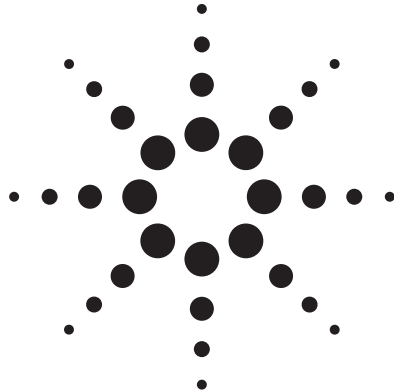


# Agilent E6601A 無線通信テスト・セット

製造ラインにも適した  
革新的なプログラミング手法

Application Note



## 目次

はじめに.....	2
<b>E6601Aテスト・セットの概要 .....</b>	<b>3</b>
自動化を容易にする各種機能.....	4
テスト・プログラムの開発.....	5
プログラミングの詳細.....	8
<b>E6601AのPCコントローラへのテスト・プログラムの移行.....</b>	<b>17</b>
まとめ.....	17
<b>付録：GSM用のサンプル・テスト・プログラム .....</b>	<b>18</b>

## はじめに

最近のテスト機器には、革新的な機能が追加され、移動機メーカーの製造ラインの効率化を実現する、新しいクラスの「スマート機器」が誕生しています。その1つが、Windows® XPオペレーティング・システムを搭載した、Agilent E6601A無線通信テスト・セットです。E6601Aテスト・セットは、効率的な自動テスト・システムのコアとなるために欠かせない制御／インタフェース機能を備え、大幅なテスト・スピードの向上、テスト・システムの簡素化、テスト・コストの削減を実現しています。

このアプリケーション・ノートでは、E6601Aテスト・セットの内蔵機能を使用した自動テスト用サンプル・プログラムの開発方法を紹介します。プログラミング例は、プログラミング環境／テスト・セットのセットアップ、接続の検証、プログラミング・フローチャートの作成、テスト・プログラムの記述で構成されています。複数のプログラミング環境を使用できますが、このアプリケーション・ノートに示されているサンプル・プログラムは、Visual Studio .NET®で作成されています。

## E6601Aテスト・セット の概要

Agilent E6601Aは、新しいプラットフォームをベースにした移動機の製造に適したワンボックス・テスト・セットです。この新しいプラットフォームの重要な特長の1つが、Windows XP Professional<sup>®</sup> オペレーティング・システムの搭載です。テスト・エンジニアやプログラマは、テスト・セットに接続されたキーボード、マウス、モニタを使用して、使い慣れたコンピュータ環境で作業が行えます。Windowsプログラムはすべて、このテスト・セットで、コード変換なしで実行できます。さらに、リモート・アクセスなどのWindowsプラットフォームの機能もフルに活用できます。

この他にも、PC内蔵のテスト・セットには以下のような利点があります。

- **スペースの節約**：製造ラインでは、常にPC／モニタの配置に悩まされます。ラックに十分なスペースがあることはほとんどありません。PCとモニタをテスト・セットに統合することにより、E6601はこうした問題を取り除いています。フロント・パネルは、自動化機能、マウス、標準的なフロントパネル・ボタンで制御できます。
- **配備／保守が簡単**：一体型ソリューションなので、移動や配備が簡単です。外部PCやアクセサリが不要なので、製造マネージャは、新しい製造ラインをより迅速に再編成／構築できます。このテスト・セットは保守がさらに容易で、気がかりな部品や接続部も少なくなっています。ソフトウェアのアップグレード、毎日／毎月の調整などの定期的な保守も、LANでリモートから行えます。
- **購入／運用コストの削減**：E6601Aは、PC、GPIBデバイス／アクセサリ、コンピュータ用のラック／フロア・スペース、コンピュータの維持費の削減により、コストダウンを実現します。テスト・エンジニアやプログラマは、専用のテスト開発ツールを使用する必要がないので、すぐにテストを開始できます。

## 自動化を容易にする さまざまな機能

E6601Aには、テスト・セットの操作やテスト環境の自動化を容易にする、さまざまな機能があります。最も重要な機能は、内蔵のWindows XPコンピュータで、短期間での習得、テスト・セットのリモート制御、テスト・セット間でのデータの共有、オンラインでのソフトウェア・アップデート／トラブルシューティング／サポートが可能です。

E6601Aはさまざまな接続機能も備え、システム・インテグレーションやリモート制御にも最適です。ユーザは、LAN、GPIB、USB、仮想LAN経由で通信することにより、テスト手順を制御／管理できます。

- **LAN**：LANは、長距離での接続を可能にし、複数のPCで測定器を共有することができます。新しいテスト・システムの接続には、LANが最適です。E6601Aでは、LAN経由でリモート・トラブルシューティング／モニタが可能です。
- **USB**：USBは、測定器とPCを接続するための最も簡単な方法です。通常は、測定器とPCの1対1の接続に用いられます。
- **GPIB**：GPIBは、最も古くから利用されている信頼性の高い測定器用インタフェースです。測定器の最も一般的なインタフェースであり、テスト・システムの統合に使用できます。
- **仮想LAN**：E6601Aの内蔵プログラムは、内部LANインタフェースを使用してテスト・セットを制御します。この場合、仮想LANは、内蔵PCとテスト・コンポーネントの内部的な接続に使用されます。

## テスト・プログラムの開発

テスト・プログラムの開発は、内蔵PCでも外部PCでも可能ですが、外部PCでテスト・プログラムを開発／デバッグしてから、E6601Aの内蔵PCにアップロードするのが、最も効率的です。以下の方法は、外部PCでのテスト・プログラムの開発に適用されます。テスト・セットへのプログラムのロード方法については、後のセクションで説明します。

**プログラミング環境のセットアップ**：E6601Aでは、C++、Visual Basic<sup>®</sup>、Agilent VEE Proなどの、多くのプログラミング環境を使用できます。今日では、Visual Studio<sup>®</sup> が一般的なプログラム開発ツールとなっているので、このアプリケーション・ノートでは、Visual Studio .NET環境とVisual Basicプログラミング言語が用いられています。これらのプログラミング・ツールの詳細については、Microsoft社のWebサイト (<http://msdn.microsoft.com/vstudio/>) をご覧ください。

