

RS-SG2040T/M

SubGiga アナログ入力ボード

ユーザーズマニュアル

2026 年 1 月

第 1.1 版



ラトックシステム株式会社

安全にお使いいただくために	
第1章 はじめに	
(1-1) 製品仕様	1- 2
(1-2) 添付品	1- 3
(1-3) 各部名称と機能	1- 4
第2章 導入手順	
(2-1) 本製品の導入手順	2- 1
(2-2) USB ホストドングルのインストール	2- 2
(2-3) PC のパワーマネージメント機能の設定	2- 5
第3章 API 仕様とサンプルプログラム	
(3-1) API ライブラリ関数一覧と仕様	3- 2
(3-2) サンプルプログラムについて	3-17
第4章 設定ツールについて	
(4-1) 設定ツールの使用手順	4- 1

安全にお使いいただくために

◆警告および注意表示◆

 警告	この表示を無視して誤った取扱いをすると、火災や感電などによる死亡または重傷を負う可能性がある内容を示しています。
 注意	この表示を無視して誤った取扱いをすると、感電やその他の事故により、人が負傷または物的損害が発生する可能性がある内容を示しています。



警告

本製品と添付品(AC アダプター含む)の取扱いについて

- 製品の分解や改造等は、絶対におこなわないでください。
- 無理に曲げる、落とす、傷つける、上に重いものを載せることはおこなわないでください。
- 製品が水・薬品・油等の液体によって濡れた場合、ショートによる火災や感電の恐れがあるため使用しないでください。
- 煙が出る、異臭や音がするなどの異常が発生したときは、ただちに電源を切り、すべての接続ケーブルを抜いたあと、弊社サポートセンターに連絡してください。

AC アダプターの取扱いについて

- 指定の製品以外に使用しないでください。
- 指定された電源電圧以外での使用やタコ足配線はしないでください。
- 雷が鳴り出したら本製品に触れないでください。
- 長時間使用しないときはコンセントから抜いておいてください。
- コンセントから抜くときはコード部を引っ張らないでください。



注意

- 本製品は電子機器ですので、静電気を与えないでください。
- モーターなどのノイズが発生する機器の近くでは誤動作することがあります。必ず離してご使用ください。
- 高温多湿の場所、温度差の激しい場所、チリやほこりの多い場所、振動や衝撃の加わる場所、強い磁気を帯びたものの近くでの使用・保管は避けてください。
- 本製品は、医療機器、原子力機器、航空宇宙機器、輸送機器など人命に関わる設備や機器、および高度な信頼性を必要とする設備、機器での使用は意図されておりません。これらの設備、機器制御システムに本製品を使用し、本製品の故障により人身事故、火災事故などが発生した器制御システムに本製品を使用し、本製品の故障により人身事故、火災事故などが発生した場合、いかなる責任も負いかねます。
- 配線を誤ったことによる損失、逸失利益等が発生した場合でも、いかなる責任も負いかねます。

有寿命部品について

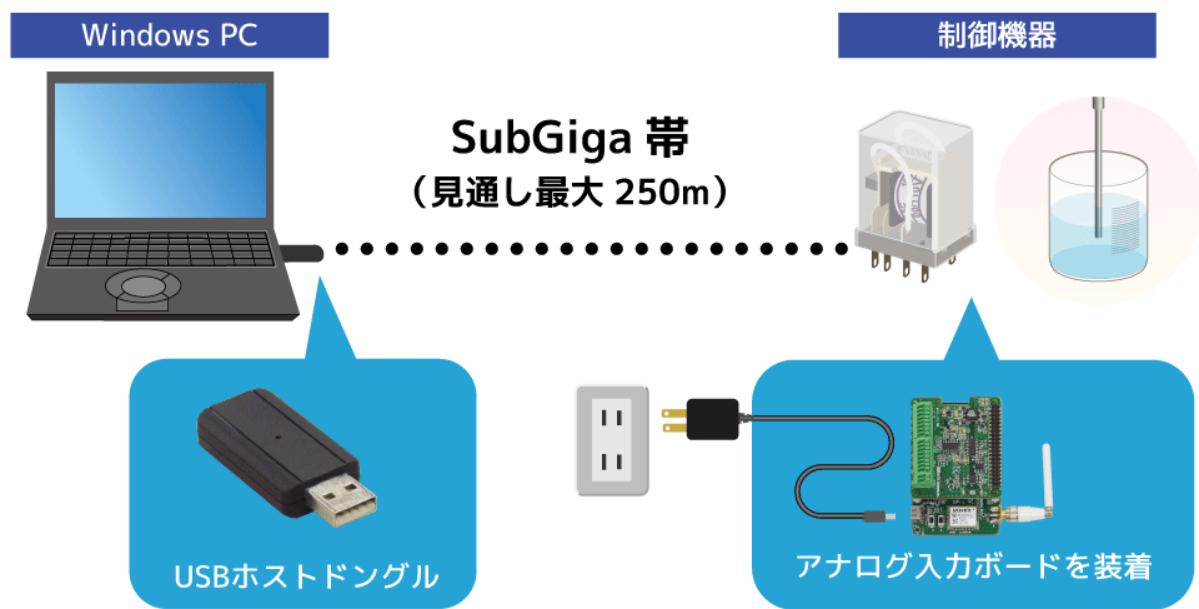
本製品に添付している AC アダプターは有寿命部品です。使用時間の経過に伴って摩耗、劣化等が進行しますので、より長く安定してお使いいただくためには、一定の期間で交換が必要となります。有寿命部品の交換時期の目安は、使用頻度や使用環境等により異なりますが、1 日約 8 時間のご使用で約 5 年です。この期間はあくまでも目安であり、この期間内に故障しないことや無償修理をお約束するものではありません。

また、長時間連続使用等、ご使用状態によっては、この目安の期間よりも早期に部品交換が必要となる場合があります。

- 本紙の内容に関しましては、将来予告なしに変更することがあります。
- 本紙の内容につきましては万全を期して作成しておりますが、万一ご不審な点や誤りなどお気づきの点がございましたらご連絡くださいますようお願いいたします。
- 本製品は日本国内仕様となっており、海外での保守、およびサポートはおこなっておりません。
- 製品改良のため、予告なく外観または仕様の一部を変更することがあります。
- 本製品の保証や修理に関しては、本紙の保証書に記載されております。必ず内容をご確認の上、大切に保管してください。
- 運用の結果につきましては責任を負いかねますので、予めご了承ください。
- 本製品の運用を理由とする損失、逸失利益等の請求につきましては、いかなる責任も負いかねますので、予めご了承ください。
- 本製品を廃棄するときは地方自治体の条例に従ってください。条例の内容については各地方自治体にお問い合わせください。
- 本製品および本紙に記載されている会社名および製品名は、各社商標または登録商標です。ただし本文中には R および TM マークは明記しておりません。

第1章 はじめに

本製品は Windows 環境で SubGiga 無線を介して A/D 変換を行う製品です。WindowsPC 側に USB ホストドングル（標準添付、オプション販売もあり）を接続します。本製品には USB-C ポートより電源供給を行います。（AC アダプター/USB A-C ケーブル標準添付）無線は 920MHz 帯のワイヤレス通信を使用し、通信距離は見通しで最大 250m 程度となります。制御用の API やサンプルプログラムは弊社ホームページよりダウンロードできます。



本製品は SubGiga 入出力ベースボード [RS-SG2000] に A/D 変換インターフェイス [RPi-GP40T] または、[RPi-GP40M] を装着した製品となります。RS-SG2040T の入出力端子は結線が容易な端子台モデル、RS-SG2040M はロック機構を持ち、ハーネスの抜けに耐性の強い MIL コネクタモデルとなります。両製品とも 8ch のアナログ入力ポート、1ch のオープンコレクタ方式のデジタル出力ポートを持ちます。また、MIL コネクタモデルにはデジタル入力(5V/10kΩ プルアップ)を搭載しています。本製品の A/D 変換部、デジタル入出力部に関する詳細につきましては、RPi-GP40T/M のユーザーズマニュアルをご参照ください。

