

## 제 3 장 AVR ATmega128 마이크로컨트롤러

마이크로프로세서는 2.6절에서 언급한 바와 같이 메모리와 입출력장치를 갖추어야 컴퓨터의 역할을 할 수 있다. 이를 제어전용으로 사용할 때는 대용량의 메모리가 필요하지 않으며, 제어대상에 따라 필요한 입출력장치는 정해진다. 따라서 이들을 하나의 칩에 집적하는 것이 가능하다. 마이크로컴퓨터의 3대 구성요소를 하나의 칩에 집적하여 칩 하나만으로 컴퓨터의 기능을 하도록 함으로써 별도의 부가장치 없이 제어장치를 구성할 수 있다. 이런 종류의 칩들은 주로 제어전용으로 사용되므로 마이크로컴퓨터 대신 마이크로컨트롤러라 부른다. 그러나 본 교재에서는 편의상 마이크로컨트롤러와 마이크로프로세서를 혼용해서 사용하기로 한다.

어떤 마이크로프로세서든 그의 특징을 알려면 제조사의 관련 매뉴얼을 보는 것이 가장 좋다. 그러나 내용이 방대하고 마이크로프로세서를 입문하는 사람에게는 이해하기 어려운 부분이 많다. 여기서는 마이크로프로세서를 처음 접하는 사람을 위해 앞으로의 내용에 필요한 부분만을 발췌하여 설명하고자 한다. 이 과정을 통하여 마이크로프로세서에 대해 어느 정도의 지식을 습득한 후 제조사 매뉴얼과 관련 응용노트(Application Note)를 보면 많은 부분을 이해할 수 있으리라 믿는다.

### 3.1 ATmega128의 특징

그림 3.1은 ATmega128제조사인 Atmel사에서 제공하는 ATmega128 마이크로컨트롤러 매뉴얼의 첫 페이지로서 ATmega128의 특징을 기술한 것이다. 상당히 많은 특징을 기술하고 있다. 이 중에서 가장 보편적으로 필요한 사항에 대해서만 설명을 하고자 한다.

그림 3.1에서 마이크로컨트롤러를 선택할 때 가장 먼저 보는 사양은 다음 네 가지이다.

- (1) 마이크로프로세서의 비트수
- (2) 프로세서의 속도
- (3) 메모리 용량
- (4) 제공하는 입출력 장치

## Features

- High-performance, Low-power **AVR<sup>®</sup> 8-bit Microcontroller** → 비트수
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz → 속도
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 128K Bytes of In-System Reprogrammable Flash  
Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
  - 4K Bytes EEPROM  
Endurance: 100,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
 → 메모리
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
  - Output Compare Modulator
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines → 입출력 장치
  - 64-lead TQFP and 64-pad MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega128L
  - 4.5 - 5.5V for ATmega128
- Speed Grades
  - 0 - 8 MHz for ATmega128L
  - 0 - 16 MHz for ATmega128



**8-bit AVR<sup>®</sup>**  
**Microcontroller**  
**with 128K Bytes**  
**In-System**  
**Programmable**  
**Flash**

**ATmega128**  
**ATmega128L**

Rev. 2467M-AVR-11/04



그림 3.1 AVR ATmega128의 특징

(1) 마이크로프로세서의 비트 수 : AVR<sup>®</sup> 8bit Microcontroller

AVR<sup>®</sup> : AVR은 Atmel사에서 등록한 이름이며 AVR 시리즈를 나타낸다. 시리즈에는 여러 종류의 마이크로컨트롤러가 있다.

8bit Microcontroller : 8bit 마이크로프로세서를 내장하고 있음

(2) 프로세서의 속도

16MHz 클럭을 사용할 때 16MIPS의 연산속도를 갖는다. MIPS는 Million Instructions per Second의 약자로서 16MIPS이면 초당 1600만 개의 명령어를 수행하는 속도이다.

(3) 메모리

프로그램 메모리와 데이터 메모리가 별도로 구성되어있는 Harvard Architecture를 채택하고 있다. 각 메모리는 다음과 같이 구성되어있다.

- 128 Kbyte In-system-Reprogrammable Flash Memory  
: In-system Programming방법으로 10000번 읽기/지우기가 가능
- 4 Kbyte EEPROM  
: 100000번 읽기/지우기가 가능
- 4 Kbyte 내부 SRAM
- 64Kbyte까지 외부메모리 장착 가능
- In-system Programming을 위한 SPI 인터페이스 제공

(4) 입출력장치

제공하고 있는 입출력 장치는 다음과 같다. 각 입출력 장치는 이후 실습을 통하여 설명할 것이다.

- 독립적인 프리스케일러와 비교모드를 갖는 8비트 타이머/카운터 2개
- 독립적인 프리스케일러, 비교모드와 캡처모드를 갖는 확장 16비트 타이머/카운터 2개
- 독립적인 오실레이터를 갖는 실시간 카운터
- 2개의 8비트 PWM 채널
- 2비트~16비트로 프로그램 가능한 분해능을 갖는 6개의 PWM 채널

- Output Compare Modulator
- 8채널의 10비트 A/D 컨버터
- Byte-oriented Two Wire Serial Interface
- 2개의 프로그램 가능한 직렬 USARTs
- Master/Slave SPI 직렬인터페이스
- 내장 오실레이터를 갖는 프로그램 가능한 와치독 타이머
- 내장 아날로그 비교기
- 53개의 프로그램 가능한 입출력 선

### 3.2 핀의 기능

그림 3.2는 ATmega128의 핀 형상을 보여준다.

