마이크로 컨트롤러 개론

컴플라자

6월 8일, 2005년

목차

- 프로세서의 역사
- 프로세서의 분류
- 마이크로 컨트롤러 (MCU)
- 개발 방법 & Tool
- 마이크로 컨트롤러의 종류
- 마이크로 컨트롤러의 구조
- 마이크로 컨트롤러를 이용한 개발
- 마이크로 컨트롤러를 공부하려면…

프로세서의 역사

■ 전자계산기의 발달

초창기의 컴퓨터는 부피가 크고, 신뢰성이 다소 떨어졌으나 CPU의 개발로 높은 신뢰도와 빠른 연산을 제공하는 컴퓨터가 개발되고 있으며 점차 컴퓨터 크기의 소형화가 이루어지고 있음

구분	내용
MARK-1	1944년 하버드 대학의 에이컨(Aiken) 교수에 의해 제작된 세계 최초의 자동 기계식 계산기
ENIAC	■1946년 에커트와 머큘리 교수에 의해 개발된 세계 최초의 전자계산기 ■진공관 18,800개로 구성되었으며 프로그램 외장방식이 사용됨 ■진공관의 열 발생률이 높고 잦은 고장이 많이 발생했음
EDSAC	■1949년 영국 켐브리지 대학의 윌키스 교수에 의해 개발 ■프로그램 내장방식을 채용한 세계 최초의 전자계산기
EDVAC	1951년 프로그램 내장방식을 완성한 전자계산기
UNIVAC-1	1951년 에커트와 머큘리 교수에 의해 개발된 최초의 사무용 전자계산기로 프로그램 내장방식을 사용

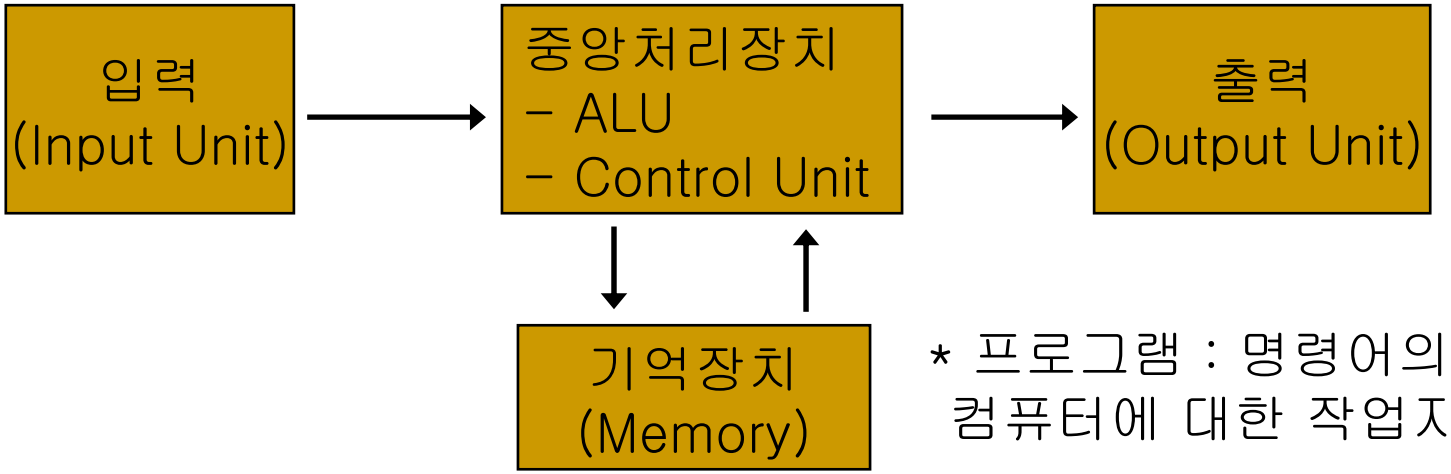
프로세서의 역사

- 프로그램 내장형 컴퓨터

* 1945년 폰 노이만(Von Neumann)
  - 프로그램 내장형 컴퓨터 제창

* 1949년 영국 캠브리지 대학 교수 윌키스(M. Wilkes)
  - EDSAC(Electronic Delayed Storage Automatic Calculator)

→ 세계 최초의 프로그램 내장 방식 컴퓨터



* 프로그램 : 명령어의 집합,
컴퓨터에 대한 작업지시서

프로세서의 역사

- 마이크로 프로세서의 발달

	제1세대 (1951년-1959년)	제2세대 (1959년-1963년)	제3세대 (1963년-1975년)	제4세대 (1975년 이후)
기억소자	진공관(Tube)	트랜지스터	집적회로(IC)	고밀도 집적회로 (LSI), 초고밀도 집적회로(VLSI)
주기억장치	자기드럼	자기코어	집적회로(IC)	LSI, VLSI
처리속도	ms(10^{-3})	μs(10^{-6})	ns(10^{-9})	ps(10^{-12})
특징	▪하드웨어 중심 ▪전력소모가 만고 신뢰성이 낮음 ▪대형화 ▪과학계산 및 통계 처리용으로 사용	▪소프트웨어 중심 ▪운영체제(OS) 개발 ▪전력소모 감소 ▪신뢰도 향상,소형화 ▪온라인 방식 도입	▪기억용량 증대 ▪시분할 처리 ▪다중처리 방식 ▪MIS 도입 ▪마이크로프로세서 탄생	▪전문가 시스템 ▪인공 지능(AI) ▪종합정보 통신망 ▪마이크로 컴퓨터
사용언어	저급 언어 (기계어, 어셈블리어)	고급 언어 (FORTRAN, ALGOL, COBOL)	고급 언어 (LISP, PASCAL, BASIC, PL/I)	문제지향적 언어

프로세서의 역사

- 마이크로 프로세서의 발달
  - Intel 의 마이크로 프로세서
    - 1971년 4004, 4bit, 0.5 MHz clock, 16 Kbyte Main Memory
    - 1973년 8080, 8bit
    - 1975년 8085, 8bit
    - 1978년 8086, 16bit
    - 1982년 80286, 16bit
    - 1985년 80386, 32bit
    - 1989년 80486, 32bit
    - 1990년대 80586(Pentium), 32bit
  - 기타 마이크로 프로세서 제조업체
    - Motorola, 680계열(68000, 68020…), 주로 산업용으로 사용됨.
    - Zilog, Z80, Intel의 8085와 유사한 8비트 프로세서

프로세서의 분류

- 아키텍처에 의한 분류
  - CISC
    - 명령어를 고속으로 수행할 수 있는 특수목적회로를 가지고 있으며, 많은 명령어들을 프로그래머에게 제공해 주므로 프로그래머의 작업이 쉽게 이루어짐
    - 구조가 복잡하므로 생산단가가 비싸고 전력소모가 큼
    - RISC 보다 많은 명령어 집합, 실행 속도 늦음
  - RISC
    - 전력소모가 적고 CISC구조보다 처리속도가 빠름
    - 필수적인 명령어들만 제공되므로 CISC구조보다 덜 복잡하고 생산단가가 낮음
    - 복잡한 연산을 수행하기 위해서는 RICS가 제공하는 명령어들을 반복 수행해야 하므로 프로그래머의 작업이 복잡한 단점이 있음
    - 축소된 명령어 집합, 많은 레지스터, 파이프라인, 실행 속도 향상

프로세서의 분류

- 처리 용량에 따른 분류

  프로세서가 한번에 처리할 수 있는 비트 수로 분류

  - 8-Bit 프로세서
    - 1970년대 마이크로 프로세서
    - 현재는 소용량 마이크로 컨트롤러가 8-Bit 프로세서이다.
      - 8051, AVR, PIC, SAM88
  - 16-Bit 프로세서
    - 1980년대 초 마이크로 프로세서
    - 현재는 고속의 마이크로 컨트롤러로 사용된다. 8-Bit 와 32-Bit 의 발달로 많이 사용하지 않는다.
      - AM188, MSP430
  - 32-Bit 프로세서
    - 1980년대 중반 이후 현재까지 마이크로 프로세서
    - PC, 또는 대용량의 데이터 처리를 하는 산업용 장치에 쓰인다.

