

*ARM Mbed OS*

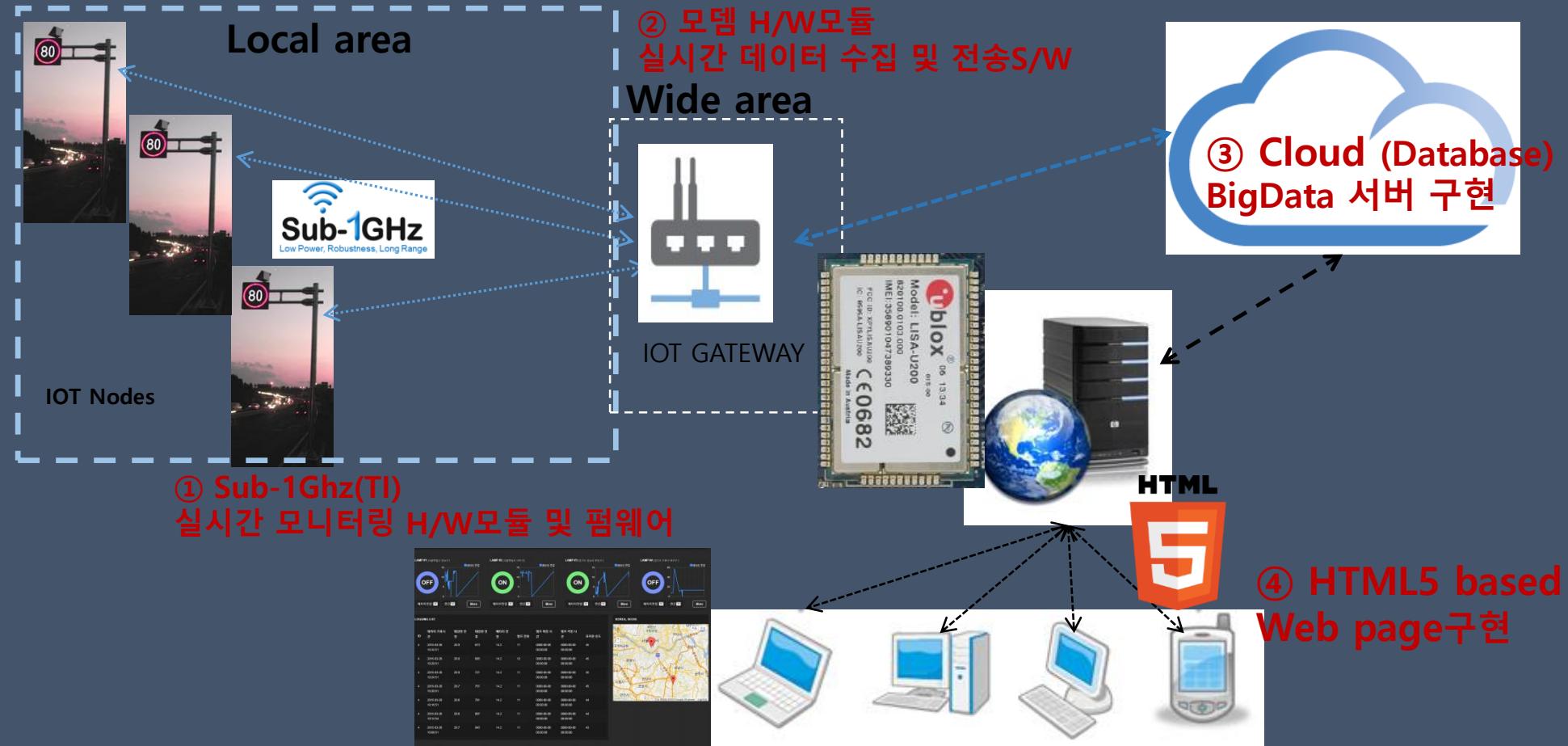
*Day 1*

# 회사소개

CodeZoo

# 회사소개

## 도로표지판 실시간 상태 측정을 위한 통신모듈 및 관제시스템

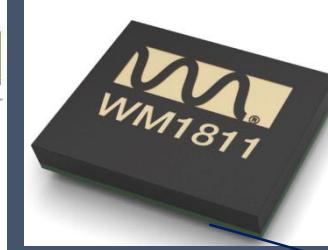
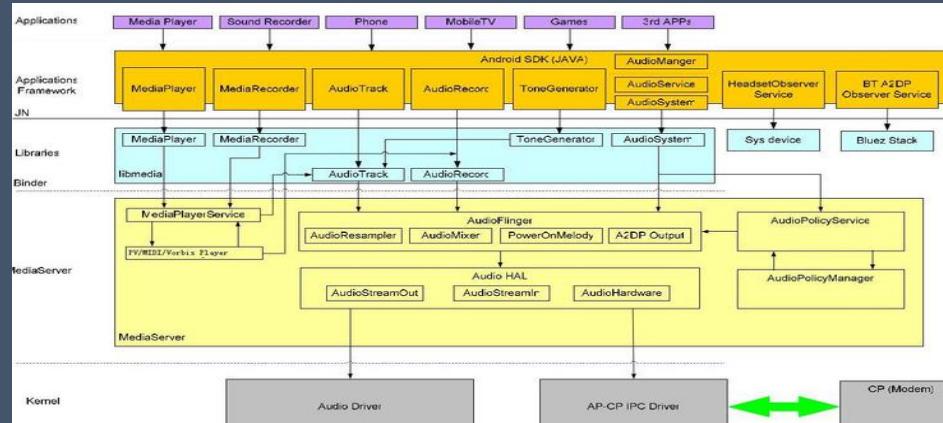


# 회사소개

## 안드로이드 오디오 및 모뎀 (Android RIL) 시스템 개발

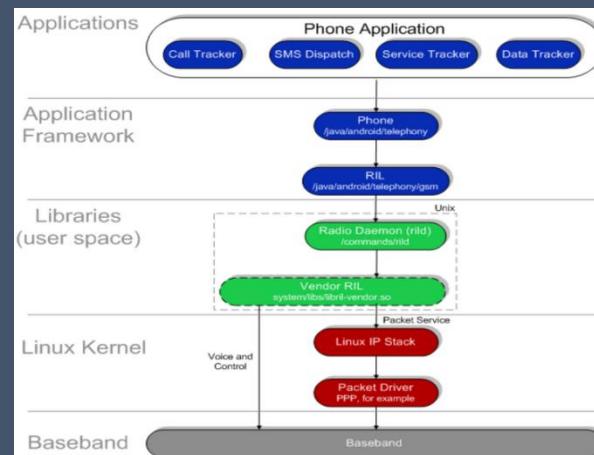


- 하드웨어 분석/검토
- PowerManagement 구현(kernel & Android)



wolfsont WM1811A

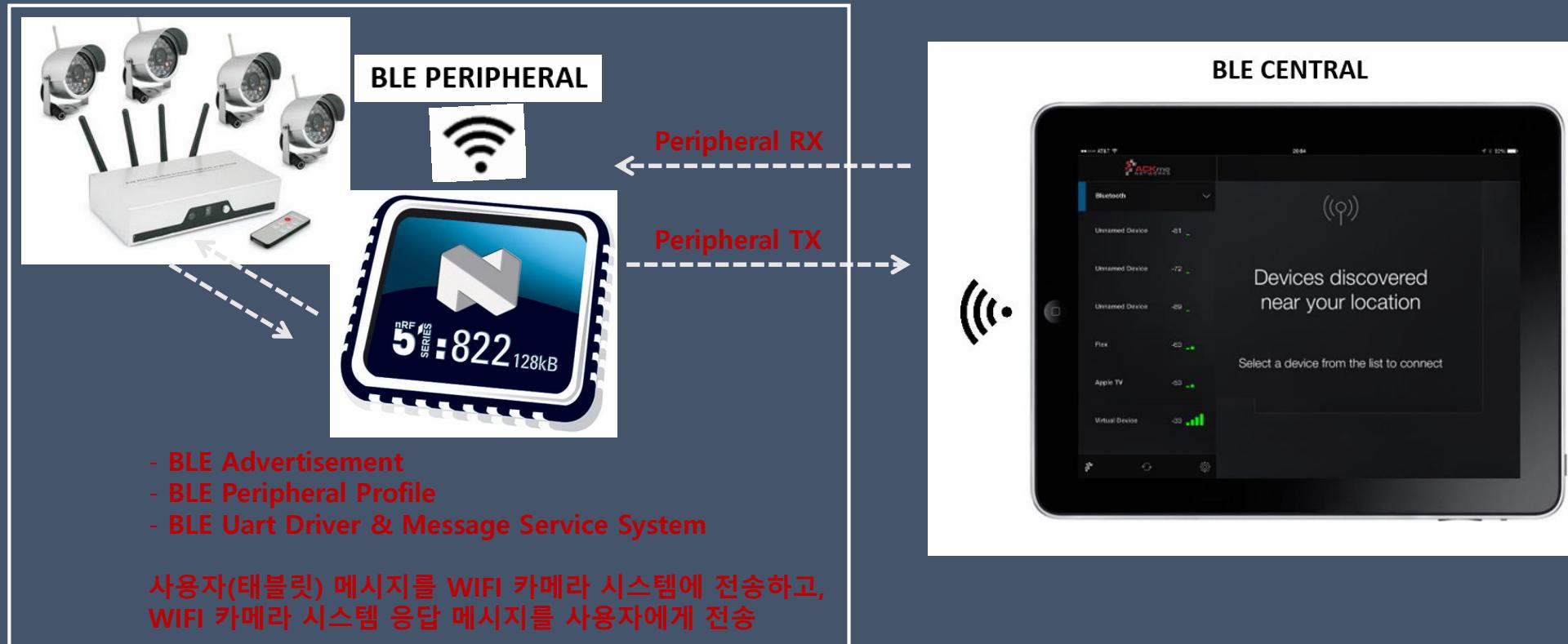
- linux device driver
- android audio HAL porting
- Call audio interface



- Android RIL (ublox SARA-U270) HAL porting
- KT OEM(Cashbee) Payment 구현

# 회사소개

## 트레일러 무선카메라 제어를 위한 BLE(nrf51822) 펌웨어 개발



# 회사소개

## Home Security System, 3G Modem S/W 개발



한화테크원 Home Security System에서 전송되는 영상 및 Security information을 모뎀을 통해 실시간 송수신 및 업데이트

- MMS 전송
- 서버 업로드, 다운로드
- 모뎀 SPI& USART AT command 병렬제어
- CA(Certification Authority)저장 및 삭제- TLS v1 Cipher Suite
- HTTPS Server communication

# 회사소개

## 전기자동차 테스트 솔루션 개발, GSM Modem S/W 개발



북경현대자동차 전기자동차 테스트에 사용되는 전장모듈용 테스트 솔루션 개발. RTOS환경에서 GSM모뎀을 통한 차량 실시간 데이터 송수신 및 모듈 펌웨어 업그레이드용 부트로더, 드라이버, 서비스 구현

- STM32F105 MCU, Quectel MC60 GSM 모뎀
- TCP Socket 사용을 위한 API 구현
- 서버 업로드 및 다운로드 기능
- 모뎀 USART AT command 제어를 위한 Parser 구현
- Serial Flash Driver 및 전장모듈 업데이트를 위한 부트로더 구현

# 회사소개

## CodeZoo Connectivity Module



실시간 위치추적을 위해 코드주 개발한 Connectivity module

- MCU : STM32F401  
(ARM Cortex-M4, 84Mhz Clock speed, 64KB RAM, 256KB Flash)
- GPS : Ublox max 7Q GPS/Glonass
- Modem : Ublox SARA-U270  
(HSPA/GSM module for Europe/Asia HSPA B1, B8, GSM B3, B8)
- compatible Arduino shield
- compatible TI boosterPack
- Supported S/W : STM32 Hal based S/W, ARM Mbed based S/W

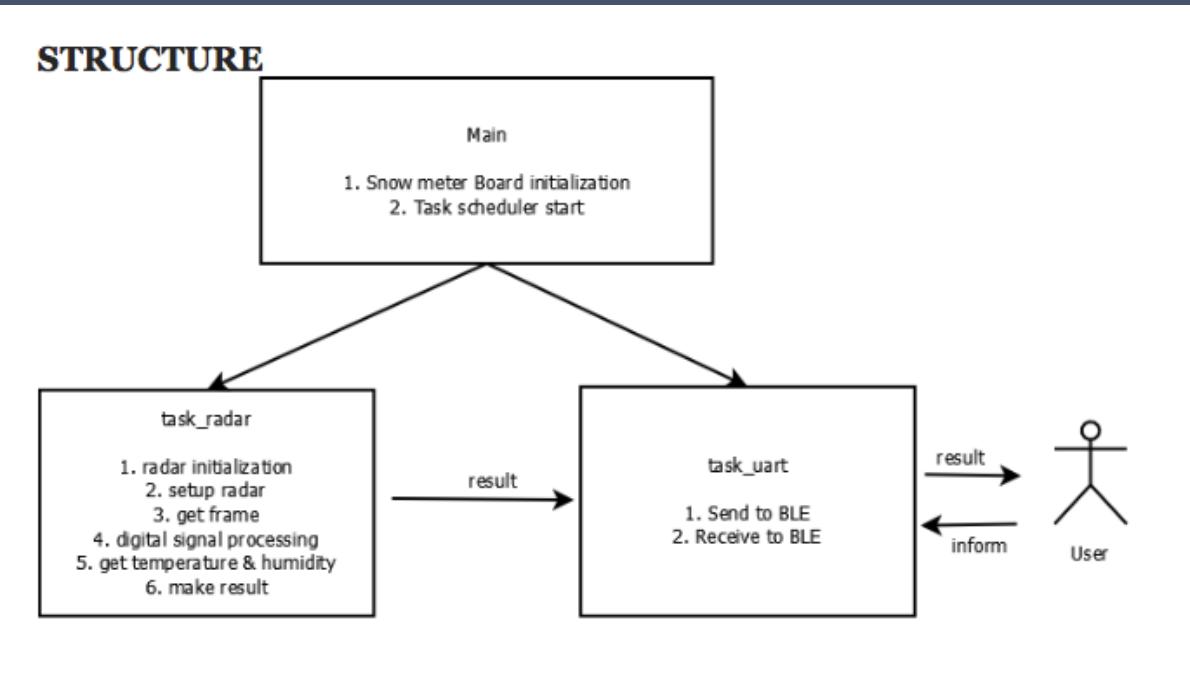
Open Source official

<https://github.com/codezoo-ltd>

www.CodeZoo.co.kr

# 회사소개

## UWB Radar Solution



1. Novelda X2로 ATMELO ATSAM4SD32B MCU RTOS 펌웨어 개발
  - FreeRTOS Porting- Novelda X2 SDK Porting
  - Matlab 으로 제작된 적설계(snow meter) 알고리즘을 C++로 코드 작성
  - BLE Driver
  - 온습도 Driver
  - BLE를 이용하여 스마트폰에 적설량 데이터 전송
2. Novelda X2에 STM32F769 MCU RTOS 펌웨어 개발
  - FreeRTOS Porting- Novelda X2 SDK Porting
  - SPI Flash Driver
  - ESP32 WIFI Driver
  - SDRAM library

# 회사소개

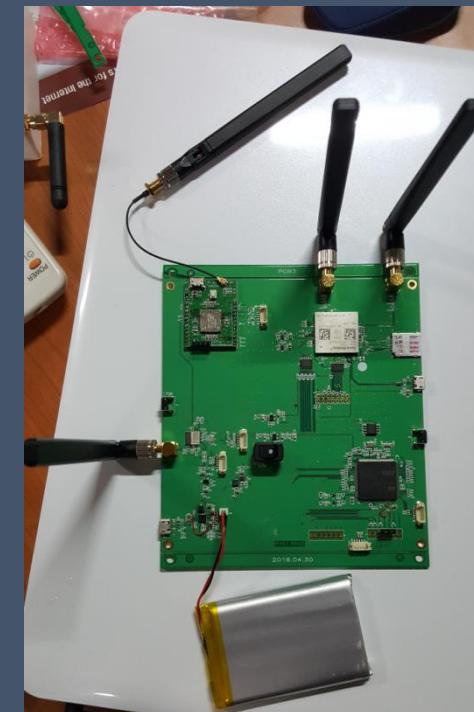
## 플랜트 안전관리 시스템



작업자 움직임 트래킹  
(BLE 5.0 Beacon)

작업공간 산소농도 측정  
(LoRa Private)

- ✓ 제작 : CodeZoo (디바이스 3종, 80세트)
- ✓ 고객 : Hanwha Total
- ✓ BLE : 작업자 사고여부 측정
- ✓ LoRa : 작업공간 산소농도 실시간 측정
- ✓ LTE : LGU+ LTE망 이용 관제시스템 데이터 송수신
- ✓ Sensor, Solution : CodeZoo



센서 데이터 수집  
LTE 게이트웨이  
(BLE, LoRa Private)

# 회사소개

## 코드주 스마트 에어 모니터



스마트폰에 USB로 직접 연결하여 초미세먼지, TVOC(휘발성유기화합물), eCO2(이산화탄소), 온도, 습도 수치를 측정하고 클라우드에 리포팅.

