ZEFA®

PULSE Power Supply

DPF 200V/10A

User's Guide



㈜ 제파텍

서울특별시 금천구 가산동 326-4

TEL. (02) 862-9300 (대표전화)

FAX. (02) 857-4365

http://www.zefa.co.kr

안전에 관한 주의 사항



안 전

◈ 사용자 임의로 제품 내부를 분리, 개조, 수리 하지 마십시오

◈ 제품 위에 금속류, 물건, 음료수, 약품 등을 올려 놓지 않도록 하십시오

• 감전, 제품 손상의 위험이 있습니다.

◈ Power Supply을 사용하지 않을 경우, 전원 플러그를 제거하십시오



주 의

◈ 전원은 220V, 50/60Hz으로 사용합니다.

◈ 전원 케이블을 다룰 때에는 굽어지거나 손상이 가지 않도록 합니다.

◈ 제품은 벤젠, 신나, 알코올 등으로 닦지 않도록 합니다



이 기호는 잘못 취급했을 때 제품내의 비 절연 전압으로 전기 충격이 발생하여 사망 또는 중상을 입을 수 있는 경우를 표시합니다.



이 기호는 작동 및 유지에 관련된 정보를 알림으로써 주의 깊게 읽어 두면 유용한 정보를 표시합니다.

목 차

1. 개요

2. 설치

2-1 제품의 이상 확인

2-2 사용 조건

2-3 입력 전원 확인

2-4 DPF -200V/10A의 구성 및 기능

3. 수동 조작

3-1 OUTPUT ON/OFF

3-2 전압 및 전류 동작

3-3 펄스 출력 모드 설정

3-4 펄스 주파수 및 듀티 설정

3-5 로컬 모드 전환

4. SCPI 명령

4-1 SCPI 명령 요약

4-2 출력 설정 및 측정 관련 명령

5-3 Trigger 관련 명령

5-4 시스템 관련 명령

5-5 Calibration 관련 명령

5. 에러 메시지

6. 응용 프로그램

7. 도면

1

개 요

DPF - 200V/10A 는 Remote Programming 조작과 수동조작이 가능한 직류전원 공급 및 펄스 형태의 출력을 공급하는 장치로써 연구, 개발, 제어 계측 등 정밀이 요구되는 분야 및 일반분야에 사용이 가능하도록 설계 되었습니다.

Remote 인터페이스는 TCP/IP 방식의 Ethernet을 채택하여 외부에서 DPF - 200V/10A 를 보다 쉽게 제어 하실 수 있습니다.

기본 기능

- 출력의 ON/OFF기능
- 전압, 전류의 수동가변 기능
- 펄스 출력 모드 설정 (Common, Separate, DC) 가능
- 펄스 출력의 (Frequency, Duty) 가변 제어 가능
- TCP/IP 방식의 Ethernet 외부 Remote Interface 기능

2

설 치

2-1. 제품의 이상확인

- KEY 등 각종 부착물들의 파손여부를 확인합니다
- Display 부 및 전면부, 기타부분에 깨진 곳이나 파손 되었는지 확인합니다

2-2. 사용 조건

- 주위 온도 : 0~40℃
- 주위 습도 : 80% 이하
- 고도 : 2000m 이하

기타 진동과 자기장의 영향이 없는 곳에서 사용하여주시고, 제품의 냉각을 위해 좌우 공기 흡입구 및 후면 공기배출구에는 적정한 공간을 확보 하여 주십시오. 만일 이 공간이 확보되지 않으면 제품의 파손이 예상 되므로 반드시 확보하여 주십시오.

