
마이크로컨트롤러 기능

제 10장 외부 메모리 인터페이스



ICAT

Integrated Circuits for Advanced Technology Lab.

외부 메모리 인터페이스

1. TEXT LCD
2. ATmega128의 외부메모리 인터페이스
3. SRAM
4. TEXT LCD에 글자쓰기
5. 외부메모리 인터페이스에 SRAM붙이기



TEXT LCD

□Text LCD(Character LCD)

- ❖ LCD화면에 정해진 형태의 문자를 정해진 개수만큼 표시할 수 있도록 만들어진 LCD 디스플레이 장치
- ❖ 각종 임베디드 장치들에서 널리 사용
- ❖ 대부분의 마이크로컨트롤러들과 TEXT LCD는 그대로 연결할 수 있도록 신호를 제공
- ❖ 마이크로 컨트롤러 개발 환경에서도 TEXT LCD 관련 함수를 제공

□HBE-MCU-Multi Text LCD

- ❖ 16문자*2라인의 표시부
- ❖ Backlight 기능 포함



ICAT

Integrated Circuits for Advanced Technology Lab.

TEXT LCD

인터페이스 커넥터 핀 기능

핀	Signal Name	기능
1	VSS	전원 GND
2	VDD	전원 +5VDC
3	VEE	Contrast 제어 전압레벨 (VDD-VEE = 13.5 ~ 0V)
4	RS	Register Select (0 = instruction, 1 = data)
5	R/W	Read/Write (0 = FPGA -> LCD, 1 : FPGA <- LCD)
6	E	Enable Signal for read/write LCD
7	DB0 (LSB)	DATA
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7 (MSB)	
15	A	+LED (backlight LED용 전원 +4.4 ~ 4.7V)
16	K	-LED (backlight LED용 전원 GND)



TEXT LCD

□Text LCD(Character LCD) 구조

- ❖ TEXT LCD는 보통 LCD 표시부와 LCD 제어부를 하나로 하여 LCD 모듈로 시판됨.
- ❖ LCD제어기 구성
 - 명령(Instruction)과 데이터(Data)를 위한 2개의 레지스터
 - BF (Busy Flag)
 - AC(Address Counter)
 - 문자발생램(CGRAM)
 - 문자발생롬(CGROM)
 - 데이터표시램(DDRAM)



TEXT LCD

□Text LCD(Character LCD) 구조

❖ 내장 레지스터

- 명령레지스터(IR):
 - DDRAM과 CGRAM에 대한 어드레스와 클리어, 커서시프트 등 제어명령을 보
유
- 데이터레지스터(DR):
 - DDRAM과 CGRAM에 쓴 데이터나 읽은 데이터를 일시적으로 저장
- 레지스터들은 RS(4번핀)과 R/W(5번핀)을 사용하여 선택

RS, R/W 신호에 따른 TEXT LCD 제어기 레지스터 선택방법

RS	R/W	레지스터 접근
0	0	IR쓰기(각종 제어 명령 쓰기)
0	1	BF읽기, AC읽기
1	0	DR쓰기
1	1	DR읽기



TEXT LCD

□ Text LCD(Character LCD) 구조

❖ BF (Busy Flag)

- 1이면 LCD의 컨트롤러가 동작 중으로 명령 수행 불능
- 0이면 다음 명령 수행 가능 표시

❖ AC(Address Counter)

- DDRAM과 CGRAM의 주소를 지정할 때 사용
- IR에 주소 정보를 쓰면 주소 정보가 AC로 전송
- DDRAM/DDROM에 데이터를 쓰면 AC는 자동으로 +1 혹은 -1이 됨

❖ 문자발생램(CGRAM)

- 사용자가 자유로이 **문자를 만들 때 사용하는 램**
- **5x7은 8개, 5x10은 4개 만들어 저장 가능**

❖ 문자발생롬(CGROM)

- 5x7, 5x10 도트의 **문자를 내장, 특수기호, 숫자, 영문자의 문자코드는 아스키코드와 일치**

❖ 데이터표시램(DDRAM)

- 80x8비트 용량으로 **80개의 8비트 아스키(ASCII)코드를 저장**
- **0x00-0F 주소가 LCD 1행의 1-16번째, 0x40-4F 주소가 LCD 2행의 1-16번째 문자로 표시됨.**



ICAT

Integrated Circuits for Advanced Technology Lab.

TEXT LCD

□Text LCD 제어 명령

❖ Text LCD DDRAM Address

- DDRAM은 표시될 각 문자의 ASCII 코드 데이터가 저장되어 있는 메모리
- 모두 80개의 번지가 있으며, 화면의 각 행과 열의 위치에는 고유한 어드레스 값이 부여되어 있음.
- 각 행과 행 사이의 어드레스가 연속하여 있지 않으므로 주의해야 함.

표시 문자의 위치에 대한 DDRAM의 어드레스

구분	0	1	2	3	13	14	15	
Line1	00	01	02	03	0D	0E	0F	
Line2	40	41	42	43	4D	4E	4F	



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128의 외부메모리 인터페이스

□ ATMega128의 외부메모리 인터페이스

- ❖ 칩 외부에 더 큰 용량의 메모리나 별도의 주변장치들을 연결하기 위한 인터페이스
 - 외부 램, 외부 플래시 롬
 - LCD나 DA변환기와 같은 주변장치
- ❖ 노말모드에서 0x1100-0xffff번지에 외부 데이터 메모리로 사용가능
- ❖ 외부 메모리 인터페이스 기능
 - 주변장치와 적절한 인터페이스를 위한 0-3의 대기 사이클을 지정 가능.
 - 2개의 섹터로 외부 데이터 메모리를 분할하고, 독립적인 대기 사이클 지정가능.
 - 16비트 주소의 상위바이트에 중 필요한 갯수의 비트만을 주소 버스로 동작 가능.



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128의 외부메모리 인터페이스

□ 외부메모리 인터페이스 신호

❖ ATMega128의 외부 메모리 연결을 위한 핀

- PA7~PA0(핀44~핀51) : AD(7:0)
 - 외부 메모리 하위주소와 데이터버스.
- PC7~PC0(핀35~핀42) : A(15:8)
 - 외부 메모리 상위주소 버스.
- ALE(PG2, 핀43) : Address Latch Enable
 - 외부메모리에 접근할 때 PA에서 출력되는 하위주소값을 래치할 수 있도록 CPU에서 1레벨 값을 출력해준다.
- RD(PG1, 핀34)
 - 외부 데이터 메모리를 읽을(Read) 때 사용되는 스트로브 신호.
- WR(PG0, 핀33)
 - 외부 데이터 메모리에 쓸(Write) 때 사용되는 스트로브 신호.

