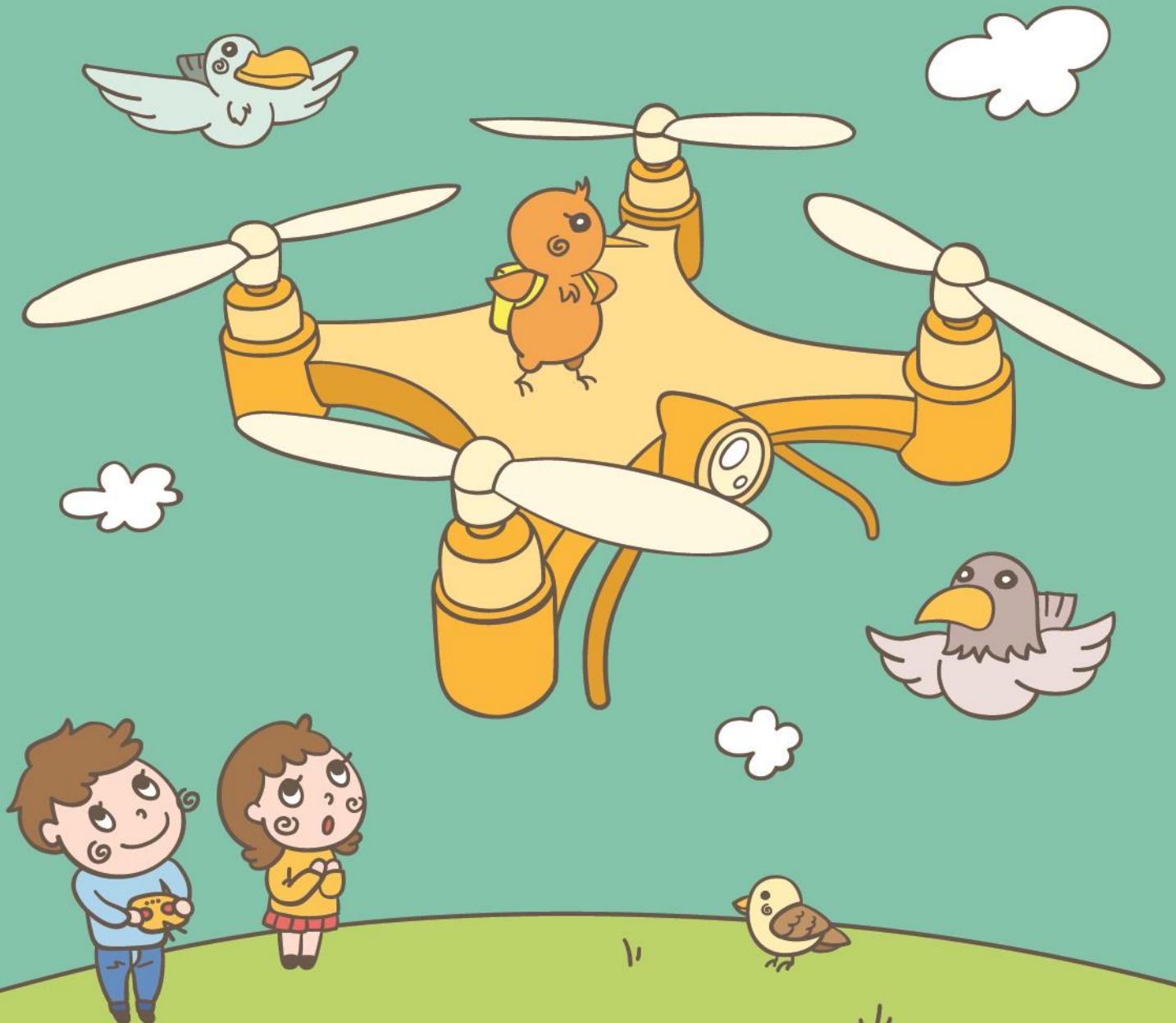




드론으로 배우는
프로그래밍 교실

캠프2h. 모터



01 가변저항에 대해서



가변저항은 간단하게 저항 값을 바꿀 수 있어서 여러가지 용도로 사용되어집니다.

라디오 볼륨조절이나 티비 볼륨 조절 등에도 쓰이며, 저항 값 테스트 용도로도 많이 쓰입니다.

이러한 가변저항을 다루는 방법에 대해 알아보시다.

가변저항이란?

가변저항

가변저항은 값이 변하는 저항입니다. 저항은 전압 또는 전류값을 조절해 줄 때 사용합니다.

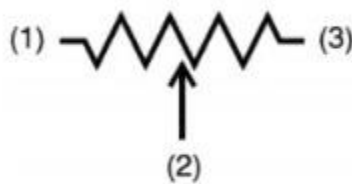
스위치의 경우 단순히 전기가 흐른다 / 흐르지 않느냐를 제어할 수 있었는데, 가변 저항에서는 전기가 흐르는 양을 제어할 수 있습니다.

이렇게 전기의 흐름(전류)가 바뀌면 그에 따라 전압이 달라지게 됩니다.



<그림1-1> 가변 저항

가변저항의 회로도는 다음과 같이 저항 회로도에 화살표가 있는 형태입니다. (2)번의 위치에 따라 저항값이 변하게 됩니다.



<그림1-2> 가변 저항 회로도

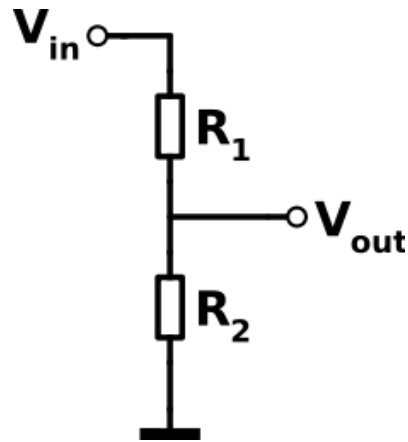
가변저항으로 저항값이 바뀔 때에는 전압 분배의 원리가 적용됩니다.

