

VisualBasic VisualC#による  
計測・統計機能を用いた  
GO-NOGO 判定サンプル  
プログラムガイド



## 内容

1. はじめに.....	3
2. 制御プログラム作成の準備.....	3
2.1. ソフトウェア環境.....	3
2.2. ハードウェア環境.....	4
2.3. 測定器の接続とVISA アドレスの確認.....	4
3. 参照ライブラリの追加.....	6
3.1. Visual Basic の場合.....	6
3.2. Visual C#の場合.....	9
4. サンプルプログラム.....	11
4.1. 機能.....	11
4.2. 操作方法.....	12
4.3. コードの表示.....	13
4.4. 使用するコマンドとクエリ.....	14
4.5. サンプルプログラムのポイント.....	15
4.5.1. セットアップファイルの読み込みと適用.....	15
4.5.2. セットアップのファイル保存.....	15
4.5.3. 自動計測の開始と終了、結果の確認.....	16
5. 測定器プログラムのサポート.....	17

## 1. はじめに

「Visual Basic Visual C#によるオシロスコープ制御入門」では、PCと測定器をLAN やUSB 等の汎用インターフェースで接続して制御する方法を、サンプルプログラムを通して紹介しました。このテキストでは、オシロスコープの計測・統計機能を用いた GO-NOGO 判定の実装を例に、測定器を制御する方法についてご案内いたします。

測定器のコマンドや使用するライブラリについては、「Visual Basic Visual C#によるオシロスコープ制御入門」を参照してください。

また、作成するプログラムでは測定器の統計機能を用います。統計機能がない測定器DSOX/MSOX2000 は、当プログラムでは想定されない動作やエラーを起こすことがありますので使用しないでください。

[ダウンロードサイト]

<http://www.agilent.co.jp/find/InfiniiVision-sample>

## 2. 制御プログラム作成の準備

### 2.1. ソフトウェア環境

本文章の内容を実施するためには、マイクロソフト社のWindows OSに、以下のものがインストールされている環境が必要です。これらは、インターネット上でも入手・購入することが可能です。なお、Agilent IO Libraries Suite は、弊社測定器でなくともお使いいただけます。

- .Net Framework 2.0 以降がインストールされている Windows OS
- Visual Basic または Visual C# (Express 版でも可)
- Agilent IO Libraries Suite (以下の弊社 Web から無償でダウンロードが可能)  
[IO Libraries Suite ソフトウェアダウンロードサイト]  
<http://www.agilent.com/find/iolib>
- Sample Program (以下の弊社 Web から無償でダウンロードが可能)  
[ダウンロードサイト]  
<http://www.agilent.co.jp/find/InfiniiVision-sample>

## 2.2. ハードウェア環境

本章の内容を実施するには、PC や測定器の種類、搭載されているインターフェースにもよりますが、必要に応じて、PC と測定器を接続するためのインターフェースカードや、ケーブル類が必要となります。現在の Agilent 社の測定器には、制御用インターフェースとして、主に、USB、GPIB、LAN などが搭載されています。個別の測定器に搭載されているインターフェースに関しては、データシートやマニュアルをご確認ください。表 1 は、オシロスコープに搭載されている制御用インターフェースです。

モデル番号	USB	GPIB	LAN
DSOX/MSOX3000	○	オプション	オプション
DSO5000	○	○	○
DSO/MSO6000	○	○	○
DSO/MSO7000	○	オプション	○
DSO/MSO8000	○	○	○
DSO80000	○	○	○
DSO/MSO9000	○	オプション	○
DSO90000	○	オプション	○

表 1 Agilent 社製オシロスコープ搭載 リモート制御用インターフェース

お使いの測定器に合わせて、制御に用いるインターフェースを検討します。その際、LAN、USB であれば、一般的なインターフェースであるため、PC に搭載されている可能性が高いですが、GPIB は、測定器用インターフェースであるため、一般的な PC に搭載されていません。そのため、GPIB にて制御を実施する場合には、GPIB インターフェース、GPIB ケーブルを、別途、ご購入する必要があります。これらは、弊社にて販売しておりますので、お気軽にお問い合わせください。

## 2.3. 測定器の接続と VISA アドレスの確認

制御する測定器を指定には、対象となる測定器の VISA アドレスを使います。Agilent IO Libraries Suite の Agilent Connection Expert を使って、VISA アドレスを確認する方法を説明します。

### ① 測定器を PC と接続します。

ここでは、USB ケーブルで接続したものととして、ご説明を進めます。

LAN で接続する場合は、PC と測定器の IP アドレスの設定/確認が必要です。また、ケーブルの種類にご注意ください。**PC と直接接続する場合、通常は、クロスタイプのケーブルが必要です。**ハブ等を用いて接続した場合は、一般に使われているストレートタイプのケーブルを使用することができます。