프로세서의 분류

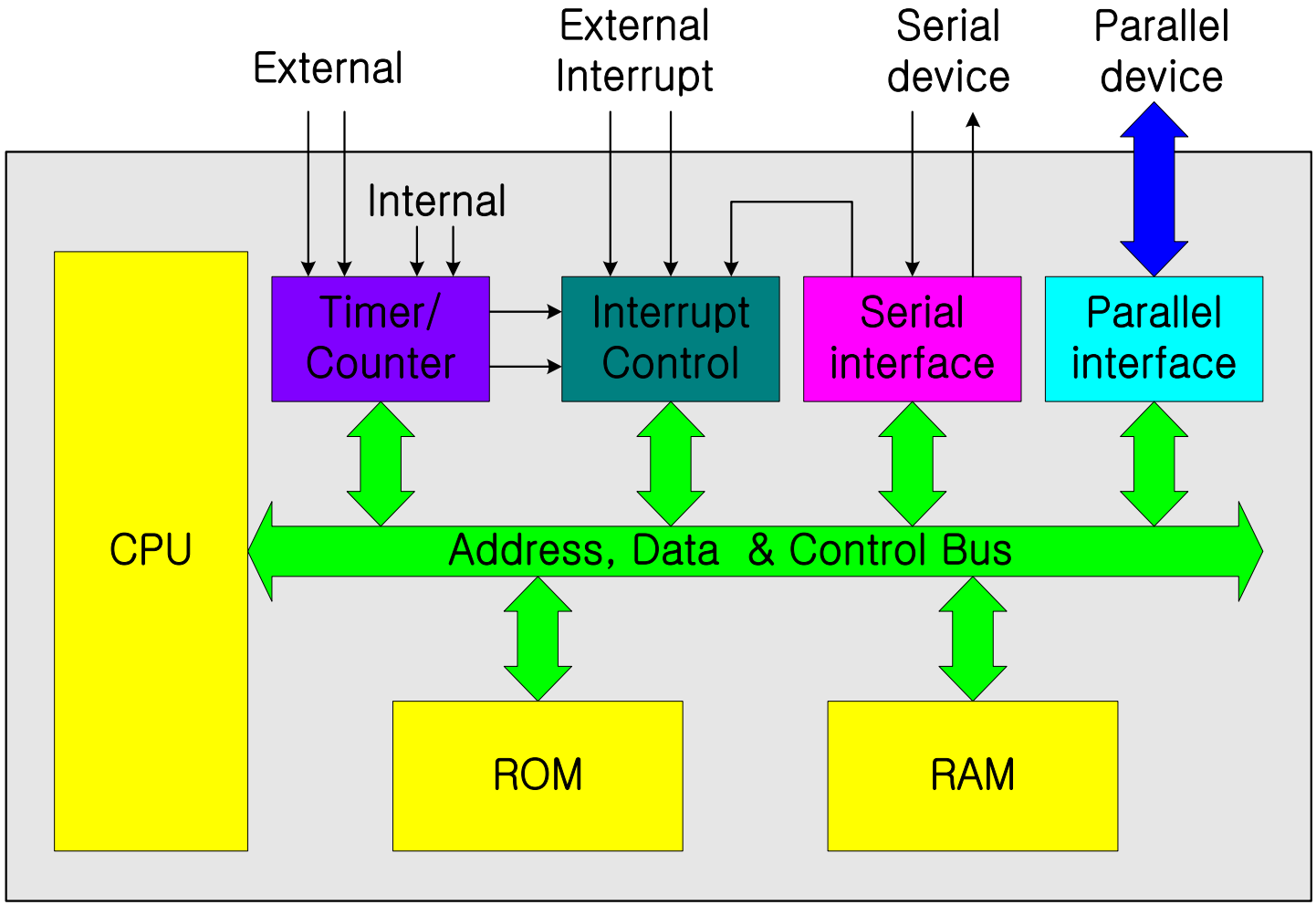
- 용도에 따른 분류
  - MPU – Micro Processor Unit
    - 중앙처리장치에서 주기억장치를 제외한 연산장치, 제어장치 및 각종 레지스터들을 단지 1개의 IC 소자에 집적시킨 것
    - 용도 : PC
  - MCU – Micro Controller Unit
    - 마이크로프로세서 중에 1개의 칩 내에 CPU 기능은 물론이고 일정한 용량의 메모리(ROM, RAM 등)와 입출력 제어 인터페이스 회로까지를 내장한 것
    - MCU, 단일 칩 마이크로컴퓨터, 마이컴 등으로 불린다.
  - DSP – Digital Signal Processor
    - 디지털 신호를 하드웨어적으로 처리할 수 있는 집적회로
    - 용도 : 영상, 음성 등 신호처리

프로세서의 분류

- 마이크로 프로세서와 마이크로 컨트롤러의 비교

마이크로 프로세서	마이크로 컨트롤러
데이터 전송 연산 저장 제어부	데이터 전송 연산 저장 제어부
메모리 없음 외부 메모리 설계 필요	메모리 포함 별도의 외부 메모리 설계가 가능
입출력 포트 없음 입출력 인터페이스 필요	입출력 포트 포함

마이크로컨트롤러 (MCU)



마이크로컨트롤러 (MCU)

- 마이크로컨트롤러(MCU - Micro Controller Unit)
  - CPU, ROM, RAM, I/O Port(직렬,병렬), Timer/Counter, Interrupt 처리기가 하나의 반도체 chip에 집적
  - 기계의 제어를 목적으로 하는 제어용 (마이크로프로세서 : 연산 및 데이터 처리를 목적)
  - 외부사건(Interrupt)에 실시간 응답해야 하는 분야에 많이 사용
  - 입출력 인터페이스는 단일 bit로도 가능 (모터, LED, 스피커 구동 등)

마이크로컨트롤러 (MCU)

- 마이크로컨트롤러(MCU)의 특징
  - 주변장치들을 센싱 및 제어하기 위한 I/O 능력이 강화
  - 타이머/카운터, 통신포트 내장 및 인터럽트 처리 능력 보유
  - Bit 조작 능력이 강화
  - 제품의 소형화 및 경량화
  - 제품의 가격이 저렴(부품비, 제작비, 개발비 및 개발시간 절감)
  - 융통성 및 확장성이 용이(프로그램만 변경)
  - 신뢰성이 향상(부품 수 적어 시스템 단순, 고장율 적고, 보수편리)

마이크로컨트롤러 (MCU)

- 마이크로컨트롤러(MCU)의 응용분야

  - 산업 : 모터 제어, 로봇 제어, 프로세스 제어등
  - 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

마이크로컨트롤러 (MCU)

- MCU(Micro Controller Unit)와 제조사
  - Intel : 8051, 80196
  - Atmel : AVR (8bit RISC M/C)
  - Microchip : PIC16/17 (RISC, A/D변환기 및 PWM 내장)
  - Motorola : MC6805, MC68HC11, MC68HC16, MC68332
  - Samsung : KS51, KS88, KS16, KS32
  - Zilog : Super-8
  - AMD : AM188 (X86 호환)
  - ARM : ARM7, ARM9

