

Smart Analog

R20AN0258JJ0100

Rev.1.00

Smart Analog 開発チュートリアル TSA-IC500 版

2013.07.19

要旨

本アプリケーションノートでは、Renesas が開発・販売している「Smart Analog」を初めてお使い頂く皆様に対して、開発の流れをご理解頂くために各ツールの使い方を実際の例に沿って説明します。

動作確認デバイス

Smart Analog IC 500(RAA730500DFP)、RL78/G1A(R5F10ELEAFB)

目次

1. 概説	2
2. ご用意していただくもの	3
3. 実際に開発するもの	8
4. 全体の流れ	9
5. Renesas VA でシミュレーションをしてみよう！	11
6. センサ基板、LED 基板を作成しよう！	23
7. SA-Designer で AFE レジスタファイルを取り込んで、CubeSuite+でプログラムを作ってみよう！ 28	
8. 実機で動作確認してみよう！	69
9. 最後に	70

1. 概説

本アプリケーションノートは、回路構成可変型の Smart Analog IC 500 と RL78/G1A を搭載した Smart Analog IC 500 評価ボード「TSA-IC500」（以降、「TSA-IC500」）と Smart Analog の開発環境を使ってフォトトランジスタの駆動と LED を制御するプログラムの開発例を説明するものです。

1.1 開発環境

Smart Analog 製品の開発環境は最新版の開発ツールをお使いください。

センサ選定用 Web シミュレータ	Renesas VA Version 2.0.0.0 以降
アナログフロントエンド回路設計ツール	SA-Designer V1.02.00 以降
統合開発環境	CubeSuite+ V2.00.00 以降

2. ご用意していただくもの

本資料の内容に沿って実際に開発を体験頂くには以下のものをご用意ください。

ご用意頂くもの一覧

- 1.1 テセラ・テクノロジー社製 Smart Analog IC 搭載評価ボード「TSA-IC500」
- 1.2 Renesas 製オンチップデバッグエミュレータ「E1」
- 1.3 新日本無線製フォトトランジスタ「NJL7502L」
- 1.4 LED2 個（何色でも構いません）
- 1.5 抵抗 470Ω(1/8W) 2 個
- 1.6 抵抗 1kΩ(1/8W)
- 1.7 コネクタ（ピンソケット メス 2x25 50p） 2 個、（ピンヘッダ オス 2x25 50p） 2 個
- 1.8 基板 2 枚

2.1 テセラ・テクノロジー社製 Smart Analog IC 搭載評価用ボード「TSA-IC500」



本ボードには Smart Analog IC 500（RAA730500DFP）及び RL78/G1A（R5F10E1EAFB）が搭載されております。

「TSA-IC500」のお求め頂き方

法人のお客様 - 弊社代理店へご注文ください。

個人のお客様 - テセラ・テクノロジー社の Web サイトからご購入頂けます。

- テセラ・テクノロジー社：<http://www.tessera.co.jp/tsa-ic500.html>

2.2 Renesas 製オンチップデバッグエミュレータ「E1」



本資料では上記評価ボード「TSA-IC500」に接続して、デバッガ/フラッシュプログラマとして使用します。

「E1」のお求め頂き方

法人のお客様 - 弊社代理店へご注文ください。

個人のお客様 - Web からご注文の場合：RS オンライン、チップワンストップ、他でご購入頂けます。

- RS オンライン： <http://jp.rs-online.com/web/p/processor-microcontroller-development-kits/7330317/>
- チップワンストップ： <http://www.chip1stop.com/dispDetail.do?partId=RNSS-0000126>

また、マルツパーツ館他での店頭でもお求めいただけます。

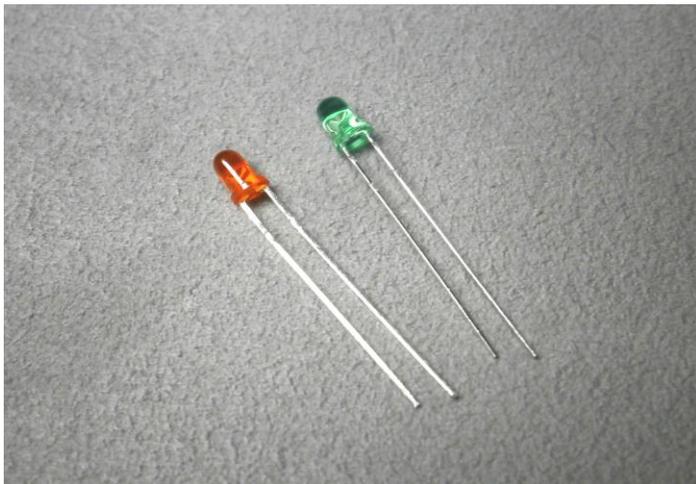
- マルツパーツ館： <http://www.marutsu.co.jp/>

2.3 新日本無線製フォトランジスタ「NJL7502L」



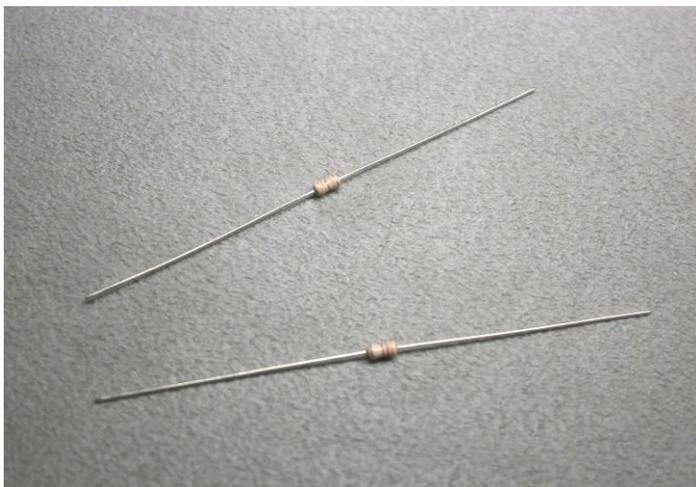
本資料で Smart Analog で駆動するセンサです。パーツ販売をされている各種ショップやオンラインショップ等でお求めいただけます。

2.4 LED2 個 (何色でも構いません)



センサの挙動を受けて RL78/G1A から制御し、センサ動作状況を"Lチカ"表示します。パーツ販売をされている各種ショップやオンラインショップ等でお求めいただけます。なお、色については何色でも構いません。

2.5 抵抗 470Ω(1/8W) 2 個



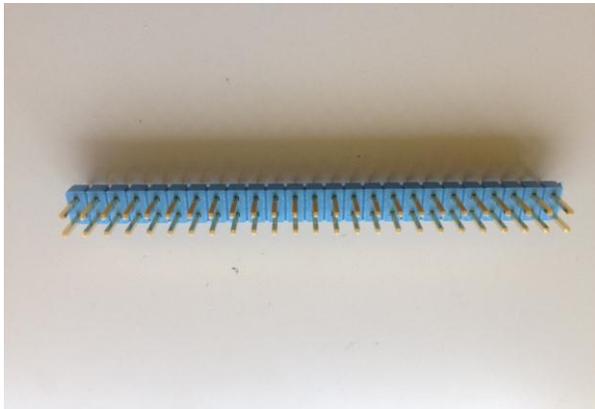
RL78/G1A のポートから LED を駆動する際の電流設定抵抗です。パーツ販売をされている各種ショップやオンラインショップ等でお求めいただけます。

2.6 抵抗 1k Ω (1/8W)

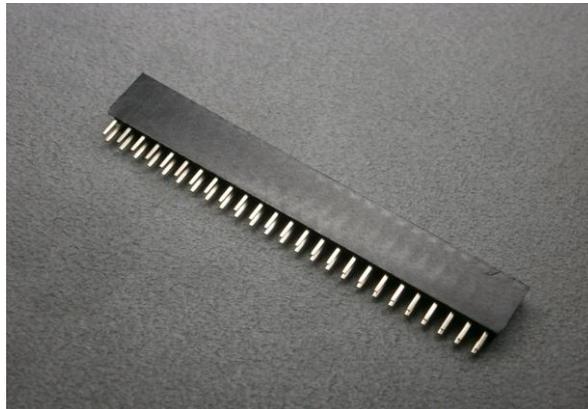


センサを駆動する際の電流設定抵抗です。パーツ販売をされている各種ショップやオンラインショップ等でお求めいただけます。

2.7 コネクタ（ピンソケット メス 2.54mm ピッチ 2x25 50p）2個、（ピンヘッダ オス 2.54mm ピッチ 2x25 50p） 2個



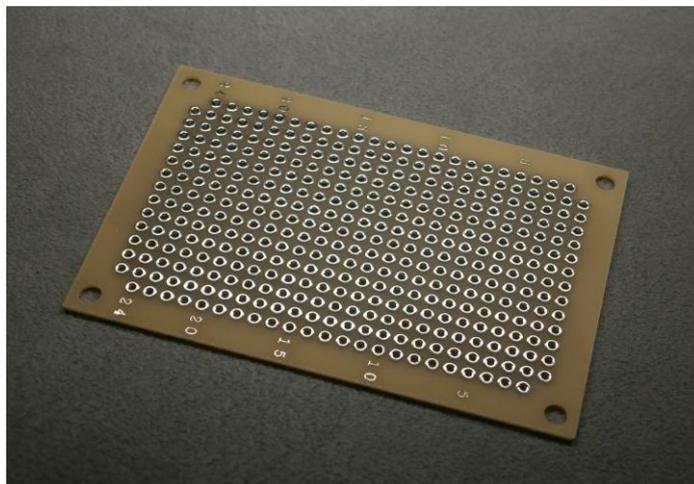
ピンヘッダ オス 2x25 50p



ピンソケット メス 2x25 50p

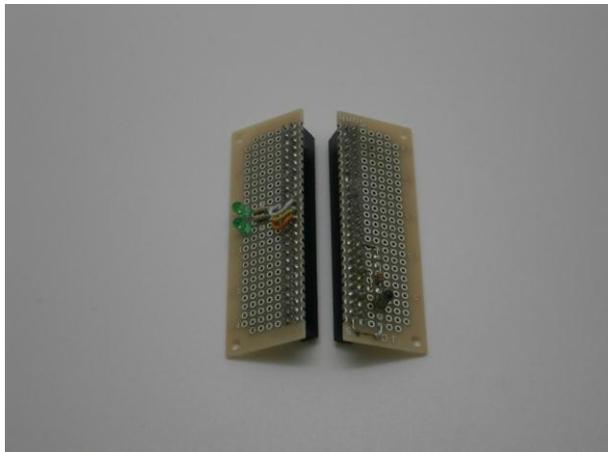
評価ボード「TSA-IC500」とセンサ、LED を接続するボード用のソケットです。それぞれ2個ずつご用意ください。パーツ販売をされている各種ショップやオンラインショップ等でお求めいただけます。

2.8 基板 2 枚



上記 2.7 のピンソケット (メス) を半田付けして、LED を載せる基板、センサを載せる基板とするための基板です。25x2 50pin のピンソケットが繋がるように 25x2 列以上の空きのある基板をご用意ください。

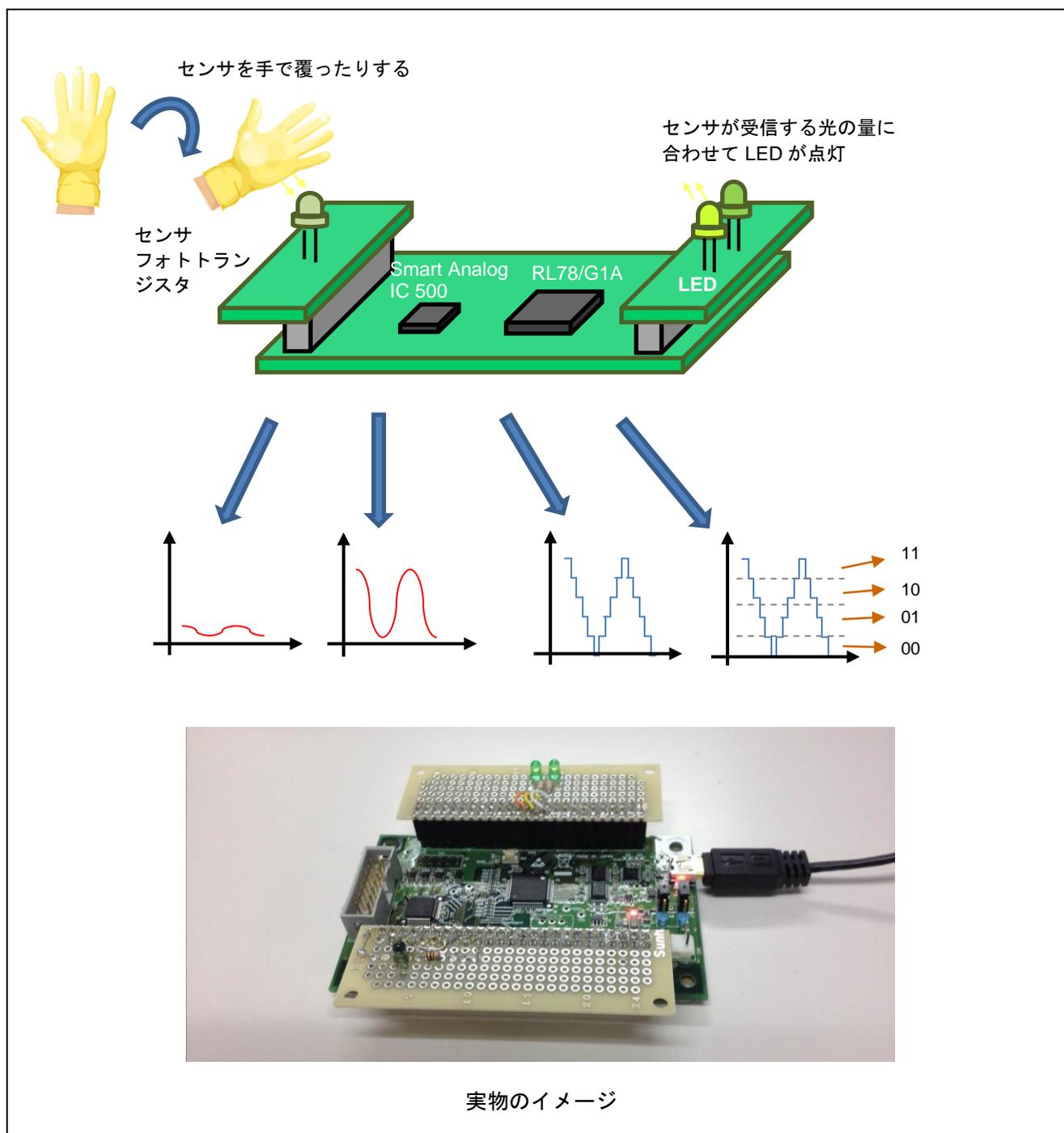
下の写真は上記基板を 2 つに割ってピンソケット及び LED、センサをそれぞれ半田付けしたものです。これを参考にして頂き、基板をご用意ください。



3. 実際に開発するもの

フォトトランジスタで光の量をセンシングして、その大きさに合わせて LED を点灯させます。フォトトランジスタから出力される信号を Smart Analog で最適な値に「増幅」して、ADC でデジタル化します。

ADC の測定レンジを 4 分割して、Smart Analog から出力される信号の大きさに合わせて LED を "00,01,10,11" で点灯させるように制御します。

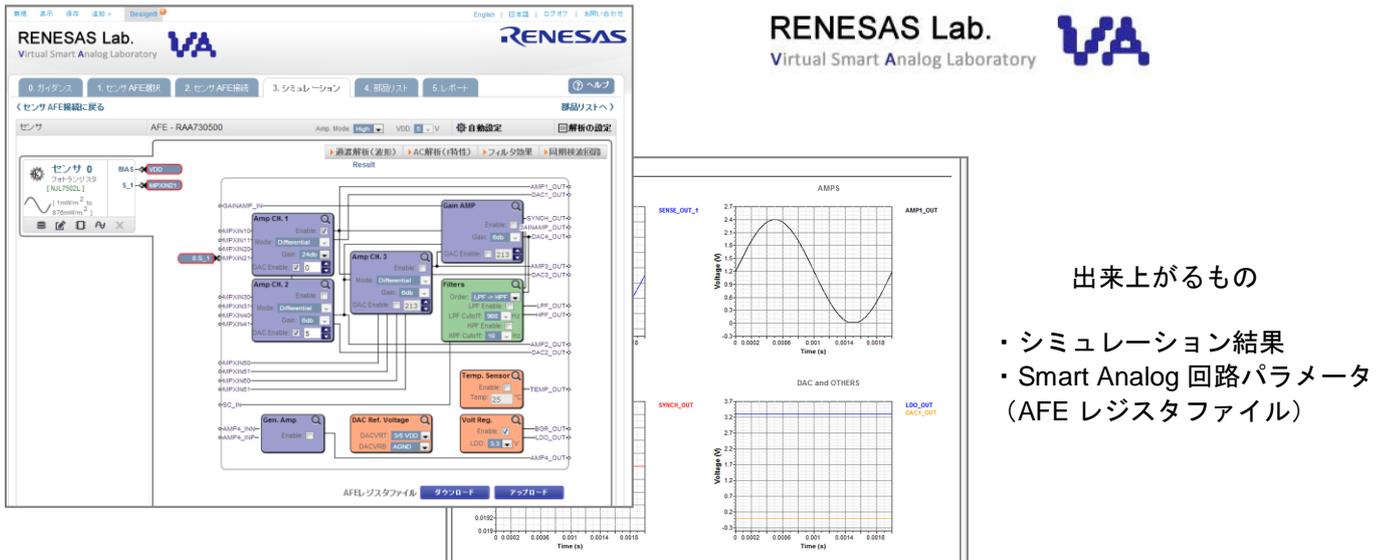


4. 全体の流れ

本資料では以下の手順に従って開発の流れを解説します。

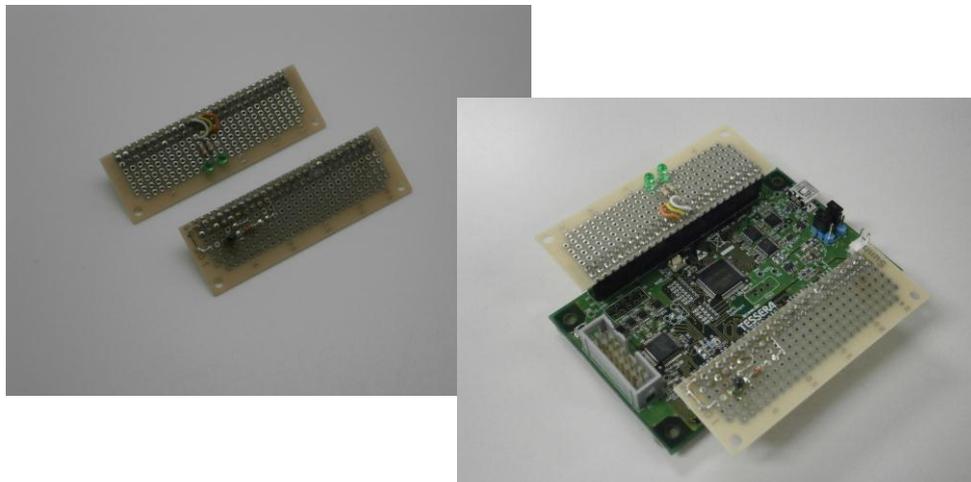
4.1 Renesas VA でシミュレーションをする。

シミュレータ(Renesas VA)を使ってシミュレーションします。



4.2 評価基板を作成する。

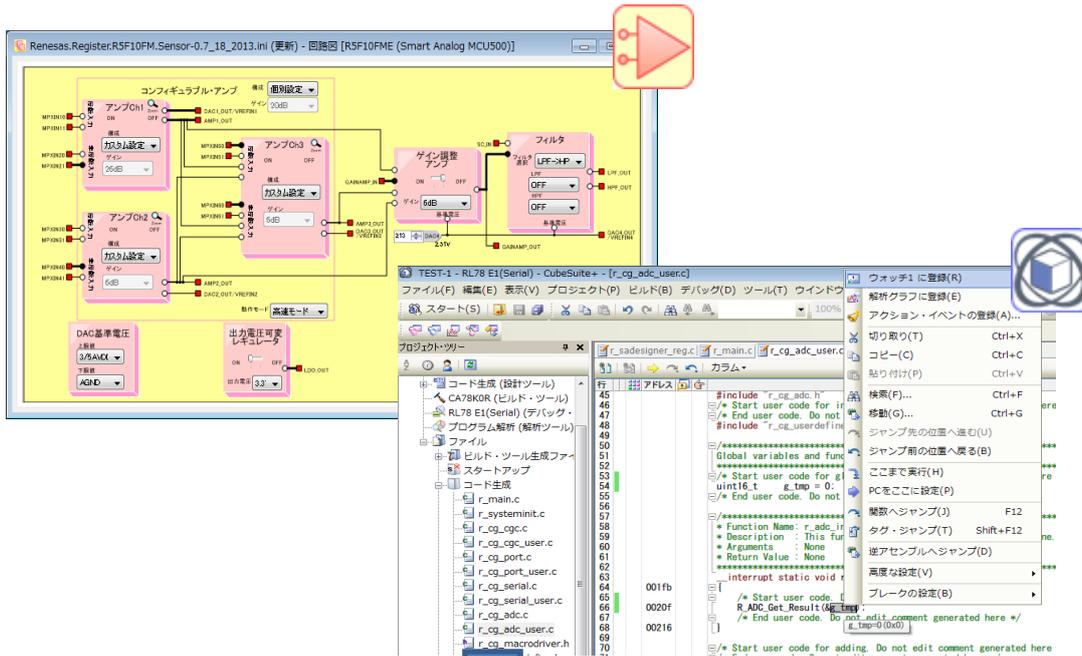
センサ、LED を評価ボードへ取り付ける基板を作り、評価ボードに取り付けます。





4.3 SA-Designer でレジスタファイル読み込み ~ CubeSuite+に取込んでプログラムを作成する。

Renesas VA で生成した Smart Analog の AFE レジスタファイルを SA-Designer で C ソースに変換し、CubeSuite+にパラメータを取り込んでセンサの駆動と LED を制御するプログラムを作成します。



