



”

# 임베디드 시스템

**-MCU(Micro Controller Unit)**

**발표시작 >**



# 목차

”

**01** 임베디드  
시스템이란?

**02** MCU

**03** 임베디드 개발  
환경 세팅

**04** 실제 프로그래밍



”

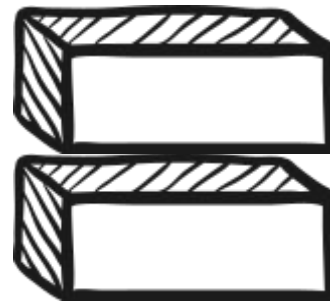
임베디드  
시스템이란?

01

## ● ● ○ 임베디드 시스템의 사전적 정의

“

# Embedded



# System

”

### 사전적 정의

특정한 제품이나 솔루션에서 주어진 작업을 수행하기 위해 추가로 탑재되는 시스템.

기계나 기타 제어가 필요한 시스템에 대해, 제어를 위한 특정 기능을 수행하는 시스템

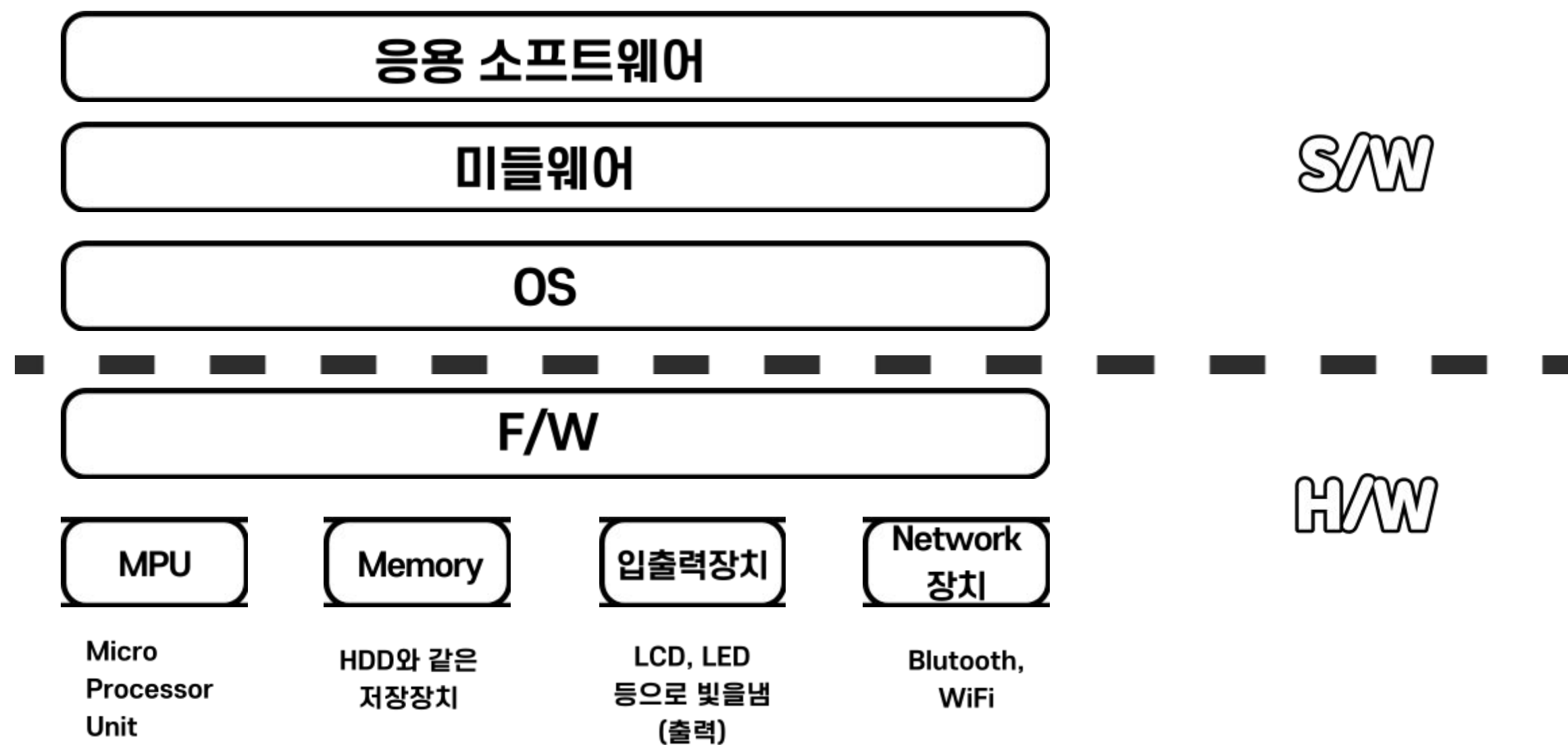
### 내가 생각하는 정의

마이크로프로세서를 내장하여 이용하는 범용이 아닌 전자기기

내부에 마이크로프로세서를 가지고 있는 점에서는 일반 범용 PC와 동일하지만, 정해진 용도로만 사용된다는 점이 다르다.

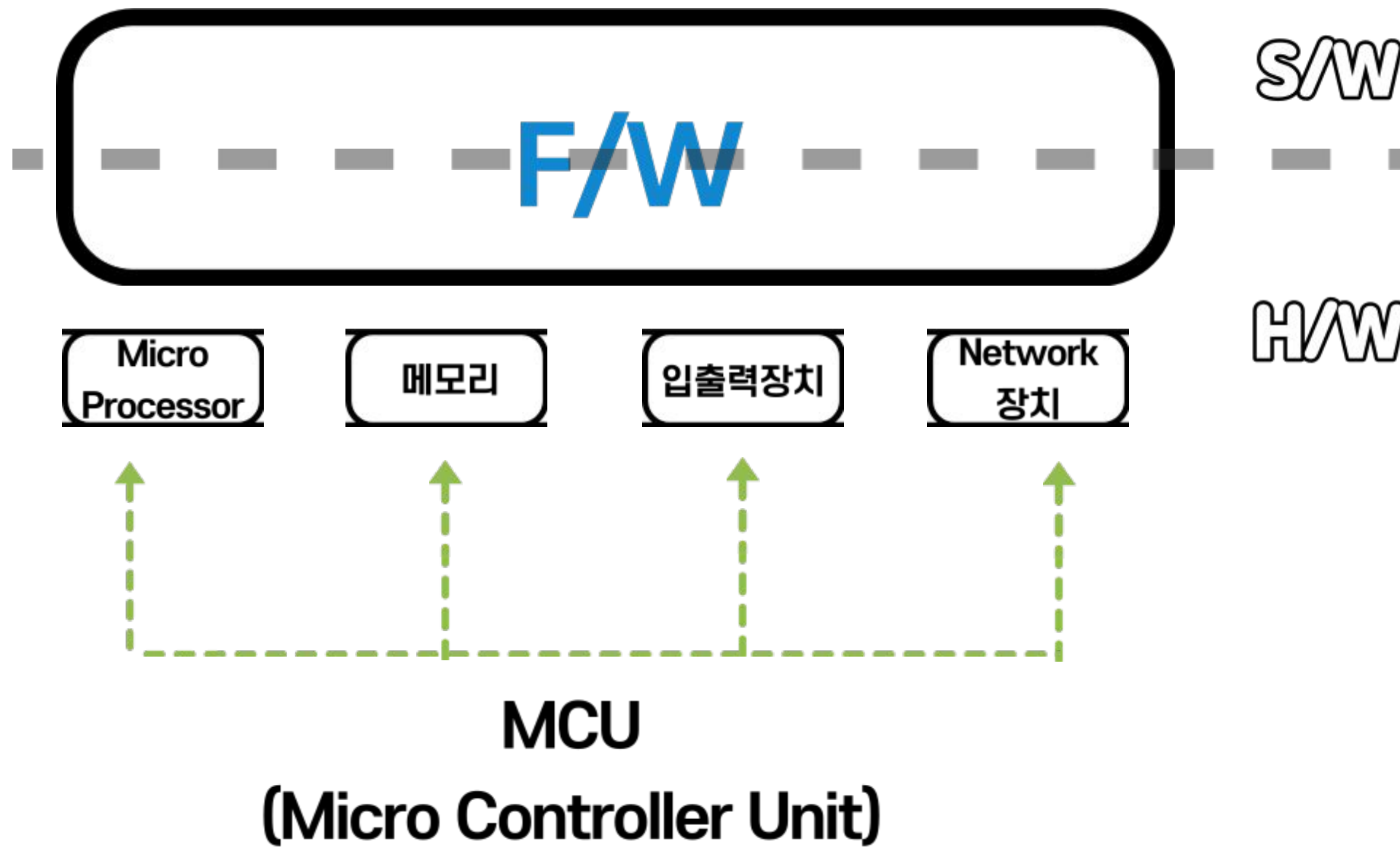
## ● ● ○ 임베디드 시스템의 구성

### Case1: OS가 있는 경우



## ● ● ○ 임베디드 시스템의 구성

### Case2: OS가 없는 경우    대부분의 임베디드 시스템에서 사용



#### F/W(펌웨어)란

임베디드 시스템에서 구현되는 하드웨어와 소프트웨어 중간에 있는 소프트웨어.