저항값이 바뀔 때 따라 중간 지점에서 확인되는 전압이 달라지게 됩니다.

달라지는 전압값을 아두이노에 확인할 수 있습니다.

전압 분배란?

전압 분배를 회로로 나타내면 다음과 같습니다.

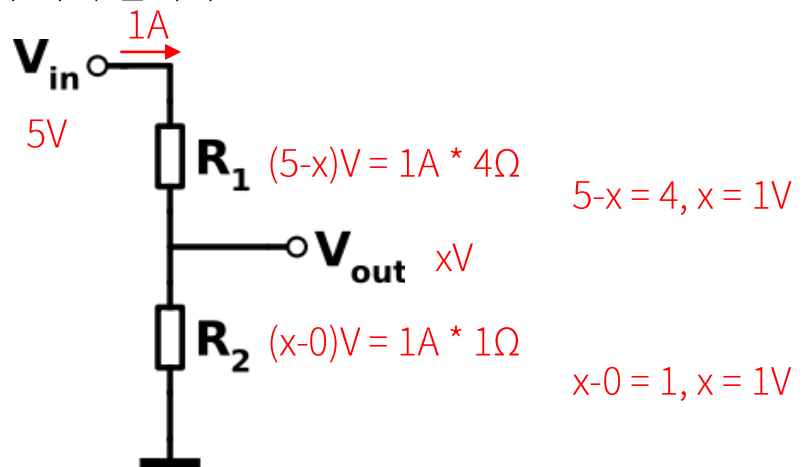


<그림1-3> 전압 분배 회로도

아두이노에서 전압은 보통 5V로 V_{in} 은 5V입니다.
이 때 R_1 과 R_2 를 합쳐서 5Ω 이라고 가정하면 전류는 1A가 됩니다. ($V = IR$)

이 때 R_1 과 R_2 의 분배에 따라서 V_{out} 이 달라지게 됩니다.

예를 들어 R_1 이 4Ω 이고 R_2 가 1Ω 인 경우에는 V_{out} 은 1이 되게 됩니다.



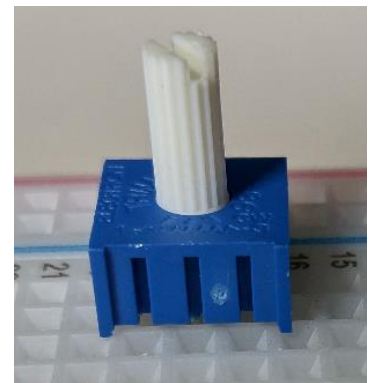
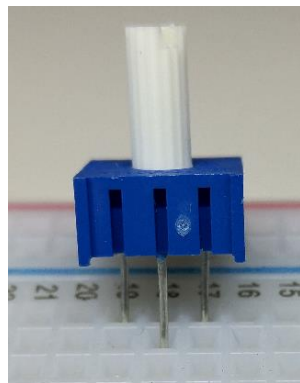
<그림1-4> 전압 분배 예시

만약 R_1 이 2이고 R_2 가 3인 경우에는 V_{out} 은 3이 되게 됩니다.

가변저항 하드웨어 구성

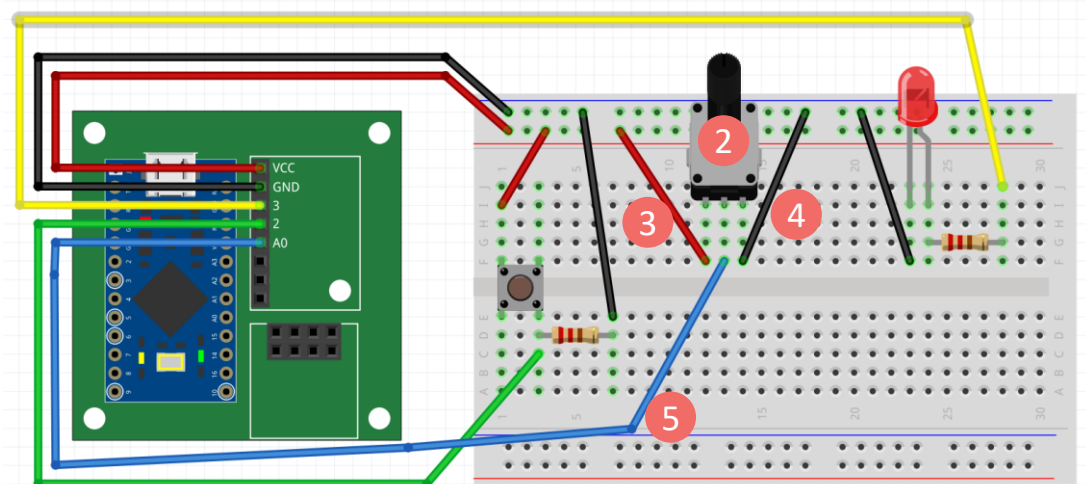
가변저항 하드웨어 구성

- 1 아두이노를 베이스 보드에 끼우고 USB를 연결합니다.(방향에 유의합니다)
- 2 가변저항을 브레드보드에 그림과 같이 꽂아 넣습니다.



<그림1-5> 가변저항 꽂는 법

- 3 3개의 가변 저항 다리 중 왼쪽다리를 빵판의 빨간줄과 연결합니다.
- 4 3개의 가변 저항 다리 중 오른쪽 다리를 빵판의 파란줄과 연결합니다.
- 5 3개의 가변 저항 다리 중 가운데 다리를 베이스 보드의 A0핀(위에서 5번째)에 연결합니다.



<그림1-6> 베이스 보드와 가변저항 연결

꿀TIP

가변저항 연결

가변저항의 가운데
다리로 신호를 받아야
합니다.