❖ MCUSR레지스터를 이용해서 설정



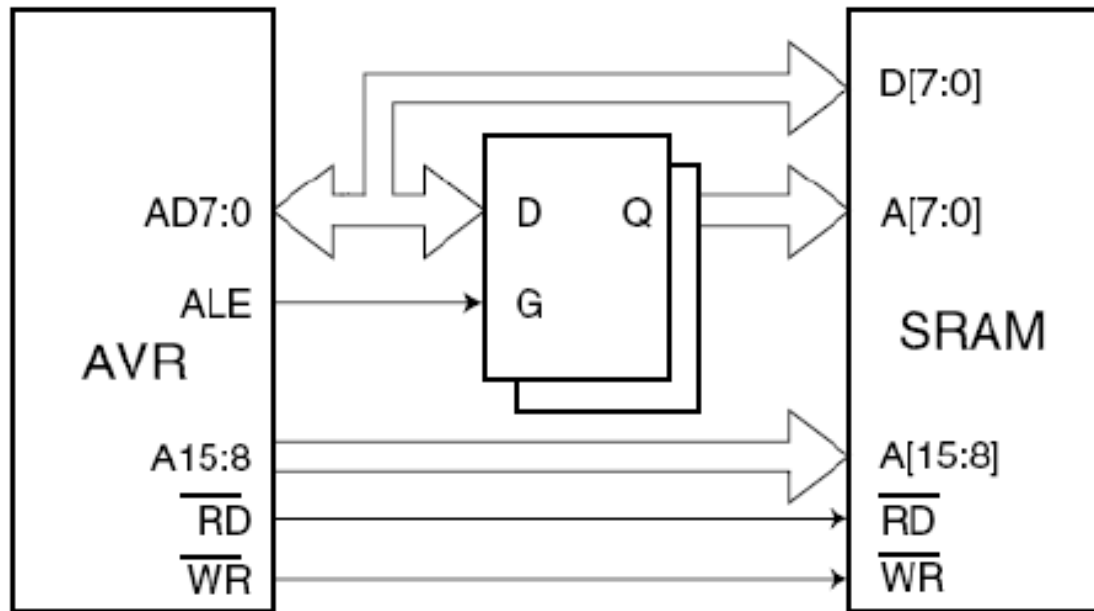
ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128의 외부메모리 인터페이스

□ 외부메모리의 연결

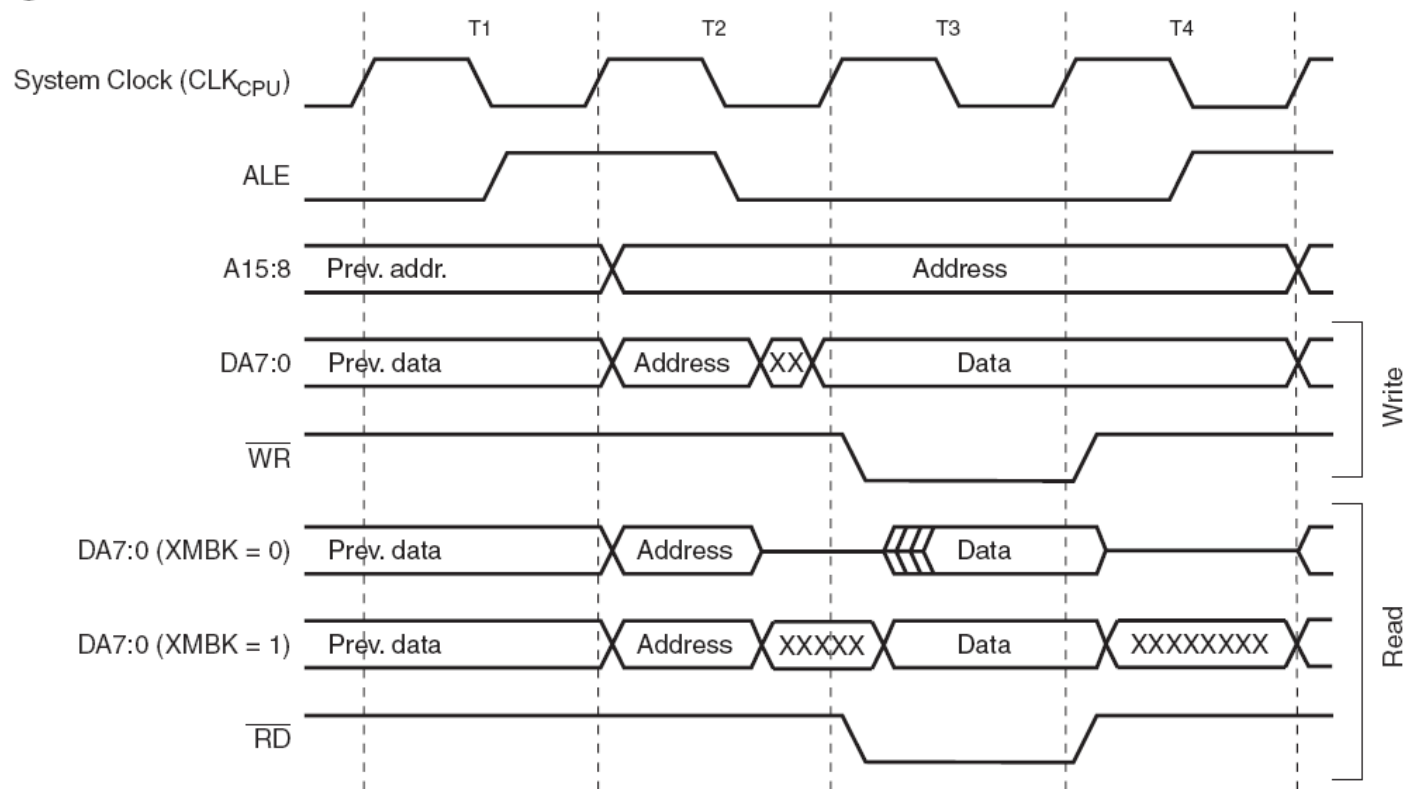
- ❖ 다중화 버스(Multiplex-BUS)방식
 - AVR 칩의 핀수를 절약



ATMega128의 외부메모리 인터페이스

□외부메모리 읽기 쓰기

외부 메모리 인터페이스와 외부 메모리 사이의 신호 흐름



ATMega128의 외부메모리 인터페이스

□외부 메모리 인터페이스 설정 레지스터

❖ MCUCR(MCU Control Register) :

- MCU 제어 레지스터

❖ XMCRA(eXternal Memory Control Register A) :

- 외부 메모리 제어 레지스터 A

❖ XMCRB(External Memory Control Register B) :

- 외부 메모리 제어 레지스터 B



ATMega128의 외부메모리 인터페이스

□MCUCR(MCU Control Register)

- ❖ MCU 제어 레지스터
- ❖ CPU의 전체적인 기능을 설정
 - 외부 메모리 설정
 - Sleep Mode 동작 설정
 - 인터럽트 동작 설정

7	6	5	4	3	2	1	0
SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE



ATMega128의 외부메모리 인터페이스

□MCUCR(MCU Control Register)

❖ 외부 메모리 설정

- 비트 7과 6을 사용해서 외부데이터 메모리 영역을 설정
- 비트 7(SRE:external SRAM/XMEM Enable)
 - 1로 세트하면 PA(7:0), PC(7:0), PG(2:0) 핀들을 외부메모리와 인터페이스를 위한 핀들로 설정
 - 0로 하면 일반 병렬포트로 설정
- 비트 6(SRW10:Wait-state Select Bit for Upper Sector)
 - 외부 데이터의 Upper 섹터에 부여하는 여분의 대기 사이클의 수를 XMCRA레지스터의 SRW11비트와 조합되어 설정.
 - [SRW11:SRW10]=00일때 대기 사이클은 0개, 01일때 1개, 10일때 2개, 11일때 3개를 준다.



