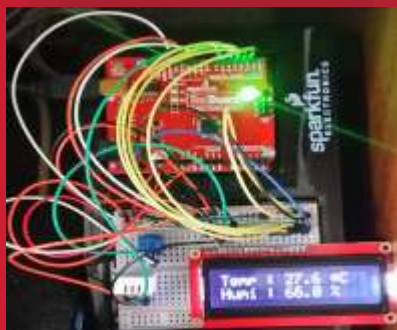




Arduino-basic

[wk12]

Motor



Learn how to code Arduino from scratch

Comsi, INJE University

2nd semester, 2018

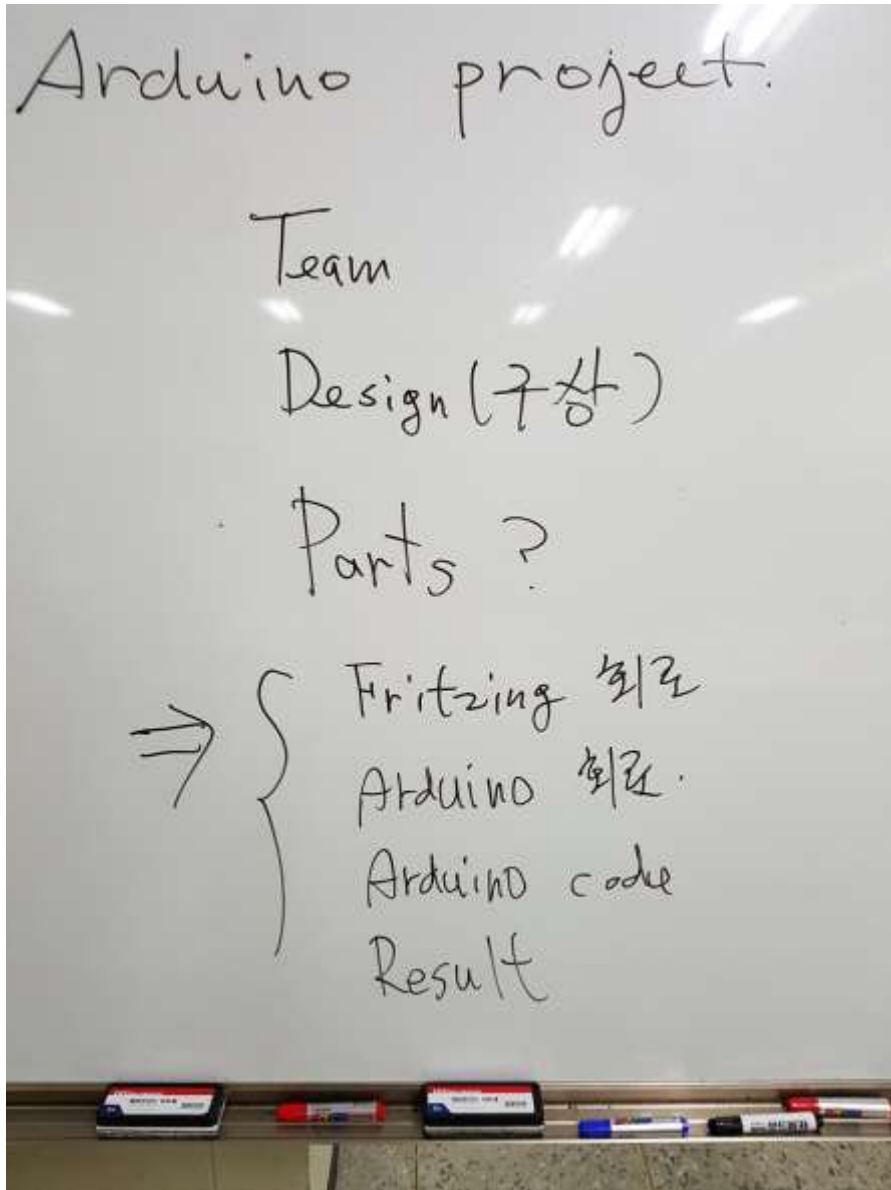
Email : chaos21c@gmail.com

My ID (ARnn)

성명	ID
백동진	AR01
김도훈	AR02
김희찬	AR03
류재현	AR04
문민규	AR05
박진석	AR06
이승현	AR07
이승협	AR08
이후정	AR09
최민구	AR10

김다영	AR11
공진영	AR12
김해인	AR13
류성현	AR14
류재환	AR15
박상현	AR16
박해주	AR17
백지혜	AR18
송원식	AR19
신송주	AR20
윤지훈	AR21
정은성	AR22

Arduino team project



- 2명/팀
- 구상 소개 (11.22, 11.29), ppt 준비
- 부품은 수업 세트 기준 (추가신청은 22일까지)
- 팀당 발표 자료 준비
- 발표 : 12월6일
- 참고



[Review]

◆ [wk11]

- **Arduino : Analog input II**
- **Complete your project**
- **Submit file : ARnn_Rpt08.zip**

wk11 : Practice-08 : ARnn_Rpt08.zip

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : **ARnn_Rpt08.zip**

- 압축할 파일들

① **ARnn_waterlevel.ino**

② **ARnn_joystick.ino**

③ **ARnn_joystick.png**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]

wk11 : Practice-08 : ARnn_Rpt08.zip

```

14 // 0번 아날로그핀을 X 축 입력으로 설정
15 const int xAxisPin = 0;
16 // 1번 아날로그핀을 Y 축 입력으로 설정
17 const int yAxisPin = 1;
18 // 2번 디지털 입력 핀을 Z 축 입력으로 설정
19 const int zAxisPin = 2;
20
21 // LED
22 const int wShoot = 3; // white Shoot
23 const int rStop = 5; // red Stop
24 const int bGo = 6; // blue Go
25 const int yRight = 9; // yellow Right
26 const int yLeft = 10; // yellow Left
27
28 void setup() {
29
30     // Z 축 입력은 디지털 입력으로 설정한다.
31     pinMode(zAxisPin, INPUT_PULLUP);
32     // LED
33     pinMode(wShoot, OUTPUT);
34     pinMode(rStop, OUTPUT);
35     pinMode(bGo, OUTPUT);
36     pinMode(yRight, OUTPUT);
37     pinMode(yLeft, OUTPUT);

