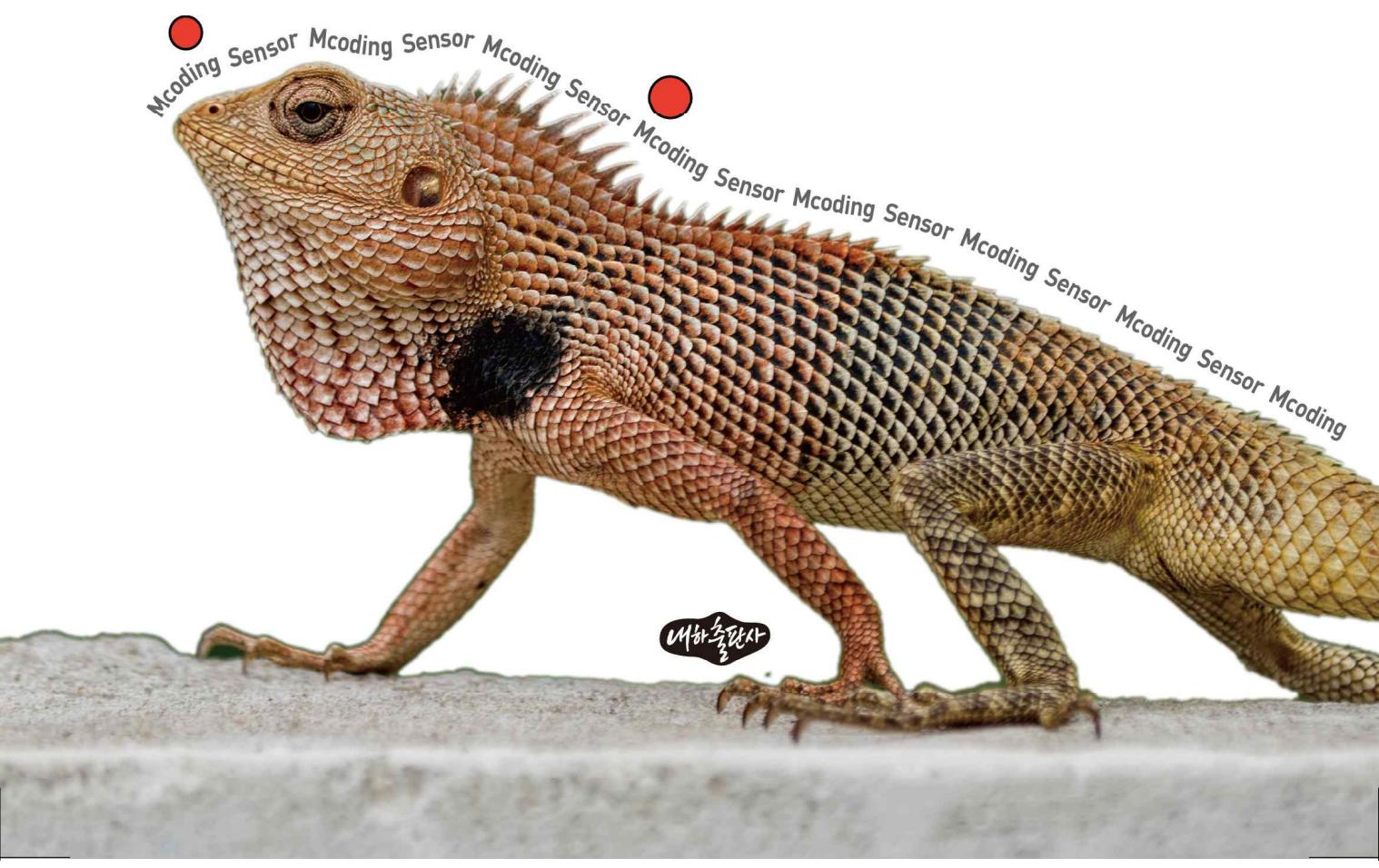


# 고액 연봉 도전 프로젝트 ! 코딩으로 배우는 센서

김형태



## 센서를 코딩으로 배우는 기술 실무서

코딩의 관심과 메이커 인구의 확산으로 프로그램을 익히고 활용하는 기회가 무척 많아졌다. 청소년들은 학교나 동아리 등의 단체를 통해 오픈하드웨어 아두이노를 이용하여 입출력 장치를 제어하는 일반 또는 정규 학습이 있고 인공지능과 IoT, 빅데이터를 활용하는 사례가 늘면서 경진대회, 발명품 개발, 창업, 졸업작품, 예술품 등에서 센서를 다루는 기회가 생겼다.