2-3. 입력전원 확인

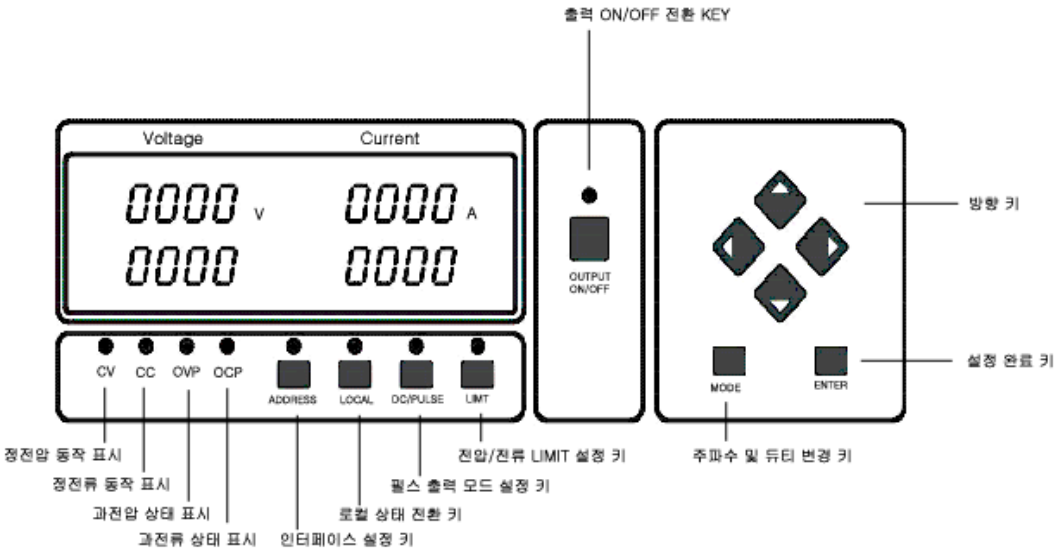
본 제품은 단상220V/ 50~60Hz로 설계되어 있습니다, 입력전원이 다른 경우에는 별도로 제작해야 하므로 문의하여 주십시오. 제품 출고 시에 함께 포장된 AC Power cord를 사용하여 주십시오



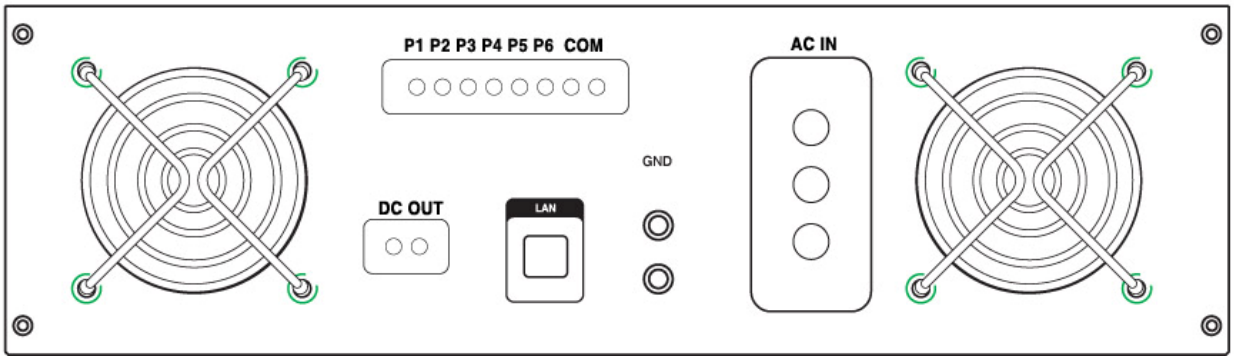
전원 케이블을 다룰 때에는 굽어지거나 손상이 가지 않도록 합니다.
반드시 Earth된 상태에서 사용하십시오.

2-4. DPF – 200V/10A 의 구성 및 기능

전면 디스플레이부의 구성 및 기능



후면부 구성





출력 ON/OFF 키

Power Supply 의 출력 On/Off 상태를 전환 하실 수 있습니다.



인터페이스 설정 키

외부 인터페이스의 (TCP/IP)의 통신 환경을 설정 하실 수 있습니다.



로컬 상태 전환 키

외부 인터페이스 동작 상태에서 로컬 상태 (수동 동작) 상태로 전환 하실 수 있습니다.



펄스 출력 모드 설정 키

펄스 및 DC 출력 모드를 변경 하실 수 있습니다.



전압/전류 LIMIT 설정 키

전압 및 전류의 LIMIT 현재 설정 확인 및 설정을 변경 하실 수 있습니다.



주파수 및 듀티 변경 키

펄스의 주파수 및 듀티를 설정 하실 수 있습니다.



설정 완료 키

현재 설정을 완료 및 저장을 하실 수 있습니다.



방향키

출력 및 설정을 변경 하실 때 자리수 및 설정치의 증감 하실 수 있습니다.



