



15주차 2차시

컴퓨팅사고력과 코딩

초음파센서 (HC-SR04) 활용하기



동양미래대학교



동의과학대학교



영진전문대학교
YEUNGJIN UNIVERSITY



전주비전대학교



충북보건과학대학교
CHUNGBUK HEALTH & SCIENCE UNIVERSITY

학습목표

- 초음파센서(HC-SR04)를 이용하여 실제 물체를 감지하는 프로그램을 작성할 수 있다.
- 초음파센서(HC-SR04)를 이용하여 범위 이내에 물체가 감지되면 경고음이 울리도록 제어하는 프로그램을 작성 및 실행할 수 있다.
- 서보모터를 이용하여 범위 이내에 물체가 감지되면 자동으로 모터가 90° 움직이도록 제어하는 프로그램을 작성 및 실행할 수 있다.

학습목차

1. [실습예제]
물체 감지 시뮬레이터

2. [실습예제]
주차 차단기 제어 시뮬레이터

생각하기

[실습예제] 물체 감지 시뮬레이터

[문제상황]

- 초음파센서를 이용해서 이동하는 방향에 물체가 있는지 감지하여 경고음을 알려주거나 이동방향을 결정한다.
초음파 센서와 물체와의 거리가 제어거리(실습에서는 30cm) 이내일 경우엔 **부저가 울리도록** 하려면 어떻게 해야 할까?
- 아두이노에 초음파센서를 연결하여 하드웨어를 구성한 다음 물체 유무에 따라 변화 하는 **초음파센서의 감지 값을 읽어**들인다. 그리고 해당 값을 거리로 변환한 후 **30(cm) 보다 작으면 피에조 부저가 울리도록** 프로그램을 작성하자.



문제정의 및 분해

1. 문제정의

- 아두이노 보드에 초음파센서 1개와 피에조 부저를 연결한 후 주변에 물체가 있는지 없는지를 감지하고 물체와의 거리가 일정범위 이내면 부저가 울리도록 하는 시뮬레이터 작성

2. 문제분해

- 아두이노 보드에 초음파센서, 피에조 부저를 연결하여 하드웨어 구성하기
- 틴커카드 서킷을 이용하여 초음파센서로부터 주변 물체를 감지하고 물체와의 거리가 일정 범위(여기에서는 30cm) 이내면 피에조 부저가 울리는 시뮬레이터 구현
 - 초음파센서 감지 값 읽기, 물체 유무 파악, 거리 값 변환, 피에조 부저 울리기, 1초 대기

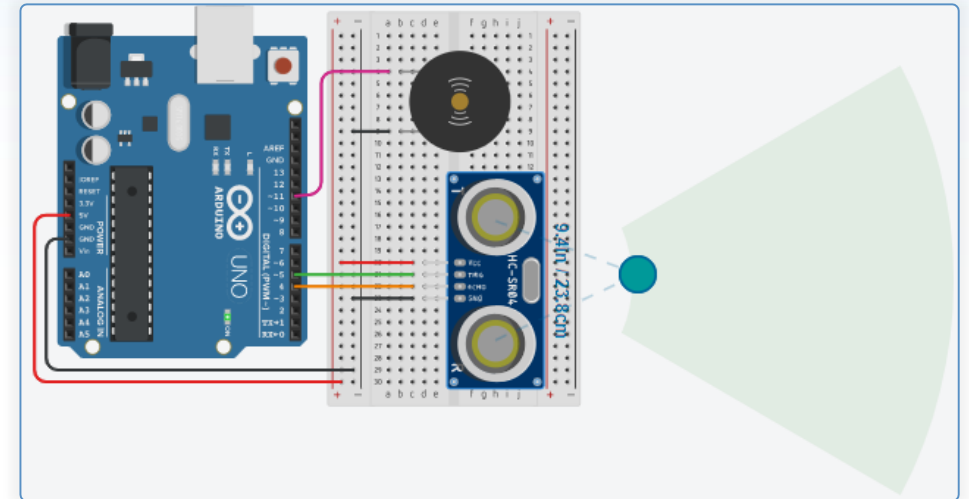
1. 실습예제 - 물체 감지 시뮬레이터 ❖❖❖ 생각하기



알고리즘

• 아래 명령어 반복하기

- ▶ 초음파센서 감지 값 읽기
- ▶ 물체 유무 파악
- ▶ 거리 값 변환
- ▶ 만일 거리가 30cm 이내라면 피에조 부저 울리기
- ▶ 1초 대기



코딩하기

[실습] 물체 감지 시뮬레이터

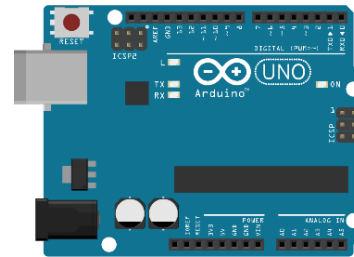
▶ 결과영상 (아두이노 활용) 미리보기



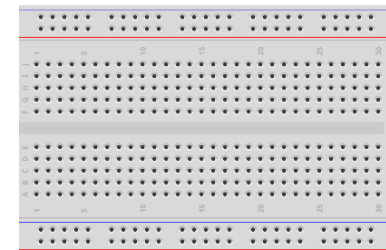
[사전준비]

- 아두이노 보드
- 브레드 보드
- 초음파센서(HC-SR04)
- 피에조 부저

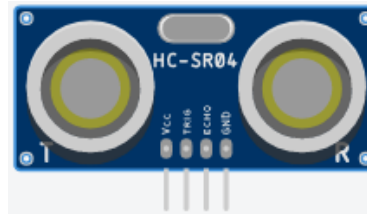
실습에 필요한 부품



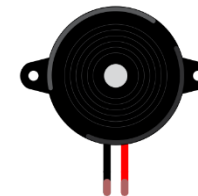
아두이노(UNO) 보드



브레드 보드



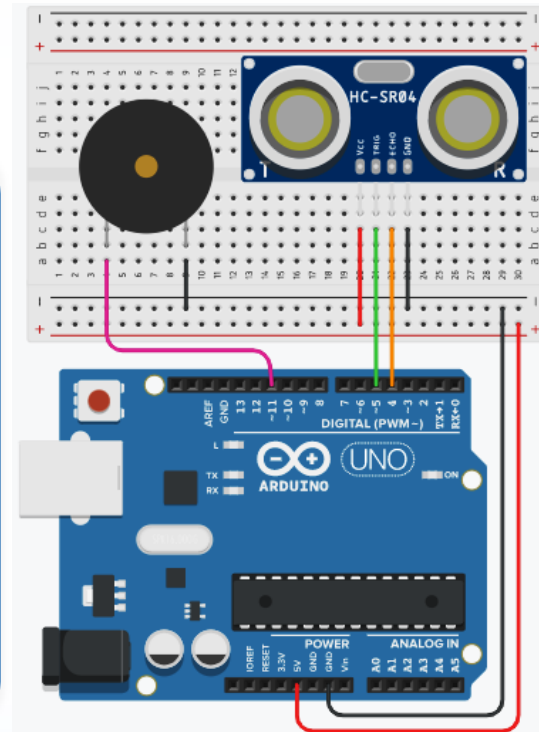
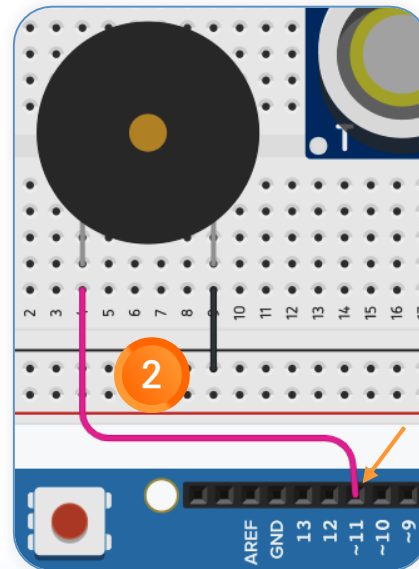
초음파센서(HC-SR04)



피에조 부저

[사전준비(배선하기)]

- 1 구성요소 - [기본]-[피에조]를 클릭 & 드래그
- 2 브레드보드를 이용하여 피에조 부저 배선
 - ▶ [양극(+)]단자 - [아두이노 11번] 포트
 - ▶ [음극(-)]단자 - 접지 연결

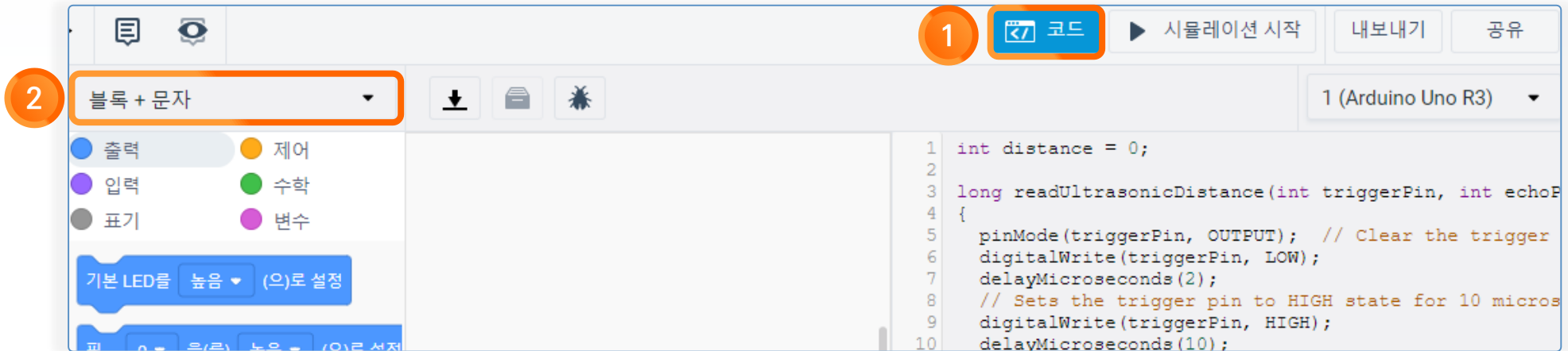




1. 스크립트 작성

[15주차 1차시 코드 활용]

- 화면 상단에서 [코드] 클릭
- [블록 + 문자] 클릭

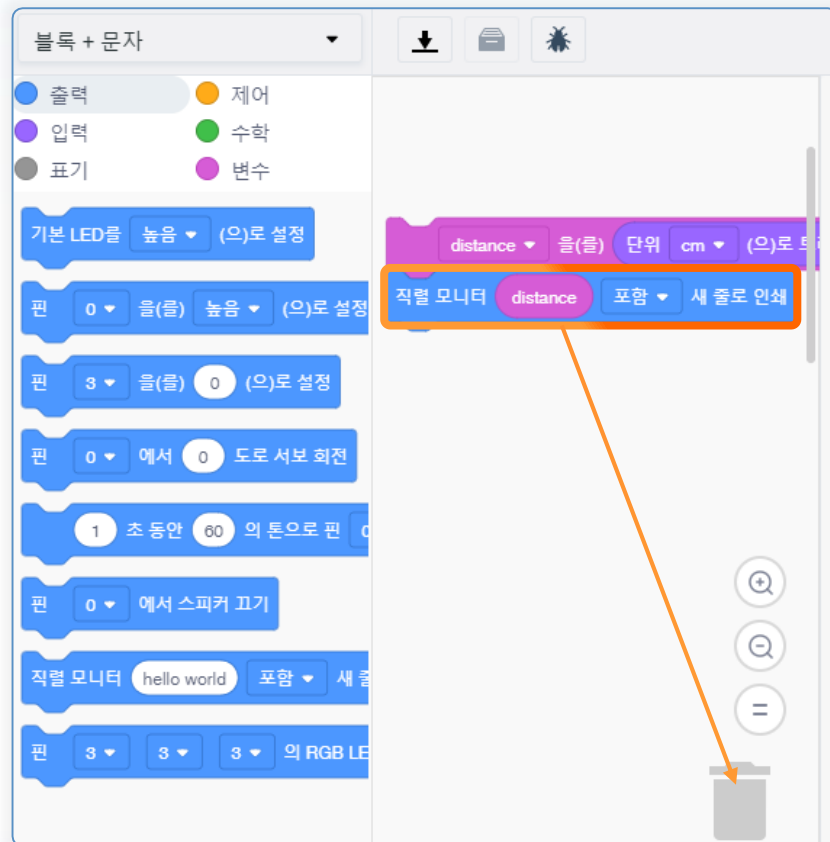


1. 실습예제 - 물체 감지 시뮬레이터 코딩하기



1. 스크립트 작성

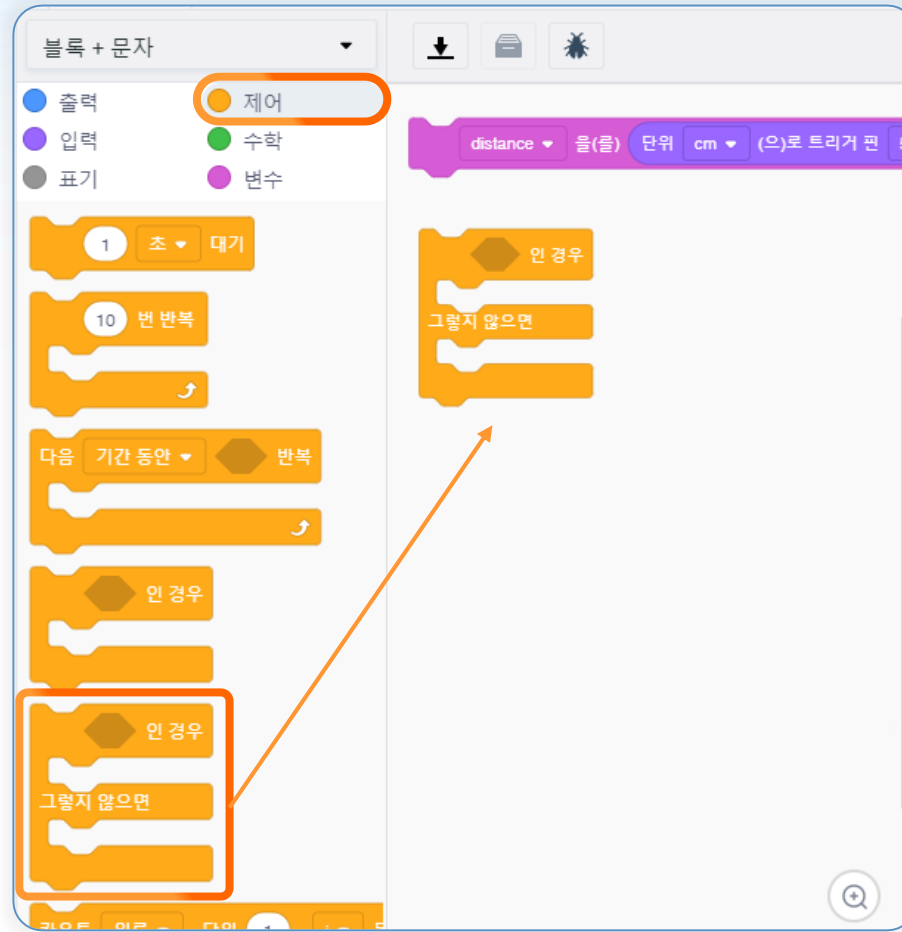
- 물체 감지 시 **피에조 부저**로 감지 상황을 알리기 위해 [**직렬 모니터**] 삭제