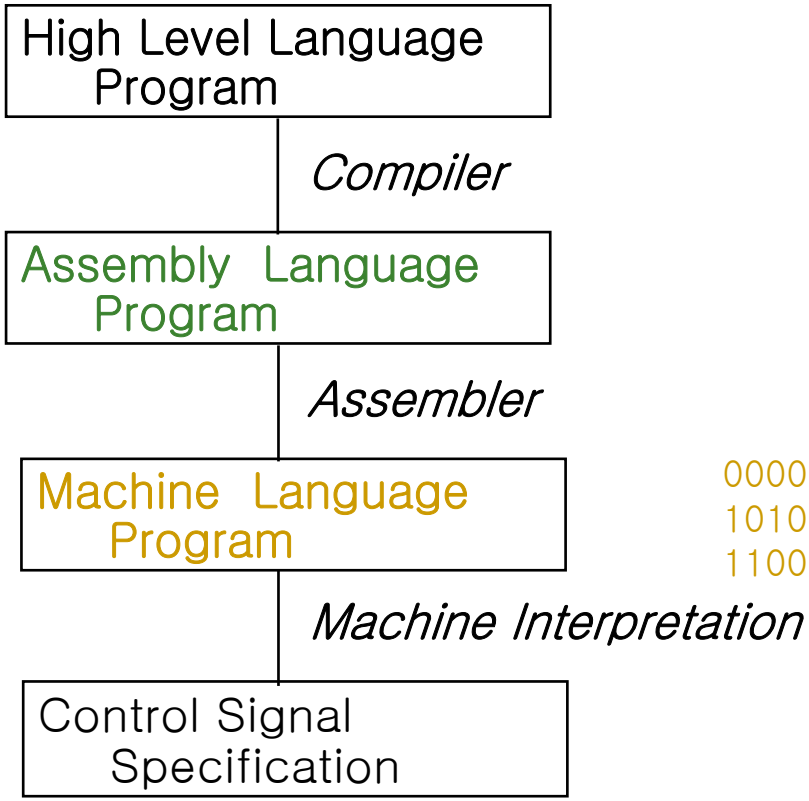
개발 방법 & Tool

- 컴파일러 및 S/W 개발환경
  - 고급언어
    - C언어
      - 대부분의 MCU에 사용 가능하다.
      - 포인터를 이용하여 메모리 엑세스를 할 수 있다.
      - Compiler에 의해 2진 형태의 기계어로 변환된다.
  - 저급언어
    - 어셈블리어(Assembly Language)
      - 기계어를 사람이 볼 수 있게 나타낸 코드.
      - Assembler에 의해 2진 형태의 기계어로 변환

* Firmware : 마이크로 컨트롤러를 제어하기 위한 프로그램.
ROM에 견고하게 저장되기 때문에 Firmware 라 불린다.

