



드론으로 배우는
프로그래밍 교실

Ch6-2. 자이로, 가속도 I2C



01 자이로와 가속도 센서	01
자이로 센서란?	02
코리올리 효과	03
가속도 센서란?	04
혹의 법칙	05
MPU6050 센서	06
02 I2C 통신	07
I2C	08
Wire 라이브러리	09
MPU6050 온도	10
MPU6050 가속도	13
MPU6050 자이로	15
모터 연결하기	17
03 에어 마우스 만들기	18
에어 마우스란?	19
에어 마우스 코드	20



드론으로 배우는
프로그래밍 교실

초판발행 2016년 9월 23일
지은이 이상준 | 펴낸이 CodingBird
펴낸곳 WHIT | 주소 안산시 한양대학교로55 창업보육센터 B01

Published by WHIT. Printed in Korea
Copyright © 2016 CodingBird & WHIT

이 책의 저작권은 CodingBird와 WHIT에 있습니다.
저작권법에 의해 보호를 받는 저작물이므로
무단 복제 및 무단 전재를 금합니다.

01 자이로와 가속도 센서



자이로 센서는 회전값을, 가속도 센서는 가속값을 알아내는데 이용할 수 있습니다.

이러한 자이로 센서와 가속도 센서는 핸드폰에서도 쓰입니다.

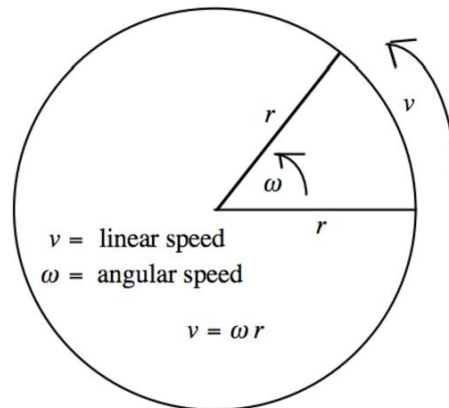
핸드폰을 세로에서 가로로 변할 때 화면이 회전하는 것을 본 적 있나요?

핸드폰 내 자이로 센서와 가속도 센서가 기울어짐을 감지해 핸드폰이 어떤 방향으로 위치하고 있는지 알 수 있습니다.

자이로 센서란?

자이로 센서

자이로 센서는 회전하는 물체의 각속도를 측정하는 센서입니다. 속도는 시간동안 움직인 거리를 나타내는 물리량인데, 각속도는 그럼 무엇일까요?



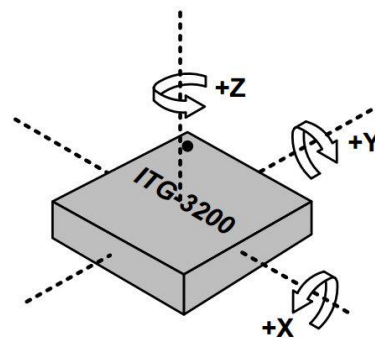
<그림1-1> 각속도

각속도는 시간동안 각도가 얼마나 변했는지를 나타내는 물리량입니다.

예를 들어 바퀴가 1초 동안 3바퀴 돌았다면

$$3 * 360\text{도} / 1\text{초} = 1080\text{DPS(Degree Per Second)}$$

$$3 * 2\pi / 1\text{초} = 6\pi \text{ rad/sec}$$

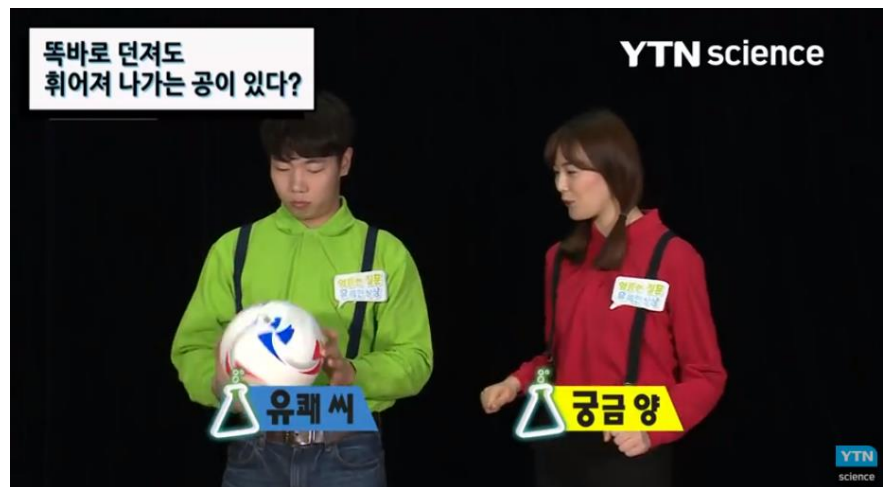


<그림1-2> 자이로 센서

위 그림처럼 자이로 센서는 물체가 x축, y축, z축으로 각각 회전함에 따라 발생하는 각속도를 읽어서 어느 방향으로 회전하고 있는지를 알아낼 수 있습니다.

전향력 (코리올리 힘)

자이로 센서는 어떤 원리로 작동하게 되는 걸까요?
회전하는 물체에는 코리올리힘이라는 가상의 힘이 작용하게 됩니다. 이를 이용하여 회전값을 측정할 수 있습니다.



