2019학년도 1학기

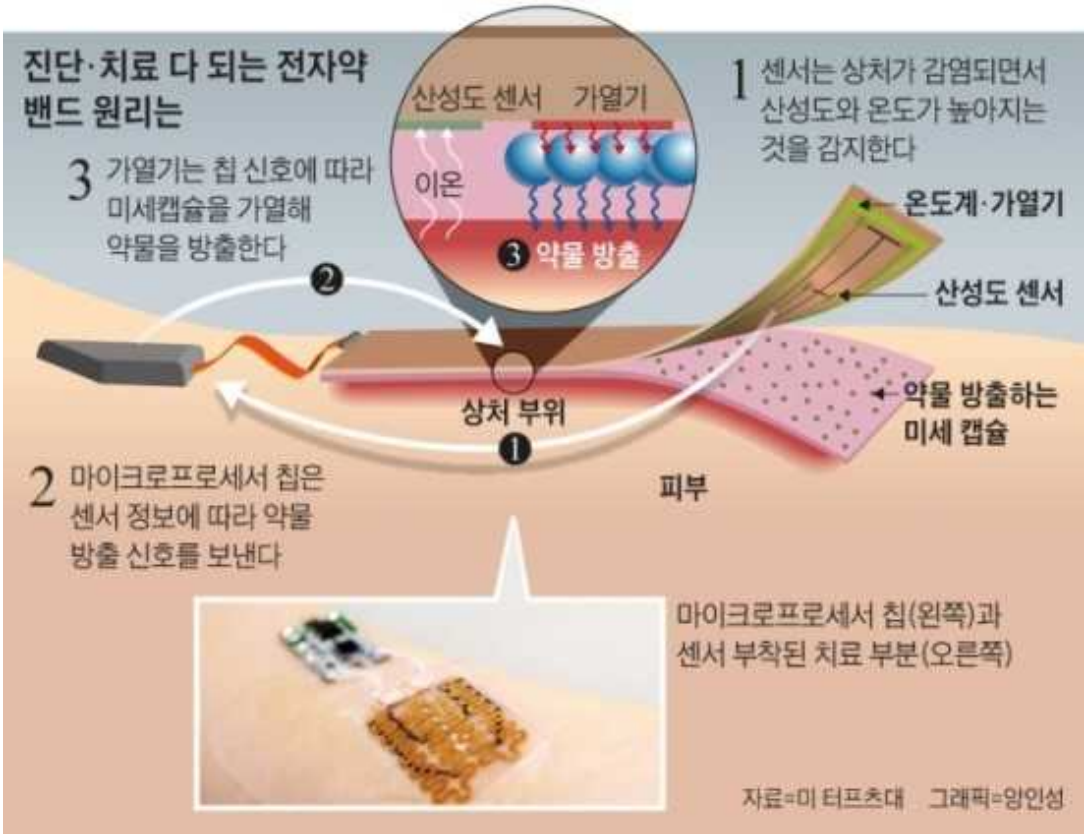
Why Microprocessor?

생각을 하며 스스로 문제를 해결해 나가는 능력을 키우는 수업

바람이 안불 때 바람개비를 돌리는 방법은 빠르게 앞으로 달려가는 것이다.

https://youtu.be/7kZP_MD39U4

■ Bio + Microprocessor 혁명



■ 4차 산업 혁명

■ VR(Virtual Reality)

자료출처: 앱스토리

■ AR(Augmented Reality)

■ AI(Artificial Intelligence)

■ Drone

■ Wearable Computer

■ Fin Tech

- Finance + Technology
- 삼성페이, 애플페이, 카카오 페이, 인터넷뱅크, 알리페이 등

■ IOT(Internet of Things) -> 1999년 Kevin Ashton

■ Why Microprocessor?

"사물인터넷(IoT)이 가져오는 중요한 기회를 잡을 수 있다"

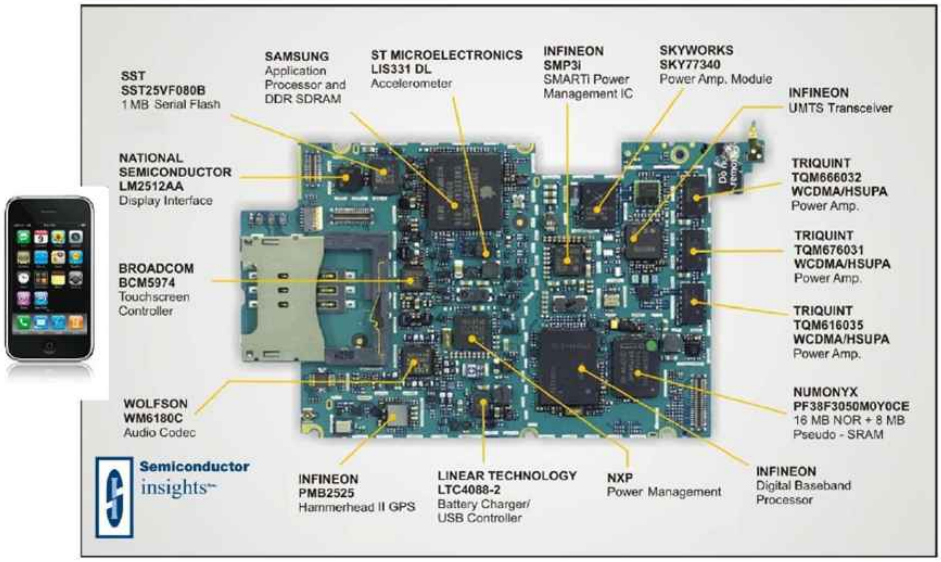
일본 소프트뱅크 손정의 사장은 영국 케임브리지에 본사를 둔 세계 2위 반도체 설계회사 ARM을 234억 파운드(약 35조 원)에 인수하는 배경을 이렇게 설명했다. 손정의 사장은 기술 업계의 패러다임 변화가 시작되는 시점에 항상 대형 투자를 해왔다.

