

AVR 마이크로 컨트롤러

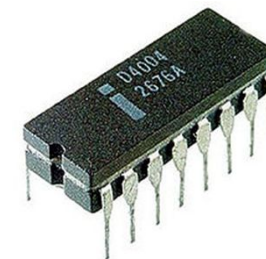
- 마이크로 컨트롤러
- AVR 마이크로 컨트롤러
- ATmega128 마이크로 컨트롤러



엣지아이랩

마이크로 컨트롤러

- 마이크로 프로세서
 - 프로세서를 하나의 칩 안에 집적하여 넣어 소형화한 형태
 - 컴퓨터의 발전과 함께 고성능의 프로세서로 발전했으며, 최근에는 64비트의 고성능 프로세서들이 출시
 - 마이크로 프로세서는 점점 고성능화 하면서 범용 컴퓨터에 사용
- 마이크로 프로세서의 발전
 - 1971년 Intel사 4bit 마이크로 프로세서 4004 개발
 - 이후 여러 회사에서 8bit 마이크로 프로세서 개발
 - Intel : 8008('72), 8080('74), 8085('76)
 - Motorola : MC6800('74), MC6805('76), MC6809('77)
 - Zilog : Z80('76)
 - 이후 8, 16, 32, 64bit 마이크로 프로세서 개발
 - Intel : 80186, 80286, 80386, 80486, Pentium,
 - Motorola : 68000, 68020, 68040, 68060,



인텔 4004
최초의 상용화된
마이크로 프로세서



인텔 80286
마이크로 프로세서

출처 : wikipedia

마이크로 컨트롤러

- 마이크로 컨트롤러(MCU : Micro Controller Unit)
 - 지능화와 소형화를 위하여 마이크로 프로세서에 메모리와 각종 주변장치들을 함께 집적하여 넣은 칩
 - 마이크로 프로세서 코어, 여러 가지 크기와 다양한 종류의 메모리, 여러 종류의 주변장치 및 입출력 포트를 하나의 칩에 집적
 - 여러 응용분야에 필요한 주변기기들을 모두 제공
- 마이크로 컨트롤러의 발전
 - 1975년 : Texas Instrument TMS1000 개발(마이크로컨트롤러의 시초)
(1971년 Intel 4bit 마이크로프로세서 4004 개발)
 - 1976년 : Intel 8bit M/C 8048(MCS-48) 개발
Motorola 8bit MC6801 개발
 - 1980년 : Intel 8bit M/C 8051(MCS-51) 개발
 - 1982년 : Intel 16bit MCS-96 개발
 - 1988년 : Intel 32bit M/C 80960 개발
 - etc



Texas Instrument
TMS1000

출처 : wikipedia

마이크로 컨트롤러

- 마이크로 컨트롤러의 특징
 - 주변장치들을 Sensing 및 제어하기 위한 I/O 능력이 강화
 - 타이머/카운터, 통신 포트 내장 및 인터럽트 처리 능력 보유
 - Bit 조작 능력이 강화
 - 제품의 소형화 및 경량화
 - 제품의 가격이 저렴(부품비, 제작비, 개발비 및 개발 시간 절감)
 - 융통성 및 확장성이 용이(프로그램만 변경)
 - 신뢰성이 향상(부품 수 적어 시스템 단순, 고장률 적고, 보수 편리)

마이크로 컨트롤러

● 마이크로 프로세서 vs 마이크로 컨트롤러

| 구분 | 마이크로 프로세서 | 마이크로 컨트롤러 |
|----|---|---|
| 특징 | <ul style="list-style-type: none"> • CPU를 단일 IC칩으로 만든 반도체 • MPU(Micro-Processor Unit) • 주변 하드웨어 필요 • 범용적인 목적의 컴퓨터에 사용 | <ul style="list-style-type: none"> • CPU+메모리+하드웨어 제어회로 • MCU(Micro-Controller Unit) • Single-Chip Microcomputer (1개의 칩으로 컴퓨터 구현 가능) • 특별한 목적의 기기 제어용으로 사용 |

