

1. 아두이노 사용 준비

1. 아두이노 하드웨어

아두이노는 Atmel 사의 마이크로컨트롤러를 사용하여 만들어진 마이크로컨트롤러 보드와 아두이노 보드를 이용하여 프로그램을 개발할 수 있도록 해주는 프로그래밍 환경을 통틀어 이르는 말이다. 아두이노는 2005년 이탈리아의 Ivrea에서 예술가와 디자이너들이 마이크로컨트롤러 기반의 장치를 쉽게 만들 수 있도록 하기 위한 목적으로 시작되었다. 이처럼 아두이노는 비전문가들을 대상으로 시작된 프로젝트이므로 간단하고 사용하기 쉬운 하드웨어와 소프트웨어 환경이 그 특징 중 하나이다.

아두이노 하드웨어는 하나의 보드만을 가리키는 것이 아니며 아두이노 환경에서 프로그램을 작성할 수 있는 여러 가지 보드를 통틀어 가리킨다. 아두이노에는 보드 제작에 사용한 마이크로컨트롤러에 따라 여러 종류의 보드가 존재하며, 동일한 마이크로컨트롤러를 사용한 보드라 할지라도 확장성과 용도에 따라 여러 가지 보드가 존재한다. 아두이노 공식 홈페이지에 나열된 아두이노 보드만도 20여종에 이르고 있다.¹⁾ 이 중 가장 기본이 되는 보드는 아두이노 우노 보드이다.

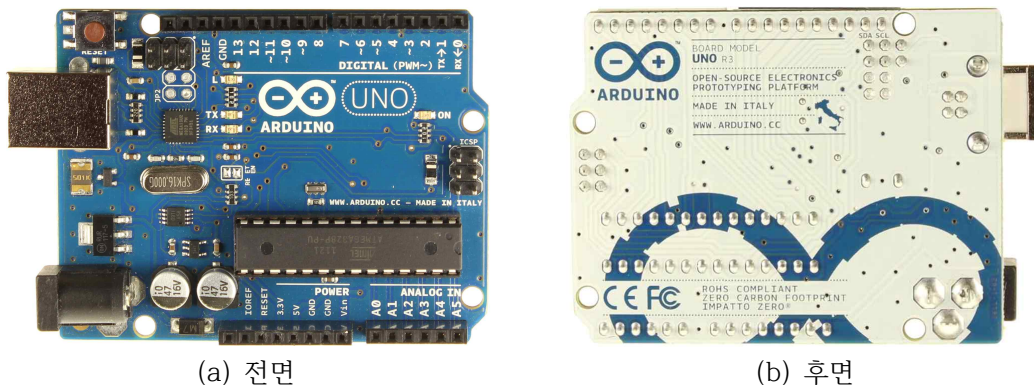


그림 1. 아두이노 우노 보드²⁾

아두이노 우노 보드는 ATmega328P 마이크로컨트롤러를 중심으로 외부 장치 연결을 위한 핀헤더, 전원 연결 잭, 프로그램 다운로드를 위한 USB 커넥터, 부트로더와 프로그램 다운로드를

1) <http://arduino.cc/en/Main/Products>

2) 이미지 출처: <http://arduino.cc/en/Main/ArduinoBoardUno>

위해 사용할 수 있는 ISP 연결 커넥터 등으로 구성되어 있다.

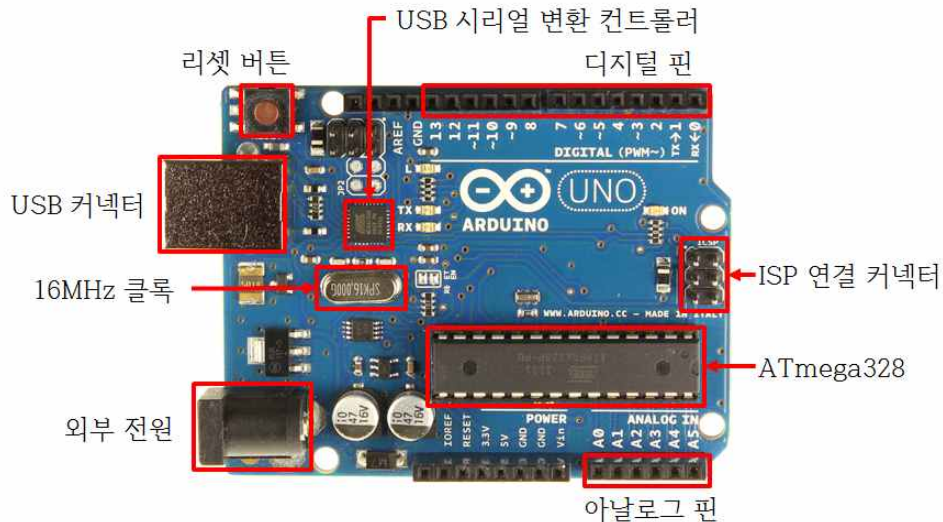


그림 2. Arduino UNO 보드³⁾

아두이노 환경에서 프로그램을 작성한 후 아두이노 보드에 프로그램을 다운하고 설치하기 위해서는 USB나 ISP 방식을 사용할 수 있다. ISP 방식을 사용하기 위해서는 별도의 장치가 필요하므로 아두이노의 특징 중 하나인 부트로더를 굽기(burn)위해서만 주로 사용하고 프로그램 다운을 위해서는 USB 방식이 기본으로 사용된다. 아두이노 우노에는 USB 연결을 통해 프로그램을 다운하고 설치할 수 있도록 USB-시리얼 변환 컨트롤러가 포함되어 있으므로 컴퓨터와 아두이노를 USB로 연결하는 것만으로 프로그램을 작성하여 실행해 볼 수 있다.⁴⁾ 전원 역시 USB를 통해 공급 가능하므로 별도의 전원을 사용하지 않아도 된다. 다만, 프로그램을 다운로드 한 후 컴퓨터와 연결하지 않고 동작시키기 위해서는 외부 전원을 연결하여야 한다. 아두이노 우노의 주요 사양은 표 1과 같다.

3) 이미지 출처: <http://arduino.cc/en/Main/ArduinoBoardUno>

4) 부트로더는 최초 1회 설치만 필요하며 아두이노 보드 내의 마이크로컨트롤러에는 부트로더가 설치된 상태로 판매되므로 별도로 부트로더를 설치할 필요는 없다. 다만, 여러 가지 이유로 마이크로컨트롤러 칩을 교체하는 경우에는 부트로더를 설치하여야 하며 이를 위해서는 전용 장치가 필요하다.

