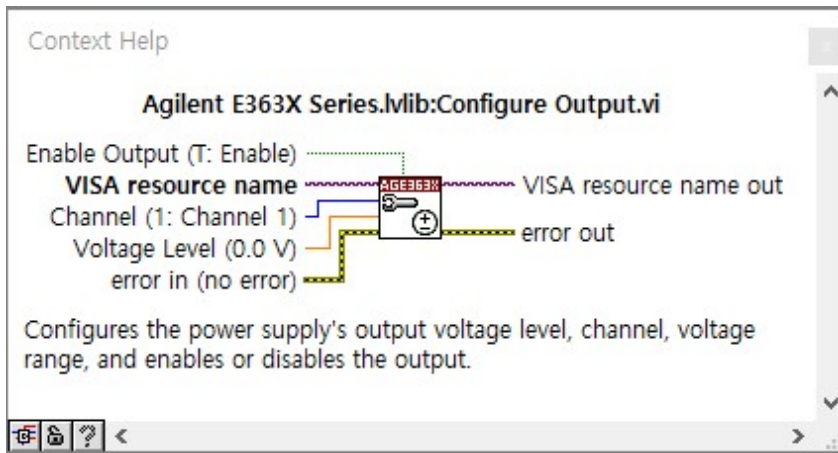


Configure Output.vi

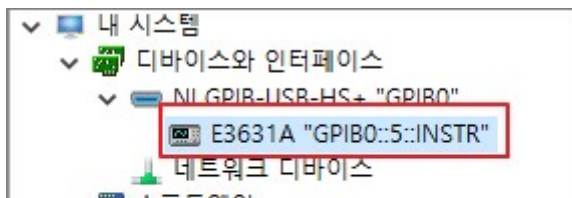
Ctrl+H 를 눌러서 도움말 창을 띄운 다음, Configure Output.vi 위에 마우스를 올려 놓으면 도움말을 확인할 수 있습니다. 보통 다운로드 받은 드라이버는 도움말이 상세하게 작성되어 있으므로, 잘 참고하여 사용하면 되겠습니다.



Configure Output.vi 의 도움말 화면

보통 랩뷰에서 SubVI를 만들 때 왼쪽은 함수의 입력부로 사용하고, 오른쪽은 출력부로 사용합니다. 따라서 여기서에서는 입력 변수들이 [Enable Output, VISA resource name, Channel, Voltage Level, error in] 이고 출력 변수는 [VISA resource name out, error out] 입니다.

- 입력 변수
- Enable Output (T: Enable)
- 출력을 Enable 할 지, Disable 할 지 결정하는 입력 변수로 초록색 라인으로 연결되어 있으므로 불리언 타입의 컨트롤 또는 상수를 연결해야 합니다. 괄호 안의 (T: Enable)은 기본값으로 이 변수의 입력이 없을 경우 기본적으로 항상 입력은 T(True)로 출력이 Enable 됩니다.
- VISA resource name
- 이 함수는 NI VISA를 이용해서 계측기와 통신하는데, 계측기를 구별하기 위한 일종의 주소입니다. NI MAX에서 확인한 계측기의 GPIB 번호를 입력해 주면 됩니다. NI MAX에서 확인했었던 GPIB0::5::NISTR 을 입력으로 사용합니다.