마이크로 컨트롤러

- 마이크로 컨트롤러의 응용
 - 산업 : 모터 제어, 로봇 제어, 프로세스 제어, 수치 제어, 장난감 등
 - 계측 : 의료용 계측기, 오실로스코프 등
 - 가전제품 : 전자레인지, 가스 오븐, 전자 밥솥, 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

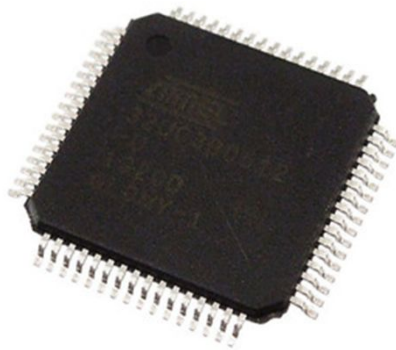
- 마이크로 컨트롤러 제조사
 - Motorola(FreeScale) : MC6805, MC68HC16, MC68332, HCS12
 - Samsung : KS51, KS88, KS16, KS32
 - Microchip : PIC16/17
 - Atmel : AVR시리즈, 8051시리즈
 - Zilog : Super-8
 - STMicro : STM32 시리즈
 - Texas Instrument : MSP시리즈

마이크로 컨트롤러

- 마이크로 컨트롤러(MCU)의 발전 방향
 - 고성능화
 - 32비트 ARM 코어를 내장
 - 다기능화
 - 다양한 특수 기능들을 내장
 - 소형화
 - 초소형 임베디드 시스템 장착
 - 저전력화
 - 소형 배터리로 장시간 동작이 가능
 - 저가격화
 - 1\$ 이하의 가격

AVR 마이크로 컨트롤러

- AVR 마이크로 컨트롤러
 - ATMEL사가 1997년에 처음 발표한 8비트 제어용 마이크로프로세서
 - Alf-Egil Bogen과 Vegard Wollan의 진보된 RISC(reduced instruction set code) 기술이 기반되었다고 하여 첫 글자를 따서 AVR이라고 명명
 - 많은 장점 때문에 상대적으로 늦게 출시되었음에도 불구하고 8051이나 PIC을 능가하는 인기를 단시간내에 얻게 되었음



출처 : wikipedia

ATMEL사
AVR 마이크로 컨트롤러

AVR 마이크로 컨트롤러

- AVR 마이크로 컨트롤러의 특징
 - RISC (Reduced Instruction Set Code) 구조
 - 컴퓨터의 실행속도를 높이기 위해 명령 세트를 축소 설계한 컴퓨터
 - 하버드 아키텍처(Harvard Architecture)
 - 프로그램과 데이터에 대해 별도의 메모리와 버스를 사용한 구조
 - 32개의 8비트 범용 레지스터를 가지는 레지스터 중심형 구조
 - CMOS 기술 채택으로 소비전력이 매우 적고 동작전압이 1.8-5.5V로 큼
 - 다양한 동작 모드를 제공해 저전력 동작 지원
 - 1K-256Kbyte 플래시롬, EEPROM 및 SRAM이 작은 칩 하나에 내장
 - 유사 RISC 구조와 32개의 레지스터 사용과 고집적으로 1MHz당 1MIPS 처리속도
 - 8핀에서 100핀의 외형과 이에 상응하는 메모리와 기능을 갖는 다양한 시리즈의 제품이 존재
 - 모델에 따라 매우 다양한 I/O 기능을 사용 가능
 - 외부 시스템 버스를 이용하여 데이터 메모리 또는 I/O 디바이스를 확장 가능
 - 다양한 인터럽트 소스와 처리 기능 보유
 - 무료 개발 도구인 Atmel Studio등, 다양하고 값싼 개발 도구 제공
 - 사용자 프로그램을 쉽게 다운로드 할 수 있는 ISP(In System Program) 기능 제공

