

Hands on Introduction to PLECS TI C2000 Code Generation PLECSによるTI C2000コード生成の紹介

Tutorial Version 1.0



1 はじめに

このチュートリアルでは、Texas Instruments(TI) C2000プロセッサファミリ向けのPLECS組み込みコード生成ツールの主要 な関数ブロックについて学びます。TI C2000マイクロコントローラ(MCU)は、リアルタイムのパワーエレクトロニクス制御用 を目的としています。

始める前に このチュートリアルを開始する前に、以下を利用できるよう、インストールしておいてください:

- 1つのPLECS Coderのライセンス。
- <u>TI C2000 Target Support Library</u>: TI C2000 Target Support User ManualのクイックスタートでTI C2000のコード生成の構成に関するステップバイステップの指示に従ってください[1]。
- 1つのTI C2000 Piccolo MCU F28069 LaunchPad [2]開発キット。以後"LaunchPad"と呼びます。
- <u>RT Box 1 Target Support Library</u>: RT Box User Manual [3]のクイックスタート、PLECSとRTボックスの構成に関するステップ バイステップの指示に従ってください。

オプションで、コントローラをリアルタイムで検証するために、次の項目をセクション4、5、および6で使用します。これらの項目 は、オフライン(PLECS)で閉ループシミュレーションを実行するときは必要ないことに注意してください。

- 1台のPLECS RT Box 1
- 1つのRT Box LaunchPadインタフェースボード[4]。以後"Interface Board"と呼びます。

RT Box(オプション)を使用したチュートリアルのハードウェアの配置は、図1に示しています。以下の構成も確認してください:

- Interface Board(緑)のジャンパ**RST**はオープンのままです。
- ・ LaunchPad(赤)のジャンパJP6はオープンのままです。

図1: RT Boxを使用したこのチュートリアルのハードウェア設定(オプション)



2 LEDの点滅

最初の演習では、MCUのGPIO 34に接続されたLaunchPad上の赤色LED "D9"を点滅させる簡単なプログラムを作成します。 ウィンドウのドロップダウンメニューからライブラリブラウザを開きます。メインの回路図に"サブシステム"ブロックを配置し、 "Controller"というラベルを付けます。 "Controller"サブシステム内で、デューティサイクル50%、0.5HzでLEDを点滅させるモデルを作成します。LEDは1秒間点灯し、 その後1秒間消灯します。簡単な方法は、図2に示す概略図を参照してください。Digital OutブロックはTI C2000 Targetライブラリ から取得する必要があることに注意してください。Digital OutブロックのDigital output GPIO number(s)パラメータを34に 設定します。

回路図を完成したら、シミュレーション -> シミュレーションパラメータ...ウィンドウの初期化タブにあるモデル初期化コマンド ウィンドウに次のコード行を入力して、実行ステップサイズを定義します: Ts.Controller = 100e-6;





"Controller"サブシステムは、構成に応じて、TI 28069 LaunchPadのターゲット固有のコードに直接変換できます。以下の 手順に従って、"Controller"サブシステムを LaunchPadにアップロードします。

🚰 あなたのタスク:

- 1 最上位レベルの回路図から"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ます。
- 2 Coder -> Coderオプション...ウィンドウで、システムリストから"Controller"を選択します。
- 3 タスクタブから、"離散化ステップサイズ"フィールドにTs.Controllerと入力します。
- 4 次に、ターゲットタブで、ドロップダウンメニューからTI2806xデバイスを選択します。
- 5 Build typeにBuild and programを選択し、BoardにLaunchPadを選択します。
- 6 LaunchPadがUSBケーブル経由でホストコンピュータに接続されていることを確認してから、ビルドをクリックします。

ビルドプロセスが完了するまでお待ちください。正しくプログラムされていれば、LaunchPadの赤色LED "D9"は0.5Hz、50%のオン時間で点滅します。

この段階では、モデルは参照モデルc2000_tutorial_1.plecsと同じになるはずです。

3 PWM出力

TI C2000 TargetライブラリのPWMコンポーネントは、パワーエレクトロニクスコンバータのスイッチングデバイスを制御 するための PWM信号を生成するために使用されます。

🚽 あなたのタスク:

- 1 "Controller"サブシステムにTI C2000 TargetライブラリからPWMブロックを追加し、PWM generator(s)パラメータが1に、 Carrier frequencyパラメータが10e3に設定されていることを確認します。
- 2 定数ブロックを使用して、PWMブロックにデューティサイクルを供給します。50%のデューティサイクルを持つPWM信号 を生成するには、定数の値を0.5に設定する必要があります。この手順については、図3に示す回路図を参照してください。