ATMega128의 외부메모리 인터페이스

□MCUCR(MCU Control Register)

❖ Sleep 모드 동작 설정

- 비트 5, 비트4,3,2를 이용해 Sleep 모드를 설정.
- 비트 5(SE:Sleep Enable) : 디폴트는 0으로 1로 세트되면 SLEEP 명령시 슬립모드의 돌입이 허용한다.
- 비트 4,3,2(SM1,SM0,SM2) : 슬립 모드 설정

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby ⁽¹⁾
1	1	1	Extended Standby ⁽¹⁾



ATMega128의 외부메모리 인터페이스

□MCUCR(MCU Control Register)

❖ 인터럽트 동작설정

- 비트 1과 0을 사용해서 인터럽트 동작을 설정
- 비트 1(IVSEL:Interrupt Vector SElect) : 인터럽트 벡터의 배치를 변경
- 비트 0(IVCE:Interrupt Vector Change Enable) : 1로 세트하면 IVSEL을 변경할 수 있다.

인터럽트 벡터의 배치

BOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	\$0000	\$0002
1	1	\$0000	Boot Reset Address + \$0002
0	0	Boot Reset Address	\$0002
0	1	Boot Reset Address	Boot Reset Address + \$0002



ATMega128의 외부메모리 인터페이스

□XMCRA(eXternal Memory Control Register A)

❖ 외부 메모리 제어 레지스터 A

- 외부 데이터 메모리 영역을 분할
- 대기 사이클의 수를 설정

7	6	5	4	3	2	1	0
-	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11	-

❖ 비트 6(SRL2)~4(SRL0)

- 외부 데이터 메모리 영역을 Lower와 Upper 2개의 섹터로 분할하여 각 섹터에 별도로 대기 사이클의 수를 설정



ATMega128의 외부메모리 인터페이스

□XMCRA(eXternal Memory Control Register A)

SRL2-SRL0를 이용한 메모리 분할

SRL2	SRL1	SRL0	Sector Limits
0	0	0	Lower sector = N/A Upper sector = 0x1100 - 0xFFFF
0	0	1	Lower sector = 0x1100 - 0x1FFF Upper sector = 0x2000 - 0xFFFF
0	1	0	Lower sector = 0x1100 - 0x3FFF Upper sector = 0x4000 - 0xFFFF
0	1	1	Lower sector = 0x1100 - 0x5FFF Upper sector = 0x6000 - 0xFFFF
1	0	0	Lower sector = 0x1100 - 0x7FFF Upper sector = 0x8000 - 0xFFFF
1	0	1	Lower sector = 0x1100 - 0x9FFF Upper sector = 0xA000 - 0xFFFF
1	1	0	Lower sector = 0x1100 - 0xBFFF Upper sector = 0xC000 - 0xFFFF
1	1	1	Lower sector = 0x1100 - 0xDFFF Upper sector = 0xE000 - 0xFFFF



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128의 외부메모리 인터페이스

□XMCRA(eXternal Memory Control Register A)

❖ 비트 3~2(SRW01~SRW00)

- Wait-state Select Bit for Lower Sector
- 외부 데이터의 **Lower 섹터**에 부여하는 여분의 대기 사이클의 수 설정
- 대기 사이클은 [SRW01:SRW00]=00일때 0개, 01일때 1개, 10일때 2개, 11일때 3개

❖ 비트 1(SRW11)

- Wait-state Select Bit for Upper Sector)
- 외부 데이터의 **Upper 섹터**에 부여하는 여분의 대기 사이클의 수를 XMCRA레지스터의 SRW11비트와 조합되어 설정
- 대기 사이클은 [SRW11:SRW10] =00일때 0개, 01일때 1개, 10일때 2개, 11일때 3개



ATMega128의 외부메모리 인터페이스

□XMCRB(eXternal Memory Control Register B)

❖ 외부 메모리 제어 레지스터 B

- 버스 키퍼 기능을 설정
- 외부 메모리 주소의 상위 바이트에서 어느 부분까지가 주소지정에 사용되는지를 설정

7	6	5	4	3	2	1	0
XMBK	-	-	-	-	XMM2	XMM1	XMM0

❖ 비트 7(XMBK:eXternal Memory Bus-Keeper enable)

- 1로 세팅되면 PA7-PA0의 값들이 보존되는 버스키퍼 기능을 활성화시킨다.



ATMega128의 외부메모리 인터페이스

□XMCRB(eXternal Memory Control Register B)

❖ 비트 2~0(XMM2~0 : eXternal Memory high Mask)

- 상위주소 버스로 사용되는 port C의 일부분만이 주소버스로 사용 되도록 하고 나머지는 입출력 포트로 설정

외부 메모리 허용시 해방되는 c 포트 핀

XMM2	XMM1	XMM0	# Bits for External Memory Address	Released Port Pins
0	0	0	8 (Full 60 KB space)	None
0	0	1	7	PC7
0	1	0	6	PC7 - PC6
0	1	1	5	PC7 - PC5
1	0	0	4	PC7 - PC4
1	0	1	3	PC7 - PC3
1	1	0	2	PC7 - PC2
1	1	1	No Address high bits	Full Port C



ICAT

Integrated Circuits for Advanced Technology Lab.

SRAM(Static RAM)

□SRAM(static random access memory)

- ❖ 전력이 공급되는 한, 메모리 내의 데이터 비트들의 내용이 계속 유지되는 램
- ❖ DRAM과는 달리, SRAM은 주기적으로 재생시킬 필요가 없다
- ❖ DRAM에 비해 더 빠르게 데이터에 액세스할 수 있음.
- ❖ 값이 상대적으로 비싸다
- ❖ 임베디드 시스템의 고속 데이터 처리용, 컴퓨터의 캐시메모리 와 비디오카드의 RAMDAC의 일부로 사용됨.



ICAT

Integrated Circuits for Advanced Technology Lab.