AVR 마이크로 컨트롤러

- AVR 마이크로 컨트롤러의 종류
 - Tiny시리즈
 - 핀수가 8-24핀 정도의 작은 외형
 - 대부분 외부 시스템버스가 없음
 - 작은 메모리 용량(내부에 1K-2K byte 정도의 플래시 메모리를 보유)
 - 저속 저성능
 - 저가격(소형제어기에 적당)
 - Mega시리즈
 - 28-100핀 정도의 외형
 - 큰 메모리 용량(내부에 8K-256Kbyte 정도의 플래시 메모리와 256-4K byte 정도의 EEPROM과 512-4K byte의 SRAM을 내장)
 - 고속 고성능(20MHz의 클럭에서 20MIPS의 속도, 다양한 기능)
 - 고가격
 - AT90시리즈
 - 중간정도 사양을 가진 시리즈로 현재는 널리 사용되지 않음

ATMega128A 마이크로 컨트롤러

- ATMega128A 마이크로 컨트롤러
 - Atmel사의 8-bit 마이크로 컨트롤러
 - 고기능의 AVR Mega 시리즈 중에서도 가장 많이 사용되는 대용량의 마이크로 컨트롤러



출처 : google

ATMega128A

- ATMega128A 사양
 - 고성능 저전력 AVR 8bit Microcontroller
 - 향상된 RISC 아키텍처(16MIPS @ 16Mhz)
 - 133종 명령세트 (in 1 cycle)
 - 32 x 8bit 범용 레지스터+주변 컨트롤 레지스터/Multiplier (in 2 cycle)
 - 16MHz에서 16MIPS progress
 - 비휘발성 프로그램과 데이터 메모리
 - 128K Byte 내부 프로그램 가능한 ISP Flash memory
 - 4K Byte EEPROM
 - 4K Byte 내부 SRAM
 - 외부 최대 64K 메모리공간 확장 가능
 - 소프트웨어 보안을 위한 프로그램 Lock 기능
 - 선택적인 Boot code section (used In-System Programming by On-chip Boot Program)
 - ISP (In System Programming)를 통해 어플리케이션 영역과 부트영역에 F/W 다운로드 가능
 - 내장 메모리 Programming과 On-Chip Debug를 위한 JTAG 포트 지원

