

Servo motor

- mech9917@sogang.ac.kr

What is Servo?

Servant -> Servo

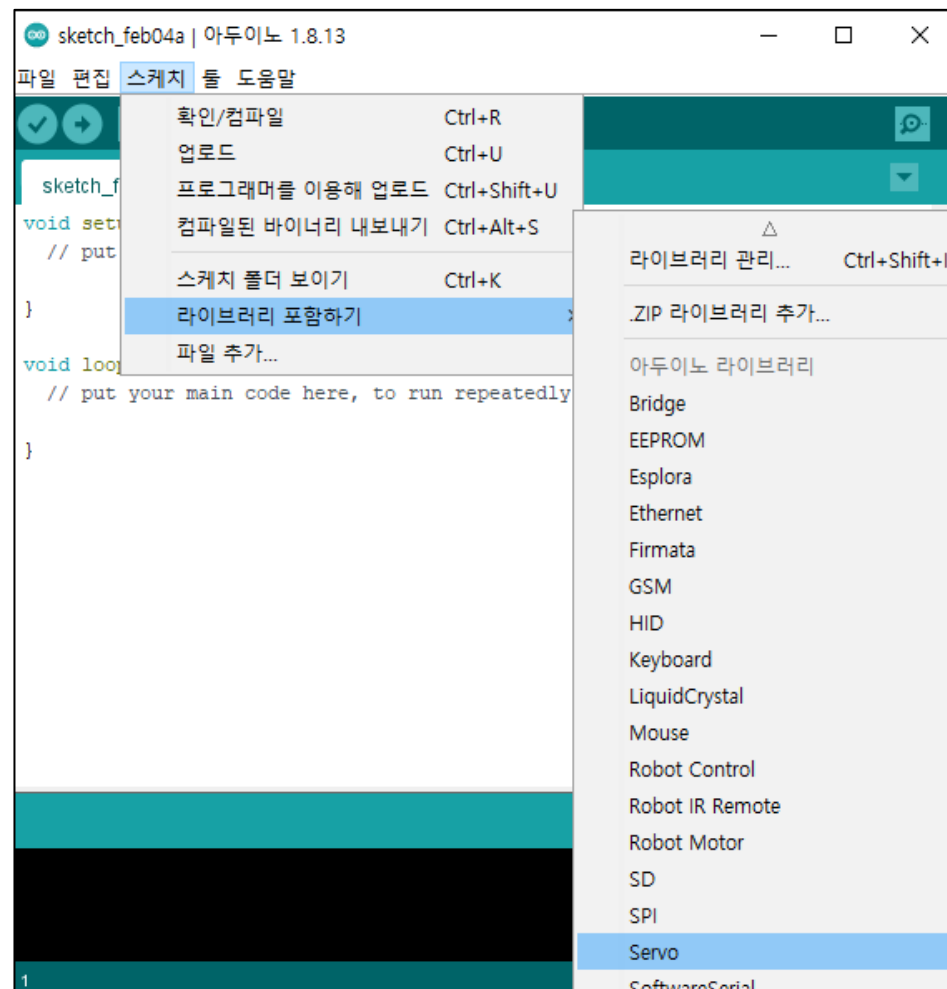
특징

- 위치 제어
