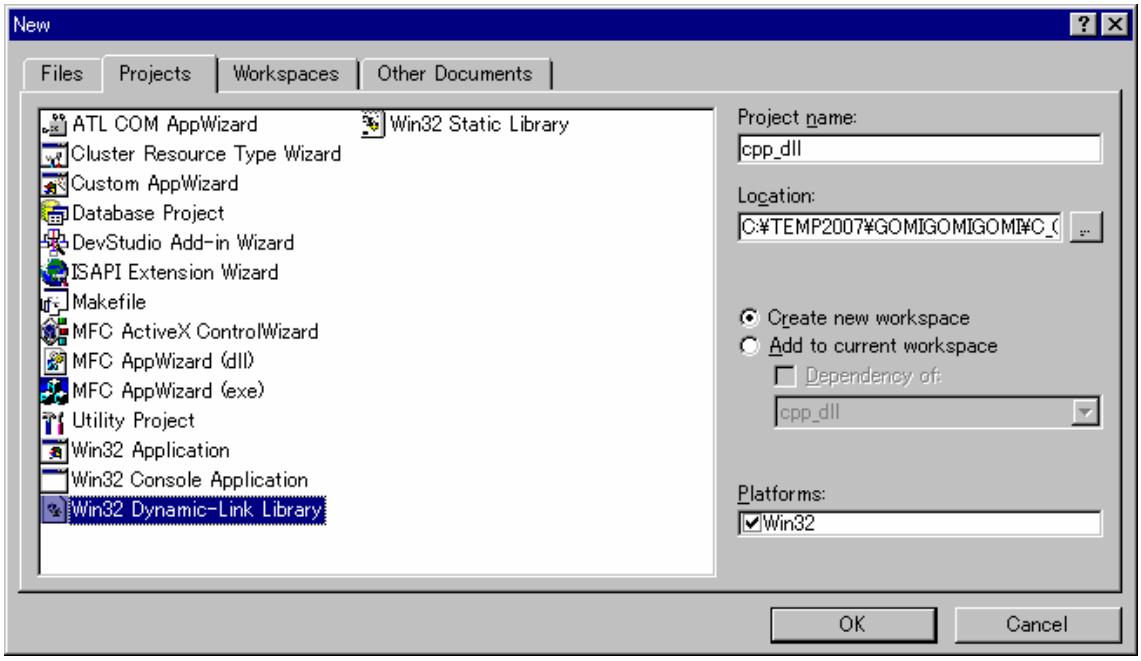
Microsoft Visual C++で作成した DLL を TestStand からコールする方法

このドキュメントでは Microsoft C++ を使用した、計測器制御（GPIB）とアナログ電圧収録(DAQ)を行う簡単なコードを紹介します。更にそのコードをビルドして DLL を作成し、TestStand からコールするための方法を紹介します。

