

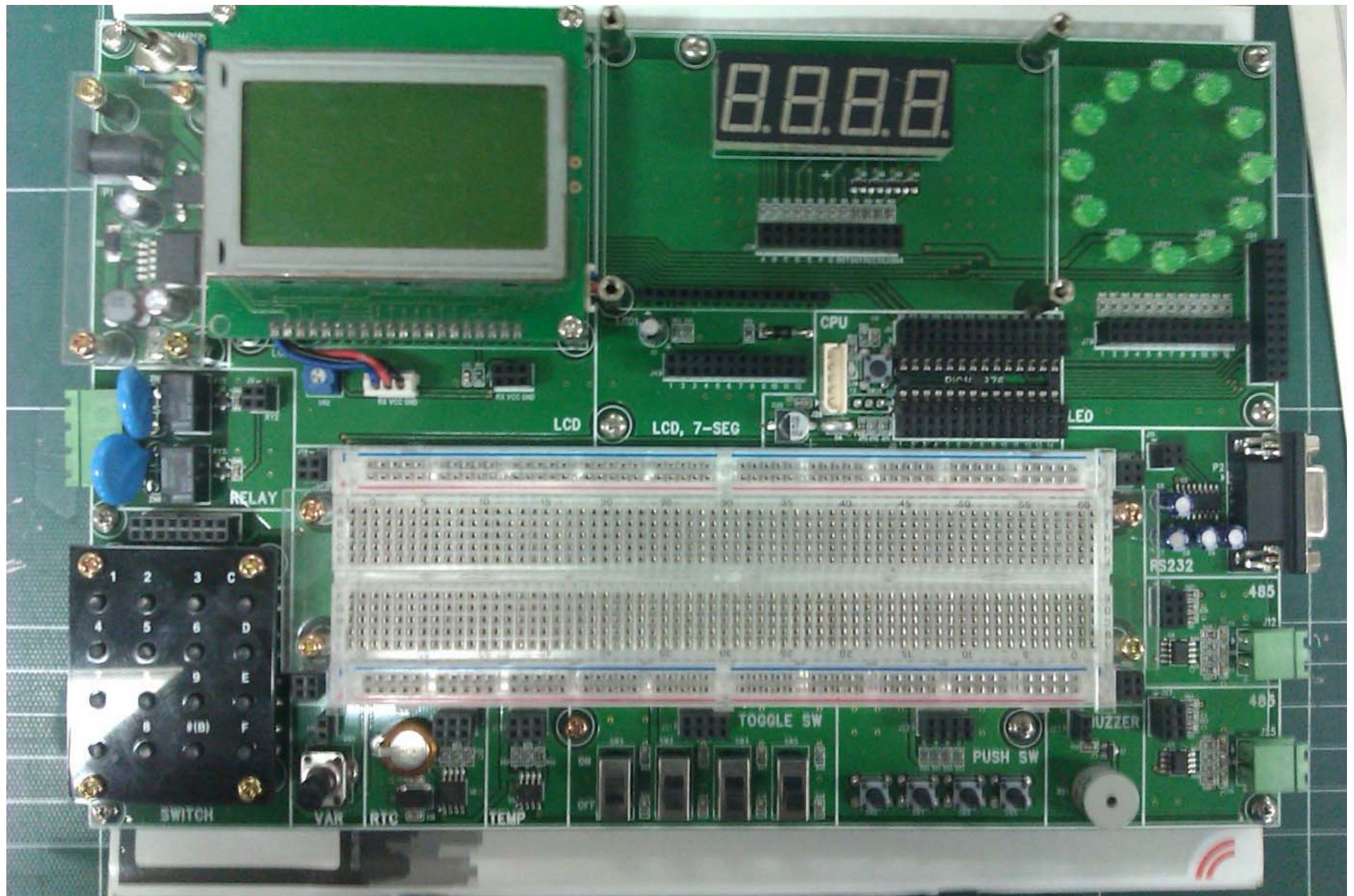
## 실험 데모 보드 소개 및 어셈블리 프로그래밍

# 개발장비

---

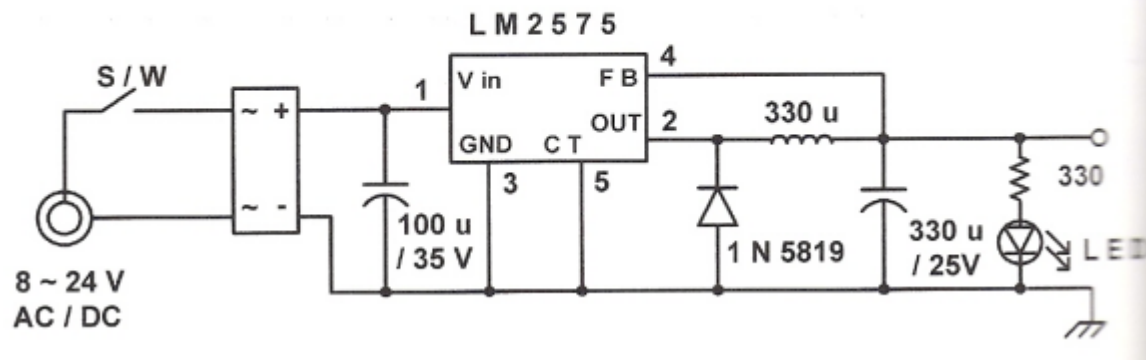
- ▶ 프로그래머 (Programmer) : 롬 라이터 (ROM-Writer)
  - ▶ 번역된 기계어 프로그램을 MCU의 기억장치에 기억시키는 작업을 수행하는 장치
  - ▶ OTP (One Time Program) : 저비용
  - ▶ 개발제품 양산시 주로 사용
- ▶ ICE (In Circuit Emulator)
  - ▶ 초기 제품 개발 시 사용
    - ▶ 개발하고자 하는 MCU와 모든 기능 동일
    - ▶ 프로그램 수행 및 컨트롤러 내부 상태 관찰 가능 (break point 설정)
- ▶ MPLAB-IDE
- ▶ Target demo board