- 신속, 정확

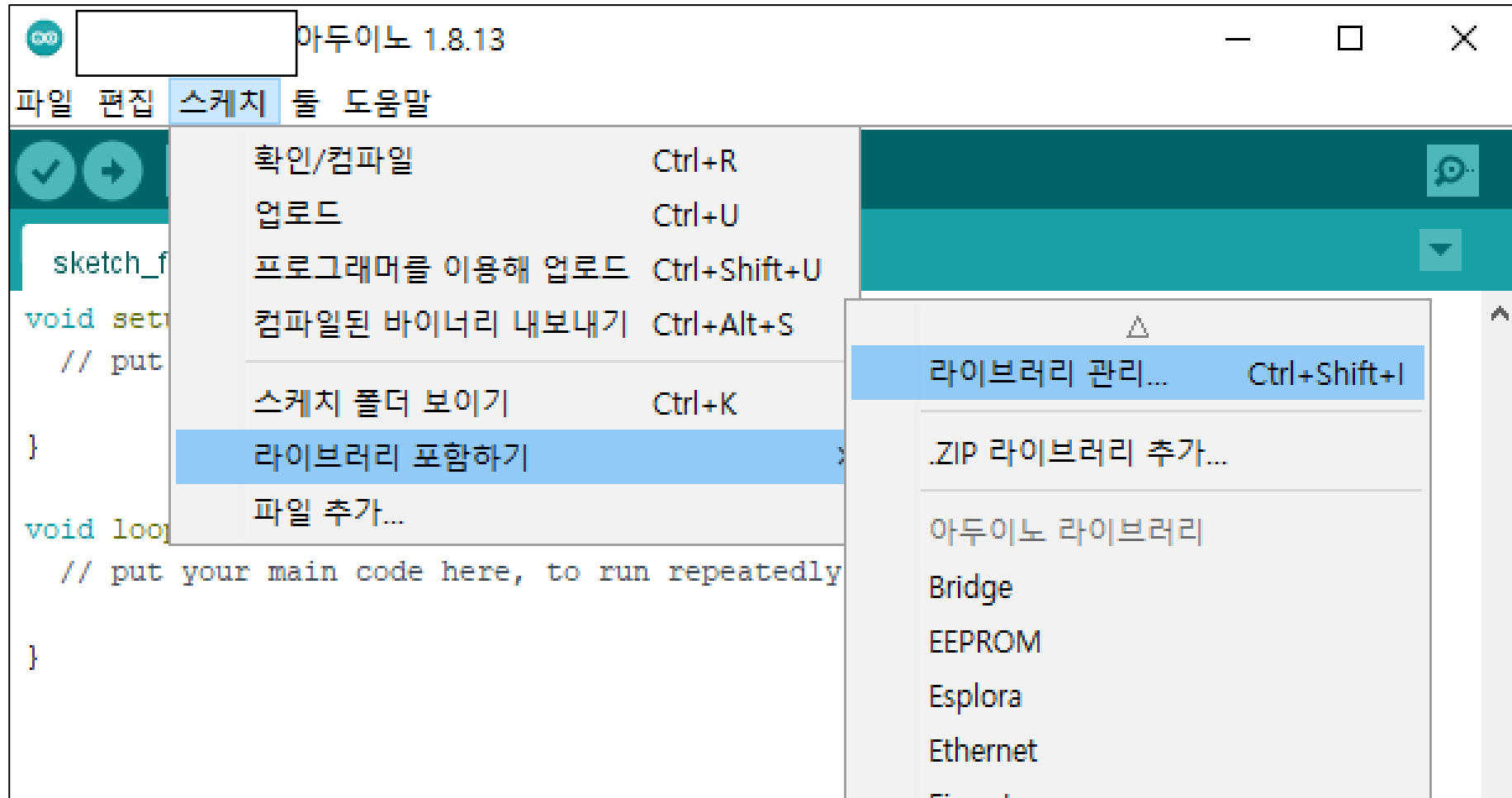
- 회전자 돌아가는 각의 범위제한
- 큰 회전력을 가질 것
- 회전자의 관성모멘트가 작을 것(곧바로 정지, 곧바로 회전)