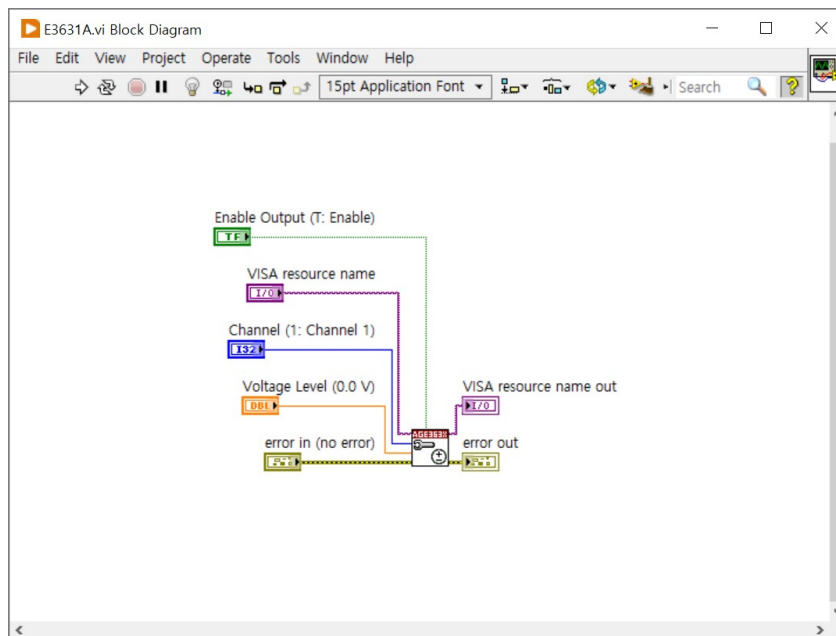




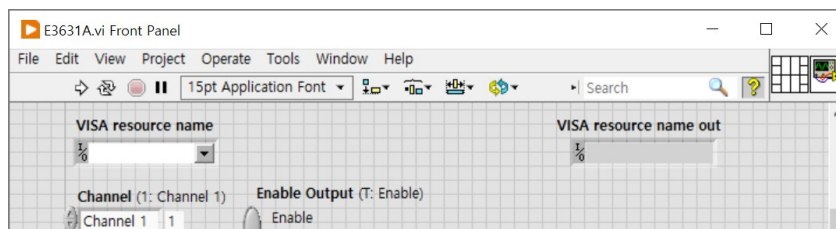
NI MAX에서 확인한 E3631A의 GPIB 번호

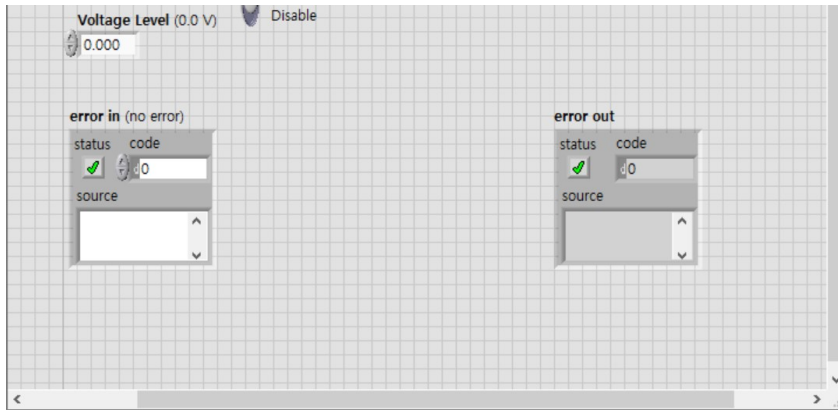
- Channel (1: Channel 1)
- E3631A는 총 3개의 채널이 있습니다. Output을 Enable 할 경우 3개의 출력이 모두 Enable 되지만, 이 함수는 Voltage Level을 정하면서 Output Enable을 하기 때문에 특정 채널을 선택해 주어야 합니다. 마찬가지로 입력하지 않으면 기본값인 Channel 1이 들어갑니다.
- Channel 1 : +6 V
- Channel 2 : +25 V
- Channel 3 : -25 V
- Voltage Level (0.0 V)
- Channel 입력에서 정한 포트의 Voltage Level을 입력해 줍니다. 전류 입력은 없으므로, 이 함수는 정전압 모드로 사용할 수 있는 함수입니다. 기본값은 0.0 V입니다.
- error in
- 랩뷰에서 사용하는 에러 핸들러 입력입니다. 에러가 없는 상태의 에러 핸들러가 이 함수를 통과하고 난 후에 어떤 에러가 있으면 출력에서 확인할 수 있습니다.

아래처럼 블록 다이어그램과 같이 함수 입력/출력에 대한 모든 변수를 컨트롤과 인디케이터로 만들었습니다. 프론트 패널도 대충 정리해 주면 ...



