인하공업전문대학
INHA TECHNICAL COLLEGE

사물인터넷
2주차

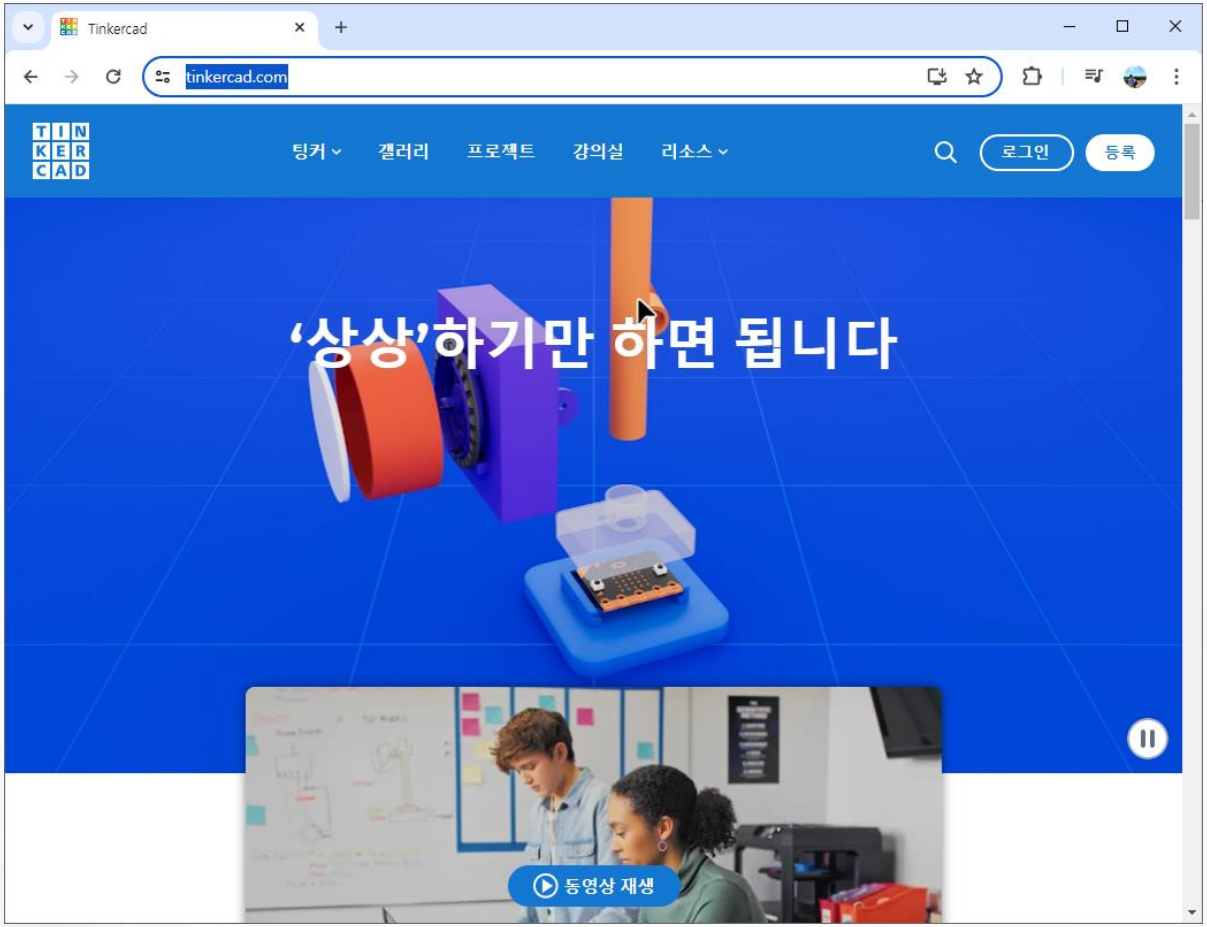
인하공업전문대학 컴퓨터 정보과

김한결

사물인터넷 - 실습

❖ 온라인 실습 사이트

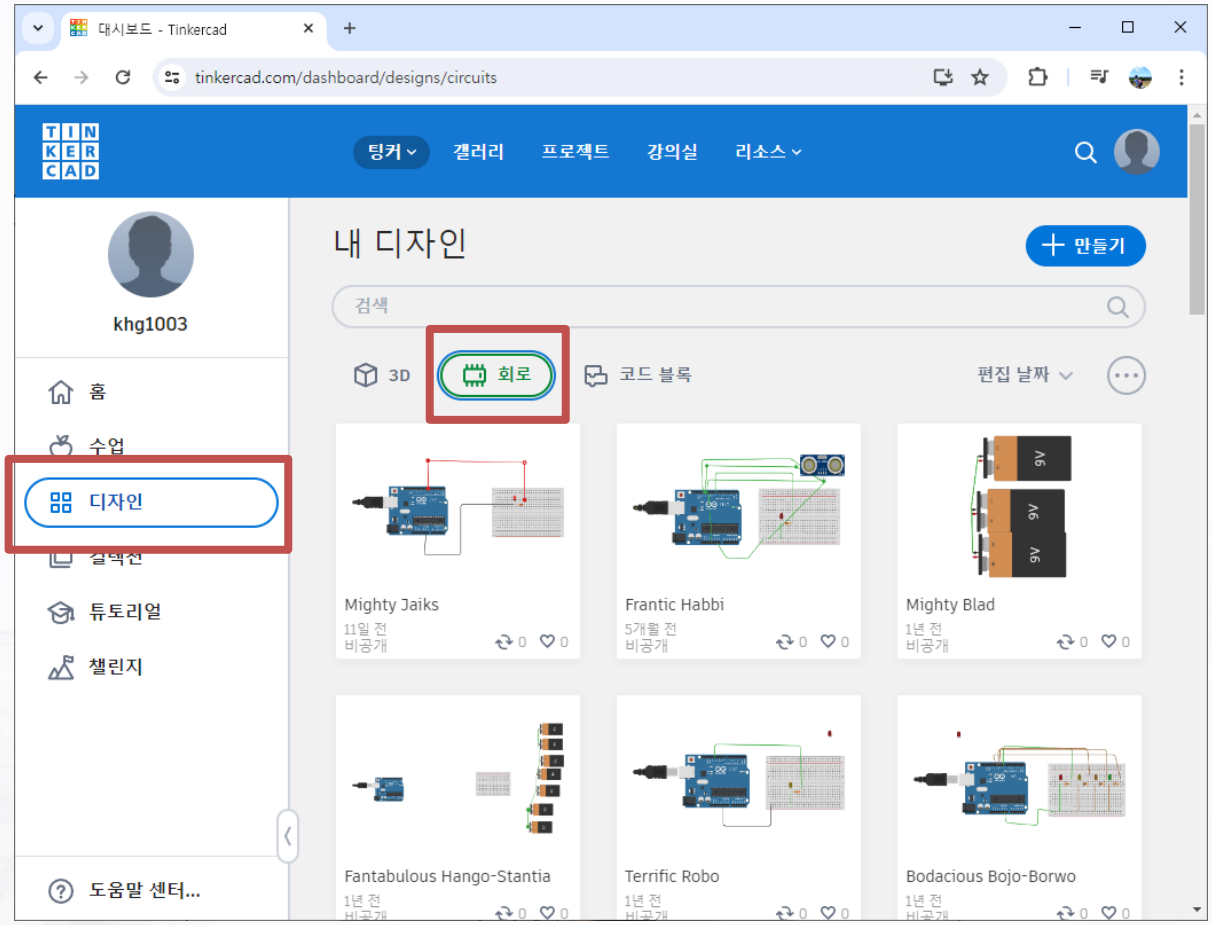
- https://www.tinkercad.com/



사물인터넷 - 실습

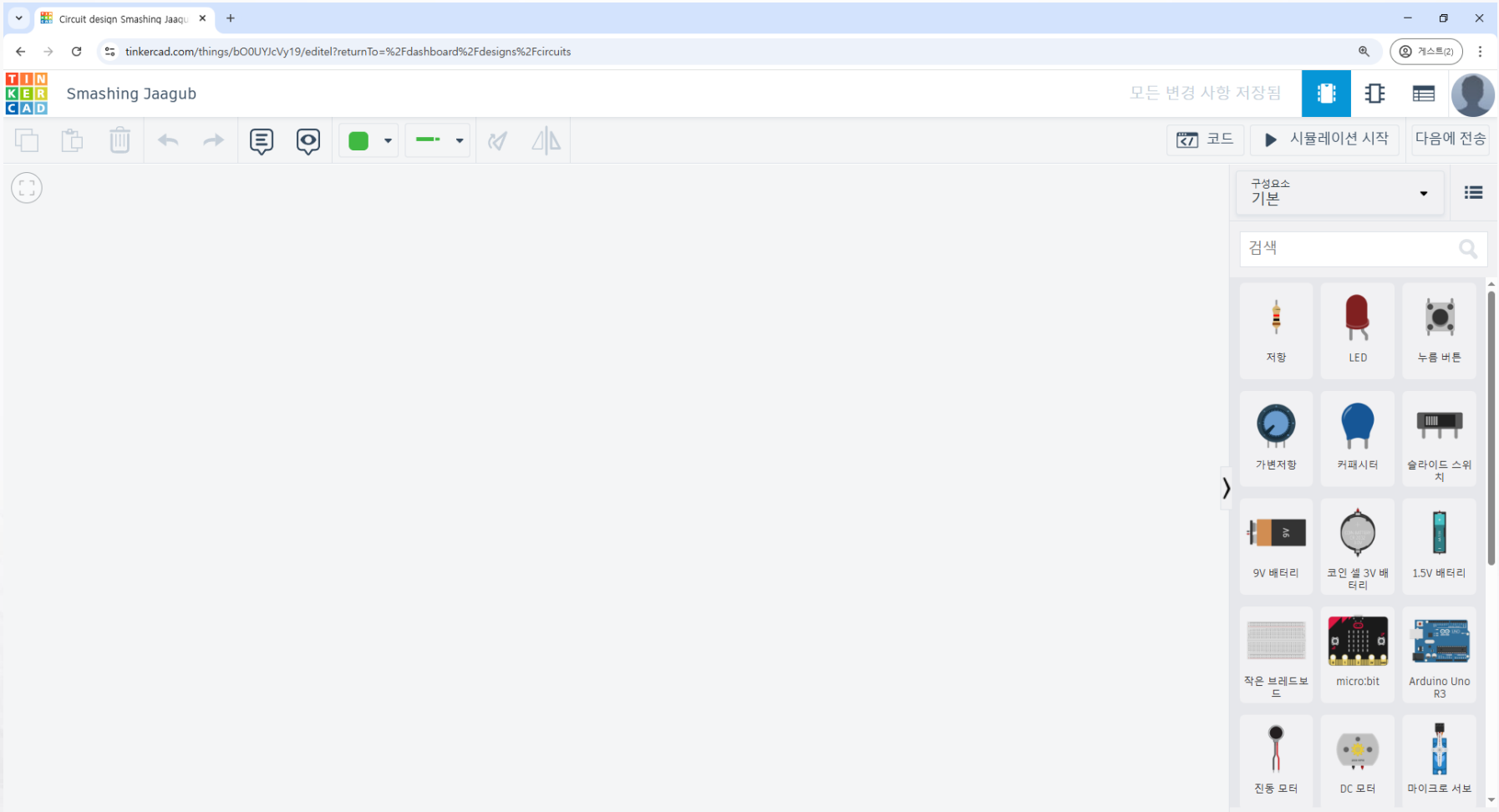
❖ 아두이노 온라인 실습 사이트

- https://www.tinkercad.com/