**I/Oライブラリ・スイートのインストール**：I/Oライブラリ・スイートは、Agilentのアプリケーションです。測定器をPCに接続する場合は、インストールする必要があります。このプログラムにより、PCから測定器を制御したり、データを転送するように、システムを簡単に設定できます。このI/Oライブラリ・スイート・ソフトウェアは、I/Oライブラリ・スイートのWebサイト ([www.agilent.co.jp/find/iolib](http://www.agilent.co.jp/find/iolib)) からダウンロードすることができます。**Software**メニューで、**IO Libraries Suite**、**IO Libraries Suite 14.1 (or above)** **Product Download**と続けて選択します。ダウンロードが完了したら、インストール手順に従ってください。

**テスト・セットの接続／セットアップ**：プログラミング用の外部PCをE6601Aテスト・セットと接続するために、さまざまなオプションが用意されています。

### ● GPIB接続

- コンピュータのGPIBインタフェースとテスト・セットのリア・パネル上のGPIBコネクタをGPIBケーブルで接続します。
- テスト・セットのフロント・パネル上の**Screens**の**Config**キーを押します。
- **Configuration Main**ウィンドウの**Instrument Connections**に、必要なGPIBアドレスを入力します。

### ● LAN接続

PCとE6601AのLAN経由での接続には2つの方法があります。1つは、クロスオーバー・ケーブルを使ってPCをE6601Aに直接接続する方法です。すなわち、小規模なPC-テスト・セット間専用LANをセットアップします。もう1つは、オフィスまたはその他の既存のLANまたはハブ経由でPCとE6601Aを接続する方法です。この場合は、ノーマルのLANケーブルを使用します。どちらの方法を選択する場合でも、以下の手順を実行します。

#### ● E6601AのIPアドレスの取得

E6601Aのデスクトップで、**Start**をクリックし、**Run**を選択します。Runウィンドウがポップアップ表示されます。コマンド・バーに**CMD**と入力し、**OK**をクリックします。黒いコマンド・ウィンドウが表示されます。点滅するカーソルの後に**ipconfig**と入力し、**Enter**キーを押します。Windows IP設定が表示されます。**Ethernet Adapter Local Area Connection**の下に示されているIPアドレスが、接続用のアドレスです。

- コンピュータ上で、タスクバーの**Agilent IO Control**アイコンをクリックし、ポップアップ・メニューから**Agilent Connection Expert**を選択します。Agilent Connection Expertが起動したら、LANフォルダの**E6601A**を選択します。ウィンドウの右側にある**Change Properties**ボタンをクリックします。**Change Configurable Properties of This LAN Device**ウィンドウが表示されます。前のステップで確認したE6601AのIPアドレスを入力し、**OK**をクリックします。

**テスト接続の確認**：コンピュータ上で、タスクバーの**Agilent IO Control**アイコンをクリックし、ポップアップ・メニューから**Agilent Connection Expert**を選択します。

Agilent Connection Expertが起動したら、外部PCのインタフェースに接続されている測定器が自動的に表示されます。**Refresh All**をクリックして、これらのインタフェース上の測定器のリストを更新します。

これで、テスト・セットにいくつかコマンドを送ることにより、接続を確認できます。図1を参照してください。

- 接続したいテスト・セットを右クリックします。この例では、E6601Aへの接続にLANが用いられています。ポップアップ・メニューの**Send Commands To This Instrument**をクリックします。Agilent Interactive IOプログラムがオープンします。
- **Agilent Interactive IO**ウィンドウのCommandボックスに、コマンド(例えば、“\*IDN?”)を入力します。**Send Command**をクリックすると、入力したコマンドがE6601Aに送られます。**Read Response**をクリックすると、E6601AのIDNが**Instrument Session History**ウィンドウに表示されます。
- Agilent Interactive IOをクローズするには、**Connect**メニューをクリックし、**Exit**をクリックします。

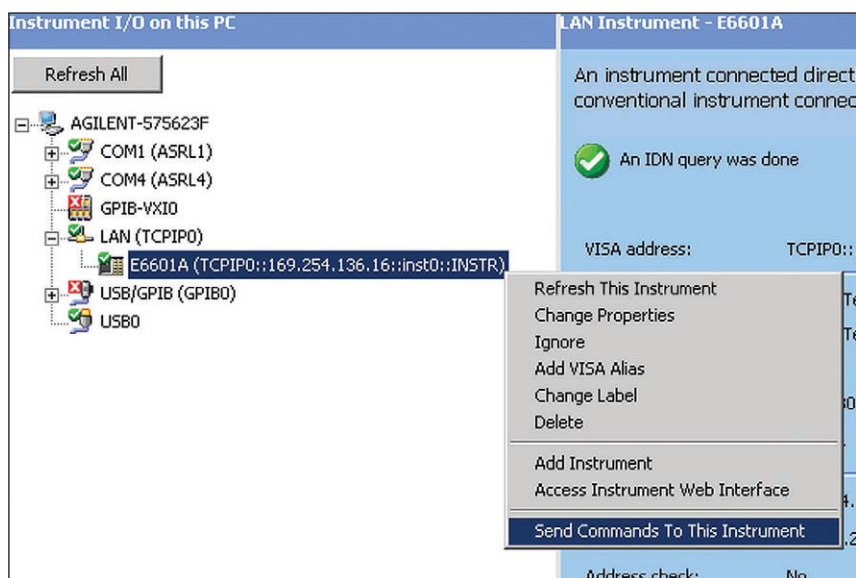


図1. Agilent Connection Expertを使用すると、測定器を簡単にセットアップ/管理できます。

**SCPIコマンドのマニュアルの表示**：E6601Aテスト・セットには、独自の校正アプリケーションがあり、移動機を校正するために、専用のトランスミッタ／レシーバ測定が行えます。このアプリケーション・ノートのプログラミング例では、GSM/GPRS移動機のテスト構成に基づいています。

