
speedgoat

Setup Guide

R2020b and beyond

Ver.1.0.0 2021年6月16日

目次

1 概要	1
2 構成	1
2-1 Host computerとTarget computerの接続	1
2-2 Target computerとTarget screenの接続	1
3 納品物の確認	2
4 カスタマーポータルサイト	2
4-1 アカウントの作成.....	2
4-2 アカウントの有効化	2
4-3 Account Home	4
4-3-1 Maintenance.....	5
4-3-2 Downloads.....	6
5 開発環境	7
5-1 用意するソフトウェア	7
5-1-1 MathWorks製品	7
5-1-2 Speedgoat製品.....	7
5-1-3 Xilinx製品.....	7
5-2 Speedgoat製品のインストール	8
5-2-1 Speedgoat I/O Blockset.....	8
6 ネットワーク設定	10
6-1 Host computer	10
6-2 Target computer.....	11
6-2-1 Kernelの書き換え	12
7 起動	13
7-1 Target computerの起動.....	13
7-2 Target computerとの接続	14
7-3 Simulinkモデルのビルド.....	15
7-3-1 オシレータモデル	15
7-3-2 サンプルモデル.....	19
8 Control and Instrumentation	20
8-1 MATLAB	20
8-1-1 起動.....	20
8-2 Simulink.....	21
8-3 Simulink Real-time Explorer	22
8-3-1 リアルタイムアプリケーションのロード	23
8-3-2 信号の表示	24
8-3-3 パラメータ設定	25
8-3-4 Data Inspector.....	26
8-3-5 TET Monitor.....	26
8-4 App Designer	27
8-4-1 起動.....	27
8-4-2 GUIの配置	29
8-4-3 Target computerとの接続	30
8-4-4 モデルのビルド	32

8-4-5	リアルタイムアプリケーションのロード	34
8-4-6	リアルタイムアプリケーションの実行	35
8-4-7	信号の表示	36
8-4-8	パラメータの設定	38

1 概要

本書は、Speedgoat 社のリアルタイムターゲットマシンでサンプルモデルを実行するまでの手順について説明します。

2 構成



Host computer	Target computer で実行するリアルタイムアプリケーションを作成します。リアルタイムアプリケーションは Simulink モデルから生成されます。
Target computer	Speedgoat 製品のリアルタイムターゲットマシンです。
Target screen	Target computer の BIOS 設定や、OS のブート画面、リアルタイムアプリケーションの実行状況を表示します。

2-1 Host computer と Target computer の接続

付属の Ethernet ケーブルで接続します。

Host computer 側は Ethernet 通信が可能な RJ-45 コネクタを使用します。

Target computer 側は host link と記載された RJ-45 コネクタを使用します。

Mobile Target Machine を参考例として示します。



2-2 Target computer と Target screen の接続

使用する Target computer の種類によって使用できるケーブルが異なります。

Target computer の解像度については、それぞれの User Manual をご参照ください。

Performance	HDMI, DVI-I, VGA
Mobile	DVI-I, DisplayPort
Baseline	DisplayPort
Unit	DisplayPort

3 納品物の確認

Speedgoat 製品はターゲットマシンやケーブル、端子台といったハードウェアとライブラリやドキュメントといったソフトウェアで構成されます。

ハードウェアについては弊社からご注文いただいた内容の Speedgoat 製品を納品します。

納品物には納品リストが含まれていますので、内訳はそちらをご確認ください。

ソフトウェアについては Speedgoat 社の HP にあるカスタマーポータルサイトよりダウンロードして入手します。

カスタマーポータルサイトについては後述します。

4 カスタマーポータルサイト

カスタマーポータルサイトでは Speedgoat I/O Blockset やサンプルモデルといったソフトウェア、それらの取扱説明書のダウンロードや購入した製品ハードウェアの保守期間やソフトウェアメンテナンス期間を閲覧できるサービスを提供します。

<https://www.speedgoat.com/extranet#/Login>

4-1 アカウントの作成

カスタマーポータルサイトを利用するためには Speedgoat 社へアカウント作成の依頼をする必要があります。

ただし、注文いただいたユーザのアカウントは納品前後に作成されております。

それ以外のユーザのアカウントを作成する場合は、弊社のサポートや営業、もしくはカスタマーポータルサイトで依頼することになります。

4-2 アカウントの有効化

アカウント作成が完了後、それを有効化することでカスタマーポータルサイトへログインすることが出来ます。

有効化するためには下記 URL へアクセスしてください。

<https://www.speedgoat.com/extranet#/Login/ActivateAccount>

アクセスすると下図のような画面が表示されますので、アカウント作成時に連絡しているメールアドレスを入力します。

Activate your Account

Enter your email address and click Submit. We will send you an email that includes a link to create your password.

Email Address *

Submit

メールアドレスを入力すると、そのメールアドレス宛に下記内容のメールが届きます。

Please create your password by clicking here: [Activate your Account](#)

Please note that this token/link is only valid for 30 minutes. If your token has expired, please request a new one on www.speedgoat.ch/login

Your Speedgoat Team

メールに記載されたリンク「Activate your Account」を選択すると、パスワードを設定する画面が表示されます。

Set your Password

new Password *

Passwords must be at least 8 characters long including letter, numerical digit and a special character

Confirm Password *

Submit

パスワードの設定が完了後、メールの「www.speedgoat.ch/login」を選択して下図のログイン画面を開きます。

Customer Log In

The customer area gives access to:

- Details of your purchased systems
- Renewal of hardware warranty and software maintenance plans
- Download the latest drivers, software and documentation

Email Address *

Password *

Login

[Forgot your password?](#)

Don't have a Speedgoat Account? [Create an account](#)

ログイン画面でメールアドレスとパスワードを入力すると、カスタマーポータルへログインできます。

4-3 Account Home

ログイン後に表示される Account Home 画面が下図になります。

 <p>Maintenance View and renew your maintenance</p>	 <p>Downloads Download drivers, software and documentation</p>
 <p>Learning Tutorials Introduction to Simulink Real-Time</p>	 <p>Documentation Documentation for the Speedgoat Library for Simulink Real-Time and HDL Coder</p>
 <p>Your Company Profile Edit your company's address</p>	 <p>My Profile Edit your name and change password</p>
 <p>Users Add, view and edit the accounts of your company's users</p>	 <p>Log out Log out of the customer area</p>

Maintenance	ハードウェアの保守期間やソフトウェアメンテナンス期間の確認ができます。
Downloads	ライブラリやサンプルモデル、ドキュメントのダウンロードができます。
Learning Tutorials	基本的な操作方法を紹介した動画が閲覧できます。
Documentation	Speedgoat I/O Blockset などの説明が記載されています。
Your Company Profile	アカウントユーザが所属する組織情報が記載されています。
My Profile	アカウントユーザの情報が記載されています。
Users	組織に所属しているユーザー一覧が記載されています。
Log out	ポータルサイトからログアウトします。

4-3-1 Maintenance

Account HomeでMaintenanceを選択すると表示されます。
購入した製品の情報が表示されます。

Maintenance

Mobile real-time target machine - serial number: 3035 >

Performance real-time target machine - serial number: 3036 >

Baseline real-time target machine - S - serial number: 3612 >

Performance real-time target machine - serial number: 4343 ▾

Main user: Souta Kawashima

Description	Item ID	Hardware Warranty expiry date	Software Maintenance expiry date
System	109200 109006 109211	01/Apr/2020	01/Apr/2028
IO322	2A322	01/Apr/2020	01/Apr/2028
IO75X	2A75X	01/Aug/2022	01/Aug/2021

Baseline real-time target machine - M - serial number: 4494 >

Components >

Maintenance Renewal

Request a quote for the renewal of all expired, or soon to expire, target machine and components. Maintaining your subscription ensures that your systems are fully compatible with future MathWorks software releases and enables access to technical support.

Please choose a Renewal Type *

1 Year Renewal

2 Years Renewal

3 Years Renewal

Renewal until end of 2022

Custom (specify below)

Please add any special instructions here

[Request Maintenance Quote](#)

Hardware Warranty expiry date ハードウェアの保守期間です。

Software Maintenance expiry date ソフトウェアのメンテナンス期間です。

4-3-2 Downloads

Account HomeでDownloadsを選択すると表示されます。

Downloads

Speedgoat I/O Blockset

Installation instructions:

1. Download the library for your MATLAB release
2. Extract the downloaded ZIP file into a temporary folder
3. In MATLAB, navigate to the extracted folder which contains speedgoat_setup.p (do not add any folder to the MATLAB path)
4. Right click on the speedgoat_setup.p and select Run
5. Follow the instructions
6. Restart MATLAB

[Important Note: Migration of existing hardware to new R2020b release](#)

[Speedgoat I/O Blockset for R2018b \(v9.2.0\)](#)

[Speedgoat I/O Blockset for R2019a \(v9.2.0\)](#)

[Speedgoat I/O Blockset for R2019b \(v9.2.0\)](#)

[Speedgoat I/O Blockset for R2020a \(v9.2.0\)](#)

[Speedgoat I/O Blockset for R2020b \(v9.2.0\)](#)

[Speedgoat I/O Blockset for R2021a \(v9.2.0\)](#)

[Speedgoat I/O Blockset Release Notes](#)

主要なソフトウェアとドキュメントについて下表に記載します。

Speedgoat I/O Blockset	Speedgoat の I/O ライブラリです。
Configuration and Testing Information	サンプルモデルの説明書です。
Simulink Test Model	サンプルモデルです。
Custom Implementation	FPGA モジュールの bitstream です。
HDL Coder Integration Package	HDL Coder 用 Speedgoat のライブラリです。

※ソフトウェアやドキュメントの名称は、通知なく変更される場合がございますのでご了承ください。

※ソフトウェアの中にはオプション品も含まれ、購入していないと表示されないものがあります。

ソフトウェアメンテナンス期間内の間は最新の Speedgoat I/O Blockset と HDL Coder Integration Package を提供しております。

5 開発環境

5-1 用意するソフトウェア

5-1-1 MathWorks 製品

Simulinkモデルでアプリケーションを作成する開発担当者とそのアプリケーションを使用する試験担当者のそれぞれに必要なMathWorks製品ソフトウェアを下表に示します。

MathWorks製品	開発担当	試験担当
MATLAB	✓	✓
Simulink	✓	✗
MATLAB Coder	✓	✗
Simulink Coder	✓	✗
Simulink Real-time	✓	✓
HDL Coder	✓*	✗
Fixed Point Designer	✓*	✗

*HDL Coder Integration Package使用時のみ必要

これらの製品は MathWorks 社からインストーラをダウンロードし、インストールします。
またMATLABのAdd-onソフトウェアをインストールする必要がありますが、こちらは無償で提供されます。

5-1-2 Speedgoat 製品

Simulinkモデルでアプリケーションを作成する場合は下記ふたつのソフトウェアをカスタマーポータルサイトよりダウンロードします。

どちらもMATLABのバージョンごとに用意されております。

- Speedgoat I/O Blockset
- Speedgoat HDL Coder Integration Package (HDL Coder使用時のみ必要)