SRAM(Static RAM)

□SRAM의 핀 기능

핀 이름	기능
/CS	Chip Select
/OE	Output Enable Input
/WE	Write Enable Input
I/O0~I/O7	Data Input/Output
A0~A16	Address Input

- ❖ **CS** : 메모리 버스에 연결된 칩들을 구분하기 위한 신호
- ❖ **OE** : CS에 의해 선택된 메모리의 데이터를 읽어 낼 때 사용
- ❖ **WE** : 선택된 메모리 칩에 데이터를 써넣을 때 사용
- ❖ **I/O0~I/O7** : 데이터 버스, 데이터를 실어 보냄
- ❖ **A0~A16** : 어드레스 버스, 메모리의 주소를 알려줌



실습 15 : TEXT LCD에 글자쓰기

□실습 개요

- ❖ ATmega128의 GPIO에 TEXT LCD를 연결하고, LCD 화면에 미리 작성된 문장("Hello! MCU World !!")을 표시
- ❖ AVR 개발 환경에서 제공하는 TEXT-LCD 관련 함수를 이해하면 쉽게 프로그램을 작성할 수 있음.

□실습 목표

- ❖ TEXT LCD의 동작 원리 이해
- ❖ AVR 개발환경에서 제공하는 TEXT-LCD 관련 함수 이해
- ❖ TEXT LCD 제어 프로그램 방법 습득



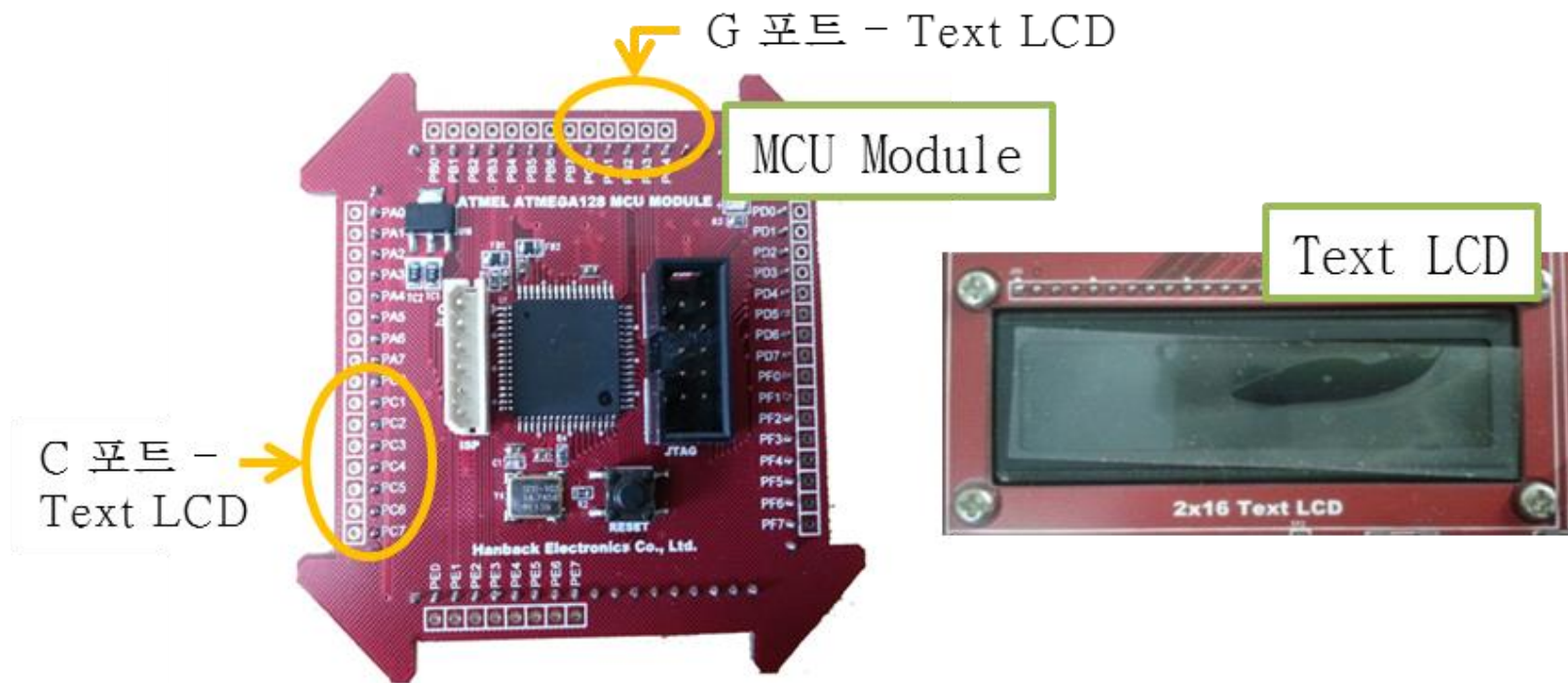
ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : TEXT LCD에 글자쓰기

□ 사용 모듈

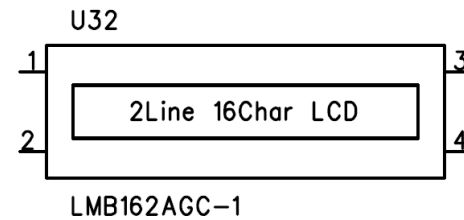
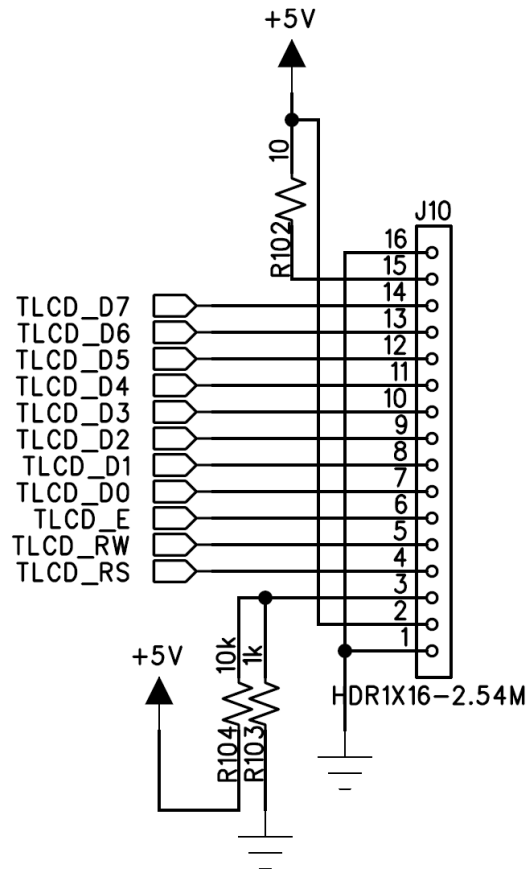
- ❖ MCU 모듈, TEXT-LCD 모듈



