

本資料は英語版を翻訳した参考資料です。本資料の第 6 章までは Renesas Synergy SSP v1.5.0 Users Manual 英語版の第 4 章 Module Guide を参照に翻訳されています。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

## 要旨 (Introduction)

本モジュールガイドは、ユーザがモジュールを効果的に使用してシステムが開発できるようになることを目的としています。このモジュールガイドを習得することで、開発システムへのモジュールの追加とターゲットアプリケーション向けの正確な設定 (configuration) ができ、さらに付属のアプリケーションプロジェクトコードを参照して、効率的なコード記述が行えるようになります。

より詳細な API や、より高度なモジュール使用法を記述した他のアプリケーションプロジェクト例もルネサス WEB サイト (本書末尾の「参考文献」の項を参照) から入手でき、より複雑な設計に役立ちます。

SPI フレームワークモジュールは ThreadX® RTOS 対応のフレームワーク API であり、sf\_spi に実装されています。SPI バス上の複数の SPI ペリフェラルの統合と同期を処理し、チップ選択処理とレベルアクティベーションを行います。SPI フレームワークによって、1 つまたは複数の SPI バスの作成と、複数の SPI ペリフェラルの SPI バスへの接続ができるようになります。SPI フレームワークモジュールは、SCI SPI と RSPI ドライバの両方にアクセスするために単一のインタフェースを使用します。このため SPI フレームワークモジュールは、Synergy MCU の SCI および RSPI ペリフェラルを使用します。

## 目次 (Contents)

1. SPI フレームワークモジュールの機能 (SPI Framework Module Features) .....	3
2. SPI フレームワークモジュール API の概要 (SPI Framework Module APIs Overview) .....	4
3. SPI フレームワークモジュールの動作の概要 (SPI Framework Module Operational Overview) .....	6
3.1 SPI フレームワークモジュールの動作に関する重要な注意事項と制限事項 (SPI Framework Module Important Operational Notes and Limitations) .....	6
3.1.1 SPI フレームワークモジュールの動作に関する注意事項 (SPI Framework Module Operational Notes) 6	
3.1.2 SPI フレームワークモジュールの制限事項 (SPI Framework Module Limitations) .....	6
4. アプリケーションへの SPI フレームワークモジュールの組み込み (Including the SPI Framework Module in an Application) .....	7
5. SPI フレームワークモジュールの設定 (Configuring the SPI Framework Module) .....	8
5.1 ローレベルモジュールの SPI フレームワークモジュールの設定 (SPI Framework Module Configuration Settings for Lower-Level Modules) .....	9
5.1.1 r_rsipi 選択時 (When r_rsipi is selected) .....	9
5.1.2 r_sci_spi 選択時 (When r_sci_spi is selected) .....	13
5.2 SPI フレームワークモジュールのクロック構成 (SPI Framework Module Clock Configuration) .....	16

---

5.3	SPI フレームワークモジュールのピン構成 (SPI Framework Module Pin Configuration) .....	16
5.4	SPI 通信フレームワークモジュールの追加の設定 (SPI Framework Module Additional Settings) ...	17
6.	アプリケーションでの SPI フレームワークモジュールの使用 (Using the SPI Framework Module in an Application) .....	17
7.	SPI フレームワークモジュールのアプリケーションプロジェクト (The SPI Framework Module Application Project) .....	24
8.	対象アプリケーションの SPI フレームワークモジュールのカスタマイズ (Customizing the SPI Framework Module for a Target Application) .....	28
9.	SPI フレームワークモジュールのアプリケーションプロジェクトの実行 (Running the SPI Framework Module Application Project) .....	29
10.	SPI フレームワークモジュールのまとめ (SPI Framework Module Conclusion) .....	30
11.	SPI フレームワークモジュールの次の手順 (SPI Framework Module Next Steps) .....	30
12.	SPI フレームワークモジュールの参考情報 (SPI Framework Module Reference Information)	30

### 1. SPI フレームワークモジュールの機能 (SPI Framework Module Features)

SPI フレームワークモジュールは、SPI モードの SCI (SCI 共通ローレベルモジュールとともに)、または RSPI ローレベルドライバモジュールを使用して、Synergy マイクロコントローラ上の SPI 周辺回路と通信を行います。

- バス上の複数のデバイスをサポート
- モジュールの初期化、転送、クローズ用のハイレベル API を提供
- 同期転送をサポート
- チップ選択操作をサポート
- バスロックをサポート

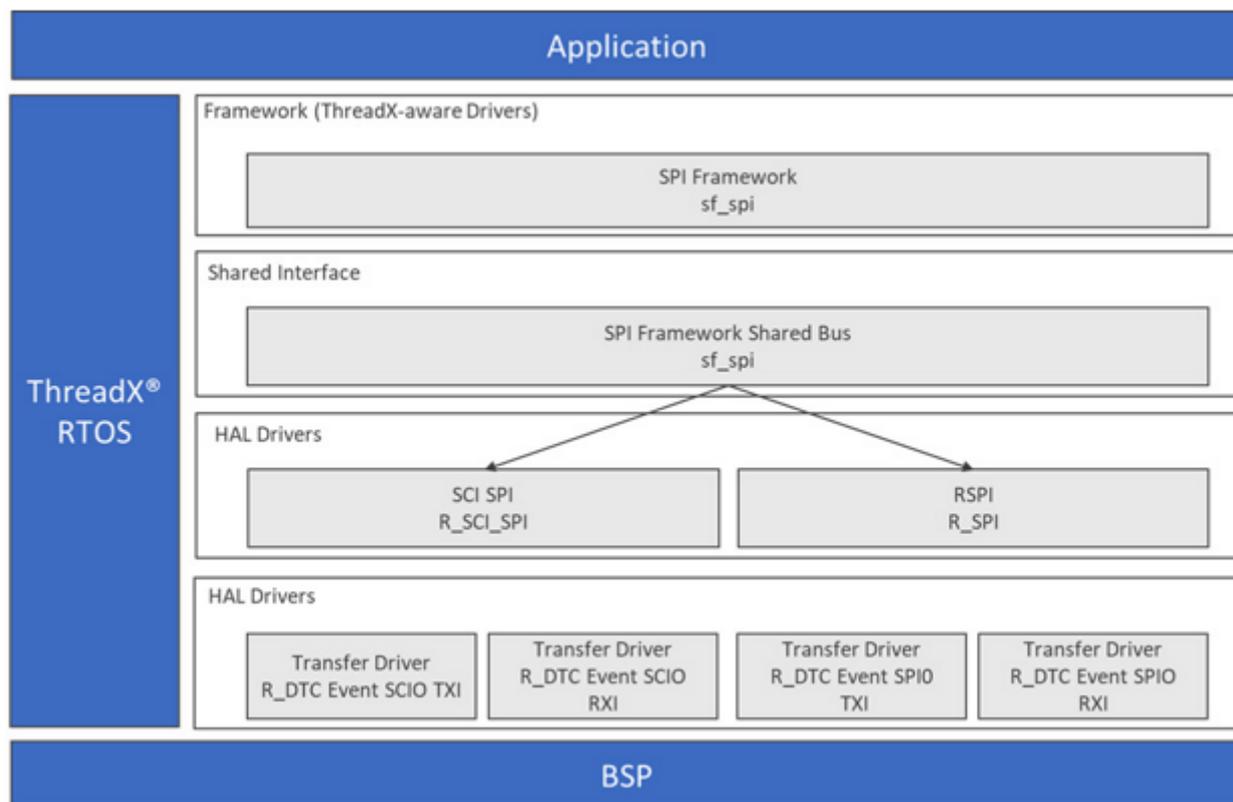


図 1 SPI フレームワークモジュールのブロック図

## 2. SPI フレームワークモジュール API の概要 (SPI Framework Module APIs Overview)

SPI フレームワークモジュールは、open、close、read、write、その他の API 機能を定義します。使用可能なすべての API のリスト、API コールの例、各 API の簡単な説明を、次の表に示します。ステータス戻り値 (status returns value) の表は API 要約表の後にあります。

表 1 SPI フレームワークモジュール API の要約

関数名	API 呼び出しの例と説明
.open	<pre>g_sf_spi_device.p_api-&gt;open(g_sf_spi_device.p_ctrl, g_sf_spi_device.p_cfg);</pre> <p>指定された SPI デバイスをバス上で開きます。</p>
.read	<pre>g_sf_spi_device.p_api-&gt;read(g_sf_spi_device.p_ctrl, dst8, length, SPI_BIT_WIDTH_8_BITS, TX_WAIT_FOREVER);</pre> <p>SPI デバイスからデータを受信します。</p>
.write	<pre>g_sf_spi_device.p_api-&gt;write(g_sf_spi_device.p_ctrl, src8, length, SPI_BIT_WIDTH_8_BITS, TX_WAIT_FOREVER);</pre> <p>SPI デバイスにデータを送信します。</p>
.writeRead	<pre>g_sf_spi_device.p_api-&gt;writeRead (g_sf_spi_device.p_ctrl, &amp;source, &amp;destination, length, SPI_BIT_WIDTH_8_BITS, TX_WAIT_FOREVER);</pre> <p>SPI デバイスとの間で、データの送信と受信を同時に実行します (全二重通信)。この読み書き API (writeRead API) は、ミューテックスオブジェクト (mutex object) を取得し、SPI HAL レイヤで SPI データ送信を処理するとともに、SPI HAL レイヤからデータを受信します。またこの API は、イベントフラグ待ちを使用して、データ転送の完了と同期します。</p>
.close	<pre>g_sf_spi_device.p_api-&gt;close(g_sf_spi_device.p_ctrl);</pre> <p>制御ハンドルで指定された SPI デバイスを無効化します。またバスにデバイスが接続されていない場合は、バスで使用される RTOS サービスを終了します。この関数は、ハンドルで指定された SPI チャンネルへの電源を遮断し、関連付けられた割り込みを無効にします。</p>
.lock	<pre>g_sf_spi_device.p_api-&gt;lock(g_sf_spi_device.p_ctrl);</pre> <p>バスをデバイスにロックします。ロックすると、一定時間 (ロックとロック解除の間など)、デバイスがバスを予約できるようになります。これにより、デバイスがバスに対する複数のリードとライトを割り込みなしで完了することができます。</p>
.unlock	<pre>g_sf_spi_device.p_api-&gt;unlock(g_sf_spi_device.p_ctrl);</pre> <p>特定のデバイスのバスをロック解除し、他のデバイスがそのバスを使用できるようにします。</p>
.version	<pre>g_sf_spi_device.p_api-&gt;version(&amp;version);</pre> <p>バージョンポインタ (version pointer) を使用して、API バージョンを取得します。</p>

注： 関数データ構造体、型定義、定義、API データ、API 構造体、関数変数の動作と定義の詳細については、『SSP ユーザーズマニュアル』で関連モジュールの API リファレンスを参照してください。