정전압 동작 모드 표시

정전압 동작 모드를 표시 합니다. 정전압 동작 모드 일 경우에는 LED가 점등 됩니다.



정전류 동작 모드 표시

정전류 동작 모드를 표시 합니다. 정전류 동작 모드 일 경우에는 LED가 점등 됩니다.



과전압 상태 표시

과전압 상태를 표시 합니다. 과전압 상태일 때 LED 점등 및 부저가 울립니다.



과전류 상태 표시

과전류 상태를 표시 합니다. 과전류 상태일 때 LED 점등 및 부저가 울립니다.

3

수동 동작

3-1. Output On/Off

• 수동 동작

1 출력 단자와 접지 상태를 확인 하십시오

2 Power Supply의 전원을 연결 하십시오

3 

OUTPUT Key을 사용하여 출력을 On 시킵시오
(Power Supply 는 전원 인가 시 Output Off 상태입니다.)

4  Key을 사용하여 출력을 Off 시킵시오.

5  UP/DOWN Key로 전압을 변경 하십시오

• Remote 인터페이스 에서의 동작

출력 On : OUTPut ON

출력 Off : OUTPut OFF

3-2. 전압 및 전류 동작

• 수동 동작

1 출력 단자와 접지 상태를 확인 하십시오

2 Power Supply의 전원을 연결 하십시오.
Power Supply는 전원인가 시 Output Off 상태 초기화 됩니다

3 

OUTPUT Key을 사용하여 출력을 On 시킵시오
(Power Supply 는 전원 인가 시 초기 Output Off 상태입니다.)

4 
LEFT/RIGHT Key를 변경 하시고자 하시는 전압 또는 전류의 범위를 선택 하십시오. 선택된 자리수는 디스플레이가 깜빡입니다.

5 
UP/DOWN Key로 전압을 변경 하십시오

정전압 동작 모드 상태에서는 전압을, 정전류 동작 모드 상태에서는 전류를 가변 하실 수 있습니다. 가변 하시기 전에 먼저 동작 모드를 확인 하십시오.

정전압 동작 모드 ● CV ○ CC

정전류 동작 모드 ○ CV ● CC

• Remote 인터페이스 에서의 동작

전류 설정 : CURRent {<current>|MIN|MAX}
전압 설정 : VOLTage {<voltage>|MIN|MAX}
OUTPUT ON 설정 : OUTPut ON

3-3. 펄스 출력 모드 설정

• 수동 동작

1 

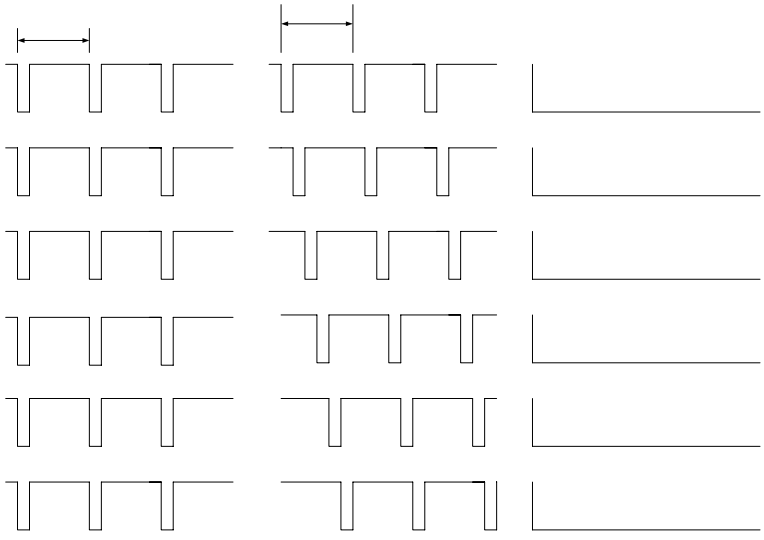
DC/PULSE Key을 펄스 출력 설정 모드로 들어 가십시오.