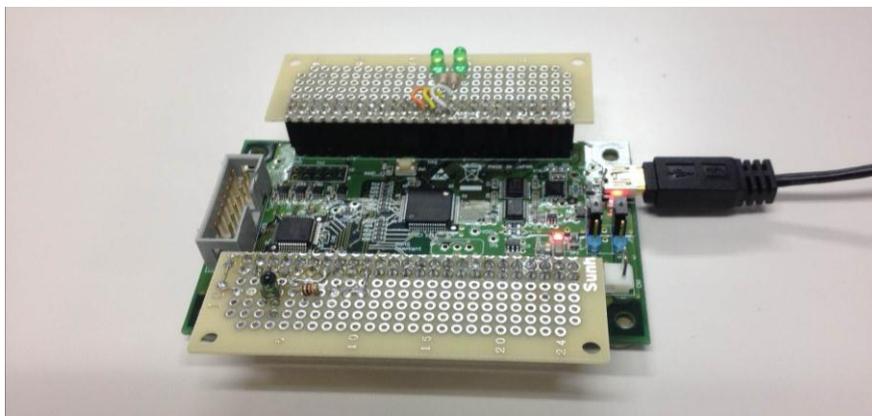
出来上がるもの

- ・コンパイル済みプログラム (ロードモジュール)



4.4 評価ボードで動作確認する。

4.3 で作ったプログラムが期待通りの動作をしているかを評価ボードで確認します。



出来上がるもの

- ・センサ評価環境及びプログラム

5. Renesas VA でシミュレーションをしてみよう！

まずは Smart Analog+センサのシミュレーションをしてみましょう。

Smart Analog + センサのシミュレーションツールとして Renesas が用意している Web シミュレータ「Renesas VA」を利用してシミュレーションします。

5.1 そもそも Web シミュレータとは？

センサ+Smart Analog のアナログ回路設計とシミュレーションを Web ブラウザのみで出来るツールです。専用のプログラムをダウンロードしてインストール、といった作業が不要で、インターネットにつながる環境があれば、何時でもアクセス、シミュレーションが可能です。

Renesas VA では以下の機能をサポートしています。

- 回路設計
Smart Analog とセンサを組み合わせ、Smart Analog 内部の回路設計が出来ます。
- 過渡解析
センサからの入力信号を Smart Analog でアナログ信号処理変換した出力波形を確認できます。
- AC 解析
Smart Analog で処理した波形の振幅周波数特性を確認できます。
- フィルタ効果解析
Smart Analog のフィルタの特性を確認出来ます。
- 同期検波回路動作解析
Smart Analog に搭載された同期検波回路の動作を確認できます。
- シミュレーション内容の読出し、保存、AFE レジスタファイルのアップロード・ダウンロード
シミュレーションした結果の保存や読出し、Smart Analog の AFE レジスタファイルのアップロードとダウンロードが出来ます。

5.2 Renesas VA を使ってみよう！

では実際に Renesas VA を使ってみましょう。

Web ブラウザから Renesas の Smart Analog の「Smart Analog Web シミュレータ」のページにアクセスしてください。

- Renesas VA :

http://japan.renesas.com/smart_analog/va

Renesas の TOP ページ → 製品情報 → Smart Analog → Smart Analog Web シミュレータ



ここをクリック

画面右上にある「すぐにお試し >>」をクリックしてください。



「同意する」をクリック

免責事項画面の最下部の「同意する」をクリックしてください。

なお、次の画面で My Renesas の ID とパスワードの入力画面となります。

Renesas VA を利用するには My Renesas の ID とパスワードが必要ですので、まだ ID をお持ちでない方は「My Renesas のご登録がまだのお客様」をクリックして、ID を新規登録(無料)してください。

【注】 ご登録の際「ご希望サービス」でカテゴリから「Smart Analog」を選択してリスト追加すると、Renesas VA、SA-Designer 等、Smart Analog に関連するニュースレターが登録されます。

MyRenesas

会員情報更新

ご希望サービス/プレミアムサービス他の登録

下記のプルダウンメニューを使い、製品を選択してリストに加えてください。
プルダウンからAllを選択すると、関連する全てのニュースレターが登録されます。

カテゴリ: Smart Analog

ファミリー: ファミリーを選択してください

シリーズ:

グループ: 追加

登録済製品	カテゴリ	ファミリー	シリーズ	グループ	
マイコンコンピュータ					登録
マイコンコンピュータ	RL78 ファミリー				登録
Smart Analog					登録
開発環境					登録

アプリケーション/システムソリューション

デジタル家電

自動車

ネットワーク

ワイヤレス

その他

ニュース&イベント

プレスリリース

セミナー情報

高周波・光デバイス

RENEAS Lab. VA RENEASAS

Virtual Smart Analog Laboratory

0 ガイダンス 1 センサAFE選択 2 センサAFE接続 3 シミュレーション 4 部品リスト 5 レポート

Renesas VA サイトへようこそ。
シミュレータの使い方

シミュレーションスタート

ここをクリック

1 センサとAFEの選択

登録済みのセンサリストから選択してシミュレーションする方法と、カスタムセンサとして電気特性を入力してシミュレーションする方法の二通りが可能です。その後、センサに接続するアンプを選択します。

Selected Sensors

Sensor 0 (Accelerometer) [Add Sensor]

Sensor 1 (VoltageSource) [Add Sensor]

Sensor 2 (Resistor) [Add Sensor]

AFE 200K 100K

Amplifiers

Non-inverting

Instrumentation

Transimpedance (IV)

Filters

Low-Pass

High-Pass

DAC

Resolution Any Channels Any

Other

Voltage Reference

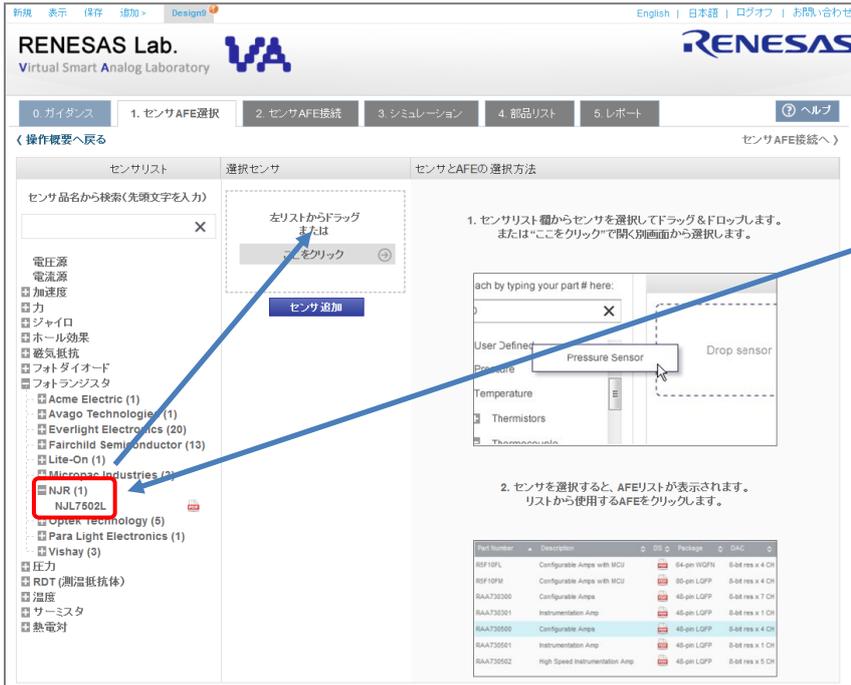
Temp. Sensor

Voltage Regulator

Part Number	Description	DS	Package
RSF10FL	Configurable Amps with MCU	64-pin WQFN	8-100-0001
RSF10FI	Configurable Amps with MCU	50-pin LQFP	8-100-0002
RAA730300	Configurable Amps	48-pin LQFP	8-100-0003
RAA730301	Instrumentation Amp	48-pin LQFP	8-100-0004
RAA730500	Configurable Amps	48-pin LQFP	8-100-0005
RAA730501	Instrumentation Amp	48-pin LQFP	8-100-0006
RAA730502	High Speed Instrumentation Amp	48-pin LQFP	8-100-0007

My Renesas の ID、パスワードを入力すると Renesas VA の画面となります。

右上の「シミュレーションスタート」をクリックしてください。

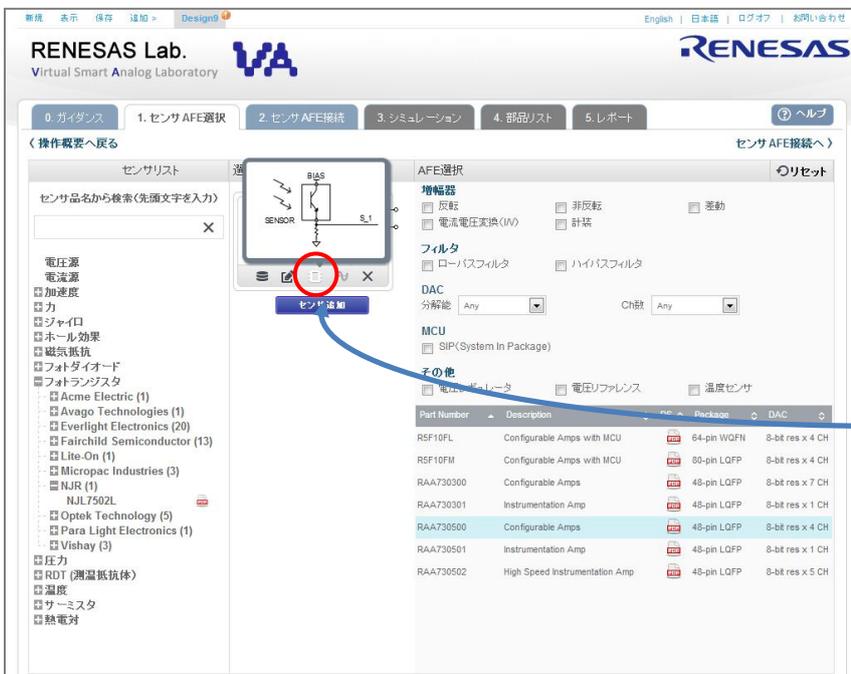


センサリストから「フォトランジスタ」
→「NJR」→「NJL7502L」を選択して
選択センサヘドラッグ

Part #: NJL7502L
 Manufacturer: NJR
 Description: Ambient Light Sensor
 Intensity (W/m2): 1n to 73
 λ^P : 5.6e-007

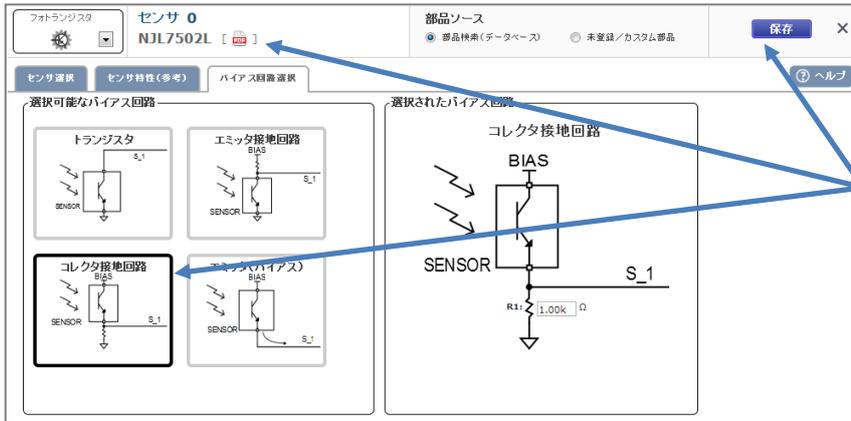
始めに Smart Analog に接続して使うセンサを選択します。

今回はフォトランジスタを使用しますので、プルダウンメニューから「フォトランジスタ」→「NJR」
→「NJL7502L」を選択して、選択センサにドラッグして、具体的型番を指定します。



バイアス回路選択「Sensor
Circuit」をクリック

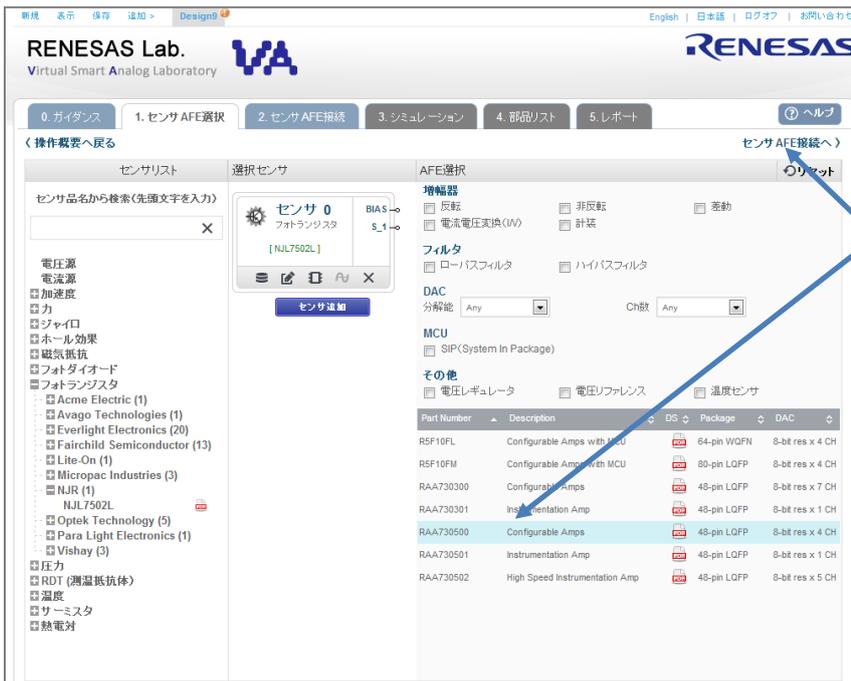
次にセンサと Smart Analog IC を接続する方法を選択します。



ここをクリックしてデータシートをダウンロード
 バイアス回路は「コレクタ接地回路」タイプを選択して、右上の「保存」ボタンをクリック

画面左側上の「PDF アイコン」(NJR: NJL7502L のとなり)は NJL7502L のデータシートのリンクです。クリックしてデータシートをダウンロードしておいてください。

「NJL7502L」と Smart Analog を接続する方法(バイアス回路)は4つの方法があります。今回はこの中で5Vで駆動した時に最も広くダイナミックレンジが取れる「コレクタ接地回路」方式を選択します。

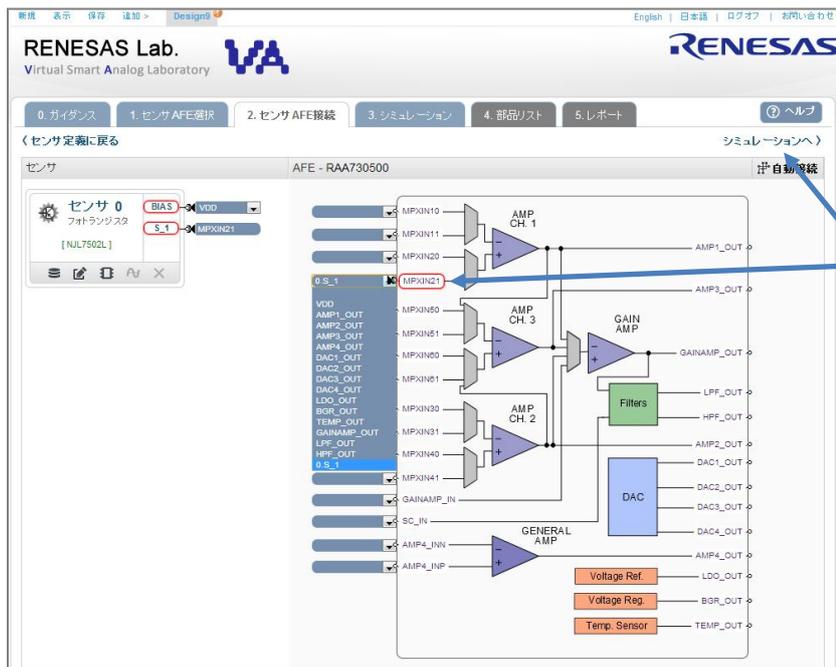


Part Number に「RAA730500」を選択し、右上の「センサ AFE 接続へ」矢印ボタンをクリック

シミュレーションで使用する Smart Analog の種類を選択します。

現在 Renesas VA では Smart Analog MCU(AFE と MCU を SiP で1つのパッケージに収めたもの)と Smart Analog IC(AFE 単体)をサポートしています。

今回は Smart Analog IC(AFE 単体)を搭載した評価ボードを使いますので、Smart Analog IC(RAA730500)を選択します。



MPXIN21 端子のプルダウンからセンサからの入力「0.S_1」を選択し、右上の「シミュレーションへ>」矢印ボタンをクリック

センサと Smart Analog IC との接続画面です。

ここで設定したセンサ - Smart Analog IC の接続情報に沿って、この後評価ボードで実際にセンサを接続します。自動接続機能により MPXIN20 端子に接続されますが、MPXIN21 端子のプルダウンからセンサからの入力「0.S_1」を選択し、MPXIN21 にセンサを接続してください。

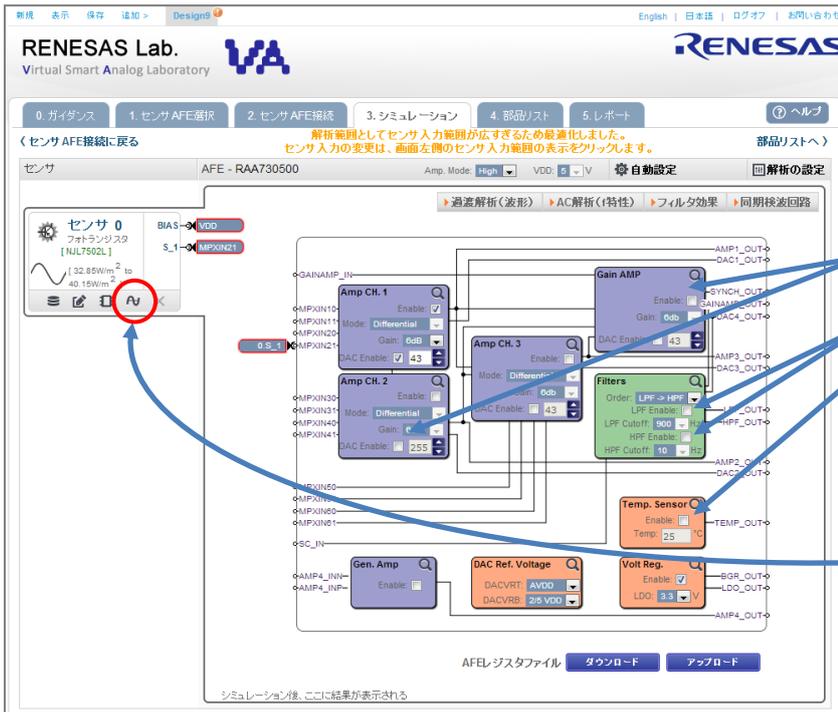
ここで、MPXIN20 端子から MPXIN21 端子へ変更すると以下のようなメッセージが表示されます。OK をクリックしてください。

メッセージ内容：

「手動によるセンサと AFE 接続は、シミュレーション画面の「自動設定」がサポートされない場合があります。デフォルト接続に戻りたい場合は、「自動接続」をクリックしてください。」



ここをクリック



Amp CH.2 の DAC Enable を OFF にする
Gain AMP の Enable、DAC Enable を OFF にする

Filters の LPF Enable、HPF Enable を OFF に、Temp Sensor の Enable も OFF にする

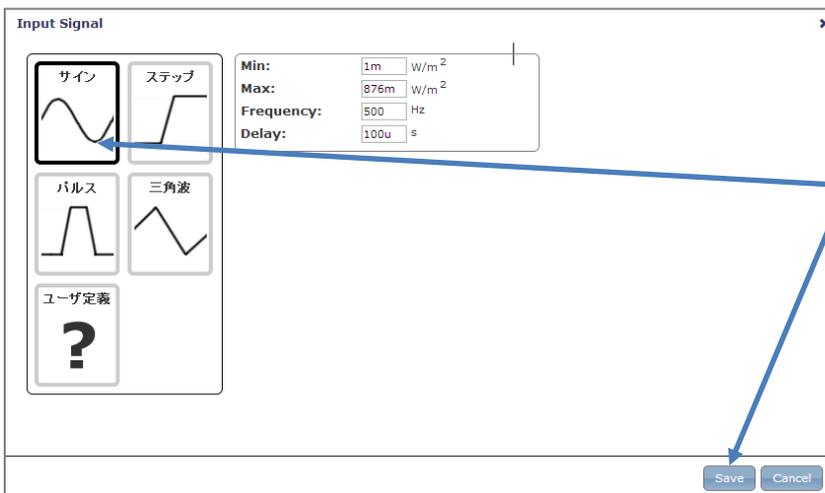
「Sensor Input」をクリック

この画面にて Smart Analog 内部の回路の設定、センサへ入力する外部信号の設定、シミュレーションの種類などを設定します。

Smart Analog IC の回路定数は既にデフォルト値が入力されていて、このままシミュレーションすることが出来ます。しかしながら今回は最も分かり易い構成とするため、Amp CH.1 のみを使用します。デフォルトで「ON」に設定されている「Amp CH.2 内の DAC Enable」、「Gain AMP」の「Enable」、「Gain AMP 内の DAC Enable」、「Filters 内の LPF Enable、HPF Enable」、「Temp. Sensor」の「Enable」も OFF にしてください。

なお、Volt Reg は AD コンバータの基準電圧として使用しますので、ここは「ON」のままにしてください。

次にセンサへ入力する信号を選択します。



Input Signal は「サイン」を選択

Min : 1m W/m²

Max : 876m W/m²

Frequency : 500 Hz (デフォルト)

Delay : 100us (デフォルト)

