

Cyber Physical System

전북대학교
이성현

목차

- I. Arduino
- II. HW 조립 메뉴얼

I. Arduino

HW 제원

- ▶ 아두이노
 - ▶ 단일 보드 마이크로컨트롤러 (=PC와 유사)
 - ▶ 통합 개발환경을 제공
 - ▶ 2005년 이탈리아에서 시작
 - ▶ 쉬움, 오픈 하드웨어
 - ▶ 방대한 참고자료, 라이브러리

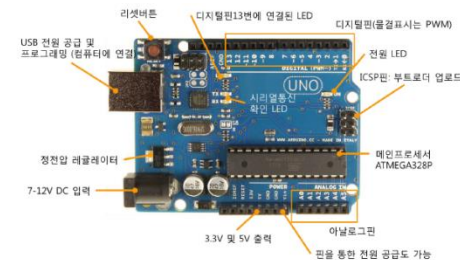


Fig 2 우노 R3의 기본 부품 및 핀 정보

HW 제원

- ▶ 아두이노 UNO
 - ▶ Arduino board 중 가장 범용적으로 쓰임

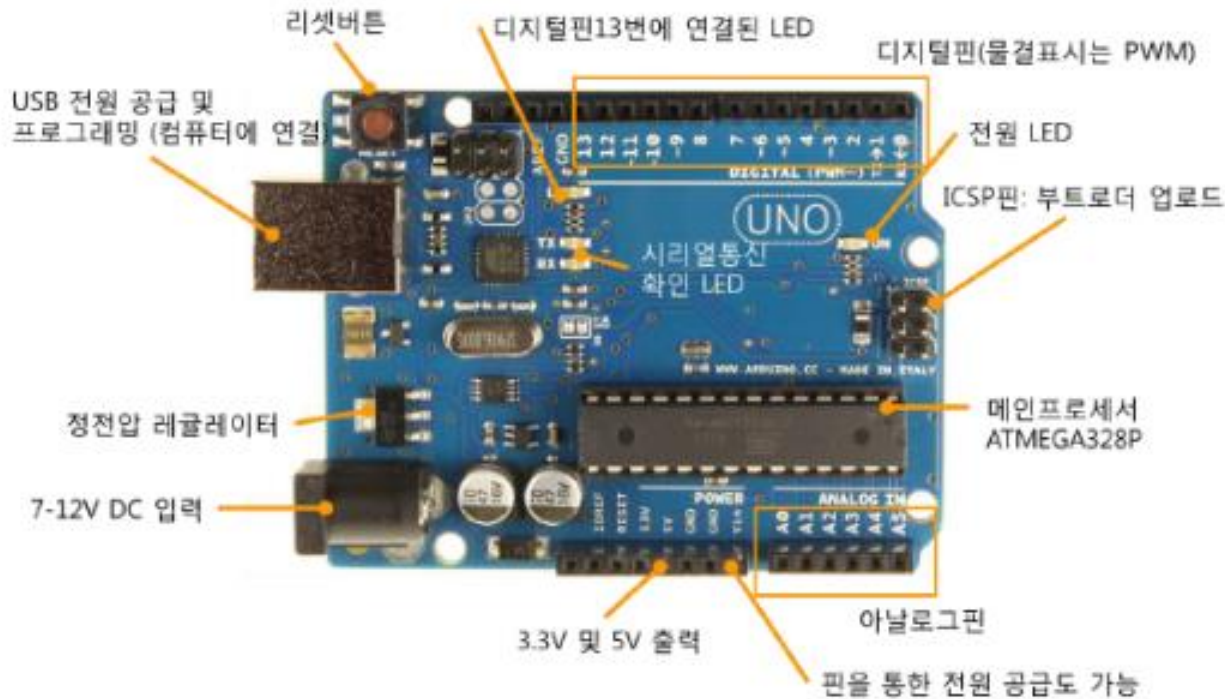
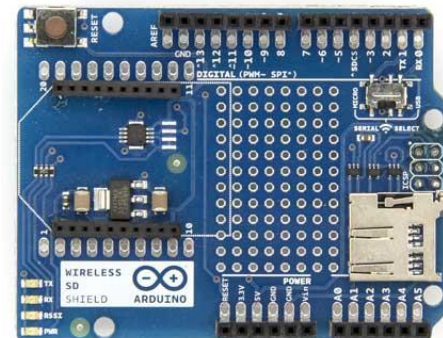
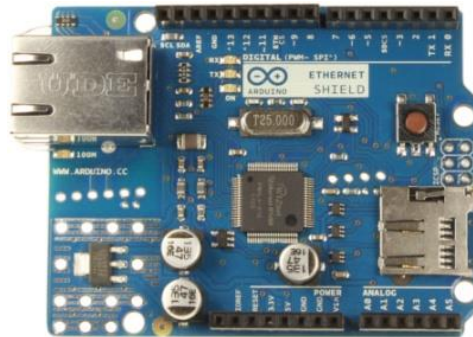
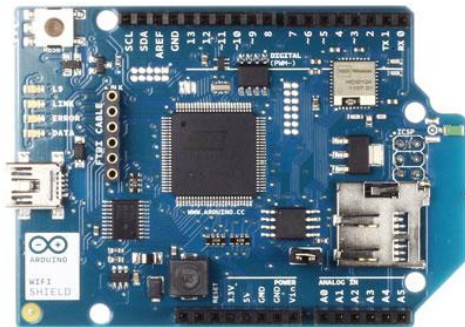


Fig. 2 우노 R3의 기본 부품 및 핀 정보

HW 제원

▶ 아두이노 쉴드

- ▶ 아두이노 UNO 보드 혹은 기타 보드 위에 층층이 쌓아서 사용 가능한 기능 확장용 보드들을 지칭 (예. WiFi 쉴드, Ethernet 쉴드)



CPS

- ▶ 실습을 위한 준비
- ▶ 1. 아두이노와 각종 장치들(LED, 모터, 센서...)을 연결
- 2. 아두이노 개발환경(IDE)을 실행
- 3. 소스코드(스케치, sketch) 작성
- 4. 컴파일 및 업로드
- 5. 아두이노 동작 확인
- 6. 디버깅, 3~5 반복