作成プロジェクト

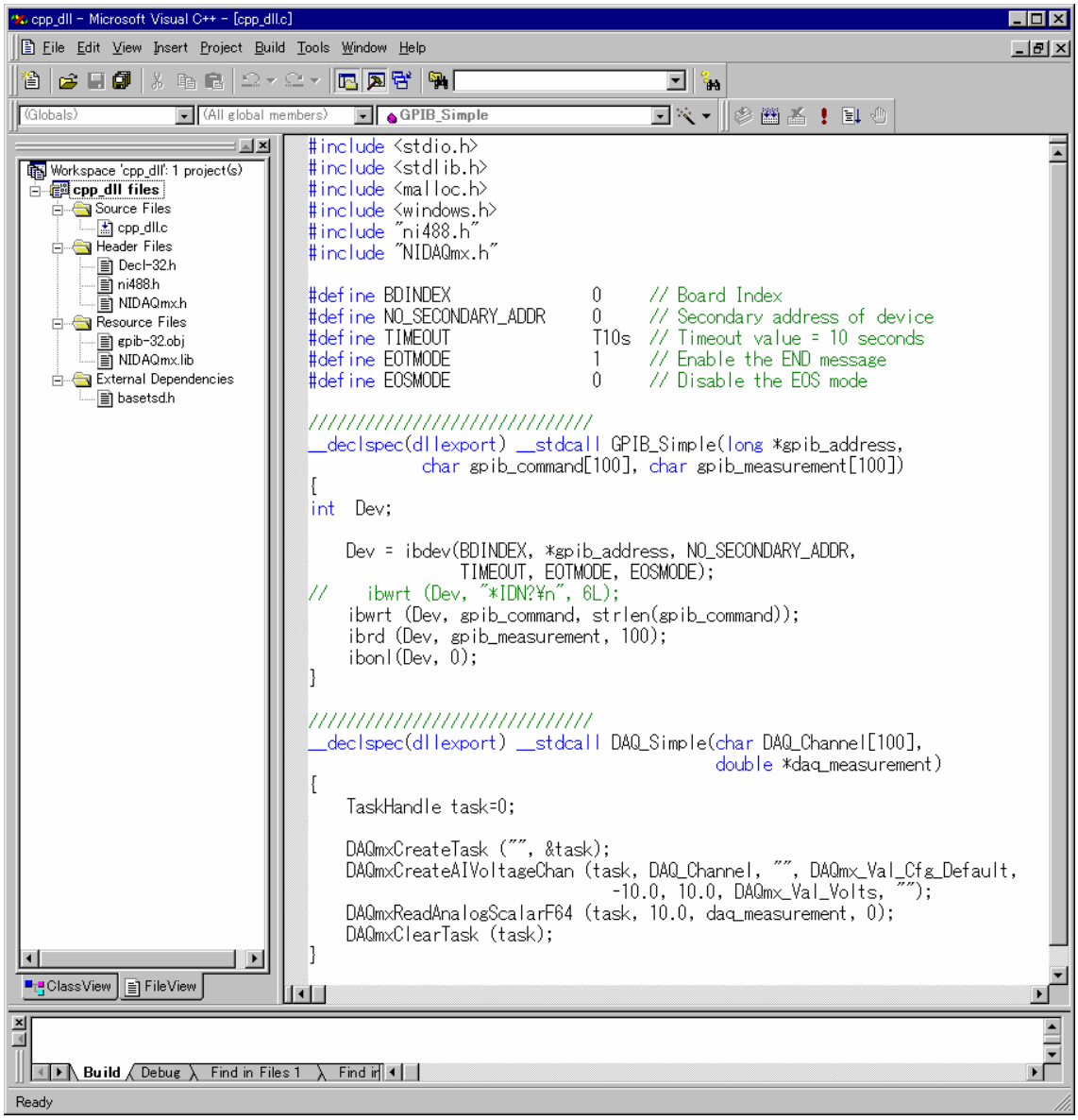


cpp_dll.c コード

```
//////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <windows.h>
#include "ni488.h"
#include "NIDAQmx.h"

#define BDINDEX                 0       // Board Index
#define NO_SECONDARY_ADDR       0       // Secondary address of device
#define TIMEOUT                 T10s    // Timeout value = 10 seconds
#define EOTMODE                 1       // Enable the END message
```

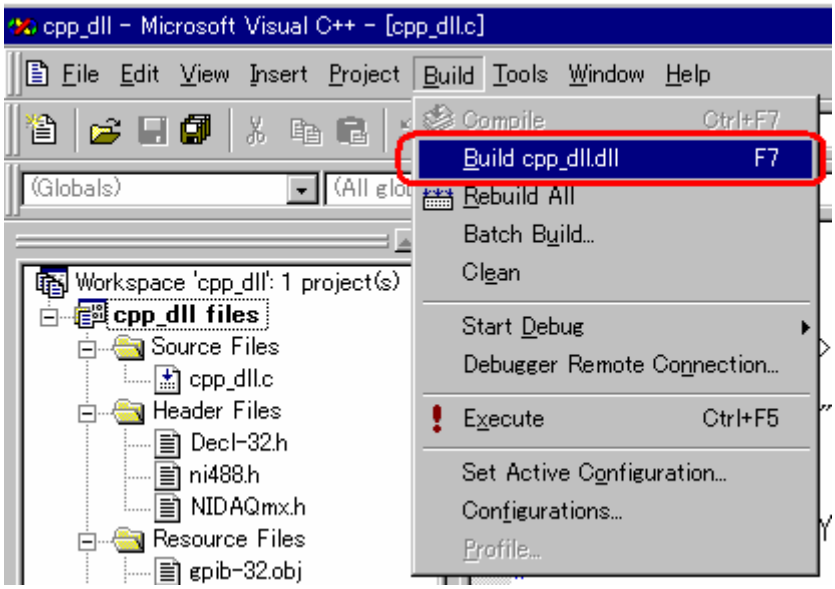
```
#define EOSMODE                     0       // Disable the EOS mode

//////////////////////////////
_declspec(dllexport) __stdcall GPIB_Simple(long *gpib_address,
                              char gpib_command[100], char gpib_measurement[100])
{
int   Dev;

      Dev = ibdev(BDINDEX, *gpib_address, NO_SECONDARY_ADDR,
                  TIMEOUT, EOTMODE, EOSMODE);
//       ibwrt (Dev, "*IDN? \n", 6L);
      ibwrt (Dev, gpib_command, strlen(gpib_command));
      ibrd (Dev, gpib_measurement, 100);
      ibonl(Dev, 0);
}

//////////////////////////////
_declspec(dllexport) __stdcall DAQ_Simple(char DAQ_Channel[100], double *daq_measurement)
{
          TaskHandle task=0;

          DAQmxCreateTask ("", &task);
          DAQmxCreateAIVoltageChan (task, DAQ_Channel, "", DAQmx_Val_Cfg_Default,
                                    -10.0, 10.0, DAQmx_Val_Volts, "");
          DAQmxReadAnalogScalarF64 (task, 10.0, daq_measurement, 0);
          DAQmxClearTask (task);
}
```

