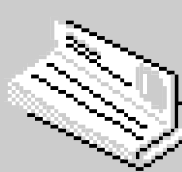
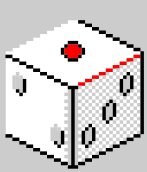


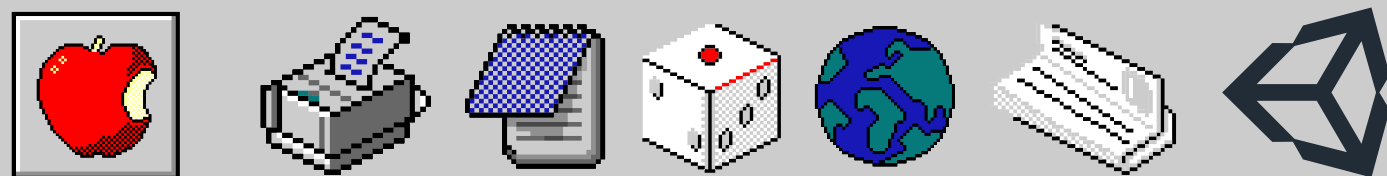
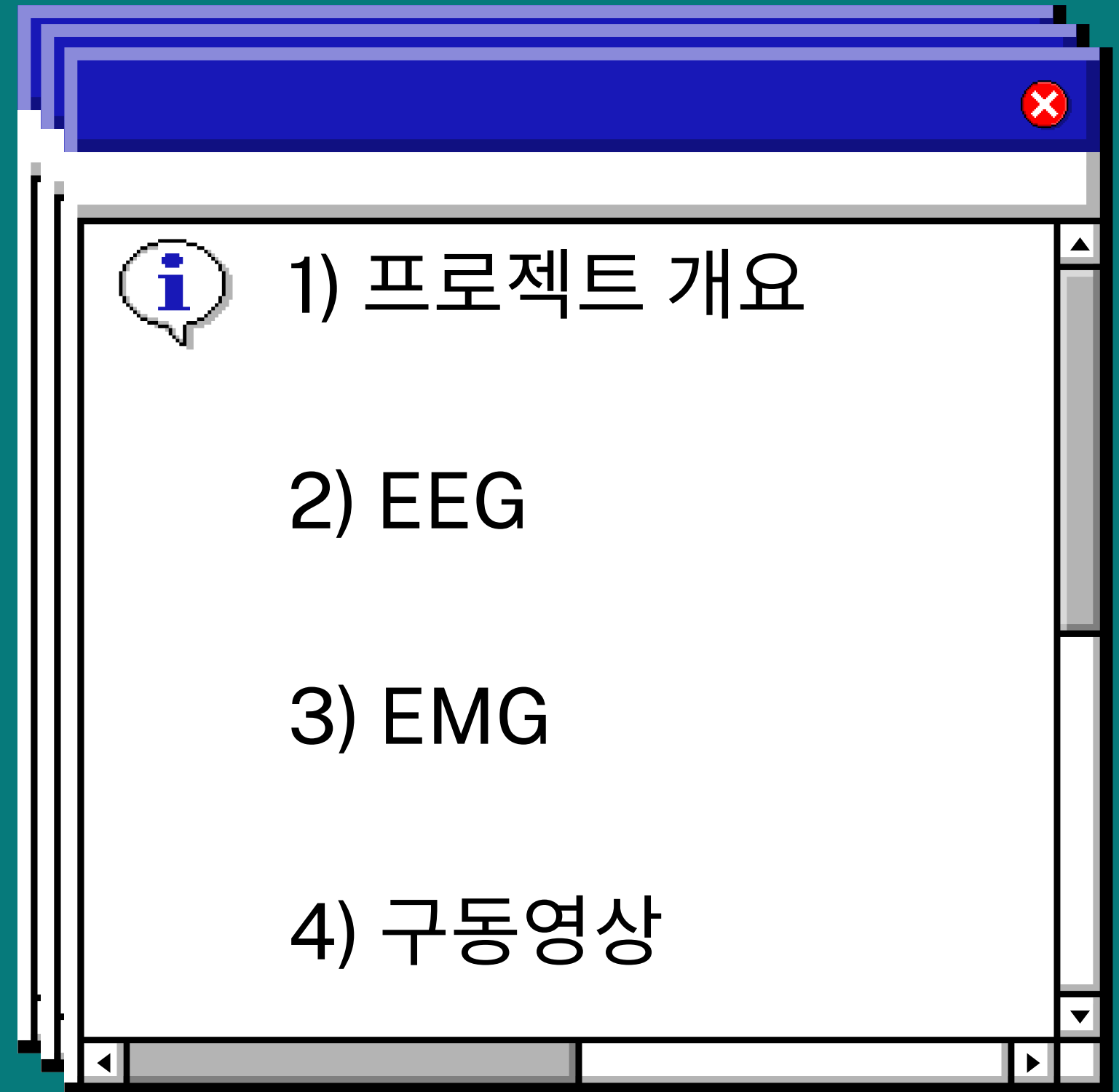
# Controlling virtual object using bio-signals



