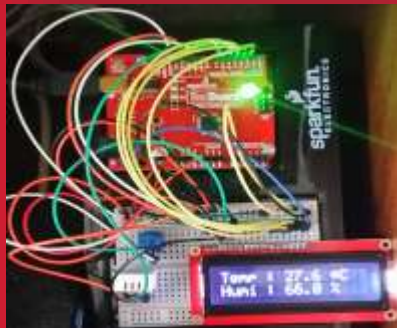


# Arduino-basic

## [wk12]

## Motor & IR



Learn how to code Arduino from scratch

Comsi, INJE University

1<sup>st</sup> semester, 2023

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)



# My ID (ARnn, github repo)

AR01	강동하
AR02	정재윤
AR03	유석진
AR04	정창민
AR05	정희서
AR06	유동기
AR07	장세진
AR08	정호기

위의 **id**를 이용해서 **github**에 **repo**를 만드시오.



# [Practice]

## ◆ [wk11]

- **Arduino : Analog input II**
- **Complete your project**
- **Submit folder : ARnn\_Rpt10**

# wk11 : Practice-10 : ARnn\_Rpt10

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

제출폴더명 : **ARnn\_Rpt10**

### 제출할 파일들

- ① **ARnn\_waterlevel.png**
- ② **ARnn\_waterlevel.ino**
- ③ **ARnn\_joystick.ino**
- ④ **ARnn\_joystick.png**
- ⑤ **ARnn\_sound\_bar.png**
- ⑥ **ARnn\_sound\_monitor.png**
- ⑦ **Arnn\_TMP36.png**
- ⑧ **\*.ino**



# 7. Motor driving



# 7. 모터 구동

7.1 스텝모터 구동

7.2 서보모터 구동

7.3 DC모터 구동



# 7.1

# 스텝모터



## 스텝모터 (Step motor, Stepper, Stepping motor)

표 7.1 실험에 사용할 펄스(1-2상 여자방식)

선 색	CW방향							
	1	2	3	4	5	6	7	8
5 별강	+	+	+	+	+	+	+	+
4 주황	-	-						-
3 노랑		-	-	-				
2 분홍				-	-	-		
1 파랑						-	-	-



그림 7. 1 실험에 사용할 스텝모터 28BYJ-48와 사양표

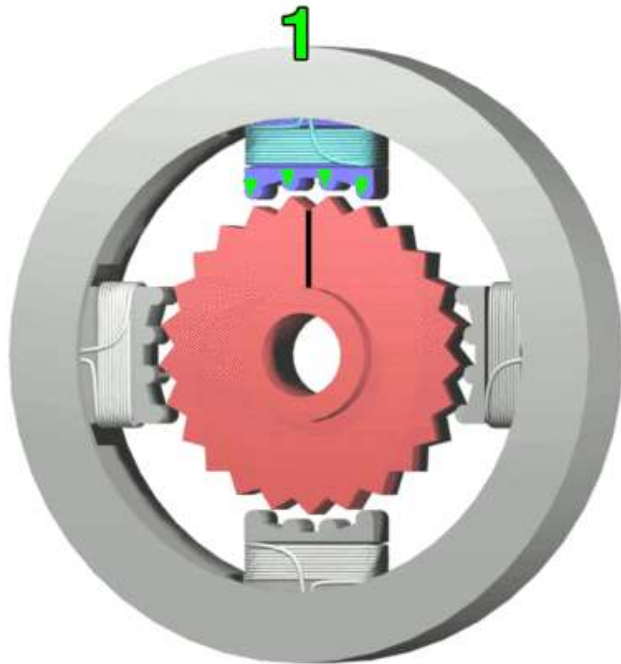
- ✓ 다수의 입력핀에 일정한 패턴이 있는 펄스를 순서대로 인가하면 정해진 방향과 각도만큼 움직이는 모터
- ✓ 펄스의 주파수에 따라 모터의 속도나 토크가 결정
- ✓ Arduino나 마이크로프로세서에서 나오는 신호를 달링턴 드라이브 IC(Darlington drive IC)나 스텝 모터용 드라이버 IC에 연결하여 사용



그림 7. 2 실험에 사용할 스텝모터 드라이버 모듈



# 7.1 스텝모터: 구동 원리



## 스텝모터란

- 모터의 회전을 잘게 쪼개서 쪼개진 조각(스텝)을 이용해서 제어하는 모터.
- 스텝모터는 펄스(Puls)에 의해 디지털적으로 제어하기 때문에 정밀한 제어가 가능하고, 아두이노, 라즈베리파이 등으로 동작시키기에 편리함.
- 제어부가 비교적 간단해서 가격이 저렴하고 브러시리스(Brushless)모터이기 때문에 오래 사용할 수 있다.

[https://en.wikipedia.org/wiki/Stepper\\_motor](https://en.wikipedia.org/wiki/Stepper_motor)

<https://m.blog.naver.com/3demp/220976664782>

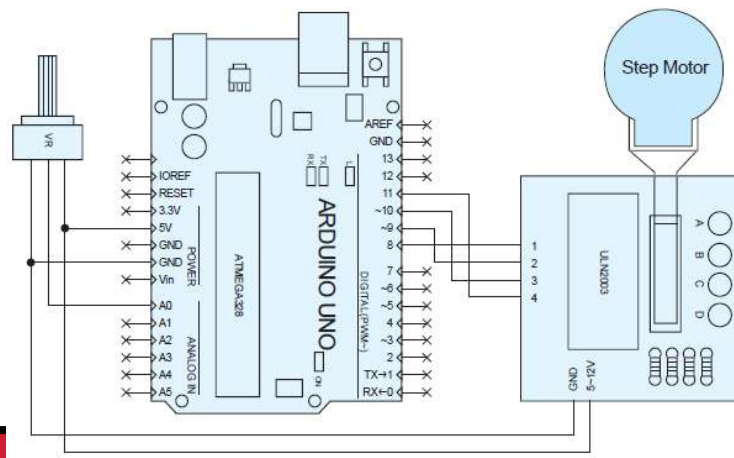
스텝모터 구동 원리

## EX 7.1

## 스텝모터 구동 (1/3)

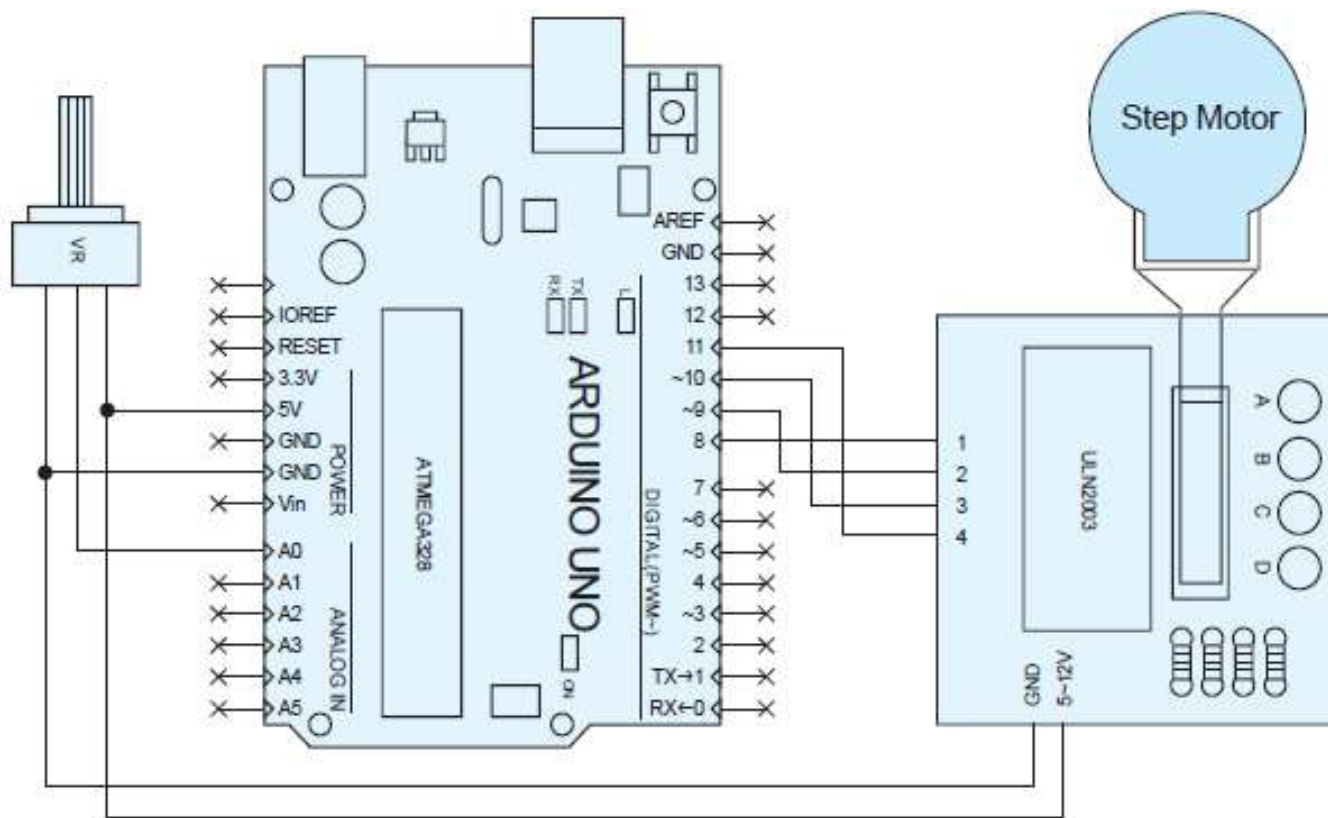
- 실습목표**
1. 포텐쇼미터를 이용하여 아날로그 입력을 받는다.
  2. 포텐쇼미터가 중간 위치이면 모터 Stop, CW로 회전시키면 CW로 회전, CCW로 회전시키면 CCW로 회전시킨다
  3. 포텐쇼미터의 각도에 따라서 모터의 속도가 조절되며 시리얼 모니터로 확인한다.