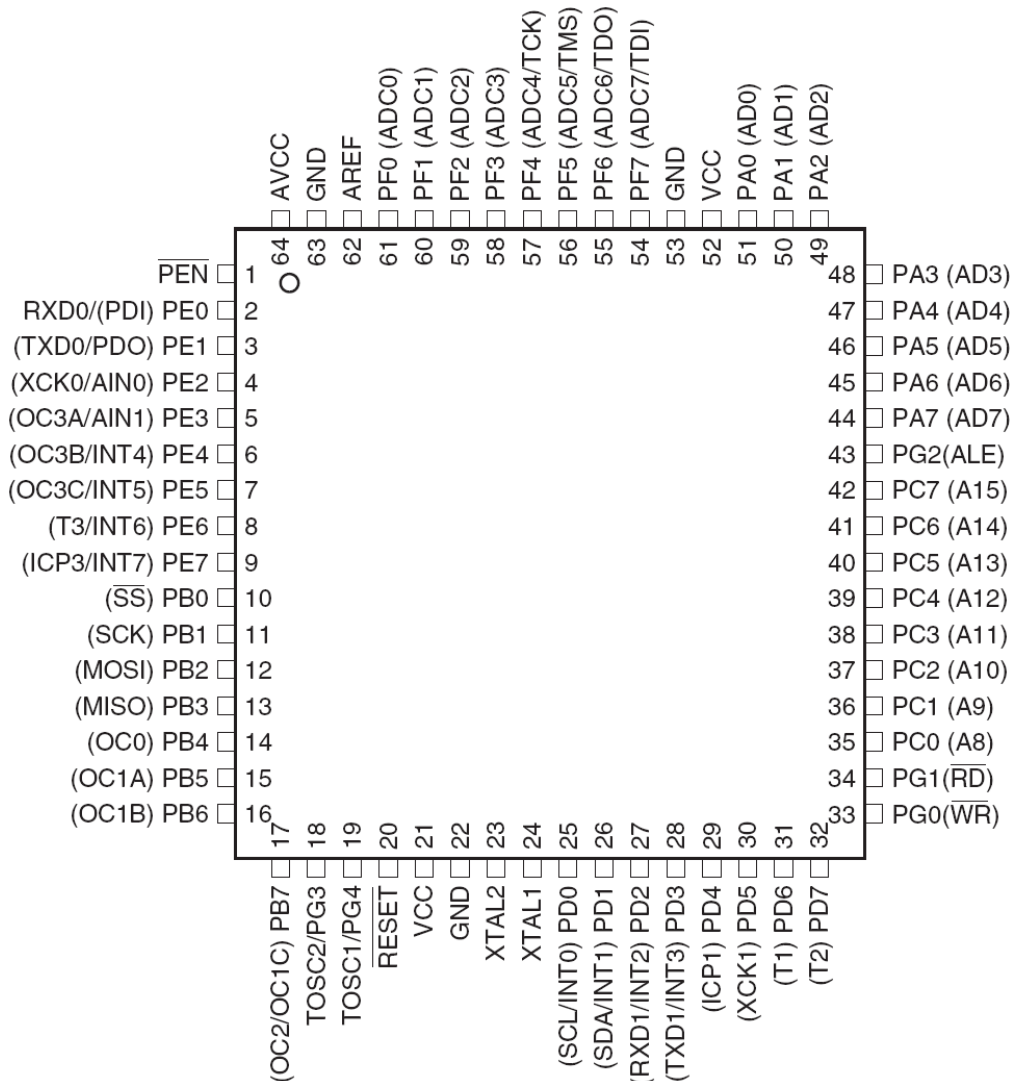


그림 3.2 ATmega128 핀 형상

각 핀의 기능은 다음과 같다.

Vcc	전원 핀
GND	접지
PORT A(PA.7..PA.0)	내부 풀업이 있는 8비트 양방향 I/O 포트이다. 그림 3.2의 특별한 기능도 수행한다.
PORT B(PB.7..PB.0)	내부 풀업이 있는 8비트 양방향 I/O 포트이다. 그림 3.2의 특별한 기능도 수행한다.
PORT C(PC.7..PC.0)	내부 풀업이 있는 8비트 양방향 I/O 포트이다. 그림 3.2의 특별한 기능도 수행한다.
PORT D(PD.7..PD.0)	내부 풀업이 있는 8비트 양방향 I/O 포트이다. 그림 3.2의 특별한 기능도 수행한다.
PORT E(PE.7..PE.0)	내부 풀업이 있는 8비트 양방향 I/O 포트이다. 그림 3.2의 특별한 기능도 수행한다.
PORT F(PF.7..PF.0)	내부 풀업이 있는 8비트 양방향 I/O 포트이다. A/D컨버터의 입력 핀으로도 동작한다.
PORT G(PG.4..PG.0)	내부 풀업이 있는 5비트 양방향 I/O 포트이다.
/RESET	리셋 입력이다. 최소 펄스 폭 보다 긴 LOW 레벨 신호를 인가하면 프로세서가 리셋 된다.
XTAL1	반전 오실레이터 증폭기에 대한 입력이며 내부 클록 동작 회로에 대한 입력이다.
XTAL2	반전 오실레이터 증폭기의 출력이다.
AVCC	A/D 컨버터의 전원이다. A/D 컨버터를 사용하지 않을 때에도 Vcc에 연결하여야 하고, A/D 컨버 터를 사용할 때는 저주파통과필터를 거쳐 Vcc와 연결되어야 한다.
AREF	A/D 컨버터의 기준전압이다.
PEN	PEN은 SPI 직렬 프로그래밍 모드에서 프로그래밍 을 가능하게 하는 핀이다. 내부적으로 풀업되어 있다.