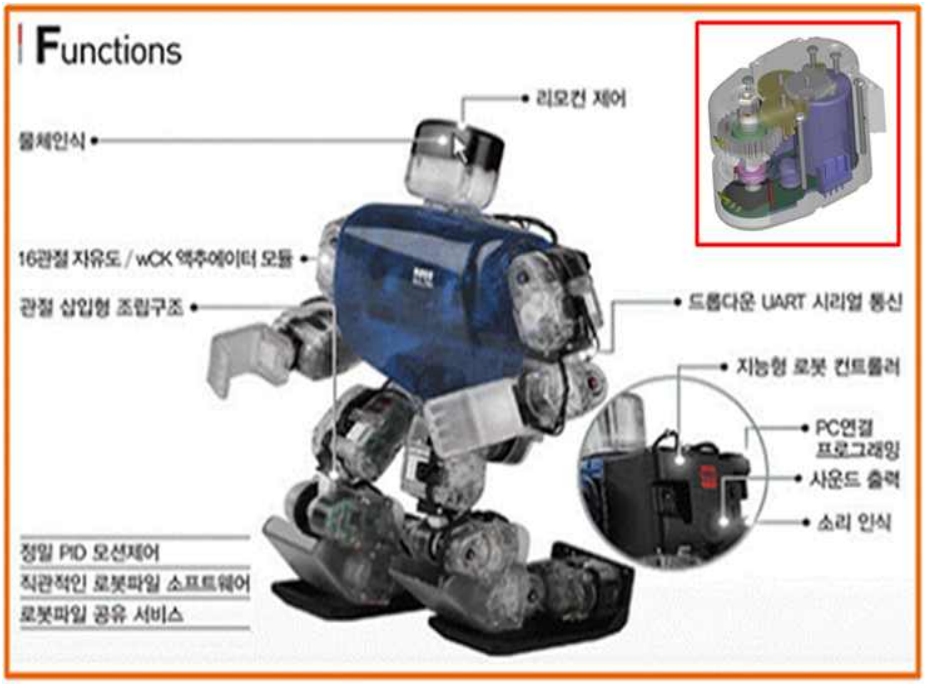
- 4차 산업 혁명의 핵심 기술
- VR, AR, AI, IOT, Connected Car, Big Data -> Arduino, AVR, ARM
- Microprocessor(Computer): Software + Hardware -> Firmware(OS, Integration) -> Application Software

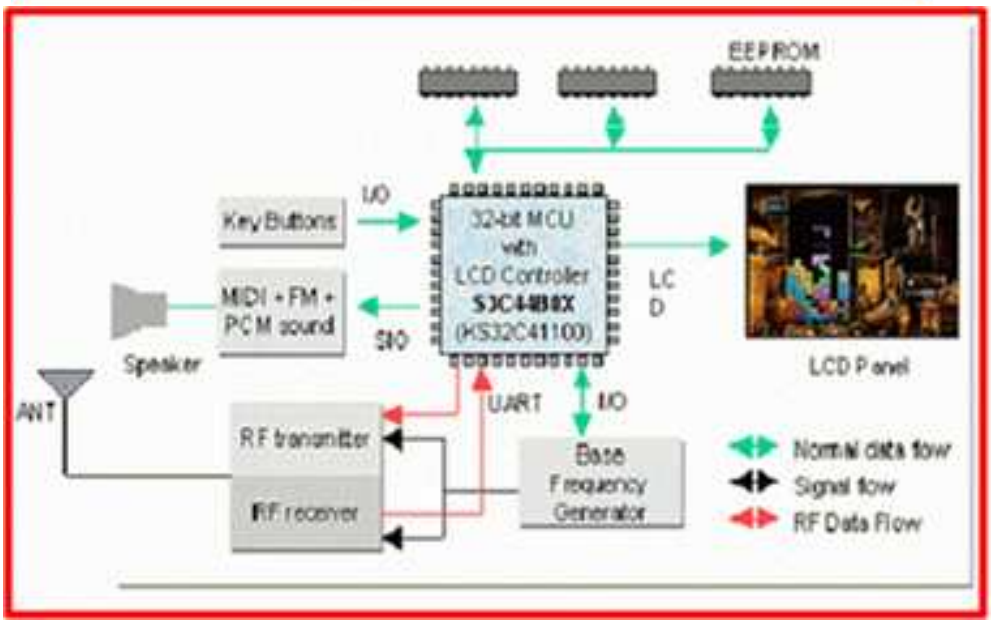
■ Connected Car(Smart Car)

