



Embedded Code Generation Tutorial

Code Generation Workflow using PLECS STM32 TSP

PLECS STM32 TSPを使用したコード生成ワークフロー

Tutorial Version 1.0

はじめに

このチュートリアルでは、PLECS CoderとSTM32 Target Support Package(TSP)を使用してSTM32マイクロコントローラ(MCU)を使用する方法を学習します。

始める前に

- "STM32のコード生成の紹介(Introduction to PLECS STM32 Code Generation)"および"STM32 TSPを使用したトリガ設定(Trigger Configuration using PLECS STM32 TSP)"というチュートリアルで紹介している基本概念を理解していることを確認してください。これは上級者向けのチュートリアルです。
- 今後もSTM32 NUCLEO-G474REボード[1]を使い続けていく予定です。
- オプションとして、[課題2.1](#)では以下の項目を使用します:
 - RT Box Target Support Library: RT Boxユーザマニュアル[2]の**クイックスタート**に示されているように、PLECSとRT Box設定に関する段階的な指示に従ってください。
 - PLECS RT Box 1を1台。
 - RT Box LaunchPad-Nucleoインタフェースボード[3]を1枚(以下"インタフェースボード"と呼びます)。

RT Boxをお持ちでない場合は、[課題2.2](#)に示すように、2つ目のSTM32 NUCLEO-G474REボード[1]を使用して、コントローラがリアルタイムでどのように動作するかを確認することもできます。

 **注意** RT Boxシミュレーションをオフライン(PLECS内)で実行する場合、RT Boxは必要ありませんが、RT Box Target Support Libraryをインストールしておく必要があります。

演習1: コード生成ワークフロー

PLECSでは、オフラインおよびリアルタイムシミュレーション用に複数のモデルファイルを作成する必要はありません。オフラインシミュレーション中、STM32 Targetライブラリブロックは通常のPLECSブロックとして動作します(PLECS回路図で任意のブロックを右クリックし、**サブシステム** -> **サブシステムのモデル表示**でマスク内を表示してオフラインの"回路図"実装を確認します)。したがって、同じPLECSモデルをオフラインシミュレーションとリアルタイムシミュレーションの両方に使用できます。

この演習では、閉ループコントローラを持つ降圧コンバータを使用したPLECSコード生成ワークフローについて説明します。

課題1.1 事前作成した降圧コンバータモデルを調査

 **あなたのタスク:** 添付ファイルのPLECSモデルstm32_codegen_workflow_buck.plecsを開きます。このモデルは、[図1](#)および[図2](#)に示すように、電圧および電流制御ループが事前に構築されている降圧コンバータのモデルです。モデルを調査してシミュレーションを開始します(**Ctrl + T**)。PLECSスコープを開いてシミュレーション結果を観察します。

 **注意** このモデルには、**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ー...** -> **初期化**タブ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。

このControllerサブシステムは、STM32 MCU上で構築可能なモデルに変換できます。これは、STM32 Targetライブラリから必要なブロックをControllerサブシステムに追加するだけで簡単に行えます。