E6831A GSM/GPRS校正アプリケーションのSCPIコマンド・マニュアルは、E6601Aのオンライン・ヘルプの「 **GPIB Commands (GSM/GPRS CA)**」にあります。このマニュアルには、VISA COMライブラリを使用するVisual Basic .NETで提供されている各コマンドの構文、説明、範囲、\*RST設定、テスト・セット要件、簡単な例が記載されています。もう1つのプログラミング・ツールがAgilent E6601A SCPI Discoveryツールで、Visual Studio .NETなどのプログラミング環境で用いられます。このツールを使用すると、コマンド構造全体にすばやくアクセスできるだけでなく、必要なコマンドをコードにドラッグできるので、プログラミングが簡単になります。図2を参照してください。SCPI Discoveryツールの詳細については、E6601AのWebページ([www.agilent.co.jp/find/e6601a](http://www.agilent.co.jp/find/e6601a))をご覧ください。詳細な使用方法については、このツールの内蔵ヘルプ・システムを参照してください。

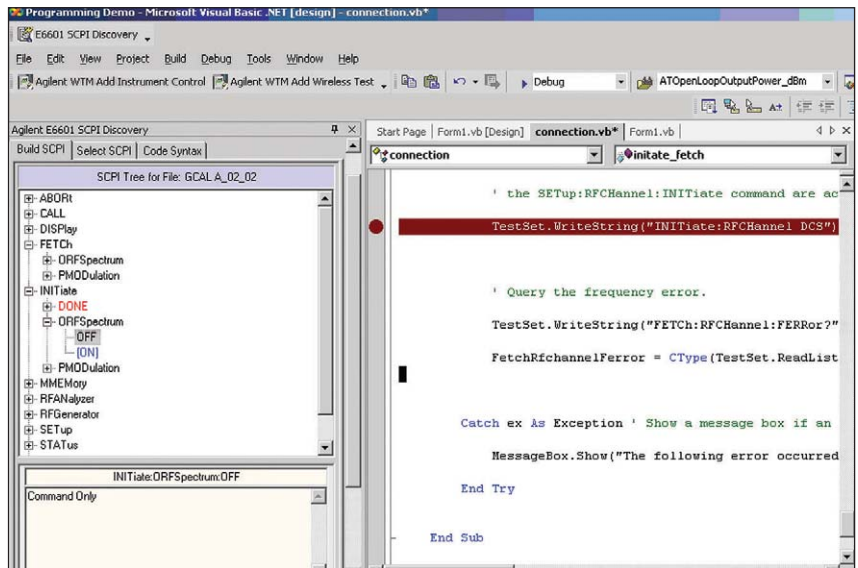


図2. Agilent E6601A SCPI Discoveryは、Visual Studio .NETなどの一般的な環境で使用できるプログラミング・ツールです。

