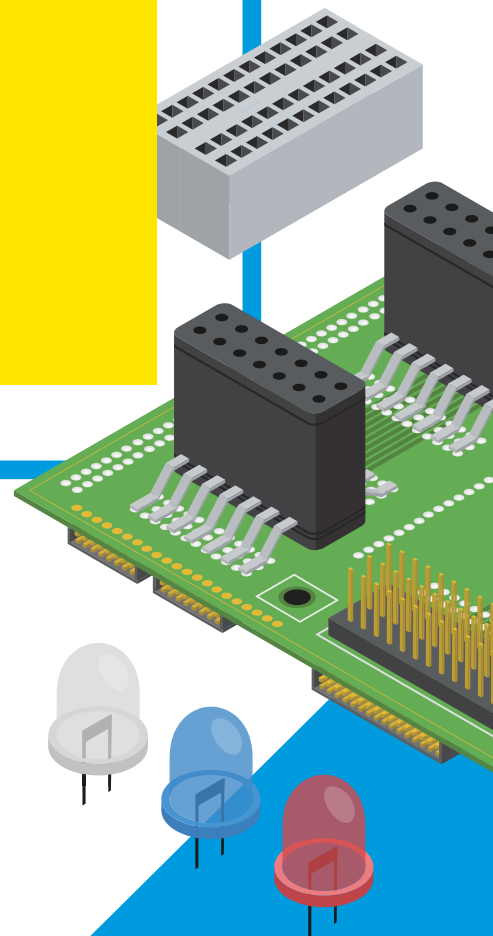
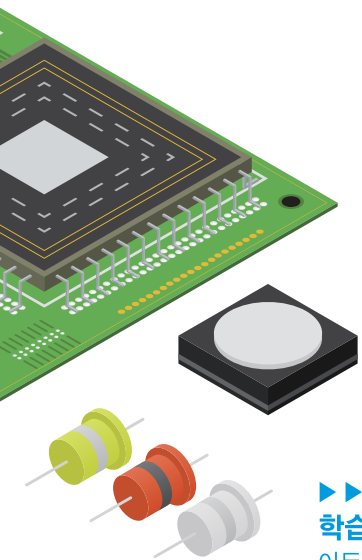




www.samyangm.com

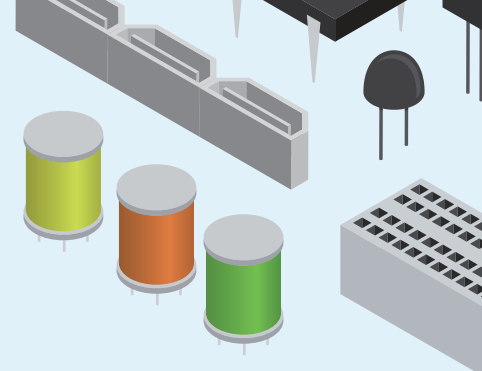
고등학교 정보·정보과학 아두이노 핵심 코드집



▶▶▶▶▶
학습에 필요한 준비 도구
아두이노와 브레드 보드, LED

(주)삼양미디어

차례



01

아두이노 시스템 구성 및 소스 코드 만들기

- 실습 1** 디지털 출력: 1초마다 LED를 ON/OFF하기
- 실습 2** 디지털 입력으로 디지털 출력: 버튼을 누르면 LED를 ON하기
- 실습 3** 아날로그 입력으로 디지털 출력: 일정 밝기 이하에서만 LED ON하기
- 실습 4** 아날로그 입력으로 아날로그(PWM) 출력: 밝기에 따라 비례해서 LED를 ON하기

02

자동차 ADAS(첨단 운전자 보조 시스템) 모방하기

- 실습 1** 자동 환풍기 만들기
- 실습 2** 자동 전조등 만들기
- 실습 3** 자동 브레이크 만들기
- 실습 4** 실습1~실습3을 모두 합쳐 실행하는 코드 만들기

03

다양한 문제 상황 해결하기

- 실습 1** 좌우 깜빡이 만들기
- 실습 2** 좌우 깜빡이 토글(toggle) 버튼 만들기
- 실습 3** 자동 가로등 만들기
- 실습 4** 시리얼 통신 출력하기
- 실습 5** PWM 출력하기
- 실습 6** PWM 출력(서보모터 제어)하기
- 실습 7** PWM 출력(서보모터 제어)_Servo.h 라이브러리 사용
- 실습 8** 아날로그 가변 저항을 사용한 모터 회전 속도 제어하기
- 실습 9** 아두이노 초음파 레이다 만들기 + 측정 결과를 프로세싱으로 비주얼로 보여 주기

01

아두이노 연결 및 소스 코드 만들기

* 문제 상황에 맞는 회도로 구성 및 소스 코드를 만들고 직접 구현해 보자.

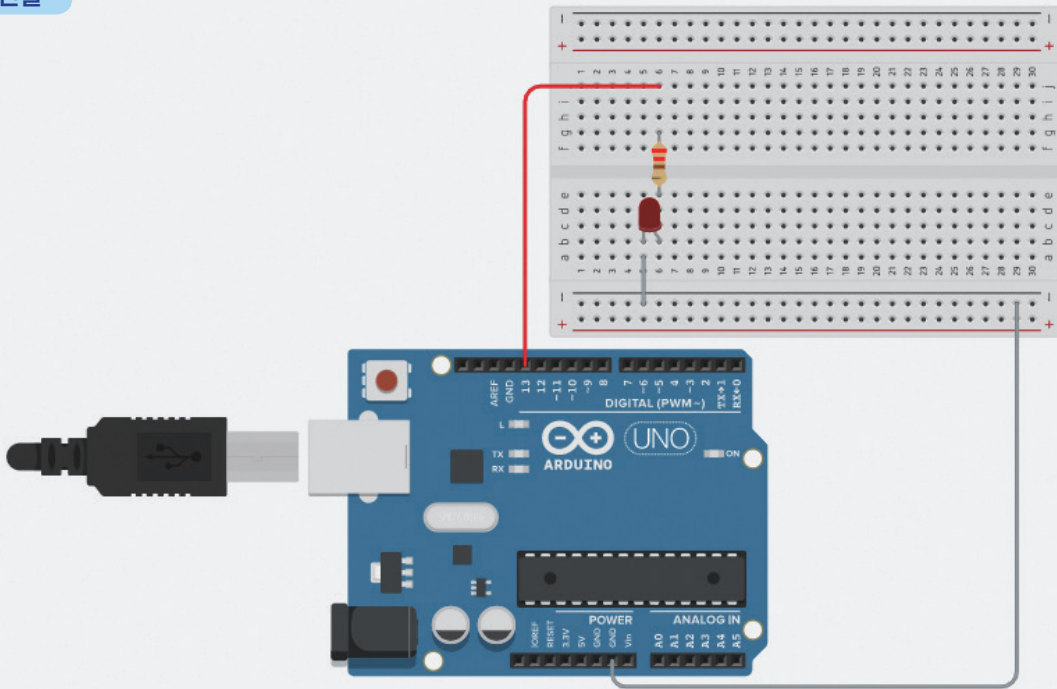
실습1 디지털 출력: 1초마다 LED를 ON/OFF하기

<https://www.tinkercad.com/things/5bZ6ZKZRih>

여기를 클릭하여 시뮬레이션을 누르면 연결 및 실제 동작하는 모습과 소스 코드를 볼 수 있습니다.

(※ AUTODESK TINKERCAD에 가입하면 회로도와 소스 코드를 복사 및 변경할 수 있습니다.)

연결



▶ 소스 코드 ◀

```
void setup( ) // 전원 공급 시 1회만 실행
{
  pinMode(13, OUTPUT); // 13번 디지털 포트를 출력으로 설정
}

void loop( ) // 무한 반복 실행
{
  digitalWrite(13, HIGH); // 13번 디지털 포트에 HIGH(ON) 출력
  delay(1000); // 현재 상태로 1000ms(1초) 유지
  digitalWrite(13, LOW); // 13번 디지털 포트에 LOW(OFF) 출력
  delay(1000); // 현재 상태로 1000ms(1초) 유지
}
```