임베디드 시스템은 하드웨어로만 구현할 수 있지만, 하드웨어와 소프트웨어를 병행해 구현할 수 있다. 하드웨어로만 구현한다면, 속도는 빠르지만 기능을 수정하거나 확장하기 어렵기 때문에, **펌웨어를 통해 소프트웨어를 병행한다면 기능의 수정이나 확장이 쉬워진다.**

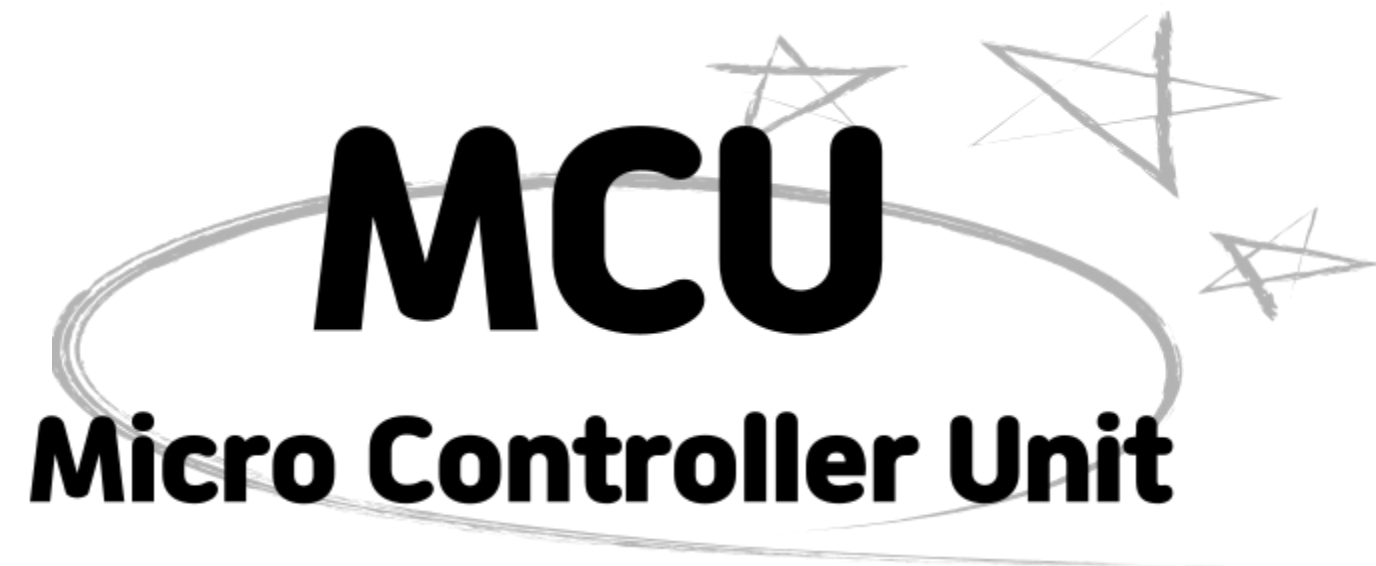
#### MCU란

마이크로컨트롤러(MCU)는 MPU, 메모리, 입출력장치, Network장치 등이 함께 집적된 장치.

