

User Manual

日本語

翻訳: アドバン オートメーション株式会社



HBM LabVIEW Driver

Version 4.0



Hottinger Brüel & Kjaer GmbH
Im Tiefen See 45
D-64239 Darmstadt
Tel. +49 6151 803-0
Fax +49 6151 803-9100
Email: info@hbm.com
Internet: www.hbm.com

アドバンオートメーション株式会社
101-0047
東京都千代田区内神田1-9-5 SF内神田ビル 5 F
Tel. 03-5282-7047
Fax. 03-5282-0808
Email: info-advan@adv-auto.co.jp
Internet: <https://adv-auto.co.jp/>

HBM public
01.2021

2023年11月 初版

本書の内容は予告なく変更する場合があります。
すべての製品説明は一般的な情報のみを紹介しています。
これらは製品の品質や耐久性を保証するものではありません。

目次

概要	5
ライセンス	7
技術サポート	9
Additional Resources	11
要件	13
サポートしている関数の概要	15
LabVIEW Driver VIの概要	17
サンプル(Demosグループ)	21
ActivateChannelDemo.vi	23
BuildApplicationDemo.vi	24
Demo.vi	25
GetSingleMeasurementDemo.vi	26
InteractiveDemo.vi	27
MGCPProgramControlledDemo.vi	28
PMXAdditionalFeaturesDemo.vi	29
PMXProgramControlledDemo.vi	30
ProgramControlledDemo.vi	31
QuantumXProgramControlledDemo.vi	32
SettingAnalogOutDemo.vi	33
SettingDigitalOutDemo.vi	34
UseCommonApiToSetupScalingAndSensorDemo.vi	35
ZeroSomeSignalsBeforeMeasurementDemo.vi	36
LabVIEW Driverの詳細	37
Systemグループ	38
ActivateSignals.vi	39
AssignSignals.vi	40
ConfigSignal.vi	41
ConnectDevice.vi	42
DeviceInfo.vi	43
DisconnectDevice.vi	44
DisposeApi.vi	45
ExecuteZeroing.vi	46
FilterSignalList.vi	47
GetAvailableDeviceFamilyNames.vi	48
GetChannelFromSignal.vi	49
GetDeviceFromSignal.vi	50
GetIpAddressFromConnectionInfo.vi	51

GetScanableDeviceFamilyNames.vi	52
GetSignals.vi	53
GetSynchronizationQuality.vi	54
Init.vi	55
Problems.vi	56
ScanForDevices.vi	57
SetAnalogOut.vi	58
SetAnalogOutSourceSignal.vi	59
SetDigitalIO.vi	60
SetZeroOffset.vi	61
SignalInfo.vi	62
StartLogging.vi	63
DAQグループ	64
GetMeasurementValues.vi	65
GetSingleMeasurementValues.vi	66
PrepareDAQ.vi	67
StartDaq.vi	68
StopDaq.vi	69
UpdateMeasurementValues.vi	70
QuantumXグループ	71
QuantumX_Device.vi	72
QuantumX_DisableBlinking.vi	73
QuantumX_EnableBlinking.vi	74
PMXグループ	75
PMX_ActivateTEDs.vi	76
PMX_AssignLimitSwitch.vi	77
PMX_ClearPeakValues.vi	78
PMX_Device.vi	79
PMX_GetDeviceStatus.vi	80
PMX_GetDigitalOutMask.vi	81
PMX_GetLimitSwitch.vi	82
PMX_GetMaxValue.vi	83
PMX_GetMinValue.vi	84
PMX_GetPeakToPeakValue.vi	85
PMX_HoldPeakValues.vi	86
PMX_LoadParameterSet.vi	87
PMX_OpenPMXBrowser.vi	88
PMX_ReadParameterSetNumber.vi	89
PMX_SendCommand.vi	90
MGCグループ	91
MGC_Device.vi	92
MGC_SendCommand.vi	93
User Interfaceグループ	94
ScanAndSelectDevices.vi	95
SelectSignals.vi	96
MeasureSignals.vi	97
SetDigitalOut.vi	98
SetAnalogOutUI.vi	99

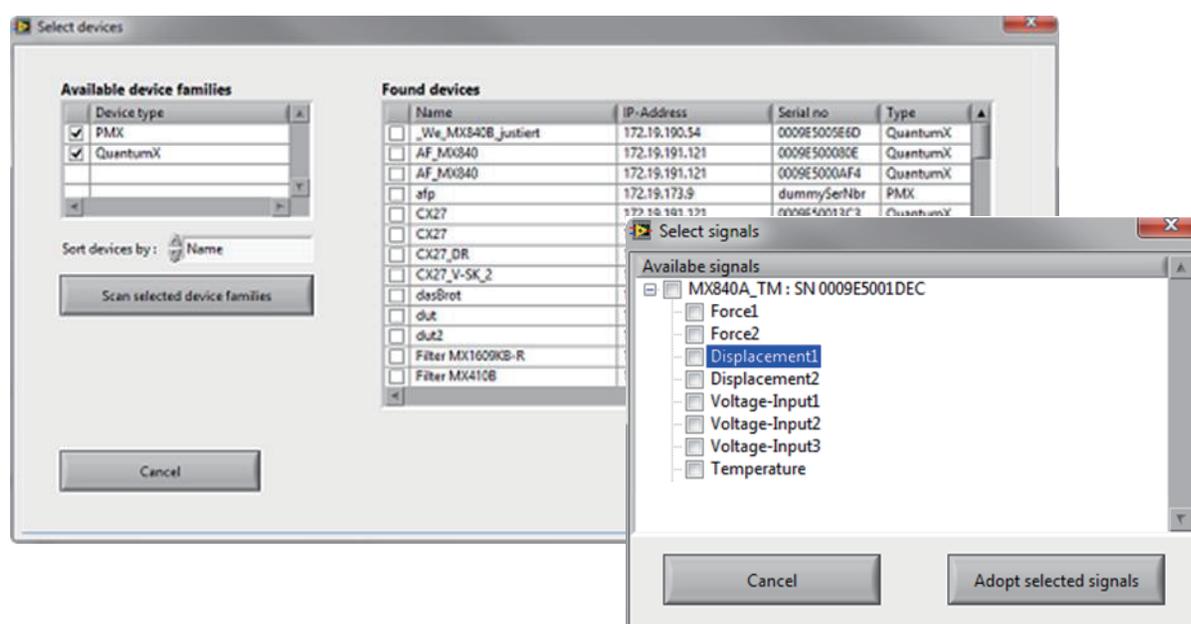
概要

HBM LabVIEW Driverは、HBMデータ収集システムQuantumX /SomatXR、PMX、およびMGCplusをLabVIEWに快適かつ簡単に統合します。ドライバは強力なVIを提供します。

わずか8つのVIで、複数のチャンネルまたはデバイスからの完全なデータ収集を実現できます：



すぐに使用できるグラフィカルユーザインタフェースを備えたVIは、デバイススキャンとチャンネル選択に使用できます：



デバイススキャン、チャンネル選択、接続、計測の開始または停止などの一般的なデバイスコマンドは、共通のVIによって実現します。したがって、データ収集システムを変更しても、アプリケーションの主要部分を再利用できます。

文書化された例により、簡単に始めることができます。

ライセンス

HBM LabVIEW Driverはライセンスファイルによって保護されています。

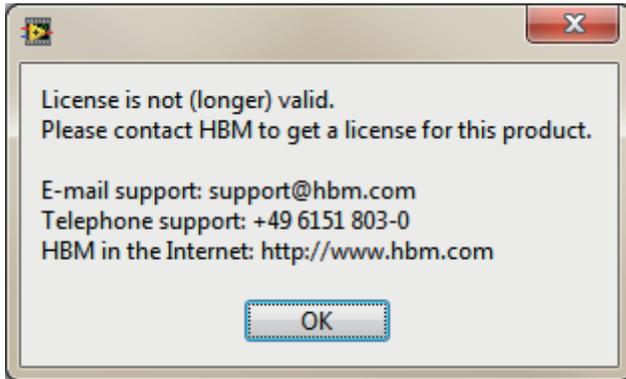
ドライバライセンスを購入するには、最寄りのHBM営業担当者にお問い合わせください。HBMは、フォルダ内の期限切れの評価ライセンスファイルを置き換える永久ライセンスファイルを送信します。

LabVIEWXXXXX\user.lib\HBM LabVIEW Driver\Dlls。

独自のLabVIEW実行可能ファイルを実行するにはライセンスファイルは必要ありません。HBM LabVIEW DriverのVIとともに実行可能ファイルを顧客に配布することができます。

ドライバーのインストール後にWindowsのスタートメニューで利用できるソフトウェアのHBM License Conditionsも参照してください。

HBM LabVIEW Driverには30日間の評価ライセンスが付属しています。それ以降は、LabVIEWの開発環境(IDE)内でVIを使用できなくなります。



技術サポート

サポートや詳細情報が必要な場合は、最寄りの HBM 営業担当者または最寄りのサポート ホットラインのいずれかにお問い合わせください:

<https://www.hbm.com/contact/>

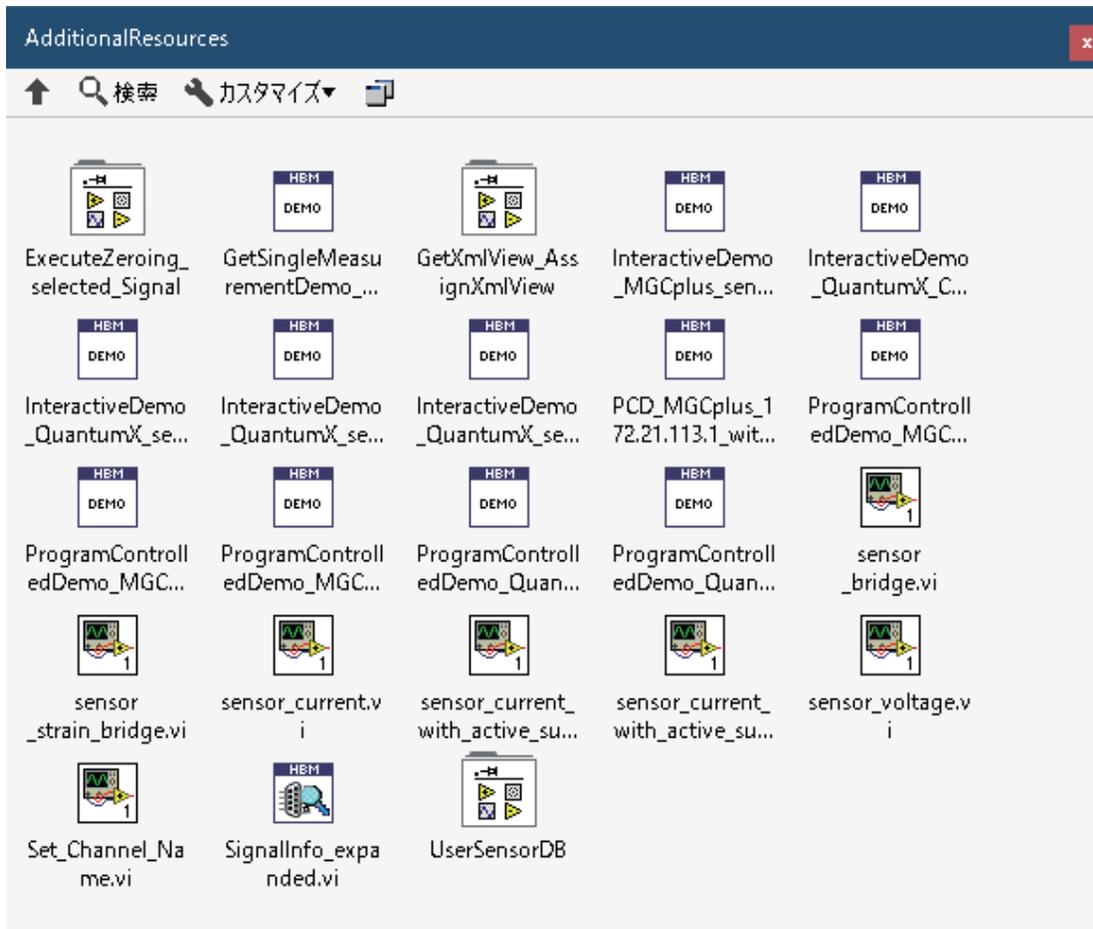
HBM on the Internet:

<http://www.hbm.com>

Additional Resources

テクニカルサポートチームは、さまざまな方法でお客をサポートします。同時に、特定の要求をサポートするために提供されたさまざまなVIが利用可能です。これらのVIは非常に特殊な要求に合わせて作成されたため、user.libの下にはインストールされませんが、HBM LabVIEW DriverのセットアップしたディレクトリのAdditionalResourcesフォルダの内のファイルとして利用できます。

これらは、HBM LabVIEW Driverのセットアップに含まれるVIではカバーされませんが、これらのVIを使用すると基礎となるHBM Common APIの使用法をさらに理解することができ、解決策または少なくとも特定の問題を解決するためのヒントが含まれている場合があります。



要件

HBM LabVIEW Driverを使用するには、システムが次の要件を満たしている必要があります:

- LabVIEW 2012以降*
- .NET Framework 4.0
- QuantumXファームウェア: 4.0.24以降
- PMXファームウェア: 2.0 以降
- MGCplus: CP42ファームウェア: 4.74 以降; CP22ファームウェア: 4.44以降; (CP32は非対応です)

* HBM LabVIEW DriverをLabVIEW 2012で使用するには、LabVIEW.exeと同じフォルダ内に次の内容を含むLabVIEW.exe.configファイルを作成または拡張する必要があります:

```
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0.30319"/>
  </startup>
</configuration>
```

HBM LabVIEW DriverのVIを使用する **プログラムを配布するには**、以下を行う必要があります:

- ターゲットシステムに .NET Framework 4.0をインストールします。
- "LabVIEW XXX/user.lib/HBM LabVIEW Driver/DLLs/"の下にあるすべてのファイルとディレクトリ(***ライセンスファイルを除く**)をアプリケーションのデータディレクトリにコピーします。
- ファイアウォールを変更します(たとえば、ディレクトリ"...\LabVIEWXXXX\user.lib\HBM LabVIEW Driver\DLLs"にある"Firewall_scan_allowed_for_all.bat"を実行します)。
 - スキャンが機能するには、UDP受信ポート31416および31417がアプリケーションに対して開かれている必要があります。
 - Windows® 7以降、これはコマンドラインで実行できます
 - netsh advfirewall firewall add rule name="AppName" direction=in action=allow enable=yes profile=any localport=31416,31417 protocol=UDP edge=yes program="AppExe"
 - Windows® XPのコマンドラインは以下の通りです
 - netsh firewall add allowedprogram name="AppName" mode=ENABLE scope=ALL profile=ALL program="AppExe"

***注意: ライセンスファイルを間違って配布しないでください。**

サポートしている関数の概要

HBM LabVIEW Driverは、HBM Common APIに基づいています。ただし、Common APIのすべての関数が、このドライバのVIで同等のものが用意されているわけではありません。

このドライバの最初のバージョンは計測に重点を置いており、たとえばセンサのパラメータ化をサポートするVIは含まれていません。それでも、VI内でCommon APIを使用して、さらなる関数を実現することが可能です。HBM LabVIEW DriverのVIはどれも保護されていないため(Init.viを除く)、独自のVIのサンプルまたはテンプレートとして使用できます。

DAQ System	Common API			LabVIEW Driver		
	QuantumX / SomatXR	PMX	MGCplus	QuantumX / SomatXR	PMX	MGCplus
Device Scan	✓	✓	✓(5)	✓	✓	✓(5)
Measurement configuration	✓	✓	✓	✓(2)	✓(2)	✓(2)
Sensor configuration	✓	✓	✓	(3)	(3)	(3)
Analog In DAQ	✓	✓	✓	✓	✓	✓
Analog Out (direct setting)	✓	✓	✓	✓	✓	✓
Analog Out (channel routing)	✓	✓	✓	✓	✓	✓
Digital In/Out DAQ	✓	✓	✓	✓	✓	✓
Digital Out (direct setting)	✓	✓	✓	✓	✓	✓
CAN DAQ	✓	✓(4)	✓	✓	✓(4)	✓
CAN Raw DAQ	✓	(1)	(1)	✓	(1)	(1)
Optical In DAQ	✓	(1)	(1)	✓	(1)	(1)

(1) ハードウェアでサポートしていません

(2) VIで設定できるのはサンプルレートとフィルタ周波数のみです

(3) センサ構成用の専用VIはありません(それでも、APIを直接使用してセンサ設定を行うことは可能です。UseCommon Api ToSetupScalingAndSensorDemo.viを確認してこれを行う方法を把握するか、[第5章](#)で説明しているAdditional Resourcesを確認してください)

(4) CODESYS/演算チャンネル経由のみ

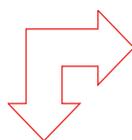
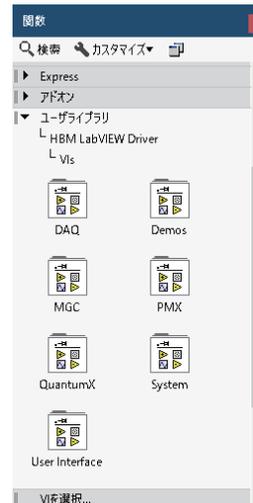
(5) CP52を搭載したMGCplusのみ

LabVIEW Driver VIの概要

HBM LabVIEW Driverが正常にインストールされると、LabVIEWの関数パレット(ユーザライブラリ - HBM LabVIEW Driver)内に新しいエントリが作成され、次の2つのディレクトリが含まれます:

DLLsディレクトリには、必要なライブラリ、ヘルプ ファイル、ライセンスファイルがすべて含まれています。LabVIEW実行可能ファイルをビルドする予定がある場合は、すべてのファイルとディレクトリがプロジェクトに追加されていることをアサートする必要があります(ディストリビューションからライセンスファイルの削除をアサートします)。

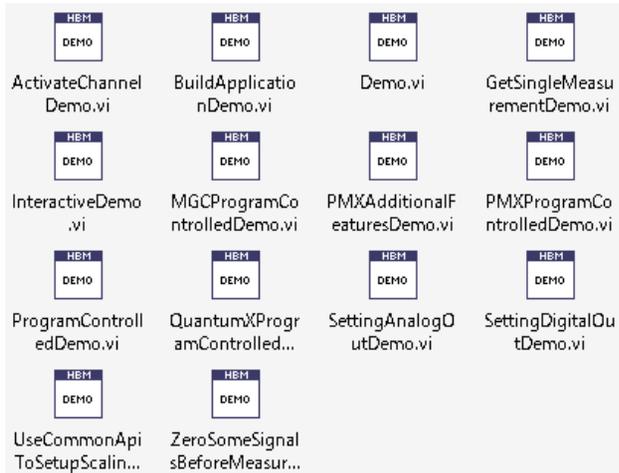
Visディレクトリには、いくつかのサブディレクトリが含まれており、その中に以下のグループがあります:



	<p>Systemグループ</p> <p>このグループのVIは次の目的で使用します:</p> <ul style="list-style-type: none"> LANアダプタのデバイスをスキャン デバイスの接続と切断 信号設定の表示 信号設定のアクティブ化、変更、割り当て フィルタを構成 デジタル出力とアナログ出力を設定 信号をアナログ出力にルーティング ゼロオフセットを設定 <p>これらのVIはすべて、デバイスが機能をサポートしている限り、すべてのデバイスで動作します(例: MGC-CP42デバイスはスキャンできません)。</p>
	<p>DAQグループ</p> <p>このグループには、データ収集に役立つすべてのVIが含まれています。以下をサポートするVIがあります:</p> <ul style="list-style-type: none"> 連続計測の準備 連続計測の開始と停止 継続的な測定値の取得 連続測定を実行せずに単一の測定定値を取得
	<p>User Interfaceグループ</p> <p>このグループのVIは、対話型で次の目的で使用できます:</p> <ul style="list-style-type: none"> 特定のデバイスをスキャンする 特定のデバイスを接続する 使用する信号を選択します 特定の信号を測定する デジタル出力とアナログ出力を設定する <p>これらはすべて、対話型操作を可能にする独自のユーザインタフェースを備えています。</p>
	<p>QuantumXグループ</p> <p>このグループには、QuantumXデバイスの追加関数(特定のコネクタでの点滅など)を実現するVIが含まれています。追加関数は、すべてのデバイスタイプに実装されるCommon関数ではカバーしていません。</p> <p>これらのVIはQuantumXデバイスでのみ使用できます!</p>

	<p>PMXグループ</p> <p>このグループには、PMXデバイスの追加関数(特定のパラメータセットのロードや特定のコネクタでのTEDのアクティブ化など)を実現するVIが含まれています。追加関数は、すべてのデバイスタイプに実装されるCommon関数ではカバーしていません。これらのVIはPMXデバイスでのみ使用できます!</p>
	<p>MGCグループ</p> <p>このグループには、MGCデバイスの追加関数(低レベルコマンドの送信など)を実現するVIが含まれています。追加関数は、すべてのデバイスタイプに実装されるCommon関数ではカバーしていません。これらのVIはMGCデバイスでのみ使用できます!</p>
	<p>Demosグループ</p> <p>このグループのVIは、計測のセットアップ方法、計測値の取得方法、デバイスの追加関数の使用方法、またはデジタルおよびアナログ出力を特定の値に設定する方法を示すさまざまな例をカバーしています。</p>

サンプル(Demosグループ)



HBM LabVIEW Driverには、VIの使用方法を示し、同時に必要なワークフローを示すいくつかのデモVI(Demosフォルダ内にあります)が付属しています。

グループには次のVIが含まれています:

- [ActivateChannelDemo.vi](#)
- [BuildApplicationDemo.vi](#)
- [Demo.vi \(continuous measurement\)](#)
- [GetSingleMeasurementDemo.vi](#)
- [InteractiveDemo.vi](#)
- [MGCPProgramControlledDemo.vi](#)
- [PMXAdditionalFeaturesDemo.vi](#)
- [PMXProgramControlledDemo.vi](#)
- [ProgramControlledDemo.vi](#)
- [QuantumXProgramControlledDemo.vi](#)
- [SettingAnalogOutputDemo.vi](#)
- [SettingDigitalOutputDemo.vi](#)
- [UseCommonApiToSetupScalingAndSensorDemo.vi](#)
- [ZeroSomeSignalsBeforeMeasurementDemo.vi](#)

ActivateChannelDemo.vi

HBM LabVIEW Driver | VIs | Demos

このデモでは、非アクティブ化された信号をアクティブ化する方法を示します。

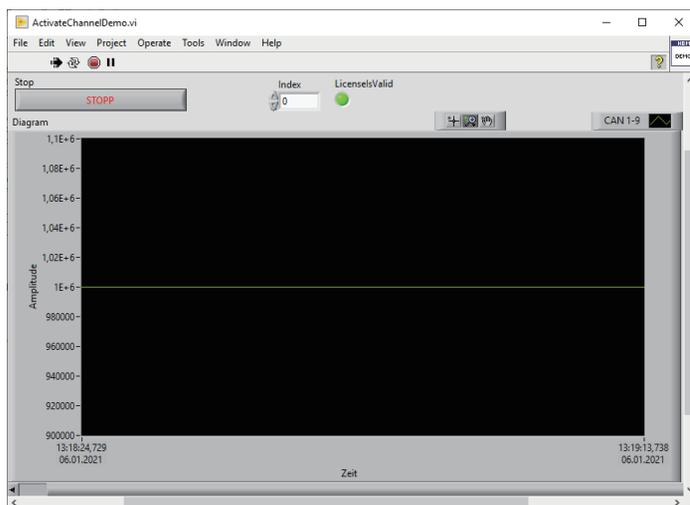
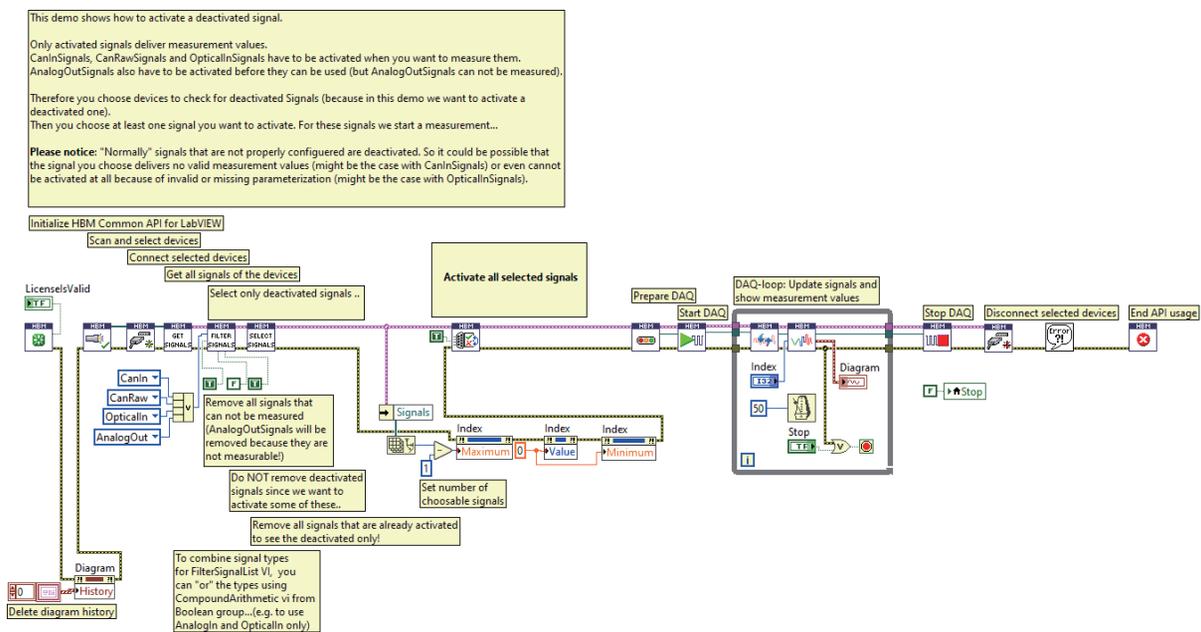
アクティブ化された信号のみが計測値を提供します。

CanInSignals、CanRawSignals、およびOpticalInSignalsを計測するには、これらをアクティブにする必要があります。AnalogOutSignalsも使用する前にアクティブにする必要があります(ただし、AnalogOutSignalsは計測できません)。

したがって、非アクティブ化された信号をチェックするデバイスを選択します(このデモでは、非アクティブ化された信号をアクティブ化したいためです)。

次に、アクティブにする信号を少なくとも1つ選択します。これらの信号について計測を開始します...

注意: "通常"、適切に設定されていない信号は非アクティブ化されます。そのため、パラメータ化が無効または欠落しているために、選択した信号が有効な測定値を提供しない(CanInSignalsの場合など)か、まったくアクティブにできない(OpticalInSignalsの場合など)可能性があります。



ここでは、以前に正しく設定されていないCANチャンネルを有効にしました。そのため、計測値はすべて無効になります(値として1000000が渡されます)。

BuildApplicationDemo.vi

HBM LabVIEW Driver	VIs	Demos
--------------------	-----	-------

このデモは、このVIからアプリケーションを構築する方法を示します。これはInteractiveDemo.viと同じVIです。

このVIにはこのテキスト領域のみがあり、HBM LabVIEW Driverの対話型VIをいくつか使用しているため、フロントパネルにウィジェットが表示されないことを心配する必要はありません。

このVIからアプリケーションを構築するには、次の手順に従ってください:

- このviからアプリケーションをビルドするには、"ツール"/"VIからアプリケーション(EXE)をビルド..."をクリックします。
- アプリケーションのプロジェクトを作成するパスを選択します (user.lib より下のパスは選択しないでください)。
- 必要に応じてVIのプロパティを設定します。
- ("プレビュー"の下の"プレビューを生成"をクリックします)。
- "ビルド"をクリックしてアプリケーションをビルドします。
- "ビルドステータス"ウィンドウで"エクスプローラ"をクリックしてWindowsのファイルエクスプローラを開きます。
- "data"ディレクトリを開きます。
- すべてのファイル(ライセンスファイルを除く)を"LabVIEW XXX/user.lib/HBM LabVIEW Driver/DLLs/"からアプリケーションのdataディレクトリにコピーします (VIによって直接参照されていない場合、LabVIEWはどのDLLが必要であるかを確認できないため、これが重要です)。

これで、構築したアプリケーションのディレクトリ全体を取得して、LabVIEW RuntimeがインストールされているWindowsベースのPCで使用できるようになります。

ターゲットPCのdataディレクトリからバッチファイル"Firewall__scan_allowed_for_all.bat"を実行をアサートしてください。したがって、ファイアウォールは、ネットワーク内のHBMデバイスをスキャンするために必要な特定のポートをブロックしません。

This demo shows how to build an Application from this VI.
This is the same VI as the InteractiveDemo.vi.
Please do not worry about missing widgets on the front panel, since this VI has only this text area and uses a couple of the interactive VIs of the HBM LabVIEW Driver.
To build an application from this VI, just follow these steps:

- to build an Application from this vi, click "Tools"/"Build Application (EXE) from VI..."
- choose a path where to create the project for the application (do not choose a path below user.lib)
- setup properties of your VI as you wish
- (under Preview click "Generate Preview")
- click "Build" to build your Application
- in the "Build status window" click "Explore" to open the windows file explorer
- open "data" directory
- copy ALL FILES (except your License file) from "LabVIEW XXX/user.lib/HBM LabVIEW Driver/DLLs/" into the data directory of your Application
(this is necessary because LabVIEW cannot check which dlls are necessary if they are not directly referenced by a VI)

- now you can take the whole directory of the Application you built and use it on any windows based pc that has the LabVIEW runtime installed.
Please assert that you execute the batchfile "Firewall__scan_allowed_for_all.bat" from data directory on the target pc. So the firewall will not block certain ports that are necessary to scan for HBM devices in the network.

This is the easiest, interactive way to execute a measurement.

Initialize API and do the license check (no license required for executables!)

Scan for devices and choose the devices you want to use

Connect the chosen devices

Get all signals of the connected devices

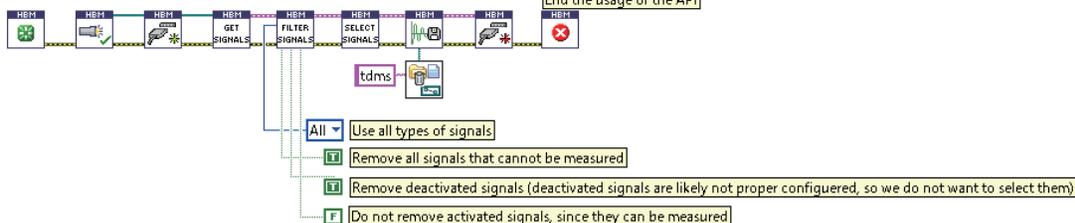
Show only measurable, activated signals

Select signals you want to use

Start an interactive measurement of the selected signals

Disconnect all connected devices

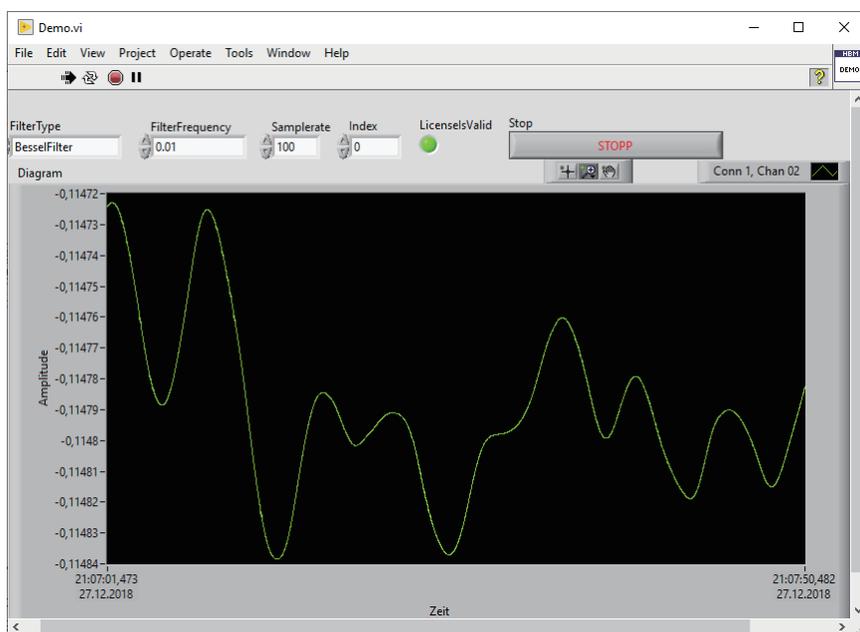
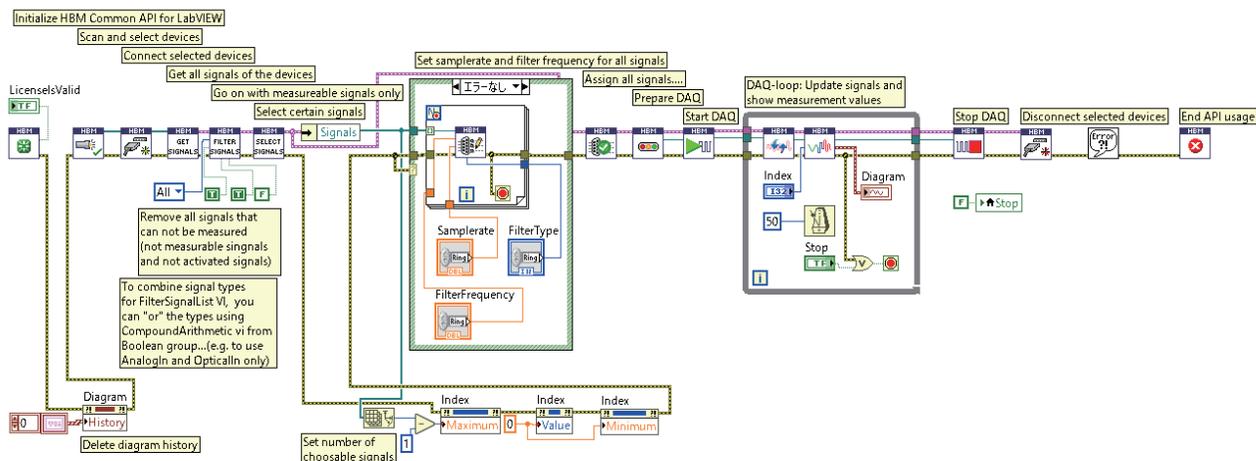
End the usage of the API



Demo.vi

HBM LabVIEW Driver	VIs	Demos
--------------------	-----	-------

このVIは、特定の信号、サンプルレート、フィルタ周波数を使用して計測をセットアップする方法を示します。また、連続計測を実行するために必要なVIの使用方法も示します。