사용자는 스마트폰앱을 통해 센서에서 수집한 공기질을 파악하고 클라우드와 연동된 에어맵을 통해 해당 지역의 공기질 상태를 파악할 수 있음

### 1. 스마트폰과 디자이스 연결에 대한 특허출원

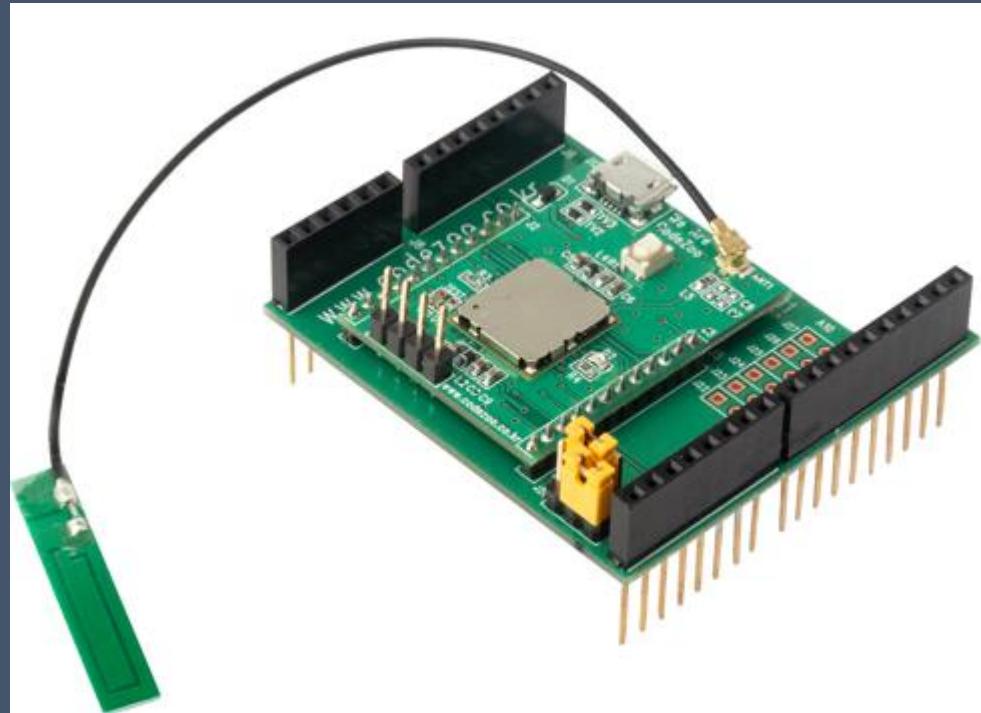
안드로이드 기반 휴대용 단말기의 유에스비 인식방법 및 안드로이드 기반 휴대용 단말기의 유에스비 인식장치 (출원번호 10-2018-0015663)

### 2. 현재 구로구청과 2018 서울산업진흥원 실증사업 진행 중.

- 구로지역 거주민 200명 대상으로 공기질을 측정하여 구로구 에어맵 제공 계획. (2018년 연내)

# 회사소개

## 코드주 로라 쉴드

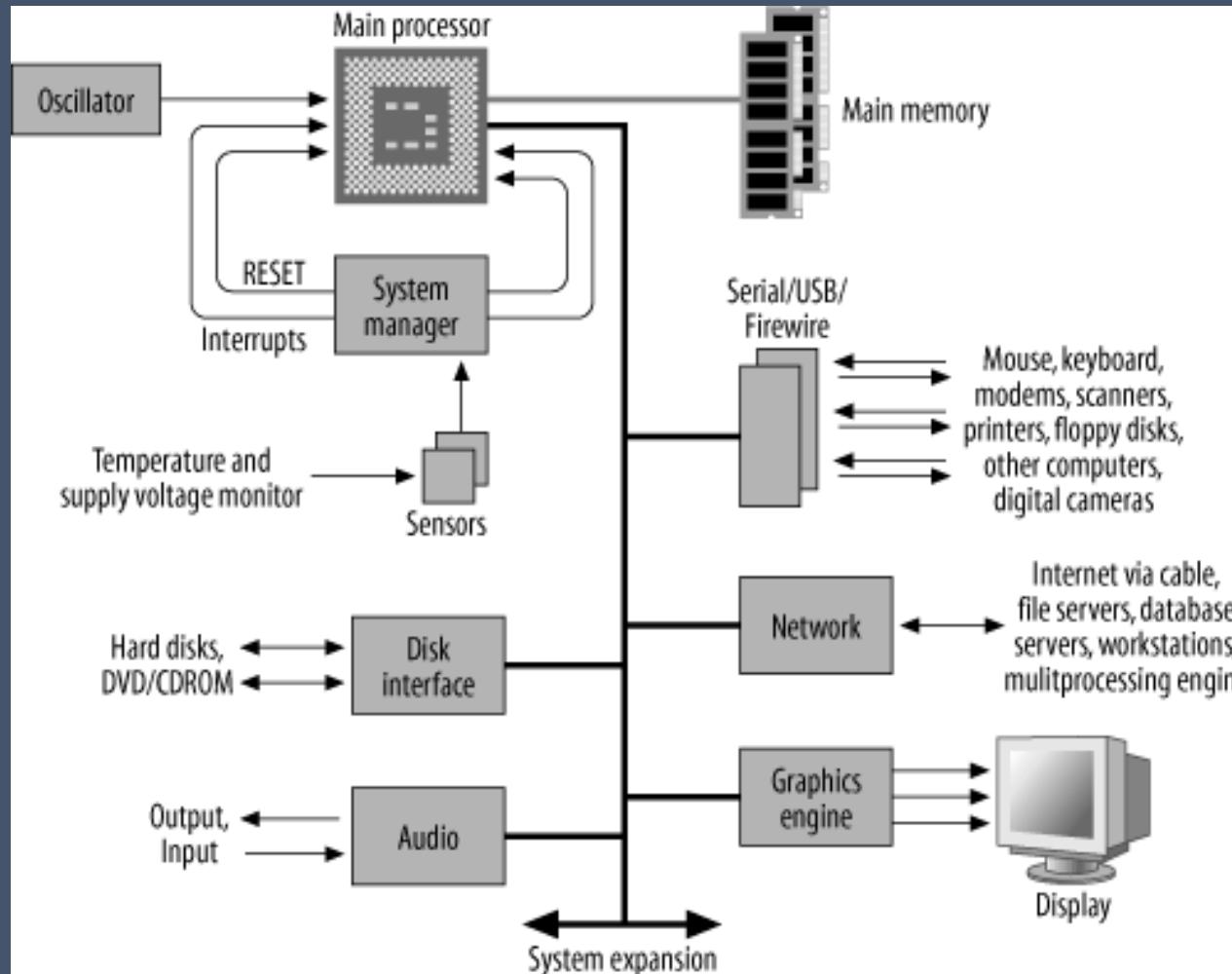


<https://github.com/codezoo-ltd/SNIPE>



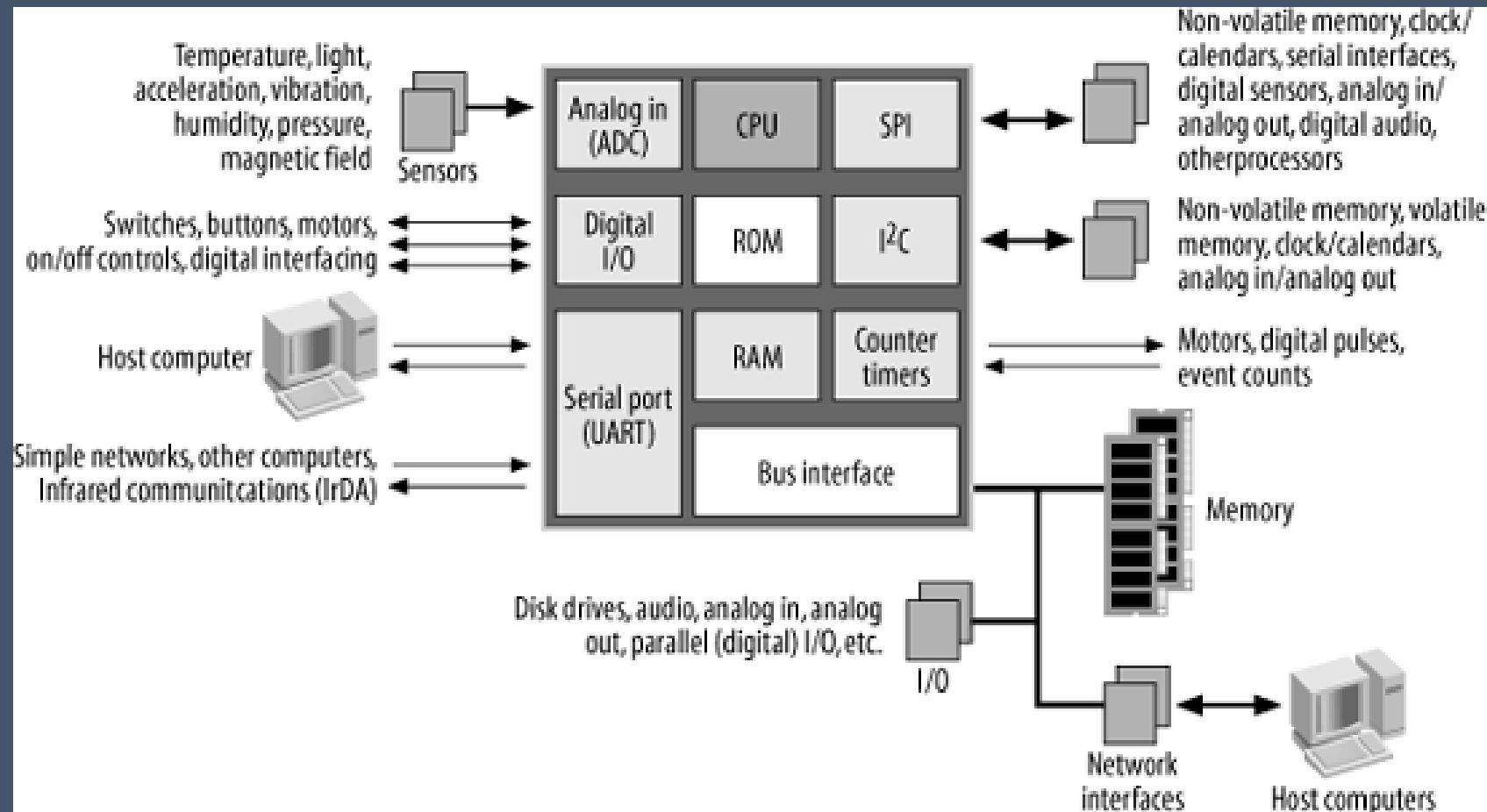
# PC 하드웨어

## PC(Personal Computer)



# 임베디드 시스템 하드웨어

## Embedded System



# ARM

## ARM Holdings

영국의 반도체 회사

- 프로세서를 설계하고 라이센싱
- 소프트웨어 개발 도구 설계 및 판매(케일 등)

## ARM Architecture

RISC 프로세서

Joint venture between  
Acorn Computers and Apple



1990

Designed into first mobile  
phones and then smartphones



1993 onwards

Now all electronic devices can  
use smart Arm technology



Today

# ARM Core

## Cortex-A

Highest performance

Optimized for  
high-level operating  
systems



## Cortex-R

Fast response

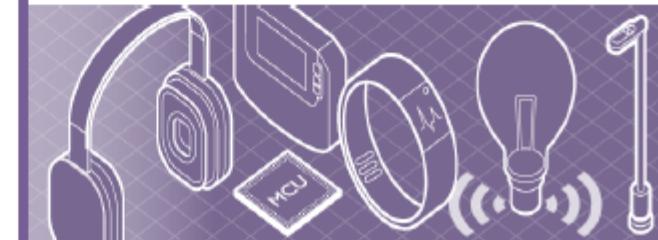
Optimized for  
high performance,  
hard real-time  
applications



