

11월 13일 예비 발표 준비 내용

- Timer에 관하여 자세하게
 - STM32의 Timer 종류 및 각각의 기능
 - STM32의 Timer 회로 및 동작 방법
 - TIM2에 설정한 interrupt가 언제 발생하는지?
- 분주 계산 방법에 관하여
- PWM에 대하여



Pusan
National
University

임베디드 시스템 설계 및 실험

수요일 분반 LCD 및 ADC

조교
이제형

dlwpgud9684@gmail.com

Contents

1. 실험 목적
2. 실험 내용
3. 실험 과정
4. 실험 주의사항
5. 실험 미션

Contents

1. 실험 목적

LCD 및 ADC

- TFT-LCD의 원리와 동작 방법에 대한 이해
- TFT-LCD 라이브러리 작성과 이해
- TFT-LCD Touch 동작 제어
- ADC 개념 이해
- 조도 센서 사용 방법 학습

Contents

2. 실험 내용

2-1. TFT-LCD에 text 출력

TFT-LCD(Thin Film Transistor Liquid Crystal Display)?

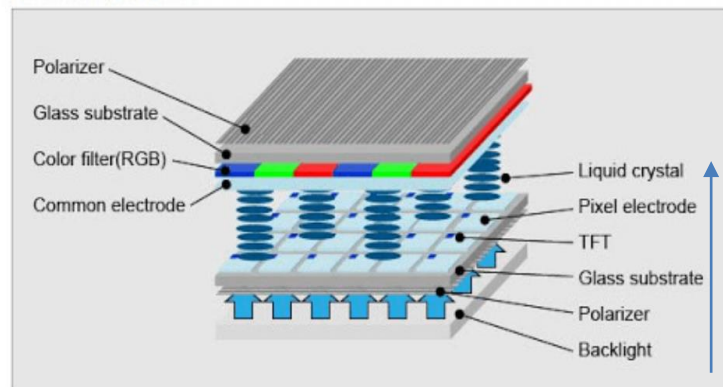
- “초 박막 액정표시장치”

- 액정과 박막 트랜지스터(TFT)를 사용해 전자 기기 화면에 이미지를 표시하는 기술
- 주로 스마트폰, 모니터 등 디스플레이 장치에 사용됨

- 구성 요소

- 백라이트 유닛
 - LCD 자체는 빛을 내지 않기 때문에 뒤에서 빛을 비춰주는 광원이 필요
 - 백라이트 유닛은 LCD 뒤에서 빛을 공급해 화면이 잘 보이도록 함
- 편광판(Polarizer)
 - 빛의 특정 방향만 통과시키는 필터로 액정의 상태 변화에 따라 빛의 양을 조절하여 화면에 정보를 표현하는데 사용
- TFT 기판
 - 픽셀을 개별적으로 제어하기 위한 회로
- 액정(Liquid Crystal)
 - 액체와 고체의 중간 상태를 가진 물질로, 전기 신호에 따라 상태를 바꾸면서 빛의 투과량 조절
- RGB 컬러 필터
 - RGB픽셀이 코팅된 필터로, 각각의 픽셀이 빛을 통과하면서 컬러 화면 형성

