

* 본 자료는 서경대학교 아두이노 프로그래밍 수업 수강자를 위해 작성된 강의 교재입니다.

강의교재- 아두이노 프로그래밍

1장 개요

서경대학교 김진헌

1

개요

내용

- 1.1 아두이노란?
- 1.2 실습 환경 및 학습 내용
- 1.3 아두이노 보드의 프로세서
- 1.4 어떤 개념을 배울 것인가?
- 1.5 고찰

1.1 아두이노란?

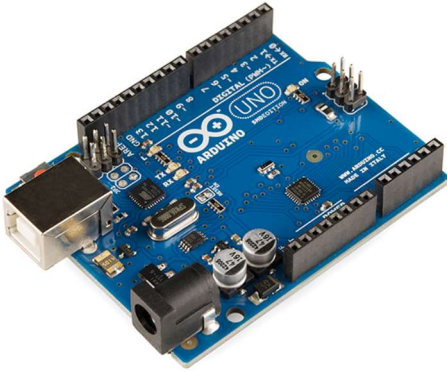


그림 1.1 Arduino Uno

어를 개발하여 공개하고 있다.

구글에서는 2011년 아두이노를 지원하는 프로그램 환경인 Android ADK(Accessory Development Kit)를 공개하였다. ADK는 안드로이드 기기에 USB로 부착된 디바이스가 USB 마스터로 동작하고 안드로이드 기기가 슬레이브 모드로 동작하게 하는 AOA(Android Open Accessory) 프로토콜을 공개하고 관련 개발 라이브러리를 지원한다³⁾.

2011년 샌프란시스코에서 열린 구글 I/O 2011 행사에서 그림 1.2(좌)에 보인 바와 같이 구글은 아두이노 쉴드(ADK board)를 아두이노의 상단부에 장착하여 USB 마스터를 지원하게 하는 가정 자동화(android@home) 데모 프로젝트를 선보였다⁴⁾. 그림 1.2(우)에는 모토롤라의 Xoom AOA 데모킷을 보인 것인데 상단의 아두이노 쉴드 아래에 아두이노 본체가 장착되어 있다. 이

아두이노(Arduino)는 사용하기 쉬운 H/W와 S/W로 구성된 소스가 공개된 시제품 제작용 개발 플랫폼(platform)이다¹⁾. 그림 1.1은 가장 기본적인 아두이노 플랫폼 중의 하나인 아두이노 우노 보드를 보인 것이다. 아두이노가 출현한 이후 여러 S/W, H/W 회사와 다양한 커뮤니티가 다양한 버전의 아두이노 보드, 쉴드(shield)²⁾, 센서, 제어기 및 소프트웨어

1) 아두이노는 최근 미국 외의 지역에서는 Genuino라는 브랜드로 판매하고 있다. 자세한 내용은 아두이노의 공식 사이트(<http://www.arduino.cc>)와 다음 자료를 참조하기 바란다.

<http://makezine.com/2015/05/16/arduino-adafruit-manufacturing-genuino/>

2) 아두이노 보드 위에 추가로 장착하는 보드를 일컫는다.

3) AOA의 연결수단으로는 USB 말고도 블루투스도 지원된다.

4) <https://www.youtube.com/watch?v=s7szcpXf2rE> (17분 경 시작),

<https://www.youtube.com/watch?v=XEe3dYBj7RE> (아두이노 보드 위에 ADK가 장착되어 있음)

데모킷 역시 휴대폰의 App.을 이용해 다양한 가전제품을 제어하는 용도로 아두이노를 활용하고 있다.



그림 1.2 (좌) IO 2011 데모, (우) 모토롤라 Xoom AOA 데모킷

아두이노 솔루션은 가격이 싸고 표준화되어 있으며 프로그래밍하기도 쉬워서 쉽게 자신이 만들고자 하는 작품을 쉽게 달성할 수 있기 때문에 공학자 뿐만 아니라 예술작품, DIY 설계, 취미 활동 등 여러 분야에서 다채롭게 활용되고 있다. 아두이노를 이용한 다채로운 프로젝트의 사례는 <http://makezine.com/>, 유튜브를 비롯한 다양한 사이트에서 쉽게 접해 볼 수 있다. 아두이노 공식 사이트(<http://arduino.cc>)에서는 아두이노 보드와 쉴드⁵⁾, 키트 정보 등을 볼 수 있고 프로그램 방법 및 예제 등을 다운 받을 수 있다.

5) shield : 아두이노 보드 위에 장착하는 여러 종류의 인터페이스 모듈을 말한다. 각종 센서류, 제어 모듈, 와이파이 및 블루투스 통신 모듈 등이 있다.