- Hardware**
1. 포텐쇼미터의 1, 3번핀을 Arduino의 5V, GND에 연결한다.
  2. 포텐쇼미터의 2번핀을 Arduino의 아날로그입력핀 A0에 연결한다.
  3. 스텝모터 드라이브 모듈의 5~12V핀과 GND 핀을 5V와 GND에 연결한다.
  4. Arduino의 8~11번핀을 스텝모터 드라이브 모듈의 1~4번핀에 연결한다.



EX 7.1

스텝모터 구동 (1/3)



## 스텝모터 구동 code

```

stepper_test
1 #include <Stepper.h>
2
3 const int stepsPerRevolution = 1024;
4
5 Stepper myStepper(stepsPerRevolution, 11, 9, 10, 8);
6
7 void setup() {
8   Serial.begin(9600);
9 }
10
11 void loop() {
12   int sensorReading = analogRead(A0);
13   int motorSpeed = map(sensorReading, 0, 1023, 0, 25);
14
15   if (motorSpeed > 0) {
16     myStepper.setSpeed(motorSpeed);
17     myStepper.step(stepsPerRevolution / 100); // CW vs. CCW
18   }
19
20   Serial.print("A0 = ");
21   Serial.println(sensorReading);
22   delay(500);
23 }

```

## EX 7.1

## 스텝모터 구동 (2/3)

### Commands

- Serial.begin(전송속도)

시리얼 통신 포트를 컴퓨터와 연결한다. 전송속도는 bps(bits per sec)로 일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다.

- Serial.print(전송내용)

괄호 안의 내용을 시리얼 통신으로 전송한다. 따옴표로 구분된 부분은 텍스트를 직접 전송하고 따옴표 없이 변수를 써주면 변수의 값이 전송된다.

- Serial.println(전송내용)

‘Serial.print’와 같으나 전송 뒤 줄 바꿈을 한다.

- **delayMicroseconds(지연시간)**

지연시간에는 잠시 동작을 지연시키기 위한 값을 넣는다. 마이크로초 단위로 넣는다.

- void 함수(변수1, 변수2, ...){

};

‘함수(변수1, 변수2)’를 이용하여 ‘{ }’ 내의 명령을 호출하여 사용한다. ‘변수1’과 ‘변수2’등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){ }

변수의 시작 값부터 조건이 만족하는 경우 ‘{ }’ 내의 명령을 수행한다. ‘변수의 증분’에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

## EX 7.1

## 스텝모터 구동 (3/3)

- Sketch 구성**
1. 포텐쇼미터값을 아날로그핀 0번으로 입력받는다.
  2. 모터의 출력신호로 사용할 핀을 설정한다.
  3. 스텝사이에는 마이크로세컨드 ( $\mu s$ ) 단위로 4500~1000의 범위를 갖는다. 각 스텝 사이에 이 값을 넣어 지연시켜 펄스를 만들어 준다. (시간지연이 클수록 속도가 느려진다.)
  4. 포텐쇼미터값이 중간에 왔을 때는 모터를 정지시키고 포텐쇼미터의 방향과 회전각에 따라 모터를 회전시킨다. → 속도와 방향을 조정하는 모터
  5. 현재 모터의 상태를 시리얼통신으로 전송한다.
- 실습 결과**
1. 시리얼 모니터를 통하여 모터의 회전 방향, 속도 (백분율)를 나타내는 메시지가 출력된다.
  2. 포텐쇼미터를 조절하면 모터의 방향과 속도를 조절할 수 있다.

## 7.1.4 스텝모터 : code-1

```
ex_7_1
1 /*
2  예제 7.1
3  스텝모터 구동
4  */
5
6 // 스텝 모터 신호핀 설정
7 int motorPin1 = 8;
8 int motorPin2 = 9;
9 int motorPin3 = 10;
10 int motorPin4 = 11;
11
12 // 포텐쇼미터 핀 설정
13 int potentiometerPin = 0;
14
15 // stop 구간 폭 설정
16 int stopRange = 100;
17
18 // 모터 속도 관련 변수 설정
19 int motorSpeed; // 스텝 사이의 지연시간으로서 4500~10000의 범위를 갖는다.
20 int motorSpeedPercent; // 속도를 0~100%로 나타낸다.
21
22 // 스텝 모터의 스텝 설정
23 // 0~7은 동작 신호, 8번째는 모터 정시지 신호
24 int steps[] = {B1000, B1100, B0100, B0110, B0010, B0011, B0001, B1001, B0000};
25
```

## 7.1.4 스텝모터 : code-2

```

37 void loop(){
38   // 포텐쇼미터 값을 읽어옴
39   int potentiometer = analogRead(potentiometerPin);
40
41   // CW로 회전시 모터를 CW방향으로 회전시킨다.
42   if(potentiometer >= 512+(stopRange/2)){
43     //모터의 속도를 계산한다.
44     motorSpeed = map(potentiometer, 512+(stopRange/2), 1023, 4500, 1000);
45     // 모터의 속도를 백분율로 변환시킨다.
46     motorSpeedPercent = map(motorSpeed, 4500, 1000, 1, 100);
47     // 시리얼 통신 메시지를 출력한다.
48     Serial.print("CW Motor Speed: ");
49     Serial.print(motorSpeedPercent);
50     Serial.println("%");
51     // CW로 회전시킨다.
52     clockwise();
53   }

```

```

54   // CCW로 회전시 모터를 CW방향으로 회전시킨다.
55   else if(potentiometer <= 512-(stopRange/2)){
56     //모터의 속도를 계산한다.
57     motorSpeed = map(potentiometer, 512-(stopRange/2), 0, 4500, 1000);
58     // 모터의 속도를 백분율로 변환시킨다.
59     motorSpeedPercent = map(motorSpeed, 4500, 1000, 1, 100);
60     // 시리얼 통신 메시지를 출력한다.
61     Serial.print("CCW Motor Speed: ");
62     Serial.print(motorSpeedPercent);
63     Serial.println("%");
64     // CCW로 회전시킨다.
65     counterClockwise();
66   }
67   // 중간에 위치 했을 경우 정지시킨다.
68   else{
69     Serial.println("Motor Stop");
70     motorStop();
71   }
72 }

```



## 7.1.3 스텝모터 : code-3

```

74 void counterClockwise(){
75     // 0~7 번째 신호를 순차적으로 출력시킨다.
76     for(int i = 0; i < 8; i++)
77     {
78         motorSignalOutput(i);
79         delayMicroseconds(motorSpeed);
80     }
81 }
82
83 void clockwise(){
84     // 7~0 번째 신호를 순차적으로 출력시킨다.
85     for(int i = 7; i >= 0; i--)
86     {
87         motorSignalOutput(i);
88         delayMicroseconds(motorSpeed);
89     }
90 }

