

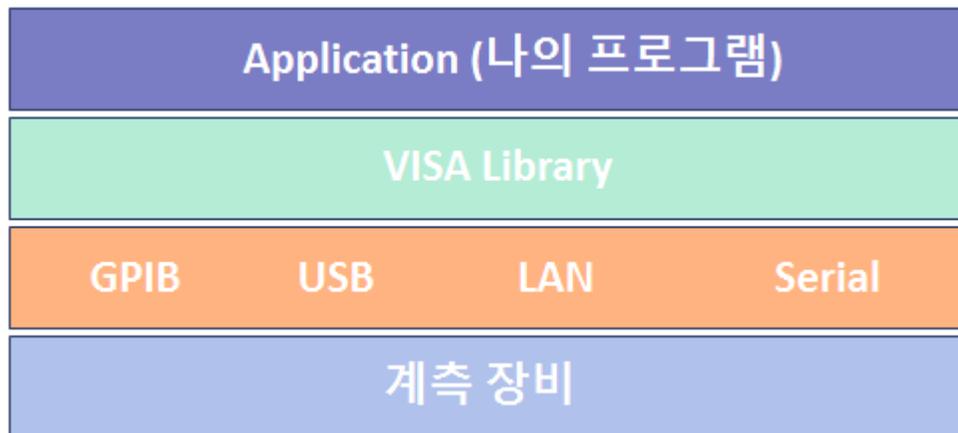
National Instrument(NI)의 GPIB 컨트롤러를 사용하여 Keithley 장비를 인터페이스 하려면 먼저 NI GPIB 컨트롤러를 컴퓨터에 연결하고,