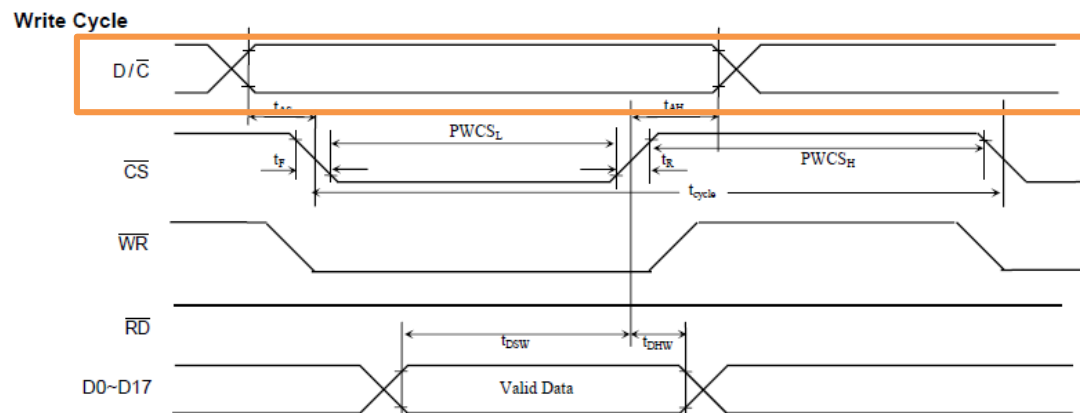
Structure of a TFT LCD



Timing Diagram

Timing Diagram

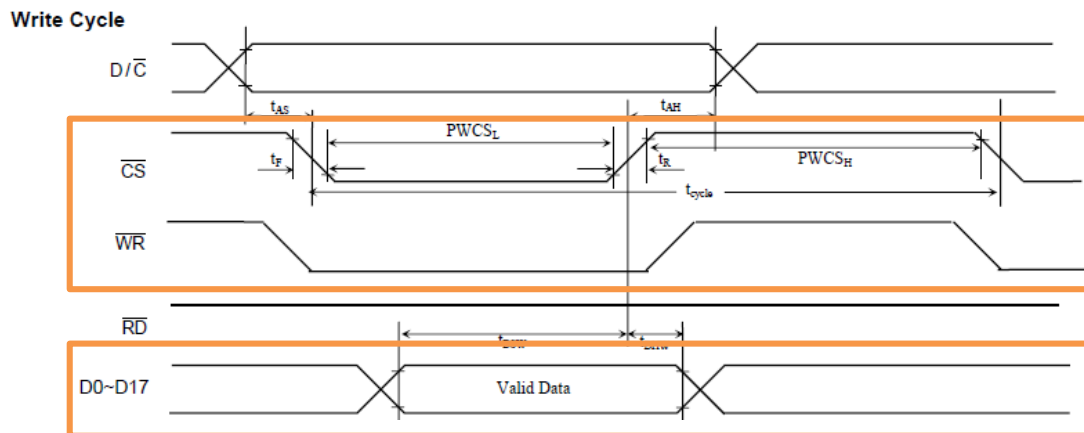
- TFT-LCD에 데이터를 쓸 때 쓰기 신호, 데이터 신호 등 신호 간의 순서와 시간 차이를 맞춰야 작동
 - **Timing Diagram:** 이러한 신호들이 시스템에서 시간에 따라 어떻게 변화하는지 시각적으로 나타낸 그래프
 - **Write Cycle:** 마이크로 컨트롤러가 LCD로 데이터를 전달할 때 필요한 신호와 타이밍을 정의한 주기
 - **D/C:** Data/Command 구분 신호
- MCU가 LCD에 데이터를 전송할지(high) 또는 LCD에 동작을 제어함을 알릴지(low) 설정



Timing Diagram

Timing Diagram

- **Write Cycle:** 마이크로 컨트롤러가 LCD로 데이터를 전달할 때 필요한 신호와 타이밍을 정의한 주기
 - **CS:** CS(Chip Select) 신호가 활성화(low)되어야 LCD가 MCU의 명령을 받을 준비를 함. 비활성화(high) 상태라면 LCD는 외부 명령을 무시
 - **WR:** low 상태가 되면 Write 작업이 시작되고, 이때 MCU는 데이터를 LCD에 전달할 수 있음. 신호가 high 상태로 돌아가면 작업이 완료.
 - **D0~D17(데이터 라인):** 실제 LCD에 기록할 데이터가 전송됨