개발 방법 & Tool

- Firmware 프로그래밍




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw $15,  0($2)
lw $16,  4($2)
sw $16,  0($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
```

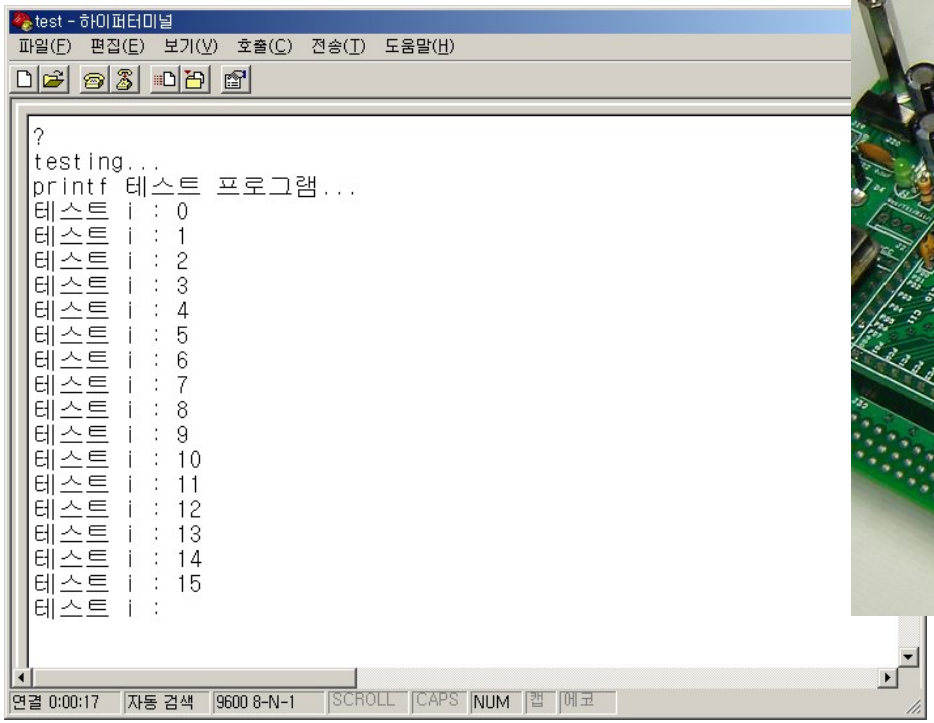
ALUOP[0:3] <= InstReg[9:11] & MASK

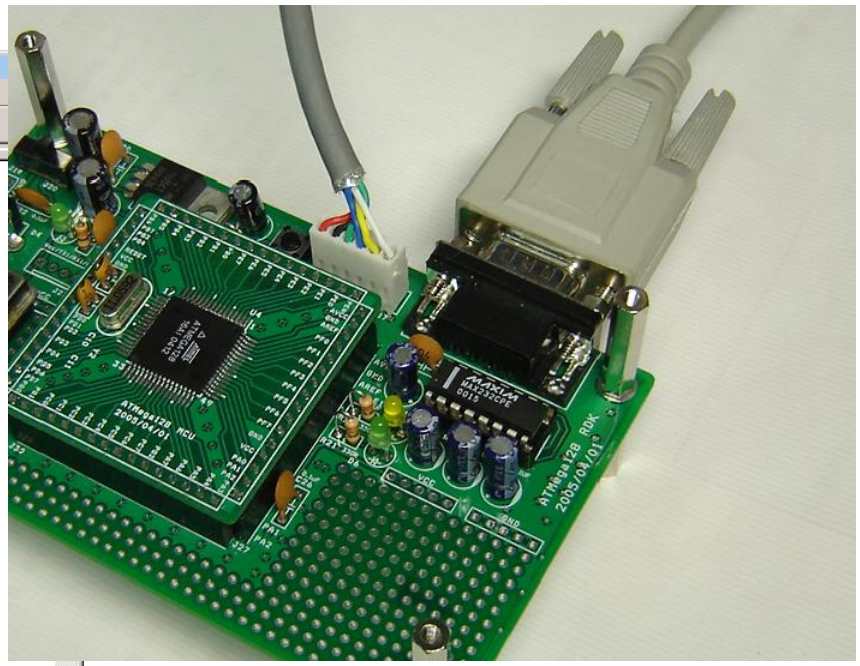
개발 방법 & Tool

- Emulator 사용
  - 보드 내에 MCU 자리에 장착되어 MCU 의 동작을 시뮬레이션 해 볼 수 있어 개발을 빠르게 할 수 있다.
  - 디버그 툴을 제공하여 효율적으로 디버깅을 할 수 있다.
  - 각 MCU 제조 회사에서 Emulator를 판매하지만 장비가 고가이다.
  - 회사에서 주로 사용한다.
- Rom Writer 사용
  - Rom Writer 라는 장비를 이용하여 MCU의 내부 롬에 프로그램을 하거나 외부 롬에 프로그램을 하여 테스트 해볼 수 있다.
  - 롬을 Write 하며 테스트 하기가 불편하다.
- ISP (In System Programming) 사용
  - 프로그램을 MCU에 다운로드하여 동작을 확인할 수 있다.
  - 가격이 저렴하다.
  - 지원하는 MCU가 적다. (AVR, 일부 8051, PIC가 지원한다.)

개발 방법 & Tool

- 시리얼 통신을 이용한 디버깅
  - 시리얼 통신이 지원될 경우 PC에서 printf 와 같은 함수를 이용하여 디버깅을 할 수 있다.



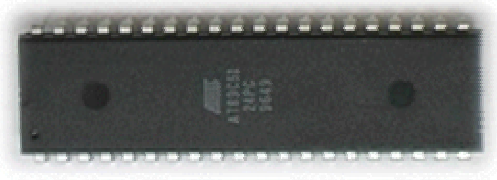


마이크로 컨트롤러 선정

- Performance
  - 속도 (Clock Speed, Interface Speed)
- Capability
  - 용량 (RAM, ROM), 주변장치(IO Port, Timer, ADC 등)
- C 언어 컴파일러 지원여부
- 개발장비 지원여부
  - Emulator
  - ISP 기능 지원여부

MCS-51

- 특징
  - 1980년 Intel에서 출시되어 지금까지 가장 많이 사용되는 MCU이다.
  - X86 과 내부 구조 비슷하여 교육용으로 적합하다.
  - 총 256 Byte 메모리 (SFR 128Byte, RAM 128Byte)
  - 8비트 단위의 4개의 입출력포트(포트 0, 1, 2, 3)
  - 2개의 16비트 타이머/카운터, 시리얼 포트
  - 4K Byte의 프로그램 메모리를 내장하고 있다.
  - 프로그램 메모리, 데이터 메모리를 각각 최대 64K 바이트까지 확장 가능
  - 일부 모델이 ISP 지원 가능



- MCU 종류
  - 80C51, 80C52 – Intel 사
  - AT89C51, AT89C52, AT89S52 (ISP 기능 지원) – ATMEL 사

MCS-51

- 제조사
  - Intel, Atmel, Hynix, Philips, TI, ISSI 등 많은 반도체 회사에서 생산한다.

- C 언어 컴파일러
  - Keil
    - 통합 개발 환경 (IDE) 이 잘되어 있다.
