

실험 결과 보고서

10 주차 LCD 및 ADC

실험날짜: 2024-11-15

실험자:3 조(강태진, 김기윤, 김도완, 임성표)

목차

| | | |
|----|------------|---|
| 1. | 실험 목적..... | 3 |
| 2. | 세부 목표..... | 3 |
| 3. | 실험 과정..... | 3 |
| 4. | 실험 결과..... | 9 |
| 5. | 참고 문헌..... | 9 |

1. 실험 목적

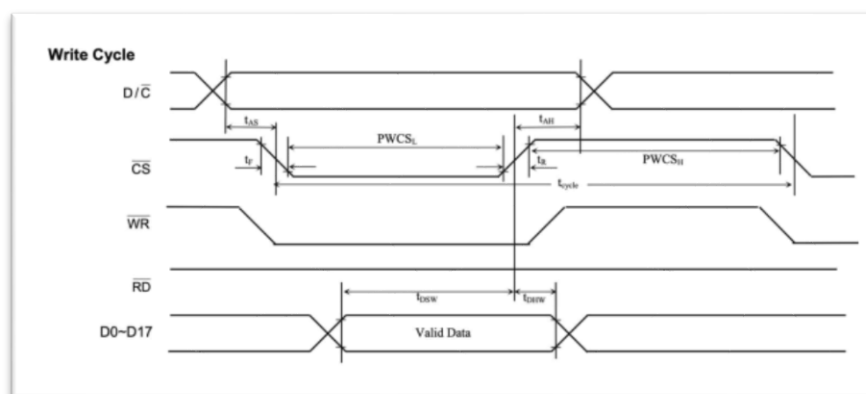
- 1) TFT-LCD 의 원리와 동작 방법에 대한 이해
- 2) TFT-LCD 라이브러리 작성과 이해
- 3) TFT-LCD Touch 동작제어
- 4) ADC 개념 이해
- 5) 조도센서 사용 방법 학습

2. 세부 목표

- 1) TFT-LCD 보드에 결착
- 2) Lcd.c 에서 write 관련 코드 작성
- 3) TFT_LCD 에 Text(팀명) 출력
- 4) ADC channel 과 인터럽트를 사용하여 조도 센서 값을 전역 변수에 저장
- 5) LCD 터치 시 해당 위치에 작은 원을 그리고 좌표(X,Y), 전역변수에 저장했던 조도 센서 값 출력

3. 실험 과정

- (1) 배경 지식 이해



- 1) Timing Diagram: TFT-LCD 에 데이터를 쓸 때 쓰기 신호, 데이터 신호 등 신호가 시간에 따라 어떻게 변화하는지 나타낸 그래프

-Write Cycle: 마이크로 컨트롤러가 LCD 로 데이터를 전달할 때 필요한 신호와 타이밍을 정의한 주기

-D/C: MCU 가 LCD 에 데이터를 전송(high) 또는 동작을 제어(low)

-CS: CS 신호가 low 가 되면 LCD 가 MCU 의 명령을 받을 준비, high 일 때 외부 명령 무시

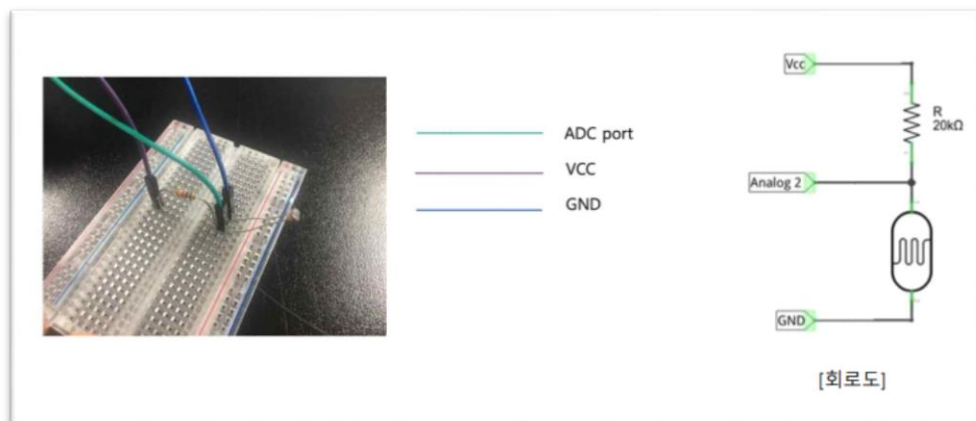
-WR: low 상태가 되면 Write 작업이 시작되고, high 상태가 되면 작업이 완료

-D0~D17: 실제 LCD 에 기록할 데이터가 전송됨

-Command 전송: D/C, CS, WR 를 low 로 두고 전송, 이 후 CS, WR 을 high 로 돌려놓는다.