## プログラミングの詳細

下の図3に示されているプログラミング・フローチャートにしたがって、簡単なGSMテスト・プログラムを開発します。

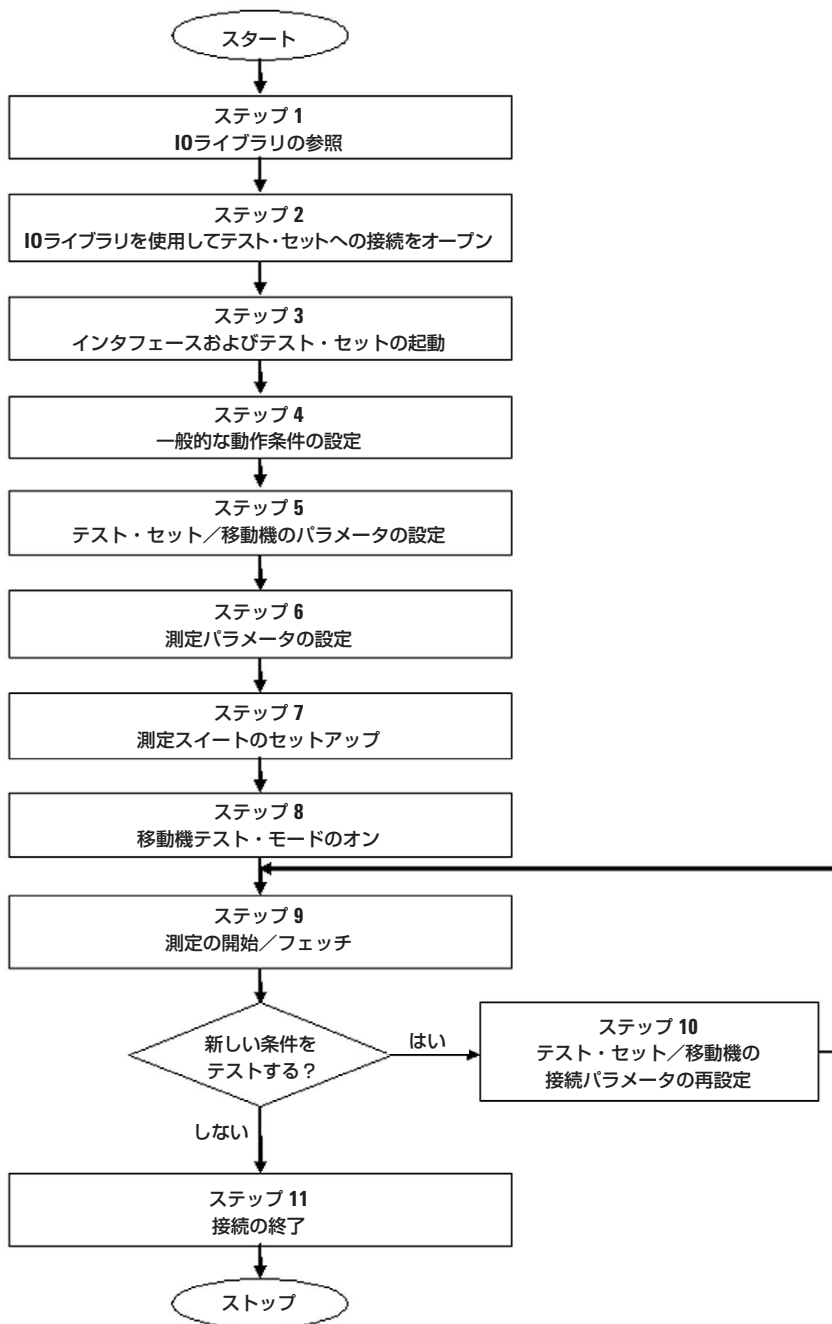


図3. E6601Aテスト・プログラム開発のフローチャート。



**ステップ 1. IOライブラリの参照**：Visual Studio .NETを起動して、新しいVisual Basic .NETアプリケーションを作成します。**Solution Explorer**ウィンドウで、**References**を右クリックし、**Add Reference**をクリックします。図4を参照してください。

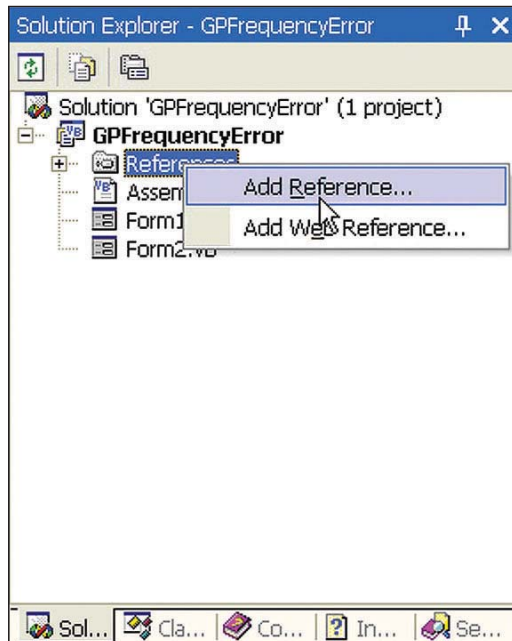


図4. Visual Studio .NETのSolution Explorerウィンドウ。

**Add Reference**ダイアログ・ボックスで、**COM**タブをクリックし、**VISA COM 3.0 Type Library**コンポーネントが表示されるまでスクロール・ダウンします。

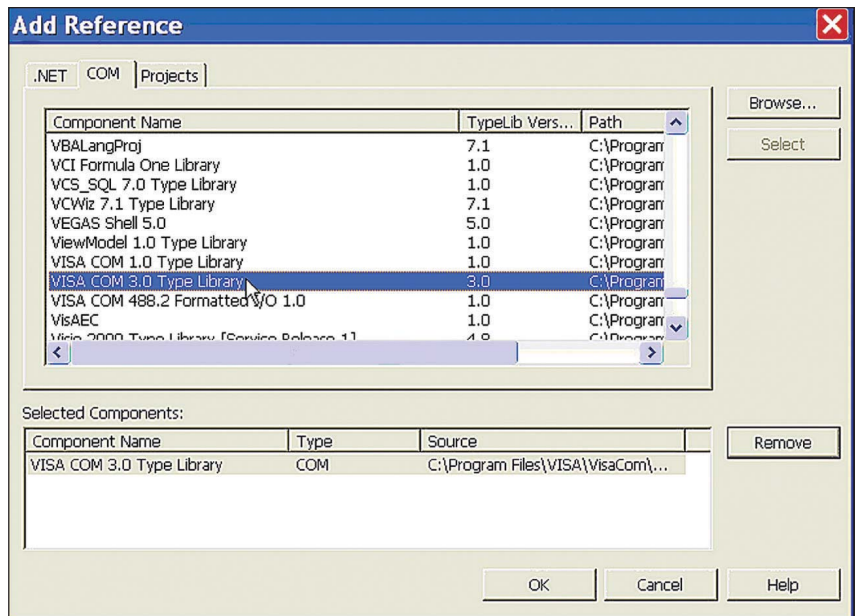


図5. Visual Studio .NETのAdd Referenceウィンドウ。

**VISA COM 3.0 Type Library**をダブルクリックしてコンポーネントを追加し、**OK**をクリックしてAdd Referenceダイアログ・ボックスを閉じます。

**ステップ 2. テスト・セットへの接続をオープン**：コンピュータは、インタフェース経由でメッセージを送受信することによりテスト・セットと通信します。Visual Basic .NETでAgilent VISA COMライブラリを使用した場合は、コマンドや問合せの送信には、FormattedIO488オブジェクトのWriteString、WriteNumber、WriteList、WriteIEEEBlockメソッドが用いられます。VISA COMライブラリを使用するには、最初にResourceManagerオブジェクトのOpenメソッドを使用して、テスト・セットへの接続をオープンする必要があります。問合せが送信されると、ReadString、ReadNumber、ReadList、またはReadIEEEBlockメソッドを使用して、応答を読みます。

以下のVisual Basic .NETステートメントは、テスト・セットへの接続をオープンし、コマンドを送信して、テスト・セットのディスプレイの輝度を変更します。

注記：プログラムのコメントは、わかりやすいように、このアプリケーション・ノートでは、日本語に翻訳してあります。

```
Option Explicit On
Option Strict On
Imports Ivi.Visa.Interop
```

```
Dim ResManager As New ResourceManager
Dim TestSet As New FormattedIO488
```

```
'テスト・セットへの接続をオープンします。Agilent Connection Expert
'(Agilent IOライブラリ・スイートによりインストール)から "VISAアドレス" を取得
'します。
```

```
'このサンプルでは、GPIBインタフェース用の "VISAアドレス" は
'"GPIB0::14::INSTR" です。
```

```
TestSet.IO = CType(ResManager.Open("GPIB0::14::INSTR"), IMessage)
```

```
'LANインタフェースの場合は、“VISAアドレス” は、
```

```
TestSet.IO = CType(ResManager.Open("TCPIP0::localhost::inst0::INSTR"),
IMessage)
```

```
'コマンドを送信します。
```

```
TestSet.WriteString("DISPlay:BRIGhtness MEDium")
```

