

AGT HAL モジュールガイド

(注1)本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。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。

(注2)本資料の第6章まで(要旨除く)の日本語訳は、「[Synergy™ Software Package \(SSP\) v1.5.0 ユーザーズマニュアル モジュール概要編\(参考資料\)](#)」の第4章「モジュールの概要」に掲載されていますのでそちらを参照ください。

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。

AGT HAL モジュールは、タイミングアプリケーション(timing applications)向けのハイレベルAPI(high-level API)であり、r_agt 内で実装されています。AGT HAL モジュールは、Synergy MCU 上にある AGT 周辺回路(peripheral)を使用します。タイマイイベント(timer event)に応答(respond)するために、ユーザ定義のコールバックを作成できます。

AGT HAL モジュールは、タイマを、ユーザが指定した期間に設定します。その期間が経過(period elapses)した時点で、次のイベントのいずれかを発生させることができます。

- CPU への割り込み(ユーザコールバック関数(user-callback function)の呼び出し)(作成した場合)
- ポート端子をトグル(Toggle a port pin)します。
- DMAC/DTC を使用したデータ転送(転送インタフェース(transfer interface)で設定した場合)。
- 別の周辺回路の起動(イベントと周辺回路の定義(events and peripheral definitions)を使用して設定した場合)。

目次

1. AGT HAL モジュールの特徴(AGT HAL Module Features)	3
2. AGT HAL モジュールのAPIの概要(AGT HAL Module APIs Overview)	3
3. AGT HAL モジュールの動作の概要(AGT HAL Module Operational Overview)	3
4. アプリケーションへのAGT HAL モジュールの組み込み(Including the AGT HAL Module in an Application)	3
5. AGT HAL モジュールの構成(Configuring the AGT HAL Module)	3
6. アプリケーションでのAGT HAL モジュールの使用(Using the AGT HAL Module in an Application)	3
7. AGT HAL モジュールのアプリケーションプロジェクト(The AGT HAL Module Application Project)	3

8. ターゲットアプリケーションに対応する AGT HAL モジュールのカスタマイズ (Customizing the AGT HAL Module for a Target Application)	6
9. AGT HAL 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 AGT HAL Module Application Project)	6
10. AGT HAL モジュールのまとめ (AGT HAL Module Conclusion)	7
11. AGT HAL モジュールの次の手順 (AGT HAL Module Next Steps)	7
12. AGT HAL モジュールの参考情報 (AGT HAL Module Reference Information)	7

1. AGT HAL モジュールの**特徴**(AGT HAL Module Features)
2. AGT HAL モジュールの API の概要(AGT HAL Module APIs Overview)
3. AGT HAL モジュールの動作の概要(AGT HAL Module Operational Overview)
4. アプリケーションへの AGT HAL モジュールの組み込み(Including the AGT HAL Module in an Application)
5. AGT HAL モジュールの構成(Configuring the AGT HAL Module)
6. アプリケーションでの AGT HAL モジュールの使用(Using the AGT HAL Module in an Application)
7. AGT HAL モジュールのアプリケーションプロジェクト(The AGT HAL Module Application Project)

このモジュールガイドに関連するアプリケーションプロジェクトは、サンプルアプリケーションに関する手順を示しています。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」の章に掲載されているリンクから入手できます。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、AGT HAL モジュールに対応する設定を表示することができます。また、完成した設計 (complete design) で、AGT HAL モジュール API を実現するために使用している AGT_HAL_MG_AP.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、タイマドライバ (timer drivers) API の一般的な使用方法を示します。2 つの AGT タイマを使用します。このアプリケーションプロジェクトのメインスレッドのエントリは、タイマを初期化し、周期とデューティサイクル (duty cycle) パラメータの起動と再設定を行います。いずれかのタイマが終了した時点で、コールバックが呼び出され、LED がトグルします。タイマに関する情報を取得する関数は、デバッグコンソールに結果を出力します。以下の表は、このアプリケーションプロジェクトが使用する関連ソフトウェアとハードウェアの対象バージョンを示します。

