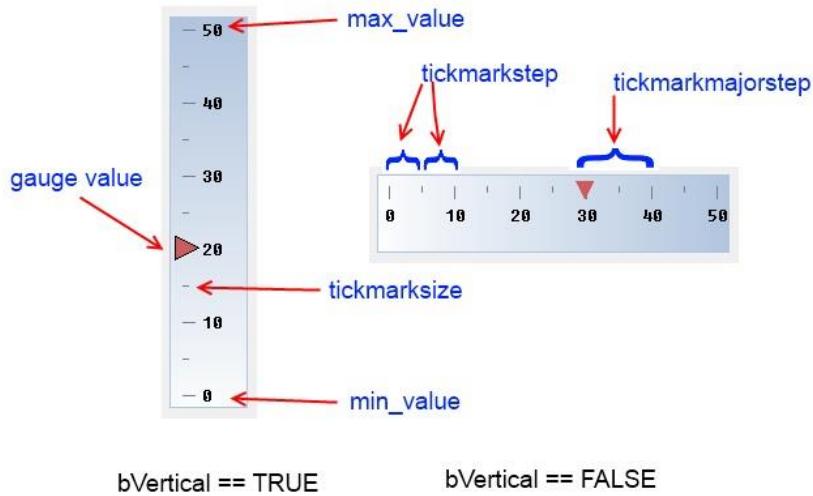
int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    SURFACE* gauge_bg;
    EGL_HANDLE c_gauge;
    EGL_CIRCLE_GAUGE_INFO CGInfo;

    egl_init();
    CGInfo.needle_x = 113;
    CGInfo.needle_y = 101;
    CGInfo.min_angle = -42;
    CGInfo.max_angle = 222;
    CGInfo.min_valie = 0;
    CGInfo.max_value = 120;
    CGInfo.needle_length = 80;
    CGInfo.needle_thick = 3;
    gauge_bg = loadbmp("gauge.bmp");
    c_gauge = egl_create_circle_gauge(gauge_bg, 286, 37, &CGInfo);
    egl_window_add_object(hWin, c_gauge);
    egl_window_show(hWin);
    egl_draw();
    int gauge_value = 0;
    int gauge_flag = 0;
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        if(gauge_flag)
        {
            gauge_value++;
            if(gauge_value > 120)
```

```
{  
    gauge_value = 120;  
    gauage_flag = 0;  
}  
}  
else  
{  
    gauge_value--;  
    if(gauge_value < 0)  
    {  
        gauge_value = 0;  
        gauge_flag = 1;  
    }  
}  
egl_circle_gauge_set_valut(c_gauge, gauge_value);  
egl_draw();  
}  
}
```

## Bar Gauge



Function	Description
<a href="#"><u>egl_create_bar_gauge</u></a>	bar gauge 를 생성한다.
<a href="#"><u>egl_bar_gauge_set_value</u></a>	bar gauge 값을 설정한다.
<a href="#"><u>egl_bar_gauge_get_value</u></a>	bar gauge 값을 가져온다.
<a href="#"><u>egl_release_bar_gauge</u></a>	bar gauge 를 제거한다.

## ► egl\_create\_bar\_gauge function

```
EGL_HANDLE egl_create_bar_gauge(
    int x,
    int y,
    int w,
    int h,
    EGL_BAR_GAUGE_INFO* pInfo
);
```

### Overview

bar gauge 를 생성한다.

### Parameter

int x	Bar gauge x 좌표.
int y	Bar gauge y 좌표.
int w	Bar gauge 가로 크기.
int h	Bar gauge 세로 크기.
EGL_BAR_GAUGE_INFO* pInfo	Bar gauge 정보

```
typedef struct _tagBAR_GAUGE_INFO
{
    int min_value;           // gauge 최소 값.
    int max_value;           // gauge 최대 값.
    int tickmarksize;        // tick size. major tick = tick size * 2
    int tickmarkstep;         // tick step size.
    int tickmarkmajorstep;   // major tick step size.
    BOOL bVertical;          // vertical mode select.
    SURFACE* bg;             // back ground image.
} EGL_BAR_GAUGE_INFO;
```

### Return Value

생성 된 bar gauge handle.

### Example

```
EGL_HANDLE b_gauge;  
EGL_BAR_GAUGE_INFO BGInfo;  
BGInfo.min_value = 0;  
BGInfo.max_value = 50;  
BGInfo.bVertical = TRUE;  
BGInfo.tickmarksize = 5;  
BGInfo.tickmarkstep = 5;  
BGInfo.tickmarkmajorstep = 10;  
BGInfo.bg = NULL;  
b_gauge = egl_create_bar_gauge(10, 40, 60, 300, &BGInfo);
```

## ► egl\_bar\_gauge\_set\_value function

```
BOOL egl_bar_gauge_set_value(  
    EGL_HANDLE h,  
    int value  
)
```

### Overview

bar gauge 값을 설정한다.

### Parameter

EGL_HANDLE h	값을 설정 할 bar gauge handle.
int value	설정 gauge 값.

### Return Value

TRUE or FALSE

### Example

```
...  
b_gauge = egl_create_bar_gauge(10, 40, 60, 300, &BGInfo);  
egl_bar_gauge_set_value(b_gauge, 20);  
...
```

## ► egl\_bar\_gauge\_get\_value function

```
int egl_bar_gauge_get_value(  
    EGL_HANDLE h  
)
```

### Overview

bar gauge 값을 가져온다.

### Parameter

EGL\_HANDLE h                      Gauge 값을 가져 올 bar gauge handle.

### Return Value

Gauge 값

### Example

```
...  
b_gauge = egl_create_bar_gauge(10, 40, 60, 300, &BGInfo);  
int gauge = egl_bar_gauge_get_value(b_gauge);  
...
```

## ► egl\_release\_bar\_gauge function

```
BOOL egl_release_bar_gauge(  
    EGL_HANDLE hObj  
)
```

### Overview

bar gauge 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

### Parameter

EGL\_HANDLE hObj                  제거 할 bar gauge handle.

### Return Value

TRUE or FALSE

### Example

```
if( bar_gauge != NULL)  
{  
    egl_windwo_delete_object( hWin, bar_gauge );  
    egl_release_bar_gauge( bar_gauge );  
    bar_gauge = NULL;  
}
```

## ► Bar Gauge Example.

### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE b_gauge;
    EGL_BAR_GAUGE_INFO BGInfo;
    egl_init();
    BGInfo.min_value = 0;
    BGInfo.max_value = 50;
    BGInfo.bVertical = TRUE;
    BGInfo.tickmarkszie = 5;
    BGInfo.tickmarkstep = 5;
    BGInfo.tickmarkmajorstep = 10;
    BGInfo.bg = NULL;
    b_gauge = egl_create_bar_gauge(10,40,60,300,&BGInfo);
    egl_window_add_object(hWin, b_gauge);
    egl_window_show(hWin);
    egl_draw();
    int gauge_value = 0;
    int gauge_flag = 0;
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        if(gauge_flag)
        {
            gauge_value++;
            if(gauge_value > 50)
            {
                gauge_value = 50;
                gauge_flag = 0;
            }
        }
    }
}
```

```
    }
}
else
{
    gauge_value--;
    if(gauge_value < 0)
    {
        gauge_value = 0;
        gauge_flag = 1;
    }
}
egl_bar_gauge_set_valut(b_gauge, gauge_value);
egl_draw();
}
}
```

# Picture



Function	Description
<a href="#"><u>egl_create_picture</u></a>	Picture object 를 생성한다.
<a href="#"><u>egl_picture_callback</u></a>	Picture 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#"><u>egl_picture_set</u></a>	Picture object 의 image 를 변경한다.
<a href="#"><u>egl_release_picture</u></a>	Picture 를 제거한다.

## ► egl\_create\_picture function

```
EGL_HANDLE egl_create_picture (
    SURFACE* surf,
    int x,
    int y,
    int w,
    int h
);
```

### Overview

Picture object 를 생성한다.

### Parameter

SURFACE* surf	Picture 에 표시할 image surface.
int x	Picture x 좌표.
int y	Picture y 좌표.
int w	Picture 가로 크기.
int h	Picture 세로 크기.

### Return Value

생성 된 Picture object handle.