### ② Agilent Connection Expert を起動します。

スタートボタンから [スタートボタン]>[Agilent IO Libraries Suite]>[Agilent Connection Expert]を実行します( 図 1)。

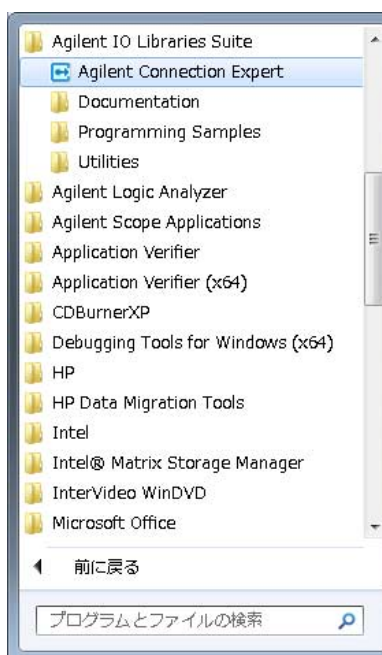


図 1 Connection Expert の起動

- ③ ケーブルで接続されている測定器は、モデル名の横に、緑色のチェック印が示されます。もし、「×」、「！」の印が付いている場合は、測定器が認識されていない事を表します。原因として、測定器の電源が入っていない、ケーブルが接続されていない、測定器によっては、測定器側の設定(お使いになるインターフェースの選択)がなされていない事が考えられます。
- ④ 使用する測定器を選択し、表示される VISA アドレスを確認してください。(図 2)

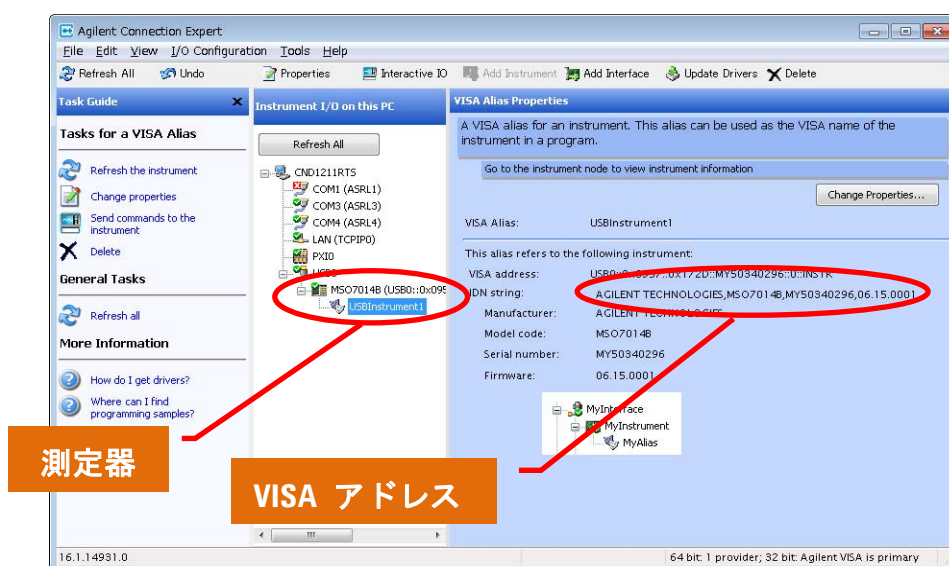


図 2 VISA アドレスの確認

### 3. 参照ライブラリの追加

プログラムから測定器と通信を行う際には、VISA-COM ライブラリを用います。ここでは開発環境毎に参照ライブラリの追加方法について説明します。このライブラリをお使いになるには、IO Libraries Suite をインストールする必要があります。

ライブラリについての説明や、簡易な通信のプログラムサンプルについては、「Visual Basic Visual C#によるオシロスコープ制御入門」を参照してください。

[ダウンロードサイト]

<http://www.agilent.co.jp/find/InfinitiVision-sample>

#### 3.1. Visual Basic の場合

- ① プロジェクトのプロパティを開きます。

ソリューションエクスプローラの My Project をダブルクリックするか(図 3)、メニューバーから [プロジェクト]>[プロパティ (P)]をクリックしてください。

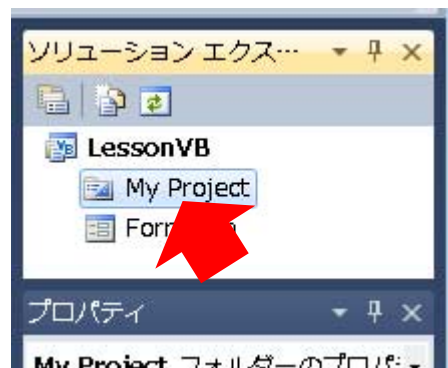


図 3 プロジェクトのプロパティウィンドウを開く(Visual Basic)

② 参照タブの追加をクリックします。(図 4)