図1: 降圧コンバータ回路とコントローラサブシステムのトップレベル回路図

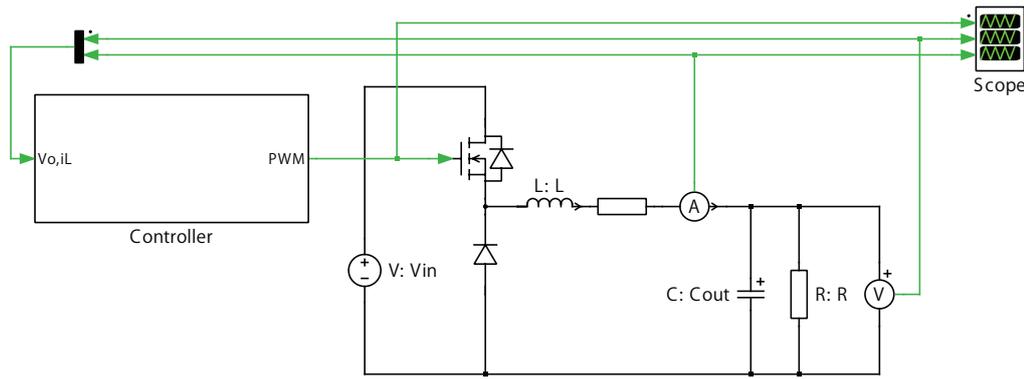
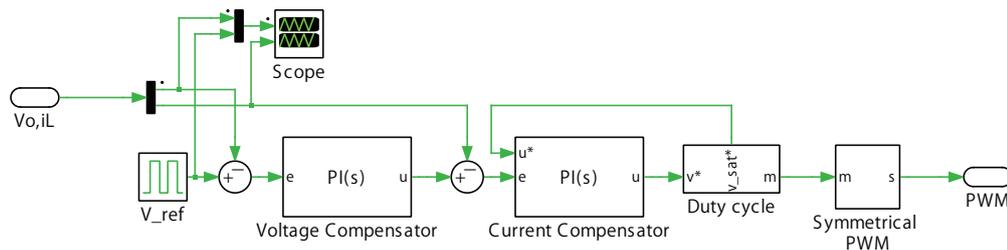


図2: Controllerサブシステムの回路図



課題1.2 PLECSモデルの修正

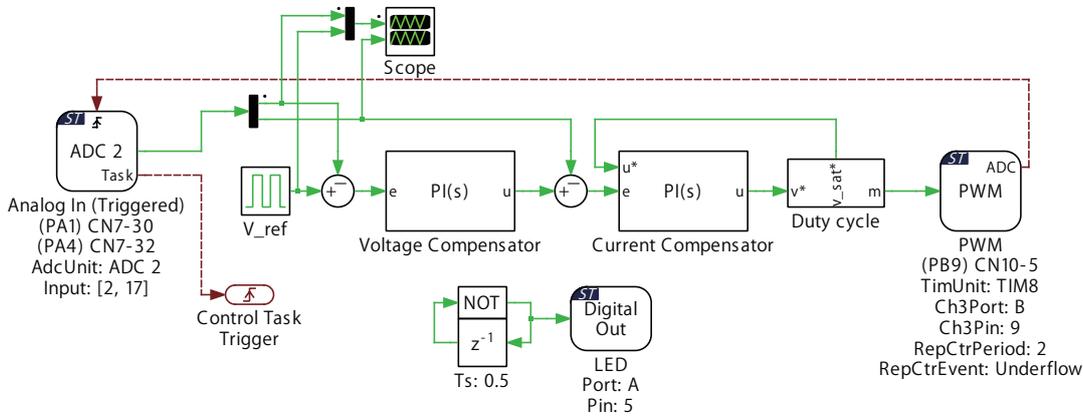


あなたのタスク:

- 1 メイン回路図から、"Controller"サブシステムを右クリックし、**サブシステム -> 実行の設定...**を選択します。構成ウィンドウで、"**コード生成機能の有効化**"チェックボックスを選択し、変更を**適用**します。これにより、PLECSモデル全体をビルドすることなく、コントローラサブシステムのみをターゲットデバイス上に構築できるようになります。
- 2 図3に示すように、"Controller"サブシステム内で、"Vo, iL"信号入力ブロックをSTM32 TargetライブラリのAnalog In (Triggered)ブロックに置き換えて、出力電圧とインダクタ電流の測定値をモデル環境に取り込みます。
- 3 ゲート信号は PWMブロックによって生成できます。このブロックへの入力は、[0, 1]の範囲のデューティ比です。したがって、"**対称PWM**"および"**PWM**"の出力信号ポートブロックを削除し、**STM32 Target**ライブラリのPWMブロックに置き換えます。
- 4 PWMブロックのパラメータ:
 - PWMブロックのパラメータウィンドウの**全般**タブで、**TIM Unit**をTIM8として設定し、**Carrier frequency**を適切に設定します。
 - 各TIMユニットは、最大3つのPWMチャンネルで単一の PWM出力または補完的なPWMペアを独立して生成できます。**Channel 3**タブで、**Mode**をSingle outputとして選択し、**Port**をB、**Pin**を9に設定します。他のすべてのチャンネルをDisable (無効)にします。
- 5 Analog In(Triggered)ブロックのパラメータ:
 - アナログ入力があるため、ADCブロックのパラメータをベクトル化できます。**ADC unit**をADC2として選択し、**Analog input channel**を[2, 17]に設定します。