※Speedgoat I/O Blocksetは最新Verから数えて6リリース前のMALTABまでをサポートします。

5-1-3 Xilinx 製品

HDL Coderを使用する場合はXilinxの統合開発環境Vivadoが必要になります。

詳細は下記URLをご参照ください。

https://www.speedgoat.com/help/hdlcoder/page/refentry_workflow

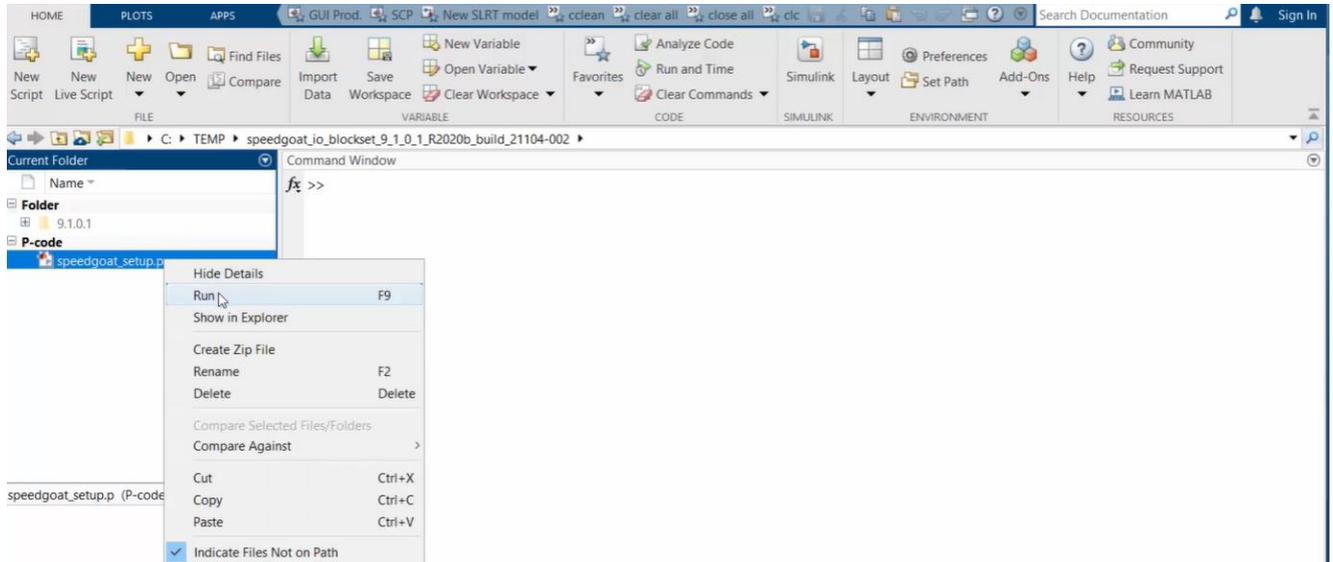
5-2 Speedgoat 製品のインストール

5-2-1 Speedgoat I/O Blockset

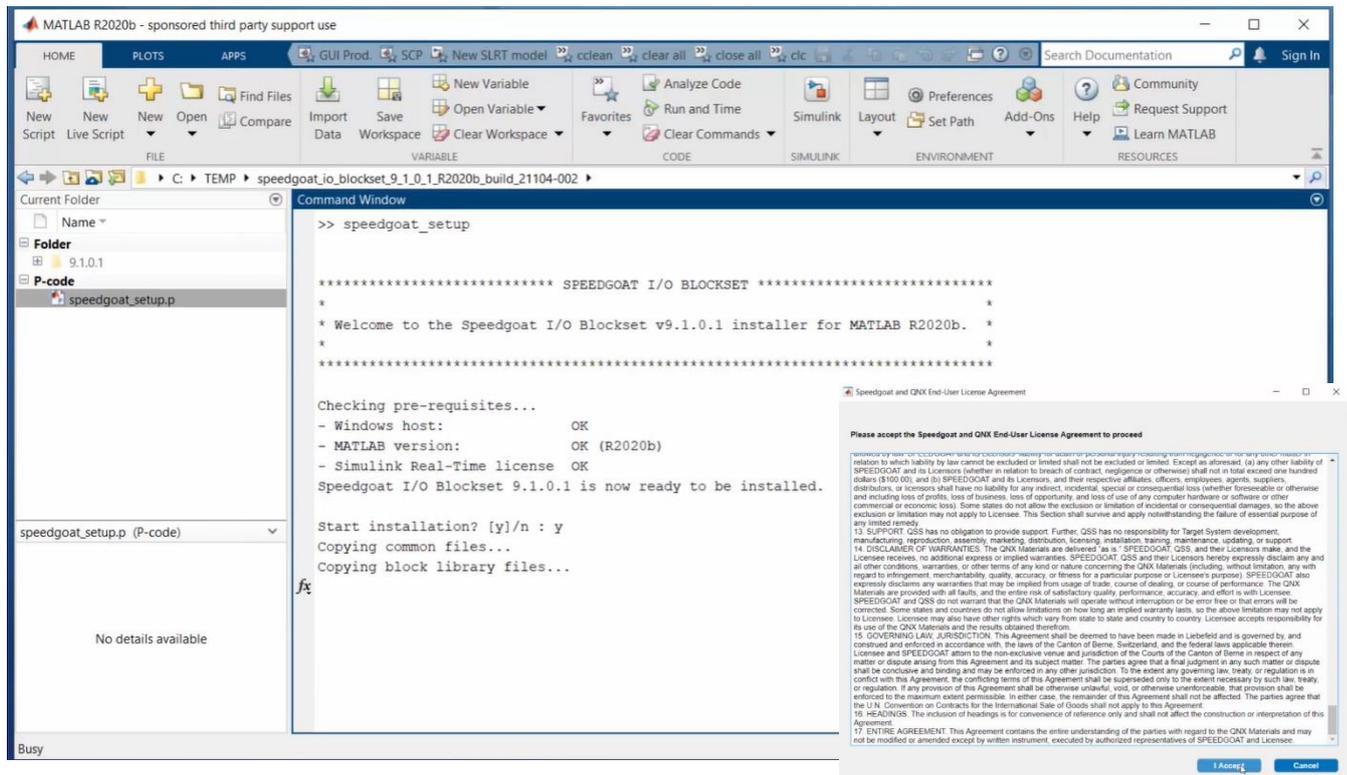
カスタマーポータルからダウンロードした Speedgoat I/O Blockset は zip ファイルとなります。

それを任意のフォルダに解凍し、そのフォルダを MATLAB の Current Folder にします。

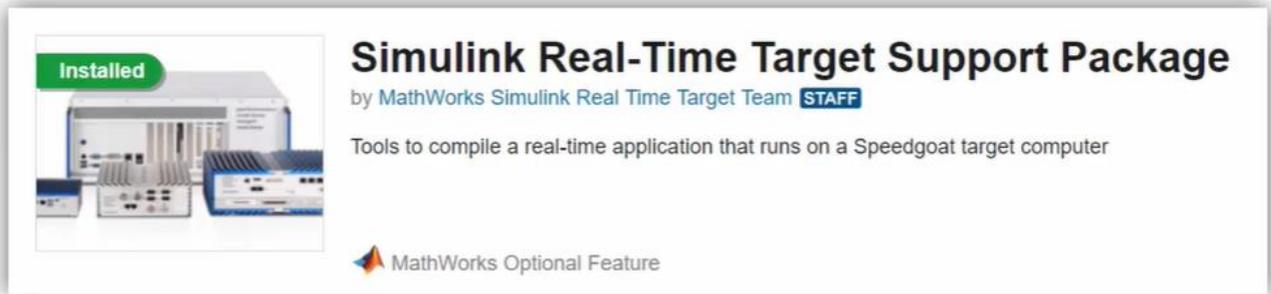
解凍したフォルダに格納されている speedgoat_setup.p を実行します。



インストールが開始されますので、表示される使用許諾契約をよく読み、許諾を選択します。



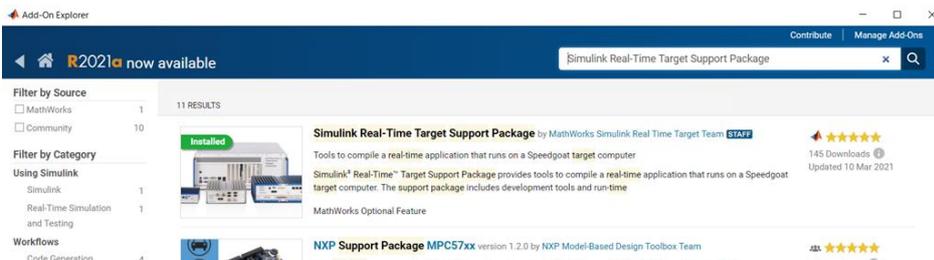
途中で Simulink Real-time Target Support Package をインストールする必要があると表示されるので、それも合わせてインストールします。



問題なければ、そのまま Speedgoat インストールは完了します。

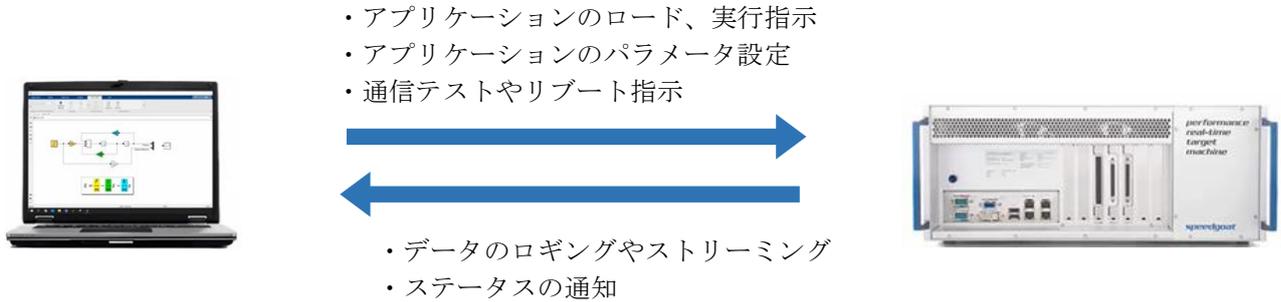
MATLAB の Command Window に `speedgoat.version` と入力すると、現在インストールされているバージョンを確認できます。

※Simulink Real-time Target Support Package は Add-On Explorer からインストールします。
検索窓に「Simulink Real-time Target Support Package」と入力すると、簡単に表示されます。