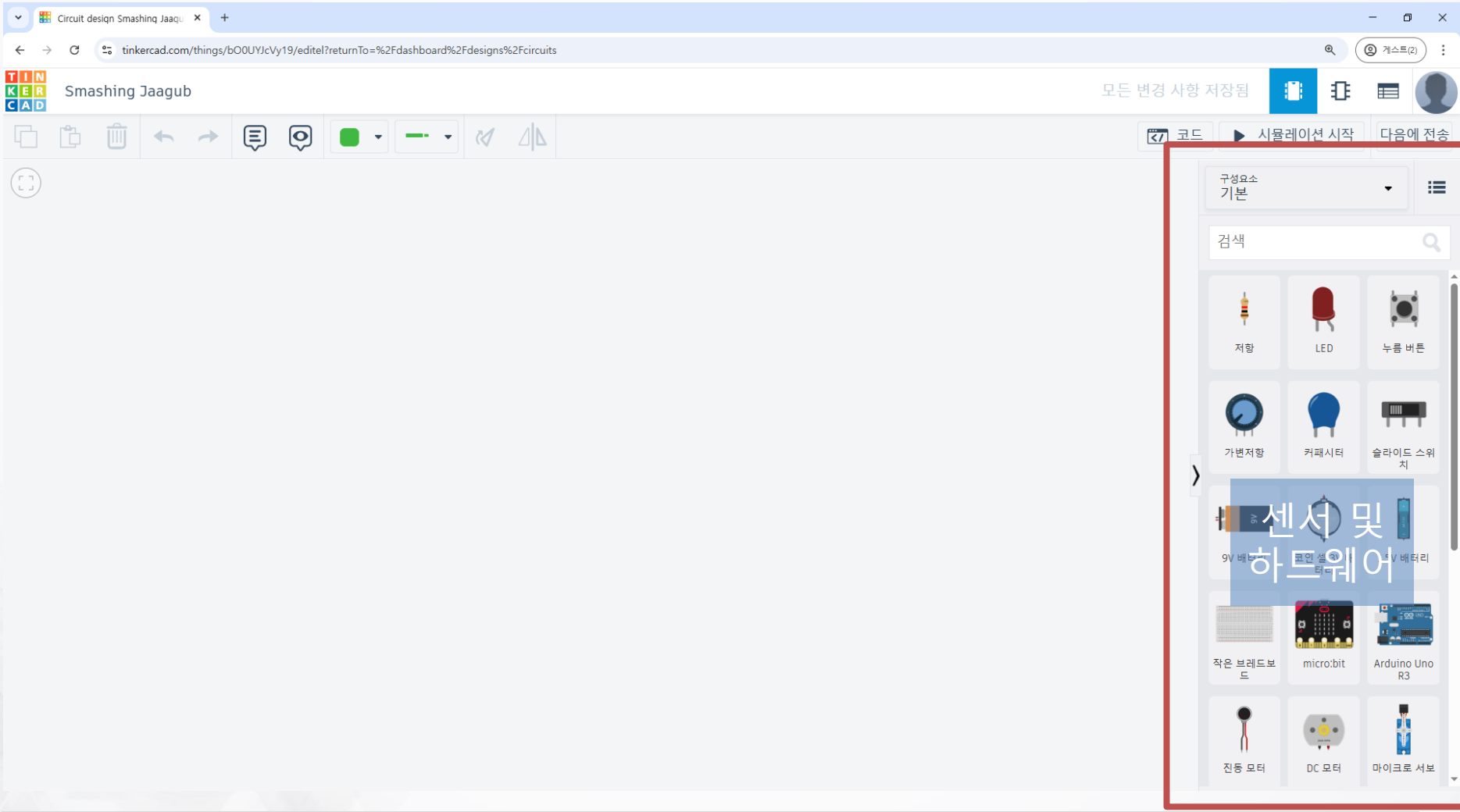
사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 실습



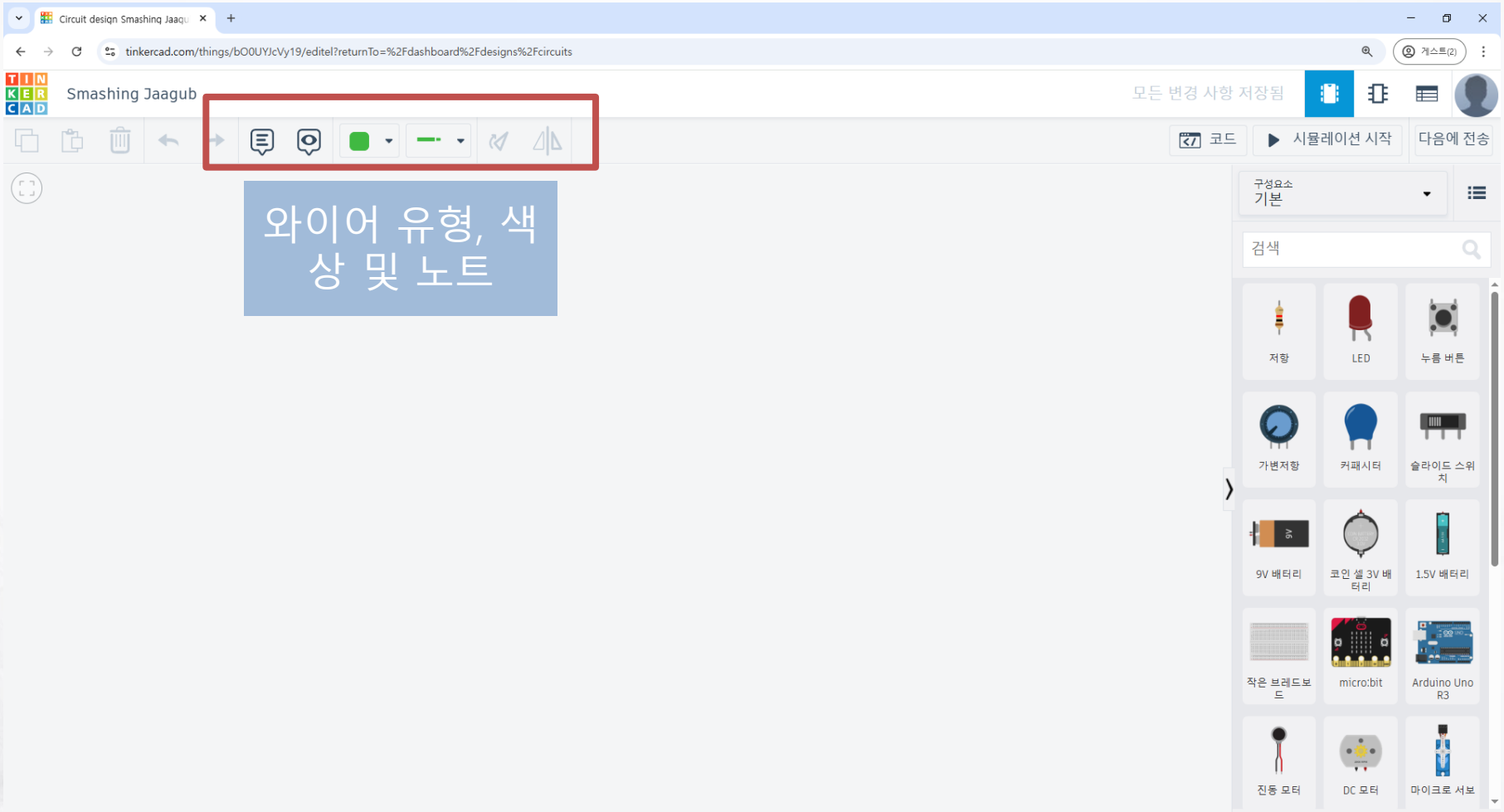
사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 실습



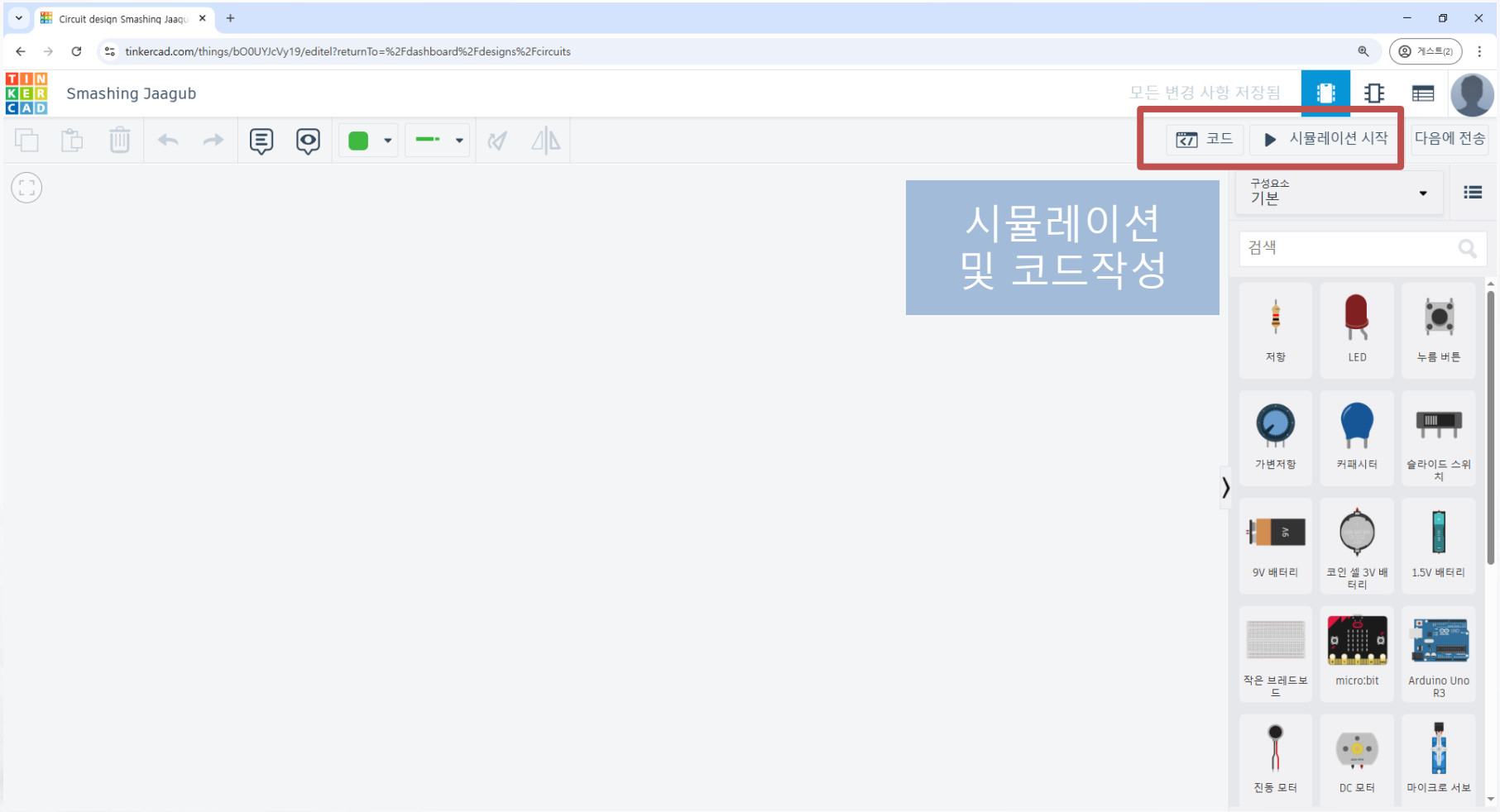
사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 실습