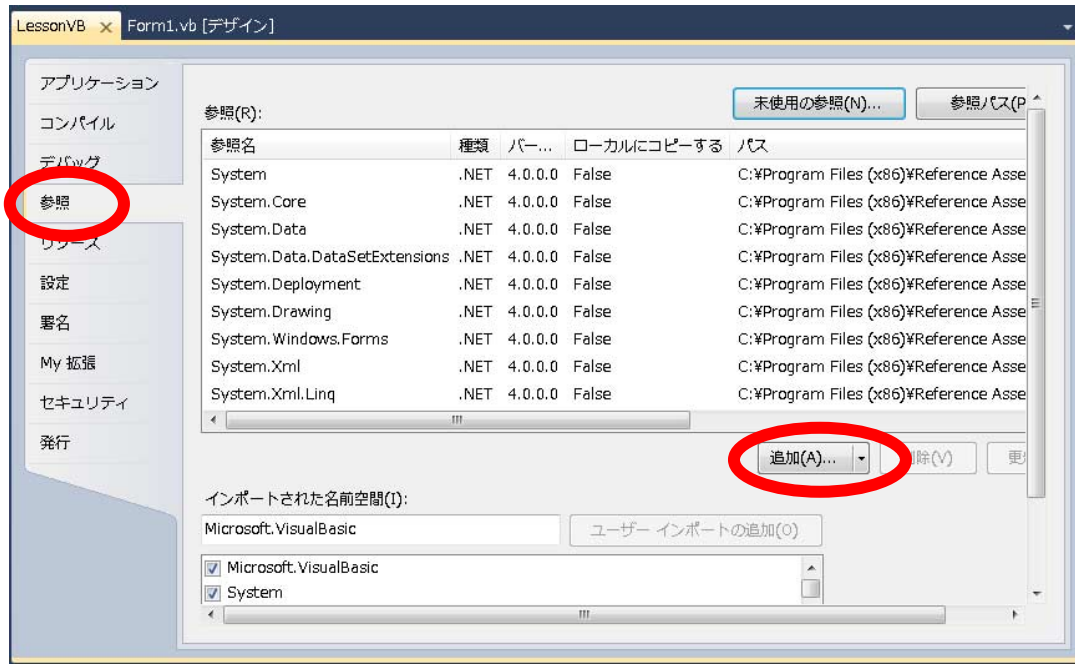


図 4 参照の追加(Visual Basic)

③ COM ライブラリから VISA COM 3.0 Library を選択して OK ボタンを押します。(図 5)

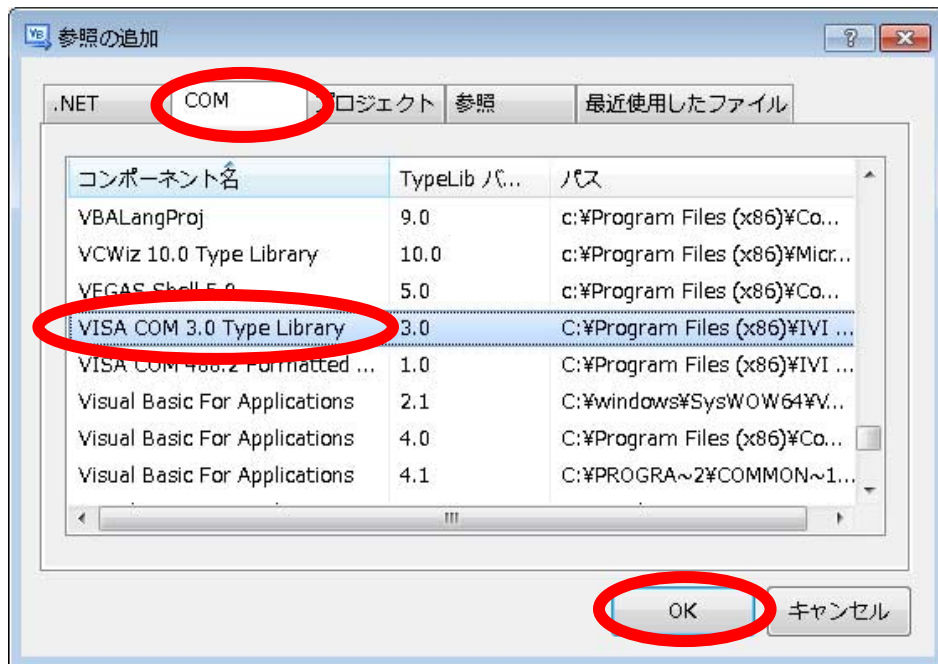


図 5 VISA COM 3.0 Library の追加(Visual Basic)

- ④ インポートされた名前空間の Ivi.Visa.Interop にチェックを入れます。(図 6)



図 6 名前空間の追加(Visual Basic)