(1-1) 製品仕様

ハードウェア仕様

項目	仕様内容
名称	SubGiga アナログ入力ボード
型番	RS-SG2040T/RS-SG2040M
SubGiga	ARIB STD-T108 準拠 ROHM RS-WSUHA-J11(USB ホストドングル) ROHM BP35C0-J11(基板に搭載されている SubGiga モジュール)
搭載モジュール	ESP32-WROOM-32E-N4
内蔵 I/O Expander	TI TCA9535
SubGiga 通信速度	最大 100kbps
A/D コンバーター	TI ADS8668
アナログ入力分解能	12bit
アナログ入力サンプルレート	500ksps
アナログ入力最大耐圧	±20V
アナログ入力電圧レンジ	±10V, ±5V, ±2.5V, ±1V, ±0.5V, 0-10V, 0-5V, 0-2.5V, 0-1V ※ソフトウェアで各 CH の電圧レンジを切り替え可能 ※入力インピーダンス : 1MΩ (いずれの電圧入力レンジ設定でも一定)
アナログ入力電流レンジ	4-20mA ※シャント抵抗 249Ω、1-5V へ変換し 0-5V レンジで測定 ※シャント抵抗はジャンパ抵抗 RAx で各 ch の有効/無効の切り替えが可能
デジタル出力方式	1ch オープンコレクタ方式 最大耐圧 : 30V (OFF 時) / 最大電流 50mA (ON 時)
絶縁耐圧	2.5kV(アナログ入力/デジタル入出力と GPIO40Pin 間)
電源	+5V/+3.3V (GPIO ポートから供給)
使用コネクタ RS-SG2040T	Phoenix Contact PTSA 0,5/**-2,5-F もしくは互換品 適合電線 : AWG24~20 ストリップ長 9mm
使用コネクタ RS-SG2040M	OMRON XG4A-2034 もしくは互換品 適合コネクタ OMRON XG5N-201-AWG26~22 フラットケーブル用 OMRON XG4M-2030
絶縁耐圧	2.5kV (デジタル入出力と GPIO40 ピン間) ※デジタル入出力の各チャンネルは非絶縁
SubGiga 伝送距離	約 250m

電源電圧	VBUS 5V(USB2.0 Type-C)
消費電流	実測値：約 55mA(アイドル時) 約 86mA(通信時平均) 約 90mA(通信時ピーク)
動作環境	温度：0~40°C 湿度：20~80% (ただし結露しないこと)
基板寸法	RS-SG2040T： L56×W85×T28(入出力端子台上部までの高さ) (mm) (突起部・アンテナを除く) RS-SG2040M： L56×W85×T24(MIL コネクタ上部までの高さ) (mm) (突起部・アンテナを除く) ※Raspberry Pi 4 Model B 用のアクリルベースに組込が 可能です。 当社では KP-SB614(共立電子産業)にて確認。
基板重量	約 52g
原産国	日本
保証期間	1年

ソフトウェア仕様

項目	仕様 内容
インストーラー	本製品を Windows 上で使用するためのドライバー ※ ダウンロード提供
サンプルプログラム	本製品を Windows 上から制御するためのサンプルプログラム (Visual C++/VB/Visual C# ソースコード付き) ※ ダウンロード提供
対応 OS	Windows 11 / 10 ※64 ビット版のみ対応

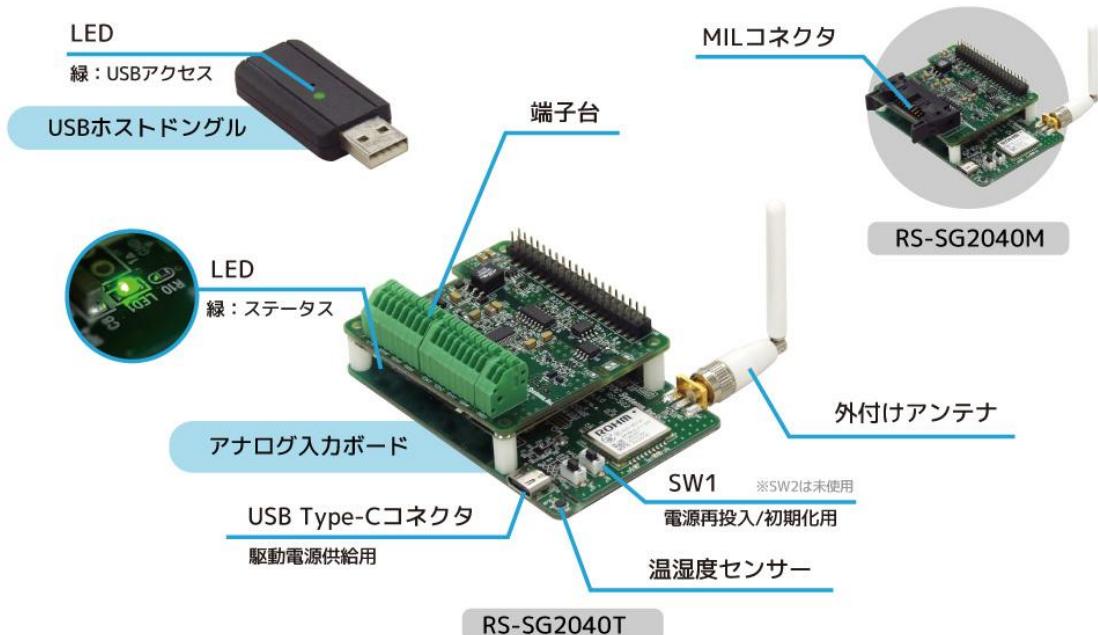
(1-2) 添付品

ご使用前に下記添付品が添付されているかをご確認願います。

- RPi-SG2040(T または M) ボード本体
- USB ホスト ドングル
- AC アダプター
- USB A-C ケーブル
- 保証書

(1-3) 各部名称と機能

※ RS-SG2040T と RS-SG2040M とは、I/O コネクタの違いと RS-SG2040M のみデジタル入力機能があります。



■ ステータス表示用 LEDについて

LED の点灯・点滅の状態で USB ホストドングルとの接続状態を確認できます。

ステータス表示用 LED の状態	製品の状態
500msec 点灯/500msec 消灯の繰り返し	未接続
100msec 点灯/100msec 消灯の繰り返し	USB ホストドングルと接続待ち
点灯	USB ホストドングルと接続完了
50msec 点灯/50msec 消灯の繰り返し	ファームウェア書き込み時 初期化時

■ スイッチ(SW1)について

スイッチ(SW1)の操作により、本基板を再起動、または工場出荷時に初期化することができます。

	スイッチの操作(電源を入れた状態)	機能
SW1	5 秒未満の短押し時	本基板を再起動
	5 秒以上の長押し時(LED が光速点滅になる)	工場出荷時に初期化

■ 温湿度センサーについて

Sensirion 社製温湿度センサー(SHTC3)を搭載しており、本製品の周辺温湿度を測定し、Windows へ通知することができます。

提供しているサンプルプログラム(ソースコード付き)に温湿度を取得する機能が含まれています。

第2章 導入手順

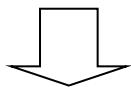
(2-1) 本製品の導入手順

本製品の導入手順について説明します。

1. USB ホストドングルのインストール。
2. 制御機器をアプリケーションで制御。

1. USB ホストドングルのインストール

「(2-2) USB ホストドングルのインストール」をご参照ください。



2. 制御機器をアプリケーションで制御

制御機器を本製品に接続し接続機器の電源を入れてから、本製品の電源を入れます。
(USB-C ポート電源供給)

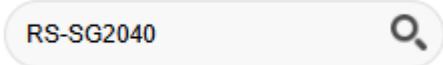
制御機器の制御はサンプルプログラムまたは自作アプリケーションで行います。

※ PC がスリープ/休止状態へ移行すると、正常に通信できなくなる場合もございますので、「(2-3) PC のパワーマネージメント機能の設定」をご参照いただき、事前に PC のパワーマネージメント機能を OFF にしてください。

● ドライバーのダウンロード

弊社ホームページを開き、画面右上部の検索欄に「RS-SG2040」と入力して検索します。

<https://www.ratocsystems.com/>



下記製品情報へのリンクをクリックします。



ラトックシステム

<https://www.ratocsystems.com/embedad/rssg2040> :

[SubGiga アナログ入力ボード RS-SG2040T/SG2040M](#)