図3: PWMブロックを追加したコントローラの回路図



- 3 MCUの動作中にPWM信号のデューティ サイクルを変更するには、Coder -> Coderオプション...ウィンドウからパラメータの インライン化タブの例外リストに定数ブロックをドラッグします。
- 4 次に、前の演習の最後と同じ手順に従って、"Controller"サブシステムをMCUにアップロードします。
- 5 アップロードが完了したら、Coderオプションウィンドウの外部モードタブからMCUをホストPCに接続します。まず、ターゲット デバイスフィールドの横にある、「ボタンをクリックします。適切なデバイス種類を選択し、検索ボタンをクリックして利用可能 なデバイスを表示します。最後に、デバイス名ドロップダウンメニューを使用して、使用可能なデバイスのリストから適切な デバイスを選択します。OKをクリックして選択したデバイスを確認し、接続ボタンをクリックして外部モードを有効にします。

この時点で、定数ブロックのパラメータウィンドウを通じてPWMのデューティサイクルを変更できるはずです。

オプション: アナログオシロスコープを使用して、LaunchPadの**J4**コネクタのピン**39**と**40**に2つのプローブを接続してみます。反対 極性の2つのPWM信号を測定します。定数ブロックの値を変更すると(外部モードでMCUに接続されている場合)、デューティ サイクルの変化をオシロスコープで観察できます。

4 降圧コンバータのハードウェアインザループシミュレー ション

ハードウェアインザループ(HIL)シミュレーションは、組み込みコントローラの動作を検証する一般的な手法です。ここで、 プログラムしたLaunchPadをテストするための簡単なHILモデルを作成します。モデルに変更を加える前に、**外部モード**タブ の**切断**ボタンをクリックして外部モードを非アクティブ化します。

次に、最上位レベルの回路図に新しいサブシステムを作成し、"Plant"という名前を付けます。"Plant"サブシステムで降圧 コンバータのシミュレーション モデルを構築するには、図4を参照してください。プラントモデルの実行ステップ サイズを定義 するには、シミュレーション -> シミュレーションパラメータ...ウィンドウの 初期化タブにあるモデル初期化コマンドウィンドウ に次のコード行を入力します。Ts.Plant = 1e-6;



図4: 降圧コンバータのHILシミュレーションモデル

🚽 あなたのタスク:

1 スイッチングセルには、電気回路/パワー素子モジュールライブラリのハーフブリッジブロックを使用します。

- 2 回路図に示されている値を使用して、インダクタ、キャパシタ、抵抗器を構成します。
- **3** デマルチプレクサブロック経由でPWM Captureブロックを2つのIGBTに接続し、Digital Input Channel(s)(デジタル入力 チャネル)をベクトル[0 1]に設定します。Averaging interval (offline only)(平均間隔)をTs.Plantに設定します。
- **4** 最上位レベルの回路図から、"Controller"サブシステムのPWM出力を、"Plant"サブシステムの PWM Capture入力に接続 します。

この段階で、PLECSで電力コンバータと開ループコントローラをシミュレートできるようになりました。モデルを実行し、"Plant" サブシステムのPLECSスコープでインダクタ電流と出力電圧を観察します。

以下の手順に従って、"Plant"サブシステムを RT Boxにアップロードします:

🖞 あなたのタスク:

- 1 最上位レベルの回路図から"Plant"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ます。
- 2 Coderオプションウィンドウのシステムリストから、"Plant"を選択します。
- 3 タスクタブから、離散ステップサイズフィールドにTs.Plantと入力します。
- 4 ターゲットタブのドロップダウンメニューからPLECS RT Box 1を選択します。
- 5 利用可能な RT Boxを選択するには、ターゲットデバイスフィールドの横にある双眼鏡ボタンをクリックします。
- 6 ビルドをクリックします。

アップロードが成功したら、Coderオプションウィンドウのシステムリストから"Plant"を選択したままにします。次に、外部モード タブから接続ボタンを押して"Plant"サブシステムの外部モードをアクティブにし、下の自動トリガを有効化ボタンをクリック します。"Plant"サブシステム内でPLECSスコープを開きます。スイッチングのリップル周波数を含むインダクタ電流が約12A を中心とし、キャパシタ電圧が12Vであることを確認します。前述のように、"Controller"サブシステムの外部モードを有効に して、PWM出力のデューティサイクルを変更することもできます。そうすることで、降圧コンバータの平均インダクタ電流と キャパシタ電圧の変化が観察できます。

での段階では、モデルは参照モデルc2000_tutorial_2.plecsと同じになるはずです。

5 ADCサンプリング

閉ループ制御では、MCUのアナログ/デジタルコンバータ(ADC)を使用して、アナログ信号の測定値を、組み込み制御コードで 使用できる、信号を表す変数に変換します。この演習では、前回の演習で紹介したPWM出力ブロックと組み合わせて、さまざまな ADCサンプリング方法を練習します。 🖉 あなたのタスク:

- 1 TI C2000 Targetライブラリから ADCブロックを"Controller"サブシステムに配置し、Analog input channel(s)パラメータ を7に設定します。
- **2** この構成は、LaunchPadのジャンパ**J1**のピン**23**から供給される ADCチャネル7の入力が制御環境に読み込まれることを示しています。
- 3 次に、Scale(s)パラメータを10に設定します。図5に示すように、PLECSスコープコンポーネントをADCブロックの出力ポート に接続します。

図5: ADCブロックを追加したControllerモデル



m

ここで、"Plant"サブシステムを変更して、LaunchPadの電流検知信号を作成します。低電圧アプリケーションでは、回路電流 は通常、ローサイドスイッチから取得され、低コストのシャントベースの電流センサを導入できます。

췯 あなたのタスク:

Scope

Task

ADC

1 降圧コンバータのローサイドスイッチと直列に電流計を配置します。図6に示す電流計の方向に注意してください。

図6: Analog Outブロックを追加した降圧コンバータモデル

PWM



- 2 PLECS RT BoxライブラリからAnalog Outブロックを"Plant"サブシステムにドラッグし、名前を"Isw"に変更して、その入力 ポートを電流計に接続します。
- 3 Analog OutブロックのScale(スケール)パラメータを0.1に指定します。次に、Minimum output voltage(最小出力電圧) を0に、Maximum output voltage(最大出力電圧)を3.3に設定します。

これらの構成では、RT Boxのアナログ出力チャネル0は、V = I_{sw}・0.1に等しく、OV ~ 3.3Vの範囲に制限されたアナログ電圧 信号を生成します。ここで、I_{sw}は電流計からの電流測定値です。インタフェースボードでは、RT Boxのこのアナログ出力チャネル 0は、LaunchPad上のMCUのADC input 7に接続されています。

変更した"Controller"サブシステムをLaunchPadにアップロードし、"Plant"サブシステムを RT Box にアップロードします。 "Controller"サブシステムの外部モードを有効にし、PLECSスコープで出力を観察します。

この段階では、モデルは参照モデルc2000_tutorial_3_1.plecsと同じになるはずです。

(?) スコープ内の ADC信号が常にゼロなのはなぜですか(小さな振幅の不規則なノイズを除く)?

A 実際の電流はパルス状の性質を持っていますが、コントローラは電流がゼロの期間のみそれを測定します。

注意: MCUでは、通常、デューティサイクル(変調指数)を搬送波と比較することによってPWM信号が生成されます。 変調指数が搬送波より大きい場合、MCUはハイサイドスイッチをオンにし、ローサイドスイッチをオフにします。これに より、降圧コンバータのインダクタ電流が増加します。そうしないと、ローサイドスイッチに電流が流れ、インダクタ電流 が減少します。デフォルトでは、キャリアが"0"に達し、インダクタ電流がハイサイドスイッチを流れ始めると、ADC入力 チャネルの検知信号が読み取られます。したがって、図7に示すように、ゼロ電流が測定されます。



平均インダクタ電流を制御する場合、最小電流リップルと最大電流リップルの中間の電流をサンプリングすることが望ましいです。この目的のために、ローサイドスイッチの電流検知では、キャリアが最大値に達したときに電流検知信号を読み取るようにADCを再構成する必要があります。図8を参照して、次の変更を行ってください:

図8: ADCサンプリングと制御タスクの実行を構成



🚽 あなたのタスク:

- 1 PWMブロックのパラメータウィンドウを開き、イベントタブに切り替えます。ADC triggerパラメータをOverflowに変更 します。
- 2 次に、ADCブロックのパラメータウィンドウで、Trigger sourceパラメータをShow trigger portに変更します。
- 3 前の手順を完了すると、PWMブロックに追加のトリガ出力ポートが表示され、ADCブロックにトリガ入力ポートが表示 されます。これら2つのポートを接続します。
- 4 最後に、TI C2000 Targetから制御タスクトリガコンポーネントを取得し、ADCブロックの"Task"ポートに接続します。

要約すると、これらの構成により、PWMブロックはキャリア最大値でADCサンプリングをトリガし、ADCがサンプリングを終了 すると、制御タスク(LEDの点滅など)が実行されます。

変更した"Controller"サブシステムをアップロードし、外部モードを有効化します。これで、PLECSスコープでゼロ以外の電流 を確認できるはずです。

この段階では、モデルは参照モデルc2000_tutorial_3_2.plecsと同じになるはずです。

6 閉ループ制御

この演習では、降圧コンバータの閉ループ電流コントローラを構成します。図9を参照して"Controller"サブシステムを変更し、 次の操作を実行してください:

図9: 電流制御用閉ループPIレギュレータを備えたControllerサブシステム



🖞 あなたのタスク:

- 1 基準電流入力として機能するパルス発生器ブロック"Iref"を含めます。周波数を20Hz、High-state出力を7、Low-state出力 を5に設定します。
- 2 2つのゲイン(利得)ブロックの"Kp"ゲインを0.01に、"Ki"ゲインを40に設定します。
- 3 飽和ブロックの上限パラメータを0.9 に、下限パラメータを0.05に設定します。

- 4 PWMが非アクティブになったときに積分のオーバーフローを回避するには、積分器ブロックの飽和上限を1に、飽和下限 を0に設定します。次に、外部リセットパラメータでレベルを選択して、積分器ブロックのリセットポートを有効にします。図9 に示すように、手動切替スイッチ、定数ブロック、およびNOTを選択した論理演算子ブロックを接続します。論理演算子の 出力を積分器のリセットポートに接続します。
- 5 ゲイン"Kp"と"Ki"、手動切替スイッチブロック、定数ブロック"Iref"をCoderオプションウィンドウのパラメータのインライン化 タブにドラッグします。

変更したコントローラモデルを MCUにアップロードする前に、オフラインシミュレーションでコントローラのパフォーマンス を事前に検証できます。シミュレーションを開始し、回路の応答を観察します。"Controller"サブシステムのPLECSスコープ内 の波形は、図10のようになるはずです。"Plant"サブシステムの電流波形にはスイッチング周波数リップルが含まれますが、 "Controller"サブシステムでは基本成分のみが表示されることに注意してください。





これらの変更が完了したら、新しい"Controller"サブシステムを LaunchPadにアップロードし、**外部モード**と自動トリガを有効 にします。"Iref"ブロックからの電流参照の変化とトリガを同期することができます。ここで、PIレギュレータのパラメータをオン ラインで調整します。

🗹 あなたのタスク:

- 1 手動切替スイッチを切り替えて、MCUのPWM出力をアクティブにします。積分器がリセットされると、非常に小さなKpゲイン のみが有効になり、その結果デューティサイクルが低くなることに注意してください。飽和ブロックは、PWMのデューティ サイクルを0.05に固定します。したがって、PLECSスコープ上では少量の電流が測定されるはずです。
- 2 "Kp"と"Ki"のゲインを他の値に設定し、結果の測定値をPLECSスコープで観察します。実行時にPIパラメータへのすべての 変更の結果を確認できるように、"Controller"サブシステムの**外部モード**がアクティブになっていることを確認してください。
- 3 振動とオーバーシュートを減らすことを目的として、PIパラメータの調整を続けます。

Cの段階では、モデルは参照モデルc2000_tutorial_4.plecsと同じになるはずです。

7 結論

これで、PLECSでコントローラモデルを構築し、制御ロジックを実際の組み込みプロセッサに展開できました。将来、パワー エレクトロニクスコンバータのプロトタイプの制御設計を行う際には、PLECSツールチェーン環境を活用して、その柔軟性と 可観測性を活用することを検討してください。

8 参照

[1] TI C2000 Target Support User Manual, Plexim GmbH, Online:

https://www.plexim.com/sites/default/files/c2000manual.pdf 日本語版は以下から:

https://adv-auto.co.jp/products/plexim/manual.html

[2] Texas Instruments, "LAUNCHXL-F28069M Overview" User's Guide, SPRUIIIB, 2019.

[3] RT Box User Manual, Plexim GmbH, Online: https://www.plexim.com/sites/default/files/rtboxmanual.pdf 日本語版は以下から:

https://adv-auto.co.jp/products/plexim/manual.html

[4] RT Box LaunchPad Interface Board, Plexim GmbH, Online:

https://www.plexim.com/sites/default/files/launchpadinterfacemanual.pdf 日本語版は以下から:

https://adv-auto.co.jp/products/plexim/manual.html

改訂履歴: Tutorial Version 1.0 初版

	Plexim ⁴ Plexim ⁴	への連絡方法: Phone
	+41 44 533 51 01	Fax
	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	info@plexim.com	Email
	http://www.plexim.com	Web
Adva	ancing Automation Auto AUTO アドバン +81 3 5282 7047	オートメーションへの連絡方法: Phone
	+81 3 5282 0808	Fax
	ADVAN AUTOMATION CO.,LTD	Mail
	Tokyo, 101-0047 Japan	
@	T-9-5 Uchikanda, Chiyoda-ku Tokyo, 101-0047 Japan plecs_adva@adv-auto.co.jp	Email

Embedded Code Generation Tutorial

© 2002–2021 by Plexim GmbH

このマニュアルで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。