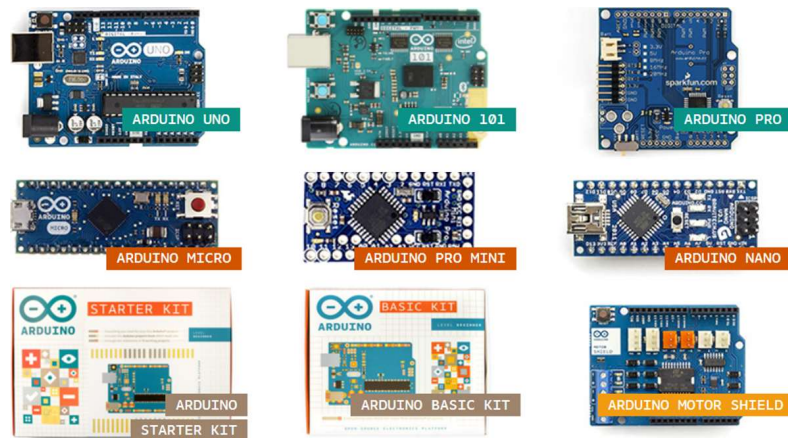


그림 1.3 입문용 아두이노 보드와 시작 키트 및 쉴드

아두이노 메인 보드는 주로 Atmel사의 AVR 프로세서를 기반으로 하며 USB 포트에 연결하여 손쉽게 프로그래밍할 수 있는 Sketch라 불리는 통합프로그래밍 개발환경(IDE, Integrated Development Environment)⁶⁾을 제공한다. 스케치는 사실상 C++과 유사한 프로그래밍 언어로 일반 C, C++과 다른 점은 H/W 장치를 제어하거나 그 상태를 읽어들이는 기능이 편리한 함수로 제공된다는 것이다⁷⁾. 그림 1.3에는 입문용으로 쓰이는 아두이노 보드와 키트 및 쉴드 보드의 사례를 보였다.

Digital I/O -pinMode() -digitalWrite() -digitalRead()	Analog I/O -analogRead() PWM out -analogWrite()	Communication -Serial : RS232 -Stream : I2C	Interrupt -attachInterrupt() Tone Control tone()
---	--	---	---

6) IDE는 소스를 입력하거나 편집할 수 있는 편집기와 프로그램 소스를 수행코드로 변환하는 컴파일러, 그리고 수행 코드를 H/W 장치(여기서는 아두이노보드)에 탑재하는 다운로드가 하나의 프로그램에 내장된 개발용 프로그램을 말한다.

7) 함수에 관한 자세한 설명은 공식 사이트의 Language Reference 부분을 참조하기 바란다. <https://www.arduino.cc/en/Reference/HomePage>

1.2 실습 환경 및 학습 내용



그림 1.4(a) Arduino 교육용 실습키트
Ardu-Ez(제작사: RnU)

그림 1.4(a)에는 아두이노 H/W 실습도구로 활용될 RnU社의 Ardu-Ez 교육용 키트를 보였다⁸⁾. 그림 1.4(b)에는 아두이노 보드의 S/W 프로그래밍 개발 환경인 Sketch의 수행 프로그램을 보였다. Sketch는 소스 편집기와 컴파일러 및 업로더⁹⁾가 모두 내장된 아두이노를 위한 통합 개발환경(IDE: Integrated Development Environment) 프로그램이다¹⁰⁾.

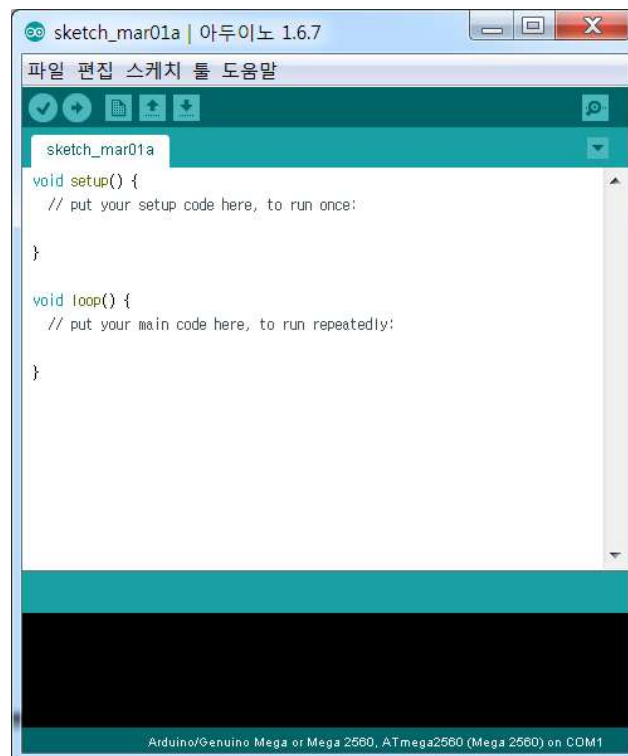


그림 1.4(b) 아두이노 프로그램 Sketch

8) http://irnu.co.kr/s03_equipment/s03_main_01.html

9) 업로더는 개발된 수행프로그램을 아두이노 보드에 탑재하는 유틸리티이다. PC에서 프로그램을 수행할 때는 이런 기능은 필요없지만 수행할 프로그램이 아두이노의 프로세서에서 수행되어야 하기 때문에 마이크로컨트롤러 보드 혹은 임베디드 시스템에서는 이러한 업로더 유틸리티 프로그램이 필요하다. 다른 개발환경에서는 다운로드라고 불리는 경우도 많다.

10) IDE는 최근의 프로그램 개발 도구의 경향이라고 할 수 있다. 과거 편집기, 컴파일러 등 여러 가지 독립된 유틸리티 프로그램들을 클릭만으로 간편하게 호출하여 사용할 수 있고 여러 가지 옵션 선택을 편리하게 설정할 수 있는 장점이 있다.