### Example

```
SURFACE* image = loadbmp("image.bmp");
EGL_HANDLE picture;
picture = egl_create_picture(image,50, 50, 128, 128);
```

## ► egl\_picture\_callback function

```
BOOL egl_picture_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
)
```

### Overview

Picture 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	이벤트가 발생되는 Picture 의 handle.
EVENT_CALLBACK cb	이벤트가 발생했을 때 호출 될 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
static void picture_callback(EGL_HANDLE h, int event)  
{  
    if(event == PICTURE_CLICKED)  
        debugprintf("picture clicked\r\n");  
}  
  
void main(void)  
{  
    EGL_HANDLE picture;  
    ...  
    egl_picture_callback(picture, picture_callback);  
}
```

## ► egl\_picture\_set function

```
SURFACE* egl_picture_set(  
    EGL_HANDLE hObj,  
    SURFACE* surf  
)
```

### Overview

Picture image 를 설정한다.

### Parameter

EGL_HANDLE hObj	Image 를 변경 할 Picture handle.
SURFACE* surf	변경 할 image.

### Return Value

설정 되어 있던 이미지 SURFACE.

### Example

```
...  
SURFACE* oldimage;  
SURFACE* image2 = loadbmp("image2.bmp");  
oldimage = egl_picture_set(picture, image2);  
...
```

## ▶ egl\_release\_picture function

```
BOOL egl_release_picture(
    EGL_HANDLE hObj,
);
```

### Overview

Picture 를 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

### Parameter

EGL_HANDLE hObj	제거 할 Picture handle.
-----------------	----------------------

### Return Value

TRUE or FALSE

### Example

```
if( picture != NULL)
{
    egl_window_delete_object( hWin, picture );
    egl_release_picture( picture );
    picture = NULL;
}
```

## ► Picture Example.

### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    egl_init();
    ...
    EGL_HANDLE picture;
    SURFACE* image = loadbmp("image.bmp");
    picture = egl_create_picture(image,10,40,60,300);
    egl_window_add_object(hWin, picture);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Animation

Function	Description
<a href="#"><u>egl_create_animation</u></a>	Animation object 를 생성한다.
<a href="#"><u>egl_release_animation</u></a>	Animation 을 제거한다.

## ► egl\_create\_animation function

```
EGL_HANDLE egl_create_animation(  
    int x,  
    int y,  
    int w,  
    int h,  
    SURFACE** surflist,  
    int surfcnt,  
    int delaycnt  
) ;
```

### Overview

Animation object 를 생성한다. Animation object 는 연속 이미지를 번갈아 가며 draw 한다.

### Parameter

int x	Animation object x 좌표.
int y	Animation object y 좌표.
int w	Animation object 가로 크기.
int h	Animation object 세로 크기.
SURFACE** surflist	Animation image list.
int surfcnt	Animation image count.
int delaycnt	Animation image 전환 delay.

### Return Value

생성 된 animation object handle.

### Example

```
EGL_HANDLE btn_ani;  
SURFACE* surf_ani[10];  
char fname[12];  
int i;  
for(i=0;i<10;i++)  
{  
    sprintf(fname, "Frame%d.bmp",i);  
    surf_ani[i] = loadbmp(fname);  
}  
btn_ani = egl_create_animation(350, 150, 128, 128, surf_ani, 10, 0);
```

## ► egl\_release\_animation function

```
BOOL egl_release_animation(
    EGL_HANDLE hObj
)
```

### Overview

Animation 을 제거한다. 해당 object 가 window 에 add 되어 있는 경우, egl\_window\_delete\_object 함수로 window 에서 object 를 제거한 후 release 해주어야 한다.

### Parameter

EGL_HANDLE hObj	제거 할 animation handle.
-----------------	------------------------

### Return Value

TRUE or FALSE

### Example

```
if( animation != NULL )
{
    egl_window_delete_object( hWin, animation );
    egl_release_animation( animation );
    animation = NULL;
}
```

## ► Animation Example.

### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn_ani;
    SURFACE* surf_ani[10];
    char fname[12];
    egl_init();
    int i;
    for(i=0;i<10;i++)
    {
        sprintf(fname, "Frame%d.bmp",i);
        surf_ani[i] = loadbmp(fname);
    }
    btn_ani = egl_create_animation(350, 150, 128, 128, surf_ani, 10, 0);
    egl_window_add_object(hWin, btn_ani);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Custom Object

Function	Description
----------	-------------

[egl\\_create\\_custom\\_object](#) Custom object 를 생성한다.

Define
--------

```
EGL_MSG_ID
typedef enum enumMSG
{
    EGL_MSG_DRAW = 0,
    EGL_MSG_DELETE,
    EGL_MSG_FOCUS,
    EGL_MSG_UNFOCUS,
    EGL_MSG_KEY_UP,
    EGL_MSG_KEY_DOWN,
    EGL_MSG_TOUCHED,
    EGL_MSG_UNTOUCHED,
    EGL_MSG_MOVE,
    EGL_MSG_TIMETICK
} EGL_MSG_ID;
```

```
EGL_MSG
typedef struct _tagMessage
{
    EGL_MSGID msgID;
    EGL_HANDLE hObj;
    EGL_HANDLE hWin;
    Union
    {
        EGL_POINT point;
        U32 key;
    } param;
} EGL_MSG;
```

## ► egl\_create\_custom\_object function

```
EGL_HANDLE egl_create_custom_object(  
    int x,  
    int y,  
    int w,  
    int h,  
    void* (*msg_handler)(EGL_MSG* pMsg)  
);
```

### Overview

Custom object 를 생성한다.

### Parameter

int x	Custom object x 좌표.
int y	Custom object y 좌표.
int w	Custom object 가로 크기.
int h	Custom object 세로 크기.
void* (*msg_handler)(EGL_MSG* pMsg)	Custom object msg 처리 함수.

### Return Value

생성 된 custom object handle.

### Example

```
Static void* custom_obj_msghandler(EGL_MSG* pMsg)  
{  
    Switch(pMsg->msgID)  
    {  
        Case EGL_MSG_DRAW:
```

```
    /* draw */
    Break;
}

Return NULL;

}

...

Int main()
{
...

EGL_HANDLE custom_obj;
custom_obj = egl_create_custom(350, 240, 100, 30, custom_obj_msghandler);
...

}
```

## ► Custom Example.

### Example

```
static char speed_str[16];
static EGL_HANDLE hSpeed;

static void speed_draw(EGL_OBJECT_PTR pObj)
{
    draw_text_in_box(pObj->pFont, &pObj->rect, speed_str, EGL_ALIGN_CENTER);
}

static void* speed_msghandler(EGL_MSG* pMsg)
{
    switch(pMsg->msgID)
    {
        case EGL_MSG_DRAW:
            speed_draw(EGL_HANDLE_TO_OBJECT(pMsg->hObj));
            break;
    }
}

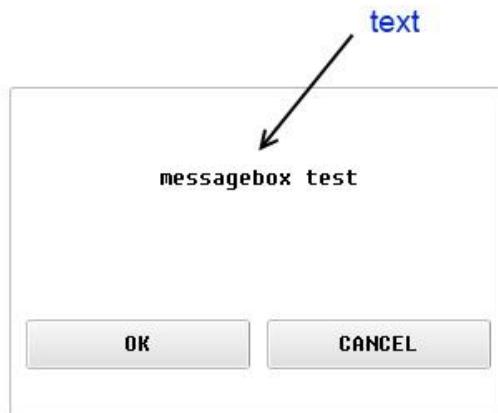
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE hSpeed;
    EGL_FONT* myfont;
    egl_init();
    myfont = egl_create_default_font();
    hSpeed = egl_create_custom_object(350, 240, 100, 30, speed_msghandler);
    egl_window_add_object(hWin, hSpeed);
    egl_set_font(hSpeed, myfont);
    strcpy(speed_str, " 0 Mile/h");
    egl_window_show(hWin);
    egl_draw();
}
```

```
while(1)
{
    if( process_touch( &touchdown, &touch_pt ) )
        egl_user_touch_input( touchdown, &touch_pt );

    egl_draw();
}
```

## Messagebox



`flags == MB_OKCANCEL`

Function	Description
<a href="#"><u>egl_show_messagebox</u></a>	MessageBox 를 출력한다.

## ► egl\_show\_messagebox function

```
int egl_show_messagebox(
    const char* text,
    int flags,
    BOOL (*user_input_function)(EGL_MSG* pMsg)
);
```

### Overview

Messagebox 를 출력한다.

### Parameter

const char* text	Messagebox 에 출력할 문자.
int flags	Messagebox type. MB_OK MB_OKCANCEL MB_YESNO MB_YESNOCANCEL
BOOL (*user_input_function)(EGL_MSG* pMsg)	사용자 입력 함수. Messagebox 안의 어떤 버튼이 눌렸는지 확인을 위한 입력 함수.

### Return Value

Message box 안의 눌린 버튼의 type 값을 반환.