1. 스크립트 작성

- [제어]- 인 경우 드래그

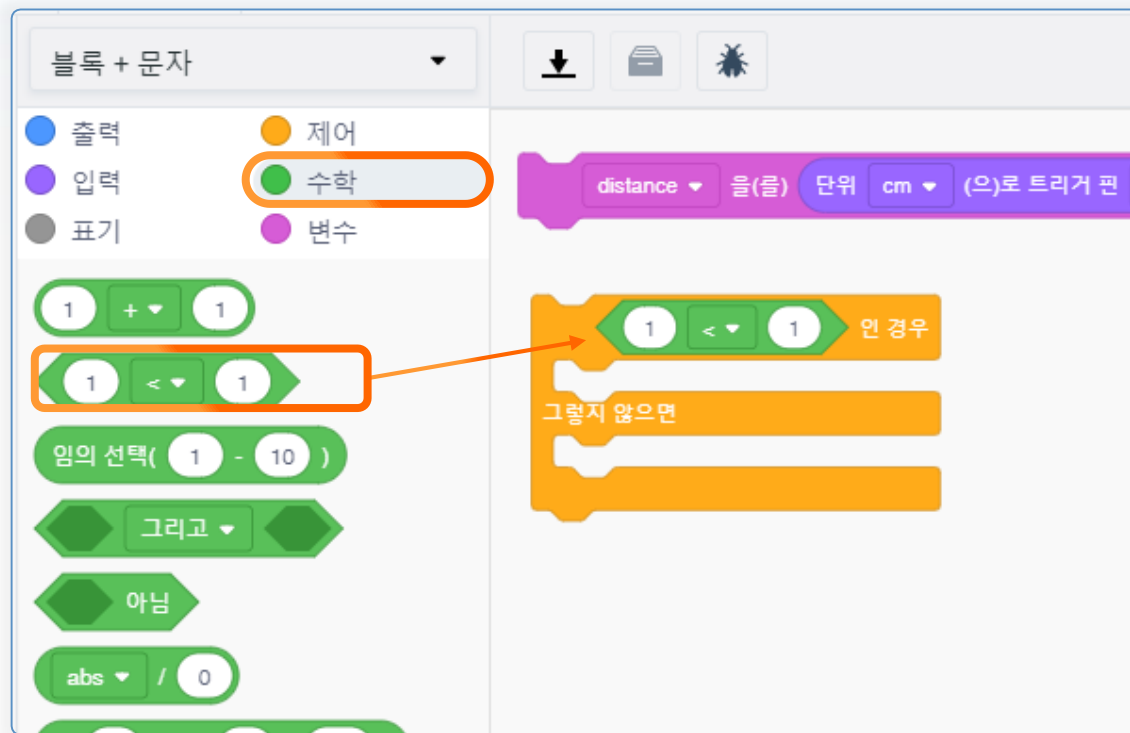


1. 실습예제 - 물체 감지 시뮬레이터 코딩하기




1. 스크립트 작성

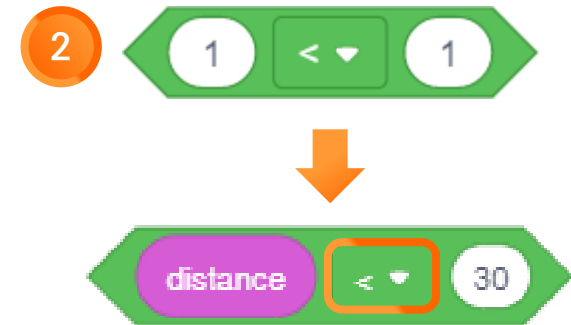
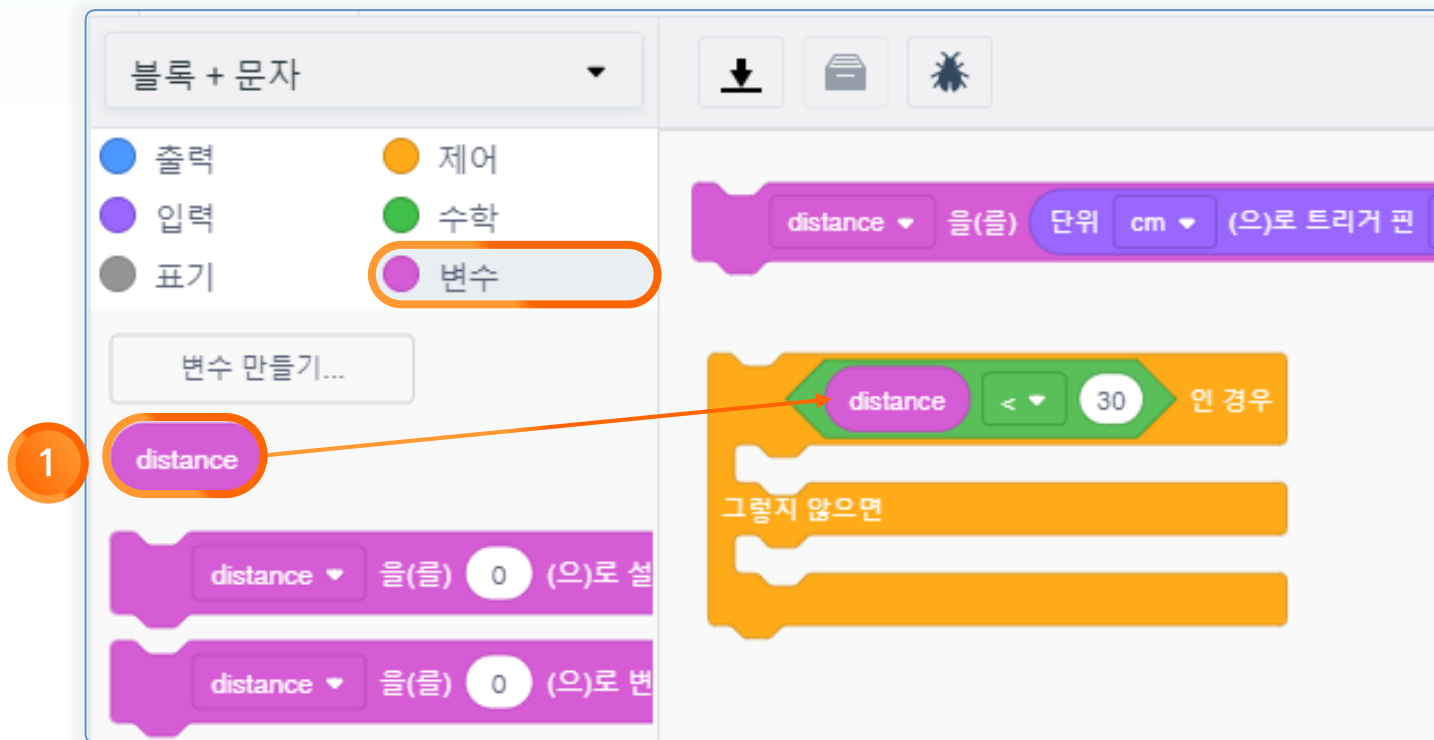
- [수학] - [] 블록을 드래그하여 조건 제어< ...인 경우> 블록 속으로 결합





1. 스크립트 작성

- [변수] - [distance] - 조건 제어 블록으로 드래그
- 조건에서 [30] 입력.  즉, 조건을 30보다 작으면 0으로 설정





1. 스크립트 작성

- 만일 $distance < 30$ 이라면,
[출력] - [스피커 재생] 을 드래그



1 초 동안 60 의 톤으로 핀 0 에서 스피커 재생

블록 + 문자

출력 제어
입력 수학
표기 변수

기본 LED를 높음 (으)로 설정

핀 0 을(를) 높음 (으)로 설정

핀 3 을(를) 0 (으)로 설정

핀 0 에서 0 도로 서보 회전

1 초 동안 60 의 톤으로 핀 0 에서 스피커 재생

핀 0 에서 스피커 끄기

distance 음(들) 단위 cm (으)로 트리거 핀 5 에:

distance < 30 인 경우

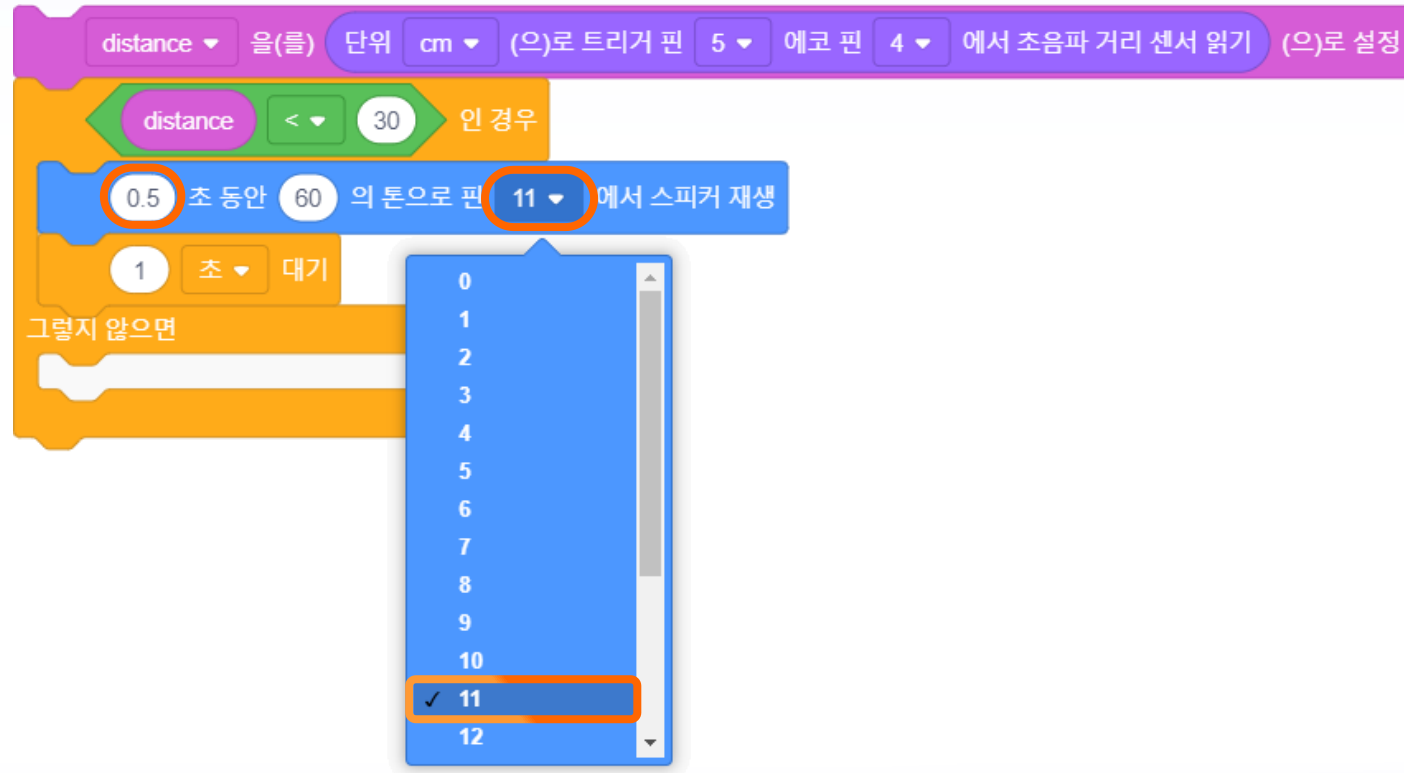
1 초 동안 60 의 톤으로 핀 0 에서 스피커 재생

그렇지 않으면



1. 스크립트 작성

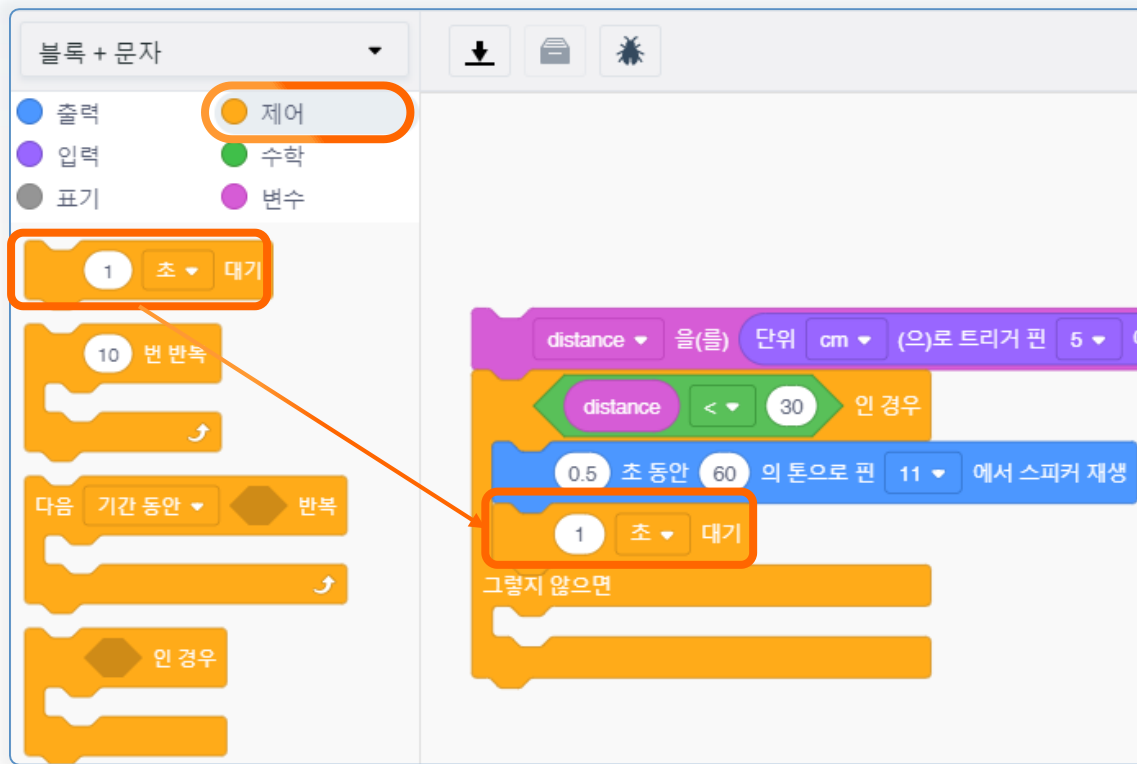
- 출력 시간을 1초에서 0.5초로 수정
- 스피커 출력 포트를 [11]로 지정





1. 스크립트 작성

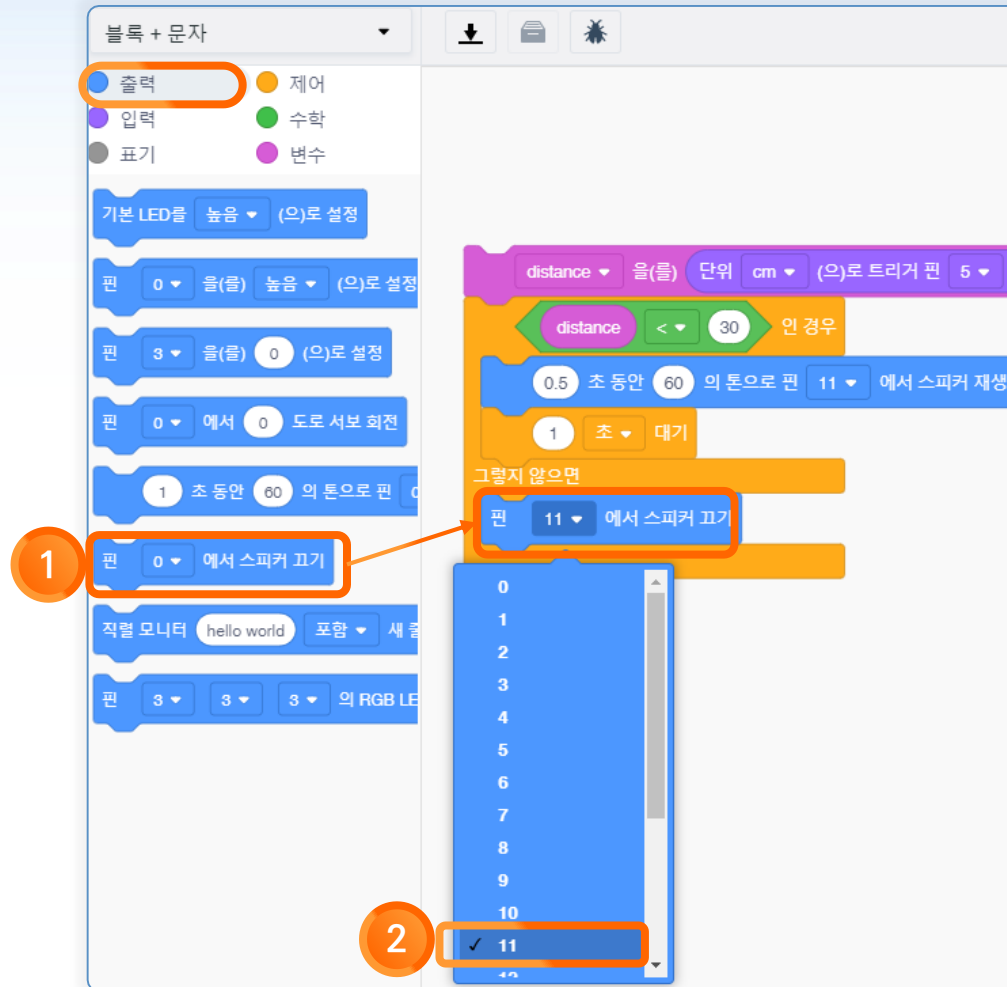
- [제어] - 1 초 대기 블록을 조건제어 블록으로 드래그





1. 스크립트 작성

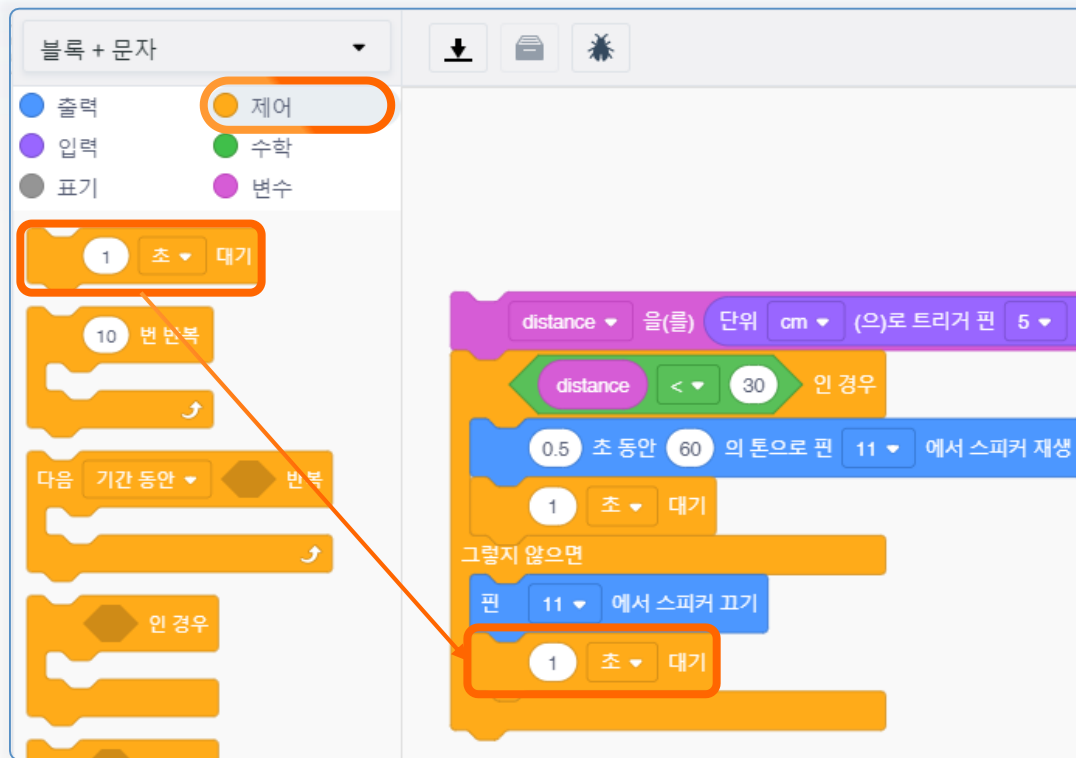
- 만일 “distance < 30”이 **아니라면**,
[출력] - **핀 0** 에서 스피커 끄기 블록을
제어문으로 드래그
- [스피커 끄기 포트] 블록에서
포트 번호를 [11]로 지정





1. 스크립트 작성

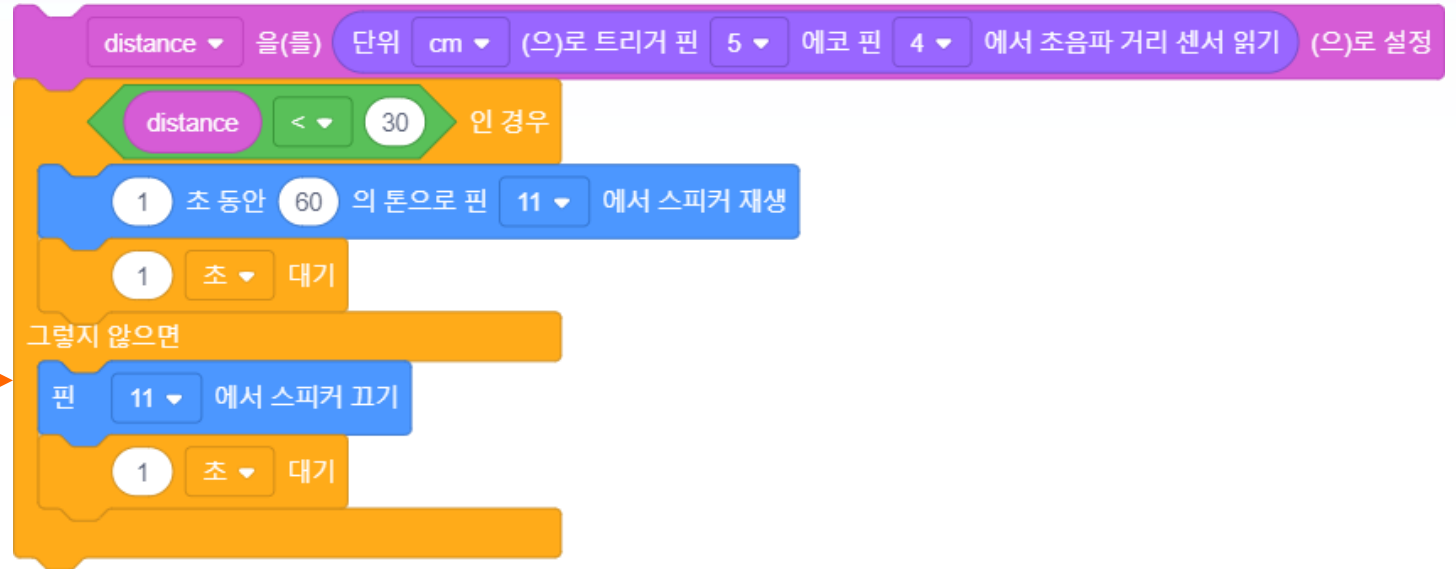
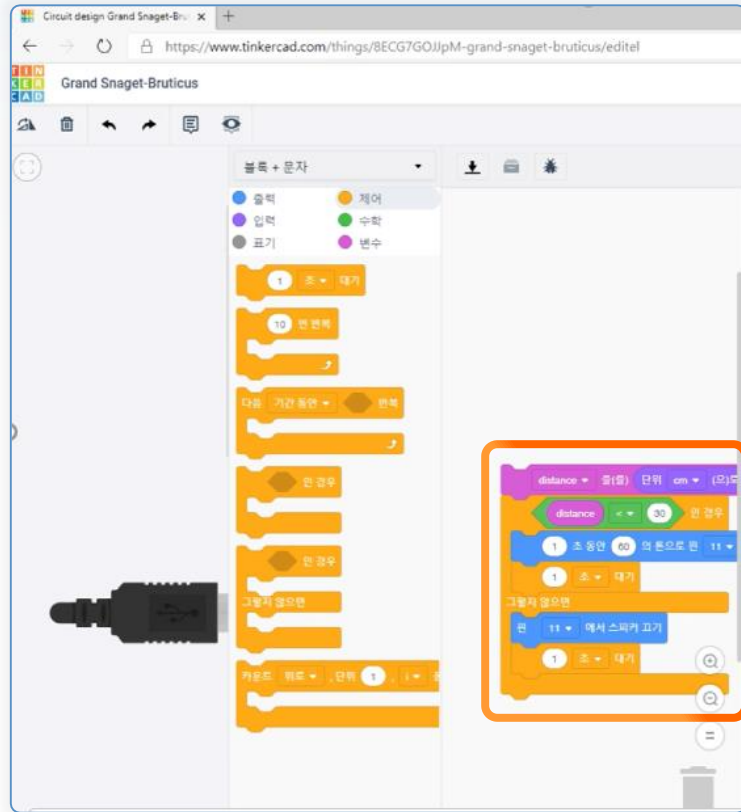
- [제어] - 1 초 대기 블록을 제어문으로 드래그