 PORT A ~ PORT G가 공유해서 가지는 특별한 기능은 기능을 사용할  
때 설명할 것이다.

그림 3.2에서  $\overline{RESET}$  과 같이 핀기호 위에 ‘-’가 표시되었거나 /RESET과 같이 핀 기호 앞에 ‘/’를 붙이면 이 신호는 LOW일 때가 활성화(Active)된다는 뜻이다. 즉 LOW신호가 가해지면 CPU에 리셋이 걸린다. 유사하게 그림 3.2의 핀 34는  $\overline{RD}$  는 이 신호가 LOW일 때 CPU가 데이터를 읽어 들인다.

### 3.3 레지스터

레지스터는 메모리처럼 임시 또는 중간 결과를 저장하는 데 사용된다. 하지만 메모리와는 별도의 구성이 된다. 대부분의 명령어에서 레지스터를 사용하는 것이 메모리를 사용하는 것보다 실행속도 빠르고 명령어 크기가 작다. 따라서 레지스터를 데이터를 저장하여 사용하는 것이 가장 우수하지만 개수가 매우 한정되어 있어 데이터를 임시로 저장하는 데 사용된다.

일반적으로 레지스터는 C-언어에서 사용자가 직접 제어할 수는 없으나 ATmega128는 레지스터에 주소를 부여하여 사용자가 C-프로그램으로 레지스터를 제어할 수 있도록 하였다. 주소를 부여한 레지스터를 Memory Mapped Register(MMR)이라 한다.

여기서는 ATmega128이 가지고 있는 레지스터 중 C-프로그램을 이해하는 데 필요한 몇 개만 설명하도록 한다.

#### 상태 레지스터(Status Register) : SREG

비트	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
읽기/쓰기	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
초기 값	0	0	0	0	0	0	0	0	

상태 레지스터는 읽기/쓰기가 가능한 레지스터로 연산 결과에 대한 상태를 나타내고 있다. C-언어 프로그램에서 비트7만 관련이 있으므로 이 비트만 설명한다.

- 비트 7 - I : Global Interrupt Enable(전역 인터럽트 가능)

CPU 인터럽트를 전역으로 제어하는 비트이다. 이 비트를 1로 세트하여야 인터럽트를 사용할 수 있다. 이 비트가 0이면 모든 인터럽트가 불가능이 된다.

### 범용 레지스터 파일(General Purpose Register File) : R0 ~ R31

그림 3.3과 같이 32개의 1바이트 범용레지스터를 가지고 있다. 여러 레지스터가 순서대로 쌓여 있어서 레지스터 파일이라 부른다.

		7	0	주소		
범용 레지스터		R0		00H		
		R1		01H		
		R2		02H		
		...				
		R13		0DH		
		R14		0EH		
		R15		0FH		
		R16		10H		
		R17		11H		
		...				
		R26		1AH	X-레지스터의	하위바이트
		R27		1BH	X-레지스터의	상위바이트
		R28		1CH	Y-레지스터의	하위바이트
		R29		1DH	Y-레지스터의	상위바이트
		R30		1EH	Z-레지스터의	하위바이트
		R31		1FH	Z-레지스터의	상위바이트

그림 3.3 범용레지스터 파일

앞에서 언급한 바와 같이 연산에 있어서 레지스터를 사용하는 것이 메모리를 사용하는 것보다 효율적이다. C-언어에서는 `register` 키워드를 사용하여 레지스터 사용을 지정할 수 있다. 특히 R26~R31은 주소지정에 사용되며 다른 이름으로 X, Y, Z 레지스터라 불린다.

원래 레지스터는 주소를 가지고 있지 않으나 그림 3.3과 같이 각 레지스터에 SRAM영역의 주소를 지정하였다. 비록 SRAM영역의 주소를 레지스터가 차지하고 있지만 물리적으로 SRAM과는 별도로 구성되어 있다. 단지 주소를 부여함으로써 어셈블리 프로그램의 유연성을 높인다.

### 3.4 메모리

마이크로컨트롤러를 프로그램 하는 데 있어서 메모리의 구조는 매우 중요하다. 일반적으로 마이크로컨트롤러는 매우 제한적인 데이터 메모리를 가지고 있으므로 이 한계 내에서 프로그램이 가능하도록 하여야 할 뿐 아니라 사용하는 메모리의 특징에 따라 수행속도가 크게 차이가 나므로 사용자가 스스로 메모리를 관리할 수 있어야 한다.

ATmega128은 프로그램메모리와 데이터메모리가 별도로 구성되는 Harvard Architecture를 채택하고 있다.

#### 3.4.1 In-system Reprogrammable FLASH 프로그램 메모리

마이크로컨트롤러의 주 용도는 제어 전용 내장형(Embedded) 시스템이므로 프로그램은 전원을 꺼도 지워지지 않는 비휘발성 메모리에 저장된다. ATmega128은 비휘발성메모리로 FLASH 메모리를 사용하고 있으며 프로그램메모리는 모두 FLASH 메모리로 구성되어 있다. 특히 이 메모리는 ISP(In-System-Programming)기술을 사용하여 PC에서 직접 프로그램을 다운로드할 수 있도록 되어있다.

일반적으로 메모리의 기본단위는 바이트이며 주소 역시 각 바이트 당 부여가 된다. 그러나 ATmega128의 프로그램메모리의 기본단위는 워드(2바이트)이다. 주소역시 워드단위로 부여된다. 이는 ATmega128의 모든 명령어가 2바이트의 배수로 되어 있어 효율성을 높이기 위해서이다. 그림 3.4는 프로그램 메모리의 구조를 보여준다. 프로그램 메모리의 크기는  $64Kword = 2 \times 64Kbyte = 128Kbyte$ 이다. 그림 3.1을 보면 ATmega128의 명칭에 128Kbyte FLASH 메모리를 가졌음을 표시한다.