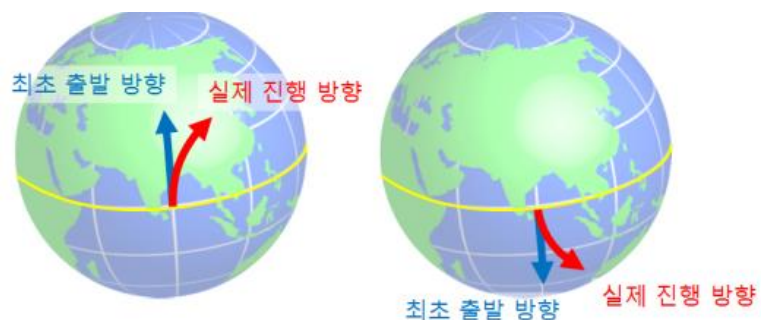
<그림1-3> 코리올리 효과

출처 : <https://www.youtube.com/watch?v=VEtUU1HJq-4>

영상에서처럼, 회전계 내부에서는 공이 휘어져서 나가게 되고, 회전계 외부에서는 직선으로 나가게 됩니다. 이렇게 휘어져서 나가게 되는 정도를 센서가 받아들이게 되고, 이 휘어짐 정도에 따라 회전이 얼마나 됐는지 구하게 됩니다.

자전과 코리올리

지구는 자전하기 때문에 최초 출발방향으로 미사일을 발사하여도, 북반구에서는 약간 오른쪽으로, 남반구에서는 약간 왼쪽으로 휘어지면서 이동하는 것처럼 보입니다.



<그림1-4> 자전과 코리올리

가속도 센서란?

가속도 센서

가속도 센서는 말 그대로 가속도를 측정하는 센서입니다. 우리 몸에 항상 작용하는 가속도 중 하나인 중력 가속도도 가속도 중 하나입니다.

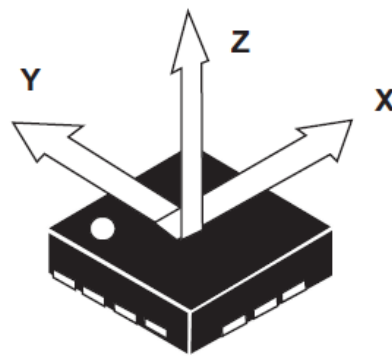
이러한 가속도는 뉴턴의 제 2법칙인 가속도의 법칙에 의해 구해집니다.

$$F=ma$$

<그림1-5> $F=ma$

몸무게가 무거운 사람은 밀어도 잘 밀리지 않습니다. 이렇게 일정한 힘이 주어졌을 때 질량에 따라 가속도는 달라지게 됩니다.

이렇게 가속도 센서는 얼마나 빠른 속도로 가속을 하게 되는지 측정할 수 있는 센서입니다.



<그림1-6> 가속도 센서

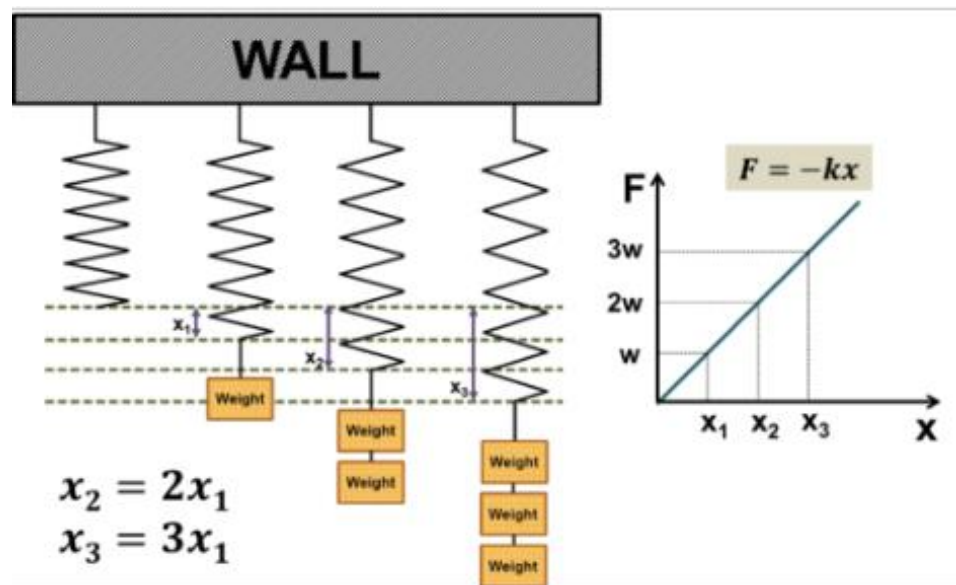
위 그림처럼 가속도 센서는 물체가 x축, y축, z축으로 각각 얼마 만큼의 가속도를 가지고 있는지 측정할 수 있으며 단위는 m/s^2 입니다.

훅(Hooke)의 법칙

이러한 가속도 센서는 어떻게 작동하는걸까요?
 훅의 법칙(Hooke's Law)은 용수철과 같이 탄성있는 물체가
 외력에 의해 늘어나거나 줄어드는 등 변형 될때, 원래
 모습으로 돌아오려고 저항하는 복원력의 크기와 변형의
 정도의 관계를 나타내는 물리법칙입니다.

$$F = kx$$

<그림1-7> 훅의 법칙



<그림1-8> 훅의 법칙 설명

$F = ma$ 랑 $F = kx$ 를 통해 가속도를 알아낼 수 있습니다.