上記設定後、右下の「Save」ボタンをクリック

まずはサイン波で確認をしてみましょう。

なお、センサへの入力信号の大きさは次項の < W/m2 とルクス[Lx]の関係 > から Max を 876mW/m2 とします。

< W/m2 とルクス[Lx]の関係 >

今回のセンサであるフォトダイオードの入力信号の単位は W/m2 で定義されています。

これでは何がなんだか、・・・ですので、まずは分かり易い単位に変換します。

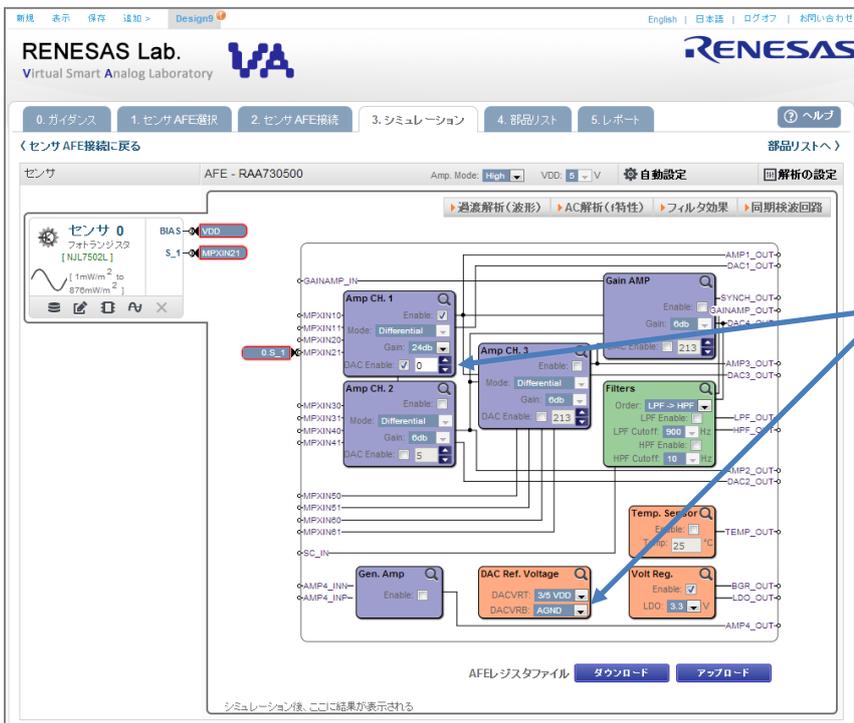
光の明るさを示す単位としてルクス[Lx]という単位が良く使われます。ルクス[Lx]と W/m2 の関係は、

$$1\text{Lx} = 1.46 \text{ mW/m}^2 \text{ 注}$$

という式ですので、まずはルクスで想定して、上記の式で W/m2 に換算して入力すれば良いわけです。

【注】 この式には正確には「可視光スペクトルの中間波長 555nm において」という条件がつきます。

一般家庭や会社の事務所ではおおよそ 150~400 ルクス程度(もちろん場所によって差は出ます。) くらいです。今回は「センサを手で覆う=部屋の光を遮断する」をセンサへの入力条件としますので、センサへの入力信号は 400 ルクス[Lx] = 584 mW/m2 程度を想定します。ただ、環境によって差がでますので、600 ルクス[Lx] = 876 mW/m2 を Max として設定することとします。

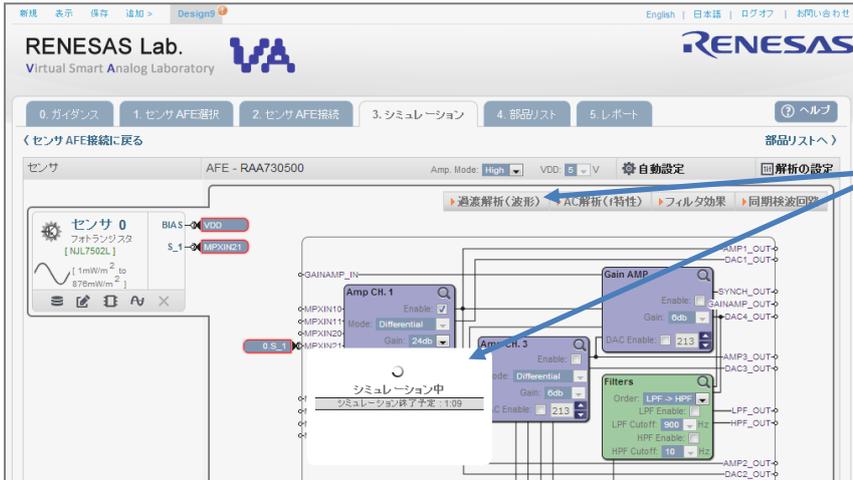


Amp CH.1 の DAC1 の設定値を「0」に設定
DAC Ref. Voltage の DACVRB を AGND に設定

最後に Smart Analog の AMP のゲインとバイアス電圧回路の設定をします。

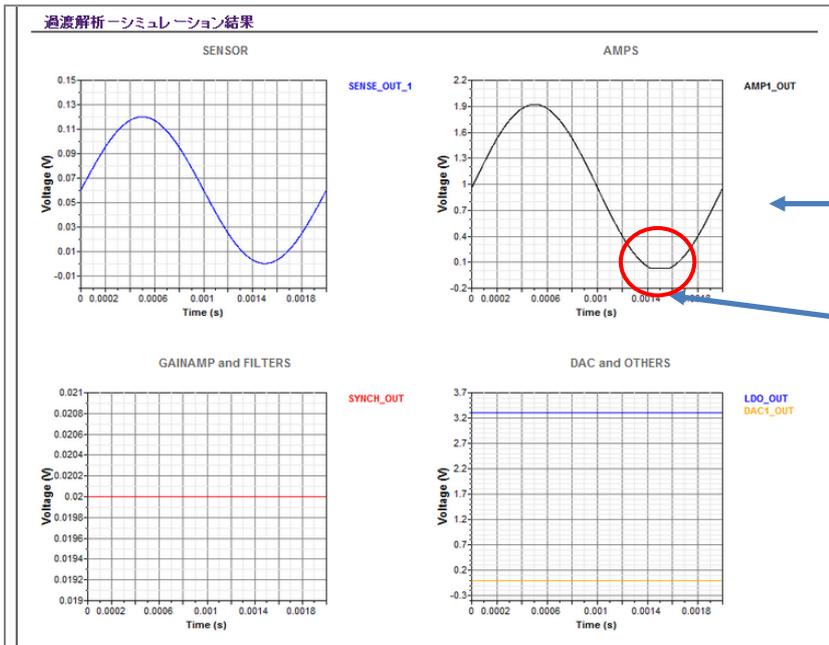
Amp CH.1 の Gain は Renesas VA が自動で最適化した数値(24dB)をそのまま使用し、バイアス電圧を設定する DAC1 の設定値を 0 にします。

さらに DAC のリファレンス電圧を設定する「DAC Ref. Voltage」の下限側「DACVRB」を AGND に設定します。



「過渡解析(波形)」ボタンをクリック
シミュレーションが開始され終了予
定時間が表示

ここまで設定が終わったら実際にシミュレーションを開始します。
上側中央の過渡解析(波形)ボタンをクリックするとシミュレーションが開始されます。

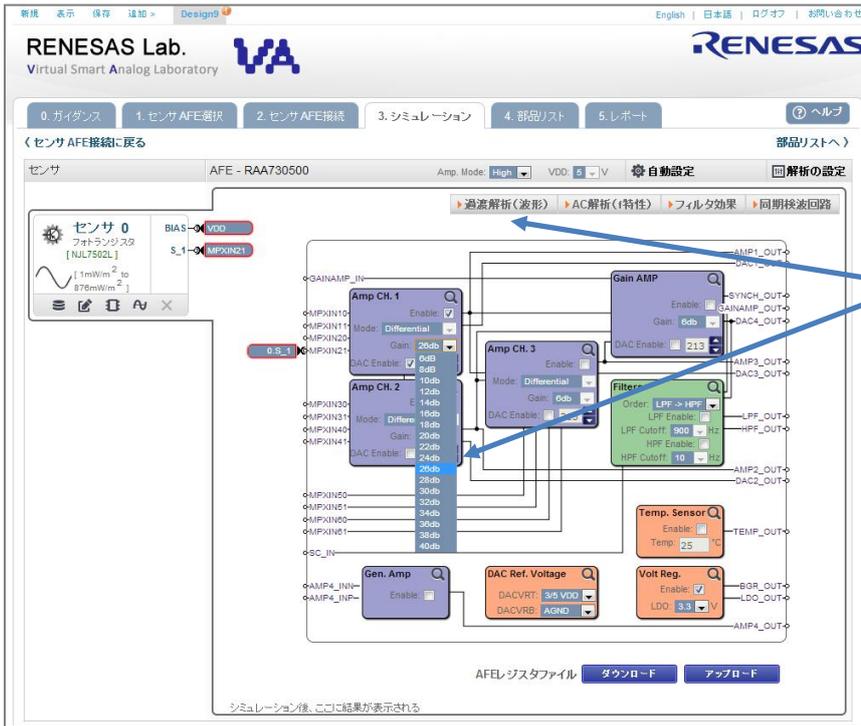


回路画面下側にシミュレーション
波形が表示される
0V 以下の信号が出力されず「ク
リップ」している事を確認

シミュレーションが終了すると過渡解析(波形)ボタンすぐ下にシミュレーション結果の「Result」リンク、
また回路画面下側にシミュレーション結果の波形が表示されます。

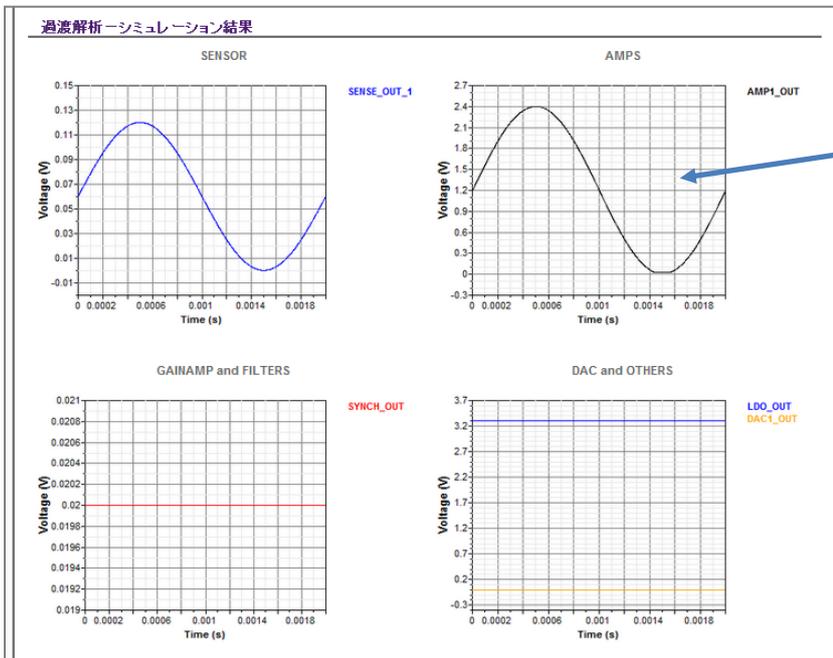
シミュレーションの結果から、AMP1 の出力の下側が「クリップ」されていることが分かります。これは
AMP1 のバイアス電圧を 0V に設定したため、AMP1 の動作点が 0V になっている、という理由です。

この結果から 600Lx 時の AMP1 出力が約 1.921V と少し小さいので、ゲインを上げてみます。



AMP1 の Gain を 26dB に設定する
過渡解析ボタンを再クリック

AMP1 の Gain を 26dB に設定してもう一度「過渡解析」ボタンをクリックしてシミュレーションします。



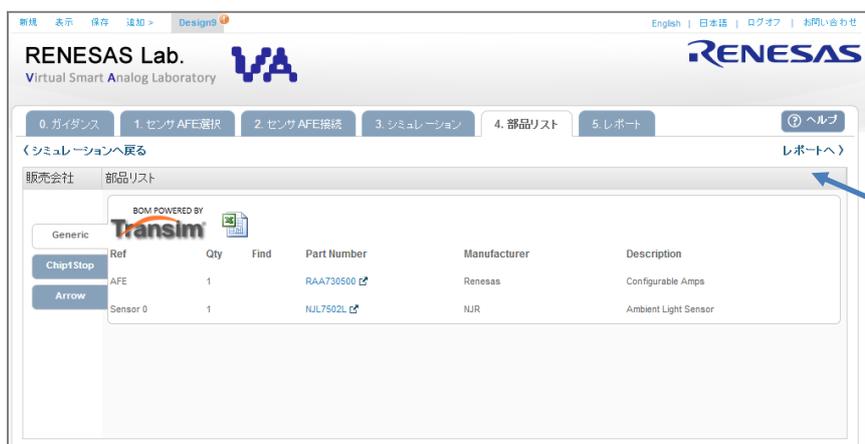
シミュレーション波形を確認する

シミュレーション結果から、AMP1 の出力は約 2.4V まで増幅されていることが確認できます。十分な出力が得られている事が確認出来ましたので、このパラメータを使う事とします。



右上の「部品リストへ>」
矢印ボタンをクリック

右上の「部品リストへ>」矢印ボタンをクリックして次の画面へ進んでください。



右上の「レポートへ>」
矢印ボタンをクリック

この画面は今回のシミュレーションで使用した部品リストの確認画面です。
右上の「レポートへ>」矢印ボタンをクリックして次の画面へ進んでください。

RENASAS Lab. VA

Virtual Smart Analog Laboratory

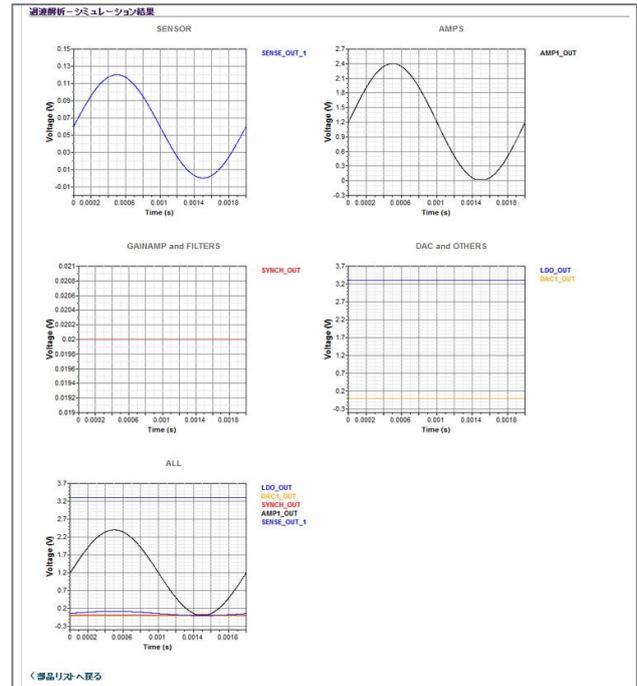
0. ガイダンス 1. センサAFE選択 2. センサAFE接続 3. シミュレーション 4. 部品リスト 5. レポート

製品リストへ戻る

Renesas RAA730500

ダウンロード

クリックして保存



今回のシミュレーションの結果が表示されます。

右上の「pdf アイコン」をクリックするとこの画面の pdf 「Renesas_RAA730500.pdf」がダウンロードできます。

画面上側の「レジスタ」の項目の「ダウンロード」ボタンをクリックして、「Renesas.Register.RAA730500.Sensor-XX_XX_XXXX.ini」をダウンロードして PC に保存してください。

このファイルが今回シミュレーションした Smart Analog のアナログ回路設定値のパラメータ (AFE レジスタファイル) となります。

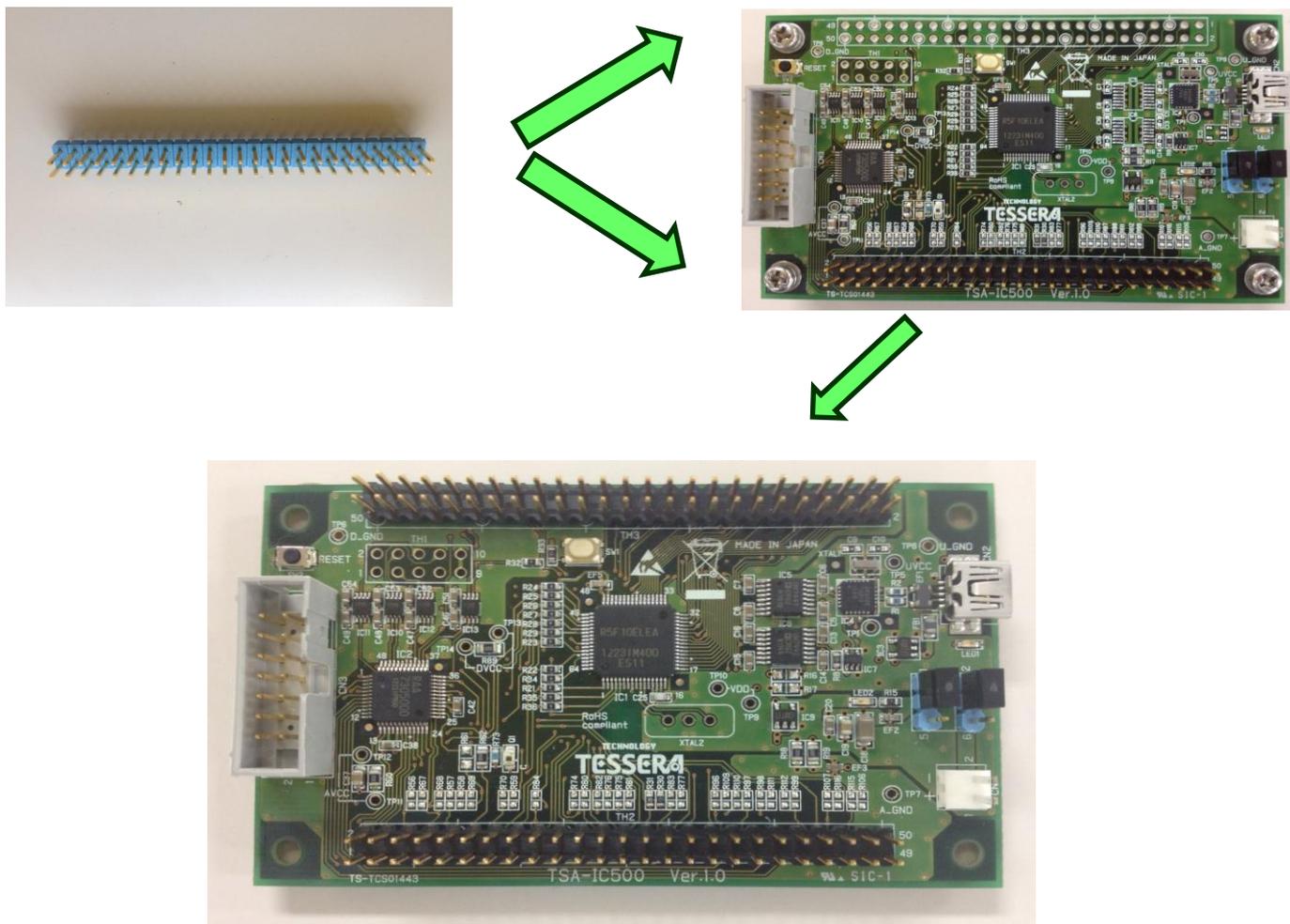
シミュレーションは以上で終了です。次に評価ボードでの開発に入ります。

6. センサ基板、LED 基板を作成しよう！

評価ボードに取り付けるセンサの基板、LED の基板を作りましょう。

6.1 評価ボード「TSA-IC500」に「ピンヘッダ オス 2x25 50p」を取り付ける。

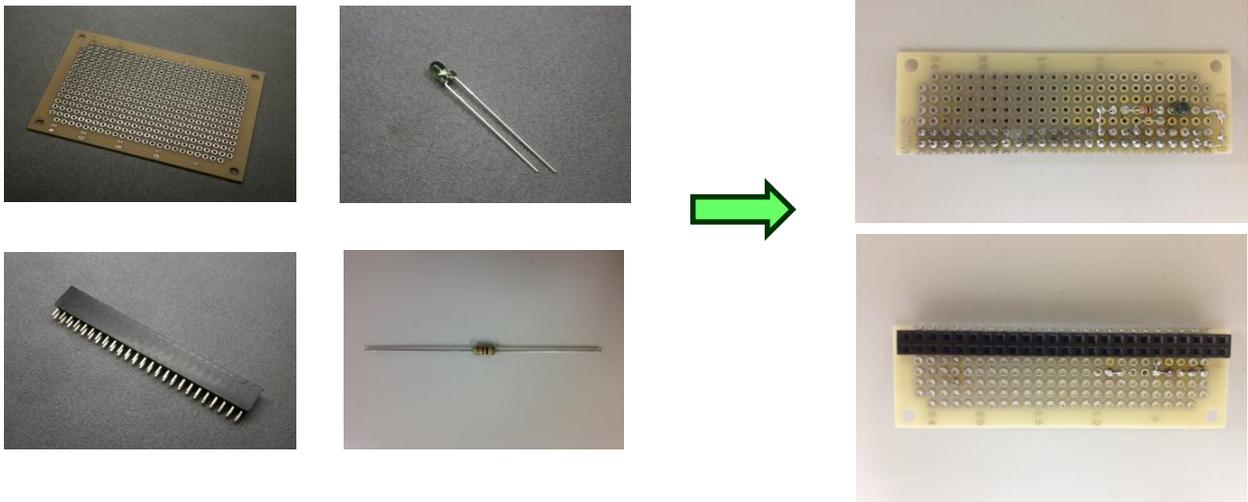
「TSA-IC 500」の上側、下側に「ピンヘッダ オス」をそれぞれ半田付けしてください。