E3631A Output Control Block Diagram





E3631A Output Control Front Panel

프론트 패널에서 적절한 값들을 선택해 준 후에 실행(Ctrl+R)하면 E3631A의 출력이 On/Off 되는 것을 확인할 수 있습니다.

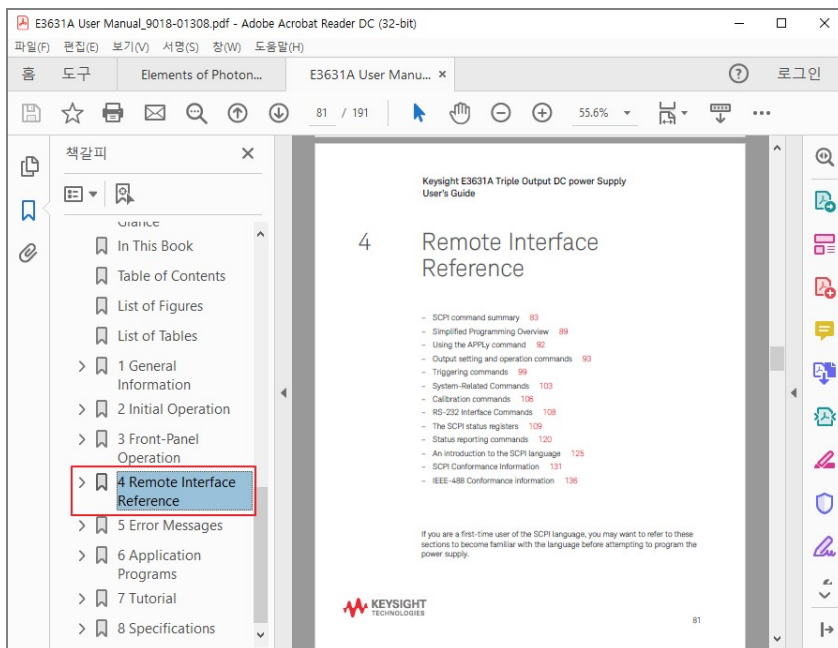
**- 매뉴얼을 참고하기**

**1) 커맨드 검색**

장비사에서 제공하는 랩뷰 드라이버를 사용하는 것이 아니라 매뉴얼을 보고도 같은 프로그램을 만들 수 있습니다. 계측기의 원격 컨트롤 매뉴얼은 User manual 처럼 한 번에 사용법과 원격 컨트롤 커맨드가 같이 정리되어 있을 수도 있고, 원격 컨트롤 커맨드가 따로 정리되어 있는 경우도 있습니다. 따로 정리되어 있는 경우는 보통 기능이 많아서 사용 매뉴얼에 포함시키기 어려운 경우입니다.

따라서 일단 원격 컨트롤 커맨드가 정리되어 있는 매뉴얼을 찾아야 합니다. E3631A의 경우 Keysight 홈페이지에서 매뉴얼을 다운로드 할 수 있습니다.

보통 아래처럼 따로 원격 커맨드가 정리되어 있는 경우가 많습니다.



E3631A User Manual

Output On/Off 커맨드를 찾아보면, 최종적으로 사용하게 될 커맨드

는 아래와 같습니다.

```
OUTPut[:STATe] {OFF|ON}
```

Output on/off and tracking operation commands

**OUTPut[:STATe] {OFF|ON}**

This command enables or disables all three outputs of the power supply. The state of the disabled outputs is a condition of less than 0.6 volts of opposite polarity with no load and less than 60 mA of opposite direction with a short circuit. At \*RST, the output state is off.

**Output On/Off Command**

여기서 사용하는 대괄호 []와 중괄호 {}의 역할은 원격 커맨드 초입에 설명되어 있으므로, 숙지가 필요합니다. 대부분의 경우 비슷한 규칙을 가지고 있기 때문에, 한 번만 익숙해지면 이후에는 쉽게 사용할 수 있습니다. 아래 SCPI command summary를 보면 다음과 같은 규칙을 가지고 있습니다.

