Embedded Code Generation DEMO MODEL

Simple SPI Model

シンプルなSPIモデル

- TI C2000 TSPでSPIプロトコルを検証するためのシンプルなモデル -

Last updated in C2000 TSP 1.5.2

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

1 はじめに

このデモでは、PLECS CoderとTI C2000 Target Support Packageを使用した、Texas Instruments (TI)社のC2000マイクロコントローラ(microcontroller: MCU)上でSerial Peripheral Interface(SPI)ブロックを使用するシンプルなモデルを紹介します。

シリアルペリフェラルインタフェース(Serial Peripheral Interface: SPI)は、プログラム可能な、長さ(1～16ビット)のシリアルビットストリームを構成可能なビット転送速度でデバイスにシフトインおよびデバイスからシフトアウトできるようにする高速同期シリアル入力/出力デバイスです。SPIは通常、MCUコントローラと外部ペリフェラル、または他のコントローラ間の通信に使用されます。

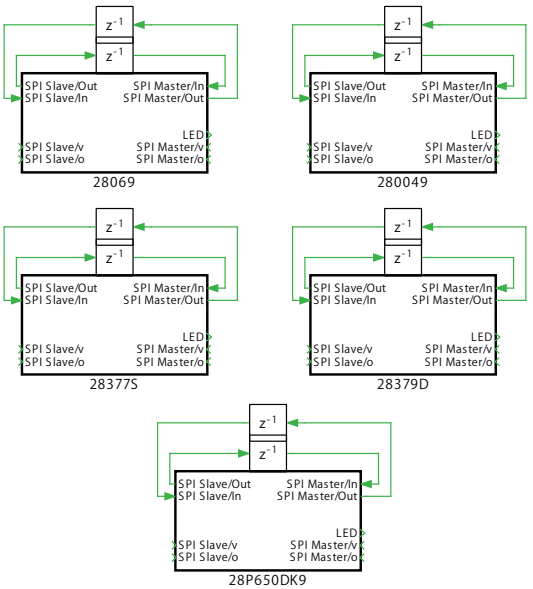
このモデルは、"28069"、"280049"、"28377S"、"28379D"、"28P650DK9"という5つの異なるサブシステムに分けられています。各サブシステムは、対応するTI C2000 LaunchPadハードウェアに個別にデプロイできます。次章では、モデルの簡単な説明と、シミュレーションする方法について説明します。

2 モデル

最上位レベルの回路図には、図1に示すように、5つの個別のサブシステムが含まれています。"28069"というラベルのサブシステムはTI 28069 LaunchPad [2]用に構成され、"280049"というラベルのサブシステムはTI 280049C LaunchPad [1]用に構成され、"28377S"というラベルのサブシステムはTI 28377S LaunchPad [3]用に構成され、"28379D"というラベルのサブシステムはTI 28379D LaunchPad [4]用に構成され、最後に"28P650DK9"というラベルのサブシステムはTI 28P650DK9 LaunchPad [5]用に構成されています。

各サブシステムはコード生成が有効になっており、サブシステムブロックの外側の枠線が太くなっています。このステップは、PLECS Coderを使用してサブシステムのモデルコードを生成するために必要です。この設定は、サブシステムを選択し、**編集** -> **サブシステム**メニューから**実行の設定...**を開いて、**コード生成機能の有効化**オプションを選択することで構成されます。

図1: 5つのサブシステムがあるモデルのトップレベルの回路図



SPIは、単一のコントローラと1つ以上のペリフェラルデバイスを備えたたコントローラ/ペリフェラルベースのインタフェースです。インタフェースは以下の信号で構成されています:

- **SPIPOCI** シリアルデータ入力(controller in/peripheral out)
- **SPISOMI** シリアルデータ出力(controller out/peripheral in)