Timing Diagram

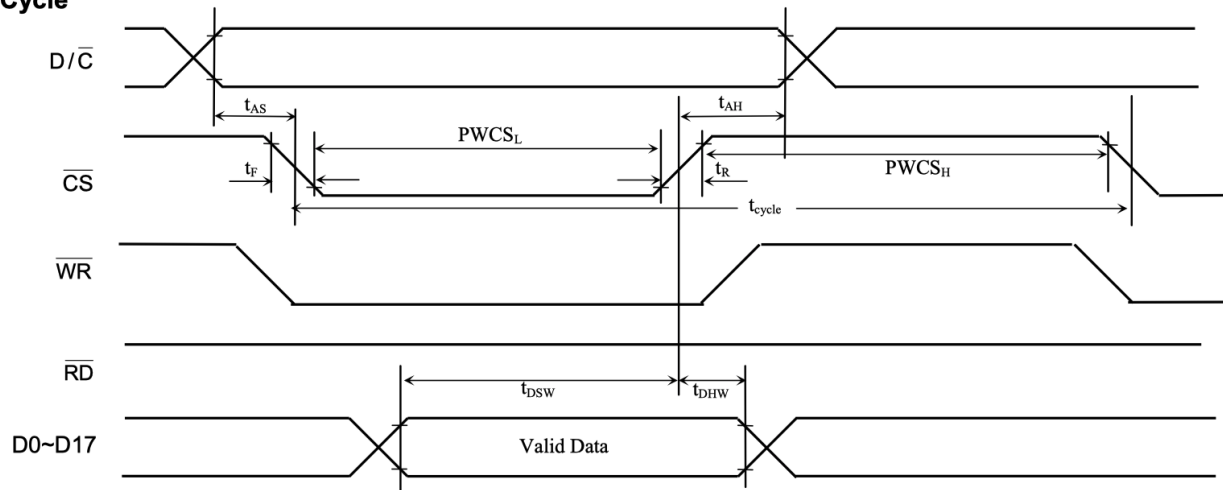
Timing Diagram

- Ex) LCD에 text 디스플레이 하기

1. COMMAND

- D/\bar{C} 를 Low(명령 모드), \overline{CS} 를 Low(chipset 활성화), \overline{WR} 를 Low(write 작업시작)로 두고 Command를 전송
- \overline{CS} 를 High, \overline{WR} 를 High로 다시 돌려놓기

Write Cycle



Timing Diagram

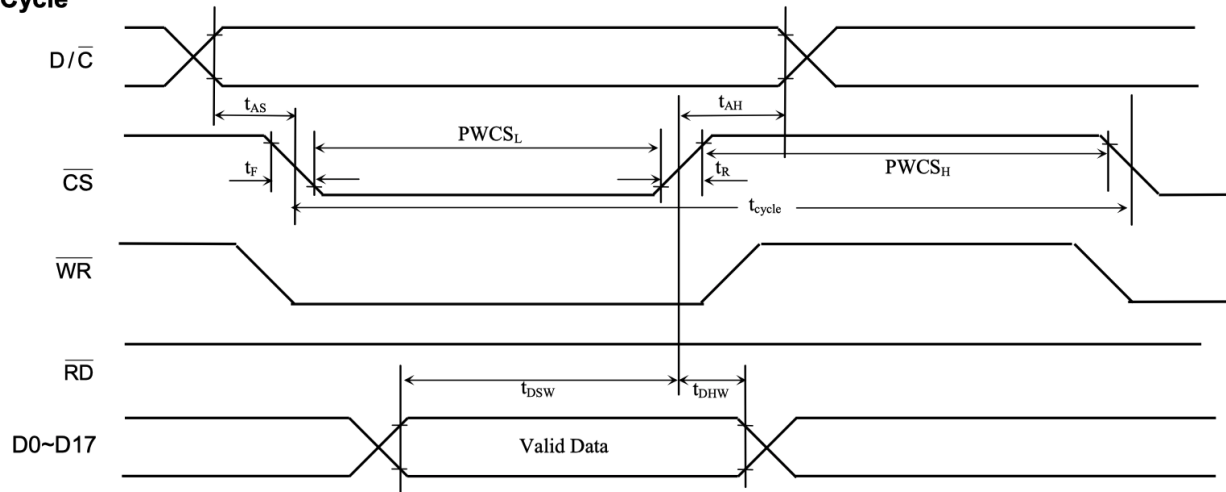
Timing Diagram

- Ex) LCD에 text 디스플레이 하기

2. DATA

- D/\bar{C} 를 High(데이터 모드), \overline{CS} 를 Low, \overline{WR} 를 Low로 두고 Data를 Display에 전송
- \overline{CS} 를 High, \overline{WR} 를 High로 다시 돌려놓기