- 대괄호 []는 옵션 커맨드를 나타냅니다. (써도 되고 안 써도 됩니다.)
- 중괄호 {}는 커맨드 문자열로 된 변수를 포함합니다.
- 브라켓 <>은 어떤 값 또는 어떤 값이 있는 변수를 넣어야 합니다.
- 수직 바 | 는 두 개 또는 많은 변수 중에 한 개를 골라야 함을 나타냅니다.

SCPI command summary

This section summarizes the SCPI (Standard Commands for Programmable Instruments) commands available to program the power supply over the remote interface. Refer to the later sections in this chapter for more complete details on each command.

Throughout this manual, the following conventions are used for SCPI command syntax.

- Square brackets ([ ]) indicate optional keywords or parameters.
- Braces ({ }) enclose parameters within a command string.
- Triangle brackets (< >) indicate that you must substitute a value or a code for the enclosed parameter.
- A vertical bar (|) separates one of two or more alternative parameters.

First-time SCPI users, see **Chapter 4: An introduction to the SCPI language**

**SCPI command summary**

여기서 나와있지는 않지만, 커맨드는 대문자와 소문자로 이루어져 있는데 소문자는 생략할 수 있습니다. 따라서 우리가 사용하는 커맨드는

```
OUTPut[:STATe] {OFF|ON}
```

아래와 같이 소문자와 대괄호를 생략해서 사용할 수 있고

```
OUTP {OFF|ON}
```

OFF 또는 ON을 입력 변수로 사용하여, Output ON을 하고 싶은 경우 OUTP ON 커맨드를 사용하면 됩니다.

```
OUTP ON
```

추가적으로, Output 상태(Status)를 확인하는 커맨드는 아래와 같고, 계측기가 0(OFF) 또는 1(ON)을 리턴한다고 합니다. 보통 이렇게 상태를 묻거나 특정 값을 반환하기를 원하는 경우에 커맨드 뒤에 ?를 붙여서 마치 우리가 계측기에 물어보는 것과 같은 커맨드를 만듭니다.

```

OUTPut[:STATe]?

This command queries the output state of the power supply. The returned value is "0" (OFF) or "1" (ON).
    
```

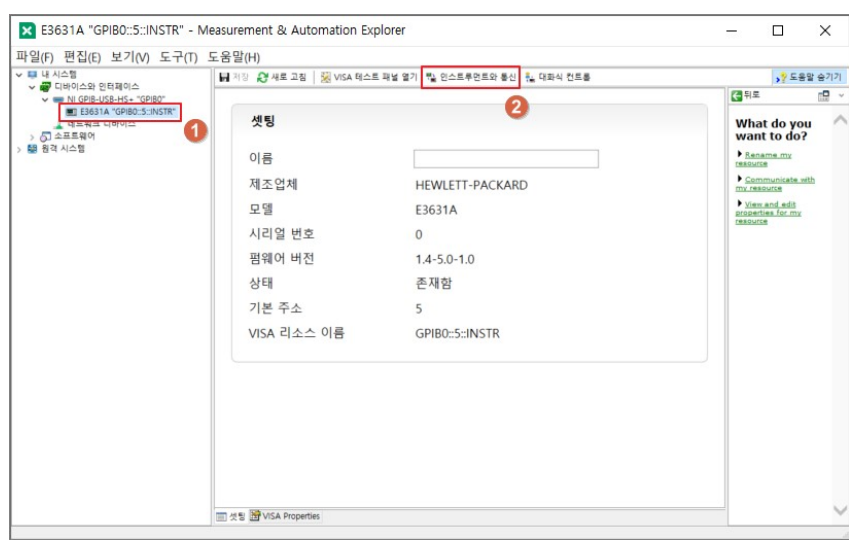
**Output Status Query**

사용해야 할 커맨드를 알았으니, 이제 계측기에 적용시켜 봐야 하겠습니다. 랩뷰로 돌아가기 전에, NI MAX에서 아래 두 가지 커맨드가 정상적으로 작동하는 지 확인해 보겠습니다.