임베디드 기술자인 저자도 센서를 프로젝트에서 상당히 많이 다루고 있는데 센서를 부품이나 모듈 상태로 구입하여 데이터시트와 웹을 통해 기초 학습을 하고 초기화한다. 하지만 손쉽게 구할 수 있는 오픈 소스 코드는 대부분 아두이노에서 사용하고 라이브러리를 이용하는 방법이어서 Embedded로 옮기기에는 시간이 많이 소요되고 나의 프로젝트에 맞게 수정하는데 큰 어려움이 수반된다. 이런 불편함을 해소하고 기술 이해의 호기심과 그동안 32비트 ARM Cortex-M 시리즈 활용방법을 집필한 기술자의 한사람으로 센서를 다루어 보자는 취지로 시작한 원고가 계획했던 것보다 기간이 길어졌다.

첫 번째로 센서의 종류가 너무 다양했다. 프로젝트에 필요한 여러 방식과 모양의 센서에서 가성비 좋은 것으로 선택하고 학습한 뒤 마이크로프로세서와 연결하고 초기화 그리고 실험하면서 요약정리하고 응용 코드를 만들고 디버깅하는 일련의 시간이 무척 많이 소요되었다. 둘째로는 센서를 응용할 곳이 많다. 자동차는 센서 집합체 가전이라는 분위기와 센싱 데이터를 클라우드에 저장하는 빅데이터, 미세먼지, 라돈, 수질 오염 등의 환경에 대한 지대한 관심으로 센서가 필요한 곳이 늘어나면서 구현하고자 하는 요구가 다양해지고 응용범위가 넓어져서 적용 예제를 여러 가지로 만들어야 했다. 셋째로는 청소년 대상과 아두이노 등의 기초 교육 분야와 저와 같은 기술자들에게 도움을 주고자 했다. 그동안 특정 직업을 가진 능력자와 ICT 분야에서만 사용하던 기술이 요즈음에는 오픈 소스를 이용하여 초등학생도 아이디어 경진대회에 센서로 근사하게 동작하는 물건을 가지고 나온다. 그런데 외국 사람들이 만든 소스를 무작정 복사하여 이용하는 것보다는 가능하다면 기술자가 집필한 우리나라 자료를 가지고 학생들이 유용하게 사용했으면 하는 마음으로 준비 하다보니 코드를 만들고 실험하는 데 좀 길어졌다.



아무리 예제를 통해 사용이 편리하도록 집필했다고 해도 센서는 응용하기가 어렵다. 기술은 순끌에서 실습을 통해 이루어져야 머리에 각인된다고 생각한다. 눈으로 보고 머리로만 센서의 코드를 익히지 말고 아두이노 또는 Cortex-M 시리즈 전자보드를 준비하여 프로그램을 업로드 한 다음 센서 입력 결과로 익히는 것을 권장한다. 차근차근 정확하면서 빠르게 익혀 IT를 잘한다는 대한민국의 센서 응용 기술자들이 되어 우리나라를 빛내 주기를 바란다.

#### 이 책을 읽고 실습을 통해 얻을 수 있는 것들

- 센서의 구동원리와 C 코드 응용방법
- 대만 Nuvoton사의 32비트 ARM 코어 Cortex-M0을 이용한 센서 소스
- 아두이노(나노, 우노) 보드에서 동작되는 센서 소스
- GPIO, ADC, UART, I2C, SPI, Serial 통신으로 인터페이스 센서 통신방법
- 센서값을 실시간 디스플레이하는 몇 가지 방법의 소스

기술서는 출판사에 원고를 제출하여 여러 번 교정을 보고 책으로 인쇄되어 나온다. 원고의 저작권은 저자에게 있지만 서적으로 출간된 이상 책은 출판사의 소유물이 된다. 출판사 측에서 일정기간의 비즈니스가 잘 될 수 있도록 저자와 협의하여 소스코드에 암호를 걸어 두었다. 웹을 통해 소스를 다운 받아도 책에 적혀 있는 암호코드 4자리가 없다면 볼 수가 없도록 해 놓아 자연스럽게 책을 구매하여 출판사와 상생하는 학습을 유도했다. 인터넷을 통해 배포되는 원고의 내용과 기술지원 그리고 샘플 예제 모든 소스는 저작권법에 의해 보호받게 되어 있다. 무단으로 도용하거나 웹에 무책임하게 암호를 올려 출판사가 법적인 책임을 묻는 일이 발생하지 않도록 주의해주기를 바란다.

#### 이 책의 최신 기술사항 주소

- 카페 : [cafe.naver.com/fws](http://cafe.naver.com/fws)
- 아두이노 : [www.arduino.cc](http://www.arduino.cc)
- ARM Cortex-M0 Nuvoton사 : [www.nuvoton.com](http://www.nuvoton.com)
- ARM Compiler : [www.keil.com](http://www.keil.com)

## 코딩과 함께 하는 센서를 효율적으로 익히는 방법

**01** 본 센서를 설명한 기술책은 2가지 종류의 마이크로프로세서가 탑재된 전자보드를 사용하여 인터페이스 구현되었다. 하드웨어가 다르기 때문에 동작은 동일하게 되어도 소스코드는 다르다. 깊은 실습을 원한다면 업로드 할 보드를 최소 한 가지 준비해서 동작 구현해 보기 바란다.