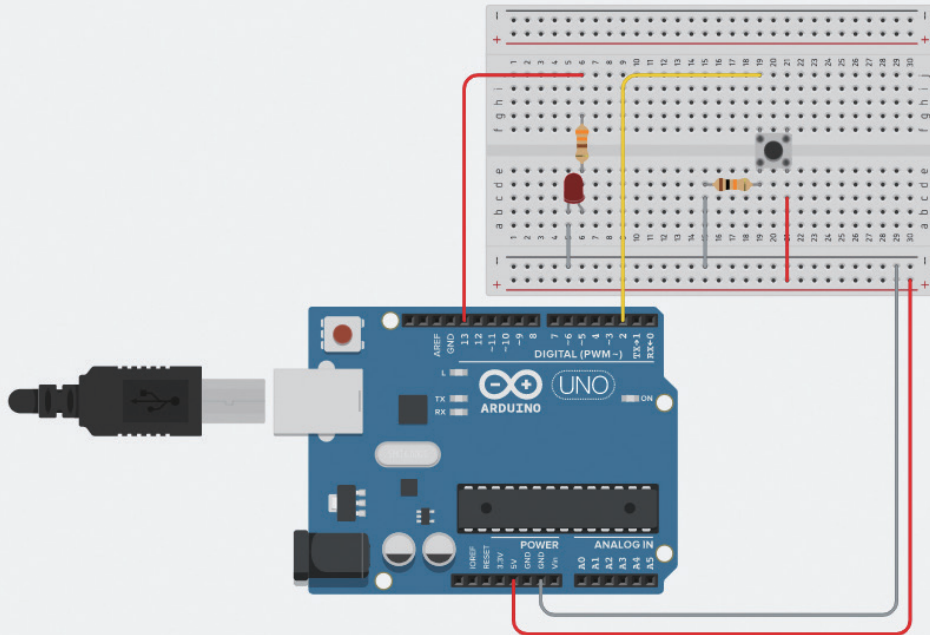
실습2

디지털 입력으로 디지털 출력: 버튼을 누르면 LED를 ON하기

<https://www.tinkercad.com/things/9bpTeVRkrTi>

시뮬레이션을 클릭하여 센서를 클릭하면 센서에 입력되는 누름, 온도, 빛, 거리 등을 마우스로 조절할 수 있습니다.

연결



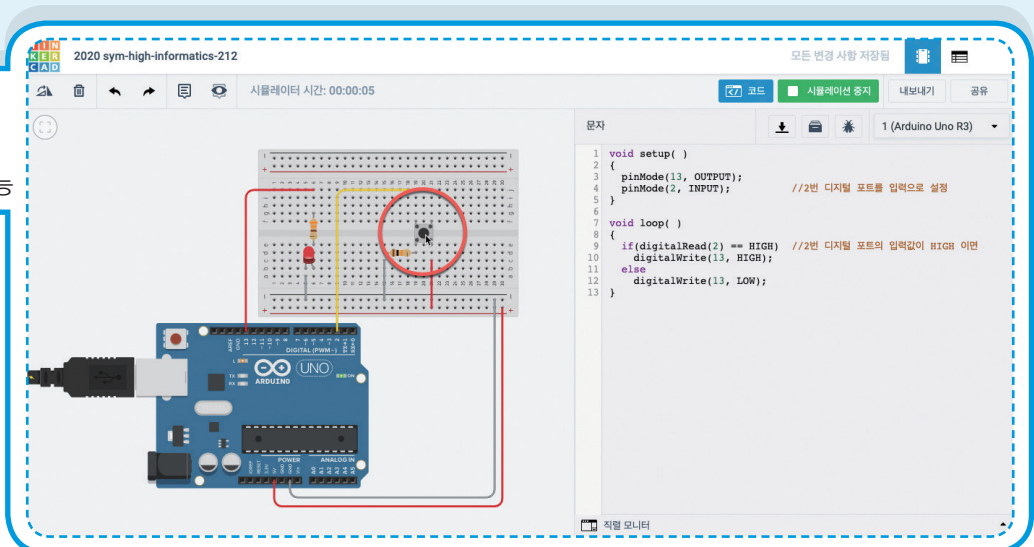
소스 코드

```
void setup( )
{
  pinMode(13, OUTPUT);
  pinMode(2, INPUT);           // 2번 디지털 포트를 입력으로 설정
}

void loop( )
{
  if(digitalRead(2) == HIGH)   // 2번 디지털 포트의 입력 값이 HIGH이면
    digitalWrite(13, HIGH);   // 13번 포트에 전원 ON
  else                          // 아니면
    digitalWrite(13, LOW);    // 13번 포트에 전원 OFF
}
```

시뮬레이션 실행 화면

- 시뮬레이션 실행 후
- 버튼을 마우스로 클릭 가능

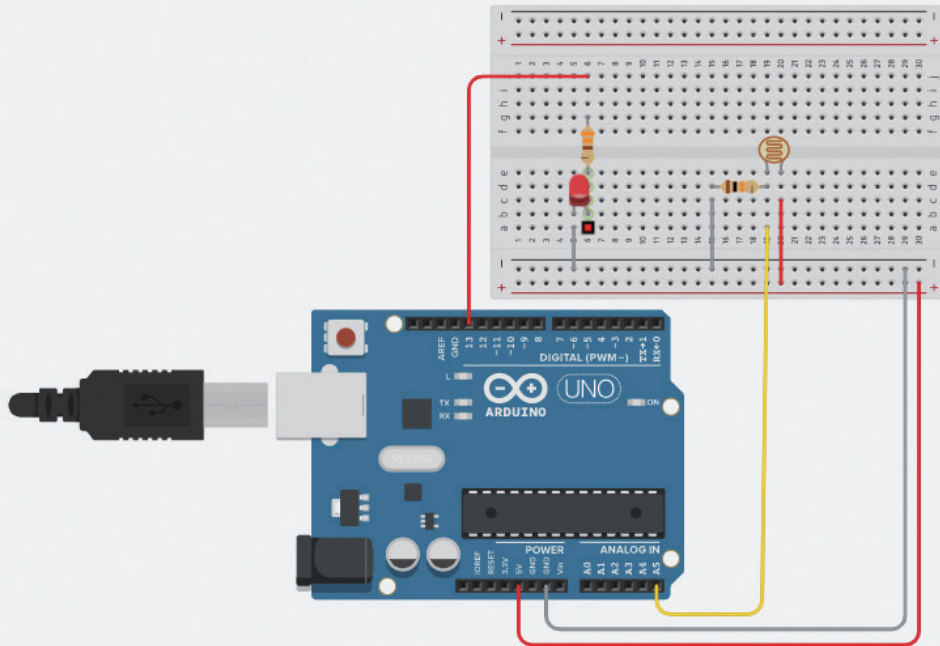


실습 3

아날로그 입력으로 디지털 출력: 일정 밝기 이하에서만 LED ON하기

<https://www.tinkercad.com/things/98wrvAA6ux1>

연결



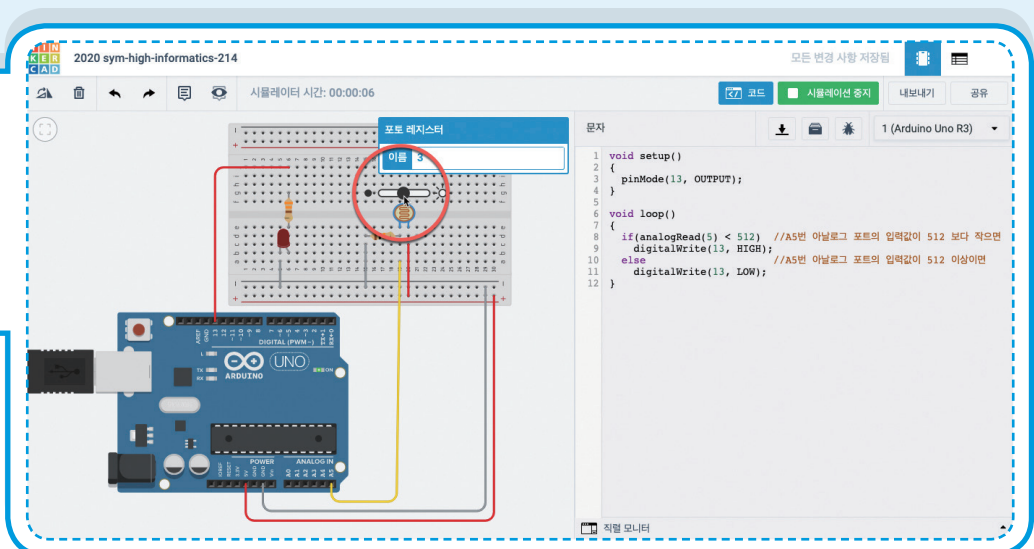
소스 코드

```
void setup( )
{
  pinMode(13, OUTPUT);
}

void loop( )
{
  if(analogRead(5) < 512)           // A5번 아날로그 포트의 입력값이 512보다 작으면
    digitalWrite(13, HIGH);
  else                               // A5번 아날로그 포트의 입력값이 512 이상이면
    digitalWrite(13, LOW);
}
```

시뮬레이션 실행 화면

- 시뮬레이션 실행 후
- 포토 레지스터를 클릭과 드래그할 수 있음.
- 센서류는 모두 비슷한 방법으로 클릭, 드래그 형태로 시뮬레이션이 가능.