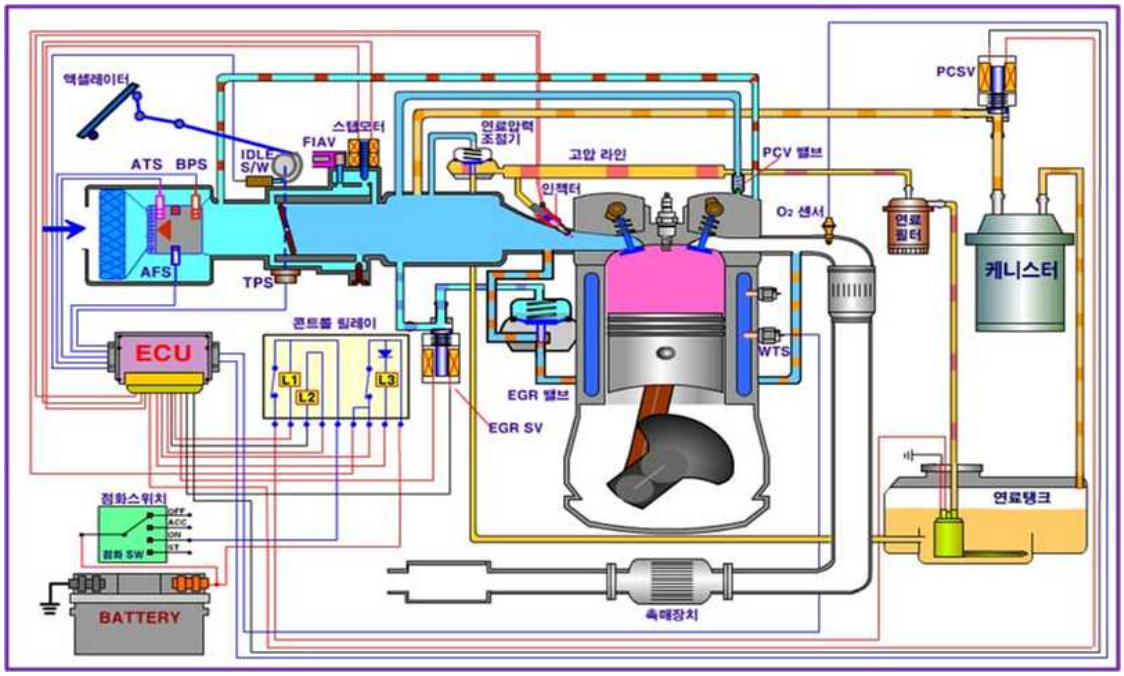
■ Microprocessor 사용예

■ Smart Device




Functions
리모컨 제어
물체인식
16관절 자유도 / wCK 액추에이터 모듈
관절 삽입형 조립구조
드롭다운 UART 시리얼 통신
지능형 로봇 컨트롤러
PC연결 프로그래밍
사운드 출력
소리 인식
정밀 PID 모션제어
직관적인 로봇파일 소프트웨어
로봇파일 공유 서비스

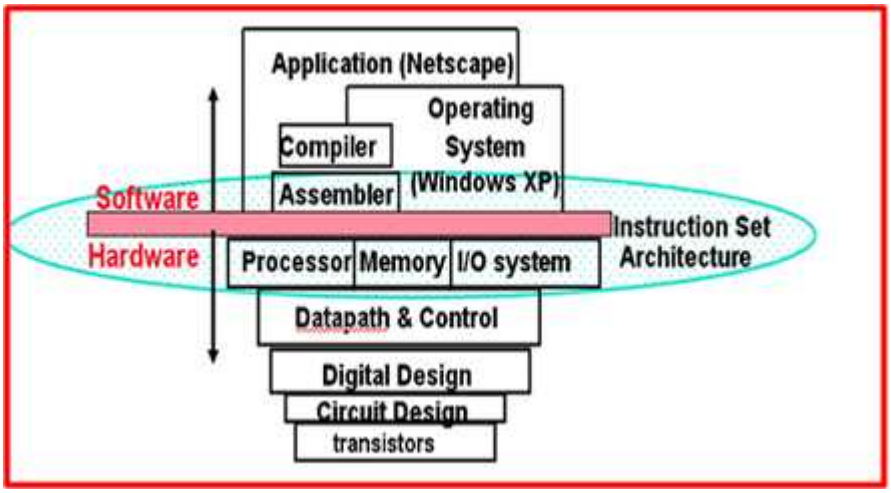

EEPROM
I/O
Key Buttons
32-bit MCU
with
LCD Controller
S3C44B0X
(KS32C41100)
LC
D
MIDI + FM +
PCM sound
SIO
Speaker
LCD Panel
ANT
UART
I/O
RF transmitter
RF receiver
Base
Frequency
Generator
Normal data flow
Signal flow
RF Data Flow


액셀레이터
ATS BPS
IDLE S/W
스텝모터
FIAV
연료압력 조절기
고압 라인
인젝터
PCSV
PCV 밸브
O2 센서
연료 필터
케니스터
AFS
TPS
콘트롤 릴레이
ECU
L1
L2
L3
WTS
EGR 밸브
EGR SV
점화스위치
점화 SW
BATTERY
연료탱크
촉매장치

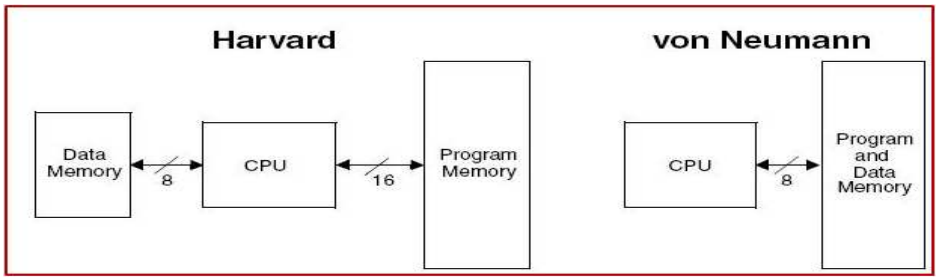
■ Why Arduino?

1. Microprocessor의 정의

■ CPU(Central Processing Unit) = MPU(Micro-Processor Unit)
  ☞ Software Program -> Compile(Assembling) -> Machine Language
  ☞ Register, ALU, BUS, Control Unit
  ☞ Software Program Base Operation
■ MCU(Microprocessor Control Unit: One Chip Micom) ->
  Arduino, ATmega128, MCS-51 family
  ☞ CPU Core + Memory + Timer + UART + A/D-D/A Converter 등
■ Embedded System: MCU + OS (Android, iOS, Window phone 7, Embedded Linux 등)
  ☞ Arduino, ARM Cortex Series
  ☞ Smart Phone등 Mobile 기기 (iPhone, Galaxy 등)
■ Hardware + Software + OS <-> Appilcation S/W