가변저항 예제 작성하기


가변저항 예제

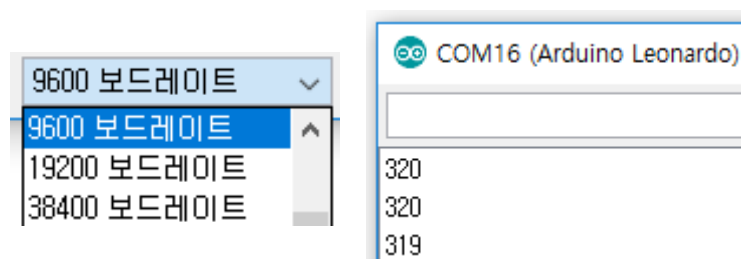
- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

ch4_1_1_variable

```
1 void setup() {
2     Serial.begin(9600);
3 }
4
5 void loop() {
6     int val = analogRead(A0);
7     Serial.println(val);
8     delay(500);
9 }
```

<그림1-7> 가변저항 예제

- 2  버튼을 눌러 시리얼 모니터를 켭니다.
- 3 보드레이트를 맞춘 후 가변저항을 돌렸을 때, 시리얼 모니터에서 값의 변화가 있는지 확인합니다.



<그림1-8> 값 확인

가변저항 해석

```
void setup() {  
  Serial.begin(9600); //시리얼 통신 시작  
}  
  
void loop() {  
  int val = analogRead(A0); //A0의 값을 받아 val에 저장  
  Serial.println(val); //val 값을 시리얼 모니터로 확인  
  delay(500); //0.5초 멈춤  
}
```



가변저항과 아날로그값에 익숙해졌다면, 아래 코드도 업로드 해 보고 어떻게 작동하는지 알아봅시다.


가변저항 활용

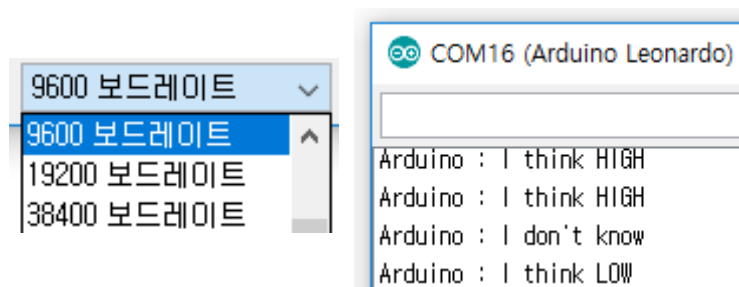
ch4_1_1_variable2

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   int val = analogRead(A0);  
7  
8   if (val > 614) {  
9     Serial.println("Arduino : I think HIGH");  
10  } else if (val < 614 && val > 307) {  
11    Serial.println("Arduino : I don't know");  
12  } else if (val < 307) {  
13    Serial.println("Arduino : I think LOW");  
14  }  
15  
16 // Serial.println(val);  
17   delay(500);  
18 }
```

<그림1-9> 아날로그값 활용

가변저항 활용 해석

- 1  버튼을 눌러 시리얼 모니터를 켭니다.
- 2 보드레이트를 맞추는 후 가변저항을 돌렸을 때, 시리얼 모니터에서 값의 변화가 있는지 확인합니다.



<그림1-10> 값 확인

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int val = analogRead(A0); //A0의 값을 받아 val에 저장

  if (val > 614) { //만약 val의 값이 614보다 크면
    Serial.println("Arduino : I think HIGH"); //문구 출력
  } else if (val < 614 && val > 307) { //val의 값이 614와 307 사이면
    Serial.println("Arduino : I don't know"); //문구 출력
  } else if (val < 307) { //만약 val의 값이 307보다 작으면
    Serial.println("Arduino : I think LOW"); //문구 출력
  }

  // Serial.println(val);
  delay(500); //0.5초 멈춤
}
```

실제 아두이노에서 HIGH와 LOW값을 읽어 들일 때에도 마찬가지로 일정 전압 이상이 들어오면 HIGH, 일정 전압보다 낮으면 LOW로 판단하게 됩니다.

02 digital과 analog

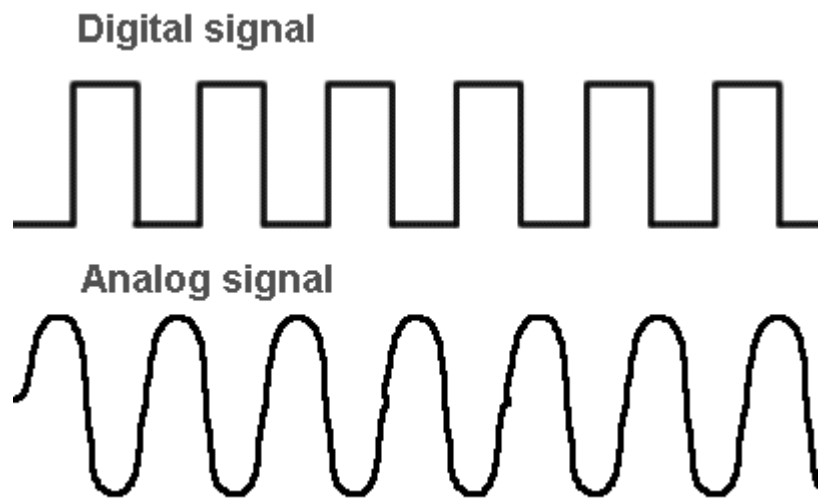


digital이란 단어를 들으면 어떤 느낌이 드나요?

digital은 현대 사회를 이루는 근간입니다. 흔히 설명할 때 analog와 대비하여 개념을 설명하게 됩니다. analog는 자연에서 흔히 발견되는 현상이며 사람의 목소리처럼 연속적으로 변하는 것이고, digital은 비연속적이며 셀 수 있는 수치로 구성되어 있습니다.

digital과 analog

digital은 0과 1을 표현할 때 주로 사용됩니다. 이와 달리 현실에서는 analog가 주로 사용됩니다. 키와 몸무게는 0과 1로 표현하기엔 부족합니다.