FLASH메모리는 ISP를 사용하여 다운로드된 후에는 프로그램에서는 내용을 변경할 수 없고 내용을 읽기만 가능하다. 이는 10000번의 쓰기 즉 다운로드가 가능하다.

☞ 휘발성 메모리(Volatile Memory) : 이름 그대로 전원을 끄면 내용이 사라지는 메모리를 나타낸다. RAM을 나타낸다.

☞ 비휘발성 메모리(Nonvolatile Memory) : ROM 종류와 Flash 메모리와 같이 전원을 끄더라도 내용이 사라지지 않는 메모리를 나타낸다.



### 3.4.2 데이터 메모리

데이터 메모리는 쓰기/읽기가 가능한 메모리이다. 그림 3.5는 데이터 메모리 맵을 보여준다. 메모리는 바이트단위로 주소가 지정된다.

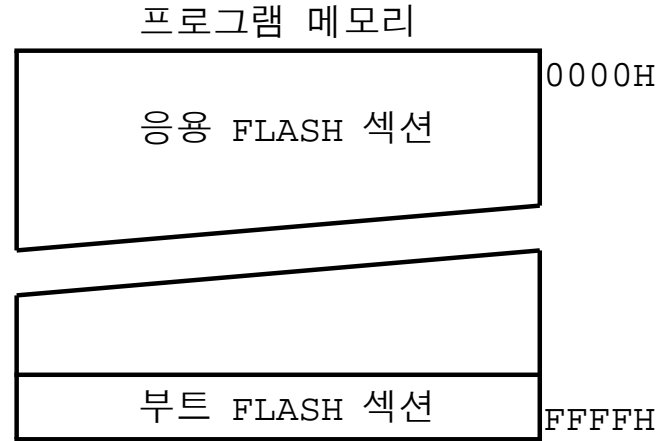


그림 3.4 FLASH 프로그램 메모리

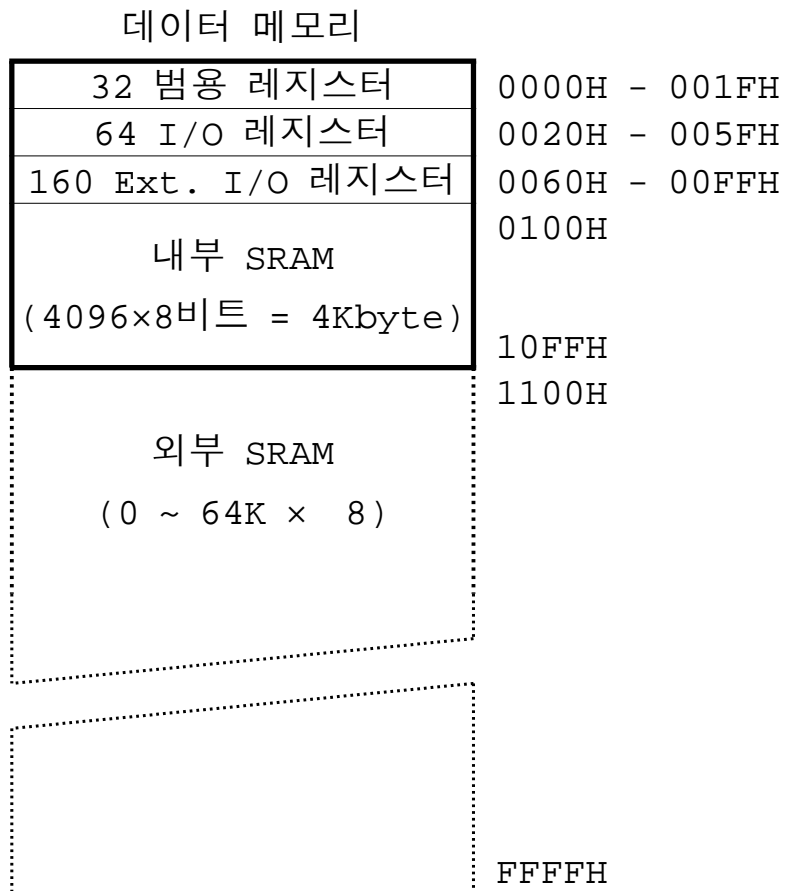


그림 3.5 데이터 메모리 맵

- 레지스터 파일 : 주소 0000H~001FH는 그림 3.3의 레지스터 파일이 사용하고 있다. 실제로 이 부분은 SRAM이 아니지만 메모리의 주소영역을 사용하고 있다.
- 내부 SRAM : 주소 0100H~10FFH에 4K 바이트의 SRAM이 내부에 장착되어 있다.
- 외부 SRAM : 내부 SRAM의 용량이 부족할 경우 외부에 주소 1100H부터 FFFFH까지 약 60K바이트의 SRAM을 장착할 수 있다. 외부 SRAM을 사용하는 것이 내부 SRAM을 사용하는 것보다 실행속도는 느리다.
- I/O 레지스터 : I/O 레지스터는 SRAM과 달리 ATmega128에서 제공하고 있는 입출력 장치를 직접 제어하는 데 사용된다. 이들 역시 SRAM의 주소영역에 주소가 할당되어 있다. 이는 주소 0020H ~ 005FH를 사용하는 I/O 레지스터 60개와 주소 0060H ~ 00FFH를 사용하는 160개의 확장 I/O 레지스터로 구성되어 있다. 표 3.1은 ATmega128 마이크로컨트롤러 매뉴얼 중 각 레지스터의 주소 및 각 비트의 기능을 나타내는 표이다. I/O 레지스터의 주소는 두 가지로 표시되어 있는데 괄호 속의 주소는 레지스터의 메모리 주소를 나타내고, 밖의 주소는 입출력 주소를 나타낸다. 메모리 오퍼레이션을 사용해서 I/O 레지스터를 조작할 때는 메모리 주소를 사용하고 입출력 오퍼레이션을 사용해서 조작할 때는 입출력 주소를 사용한다. 각 레지스터에 대한 설명은 실습을 통하여 필요할 때 설명을 할 것이다.