1. 스크립트 작성

- 전체 스크립트 완성 화면(블록 기반 언어 - 스크래치)





1. 스크립트 작성

- 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

블록 + 문자

- 출력 (blue)
- 입력 (purple)
- 표기 (grey)
- 제어 (orange)
- 수학 (green)
- 변수 (pink)

스크립트 블록:

- 기본 LED를 높음 (으)로 설정
- 핀 0 을(를) 높음 (으)로 설정
- 핀 3 을(를) 0 (으)로 설정
- 핀 0 에서 0 도로 서보 회전
- 1 초 동안 60 의 톤으로 핀 0
- 핀 0 에서 스피커 끄기
- 직렬 모니터 hello world 포함 새 줄
- 핀 3 3 3 의 RGB LED

스크립트 블록 (조건문):

- distance 을(를) 단위 cm (으)로 트리거 핀
- distance < 30 인 경우
- 1 초 동안 60 의 톤으로 핀 11 에서 스피커
- 1 초 대기
- 그렇지 않으면
- 핀 11 에서 스피커 끄기
- 1 초 대기

C++ 코드 (1 (Arduino Uno R3))

```

int distance = 0;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  pinMode(11, OUTPUT);
}

void loop()
{
  distance = 0.01723 * readUltrasonicDistance(5, 4);
  if (distance < 30) {
    tone(11, 523, 1000); // play tone 60 (C5 = 523 Hz)
    delay(1000); // Wait for 1000 millisecond(s)
  } else {
    noTone(11);
    delay(1000); // Wait for 1000 millisecond(s)
  }
}

```




1. 스크립트 작성

- 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

```
1  int distance = 0;
2
3  long readUltrasonicDistance(int triggerPin, int echoPin)
4  {
5      pinMode(triggerPin, OUTPUT); // Clear the trigger
6      digitalWrite(triggerPin, LOW);
7      delayMicroseconds(2);
8      // Sets the trigger pin to HIGH state for 10 microseconds
9      digitalWrite(triggerPin, HIGH);
10     delayMicroseconds(10);
11     digitalWrite(triggerPin, LOW);
12     pinMode(echoPin, INPUT);
13     // Reads the echo pin, and returns the sound wave travel time in microseconds
14     return pulseIn(echoPin, HIGH);
15 }
16
17 void setup()
18 {
19     pinMode(11, OUTPUT);
20 }
```

readUltrasonicDistance() 함수
: 이전 실습의 코딩내용과 동일

[설명]

☞ 11번 포트를 피에조 부저 출력용으로 지정



1. 스크립트 작성

- 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

```
22 void loop()
23 {
24     distance = 0.01723 * readUltrasonicDistance(5, 4);
```

[설명]

☞ 함수를 호출하여 거리 측정

```
25     if (distance < 30) {
26         tone(11, 523, 1000); // play tone 60 (C5 = 523 Hz)
27         delay(1000); // Wait for 1000 millisecond(s)
```

☞ 거리가 30cm미만이면
1000ms(1초)동안 11번 포트에 연결된
피에조 부저를 높은 도(C5 = 523 Hz)음으로 울림

```
28     } else {
29         noTone(11);
30         delay(1000); // Wait for 1000 millisecond(s)
31     }
32 }
```

☞ 조건을 만족하지 않으면(즉, 30cm이상이면)
11번 포트에 연결된 피에조 부저 울림 멈춤(1초 동안)

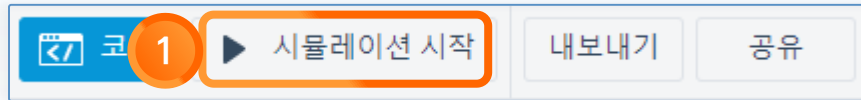
부저 연주(소리) 함수

- **Tone**(핀번호, 주파수, [출력시간(ms)] : 부저연주(소리출력)
- **noTone**(핀번호) : Tone()에 의한 출력 중단

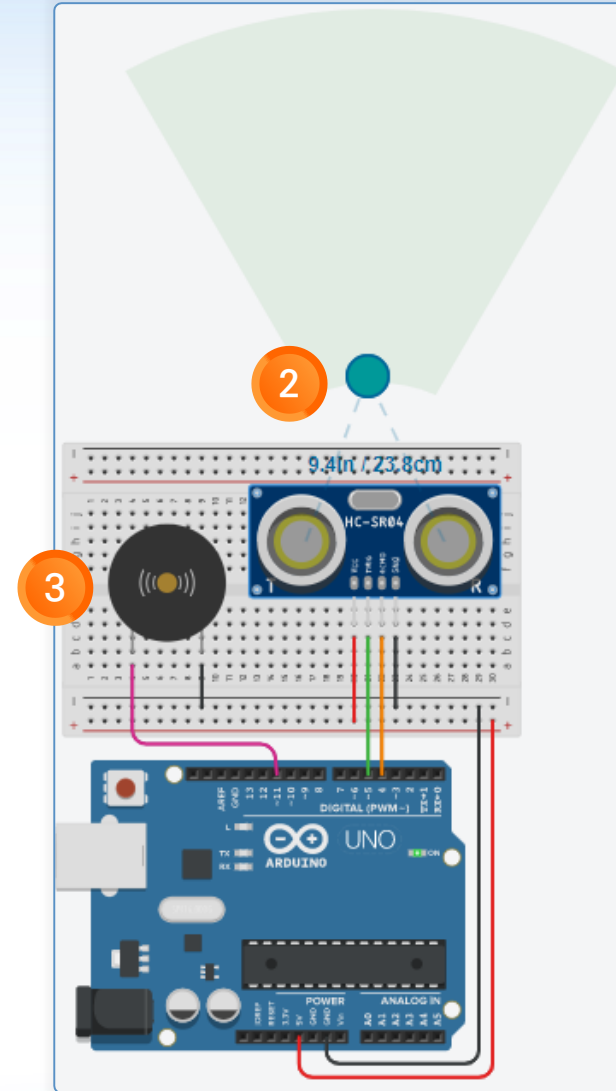


2. 스크립트 실행

- 화면 상단의 [시뮬레이션 시작] 메뉴 클릭



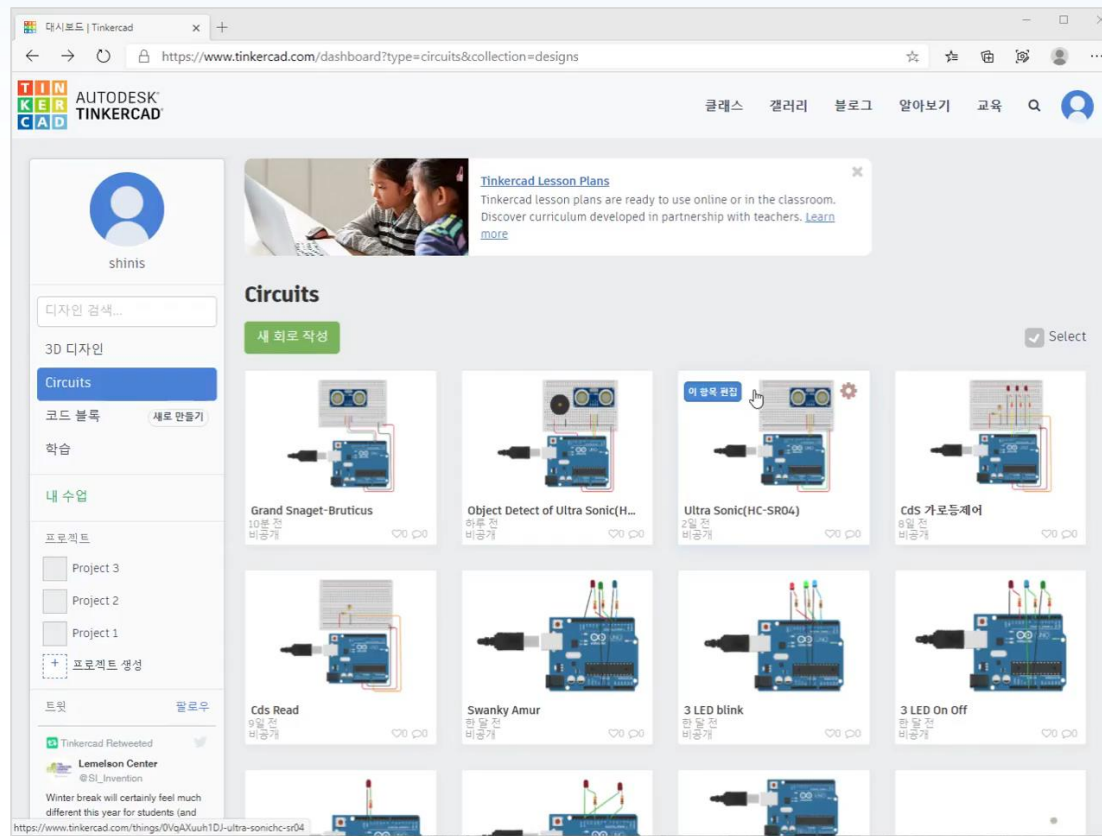
- 초음파센서(HC-SR04)를 클릭,
녹색 원형 물체를 클릭하여 앞뒤, 좌우로 이동
- 거리(distance)가 30cm 이내이면
피에조 스피커 울림





2. 스크립트 실행

- 실습영상(틴커캐드 서킷을 이용하여 초음파센서 물체감지 경고제어 시뮬레이터)



생각하기

[실습예제] 주차 차단기 제어 시뮬레이터

[문제상황]

- 자동차가 주차 차단기와의 거리가 제어 거리(실습에서는 30cm) 이내일 경우엔 **자동으로 차단기 바를 열어주게** 하려면 어떻게 해야 할까?
 - 아두이노에 **초음파 센서와 서보모터를 연결하여 하드웨어를 구성**한 다음 물체 유무에 따라 변화하는 **초음파센서의 감지 값을 읽어들인다.** 그리고 해당 값을 거리로 변환한 후 제어거리(30cm)보다 작으면 **서보모터를 90도 회전시켜 차단기 바를 자동으로 열어주는 주차 차단기** 프로그램을 작성하자.



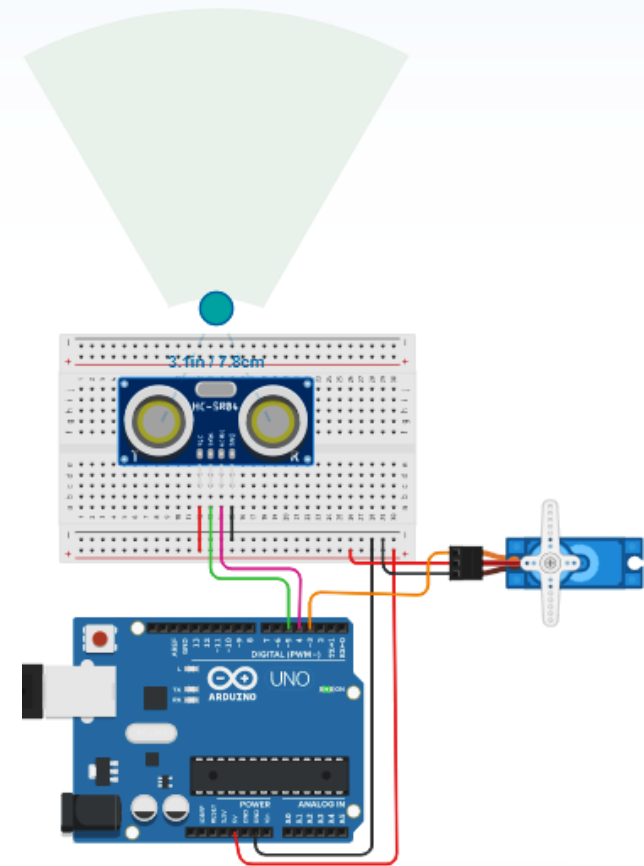
문제정의 및 분해

1. 문제정의

- 아두이노 보드에 초음파센서 1개와 서보모터를 연결한 후 **전방에 자동차와 같은 이동 물체를 감지하고 자동차와의 거리가 일정범위 이내면 차단기 바가 올라가는 주차 차단기 시뮬레이터 작성**

2. 문제분해

- 아두이노 보드에 **초음파센서, 서보모터를 연결하여 하드웨어 구성하기**
- 틴커캐드를 이용하여 **초음파센서로부터 주변 물체를 감지하고 물체와의 거리가 일정범위 (여기에서는 30cm) 이내면 서보 모터가 90° 이동하여 차단기 바를 열어주는 주차게이트 시뮬레이터 구현**
 - ▶ 초음파센서 감지 값 읽기, 물체 유무 파악, 거리 값 변환, 서보모터 90° 이동

