Embeded GUI Library Manual (for EBTB1200)

- E2BOX N.K.J

Ver 1.0

1. 개요 및 사용

EUI ?

EUI는 마이크로컨트롤러 기반 보드에서 OS없이 쉽게 GUI(Graphic User Interface)를 구성할 수 있도록 만든 라이브러리이다.

gui.c

EUI의 모든 소스가 담겨져 있는 파일이다.

별도의 헤더파일을 두지 않는다.

따라서 메인프로그램 파일에서 gui.c를 직접 include하여 컴파일 하면 된다.

헤더파일과 소스를 분리하여 binary 라이브러리화 할 수 있다.

#include "gui.c"

gui.c를 include 하기전에 선언되어 있어야 하는 외부함수들이 있다.

void setpixel(int x,int y, unsigned int rgb);

- LCD의 x,y 위치에 rgb칼라의 1점을 찍는 함수

void LcdStringKS5601CT(unsigned int x, unsigned int y, unsigned char *str, unsigned int rgb);

- LCD의 x,y 위치에 rgb칼라의 16*16한글 8*16ASCII문자열을 출력하는 함수.
  글자배경은 변경되지 않음.

void LcdAscii5x7StringCT(unsigned int x, unsigned int y, unsigned char *str, unsigned int rgb);

- LCD의 x,y 위치에 rgb칼라의 5*7ASCII문자열을 출력하는 함수.
  글자배경은 변경되지 않음.

void LcdAscii8x8StringCT(unsigned int x, unsigned int y, unsigned char *str, unsigned int rgb);

- LCD의 x,y 위치에 rgb칼라의 8*8ASCII문자열을 출력하는 함수.
  글자배경은 변경되지 않음.

void LcdAscii8x16StringCT(unsigned int x, unsigned int y, unsigned char *str, unsigned int rgb);

- LCD의 x,y 위치에 rgb칼라의 8*16ASCII문자열을 출력하는 함수.
  글자배경은 변경되지 않음.

void LcdSetWindow(unsigned short x, unsigned short y, unsigned short width, unsigned short height);

- LCD에 출력할 영역을 설정하는 함수.

LcdDataStart();

- LCD Data Write 시작 매크로

LcdDataWrite();

- LCD Data Write 매크로

LcdDataEnd();

- LCD Data Write 종료 매크로

2. EUI Library Definition

EUI Touch Pointer : EuiP

```
//////////////// EUI_TOUCH_POINTER ///////////////////////
#define EUI_TOUCH_NODATA   0
#define EUI_TOUCH_DOWN     10
#define EUI_TOUCH_UP       11
#define EUI_TOUCH_MOVE     12
#define EUI_TOUCH_NOMOVE   13

typedef struct{
signed int x;
signed int y;
signed int down_x;
signed int down_y;
unsigned int action;
}EUI_POINTER;

EUI_POINTER EuiP={-1,-1,-1,-1,EUI_TOUCH_NODATA};
//////////////////////////////////////////////////////////////////////
```

외부의 사용자 입력 디바이스로 부터 좌표를 얻기 위한 전역 구조체이다.
x,y는 현재의 좌표이고 down_x, down_y는 터치(클릭)을 시작한 위치이다.(touch down)
action은 사용자 입력이 있을경우 EUI_TOUCH_DOWN(LCD접촉), EUI_TOUCH_UP(LCD접촉해제), EUI_TOUCH_MOVE(LCD 접촉후 드래그)를 저장해야 한다. EUI_TOUCH_NOMOVE는 현재 라이브러리 에서는 사용되지 않는다.
USER프로그램에서는 EuiP.action을 감시하여 사용자 입력이 있을 경우 EUI각 컨트롤의 Op함수를 실행시키고, Euip.action은 반드시 EUI_TOUCH_NODATA로 만들어 줘야 한다.
예)

```
if(EuiP.action!=EUI_TOUCH_NODATA)
    {
      OpDialog(&dlg1);
      OpSlider(&sld1);
      EuiP.action=EUI_TOUCH_NODATA;

    }
```

EuiP라는 전역 구조체가 선언되어 있으며,
x=(-1), y=(-1), down_x=(-1), down_y=(-1), action=EUI_TOUCH_NODATA
로 초기화 되어 있다.