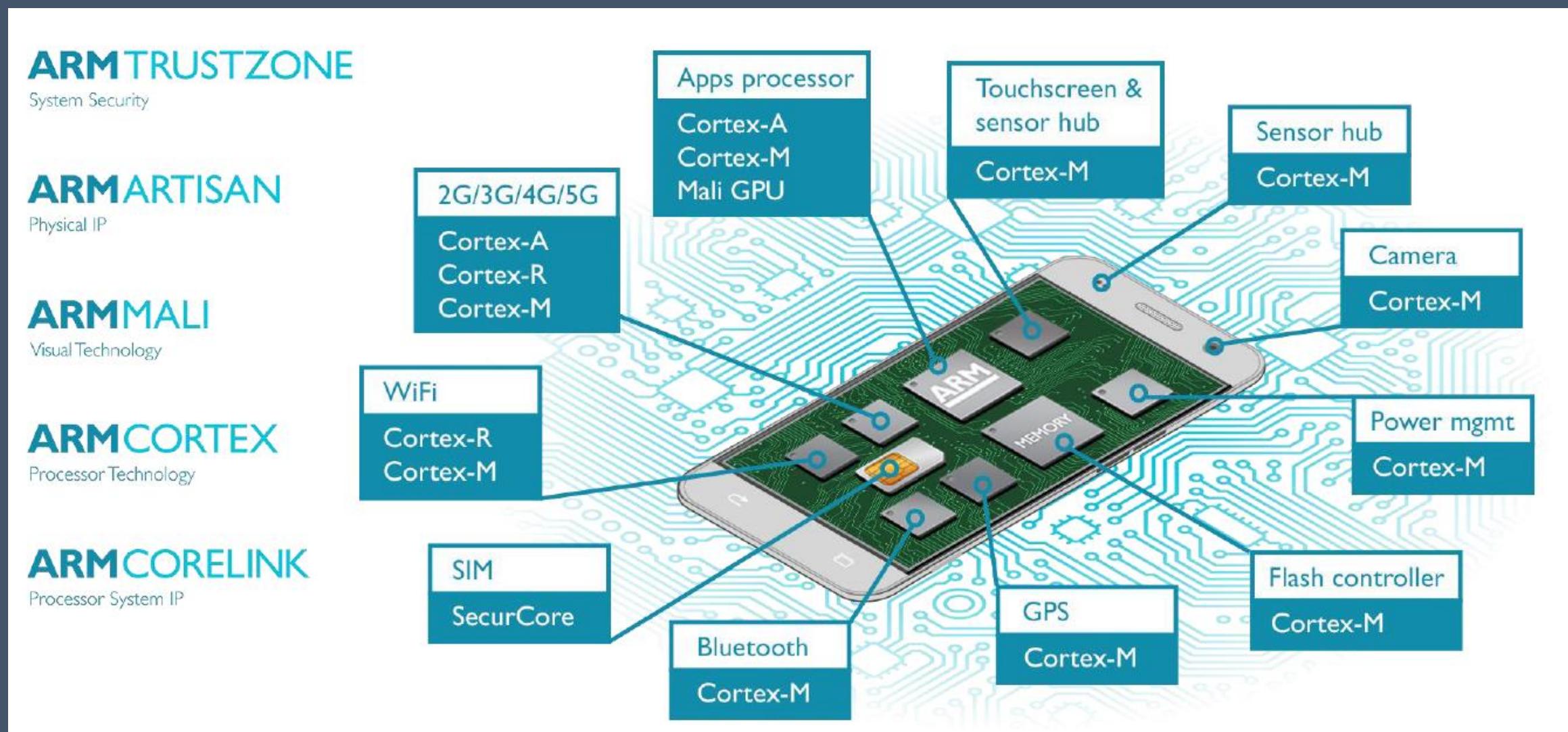
## Cortex-M

Smallest/lowest power

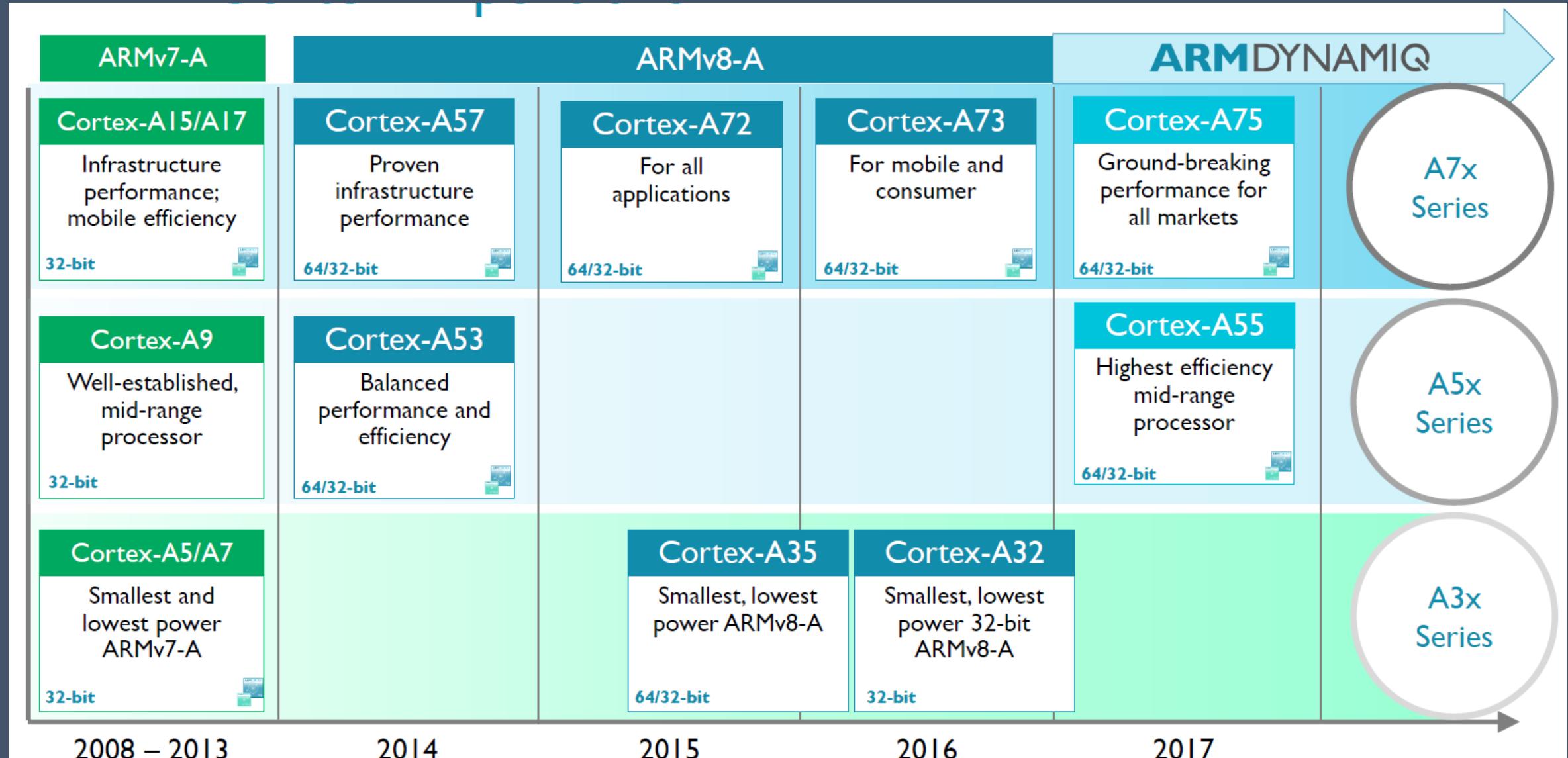
Optimized for  
discrete processing  
and  
microcontrollers



# ARM Core



# ARM Core



# ARM Core

## Cortex-R4

Real-time performance

## Cortex-R5

Real-time performance and peripheral control

## Cortex-R7

High performance 4G modem and storage

## Cortex-R8

Highest performance 5G modem and storage

Storage and modem

## Cortex-R5

Real-time performance with functional safety

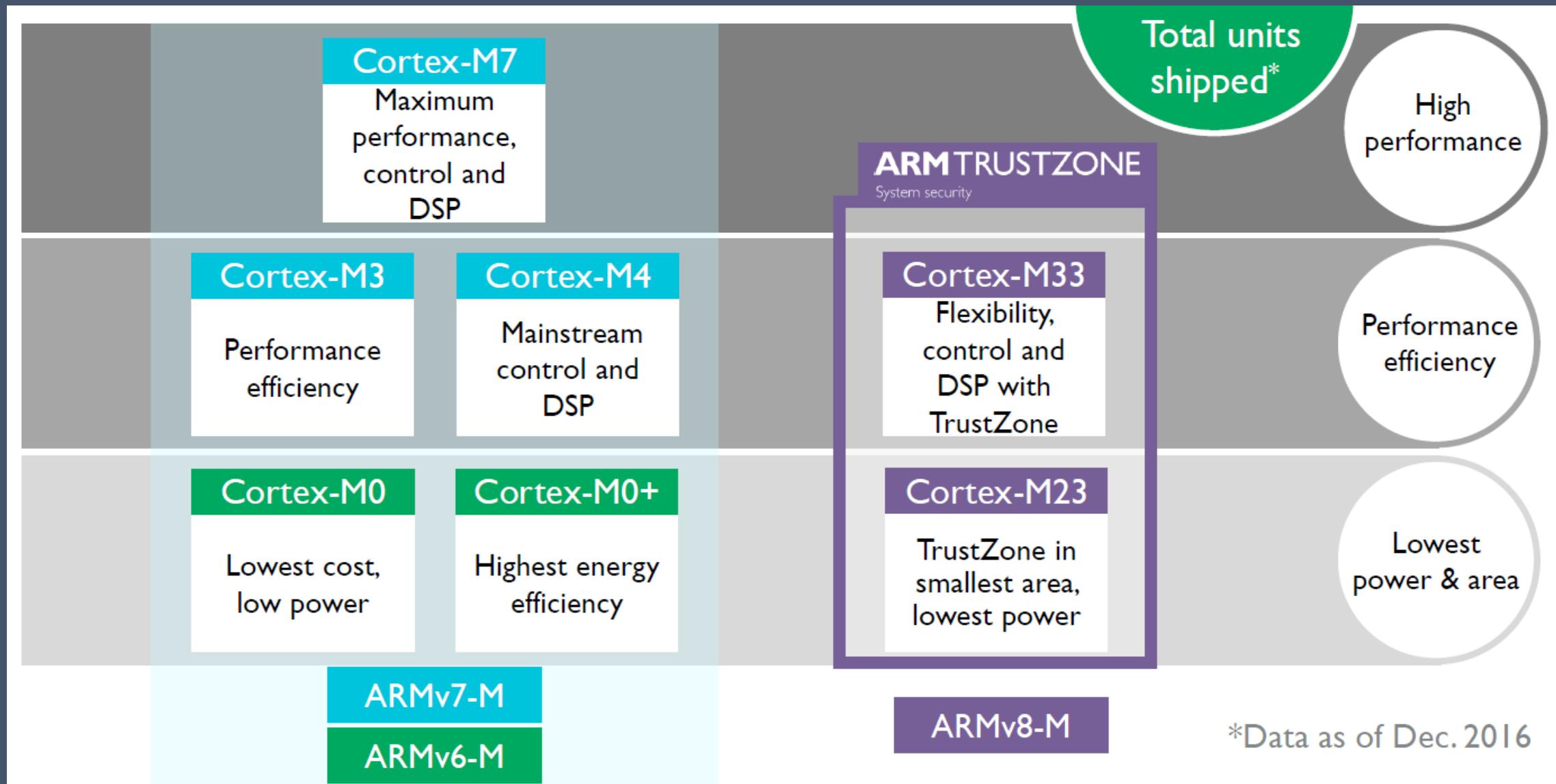
## Cortex-R52

Most advanced processor for functional safety

ARMv7-R

ARMv8-R

# ARM Core



# ARM Core

	Cortex-A	Cortex-R	Cortex-M
설계	고주파 클럭, 긴 파이프 라인, 고성능, 멀티미디어 지원 (NEON 명령어 세트 확장)	고주파 클럭, 중간에서 긴 길이의 파이프 라인, 확정적 (저지연 인터럽트)	보통 더 짧은 파이프라인, 초저전력, 확정적 (저지연 인터럽트)
시스템 기능	메모리 관리 장치(MMU), 캐시 메모리, ARM TrustZone 보안 확장	메모리 보호 장치(MPU), 캐시 메모리, 밀착 결합 메모리(Tightly Coupled Memory)	메모리 보호 장치(MPU), 중첩 벡터형 인터럽트 컨트롤러(NVIC), 웨이크업 인터럽트 컨트롤러(WIC), 최신 ARM TrustZone 보안 기능 확장
용도	모바일 컴퓨팅, 스마트폰, 에너지 효율 서버, 하이엔드 마이크로프로세서	산업용 마이크로컨트롤러, 자동차, 하드 디스크 컨트롤러, 베이스밴드 모뎀	마이크로컨트롤러, 심층 임베디드 (Deeply embedded) 시스템 (예: 센서, MEMS, 혼합 신호 IC), 사물인터넷 (IoT)