表 1 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 ² studio	6.2.1 またはそれ以降	統合ソリューション開発環境 (ISDE)
SSP	1.5.0 またはそれ以降	Synergy ソフトウェアプラットフォーム
IAR EW for Synergy	8.23.1 またはそれ以降	IAR Embedded Workbench® for Renesas Synergy™
SSC	6.2.1 またはそれ以降	Synergy Standalone Configurator
SK-S7G2	v3.0, v3.1 またはそれ以降	スタータキット

以下の図は、このアプリケーションプロジェクトの簡単なフローを示します。

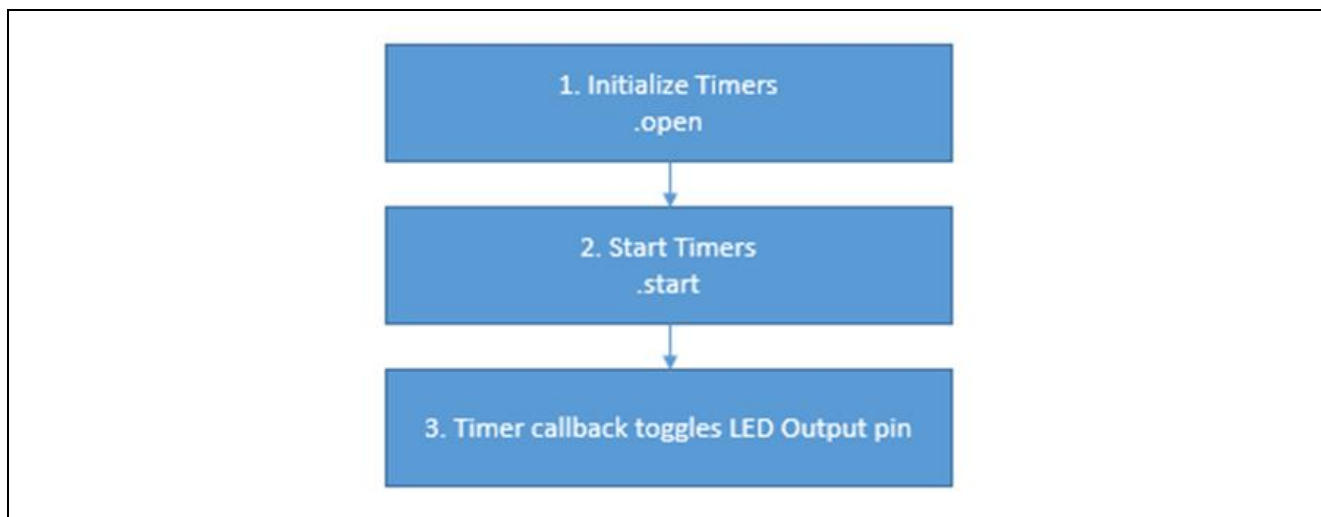


図 1 AGT HAL モジュールのアプリケーションプロジェクトのフロー図

このプロジェクトは、2 個のタイマを使用します。

1. g_timer_agt_0 は、16 ビットの周期 AGT (periodic AGT)を設定します。
2. g_timer_agt_1 は、32 ビットの周期 AGT を設定し、g_timer_agt_0 がクロックを供給します。

以下の表に、使用する各タイマの設定プロパティを示します。他の項目にはデフォルト値を設定します。

表 2 g_timer_agt_0 に対応する設定プロパティ

ISDE のプロパティ	設定値
Name	g_timer_agt_0
Channel	0
Mode	Periodic
Period Value	100
Period Unit	Microseconds
Auto Start	False
AGT0 Output Enabled	True
Interrupt Priority	Priority 5
Count Source	PCLKB

表 3 g_timer_agt_1 に対応する設定プロパティ (ソースは AGT0 アンダーフロー)

ISDE のプロパティ	設定値
Name	g_timer_agt_1
Channel	1
Mode	Periodic
Period Value	1
Period Unit	Seconds

ISDE のプロパティ	設定値
Auto Start	False
Callback	agt_32b_callback
AGT0 Output Enabled	True
Interrupt Priority	Priority 5
Count Source	AGT0 Underflow