```
enum
{
    IDOK = 0,
    IDCANCEL,
    IDABORT,
    IDRETRY,
    IDIGNORE,
    IDYES,
    IDNO,
    IDCONTINUE,
}
```

### Example

```
BOOL my_touchinput(EGL_MSG* pMsg)
{
    BOOL touchdown;
    EGL_POINT touch_pt;
    if(process_touch(&touchdown, &touch_pt))
    {
        if(touchdown)
            pMsg -> msgID = EGL_MSG_TOUCHED;
        else
            pMsg -> msgID = EGL_MSG_UNTOUCHED;
        pMsg -> param.point.x = touch_pt.x;
        pMsg -> param.point.y = touch_pt_y;
        return TRUE;
    }
    return FALSE;
}

int main()
{
...
...
int type = egl_show_messagebox("messagebox test", MB_OKCANCEL, my_touchinput);
...
}
```

## ► MessageBox Example.

### Example

```
void btn_callback(EGL_HANDLE h, int event)
{
    if(event == BTN_CLICKED)
    {
        debugprintf("button clicked.");
    }
}

BOOL my_touchinput(EGL_MSG* pMsg)
{
    BOOL touchdown;
    EGL_POINT touch_pt;
    if(process_touch(&touchdown, &touch_pt))
    {
        if(touchdown)
            pMsg -> msgID = EGL_MSG_TOUCHED;
        else
            pMsg -> msgID = EGL_MSG_UNTOUCHED;
        pMsg -> param.point.x = touch_pt.x;
        pMsg -> param.point.y = touch_pt_y;
        return TRUE;
    }
    return FALSE;
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn;
    egl_init();
    btn = egl_create_button(100, 100, 100, 50, "Button Ex");
    egl_button_callback(btn, btn_callback);
```

```
egl_window_add_object(hWin, btn);
egl_window_show(hWin);
egl_draw();

int type = egl_show_messagebox("messagebox test", MB_OKCANCEL, my_touchinput);

while(1)
{
    if( process_touch( &touchdown, &touch_pt ) )
        egl_user_touch_input( touchdown, &touch_pt );

    egl_draw();
}
}
```

# EGL Font

Function	Description
<a href="#"><u>egl_get_font</u></a>	Object 가 사용하는 font 정보를 반환한다.
<a href="#"><u>egl_set_font</u></a>	Object 가 사용 할 font 를 설정한다.
<a href="#"><u>egl_font_set_bkmode</u></a>	Font 의 배경색 사용 여부를 결정한다.
<a href="#"><u>egl_font_get_bk_color</u></a>	Font 의 배경색 정보를 가져온다.
<a href="#"><u>egl_font_set_bk_color</u></a>	Font 의 배경색을 설정한다.
<a href="#"><u>egl_font_get_color</u></a>	Font 의 색 정보를 가져온다.
<a href="#"><u>egl_font_set_color</u></a>	Font 의 색을 설정한다.
<a href="#"><u>create_bitfont</u></a>	Font 사용을 위해 Bit 방식의 font 를 생성한다.
<a href="#"><u>release_bitfont</u></a>	Create 한 bit 방식의 font 를 release 한다.
<a href="#"><u>create_bmpfont</u></a>	Font 사용을 위해 이미지 font 를 생성한다.
<a href="#"><u>bmfont_release</u></a>	Create 한 이미지 font 를 release 한다.
<a href="#"><u>draw_text</u></a>	문자를 출력한다.
<a href="#"><u>draw_text_pivot</u></a>	문자를 좌/우 90 도 회전하여 출력한다.
<a href="#"><u>draw_text_len</u></a>	문자를 지정한 길이만큼 출력한다.
<a href="#"><u>draw_text_in_box</u></a>	지정한 영역안에 문자를 출력한다.
<a href="#"><u>text_width</u></a>	문자열의 길이를 확인한다.

## ► egl\_get\_font function

```
EGL_FONT* egl_get_font(  
    EGL_HANDLE h  
)
```

### Overview

Object 가 사용하는 font 정보를 반환한다.

### Parameter

EGL\_HANDLE h                   Font 정보를 가져 올 object handle.

### Return Value

Object 가 사용하고 있는 font 정보. (EGL\_FONT Struct)

### Example

```
EGL_HANDLE object;  
...  
...  
EGL_FONT* cur_font = egl_get_font(object);
```

## ► egl\_set\_font function

```
EGL_FONT* egl_set_font(  
    EGL_HANDLE h,  
    EGL_FONT* font  
) ;
```

### Overview

Object 가 사용 할 font 를 설정한다.

### Parameter

EGL_HANDLE h	Font 를 설정할 object handle.
EGL_FONT* font	설정 할 font.

### Return Value

이전에 설정되어 있던 font 정보.

### Example

```
EGL_HANDLE object;  
EGL_FONT* font;  
...  
font = create_bitfont();  
egl_set_font(object , font);
```

## ► egl\_font\_set\_bkmode function

```
void egl_font_set_bkmode(  
    EGL_FONT* font,  
    int mode  
)
```

### Overview

Font 의 배경색 사용 여부를 결정한다.

### Parameter

EGL_FONT* font	설정을 변경 할 font.
int mode	배경 색 사용 여부. TRUE or FALSE.

### Return Value

없음.

### Example

```
EGL_FONT* font;  
...  
font = create_bitfont();  
egl_set_font(object , font);  
egl_font_set_bkmode(font, TRUE);
```

## ► egl\_font\_get\_bk\_color function

```
EGL_COLOR egl_font_get_bk_color(  
    EGL_FONT* font  
) ;
```

### Overview

Font 의 배경색 정보를 가져온다.

### Parameter

EGL\_FONT\* font                  배경색 정보를 가져 올 font..

### Return Value

현재 설정 되어 있는 font 의 배경색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_bk_color;  
...  
font = create_bitfont();  
font_bk_color = egl_font_get_bk_color(font);
```

## ► egl\_font\_set\_bk\_color function

```
EGL_COLOR egl_font_set_bk_color(  
    EGL_FONT* font,  
    EGL_COLOR clr  
)
```

### Overview

Font 의 배경색을 설정한다.

### Parameter

EGL_FONT* font	배경색을 설정 할 font.
EGL_COLOR clr	배경색. ( MAKE_COLORREF( ) 매크로를 사용하여 설정)

### Return Value

이전에 설정 되어 있던 font 의 배경색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_bk_color;  
...  
font = create_bitfont();  
font_bk_color = egl_font_set_bk_color(font, MAKE_COLORREF(0, 255, 0));
```

## ► egl\_font\_get\_color function

```
EGL_COLOR egl_font_get_color(  
    EGL_FONT* font  
)
```

### Overview

Font 의 색 정보를 가져온다.

### Parameter

EGL_FONT* font	색 정보를 가져 올 font.
----------------	------------------

### Return Value

현재 설정 되어 있는 font 색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_color;  
...  
font = create_bitfont();  
font_color = egl_font_get_color(font);
```

## ► egl\_font\_set\_color function

```
EGL_COLOR egl_font_set_color(  
    EGL_FONT* font,  
    EGL_COLOR clr  
)
```

### Overview

Font 의 색을 설정한다.

### Parameter

EGL_FONT* font	색을 설정 할 font.
EGL_COLOR clr	Font 색. ( MAKE_COLORREF( ) 매크로를 사용하여 설정)

### Return Value

이전에 설정 되어 있던 font 색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_color;  
...  
font = create_bitfont();  
font_color = egl_font_set_color(font, MAKE_COLORREF(0, 0, 255));
```

## ► **create\_bitfont** function

```
EGL_FONT* create_bitfont( void );
```

### Overview

Font 사용을 위해 Bit 방식의 font 를 생성한다.

### Parameter

없음.

### Return Value

생성된 bit font 구조체 포인터.

### Example

```
EGL_FONT* bit_font;
EGL_HANDLE object;
...
bit_font = create_bitfont();
egl_set_font(object, bit_font );
...
draw_text(bit_font, 100, 100, "font test");
```

## ► release\_bitfont function

```
void release_bitfont(  
    EGL_FONT* pFont  
)
```

### Overview

Create 한 bit 방식의 font 를 release 한다.

### Parameter

EGL\_FONT\* pFont                      Release 대상 bit font.

### Return Value

없음.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
release_bitfont(bit_font);
```

## ► **create\_bmpfont** function

```
EGL_FONT* create_bmpfont(  
    const char *fname  
)
```

### Overview

Font 사용을 위해 이미지 font 를 생성한다.

### Parameter

const char *fname	이미지 font file name.
-------------------	---------------------

### Return Value

생성된 이미지 font 구조체 포인터.

