



Pusan
National
University



Nov 1, 2021

조교
최호진

tiger981228@pusan.ac.kr

임베디드 시스템 설계 및 실험

목요일 분반

10주차
LCD 및 ADC

11주차 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**

Contents

10주차 실험 내용

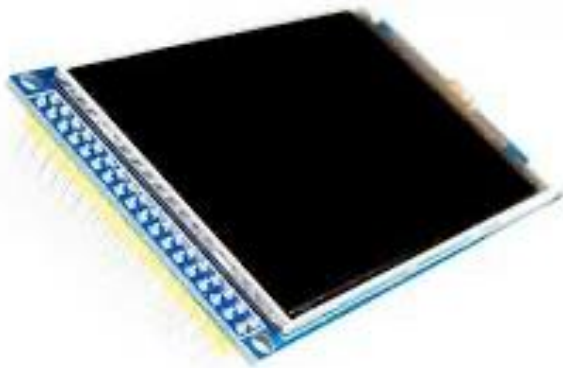
LCD 및 ADC

- TFT-LCD의 원리와 동작 방법에 대한 이해
- TFT-LCD 라이브러리 작성과 이해
- TFT-LCD Touch 동작 제어

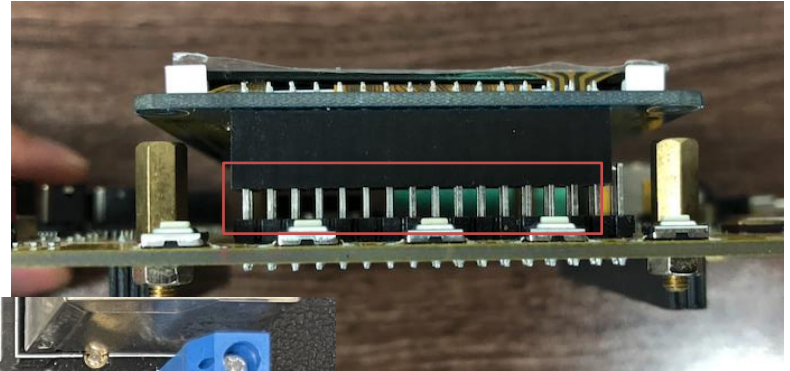
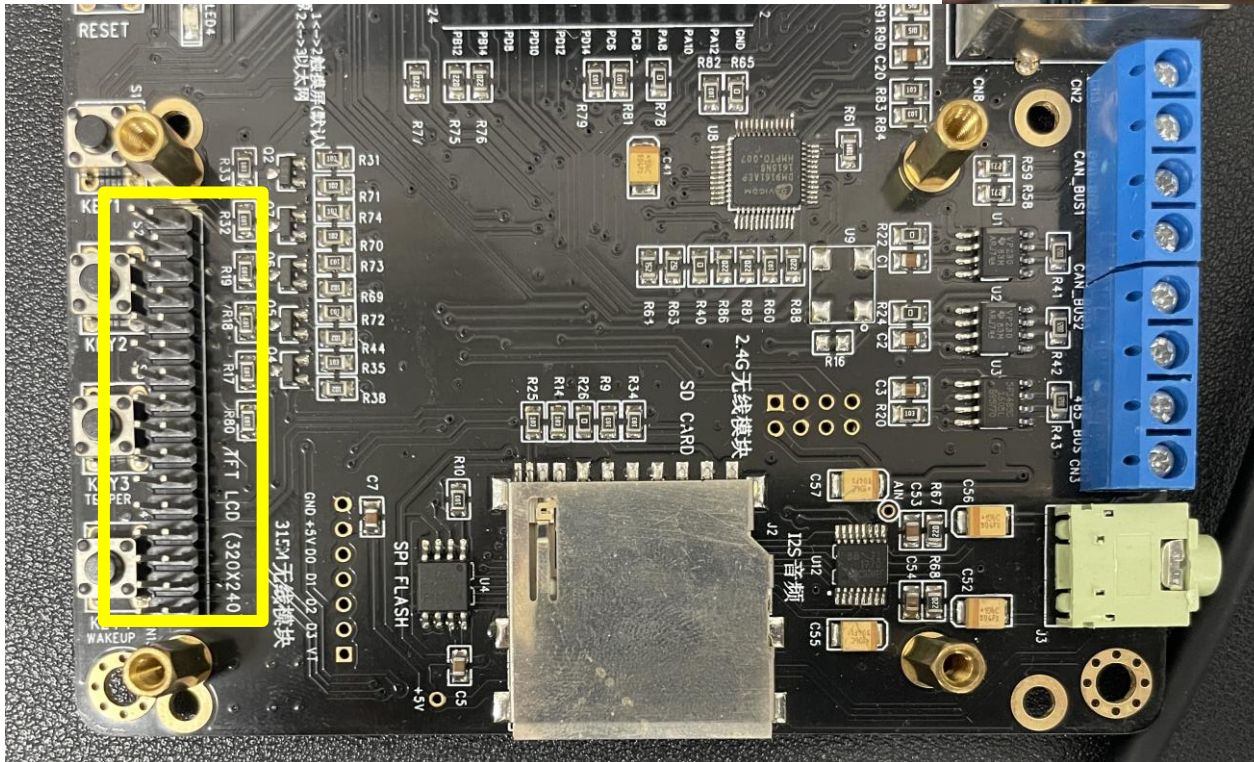
- ADC 개념 이해
- 조도 센서 사용 방법 학습