<그림2-1> 디지털과 아날로그

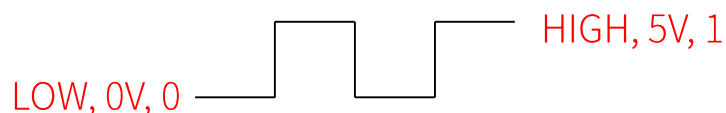
digital

digitalRead()

앞에서 배운 digitalRead() 함수는 아두이노의 특정 핀의 상태가 1인지 0인지를 읽어 들입니다. 이 때 읽어 들인 핀의 전압이 0V이면 LOW, 0으로 받아들이고 읽어 들인 핀의 전압이 5V이면 HIGH, 1로 받아들이습니다.

digitalWrite()

digitalWrite()함수는 아두이노의 특정 핀을 전압이 높은 HIGH상태로 만들거나 전압이 낮은 LOW상태로 만들 수 있습니다.



디지털 실습
해보기

- ① 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

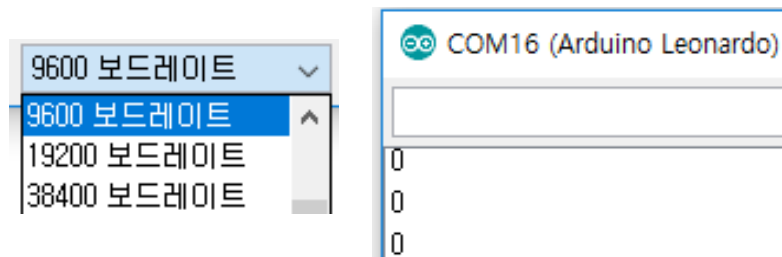
ch4_1_2_sw_led

```
1 void setup() {  
2     Serial.begin(9600);  
3     pinMode(3, OUTPUT);  
4     pinMode(2, INPUT);  
5 }  
6  
7 void loop() {  
8     int sw = digitalRead(2);  
9     Serial.println(sw);  
10    if (sw == HIGH) {  
11        digitalWrite(3, HIGH);  
12    } else {  
13        digitalWrite(3, LOW);  
14    }  
15 }
```

<그림2-2> 디지털 실습 해보기

- ②  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 스위치를 누르고 시리얼 모니터에서 값이 변하는지, LED의 밝기 변화가 있는지 확인합니다.



<그림2-3> 값 확인

디지털 실습 해석

```
void setup() {  
  Serial.begin(9600);  
  pinMode(3, OUTPUT);  
  pinMode(2, INPUT);  
}
```

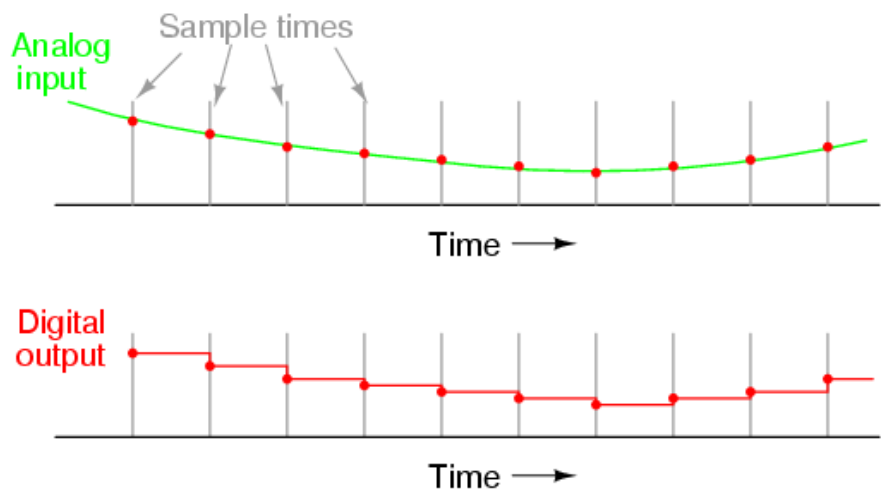
```
void loop() {  
  int sw = digitalRead(2); //2번핀의 값을 읽어서 sw에 저장  
  Serial.println(sw); //sw값 출력  
  if (sw == HIGH) { //만약 sw가 HIGH와 같다면  
    digitalWrite(3, HIGH); // LED 켜  
  } else { //그 외의 경우  
    digitalWrite(3, LOW); // LED 끄  
  }  
}
```

analog

analogRead()

아두이노에서 `analog`를 읽을 땐 A0 핀을 사용합니다. 이를 통해서 0에서부터 1023까지의 숫자를 받아들일 수 있습니다.

이 때 사용되는 개념이 ADC(Analog to Digital Converter)입니다. 쉽게 말해 아날로그 신호(0~5V의 전압)를 디지털인 0에서 1023의 숫자로 바꿔 주는 것입니다.



<그림2-4> ADC

위 그림에서처럼 연속적인 `analog`값들을 일정 구간별로 나누어 `digital`화하게 됩니다.

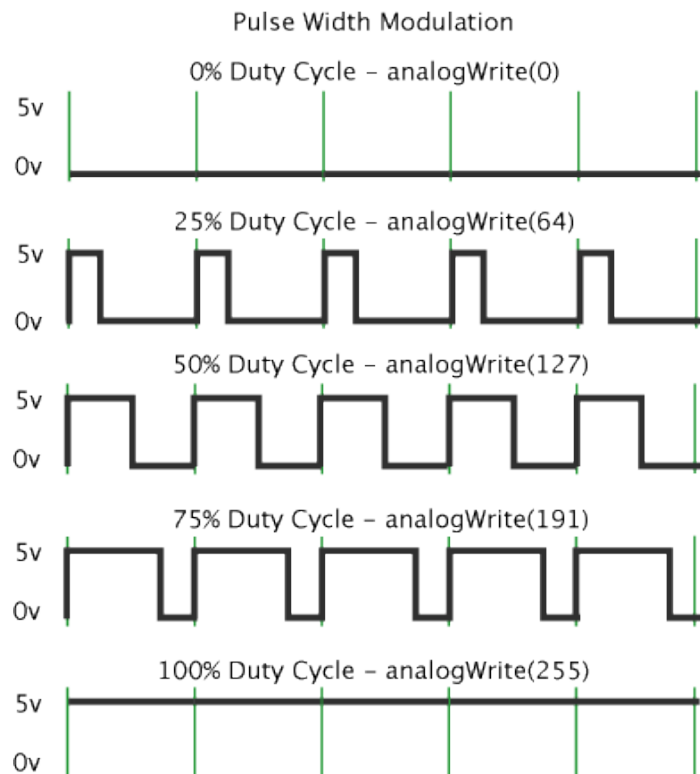
아두이노에서는 0V에서 5V까지를 1024단계로 나누어 $5/1024 = 49\text{mV}$ 단위로 `analog`값을 적용할 수 있게 됩니다.

analog

analogWrite()

analogWrite()은 0에서 255까지의 값을 내보낼 수 있습니다. 이를 통해 LED의 밝기를 256단계로 조절할 수 있게 됩니다.