### Example

```
EGL_FONT* bm_font;  
EGL_HANDLE object;  
...  
bm_font = create_bmpfont("font/batang_24.fnt");  
egl_set_font(object, bm_font);  
...  
draw_text(bm_font, 100, 100, "image font test");
```

## ► **bmfont\_release function**

```
void bmfont_release(  
    EGL_FONT* pFont  
)
```

### Overview

Create 한 이미지 font 를 release 한다.

### Parameter

EGL\_FONT\* pFont                      Release 대상 이미지 font.

### Return Value

없음.

### Example

```
EGL_FONT* bm_font;  
...  
bm_font = create_bmpfont("font/batang_24.fnt");  
...  
bmfont_release(bm_font);
```

## ►draw\_text function

```
int draw_text(  
    EGL_FONT* pFont,  
    int x,  
    int y,  
    const char* text  
)
```

### Overview

문자를 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
int x	출력할 x 좌표.
int y	출력할 y 좌표.
const char* text	출력할 문자열.

### Return Value

출력한 문자열 갯수.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
draw_text(bit_font, 100, 100, "font test");
```

## ► draw\_text\_pivot function

```
int draw_text_pivot(  
    EGL_FONT* pFont,  
    int x,  
    int y,  
    const char* text,  
    int pivot  
) ;
```

### Overview

문자를 좌/우 90 도 회전하여 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
int x	출력할 x 좌표.
int y	출력할 y 좌표.
const char* text	출력할 문자열.
int pivot	좌/우 90 도 회전 여부. PIVOT_RIGHT / PIVOT_LEFT

### Return Value

출력한 문자열 갯수.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
draw_text_pivot(bit_font, 100, 100, "font test",PIVOT_RIGHT);
```

## ►draw\_text\_len function

```
void draw_text_len(  
    EGL_FONT* pFont,  
    int x,  
    int y,  
    const char* text,  
    int len  
)
```

### Overview

문자를 지정한 길이만큼 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
int x	출력할 x 좌표.
int y	출력할 y 좌표.
const char* text	출력할 문자열.
int len	출력할 문자 개수

### Return Value

없음.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
draw_text_len(bit_font, 100, 100, "font test",4);
```

## ► draw\_text\_in\_box function

```
void draw_text_in_box(  
    EGL_FONT* pFont,  
    EGL_RECT* pRect,  
    const char* text,  
    int align  
) ;
```

### Overview

지정한 영역 안에 문자를 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
EGL_RECT* pRect	출력할 영역. <pre>typedef struct _tag_RECT{     int x;     int y;     int w;     int h; } EGL_RECT;</pre>
const char* text	출력할 문자열.
int align	영역 안에서의 정렬. <pre>EGL_ALIGN_LEFT, EGL_ALIGN_RIGHT, EGL_ALIGN_TOP, EGL_ALIGN_BOTTOM, EGL_ALIGN_CENTER, EGL_ALIGN_MULTILINE</pre>

### Return Value

없음.

### Example

```
EGL_FONT* bit_font;
EGL_RECT rect;
...
bit_font = egl_create_default_font();
rect.x = 100;
rect.y = 100;
rect.w = 200;
rect.h = 40;
draw_text_in_box(bit_font, &rect, "font test", EGL_ALIGN_CENTER);
```

## ► **text\_width function**

```
int text_width(  
    EGL_FONT* font,  
    const char* str  
)
```

### Overview

문자열의 길이(pixel 단위)를 확인한다.

### Parameter

EGL_FONT* font	문자열에 사용 할 font. (font마다 한글자의 width가 다르기 때문에 지정해 주어야 된다.)
const char* str	Width 값을 확인 할 문자열.

### Return Value

문자열 width.

### Example

```
EGL_FONT* bit_font;  
int font_width;  
  
...  
bit_font = egl_create_default_font();  
  
...  
font_width = text_width(bit_font, "font test");
```

# EGL Primitives

Function	Description
<a href="#"><u>draw_line</u></a>	line 을 그린다.
<a href="#"><u>draw_hline</u></a>	가로 line 을 그린다.
<a href="#"><u>draw_vline</u></a>	세로 line 을 그린다.
<a href="#"><u>draw_thickline</u></a>	Thickline line 을 그린다.
<a href="#"><u>draw_rect</u></a>	사각형을 그린다.
<a href="#"><u>draw_rectfill</u></a>	안이 채워진 사각형을 그린다.
<a href="#"><u>draw_rectfill_gradient</u></a>	gradient 효과가 들어간 사각형을 그린다.
<a href="#"><u>draw_rectfill_h_gradient</u></a>	가로로 gradient 효과가 들어간 사각형을 그린다.
<a href="#"><u>draw_rectfill_v_gradient</u></a>	세로로 gradient 효과가 들어간 사각형을 그린다.
<a href="#"><u>draw_roundrect</u></a>	모서리가 둥근 사각형을 그린다.
<a href="#"><u>draw_roundrectfill</u></a>	모서리가 둥근 안이 채워진 사각형을 그린다.
<a href="#"><u>draw_arc</u></a>	호, 반원, 원을 그린다.
<a href="#"><u>draw_pie</u></a>	파이 모양의 도형을 그린다.
<a href="#"><u>draw_piefill</u></a>	안이 채워진 파이 모양의 도형을 그린다.
<a href="#"><u>draw_ellipse</u></a>	타원을 그린다.
<a href="#"><u>draw_ellipselfill</u></a>	안이 채워진 타원을 그린다.
<a href="#"><u>draw_circle</u></a>	원을 그린다.
<a href="#"><u>draw_circelfill</u></a>	안이 채워진 원을 그린다.

<a href="#"><u>draw_bezier</u></a>	베이어 곡선을 그린다.
<a href="#"><u>draw_polyline</u></a>	지정한 좌표를 잇는 선을 그린다.
<a href="#"><u>draw_polygon</u></a>	지정한 좌표를 잇는 다각형을 그린다.
<a href="#"><u>draw_polygonfill</u></a>	지정한 좌표를 잇는 안이 채워진 다각형을 그린다.

## ►draw\_line function

```
void draw_line(  
    int x,  
    int y,  
    int x2,  
    int y2,  
    EGL_COLOR c  
>);
```

### Overview

(x, y)에서 (x2,y2)까지 line 을 그린다.

### Parameter

int x	Line 의 시작 x 좌표.
int y	Line 의 시작 y 좌표.
int x2	Line 의 끝 x 좌표.
int y2	Line 의 끝 y 좌표.
EGL_COLOR c	Line 의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_line(100, 100, 200, 200, MAKE_COLORREF(0, 0, 0));  
draw_line(100, 100, 100, 200, MAKE_COLORREF(255, 0, 0));
```

## ► draw\_hline function

```
void draw_hline(  
    int x,  
    int y,  
    int x2,  
    EGL_COLOR c  
)
```

### Overview

(x, y)에서 (x2, y)까지 가로 line 을 그린다.

### Parameter

int x	Line 의 시작 x 좌표.
int y	Line 의 시작과 끝 y 좌표.
int x2	Line 의 끝 x 좌표.
EGL_COLOR c	Line 의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_hline(100, 100, 200, MAKE_COLORREF(0, 0, 0));  
draw_hline(100, 200, 200, MAKE_COLORREF(0, 0, 255));
```

## ►draw\_vline function

```
void draw_vline(  
    int x,  
    int y,  
    int y2,  
    EGL_COLOR c  
)
```

### Overview

(x, y)에서 (x, y2)까지 세로 line 을 그린다.

### Parameter

int x	Line 의 시작과 끝 x 좌표.
int y	Line 의 시작 y 좌표.
int y2	Line 의 끝 y 좌표.
EGL_COLOR c	Line 의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_vline(100, 100, 200, MAKE_COLORREF(0, 0, 0));  
draw_vline(200, 100, 200, MAKE_COLORREF(0, 255, 0));
```

## ► draw\_thickline function

```
void draw_thickline(  
    int x1,  
    int y1,  
    int x2,  
    int y2,  
    U8 width,  
    EGL_COLOR color  
) ;
```

### Overview

(x1, y1)에서 (x2,y2)까지 설정한 두께의 line 을 그린다.

### Parameter

int x1	Thick line 의 시작 x 좌표.
int y1	Thick line 의 시작 y 좌표.
int x2	Thick line 의 끝 x 좌표.
int y2	Thick line 의 끝 y 좌표.
U8 width	Thick line 의 두께.
EGL_COLOR color	Thick line 의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_thickline(100, 130, 200, 130, 2, MAKE_COLORREF(0, 0, 0));  
draw_thickline(100, 200, 100, 300, 3, MAKE_COLORREF(255, 0, 0));
```

## ►draw\_rect function

```
void draw_rect(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR c  
)
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h 인 사각형을 그린다.