LCD WRITE MODE

```
//////////////////// LCD WRITE MODE..//////////////////////
#define EUI_LCD_DIRECTWRITE
//#define EUI_LCD_USEFUCTION
////////////////////////////////////////////////////////////
```

EUI에서 LCD에 각 컨트롤들을 그릴때 사용하는 모드를 정의할 수 있다.
EUI_LCD_DIRECTWRITE 는 LcdDataStart() LcdDataWrite() LcdDataEnd() 매크로 등을 사용하여 LCD에 고속으로 데이타를 WRITE함.
EUI_LCD_USEFUCTION 는 setpixel(int x,int y, unsigned int rgb) 함수를 사용하여 LCD에 데이타를 WRITE함.

EUI_LCD_DIRECTWRITE 의 장점은 고속 출력이 가능하다. 하지만 컨트롤의 좌표 위치가 lcd화면을 이탈한 경우 화면이 깨진다.
EUI_LCD_USEFUCTION 의 장점은 컨트롤의 좌표가 lcd화면을 벗어나더라도 정상적이 화면이 출력된다. 하지만 lcd화면 출력 속도가 느리다.

HANGUL FONT USING

```
//////////////////// HANGUL FONT USING ////////////////////
#define EUI_USE_HANGUL_FONT
////////////////////////////////////////////////////////////
```

한글을 사용하기 위해서는 위의 정의가 있어야 한다. 완성형 한글코드를 지원하므로 한글출력을 쉽게 할 수 있다.
한글 조합형폰트가 약11Kbyte 정도 이고 완성형변환코드가 4.7Kbyte 정도 된다. 코드사이즈가 커지기 때문에 굳이 한글을 사용할 필요가 없다면 위의 정의를 주석처리하면 된다.

THEMA settings

```
//////////////////// THEMA settings ////////////////////
#define EUI_COLOR_BG        (RGB(58,110,165))

#define EUI_COLOR_DLG   (RGB(212,208,200))
#define EUI_COLOR_DLG_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_DLG_LIGHT2   (RGB(212,208,200))
#define EUI_COLOR_DLG_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_DLG_SHADOW2   (RGB(64,64,64))
#define EUI_COLOR_DLG_TITLE    (RGB(10,36,106))

#define EUI_GRAID_LEVEL   12 // title bar gradation level
```

```
#define EUI_COLOR_BTN   (RGB(212,208,200))
#define EUI_COLOR_BTN_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_BTN_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_BTN_SHADOW2   (RGB(64,64,64))

#define EUI_COLOR_PROG   (RGB(212,208,200))
#define EUI_COLOR_PROG_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_PROG_SHADOW1   (RGB(128,128,128))

#define EUI_COLOR_EDIT   (RGB(255,255,255))
#define EUI_COLOR_EDIT_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_EDIT_LIGHT2   (RGB(212,208,200))
#define EUI_COLOR_EDIT_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_EDIT_SHADOW2   (RGB(64,64,64))

#define EUI_COLOR_CHECK   (RGB(255,255,255))
#define EUI_COLOR_CHECK_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_CHECK_LIGHT2   (RGB(212,208,200))
#define EUI_COLOR_CHECK_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_CHECK_SHADOW2   (RGB(64,64,64))

#define EUI_COLOR_SLIDER   (RGB(212,208,200))
#define EUI_COLOR_SLIDER_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_SLIDER_LIGHT2   (RGB(212,208,200))
#define EUI_COLOR_SLIDER_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_SLIDER_SHADOW2   (RGB(64,64,64))

#define EUI_COLOR_LIST   (RGB(255,255,255))
#define EUI_COLOR_LIST_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_LIST_LIGHT2   (RGB(212,208,200))
#define EUI_COLOR_LIST_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_LIST_SHADOW2   (RGB(64,64,64))
#define EUI_COLOR_LIST_SELECTAREA    (RGB(10,36,106))
#define EUI_COLOR_LIST_SELECTTEXT    (RGB(255,255,255))

#define EUI_COLOR_SPIN   (RGB(212,208,200))
#define EUI_COLOR_SPIN_LIGHT1   (RGB(255,255,255))
#define EUI_COLOR_SPIN_SHADOW1   (RGB(128,128,128))
#define EUI_COLOR_SPIN_SHADOW2   (RGB(64,64,64))
//////////////////////////////////////////////////////////
```

