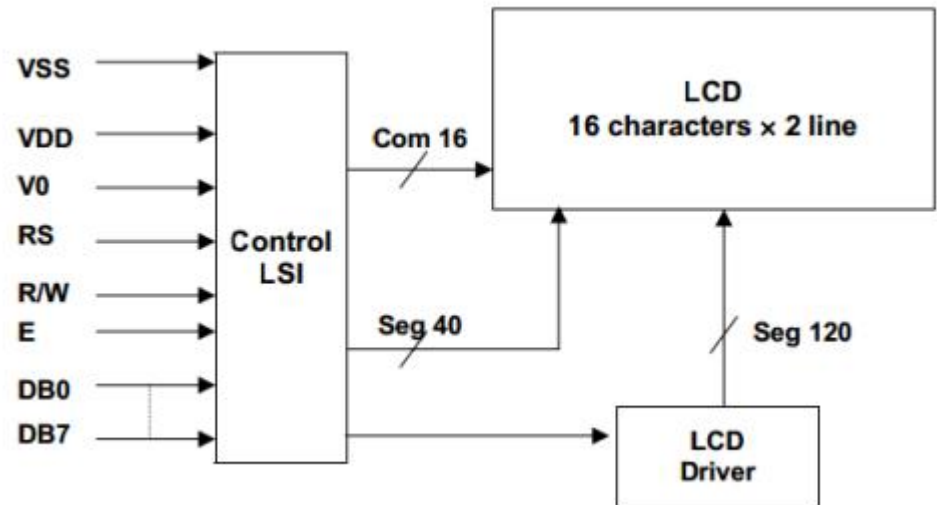
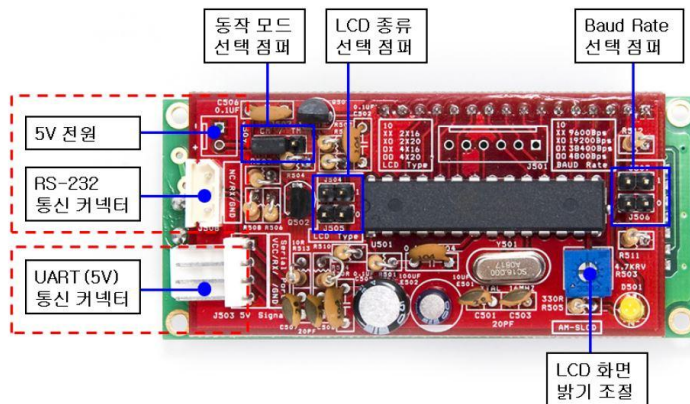


전자회로실험2

Lab3. USART LCD

LCD



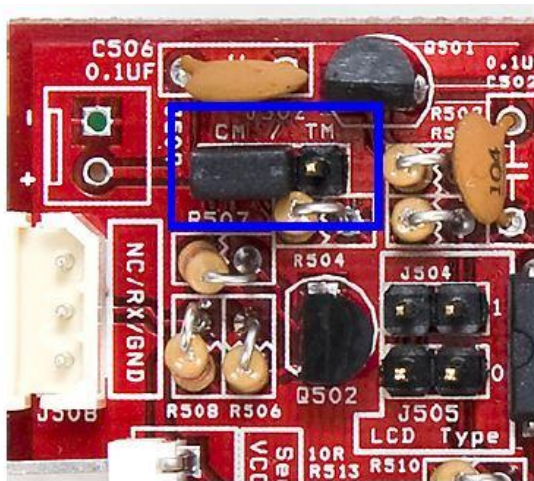
No.	Symbol	Function
1	VSS	Ground, 0V
2	VDD	Logic power supply, +5V
3	V0	Voltage for LCD drive
4	RS	Data / Instruction register select
5	R/W	Read / Write
6	E	Enable signal, start data read/write
7	DB0	Data Bus Line
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED A	LED Anode, power supply +
16	LED K	LED Cathode, ground 0V

