

# ATmega128 기초

마이크로프로세서

HRI연구실

김동한



Human-Robot Interaction  
Laboratory

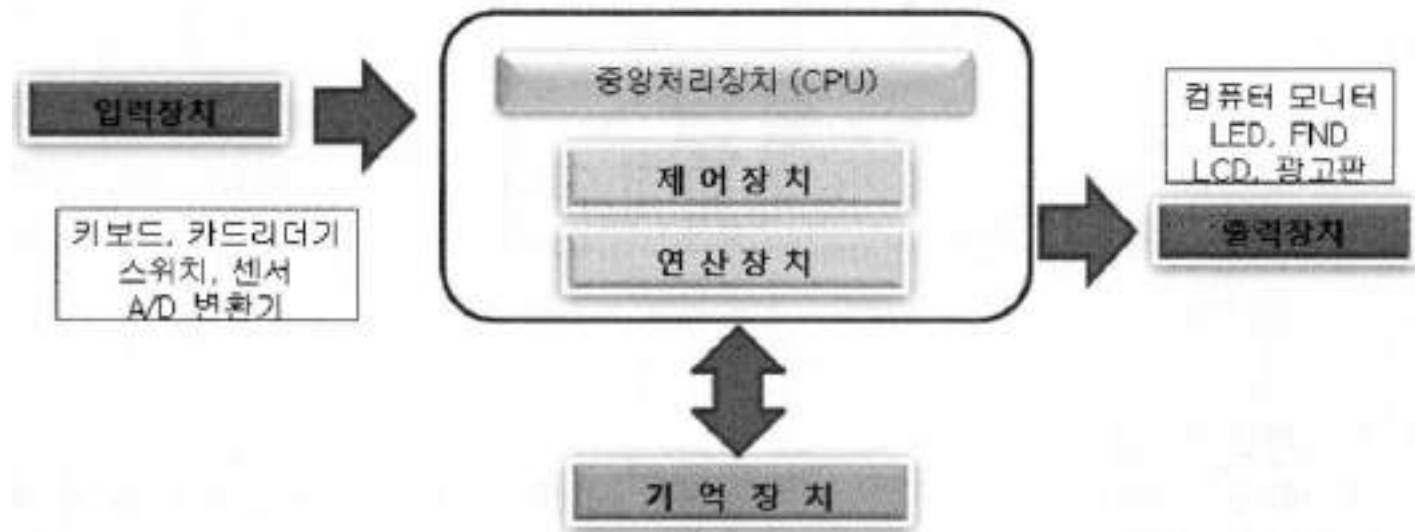


KYUNG HEE  
UNIVERSITY

## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

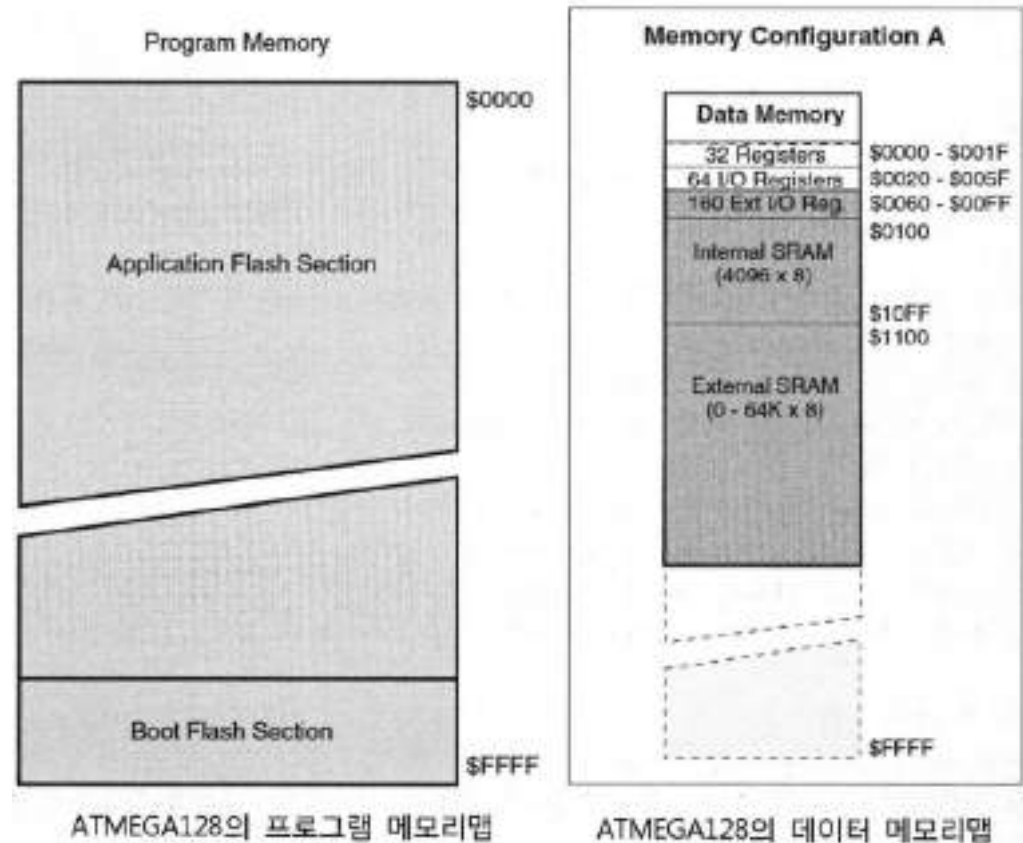
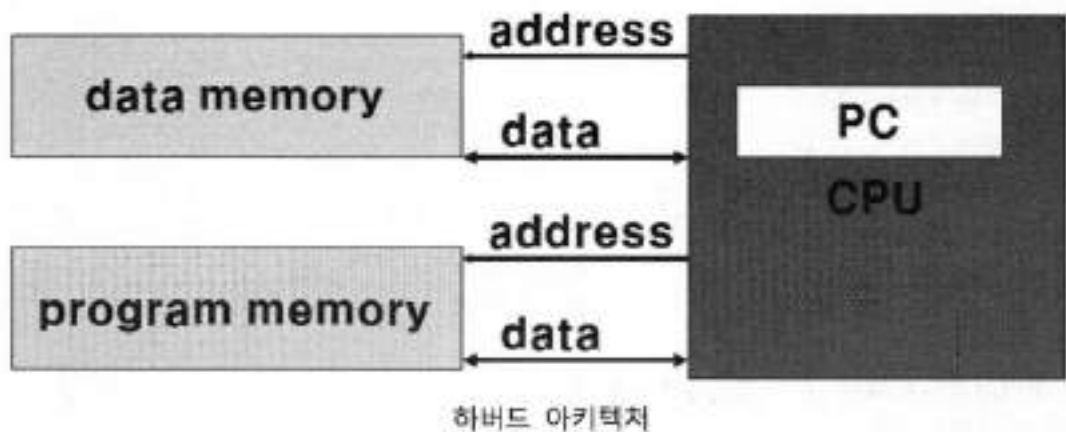
- 개인용 컴퓨터 : CPU, 하드디스크, 메인보드, 램, 그리고 비디오카드 등으로 구성되어 짐.  
작업이 끝나면 하드디스크나 메모리 스틱에 저장하거나 프린터를 이용하여 결과물을 출력하게 됨.
- 마이크로컨트롤러 : CPU, ROM, RAM과 전류 Fanout을 늘린 I/O확장용 GPIO제어기가 **한 칩**에 함께 구성되어 있음.  
주로 단순한 반복작업을 위한 제품에 장착되는 작은 보드 등을 통한 자동화 장치 등에 쓰임.
  - 1) 스위치, 센서 등의 입력장치 → 키보드의 역할  
- 제어입력신호를 발생하는 역할
  - 2) LED, LCD, Motor, Relay 등의 출력장치 → 모니터의 역할  
- 제어 신호를 이용하여 시스템을 제어하는 역할



## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

- 하버드 아키텍처 : 프로그램과 데이터가 서로 다른 메모리 영역을 차지하여, 각각의 어드레스, 데이터, 제어 버스가 존재함  
프로그램 명령과 데이터를 동시에 읽을 수 있으며, 명령 구조는 RISC(Reduced instruction set computer) 구조를 사용함.  
AVR도 이러한 하버드 아키텍처로 되어 있음.



## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

- 마이크로프로세서 vs. 마이크로컨트롤러 :

- ✓ 기능에 따른 분류: 의미 없음

MPU - Micro Processor Unit

MCU - Micro Controller Unit

DSP - Digital Signal Processor

→ 많은 업체에서 여러 종류의 칩을 각각 특별한 기능을 내장하고 여러 응용분야에서 사용하기 때문

- 데이터 버스의 크기 산술처리기 (Arithmetic Logic Unit) 및 레지스터들의 크기에 따른 분류:

- ✓ 8 비트 프로세서: 응용목적이 간단한 경우

유명한 제품군: 8051, AVR, PIC, SAM8, STM8, R8C

- ✓ 16비트 프로세서: 중간정도의 복잡도를 가지는 응용분야

유명한 제품군: RL78, ST16, MSP430

- ✓ 32 비트 프로세서 : 임베디드(내장형) 시스템 시장에서 주류

유명한 제품군: RX, RZ, V850, RH850, Superba ARM, Cortex-NR!M, PIC32

- ✓ 64 비트 프로세서 슈퍼컴퓨터용 워크스테이션용 서버용

유명한 제품군: Intel64, AMD64, PowerPC 970, SPARC V9, IA-64, MIPS64, AArch64

## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

- 마이크로 컨트롤러의 응용분야:

산업	: 모터 제어, 로봇 제어, 프로세스 제어 등
계측	: 의료용 계측기, 오실로스코프 등
가전제품	: 전자레인지, 가스오븐, 전자밥솥, 세탁기 등
군사	: 미사일 제어, Torpedo 제어, 우주선 유도 제어 등
통신	: 휴대폰, 모뎀, 유무선 전화기, 중계기 등
사무기기	: 복사기, 프린터, plotter, 하드디스크 구동장치 등
자동차	: 점화 타이밍 제어, 연료 분사 제어, 변속기 제어 등
생활	: 전자시계, 계산기, 게임기, 금전등록기, 온도조절기 등

- 마이크로 컨트롤러 - AVR 계열:

AVR은 ATMEL사가 개발 제조하고 있는 유사 RISC 구조의 저전력 CMOS 8-bit 마이크로 컨트롤러이다.