2. Harvard/Von Neumann Architecture



3. RISC와 CISC

■ Complex instruction set computer

☞ MCS-51 Family

■ Reduced instruction set computer

☞ Arduino, AVR, ARM

4. Microprocessor 발전과정

■ Intel: 80계열 -> x86 Series: CISC

1971년 4004,

1972년 8008(본격시대),

1973년 8080, 8085,

1976년 MCS-48, MCS-51(8051, 8031, 8751, 8951)

1978년 8086(16Bit시대개막), 80186, 80286,

1981년 80386(32Bit 시대), 80486, 80586(Pentium)

☞ IBM Computer CPU, 80486 펜티엄부터 RISC86

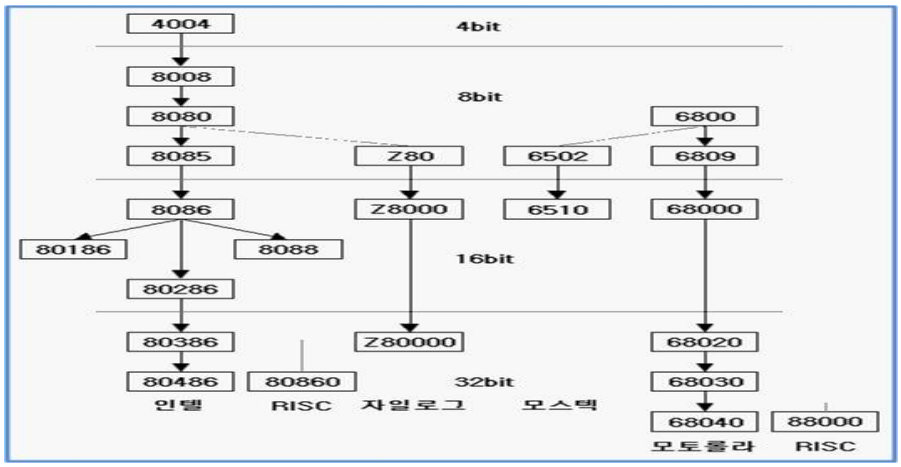
■ Motorola: 68계열

1974 MC6800(8), MC6802,

1978년 MC68000(16Bit 시대개막),

1981년 68020(32Bit 시대), 68030, 68040

☞ Apple-Macintosh Computer CPU



■ Arduino: 2005년 개발 됨

■ Arduino Uno, Mega, Due

■ AVR: RISC, ISP Technology 채택 -> ATtiny, ATmega Series

■ ARM(Advanced Risc Machine) Core -> Smartphone 시대

☞ 32-bit RISC 프로세서

☞ Smartphone, Tablet PC, Navigation, DMB 등

☞ Galaxy, iphone Series -> ARM Core

☞ Linux 기반 ARM based Samrtphone OS -> Android, ios, Window phon 7. Symbian, 바다, Window 8

■ Android OS

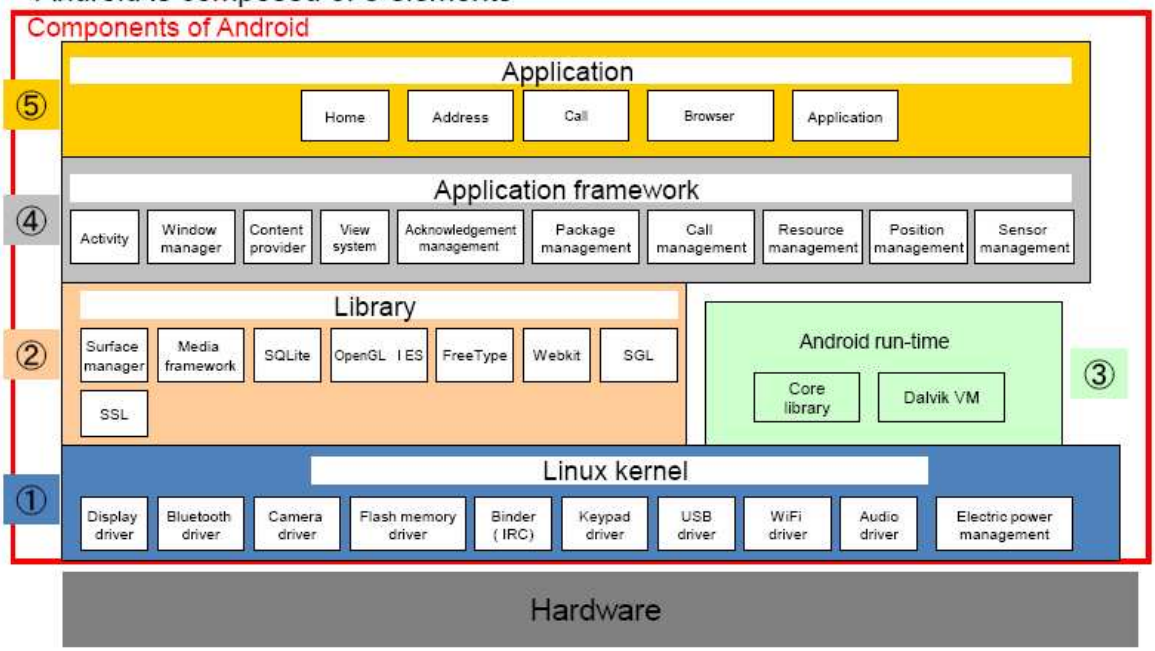
☞ OHA ; Open Handset Alliance가 2007년 11월에 공개

