

Exercise #08 REPORT

Name: 송예린

Student ID: 2171023

1. Exercise Objective (10%)

- Extension Board 를 사용해서 LCD 를 Control 할 수 있다.
- 임베디드 시스템에서 문자열, 색상, 글자 위치가 어떻게 표현되는 지를 이해하고 Sample code 를 이용하여 원하는 이미지를 LCD 에 작성할 수 있다.
- Extension Board 에서 Push Button 을 눌러서 LCD 이미지를 이동시킬 수 있다.

1.2 이론적 배경

Extension Board 의 LCD 는 128 pixels by pixels 로 구성되어 있고 RGB 24bit 의 sequence 로 색상을 표현할 수 있다. R -8bit G -8bit B - 8bit 이고 bit 를 잘 조합하면 다양한 색상을 만들 수 있다. 문자 하나의 픽셀은 6X8 이고 표현하고자 하는 글자를 6 개의 8bit 숫자로 전달해주면 된다. 0 인 부분은 꺼지고 1 인 부분만 켜지므로 해당 위치에 1 을 주면 원하는 글자를 만들 수 있다. 색상은 총 2^{24} 개의 색상으로 표현 할 수 있고 Sample code 에서 API 로 원하는 이미지를 만들 수 있다.

텍스트의 방향도 설정할 수 있는데 sample code 에서 글자를 기준으로 up,down,left,right 를 입력하면 방향이 바뀌게 define 되어있고 배경색상, 글자색상, 각 글자의 bit 를 대신 계산해주는 API 가 있다, 드라이버란 하드웨어 디바이스를 Control 하기위한 코드들이다. 드라이버를 사용하면 소프트웨어적으로 장치를 control 하여 하드웨어적으로 출력할 수 있다.

Extension Board 에서 Push Button S1,S2 의 port 번호는 각각 P5 3 번/P3 5 번이다. 이 버튼을 input 으로 사용할 수 있다

2. Implementation (60%)

사용할 변수들을 선언해주고 display 를 초기화해준다. API 를 사용해서 LCD 의 텍스트 방향 축이 0°가 되도록 설정하고 배경색상은 하얀색, 글자색은 빨강색이 되도록 한다. 128X128 LCD 에서 문자의 비트는 6X8 이다. 문자열들은 21X16 의 X,Y 그래프로 표현 할 수 있다. 그 그래프에서 위치가 (12,0)부분에 이름을 쓰고 LCD 정가운데에 '*'이 찍히게 한다. (21X16 에서 (10,8)부분) Extension Board 에서 Push Button S1 S2 를 input 으로 사용하기 위해 P3 핀 번호 5, P5 핀번호 1 에 0 을 넣고 pull up 을 enable 하게 해준 후 각 버튼이 pull up 으로 사용되므로 p3,p5 out 에 헤딩 핀 번호를 1 로 설정한다.

Pull up 은 열렸을 때 값이 1 닫혔을 때 0 을 나타낸다 스위치가 닫히면 Pn->IN 값의 bit 가 0 이 되므로 0 이 될 때 즉 스위치가 닫혀있을 때 변수 bsNcount 가 점점 증가하다가 100000 이 되면 글자를 흰색으로 덮고 바로 위나 아래에(y-1,y+1)에 '*'을 찍는다 그리고 bsNcount 를 0 으로 바꾼다. 스위치를 눌렀다가 뗐을 때도 *이 이동해야 하므로 Pn->IN 값의 bit 가 0 이 될 때 bsNstate=DOWN;으로 바꾸고 Pn->IN 값의 bit 가 1 즉 스위치가 열렸을 때 bsNstate 가 down 이면 되면 글자를 흰색으로 덮고 바로 위나 아래에(y-1,y+1)에 *을 찍는다 한번만 실행되어야 하므로 bsNstate 을 UP 으로 바꾼다. bsNcount 는 만약 100000 이 되지 않았는데 스위치가 열렸을 때를 대비하기 위해 스위치가 열린 if 문이 실행되면 bsNcount=0 한다. 이름이 (y=1)에 적었으므로 y 의 범위는 1<y<16 이다 이 범위를 벗어나면 if 문이 실행되지 않게 한다.

3. Discussion & Conclusion (10%)

LCD 에서 이미지가 어떻게 표현되는 지를 학습하고 Extension Board 의 Push Button 를 사용하여 LCD 를 Control 해보았다. API 를 활용하여 원하는 이미지를 출력하는 법, Extension Board 의 Push Button 을 input 으로 설정해서 control 등을 알게되었다.