MCU사용을 위해서는 펌웨어 프로그래밍을 하여야함

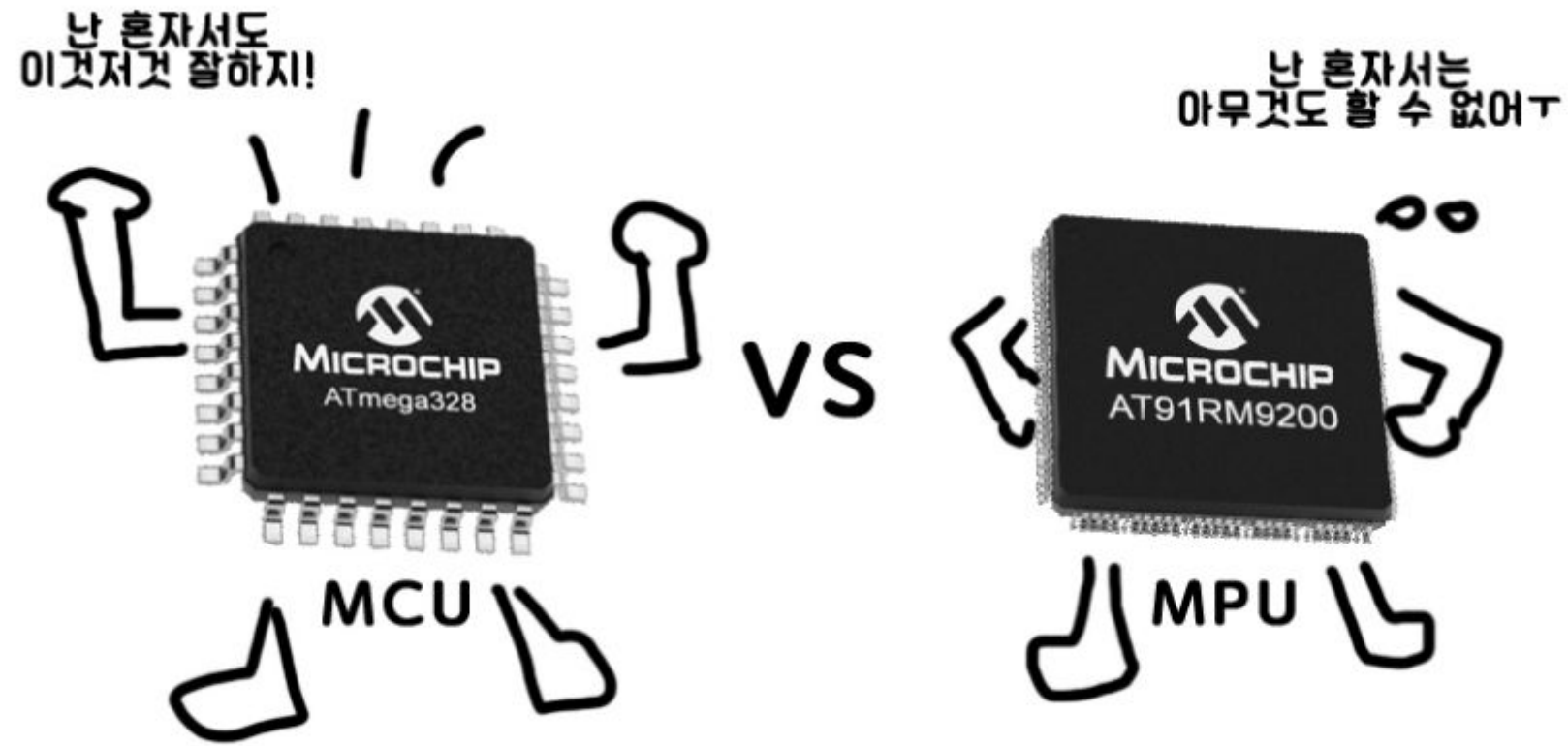


”



02

## ● ● ○ MPU와 MCU의 차이점



### MPU(Micro Processor Unit)

MPU는 컴퓨터의 핵심 기능인 주어진 기계어를 해석하고, 연산을 수행하는 기능에 중점을 두고 있다.  
그렇기 때문에 MPU는 주변에 RAM, ROM, I/O 등의 장치를 추가해주지 않으면 작동을 하는 것이 불가능함

### MCU(Micro Controller Unit)

CPU의 기능을 하는 핵심 장치와 그 주변 장치들을 포함하고 있는 통합형 칩셋.  
그렇기 때문에 MCU하나만으로 LED나 모터와 같은 다른 부품들을 제어 할 수 있다.



## ● ● ○ MCU가 활용되는 곳



- ▲밥솥이나 TV 녹화 장치의 타이머 기능
- ▲휴대폰의 배터리 전압 측정 및 잔여 배터리 용량 표시
- ▲시계의 시간 표시
- ▲버튼을 누르면 기능 수행

전자제품에서 우리가 익숙하게 사용하는 대부분의 기능이 MCU로 구현된다.