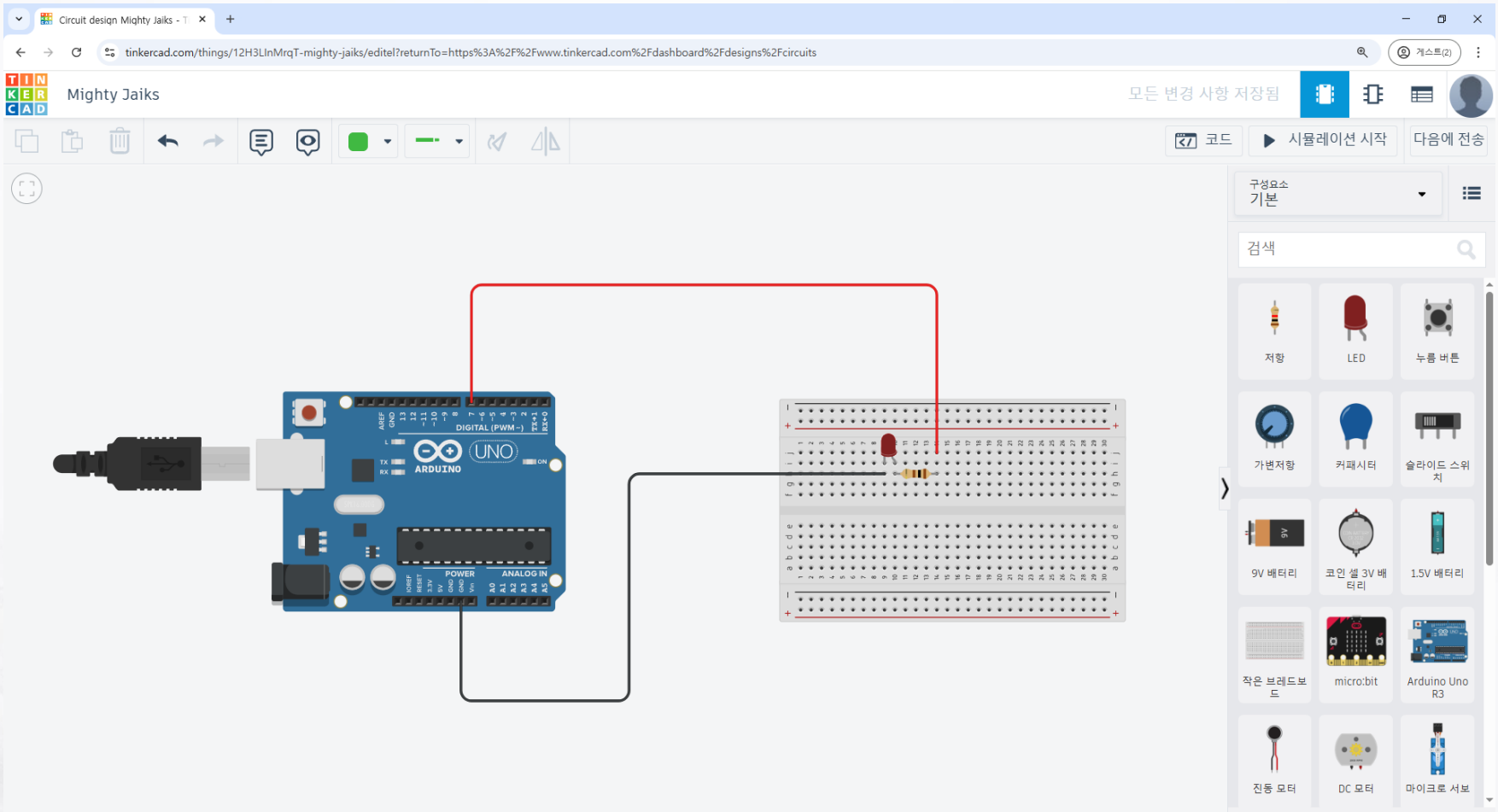
사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 실습



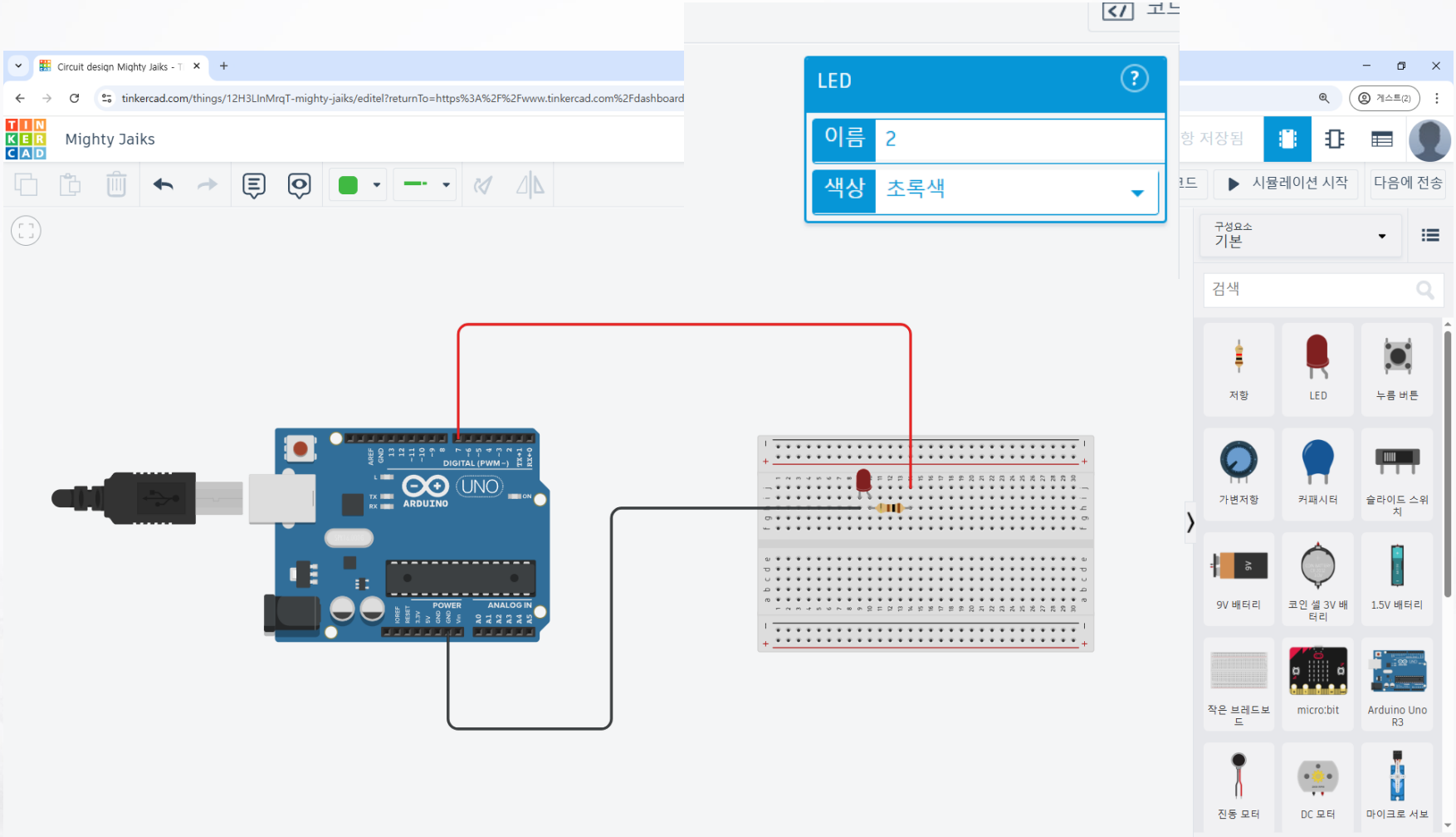
사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 LED 실습

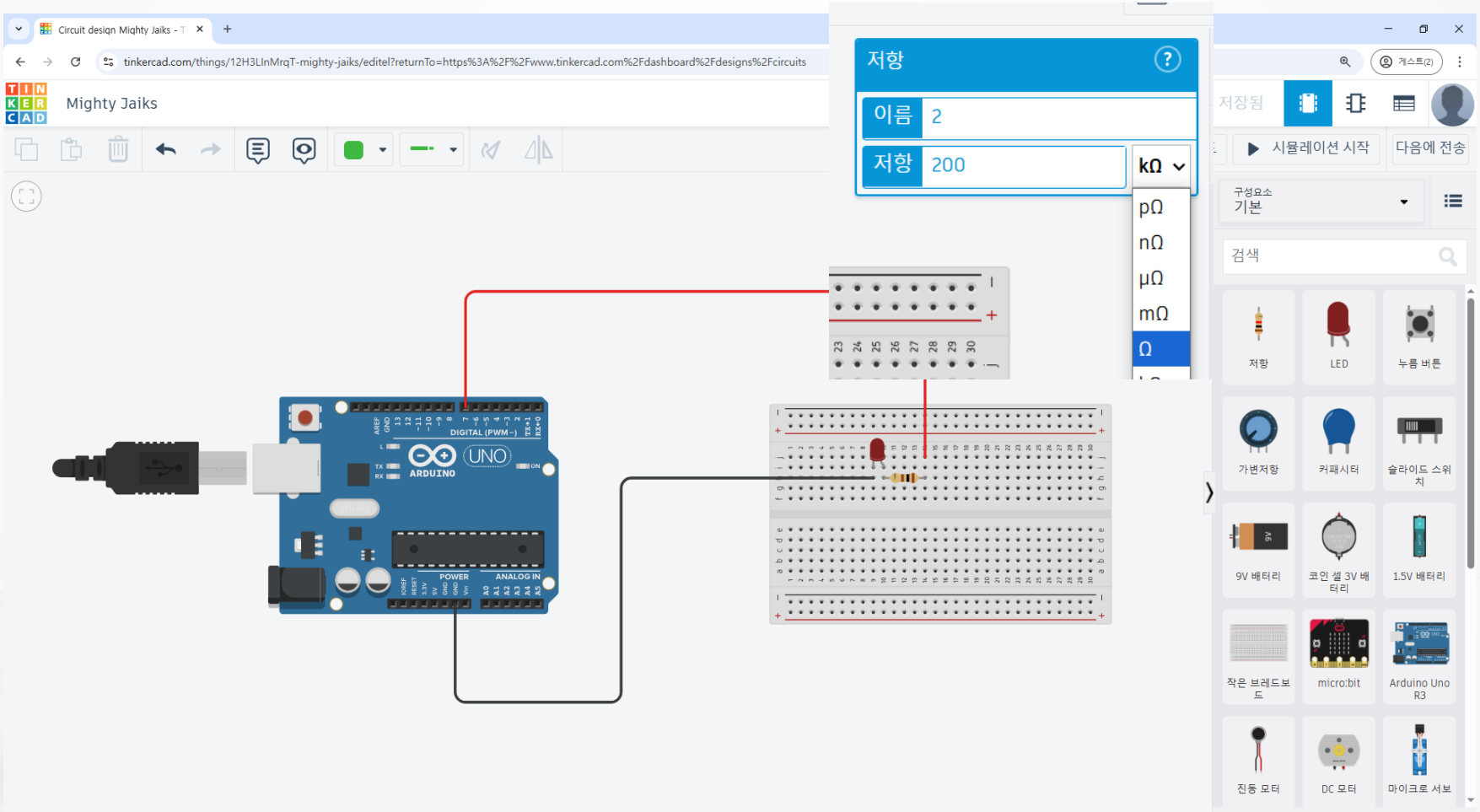


사물인터넷 - 실습

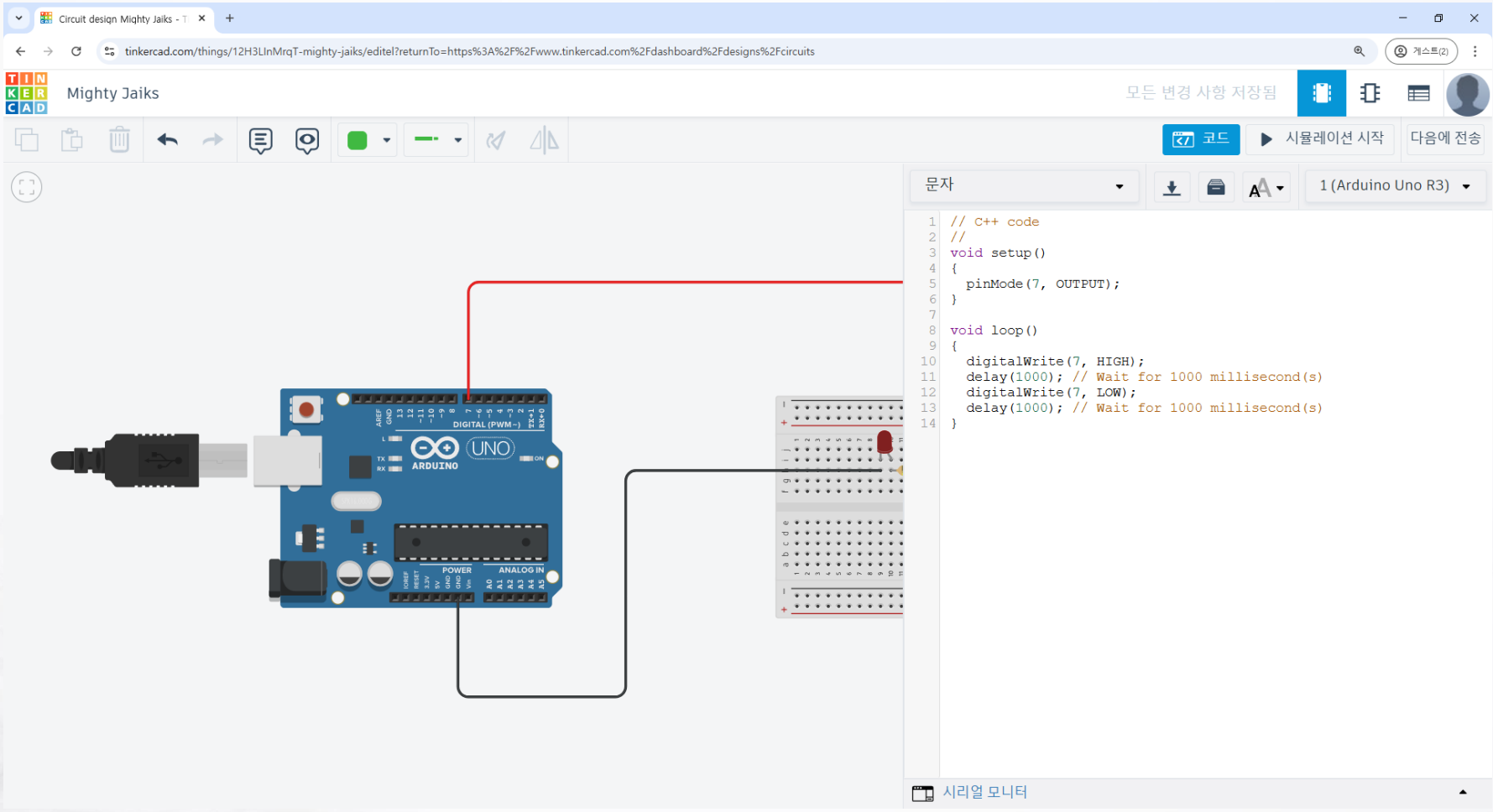
❖ 팅커캐드를 이용한 아두이노 LED 색상 변경 (LED 클릭 -> 색상변경)