항목	내용	비고
마이크로컨트롤러	ATmega328	
동작 전압	5V	
입력 전압	7V~12V	추천 입력 범위
디지털 입출력 핀	14개	6개 PWM 출력 핀
아날로그 입력 핀	6개	
플래시 메모리	32KB	ATmega328, 부트로더 0.5KB
SRAM	2KB	ATmega328
EEPROM	1KB	ATmega328
클록 주파수	16MHz	

표 1. Arduino UNO 주요 사양

아두이노 우노는 Atmel사의 16MHz 마이크로컨트롤러인 ATmega328을 사용하며 총 20개의 입출력 핀을 통해 다양한 주변 장치를 연결할 수 있으며 연결을 위해 핀 헤더를 제공하고 있다. 다운로드 할 수 있는 프로그램의 최대 크기는 32KByte이지만 아두이노의 특징 중 하나인 부트로더가 그 중 일부를 사용하므로 실제로는 부트로더 크기를 뺀 나머지를 프로그램을 위해 사용할 수 있다. 이외에도 프로그램 실행 중 데이터 저장을 위한 2KByte 크기의 SRAM과 전원이 꺼진 이후에도 데이터가 보존되는 1KByte 크기의 EEPROM을 사용할 수 있다.

아두이노의 또 다른 특징 중 하나는 쉴드(shield)라고 불리는 다양한 확장 모듈이 존재한다는 점이다. 아두이노 우노를 포함하여 핀 헤더가 제공되는 아두이노 보드들은 동일한 배열의 핀 헤더를 사용하고 있으므로 누구든 손쉽게 원하는 기능을 수행하는 쉴드를 제작할 수 있다. 아두이노 공식 홈페이지도 여러 종류의 쉴드들이 판매되고 있으며⁵⁾ 이외에도 오픈 소스에 따라 다양한 쉴드들이 여러 회사에서 제작되어 판매되고 있다. 쉴드는 아두이노 보드에 수직으로 장착하여 사용하며 쉴드가 사용하는 핀이 중복되지 않는다면 여러 개의 쉴드를 적층하여 동시에 사용하는 것도 가능하다.

5) <http://arduino.cc/en/Main/Products>

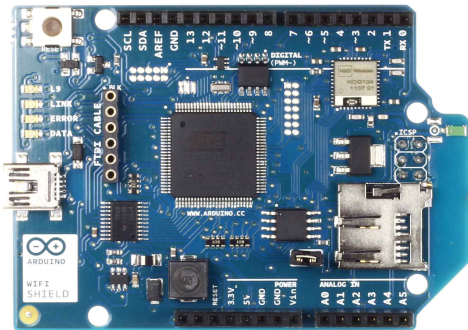
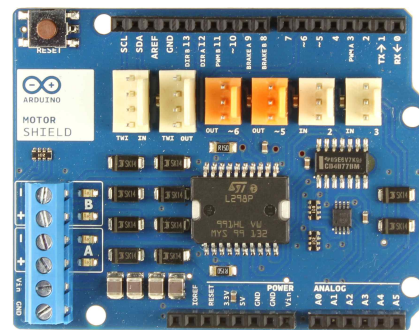
(a) 와이파이 쉴드⁶⁾(b) 모터 쉴드⁷⁾

그림 3. 아두이노 쉴드

2. 아두이노 소프트웨어

아두이노용 프로그램 개발을 위해 아두이노에서는 아두이노 보드와 마찬가지로 오픈 소스를 바탕으로 하는 통합 개발 환경(Integrated Development Environment, IDE)을 제공하고 있다. 아두이노 프로그램은 복잡하고 전문적인 기능들은 제외하고 초보자들도 쉽게 프로그램을 작성할 수 있도록 직관적이고 간편한 인터페이스를 제공하고 있다. 아두이노의 통합 개발 환경은 다양한 운영체제에서 실행이 가능하도록 Java로 개발되었으며 IDE를 통해 한 번의 클릭으로 코드를 컴파일해서 아두이노 보드에 프로그램을 업로드 하고 실행시킬 수 있다. 일반적으로 아두이노를 위해 만들어진 프로그램이나 코드를 스케치(sketch)라 부르며 단어의 의미 그대로 그림을 그리듯이 프로그램을 쉽게 작성할 수 있음을 의미한다. 아두이노 프로그램은 아두이노 공식 홈페이지⁸⁾에서 무료로 다운받을 수 있다.

6) 이미지 출처: <http://arduino.cc/en/Main/ArduinoWiFiShield>

7) 이미지 출처: <http://arduino.cc/en/Main/ArduinoMotorShieldR3>

8) <http://arduino.cc>



그림 4. 아두이노 소프트웨어 다운로드 페이지

현재 아두이노 정식 버전은 1.0.5이며 Windows 운영체제는 물론 Mac OS X와 Linux도 지원한다.⁹⁾ 아두이노 프로그램은 설치가 필요 없으며 압축을 해제하는 것만으로 사용할 수 있다. 윈도우즈 환경을 위한 인스톨 프로그램도 제공되지만 이는 별도의 압축 프로그램 없이 압축을 해제할 수 있도록 만들어진 것으로 압축파일과 내용은 동일하다. 다운 받은 파일을 'C:\Arduino'에 압축을 해제하였다고 가정하자. 'C:\Arduino\arduino.exe' 파일을 실행시켜 보면 그림 5와 같이 아두이노용 프로그램을 작성할 수 있는 개발 환경을 볼 수 있다.

9) 이 책의 모든 코드들은 Windows 7, 64bit 환경에서 테스트되었다.