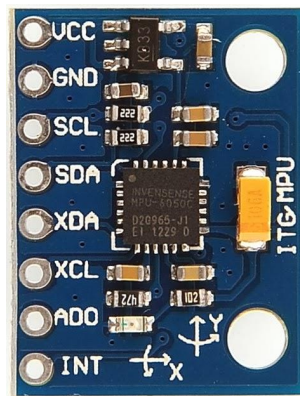
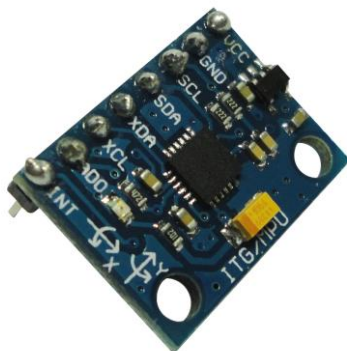
- $ma = F = kx$
- $ma = kx$
- $a = kx/m$

이 때, k 는 상수이고, m 은 질량으로 일정합니다. 즉,
 용수철의 늘어난 길이 x 를 통해 가속도 a 를 구할 수
 있습니다.

MPU6050 센서

MPU6050

GY-521 MPU 6050 은 3축 자이로 센서와 3축 가속도 센서, 온도 센서가 한 칩 안에 구성되어 있는 센서입니다.



<그림1-9> MPU6050

GY-521 모듈은 아두이노와 I2C통신을 하게 됩니다. 이 때 VCC, GND, SCL, SDA의 4개의 핀을 이용합니다.

PIN	설명
VCC	전압
GND	접지
SCL	I2C Serial Clock
SDA	I2C Serial Data
XDA	마스터로 동작할 때 사용 (외부센서와 연결시)
XCL	마스터로 동작할 때 사용 (외부센서와 연결시)
AD0	I2C slave 주소
INT	인터럽트핀

<그림1-10> MPU6050 핀

02 I2C 통신



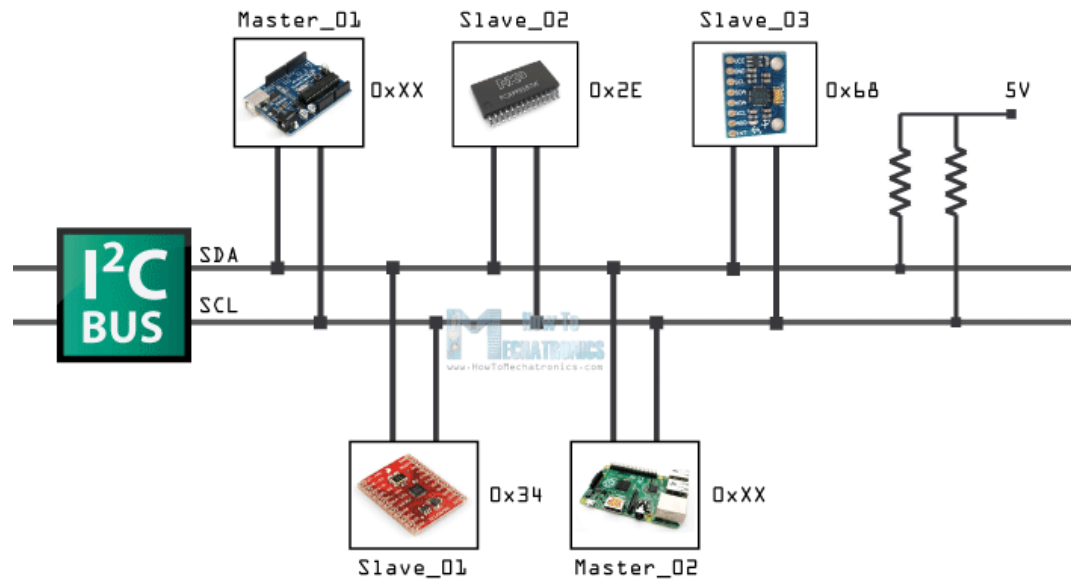
아두이노의 통신에는 시리얼 통신만 있는게 아닙니다.

통신 방식에는 I2C통신이나 SPI통신 등 다양한 통신 방법이 있는데, 앞서 배운 자이로, 가속도 센서를 사용할 땐 I2C통신을 사용하게 됩니다.

I2C통신이라고도 불리는 I2C통신은 단 두개의 선 연결만으로 최대 128개의 장치가 연결될 수 있습니다.

Inter
Integrated
Circuit

I2C통신은 2개의 신호선을 통해 여러 개의 디바이스들끼리 정보를 주고 받을 수 있는 통신 방식입니다.



<그림2-1> I2C통신

I2C통신
특징

이 때 사용되는 두 개의 핀은 각각 SCL, SDA로 각각 클락과 데이터를 담당합니다.

- SCL(Serial Clock) : 일정 주기의 클락신호를 내보낸다.
- SDA(Serial Data) : 클락신호에 따라 데이터를 보낸다.

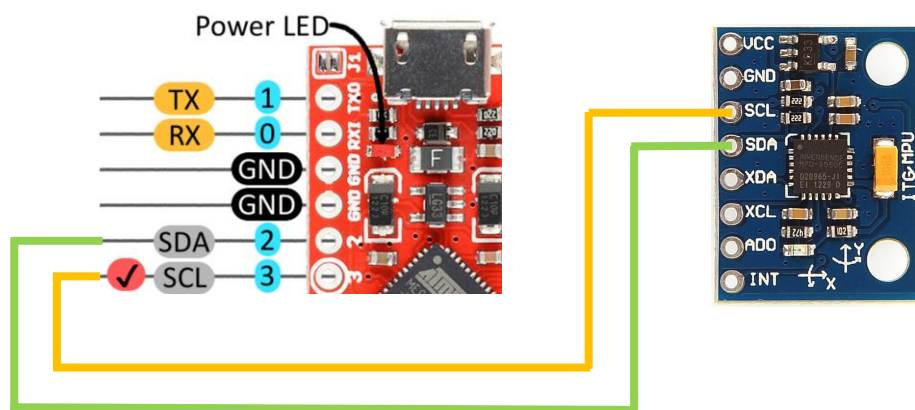
I2C통신에서 각각의 디바이스들은 고유의 주소를 가지고 있고, 이 주소를 통해서 서로를 구분할 수 있습니다.