6 ネットワーク設定

Host computer と Target computer で Ethernet 通信するためにネットワークの設定をする必要があります。



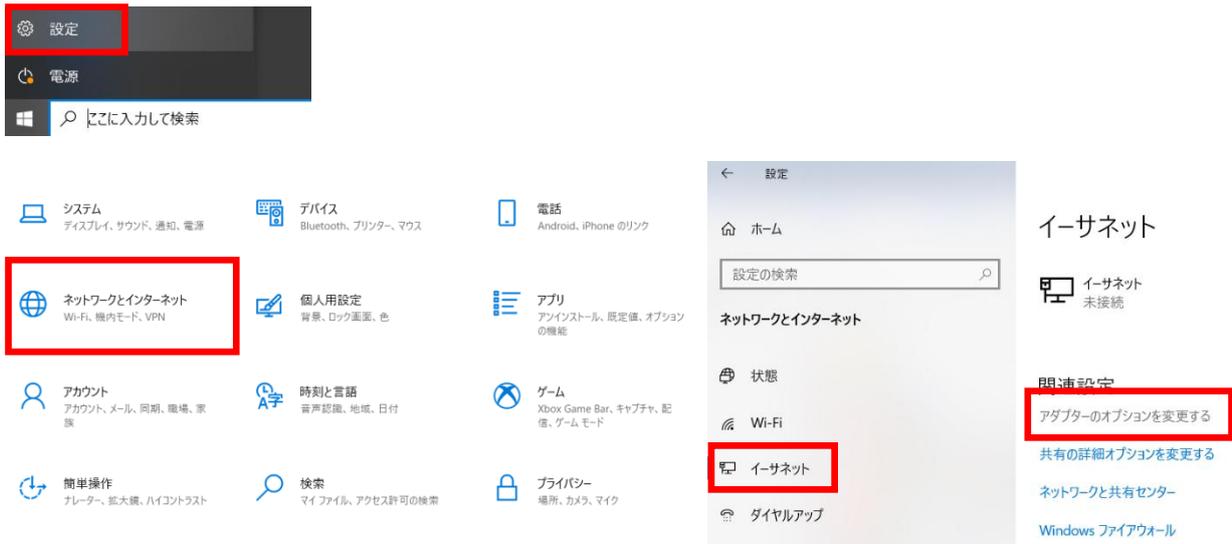
Target computer の出荷時のネットワーク設定は以下のとおりです。

IP Address	192.168.7.1
Subnet mask	255.255.255.0

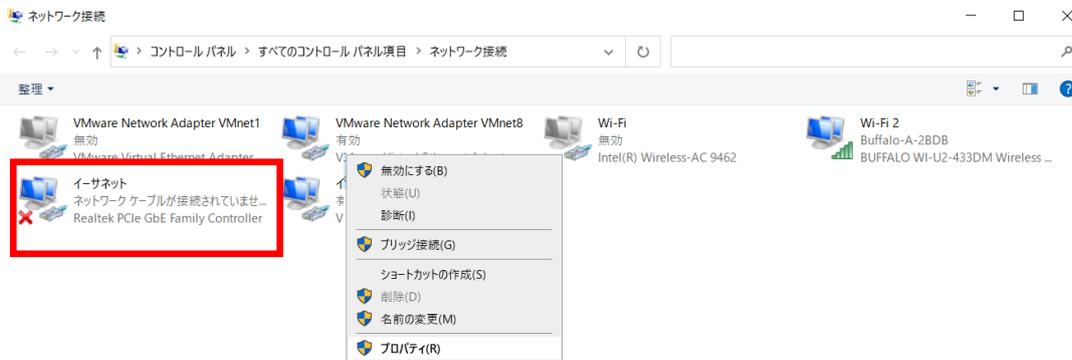
6-1 Host computer

下記の項目から Host computer の設定を変更することができます。

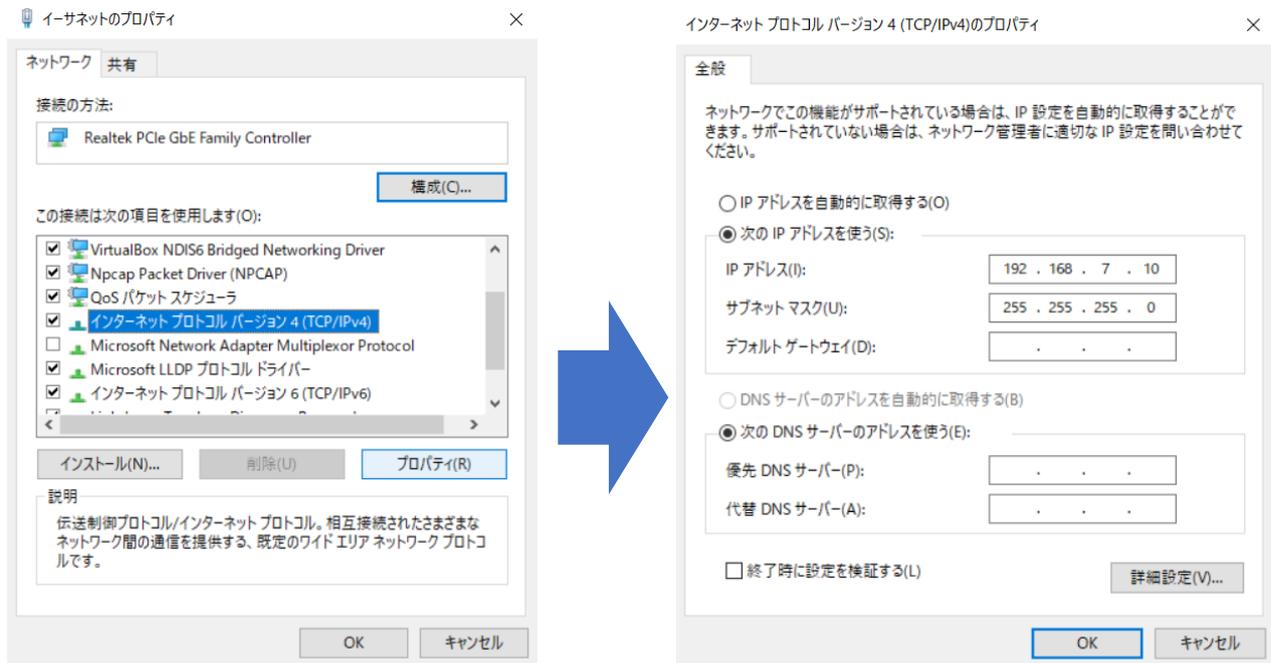
PC の設定 > ネットワークとインターネット > イーサネット > アダプターのオプションを変更する



使用するネットワークカード（ネットワークアダプタ）を右クリックし、プロパティを選択します。下図の例では最も左下の項目が使用するネットワークカードとなります。



インターネットプロトコルバージョン 4(TCP/IPv4)をクリックした後、プロパティをクリックします。
 下図はデフォルト設定の Target computer と接続できる設定となります。

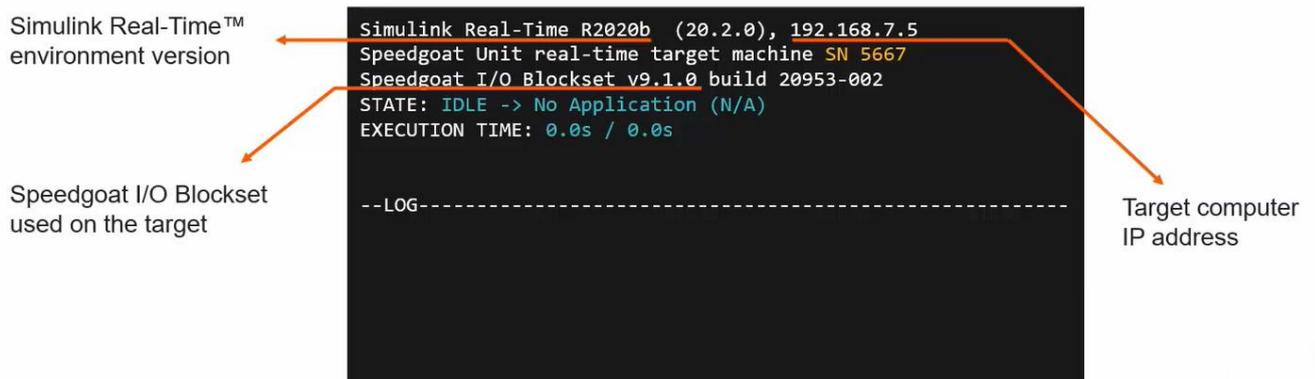


IP Address	192.168.7.10
Subnet mask	255.255.255.0

6-2 Target computer

Target computer のネットワーク設定は、Simulink Real-time Explorer で変更することができます。
 ただし、Host computer と Ethernet 通信するためには Host computer で使用する各ソフトウェアのバージョンと Target computer のそれらのバージョンを合わせる必要があります。

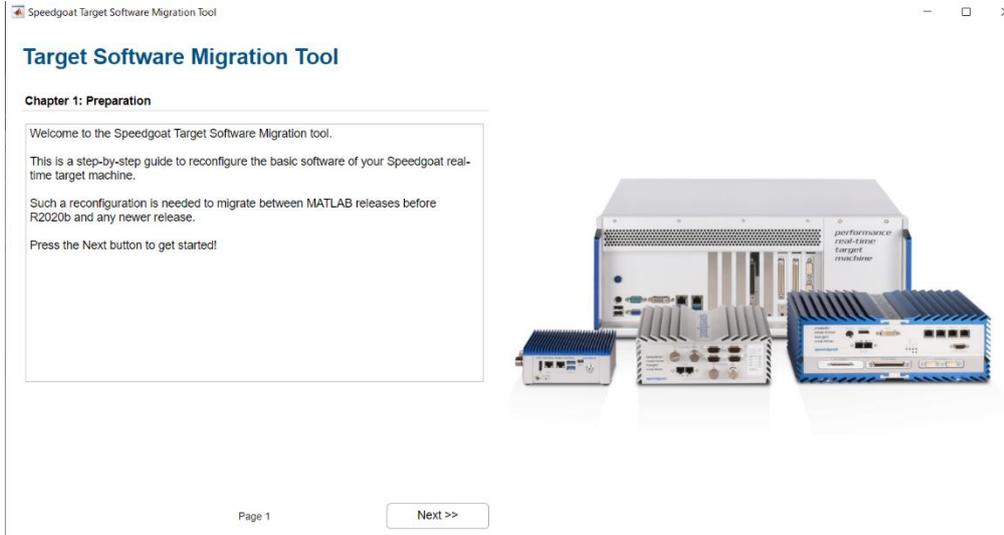
Target computer を起動し、ブートするとバージョン情報が表示されます。



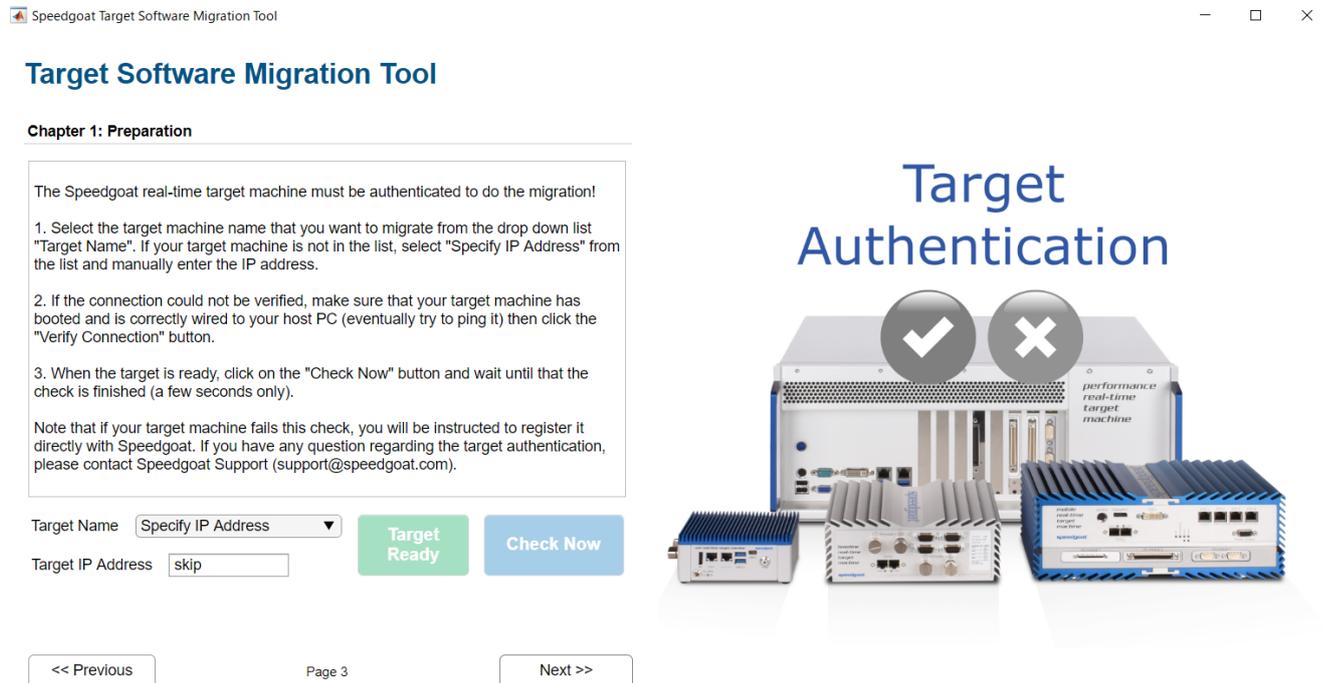
※上記画面が表示されず、Target computer のアップデートを行う説明を表示する場合があります。
 この場合でも後述する Kernel の更新を実施することで、上記画面が表示されるようになります。

6-2-1 Kernel の書き換え

MATLAB の Command Window 上で `speedgoat.migrateTarget` と入力します。
 入力すると、下図のような画面が表示されます。
 画面に従って、Kernel の書き換えを行います。



手順を進めていくと、Host computer と Target computer との接続を確認する場面があります。
 接続がうまくできない場合は、下図のように設定することで手順を進めることができます。



※*Target Name* を *Specify IP Address* とし、*Target IP Address* へ *skip* と入力し、*Next* ボタンを押します。

Kernel の書き込み後、Host computer から Kernel の更新を行います。その方法について「7-2 Target computer との接続」を参照してください。

7 起動

7-1 Target computer の起動

Target computer の電源を起動します。

Target computer は電源を起動すると、BIOS > OS の順でブートします。

OS の Boot に成功すると、Target screen 上にネットワークなどの各種情報が表示されます。

Simulink Real-Time™
environment version

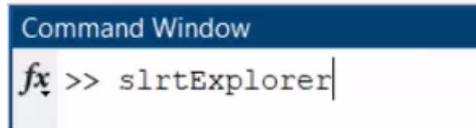
Speedgoat I/O Blockset
used on the target

```
Simulink Real-Time R2020b (20.2.0), 192.168.7.5  
Speedgoat Unit real-time target machine SN 5667  
Speedgoat I/O Blockset v9.1.0 build 20953-002  
STATE: IDLE -> No Application (N/A)  
EXECUTION TIME: 0.0s / 0.0s
```