表 2 ステータス戻り値

Name	説明
SSP_SUCCESS	関数が正常に完了しました
SSP_ERR_INVALID_MODE	無効なモード
SSP_ERR_INVALID_CHANNEL	無効なチャンネル
SSP_ERR_IN_USE	使用中エラー
SSP_ERR_INVALID_ARGUMENT	無効な引数
SSP_ERR_QUEUE_UNAVAILABLE	キューが使用不能
SSP_ERR_INVALID_POINTER	無効なポインタ
SSP_ERR_INTERNAL	内部エラー
SSP_ERR_TRANSFER_ABORTED	転送中断
SSP_ERR_MODE_FAULT	モードフォールト（障害）
SSP_ERR_READ_OVERFLOW	リードオーバーフロー
SSP_ERR_PARITY	パリティエラー
SSP_ERR_OVERRUN	オーバーランエラー
SSP_ERR_UNDEF	不明なエラー
SSP_ERR_TIMEOUT	タイムアウトエラー
SSP_ERR_NOT_OPEN	デバイスが開かれていない

注： ローレベルドライバ（Lower-level drivers）は、一般的なエラーコード（common error codes）を返すことがあります。すべての関連ステータス戻り値の定義については、『SSP ユーザーズマニュアル』で関連モジュールの API リファレンスを参照してください。

### 3. SPI フレームワークモジュールの動作の概要 (SPI Framework Module Operational Overview)

SPI フレームワークモジュールは、SSP の階層化ドライバアーキテクチャに準拠しています。SPI モジュールの SCI もしくは RSPI モジュールを使用して、Synergy マイクロコントローラに接続された SPI 周辺デバイスと通信をおこないます。

#### 同一バス上の複数のスレーブデバイス (Multiple Slave Devices on the Same bus)

SPI フレームワークドライバアーキテクチャは、「バスと、そのバス上のデバイス」アーキテクチャを採用しています。ローレベルドライバでは一度に1つのデバイスのみが設定され (configured)、残りのデバイスは必要に応じてリードまたはライト動作時に再設定 (reconfigured) されます。ローレベルドライバは、バスがロックされていないときのみ再設定できます。全てのスレーブデバイスは接続先のバスにリンクされ、他の全てのスレーブデバイスとバスを共有します。

ユーザは SPI フレームワークモジュールを共有バスとして設定し、各 SPI フレームワークモジュール向けローレベル SPI HAL レイヤをそのバスに接続する設定にします。ユーザは、複数のデバイスを同じバス上で設定するときに、既存のフレームワーク共有バスを追加することができます。各 SPI フレームワークは、ISDE コンフィギュレータで一意の名前を使用して設定する必要があります。

共通の開始および停止手順が、すべての SPI データ転送動作 (リード、ライト、読み書き) に使用されます。開始プロセスの際に、フレームワークモジュールは再設定が必要かどうかを確認します。バスがロックされていない場合、チップ選択は、転送開始処理の中でアサートされ、転送終了処理の中でアサート解除されます。ユーザは、チップ選択 IO ピンとチップ選択アクティブレベルを設定する必要があります。

#### バスのロック (Bus Locking)

SPI フレームワークモジュールではバスロック機能がサポートされており、ユーザは特定のスレーブ周辺デバイスのバスをロックできます。バスがロックされている場合、一定時間 (ロックとロック解除の間) スレーブデバイスはバスを予約できるようになります。これにより、デバイスはバス上で、(ある条件下で) 複数の読み書き操作を中断することなく実行できるようになります。ロックしている間のチップ選択はアクティブになり、ロック解除するとチップ選択は非アクティブになります。ロックとロック解除の間にライトとリードを行っても、チップ選択ライン (chip select line) は変更されません。

### 3.1 SPI フレームワークモジュールの動作に関する重要な注意事項と制限事項 (SPI Framework Module Important Operational Notes and Limitations)

#### 3.1.1 SPI フレームワークモジュールの動作に関する注意事項 (SPI Framework Module Operational Notes)

- 複数の SPI デバイスが共通バスを共有するように設定できます。一度 SPI フレームワークバスモジュールを設定したら、各種の SPI 周辺デバイスをそのバスに接続できます。
- バスに接続している SPI デバイスごとに、SPI HAL モジュール (新規もしくは共有) と SPI フレームワークデバイスモジュールを追加する必要があります。
- フレームワークが内部的に対応するため、ユーザ定義コールバックは不要です。
- 割り込みを異なる優先度に設定すると、適切に動作しなくなる可能性があります。

#### 3.1.2 SPI フレームワークモジュールの制限事項 (SPI Framework Module Limitations)

- SPI バスの互換性については、MCU マニュアルを参照してください。SPI バスとのデバイスの互換性はフレームワークでチェックされません。このため、互換性のない SPI デバイスによって不適切な動作が発生することがあります。
- このモジュールの動作に関するその他の制限事項については、最新の SSP リリースノートを参照してください。

#### 4. アプリケーションへの SPI フレームワークモジュールの組み込み (Including the SPI Framework Module in an Application)

この章では、SSP コンフィギュレータを使用して SPI フレームワークモジュールをアプリケーションに組み込む方法について説明します。

注： この章を読むには、プロジェクトの作成、スレッドの追加、スレッドへのスタックの追加、およびスタック内でのブロックの構成について理解している必要があります。これらの項目に精通していない場合は、『SSP ユーザーズマニュアル』の最初のいくつかの章を参照して、SSP ベースのアプリケーション作成時の重要な各手順の取り扱い方を修得してください。

SPI フレームワークモジュールをアプリケーションに追加するには、次の表に示すスタック選択シーケンスを使用してスレッドに追加します。(SPI フレームワークのデフォルト名は `g_sf_spi_device0` です。この名前は対応する [Properties] ウィンドウで変更できます。)

表 3 SPI フレームワークモジュールの選択シーケンス

リソース	ISDE Tab	Stacks Selection Sequence
<code>g_sf_spi_device0</code> on <code>sf_spi</code>	Threads	New Stack> Framework> Connectivity > SPI Framework Device on <code>sf_spi</code>

次の図に示すように、`sf_spi` の SPI フレームワークモジュールがスレッドスタックに追加されると、コンフィギュレータは必要なローレベルモジュールを自動的に追加します。追加の設定情報を必要とするドライバは、ボックスのテキストが赤くハイライト表示されます。グレーの帯が表示されているモジュールは個別のスタンドアロンモジュールです。青い帯で示されるモジュールは共有または共通で、一度追加すれば複数のスタックで使用できます。ピンクの帯で示されるモジュールには、ローレベルドライバの選択が必要です。これらはオプションの場合と推奨の場合があります (ブロック内でテキストによって示されます)。ローレベルモジュールの追加が必要な場合は、モジュールの説明に **Add** というテキストが含まれます。ピンクの帯で示されるモジュールをクリックすると、[New] アイコンが表示され、可能な選択肢が表示されます。

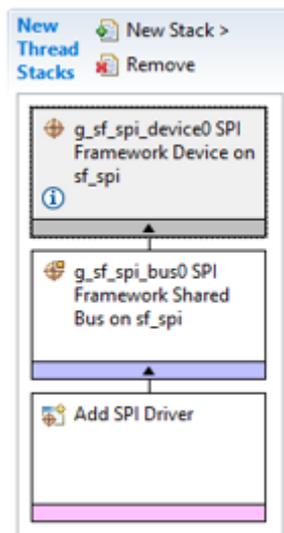


図 2 SPI フレームワークモジュールのスタック

## 5. SPI フレームワークモジュールの設定 (Configuring the SPI Framework Module)

SPI フレームワークモジュールはシステム要求に合わせた設定 (configured) が必要です。[SSP configuration] ウィンドウでは、割り込みや動作モードなどの必須の設定選択項目が自動的に識別されます (ブロックが赤でハイライト表示される)。これらは、ローレベルモジュールが正常に作動するために設定される必要があります。変更しても競合が発生しないプロパティのみが変更可能ですが、その他のプロパティは「ロック」されていて変更できません。これらは ISDE 内の [Properties] ウィンドウで「ロック」されているプロパティとして、ロックアイコンで識別されます。この方法により構成プロセスが簡略化され、従来の手動による構成アプローチに比べて大幅にエラーが発生しにくくなっています。ユーザがアクセスできるすべてのプロパティの使用可能な構成設定とデフォルトは、SSP コンフィギュレータ内の [Properties] タブにあります。この内容を下表に示します。

注： 次の表に示す構成テーブルの設定を参照しながら、ISDE を開き、モジュールを作成し、プロパティ設定を調べることができます。SSP を使用した開発の参考になります。

表 4 sf\_spi 上の SPI フレームワークモジュールの設定

Parameter	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled  Default: Default (BSP)	パラメータチェックのコードをビルドに含めるかどうかを選択します。
Name	Default: g_sf_spi_device0	モジュール名
Clock Phase	Data sampling on odd edge, data variation on even edge/ Data sampling on even edge, data variation on odd edge  Default: Data sampling on odd edge, data variation on even edge	クロックフェーズを選択します。
Clock Polarity	Low when idle, High when idle  Default: Low when idle	クロックの極性を選択します。
Chip Select Port	00~11 Default: 00	チップ選択に使用される GPIO ポートを選択します。
Chip Select Pin	00~15 Default: 00	チップ選択に使用される GPIO ピンを選択します。
Chip Select Active Level	Low, High  Default: Low	チップ選択信号の極性で、アクティブハイまたはローです。

注： 設定例とデフォルトは、Synergy S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

ある場合には、デフォルト以外の値にしたい場合もあります。たとえば、異なるチップセレクト GPIO やレベルの選択を行う場合です。ローレベルスタックモジュールで設定可能なプロパティについては、すべての後の章に記載されています。

注： ローレベルモジュールのプロパティ設定の大半は、通常は SSP コンフィギュレータから対応する [Properties] ウィンドウを調べることで決定できます。

## 5.1 ローレベルモジュールの SPI フレームワークモジュールの設定 (SPI Framework Module Configuration Settings for Lower-Level Modules)

通常、ローレベルドライバのデフォルト設定から変更する必要がある設定項目は少なく、これらはスレッドスタックブロックに赤いテキストで表示されます。一部の設定プロパティは、フレームワークが適切に作動するために特定の値に設定されており、それらはロックされていてユーザは変更できません。次の表に、ローレベルモジュールの [Properties] セクションのすべての設定を示します。