### 3.4.3 EEPROM 메모리

ATmega128은 프로그램 메모리와 데이터 메모리 이외에 4K바이트의 EEPROM 메모리를 제공한다. 이는 프로그램 메모리 및 데이터 메모리와는 별도의 주소 영역을 사용한다. 따라서 이 부분은 I/O 레지스터를 사용하여 주소를 지정하는 방법으로 접근이 가능하다. 매우 접근하기가 번거로운 부분이지만 제어전용시스템을 구축할 때 유용하게 사용할 수 있다.

제공하는 EEPROM은 비휘발성 메모리이지만 FLASH 메모리와는 달리 프로그램에서 데이터를 쓰기/읽기를 할 수 있다. 프로그램 수행 중에는 새로운 데이터들이 만들어지는 데 이들 데이터가 전원이 꺼지더라도 보존을 해야 하는 데이터이면 휘발성 메모리인 SRAM에 보관을 할 수 없고

비휘발성 메모리인 EEPROM에 저장하여야 한다. PC에서는 이런 데이터를 하드디스크에 저장하면 되지만 하드디스크와 같은 장치가 없는 제어전용시스템에서는 매우 유용한 기능이다.

EEPROM의 내용을 읽을 때는 회수에 제한이 없으나 쓰기는 100000번까지 할 수 있다. 이 정도 회수이면 응용에 따라 제한이 있을 것이나 갑작스런 전원공급의 중단에 대해 EEPROM을 사용한다면 시스템의 수명보다 더 길게 사용할 수 있을 것이다.

## Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(\$FF)	Reserved	–	–	–	–	–	–	–	–	
..	Reserved	–	–	–	–	–	–	–	–	
(\$9E)	Reserved	–	–	–	–	–	–	–	–	
(\$9D)	UCSR1C	–	UMSEL1	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1	191
(\$9C)	UDR1	USART1 I/O Data Register								189
(\$9B)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	UPE1	U2X1	MPCM1	189
(\$9A)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81	190
(\$99)	UBRR1L	USART1 Baud Rate Register Low								193
(\$98)	UBRR1H	–	–	–	–	USART1 Baud Rate Register High				193
(\$97)	Reserved	–	–	–	–	–	–	–	–	
(\$96)	Reserved	–	–	–	–	–	–	–	–	
(\$95)	UCSR0C	–	UMSEL0	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0	191
(\$94)	Reserved	–	–	–	–	–	–	–	–	
(\$93)	Reserved	–	–	–	–	–	–	–	–	
(\$92)	Reserved	–	–	–	–	–	–	–	–	
(\$91)	Reserved	–	–	–	–	–	–	–	–	
(\$90)	UBRR0H	–	–	–	–	USART0 Baud Rate Register High				193
(\$8F)	Reserved	–	–	–	–	–	–	–	–	
(\$8E)	Reserved	–	–	–	–	–	–	–	–	
(\$8D)	Reserved	–	–	–	–	–	–	–	–	
(\$8C)	TCCR3C	FOC3A	FOC3B	FOC3C	–	–	–	–	–	135
(\$8B)	TCCR3A	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	131
(\$8A)	TCCR3B	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	134
(\$89)	TCNT3H	Timer/Counter3 – Counter Register High Byte								136
(\$88)	TCNT3L	Timer/Counter3 – Counter Register Low Byte								136
(\$87)	OCR3AH	Timer/Counter3 – Output Compare Register A High Byte								136
(\$86)	OCR3AL	Timer/Counter3 – Output Compare Register A Low Byte								136
(\$85)	OCR3BH	Timer/Counter3 – Output Compare Register B High Byte								137
(\$84)	OCR3BL	Timer/Counter3 – Output Compare Register B Low Byte								137
(\$83)	OCR3CH	Timer/Counter3 – Output Compare Register C High Byte								137
(\$82)	OCR3CL	Timer/Counter3 – Output Compare Register C Low Byte								137
(\$81)	ICR3H	Timer/Counter3 – Input Capture Register High Byte								137
(\$80)	ICR3L	Timer/Counter3 – Input Capture Register Low Byte								137
(\$7F)	Reserved	–	–	–	–	–	–	–	–	
(\$7E)	Reserved	–	–	–	–	–	–	–	–	
(\$7D)	ETIMSK	–	–	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C	138
(\$7C)	ETIFR	–	–	ICF3	OCF3A	OCF3B	TOV3	OCF3C	OCF1C	139
(\$7B)	Reserved	–	–	–	–	–	–	–	–	
(\$7A)	TCCR1C	FOC1A	FOC1B	FOC1C	–	–	–	–	–	135
(\$79)	OCR1CH	Timer/Counter1 – Output Compare Register C High Byte								136
(\$78)	OCR1CL	Timer/Counter1 – Output Compare Register C Low Byte								136
(\$77)	Reserved	–	–	–	–	–	–	–	–	
(\$76)	Reserved	–	–	–	–	–	–	–	–	
(\$75)	Reserved	–	–	–	–	–	–	–	–	
(\$74)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	206
(\$73)	TWDR	Two-wire Serial Interface Data Register								208
(\$72)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	208
(\$71)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	207
(\$70)	TWBR	Two-wire Serial Interface Bit Rate Register								206
(\$6F)	OSCCAL	Oscillator Calibration Register								39
(\$6E)	Reserved	–	–	–	–	–	–	–	–	
(\$6D)	XMCRA	–	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11	–	29
(\$6C)	XMCRA	XMBK	–	–	–	–	XMM2	XMM1	XMM0	31
(\$6B)	Reserved	–	–	–	–	–	–	–	–	
(\$6A)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	87
(\$69)	Reserved	–	–	–	–	–	–	–	–	
(\$68)	SPMCSR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	279
(\$67)	Reserved	–	–	–	–	–	–	–	–	
(\$66)	Reserved	–	–	–	–	–	–	–	–	
(\$65)	PORTG	–	–	–	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	86
(\$64)	DDRG	–	–	–	DDG4	DDG3	DDG2	DDG1	DDG0	86
(\$63)	PING	–	–	–	PING4	PING3	PING2	PING1	PING0	86
(\$62)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	85