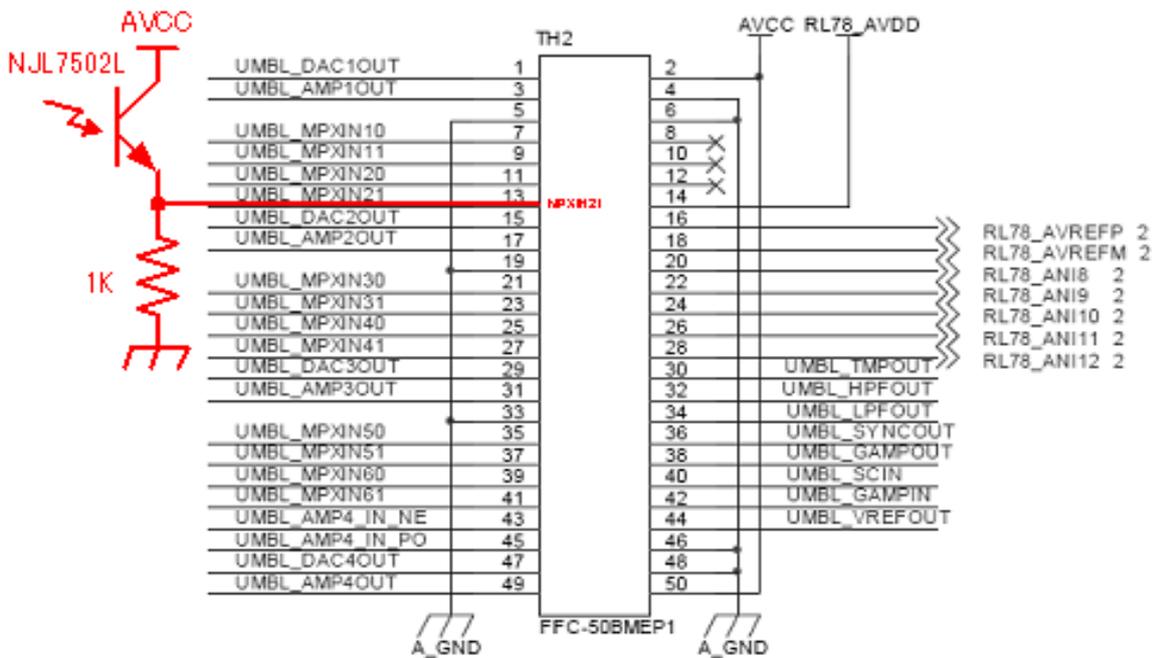
6.2 センサ基板、LED 基板を作成する。

「TSA-IC 500」に取り付けるセンサ基板、LED 基板を作ります。

6.2.1 センサ基板

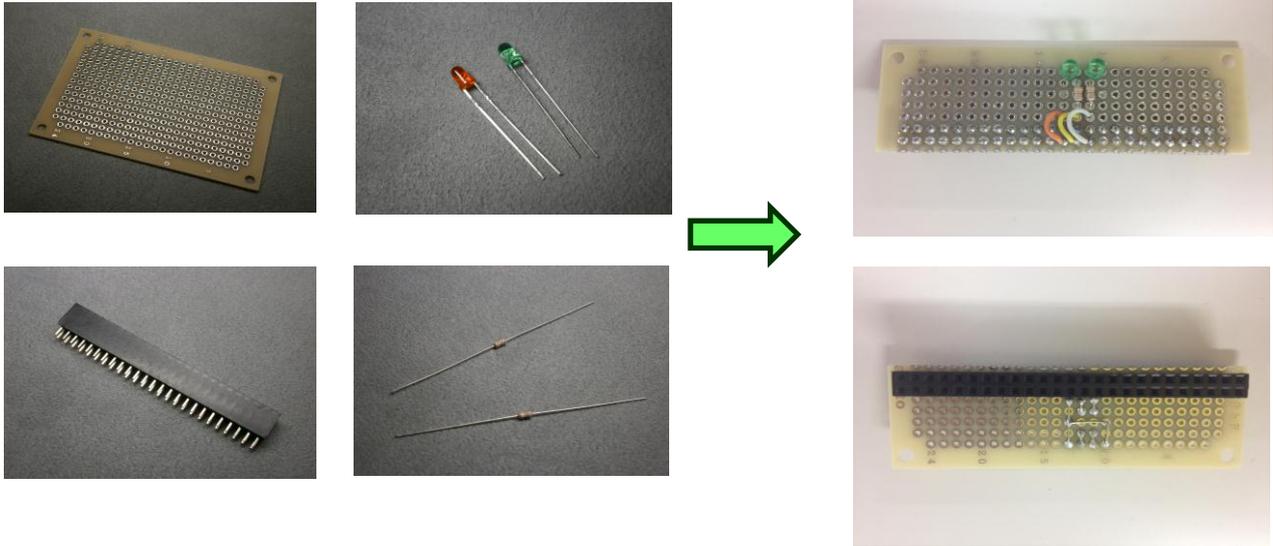


センサ基板の回路は以下のように半田付けしてください。

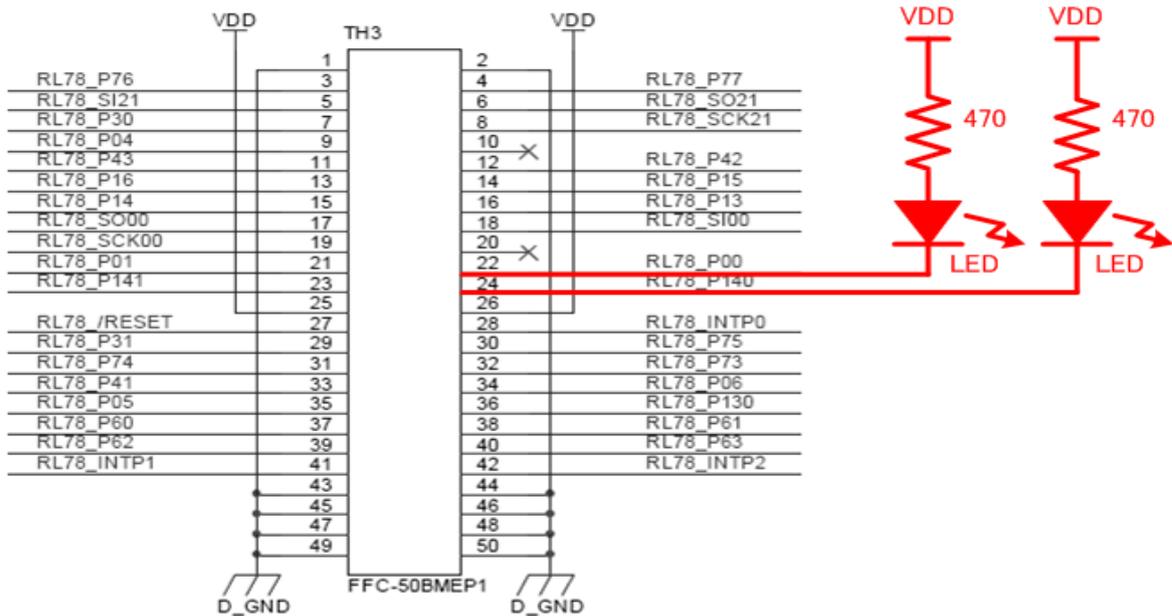


センサ基板回路図

6.2.2 LED 基板



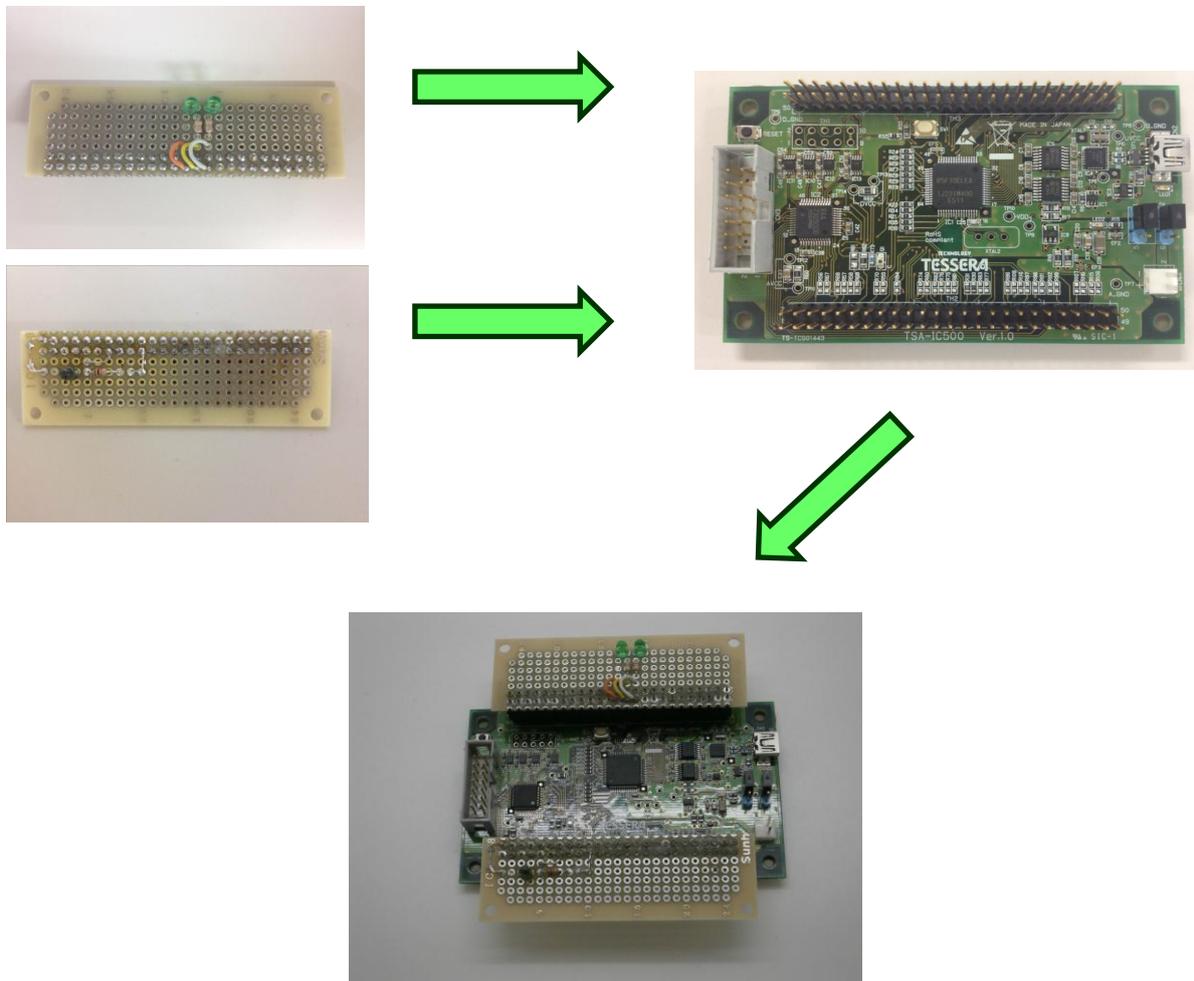
LED 基板の回路は以下のように半田付けしてください。



LED 基板回路図

6.3 出来上がったセンサ基板、LED 基板を評価ボードに取り付ける。

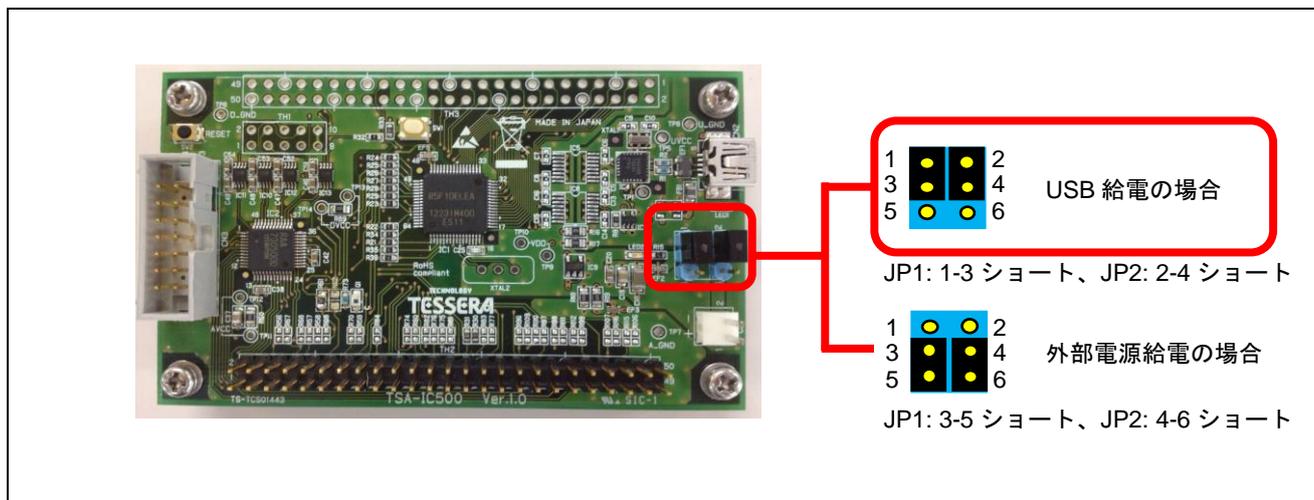
出来上がったセンサ基板、LED 基板を評価ボードに取り付けてください。



【注】LED 基板はボード正面(写真の向き)で上側、センサ基板は下側に取り付けてください。

6.4 評価ボードの電源を設定する。

今回使用する評価ボード「TSA-IC 500」は、電源供給方法を「外部給電」「USB 給電」のどちらかを選択できます。今回は「USB 給電」で使用しますので TSA-IC 500 のボードの設定が下記のようにになっているかを確認してください。



以上で評価ボードの準備は完了です。

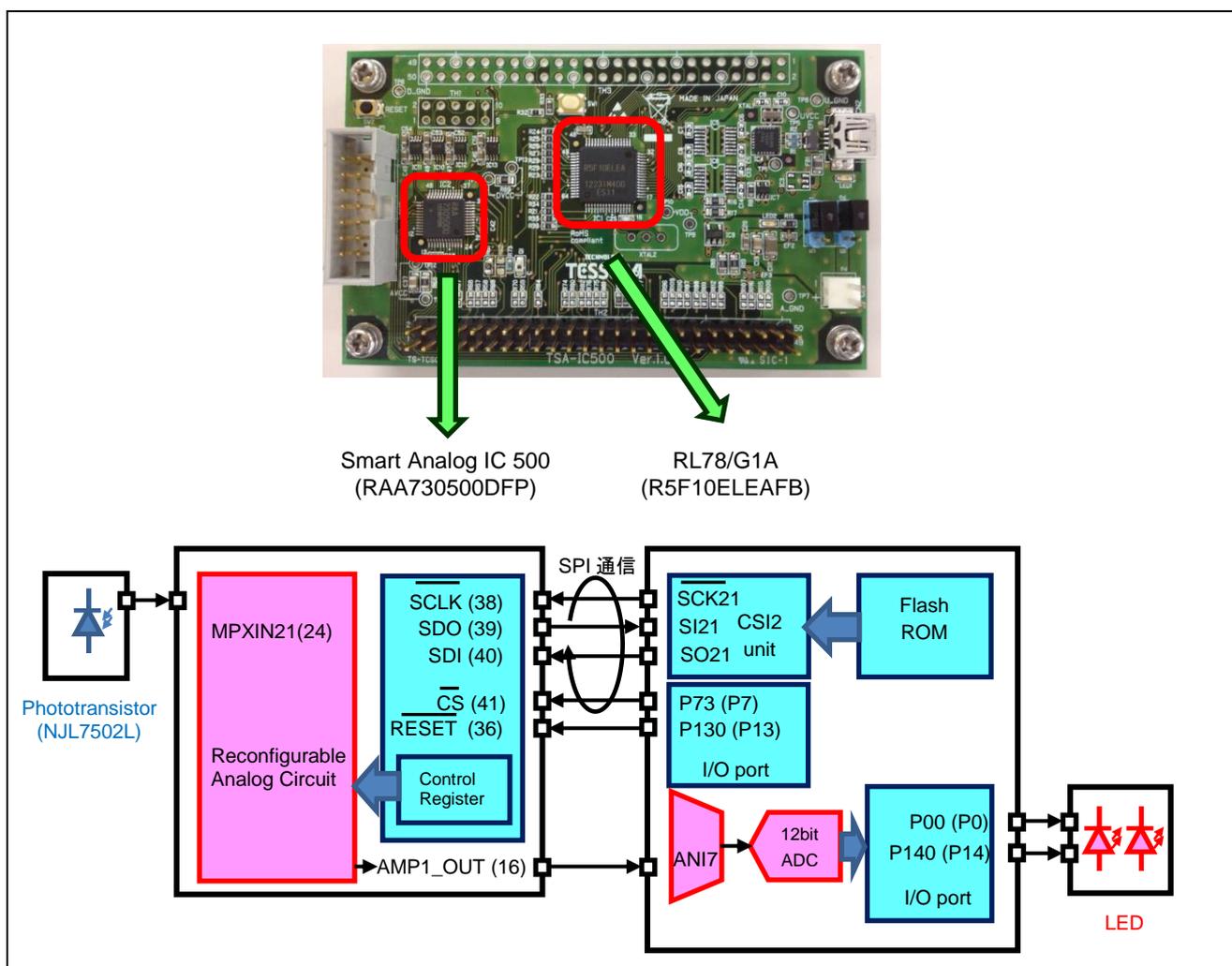
次にこのボードを使ってプログラムを作ります。

7. SA-Designer で AFE レジスタファイルを取り込んで、CubeSuite+でプログラムを作ってみよう！

5.で作った Renesas VA での Smart Analog IC の AFE レジスタファイル、6.で作った LED 基板、センサ基板及び評価ボードを使って、評価ボード上で動かすためのプログラムを作ってみましょう。

7.1 評価ボードに搭載されている IC とマイコンの接続状態とこのパートで作るプログラムの内容

実際にプログラムを開発する前に、評価ボードに搭載されている Smart Analog IC とマイコンの構成と端子情報について説明します。



Smart Analog IC 500(RAA730500DFP)と RL78/G1A(R5F10ELEAFB)との端子接続と内部回路は上の図のような関係になっています。Smart Analog IC 500 は RL78/G1A と SPI で接続されていて、RL78/G1A から SPI で制御されて動作します。

5.で作った Smart Analog IC の AFE レジスタファイルはこのレジスタの値で、このレジスタ値をマイコンの Flash ROM に予め保存しておいて、マイコンの SPI ユニットから Smart Analog IC へ転送して、Smart Analog IC を動作させるようにプログラムを作ります。

また、センサ信号は Smart Analog IC を介して RL78/G1A の ADC で受け取り、A/D コンバートした結果から LED ポートを制御して LED を光らせる部分のプログラムも作ります。

7.2 SA-Designer をインストールしよう！

プログラムを組む前にまずは SA-Designer をインストールしましょう。

7.2.1 そもそも SA-Designer とは？

SA-Designer は、Smart Analog 製品のアナログフロントエンド回路を設計して、その回路データを C ソースコードとして生成するツールです。

従来、センサ動作に合わせたアナログフロントエンド回路の部品調達や部品変更に伴ってハードウェアの準備に時間がかかっていましたが、Smart Analog と SA-Designer であればそれらが不要となり、SA-Designer による PC 画面でのマウス操作だけで簡単に回路設計をすることができるようになります。

また、センサの選定に利用できる Web シミュレータ「Renesas VA」の AFE レジスタファイルをインポートできるほか、ユーザプログラム統合開発環境「CubeSuite+」との連携もできるため、センサの選定から、回路設計、およびプログラム開発までをすべてソフトウェアで完結でき、効率よくシステム開発を進めることができます。

SA-Designer のサポートする機能：

- 回路ブロック図スタイルの GUI による回路設計・カスタマイズ
- 作成した回路データ（レジスタ値等）の一覧表示、保存、復元
- 「Renesas VA」で作成した回路デザイン（AFE レジスタファイル）の読み込み
- 回路設計用の C ソースファイル生成
- 統合開発環境 CubeSuite+へのプロジェクト登録、CubeSuite+でのビルド/ダウンロード/デバッグの実行
- チュートリアル搭載

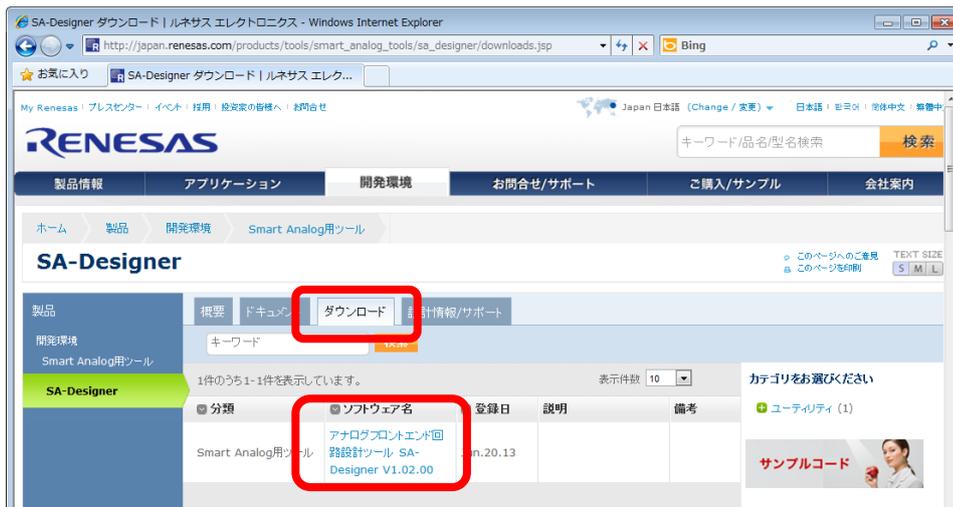
7.2.2 SA-Designer のインストール

Web ブラウザで Renesas の開発環境の「Smart Analog 用ツールスタータキット/プラットフォーム」の「SA-Designer」 ページにアクセスしてください。

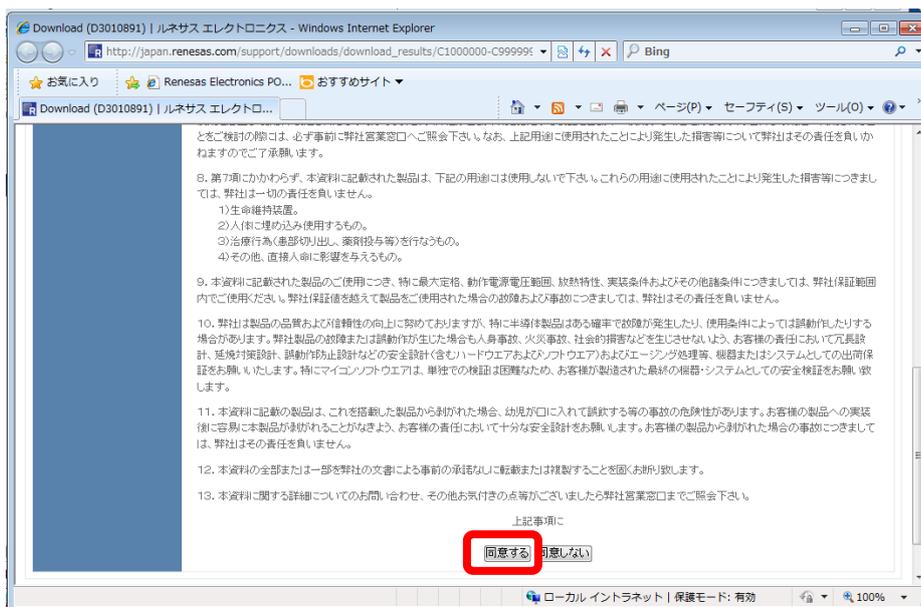
- SA-Designer

http://japan.renesas.com/sa_designer

Renesas の TOP ページ → 開発環境 → Smart Analog 用ツール → SA-Designer



「ダウンロード」タブの「アナログフロントエンド回路設計ツール SA-Designer Vx.xx.xx」をクリックしてください。



免責事項画面が表示されますので「同意する」をクリックしてください。

この後、My Renesas の ID とパスワードの入力画面が表示されますので、ID とパスワードを入力してください。

まだ ID をお持ちでない方は「My Renesas のご登録がまだのお客様」をクリックして、ID を新規登録(無料)してください。

【注】 ご登録の際、「ご希望サービス」でカテゴリから「Smart Analog」を選択してリスト追加すると、Renesas VA、SA-Designer 等、Smart Analog に関連するニュースレターが登録されます。