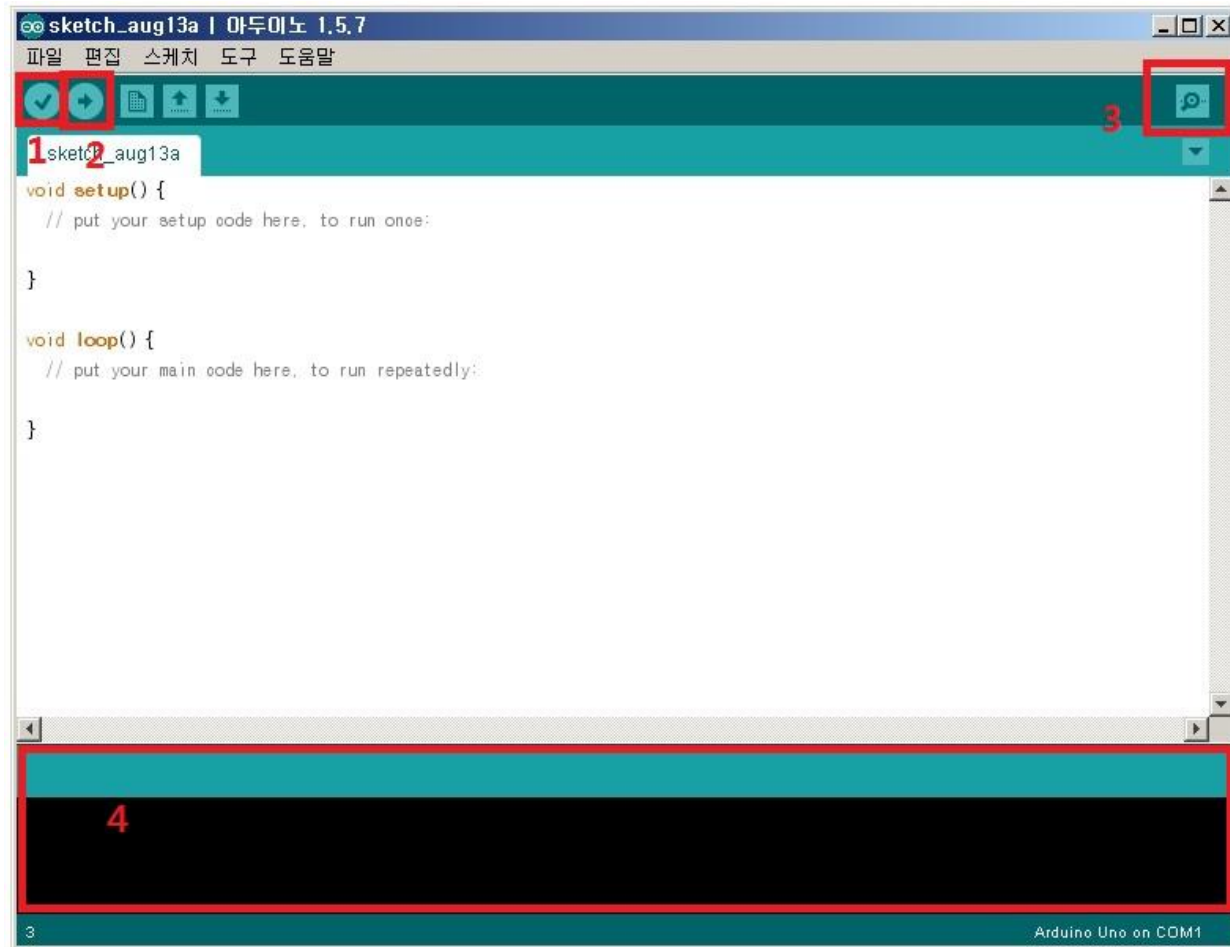
- ▶ 아두이노 테스트를 위한 준비물
- 아두이노 UNO 보드 (혹은 기타 보드)
- Starter Kit (LED, 스위치, 각종 센서, 저항, 케이블 등)
- 아두이노 개발환경 (IDE)
- 인두기

아두이노 IDE

- ▶ 아두이노 개발환경
 - ▶ - 아두이노 소스 작성
 - ▶ - 컴파일, 업로드
 - ▶ - 디버깅 (Serial monitor)
 - ▶ - 배포 : <https://www.arduino.cc/en/Main/Software>

아두이노 개발환경 기본 구조

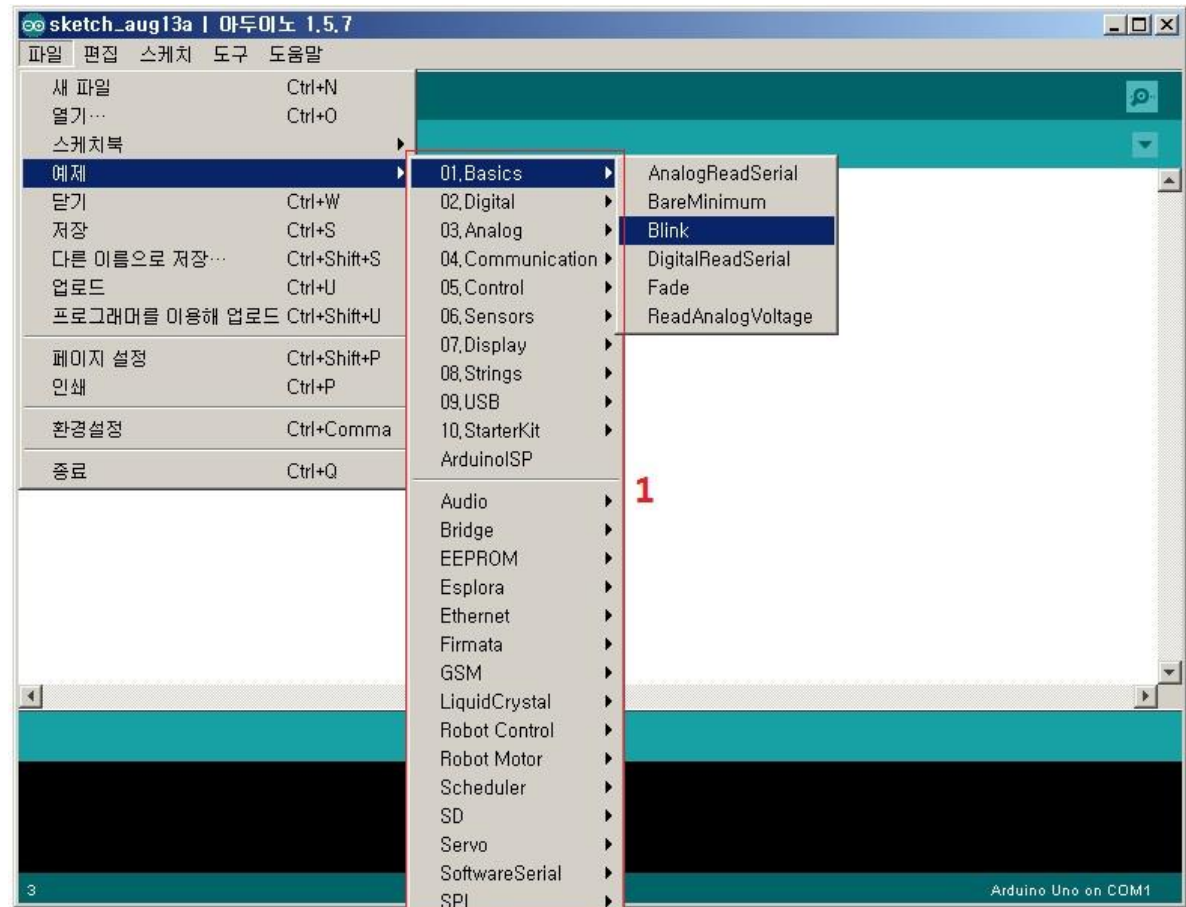
- ▶ 메뉴
- ▶ 컴파일, 업로드 버튼
- ▶ 시리얼 모니터 버튼
- ▶ 소스수정영역
- ▶ 메시지 영역