[Pin] タブで、以下の端子を設定します。これらの端子から、タイマ出力の状態を確認することができます。

1. Peripherals > Timer:AGT > AGT0 > Operation Mode: Timer Output
Peripherals > Timer:AGT > AGT0 > AGTO > P102。次に、P102 を開けて移動します。この値を AGT0_AGTO に設定する必要があります。
2. Peripherals > Timer:AGT > AGT0 > Operation Mode:Timer Output
Peripherals > Timer:AGT > AGT1 > Operation Mode:Timer Output
Peripherals > Timer:AGT > AGT1 > AGTO > P205。次に、P205 を開けて移動します。この値を AGT1_AGTO に設定する必要があります。

注: AGT0 と AGT1 で AGTO を解放するために、SPI0 と CTSU0 を無効にする必要があります。アプリケーションプロジェクト全体は、このドキュメントの末尾にある「参考情報」の章に掲載されているリンクにあります。AGT_HAL_MG.c ファイルは、このプロジェクトを ISDE にインポートした後、プロジェクト内に置かれます。ISDE でこのファイルを開き、API の主な使い方を確認するのに有効な説明を参照することができます。

AGT_HAL_MG.c は、タイマの設定に使用する関数のプロトタイプを含んでいます。

Hal_entry.c は、AGT_HAL_MG.c 内の init_timers() 関数を呼び出し、タイマを開き、起動して設定します。

init_timers() で、open_timers() 関数を使用してタイマを開きます。各タイマに対して open API を呼び出します。start_timers()関数は、各タイマに対して start API を呼び出します。次に、init_timers() は以下の処理を行います。

1. g_timer_agt_1 のタイマ期間(timer period)を 2 秒に変更します。したがって、オンボード LED1 は 2 秒ごとにトグルします。AGT_HAL_MG.c 内で定義されている set_timer_period 関数を使用して、この操作を実行することができます。この関数では SSP の API である periodSet を呼び出しています。
2. すべてのタイマのカウンタ値を取得し、デバッグコンソールに出力します。get_timer_value 関数は、counterGet API を呼び出して、この操作を実行します。
3. すべてのタイマから情報を取得し、デバッグコンソールに出力します。get_timer_info 関数は、infoGet API を呼び出して、この操作を実行します。

AGT_HAL_MG.c の最後のセクションは、ユーザコールバック関数です。g_timer_agt_1 が期限切れになった時点で agt_32b_callback を呼び出し、LED1 をトグルします。

注記: この説明は、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。このような経験がない場合は、下記 WEB サイトの FAQ 2000008 「Synergy ソフトウェアパッケージのデバッグコンソールで Printf_使用方法」という記事を参照してください。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。

<https://ja-support.renesas.com/knowledgeBase/17792531>

あるいは、デバッグモードで変数ウォッチ機能(watch variables)を使用して結果を表示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では、ターゲットボードや MCU の必須の操作と物理プロパティ (physical properties)をサポートするために、いくつかの重要なプロパティを設定しています。以下の表に、それらのプロパティと、このプロジェクトにおける設定値を示します。この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。

8. ターゲットアプリケーションに対応する AGT HAL モジュールのカスタマイズ (Customizing the AGT HAL Module for a Target Application)

いくつかの設定項目は通常、アプリケーションプロジェクトで示している値に対し、ユーザが変更を加えます。たとえば、ユーザは [Clock] タブで [PCLKB] を更新する方法により、ADC クロックに関する設定項目を簡単に変更することができます。またユーザは、個別の AGT モジュールの [properties] タブで、AGT クロックのソースを変更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、タイマをリセットしてクローズする関数も備えています。ユーザのアプリケーションに合わせて、これらの関数を使用することができます。