表 5 sf\_spi 上の SPI フレームワーク共有バスの設定

ISDE のプロパティ	Value	説明
Name	g_sf_spi_bus0	モジュール名

注： 設定例とデフォルトは、Synergy S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

### 5.1.1 r\_rsipi 選択時 (When r\_rsipi is selected)

表 6 r\_rsipi 上の RSPI HAL ドライバの設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled Default: Default (BSP)	パラメータチェック用に選択されたコードがビルドに含まれている場合。
Name	Default: g_spi0	モジュール名
Channel	Default: 0	SCI または SPI デバイスが接続されているチャンネル番号。
Operating Mode	Master, Slave Default: Master	マスタまたはスレーブデバイスとして設定します。  注：現在のバージョンの SSP は、SPI マスターモードのみをサポートしています。
Clock Phase	(Locked) Data sampling on odd edge, data variation on even edge	奇数または偶数クロックエッジでのデータサンプリング。(ロックされています)
Clock Polarity	(Locked) Low when idle	アイドル時のクロックレベル。(ロックされています)
Mode Fault Error	Enable, Disable Default: Disable	モード障害エラー (マスタ/スレーブ競合) フラグを示します。
Bit Order	MSB First, LSB First Default: MSB First	送信順序 MSB / LSB を最初に選択します。
Bitrate	Default: 500000	送信または受信レート。 単位：ビット/秒
Callback	(Locked) NULL	オプションのコールバック関数ポインタ。(ロックされています)
SPI Mode	SPI Operation, Clock Synchronous operation Default: SPI Operation	SPI または Clock Syn モード操作を選択します。
Slave Select Polarity (SSL0)	Active Low, Active High Default: Active Low	SSL0 信号の極性を選択します。

ISDE のプロパティ	Value	説明
Slave Select Polarity (SSL1)	Active Low, Active High Default: Active Low	SSL1 信号の極性を選択します。
Slave Select Polarity (SSL2)	Active Low, Active High Default: Active Low	SSL2 信号の極性を選択します。
Slave Select Polarity (SSL3)	Active Low, Active High Default: Active Low	SSL3 信号の極性を選択します。
Select Loopback 1	Normal, Not Inverted Default: Normal	ループバック 1 のデータモードを選択します。
Select Loopback 2	Normal, Not Inverted Default: Normal	ループバック 2 のデータモードを選択します。
Enable MOSI Idle State	Enable, Disable Default: Disable	MOSI アイドル状態の有効/無効の選択
MOSI Idle State	MOSI Low, MOSI High Default: MOSI Low	MOSI アイドル状態を選択
Enable Parity	Enable, Disable Default: Disable	パリティを有効/無効にする
Parity Mode	Parity Odd, Parity Even Default: Parity Odd	パリティモードの選択
Select SSL (Slave Select)	SSL0, SSL1, SSL2, SSL3 Default: SSL0	使用するスレーブを選択します。 0-SSL0; 1-SSL1; 2-SSL2; 3-SSL3
Select SSL Level After Transfer	SSL Level Keep, SSL Level Do Not Keep Default: SSL Level Do Not Keep	転送完了後に SSL レベルを選択します。 0 ネゲート; 1 キープ
Clock Delay Enable	Clock Delay Enable, Clock Delay Disable Default: Clock Delay Disable	クロック遅延有効化の選択
Clock Delay Count	Clock Delay 1~8 RSPCK Default: Clock Delay 1 RSPCK	クロック遅延カウンターの選択
SSL Negation Delay Enable	Negation Delay Enable, Negation Delay Disable Default: Negation Delay Disable	SSL ネゲーション遅延有効化の選択
Negation Delay Count	Negation Delay 1~8 RSPCK Default: Negation Delay 1 RSPCK	ネゲーション遅延カウンターの選択
Next Access Delay Enable	Next Access Delay Enable, Next Access Delay Disable Default: Next Access Delay Disable	次アクセス遅延有効化の選択

ISDE のプロパティ	Value	説明
Next Access Delay Count	Next Access Delay 1~8 RSPCK  Default: Next Access Delay 1 RSPCK	次アクセス遅延カウンタの選択
Receive Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X)  Default: Priority 12	受信割り込み優先順位の選択。
Transmit Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X)  Default: Priority 12	送信割り込み優先順位の選択。
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X)  Default: Priority 12	送信終了割り込み優先順位の選択。
Error Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X)  Default: Priority 12	エラー割り込み優先順位の選択。

注： 設定例とデフォルトは、Synergy SK-S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

表 7 r\_dtc Event SPI0 TXI 上の Transfer Driver の設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled  Default: Default (BSP)	パラメータチェック用のコードをビルドに含めるかどうかを選択
Software Start	Enabled, Disabled  Default: Disabled	ソフトウェアスタートの選択
Link section to keep DTC vector table	Default: .ssp_dtc_vector_table	DTC ベクタテーブルを保持するリンクセクションの選択
Name	Default: g_transfer0	モジュール名
Mode	(Locked) Normal	モードの選択 (ロックされています)
Transfer Size	1 Bytes, 2 Bytes, 4 Bytes  Default: 2 Bytes	転送サイズの選択
Destination Address Mode	(Locked) Fixed	宛先アドレスモードの選択 (ロックされています)
Source Address Mode	(Locked) Incremented	ソースアドレスモードの選択 (ロックされています)

ISDE のプロパティ	Value	説明
Repeat Area (Unused in Normal Mode)	(Locked) Source	リピートエリアの選択 (ロックされています)
Interrupt Frequency	(Locked) After all transfers have completed	割り込み頻度の選択 (ロックされています)
Destination Pointer	(Locked) NULL	宛先ポインタの選択 (ロックされています)
Source Pointer	(Locked) NULL	ソースポインタの選択 (ロックされています)
Number of Transfers	(Locked) 0	転送回数の選択 (ロックされています)
Number of Blocks (Valid only in Block Mode)	(Locked) 0	ブロック数の選択 (ロックされています)
Activation Source (Must enable IRQ)	(Locked) Event SPI0 TXI	アクティベーションソースの選択 (ロックされています)
Auto Enable	(Locked) False	自動有効の選択 (ロックされています)
Callback (Only valid with Software start)	(Locked) NULL	コールバックの選択 (ロックされています)
ECL Software Event Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X), Disabled  Default: Disabled	ELC ソフトウェアイベント割り込み優先順位選択

注： 設定例とデフォルトは、Synergy S7 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

表 8 r\_dtc Event SPI0 RXI の Transfer Driver の設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default(BSP), Enabled, Disabled  Default: Default (BSP)	パラメータチェック用のコードをビルドに含めるかどうかを選択
Software Start	Enabled, Disabled  Default: Disabled	ソフトウェアスタートの選択
Link section to keep DTC vector table	Default: .ssp_dtc_vector_table	DTC ベクタテーブルを保持するリンクセクションの選択
Name	Default: g_transfer1	モジュール名
Mode	(Locked) Normal	モードの選択 (ロックされています)
Transfer Size	1, 2, 4 Bytes  Default: 2 Bytes	転送サイズの選択
Destination Address Mode	(Locked) Incremented	宛先アドレスモードの選択 (ロックされています)
Source Address Mode	(Locked) Fixed	ソースアドレスモードの選択 (ロックされています)
Repeat Area (Unused in Normal Mode)	(Locked) Destination	リピートエリアの選択 (ロックされています)
Interrupt Frequency	(Locked) After all transfers have completed	割り込み頻度の選択 (ロックされています)

ISDE のプロパティ	Value	説明
Destination Pointer	(Locked) NULL	宛先ポインタの選択 (ロックされています)
Source Pointer	(Locked) NULL	ソースポインタの選択 (ロックされています)
Number of Transfers	(Locked) 0	転送回数の選択 (ロックされています)
Number of Blocks (Valid only in Block Mode)	(Locked) 0	ブロック数の選択 (ロックされています)
Activation Source (Must enable IRQ)	(Locked) Event SPI0 RXI	アクティベーションソースの選択 (ロックされています)
Auto Enable	(Locked) False	自動有効の選択 (ロックされています)
Callback (Only valid with Software start)	(Locked) NULL	コールバックの選択 (ロックされています)
ECL Software Event Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X), Disabled  Default: Disabled	ELC ソフトウェアイベント割り込み優先 順位選択

注： 設定例とデフォルトは、Synergy S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

### 5.1.2 r\_sci\_spi 選択時 (When r\_sci\_spi is selected)

表 9 r\_sci\_spi 上の SPI ドライバの設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled  Default: Default (BSP)	パラメータのエラーチェックを有効または無効にします。
Name	Default: g_spi0	モジュール名
Channel	Default: 0	SCI または SPI デバイスが接続されているチャンネル番号。
Operating Mode	Master, Slave  Default: Master	マスタまたはスレーブデバイスとして設定します。 注：現在のバージョンの SSP は、SPI マスターモードのみをサポートしています。
Clock Phase	(Locked) Data sampling on odd edge, data variation on even edge	奇数または偶数クロックエッジでのデータサンプリング。(ロックされています)
Clock Polarity	(Locked) Low when idle	アイドル時のクロックレベル (ロックされています)
Mode Fault Error	Enable, Disable  Default: Disable	モード障害エラー (マスタ/スレーブ競合) フラグを示します。
Bit Order	MSB First, LSB First  Default: MSB First	送信順序 MSB / LSB を最初に選択してください。
Bitrate	Default: 100000	送信または受信レート。 単位：ビット/秒

ISDE のプロパティ	Value	説明
Bitrate Modulation Enable	Enable, Disable Default: Enable	ビットレートモジュレーション機能のイネーブルまたはディセーブル。
Callback	(Locked) NULL	オプションコールバック関数ポインタ。(ロックされています)
Receive Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X) Default: Priority 12	ビットレート変調機能の有効/無効を切り替えます。  注：これは SCI SPI にのみ適用されません。
Transmit Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X) Default: Priority 12	送信割り込み優先順位の選択。
Transmit End Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X) Default: Priority 12	送信終了割り込み優先順位の選択。
Error Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X) Default: Priority 12	エラー割り込み優先順位の選択。

注： 設定例とデフォルトは、Synergy S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

表 10 r\_dtc Event SCI0 TXI の Transfer Driver の設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled Default: (Default) BSP	パラメータチェック用のコードをビルドに含めるかどうかを選択
Software Start	Enabled, Disabled Default: Disabled	ソフトウェアスタートの選択
Link section to keep DTC vector table	Default: .ssp_dtc_vector_table	DTC ベクタテーブルを保持するリンクセクションの選択
Name	Default: g_transfer0	モジュール名
Mode	(Locked) Normal	モードの選択 (ロックされています)
Transfer Size	(Locked) 1 Byte	転送サイズの選択 (ロックされています)
Destination Address Mode	(Locked) Fixed	宛先アドレスモードの選択 (ロックされています)

