

제1강 : 개요

서울시립대학교
컴퓨터과학부

양 정 훈

임베디드 시스템 개요 (1)

▶ 임베디드 시스템

- ▶ 지능적인 처리를 위해 마이크로 프로세서를 ‘내장(embedded)’ 한 시스템
- ▶ 마이크로프로세서 혹은 마이크로컨트롤러를 내장하여 원래 제작자가 의도했던 특정 기능만을 수행하도록 제작된 장치, 범용 목적의 일반 PC와는 차이가 있음
- ▶ 임베디드 시스템은 보통 제품에 숨겨져 있으며 특정 기능만을 수행하는 컴퓨터시스템
- ▶ 휴대전화, PDA, 마이컴 밥통, VCR, 산업용 로봇, 과학 또는 의학 도구, 핸드폰 등의 제품들
- ▶ 포스트 피씨, 유비쿼터스의 물결로 인해 임베디드 시스템의 중요성이 더욱 부각



임베디드 시스템 특징 (1)

▶ 임베디드 시스템 하드웨어 구성요소

- ▶ 프로세서 : 가장 중요
- ▶ 메모리
- ▶ 입출력장치
- ▶ 네트워크 장치 등

▶ 임베디드 시스템 하드웨어의 특징

- ▶ 모든 임베디드 시스템은 마이크로프로세서가 필요하다.
- ▶ 응용 목적에 따라 제한된 주변장치와 응용 프로그램을 저장할 수 있는 최소한의 메모리 및 최소한의 성능을 가진 프로세서 사용.
- ▶ 임베디드 시스템은 프로그램을 저장하기 위한 메모리와 데이터를 저장하기 위한 메모리가 필요하다.
- ▶ 임베디드 시스템은 키보드, 스크린, 디스크드라이브 등을 가지고 있지 않은 경우가 많다.

임베디드 시스템 특징 (2)

▶ 임베디드 시스템 소프트웨어의 특징

▶ 운영체제

- 데스크탑 운영체제(WindowsXP...) 대신 소형화가 가능하며 즉각적인 응답을 보장하는 실시간 운영체제(RTOS)가 많이 사용, 값싼 장점.

▶ 시스템소프트웨어

- 운영체제, 각종 H/W 및 주변장치들을 구동하는데 필요한 드라이버 프로그램, TCP/IP를 비롯하여 시스템을 네트워크로 연결하는 데 필요한 각종 S/W, Programming Language Compiler, Debugger, Loader, ...

▶ 응용 소프트웨어

- 타겟 시스템 제어/모니터링 소프트웨어, 웹 브라우저 및 웹 응용 소프트웨어 등

임베디드 시스템 개발의 난점 (1)

