양방향 API 통신 매뉴얼

㈜카이트로닉스

문서 이력

작성일	내용	ForceLAB2 버전
23.06.23	초안 작성	V2.4.1
24.01.05	Github 링크 추가 및 Copyright 연도 수정	V2.7.1

기본 필수 파일

- KLib3.dll
  - C++ x64기반으로 개발
  - ForceLAB2 v2.5.0 Beta 버전부터 호환
  - TCP/IP을 이용하여 자사 SW ForceLAB2와 통신(방화벽 허용 필수)
  - ForceLAB2 내부에서 상시 대기 중이므로 별도 설정X
- 개발 관련 다운로드 링크
  - Github: https://github.com/kitronyx/KLib3

지령(Command)

- 지령 구조

앞 1byte 중 4bit는 지령 타입(TypeCommand) 뒤 2byte는 수행 지령 목록(Work Command)

- 기본 수행 지령 목록(Work Command)
  - Version – 0xF00000 : ForceLAB2 버전 확인
  - Update – 0xF00001 : 양방향 API 지령을 ForceLAB2로부터 업데이트
- 지령 타입 목록(Type Command)
  - Request – 0x000000
  - Complete – 0x010000
  - Loop – 0x020000
  - Repeat – 0x080000

전체 Extern 함수

- void* CreateApiClient(void)
  - 생성자 초기화 함수
- void DisposeApiClient(void* DllHandler)
  - 소멸자 함수
- bool ApiClient_Open(void* DllHandler)
  - 양방향 API 시작 함수로 서버와의 통신 접속 수행
- bool ApiClient_Close(void* DllHandler)
  - 양방향 API 종료 함수로 서버와의 통신 종료 수행
- void ApiClient_GetReceiveStackCommand(void* DllHandler, uint& CommandCode, char*& CommandData, int& CommandDataLen)
  - 쌓여 있는 수신 지령 중 가장 먼저 들어온 지령을 내보내는 함수
- void ApiClient_SendCommandByCode(void* DllHandler, uint CommandCode, char* CommandData,int CommandDataLen)
  - 지령코드(WorkCommandCode)를 Integer 값으로 지령을 송신하는 함수
- void ApiClient_SendCommandByStr(void* DllHandler, char* CommandStr, int CommandStrLen, char* CommandData, int CommandDataLen)
  - 지령문자열(WorkCommandStr)를 char*로 지령을 송신하는 함수
- void ApiClient_GetCommandList(void* DllHandler, char*& WorkCommandStr, int& WorkCommandStrLen, char*& TypeCommandStr, int& TypeCommandStrLen)
  - 수행 지령과 지령 타입의 문자열 및 해당 지령 코드를 출력하는 함수
- void ApiClient_GetTimeOut(void* DllHandler, int& ConnectTimeout_ms, int& SendTimeout_ms)
  - 설정된 통신 만료 시간(Timeout) 값을 출력 받는 함수
- Void ApiClient_SetTimeOut(void* DllHandler, int ConnectTimeout_ms, int SendTimeout_ms)
  - 통신 만료 시간(Timeout) 값을 설정하는 함수

기본 사용 수도코드(pseudocode)

```
var kLib3Dll = DLL_Load("KLib3.dll");
void* kLib3ApiHandler = kLib3Dll .CreateApiClient();

kLib3Dll.ApiClient_SetTimeOut(kLib3ApiHandler,ConnectTiemOutmsValue,SendTimeOutmsValue)
kLib3Dll.ApiClient_GetTimeOut(kLib3ApiHandler,ConnectTiemOutmsValue,SendTimeOutmsValue)

kLib3Dll. ApiClient_Open(kLib3ApiHandler);

kLib3Dll.ApiClient_GetCommandList(kLib3ApiHandler,WorkCommandList, WorkCommandListStrLen,
TypeCommandList, TypeCommandListStrLen);

kLib3Dll.ApiClient_GetStackCommand(kLib3ApiHandler, CommandCode, CommandDataStr, CommandDataStrLen);

kLib3Dll.ApiClient_SendCommandByStr(kLib3ApiHandler, CommandStr, CommandDataStr, CommandDataStrLen);

kLib3Dll.ApiClient_SendCommandByCode(kLib3ApiHandler, CommandCode, CommandDataStr, CommandDataStrLen);

kLib3Dll.ApiClient_Close(kLib3ApiHandler);
```

Create()

- 함수 원형

  void* CreateApiClient(void)

- 생성자 초기화 함수
- 호출 시 API DLL의 Handler 반환
  (C++은 void*, C#은 IntPtr, Python은 ctypes.c_void_p)

Dispose()