### 3.2. Visual C#の場合

- ⑤ プロジェクトのプロパティを開きます。

ソリューションエクスプローラのプロジェクトから参照設定を右クリックして[参照の追加(R)]をクリックするか(図 7)、メニューバーから[プロジェクト]>[参照の追加 (R)]をクリックしてください。

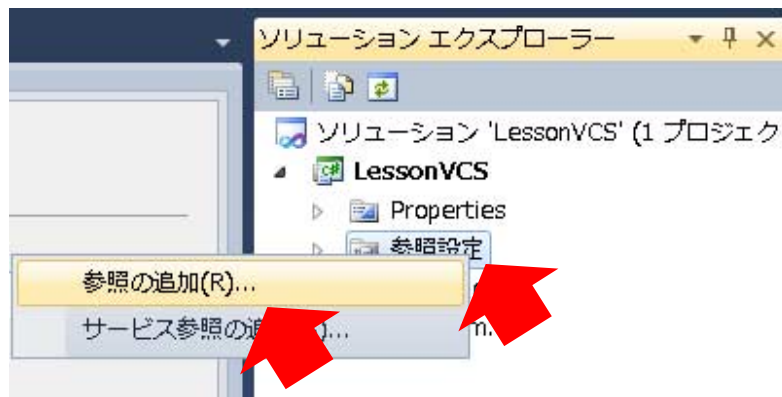


図 7 プロジェクトのプロパティウィンドウを開く(Visual C#)

- ⑥ COM ライブラリから VISA COM 3.0 Library を選択して OK ボタンを押します。(図 8)

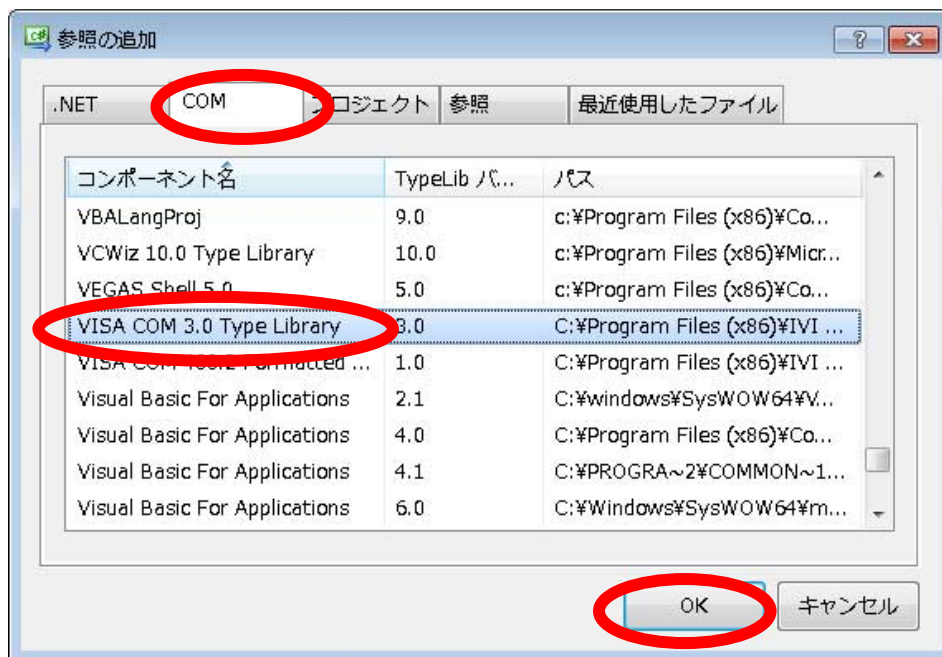


図 8 VISA COM 3.0 Library の追加(Visual C#)

- ⑦ ソリューション エクスプローラの Form1.cs を右クリックし、コードの表示を選択します。(図 9)