MyRenesas

会員情報更新

ご希望サービス / プレミアムサービス他の登録

下記のプルダウンメニューを使い、製品を選択してリストに加えてください。
プルダウンから4つを選択すると、関連する全てのニュースレターが登録されます。

カテゴリ: Smart Analog

ファミリー: ファミリーを選択してください

シリーズ:

グループ: 追加

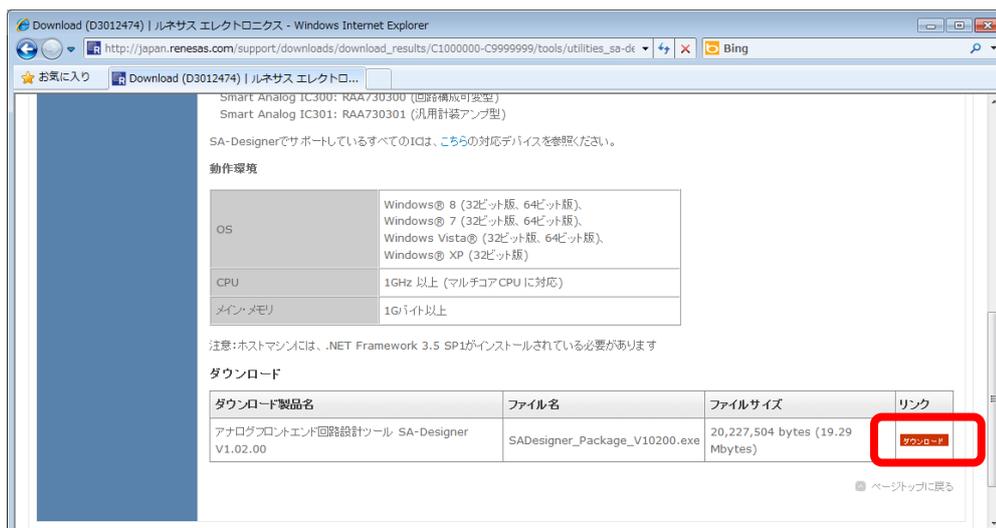
登録済製品	カテゴリ	ファミリー	シリーズ	グループ	
マイクロコンピュータ					削除
マイクロコンピュータ		RL78 ファミリー			削除
Smart Analog					削除
開発環境					削除

アプリケーション/システムソリューション

- デジタル家電
- 自動車
- ネットワーク
- ワイヤレス

その他

- ニュース & イベント
- プレスリリース
- セミナ情報
- 高周波・光デバイス

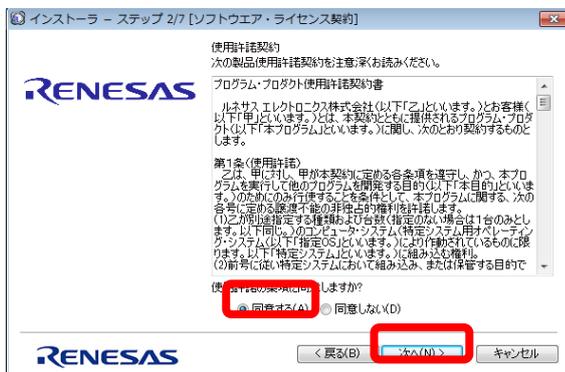


画面右下の「ダウンロード」をクリックしてファイルをパソコンに保存してください。

ダウンロードが完了したら、ダウンロードしたファイル「SADesigner_Package_VXXXXX.exe」をダブルクリックしてください。インストーラが起動します。



「次へ」をクリックしてください。



本ソフトウェアの使用許諾契約書が表示されます。

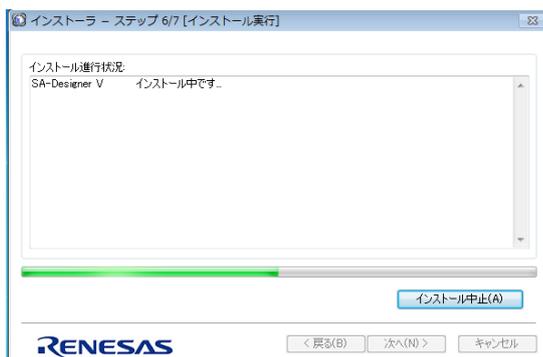
「同意する」をチェックして、「次へ」をクリックしてください。



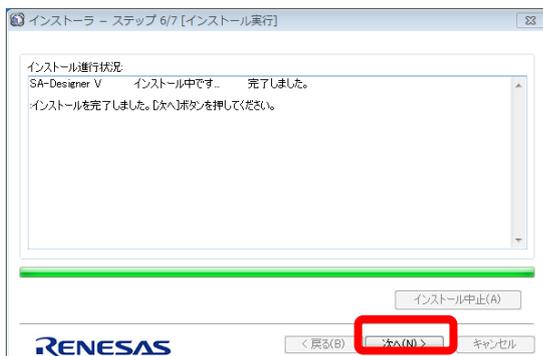
SA-Designer のインストール先が表示されますので、内容を確認して「次へ」をクリックしてください。



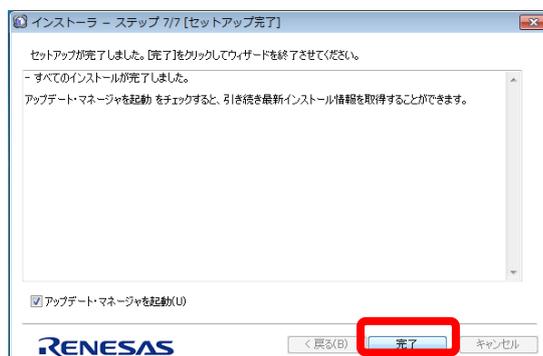
「次へ」をクリックしてください。



インストールが開始されます。



「次へ」をクリックしてください。



「完了」をクリックしてください。これで SA-Designer のインストールは終了です。

7.3 CubeSuite+をインストールしよう！

CubeSuite+をインストールしよう！

既に CubeSuite+をインストール済みの方はここを読み飛ばして、7.4に進んでください。

【注】 Ver1.02 未満をお使いの方は最新版へアップグレードしてください。
SA-Designer と CubeSuite+の連動は Ver1.02 以上が必要です。

7.3.1 CubeSuite+のインストール方法について

CubeSuite+の機能やサポート環境については、下記ページを参照ください。

- 統合開発環境 CubeSuite+
<http://japan.renesas.com/cubesuite+>
Renesas の TOP ページ → 開発環境 → 統合開発環境(IDE) → CubeSuite+

インストールについては、上記ページの「ダウンロードタブ」から「【無償評価版】統合開発環境 CubeSuite+ Vx.xx.xx」をダウンロードしてインストールしてください。

【注】 Ver1.02 以上の最新版をインストールしてください。

7.4 SA-Designer を起動する。

では実際にプログラムを作ってみましょう。まずは SA-Designer を起動させます。
Windows のスタートメニューから SA-Designer を起動します。

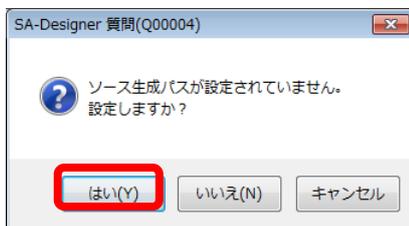
[Renesas Electronics Utilities] → [スマート・アナログ・ツール] → [SA-Designer]

SA-Designer のスタート画面で、「既存の回路を開く」の GO ボタンをクリックしてください。



7.5 Renesas VA で作成した AFE レジスタファイル(.ini ファイル)を読み込ませる。

Renesas VA でシミュレーションした Phototransistor(NJL7502L)の AFE レジスタファイル (Renesas.Register.RAA730500.Sensor-XX_XX_XXXX.ini ファイル) を開いてください。

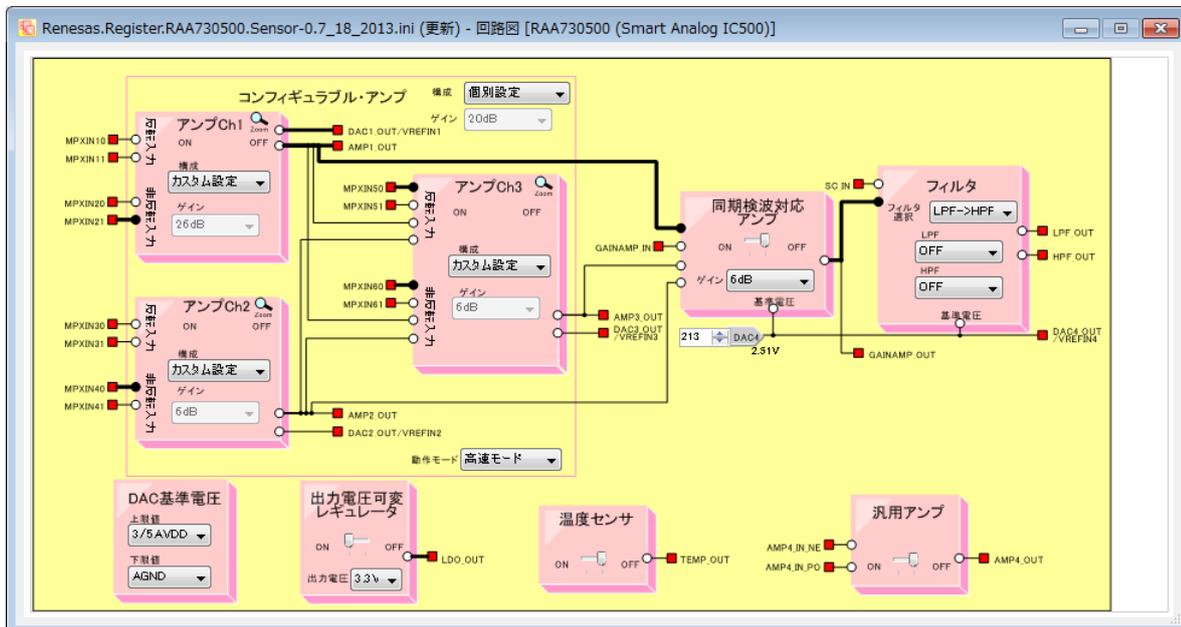


「はい」をクリックしてください。



生成パスの指定を聞かれますので、任意の場所を設定してください。使用するコンパイラは「CA78K0R (IDE: CubeSuite+)」を選択して OK ボタンをクリックしてください。

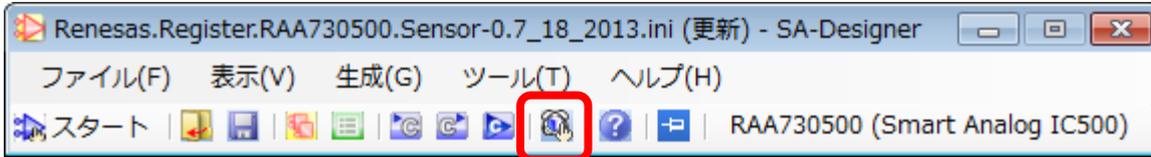
SA-Designer の回路設計画面が以下のように開きます。



7.6 CubeSuite+を起動する

SA-Designer を CubeSuite+ と連携させます。

以下の赤枠で囲ったボタンをクリックして CubeSuite+ を起動してください。

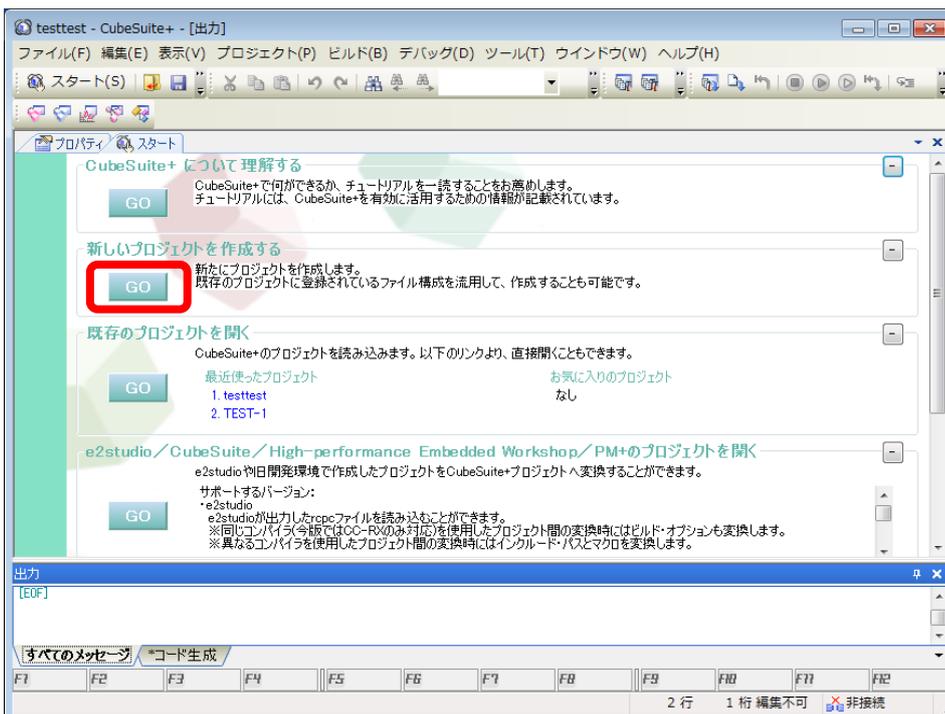


今回のプログラム用のプロジェクトを作成します。

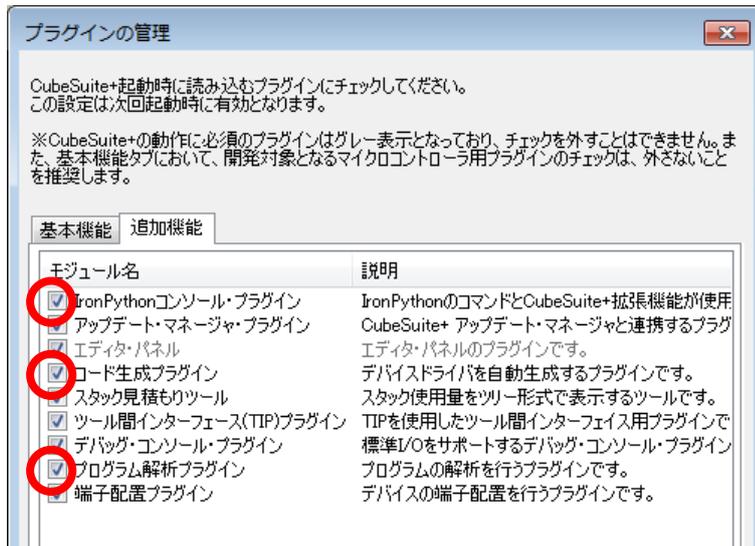


OK ボタンを押してください。

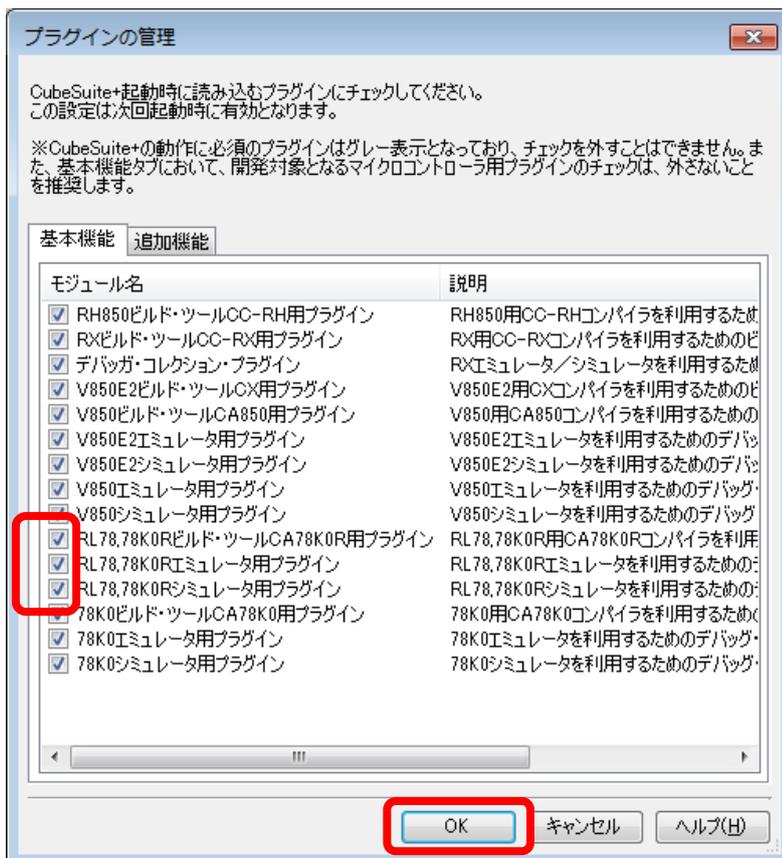
スタート画面から「新しいプロジェクトを作成する」の GO ボタンをクリックしてください。



【注】 CubeSuite+の起動時に、上記スタート画面が表示されない場合はプラグインの設定が原因です。「ツール」メニューから「プラグインの管理」を選択し、下記の通り設定してください。
スタート画面が表示される方は、次頁の「プロジェクト作成」画面へ進んでください。



CubeSuite+の起動時に読み込むプラグインをチェックします。「追加機能」タブから、「IronPython コンソール・プラグイン」、「コード生成プラグイン」、「プログラム解析プラグイン」にチェックしてください。



「基本機能」タブから、「RL78,78K0R ビルド・ツール CA78K0R 用プラグイン」、「RL78,78K0R エミュレータ用プラグイン」、「RL78,78K0R シミュレータ用プラグイン」にチェックしてください。

OK をクリックしてください。

スタート画面の「新しいプロジェクトを作成する」の GO ボタンをクリック後、以下のように設定してください。

コード生成ツールがまだ RL78/G1E に対応していないので今回は RL78/G1A を選択してコード生成を行います。

プロジェクト作成

マイクロコントローラ(T): RL78

使用するマイクロコントローラ(M):

- RL78/G1A (ROM:64KB)
- R5F10E8E(25pin)
- R5F10EBE(32pin)
- R5F10EGE(48pin)
- R5F10ELE(64pin)
- RL78/G1E (ROM:32KB)
- RL78/G1E (ROM:48KB)
- RL78/G1E (ROM:64KB)

プロジェクトの種類(K): アプリケーション(CA78K0R)

プロジェクト名(N): TEST-1

作成場所(L): C:\...#Desktop#TEST

作成(C) キャンセル ヘルプ(H)

マイクロコントローラ → RL78 を選択

使用するマイクロコントローラ
→ RL78/G1A(ROM64KB)
→ R5F10ELE(64pin)
を選択

プロジェクト種類
→ アプリケーション(CA78K0R)
プロジェクト名
→ 任意
作成場所
→ 任意

上記を設定したら、作成ボタンをクリックしてください。

7.7 コード生成の設定をする

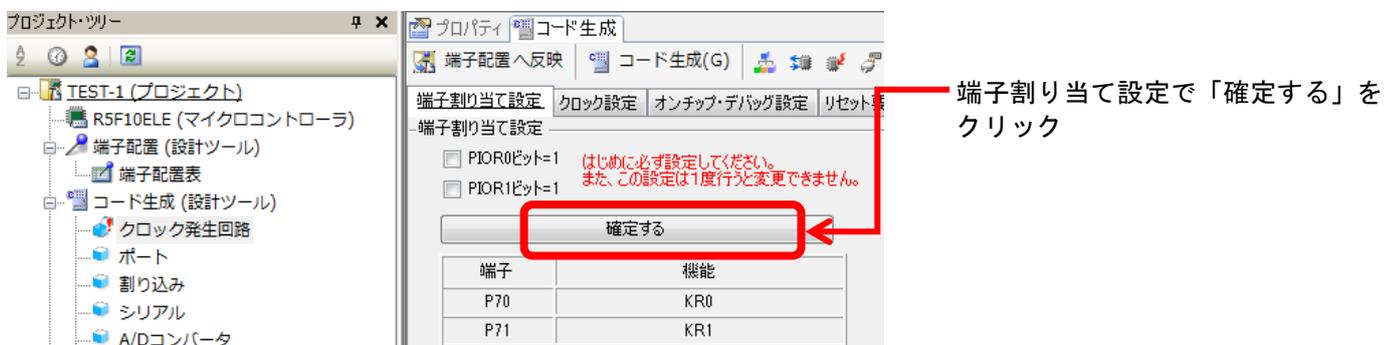
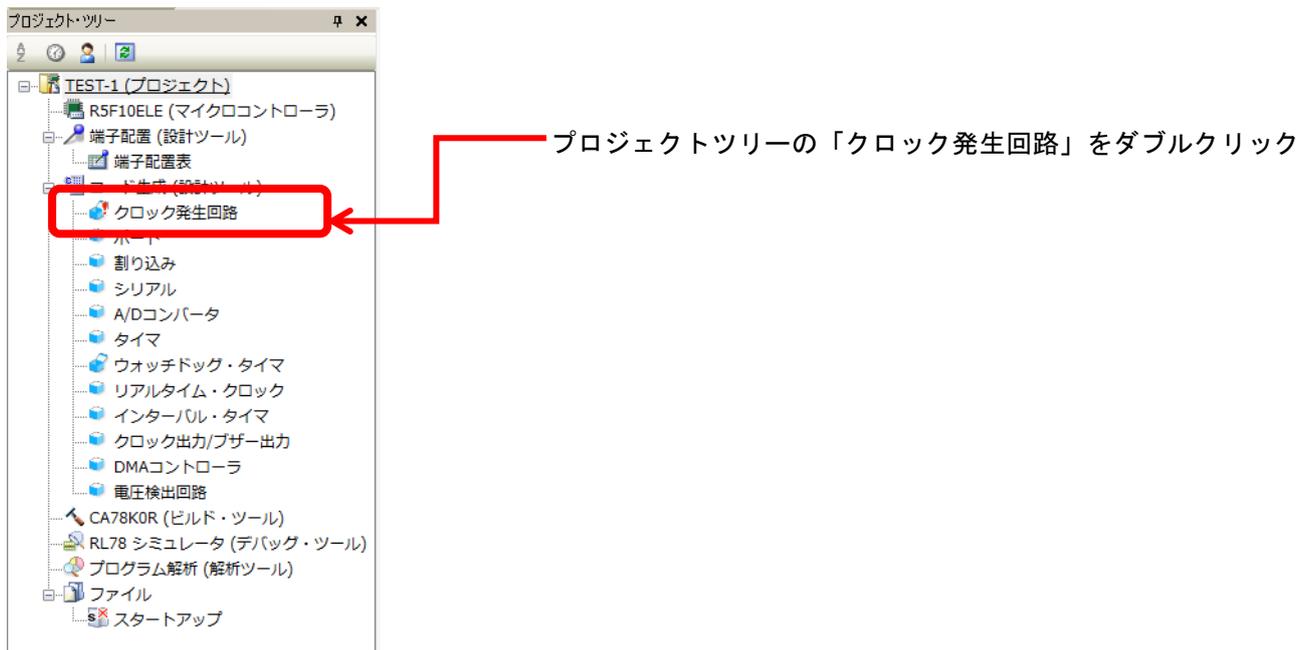
プロジェクト作成したら、実際にマイコンを動作させるプログラムを作ります。