    - http://www.keil.com/
  - SDCC
    - 무료로 사용할 수 있다.
    - 통합 개발 환경을 지원하지 않는다.
    - http://sdcc.sourceforge.net/
  - IAR
  - ETC

AVR

- 특징
  - 8-비트 RISC구조로 명령어가 간단하며 동작 속도가 빠르다
  - 최대 약 16MIPS (Million Instruction Per Second)의 성능을 보인다.
  - 소비 전력이 적다.
  - 다른 마이크로 컨트롤러에 비해 대용량의 SRAM을 가지고 있다. (ATMega128 의 경우 4Kbyte)
  - ISP 기능을 지원하며 Flash memory의 내장하여 프로그래밍이 용이하다.
  - 10 비트 ADC 내장하고 있다.(일부 패밀리)
  - C언어에 최적화된 설계 (실제로 칩 설계와 동시에 C 컴파일러 설계하였다고 한다.)
  - UART, PWM(Pulse Width Modulation) 등을 내장하고 있다.

- 패밀리 종류
  - Tiny 시리즈 : RAM이 없거나 적은 모델이 대부분이며 핀 수 또한 적어 간단한 어플리케이션에 적합하다.
  - Mega 시리즈 : 플레시 메모리와 램의 용량이 크고 핀 수 또한 많아서 복잡한 어플리케이션에 적합하다.

AVR

- C 언어 컴파일러
  - AVR-GCC
    - GCC 를 AVR 로 포팅한 무료 컴파일러
  - ICC-AVR
    - 통합 개발 환경 (IDE) 이 잘되어 있다.
    - ISP Cable 지원하여 자동으로 다운로드하는 기능이 있다.
  - CodeVision AVR C
  - ETC
- 참고 사이트
  - http://www.atmel.com/
  - http://micro.new21.org/avr/
  - http://www.avrfreaks.net/
  - http://www.imagecraft.com/software/

PIC

- 특징
  - OTP (One Time Programmable) 타입으로 초저가 이며, 소량부터 대량 생산에 최적이다.
  - 병렬처리 RISC 구조이고 하버드 구조이다.
  - 다양한 종류의 IC 가 있어 크기와 타입이 다양하다.
  - 개발 장비가 저가형으로 판매, 널리 퍼져있다.
  - 일부 모델이 ISP 지원 가능 (PIC 에서는 ICE 라 불림)
  - 회사에서 많이 사용된다.

- 패밀리 종류
  - PIC17CXXX 8비트 High-Performance
  - PIC16CXXX 8비트 Mid-Range
  - PIC12CXXX 8비트 Low-Performance

PIC

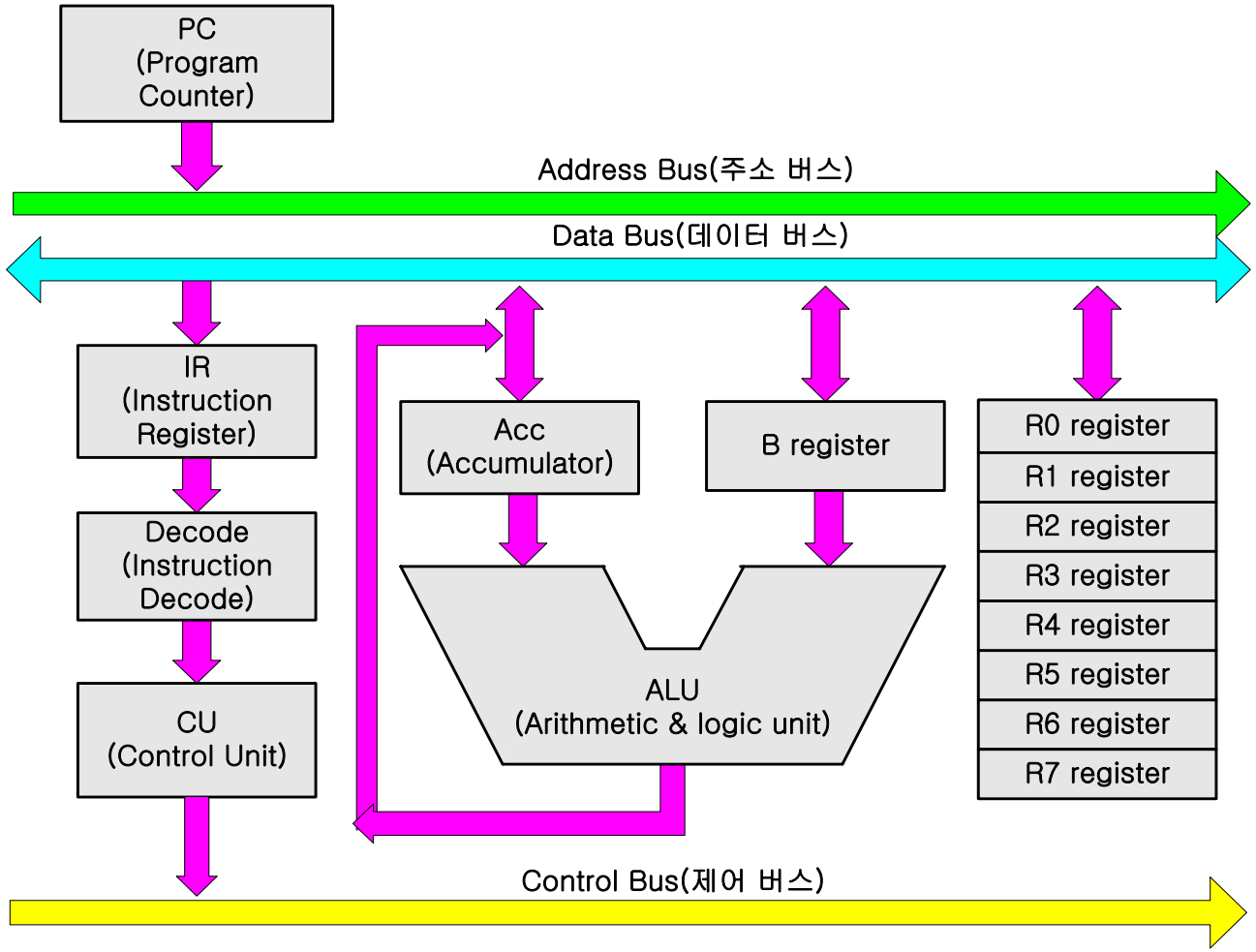
- C 언어 컴파일러
  - CCS-C
  - Hitech-C
  - ETC

- 참고 사이트
  - http://www.microchip.com/
  - http://micro.new21.org/pic/
  - http://www.htsoft.com/

마이크로 컨트롤러의 구조

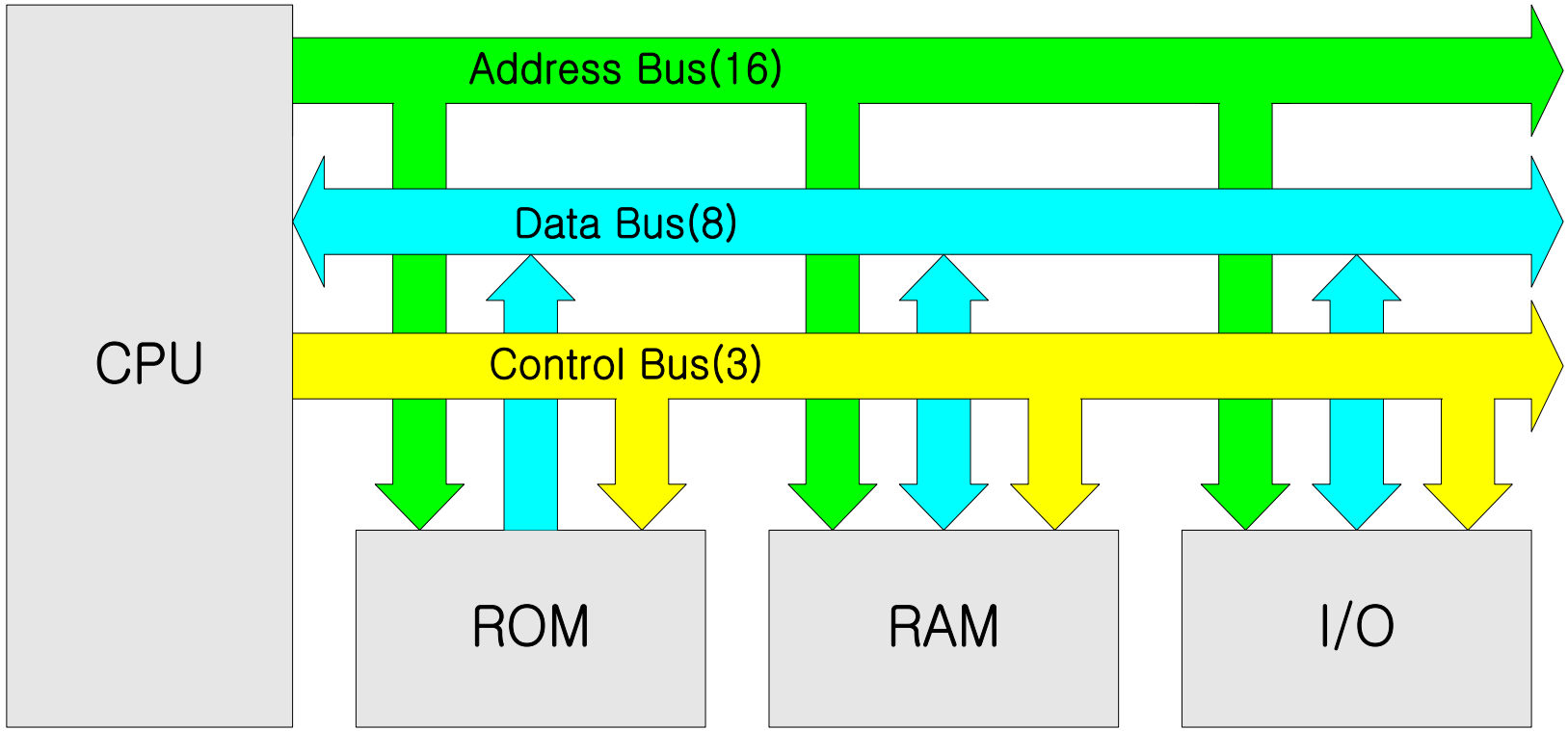
- MCU의 내부 구조
- Address/Data Bus 개념
- Memory 구조와 Access 방법