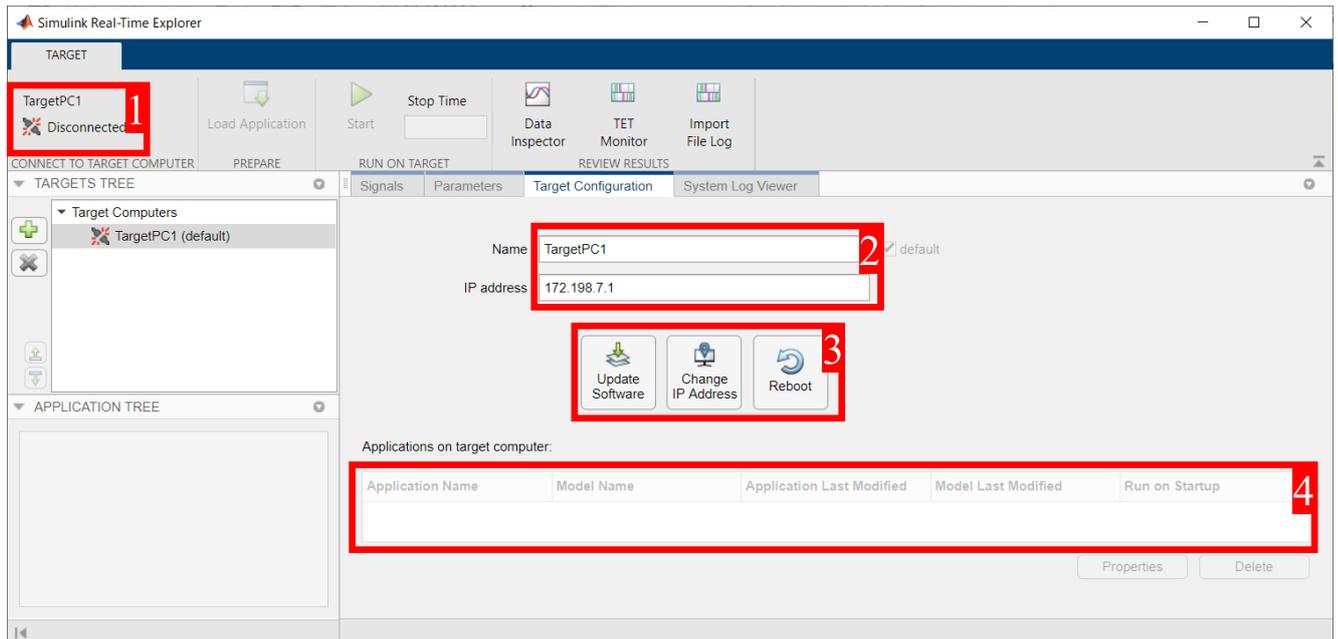
Target computer
IP address

7-2 Target computer との接続

Host computer から Target computer へ接続確認を行います。
 接続確認は MATLAB の Simulink Real-time Explorer から行います。
 MATLAB の Command Window で slrtExplorer と入力します。



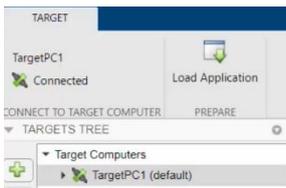
• Simulink Real-time Explorer の画面



- 1 接続ボタンです。このボタンを押すと、2 で指定した IP アドレスに対して接続を試みます。接続が成功すると、ボタンが「connected」と表示されます。
- 2 Target computer の名称と IP アドレスです。
- 3 Update Software ボタンは Target computer の Kernel を更新します。
Change IP Address ボタンは Target computer の IP アドレスを 2 で指定したものに変わります。
Reboot ボタンは Target computer をリブートします。
- 4 Target computer にロードされたアプリケーションが表示されます。ブート時に実行されるアプリケーションも指定することができます。

Application Name	Model Name	Application Last Modified	Model Last Modified	Run on Startup
ref_main_model	ref_main_model	2020-10-19 17:00:34	2020-10-19 09:13:23	<input type="checkbox"/>
slrt_ex_osc	slrt_ex_osc	2020-10-26 11:12:38	2020-09-29 11:36:23	<input checked="" type="checkbox"/>

接続ボタンを押して、Target computer との接続が成功すると、接続ボタンが「connected」となります。



7-3 Simulink モデルのビルド

カスタマーポータルサイトからダウンロードしたサンプルモデルをビルドし、リアルタイムアプリケーションを生成します。

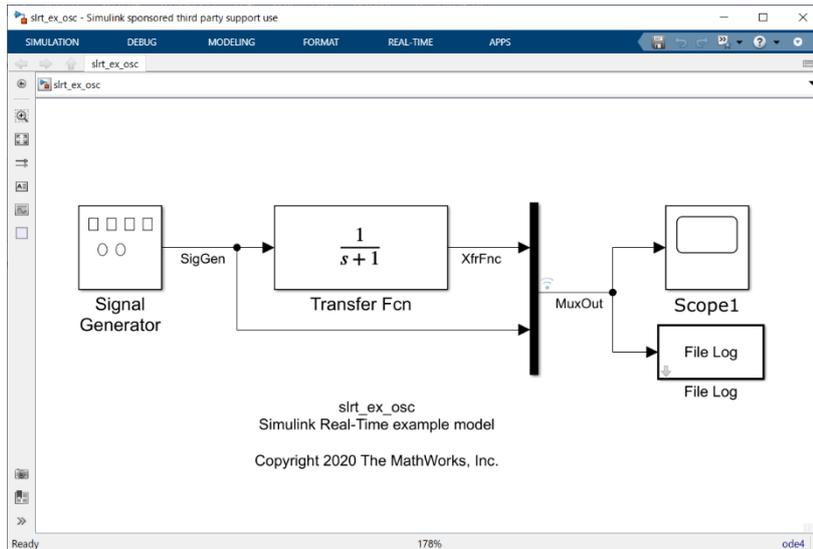
本項では Simulink I/O Blockset を使用していないオシレータモデルをビルドする方法を説明します。

7-3-1 オシレータモデル

MATLAB の Command Window で `slrt_ex_osc` と入力します。

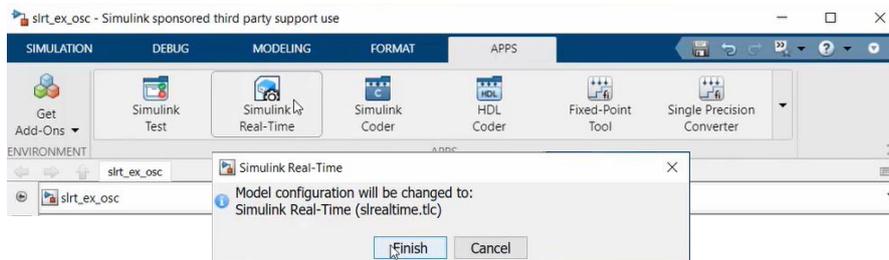
入力すると Simulink 上でオシレータモデル（モデル名：`slrt_ex_osc`）が開かれます。

この `slrt_ex_osc` を別名保存して、使用してください。

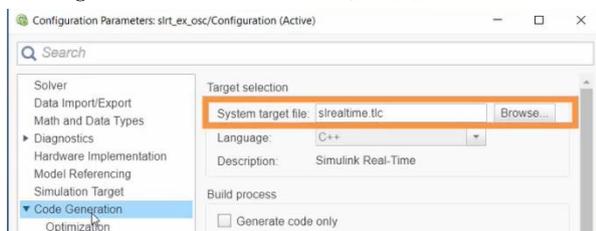


Simulation > Run をクリックすると Host computer 上でオシレータモデルをシミュレーション実行します。

次に APPS > Simulink Real-time をクリックすると、モデルが Simulink Real-time 用のテンプレートファイルを使用する設定になります。



Configuration Parameters で変更されたテンプレートファイルを確認することができます。



※ただし `slrt_ex_osc` の初期設定で Simulink Real-time 用の設定となっております。

Configuraion Parameters の Stop time を inf にした後ビルドすることで、Host computer から停止を指令するまでリアルタイムアプリケーションを実行します。この設定を行った前提で、以降の説明を行います。



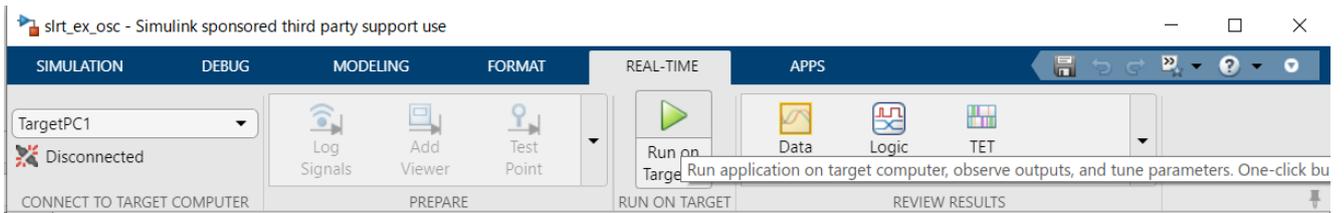
※Configuraion Parameters は ボタンを押すことで表示されます。

この画面ではモデルの動作周期なども変更することができます。

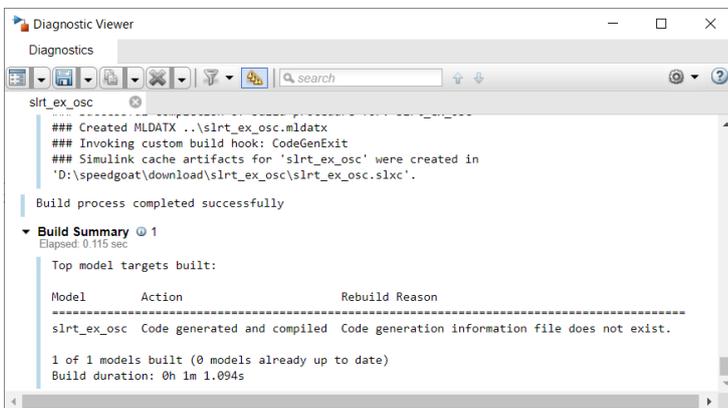


Run on Target をクリックすると、Simulink モデルをビルドし、Target computer で実行することができるリアルタイムアプリケーションを生成します。

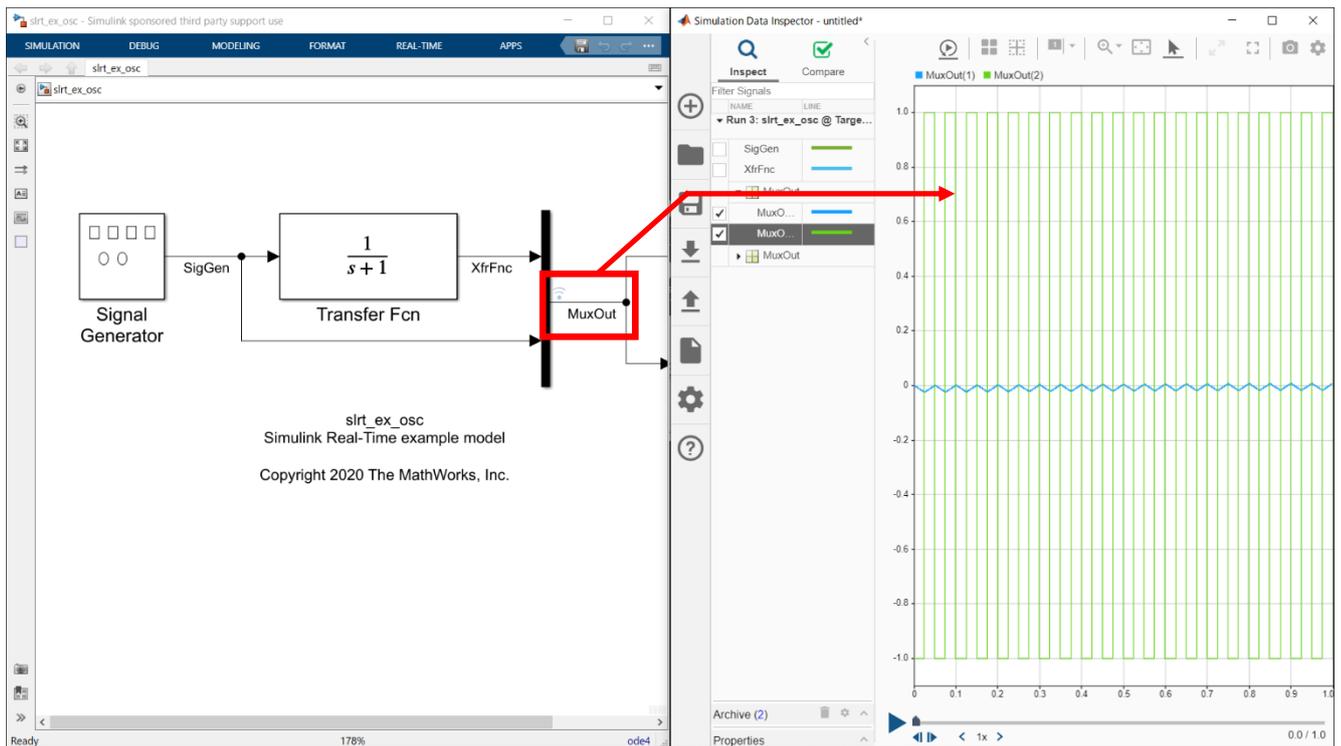
生成後、Target computer へそのリアルタイムアプリケーションをロードし、実行するまでの一連の動作を自動で行います。