```

```

92 void motorStop(){
93     // 정지신호를 출력시킨다.
94     motorSignalOutput(8);
95 }
96
97 void motorSignalOutput(int out)
98 {
99     // out 변수에 해당하는 모터 시그널을 출력한다.
100    digitalWrite(motorPin1, bitRead(steps[out], 0));
101    digitalWrite(motorPin2, bitRead(steps[out], 1));
102    digitalWrite(motorPin3, bitRead(steps[out], 2));
103    digitalWrite(motorPin4, bitRead(steps[out], 3));
104 }

```

bitRead(x, n)

매개변수

x: 읽을 숫자

n: 읽을 비트, LSB(맨 오른쪽 비트)가 0, 왼쪽으로 갈수록 1씩 증가

// 스텝 모터의 스텝 설정

// 0~7은 동작 신호, 8번째는 모터 정지 신호

```
int steps[] = {B1000, B1100, B0100, B0110, B0010, B0011,
B0001, B1001, B0000};
```

# 7.1.5 스텝모터 : DIY

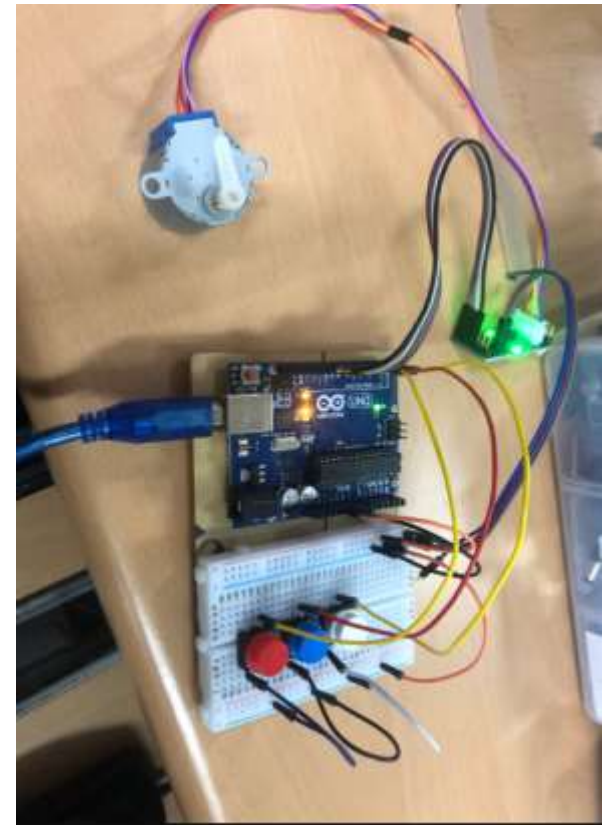
## DIY

## 응용 문제

1. 3개의 스위치를 디지털 입력으로 연결하자.
2. CW, STOP, CCW 기능을 하도록 스케치를 작성하자.
3. STOP일 경우 모든 신호는 스텝모터로 입력되지 않아야 한다.

아두이노 회로가 포함된 동작 사진을  
**ARnn\_step\_motor.png**

아두이노 스케치 코드를  
**ARnn\_step\_motor.ino** 로  
저장...





# 7.2

## 서보모터



## 서보모터 (Servo Motor)

- ✓ 기계적인 위치, 속도, 가속도 등을 제어하는 모터
- ✓ 산업용 서보모터는 로봇의 관절, 공작 기계의 위치제어 등에 사용
- ✓ RC용 서보모터는 RC 자동차나 RC 비행기에 사용



그림 7. 4 실험에 사용할 소형 RC 서보모터 (SG90)



그림 7. 3 산업용 서보모터(a)와 RC용 서보모터(b)

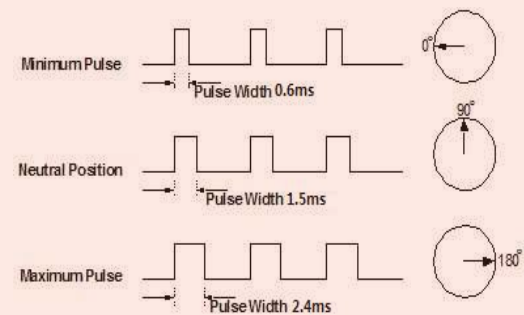


그림 7. 5 PWM 신호와 RC 서보모터의 회전각

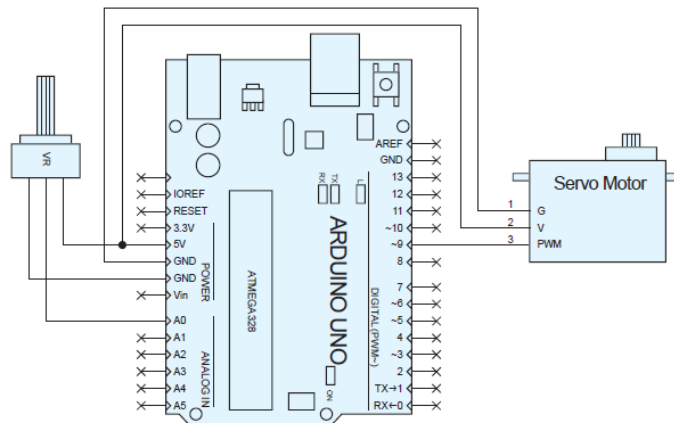
**PWM 펄스의 길이로 각도를 조정한다.**

## EX 7.2

## 서보모터 구동 (1/3)

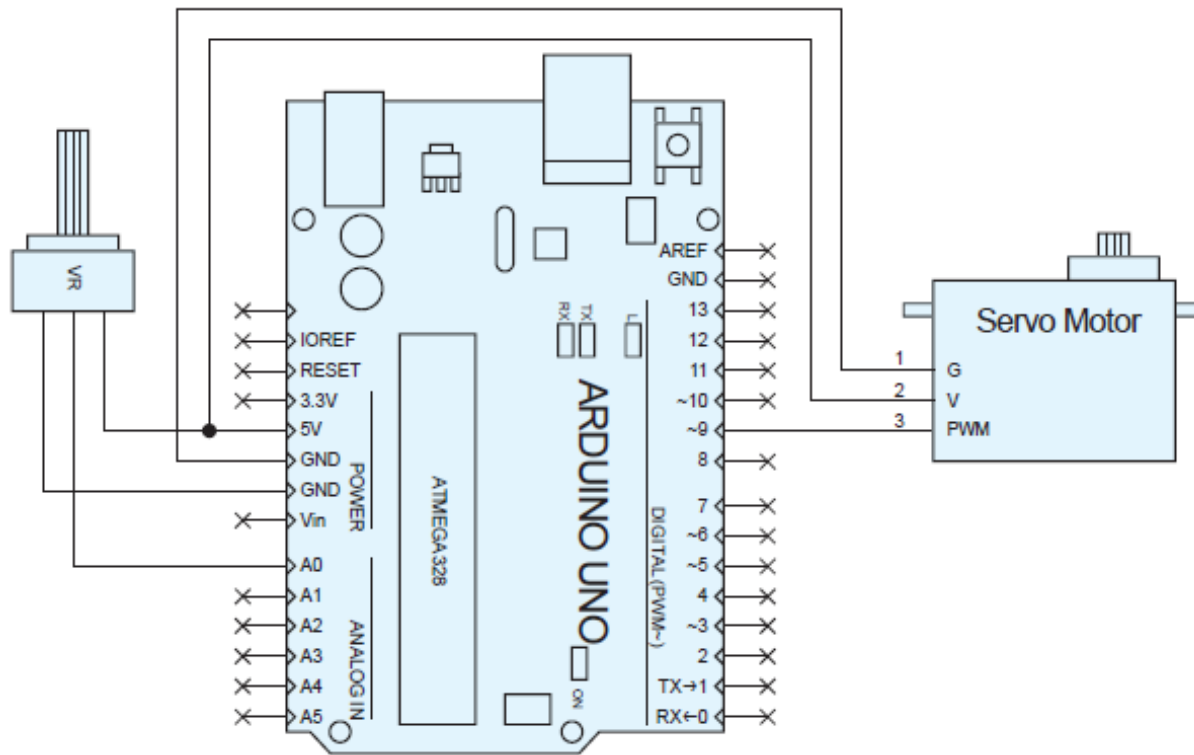
- 실습목표**
1. 소형 RC용 서보모터를 구동한다.
  2. 포텐쇼미터의 각도에 따라서 서보모터의 각도를 조절한다.
  3. 현재 각도를 시리얼 통신으로 전송한다.