Write Cycle

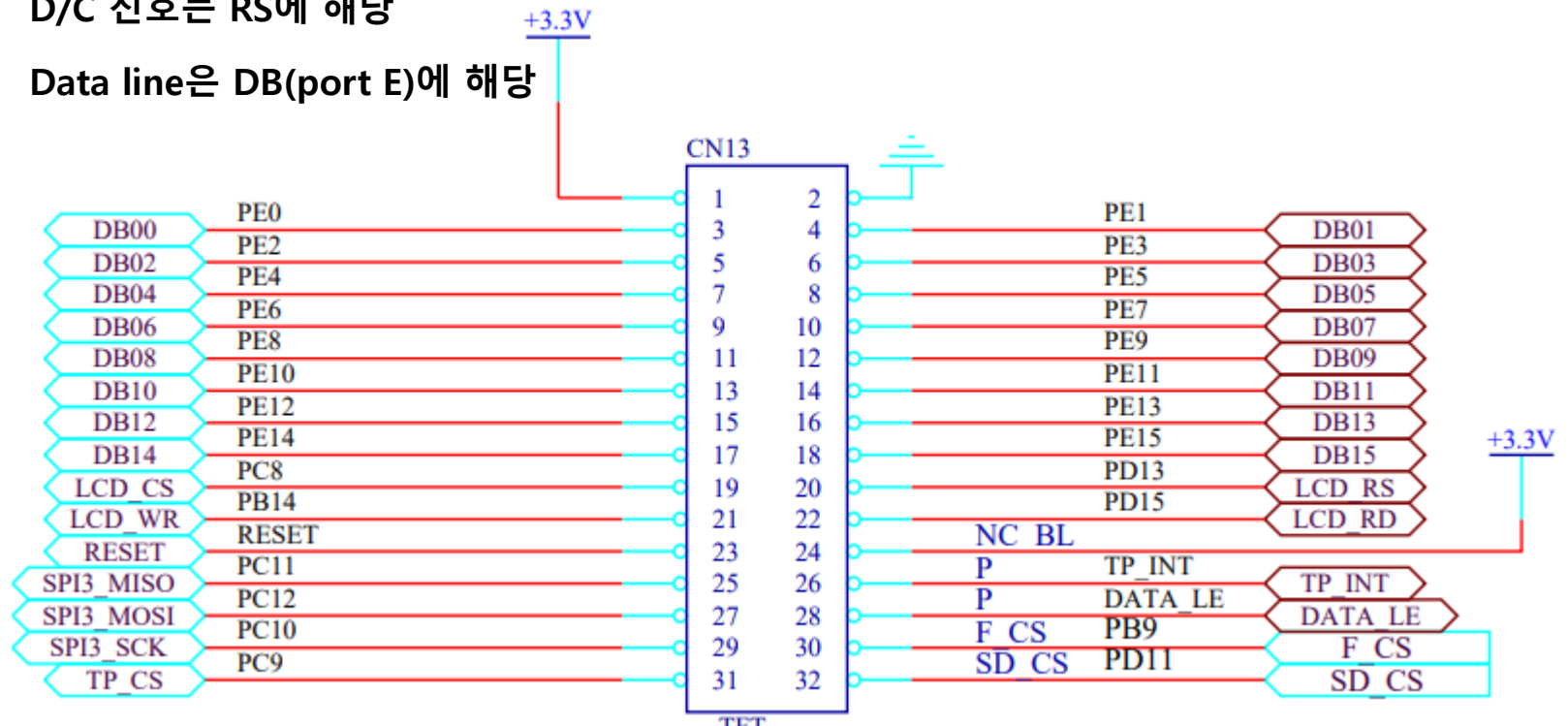


Timing Diagram

Timing Diagram

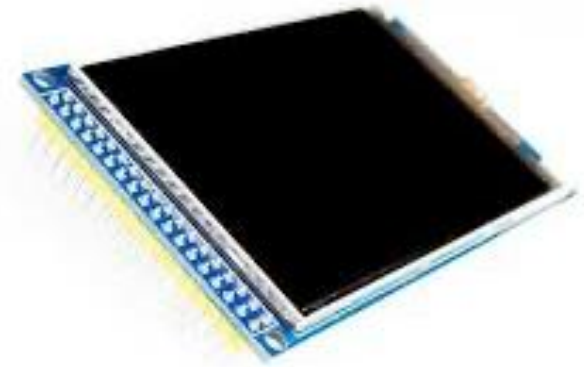
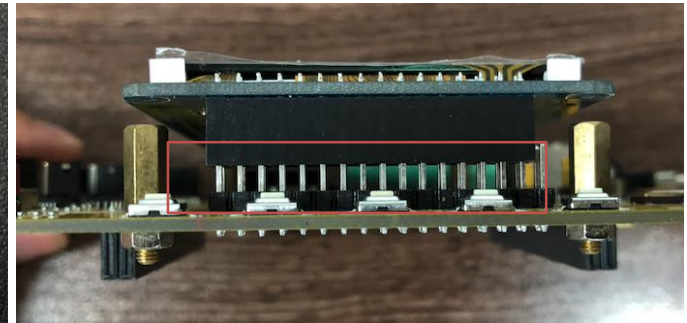