- I/O 방식
- SFR (Special Function Register)
- Interrupt 개념

MCU의 내부 구조


PC
(Program
Counter)
Address Bus(주소 버스)
Data Bus(데이터 버스)
IR
(Instruction
Register)
Decode
(Instruction
Decode)
CU
(Control Unit)
Acc
(Accumulator)
B register
ALU
(Arithmetic & logic unit)
R0 register
R1 register
R2 register
R3 register
R4 register
R5 register
R6 register
R7 register
Control Bus(제어 버스)

Address/Data Bus

- 외부의 Address/Data Bus 의 연결



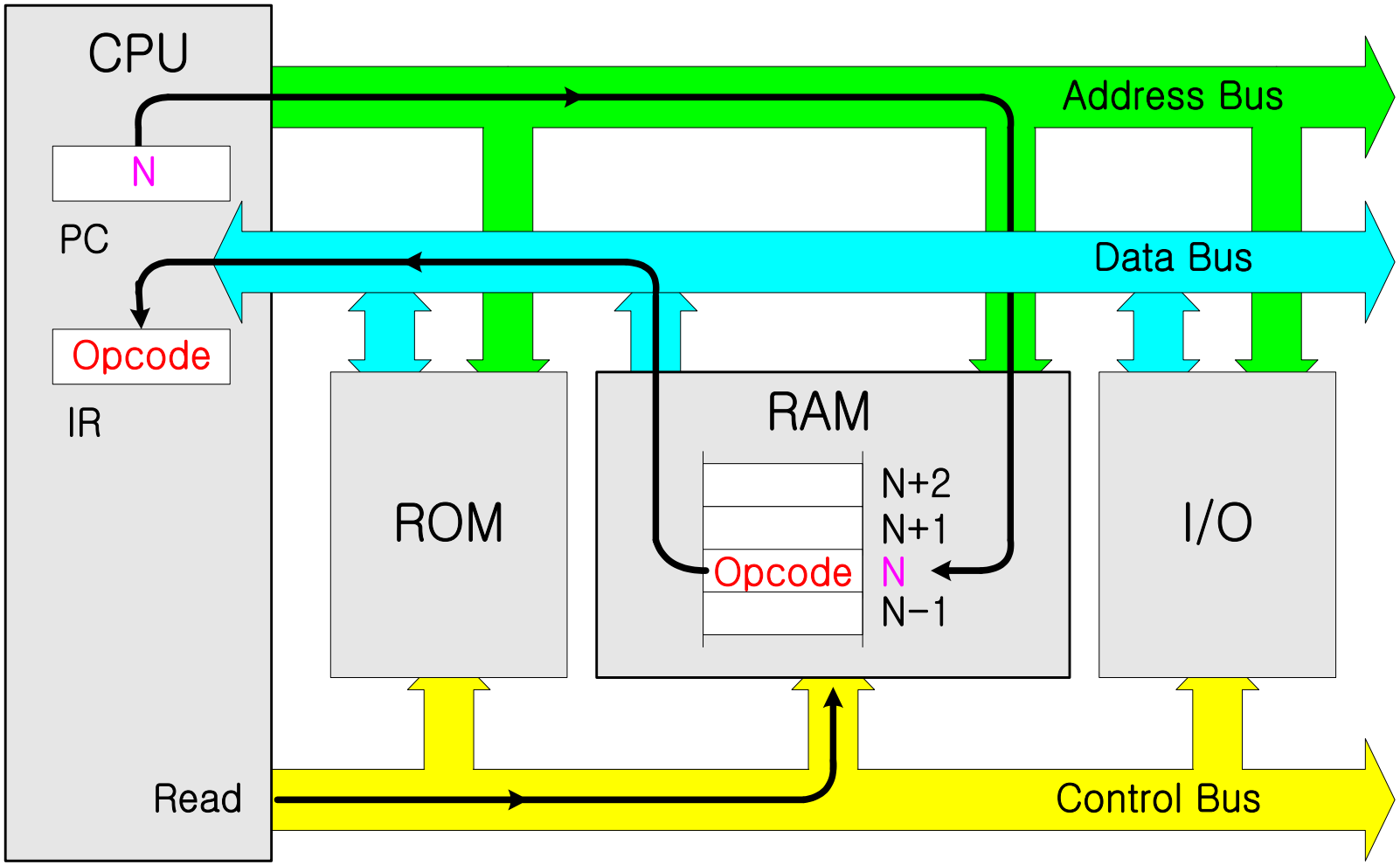
ROM(Read Only Memory)

- 읽기만 가능, 비휘발성
  - Mask ROM : 공장에서 매스킹되어 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 쪼여 내용을 지움
  - EEPROM(Electrically Erasable Programmable ROM) : 전기적으로 지우며, ROM Writer 없이 프로그램
  - Flash ROM : EEPROM과 같은 기능, EEPROM 보다 적은 TR 사용

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 사용

RAM 데이터 읽기 과정


CPU
N
PC
Opcode
IR
Read
Address Bus
Data Bus
Control Bus
ROM
RAM
N+2
N+1
Opcode
N
N−1
I/O

I/O 방식

- 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/O 보다 간단한 구조로 되어있다.
  - I/O 동작을 할 때 훨씬 더 융통성이 있다
  - 대부분의 CPU 에서 지원한다.

- Isolated I/O
  - I/O 명령이 Memory Access 와 분리되어 있다.

SFR

- SFR (Special Function Register) 이란 ?
  - MCU에 특정 기능으로 지정되어 있는 레지스터이다. (레지스터는 MCU 내의 임시 기억 장소)
  - 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.)
  - MCU 가 어떤 동작을 하기 위해서는 이 SFR 에 특정 데이터를 쓰거나 데이터를 읽는다.

SFR

- Example
  - PORT1에 연결되어 있는 8개의 LED 를 켜기
  - PORT1 에 5V 또는 0V 의 출력을 주면 연결되어 있는 LED 가 켜지거나 꺼지게 됩니다. SFR에 PORT1으로 데이터를 출력할 수 있게 하는 P1 이라는 레지스터가 있다.

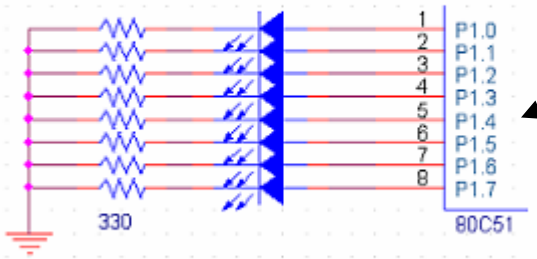
C code

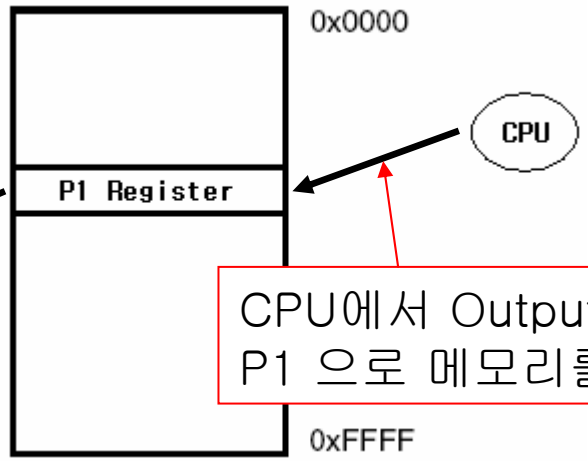
```
P1 = 0x55;
```

Assembley code