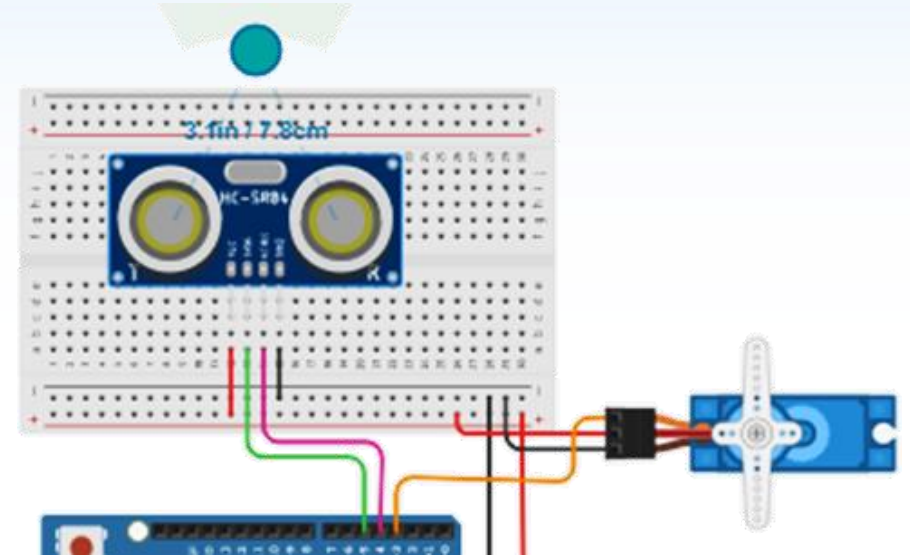




알고리즘

• 아래 명령어 반복하기

- ▶ 초음파센서 감지 값 읽기
- ▶ 물체 유무 파악
- ▶ 거리 값 변환
- ▶ 만일 거리가 30cm 이내라면 서보모터 90° 구동
- ▶ 아니면 원래 위치(0°)로 이동

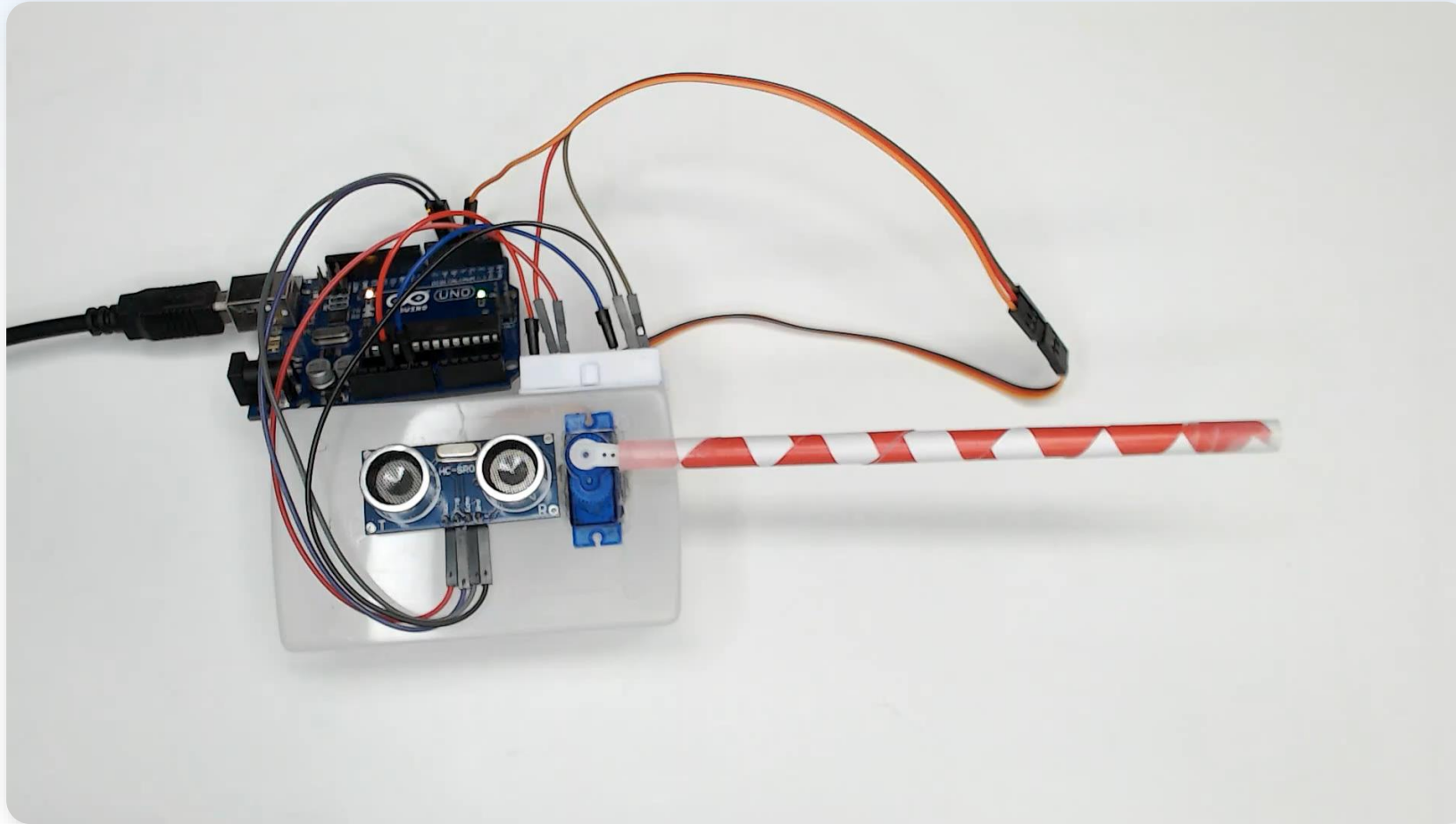


코딩하기

[실습예제] 주차 차단기 제어 시뮬레이터

▶ 결과영상 (아두이노 활용) 미리보기

(물체와의 거리가 30cm 이내면 차단 바를 문을 열어주는 주차 차단기)



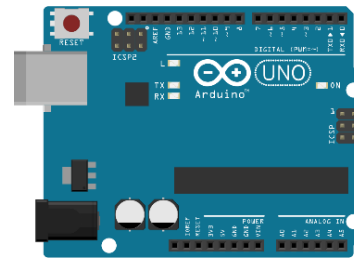
[사전준비]

- 아두이노 보드
- 브레드 보드
- 초음파센서(HC-SR04)
- 서보모터(Servo Motor)

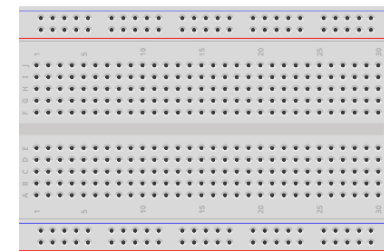


서보(servo) 모터는 보통 0°부터 180°까지
위치 제어가 가능한 모터

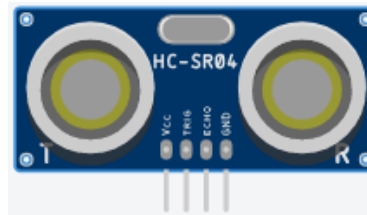
실습에 필요한 부품



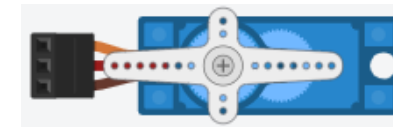
아두이노(UNO) 보드



브레드 보드



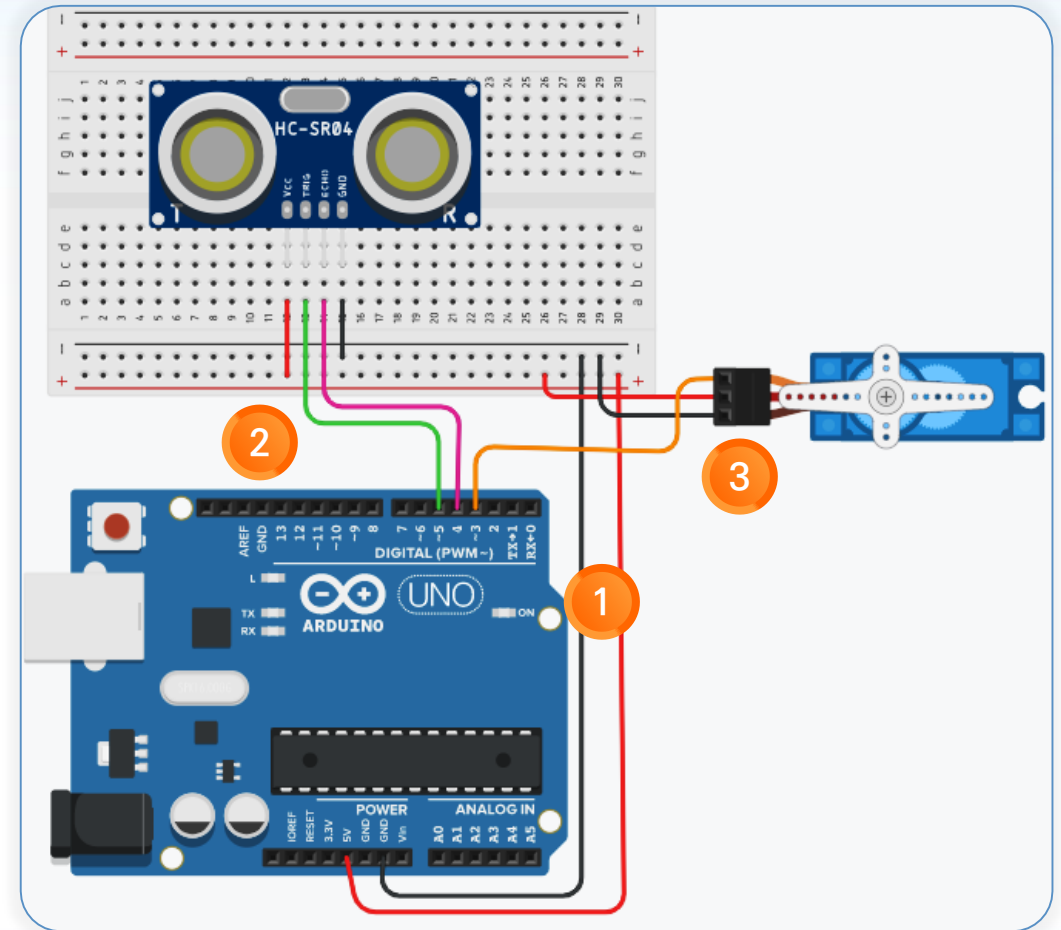
초음파센서(HC-SR04)



서보모터

[사전준비(배선하기)]

- ① 아두이노의 +5V, GND(접지)를 브레드 보드의 전원 공통 단자(+,-)와 연결
- ② 초음파 센서의 Trig, ECHO단자를 각각 아두이노의 5번, 4번 포트에 연결
- ③ 서보모터 배선
 - ▶ [전원] 단자 : +
 - ▶ [음극(-)] 단자 : -(GND)
 - ▶ [신호] 단자 : 아두이노 Pin

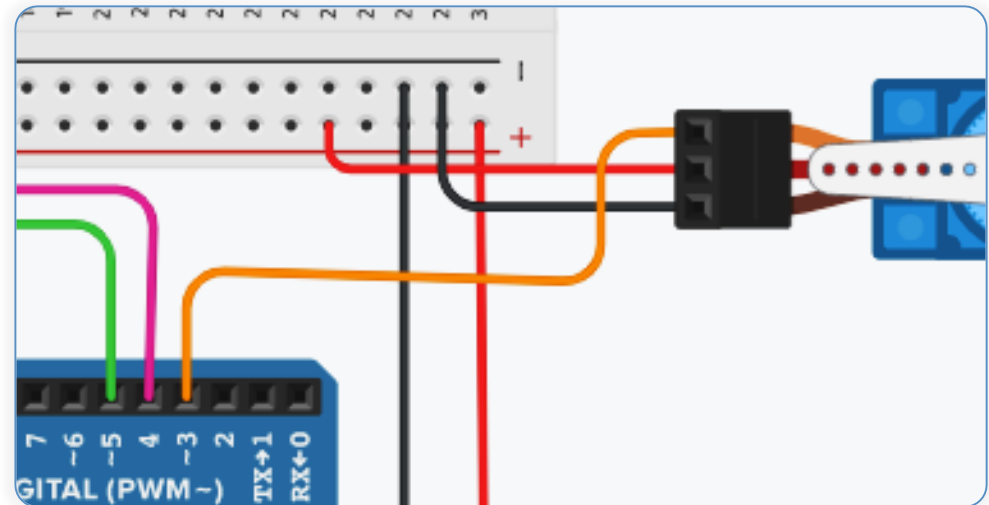


[사전준비(배선하기)]

③ 서보모터 배선(확대)

- ▶ **[전원]** 단자 : 브레드보드의 **[+]**
- ▶ **[음극(-)]** 단자 : 브레드보드 접지**[-]** 연결
- ▶ **[신호]** 단자 : 아두이노 **3번** 연결

3 : 신호
+ : 전원
- : 접지





1. 스크립트 작성

- 화면 상단 우측에 있는 구성요소 - [기본]-[Arduino] 클릭 & 드래그
- [기본]-[작은 브레드보드]를 클릭 & 드래그
- [모두]-[초음파 거리센서(4핀)]를 클릭 & 드래그