ISDE のプロパティ	Value	説明
Source Address Mode	(Locked) Incremented	ソースアドレスモードの選択 (ロックされています)
Repeat Area (Unused in Normal Mode)	(Locked) Source	リピートエリアの選択 (ロックされています)
Interrupt Frequency	(Locked) After all transfers have completed	割り込み頻度の選択 (ロックされています)
Destination Pointer	(Locked) NULL	宛先ポインタの選択 (ロックされています)
Source Pointer	(Locked) NULL	ソースポインタの選択 (ロックされています)
Number of Transfers	(Locked) 0	転送回数の選択 (ロックされています)
Number of Blocks (Valid only in Block Mode)	(Locked) 0	ブロック数の選択 (ロックされています)
Activation Source (Must enable IRQ)	(Locked) Event SCIO TXI	アクティベーションソースの選択 (ロックされています)
Auto Enable	(Locked) False	自動有効の選択 (ロックされています)
Callback (Only valid with Software start)	(Locked) NULL	コールバックの選択 (ロックされています)
ECL Software Event Interrupt Priority	Priority 0 (highest), Priority 1:14, Priority 15 (lowest, not valid if using Thread X), Disabled  Default: Disabled	ELC ソフトウェアイベント割り込み優先順位選択

注： 設定例とデフォルトは、Synergy S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

表 11 r\_dtc Event SCIO RXI の Transfer Driver の設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled  Default: Default (BSP)	パラメータチェック用のコードをビルドに含めるかどうかを選択
Software Start	Enabled, Disabled  Default: Disabled	ソフトウェアスタートの選択
Link section to keep DTC vector table	Default: .ssp_dtc_vector_table	DTC ベクタテーブルを保持するリンクセクションの選択
Name	Default: g_transfer1	モジュール名
Mode	(Locked) Normal	モードの選択 (ロックされています)

ISDE のプロパティ	Value	説明
Transfer Size	(Locked) 1 Bytes	転送サイズの選択 (ロックされています)
Destination Address Mode	(Locked) Incremented	宛先アドレスモードの選択 (ロックされています)
Source Address Mode	(Locked) Fixed	ソースアドレスモードの選択 (ロックされています)
Repeat Area (Unused in Normal Mode)	(Locked) Destination	リピートエリアの選択 (ロックされています)
Interrupt Frequency	(Locked) After all transfers have completed	割り込み頻度の選択 (ロックされています)
Destination Pointer	(Locked) NULL	宛先ポインタの選択 (ロックされています)
Source Pointer	(Locked) NULL	ソースポインタの選択 (ロックされています)
Number of Transfers	(Locked) 0	転送回数の選択 (ロックされています)
Number of Blocks (Valid only in Block Mode)	(Locked) 0	ブロック数の選択 (ロックされています)
Activation Source (Must enable IRQ)	(Locked) Event SCI0 RXI	アクティベーションソースの選択 (ロックされています)
Auto Enable	(Locked) False	自動有効の選択 (ロックされています)
Callback (Only valid with Software start)	(Locked) NULL	コールバックの選択 (ロックされています)
ECL Software Event Interrupt Priority	Priority 0 (highest), Priority 1~14, Priority 15 (lowest, not valid if using Thread X), Disabled  Default: Disabled	ELC ソフトウェアイベント割り込み優先順位選択

注： 設定例とデフォルトは、Synergy S7G2 MCU グループを使用するプロジェクトに対するものです。その他の MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

## 5.2 SPI フレームワークモジュールのクロック構成 (SPI Framework Module Clock Configuration)

SPI 周辺モジュールは PCLKB をクロックソースとして使用します。PCLKB 周波数を設定するには、ビルドの前に SSP コンフィギュレータの [clock] タブを使用するか、ランタイムで CGC インタフェースを使用します。

## 5.3 SPI フレームワークモジュールのピン構成 (SPI Framework Module Pin Configuration)

SPI 周辺モジュールは、MCU のピンを使用して外部デバイスと通信します。I/O ピンは、外部デバイスの要件に合うように選択して構成する必要があります。次の表は [SSP configuration] ウィンドウ内でのピンの選択方法を示し、その次の表は SPI ピンの選択例を示しています。

注： 一部の周辺回路では、動作モードの選択によって、使用可能な周辺回路信号と必要な MCU ピンが決定されます。

表 12 SPI フレームワークモジュールのピン選択

リソース	ISDE Tab	Pin selection Sequence
SCI	Pins	Peripherals > Connectivity:SCI> SCI1
RSPi	Pins	Peripherals > Connectivity:SPI > SPI0

注： 上の選択シーケンスでは SCI1 と SPI0 がドライバに必要なハードウェアターゲットであることを想定しています。

表 13 SPI フレームワークモジュールのピン設定

Property	設定値	説明
Pin Group Selection	Mixed, _A only, _B only Default: Disable	ピングループの選択
Operation Mode	Disabled, Custom Asynchronous UART, Simple SPI, Simple I2C, Synchronous UART, Smart Card Default: Disabled	SCI 上の SPI の動作モードとして簡易 SPI を選択します。
TXD_MOSI	None, P213, P709 Default: None	MOSI ピンに選択
RXD_MISO	None, P212, P708 Default: None	MISO ピンの選択
SCK	None, P100, P112, P710 Default: None	SCK ピンの選択
CTS_RTS_SS	None, P101, P711 Default: None	SS ピンの選択

注： 設定例とデフォルトは、Synergy S7G2 MCU グループと SK-S7G2 Kit を使用するプロジェクトに対するものです。その他の Synergy Kit と MCU のデフォルト値と使用可能な構成設定は異なる可能性があります。

## 5.4 SPI 通信フレームワークモジュールの追加の設定（SPI Framework Module Additional Settings）

外部チップ選択が使用されている場合は、チップ選択ピンを GPIO 出力として設定します。

## 6. アプリケーションでの SPI フレームワークモジュールの使用（Using the SPI Framework Module in an Application）

SPI フレームワークモジュールの一般的なアプリケーションでは、単一バス上に複数のスレーブデバイスが必要です。この一般的なアプリケーションの実装例については後述します。第 2 の実装例では、それぞれ 2 つのスレーブデバイスが取り付けられた 2 つのバスを示します。

単一の共有バスに 2 つのスレーブデバイスを実装する方法

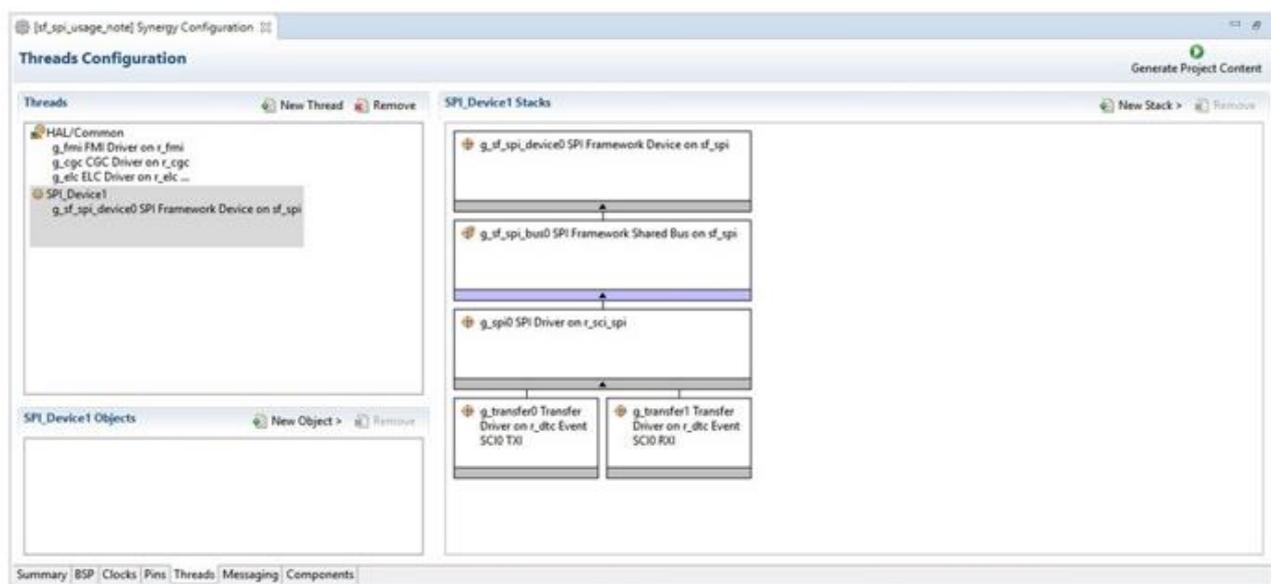
SPI フレームワークモジュールを使用して、複数のスレーブデバイスを持つ単一のバスを作成する場合は、それぞれ SPI フレームワークインスタンス（framework instance）を持つ 2 つのスレッドスタック（thread

stack) を作成します。これらのインスタンスは、同じ共有バスインスタンスを使用します。以下の手順に従って、SSP コンフィグレータ内でこれがどのように行われるかを確認してください。

注：次の手順は、ユーザが SSP 開発環境に習熟していることを前提としています。そうでない場合は、「SSP ユーザーズマニュアル」の第 3 章を参照してください。

ステップ 1：新しいスレッドまたは既存のスレッドに、第一の SPI フレームワークデバイスモジュールを追加します。これにより、SPI マスタスタックが作成されます。sf\_spi の共有バスが SPI ドライバとともに追加されます。SPI ドライバは、r\_rspi または r\_sci\_spi での実装のために選択できます。DTC 転送ドライバもデフォルトで追加されます。これは、CPU 転送モードが代わりに必要な場合は削除できます。結果のモジュールスタックを次の図に示します。

結果のモジュールスタックを次の図に示します。設定例は、図に続く表に示されます。



スレーブデバイス #1 のキー第 1 スレッドスタックモジュールの構成設定の例は、次のとおりです。

表 14 sf\_spi 上の SPI フレームワークモジュールの設定

Property	Value	説明
Parameter Checking	Disabled	パラメータチェックを有効または無効にする。
Name	g_sf_spi_device1	SPI フレームワークデバイスを識別するための名前を付けます。この名前に基づいて API、Config および Control インスタンスが作成されます。
Clock Phase	Data sampling on odd edge, data variation on even edge	データ変化とデータサンプリングのクロック位相を指定します。
Clock Polarity	Low when idle	クロックがアイドル状態のときは、クロックの極性を選択します。
Chip Select Port	01	チップセレクトに使用する GPIO ポートを選択します。

Property	Value	説明
Chip Select Pin	04	チップセレクトに使用する GPIO ピンを選択します。
Chip Select Active Level	Low	チップセレクトアクティブレベルを選択します。

表 15 sf\_spi の SPI フレームワーク共有バスの設定

Property	Value	説明
Name	g_sf_spi_bus0	SPI フレームワーク共有バスを識別するための名前を付けます。この共有バスは、複数の SPI フレームワークデバイスによって共有されます。