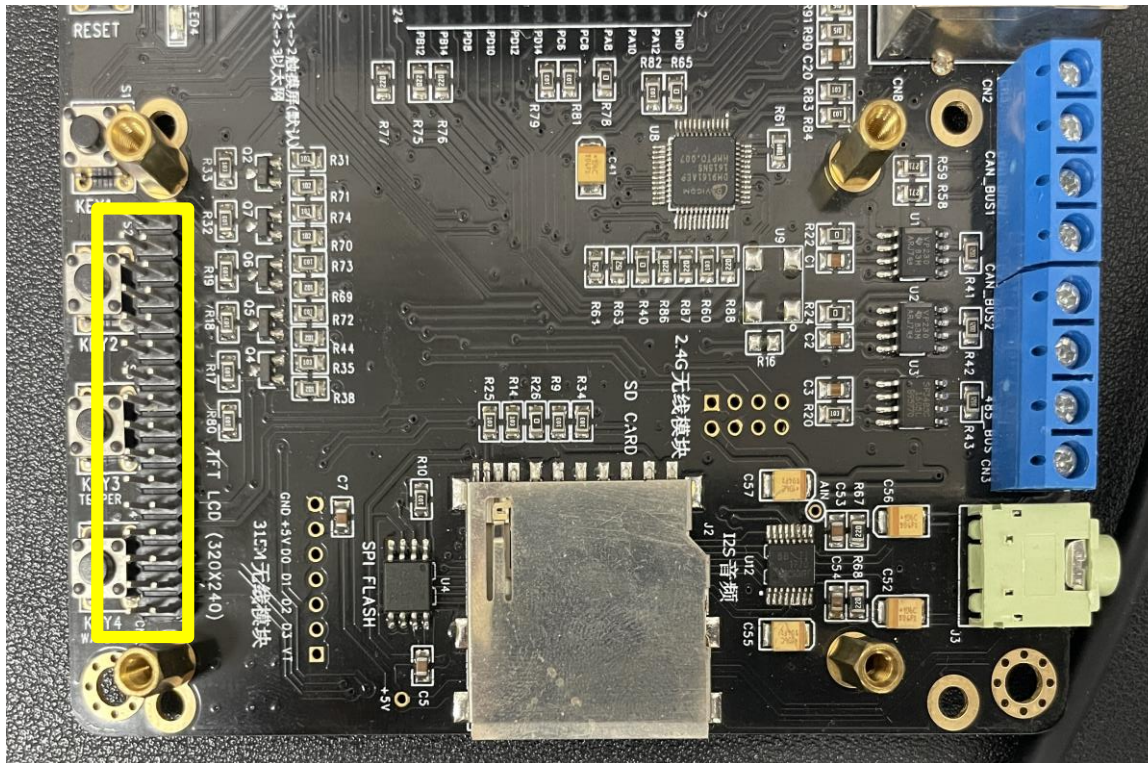
- TFT – LCD Pin 맵

- 핀 맵을 참고하여 각 신호를 인가할 핀 번호를 알 수 있음
- D/C 신호는 RS에 해당
- Data line은 DB(port E)에 해당



TFT-LCD 연결

TFT-LCD 연결



[TFT-LCD 모듈]