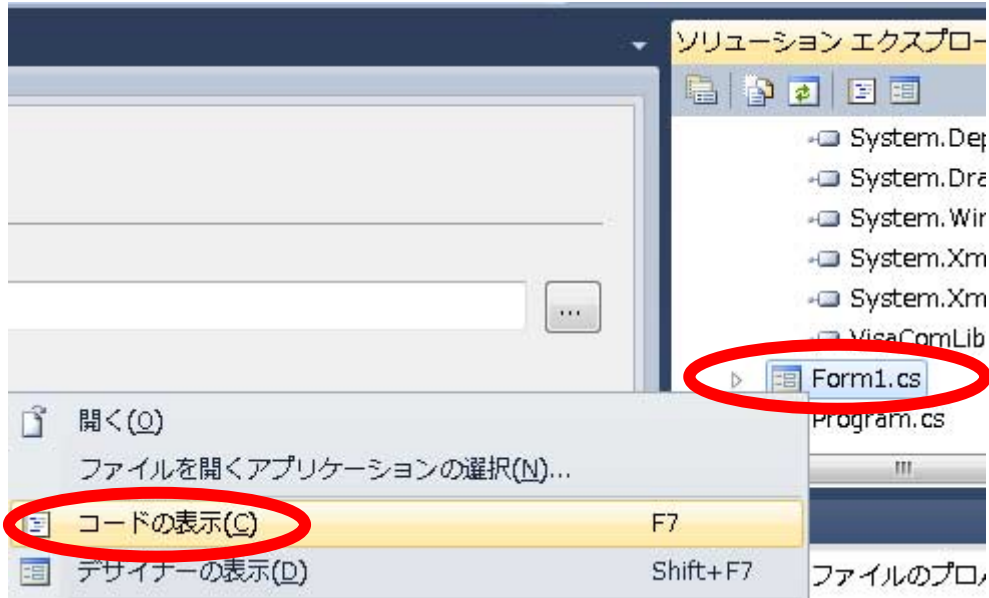


図 9 コードの表示

- ⑧ インポートする名前空間に「using Ivi.Visa.Interop;」を追加します。(図 10)

```
LessonVCS.Form1
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Windows.Forms;
9  using Ivi.Visa.Interop;
10
11 namespace LessonVCS
12 {
13     public partial class Form1 : Form
14     {
```

図 10 名前空間の追加(Visual C#)

## 4. サンプルプログラム

ここからは、Agilent 製オシロスコープ (InfiniiVision シリーズ) のサンプルプログラムについて説明します。InfiniiVision シリーズ オシロスコープのサンプルプログラムは、下記よりダウンロードが可能です。測定器のプログラム方法やライブラリの利用、大まかな流れについては「Visual Basic Visual C#によるオシロスコープ制御入門」を参照してください。同じダウンロードサイトより入手することができます。

[ダウンロードサイト]

<http://www.agilent.co.jp/find/InfiniiVision-sample>

※ InfiniiVision シリーズ

DSOX/MSOX2000,3000、DSO5000、DSO/MSO6000、DSO/MSO7000 の各シリーズ

### 4.1. 機能

サンプルプログラムそのまま実行してご利用いただけます。以下のことが可能です。

- 測定器との接続確認
- セットアップファイルの取得と、オシロスコープへのセットアップファイルの送信
- 指定したテスト回数に達するまでの自動計測機能の実行
- テストの判定方法と判定値の選択
- 合否判定と測定値・スクリーンショットの表示、ファイル出力
- リセット、エラー取得