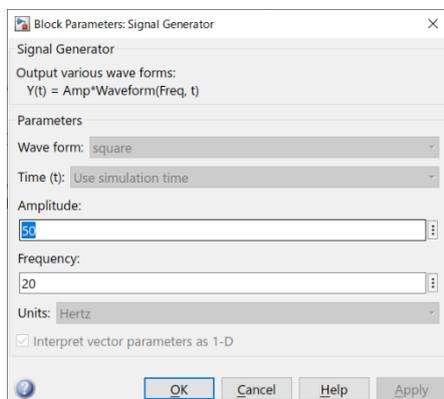
下図はビルド結果を表示する診断ビューア (Diagnostic Viewer) です。



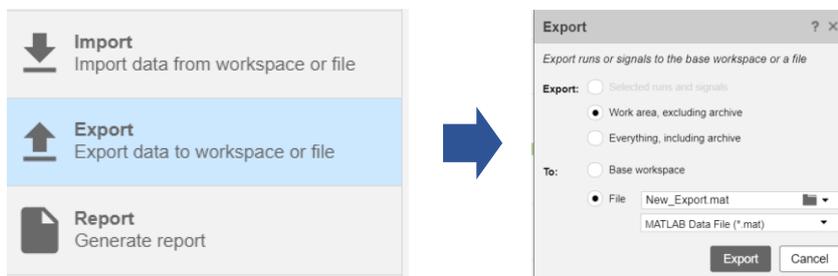
Data Inspector をクリックすると、Simulation Data Inspector が起動し、ロギング設定している信号線のデータを表示します。



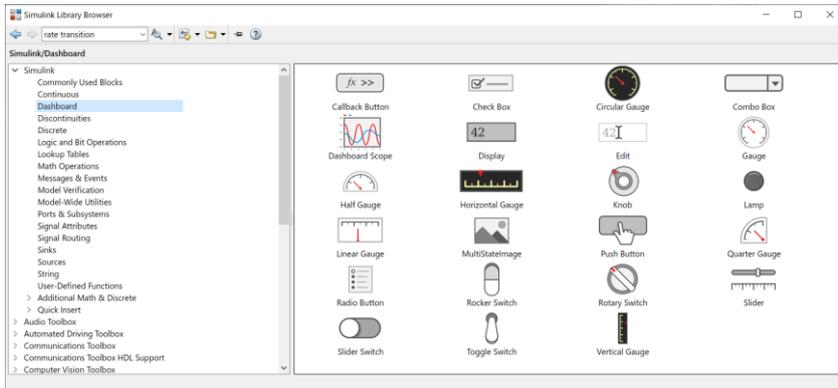
リアルタイムアプリケーションを実行中に、ブロックのパラメータを変更することで、Target computer 上のパラメータも変更されます。これにより、ロギング中のデータには変化が生じます。



Simulation Data Inspector ではロギングしたデータをファイル保存する機能もありますので、必要に応じて使用してください。



Simulink の Dashboard を使用することでも、信号やパラメータの表示や変更が可能ですので、必要に応じて使用してください。



7-3-2 サンプルモデル

Speedgoat の Target computer (Real-time target machine) を購入いただいたユーザのポータルサイトではサンプルモデルとその説明書 (Configuration and Testing Information) がダウンロードできます。

System

[Configuration and Testing Information](#)
[Simulink Test Model](#)

※ユーザごとに説明書の名称 (例: *Technical Reference Manual*) が異なっておりますのでご注意ください。

サンプルモデルを実行するためにはダウンロードした説明書に記載されたとおりに準備をします。

基本的には以下のような流れになります。

- サンプルモデルを任意のフォルダに格納し、MATLAB 上でそのフォルダを Current Folder にした後、サンプルモデル (.slx) を開きます。
- Target computer と Terminal board (端子台) をケーブルで接続します。
Terminal board とケーブルは Target computer に同梱しております。
- オシレータモデルと同様にビルド、ロード、実行を行います。
- Data Inspector 上で正常に信号データが表示されることを確認します。

8 Control and Instrumentation

Target computer で実行しているモデル（リアルタイムアプリケーション）のブロックパラメータを設定したり、信号のデータを表示したりする方法は複数あります。

- MATLAB
- Simulink
- Simulink Real-time Explorer
- App Designer

本章では、これらの使用方法について説明します。

8-1 MATLAB

MATLAB ではコマンドウィンドウを使用して Target computer のパラメータや信号データへアクセスすることができます。

コマンドウィンドウ上で下記のように入力すると、Target computer の情報が tg に格納され、以降 tg を使用して Target computer を制御することができます。

```
>>tg=slrealtime
```

8-1-1 起動

下記コマンド群でモデルのビルド、Target computer にアプリケーションをロードすることや、開始・停止することを指示することができます。

```
>>rtwbuild('ModelName')
>>tg.load('ModelName')
>>tg.start
>>tg.stop
```

詳しくは Simulink Real-time のヘルプを参照してください。

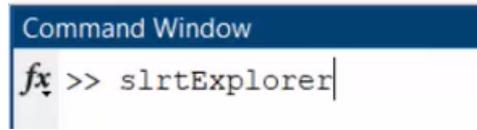
8-2 Simulink

Simulink での操作方法は前章で説明しましたので、そちらをご参照ください。
これが最も簡単な操作方法となります。

8-3 Simulink Real-time Explorer

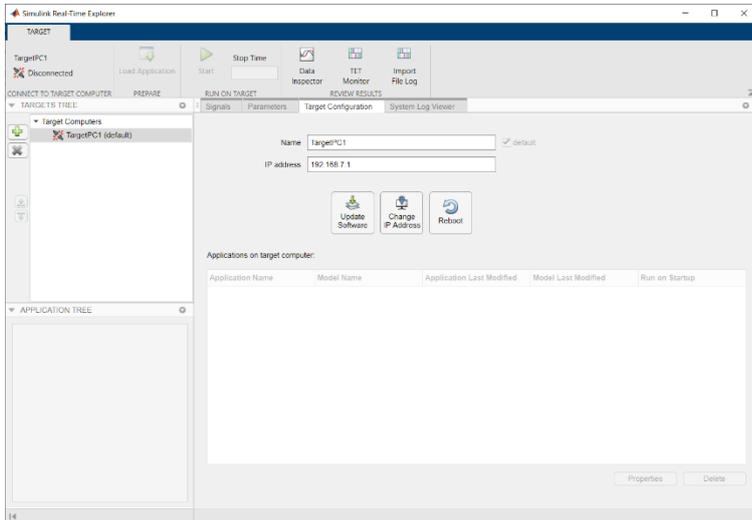
Simulink Real-time Explorer での操作方法を説明します。

MATLAB のコマンドウィンドウで下記コマンドを実行することで Simulink Real-time Explorer が起動します。



```
fx >> slrtExplorer
```

起動後の画面を下図に示します。

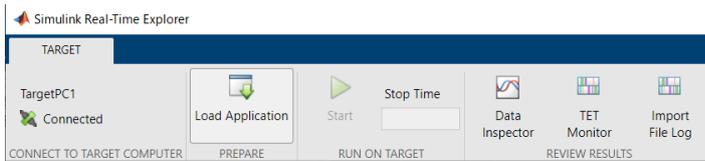


Target Computer との接続までは 7 章で説明しておりますので、そちらをご覧ください。

本項ではリアルタイムアプリケーションのロードから、パラメータの設定や信号の確認方法について説明します。

8-3-1 リアルタイムアプリケーションのロード

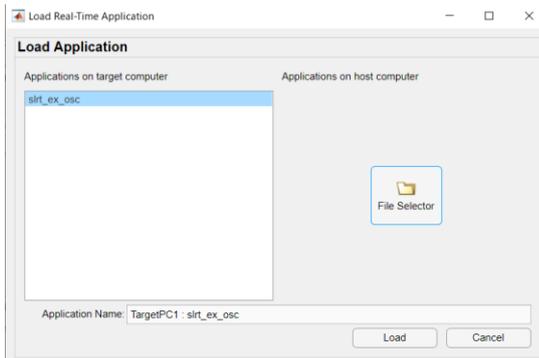
Load Application ボタンを押すことで、リアルタイムアプリケーションをロードすることができます。



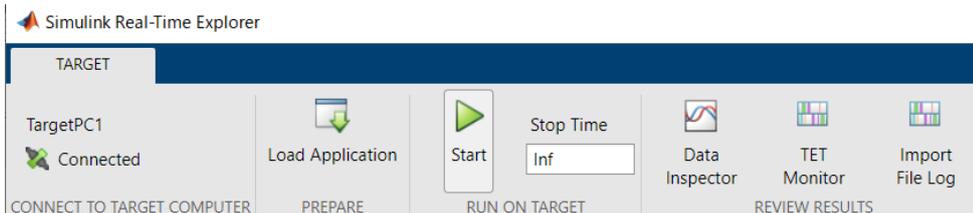
ボタンを押すと、下図のような画面が表示されます。

Application on target computer のリストには以前に Target computer へロードしたリアルタイムアプリケーションが表示されます。Application on host computer には File Selector ボタンがあり、PC 上のリアルタイムアプリケーションを指定することができます。

ロードしたいリアルタイムアプリケーションを選択し、Load ボタンを押します。



ロードが成功した後、Start ボタンを押すことで Target computer でリアルタイムアプリケーションが実行されます。



8-3-2 信号の表示

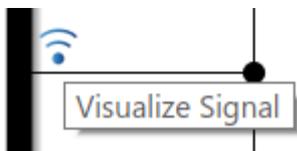
Simulink Real-time Explorer 上のタブ Signal を選択すると、下図のような画面が表示されます。

The screenshot shows the Simulink Real-time Explorer interface. The 'Signals' tab is active, displaying two tables of signal information.

Signals available on target computer	
Block Path	Signal Name
slrt_ex_osc/TmpSignal ConversionATAQSigL...	MuxOut
slrt_ex_osc/Signal Generator:1	SigGen
slrt_ex_osc/Transfer Fcn:1	XtrFnc

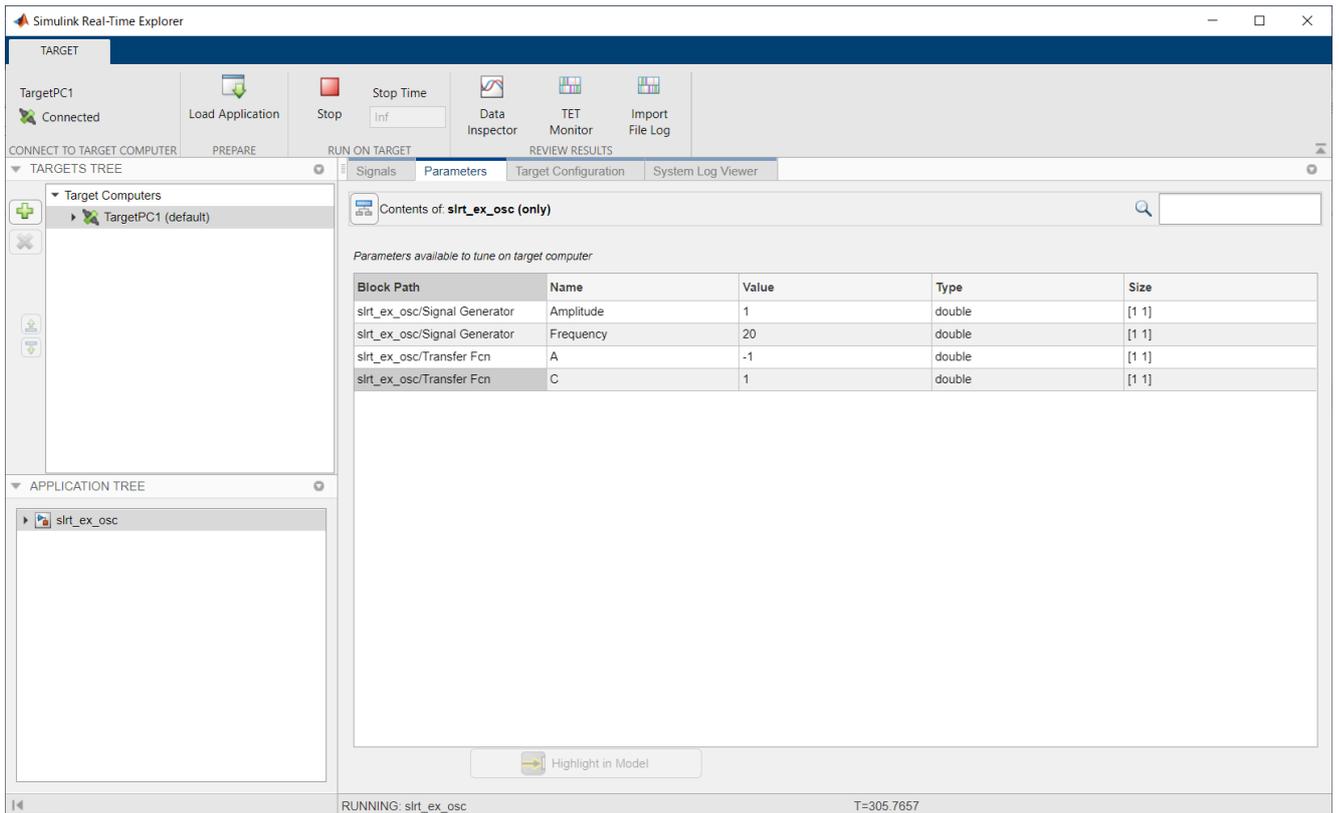
Signals in Instrument	
Block Path	Value
slrt_ex_osc/TmpSignal ConversionATAQSigLogging_Insert...	[-0.010182 1]