Contents

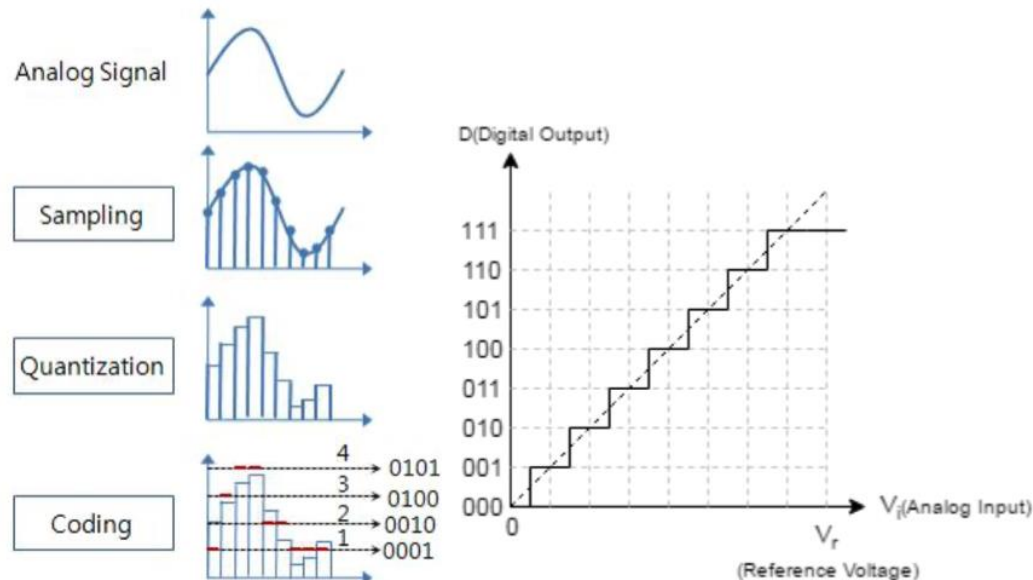
2. 실험 내용

2-2. TFT-LCD에 ADC를 통해 조도 센서 값 출력

ADC(Analog to Digital Converter)

ADC(Analog to Digital Converter)

- 아날로그 신호를 디지털 값으로 변환하는 converter
- 변환 과정
 - 표본화(Sampling): 아날로그 신호의 값을 일정한 간격으로 추출하여 샘플링
 - 양자화(Quantization): 샘플링한 값들을 일정한 레벨로 구분하여 단계별로 표현
 - 부호화(Coding): 양자화된 각 레벨 값을 이진수로 변환하여 디지털 형태로 나타냄



조도센서

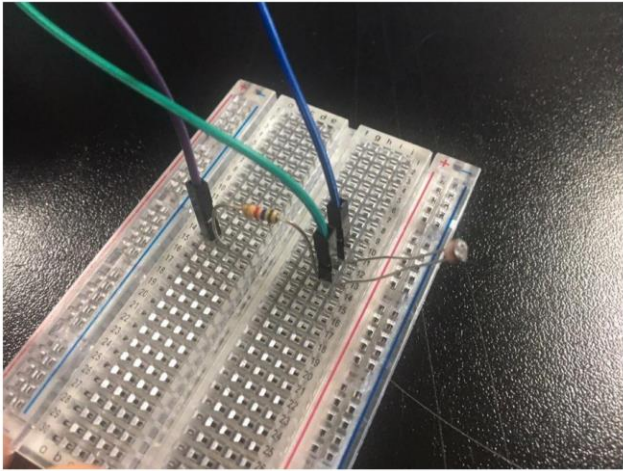
- 주변의 밝기를 측정하는 센서
 - 빛의 양이 많을 때는 저항이 낮아지고, 빛이 적을 때는 저항이 높아짐
 - 이 저항의 변화는 **아날로그 신호**로 나타나며, ADC를 통해 **디지털 신호로 변환**되어 MCU에서 처리됨
 - MCU는 ADC에서 변환된 디지털 값을 받아, 이를 LCD에 출력



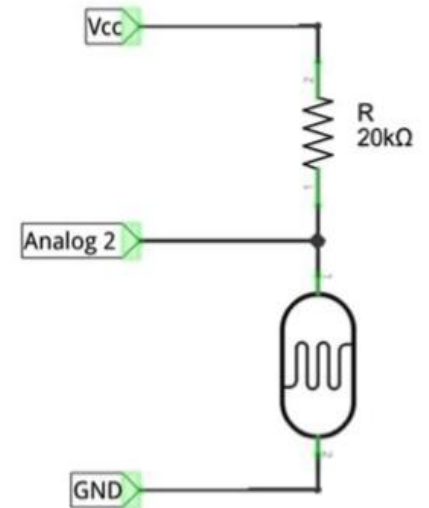
조도센서 회로 연결

조도센서 회로 연결

- 조도 센서와 저항을 직렬로 연결하여 전압 분배 회로를 구성
 - 조도 센서의 저항 변화에 따라 ADC 포트에 전달되는 전압이 변화($V=IR$)
 - ADC 포트에 전달된 아날로그 전압은 디지털 신호로 변환됨