API 부분에서 한가지 아쉬운 점은 LCD Configuration 에서 방향 즉 LCD_ORIENTATION_UP/LEFT/DOWN/RIGHT 로 define 되어있는데 define 하지말고 각도를 90°로 나눈 0,1,2,3 을 쓰면 더 이해하기 쉽고 코드도 간단해질 것 같다 여러 사람이 사용하는 API 이므로 기준으로 혼돈을 주는 left,right 말고 각 방향의 각도로 표현한다면 더 효율적일 것이라고 생각한다.

4. Reference(s)

5. Code (20%)

FILE NAME (ex: main.c)
<pre>#include <ti/devices/msp432p4xx/inc/msp.h> #include <ti/devices/msp432p4xx/driverlib/driverlib.h> #include <ti/glib/glib.h> #include "LcdDriver/Crystallfontz128x128_ST7735.h" #include "LcdDriver/msp432p4111_classic.h" /* Graphic library context */ Graphics_Context g_sContext;</pre>

```

/*
 * Main function
 */

#define INIT -1
#define DOWN 0
#define UP 1
#define TH 100000

void main(void)
{
    int x=10, y=8;
    int bs1state=INIT;
    int bs1count=0;
    int bs2state=INIT;
    int bs2count=0;

    /* Initializes display */
    Crystalfontz128x128_Init();

    /* Set default screen orientation */
    Crystalfontz128x128_SetOrientation(LCD_ORIENTATION_UP);

    /* Initializes graphics context */
    Graphics_initContext(&g_sContext, &g_sCrystalfontz128x128, &g_sCrystalfontz128x128_funcs);
    Graphics_setForegroundColor(&g_sContext, GRAPHICS_COLOR_RED);
    Graphics_setBackgroundColor(&g_sContext, GRAPHICS_COLOR_WHITE);
    GrContextFontSet(&g_sContext, &g_sFontFixed6x8);
    Graphics_clearDisplay(&g_sContext);
    Graphics_drawString(&g_sContext,
                        (int8_t *)"Song Yelin",
                        AUTO_STRING_LENGTH,
                        12,
                        0,
                        OPAQUE_TEXT);

    /* Configuring S1 & S2 buttons in boostxl-edumkii */
    P3->DIR &= ~BIT5;
    P5->DIR &= ~BIT1;
    P3->REN |= BIT5;
    P5->REN |= BIT1;
    P3->OUT |= BIT5;
    P5->OUT |= BIT1;

    Graphics_drawString(&g_sContext, (int8_t *)"*", 1, x*6, y*7, OPAQUE_TEXT);

    while(1)
    {

```

```

if (!(P5->IN & BIT1)) {
    if ((++bs1count) == TH) {
        bs1count=0;
        Graphics_drawString(&g_sContext, (int8_t *)" ", 1, x*6, y*7, OPAQUE_TEXT);
        if (y>1) y--;
        Graphics_drawString(&g_sContext, (int8_t *)"*", 1, x*6, y*7, OPAQUE_TEXT);
    }
    bs1state=DOWN;
}
else if (P5->IN & BIT1){
    if (bs1state==DOWN) {
        bs1count=0;
        Graphics_drawString(&g_sContext, (int8_t *)" ", 1, x*6, y*7, OPAQUE_TEXT);
        if (y>1) y--;
        Graphics_drawString(&g_sContext, (int8_t *)"*", 1, x*6, y*7, OPAQUE_TEXT);
        bs1state=UP;
    }
}
if (!(P3->IN & BIT5)) {
    if ((++bs2count) == TH) {
        bs2count=0;
        Graphics_drawString(&g_sContext, (int8_t *)" ", 1, x*6, y*7, OPAQUE_TEXT);
        if (y<16) y++;
        Graphics_drawString(&g_sContext, (int8_t *)"*", 1, x*6, y*7, OPAQUE_TEXT);
    }
    bs2state=DOWN;
}
else if (P3->IN & BIT5){
    if (bs2state==DOWN) {
        bs2count=0;
        Graphics_drawString(&g_sContext, (int8_t *)" ", 1, x*6, y*7, OPAQUE_TEXT);
        if (y<16) y++;
        Graphics_drawString(&g_sContext, (int8_t *)"*", 1, x*6, y*7, OPAQUE_TEXT);
        bs2state=UP;
    }
}
}
}

```

FILE NAME (ex: main2.c)

A=b;

C=d;

...