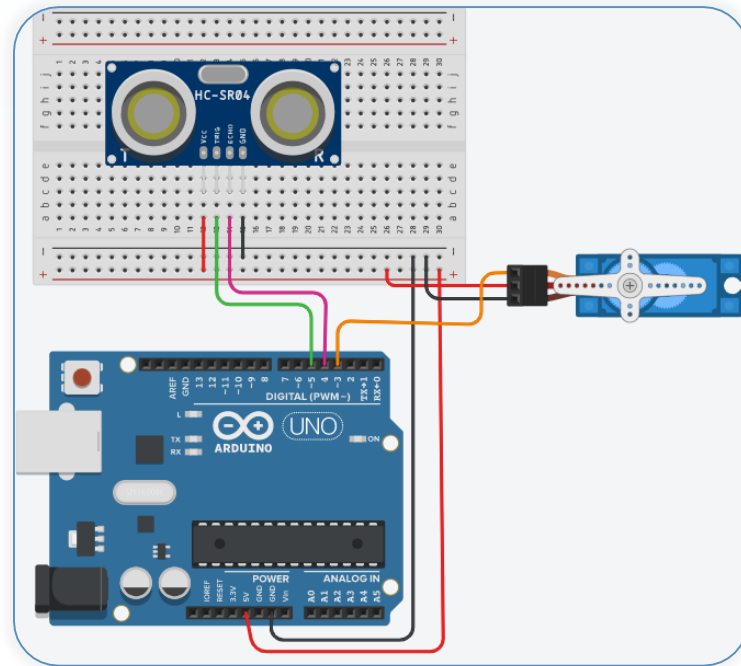


2. 실습예제 - 주차 차단기 제어 시뮬레이터 ❖ 코딩하기



1. 스크립트 작성

- 구성요소-[기본]-[마이크로 서보]를 클릭 & 드래그





1. 스크립트 작성

- 화면 상단에서 [코드] 클릭
- [블록 + 문자] 클릭

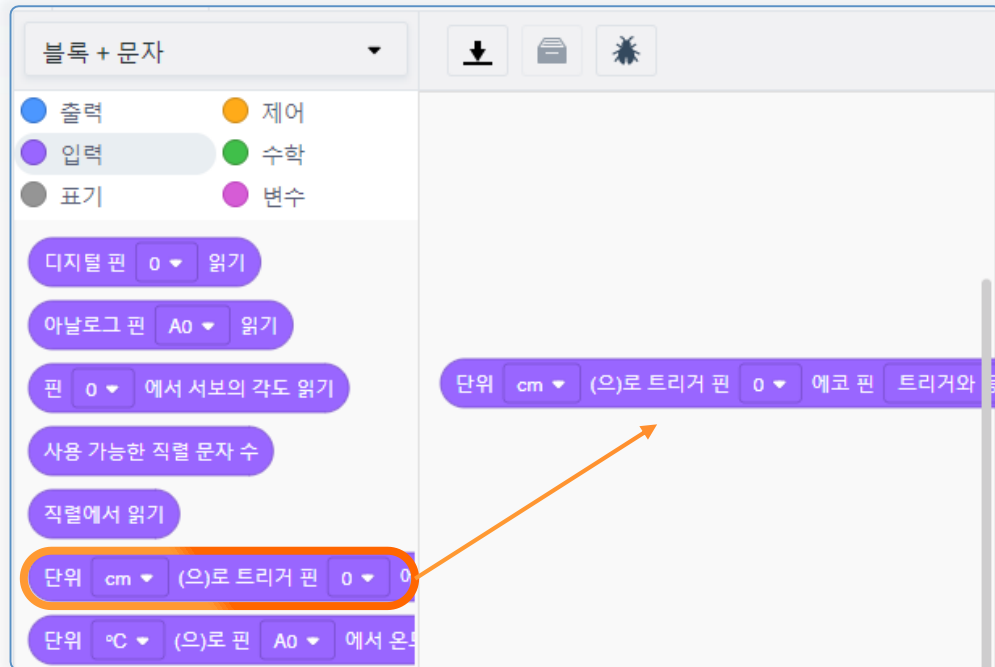
The screenshot shows the Arduino IDE interface. At the top, the '코드' (Code) button is highlighted with a red circle and the number 1. Below it, the '블록 + 문자' (Block + Text) button is highlighted with a red circle and the number 2. The block palette on the left shows various categories: 출력 (Output), 제어 (Control), 입력 (Input), 수학 (Math), 표기 (Text), and 변수 (Variable). The code editor on the right displays the following C++ code:

```
1 int distance = 0;
2
3 long readUltrasonicDistance(int triggerPin, int echoPin)
4 {
5     pinMode(triggerPin, OUTPUT); // Clear the trigger
6     digitalWrite(triggerPin, LOW);
7     delayMicroseconds(2);
8     // Sets the trigger pin to HIGH state for 10 microseconds
9     digitalWrite(triggerPin, HIGH);
10    delayMicroseconds(10);
11    digitalWrite(triggerPin, LOW);
12    pinMode(echoPin, INPUT);
13    // Reads the echo pin, and returns the sound wave travel time in microseconds
```



1. 스크립트 작성

- [입력] - [단위 cm (으)로 트리거 핀 0 에코 핀 ... 초음파 거리센서읽기] 드래그

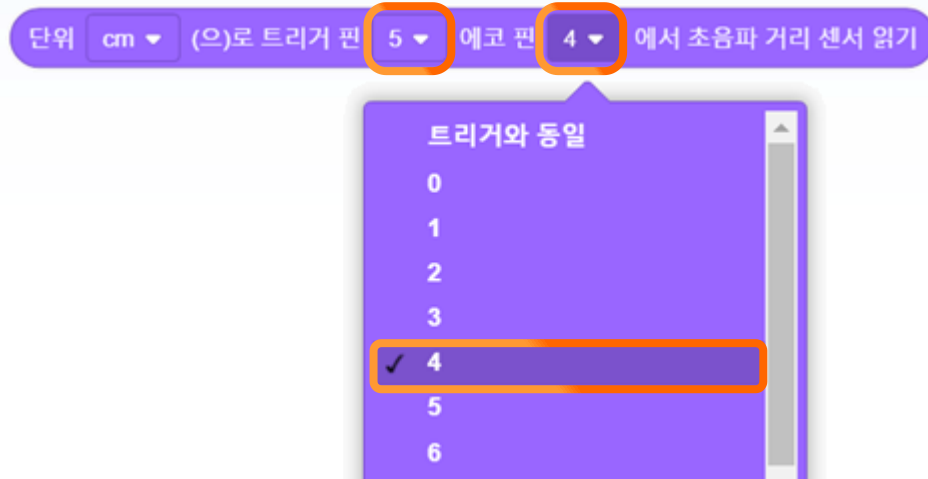


단위 cm (으)로 트리거 핀 5 에코 핀 4 에서 초음파 거리 센서 읽기



1. 스크립트 작성

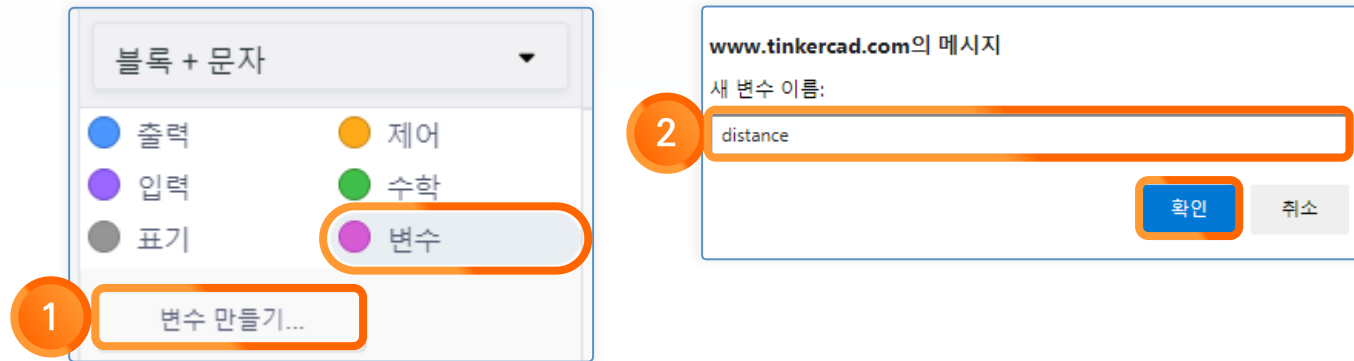
- [트리거 핀] -[5], [에코 핀] -[4]로 설정





1. 스크립트 작성

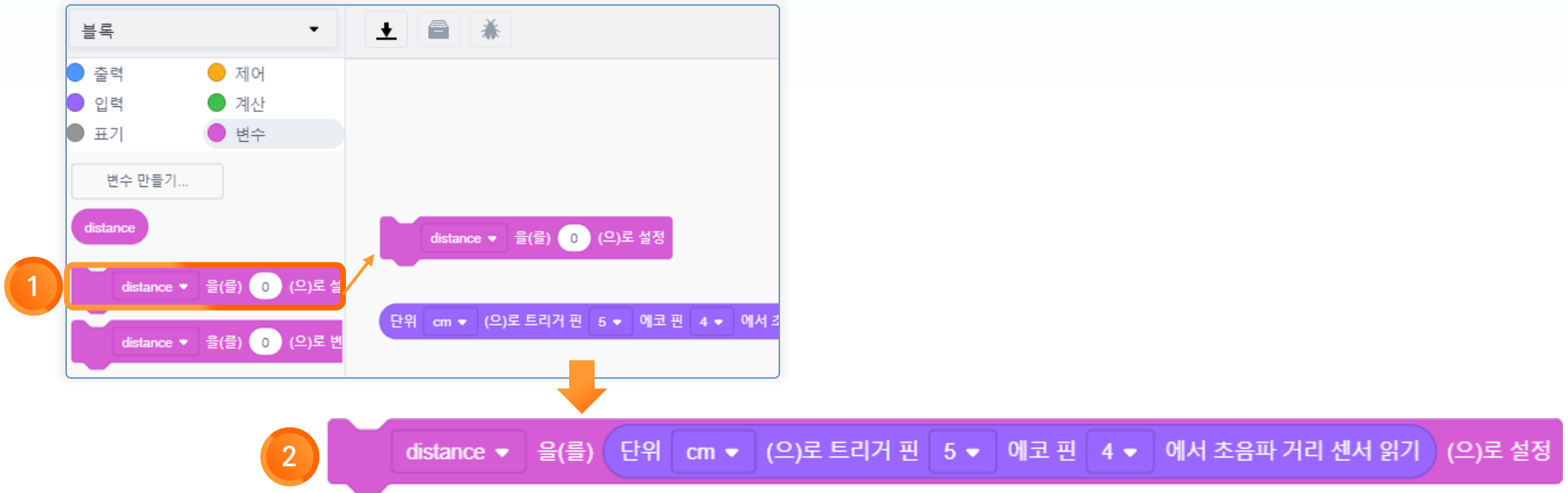
- 화면에서 [변수] 클릭 후 [변수 만들기] 클릭
- 변수로 “distance” 생성 후 [확인] 클릭





1. 스크립트 작성

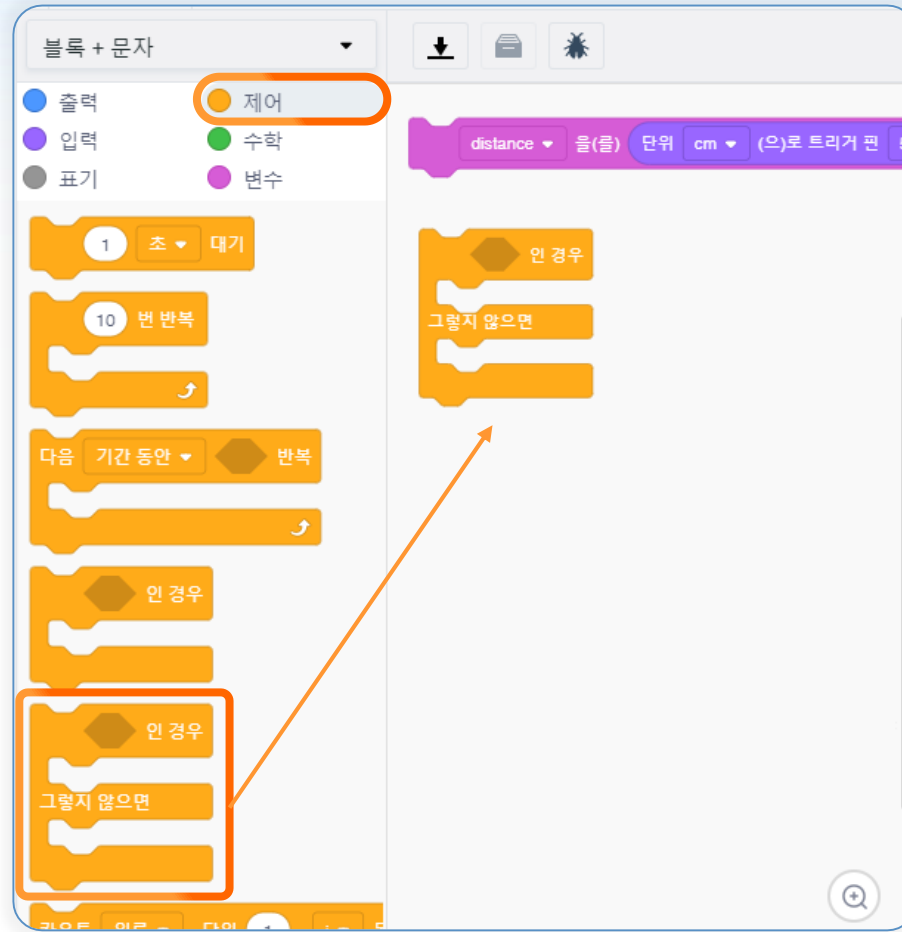
- [distance 설정] 드래그
- 단위 cm (으)로 트리거 핀 5 에코 핀 4 에서 초음파 거리 센서 읽기 블록과 “distance 설정” 결합





