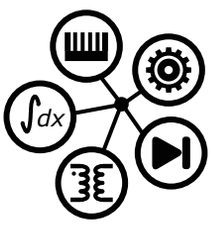


RT Box

DEMO MODEL

Data Logging Demo

データロギングデモ

- RT Boxのシミュレーションデータを記録するさまざまな方法 -

Last updated in RT Box Target Support Package 2.2.1

Advancing Automation

ADVANMATION AUTO

アドバン オートメーション株式会社

info-advan@adv-auto.co.jp www.adv-auto.co.jp

1 はじめに

モデルをリアルタイムで実行しているRT Boxは外部モードに接続し、波形を可視化するためのPLECSスコープを回路内に配置することができます。

多くの場合、ユーザはさらなる処理のためにさまざまなシミュレーションデータをログに記録したいと考えます。RT Boxは、さまざまなプロトコルを介してこれを実行できます。このデモモデルでは、次の5つの方法を紹介します:

- To File - RT Box 2または3の内蔵SSD、またはすべてのRT BoxのプラグインUSBスティックに`.csv`や`.mat`ファイルを書き込み。
- Data Capture - XML/JSON-RPCプロトコル経由で、クライアントはPythonスクリプトで実装。
- UDP - UDP経由でクライアントはPythonスクリプトで実装。
- XCP - 電子制御ユニット(Electronic Control Unit: ECU)のメモリへの読み取りおよび書き込みアクセス用のインタフェース。CANape[1]は自動車業界で広く使用されているXCPマスタです。
- PLECSスコープ - PLECSスコープから外部モードでキャプチャした波形データをエクスポートすることも可能。

注意 このモデルには、次からアクセスできるモデル初期化コマンドが含まれています:
PLECS Standalone: **シミュレーション**メニュー -> **シミュレーション・パラメータ...** -> **初期化**
PLECS Blockset: **Simulinkモデルウィンドウ**で右クリック -> **モデル プロパティ** -> **コールバック** -> **InitFcn***

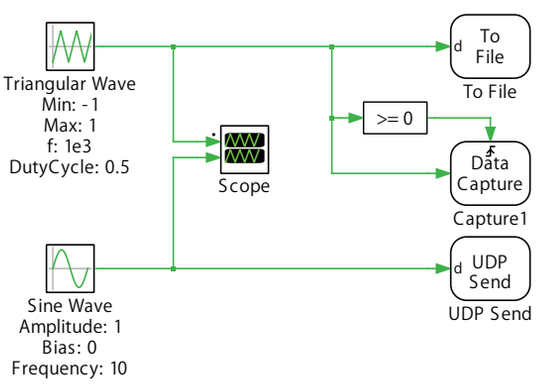
1.1 要求仕様

- PLECS RT Box1台とPLECSおよびPLECS Coderライセンス1つづつ
- RT Box Target Support Package(2.2.1以降)
- RT Box User Manualのクイックスタートガイドに記載されている、PLECSとRT Boxの設定手順に従います

2 モデル

回路図を図1に示します。

図1: データロギングデモの回路図



- To Fileブロックは、RT Boxの離散化ステップサイズに達するまで、数マイクロ秒のサンプルステップまでデータを連続的に書き込むことができます。

- Data Captureブロックは、データポイントが RT Boxの離散化ステップサイズであるデータパッケージを転送しますが、各データパッケージにはサンプル数が制限されています。
- UDPブロックは、設定されたサンプル時間(通常は数msの範囲)に従ってデータパケットを送信します。
- XCPは、RT Boxの離散化ステップサイズでデータを送信します。

概要を表1にまとめます。

表1: RT Boxのデータロギング方法の比較

Method	Sample Time (min.)			Buffer	Continuous Streaming	Physical Layer	Host Inferface (demonstrated)
To File	RT Box 1/CE	.csv	100 µs	No	Yes	-	-
		.mat	10 µs				
	RT Box 2/3	.csv	10 µs				
		.mat	2 µs				
Data Capture	RT Box disc. step			Yes	No	Ethernet	Python
UDP Send	1ms			No	Yes	Ethernet	Python
XCP	RT Box disc. step			Yes	Yes	Ethernet	CANape
PLECS Scope (External Mode)	RT Box disc. step			Yes	Yes	Ethernet	PLECS

したがってデモとして、かなり高速な1kHzの三角波が、To FileとData Captureブロックによって10 µsのサンプル時間で記録されます。一方、ゆっくり変化する10Hzの**正弦波信号**は、10msのサンプル時間でUDP経由で送信されます。**三角波発生器**と**正弦波信号**の両方は**PLECSスコープ**に接続されているため、XCPおよび**PLECSスコープ**の値も外部モードを介して送信されます。表1を参照してください。詳細については、To Fileブロックのヘルプページを参照してください。

2.1 To File

このブロックは:

- RT Box 1およびCEで動作し、外部USBフラッシュドライブにのみ接続できます。
- RT Box 2および3では内部SSDまたは外部USBフラッシュドライブに接続できます。