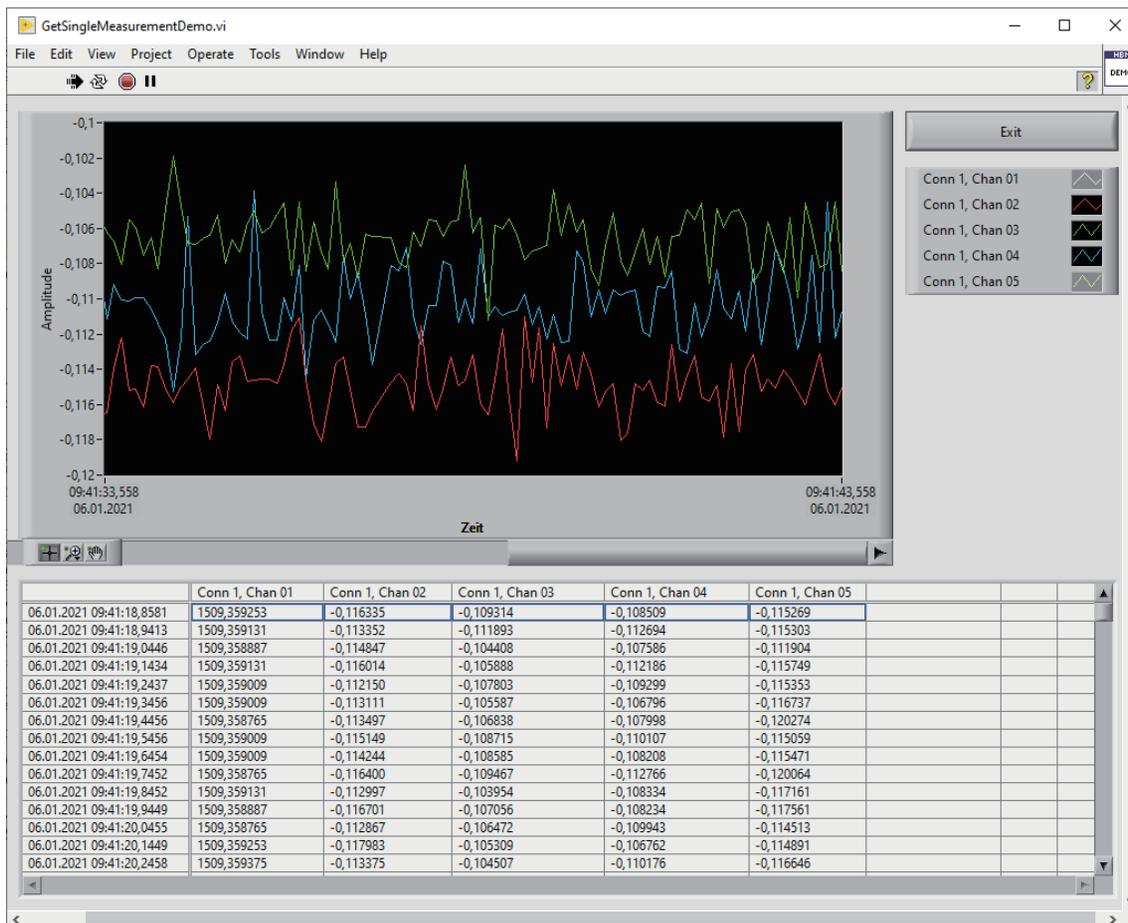
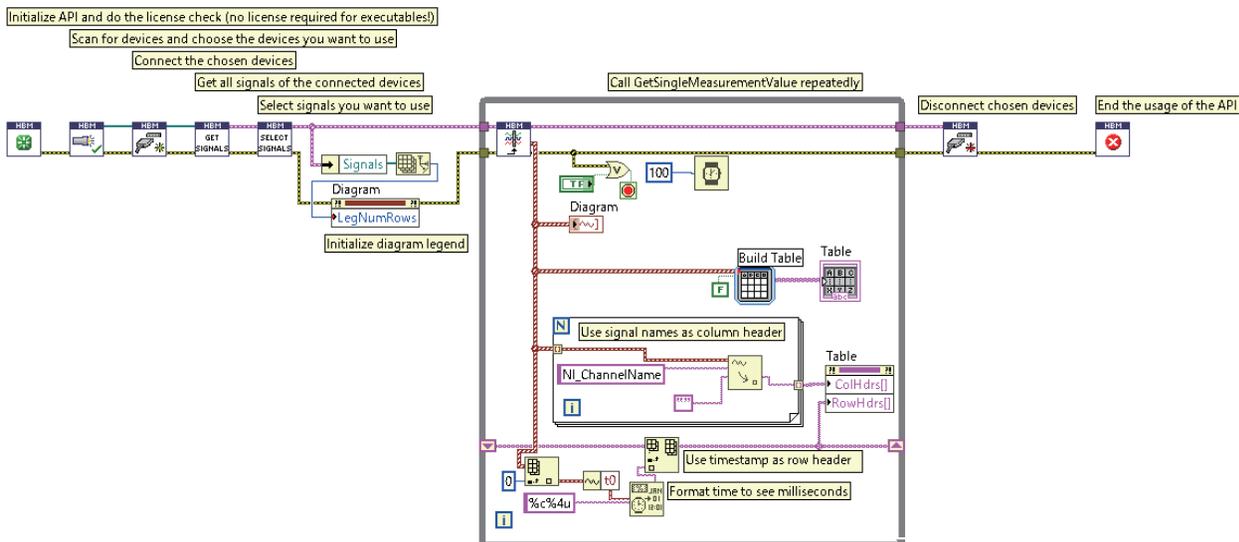


GetSingleMeasurementDemo.vi

HBM LabVIEW Driver | VIs | Demos

このVIは、連続計測を開始せずに計測値を取得する方法を示します。
計測値は、GetSingleMeasurementValueを繰り返し呼び出すことによって取得します。計測値はチャートと表に表示します。

This VI demonstrates how to get measurement values without starting a continuous measurement. Measurement values are obtained by repeatedly calling of the GetSingleMeasurementValue.vi and are rendered in a diagram and also in a table.



InteractiveDemo.vi

HBM LabVIEW Driver | VIs | Demos

このVIは、最小限の数のVIで測定を実行する非常に簡単で対話的な方法を示しています。User Interfaceグループの多くのVIを使用しています。

これらのVIはサポートしているすべてのデバイスタイプで動作することに注意してください。

This is the easiest, interactive way to execute a measurement.

Initialize API and do the license check (no license required for executables!)

Scan for devices and choose the devices you want to use

Connect the chosen devices

Get all signals of the connected devices

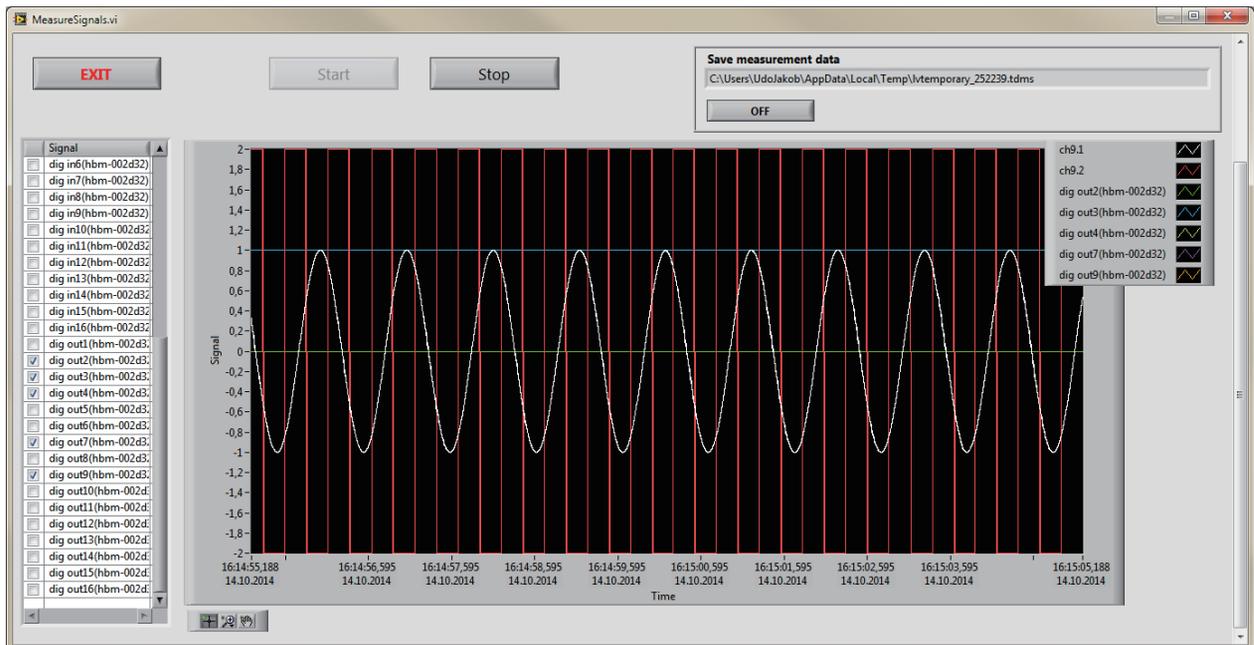
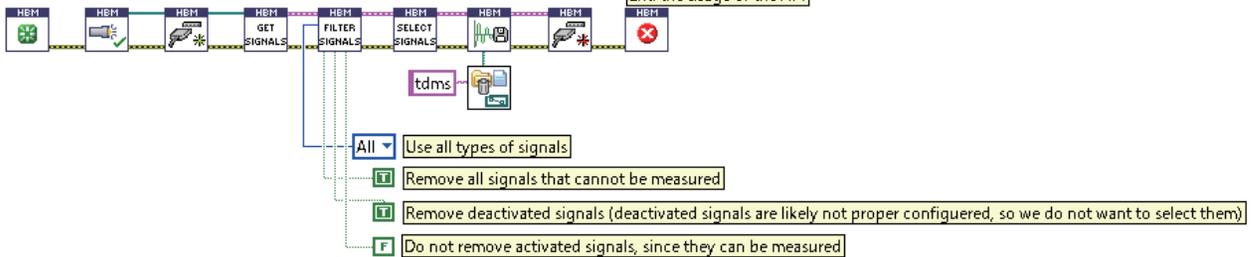
Show only measurable, activated signals

Select signals you want to use

Start an interactive measurement of the selected signals

Disconnect all connected devices

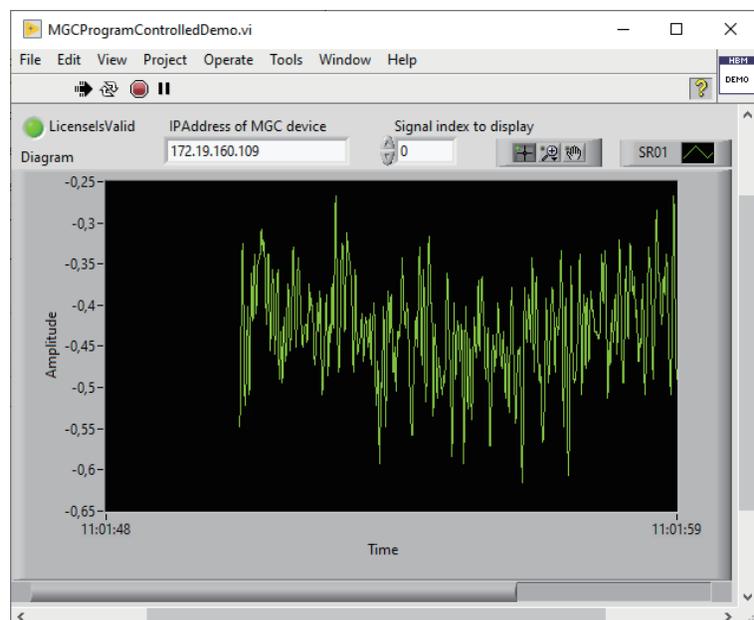
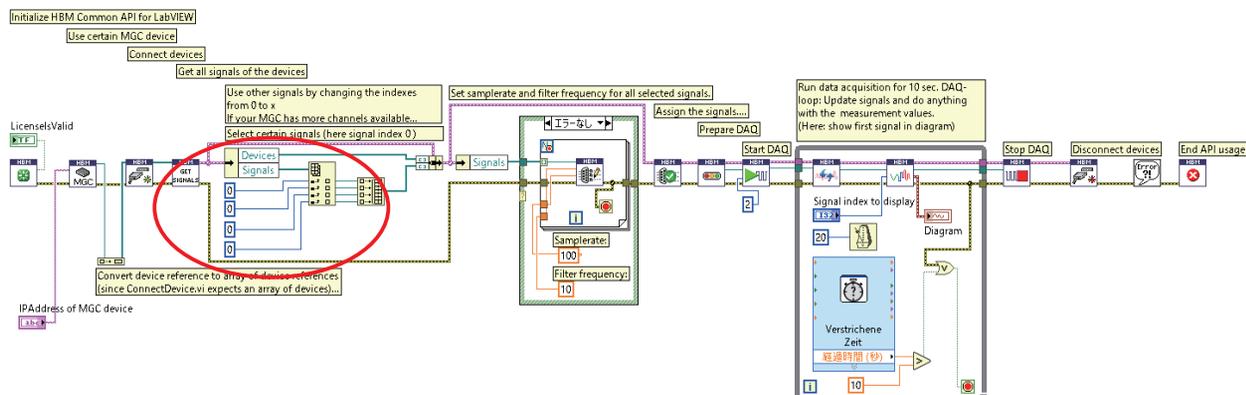
End the usage of the API



MGCProgramControlledDemo.vi

HBM LabVIEW Driver | VIs | Demos

このVIは、ユーザが入力することなくMGCデバイスを使用して計測をセットアップして実行する方法を示します。最初の信号以外にも多くの信号を使用したい場合は、"Select certain signals (here...)"の下の信号番号を調整する必要があります。



PMXAdditionalFeaturesDemo.vi

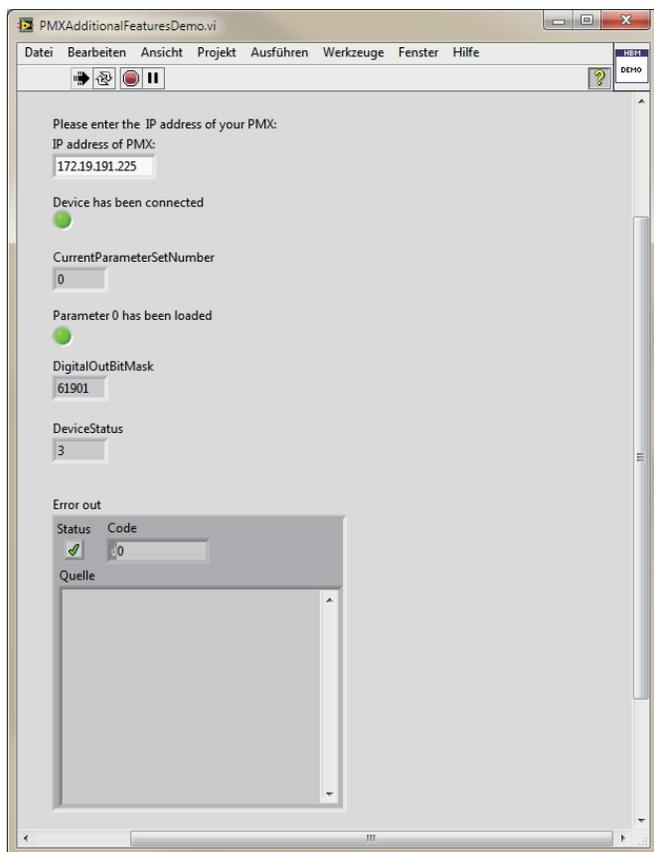
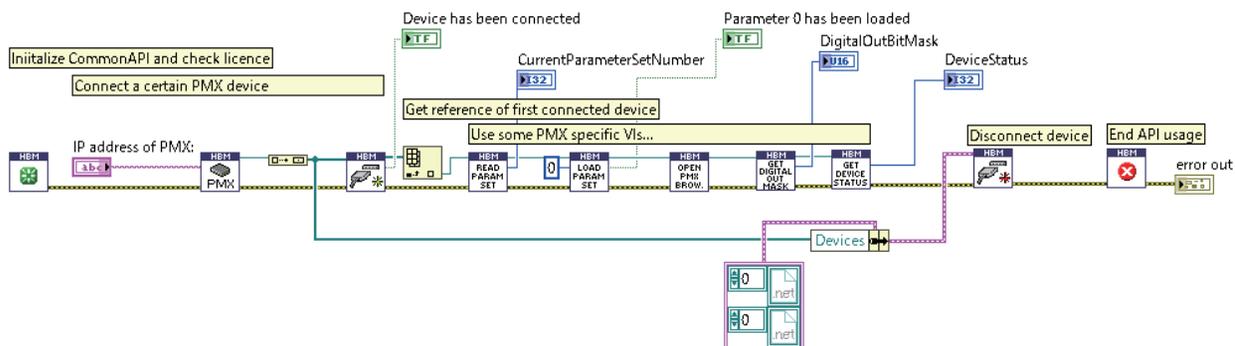
HBM LabVIEW Driver | VIs | Demos

このVIは、特定のデバイスファミリ(ここではPMX)の追加機能の使用法を示します。

これらのVI(ReadParameterSetNumber.viなどはPMXフォルダ内にあり、PMXデバイスでのみ動作します。

これらVIのいずれかをMGCまたはQuantumXで使用しようとすると、エラーをスローします。

This VI demonstrates the usage of the additional features of a certain device family (here PMX). These VIs (e.g. ReadParameterSetNumber.vi) are only working within the folder PMX. If you try to use one of these VIs with a MGC or with a QuantumX, an error will be thrown.