표 3.1 I/O 레지스터

## Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(\$61)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	86
(\$60)	Reserved	–	–	–	–	–	–	–	–	
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C	9
\$3E (\$5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	12
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
\$3C (\$5C)	XDIV	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	41
\$3B (\$5B)	RAMPZ	–	–	–	–	–	–	–	RAMPZ0	12
\$3A (\$5A)	EICRB	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	88
\$39 (\$59)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	89
\$38 (\$58)	EIFR	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	89
\$37 (\$57)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	106, 138, 158
\$36 (\$56)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	106, 139, 158
\$35 (\$55)	MCUCR	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	29, 42, 61
\$34 (\$54)	MCUCSR	JTD	–	–	JTRF	WDRF	BORF	EXTRF	PORF	51, 256
\$33 (\$53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	101
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bit)								103
\$31 (\$51)	OCR0	Timer/Counter0 Output Compare Register								103
\$30 (\$50)	ASSR	–	–	–	–	AS0	TCN0UB	OCR0UB	TCR0UB	104
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	131
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	134
\$2D (\$4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								136
\$2C (\$4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								136
\$2B (\$4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								136
\$2A (\$4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								136
\$29 (\$49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								136
\$28 (\$48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								136
\$27 (\$47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								137
\$26 (\$46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								137
\$25 (\$45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	156
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bit)								158
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register								158
\$22 (\$42)	OCDR	IDRD/OCDR7	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	253
\$21 (\$41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	53
\$20 (\$40)	SFIOR	TSM	–	–	–	ACME	PUD	PSR0	PSR321	70, 107, 143, 228
\$1F (\$3F)	EEARH	–	–	–	–	EEPROM Address Register High				19
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte								19
\$1D (\$3D)	EEDR	EEPROM Data Register								20
\$1C (\$3C)	EEDR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	20
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	84
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	84
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	84
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	84
\$17 (\$37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	84
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	84
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	84
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	84
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	85
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	85
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	85
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	85
\$0F (\$2F)	SPDR	SPI Data Register								168
\$0E (\$2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	168
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	166
\$0C (\$2C)	UDR0	USART0 I/O Data Register								189
\$0B (\$2B)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	189
\$0A (\$2A)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	190
\$09 (\$29)	UBRR0L	USART0 Baud Rate Register Low								193
\$08 (\$28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	228
\$07 (\$27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	244
\$06 (\$26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	245
\$05 (\$25)	ADCH	ADC Data Register High Byte								246
\$04 (\$24)	ADCL	ADC Data Register Low byte								246
\$03 (\$23)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	85
\$02 (\$22)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	85

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$01 (\$21)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	85
\$00 (\$20)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	86

표 3.1(계속) I/O 레지스터

## 제 4 장 ATmega128 보드

### 4.1 실습보드

[www.avrmall.com](http://www.avrmall.com)에서 구입할 수 있는 Avrmall ATmega128 v3.2 보드(그림 4.1)를 교재에서 사용한다. 보드의 특징은 다음과 같다.

- 사용 CPU : ATmega128-16AI
- 사용 클럭 : 16MHz
- 32Kbyte 외부 SRAM 장착 :  
내부 SRAM영역을 제외한 1000H ~ 7FFFH까지 사용
- 두 채널 RS232C 가능(Max232 내장)
- RTC-32(RealTime Clock), 32.768KHz 내장
- Memory-mapped I/O 방식으로 LCD 구동
- 명령 읽기/쓰기 주소 : 8000H  
데이터 읽기/쓰기 주소 : C000H
- Atmel ISP 포트, JTAG 포트 지원
- 동작 전원 : 4.5V ~ 5.5V