TFT-LCD

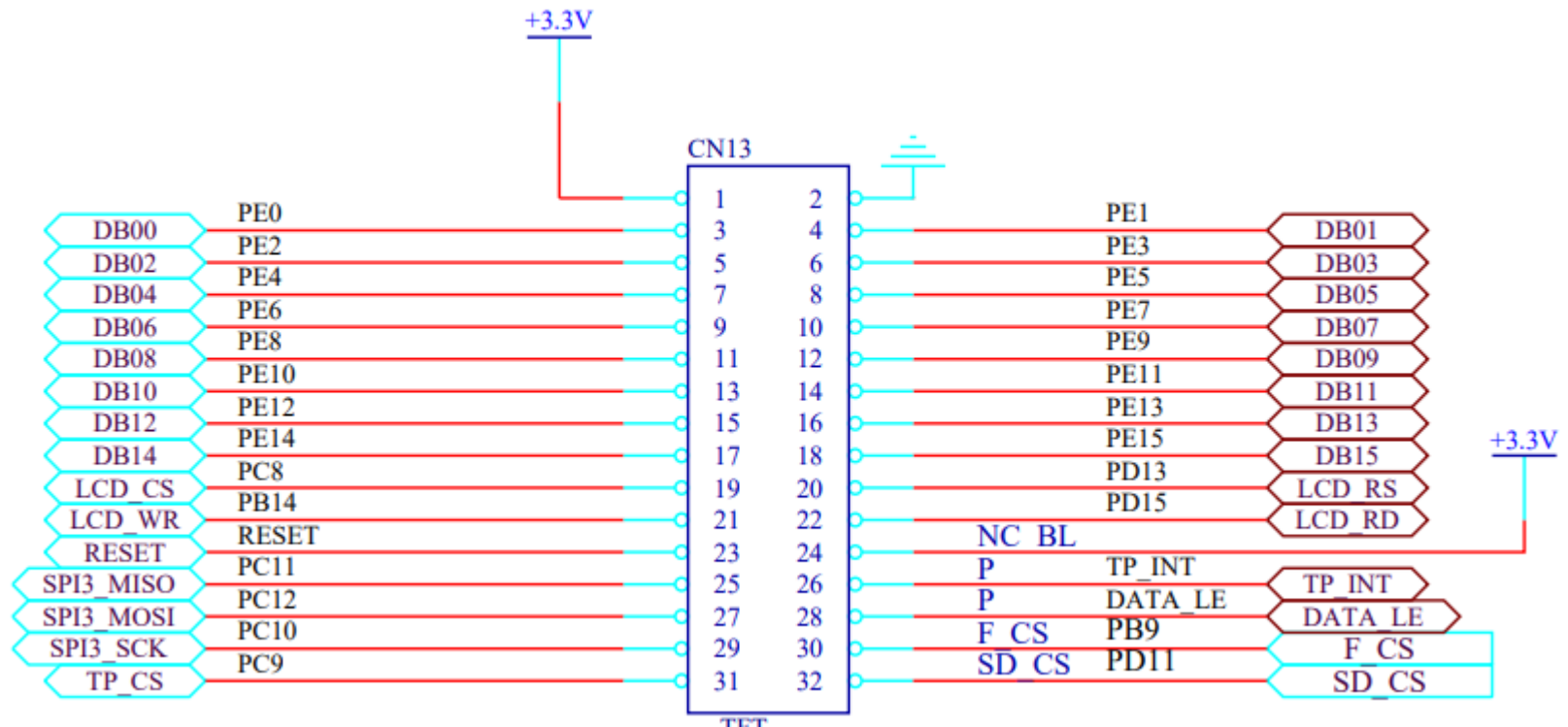
- 초 박막 액정표시장치
- 액체와 고체의 중간 특성을 가진 액정의 상태 변화와 편광판의 편광 성질을 이용하여 통과하는 빛의 양을 조절함으로써 정보를 표시
- RGB 픽셀이 유리판에 코팅 되어 컬러 영상을 구현하는 Color Filter
- 액정을 제어하기 위해 조박형 유리 기판 위에 반도체 막을 형성한 회로인 TFT 기판
- Filter와 기판 사이에 주입된 액정과 광원인 Black light unit으로 구성