실습 15 : TEXT LCD에 글자쓰기

□ 사용 모듈

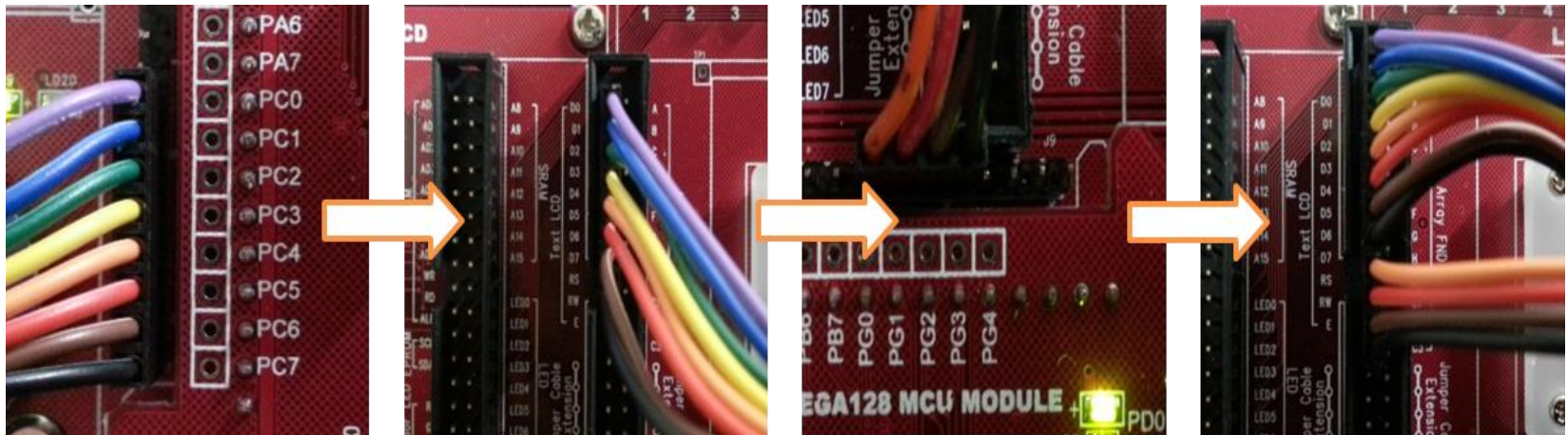
❖ TEXT LCD 모듈 회로



실습 15 : TEXT LCD에 글자쓰기

□모듈 결선 방법

- ❖ MCU 모듈 포트 C의 PC0~PC7을 TEXT LCD 모듈의 D0~D7까지 연결
- ❖ MCU 모듈 포트 G의 PG0는 TEXT LCD 모듈의 RS, PG1은 RW, PG2는 E핀에 연결



실습 15 : TEXT LCD에 글자쓰기

□ 구동 프로그램 : 사전지식

- ❖ AVR 개발 환경에서 TEXT LCD 관련 라이브러리 함수를 제공
 - lcd.c 와 avr_lib.c 라는 파일에 포함
 - AVR Studio에서 이미지를 Build할 때 이 lcd.c파일과 avr_lib.c 파일을 함께 포함시켜야 함.
- ❖ TEXT LCD용 라이브러리 함수
 - lcdInit : TEXT LCD 초기화
 - lcdGotoXY(unsigned char x, unsigned char y) : TEXT LCD의 커서를 원하는 위치로 이동.
 - lcdDataWrite(unsigned char data): TEXT LCD에 하나의 문자 출력
 - lcdPrintData(char* data, unsigned char nBytes) : TEXT LCD에 nBytes 길이의 문자열을 출력
 - lcdClear() : TEXT LCD의 화면을 지움.



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : TEXT LCD에 글자쓰기

□ 구동 프로그램 : 사용 포트와 핀 선언

❖ 사용할 포트를 선언하는 헤더화일 : lcdconf.h

- 이 예제에서는 MCU 모듈의 C 포트와 G 포트를 사용

lcdconf.h

```
#ifdef LCD_PORT_INTERFACE
    #ifndef LCD_CTRL_PORT
        #define LCD_CTRL_PORT    PORTG
        #define LCD_CTRL_DDR    DDRG
        #define LCD_CTRL_RS      0
        #define LCD_CTRL_RW      1
        #define LCD_CTRL_E       2
    #endif
    #ifndef LCD_DATA_POUT
        #define LCD_DATA_POUT    PORTC
        #define LCD_DATA_PIN     PINC
        #define LCD_DATA_DDR     DDRC
    #endif
#endif
```

TEXT LCD
제어포트
선언

TEXT LCD
데이터
포트선언



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : TEXT LCD에 글자쓰기

□ 구동 프로그램 : 사용 포트와 핀 선언

❖ 사용할 포트를 선언하는 헤더파일 : lcdconf.h

TEXT LCD 제어 포트 선언

매크로 선언	매크로 상수 의미	예
#define LCD_CTRL_PORT	포트 데이터 레지스터	PORTA, PORTC
#define LCD_CTRL_DDR	데이터 방향 레지스터	DDRA, DDRC
#define LCD_CTRL_RS	TextLCD RS핀	0 ~ 7 의 값
#define LCD_CTRL_RW	TextLCD RW핀	0 ~ 7 의 값
#define LCD_CTRL_E	TextLCD E핀	0 ~ 7 의 값

TEXT LCD 데이터 포트 선언

매크로 선언	매크로 상수 의미	사용 예
#define LCD_DATA_POUT	포트 데이터 레지스터	PORTA, PORTC
#define LCD_DATA_PIN	입력 핀 어드레스	PINA, PINC
#define LCD_DATA_DDR	데이터 방향 레지스터	DDRA, DDRC



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : TEXT LCD에 글자쓰기

□ 구동 프로그램 : 소스 분석

❖ Textlcd.c

1)	<pre>#include<avr/io.h> #include"lcd.h" int main(){</pre>
2)	<pre>lcdInit();</pre>
3)	<pre>lcdGotoXY(0,0);</pre>
4)	<pre>lcdDataWrite('H'); // 'H' 출력 lcdDataWrite('e'); // 'e' 출력 lcdDataWrite('l'); // 'l' 출력 lcdDataWrite('l'); // 'l' 출력 lcdDataWrite('o'); // 'o' 출력 lcdGotoXY(3,1); // 현재 커서위치를 (3,1) 위치로 이동</pre>
5)	<pre>lcdPrintData("MCU World !!",12); return 0; }</pre>



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : TEXT LCD에 글자쓰기

□ 실행 결과

H	e	l	l	o															
			M	C	U		W	o	r	l	d	!	!						