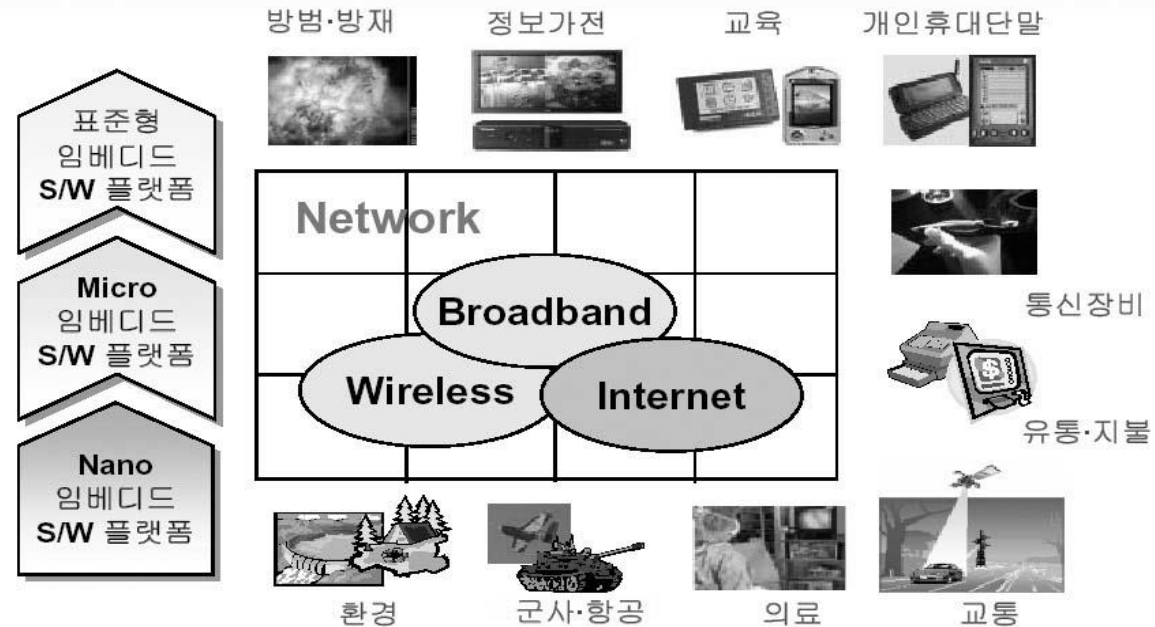
▶ 임베디드 시스템 개발의 난점

- ▶ 작업 처리량 : 만들려는 시스템이 짧은 시간동안 너무도 많은 데이터를 처리해야 할 필요가 있을 수도 있다.
- ▶ 응답성 : 시스템이 즉시 반응 해야 할 필요가 있을 수 있다.
- ▶ 검증성 : 임베디드 시스템의 소프트웨어를 시험하기 위한 환경을 만드는 것은 매우 어려운 일이다.
- ▶ 오류수정능력 : 화면 출력 장치나 키보드 없이 소프트웨어가 어떤 문제가 있는 건지 (단순히 동작을 안 한다는 사실 말고)를 찾아 낸다는 것은 정말 어려운 일이다.
- ▶ 저장공간 : 임베디드 시스템은 메모리의 크기가 제한되어 있기 때문에 프로그램과 데이터를 사용하는 메모리의 크기에 맞추어야 한다.
- ▶ 임베디드 시스템에서의 소프트웨어 실패는 데스크탑에서보다 훨씬 심각한 문제를 야기한다.

임베디드 시스템 개발의 난점 (2)

- ▶ 임베디드 시스템은 어떤 환경조건에서도 동작할 수 있어야 한다.
- ▶ 프로그램 설치 : 소프트웨어를 임베디드 시스템에 설치하기 위해서는 특별한 툴을 사용해야 한다. 임베디드 시스템은 ROM에 모든 코드를 넣는다.
- ▶ 전력소비 : 휴대용 시스템은 배터리 전력을 이용해야 하기 때문에, 시스템의 소프트웨어는 전력을 헛되이 쓰지 않도록 해야 한다.
- ▶ 비용 : 대부분의 임베디드 시스템 프로젝트에서는 하드웨어의 비용을 줄이는 것은 매우 중요한 문제이다. 소프트웨어들은 늘 목적에만 맞게끔 설계된 하드웨어 위에서 동작해야 한다.
- ▶ 임베디드 시스템의 용어가 통일이 되어 있지 않다는 것도 임베디드 시스템 개발에서의 어려운 점이다.

임베디드 시스템의 응용분야



- ▶ 정보가전
 - ▶ Digital TV(VOD, 양방향 TV), 냉장고, 세탁기, 전자레인지, ...
- ▶ 정보단말
 - ▶ PDA, Mobile Phone, Smart Phone(PDA & Mobile) ...
- ▶ 통신장비
 - ▶ 디지털 교환기, 사설 자동 전화교환기(PABX; Private Automatic Branch Exchange), Router, Switch, ...

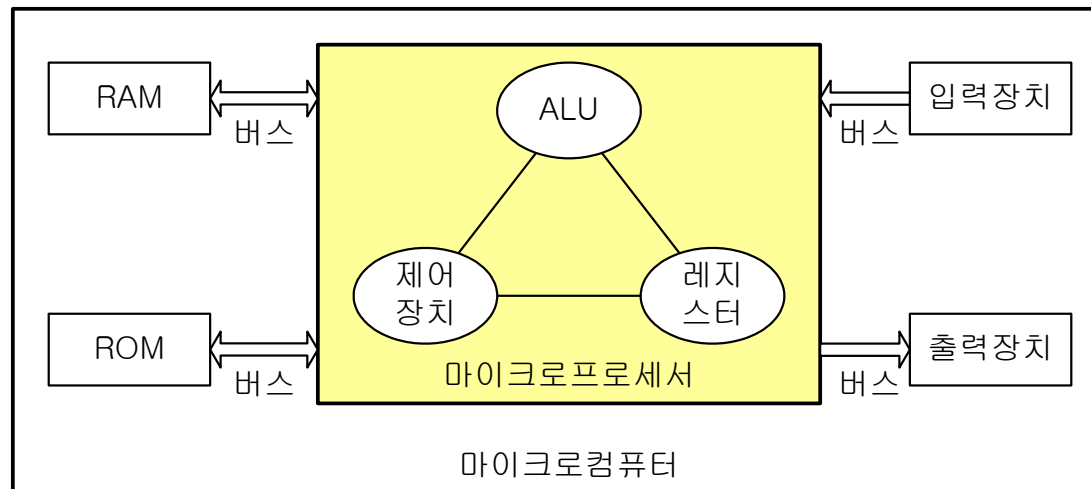
마이크로프로세서 및 마이크로컴퓨터 (1)