2 
LEFT/RIGHT Key를 변경 하시고자 하시는 전압범위를 선택 하십시오.
선택된 자리수는 디스플레이가 깜빡입니다.

3 
UP/DOWN Key로 전압을 변경 하십시오

※ 펄스 출력 모드

MODE 1	**전체 동일 펄스 출력 모드**
MODE 2	**주파수의 6 시분할 출력 모드**
MODE 3	**DC 출력 모드**



• Remote 인터페이스 에서의 동작

Function:PWM <mode>

3-4. 펄스 주파수 / 듀티 설정

• 수동 동작

1 

MODE Key을 펄스 출력 설정 모드로 들어 가십시오.

2 
LEFT/RIGHT Key를 변경 하시고자 하시는 주파수 또는 듀티의 범위를 선택 하십시오. 선택된 자리수는 디스플레이가 깜빡입니다.

3 
UP/DOWN Key로 주파수 및 듀티를 변경 하십시오

※ 주파수 및 듀키 출력 범위

	주파수	듀티
MODE	100 ~ 6500Hz	0 ~ 98%
MODE 2	100 ~ 3000Hz	0 ~ 15%

• Remote 인터페이스 에서의 동작

주파수 설정 : CONFigure:FREQuency {<frequency>|MIN|MAX}
듀티 설정 : CONFigure:DUTY {<duty>|MIN|MAX}

3-5. 로컬 모드 전환

- **로컬/리모트 동작 모드**

Ethernet 인터페이스 (TCP/IP)로 제어 하는 Remote Mode 상태에서 로컬 모드로 변경하여 수동으로 사용하실 수 있습니다. Remote Mode 상태는 통신 명령에 위하여 자동으로 이루어집니다.

우선순위		상태 변경
REMOTE	1	Ethernet 인터페이스로 통신 명령에 위하여 자동 변경
LOCAL	2	Local Key를 사용한 수동 변경

- **로컬 동작 모드 상태 변경**

1 

MODE Key을 펄스 출력 설정 모드로 들어 가십시오.

4

Remote 인터페이스

4-1. SCPI 명령 요약

출력 설정 및 측정 관련 명령

CURRent {MIN|MAX| < current >}

CURRent?

VOLTage {MIN|MAX|<volts>}

VOLTage?

MEASure?

MEASure:VOLTage?

MEASure:CURRent?

CONFigure:FREQuency {MIN|MAX|<frequency>}

CONFigure:DUTY {MIN|MAX|<duty>}

FUNCtion:PWM {<mode>}

시스템 관련 명령

SYSTem:ERRor?

SYSTem:VERSion?

SYSTem:BEEPer[:IMMediate]

OUTPut[:STATe] {0|1|OFF|ON}

OUTPut[:STATe]?

*IDN?

4-2. 출력 설정 및 측정 관련 명령

CURRent {MIN|MAX|amps>}

Power Supply 전류를 변경하기 위한 명령어입니다. 출력 범위는 인자 값을 변경하여 전류의 값을 변경 시킬 수 있습니다. 출력 범위는 정의 되어 있는 “MIN” 또는 “MAX”, 그리고 “0” 에서 MAX의 설정 값 까지 변경 시킬 수 있습니다. 인자 값의 값들이 측정범위를 벗어 났을 경우 비프음 과 같이 에러 메시지에 “err- 222 Data out of range” 표시 됩니다.

CURRent?

현재 설정 되어 있는 전류 값을 반환합니다.

VOLTage {MIN|MAX|<volts>}

Power Supply 전압를 변경하기 위한 명령어입니다. 출력 범위는 인자 값을 변경하여 전압의 값을 변경 시킬 수 있습니다. 출력 범위는 정의 되어 있는 “MIN” 또는 “MAX”, 그리고 “0” 에서 MAX의 설정값 까지 변경 시킬 수 있습니다. 인자 값의 값들이 측정범위를 벗어 났을 경우 비프음 과 같이 에러 메시지에 “err- 222 Data out of range” 표시 됩니다.

VOLTage?

현재 설정 되어 있는 전압 값을 반환합니다.

MEASure?

현재 Power Supply 에서 측정되고 있는 전압 및 전류값을 반환합니다.

MEASure:VOLTage?

현재 Power Supply 에서 측정되고 있는 전압값을 반환합니다.

MEASure:CURRent?

현재 Power Supply 에서 측정되고 있는 전류값을 반환합니다.