— ADC port
— VCC
— GND



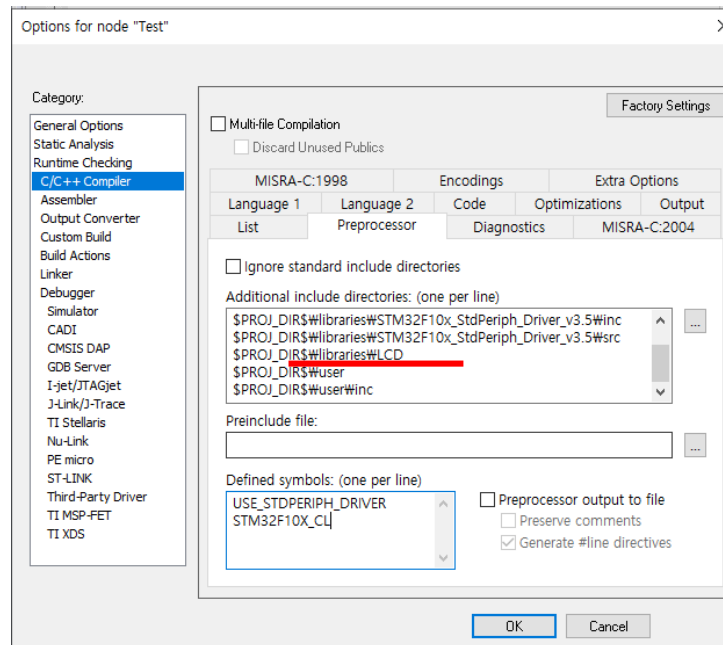
[회로도]

Contents

3. 실험 과정

LCD 라이브러리

- LCD 라이브러리 등록
 - font.h, lcd.c, lcd.h, touch.c, touch.h
 - Libraries 폴더 밑에 LCD 폴더 생성 후 위 5개 라이브러리 파일 추가
 - 프로젝트 옵션 – C/C++ Compiler – Preprocessor
 - 생성한 LCD 라이브러리 폴더 경로 추가

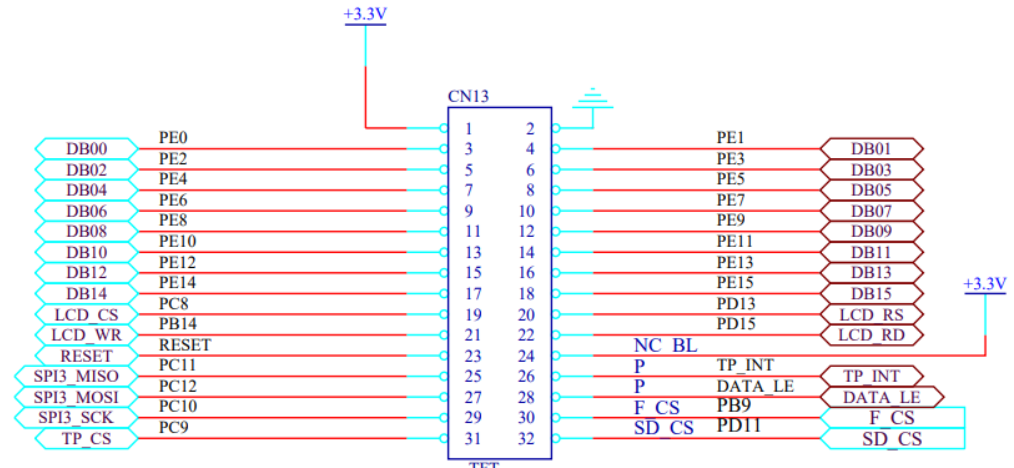


LCD 라이브러리 코드

- LCD 라이브러리의 Write 관련 코드
 - pin맵 참고하여 라이브러리(lcd.c) TODO 부분 완성

```
static void LCD_WR_REG(uint16_t LCD_Reg)
{
    // TODO implement
    GPIO_Write(GPIOE, LCD_Reg);
    // TODO implement
}

static void LCD_WR_DATA(uint16_t LCD_Data)
{
    // TODO implement
    GPIO_Write(GPIOE, LCD_Data);
    // TODO implement
}
```



Main 코드 작성 시 참고

```
#include "stm32f10x.h"
#include "core_cm3.h"
#include "misc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_usart.h"
#include "stm32f10x_adc.h"
#include "lcd.h"
#include "touch.h"

//색상 배열 정의
Int color[12] =
{WHITE,CYAN,BLUE,RED,MAGENTA,LGRAY,GREEN,YELLOW,BROWN,BR
RED,GRAY};

Int main() {

// LCD 관련 설정은 LCD_Init에 구현되어 있으므로 여기서 할 필요 없음
SystemInit();
RCC_Configure();
GPIO_Configure();
ADC_Configure();
NVIC_Configure();
// -----

LCD_Init();
Touch_Configuration();
Touch_Adjust();
LCD_Clear(WHITE);

while(1){
// TODO : LCD 값 출력 및 터치 좌표 읽기
}
}
```