- STM32 MCUは0~3.3V間の値のみを送受信するため、ADC入力をそれに応じてスケーリングする必要があります。したがって、[図4](#)に示すように、1/5のゲインを追加して、Plantから取得したフィードバックを縮小します。次に、Analog Inブロックのパラメータから、**Scale**を5に変更します。

図3: STM32 Targetライブラリのブロックを含むControllerサブシステム



6 トリガ構成:

- PWMジェネレータ、ADC、および制御タスクトリガ間のトリガ信号を接続して、割り込みシーケンスを明示的に定義します。
- PWMブロックのパラメータウィンドウの**Trigger**タブで、**ADC Trigger**をEnable(有効)にし、**Repetition counter period**を2に設定し、**Trigger event**をUnderflowとして構成します。つまり、PWMブロックは、キャリア値が最小値に達するたびに、ADCの変換開始信号を生成します。詳細な説明については、PWMブロックのヘルプセクションを参照してください。
- 次に、Analog In (Triggered)ブロックのパラメータ ウィンドウで、**Trigger Source**をShow Trigger Portとして設定し、PWMブロックからのADCトリガ出力をADCブロックのトリガ入力に接続します。
- ADCの変換終了後に制御タスクを実行するように構成するには、[図3](#)に示すように、ADCのタスクトリガ出力を **STM32 Target**ライブラリの制御タスクトリガブロックに接続します。
- ADCとTask triggerの関係を[図5](#)に示します。

7 最後に、"STM32のコード生成の紹介(Introduction to PLECS STM32 Code Generation)"チュートリアルで説明しているように、LEDを点滅させる回路を組み込みます ([図3](#)参照)。

図4: コード生成用にコントローラサブシステムが有効になっている降圧コンバータ回路

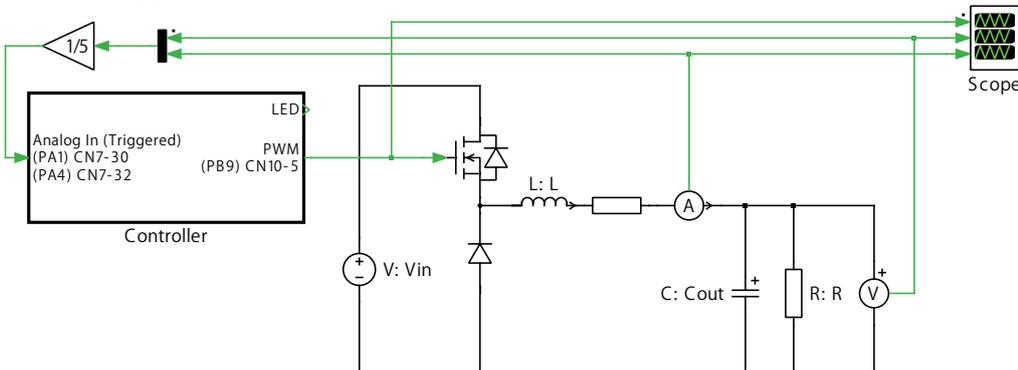
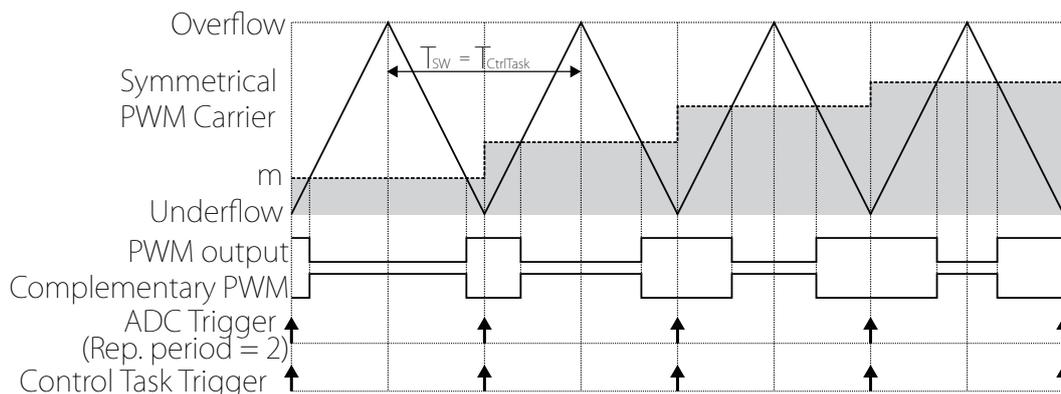