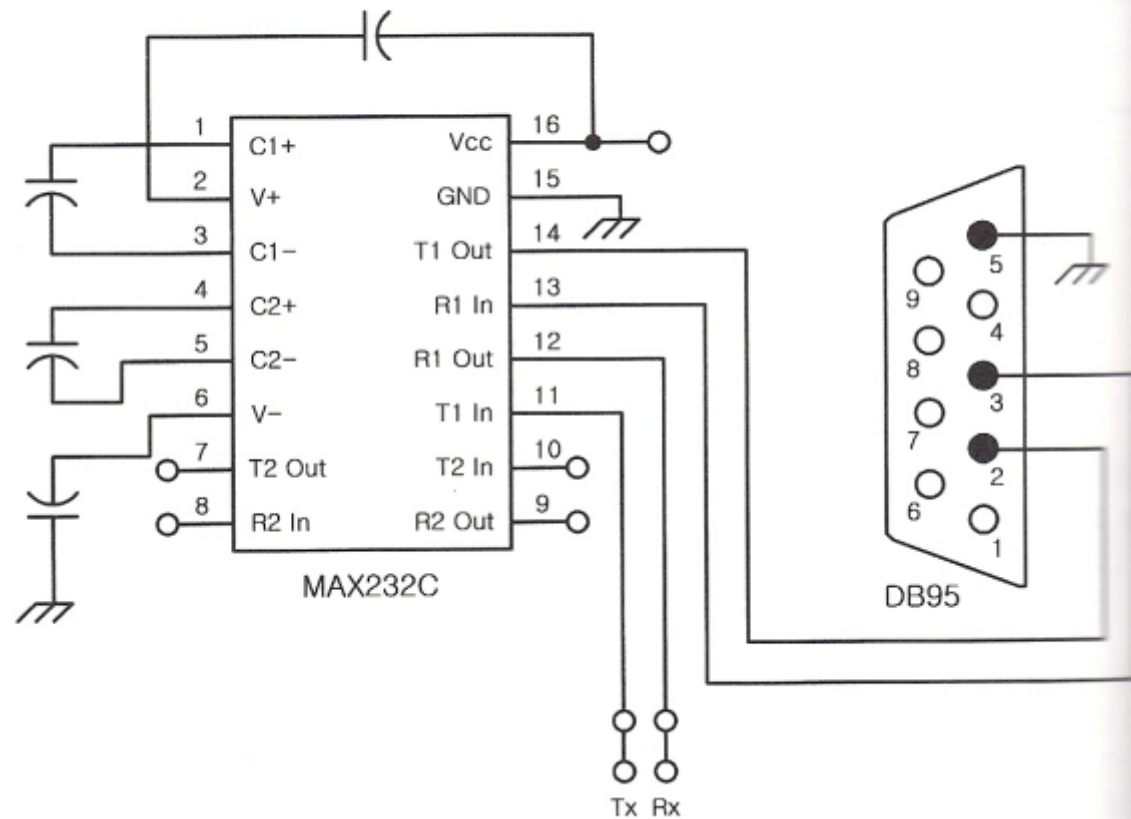


마이크로프로세서설계실험

## 1. POWER

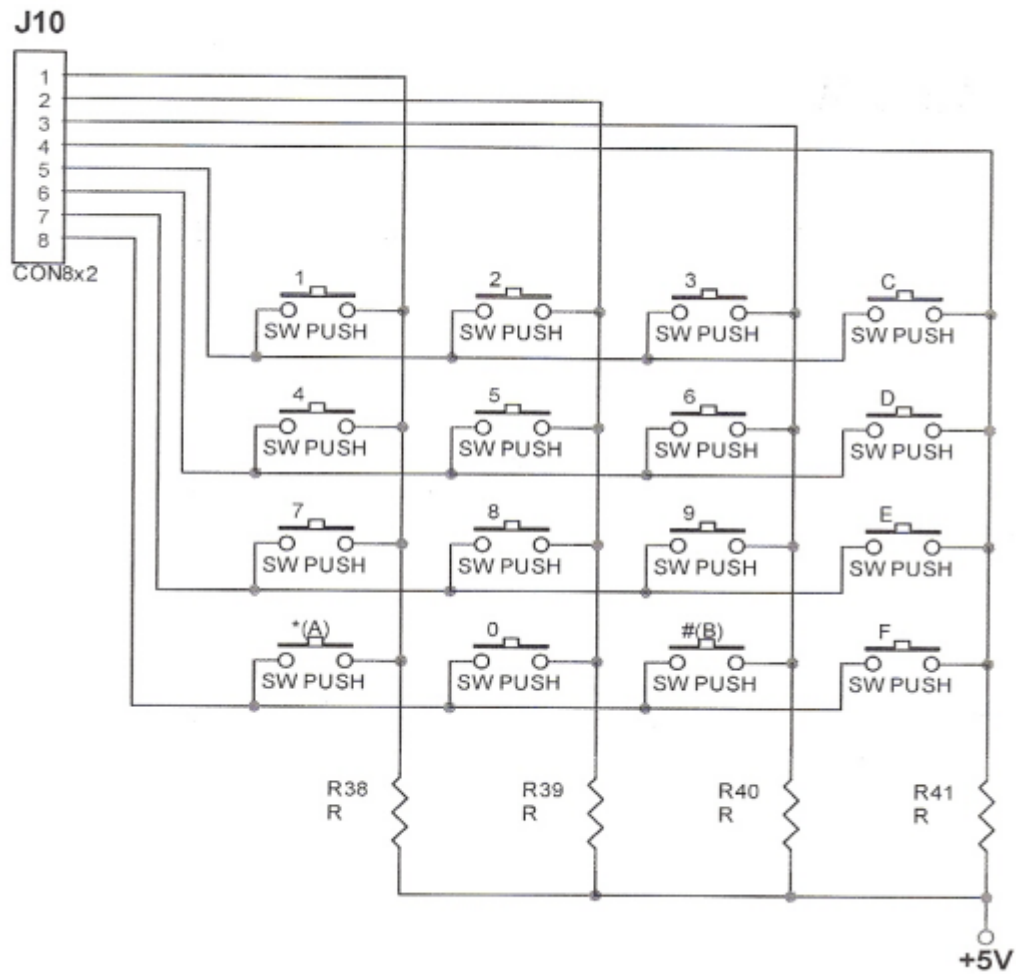


## 2. RS232C

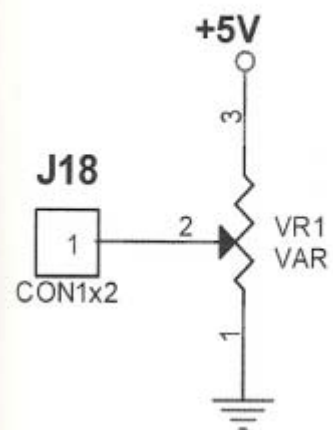




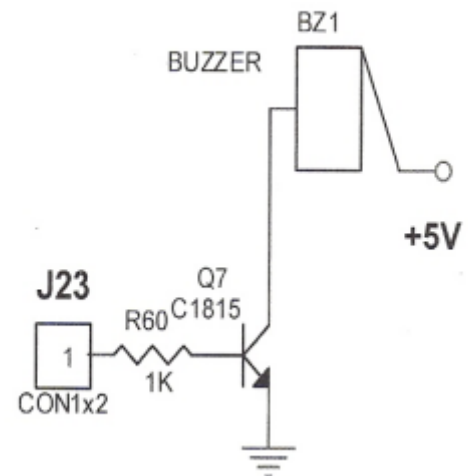
### 3. 4x4 KEYPAD



#### 4. A/D

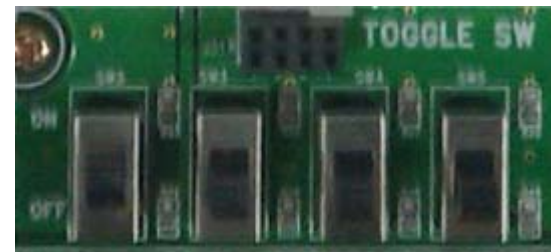