- ARM사의 32Bit RISC 마이크로프로세서 Cortex™-M0 MINI54FDE
- 아두이노 나노 보드 또는 아두이노 우노 보드

**02** 센서 인터페이스의 소스는 C 언어로 제작 실현되어 C 언어에 대해 기본적으로 알고 있어야 한다. 마이크로프로세서 응용으로 저자가 집필한 책이 있지만 8비트를 기준으로 한 것이다. 유사한 관련 서적을 처음부터 끝까지 가볍게 독파하고 이 책을 참고하면 이해가 더욱 빠르다.

- C 언어 참고 서적 : "펌웨어를 위한 ANSI C", 김형태 저자, 내하출판사

**03** RISC 같은 고성능 Microprocessor는 고급 언어를 기계어로 변경할 수 있는 컴파일러가 있어야 하는데 책의 소스코드는 아두이노 스케치와 ARM사의 MDK-ARM Compiler를 이용하였다.

- 아두이노 스케치 다운로드 : [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software)
- ARM 컴파일러 무료 다운로드 : [www.keil.com/download](http://www.keil.com/download)

**04** 센서를 C 소스로 구현하는 것이 처음인 독자라면 개발 환경이 어색할 수밖에 없다. 센서 코드로 들어가기 전에 Quick Start를 통해 빠르게 전체 방법을 익혀야 한다. 특히 코드를 에러 없이 컴파일하여 타겟보드에 이식하고 C 언어 상태로 실행하면서 오류를 찾는 노하우를 익혀야 한다.



**05** 메인보드만 있고 센서가 없다면 대부분 소스는 정상으로 동작되지 않는다. 필요하다면 사이트를 통해 준비해 주기 바란다. 센서는 대부분 고가이다. 전기적으로 연결할 때 한 번의 실수도 전자 부품에는 치명적일 수 있다. 대충하지 말고 신중히 연결해야 동작을 한다.

**06** 센서는 스스로 동작하지 않는다. 사용자가 센서를 초기화하고 센싱 출력 범위의 값을 받아서 처리해야 한다. 예로 온도센서는 저항이 변하는데 이를 현재 온도로 변환해 주어야 하는 방정식이 필수이다. 저자가 집필할 때 만든 소스 코드는 일괄 압축하여 네이버 카페에 올렸지만 자신의 프로젝트에 맞게 코드를 각색하기를 바란다.

• cafe.naver.com/fws : 오른쪽 "단행본/코딩으로 배우는 센서"

**07** 알나간(알고 나면 간단)의 용어정리가 있다. 기술 서적에는 외국어로 된 약어 정리와 이론에 대한 설명 그리고 센서의 원리가 있으므로 관련 기초 지식도 같이 학습해야 이해가 빠르고 잘 잊어버리지 않는다.

**08** 알나지(알고 나면 지식)로 궁금증과 과학 트렌드, 기초이론에 대해 매스컴을 통해 얻은 지식을 저자의 생각으로 재편집하거나 그대로 옮겨 놓은 정보이다. 칼럼을 통해 독자들은 개발하는데 영감을 얻기 바란다.

## CONTENTS

### ● Quick Start

- ▶ 센서와 코딩 시작 \_003
- ▶ ARM Cortex-M0와 컴파일러 시작하기 \_008
- ▶ 아두이노 스케치 설치와 프로젝트 시작하기 \_021

### ● Experience

- ▶ 가속도센서(GY-61/ADXL335) \_029
- ▶ 가속도센서(GY-45/MMA845X) \_036
- ▶ 가스센서(MH-Z19/CO2) \_050
- ▶ 가스센서(FC-22/MQ135/CO2) \_065
- ▶ 가스센서(MQ4/CH4/메탄) \_073
- ▶ 가스센서(MQ6/LPG) \_079
- ▶ 디스플레이(7-세그먼트/TM1637) \_084
- ▶ 디스플레이(Text LCD 8Bit 방식) \_096
- ▶ 디스플레이(Text LCD 4Bit 방식) \_107
- ▶ 디스플레이(Text LCD 소프트웨어 I2C 방식) \_116

Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor  
Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor

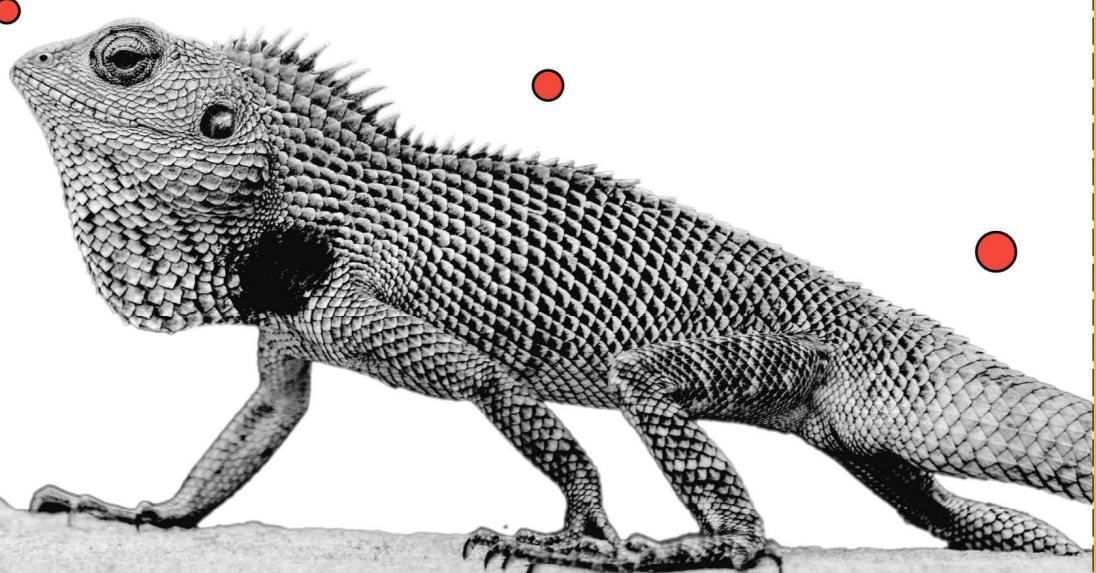
▶ 디스플레이(Text LCD 인터럽트 I2C 방식)	_131
▶ 리얼타임클럭RTC(DS3231)	_141
▶ 먼지센서(GP2Y1010AU0F)	_149
▶ 빛센서(PT550)	_158
▶ 빛센서(YG1006)	_164
▶ 빛센서(D3VS)	_169
▶ 빛센서(CDs)	_180
▶ 사운드감지(Sound Detect)센서	_190
▶ 사운드트리거(DFR0034)	_197
▶ 스위치(버튼A접점B접점)	_211
▶ 압전(Piezoelectric)센서(EMFIT L-Series)	_222
▶ 온도센서(PT100)	_228
▶ 온도센서(NTC)	_237
▶ 온도센서(LM35)	_249
▶ 온도센서(DS18B20)	_254
▶ 온도습도센서(DHT11)	_263
▶ 위치감지/포지션센서(PR101)	_277
▶ 엔코더(EVQ-VVE00203B 스위치타입)	_280
▶ 엔코더(IE216 모터내장타입)	_289
▶ 엔코더(RA-12WGM 02타입)	_291
▶ 인체저항감지센서(NE555)	_300
▶ 인체감지PIR센서(NT0081)	_310
▶ 인체감지PIR센서(SEN0171)	_315
▶ 인체감지PIR센서(HC-SR501)	_321
▶ 적외선수광센서(ST5811)	_332