To Fileブロックは、RT Box上でシミュレーションが実行されている間、値を継続的にファイルに書き込みます。**File type**は`.csv`または`.mat`形式のいずれかです。このブロックの**Sample time**は、入力信号がファイルに書き込まれる時間ステップを定義します。ファイル形式が異なると、RT Boxごとに最小サンプル時間が異なります。

このブロックの**Write Device**フィールドは、ファイルを書き込むメディアを定義します。Webブラウザで`http://<rtbox name>/dav`にアクセスすることで、RT BoxとのWebDAV接続を確立できます。`<rtbox name>`をRT Boxのホスト名に置き換えます(例: `rtbox-20b0f7049846.local`)。図2は、WebDAVアクセスによるフォルダ構造の一覧を示しています。

- Internal SSD(内蔵SSD)
  `ssd`フォルダをクリックすると、図3に、このデモを3回実行して停止した後のフォルダ構造の例が表示されます。`latest.txt`という名前のファイルは、最新の実行のディレクトリ名を指定しています。この場合、フォルダ`data_logging_0003`内のファイル`data.csv`に最新の実行のデータが保存されます。

図2: RT Box 2または3の内部SSDとUSBフラッシュドライブの両方にアクセスするWebインタフェース

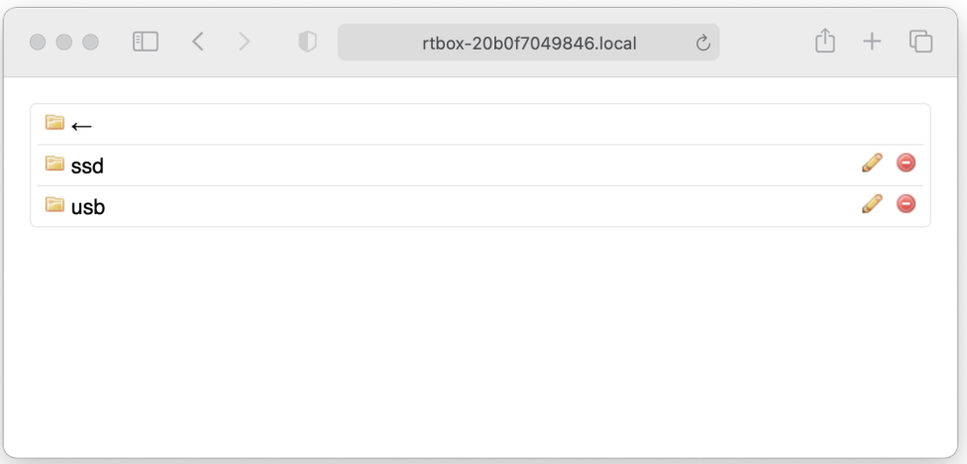
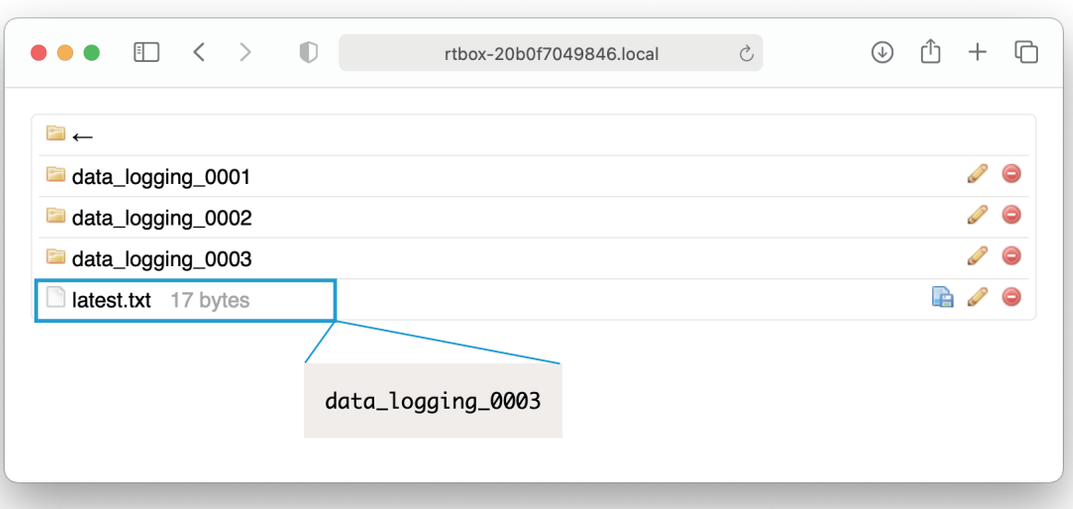


図3: RT Boxの内蔵SSDにアクセスするWebインタフェース



- USB Flash Drive(外付USBフラッシュドライブ)
  USBフラッシュドライブに書き込むと、図3に示すようなフォルダ構造になります。`usb`フォルダをクリックすると、WebDAV経由でUSBコンテンツにアクセスできます。リアルタイムシミュレーションを停止した後、RT BoxからUSBフラッシュドライブを取り外し、PCに接続してファイルを参照することもできます。高いデータスループットを実現するには、USB 3.0デバイスを使用する必要があります。