- OUTP ON
- OUTP?

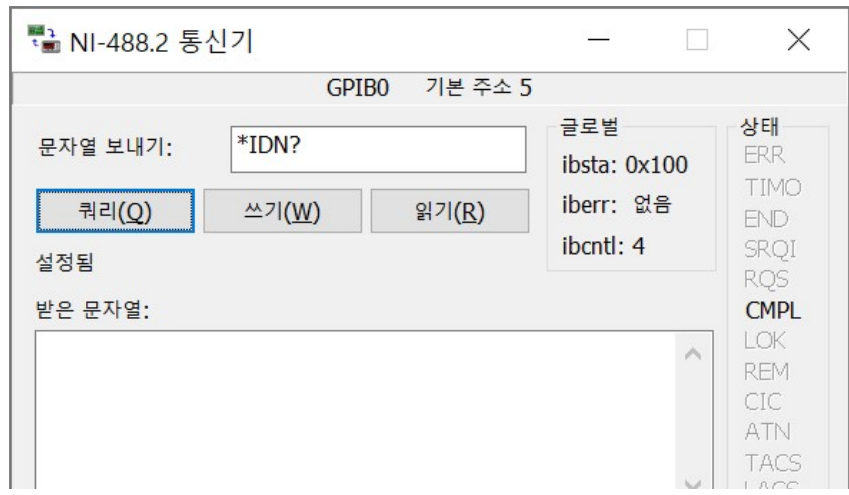
**2) NI MAX를 사용하여 커맨드 확인하기**

NI MAX를 실행하고 왼쪽 목록에서 계측기를 선택한 후, 인스트루먼트와 통신 버튼을 클릭합니다.

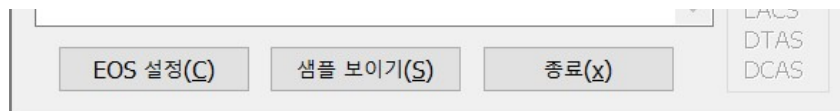


NI MAX에서 인스트루먼트와 통신하기

아래와 같이 NI-488.2 통신기가 실행될 것입니다.



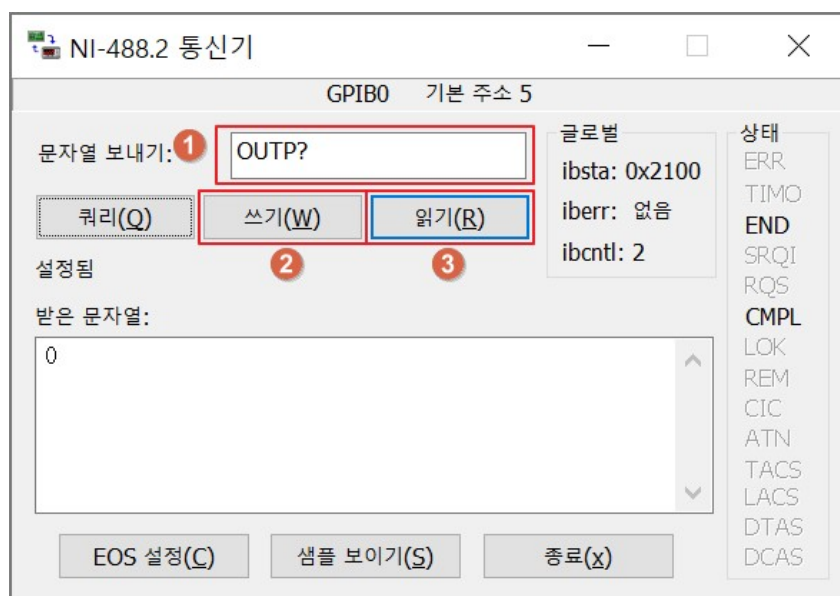




### NI-488.2 통신기

여기서 문자열 보내기 오른쪽에 있는 \*IDN? 부분이 우리가 보내야 할 커맨드를 넣는 곳이고, 아래의 받은 문자열이라고 표기된 부분이 계측기에서 보낸 리턴 값을 표기해 주는 부분입니다.