- 오른쪽 한 칸을 비우고 보드와 LCD를 연결

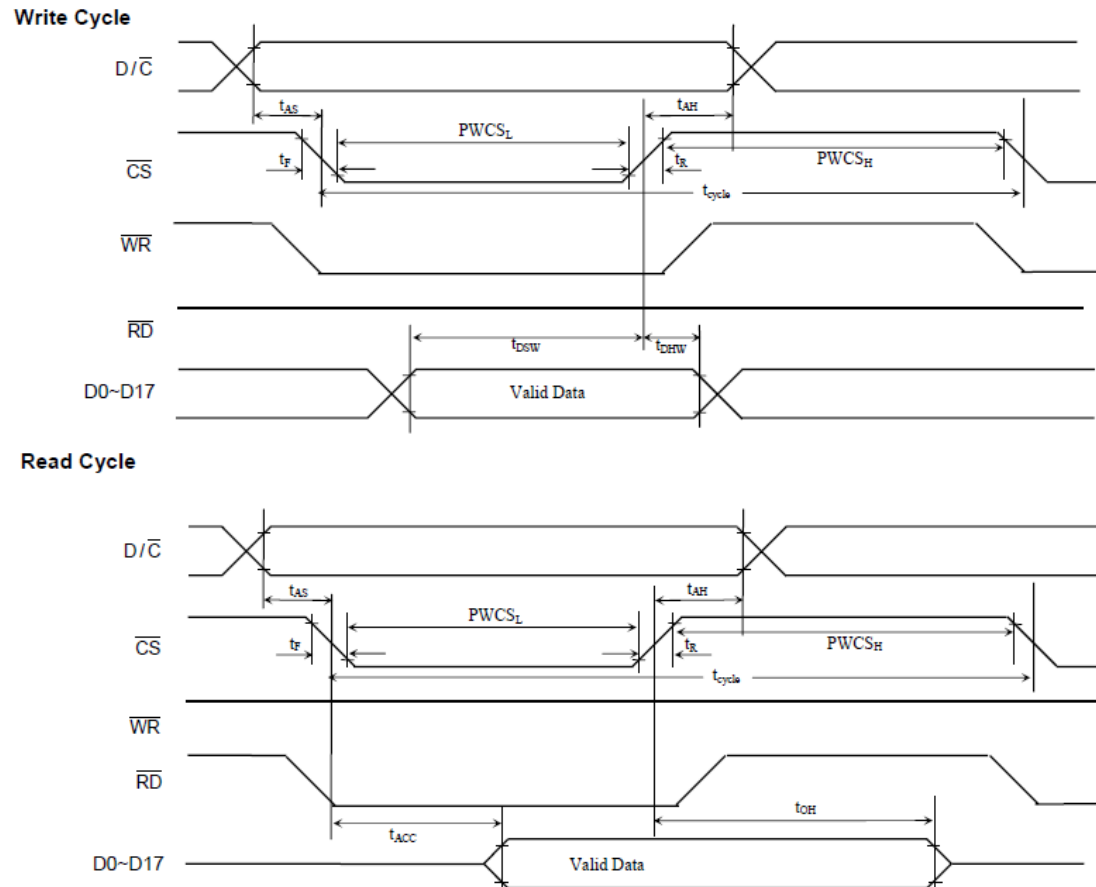


TFT-LCD Pin맵

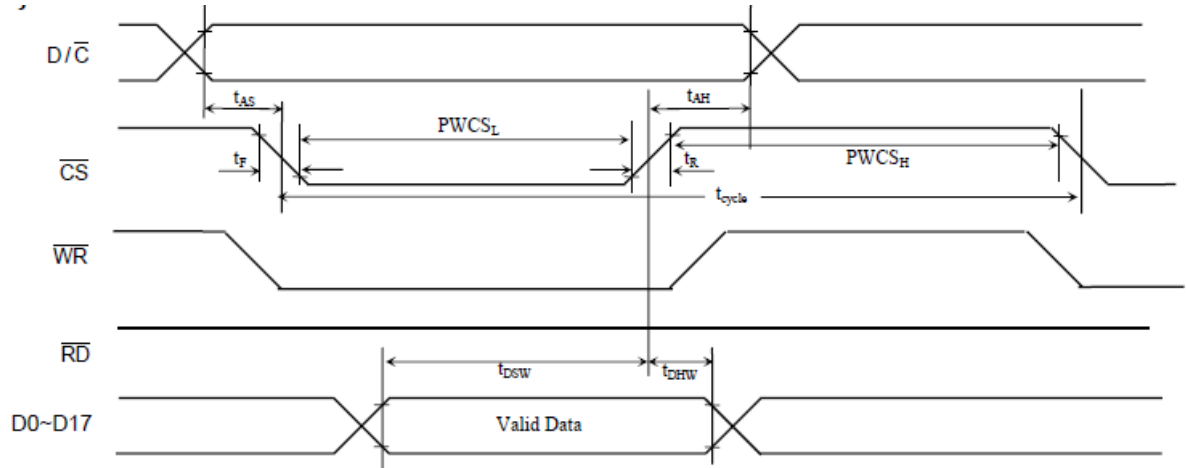


Timing Diagram

- 각 신호들이 시간 별로 처리되는 과정을 그림으로 나타냄



Timing Diagram

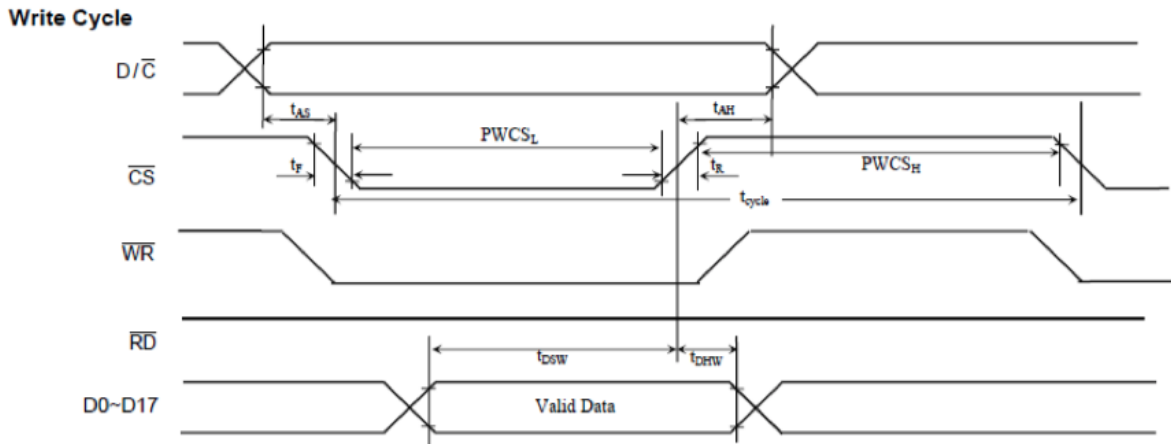


- Low에서 High로 올라가는 구간을 Rising Edge