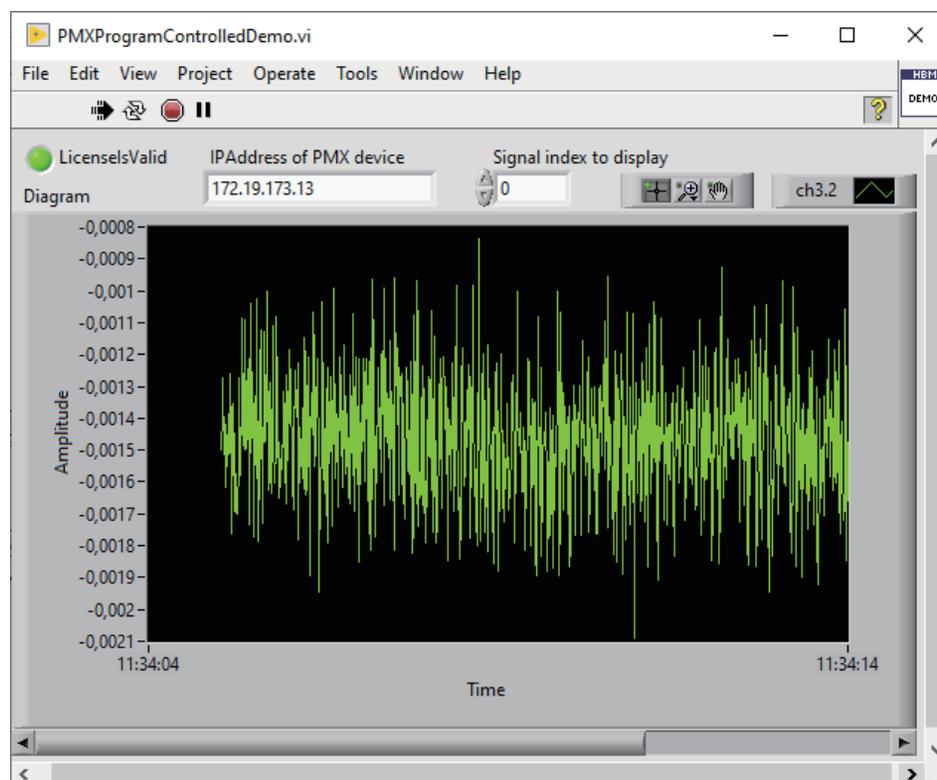
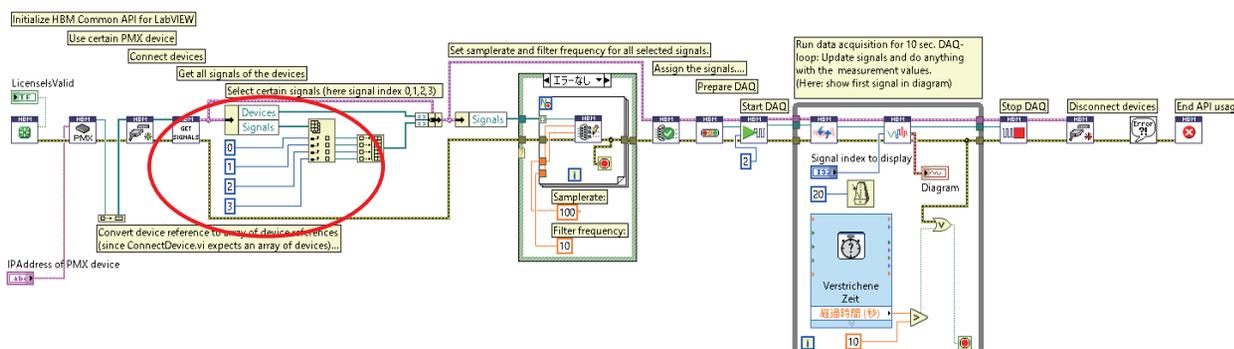


PMXProgramControlledDemo.vi

HBM LabVIEW Driver	VIs	Demos
--------------------	-----	-------

このVIは、ユーザが入力することなくPMXデバイスを使用して計測をセットアップして実行する方法を示します。最初の信号以外にも多くの信号を使用したい場合は、"Select certain signals (here...)"の下の信号番号を調整する必要があります。

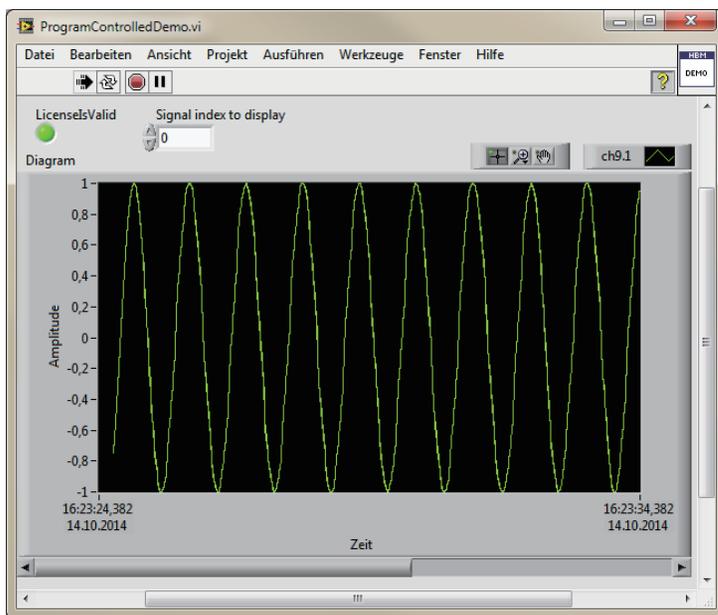
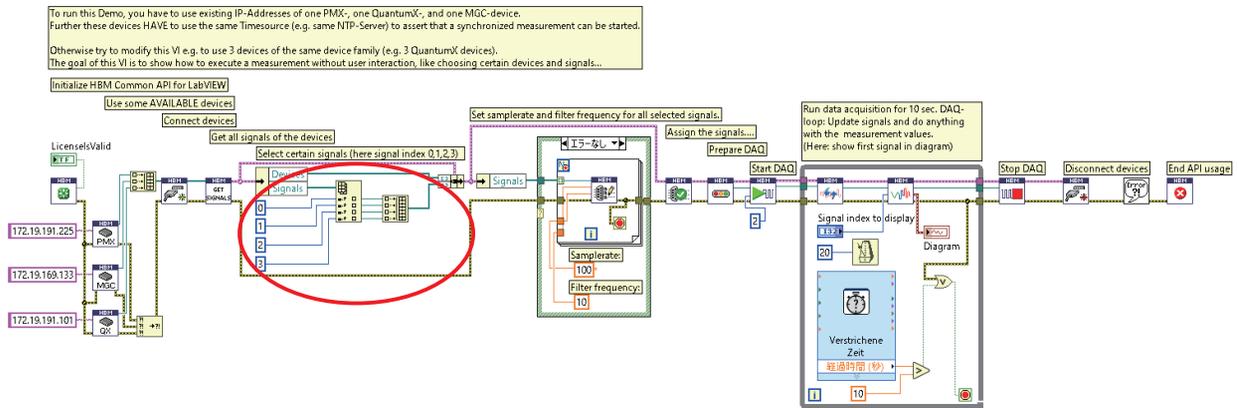
PMXに組み込まれているアンプによっては、信号が計測できない場合があります。この場合、使用する信号のインデックスを調整する必要があります。



ProgramControlledDemo.vi

HBM LabVIEW Driver | VIs | Demos

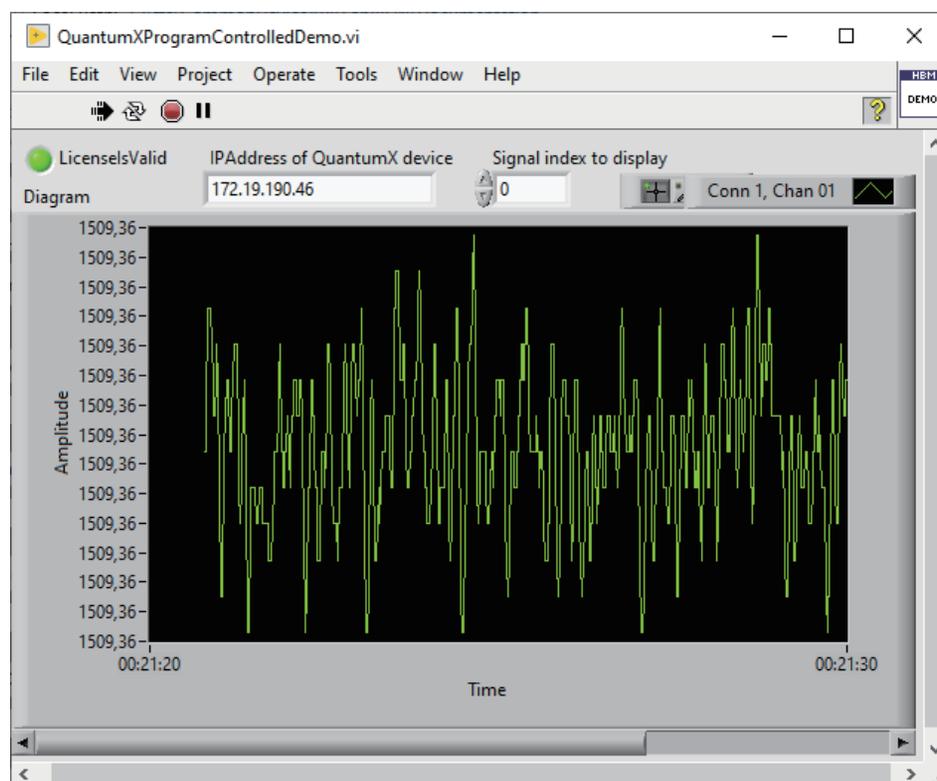
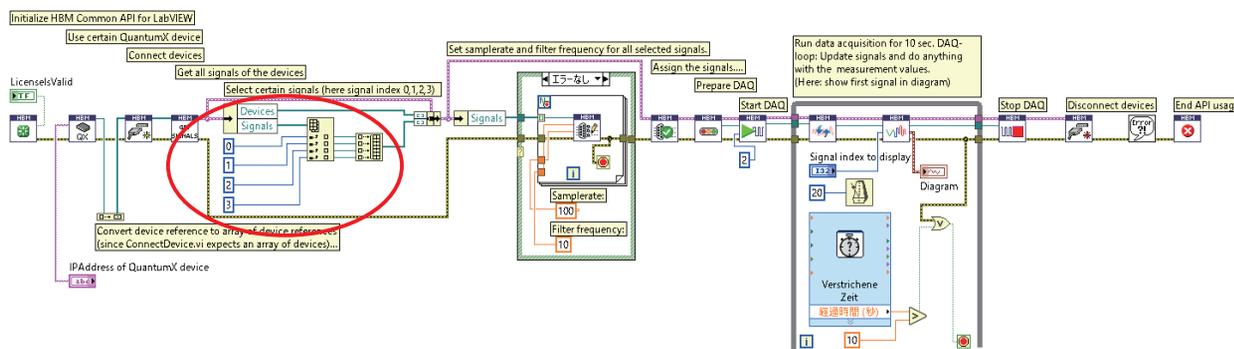
このVIは、ユーザが入力することなく計測をセットアップして実行する方法を示します。ほとんどの場合、3つの異なるDAQデバイスを同時に使用することはないため、おそらくこのデモを調整する必要があります。使用しないデバイスタイプをすべて削除し、"BuildArray.vi"を新しい使用デバイス数に合わせて調整してください。エラーなしでデモを実行するには、"Select certain signals (here...)"の下の信号番号も調整する必要があります。



QuantumXProgramControlledDemo.vi

HBM LabVIEW Driver	VIs	Demos
--------------------	-----	-------

このVIは、ユーザが入力することなくQuantumXデバイスを使用して測定をセットアップして実行する方法を示します。最初の信号以外にも多くの信号を使用したい場合は、"Select certain signals (here...)"の下の信号番号を調整する必要があります。



SettingDigitalOutDemo.vi

HBM LabVIEW Driver	VIs	Demos
--------------------	-----	-------

このVIは、デジタル出力信号を調整する方法と、特定のタイプの信号(ここではデジタル出力信号)をフィルタリングする方法を示します。

Set digital out signals of certain devices to high or low

Initialize API and do the license check (no license required for executables!)

Scan for devices and choose the devices you want to use

Connect the chosen devices

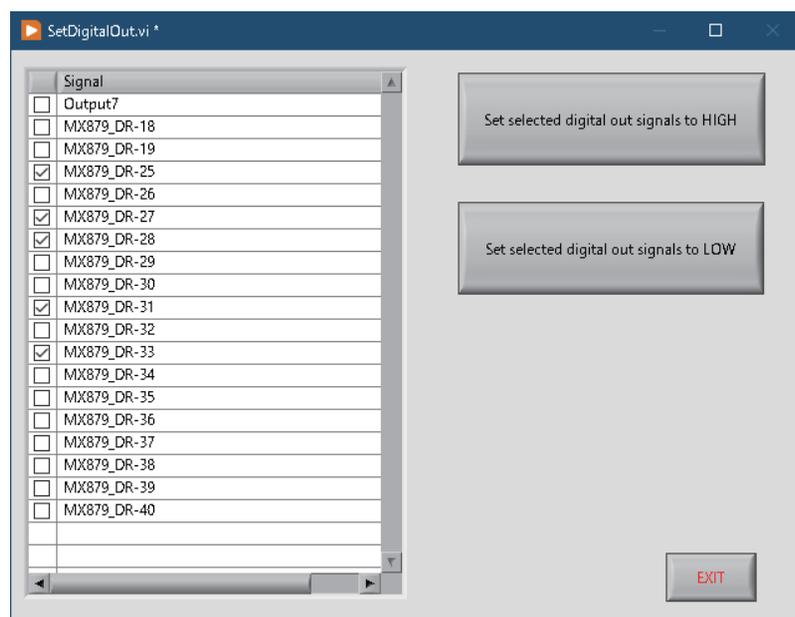
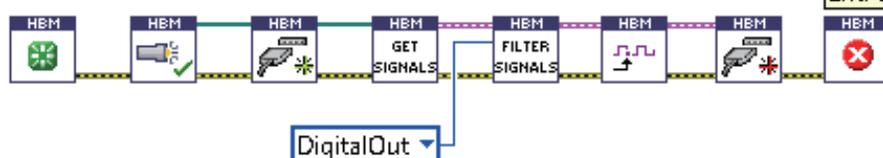
Get all signals of the connected devices

Select only digital out signals

Set digital out signals to high or low

Disconnect devices

End the usage of the API



UseCommonApiToSetupScalingAndSensorDemo.vi

HBM LabVIEW Driver | VIs | Demos

このデモでは、Common APIを使用してセンサとそのスケーリングをパラメーター化して割り当てる方法を示します。複雑になりすぎないように、このVI内でエラー処理を行わずに必要なワークフローを示します。

基本的に、Common APIで実行できることはすべてLabVIEWでも実行できます。

ただし、Common APIの多くの機能と可能性は、LabVIEWで準備されたVIでは直接サポートされていません。

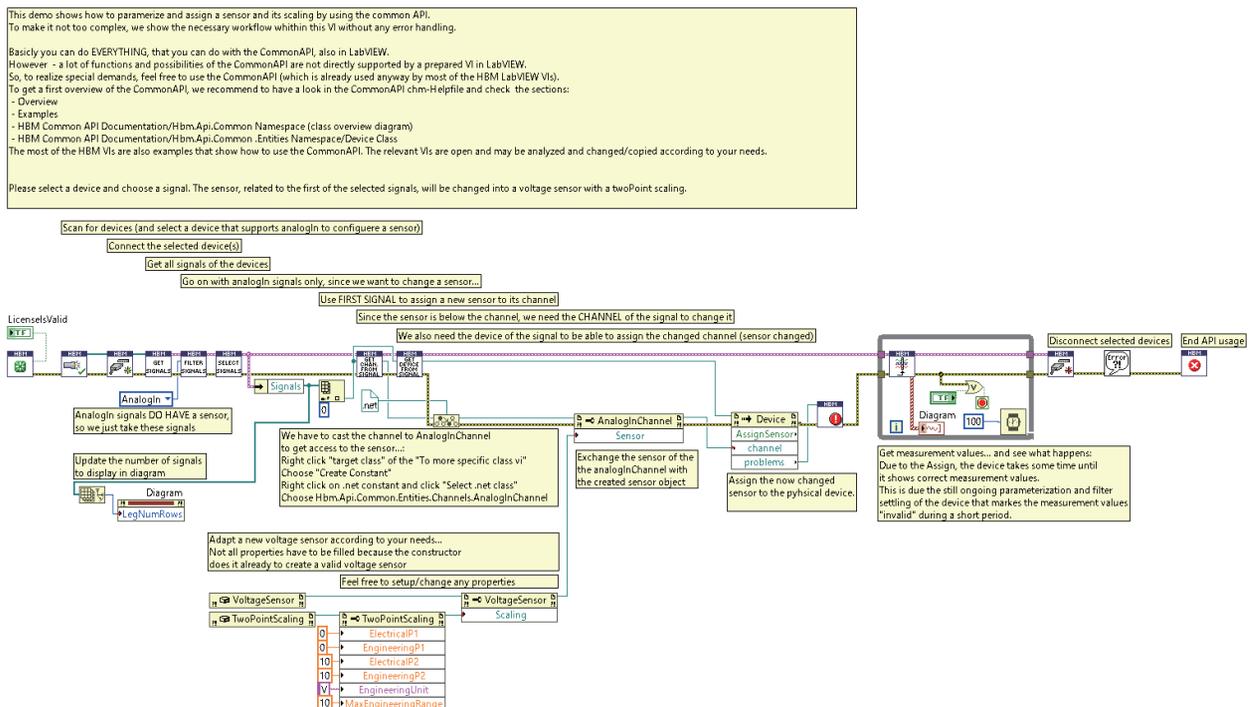
したがって、特別な要求を実現するには、Common API(ほとんどのHBM LabVIEW VIで既に使用されています)を自由に使用してください。