▶ 마이크로프로세서 (Microprocessor)

컴퓨터의 CPU(Central Processing Unit)가 가진 기능의 대부분을 하나 혹은 몇 개의 반도체 칩(Chip)으로 집적한 것

▶ 마이크로컴퓨터 (Microcomputer)

마이크로프로세서를 중심으로 ROM, RAM, I/O 장치 등으로 구성된 작은 규모의 컴퓨터 시스템



마이크로프로세서 및 마이크로컴퓨터 (2)

▶ 레지스터(register)

데이터를 보관하는 조그만 메모리로, 프로그램의 실행 중에 사용되며 고속 액세스 가능

▶ 산술/논리 유닛(ALU : 연산장치)

가산이나 승산 등의 산술 연산을 수행하고, 레지스터 내의 하나 혹은 두 개의 값 사이에서 AND 조작과 같은 논리 연산을 수행.

▶ 제어 장치(control unit)

명령을 해석하고 그것을 실행하는데 필요한 컴퓨터 내부의 각 유닛 사이의 데이터의 흐름을 제어

▶ 버스(Bus)

마이크로프로세서와 각 장치들이 서로 정보를 교환하기 위해 필요한 전송로

- **주소 버스** : 메모리 내의 특정 장소나 입출력 장치의 특정 포트(port)를 지정하는 주소가 실린다.

- **데이터 버스** : 각 장치간에 주고받는 정보가 실린다.

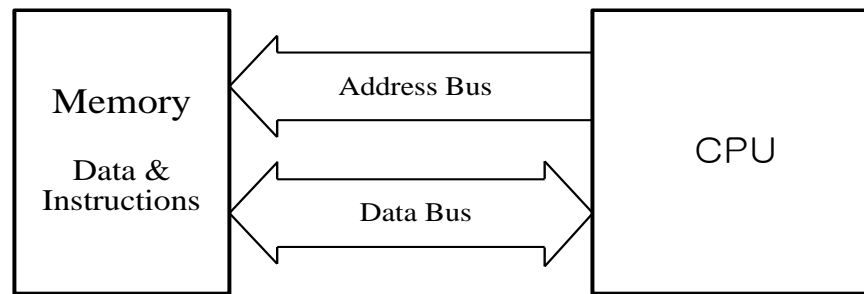
- **제어 버스** : CPU 내부에서 또는 외부로부터 시스템 동작을 제어하는 신호가 실린다.

참고 : 마이크로컴퓨터 구조 (1)

- 마이크로프로세서 출현 배경
 - 1969년 인텔의 연구개발자 테드 호프 박사는 슈퍼컴퓨터의 중앙처리장치(CPU)를 대신할 수 있는 초소형 연산기를 연구.
 - 비슷한 시기 인텔은 일본의 부지컴이라는 계산기를 만드는 전자회사로부터 전자식 탁상시계 CPU를 12개의 칩으로 만들어 줄 것을 요구받음.
 - 호프 박사는 부지컴이 요구한 기능을 하나의 실리콘 칩에 모두 집적화할 수 있는 방법을 고안함.
 - ▶ 그러나 부지컴이 파산하면서 인텔은 고민 끝에 이를 ‘4004’라는 독자 브랜드로 출시(1971년). 72년 8비트 8008. 74년 8080(최초의 PC인 Altair 8800의 CPU)
 - ▶ 칩 안에 컴퓨터가 있다는 개념으로 프로그램만 바꾸면 다양한 연산기능을 수행할 수 있고 다른 용도의 제품 개발이 가능하여 폭발적인 인기를 얻음.