CONFigure:FREQuency {MIN|MAX|<frequency>}

펄스 주파수를 변경하기 위한 명령어입니다. 출력 범위는 인자 값을 변경하여 주파수의 값을 변경 시킬 수 있습니다. 출력 범위는 정의 되어 있는 “MIN” 또는 “MAX”, 그리고 “0” 에서 MAX의 설정값 까지 변경 시킬 수 있습니다. 인자 값의 값들이 측정범위를 벗어 났을 경우 비프음 과 같이 에러 메시지에 “err- 222 Data out of range” 표시 됩니다.

CONFigure:DUTY {MIN|MAX|<duty>}

펄스 듀티를 변경하기 위한 명령어입니다. 출력 범위는 인자 값을 변경하여 듀티 값을 변경 시킬 수 있습니다. 출력 범위는 정의 되어 있는 “MIN” 또는 “MAX”, 그리고 “0” 에서 MAX의 설정값 까지 변경 시킬 수 있습니다. 인자 값의 값들이 측정범위를 벗어 났을 경우 비프음 과 같이 에러 메시지에 “err- 222 Data out of range” 표시 됩니다.

FUNCtion:PWM {<mode>}

펄스 출력 모드를 변경하기 위한 명령어 입니다.

<mode>의 인자값은 1, 2, 3 의 인자값을 받습니다.

4-3. SCPI 란?

SCPI (Standard Commands for Programmable Instruments)는 ASCII 기반으로 된 Test & Measurement(T&M) 기기들의 명령 언어입니다.
SCPI 명령은 트리 구조의 계층적 구조를 기반으로 되어 있습니다. 계층적 구조에는 연관된 명령은 공통 노드 또는 루트 아래 하나의 서브 시스템을 구성 합니다.

[SOURce:]

CURRent {<current>|MIN|MAX|UP|DOWN}
CURRent? [MIN|MAX]
CURRent:
TRIGgered {<current>|MIN|MAX}
TRIGgered?{MIN|MAX}
VOLTage {<voltage>|MIN|MAX|UP|DOWN}
VOLTage? [MIN|MAX]
VOLTage:
TRIGgered {<voltage>|MIN|MAX}
TRIGgered? {MIN|MAX}

위에서 SOURce 는 루트 키워드 아래 CURRent, VOLTage 의 첫 번째 키워드와 TRIGgered 란 두 번째 키워드로 구성 된 구조입니다.

• **SCPI 명령 포맷**

CURRent {<current>|MIN|MAX|UP|DOWN}

명령구문은 영문 대소문자의 혼합 형태로 구성되어 있습니다. 대문자는 명령어의 약어를 나타냅니다.
명령어의 여러 변수를 구분 하기 위하여 중괄호 ({}), 수직막대(|) 직각 괄호 (<>) 또는 대괄호([])를 사용 합니다.

• {} : 명령 문자열에서 변수를 표시 합니다.
• | : 명령 문자열 안에 있는 여러 변수를 구분합니다.
• <> : <> 안에 있는 변수에 값을 지정해야 한다는 의미로 사용 됩니다.
• [] : 명령 문자열에서 변수가 선택 사항임을 가리킵니다. 변수 및 명령 문자열은 생략 가능합니다.

• **명령 구분자**

콜론 (:) 을 사용하여 명령 키워드 사이를 구분 합니다. 변수와 명령 키워드를 구분하기 위해서는 공백 문자를 사용합니다. 변수가 여러 개를 필요로 하는 경우네는 쉼표(,) 을 사용 합니다.

''SOURce:CURRent:TRIGgered''

세미콜론(;) 을 사용하여 동일한 서브시스템에 명령어를 동시에 전송 할 수 있습니다.

''SOUR:VOLT MIN;CURR MAX''

콜론 및 세미콜론을 사용하여 다른 서브시스템의 명령과 연결하여 사용 하실 수 있습니다. 세미콜론과 콜론을 모두 사용 하지 않을 시에는 명령 오류가 발생 합니다.

''DISP:TEXT:CLE;:SOUR:CURR MIN''

- **Query 명령**

물음표(?)가 들어 있는 명령은 현재 값을 반환하라는 명령어입니다.

- **SCPI 명령 종결자**

Power Supply는 전송된 명령 문자열의 종결을 나타내기 위하여 <new line> 문자를 사용합니다. <new line> 문자의 ASCII 표현형식은 0AH 입니다.

