

임베디드 시스템 설계 및 실험 10주차 결과 보고서

7조 이석우, 손수민, 정지용, 김춘수

1. 실험 목표

- TFT LCD의 원리와 동작방법 이해
- 조도 센서의 원리와 이해 및 TFT LCD제어

2. 배경 지식

2.1 실험에 필요한 용어와 장치 설명

TFT LCD

초 박막 액정표시장치로 액체와 고체의 중간 특성을 가진 액정의 상태 변화와 편광판의 편광 성질을 이용하여 통과하는 빛의 양을 조절함으로써 정보를 표시한다. RGB 픽셀이 유리판에 코팅되어 컬러 영상을 구현하는 Color Filter 와 액정을 제어하기 위해 초박형 유리 기판 위에 반도체 막을 형성한 회로인 TFT 기판, Filter와 기판 사이에 주입된 액정과 광원인 Black light unit으로 구성 되어있다.

TFT LCD Timing

TFT LCD는 사용하는 인터페이스에 따라 timing 특성이 바뀐다. 이번 실험에서는 parallel 8080 series interface를 이용한다.

- Parallel 8080 series interface의 제어신호

- CS (Chip Select)

high에서 low일 때, chip이 동작되도록 하는 신호

- D/C (RS핀)

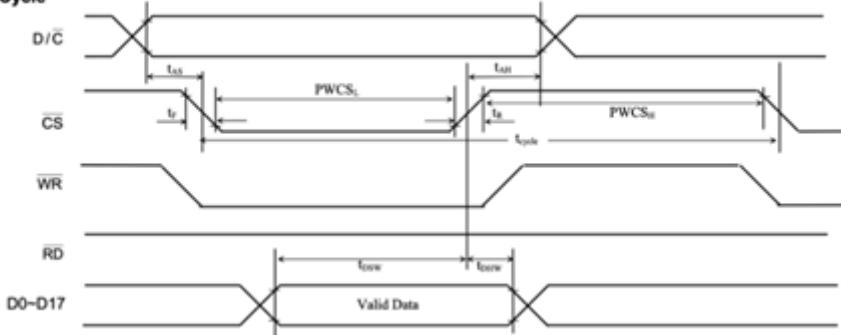
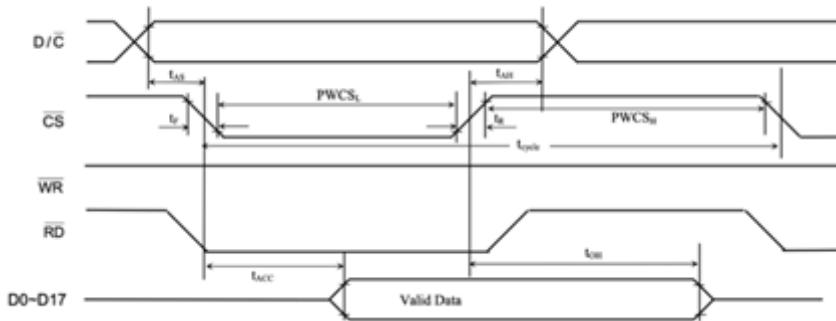
low시에 command를 전송하고, high시에 data를 전송한다.

- WR

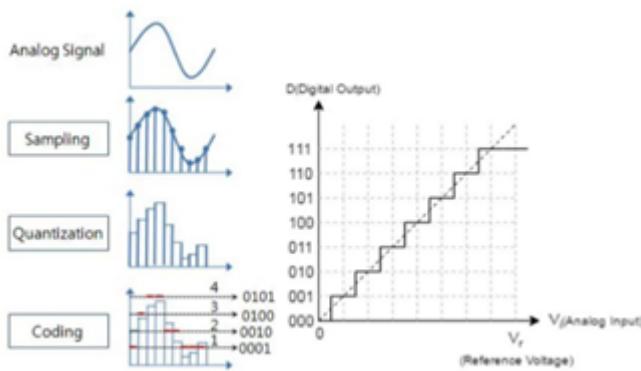
high에서 low로 변할 때, data를 display에 write한다.

- RD

high에서 low로 변할 때, data를 display로부터 read한다.

Write Cycle**Read Cycle****ADC (Analog to Digital Conversion)**

ADC는 온도, 습도, 조도 등의 analog 물리량을 digital신호로 변환하는 과정이다. 물리적 양을 측정하는 센서의 값들은 analog값이다. 이 값을 장치 내에서 인식하기 위해서는 디지털 샘플링 과정을 거쳐야 하는데 이를 adc라고 한다.

**조도센서**

주변의 밝기를 측정하는 센서로, 빛의 양이 많아질수록 전도율이 높아져 저항이 낮아진다. Adc와 조도 센서에 해당하는 클락을 제대로 열어주지 않으면 작동하지 않으므로 주의가 필요하다.