- 함수 원형

  void DisposeApiClient(void* DllClientHandler)

- 소멸자 함수
- API DLL의 Handler를 인자로 받음
- ApiClient_Close()에서 자동으로 호출

Open()

- 함수 원형

  bool ApiClient_Open(void* DllHandler)

- API 통신 시작 함수
- TCP/IP로 ForceLAB2와 연결 수행
- API DLL의 Handler를 인자로 받음
- 연결 성공 여부를 Boolean으로 받음
- 설정된 Timeout만큼 연결 시도(연결 Timeout 초기값은 5초)
- 연결 실패 시 전부 초기화하여 다시 호출하여 연결 시도가 필요함

Close()

- 함수 원형

  bool ApiClient_Close(void* DllHandler)

- API 통신 종료 함수
- TCP/IP로 연결된 ForceLAB2와 연결 종료
- API DLL의 Handler를 인자로 받음
- 정상 연결 종료 여부 Boolean으로 반환

GetReceiveStackCommand()

- 함수 원형

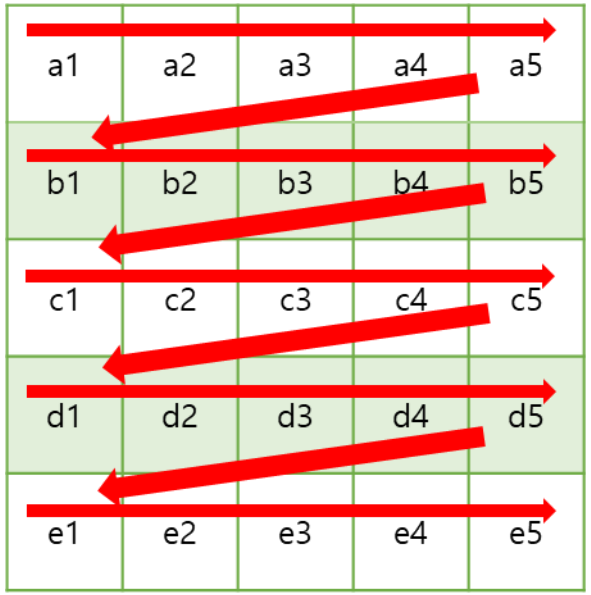
  void ApiClient_GetReceiveStackCommand(void* DllHandler, uint& CommandCode, char*& CommandStrData, int& CommandStrDataLen)

- 수신 받은 지령(Command)들 중 가장 먼저 쌓인 지령을 출력
- 인자 목록
  - DllHandler: API DLL의 Handler 대입
  - CommandCode: 출력된 지령 코드 unsigned integer로 반환
  - CommandStrData: 출력된 지령 내용 char*로 반환
    단, MatrixData 받을 시 Byte[]로 재 변환 필요(데이터를 0~255 사용)
  - CommandStrDataLen: 출력된 지령 내용 문자열 길이를 integer로 반환

수신 Matrix 데이터 순서

데이터는 ForceLAB2 RTA화면상 x축 왼쪽부터 오른쪽으로 데이터를 연달아 패킷으로 보낸다.

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



SendCommandByCode()

- 함수 원형

  void ApiClient_SendCommandByCode(void* DllHandler, uint CommandCode, char* CommandDataStr,int CommandDataStrLen)

- 지령을 ForceLAB2로 보내는 함수
- 인자 목록
  - DllHandler: API DLL의 Handler 대입
  - CommandCode: 보낼 지령 코드를 unsigned integer로 대입
  - CommandDataStr: 보낼 지령 내용을 문자열 16bit Unicode 로 대입
  - CommandDataStrLen: 보낼 지령 내용 문자열 길이를 integer로 대입

SendCommandByStr()

- 함수 원형

  void ApiClient_SendCommandByStr(void* DllHandler, char* CommandStr, int CommandStrLen, char* CommandData, int CommandDataLen)

- 지령을 ForceLAB2로 보내는 함수
- 인자 목록
  - DllHandler: API DLL의 Handler 대입
  - CommandCode: 보낼 지령 문자열 8bit char*로 대입
  - CommandDataStr: 보낼 지령 내용을 문자열 16bit Unicode 로 대입
  - CommandDataStrLen: 보낼 지령 내용 문자열 길이를 integer로 대입

GetCommandList()

- 함수 원형

  void ApiClient_GetCommandList(void* DllHandler, char*& WorkCommandStr, int& WorkCommandStrLen, char*& TypeCommandStr, int& TypeCommandStrLen)