Simulink モデル上で下図のマークをつけた信号線の値を確認することができます。



8-3-3 パラメータ設定

Simulink Real-time Explorer 上のタブ Parameters を選択すると、下図のような画面が表示されます。



Simulink モデル上に配置した各ブロックのパラメータを表示します。

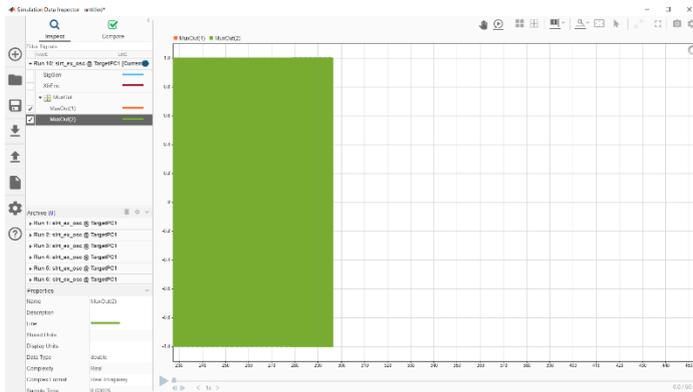
この画面の Value の値を変更することで、Target computer 上のパラメータも変更されます。

Parameters available to tune on target computer

Block Path	Name	Value	Type	Size
slrt_ex_osc/Signal Generator	Amplitude	50	double	[1 1]
slrt_ex_osc/Signal Generator	Frequency	20	double	[1 1]
slrt_ex_osc/Transfer Fcn	A	-1	double	[1 1]
slrt_ex_osc/Transfer Fcn	C	1	double	[1 1]

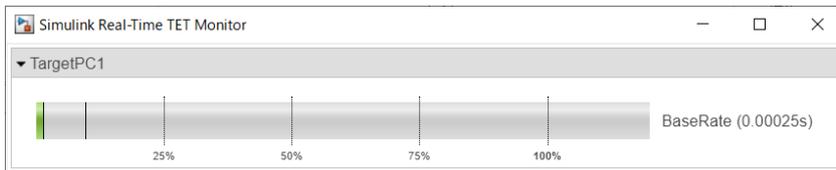
8-3-4 Data Inspector

Simulink Real-time Explorer 上の Data Inspector ボタンを押すと、Simulink に存在する Simulink Data Inspector が起動します。



8-3-5 TET Monitor

Simulink Real-time Explorer 上の TET Monitor ボタンを押すと、下図の画面が表示されます。この画面ではリアルタイムアプリケーションの周期処理時間が表示されます。



8-4 App Designer

App Designer では GUI の作成と、その GUI で Target computer を制御することができます。本章では App Designer で作成した GUI アプリケーションを使用して、Target computer へモデル `slrt_ex_osc` をロード、実行する手順について説明します。

※本章の手順は `slrt_ex_osc.slx` が MATLAB の Current Folder に存在する状態で行ってください。

8-4-1 起動

MATLAB のコマンドウィンドウで下記コマンドを入力します。

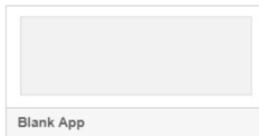
```
>>appdesigner
```

新規作成するアプリケーションの種類を選ぶことができるので、用途に合わせて選択します。今回は Blank App を選択します。

アプリケーションは `slrt_ex_osc.slx` と同じフォルダに作成します。

ファイル名は任意の名称 (`slrt_ex_osc_app.mlapp` 等) を入力します。

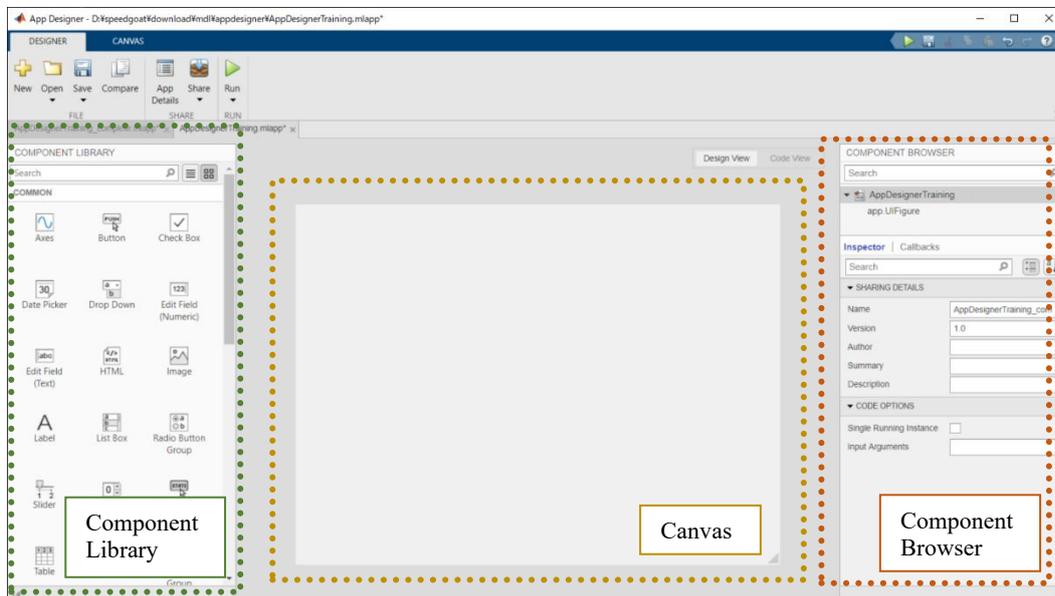
▼ New



下図が App Designer の画面となります。

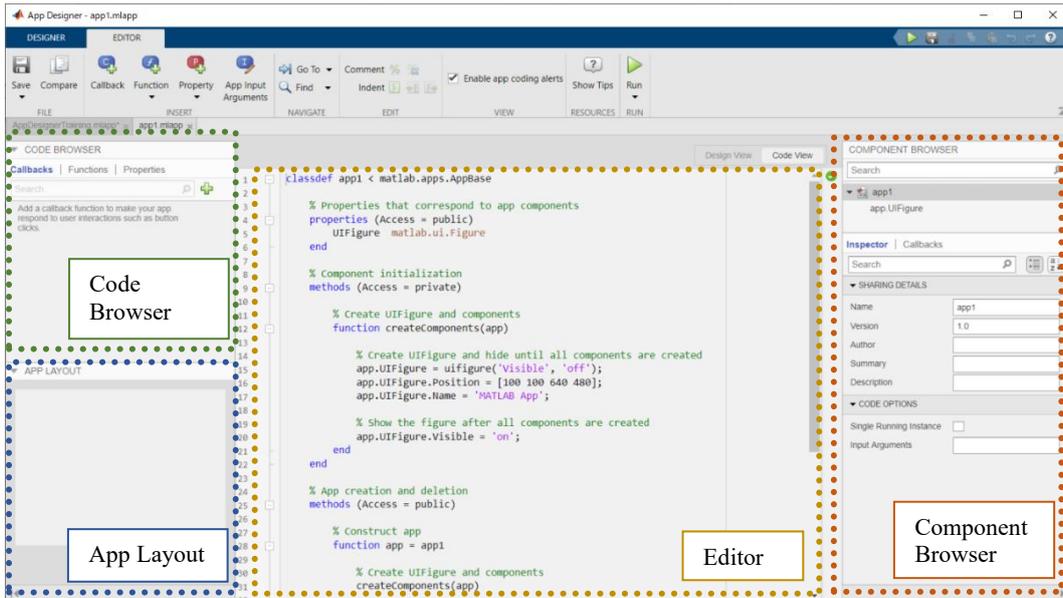
App Designer はアプリケーションの GUI の配置や設定を行う Design view とアプリケーションの挙動をコーディングする Code View という二つの画面で構成されます。

• Design view



Component Library	Canvas に配置できる GUI の部品です。
Component Browser	Canvas に配置した各 GUI の部品のプロパティやコールバックを設定します。
Canvas	GUI を配置するアプリケーションの画面です。

・ Code view



Code Browser

Editor で作成した関数などが一覧表示されます。

Component Browser

Canvas に配置した各 GUI の部品のプロパティやコールバックを設定します。

App Layout

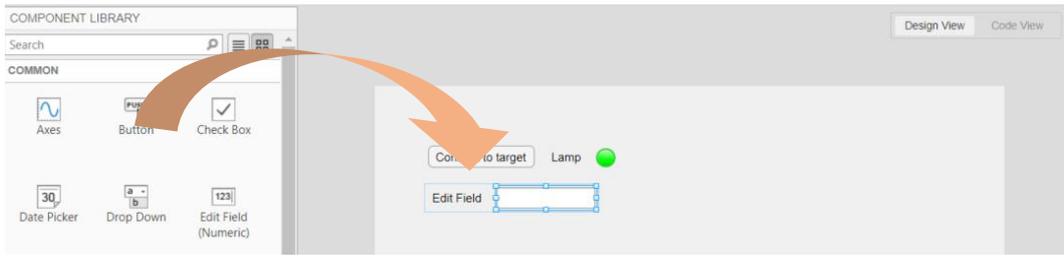
Design View の Canvas が縮小表示されます。

Editor

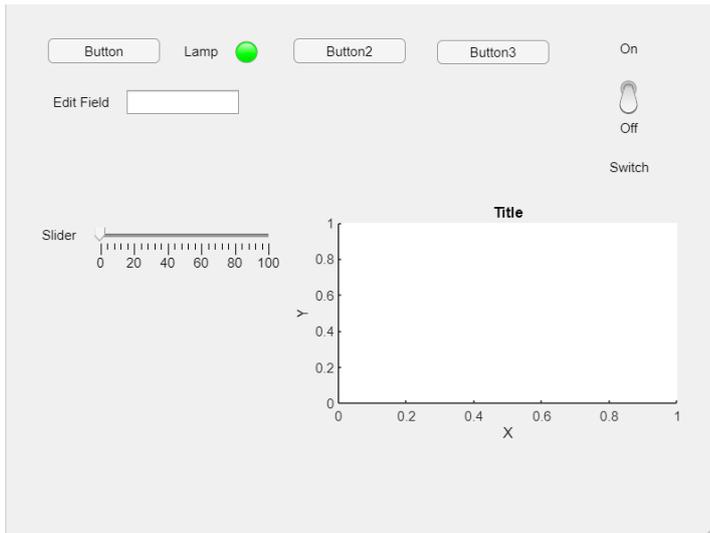
アプリケーションのコードを編集します。

8-4-2 GUIの配置

Component Library から配置したい部品を Canvas へドラッグ&ドロップします。



今回は以下のパーツを配置してください。



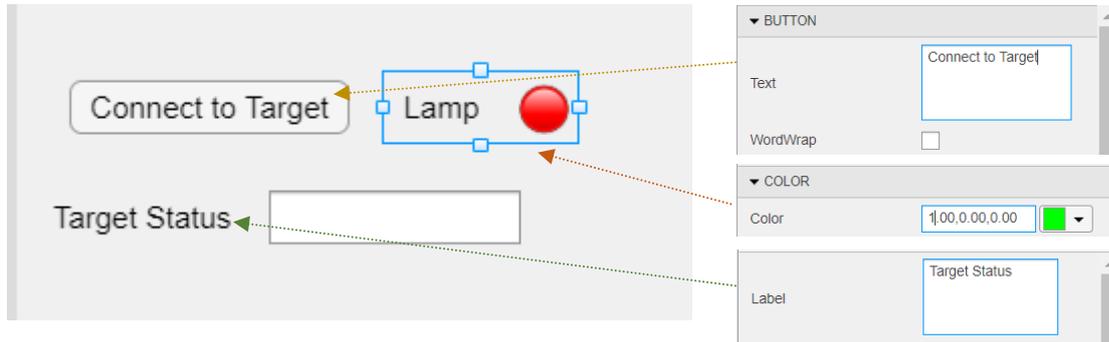
Button x3	Target computer と接続をするための Button モデル slrt_ex_osc. slx をビルドするための Button Target computer へ slrt_ex_osc のアプリケーションロードするための Button
Lamp	Target computer との接続状態を示す Lamp
Edit Field(Text)	Target computer とのあらゆる状態を示す Edit Field(Text)
Toggle Switch	Target computer へロードしたアプリケーションを実行開始/停止する Switch
Slider	Signal Generator のパラメータ Amplitude を設定するための Slider
Axes	信号 MuxOut を表示するための Axes