Common APIの概要を最初に理解するには、Common API chm-Helpfileを参照し、次のセクションを確認することをお勧めします:

- Overview
- Examples
- HBM Common API Documentation/Hbm.Api.Common Namespace (class overview diagram)
- HBM Common API Documentation/Hbm.Api.Common.Entities Namespace/Device Class

HBM VIのほとんどは、Common APIの使用法を示すサンプルでもあります。関連するVIを開き、必要に応じて解析や変更/コピーをしてください。

デバイスを選択し、信号を選択します。選択した最初の信号に関連するセンサは、2ポイントスケーリングを持つ電圧センサに変更されます。



ZeroSomeSignalsBeforeMeasurementDemo.vi

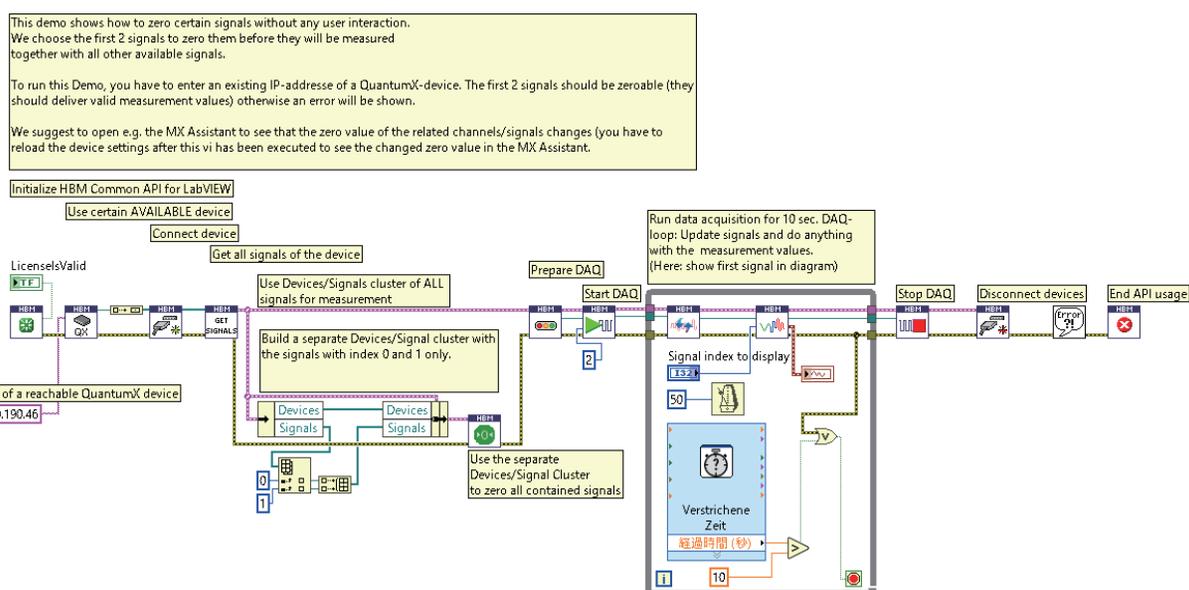
HBM LabVIEW Driver	VIs	Demos
--------------------	-----	-------

このデモでは、ユーザ操作を行わずに特定の信号に対してゼロバランスを実行する方法を示します。

最初の2つの信号を選択して、他のすべての利用可能な信号と一緒に計測する前にそれらをゼロにします。

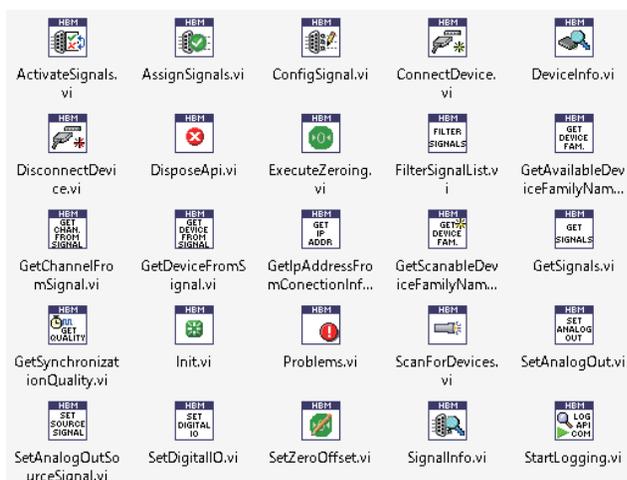
このデモを実行するには、QuantumXデバイスの既存のIPアドレスを入力する必要があります。最初の2つの信号はゼロ調整可能である必要があります(有効な計測値を提供する必要があります)。それ以外の場合はエラーが表示されます。

たとえば、MX Assistantを開いて、関連するチャンネル/信号のゼロ値が変化することを確認することをお勧めします(MX Assistantで変更されたゼロ値を確認するには、このviの実行後にデバイス設定をリロードする必要があります)。



LabVIEW Driverの詳細

Systemグループ



このグループのVIは次の目的で使用します:

- LANアダプタのデバイスをスキャン
- デバイスの接続と切断
- 信号設定の表示
- 信号設定のアクティブ化、変更、割り当て
- フィルタを構成
- デジタル出力とアナログ出力を設定
- 信号をアナログ出力にルーティング
- ゼロオフセットを設定

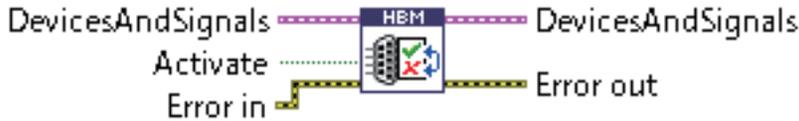
これらのVIはすべて、デバイスが機能をサポートしている限り、すべてのデバイスで動作します(例: MGC-CP42デバイスはスキャンできません)。

ActivateSignals.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定した信号を有効または無効にします (Activate設定に従って)。有効化された信号のみが計測に使用できます! サポートされている信号タイプは、CanInSignal、CanRawSignal、FbgSignal(光)、およびAnalogOutSignalです。有効/無効化できるすべての信号が処理されます。有効/無効化をサポートしていない信号タイプは無視されます。信号はデバイス上で直接有効/無効化されるため、後でAssignSignal.viを実行する必要はありません。エラー出力は、有効/無効化中に発生した問題 (デバイスが接続されなくなったことなどによる) を通知します。

注意: 適切に設定されていない信号を有効にすると、問題が発生します。



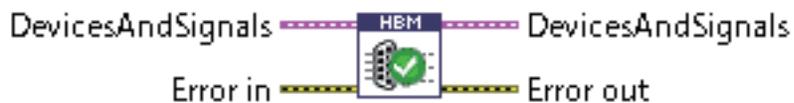
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Activate	信号(実際にはチャンネル)を有効にするにはtrueに設定し、無効にするには falseに設定します。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

AssignSignals.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定された信号の設定を、対応するデバイスの物理信号に割り当てます。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

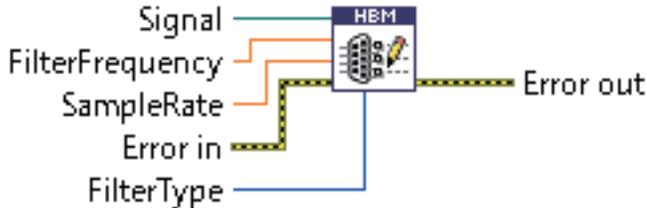
戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

ConfigSignal.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定されたパラメータに従って信号を設定します。信号がフィルタ周波数をサポートしていない場合(演算チャネルなどの仮想信号など)、ここではエラーは発生しません。

備考: これらの設定で物理デバイスを構成するには、信号の構成時にAssignSignals VIを使用する必要があります。



引数	説明
Signal	DevicesAndSignalsのSignals配列から出力された、設定したい信号。
FilterFrequency	使用するフィルタ周波数。
SampleRate	計測に使用するサンプルレート。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。
FilterType	使用するFilterType。PresentFilterはデフォルト設定であり、デバイスに設定されている現在アクティブなFilterTypeが使用されることを意味します。 0: PresentFilter 1: AutoFilter 2: BesselFilter 3: ButterworthFilter 4: HighpassFilter 5: LinearPhaseFilter 6: NoFilter

戻り値	説明
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

注意: すべてのFilterTypeとSampleRate/FilterFrequencyの組み合わせがすべてのデバイスでサポートされているわけではありません。無効な値を選択すると、AssignSignal.viを介してデバイスへの設定割り当て時にエラーが発生します。

ConnectDevice.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定したすべてのデバイスを接続します。



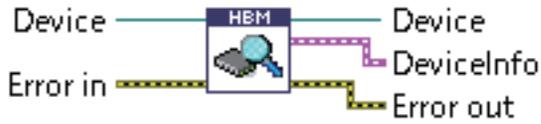
引数	説明
Devices	DevicesAndSignalsのDevices配列からの入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Devices	接続するデバイスの配列出力。
Result	接続中に警告やエラーが発生しなかった場合はTrue。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

DeviceInfo.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

さまざまなデバイスのプロパティを提供します。



引数	説明
Devices	DevicesAndSignalsのDevices配列からの入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Devices	接続するデバイスの配列出力。
DeviceInfo	デバイスに関する情報を含むクラスタ。下の表を参照してください
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

DeviceInfoクラスタ	説明
FamilyName	このデバイスが属するファミリーの名前。
Name	デバイス名。
Model	デバイスのモデルタイプ。
SerialNo	デバイスのシリアル番号。
FirmwareVersion	ファームウェアバージョン。
HardwareVersion	ハードウェアバージョン。
ProtectionType	デバイスのIP保護タイプ(例: IP55)。
IsConnected	デバイスが接続されているかどうかを示します。
IsReadingDaqValues	デバイスが計測値を更新しているかどうかを示します。
IsUnsupportedModel	特定のモデルがAPIでサポートされているかどうかを示します。
Timeout	デバイスとの通信のタイムアウト。
SyncMode	計測開始時のデバイスの役割を定義するモード。

DisconnectDevice.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定したすべてのデバイスの接続を解除します。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

DisposeApi.vi

HBM LabVIEW Driver	VI	System
--------------------	----	--------

Common APIの使用を終了するにはこのVIを使用します。



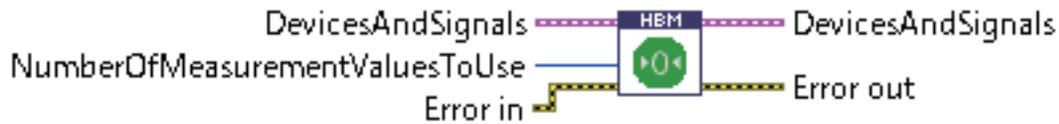
引数	説明
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

ExecuteZeroing.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

すべての信号に対してゼロバランスを実行します(実際にはチャンネルのゼロオフセットが設定されるため、同じチャンネルに属する他の信号も影響を受けることに注意してください)。



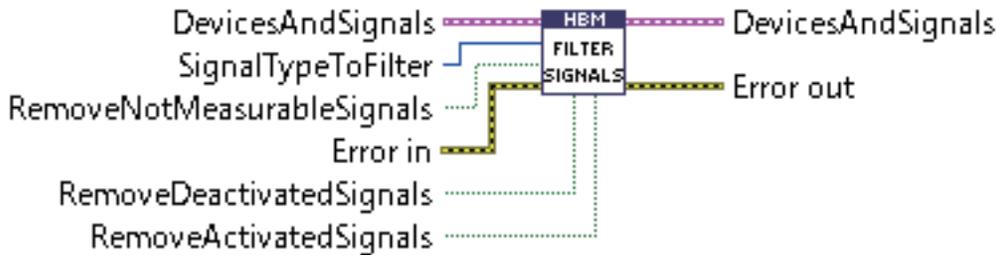
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。配列内のすべての信号に対してゼロ調整が行われます。ゼロ調整は、有効な計測値を提供する信号に対してのみ実行できます。
NumberOfMeasurement ValuesToUse	信号のゼロ オフセットとして使用される平均値を計算するために、各信号の計測値数を定義します(デフォルト値は1)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

FilterSignalList.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定した信号タイプとその他のプロパティに従って、指定したすべての信号をフィルタします。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
SignalTypeToFilter	フィルタリングする信号タイプ(さまざまな信号タイプを追加できます。例: DigitalOut + AnalogOut)。実行後のデバイスと信号の出力には、これらの信号のみが含まれます。 デフォルト値は"All"で、すべての信号タイプがこのVIをパスします。 65535: All 1: AnalogIn 2: AnalogOut 4: DigitalIn 8: DigitalOut 16: Virtual 32: OpticalIn 64: CanIn 128: CanRaw
RemoveNotMeasurableSignals	デフォルトはfalseです。計測できない信号はVIをパスします。計測可能な信号のみを使用するには、これをtrueに設定します (AnalogOutSignalを使用しないなど)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。
RemoveDeactivatedSignals	デフォルトはtrueです。無効化された信号は適切に構成されていない可能性があります。
RemoveActivatedSignals	デフォルトは falseです。trueに設定すると、無効化された信号のみをパスします。

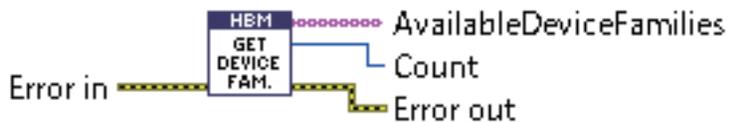
戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

注意: (DevicesAndSignalsクラスタ出力の)Signalsのリファレンス(.net - CloseReference.vi)を閉じる必要があります。そうしないと、このviを頻繁に使用した後にメモリリークが発生します。

GetAvailableDeviceFamilyNames.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

利用可能なすべてのデバイスファミリ(デバイスドライバ) のリストを確定します。



引数	説明
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
AvailableDevices	利用可能なデバイスファミリ名の配列。
Count	利用可能なデバイスファミリの数。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

GetChannelFromSignal.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定した信号が属するチャンネルを検索します。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	DevicesAndSignalsのSignals配列から出力された、どのチャンネルに属しているかを確認するための信号。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Channel	信号が属しているチャンネル。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

GetDeviceFromSignal.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定した信号が属するデバイスを検索します。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	DevicesAndSignalsのSignals配列から出力された、どのチャンネルに属しているかを確認するための信号。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Device	信号が属しているデバイス。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

GetIpAddressFromConnectionInfo.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

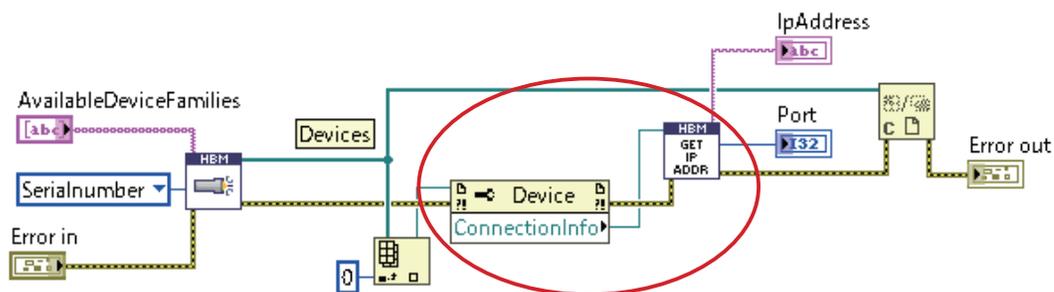
接続情報のIPアドレスとポート番号を取得します。



引数	説明
ConnectionInfo	ScanForDevices.viから出力された、Devices配列のConnectionInfoプロパティ。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
IpAddress	接続するデバイスのIPアドレス。
Port	接続するデバイスのポートTCPポート。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

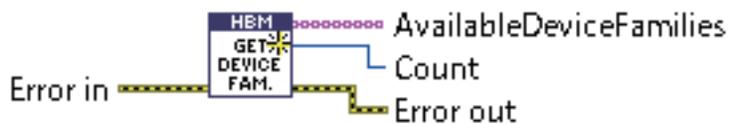
使用例



GetScanableDeviceFamilyNames.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

利用可能なすべてのデバイスファミリ(デバイスドライバ) のリストを確定します。



引数	説明
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
AvailableDeviceFamilies	利用可能なデバイスファミリ名の配列。
Count	利用可能なデバイスファミリの数。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

GetSignals.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定したデバイスの(すべての)信号を付与します。



引数	説明
Devices	DevicesAndSignalsのDevices配列からの入力。
FirstSignalsOnly	各チャンネルのそれぞれの最初の信号のみを取得する場合は、true(デフォルト)に設定します (falseに設定した場合、QuantumXデバイスはチャンネルごとに2つの信号を返します)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

GetSynchronizationQuality.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

オフセット(同期時間ソースとデバイスの時間の間のタイムスパン)をミリ秒単位で返し、TimeSourceに依存する文字列と同期の品質に関する詳細情報を返します。

NtpTimeSourceの場合、この文字列には次のパラメータ(カンマ区切り)が含まれます: "remote, refid, st, t, when, poll, reach, lay, offset, jitter"。

PtpTimeSourceの場合、この文字列には次のパラメータ(カンマ区切り)が含まれます: "GrandmasterID, SyncMode, TimeScaleFlags, UtcOffset, UtcOffsetused, MasterOffset"。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
DevicesIndex	タイムソースの品質情報を取得するデバイスのインデックス。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
OffsetInMs	ミリ秒単位のタイムソースへのオフセット(double: 倍精度浮動小数点数)。オフセットを決定できない場合はMaxValue。
Quality	TimeSourceタイプに依存する同期品質に関する情報。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