본 키트를 통하여 표 1.1과 같은 다양한 I/O 장치와의 인터페이싱 연습을 실행할 수 있다. 좌측 열의 장 번호는 본 교재의 장(chapter) 번호와의 연관성을 보인 것이다. 항목 우측 열의 **상부**는 H/W 장치를 보인 것이며 **하부**는 관련된 컴퓨터 구조상의 기술적 요소를 보인 것이다.

표 1.1 Ardu-Ez에서 제공하는 IO 장치들

장 번호	항목	학습 내용
2		<p>LED</p> <p>프로그램에 의해 디지털 데이터(논리 1 혹은 0)를 출력하면 LED가 점등 혹은 소등된다. 디지털 데이터를 입출력하는 GPIO 장치에 대하여 학습한다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-digitalWrite
	GPIO	
3		<p>Push Button Switch</p> <p>프로그램에 의해 외부 장치(스위치)의 디지털 데이터(논리 1 혹은 0)를 입력한다. 디지털 데이터를 입출력하는 GPIO 장치에 대하여 학습한다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-digitalRead
	GPIO	
4		<p>Buzzer</p> <p>프로그램에 의해 디지털 데이터(논리 1 혹은 0)를 출력하면 부저가 일정한 톤의 음향을 발생하거나 정지한다. 디지털 데이터를 입출력하는 GPIO 장치에 대하여 학습한다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-digitalWrite
	GPIO	
5		<p>스피커</p> <p>특정 주파수의 파형 신호를 공급하여 지정한 음정의 소리를 출력한다.</p>

		(Timer)	<ul style="list-style-type: none"> ■ 관련 아두이노 함수-tone
6		직렬통신	<p>아두이노보드와 사용자간의 명령 및 상태 교신을 위한 데이터 통신을 지원한다. USB 연결로 직렬통신을 흉내(emulation) 낸다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-Serial.println
		RS-232	
7		PIR 센서	<p>열을 감지하는 적외선 센서. 열반응이 감지되면 논리 1을 출력한다. 감도와 응답시간은 knob을 통해 제어할 수 있다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-digitalRead
		GPIO	
8		초음파센서	<p>입력되는 펄스의 길이를 이용하여 측정물체와 의 거리를 산출할 수 있다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-pulseIn
		GPIO	
9		DC 모터	<p>DC(직류) 모터에 인가되는 전류의 양을 PWM으로 통제하여 모터의 회전 속도를 제어한다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-analogWrite
		PWM Timer	
10		스텝모터	<p>스텝 모터에 인가되는 펄스의 개수 만큼 회전, 인가되는 펄스의 주파수에 따라 속도 변환을 시행한다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-tone
		(Timer)	
11		조도센서	<p>조도센서의 출력 전압의 값을 ADC를 통해 디지털 값으로 읽어들인다.</p> <ul style="list-style-type: none"> ■ 관련 아두이노 함수-analogRead
		ADC	

12		Piezo 센서	압력을 디지털 값으로 읽어들이 수 있다. ■ 관련 아두이노 함수-analogRead 피에조를 스피커와 같은 출력장치로 간주하여 소리를 발생시킬 수 있다. ■ 관련 아두이노 함수-tone
		압력 입력:ADC 소리출력:(Timer)	
13		LCD	16*2개의 문자를 화면에 출력한다. 4비트 GPIO 기반으로 인터페이싱되어 있다. ■ 관련 아두이노 함수-GPIO함수등을 이용하여 제작 ■ 관련 Ardu-Ez 클래스-LCD
		GPIO	
14		인터럽트	메인 프로그램 수행 도중에 H/W 단자로 인터럽트가 유입되면 하던 작업을 잠시 정지하고 인터럽트 서비스 루틴으로 분기하여 미리 설정한 루틴을 수행한 후 복귀한다. ■ 관련 아두이노 함수-attachInterrupt
		Interrupt	
15		LED 모듈	8개의 LED 모듈을 I2C 통신을 통해 점등/소등을 제어한다. GPIO 확장 소자 (PCA9535)에 I2C 기반의 통신으로 구현한다. ■ 관련 아두이노 클래스-Wire ■ 관련 Ardu-EZ 클래스-LED
		I2C	
16		세븐 세그먼트 디스플레이	4 단위의 세븐 세그먼트 표시장치에 원하는 문자 혹은 숫자를 출력한다. GPIO 확장 소자(PCA9535)에 I2C 기반의 통신으로 구현한다. ■ 관련 아두이노 클래스-Wire ■ 관련 Ardu-Ez 클래스-FND
		I2C	
17		온도센서	I2C 기반의 온도 센서(TMP102)에 연결된 소자에서 현재의 온도를 읽어낸다. ADC가 내장되어 디지털 값으로 읽어낸다. ■ 관련 아두이노 클래스-Wire ■ 관련 Ardu-Ez 클래스-Temperature
		I2C	

1.3 아두이노 보드의 프로세서

아두이노 보드에서는 Atmel사의 AVR 프로세서를 사용하였다¹¹⁾. AVR 프로세서는 CPU core와 flash ROM, RAM 및 GPIO, 타이머, ADC, RS-232 UART 제어기, I2C 제어기 등의 주요 I/O 제어 장치를 한 칩에 모두 내장한 소형 마이크로컨트롤러(microcontroller)이다¹²⁾. 본 교재에서 사용하는 Ardu-Ez 아두이노 보드는 Atmega 2560 AVR 프로세서를 내장하였다. Atmega 2560의 특징은 다음과 같다.