表示された製品情報ページの「ダウンロード」をクリックします。



(2-2) USB ホストドングルのインストール

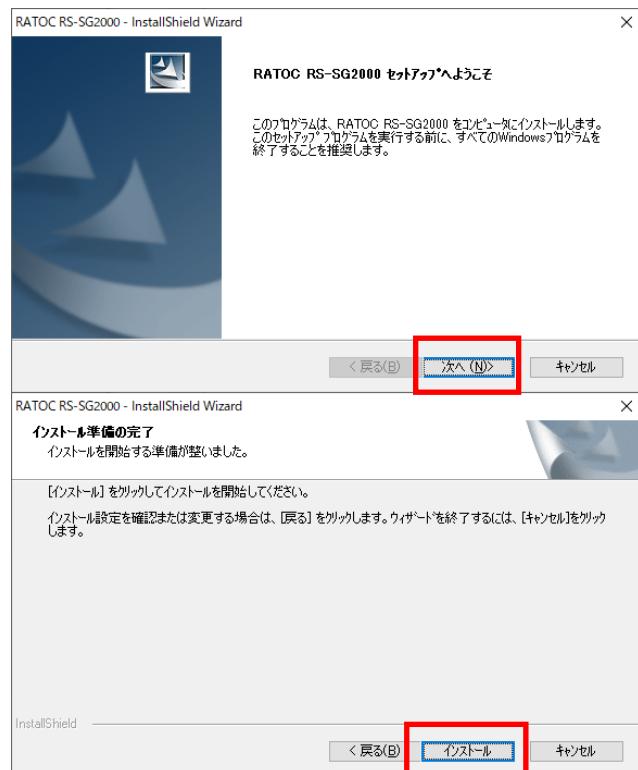
USB ホストドングルのインストールと確認方法について説明します。

USB ホストドングルを PC に接続する前に、ダウンロードしたインストーラーを実行します。

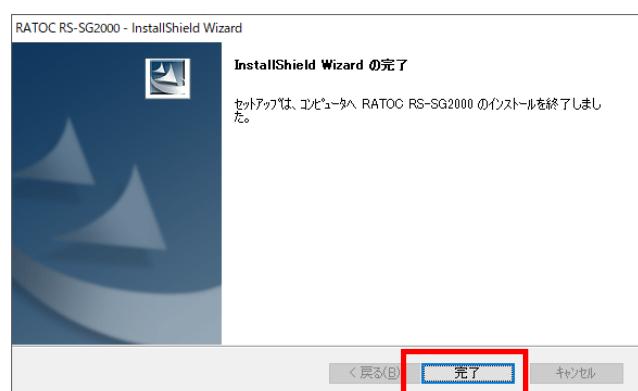
ユーザー アカウント制御画面が表示される場合は「はい」をクリックします。



「次へ」をクリックします。



「インストール」をクリックします。

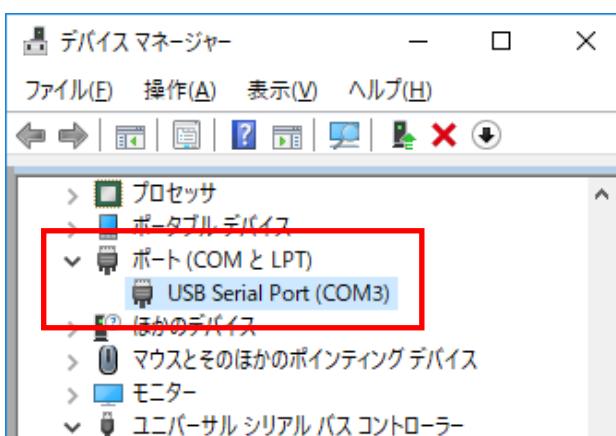


「完了」をクリックします。

以上で、USB ドングルのドライバーセットアップは完了です。

次に、USB ドングルを PC に接続し、正常にインストールされているかを確認します。

[コントロールパネル]-[デバイスマネージャー]を開き、「ポート(COM と LPT)」に「USB Serial Port(COMxx)」が表示されていればインストールは正常に完了しています。
(※ xx の数字は環境により異なります。)



■ アンインストール方法について

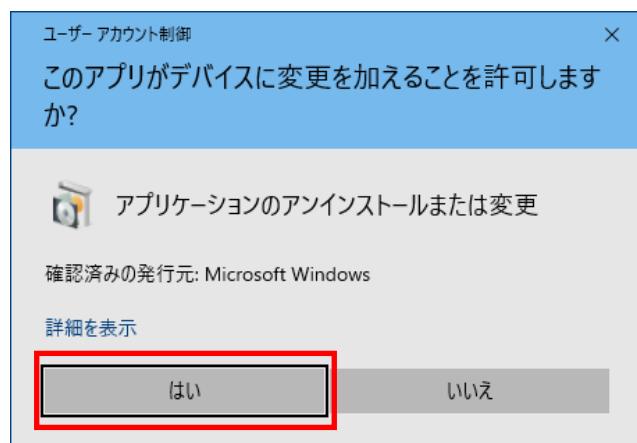
USB ドングルのドライバーを削除する方法について説明します。

[コントロールパネル]-[プログラムと機能]を開き、「RATOC RS-SG2000」を選択し「アンインストール」をクリックします。

ユーザー アカウント制御画面が表示される場合は「はい」をクリックします。

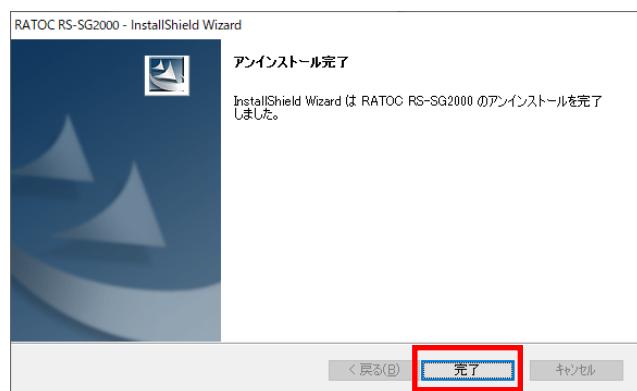
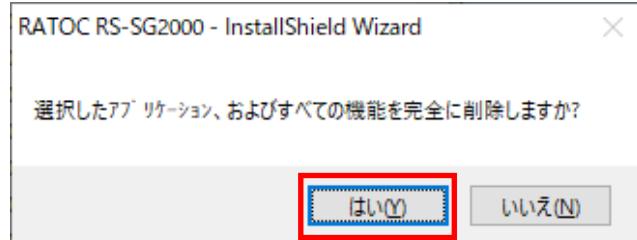


アンインストールを実行する場合は「はい」をクリックします。



「完了」をクリックします。

以上で、USB ドングルのドライバーとユーティリティのアンインストールは完了です。

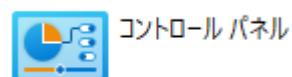
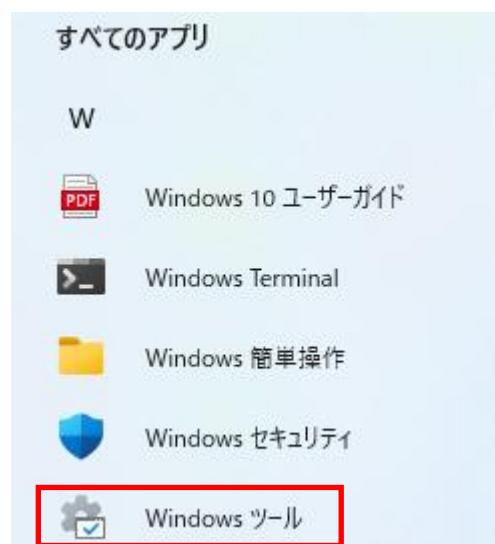


(2-3) PC のパワーマネージメント機能の設定

次の手順でコントロールパネルを起動します。

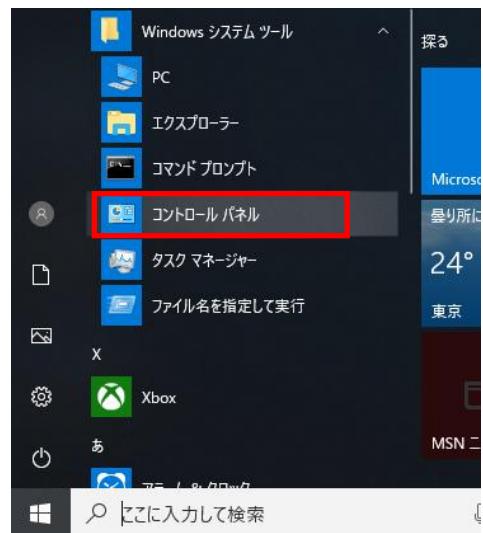
Windows11 の場合

Windows スタートメニューの[すべてのアプリ]-[Windows ツール]より
コントロールパネルを起動します。

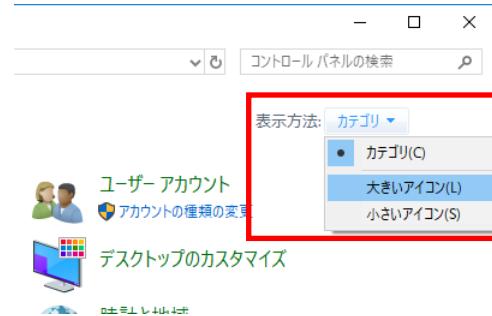


Windows10 の場合

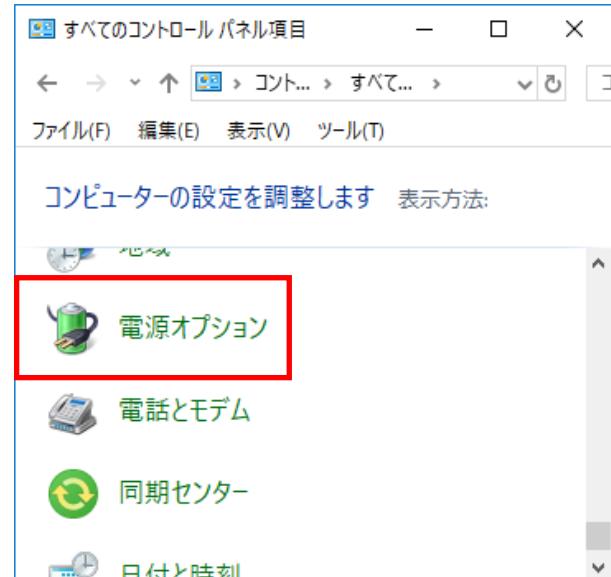
Windows スタートメニューの
[Windows システムツール]より
コントロールパネルを起動します。