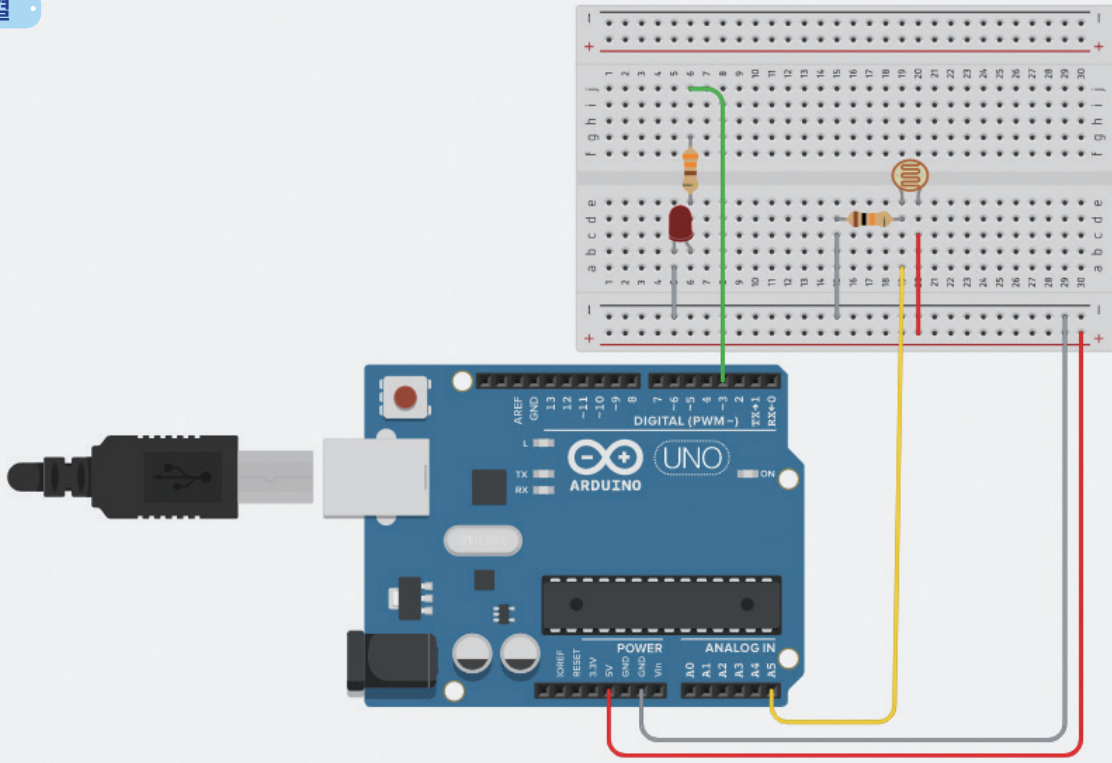


실습 4

아날로그 입력으로 아날로그(PWM) 출력: 밝기에 따라 비례해서 LED를 ON하기

<https://www.tinkercad.com/things/eNze7SkbPBx>

연결



▶ 소스 코드 ◀

```
void setup( )
{
  pinMode(3, OUTPUT);
}

void loop( )
{
  analogWrite(3, (1023-analogRead(5))/4); // 3번 디지털 포트에 PWM 모드로 출력
}
```

02

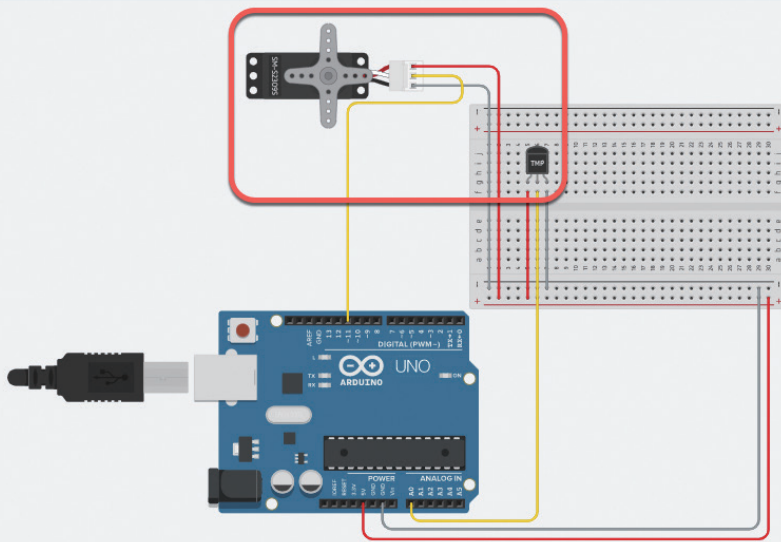
자동차 ADAS(첨단 운전자 보조 시스템) 모방하기

실습1 자동 환풍기 만들기

아날로그 입력으로 PWM 출력: 온도(온도 트랜지스터)에 따라 비례해서 환풍기(서보모터) 열기

<https://www.tinkercad.com/things/2JXYB1jsNoA>

연결



▶ 소스 코드 ◀

```
//서보모터 제어를 위한 라이브러리 참조
#include <Servo.h>

Servo servo1; // servo1 생성

void setup( )
{
  servo1.attach(11); // 11번 PWM 포트에 Servo1 연결
}

void ta( ) // 온도에 따른 서보모터 회전각 조절 함수
{
  int t, ta; float c;
  t = analogRead(0); // 온도 센서값 읽기
  c = (5.0*t/1024-0.5)*100; // 섭씨온도로 변환
  if(c<=25) ta = 0; // 25도 이하이면 서보모터 출력값을 0으로 설정
  else if(c>34) ta = 180; // 34도 초과이면 서보모터 출력값을 180으로 설정
  else ta = (c-24)/(34-24)*180; // 온도에 비례하여 서보모터 출력값 설정
  servo1.write(ta); // 서보모터 출력
}

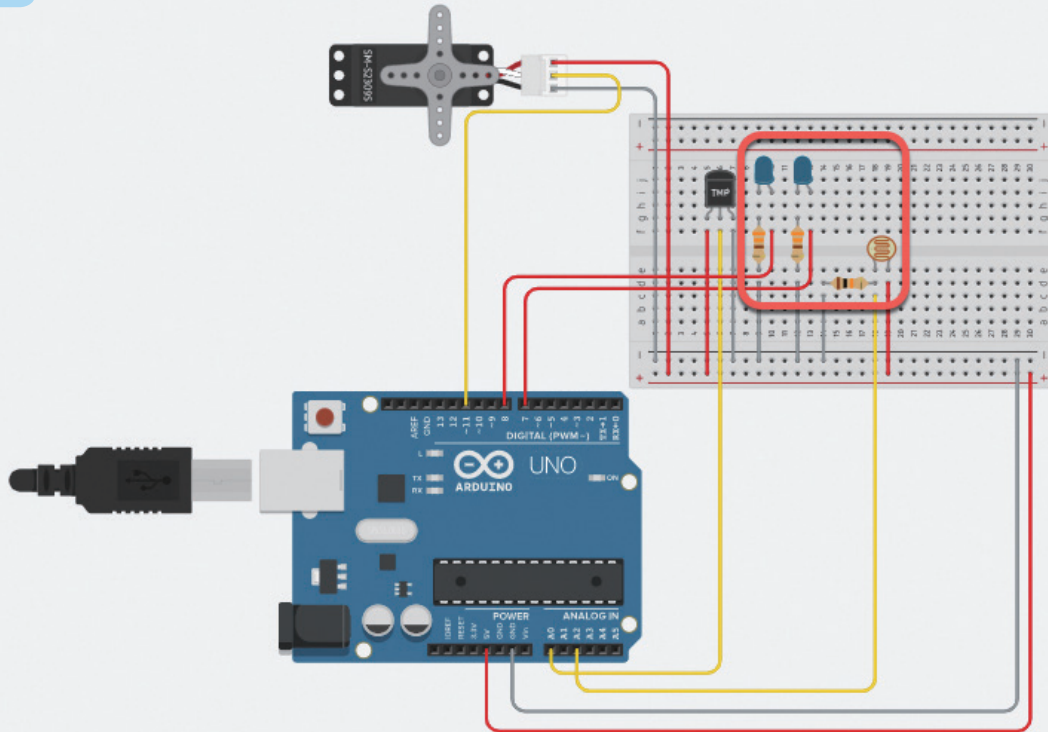
void loop( )
{
  ta( );
}
```