Team 세얼간이  
신영훈, 이백승, 이성민



11:11PM



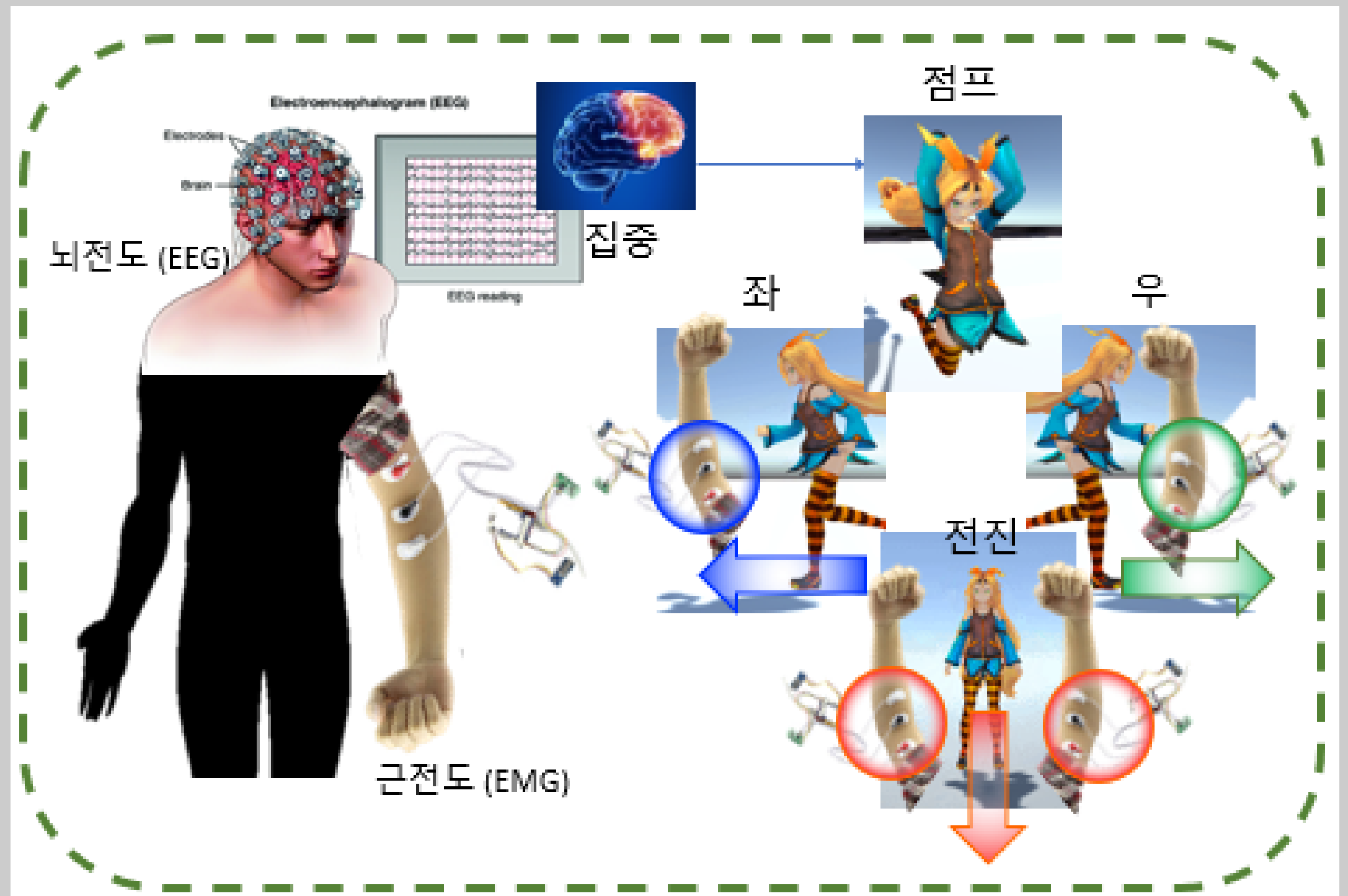
11:11PM

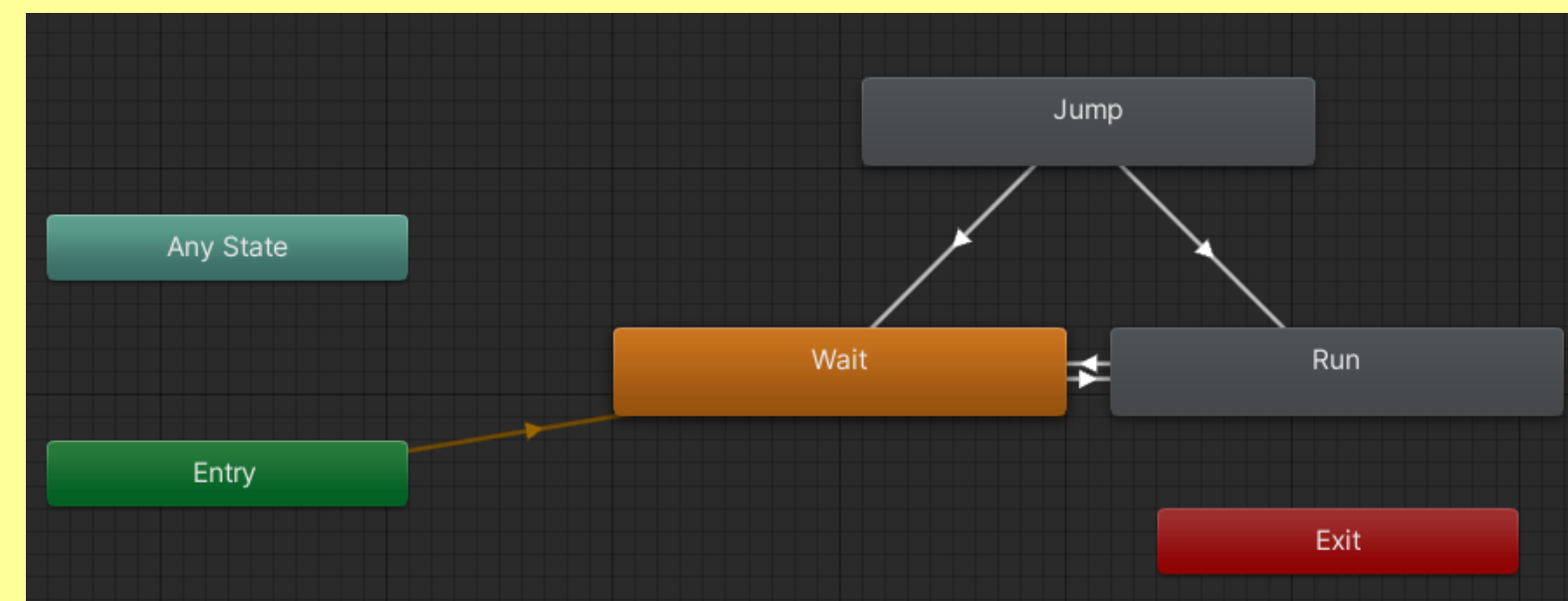
## 프로젝트 개요

# Bio-Signals

신체의 부자유를 가진 사람이 기기를  
조작할 수 있는 방법을 모색하기 위해  
생체신호를 선정

또는 추가적인 입력장치가 될 수 있음





## 캐릭터의 모션 흐름을 정의하는 Animator 설정

Unity JP의 공식 캐릭터인 Unity chan 에셋을 사용

근전도로 캐릭터의 회전, 직진을 명령  
뇌전도로 추가적인 명령 (본 프로젝트에서는 점프)

1. Wait :

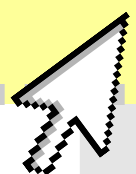
제자리에서 숨을 쉬는 모션 (speed=0, eegAct=0)

2. Run :

달리는 모션 (speed>0, eegAct=0)

3. Jump :

점프하는 모션 (eegAct=1)