☞ Google 사가 인수하여 Google Android 만듬

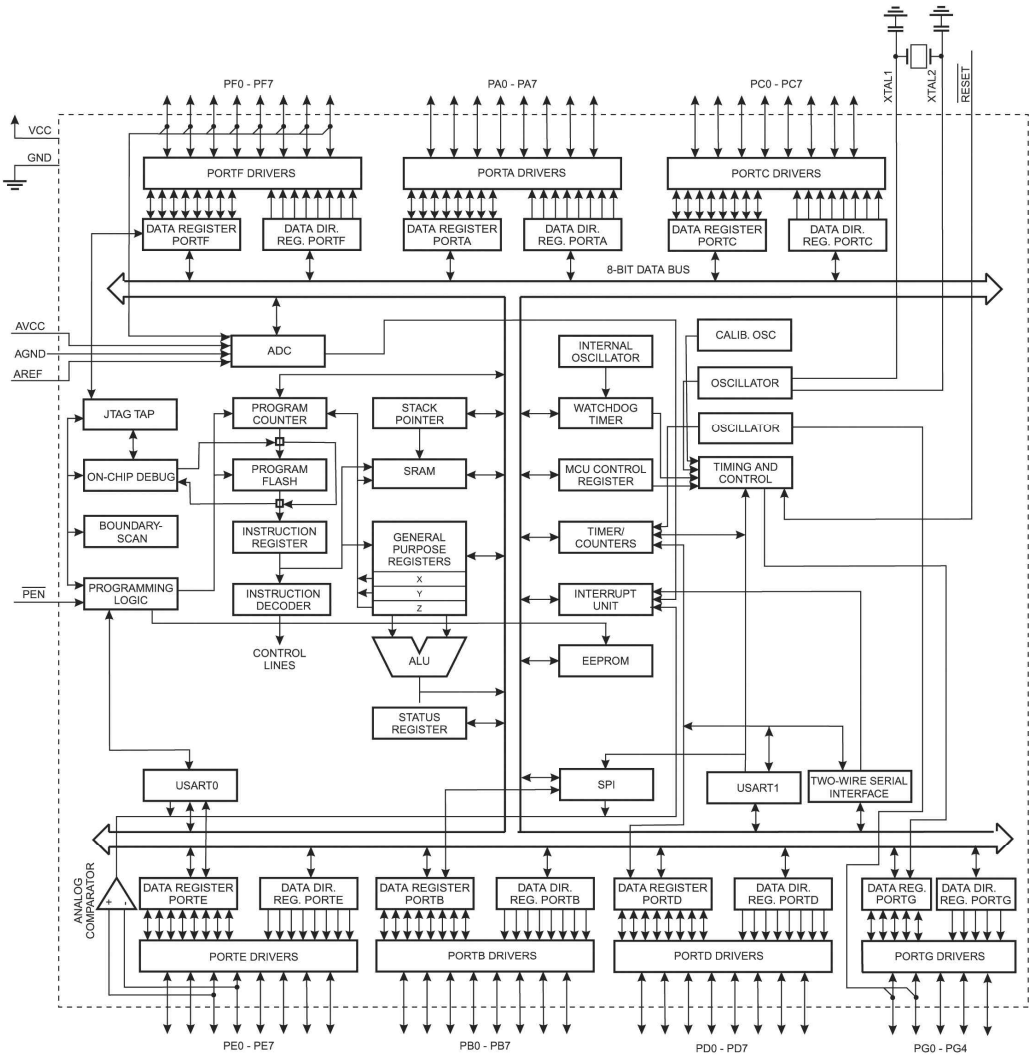
☞ Linux 2.6 커널을 기반

Android의 전체 구조

•Android is composed of 5 elements



5. MCU Internal Architecture 예



- Register: CPU내의 작은 메모리, 고속 Access Time
- ALU: Arithmetic Logic Unit
- Control Unit: Instruction Fetch, Analysis, Execution
- Memory
  - ☞ ROM, RAM, Flash Memory
- RAM(Random Access Memory)
  - ☞ Data Memory
  - ☞ DRAM(Dynamic RAM)
    - Refresh 회로
    - 전력 소모가 적고 기억용량이 매우 큼
  - ☞ SRAM(Static RAM)

- Battery Backup 시 계속 기억함
- 고속의 소용량 기억장치에 적합

■ ROM(Read Only Memory)
  ☞ Program Memory
  ☞ Only Read, Non-Volatility
  ☞ Program 입력시: ROM Writer 필요
■ MASK ROM : 대량주문용으로 생산자가 Only one 입력 가능
■ PROM(Programmable ROM) : ROM Writer로 한 번만 입력 가능
■ EPROM(Erasable PROM) : 여러 번 쓸 수가 있음
  ☞ ROM Writer: Program 입력
  ☞ Eraser: 자외선으로 Program Erase
  ☞ 27C256(32K: 256K Bit / 8 Bit = 32K Byte)
  ☞ 27C010(1M Bit), 27C040(4M Bit)
■ EEPROM(Electrically EPROM) : 자외선 대신 전기 신호로 지움
  ☞ RAM과 같이 수시 입력이 가능
  ☞ 단점: 입력시간이 오래걸리고, 횟수 한정(10만 회)
  ☞ 28C16(16K Bit=2K Byte), 28C64
■ Flash Memory:
  ☞ EEPROM과 같이 자유롭게 수시 입력 가능
  ☞ 가장 많이 쓰이는 Memory
  ☞ NAND(Data Memory, 삼성), NOR(Code Memory, 인텔)
  ☞ Mobile Phone, Digital Camera, MP3, PMP, Navigation 등
  ☞ 29C010(128K Byte), 29C040

7. Microprocessor Language

■ High Level Language -> C, C++, Visual C, C#, JAVA 등
■ Low Level Language -> Assembly
■ High Level Language -> Assembly Language -> Machine Language
■ Assembly Language: Mnemonic Code + Pseudo instruction

■ Machine Language: CPU는 2진수만을 처리할 수 있음
  ☞ 2진수로 구성된 언어

☞ MOV A, #7F

0111 0100 0111 1111

☞ bit, nibble, byte, word

☞ MSB(Most Significant Bit), LSB

☞ Hexadecimal:

* 2진수의 단점 보완, PGM에서는 주로 사용
* 2진수 -> 16진수:1248 Code, 4비트씩 나눔

1001 1101 -> 9D

* 16진수 -> 2진수

5C -> 0101 1100

☞ PGM상 숫자표시: 11B, 11b, 11H, 11h, 11D, 11d

■ Assembly Language

☞ 2진수, 16진수의 단점 보완

☞ 기호, 문자, 16진수로 알기 쉽게 구성됨

☞ CPU 제조사에 따라 다름

☞ Program Size 작음 - Memory Save, Cost Down 등

☞ Mnemonic Code + Pseudo Instruction

(ORG, End, Comment 등)

☞ MOV A, B - Mnemonic Code

0111 0100 0111 1111 - 2진수

74 7F - 16 진수

☞ MOV A, B

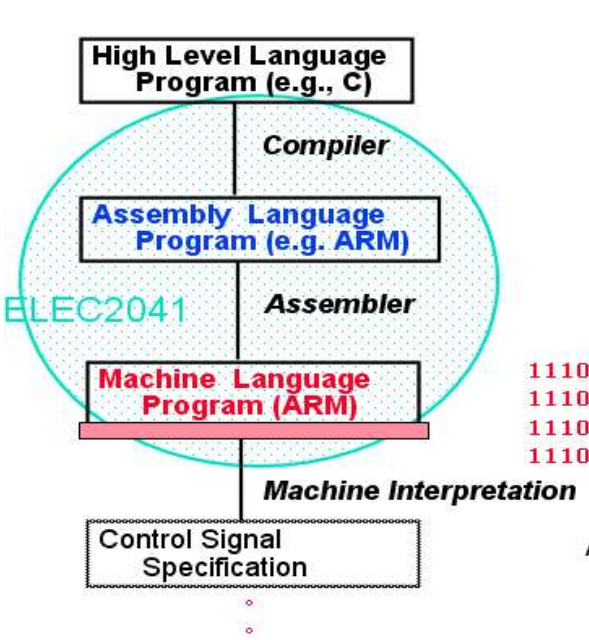
(OPeration Code+ 1st Operand + 2nd Operand)

■ Compiler: High Level Language -> Machine Language

■ Assembler: Assembly Language-> Machine Language

■ Source program(*.C) -> assembly program -> Object program -> Execution program -> ROM

■ Compiling/Assembling Process

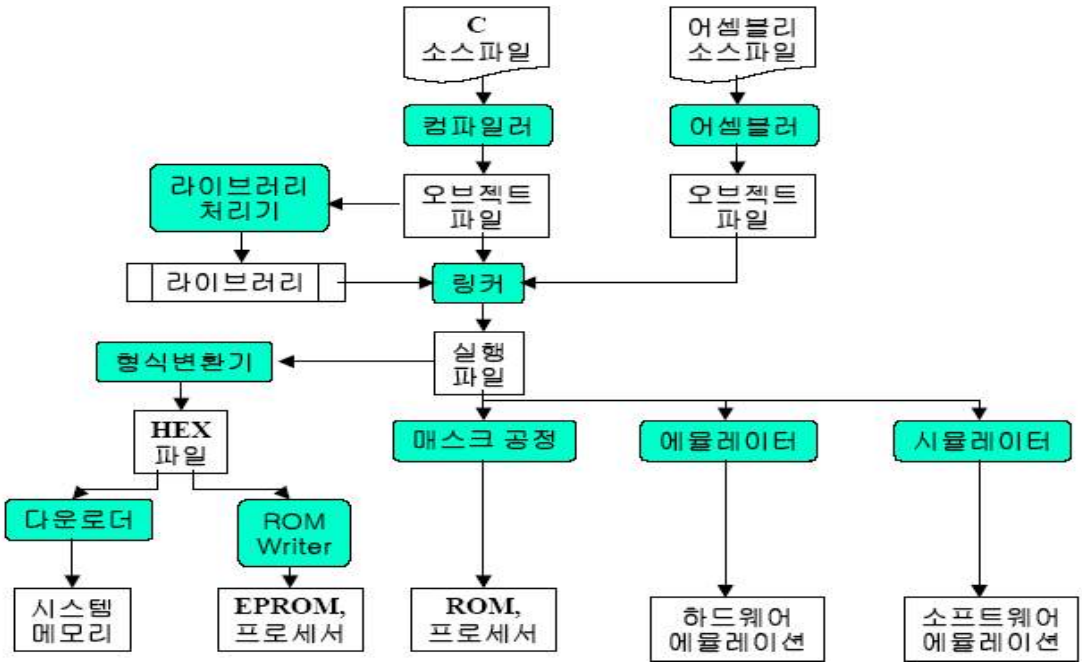



```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
ldr  r0 , [r2, #0]
ldr  r1 , [r2, #4]
str  r1 , [r2, #0]
str  r0 , [r2, #4]
```

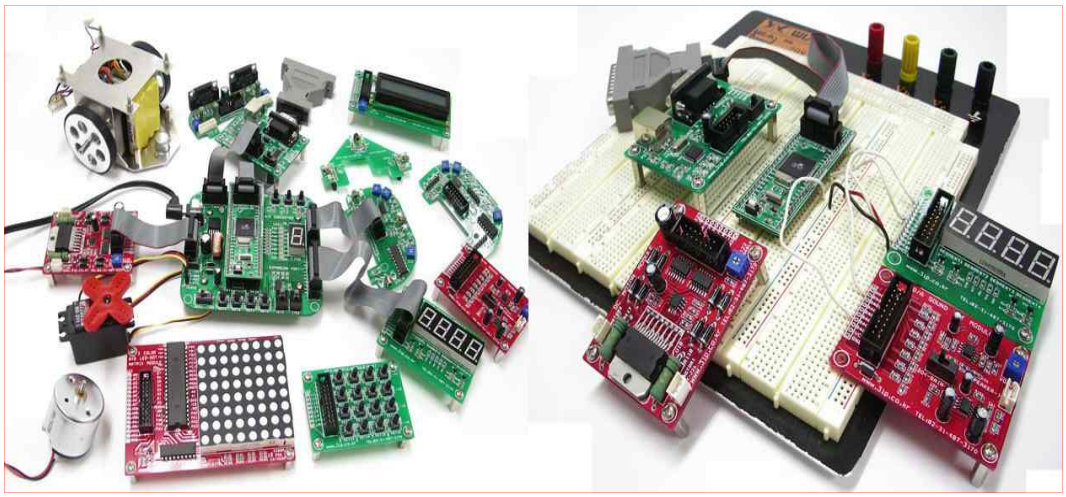
```
1110 0101 1001 0010 0000 0000 0000 0000
1110 0101 1001 0010 0000 0000 0000 0100
1110 0101 1000 0010 0001 0000 0000 0000
1110 0101 1000 0010 0001 0000 0000 0100
```