아두이노 개발환경 기본 구조

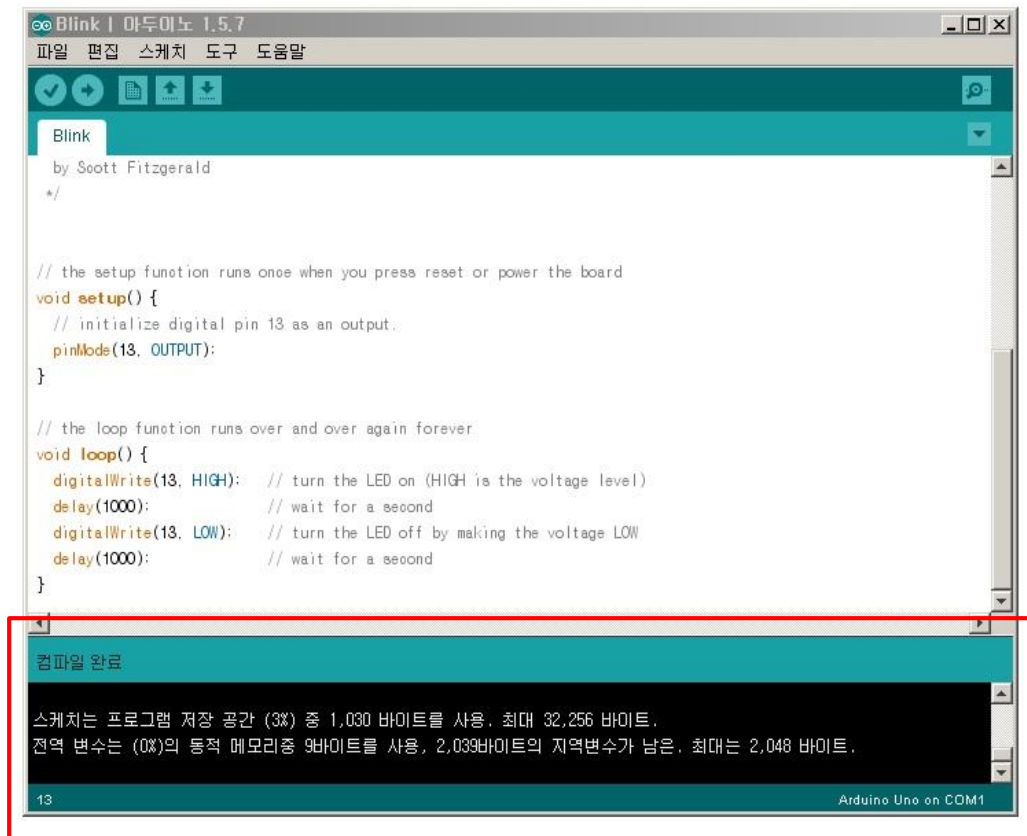
▶ Blink 예제 불러오기

- ▶ 메뉴
- ▶ 파일
- ▶ 예제
- ▶ Basics
- ▶ Blink



아두이노 개발환경 기본 구조

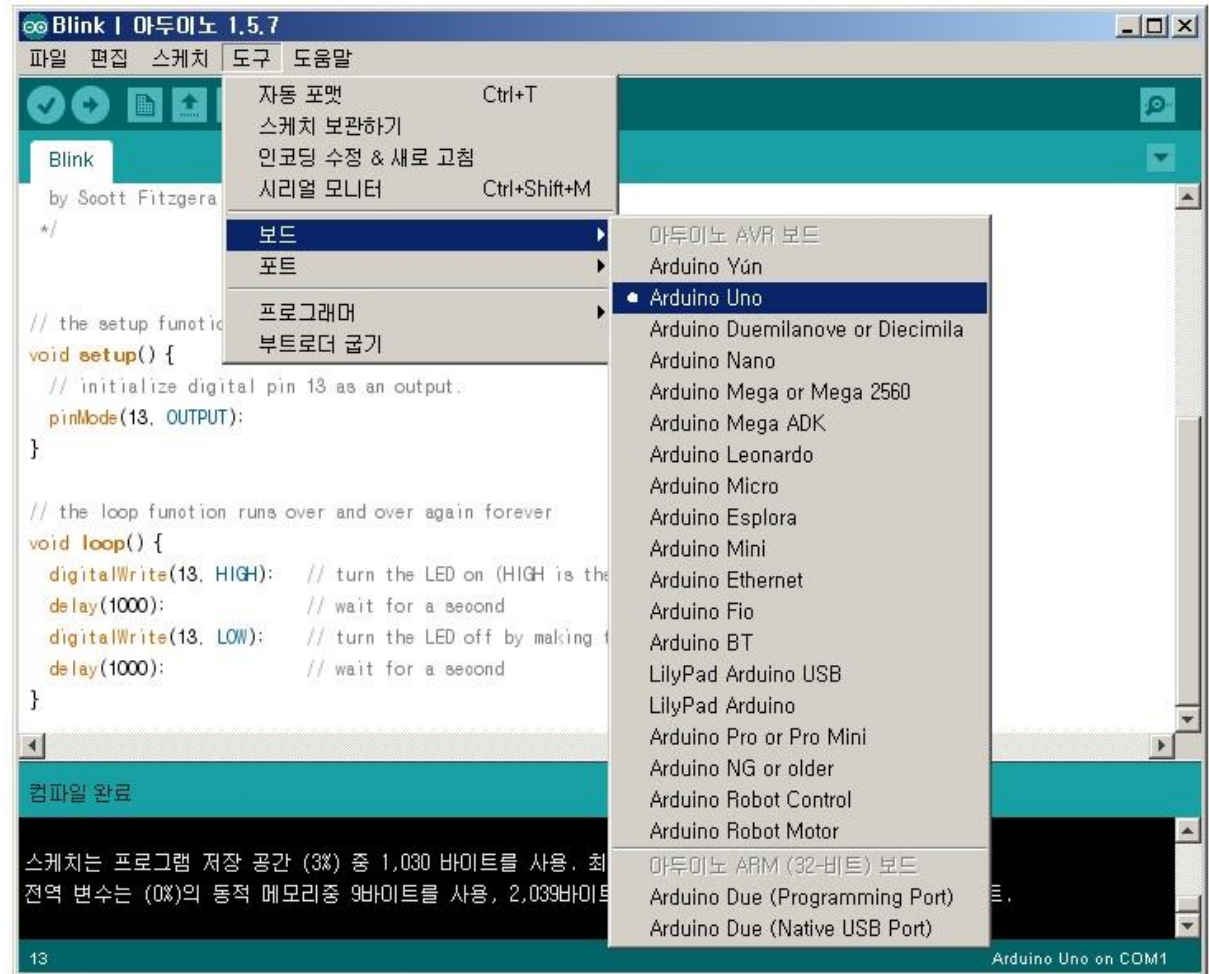
- ▶ 컴파일
- ▶ 정상 컴파일되면 아두이노를 USB 케이블로 연결



아두이노 개발환경 기본 구조

▶ 보드 선택

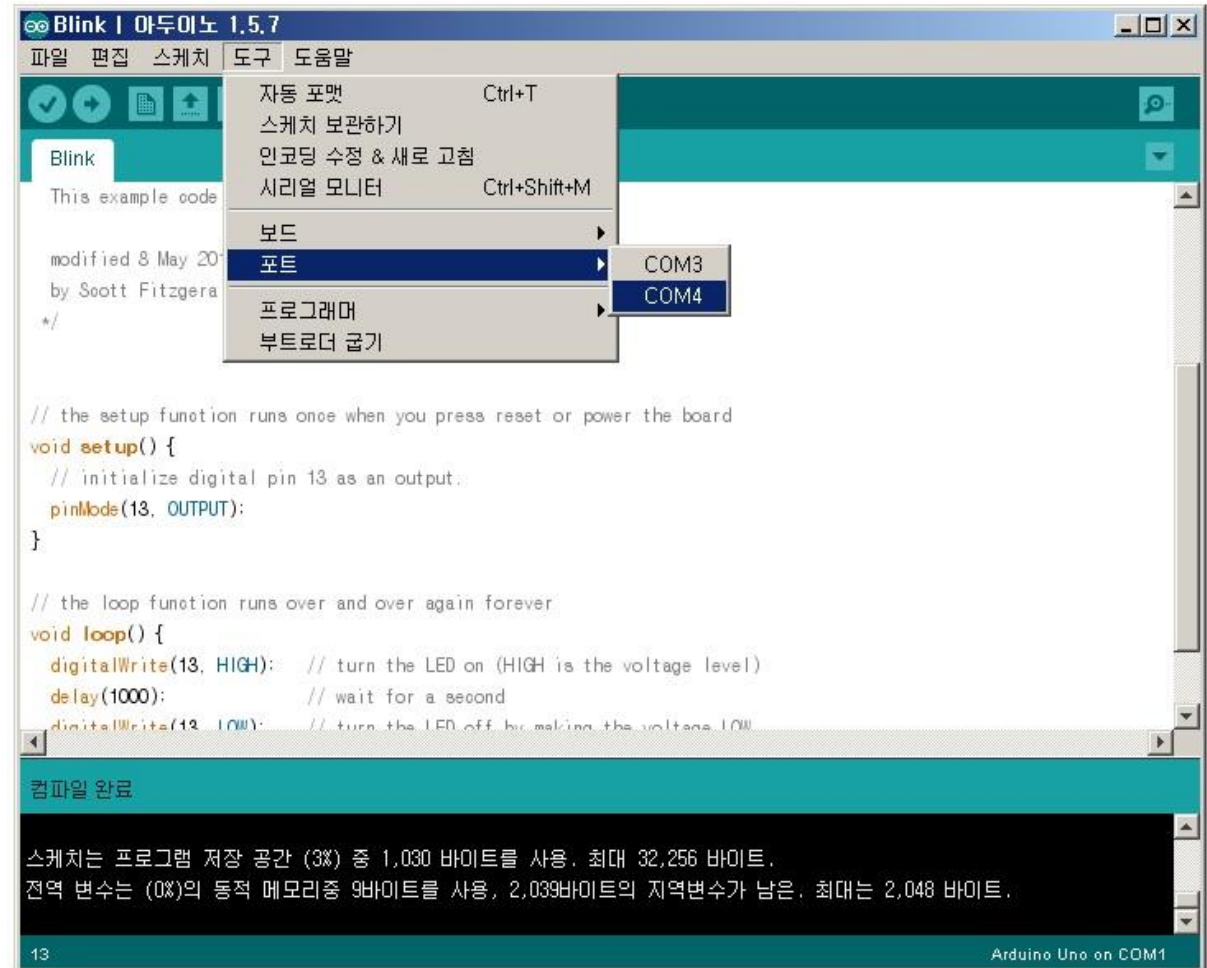
- ▶ 도구
- ▶ 보드
- ▶ 연결된 보드 선택