- Hardware**
1. 포텐쇼미터의 1, 3번핀을 Arduino의 5V, GND에 연결한다.
  2. 포텐쇼미터의 2번핀을 Arduino의 아날로그입력핀 A0에 연결한다.
  3. 서보모터의 V(적색)와 GND(검정 혹은 갈색)핀을 Arduino의 5V와 GND에 연결한다.
  4. Arduino의 9번핀을 서보모터의 PWM핀(흰색 혹은 주황)과 연결한다.



EX 7.2

서보모터 구동 (1/3)



## EX 7.2

## 서보모터 구동 (2/3)

### Commands • Serial.begin(전송속도)

시리얼 통신 포트를 컴퓨터와 연결한다. 전송속도는 bps(bits per sec)로 일반적으로 9600으로 설정한다. 19200, 57600, 115200 등의 값을 설정할 수 있다.

### • Serial.print(전송내용)

괄호 안의 내용을 시리얼 통신으로 전송한다. 따옴표로 구분된 부분은 텍스트를 직접 전송하고 따옴표 없이 변수를 써주면 변수의 값이 전송된다.

### • Serial.println(전송내용) 'Serial.print'와 같으나 전송 뒤 줄 바꿈을 한다.

### • 서보모터이름.attach(핀번호)

이름을 설정한 서보모터를 핀번호에 설정한다.

### • 서보모터이름.attach(핀번호, 최소펄스, 최대펄스)

이름을 설정한 서보모터를 핀번호에 설정한다. 서보모터가 동작하는 최소 펄스와 최대 펄스를 마이크로세컨드 단위로 설정한다.

### • 서보모터이름.write(각도)

이름을 설정한 서보모터를 정해진 각도로 위치시킨다.

### EX 7.2

### 서보모터 구동 (3/3)

- Sketch 구성**
1. 서보모터 라이브러리를 추가한다.
  2. 서보모터 이름을 설정하고 9번핀을 서보모터 출력으로 사용한다.
  3. 서보모터의 회전범위를 설정한다 라이브러리의 attach()함수를 사용한다.
  4. 포텐쇼미터값을 아날로그 0번핀으로 입력받아 0~1023 범위의 포텐쇼미터값 범위를 0~180도로 환산하여 서보모터를 동작시킨다. → **회전각도 정밀 조정**

- 실습 결과**
1. 포텐쇼미터의 회전각에 따라 서보모터가 회전한다.
  2. 포텐쇼미터 값이 변화하면 시리얼 통신으로 회전각을 전송한다.
  3. 현재의 각도를 LCD 모듈로 출력한다.



## 7.2.4 서보모터: code

ex\_7\_2

```

1 /*
2  예제 7.2
3  서보모터 구동
4 */
5
6 // 서보모터 라이브러리 불러오기
7 #include <Servo.h>
8
9 // 서보모터 이름 설정
10 Servo motor1;
11
12 // 서보 모터 신호핀 설정
13 int servoMotorPin = 9;
14
15 // 포텐쇼미터 핀 설정
16 int potentiometerPin = 0;
17
18 // 모터 각도 변수 설정
19 int motorAngle;
20 int motorAngleOld;
21

```

```

22 void setup() {
23
24   // 서보모터 설정. 0.6ms 부터 2.4ms 범위로 설정
25   motor1.attach(servoMotorPin, 600, 2400);
26
27   // 시리얼 통신 설정
28   Serial.begin(9600);
29 }
30
31 void loop(){
32   // 포텐쇼미터 값을 읽어옴
33   int potentiometer = analogRead(potentiometerPin);
34
35   // 포텐쇼미터 값을 모터 각도로 변환한다
36   motorAngle = map(potentiometer, 0, 1023, 0, 180);
37
38   // 모터에 각도값을 전달한다
39   motor1.write(motorAngle);
40
41   // 이전각도와 현재 각도가 같지 않으면 시리얼 모니터에 각도를 출력한다.
42   if(motorAngle != motorAngleOld){
43     Serial.print("Servo Motor Angle is: ");
44     Serial.println(motorAngle);
45   }
46
47   // 현재의 모터 각도를 저장한다.
48   motorAngleOld = motorAngle;
49
50   delay(20);
51
52 }

```

DIY

1. 좌 우 두 개의 스위치 입력을 받는다.

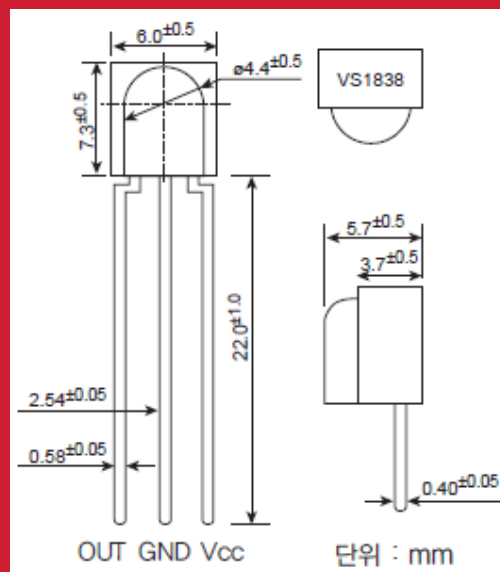
응용 문제

2. 스위치를 누를 때 마다 해당 방향으로 회전하게 한다.

아두이노 스케치 코드를  
**ARnn\_servo\_motor.ino** 로  
저장...