Microsoft Visual C++ プロジェクトウィンドウ

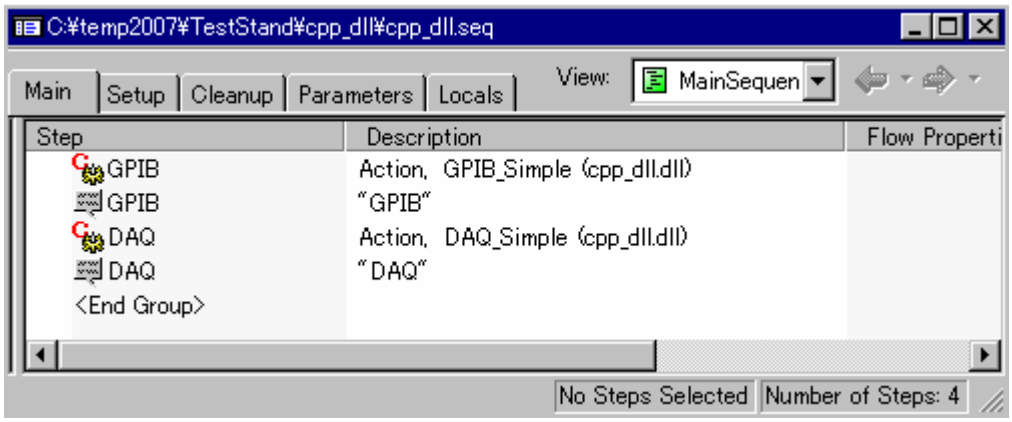



```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <windows.h>
#include "ni488.h"
#include "NIDAQmx.h"

#define BDINDEX              0     // Board Index
#define NO_SECONDARY_ADDR    0     // Secondary address of device
#define TIMEOUT              T10s  // Timeout value = 10 seconds
#define EOTMODE              1     // Enable the END message
#define EOSMODE              0     // Disable the EOS mode

//////////////////////////////
__declspec(dllexport) __stdcall GPIB_Simple(long *gpib_address,
           char gpib_command[100], char gpib_measurement[100])
{
int  Dev;

    Dev = ibdev(BDINDEX, *gpib_address, NO_SECONDARY_ADDR,
                TIMEOUT, EOTMODE, EOSMODE);
//    ibwrt (Dev, "*IDN?¥n", 6L);
    ibwrt (Dev, gpib_command, strlen(gpib_command));
    ibrd (Dev, gpib_measurement, 100);
    ibonl(Dev, 0);
}

//////////////////////////////
__declspec(dllexport) __stdcall DAQ_Simple(char DAQ_Channel[100],
                                           double *daq_measurement)
{
    TaskHandle task=0;

    DAQmxCreateTask ("", &task);
    DAQmxCreateAIVoltageChan (task, DAQ_Channel, "", DAQmx_Val_Cfg_Default,
                              -10.0, 10.0, DAQmx_Val_Volts, "");
    DAQmxReadAnalogScalarF64 (task, 10.0, daq_measurement, 0);
    DAQmxClearTask (task);
}
```