먼저, OUTP? 커맨드를 쓰고 [쓰기] 버튼을 누르고, 이어서 [읽기] 버튼을 눌러 보겠습니다.



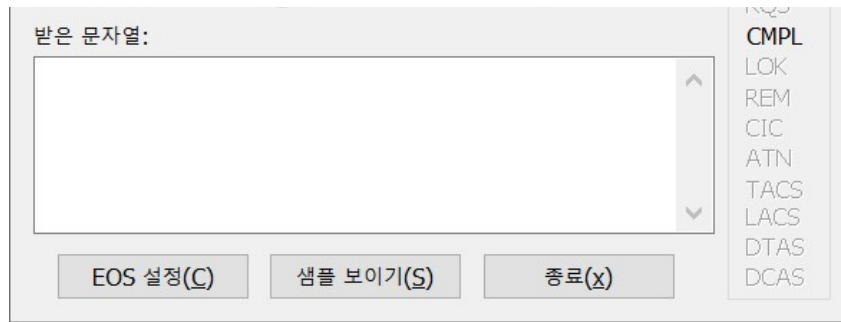
### OUTP? 커맨드

받은 문자열에 0 이 올라온 것을 확인 할 수 있습니다. 0 이므로 현재 Output 상태는 OFF 라는 것을 알 수 있습니다. [쓰기] 버튼은 OUTP? 라는 커맨드를 계측기에 보내는 명령이고, 계측기는 OUTP? 커맨드를 받아서 0 또는 1 의 값을 리턴합니다. [읽기] 버튼은 계측기가 보낸 리턴 값을 읽어오는 작업입니다.

왼쪽에 있는 [쿼리] 버튼은 [쓰기]+[읽기] 를 한 번에 하는 버튼으로, OUTP? 커맨드를 넣고 [쿼리] 버튼만 눌러도 같은 결과를 얻을 수 있습니다. 그런데 [읽기] 는 계측기가 리턴한 값을 읽어오는 작업이기 때문에, 리턴 값이 없으면 리턴 값이 있을 때 까지(또는 제한시간 timeout까지) 기다리게 되므로 리턴 값이 있을 때에만 사용해야 합니다.

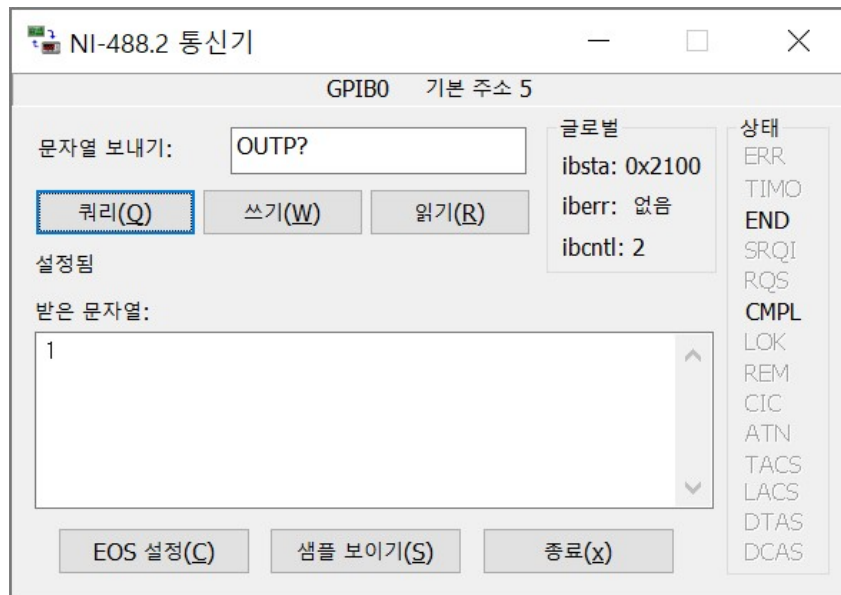
이번에는 Output을 ON 시키는 명령어 OUTP ON 을 사용해 보겠습니다. OUTP ON 커맨드는 리턴값이 없으므로, [쓰기] 버튼만 클릭합니다.