각 디바이스는 마스터 또는 슬레이브로 동작할지가 정해지게 되는데, 대부분 아두이노가 마스터로, 센서가 슬레이브로 동작하게 됩니다.

이 때 마스터와 슬레이브는 둘 다 데이터의 입력과 출력이 가능합니다.

아두이노 I2C 연결

아두이노에서 I2C 통신을 할 때에는 다음과 같이 2번, 3번핀이 사용됩니다.(수업에서 사용하는 메인 보드를 통해 자체적으로 연결이 되어 있습니다)



<그림2-2> I2C 연결

Wire

Wire라이브러리는 아두이노에서 I2C통신을 쉽게 사용할 수 있게 해줍니다.

함수	설명
Wire.begin()	마스터 모드로 I2C통신을 시작합니다.
Wire.begin(addr)	슬레이브 모드(주소는 addr)로 I2C통신을 시작합니다.
Wire.beginTransmission(addr)	마스터에서 슬레이브로 전송을 시작하기 위한 슬레이브의 주소를 지정합니다.
Wire.write(value)	전송될 데이터를 임시 저장합니다.
Wire.endTransmission(true)	시작신호, 슬레이브 주소, 데이터, 정지신호 등을 전송합니다.
Wire.requestFrom(addr, quantity, true)	addr라는 주소의 슬레이브에게 quantity만큼의 데이터를 요청합니다.

MPU6050
온도

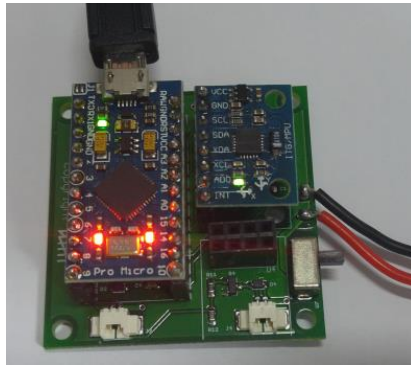
- 1 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

```


ch6_1_2_tmp
1
2
3 const int MPU_ADDRESS = 0x68;
4 int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
5
6 void setup() {
7   Wire.begin();
8   Wire.beginTransmission(MPU_ADDRESS);
9   Wire.write(0x6B);
10  Wire.write(0);
11  Wire.endTransmission(true);
12  Serial.begin(115200);
13  pinMode(5, OUTPUT);
14 }
15
16 void loop() {
17   Wire.beginTransmission(MPU_ADDRESS);
18   Wire.write(0x3B);
19   Wire.endTransmission(false);
20   Wire.requestFrom(MPU_ADDRESS, 14, true);
21
22   AcX = Wire.read() << 8 | Wire.read();
23   AcY = Wire.read() << 8 | Wire.read();
24   AcZ = Wire.read() << 8 | Wire.read();
25   Tmp = Wire.read() << 8 | Wire.read();
26   GyX = Wire.read() << 8 | Wire.read();
27   GyY = Wire.read() << 8 | Wire.read();
28   GyZ = Wire.read() << 8 | Wire.read();
29
30   // Serial.print("AcX = "); Serial.println(AcX);
31   // Serial.print("AcY = "); Serial.println(AcY);
32   // Serial.print("AcZ = "); Serial.println(AcZ);
33   Serial.print("Tmp = "); Serial.println(Tmp / 340.00 + 36.53);
34   // Serial.print("GyX = "); Serial.println(GyX);
35   // Serial.print("GyY = "); Serial.println(GyY);
36   // Serial.print("GyZ = "); Serial.println(GyZ);
37
38   //if (GyX < 0) GyX = -GyX;
39   //analogWrite(5, GyX / 400);
40 }

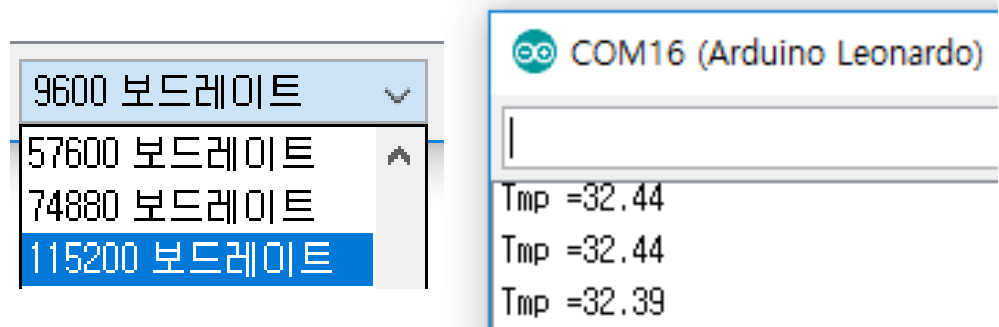
```

- 2 아두이노와 자이로 센서를 메인 보드에 끼워 넣습니다.



<그림2-4> 아두이노 자이로 연결

- 3 아두이노를 컴퓨터와 USB로 연결합니다.
- 4  버튼을 눌러 시리얼 모니터를 켭니다.
- 5 보드레이트를 맞춘 후, 자이로 센서에 손을 올려 온도가 변하는 것을 확인합니다.



<그림2-5> 시리얼 통신 확인

Tmp는 Temperature의 줄임말로, MPU6050센서에서 온도를 확인할 수 있습니다.