ATMega128A

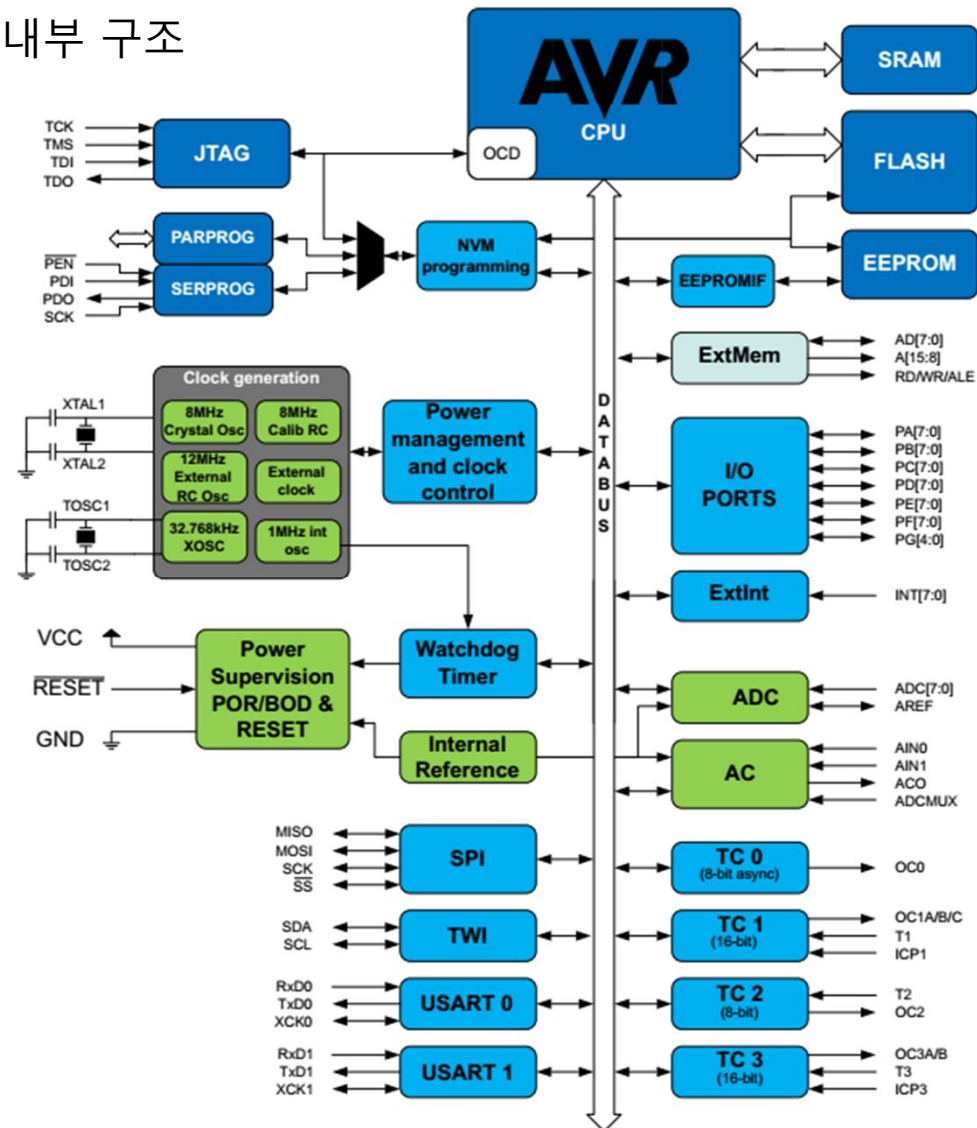
- ATMega128A 주변장치 특징
 - 분리된 프리스케일러와 비교모드를 가진 두 개의 확장 8비트 타이머/카운터
 - 분리된 프리스케일러와 비교모드, 캡처모드를 가진 두 개의 확장 16비트 타이머/카운터
 - 분리된 오실레이터에 의한 Real Time Count
 - 2개의 8bit PWM 채널
 - 6개의 프로그램 가능한 2~16bit PWM 채널
 - Output Compare Modulator
 - 8 채널, 10bit ADC
 - Two-wire Serial 인터페이스
 - 두개의 시리얼 USART
 - Master/Slave SPI 시리얼 인터페이스
 - 프로그램 가능한 와치독(Watchdog) 타이머
 - 아날로그 비교기

ATMega128A

- ATMega128A 특별한 특징
 - Power-on Reset, 안정된 전원공급을 위한 Programmable Brown-out Detection
 - 내부의 보정가능한 RC 오실레이터
 - 외부와 내부의 인터럽트 소스
 - 6개의 Sleep 모드
 - Idle, ADC Noise Reduction, Power-save, Power-down, Standby, Extended Standby
 - 소프트웨어적으로 선택 가능한 클럭 주파수
 - 전체 풀업 Disable* I/O 와 Package
 - 프로그램 가능한 53개의 I/O
 - 64-lead TQFP and 64-pad QFN/MLF
 - Operation Voltage : 2.7 ~ 5.5V