今回は、GUI 画面で指定するだけでマイコンの初期設定を簡単にコード生成する方法でプログラムを作ります。

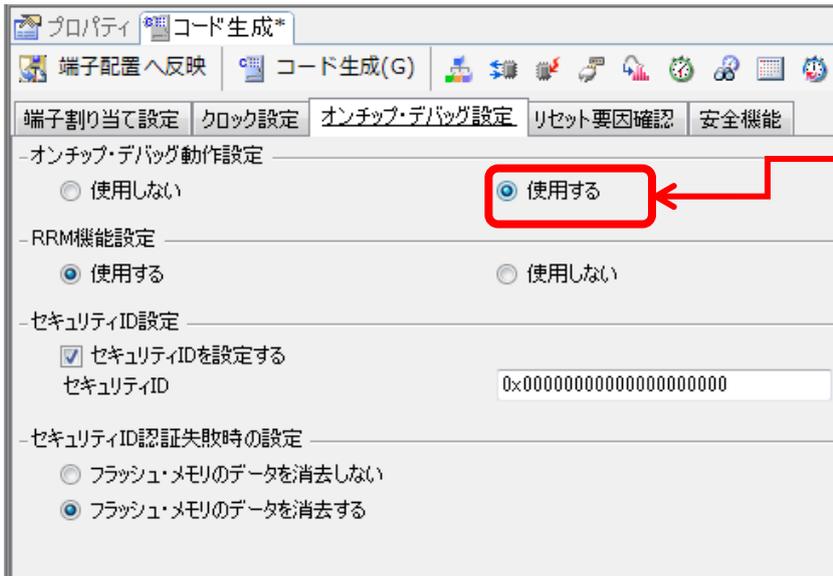
プロジェクトツリーから設定したい機能をダブルクリックして設定していきます。

以下の手順にて各機能を設定していきます。

7.7.1 クロック発生回路の端子割り当て設定

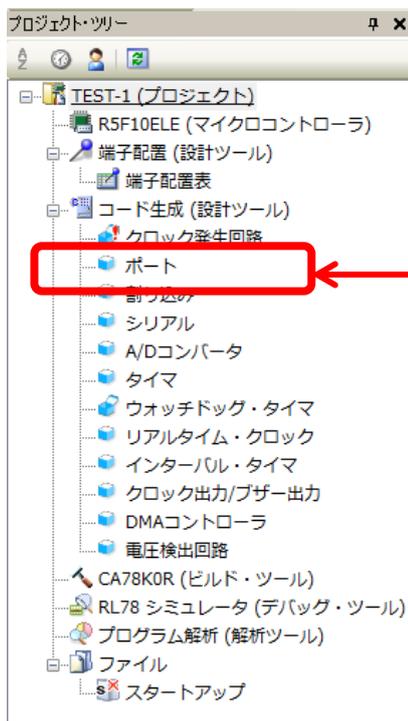


クロック発生回路のオンチップ・デバッグ設定



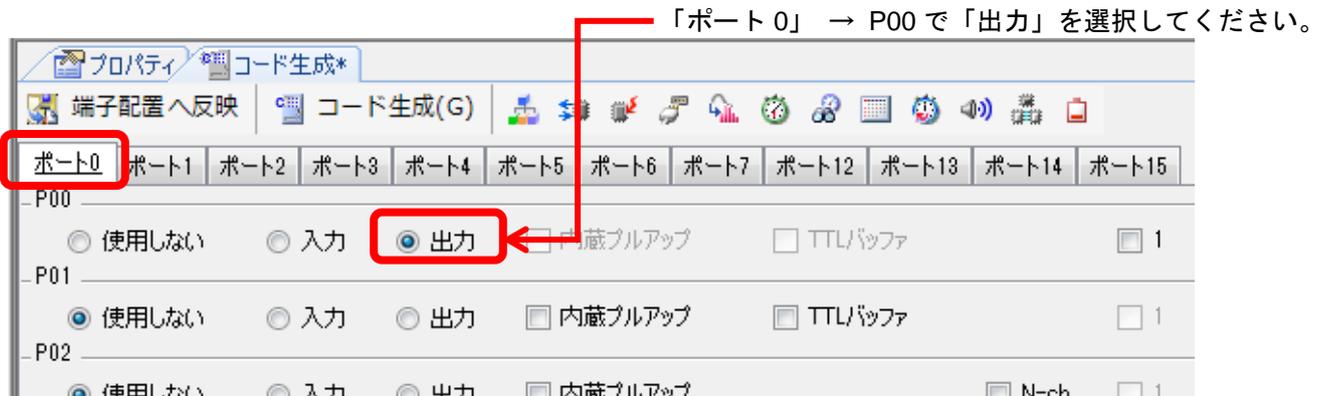
オンチップ・デバッグ動作設定
→ 使用する

7.7.2 ポートの端子割り当て設定



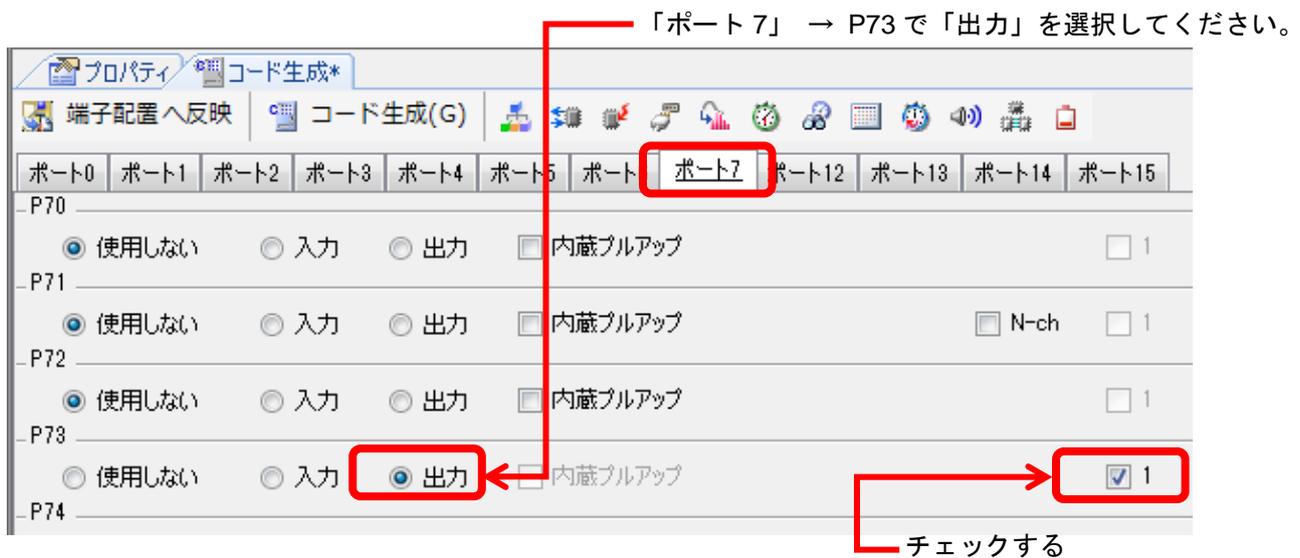
プロジェクトツリーの「ポート」をダブルクリック

「ポート 0」を設定します。ポート 0 はセンサの挙動を受けて点滅する LED を制御します。

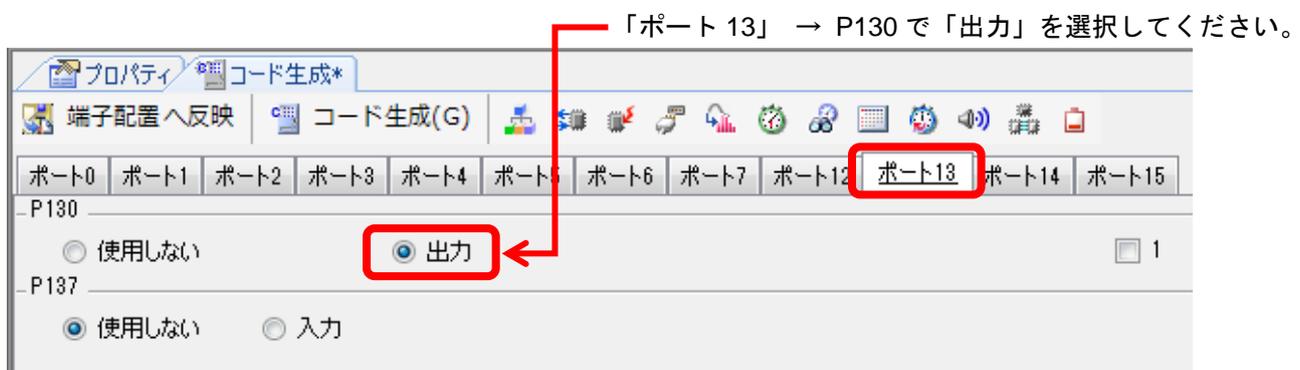


「ポート 7」を設定します。ポート 7 は Smart Analog IC の「CS」端子を制御します。

このポートは「Active High」で使用しますので、ポート設定の右端の「1」にチェックを入れてください。



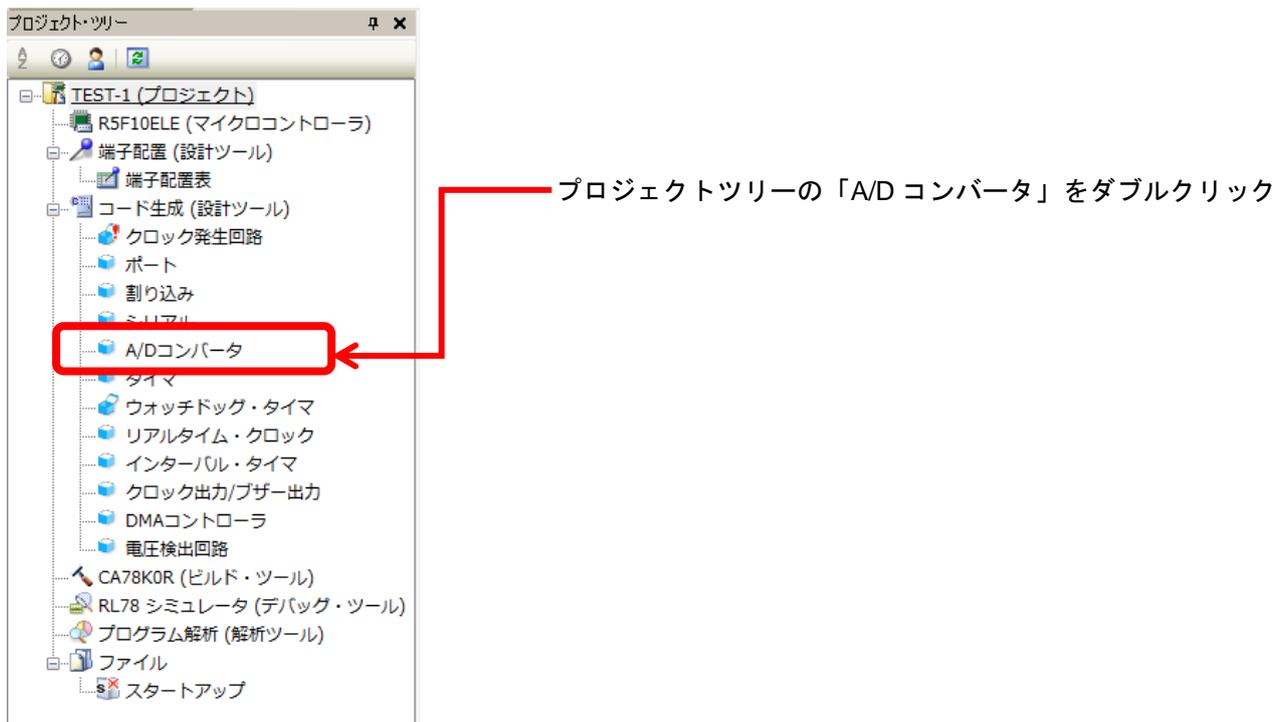
「ポート 13」を設定します。ポート 13 は Smart Analog IC の「RESET」端子を制御します。

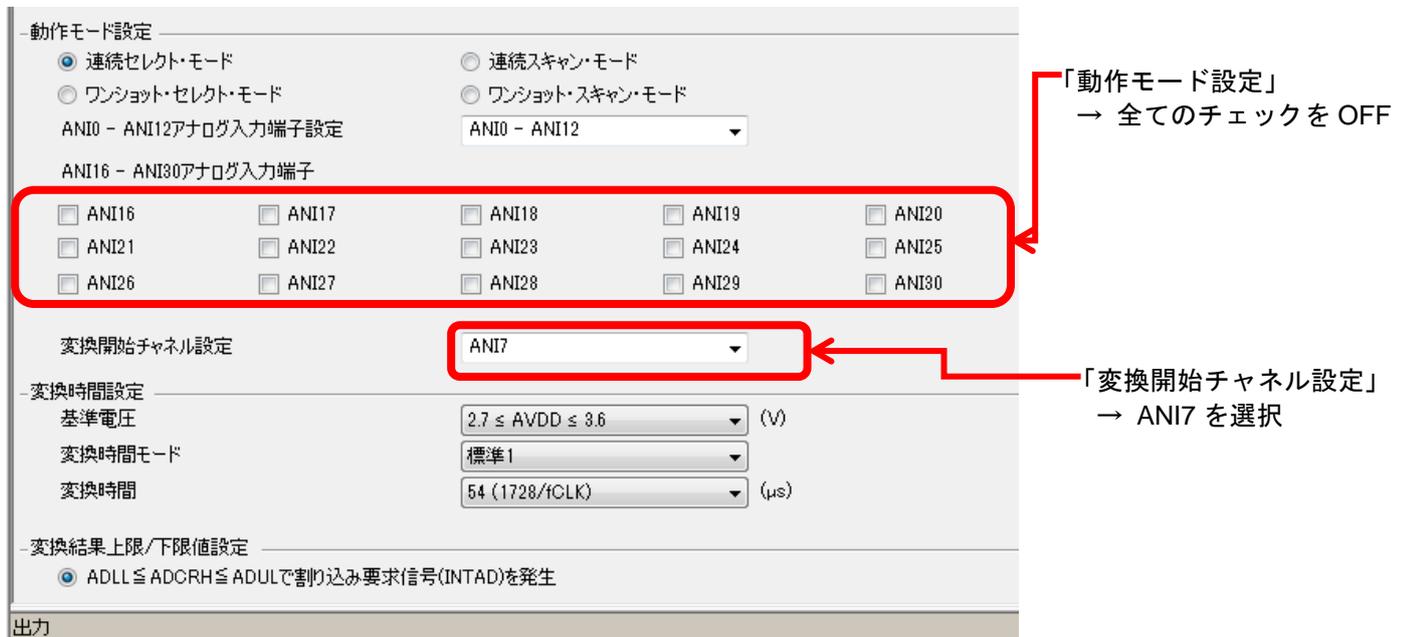
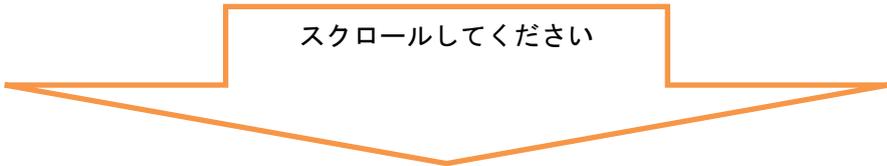
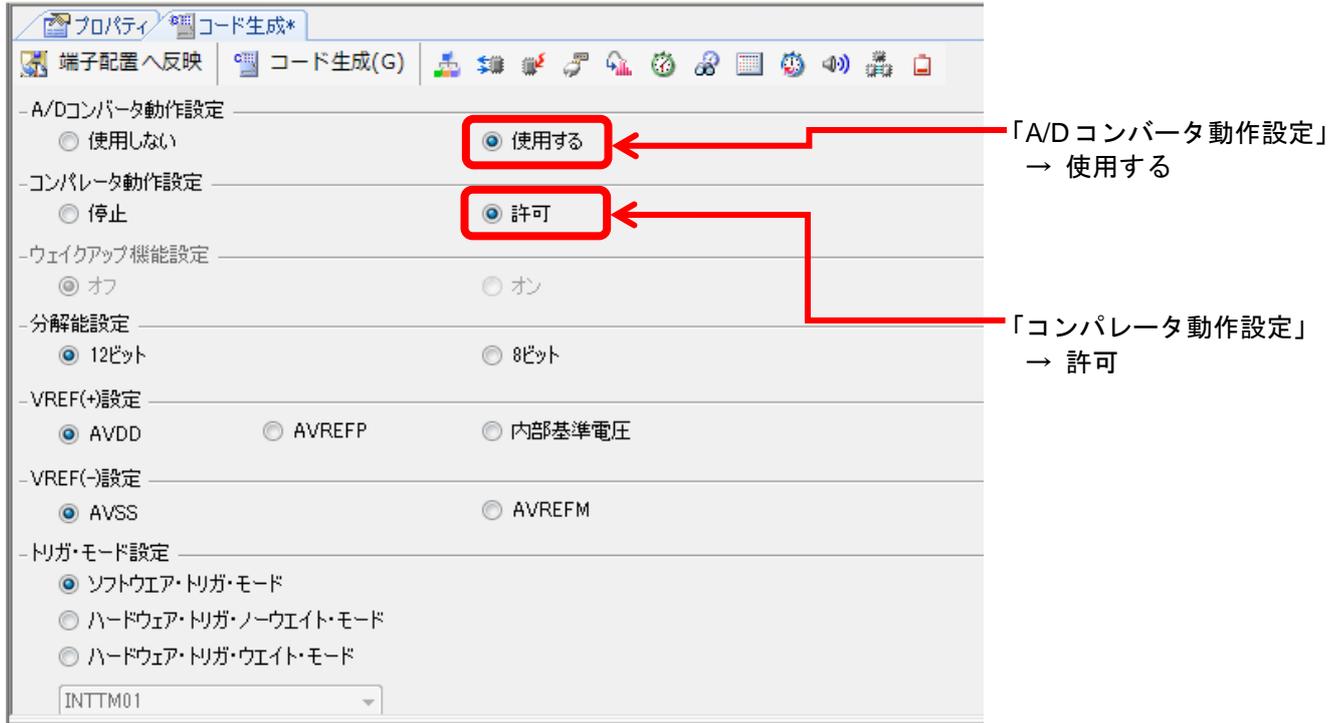