이렇게 자이로, 가속도 센서가 온도센서와 함께 존재하는 이유는 자이로, 가속도를 측정하는 센서가 온도의 영향을 받기 때문입니다.

꿀TIP

MPU6050 온도

코드

<https://goo.gl/64M7DL>

MPU6050

온도 해석

```
#include<Wire.h>

const int MPU_ADDRESS = 0x68; //센서 주소값
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ; //가속도, 자이로,
온도값을 받을 변수

void setup() {
  Wire.begin(); //I2C 실행
  Wire.beginTransmission(MPU_ADDRESS); //센서 주소 설정
  Wire.write(0x6B); //보낼 데이터 저장
  Wire.write(0); //보낼 데이터 저장
  Wire.endTransmission(true); //데이터 전송
  Serial.begin(115200);
  pinMode(5, OUTPUT);
}

void loop() {
  Wire.beginTransmission(MPU_ADDRESS); //센서 주소 설정
  Wire.write(0x3B); // 보낼 데이터 저장
  Wire.endTransmission(false); //데이터 전송
  Wire.requestFrom(MPU_ADDRESS, 14, true); //반환값 확인

  AcX = Wire.read() << 8 | Wire.read(); //데이터 저장
  AcY = Wire.read() << 8 | Wire.read();
  AcZ = Wire.read() << 8 | Wire.read();
  Tmp = Wire.read() << 8 | Wire.read();
  GyX = Wire.read() << 8 | Wire.read();
  GyY = Wire.read() << 8 | Wire.read();
  GyZ = Wire.read() << 8 | Wire.read();

  // Serial.print("AcX="); Serial.println(AcX); // 데이터 확인
  // Serial.print("AcY="); Serial.println(AcY);
  // Serial.print("AcZ="); Serial.println(AcZ);
  Serial.print("Tmp="); Serial.println(Tmp / 340.00 + 36.53);
  // Serial.print("GyX="); Serial.println(GyX);
  // Serial.print("GyY="); Serial.println(GyY);
  // Serial.print("GyZ="); Serial.println(GyZ);

  // if(AcX < 0) AcX = -AcX;
  // analogWrite(5, AcX / 200);
}
```


MPU6050 자이로

1 위에서 작성한 코드를 다음과 같이 변형하여 아두이노에 업로드합니다.

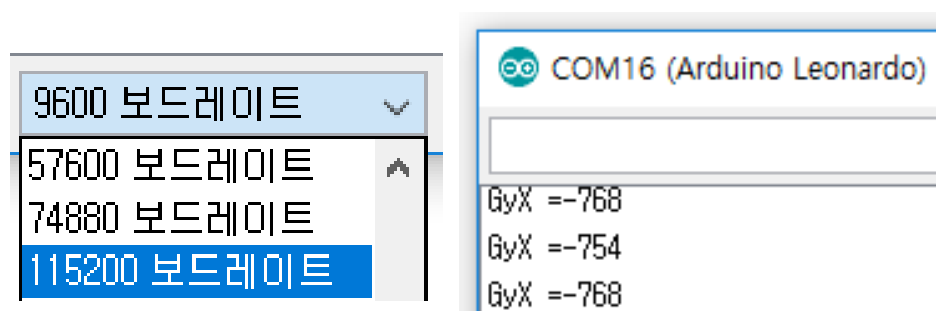
- 33번째줄에 주석을 넣습니다.
- 34번째줄에 주석을 해제합니다.
- 38, 39번째 줄의 주석을 해제합니다.

```
30 // Serial.print("AcX ="); Serial.println(AcX);
31 // Serial.print("AcY ="); Serial.println(AcY);
32 // Serial.print("AcZ ="); Serial.println(AcZ);
33 // Serial.print("Tmp ="); Serial.println(Tmp / 340.00 + 36.53);
34 Serial.print("GyX ="); Serial.println(GyX);
35 // Serial.print("GyY ="); Serial.println(GyY);
36 // Serial.print("GyZ ="); Serial.println(GyZ);
37
38 if (GyX < 0) GyX = -GyX;
39 analogWrite(5, GyX / 400);
40 }
```

<그림2-6> MPU6050 자이로 확인

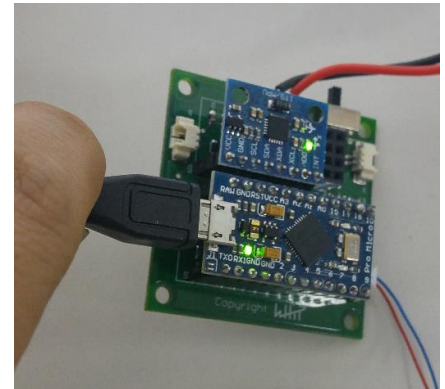
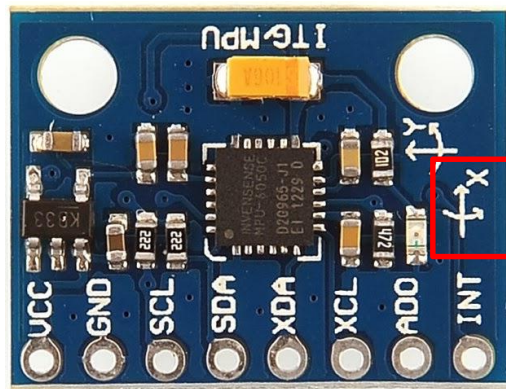
2  버튼을 눌러 시리얼 모니터를 켭니다.

3 보드레이트를 맞춘 후, GyX의 값을 확인합니다.