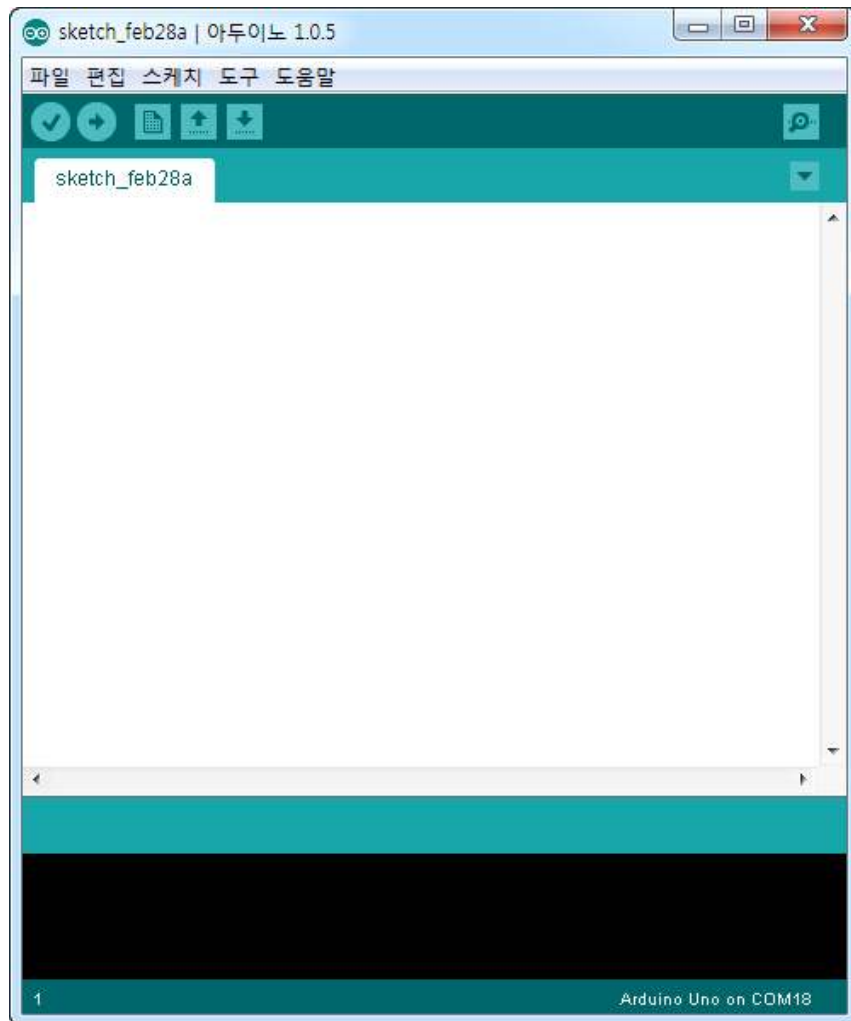


그림 5. 아두이노 개발 환경

컴퓨터와 아두이노 보드를 USB로 연결하면 아두이노 보드가 부팅을 거쳐 동작하기 시작한다. 처음 컴퓨터와 아두이노 보드를 연결한 경우에는 드라이버를 설치해 주어야 하며 아두이노 보드의 드라이버는 자동으로 설치되지 않으므로 직접 설치하여야 한다. 윈도우즈 제어판에서 장치 관리자를 열어보자. 아두이노 보드를 처음 PC와 연결하면 장치 관리자에 “알 수 없는 장치”가 나타난다.

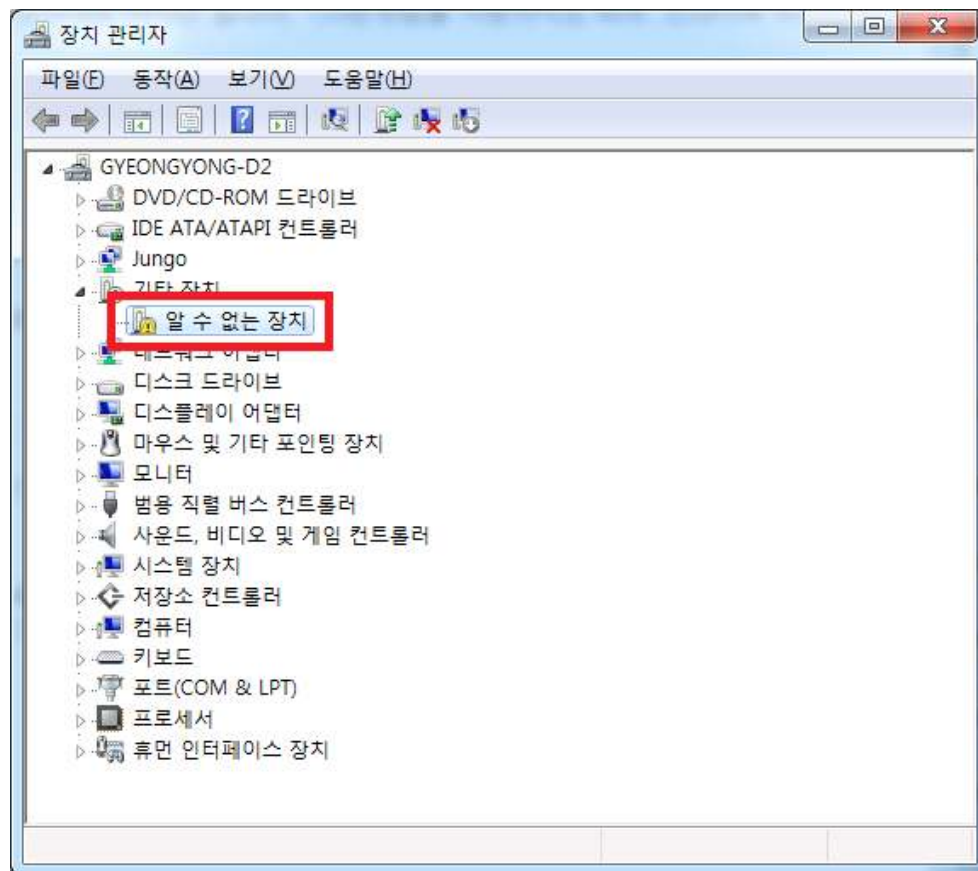


그림 6. 윈도우즈 장치 관리자

마우스 오른쪽 버튼을 눌러 “드라이버 소프트웨어 업데이트”를 선택하고 “컴퓨터에서 드라이버 소프트웨어 찾아보기”를 선택한 후 아두이노 프로그램이 설치된 디렉터리 아래의 ‘drivers’ 디렉터리를 지정하면 드라이버가 설치된다. 드라이버 소프트웨어 업데이트가 성공적으로 끝나면 장치 관리자에서 “알 수 없는 장치”는 사라지고 “Arduino UNO”가 나타난다.

