



API 사용메뉴얼

How to Contact Kitronyx

Latest news: www.kitronyx.com

Technical support: <https://www.kitronyx.com/contact.html>

Phone: +82-2-564-0441

Address: #1201, 46, Gasan digitalro9gil, Geumcheon-gu, Seoul, Korea 08512


© COPYRIGHT 2014-2024 by Kitronyx, Inc.

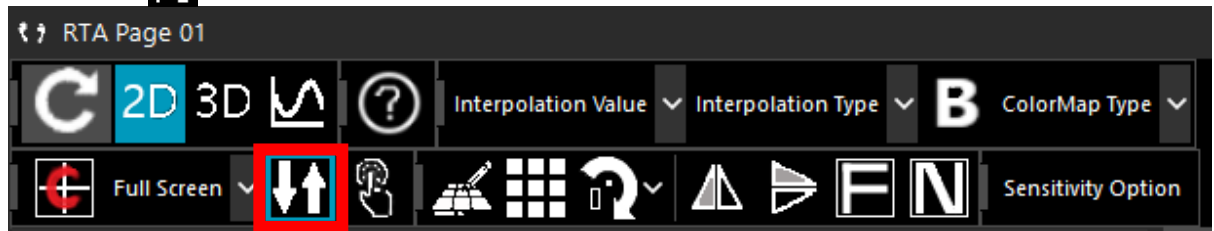
The software described in this document is furnished under license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from Kitronyx, Inc.

목차

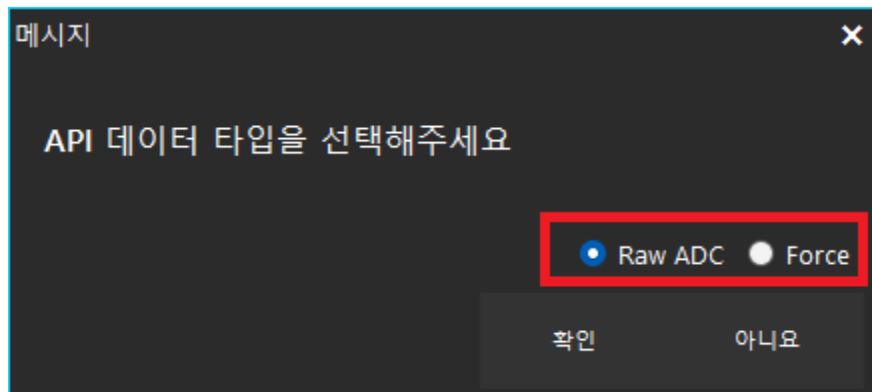
ForceLAB2 API 사용법	3
SnowForce3 API 사용법	5
API C# 사용법	6
API C++ OpenFreamWorks 사용법	8
API C++ ConsoleProject 사용법	9
API Python 사용법	10
API Processing 사용법	11

ForceLAB2 API 사용법

1, API 기능  클릭하여 기능 ON/OFF



2. 데이터 타입 설정



3. 코드 주소

- C++ 기반 라이브러리 : <https://github.com/kitronyx/klib2-cpp>
- C# 기반 라이브러리 : <https://github.com/kitronyx/klib2-csharp>
- Python 기반 라이브러리 : <https://github.com/kitronyx/klib2-python>
- Java 기반 라이브러리 : <https://github.com/kitronyx/klib2-processing>