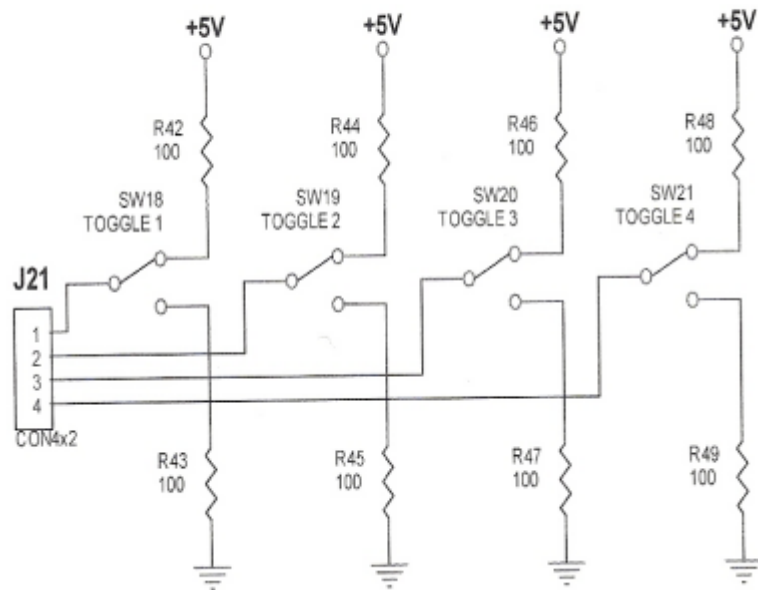


## 5. BUZZER

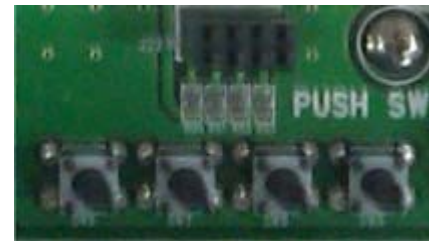
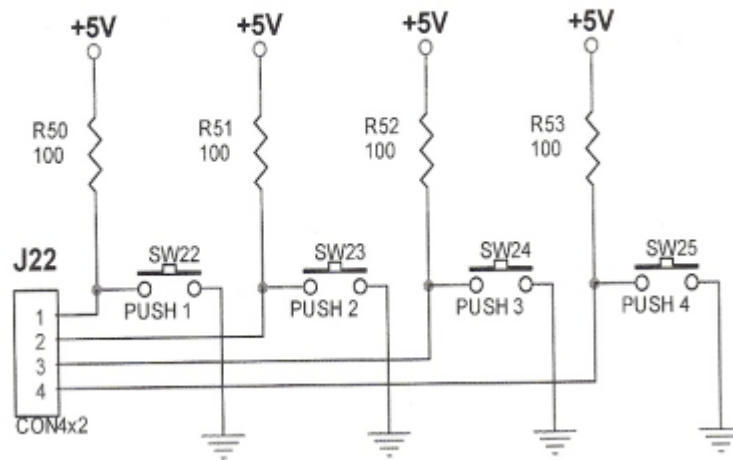




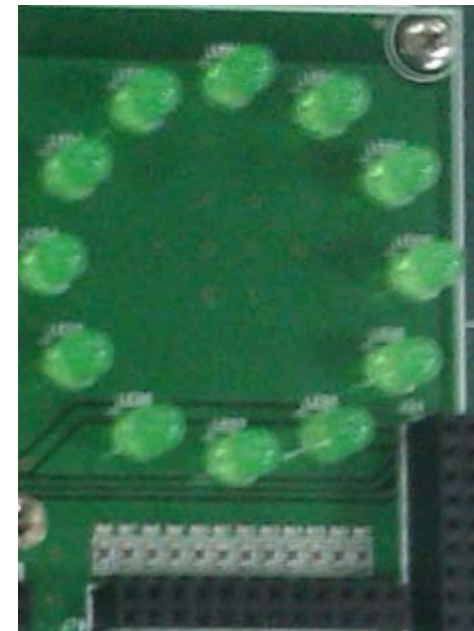
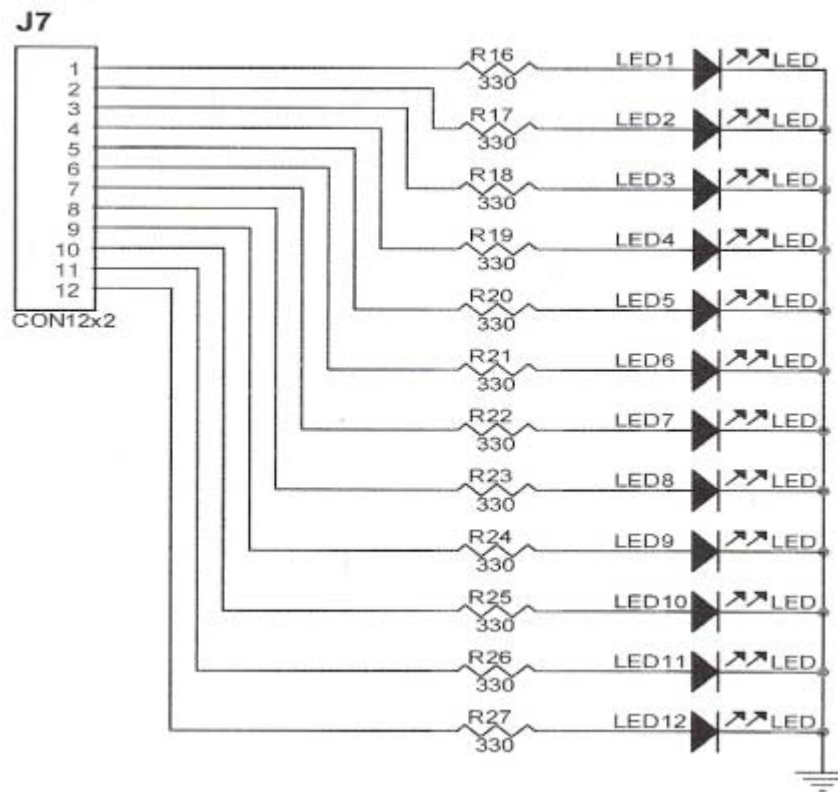
## 6. TOGGLE S/W