참고 : 마이크로컴퓨터 구조 (2)

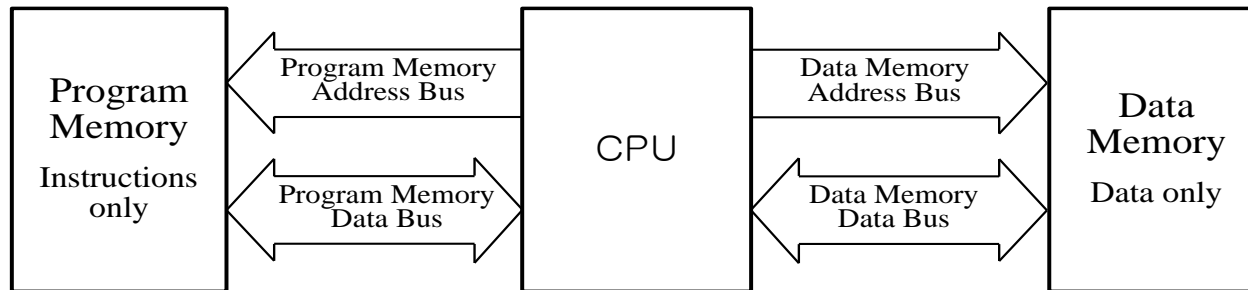
▶ 하버드 구조와 폰 노이만 구조



데스크 탑 PC는
폰 노이만 구조

8051, PIC, AVR
는 하버드 구조

폰 노이만 구조



하버드 구조

참고 : 마이크로컴퓨터 구조 (3)

▶ RISC(Reduced Instruction Set Computer) 구조

- 주로 대형컴퓨터나 워크스테이션에 사용되는 CPU의 한 방식
- CPU에 내장된 명령어를 줄여 보다 빠른 처리속도가 특징
- CPU의 구조를 보다 단순화
- 대형컴퓨터나 워크스테이션과 같이 대용량의 데이터를 고속으로 처리하는 컴퓨터에서 RISC 방식의 CPU를 사용
- 복잡한 명령어는 단순한 명령어를 조합해서 사용

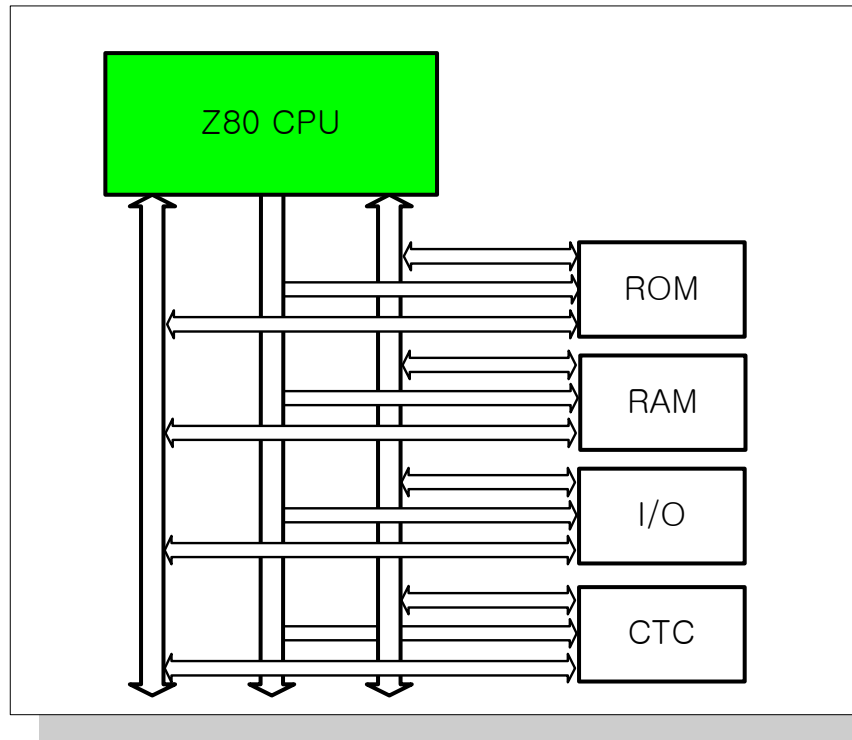
▶ CISC(Complex Instruction Set Computer) 구조

- 대표적으로 인텔의 x86계열의 CPU가 CISC 방식을 사용하며, 펜티엄 IV도 CISC 방식을 사용
- CPU의 구조가 매우 복잡
- CISC 방식의 장점은 뛰어난 호환성에 있음
- 개인용 컴퓨터인 PC에 많이 사용