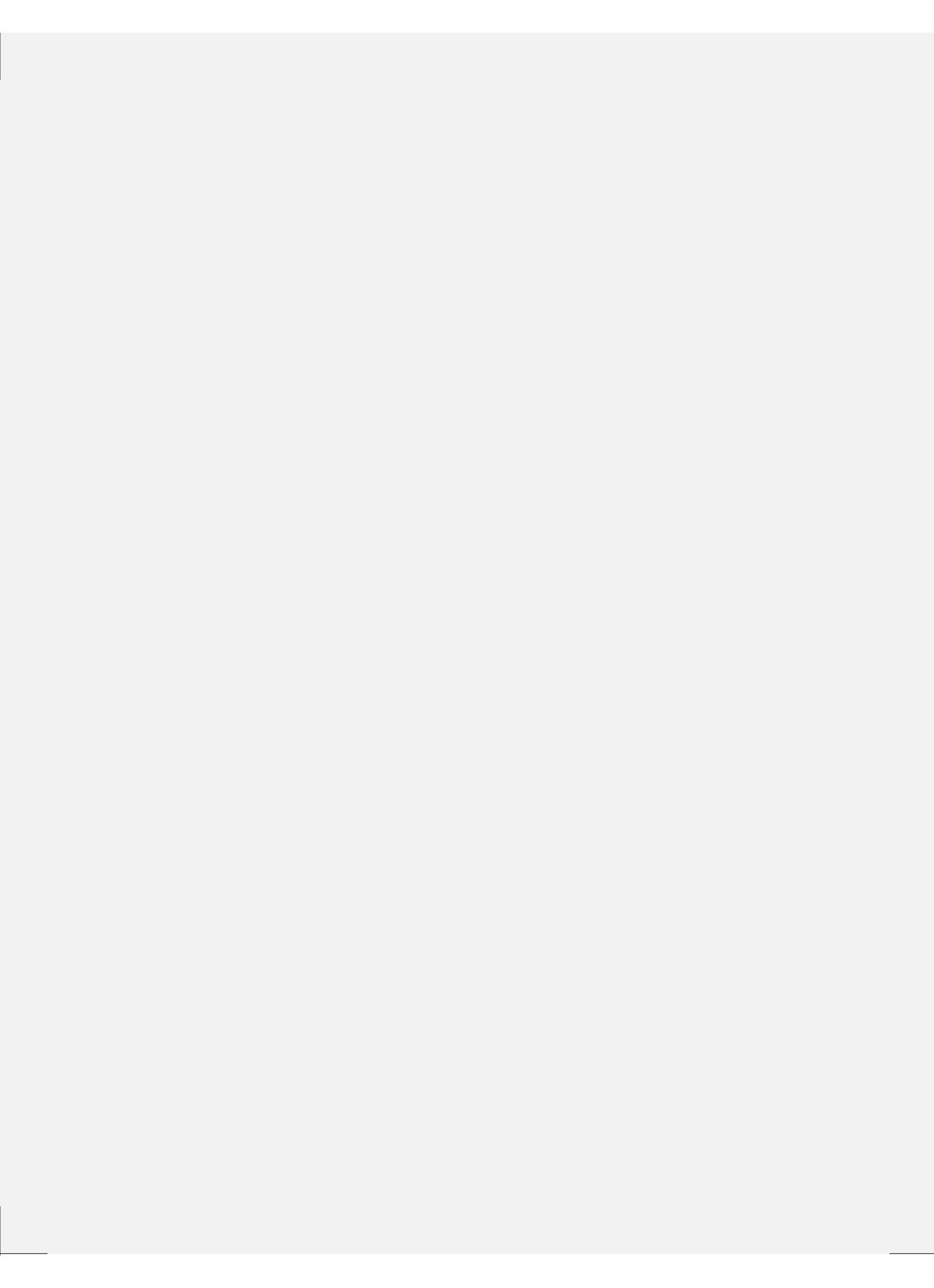
▶ 적외선거리측정센서(GP2Y0A21YK0F)	_337
▶ 전류센서(TLS77V)	_341
▶ 조이스틱(Joystick)	_347
▶ 진동센서(Vibration/Tilt/Ball Switch)	_351
▶ 초음파센서(HC-SR04/NT-TS601)	_355
▶ 초음파센서(URM37)	_373
▶ 카드리더(RFID-RC522)	_383
▶ 칼라센서(TCS3200)	_390
▶ 토양수분(Soil Moisture)검출센서	_400
▶ 터치센서(NT084DA-8 버튼)	_409
▶ 터치센서(TS08N-8 버튼)	_426
▶ 포토센서(Photo Interrupter)	_431
▶ 훌센서(SS49E)	_437
▶ 힘센서	_452

찾아보기	_455
------	------

Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor Mcoding Sensor

## Quick Start







센서를 이용하여 처음 코드를 설계할 때는 쉽게 생각했다. 비싼 부품을 사용하니까 마이크로프로세서와 함께 센서의 결과값으로 처리하면 될 것이라고 배운 대로 생각했는데 하면 할수록 점점 더 어려워진다. 센서는 대부분 입력의 크기에 비례 또는 반비례해서 출력 전압으로 나오게 되는데 출력값이 모듈마다 다르고 심지어는 기준 전압이 달라지면서 출력 결과값도 공급 전압에 영향을 받는다는 것이다. 여기에 코드의 테크닉에 따라 효율이 좋을 수도 있으며 정밀성도 달라진다.

본 기술서는 IoT와 빅데이터와 함께 증가되는 센서 인터페이스와 코딩 교육의 일환으로 우리의 궁금증과 사회적 이슈에 부응하고자 집필하게 되었으므로 센서가 전자보드에서 바로 이용되도록 C 언어 기반의 소스코드로 작성되었다. C 언어는 MCU(마이크로 컨트롤 유닛)에 바로 이식할 수가 없기 때문에 기계어 코드로 변경해 주는 소프트웨어가 있어야 한다. 또 변경된 소프트웨어는 마이크로컴퓨터가 납땜되어 있는 아두이노 우노/나노(8비트), ARM(32비트) 같은 기판의 MCU (Micro Controller Unit) 플래시 메모리에 다운로드하여 실행해 볼 수 있다.

이 기술서는 센서의 이론과 스펙, 응용, 회로 연결방법 그리고 코드에 대한 설명을 해주며 독자들은 책 각각의 챕터에 있는 암호를 이용하여 네이버의 카페에서 다운 받은 소스코드에 암축 암호를 넣고 풀어서 진행하면 된다. 암축 암호는 저자가 소스코드를 실험한 그때의 시간에서 가지고 온 4자리 숫자이며 출판업계의 상생과 무단 복제 방지를 위한 자구책이면서 출판사의 이익을 보호하기 위한 수단임을 밝힌다.