**ステップ 3. インタフェースおよびテスト・セットの起動**：一貫した再現性の高い性能を保証するには、プログラム、コンピュータ、テスト・セットを既知の状態に起動する必要があります。正しく初期化しないと、プログラムが正常に動作したり、しなかったりします。このようなバツキは、前のプログラムの実行中またはフロント・パネルから行った、テスト・セットの設定の変更に起因する場合があります。

Agilent VISA COMライブラリを使用している場合は、リソース・セッション・オブジェクトのClearメソッドを使用して、インタフェースのバッファをクリアします。GPIBを使用している場合は、リソース・セッション・オブジェクトのClearメソッドによりテスト・セットのパーサもリセットされます。パーサは、テスト・セットに送られてきた命令を読み取るプログラムです。

- **テスト・セットのフル・プリセット**：各製造セッションの前に、テスト・セットを既知の状態にすることが重要です。\*RSTコマンドを送信すると、テスト・セットがフル・プリセットされ、現在の測定プロセスがすべて終了され、すべての値がデフォルトに復元されます。
- **テスト・セットのエラー待ち行列のクリア**：各製造セッションの前に、古いメッセージのエラー待ち行列をクリアすることも有効です。エラー待ち行列をクリアすることにより、記録されているすべてのメッセージが現在の製造セッションに関連したものになります。

下の例のコードは、フル・プリセットし、エラー待ち行列をクリアします。

```
'インタフェース・バッファをクリアし、テスト・セットのパーサをリセットします。  
TestSet.IO.Clear()
```

```
'テスト・セットをフル・プリセットします。  
TestSet.WriteString("*RST")
```

```
'テスト・セットのエラー待ち行列をクリアします。  
TestSet.WriteString("*CLS")
```

**ステップ 4. 一般的な動作条件の設定**：一般的な動作条件には、振幅オフセット、表示モードがあります。

- **振幅オフセットの設定**：振幅オフセットは、テスト・セットと被試験移動機間のケーブルやフィクスチャの損失／利得を補正します。SYSTem:CORRectionコマンドを使って、最大100個の周波数を割り当てることができます。
- **表示モードの設定**：テストの実行を高速化するために、DISPlay:MODE:TRACKingコマンドを使って、テスト・セットのフロントパネル・ディスプレイをオフにできます。

下の例は、振幅オフセットと表示モードの設定方法を示しています。

'オフセット周波数を設定し、周波数、オフセット、RF IN/OUT Amplitude Offset 'Stateのステートをオンに設定します。

```
TestSet.WriteString("SYSTem:CORRection:SFRrequency 1710.2  
MHZ,1805.2 MHZ,1784.8 MHZ,1879.8 MHZ")
```

'オフセット値(単位：dB)を設定して損失を補正します。

```
TestSet.WriteString("SYSTem:CORRection -2.55,-3.12,-3.68,-4.23")
```

下の例は、テスト・セットの表示モードの設定方法を示しています。

'トラッキング・モードをオフにします。

```
TestSet.WriteString("DISPlay:MODE:TRACKing OFF")
```

**ステップ 5. テスト・セット／無線機器のパラメータの設定**：測定毎に設定するパラメータは異なります。ただし、ほとんどの測定で、チャンネル、パワー、バンドの設定が必要です。テスト・セット／移動機のテスト・パラメータを設定するには、以下の手順を実行する必要があります。

- **チャンネルの設定**：各バンドに対応するチャンネルを指定し、アクティブ・バンドを設定します。
- **RFジェネレータ・パワーの設定**：テスト・セットの出力パワーを設定します。
- **RFアナライザ・レベルの設定**：RFアナライザへの予想入力パワー・レベル、デュアル・アップリンクPDTCHでのシングルスロット測定用のアップリンク・バーストを設定します。
- **予想バースト・タイプの設定**：このパラメータは、測定の同期用です。

下の例は、以下を実行します。

- 各バンドのARFCNの設定
- アクティブ・バンドの設定
- RFジェネレータのパワーの設定
- RFアナライザのレベルの設定

\*\*\*バンド／チャンネル・パラメータの設定\*\*\*

,

'RFアナライザの周波数制御を自動に設定します。

```
TestSet.WriteString("RFANalyzer:CONTROL:FREQUENCY:AUTO:GCALibration ON")
```

'DCSバンドの場合は、ARFCNを512に設定します。

```
TestSet.WriteString("TRANsceiver:CHANnel:DCS 512")
```

'EGSMバンドの場合は、ARFCNを5に設定します。

```
TestSet.WriteString("TRANsceiver:CHANnel:EGSM 5")
```

'GSM450バンドの場合は、ARFCNを259に設定します。

```
TestSet.WriteString("TRANsceiver:CHANnel:GSM450 259")
```

'アクティブGSM/GPRSバンドを設定します。

```
TestSet.WriteString("TRANsceiver:BAND:GCALibration DCS")
```

'RFジェネレータのパワー・ステートをオン、レベルを-70 dBmに設定します。

```
TestSet.WriteString("RFGenerator:POWER:GCALibration -70")
```

'RFアナライザの予想入力パワーの制御モードを手動に設定します。

```
TestSet.WriteString("RFANalyzer:CONTROL:POWER:AUTO:GCALibration ON")
```

'シングルスロット測定で測定するアップリンク・バースト2を設定します。

```
TestSet.WriteString("RFANalyzer:BURSt 2")
```

'DCSバンドのバースト1のMobile Station TX Levelを10に設定します。

```
TestSet.WriteString("RFANalyzer:MS:TXLevel:DCS:BURSt1 10")
```

'DCSバンドのバースト2のMobile Station TX Levelを12に設定します。

```
TestSet.WriteString("RFANalyzer:MS:TXLevel:DCS:BURSt2 12")
```