```
MOV P1, 0x55
```

Output Register 에 써진 값이 하드웨어 출력으로 나간다.





CPU에서 Output Register P1 으로 메모리를 Write 한다

SFR

- I/O 동작 시키기
  - SFR 은 MCU 의 내부 RAM에 있는 특정 메모리 영역이다.
  - 8051 의 포트 출력 동작은 이 SFR 중 포트 출력으로 설정되어 있는 P1 이라는 레지스터에 출력할 데이터를 써주는 동작이다.

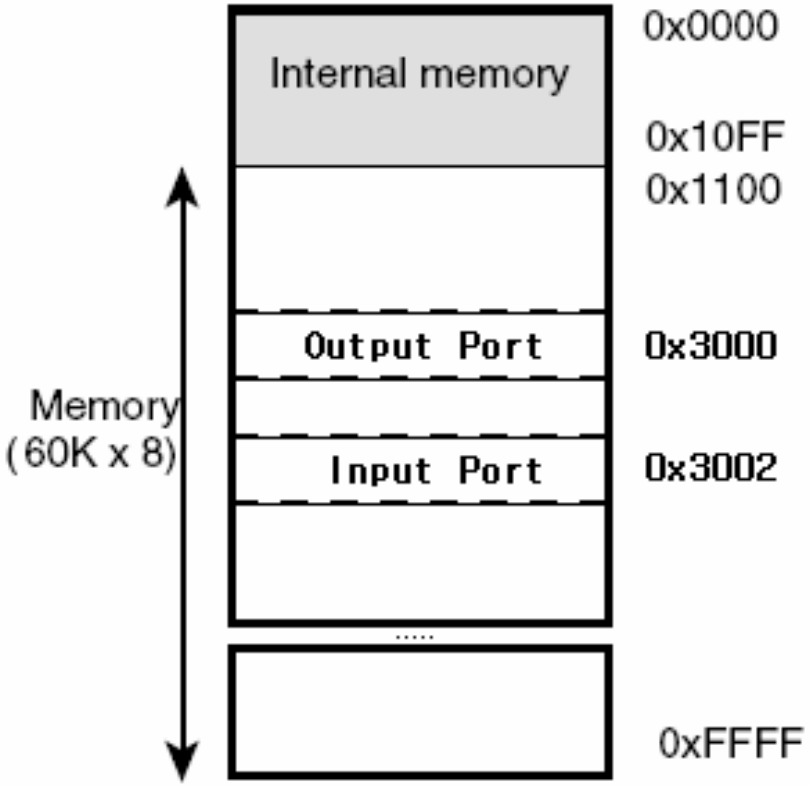
Memory Mapped I/O

메모리와 I/O 의 개념

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

앞에서 봤던 4가지 개념은
MCU가 메모리와 I/O 를 어떻게
동작 시키는가로 연결된다.

오른쪽 그림의 메모리 맵을 보면
총 0xFFFF 메모리 중 하위 0x10FF는
내부 메모리가 사용하고 있고
0x3000 번지는 Output Port 가
0x3002 번지는 Input Port 로 맵핑이 되어 있다.



Interrupt

- 인터럽트란 ?
  - 인터럽트란 주프로그램 수행 중에 주프로그램을 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 확인한다.
    - 방법 2 입력이 있을 경우 CPU 에 알려준다.

Interrupt

- 인터럽트 처리 순서
  - 인터럽트가 발생하면 주프로그램 중단.
  - 현재 명령어의 수행을 마친다.
  - 다음 수행할 명령의 주소를 메모리에 저장한다.
  - 인터럽트가 더 이상 받아들여지지 않는다.
  - ISR이 저장되어 있는 주소를 불러온다.
  - ISR이 수행된다.
  - 저장했던 주프로그램 명령의 주소를 읽어온다.
  - 전에 수행 중이던 주 프로그램으로 복귀한다.

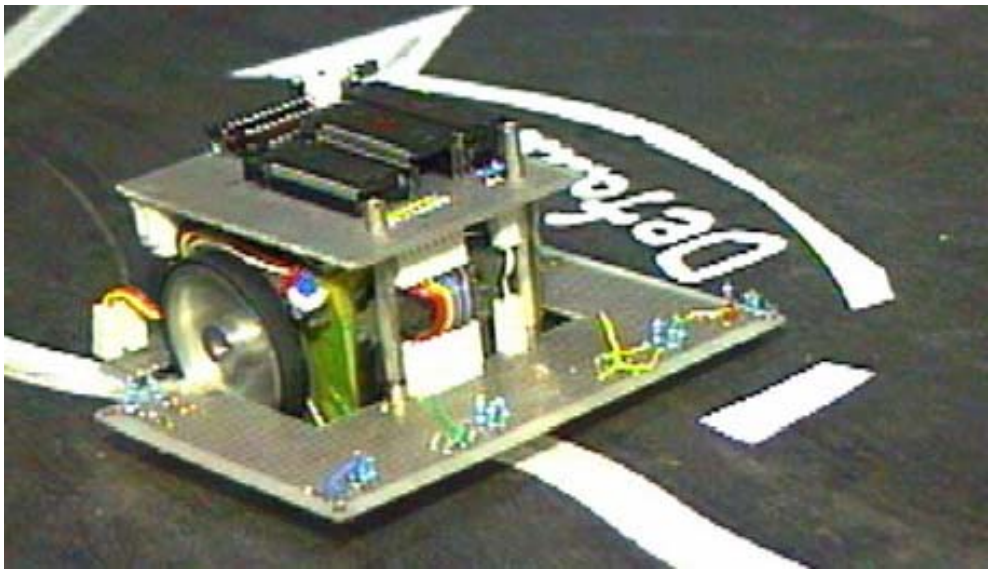
마이크로 컨트롤러 응용

- 마이크로 컨트롤러를 이용한 개발 순서
  - 시스템의 속도와 필요한 메모리 용량을 고려하여 적합한 CPU를 선정한다.
  - 시스템의 입력과 출력을 정의한다.
  - 시스템의 동작을 정의한다.
  - 사용되는 모듈을 분석한다. (데이터 수집)
  - 각 모듈별로 독립적 (또는 연계하여) 으로 구성하여 테스트를 한다.
  - 완성된 각 모듈을 연결한다.
  - 디버깅
  - 작품 (제품) 개발 완료 !!!

라인트레이서

- 라인트레이서 (AGV : Automatic Guided Vehicle)
  - 라인트레이서는 정해진 주행선을 따라 움직이는 자율이동로봇이다. 현재 공장자동화 부분에서 이용되고 있는 무인 반송차가 라인트레이서이다.
  - 라인트레이서의 기본적인 원리는 주어진 주행선을 센서로 검출하여 이것에 따라 목적 위치 까지 이동하는 것이다.

라인트레이서는
검은색 바탕 위의 흰 선을
따라서 주행 한다.



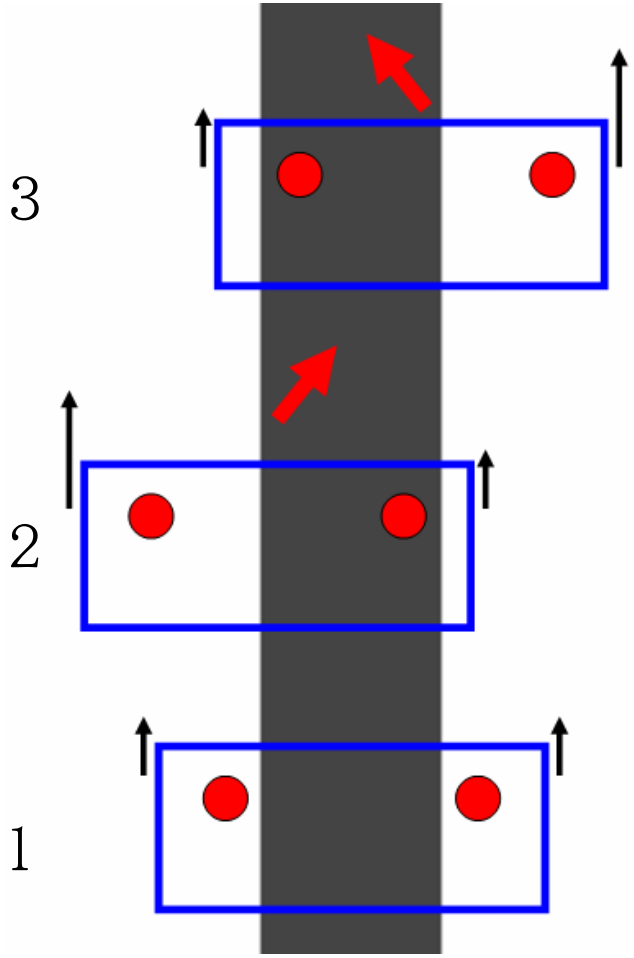
라인트레이서

- 개발 과정
  - CPU 선정
    - 기본적인 라인트레이서를 동작에는 고성능의 CPU 를 필요로 하지 않는다. 메모리도 많이 필요하지 않다.
  - 입력과 출력을 정의
    - 입력 : Line 을 감지할 수 있는 센서 입력 (2개)
    - 출력 : 바퀴를 돌리기 위하여 모터제어 출력
  - 시스템의 동작 정의
    - 오른쪽 센서에 라인이 감지되었을 경우 왼쪽 바퀴의 속도를 증가시켜 라인트레이서 차체를 오른쪽으로 회전한다.
    - 왼쪽 센서에 라인이 감지되었을 경우 반대로 오른쪽 바퀴의 속도를 증가시켜 라인트레이서 차체를 왼쪽으로 회전한다.
    - 다음장에서 계속…

라인트레이서

- 라인트레이서 동작
  - 1 번 상황에서 센서는 라인을 감지 하지 못하였으므로 직진한다.
  - 2 번 상황에서 오른쪽 센서가 인식 되었으므로 왼쪽 바퀴의 속도를 증가시켜 라인트레이서를 오른쪽으로 향하게 한다.
  - 3 번 상황에서 왼쪽 센서가 인식 되었으므로 오른쪽 바퀴의 속도를 증가시켜 라인트레이서를 왼쪽으로 향하게 한다.
  - 위의 동작을 반복하면 라인트레이서는 라인을 따라 진행하게 된다.

(빨강색 점이 센서이다.)



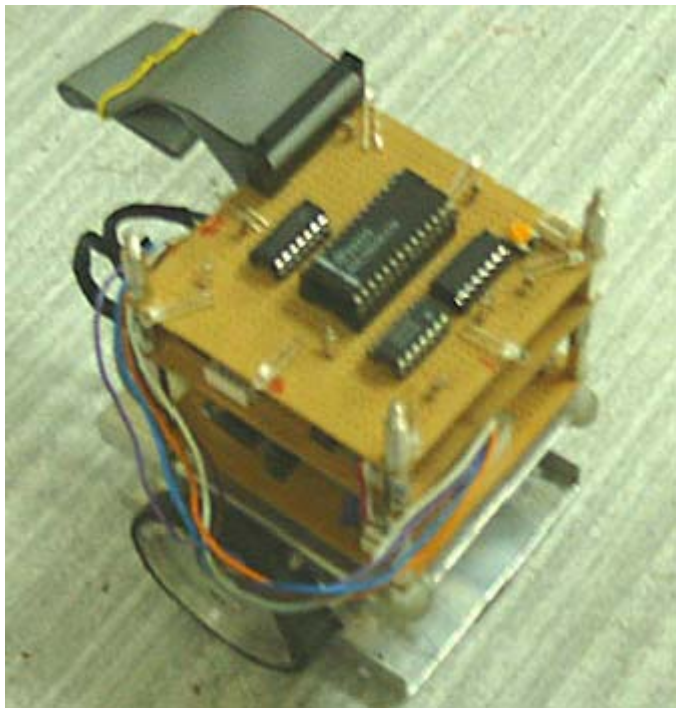
라인트레이서

- 개발 과정
  - 각 모듈별 분석
    - Line 을 감지할 수 있는 센서부
      - 적외선 센서를 사용한다.