図5: PWMキャリア、ADCおよび制御タスクトリガの関係



課題1.3 Coderオプション、パラメータのインライン化、外部モードの設定



あなたのタスク: Coderオプションウィンドウで、左側のシステムリストの下にある“Controller”を選択します。**タスクタブ**で、**離散化ステップサイズ**を $1/f$ (f はスイッチング周波数)に設定します。出力電圧リファレンスをリアルタイムで調整できるようにするには、“V_ref”ブロックを**パラメータのインライン化タブ**の**例外リスト**ウィンドウにドラッグアンドドロップします。**ターゲットタブ**で、“STM32のコード生成の紹介”チュートリアルの説明に従って、STM32 G474REのコード生成パラメータを設定します。次に、**External Mode**サブタブから通信オプションとしてJTAGを選択し、**Target buffer size**を10000に設定します[4]。



この段階で、モデルはstm32_codegen_workflow_1.plecsと同じになります。ControllerサブシステムをSTM32 MCUに直接構築できるようになりました。PLECSでオフラインシミュレーションを実行し、シミュレーション結果を確認してください。

演習2: プラントを使ったリアルタイムの例

プログラムしたMCUを使用して実際の電力ステージを制御する前に、まずハードウェアインザループ(Hardware-in-the-loop: HIL)シミュレーションを使用して組み込みコントローラの動作を検証します。PLECS RT Boxを使用してHILシミュレーションを実行します。PLECS RT Boxをお持ちでない場合は、[課題2.2](#)で説明しているように、2台のSTM32 MCUを“Plant”として使用します。

課題2.1 PLECS RT Boxを使用したHILシミュレーション



あなたのタスク: この課題では、ControllerサブシステムをSTM32 MCUにデプロイし、PLECS RT Boxとアナログ信号およびデジタル信号を交換して、そのパフォーマンスをリアルタイムで検証します。

- 1 添付ファイルから PLECSモデルstm32_codegen_workflow_plant.plecsを見つけて開きます。このモデルには、事前に構築された降圧コンバータの“Plant_RT Box”サブシステムが含まれています。[演習1](#)の降圧コンバータは、PLECS RT Boxライブラリのコンポーネントを追加することで、PLECS RT Boxとアナログ信号およびデジタル信号を交換できるように変更されています。
- 2 [課題1.3](#)の最後にある同じPLECSモデルファイルで、降圧コンバータの既存の回路を“Plant_RT Box”サブシステムに置き換えます([図7](#)を参照)。