**Output ON Command**

NI-488.2 통신기에서는 아무 반응이 없지만 실제로 계측기의 Output 출력이 켜졌습니다. 이 상태에서 다시 OUTP? 커맨드로 출력이 ON 되었는지 확인해 보겠습니다.



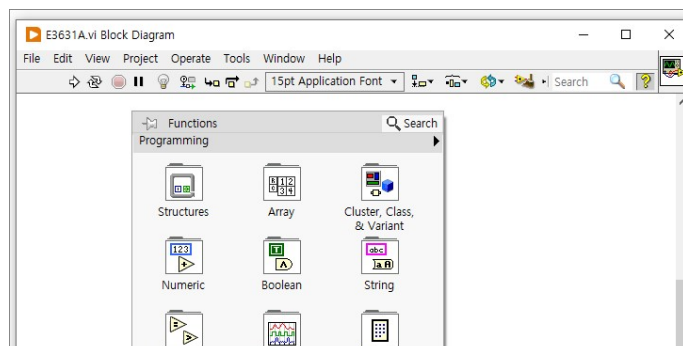
출력 ON 후에 OUTP? 커맨드로 확인

리턴값이 1이므로 출력이 ON인 것을 확인 할 수 있습니다.

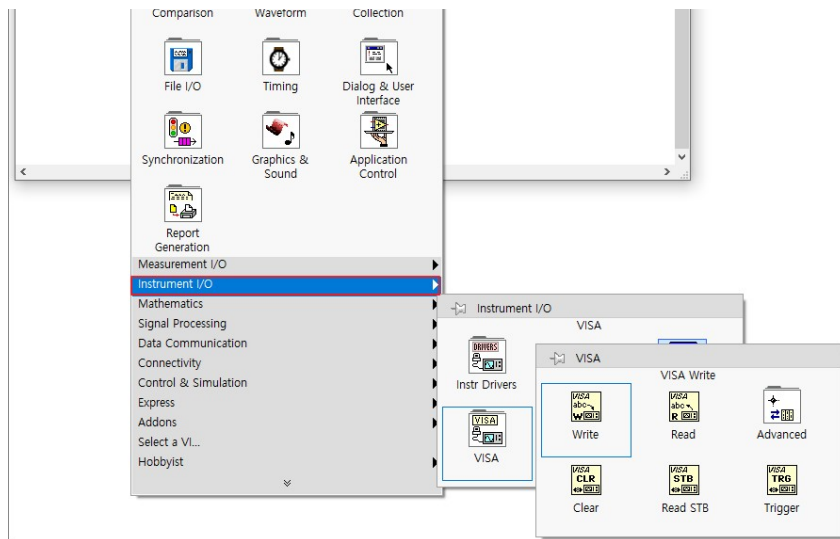
**3) 랩뷰에서 NI VISA를 사용하여 컨트롤하기**

NI VISA(Virtual Instrument Software Architecture)는 이더넷, GPIB, 시리얼 통신, USB 등 여러가지 프로그래밍 인터페이스를 NI에서 통합시켜 놓은 드라이버입니다. 우리는 GPIB를 사용하므로 NI VISA를 통해서 계측기와 통신을 할 수 있습니다.

VISA는 함수 팔레트의 Instrument I/O - VISA 에서 사용할 수 있습니다. 우리는 계측기에 OUTP ON 이라는 커맨드를 보내야(써야) 하므로, VISA Write 함수를 사용하겠습니다.