1. 스크립트 작성

- [제어]- 인 경우 드래그

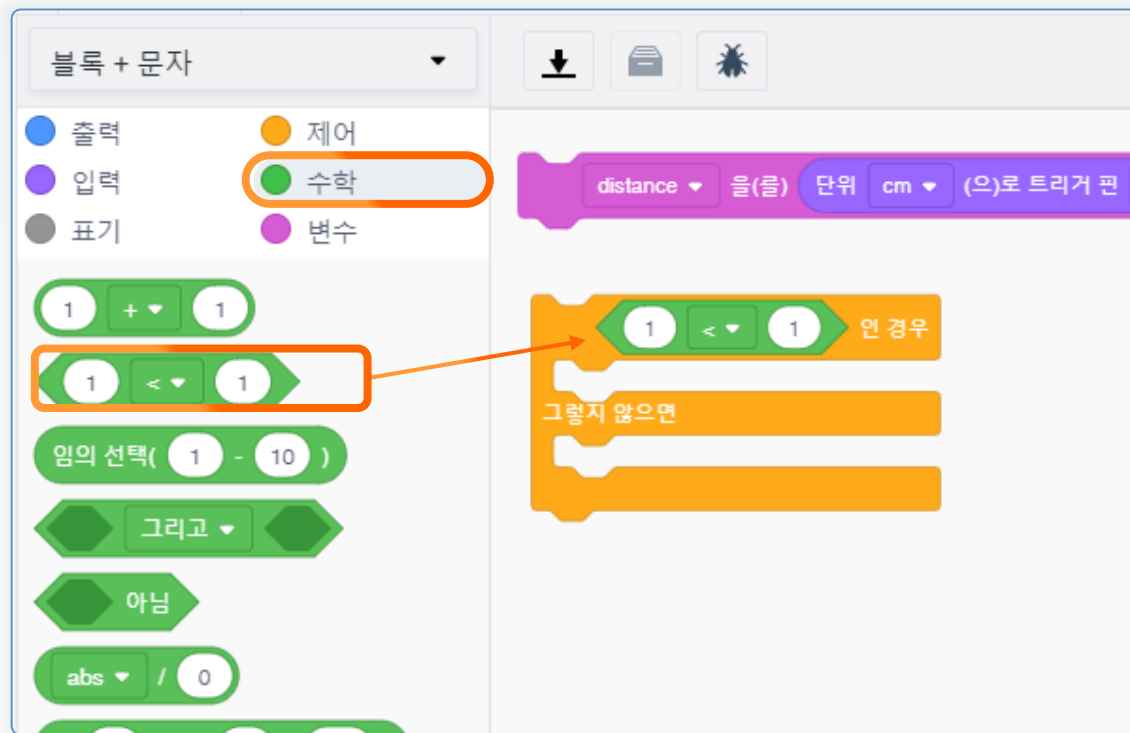


2. 실습예제 - 주차 차단기 제어 시뮬레이터 ❖ 코딩하기





1. 스크립트 작성

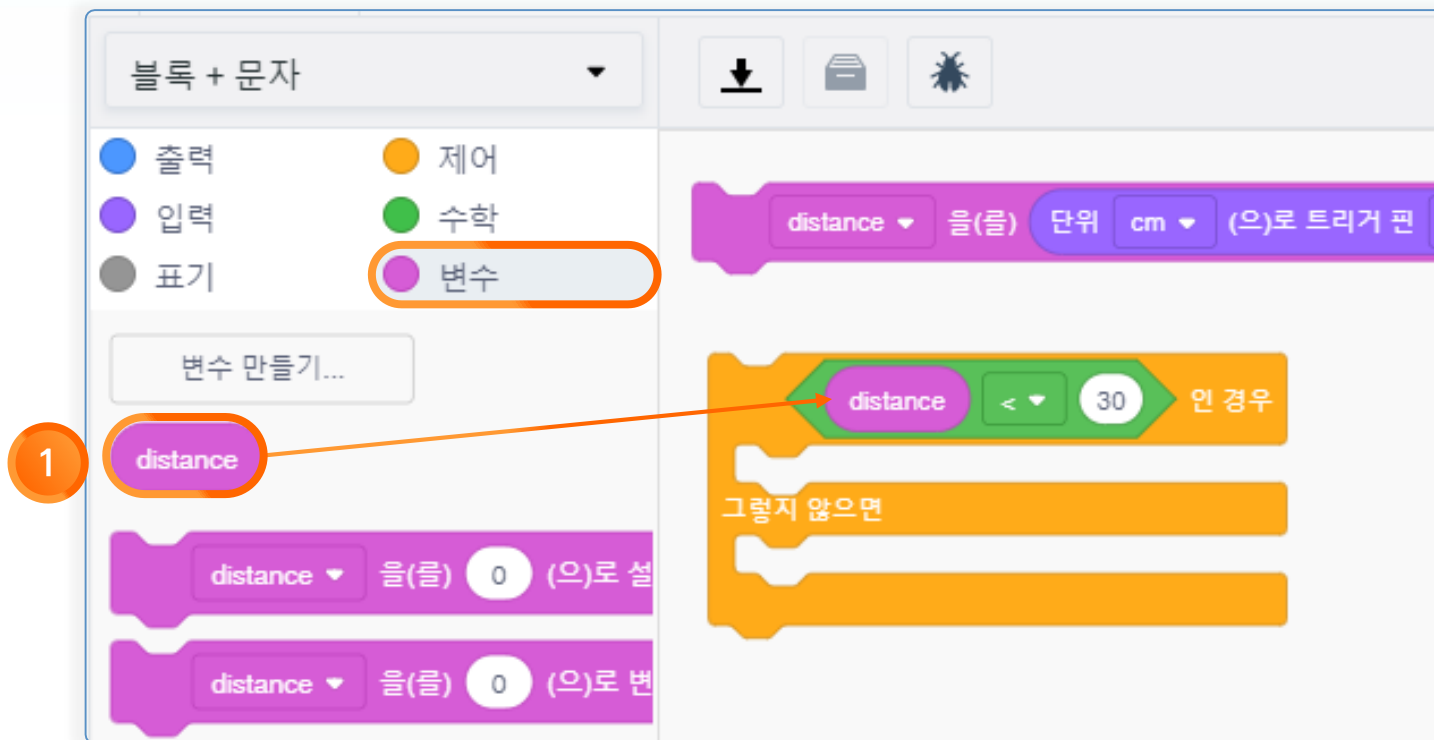
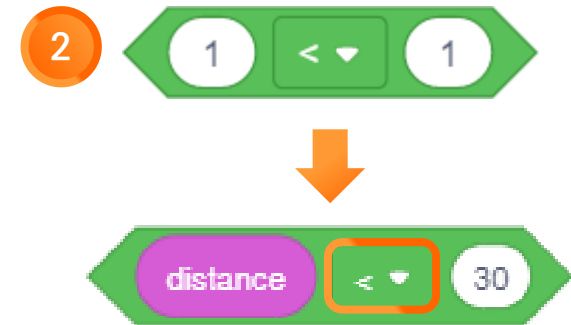
- [수학] - [] 블록을 드래그하여 조건 제어< ...인 경우> 블록 속으로 결합





1. 스크립트 작성

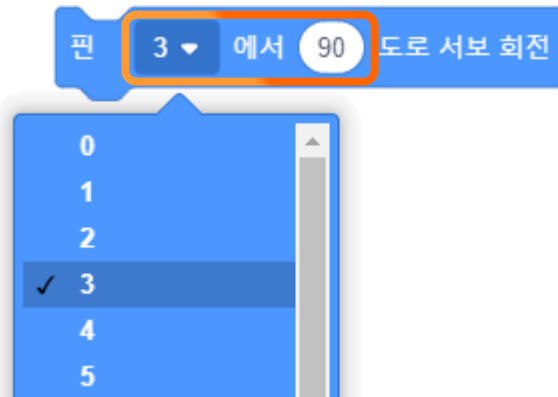
- [변수] - [distance]를 조건 제어 블록 속의  블록과 결합
- 조건에서 [30] 입력.  즉, 조건을 30보다 작으면 0으로 설정



1. 스크립트 작성

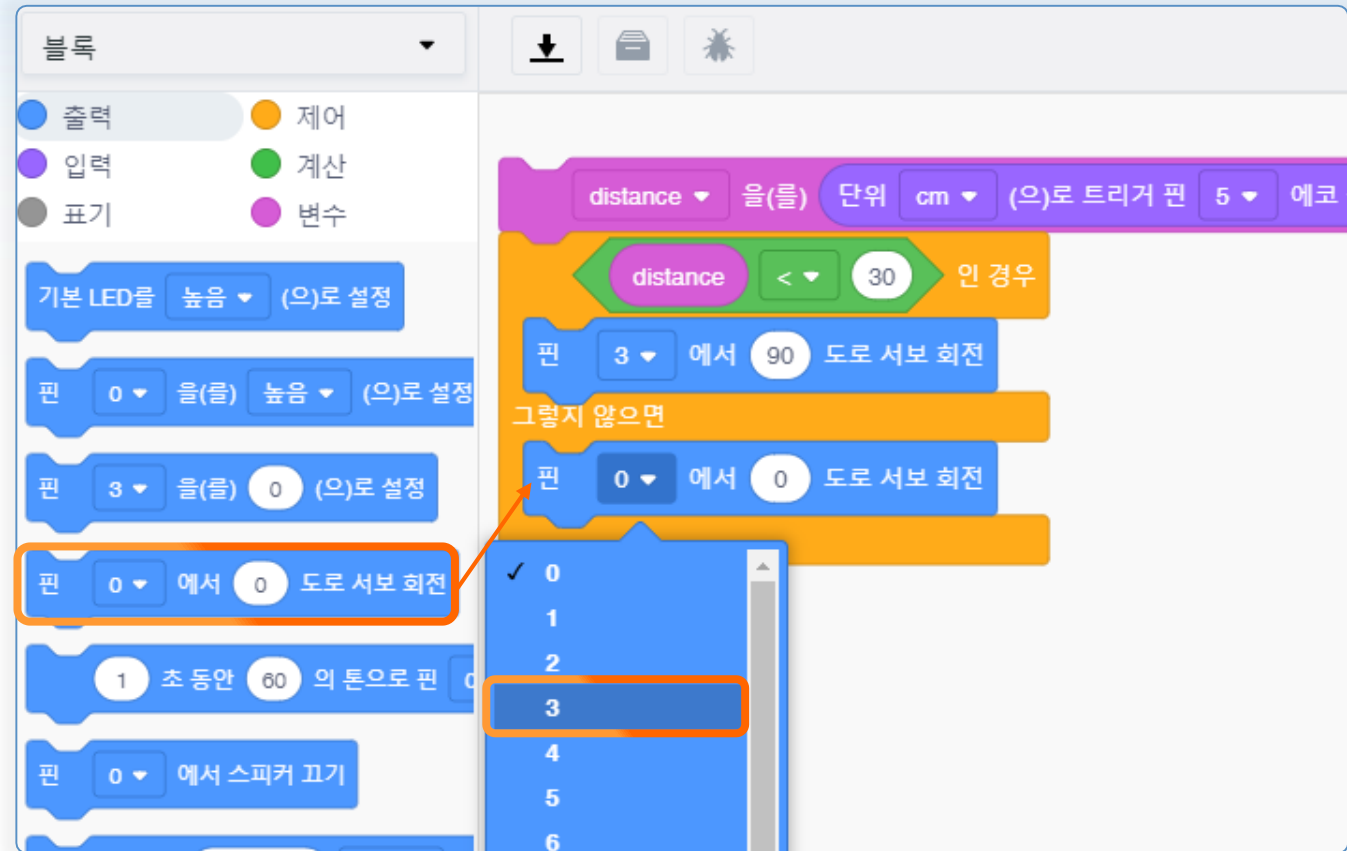
- 만일 $distance < 30$ 인 경우,
[출력] - 핀 0 에서 0 도로 서보 회전
드래그
- 핀을 3으로 수정하고
각도를 90°로 수정





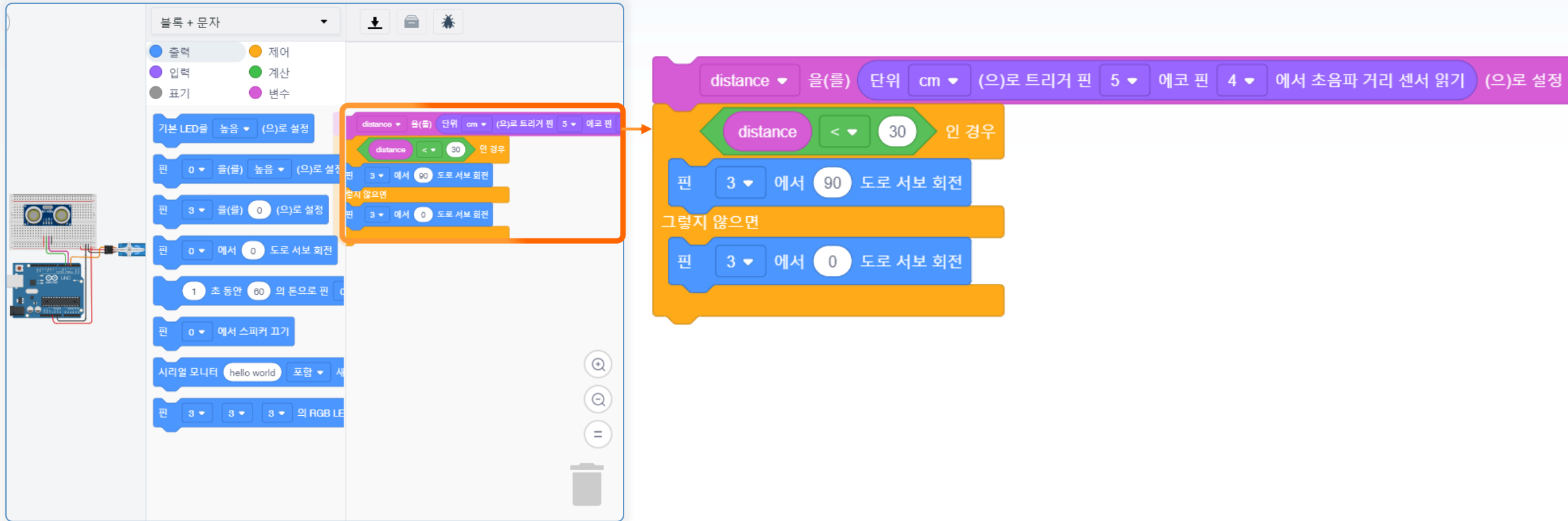
1. 스크립트 작성

- 만약 거리가 30cm 이상이면
원래 위치로 되돌아와야 하므로
그렇지 않으면 블록 내에
[출력] - [핀 0 에서 0 도로 서보 회전]을
드래그
- 핀을 3으로 수정하고
각도를 0° 그대로 유지