図7: プラントとコントローラサブシステムのトップレベルの回路図

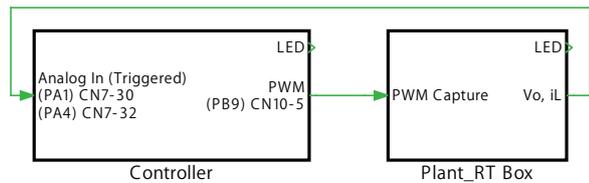
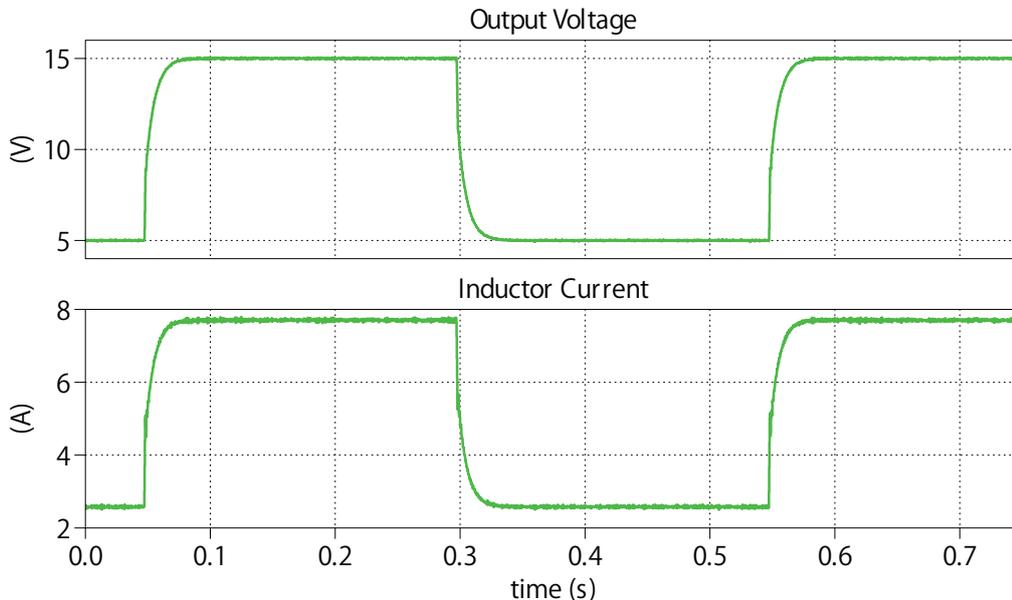