このデモでは、1kHzの**三角波発生器**のデータが、サンプル時間10 μ sでRT Box 2/3の内蔵SSDに`data.mat`ファイルとして書き込まれます。RT Box上にモデルを構築し、しばらくしてから、RT Box Web Web Interfaceから実行を**Stop**します。

最新の実行のフォルダ内のファイル`data.mat`を確認します。ファイルには、各サンプルステップ間で0.04の増分/減分が観測される1行のデータが必要です。そして、100ポイントごとに1kHzの**三角波発生器**の周期が完了します。

ユーザがRT Box 1のみを持っている場合は、To Fileブロックの設定を**Write Device**: `USB Flash Drive`に変更し、外付USBフラッシュドライブを接続した状態でRT Box 1上にモデルを構築することもできます。生成された`data.mat`には、各データポイント間の増分/減分が0.04の1行のデータが含まれる必要があります。

2.2 Data Capture

Data Captureブロックは、外部スクリプトによって制御されるリアルタイムシミュレーションからデータをキャプチャします。

このブロックによってキャプチャされたすべてのデータは、まず内部バッファに書き込まれます。このブロック内の**Number of samples**はバッファのサイズを定義します。このデモでは、データ キャプチャの開始点は、三角波発生器が0クロスでトリガし、カウントアップします。三角波発生器の正の半サイクルを正確にキャプチャするためにData Captureのバッファサイズは`50`に設定されています。

注意 サンプル数と信号幅の積は65536を超えてはなりません。

このデモのフォルダ内には、`data_logging_datacapture.py`という名前のPythonスクリプトがあります。使用するPythonコードは、Data Captureブロックのヘルプページ内のサンプルスクリプトと似ています。

シミュレーション

Pythonスクリプトを実行する前に、ユーザは次のコードを自分のRT Boxホスト名に合わせて変更する必要があります。

```
HOST_NAME = "rtbox-123.local"
```

以前に RT Box上にモデルを構築していなかった場合は、Pythonスクリプトを実行する前に一度ビルドしてください。

Pythonスクリプトを実行すると、事前に構築されたRT Box実行可能ファイル(`.elf`形式)が読み込まれ、シミュレーションを開始します。Data Captureがトリガされ、`50`個のサンプルがキャプチャされると、リアルタイムシミュレーションは停止します。Pythonスクリプトからの出力は次のようになります:

```
Uploading executable
Starting executable
Real-time simulation running
Waiting for data
Stopping executable
Captured Data are:
[0.0, 0.040000000000000036, 0.08000000000000007, 0.1200000000000001,
0.15999999999999992, 0.19999999999999996, 0.24, 0.28, 0.32000000000000006,
0.3600000000000001, 0.40000000000000013, 0.43999999999999995, 0.48, 0.52,
0.56, 0.6000000000000001, 0.6400000000000001, 0.6799999999999999, 0.72, 0.76,
0.8, 0.8400000000000001, 0.8800000000000001, 0.9199999999999999, 0.96, 1.0,
0.96, 0.9199999999999999, 0.8799999999999999, 0.8400000000000001, 0.8, 0.76,
0.72, 0.6799999999999999, 0.6399999999999999, 0.6000000000000001, 0.56, 0.52,
0.48, 0.43999999999999995, 0.3999999999999999, 0.3599999999999999,
0.32000000000000006, 0.28, 0.24, 0.19999999999999996, 0.15999999999999992,
0.11999999999999988, 0.08000000000000007, 0.040000000000000036]
```

合計50個のデータポイントがコンソールに出力されます。10 μsのサンプル ステップで`[-1,1]`の範囲の1kHzの三角波発生器をキャプチャすると、連続する2つのサンプル間で0.04の増分/減少が観測されます。

2.3 UDP

UDPの転送量は通常、時々パケットが失われても問題にならないストリーミングメディアアプリケーションに使用されています。したがって、リアルタイムシミュレーションでゆっくり変化する信号を送信するのに適しています。

このデモでは、RT Boxは10msの**Sample time**で10Hzの正弦波信号データの情報を含むUDPパケットを送信します。RT Boxは、UDP Sendブロックの**Remote host name or IP address**フィールドで定義されているIPアドレスを持つリモート UDPクライアントにUDPパケットを送信します。このデモでは、RT Boxに接続されたホストPCのIPになります。UDP Sendブロックで構成された**Remote host name or IP address**と**Remote IP port**は、UDPクライアントのPythonスクリプト側に正しく反映される必要があり、次のように記述します:

```
UDP_clientIP = "10.0.0.103" #UDP client IP
UDP_clientPORT = 52345
```

UDPクライアントのPythonスクリプトでは、UDPパケットがアンパックされ、データがPythonコンソールに出力されます。このデモのフォルダ内には、`data_logging_udpclient.py`という名前のPythonスクリプトがあります。

シミュレーション

まず、デモがRT Box上にビルドされ、実行が開始されていることを確認します。

次にPythonスクリプトを実行すると、Pythonコンソールへの出力は次のように続きます:

```
UDP received data: (4.7505016142600914e-14,)
UDP received data: (0.5877852439880371,)
UDP received data: (0.9510565400123596,)
UDP received data: (0.9510565400123596,)
UDP received data: (0.5877852439880371,)
UDP received data: (-4.7505016142600914e-14,)
UDP received data: (-0.5877852439880371,)
UDP received data: (-0.9510565400123596,)
UDP received data: (-0.9510565400123596,)
UDP received data: (-0.5877852439880371,)
UDP received data: (4.7505016142600914e-14,)
UDP received data: (0.5877852439880371,)
...
```