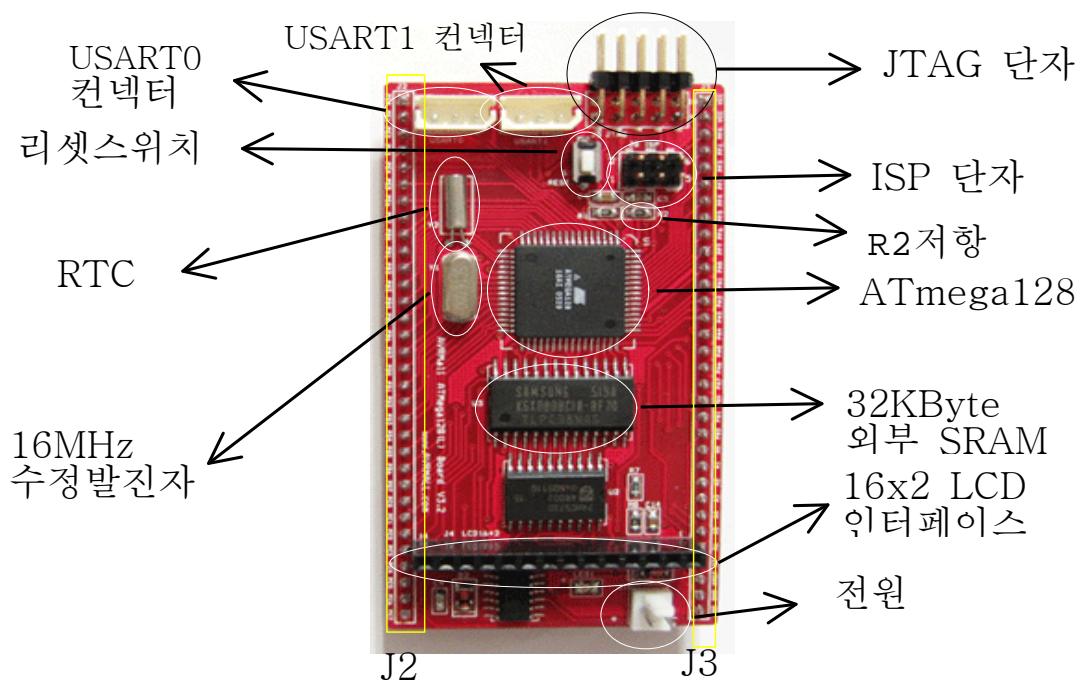


그림 4.1 실습보드

보드 양편의 32핀 단자 J2와 J3를 사용하여 외부 장치와 연결할 수 있다. 각 핀의 기능은 다음과 같다.

J2		J3	
핀 번호	기능	핀 번호	기능
1	PE0 (RXD0/PDI)	1	Vcc
2	PE1 (TXD0/PDO)	2	Vcc
3	PE2 (XCK0/AIN0)	3	PF0 (ADC0)
4	PE3 (OC3A/AIN1)	4	PF1 (ADC1)
5	PE4 (OC3B/INT4)	5	PF2 (ADC2)
6	PE5 (OC3C/INT5)	6	PF3 (ADC3)
7	PE6 (T3/INT6)	7	PF4 (ADC4/TCK)
8	PE7 (ICP3/INT7)	8	PF5 (ADC5/TMS)
9	PB0 (/SS)	9	PF6 (ADC6/TDO)
10	PB1 (SCK)	10	PF7 (ADC7/TDI)
11	PB2 (MOSI)	11	PA0 (AD0)
12	PB3 (MISO)	12	PA1 (AD1)
13	PB4 (OC0)	13	PA2 (AD2)
14	PB5 (OC1A)	14	PA3 (AD3)
15	PB6 (OC1B)	15	PA4 (AD4)
16	PB7 (OC2/OC1C)	16	PA5 (AD5)
17	PD0 (SCL/INT0)	17	PA6 (AD6)
18	PD1 (SDA/INT1)	18	PA7 (AD7)
19	PD2 (RXD1/INT2)	19	PG0 (ALE)
20	PD3 (TXD1/INT3)	20	PG1 (/RD)
21	PD4 (ICP1)	21	PG2 (/WR)
22	PD5 (XCK1)	22	A0
23	PD6 (T1)	23	A1
24	PD7 (T2)	24	A2
25	PC0 (A8)	25	A3
26	PC1 (A9)	26	A4
27	PC2 (A10)	27	A5
28	PC3 (A11)	28	A6
29	PC4 (A12)	29	A7
30	PC5 (A13)	30	AREF
31	PC6 (A14)	31	GND
32	PC7 (A15)	32	GND

표 4.1 단자 J3, J4의 핀 기능

그림 4.2는 보드의 메모리 맵을 나타낸다. 32Kbyte의 외부 SRAM이 장착되어있고 메모리 주소영역 8000H와 C000H에 LCD가 위치하고 있다. 외부 SRAM과 LCD를 사용하려면 프로그램 상에서 외부메모리를 사용하도록 설정하여야 한다.

※교재의 예제 프로그램은 16MHz 클록을 사용하는 것으로 가정하여 작성하였다. 따라서 8MHz 클록이 장착된 보드를 구입하면 예제 프로그램이 오동작을 할 수 있다.



그림 4.2 ATmega128 보드의 메모리 맵



## 4.2 AVR Programmer

FLASH 메모리에 프로그램을 다운로드하는 장치이다. 그림 4.3은 PC의 USB 포트를 이용하여 AVR 마이크로컨트롤러에 프로그램을 다운로드하는 USBISP v2.0이다. 이는 Atmel사 AVRISP와 호환되는 Programmer이다.