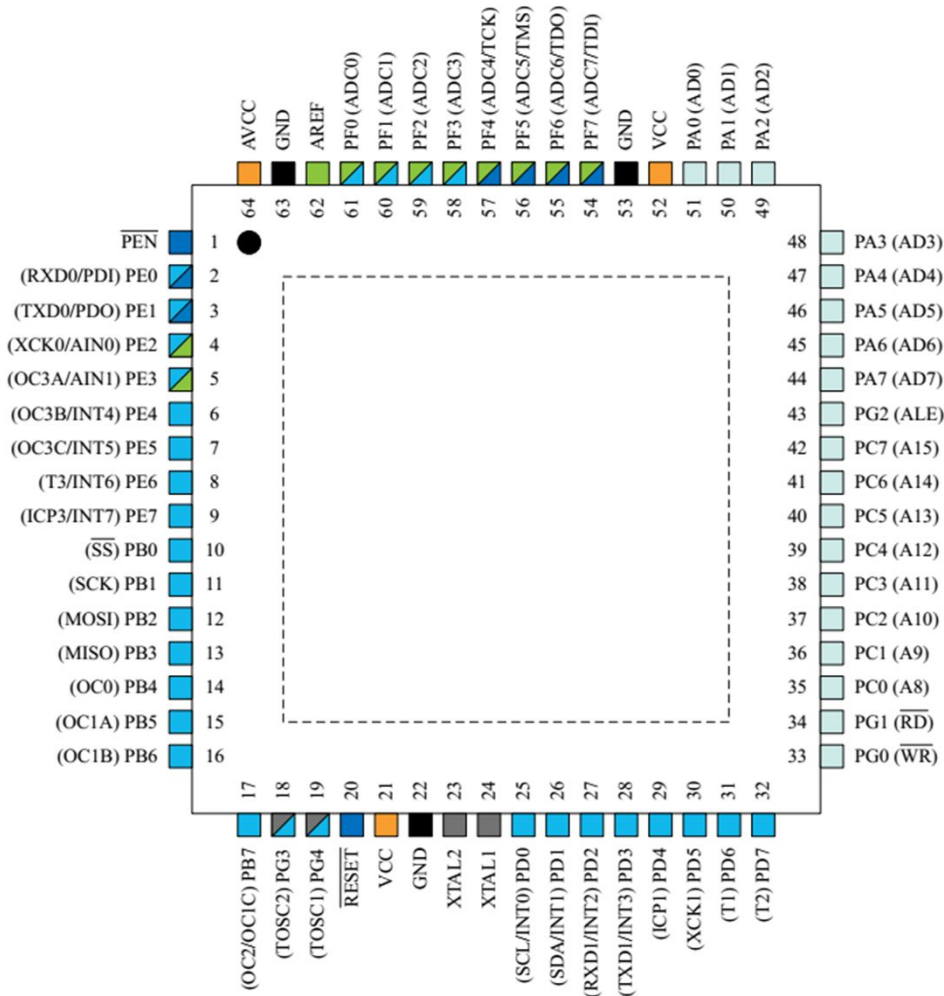
ATMega128A

- ATMega128A 내부 구조



ATMega128A

● ATMega128A 핀기능



● ATmega128A 패키지

- 64pin
- TQFP 혹은 MLF
- 6개의 범용 입출력 포트 제공

● 제어 신호

- PEN(핀1) : Programing Enable 신호. 파워 온 리셋시 0상태로 유지해 SPI를 허용하게 한다.
- RESET(핀20) : 시스템 리셋신호.
- Vcc(핀21,51) : 전원 입력 단자.
- AVCC(핀64) : AD변환기 및 포트 F에 대한 공급 전압
- XTAL1, XTAL2(핀24,23) : 발진용 증폭기 입력 및 출력 단자.
- GND (핀22,53,63) : 그라운드 입력 단자.
- AREF(핀62) : ADC 참조 전압 (Reference Voltage)
- Analog 핀(핀54~61)
- Digital 핀
- External Memory 핀

ATMega128A

- 범용 입출력 신호

- 포트A (PA7~PA0 : 핀44-51)
 - 8비트 양방향 입출력 단자, 외부메모리를 둘 경우에는 주소버스(A7-A0)와 데이터버스(D7-D0)로 사용
- 포트B (PB7~PB0 : 핀10-17)
 - 8비트 양방향 입출력 단자, SPI용 단자 혹은 PWM 단자로도 사용
- 포트C (PC7~PC0 : 핀35-42)
 - 8비트 양방향 입출력 단자, 외부메모리를 둘 경우 주소버스(A15-A8)로 사용
- 포트D (PD7~PD0 : 핀25-32)
 - 8비트 양방향 입출력 단자, 타이머용 단자 혹은 외부인터럽트용 단자로 사용
- 포트E (PE7~PE0 : 핀2-9)
 - 8비트 양방향 입출력 단자, 타이머용 단자, 외부인터럽트, 아날로그 비교기, USART용 단자로 사용
- 포트F (PF7~PF0 : 핀54-61)
 - 8비트 양방향 입출력 단자, AD변환기 or JTAG 인터페이스용 단자로도 사용
- 포트G (PG4~PG0 : 핀19, 18, 43, 34, 33)
 - 5비트 양방향 입출력 단자, 외부 메모리 스트로브 신호, RTC 타이머용 발진기 단자로 사용