# Cortex-M

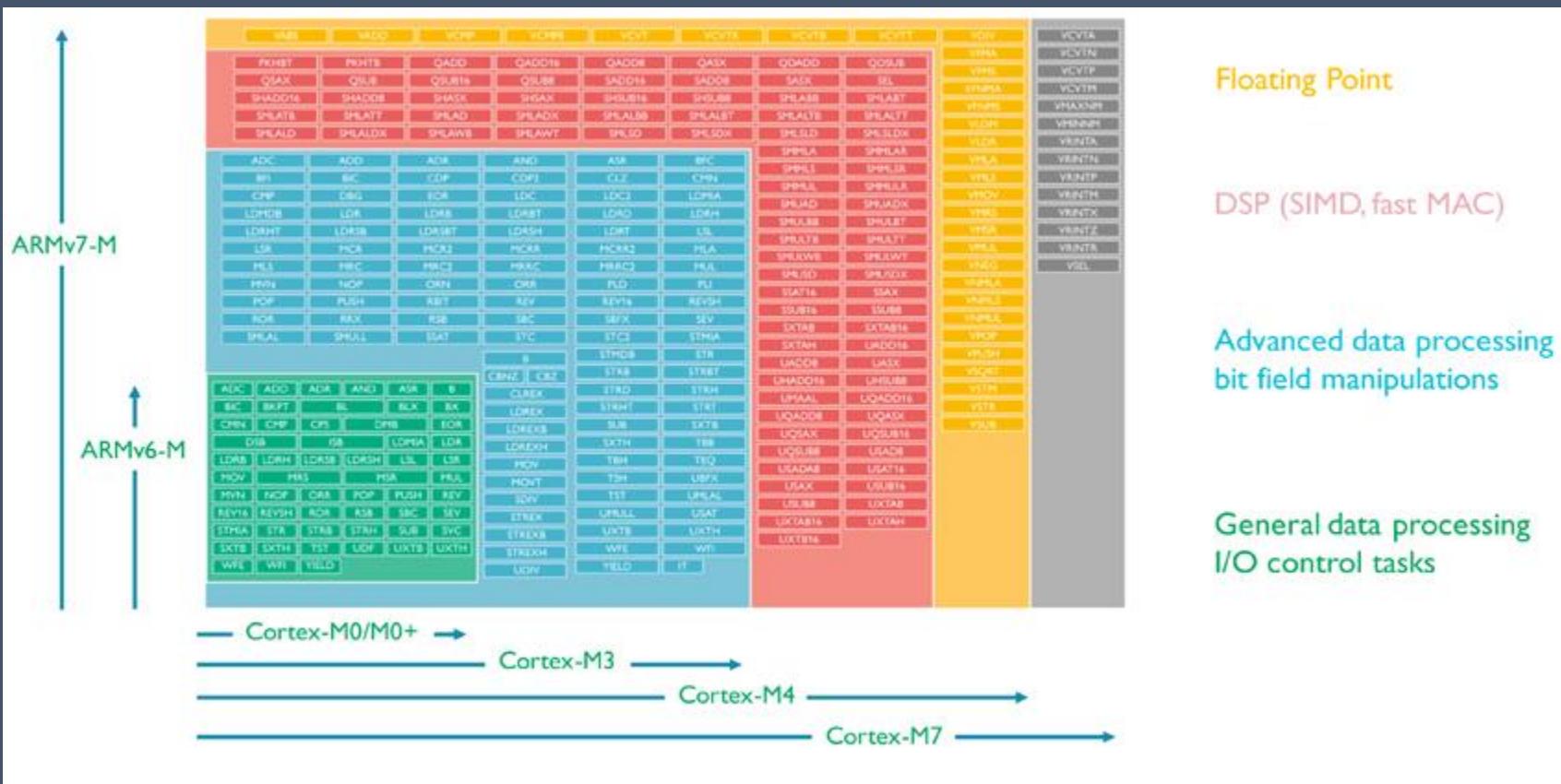
프로세스	설명
Cortex-M0	저비용, 초저전력 마이크로컨트롤러 및 고도의 임베디드 애플리케이션용 초소형 프로세서(12K 게이트에서 시작)
Cortex-M0+	소형 임베디드 시스템을 위한 최고 수준의 에너지 효율 프로세서. Cortex-M0 프로세서와 비슷한 크기와 프로그래머 모델이지만 단일 사이클 I/O 인터페이스와 벡터 테이블 재배치 등 추가 기능이 포함되어 있다.
Cortex-M1	FPGA 설계에 최적화된 소형 프로세서 설계이며 FPGA의 메모리 블록을 통해 밀착 결합 메모리(Tightly Coupled Memory)를 구현한다. Cortex-M0과 동일한 명령어 세트
Cortex-M3	복잡한 작업을 보다 신속하게 처리할 수 있도록 많은 명령어가 설정된 저전력 마이크로 컨트롤러를 위한 작지만 강력한 임베디드 프로세서. 하드웨어 디바이더(Hardware Divider)와 Multiply-Accumulate(MAC) 명령어를 탑재하고 있다. 이와 함께 소프트웨어 개발자가 애플리케이션을 더 빠르게 개발할 수 있도록 포괄적인 디버그 및 추적 기능을 지원한다.
Cortex-M4	이 프로세서는 Cortex-M3의 모든 기능을 제공하며, Single Instruction Multiple Data(SIMD)와 보다 빠른 단일 사이클 MAC 연산과 같은 디지털 신호 처리(Digital Signal Processing) 작업에서 추가 명령어 대상을 제공한다. 또한 IEEE 754 부동 소수점 표준을 지원하는 <b>단정도(Single Precision) 부동 소수점 유닛</b> (옵션)도 제공한다.
Cortex-M7	하이엔드 마이크로컨트롤러 및 프로세싱 집약적인 애플리케이션용 고성능 프로세서. Cortex-M4에서 사용할 수 있는 모든 ISA 기능과 캐시 및 밀착 결합 메모리(Tightly Coupled Memory)와 같은 추가 메모리 기능뿐만 아니라 <b>배정도(double precision) 부동 소수점</b> 을 위한 추가 지원을 제공한다.
Cortex-M23	초저전력과 저비용 설계용으로 설계된 소형 프로세서로 <b>Cortex-M0+ 프로세서와 비슷하지만 명령어 세트 및 시스템 수준의 기능이 다양하게 향상됐다.</b> 이와 함께 TrustZone 보안 기능 확장을 지원한다.
Cortex-M33	기존 <b>Cortex-M3 및 Cortex-M4 프로세서와 비슷하지만 더욱 향상된 유연성을 갖춘 시스템 설계와 보다 나은 에너지 효율성과 성능을 제공하는 주류 프로세서 설계.</b> 또한 이 프로세서는 TrustZone 보안 확장을 지원한다.

# Cortex-M

Architecture	설명
ARMv6-M	Cortex-M0, Cortex-M0+ 및 Cortex-M1 프로세서용.
ARMv7-M	Cortex-M3, Cortex-M4 및 Cortex-M7 프로세서용. DSP 유형의 명령어 (예: SIMD)를 지원하기 위한 ARMv7-M의 확장은 ARMv7E-M으로 명명된다.
ARMv8-M	이 아키텍처 릴리스는 다음과 같이 나누어진다. Cortex-M23 프로세서용 Baseline 서브 프로파일 Cortex-M33 프로세서용 Mainline 서브 프로파일

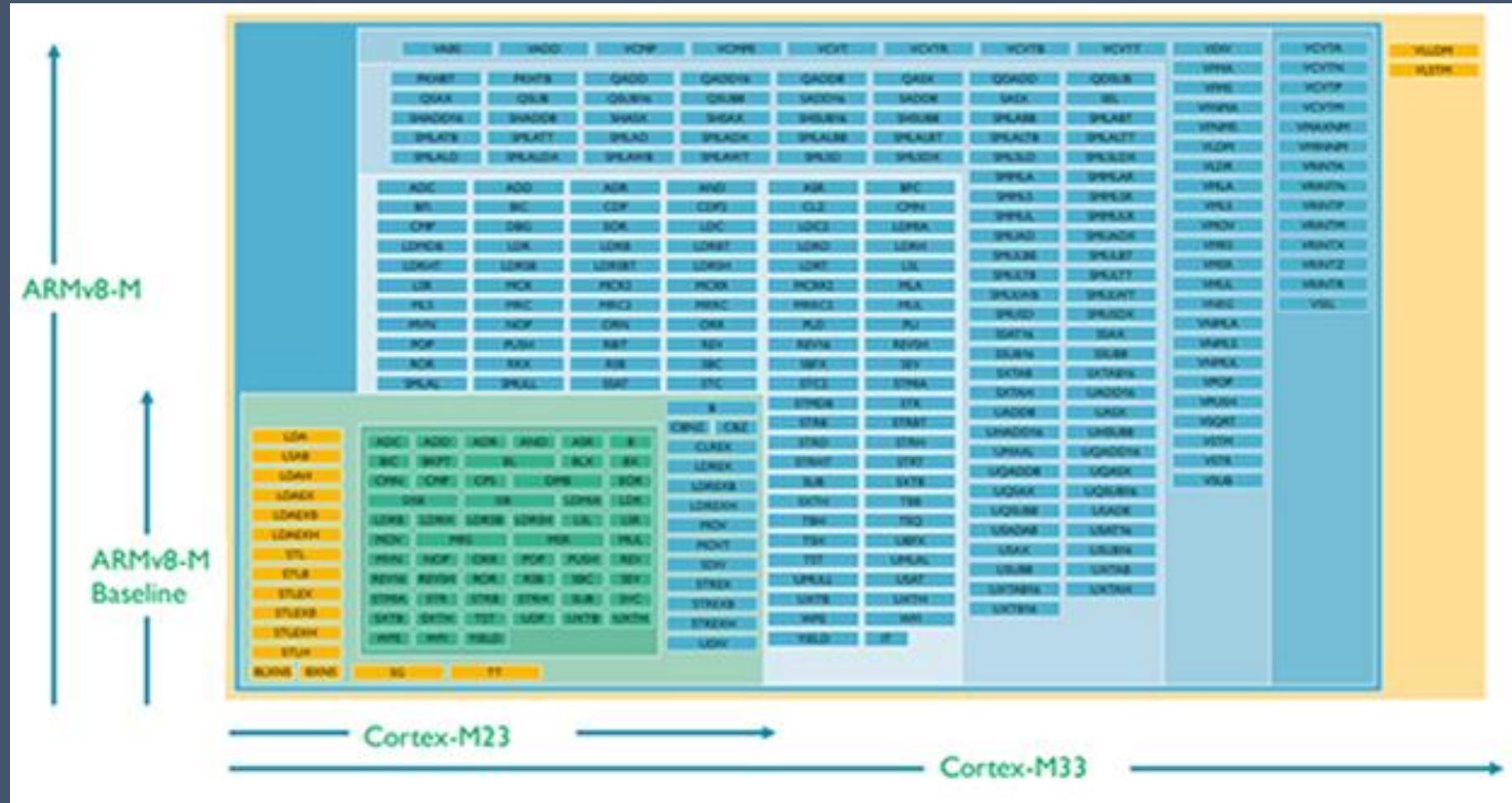
# *Cortex-M*

## Command Set



# Cortex-M

## Command Set



# Cortex-M



# ARM Processor 종류



# **STM32**

## **STMicroelectronics**

스위스 제네바에 본사를 둔 전자제품과 반도체를 생산하는 기업



## **STM32**

32-bit ARM Cortex-m processor core를 사용하는 STMicroelectronics에서 생산하는 Microcontroller 제품군



# STM32

## High-performance

STM32H7 series – High performance with DSP, Double-precision FPU, JPEG Codec and Chrom-ART Accelerator™										
400 MHz Cortex-M7 L1-Cache	Up to 2-Mbyte dual-bank Flash	Up to 1-Mbyte SRAM	2x USB 2.0 OTG FS/HS	2x 16-bit advanced MC timer	DFSDM HDMI-CEC Ethernet S/PDIF	Quad-SPI FMC MDIO Camera IF SDIO	Crypto-hash TRNG	4x SAI 3x I²S 2x FDCAN LCD-TFT	3x 16-bit ADC Op-amps comp.	STM32 H7
STM32F7 series – High performance with DSP, FPU, ART Accelerator™ and Chrom-ART Accelerator™										
216 MHz Cortex-M7 L1-Cache	Up to 2-Mbyte dual-bank Flash	Up to 512-Kbyte SRAM	2x USB 2.0 OTG FS/HS	2x 16-bit advanced MC timer	DFSDM HDMI-CEC Ethernet S/PDIF	Quad-SPI FMC MDIO Camera IF SDIO	Crypto-hash TRNG	2x SAI 2x I²S LCD-TFT Up to 3x CAN	MIPI-DSI	STM32 F7
STM32F4 series – High performance with DSP, FPU, ART Accelerator™ and Chrom-ART Accelerator™										
Up to 180 MHz Cortex-M4	Up to 2-Mbyte dual-bank Flash	Up to 384-Kbyte SRAM	2x USB 2.0 OTG FS/HS	2x 16-bit advanced MC timer	DFSDM HDMI-CEC Ethernet S/PDIF	Quad-SPI FMC MDIO Camera IF SDIO	Crypto-hash TRNG	2x SAI 5x I²S LCD-TFT Up to 2x CAN	MIPI-DSI	STM32 F4
STM32F2 series – High performance with ART Accelerator™										
120 MHz Cortex-M3 CPU	Up to 1-Mbyte Flash	Up to 128-Kbyte SRAM	2x USB 2.0 OTG FS/HS	2x 16-bit advanced MC timer	Ethernet	FSMC Camera IF SDIO	Crypto-hash TRNG	2x I²S Up to 2x CAN		STM32 F2

## Mainstream

STM32F3 series – Mixed-signal with DSP and FPU								
72 MHz Cortex-M4	Up to 512-Kbyte Flash	Up to 80-Kbyte SRAM CCM-RAM	USB 2.0 FS	3x 16-bit advanced MC timer	3x DAC 7x comp. 4x PGA	FSMC CAN	HR-Timer	ADC 3x 16-bit ΣΔ 4x 12-bit (5 MSPS)
STM32F1 series – Mainstream								
Up to 72 MHz Cortex-M3 CPU	Up to 1-Mbyte Flash	Up to 96-Kbyte SRAM	USB 2.0 OTG FS	2x 16-bit advanced MC timer	HDMI-CEC Ethernet	FSMC SDIO	2x I²S 2x CAN	
STM32F0 series – Entry-level								
48 MHz Cortex-M0+ CPU	Up to 256-Kbyte Flash	Up to 32-Kbyte SRAM 20-byte backup data	USB 2.0 FS device Crystal less	Comp. HDMI-CEC	CAN DAC			

## Ultra-Low-Power

STM32L4+ series – Ultra-Low-Power and more Performance with DSP, FPU, ART Accelerator™ and Chrom-ART Accelerator™										
120 MHz Cortex-M4 CPU	Up to 2-Mbyte dual-bank Flash	Up to 640-Kbyte SRAM	USB 2.0 OTG Crystal less	2x 16-bit advanced MC timer	DFSDM Op-amps comp.	2x Octo-SPI FSMC SDIO 2x SAI	SHA-256 AES-256 TRNG CAN	MIPI-DSI LCD-TFT Chrom-GRC™	STM32 L4+	
STM32L4 series – Ultra-Low-Power and Performance with DSP, FPU, ART Accelerator™ and Chrom-ART Accelerator™										
80 MHz Cortex-M4 CPU	Up to 1-Mbyte dual-bank Flash	Up to 320-Kbyte SRAM	USB 2.0 OTG FS	2x 16-bit advanced MC timer	DFSDM Op-amps comp.	Quad-SPI FSMC SDIO 2x SAI	SHA-256 AES-256 TRNG 2x CAN	Up to LCD 8x40	STM32 L4	
STM32L1 series – Ultra-Low-Power										
32 MHz Cortex-M3 CPU	Up to 512-Kbyte Flash	Up to 80-Kbyte SRAM	Up to 16-Kbyte EEPROM	USB 2.0 FS Device	Op-amps comp.	FSMC SDIO	AES-128	Up to LCD 8x40	STM32 L1	
STM32L0 series – Ultra-Low-Power										
32 MHz Cortex-M0+ CPU	Up to 192-Kbyte SRAM	Up to 20-Kbyte SRAM	Up to 6-Kbyte EEPROM	USB 2.0 FS device Crystal less	DAC comp.	LP ADC 12-/16-bit	TRNG AES-128	LCD 8x48 / 4x52	STM32 L0	