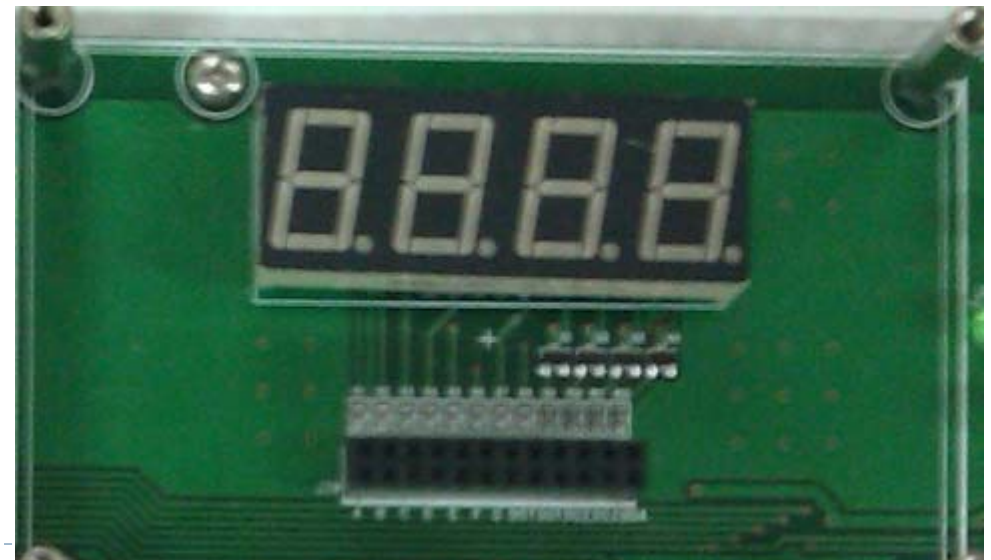
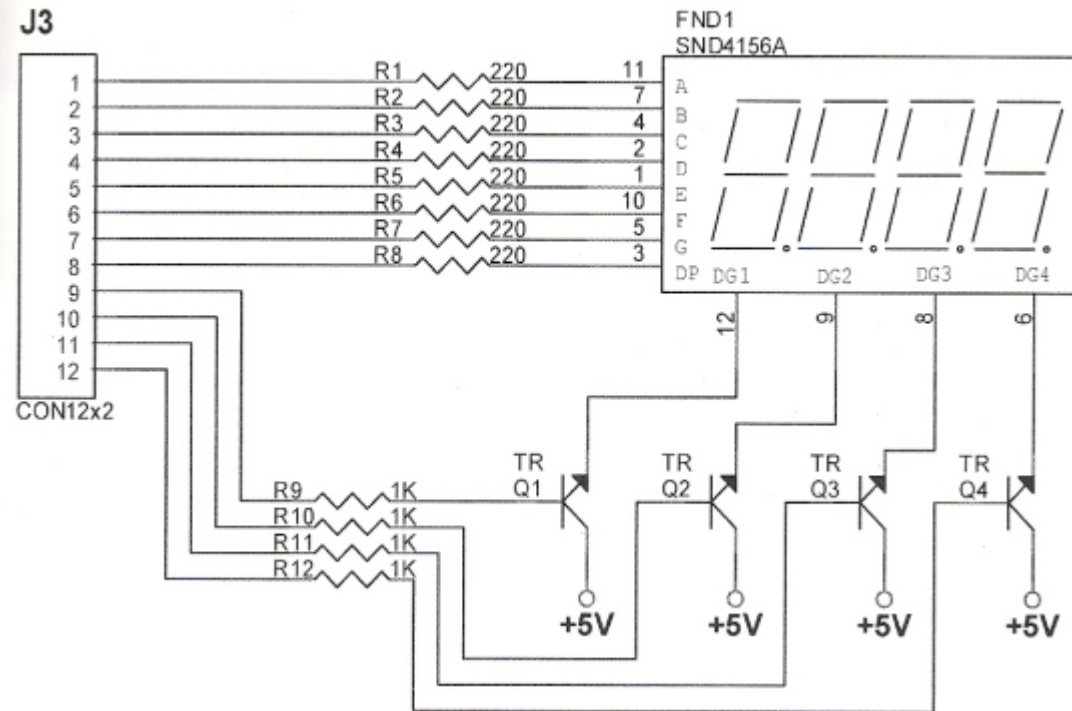
## 7. PUSH S/W