❖ 팅커캐드를 이용한 아두이노 저항 옴 변경 (저항 클릭 -> 옴 변경)

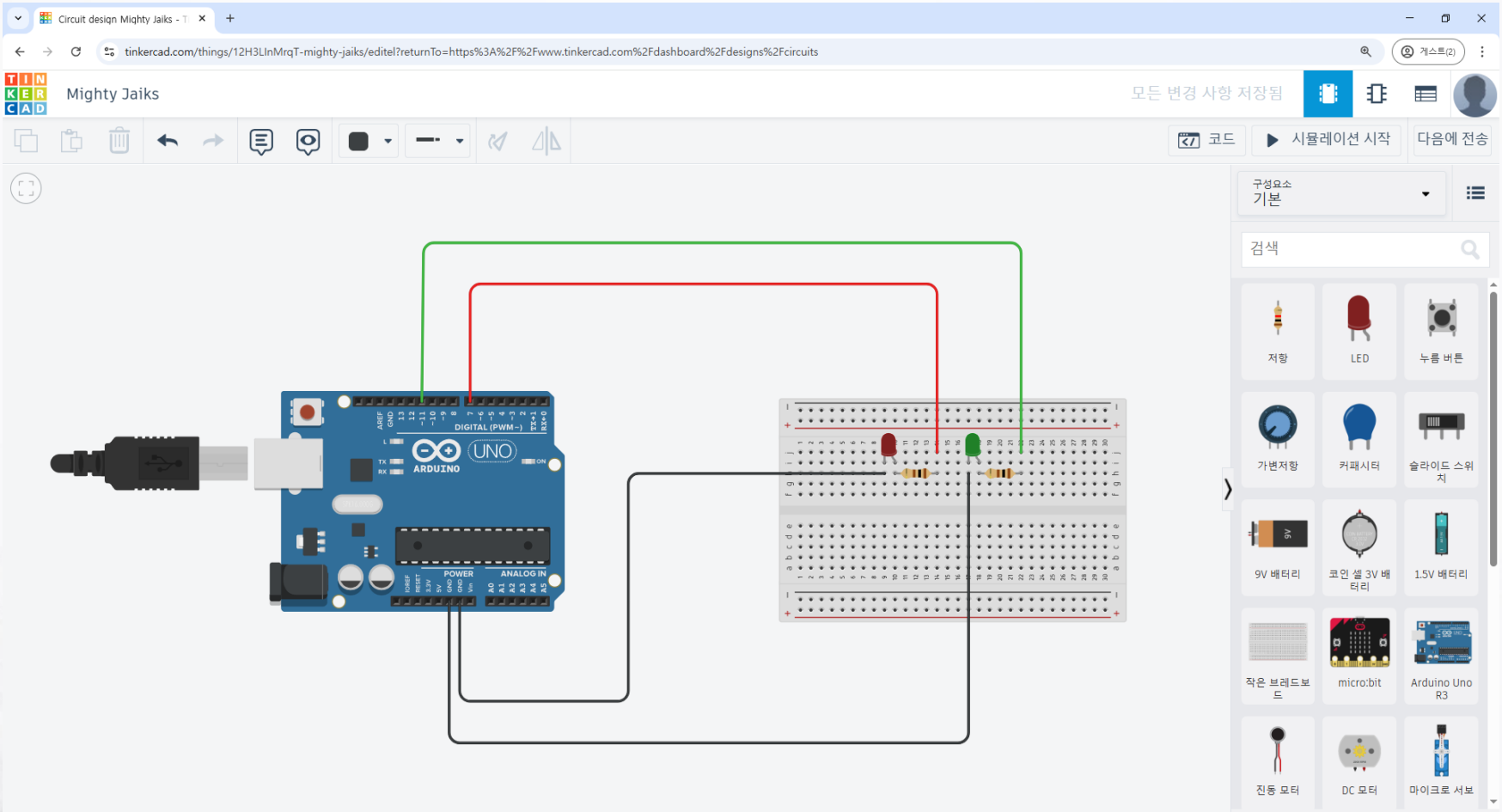


❖ 팅커캐드를 이용한 아두이노 LED 실습

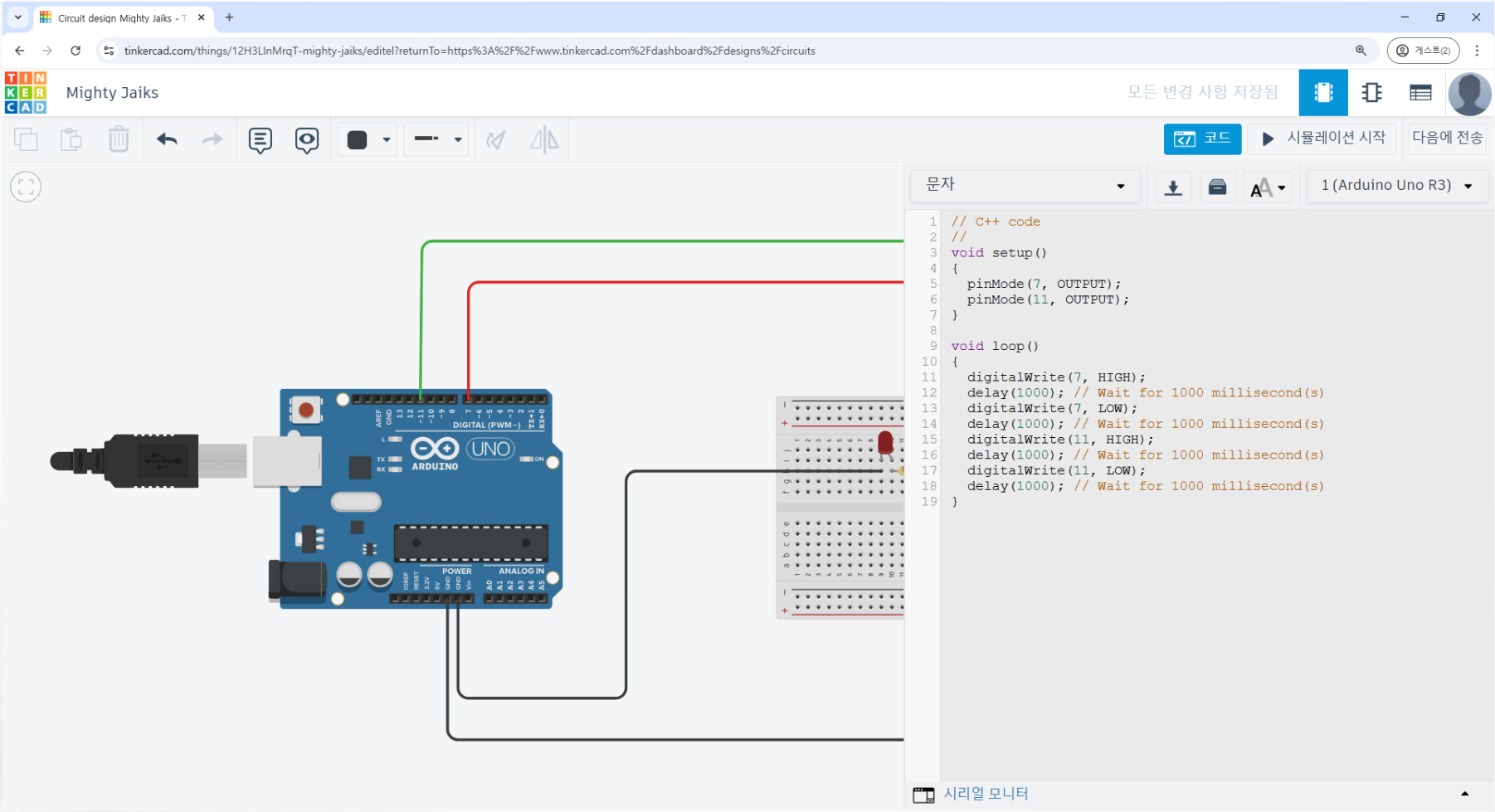


사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 LED 2개 실습

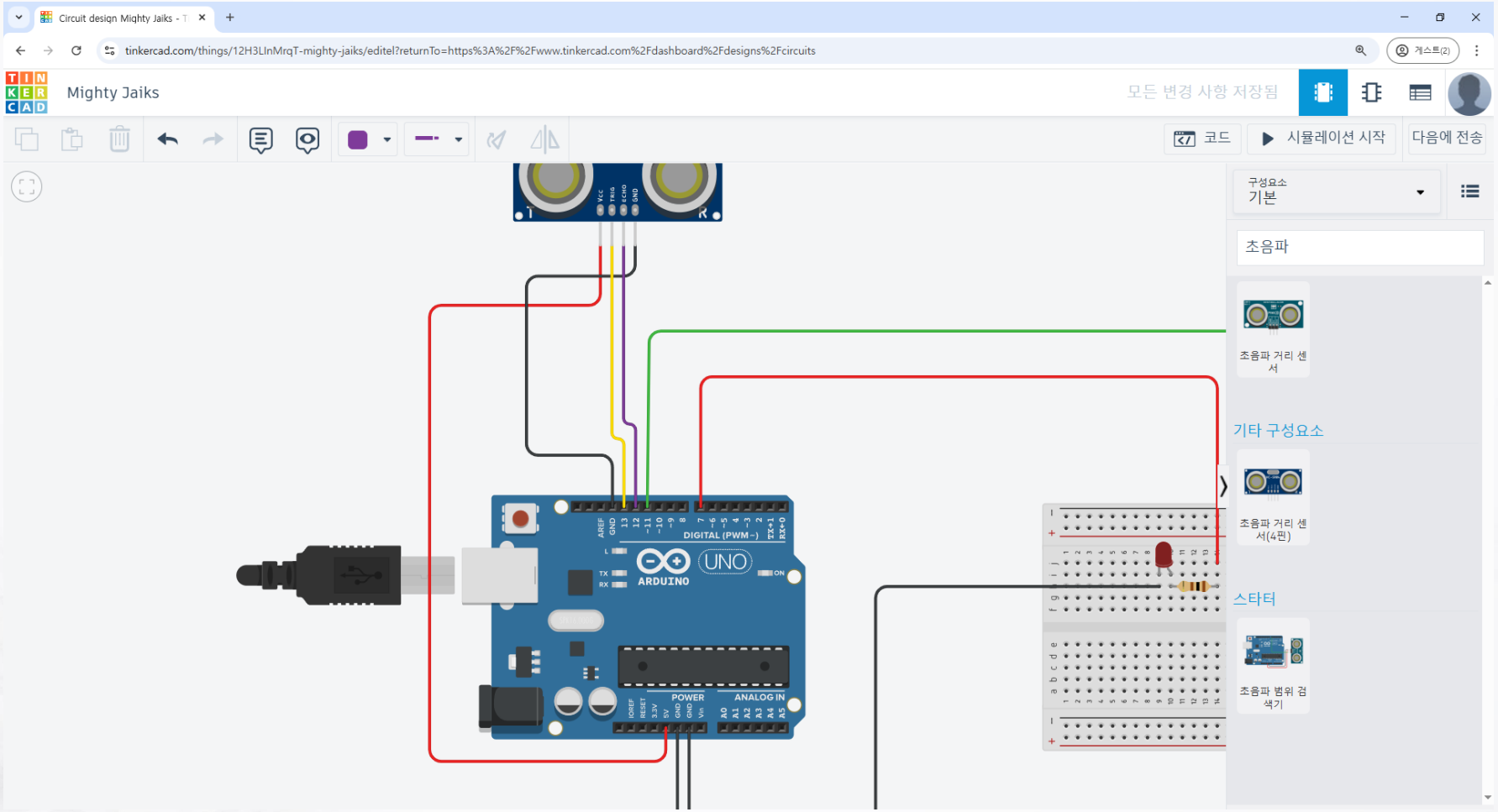


❖ 팅커캐드를 이용한 아두이노 LED 2개 실습



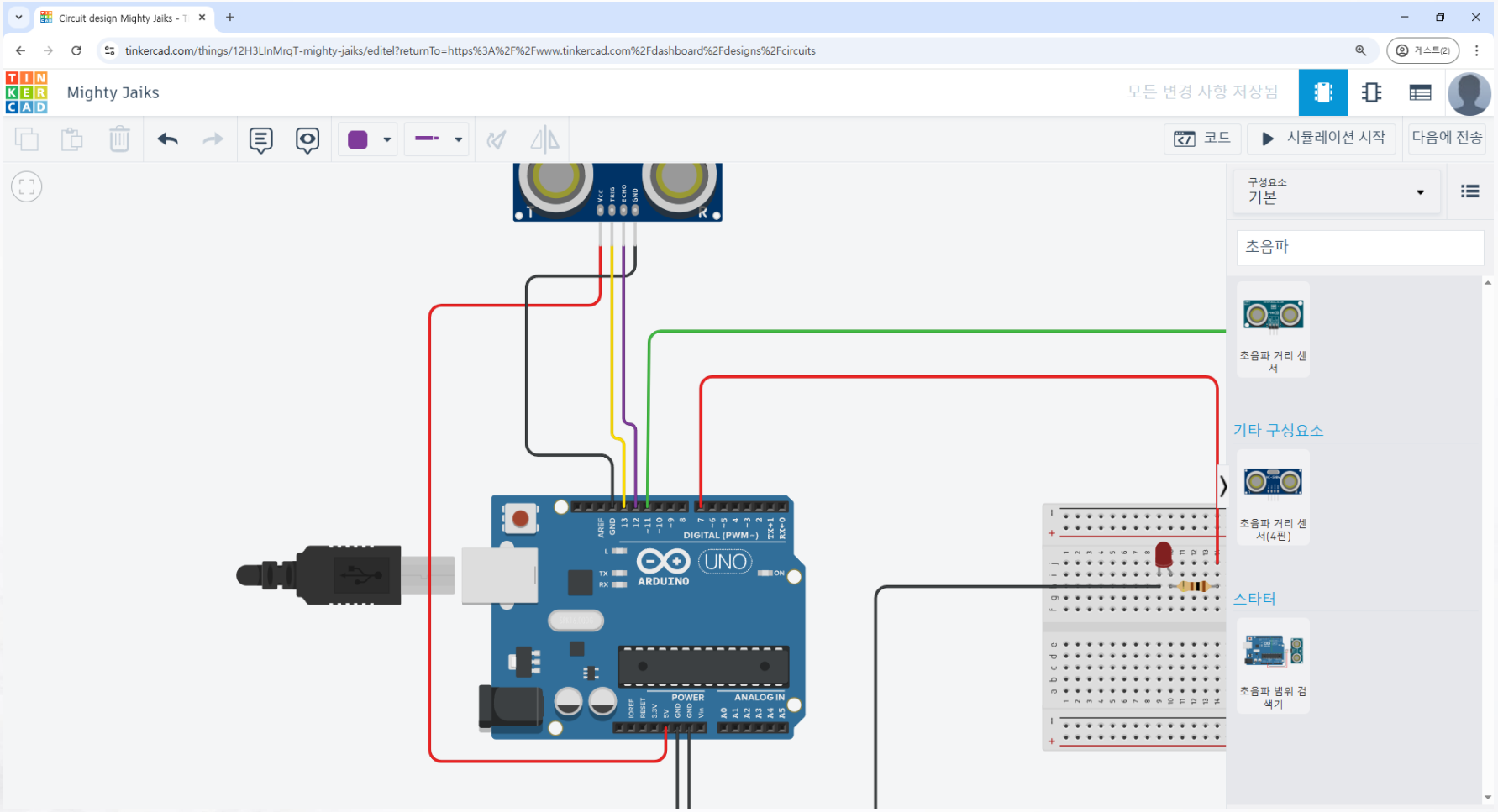
사물인터넷 - 실습

❖ 팅커캐드를 이용한 아두이노 LED 2개 && 초음파(4핀) 실습



사물인터넷 - 실습

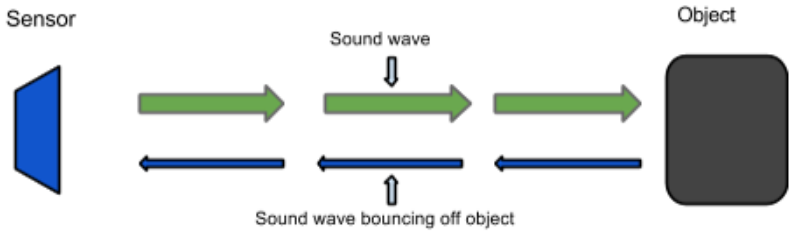
❖ 팅커캐드를 이용한 아두이노 LED 2개 && 초음파(4핀) 실습



❖ HC-SR04 초음파 센서 (DataSheet)

1. How Ultrasonic Sensors Work

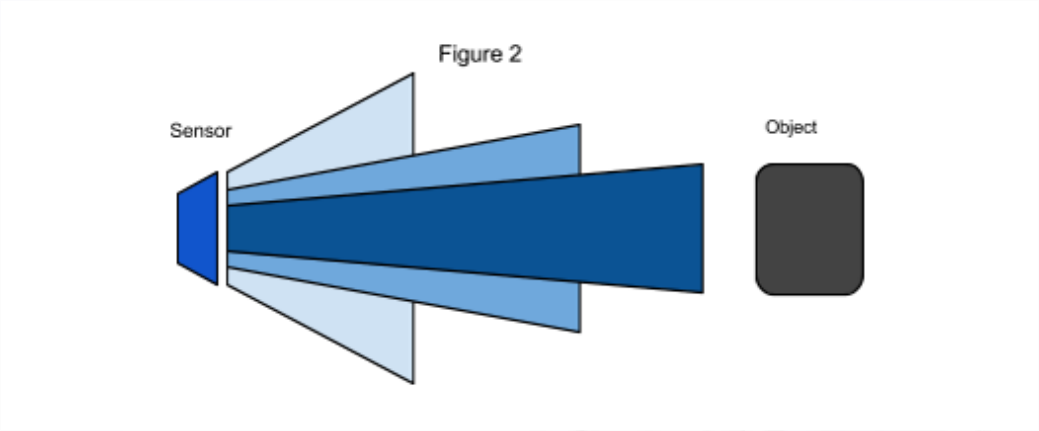
Ultrasonic sensors use sound to determine the distance between the sensor and the closest object in its path. How do ultrasonic sensors do this? Ultrasonic sensors are essentially sound sensors, but they operate at a frequency above human hearing.



The sensor sends out a sound wave at a specific frequency. It then listens for that specific sound wave to bounce off of an object and come back (Figure 1). The sensor keeps track of the time between sending the sound wave and the sound wave returning. If you know how fast something is going and how long it is traveling you can find the distance traveled with equation 1.

Equation 1. $d = v \times t$

❖ HC-SR04 초음파 센서 (DataSheet) - 1



- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2-400 cm
- Resolution: 0.3 cm
- Measuring Angle: 30°
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm
- Weight: approx. 10 g