- 8-Bit Microcontroller - RISC Architecture
 - 135 Powerful Instructions - Most Single Clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
- Non-volatile Memory Segments
 - 256KBytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
- **In-System Programming** by On-chip Boot Program
- **JTAG** (IEEE® std. 1149.1 compliant) Interface
- 8-bit GPIO(A~L) Ports 11 EA
- 8-bit, 16-bit Timer/Counters
- PWM Channels with Programmable Resolution from 2 to 16 Bits

11) AVR에 대한 상세한 자료는 다음 URL을 참조하기 바란다.

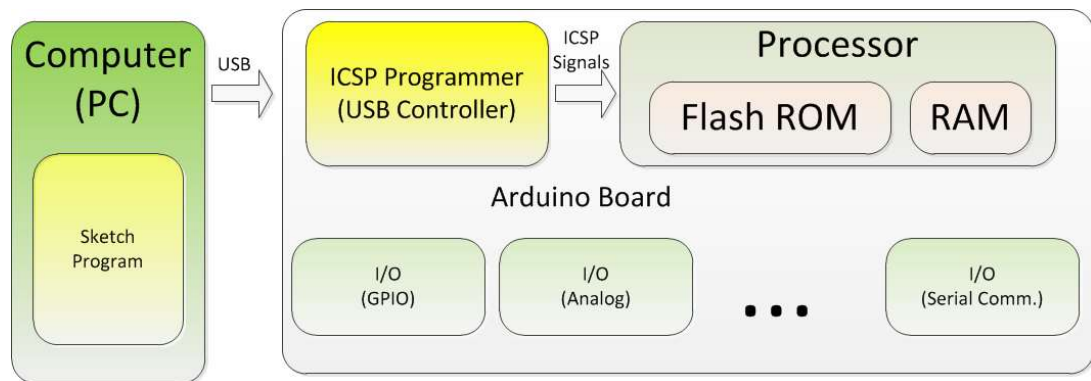
https://en.wikipedia.org/wiki/Atmel_AVR

12) microcontroller에 대한 정의는 시대에 따라 혹은 용도에 따라 조금씩 다르게 정의되고 있으나 대체로 8비트(혹은 16비트까지) CPU core와 ROM, RAM 및 여러 I/O 제어기를 내장한 프로세서를 말한다. 이것은 오늘날 embedded processor와 유사한 기능을 갖고 있으나 임베디드 프로세서는 32 비트 프로세서를 내장하고 보통 embedded Linux와 같은 OS를 탑재할 수 있는 프로세서류를 말하는 경향이 있다. 휴대폰 용 임베디드 프로세서처럼 더 큰 기능이 내장된 프로세서는 AP(Application Processor)로 분류하는 경향이 있다.

- 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
- Four Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte Oriented 2-wire Serial Interface => I2C
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

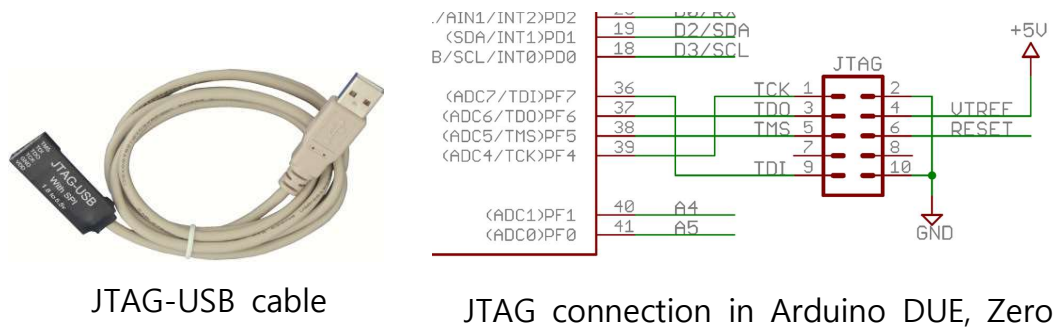
In-System Programming

ICSP(In Circuit Serial Programming)이라고도 함. 보드에 프로세서가 장착된 상태에서 프로세서의 메모리에 프로그램을 탑재하는 동작을 말한다. 보통 PC의 USB를 통해 보드 내에 장착된 USB 제어가 ICSP 프로그래머 역할을 담당한다. AVR의 경우 6핀 혹은 10핀의 ICSP 단자를 통해 프로그램한다.