실습 15 : 외부메모리 I/F에 SRAM붙이기

□실습 개요

- ❖ ATmega128의 외부 메모리 인터페이스를 이용하여 SRAM을 제어하는 방법을 실습
- ❖ 외부 메모리 인터페이스에 SRAM과 TEXT LCD를 연결
- ❖ UART포트를 이용하여 PC와 통신 연결
- ❖ PC에서 입력하는 문자열을 SRAM에 한문자씩 차례로 저장하였다가, 한번에 꺼내서 TEXT LCD에 출력

□실습 목표

- ❖ ATmega128 외부 메모리 인터페이스 동작 원리 이해 (레지스터 설정)
- ❖ SRAM 동작 원리 이해
- ❖ SRAM의 제어 방법 습득



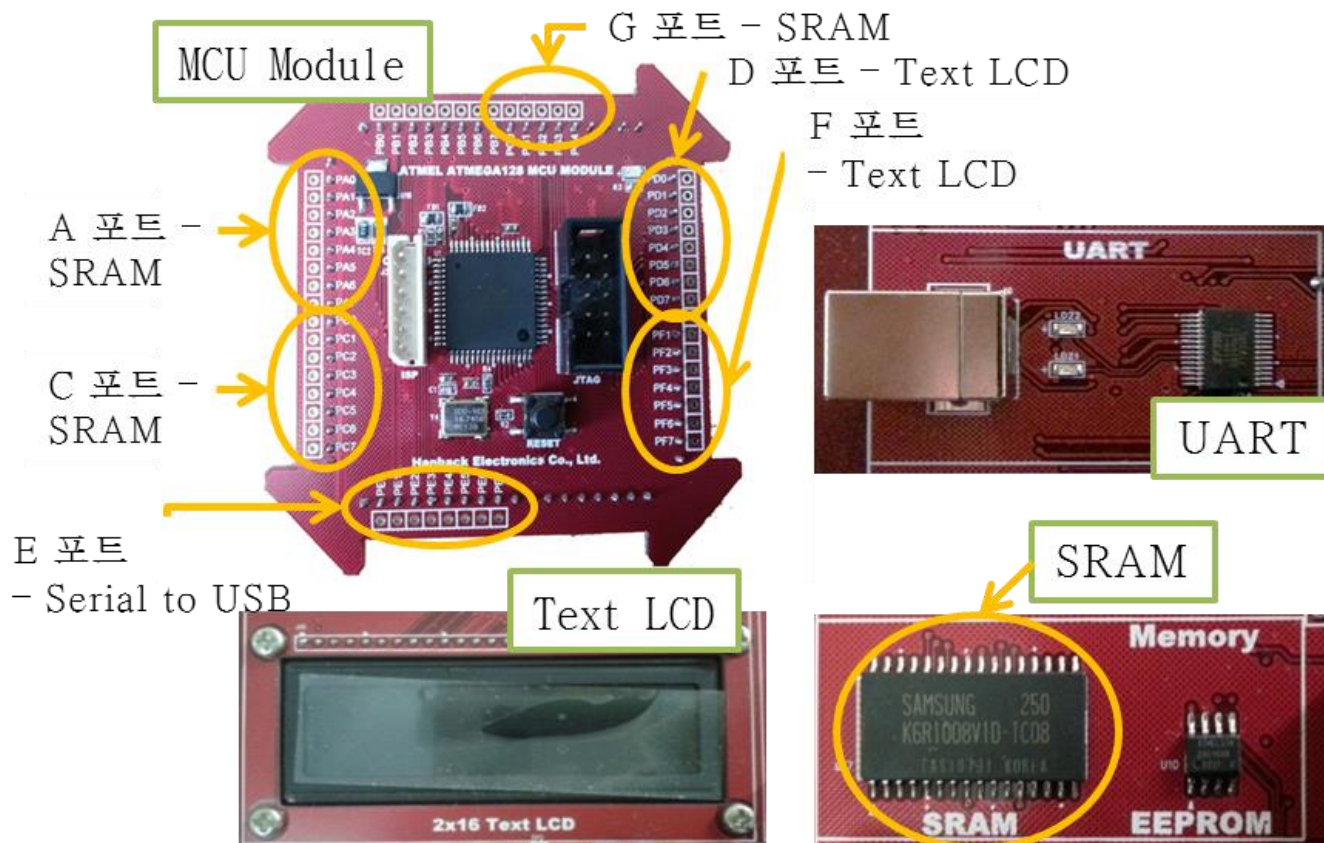
ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : 외부메모리 I/F에 SRAM붙이기