- High에서 Low로 떨어지는 구간을 Falling Edge



- D/\overline{CS} 처럼 교차 형태를 취하고 있으면, High / Low 둘 중 하나의 값을 가질 수 있다는 것을 의미

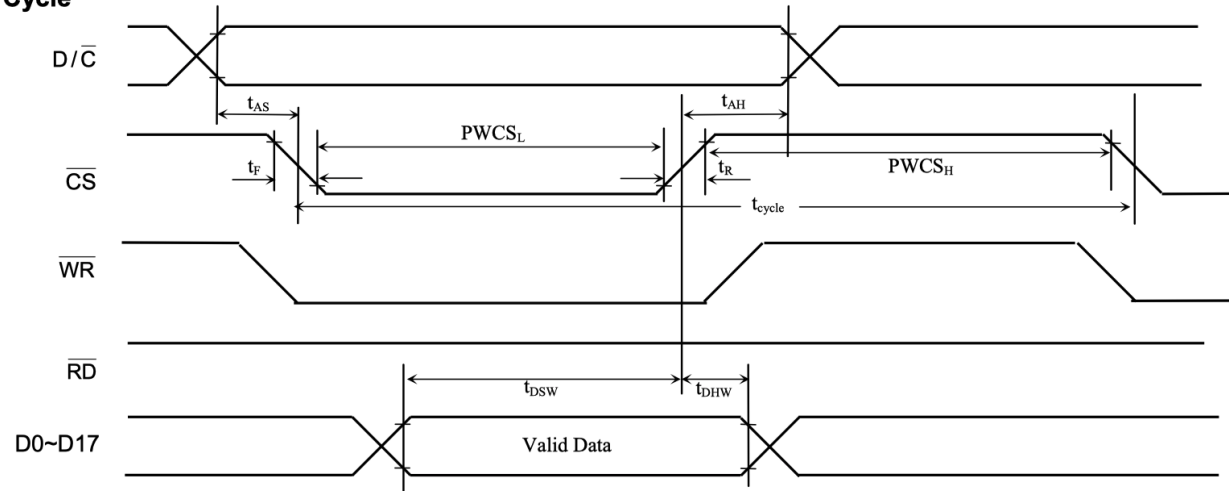
Timing Diagram - Write / Read Cycle



- \overline{CS} : Chip Select (Chip Enable)
 - High 에서 Low로 Falling Edge 일 때 LCD Chip 을 사용
- D/\overline{C} : Data / Command (핀맵에서 RS)
 - LCD는 Data 와 명령어 레지스터를 함께 사용
 - High 로 두고 Data를 전송, Low 로 두고 Command를 전송
- $\overline{WR}, \overline{RD}$: Write / Read
 - High 에서 Low로 Falling Edge 일 때, Data를 display에 Write / Read 한다