❖ HC-SR04 초음파 센서 (DataSheet) - 2

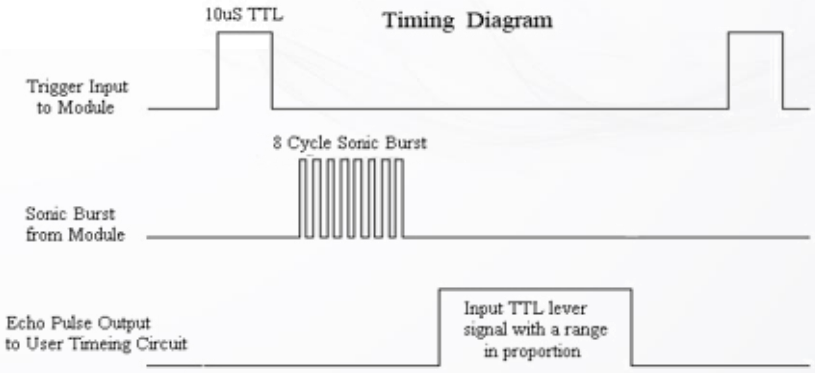
3. Timing Chart and Pin Explanations

The HC-SR04 has four pins, VCC, GND, TRIG and ECHO; these pins all have different functions. The VCC and GND pins are the simplest -- they power the HC-SR04. These pins need to be attached to a +5 volt source and ground respectively. There is a single control pin: the TRIG pin. The TRIG pin is responsible for sending the ultrasonic burst. This pin should be set to HIGH for 10 µs, at which point the HC-SR04 will send out an eight cycle sonic burst at 40 kHZ. After a sonic burst has been sent the ECHO pin will go HIGH. The ECHO pin is the data pin -- it is used in taking distance measurements. After an ultrasonic burst is sent the pin will go HIGH, it will stay high until an ultrasonic burst is detected back, at which point it will go LOW.

Taking Distance Measurements

The HC-SR04 can be triggered to send out an ultrasonic burst by setting the TRIG pin to HIGH. Once the burst is sent the ECHO pin will automatically go HIGH. This pin will remain HIGH until the the burst hits the sensor again. You can calculate the distance to the object by keeping track of how long the ECHO pin stays HIGH. The time ECHO stays HIGH is the time the burst spent traveling. Using this measurement in equation 1 along with the speed of sound will yield the distance travelled. A summary of this is listed below, along with a visual representation in Figure 2.