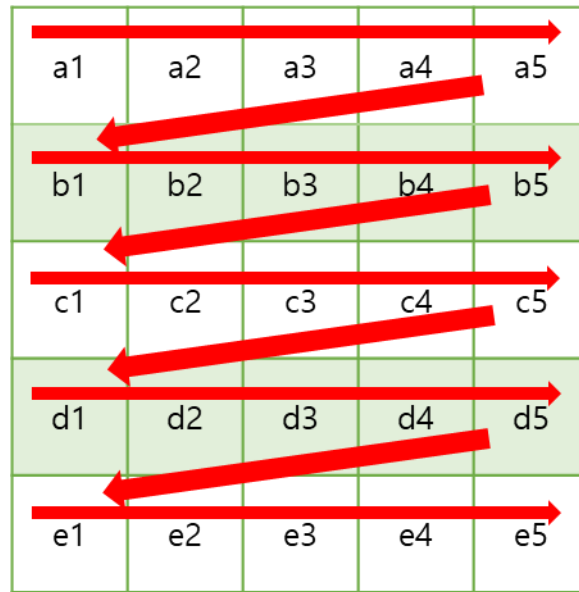
※ 사용 시, 주의사항

- Force 타입 통신 시, 캘리브레이션 데이터를 삭제하면 데이터가 제대로 송수신되지 않을 수 있습니다.

해당 예제 코드는 라이브러리를 사용하여 데이터를 수신하는 응용프로그램 예제입니다.

- Klib 초기화는 IP(보통 127.0.0.1), Port(3800)를 인자로 넣어주어야 한다.
- Start() ForceLAB2 와 통신을 연결한다.
- Read() 패킷을 받아서 멤버변수 adc 에 저장한다.