## Wireless

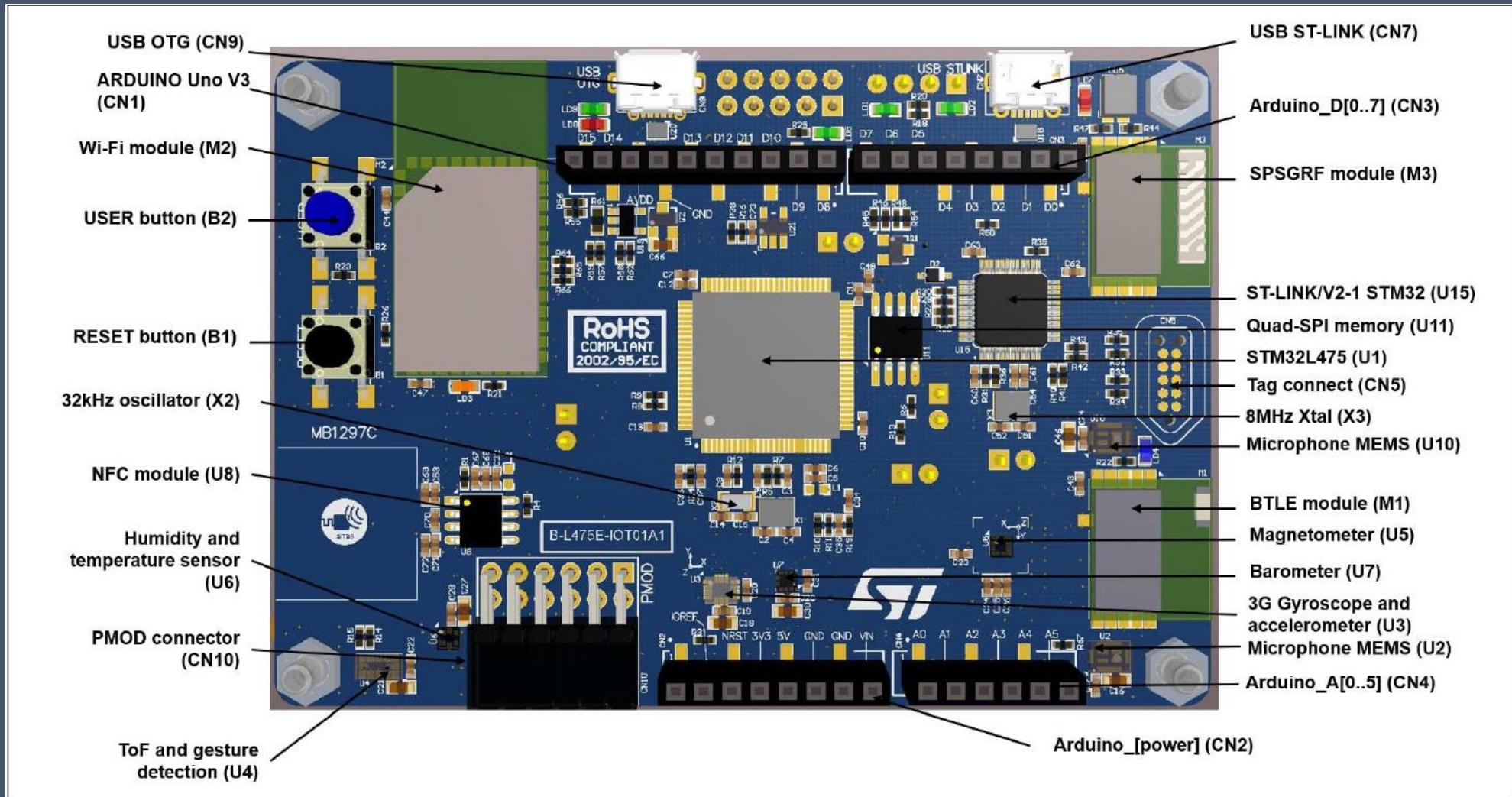
STM32WB series – Multiprotocol and ultra-low-power 2.4 GHz radio with DSP, FPU, ART Accelerator™ and IP Protection									
64 MHz Cortex-M4 CPU	Up to 1-Mbyte Flash	Up to 256-Kbyte SRAM	USB 2.0 FS Crystal less BCD / LPM	1x 16-bit advanced MC timer	Cortex-M0+ BLE 5.0 802.15.4 Concurrent	LP ADC 12x16bit 2x comp.	Quad-SPI 1x SAI (2ch)	PKA AES-256 TRNG CKS*	LCD 8x40 4x44

\* Customer Key Storage



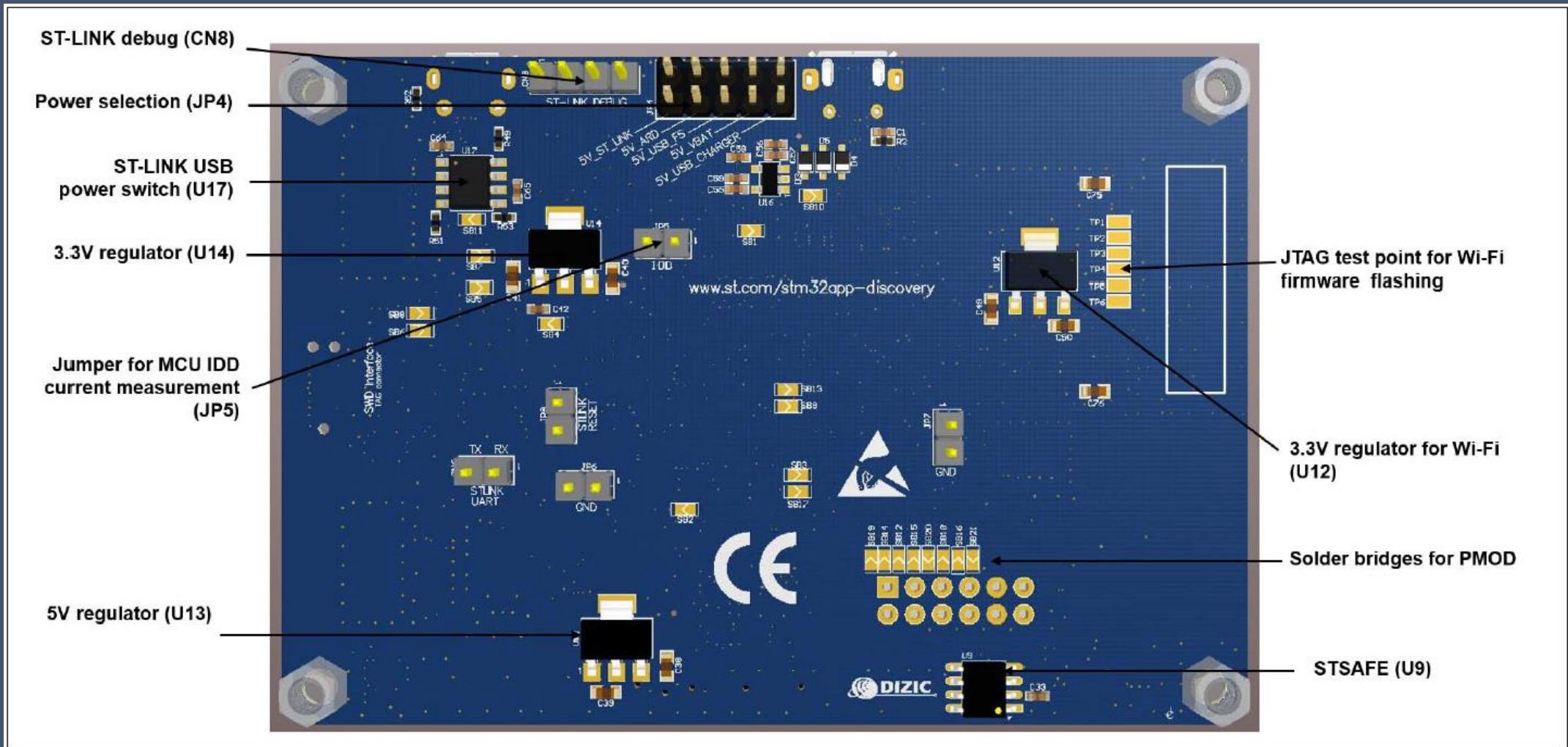
# 실습보드

## B-L475E-IOT01A



# 실습보드

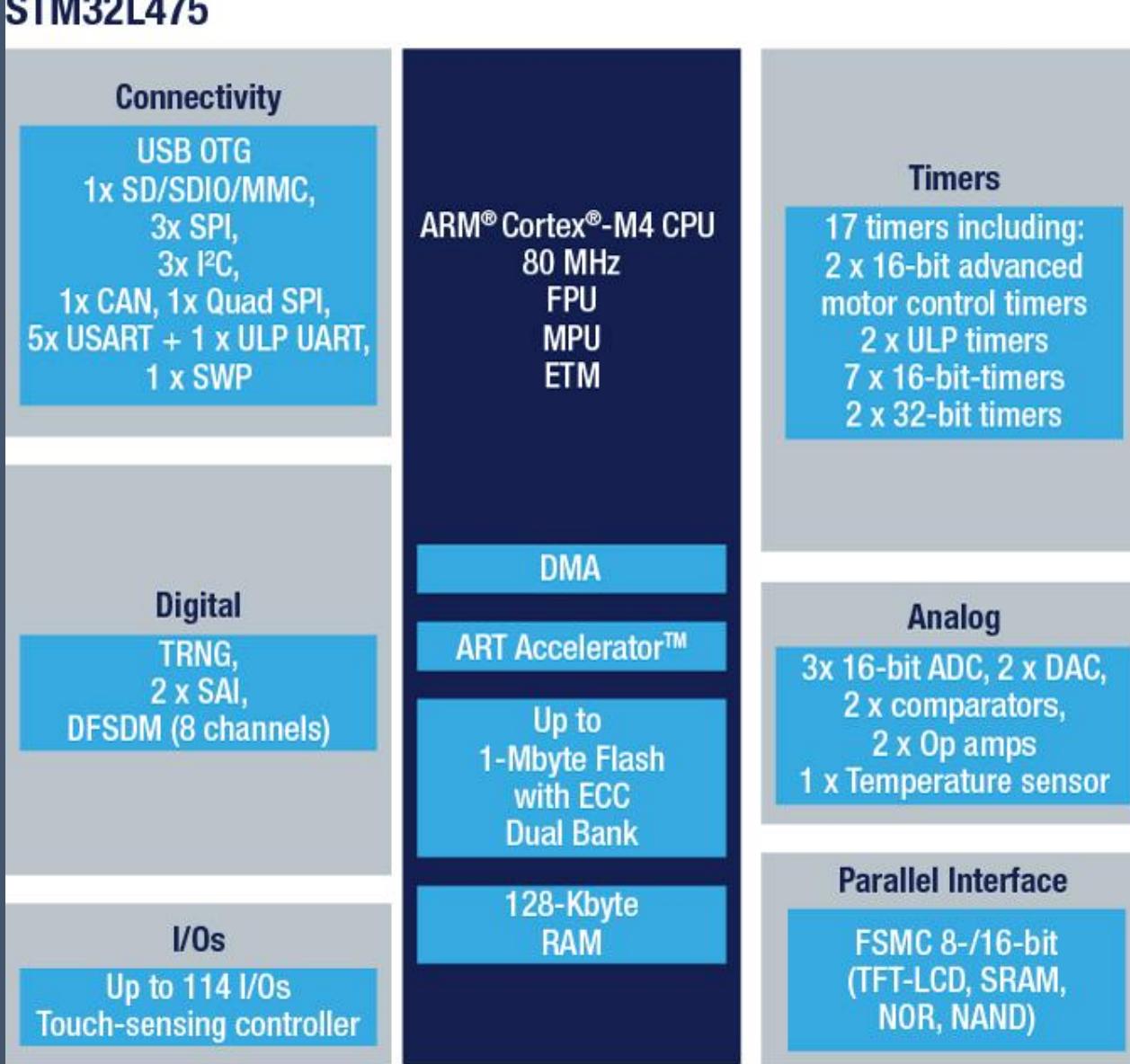
B-L475E-IOT01A



# 실습보드

## STM32L475

### STM32L475



# 실습보드

STM32L475

보드 설명서 다운로드



# Open-source Hardware Platform

- Open source Hardware
- Arduino
- ARM Mbed OS

## Open-source Hardware

- **Open Source Hardware (OSHW) 란?**

각종 HW 제작에 피요한 자료(회로도, 기판도면, 관련 설명 등)을 공개해서 누구나 이 디자인에 근거한 제품을 수정, 배포, 제조를 할 수 있도록 하는 H/w

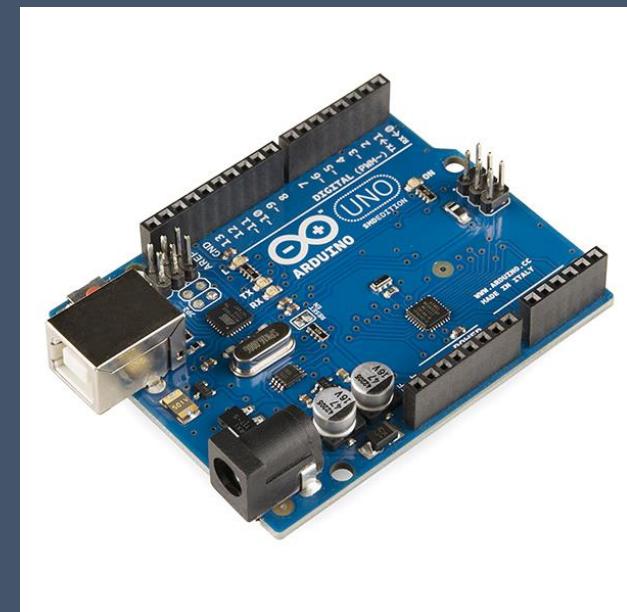
- ❖ OSHW 정의 ( <https://www.oshwa.org/definition/korean/> )

# Arduino

## ● Arduino

- 2005년 이탈리아의 IDII(Interaction Design Institutevera)에서 개발
- 하드웨어에 익숙하지 않은 학생들의 디자인 작품에 이용하기 위한 목적
- 처음은 AVR을 이용
- 지원되는 프로세서가 증가 중
- 확장보드인 Shield를 제공
- 통합 개발환경(IDE)제공
- 펌웨어 프로그램이 쉬움
- 종류 (<https://www.arduino.cc/en/Main/Products>)

아두이노 보드 [1]		
	MCU	
AVR	ATmega168	Pro(168), Mini(168), LilyPad (168V)
	ATmega328	UNO, ETHERNET, Fio, Nano, Pro(328), Mini(328, Rev5), Pro Mini, LilyPad/Simple (328V)
	ATmega2560	Mega 2560, Mega ADK
	ATmega32U4	YÙN, Leonardo, Esplora, Micro, LilyPad USB, ROBOT(Control,Motor)
	ATtiny85	GEMMA
ARM	Cortex-M0+	Zero, Zero Pro, M0, M0 PRO
	Cortex-M3	Due



# Arduino



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.5". The main window displays the "Blink" example code. The code is as follows:

```
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)  
    delay(1000);                      // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);       // turn the LED off by making the voltage LOW  
    delay(1000);                      // wait for a second  
}
```

The status bar at the bottom right indicates "Arduino/Genuino Uno on COM1".