External Library

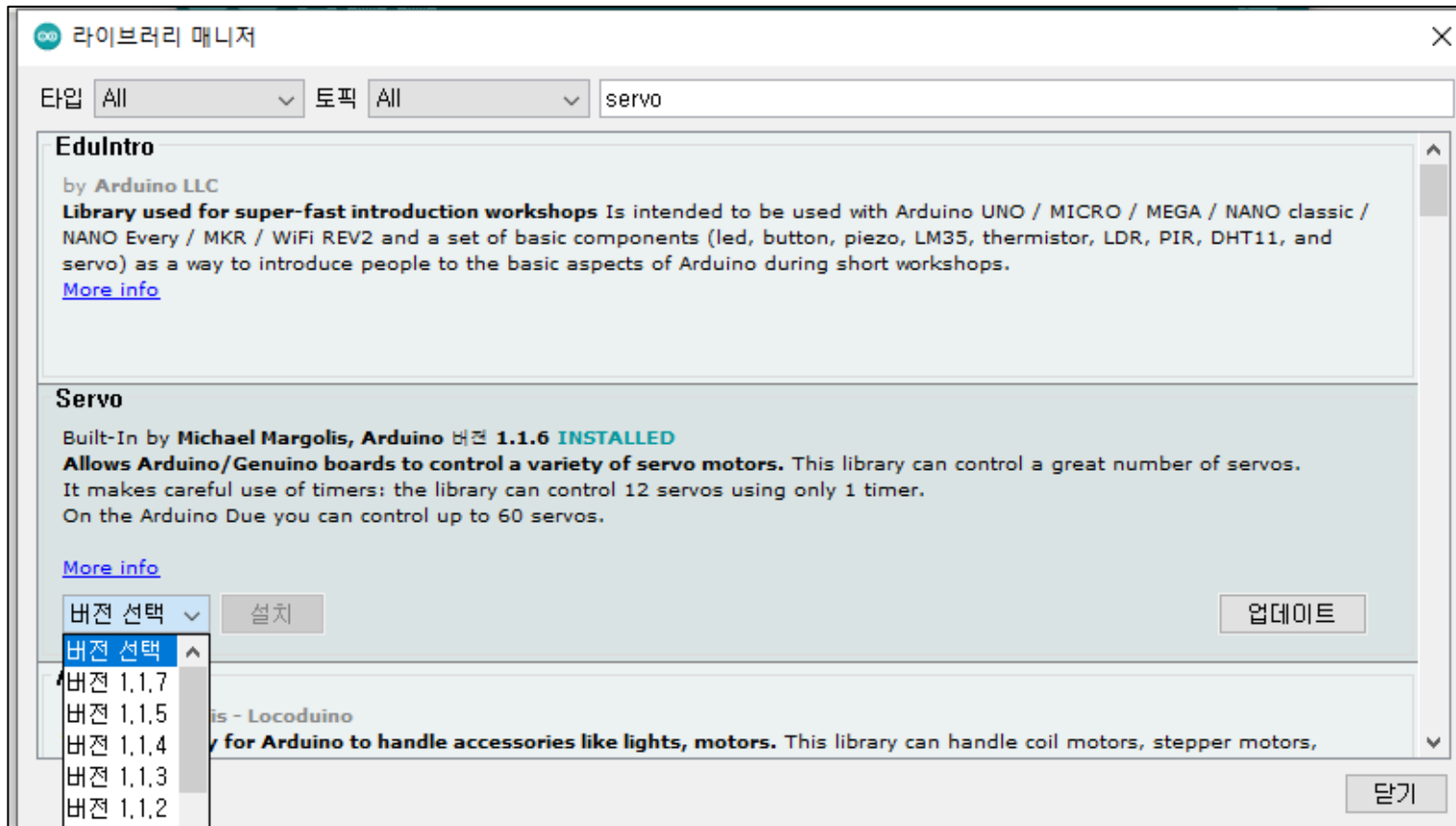


External Library



External Library

1. Adding Libraries