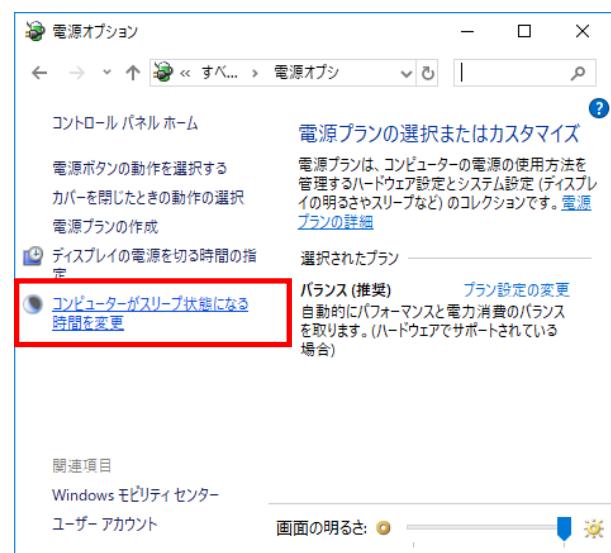
コントロールパネルの表示方法を
「大きいアイコン」に変更します。



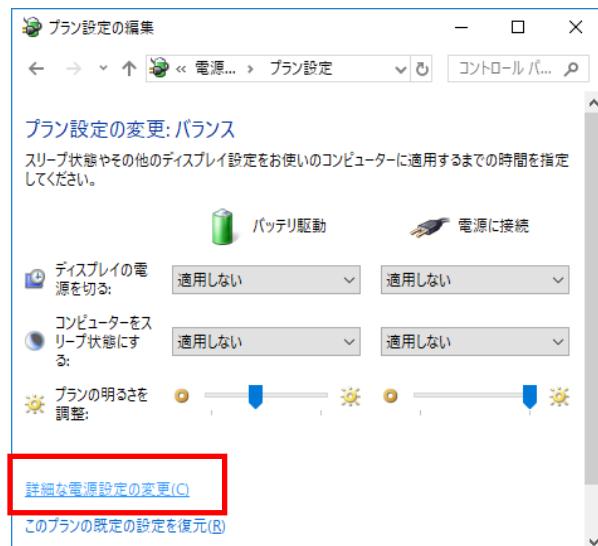
「電源オプション」を起動します。



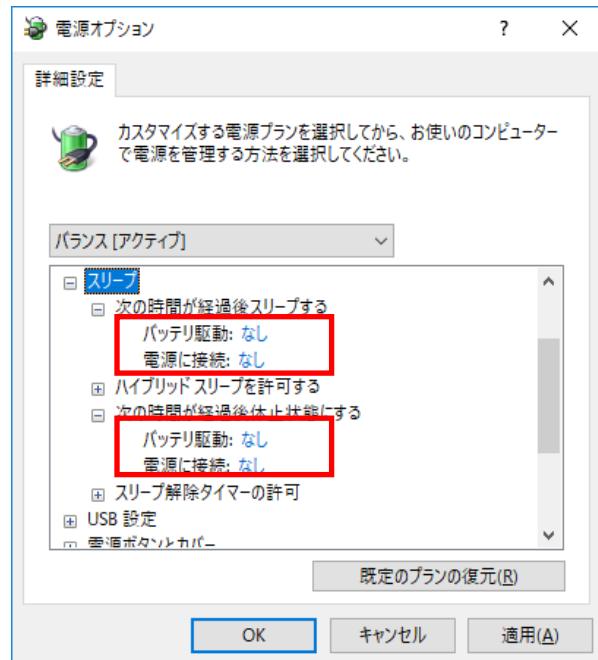
「コンピューターがスリープ状態になる時間を変更」をクリックします。



「詳細な電源設定の変更」をクリックします。



[スリープ]項目にある
[次の時間が経過後スリープする]
[次の時間が経過後休止状態にする]
の時間を「なし」に設定します。
(「0」に設定すると「なし」となります)



第3章 API 関数仕様とサンプルプログラム

本 API 関数は、RS-SG2040 を Windows 上で使用するためのソフトウェア開発を支援するライブラリソフトウェアです。

API 関数を使用することで、本製品に接続されたデバイスの制御を自作のアプリケーションプログラムに組み込むことが可能となります。

VC++で作成したライブラリ(RSSG2000.lib)と DLL(RSSG2000.dll)、

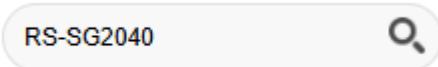
VC++用のヘッダーファイル(RSSG2000.h)、VB 用の宣言ファイル(RSSG2000.vb)、

C#用の宣言ファイル(RSSG2000.cs)を提供しています。

● サンプルプログラムのダウンロード

弊社ホームページを開き、画面右上部の検索欄に「RS-SG2040」と入力して検索します。

<https://www.ratocsystems.com/>



下記製品情報へのリンクをクリックします。

 ラトックシステム
<https://www.ratocsystems.com/embedad/rssg2040> :

SubGiga アナログ入力ボード RS-SG2040T/SG2040M

表示された製品情報ページの「ダウンロード」をクリックします。

ダウンロード ↓

(3-1) API ライブライリ関数一覧と仕様

ライブラリ関数一覧

関数名	機能概要
RS-SG2000(SubGiga 入出力ベースボード) に対する制御関数	
RSSG2000Open	USB ホストドングルの COM ポートオープンと接続
RSSG2000Close	USB ホストドングルの COM ポートクローズ
RSSG2000GetMacAdrs	接続している RS-SG2000 の MAC アドレスを取得
RSSG2000RegistMacAdrs	指定した MAC アドレスの登録
RSSG2000DeleteMacAdrs	登録した MAC アドレスの削除
RSSG2000Start	登録した RS-SG2000 との通信を開始
RSSG2000Stop	RS-SG2000 との通信を停止
RSSG2000Reset	RS-SG2000 との通信情報をリセット
RSSG2000GetVersion	RS-SG2000 のファームウェアバージョンを取得
RSSG2000GetRSSI	RS-SG2000 との RSSI 値を取得
RSSG2000GetTemp	RS-SG2000 に搭載されているセンサーから温湿度を取得
RSSG2000GetAlarmData	アラーム通知を受信した時のデータを取得(VB 用)
RSSG2000GetTimerData	タイマー通知を受信した時のデータを取得(VB 用)
RSSG2000GetErrorData	エラー通知を受信した時のデータを取得(VB 用)
RPi-GP40(アナログ入力 HAT ボード) に対する制御関数	
RPiGP40Open	RPi-GP40 の初期化、CallBack 関数の登録
RPiGP40Close	RPi-GP40 の CallBack 関数の解除
RPiGP40SetSetting	RPi-GP40 の設定書き
RPiGP40GetSetting	RPi-GP40 の設定読み
RPiGP40OneShot	RPi-GP40 のチャンネル毎のデータ取得(一回)
RPiGP40TimerData	RPi-GP40 のチャンネル毎のデータ取得(連続)
RPiGP40OutPort	RPi-GP40 の DIO 出力
RPiGP40InPort	RPi-GP40 の DIO 入力