□ 사용 모듈

- ❖ MCU 모듈, SRAM 모듈, TEXT-LCD 모듈, UART 모듈

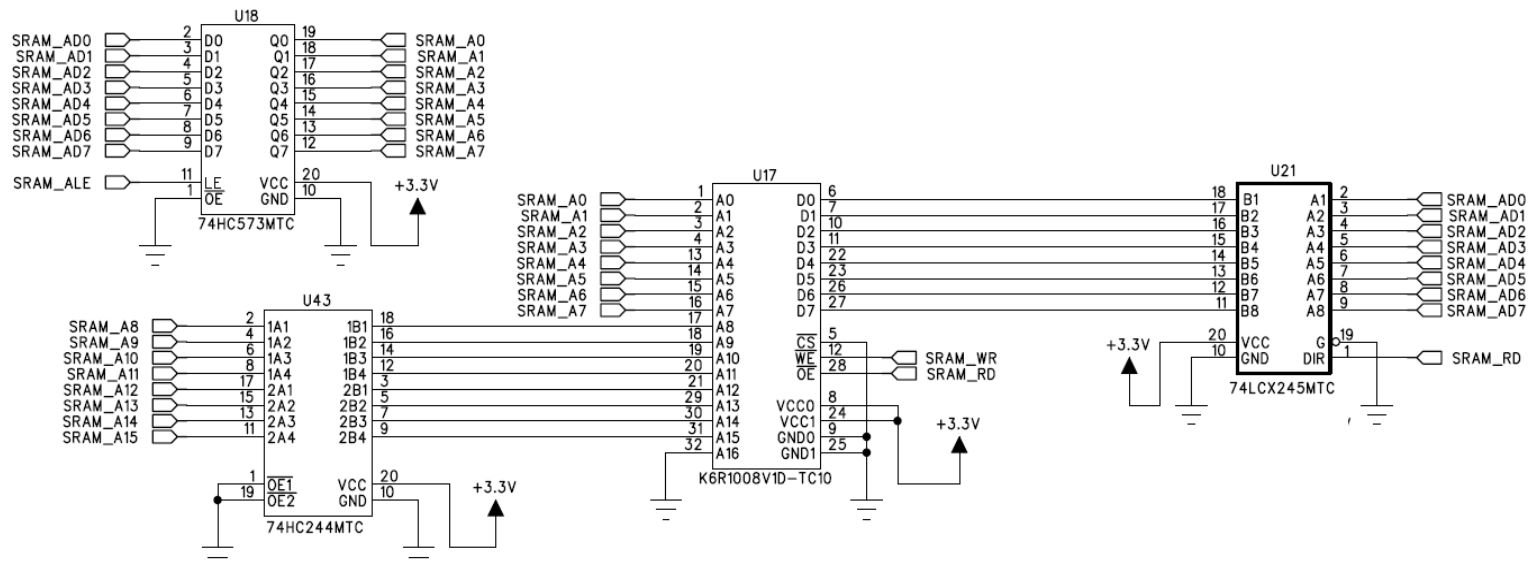


실습 15 : 외부메모리 I/F에 SRAM붙이기

□사용 모듈

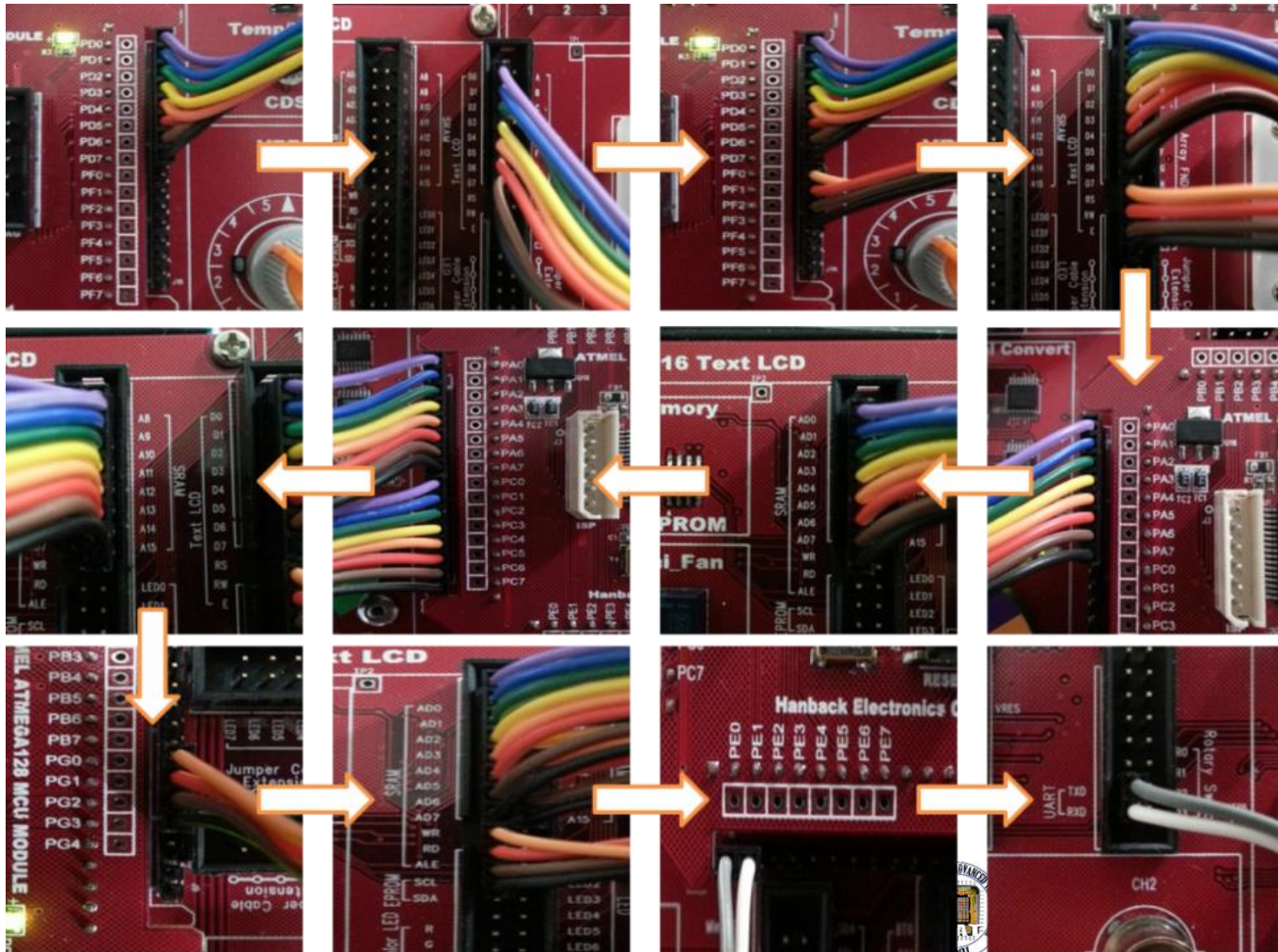
❖ 메모리 모듈의 SRAM부 회로

- 74LCX573 : D-Type Latch
- K6R1008 : 64KByte * 8 비트 SRAM



실습 15 : 외부메모리 I/F에 SRAM붙이기

□모듈 결선 방법(자세한 설명은 다음 장에서 기술)



실습 15 : 외부메모리 I/F에 SRAM붙이기

□모듈 결선 방법

MCU 모듈	TEXT LCD 모듈
PD0~PD7	D0~D7
PF0	RS
PF1	RW
PF2	E

MCU 모듈	UART 모듈
PE0	TXD
PE1	RXD

MCU 모듈	Memory 모듈
PA0~PA7	S_AD0~S_AD7
PC0~PC7	S_A8~S_A15
PG0	S_WR
PG1	S_RD
PG2	S_ALE



실습 15 : 외부메모리 I/F에 SRAM붙이기

❖ 구동 프로그램 : 사전 지식

❖ 외부 메모리를 사용을 위한 제어 레지스터 설정

- 외부 메모리 Enable
 - MCUCR 레지스터의 SRE 비트를 '1'로 세팅
- 메모리의 영역 분할 및 대기 사이클을 설정
 - XMCR A의 SRL(2:0) 비트를 설정하여 메모리의 섹터를 나눌 것인지, 아닌지를 결정
 - 상위(Upper)와 하위(Lower) 메모리 섹터들에 대한 대기 사이클(Wait-State)를 설정
 - 여기서는 메모리 영역을 0x1100~0x7FFF까지의 하위 섹터와 0x8000~0xFFFF까지의 상위 섹터로 나누어 사용
 - 각 메모리 섹터의 대기 사이클은 0
- 메모리를 위해 어느 정도의 핀을 할당할 것인지를 결정
 - XMCR B 레지스터의 XMBK 비트를 세팅하여 BUS-Keeper를 사용할 것인지 아닌지를 결정
 - 여기서는 BUS-Keeper는 사용하지 않고, 메모리는 60KByte 전체 영역을 사용
 - 따라서, C 포트의 모든 핀들은 외부 메모리(SRAM)에 할당
 - 외부 메모리의 주소영역 : 0x1100~0xFFFF



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : 외부메모리 I/F에 SRAM붙이기

❖ 구동 프로그램 : 사전 지식

❖ 외부 메모리(여기서는 SRAM)을 액세스하는 방법

- 포인터 변수를 이용하여 접근
- 외부 메모리의 주소영역이 0x1100~0xFFFF
- 메모리 읽기
 - `A = *(volatile unsigned char *)0x1100;`
- 메모리 쓰기
 - `*(volatile unsigned char *)0x1100 = 0xAA;`
- `volatile` : 컴파일러가 최적화 과정에 해당 메모리의 영역을 제거해 버리는 것을 막기 위해 사용