NI-VISA(Virtual Instrument Software Architecture)를 설치

Virtual Instrument Software Architecture(VISA) 의 약자로 계측기와 통신을 하기 위한 표준 라이브러리

NI-488.2 드라이버 설치한다. 설치되면 NI-Max 설치됨

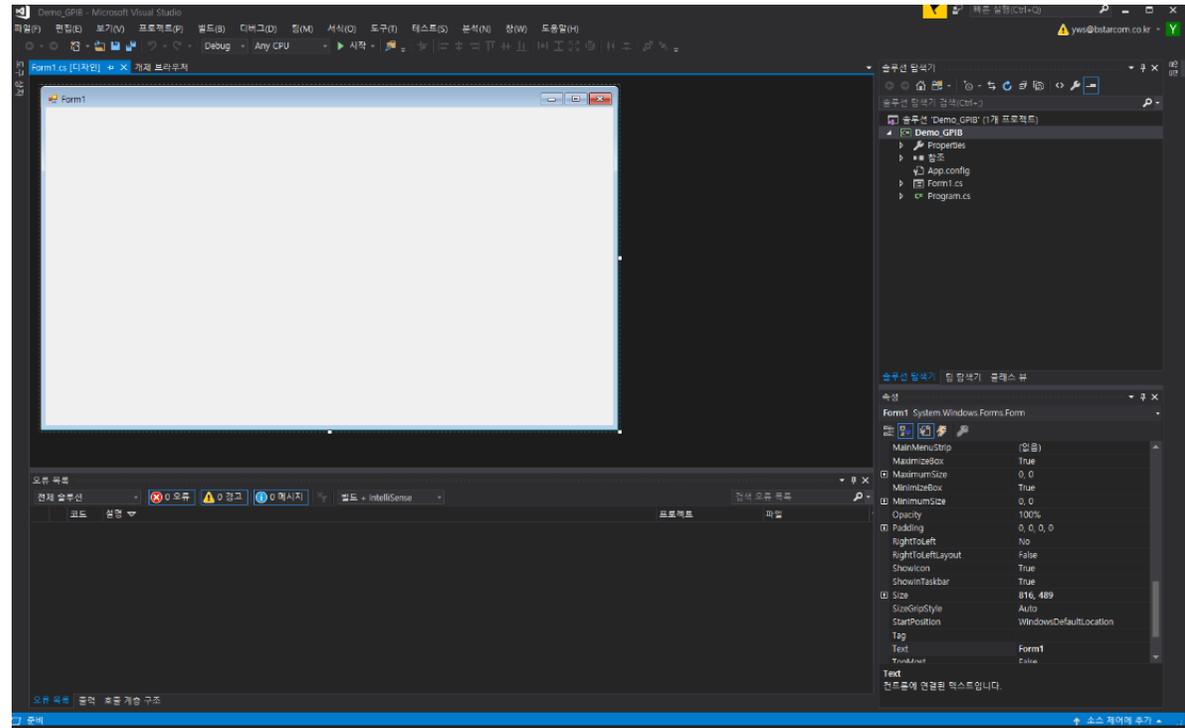
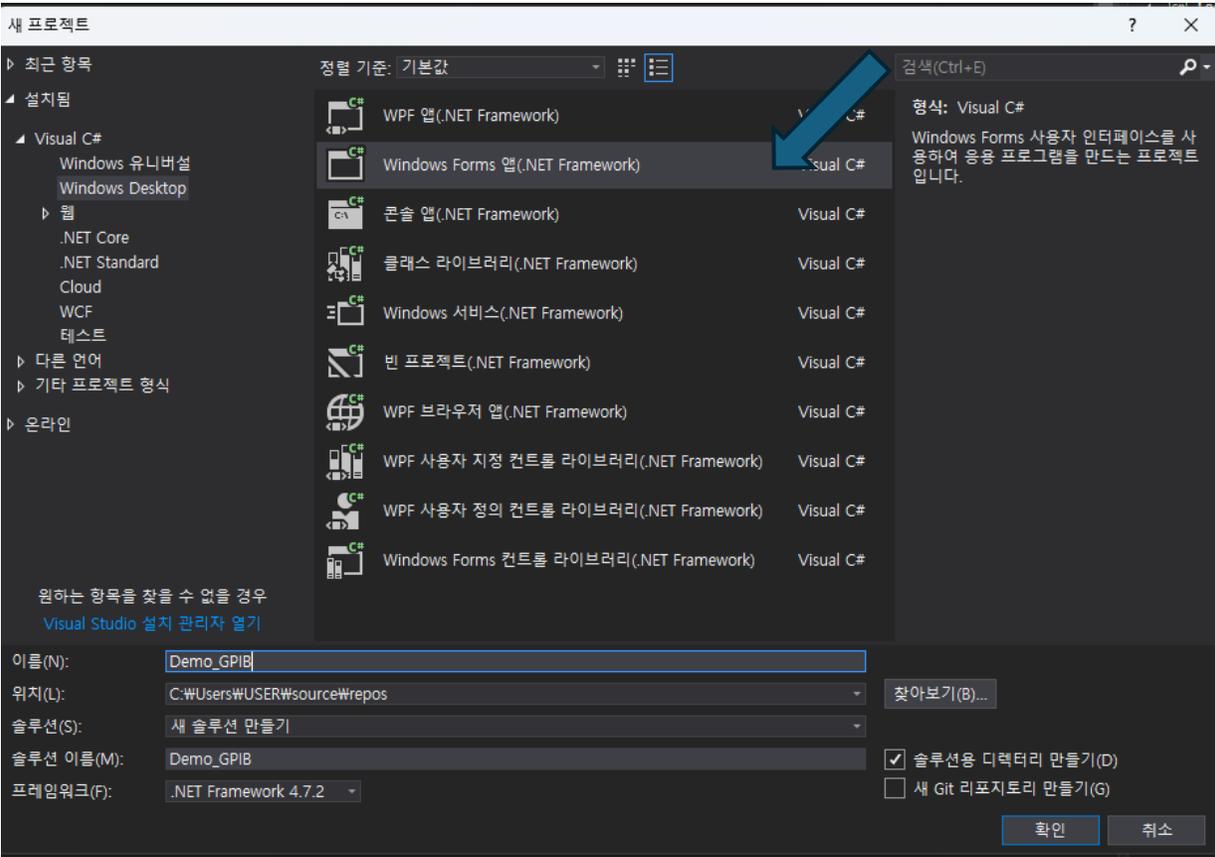
설치 링크 : https://www.ni.com/ko/support/downloads/drivers/download.ni-488-2.html?srsId=AfmBOoowMdRkfG8Qysg8yrANBjHCOufYMqDIPA_AikvxwqAPw2G90vrL#544048



- C#으로 프로그램을 제작하기 위해 Visual Studio를 실행
- NI-GPIB 컨버터를 USB포트에 연결하고 장비가 준비가 되었는지 확인한다.
- Visual Studio를 실행하여 새 프로젝트를 시작한다.
- Windows Form(.NET Framework)선택한다.