Figure 3
Source 2



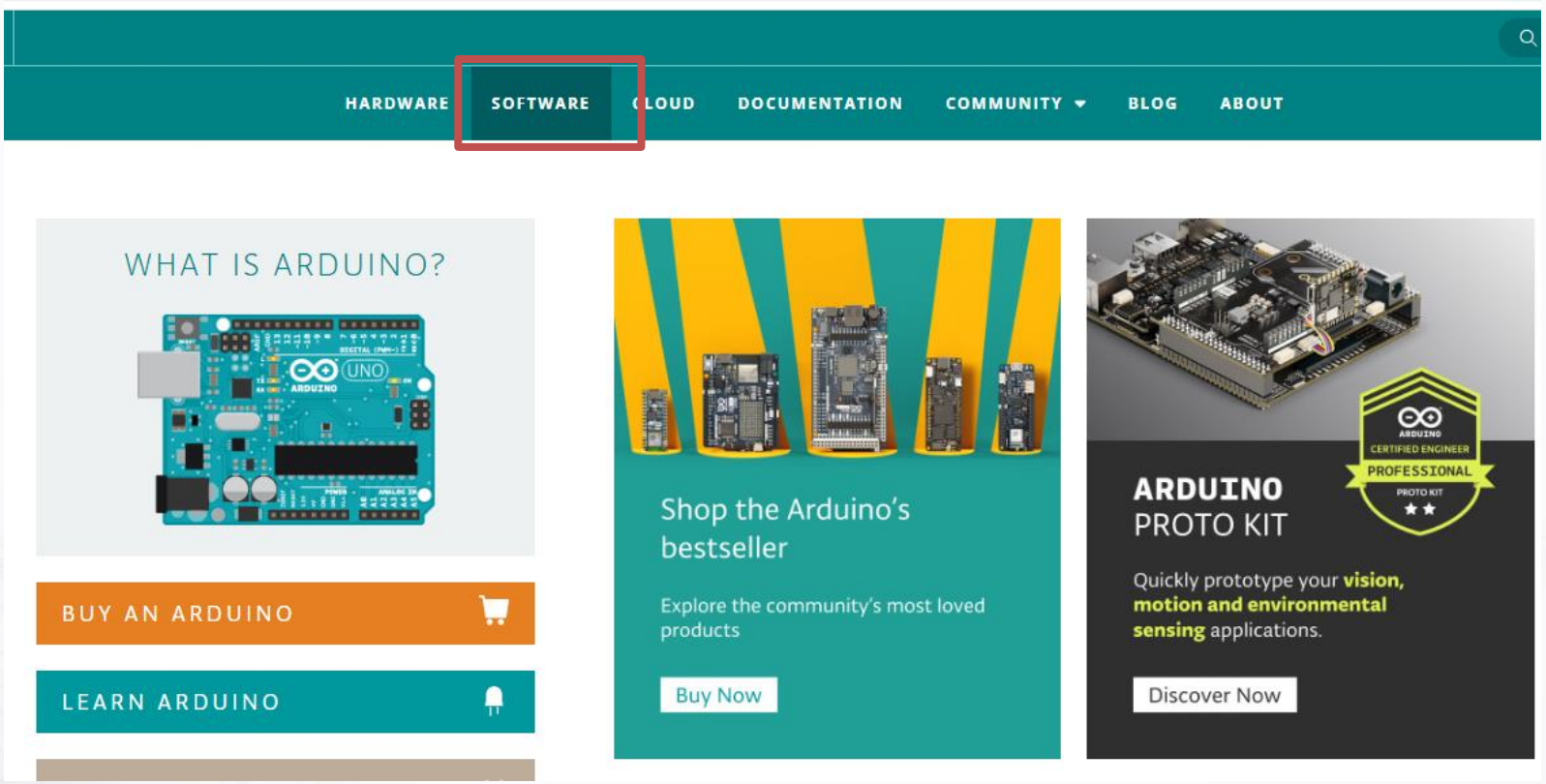
❖ **HC-SR04 초음파 센서 CODE Serial 사용**

LED_test.ino

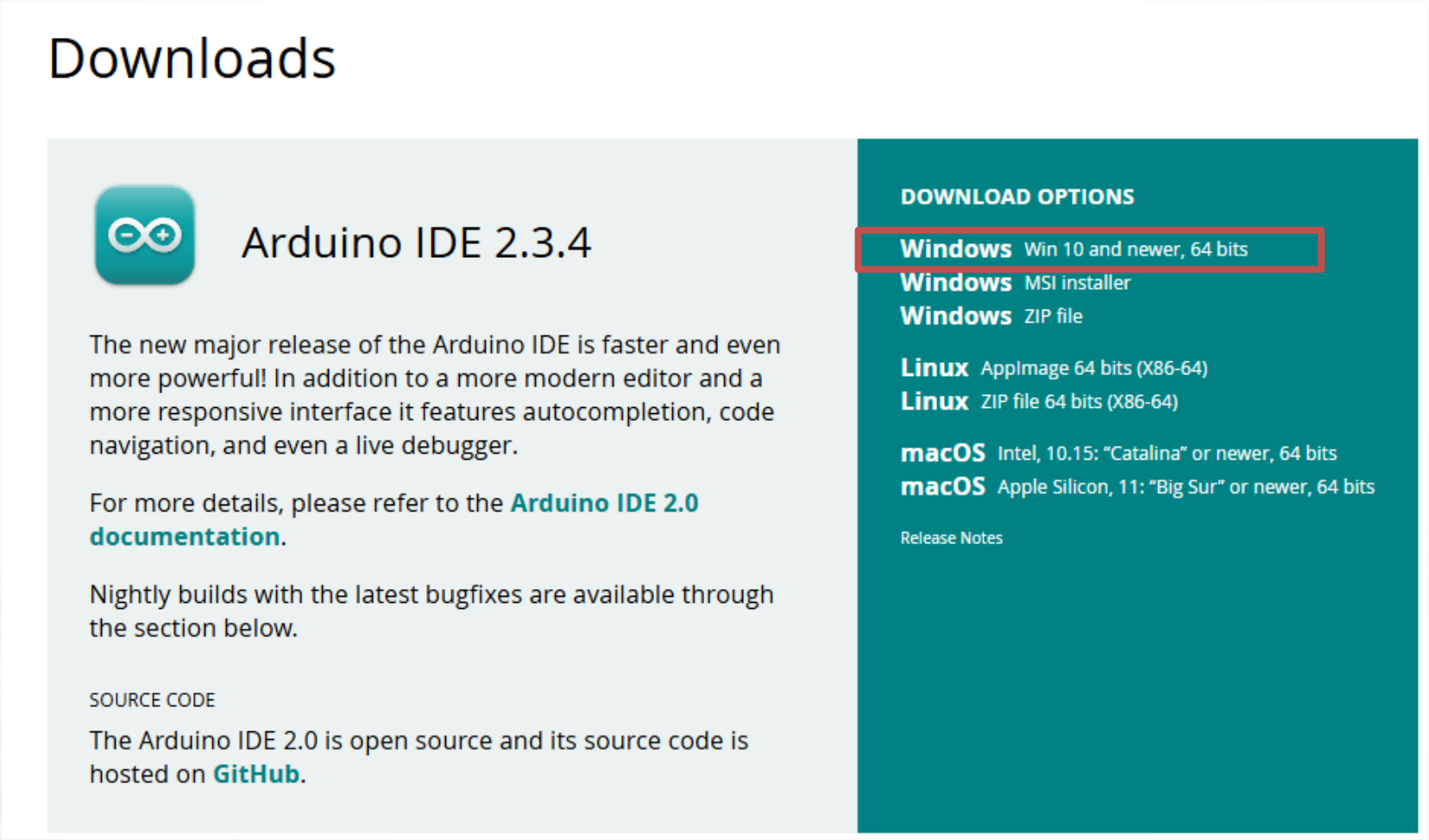
```
1
2   #define TRIG 9 //TRIG 핀 설정 (초음파 보내는 핀)
3   #define ECHO 8 //ECHO 핀 설정 (초음파 받는 핀)
4
5   void setup() {
6
7     Serial.begin(9600);
8
9     pinMode(TRIG, OUTPUT);
10    pinMode(ECHO, INPUT);
11  }
12
13
14
15  void loop()
16  {
17    long duration, distance;
18
19    digitalWrite(TRIG, LOW);
20    delayMicroseconds(2);
21    digitalWrite(TRIG, HIGH);
22    delayMicroseconds(10);
23    digitalWrite(TRIG, LOW);
24
25    duration = pulseIn (ECHO, HIGH);
26    distance = duration * 17 / 1000;   //distance = duration / 58.2;
27
28    Serial.println(duration );
29    Serial.print("\nDIstance : ");
30    Serial.print(distance);
31    Serial.println(" Cm");
32    delay(1000);
33
34  }
35
```