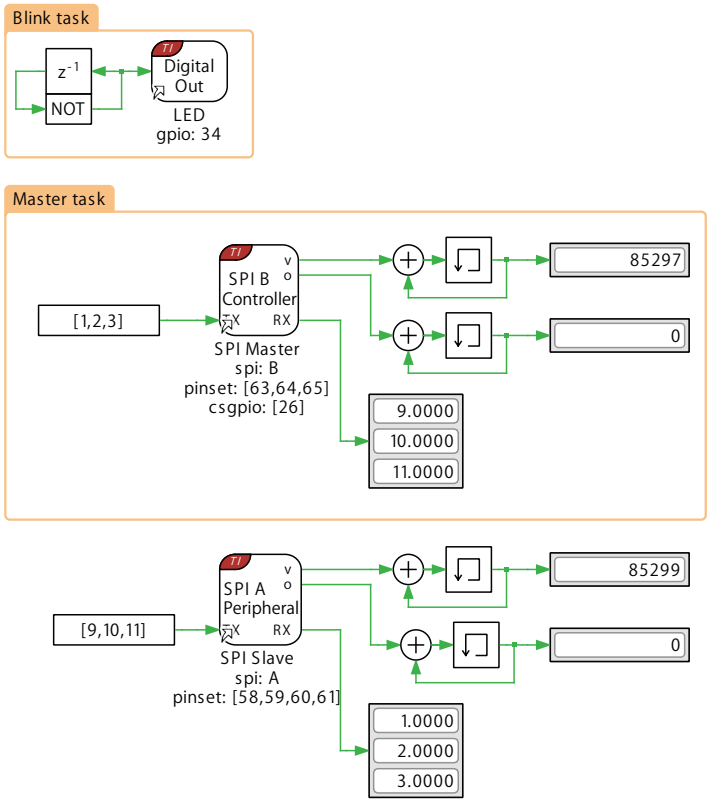
- **SPICLK** SPI Controllerによって生成されたシフトクロック
- **/SPICS** チップセレクトまたはペリフェラル-イネーブル信号。SPIPTEとも呼ばれます。チップセレクト信号は、SPI PeripheralのPOCIおよびPICOポートを有効にするアクティブLow信号です。

SPI Controllerブロックは、各シミュレーションステップ中に設定可能な数のクロックパルスを生成するクロック信号(SPICLK)を提供します。ペリフェラルとコントローラの両方で、データはSPICLKの一方のエッジ(立上りまたは立下り)でシフトレジスタからシフトアウトされ、反対側のクロックエッジでシフトレジスタにラッチされます。クロックフェーズ(CPHA)ビットが1に設定されている場合、データはSPICLK遷移の半サイクル前に送受信されます。

このモデルでは、各サブシステムにシンプルなSPIモデルが含まれており、図2に示すように、同じMCUターゲットの1つのSPI ControllerとSPI Peripheral間の信号交換を示しています。

チップセレクト(/CS)信号を介して、一つのコントローラで複数のSPI Peripheralをサポートできます。複数のSPI Peripheralの詳細については、SPI Controllerブロックの"ヘルプ"セクションを参照するか、TI C2000 Target Support User Manual[6]を参照してください。

図2: 28379Dとラベルの付いたサブシステムの回路図



SPI ControllerブロックとSPI Peripheralブロックの両方において、送信されるデータは入力**TX**で提供され、受信されたデータは出力**RX**で利用できます。

2.1 SPI Controllerに関する考慮事項

SPI Controllerの**v**ポートの出力値が1の場合、有効なデータがすべてのペリフェラルに送信されることを示します。ブロックが再度実行される前にSPI Controllerに送信を完了するのに十分な時間がない場合、出力**o**は1になり、オーバーランエラーを示します。図2に示すように、モデルでは**v**ポートと**o**ポートにカウンタロジックが組み込まれています。これはオフラインではあまり意味がありません。

SPI Controllerの**o**ポートでオーバーランエラーが通知されている場合は、SPI Controllerが関連付けられているタスクの実行が速すぎる可能性があります。この場合、SPI Controller実行タスクレートを下げるか、SPIクロックレートを上げてください。

たとえば、SPICLKが180000Hzに設定されていて、1ワードあたり8ビットで4ワードのパケットを送信すると予想される場合、1つのパケットの送信にかかる時間は次のようになります:

$$\frac{1}{180000} * 4 * 8 = 1.78 * 10^{-4} \text{ seconds}$$

この場合、SPI Controllerの実行ステップサイズは0.178ミリ秒より大きい値に設定する必要があります。

2.2 SPI Peripheralに関する考慮事項

vポートの出力値1は、SPI Controllerとの有効なデータ交換を示します。SPI PeripheralのSPI **RX**ポートが、前のデータが読み取られる前に新しいデータを受信すると、既存のデータは上書きされ、失われます。このような状況が発生すると、出力**o**が1になり、オーバーランエラーが示されます。図2に示すように、モデルでは**v**ポートと**o**ポートにカウンタロジックが組み込まれています。これはオフラインではあまり意味がありません。

オーバーランエラーが発生したときに注意すべき2つの考慮事項があります:

- コントローラは、ペリフェラルの起動前に送信を開始することは許されていません。ペリフェラルが起動中にコントローラが送信すると、ペリフェラルは最初のメッセージを不完全に受信する可能性があり、回復することはできません。
- オーバーランを回避するには、SPI Controllerがデータを送信する速度よりも速くSPI Peripheralブロックを実行する必要があります。