表 16 r\_spi 使用時の SPI ドライバの設定

Property	Value	説明
Parameter Checking	Default (BSP)	パラメータチェックを有効または無効にする。
Name	g_spi0	SPI ドライバデバイスを識別するための名前を付けます。これは Framework によって内部的に使用されます。
Channel	0	チャンネル番号
Operating Mode	Master	動作モードの選択
Clock Phase	(Locked) Data sampling on odd edge, data variation on even edge	クロック位相選択。このフィールドは、sf_spi モジュール上の SPI Framework Device ですすでに設定されているため、ロックされます。
Clock Polarity	(Locked) Low when idle	クロック極性の選択。このフィールドは、sf_spi モジュール上の SPI Framework Device ですすでに設定されているため、ロックされます。
Mode Fault Error	Enable	モード障害エラーの選択
Bit Order	MSB First	ビットオーダの選択
Bitrate	500000	ビットレートの選択
Callback	(Locked) NULL	コールバック関数名 (ロックされています)
SPI Mode	SPI Operation	SPI モードの選択
Slave Select Polarity (SSL0)	Active Low	スレーブ選択極性 0 の選択
Slave Select Polarity (SSL1)	Active Low	スレーブ選択極性 1 の選択
Slave Select Polarity (SSL2)	Active Low	スレーブ選択極性 2 の選択
Slave Select Polarity (SSL3)	Active Low	スレーブ選択極性 3 の選択
Select Loopback 1	Normal	ループバック 1 の選択
Select Loopback 2	Normal	ループバック 12 の選択
Enable MOSI Idle	Disable	MOSI アイドルの有効化の選択
MOSI Idle State	MOSI Low	MOSI アイドル状態の有効化の選択
Enable Parity	Disable	パリティ有効化の選択
Parity Mode	Parity Odd	パリティモードの有効化の選択
Select SSL (Slave Select)	SSL0	SSL 選択の選択
Select SSL Level After Transfer	SSL Level Keep	転送後の SSL レベル選択の選択
Clock Delay Enable	Clock Delay Disable	クロック遅延有効化の選択
Clock Delay Count	Clock Delay 1 RSPCK	クロック遅延カウンタの選択
SSL Negotiation Delay Enable	Negotiation Delay Disable	SSL ネゲート遅延有効化の選択

Property	Value	説明
Negation Delay Count	Negation Delay 1 RSPCK	ネゲーション遅延カウンタの選択
Next Access Delay Enable	Next Access Delay Disable	次のアクセス遅延有効の選択
Next Access Delay Count	Next Access Delay 1 RSPCK	次のアクセス遅延カウンタの選択
Receive Interrupt Priority	Priority 2	割り込み優先順位選択を受信
Transmit Interrupt Priority	Priority 2	割り込み優先順位の選択を送信
Transmit End Interrupt Priority	Priority 2	送信終了割り込み優先順位の選択
Error Interrupt Priority	Priority 2	エラー割り込み優先順位の選択

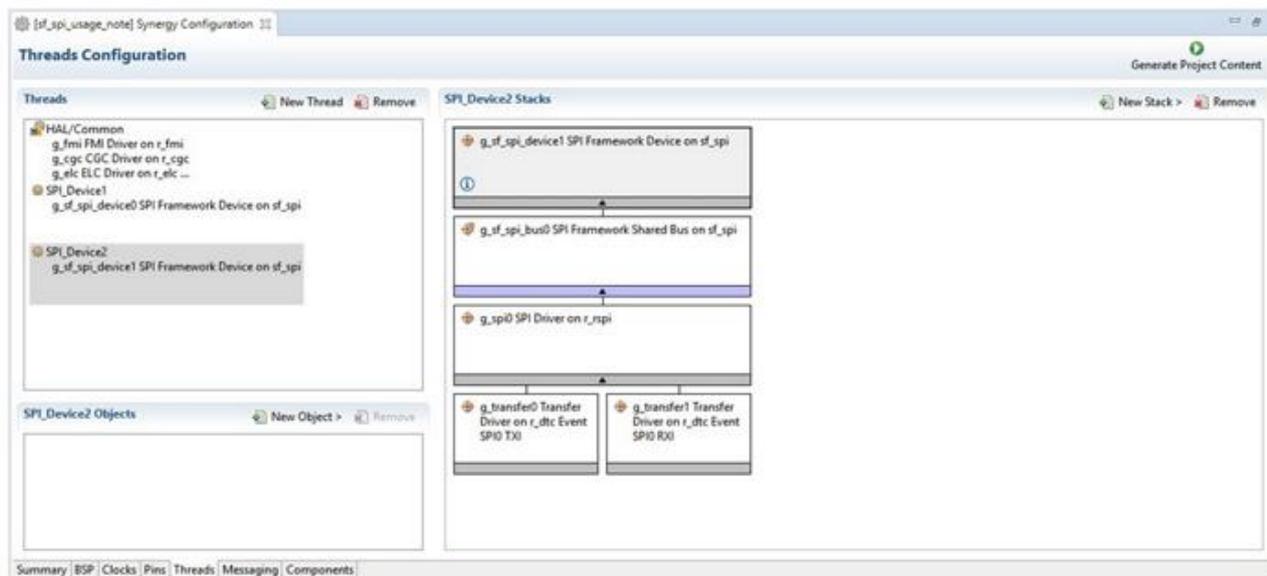
表 17 r\_sci\_spi 使用時の SPI ドライバの設定

Property	Value	説明
Parameter Checking	Default (BSP)	パラメータチェックを有効または無効にする。
Name	g_spi0	SPI ドライバデバイスを識別するための名前を付けます。これは Framework によって内部的に使用されます。
Channel	0	チャンネル番号
Operating Mode	Master	動作モードの選択
Clock Phase	(Locked) Data sampling on odd edge, data variation on even edge	クロック位相選択。このフィールドは、sf_spi モジュール上の SPI Framework Device ですすでに設定されているため、ロックされます。
Clock Polarity	(Locked) Low when idle	クロック極性の選択。このフィールドは、sf_spi モジュール上の SPI Framework Device ですすでに設定されているため、ロックされます。
Mode Fault Error	Disable	モード障害エラーの選択
Bit Order	MSB First	ビットオーダーの選択
Bitrate	500000	ビットレートの選択
Bit Rate Modulation Enable	Enable	ビットレート変調を有効/無効にする
Callback	(Locked) NULL	コールバック関数名 (ロックされています)
Receive Interrupt Priority	Priority 2	割り込み優先順位選択を受信
Transmit Interrupt Priority	Priority 2	送信割り込み優先順位の選択
Transmit End Interrupt Priority	Priority 2	送信終了割り込み優先順位の選択
Error Interrupt Priority	Priority 2	エラー割り込みの優先順位の選択

注：DTC の設定は簡略化の為、示されていません。

ステップ 2：2 番目の SPI フレームワークデバイスを別のスレッドに追加します。sf\_spi 上の SPI フレームワーク共有バスは自動的に追加されません。追加するには、既存の共有バスを使用するオプションを選択します。コンフィギュレータは、sf\_spi と残りのモジュールに SPI フレームワーク共有バスを自動的に追加します。ローレベルモジュールは、最初に定義された SPI フレームワークインスタンスの設定と一致するように自動的に設定されます。これらは SPI フレームワークデバイスモジュールで定義されており、各スレーブデバイス向けに異なる設定ができますので、クロックフェーズ、クロック極性、チップセレクトピンとポート、およびチップセレクトアクティブレベルプロパティを除いて、両方のデバイスで SPI ドライバのコンフィギュレーションが同じになります。

モジュールスタックの結果を次の図に示します。



2 番目のスタックのコンフィギュレーションパラメータの唯一の違いは、2 番目の SPI フレームワークデバイスモジュールの名前と、非共有スレーブ設定（クロックフェーズ、クロック極性、クロックセレクトポート、チップセレクトピン、チップセレクトアクティブレベル）です。設定例を次の表に示します。

表 18 sf\_spi 上の SPI フレームワークデバイスの設定（スレーブ #2）

Property	Value	説明
Parameter Checking	Default (BSP)	パラメータチェックを有効または無効にする。
Name	g_sf_spi_device2	SPI ドライバデバイスを識別するための名前を付けます。これは Framework によって内部的に使用されます。
Clock Phase	Data sampling on odd edge, data variation on even edge	データ変化とデータサンプリングのクロック位相を指定する。
Clock Polarity	Low when idle	クロック極性選択
Clock Select Port	05	クロック選択に使用する GPIO ポートを選択
Chip Select Pin	01	チップ選択に使用する GPIO ピンを選択
Chip Select Active Level	Low	チップ選択信号の極性を選択

### 2つの共有バス上の2つのスレーブデバイスの実装手順

SPI フレームワークモジュールを使用して複数のスレーブデバイスを持つ2つの共有バスを作成する場合は、前の例(単一の共有バスに対して2つのスレーブデバイスが付く)で実装された2つのスレッドスタック (SPI フレームワークインスタンスを持つ) に対して、2つ目の共有バスを使用する SPI フレームワークインスタンスをそれぞれ追加で作成します。

これら4つの SPI フレームワークインスタンスは、2つの共有バスインスタンスを使用します。以下の手順に従って、SSP コンフィグレータ内でこれがどのように行われるかを確認してください。

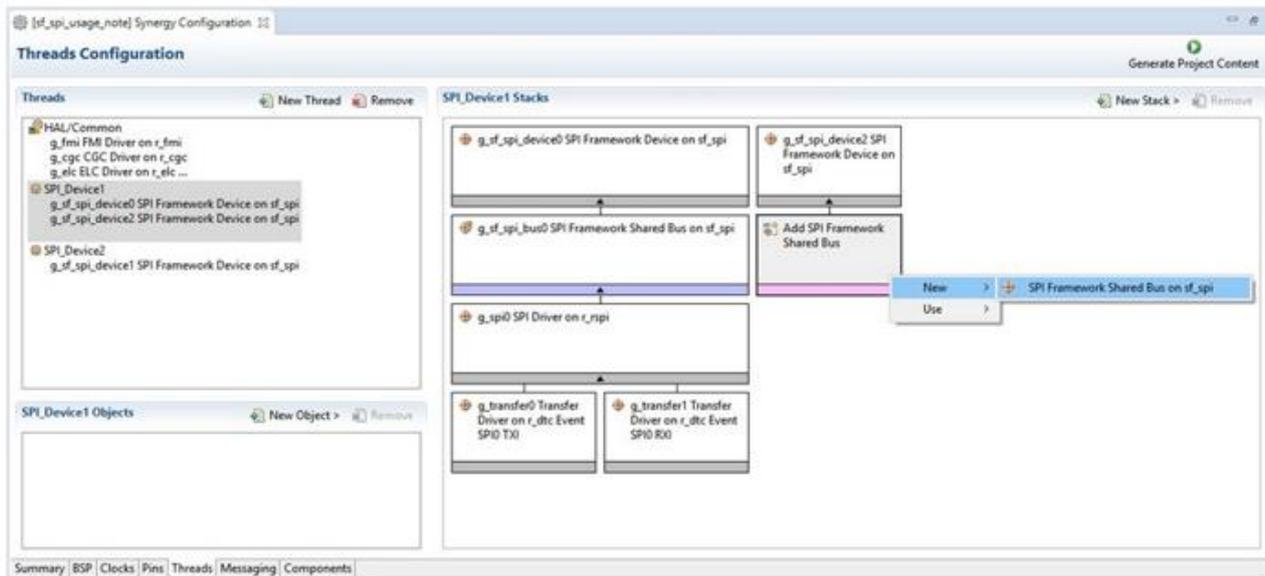
注：次の手順は、ユーザが SSP 開発環境に習熟していることを前提としています。そうでない場合は、「SSP ユーザーズマニュアル」の第3章を参照してください。