여기선 PWM(Pulse Width Modulation)이란 개념이 적용됩니다. PWM은 펄스의 폭을 제어하여 출력을 조절하는 방식입니다.



<그림2-5> PWM

위 그림에서처럼 analogWrite(0)을 했을 때엔 0V가 유지됩니다.

두 번째 그림에서 analogWrite(64)를 하게 될 경우 64는 255의 1/4이기에 1/4만큼 5V를 주게되어 5V와 0V가 1:3 비율로 적용되어 전압이 나타나게 됩니다.


아날로그
실습
해보기

- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

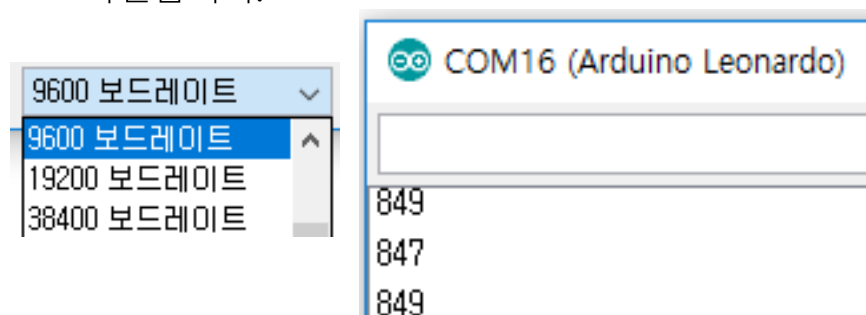
ch4_1_2_variable_led

```
1 void setup() {  
2     Serial.begin(9600);  
3     pinMode(3, OUTPUT);  
4     pinMode(2, INPUT);  
5 }  
6  
7 void loop() {  
8     int val = analogRead(A0);  
9     Serial.println(val);  
10    analogWrite(3, val / 5);  
11 }
```

<그림2-6> 아날로그 실습 해보기

- 2  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 가변저항을 돌려서 시리얼 모니터에서 값이 변하는지, LED의 밝기 변화가 있는지 확인합니다.



<그림2-7> 값 확인

아날로그 실습 해석

```
void setup() {  
  Serial.begin(9600);  
  pinMode(3, OUTPUT);  
  pinMode(2, INPUT);  
}  
  
void loop() {  
  int val = analogRead(A0); //A0핀의 값을 받아 val에 저장  
  Serial.println(val); //val값 출력  
  analogWrite(3, val / 5); //val을 5로 나눈 값을 3번핀에 출력  
}
```


03 모터에 대해서



드론에 들어가는 모터는 대략 두 종류가 있습니다.

DC모터는 보통 손바닥만한 드론이나 그 보다 작은 초소형 드론에 많이 사용됩니다.

BLDC모터는 그보다 큰 중형에서 대형 드론에 사용되며 DC모터에 비해 비싸지만 그만큼 내구성이 좋다는 특징이 있습니다.

모터란?

모터

우리가 흔히 아는 모터는 전류를 받았을 때 회전을 하게 되는 원기동 형태의 물체입니다.

좀 더 상세히 말하자면, 모터는 전동기라고 불리며 전류가 흐르는 도체가 자기장 속에서 받는 힘을 이용하여 전기에너지를 역학적 에너지로 바꾸는 장치입니다.



<그림1-1> 모터

모터의 종류

흔히 사용되는 모터에는 다음과 같이 DC모터, BLDC모터, Servo모터, 스텝 모터 등이 있습니다.(사실 명확한 분류는 아니지만 편의상 4종류로 구분을 합니다.)



<그림1-2> 좌측부터 DC, BLDC, 서보, 스텝 모터

드론의 날개 회전을 담당하는 모터는 DC모터 또는 BLDC모터이며, 소형 드론에서는 DC, 대형 드론에서는 BLDC모터가 주로 사용됩니다.

서보모터는 모터가 특정 각도를 유지해야하는 상황에 주로 사용되며, 드론의 카메라 짐벌을 만들 때도 사용됩니다.

DC모터

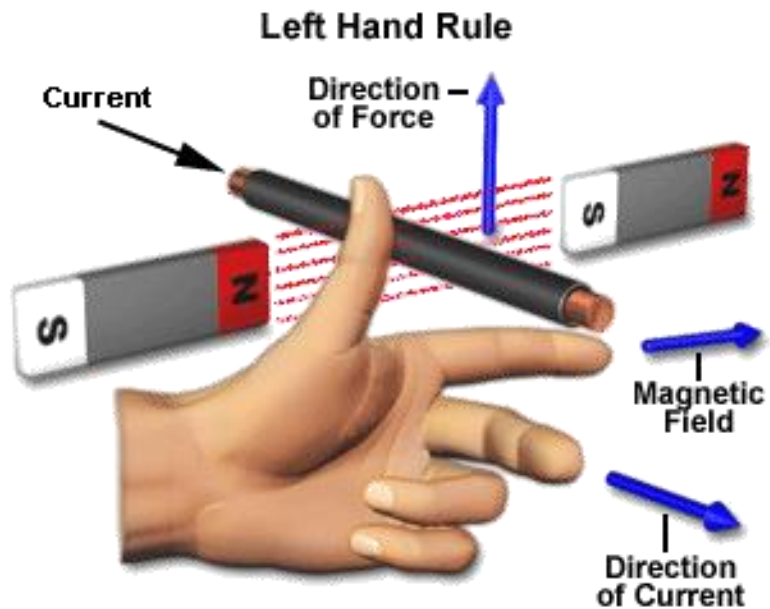
DC모터는 건전지나 배터리 등으로부터 공급되는 직류(DC)전압으로 작동되는 모터입니다.
우리가 흔히 알고있는 모터가 DC모터입니다.