일례로 예약 기능을 갖춘 밥솥을 한시간 안에 조리가 시작되도록 설정하려면 밥솥의 MCU가 이런 일을 수행하도록 프로그램을 짜 넣어야 한다.

MCU가 특정한 기능을 구현하려면 프로그래밍 과정을 거쳐야 한다. 따라서 **MCU의 활용 분야는 프로그래밍을 어떻게 하느냐에 따라 달라진다.**



# MCU의 종류

AVR, 8051, PIC, ARM

여러 반도체 회사들은 다양한 종류의 MCU를 생산, 판매하고 있다. 같은 계열의 제품이라도 적용 시 요구되는 사양과 구현하는 기능도 다양하기 때문에 같은 제품군 안에서 다양한 모델이 출시된다.

## AVR

1996년 아트멜사에서 개발되었다. 8비트 MCU인 만큼 단순 제어 분야에서 활용이 높다. 오랫동안 사용돼 오면서 안정성이 검증됐고, 32비트급 MCU에 비해 단가가 매우 싸기 때문에 여전히 널리 사용되고 있다.

## 8051

인텔(Intel)에서 만든 8051은 앞서 설명한 1975년에 개발된 아주 초창기의 MCU다. 범용적인 기능을 가진 CPU만을 만들어오던 인텔이 보다 사용하기 쉽고 주변장치 제어용으로 적합한 제품을 구상하다 만들어 낸 제품이다.

## PIC

미국의 마이크로칩(MicroChip Technology)에서 만든 PIC는 AVR이나 8051와 함께 널리 사용되는 MCU 중 하나이고, 특히 산업용으로 많이 사용되고 있어 주변의 가전제품 속에서 쉽게 볼 수가 있다. PIC는 8비트 MCU부터 32비트 MCU까지 다양하게 존재한다.

## ARM

ARM은 MCU 제품 그 자체가 아닌, MCU의 코어(Core)를 구성하는 회로를 의미한다. ARM에서는 오직 ARM 코어의 회로를 설계하고 연구할 뿐이며, 직접 생산하지 않는다. 즉, 실제 ARM 기반의 MCU 생산은 다양한 업체에서 수행하고, 회로도 사용에 대한 라이선스 비용을 지불하는 방식으로 운영된다.








”

# 임베디드 개발 환경 세팅 with nRF52832

MCU칩을 직접 조작하기 위해서 펌웨어를 프로그래밍하는 것

# 03

	
 <b>Step 1</b>	펌웨어 프로그래밍을 하기 위해 크로스 컴파일 환경을 제공하는 툴을 선택한다. Segger Embedded Studio, Visual Studio Code등을 사용할 수 있다.
 <b>Step 2</b>	개발 및 디버깅을 위해서 컴퓨터와 MCU를 연결한다. 두 개가 연결되면 다양한 입출력 동작도 가능해진다. J-Link 등을 사용한다.
 <b>Step 3</b>	Step 1에서 선택한 툴에 사용하는 MPU의 헤더파일 등을 삽입하여 개발을 준비한다.
 <b>Step 4</b>	실제로 테스트, 개발, 디버깅 등을 할 수 있다.