RS-SG2000(SubGiga 入出力ベースボード) に対する制御関数

関数	VC ▶	HANDLE RSSG2000Open(UCHAR ComNo, UCHAR bChannel, UCHAR MaxSG2000Count);
	VB ▶	Function RSSG2000Open(ByVal ComNo As Byte, ByVal bChannel As Byte, ByVal MaxSG2000Count As Byte) As UInt32
	C#▶	UInt32 RSSG2000Open(Byte ComNo, Byte bChannel, Byte MaxSG2000Count);
機能	USB ホストドングルに割り当てられた COM ポートをオープンし、指定したチャンネル番号と台数分の RS-SG2000 と接続する。	
引数	ComNo Windows で割り当てられた USB ホストドングルの COM ポート番号 bChannel 設定するチャンネル番号(4 ~ 17) MaxSG2000Count 接続する RS-SG2000 の台数	
戻値	-1 以外 : 成功、オープンした USB ホストドングルのハンドル -1 : エラー	
複数台の RS-SG2000 と接続する場合は、接続するすべての RS-SG2000 のチャンネル番号を本 API で指定するチャンネル番号と同じ値に設定しておく必要があります。 30 秒経過しても指定台数分の RS-SG2000 が見つからない場合はエラーとなります。		

関数	VC ▶	DWORD RSSG2000Close(HANDLE hCom);
	VB ▶	Function RSSG2000Close(ByVal hCom As UInt32) As UInt32
	C#▶	UInt32 RSSG2000Close(UInt32 hCom);
機能	USB ホストドングルに割り当てられた COM ポートをクローズする。	
引数	hCom オープンした USB ホストドングルのハンドル	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RSSG2000GetMacAdrs(HANDLE hCom, PUCHAR pMacAdrs);
	VB >	Function RSSG2000GetMacAdrs(ByVal hCom As UInt32, ByRef pMacAdrs As Byte) As UInt32
	C#>	UInt32 RSSG2000GetMacAdrs(UInt32 hCom, Byte[] pMacAdrs);
機能	RSSG2000Open で接続した RS-SG2000 の MAC アドレスを取得する。	
引数	hCom オープンした USB ホストドングルのハンドル pMacAdrs RS-SG2000 の MAC アドレスを格納するバッファ	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RSSG2000RegistMacAdrs(HANDLE hCom, PUCHAR pMacAdrs);
	VB >	Function RSSG2000RegistMacAdrs(ByVal hCom As UInt32, ByRef pMacAdrs As Byte) As UInt32
	C#>	UInt32 RSSG2000RegistMacAdrs(UInt32 hCom, Byte[] pMacAdrs);
機能	RSSG2000GetMacAdrs で取得した RS-SG2000 の MAC アドレスを登録する。	
引数	hCom オープンした USB ホストドングルのハンドル pMacAdrs 登録する RS-SG2000 の MAC アドレス	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RSSG2000DeleteMacAdrs(HANDLE hCom, PUCHAR pMacAdrs);
	VB >	Function RSSG2000DeleteMacAdrs(ByVal hCom As UInt32, ByRef pMacAdrs As Byte) As UInt32
	C#>	UInt32 RSSG2000DeleteMacAdrs(UInt32 hCom, Byte[] pMacAdrs);
機能	RSSG2000RegistMacAdrs で登録した RS-SG2000 の MAC アドレスを削除する。	
引数	hCom オープンした USB ホストドングルのハンドル pMacAdrs 削除する RS-SG2000 の MAC アドレス	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RSSG2000Start(HANDLE hCom);
	VB >	Function RSSG2000Start(ByVal hCom As UInt32) As UInt32
	C#>	UInt32 RSSG2000Start(UInt32 hCom);
機能	RSSG2000RegistMacAdrs で登録した RS-SG2000 との通信を開始する。	
引数	hCom オープンした USB ホストドングルのハンドル	
戻値	0 : 成功 0 以外 : エラー	
30 秒経過しても指定台数分の RS-SG2000 と通信が開始されない場合はエラーとなります。		

関数	VC >	DWORD RSSG2000Stop(HANDLE hCom);
	VB >	Function RSSG2000Stop (ByVal hCom As UInt32) As UInt32
	C#>	UInt32 RSSG2000Stop(UInt32 hCom);
機能	RSSG200020Start で開始した RS-SG2000 との通信を停止する。	
引数	hCom オープンした USB ホストドングルのハンドル	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RSSG2000Reset(HANDLE hCom);
	VB >	Function RSSG2000Reset (ByVal hCom As UInt32) As UInt32
	C#>	UInt32 RSSG2000Reset(UInt32 hCom);
機能	RS-SG2000 との通信情報をリセットする。	
引数	hCom オープンした USB ホストドングルのハンドル	
戻値	0 : 成功 0 以外 : エラー	

他の関数でエラーが発生する場合に呼び出します。

リセット後は RSSG2000Open からの手順を実行する必要があります。

関数	VC >	DWORD RSSG2000GetVersion(HANDLE hCom, PUCHAR pMacAdrs, PUCHAR pVersion);
	VB >	Function RSSG2000GetVersion(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByRef pVersion As Byte) As UInt32
	C#>	UInt32 RSSG2000GetVersion(UInt32 hCom, Byte[] pMacAdrs, Byte[] pVersion);
機能	接続されている RS-SG2000 のファームウェアバージョンを取得する。	
引数	hCom オープンした USB ホストドングルのハンドル pMacAdrs RS-SG2000 の MAC アドレス pVersion RS-SG2000 のファームウェアバージョン	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RSSG2000GetRSSI(HANDLE hCom, PUCHAR pMacAdrs, PCHAR pRSSI);
	VB >	Function RSSG2000GetRSSI (ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByRef pRSSI As SByte) As UInt32
	C#>	UInt32 RSSG2000GetRSSI(UInt32 hCom, Byte[] pMacAdrs, SByte* pRSSI);
機能	接続中の RS-SG2000 の RSSI 値を取得する。	
引数	hCom オープンした USB ホストドングルのハンドル pMacAdrs RS-SG2000 の MAC アドレス pRSSI RSSI 値	
戻値	0 : 成功 0 以外 : エラー	
RSSI 値は、0x98～0xDE で返され、-104～-34 dBm となります。		

関数	VC >	DWORD RSSG2000GetTemp(HANDLE hCom, PUCHAR pMacAdrs, double* pTemp, double* pHumidity);
	VB >	Function RSSG2000GetTemp (ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByRef pTemp As Double, ByRef pHumidity As Double) As UInt32
	C#>	UInt32 RSSG2000GetTemp(UInt32 hCom, Byte[] pMacAdrs, Double* pTemp, Double* pHumidity);
機能	接続中の RS-SG2000 に搭載された温湿度センサーの値を取得する。	
引数	hCom オープンした USB ホストドングルのハンドル pMacAdrs RS-SG2000 の MAC アドレス pTemp 温度(°C) pHumidity 湿度(%)	
戻値	0 : 成功 0 以外 : エラー	

RS-SG2040/VB 用

関数	VB >	Function RSSG2000GetAlarmData(ByVal pAlarmData As Byte()) As UInt32
機能	RS-SG2000 からのアラーム通知を受信した時のデータを取得する。(VB 用)	
引数	pAlarmData 通知した RS-SG2000 の MAC アドレスと 50 件分のデータを返す	
戻値	0 : エラー 0 以外 : 受信したバイト数	
	RPiGP40Open でのコールバック関数 AlarmCallBackFunc にてアラーム通知設定を行います	

関数	VB >	Function RSSG2000GetTimerData(ByVal pTimerData As Byte()) As UInt32
機能	RS-SG2000 からのタイマー通知を受信した時のデータを取得する。(VB 用)	
引数	pTimerData 通知した RS-SG2000 の MAC アドレスと 1 件分のデータを返す	
戻値	0 : エラー 0 以外 : 受信したバイト数	
	RPiGP40Open でのコールバック関数 TimerDataCallBackFunc にてタイマー通知設定を行います	

VB 用

関数	VB ➤	Function RSSG2000GetErrorHandlerData(ByVal pErrorHandler As Byte0) As UInt32
機能		接続中の RS-SG2000 から通知されたエラーコードを取得する。
引数	pErrorHandler	通知した RS-SG2000 の MAC アドレスとエラーコードを返す
戻値	0 : 成功 0 以外 : エラー	
		RPiGP40Open でのコールバック関数 ErrorCallBackFunc にてエラー通知設定を行います