### Parameter

int x	사각형의 시작 x 좌표.
int y	사각형의 시작 y 좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR c	사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rect(100, 100, 200, 200, MAKE_COLORREF(0, 0, 0));  
draw_rect(150, 150, 100, 100, MAKE_COLORREF(0, 0, 255));
```

## ► draw\_rectfill function

```
void draw_rectfill(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR c  
)
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h 인 안이 채워진 사각형을 그린다.

### Parameter

int x	사각형의 시작 x 좌표.
int y	사각형의 시작 y 좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR c	사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill(100, 100, 50, 50, MAKE_COLORREF(0, 255, 0));  
draw_rectfill(200, 200, 50, 50, MAKE_COLORREF(0, 0, 255));
```

## ► draw\_rectfill\_gradient function

```
void draw_rectfill_gradient(
    int x,
    int y,
    int w,
    int h,
    EGL_COLOR startcolor,
    EGL_COLOR endcolor,
    BOOL bVertical
);
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h 인 gradient 사각형을 그린다.

### Parameter

int x	사각형의 시작 x 좌표.
int y	사각형의 시작 y 좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR startcolor	Gradient 시작 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
EGL_COLOR endcolor	Gradient 끝 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
BOOL bVertical	Gradient 방향. TRUE == vertical, FALSE == horizontal.

### Return Value

없음.

### Example

```
draw_rectfill_gradient(100,100,200,50,MAKE_COLORREF(0,255,0),MAKE_COLORREF(255,255,255),FALSE);
draw_rectfill_gradient(100,200,50,200,MAKE_COLORREF(255,255,255),MAKE_COLORREF(0,0,255),TRUE);
```

## ► draw\_rectfill\_h\_gradient function

```
void draw_rectfill_h_gradient(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR startcolor,  
    EGL_COLOR endcolor  
) ;
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h 인 가로방향 gradient 사각형을 그린다.

### Parameter

int x	사각형의 시작 x 좌표.
int y	사각형의 시작 y 좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR startcolor	gradient 시작 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
EGL_COLOR endcolor	gradient 끝 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill_h_gradient(50,50,200,50,MAKE_COLORREF(255,255,255),MAKE_COLORREF(255,0,0));
```

## ► draw\_rectfill\_v\_gradient function

```
void draw_rectfill_v_gradient(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR startcolor,  
    EGL_COLOR endcolor  
)
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h 인 세로방향 gradient 사각형을 그린다.

### Parameter

int x	사각형의 시작 x 좌표.
int y	사각형의 시작 y 좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR startcolor	gradient 시작 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
EGL_COLOR endcolor	gradient 끝 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill_v_gradient(50,50,200,50,MAKE_COLORREF(255,255,255),MAKE_COLORREF(255,0,0));
```

## ► draw\_rect function

```
void draw_rect(  
    int x0,  
    int y0,  
    int w,  
    int h,  
    int corner,  
    EGL_COLOR c  
)
```

### Overview

(x0, y0)부터 가로크기가 w, 세로크기가 h 인 둥근 사각형을 그린다.

### Parameter

int x0	둥근 사각형의 시작 x 좌표.
int y0	둥근 사각형의 시작 y 좌표.
int w	둥근 사각형의 가로 크기.
int h	둥근 사각형의 세로 크기.
int corner	모서리 둥근 정도.
EGL_COLOR c	둥근 사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rect(100, 100, 100, 100, 10, MAKE_COLORREF(0, 0, 0));  
draw_rect(400, 200, 200, 200, 20, MAKE_COLORREF(0, 255, 0));
```

## ►draw\_rectfill function

```
void draw_rectfill(  
    int x0,  
    int y0,  
    int w,  
    int h  
    int corner,  
    EGL_COLOR c  
)
```

### Overview

(x0, y0)부터 가로크기가 w, 세로크기가 h 인 안이 채워진 둥근 사각형을 그린다.

### Parameter

int x0	둥근 사각형의 시작 x 좌표.
int y0	둥근 사각형의 시작 y 좌표.
int w	둥근 사각형의 가로 크기.
int h	둥근 사각형의 세로 크기.
int corner	모서리 둥근 정도.
EGL_COLOR c	둥근 사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill(100, 100, 100, 100, 10, MAKE_COLORREF(255, 0, 0));  
draw_rectfill(300, 200, 200, 200, 15, MAKE_COLOEREF(0, 255, 0));
```

## ► draw\_arc function

```
void draw_arc(
    int x,
    int y,
    int rx,
    int ry,
    int a1,
    int a2,
    EGL_COLOR c
);
```

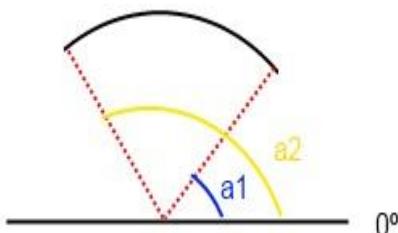
### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry 인 곡면을 그린다.

a1 을 시작 각으로 하여 각 a2 까지 곡면을 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int rx	곡면의 가로 반지름.
int ry	곡면의 세로 반지름.
int a1	곡면의 시작 각도.
int a2	곡면의 끝 각도.
EGL_COLOR c	곡면의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)



### Return Value

없음.

### Example

```
draw_arc(100, 100, 30, 30, 0, 90, MAKE_COLORREF(255, 0, 0));
draw_arc(200, 200, 40, 30, 90, 180, MAKE_COLORREF(0, 0, 255));
```

## ►draw\_pie function

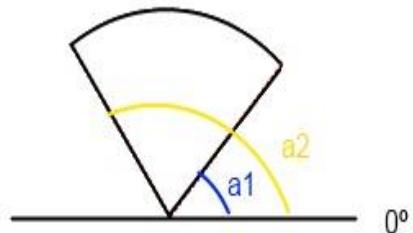
```
draw_pie(
    int x,
    int y,
    int rx,
    int ry,
    int a1,
    int a2,
    EGL_COLOR c
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry 인 파이모양을 그린다.  
a1 을 시작 각으로 하여 각 a2 까지 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int rx	파이의 가로 반지름.
int ry	파이의 세로 반지름.
int a1	파이의 시작 각도.
int a2	파이의 끝 각도.
EGL_COLOR c	파이의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)



### Return Value

없음.

### Example

```
draw_pie(100, 100, 50, 50, 45, 90, MAKE_COLORREF(0, 0, 255));
draw_pie(200, 100, 60, 60, 90, 180, MAKE_COLORREF(255, 0, 0));
```

## ► draw\_piefill function

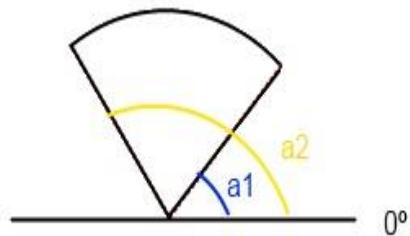
```
void draw_piefill(
    int x,
    int y,
    int rx,
    int ry,
    int a1,
    int a2,
    EGL_COLOR c
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry 인 채워진 파이모양을 그린다.  
a1 을 시작 각으로 하여 각 a2 까지 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int rx	파이의 가로 반지름.
int ry	파이의 세로 반지름.
int a1	파이의 시작 각도.
int a2	파이의 끝 각도.
EGL_COLOR c	파이의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)



### Return Value

없음.