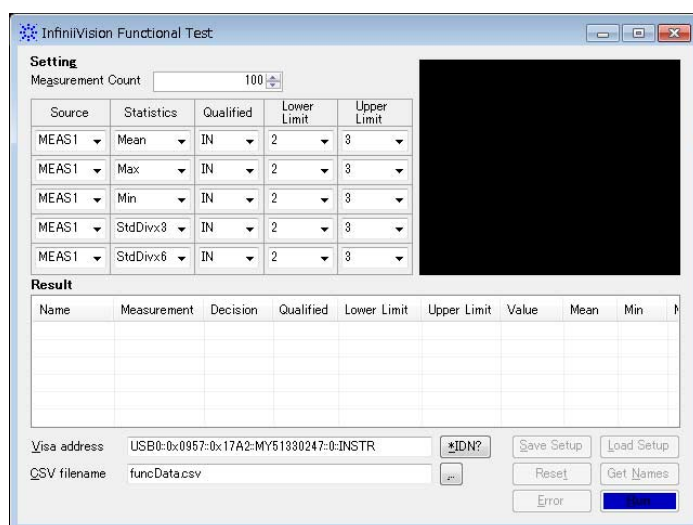


図 11 サンプルプログラム

## 4.2. 操作方法

解凍したフォルダより、以下を実施します。

(ア) Visual Basic の場合

¥InfiniiVisionSampleVB¥bin¥Release¥InfiniiVisionFunctionalTestVB.exe

(イ) Visual C#の場合

¥InfiniiVisionSampleVCS¥bin¥Release¥InfiniiVisionFunctionalTestVCS.exe

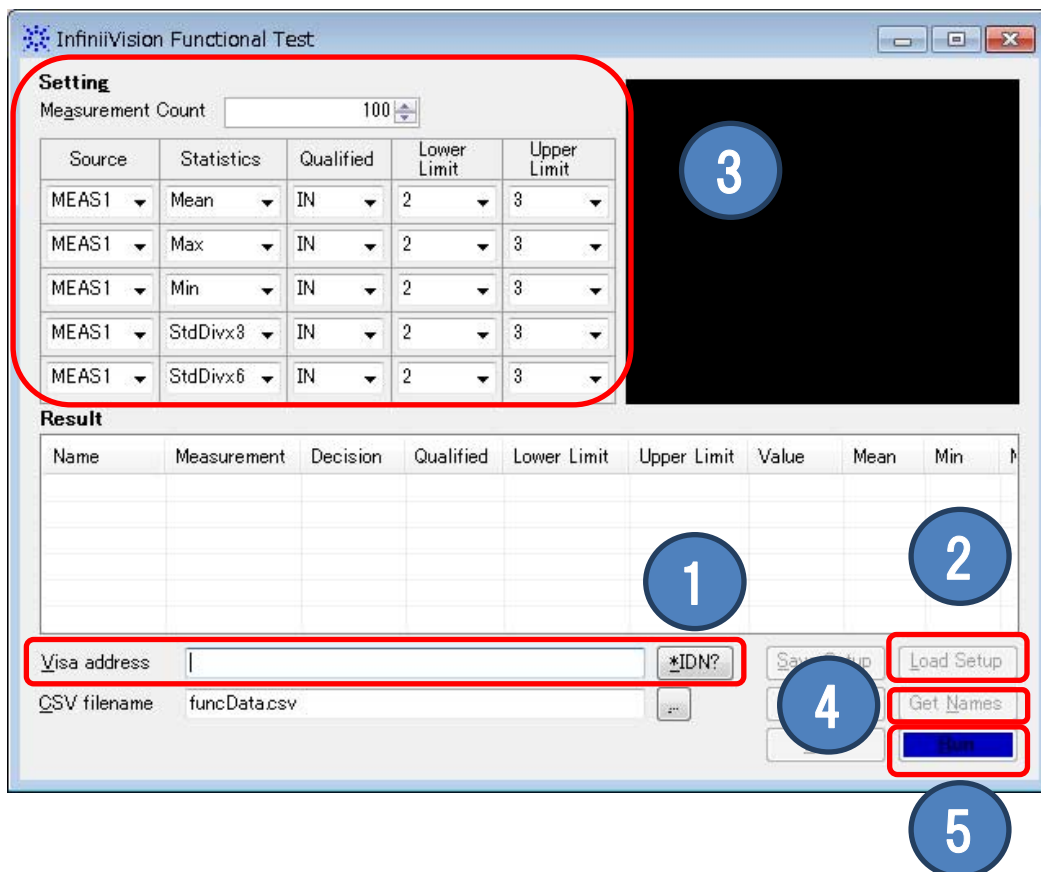


図 12 サンプルプログラムの操作方法

- ① VISA Address 欄に、制御するオシロスコープの VISA アドレスを入力し、\*IDN?ボタンを押します。オシロスコープの VISA アドレスは、前述の Agilent Connection Expert にて確認できます。  
例: USB0::0x0957::0x17A2::MY50500003::0::INSTR  
\*IDN?ボタンを押すと、測定器のモデル名が表示されます。表示されない場合には、VISA アドレスを再確認してください。また、Agilent Connection Expert にて、測定器が接続されているか確認してください。

- ② [Csv filename]に測定結果を保存するファイルパスを入力します。ダイアログを表示して入力する場合は[...]ボタンを押してください。
- ③ 必要であれば、セットアップファイルを適用します。[Load Setup]ボタンを押した後、表示されるダイアログより適用するセットアップファイルを選択してください。  
セットアップファイルをお持ちでない場合は、測定器を操作して測定器内に保存されているセットアップファイルをご利用いただくか、または、測定器を手動で設定後、[Save Setup]ボタンを押下してセットアップファイルを作成することができます。
- ④ テストの判定方法を設定してください。  
各設定内容は以下の通りです。
- Measurement Count ... 測定回数
  - Source ... 測定対象の選択。MEAS1-5 は測定 1-5 に対応。
  - Statistics ... 判定対象の統計項目。
  - Qualified ... 判定方法の選択。
  - Lower Limit, Upper Limit ... 閾値の設定。
- ⑤ 必要であれば、測定対象の確認を行ってください。[Get Names]ボタンを押すと測定対象の名前がリストに表示されます。目的の測定対象がリストに表示されない場合は、測定器の Measure の設定や④の設定を見直してください。
- ⑥ RUN ボタンを押すと、テストが開始されます。  
テスト終了後、オシロスコープのスクリーン画像が表示され、テスト結果と詳細が表示されます。スクリーン画像とテスト結果は[Csv filename]で指定したファイルに追加保存されます。  
入力されている波形を測定できない場合や測定間隔が大きすぎる場合、オシロスコープに波形が表示されないことや、テストが終了しないこと、プログラムが停止することがあります。適切な設定に変更して再度実施してください。テストが終了しない場合は、測定器を手動で操作し、プログラムを停止するなどしてテストを終了してください。

#### **4.3. コードの表示**

サンプルプログラムのコードを表示する場合には、プログラミングソフトを起動し、[ファイル] > [プロジェクトを開く] から、以下のプロジェクトファイルを開いてください。