-Data 전송: D/C 를 high, CS, WR 를 low 로 두고 전송, 이 후 CS, WR 를 high 로 돌려놓는다.

2)조도센서



-주변의 밝기를 측정하는 센서, 빛의 양이 많을 때는 저항이 낮아지고 적을 때는 저항이 높아짐

-저항의 변화는 아날로그 신호로 나타나며, ADC 를 통해 디지털 신호로 변환한다.

-MCU 는 ADC 에서 변환된 값을 받아 LCD 에 출력

(2)코드 작성

1) LCD_WR_REG(uint16_t LCD_Reg)/ LCD_WR_DATA(uint16_t LCD_Data)

```
static void LCD_WR_REG(uint16_t LCD_Reg)
{
    // TODO implement using GPIO_ResetBits/GPIO_SetBits
    GPIO_ResetBits(GPIOD, GPIO_Pin_13); // LCD_RS(0);
    GPIO_ResetBits(GPIOB, GPIO_Pin_14); // LCD_WR(0);
    GPIO_ResetBits(GPIOC, GPIO_Pin_8); // LCD_CS(0);

    GPIO_Write(GPIOE, LCD_Reg);
    // TODO implement using GPIO_ResetBits/GPIO_SetBits
    GPIO_SetBits(GPIOC, GPIO_Pin_8); // LCD_CS(1);
    GPIO_SetBits(GPIOB, GPIO_Pin_14); // LCD_WR(1);
}
```

- LCD_RS, LCD_WR, LCD_CS 를 0 으로 설정하여 명령어 전송을 준비한다.
- GPIO_Write(GPIOE, LCD_Reg): LCD_Reg 변수에 저장된 명령어 값을 GPIOE 포트에 쓰는데, LCD 에 전달할 명령어 데이터가 된다.
- 전송이 완료되면 LCD_CS, LCD_WR 를 1 로 설정하여 핀을 초기화 시킨다.

```
static void LCD_WR_DATA(uint16_t LCD_Data)
{
    // TODO implement using GPIO_ResetBits/GPIO_SetBits
    GPIO_SetBits(GPIOD, GPIO_Pin_13); // LCD_RS(1);
    GPIO_ResetBits(GPIOB, GPIO_Pin_14); // LCD_WR(0);
    GPIO_ResetBits(GPIOC, GPIO_Pin_8); // LCD_CS(0);

    GPIO_Write(GPIOE, LCD_Data);
    // TODO implement using GPIO_ResetBits/GPIO_SetBits
    GPIO_SetBits(GPIOC, GPIO_Pin_8); // LCD_CS(1);
    GPIO_SetBits(GPIOB, GPIO_Pin_14); // LCD_WR(1);
}
```

- LCD_RS 를 1, LCD_WR, LCD_CS 를 0 으로 설정하여 데이터 전송을 준비한다.
- GPIO_Write(GPIOE, LCD_Data): GPIOE 포트를 통해 LCD_Data 에 저장된 데이터를 LCD 로 전송한다.
- 전송이 완료되면 LCD_CS, LCD_WR 를 1 로 설정하여 핀을 초기화 한다.

2) RccInit

```
void RccInit(void) {
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
}
```

- RCC_APB2Periph_ADC1 을 사용해 ADC1 활성화
- RCC_APB2Periph_GPIOB 을 사용해 GPIOB 활성화

3) GpioInit

```
void GpioInit(void) {
    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_10MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
    GPIO_Init(GPIOB, &GPIO_InitStructure);
}
```

- GPIO_Pin_0: 핀 번호 0 을 지정
- GPIO_Speed_10MHz: 핀의 속도를 10Mhz 로 설정
- GPIO_Mode_AIN: GPIO 모드를 아날로그 입력으로 설정하여, 0 번핀에서 아날로그 데이터를 읽을 수 있도록 한다.

4) NvicInit

```
void NvicInit(void) {
    NVIC_InitTypeDef NVIC_InitStructure;
    NVIC_InitStructure.NVIC_IRQChannel = ADC1_2_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
```

- NVIC 에서 ADC1, ADC2 의 인터럽트를 처리하도록 설정
- 우선순위를 0 으로 설정하여 가장 높은 우선순위를 부여하고 활성화한다.