<그림2-7> 시리얼 통신 확인

- 4 배터리와 모터를 연결하고 스위치를 켭니다.(17페이지 참조)
- 5 자이로 센서의 x축을 넘어서는 회전 방향을 시킬 때 모터 속도의 변화를 확인합니다.



<그림2-8> MPU6050 방향 변화

MPU6050 자이로 해석

```
// Serial.print("AcX="); Serial.println(AcX);
// Serial.print("AcY="); Serial.println(AcY);
// Serial.print("AcZ="); Serial.println(AcZ);
// Serial.print("Tmp="); Serial.println(Tmp / 340.00 + 36.53);
Serial.print("GyX="); Serial.println(GyX); // 데이터 확인
// Serial.print("GyY="); Serial.println(GyY);
// Serial.print("GyZ="); Serial.println(GyZ);
```

```
if(GyX < 0) GyX = -GyX; // 만약 GyX가 0보다 작으면 양수로 전환
analogWrite(5, GyX / 400); // 모터 출력
}
```

GyX는 Gyro Sensor의 줄임말로 x축으로의 회전을 의미합니다. GyX의 값은 대략 32000 ~ -32000 사이의 값을 가집니다. x축을 넘어서는 회전 방향으로 회전을 했을 때 최대값을 가집니다.

이는 회전 방향으로 작용하는 각속도의 값을 의미합니다.


MPU6050 가속도

1 위에서 작성한 코드를 다음과 같이 변형하여 아두이노에 업로드합니다.

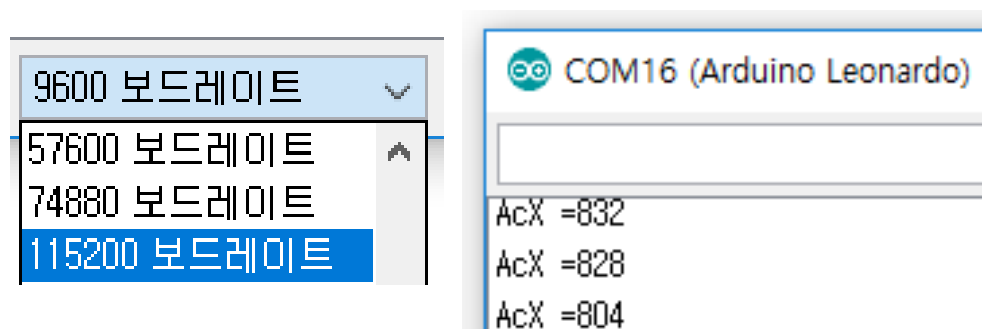
- 34번째줄에 주석을 넣습니다.
- 30번째줄의 주석을 해제합니다.
- 38, 39번째 줄의 GyX를 AcX로, 400을 200으로 바꿉니다.

```
30 Serial.print("AcX ="); Serial.println(AcX);
31 // Serial.print("AcY ="); Serial.println(AcY);
32 // Serial.print("AcZ ="); Serial.println(AcZ);
33 // Serial.print("Tmp ="); Serial.println(Tmp / 340.00 + 36.53);
34 // Serial.print("GyX ="); Serial.println(GyX);
35 // Serial.print("GyY ="); Serial.println(GyY);
36 // Serial.print("GyZ ="); Serial.println(GyZ);
37
38 if(AcX < 0) AcX = -AcX;
39 analogWrite(5, AcX / 200);
40 }
```

<그림2-9> MPU6050 가속도 확인

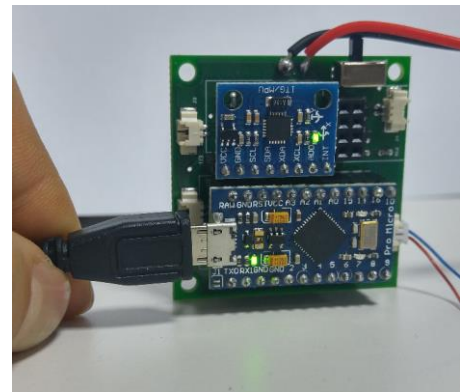
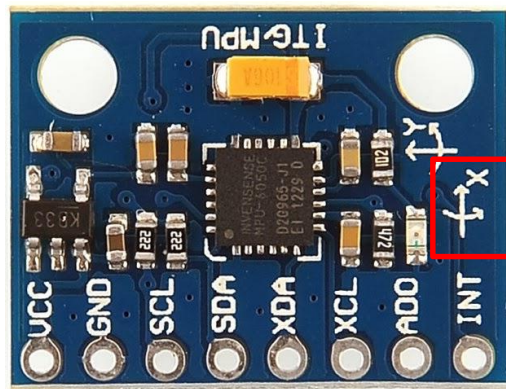
2  버튼을 눌러 시리얼 모니터를 켭니다.

3 보드레이트를 맞춘 후, AcX의 값을 확인합니다.



<그림2-10> 시리얼 통신 확인

- 4 배터리와 모터를 연결하고 스위치를 켭니다.(17페이지 참조)
- 5 자이로 센서의 x화살표가 위를 향하게 하면서 모터 속도의 변화를 확인합니다.