- 이때 데이터는 ForceLAB2 RTA 화면상 x 축 왼쪽부터 오른쪽으로 데이터를 전달아 패킷으로 보낸다.
-

- Ex) 패킷: a1,a2,a3,a4,a5,b1,b2,b3,b4,b5,c1,...,e5



- Stop() ForceLAB2 와 통신을 끊는다.

```
if __name__ == "__main__":
    klib = KLib("127.0.0.1", 3800)

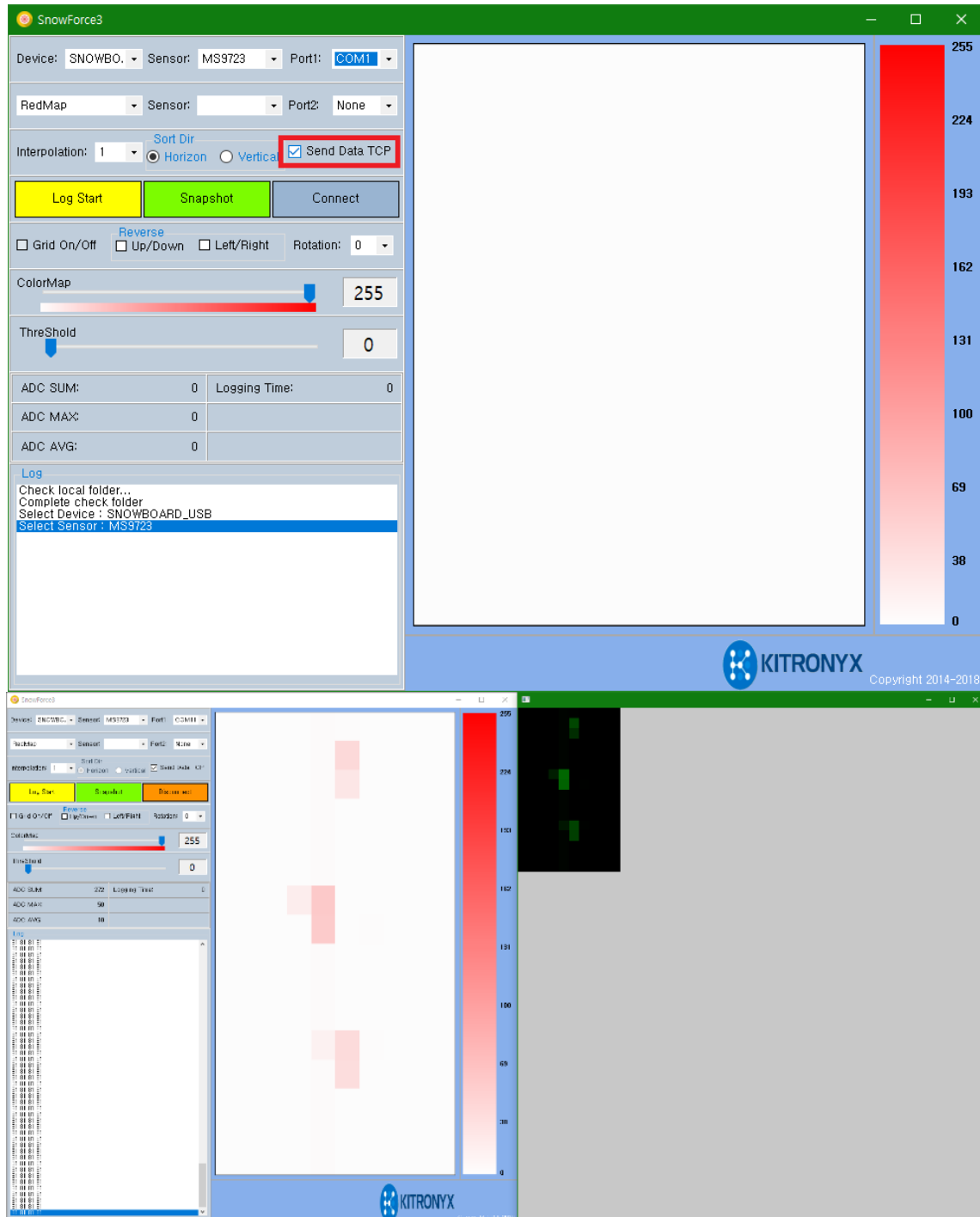
    klib.start()
    while(1):
        klib.read()
        klib.printadc()
```