Init.vi

HBM LabVIEW Driver	VI	System
--------------------	----	--------

基礎となるCommon API を初期化します。

このVIをLabVIEW IDE内で実行する場合、指定したライセンスファイルを検証します。



引数	説明
LicenseFileName	ライセンスファイルのフルパスとファイル名。この文字列が空の場合、ライセンスファイルはDLLディレクトリ内を検索します(検索パターンは"*.license")。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
LicenseIsValid	指定されたライセンスファイルが有効であるか、LabVIEW Runtimeを使用している場合はTrue。
LicenseInfo	ライセンス所有者に関する情報(ライセンスが評価ライセンスでない場合)。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

Problems.vi

HBM LabVIEW Driver	VI	System
--------------------	----	--------

問題にエラーが含まれている場合(割り当て関数などによる)は、エラーを生成します。

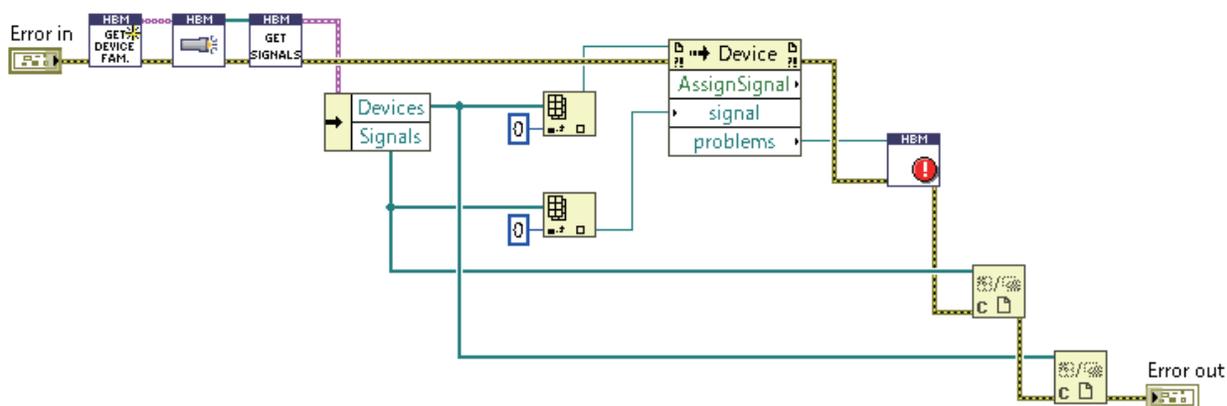
問題に警告は含まれるがエラーは含まれない場合、警告を生成します。



引数	説明
Problems	ScanForDevices.vi から出力された、Devices配列のAssign...メソッドからの入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

使用例

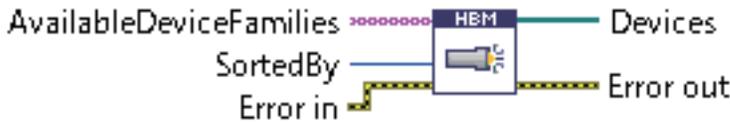


ScanForDevices.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

デバイスをスキャンします。さらにデバイスを見つけるには、複数回繰り返します。

注意: Init.vi の実行後、スキャンメカニズムをサポートするすべてのデバイスを収集し、このVIから返されるまで通常6秒程度待つ必要があります。したがって、ScanForDevices.viを1回だけ使用したい場合(非対話型ワークフロー内など)、利用可能なすべてのデバイスを確実に取得するために、Init.viの実行から6秒が経過したことをアサートする必要があります。



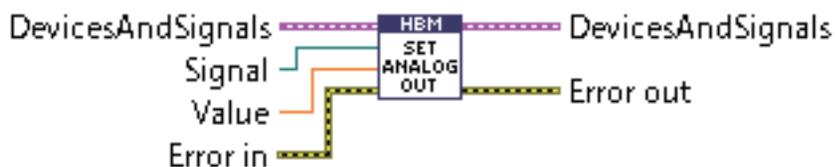
引数	説明
AvailableDeviceFamilies	GetAvailableDeviceFamilyNames.vi から出力された、デバイスファミリ名の配列。
SortedBy	見つかったデバイスの並べ替えに使用するプロパティ。を定義する列挙体。 SerialNumber: シリアル番号 Name: 名前 IpAddress: IPアドレス
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Devices	接続するデバイスの配列出力。 指定したデバイスファミリ名と並べ替え設定に従って、デバイスを検索して並べ替えます。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

SetAnalogOut.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

アナログ出力信号を指定した値に設定します。



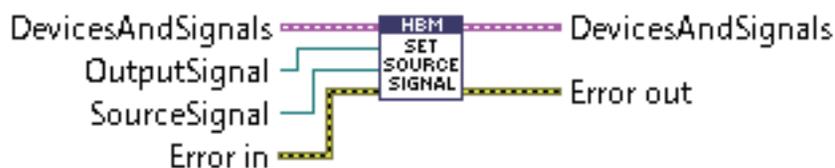
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	DevicesAndSignalsのSignals配列から出力された、設定したい信号(AnalogOutSignalタイプである必要があります)。
Value	設定する値。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

SetAnalogOutSourceSignal.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

アナログ出力信号のソース信号を指定したソースに設定します。



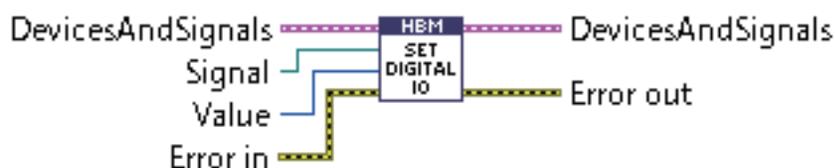
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
OutputSignal	設定するアナログ出力信号(AnalogOutSignalタイプである必要があります)。
SourceSignal	計測値を、指定した出力信号の物理コネクタで出力電圧に変換する必要がある信号(出力信号の出力スケール設定による)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

SetDigitalIO.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定された値に従ってデジタル信号を設定します。



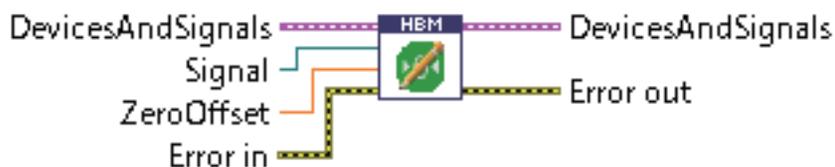
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	DevicesAndSignalsのSignals配列から出力された、設定したい信号(DigitalSignalタイプである必要があります)。
Value	DigitalValueType(LowまたはHigh)を設定します。 0: Low 1: High
Error in	エラーのステータス(プール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(プール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

SetZeroOffset.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

指定したアナログ入力信号のゼロオフセットを指定の値に設定します。



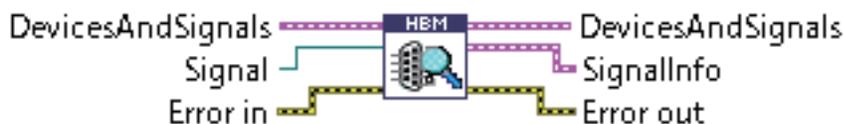
引数	説明
DevicesAndSignals	GetSignals.viから出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	ゼロオフセットを設定する必要があるアナログ入力信号(実際にはチャンネルのゼロオフセットが設定されるため、同じチャンネルに属する他の信号も影響を受けることに注意してください)。
ZeroOffset	使用するゼロオフセット値。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

SignalInfo.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

さまざまな信号のプロパティを提供します。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	DevicesAndSignalsのSignals配列から出力された、プロパティを読み取る必要がある信号。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
SignalInfo	信号に関する情報を含むクラスタ。下の表を参照してください。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

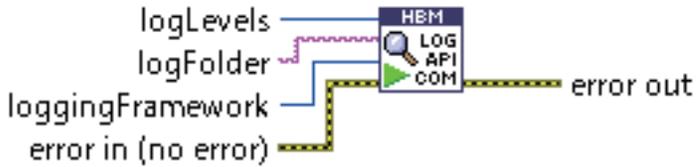
SignalInfoクラスタ	説明
Name	信号名。
SynchronMode	信号が計測値を送信する方法を定義します。 Equidistant(等距離): 1秒内の計測値の数 = サンプルレート NonEquidistant(非等距離): 計測値は予測できない時間に配信される
SampleRate	連続計測中に計測値を収集するために使用するサンプルレート。
FilterFrequency	周波数(Hz)。
UpdatedValueCount	最後のHbm.Api.Common.DaqMeasurement.FillMeasurementValues呼び出し以降の新しい計測値の数。
Values	計測値。
Timestamps	タイムスタンプ。計測開始からの秒数(DaqMeasurement.MeasurementStartTime、DaqMeasurement.MeasurementStartUTCTime)。
States	値の状態。
BufferOverrunOccurred	基になる循環バッファがバッファオーバーランを登録した場合、True。これは、計測値の取得時間が長すぎる場合に発生します。基礎となる循環バッファ内の"最も古い"値は、最後のリクエスト以降に上書きされます。
EngineeringUnit	物理単位(例: "kN"や"mm")。
ChannelName	チャンネル名。

StartLogging.vi

HBM LabVIEW Driver	VIs	System
--------------------	-----	--------

このVIは、このLabVIEW Driverによって使用されるHBM Common APIのアクションを記録します。このドライバのVIを使用するときに発生するエラーや結果に関連する問題を見つけるのに役立ちます。

このVIを使用するとプログラムの速度が低下します。HBMサポートから要請されるまでは使用しないでください。



引数	説明
LogLevel	どのアクションをログに記録するかを決定します。完全なログを記録するには"All"を選択します: 0: Off 1: All 2: Trace 4: Debug 8: Info 16: Warn 32: Error 64: Database
LogFolder	ログファイルを作成するディレクトリ名。
LoggingFramework	ロギングに使用するフレームワーク(以下をサポートしています): 0: Loupe 1: NLog
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。
戻り値	説明
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

DAQグループ



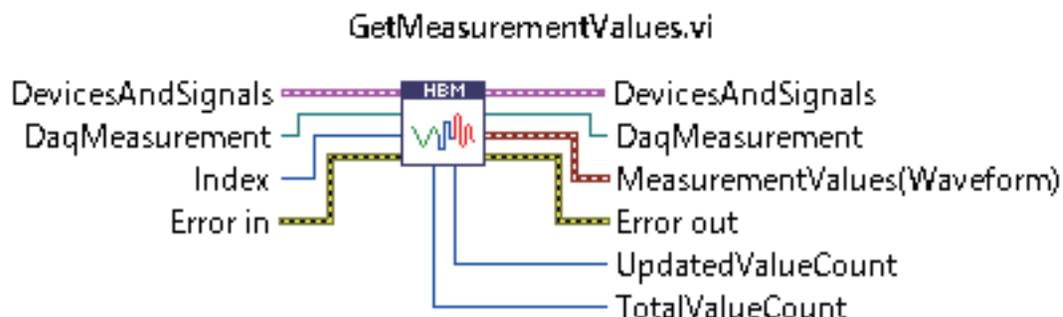
このグループには、データ収集に役立つすべてのVIが含まれています。以下をサポートするVIがあります：

- 連続計測の準備
- 連続計測の開始と停止
- 継続的な測定値の取得
- 連続測定を実行せずに単一の測定値を取得

GetMeasurementValues.vi

HBM LabVIEW Driver	VIs	DAQ
--------------------	-----	-----

計測値をLabVIEW標準の波形形式で提供します(utc時間形式の計測値の最初のタイムスタンプ、1/サンプルレート(Hz)、および計測値の配列)。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス入力。
Index	計測値を提供する、指定した信号配列内の信号のインデックス。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

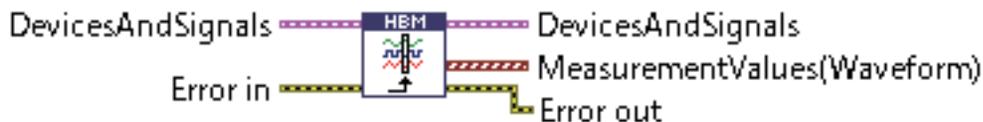
戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス出力。
MeasurementValues	開始時間、 Δt 、波形データ、および属性を含む要素で構成されている計測値。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。
UpdatedValueCount	UpdateMeasurementValues.vi を実行して最後に更新された値の数。
TotalValueCount	指定したインデックスの信号の実行データ収集中の計測値の合計数。

GetSingleMeasurementValues.vi

HBM LabVIEW Driver	VIs	DAQ
--------------------	-----	-----

連続計測を初期化せずに、指定した信号ごとに1つの計測値を取得します。フィルタ設定は無視されます。

GetSingleMeasurementValues.vi



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

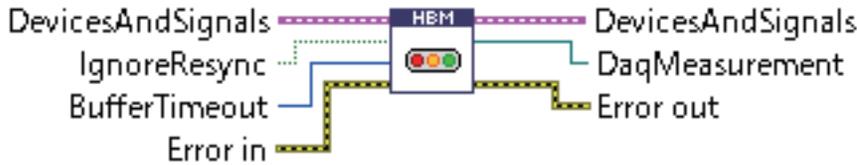
戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
MeasurementValues	開始時間、 Δt 、波形データ、および属性を含む要素で構成されている計測値。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PrepareDAQ.vi

HBM LabVIEW Driver	VIs	DAQ
--------------------	-----	-----

連続計測を準備します。

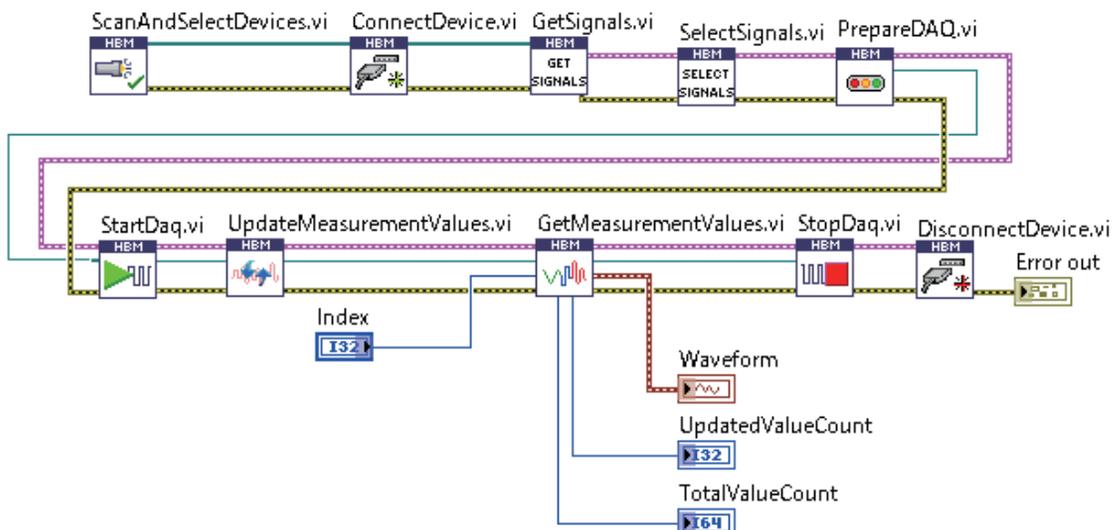
信号を指定したデバイスに属していない場合、またはデバイスが要求されたサンプルレートをサポートしていない場合、エラーがスローされます。



引数	説明
DevicesAndSignals	GetSignals.viから出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
IgnoreResync	データ収集実行中のタイムソースの再同期を無視するかどうかを決定します。現在のところ、ストリーミングデバイス(QuantumXファミリ)でのみ可能です。再同期が検出された場合にデータ取得を停止する必要がある場合はfalseに設定します (デフォルトはtrue - したがって、接続しない場合、再同期は常に無視されます)。
BufferTimeout	ミリ秒単位のバッファタイムアウト(デフォルトは 3000ミリ秒)。内部循環バッファのサイズを計算するために使用されます。信号ごとに少なくとも1000個の値のバッファがあります。通常、バッファサイズは(bufferTimeout/1000) * 信号のサンプルレートで、計測に追加されます(例: Signal.SampleRate=1200Hz, bufferTimeOut=1000ms => 内部循環バッファのサイズは 1200 エントリ)
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
DaqMeasurement	DaqMeasurementオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