👉 JTAG(Joint Test Action Group)

조립된 보드의 검증과 실험을 위해 1985년부터 수립되기 시작한 산업 표준. 2x5, 2x7, 8핀 등의 다양한 콘넥터가 사용된다. 원래는 디버깅 목적이지만 대상이 프로세서인 경우에는 ISP처럼 펌웨어를 다운로드하는 역할에도 활용된다. 컴퓨터와 연결될 때는 아래 그림처럼 USB를 통한 케이블을 쓰는 것이 일반적이다. 타겟 보드 내에서도 USB 제어가 JTAG를 흉내 내는 경우가 있기 때문에 일반 USB 케이블만 있어도 JTAG 기능을 활용할 수 있는 경우도 있다. 실험 키트 Ardu-Ez와 Arduino Uno에는 JTAG 단자는 지원되지 않는다. 일부 버전의 아두이노만 지원되고 있다.



Atmega 계열 프로세서의 내장 기능을 간략히 비교하여 보이면 다음과 같다.

표 1.2 Atmega 계열 프로세서의 내장 기능 비교

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

그림 1.5에 Atmega 2560 프로세서의 블록 다이어그램을 보였다.

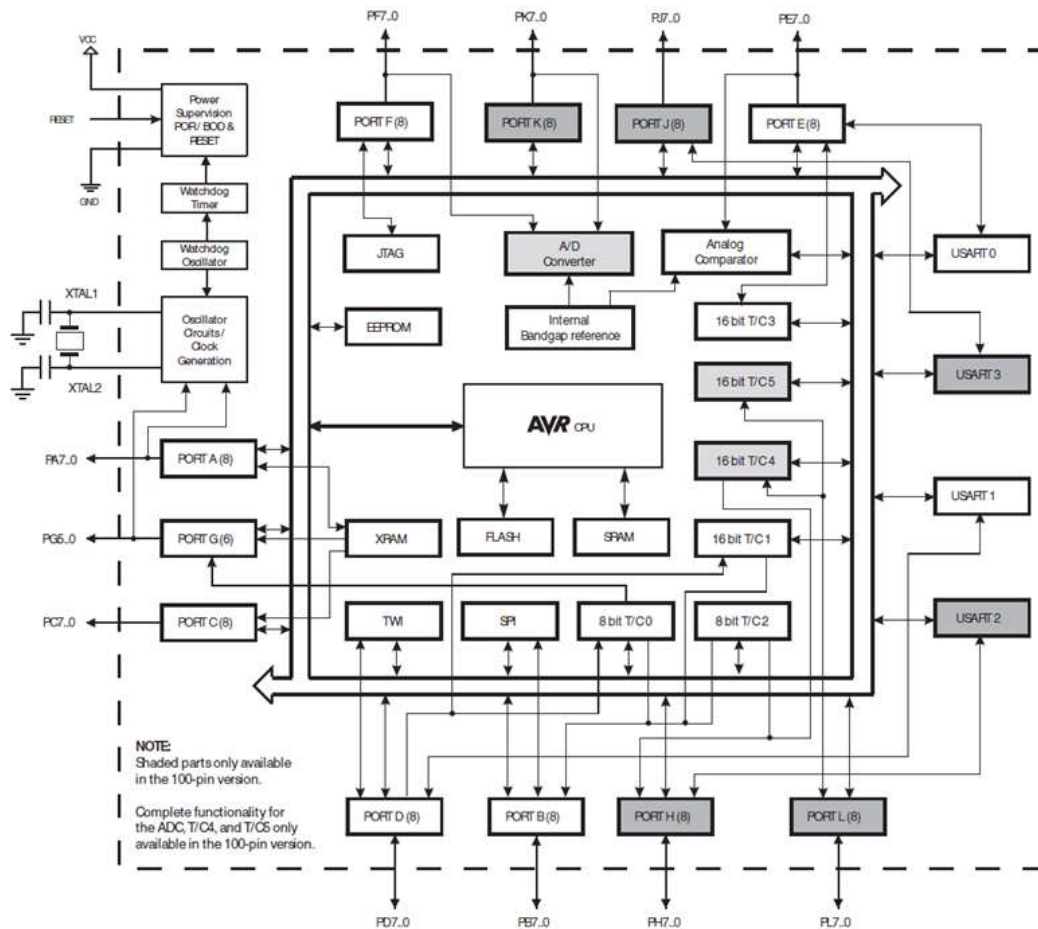


그림 1.5 Atmega 2560 프로세서의 블록 다이어그램

아두이노 Mega2560에 관한 상세한 자료는 아래 공식 사이트의 자료를 참조하기 바란다. <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

1.4 어떤 개념을 배울 것인가?

위에 언급한 내용은 컴퓨터 입문자에게는 매우 낯선 기술 용어들로 이루어져 있음에 틀림이 없다. 사실 아두이노 프로그램에 이런 내용들이 반드시 필요한 것도 아니다. 그러나 아두이노를 통해 컴퓨터 설계 실력을 함양하고자 하는 본 교과 과정의 목표를 생각해 보면 위 내용 중에 본 교재를 통해 꼭 알아야 할 개념은 다음과 같다. 각 기술 용어에 대한 상세한 설명은 해당 기술의 본문에서 상세히 다루게 되므로 여기서는 용어만 간략히 살피고 넘어가기 바란다.

□ GPIO(General Purpose Input Output)

범용 입출력장치. 디지털 데이터를 받아들이거나 출력하는 장치를 말한다. 디지털 데이터는 논리기호 1, 0 혹은 심볼 High, Low를 뜻하는 이는 물리적인 전기 신호 5V, 0V 혹은 3.3V, 0V를 출력하거나 그 상태를 읽어 들이는 장치를 말한다. 디지털 신호에 대한 특징을 표 1.3에 3가지 관점으로 조망하였다.