2.3 追加の考慮事項

- SPI経由で通信するすべてのMCUに必ず共通の電源を供給するようにしてください。
- RT Box LaunchPad Interface User Manual[7]によると、いくつかのSPI信号がデジタル信号に接続されており、SPI通信に干渉を引き起こす可能性があることがわかります。したがって、SPIを使用するには、LaunchpadボードをRT Box LaunchPad Interfaceボードから切断または分離する必要があります。
- RT Boxを使用して"CLK"信号を測定しないでください。オーバーランエラーが発生します。

2.4 マルチタスクコード

生成されたコードは、MCUの基本サンプル時間または**離散化ステップサイズ**で実行されます。このモデルでは、各サブシステムの離散化ステップサイズは100 μ sに設定されています。オーバーランを回避するために、SPI PeripheralブロックはSPI Controllerのデータ送信速度よりも高速に実行する必要があります。したがって、マルチタスク コードを使用して、SPI Controllerブロックを200 μ sで実行します。

マルチタスクコード生成は、**Coder** -> **Coderオプション...** ダイアログの**タスク**タブから構成します。**タスクモード**をマルチタスクに変更し、**タスク設定**を指定することにより、各タスクのサンプリング時間を設定できます。基本サンプル時間は常に**離散化ステップサイズ**と同じになります。優先度の低いタスクの**サンプリング時間**設定は、基本サンプリング時間の整数倍である必要があります。異なる速度で実行される、より低速で優先度の低いタスクを最大15個指定できるため、アプリケーション内で最も高速で優先度の高いタスクのプロセッサリソースが確保されます。詳細については、PLECS User Manual [8]の"コード生成"の章を参照してください。

3 シミュレーション

各サブシステムは、対応するTI LaunchPadハードウェアのターゲット固有のコードに直接変換できます。

注意 続行する前に、LaunchPadデバイスのDIPスイッチの位置とジャンパ構成が正しく設定されていることを確認してください。各LaunchPadデバイスのガイダンスは、TI C2000 Target Support User Manual [6]の"C2000開発キットのプログラミングに関するヒント"セクションに記載されています。

3.1 ハードウェアの接続

続いて、目的のMCUのジャンパ線を使用して、以下にリストされているピン番号を接続します。2つの異なるMCUターゲット間のSPI経由の信号交換を調べるには、任意の2つのターゲットを選択します。次に、一方をSPI Controllerとして設定し、もう一方を SPI Peripheralとして設定します。

28069	Controller	Peripheral
SIMO	J6-55 (GPIO 24)	J2-15 (GPIO 16)
SOMI	J6-54 (GPIO 25)	J2-14 (GPIO 17)
CLK	J5-47 (GPIO 14)	J1-7 (GPIO 18)
CS	J6-53 (GPIO 52)	J2-19 (GPIO 19)

280049	Controller	Peripheral
SIMO	J6-55 (GPIO 24)	J2-15 (GPIO 16)
SOMI	J6-54 (GPIO 31)	J2-14 (GPIO 17)
CLK	J5-47 (GPIO 22)	J1-7 (GPIO 56)
CS	J6-59 (GPIO 27)	J2-19 (GPIO 57)

28377S	Controller	Peripheral
SIMO	J6-55 (GPIO 63)	J2-15 (GPIO 58)
SOMI	J6-54 (GPIO 64)	J2-14 (GPIO 59)
CLK	J5-47 (GPIO 65)	J1-7 (GPIO 60)
CS	J6-53 (GPIO 99)	J1-8 (GPIO 61)

28379D	Controller	Peripheral
SIMO	J6-55 (GPIO 63)	J2-15 (GPIO 58)
SOMI	J6-54 (GPIO 64)	J2-14 (GPIO 59)
CLK	J5-47 (GPIO 65)	J1-7 (GPIO 60)
CS	J6-53 (GPIO 26)	J2-19 (GPIO 61)

28P650DK9	Controller	Peripheral
SIMO	J6-55 (GPIO 91)	J2-15 (GPIO 16)
SOMI	J6-54 (GPIO 92)	J2-14 (GPIO 17)
CLK	J8-74 (GPIO 32)	J1-7 (GPIO 18)
CS	J6-59 (GPIO 94)	J2-19 (GPIO 57)

3.2 MCUをフラッシュ

サブシステムをTI MCUにアップロードするには、以下の手順に従います:

- 目的のMCUをUSBケーブルを介してホストコンピュータに接続します。
- **Coder** -> **Coderオプション...**ウィンドウの**システム**リストから、目的のMCUを選択します。
- **ターゲット**タブのドロップダウンメニューから適切なターゲットを選択します。次に、**General**サブタブで、必要な**Build type**を選択します。
- 次に、**Build type**でBuild andを選択し、**Build configuration**でRun from FlashまたはRun from RAMのいずれかを選択し、**Board type**でLaunchPadを選択して、**ビルド**をクリックします。