10Hzの正弦波信号を記録するために10msのUDP送信ステップを使用するため、UDP受信データの10データポイントごとに1つの完全な正弦波サイクルを表します。

2.4 XCP

XCP(Universal Measurement and Calibration Protocol)は、校正システムをECUに接続するための、ASAM(Association for Standardization of Automation and Measuring Systems)発のネットワークプロトコルです。XCPの最初の文字Xは、このプロトコルが様々なバスシステム向けに設計されていることを示しています[2]。

RT Box 2および3は、XCPスレーブとして使用して、測定値をXCPマスタにリアルタイムで配信できます。このデモでは、CANapeをXCPマスタとして使用しています。

Coderオプション

XCPは、このデモモデルの **Coderオプション**ダイアログで有効にできます。**ターゲット**タブに移動し、**Enable XCP**オプションにチェックを入れます。その後、**XCP slave identity**フィールドで、`Specify IP address`を選択します。次に、**XCP slave**フィールドに、使用中のRT Box 2または3のIPアドレスを入力します。

ビルドをクリックすると、デモがRT Box上にビルドされ、実行を開始します。デモファイル`data_logging.plecs`と同じディレクトリのフォルダ`data_logging_codegen`内に、`data_logging.a2l`ファイルが生成されます。このファイルをCANape環境にインポートする必要があります。

注意 XCPプロトコルを介して送信される信号は、PLECS回路図内のPLECSスコープに接続されている信号です。

CANape構成

以下のデモではCANapeバージョン20.0を使用しています。

新しいプロジェクトを作成します。Windowsのファイルエクスプローラーで上記の`data_logging.a2l`ファイルを参照します。`data_logging.a2l`ファイルをドラッグし、CANapeを表示し、**Devices**フォルダにマウスを移動して、放します。