- ✓ 임베디드 환경에 적합한 저전력 소모의 집이며, 내부에 32개의 범용 레지스터 내장
- ✓ 유비쿼터스 및 센서 네트워크를 위한 소형 시스템에 적합
- ✓ 유사 RISC 아키텍처 (RISC like Architecture) - 유사 RISC 아키텍처라고 하는 것은 RISC 구조를 따르고는 있으나 명령어와 주소 지정 방식의 수가 RISC에 비해 많음

## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

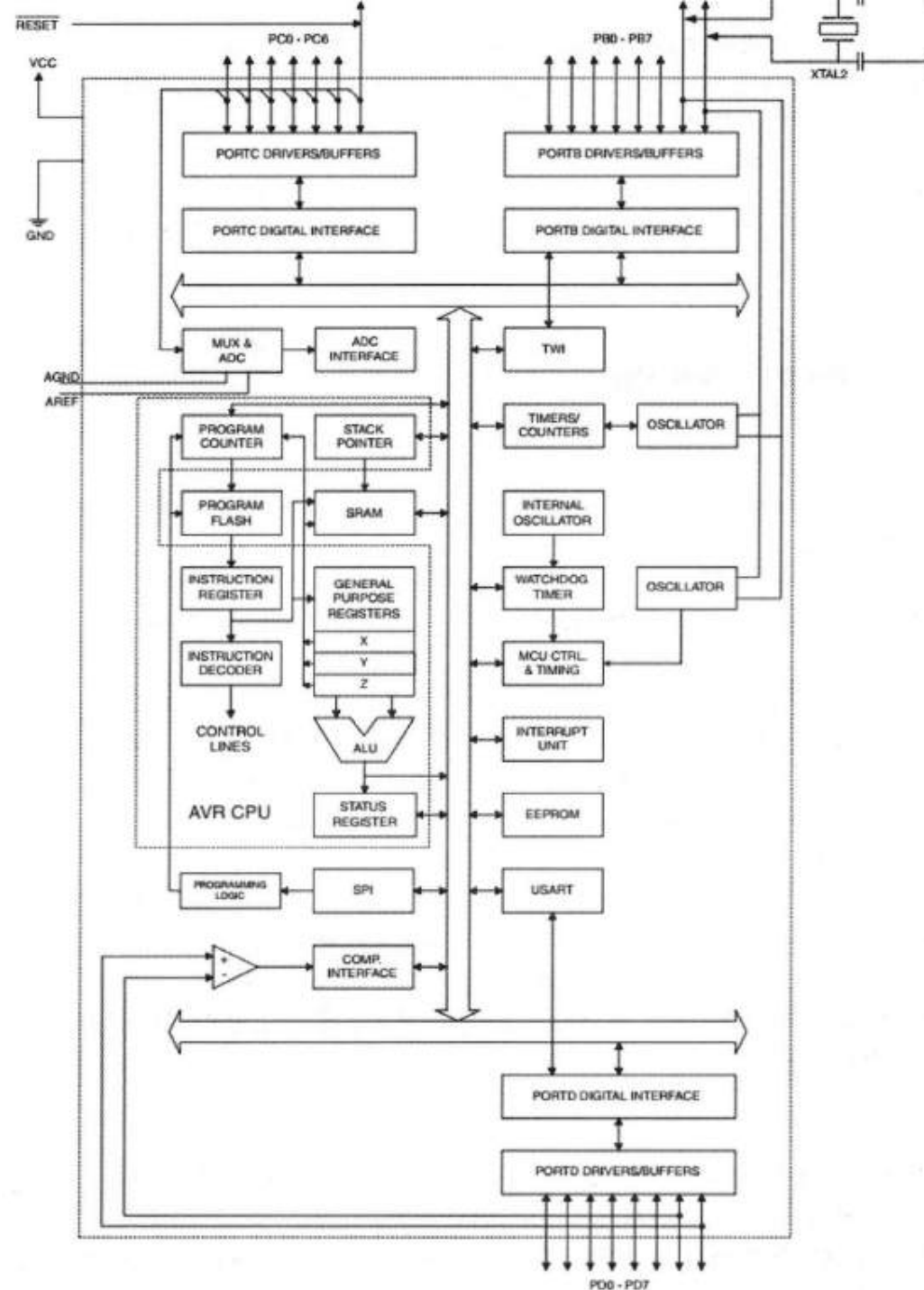
- 8비트 AVR의 종류 :
  - ✓ Tiny(ATtiny) : 소형시스템에 적합 외형핀 8 ~ 20개 메인 MCU보다 여러 회로를 하나의 소자로 구현한 보조 역할의 집으로 사용 (리모컨 등에 사용)
  - ✓ Mega (ATmega) : 대형이고 입/출력 핀의 수가 많은 시스템에 적합, 무선 전화기 및 프린터용 제어기 FAX 및 CD-ROM 제어기 | 통신장비 등에 응용
  - ✓ XMEGA(ATxmega) : 일반적인 목적. 용도는 Mega 시리즈와 같음
  - ✓ Automotive AVR : Automotive, 자동차와 같은 환경을 위한 환경에 강한 튼튼한 디자인 및 CAN통신
- 특징 :
  - ✓ 프로그램을 다운로드하여 저장 할 수 있는 2K ~128K 바이트의 플래시 메모리를 내장
  - ✓ 내부 EEPROM 용량 : 128에서 512 바이트까지 가능
  - ✓ 내부 오실레이터에 의해 동작하는 위치독 타이머와 전 이중방식의 다양한 외부 접속장치를 내장
  - ✓ 다양한 인터럽트 소스와 파워 절약모드 외부 메모리 추가가 가능
  - ✓ 내장 디바이스 : 타이머/카운터 직렬 통신 장치 아날로그 비교기, 외부 데이터 장치 인터페이스 입출력 포트 SPI(Serial Peripheral Interface), 아날로그 디지털 변환기 (ADC), TWI 등.



## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

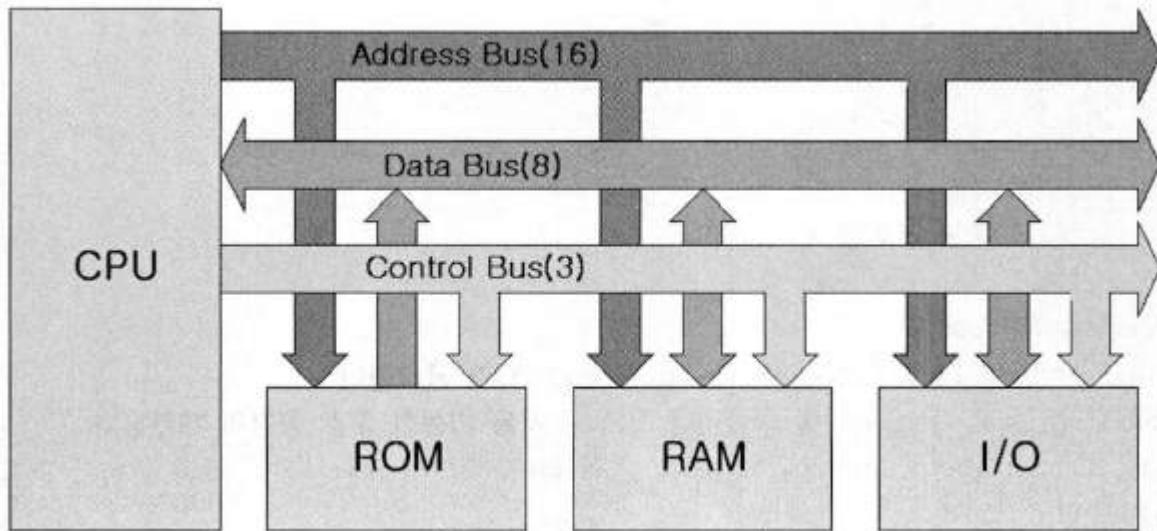
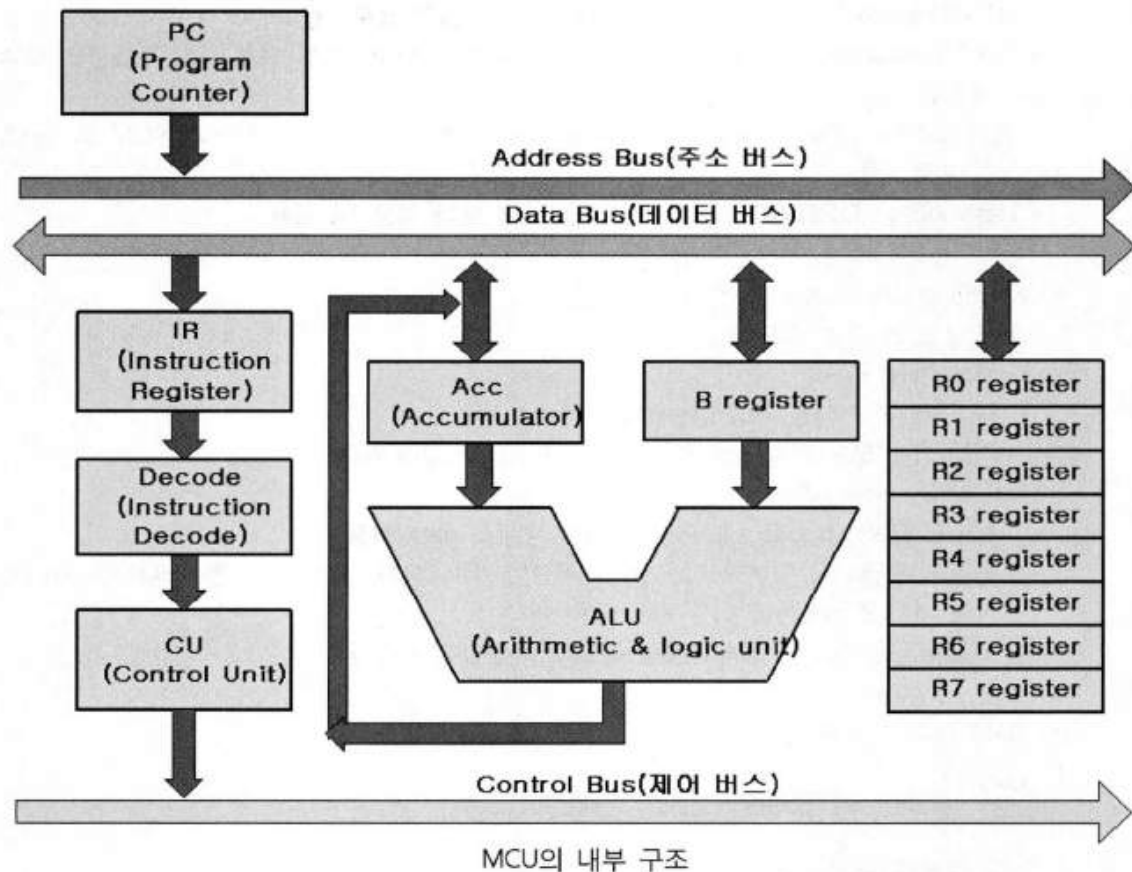
- 8비트 AVR의 내부구조 :



## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

- MCU의 내부 구조:



#### (b) ROM( read only memory)

- Mask ROM : 공장에서 마스크되어 프로그램
- PROM(Programmable ROM) : 사용자가 프로그램(지우지 못함)
- OTPROM(One Time Programmable ROM) : 한 번만 프로그램
- EPROM(Erasable Programmable ROM) : ROM Writer에 의해 프로그램, 자외선을 쬌어 내용을 지움
- EEPROM(Electrically Erasable Programmable ROM) : 전기적으로 지우며, ROM Writer없이 프로그램
- Flash ROM : EEPROM과 같은 기능, EEPROM 보다 적은 TR 사용

#### (c) RAM(Random Access Memory)

- 읽기 / 쓰기 가능, 휘발성
- SRAM(Static RAM)
  - Flip-Flop에 data를 저장하는 메모리
  - Refresh가 필요 없음, 소규모 고속 메모리 시스템에 사용
- DRAM(Dynamic RAM)
  - 하나의 TR와 Capacitor로 구성, 전하의 유무로 data 저장
  - 전원이 공급되어 있어도 짧은 시간 동안만 data 유지, 따라서 이 짧은 시간이 지나기 전에 다시 충전하는데 이를 Refresh라 한다.
  - 구조가 간단하고 고집적 대용량 메모리 시스템에 사용한다.

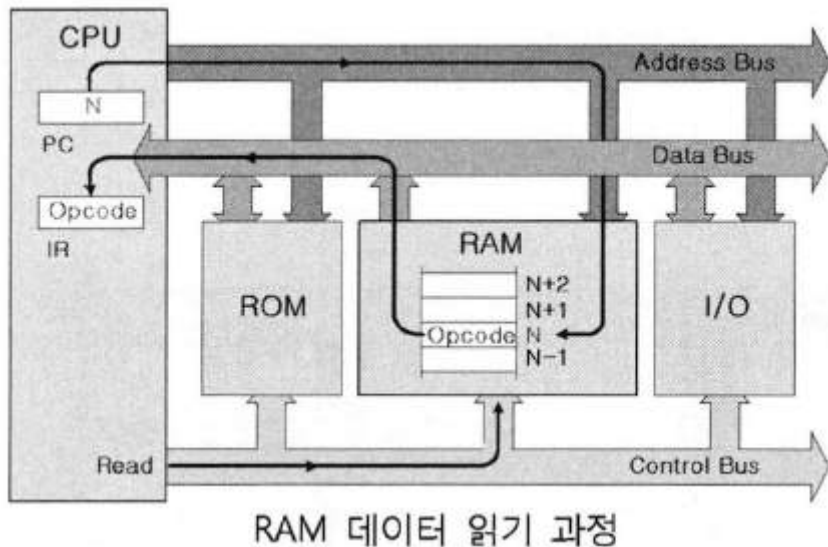


# 5.1 마이크로프로세서 (ATmega128)의 개요

## 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

- MCU의 내부 구조:

(d) RAM 데이터 읽기 과정



- Memory Mapped I/O
  - 하나의 Address, Data, Control Bus을 memory와 I/O가 공유한다.
  - Memory와 I/O가 주소 공간을 공유한다. 따라서 사용 가능한 주소 공간이 제한된다.
  - Input, Output 명령이 Memory access할 때와 같다. 따라서 Isolated I/O보다 간단한 구조로 되어있다.
  - I/O 동작을 할 때 훨씬 더 융통성이 있다
  - 대부분의 CPU에서 지원한다

- Isolated I/O
  - I/O 명령이 Memory 영역 Access 와 분리되어 있다. 즉 예를 들어서 마이크로프로세서에서 영역은 64Kb이고 메모리 영역은 256 byte 임.
  - I/O를 사용해도 메모리 용량은 감소하지 않음.

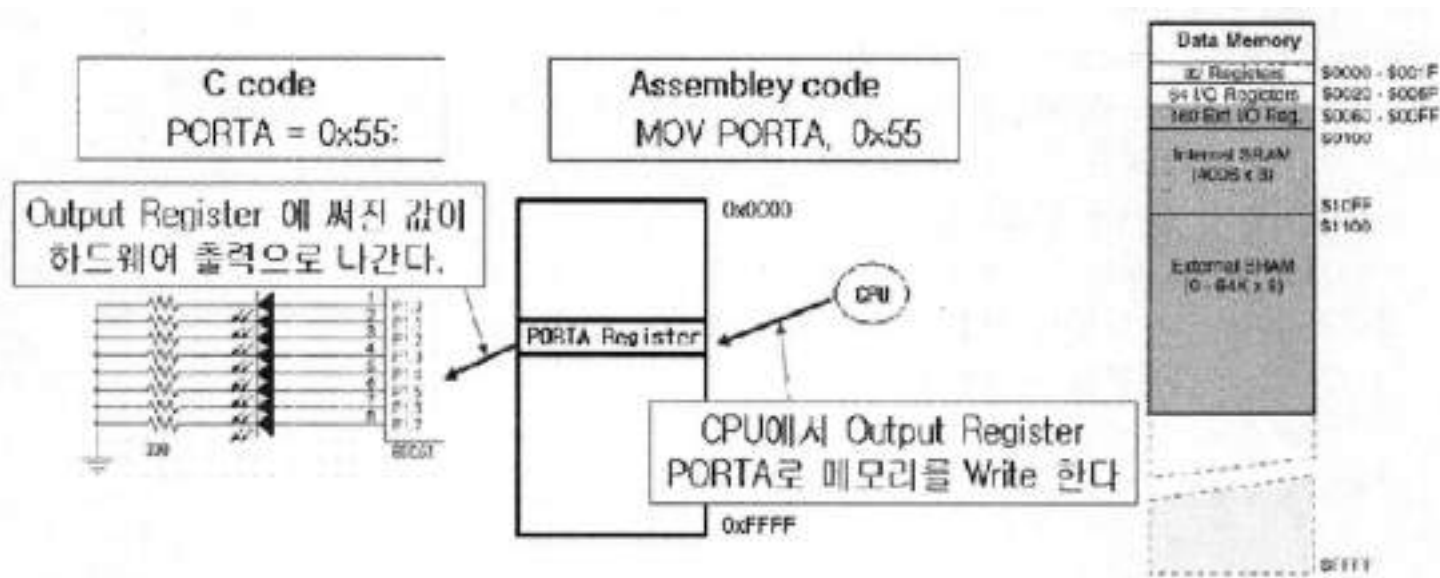
### 중요!

- SRF(Special Function Register)
  - MCU에 특정 기능으로 지정되어 있는 레지스터이다
  - 프로그램 제어 및 연산용 레지스터와 마이크로프로세서 주변의 기능을 제어하는 레지스터가 있다.
  - MCU 내부 램 중 일부 영역을 SFR 영역으로 사용한다. (사용하는 방법은 메모리에 데이터를 쓰거나 읽는 것과 같다.)
  - MCU가 어떤 동작을 하기 위해서는 이 SFR에 특정 데이터를 쓰거나 데이터를 읽는다.
  - Example : PORTA에 연결되어 있는 8개의 LED를 켜기  
PORTA에 5V 또는 0V의 출력을 주면 연결되어 있는 LED가 켜지거나 꺼지게 됩니다.  
SFR에 PORTA로 데이터를 출력할 수 있게 하는 PORTA라는 레지스터가 있습니다.

## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

- 메모리와 I/O의 개념:



- Address/Data Bus 개념
- Memory 구조와 Access 방법
- I/O 방식
- SFR (Special Function Register)

왼쪽 그림의 메모리 맵을 보면

총 0xFFFF 메모리 중 하위 0x10FF는

내부 메모리가 사용하고 있고

0x3000 번지는 Output Port 로 매핑이 되어 있다.

0x3002 번지는 Input Port 로 매핑이 되어 있다.

## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.1 개인용 컴퓨터와 마이크로컨트롤러의 차이

#### 중요!

- 인터럽트(Interrupt) :

- ✓ 인터럽트란? 주 프로그램 수행 중에 주 프로그램을 일시적으로 중지시키는 조건이나 사건 (event, 비동기적)의 발생을 말함 (인터럽트 처리 프로그램 → ISR (Interrupt Service Routine) or Interrupt handler)

- ✓ 인터럽트의 사용

I/O 장치를 사용할 때 어떤 특정 입력이 들어왔을 경우 어떻게 그것을 감지하는가?