### Example

```
draw_pie(100, 100, 50, 50, 45, 90, MAKE_COLORREF(0, 0, 255));
draw_pie(200, 100, 60, 60, 90, 180, MAKE_COLORREF(255, 0, 0));
```

## ►draw\_ellipse function

```
void draw_ellipse(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    EGL_COLOR c  
)
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry 인 타원을 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int rx,	가로 반지름.
int ry	세로 반지름.
EGL_COLOR c	타원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_ellipse(100, 100, 50, 100, MAKE_COLORREF(0, 0, 255));  
draw_ellipse(300, 200, 100, 50, MAKE_COLORREF(255, 0, 0));
```

## ► draw\_ellipsefill function

```
void draw_ellipsefill(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    EGL_COLOR c  
)
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry 인 채워진 타원을 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int rx,	가로 반지름.
int ry	세로 반지름.
EGL_COLOR c	타원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_ellipsefill(100, 100, 50, 100, MAKE_COLORREF(0, 0, 255));  
draw_ellipsefill(300, 200, 100, 50, MAKE_COLORREF(255, 0, 255));
```

## ►draw\_circle function

```
void draw_circle(  
    int x,  
    int y,  
    int r,  
    EGL_COLOR c  
)
```

### Overview

원점이 (x, y)고 반지름이 r 인 원을 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int r	반지름.
EGL_COLOR c	원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_circle(100, 100, 50, MAKE_COLORREF(255, 0, 255));  
draw_circle(300, 200, 100, MAKE_COLORREF(0, 0, 255));
```

## ► draw\_circlefill function

```
void draw_circlefill(  
    int x,  
    int y,  
    int r,  
    EGL_COLOR c  
)
```

### Overview

원점이 (x, y)고 반지름이 r 인 채워진 원을 그린다.

### Parameter

int x	원점의 x 좌표.
int y	원점의 y 좌표.
int r	반지름.
EGL_COLOR c	원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_circle(100, 100, 70, MAKE_COLORREF(0, 0, 255));  
draw_circle(300, 200, 50, MAKE_COLORREF(0, 255, 0));
```

## ► draw\_bezier function

```
void draw_bezier(  
    EGL_POINT* pts,  
    int n,  
    int s,  
    EGL_COLOR c  
)
```

### Overview

Bezier curve 를 그린다.

### Parameter

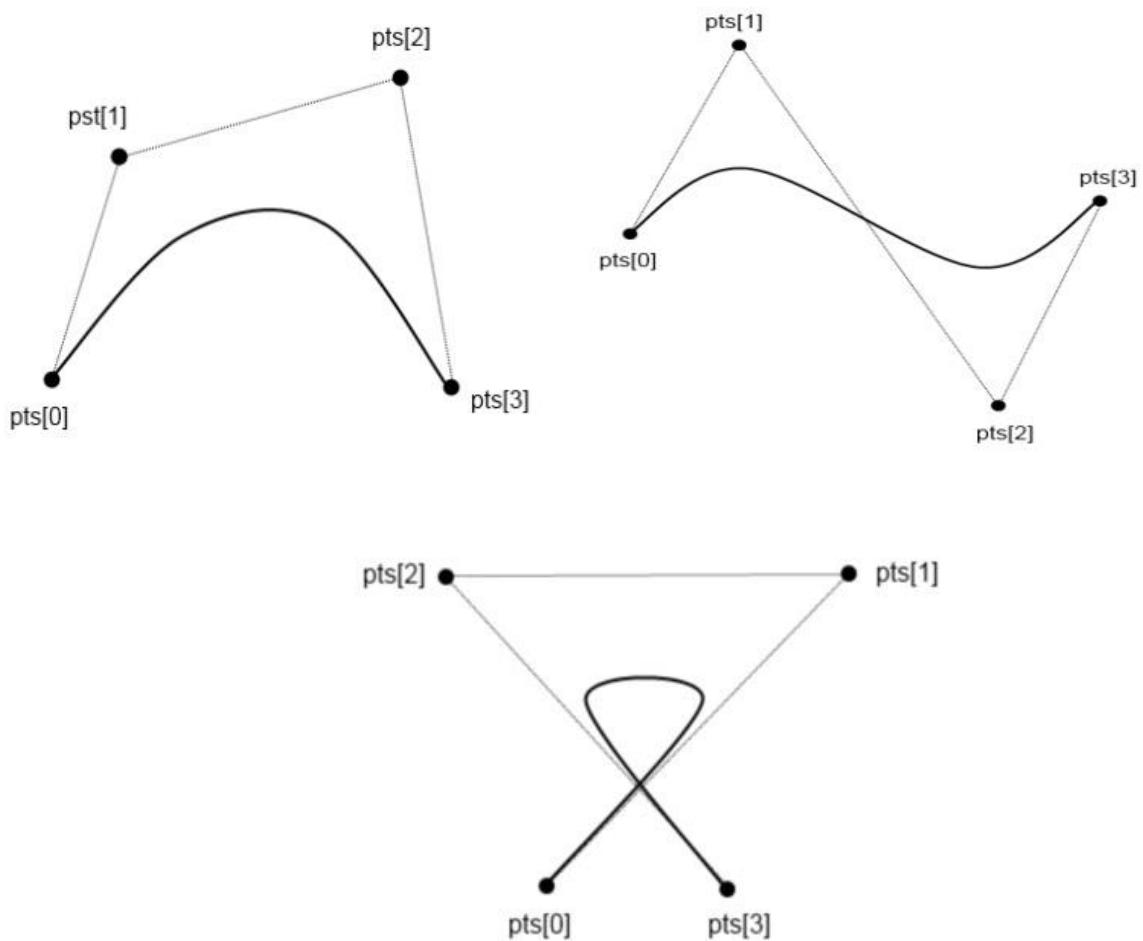
EGL_POINT* pts	좌표 배열.
int n	좌표 개수.
int s	point 개수. 좌표와 좌표사이에 몇 개의 point 를 draw 할지 결정.
EGL_COLOR c	curve 의 color.

### Return Value

없음.

### Example

```
EGL_POINT* pts[4];  
pts[0].x = 100; pts[0].y = 200;  
pts[1].x = 200; pts[1].y = 50;  
pts[2].x = 300; pts[2].y = 50;  
pts[3].x = 400; pts[3].y = 200;  
  
draw_bezier(pts, 4, 20, MAKE_COLORREF(255, 0, 255));
```



< Bezier curve image >

## ►draw\_polyline function

```
void draw_polyline(  
    EGL_POINT* p,  
    int n,  
    EGL_COLOR c  
)
```

### Overview

지정한 좌표를 있는 선을 그린다.

### Parameter

EGL_POINT* p	좌표 배열.
int n	좌표 개수.
EGL_COLOR c	선의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
EGL_POINT p[3];  
p[0].x = 100; p[0].y = 100;  
p[1].x = 130; p[1].y = 100;  
p[2].x = 150; p[2].y = 150;  
draw_polyline(p, 3, MAKE_COLORREF(0, 0, 0));
```

## ► draw\_polygon function

```
void draw_polygon(  
    EGL_POINT* ptable,  
    int cnt,  
    EGL_COLOR c  
) ;
```

### Overview

지정한 좌표를 잇는 다각형을 그린다.

### Parameter

EGL_POINT* p	좌표 배열.
int n	좌표 개수.
EGL_COLOR c	다각형의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
EGL_POINT ptable[5];  
ptable[0].x = 100; ptable[0].y = 100;  
ptable[1].x = 150; ptable[1].y = 50;  
ptable[2].x = 200; ptable[2].y = 100;  
ptable[3].x = 135; ptable[3].y = 150;  
ptable[4].x = 115; ptable[4].y = 150;  
draw_polygon(ptable, 5, MAKE_COLORREF(0, 0, 255));
```

## ►draw\_polygonfill function

```
void draw_polygonfill(  
    EGL_POINT* ptable,  
    int cnt,  
    EGL_COLOR c  
)
```

### Overview

지정한 좌표를 있는 채워진 다각형을 그린다.

### Parameter

EGL_POINT* p	좌표 배열.
int n	좌표 개수.
EGL_COLOR c	다각형의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
EGL_POINT ptable[3];  
ptable[0].x = 100; ptable[0].y = 100;  
ptable[1].x = 150; ptable[1].y = 150;  
ptable[2].x = 100; ptable[2].y = 150;  
draw_polygon(ptable, 3, MAKE_COLORREF(0, 255, 0));
```

## EGL etc.

Function	Description
<a href="#">egl_init</a>	egl 초기화 함수.
<a href="#">egl_show_object</a>	Object 를 draw 여부를 결정한다.
<a href="#">egl_object_set_redraw</a>	Object redraw 할지를 결정한다.

Define
<pre>typedef enum enumMSG {     EGL_MSG_DRAW = 0,     EGL_MSG_DELETE,     EGL_MSG_FOCUS,     EGL_MSG_UNFOCUS,     EGL_MSG_KEY_UP,     EGL_MSG_KEY_DOWN,     EGL_MSG_TOUCHED,     EGL_MSG_UNTOUCHED,     EGL_MSG_MOVE,     EGL_MSG_TIMETICK } EGL_MSG_ID;</pre>
<pre>typedef struct _tagMessage {     EGL_MSGID msgID;     EGL_HANDLE hObj;     EGL_HANDLE hWin;     Union     {         EGL_POINT point;         U32 key;     } param; } EGL_MSG;</pre>

## ► egl\_init function

```
BOOL egl_init( void );
```

### Overview

egl 초기화를 수행한다. embedded graphic library 를 사용하기 위해서 반드시 호출해주어야 한다.

### Parameter

없음.

### Return Value

TRUE or FALSE

## ► egl\_show\_object function