- Visual Basic の場合  
¥InfiniiVisionFunctionalTestVB.vbproj
- Visual C#の場合  
¥InfiniiVisionFunctionalTestVCS.csproj

#### 4.4. 使用するコマンドとクエリ

このサンプルプログラムには、表 2 のコマンドまたはクエリが使われています。コマンドやクエリの概要や送受信方法については、「Visual Basic Visual C#によるオシロスコープ制御入門」を参照してください。各コマンド・クエリの詳細に関しましては、各測定器の Programmer's Guide をご参照ください。

※Programmer's Guide は英語版ガイドのみのご提供です。日本語版はありません。

[Visual Basic Visual C#によるオシロスコープ制御入門のダウンロードサイト]

<http://www.agilent.co.jp/find/InfiniiVision-sample>

[InfiniiVision Oscilloscopes Programmer's Guide]

DSOX/MSOX3000 Series Oscilloscopes Programmer's Guide

[http://www.home.agilent.com/upload/cmc\\_upload/All/3000\\_series\\_prog\\_guide.pdf](http://www.home.agilent.com/upload/cmc_upload/All/3000_series_prog_guide.pdf)

DSO5000 Series Oscilloscopes Programmer's Guide

[http://www.home.agilent.com/upload/cmc\\_upload/All/5000\\_series\\_prog\\_guide.pdf](http://www.home.agilent.com/upload/cmc_upload/All/5000_series_prog_guide.pdf)

DSO/MSO7000 Series Oscilloscopes Programmer's Guide

[http://www.home.agilent.com/upload/cmc\\_upload/All/7000B\\_series\\_prog\\_guide.pdf](http://www.home.agilent.com/upload/cmc_upload/All/7000B_series_prog_guide.pdf)

コマンド	解説
*IDN?	測定器の ID 取得
*RST	リセット状態にする。
:SYSTem:ERRor?	エラー・キュー内容取得。
:SYSTem:SETup?	セットアップファイルの取得。(ブロック転送)
:SYSTem:SETup	セットアップファイルの送信。(ブロック転送)
:MEASure:STATistics	:MEASure:RESults? で取得するデータを制御する。
:MEASure:RESults?	設定した測定機能の統計値を取得する。
:DISPlay:PERsistance INFinite	画面を無限残光モードにする。
:MEASure:STATistics:MCOunt <n>	<n>で指定した測定回数で測定の表示を止める。
:STOP	更新停止。
:MEASure:STATistics:RESet	測定をリセットする
:RUN	測定・テストの開始
:DISPlay:DATA? BMP,COLor	画面の画像データの取得。(ブロック転送)
:SAVE:IMAGE:INKSaver ON	取得する画像データを白黒反転する。
:HARDcopy:LAYout PORTrait	取得する画像を標準の状態にする。

表 2 使用コマンド一覧

## 4.5. サンプルプログラムのポイント

このサンプルプログラムのポイントについて解説します。

- ・ セットアップファイルの読み込みと適用
- ・ セットアップファイル保存
- ・ 自動計測の開始と終了、結果の確認。

### 4.5.1. セットアップファイルの読み込みと適用

サンプルプログラムでは、MainForm.cs中の `void LoadSetupFile(String filename)` 関数に実装されています。

測定器にセットアップファイルを適用するには、セットアップファイルを:SYSTem:SETupコマンドでブロック転送する必要があります。そのために、サンプルプログラムではVISA-COMライブラリのAPIで

`FormattedIO488. WriteIEEEBlock (String Command, object Data)`

を使います。この時、第1引数は、コマンドの文字列 `":SYSTem:SETup"`、第2引数にByte型の配列でセットアップファイルのデータを渡します。第2引数の型に合わせてセットアップファイルのデータを用意するには、バイナリデータとしてセットアップファイルを読み出します。

バイナリデータとしてセットアップファイルを読み出すには、.net Frameworkの `System.IO.BinaryReader`を使います。ファイルのサイズを `BinaryReader` オブジェクトから取得し、その大きさのByte型の配列を用意して、ファイルのデータを取得します。

ブロック転送の際、データ量が大きいと通信がタイムアウトすることがあります。これは、タイムアウトプロパティの値を十分大きくとっておくことで回避できます。

### 4.5.2. セットアップのファイル保存

サンプルプログラムでは、MainForm.cs中の `void SaveSetupFile(String filename)` 関数に実装されています。

測定器からのセットアップファイルの取得は、:SYSTem:SETup? クエリの返信データとして取得することができます。取得したデータはセットアップファイルの形式になっているので、そのままファイルに出力することで、セットアップファイルを得ることができます。