ATMega128

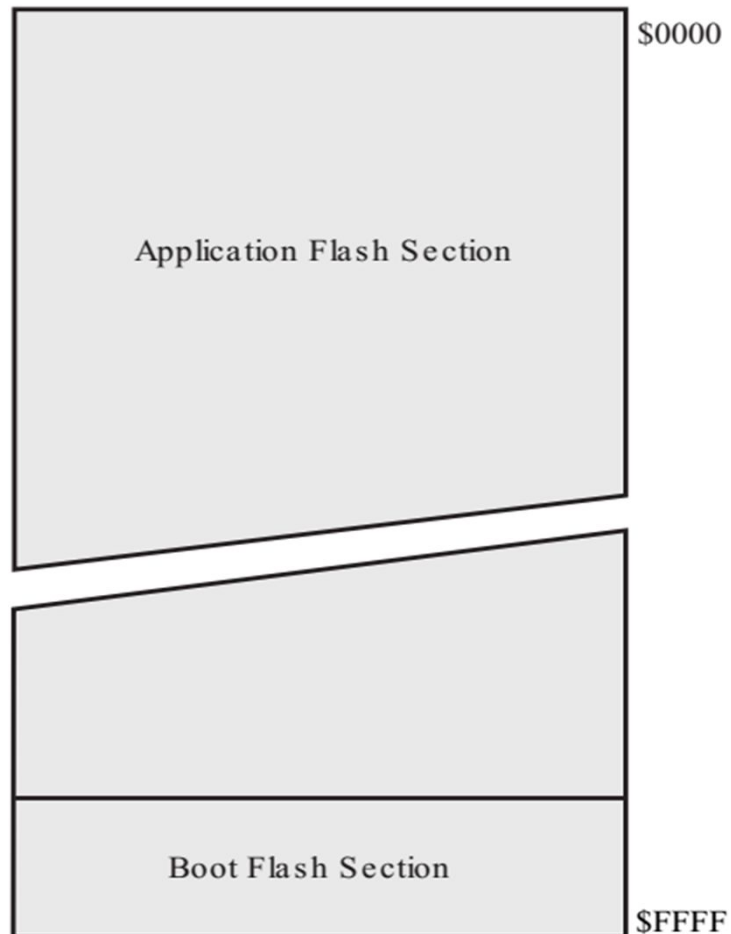
- ATMega128 메모리 구조
 - 하버드 구조
 - 프로그램 메모리와 데이터 메모리로 구분, 별개의 버스에 의해 접근
 - 프로그램 메모리 : 프로그램 코드를 저장하고 실행시키기 위해 필요한 메모리
 - 데이터 메모리 : 프로그램 실행에 필요한 데이터를 저장하는 메모리
 - RAM : 레지스터, SRAM
 - ROM : EEPROM
 - 외부 데이터 메모리

| 구분 | 프로그램 메모리 | 데이터 메모리 |
|-------|--------------------------|----------------------|
| 소자 | 플래쉬 메모리 | SRAM |
| 1워드 폭 | 16비트 | 8비트 |
| 크기 | 64[K] × 16비트 (128[K]바이트) | 4[K] × 8비트 (4[K]바이트) |
| 용도 | 프로그램 저장 | 데이터 저장 |