```

```

// LED control
if(xDisplay > 10){
    digitalWrite(yRight, HIGH);
    delay(300);
    digitalWrite(yRight, LOW);
}else if(xDisplay < 10){
    digitalWrite(yLeft, HIGH);
    delay(300);
    digitalWrite(yLeft, LOW);
}

if(yDisplay > 11){
    digitalWrite(rStop, HIGH);
    delay(300);
    digitalWrite(rStop, LOW);
}else if(yDisplay < 10){
    digitalWrite(bGo, HIGH);
    delay(300);
    digitalWrite(bGo, LOW);
}

```

wk11 : Practice-08 : ARnn_Rpt08.zip



```
if(xValue>1000)
digitalWrite(rStop, HIGH);
else if (xValue < 10)
    digitalWrite(wShoot, HIGH);
else
{
digitalWrite(rStop, LOW);
    digitalWrite(wShoot, LOW);
}
if(yValue>1000)
digitalWrite(yRight, HIGH);
else if (yValue<10)
digitalWrite(yLeft, HIGH);

else
{
    digitalWrite(yRight, LOW);
    digitalWrite(yLeft, LOW);
}
```

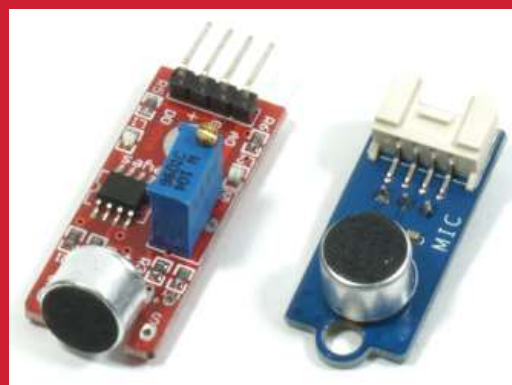
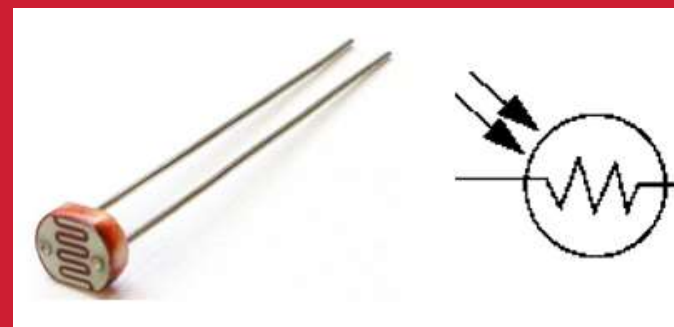
```
if(xValue>550){
    digitalWrite(bGo,HIGH);
}else if(xValue<500){
    digitalWrite(rStop,HIGH);
}else if(yValue>550){
    digitalWrite(yRight,HIGH)
}else if(yValue<500){
    digitalWrite(yLeft,HIGH);
}
else{
    digitalWrite(bGo,LOW);
    digitalWrite(rStop,LOW);
    digitalWrite(yRight,LOW);
    digitalWrite(yLeft,LOW);
}
```

```
if(yValue<500)
{
    digitalWrite(r,HIGH);
}
else
{
    digitalWrite(r,LOW);
}
if(yValue>600)
{
    digitalWrite(b,HIGH);
}
else
{
    digitalWrite(b,LOW);
}
if(xValue>600)
{
    digitalWrite(g,HIGH);
}
else
{
    digitalWrite(g,LOW);
}
if(xValue<500)
{
    digitalWrite(y,HIGH);
}
else
{
    digitalWrite(y,LOW);
}
```

```
if (yValue < 5)
{
    digitalWrite(bGo,HIGH);
}
else
{
    digitalWrite(bGo,LOW);
}
if (xValue > 1021 && yValue < 520)
{
    digitalWrite(yRight,HIGH);
}
else
{
    digitalWrite(yRight,LOW);
}
if (yValue > 1020)
{
    digitalWrite(rStop,HIGH);
}
else
{
    digitalWrite(rStop,LOW);
}
if (xValue < 8 && yValue < 520)
{
    digitalWrite(yLeft,HIGH);
}
else
{
    digitalWrite(yLeft,LOW);
}
```



6. Analog input



조이스틱 모듈

6. 아날로그 신호 입력

- 6.1 포텐쇼미터 입력 (가변저항기)
- 6.2 빛 입력 (CdS, LDR)
- 6.3 온도 측정 (LM35, TMP36)
- 6.4 수위 측정
- 6.5 아날로그 조이스틱
- 6.6 소리 입력



6.5

아날로그 조이스틱



6.5.4 아날로그 조이스틱 - DIY (참고)

DIY

1. 5개의 LED를 브레드보드에 ‘+’ 모양으로 배치시킨다.

응용 문제

2. 조이스틱의 방향에 따라 해당하는 LED를 점등시키는 스케치를 작성해 보자.



6.5.4 아날로그 조이스틱 - DIY

DIY

1. 5개의 LED를 브레드보드에 ‘+’ 모양으로 배치시킨다.

응용 문제

2. 조이스틱의 방향에 따라 해당하는 LED를 점등시키는 스케치를 작성해 보자.