[Python 기반 예제 코드]

SnowForce3 API 사용법

- SnowForce3 Send Data TCP 체크박스를 클릭하여 체크 후 Connect 버튼을 클릭하여 디바이스와 연결한다.
- Snowforce3 의 경우 Raw ADC 타입 데이터만 전달한다.

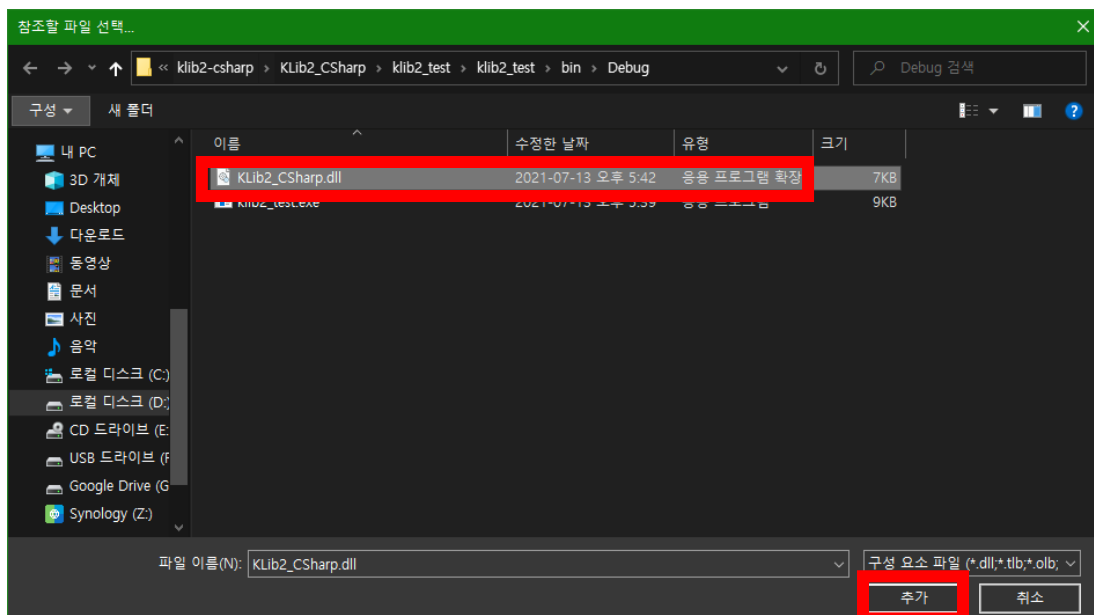
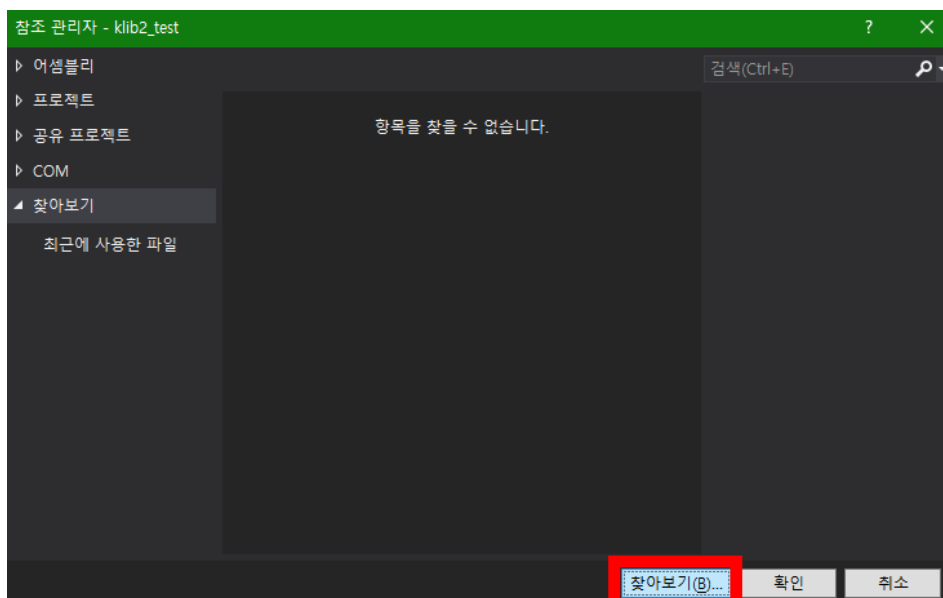
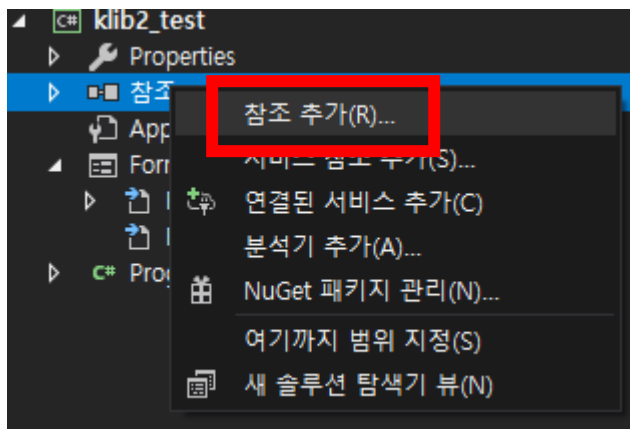
※ 반드시 디바이스와의 연결을 끊고 체크박스에 체크 후 다시 연결 시 정상 작동됨