- 수행 지령과 지령 타입의 문자열 및 해당 지령코드를 출력하는 함수
- 인자 목록
  - DllHandler: API DLL의 Handler 대입
  - WorkCommandStr: 수행 지령 목록을 문자열 char*로 반환
  - WorkCoomandStrLen: 수행 지령 목록 문자열 길이 Integer로 반환
  - TypeCommandStr: 지령 타입 목록을 문자열로 char*로 반환
  - TypeCommandStrLen: 지령 타입 목록 문자열 길이 Integer로 반환

GetTimeOut()

- 함수 원형

  void ApiClient_GetTimeOut(void* DllHandler, int& ConnectTimeout_ms, int& SendTimeout_ms)

- 설정된 연결(connect)및 송신(send) 타임아웃 값을 가져오는 함수
- 인자 목록
  - DllHandler: API DLL의 Handler 대입
  - ConnectTimeout_ms: 설정된 연결 타임 아웃 값을 밀리초(ms)로 Integer 값 반환
  - SendTimeout_ms: 설정된 송신 타임 아웃 값을 밀리초(ms)로 Integer 값 반환

SetTimeOut()

- 함수 원형

  void ApiClient_SetTimeOut(void* DllHandler, int ConnectTimeout_ms, int SendTimeout_ms)

- 연결(connect)및 송신(send) 타임아웃 값을 설정하는 함수
- 인자 목록
  - DllHandler: API DLL의 Handler 대입
  - ConnecTimeout_ms: 설정할 연결 타임 아웃 값을 밀리초(ms)로 Integer 값 대입
  - SendTimeout_ms: 설정할 송신 타임 아웃 값을 밀리초(ms)로 Integer 값 대입

C++ Extern 함수 목록 예시

```cpp
#include <iostream>
#include <Windows.h>
#include <string>
#include <thread>
#include <locale>
#include <codecvt>

using namespace std;

typedef void* (*CreateApiClient)(); //생성자
typedef void (*DisposeApiClient)(void*); //소멸자
typedef bool (*ApiClient_Open)(void*); //양방향 API 시작
typedef bool (*ApiClient_Close)(void*); //양방향 API 끝
typedef void (*ApiClient_GetReceiveStackCommand)(void*, int&, char*&, int&); //수신 지령 받기
typedef void (*ApiClient_SendCommandByCode)(void*, int, const char*, int); //지령을 Code(Integer)로, 지령 내용을 문자열로 보내기
typedef void (*ApiClient_SendCommandByStr)(void*, const char*, int, const char*, int); //지령 및 지령 내용을 문자열로 보내기
typedef void (*ApiClient_GetCommandList)(void*, char*&, int&, char*&, int&); //지령 리스트 얻기
typedef void (*ApiClient_GetTimeOut)(void*, int&, int&);
typedef void (*ApiClient_SetTimeOut)(void*, int, int);
```

C++ DLL 사용법

- Extern 함수 추가 시, 아래와 같은 형식으로 함수 정의

  typedef {반환타입} (*{함수명})({인자 목록})

- Extern 함수명 및 인자 형식, 반환 형식은 일치 필수
- SendCommandByStr 또는 SendCommandByCode에 지령 내용을 대입할 때 char16_t*로 변환 후 char*로 대입 (1개 단일 문자당 2개 byte 사용)

C# Extern 함수 목록 예시

```
//초기화
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern IntPtr CreateApiClient();

//양방향 통신 시작
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern bool ApiClient_Open(IntPtr _apiClientPtr);

//양방향 통신 끝
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_Close(IntPtr _apiClientPtr);

//지령을 Code(integer)로, 지령 내용은 문자열로 송신
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_SendCommandByCode(IntPtr _apiClientPtr, int _commandCode, IntPtr _commandData, int _commandLength);

//지령 및 지령 내용을 문자열로 송신
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_SendCommandByStr(IntPtr _apiClientPtr, IntPtr _commandWorkCodeStr, int _commandWorkCodeStrLength, IntPtr _commandData, int _commandLength);

//수신 지령 확인 및 가져오기
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_GetReceiveStackCommand(IntPtr _apiClientPtr, ref int _commandType,ref IntPtr _commandData, ref int _commandDataLength);

//지령 리스트 업데이트
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_GetCommandList(IntPtr _apiClientPtr, ref IntPtr _resultCommandWorkListData, ref int _resultCommandWorkDataLength, ref IntPtr _resultCommandTypeListData, ref int _resultCommandTypeDataLength);

//설정된 시간만료 값 불러오기
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_GetTimeOut(IntPtr _apiClientPtr, ref int _connectTimeOut, ref int _sendTimeOut);

//시간만료 값 설정하기
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
참조 1개
public static extern void ApiClient_SetTimeOut(IntPtr _apiClientPtr, int _connectTimeOut, int _sendTimeOut);
```