아두이노 개발환경 기본 구조

▶ COM 포트 선택

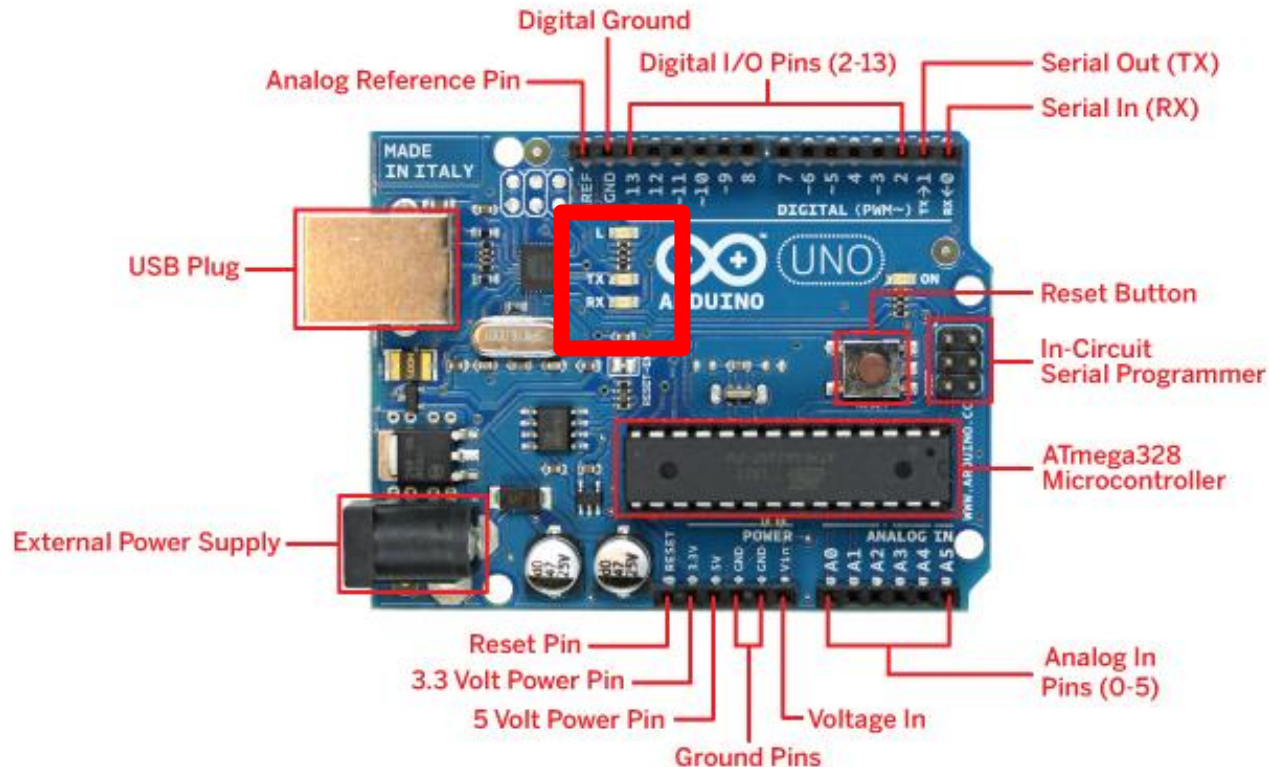
- ▶ 도구
- ▶ 포트
- ▶ 연결된 포트 선택



아두이노 개발환경 기본 구조

▶ 업로드

- ▶ 업로드 버튼 클릭후 보드 중앙의 TX/RX LED가 깜빡이고 메시지 영역에 "업로드 완료" 메시지가 출력되면 정상



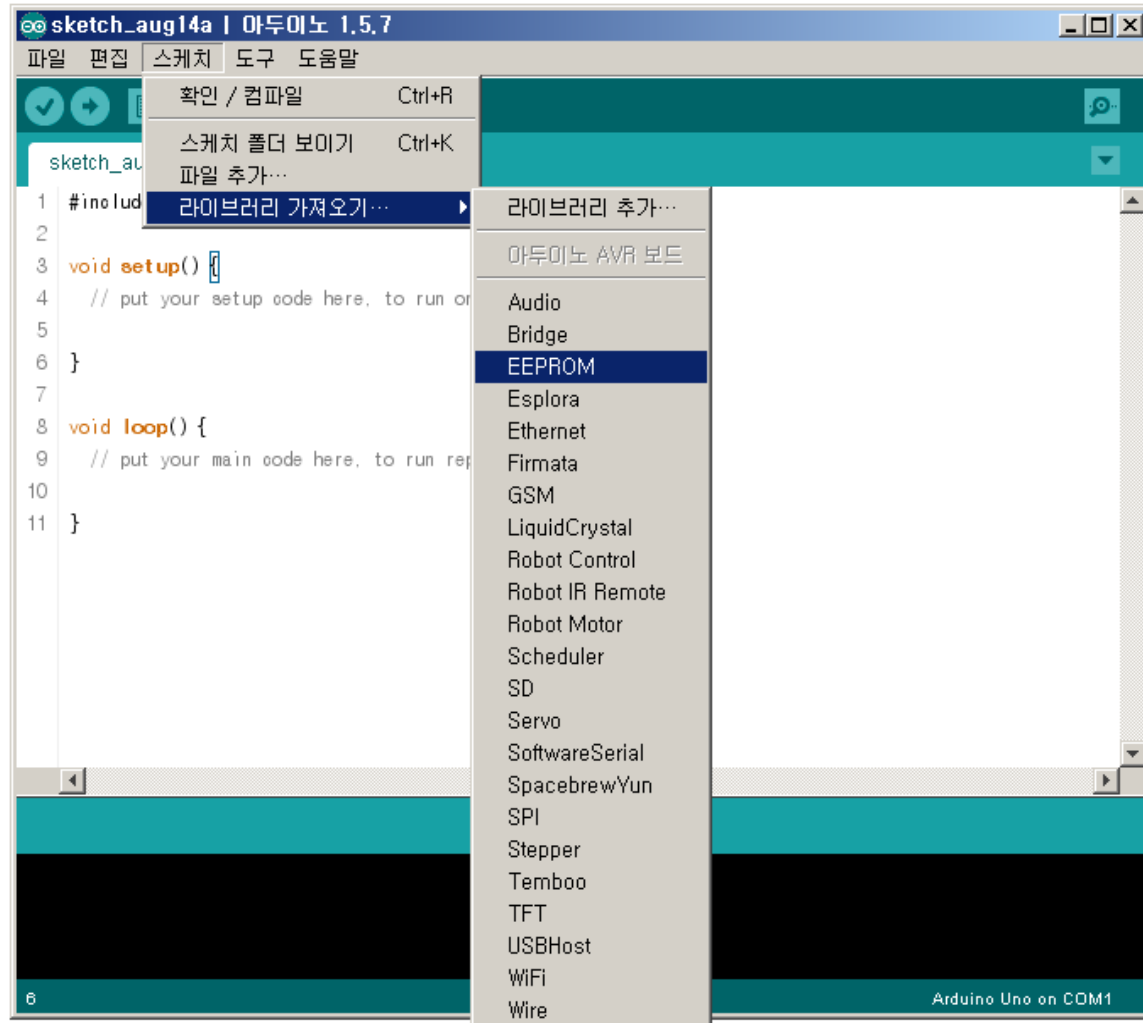
아두이노 라이브러리

- ▶ 아두이노 라이브러리
 - ▶ - 다양한 센서, 모터, 모듈을 쉽게 사용할 수 있도록 지원
 - ▶ - 일반적으로 ZIP 압축 파일로 배포
 - ▶ - 헤더 파일 include 한 뒤, 초기화 해서 사용
 - ▶ - ex.
 - ▶ `#include <SoftwareSerial.h>`
 - ▶ `SoftwareSerial mySerial(2, 3) // Rx, Tx pin`
- ▶ 라이브러리 폴더
 - ▶ - 외부 라이브러리 : [C:\Program Files\Arduino\libraries]
 - ▶ - 내부 라이브러리 (기본 라이브러리) :
 - ▶ [C:\Program Files\Arduino\hardware\arduino\avr\libraries]