-LCD 후면에 Atmega 8A 기반의 컨트롤 보드가 이미 부착되어 있음

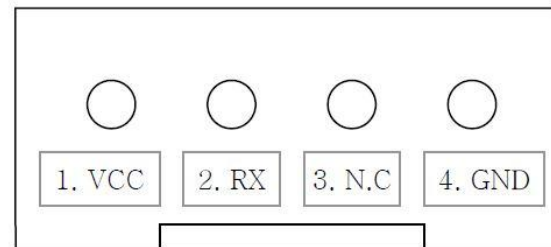
UART LCD

3 Pin 점퍼를 설정하여 터미널 모드(TM) / 커맨드 모드(CM) 를 설정

- 터미널 모드 (점퍼를 "TM" 위치로 설정 또는 제거)
- 커맨드 모드 (점퍼를 "CM" 위치로 설정)



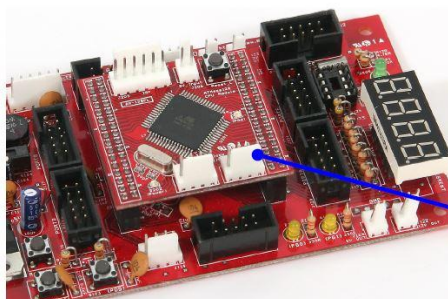
동작모드 선택점퍼



커넥터 배선 방법 (※ 커넥터로 입력 되는 신호는 TTL Level (5V)입니다.)

1	VCC (5V 출력)
2	UART RX (시리얼 데이터 수신)
3	No Connect
4	GND

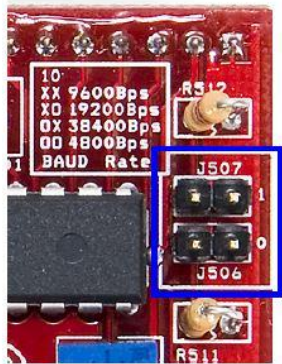
UART커넥터로 MCU보드와 연결



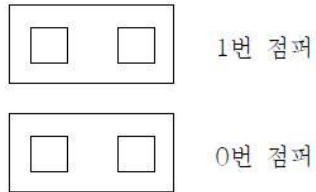
128 메인보드와 연결함

UART LCD

통신속도 설정



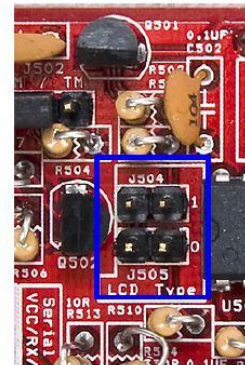
왼쪽 그림의 2Pin 점퍼 2개를 이용하여 통신속도를 설정할 수 있습니다.



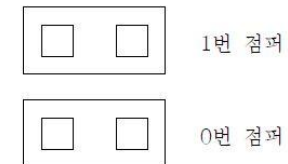
Baud Rate 설정



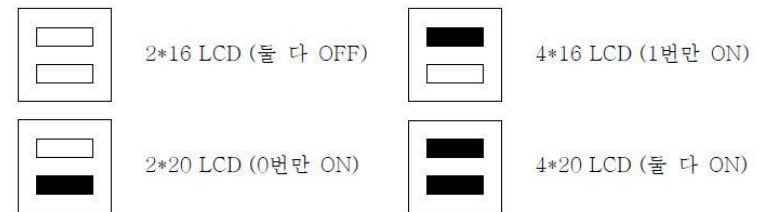
LCD종류 설정



왼쪽 그림의 2Pin 점퍼 2개를 이용하여 LCD 종류를 설정할 수 있습니다.