EUI의 각 컨트롤 들에 대한 색상 정의다.
아래와 같은 규칙을 가진다.
EUI_COLOR_컨트롤 : 컨트롤의 BODY색상
EUI_COLOR_컨트롤_LIGHT1 : 컨트롤이 입체적으로 보이기 위한 OUTLINE의 밝은부분 색상
EUI_COLOR_컨트롤_LIGHT2 : 컨트롤이 입체적으로 보이기 위한 OUTLINE의 밝은부분 색상2
EUI_COLOR_컨트롤_SHADOW1 : 컨트롤이 입체적으로 보이기 위한 OUTLINE의 어두분부분 색상1
EUI_COLOR_컨트롤_SHADOW2 : 컨트롤이 입체적으로 보이기 위한 OUTLINE의 어두분부분 색상2

위의 정의를 변경하여 다른 색상의 UI를 구성할 수 있다.

USER EVENT LIST

```
//////////////////// USER EVENT LIST ////////////////////
#define EUI_EVENT_NO                    0

#define EUI_EVENT_DLG_TITLEBAR_DOWN     10
#define EUI_EVENT_DLG_TITLEBAR_UP       11
#define EUI_EVENT_DLG_EXITBTN_DOWN      20
#define EUI_EVENT_DLG_EXITBTN_UP        21
#define EUI_EVENT_DLG_ICON_DOWN         30
#define EUI_EVENT_DLG_ICON_UP           31

#define EUI_EVENT_BTN_DOWN              10
#define EUI_EVENT_BTN_UP                11

#define EUI_EVENT_PROG_DOWN             10
#define EUI_EVENT_PROG_UP               11

#define EUI_EVENT_EDIT_DOWN             10
#define EUI_EVENT_EDIT_UP               11

#define EUI_EVENT_CHECK_CHANGED         41

#define EUI_EVENT_SLIDER_CHANGING       40
#define EUI_EVENT_SLIDER_CHANGED        41

#define EUI_EVENT_LIST_CHANGING         40
#define EUI_EVENT_LIST_CHANGED          41
```

```
#define EUI_EVENT_STATIC_DOWN              10
#define EUI_EVENT_STATIC_UP                11

#define EUI_EVENT_PICTURE_DOWN             10
#define EUI_EVENT_PICTURE_UP               11

#define EUI_EVENT_SPIN_INCREMENT_DOWN      10
#define EUI_EVENT_SPIN_INCREMENT_UP        11
#define EUI_EVENT_SPIN_DECREMENT_DOWN      20
#define EUI_EVENT_SPIN_DECREMENT_UP        21

#define EUI_EVENT_OWNERBTN_DOWN            10
#define EUI_EVENT_OWNERBTN_UP              11
//////////////////////////////////////////////////////////
```

컨트롤 생성후 EuiP를 통해 사용자입력을 받았을 경우 Op함수를 실행 시키면 각 컨트롤의 event가 발생하게 된다. 컨트롤 별로 발생하게 되는 event를 정의한 것이다. 컨트롤에서 발생한 event는 이벤트처리 후 EUI_EVENT_NO를 입력하여 clear 해 줘야 한다.

예)
```
if(EuiP.action!=EUI_TOUCH_NODATA)
    {
      OpDialog(&dlg1);
      OpButton(&btn1);
      EuiP.action=EUI_TOUCH_NODATA;

      if(dlg1.event==EUI_EVENT_DLG_EXITBTN_UP)
       {   dlg1.event=EUI_EVENT_NO; break;
       }
      if(btn1.event==EUI_EVENT_BTN_UP)
       {   mystrcpy("Button1 UP",edit1.text);
           ViewEdit(&edit1);
           btn1.event=EUI_EVENT_NO;
       }
    }
```

기타정의

```
#define EUI_FONT5X7      0
#define EUI_FONT8X8      1
#define EUI_FONT8X16     3
#define EUI_FONTHangul   4
```

컨트롤에 사용되는 text에 대한 폰트크기를 지정할 경우 위의 정의를 사용한다.

```
#define EUI_MAX_TEXT    50
```
컨트롤에 사용되는 text가 가질 수 있는 최대 문자수를 지정한다.
(각 컨트롤의 text가 지정된 문자수를 넘지 않도록 주의해야 한다.)

```
#define EUI_ALIGN_LEFT   0
#define EUI_ALIGN_CENTER 1
#define EUI_ALIGN_RIGHT 2
```
컨트롤에 사용되는 text의 정렬 방식을 지정할 경우 위의 정의를 사용한다.