図4に示すように、**New network**ボタンをクリックします。

図4: CANapeで新しいネットワークを作成

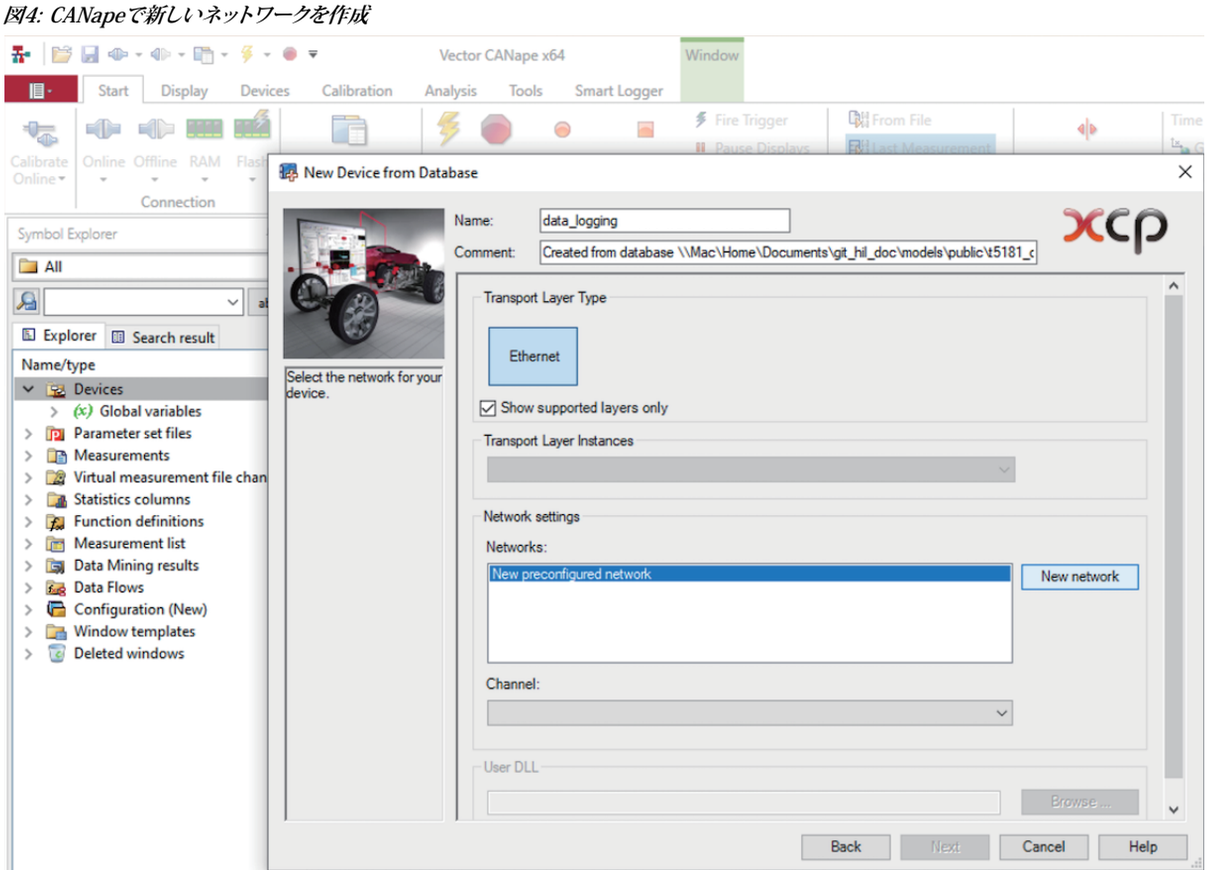
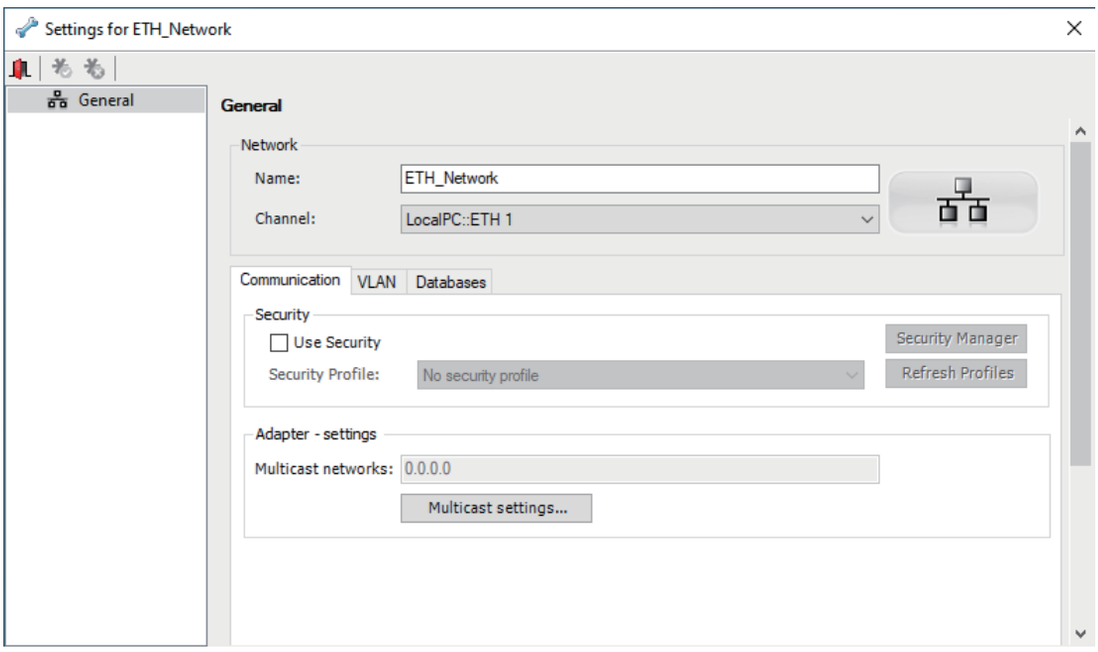


図5: CANapeの新しいネットワークのデフォルト設定



ポップアップ表示された`Settings for ETH_Network`ウィンドウで、図5に示すようにすべてデフォルト設定で、左上隅にある**Close window accept changes**をクリックして変更を適用します。インポートプロセスが完了するまで、**Next**をクリックし続け、最後に**OK**をクリックします。

`Settings for data_logging`というウィンドウが表示されます。以下の2つの属性を変更する必要があります。

Protocol -> Event List -> Expert settingsに移動し、図6に示すように、フィールド**TIME_CORRELATION_GETDAQCLK**をデフォルトの`Multicast`から`Extended response`に変更します。

Protocol -> Transport Layer -> Expert settingsに移動し、図7に示すように、**COUNTER_HANDLING**フィールドをデフォルトの`Include command response`から`Exclude command response`に変更する必要します。

最後に、ウィンドウの左上隅にある**Close window accept changes**をクリックして、すべての変更を適用します。

次に、Project Explorerで、**Devices** -> **data_logging** -> **data_logging.a2l** -> **Scope.Plot_1** -> **Triangular_Wave**と、**Devices** -> **data_logging** -> **data_logging.a2l** -> **Scope.Plot_2** -> **Sine_Wave**をダブルクリックして、両方の信号が CANapeのGraphicウィンドウに追加されるようにします。

これまで、CANapeでの準備作業は完了しました。以前にモデルを実行していなかった場合は、RT Boxでモデルが適切に実行されていることを確認します。CANapeのメニューバーで**Start**をクリックすると、2つの信号の波形がリアルタイムでストリーミングを開始します。Y軸と時間軸の最小/最大値を調整することで、2つの信号の周波数と振幅を簡単に読み取ることができます。