# 8. Infrared remote



## 8. Infrared remote

8.1 적외선 리모컨 코드 읽기

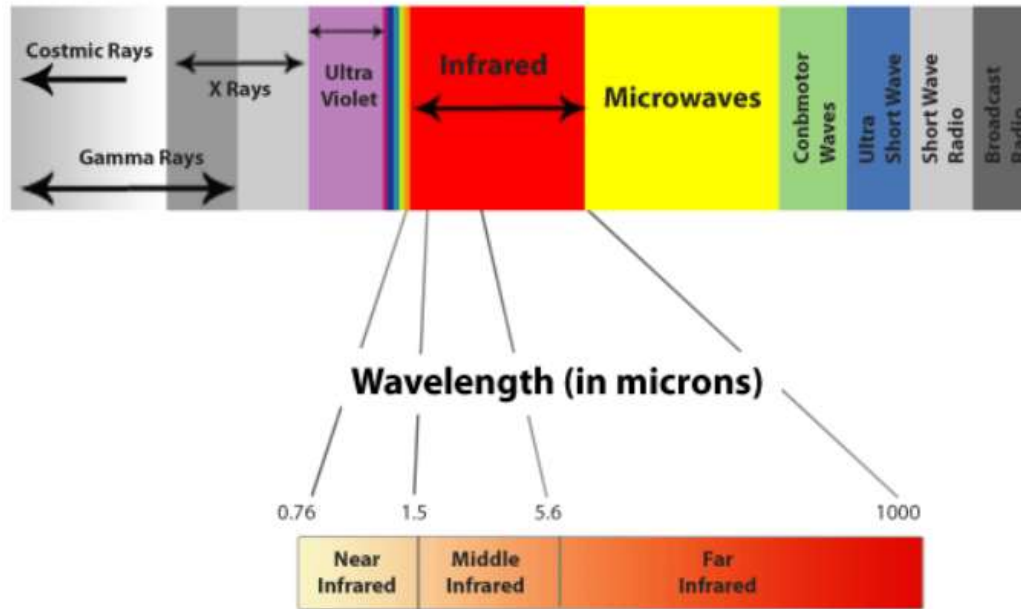
8.2 적외선 리모컨을 이용한 LED 제어



# 8.1

## 적외선 리모컨





전자기파

<적외선의 파장은 0.75um부터 1,000um사이에 위치합니다>



<https://kocoafab.cc/tutorial/view/703>

# 8.1 적외선 리모컨

## 적외선 리모컨



그림 8. 1 실험에 사용할 리모컨과 수신부

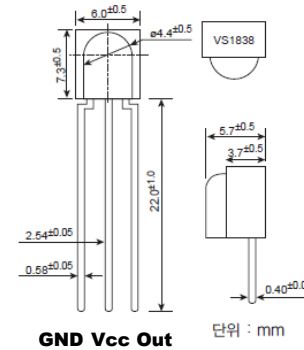


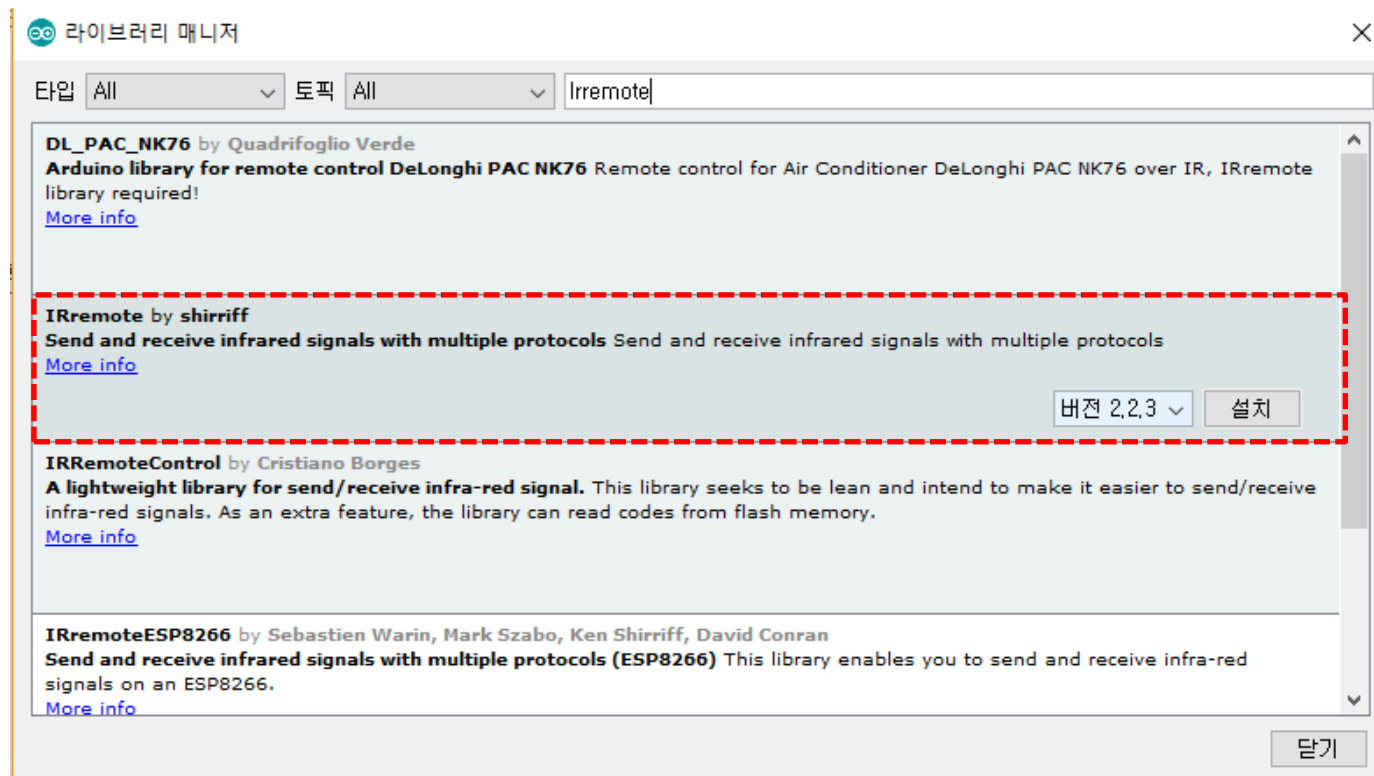
그림 8. 2 리모컨 수신부의 핀 설정

- ✓ TV, 에어컨 등 대부분의 가전제품에 사용
- ✓ 적외선 램프의 점멸을 이용하여 데이터 송 수신
- ✓ 송신 시 캐리어 주파수로 변조하여 전송. 수신부와 캐리어주파수가 일치하여야함

※ 실험에 사용할 라이브러리를 하기의 주소에서 다운받아 설치할 것

<https://github.com/shirriff/Arduino-IRremote>

## 적외선 리모컨 – 라이브러리 설치



✓ 라이브러리 매니저를 실행 한 후 IRremote 으로 검색하여 라이브러리를 설치



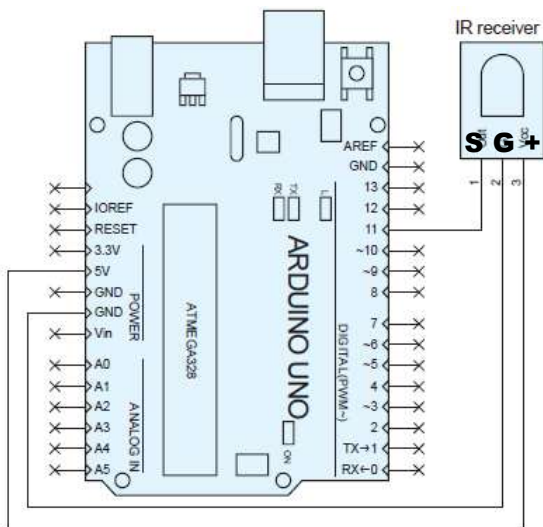
# 8.1.1 적외선 리모컨

## EX 8.1

## 적외선 리모컨 코드 읽기 (1/3)

- 실습목표**
1. 적외선 리모컨의 신호를 수신한다.
  2. 신호가 수신 될 때 내장된 13번 핀 LED를 점멸시킨다.
  3. 수신된 코드를 시리얼 통신으로 전송한다.
  4. 리모컨 버튼마다 어떤 코드가 수신되는지 확인한다.

- Hardware**
1. IR Receiver의 Vcc와 G를 Arduino의 5V와 GND에 연결한다.
  2. IR Receiver Out은 디지털입출력 11번 핀에 연결한다.



## 8.1.2 적외선 리모컨

### EX 8.1

### 적외선 리모컨 코드 읽기 (2/3)

#### Commands

- `IRrecv` 리모컨이름(리모컨 수신 핀번호)

리모컨 이름으로 리모컨 수신핀을 설정한다.

- `리모컨이름.enableIRIn()`

리모컨 수신시 타이머 인터럽트를 활성화한다.

- `decode_result` 실습결과

입력된 신호의 결과를 '실습결과' 변수에 입력한다.

- `리모컨이름.decode(&실습결과)`

리모컨 수신이 이루어졌을 때 그 값을 회신한다.

- `실습결과.decode_type`

수신된 리모컨 코드의 타입이 저장된다(NEC, SONY, RC5, RC6 등).

- `실습결과.value`

수신된 리모컨의 코드가 저장된다.

- `실습결과.bits`

수신된 리모컨 코드의 비트수가 저장된다.

- `실습결과.rawbuf`

수신된 적외선 펄스의 시간이 배열형태로 저장된다.

- `실습결과.rawlen`

수신된 적외선 펄스의 시간이 배열의 개수가 저장된다.

- `리모컨이름.resume()`

수신 후에 다른 신호를 수신하기 위하여 준비한다.

- `리모컨이름.blink13(true)`

Arduino에 내장된 LED(13번핀)을 리모컨 수신할 때 마다 점멸시킨다.

## 8.1.3 적외선 리모컨

### EX 8.1

### 적외선 리모컨 코드 읽기 (3/3)

- Sketch 구성**
1. 리모컨 라이브러리를 불러온다.
  2. 디지털입력 11번 핀을 리모컨 수신핀으로 설정한다.
  3. 'irrecv'라는 이름으로 리모컨이름을 설정한다.
  4. 리모컨신호 수신 결과를 'results'라는 이름으로 설정한다.
  5. 리모컨 수신이 발생했을 때 수신 값을 시리얼 통신으로 전송한다.

- 실행 결과**
1. 'Received Code is 수신된 코드' 의 메시지가 출력된다.
  2. 키를 누를 때 마다 수신된 코드가 변경된다.
  3. 모든 키에 대하여 수신된 코드를 표로 완성하자.