3. 컨트롤 생성과 이벤트 처리

컨트롤 생성

(1) 정의되어 있는 구조체 typedef 를 이용하여 만들고자 하는 컨트롤의 구조체를 선언한다.

예) 버튼만들기

```
BUTTON btn;   // BUTTON형 구조체 btn 선언.
```

(2) 컨트롤의 구조체를 초기화 한다.

예) 버튼구조체 초기화

```
InitButton(&btn); //구조체내의 각 변수들은 지정된 초기값을 가지게 된다.
```

(3) 컨트롤의 속성 및 위치를 지정한다.

예) 버튼속성 지정, 위치지정

```
btn.x=100; btn.y=50;
btn.text_color=RGB(0,0,255);
```

(4) 화면에 출력한다.

```
ViewButton(&btn,0);
```

컨트롤의 이벤트처리

터치 조작 등으로 인한 EuiP.action이 발생 했을 경우 Op함수를 실행하여 컨트롤을 동작시키고 컨트롤.event 의 내용을 확인하여 해당 event에 따르는 동작 수행

예) 버튼컨트롤의 동작 및 이벤트 수행

```
while(1)
{
  if(EuiP.action!=EUI_TOUCH_NODATA)
    {
      OpButton(&btn);
      EuiP.action=EUI_TOUCH_NODATA;

      if(btn.event==EUI_EVENT_BTN_UP)
        {  put_string1("버튼눌러짐“); btn1.event=EUI_EVENT_NO;
        }
    }
}
```

4. 사용자 함수 및 구조체

(1) Dialog

구조체

typedef struct {

unsigned int event;

OpDialog함수 수행시 이벤트가 저장되는 변수

unsigned int x;

컨트롤의 x축 위치

unsigned int y;

컨트롤의 y축 위치

unsigned int width;

컨트롤의 가로크기

unsigned int height;

컨트롤의 세로크기

unsigned char text[EUI_MAX_TEXT];

Dialog컨트롤의 타이틀바에 들어갈 문자열

unsigned int text_color;

Dialog컨트롤의 타이틀바에 들어갈 문자열의 색상

unsigned char font;

Dialog컨트롤의 타이틀바에 들어갈 문자열의 폰트

EUI_FONT5X7, EUI_FONT8X8, EUI_FONT8X16, EUI_FONTHangul이 지정될 수 있다.

unsigned char put_icon;

Dialog컨트롤의 타이틀바에 아이콘생성 유무지정변수

(1 : 아이콘 생성, 0 : 아이콘 없음)

아이콘이 생성되어 있을 경우 EUI_EVENT_DLG_ICON_DOWN ,

EUI_EVENT_DLG_ICON_UP 이벤트를 발생 시킨다.

unsigned short *icon;

Dialog컨트롤의 타이틀바에 아이콘을 생성할 경우 이미지 지정 포인트

(16 * 16 사이즈)

unsigned char put_exit_btn;

Dialog컨트롤의 타이틀바에 Exit 버튼을 생성한다.

(1 : Exit 버튼 생성, 0 : Exit 버튼 없음)

Exit 버튼이 생성되어 있을 경우 EUI_EVENT_DLG_EXITBTN_DOWN , EUI_EVENT_DLG_EXITBTN_UP 이벤트를 발생 시킨다.