참고 : 마이크로컴퓨터 기타 (1)

▶ 단일 보드 마이크로컴퓨터(One Board Microcomputer)

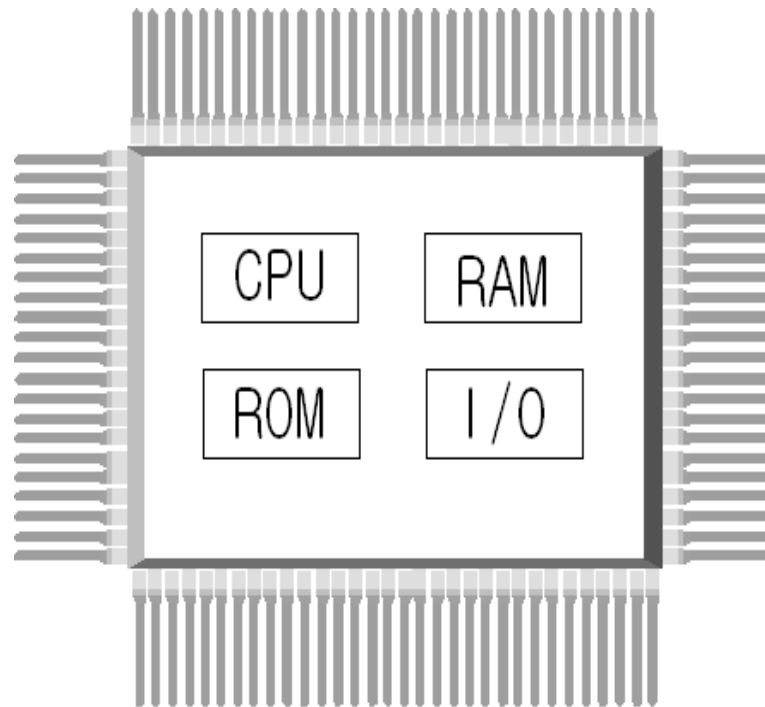
마이크로컴퓨터 중에서 부품들이 한 장의 기판 위에 배치해 놓은 것 .



참고 : 마이크로컴퓨터 기타 (2)

▶ 단일 칩 마이크로컴퓨터(Single Chip Microcomputer)

단일보드 마이크로컴퓨터에 들어가는 모든 부품들을 하나의 반도체 칩 (VLSI)에 집적



**One Chip
Microcomputer**

마이크로컨트롤러 (1)