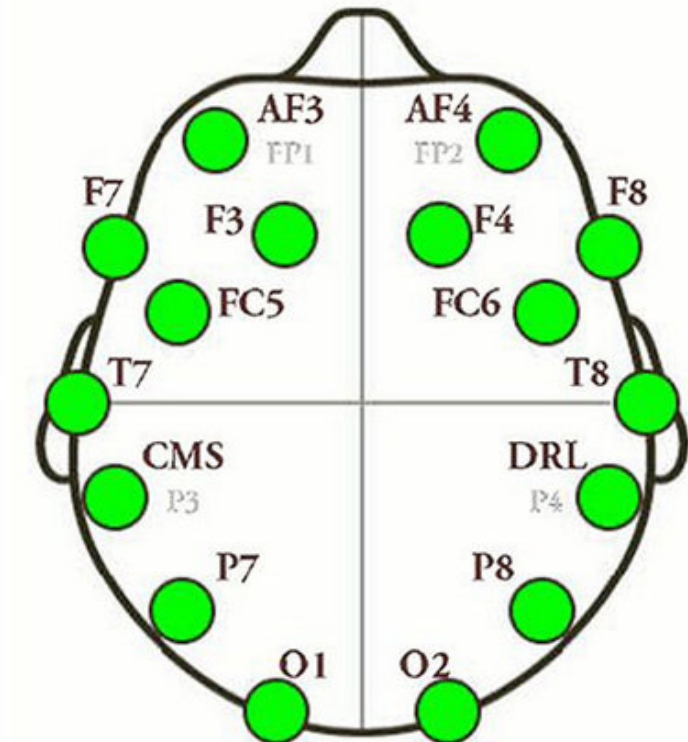
# EEG



본 프로젝트에서는 P7, P8  
두 채널만을 사용함



(a)



(b)

H/W, S/W : 장비는 Emotiv Epoc을 사용, 16개 전극 (2개는 Ref)의 **14채널**인 **무선** 뇌파 측정 장비임. 뇌파 Raw data를 획득해오기 위해 Emotiv사의 S/W 를 사용. 학생 라이선스를 구독하였음.

Frontal 전두엽 8개,  
**Parietal 두정엽 2개** (나머진 Ref)

Temporal 측두엽 2개  
Occipital 후두엽 2개

EEG

Emotiv 장비 -> 수신 USB 동글 -> 호스트 컴퓨터 -> Emotiv Pro(S/W) -> Stream 형성 -> Unity 에서 Stream 수신 ->

오브젝트 제어

## Topics Covered



각 채널은 International 10-20 system 을 따르고 있으며 위 사진과 같은 전극들이 한 쌍을 이룸

Stream 에서 Raw EEG 데이터의 데이터포맷은 다음과 같음.

EEG	{"Timestamp", "Counter", "Interpolate", , "HardwareMarker", "Markers" }	128Hz/ 256 Hz
-----	---	------------------

Emotiv Pro 내에서는 이 raw data에 FFT (Fast Fourier transform, 고속 푸리에 변환) 이 적용된 데이터인 'Band-Power'를 함께 출력할 수 있음