→ **ARnn_joystick.ino** 로 저장하고 제출

```
21 // LED
22 const int wShoot = 3; // white Shoot
23 const int rStop = 5; // red Stop
24 const int bGo = 6; // blue Go
25 const int yRight = 9; // yellow Right
26 const int yLeft = 10; // yellow Left
```

```
32 // LED
33 pinMode(wShoot, OUTPUT);
34 pinMode(rStop, OUTPUT);
35 pinMode(bGo, OUTPUT);
36 pinMode(yRight, OUTPUT);
37 pinMode(yLeft, OUTPUT);
```

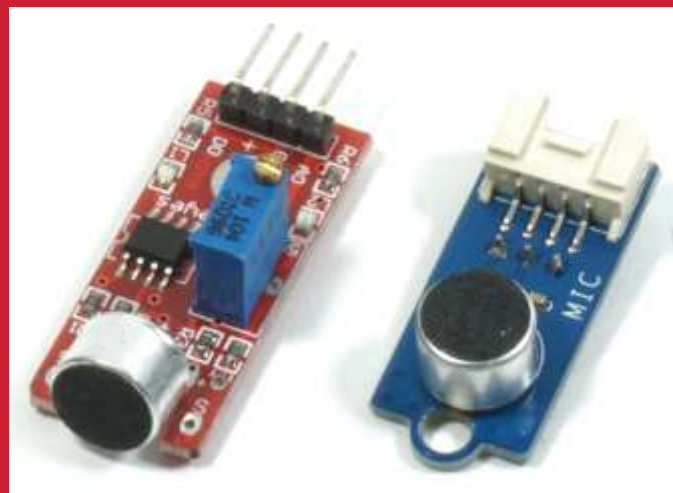
동작 중 사진을
ARnn_joystick.png 로 저장...





6.6

마이크로폰 모듈



마이크론 모듈

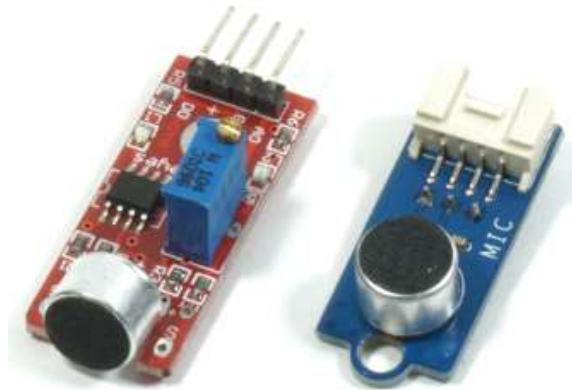


그림 6.7 마이크론 모듈

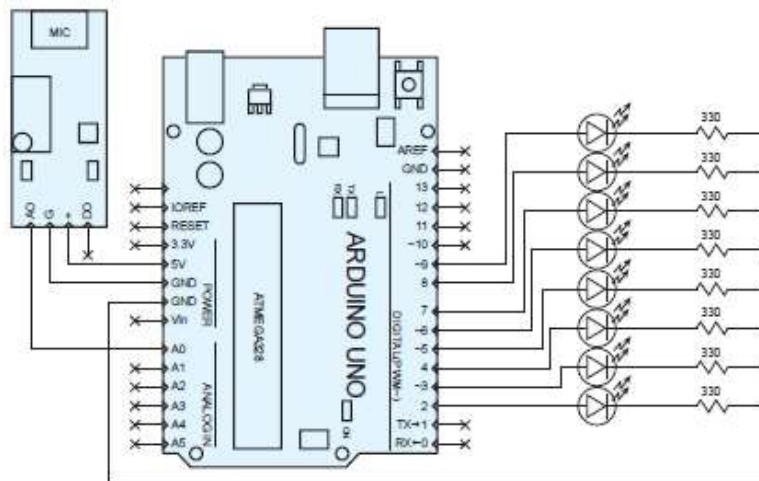
- ✓ 입력되는 소리 신호와 비례하여 아날로그 신호 출력
- ✓ 디지털 출력으로 사용할 때는 내장된 포텐쇼미터로 임계값 조절

6.6.1 마이크론 모듈

EX 6.6 소리 입력 (1/3)

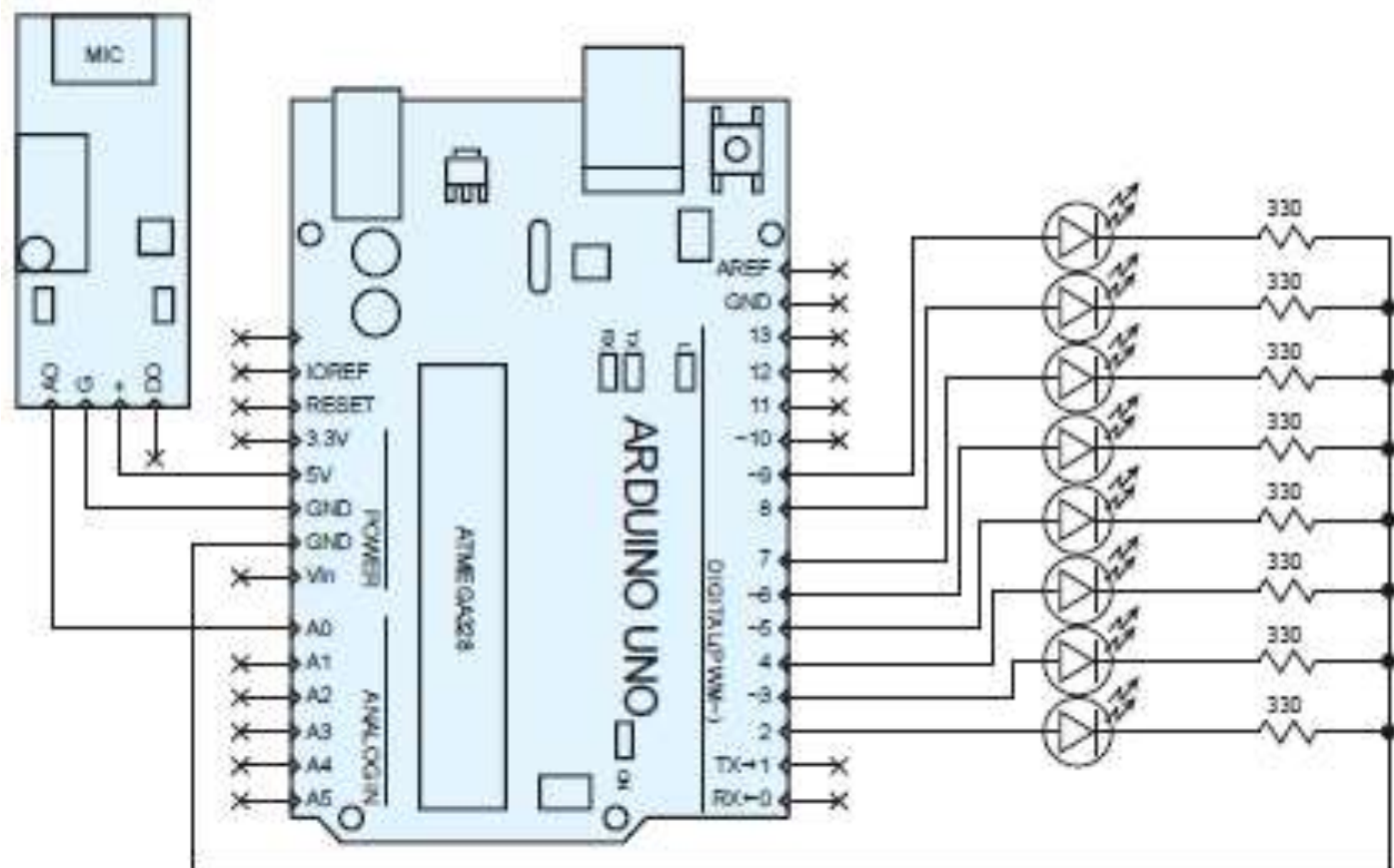
- 실습목표**
1. 마이크론 모듈을 이용하여 소리를 아날로그 신호로 입력 받는다.
 2. 소리의 크기에 따라 8개의 LED로 그래프 바를 만들어 보자.