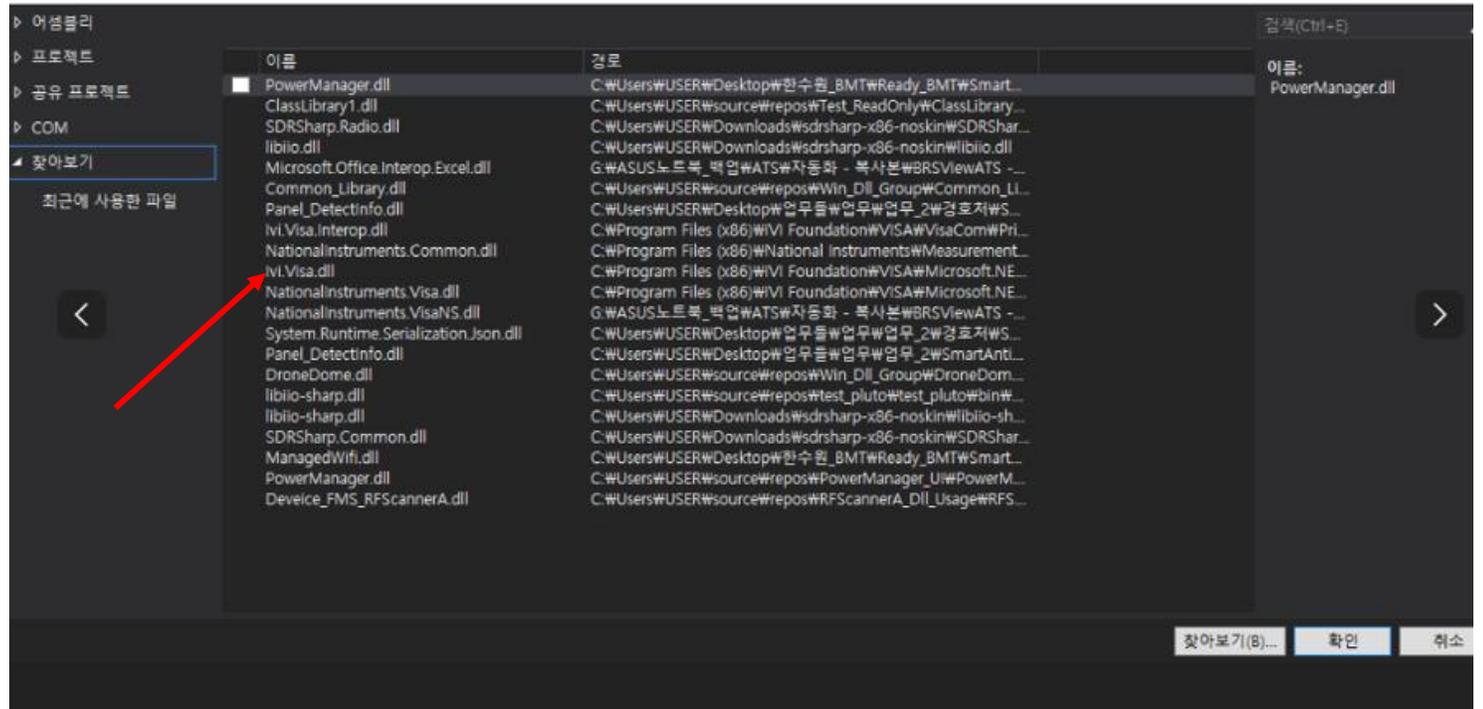
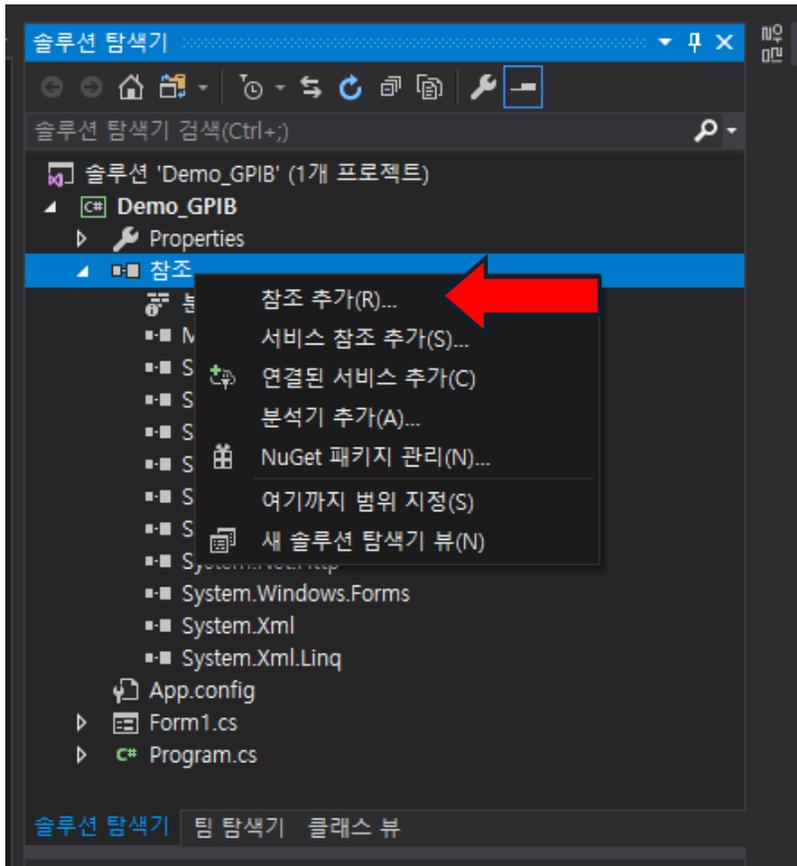
NI-GPIB 컨버터