방법 1 CPU가 주기적으로 감시하여 입력 여부를 확인함 → 폴링 방식

방법 2 입력이 있을 경우 CPU 에 알려줌 → 인터럽트 방식

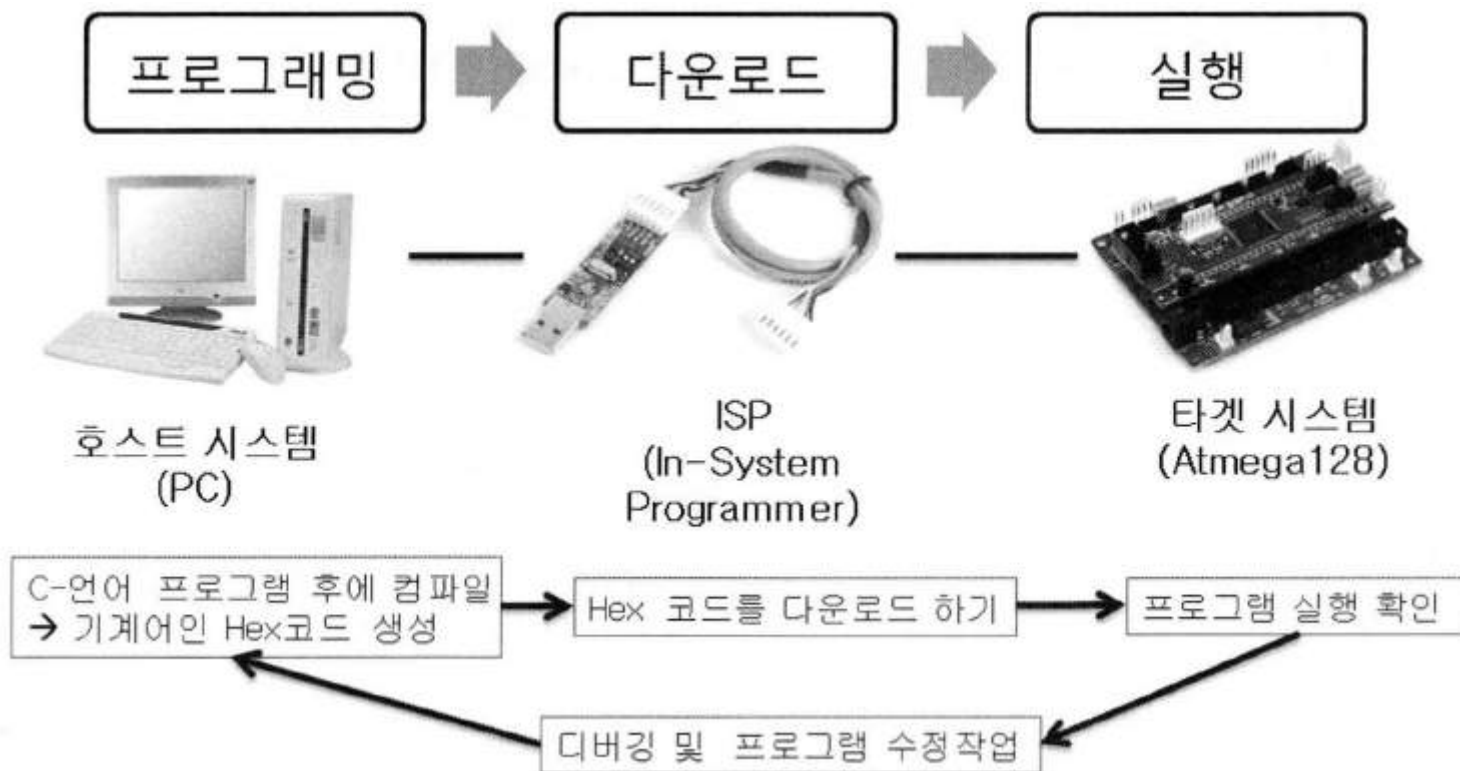
방법 3 입력이 있을 경우 CPU 에 알리지 않고 전용 회로가 자체적으로 처리함 → DMA 방식

- ✓ 인터럽트 처리의 일반적인 순서

인터럽트가 발생하면 주 프로그램 중단. > 현재 명령어의 수행을 마친다. > 다음 수행할 명령의 주소를 메모리에 저장한다 > 인터럽트가 더 이상 받아들여지지 않는다 > ISR 이 저장되어 있는 주소를 불러온다. > ISR이 수행된다 > 저장했던 주 프로그램 명령의 주소를 읽어온다. > 전에 수행 중이던 주 프로그램으로 복귀한다.

## 5.1 마이크로프로세서 (ATmega128)의 개요

### 5.1.2 하드웨어 구조 - ATmega128 개발 환경



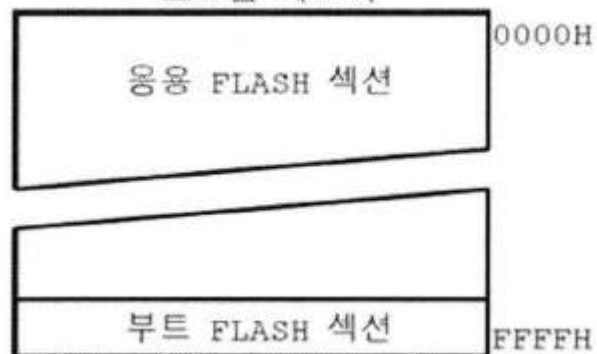
# 5.1 마이크로프로세서 (ATmega128)의 개요

## 5.1.3 마이크로프로세서의 기억장치

### 1. FLASH 프로그램 메모리

- ❖ 제어 전용 내장형(Embedded) 시스템  
이므로 프로그램은 전원을 꺼도 지워지  
지 않는 비휘발성 메모리에 저장된다.
- ❖ ISP(In-System-Programming)기술을 사  
용하여 PC에서 직접 프로그램을 다운  
로드할 수 있도록 되어있다.

#### 프로그램 메모리



### 2. 데이터 메모리

#### 데이터 메모리

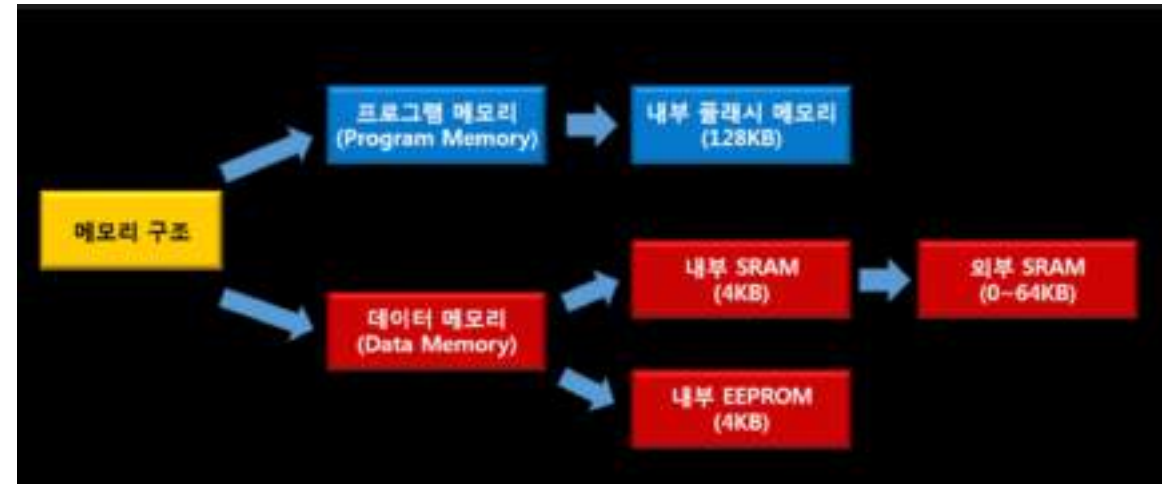
32 범용 레지스터	0000H - 001FH
64 I/O 레지스터	0020H - 005FH
160 Ext. I/O 레지스터	0060H - 00FFH
내부 SRAM (4096×8비트 = 4Kbyte)	0100H 10FFH 1100H
외부 SRAM (0 ~ 64K   × 8)	FFFFH



## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.1. ATmega128의 특징

- 8bit microcontroller :
  - 8개의 포트입력과 포트출력을 한 조로 구성하여 시스템을 구성
- ATmega128의 메모리:
  - 플래시 메모리로서 용량은 128 Kbytes
  - 4 Kbytes EEPROM
  - 64Kbytes의 확장 외부메모리 공간
- ATmega128의 입력과 출력장치 구성:
  - 53개의 프로그램이 가능한 입력과 출력 라인으로 구성
  - 오실레이터가 발생하는 실시간 카운터
  - 8 비트, 16 비트 타이머/카운터가 별도로 구성
  - 타이머 카운터를 이용하여 PWM제어 신호를 발생
  - 8 개의 10 비트 ADC 채널을 가지고 있다



## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.2. ATmega128의 하드웨어 구조

ATmega128은 64 Pin 으로 구성되어 다음과 같이 핀의 구조가 구성되어 있다.