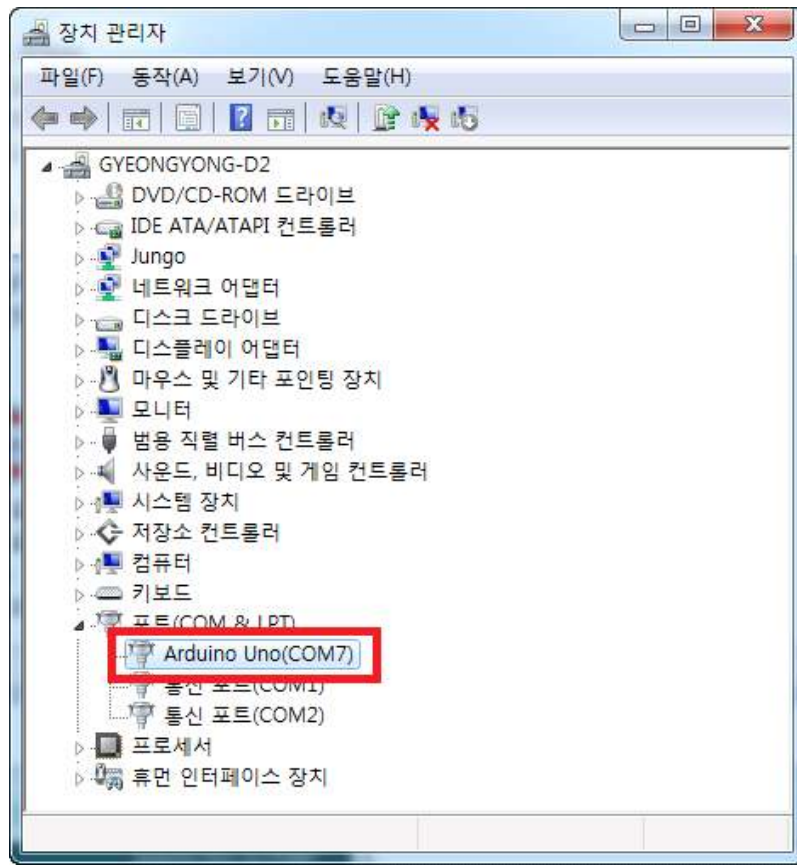


그림 7. 드라이버 설치 후의 윈도우즈 장치 관리자

아두이노 우노는 USB 연결을 사용하지만 USB 연결은 USB-시리얼 변환 컨트롤러를 거쳐 직렬 통신 포트를 통해 이루어지므로 장치 관리자에서 '포트' 부분에 나타난다. 이 때 포트 번호는 아두이노와 컴퓨터 사이의 통신에서 사용되는 번호이므로 반드시 기억하도록 한다.

드라이버 설치가 완료되었으므로 다시 아두이노 프로그램을 실행시켜 보자. 아두이노 보드는 여러 종류가 있으므로 사용하고자 하는 보드의 종류와 연결된 포트 번호를 아두이노 프로그램에서 지정해 주어야 한다. “도구 → 보드” 메뉴를 선택하면 사용 가능한 보드의 종류가 나열된다. 아두이노 우노 보드를 연결하였으므로 아두이노 프로그램에서도 'Arduino Uno'를 선택한다.

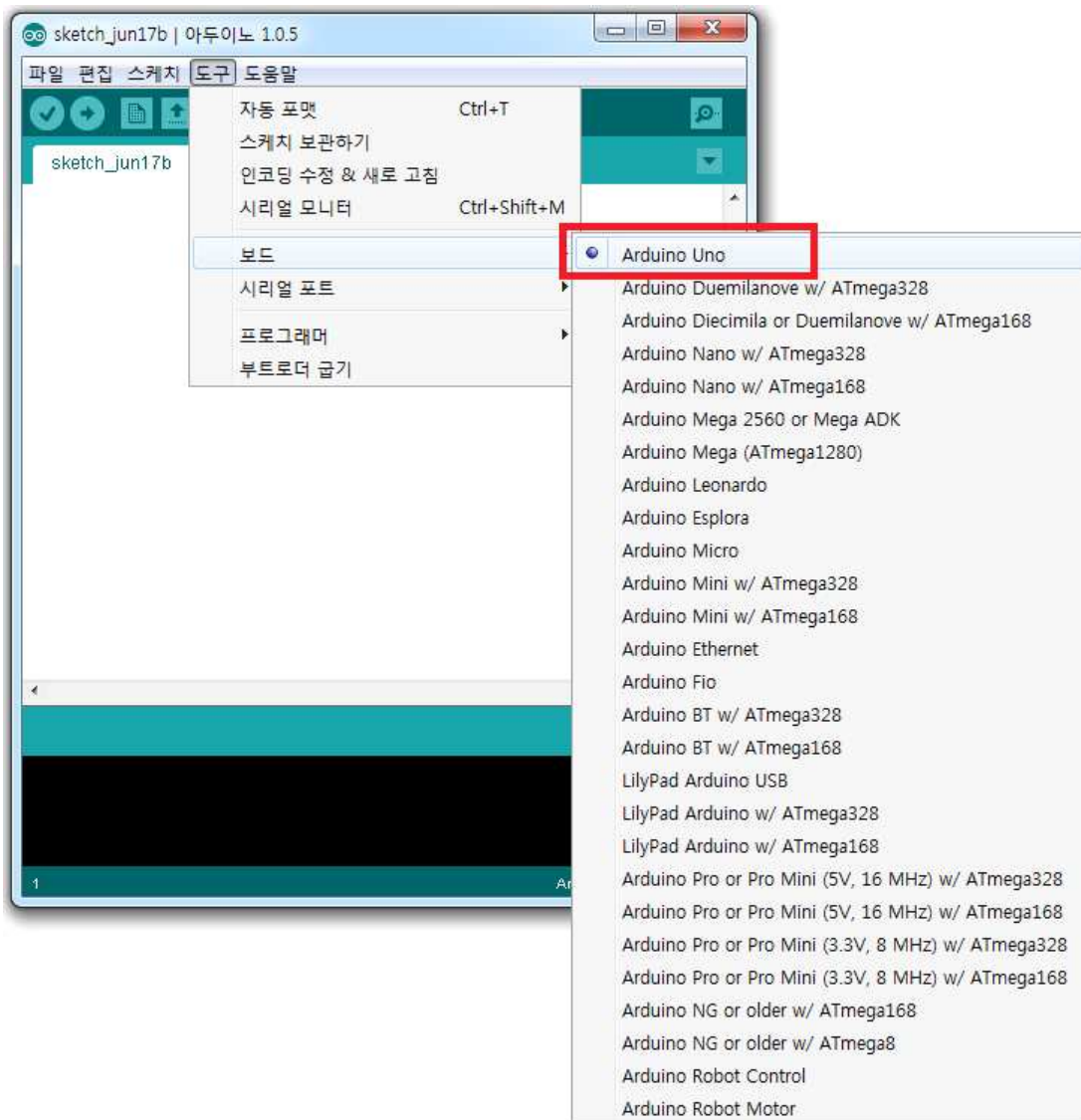


그림 8. 아두이노 프로그램에서 보드 선택

다음은 컴퓨터와 아두이노 보드 사이에 데이터 전송을 위해 사용할 포트를 설정해 주어야 한다. “도구 → 시리얼 포트” 메뉴를 선택하고 그림 7에서 아두이노 보드가 연결된 COM 포트를 선택해 준다.