# • • ◦ nRF52832 기준의 개발 환경 세팅

## Step 1

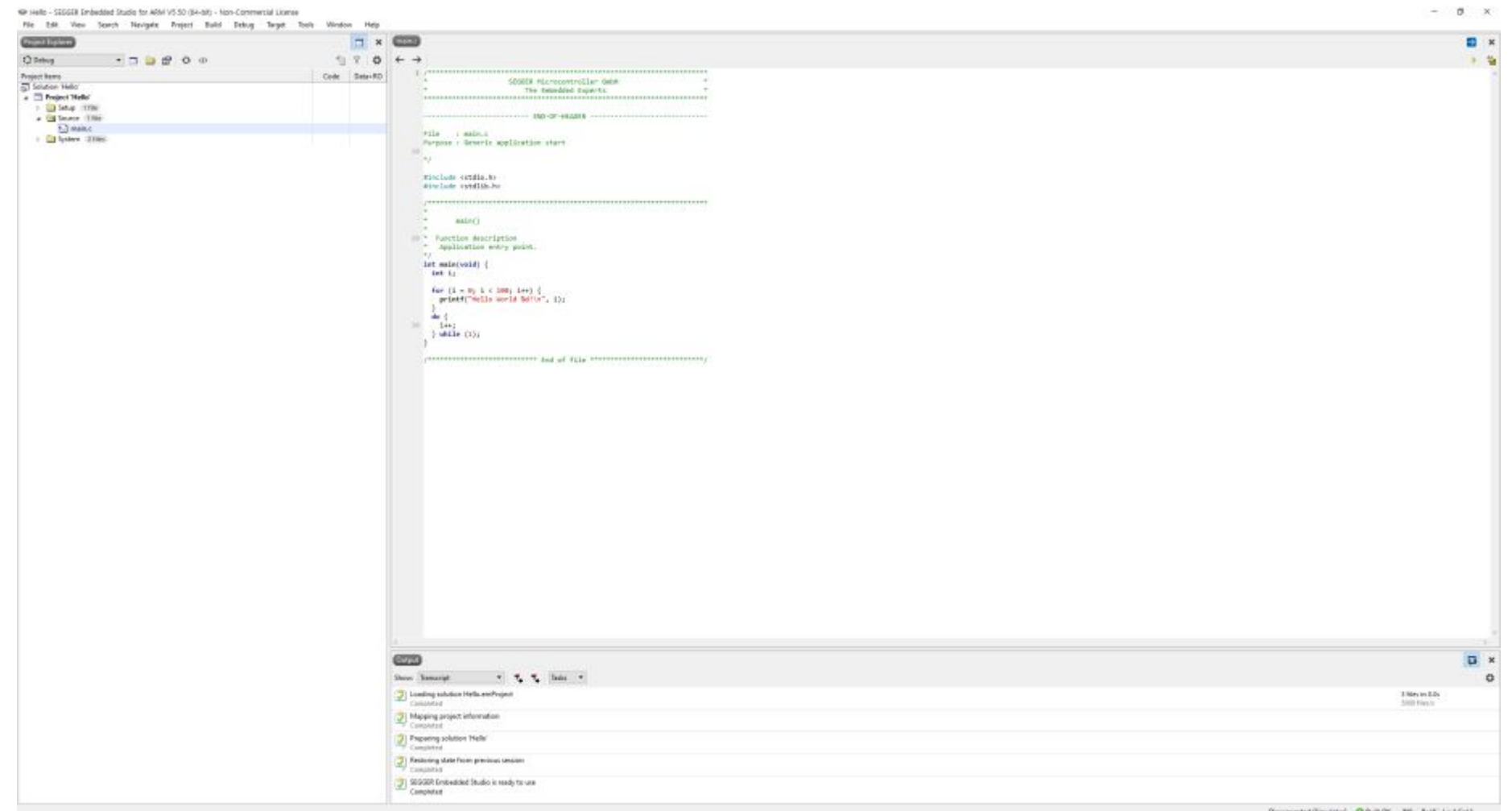
펌웨어 프로그래밍을 하기 위해 크로스 컴파일 환경을 제공하는 툴 선택

Segger Embedded Studio

Download: <https://www.segger.com/downloads/embedded-studio>

Embedded Studio for ARM				
	Version	Date	File size	
Embedded Studio for ARM, Windows, 64-bit Simply download and run the installer.	V4.30c	[2019-12-03]	284,093 KB	DOWNLOAD
Embedded Studio for ARM, Windows, 32-bit Simply download and run the installer.	V4.30c	[2019-12-03]	264,660 KB	DOWNLOAD
Embedded Studio for ARM, macOS Download and mount the image, then run the installer.	V4.30c	[2019-12-03]	379,565 KB	DOWNLOAD
Embedded Studio for ARM, Linux, 64-bit Download and extract the archive, then run the installer.	V4.30c	[2019-12-03]	314,983 KB	DOWNLOAD
Embedded Studio for ARM, Linux, 32-bit Download and extract the archive, then run the installer.	V4.30c	[2019-12-03]	330,704 KB	DOWNLOAD

Nordic Semiconductor은 IDE를 제공하지 않아서 다른 회사의 MCU들보다는 사용이 불편하다. 그렇기 때문에 J-Link를 만든 SEGGER라는 회사와 제휴를 맺어서 무료로 Segger Embedded Studio라는 툴을 제공한다. 물론 통합 툴이기때문에 다른 회사의 MCU개발도 가능하다.