실습2 자동 전조등 만들기

아날로그 입력으로 디지털 출력: 빛 밝기(포토 레지스터)에 따라 어느 정도 어두워지면 전조등(LED) 켜기

<https://www.tinkercad.com/things/k0JzAbHuqkR>

연결



▶ 소스 코드 ◀

```
void setup( )
{
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
}

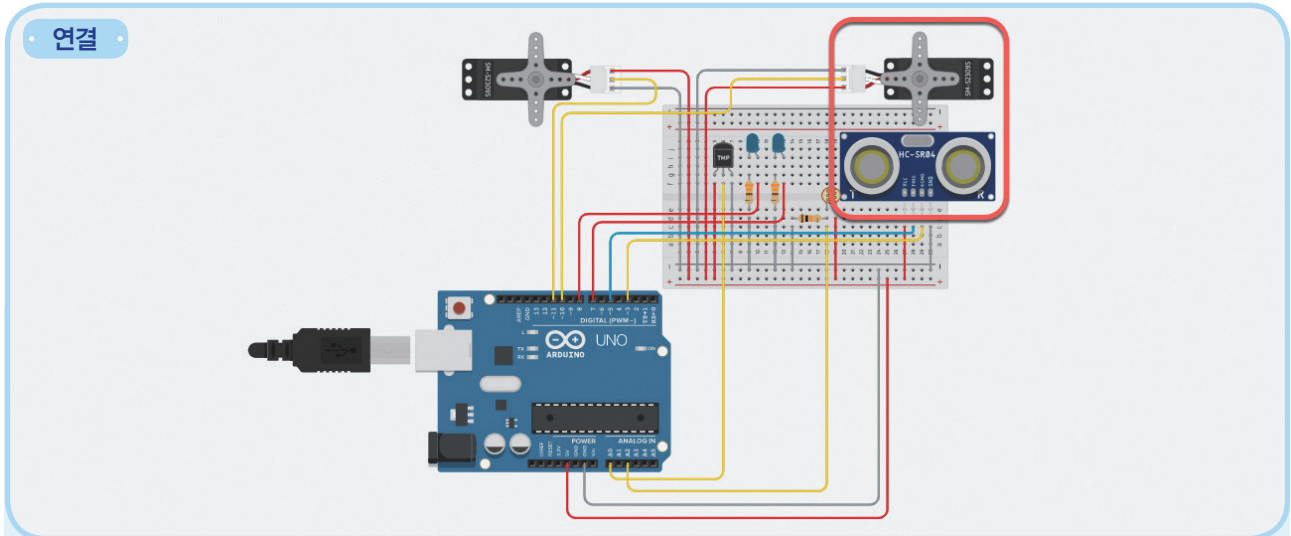
void light( )
{
  if(analogRead(2) < 30.0/100*1024)           // 빛 센서의 입력값이 30% 미만이면
  {
    digitalWrite(8, HIGH);                   // 왼쪽 LED ON
    digitalWrite(7, HIGH);                   // 오른쪽 LED ON
  }
  else
  {
    digitalWrite(8, LOW);                    // 아니면
    digitalWrite(7, LOW);                    // 왼쪽 LED OFF
    digitalWrite(7, LOW);                    // 오른쪽 LED OFF
  }
}

void loop( )
{
  light( );
}
```

실습3 자동 브레이크 만들기

디지털 입력으로 디지털(PWM) 출력: 앞 차와의 거리(초음파 센서)에 따라 비례해서 브레이크(서보모터) 밝기

<https://www.tinkercad.com/things/9Uk3ZZNTJJ7>



연결

소스 코드

```
#include <Servo.h>
Servo servo2;

int dist( )
{
    long duratiON;
    int cm;
    pinMode(5, OUTPUT);
    pinMode(3, INPUT);

    digitalWrite(5, LOW); delayMicrosecONds(2);
    digitalWrite(5, HIGH); delayMicrosecONds(10);
    digitalWrite(5, LOW);

    duratiON = pulseIn(3, HIGH);
    cm = (duratiON*0.0343)/2;
    return cm;
}
```

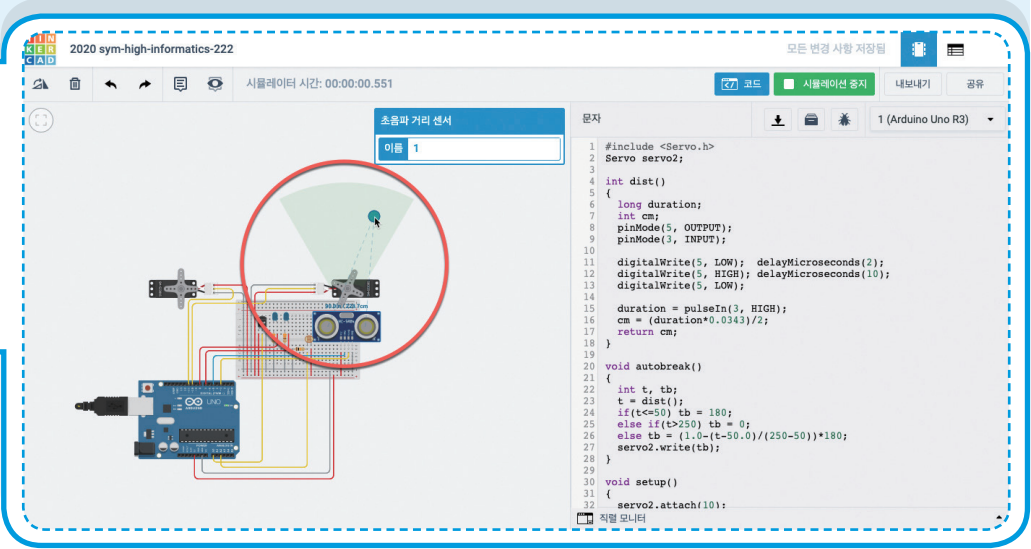
```
void autobreak( )
{
    int t, tb;
    t = dist( );
    if(t<=50) tb = 180;
    else if(t>250) tb = 0;
    else tb = (1.0-(t-50.0)/(250-50))*180;
    servo2.write(tb);
}

void setup( )
{
    servo2.attach(10);
}

void loop( )
{
    autobreak( );
}
```

시뮬레이션 실행 화면

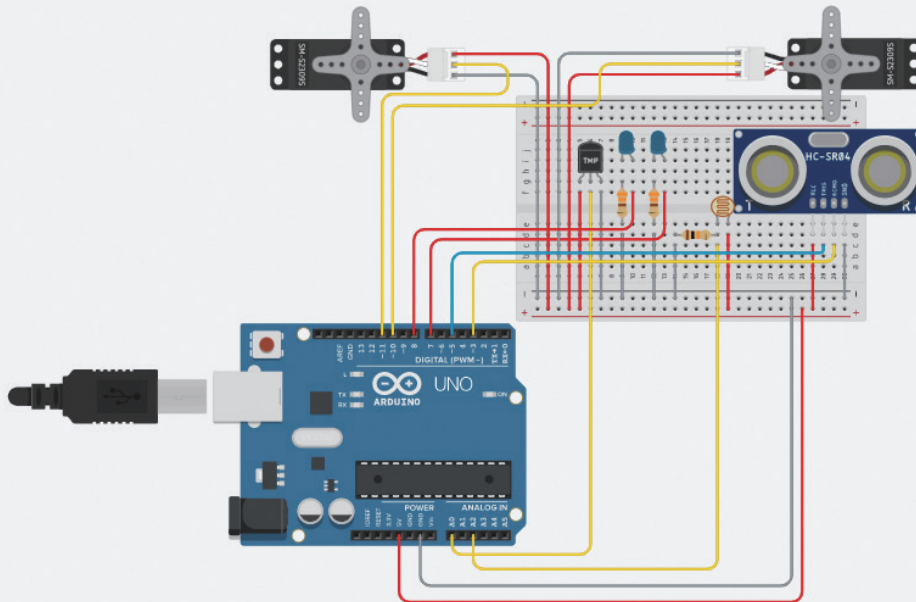
- 시뮬레이션 실행 후
- 초음파 센서를 클릭하여 가상의 물체를 센서 앞에 놓을 수 있음.
- 센서류는 모두 비슷한 방법으로 클릭, 드래그 형태로 시뮬레이션 가능



실습 4 실습 1~실습 3을 모두 합쳐 실행하는 코드 만들기

<https://www.tinkercad.com/things/ag5e2b8ufBT>

연결



소스 코드

```
#include <Servo.h>
Servo servo1; Servo servo2;

void ta( )
{
  int t, ta; float c;
  t = analogRead(0);
  c = (5.0*t/1024-0.5)*100;
  if(c<=25) ta = 0;
  else if(c>34) ta = 180;
  else ta = (c-24)/(34-24)*180;
  servo1.write(ta);
}

void light()
{
  if(analogRead(2) < 30.0/100*1024)
  {
    digitalWrite(8, HIGH);
    digitalWrite(7, HIGH);
  }
  else
  {
    digitalWrite(8, LOW);
    digitalWrite(7, LOW);
  }
}

int dist()
{
  long duratiON;
  int cm;
  pinMode(5, OUTPUT);
  pinMode(3, INPUT);

  digitalWrite(5, LOW); delayMicrosecONds(2);
  digitalWrite(5, HIGH); delayMicrosecONds(10);
  digitalWrite(5, LOW);

  duratiON = pulseIn(3, HIGH);
  cm = (duratiON*0.0343)/2;
  return cm;
}

void autobreak( )
{
  int t, tb;
  t = dist();
  if(t<=50) tb = 180;
  else if(t>250) tb = 0;
  else tb = (1.0-(t-50.0)/(250-50))*180;
  servo2.write(tb);
}

void setup( )
{
  servo1.attach(11);

  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);

  servo2.attach(10);
}

void loop( )
{
  ta( );
  light( );
  autobreak( );
}
```