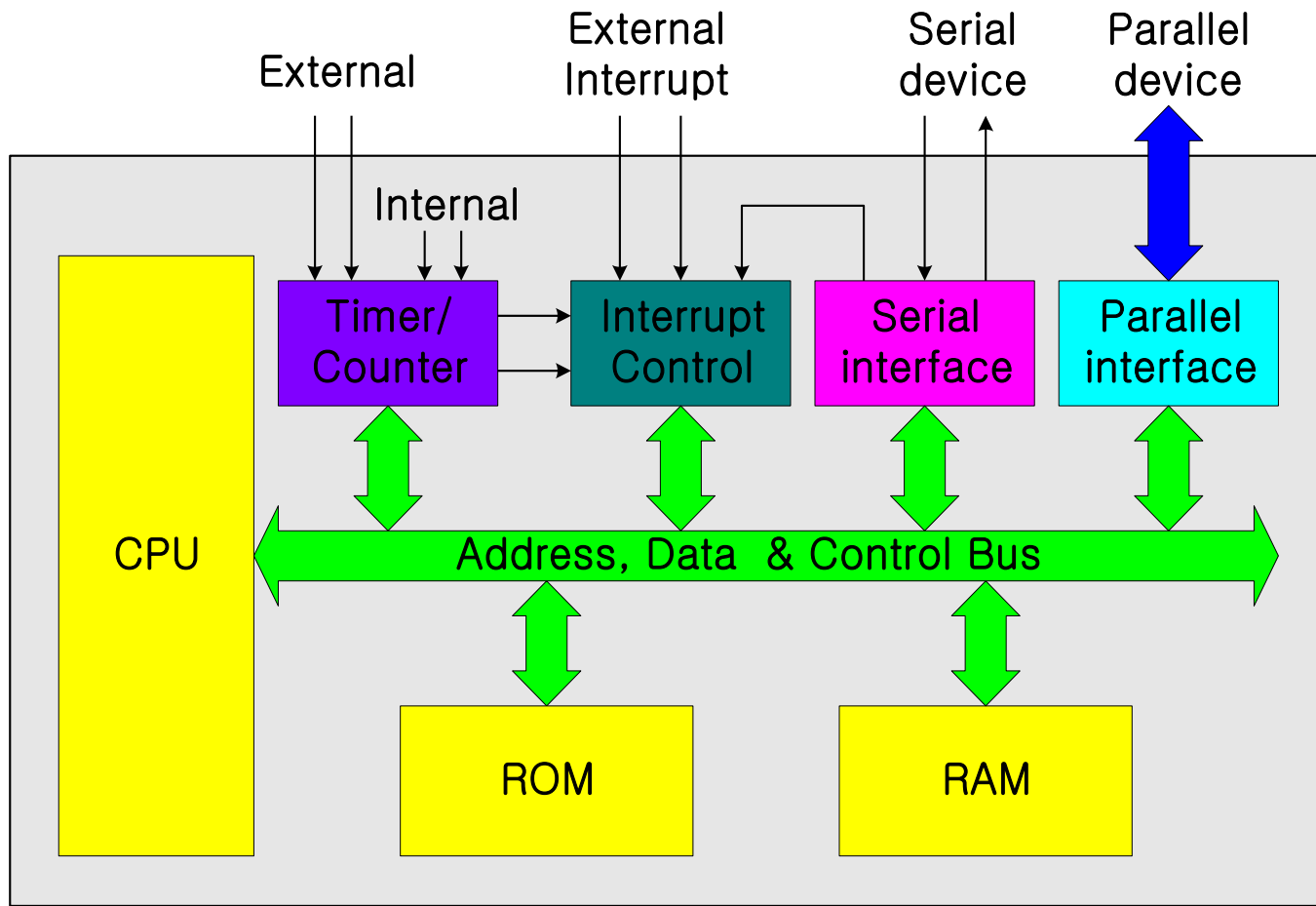
▶ 마이크로컨트롤러 (Microcontroller)

- 단일 칩 마이크로컴퓨터는 용도에 따라 연산 및 데이터의 처리를 목적으로 하는 **연산용**과 기계의 제어를 목적으로 하는 **제어용**으로 분류
- 마이크로컨트롤러 : 제어를 목적으로 하는 단일 칩 마이크로컴퓨터

▶ 마이크로컨트롤러 특징

- 마이크로컨트롤러는 단일 기계, 컴퓨터 주변장치, 통신장치, 공정 등의 제어를 목적으로 입출력 기능을 강화
- 외부 핀(pin)의 대부분을 입출력 기능에 할애하고, A/D 변환기, D/A 변환기와 PWM을 내장하기도 하며, 타이머/카운터도 내장하고 동기식/비동기식 통신포트도 내장
- 인터럽트 기능 강화
- 비트 조작 명령어(Bit Manipulation Instruction)을 강화

마이크로컨트롤러 (2)



마이크로컨트롤러 (3)