# *ARM Mbed OS*

- Mbed OS

- ARM사에서 주도적으로 진행
- ARM Cortex-M 코어가 탑재된 시리즈 지원 (M0, M0+, M3, M4, M7)
- IoT platform
  - ✓ RTOS(mbed os)
  - ✓ Cloud(mbed pelion)
- Web-IDE, CLI 환경제공
- Cortex-M MCU를 생산하는 많은 회사들이 참여 중

# Arduino to mbed driver porting (1) I2C



**Grove - LCD RGB Backlight**

```
<Arduino>
void *__I2C_Device;

void i2c_send_byte(uint8 dta)
{
    suli_i2c_write(__I2C_Device, LCD_ADDRESS, &dta, 1);
}

void i2c_send_byteS(uint8 *dta, uint8 len)
{
    suli_i2c_write(__I2C_Device, LCD_ADDRESS, dta, len);
}

<MBED>
I2C i2c(PB_9, PB_8);          //sda, scl

void i2c_send_byte(char dta)
{
    i2c.write(LCD_ADDRESS, &dta, 1);      // transmit to device #4
}

void i2c_send_byteS(char *dta, char len)
{
    i2c.write(LCD_ADDRESS, dta, len);      // transmit to device #4
}
```

## Arduino to mbed driver porting (2) Serial



Sierra Wireless LTE Modem

```
#ifdef TARGETMBED
#include <mbed.h>
#include "Countdown_mbed.h"
#include <string>
#define DEBUG_SERIAL_CLASS
#define MODULE_SERIAL_CLASS
#define MODULE_SERIAL_OBJECT
#define COUNTDOWN_CLASS
#define GET_ELLAPSED_MS
#define PORT_TYPE
#define F(arg)
#define SERIAL_SET_BAUD_RATE
#define SERIAL_READ_AVAILABLE
#define SERIAL_READ_CHAR
#define SERIAL_Writeln_STRING(obj, arg)
#define STRING_CLASS
#define DELAY_MS
#endif

#endif TARGETMBED

#endif TARGETARDUINO
#include <SPI.h>
#include <SoftwareSerial.h>
#include "Countdown.h"
#define DEBUG_SERIAL_CLASS
#define MODULE_SERIAL_CLASS
#define MODULE_SERIAL_OBJECT
#define COUNTDOWN_CLASS
#define GET_ELLAPSED_MS
#define PORT_TYPE
#define SERIAL_SET_BAUD_RATE
#define SERIAL_READ_AVAILABLE
#define SERIAL_READ_CHAR
#define SERIAL_Writeln_STRING(obj, arg)
#define STRING_CLASS
#define DELAY_MS
#endif
```

Serial  
Serial  
Serial  
Serial  
Countdown\_mbed  
time(NULL)  
PinName  
arg  
baud  
readable  
getc  
obj->printf(arg); obj->printf("WrWn")  
string  
wait\_ms

SoftwareSerial  
HardwareSerial  
Serial1  
Countdown  
millis();  
int  
begin  
available  
read  
obj->println(arg)  
String  
delay

# Arduino 와 mbed

- Arduino

- 국내는 압도적 (코딩교육과 메이커 운동의 선봉)
- 이용할 수 있는 개발 리소스가 많다 (센서, 액츄에이터)
- 교육 영상, 자료를 구하기가 쉽다

- mbed

- 국내는 사실 비주류 (아두이노로도 충분하다)
- 이용할 수 있는 개발 리소스가 아두이노에 비해 상대적으로 적다
- 교육 영상, 자료를 구하기가 어렵다
- 개발보드를 구하기가 어렵다 (해외구매 필요)

- 비슷한 점

- CPU Core, Driver와 관련된 구현부를 추상화
- 센서, 통신, 액츄에이터 관련 드라이버가 비슷하다 (GPIO, Serial, SPI, I2C)

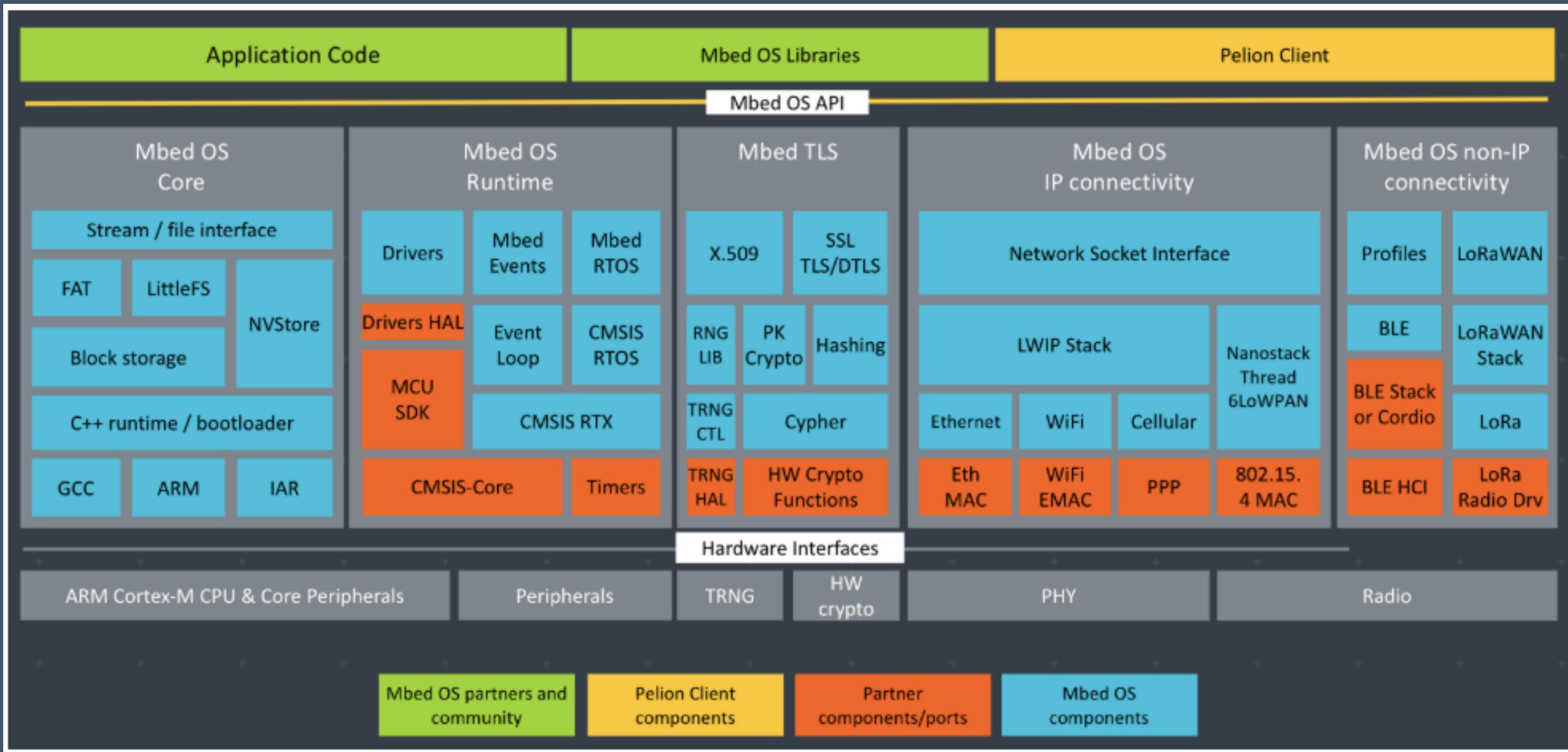
- Arduino 와 mbed

- 경쟁관계라기 보다 지향점이 다름 (메이커 vs IoT Service Platform)
- 모두 훌륭한 플랫폼

# Mbed OS

- Mbed OS ?
- mbed OS 개발환경 소개
- mbed OS 프로그래밍 흐름
- mbed OS 시작하기
- mbed OS 데이터형
- mbed.h 분석

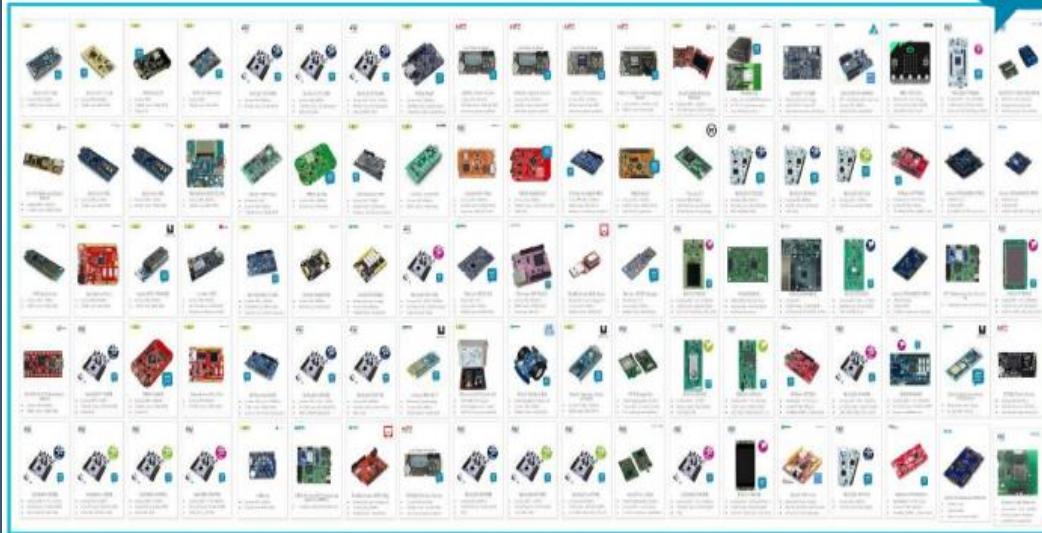
# Mbed OS



# Mbed OS ?

mbed OS는 IoT제품의 빠른 개발을 위한 ARM사에서 제공하는 오픈 IoT플랫폼으로 Cortex-M Core를 사용하는 MicroController를 대상으로 한다.

# Mbed Board



## Mbed Components



# Mbed OS 개발환경 소개

## Mbed OS web compiler

- Board Select
- Compile
- Import/Export
- Source version control
- ARM compiler based

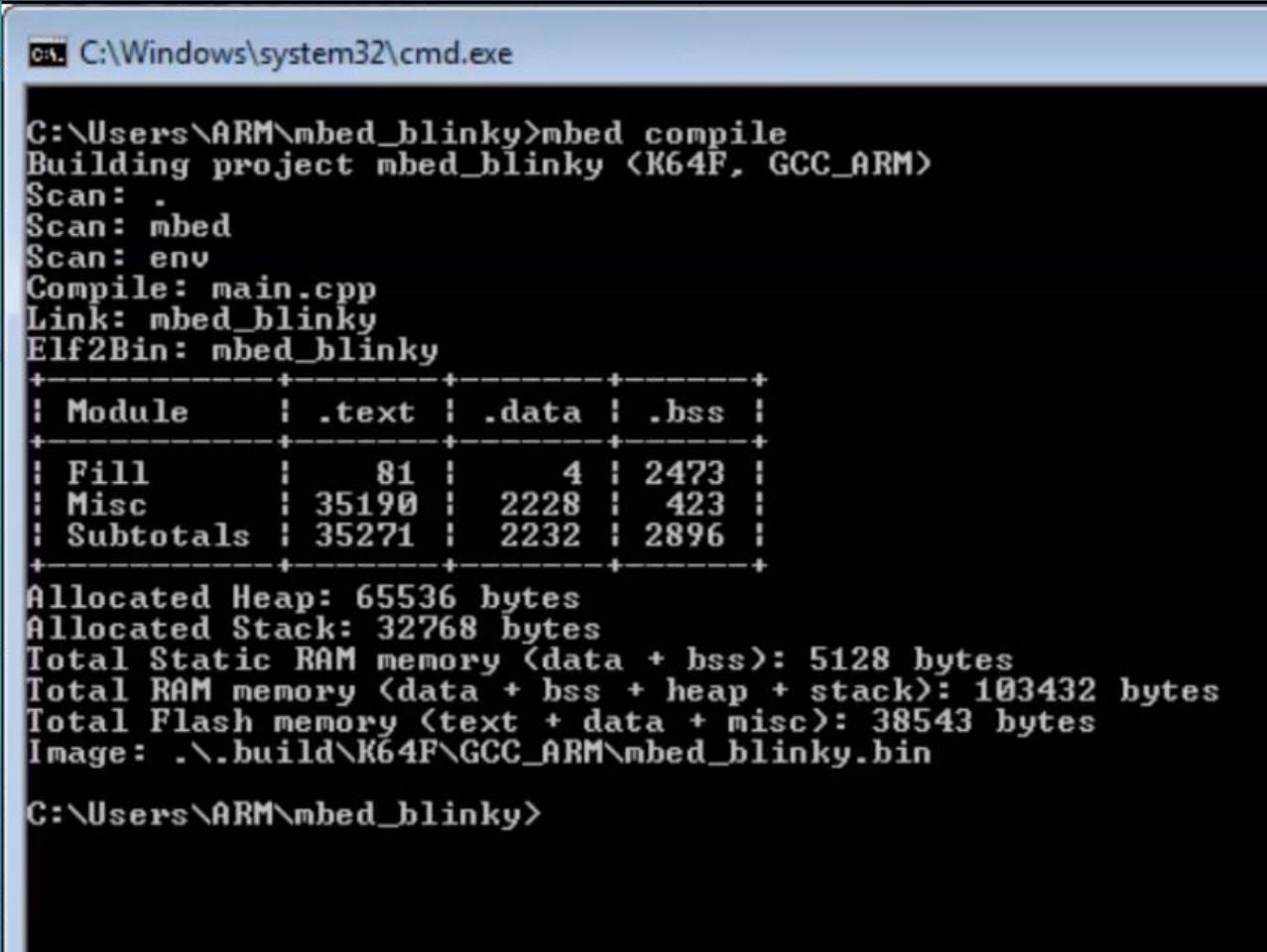
The screenshot shows the 'Workspace Management' section of the Mbed OS Web Compiler. On the left, there's a tree view of 'My Programs' containing numerous project icons. The main area displays a table titled 'Manage your Program Workspace' with columns for Name, Tags, Modified, and Description. The table lists over 60 projects, mostly named 'main.cpp', with various modification dates from 2014 to 2017. To the right, there's a 'Workspace Details' panel showing the user 'jbmaster' has 69 total programs and was modified 22 hours, 22 minutes ago. A 'Recently Modified' list shows several 'main.cpp' files with their last modification dates.

Name	Tags	Modified	Description
ADKTerm		23 Oct 2014	
AndroidAccessory		06 Nov 2014	
bb_uart_test		04 Sep 2015	
BLE_Color_Pixels_arch		17 Dec 2014	
BLE_HeartRate		17 Dec 2014	
BLE_Beacon		30 Jan 2015	
BLE_LoopbackUART		17 Aug 2015	
BLE_UARTConsole		26 Dec 2014	
Bluetooth_demo		16 Oct 2014	
BrainBridge_SimpleC		17 Aug 2015	
C027_HelloWorld		26 Dec 2014	
C027_I2C_Test		27 Oct 2014	
C027_LoRa		27 Apr 2017	
C027_ModemTransp		14 Oct 2014	
C027_SupportTest		18 Apr 2016	
C027_SupportTest_2		28 Jan 2017	
DISCO-L072C2-LRW		26 Sep 2017	
DISCO-L072C2-LRW		27 Jul 2017	
Discovery401_toggle		17 Aug 2016	
Discovery_blink_led		18 Aug 2016	
Dragonfly_Cellular_r		23 Mar 2016	
frdm_rtos		15 Dec 2014	
GTC_BLE		06 Feb 2015	
GTC_BLE_Nucleo_pr		07 Feb 2015	
GTC_Cable_Test		22 Jun 2015	
GTC_FreeTEST_Key		09 Feb 2015	
GTC_Measurement		24 Jun 2015	
H005_Transparent_r		24 Nov 2014	
HelloWorld_IKS01A1		07 Oct 2016	
HTTPClient_Cellular_		17 Dec 2015	
I2C_Demo		25 Dec 2014	
L073_ATCMD_Master		03 May 2017	
L073_ATCMD_slave		17 May 2017	
L073RZ_I2c_master			
L073RZ_Printf			
LoRaWAN-Demo-76			
mbed-os-example-bl			
mbed-os-example-bl			
mbed_blinky			
mbed_blinky_arch_b			
Motor_Speed_Test_f			
murata_Lora_PingP			
Murata_Node			
Murata_Server			
nRF51822_SimpleCo			
Nucleo73_display_tir			
Nucleo_blink_color_			

# Mbed OS 개발환경 소개

## Mbed CLI

- Command Line Interface
- Compile
- Import Source, Add library
- Select compiler (ARM, GCC, etc)



```
C:\Windows\system32\cmd.exe
C:\Users\ARM\mbed_blinky>mbed compile
Building project mbed_blinky (K64F, GCC_ARM)
Scan: .
Scan: mbed
Scan: env
Compile: main.cpp
Link: mbed_blinky
Elf2Bin: mbed_blinky
+-----+-----+-----+-----+
| Module | .text | .data | .bss |
+-----+-----+-----+-----+
| Fill   |    81 |     4 | 2473 |
| Misc   | 35190 | 2228 |   423 |
| Subtotals | 35271 | 2232 | 2896 |
+-----+-----+-----+-----+
Allocated Heap: 65536 bytes
Allocated Stack: 32768 bytes
Total Static RAM memory (data + bss): 5128 bytes
Total RAM memory (data + bss + heap + stack): 103432 bytes
Total Flash memory (text + data + misc): 38543 bytes
Image: ..\build\K64F\GCC_ARM\mbed_blinky.bin

C:\Users\ARM\mbed_blinky>
```

# Mbed OS 맛보기

<https://labs.mbed.com>

arm MBED

## ocab Mbed Labs

Experimental projects from the Arm Mbed team

**Arm Mbed OS simulator**



Serial output

```
#include "stdio.h"
DigitalOut ledPin5;
Leds led;
while(1) {
    led = !led;
    printf("LED %d\n", led.read());
    wait_ms(500);
}
```

**Mbed OS Simulator**

Write, test and debug Mbed OS 5 applications in your browser.