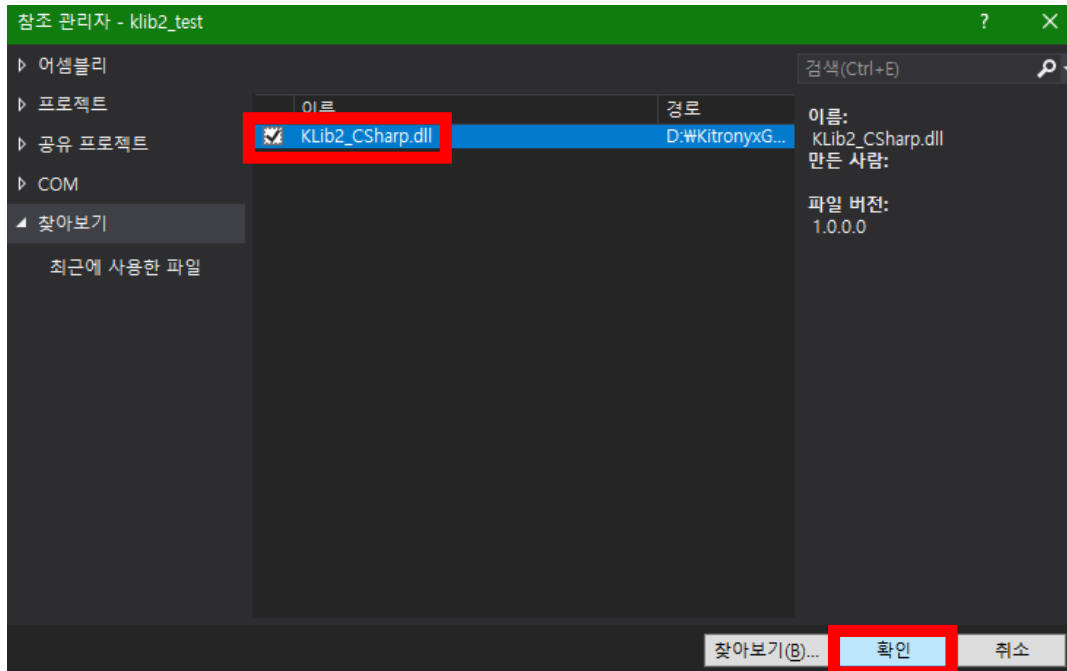


[API 작동 예시]

API C# 사용법

- 1) KLib2_CSharp 라이브러리 파일을 참조로 추가한다.





- 2) KLib2_CSharp 사용 편의를 위한 using문을 사용하며 KLib2를 초기화한다.
이때, 매개변수로 IP 주소를 설정할 수 있다

```
using KLib2_CSharp;

namespace klib2_test
{
    참조 3개
    public partial class MainForm : Form
    {
        // IP 주소를 입력하여 변경 가능하게 구현
        KLib2 klib = new KLib2("127.0.0.1");
    }
}
```

- 3) Start()를 호출하면 ForceLAB2/SnowForce3와는 TCP IP로 연결된다.

```
public bool Start();
```

- 4) Read()를 호출하면 ForceLAB2/SnowForce3의 패킷을 읽으며 byte[] 형식으로 반환한다.

```
//get API data
byte[] data = klib.Read();
```

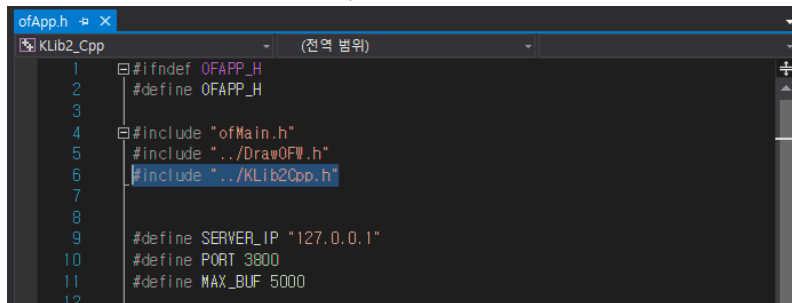
- 5) Stop()을 호출하면 ForceLAB2/SnowForce3와 연결 해제된다.

```
public bool Stop();
```