C# DLL 사용법

- Extern 함수 추가시 아래와 같은 DLL Attribute를 사용하여 정의 및 호출

```
[DllImport("KLib3.dll", CallingConvention = CallingConvention.Cdecl)]
public static extern {반환타입} {함수이름}({인자 목록 및 형식});
```

- 반드시 Extern 함수명, 인자 목록 및 형식, 반환 형식은 일치 필수
- SendCommandByStr 또는 SendCommandByCode에 지령 내용을 대입할 때 16bit Unicode로 변환 후 byte[]로 대입(1개 단일 문자당 2개 byte 사용)
- Pointer(* 또는 *&) 인자 필요 시 IntPtr 대입 - 예) char*& _str -〉 IntPtr _str
- Reference(&)인자 필요 시 ref 사용 - 예)int& _length -〉 ref int _length

Python Extern 함수 목록 예시

```python
import ctypes
from ctypes import cdll

class CDllWrapper:
    def __init__(self):

        self.dll = cdll.LoadLibrary('KLib3.dll')

        # 생성자 초기화
        self.dll.CreateApiClient.restype = ctypes.c_void_p
        # API Handler
        self.dllWrapperPtr = ctypes.c_void_p()
        self.dllWrapperPtr = self.dll.CreateApiClient()
        # 소멸자
        self.dll.DisposeApiClient.restype = None
        self.dll.DisposeApiClient.argtypes = [ctypes.c_void_p]
        # API 통신 시작
        self.dll.ApiClient_Open.restype = ctypes.c_bool
        self.dll.ApiClient_Open.argtypes = [ctypes.c_void_p]
        # 수신 지령 중 가장 먼저 쌓인 지령을 출력
        self.dll.ApiClient_GetReceiveStackCommand.restype = None
        self.dll.ApiClient_GetReceiveStackCommand.argtypes = [ctypes.c_void_p,ctypes.POINTER(ctypes.c_int),ctypes.POINTER(ctypes.c_char_p),ctypes.POINTER(ctypes.c_int)]
        # 수행 지령(Work Command) 목록와 지령 타입(Type Command)을
        self.dll.ApiClient_GetCommandList.restype = None
        self.dll.ApiClient_GetCommandList.argtypes = [ctypes.c_void_p,ctypes.POINTER(ctypes.c_char_p),ctypes.POINTER(ctypes.c_int),ctypes.POINTER(ctypes.c_char_p),ctypes.POINTER(ctypes.c_int)]
        # 지령 이름 문자열로 지령 송신
        self.dll.ApiClient_SendCommandByStr.restype = None
        self.dll.ApiClient_SendCommandByStr.argtypes = [ctypes.c_void_p,ctypes.c_char_p,ctypes.c_int,ctypes.c_char_p,ctypes.c_int]
        # 지령 코드로 지령 송신
        self.dll.ApiClient_SendCommandByCode.restype = None
        self.dll.ApiClient_SendCommandByCode.argtypes = [ctypes.c_void_p,ctypes.c_char_p,ctypes.c_int]
        # API 통신 종료
        self.dll.ApiClient_Close.restype = ctypes.c_bool
        self.dll.ApiClient_Close.argtypes = [ctypes.c_void_p]
        # 지령 인자 예시
        self.CommandWorkListLength = ctypes.c_int()
        self.CommandWorkList = ctypes.c_char_p()
        self.CommandTypeListLength = ctypes.c_int()
        self.CommandTypeList = ctypes.c_char_p()
        self.stackCommandType = ctypes.c_int()
        self.stackCommandDataLength = ctypes.c_int()
        self.stackCommandData = ctypes.c_char_p()
```

Python DLL 사용법

- DLL 사용을 위한 아래 라이브러리 추가

```
import ctypes
from ctypes import cdll
```

- Extern 함수를 아래와 같이 정의

```
self.dll = cdll.LoadLibrary('KLib3.dll')
self.dll.{함수이름}.restype = ctypes.{반환 타입}
self.dll.{함수이름}.argtypes = [{인자 목록 및 타입}]
```

- 반드시 Extern 함수 이름, 인자 목록 및 타입, 반환 타입은 일치 필수
- SendCommandByStr 또는 SendCommandByCode에 지령 내용(CommandData)을 대입할 때 “utf-16le”로 변환하여 대입(1개 단일 문자당 2개 byte 사용)
- Pointer(* 또는 *& 또는 &) 인자 필요 시 ctype.byref({변수이름}) 사용
  예) char*& _str -〉 ctype.byref(_str)

KITRONYX

압력센서 분야의 기술을 선도하는 글로벌 기업

KITRONYX

T H A N K Y O U