from Library Manager

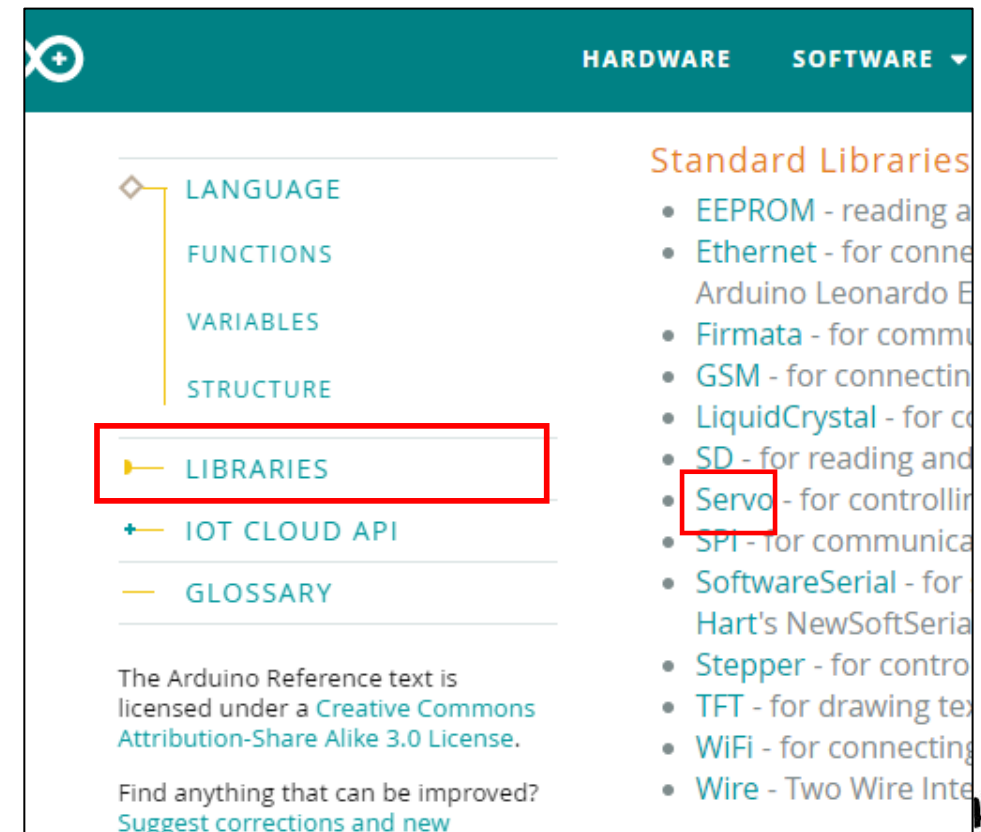
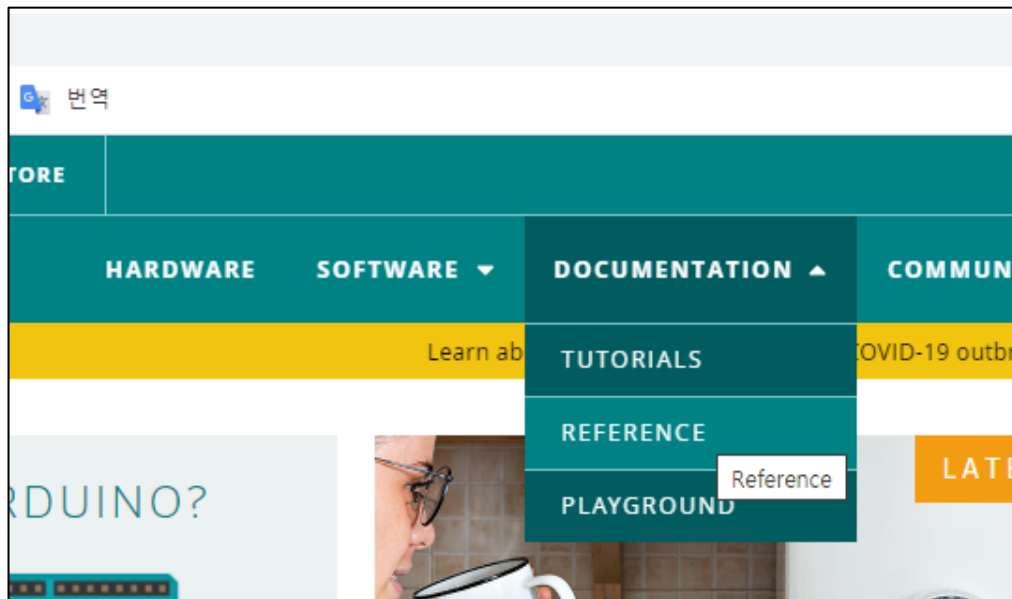


버전 선택 후 설치