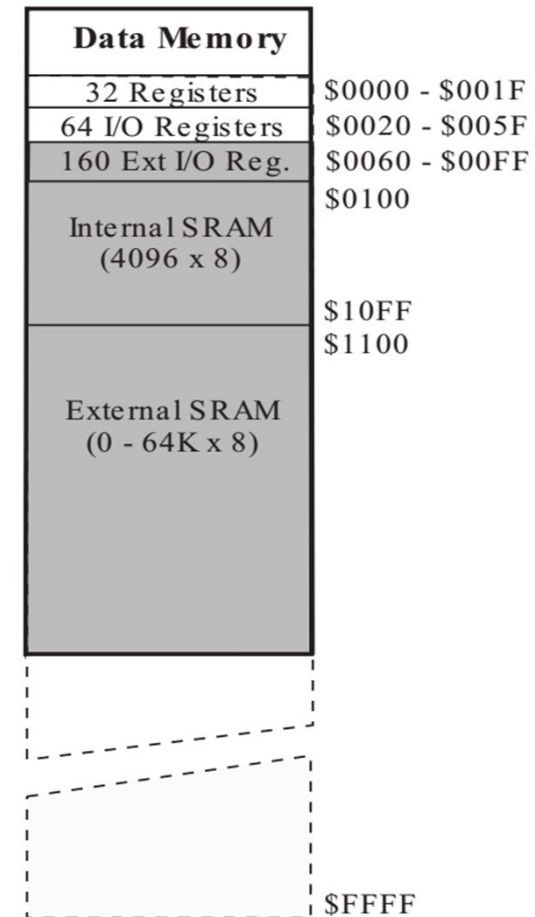
ATMega128A

- ATMega128 메모리 구조

Program Memory Map



Memory Configuration A



ATMega128A

- 프로그램 메모리
 - 프로그램 코드를 저장하고 실행하기 위해 필요한 메모리
 - 내부 128KB(byte)에 해당되는 플래시 메모리
 - 각 번지가 16bit씩 이뤄져 있음
 - Boot flash Section과 Application flash Section으로 나뉘어져 있음(Self-Programming 지원)
 - Application flash Section : 프로그램 코드를 저장하는 공간
 - Boot Flash Section : Boot Loader HEX코드를 저장하는 공간
 - BootLoader를 이용하여 ISP 없이도 소프트웨어 업그레이드 가능
- 데이터 메모리(레지스터)
 - 프로그램 이 실행될 때 임시로 데이터를 저장하는 고속 메모리
 - 2가지 종류로 나뉘어 짐
 - 범용 레지스터(General Purpose Register)
 - ALU 연산에 필요한 데이터들을 임시로 저장하는데 사용
 - 특수기능 레지스터(Special Function Register)
 - 칩의 I/O 제어나 상태보고 등의 특별 기능들을 수행

ATMega128A

- 범용 레지스터
 - R0 ~ R31까지 32개의 범용레지스터를 보유

| | 7 | 0 | Addr. | |
|--|-----|---|-------|----------------------|
| General Purpose Working Registers | R0 | | 0x00 | |
| | R1 | | 0x01 | |
| | R2 | | 0x02 | |
| | ... | | | |
| | R13 | | 0x0D | |
| | R14 | | 0x0E | |
| | R15 | | 0x0F | |
| | R16 | | 0x10 | |
| | R17 | | 0x11 | |
| | ... | | | |
| | R26 | | 0x1A | X-register Low Byte |
| | R27 | | 0x1B | X-register High Byte |
| | R28 | | 0x1C | Y-register Low Byte |
| | R29 | | 0x1D | Y-register High Byte |
| | R30 | | 0x1E | Z-register Low Byte |
| | R31 | | 0x1F | Z-register High Byte |

ATMega128A

- 특수기능 레지스터
 - 칩의 I/O 제어나 상태보고 등의 특별 기능들을 수행
 - 2가지 종류로 나뉘어 짐
 - I/O 레지스터
 - 64바이트 (0x20에서 0x5f번지) 로 구성
 - 내장된 각종 I/O 장치를 제어
 - 상태레지스터(SREG) : ALU의 연산 후 상태와 결과를 표시
 - 스택 포인터 (SP) : 스택 위치 표시
 - 확장 I/O 레지스터
 - 160바이트(0x60에서 0xff번지)로 구성
 - ATmega128A에 추가된 각종 I/O를 제어

ATMega128A

- 데이터 메모리(내부 SRAM)
 - 4K의 SRAM이 내장(메모리 번지는 \$0100~\$10FF까지 포함)
 - 내부 프로그램 코드에 의해 발생하는 데이터를 일시적으로 저장
 - 프로그램에서 사용자 변수의 저장영역이나 스택 영역으로 사용
 - 일반 모드와 ATmega103호환 모드로 나뉘어 짐
- 데이터 메모리(EEPROM)
 - 전원이 꺼지더라도 지속적으로 그 값을 유지해야 할 필요가 있는 별도의 데이터를 저장하기 위해 사용되는 메모리
 - 총 용량 4KB(byte)로 구성