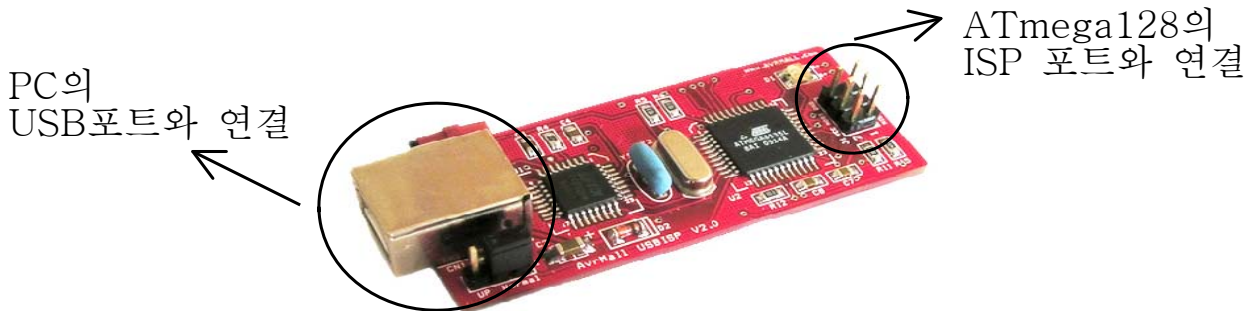


그림 4.3 USBISP v2.0 Programmer

## 4.3 ATmega128보드와 AVR Programmer의 연결

USB 케이블을 사용하여 USBISP를 연결한다. USBISP의 6핀 플랫케이블로 ATmega128 보드와 그림 4.4와 같이 연결한 후 보드의 전원을 켜면 USBISP의 LED에 불이 켜진다. ATmega128 보드에 전원을 공급하는 방법은 두 가지가 있다.

- (1) ATmega128 보드의 전원 단자에 외부전원을 직접 연결한다. 그리고 USBISP의 스위치 sw1을 EXT\_VCC위치로 세팅한다.
- (2) PC의 USB포트 전원을 사용하는 방법으로 USBISP의 스위치 sw1을 VBUS위치로 세팅한다. 프로그램 개발 중에는 프로그램을 다운로드하기 위해 USBISP를 사용하므로 매우 편리하다.

이 후 USBISP의 드라이버를 요구하면 USBISP 드라이버가 있는 CD를 삽입하고 다음 순서로 설치한다.

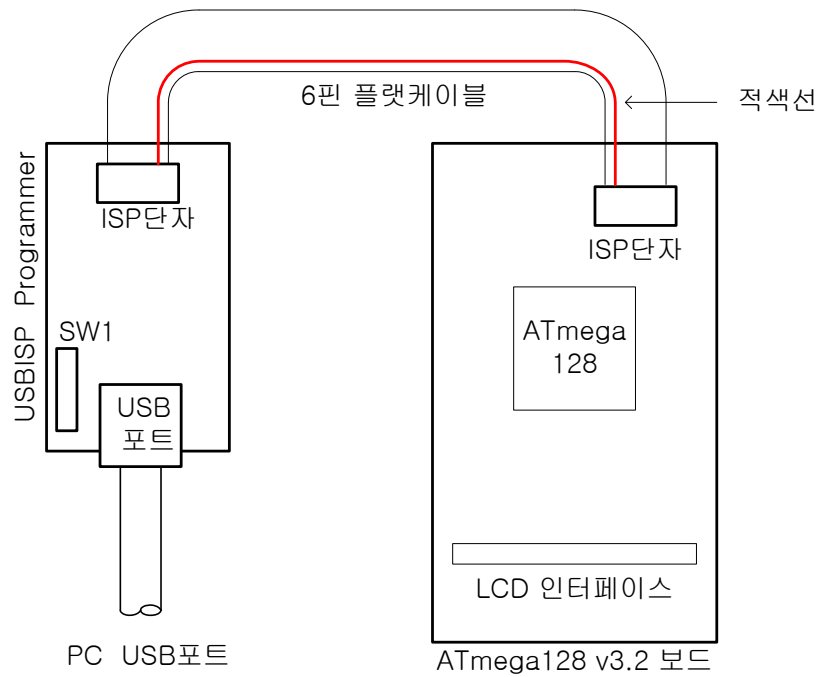


그림 4.4 ATmega128 보드와 USBISP의 연결

드라이버는 두 가지가 설치된다. USB를 직렬통신으로 변환시켜주는 USB High Speed Serial Converter와 직렬통신 포트인 USB Serial Port이다.

1. 새 하드웨어 발견(USB<->Serial)
2. 새 하드웨어 검색 마법사 시작(목록 또는 특정 위치에서 설치 선택)
3. 검색 안 함. 설치할 드라이버 직접 선택
4. 하드웨어 종류(모든 장치 표시 선택)
5. 설치할 하드웨어 장치 드라이버 선택(디스크 있음에서 설치 드라이버 선택 -> FTDIPOINT.INF 선택)
6. USB High Speed Serial Converter 선택
7. OS가 Windows-XP일 경우 호환성 테스트
8. 새 하드웨어 검색 마법사 완료(USB High Speed Serial Converter 마침)

1. 새 하드웨어 발견(USB Serial Port)
2. 새 하드웨어 검색 마법사 시작(목록 또는 특정 위치에서 설치 선택)

3. 검색 안 함. 설치할 드라이버 직접 선택
4. 하드웨어 종류(모든 장치 표시 선택)
5. 설치할 하드웨어 장치 드라이버 선택(디스크 있음에서 설치 드라이버 선택 -> FTDIPORT.INF 선택)
6. USB Serial Port 선택
7. OS가 Windows-XP일 경우 호환성 테스트
8. 새 하드웨어 검색 마법사 완료(USB Serial Port 마침)

드라이버 설치 이후 제어판->시스템->하드웨어->장치관리자->포트 (COM 및 LPT)를 선택하면 USB Serial Port가 나온다. 이 때 USB Serial Port에 할당된 COM포트를 확인한다.(주로 COM3) 이 COM포트를 사용하여 프로그램을 다운로드하므로 꼭 기억하기 바란다.

※ USBISP를 연결한 USB포트가 변경되면 드라이버를 다시 설치하여야 한다. 따라서 드라이버를 재설치하는 번거로움을 없애기 위해서는 드라이브를 인스톨할 때 연결하였던 USB포트를 계속 사용하는 것이 좋다.

<빈 페이지>