전원 전압 공급핀 ( Power )

- VCC : 21, 52 ( +5V )
- GND : 22, 53, 63 ( 0V )

(2) 발진기 입력단자

- XTAL 또는 Oscillator : 23(XTAL1), 24(XTAL2)

(3) 리셋 입력핀

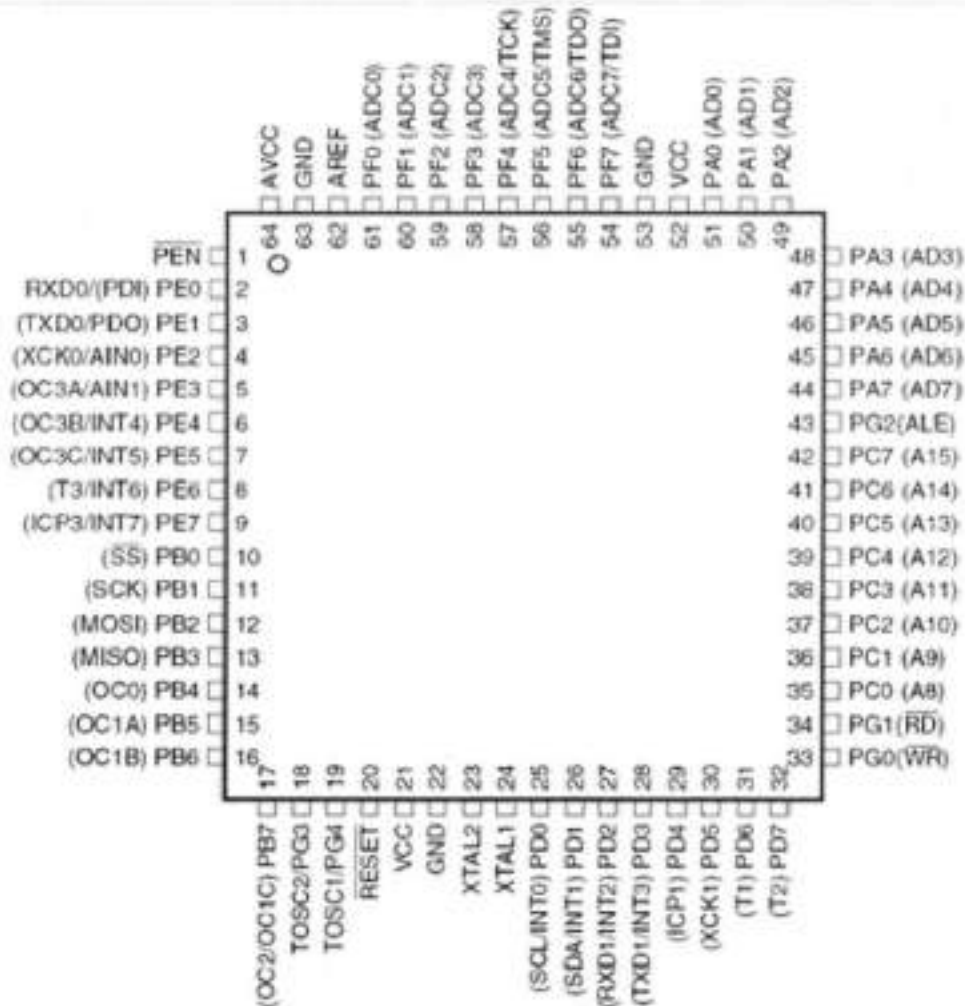
- /reset : 20

(4) 입력 ( Input ) 과 출력 ( Output )

- 8 bit port (6개) PORTA ~ PORTF
- 5 bit port (1개) → PORTG

(5) ADC포트

- PORTF의 멀티기능핀을 ADC로 선언하여 사용한다.



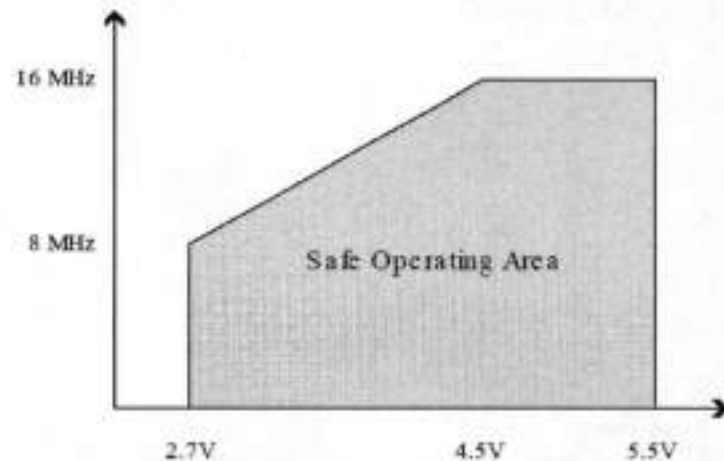
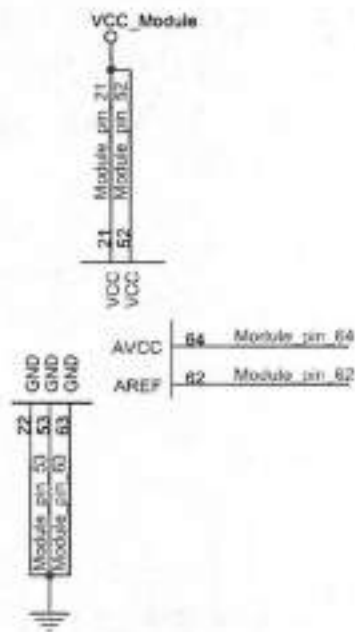
## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.3. ATmega128A-AU의 전원부 구성도 작동 전압

노이즈 감쇄가 중요한 경우, AGND(63번)와 GND(22, 53번) 사이는 LPF 역할을 하는 비드로 연결한다. 마이크로컨트롤러의 최소 단위는 비트(bit) 0(0V)과 1(5V)인 2진수를 의미한다. ATmega128은 직류(DC) 5V로 작동되면서 모든 신호는 0과 1로서 인식된다. ATmega128은 8Bit microprocessor로서 0과 1의 신호가 8개가 한 조가 되어 움직인다.

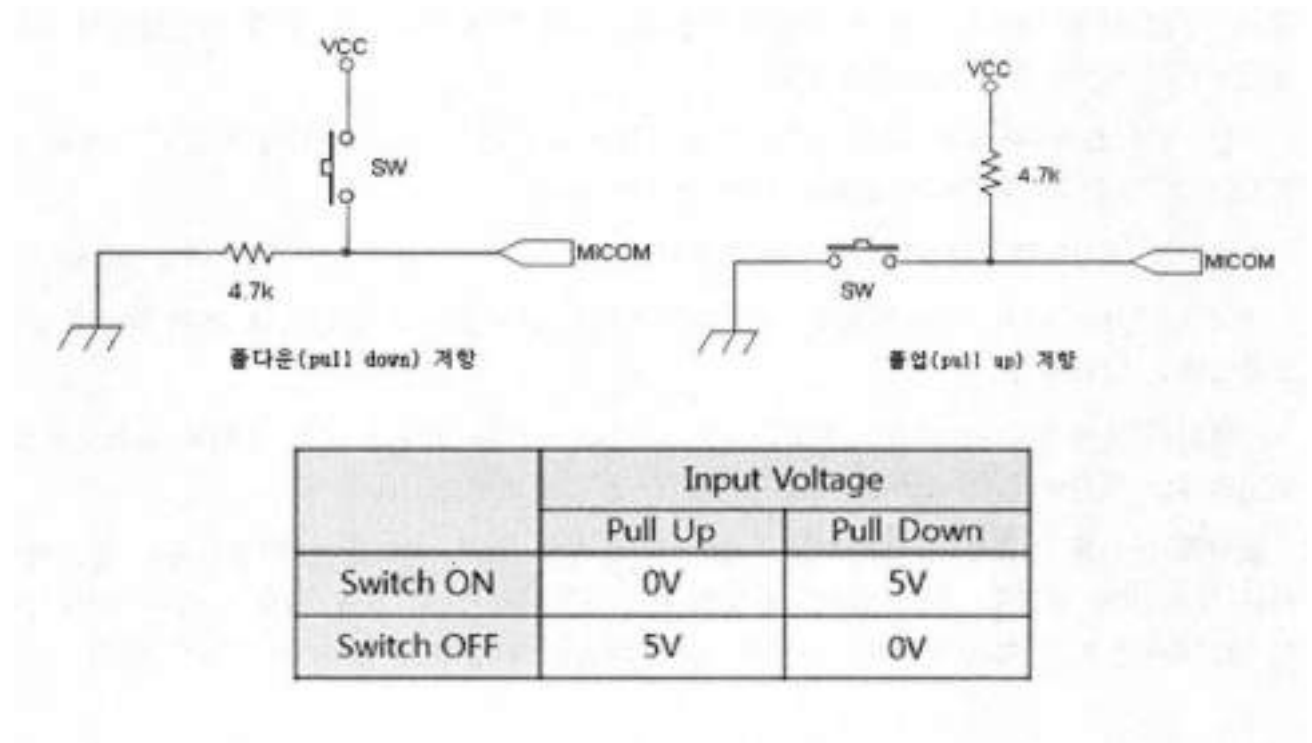
1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

TIP : ATmega128의 입력신호는 다음과 같이 Pull-Up과 Pull-Down 2개로 구성된다.



## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.3. ATmega128A-AU의 전원부 구성도 작동 전압



- Pull down : 평상시에 0V (low, 0) 에서 스위치가 On되면 5V(High, 1)가 된다.
- Pull Up : 평상시에 5V(High, 1)에서 스위치가 On되면 0V (low, 0)가 된다.

## 5.2 ATmega128 하드웨어 구성 이해

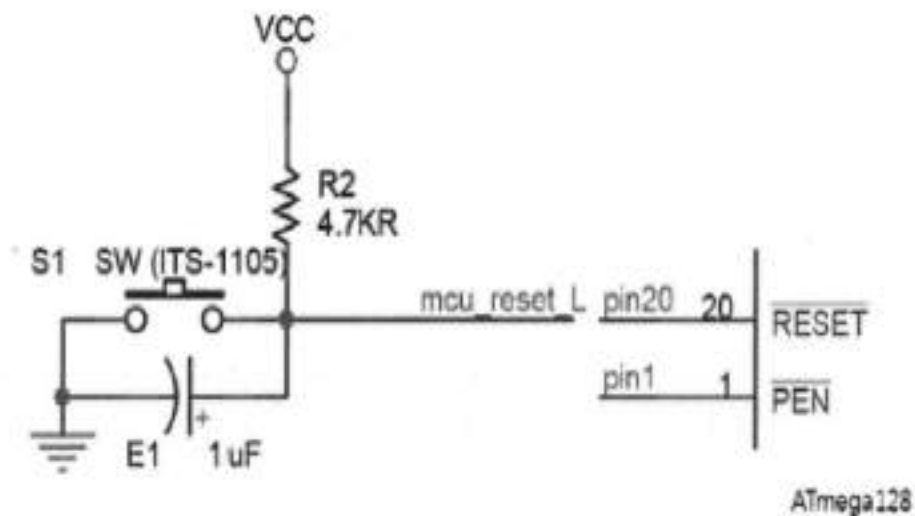
### 5.2.4. 리셋부

- 리셋회로란 ?  
ATmega128 이 동작이 멈추었을 때에 재작동 시키는 신호
- 마이크로프로세서의 작동이 멈추었을 때에 재부팅을 위해 신호가 리셋  
[" 0V" -> "5V " (Pull-Up 회로) ] 되는데, 수동으로 리셋 스위치를 누르면  
["5V" -> "0V" ] 가 된다



## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.4. 리셋부



보드에 전원을 ON하면은 전원이 들어오면서 제일먼저 보드를 초기화하는 리셋이 수행된다. 전원이 ON 되면, 5V(VCC)가 공급되고 저항(R2)을 통해 캐패시터(E1)로 충전된다. 마이크로컨트롤러 리셋(/RESET) 신호는 캐패시터에 충전되기 전에 짧은 시간동안 리셋상태가 되며, 충전되면서 리셋신호는 비활성화 된다.

