

3차 필수프로젝트 결과보고서

2024-50924 이상욱

목차

1. 개요
 - A. 프로젝트 목적
 - B. 구현 기능
2. 주요 코드 구성
3. CubeIDE IOC 화면(pin 구성)

1. 개요 및 기능 설명

A. 프로젝트 목적

STM32F401CC를 사용하여 ATtiny로 작동시켰던 여러 하드웨어들을 작동시킨다.

추가로 Addressable LED, ILI9341 display를 활용하여 여러 기능을 구현한다.

B. 구현기능

Timer Encoder를 활용하여 Rotary switch를 돌릴 때마다 값을 0부터 23까지 증감시킨다. 그 값에 따라 Addressable LED가 위치에 맞게 움직이며 색상이 변화하고 SH1106의 bar 길이가 증가하고 MAX7219는 두 자리 수 값을 출력한다.

ILI9341에 RGB gradation을 구현한다. DMA를 활용하여 render와 transmit 과정을 효율적으로 하였고, Gamma Correction 기법을 활용하여 이미지를 보다 자연스럽게 하였다. Sprite Rendering을 활용하여 전체 이미지에서 특정 위치만 출력할 수 있게 하였다.

2. 주요 코드 구성

● Core/Inc header file 구성

1. fonts.h
2. ili9341.h
3. main.h
4. spi_allRun.h
5. sprite_data.h
6. stm32f4xx_hal_conf.h
7. stm32f4xx_it.h
8. test_tilemap.h
9. testing.h
10. village_tileset.h

● Core/Src source file 구성

1. fonts.c
2. ili9341.c
3. main.c
4. render.c
5. spi_allRun.c
6. stm32f4xx_hal_msp.c
7. stm32f4xx_it.c
8. syscalls.c
9. sysmem.c
10. system_stm32f4xx.c

11. spi_allRun.h에서는 spi_allRun.c의 함수 및 사용되는 변수들이 선언되어 있다.

Spi_allRun.c는 rotary switch의 값을 받아 MAX7219, SH1106, Addressable LED를 조작하는 모든 함수들과 예제가 저장되어 있다. main.c에서는 spi_all_init()함수와 spi_all_run()함수를 실행하면

첨부한 영상과 같이 동작한다.

ili9341.h는 ili9341.c와 관련된 define들이 저장되어 있다. ili9341.c에는 화면에 한 pixel만 그리거나 사각형을 그리거나 image를 그릴 수 있고 색상 반전을 할 수 있는 함수들이 포함되어 있다. draw_line_gradation(), end_of_frame_gradation() 함수를 사용하면 RGB gradation을 구현할 수 있다. main.c는 다음과 같이 구성된다.

```
static uint8_t line_buf[2][ILI9341_WIDTH*2]={0};
uint32_t line_num=0;
ILI9341_Init();
while (1)
{
    unsigned buf_num;
    buf_num = line_num%2;
    ILI9341_Select();
    draw_line_gradation(line_num, line_buf[buf_num]); //line_buf <= value
    if(HAL_SPI_GetState(&hspi1) == HAL_SPI_STATE_READY){
        if (line_num==0) {
            // ILI9341_SetAddressWindow setting full screen
            ILI9341_SetAddressWindow(0,0,320-1,240-1);
        }
        HAL_GPIO_WritePin(ILI9341_DC_GPIO_Port, ILI9341_DC_Pin, 1);
        HAL_SPI_Transmit_DMA(&hspi1, line_buf[buf_num], ILI9341_WIDTH*2);

        if (++line_num == ILI9341_HEIGHT) {
            line_num=0;
            end_of_frame_gradation();
        }
    }
}
```

sprite render와 관련된 이미지들은 sprite_data.h, test_tilemap.h, village_tileset.h에 저장되어 있고 render.c에서 이미지를 처리한다. main.c에 구성은 다음과 같다.

```
static uint8_t line_buf[2][ILI9341_WIDTH*2]={0};
uint32_t line_num=0;
ILI9341_Init();
setup_entities();
setup_tilemap();
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    unsigned buf_num;
    buf_num = line_num%2;
    ILI9341_Select();

    draw_line(line_num, line_buf[buf_num]); //line_buf <= value
    if(HAL_SPI_GetState(&hspi1) == HAL_SPI_STATE_READY){
        if (line_num==0) {
            // ILI9341_SetAddressWindow setting full screen
            ILI9341_SetAddressWindow(0,0,320-1,240-1);
        }
        HAL_GPIO_WritePin(ILI9341_DC_GPIO_Port, ILI9341_DC_Pin, 1);
        //HAL_SPI_Transmit(&hspi1, line_buf, ILI9341_WIDTH*2, 20);
        HAL_SPI_Transmit_DMA(&hspi1, line_buf[buf_num], ILI9341_WIDTH*2);

        if (++line_num == ILI9341_HEIGHT) {
            line_num=0;
            end_of_frame();
        }
    }
}
```

DMA 및 buffer 두 개를 활용하여 전송속도를 빠르게 하였다.

3. CubeIDE IOC 화면(pin구성)