} DIALOG;

사용함수

void InitDialog(DIALOG *dlg)

Dialog 컨트롤 초기화함수.

void ViewDialog(DIALOG *dlg)

Dialog 컨트롤을 화면에 표시/갱신 한다.

void OpDialog(DIALOG *dlg)

EuiP의 action에 따른 Dialog 컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_DLG_TITLEBAR_DOWN 10

타이틀바를 TouchDown 했을 경우 발생

#define EUI_EVENT_DLG_TITLEBAR_UP 11

타이틀바를 TouchUp 했을 경우 발생

#define EUI_EVENT_DLG_EXITBTN_DOWN 20

타이틀바의 Exit버튼을 TouchDown 했을 경우 발생

#define EUI_EVENT_DLG_EXITBTN_UP 21

타이틀바의 Exit버튼을 TouchUp 했을 경우 발생

#define EUI_EVENT_DLG_ICON_DOWN 30

타이틀바의 아이콘을 TouchDown 했을 경우 발생

#define EUI_EVENT_DLG_ICON_UP 31

타이틀바의 아이콘을 TouchUp 했을 경우 발생

(2) Button

구조체

typedef struct {

unsigned int event;

OpButton함수 수행시 이벤트가 저장되는 변수

unsigned int x;

컨트롤의 x축 위치

unsigned int y;

컨트롤의 y축 위치

unsigned int width;

컨트롤의 가로크기

unsigned int height;

컨트롤의 세로크기

unsigned char text[EUI_MAX_TEXT];

Button컨트롤에 들어갈 문자열

unsigned int text_color;

Button컨트롤에 들어갈 문자열의 색상
unsigned char font;
Button컨트롤에 들어갈 문자열의 폰트
EUI_FONT5X7, EUI_FONT8X8, EUI_FONT8X16, EUI_FONTHangul이 지정될 수 있다.
} BUTTON;

사용함수

void InitButton(BUTTON *btn)
Button 컨트롤 초기화함수.
void ViewButton(BUTTON *btn,unsigned char onoff)
Button 컨트롤을 화면에 표시/갱신 한다.
(onoff : 1 일 경우 눌러진 버튼표시, 0 일 경우 눌러지지않은 버튼 표시)
void OpButton(BUTTON *btn)
EuiP의 action에 따른 Button 컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_BTN_DOWN 10
Button이 눌러 졌을 경우 발생
#define EUI_EVENT_BTN_UP 11
Button이 떨어 졌을 경우 발생

(3) Edit

구조체

typedef struct {
unsigned int event;
OpEdit함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
unsigned int height;
컨트롤의 세로크기
unsigned char text[EUI_MAX_TEXT];
Edit컨트롤에 들어갈 문자열
unsigned int text_color;
Edit컨트롤에 들어갈 문자열의 색상
unsigned char font;

Edit컨트롤에 들어갈 문자열의 폰트
EUI_FONT5X7, EUI_FONT8X8, EUI_FONT8X16, EUI_FONTHangul이 지정될 수 있다.
unsigned char align;
Edit컨트롤에 들어갈 문자열의 정렬방식
EUI_ALIGN_LEFT, EUI_ALIGN_CENTER, EUI_ALIGN_RIGHT가 지정될 수 있다.
} EDIT;

사용함수

void InitEdit(EDIT *edit)
Edit컨트롤 초기화함수.
void ViewEdit(EDIT *edit)
Edit컨트롤을 화면에 표시/갱신 한다.
void OpEdit(EDIT *edit)
EuiP의 action에 따른 Edit컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_EDIT_DOWN 10
Edit컨트롤을 TouchDown 했을 경우 발생
#define EUI_EVENT_EDIT_UP 11
Edit컨트롤을 TouchUp 했을 경우 발생

(4) Progress

구조체

typedef struct {
unsigned int event;
OpProgress함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
unsigned int height;
컨트롤의 세로크기
unsigned char vertical;
Progress 컨트롤의 세로방향지정
(1 : 세로방향, 0 : 가로방향)
unsigned int color;
Progress 컨트롤의 Bar색상을 지정한다.

unsigned char value; // 0~100
Progress컨트롤의 진행정도를 0~100까지 지정한다.
} PROGRESS;

사용함수

void InitProgress(PROGRESS *prog)
Progress컨트롤 초기화함수.
void ViewProgress(PROGRESS *prog)
Progress컨트롤을 화면에 표시/갱신 한다.
void OpProgress(PROGRESS *prog)
EuiP의 action에 따른 Progress컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_PROG_DOWN 10
Progress컨트롤을 TouchDown 했을 경우 발생
#define EUI_EVENT_PROG_UP 11
Progress컨트롤을 TouchUp 했을 경우 발생