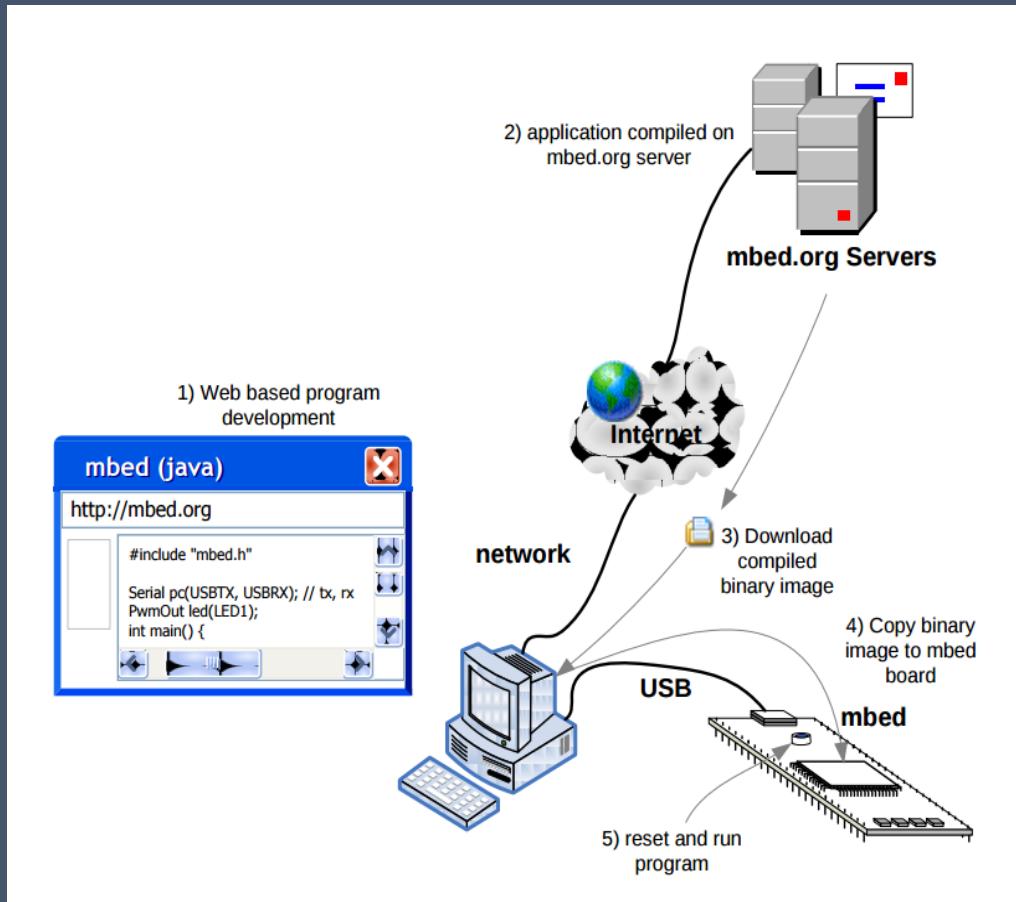
**JavaScript on Mbed**



Build your IoT applications in JavaScript, and run them on any Mbed-enabled microcontroller.

		
LoRaWAN FUOTA	Bluetooth Devicelink	uTensor
Multicast firmware updates over the air using LoRaWAN.	Connect BLE devices to the cloud through Pelion Device Management.	Deep-learning on microcontrollers by running TensorFlow models on

# mbed OS 프로그래밍 흐름 (Web IDE)

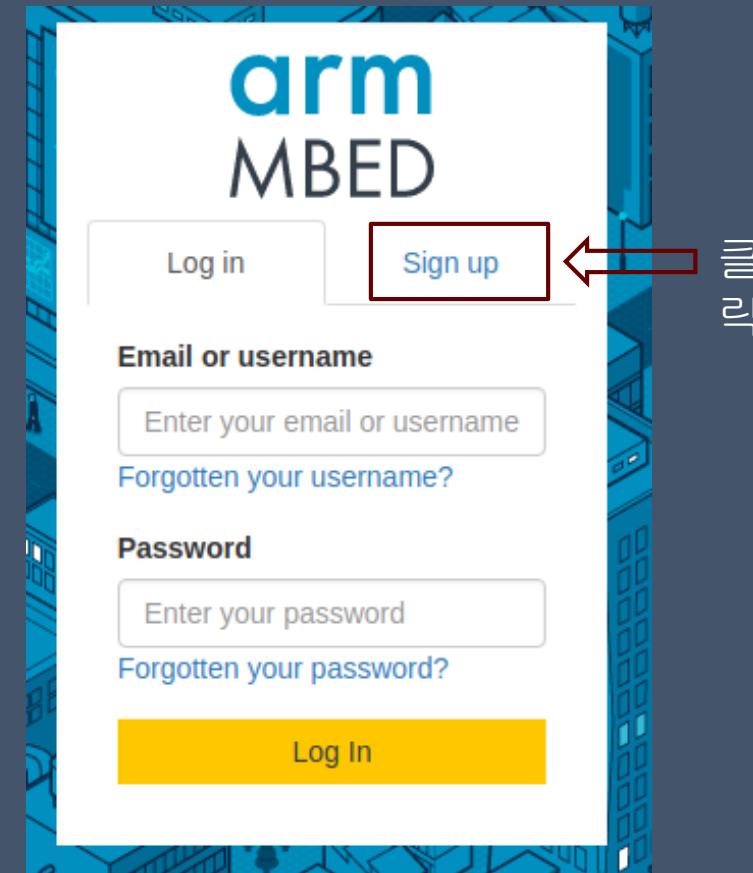
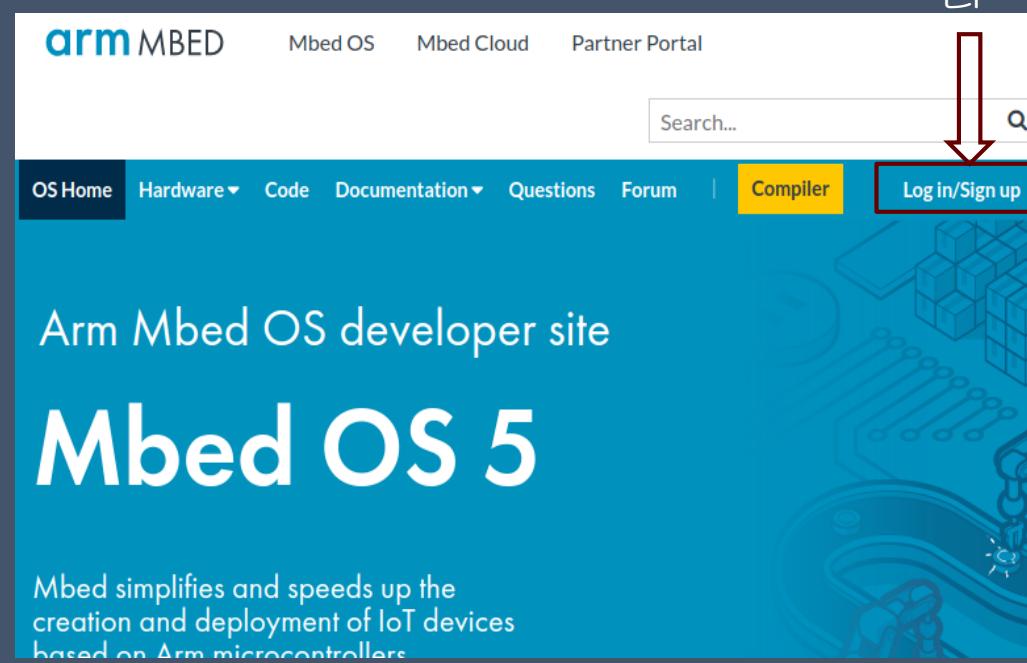


1. 웹기반 mbed 개발툴을 이용해서 프로그램 작성
2. 작성한 프로그램을 개발툴에서 컴파일하게 되면, mbed.com 서버를 통해 프로그램이 컴파일
3. 컴파일된 실행파일을 사용자 PC로 다운로드
4. 다운로드 받은 실행파일을 mbed 보드에 복사
5. mbed 보드 리셋버튼을 누르고 프로그램을 실행

# mbed 시작하기

## mbed 가입하기

<https://os.mbed.com/>에 접속



## DISCO-L475VG-IOT01A 와 PC 연결



windows의 경우 board가 인식이 안되면  
<https://os.mbed.com/docs/latest/tutorials/windows-serial-driver.html>  
"Arm Mbed Windows serial port driver" 를 download 후 설치

# DISCO-L475VG-IOT01A

<https://os.mbed.com/platforms/ST-Discovery-L475E-IOT01A/>

## DISCO-L475VG-IOT01A

STM32L4 Discovery kit IoT node, low-power wireless, BLE, NFC, SubGHz, Wi-Fi



### Overview

The B-L475E-IOT01A Discovery kit for IoT node allows users to develop applications with direct connection to cloud servers. The Discovery kit enables a wide diversity of applications by exploiting low-power communication, multiway sensing and ARM® Cortex® -M4 core-based STM32L4 Series features. The support for Arduino Uno V3 and PMOD connectivity provides unlimited expansion capabilities with a large choice of specialized add-on boards.

### Microcontroller features

<a href="#">mbed-os-example-blink</a>	Team ST	3512	22 Jun 2017	Fork of <a href="https://developer.mbed.org/teams/mbed-os-examples/code/mbed-os-example-blinky/">https://developer.mbed.org/teams/mbed-os-examples/code/mbed-os-example-blinky/</a>
<a href="#">mbed-os-example-blink</a>	Team mbed-os-e	54008	17 Jan 2019	This is a very simple guide, reviewing the steps required to get Blinky working on an mbed OS platform.
<a href="#">mbed-os-example-blink</a>	Team nuBorn Me	1	29 Jun 2018	Simple LED blinking test code
<a href="#">mbed-os-example-blink</a>	Team Micromouse	2	04 Nov 2016	here is not a real program, just test
<a href="#">blinky1</a>	Team SpringBoar	2	28 Jun 2018	Test program - visibility

To compile a program for this board using Mbed CLI, use DISCO\_L475VG\_IOT01A as the target name.

### Board Partner



ST 클릭

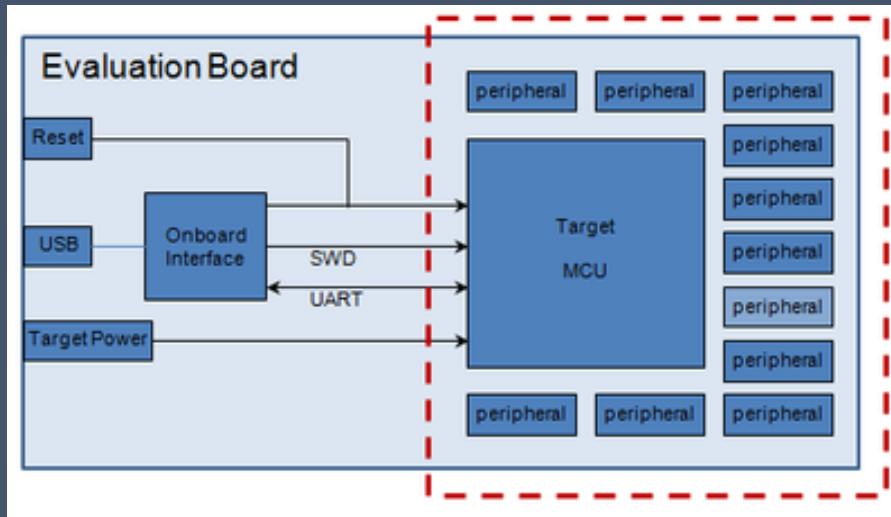
A world leader in providing the semiconductor solutions that make a positive contribution to people's lives, both today and in the future.

Add to your Mbed Compiler

Buy Now

# Debug message

DAPLink의 USB to Serial을 이용



Serial 터미널 프로그램으로  
메시지 확인 (Tera Term)

```
#include "mbed.h"
DigitalOut myled(LED1);
Serial pc(USBTX, USBRX);

int main() {
    pc.baud(115200);
    pc.printf("START Firmware\r\n");

    while(1) {
        myled = 1; // LED is ON
        wait(0.2); // 200 ms
        myled = 0; // LED is OFF
        wait(1.0); // 1 sec
    }
}
```

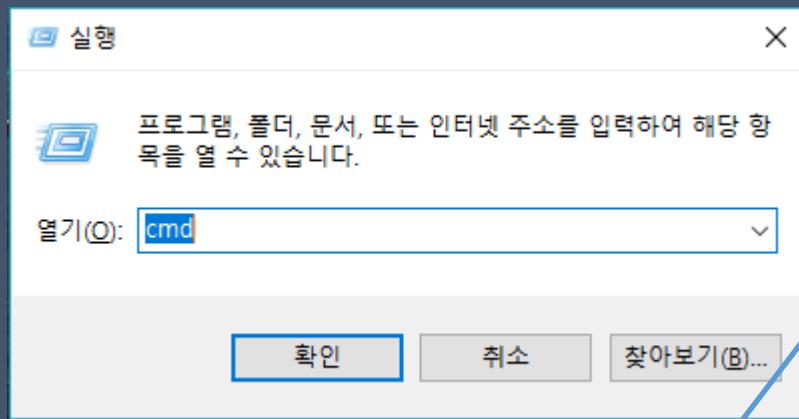
# Mbed CLI

<https://os.mbed.com/docs/mbed-os/v5.11/tools/windows.html>

Manual  
installation

Why??