<그림2-11> MPU6050 방향 변화

MPU6050 가속도 해석

```
Serial.print("AcX="); Serial.println(AcX); // 데이터 확인
// Serial.print("AcY="); Serial.println(AcY);
// Serial.print("AcZ="); Serial.println(AcZ);
// Serial.print("Tmp="); Serial.println(Tmp / 340.00 + 36.53);
// Serial.print("GyX="); Serial.println(GyX);
// Serial.print("GyY="); Serial.println(GyY);
// Serial.print("GyZ="); Serial.println(GyZ);
```

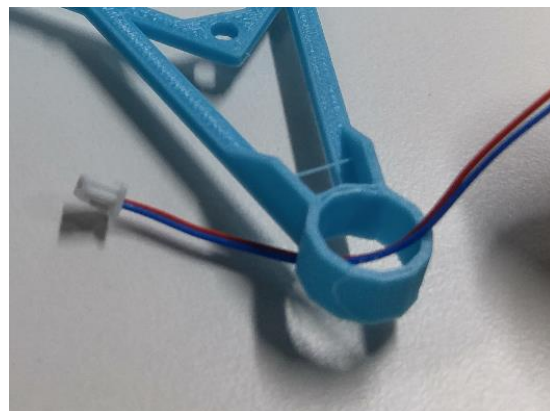
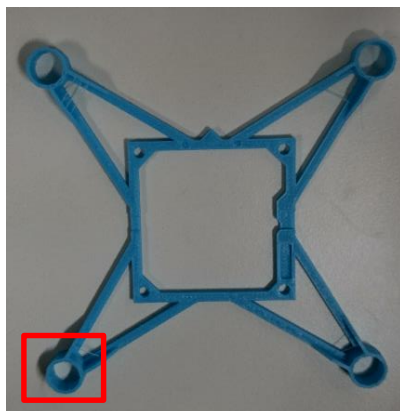
```
if(AcX < 0) AcX = -AcX; //만약 AcX가 0보다 작으면 양수로 전환
analogWrite(5, AcX / 200); // 모터 출력
}
```

AcX는 Accelerometer의 줄임말로 X축으로의 가속도를 의미합니다. AcX의 값은 대략 16000 ~ -16000 사이의 값을 가집니다. x를 화살표 방향 위로 했을 때 최대값을 갖습니다. 이는 x축 방향으로 작용하는 중력가속도의 값을 의미합니다.

모터 연결하기

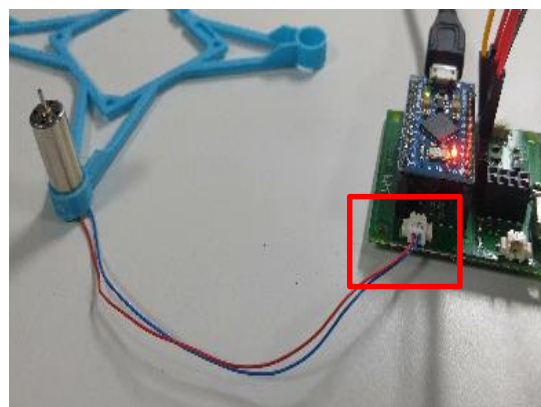
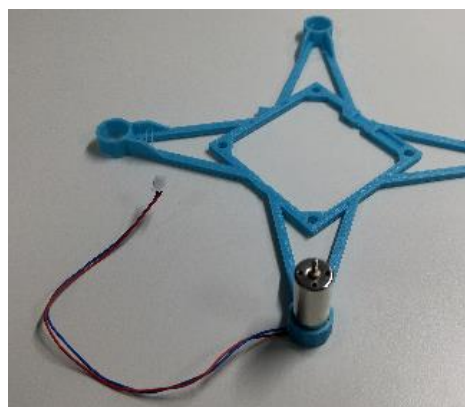
모터 연결 구성

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



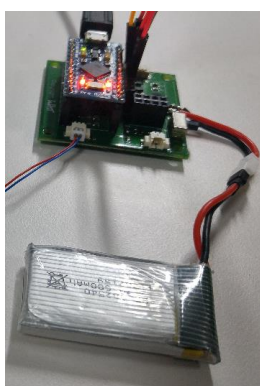
<그림2-12> 모터 연결하기

- 2 베이스 보드의 좌측 하단에 모터 꼬리를 연결합니다.



<그림2-13> 모터 연결하기

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

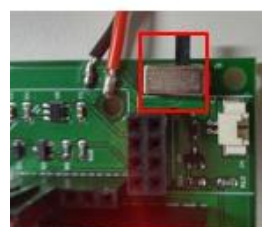
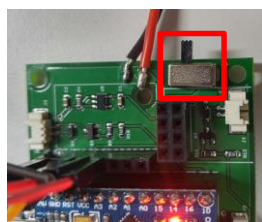


<그림2-14> 모터 연결하기

꿀TIP

베이스보드 스위치

베이스 보드의 스위치는 좌측이 꺼짐, 우측이 켜짐입니다.



03 에어 마우스 만들기



마우스는 보통 평평한 책상 위에 올려놓고 사용합니다.

하지만 아두이노 가속도 센서를 활용하면 공중에서 마우스를 사용할 수 있게 됩니다.

자이로, 가속도 센서를 통해 기울어진 정도를 아두이노가 받아서 기울어진 만큼 아두이노를 움직이게 됩니다.

에어 마우스란?

에어 마우스

에어마우스란 자이로 센서를 이용하여 기존 마우스와는 달리, 공중에서도 쓸 수 있게 만들어진 마우스입니다.



<그림3-1> 에어 마우스

출처 : <https://www.youtube.com/watch?v=0Fbm6O0CxLU>

Mouse
라이브러리

아두이노 우노와는 달리 아두이노 프로 마이크로는 컴퓨터와 직접적인 USB통신을 통해, 아두이노를 마우스 또는 키보드 등으로 인식하게 할 수 있습니다.