```
void egl_show_object(  
    EGL_HANDLE h,  
    BOOL bShow  
)
```

### Overview

Object 를 draw 여부를 결정한다. bShow 값을 FALSE 설정 하면 egl\_draw( ) 호출 시 해당 object 는 draw 되지 않는다.

### Parameter

EGL\_HANDLE h                                   Draw 여부를 결정할 object handle.

BOOL bShow                                   Draw 여부. TRUE or FALSE.

### Return Value

없음.

## ► egl\_object\_set\_redraw function

```
void egl_object_set_redraw(  
    EGL_HANDLE handle  
)
```

### Overview

Object redraw 할지를 결정한다. 참고로 library에서 object의 변화가 있을 경우 이 함수를 호출하지 않아도 redraw가 자동으로 설정된다.

### Parameter

EGL\_HANDLE handle      Object를 redraw 설정 할 object handle.

### Return Value

없음.

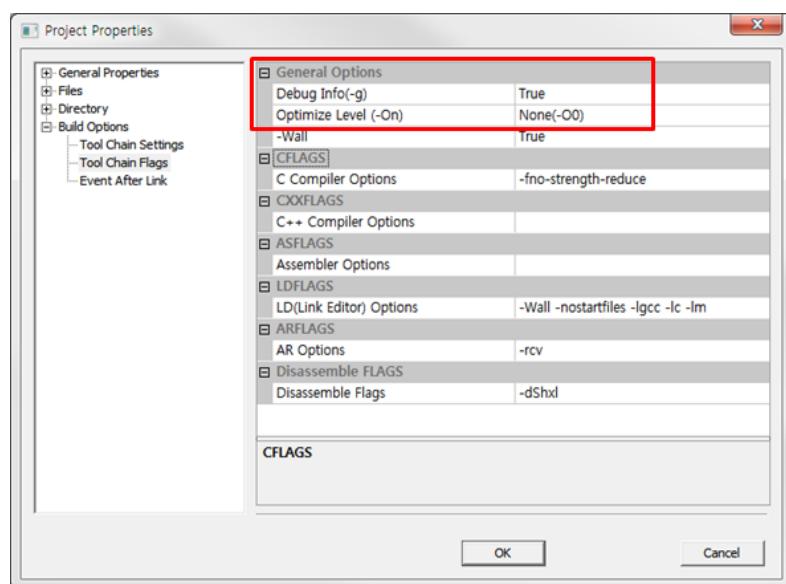
# 15. Debugging

이번 장에서는 debugging 에 대해서 설명한다.

Debugging 은 E-Con 과 EConMan 프로그램을 필요로 한다. E-Con 을 사용하기 위해서는 device driver 가 설치 되어야 하는데, 설치를 하지 않았다면, "1. Software 개발환경"의 device driver 설치하는 부분을 참고하기 바란다.

## 15.1 Debugging 준비

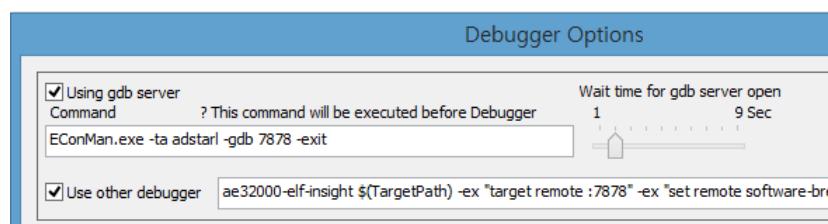
E-Con 과 EConMan 이 준비 되었다면 debugging 을 할 project 를 열어 Project → Properties 를 실행한다.



실행하면 위와 같은 창이 뜨게 되고, 빨간 상자 안의 내용, Debug Info(-g)를 TRUE 로 설정하고, Optimize Level(-O0)을 None(-O0)로 설정을 한 후 OK 를 누른다. 그런 다음 Build → Rebuild Project 를 실행하여 Rebuild 를 진행한다.

Debugging 을 위해서는 Debug Info 를 TRUE 로 하여 Build 에 의해 생성되는 elf 파일에 Debug 정보를 포함시켜야 하고, Optimize Level 을 None 으로 하여 최적화 없이 compile 해야 debugging 시 보다 정확하게 source 를 debugging 할 수 있다.

Project properties 를 설정 했으면, 다음으로 Debug 메뉴의 Debug option 을 실행하여 다음과 같이 -ta 뒤에 adstarl 을 입력한다.



그런 다음 ok 를 누르면, Debugging 준비과정은 끝난다.

## 15.2 Debugging

Debugging 은 다음의 두 가지 상황에 따라 시작하는 방법이 다르다.

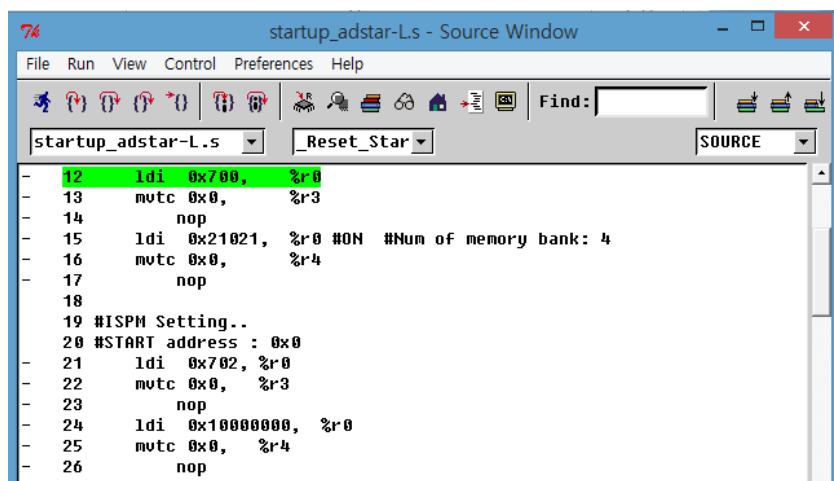
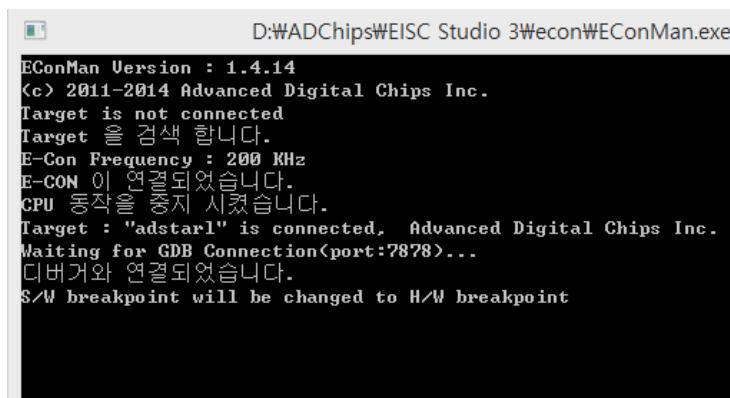
1. Bootloader 없이 0 번지(Serial Flash)에서 Program 이 실행되는 경우.
2. Bootloader 를 사용하여 Ram 에서 Program 이 실행되는 경우.

위의 두 경우는 linker script 를 가지고도 구분할 수 있다. Linker Script 가 adStar-L.ld 파일로 되어 있으면 0 번지에서 Program 이 실행되는 것이고, adStar-L\_ram.ld 파일로 되어 있으면 Ram 상에서 Program 이 실행되는 경우이다.

### ▶ 0 번지(Serial Flash)에서 Program 이 실행되는 경우

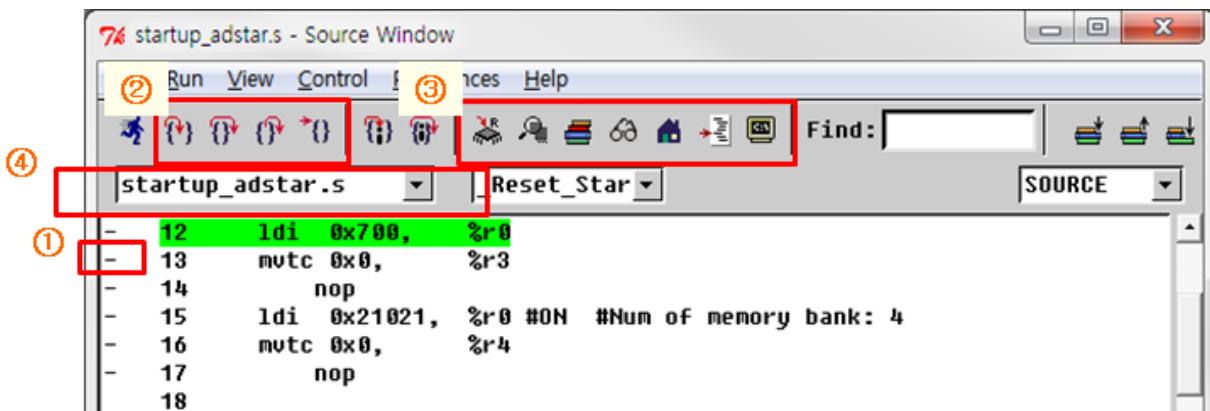
Program 이 0 번지(Serial Flash)에서 실행되는 경우 다음과 같이 debugging 을 진행한다.

- 1) Build 하여 생성 된 bin 파일을 0 번지에 download 한다.
  - 2) SW2 을 설정하여 Debug mode (SWD mode)로 놓은 상태에서 E-Con 을 연결하고 전원을 켠다. (SW2 를 Down 시키면 Debug mode 상태이다.)
  - 3) Debugging 할 project 가 열려있는 상태에서 Debug → Start Debugger (F5)를 실행한다.
- 그러면 다음과 같이 두개의 창이 뜨면서 debugging 을 시작할 수 있다.



위와 같은 창이 뜨고, 녹색 바가 표시되면 정상 연결된 것으로 debugging을 시작할 수 있다.  
녹색 바가 아닐 경우 연결이 제대로 안된 것으로 다시 시도해야 한다.

4) Debugger 창에 대해 설명하면 다음과 같다.



- ① : 마우스로 '-' 클릭 시 break point 설정/해제
- ② : 앞에서부터 'step', 'next', 'finish', 'continue' 등작
- ③ : 앞에서부터 'register', 'memory', 'stack', 'watch expressions', 'local variable' 확인창 출력.  
제일 마지막 'c:'는 'console'창을 띄운다. Console 창에서는 GDB 명령을 사용하여  
debugging을 수행할 수 있다.
- ④ : Source code 이동.

5) console 창에서의 debugging.