- Hardware**
1. LED 바를 만들기 위해 2~9번핀에 8개의 LED를 연결한다. Anode를 Arduino의 핀에 연결하고 Cathode에 330 Ω 저항을 연결하여 GND에 연결한다.
 2. 마이크론 모듈의 +와 G를 Arduino의 5V와 GND에 연결한다.
 3. 마이크론 모듈의 AO핀을 Arduino의 아날로그 입력핀 A0에 연결한다.
 4. 입력되는 소리의 크기를 ADC로 읽어 LED바를 동작하는데 참고한다.



6.6.1 마이크론 모듈

EX 6.6 소리 입력 (1/3)



EX 6.6

소리 입력 (2/3)

Commands

- `analogRead(아날로그 핀번호)`

아날로그핀에서 아날로그 값을 읽는다. 0~5V사이의 전압을 0~1023사이의 값으로 표현한다.

- `map(변수명, 범위1 최소값, 범위1 최대값, 범위2 최소값, 범위2 최대값)`

변수명의 변수의 범위1의 범위와 범위2의 범위에 매칭시킨다. 즉 변수가 0~100의 범위를 갖고 이를 50~200의 범위로 매칭하려면 'map(변수명, 0, 100, 50, 200)'의 명령어로 매칭시킬 수 있다.

- `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.

Sketch 구성

1. MIC 모듈에서 출력되는 아날로그 신호를 아날로그 입력핀 0번에서 입력받는다.
2. 디지털 출력 2~9번핀에 LED를 입력된 아날로그 값과 대응하여 단계별로 출력한다.
3. 아날로그 신호가 클수록 많은 수의 LED를 켜다.

6.6.2 마이크로폰 모듈 – code

ex_6_6

```

1  /*
2  예제 6.6
3  소리 입력
4  */
5
6  // A0번 핀에서 사운드 입력을 받는다
7  char soundInputPin = 0;
8
9  // 그래프 바 LED 출력핀을 level 변수에 저장
10 char ledLevel[8] = {2, 3, 4, 5, 6, 7, 8, 9};
11
12 void setup() {
13   // 그래프 바 LED 핀을 출력으로 설정
14   for(int i=0 ; i<=7 ; i++){
15     pinMode(ledLevel[i], OUTPUT);
16   }
17 }

```

```

19 void loop(){
20
21   // A0번 핀에서 사운드 입력을 받는다
22   int soundInput = analogRead(soundInputPin);
23   // 노이즈부분을 제외한 50~900의 범위로 입력받은 사운드 크기를
24   // 0~7단계로 변경한다.
25   int soundLevel = map(soundInput, 50, 900, 0, 7);
26
27
28   // 전체 LED를 소등한다.
29   for(int i = 0 ; i <= 7 ; i++){
30     digitalWrite(ledLevel[i], LOW);
31   }
32
33   // 0~7 단계 중 입력보다 작은 레벨의 LED는 점등한다.
34   for(int i = 0 ; i <= soundLevel ; i++){
35     digitalWrite(ledLevel[i], HIGH);
36   }

```

6.6.3 마이크론 모듈

EX 6.6 소리 입력 (3/3)

- Sketch 구성
1. MIC 모듈에서 출력되는 아날로그 신호를 아날로그 입력핀 0번에서 입력받는다.
 2. 디지털 출력 2~9번핀에 LED를 입력된 아날로그 값과 대응하여 단계별로 출력한다.
 3. 아날로그 신호가 클수록 많은 수의 LED를 켜다.

실행 결과 소리의 크기에 따라서 LED 바가 점등된다.

사진을 **ARnn sound bar.png**
로 저장...

DIY

응용 문제

시리얼 통신을 통하여 소리의 크기를 PC 모니터에 출력해 보자.

```
COM11 (Arduino/Genuino Uno)
Sound input: 34, Sound level: 0
Sound input: 267, Sound level: 1
Sound input: 1021, Sound level: 7
Sound input: 1021, Sound level: 7
```

직렬모니터 출력화면을
ARnn_sound monitor.png
로 저장...