図8: STM32 MCUを使用した出力電圧とインダクタ電流のリアルタイム測定



課題2.2 2つのSTM32 MCUを使用したHILシミュレーション

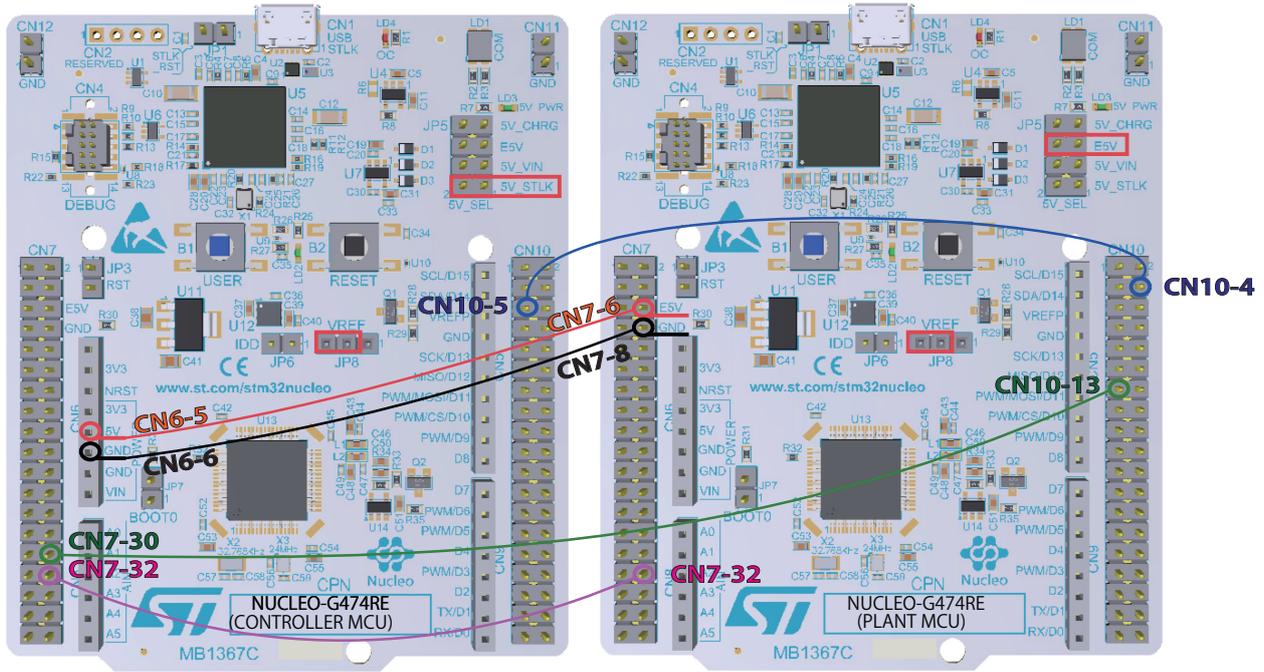
PLECS RT Box が利用できない場合は、2つ目のSTM32 MCUを"Plant"として使用します。以下、このMCUを"Plant MCU"と呼びます。



あなたのタスク: この課題は、ControllerサブシステムをSTM32 MCU(以下、"Controller MCU"と呼びます)にデプロイし、Plant MCU間でアナログ信号とデジタル信号を交換して、そのパフォーマンスをリアルタイムで検証します。

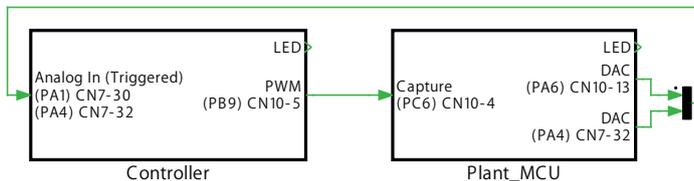
- 1 添付ファイルのPLECSモデルstm32_codegen_workflow_plant.plecsを開きます。このモデルには、[演習1](#)の降圧コンバータの平均モデルを含む"Plant_MCU"サブシステムが含まれています。2つ目のSTM32 MCU間でアナログ信号およびデジタル信号を交換するためのSTM32ライブラリコンポーネントも含まれています。
- 2 [課題1.3](#)の最後にある同じPLECSモデルファイルで、降圧コンバータの既存の回路を"Plant_MCU"サブシステムに置き換えます([図10](#)を参照)。
- 3 Plant MCUをフラッシュ:
 - まず、USBケーブルを介してPlant MCUをホストコンピュータに接続し、次に**Coderオプション**ウィンドウから、Plant MCU上に"Plant"サブシステムを**ビルド**します。
 - 正しくプログラムされていれば、STM32ボード上のLED "LD2"が点滅します。
- 4 必要なハードウェア接続を行います:
 - まず、Plant MCUをホストコンピュータから切断し、USBケーブルを介して Controller MCUをホストコンピュータに接続します。次に[図9](#)のように、ジャンプワイヤを使用して、[表1](#)に従って2つのMCUのピンを接続します。

図9: 左側がController MCU, 右側がPlant MCU



5 Controller MCUをフラッシュし、外部モードに接続: **Coderオプション**ウィンドウから、Controller MCU上に"Controller"サブシステムを**ビルド**します。生成されたコードが STM32ターゲット上で実行されたら、"STM32のコード生成の紹介"チュートリアル¹の指示に従って、外部モードに**接続**してPLECSスコープを更新します。図8²に示すように、出力電圧とインダクタ電流のステップ応答がリアルタイムで表示されます。