▶ 마이크로컨트롤러의 응용

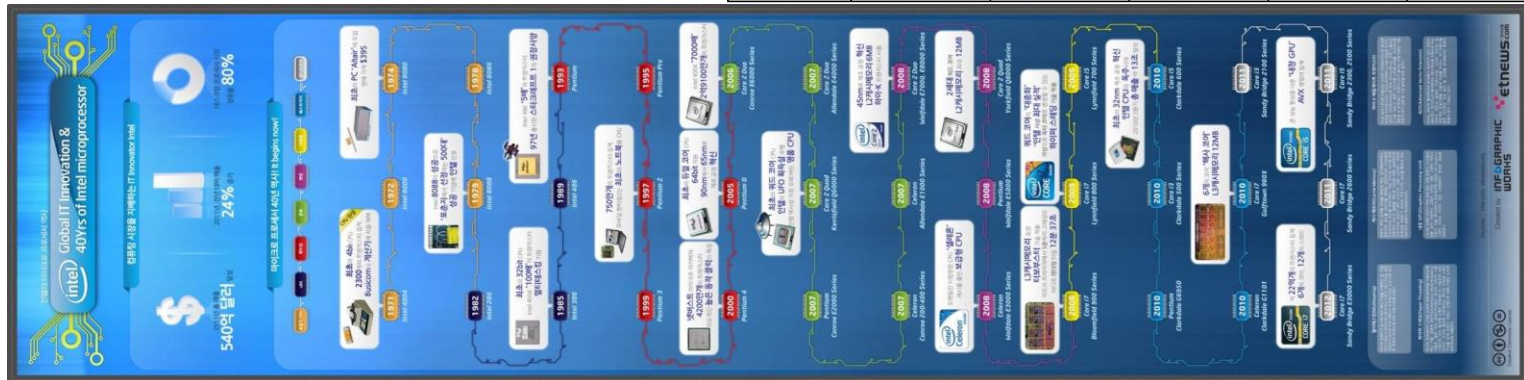
- 산업 : 모터 제어, 로봇 제어, 프로세스 제어, 수치 제어, 장난감 등
- 계측 : 의료용 계측기, 오실로스코프 등
- 가전제품 : 전자레인지, 가스오븐, 전자밥솥, 세탁기 등
- 군사 : 미사일 제어, Torpedo 제어, 우주선 유도 제어 등
- 통신 : 휴대폰, 모뎀, 유무선 전화기, 중계기 등
- 사무기기 : 복사기, 프린터, plotter, 하드디스크 구동장치 등
- 자동차 : 점화 타이밍 제어, 연료 분사 제어, 변속기 제어 등
- 생활 : 전자시계, 계산기, 게임기, 금전등록기, 온도조절기 등

마이크로프로세서 및 마이크로컨트롤러의 발달과정 (1)

[마이크로프로세서 발달과정]

80 계열	8 비트	8 비트 성능향상	16 비트	16 비트 성능향상	32 비트
Intel	8008 8080	8085	8088 8086	80188 80186 80286	80586 80486 80386
Zilog		Z80	Z8000		Z80000

68 계열	8 비트	8 비트 성능향상	16 비트	16 비트 성능향상	32 비트
Motorola	6800	6809	68000 68008	68010	68020 68030 68040 68060
MOS Technology		6502			



마이크로프로세서 및 마이크로컨트롤러의 발달과정 (2)

▶ 마이크로컨트롤러유닛(MCU)

- 마이크로컨트롤러 = CPU + RAM + ROM + 주변 장치(타이머 등)
- 마이크로프로세서로부터 분화되었음
- 최초의 마이크로컨트롤러: TI의 TMS1000(1975년)

연도	내용	비고
1971	Intel에서 최초의 마이크로프로세서인 4bit 4004 개발	
1973	Intel에서 8bit 마이크로프로세서인 8080 개발	
1975	TI에서 최초의 마이크로컨트롤러인 TMS1000 개발 Intel에서 8051 개발	
1976	Intel에서 8 bit 마이크로컨트롤러 8048(MCS-48) 개발 Motorola에서 8bit 마이크로컨트롤러 MC6801 개발	
1978	Intel에서 16bit 8086 마이크로프로세서 개발	
1980	Intel에서 8 bit 마이크로컨트롤러인 8051 (MCS-51)개발	
1982	Intel에서 16 bit 마이크로컨트롤러 인 MCS-96 개발	
1988	Intel에서 32 bit 마이크로컨트롤러 인 80960 개발	

마이크로프로세서 및 마이크로컨트롤러의 발달과정 (3)

▶ 다양한 마이크로컨트롤러의 종류

- 8051 계열
- PIC 계열 (마이크로칩)
- AVR 계열 (Atmel)