RPi-GP40(アナログ入力 HAT ボード)に対する制御関数

関数	VC ➤	DWORD RPiGP40Open(HANDLE hCom, PUCHAR pMacAdrs, TIMERDATA_PROC pTimerDataFunc, ALARM_PROC pAlarmFunc, ERROR_PROC pErrorFunc);
	VB ➤	Function RPiGP40Open(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByVal pTimerDataFunc As TimerDataCallback, ByVal pAlarmFunc As AlarmCallback, ByVal pErrorFunc As ErrorCallback) As UInt32
	C# ➤	UInt32 RPiGP40Open(UInt32 hCom, Byte[] pMacAdrs, TimerDataCallback pTimerDataFunc, AlarmCallback pAlarmFunc, ErrorCallback pErrorFunc);
機能	RPi-GP40 の初期化と CallBack 関数の登録をする。	
引数	<p>hCom オープンした USB ホストドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>pTimerDataFunc チャンネル毎の連続データ取得時に呼び出される関数 VOID CALLBACK TimerDataCallBackFunc(P UCHAR pData); pData : MAC アドレス(8byte) + 各チャンネルでの A/D 変換データ(16byte)が格納 ※ Ch0・Ch7のデータ順に保存されており、各データは入力電圧を12bitデータに変換して 2byteのデータとして保存される</p> <p>pAlarmFunc アラーム通知時に呼び出される関数 VOID CALLBACK AlarmCallBackFunc (P UCHAR pData); pData : MACアドレス(8byte) ステータス(1byte 0:正常) 閾値を超えたチャンネルをbit指定(1byte) *1 有効データ件数(1byte 1-50) 予約(5byte) データ(800byte) *2</p> <p>*1 「閾値を超えたチャンネル」について Ch0で閾値を超えた場合、閾値超えたチャンネルの1byteデータは0x01 Ch7で閾値を超えた場合、閾値超えたチャンネルの1byteデータは0x80 のように閾値を超えたChがbit指定される</p>	

	<p>*2 「データ」について</p> <p>Ch0・・Ch7のデータ順に保存されており、各データは入力電圧を12bitデータに変換して2byteのデータとして保存される</p> <p>データは最大50件まで通知されるが、サンプリングが始まつてから閾値を超えるまでの時間が短い場合、RPi-GP40に保存されていない場合もあり、その場合の有効データ件数も通知される</p> <p>※ VBでデータ取得するには、RSSG2000GetAlarmDataを呼び出す必要がある</p> <p>pErrorFunc</p> <p>エラー通知時に呼び出される関数</p> <pre>VOID CALLBACK ErrorCallBackFunc (PUCHAR pData);</pre> <p>pData : MACアドレス(8byte) + エラーコード(1byte)が格納</p>
戻値	<p>0 : 成功</p> <p>0以外 : エラー</p>

本API呼び出し時に、RS-SG2000からRPi-GP40への電源供給がONになります。

関数	<p>VC ></p> <pre>DWORD RPiGP40Close(HANDLE hCom, PUCHAR pMacAdrs);</pre> <p>VB ></p> <pre>Function RPiGP40Close(ByVal hCom As UInt32, ByRef pMacAdrs As Byte) As UInt32</pre> <p>C#></p> <pre>UInt32 RPiGP40Close(UInt32 hCom, Byte[] pMacAdrs);</pre>
機能	登録した RPi-GP40 のCallBack関数を解除する。
引数	<p>hCom</p> <p>オープンしたUSBホストドングルのハンドル</p> <p>pMacAdrs</p> <p>制御するRS-SG2000のMACアドレス</p>
戻値	<p>0 : 成功</p> <p>0以外 : エラー</p>

本API呼び出し時に、RS-SG2000からRPi-GP40への電源供給をOFFにして連続取得を停止します。

関数	VC >	DWORD RPiGP40SetSetting(HANDLE hCom, PUCHAR pMacAdrs, GP40Setting SettingData);
	VB >	Function RPiGP40SetSetting(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByVal SettingData As GP40Setting) As UInt32
	C#>	UInt32 RPiGP40SetSetting(UInt32 hCom, Byte[] pMacAdrs, GP40Setting SettingData);
機能	RPi-GP40 の設定を書き込む	
引数	<p>hCom オープンした USB ホスト ドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>SettingData 書き込む RPi-GP40 の設定情報 ※GP40Setting 構造体については下記参照</p>	
戻値	0 : 成功 0 以外 : エラー	

```
typedef struct _GP40Setting {
    BYTE      ChannelEnable;
    BYTE      AlarmEnable;
    WORD     SampTime;
    BYTE      Trigger;
    BYTE      Reserved;
    DWORD    ChannelRange;
    WORD     ChUpperThreshold[8];
    WORD     ChLowerThreshold[8];
} GP40Setting, *PGP40Setting;
```

ChannelEnable

Bit0-Bit7 が Ch0-Ch7 に対応。サンプリングの有効./無効設定 0:無効 1:有効

AlarmEnable

アラーム通知の有効/無効設定 0x00:無効 0x10:有効

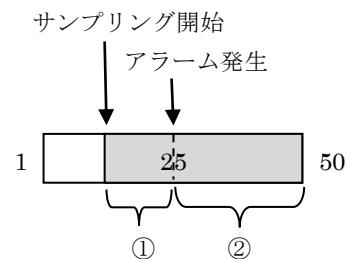
SampTime

サンプリング間隔(1-1000ms) ※ 8Ch 分のデータ(16byte)の取得間隔

Trigger

トリガー ポイント(1-50) ※デフォルト:25 (右図は設定値 25 での例)

- ① アラーム発生時、トリガー ポイント以前のサンプリングデータを取得
- ② トリガー ポイント以降、50 件になるまでのサンプリングデータを取得
- ③ アラーム通知を行う



Reserved

予約

ChannelRange

レンジ設定。4bit 毎に各 Ch のレンジ設定を行う。(上位が Ch7)

0b0000: ±10V

0b0001: ±5V

0b0010: ±2.5V

0b0011: ±1V

0b1011: ±0.5V

0b0101: 0~10V

0b0110: 0~5V

0b0111: 0~2.5V

0b1111: 0~1V

ChUpperThreshold

上限の閾値 ※ 以下の計算式の結果を設定する

レンジ	dbRes(分解能)	ChUpperThreshold (dbData は実際のアナログデータ閾値)
±10V	$(10.24 - (-10.24)) / 4095.0$	$(\text{dbData} + 10.24) / \text{dbRes}$
±5V	$(5.12 - (-5.12)) / 4095.0$	$(\text{dbData} + 5.12) / \text{dbRes}$
±2.5V	$(2.56 - (-2.56)) / 4095.0$	$(\text{dbData} + 2.56) / \text{dbRes}$
±1V	$(1.28 - (-1.28)) / 4095.0$	$(\text{dbData} + 1.28) / \text{dbRes}$
±0.5V	$(0.64 - (-0.64)) / 4095.0$	$(\text{dbData} + 0.64) / \text{dbRes}$
0~10V	$(10.24 - 0) / 4095.0$	$\text{dbData} / \text{dbRes}$
0~5V	$(5.12 - 0) / 4095.0$	$\text{dbData} / \text{dbRes}$
0~2.5V	$(2.56 - 0) / 4095.0$	$\text{dbData} / \text{dbRes}$
0~1V	$(1.28 - 0) / 4095.0$	$\text{dbData} / \text{dbRes}$

ChLowerThreshold

下限の閾値 ※ ChUpperThreshold と同様に上表の結果を設定する

関数	VC ▶	DWORD RPiGP40GetSetting(HANDLE hCom, PUCHAR pMacAdrs, PGP40Setting pSettingData);
	VB ▶	Function RPiGP40GetSetting(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByRef pSettingData As GP40Setting) As UInt32
	C#▶	UInt32 RPiGP40GetSetting(UInt32 hCom, Byte[] pMacAdrs, GP40Setting* pSettingData);
機能	RPi-GP40 の設定値を読み込む。	
引数	<p>hCom オープンした USB ホストドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>pSettingData 読み込んだ RPi-GP40 の設定情報 ※GP40Setting 構造体については RPiGP40SetSetting 参照</p>	
戻値	0 : 成功 0 以外 : エラー	