## 8.1.4 적외선 리모컨: code

ex\_8\_1

```
1 /*
2  예제 8.1
3  적외선 리모컨 코드 읽기
4 */
5
6 // 적외선 리모컨 라이브러리를 불러온다.
7 #include <IRremote.h>
8
9 // 적외선 수신부가 연결될 핀을 설정한다.
10 int irPin = 11;
11
12 // 적외선 수신부가 연결된 핀을 리모컨 수신 핀으로 설정한다.
13 IRrecv irrecv(irPin);
14
15 // 수신된 신호의 결과를 results 변수로 설정한다.
16 decode_results results;
17
```

```
18 void setup()
19 {
20   // 시리얼 통신을 설정한다.
21   Serial.begin(9600);
22
23   // 적외선 리모컨 수신을 시작한다.
24   irrecv.enableIRIn();
25
26   // 13번 핀에 연결된 LED를 리모컨 수신시 점멸시킨다.
27   irrecv.blink13(true);
28
29 }
30
31 void loop()
32 {
33   // 수신된 코드가 있을 때 실행한다.
34   if (irrecv.decode(&results)){
35     // 0xFFFFFFFF 값을 제외하고 출력한다.
36     if(results.value != 0xFFFFFFFF){
37       // 수신된 값을 16진수 형태로 출력한다.
38       Serial.print("Received Code is ");
39       Serial.println(results.value, HEX);
40     };
41     // 다음 수신을 위해서 준비한다.
42     irrecv.resume();
43   }
44 }
```

## 8.1.5 적외선 리모컨: DIY

DIY

1. 수신기 모듈에 수신된 코드를 직렬모니터에 표시해 보자.

응용 문제

2. 집에서 사용하는 TV의 리모컨의 코드를 확인해 보자.

수신된 코드가 출력된 직렬모니터 사진을

**ARnn\_remote\_Serial.png** 로 저장...

```
COM11 (Arduino/Genuino Uno)

Received Code is 12B07291
Received Code is 293C1067
Received Code is 4AB0F7B5
Received Code is FFA857
Received Code is FFA857
Received Code is FFA857
Received Code is FFA857
Received Code is FFA857
Received Code is FFA857
Received Code is FF38C7
Received Code is 488F3CBB
```



## 8.2

# 적외선 리모컨으로 LED 제어



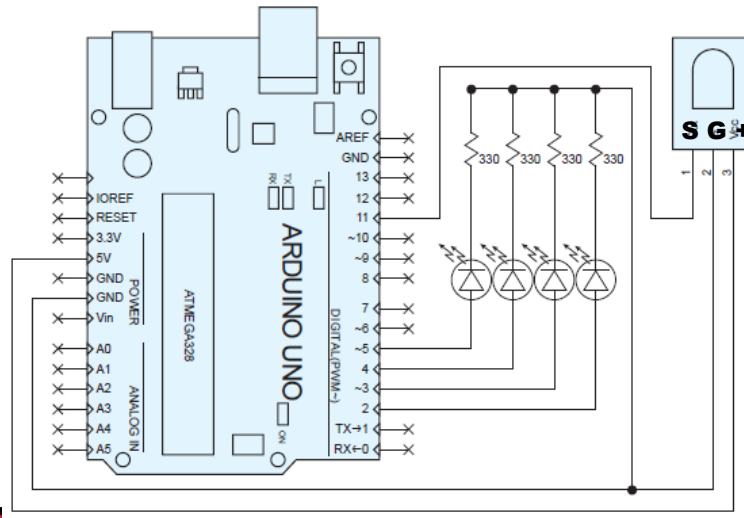
# 8.2.1 적외선 리모컨을 이용한 LED 제어

## EX 8.2

## 적외선 리모컨을 이용한 LED 제어 (1/2)

- 실습목표**
1. 적외선 리모컨의 신호를 수신한다.
  2. 수신된 코드에 따라 LED를 제어한다.
  3. 메시지를 시리얼 통신으로 전송한다.

- Hardware**
1. IR Receiver의 Vcc와 G를 Arduino의 5V와 GND에 연결한다.
  2. IR Receiver Out은 디지털입출력 11번 핀에 연결한다.
  3. Arduino의 디지털 입출력 2~5번핀에 4개의 LED의 Anode 핀을 연결한다.
  4. 각 LED의 Cathode에는 220 $\Omega$  저항을 연결하여 GND에 연결한다.



## 8.2.2 적외선 리모컨을 이용한 LED 제어

EX 8.2

### 적외선 리모컨을 이용한 LED 제어 (2/2)

- Sketch 구성**
1. 리모컨 라이브러리를 불러온다.
  2. 디지털입력 11번 핀을 리모컨 수신핀으로 설정한다.
  3. 'irrecv'라는 이름으로 리모컨이름을 설정한다.
  4. 리모컨신호 수신 결과를 'results'라는 이름으로 설정한다.
  5. 디지털 입출력핀 2, 3, 4, 5를 LED 출력으로 사용하기 위해 출력모드로 설정한다.
  6. 각 LED의 점등과 소등 신호를 미리 설정해 준다.
  7. 적외선 리모컨 신호가 수신되었을 때 신호에 맞춰 LED를 동작시킨다.

- 실습 결과**
1. 리모컨 키를 누를 때 마다 해당 LED가 점등/소등 한다.
  2. 현재 동작 상태에 대한 메시지가 시리얼 통신으로 전송된다.



## 8.2.3 적외선 리모컨을 이용한 LED 제어:code

```

1 /*
2  * 예제 8.2
3  * 적외선 리모컨을 이용한 LED 제어
4  */
5
6 // 적외선 리모컨 라이브러리를 불러온다.
7 #include <IRremote.h>
8
9 // 적외선 수신부가 연결될 핀을 설정한다.
10 int irPin = 11;
11
12 // LED에 연결된 핀을 설정한다.
13 int led1 = 2;
14 int led2 = 3;
15 int led3 = 4;
16 int led4 = 5;
17
18 // LED 제어용 코드 (리모컨에 맞게 수정한다.)
19 long on1 = 0xFFA25D;
20 long off1 = 0xFF629D;
21 long on2 = 0x6182021B;
22 long off2 = 0x8C22657B;
23 long on3 = 0x488F3CB8;
24 long off3 = 0x449E79F;
25 long on4 = 0x3206FDF7;
26 long off4 = 0x1BC0157B;
27
28 // 적외선 수신부가 연결된 핀을 리모컨 수신 핀으로 설정한다.
29 IRrecv irrecv(irPin);
30
31 // 수신된 신호의 결과를 results 변수로 설정한다.
32 decode_results results;
33

```

```

34 void setup()
35 {
36   // 시리얼 통신을 설정한다.
37   Serial.begin(9600);
38
39   // 적외선 리모컨 수신을 시작한다.
40   irrecv.enableIRIn();
41
42   // 13번 핀에 연결된 LED를 리모컨 수신시 점멸시킨다.
43   irrecv.blink13(true);
44
45   pinMode(led1, OUTPUT);
46   pinMode(led2, OUTPUT);
47   pinMode(led3, OUTPUT);
48   pinMode(led4, OUTPUT);
49 }
50
51 void loop()
52 {
53   // 수신된 코드가 있을 때 실행한다.
54   if (irrecv.decode(&results)){
55     // 0xFFFFFFFF 값을 제외하고 출력한다.
56     if(results.value != 0xFFFFFFFF){
57       // 수신된 코드가 on1과 같을 때
58       if(results.value == on1){
59         digitalWrite(led1, HIGH);
60         Serial.println("LED1 is ON");
61       }
62       // 수신된 코드가 off1과 같을 때
63       if(results.value == off1){
64         digitalWrite(led1, LOW);
65         Serial.println("LED1 is OFF");
66       }
67     }
68   }
69 }

```

## 8.2.4 적외선 리모컨을 이용한 LED 제어

DIY

1. LCD 모듈에 수신된 코드를 표시해 보자.

응용 문제

2. 집에서 사용하는 TV의 리모컨을 사용하여 LED를 제어해 보자.

```
COM11 (Arduino/Genuino Uno)

LED1 is ON
LED1 is OFF
LED1 is OFF
LED1 is ON
LED1 is OFF
LED1 is ON
LED1 is OFF
LED1 is ON
LED1 is OFF
LED1 is ON
LED1 is OFF
LED1 is ON
LED1 is OFF
LED1 is ON
LED1 is OFF
```

수신된 코드가 출력된 LCD와 지정된  
LED가 켜진 사진을  
**ARnn\_remote\_LCD.png** 로 저장...