API C++ OpenFrameWorks 사용법

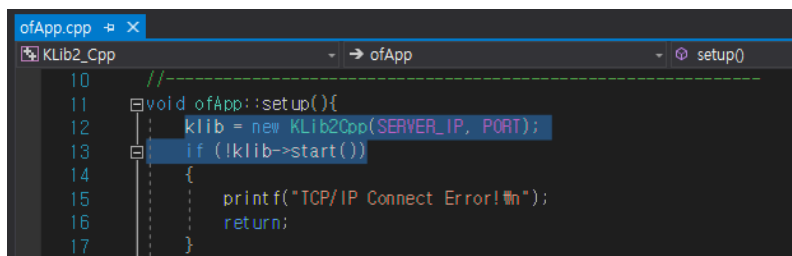
- ❖ OpenFrameWorks 프로젝트는 C 드라이브 바로 아래 위치하는 것을 권장
- ❖ OpenFrameWorks 프로젝트를 추가 시 OpenFrameWorks 내에 있는 apps 폴더 아래에 생성 권장
- ❖ 예제 코드 경로: of_v0.10.0_vs2017_release\apps\KLib2\KLib2

- 1) KLib2 라이브러리 파일을 OpenFrameWorks에 include한다.



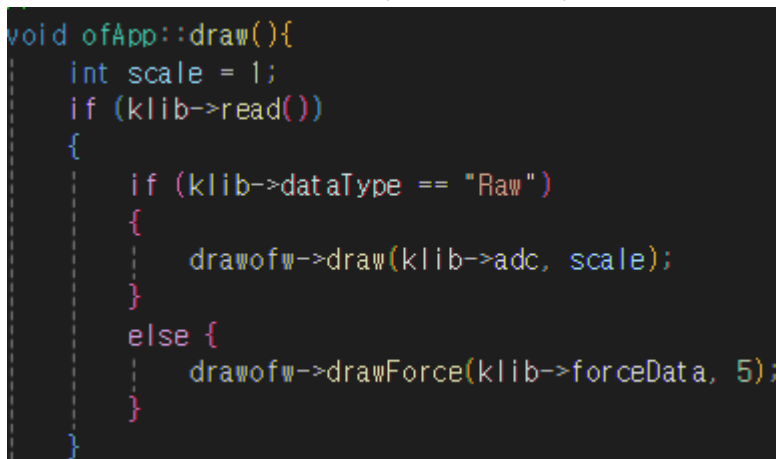
```
1  #ifndef OFAPP_H
2  #define OFAPP_H
3
4  #include "ofMain.h"
5  #include "../DrawFW.h"
6  #include "../KLib2Ccp.h"
7
8
9  #define SERVER_IP "127.0.0.1"
10 #define PORT 3800
11 #define MAX_BUF 5000
12
```

- 2) 초기화하면서 서버주소와 포트를 입력한다. start()를 호출하면 ForceLAB2/SnowForce3와 TCP로 연결된다.



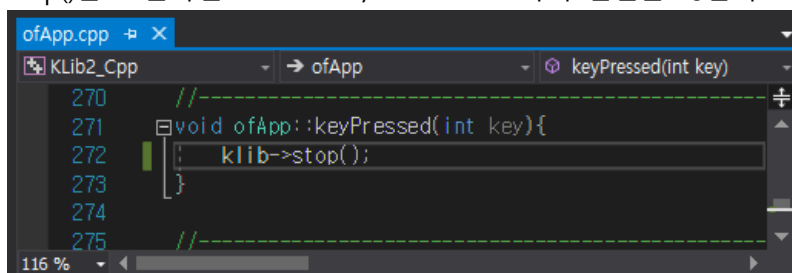
```
10 //-----
11 void ofApp::setup(){
12     klib = new KLib2Ccp(SERVER_IP, PORT);
13     if (!klib->start())
14     {
15         printf("TCP/IP Connect Error!\n");
16         return;
17     }
18 }
```