5)AdcInit

```

void AdcInit(void) {
    ADC_InitTypeDef ADC_InitStructure;
    ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
    ADC_InitStructure.ADC_ScanConvMode = DISABLE;
    ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
    ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
    ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
    ADC_InitStructure.ADC_NbrOfChannel = 1;
    ADC_Init(ADC1, &ADC_InitStructure);

    ADC_RegularChannelConfig(ADC1, ADC_Channel_8, 1, ADC_SampleTime_28Cycles5);
    ADC_ITConfig(ADC1, ADC_IT_EOC, ENABLE);
    ADC_Cmd(ADC1, ENABLE);

    ADC_ResetCalibration(ADC1);
    while(ADC_GetResetCalibrationStatus(ADC1)) {}

    ADC_StartCalibration(ADC1);
    while(ADC_GetCalibrationStatus(ADC1)) {}

    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
}

```

-ADC1 의 설정 및 초기화

6) ADC1_2_IQRHandler

```

void ADC1_2_IQRHandler() {
    if (ADC_GetITStatus(ADC1, ADC_IT_EOC) != RESET) {
        value = ADC_GetConversionValue(ADC1);
        ADC_ClearITPendingBit(ADC1, ADC_IT_EOC);
    }
}

```

-ADC_GetITStatus: ADC 변환이 완료되었는지 확인

-변환된 값을 읽어서 value 에 저장

-ADC_ClearITPendingBit: 인터럽트 플래그를 초기화하여 다음 인터럽트를 받는다.

-GPIOB 핀 0 에서 데이터를 연속으로 변환하고, 변환이 완료될 때마다 인터럽트를 통해 값을 처리할 수 있다.

7)Init

```
void Init(void) {
    SystemInit();
    RccInit();
    GpioInit();
    AdcInit();
    NvicInit();

    LCD_Init();
    Touch_Configuration();
    Touch_Adjust();
}
```

-LCD_Init: LCD 디스플레이 초기화 함수

-Touch_Configuration: 터치 스크린 하드웨어 설정 및 초기화

-Touch_Adjust: 터치 입력의 정확성을 위한 보정작업 수행

8)main

```
int main() {
    Init();
    LCD_Clear(WHITE);

    uint16_t x = 30, y = 0;
    LCD_ShowString(100, 20, "TEAM_03", BLACK, WHITE);
    x = 90;
    y += 25;

    uint16_t touch_x, touch_y;
    while (1) {
        Touch_GetXY(&touch_x, &touch_y, 1);

        Convert_Pos(touch_x, touch_y, &touch_x, &touch_y);

        // Draw a circle at the touched location
        LCD_DrawCircle(touch_x, touch_y, 5);
        LCD_ShowNum(100, 120, touch_x, 3, BLACK, WHITE);
        LCD_ShowNum(100, 150, touch_y, 3, BLACK, WHITE);
        LCD_ShowNum(100, 170, value, 4, BLACK, WHITE);
    }
}
```

-LCD_Clear(WHITE): LCD 화면을 흰색으로 초기화

-LCD_ShowString: 좌표(100,20)에 텍스트 표시

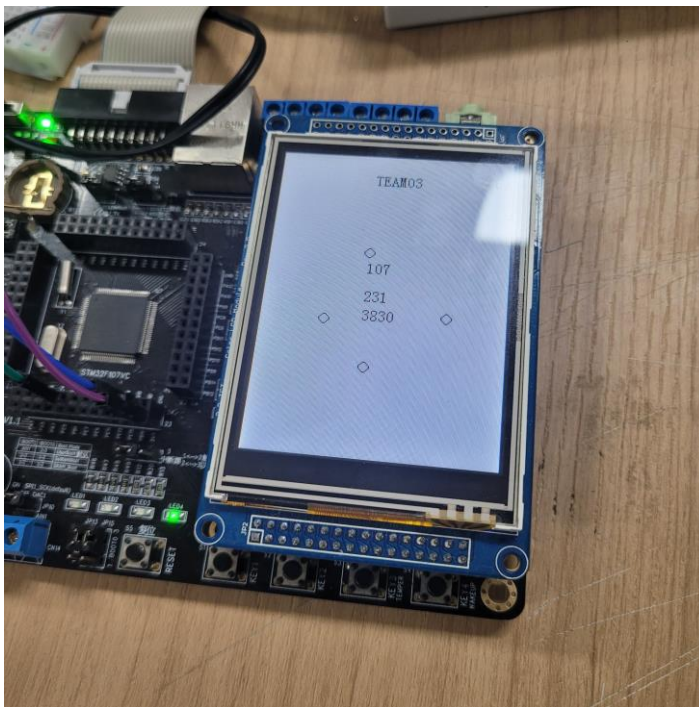
-Touch_GetXY: 현재 터치된 좌표를 touch_x, touch_y 에 저장

-Conver_Pos: 터치 좌표를 LCD 에 맞게 변환

_LCD_ShowNum: touch_x, touch_y, value(ADC)값을 화면에 출력

-터치스크린 입력을 기반으로 원을 그리고, 터치 좌표와 ADC 값을 화면에 표시

4. 실험 결과



-LCD 화면에 텍스트 표기, 터치에 따른 원 생성 및 ADC, x, y 좌표 표기

5. 참고 문헌

1) 10 주차 LCD 및 ADC 강의자료