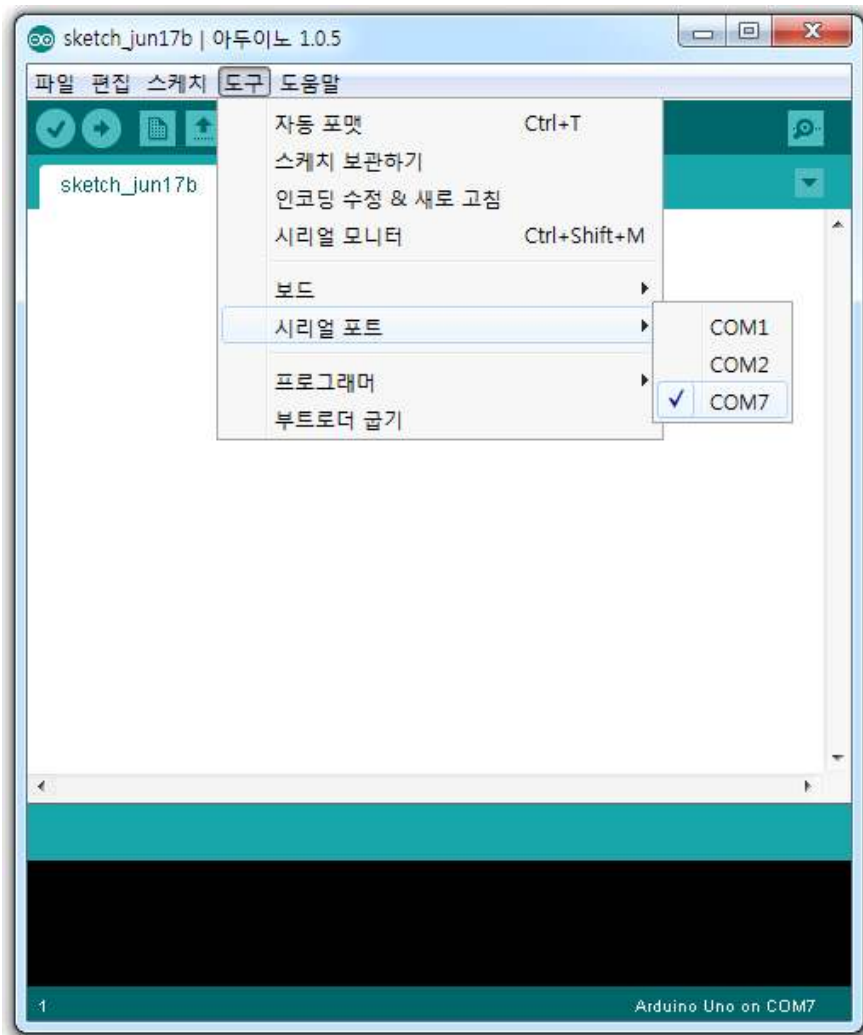


그림 9. 아두이노 프로그램에서 시리얼 포트 선택

이로써 아두이노용 프로그램을 아두이노 프로그램에서 작성할 수 있는 준비는 끝이다.

3. 아두이노 학습 보드

아두이노 학습을 위해서는 아두이노 보드 중 가장 기본이 되는 우노 보드를 사용할 것을 추천한다. 하지만 전자 회로에 대한 경험이 부족한 경우 회로를 구성하고 연결하는 일이 쉽지만은 않은 것이 사실이다. 따라서 이 책에서는 아두이노 우노 보드를 기반으로 실험에 필요한 다양한 주변장치들을 추가하여 복잡한 결선 없이 동작을 확인해 볼 수 있는 학습 보드인 DM-UStyle EX¹⁰⁾를 사용한다. 회로도도 공개되어 있으므로 우노 보드에 직접 연결하여 사용

10) 학습 보드는 <http://www.newtc.co.kr/>에서 구입할 수 있다.

하는 것도 가능하다.

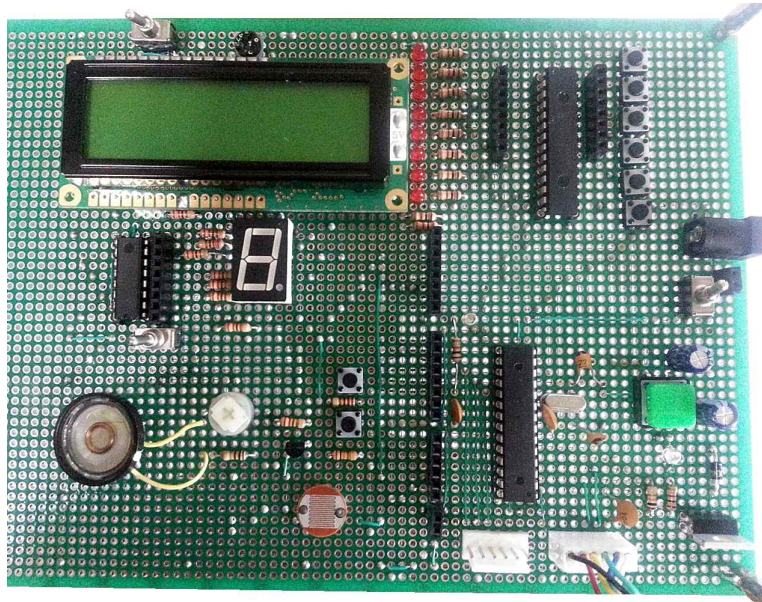


그림 10. DM-UStyle EX 학습 보드¹¹⁾

DM-UStyle EX 보드는 시리얼 및 ISP 연결을 지원한다. 아두이노 우노 보드의 경우 USB 연결을 지원하기 위해 별도의 마이크로컨트롤러가 포함되어 있지만 DM-UStyle EX 보드는 USB 연결은 지원하지 않으며 별도의 시리얼 업로더 모듈을 사용하여 시리얼 연결을 통해 사용 가능하다.