- 3) read()를 호출하면 ForceLAB2/SnowForce3의 패킷을 읽으며 읽은 데이터는 데이터 타입에 따라 각각의 멤버 변수(adc, forceData)에 저장된다.



```
void ofApp::draw(){
    int scale = 1;
    if (klib->read())
    {
        if (klib->dataType == "Raw")
        {
            drawofw->draw(klib->adc, scale);
        }
        else {
            drawofw->drawForce(klib->forceData, 5);
        }
    }
}
```

- 4) stop()를 호출하면 ForceLAB2/SnowForce3와의 연결을 끊는다.



```
270 //-----
271 void ofApp::keyPressed(int key){
272     klib->stop();
273 }
274
275 //-----
```

API C++ ConsoleProject 사용법

❖ 예제 코드 경로: KLib2CPP_ConsoleProject

- 1) main.cpp에 "KLib2Cpp.h"를 include한다.

```
#include "KLib2Cpp.h"
```

- 2) KLib2Cpp 객체를 생성한 후 start 함수를 호출하여 TCP/IP 통신 연결과 데이터를 초기화한 후 ForceLAB2/SnowForce3와 연결한다.

```
KLib2Cpp* klib0bject = new KLib2Cpp();  
  
klib0bject->start();
```

- 3) 데이터 타입 종류에 따라 adc, force 데이터를 출력한다.

```
while (1) {  
    if (klib0bject->dataType == "Raw") {  
        int** _adc = klib0bject->read();  
  
        klib0bject->printadc();  
  
        printf("\n");  
        //Sleep(1);  
        system("cls");  
  
        for (int i = 0; i < klib0bject->row; ++i)  
        {  
            delete[] _adc[i];  
        }  
        delete[] _adc;  
    }  
    else {  
        double** forceData = klib0bject->forceRead();  
  
        klib0bject->printForceData();  
  
        printf("\n");  
        //Sleep(1);  
        system("cls");  
  
        for (int i = 0; i < klib0bject->row; ++i)  
        {  
            delete[] forceData[i];  
        }  
        delete[] forceData;  
    }  
}
```

API Python 사용법

```
class KLib(): ...

if __name__ == "__main__":
    klib = KLib("127.0.0.1", 3800)
    tick = 0
    FPS = 0
    prevTime = time.time()

    klib.start()
    while(1):
        klib.read()
        klib.printData()
```

❖ KLib 클래스가 같은 .py 파일에 있는 것을 전제로 설명

1) KLib 객체를 생성한다.

IP(127.0.0.1 or 서버 PC의 IP)와 포트(3800)로 설정한다.