1. 스크립트 작성

- 전체 스크립트 완성 화면(블록 기반 언어 – 스크래치)





1. 스크립트 작성

- 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

The image displays the Arduino IDE interface with a block-based script on the left and its corresponding C++ code on the right. The block script is for an Arduino Uno R3 and includes the following logic:

- 기본 LED를 높음 (으)로 설정
- 핀 0 을(를) 높음 (으)로 설정
- 핀 3 을(를) 0 (으)로 설정
- 핀 0 에서 0 도로 서보 회전
- 1 초 동안 60 의 톤으로 핀 0 을(를) 펄스 출력
- 핀 0 에서 스피커 끄기
- 시리얼 모니터 hello world 포함
- 핀 3 3 3 의 RGB LED를 점등

The C++ code on the right is as follows:

```

1 #include <Servo.h>
2
3 int distance = 0;
4
5 long readUltrasonicDistance(int triggerPin, int echoPin)
6 {
7     pinMode(triggerPin, OUTPUT); // Clear the trigger
8     digitalWrite(triggerPin, LOW);
9     delayMicroseconds(2);
10    // Sets the trigger pin to HIGH state for 10 microseconds
11    digitalWrite(triggerPin, HIGH);
12    delayMicroseconds(10);
13    digitalWrite(triggerPin, LOW);
14    pinMode(echoPin, INPUT);
15    // Reads the echo pin, and returns the sound wave travel time in microseconds
16    return pulseIn(echoPin, HIGH);
17 }
18
19 Servo servo_3;
20
21 void setup()
22 {
23     servo_3.attach(3, 500, 2500);
24 }
25
26
27 void loop()
28 {
29     distance = 0.01723 * readUltrasonicDistance(5, 4);
30     if (distance < 30) {
31         servo_3.write(90);
32     } else {
33         servo_3.write(0);
34     }
35     delay(10); // Delay a little bit to improve simulation
36 }

```



1. 스크립트 작성

- 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

```

1 #include <Servo.h>
2
3 int distance = 0;
4
5 long readUltrasonicDistance(int triggerPin, int echoPin)
6 {
7     pinMode(triggerPin, OUTPUT); // Clear the trigger
8     digitalWrite(triggerPin, LOW);
9     delayMicroseconds(2);
10    // Sets the trigger pin to HIGH state for 10 microseconds
11    digitalWrite(triggerPin, HIGH);
12    delayMicroseconds(10);
13    digitalWrite(triggerPin, LOW);
14    pinMode(echoPin, INPUT);
15    // Reads the echo pin, and returns the sound wave travel time in microseconds
16    return pulseIn(echoPin, HIGH);
17 }

```

readUltrasonicDistance()함수
: 이전 실습의 코딩내용과 동일

[설명]

- ☞ 서보모터 라이브러리를 포함
- ☞ 변수 distance 초기값 0으로 설정
- ☞ 사용자가 만든 사용자 정의 함수()
- ☞ 트리거 단자모드 설정 : 출력모드
- ☞ 트리거 신호 2ms동안 LOW상태 설정(동작 시 권장사항)
- ☞ 트리거 신호를 10ms동안 HIGH 상태 설정(동작 시 권장사항)
- ☞ 트리거 신호 LOW 상태 설정
- ☞ 에코 단자 모드설정 : 입력모드
- ☞ 에코 핀의 HIGH상태 시간(duration)을 마이크로 초 단위로 반환



1. 스크립트 작성

● 전체 스크립트 완성 화면(텍스트 기반 언어 - C)

```

19 Servo servo_3;
20
21 void setup()
22 {
23   servo_3.attach(3, 500, 2500);
24 }
25
26
27 void loop()
28 {
29   distance = 0.01723 * readUltrasonicDistance(5, 4);
30   if (distance < 30) {
31     servo_3.write(90);
32   } else {
33     servo_3.write(0);
34   }
35   delay(10); // Delay a little bit to improve simulation
36 }

```

[설명]

- ☞ 서보 모터의 이름을 Servo_3으로 지정
- ☞ 아두이노 3번 포트를 서보모터 출력용으로 지정 (회전 범위는 0°(500), 180°(2500))
- ☞ 물체(자동차)와의 거리가 30cm이내면 서보모터가 90° 이동
- ☞ 아니면(30cm이상) 서보모터가 원래 제자리(0°)로 이동

Servo.attach(pin,min,max)

.pin : 아두이노 Pin번호

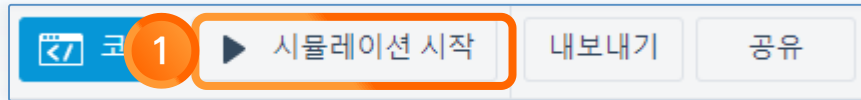
.min : 서보의 최소각도(0도)에 해당하는 us단위의 펄스 폭. (기본값은 544)

.max : 서보의 최대각도(180도)에 해당하는 us단위의 펄스 폭. (기본값은 2400)

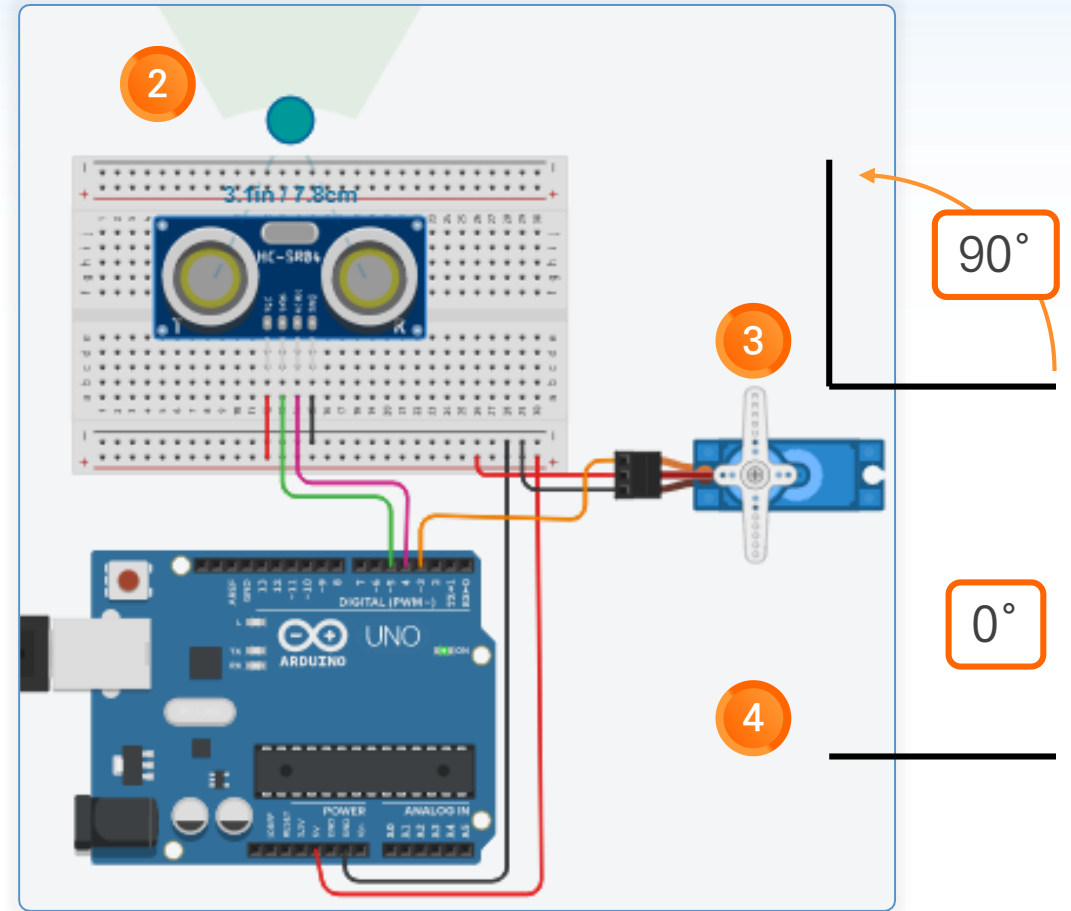


2. 스크립트 실행

- 화면 상단의 [시뮬레이션 시작] 메뉴 클릭



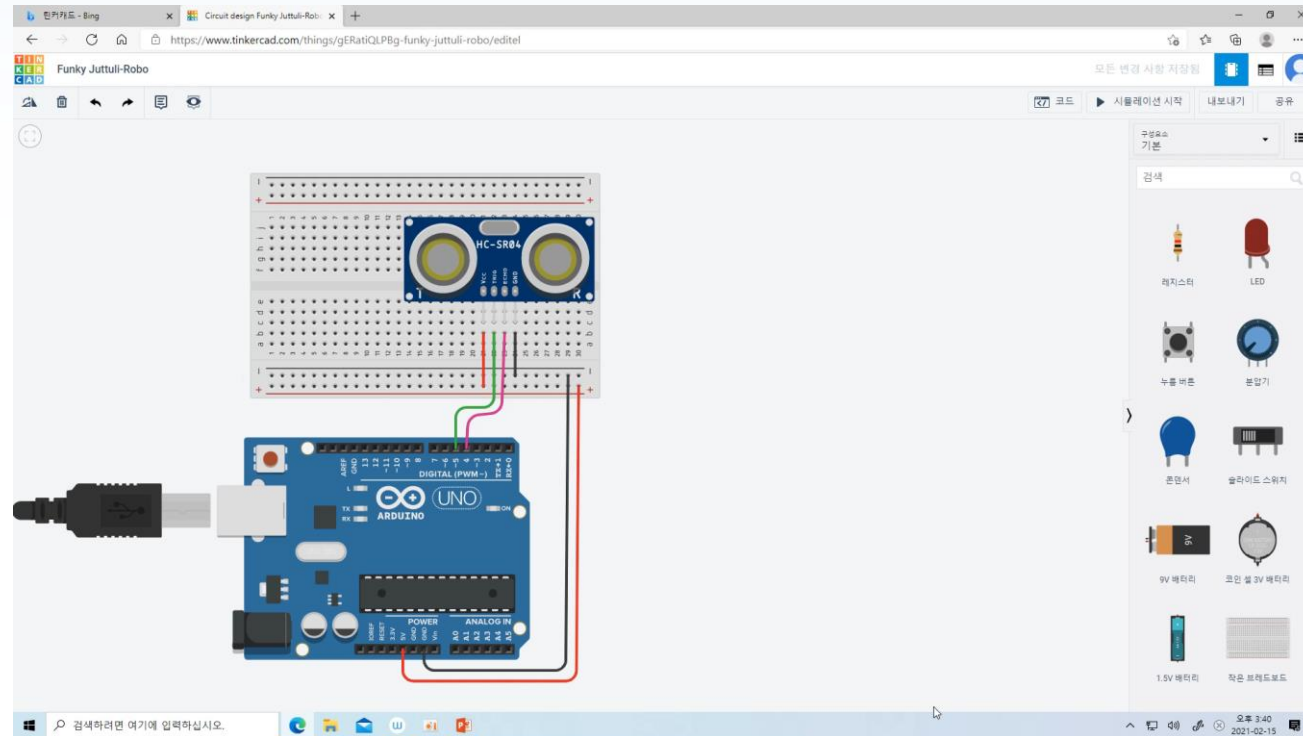
- 초음파센서(HC-SR04)를 클릭,
녹색 원형 물체를 클릭하여 앞뒤, 좌우로 이동
- 거리(distance)가 30cm 이내이면
서보모터가 90° 회전
- 아니면(30cm이상)
원래 위치(0°)로 회전





2. 스크립트 실행

- 실습영상(틴커캐드 서킷을 이용하여 초음파 센서 주차 차단기 제어 시뮬레이터)



학습정리

- ✓ 초음파센서 단자의 pin모드는 송신단자는 출력모드, 수신단자는 입력모드로 설정해야 한다.
`pinMode(triggerPin, OUTPUT); , pinMode(echoPin, INPUT);`
- ✓ 초음파 센서(HC-SR04)를 이용하여 거리 계산에 사용된 아두이노 함수 `PulseIn()` 함수는 Echo단자로 유입되는 입력 신호가 HIGH 또는 LOW가 될 때까지의 시간을 측정하여 us(마이크로 초) 단위로 돌려주는 함수이다.
- ✓ 피에조 부저의 소리 출력은 `Tone()`, `noTone()` 함수로 제어한다.
- ✓ 서보모터는 보통 0도부터 180도 까지 이동하는 위치제어 모터이다.
- ✓ 서보모터는 아두이노에서 `write(각도)` 함수를 이용하여 회전 각도를 제어한다.
[예시 : `servo_3.write(90); // 90도 회전`]

본 수업자료는 저작권법 제 25조 2항에 따라
학교 수업을 목적으로 이용되었으므로,
본 수업자료를 외부에 공개, 게시하는 것을 금지하며,
이를 위반하는 경우 저작권 침해로서 관련법에 따라 처벌될 수 있습니다.