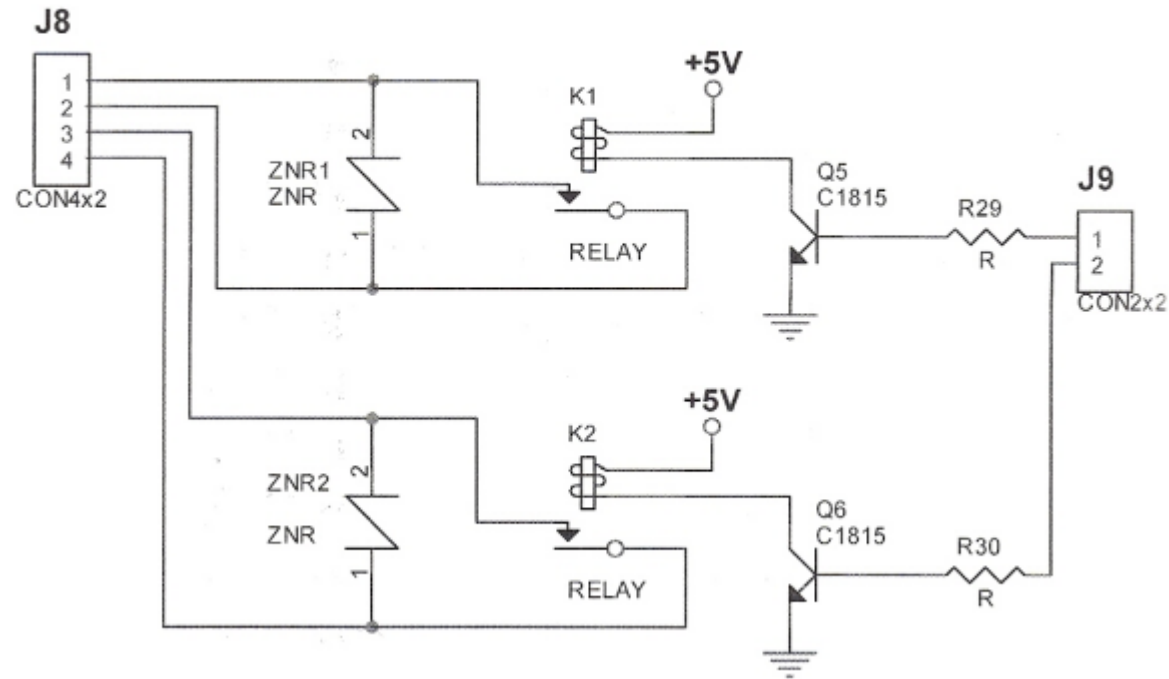
## 8. LED



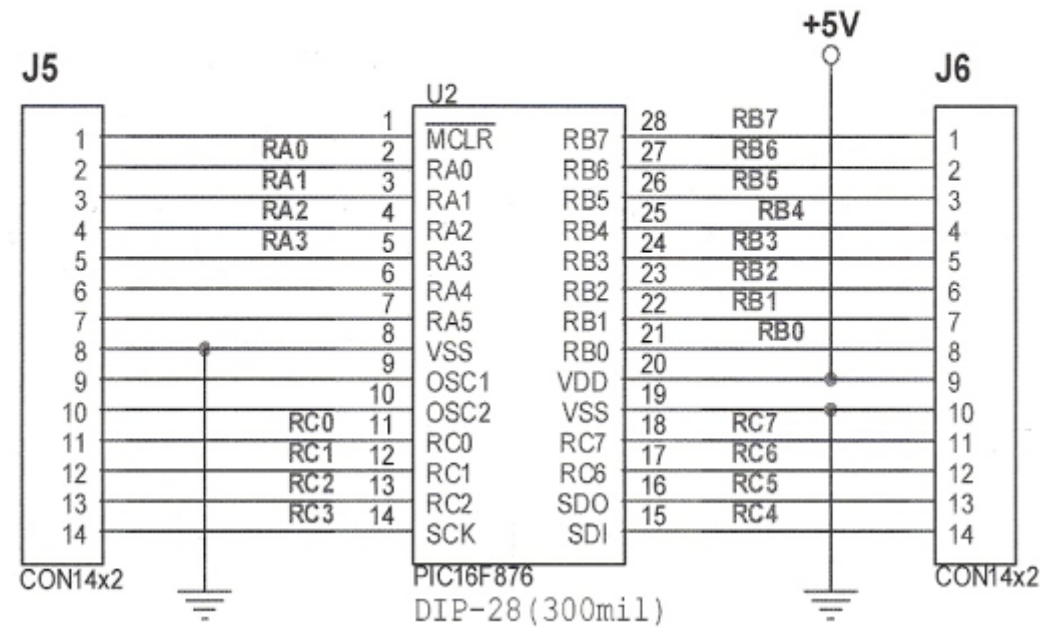
## 9. FND DISPLAY



## 10. RELAY

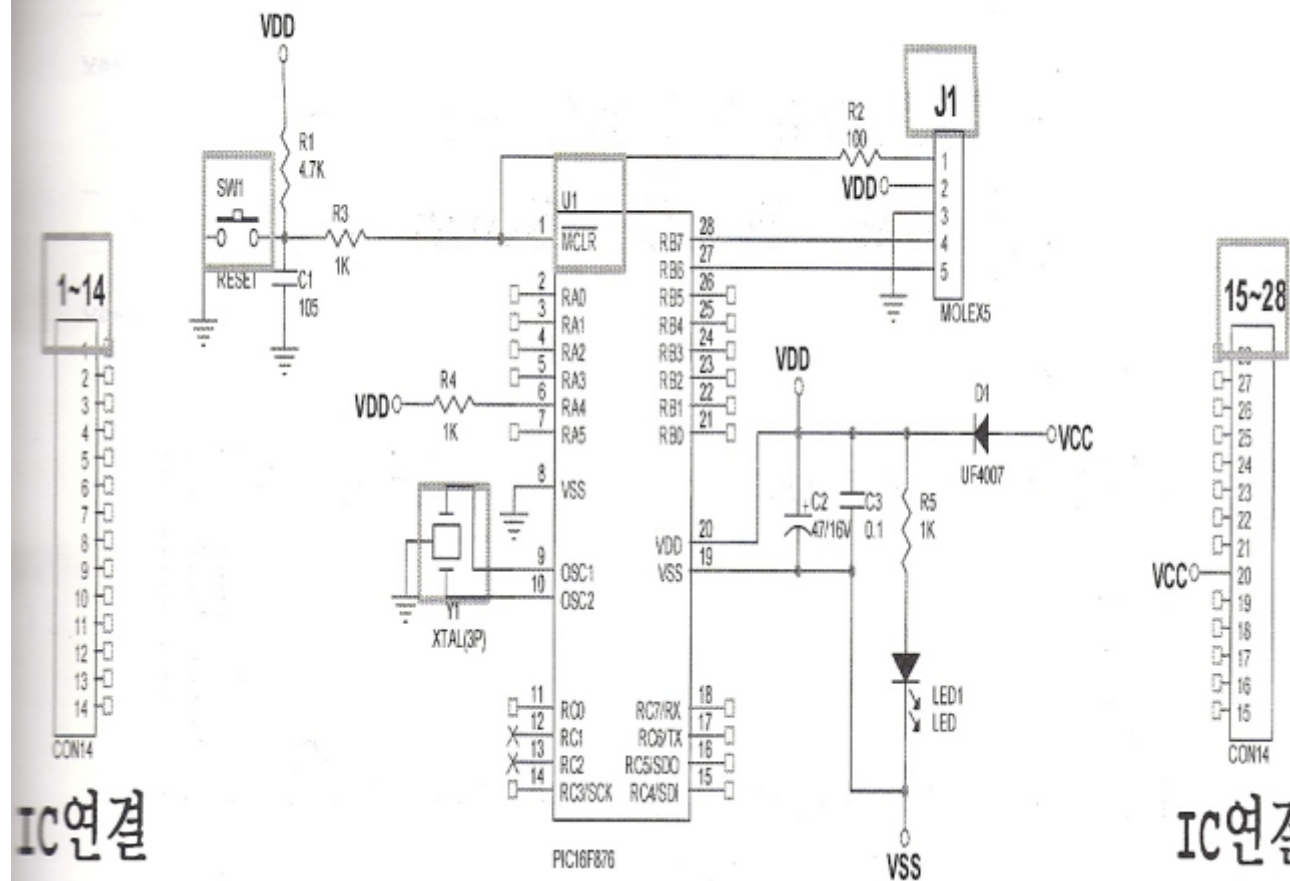