- **SCPI 변수 형식**

1. 숫자 형식: 10 진수로 표현합니다. 소수점, 음수, 단위 사용 또한 가능 합니다.
2. 개별 변수: 제한 적인 기능을 가진 인자 값을 사용할 수 있습니다. (BUS)
3. 불린 변수(Boolean 변수): 참이나 거짓형태의 변수를 나타냅니다. 거짓인 경우에는
"OFF"나 "0"을 참인 경우에는 "ON"이나 "1"을 사용합니다.
4. 문자열 변수 : 문자열 변수는 작은 따옴표('')나 큰 따옴표("")과 함께 사용할 수 있습니다.

5

에러 메시지

에러 Queue 는 FIFO (First In First Out) 구조로 되어 있으며 총 10 개의 에러 메시지를 가질 수 있습니다. 20 개의 에러 메시지를 초과 한다면 -350 , “overflow 에러 queue” 에러 가 발생됩니다. 모든 에러 메시지를 읽었을 경우 에러 Indicator 는 꺼집니다.
에러 메시지가 없을 경우 이 명령을 실행 할 경우 “+0, No 에러” 메시지를 반환합니다.

*CLS 명령을 사용하시면 에러 Queue에 저장된 메시지 는 지워집니다.
*SYST:ERR? 명령으로 에러 메시지를 읽을 수 있습니다.

이 Chapter 에러 메시지에 대해 자세히 설명 하고 있습니다.

-100, “Command error”

SCPI 명령의 구문이 틀렸을 경우 발생 합니다. 사용되지 않는 구분 문자를 사용하였을 경우 발생 될 수 있습니다.

-104, “Data type error”

인자 값의 형식이 틀렸을 경우 발생됩니다. Numeric 형식을 string 형식으로 사용하였을 경우에 발생 될 수 있습니다.

-108, “Parameter not allowed”

이 명령에서 예상되는 변수보다 많은 변수가 수신되었습니다. 변수가 필요 없는 명령에 여분의 변수나 추가 변수를 입력했습니다.

-109, “Missing parameter”

사용 인자 값이 사용되지 않았을 경우에 발생 되는 에러 입니다. 예를 들어 전압 설정 명령에서 설정 전압 값을 설정 하지 않았을 경우 발생 합니다.

-113, “Undefined header”

Power Supply에서 사용되고 있지 않은 명령어를 사용하였을 경우 발생 됩니다.

-222, “Data Out of range”

numeric 인자 값이 Programming된 유효 사용 범위를 넘었을 경우 발생 됩니다.

-350, “Queue overflow”

20개 이상의 오류가 발생하여 Error Queue가 가득 찼습니다. 오류를 삭제하지 않으면 오류가 추가 저장되지 않습니다. *CLS명령 또는 전원의 리셋될때 삭제됩니다. *RST명령인 경우는 오류가 삭제 되지 않습니다.

-410, “Query INTERRUPED”

실행 중인 Query 명령에 대한 응답이 전송되기 전에 Query 명령을 받았을 경우 발생됩니다.

-410, “Query UNTERMINATED”

Power Supply 가 메시지 전송 상태로 지정되었지만 Out buffer에 전송 메시지가 수신되지 않았을 경우 발생 합니다. 예를 들어 응답이 없는 명령에 실행 후, 응답을 요구하는 경우에 발생 합니다.

-430, “Query DEADLOCKED”

출력 버퍼에 비해 너무 많은 데이터를 작성하는 명령이 수신되었으며 입력 버퍼 역시 가득 찼습니다. 명령은 계속 실행되지만 모든 데이터가 손실됩니다.

-440, “Query UNTERMINATED after indefinite response”

부정의 응답형식 쿼리문 다음에 다른 쿼리문이 사용 되었습니다.

(IEEE 488.2, 6.5.7.5 참고)

*IDN? 명령은 명령 문자열의 마지막 쿼리명령으로 사용해야 합니다.

601, “Front Panel does not respond”

602, “RAM Read/Write failed”

625, ”IO Processor does not respond”

626, “IO Processor Self-test failed”

6

응용 프로그램

이 Chapter 는 Remote 인터페이스를 사용하여 Power Supply에 대한 응용 프로그램 제작에 관련하여 Remote 인터페이스에 대한 단한예제 그램을 " ZEFA IO library " 을 사용한 예제를 설명합니다.