그리고 아두이노 스케치 코드를  
**ARnn\_remote\_LCD.ino** 로 저장





# [Practice]

## ◆ [wk12]

- **Arduino : Motors & IRremote**
- **Complete your project**
- **Submit folder : ARnn\_Rpt11**

# wk12 : Practice-11 : ARnn\_Rpt11

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

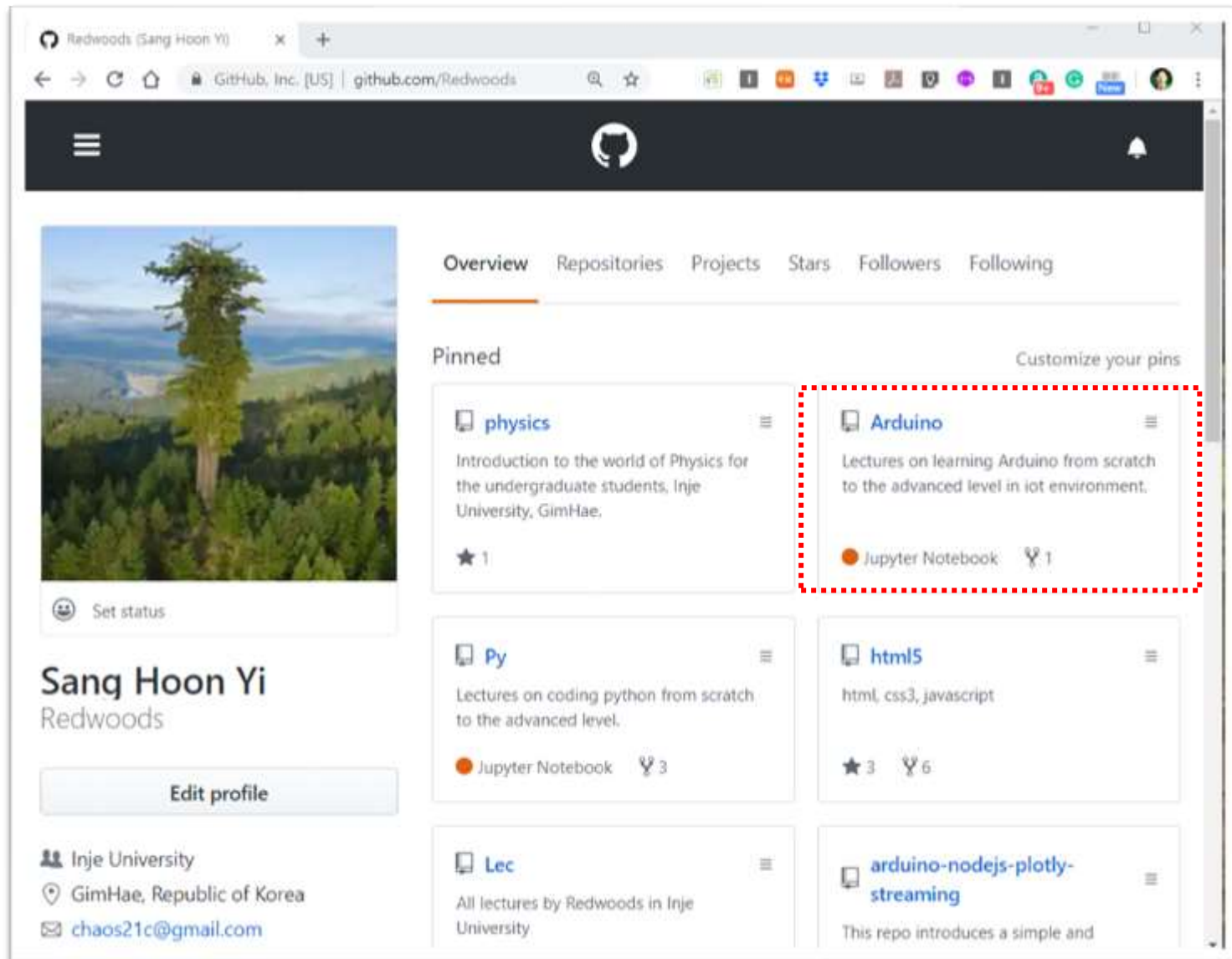
제출폴더명 : **ARnn\_Rpt11**

### 제출할 파일들

- ① **ARnn\_step\_motor.png**
- ② **ARnn\_step\_motor.ino**
- ③ **ARnn\_servo\_motor.ino**
- ④ **ARnn\_remote\_Serial.png**
- ⑤ **ARnn\_remote\_LCD.png**
- ⑥ **ARnn\_remote\_LCD.ino**
- ⑦ **\*.ino**

## ● References & good sites

- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



Redwoods (Sang Hoon Yi)

GitHub, Inc. [US] | github.com/Redwoods

Overview Repositories Projects Stars Followers Following

Pinned Customize your pins

**physics**  
Introduction to the world of Physics for the undergraduate students, Inje University, GimHae.  
★ 1

**Arduino**  
Lectures on learning Arduino from scratch to the advanced level in iot environment.  
Jupyter Notebook 🍴 1

**Py**  
Lectures on coding python from scratch to the advanced level.  
Jupyter Notebook 🍴 3

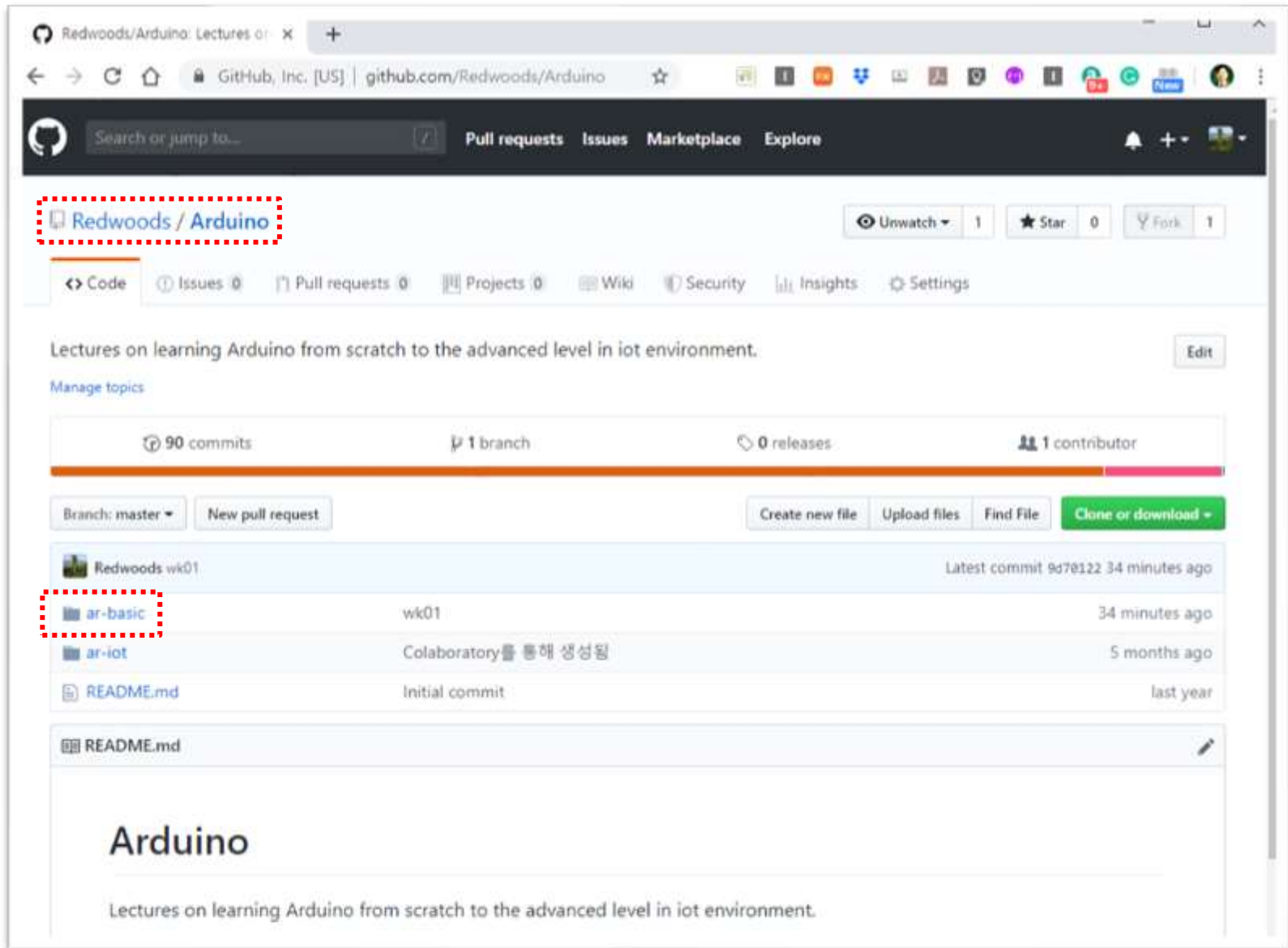
**html5**  
html, css3, javascript  
★ 3 🍴 6