Timing Diagram - Write / Read Cycle

Write Cycle

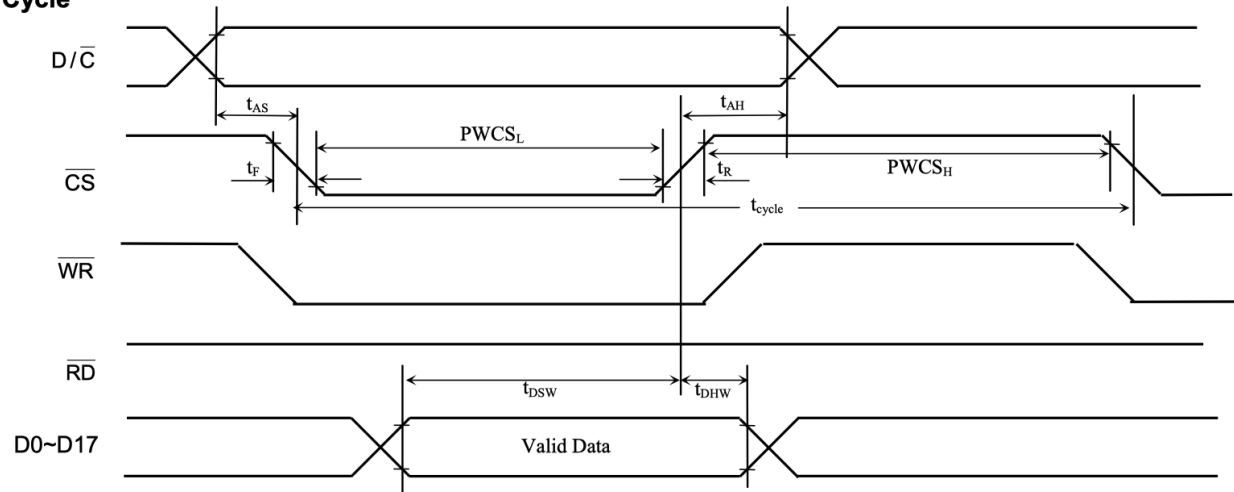


- 해당 Symbol의 Min Time 내에 Falling / Rising 을 해야한다.

Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time (write cycle)	100	-	-	ns
t_{cycle}	Clock Cycle Time (read cycle)	1000	-	-	ns
t_{AS}	Address Setup Time	0	-	-	ns
t_{AH}	Address Hold Time	0	-	-	ns
t_{DSW}	Data Setup Time	5	-	-	ns
t_{DHW}	Data Hold Time	5	-	-	ns
t_{ACC}	Data Access Time	250	-	-	ns
t_{OH}	Output Hold time	100	-	-	ns
$PWCS_L$	Pulse Width /CS low (write cycle)	50	-	-	ns
$PWCS_H$	Pulse Width /CS high (write cycle)	50	-	-	ns
$PWCS_L$	Pulse Width /CS low (read cycle)	500	-	-	ns
$PWCS_H$	Pulse Width /CS high (read cycle)	500	-	-	ns
t_R	Rise time	-	-	4	ns
t_F	Fall time	-	-	4	ns

Timing Diagram - Write Cycle

Write Cycle



- **COMMAND**

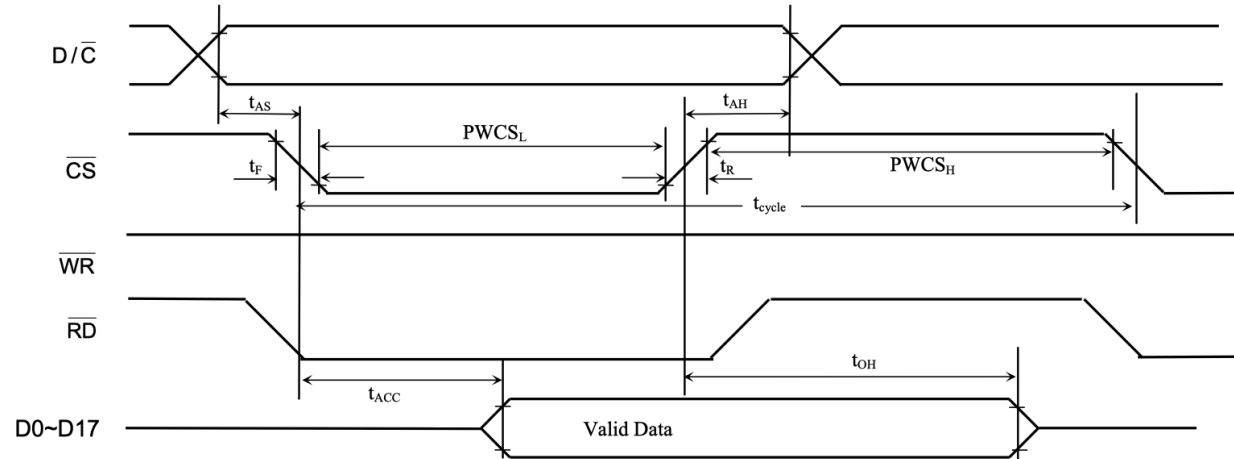
- D/\overline{C} 를 Low, \overline{CS} 를 Low, \overline{WR} 를 Low로 두고 Command를 전송
- \overline{CS} 를 High, \overline{WR} 를 High로 다시 돌려놓기