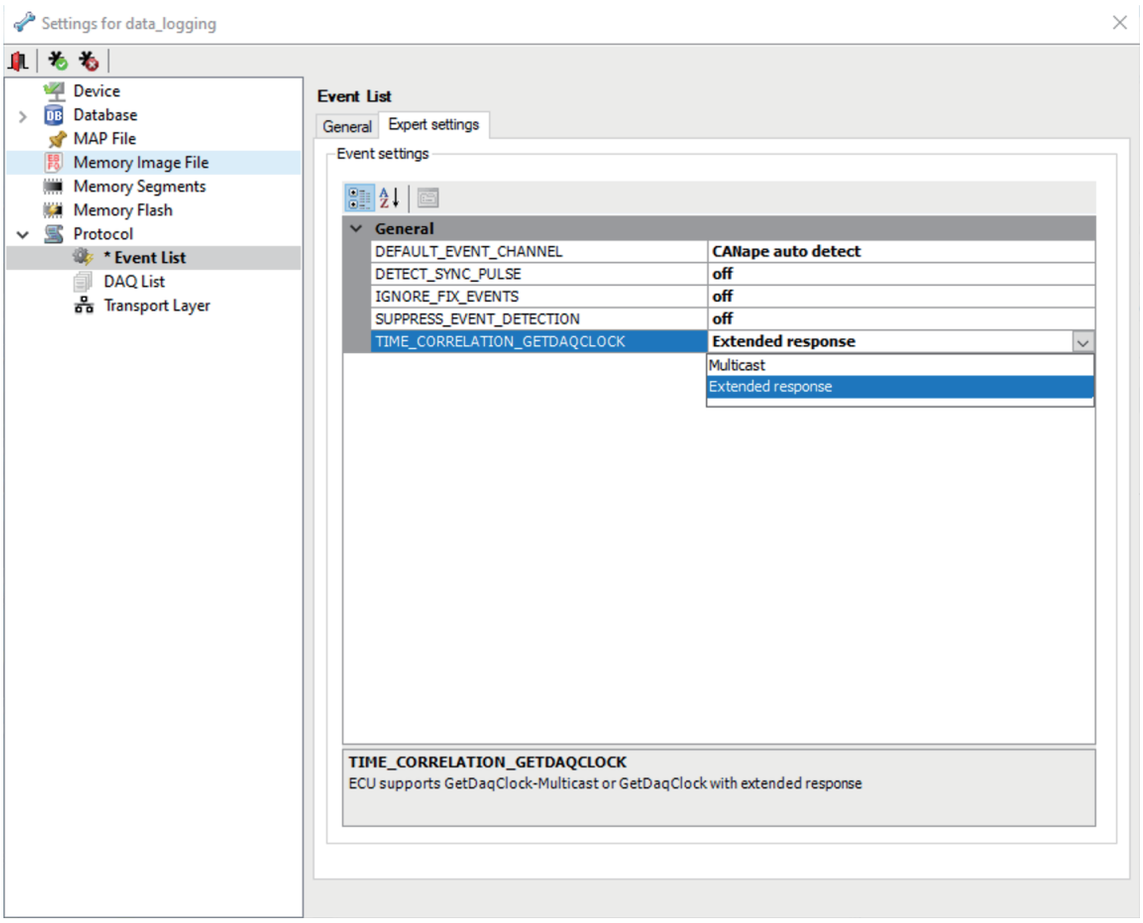
メニューバーの**Start Recording**ボタンをクリックし、しばらくしてから**Stop Recording**をクリックします。データが記録され、エクスポートできる状態になります。

図8は記録された**三角波発生器**と**正弦波信号**のデータの例を示しています。三角波と正弦波の両方のデータ ポイントは、RT Boxの離散化ステップサイズ(10 μs)でキャプチャされます。

2.5 PLECSスコープ

モデルをRT Box上にビルドした後、**外部モード**に**接続**し、**自動トリガを有効化**します。

図6: CANapeのプロジェクト設定におけるProtocol -> Event List -> Expert settingsの変更



このデモでは、**Coderオプション**ダイアログの**外部モード**タブで、**サンプル数**が5000に設定されています。この数値は、外部モードでPLECSスコープに表示されるデータポイントの数を定義します。各ポイント間の間隔は、RT Boxの離散化ステップサイズ(このデモでは10 μs)に設定しています。したがって、PLECSスコープの時間軸の全長は$10\mu s \cdot 5000 = 0.05s$となります。図9に結果を示します。

また、**Decimation**フィールドに整数nを指定して、RT Boxの離散化ステップnごとにダウンサンプリングすることもできます。これにより、同じ**Number of samples**で、有効なPLECSスコープの時間軸が長くなります。

PLECSスコープのデータが更新されないようにするには、**外部モード**の**自動トリガを停止**をクリックします。次に、PLECSスコープのビューで、の**ファイル**メニューから**エクスポート** -> **CSV(カンマ区切り)ファイル** -> **全領域データ...**に移動し、たとえば`data.csv`として保存します。生成されたcsvファイルでは、最初の列にRT Boxの離散化ステップ サイズ(10 μs) ごとに時間軸の値が格納され、その後に三角波データの2番目の列、正弦波データの3番目の列が続きます。

3 まとめ

このデモでは、RT Boxで実行するリアルタイムシミュレーションデータを記録する方法を紹介しました。

4 参考文献

[1] CANape, [Online]. Available: https://www.vector.com/int/en/products/products-a-z/software/canape. [Accessed: Feb. 16, 2022].