## 11. CPU 회로 (외부연결도)

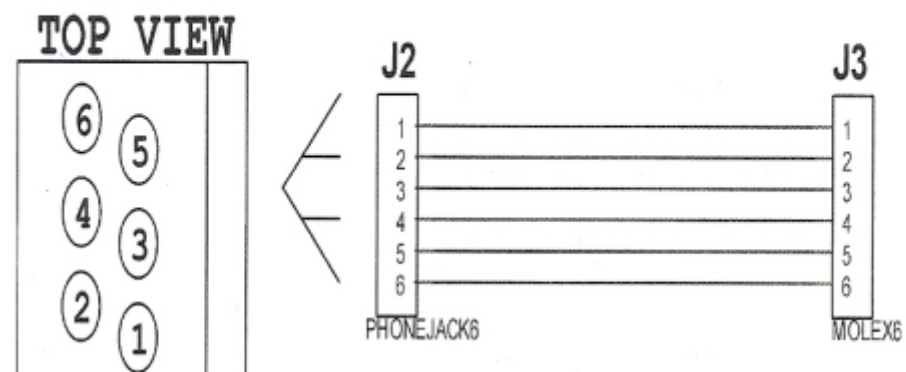




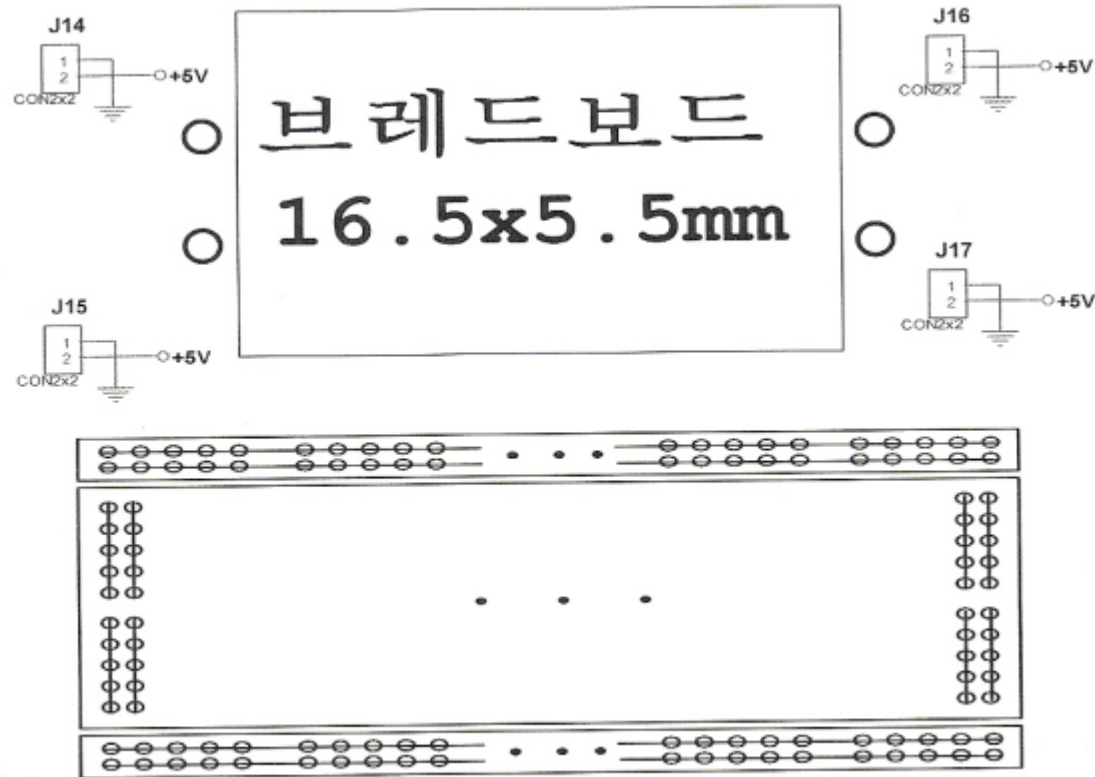
(CPU 보드 내부 회로도)



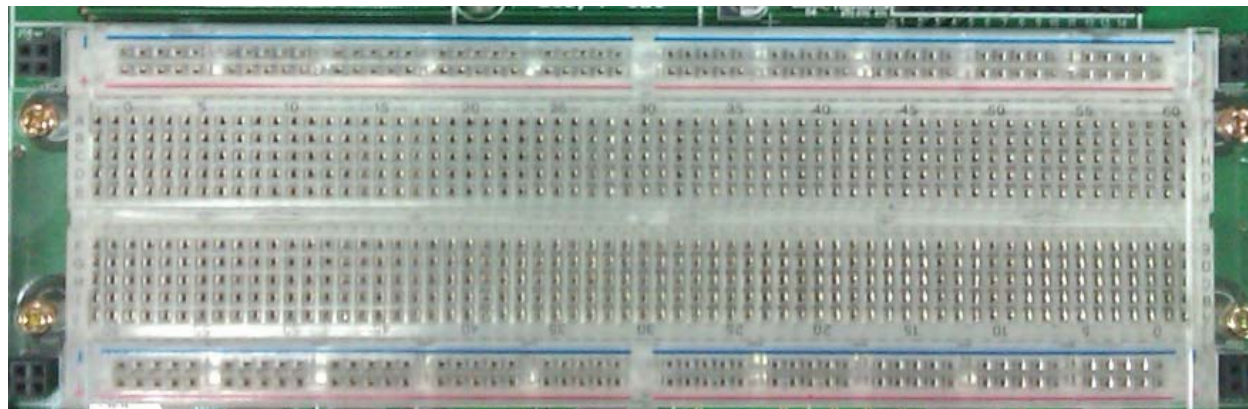
(CPU 보드와 개발장비 연결도)