ATMega128A

- 외부 데이터 메모리
 - 0x1100-0xffff번지에 외부 데이터 메모리를 연결하여 사용 가능
 - 외부 램, 외부 플래시 롬 또는 LCD나 AD변환기와 같은 주변장치의 인터페이스용으로 사용 가능
 - 외부메모리 인터페이스 기능
 - 주변장치와 적절한 인터페이스를 위한 0-3의 대기 사이클 지정 가능
 - 2개의 섹터로 외부 데이터 메모리를 분할하고 이들에 독립적인 대기 사이클을 지정할 수 있음
 - 16비트 주소의 상위바이트 중에서 필요한 갯수의 비트만을 주소 버스로 동작하게 할 수 있음
 - 데이터 버스의 신호들이 동작할 때 전류 소비량이 감소되도록 BUS-keeper 기능을 설정할 수 있음

ATMega128A

- ATMega128A 클럭
 - 다양한 소스에 의해 클럭을 발생시키고 분배가 가능
 - 소비 전력을 절약시키기 위해 개별적으로 공급 차단 가능
 - 입력 클럭의 종류
 - CPU 클럭
 - 범용레지스터, 상태레지스터, 데이터 메모리와 같은 AVR의 핵심적인 동작과 관련된 클럭
 - I/O 클럭
 - 타이머, SPI, USART 등 I/O모듈 대부분에서 사용되는 클럭
 - 플래쉬 클럭
 - 플래쉬롬과의 인터페이스를 제어하는 클럭
 - 비동기 타이머 클럭
 - 외부 32kHz 수정 발진기를 소스로 하는 비동기 타이머용 클럭
 - AD변환기 클럭
 - AD변환기용의 클럭

ATMega128A

- ATMega128A 클럭 발생원
 - 5가지의 클럭 발생원이 존재
 - CKSEL3~0와 SUT1~0, XDIV 레지스터를 이용하여 클럭 발생원과 주파수를 설정
 - 다섯가지 클럭 발생원
 - 내부 RC(디폴트 클럭) 발진기
 - 내장된 RC 발진기를 사용하는 경우
 - 외부 RC 발진기
 - 정밀한 타이밍이 요구되지 않는 용도로 외부에 RC 소자를 접속한 발진회로를 사용
 - 외부 수정 발진기
 - 외부에 크리스탈 또는 세라믹 레조네이터를 사용하는 경우
 - 저주파 수정 발진기
 - 외부에 32.768kHz의 낮은 주파수 크리스탈을 사용하는 경우
 - 외부 클록
 - 외부 다른 보드(8051보드, PIC보드)의 클럭을 가져와 XTAL1단자 연결

ATMega128A

- ATMega128A Sleep 모드
 - 전원을 절약할 수 있는 6가지의 다양한 슬립 모드가 제공
 - MCUCR(MCU Control Register) 레지스터를 설정하여 모드를 선택하고, SLEEP 명령을 실행하여 슬립 모드에 돌입하도록 함
 - 6가지 Sleep Mode
 - Idle모드
 - ADC noise reduction 모드
 - Power-save 모드
 - Power-down 모드
 - Standby 모드
 - Extended Standby 모드

ATMega128A

- ATMega128A Reset 모드
 - 정상적으로 동작하고 있는 마이크로컨트롤러가 리셋되면 모든 I/O레지스터값이 디폴트값으로 초기화되고, 프로그램은 리셋 벡터에서 시작
 - 리셋의 5가지 발생원
 - Power-on Reset
 - 전원전압 VCC의 전압 레벨에 따른 리셋
 - External Reset
 - /RESET 핀에 의한 외부 리셋
 - Watchdog Reset
 - 워치독 타이머에 의한 리셋
 - Brown-out Reset
 - Brown-out Detector에 의한 리셋
 - JTAG AVR Reset
 - JTAG 시스템에 의한 리셋