그림 11. USB-Serial 변환 모듈¹²⁾

DM-UStyle EX 학습 보드에 포함되어 있는 주요 장치들은 표 1과 같다.

11) 이미지 출처: <http://www.newtc.co.kr>

12) 이미지 출처: <http://www.newtc.co.kr>

장치	설명
버튼	풀업 및 풀다운 버튼
MCP23017	16개 디지털 입출력 확장 (8개 버튼과 8개 LED가 MCP23017 칩에 연결되어 있음)
LED	13번 디지털 핀에 연결된 LDE 및 PWM 테스트 용
가변저항	아날로그 입력 테스트 용
스피커	
온도 센서	
조도 센서	
74595	직렬 입력 병렬 출력 쉬프트 레지스터 (1자리 7 세그먼트 장치가 74595 칩에 연결되어 있음)
텍스트 LCD	
시리얼 업로드용 커넥터	업로드 시 자동 리셋 지원
ISP 커넥터	

표 1. DM-UStyle EX 학습 보드 사양

DM-UStyle EX 학습 보드에는 많은 장치를 포함되어 있지만 ATmega328의 입출력 핀 수가 제한되어 있는 만큼 많은 개수의 입출력 핀을 필요로 하는 장치를 포함시킬 수는 없으므로 간단히 연결할 수 있는 몇 종류의 확장 모듈들을 추가로 사용한다. 확장 모듈에는 풀다운 및 풀업 버튼으로 구성되는 버튼 모듈, 8개의 LED로 구성되는 LED 모듈, 그리고 1자리 및 4자리 7 세그먼트 모듈 등이 있다.

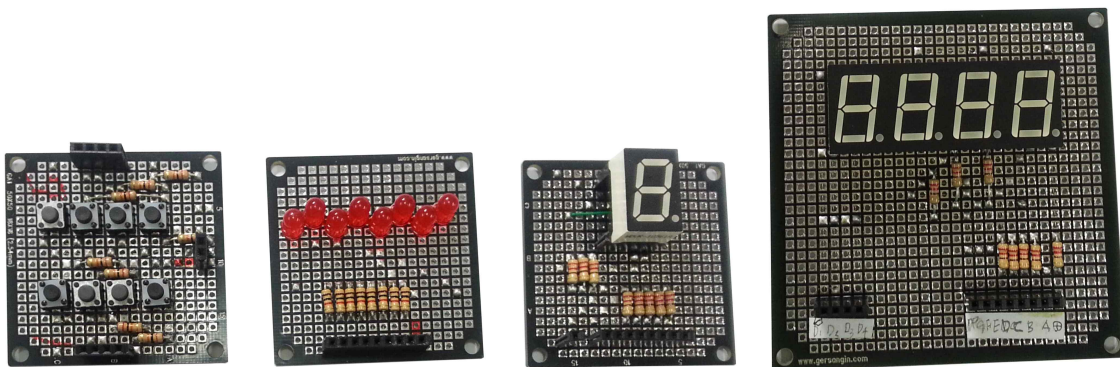


그림 12. 확장 모듈

확장 모듈들은 ATmega328의 입출력 핀에 직접 연결할 수도 있고 74595나 MCP23017 칩을

통해 연결할 수 있도록 핀 헤더가 보드에 마련되어 있다.



그림 13. DM-UStyle EX 학습 보드의 확장 모듈 연결용 핀 헤더

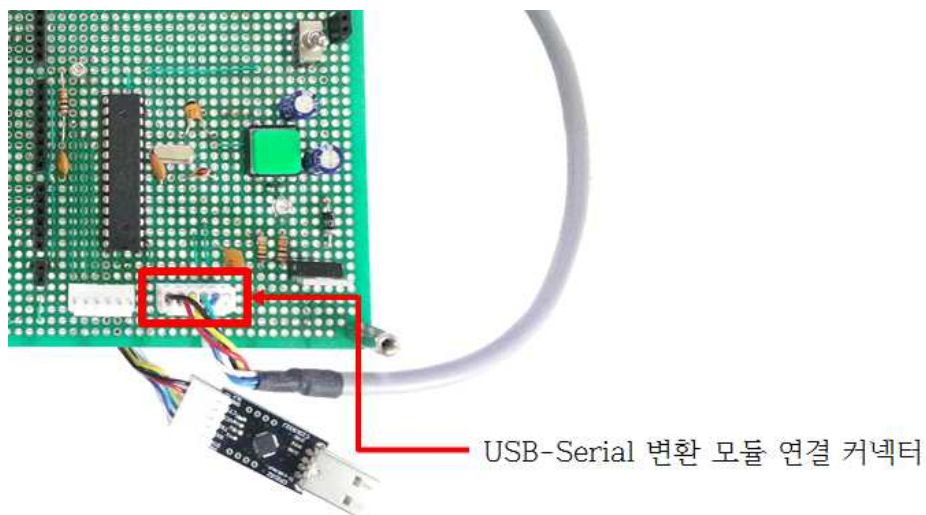
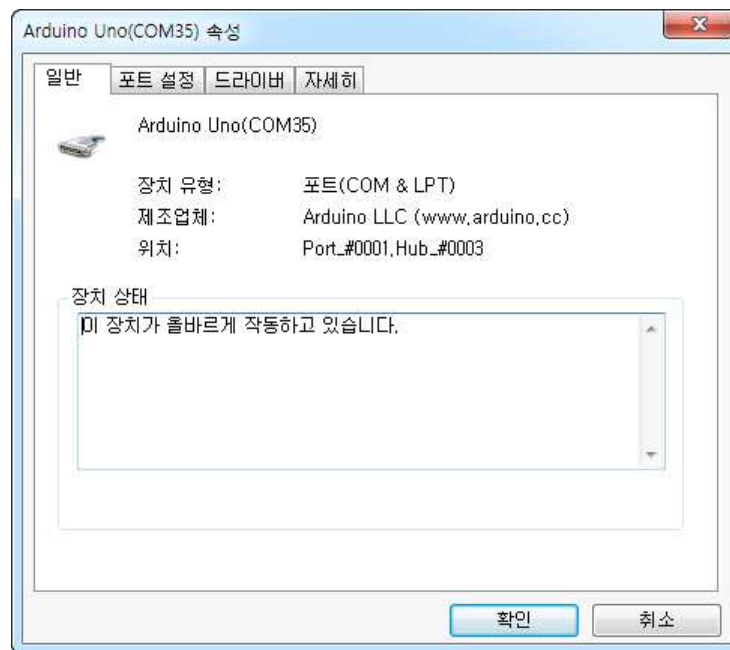