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(2상 여자방식)

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

- ✓ 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

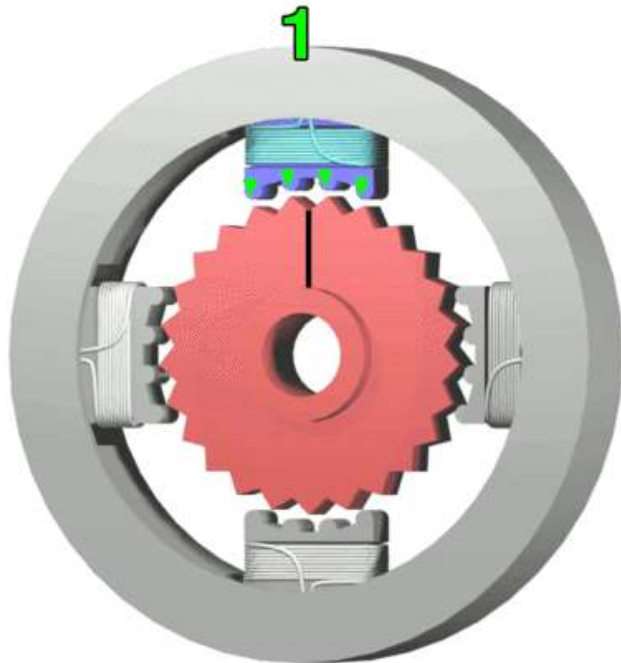


그림 7. 1 실험에 사용할 스텝모터 28BYJ-48와 사양표



그림 7. 2 실험에 사용할 스텝모터 드라이브 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, 라즈베리파이등으로 동작시키기에 편리함.
- 제어부가 비교적 간단해서 가격이 저렴하고 브러시리스(Brushless)모터이기 때문에 오래 사용할 수 있다.

https://en.wikipedia.org/wiki/Stepper_motor

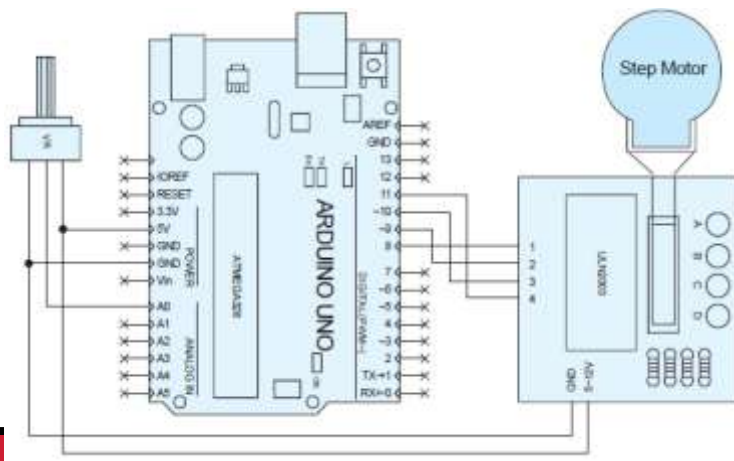
<https://m.blog.naver.com/3demp/220976664782>

EX 7.1

스텝모터 구동 (1/3)

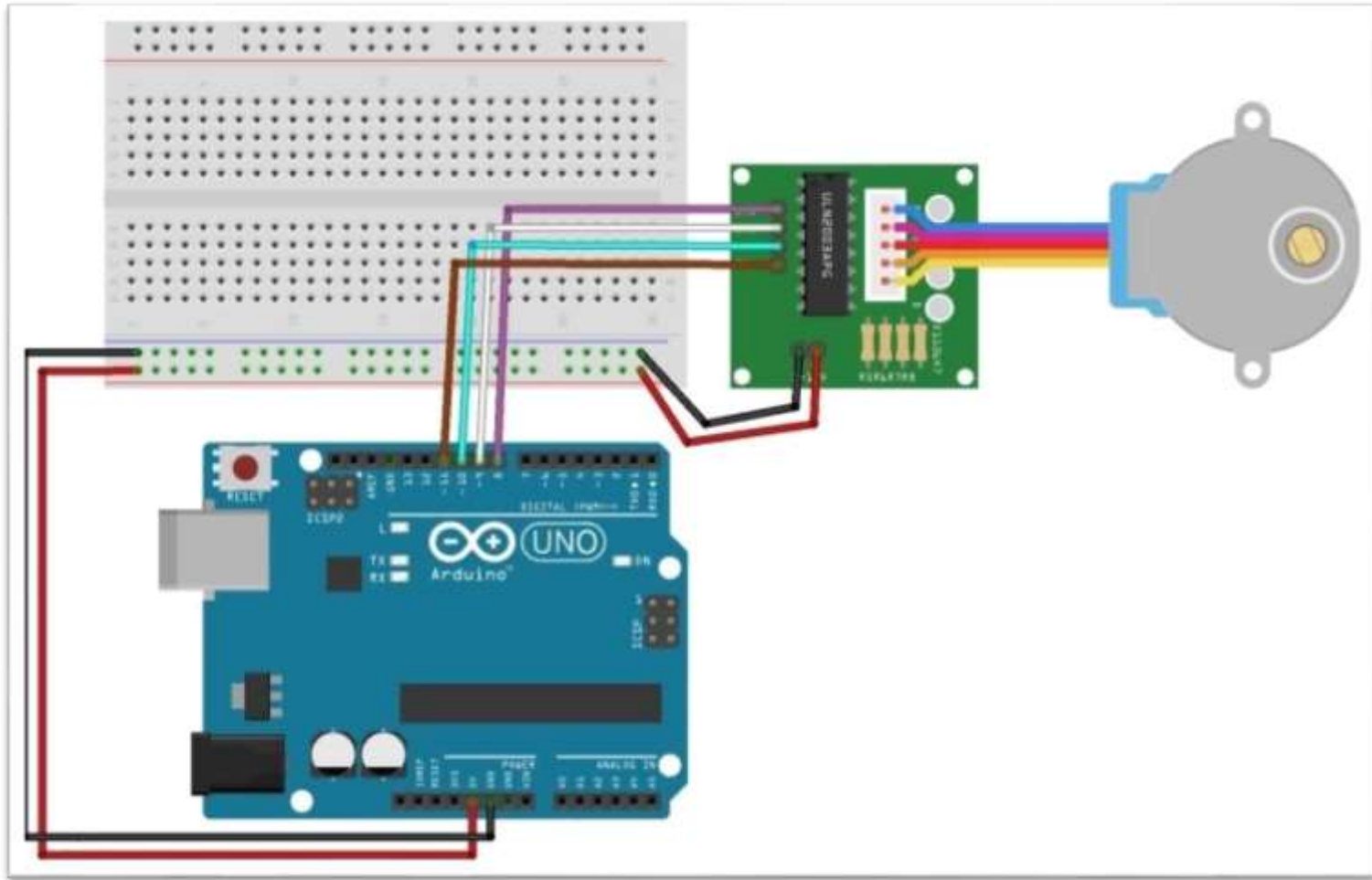
- 실습목표**
1. 포텐쇼미터를 이용하여 아날로그 입력을 받는다.
 2. 포텐쇼미터가 중간 위치이면 모터 Stop, CW로 회전시키면 CW로 회전, CCW로 회전시키면 CCW로 회전시킨다
 3. 포텐쇼미터의 각도에 따라서 모터의 속도가 조절되며 시리얼 모니터로 확인한다.