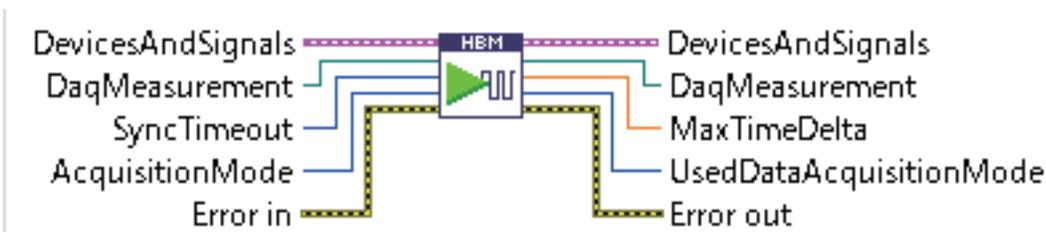
使用例



StartDaq.vi

HBM LabVIEW Driver	VI	DAQ
--------------------	----	-----

計測に追加されたすべての信号の(同期)計測を開始します。



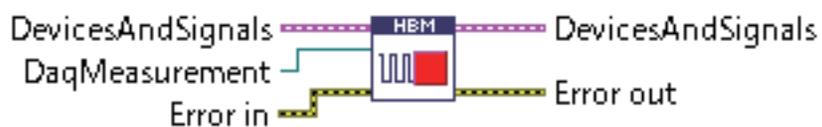
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス入力。
SyncTimeout	同期計測の開始に使用する最大時間(ミリ秒)。この時間内に同期計測を開始できない場合は、エラーがスローされます。
AcquisitionMode	<p>データ取得の方法。</p> <p>0: ソフトウェア同期 1: ハードウェア同期 2: 非同期 3: 自動</p> <p>すべての信号の最初のタイムスタンプをできるだけ互いに近づける必要がある場合は、ソフトウェア同期を使用し、デバイスを同期する必要があります!</p> <p>デバイスがハードウェア(ケーブル)によって同期され、各信号に共通の最初のタイムスタンプを提供する場合は、ハードウェア同期を使用します。</p> <p>すべての信号に共通の最初のタイムスタンプを持たずに測定を開始するには、非同期を使用します(その場合、syncTimeOutは考慮されず、関数は0.0を返します)。</p> <p>APIにどのモードを使用するかを決定させるには、自動モードを使用します。Common APIは最適なモードの使用を試みます。デバイスが相互に同期していない場合は、非同期モードが選択されます。</p>
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス出力。
MaxTimeDelta	最初の計測値のタイムスタンプ間の最大差。0は、各信号の最初のタイムスタンプが同じ値を持つことを意味します。
UsedDataAcquisitionMode	最新の計測を開始するために使用したAcquisitionModeを返します。最新の計測がAcquisitionMode.Autoで開始している場合、使用するデータ収集モードは、選択された実際のモード(非同期、ハードウェア同期、またはソフトウェア同期)になります。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

StopDaq.vi

HBM LabVIEW Driver	VIs	DAQ
--------------------	-----	-----

計測に参加するすべてのデバイスのデータ収集を停止します。



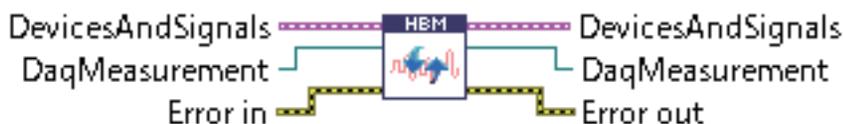
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

UpdateMeasurementValues.vi

HBM LabVIEW Driver	VIs	DAQ
--------------------	-----	-----

信号の計測値を更新します。計測中に定期的にこの関数を呼び出します。



この関数は、計測に関与するすべての信号のすべての計測値を更新します。

データ収集が同期して開始される場合、同じサンプルレート(各デバイスに割り振られた)の各信号が同数の新しい計測値を取得することをアサートします。たとえば、サンプルレート20Hzのデバイス1のsignal_Aは、サンプルレート20Hzのデバイス2のsignal_Bと同じ数の新しい計測値を取得します!

それ以外の場合(StartDaqが非同期計測を開始した場合、信号はこの関数の最後の呼び出し以降に蓄積されたすべての計測値を取得します。

この関数は、各信号の計測値に十分なメモリが割り当てられていることもアサートします。

引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
DaqMeasurement	DaqMeasurement オブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

QuantumXグループ

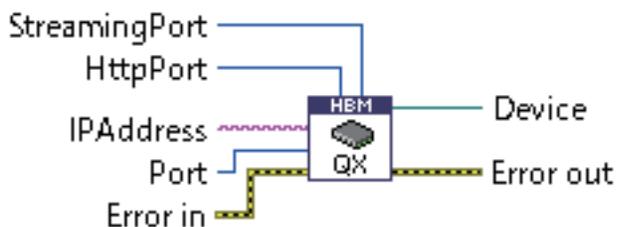


このグループには、QuantumXデバイスの追加関数(特定のコネクタでの点滅など)を実現するVIが含まれています。追加関数は、すべてのデバイスタイプに実装されるCommon関数ではカバーしていません。これらのVIはQuantumXデバイスでのみ使用できます!

QuantumX_Device.vi

HBM LabVIEW Driver	VIs	QuantumX
--------------------	-----	----------

新規のQuantumXデバイスを生成します。



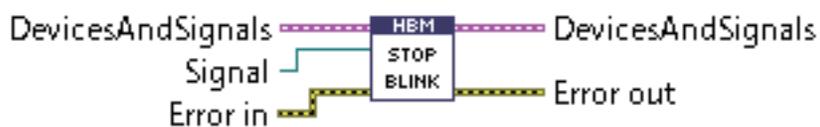
引数	説明
StreamingPort	ストリーミングに使用するポート(デフォルトは7411)。
HttpPort	デバイスのHTTPポート(デフォルトは80)。
IPAddress	接続するデバイスのIPアドレス。
Port	接続するデバイスのポートTCPポート(デフォルトのポートは5001)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

QuantumX_DisableBlinking.vi

HBM LabVIEW Driver	VIs	QuantumX
--------------------	-----	----------

指定した信号が属するコネクタの点滅を無効にします。



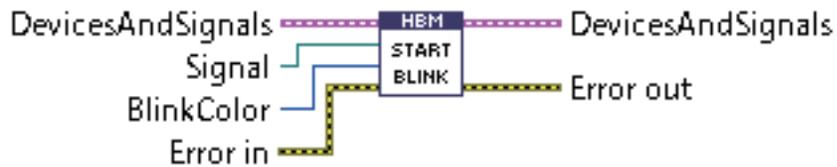
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	コネクタの点滅を停止させたい信号。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

QuantumX_EnableBlinking.vi

HBM LabVIEW Driver	VIs	QuantumX
--------------------	-----	----------

指定した信号が属するコネクタの点滅を有効にします。



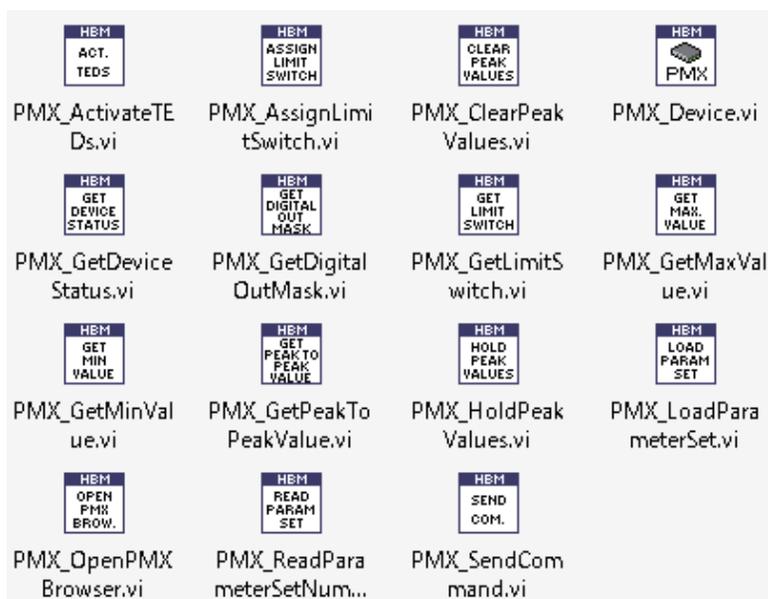
引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Signal	指定した色でコネクタの点滅を開始させたい信号。
BlinkColor	LEDの点滅色。 1: 緑色 2: 橙色 3: 赤色
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMXグループ

このグループには、PMXデバイスの追加関数(特定のパラメータセットのロードや特定のコネクタでのTEDのアクティブ化など)を実現するVIが含まれています。追加関数は、すべてのデバイスタイプに実装されるCommon関数ではカバーしていません。

これらのVIはPMXデバイスでのみ使用できます!

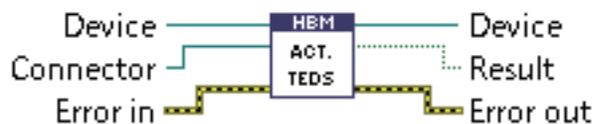


PMX_ActivateTEDs.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

指定したコネクタでTED設定をロードしてアクティブにします。

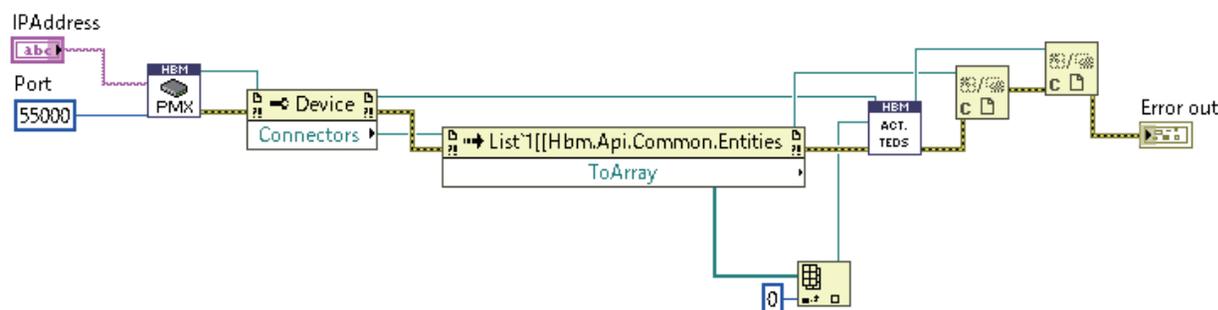
成功すると、指定したコネクタのセンサオブジェクトは、TEDの設定に従って更新されたバージョンに置き換えられます。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Connector	TEDが接続されるコネクタ。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Result	TED設定をロードしてアクティブ化できる場合はTrue。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

使用例

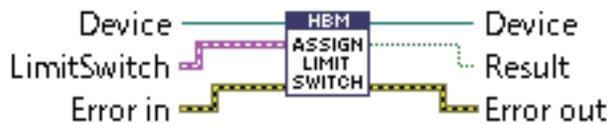


PMX_AssignLimitSwitch.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

LimitSwitch設定を指定したデバイスに割り当てます。

すでに使用されているLimitSwitchNumberを持つリミットスイッチが割り当てられている場合、その既存のリミットスイッチは上書きされます。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
LimitSwitch	割り当てるリミットスイッチを構成するクラスタ。 下のLimitSwitchクラスタの表を参照してください。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

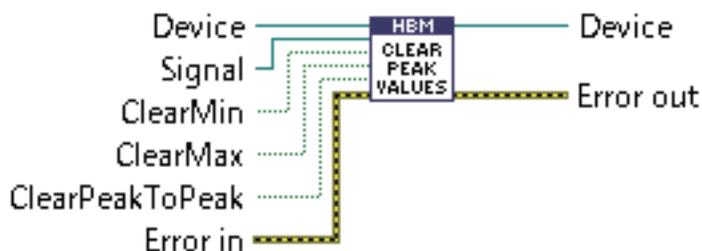
LimitSwitchクラスタ	説明
Enabled	リミットスイッチの動作を有効にするかどうかを決定します。
Hysteresis	ヒステリシス値。OperationDirectionに応じて、帯域幅を定義するために使用します。
IgnoreMeasurementValueStatus	リミットスイッチステータスの評価時に計測値のステータスを無視するかどうかを決定します。
InputSignal	計測値がリミットスイッチの評価するために使用する信号。
InvertResetBehaviour	trueの場合、定義されたResetBehaviorMaskを反転します。
Limit	限界値。OperationDirectionに応じて、下位バンド値を定義するために使用することもできます。
LimitSwitchNumber	割り当てるリミットスイッチの番号(1...32)
OperatingDirection	リミットスイッチの動作方向。
ResetBehaviorMask	動作をリセットします。すべてのデジタル入力とAND演算されるバイナリマスク。リセット動作は、InvertResetBehaviorを使用して反転できます。デフォルト設定は0です。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Result	LimitSwitch設定を指定したデバイスに割り当てが成功した場合、True。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_ClearPeakValues.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

指定した信号のピーク値(最小、最大、最大振幅)をクリアします。



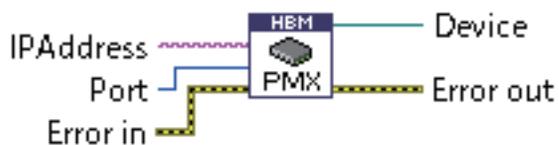
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Signal	ピーク値を保持した信号。
ClearMin	最小値をクリアしたい場合はTrueに設定します(デフォルト値はTrue)。
ClearMax	最大値をクリアしたい場合はTrueに設定します(デフォルト値はTrue)。
ClearPeakToPeak	最大振幅をクリアしたい場合はTrueに設定します(デフォルト値はTrue)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_Device.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

新しいPMXデバイスを生成します。



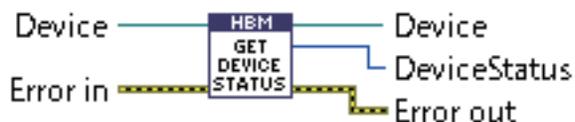
引数	説明
IPAddress	接続するデバイスのIPアドレス。
Port	接続するデバイスのポート(デフォルトのポートは55000)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_GetDeviceStatus.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

デバイスの現在のステータスを返します。



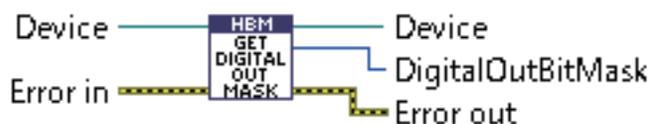
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Error in	エラーのステータス(プール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
DeviceStatus	ステータスは以下の通りです: 0: NoError(エラーなし) 1: FactorySettingsError(工場出荷時の設定エラー) 2: SyncMaster(同期マスタ) 4: SyncMessageError(欠落または無効な同期メッセージ) 8: SyncUnlockedSlaveError(ロックできません。レギュレータはマスタと同期できません) 16: Alive(有効な接続、~1Hz で切り替え) 32: PowerOverLoad(電源の過負荷が発生) 64: CatBufferOverrun("catman" インタフェースのバッファオーバーラン) 128: SystemNotReady(システムの準備できていません。例: 進行中のパラメータセットの変更に失敗など) 256: DSPOverRun(DSPオーバーラン。例: 演算チャンネルが多すぎるなど)
Error out	エラーのステータス(プール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_GetDigitalOutMask.vi

HBM LabVIEW Driver	VI	PMX
--------------------	----	-----

PMXデバイスのデジタル出力ポートを読み取ります(16ビット)。



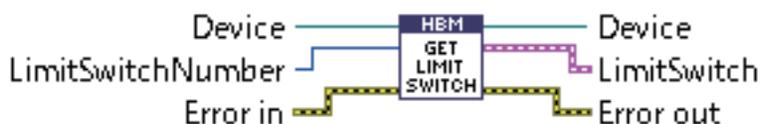
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
DigitalOutBitMask	16ビットデジタル出力(0=オフ、1=オン)
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_GetLimitSwitch.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

指定されたリミットスイッチ番号のリミットスイッチ情報を取得します。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
LimitSwitchNumber	問い合わせるリミットスイッチの番号(1 ...32)
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
LimitSwitch	リミットスイッチを構成するクラスタ。 下のLimitSwitchクラスタの表を参照してください。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

LimitSwitchクラスタ	説明
Enabled	リミットスイッチの動作を有効にするかどうかを決定します。
Hysteresis	ヒステリシス値。OperationDirectionに応じて、帯域幅を定義するために使用します。
IgnoreMeasurementValueStatus	リミットスイッチステータスの評価時に計測値のステータスを無視するかどうかを決定します。
InputSignal	計測値がリミットスイッチの評価するために使用する信号。
InvertResetBehaviour	trueの場合、定義されたResetBehaviorMaskを反転します。
Limit	限界値。OperationDirectionに応じて、下位バンド値を定義するために使用することもできます。
LimitSwitchNumber	割り当てるリミットスイッチの番号(1...32)
OperatingDirection	リミットスイッチの動作方向。
ResetBehaviorMask	動作をリセットします。すべてのデジタル入力とAND演算されるバイナリマスク。リセット動作は、InvertResetBehaviorを使用して反転できます。デフォルト設定は0です。

PMX_GetMaxValue.vi

HBM LabVIEW Driver	VI	PMX
--------------------	----	-----

指定した信号の最大値を取得します。



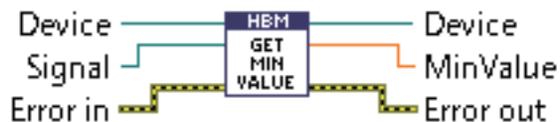
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Signal	最大値を取得する信号のリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
MaxValue	指定した信号の前のクリア以降での最大値。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_GetMinValue.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

指定した信号の最小値を取得します。



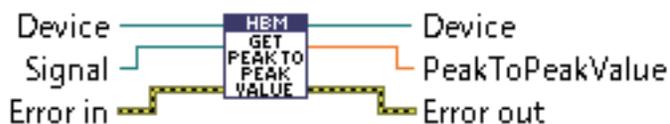
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Signal	最小値を取得する信号のリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
MinValue	指定した信号の前のクリア以降での最小値。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_GetPeakToPeakValue.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

指定した信号の最大振幅値を取得します。



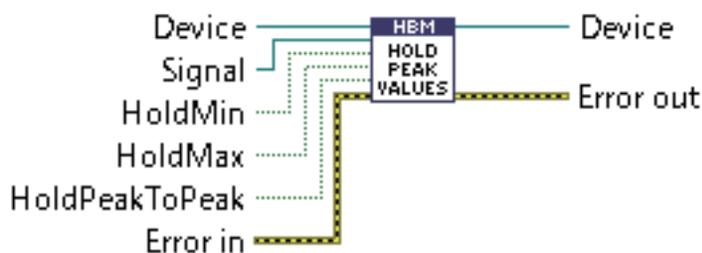
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Signal	最大振幅値を取得する信号のリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
PeakToPeakValue	指定した信号の前のクリア以降での最大振幅値。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_HoldPeakValues.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