GPIO는 최근의 임베디드 프로세서에 기본적으로 내장되어 있다. 초기화 과정에서 출력 혹은 입력 장치로 설정하여 사용하거나 프로그래밍을 통해 수시로 입출력을 바꾸어 사용할 수 있다.

출력장치의 단순한 응용 사례를 들어보면 LED(Light Emitting Diode)를 들 수 있다. LED에 논리 1(5V, H)을 출력하여 LED를 점등할 수 있고, 논리 0(0V, L)을 출력하여 소등할 수 있다. 입력장치의 단순한 사례로는 스위치(switch) 장치를 들 수 있다. GPIO를 입력으로 설정하여 스위치의 H, L 상태를 GPIO를 통해 읽어 들여 ON, OFF 상태 여부를 프로그램으로 점검할 수 있다.

표 1.3 디지털 신호의 3가지 관점(논리, 심볼, 물리)

구분	상태 값		비고	
논리기호	1	0	bit. 8개가 모이면 byte라 칭함	
심볼	H	L	High, Low 문자로 디지털 신호선의 상태를 표기	
물리적 전기 신호	5V	0V	정통 디지털 회로의 전압	실제로 회로 상에서 흐르는 전압. 전압 값이 중요한 것이 아니라 두 상태의 전압 값이 인정되는 것이 디지털 회로의 특징이다. 이밖에 다른 사례의 전압이 활용되기도 한다. 동일 보드에서 여러 전압의 조합이 활용될 수도 있다.
	3.3V	0V	최근의 저전력 추구형의 디지털 회로 전압	
	1.8V	0V	CPU core 내부의 동작 전압	
	-12V	+12V	원거리 통신을 위한 RS-232의 사례	

아두이노의 경우 digitalWrite() 함수로 GPIO 장치를 통해 디지털 데이터를 출력할 수 있으며, digitalRead() 함수를 통해 외부 장치의 디지털 상태 값을 입력 받을 수 있다.

□ RS-232

직렬통신의 표준규약 중의 하나. 접지선과 송신선로 1EA, 수신선로 1EA 총 3개의 신호선으로 다른 컴퓨터 장치와 통신을 시행하기 위한 표준 규약이다. 데이터는 전용 통신 장치인 UART(Universal Asynchronous Receiver and Transmitter) 제어기(Controller) 내부에서 직렬화 혹은 병렬화 변환된다. 아두이노의 경우 문자열을 출력하는 print, println 등이 지원된다. RS-232로 출력된 문자는 Sketch 프로그램 내부의 Serial Monitor를 통해서 받아볼 수 있으며 또, 이를 통해서 아두이노에 전달된 문자, 문자열을 받아볼 수 있는 read 함수 등을 지원하고 있다.

❑ PWM(Pulse Width Modulation)

디지털 펄스(pulse)의 신호의 듀티비(duty ratio)를 제어하여 전력공급을 제어하는 기술을 말한다. 듀티비는 펄스 신호에서 H폭이 차지한 비율(0~100%)을 말하는데 이 비율의 제어를 통해 H가 공급되는 시간을 제어하여 모터 등의 전류 제공시간을 통제하여 그 속도를 제어하는데 활용할 수 있다. 아두이노가 사용하는 Atmega에서는 타이머를 이용하여 듀티비를 제어한다. analogWrite() 함수는 이러한 듀티비를 256 단계로 PWM 신호를 생성할 수 있다.

❑ ADC(Analog Digital Converter)

아날로그 신호(전압)를 디지털 값으로 변환하는 장치이다. 본 교재의 활용 사례로 성명하면 온도가 변하면 이는 전압의 변화로 바꿀 수 있는 온도 센서가 있는데 이 전압을 읽어서 온도를 측정할 수 있다. 아두이노가 사용하는 Atmega 프로세서에는 ADC가 내장되어 있으며 이는 analogRead() 함수를 이용해 그 값을 읽어볼 수 있다.

❑ Interrupt

인터럽트는 H/W 신호의 유입에 의해 메인 프로그램의 수행을 잠시 멈추고 미리 준비한 ISR(Interrupt Service Routine)을 수행하게 하는 프로세서 수행 + 메카니즘이다. H/W 신호는 프로세서의 단자를 통해 H혹은 L의 상태로 유입된다. 아두이노는 인터럽트 설정을 위한 attachInterrupt() 함수를 통해 ISR의 함수 이름을 지정하게 되어 있다. 아두이노는 지정한 주파수의 파형을 발생시키는 tone()함수가 GPIO 단자를 통해 원하는 주파수의 신호를 생성하는데 내부에서 인터럽트를 활용한다.

❑ Timer

타이머는 특정 주파수 및 파형의 클럭 신호를 생성하는 장치를 말하며 이를 이용해 PWM 신호 혹은 프로그램을 통해 원하는 주파수의 펄스신호를 생성한다. 아두이노의 경우 PWM 신호를 생성할 때 내부의 타이머 장치를

활용한다. 타이머 장치는 보통 카운터(counter) 기능도 갖고 있다. 카운터는 입력되는 디지털 펄스 신호의 개수를 세는 장치로서 이를 이용해 프로그램한 만큼의 펄스를 세면 인터럽트 등을 발생하게 할 수 있다. 이 같은 기능을 잘 활용하면 경과 시간을 정확히 세거나 스톱워치 등을 구현하는 데 활용할 수 있다.