別の共有バスの追加

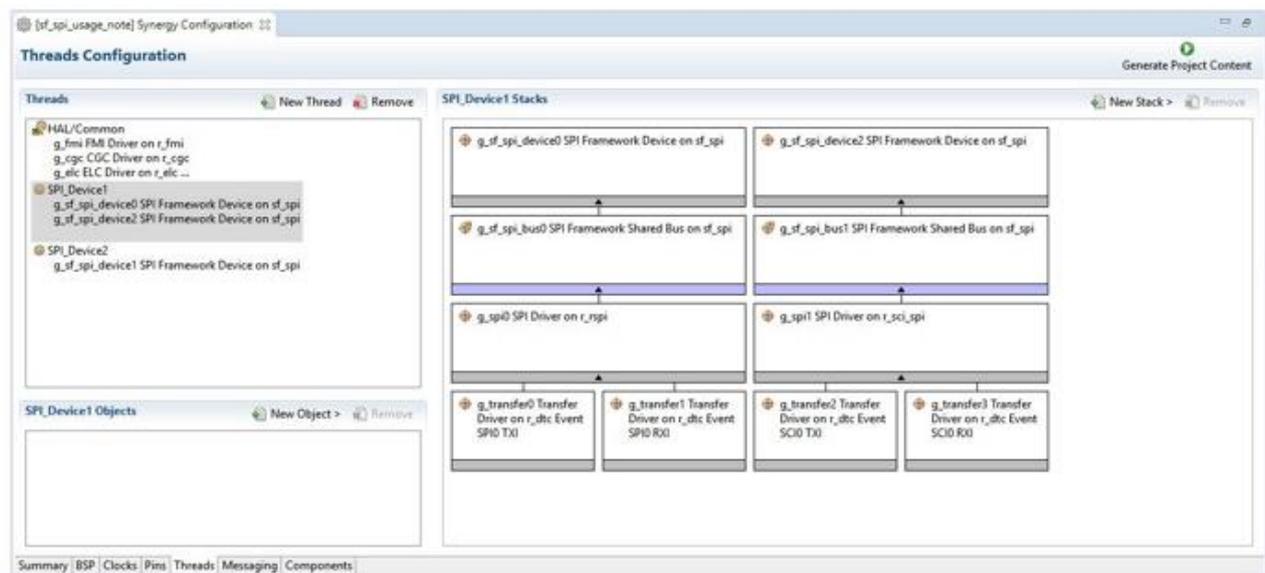
別の共有バスを追加するには、以下の手順に従います。前の例は開始点として使用されています。

1) 2番目の共有バスを使用する SPI フレームワークモジュールを任意のスレッドに追加できます。前の例から始めて、SPI\_Device1 スレッドに追加された場合、モジュールスタックは次のようになります。

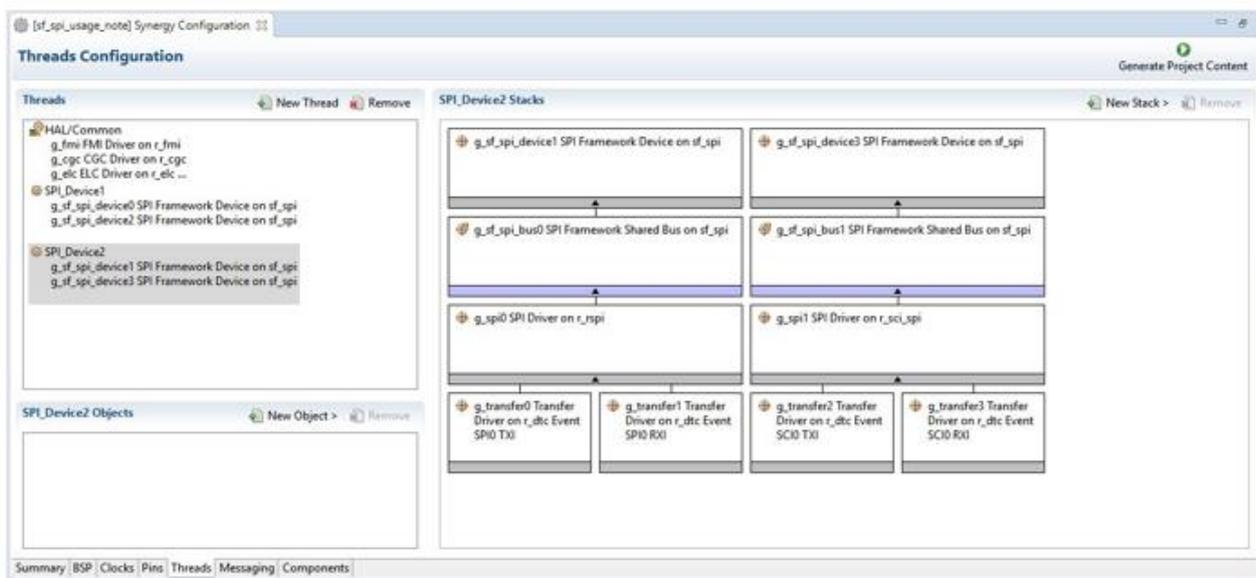
共有バスに使用できるオプションは、新規または既存バスの使用(Use)です。



2) [新規作成]を選択し、sf\_spi モジュールに別の SPI フレームワーク共有バスを追加します。アプリケーションの必要に応じて、共有バスのプロパティを設定します。目的のローレベル SPI ドライバを選択します。g\_spi1 SPI ドライバモジュールのチャンネル番号は、g\_spi0 SPI ドライバモジュールのチャンネル番号と異なる必要があります。結果のスレッドスタックを以下に示します。



3) 上記と同じ手順を使用して、2 番目のデバイスを SPI\_Device2 スレッドに追加することができます。結果のスレッドスタックを以下に示します。



アプリケーションで SPI フレームワークモジュールを使用する際の一般的な手順は次のとおりです。

1. オープン API 関数を使用して SPI フレームワークデバイスモジュールを初期化します。各 SPI フレームワークデバイスモジュールは、バス上で操作を実行する前に、少なくとも 1 回オープン API 関数を呼び出す必要があります。
2. 特定の SPI フレームワークデバイスモジュールのロック API 機能を使用して、連続転送のためにバスをロックします。バスが特定の SPI フレームワークデバイスモジュールによってロックされると、そのバス上の他の SPI フレームワークデバイスモジュールは使用できません。これにより、明示的にロックを解除するまで、バスの所有権がロックされたモジュールに残ることが保証されます。バス上の他の SPI フレームワークデバイスモジュールからのあらゆる種類の操作は、この期間中にフェイル・ステータスを返します。バス上の読み取り/書き込み操作の前にバスをロックすることは必須ではありません。これはオプションです。
3. 読み取り API 関数を使用してデータを読み取ります。バスがすでに他の SPI フレームワークデバイスモジュールによってロックされている場合、読取り操作は成功しません。
4. 書き込み API 関数を使用してデータを書き込みます。バスが他の SPI フレームワークデバイスモジュールによってすでにロックされていると、書き込み操作は成功しません。
5. writeRead API 関数を使用して同時にデータを書き込み、読み取ります。バスがすでに他の SPI フレームワークデバイスモジュールによってロックされている場合、同時読出しおよび書き込み動作は成功しません。
6. ロック解除 API 機能が既に同じデバイスによってロックされている場合、バスを連続転送からロック解除します。バスがロック解除されると、他の SPI フレームワークデバイスモジュールがそれを使用することができます。意図された読み出し/書き込み動作が完了した後で、ロックされたバスのロックを解除する必要があります。
7. close API 関数を使用して SPI フレームワークデバイスモジュールを閉じます。各 SPI フレームワークデバイスモジュールは、バス上のすべての読み出し/書き込み動作が終了した後で、クローズ API 機能を呼び出すことができます。

これらの一般的な手順を、次の図の通常の動作フロー図に示します。

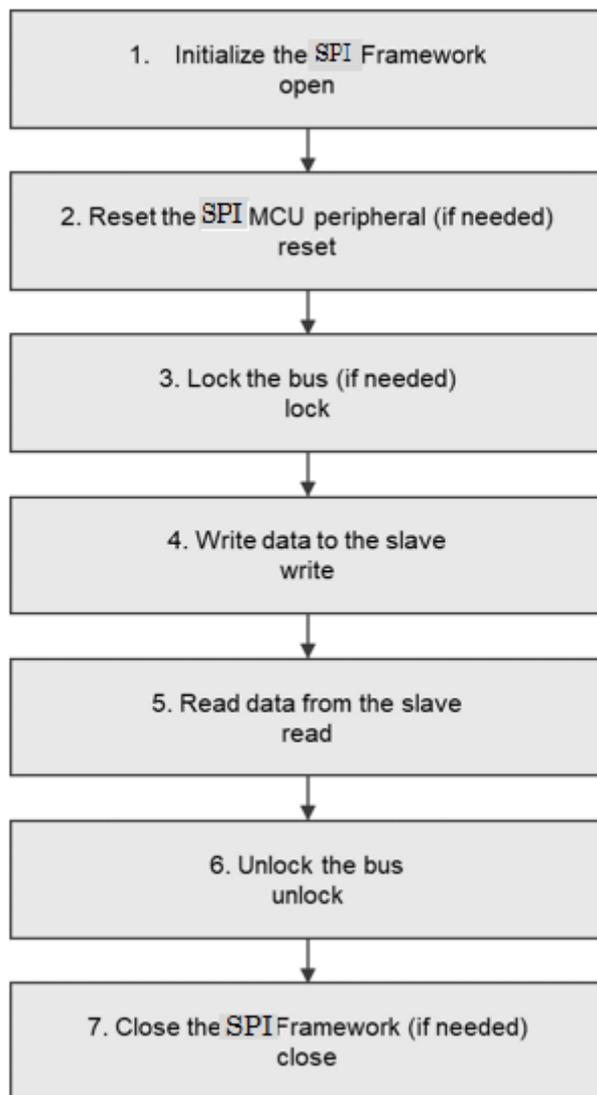


図 3 通常の SPI フレームワークアプリケーションのフロー図

## 7. SPI フレームワークモジュールのアプリケーションプロジェクト (The SPI Framework Module Application Project)

このモジュールガイドで説明するアプリケーションプロジェクトを実際に使うことで、前述の設計全体の手順を体験することができます。このプロジェクトは、このドキュメントの末尾にある「参考情報」章に掲載されているリンクにあります。ISDE でアプリケーションプロジェクトをインポートして開き、SPI フレームワークモジュールに対応する設定項目を表示することができます。また、完成した設計で、SPI フレームワークモジュール API を示すために使用している `temperature_thread_entry.c` 内のコードを確認することもできます。