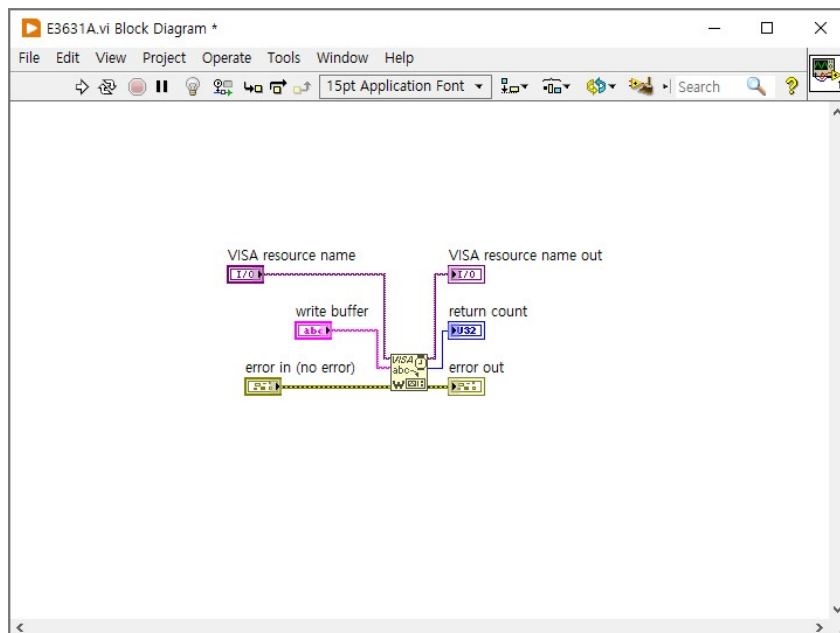






### VISA Write 함수

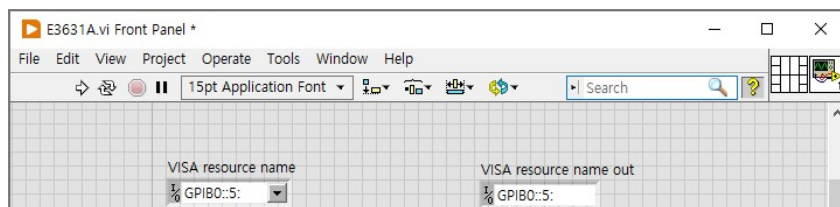
그리고 아래와 같이 모든 터미널에 컨트롤을 연결해 주었습니다. 드라이버를 사용하여 통신할 때와 다른 점은 write buffer와 return count 터미널이 있다는 것 정도입니다.

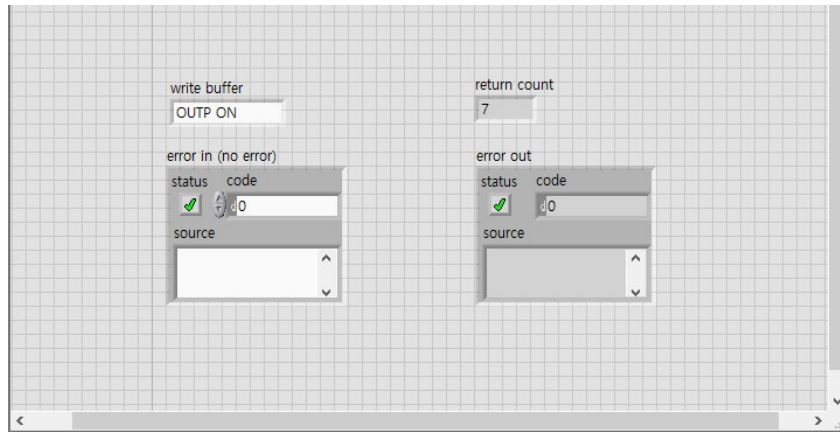


### VISA Write 하기

NI MAX에서 통신했던 것과 마찬가지로, write buffer 문자열 컨트롤에는 우리가 사용할 커맨드인 OUTP ON을 써주면 됩니다. return count 출력 변수는 write buffer 문자열에 쓰여진 바이트를 카운트해서 출력해 주는 것으로, OUTP ON은 공백까지 총 7문자이므로 7byte가 출력 됩니다.

프론트 패널에서 각 컨트롤에 대응하는 값들을 넣어 주고 실행해 보겠습니다. 아래처럼 write buffer에 OUTP ON 커맨드를 써 넣으면 명령이 전달되어 Output 출력이 ON이 됩니다.





Front Panel