**3. 실험 과정**

lcd.c

```
static void LCD_WR_REG(uint16_t LCD_Reg)
{
    GPIO_ResetBits(GPIOC, GPIO_Pin_6); // LCD_CS(0);
    GPIO_ResetBits(GPIOD, GPIO_Pin_13); // LCD_RS(0);
    GPIO_ResetBits(GPIOD, GPIO_Pin_14); // LCD_WR(0);
    GPIO_ResetBits(GPIOC, GPIO_Pin_6);

    GPIO_Write(GPIOE, LCD_Reg);
    GPIO_SetBits(GPIOD, GPIO_Pin_14); // LCD_WR(1);
    GPIO_SetBits(GPIOC, GPIO_Pin_6); // LCD_CS(1);
}
```

LCD_CS, LCS_RS, LCD_WR을 Low 상태로 둔 뒤, 전송. 그리고 WR과 CS를 High로 둔다

```
static void LCD_WR_DATA(uint16_t LCD_Data)
{
    GPIO_ResetBits(GPIOC, GPIO_Pin_6); // LCD_CS(0);
    GPIO_SetBits(GPIOD, GPIO_Pin_13); // LCD_RS(1);
    GPIO_ResetBits(GPIOD, GPIO_Pin_14); // LCD_WR(0);
    GPIO_Write(GPIOE, LCD_Data);

    GPIO_SetBits(GPIOD, GPIO_Pin_14); // LCD_WR(1);
    GPIO_SetBits(GPIOC, GPIO_Pin_6); // LCD_CS(1);
}
```

LCD_CS, LCD_WR을 Low 상태로 LCD_RS를 High로 둔 뒤, 전송. 그리고 WR과 CS를 High로 둔다

main.c

먼저 rcc_configure함수로 rcc관련 설정을 한다. 사용할 핀을 인가하고 이전 실험과 마찬가지로 GPIO, NVIC에 대한 설정을 해준다. ADC 값을 읽기 위해서는 인터럽트를 이용하기 때문에 별도의 핸들러 함수를 작성해준다.

```
LCD_ShowString(10, 10, "THU_Team07", BLACK, WHITE);

while (1) {
    // TODO: implement

    Touch_GetXY(&x1, &y1, 1);
    Convert_Pos(x1, y1, &x1, &y1);

    LCD_ShowNum(50, 50, x1, 4, BLACK, WHITE);
    LCD_ShowNum(50, 70, y1, 4, BLACK, WHITE);

    ADC_ITConfig(ADC1, ADC_IT_EOC, ENABLE);
```

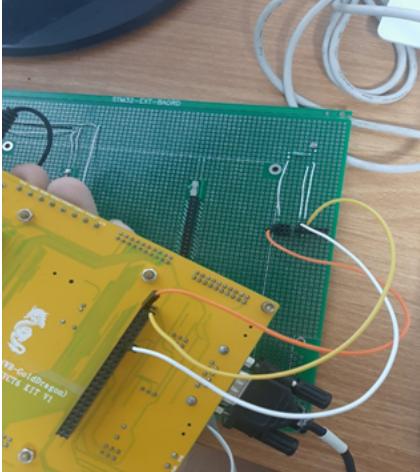
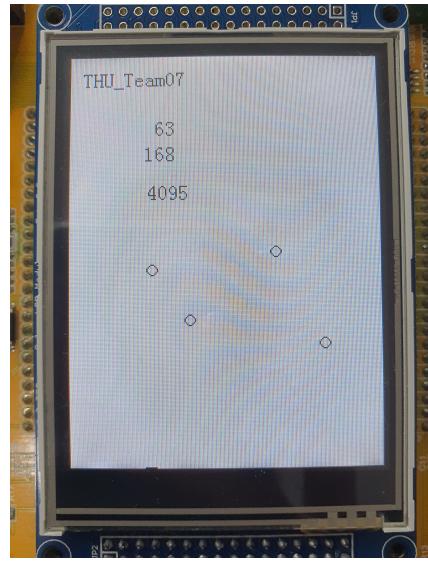
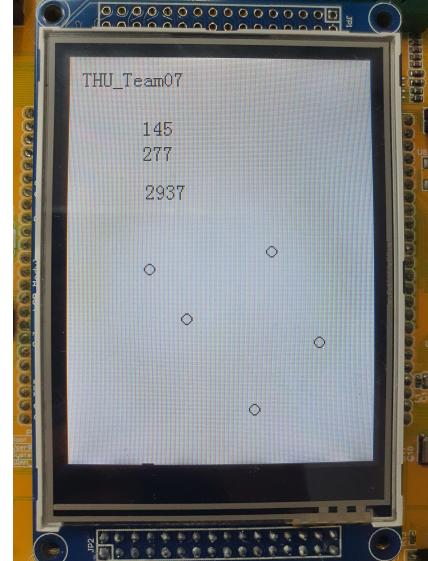
```

LCD_ShowNum(60,100, lightSenser, 4, BLACK, WHITE);
ADC_ITConfig(ADC1,ADC_IT_EOC,DISABLE);
LCD_DrawCircle(x1, y1, 4);
}
return 0;
}

```

While문 진입 전 LCD_showString를 통해 팀을 출력해주고 While문 내에서 터치를 통해 x, y 값을 입력 받으면 값을 출력한다. 그리고 LCD_DrawCircle 함수를 통해 받은 x, y 위치에 지정된 반지름 만큼의 크기를 가진 원을 출력해낸다.

4. 실험 사진

		
조도센서 연결	디스플레이 출력	조도센서를 가렸을 때

5. 결론

이번 미션에서 주어진 lcd.c 파일작성과 while 문작성은 어렵지 않았다. 이전 실험들과 마찬가지로 설정에 대해 파악한 후, 주어진 함수들을 적절히 이용해 미션을 수행해 나갔다. 다만 함수의 세부적인 이해가 부족하여, 화면 조정 및 좌표를 받는 것, 조도 센서를 연결하여 아날로그 신호를 디지털로 전환하는 것에 어려움이 있었다. 초반에는 하드웨어 구성을 잘못하였다고 착각하여, 새로운 브레드보드와 조도 센서를 준비하려 하였으나, GPIO의 설정 변경 및 핀 위치를 조정해주니 예상대로 작동하였다.