ZEFA IO Library란?

ZEFA IO Library는 계측장비를 컨트롤 하기 위한 어플리케이션 프로그래밍 인터페이스입니다. 이것으로 계측장비에 명령어를 전송하고 수신하여 장비를 측정하거나 테스트 할 수 있습니다.

개발 환경

ZEFA IO Library는 Microsoft Windows2000과 Windows XP에서 지원되며 이 Windows 버전에서의 C, C++와 Visual Basic이 지원됩니다.
지원되는 I/O 인터페이스는 GPIB, LAN, USB HID가 있습니다.

전송, 수신에 관련된 function을 제공하며 이 function과 사용되는 data type은 VISA 3.0에 기준하여 제공됩니다.

※ 윈도우용 VISA Application 제작

▪ C/C++언어를 이용한 프로그래밍

여기서 Win32 어플리케이션 작성시 컴파일러에 설정해야 할 특성들을 설명합니다.

ZEFA IO Library Linking:

ZEFA IO Library를 인스톨을 하면 다음과 같은 경로에 lib파일이 생성됩니다.
C:₩Program Files₩ZEFATEK₩ZEFA IO Library₩Lib₩ZefaIOLibrary.lib

Microsoft Visual C++ Version6.0환경에서 개발하기

1. File 메뉴에서 새로운 프로젝트를 생성하거나 기존의 프로젝트를 불러옵니다.
2. Project>Settings 메뉴를 선택하여 C/C++ 탭을 클릭합니다.
3. Category의 Code Generation항목을 선택하고 Use Run-Time Libraries에서 Multi-Thread using DLL을 선택하고 OK버튼을 클릭하여 다이얼로그 박스를 닫습니다.
4. Project>Setting 메뉴를 선택하여 Link 탭을 클릭하고 Object/Library Modules에 ZefaIOLibrary.lib를 입력합니다. OK버튼을 클릭하여 다이얼로그 박스를 닫습니다.
5. include files와 library files의 경로를 다음과 같이 입력합니다.
   -Tools>Options메뉴를 선택합니다.
   -Directories 탭을 을 클릭하여 include file path를 설정합니다.
   -Show Directories For에서 Include Files를 선택합니다.
   하단의 리스트 박스를 클릭하여 다음과 같이 경로를 입력합니다.
   C:₩Program Files₩ZEFATEK₩ZEFA IO Library₩include
   (ZEFA IO Library 설치 시에 위와 같은 경로에 설치 된다고 가정합니다..)
   - Show Directories For에서 Library Files를 선택합니다.
   하단의 리스트 박스를 클릭하여 다음과 같이 경로를 입력합니다.
   C:₩Program Files₩ZEFATEK₩ZEFA IO Library₩lib
6. C또는 C++파일을 추가하거나 새로 생성합니다.
   예를 들어 다음과 같은 경로의 파일을 추가합니다.
   C:₩Program Files₩ZEFATEK₩ZEFA IO Library₩Sample₩idn.c
7. Build > Rebuild All 을 클릭하여 프로그램을 완성시킵니다.

Sample Program (C/C++)

다음은 idn.c파일의 소스코드 입니다.

/*idn.c*/

```
#include <ZefaIOLibrary.h>
#include <stdio.h>

void main ()
{
        ViSession defaultRM, vi;
        char buf [256] = {0};
        unsigned int retLen;
        unsigned int ret;

        /* Open session to GPIB device at address 0 */
        ziOpenDefaultRM(&defaultRM);
        ret = ziOpen(defaultRM, "GPIB::0::INSTR", VI_NULL, VI_NULL, &vi);

        /* Send an *IDN? string to the device */
        ziWrite (vi, (unsigned char*)"*IDN?₩n", 6, &retLen);

        /* Read results */
        ziRead(vi, buf, 255, &retLen);

        /* Print results */
        printf("Instrument identification string:%s₩n", buf);

        /* Close session */
        ziClose(vi);
        ziClose(defaultRM);
}
```

<ZefaIOLibrary.h>

프로그램 시작 부분에 이 파일을 포함시켜서 Zefa IO Library의 function과 prototypes들을 제공합니다.