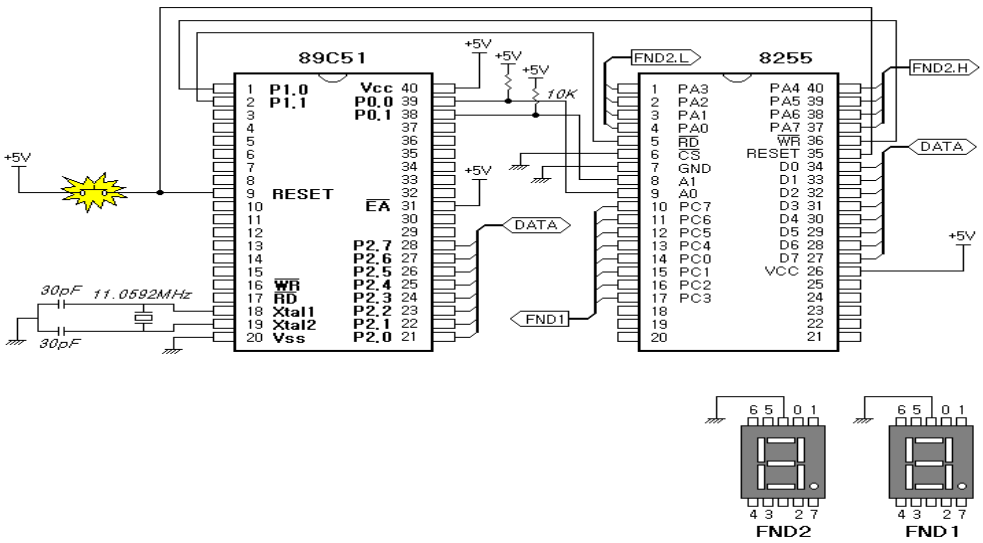
ALUOP[0:3] <= InstReg[9:11] & MASK

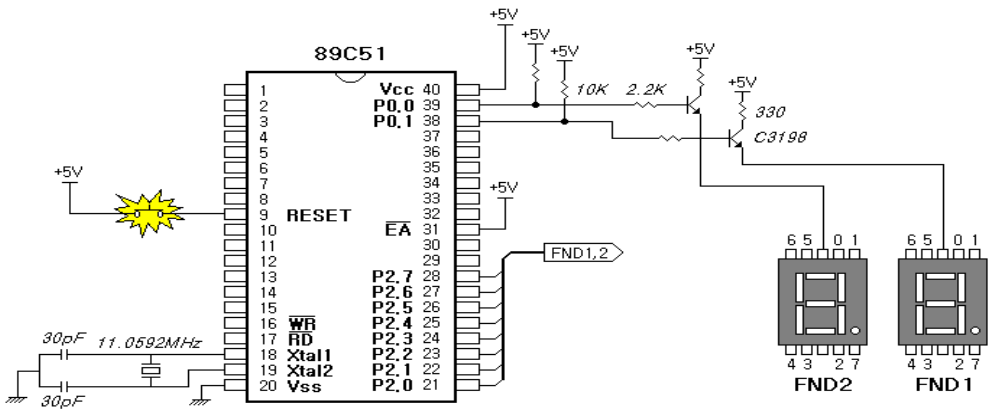


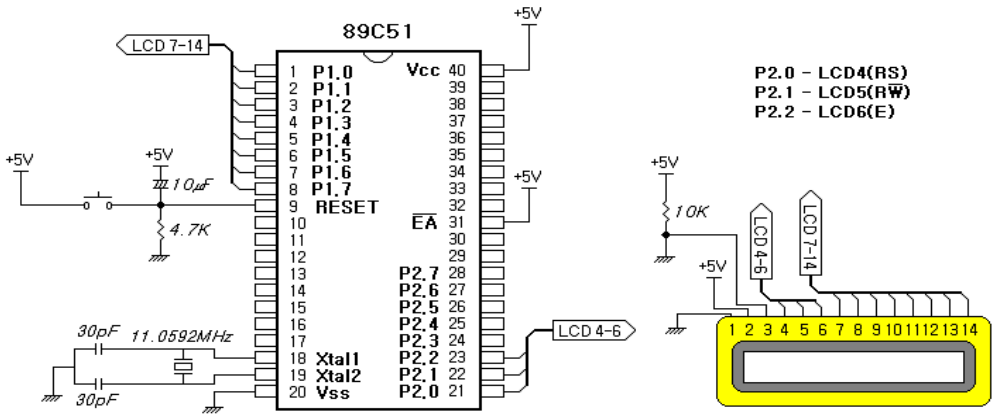
8. Microprocessor Applications

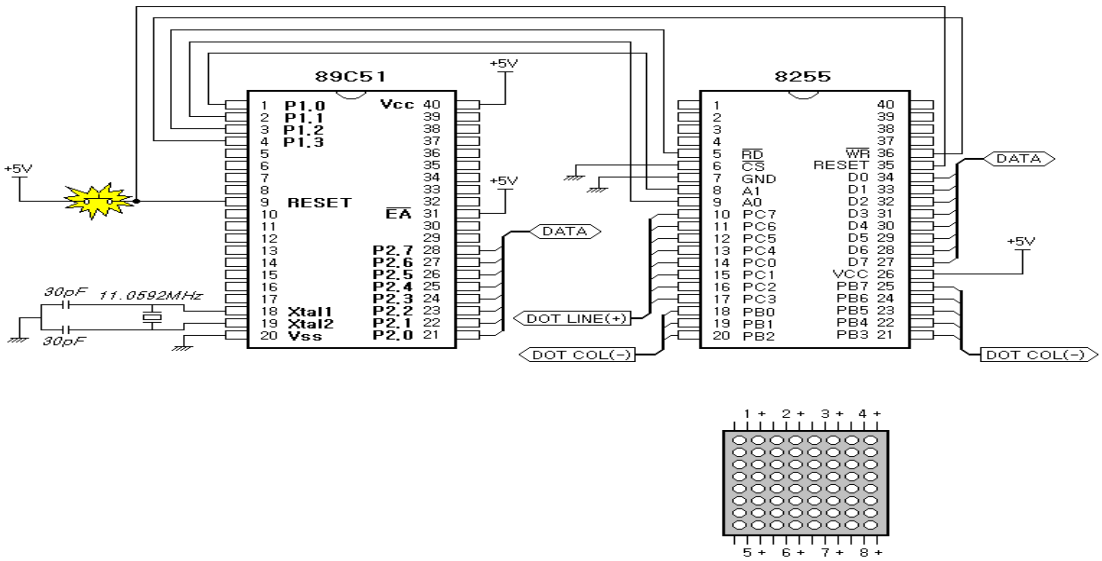
9. Microprocessor의 간단한 Application Circuit

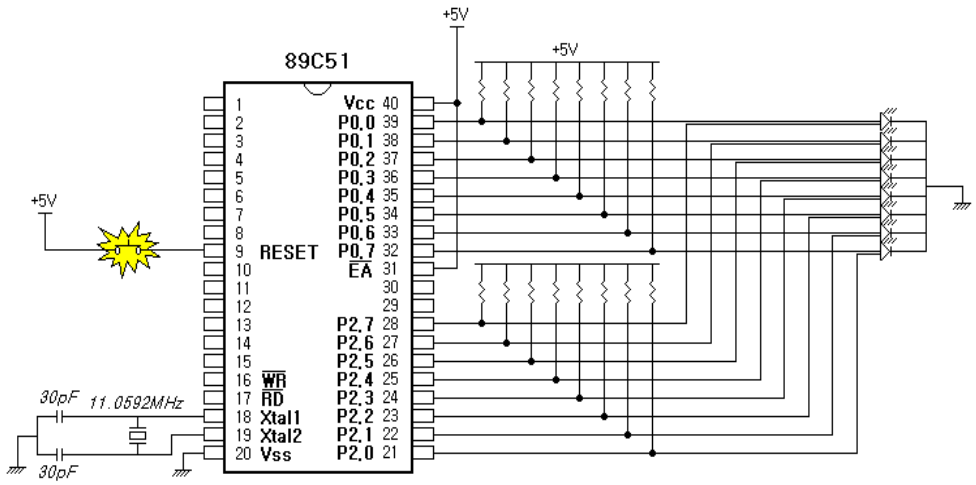


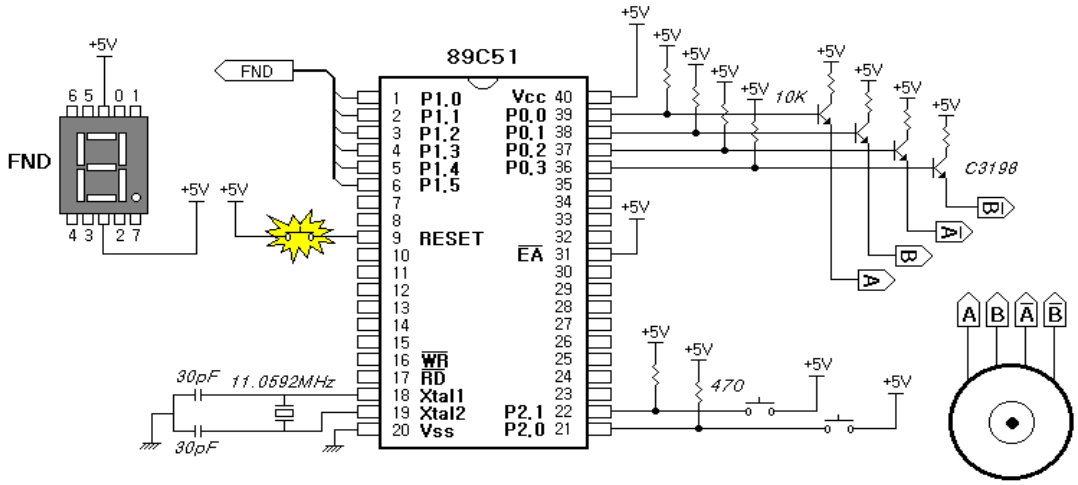





89C51
LCD 7-14
+5V
10μF
4.7K
30pF
11.0592MHz
30pF
LCD 4-6
P2.0 - LCD4(RS)
P2.1 - LCD5(R/W)
P2.2 - LCD6(E)
10K
LCD 4-6
LCD 7-14
1 2 3 4 5 6 7 8 9 10 11 12 13 14


89C51
8255
+5V
RESET
DATA
DOT LINE(+)
DOT COL(-)
30pF
11.0592MHz
30pF
1 + 2 + 3 + 4 +
5 + 6 + 7 + 8 +


89C51
+5V
RESET
30pF
11.0592MHz
30pF



Thinking & Idea !!!