External Library

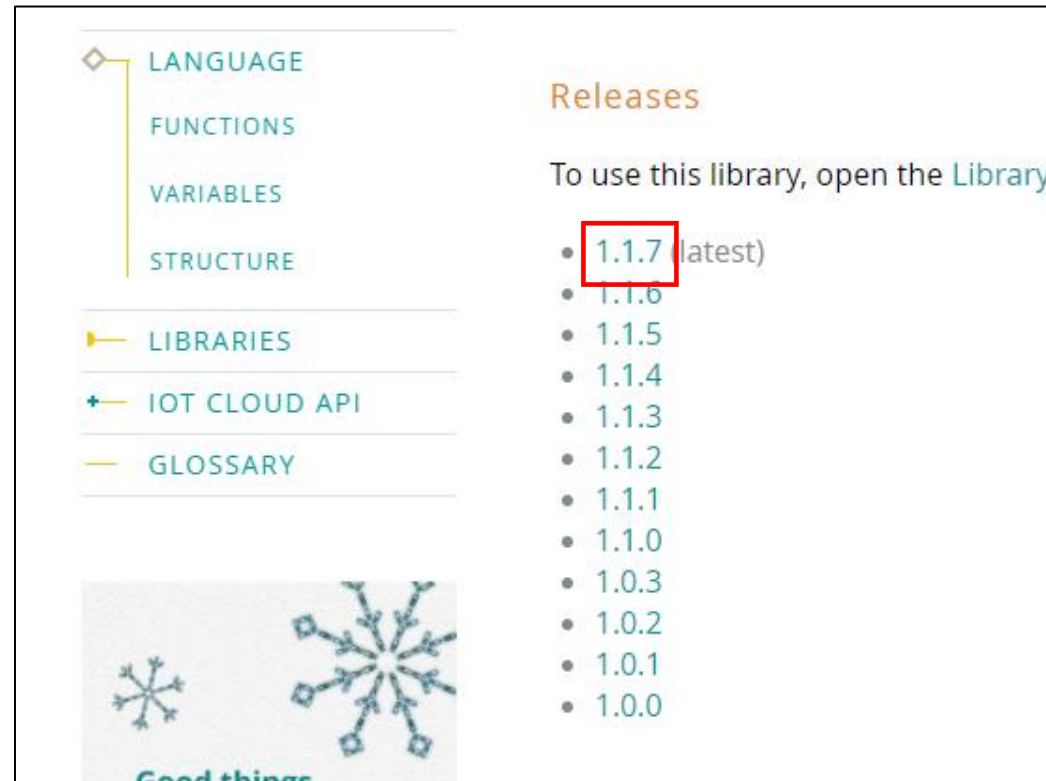
2. Manually Download from Github, Arduino Homepage etc.