'ミッドアンブル・バーストのTraining Sequence Code (TSC) 2を期待するように、

'テスト・セットを設定します。

```
TestSet.WriteString("RFANalyzer:BURSt:TYPE TSC2")
```

**ステップ 6. 測定パラメータの設定**：以下の2種類のパラメータを設定する必要があります。

- **汎用パラメータ**：カウント、タイムアウト、トリガ
- **固有のパラメータ**：特定の測定のセットアップ構成

下の例は、測定または測定スイート・パラメータの設定方法を示しています。

```
'パワー／変調品質測定スイートのパラメータを設定します  
'パワー／変調品質スイートのトリガ・アームをシングルに設定します。  
TestSet.WriteString("SETup:PMODulation:CONTinuous OFF")
```

```
'パワー／変調品質スイートのマルチ測定カウントを100に設定します。  
TestSet.WriteString("SETup:PMODulation:COUNt 100")
```

```
'VISAタイムアウトを10秒に設定します。このタイムアウトは、以下の測定の  
'タイムアウト値より長くする必要があります。  
TestSet.IO.Timeout = 10000
```

```
'パワー／変調品質スイートのタイムアウトを5秒に設定します。  
TestSet.WriteString("SETup:PMODulation:TIMEout 5S")
```

```
'パワー対時間パワー測定の時間オフセットを設定します。  
TestSet.WriteString("SETup:PMODulation:PVTime:TIME -28uS,-  
10uS,321.2uS,552.8uS,570.8uS")
```

```
'バースト同期モードをミッドアンプルに設定します。  
TestSet.WriteString("SETup:PMODulation:SYNC MIDamble")
```

```
'RFチャンネル測定スイートのパラメータを設定します。
```

```
'RFチャンネル・スイートのトリガ・アームをシングルに設定します。  
TestSet.WriteString("SETup:RFCHannel:CONTinuous OFF")
```

```
'ステートをオンに、マルチ測定カウント値を150に設定します。  
TestSet.WriteString("SETup:RFCHannel:COUNt 150")
```

```
'RFチャンネル・スイートのタイムアウトを5秒に設定します。  
TestSet.WriteString("SETup:RFCHannel:TIMEout 5S")
```

```
'RFチャンネル・スイートのフィルタを1 kHzに設定します。  
TestSet.WriteString("SETup:RFCHannel:FILTer BWKHZ1")
```

```
'1 kHzのフィルタの場合は、トリガ遅延時間を1.5 msに設定します。  
TestSet.WriteString("SETup:RFCHannel:TRIGger:DELaY:BWKHZ1 1.5mS")
```

**ステップ 7. 測定スイートのセットアップ**：測定スイートとは、同じ信号サンプル・セットを使用して実行される測定の集まりです。

測定スイート内で測定を実行するには、以下の2つの方法があります。

```
SETup:<Measurement Suite>  
INITiate:<Measurement Suite>, ...  
FETCh results
```

または

```
INITiate:<Measurement Suite> <Measurement>, ...  
FETCh results
```

これら2つの方法を使用した測定スイートの設定方法の例を以下に示します。

#### **SETup:<Measurement Suite>メソッドの例**

'セットアップを起動します。

```
TestSet.WriteString("SETup:PMODulation:INITiate PFERror")
```

'位相／周波数誤差測定を開始します。

```
TestSet.WriteString("INITiate:PMODulation")
```

'位相／周波数誤差の結果をフェッチします。

```
TestSet.WriteString("FETCh:PMODulation:PFERror?")  
FetchPmodulationPfererror = CType(TestSet.ReadList(), Array)
```

#### **INITiate:<Measurement Suite> <Measurement>, ...メソッドの例**

'位相／周波数誤差測定を開始します。

```
TestSet.WriteString("INITiate:PMODulation PFERror")
```

'位相／周波数誤差の結果をフェッチします。

```
TestSet.WriteString("FETCh:PMODulation:PFERror?")  
FetchPmodulationPfererror = CType(TestSet.ReadList(), Array)
```

**ステップ 8. 接続の実行：**このステップでは、テスト・セットと移動機を接続する必要があります。通常は、テスト・セットの信号に同期し、適切な信号の返信を開始するように、移動機に指示する必要があります。ただし、信号デコードが実行されない測定（例えば、Tx/Rx校正）の場合は、同期が不要な場合もあります。

**ステップ 9. 測定の開始／フェッチ。**このステップは、ステップ 7の測定手順の続きです。

**ステップ 10. テスト・セット／無線機器の接続パラメータの再設定：**ステップ 5で設定した構成を用いて移動機の測定を実行した後、必要に応じてこの構成を変更して、無線機器を再度テストできます。このステップには、チャンネル、無線機器の送信パワー・レベル、セル・パワーなどのテスト条件の変更も含まれます。

下の例は、新しいバンドの設定を変更し(ステップ 5で設定したARFCNを使用)、新しいバンドのアップリンク・パワー制御レベルを新たに設定します。

```
'アクティブGSM/GPRSバンドをGSM450に設定します。これには、前に定義した  
'ARFCN 259を使用します。  
TestSet.WriteString("TRANsceiver:BAND:GCALibration GSM450")
```

```
'GSM450バンドのトラフィック・チャンネルの無線機器アップリンク・パワー制御  
'レベルを21に設定します。  
TestSet.WriteString("RFANalyzer:MS:TXLevel:GSM450:BURSt1 21")
```

**ステップ 11. 接続の終了：**このステップでは、移動機による送信を終了します。通常は、以下の3つの手順が含まれます。

- **無線機器の送信の終了：**送信を終了するには、必要なテスト・モード・コマンドを移動機に送る必要があります。これらのコマンドは移動機に固有のものであります。
- **テスト・セットの部分的プリセット：**すべてのパラメータ値をデフォルトにリセットするのではなく、テスト・セットを部分的にプリセットし、すべての測定プロセスを停止するのが実用的です。

```
'パラメータ値をデフォルトにリセットするのではなく、このプリセット・コマンドを  
'送信して、テスト・セットを部分的にプリセットします。  
TestSet.WriteString("SYSTem:PRESet3")
```