      - 센서의 데이터는 0,1 Level 로 인식할 것인지 ADC 로 디지털로 변환해서 인식할 것인지 결정한다.
      - 센서 인식을 위해 센서 동작 테스트가 필요하다.
    - 연산 처리부
      - 선정된 MCU 로 처리한다.
      - MCU 에 관한 학습이 필요하다.
    - 모터 구동부
      - 모터는 제어하기 간편하고 많이 사용되는 스테핑 모터를 사용한다.
      - 스테핑 모터 구동을 위한 회로를 구성하고 테스트가 필요하다.

광량 추적 로봇

- 광량 추적 로봇
  - 상단에 장착된 광량센서를 이용하여 가장 빛이 밝은 방향으로 이동하는 로봇

외부 빛을 감지하거나 라이터 불과 같은 열을 발생하는 물체를 따라간다.



광량 추적 로봇

- 개발 과정
  - CPU 선정
    - 동작이 빠르지 않아 고성능의 CPU 를 필요로 하지 않는다. 메모리도 많이 필요하지 않다.
  - 입력과 출력을 정의
    - 입력 : 광량을 감지할 수 센서 입력 (8개)
    - 출력 : 바퀴를 돌리기 위하여 모터제어 출력
  - 시스템의 동작 정의
    - 8방향으로 센서를 달고 ADC 를 이용하여 각 센서를 컨버팅하여 가장 빛이 밝은 방향으로 이동한다.
    - 전방 센서가 가장 밝을 경우 직진한다.
    - 후방 센서가 가장 밝을 경우 후진한다.

광량 추적 로봇

- 개발 과정
  - 각 모듈별 분석
    - 광량을 감지할 수 있는 센서 입력부
      - 적외선 센서를 사용한다. (8방향이므로 8개)
      - 광량의 입력만을 받으므로 적외선 수광센서 사용
      - 광량을 감지하기 위해서는 센서의 데이터를 ADC 로 디지털로 변환해서 인식해야 한다.
      - 센서 인식을 위해 센서 동작 테스트가 필요하다.
      - ADC를 이용하여 센서 입력을 디지털로 변환하여 받아야 한다.
    - 연산 처리부
      - 선정된 MCU 로 처리한다.
      - MCU 에 관한 학습이 필요하다.
    - 모터 구동부
      - 모터는 제어하기 간편하고 많이 사용되는 스테핑 모터를 사용한다.
      - 스테핑 모터 구동을 위한 회로를 구성하고 테스트가 필요하다.

리모컨으로 제어하는 로봇

- 리모컨으로 제어하는 로봇
  - 적외선 리모컨의 입력을 받아 이동하는 로봇

- 개발 과정
  - 입력과 출력을 정의
    - 입력 : 리모컨 신호를 받을 수 있는 센서 입력
    - 출력 : 바퀴를 돌리기 위하여 모터제어 출력
  - 시스템의 동작 정의
    - 리모컨 버튼이 눌리면 리모컨에서 출력되는 신호를 분석하여 신호에 따라 이동을 한다.
  - 각 모듈별 분석
    - 리모컨 입력을 받는 있는 입력부
      - 리모컨에 사용하는 신호는 적외선 신호로 전용 리모컨 센서가 있다.
      - 리모컨 센서 인식을 위해 센서 동작 테스트와 학습이 필요하다.
    - 연산 처리부
    - 모터 구동부

마이크로 컨트롤러 학습

- MCU의 학습
  - “MCU 는 어떤 것이다” 라는 개념
  - MCU의 내부 구조
  - Address/Data Bus 개념
  - Memory 구조와 Access 방법
  - I/O 방식 (Memory Mapped I/O)
  - SFR (Special Function Register)
  - Interrupt 개념

- C 언어 학습
  - 마이크로 컨트롤러는 하드웨어만으로 아무런 동작도 할 수 없다.
  - 따라서 마이크로 컨트롤러에 프로그램을 해야 한다. C 언어는 대부분의 마이크로 컨트롤러 용 컴파일러가 있어 기본이 된다.

마이크로 컨트롤러 학습

- Target CPU 선정
  - H/W 적인 장비를 구비해야 하므로 처음에는 개발환경 구비가 용의한 CPU 를 선정하여 학습한다. (MCU, 컴파일러, ISP 등)

- 작품 제작
  - 마이크로 컨트롤러 학습을 위해서는 책을 보는 것 만으로는 부족하다.
  - 꼭 자신의 작품을 만들어야 한다.

- H/W vs S/W
  - H/W와 S/W 는 상호 보완적인 관계에 있다. 적정한 선에서 H/W와 S/W 의 역할을 분담하여 작품을 구성해야 한다. 경우에 따라서는 H/W 로 간단하게 해결되기도 하고 S/W 로 간단하게 해결 되기도 한다.
  - 실제 제품 개발에 있어서 H/W 작업의 비중은 약 30% 정도이고 S/W 의 비중이 약 70% 정도이다. S/W의 학습이 중요하다.