「ポート 14」を設定します。ポート 14 もポート 0 と同様にセンサの挙動を受けて点滅する LED を制御します。



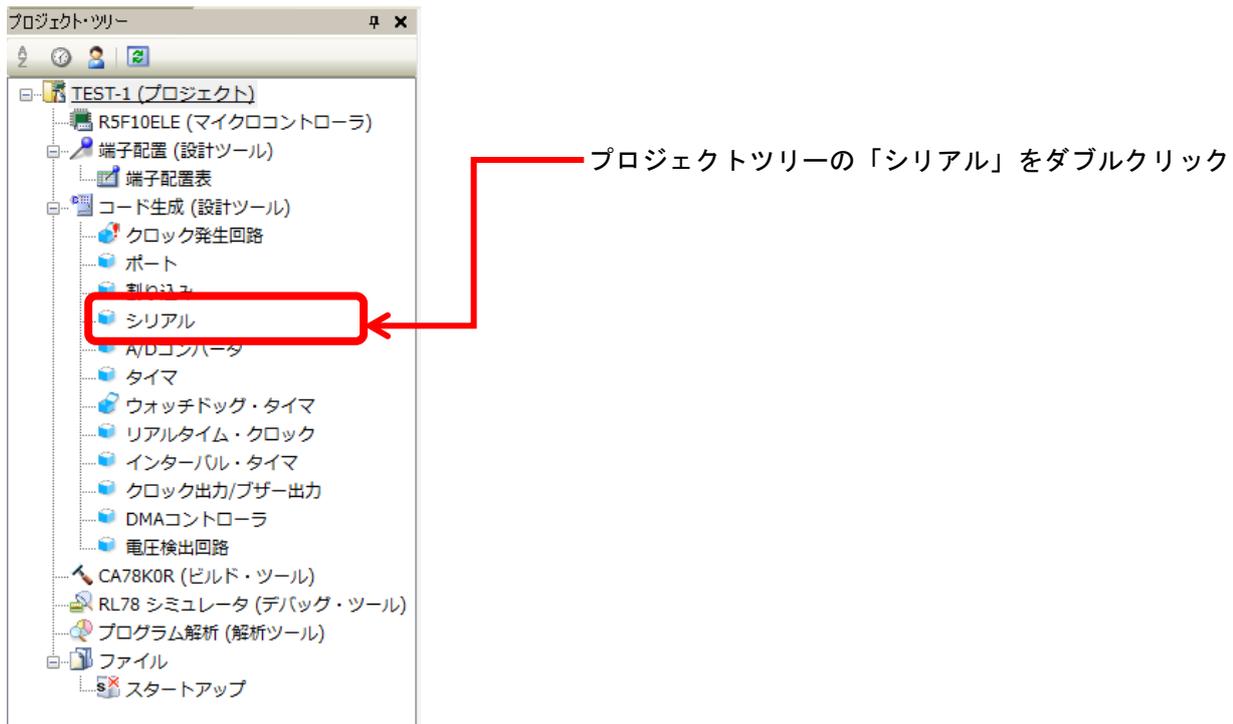
7.7.3 A/D コンバータの設定



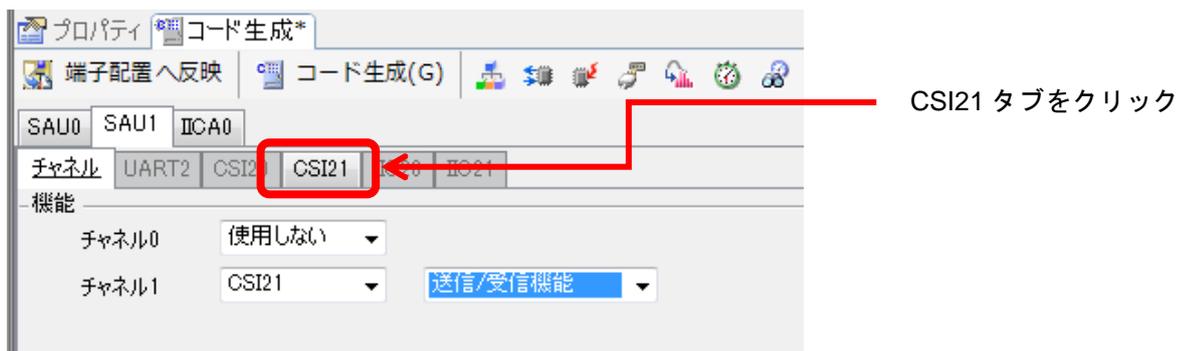


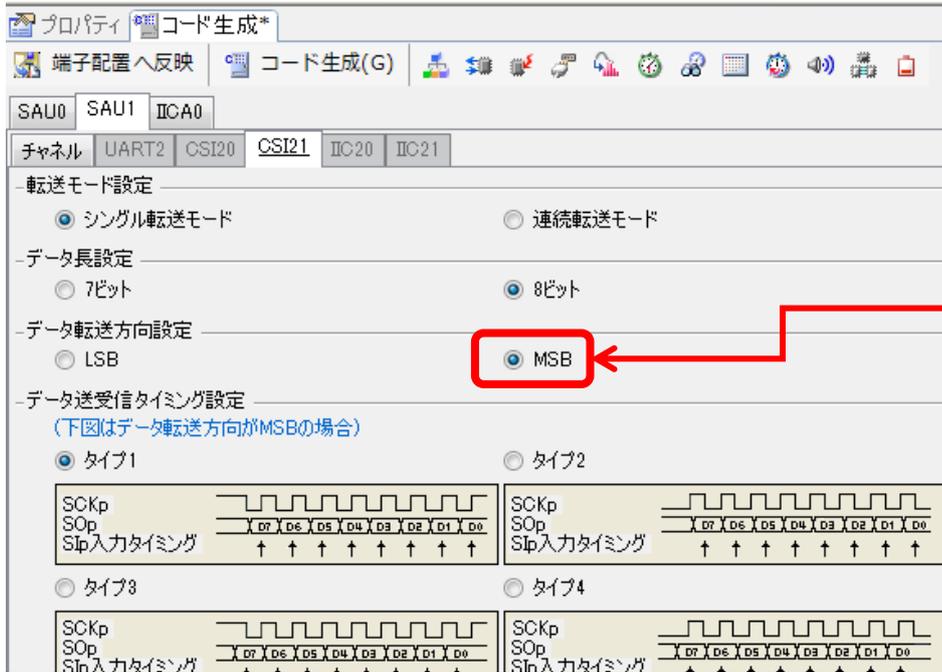
AD コンバータの端子設定については、今回のプログラムは ANI7 のみを使用しますので、AIN16-AIN30 は全て OFF にしてください。

7.7.4 シリアルの設定



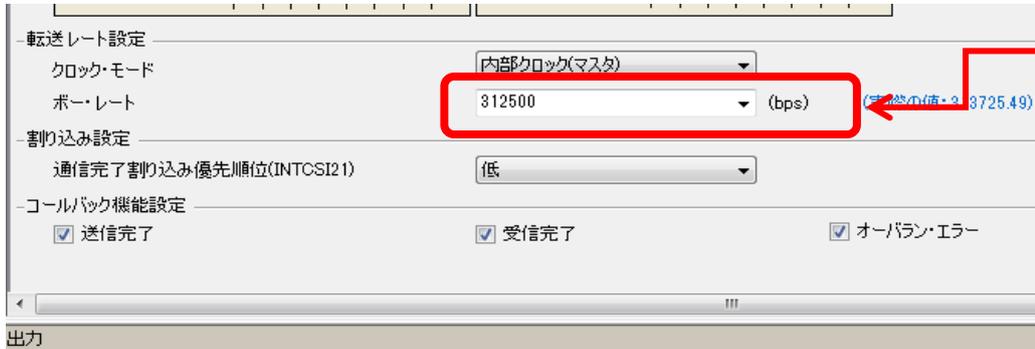
シリアルユニットは「CSI21」を使用しますので、CSI21 を選択して各種設定をします。





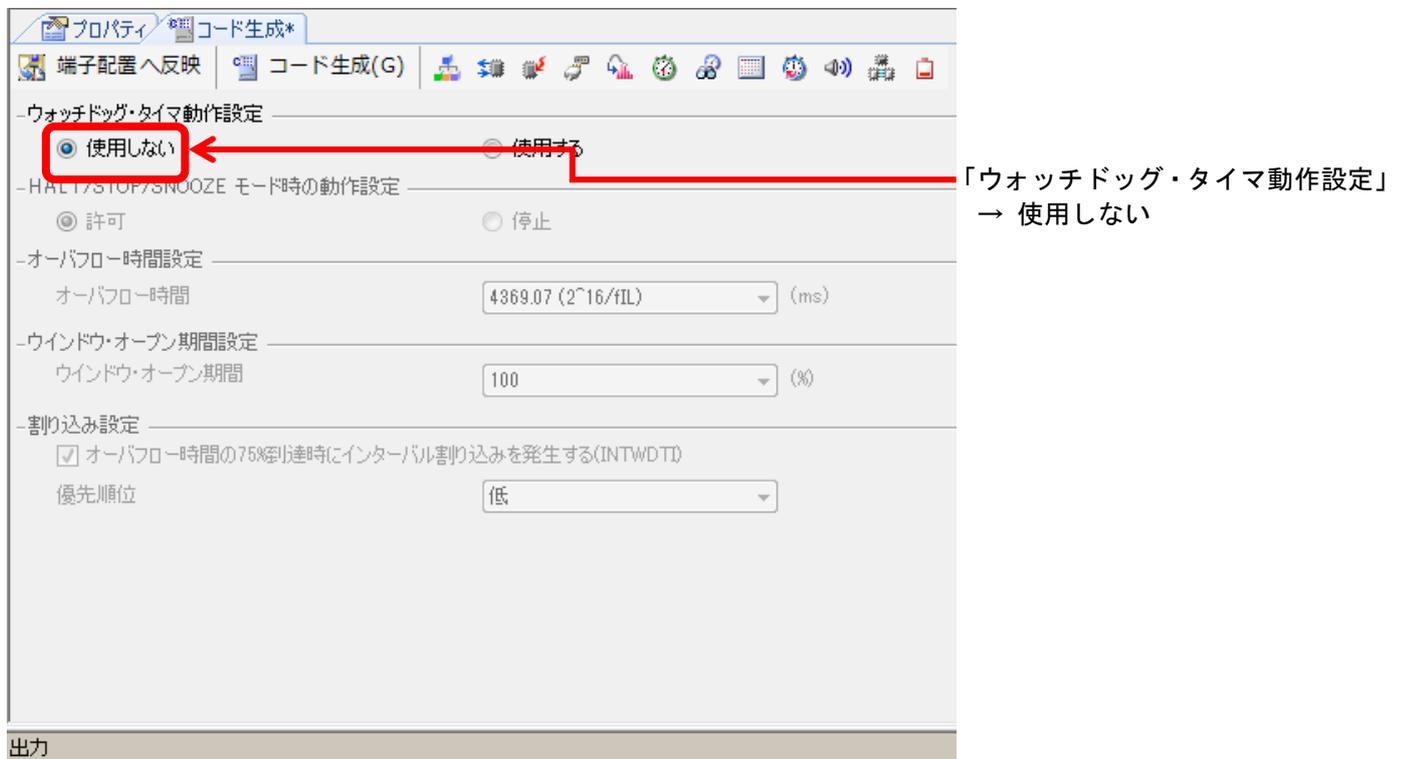
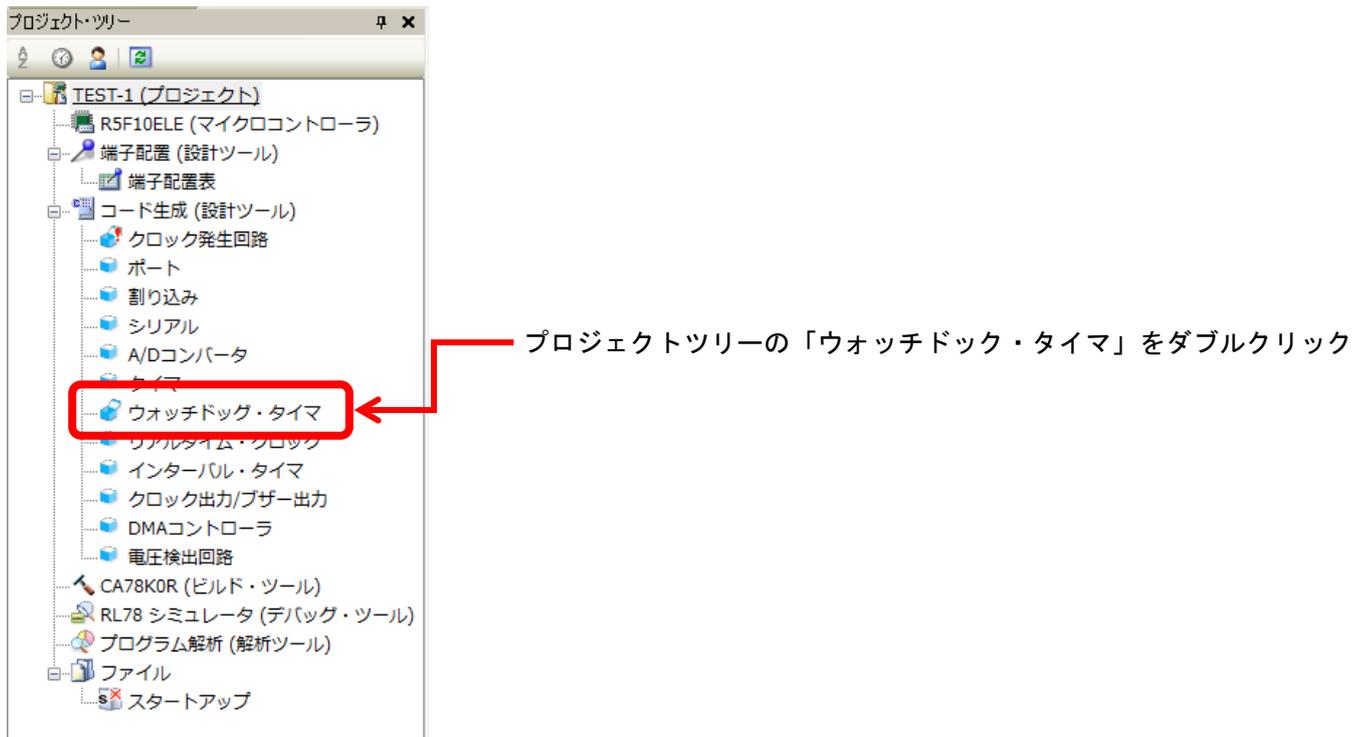
「データ転送方向設定」
→ MSB

スクロールしてください



「転送レート設定」
「ボー・レート」
→ 312500

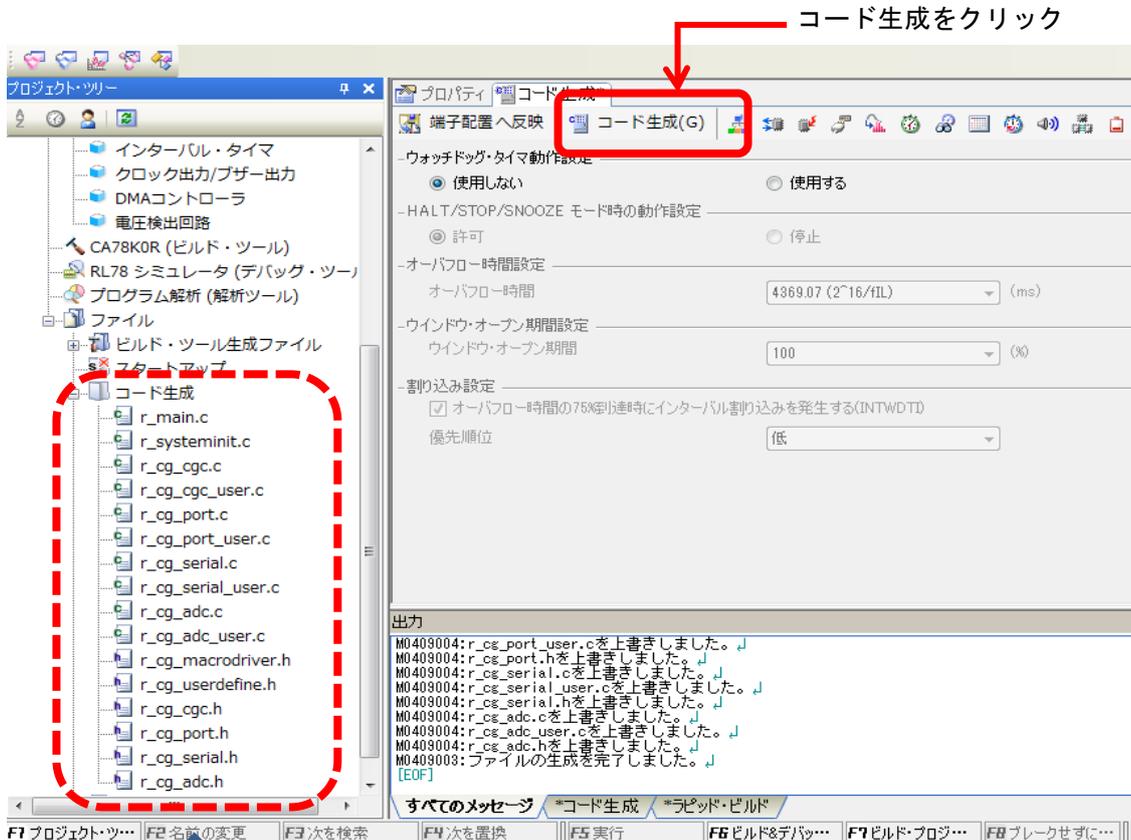
7.7.5 ウォッチドッグ・タイマの設定



7.8 コード生成実行

最後にコード生成ボタンをクリックしてコードを生成します。

プロジェクトツリーの「ファイル」の中に C ソースやヘッダーが出力されます。



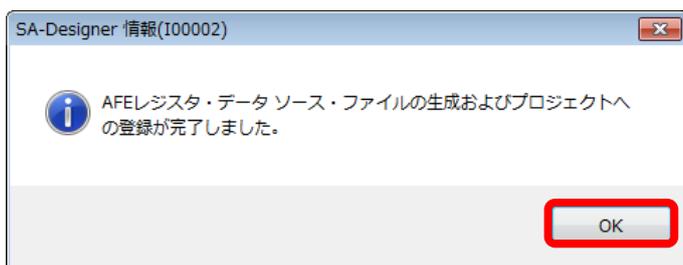
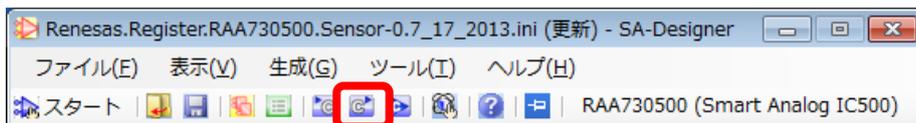
マイコン初期化用のコードが生成されます。
また各種機能を制御する API も生成されます。

7.8.1 回路パラメータの C ソース生成と登録

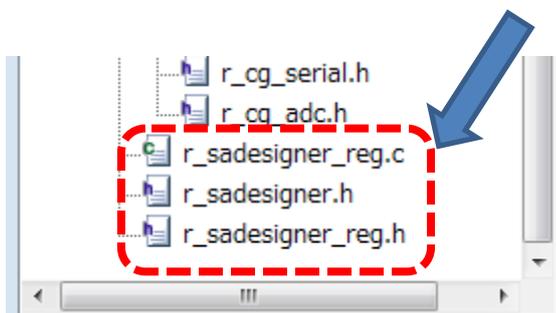
ここで一旦 SA-Designer に戻ります。

SA-Designer で Smart Analog IC の回路パラメータの C ソースを生成し、生成した C ソースを Cube Suite+ に自動登録します。

下記の赤枠のアイコンをクリックしてください。



「OK」をクリックしてください。

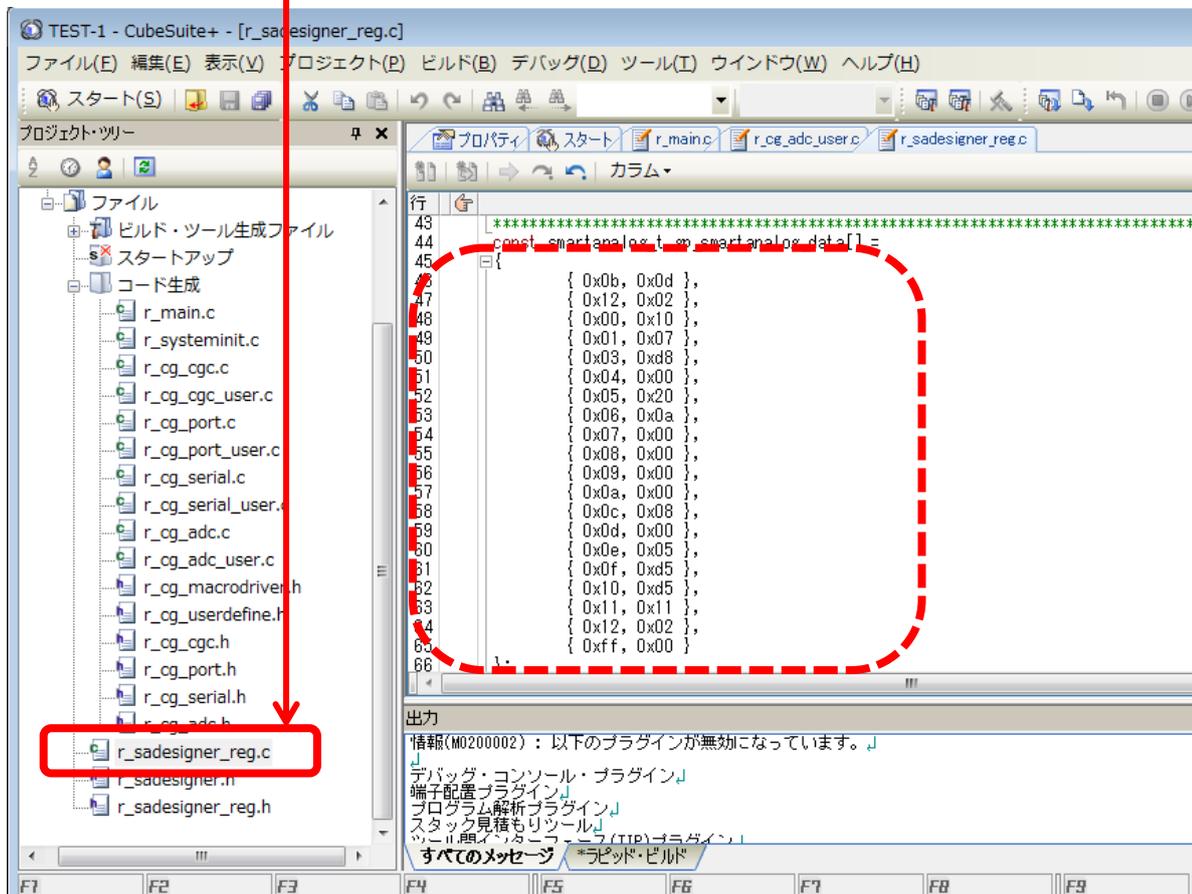


CubeSuite+のプロジェクトツリーに生成した回路パラメータの C ソースファイルが登録されます。

7.9 SA-Designer が生成したファイルの補足

Smart Analog IC の回路パラメータは CubeSuite+上では構造体変数(const)で定義されます。

r_sadesigner_reg.c をダブルクリック



回路パラメータは{アドレス, データ}で定義された構造体変数にて生成されます。
最後の{0xff, 0x00}はデリミタ用の値です。

7.10 プログラムを編集する

7.10.1 r_main.c の編集

プロジェクトツリーにある「r_main.c」を開いてください。
下記の青文字の部分を追加してください。

```

/*****
*****
Includes
*****
*****/
#include "r_cg_macrodriver.h"
#include "r_cg_cgc.h"
#include "r_cg_port.h"
#include "r_cg_serial.h"
#include "r_cg_adc.h"
/* Start user code for include. Do not edit comment generated here */
#include "r_sadesigner.h"
/* End user code. Do not edit comment generated here */
#include "r_cg_userdefine.h"
/*****
*****
Global variables and functions
*****
*****/
/* Start user code for global. Do not edit comment generated here */
void R_SAIC_Write(smartanalog_t * const p_saic_data);
void R_SAIC_Create(void);
extern const smartanalog_t gp_smartanalog_data[];
/* End user code. Do not edit comment generated here */
/*****
*****
* Function Name: main
* Description : This function implements main function.
* Arguments : None
* Return Value : None
*****
*****/
void main(void)
{
R_MAIN_UserInit()
/* Start user code. Do not edit comment generated here */
R_CSI21_Start();
R_SAIC_Create();
R_ADC_Start();
while (1U)
{
;
}
/* End user code. Do not edit comment generated here */
}
/*****
*****
* Function Name: R_MAIN_UserInit
* Description : This function adds user code before implementing main function.
* Arguments : None
* Return Value : None
*****
*****/
void R_MAIN_UserInit(void)
{
/* Start user code. Do not edit comment generated here */
EI();
/* End user code. Do not edit comment generated here */
}