このアプリケーションプロジェクトは、SPI フレームワークモジュール API の一般的な使用方法を示します。温度センサ MAX31723 は PMODA に接続されています。このセンサは SPI スレーブデバイスとして動作し、現在の環境の温度を読み取ります。複数の LED は、温度差に基づいて点灯します。温度は 1 秒の間隔で定期的に測定されます。

表 19 このアプリケーションプロジェクトが使用するソフトウェアとハードウェアのリソース

リソース	リビジョン	説明
e <sup>2</sup> studio	5.3.1 またはそれ以降	統合ソリューション開発環境 (ISDE)
IAR EW for Synergy	7.71.2 またはそれ以降	IAR Embedded Workbench® for Renesas Synergy™
SSP	1.2.0 またはそれ以降	Synergy ソフトウェアパッケージ
SSC	5.3.1 またはそれ以降	Synergy Standalone Configurator
SK-S7G2	v3.0 と v3.1	スタータキット

次の図に、このアプリケーションプロジェクトの基本的なフロー図を示します。

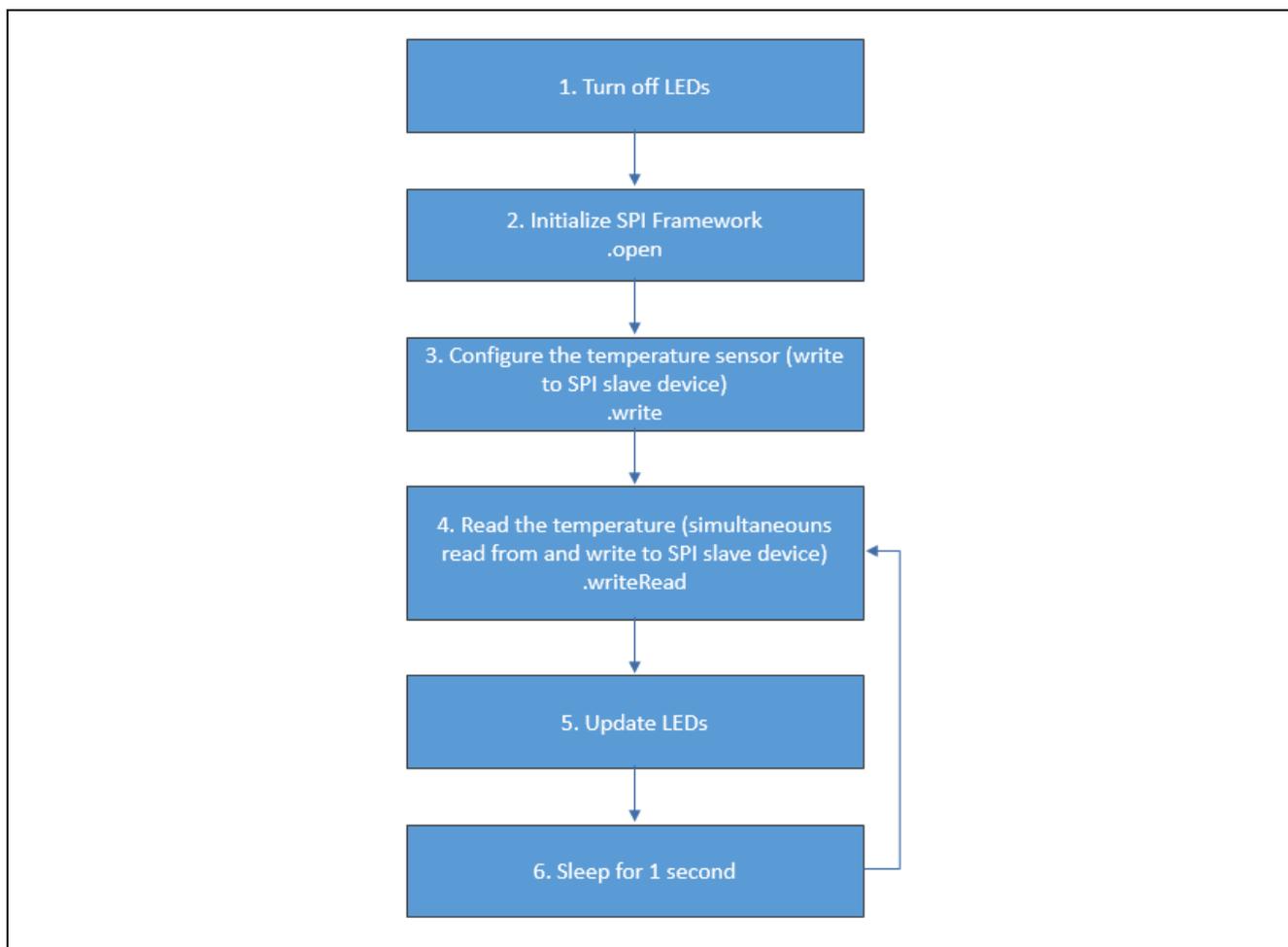


図 4 SPI フレームワークモジュールのアプリケーションプロジェクトのフロー図

temperature\_thread\_entry.c ファイルは、このプロジェクトを一旦 ISDE にインポートすると、プロジェクト内に格納されます。ISDE でこのファイルを開くことができ、API の主な使い方を確認するための説明も参照できます。

temperature\_thread\_entry.c の最初のセクションはヘッダファイルであり、SPI インスタンスの構造体と、温度の浮動小数点演算を実行するために使用する数学関数 (math functions) を参照しています。#define を通じてこのセクションを有効にすると、printf() を使用して結果を表示するセミホスト機能を可能にするコードセクション (code section) がインクルードされます。その後、このアプリケーションで使用するグローバル変数の定義や、関数プロトタイプが続きます。

このスレッドのエントリ関数 (entry function) は、`temperature_thread_entry()` です。関数内には、温度計算用のローカル変数が定義されているほか、温度センサや、受信した温度データの保存場所の設定を行う設定データを格納するデータ配列があります。

このアプリケーションプロジェクトが計算した温度に従って LED の点灯を行う過程で、LED は開始時 (すべて消灯) の状態に設定されます。

次に、SPI フレームワークを開きます。正常に開いた場合、その次の手順で温度センサを設定します。`write` API を使用して、設定データを温度センサに書き込みます。このデータは、12 ビットの分解能で温度センサを設定します。`write` 関数が正常に完了すると、`SSP_SUCCESS` が返されます。

次に、このアプリケーションはスレッド `while(1)` ループに入ります。ここでは、`writeRead` API を使用して温度を読み取ります。ここでも、`writeRead` 関数が正常に完了すると、`SSP_SUCCESS` が返されません。書き込まれるデータは、温度データの読み取られたアドレスです。温度データは 12 ビットのサイズなので、少なくとも 2 バイトのデータを読み取る必要があります。温度データの保存場所は 3 バイトのサイズです。これはアドレスを書き込むプロセスで、ダミーデータを受信するためです。

次に、受信した有効な 2 バイトを使用して、温度を計算します。

アプリケーションが計算した最初の温度を、基準温度として保存します。これ以降、1 秒ごとに発生するすべての温度計算を、最初の基準温度と比較します。新しい温度が基準温度と異なっている場合、温度差に応じて複数の LED が点灯します。温度差が 1 度の場合は緑の LED が点灯し、温度差が 2 度の場合は緑と橙の LED が点灯し、温度差が 3 度の場合は緑と橙と赤の LED が点灯します。

表示機能が有効になっている場合、測定した温度をデバッグコンソールに表示します。

注： ここまでの説明は、Synergy ソフトウェアパッケージ内のデバッグコンソールで `printf()` を使用方法をユーザが理解していることを想定しています。デバッグコンソールで `printf()` を使用方法については、このドキュメントの末尾にある「参考情報」の章で紹介しているナレッジベースの記事、「*How do I Use Printf() with the Debug Console in the Synergy Software Package*」(Synergy ソフトウェアパッケージのデバッグコンソールで `Printf()` を使用方法) を参照してください。代わりに、デバッグモードで変数ウォッチ機能を使用して結果を表示することもできます。

対象ボードや MCU の必要な動作と物理プロパティをサポートするために、このアプリケーションプロジェクトではいくつかの重要なプロパティを設定しています。この特定のプロジェクトで設定するプロパティを、以下の表に示します。アプリケーションプロジェクトを開き、[Properties] ウィンドウでこれらの設定を表示することもできます。

表 20 アプリケーションプロジェクトに対応する SPI フレームワークモジュールの設定項目

ISDE のプロパティ	設定済みの値
Name	<code>g_sf_spi_device</code>
Chip Select Port	01
Chip Select Pin	03
Chip Select Active Level	High

注： この設定が想定しているのは、103 ピンを GPIO ピンとして使用し、モードを「Output mode (Initial Low)」にするよう [Pins] タブで設定した状況です。

表 21 アプリケーションプロジェクトに対応する SPI フレームワークモジュール共有バスの設定項目

ISDE のプロパティ	設定済みの値
Name	<code>g_sf_spi_bus0</code>

表 22 アプリケーションプロジェクトに対応する SPI HAL モジュール (r\_rsapi) の設定項目

ISDE のプロパティ	設定済みの値
Name	g_spi
Channel	0
Operating Mode	Master
Clock Phase	Data sampling on even edge, data variation on odd edge (奇数エッジでデータをサンプリングし、偶数エッジでデータを検証)
Clock Polarity	High when idle
Mode Fault Error	Disable
Bit Order	MSB First
Bitrate	500000
Callback	NULL
SPI Mode	SPI Operation
Slave Select Polarity (SSL0)	Active Low
Slave Select Polarity (SSL1)	Active Low
Slave Select Polarity (SSL2)	Active Low
Slave Select Polarity (SSL3)	Active Low
Select Loopback1	Normal
Select Loopback2	Normal
Enable MOSI Idle	Disable
MOSI Idle State	MOS Low
Enable Parity	Disable
Parity Mode	Parity Odd
Select SSL (Slave Select)	SSL0
Select SSL Level After Transfer	SSL Level Do Not Keep
Clock Delay Enable	Clock Delay Disable
Clock Delay Count	Clock Delay 1 RSPCK
SSL Negation Delay Enable	Negation Delay Disable
Negation Delay Count	Negation Delay 1 RSPCK
Next Access Delay Enable	Next Access Delay Disable
Next Access Delay Count	Next Access Delay 1 RSPCK
Receive Interrupt Priority	Priority 2
Transmit Interrupt Priority	Priority 2
Transmit End Interrupt Priority	Priority 2
Error Interrupt Priority	Priority 2