<그림3-2> 마우스와 아두이노 프로 마이크로

아두이노 프로 마이크로를 통해 마우스를 만드려면 아두이노를 마우스로 동작하게 할 수 있는 라이브러리를 사용해야 합니다.

```
2 | #include<Mouse.h>
```

에어마우스 코드


에어 마우스 코드

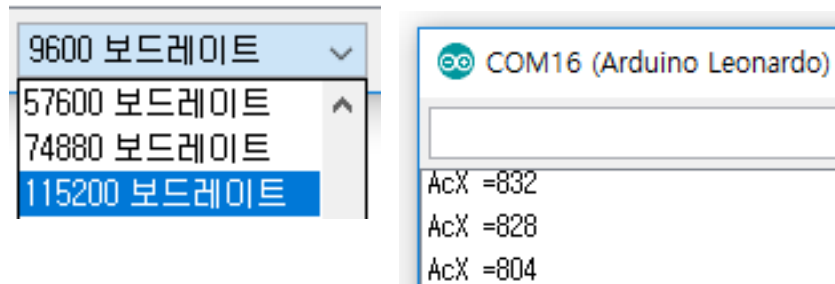
1 다음과 같이 코드를 변형하여 아두이노에 업로드합니다.

```
ch6_1_2_acc
1 #include<Wire.h>
2
3 const int MPU_ADDRESS = 0x68;
4 int16_t AcX, AcY, AcZ, Tmp, GyX,
5
6 void setup() {
7   Wire.begin();
8   Wire.beginTransmission(MPU_ADDRF
9   Wire.write(0x6B);
10  Wire.write(0);
11  Wire.endTransmission(true);
12  Serial.begin(115200);
13  pinMode(5, OUTPUT);
14 }
15
16 void loop() {
17   Wire.beginTransmission(MPU_ADDRF
18   Wire.write(0x3B);
19   Wire.endTransmission(false);
20   Wire.requestFrom(MPU_ADDRESS, 1
21
22   AcX = Wire.read() << 8 | Wire.r
23   AcY = Wire.read() << 8 | Wire.r
24   AcZ = Wire.read() << 8 | Wire.r
25   Tmp = Wire.read() << 8 | Wire.r
26   GyX = Wire.read() << 8 | Wire.r
27   GyY = Wire.read() << 8 | Wire.r
28   GyZ = Wire.read() << 8 | Wire.r
29
30   Serial.print("AcX = "); Serial
31   // Serial.print("AcY = "); Ser
32   // Serial.print("AcZ = "); Ser
33   // Serial.print("Tmp = "); Ser
34   // Serial.print("GyX = "); Ser
35   // Serial.print("GyY = "); Ser
36   // Serial.print("GyZ = "); Ser
37
38   if (AcX < 0) AcX = -AcX;
39   analogWrite(5, AcX / 200);
40 }
```

```
ch6_1_3_air_mouse
1 #include<Wire.h>
2 #include<Mouse.h>
3
4 const int MPU_ADDRESS = 0x68;
5 int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
6
7 void setup() {
8   Wire.begin();
9   Wire.beginTransmission(MPU_ADDRESS);
10  Wire.write(0x6B);
11  Wire.write(0);
12  Wire.endTransmission(true);
13  Serial.begin(115200);
14  pinMode(5, OUTPUT);
15
16  Mouse.begin();
17 }
18
19 void loop() {
20   Wire.beginTransmission(MPU_ADDRESS);
21   Wire.write(0x3B);
22   Wire.endTransmission(false);
23   Wire.requestFrom(MPU_ADDRESS, 14, true);
24
25   AcX = Wire.read() << 8 | Wire.read();
26   AcY = Wire.read() << 8 | Wire.read();
27   AcZ = Wire.read() << 8 | Wire.read();
28   Tmp = Wire.read() << 8 | Wire.read();
29   GyX = Wire.read() << 8 | Wire.read();
30   GyY = Wire.read() << 8 | Wire.read();
31   GyZ = Wire.read() << 8 | Wire.read();
32
33   Serial.print("AcX = "); Serial.println(AcX);
34   // Serial.print("AcY = "); Serial.println(AcY
35   // Serial.print("AcZ = "); Serial.println(AcZ
36   // Serial.print("Tmp = "); Serial.println(Tmp
37   // Serial.print("GyX = "); Serial.println(GyX
38   // Serial.print("GyY = "); Serial.println(GyY
39   // Serial.print("GyZ = "); Serial.println(GyZ
40
41   if (AcX < -8000){
42     Mouse.move(5,0,0);
43   } else if(AcX > 8000){
44     Mouse.move(-5,0,0);
45   }
46 }
```


변형한 부분 설명

- 2  버튼을 눌러 시리얼 모니터를 켵니다.
- 3 보드레이트를 맞춘 후, AcX의 값에 따른 마우스의 움직임을 확인합니다.



<그림3-3> 시리얼 통신 확인

1. Mouse 관련 라이브러리 추가 코드 작성

```
1 | #include<Wire.h> → 1 | #include<Wire.h>
2 |                       2 | #include<Mouse.h>
```

2. Mouse 사용 함수 실행

```
16 | Mouse.begin();
```

3. 만약 AcX값이 -8000보다 작으면 마우스를 왼쪽으로 이동, 만약 AcX값이 8000보다 크면 마우스를 오른쪽으로 이동

```
41 | if (AcX < -8000){
42 |     Mouse.move(5,0,0);
43 | } else if(AcX > 8000){
44 |     Mouse.move(-5,0,0);
45 | }
```

Y좌표 이동

- Q** Mouse.move()의 입력값은 x좌표, y좌표, 마우스 휠 3가지입니다. 위 코드를 바꾸어 마우스가 위 아래로도 움직일 수 있게 만들어봅시다.

```
Mouse.move(0, -5, 0);
```



WHIT