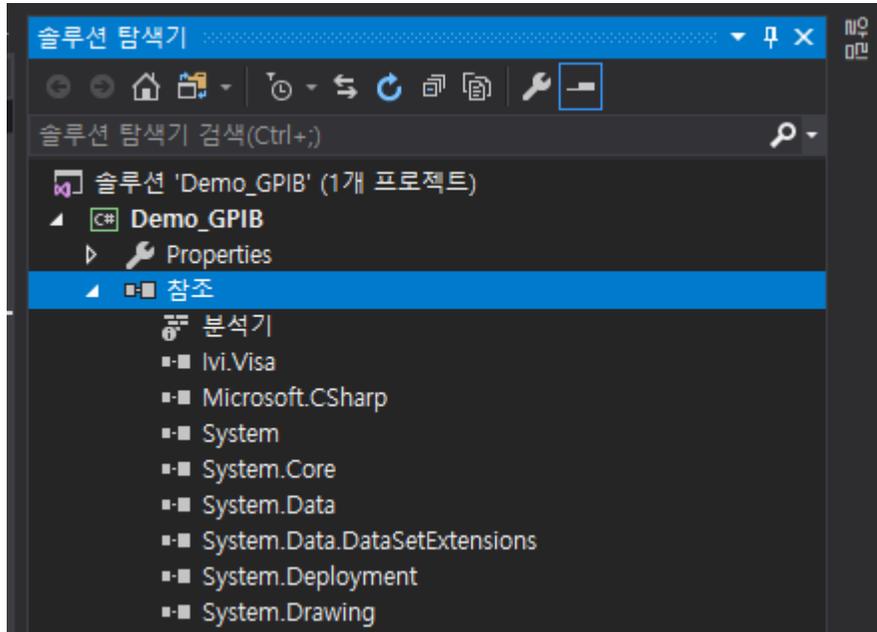


프로젝트 화면

- 프로젝트를 실행 할 때 제일 먼저 할 일은 라이브러리 추가

솔루션 탐색기에서 [참조]라는 부분 클릭하고 우클릭하면 Context 메뉴가 나오는데 여기서 [참조 추가] 클릭





참조에 Ivi.Visa 추가 된 것을 확인합니다.