소스코드는 저자가 센서를 전자보드와 연결하고 코드를 타겟보드에 이식한 뒤 동작 실험을 거친 결과물에 해당한다. 동작 확인과 이해를 돋기 위해 4채널, 로직애널라이저(Logic Analyzer)가 연결된 오실로스코프의 화면 캡처와 표준 계측기를 이용하여 가능한 비교를 했다.

코드의 검증과 사용된 보드는 32비트 ARM Cortex-M0, Cortex-M4, 그리고 아두이노 UNO/NANO이며 이때 사용되는 소프트웨어와 컴파일러는 다음과 같다. 실험과 프로젝트를 위해 책을

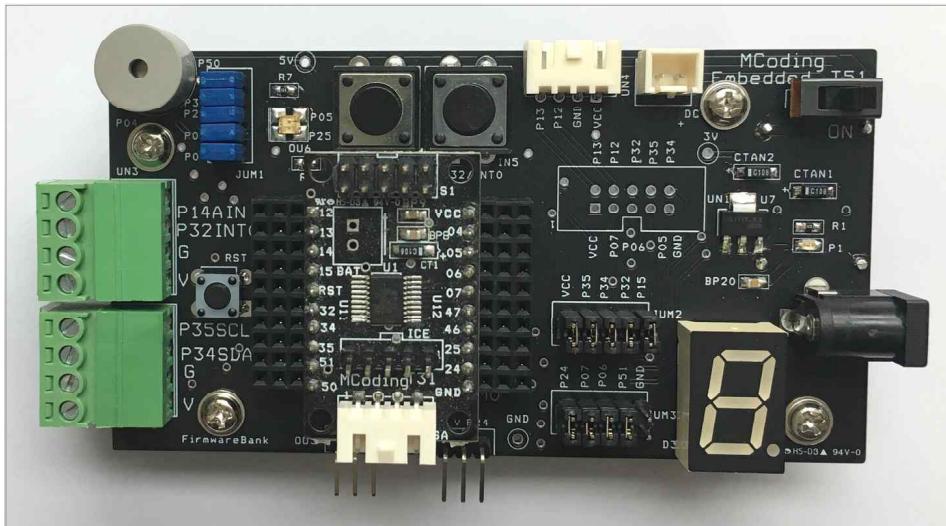
보고 구입을 원하는 독자들은 이중에서 한 가지를 선택하여 진행하면 되겠다. 저자가 만든 코드 중에는 상대적으로 아두이노 관련한 센서 코드는 많지 않다. 이유는 아두이노 사용자가 많아 양질의 초기화 코드를 웹을 통해 구할 수 있다고 생각되고 저자는 아두이노에서 코드의 잘못을 확인하는 디버깅 환경이 불편했다. 그래서 상대적으로 C 언어 기반의 코드생성을 주로 하고 손에 익숙한 장비여서 ARM Cortex-M0 기반의 예제가 많다. 따라서 ARM 기반의 Embedded 기술자들에게는 코드를 MDK-ARM 컴파일러에서 프로젝트 단위로 불러와서 바로 적용이 가능하다.

사용된 보드	컴파일러	디버그 장치	비고/MCU
MCoding	MDK-ARM	Nu-Llink-Me, JTAG	Nuvoton, Cortex-M0, Mini54FDE
M4F		ULINK2 JTAG	TI, Cortex-M4F, TM4C1230C3PM
Arduino Uno	ARDUINO 1.8.5	소프트웨어	ATMEL, AVR
Arduino NANO			

실험을 위해서는 아두이노, ARM Cortex-M0 상관없다. 소스코드를 타겟보드에 업로드하여 센서를 익힐 수 있는 기회이다. 그러나 센서는 구경한 경험이 있는데 마이크로프로세서 타겟보드가 처음인 독자들이라면 코드를 불러오고 어떻게 빠르게 적용하는지 알려주는 Quick Start를 해 보면 된다. 설명된 순서는 처음 챕터에는 ARM 개발환경을 다음 챕터는 아두이노에 관한 개발 환경이다. 또 센서와 코드 실험 중에 동작이 실시간으로 필요할 경우 동영상을 촬영하여 유튜브에 올려놓았으므로 보드가 없을 경우라도 확인이 가능하다. 여러 가지 이유로 저자와 소통이 필요하거나 독자들의 더 좋은 방법, 나아가서는 저자의 수정이 필요한 공지 등은 네이버 카페(펌웨어스터디)를 통해 정보를 얻길 바란다.