ViSession

ViSession은 Zefa IO Library 의 data type입니다. 각 object를 통신을 하기 위해서 ViSession이 정의 되야 합니다.

ziOpenDefaultRM

viOpenDefaultRM을 사용하여 default resource manager의 session을 open시킵니다.
viOpenDefaultRM은 default resource manager를 초기화 시키고
default resource manager session의 Pointer를 반환합니다.

ziOpen

이 function은 계측장비의 I/O에 통신을 연결합니다.
연결이 성공되면 Zefa IO Library의 다른 function을 사용하기 위한 device session이 반환됩니다.

ziWrite/ziRead

device session을 통하여 계측장비에 명령어를 전송하고 응답 데이터를 수신합니다.

ziClose

session 을 종료하기 위해 사용됩니다.
device session을 종료하면 이 session에 할당된 모든 data structure들이 해제 됩니다.
default manager session을 종료하면 이 session으로 open된 모든 session이 종료됩니다.

▪Visual Basic을 이용한 프로그래밍

Visual Basic으로 프로그래밍하기 위해 고려할 사항들은 다음과 같습니다.
(Microsoft Visual Basic version 6.0 기준)

필요 Module 추가

Zefa IO Library의 function을 사용하기 위해 ZefaIOLibrary.bas Visual Basic module을 다음 C:₩Program Files₩ZEFATEK₩Zefa IO Library₩Include 에서 찾아 추가시킵니다.

프로젝트에 ZefaIOLibrary.bas 파일 추가하기

1. Visual Basic 6.0 메뉴에서 Project>Add Module를 선택합니다.
2. Existing 탭을 선택합니다.
3. ZefaIOLibrary.bas파일이 있는 경로를 찾아 선택합니다.
4. Open버튼을 클릭합니다.

Sample Program (Visual Basic)

다음은 GPIB 계측장비에 idn을 보내 identification하는 program 예제 입니다.

1. 계측장비에 GPIB인터페이스로 연결합니다.
2. Visual Basic 6.0 어플리케이션을 시작합니다.
3. Standard EXE 프로젝트를 생성합니다. Project1에 빈 Form1이 생성될것입니다.
4. 메뉴에서 Project>Add Module을 선택합니다.
Existing 탭을 선택하여 idn.bas 예제파일을 추가합니다.
c:₩Program Files₩ZEFATEK₩Zefa IO Library₩Sample₩VB₩IDN
5. idn.bas파일의 Main()서브루틴은 Form없이도 실행됩니다.
만일 Form1을 필요하지 않으면 Project Explorer window에서 Form1에서 마우스 오른쪽 버튼을 클릭한 후 Remove Form1을 선택하여 삭제합니다.
6. Zefa IO Library의 function을 사용하기 위해 이 프로젝트에 ZefaIOLibrary.bas파일을 다음과 같이 추가합니다.
메뉴의 Project>Add Module을 선택하고 Existiing 탭을 선택하여 idn.bas파일이 있는 위치를 찾아 추가 시킵니다.
7. 프로젝트를 컴파일하여 실행시킵니다. 메뉴의 File>Make idn.exe을 클릭하여 EXE파일을 생성하고 실행시킵니다.

다음은 idn.bas 파일의 소스코드 입니다.

```
Sub Main()

    Dim defRM As Long                'Session to Default Resource Manager
    Dim vi As Long                   'Session to instrument
    Dim strRes As String * 200       'Fixed length string to hold results
    Dim results As Long
    Dim retCnt   As Long

    'Open the default resource manager session
    results = ziOpenDefaultRM(defRM)
    'Open the session to the resource
    results = ziOpen(defRM, "GPIB0::0::INSTR", 0, 0, vi)

    'Initialize device
    results = ziWrite(vi, "*RST" + Chr$(10), 5, retCnt)
    'Ask for the device's *IDN string.
    results = ziWrite(vi, "*IDN?" + Chr$(10), 6, retCnt)

   'Read the results as a string.
    results = ziRead(vi, strRes, 200, cnt)

    'Display the results
    MsgBox strRes, vbOKOnly, "*IDN?"

    'Close the vi session and the resource manager session
    results = ziClose(vi)
    results = ziClose(defRM)

End Sub
```

7

도면