- 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)



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

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 (μs) 단위로 4500~1000의 범위를 갖는다. 각 스텝 사이에 이 값을 넣어 지연시켜 펄스를 만들어 준다.
 4. 포텐쇼미터값이 중간에 왔을 때는 모터를 정지시키고 포텐쇼미터의 방향과 회전각에 따라 모터를 회전시킨다.
 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 4500~1000의 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};
```


7.1.4 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 }

```

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)

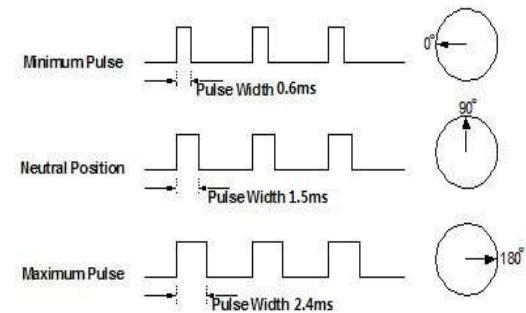
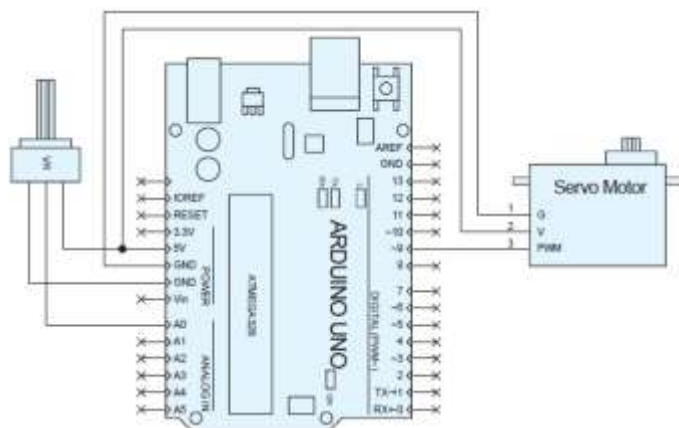


그림 7. 5 PWM 신호와 RC 서보모터의 회전각

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 (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.

7.2.3 서보모터

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.