아두이노 라이브러리

- ▶ 스케치
- ▶ 라이브러리 가져오기...
- ▶ 원하는 라이브러리 선택
- ▶ 소스 코드에서 확인

#include <EEPROM.h>

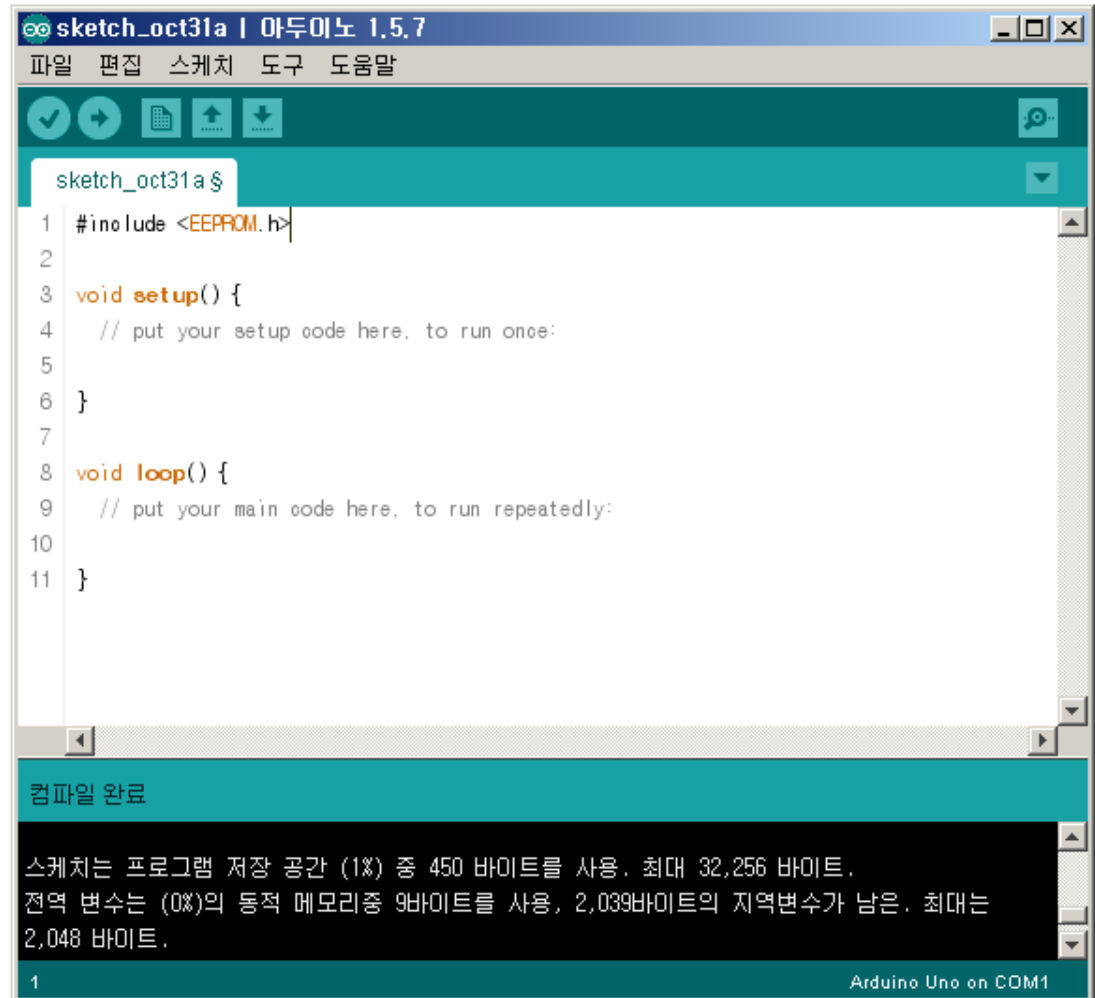


아두이노 라이브러리

- ▶ Include 구문을 직접 입력

`#include <EEPROM.h>`

- ▶ 컴파일해서 확인



```
sketch_oct31a | 아두이노 1.5.7
파일 편집 스케치 도구 도움말

sketch_oct31a $
1 #include <EEPROM.h>
2
3 void setup() {
4   // put your setup code here, to run once:
5
6 }
7
8 void loop() {
9   // put your main code here, to run repeatedly:
10
11 }

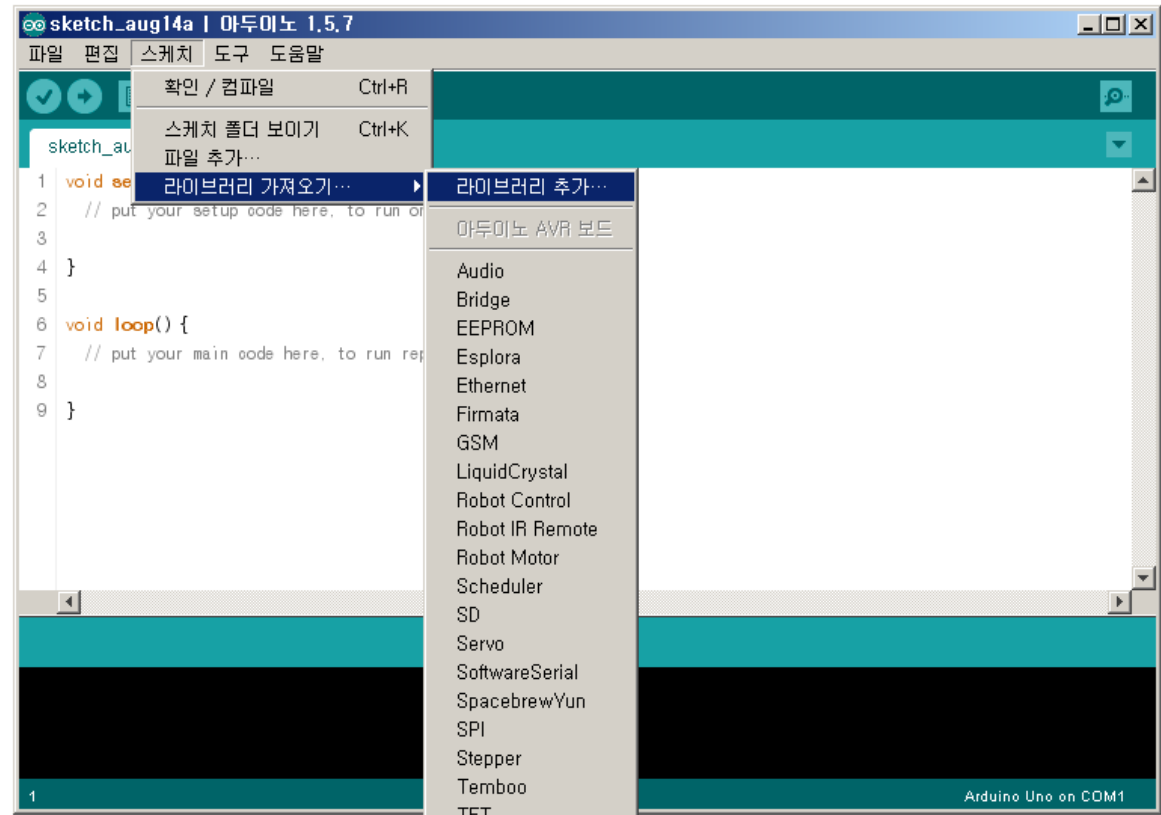
컴파일 완료

스케치는 프로그램 저장 공간 (1%) 중 450 바이트를 사용, 최대 32,256 바이트.
전역 변수는 (0%)의 동적 메모리 중 9바이트를 사용, 2,039바이트의 지역변수가 남은, 최대는
2,048 바이트.

1 Arduino Uno on COM1
```

아두이노 라이브러리

- ▶ 스케치
- ▶ 라이브러리 가져오기...
- ▶ 라이브러리 추가...
- ▶ 다운로드 받은
- ▶ 아두이노 IDE 재실행
- ▶ 메뉴에서 확인



아두이노 라이브러리

- ▶ 설치 폴더로 이동
- ▶ - [C:\Program Files\Arduino\libraries]
- ▶ - ZIP 파일 압축 해제
- ▶ - 아두이노 IDE 재실행
- ▶ - 메뉴에서 설치여부 확인
- ▶ - 헤더 파일 include 한 뒤, 초기화 해서 사용
- ▶ - ex. ArduinoTest 라이브러리
 - ▶ libraries\ArduinoTest (폴더)
 - ▶ libraries\ArduinoTest\ArduinoTest.cpp
 - ▶ libraries\ArduinoTest\ArduinoTest.h
 - ▶ libraries\ArduinoTest\examples (폴더)
- ▶ - 사용 예
 - ▶ #include <ArduinoTest.h>

아두이노 시리얼 모니터

- ▶ USB Serial 통신
 - ▶ - USB를 이용해서 PC와 아두이노 통신
 - ▶ - Serial 통신을 사용 (Hardware serial)
 - ▶ - Serial 통신은 2개의 데이터 통신용 핀을 사용 (Tx, Rx)
 - ▶ - 아두이노 UNO의 경우 digital 0번, 1번 핀이 이 용도로 예약되어 있음
- ▶ Serial Monitor
 - ▶ - 아두이노와 데이터를 주고 받기 위해 사용하는 툴
 - ▶ - 주로 아두이노 디버깅, 데이터 입력을 위해 사용
 - ▶ - Serial monitor를 이용한 디버깅 메시지를 보는 것이 거의 유일한 디버깅 방법이므로 중요