또한 리셋 스위치(S1)은 외부 리셋으로 스위치를 누르면 캐패시터(E1)에 충전된 전하를 방전되어 0V가 된다. 즉 /RESET신호는 리셋 상태가 된다.

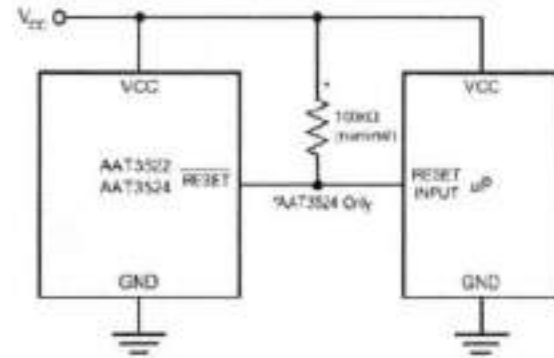
스위치를 개방하면 전원이 공급으로 캐패시터에는 충전되면서 리셋신호는 비활성화된다.

마이크로컨트롤러가 리셋이 되면 모든 레지스터는 초기값으로 설정되고, 리셋 벡터로부터 프로그램이 실행된다.

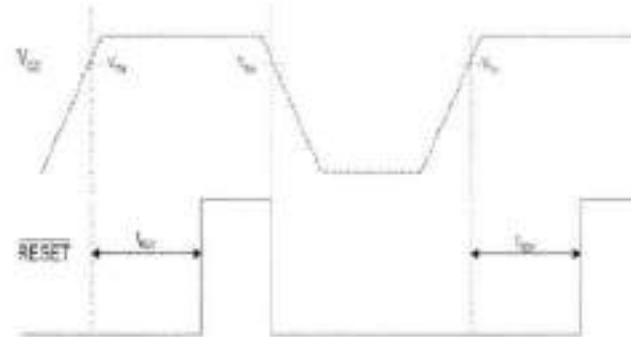
## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.5. 리셋회로의 보완

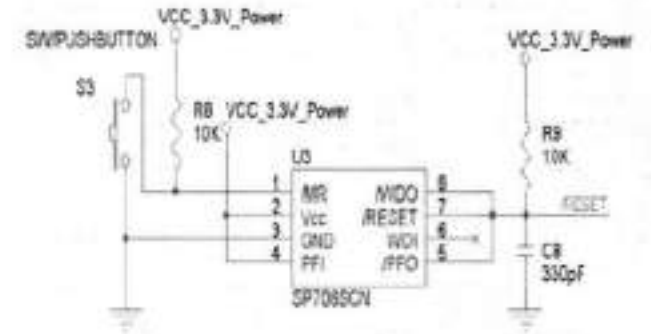
- 상용 회로에서는 안정적인 리셋 회로의 구성이 필수적(리셋전용 IC나 인버터 회로와 버퍼 등을 이용)
- 상용회로에서 사용되는 보다 안정적인 리셋회로 구성의 예



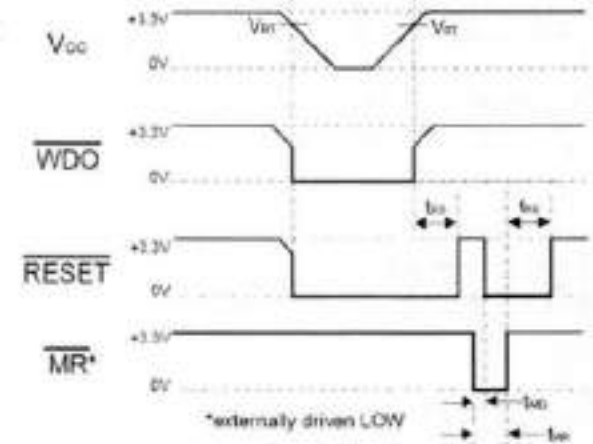
5V용 리셋회로 보완



5V용 리셋타이밍도



3.3V용 리셋회로 보완

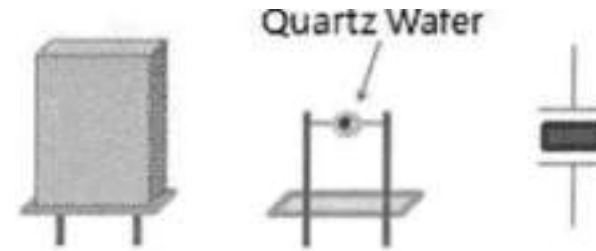


3.3V용 리셋타이밍

## 5.2 ATmega128 하드웨어 구성 이해

### 5.2.6. 수정 발진부

- 수정발진부란 수정발진부에 전압을 인가하여 기계적인 진동을 통하여 클럭을 발생시키는구 성장치
- 발진 (Oscillation) : 주기적이고 반복적인 진동  
정해진 공간에서 같은 운동을 반복하는 주기 운동(Periodicity)
- 발진자 (Oscillator) : 증폭 및 양의 피드백 현상을 이용한 주파수 발생 장치



ATmega128의 수정 발진자 ( Crystal Oscillation )  
수정 발진자 기본 클럭 소스로서 주파수 16MHz  
주파수 (초당 펄스 수) → 16,000,000 펄스 / 1초  
XTAL1 → input, XTAL → output

## 5.3 클럭부 연결

### 5.3.1. 외부 크리스탈 사용

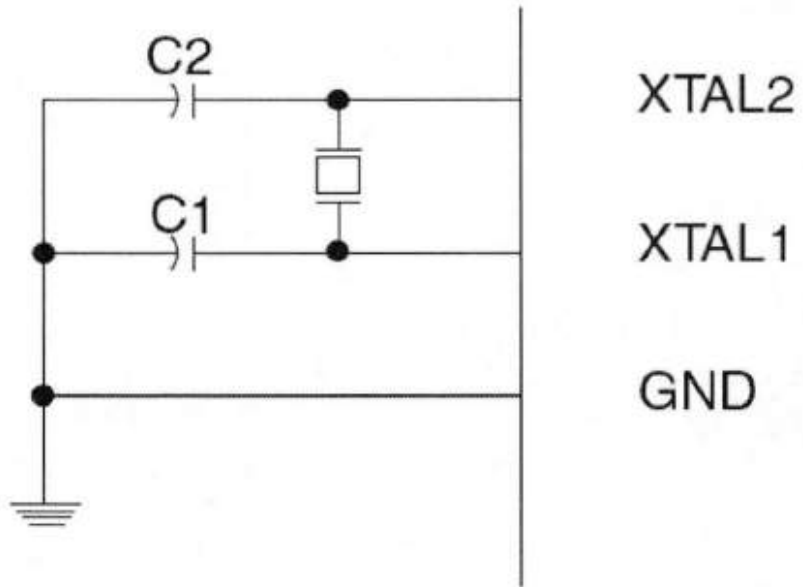
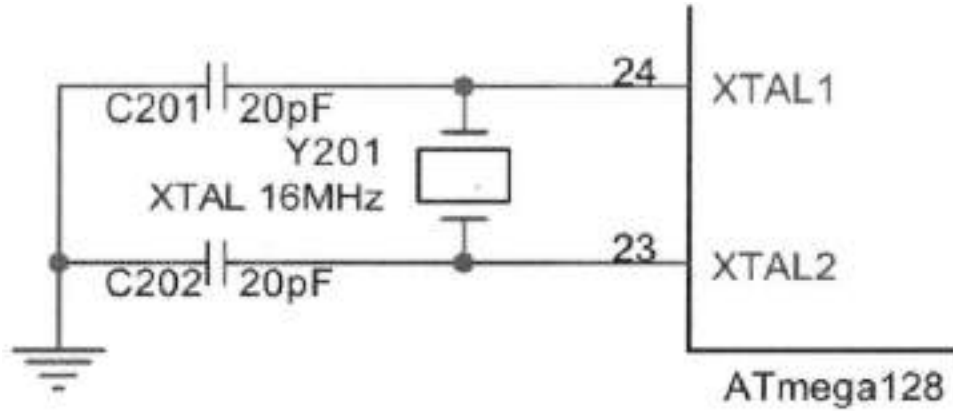


그림 2-6 외부 크리스탈의 사용

- XTAL1, XTAL2가 입력과 출력으로 사용되며, 수정 크리스탈 또는 세라믹 공진기 둘 다 사용이 가능
- C1과 C2는 크리스탈과 공진기 모두에서 같은 값이어야 하며 커패시터들의 이상적인 값은 사용 중인 크리스탈 또는 공진기, PCB 및 실제 마운팅되는 커패시턴스, 크리스탈에 들어 있는 커패시턴스 등을 합한 부유커패시터의 양, 전자기파 노이즈 등에 의존

## 5.3 클럭부 연결

### 5.3.2. 클럭부 연결



- 16MHz 크리스탈에 주파수 입력핀인 XTAL1과 주파수를 출력하는 XTAL2이 연결되어 있다.
- ATmega128에서는 클럭소스, 모드, 스타트업 시간 등을 선택하는 비트들이 있다.