(5) Check

구조체

typedef struct {
unsigned int event;
OpCheck함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
ViewCheck함수 수행시 문자열크기만큼 크기가 지정됨.
unsigned int height;
컨트롤의 세로크기
ViewCheck함수 수행시 문자열크기만큼 크기가 지정됨.
unsigned char text[EUI_MAX_TEXT];
Edit컨트롤의 문자열을 지정
unsigned int text_color;
Edit컨트롤의 문자열 색상을 지정
unsigned char font;
Edit컨트롤의 문자열 폰트을 지정
unsigned char check;

Edit컨트롤의 Check상태를 표시 (1 : check 됨 , 0 : check 되지 않음)
} CHECK;

사용함수

void InitCheck(CHECK *chk)
Check컨트롤 초기화함수.
void ViewCheck(CHECK *chk)
Check컨트롤을 화면에 표시/갱신 한다.
void OpCheck(CHECK *chk)
EuiP의 action에 따른 Check컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_CHECK_CHANGED 41
Check컨트롤의 check상태가 변경되었을 경우 발생

(6) Static

구조체

typedef struct {
unsigned int event;
OpStatic함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
ViewStatic함수 수행시 문자열크기만큼 크기가 지정됨.
unsigned int height;
컨트롤의 세로크기
ViewStatic함수 수행시 문자열크기만큼 크기가 지정됨.
unsigned char text[EUI_MAX_TEXT];
Static컨트롤의 문자열
unsigned int text_color;
Static컨트롤의 문자열 색상
unsigned char font;
Static컨트롤의 문자열 폰트
} STATIC;

사용함수

void InitStatic(STATIC *stt)

Static컨트롤 초기화함수.
void ViewStatic(STATIC *stt)
Static컨트롤을 화면에 표시/갱신 한다.
void OpStatic(STATIC *stt)
EuiP의 action에 따른 Static컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트
#define EUI_EVENT_STATIC_DOWN 10
Static컨트롤을 TouchDown 했을 경우 발생
#define EUI_EVENT_STATIC_UP 11
Static컨트롤을 TouchUp 했을 경우 발생

(7) Picture

구조체
typedef struct {
unsigned int event;
OpPicture함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
unsigned int height;
컨트롤의 세로크기
unsigned short * image;
width * height 크기의 이미지 지정 포인터
unsigned char outline;
이미지 외곽선을 표시한다.
(1 : 외곽선 표시 , 0 : 외곽선 표시 안함)
} PICTURE;

사용함수
void InitPicture(PICTURE *pic)
Picture컨트롤 초기화함수.
void ViewPicture(PICTURE *pic)
Picture 컨트롤을 화면에 표시/갱신 한다.
void OpPicture(PICTURE *pic)
EuiP의 action에 따른 Picture컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_PICTURE_DOWN 10
Picture컨트롤을 TouchDown 했을 경우 발생
#define EUI_EVENT_PICTURE_UP 11
Picture컨트롤을 TouchUp 했을 경우 발생

(8) Slider

구조체

typedef struct {
unsigned int event;
OpSlider함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
unsigned int height;
컨트롤의 세로크기
unsigned int position;
OpSlider함수 내부적으로 사용되는 Slider 포인터 위치 변수
unsigned char value; // 0~100 read-only
OpSlider함수 수행후 현재 Slider 포인터의 위치를 0~100으로 표시
unsigned char vertical;
Slider의 세로방향을 지정
(1 : 세로방향 , 0 : 가로방향)
} SLIDER;

사용함수

void InitSlider(SLIDER *sld)
Slider컨트롤 초기화함수.
void ViewSlider(SLIDER *sld)
Slider컨트롤을 화면에 표시/갱신 한다.
void OpSlider(SLIDER *sld)
EuiP의 action에 따른 Slider컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_SLIDER_CHANGING 40
Slider 포인터가 사용자에 의해 변경되고 있을 경우 발생(TouchUp 하기전)
#define EUI_EVENT_SLIDER_CHANGED 41

Slider 포인터가 변경완료 되었을 경우 발생(TouchUp)

(9) List

구조체

typedef struct {

unsigned int event;

OpList함수 수행시 이벤트가 저장되는 변수

unsigned int x;

컨트롤의 x축 위치

unsigned int y;