**Lec**  
All lectures by Redwoods in Inje University.

**arduino-nodejs-plotly-streaming**  
This repo introduces a simple and

**Sang Hoon Yi**  
Redwoods  
Edit profile

**Inje University**  
GimHae, Republic of Korea  
chaos21c@gmail.com



The screenshot shows the GitHub repository page for **Redwoods/Arduino**. The repository description is "Lectures on learning Arduino from scratch to the advanced level in iot environment." The repository has 90 commits, 1 branch, 0 releases, and 1 contributor. The file list includes **ar-basic** (34 minutes ago), **ar-iot** (5 months ago), **README.md** (last year), and another **README.md** file. The **ar-basic** file is highlighted with a red dashed box. The repository name **Redwoods / Arduino** is also highlighted with a red dashed box. The page includes navigation links for Code, Issues, Pull requests, Projects, Wiki, Security, Insights, and Settings. The main content area shows the file **ar-basic** with its commit history and a description of the repository.

Redwoods/Arduino: Lectures on learning Arduino from scratch to the advanced level in iot environment.

Manage topics

90 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

Redwoods wk01 Latest commit 9d7e122 34 minutes ago

ar-basic wk01 34 minutes ago

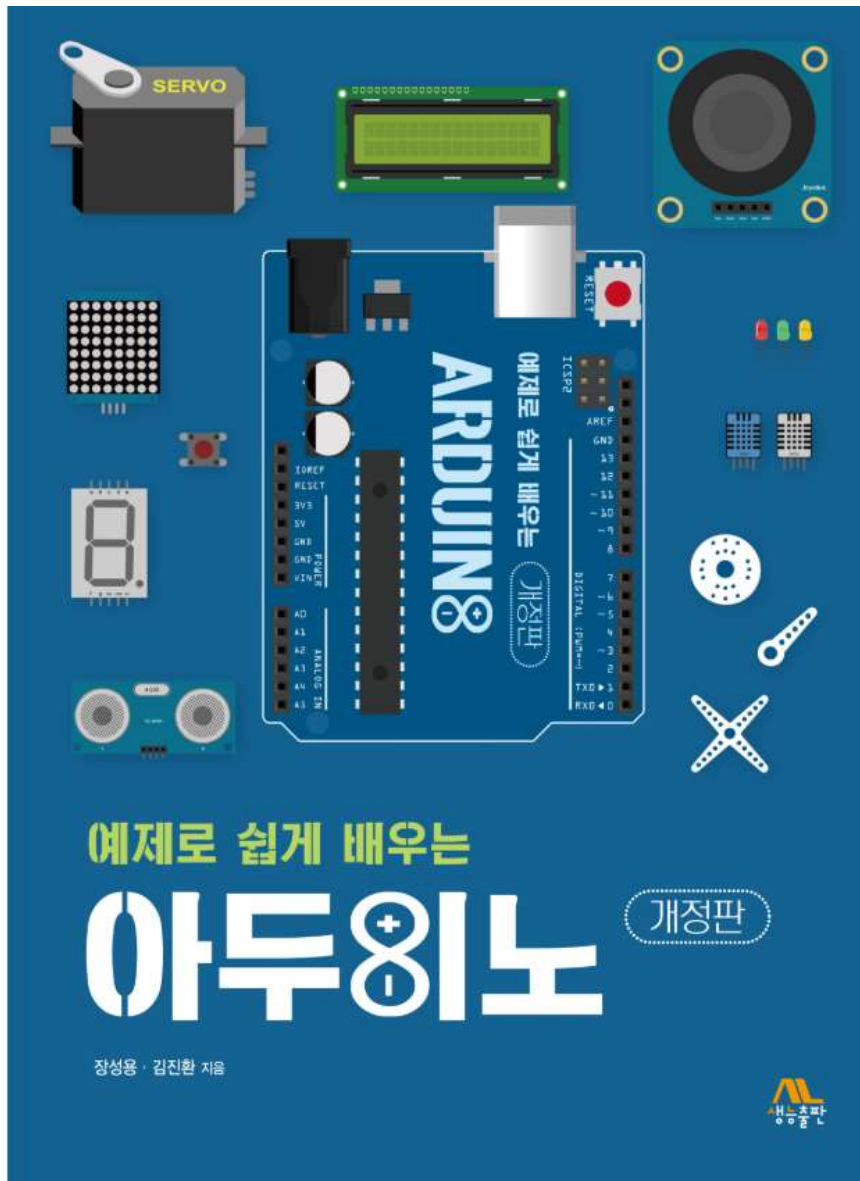
ar-iot Colaboratory를 통해 생성됨 5 months ago

README.md Initial commit last year

README.md

## Arduino

Lectures on learning Arduino from scratch to the advanced level in iot environment.

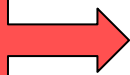




# 아두이노 키트(Kit)



web



<https://www.devicemart.co.kr/goods/view?no=12170416>

## 아두이노 레벨업키트(골드) 구성품

						
아두이노 UNO	USB 케이블	830핀 브레드보드	미니 브레드보드	점퍼와이어 세트	듀폰케이블 M/F	듀폰케이블 M/M
						
저항 220Ω	저항 1KΩ	저항 10KΩ	가변저항 10KΩ	빨강 LED	녹색 LED	파랑 LED
						
노랑 LED	RGB LED (CA)	RGB LED 모듈	1digit FND (CA)	4digit FND (CA)	8x8 도트 매트릭스	택트 스위치
						
택트 스위치 캡	벨 스위치	리드 스위치 센서	4x4키 매트릭스	5V 릴레이 모듈	조이스틱 모듈	수위 센서
						
온도센서 LM35	터미스터	습도센서	CDS 조도센서	불꽃감지센서	적외선 수신기	IR 리모컨

# 아두이노 키트(Kit) : Part-2

						
TCRT5000 적외선 센서	인체감지센서 모듈	사운드센서	능동부저	수동부저	초음파센서	I2C 1602 LCD 모듈
						
서보모터	스텝모터	스텝모터드라이버	RFID 수신 모듈	RFID 카드	RFID 태그	DS1302 RTC 모듈
						
1N4001 다이오드	2N2222 트랜지스터	74HC595	1x40 핀헤더	9V 배터리 스냅	아크릴 고정판	

■ 아두이노 UNO × 1	■ USB 케이블 × 1	■ 830핀브레드보드 × 1	■ 미니 브레드보드 × 1	■ 점퍼와이어세트 × 1
■ 듀폰케이블 × 80 (M/F,M/M)	■ 저항 × 30	■ 가변저항 × 1	■ LED × 20	■ RGB LED × 1
■ RGB LED 모듈 × 1	■ 1digit FND(CA) × 1	■ 4digit FND(CA) × 1	■ 8×8도트 매트릭스 × 1	■ 탭스위치 × 5
■ 탭스위치 캡 × 5	■ 볼스위치 × 1	■ 리드 스위치 센서 × 1	■ 4×4 키 매트릭스 × 1	■ 5V 릴레이 모듈 × 1
■ 조이스틱 모듈 × 1	■ 수위 센서 × 1	■ 온도센서 LM35 × 1	■ 써미스터 × 1	■ 온습도센서 × 1
■ CdS 조도센서 × 1	■ 불꽃감지센서 × 1	■ 적외선 수신기 × 1	■ IR 리모컨 × 1	■ TCRT5000 적외선 센서 × 1
■ 인체감지센서 모듈 × 1	■ 사운드센서 × 1	■ 능동부저 × 1	■ 수동부저 × 1	■ 초음파센서 × 1
■ I2C 1602 LCD 모듈 × 1	■ 서보모터 × 1	■ 스텝모터 × 1	■ 스텝모터드라이버 × 1	■ RFID 수신 모듈 × 1
■ RFID 카드 × 1	■ RFID 태그 × 1	■ DS1302 RTC 모듈 × 1	■ 1N4001 다이오드 × 1	■ 2N2222 트랜지스터 × 1
■ 74HC595 × 1	■ 1x40 핀헤더 × 1	■ 9V 배터리 스냅 × 1	■ 아크릴 고정판 × 1	