실습 15 : 외부메모리 I/F에 SRAM붙이기

□ 구동 프로그램 : 소스 분석

❖ Lcdconf.h

```
#ifdef LCD_PORT_INTERFACE
    #ifndef LCD_CTRL_PORT
        #define LCD_CTRL_PORT    PORTF
        #define LCD_CTRL_DDR    DDRF
        #define LCD_CTRL_RS      0
        #define LCD_CTRL_RW      1
        #define LCD_CTRL_E       2
    #endif
    #ifndef LCD_DATA_POUT
        #define LCD_DATA_POUT    PORTD
        #define LCD_DATA_PIN     PIND
        #define LCD_DATA_DDR     DDRD
    #endif
#endif
#endif
```



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : 외부메모리 I/F에 SRAM붙이기

□ 구동 프로그램 : 소스 분석

❖ EXT_SRAM.c

```
1) #include<avr/io.h>
#include<util/delay.h>
#include"lcd.h"

#define read_XRAM(address) (*(volatile unsigned char *)(0x1100+address))
#define write_XRAM(address,value) (*(volatile unsigned char
*)(0x1100+address))=value)

void putch(unsigned char data){
while((UCSR0A & 0x20) == 0);
UDR0 = data;
UCSR0A |= 0x20;          /* UCSR0A 레지스터 중에서 5번째 비트(UDREN) 를
                           세트하여 UDRE0를 클리어 함. */
}
```



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 15 : 외부메모리 I/F에 SRAM붙이기

```
1) unsigned char getch(){
    unsigned char data;
    while((UCSR0A & 0x80) == 0);
    data = UDR0;
    UCSR0A |= 0x80;      /* UCSR0A 레지스터 중에서 7번째 비트를 세트하여
                           RXC0를 클리어 함. */

    return data;
}

int main(){
    unsigned char i=0,j=0;
    unsigned char line1,line2;
    unsigned char Tmp=0;
    unsigned char text[]="\n\r*****"
    "\n\r EXT-SRAM TEST Program ! "
    "\n\r TextLCD에 출력할 데이터를 입력 하세요\n\r"
    " (종료 : Enter키, 32바이트 이내)\n\r"
    "\n\r *****\n\r >> ";
```



실습 15 : 외부메모리 I/F에 SRAM붙이기

2)	<pre>MCUCR = 0x80; // 외부 메모리 enable XMCRA = 0x40; // SRL(2:0)=100, lower sector = 0x1100 - 0x7fff, // Upper sector = 0x8000 - 0xffff, SRW11~00=0000, 대기사이클 0 XMCRB = 0x00; //BUS-keeper 사용X, XMM2~0=000, 포트 C 전체 할당</pre>
3)	<pre>DDRE = 0xfe; // Rx(입력 0), Tx (출력, 1) UCSR0A = 0x00; UCSR0B = 0x18; // Rx, Tx enable UCSR0C = 0x06; // 비동기 방식, No Parity bit, 1 Stop bit UBRR0H = 0x00; UBRR0L = 0x07; //14.7456 MHz -> 115200 bps /* UBRRnH/L레지스터는 16비트 중에서 12비트만 사용하여 USARTn모듈의 송수신 속도를 설정 */</pre>
4)	<pre>lcdInit(); //TextLCD 초기화</pre>



실습 15 : 외부메모리 I/F에 SRAM붙이기

```
5) while(text[j]!='\0')
    putchar(text[j++]);           //text[j]를 증가시켜 가면서 출력
while(1){
    Tmp = getch();
    putchar(Tmp);                 //Tmp를 UART로 전송
    if((Tmp == '\r' || i > 32) {
        break;
    }
    write_XRAM(0x0100+i,Tmp);
    _delay_ms(10);                //10ms단위의 딜레이 간격
    i++;
}
```



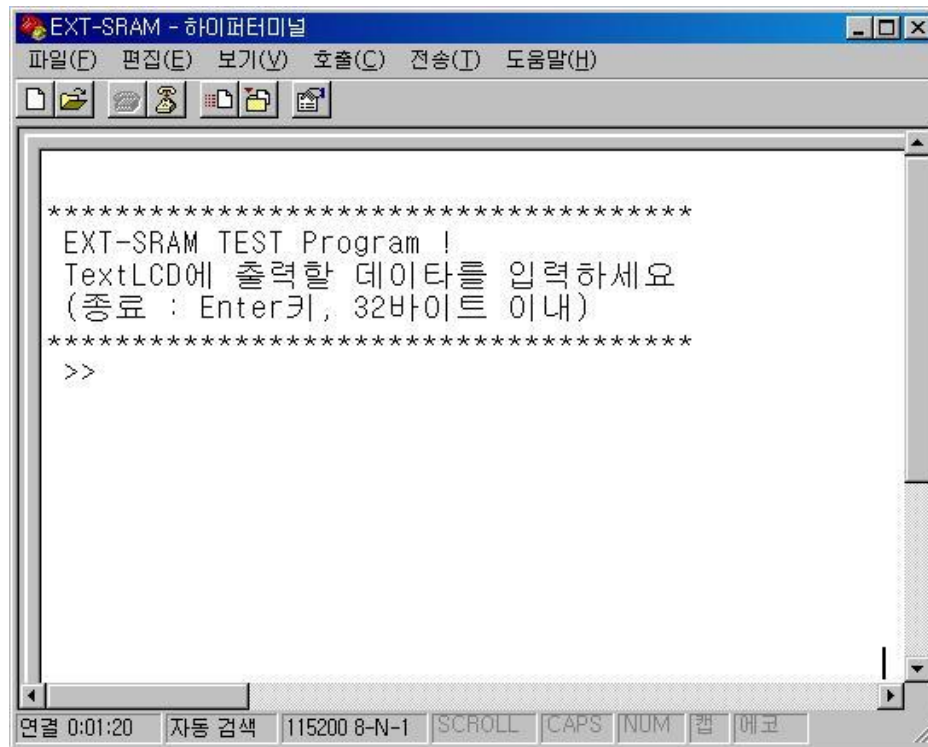
실습 15 : 외부메모리 I/F에 SRAM붙이기

6)	<pre>if(i<=16){ line1 = i; line2 = 0; } else{ line1 = 16; line2 = i-17; } _delay_ms(100); //100ms단위의 딜레이 간격</pre>
7)	<pre>for(j=0;j<line1;j++){ //읽어 들인 문자열 출력 lcdDataWrite(read_XRAM(0x0100+j)); _delay_ms(10); //10ms단위의 딜레이 간격 } lcdGotoXY(0,1); //현재 커서 위치를 두번째 열로 이동 for(j=0;j<line2;j++){ //읽어 들인 문자열 출력 lcdDataWrite(read_XRAM(0x0100+j+line1)); _delay_ms(10); //10ms단위의 딜레이 간격 } return 0; }</pre>



실습 15 : 외부메모리 I/F에 SRAM붙이기

□실행 결과



```
*****
EXT-SRAM TEST Program !
TextLCD에 출력할 데이터를 입력하세요
(종료 : Enter키, 32바이트 이내)
*****
>>
```