注意 正しくプログラムされていれば、LaunchPadボード上のLEDが点滅します。

Code Composer Studio(CCS)に精通している上級ユーザ向けには、`Generate code into CCS project`オプションがあります。TI C2000 Target Supportパッケージには、`projects`というタイトルのフォルダが含まれています。フォルダ内には、各MCU用に事前に構築されたCCSプロジェクトを含むZIPアーカイブが含まれています。目的のターゲットに対応するzipアーカイブフォルダをCCSにインポートします。CCSワークスペースに新しいプロジェクトが作成されます。CCSプロジェクトの`${workspace_loc}/dev_28xx/cg/`フォルダの場所を**CCS project directory**フィールドに入り、**ビルド**をクリックします。その後、通常のCCSプロジェクトと同様にプロジェクトのビルドとデバッグに進みます。詳細な手順については、TI C2000 Target Support User Manual[6]の"クイックスタート"セクションを参照してください。

3.3 外部モード

生成されたコードがC2000ターゲット上で実行されると、ユーザは外部モードに入り、PLECSスコープのリアルタイム波形の更新を確認しながら、特定のシミュレーションパラメータを変更できます。以下の手順はターゲットデバイスへの接続方法の概要を示しており、さらなるデバッグの詳細についてはユーザマニュアル[6]の"外部モードの開始"セクションに記載されています。

- まず、**Coder** -> **Coderオプション...**ウィンドウの左側にある**システム**のリストから、目的のMCUを選択します。
- 次に、**外部モード**タブで、**ターゲットデバイス**フィールドの横にあるアイコンをクリックして、**ターゲットデバイス**を選択します。
- 最後に、**接続**をクリックし、自動**トリガを有効化**をクリックして、サブシステムのPLECSスコープでテスト結果を確認します。

28379Dターゲット上で交換されたSPI信号を図2に示します。**v**ポートと**o**ポートの詳細、およびオーバーランエラーを処理するためのヒントについては、第2章を参照してください。

3.4 パラメータのインライン化

パラメータを調整可能として構成するには、**Coder**メニュー -> **Coderオプション...**を開き、**パラメータのインライン化**タブに移動します。回路図のコンポーネントを**例外**リストにドラッグアンドドロップすると、そのコンポーネントに関連付けられたパラメータが実行時に調整可能になります。この動作は**デフォルト設定**に依存することに注意してください。例外リストには、デフォルト設定とは逆の動作をするコンポーネントを指定します。

この場合、SPIブロックへの"TX"入力は、モデルを外部モードに接続し、組み込みターゲットデバイス上で実行中にオンザフライで調整できます。パラメータの変更は、有効になるとすぐにPLECSスコープと数値表示に反映されます。

4 まとめ

このモデルでは、単純なモデルを使用してTI C2000 TSPのSPIプロトコルを検証しました。

5 参考文献

[1] TI C2000 Piccolo MCU F280049C LaunchPad Development Kit
URL: http://www.ti.com/tool/LAUNCHXL-F280049C

[2] TI C2000 Piccolo MCU F28069M LaunchPad Development Kit
URL: http://www.ti.com/tool/LAUNCHXL-F28069M

[3] TI C2000 Delfino MCUs F28377S LaunchPad Development Kit
URL: https://www.ti.com/lit/pdf/sprui25

[4] TI C2000 Delfino MCUs F28379D LaunchPad Development Kit
URL: http://www.ti.com/tool/LAUNCHXL-F28379D

[5] TI C2000 F28P650DK9 LaunchPad development kit
URL: https://www.ti.com/tool/LAUNCHXL-F28P65X

[6] PLECS TI C2000 Target Support User Manual
URL: https://www.plexim.com/sites/default/files/c2000manual.pdf
日本語マニュアル: https://adv-auto.co.jp/products/plexim/manual.html

[7] RT Box LaunchPad Interface User Manual
URL: https://plexim.com/sites/default/files/launchpadinterfacemanual.pdf
日本語マニュアル: https://adv-auto.co.jp/products/plexim/manual.html

[8] PLECS User Manual
URL: https://www.plexim.com/sites/default/files/plecsmanual.pdf
日本語マニュアル: https://adv-auto.co.jp/products/plexim/manual.html

改訂履歴:

C2000 TSP 1.3.1	初版
C2000 TSP 1.4.5	Webのリンクを更新
C2000 TSP 1.5.2	SPIモード設定を更新

Plexim

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

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 Demo Model