<https://www.arduino.cc/reference/en/libraries/>



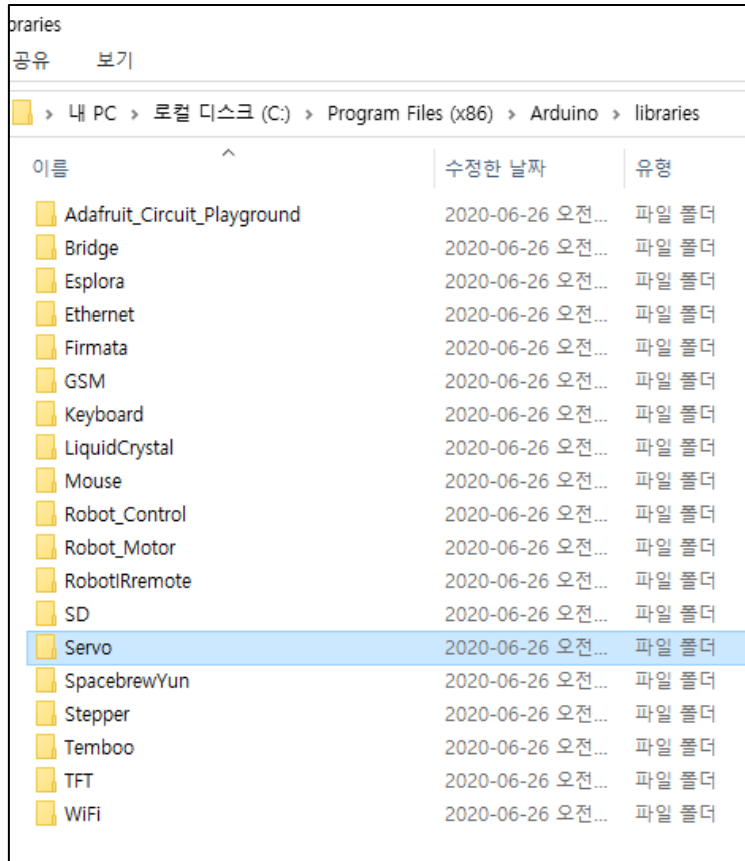
External Library

2. Manually Download from Github, Arduino Homepage etc.

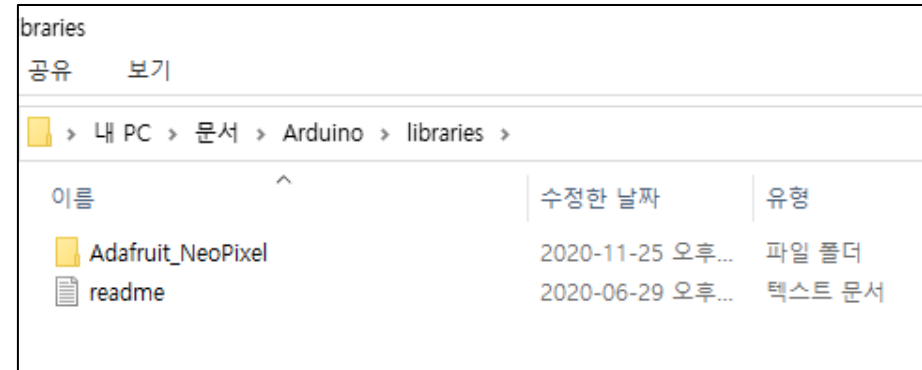


External Library

2. Manually Download from Github, Arduino Homepage etc.



이름	수정된 날짜	유형
Adafruit_Circuit_Playground	2020-06-26 오전...	파일 폴더
Bridge	2020-06-26 오전...	파일 폴더
Esplora	2020-06-26 오전...	파일 폴더
Ethernet	2020-06-26 오전...	파일 폴더
Firmata	2020-06-26 오전...	파일 폴더
GSM	2020-06-26 오전...	파일 폴더
Keyboard	2020-06-26 오전...	파일 폴더
LiquidCrystal	2020-06-26 오전...	파일 폴더
Mouse	2020-06-26 오전...	파일 폴더
Robot_Control	2020-06-26 오전...	파일 폴더
Robot_Motor	2020-06-26 오전...	파일 폴더
RobotIRremote	2020-06-26 오전...	파일 폴더
SD	2020-06-26 오전...	파일 폴더
Servo	2020-06-26 오전...	파일 폴더
SpacebrewYun	2020-06-26 오전...	파일 폴더
Stepper	2020-06-26 오전...	파일 폴더
Temboo	2020-06-26 오전...	파일 폴더
TFT	2020-06-26 오전...	파일 폴더
WiFi	2020-06-26 오전...	파일 폴더



이름	수정된 날짜	유형
Adafruit_NeoPixel	2020-11-25 오후...	파일 폴더
readme	2020-06-29 오후...	텍스트 문서

Servo.h

```
sketch_feb04a | 아두이노 1.8.13
파일 편집 스케치 툴 도움말
sketch_feb04a $
#include <Servo.h>

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