## ● ● ○ nRF52832 기준의 개발 환경 세팅

### Step 2

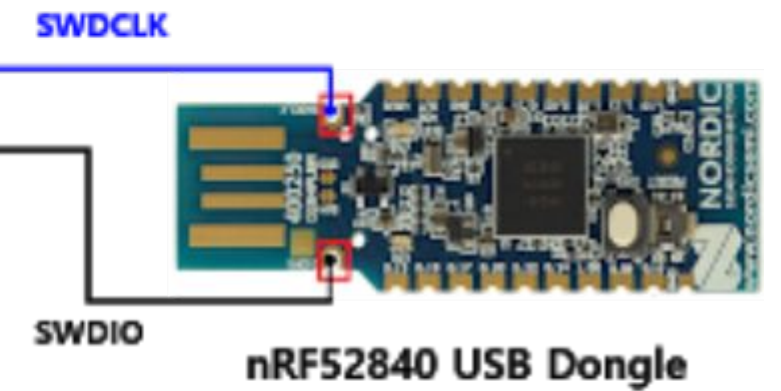
개발 및 디버깅을 위해서 컴퓨터와 MCU를 연결한다.



J-Link

\* J-Link 핀맵

2	1
4	3
6	5
8	7
10	9
12	11
14	13
16	15
18	17
20	19



MCU를 연결하는 곳과 USB포트를 J-Link에 연결시킨다. MCU는 여러가지 장치들이 한 번에 연결된 컨트롤러이기 때문에, USB Dongle이 존재하는것을 알 수 있다.



## ● ● ○ nRF52832 기준의 개발 환경 세팅

### Step 3

Step 1에서 선택한 툴에 사용하는 MPU의 헤더파일 등을 삽입하여 개발을 준비한다.

```
bool MPU6050_ReadGyro(int16_t *pGYRO_X , int16_t *pGYRO_Y , int16_t *pGYRO_Z )
{
    uint8_t buf[6];

    bool ret = false;

    if(mpu6050_register_read(MPU6050_GYRO_OUT, buf, 6) == true)
    {
```

```
bool MPU6050_ReadAcc( int16_t *pACC_X , int16_t *pACC_Y , int16_t *pACC_Z )
{
    uint8_t buf[6];
    bool ret = false;

    if(mpu6050_register_read(MPU6050_ACC_OUT, buf, 6) == true)
    {
        mpu6050_register_read(MPU6050_ACC_OUT, buf, 6);
```

해당 코드는 nRF52832내부에 사용되는 mpu6050.c의 소스 파일이다. 해당 소스를 사용하면 mcu 내부의 가속도 센서 및 자이로센서의 값을 불러올 수 있다.



”