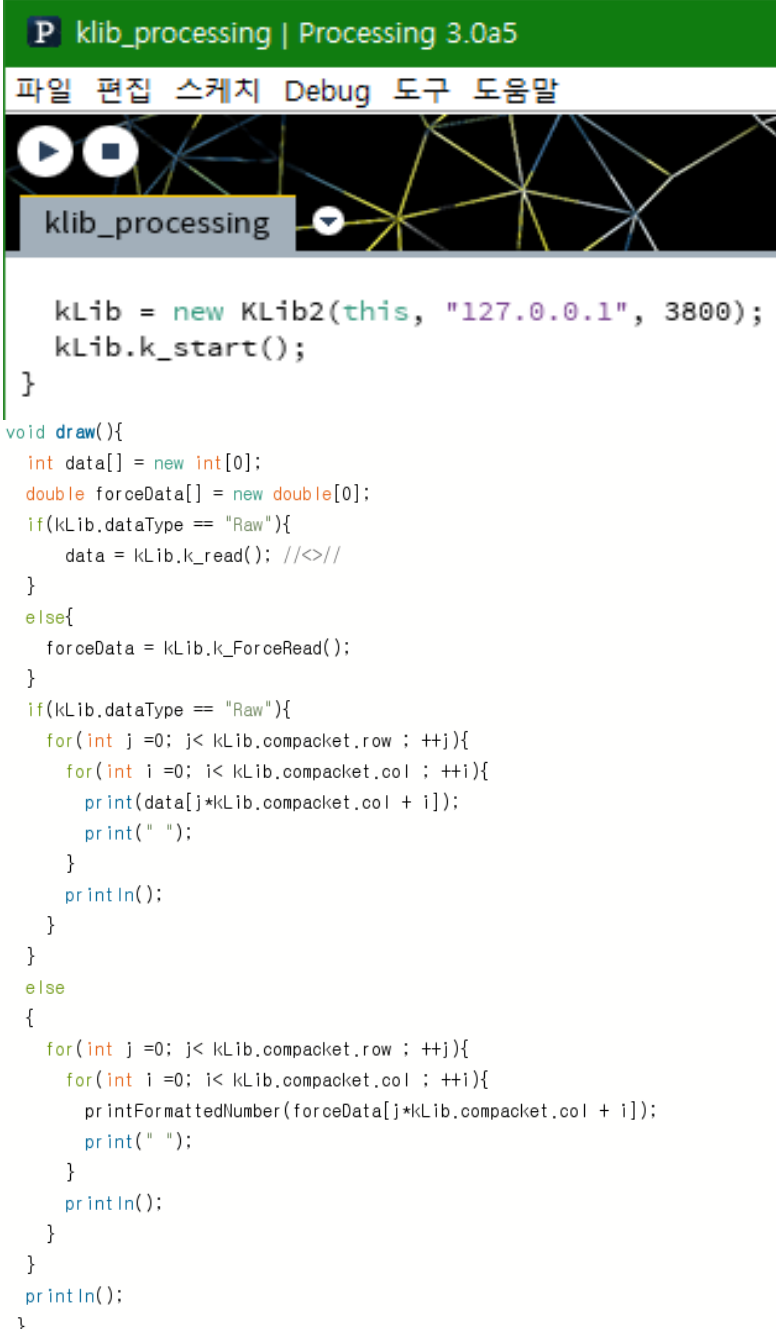
2) start() 호출하여 ForceLAB2/SnowForce3와 연결한다.

3) read() 호출하여 ForceLAB2/SnowForce3가 보내는 패킷을 받는다.

4) printData() 호출하여 받은 패킷을 데이터 타입에 맞춰 화면에 표시한다.

5) stop() 호출하여 ForceLAB2/SnowForce3와 연결을 끊는다.

API Processing 사용법



```
klib_processing | Processing 3.0a5
파일 편집 스케치 Debug 도구 도움말

klib_processing

kLib = new KLib2(this, "127.0.0.1", 3800);
kLib.k_start();
}

void draw(){
  int data[] = new int[0];
  double forceData[] = new double[0];
  if(kLib.dataType == "Raw"){
    data = kLib.k_read(); //<>
  }
  else{
    forceData = kLib.k_ForceRead();
  }
  if(kLib.dataType == "Raw"){
    for(int j =0; j< kLib.compaket.row ; ++j){
      for(int i =0; i< kLib.compaket.col ; ++i){
        print(data[j*kLib.compaket.col + i]);
        print(" ");
      }
      println();
    }
  }
  else
  {
    for(int j =0; j< kLib.compaket.row ; ++j){
      for(int i =0; i< kLib.compaket.col ; ++i){
        printFormattedNumber(forceData[j*kLib.compaket.col + i]);
        print(" ");
      }
      println();
    }
  }
  println();
}
```

- KLib 객체를 생성할 때 PApplet(특별한 경우가 아닐 경우 this라 쓰면 된다.), 서버IP(127.0.0.1 or localhost)와 포트(3800)를 설정한다
- K_start()를 호출하여 ForceLAB2/SnowForce3와 연결한다.
- Processing의 draw()를 통하여 데이터를 시각화합니다.
데이터 타입에 따라 k_read(), k_ForceRead()를 호출하여 데이터 패킷을 읽은 후 데이터 타입에 따라 시각화합니다.