指定した信号のピーク値(最小、最大、最大振幅)を保持または有効にします。



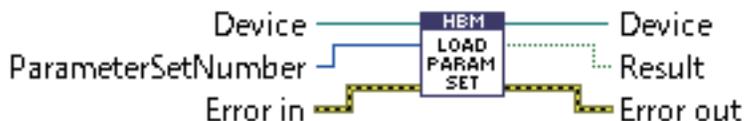
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Signal	ピーク値を有効/無効にする信号。
HoldMin	最小値を保持したい場合はTrueに設定します(デフォルト値はTrue)。
HoldMax	最大値を保持したい場合はTrueに設定します(デフォルト値はTrue)。
HoldPeakToPeak	最大振幅を保持したい場合はTrueに設定します(デフォルト値はTrue)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_LoadParameterSet.vi

HBM LabVIEW Driver	Vis	PMX
--------------------	-----	-----

指定されたパラメータセット番号をロードします。
 パラメータセットが存在し、ロードされている場合、結果はtrue になり、それ以外の場合はfalseになります。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
ParametersetNumber	ロードする必要があるパラメータ セット(0、1、2、3...)。-1は、工場出荷時の設定を現在アクティブなパラメータセットにロードし、アクティブ化することを意味します。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Result	パラメータ セットが存在し、ロードされている場合はTrue。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_OpenPMXBrowser.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

デフォルトのWebブラウザを開き、指定したPMXデバイスの構成パネルを表示します。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_ReadParameterSetNumber.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

現在ロードされているパラメータセットの番号(0、1、2、3... を読み取ります。



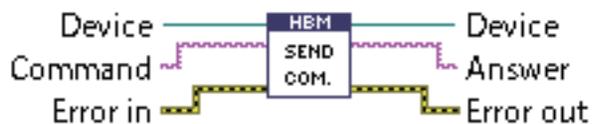
引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
CurrentParametersetNumber	現在ロードされているパラメータ セットの番号(0、1、2、3...)
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

PMX_SendCommand.vi

HBM LabVIEW Driver	VIs	PMX
--------------------	-----	-----

デバイスにコマンドを送信し、その応答を返します。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Command	デバイスに送信するコマンド(詳細はデバイスのマニュアルを参照)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Answer	送信されたコマンドによるデバイスの応答。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

MGCグループ

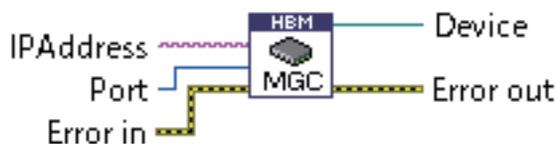


このグループには、MGCデバイスの追加関数(低レベルコマンドの送信など)を実現するVIが62。追加関数は、すべてのデバイス タイプに実装されるCommon関数ではカバーしていません。これらのVIはMGCデバイスでのみ使用できます!

MGC_Device.vi

HBM LabVIEW Driver	VI	MGC
--------------------	----	-----

新規のMGCデバイスを生成します。



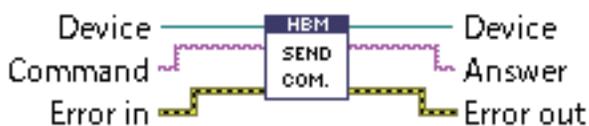
引数	説明
IPAddress	接続するデバイスのIPアドレス。
Port	接続するデバイスのポート(デフォルトのポートは7)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

MGC_SendCommand.vi

HBM LabVIEW Driver	VIs	MGC
--------------------	-----	-----

デバイスにコマンドを送信し、その応答を返します。



引数	説明
Device	Deviceオブジェクトのリファレンス入力。
Command	デバイスに送信するコマンド(詳細はデバイスのマニュアルを参照)。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
Device	Deviceオブジェクトのリファレンス出力。
Answer	送信されたコマンドによるデバイスの応答。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。

User Interfaceグループ



このグループのVIは、対話型で次の目的で使用できます:

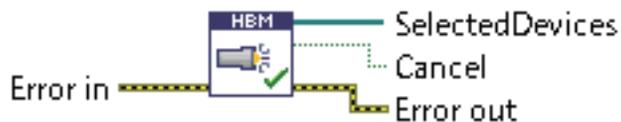
- 特定のデバイスをスキャンする
- 特定のデバイスを接続する
- 使用する信号を選択します
- 特定の信号を計測する
- デジタル出力とアナログ出力を設定する

これらはすべて、対話型操作を可能にする独自のユーザインタフェースを備えています。

ScanAndSelectDevices.vi

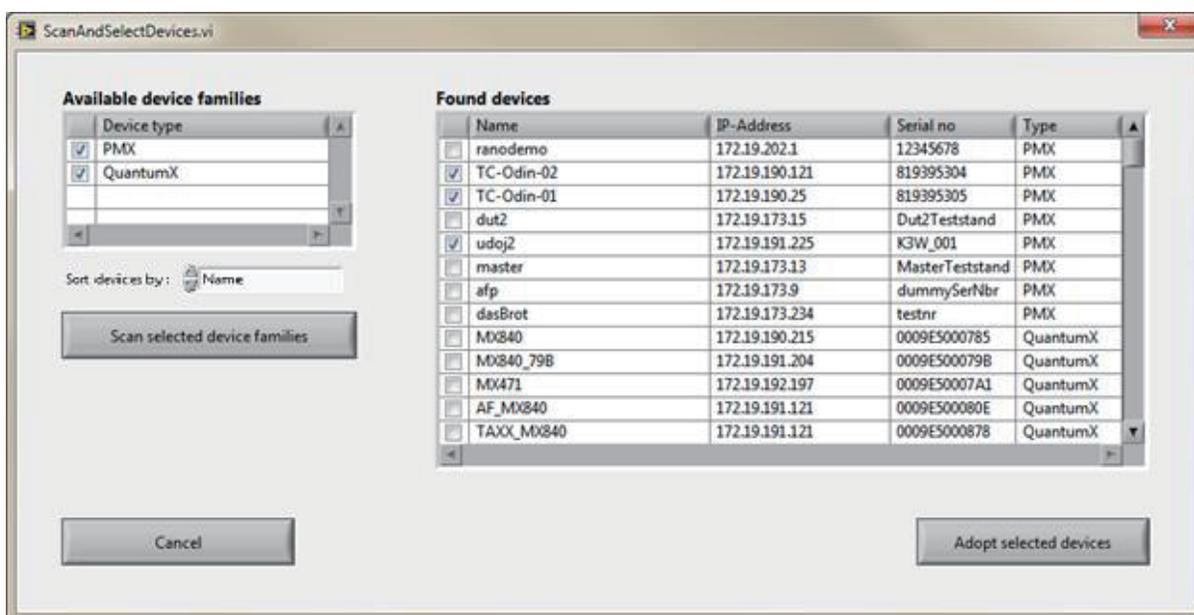
HBM LabVIEW Driver	VIs	User Interface
--------------------	-----	----------------

ネットワークをスキャンして特定のデバイスファミリ(PMXやQuantumXなど)のデバイスを検索し、使用するデバイスを選択します。



引数	説明
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
SelectedDevices	選択したデバイスの配列。
Cancel	キャンセル。ユーザがCancelボタンをクリックするとTrueを出力し、VIの実行を停止します。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。



SelectSignals.vi

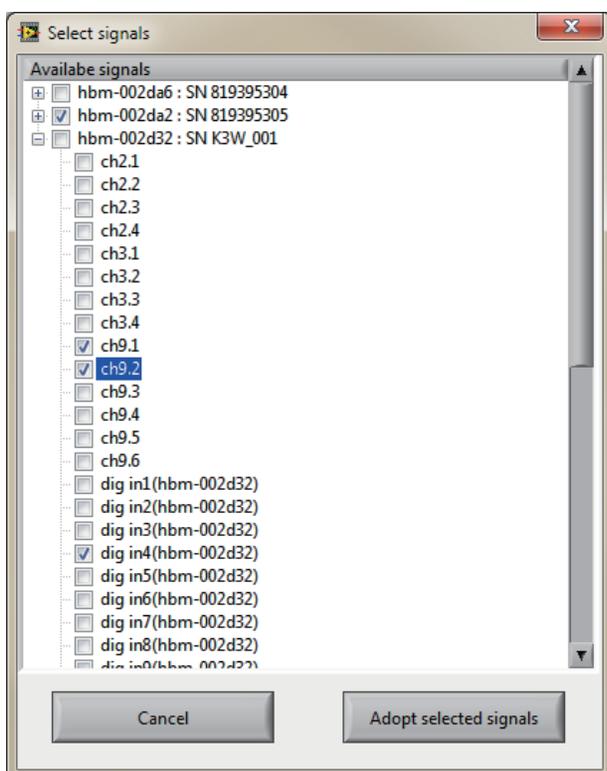
HBM LabVIEW Driver	VIs	User Interface
--------------------	-----	----------------

後で使用する必要がある特定の信号を選択します(計測など)。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

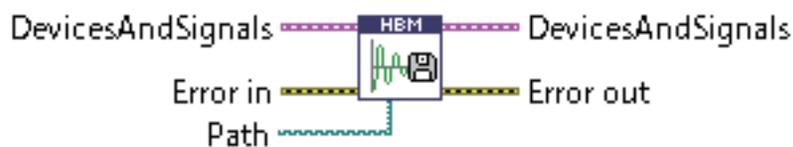
戻り値	説明
DevicesAndSignals	デバイスの配列と 選択した 信号の配列で構成されるクラスタ出力。
Cancel	キャンセル。ユーザがCancelボタンをクリックするとTrueを出力し、VIの実行を停止します。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。



MeasureSignals.vi

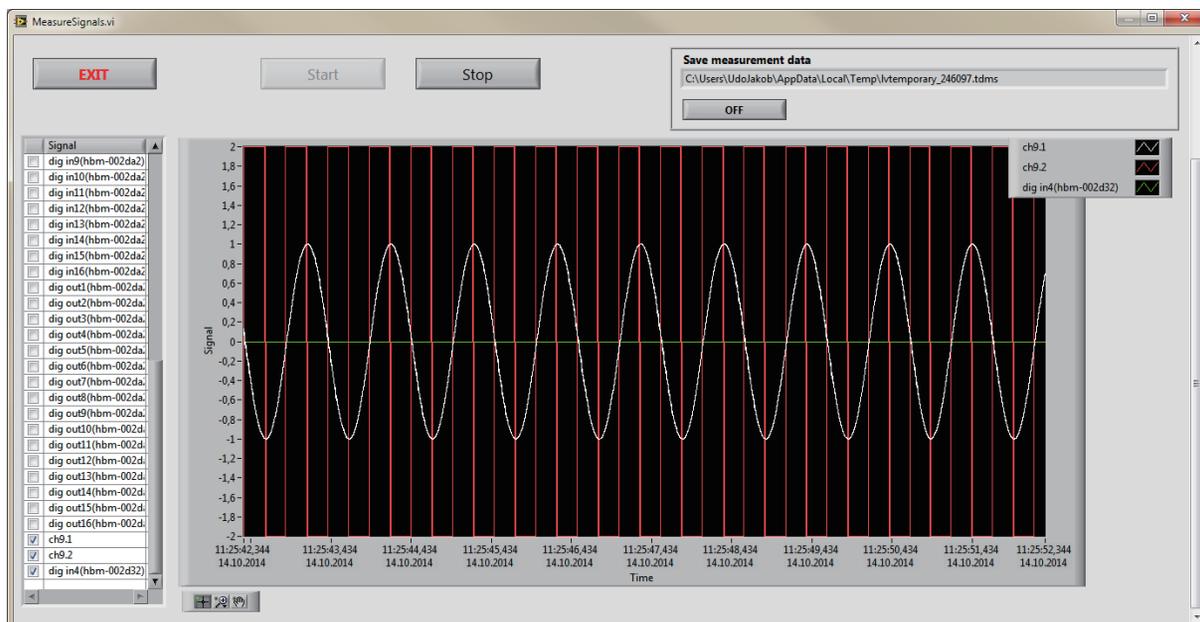
HBM LabVIEW Driver	VIs	User Interface
--------------------	-----	----------------

このVIを使用して、計測および視覚化または保存したい信号を対話的に選択します。



引数	説明
DevicesAndSignals	GetSignals.viから出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。
Path	計測データをの保存先のパス。

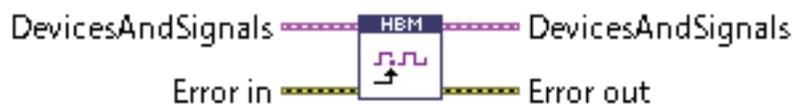
戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。



SetDigitalOut.vi

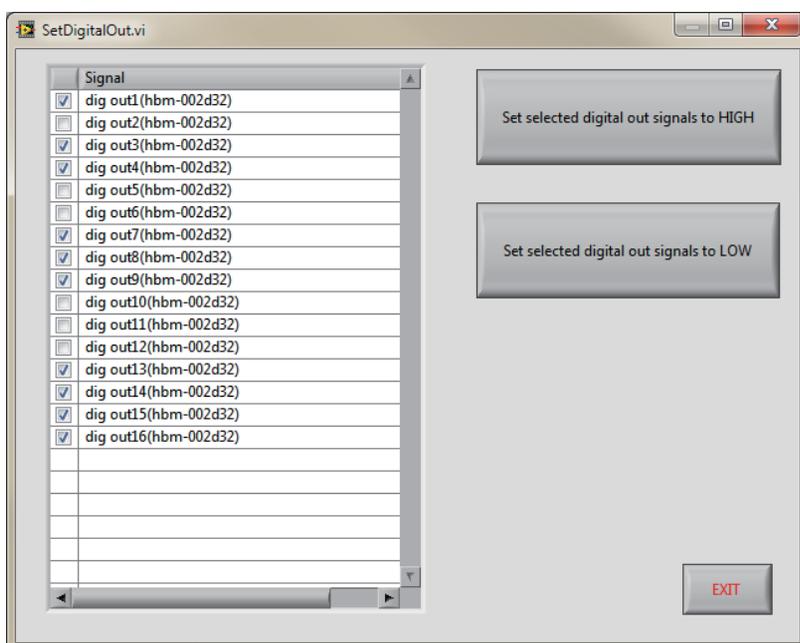
HBM LabVIEW Driver	VIs	User Interface
--------------------	-----	----------------

このVIを使用して、選択したデジタル出力信号をHighまたはLowに設定します。



引数	説明
DevicesAndSignals	GetSignals.vi から出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。



SetAnalogOutUI.vi

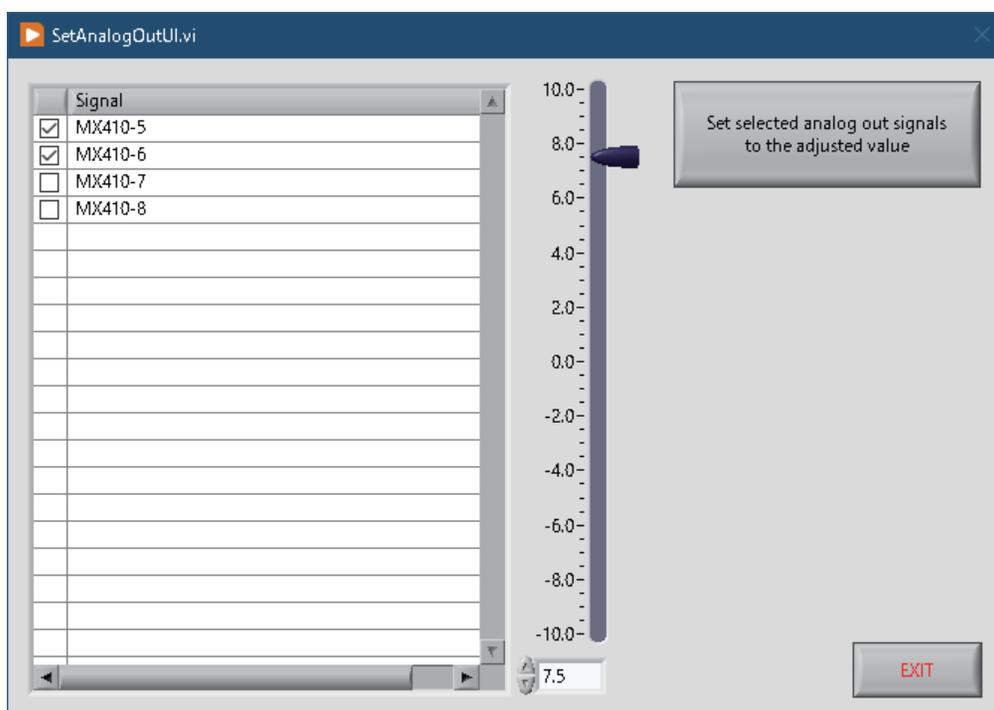
HBM LabVIEW Driver	Vis	User Interface
--------------------	-----	----------------

このVIを使用して、選択したアナログ出力信号を指定した値に設定します。



引数	説明
DevicesAndSignals	GetSignals.viから出力された、デバイスの配列と信号の配列で構成されるクラスタ入力。
Error in	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ入力。

戻り値	説明
DevicesAndSignals	デバイスの配列と信号の配列で構成されるクラスタ出力。
Error out	エラーのステータス(ブール)、番号(数値)およびソース(文字列)で構成されるクラスタ出力。



www.hbm.com

HBM Test and Measurement

Tel. +49 6151 803-0

Fax +49 6151 803-9100

info@hbm.com

アドバンオートメーション株式会社

Tel. 03-5282-7047

Fax. 03-5282-0808

info-advan@adv-auto.co.jp

measure and predict with confidence