<그림1-7> DC 모터

플레밍의
원손 법칙

DC모터가 도는 원리는 플레밍의 왼손법칙(FBI)에 따라 전류와 자기장의 방향에 의해 힘을 받게 되는 원리입니다.



<그림1-8> 플레밍의 왼손 법칙

이 때, 엄지 손가락이 F(힘), 검지 손가락이 B(자기장), 중지 손가락이 I(전류)가 됩니다.

배터리 충전하기

충전기와 배터리

이번 시간에는 배터리를 사용하여 모터를 돌려볼 것입니다.
그전에 배터리를 충전하는 방법에 대해 알아보시다.



<그림2-1> 충전기와 배터리

충전기 연결

- 1 충전기의 USB를 컴퓨터에 꽂고, 꼬리 부분을 배터리와 연결합니다.



<그림2-2> 컴퓨터와 연결



<그림2-3> 배터리 연결

꿀TIP

충전기 종류

충전기의 종류에 따라
충전 중일 때 불이
들어오는 충전기가 있고,
충전이 완료 되었을 때
불이 들어오는 충전기가
있습니다.

- 2 충전중일 땐 불이 꺼져 있고, 완충 시 빨간 불이 켜지게 됩니다.
- 3 1시간 정도 후 충전기에 빨간 불이 들어오면 충전기와 배터리를 분리합니다.

배터리 안전

배터리 안전 유의 사항

드론은 방전률이 높은 리튬폴리머 배터리를 사용합니다. 이러한 고 방전률을 가진 리튬폴리머 배터리는 한번에 많은 전류를 내보내, 모터가 원활히 돌 수 있게 해줍니다.

하지만, 고 방전 배터리라는 특성 때문에 폭발 위험이 있어 사용에 유의해야 합니다.



<그림2-4> 배터리 화재

링크 : <https://www.youtube.com/watch?v=0boQGKE-6Y4>

충전중에는 절대 자리를 비우면 안되고, 배터리가 부풀어 오르는 등의 이상이 있는지 확인해야 합니다. 배터리에 이상이 있으면 즉시 충전을 멈춘 후, 배터리를 분리하고 멀리 떨어뜨려 놓습니다.

배터리의 보관은 50퍼센트 정도 충전한 후 직사광선이 없는 서늘한 곳에 보관해야 합니다.

배터리
안전 수칙

배터리 안전수칙

1. 충전기를 켜 둔 채 방치하지 않습니다.
2. 충전 중 부풀어 오르는지, 뜨거워지는 확인합니다.
3. 배터리를 분해하거나 압력을 가하지 않습니다.

모터 안전

모터 사용 안전 유의 사항

드론의 모터가 회전할 때 약 36000rpm으로 회전을 하게 됩니다. rpm(revolutions per minute)은 분당 회전수로 36000rpm이면 대략 1초에 600번 회전을 합니다.

이런 속도로 회전하는 프로펠러에 손가락을 집어넣으면 큰 상처를 입거나 잘못하면 손가락이 잘릴 수 있으니 굉장히 조심해야 합니다.



<그림2-5> 드론 모터 위험

링크: <https://www.youtube.com/watch?v=Hyp3JLHPHw>

위 영상에서는 플라스틱도 쉽게 뚫리는 것을 확인할 수 있습니다. 또한, 회전을 막 마친 모터의 경우 온도가 높아 화상의 위험이 있습니다.

모터
안전 수칙

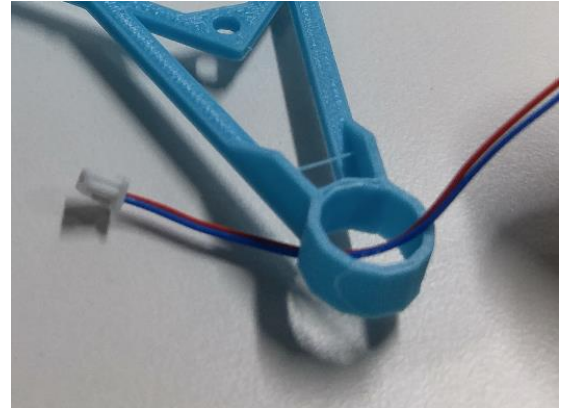
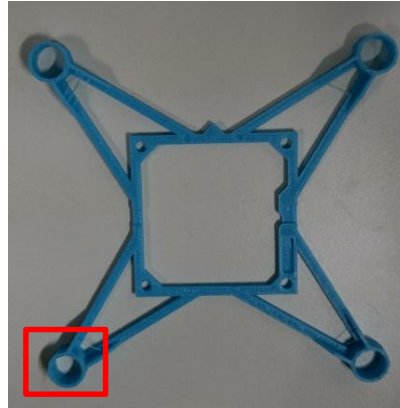
모터 안전수칙

1. 스위치가 켜진 상태에서 모터를 만지지 않습니다.
2. 회전 중 프로펠러가 빠지지 않도록 꼭 끼웁니다.
3. 막 회전을 마친 모터를 손으로 잡지 않습니다.