クエリの返信データはブロック受信で取得できます。これには、VISA-COMライブラリのAPIの

`FormattedIO488. ReadIEEEBlock (IEEEBinaryType.BinaryType_UI1)`

を使います。この関数の返値をByte型の配列で受け、バイナリとしてファイルに出力します。

ブロック受信の際、データ量が大きいと通信がタイムアウトすることがあります。これは、タイムアウトプロパティの値を十分大きくとっておくことで回避することができます。サンプルプログラムでは10秒としています。

バイナリでのファイル出力は、.net Frameworkの `System.IO.BinaryWriter.Write`を使います。

### 4.5.3. 自動計測の開始と終了、結果の確認

テストの状態は測定のカウントをプログラムで監視することで推定します。測定のカウントは“:MEASure:RESults?” クエリで取得できる文字列から取り出します。このクエリですべての統計値を取得できるよう、このクエリの前に“MEASure:STATistics 1”のコマンドを用いています。この状態でクエリによって取得できる項目は、図 13 のレコードが設定された測定の順に連続したものとなります。

Measurement label	current	min	max	mean	std dev	count
-------------------	---------	-----	-----	------	---------	-------

図 13 :MEASure:RESults?のレスポンスの順番

- Measurement label ... 測定の種類
- current ... 表示されている波形での測定値
- min ... 最小値
- max ... 最大値
- mean ... 平均値
- std dev ... 標準偏差
- count ... 測定回数

サンプルプログラムでは、測定が更新され続けていることを確認するために、先に取得した測定回数の最小値と新たに取得した測定回数の最小値を比較しています。

クエリを繰り返し送る場合は、十分に時間を空けて行ってください。クエリが連続して送られると、測定器が通信を十分に処理できないことがあります。サンプルプログラムでは、0.5 秒間プログラムをスリープすることで時間を取っています。

また、プログラムの処理が長時間終了しない場合、GUI が更新されず「応答なし」と表示されることがあります。これは、内部の処理がシングルスレッドで作成されている場合にみられる現象です。これを避けるには、時間のかかる処理をマルチスレッドで実装するか、処理中に GUI を更新する必要があります。

マルチスレッドプログラミングについては、この文章では扱いません。一般の書籍または web 等の情報を参照してください。

処理中にGUIを更新するには、System.Windows.Forms.Application.DoEvents()を実行します。これによって、アプリケーションのメッセージキューが処理されます。サンプルプログラムでは、テストの開始や終了を待つ部分で実行しています。この方法は簡易に実装できますが、マルチスレッドプログラミングに比べてGUIが更新されにくくなることやハードウェアリソースを多く消費することがあります。



## 5. 測定器プログラムのサポート

このガイドやサンプルプログラムに関しては、下記まで、お問い合わせください。

アジレント・テクノロジー株式会社

電子計測本部・計測お客様窓口

フリーダイヤル: 0120-421-345

Email : [contact\\_japan@agilent.com](mailto:contact_japan@agilent.com)

計測お客様窓口では、このガイドに対する質問、サンプルプログラムについてのインストール方法、実行方法、プログラム内容等のお問い合わせを承っております。お問い合わせの際は、Windows のバージョン、言語のバージョン、ご利用の測定器モデル名、具体的なお問い合わせ内容をお知らせ下さい。その他、Agilent 社製測定器のコマンドや動作に関する説明、他社製測定器から弊社測定器へのポーティング、IO Libraries Suite による測定器の制御など、弊社製品に関してのご質問、ご相談も承っております。

新規プログラム作成のご依頼や、サンプルプログラムのカスタマイズのご依頼、お客様が作成・カスタマイズなされたプログラムのデバッグ作業は承っておりません。

Visual Studio 2010 Express Edition の使い方や文法など、プログラム開発環境やプログラム言語の一般的なご質問に関しましては、サポートしておりません。一般の書籍または web、Help 等の情報を、ご確認いただきますようお願いいたします。

### 【著作権および免責事項】

著作権はアジレント・テクノロジー株式会社が保有しています。このガイド、ソフトウェアを使用したことによって生じたすべての障害・損害・不具合等(含、ソフトウェアのバグ)に関しては、弊社および弊社の所属するいかなる団体・組織とも、一切の責任を負いません。各自の責任においてご使用ください。

### 【転載条件】

無断での転載・配布はご遠慮下さい。

アジレント・テクノロジー株式会社

本社〒192-8510 東京都八王子市高倉町 9-1

計測お客様窓口

受付時間 9:00-18:00(土・日・祭日を除く)

TEL ■■ 0120-421-345

(042-656-7832)

FAX ■■ 0120-421-678

(042-656-7840)

Email [contact\\_japan@agilent.com](mailto:contact_japan@agilent.com)

電子計測ホームページ

[www.agilent.co.jp](http://www.agilent.co.jp)

記載事項は変更になる場合があります。  
ご発注の際にご確認ください。

©Agilent Technologies, Inc. 2012

Published in Japan, June 26, 2012  
5991-0788JAJP  
0000-08A



Agilent Technologies