LCD 종류 설정



LCD

AM-SLCD 커맨드 표

구분	명령	Data1	Data2	End	Example
초기화	\$I			<CR>	\$I<CR>
화면 Clear	\$C			<CR>	\$C<CR>
커서 위치 지정	\$G	행 위치(1-4)	열 위치(1-20)	<CR>	\$G,1,1<CR>
문자열 출력	\$T	Text		<CR>	\$T,Testing...<CR>
커서 OFF/ON/ Blink	\$B	0/1/B		<CR>	\$B,1<CR>
Display (On/Off)	\$D	1/0		<CR>	\$D,1<CR>
Back Light (On/Off)	\$L	1/0		<CR>	\$L,1<CR>
Display Shift Left / Right	\$S	L/R		<CR>	\$S,R<CR>

UART LCD

- UART 통신을 통해서 해당 보드로 다음 슬라이드에 있는 표에 해당하는 프로토콜에 따라 문자열을 전송하면, LCD 에 원하는 출력값이 나타남

- 예시: 2번째 줄 3번째 칸에서 부터 Hello 문자열을 출력하고 싶은 경우

UART 를 통해 다음과 같은 문자열 전송

\$I\n\r\$G,2,3\n\r\$T,Hello\n\r

- AVR 예제 소스

```
printf("$B,0\r");           // 커서 끄기
printf("$D,1\r");           // LCD 화면 켜기
printf("$L,1\r");           // 백라이트 켜기
while(1){
    printf("$C\r");          // 화면 지우기
    printf("$G,1,1\r");      // 커서를 첫번째줄 첫번째칸으로 이동
    printf("$T,N.T.C Testing..\r"); // "N.T.C Testing.." 문자 출력
    printf("$G,2,1\r");      // 커서를 두번째줄 첫번째칸으로 이동
    printf("$T,Temp : %d\r", temp); // "Temp : XXX " 문자 출력
    temp ++;
    delay(1000);
}
```


LCD

```
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1

#include <avr/io.h>
#include <util/delay.h>

void USART_Init(unsigned int ubrr){
    /*
     * Baud Rate 9600, Stop Bit 1
     * Character Size: 8-Bit
     * No Parity
     */
    UBRROH = (unsigned char)(ubrr >> 8);
    UBRROL = (unsigned char)ubrr;
    UCSROB = (1<<RXEN0) | (1<<TXEN0);
    UCSROC = (3 << UCSZ0);
}
```

```
void USART_Transmit(char data){
    while(!((UCSROA) & (1<<UDRE0))); //Data Register Not Empty: Wait
    UDRO = data;                      //Data Register Empty: Transmit
}
```

```
void USART_Transmit_String(char *str){
    while(*str != '\0')
        USART_Transmit(*str++);
}
```

```
char USART_Receive(){
    while(!(UCSROA & (1<<RXCO))); //Receive Not complete: Wait
    return UDRO;                  //Receive Complete : Receive
}
```

```
int main(void)
{
    USART_Init(MYUBRR);
    char str[] = "Hello"; //String that

    USART_Transmit_String("$I\r"); //LCD Initialize
    USART_Transmit_String("$G,1,1\r"); //Cursor setup at 1,1
    USART_Transmit_String("$T,");
    USART_Transmit_String(str);
    USART_Transmit('\r');

    while (1)
    {
        _delay_ms(100);
    }
}
```

LCD에 Hello 라는 문구가 출력됨

과제

1. LCD를 초기화 하는 함수를 작성하시오

2. 다음과 같은 함수를 작성하시오

`void LCD_Transmit(int x, int y, char data)`

- 함수는 커서의 x 축 위치 int x, y 축 위치 int y, 출력 데이터 char data를 받아서 x,y 위치에 data에 해당하는 문자를 출력

`void LCD_Transmit_String(int x, ,int y, char *string)`

- 함수는 커서의 x 축 위치 int x, y 축 위치 int y, 출력 데이터 char data를 받아서 x,y 위치로 부터 data에 해당하는 문자를 출력

과제

3. 2에서 작성한 함수를 바탕으로 다음과 같은 프로그램을 작성하시오

- LCD에 '팀원 중 한명의 영어이름' Department of Electronic Engineering의 문자열이 출력되도록 설정
- 이때 문자열의 길이가 LCD의 최대 출력인 32 글자를 넘어가게 되므로, 글자가 한글자씩 흘러가도록 출력