```
Servo.h - Microsoft Visual Studio
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
Servo.h
기타 파일 (전역 범위)
47
48 #ifndef Servo_h
49 #define Servo_h
50
51 #include <inttypes.h>
52
53 /* ... */
54
55 // Architecture specific include
56 #if defined(ARDUINO_ARCH_AVR)
57 #include "avr/ServoTimers.h"
58 #elif defined(ARDUINO_ARCH_SAM)
59 #include "sam/ServoTimers.h"
60 #elif defined(ARDUINO_ARCH_SAMD)
61 #include "samd/ServoTimers.h"
62
```

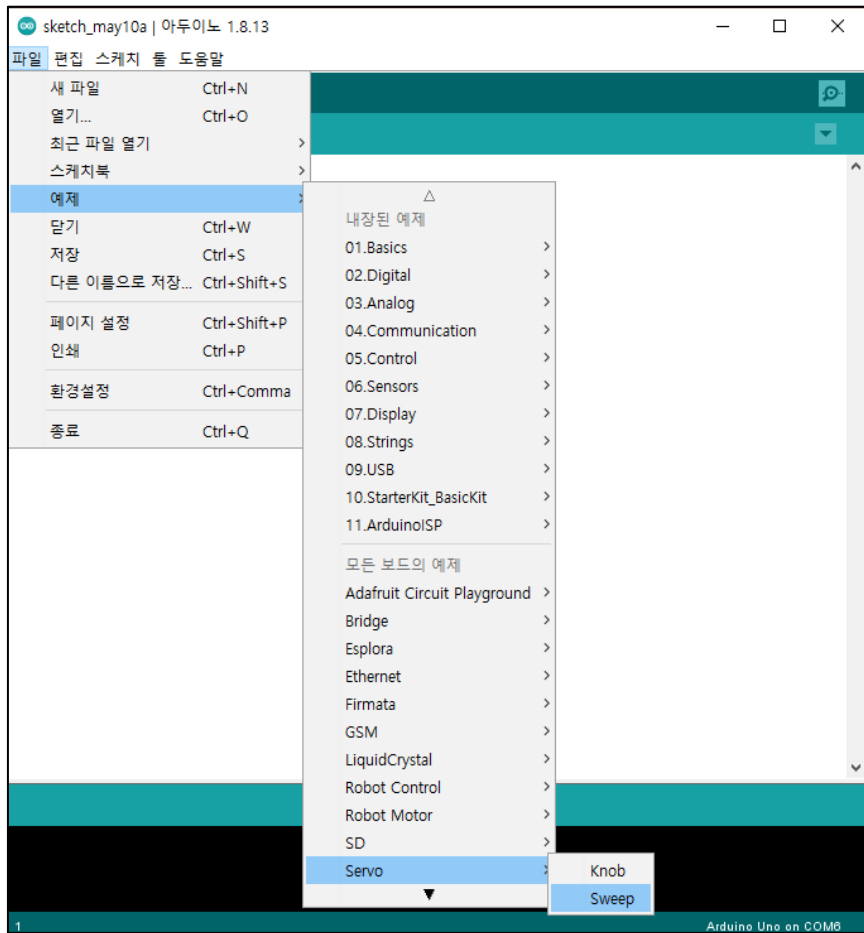
(X)

헤더 파일 개발자가 이렇게
친절하게 설명을 써 놓으셨다

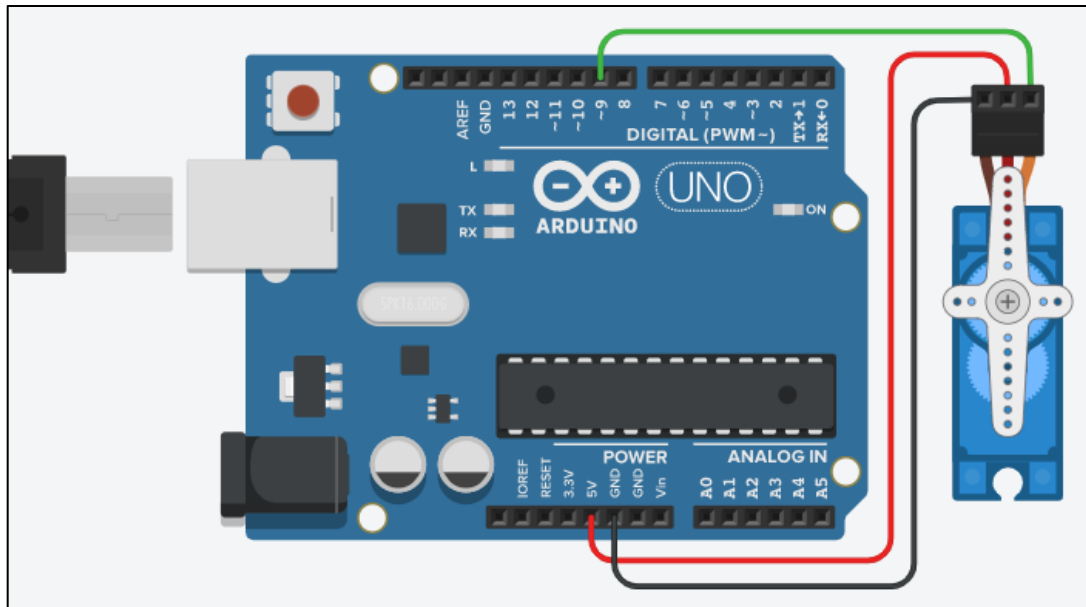
```
31
32 The methods are:
33
34 Servo - Class for manipulating servo motors connected to Arduino pins.
35
36 attach(pin ) - Attaches a servo motor to an i/o pin.
37 attach(pin, min, max ) - Attaches to a pin setting min and max values in microseconds
38 default min is 544, max is 2400
39
40 write() - Sets the servo angle in degrees. (invalid angle that is valid as pulse in microseconds is treated as microseconds)
41 writeMicroseconds() - Sets the servo pulse width in microseconds
42 read() - Gets the last written servo pulse width as an angle between 0 and 180.
43 readMicroseconds() - Gets the last written servo pulse width in microseconds. (was read_us() in first release)
44 attached() - Returns true if there is a servo attached.
45 detach() - Stops an attached servos from pulsing its i/o pin.
46 */
```

Servo in TinkerCAD

or



Servo in TinkerCAD