컨트롤의 y축 위치

unsigned int width;

컨트롤의 가로크기

unsigned int height;

컨트롤의 세로크기

ViewList함수 수행시 LIST항목의 세로크기보다 작을 경우 LIST항목의 세로크기만큼 크기가 지정됨.

unsigned char **text;

List항목 지정 포인터

unsigned char number_of_list;

List항목 수

unsigned char select; // 0: not selected

현재 선택된 항목을 표시 (0일경우 선택되지 않음)

unsigned int text_color;

항목의 문자열 color를 지정

unsigned char font;

항목의 문자열 font를 지정

EUI_FONT5X7, EUI_FONT8X8, EUI_FONT8X16, EUI_FONTHangul이 지정될 수 있다.

unsigned char align;

항목의 문자열 정렬방식을 지정

EUI_ALIGN_LEFT, EUI_ALIGN_CENTER, EUI_ALIGN_RIGHT가 지정될 수 있다.

} LIST;

사용함수

void InitList(LIST *list)

List컨트롤 초기화함수.

void ViewList(LIST *list)

List컨트롤을 화면에 표시/갱신 한다.

void OpList(LIST *list)

EuiP의 action에 따른 List컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_LIST_CHANGING 40
List컨트롤이 사용자에 의해 변경되고 있을 경우 발생(TouchUp 하기전)
#define EUI_EVENT_LIST_CHANGED 41
List컨트롤이 변경완료 되었을 경우 발생(TouchUp)

(10) Spin

구조체

typedef struct {
unsigned int event;
OpSpin함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
unsigned int height;
컨트롤의 세로크기
unsigned int color;
Spin컨트롤의 화살표 색상을 지정
unsigned char vertical;
Spin컨트롤의 세로방향을 지정
(1 : 세로방향 , 0 : 가로방향)
} SPIN;

사용함수

void InitSpin(SPIN *spin)
Spin컨트롤 초기화함수.
void ViewSpin(SPIN *spin,unsigned char inc, unsigned char dec)
List컨트롤을 화면에 표시/갱신 한다.
inc,dec 는 Spin컨트롤의 증가/감소 버튼의 눌려진 상태를 지정한다.
(1 : 눌러짐 , 0 : 눌러지지 않음)
void OpSpin(SPIN *spin)
EuiP의 action에 따른 List컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_SPIN_INCREMENT_DOWN 10

증가버튼이 눌러 졌을 경우 발생
#define EUI_EVENT_SPIN_INCREMENT_UP 11
증가버튼이 떨어 졌을 경우 발생
#define EUI_EVENT_SPIN_DECREMENT_DOWN 20
감소버튼이 눌러 졌을 경우 발생
#define EUI_EVENT_SPIN_DECREMENT_UP 21
감소버튼이 떨어 졌을 경우 발생

(11) OwnerButton

구조체

typedef struct {
unsigned int event;
OpOwnerButton함수 수행시 이벤트가 저장되는 변수
unsigned int x;
컨트롤의 x축 위치
unsigned int y;
컨트롤의 y축 위치
unsigned int width;
컨트롤의 가로크기
unsigned int height;
컨트롤의 세로크기
unsigned short *btn_down;
OwnerButton이 눌러졌을 경우 표시될 이미지 지정 포인터
unsigned short *btn_up;
OwnerButton이 떨어졌을 경우 표시될 이미지 지정 포인터
(btn_down과 btn_up의 이미지 크기는 동일 해야 한다.)
} OWNERBUTTON;

사용함수

void InitOwnerButton(OWNERBUTTON *obtn)
OwnerButton컨트롤 초기화함수.
void ViewOwnerButton(OWNERBUTTON *obtn,unsigned char onoff)
OwnerButton컨트롤을 화면에 표시/갱신 한다.
void OpOwnerButton(OWNERBUTTON *obtn)
EuiP의 action에 따른 OwnerButton컨트롤의 동작 및 이벤트를 발생 시킨다.

이벤트

#define EUI_EVENT_OWNERBTN_DOWN 10
OwnerButton버튼이 눌러 졌을 경우 발생
#define EUI_EVENT_OWNERBTN_UP 11
OwnerButton버튼이 떨어 졌을 경우 발생