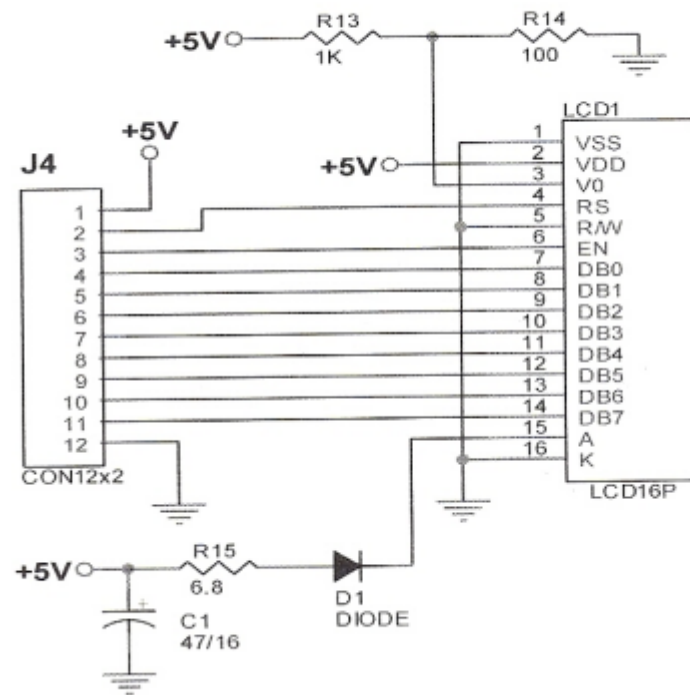
## 12. BREAD BOARD



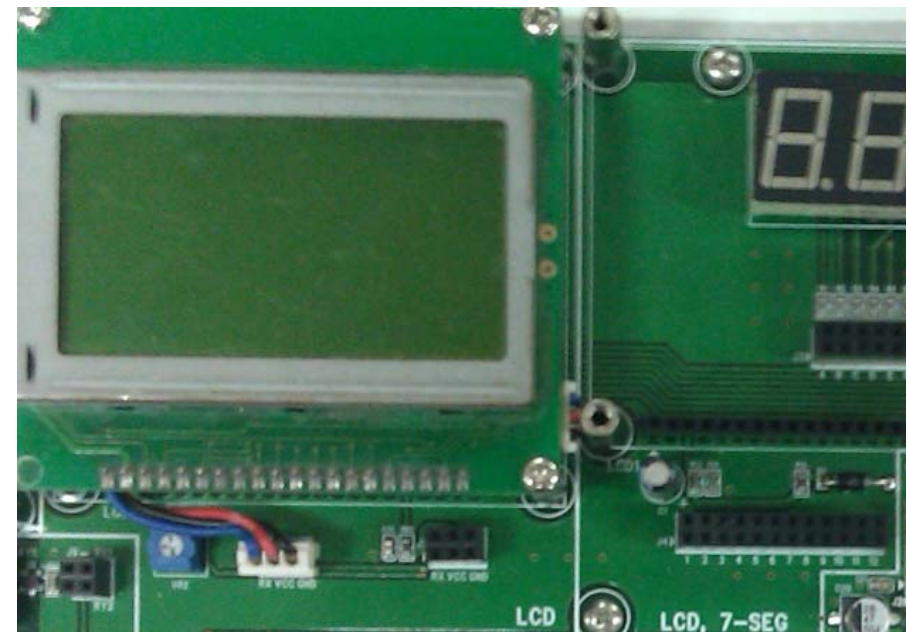
\* 실선은 내부연결상태를 나타냄



### 13. TEXT LCD

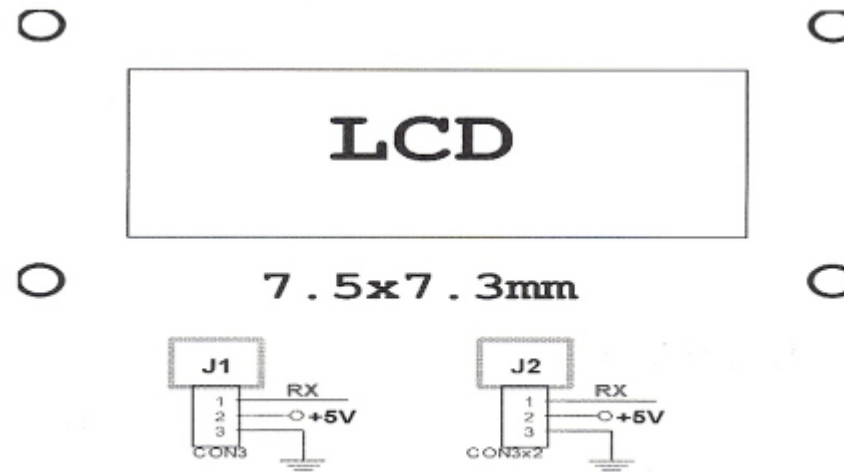


20\*4 TYPE

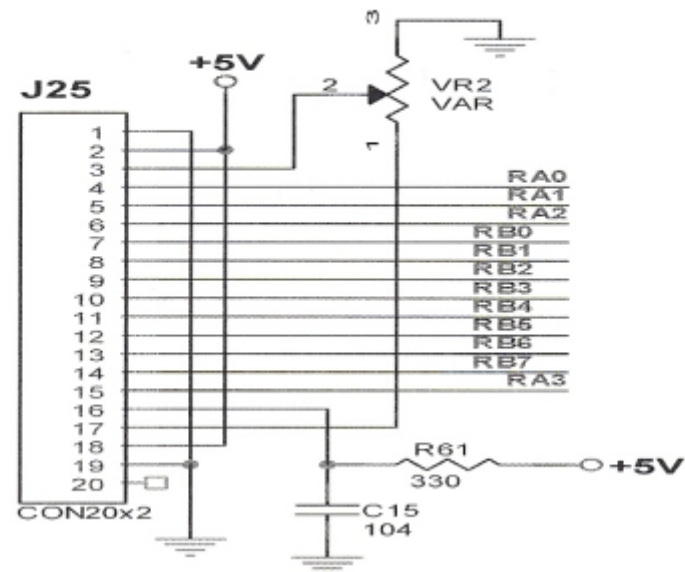


#### 14. 시리얼 한글 LCD 모듈(HLCD114)

- 최대 11 by 4의 한글 표시 가능
- 최대 20 by 4의 영문 표시 가능
- LED 백라이트
- 한글 4층, 영문 2층
- 그래픽 LCD 도트수 128 byte
- 화면크기  $70.7 \times 30.8 \text{ mm}$
- 통신방식은 5V 레벨, 4800, N, 8, 1 사용



## 15. GRAPHIC LCD

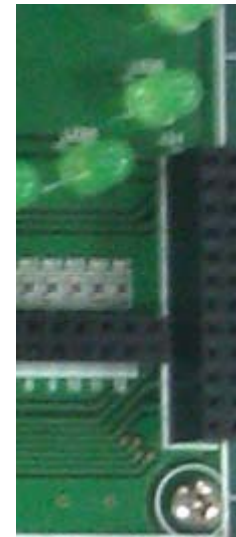
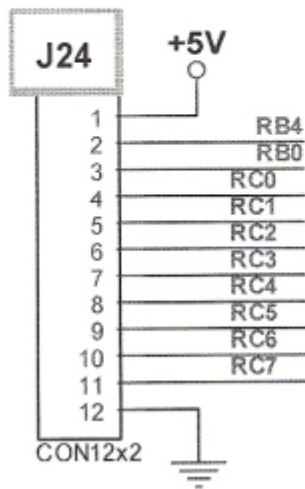




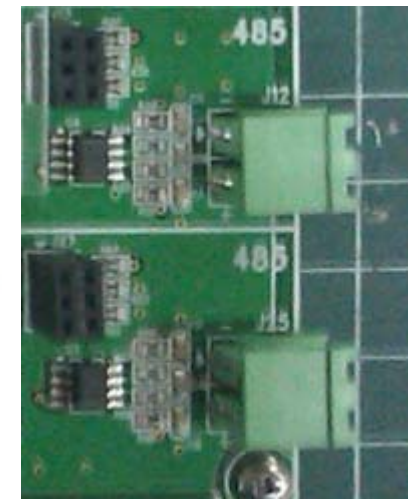
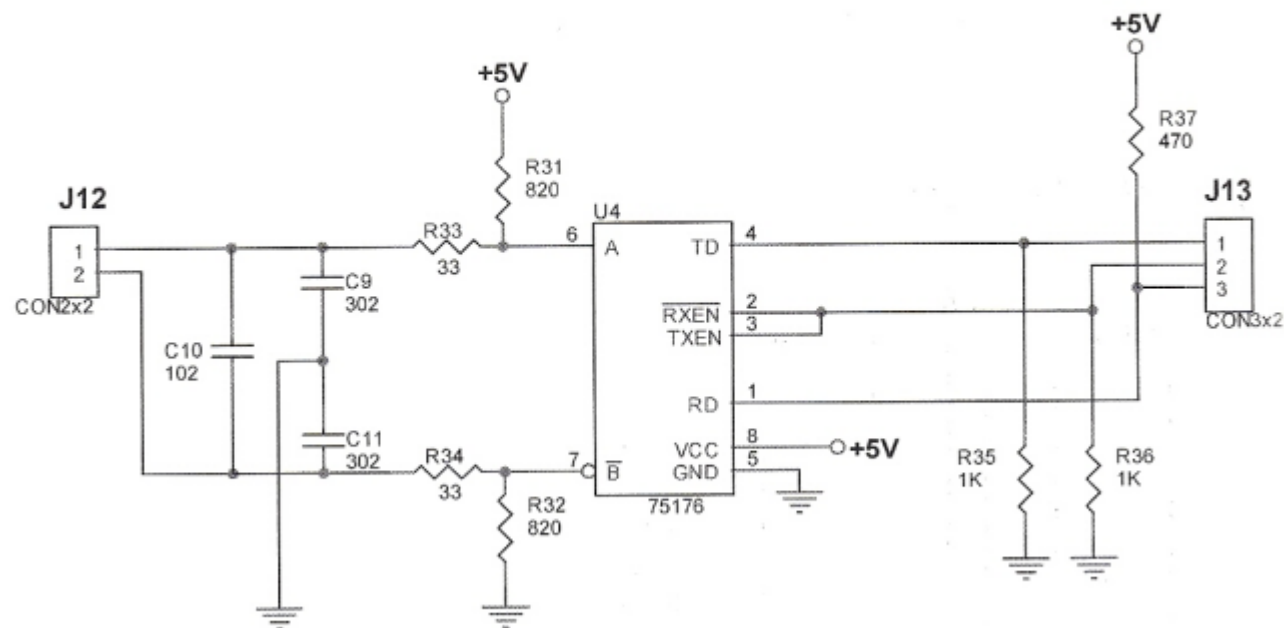
## 16. 외부 회로 연결용 단자대