□ I2C(I²C, Inter IC Communication)

외부 장치가 간단하면 GPIO를 통해 제어할 수 있고 조금 더 복잡해지면 Controller를 사용한다. 데이터 전송량이 많으면 병렬로 접속이 이루어지지만 데이터량이 적으면 직렬로 연결한다. 직렬로 연결하면 IC와 IC 사이의 신호 전송 선로를 손쉽게 구성할 수 있기 때문이다. I2C는 제어기와 프로세서간의 직렬통신 표준 프로토콜 중의 하나이다. RS-232와 다른 점은 RS-232가 컴퓨터와 컴퓨터 간의 통신으로 비교적 먼 거리를 지원하는 데 반해 I2C는 디바이스 vs 디바이스 통신으로 동일 보드 상에서 가까운 거리(약 30cm 이내)간의 통신에 활용된다는 것이다. 공식 아두이노에서는 이를 지원하는 함수는 없지만 Ardu-Ez에서는 Wire 클래스로 이를 지원하고 있다.

1.5 고찰

1. 실습 장비 Ardu-Ez에서 WiFi, Ethernet 통신을 구현하고자 한다. 이 기능의 구현을 위해 Shield가 필요한가?
2. Ardu-Ez 장비는 전원을 별도로 공급해야 동작하는가?
3. Arduino 보드에 프로그램하고 나면 나중에 다시 켜면 그 프로그램은 사라질 것인가 아니면 그대로 남아 있을 것인가? 그 이유를 설명해 보자.
4. ISP혹은 ICSP란 무엇인가? 실습을 위해 ISP cable은 필요 없는가?
5. JTAG에 대해 설명해 보자. Arduino-Ez에서 JATG를 활용할 수 있는가?
6. 일반 CPU, microcontroller, embedded processor 및 Application Processor 등에서 서로 공통적인 특징과 이들 간의 서로 구분 짓는 특징을 제시해 보자.
7. **본 아두이노 프로그래밍 교재를 통해 익혀야 할 컴퓨터 설계상의 기술적 주제(개념) 7가지를 제시하고 이를 간단히 설명하라¹³⁾.**
8. Arduino Uno 보드 내의 주요 연결단자(콘넥터, 점퍼 등)의 용도에 대해 기술해 보자. Uno-그림 1.6(a), (b) 참조. Mega 2560-그림 1.6(c) 참조.

13) 현 단계에서는 그 의미를 이해하기 쉽지 않을 것이다. 본 교재의 모두 마치면 다시 이 질문에 대답할 수 있을 것인지 염두에 두고 학습하기 바란다.