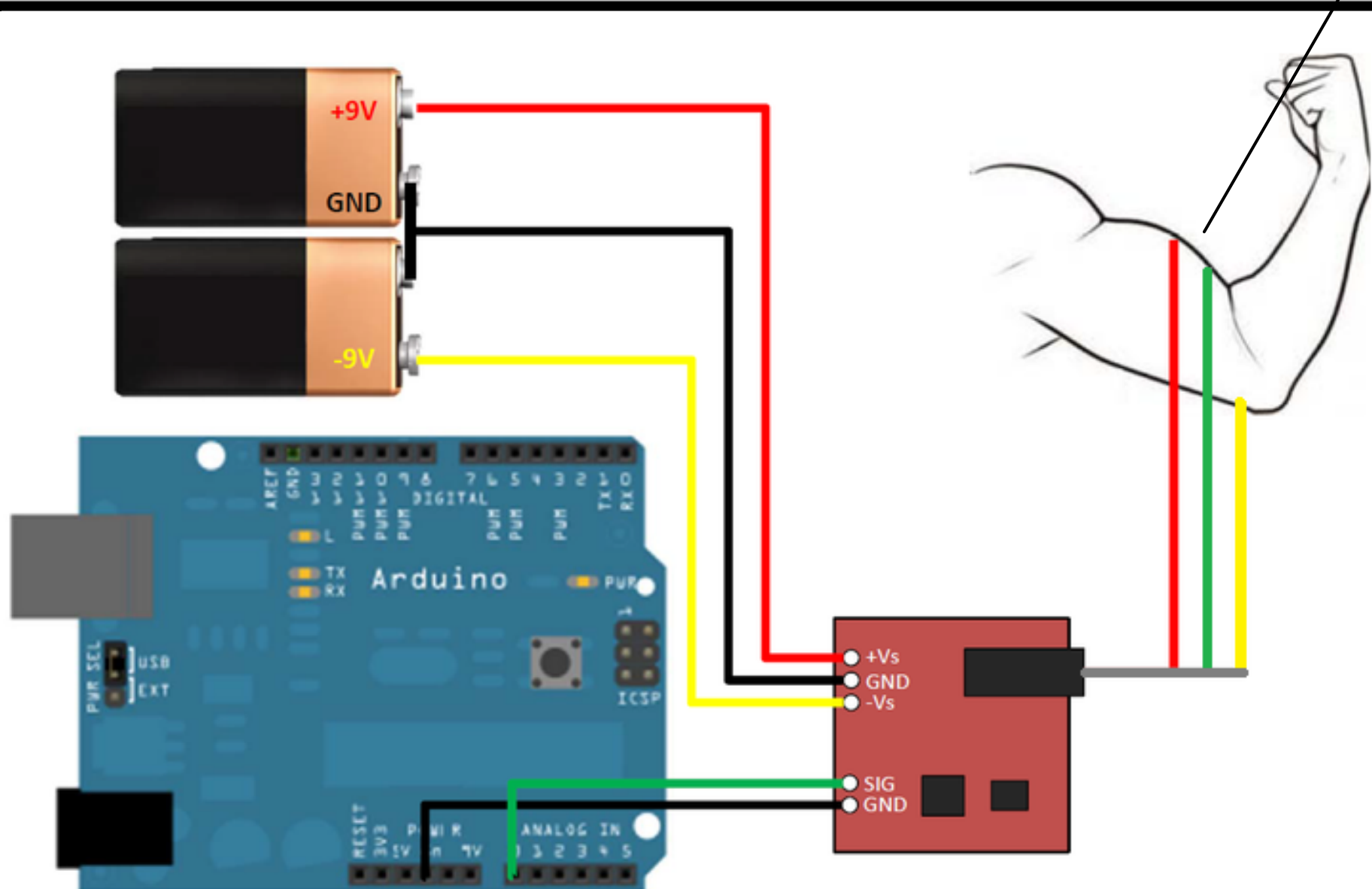
비침습형 & 무선 장비이므로  
심한 노이즈가 있음.  
FFT 알고리즘은 노이즈를 줄이기 위해 쓰였다.

Band-Power	{"Timestamp", <EEG sensors/Band>}	8 Hz
------------	-----------------------------------	------

'Band-Power'의 스트림은 71개 채널의 데이터를 가짐.  
TimeStapm + 14개 채널 \* ( 5개 주파수 대역  
[theta/alpha/betaL/betaH/gamma] )

=> P7 betaL [28]와 P8 betaL [43] 만을 선택해 사용함.

# EMG



3개의 전극을 이용.  
빨간선과 초록선은 같은 근육에 위치시  
키고 노란선은 근육이 없는 위치에 부  
착시켜 측정.

## EMG 목표 - 오브젝트의 이동

- 2개의 센서를 사용하여
- 1) 움직이지 않는 상태
  - 2) 몸을 오른쪽으로 돌리기
  - 3) 몸을 왼쪽으로 돌리기
  - 4) 앞으로 움직이기
- 총 4개의 상태를 만들어낸다.



```

for(int d = 0; d < 50; d++){
    val1 = 0;
    sum1 = 0;
    for(int a = 0; a < 10; a++){
        val1 = analogRead(A0);
        sum1 = sum1 + val1;
    }
    float avel = sum1 * 0.1;
    if (avel > 90){
        count1++;
    }
}
int data1 = 0;
if (count1 >= 20){ data1 = 1; }
else{ data1 = 0; }

```

보다 정밀한 제어를 위해 오차를 줄이는 방법으로 10개의 데이터를 평균 내어 총 50개를 만든 후 임계값을 넘으면 반응하게 설정.

A0, A1핀에 근전도 센서를 연결

A0에서는 0과 1의 값을

A1에서는 0과 2의 값을

출력하도록 설정하여

0, 1, 2, 3 총 4개의 신호가 만들어진다.

=> 1, 2의 값은 오브젝트의 Rotation을 움직이고, 3의 값은 sin, cos을 이용하여 360° 자유롭게 움직인다.