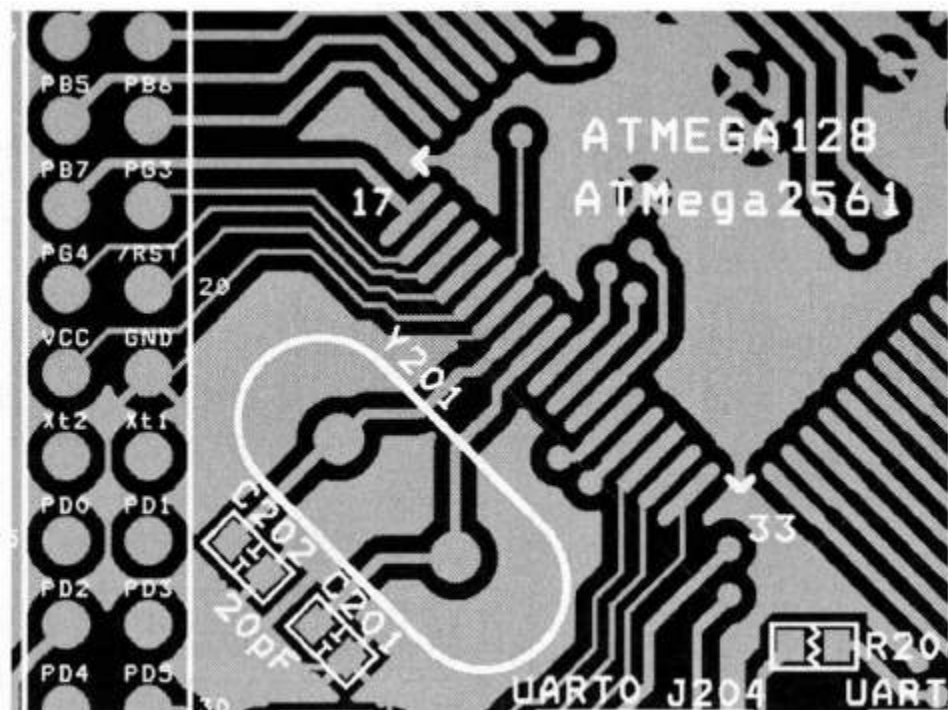
- 클럭부 소자 연결 시 주의사항

- ✓ MCU회로의 기본 클럭인 크리스탈 부분에 노이즈가 생기거나 타지 않도록, 가능한 짧은 선으로 AVR핀에 연결
- ✓ 커패시터에도 짧은 선을 사용하고, PCB설계 시에는 LGP(Localized Ground Plane)으로 설계하여, 다른 선들이 클럭부 주변을 지나가지 않도록 배선해야함

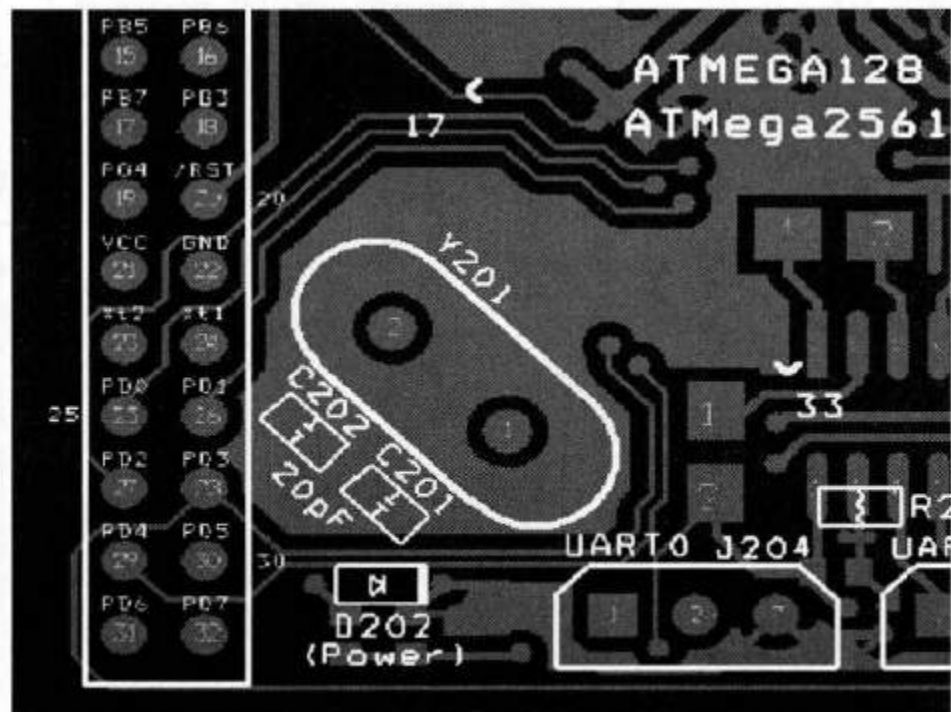


## 5.3 클럭부 연결

### 5.3.3. PCB에서의 클럭부 설계의 예



TOP(앞)면

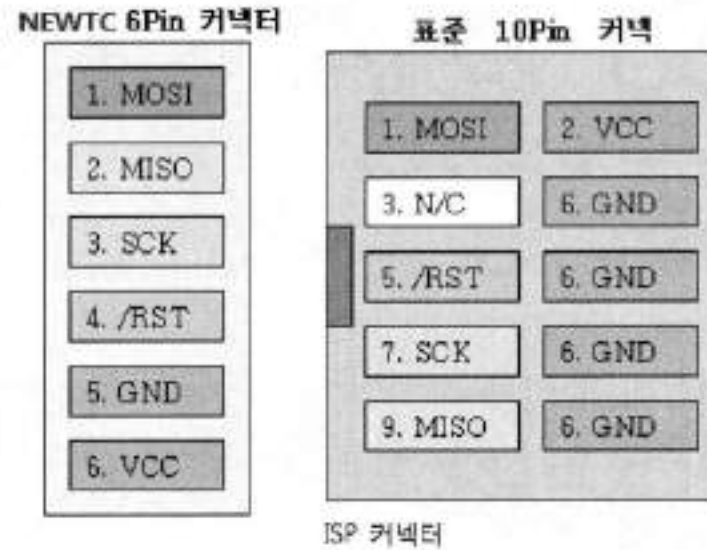


BOT(밑)면

## 5.3 클럭부 연결

### 5.3.4. ISP(In System Programming) 다운로드 커넥터

- ISP 커넥터는 개발용 PC와 ISP 케이블로 연결하여 사용자가 작성한 프로그램을 AVR내부 프로그램 메모리로 다운로드하는데 사용

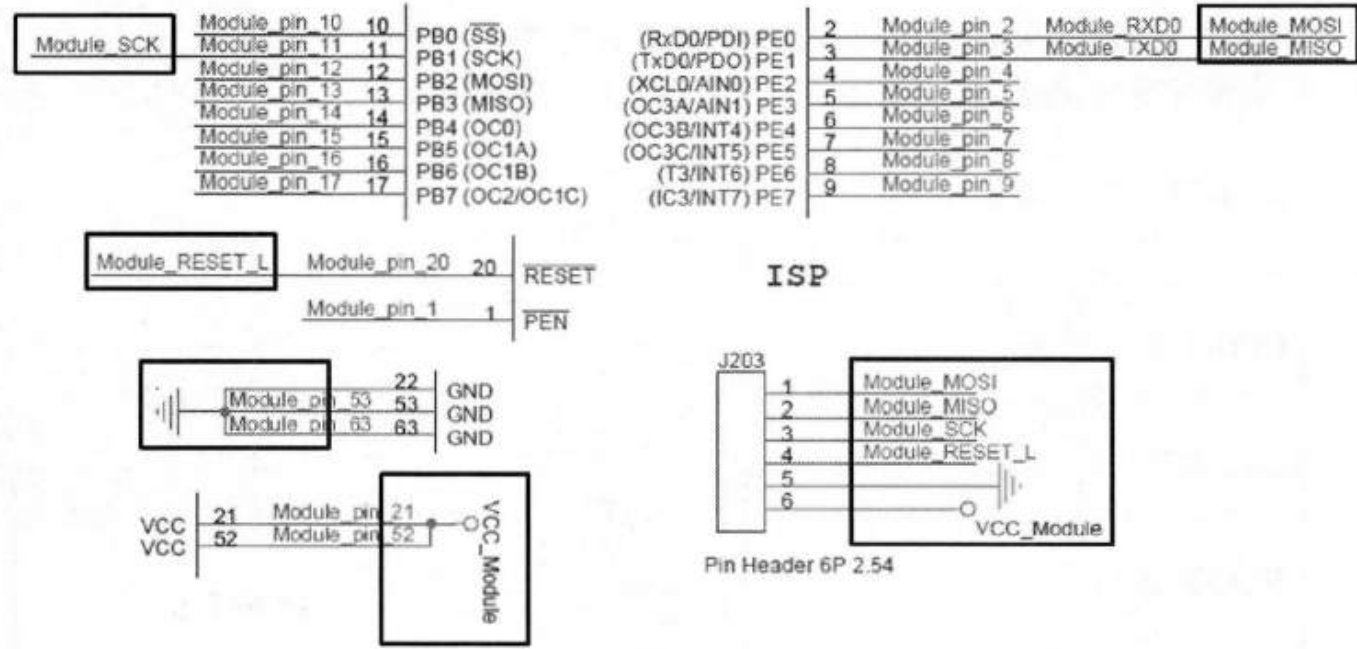


USB방식의 AD-USBISP V03.6과 USB방식의 AD-USBISP V6.0

## 5.3 클럭부 연결

### 5.3.4. ISP(In System Programming) 다운로드 커넥터

- ATMEL의 AVR은 USB 방식의 USBISP로 상용 비휘발성 메모리공간에 쓴 실행 프로그램에 의하여 Power ON Reset시에 자동으로 플래쉬의 첫 번째 주소부터 프로그램카운터가 시작 되어 플래시 메모리에 미리 써두었던 프로그램이 로드 됨
- 이 영역부터 SPI 시리얼 인터페이스를 통하여 STK500프로토콜로 CodeVision과 같은 프로그램을 사용하여 시스템의 플래시롬 영역인 프로그램 메모리에 재프로그램 될 수 있도록 되어있음



ATMEGA128에서의 ISP 회로

## 5.3 클럭부 연결

### 5.3.5. 입력(Input)과 출력(Output)

- 병렬 I/O 포트의 기본 구조

8개의 8비트 양방향 병렬 I/O 포트로 구성되어 있다. Port A ~ Port F

1	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---

1개의 5비트 양방향 병렬 I/O 포트 구성되어 있다. Port G

0	1	1	0	1
---	---	---	---	---

범용 I/O 포트 사용될 경우 read-modify-write 동작 가능

즉, 입출력 방향 변경 없이 SBI 및 CBI 명령에서 포트의 동작방향 달라질 수 있다.