03

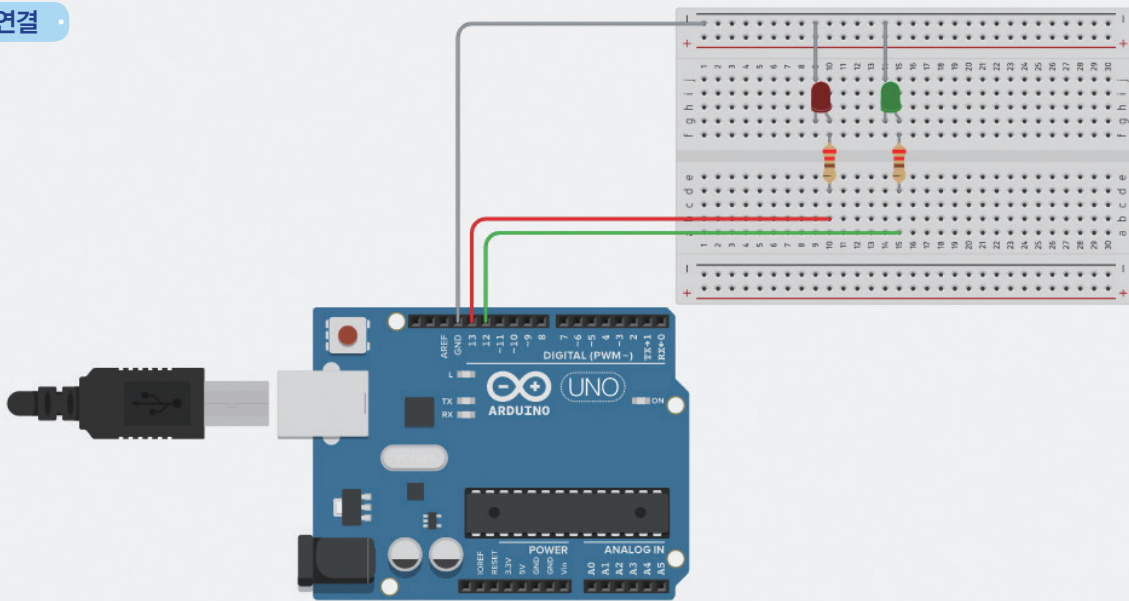
다양한 문제 상황 해결하기

실습1 좌우 깜박이 만들기

디지털 출력: 좌우를 번갈아가며 바뀌는 깜박이 만들기(경찰차)

<https://www.tinkercad.com/things/9RwAdIG68Mu>

연결



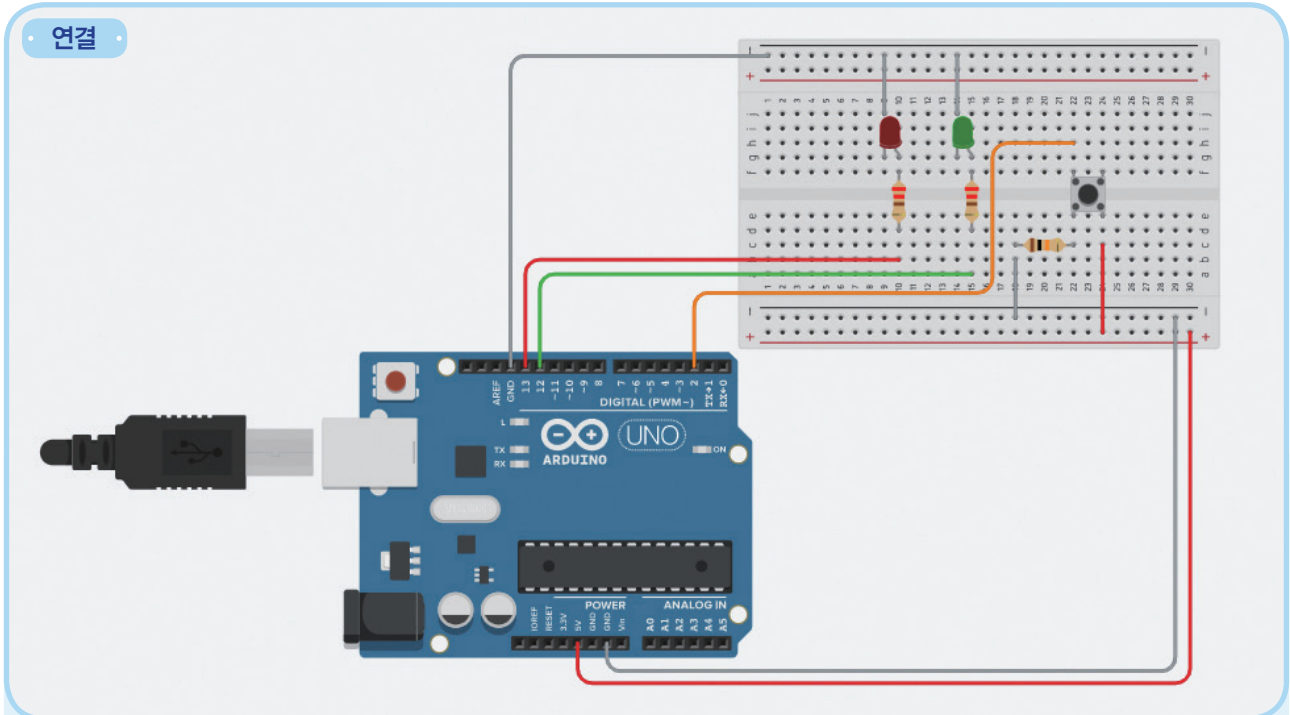
소스 코드

```
void setup( )  
{  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
}  
  
void loop( )  
{  
  digitalWrite(13, HIGH);  
  digitalWrite(12, LOW);  
  delay(1000);  
  
  digitalWrite(13, LOW);  
  digitalWrite(12, HIGH);  
  delay(1000);  
}
```

실습2 좌우 깜박이 토글(toggle) 버튼 만들기

디지털 입력으로 깜박이 토글 만들기: 버튼을 한 번씩 누를 때마다 상태가 토글(깜박)되는 토글 버튼 구현

<https://www.tinkercad.com/things/eTWBDd5Uxay>



▶ 소스 코드 ◀

```
int t;
void setup( )
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(2, INPUT);
  t = 1;
}

void loop( )
{
  if(t == 1)
  {
    digitalWrite(13, HIGH); digitalWrite(12, LOW);
    delay(100);
    digitalWrite(13, LOW); digitalWrite(12, HIGH);
    delay(100);
  }

  if(digitalRead(2) == HIGH)
  {
    digitalWrite(13, LOW); digitalWrite(12, LOW);
    t = (t+1)%2;

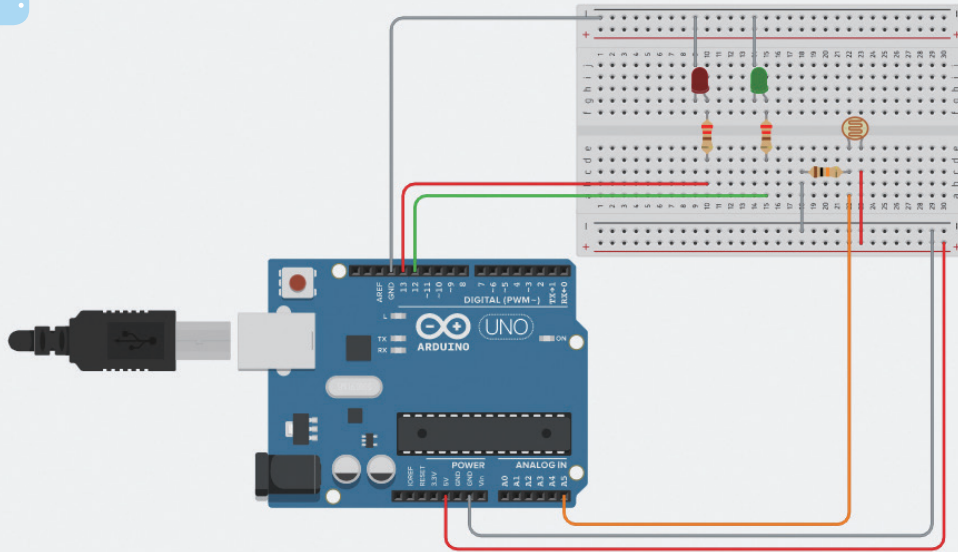
    while(digitalRead(2) == HIGH);
  }
}
```