```
문자 1 (Arduino Un
10
11 #include <Servo.h>
12
13 int pos = 0;
14 Servo servo_9;
15
16 void setup()
17 {
18   servo_9.attach(9, 500, 2500);
19 }
20
21
22 void loop()
23 {
24   // sweep the servo from 0 to 180 degrees in steps
25   // of 1 degrees
26   for (pos = 0; pos <= 180; pos += 1) {
27     // tell servo to go to position in variable 'pos'
28     servo_9.write(pos);
29     // wait 15 ms for servo to reach the position
30     delay(15); // Wait for 15 millisecond(s)
31   }
32   for (pos = 180; pos >= 0; pos -= 1) {
33     // tell servo to go to position in variable 'pos'
34     servo_9.write(pos);
35     // wait 15 ms for servo to reach the position
36     delay(15); // Wait for 15 millisecond(s)
37   }
38 }
```

int i; 하듯이
Servo servo_9;

Ultrasonic Sensor

- mech9917@sogang.ac.kr

What is Ultrasonic?

Ultrasonic Sensor HC-SR04

- 40kHz의 주파수 발생. 최대 4.5m까지 측정 가능



특징

- Vcc, Trig, Echo, GND 4pins
- Trig : 초음파 발사
- Echo : 초음파 수신



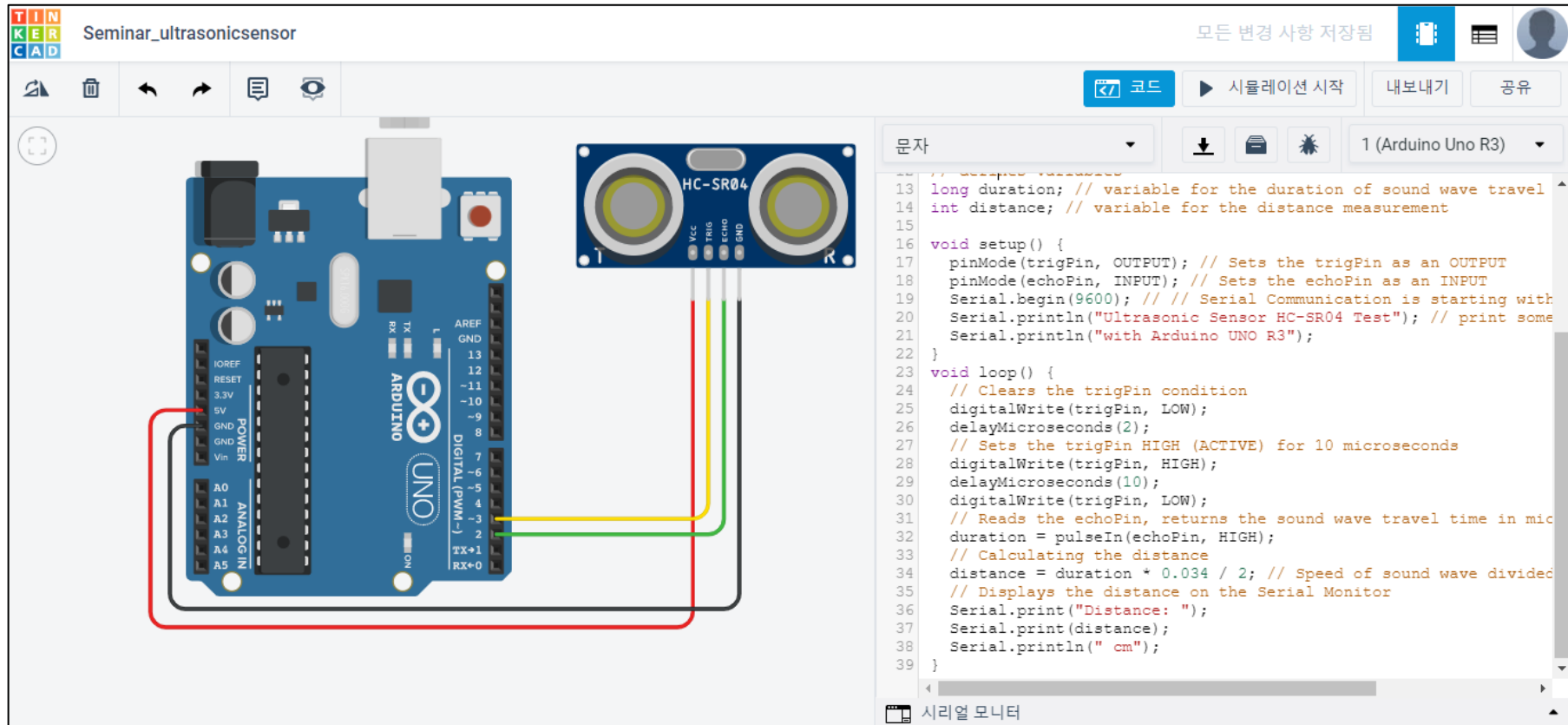
HC-SR04 in TinkerCAD

다리가 4개 달린 것으로!!



HC-SR04 in TinkerCAD

ultrasonicsensor.txt 가져오기



Distance Conversion



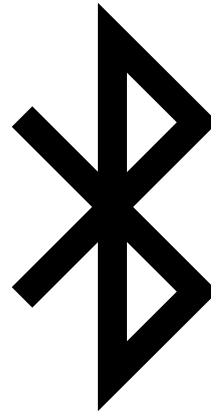
```
duration = pulseIn(echoPin, HIGH);  
distance = duration * 0.034 / 2;
```

$\text{distance(cm)} = \text{duration(microseconds)} * 10^{-6} \text{s/microseconds} * 100\text{cm/m} * 340\text{m/s} * 1/2$ (왕복)

Next Seminar



Bluetooth Remote Control



Q&A