- 실행 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 }

```


7.2.5 서보모터 - DIY

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...



[Practice]

◆ [wk12]

- **Arduino : Motors**
- **Complete your project**
- **Submit file : ARnn_Rpt09.zip**

wk12 : Practice-09 : ARnn_Rpt09.zip

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : **ARnn_Rpt09.zip**

- 압축할 파일들

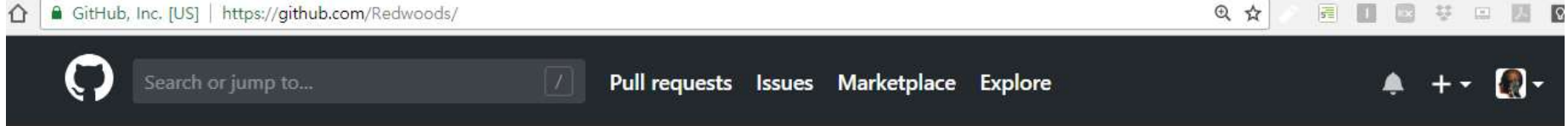
- ① **ARnn_sound_bar.png**
- ② **ARnn_sound_monitor.png**
- ③ **ARnn_step_motor.png**
- ④ **ARnn_step_motor.ino**
- ⑤ **ARnn_servo_motor.ino**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]

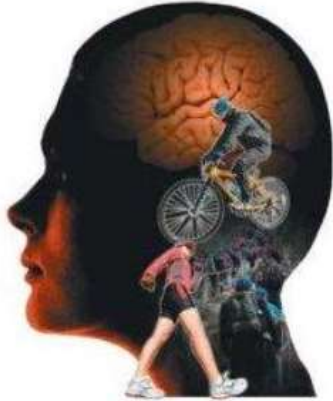
● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling
- ✓ <https://www.youtube.com> Youtube



Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



Redwoods Yi

Redwoods

Add a bio

 GimHae, Republic of Korea

 chaos21c@gmail.com

Overview

Repositories 7

Stars 2

Followers 1

Following 0

Pinned repositories

[Customize your pinned repositories](#)

≡ [Py](#)

Lectures on coding python from scratch to the advanced level.

 Jupyter Notebook

≡ [Arduino](#)

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

≡ [Lec](#)

All lectures by Redwoods in Inje University

 Jupyter Notebook

≡ [hw-coding](#)

Resource for lecture of Hardware Programming (2017, Inje university)


Arduino

171 contributions in the last year

[Contribution settings](#) ▼

Redwoods/Arduino: Lect

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






Search or jump to...

Pull requests

Issues

Marketplace

Explore



Redwoods / Arduino

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

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

Edit

Add topics

2 commits

1 branch

0 releases

1 contributor

Branch: master

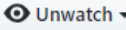
New pull request

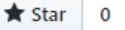
Create new file


Upload files


Find file

Clone or download


 Redwoods 2018 start Latest commit 38ca9e0 28 minutes ago

 ar-basic 2018 start 28 minutes ago

 ar-iot 2018 start 28 minutes ago

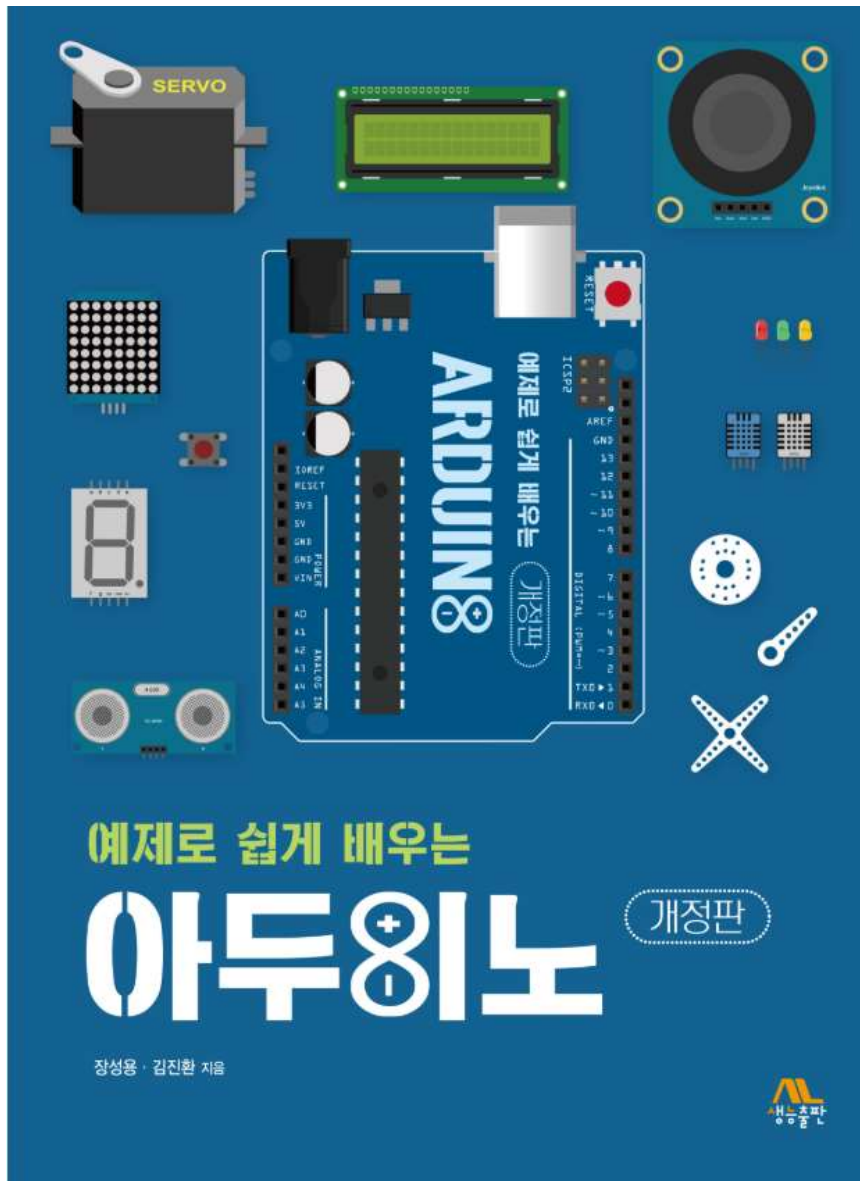
 README.md Initial commit 43 minutes ago

README.md



Arduino

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





http://arduinostory.com/goods/goods_view.php?goodsNo=1000000306

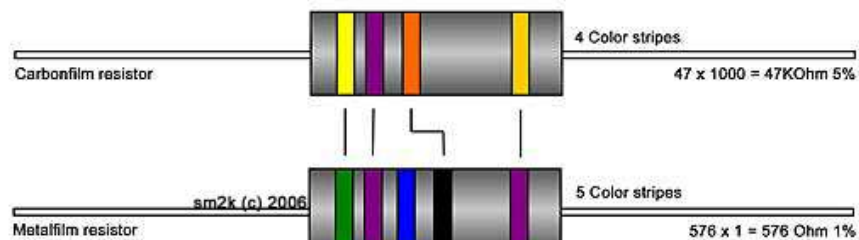
상급키트 구성품

1 1EA 아두이노 우노 R3 DIP 아두이노 우노 R3 (DIP) 호환보드 기본 메인보드입니다.	2 1EA 9V 배터리 홀더 9V 배터리를 연결하여 아두이노에 외부전원을 공급할 수 있습니다.	3 1EA 7세그먼트 4채널 7세그먼트가 4개 연결된 형태의 부품입니다. 총 12개의 핀을 사용합니다.	4 1EA 7세그먼트 1채널 공통 음극 7세그먼트 시계나 점수 등의 숫자를 표현 할 때 많이 사용됩니다.	13 1EA 수동부저 아두이노의 tone함수를 통해 소리를 내는 부저입니다.	14 6EA 택트스위치 (12x12x7) 스위치를 누르고 있을 경우만 ON됩니다.	15 각3EA 택트스위치 캡 (피랑,노랑,초록,빨강,하양) 택트스위치를 사용할 때 스위치간의 구분을 할 수 있습니다.	16 3EA 조도센서 빛을 감지하거나 빛의 밝기를 아날로그로 출력해주는 CDS 센서입니다.