그림 14. DM-UStyle EX 학습 보드와 USB-Serial 변환 모듈 연결

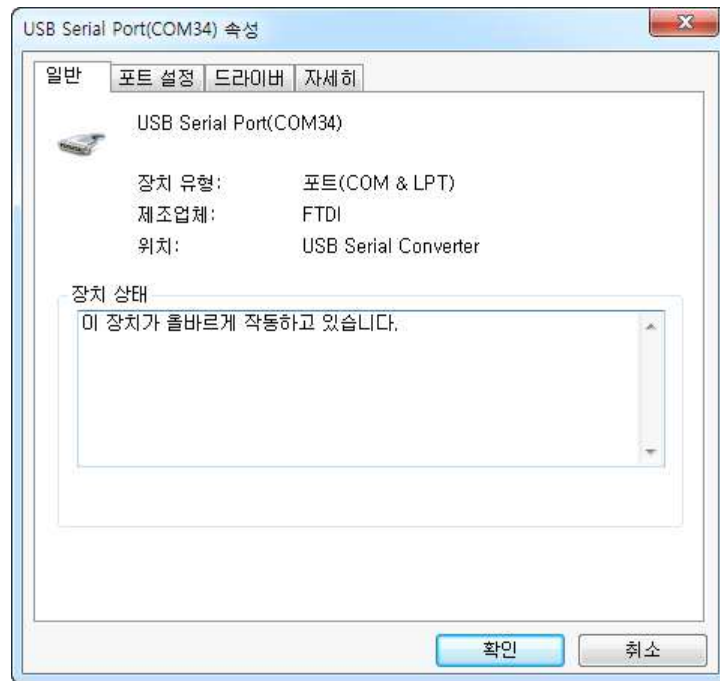
DM-UStyle EX 보드를 USB-Serial 변환 모듈을 통해 처음 컴퓨터와 연결하면 USB-Serial 변환 모듈을 위한 드라이버를 설치해야 한다. 아두이노 우노의 경우 USB-Serial 변환을 위해

Atmel사의 Mega17u2 마이크로컨트롤러를 사용하며 드라이버는 아두이노 소프트웨어와 함께 배포된다. 반면 DM-UStyle EX 보드를 위한 USB-Serial 변환 모듈은 FTDI 칩을 사용하므로 다른 드라이버를 설치하여야 한다.¹³⁾ 드라이버 설치 상태는 장치관리자에서 확인할 수 있다.



(a) 아두이노 우노

13) USB-Serial 변환 모듈을 위한 드라이버는 <http://www.newtc.co.kr>에서 다운받을 수 있다.



(b) DM-UStyle EX

그림 14. USB-Serial 변환을 위한 드라이버 설치¹⁴⁾

DM-UStyle EX 학습 보드는 ATmega328을 이용하여 만들어진 아두이노 우노 호환 보드이므로 드라이버가 설치된 후 아두이노 프로그램에서 ‘도구 → 보드’ 및 ‘도구 → 시리얼 포트’ 설정은 아두이노의 경우와 동일하며 이후 아두이노 우노와 같은 방법으로 사용할 수 있다.

소프트웨어 및 드라이버의 설치가 정상적으로 완료되었는지 확인하기 위해 아두이노 프로그램과 함께 설치된 예제를 실행시켜 보자. “파일 → 예제 → 01.Basics → Blink”를 선택하면 해당 스케치가 새로운 창에 열린다. Blink 예제는 아두이노 보드의 13번 핀에 연결된 LED를 1초 간격으로 점멸하는 프로그램으로 DM-UStyle EX 보드에도 13번 핀에 연결된 LED가 포함되어 있다.

Blink 코드의 창이 열리면 아두이노 프로그램 툴바의 가장 왼쪽에 있는 ‘확인’ 버튼을 눌러보자. ‘확인’ 버튼은 소스 코드에 문법 오류가 있는지 검사하고 오류가 없으면 바이너리 파일을 생성하는 역할을 한다. 오류가 발견되지 않았으면 ‘확인’ 버튼 바로 옆에 있는 ‘업로드’ 버튼을 눌러보자. ‘업로드’ 버튼은 문법 오류를 검사하여 바이너리 파일을 생성하고 이를 USB-Serial 연결을 통해 마이크로컨트롤러에 설치하여 실행시키는 역할을 한다. 즉, ‘업로드’에는 ‘확인’의 기능까지 포함되어 있다.

14) 아두이노 우노와 DM-UStyle EX 학습 보드를 컴퓨터와 연결하는 경우 할당되는 포트는 컴퓨터에서 임의로 결정된다. 아두이노 우노 보드 2개를 컴퓨터와 연결하는 경우 보드 별로 각각 별도의 포트가 할당되므로 그림 7과 그림 14-(a)에서 포트 번호가 다르게 나타나고 있다.



그림 15. 아두이노 프로그램의 툴바 버튼

업로드가 성공적으로 수행되면 DM-UStyle EX 보드의 LED가 1초 간격으로 점멸하는 것을 확인할 수 있다.

4. 아두이노 프로그램의 구조

아두이노는 C/C++을 기반으로 만들어졌으므로 시작점인 main 함수가 반드시 필요하다. 하지만 Blink 예제를 살펴보면 main 함수를 찾아볼 수 없고 setup과 loop의 두 함수만이 존재한다. 사실 아두이노에도 main 함수가 존재하지만 프로그램 작성을 최소화하고 직관적인 구조로 만들기 위해 main 함수를 숨겨놓고 있다. 반면 setup 함수와 loop 함수는 프로그램 실행을 위해 반드시 필요한 함수로 사용할 필요가 없는 경우에도 반드시 정의하여야 한다.

setup 함수는 스케치가 시작될 때 호출되는 함수로 이 함수에서는 변수의 초기화, 핀 모드 설정, 필요한 라이브러리 적재 등의 작업을 수행한다. setup 함수는 전원이 켜지거나 리셋 버튼이 눌러질 때 한 번만 호출된다. loop 함수는 그 이름에서 알 수 있듯이 스케치가 수행되는 동안 지속적으로 호출되는 함수로 스케치가 수행됨에 따라 프로그램의 상태를 바꾸거나 다양한 입력에 반응하여 특정 작업을 수행할 수 있도록 해준다. 그림 16은 아두이노 스케치의 전형적인 처리 순서를 나타낸 것이다.