실습3 자동 가로등 만들기

아날로그 입력으로 디지털 출력: 일정 밝기 이하에서 점등하기

<https://www.tinkercad.com/things/c3ksb6haFrh>

연결



소스 코드

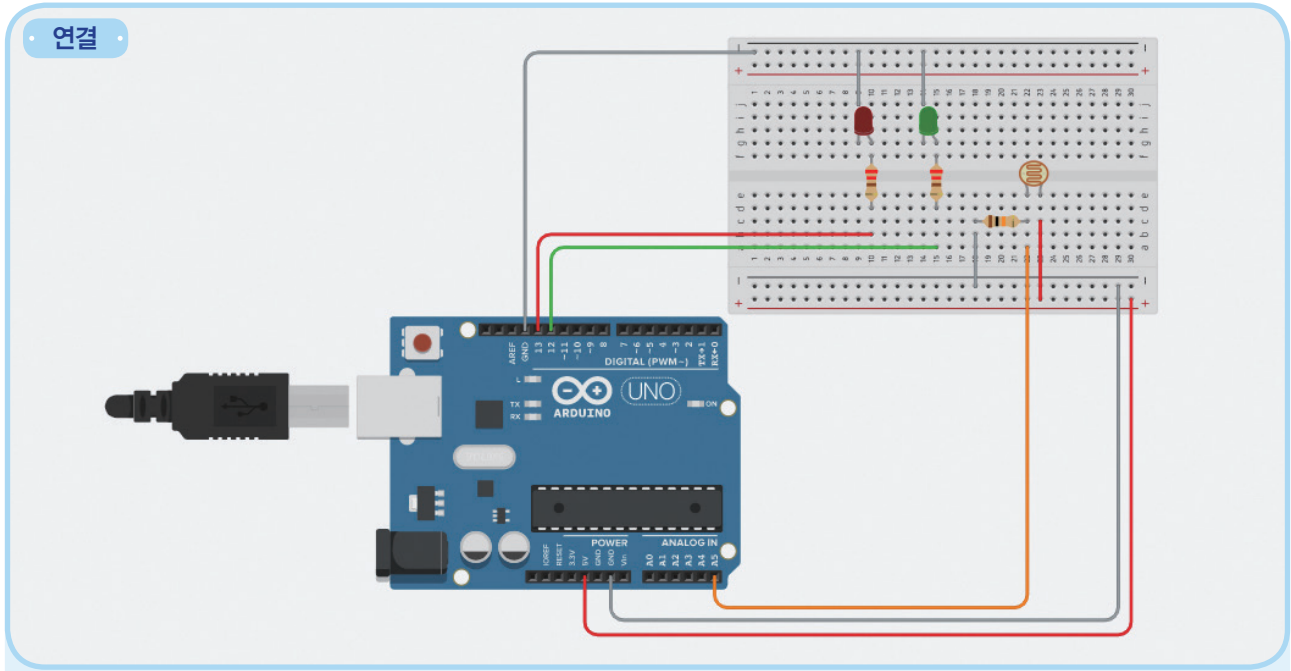
```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(A5, INPUT);
}

void loop()
{
  if(analogRead(A5) < 512)
  {
    digitalWrite(13, LOW); digitalWrite(12, HIGH);
  }
  else
  {
    digitalWrite(13, HIGH); digitalWrite(12, LOW);
  }
}
```

실습 4 시리얼 통신 출력하기

아날로그 입력으로 '디지털 출력 + 시리얼 통신 출력': 일정 밝기 이하에서 점등하고, 밝기 값을 시리얼 통신으로 전송하기

<https://www.tinkercad.com/things/9n9LfWtv6Gm>



연결

▶ 소스 코드 ◀

```

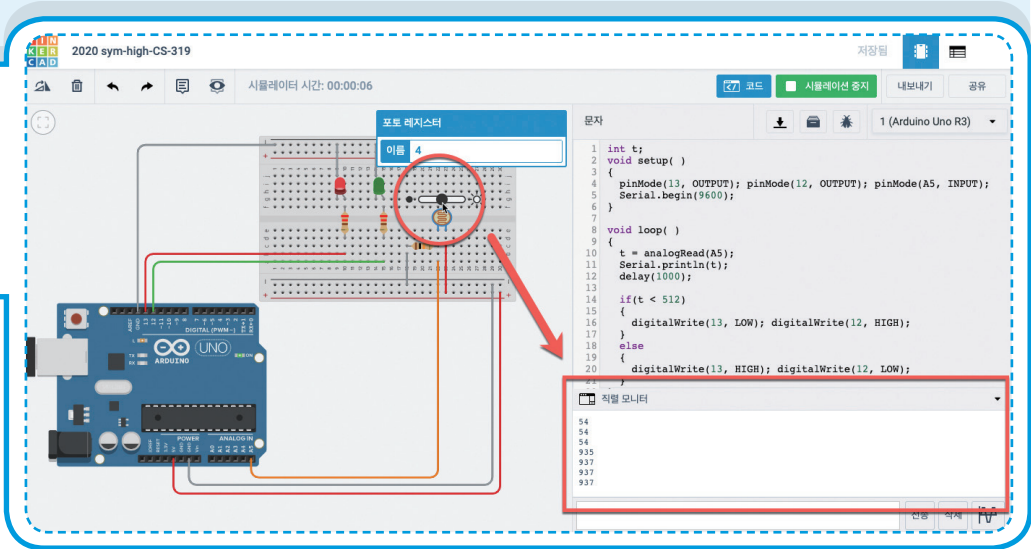
int t;
void setup( )
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(A5, INPUT);
  Serial.begin(9600);
}

void loop( )
{
  t = analogRead(A5);
  Serial.println(t); //시리얼 통신으로 값 출력하고 줄 바꿈
  delay(1000);

  if(t < 512)
  {
    digitalWrite(13, LOW); digitalWrite(12, HIGH);
  }
  else
  {
    digitalWrite(13, HIGH);
    digitalWrite(12, LOW);
  }
}
    
```

시뮬레이션 실행 화면

- 시뮬레이션 실행 후
- 오른쪽 아래 직렬(시리얼) 모니터 창
- 시리얼 통신 출력 값들을 볼 수 있음.



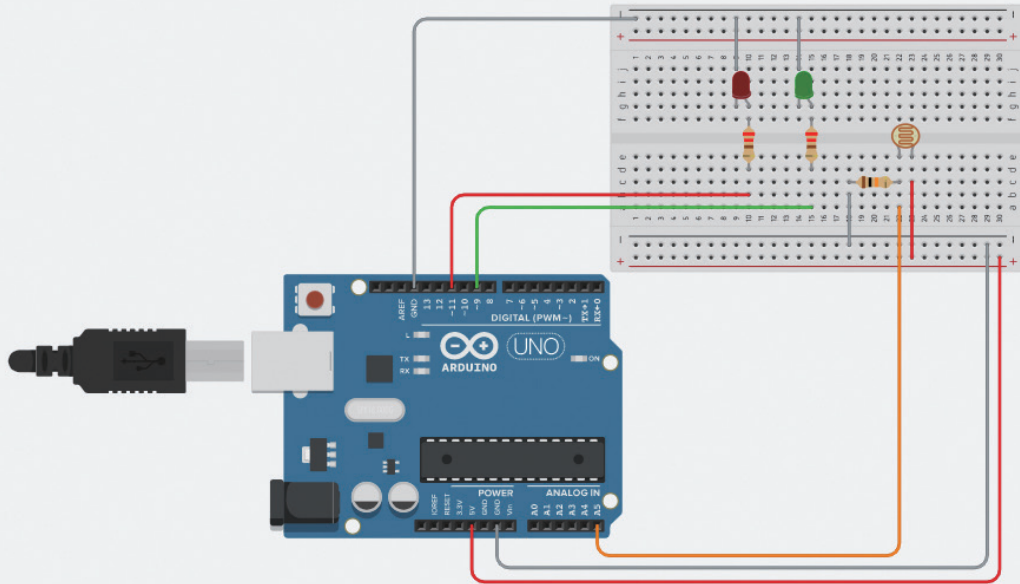
실습 5

PWM 출력하기

아날로그 입력으로 PWM(아날로그 모방) 출력: 밝기에 따라 LED 밝기를 변화 주기

<https://www.tinkercad.com/things/j0ftkxTVJbh>

연결



▶ 소스 코드 ◀

```
int t;
void setup( )
{
  pinMode(A5, INPUT);
}

void loop( )
{
  t = analogRead(A5);

  analogWrite(11, t/4);
  analogWrite(9, t/4);
}
```