- **DATA**

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

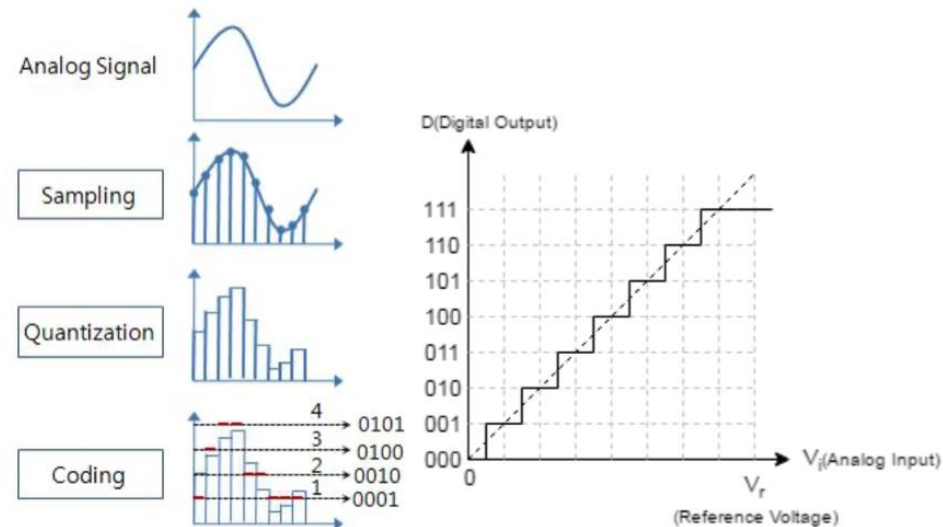
Timing Diagram - Read Cycle

Read Cycle



- D/\overline{C} 가 high, \overline{CS} 가 Low, \overline{RD} 신호가 Low 일 때 $D0-D17$ 의 Display를 읽는다

ADC(Analog to Digital Converter)



- 아날로그 신호를 디지털로 변환하는 것
- 아날로그 신호가 들어오면 이를 표본화, 양자화를 거쳐 부호화 한다.
- 표본화: 일정한 간격으로 아날로그 신호의 값을 추출
- 양자화: 추출한 표본 샘플 신호의 레벨을 단계를 나누어 나타내는 과정
- 부호화: 양자화로 나눈 레벨에 속한 값을 이진수로 변환