앞에 느낌표가 있을 경우 문제가 있는 것이니 에러를 확인

Ni488.2 도 추가

VISA"와 "NI-488.2 드라이버"는 둘 다 계측기 제어(Instrument Control)에 사용되는 소프트웨어

항목	VISA (Virtual Instrument Software Architecture)	NI-488.2
개발사	National Instruments (NI) 등	National Instruments (NI)
기능	다양한 계측기 인터페이스 통합 API 제공	GPIB (IEEE-488) 인터페이스 제어 전용 드라이버
지원 인터페이스	GPIB, USB, Serial (RS-232/485), Ethernet, PXI 등	GPIB 전용
레벨	고수준 API (Application Layer)	저수준 드라이버 (Driver Layer)
표준화	VXIplug&play/IVI 재단에서 표준화	NI 전용
사용 용도	다양한 장비를 공통 API로 제어	GPIB 장비 제어 전용
예시 API	<code>viOpen()</code> , <code>viWrite()</code> , <code>viRead()</code> 등	GPIB 특정 함수 (e.g., <code>ibwrt</code> , <code>ibrd</code> , <code>ibdev</code>)

주요 차이점 설명

1. 범용성 vs. 특수성

VISA는 범용 계측기 제어 라이브러리. 다양한 통신 인터페이스를 하나의 API 집합으로 추상화 하여 장비를 제어

NI-488.2는 GPIB 장비만을 위한 저수준 드라이버입니다. VISA보다 더 하드웨어에 밀접하게 작동

2. 계층 차이 NI-488.2는 VISA 아래의 레이어에서 작동. 즉, VISA가 NI-488.2를 내부적으로 호출하여 GPIB 장비를 제어할 수 있음. 사용자는 VISA만 사용해도 GPIB 장비를 제어할 수 있지만, 특정 성능 또는 디버깅 목적에는 NI-488.2 직접 호출이 필요할 수 있음

3. 소스 코드/사용 편의성 VISA는 viRead, viWrite 등 직관적인 명령어를 제공하며, IVI 드라이버와 함께 사용 가능. NI-488.2는 ibrd, ibwrt, ibdev 같은 저수준 명령어를 사용.

선택 가이드

상황	추천
여러 종류의 장비 (USB, RS-232, GPIB 등)를 통합 제어	VISA 사용
GPIB 장비만 다루고, 성능/디버깅이 중요한 경우	NI-488.2 직접 사용 고려
LabVIEW 또는 IVI 드라이버와 함께 사용	VISA 기반 통신 권장

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Ivi.Visa;

namespace Demo_GPIB
{
    public partial class Form1 : Form
```

using Ivi.Visa; 입력
참조된 라이브러리를 클래스에 등록

VISA기반 c#에서 keithley electrometer 전류 읽는 코드

1. NI-VISA 설치

NI-VISA 다운로드 (National Instruments)

2. NI VISA .NET 어셈블리 참조 추가

- NationalInstruments.VisaNS.dll (구버전) 또는 Ivi.Visa.Interop (신버전)

Visual Studio에서 참조에 추가

SCPI 명령어	설명
*RST	장비 리셋
FUNC 'CURR'	전류 측정 모드 설정
:INIT	측정 시작
:FETCH? 또는 :READ?	측정값 읽기

💡 사용 장비가 Keithley 6517B라면 SCPI 명령어는 위와 거의 유사합니다. 모델에 따라 :MEAS:CURR? 와 같은 간단 명령어도 사용할 수 있습니다.

```

using System;
using NationalInstruments.VisaNS;

namespace KeithleyCurrentRead
{
    class Program
    {
        static void Main(string[] args)
        {
            string resourceString = "GPIB0::22::INSTR"; // 장비 주소 확인 후 변경
            try
            {
                using (MessageBasedSession mbSession =
                    (MessageBasedSession)ResourceManager.GetLocalManager().Open(resourceString))
                {
                    // 통신 설정 (선택 사항)
                    mbSession.Timeout = 5000; // 5초 timeout

                    // 장비 초기화
                    mbSession.Write("*RST"); // 장비 리셋
                    mbSession.Write("*CLS"); // 상태 클리어

                    // 측정 설정 - 전류 측정 모드로 설정
                    mbSession.Write("FUNC 'CURR'");

                    // 측정 트리거
                    mbSession.Write("INIT");
                    mbSession.Write("FETCH?"); // 또는 :READ? 또는 :MEAS:CURR? 사용

                    string response = mbSession.ReadString();

                    Console.WriteLine("Current (A): " + response.Trim());
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine("VISA Communication Error: " + ex.Message);
            }
        }
    }
}

```