# Mbed CLI



mbed 입력해서  
아래와 같이 출력되면  
설치 성공

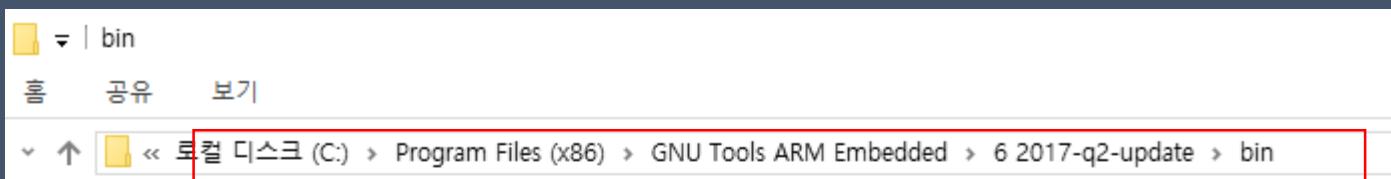
A screenshot of a Windows Command Prompt window titled 'cmd C:\WINDOWS\system32\cmd.exe'. The command entered is 'mbed'. The output is the help documentation for the mbed CLI:

```
C:\Users\jbmast...>mbed
usage: mbed [-h] [--version] ...
Command-line code management tool for ARM mbed OS - http://www.mbed.com
version 1.2.2
Use 'mbed <command> -h|--help' for detailed help.
Online manual and guide available at https://github.com/ARMmbed/mbed-cl...
optional arguments:
-h, --help    show this help message and exit
--version    print version number and exit
Commands:
new          Create new mbed program or library
import       Import program from URL
add          Add library from URL
remove       Remove library
deploy       Find and add missing libraries
publish      Publish program or library
update       Update to branch, tag, revision or latest
sync         Synchronize library references
ls           View dependency tree
status       Show version control status
compile     Compile code using the mbed build tools
test         Find, build and run tests
export       Generate an IDE project
detect      Detect connected mbed targets/boards
config      Tool configuration
target      Set or get default target
toolchain   Set or get default toolchain
help        This help screen
C:\Users\jbmast...>
```

# Mbed CLI

< ARM GCC path check >

C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\bin



로 가기	이름	수정한 날짜	유형	크기
내장 화면	arm-none-eabi-addr2line.exe	2017-06-21 오후...	응용 프로그램	728KB
내운로드	arm-none-eabi-ar.exe	2017-06-21 오후...	응용 프로그램	751KB
온서	arm-none-eabi-as.exe	2017-06-21 오후...	응용 프로그램	1,208KB
내진	arm-none-eabi-c++.exe	2017-06-21 오후...	응용 프로그램	1,718KB
Cloud 사진	arm-none-eabi-c++filt.exe	2017-06-21 오후...	응용 프로그램	726KB
Cloud Drive	arm-none-eabi-cpp.exe	2017-06-21 오후...	응용 프로그램	1,716KB
비트러스트	arm-none-eabi-elfedit.exe	2017-06-21 오후...	응용 프로그램	35KB
BA강의교재	arm-none-eabi-g++.exe	2017-06-21 오후...	응용 프로그램	1,718KB
내의	arm-none-eabi-gcc.exe	2017-06-21 오후...	응용 프로그램	1,715KB
내카오톡 받은 파일	arm-none-eabi-gcc-6.3.1.exe	2017-06-21 오후...	응용 프로그램	1,715KB
eDrive	arm-none-eabi-gcc-ar.exe	2017-06-21 오후...	응용 프로그램	53KB
PC	arm-none-eabi-gcc-nm.exe	2017-06-21 오후...	응용 프로그램	53KB
D 개체	arm-none-eabi-gcc-ranlib.exe	2017-06-21 오후...	응용 프로그램	53KB
	arm-none-eabi-gcov.exe	2017-06-21 오후...	응용 프로그램	1,350KB
	arm-none-eabi-gcov-dump.exe	2017-06-21 오후...	응용 프로그램	1,298KB
	arm-none-eabi-gcov-tool.exe	2017-06-21 오후...	응용 프로그램	1,350KB

# Mbed CLI

< Mbed CLI Configuration >

1. install git

<https://git-scm.com/download/win>

2. GCC Path

₩> mbed config -G GCC\_ARM\_PATH "C:\Program Files (x86)\GNU Tools ARM Embedded\2017-q2-update\bin"

3. Toolchain Path

₩> mbed config -G TOOLCHAIN "GCC\_ARM"

4. cmd 프롬프트를 재실행 하고, 프로젝트를 가져 옵니다.

₩> mbed import http://github.com/ARMmbed/mbed-os-example-blinky

5. ₩> cd mbed-os-example-blinky

₩> mbed config TARGET "DISCO\_L475VG\_IOT01A"

₩> mbed compile

The screenshot shows a Windows Command Prompt window titled '관리자: C:\Windows\system32\cmd.exe'. The command 'mbed config --list' has been run, displaying configuration details. It includes global settings like 'GCC\_ARM\_PATH' pointing to the GNU Tools ARM Embedded directory, and a local configuration for the 'DISCO\_L475VG\_IOT01A' target.

```
D:\NB-IoT\work\mbed-os-example-blinky>mbed config --list
[mbed] Global config:
GCC_ARM_PATH=C:\Program Files (x86)\GNU Tools ARM Embedded\2017-q2-update\bin
TOOLCHAIN=GCC_ARM
CLOUD_SDK_API_KEY=ak_1MDE2NjVjZWE5Zjc0WVIMjZmOWY2YjIwMDAwMDAwMDA01666d482f639ee
26f9f6b2000000000qCN6J0oyZXSqYApMZ5UhsfaaVgZFwqaW
[mbed] Local config (D:\NB-IoT\work\mbed-os-example-blinky):
TARGET=DISCO_L475VG_IOT01A
```

# Mbed CLI

## < Mbed CLI Configuration >

6. mbed-os-example-blinky₩BUILD₩NUCLEO\_F429ZI₩GCC\_ARM 아래에 있는,  
mbed-os-example-blinky.bin 파일을 NODE\_429ZI 디렉토리로 복사

7. ₩>mbed compile -S 해볼것!!

- 사용할 수 있는 플랫폼 약 150종
- ₩>mbed config TARGET "WIZWIKI\_W7500"
- ₩>mbed compile

동일한 소스 코드로 다른 Vendor의 MCU용 바이너리를 만들 수 있다.  
mbed-os-example-blinky₩BUILD₩WIZWIKI\_W7500₩GCC\_ARM

# mbed-cli 사용하기

새로운 프로젝트 생성

```
mbed new new-project
```

기존 프로젝트 가져오기

```
mbed import http://github.com/ARMmbed/mbed-os-example-blinky
```

프로젝트에 라이브러리 추가

```
mbed add https://os.mbed.com/teams/ST/code/X_NUCLEO_IKS01A2/
```

다른 iDE용으로 프로젝트 출력

```
mbed export -i uvision -m UBLOX_EVK_ODIN_W2
```

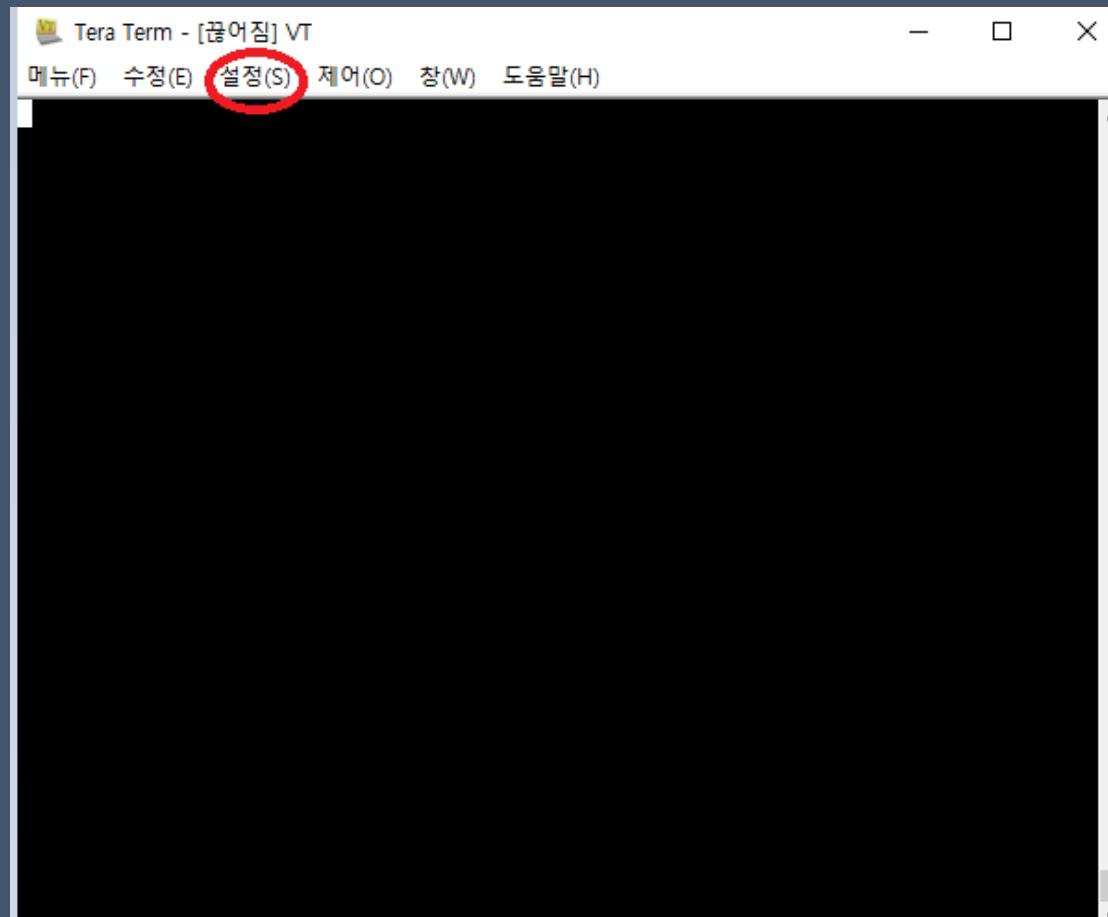
## 터미널 프로그램 설치

PC에서 mbed보드 데이터를 확인하기 위해 터미널 프로그램을 설치해야 합니다. 여러가지 프로그램이 있는데, 그중에서 무료로 사용할 수 있는 테라텀(Tera Term)을 설치하도록 하겠습니다. 테라텀은 아래 경로에서 다운로드 하실수 있습니다.

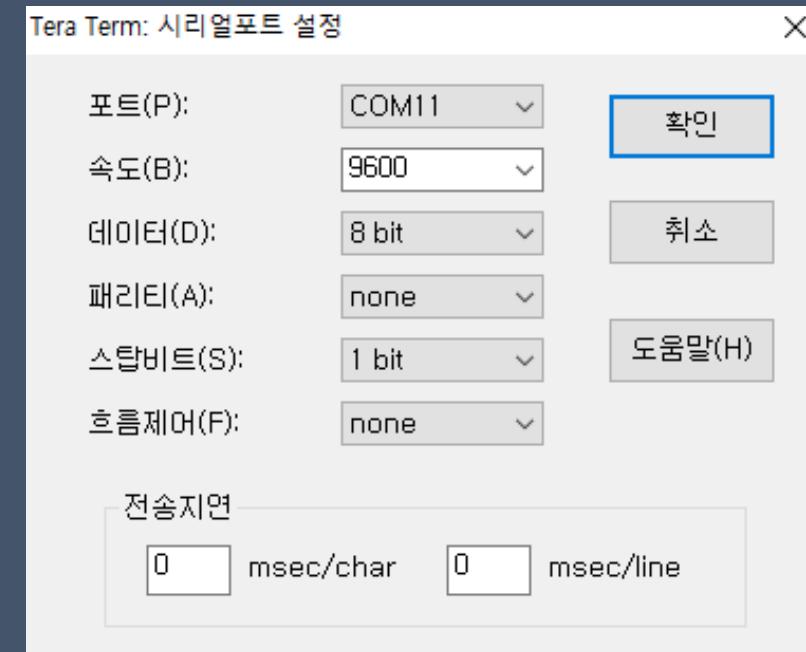
<https://osdn.net/projects/ttssh2/releases/>



## 터미널 프로그램 설정



설정 → 시리얼포트 를 선택하고 아래와 같이 설정하고 확인을 클릭합니다.  
Mbed는 별도 설정이 없을 경우 9600bps로 설정 됨  
(COM포트값, 9600bps/115200bps 속도 주의)



# mbed 데이터형

## 정수형 데이터 타입

C type	stdint.h type	Bits	Sign	변수값의 범위
char	uint8_t	8	Unsigned	0 ~ 255
signed char	int8_t	8	Signed	-128 ~ 127
unsigned short	uint16_t	16	Unsigned	0 ~ 65,535
short	int16_t	16	Signed	-32,768 ~ 32,767
unsigned int	uint32_t	32	Unsigned	0 ~ 4,294,967,295
int	int32_t	32	Signed	-2,147,483,648 ~ 2,147,483,647
unsigned long long	uint64_t	64	Unsigned	0 ~ 18,446,744,073,709,551,615
long long	int64_t	64	Signed	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807

## 논리형 데이터 타입

C type	Bits	변수값의 범위
bool	8	true(=1) 또는 false(=0)

## 실수형 데이터 타입

C type	IEEE754 Name	Bits	Range
float	Single Precision	32	$-3.4 \times 10^{-38}$ ~ $3.4 \times 10^{38}$
double	Double Precision	64	$-1.7 \times 10^{-308}$ ~ $1.7 \times 10^{308}$

# mbed.h 분석

```
#ifndef MBED_H
#define MBED_H

#define MBED_LIBRARY_VERSION 149
// Useful C libraries
// C/C++언어로 프로그램 작성시 주로 사용하는 헤더파일이 선언되어 있습니다.
#include <math.h>
#include <time.h>

// mbed Debug libraries -> mbed 디버깅시 사용하는 헤더파일이 선언되어 있습니다.
#include "platform/mbed_error.h"
#include "platform/mbed_interface.h"
#include "platform/mbed_assert.h"
#include "platform/mbed_debug.h"

// mbed Peripheral components
// mbed 보드에 다양한 하드웨어를 연결하기 위해 필요한 헤더파일이 선언되어 있습니다.
#include "drivers/DigitalIn.h"
#include "drivers/DigitalOut.h"
#include "drivers/DigitalInOut.h"
#include "drivers/BusIn.h"
#include "drivers/BusOut.h"
#include "drivers/BusInOut.h"
#include "drivers/PortIn.h"
#include "drivers/PortInOut.h"

#include "drivers/PortOut.h"
#include "drivers/AnalogIn.h"
#include "drivers/AnalogOut.h"
#include "drivers/PwmOut.h"
#include "drivers/Serial.h"
#include "drivers/SPI.h"
#include "drivers/SPISlave.h"
#include "drivers/I2C.h"
#include "drivers/I2CSlave.h"
#include "drivers/CAN.h" ...

// mbed Internal components
// mbed 타이머, 파일시스템, 인터럽트 등을 사용하기 위한 헤더파일
#include "drivers/Timer.h"
#include "drivers/Ticker.h"
#include "drivers/Timeout.h"
#include "drivers/LocalFileSystem.h"
#include "drivers/InterruptIn.h" ...

using namespace mbed;
using namespace std;

#endif
```

감사합니다.