```
(gdb) b main
Breakpoint 1 at 0x68c: file main.c, line 229.

(gdb) c
Continuing.

Breakpoint 1, main () at main.c:229
Current language: auto; currently c

(gdb) n
(gdb) s
uart_config (ch=0, baud=115200, databits=DATABITS_8, stopbit=STOPBITS_1, parity=
```

b : break point 설정.

c : continue 명령. 다음 break point 까지 진행.

n : next 명령. 하나의 코드 실행. (함수일 경우에도 그대로 진행)

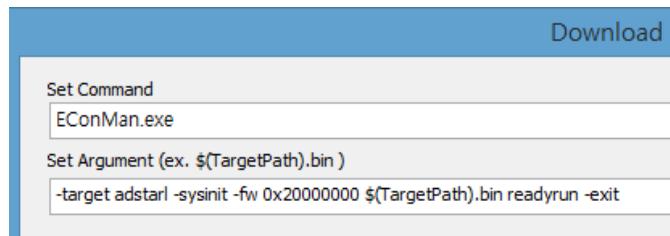
s : step 명령. 하나의 코드 실행. (함수인 경우 함수 내부로 들어가 진행)

q : 종료.

## ▶ Ram에서 Program이 실행되는 경우

Bootloader를 사용하는 경우 Program이 Ram에서 동작하게 되는데, 이 경우에는 이전과 debugging을 시작하는 방법이 조금 다르다.

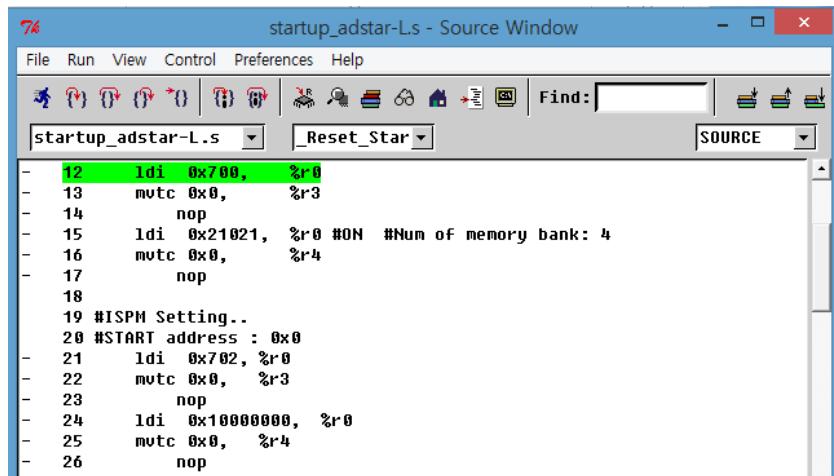
- 1) Bootloader가 download되어 있는 상태에서, SW3을 설정하여 Debug mode(JTAG mode)로 놓은 상태에서 E-Con을 연결하고 전원을 켠다.(SW3을 High로 놓으면 Debug mode 상태이다.)
- 2) Debugging 할 project를 열어서 build를 한 후 download option을 실행하여 다음과 같이 argument를 적어준다.



```
-target adStarL -sysinit -fw 0x20000000 $(TargetPath).bin readyrun -exit
```

(-fw는 파일을 원하는 ram 영역에 write하는 명령이고, 끝에 readyrun을 쓰게 되면, ram 영역에 write된 프로그램이 처음 실행되는 위치에서 대기하게 된다. 이렇게 실행 위치에 대기시켜 놓은 뒤에 Start Debugger를 실행하여 debugging을 수행할 수 있다.)

- 3) 위처럼 Download option을 작성한 후 Download to Target을 실행하여 프로그램을 ram 영역에 download하고, Debug → Start Debugger(F5)를 실행한다. 그러면 다음과 같은 창이 뜨면서 debugging을 시작할 수 있다.



- 4) 이제부터는 이전 0번지에서 동작하는 program을 debugging 할 때와 같은 방법으로 진행하면 된다.

# 16. ETC

이번 장에서는 프로그램 동작 시간과 남은 메모리(Ram size)를 확인하는 방법에 대해 설명하겠다.

## 16.1 동작시간 측정방법

어느 특정 구간의 동작시간을 측정하고 싶을 경우 etc\_driver/misc.c 파일에 있는 StartEstimate() 와 EndEstimate(char\* str) 함수를 사용하면 된다.

사용방법은 project 에 etc\_driver/misc.c 파일을 추가하고, 시간 측정을 원하는 시작지점에서 StartEstimate() 함수를 호출하고, 끝지점에서 EndEstimate("time : ")함수를 호출하면 된다. EndEstimate 의 str 인자는 시간을 출력할 때 함께 출력할 문자열이다.

```
StartEstimate();
SURFACE* surf = loadbmp("mp3bg.bmp");
EndEstimate("Time : ");
```

위처럼 하면 loadbmp 를 수행하는데 소요된 시간을 확인할 수 있다.

```
StartEstimate();
SURFACE* surf = loadbmp("mp3bg.bmp");
draw_surface(surf,0,0);
SURFACE* surf2 = loadpng((U8*)Anteprima2_png,Anteprima2_png_len);
draw_surface(surf2,100,100);
EndEstimate("Time : ");
```

한 line 동작이 아닌 복수 개 line 동작에 대해서도 시간을 확인할 수 있다.

## 16.2 여유 메모리(Ram size) 확인 방법

사용할 수 있는 메모리 size 가 얼마나 되는지, stack pointer address 와 heap 영역의 end address 의 차를 통해 대략적으로 확인이 가능한다.

get\_free\_mem() 함수가 stack pointer address 와 heap end address 의 차이 값을 return 해주므로, 이 함수를 사용하여 현재 사용할 수 있는 메모리 size 를 확인할 수 있다.

참고로 Return 하는 값은 stack pointer address 와 heap end address 의 차이를 통한 대략적인 값이지 정확한 값이 아니라는 것을 알고 있어야 한다.

Heap 영역 중간에 할당되었던 메모리를 free 할 경우 그 부분이 사용 가능해 지면서 사용할 수 있는 메모리 size 는 증가하지만 heap end address 가 이동하는 것은 아니기 때문에 get\_free\_mem 함수로 메모리 확인 시 값에 변화는 없게 된다.

Malloc 의 경우에도 할당 받을 메모리 size 는 줄겠지만, heap end address 의 변화는 발생하지 않아 get\_free\_mem 함수로 메모리 size 확인 시 값에 변화가 없게된다.

```
int memsize = get_free_mem();
```