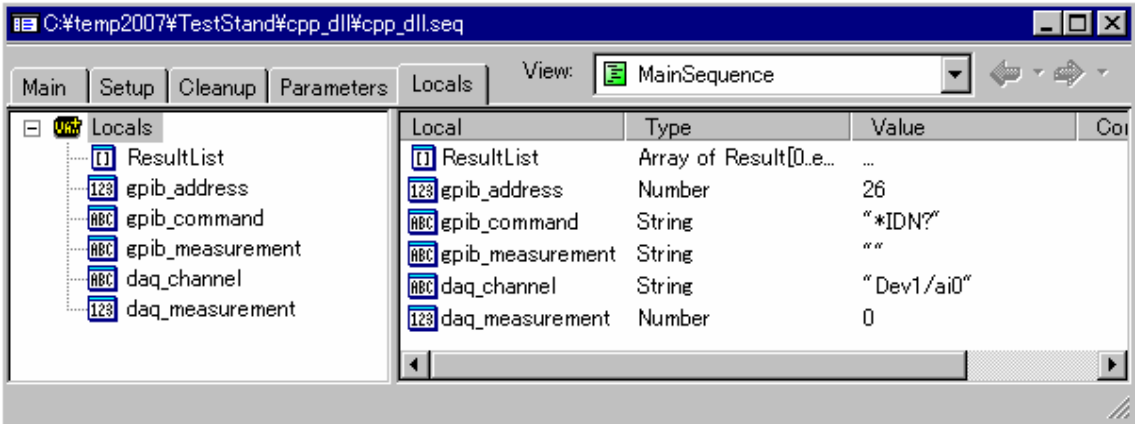
DLL ファイルをビルドするための設定



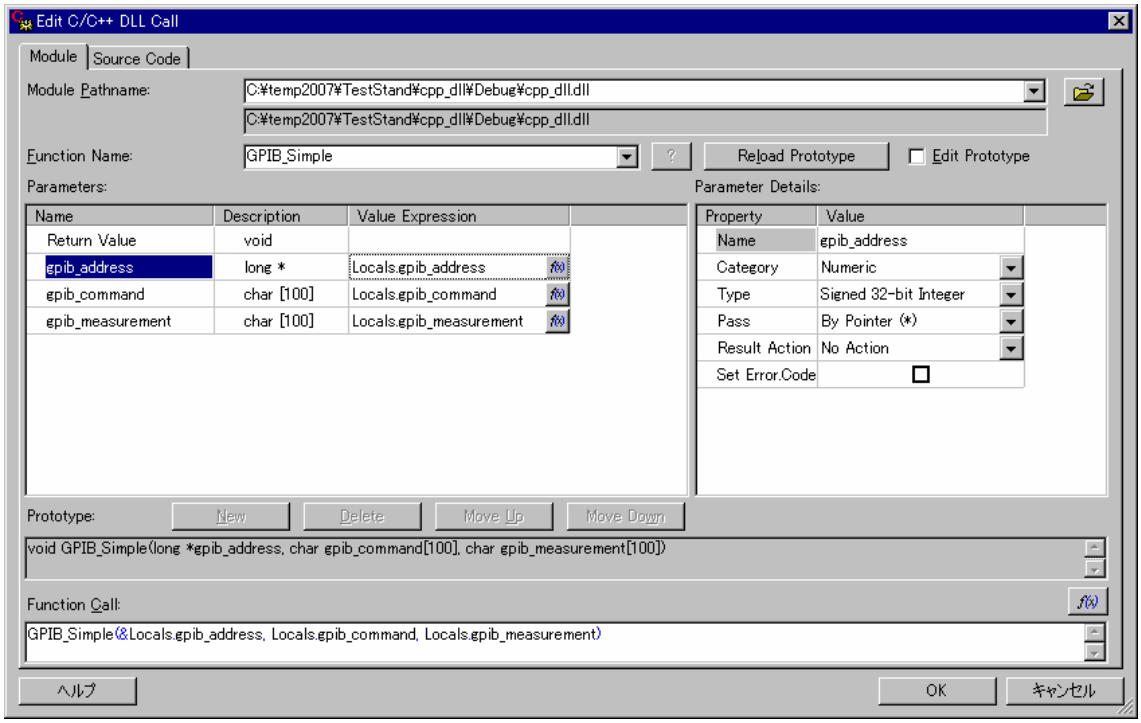
TestStand シーケンスファイル メインステップウィンドウ



TestStand シーケンスファイル ローカル変数ウィンドウ



計測器制御（GPIB）ステップ



アナログ電圧収録（DAQ）ステップ