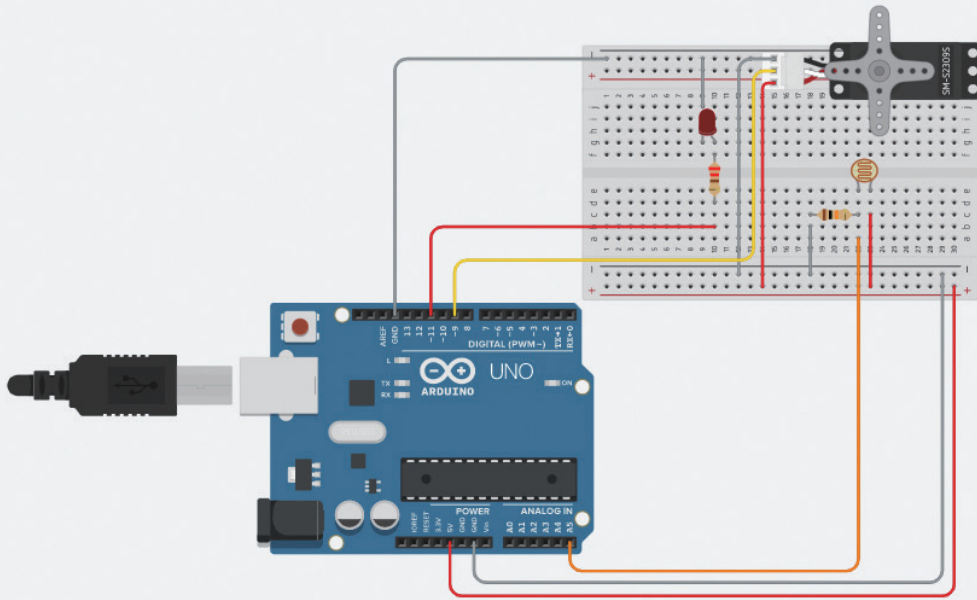
// analogWrite() 함수로 출력하면 기본으로 PWM 출력됨.

실습 6 PWM 출력(서보모터 제어)하기

아날로그 입력으로 PWM 출력 서보모터 제어: 밝기에 따라 서보모터의 회전 각도를 바꿈(무한 회전 모터의 경우에는 회전 속도도 제어 가능).

<https://www.tinkercad.com/things/gHYGMMGHJf0>

연결



▶ 소스 코드 ◀

```
int t;
void setup( )
{
  pinMode(A5, INPUT);
}

void loop()
{
  t = analogRead(A5);

  analogWrite(11, t/4);
  analogWrite(9, t/4);           // 서보모터에 연결해도 제어됨.
}
```

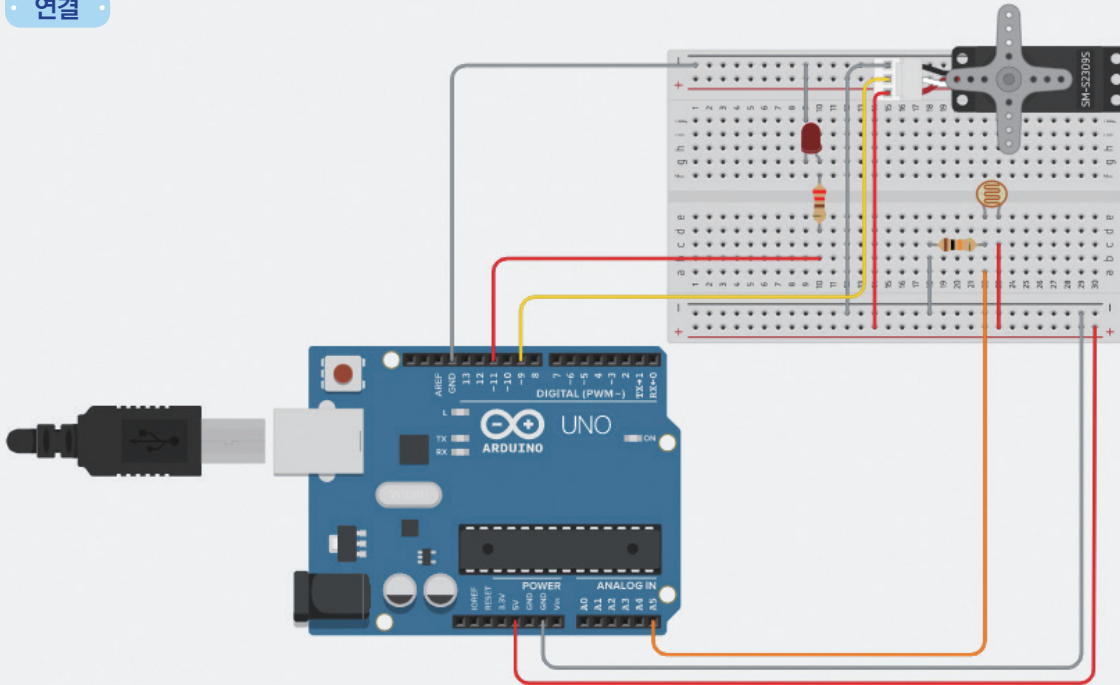
실습7

PWM 출력(서보모터 제어)_Servo.h 라이브러리 사용

아날로그 입력으로 PWM 출력 서보모터 제어: 밝기에 따라 서보모터의 회전 각도를 바꿈(무한 회전 모터의 경우에는 회전 속도도 제어 가능).

<https://www.tinkercad.com/things/04XfbStr7XN>

연결



▶ 소스 코드 ◀

```
#include <Servo.h>
Servo servo;           // 서보모터 선언
int t;
void setup()
{
  pinMode(A5, INPUT);
  servo.attach(9);     // PWM 9번 포트에 연결
}

void loop( )
{
  t = analogRead(A5);
  analogWrite(11, t/4);
  servo.write(t/4)     // 0~180도 범위로 출력 가능
}
```

실습 8

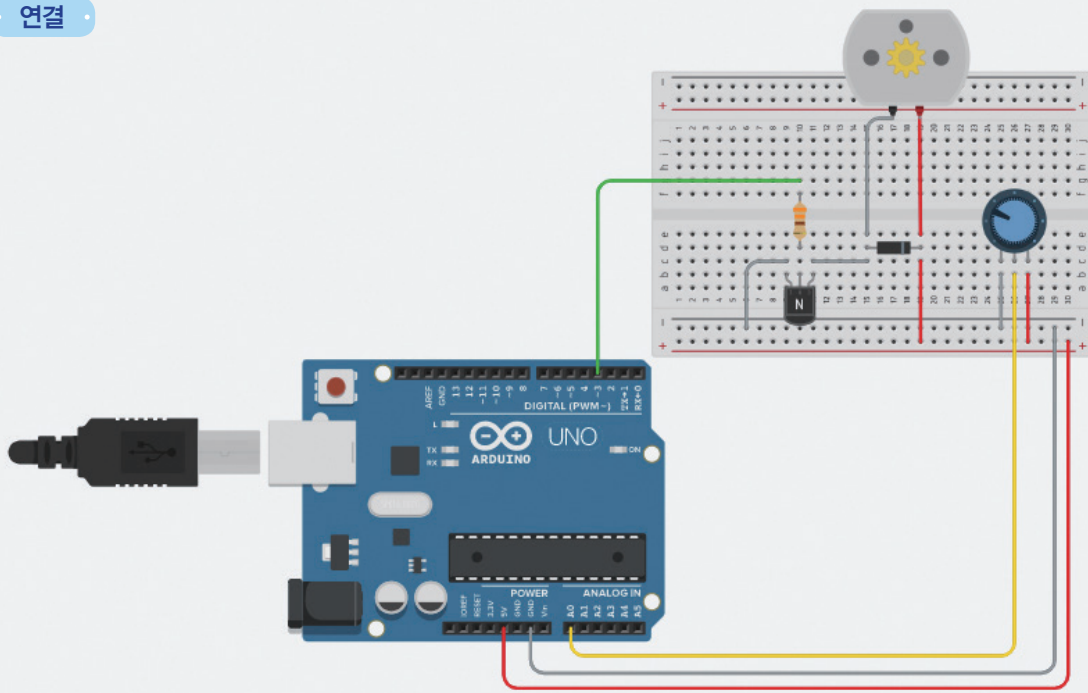
아날로그 가변 저항을 사용한 모터 회전 속도 제어하기

아날로그 입력으로 트랜지스터(증폭 작용) → 모터 제어: 아날로그적 방법을 사용한 모터 제어

- 가변 저항을 돌리면 저항 값이 바뀌면서 모터 회전 속도가 제어됨.
- 가변 저항을 직접 모터 전원부로 연결시키는 방법도 있지만, 아날로그 저항 값을 디지털로 읽은 후 다시 트랜지스터를 사용해 출력 전압 제어로 회전 속도를 제어하는 예시임.