パーツを配置するごとに Component Browser 上でパーツに対応する変数が作成されます。

8-4-3 Target computer との接続

まずは接続に使用する Button と Edit Field の Text や Lamp の初期の色を変更します。

変更するには各パーツを選択後（アクティブにした後）、Component Browser に表示される設定を変更します。



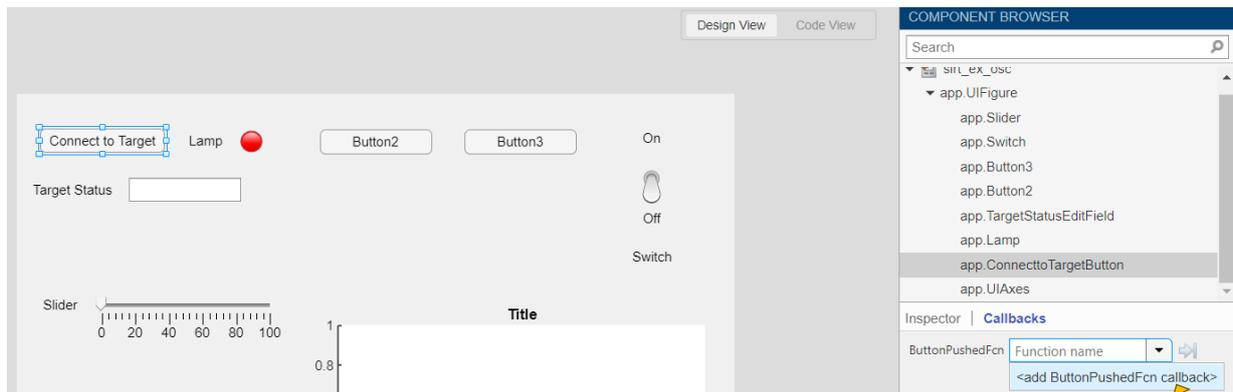
また Component Browser の設定を変更することで Code Browser 上の各パーツに対応する変数の名称も自動的に変更されます。

Connect to Target ボタンを押したときに Target computer と接続するように callback function(コールバック関数)を設定します。

Connect to Target ボタンを選択後、Component Browser の Callbacks でコールバックのイベントを選択します。

ボタンの場合は、ButtonPushedFcn というコールバックのイベントが用意されています。

これを選択すると、Code View 上でコールバック関数が作成されます。



```
% Button pushed function: ConnecttotargetButton
function ConnecttotargetButtonPushed(app, event)

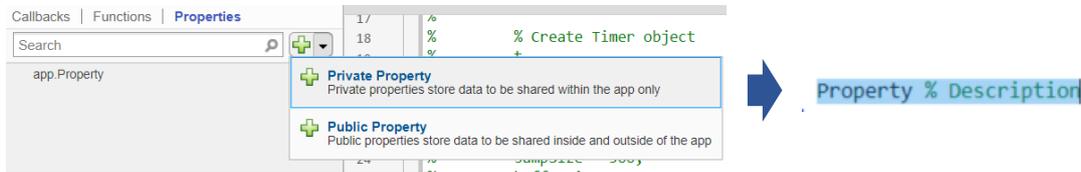
end
```

次にプライベート変数を作成します。

このプライベート変数に Target computer の情報を格納します。

Code Browser の Property > Private Property を選択します。

選択するとプライベート変数が作成されます。



分かりやすいように変数名を変更し、コメントを追加しておきます。

```
properties (Access = private)
    tg % Target Object
end
```

コールバック関数 ConnectToTargetButtonPushed に Target computer との接続するためのコードを追加します。

```
% Button pushed function: ConnectToTargetButton
function ConnectToTargetButtonPushed(app, event)
    app.TargetStatusEditField.Value = 'Connecting to target...';

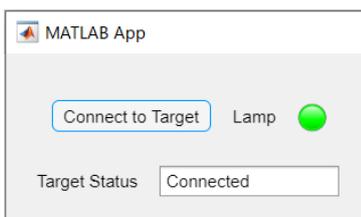
    % Get default target settings
    app.tg = slrealtime;

    try
        % Connect to target
        app.tg.connect;
        % Check if the target is correctly connected and if yes, change color of lamp to green
        if app.tg.isConnected
            app.Lamp.Color = [0,1,0];
            app.TargetStatusEditField.Value = 'Connected';
        else
            app.Lamp.Color = [1,0,0];
            app.TargetStatusEditField.Value = 'Connection failed!';
        end
    catch E
        app.Lamp.Color = [1,0,0];
        app.TargetStatusEditField.Value = 'Connection failed!';
    end
end
```

コードを入力し終わったら、Run ボタンでアプリケーションを実行します。



Connect to Target ボタンを押して、正常に Target computer と接続できた場合、下記のように表示されます。



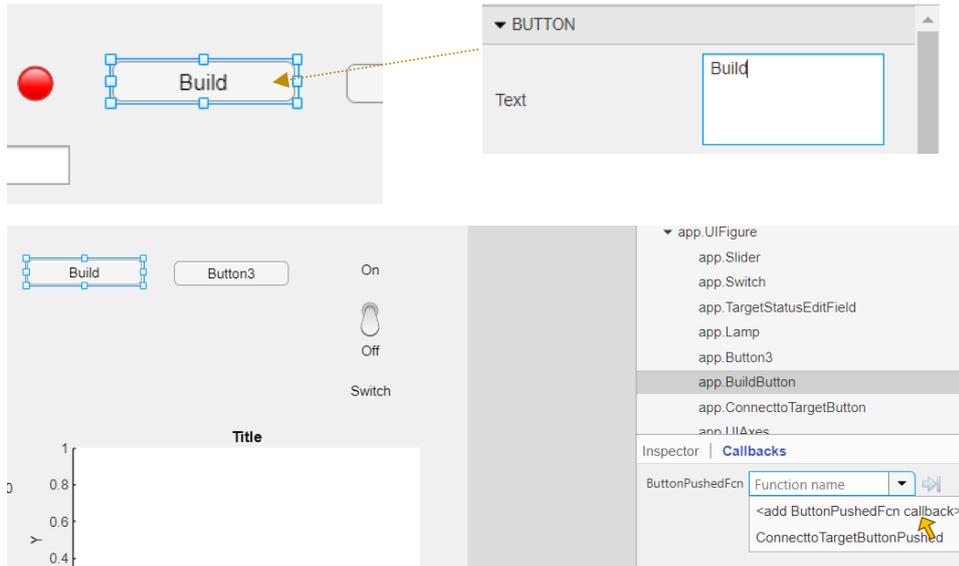
※もし接続できない場合は *Simulink Real-time Explorer* 上で接続できることを確認してください。

Simulink Real-time Explorer でも接続できない場合はネットワーク設定など App Designer 以外に問題があります。

8-4-4 モデルのビルド

ふたつめのボタンでモデル slrt_ex_osc.slx をビルドできるようにします。

Connect to Target ボタンと同様に、ふたつめのボタンの名称を変更し、コールバック関数を作成します。



またビルドするモデルの名称を格納するプライベート変数を作成します。

変数宣言時にモデルの名称も入力します。

```
properties (Access = private)
    tg % Target Object
    modelName = 'slrt_ex_osc' % Name of the Simulink model
end
```

※モデル名称を変更している場合は、変更後の名称を入力します。

作成されたコールバック関数 BuildButtonPushed にビルドを実行するためのコードを追加します。

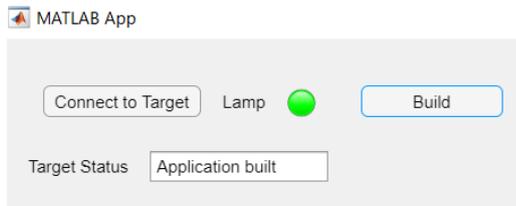
```
% Button pushed function: BuildButton
function BuildButtonPushed(app, event)
    app.TargetStatusEditField.Value = 'Building model ...'; % Update status output
    try
        rtwbuild(app.modelName);

        app.TargetStatusEditField.Value = 'Application built';
    catch E
        app.TargetStatusEditField.Value = 'Error building model.';
    end
end
```

コードの入力が完了したら slrt_ex_osc.slx を Simulink で開き、Configuration Parameters の Stop time を inf にしておいてください。これは後にモデルのパラメータの設定や信号の確認をするためです。



Run ボタンでアプリを起動して正常にビルドできた場合、下記のように表示されます。



MATLAB のコマンドウィンドウ上でもビルドの様子を確認することができます。

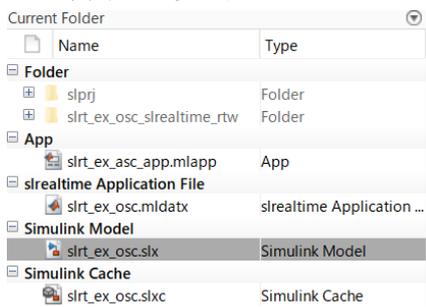
```
D:\speedgoat\download\mdl\slrt_ex_osc\slrt_ex_osc_slrealtime_rtw\instrumented>exit 0
### Invoking custom build hook: CodeGenAfterMake
### Successful completion of build procedure for: slrt_ex_osc
### Created MLDATX ..\slrt_ex_osc.mldatx
### Invoking custom build hook: CodeGenExit
### Simulink cache artifacts for 'slrt_ex_osc' were created in 'D:\speedgoat\download\mdl\
Build Summary

Top model targets built:

Model      Action                      Rebuild Reason
-----
slrt_ex_osc Code generated and compiled  Generated code was out of date.

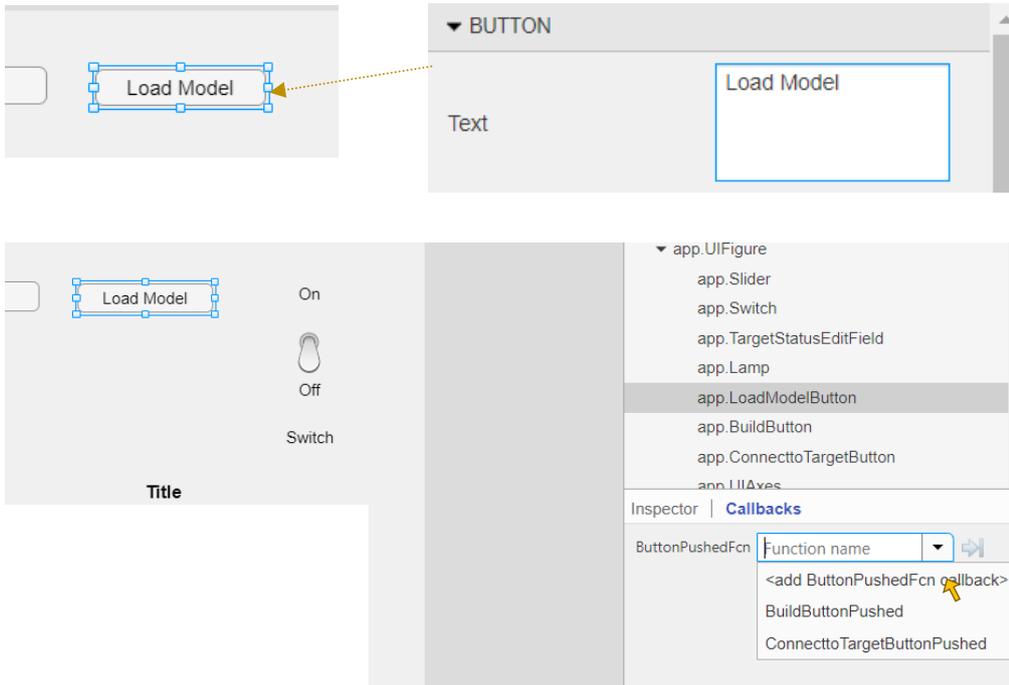
1 of 1 models built (0 models already up to date)
Build duration: 0h 0m 40.935s
^^
```