❖ 아두이노 설치

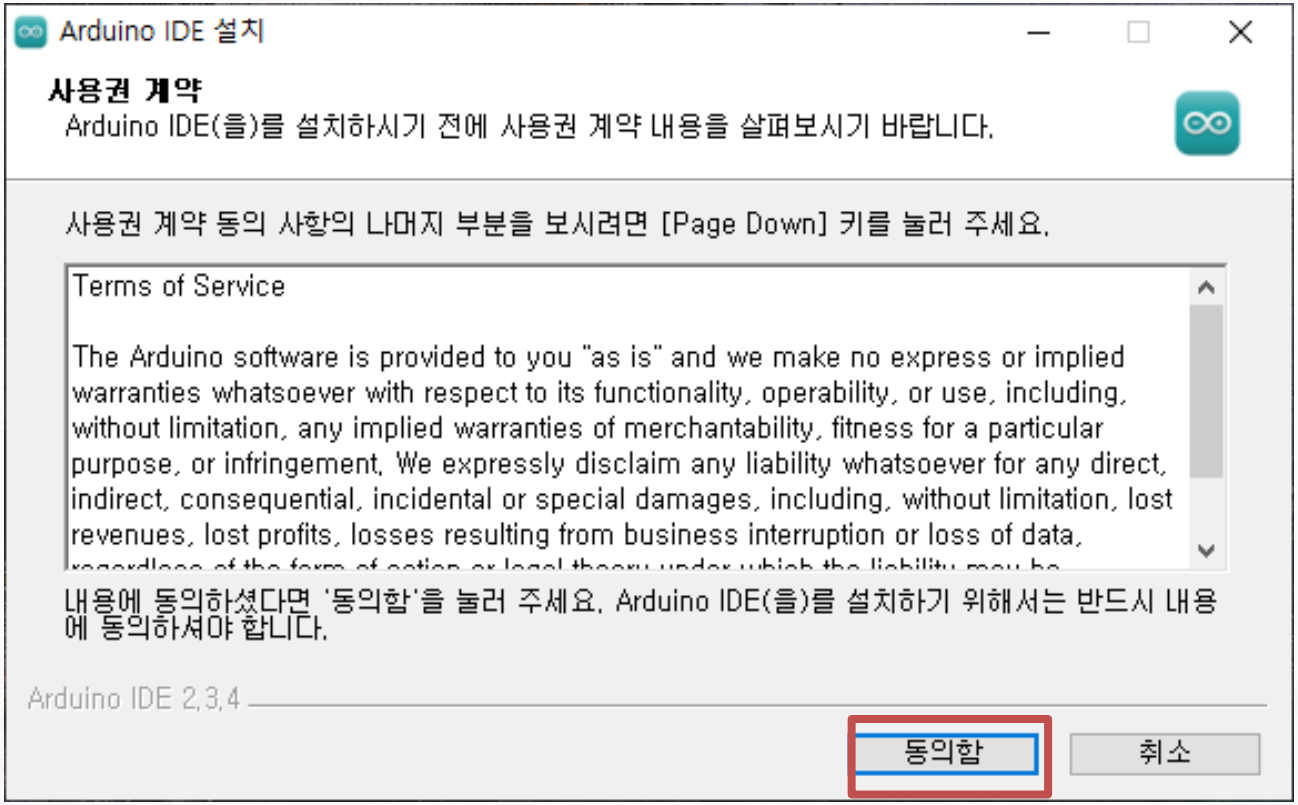
https://www.arduino.cc/



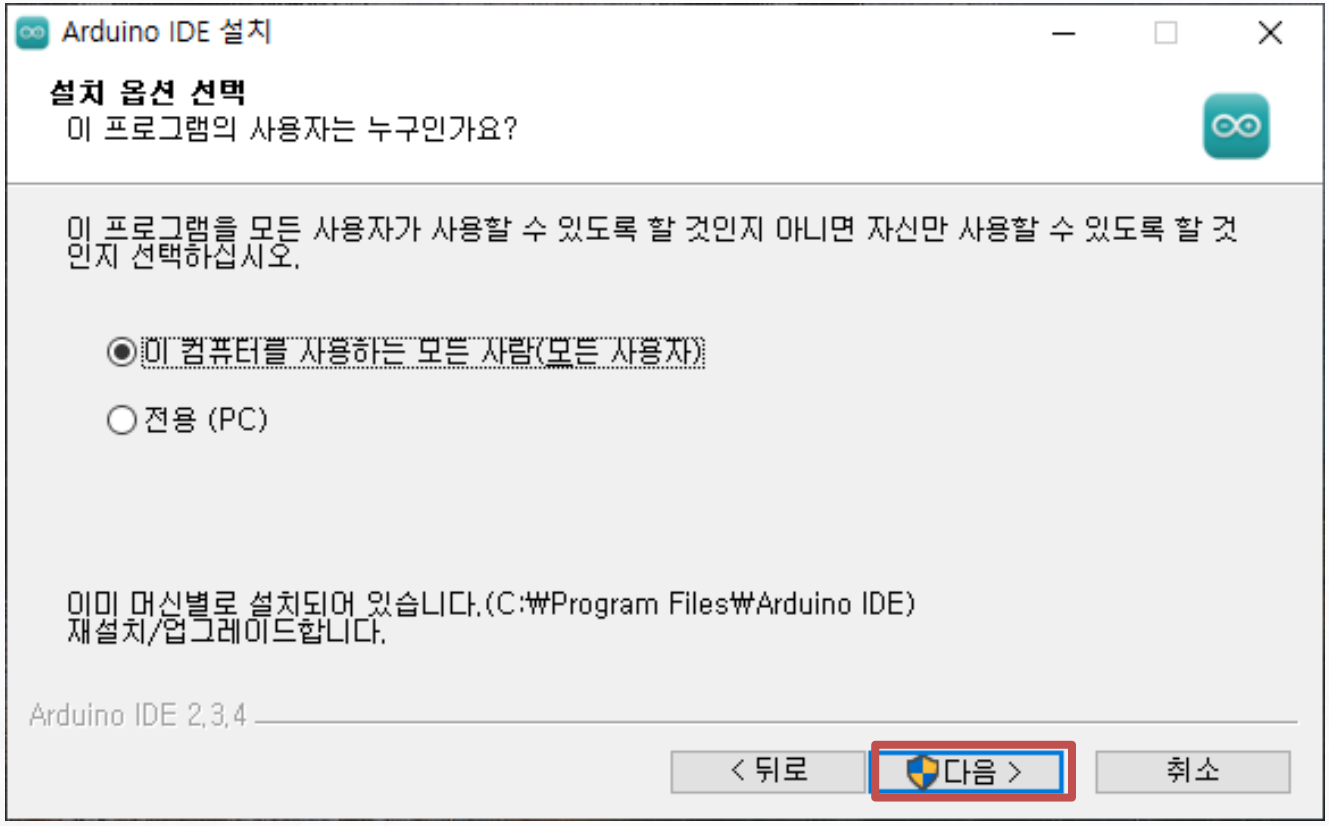
❖ 아두이노 설치 - 1


Downloads
Arduino IDE 2.3.4
The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.
For more details, please refer to the Arduino IDE 2.0 documentation.
Nightly builds with the latest bugfixes are available through the section below.
SOURCE CODE
The Arduino IDE 2.0 is open source and its source code is hosted on GitHub.
DOWNLOAD OPTIONS
Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file
Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)
macOS Intel, 10.15: "Catalina" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits
Release Notes

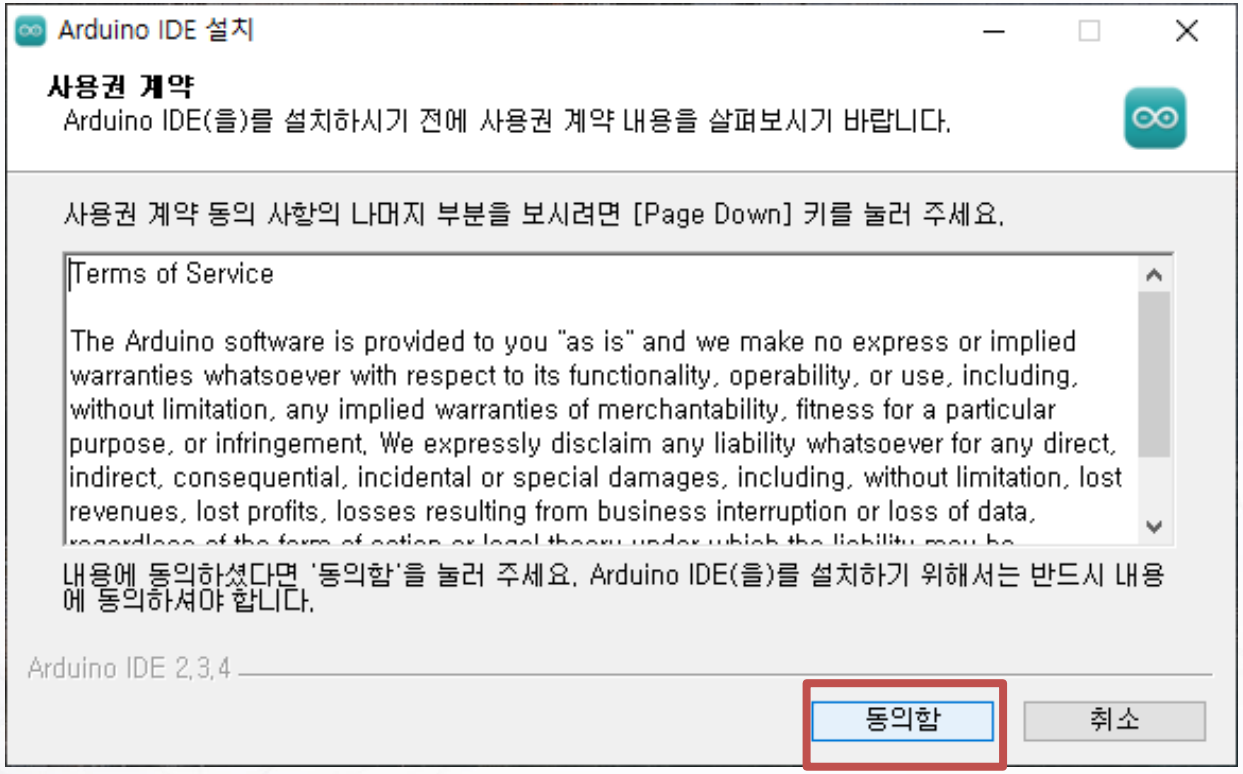
❖ 아두이노 설치 - 2


Arduino IDE 설치
사용권 계약
Arduino IDE(을)를 설치하시기 전에 사용권 계약 내용을 살펴보시기 바랍니다.
사용권 계약 동의 사항의 나머지 부분을 보시려면 [Page Down] 키를 눌러 주세요.
Terms of Service
The Arduino software is provided to you "as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data,
내용에 동의하셨다면 '동의함'을 눌러 주세요. Arduino IDE(을)를 설치하기 위해서는 반드시 내용에 동의하셔야 합니다.
Arduino IDE 2.3.4
동의함
취소

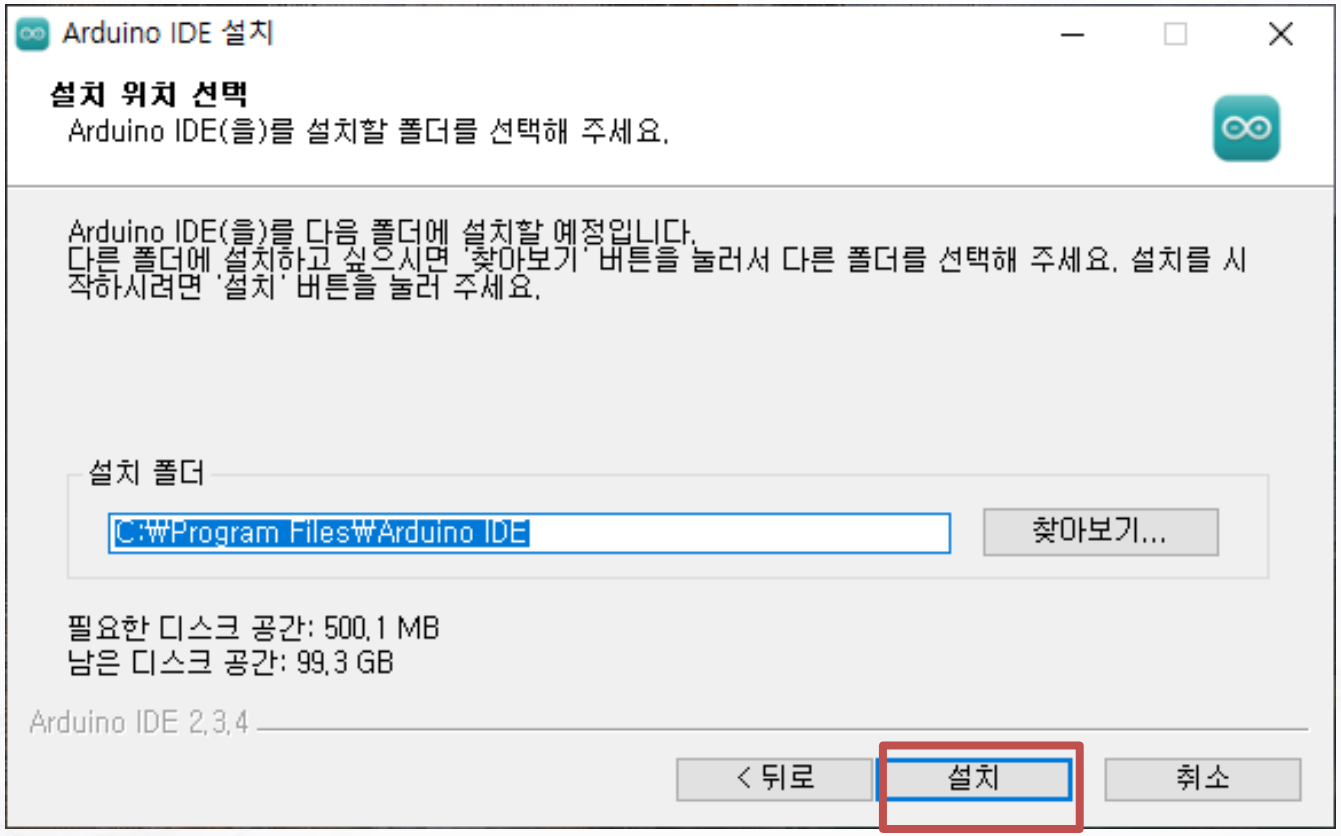
❖ 아두이노 설치 - 3


Arduino IDE 설치
설치 옵션 선택
이 프로그램의 사용자는 누구인가요?
이 프로그램을 모든 사용자가 사용할 수 있도록 할 것인지 아니면 자신만 사용할 수 있도록 할 것인지 선택하십시오.
이 컴퓨터를 사용하는 모든 사람(모든 사용자)
전용 (PC)
이미 머신별로 설치되어 있습니다.(C:₩Program Files₩Arduino IDE)
재설치/업그레이드합니다.
Arduino IDE 2.3.4
< 뒤로
다음 >
취소