## 8. 対象アプリケーションの SPI フレームワークモジュールのカスタマイズ (Customizing the SPI Framework Module for a Target Application)

ユーザは、スレーブデバイスに応じて、SPI 送信、ビットレート、その他のパラメータに使用するチャンネルを変更することもできます。最も重要な変更は、代替 SPI を実装することによって、SCI の Simple SPI を使用することです。SPI フレームワークデバイスは、r\_sci\_spi の SPI HAL モジュールを使用して設定する必要があります。温度センサに対して PMODB を使用する必要があります。次の表に詳細を示します。

**表 23 SCI SPI オプションに対応する SPI フレームワークモジュールの構成設定**

ISDE のプロパティ	設定済みの値
Name	g_sf_spi_device
Chip Select Port	04
Chip Select Pin	13
ChipSelect Active Level	High

注： この設定が想定しているのは、413 ピンを GPIO ピンとして設定し、モードを「Output mode (初期値 Low)」にするよう [Pins] タブで設定した状況です。

**表 24 SCI SPI オプションに対応する SPI フレームワークモジュール共有バスの設定**

ISDE のプロパティ	設定済みの値
Name	g_sf_spi_bus

**表 25 SCI SPI オプションに対応する SPI HAL モジュール (r\_sci\_spi) の構成設定**

ISDE のプロパティ	設定済みの値
Name	g_spi
Channel	0
Operating Mode	Master
Clock Phase	Data sampling on even edge, data variation on odd edge (奇数エッジでデータをサンプリングし、偶数エッジでデータを検証)
Clock Polarity	High when idle
Mode Fault Error	Disable
Bit Order	MSB First
Bitrate	100000
Bit Rate Modulation Enable	Enable
Callback	NULL
Receive Interrupt Priority	Priority 8
Transmit Interrupt Priority	Priority 8
Transmit End Interrupt Priority	Priority 8
Error Interrupt Priority	Priority 8

## 9. SPI フレームワークモジュールのアプリケーションプロジェクトの実行 (Running the SPI Framework Module Application Project)

ISDE にこのプロジェクトをインポートし、コンパイルしてデバッグを実行するだけで、SPI フレームワークのアプリケーションプロジェクトを実行させ、対象キットでその動作を観察することができます。

新しいプロジェクト内で SPI フレームワークアプリケーションを実装するには、対象キットで定義付け、設定、ファイルの自動生成、コードの追加、コンパイル、デバッグを行うための手順に従います。これらの手順を実践することで、SSP を使用する開発プロセスの習得が容易になります。

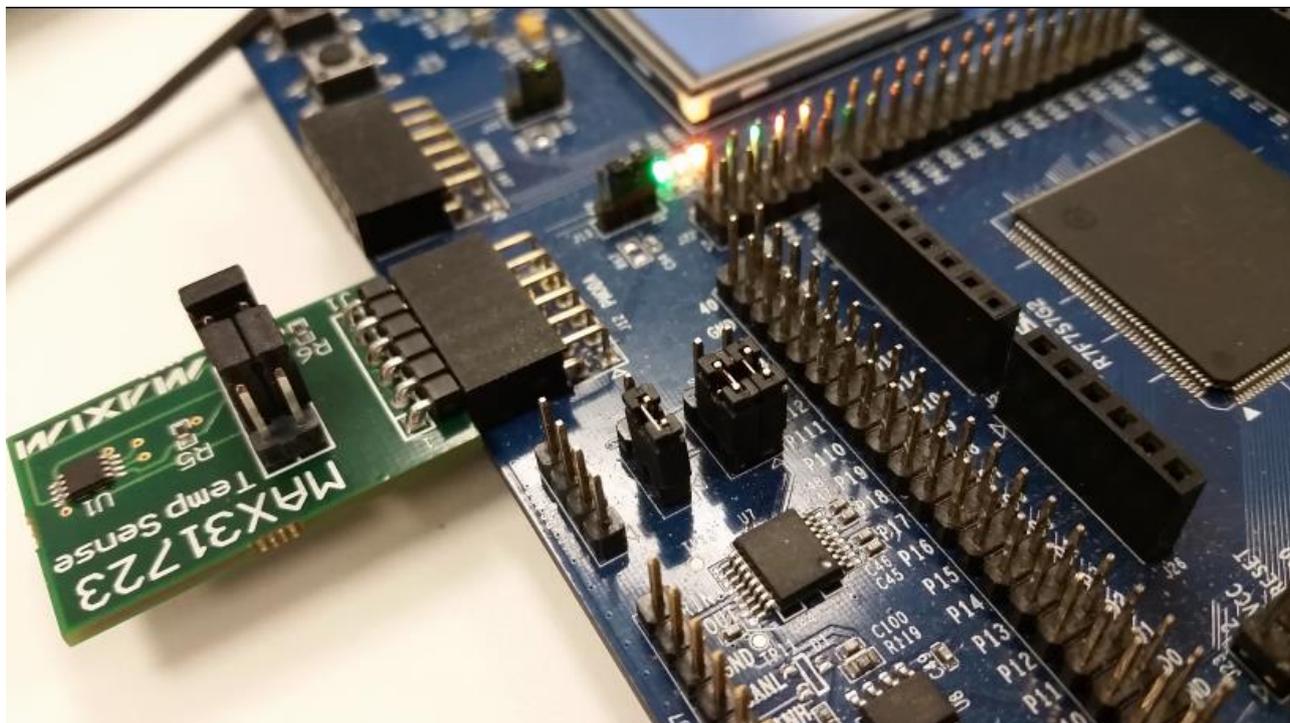


図 5. PMODA に接続されている温度センサ

注： Synergy 開発プロセスの基本的な流れを経験したことがあるユーザにとって、以下の手順は十分になっています。これらの手順をまだ理解していない場合、このドキュメントの末尾にある「参考情報」章に掲載されている『SSP ユーザーズマニュアル』の最初にあるいくつかの章を参照してください。

SPI フレームワークのアプリケーションプロジェクトを作成し、実行するために、以下の手順に従ってください。

1. 図 5 のように温度センサを接続します。
2. SK-S7G2 ボードに対応する新しい Renesas Synergy プロジェクトを作成し、**SPI\_Framework\_MG\_AP** という名前を付けます。
3. **[Threads]** タブを選択します。
4. 次のような新しいスレッドを追加します。

Symbol	<b>temperature_thread</b>
Name	<b>Temperature Thread</b>
5. **[Temperature Thread the SPI Framework]** にスレッドを追加し、設定を行います。
6. **[Generate Project Content]** ボタンをクリックします。
7. 付属のプロジェクトファイル `temperature_thread_entry.c` からコードを追加するか、生成された `temperature_thread_entry.c` ファイルに上書きする形でコピーします。
8. micro USB ケーブルを SK-S7G2 スタータキットの J19 につないで、ホスト PC に接続します。
9. アプリケーションのデバッグを開始します。
10. 温度センサに触り (温度を変化させる)、複数の LED の点灯を観察します。

予期される出力結果の例：

```
Init temperature: 25.50
Current temperature: 25.06
Init temperature: 25.50
Current temperature: 25.00
Init temperature: 25.50
Current temperature: 25.00
Init temperature: 25.50
Current temperature: 25.00
```

出力される温度値が 100°C を超えている場合は、温度センサが接続されていないか、故障しています。

## 10. SPI フレームワークモジュールのまとめ (SPI Framework Module Conclusion)

このモジュールガイドは、サンプルプロジェクトでモジュールの選択、追加、設定、使用を行うために必要な背景となる情報全般を説明しました。従来の組み込みシステムでは、これらの手順を理解することに多くに時間を必要とし、また間違いが起こりやすい操作でした。Renesas Synergy プラットフォームにより、これら手順の所要時間が短くなり、設定項目の競合や、ローレベルドライバの誤った選択など、誤りが防止できるようになりました。アプリケーションプロジェクトで示したように、ハイレベル API を使用することで高いレベルの開発からスタートし、ローレベルドライバを作成するような従来の開発環境で必要とされる時間が不要になり、開発時間を短縮できます。

## 11. SPI フレームワークモジュールの次の手順 (SPI Framework Module Next Steps)

シンプルな SPI フレームワークモジュールのプロジェクトを習得した後は、より複雑なサンプルを確認するか、他の SPI スレーブデバイスを使用することができます。SCI SPI の実装を使用して実験することも可能です。SCI SPI や RSPI に関連する各アプリケーションプロジェクトを比較し、SSP フレームワークを使用する場合の利点を確認してください。

## 12. SPI フレームワークモジュールの参考情報 (SPI Framework Module Reference Information)

『SSP ユーザーズマニュアル』：SSP distribution package の一部として html 形式を入手できるほか、Synergy WEB サイト (<https://www.renesas.com/jp/ja/products/synergy.html>) の SSP サイト (<https://www.renesas.com/jp/ja/products/synergy/software/ssp.html>) から PDF 版を入手することもできます。

本プロジェクト (ソフトウェア) は英文ドキュメントとともに、下記サイトから入手できます。未登録の場合は MyRenesas への登録が必要です。

<https://www.renesas.com/jp/ja/software/D6001524.html>

さらに関連参考資料、リソースに関する最新情報は Synergy WEB サイト (下記) から入手できます。

日本語版：<https://www.renesas.com/jp/ja/products/synergy.html>

英語版：<https://www.renesas.com/us/en/products/synergy.html>

## ホームページとサポート窓口

サポート: <https://synergygallery.renesas.com/support>

テクニカルサポート:

- アメリカ: <https://www.renesas.com/en-us/support/contact.html>
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

すべての商標および登録商標はそれぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	2018.11.15		<ul style="list-style-type: none"><li>第 1.01 版 発行</li><li>第 2 章～6 章：SSP v1.5.0 UM 英文版から和訳</li><li>第 7 章～：英文版（資料番号 r11an0117eu0101-synergy-spi-fw-mod-guide Rev1.01、発効日 2017 年 8 月 30 日）を和訳</li><li>第 12 章：シナジーナレッジデータベースの HTTP 変更</li></ul>

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
  5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、  
金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシートにおいて高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
  9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
  11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>