- **テスト・セットへの接続のクローズ：**プログラムがテスト・セットへの接続を必要としなくなった場合は、FormattedIO488オブジェクトのCloseメソッドを使用して接続をクローズすることが重要です。

```
'テスト・セットへのIOインタフェースをクローズします  
TestSet.IO.Close()
```



## E6601AのPC コントローラへの テスト・プログラムの 移動

外部PCで開発／デバッグしたテスト・プログラムをE6601Aにロードすると、外部PCが不要になります。プログラムの移動は簡単です。これまで説明してきたプログラミングでは、本当に変更が必要なのはテスト・セットのリソース記述子だけです。

PCとテスト・セットの間にGPIBインタフェースを使用する例では、コードは以下のようにになっています。

```
TestSet.IO = CType(ResManager.Open("GPIB0::14::INSTR"), IMessage)
```

同じプログラムを内部で実行するには、前述の仮想LANを使用します。これには、以下の2つの方法があります。

**1. Hard coded** : 測定器名 (例 : E6601A0429) を使用 :

```
TestSet.IO = CType(ResManager.Open("TCPIP0::E6601A0429::inst0::INSTR"),  
IMessage)
```

**2. Generic** : 他のテスト・セットへの簡単な移動が可能 :

```
TestSet.IO = CType(ResManager.Open("TCPIP0::localhost::inst0::INSTR"),  
IMessage)
```

プログラムのリソース記述子を変更した後、.EXEをリビルドし、E6601Aに移動することにより (このステップには、USBフラッシュ・ドライバが便利)、前と同様に実行します。これで、これまでの例で使用したGPIBに代わって、仮想LANを使用した通信が行われるようになります。

## まとめ

この例からもわかるように、Windowsベースのプログラミング・ツールを使用すると、E6601Aテスト・セットの自動化プログラムを簡単に記述できます。プログラミング環境／テスト・セットのセットアップ、接続の検証、プログラミング・フローチャートの作成、実際のテスト・プログラムの記述はすべて、比較的簡単に実現できます。記述したプログラムは、E6601Aの内蔵PCにアップロードして、実行できます。このような高度な機能を備えたE6601Aは、変化の速い高スループットの製造ラインに非常に有効なソリューションです。

E6601Aの詳細については、AgilentのWebサイトをご覧ください。

[www.agilent.co.jp/find/e6601a](http://www.agilent.co.jp/find/e6601a)

## 付録：GSM用の サンプル・テスト・ プログラム

以下のプログラミング例は、GSMテストを実行します。パワー／変調品質測定とRFチャンネル測定から構成されています。

```
Option Explicit On
Option Strict On
Imports Ivi.Visa.Interop

Public class Demo
Private ResManager As New ResourceManager
Private TestSet As New FormattedIO488
Friend Sub Measurement()
Dim FetchPmodulationPfferror as Array
'テスト・セットへの接続をオープンします。Agilent Connection Expert
'(Agilent IOライブラリ・スイートによりインストール)から "VISAアドレス" を取得
'します。
'このサンプルでは、GPIBインタフェース用の "VISAアドレス" は
'"GPIB0::14::INSTR"です
TestSet.IO = CType(ResManager.Open("GPIB0::14::INSTR"), IMessage)
'インタフェース・バッファをクリアし、テスト・セットのパーサをリセットします。
TestSet.IO.Clear()

'テスト・セットをフル・プリセットします。
TestSet.WriteString("**RST")

'テスト・セットのエラー待ち行列をクリアします。
TestSet.WriteString("**CLS")

'オフセット周波数を設定し、周波数、オフセット、RF IN/OUT Amplitude Offset
'Stateのステートをオンに設定します。
TestSet.WriteString("SYSTem:CORRection:SFRequency 1710.2 MHZ,1805.2
MHZ,1784.8 MHZ,1879.8 MHZ")

'オフセット値 (dB単位) を設定して損失を補正します。
TestSet.WriteString("SYSTem:CORRection -2.55,-3.12,-3.68,-4.23")

'トラッキング・モードをオフにします。
TestSet.WriteString("DISPlay:MODE:TRACking OFF")
'***バンド／チャンネルのパラメータの設定***

'RFアナライザの周波数制御を自動に設定します。
TestSet.WriteString("RFANalyzer:CONTRol:FREQUency:AUTO:GCALibration
ON")

'DCSバンドの場合は、ARFCNを512に設定します。
TestSet.WriteString("TRANsceiver:CHANnel:DCS 512")

'EGSMバンドの場合は、ARFCNを5に設定します。
TestSet.WriteString("TRANsceiver:CHANnel:EGSM 5")

'GSM450バンドの場合は、ARFCNを259に設定します。
TestSet.WriteString("TRANsceiver:CHANnel:GSM450 259")

'アクティブGSM/GPRSバンドを設定します。
TestSet.WriteString("TRANsceiver:BAND:GCALibration DCS")

'RFジェネレータのパワー・ステートをオン、レベルを-70 dBmに設定します。
TestSet.WriteString("RFGenerator:POWer:GCALibration -70")

'RFアナライザの期待する入力パワーの制御モードを手動に設定します。
TestSet.WriteString("RFANalyzer:CONTRol:POWer:AUTO:GCALibration ON")

'シングル・スロット測定測定で測定するアップリンク・バースト2を設定します。
TestSet.WriteString("RFANalyzer:BURSt 2")

'DCSバンドのバースト1のMobile Station TX Levelを10に設定します。
TestSet.WriteString("RFANalyzer:MS:TXLevel:DCS:BURSt1 10")

'DCSバンドのバースト2のMobile Station TX Levelを12に設定します。
TestSet.WriteString("RFANalyzer:MS:TXLevel:DCS:BURSt2 12")
```

```

'ミッドアンプル・バーストのTraining Sequence Code (TSC) 2を期待するように、
'テスト・セットを設定します。
TestSet.WriteString("RFANalyzer:BURSt:TYPE TSC2")

'パワー／変調品質測定スイートのパラメータを設定します。

'パワー／変調品質スイートのトリガ・アームをシングルに設定します。
TestSet.WriteString("SETup:PMODulation:CONTinuous OFF")

'パワー／変調品質スイートのマルチ測定カウントを100に設定します。
TestSet.WriteString("SETup:PMODulation:COUNt 100")

'VISAタイムアウトを10秒に設定します。このタイムアウトは以下の測定の
'タイムアウト値より長くする必要があります。
TestSet.IO.Timeout = 10000

'パワー／変調品質スイートのタイムアウトを5秒に設定します。
TestSet.WriteString("SETup:PMODulation:TIMEout 5S")

'パワー対時間パワー測定的时间オフセットを設定します。
TestSet.WriteString("SETup:PMODulation:PVTime:TIME -28uS,-
10uS,321.2uS,552.8uS,570.8uS")

'バースト同期モードをミッドアンプルに設定します。
TestSet.WriteString("SETup:PMODulation:SYNC MIDamble")

'RFチャンネル測定スイートのパラメータを設定します。

'RFチャンネル・スイートのトリガ・アームをシングルに設定します。
TestSet.WriteString("SETup:RFCHannel:CONTinuous OFF")

'ステートをオンに、マルチ測定カウント値を150に設定します。
TestSet.WriteString("SETup:RFCHannel:COUNt 150")

'RFチャンネル・スイートのタイムアウトを5秒に設定します。
TestSet.WriteString("SETup:RFCHannel:TIMEout 5S")

'RFチャンネル・スイートのフィルタを1 kHzに設定します。
TestSet.WriteString("SETup:RFCHannel:FILTer BWKHZ1")

'1 kHzのフィルタの場合は、トリガ遅延時間を1.5 msに設定します。
TestSet.WriteString("SETup:RFCHannel:TRIGger:DELay:BWKHZ1 1.5mS")

'開始の設定
TestSet.WriteString("SETup:PMODulation:INITiate PFERror")

'位相／周波数誤差測定を開始します。
TestSet.WriteString("INITiate:PMODulation PFERror")

'位相／周波数誤差の結果をフェッチします。
TestSet.WriteString("FETCh:PMODulation:PFERror?")
FetchPmodulationPfererror = CType(TestSet.ReadList(), Array)

'接続の終了
TestSet.WriteString("SYSTem:PRESet3")

'テスト・セットへのIOインタフェースをクローズします
TestSet.IO.Close()

End sub
End Class