// ADC 구조체 관련 코드

// ADC 값 읽기는 인터럽트 이용

uint16_t value -> 센서값 전역 변수 설정

ADC1사용

// ADC 설정 함수 안에서 사용될 함수 (레퍼런스 참조)

ADC_Init

ADC_RegularChannelConfig

ADC_ITConfig

ADC_Cmd

ADC_ResetCalibration

While(ADC_GetResetCalibrationStatus)->while 사용하여 체크

ADC_StartCalibration

While(ADC_GetCalibrationStatus) ->while 사용하여 체크

ADC_SoftwareStartConvCmd

// handler 콜백 함수 안에서

ADC_GetITStatus->체크

value = ADC_GetConversionValue(ADC1);

ADC_ClearITPendingBit

Contents

4. 실험 주의사항

실험 주의사항

- 실험 장비들을 연결 및 분리할 때 반드시 모든 전원을 끄고 연결해주세요.
- 장비사용시 충격이 가해지지 않도록 주의해주세요.
- 자리는 항상 깔끔하게 유지하고 반드시 정리 후 퇴실해주세요.
- 실험 **소스 코드와 프로젝트 폴더**는 **백업** 후 반드시 **삭제**해주세요.
- 장비 관리, 뒷정리가 제대로 되지 않을 경우 해당 조에게 감점이 주어집니다.

- **동작 중 케이블 절대 뽑지말것**
- **보드는 전원으로 USBPort나 어댑터(5V,1A)를 사용할것 (5V 5A 어댑터(비슷하게 생김)와 혼동하지 말 것, 사용시 보드가 타버림 -> 감점)**
- **디버깅 모드 중에 보드 전원을 끄거나 연결 케이블을 분리하지 말 것!!!**

- **-> 지켜지지 않을 시 해당 조 감점**

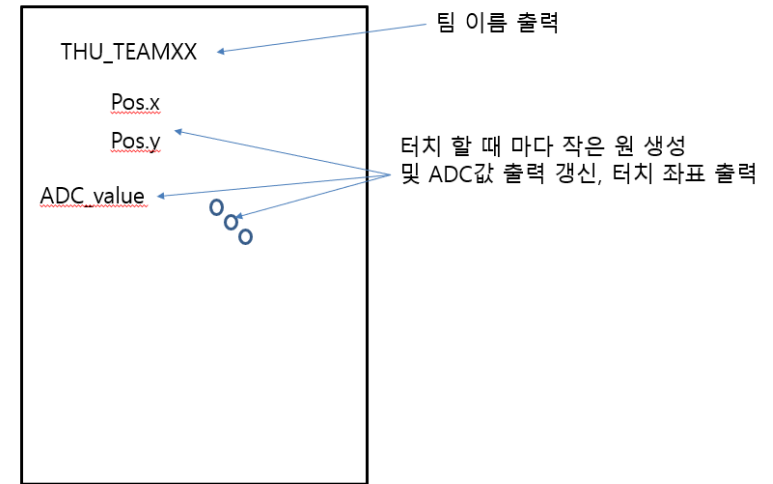
Contents

5. 실험 미션

실험미션

미션 ! 별도 미션지 참고

LCD 출력 업데이트는 터치 때만 되면 됩니다 (실시간으로 계속 바뀔 필요 없음)



실험 검사

오늘 검사 받을 수 있는 조는 오늘 받고 못 받는 조는 따로 미션 수행 후 다음 주 수업 시작할 때 검사
이번 주 실험 결과 보고서 및 소스 코드 및 실험 동작 영상

- A. 이론부터 실습까지 전반적인 내용을 포함하도록 작성
- B. 다음 실험시간 전까지 PLATO 제출 (보고서와 동작영상)

나가실 때, 만드신 코드 및 프로젝트 폴더는 모두 백업하시고 삭제해주세요.
다른 분반 파일은 만지지 마시고 조교에게 알려주세요.
자리 정리정돈 안 되어 있으면 **감점**합니다!!!