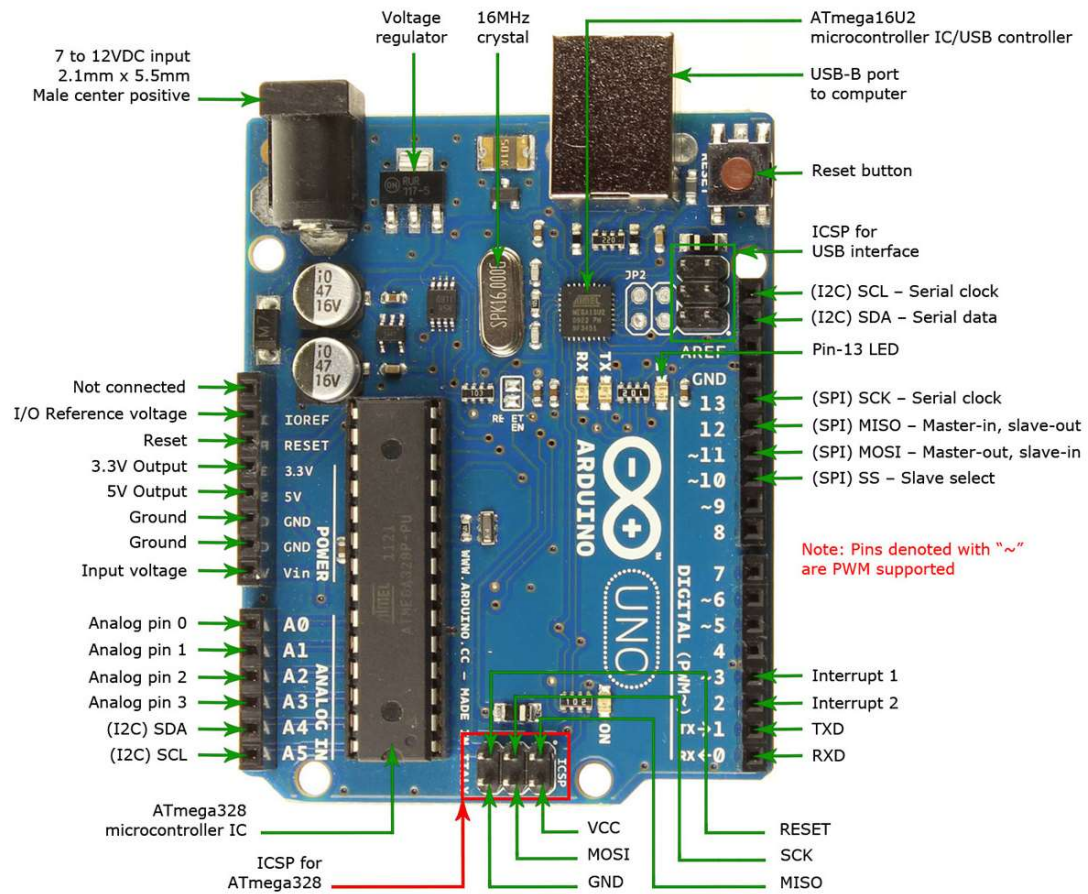


그림 1.6(a) Arduino Uno의 연결단자

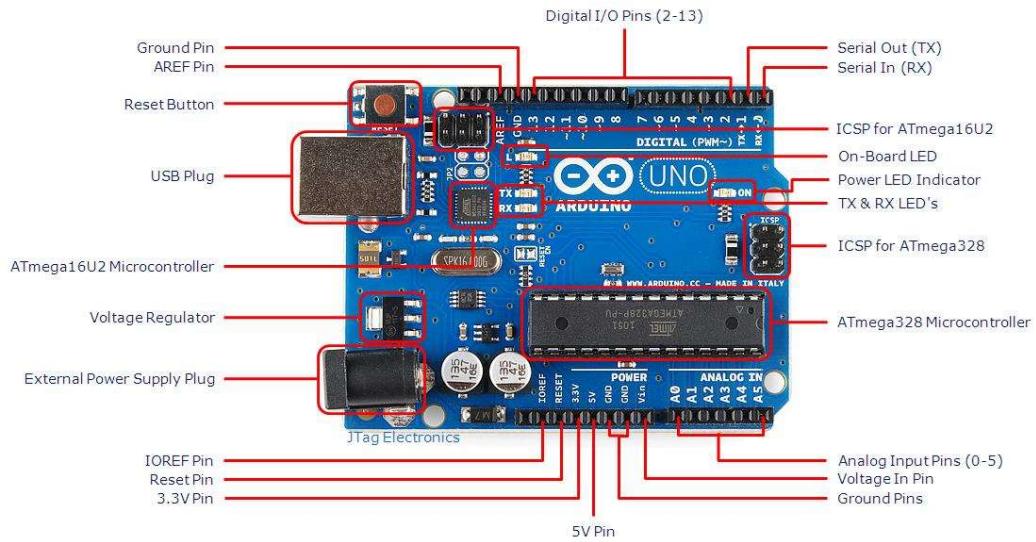


그림 1.6(b) Arduino Uno의 연결단자

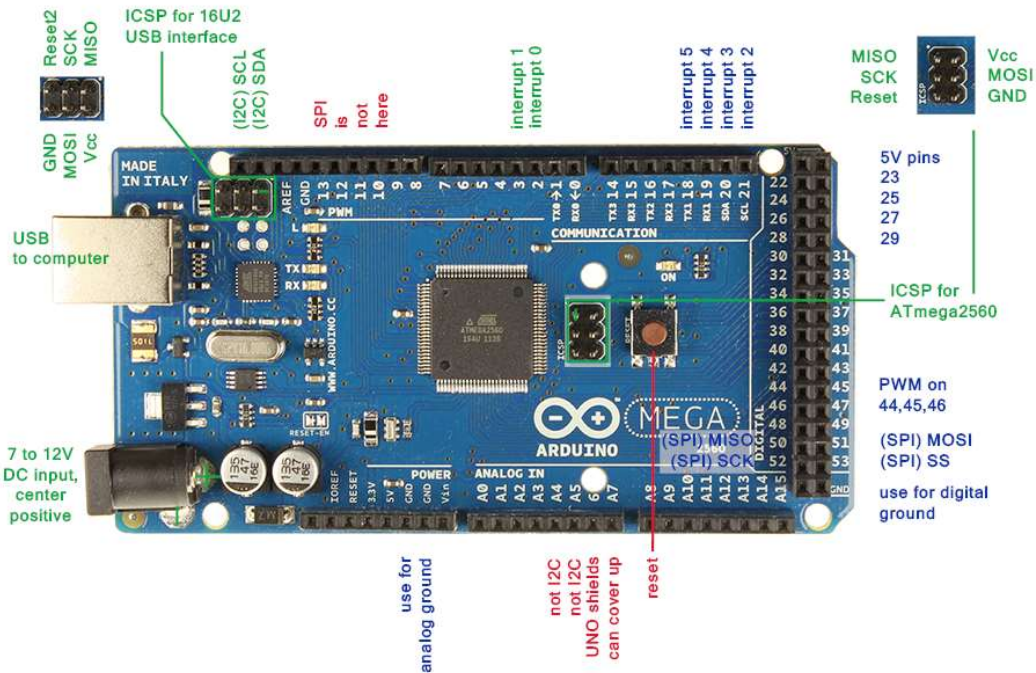


그림 1.6(c) Arduino Mega 2560의 연결단자