図10: プラントとコントローラサブシステムのトップレベルの回路図



 この段階で、モデルはstm32_codegen_workflow_3.plecsと同じになります。

 **注意** 目的のステップ応答でPLECSスコープを更新するには、**Coder オプション**ウィンドウの**外部モード**タブで、"ターゲットチャンネル"、"感度"、"ターゲットレベル"、および"ターゲット遅れ"パラメータを適切に設定します。より長い時間範囲を表示するには、"間引き"パラメータを増やすか、**ターゲット**タブ -> **External Mode**サブタブから"Target buffer size"を増やします。

表1: ジャンパ線を使用して2つのMCUのピンを接続

Controller MCU	Plant MCU	機能
ジャンパJP8を2-3に設定	ジャンパJP8を2-3に設定	外部ADリファレンス(Vref+)として3.3Vを選択
	ジャンパJP5を“E5V”に設定	Controller MCUから供給される5V
ジャンパJP5を“5V_STLK”に設定		ホストPCのUSBポートから供給される5V
CN6-5, 5V	CN7-6, labeled E5V	5V主つよく
CN6-6, GND	CN7-8, labeled GND	GND
CN7-30 (PA1)	CN10-13 (PA6)	出力電圧を検出
CN7-32 (PA4)	CN7-32 (PA4)	インダクタ電流を検出
CN10-5 (PB9)	CN10-4 (PC6)	生成したPWM信号

参考文献

- [1] STMicroelectronics, NUCLEO-G474RE: <https://www.st.com/en/evaluation-tools/nucleo-g474re.html>
- [2] *RT Box User Manual*, Plexim GmbH, Online: <https://www.plexim.com/sites/default/files/rtboxmanual.pdf>
日本語訳RT BOXユーザマニュアル: <https://adv-auto.co.jp/products/plexim/manual.html>
- [3] *RT Box LaunchPad-Nucleo Interface Board*, Plexim GmbH, Online:
https://www.plexim.com/sites/default/files/launchpadnucleointerface_manual.pdf
日本語訳LaunchPad-Nucleoインタフェースボードユーザマニュアル: <https://adv-auto.co.jp/products/plexim/manual.html>
- [4] STM32 Target Support User Manual: <https://plexim.com/sites/default/files/stm32manual.pdf>
日本語訳STM32 Target Supportユーザマニュアル: <https://adv-auto.co.jp/products/plexim/manual.html>
- [5] STM32G4 Nucleo-64 boards (MB1367) User Manual:
https://www.st.com/resource/en/user_manual/um2505-stm32g4-nucleo64-boards-mb1367-stmicroelectronics.pdf

改訂履歴:

Tutorial Version 1.0 初版

plexim

☎ +41 44 533 51 00

+41 44 533 51 01

✉ Plexim GmbH

Technoparkstrasse 1

8005 Zurich

Switzerland

@ info@plexim.com

<http://www.plexim.com>

Pleximへの連絡方法:

Phone

Fax

Mail

Email

Web

KESCO KEISOKU ENGINEERING SYSTEM

計測エンジニアリングシステム株式会社

<https://kesco.co.jp>

Embedded Code Generation Tutorial

© 2002–2023 by Plexim GmbH

このマニュアルに記載されているソフトウェアPLECSは、ライセンス契約に基づいて提供されています。ソフトウェアは、ライセンス契約の条件の下でのみ使用またはコピーできます。Plexim GmbHの事前の書面による同意なしに、このマニュアルのいかなる部分も、いかなる形式でもコピーまたは複製することはできません。

PLECSはPlexim GmbHの登録商標です。MATLAB、Simulink、およびSimulink Coderは、The MathWorks、Inc.の登録商標です。その他の製品名またはブランド名は、それぞれの所有者の商標または登録商標です。