関数	VC ▶	DWORD RPiGP40OneShot(HANDLE hCom, PUCHAR pMacAdrs, PGP40ChannelData pChData);
	VB ▶	Function RPiGP40OneShot(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByRef pChData As RsUshortData) As UInt32
	C#▶	UInt32 RPiGP40OneShot(UInt32 hCom, Byte[] pMacAdrs, RsUshortData* pChData);
機能	RPi-GP40 のチャンネル毎の A/D 変換データを 1 回取得する。	
引数	<p>hCom オープンした USB ホストドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>pChData 各チャンネルでの A/D 変換データ(16byte)が格納 ※ Ch0 ・ Ch7 のデータ順に保存されており、各データは入力電圧を12bitデータに変換して 2byte のデータとして保存される</p> <p>GP40ChannelData 構造体について typedef struct _GP40ChannelData { WORD ChData [8]; } GP40ChannelData, *PGP40ChannelData; ChData : Ch7-Ch0 のデータ(12bit データ)</p>	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RPiGP40TimerData(HANDLE hCom, PUCHAR pMacAdrs, UCHAR bStart, UCHAR bTime);
	VB >	Function RPiGP40TimerData(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByVal bStart As Byte, ByVal bTime As Byte) As UInt32
	C#>	UInt32 RPiGP40TimerData(UInt32 hCom, Byte[] pMacAdrs, Byte bStart, Byte bTime);
機能	RPi-GP40 での A/D 変換を行う時間間隔と開始/停止を設定する。	
引数	<p>hCom オープンした USB ホストドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>bStart 連続取得 0:停止 1:開始</p> <p>bTime 連続取得時間 1~255(分)</p>	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RPiGP40OutPort(HANDLE hCom, PUCHAR pMacAdrs, UCHAR bData);
	VB >	Function RPiGP40OutPort(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByVal bData As Byte) As UInt32
	C#>	UInt32 RPiGP40OutPort(UInt32 hCom, Byte[] pMacAdrs, Byte bData);
機能	RPi-GP40 の DIO ポートから出力する。	
引数	<p>hCom オープンした USB ホストドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>bData 0x00 High 出力 0x01 Low 出力 ※オープンコレクタのため 0x00 で High 出力となる</p>	
戻値	0 : 成功 0 以外 : エラー	

関数	VC >	DWORD RPiGP40InPort(HANDLE hCom, PUCHAR pMacAdrs, PUCHAR pData);
	VB >	Function RPiGP40InPort(ByVal hCom As UInt32, ByRef pMacAdrs As Byte, ByRef pData As Byte) As UInt32
	C#>	UInt32 RPiGP40InPort(UInt32 hCom, Byte[] pMacAdrs, Byte* pData);
機能	RPi-GP40 の DIO ポートに入力されたデータを取得する。	
引数	<p>hCom オープンした USB ホストドングルのハンドル</p> <p>pMacAdrs 制御する RS-SG2000 の MAC アドレス</p> <p>pData 入力されたデータ</p>	
戻値	0 : 成功 0 以外 : エラー	

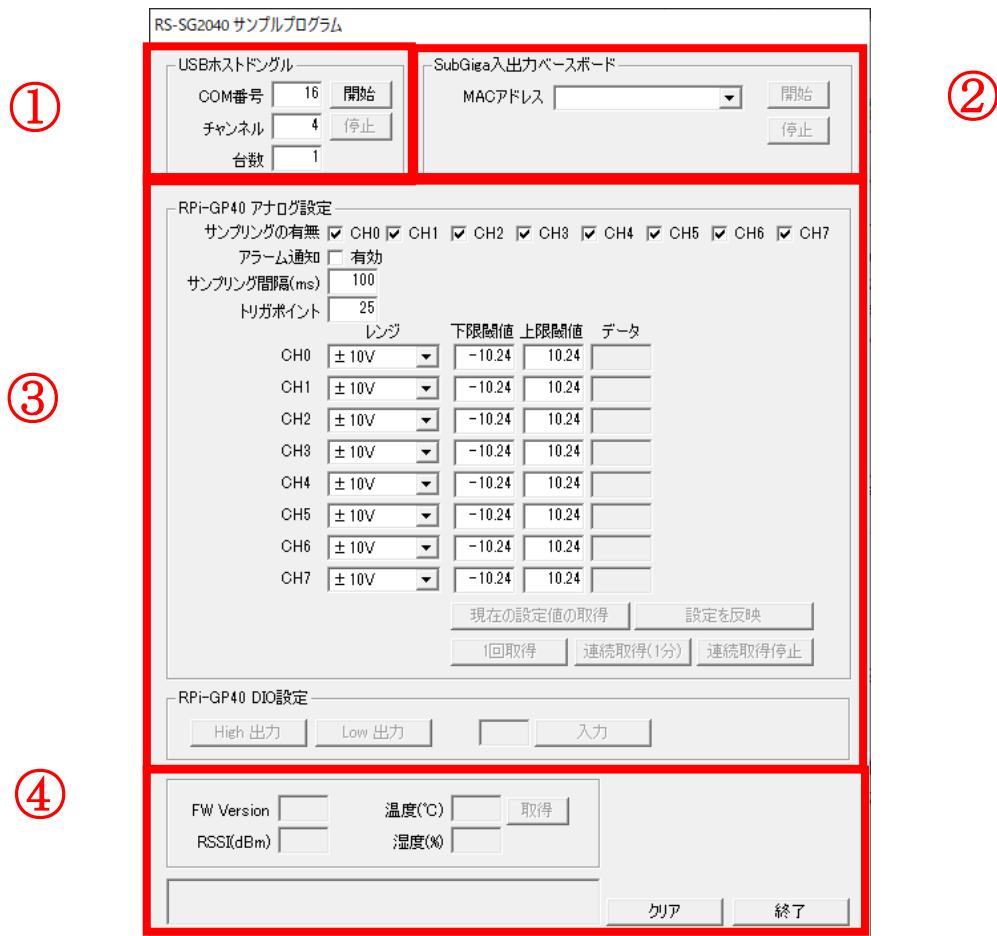
エラーコード一覧

エラーコード	概要
0	正常終了
0x00xx	USB ホストドングルエラー
0x10xx	SubGiga 入出力ベースボードエラー
0x2081	オープンエラー
0x2082	送信エラー
0x2083	受信エラー
0x2085	接続エラー
0x2088	設定エラー
0x2xxx	ライブラリエラー
その他	Windows から返されるエラー

(3-2) サンプルプログラムについて

サンプルプログラムの各機能について、以下の項目に分けて説明します。

- ① SubGiga 入出力ベースボードの検出
- ② SubGiga 入出力ベースボードとの通信開始
- ③ A/D 変換・DIO 制御
- ④ その他データ取得など



① SubGiga 入出力ベースボードの検出

COM番号 -- USBホストドングルに割り当てられたCOMポート番号を指定します。

チャンネル -- 接続するSubGiga入出力ベースボード(出荷時は4)のチャンネル番号を指定します。

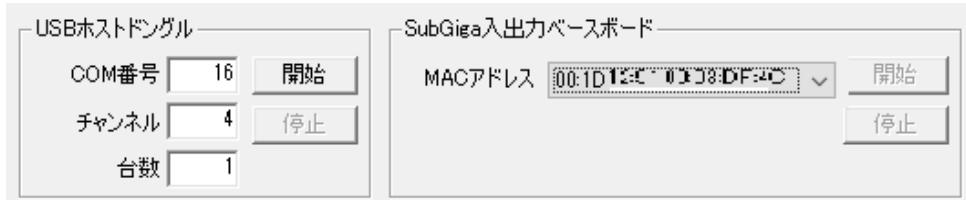
台数 -- USBホストドングルと直接接続するSubGiga入出力ベースボードの台数を指定します。

開始 -- SubGiga入出力ベースボードの検索を開始します。

停止 -- SubGiga入出力ベースボードとの通信を停止します。

SubGiga 入出力ベースボードを検出します。

COM番号/チャンネル/台数を指定し「開始」をクリックし、SubGiga 入出力ベースボードが検出されると「MACアドレス」に SubGiga 入出力ベースボードの MACアドレスが表示されます。



② SubGiga 入出力ベースボードとの通信開始

MACアドレス -- 検出されたSubGiga入出力ベースボードのMACアドレスが表示されます。

開始 -- 表示されているSubGiga入出力ベースボードとの通信を開始します。

停止 -- SubGiga入出力ベースボードとの通信を停止します。

(「開始」をクリックすると、表示されているMACアドレスを持つSubGiga入出力ベースボードと通信可能な状態となります)

③ A/D変換・DIO制御

サンプリングの有無 -- チェックを入れ、設定ボタンを押すとチャンネルが有効になります。
有効になったチャンネルは1回取得ボタン/連続取得で入力電圧を取得できます。

アラーム通知 -- チェックを入れ、設定ボタンを押すとアラーム通知が有効になります。

アラーム通知設定時は常に監視し、下限の閾値を下回るか、上限の閾値を上回った場合にメッセージ欄に「AlarmCallBackFunc」と表示されます。

サンプリング間隔 -- アラーム通知が有効になっているとき、RPi-GP40 内部に A/D 変換値を取得して保存する間隔を設定します。

トリガポイント -- アラーム通知が有効になっているとき、トリガ位置を設定して取得するデータの範囲を設定します。

(RPi-GP40には50件分のA/D変換値を保存できますが、「30」と設定して閾値を超えてトリガがかかった時からその前30件、その後20件が保存され通知されます)