#### ARM Cortex-M0 Set :

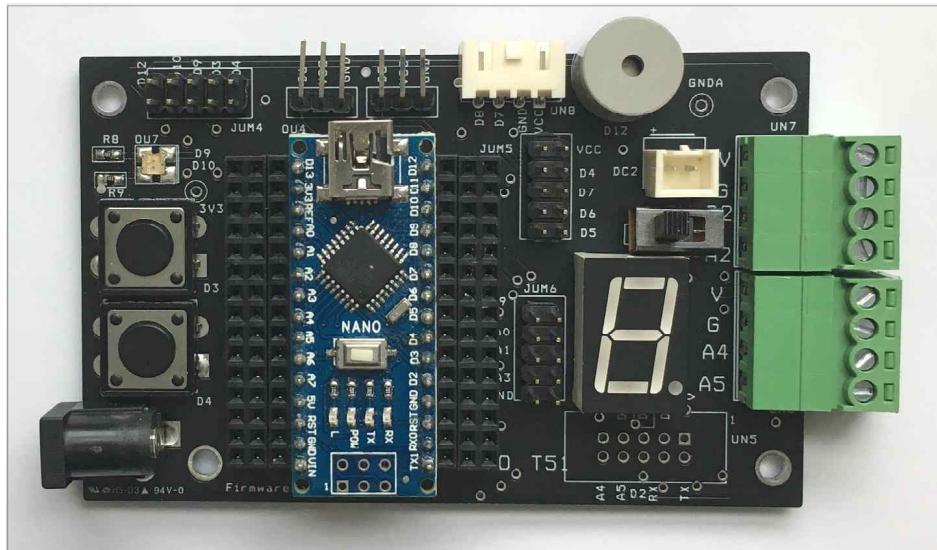
- MCoding Embedded 보드 + ARM Cortex-M0 모듈 + ARM Nuvoton 전용 JTAG
- **Gmarket** : <http://gmkt.kr/gBaNiBo> : 검색어 "펌웨어뱅크 ARM Cortex-M0 Set"



▶ MCoding Embedded 보드에 ARM Cortex-M0 탑재된 Target

### 아두이노 NANO Set :

- MCoding Arduino 보드 + 아두이노 NANO 모듈 + 다운로드 USB Cable
- **Gmarket** : <http://gmkt.kr/gBaNjYj> : 검색어 "펌웨어뱅크 아두이노 NANO Set"



▶ MCoding Arduino 보드에 NANO 탑재된 Target

### "ARM Cortex-M0 Nuvoton 전용 JTAG" :

- 디바이스마트 : <http://www.devicemart.co.kr/31015>
- **Gmarket** : <http://gmkt.kr/gBXbRHs>



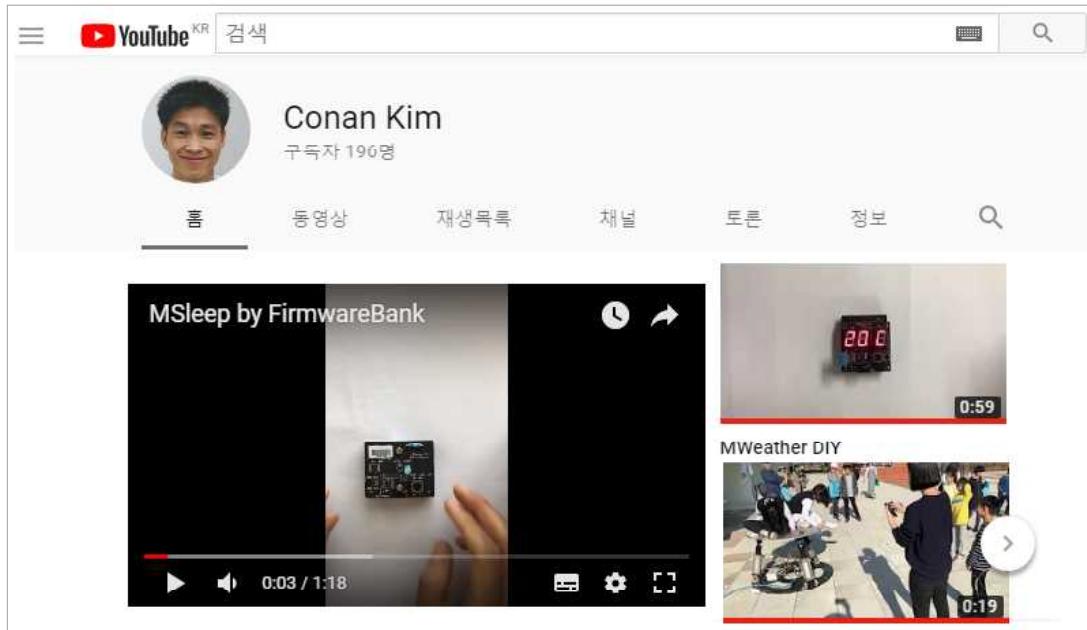
▶ ARM Nuvoton 전용 JTAG

- 네이버 FWS 카페 방문 : "단행본/코딩으로 배우는 센서" 항목
- **NAVER** : <https://cafe.naver.com/fws>