```

## サポート、サービス、およびアシスタンス

アジレント・テクノロジーが、サービスおよびサポートにおいてお約束できることは明確です。リスクを最小限に抑え、さまざまな問題の解決を図りながら、お客様の利益を最大限に高めることにあります。アジレント・テクノロジーは、お客様が納得できる計測機能の提供、お客様のニーズに応じたサポート体制の確立に努めています。アジレント・テクノロジーの多種多様なサポート・リソースとサービスを利用すれば、用途に合ったアジレント・テクノロジーの製品を選択し、製品を十分に活用することができます。アジレント・テクノロジーのすべての測定器およびシステムには、グローバル保証が付いています。アジレント・テクノロジーのサポート政策全体を貫く2つの理念が、「アジレント・テクノロジーのプロミス」と「お客様のアドバンテージ」です。

### アジレント・テクノロジーのプロミス

お客様が新たに製品の購入をお考えの時、アジレント・テクノロジーの経験豊富なテスト・エンジニアが現実的な性能や実用的な製品の推奨を含む製品情報をお届けします。お客様がアジレント・テクノロジーの製品をお使いになる時、アジレント・テクノロジーは製品が約束どおりの性能を発揮することを保証します。それらは以下のようなことです。

- 機器が正しく動作するか動作確認を行います。
- 機器操作のサポートを行います。
- データシートに記載している基本的な測定に係わるアシストを提供します。
- セルフヘルプ・ツールの提供。
- 世界中のアジレント・テクノロジー・サービス・センターでサービスが受けられるグローバル保証。

### お客様のアドバンテージ

お客様は、アジレント・テクノロジーが提供する多様な専門的テストおよび測定サービスを利用することができます。こうしたサービスは、お客様それぞれの技術的ニーズおよびビジネス・ニーズに応じて購入することが可能です。お客様は、設計、システム統合、プロジェクト管理、その他の専門的なサービスのほか、校正、追加料金によるアップグレード、保証期間終了後の修理、オンサイトの教育およびトレーニングなどのサービスを購入することにより、問題を効率良く解決して、市場のきびしい競争に勝ち抜くことができます。世界各地の経験豊富なアジレント・テクノロジーのエンジニアが、お客様の生産性の向上、設備投資の回収率の最大化、製品の測定精度の維持をお手伝いします。

## 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 9:00-19:00 (土・日・祭日を除く)

FAX、E-mail、Webは24時間受け付けています。

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。  
ご発注の際はご確認ください。

Copyright 2006  
アジレント・テクノロジー株式会社



### 電子計測 UP DATE

[www.agilent.co.jp/find/emailupdates-Japan](http://www.agilent.co.jp/find/emailupdates-Japan)

Agilentからの最新情報を記載した電子メールを無料でお送りします。



### Agilent Direct

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

テスト機器ソリューションを迅速に選択し使用できます。



### Agilent Open

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

Agilentは、テスト・システムの接続とプログラミングのプロセスを簡素化することにより、電子製品の設計、検証、製造に携わるエンジニアを支援します。Agilentのシステム対応測定器、オープン・ソフトウェア、PC標準I/O、ワールドワイドのサポートは、テスト・システムの開発を加速します。

Windows、Visual Studio、Visual Studio .NET、Visual Basic、XP Professionalは、Microsoft社の米国登録商標です。



Agilent Technologies

December 6, 2006  
5989-5528JAJP  
0000-00DEP