レンジ -- $\pm 10V / \pm 5V / \pm 2.5V / \pm 1V / \pm 0.5V / 0 \sim 10V / 0 \sim 5V / 0 \sim 2.5V / 0 \sim 1V$ から選択します。

下限閾値 -- 指定の値を下回った時にアラーム通知が発生します。

上限閾値 -- 指定の値を上回った時にアラーム通知が発生します。

データ -- 1回取得ボタンをクリックすると現在入力されているデータが表示されます。

現在の設定値の取得 -- 各種設定項目の設定状況が画面に反映されます。

設定を反映 -- 現在表示されている各種設定項目の設定を反映します。

1回取得 -- 各チャンネルの入力データを取得し[データ]に表示します。

連続取得(1分) -- 1分毎に各チャンネルの入力データを取得し④のメッセージ欄に表示します。

連続取得停止 -- [連続取得(1分)]の動作を停止します。

High 出力 -- デジタル出力を High 設定にします。

Low 出力 -- デジタル出力を Low 設定にします。

入力 -- デジタル入力値を取得します。※MILコネクタモデルのみ

④ その他データ取得など

温度(°C)/湿度(%) -- SubGiga 入出力ベースボード上にある温湿度センサーで測定した温度と湿度を表示します。

RSSI(dBm) -- USB ホストドングルと SubGiga 入出力ベースボード間の電波強度を表示します。

FW Version -- SubGiga 入出力ベースボードのファームウェアバージョンを表示します。

取得 -- 温度/湿度/RSSI/FW Version を取得します。

クリア -- メッセージをクリアします

終了 -- 本サンプルプログラムを終了します

第4章 設定ツールについて

設定ツールでは、SubGiga 入出力ベースボードの SubGiga チャンネル番号の変更、動作モード設定(USB ホストドングルと通信/中継器として通信)、ファームウェアアップデートを行なうことができます。

● 設定ツールのダウンロード

弊社ホームページを開き、画面右上部の検索欄に「RS-SG2040」と入力して検索します。

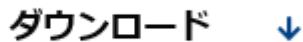
<https://www.ratocsystems.com/>



下記製品情報へのリンクをクリックします。

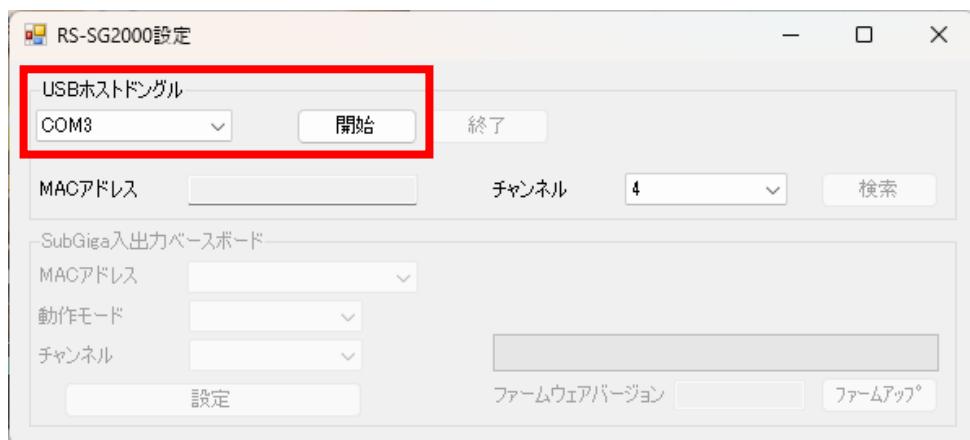


表示された製品情報ページの「ダウンロード」をクリックします。



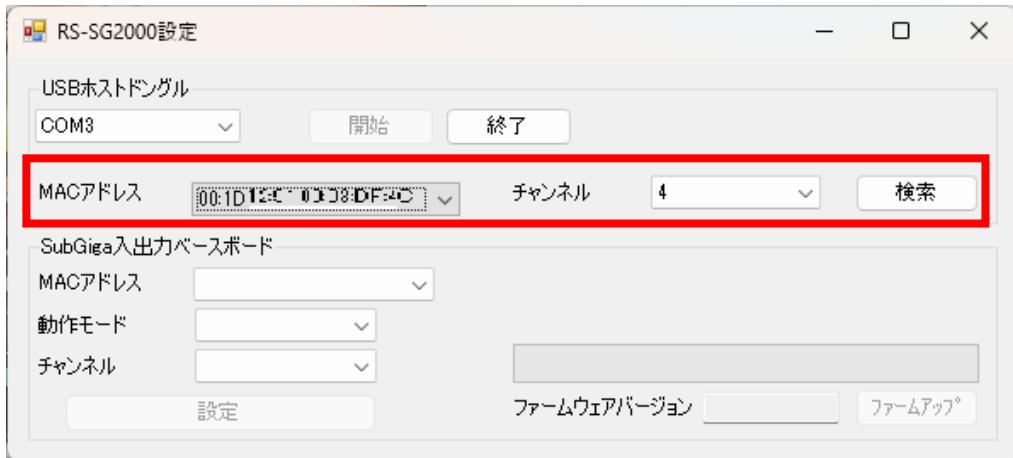
(4-1) 設定ツールの使用手順

USB ホストドングルに割り当たっている COM ポート番号を選択し「開始」をクリックします。

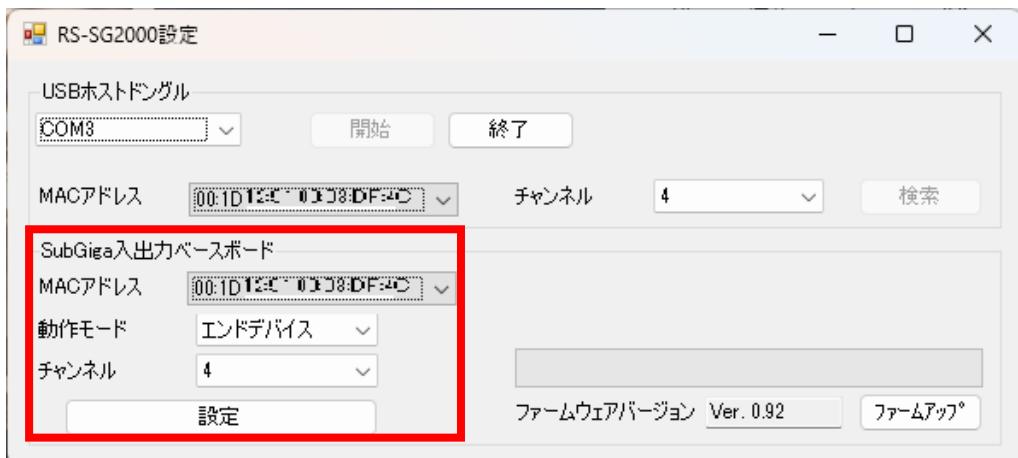


USB ホストドングルの MAC アドレスが表示されます。

設定する SubGiga 入出力ベースボードのチャンネルを指定し「検索」をクリックします。



SubGiga 入出力ベースボードが検出されると MAC アドレス/ファームウェアバージョンが表示されます。



設定を変更する場合は、変更後に「設定」をクリックすることで反映されます。

動作モード

- ・ エンドデバイス -- 制御デバイスと接続する場合に設定します。(出荷時設定)
- ・ 中継器 -- USB ホストドングルと SubGiga 入出力ベースボード(エンドデバイス)間の中継器として使用する場合に設定します。(1 中継のみ対応)

チャンネル -- SubGiga チャンネルを設定します。(設定値 : 4~17)

ファームウェアをアップデートする場合は「ファームアップ」ボタンをクリックしファームウェアファイルを選択します。

製品に対するお問い合わせ

RS-SG2040T/M の技術的なご質問やご相談の窓口を用意していますのでご利用ください。

ラトックシステム株式会社

I&L サポートセンター

〒550-0015

大阪市西区南堀江 1-18-4 Osaka Metro 南堀江ビル 8F
TEL 06-7670-5064

〈サポート受付時間〉

月曜ー金曜（祝祭日は除く）AM 10:00 – PM 1:00
PM 2:00 – PM 5:00

また、インターネットのホームページでも受け付けています。

HomePage ☡ <https://www.ratocsystems.com>



個人情報取り扱いについて
ご連絡いただいた氏名、住所、電話番号、メールアドレス、
その他の個人情報は、お客様への回答など本件に関わる業務のみ
に利用し、他の目的では利用致しません。



<http://www.ratocsystems.com/>

©RATOC Systems, Inc. All rights reserved.