ATmega128의 병렬 I/O 포트 Port A ~ Port G

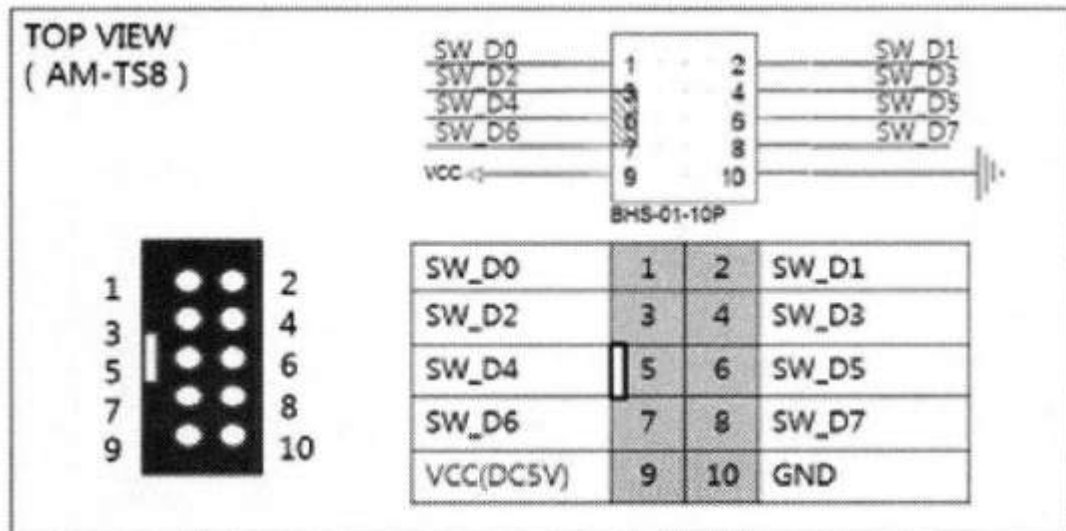
<b>PORT A</b> Module_PA0 Module_pin_51 51 Module_pin_50 50 Module_pin_49 49 Module_pin_48 48 Module_pin_47 47 Module_pin_46 46 Module_pin_45 45 Module_pin_44 44 PA0 (AD0) PA1 (AD1) PA2 (AD2) PA3 (AD3) PA4 (AD4) PA5 (AD5) PA6 (AD6) PA7 (AD7)	VCC VCC	(ACL/INT0) PD0 25 (SDA/INT1) PD1 26 (RXD1/INT2) PD2 27 (TXD1/INT3) PD3 28 (IC1) PD4 29 (XCK1) PD5 30 (T1) PD6 31 (T2) PD7 32 Module_pin_25 25 Module_pin_26 26 Module_pin_27 27 Module_pin_28 28 Module_pin_29 29 Module_pin_30 30 Module_pin_31 31 Module_pin_32 32 Module_RXD1 Module_TXD1	<b>PORT D</b>
<b>PORT B</b> Module_SCK Module_pin_10 10 Module_pin_11 11 Module_pin_12 12 Module_pin_13 13 Module_pin_14 14 Module_pin_15 15 Module_pin_16 16 Module_pin_17 17 PB0 (SS) PB1 (BCK) PB2 (MOSI) PB3 (MISO) PB4 (OC0) PB5 (OC1A) PB6 (OC1B) PB7 (OC2/OC1C)		(RXD0/PD0) PE0 2 (TXD0/PD0) PE1 3 (XCLK/AIN0) PE2 4 (OC3A/AIN1) PE3 5 (OC3B/INT4) PE4 6 (OC3C/INT5) PE5 7 (T3/INT6) PE6 8 (IC3/INT7) PE7 9 Module_pin_2 2 Module_pin_3 3 Module_pin_4 4 Module_pin_5 5 Module_pin_6 6 Module_pin_7 7 Module_pin_8 8 Module_pin_9 9 Module_RXD0 Module_TXD0 Module_MOSI Module_MISO	<b>PORT E</b>
<b>PORT C</b> Module_pin_35 35 Module_pin_36 36 Module_pin_37 37 Module_pin_38 38 Module_pin_39 39 Module_pin_40 40 Module_pin_41 41 Module_pin_42 42 PC0 (A8) PC1 (A9) PC2 (A10) PC3 (A11) PC4 (A12) PC5 (A13) PC6 (A14) PC7 (A15)		(ADC0) PF0 61 (ADC1) PF1 60 (ADC2) PF2 59 (ADC3) PF3 58 (TCK/ADC4) PF4 57 (TMS/ADC5) PF5 56 (TDO/ADC6) PF6 55 (TDI/ADC7) PF7 54 Module_pin_51 51 Module_pin_50 50 Module_pin_59 59 Module_pin_58 58 Module_pin_57 57 Module_pin_56 56 Module_pin_55 55 Module_pin_54 54	<b>PORT F</b>



## 5.3 클럭부 연결

### 5.3.6. ATmega128의 포트 핀 구성도

- 입력과 출력 PIN 연결선도



ATmega128의 Port A ~ Port G 핀 구성도

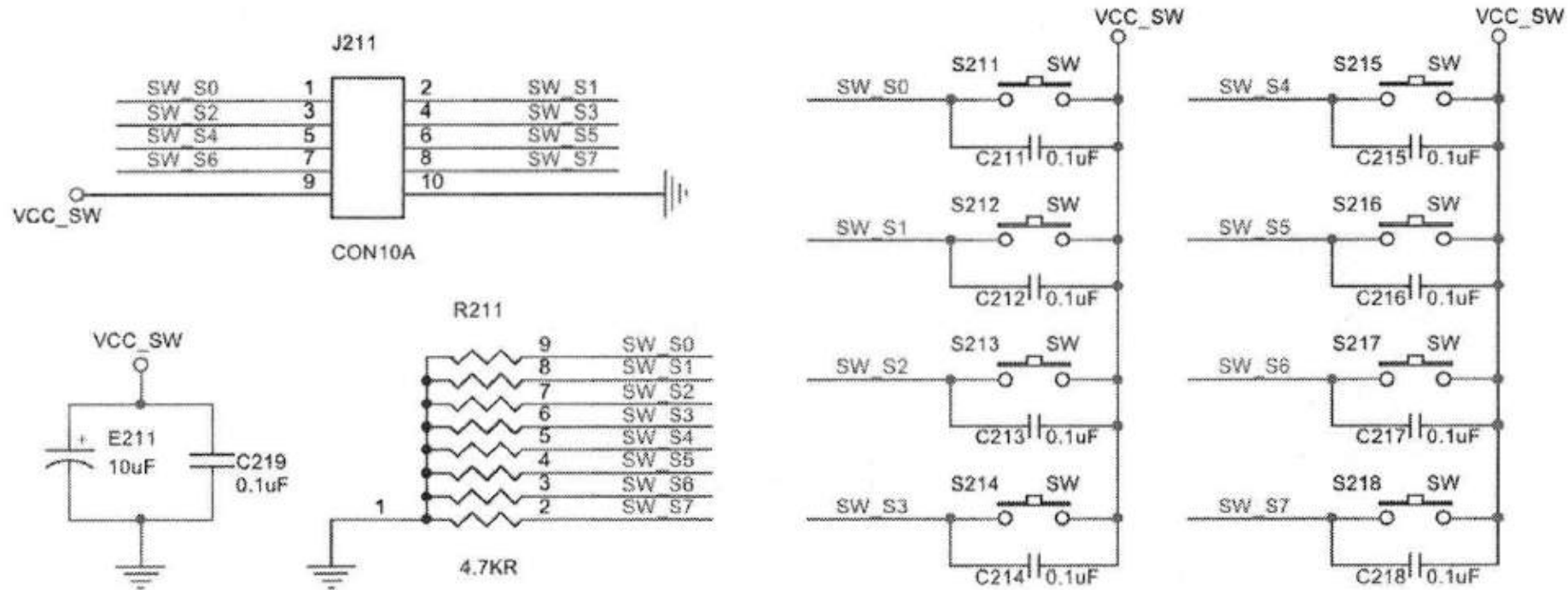
- (1) 직류전원 5V를 공급하기 위해 9번핀 VCC(5V), 그리고 10번핀 GND(0V)로 구성되어 있다.
- (2) 1번부터 8번까지는 8 비트 포트의 연결로서 Port에 0번부터 7번까지의 비트로 구성되어 있다.
- (3) PORTA.0 1, PORTA.1 2, PORTA.2 3,  
PORTA.3 4, PORTA.4 5, PORTA.5 6,  
PORTA.6 7, PORTA.7 8 으로 핀이 연결됨.



## 5.3 클럭부 연결

### 5.3.7. ATmega128의 입력장치 구성 (Pull-Down으로 구성)

- 입력부 구성 회로



- 채터링(Chattering) 현상 잡는 방법

- ✓ 디바운싱(Debouncing), 소프트웨어적 해결
- ✓ 신호 지연용 캐패시터 추가, 하드웨어 적 해결

## 5.3 클럭부 연결

### 5.3.8. ATmega128의 출력장치 구성 (LED를 출력으로 구성)

출력부 구성 회로 - LED