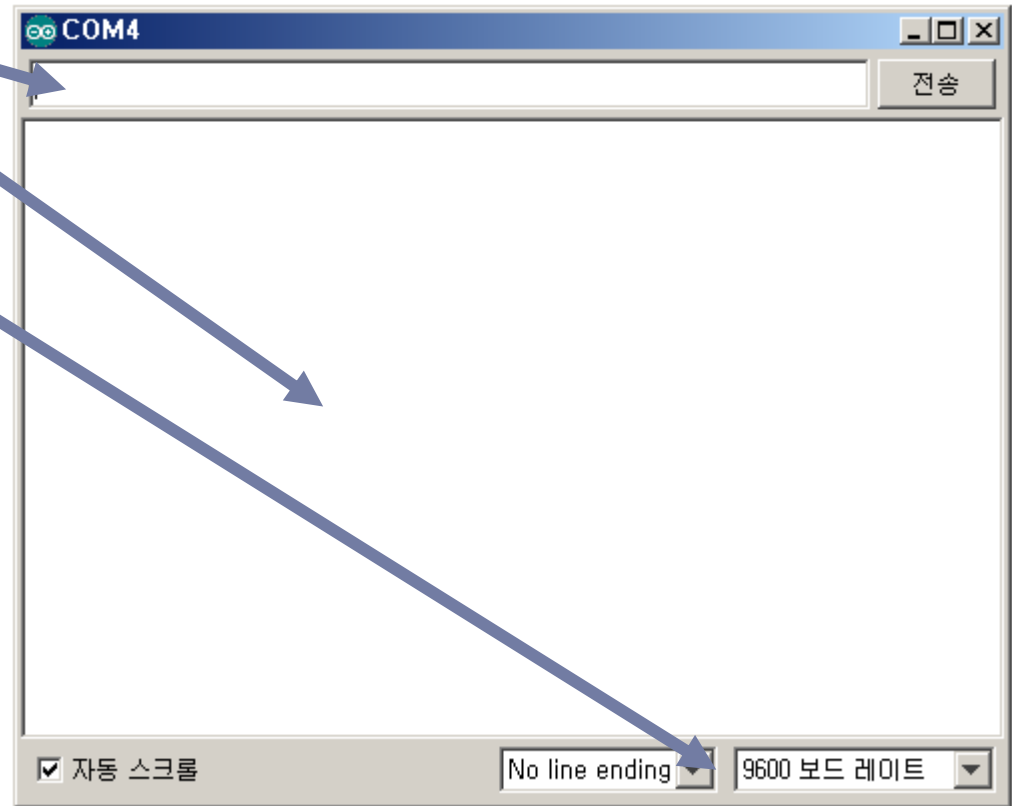
아두이노 시리얼 모니터

- ▶ 아두이노 연결된 상태
- ▶ 보드 종류, COM 포트 선택
- ▶ 도구 > 시리얼 모니터



아두이노 시리얼 모니터

- ▶ 메시지 입력창
- ▶ 수신 메시지 표시창
- ▶ 통신 속도 설정
 - ▶ Baud Rate
 - ▶ 기본 9600 bps



아두이노 시리얼 모니터

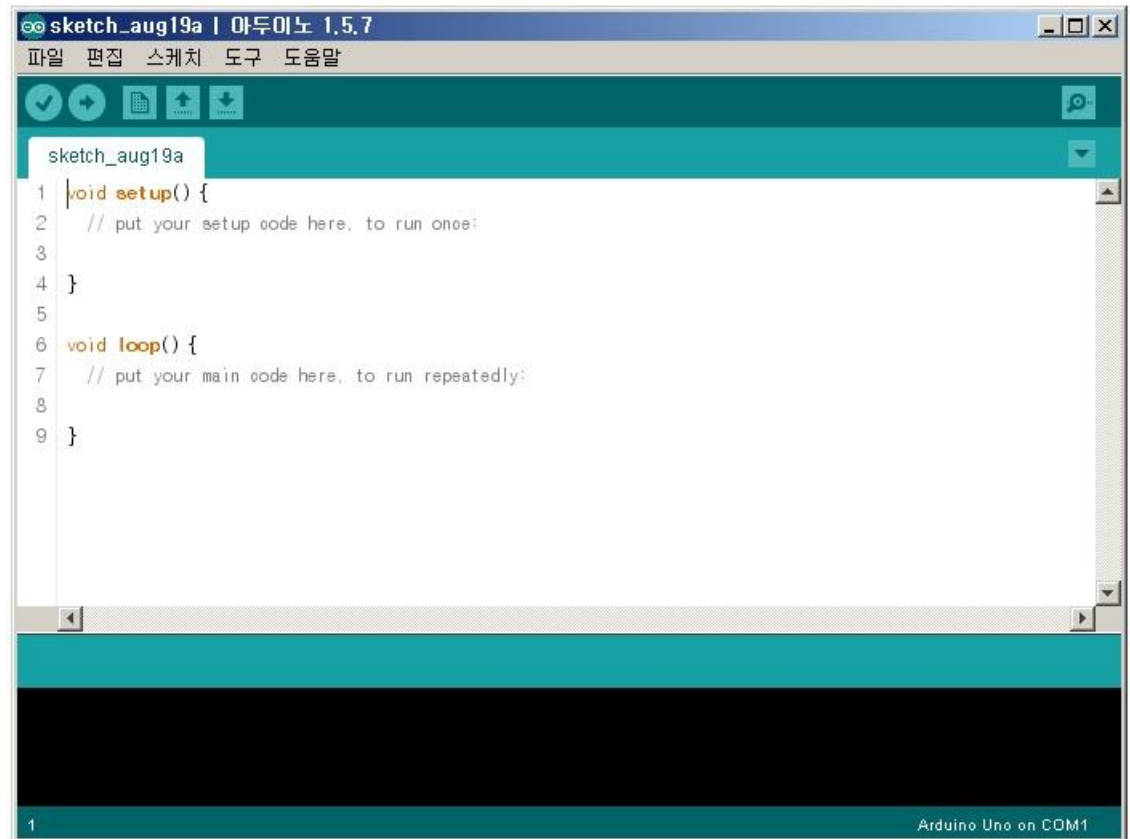
▶ Serial Monitor 출력 방법

- ▶ 초기화
- ▶ 메시지 출력

```
void setup() {  
    // initialize digital pin 13 as an output.  
    pinMode(13, OUTPUT);  
    // initialize serial communication at 9600 bits per second:  
    Serial.begin(9600);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
    delay(1000);           // wait for a second  
    digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW  
    delay(1000);           // wait for a second  
    Serial.println("LED on, off");  
}
```

아두이노 소스 기본 구조

- ▶ 초기화 루틴
 - ▶ setup() 함수
 - ▶ 최초 1회만 실행
- ▶ 반복 루틴
 - ▶ loop() 함수
 - ▶ setup 이후 무한반복