<https://www.tinkercad.com/things/eaBZZkH8wNA>

연결



소스 코드

```
int potPin = 0;
int trBasePin = 3;           // PWM
int potValue = 0;

void setup( )
{
  pinMode(trBasePin, OUTPUT);
  Serial.begin(9600);
}

void loop( )
{
  potValue = analogRead(potPin)/4;
  Serial.println(potValue);
  analogWrite(trBasePin, potValue);
}
```

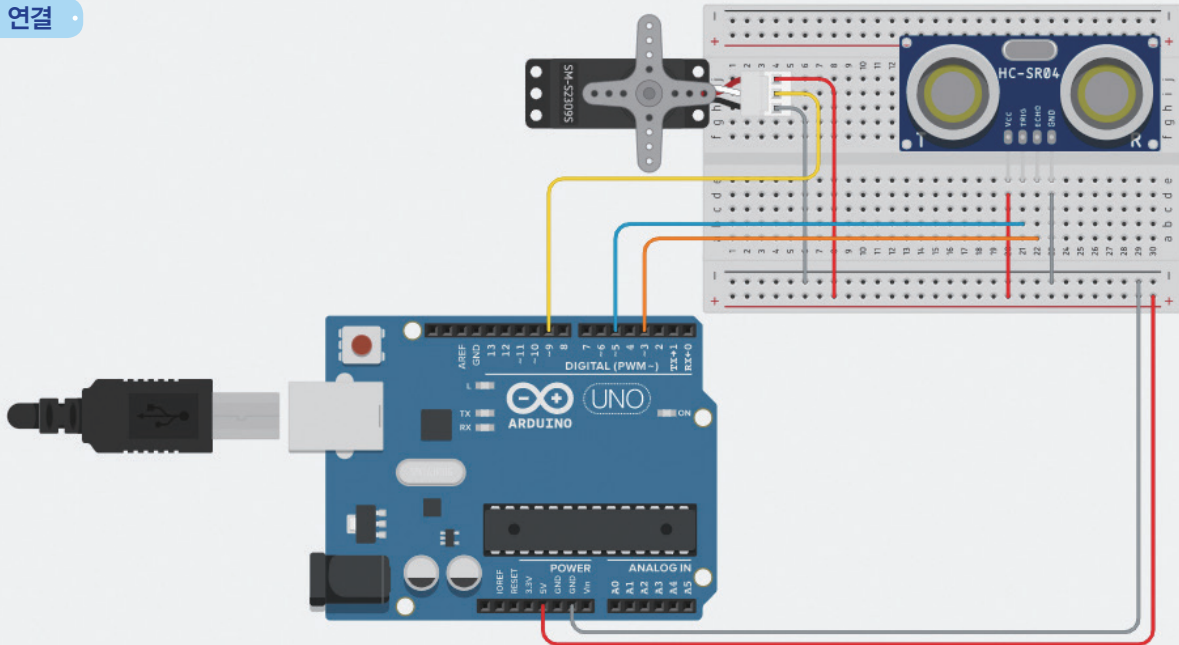
실습 9

아두이노 초음파 레이다 만들기 + 측정 결과를 프로세싱으로 비주얼로 보여 주기

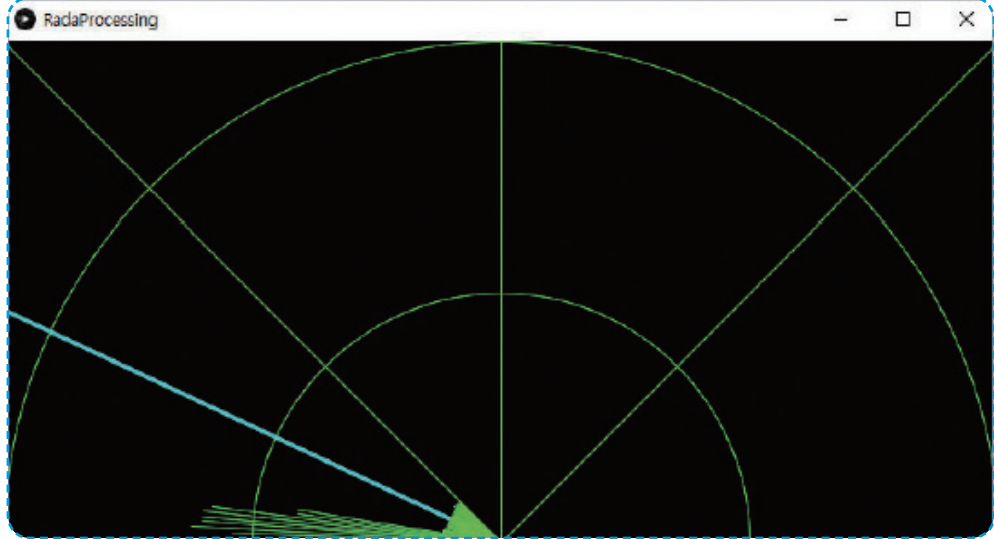
- arduino + processing 프로젝트(아두이노를 이용하여 데이터를 수집하고, 수집한 결과를 프로세싱으로 출력)
- 서보모터를 일정 각도로 좌우 회전하면서 그 위에 초음파 센서를 달아서 마치 레이다처럼 모방할 수 있음.
- 초음파 센서를 통해 입력되는 각도와 거리 값을 시리얼 통신으로 전송하기
- 시리얼 통신으로 전송된 각도와 거리 값을 프로세싱으로 화면에 뿌려 주기

<https://www.tinkercad.com/things/1fRTJkgNTII>

연결



시뮬레이션 실행 화면



▶ 소스 코드 ◀

```
#include <Servo.h>
Servo servo;

int dist()
{
    long duratiON;
    int cm;
    pinMode(5, OUTPUT);
    pinMode(3, INPUT);
    digitalWrite(5, LOW); delayMicrosecONds(2);
    digitalWrite(5, HIGH); delayMicrosecONds(10);
    digitalWrite(5, LOW);
    duratiON = pulseIn(3, HIGH);
    cm = (duratiON*0.0343)/2;
    return cm;
}

void send(int pos, int dist)
{
    Serial.print(pos); Serial.print(",");
    Serial.print(dist); Serial.print('\n');
    delay(30);
}

void setup()
{
    Serial.begin(9600);
    servo.attach(9);
}

void loop()
{
    for(int pos=0; pos<90; pos+=1)
    {
        if(pos <= 45)
            servo.write(pos);
        else
            servo.write(90 - pos);
        send(pos, dist());
    }
}
```

▶ 프로세싱 코드 ◀

```
import processing.serial.*;

Serial P; //The serial port
String S=null; //Input string from serial port
int pos, dist, res; //positiON, distance, resolutiON
int[] DIST = new int[45];

void setup( )
{
    size(720, 360);

    surface.setResizable(true);
    ellipseMode(RADIUS);
    ellipseMode(CENTER);

    strokeWeight(1);
    stroke(255, 0, 0);

    printArray(Serial.list());

    P = new Serial(this, Serial.list()[0], 9600);
    P.bufferUntil('\n');

    pos = 0; dist = 0; res = 150;
}

void draw()
{
    int i = 0;
    float xt, yt;
    background(0);

    pushMatrix( );

    translate(width/2, height);
    drawRadar( );

    stroke(0, 255, 255);
    strokeWeight(3);
    xt = -max(width,height)*cos(radians(pos));
    yt = -max(width,height)*sin(radians(pos));

    line(0, 0, xt, yt);

    stroke(0, 255, 0);
    strokeWeight(1);

    for(int j: DIST)
    {
        xt = -(j*(width/res/2))*cos(radians(i));
        yt = -(j*(height/res))*sin(radians(i));
        line(0, 0, xt, yt);
        i++;
    }

    popMatrix();
}

void serialEvent(Serial p)
{
    S = p.readStringUntil('\n');

    if(S! = null)
    {
        S = trim(S);
        String[] list = split(S, ',');
        pos = int(list[0]);
        dist = int(list[1]);

        if(pos > 44)
            pos = 89-pos;

        DIST[pos] = dist;
    }
}

void drawRadar()
{
    stroke(0, 255, 0);

    line(0, 0, -width/2, -height);
    line(0, 0, 0, -height);
    line(0, 0, width/2, -height);

    noFill();

    arc(0, 0, width/2, height, PI, PI+PI);
    arc(0, 0, width, height*2, PI, PI+PI);
}
```


고등학교
정보·정보과학
아두이노
핵심 코드집

저 자 | 전현석

발 행 인 | 신재석

발 행 처 | (주)삼양미디어

등록번호 | 제10-2285호

주 소 | 서울시 마포구 양화로 6길 9-28

전 화 | 02-335-3030

팩 스 | 02-335-2070

홈페이지 | www.samyangm.com

*이 책은 저작권법에 따라 보호받는 저작물이므로 무단전재와 복제를 금합니다.

고등학교
정보·정보과학
아두이노
핵심 코드집