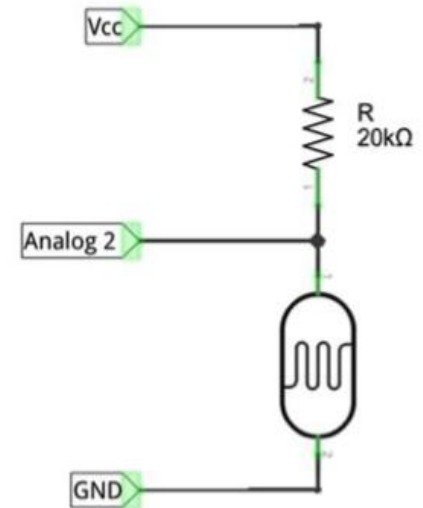
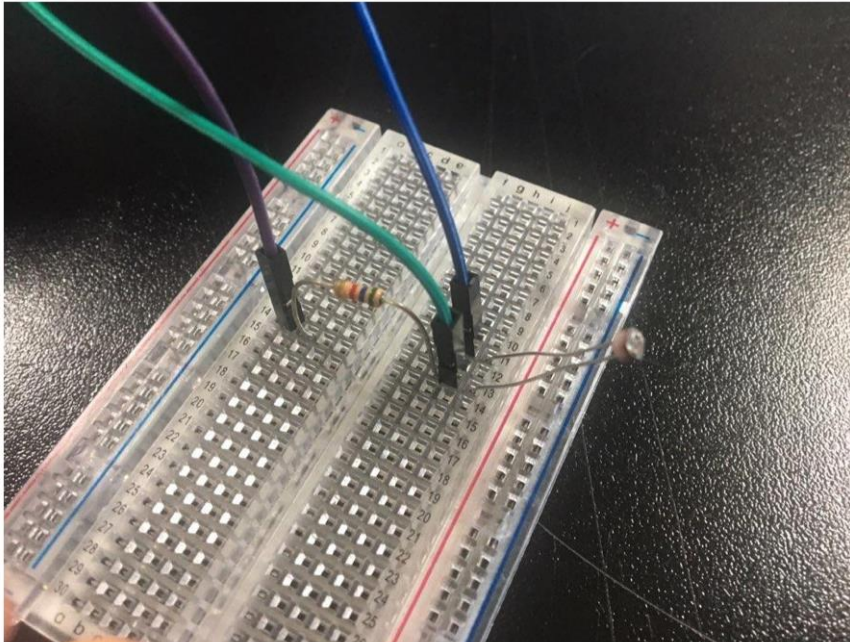
조도센서

- 주변의 밝기를 측정하는 센서
- 빛의 양이 많아질수록 전도율이 높아져 저항이 낮아짐



조도센서 회로구성

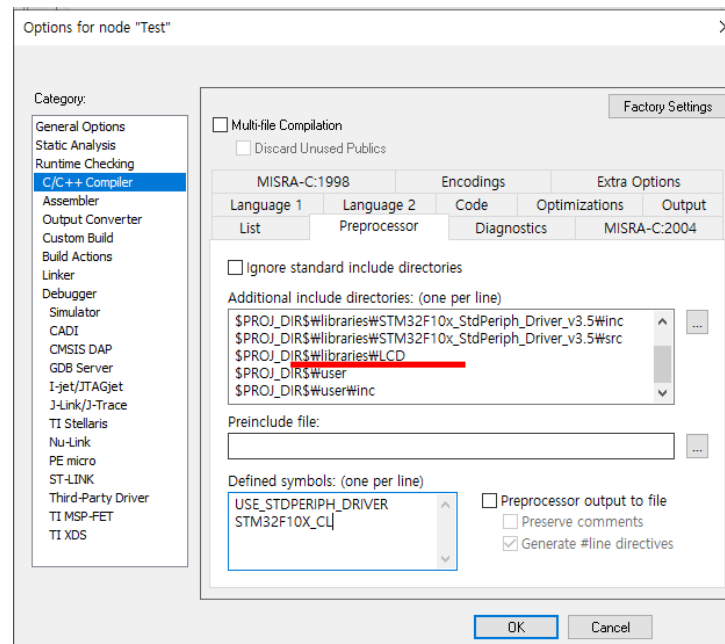
- 조도 센서를 저항을 사용하여 연결
- 전압 분배가 일어나 조도 센서에 걸리는 전압을
신호로 전달받아 측정



- ADC port
- VCC
- GND

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 완성

- LCD 라이브러리의 Write 관련 코드
 - 데이터시트의 Timing Diagram 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
}
```

// 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

// main 참고 코드

```
#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"
```

```
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

```
}
}
```

- 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**

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!!!