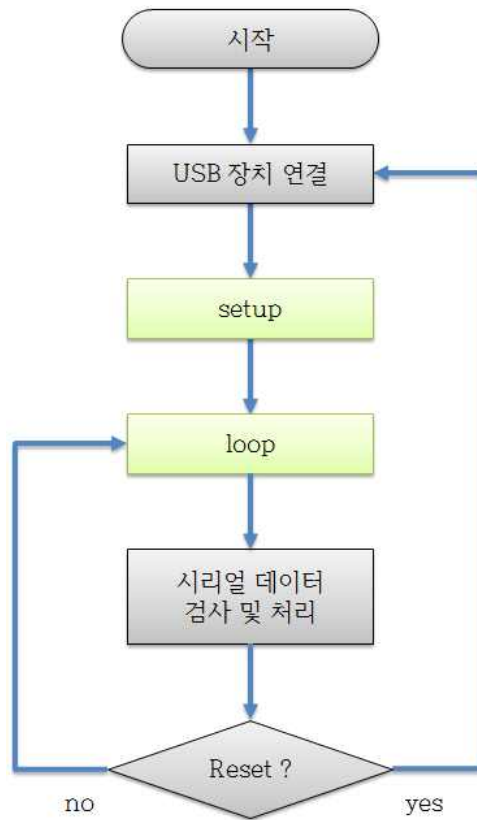


그림 16. 아두이노 스케치의 전형적인 처리 흐름도

마이크로컨트롤러는 기본적으로 디지털 데이터를 입력 받고 이를 처리하여 디지털 데이터를 출력한다. 먼저 코드 1을 입력하고 'MyBlink'라는 이름으로 저장하자. MyBlink 프로그램은 Blink 예제에서 주석을 제거하고 코드만을 모아 놓은 것으로 내용은 Blink와 동일하다. 아두이노의 디지털 핀은 입력 또는 출력으로 사용할 수 있지만 입출력으로 동시에 사용할 수는 없다. 따라서 데이터를 입출력하기 이전에 반드시 입력과 출력 중 어느 용도로 사용할 것인지 명시하여야 하며 주로 setup 함수에서 이루어진다. LED에 불을 켜는 것은 LED로 데이터를 출력하는 동작이므로 코드 1에서는 출력으로 설정하였다. 데이터 출력은 digitalWrite 함수를 통해 이루어지며, 디지털 데이터 0이나 1의 값을 출력하기 위해 미리 정의된 상수 LOW나 HIGH를 사용할 수 있다. 데이터 출력은 반복적으로 이루어지므로 loop 함수에서 이루어지고 있다. 마지막 delay 함수는 밀리초 단위의 시간 지연을 위해 사용되었다.

코드 1. MyBlink

```
int led = 13;           // 13번 핀에 LED 연결
```

```

void setup() {
  pinMode(led, OUTPUT);      // 13번 핀을 출력 모드로 설정
}

void loop() {
  digitalWrite(led, HIGH);    // 13번 핀으로 HIGH (1 또는 ON) 출력
  delay(1000);                // 1초 (1000ms) 대기
  digitalWrite(led, LOW);     // 13번 핀으로 LOW (0 또는 OFF) 출력
  delay(1000);                // 1초 (1000ms) 대기
}

```

툴바의 ‘확인’ 버튼 또는 단축키 ‘Ctrl + R’을 눌러 코드에 오류가 없는 지 확인하고, 툐바의 ‘업로드’ 버튼 또는 단축키 ‘Ctrl + U’를 눌러 프로그램을 업로드 해보자. LED가 일초 간격으로 깜박거리는가?

5. 아두이노의 한계

아두이노는 마이크로컨트롤러를 사용하는 제어 장치를 쉽고 간단하게 만들 수 있도록 해주며 다양한 응용이 가능하다. 하지만 그 한계도 분명 존재하므로 아두이노를 이용하여 또는 보다 일반적으로 마이크로컨트롤러를 이용하여 할 수 있는 일과 할 수 없는 일을 구분하는 것이 중요하다. 아두이노의 한계는 대부분 사용한 마이크로컨트롤러에서 온다. 아두이노 우노를 이 글을 쓰고 있는 컴퓨터를 비교해 보면 메모리는 약 26만분의 1, 속도는 약 212분의 1에 지나지 않는다.

항목	아두이노 우노	데스크톱 컴퓨터
CPU	ATmega328	Intel Core i7
프로그램 저장 메모리	32 KByte	8 GByte
클럭	16 MHz	3.4 GHz

표 2. 아두이노 우노와 데스크톱 컴퓨터 비교

마이크로컨트롤러가 싱글 칩 컴퓨터(single chip computer)라고 불리기는 하지만 일반적으로

생각하는 컴퓨터에 비해 느리고 아주 작은 크기의 프로그램만을 실행할 수 있다. 애초 마이크로컨트롤러의 설계 목적이 데스크톱 컴퓨터와는 다른 것이다. 따라서 연산량이 많은 경우, 빠른 연산을 필요로 하는 경우, 고속의 데이터 전송을 요하는 경우 등에서 아두이노는 적합하지 않다. 예를 들어, 영상의 실시간 전송이나 분석, 음성 인식, 모니터링을 위한 웹 서버 구축 등은 확장 보드를 사용하여 구현이 가능하기는 하지만 이에 필요한 확장 보드의 가격은 아두이노 보드보다 비싼 것이 사실이다. 따라서 이러한 응용 분야에서는 마이크로컨트롤러가 아닌 라즈베리 파이(Raspberry Pi)¹⁵⁾를 포함한 싱글 보드 컴퓨터를 사용하는 것이 더 효율적이다. 아두이노는 간단한 제어 장치를 만드는데 적합하며 아두이노가 “예술가, 디자이너 및 학생들이 쉽게 사용할 수 있는 저렴한 컨트롤 장치를 만들기 위해” 시작되었다는 점을 잊지 말아야 한다.

15) <http://www.raspberrypi.org>