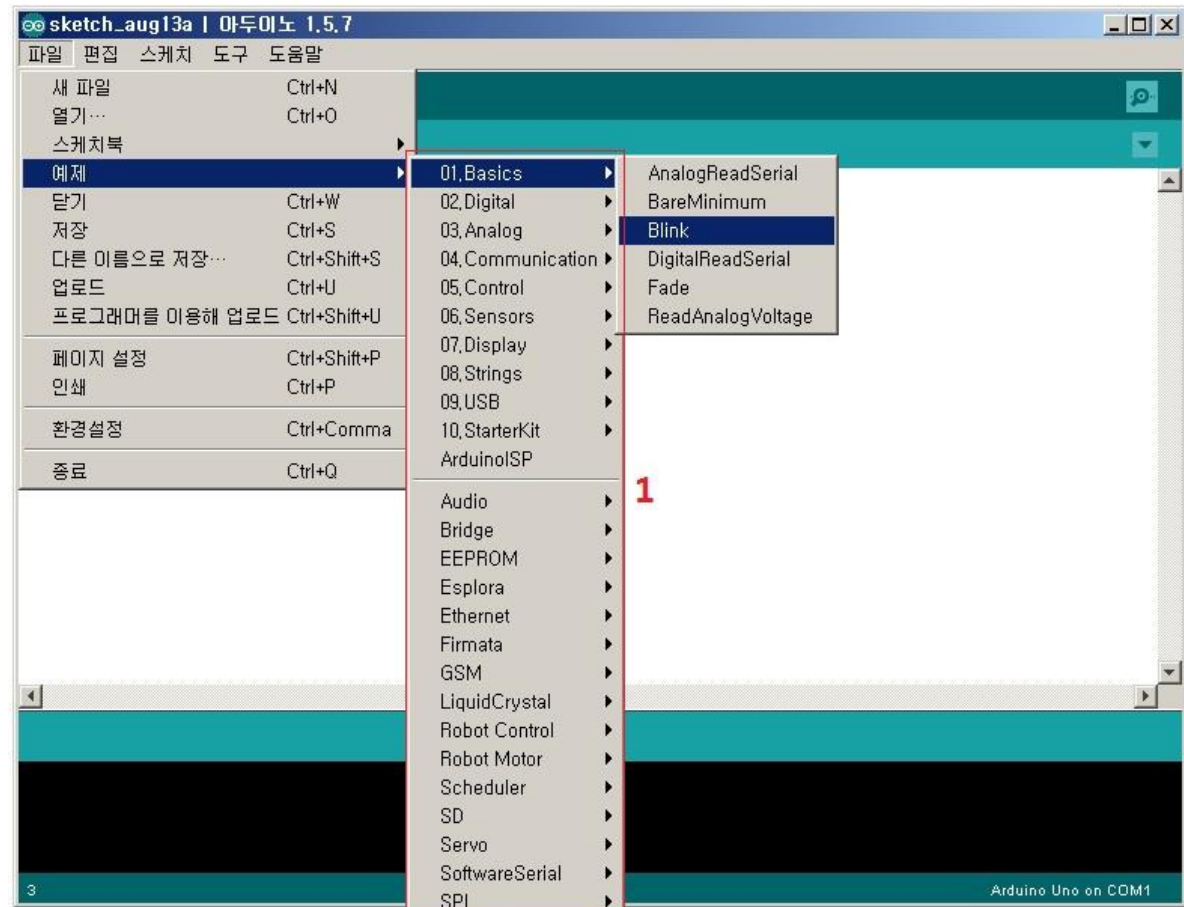
Blink 예제 테스트

- ▶ Blink 예제
 - ▶ - 아두이노에 기본으로 내장된 LED를 제어한다
 - ▶ - 아두이노에 내장된 LED는 내부적으로 D13 핀에 연결되어 있음
- ▶ 목표
 - ▶ - 아두이노에 기본으로 내장된 LED를 1초간 on / 1초간 off 반복

Blink 예제 테스트

▶ Blink 예제 불러오기

- ▶ 메뉴
- ▶ 파일
- ▶ 예제
- ▶ Basics
- ▶ Blink



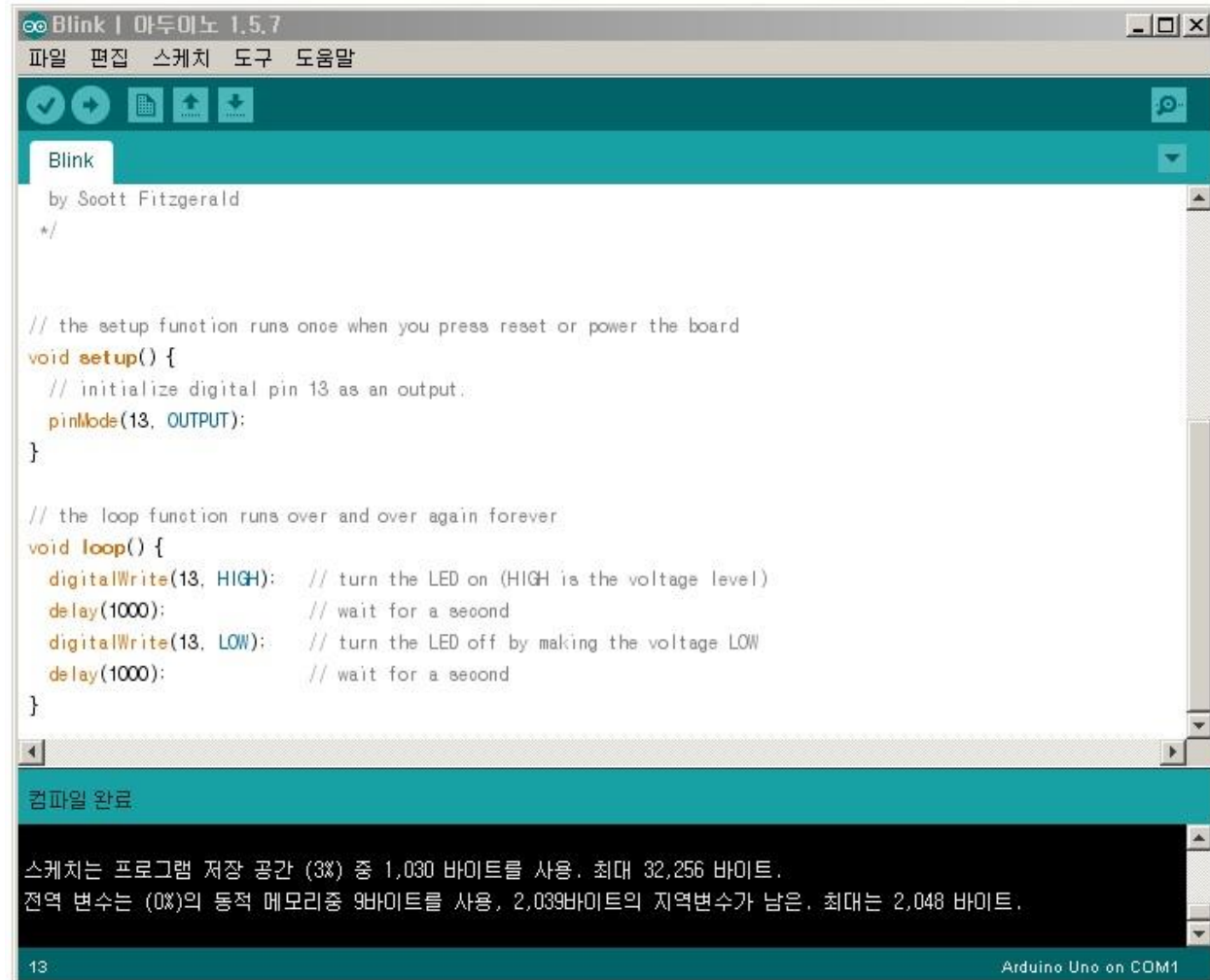
Blink 예제 테스트

▶ Setup

- ▶ 13번 핀 초기화
- ▶ pinMode 함수

▶ Loop

- ▶ digitalWrite 함수
- ▶ delay 함수



```
Blink | 마두이노 1.5.7
파일 편집 스케치 도구 도움말

Blink
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

컴파일 완료

스케치는 프로그램 저장 공간 (3%) 중 1,030 바이트를 사용, 최대 32,256 바이트.
전역 변수는 (0%)의 동적 메모리 중 9바이트를 사용, 2,039바이트의 지역변수가 남음, 최대는 2,048 바이트.

13 Arduino Uno on COM1
```

HW 조립

- ▶ 자동차 조립 설명서

- ▶ <http://www.gameplusedu.com/pds/gpshop/html/4wd%20ebook/index.html#p=1>

- ▶ 기계팔 조립 설명서

- ▶ <https://www.sinoning.cc/single-post/2017/05/25/SNARM-4dof-acrylic-parts-install-guide-A>
 - ▶ <https://www.sinoning.cc/single-post/2017/05/25/SNARM-4dof-acrylic-parts-install-guide-B>
 - ▶ <https://www.sinoning.cc/single-post/2017/05/25/4DOF-Upgrade-Robot-Arm-acrylic-parts-install-guide-C>

조립 및 실습 시 참고사항

- ▶ 가급적이면 조립 전 브래드보드에 센서 구동 예제부터 실습해볼 것
 - ▶ 조립 전 부품고장 여부확인 및 센서 구동 원리 파악
 - ▶ <http://www.gameplusedu.com/pds/gpshop/html/4wd%20ebook/index.html#p=1>
- ▶ 자동차
 - ▶ 초음파 센서
 - ▶ 색깔 감지 센서
 - ▶ DC 모터
 - ▶ 서보모터
 - ▶ 블루투스
- ▶ 기계팔
 - ▶ 서보모터
 - ▶ 블루투스