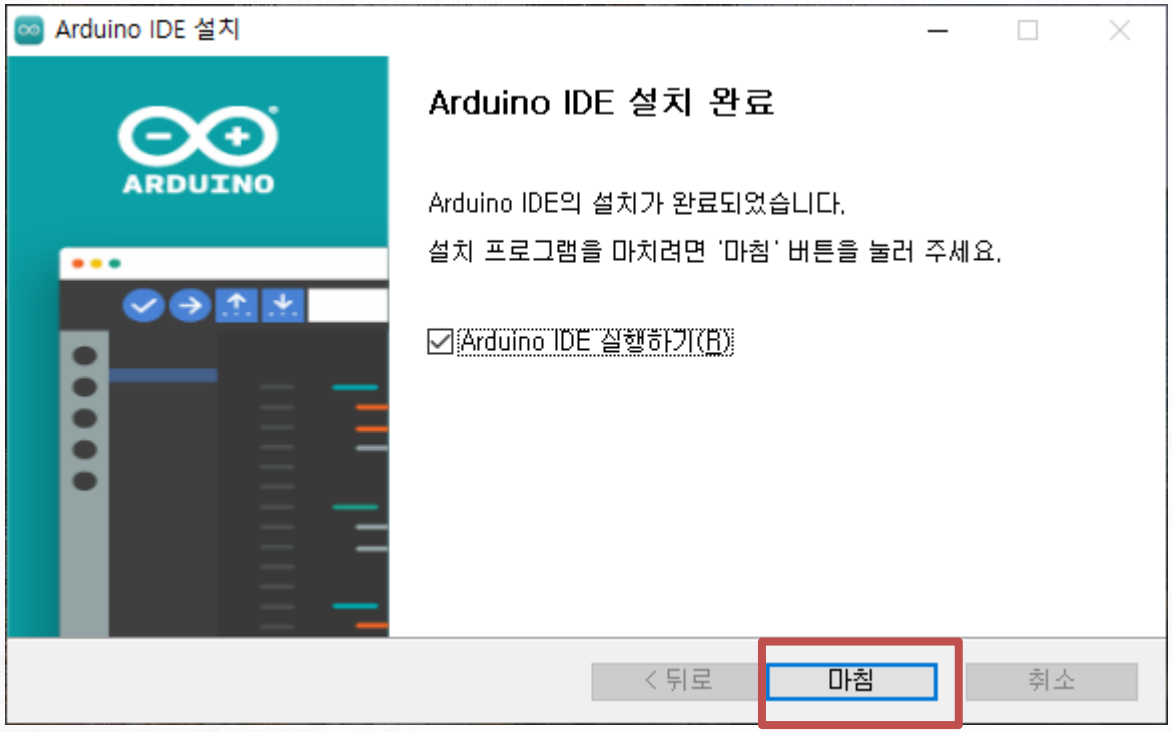
❖ 아두이노 설치 - 4


Arduino IDE 설치
사용권 계약
Arduino IDE(을)를 설치하시기 전에 사용권 계약 내용을 살펴보시기 바랍니다.
사용권 계약 동의 사항의 나머지 부분을 보시려면 [Page Down] 키를 눌러 주세요.
Terms of Service
The Arduino software is provided to you "as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data,
내용에 동의하셨다면 '동의함'을 눌러 주세요. Arduino IDE(을)를 설치하기 위해서는 반드시 내용에 동의하셔야 합니다.
Arduino IDE 2.3.4
동의함
취소

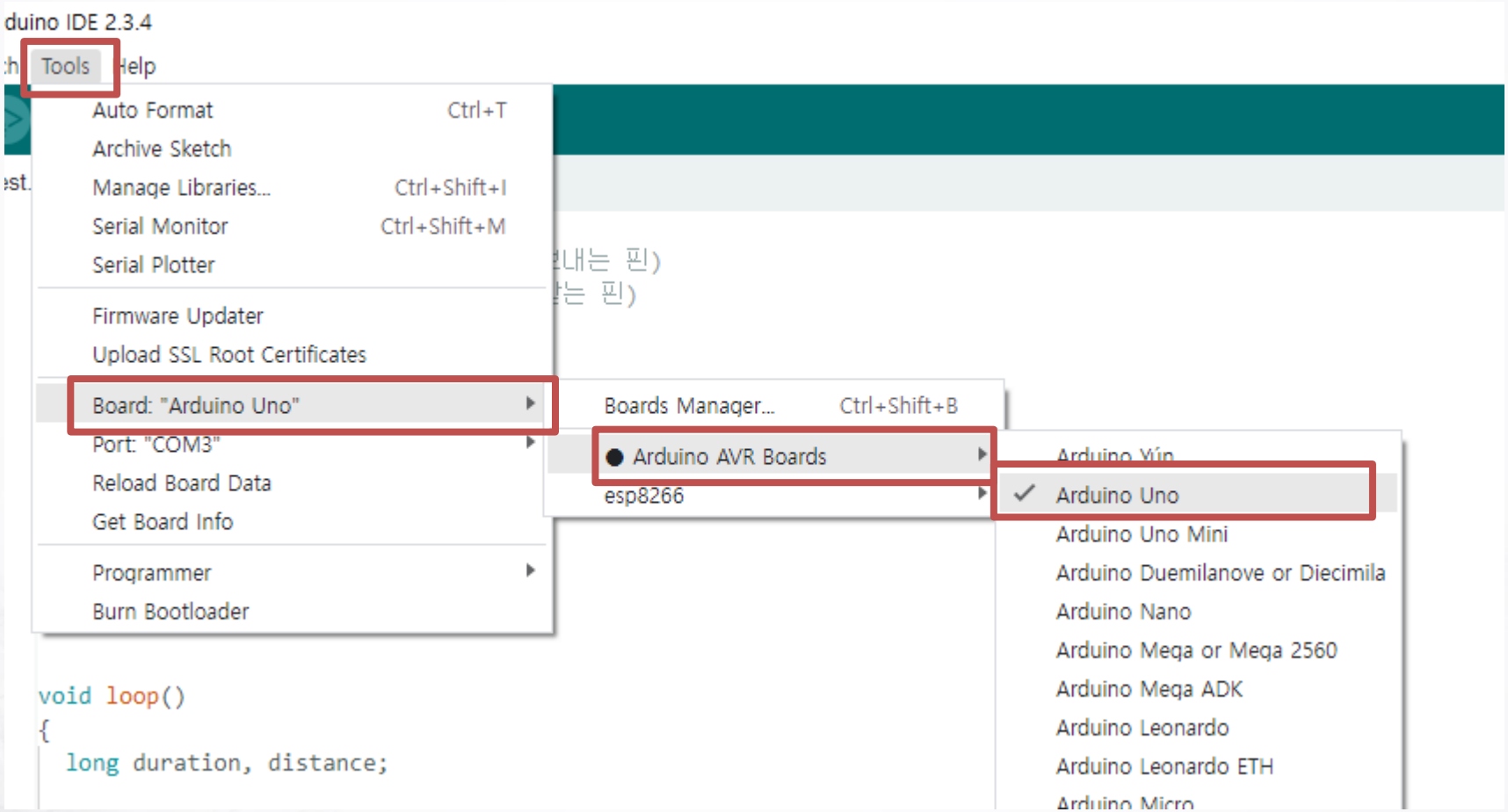
❖ 아두이노 설치 - 5


Arduino IDE 설치
설치 위치 선택
Arduino IDE(을)를 설치할 폴더를 선택해 주세요.
Arduino IDE(을)를 다음 폴더에 설치할 예정입니다.
다른 폴더에 설치하고 싶으시면 '찾아보기' 버튼을 눌러서 다른 폴더를 선택해 주세요. 설치를 시작하시려면 '설치' 버튼을 눌러 주세요.
설치 폴더
C:₩Program Files₩Arduino IDE
찾아보기...
필요한 디스크 공간: 500.1 MB
남은 디스크 공간: 99.3 GB
Arduino IDE 2.3.4
< 뒤로
설치
취소

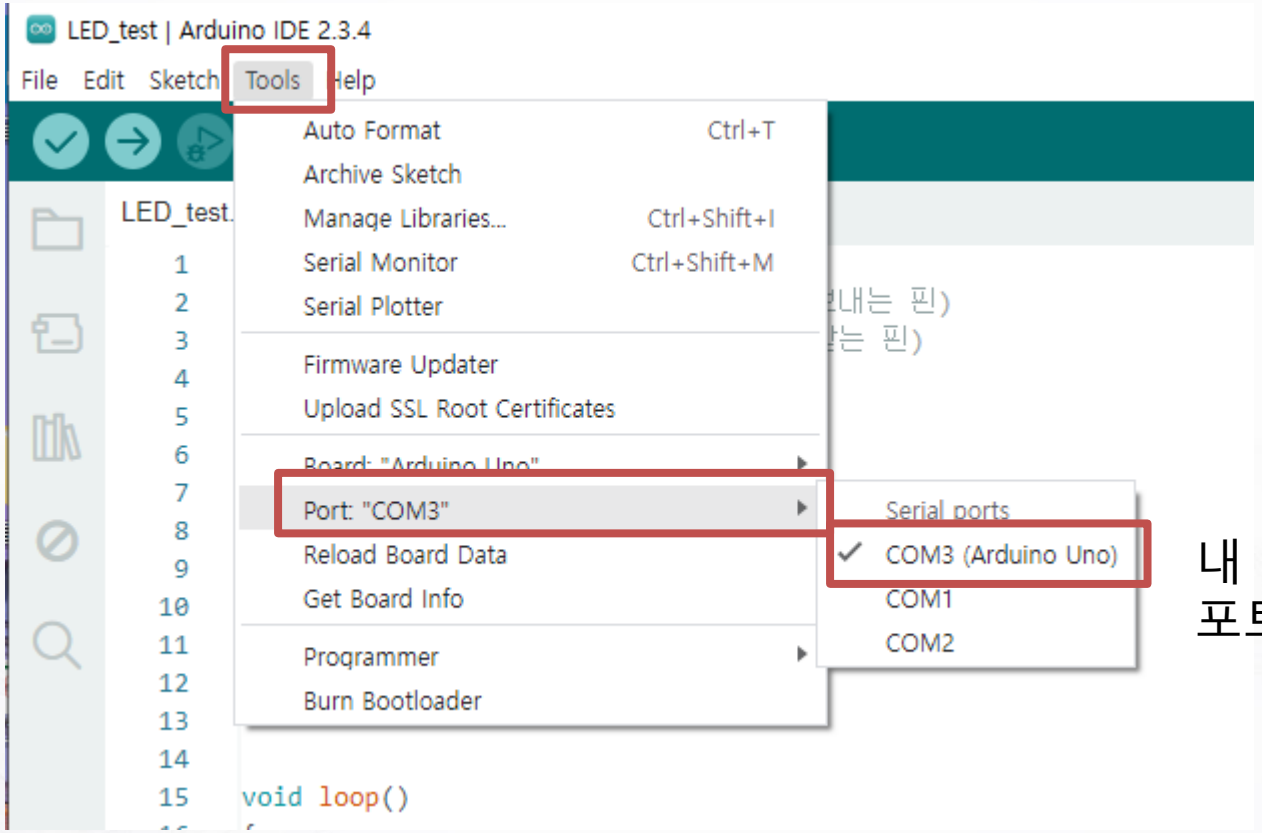
❖ 아두이노 설치 - 6


Arduino IDE 설치
ARDUINO
Arduino IDE 설치 완료
Arduino IDE의 설치가 완료되었습니다.
설치 프로그램을 마치려면 '마침' 버튼을 눌러 주세요.
Arduino IDE 실행하기(R)
< 뒤로
마침
취소

❖ 아두이노 IDE 실행 -> 보드설정 (Arduino Uno)

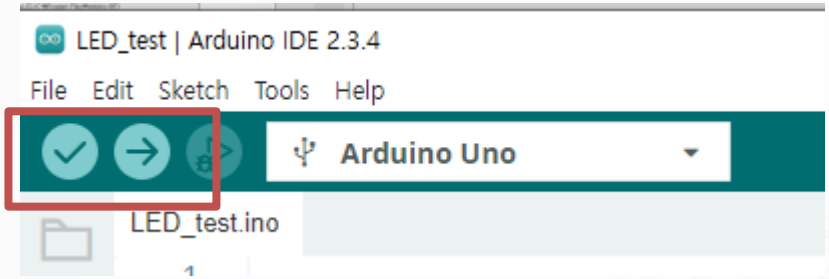


❖ 아두이노 IDE 실행 -> 포트설정(Arduino Uno)



내 컴퓨터에 연결된 포트설정

❖ 아두이노 IDE 실행 -> 컴파일 및 펌웨어 업로드



컴파일 (문법체크)

컴파일 및 펌웨어 업로드

❖ 응용실습

초음파 거리에 따른 LED 제어

LED_test.ino

```
#define TRIG 13 //TRIG 핀
#define ECHO 12 //ECHO 핀

int led_r = 7;
int led_g = 9;

void setup()
{
  Serial.begin(9600);
  pinMode(led_g, OUTPUT);
  pinMode(led_r, OUTPUT);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
}

void loop()
{
  long duration,distance;

  digitalWrite(TRIG, LOW);
  delayMicroseconds(2); //delay(2)
  digitalWrite(TRIG,HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);

  duration = pulseIn(ECHO, HIGH);

  distance = duration / 58.2;
  Serial.println(duration );
```

```
  Serial.print("\nDIstance : ");
  Serial.print(distance);
  Serial.println(" Cm");

  if(distance >=100)
  {
    digitalWrite(led_r, HIGH);
    digitalWrite(led_g, LOW);
  }
  else
  {
    digitalWrite(led_r, LOW);
    digitalWrite(led_g, HIGH);
  }
  delay(1000); // Wait for 1000 millisecond(s)
}
```

2주차 강의가 끝났습니다,

모두 고생하셨습니다.