また下図のようにリアルタイムアプリケーション slrt_ex_osc.mldatx が生成されます。



8-4-5 リアルタイムアプリケーションのロード

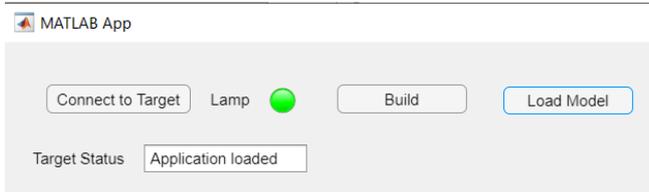
ビルドで生成されたリアルタイムアプリケーションをロードできるようにします。
最後のボタンの名称を変更し、コールバック関数を作成します。



作成されたコールバック関数 `LoadModelButtonPushed` にリアルタイムアプリケーションをロードするためのコードを追加します。

```
% Button pushed function: LoadModelButton
function LoadModelButtonPushed(app, event)
    app.TargetStatusEditField.Value = 'Loading application...';
    try
        load(app.tg, app.modelName);
    |
        app.TargetStatusEditField.Value = 'Application loaded';
    catch E
        app.TargetStatusEditField.Value = 'Error loading application on target.';
    end
end
end
```

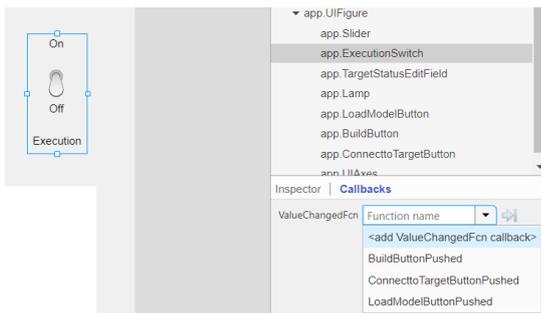
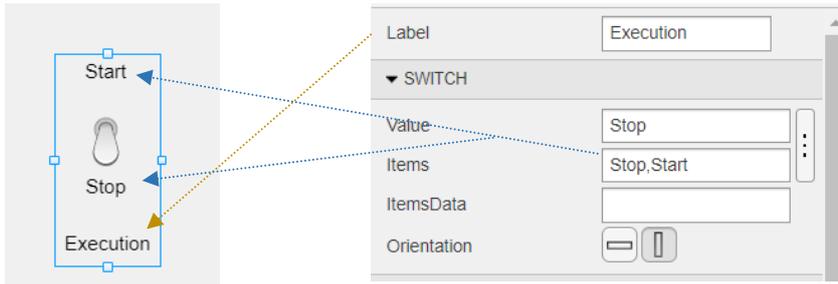
正常にロードできた場合、下記のように表示されます。



Target screen 上にもリアルタイムアプリケーションがロードされた则表示されます。

8-4-6 リアルタイムアプリケーションの実行

Target computer へロードしたリアルタイムアプリケーションを開始・停止できるようにします。
トグルスイッチの名称を変更し、コールバック関数を作成します。



作成されたコールバック関数 ExecutionSwitchValueChanged にリアルタイムアプリケーションを開始・停止するためのコードを追加します。

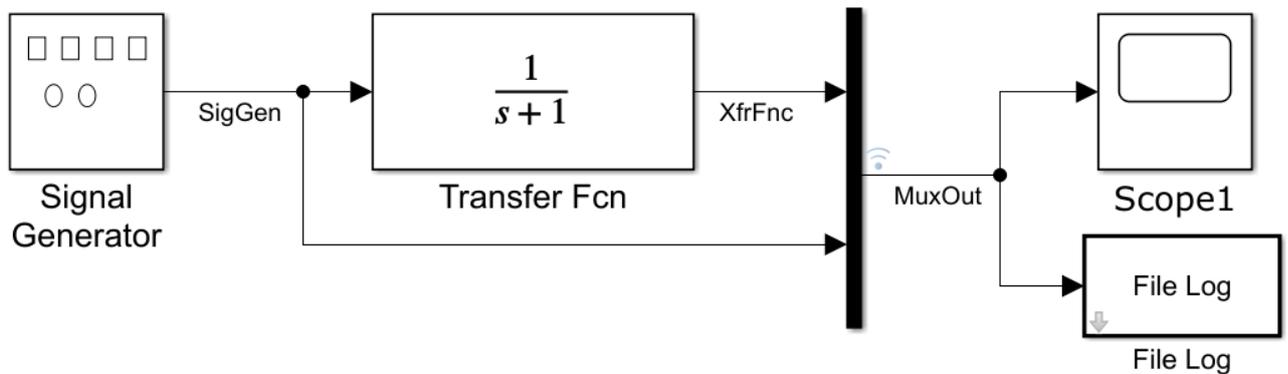
```

% Value changed function: ExecutionSwitch
function ExecutionSwitchValueChanged(app, event)
    value = app.ExecutionSwitch.Value;
    switch value
        case 'Start'
            start(app.tg);    % Start real-time application
            app.TargetStatusEditField.Value = 'Application running';
        case 'Stop'
            stop(app.tg);    % Stop real-time application
            app.TargetStatusEditField.Value = 'Application stopped';
    end
end
    
```

正常に開始できた場合、Target screen 上の EXECUTION TIME にリアルタイムアプリケーションを実行している時間が表示されます。

8-4-7 信号の表示

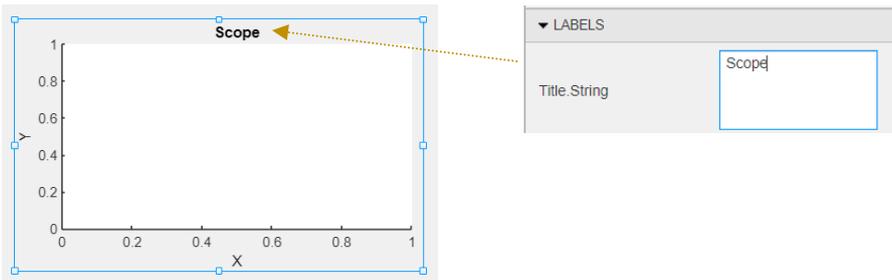
Simulink モデルのブロック線図上の信号線の値を表示できるようにします。
今回は下図の SigGen と XfrFcn を表示します。



slrt_ex_osc
Simulink Real-Time example model

Copyright 2020 The MathWorks, Inc.

信号線の値は作成しておいた Axes に表示します。
Axes のタイトル名称を変更しておきます。



次にパーツのハンドラとなるプライベート変数 hInst を作成します。

```
properties (Access = private)
    tg % Target Object
    modelName = 'slrt_ex_osc' % Name of the Simulink model

    hInst % Instrumentation for real-time visualization
end
```

リアルタイムアプリケーションのロード時に、信号線の値を Axes に紐づけたり、Axes の横軸幅（時間幅）を変更したりといった設定をまとめて行うコードを追加します。

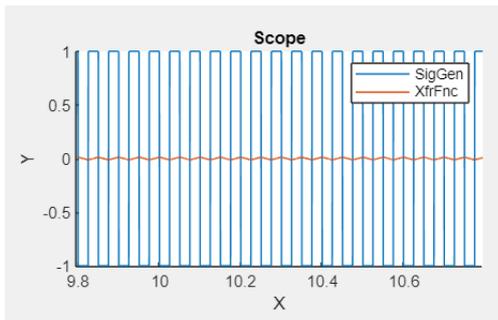
```
% Button pushed function: LoadModelButton
function LoadModelButtonPushed(app, event)
    app.TargetStatusEditField.Value = 'Loading application...';
    try
        load(app.tg, app.modelName);

        app.TargetStatusEditField.Value = 'Application loaded';

        % Set Instrument Handler
        app.tg.removeAllInstruments;
        app.hInst = slrealtime.Instrument();
        % connect signals to the axes
        app.hInst.connectLine(app.UIAxes, 'SigGen');
        app.hInst.connectLine(app.UIAxes, 'XfrFnc');
        % Set time span
        app.hInst.AxesTimeSpan = 1;
        % add a legend
        legend(app.UIAxes);
        % Clear previous data
        app.hInst.clearScalarAndLineData;
        % Validate and add instrumentation to target
        app.tg.addInstrument(app.hInst);
```

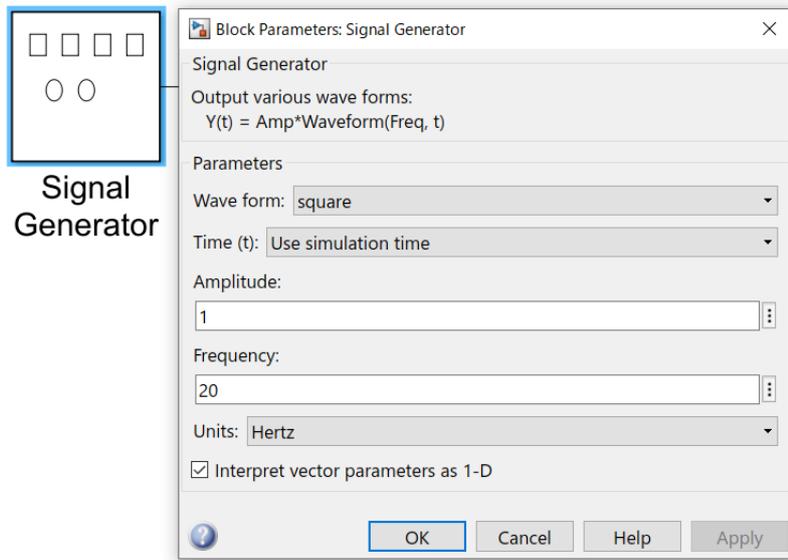
catch E

アプリケーションを実行して、下図のように表示されれば正常に動作しています。

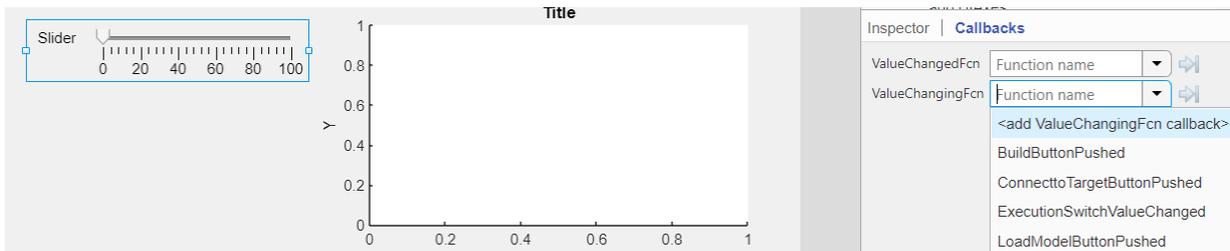


8-4-8 パラメータの設定

Simulink モデルのブロックのパラメータを変更できるようにします。
今回は下図の Signal Generator の Amplitude を変更対象とします。



まずは Slider のコールバック関数を追加します。
Slider にはふたつのイベントにコールバック関数を設定できます。
今回は ValueChangingFcn を追加します。



Amplitude を設定するためのコードを追加します。

```
% Value changing function: Slider
function SliderValueChanging(app, event)
    changingValue = event.Value;
    app.tg.setparam([app.modelName, '/Signal Generator'], 'Amplitude', changingValue);
end
```

アプリケーションを実行し、スライダを操作してください。
スライダの値に応じて、Axes の縦軸の幅が変動すれば正常に動作しています。