[2] ASAM MCD-1 XCP, [Online]. Available: https://www.asam.net/standards/detail/mcd-1-xcp/. [Accessed: Feb. 18, 2022].

図7: CANapeのプロジェクト設定におけるProtocol -> Transport Layer -> Expert settingsの変更

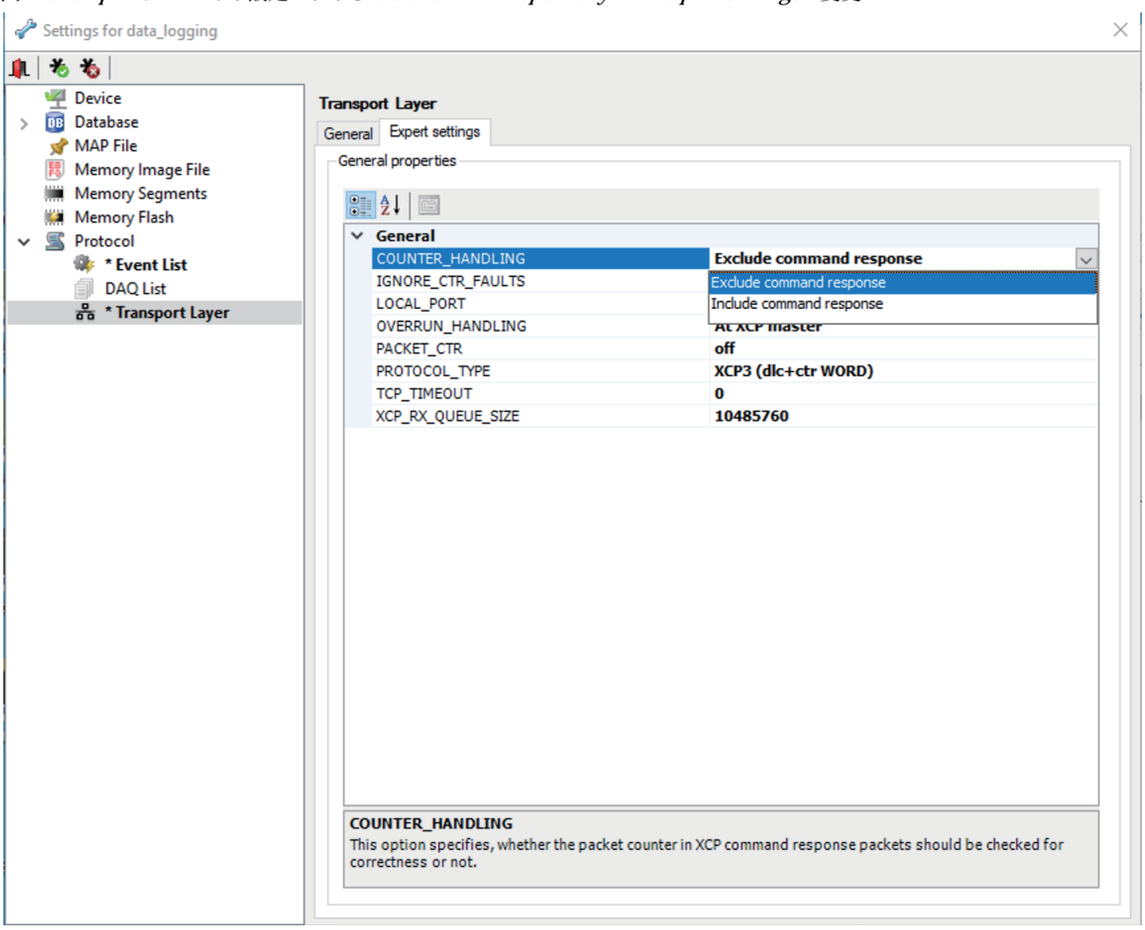


図8: XCPプロトコル経由でCANapeに記録されたリアルタイムシミュレーション結果

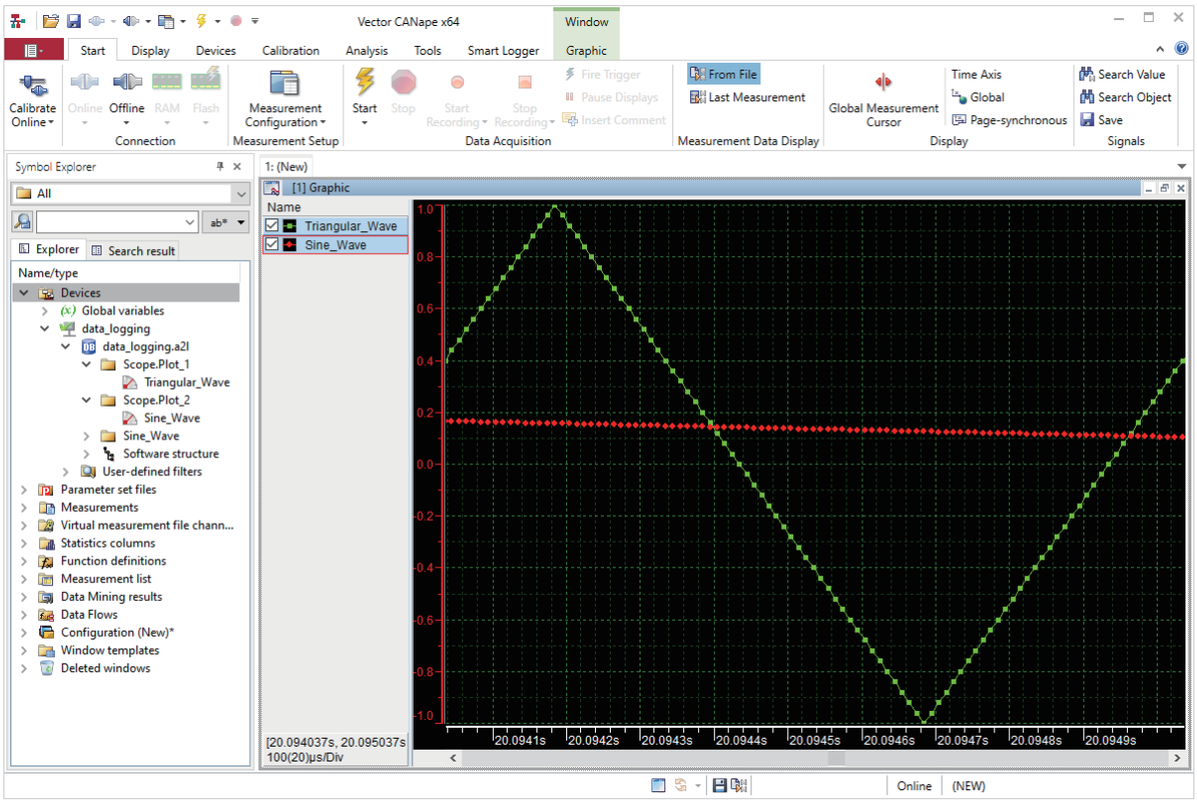
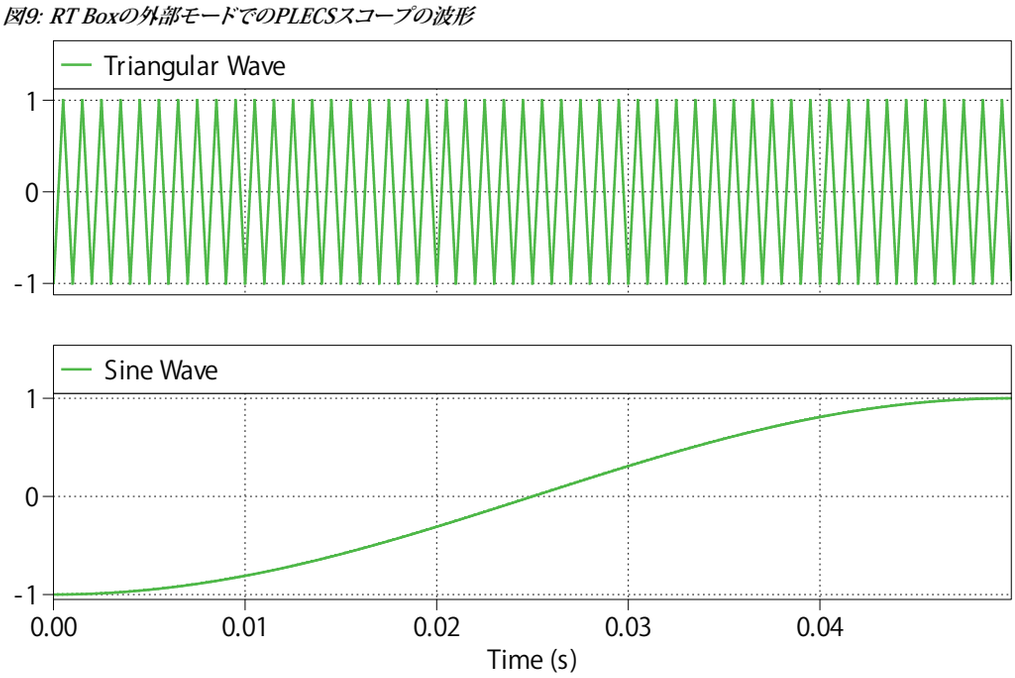


図9: RT Boxの外部モードでのPLECSスコープの波形



改訂履歴:

RT Box Target Support Package 2.2.1　初版

Pleximへの連絡方法:

☎ +41 44 533 51 00　Phone

+41 44 533 51 01　Fax

✉ Plexim GmbH　Mail
Technoparkstrasse 1
8005 Zurich
Switzerland

@ info@plexim.com　Email

http://www.plexim.com　Web

アドバンオートメーションへの連絡方法:

☎ +81 3 5282 7047　Phone

+81 3 6285 0250　Fax

✉ ADVAN AUTOMATION CO.,LTD　Mail
1-9-5 Uchikanda, Chiyoda-ku
Tokyo, 101-0047
Japan

@ info-advan@adv-auto.co.jp　Email

https://adv-auto.co.jp/　Web

RT Box Demo Model