모터 연결하기

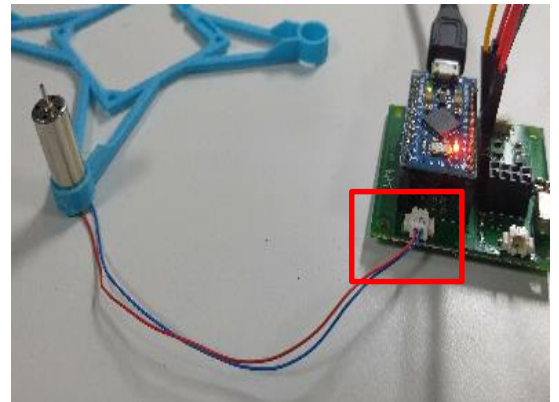
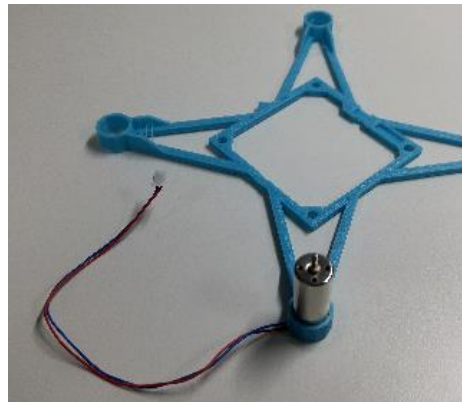
모터 연결 구성

- 1 드론 몸체의 좌측 하단에 빨파모터(선이 빨강, 파랑)를 꼬리부터 넣어서 절반정도 끼웁니다.(너무 꽉 끼우지 않습니다.)



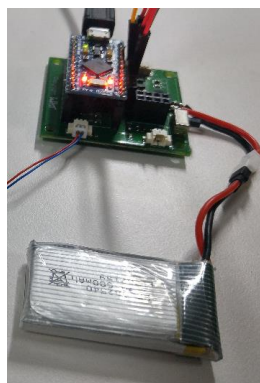
<그림3-1> 모터 연결하기

- 2 베이스 보드의 좌측 하단에 모터 꼬리를 연결합니다.



<그림3-2> 모터 연결하기

- 3 베이스 보드에 배터리를 연결하고, 모터에 R프로펠러를 끼웁니다. (L을 끼울 시 바람이 밑으로 나옵니다)



<그림3-3> 모터 연결하기

가변저항으로 모터 제어


가변저항으로 모터 제어

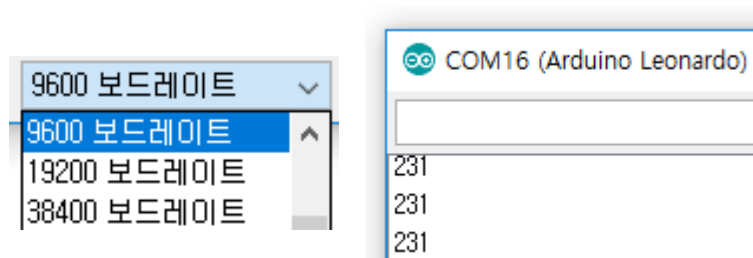
- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

ch4_3_3_motor_variable

```
1 void setup() {
2     Serial.begin(9600);
3     pinMode(5, OUTPUT);
4 }
5
6 void loop() {
7     int val = analogRead(A0);
8     Serial.println(val);
9     analogWrite(5, val / 10);
10 }
```

<그림3-4> 가변저항으로 모터 제어 코드

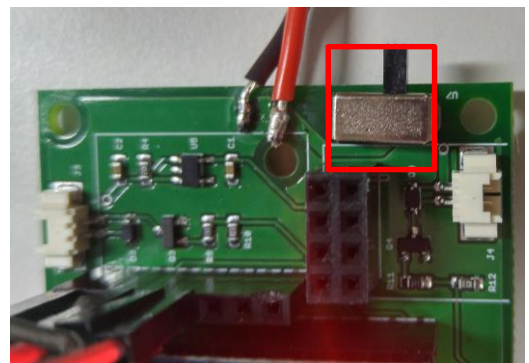
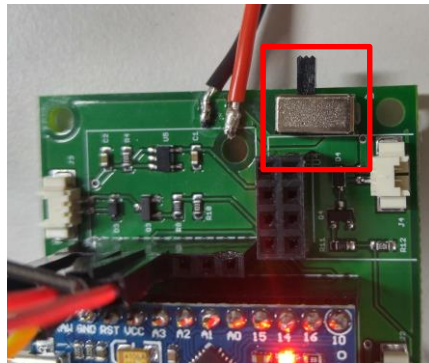
- 2  버튼을 눌러 시리얼 모니터를 켭니다.
- 3 보드레이트를 맞춘 후 가변저항을 돌려 시리얼 모니터에서 값이 변하는지 확인합니다.



<그림3-5> 값 확인

가변저항으로
모터 제어
해석

- 4 시리얼 모니터에서 확인되는 값을 0에 맞춘 후, 베이스 보드의 스위치를 켭니다.



<그림3-6> 베이스 보드 스위치 켜기

- 5 가변저항을 돌려 값이 올라감에 따라 모터가 회전하는 것을 확인합니다.

```
void setup() {  
  Serial.begin(9600);  
  pinMode(5, OUTPUT); //5번핀(모터)를 출력으로 정합니다.  
}  
  
void loop() {  
  int val = analogRead(A0); //가변저항의 값을 읽어옵니다.  
  Serial.println(val); //저장된 값을 출력합니다.  
  analogWrite(5, val / 10); //val/10 만큼 모터를 동작시킵니다.  
}
```

스위치로 모터 제어

스위치로
모터 제어

- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

ch4_3_3_motor_sw

```

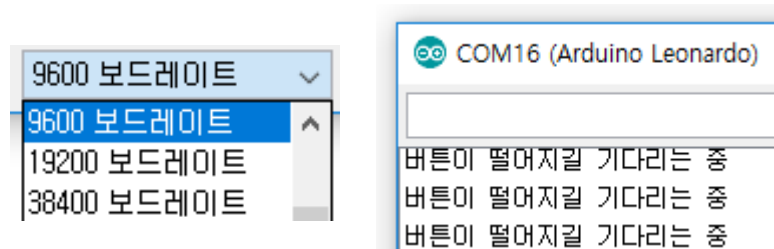
1 int cnt = 0;
2
3 void setup() {
4   Serial.begin(9600);
5   pinMode(2, INPUT);
6   pinMode(5, OUTPUT);
7 }
8
9 void loop() {
10  int val = getCnt();
11  analogWrite(5, val * 30);
12 }
13
14 int getCnt() {
15  if (digitalRead(2) == HIGH) {
16    cnt++;
17    if (cnt > 3) cnt = 0;
18    while (digitalRead(2) == HIGH) {
19      Serial.println("버튼이 떨어지길 기다리는 중");
20    }
21    Serial.print("버튼 눌림 횟수 : ");
22    Serial.println(cnt);
23  }
24  return cnt;
25 }

```

<그림3-7> 스위치로 모터 제어

- 2  버튼을 눌러 시리얼 모니터를 켭니다.