5 1EA 74HC595N 기본 메인보드입니다. 74HC595N LED, 드로메트릭스, NFD 제어 IC 입니다.	6 1EA 65핀 점퍼 와이어 브레드보드에 연결할 때 사용하는 65핀 점퍼와이어 입니다.	7 1EA 무지개 점퍼선 F-M 20cm M타입과 F타입이 양쪽으로 달린 무지개 점퍼선입니다.	8 1EA 투명 부품 케이스 대,소 키트 구성품을 담을 수 있는 투명 부품 케이스입니다.	17 각 5EA LED 5mm (빨강,노랑,초록,하양,파랑) 기본으로 사용되는 LED입니다. 동작전압 : 2.2~2.4V 사용전류 : 20mA 미만	18 1EA 헤더핀 1x40/2.54mm 핀 간격은 2.54mm이며 헤더핀의 길이는 약 1.15cm입니다.	19 1EA USB케이블 50cm PC와 아두이노 우노 보드를 연결하여 프로그램을 다운로드 할 때 사용합니다.	20 1EA 저항값 카드 저항값을 쉽게 확인 할 수 있는 카드입니다. 사이즈 : 60mm x 50mm
9 1EA 가변저항10K 물리변 저항값이 가능합니다. (0~10KΩ)	10 1EA 1602 I2C LCD 아두이노 16x2 I2C LCD 모듈입니다. LCD입니다.	11 각 10EA 저항 100, 220, 330, 1K, 2K, 4.7K, 10K, 47K, 100K	12 1EA 브레드 보드 830울 브레드 보드 830울(봉투형) 센서 테스트나, 회로 프로토타입을 작성할 때 사용합니다.	21 1EA 능동부저 Signal 단자가 HIGH 일 때 약 2.5kHz의 음이 발생됩니다.	22 1EA 5V 1채널 릴레이 모듈 아두이노의 디지털 핀과 모듈 하단의 IN 핀들을 연결해 릴레이를 제어할 수 있는 모듈입니다.	23 1EA 8x8 도트 매트릭스 모듈 LED로 다양한 연출을 할 수 있습니다.	24 1EA 4x4 16 키패드 모듈 16개의 버튼을 사용할 수 있습니다.

아두이노 키트(Kit) : Part-2

<p>25 1EA</p> <p>무선 리모콘 키트</p> <p>핵파선을 사용해서 리모콘 기능을 구현할 수 있습니다.</p>	<p>26 2EA</p> <p>가열기 센서 스위치</p> <p>센서의 가열기에 따라 스위치 역할을 합니다.</p>	<p>27 1EA</p> <p>or</p> <p>사운드 센서 모듈</p> <p>아두이노와 호환되는 사운드센서 모듈입니다.</p>	<p>28 1EA</p> <p>불꽃 센서</p> <p>근거리 화재, 불꽃을 감지하는 센서입니다.</p>	<p>37 1EA</p> <p>DC 5V 스텝 모터</p> <p>28BYJ-48 스텝 모터 중 저렴한 편에 속하는 모델입니다. 5개의 핀을 사용합니다.</p>	<p>38 1EA</p> <p>DS1302 RTC 모듈</p> <p>아두이노 등 마이크로컨트롤러에 사용이 가능합니다.</p>	<p>39 1EA</p> <p>아두이노 우노 프로토 쉴드</p> <p>UNO 보드에서 회로를 간단히 짜기 위해 보드 위에 얹어 사용하는 쉴드입니다.</p>	<p>40 1EA</p> <p>3축 가속도 센서 모듈</p> <p>가속도를 측정할 수 있는 센서입니다.</p>
<p>29 1EA</p> <p>모터 드라이버 모듈</p> <p>ULN2003 스텝 모터 드라이버 모듈 5V ~ 12V를 사용할 수 있습니다.</p>	<p>30 1EA</p> <p>LM35 온도 센서</p> <p>온도를 마닐로그 값으로 출력합니다.</p>	<p>31 1EA</p> <p>수위 센서 모듈</p> <p>센서 역할에 잠긴 정도를 마닐로그 값으로 출력합니다.</p>	<p>32 1EA</p> <p>SG90 서보모터</p> <p>Vcc, GND, 신호선, 총 3개의 핀이 있습니다. 로봇팔이나 자동차, 비행기 조종에 사용됩니다.</p>	<p>41 1EA</p> <p>5V DC모터</p> <p>5V DC모터</p>	<p>42 1EA</p> <p>인체 감지 센서 모듈</p> <p>핵파선을 이용해 움직임을 감지하는 센서입니다. 오선이 감지되면 HIGH 신호를 출력합니다.</p>	<p>43 5EA</p> <p>다이오드 1N4001</p> <p>다이오드 1N4001</p>	<p>44 5EA</p> <p>세라믹 캐패시터 (22pF)</p> <p>세라믹 캐패시터 (22pF)</p>
<p>33 1EA</p> <p>초음파 거리 센서 모듈</p> <p>5V를 사용하여 만직 거리는 2cm에서 500cm입니다.</p>	<p>34 1EA</p> <p>조이스틱 모듈</p> <p>기본적으로 조이스틱 모듈은 두개의 가변저항이 서로 수직으로 회전하는 형태로 되어 있습니다.</p>	<p>35 1EA</p> <p>온습도 센서 모듈</p> <p>아두이노 온습도 센서중 가장 대중적으로 사용되는 DHT11 디지털 센서입니다.</p>	<p>36 1EA</p> <p>RGB LED 모듈</p> <p>RGB LED 모듈로 RGB LED 세개를 하나로 묶은 상품입니다.</p>	<p>45 5EA</p> <p>세라믹 캐패시터 (1uF)</p> <p>세라믹 캐패시터 (1uF)</p>	<p>46 5EA</p> <p>트랜지스터 2N2222</p> <p>트랜지스터 2N2222</p>	<p>47 5EA</p> <p>트랜지스터 BC547</p> <p>트랜지스터 BC547</p>	<p>48 5EA</p> <p>트랜지스터 BC557</p> <p>트랜지스터 BC557</p>
<p>49 2EA</p> <p>전해 캐패시터 (50V 10uF)</p> <p>전해 캐패시터 (50V 10uF)</p>	<p>50 2EA</p> <p>전해 캐패시터 (50V 100uF)</p> <p>전해 캐패시터 (50V 100uF)</p>						

[참고 : 저항 값 읽기]



Color	First	Second	Third	Multiplier	Tolerance
Black	0	0	0	x1	
Brown	1	1	1	x10	1%
Red	2	2	2	x100	2%
Orange	3	3	3	x1000	
Yellow	4	4	4	x10 000	
Green	5	5	5	x100 000	0,50%
Blue	6	6	6	x1 000 000	0,25%
Violette	7	7	7	x10 000 000	0,10%
Gray	8	8	8		
White	9	9	9		
Silver				x0,01	10%
Gold				x0,1	5%