**실제 프로그래밍**

**with nRF52832**

**04**



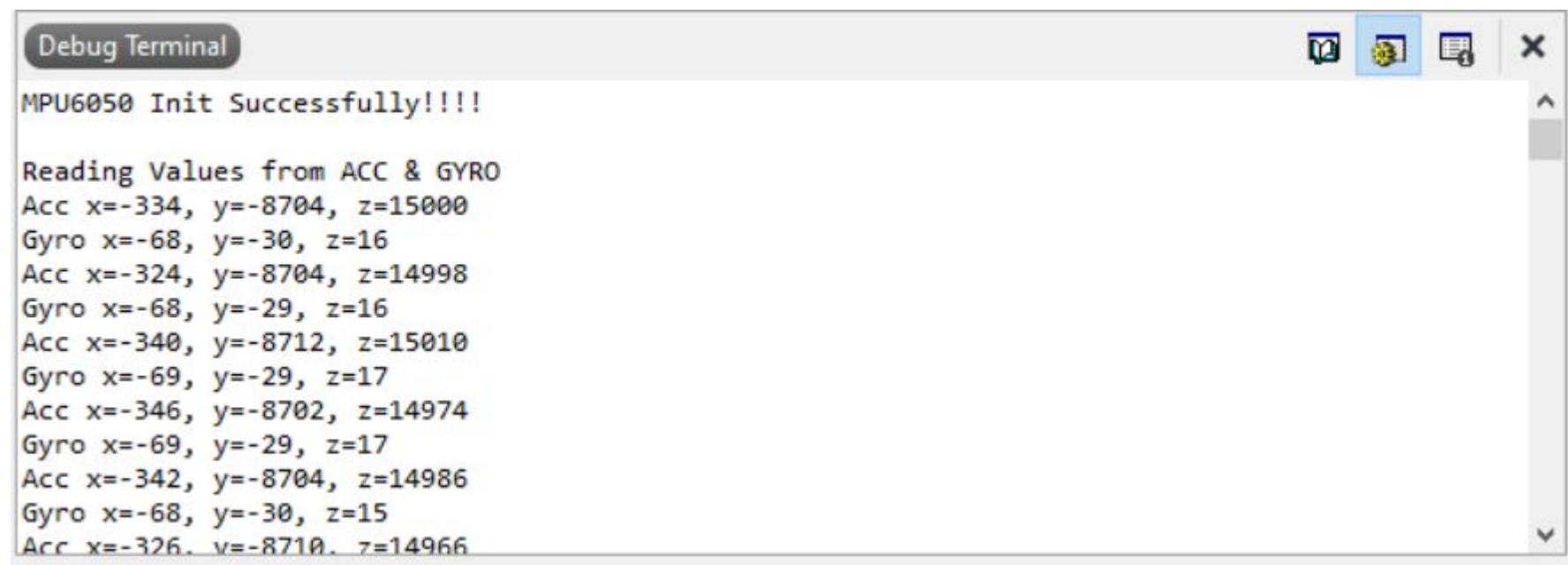
# 자이로 센서, 가속도 센서 값 불러오기

```
while(mpu6050_init() ==false)
{
    printf("MPU_6050 initialization failed!!\n");
}

printf("MPU_6050 initialization Success!!\n");

if(MPU6050_ReadAcc(&acc[0], &acc[1], &acc[2]) ==true)
{
    printf("Acc x=%d, y=%d, z=%d\n", acc[0], acc[1], acc[2]);
}
else{
    printf("Reading ACC values Failed!!!");
}

if(MPU6050_ReadGyro(&gyro[0], &gyro[1], &gyro[2]) ==true)
{
    printf("Gyro x=%d, y=%d, z=%d\n", gyro[0], gyro[1], gyro[2]);
}
else
{
    printf("Reading Gyro values Failed!!!");
}
```



```
Debug Terminal
MPU6050 Init Successfully!!!!

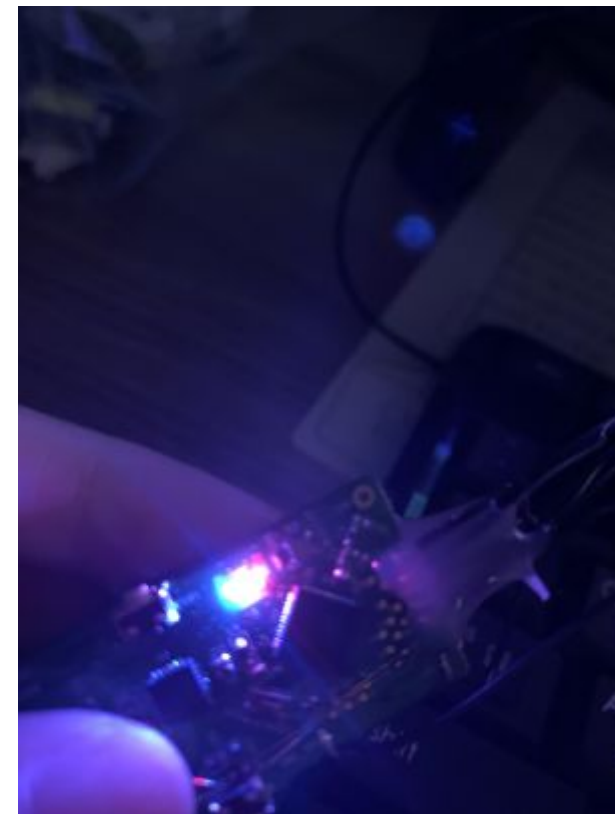
Reading Values from ACC & GYRO
Acc x=-334, y=-8704, z=15000
Gyro x=-68, y=-30, z=16
Acc x=-324, y=-8704, z=14998
Gyro x=-68, y=-29, z=16
Acc x=-340, y=-8712, z=15010
Gyro x=-69, y=-29, z=17
Acc x=-346, y=-8702, z=14974
Gyro x=-69, y=-29, z=17
Acc x=-342, y=-8704, z=14986
Gyro x=-68, y=-30, z=15
Acc x=-326, y=-8710, z=14966
```



# LED 켜기

```
nrf_gpio_range_cfg_output(30,31);

while(1){
  for (int i = 0; i < 2; i++)
  {
    nrf_gpio_pin_toggle(1<<(i+30));
    nrf_delay_ms(100);
  }
}
```





”

# 감사합니다!

발표에 대한 질의응답은 언제나 환영입니다 :)

**발표 끝!**