- 3 보드레이트를 맞춘 후 스위치를 누르며 시리얼 모니터에서 값이 변하는지, 모터가 작동하는지 확인합니다.



<그림3-8> 값 확인

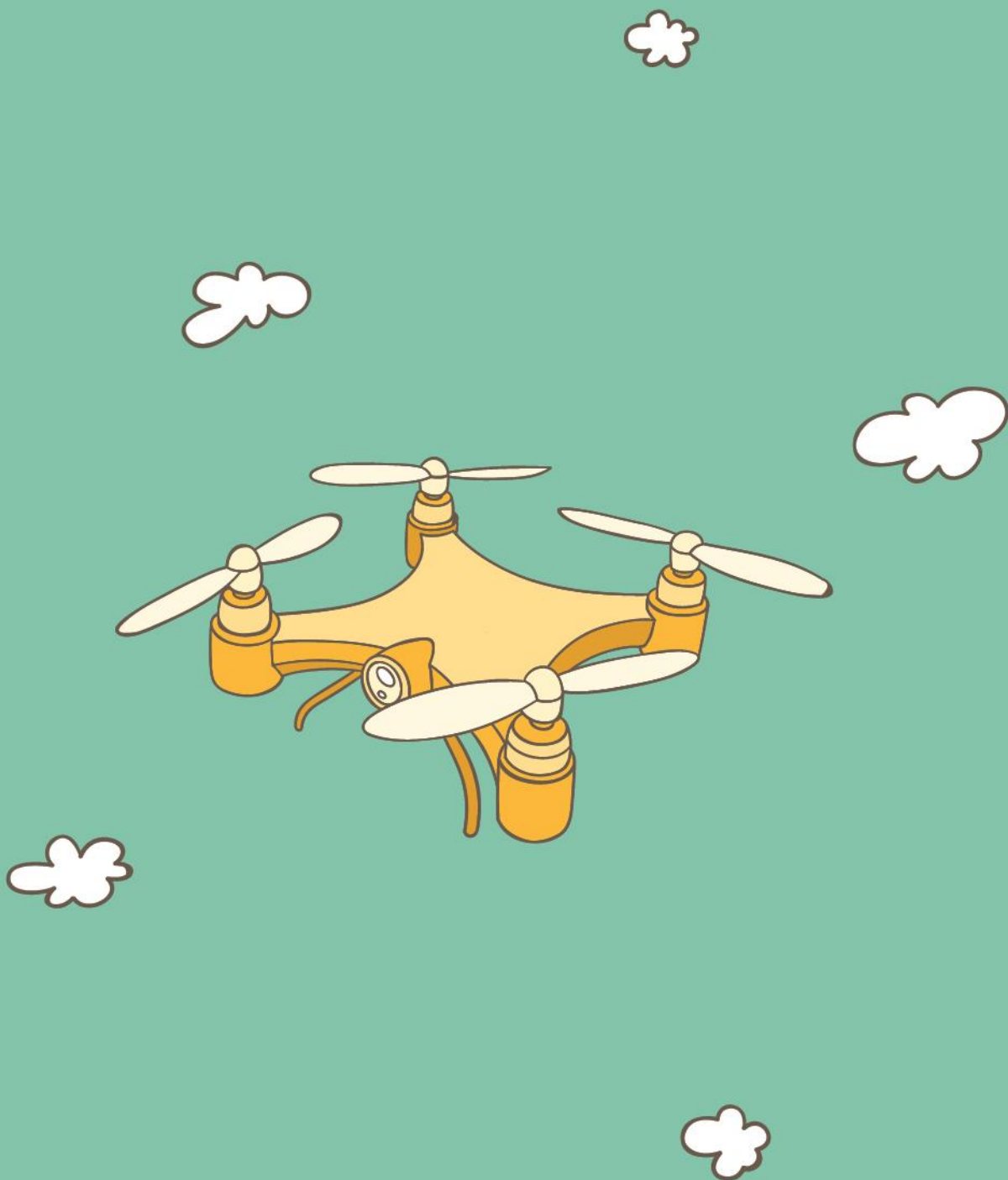
- 4 스위치를 눌러보며 모터의 속도가 변하는지 확인합니다.

```
int cnt = 0; //스위치 눌림 횟수 카운트 변수

void setup() {
  Serial.begin(9600);
  pinMode(2, INPUT);
  pinMode(5, OUTPUT);
}

void loop() {
  int val = getCnt(); //함수 호출 및 출력값 저장
  analogWrite(5, val * 30); //출력값*30을 모터에 전달
}

int getCnt() { // 스위치 눌린 횟수 받는 함수 작성
  if (digitalRead(2) == HIGH) { //만약 스위치가 눌렸다면
    cnt++; // cnt값 증가
    if (cnt > 3) cnt = 0; //만약 cnt가 4이상이면 0으로
    while (digitalRead(2) == HIGH) { //스위치가 눌린 상태면 반복
      Serial.println("버튼이 떨어지길 기다리는 중"); //문구 출력
    }
    Serial.print("버튼 눌림 횟수 : ");
    Serial.println(cnt); //스위치 눌림 횟수 출력
  }
  return cnt; //cnt값 반환
}
```



WHIT