▶ 네이버 카페 FWS(FirmWare Study) 대문

◆ 유튜브 : <https://www.youtube.com/conan3208>



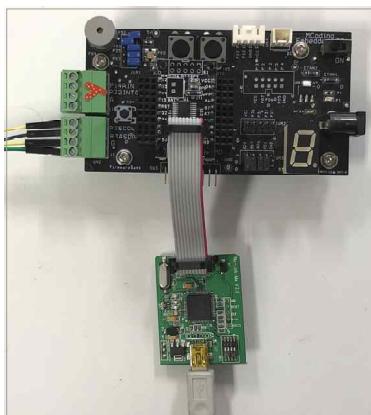
▶ 저자의 동영상 참고 유튜브 Conan3208

## ARM Cortex-M0와 컴파일러 시작하기

우리가 센서에서 값을 얻기 위해서는 마이크로프로세서를 통해 센서에 특별한 노력을 할 수 있는 코드를 실행할 하드웨어가 있어야 한다. 하드웨어는 PCB(Printed Circuit Board) 기판에 MCU를 비롯한 전자 부품이 납땜되어 있는 타겟보드(Target Board)라고 하는 전자장치이며 센서와 인터페이스가 가능하도록 여러 가지 연결 콘넥터가 부가적으로 구성되어 있다. ARM사의 32비트 Cortex-M0 시리즈는 고성능의 빠르고 함수가 잘 정리되어 Embedded 프로젝트에 적합한 마이크로컴퓨터라고 할 수 있다. 칩의 내부에 디버깅 기능을 설계해 놓아 핀 제어와 동작 모니터링이 가능한 장점 때문에 좋은 환경의 컴파일러와 장비를 통해 빠르고 정확히 C 언어상에서 오류를 찾을 수 있다. 이런 장치를 JTAG 장비라고 한다. 센서를 효율적으로 익히기 위해서는 C 언어를 해당 MCU의 기계어 코드로 변환해 주는 통합개발환경(IDE) 컴파일러와 다운로드 장비 그리고 마이크로프로세서 타겟보드가 있어야 한다.

### ARM Cortex-M0 Set :

- MCoding Embedded 보드 + ARM Cortex-M0 모듈 + ARM Nuvoton 전용 JTAG
- **Gmarket** : <http://gmkt.kr/gBaNiBo> : 검색어 "펌웨어뱅크 ARM Cortex-M0 Set"



▶ 센서 실습을 위한 MCoding Embedded 보드와 JTAG



## JTAG(Joint Test Access Group)

70년대 중반에는 "bed-of-nails"라는 테크닉을 사용하여 직접 PCB(Print Circuit Board)에 접촉하는 방식으로 Board를 테스트하였으나 기술이 발달함에 따라 Board 단자 사이의 거리가 좁아지고 다층기판이 나타남에 따라 기존의 방법으로는 검사가 불가능하게 되었다. 문제점을 해결하기 위해 80년대 중반에 Joint European Test Access Group이란 단체가 결성되었으며 그들은 a serial shift register around the boundary of the device라는 개념을 발전시켰다. 후에 이 단체는 North American company와 join하게 되었으며 이에 따라 E(European) 문자가 없어지고 JTAG으로 남게 되었고 결국에는 IEEE에서 1990년에 표준화하여 IEEE std 1149.1로 제정되었다.

JTAG란 말보다는 Boundary-Scan이란 말을 더 많이 사용하는데 JTAG은 칩 내부에 Boundary Cell이란 것을 두어 외부의 핀과 일대 일로 연결시켜 프로세서가 할 수 있는 동작을 중간에 Cell을 통해 모든 동작을 인위적으로 수행할 수 있어 여러 가지 하드웨어 테스트나 연결 상태 등을 체크할 수 있다.

JTAG은 프로세서(CPU)의 상태와는 상관없이 디바이스의 모든 외부 핀을 구동시키거나 값을 읽어 들일 수 있는 기능을 제공하므로 이를 이용해서 메모리나 I/O에 직접 명령을 보낼 수 있어 디버그용으로도 사용된다. JTAG 기능을 자세히 보면 다음과 같다.

- ① 디바이스 내에서 외부로 나가는 각각의 핀들과 일대 일로 연결되어 연결점을 가로챌 수 있다.
- ② 각각의 셀은 시리얼 쉬프트 레지스터(바운더리 스캔 레지스터)를 형성하기 위해서 서로 연결되어 있다.
- ③ 전체적인 인터페이스는 TDI, TMS, TCK, nTRST, TDO 5개의 핀에 의해서 제어된다.
- ④ 회로의 배선과 소자의 전기적 연결 상태 test
- ⑤ 디바이스간의 연결 상태 test
- ⑥ Flash memory fusing