9. AGT HAL 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 AGT HAL Module Application Project)

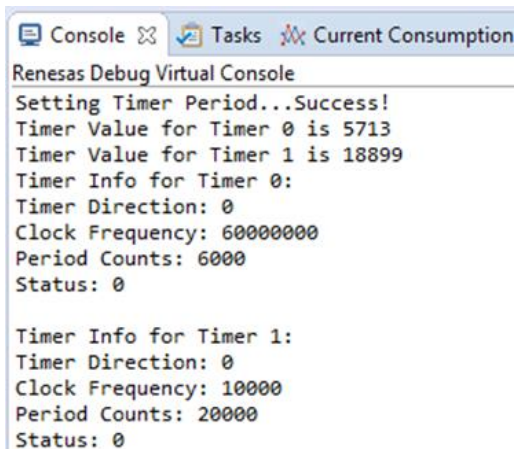
AGT HAL モジュールのアプリケーションプロジェクトを実行し、ターゲットキットでその動作を確認するために、ISDE にこのプロジェクトを単純にインポートし、コンパイルしてデバッグを実行することができます。

新しいプロジェクト内で AGT HAL モジュールアプリケーションを実装するには、ターゲットキットで定義、設定、ファイルの自動生成、コードの追加、コンパイル、デバッグを行う手順に従います。このガイドに示す手順に従うことで SSP での開発プロセスをより実践的に習得するのに役立ちます。

注: Synergy 開発プロセスの基本的な流れを経験したことのあるユーザにとって、以下の手順は十分詳細なものです。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」の章に掲載されている『SSP ユーザーズマニュアル』の「Getting Started with SSP」(SSP 入門) という章を参照してください。

AGT HAL モジュール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. AGT_HAL_MG_AP という名称で SK-S7G2 キットに対応する新規の Renesas Synergy プロジェクトを作成します。
2. AGT ドライバを [HAL] / [common] に追加し、第 7 章で説明した適切なパラメータを使用して、それらのドライバを設定します。
3. **[Generate Project Content]** ボタンをクリックします。
4. 付属のプロジェクトファイル AGT_HAL_MG.c、AGT_HAL_MG.h、および hal_entry.c からコードを追加します。(hal_entry.c を、生成された hal_entry.c ファイルに上書きする形でコピーすることもできます)。
5. micro USB ケーブルで、SK-S7G2 の J19 とホスト PC を接続します。
6. アプリケーションのデバッグを開始します。
7. 出力は、Renesas デバッグ仮想コンソール(Renesas Debug Virtual Console) に表示されます。



```
Console Tasks Current Consumption
Renesas Debug Virtual Console
Setting Timer Period...Success!
Timer Value for Timer 0 is 5713
Timer Value for Timer 1 is 18899
Timer Info for Timer 0:
Timer Direction: 0
Clock Frequency: 60000000
Period Counts: 6000
Status: 0

Timer Info for Timer 1:
Timer Direction: 0
Clock Frequency: 10000
Period Counts: 20000
Status: 0
```

図 2 タイマドライバアプリケーションプロジェクトのサンプル出力

AGT0 (P102) と AGT1 (P205) からの出力を、オシロスコープで表示することもできます。

10. AGT HAL モジュールのまとめ (AGT HAL 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. AGT HAL モジュールの次の手順 (AGT HAL Module Next Steps)

シンプルな AGT HAL モジュールのプロジェクトをマスターした後、より複雑なサンプルを確認することができます。タイマに関する他の例や複雑な使用方法が必要な場合、『GPT HAL モジュールガイド』を参照することができます。

また、他の各種アプリケーションでの AGT の使用方法を試すこともできます。たとえば、AGT ドライバを低消費電力モードのアプリケーションで使用し、アプリケーションをウェイクアップさせる (wake an application up) ことができます。

12. AGT HAL モジュールの参考情報 (AGT HAL Module Reference Information)

『SSP ユーザーズマニュアル』: SSP ディストリビューションパッケージの一部として HTML 形式が入手できるほか、Renesas Synergy™ WEBサイトのSSPページ

<https://www.renesas.com/jp/ja/products/synergy/software/ssp.html> から pdf を入手することもできます。

最新版の r_agt モジュールの参考資料やリソースへのリンクは、以下の Synergy WEBサイトから入手できます。

<https://www.renesas.com/jp/ja/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	2019.06.17		・初版 ・英文版(R11AN0176EU0101、Rev.1.01、2018.Dec.12)の 巻頭と第 7 章以降を翻訳

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）、交通制御（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も、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、データシート等において高信頼性、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て、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/>