▼ 제조사별 중요 마이크로컨트롤러 제품군

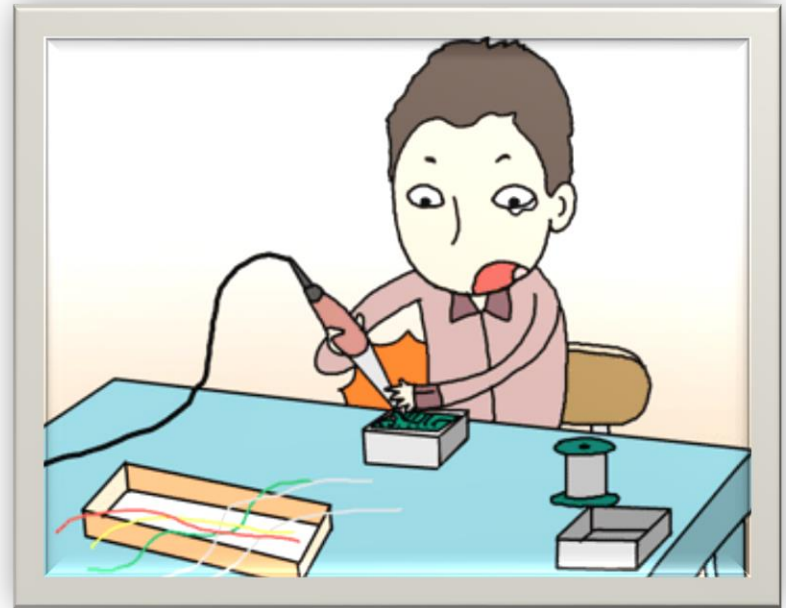
번호	제조사	모델명	비고
1	마이크로칩Microchip	PICxx계열	
2	히다치HITACHI	H8 계열	
3	MIPS	MIPS (VR7701)	
4	ARM	ARM 계열	
5	자일로그Zilog	Z80 계열	
6	인텔Intel	8051 계열, 80xxx 계열	
7	아트멜Atmel	AVR 계열	
8	삼성Samsung	KSxx 계열	
9	모토롤라Motorola	68xxx 계열	
	(이하 생략)		

하드웨어 실습 주의사항 (1)

▶ 안전 최우선 !

▶ 안전 사고 예

- 전기 인두를 켜 놓고 인두 팁을 손으로 잡아 화상 사고
- 기판 뒷면 소자 철선을 니퍼로 자르다가 튀어 눈의 손상 사고
- 물 묻은 손으로 기구나 배선을 만져 감전 사고 발생
- 소자의 극성을 반대로 하여 부품의 파손에 의한 부상



하드웨어 실습 주의사항 (2)

※ 주의 사항

- 작업을 시작하기 전에 항상 주변을 깨끗하게 청소합니다.
- 작업하기에 편리하도록 재료나 용구를 정리하는 습관을 가져야 합니다.
- 납땜 인두를 가지고 장난하면 안됩니다.
- 사용하지 않을 때에는 항상 플러그를 콘센트에서 빼놓습니다.
- 인두 받침을 사용하는 등 안전하게 거치하여 화상이나 화재가 나지 않도록 안전하게 다뤄야 합니다.
- 화상을 입었을 경우 찬물이나 얼음으로 화기를 제거하고, 적절한 치료를 합니다.
- 창문을 열어 자주 환기시킵니다.
- 사용 후 용구를 잘 정리해 둡니다.

하드웨어 실습 도구 (1)



- 사진 오른쪽부터 시계 방향으로 납땜인두, soldering paste, tip cleaner, 납, desoldering tool, desoldering wire, 철수세미, 인두 스탠드

하드웨어 실습 도구 (2)

- 솔더링 페이스트(Soldering Paste) : 납이 녹아서 표면에 잘 붙게 하는 역할을 합니다. 달궈진 인두 팁이나 잘라낸 와이어의 끝에 묻힌 후에 납을 녹여서 대면 납이 표면에 잘 달라붙습니다. 인두 팁을 청소해주는 역할도 어느 정도는 있지만 그보다는 일종의 "풀" 같은 용도로 사용.
- 팁 클리너(Tip cleaner) : 납땀을 계속하다보면 인두 끝에 검은 그을음이 생기게 되는데 이걸 제거하는데 사용합니다. 없다면 가격이 훨씬 싼 철 수세미로 대체 가능합니다. 달궈진 인두끝을 수세미에 살짝 문지르면 검은 그을음이 말끔히 제거됩니다.
- 디솔더링 툴(Desoldering tool) : 이건 납땀이 되어있는 상태에서 납을 제거하는 역할을 합니다. 주사기와 비슷한 원리로서 단추를 누르면 손잡이가 튀어나오면서 공기의 흡입력으로 납을 빨아들이는 구조입니다.
- 디솔더링 와이어(Desoldering wire) : 역시 납을 제거하는 용도인데, 먼저 많은 양의 납은 위의 흡착기로 제거를 한 후에 표면에 남아있는 적은 양의 납은 이 와이어와 인두를 사용해서 빨아들이면 됩니다. 남아있는 납 위에 이 와이어를 대고 그 위를 인두로 달구면 납이 와이어로 흡수됩니다.
- 인두 스탠드 : 달궈진 인두를 세워놓는 용도로 사용되는 물건입니다.