- 용도: 별도로 주어지는 확장 보드를 이용하여 다양한 실험을 할 수 있음.

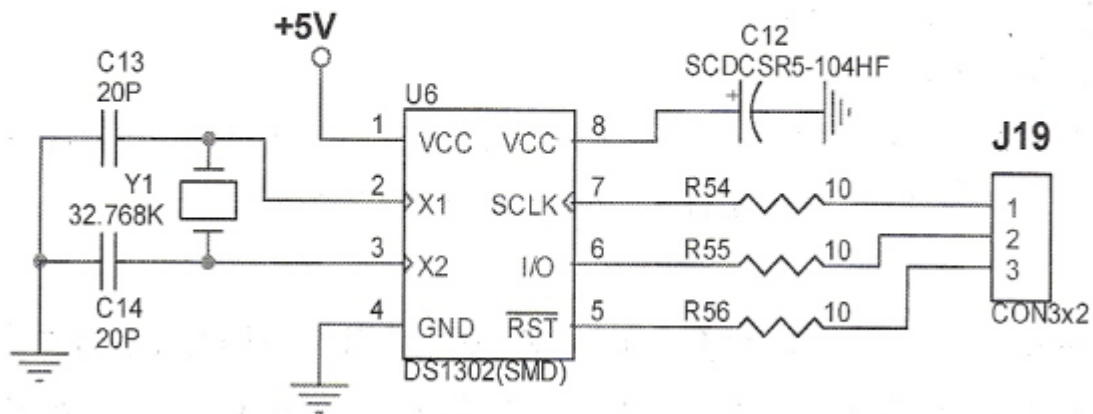
HARDWARE 구성은 CABLE로 연결하면 됨.



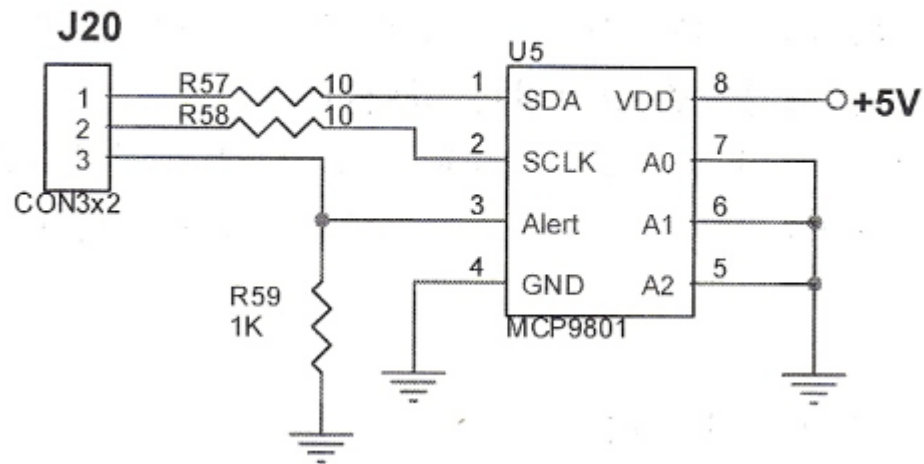
## 17. RS485 통신 INTERFACE



## 18. RTC(REAL TIME CLOCK) 회로



## 19. 온도 측정 회로



# 어셈블리 프로그래밍

## ▶ 어셈블리 명령어 기본 형식

**label**            **mnemonic**            **operand**            **comment**

예) HERE    ADDLW    20H            ; (W) + 20H → (W)

## ▶ 명령어의 operand 상수 표현 방법

type	표현	예
10진수	D'10진수', .10진수	D'123', <b>.234</b>
16진수	H'16진수', 0x16진수, 16진수H	H'123', 0x3F, 0F4H, <b>78</b>
2진수	B'2진수'	B'1010'
문자	'문자'	'W'

## 자주 사용되는 지시어

- ① LIST : list file option으로 특히 마이크로프로세서의 TYPE를 선언함.

```
list p=<type>
```

- ② PROCESSOR : 마이크로프로세서의 TYPE를 선언함. LIST와 동일

```
processor 16f84
```

- ③ ORG : origin의 의미로 프로그램이 수행되는 시작번지를 의미한다.

```
<label> org < expression >
```

- ④ INCLUDE : 주어진 file를 삽입하여 사용하라.

```
include <<include_file>> : 첫 번째 열부터 시작함  
include "c:\mr_pic\3\16f84.h" OR <PIC16F84.h>
```

- ⑤ EQU : equal의 의미로 사용되며, 선언된 label 값이 expression이 된다.

```
<label> equ <expr> : 첫 번째 열부터 시작함
```



---

LOOP_CNT1	EQU	11H
LOOP_CNT2	EQU	12H

cf.) 사용자 변수

VARIABLE	LOOP_CNT1	=	11H
VARIABLE	LOOP_CNT2	=	12H

⑥ END: end의 의미로 compiler에게 번역을 끝내라는 의미로 사용된다.

→ 항상 source program의 끝에 위치해야한다.

→ 이 명령어 다음에 프로그램이 들어 있어도 번역하지 않는다.

⑦ IF: ELSE: ENDIF : if 문의 시작과 끝을 나타냄

```
if <expr> ~  
    else if OR else ~  
endif
```

⑧ MACRO: ENDM : macro 정의 시작과 끝을 나타냄

```
<label> macro <<arg>,,,<arg>> ~ endm
```

⑨ EXITM : macro 문에서 빠져나올 때 사용

예) test macro num, count

```
if num==count
```

```
    movlw 1
```

```
    exitm
```

```
else
```

```
    movlw 0
```

```
endif
```

```
    movlw 4
```

```
endm
```

## ABC.ASM 소스 파일 어셈블링 결과 생성 파일

---

파일명	설명
ABC.COD	에뮬레이터용 코드파일
ABC.HEX	인텔 HEX 파일(프로그래머에서 사용)
ABC.LST	프린트할 수 있는 리스팅 파일
ABC.ERR	에러 발생시 생성되는 에러 레포트 파일