```

SA-Designer 生成ソースをインクルード

SPI 関数の宣言、変数の広域宣言

SPI、Smart Analog IC 初期設定、A/D のスタート

```

}
/* Start user code for adding. Do not edit comment generated here */
void R_SAIC_Create(void)
{
volatile uint16_t w_count;
/* Analog IC Reset */
P13.0 = 0U;
/* Change the waiting time according to the system */
for (w_count = 0U; w_count < 0x1A; w_count++)
{
NOP();
}
/* Analog IC Reset release */
P13.0 = 1U;
R_SAIC_Write(gp_smartanalog_data);
}
void R_SAIC_Write(smartanalog_t * const p_saic_data)
{
uint8_t adrs;
uint8_t dat;
uint8_t wait;
smartanalog_t *p_saic_write;
PMC3 = 0xfc;
CSIMK21 = 1U; /* mask INTCSI21 */
p_saic_write = p_saic_data;
while (p_saic_write->address != 0xff)
{
P7.3 = 0U; /* SAIC CS=L */
/* SA stable waiting time (tSKA) */
for (wait = 0U; wait < 1U; wait++)
{
NOP();
}
adrs = (p_saic_write->address & 0x7f) | 0x80; /* 0x80 is data write mode */
SIO21 = adrs; /* send SAIC Address data */
while (CSIIIF21 == 0U); /* wait for CSI send */
CSIIIF21 = 0U;
dat = p_saic_write->data;
SIO21 = dat; /* send SAIC Register Data */
while (CSIIIF21 == 0U); /* wait for CSI send */
CSIIIF21 = 0;
/* SA Stable waiting time (tKSA) */
for (wait = 0U; wait < 1U; wait++)
{
NOP();
}
P7.3 = 1U; /* SAIC CS=H */
/* SA Stable waiting time (tSHA) */
for (wait = 0U; wait < 1U; wait++)
{
NOP();
}
p_saic_write++;
}
}
/* End user code. Do not edit comment generated here */

```

Smart Analog IC リセット

Smart Analog IC リセット解除、設定関数コール

デリミタ検出

チップセレクト

アドレス送信

データ送信

チップセレクト解除

ここまで記入したら一旦「r_main.c」を保存します。

7.10.2 r_cg_adc_user.c の編集

プロジェクトツリーにある「r_cg_adc_user.c」を開いてください。

下記の青文字の部分を追加してください。

```

/*****
*****
Global variables and functions
*****
*****/
/* Start user code for global. Do not edit comment generated here */
uint16_t g_tmp = 0;
/* End user code. Do not edit comment generated here */
/*****
*****/
* Function Name: r_adc_interrupt
* Description : This function is INTAD interrupt service routine.
* Arguments : None
* Return Value : None
*****
*****/
__interrupt static void r_adc_interrupt(void)
{
/* Start user code. Do not edit comment generated here */
uint16_t adcr_2bit = 0;
R_ADC_Get_Result(&g_tmp);
/* A/D 値を 4 分割して LED 表示更新の参考例 */
/* g_tmp 下位 12bit の A/D 値の、上位 2bit を使用 */
adcr_2bit = g_tmp >> 10;
/* 2bit 値から LED 制御ポートの端子出力を変更。1:LED が OFF,0:LED が ON */
switch (adcr_2bit)
{
case 0U: P0.0 = 1;
P14.0 = 1;
break;
case 1U: P0.0 = 0;
P14.0 = 1;
break;
case 2U: P0.0 = 1;
P14.0 = 0;
break;
case 3U: P0.0 = 0;
P14.0 = 0;
break;
default: /* Through */
break;
}
/* End user code. Do not edit comment generated here */
}
/* Start user code for adding. Do not edit comment generated here */
/* End user code. Do not edit comment generated here */

```

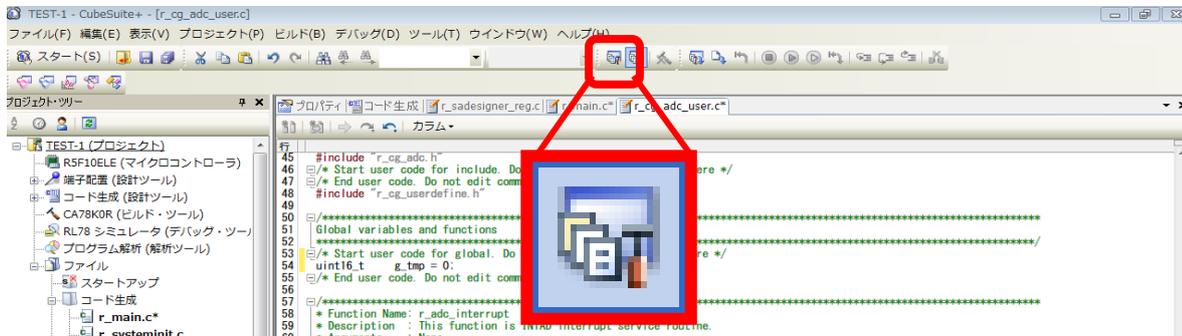
A/D 変換結果取得用変数の定義

A/D 変換結果を 4 分割して LED の点灯制御

ここまで記入したら「r_cg_adc_user.c」を保存します。

7.11 プログラムをビルドする

上記のプログラムの編集が終わったら、ビルドボタンをクリックしてプログラム(ロードモジュール)を生成します。下記の赤枠で囲ったボタンをクリックしてください。



ここでエラーが出力されなかったら、プログラムは無事完成しています。

もし、エラー等が表示される場合は、エラーメッセージを確認し、7.7 や 7.10 で設定した内容を確認してみてください。

7.12 プログラム実行環境を作る(エミュレータと接続する)。

評価ボード、E1 エミュレータ、パソコンを接続してください。



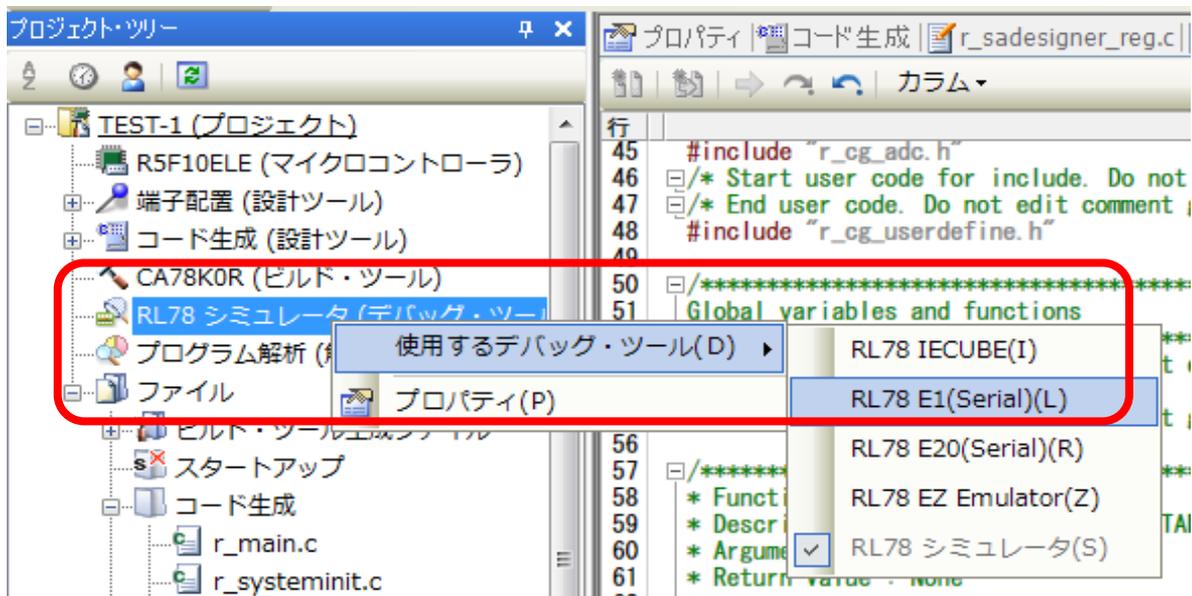
USB ケーブルを 2 つ用意してください。

- 1 つは E1 へ接続
- 1 つは評価ボード(TSA-IC 500)へ接続

E1 と評価ボード(TSA-IC 500)を E1 付属のケーブルで接続してください。

7.12.1 デバッグ・ツールを設定する。

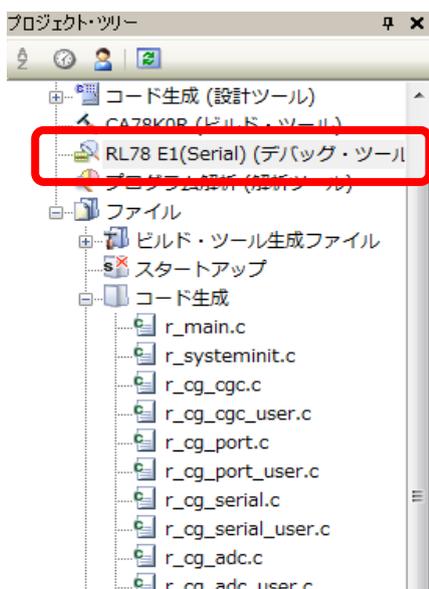
デバッグ・ツールの設定をします。



プロジェクトツリーの「RL78 シミュレータ(デバッグ・ツール)」を右クリック。

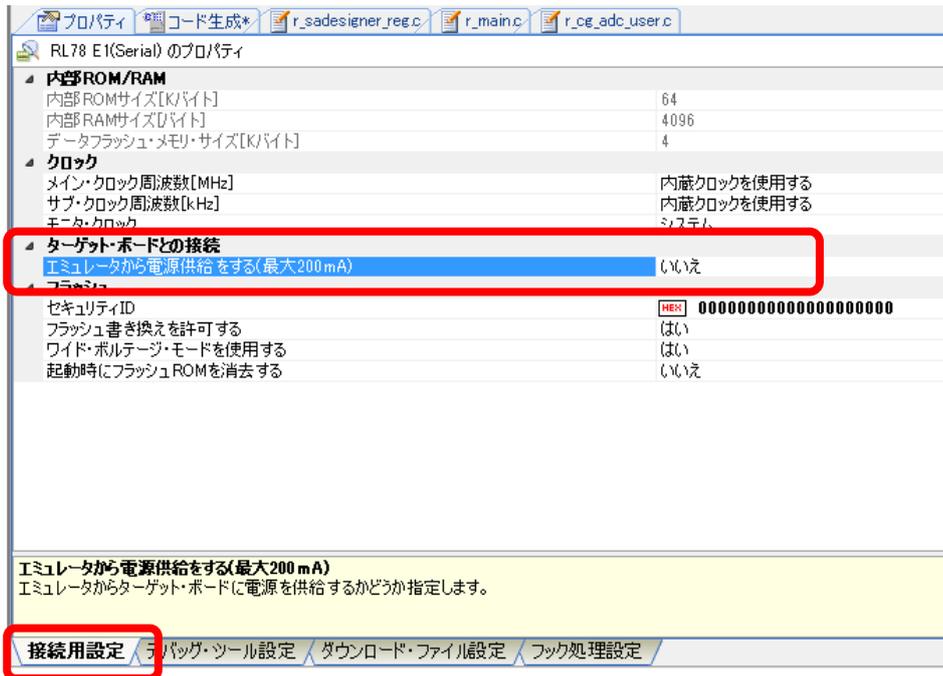
→ 「使用するデバッグ・ツール」 → 「RL78 E1(Serial)(L)」を選択

続いて、E1 エミュレータを接続する際の電源設定、プログラム実行中にメモリ空間を読み込めるようにする設定をします。

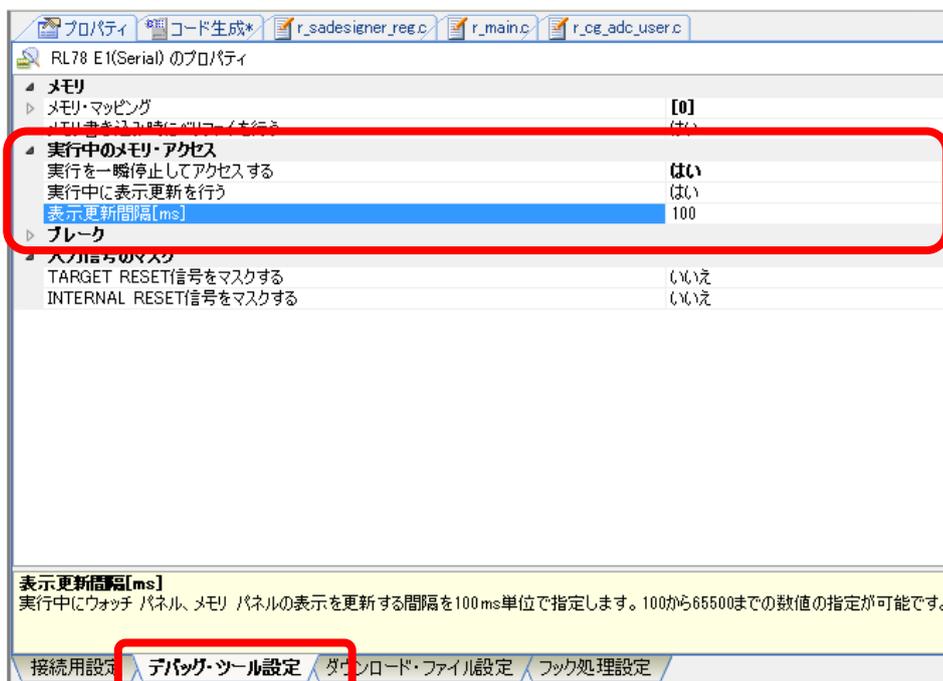


プロジェクトツリーの「RL78 E1(Serial)(デバッグ・ツール)」をダブルクリック。

続いて以下を設定してください。



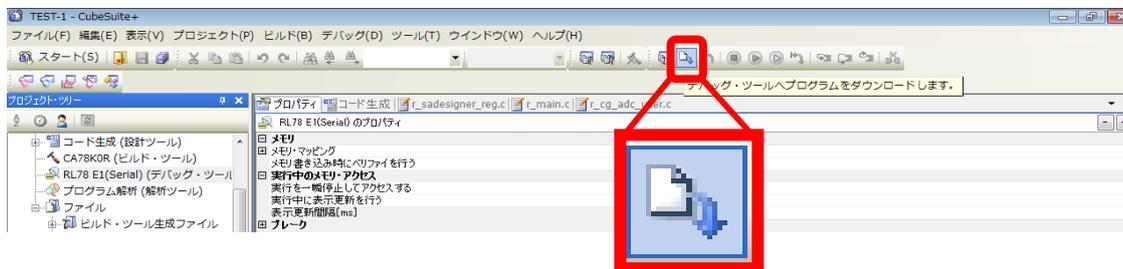
「接続用設定」タブの「ターゲット・ボードとの接続」 → エミュレータから電源供給をする(最大 200mA)
→ いいえ



「デバッグ・ツール設定」タブの「実行中のメモリ・アクセス」
→ 実行を一瞬停止してアクセスする → はい
→ 表示更新間隔[ms] → 100

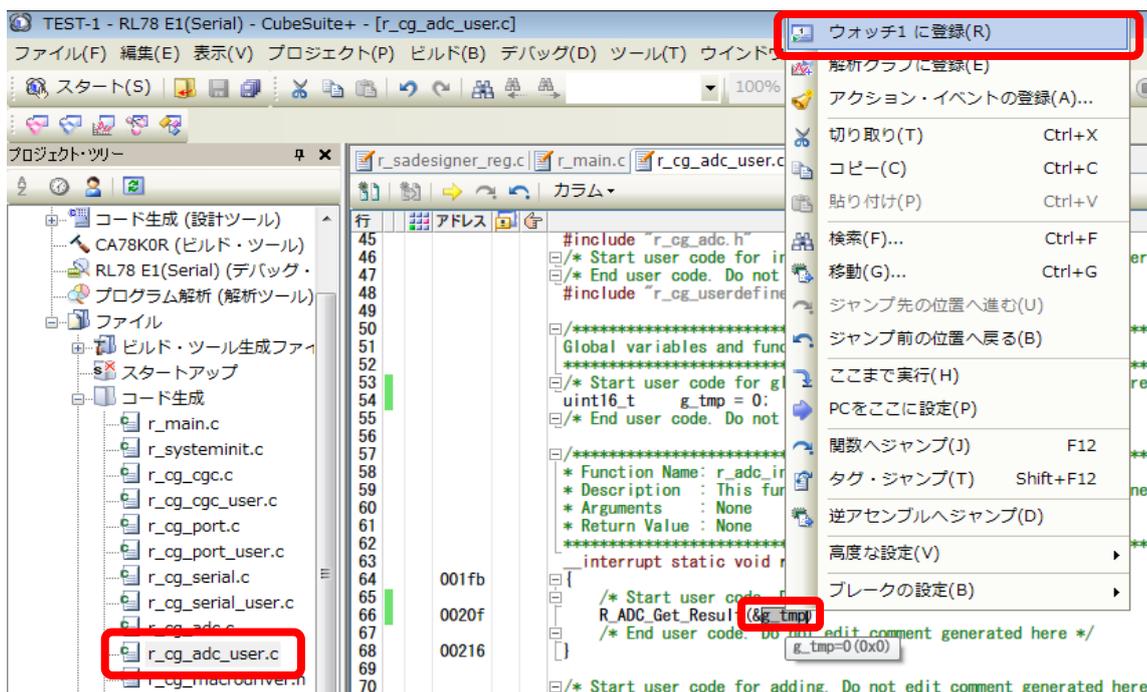
7.13 マイコンにプログラムをダウンロードする。

デバッグ・ツールの設定が終了したら、作ったプログラム(ロードモジュール)をマイコンにダウンロードします。下記の赤枠で囲ったアイコンをクリックしてください。



7.14 A/D 変換で取得した値をモニターする

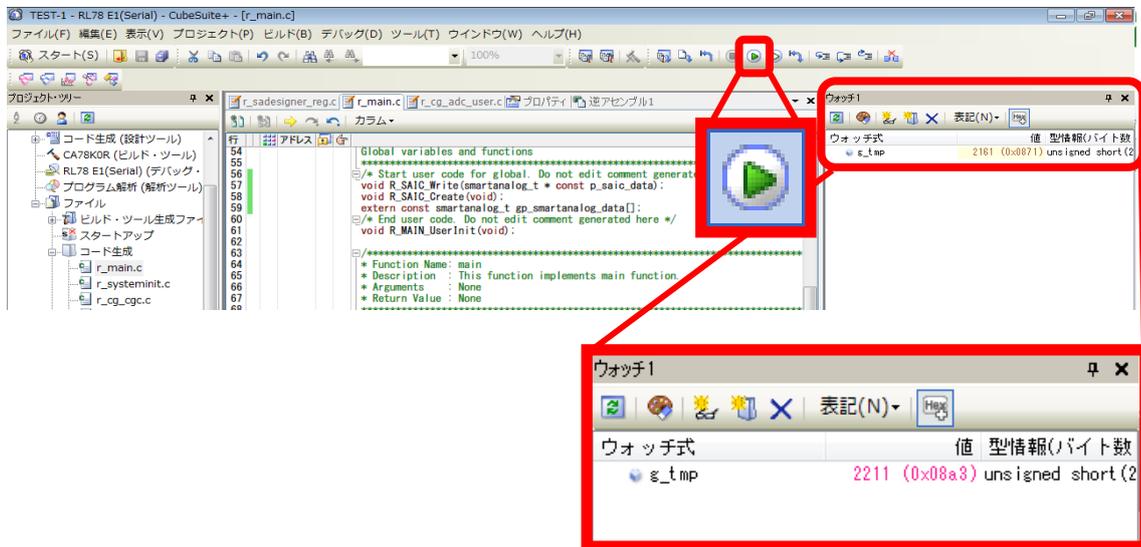
プログラムの中で A/D 変換にて取得した値は「g_tmp」に格納されます。そこで g_tmp を「ウォッチ登録」して動作をモニターできるようにします。



プロジェクトツリーの「r_cg_adc_user.c」をダブルクリック → 変数「g_tmp」の箇所をダブルクリックしてさらに右クリック → 「ウォッチ 1 に登録」

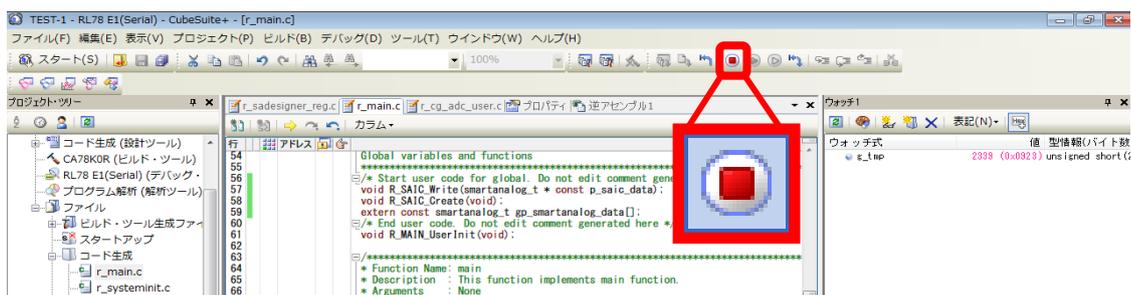
7.15 プログラムを実行してセンサの動作を確認

では実際にマイコンにダウンロードしたプログラムを実行してセンサの動作を確認します。
下記の赤枠で囲ったアイコンをクリックしてください。



先ほど「ウォッチ登録」した「`g_tmp`」の値が表示されます。
センサの動きに合わせて値が変化していることを確認してください。
(センサを手で覆ったり、蛍光灯などの光源に近づけたりしてみてください。)
なお、この時点で既に LED も点灯するので合わせて確認してください。

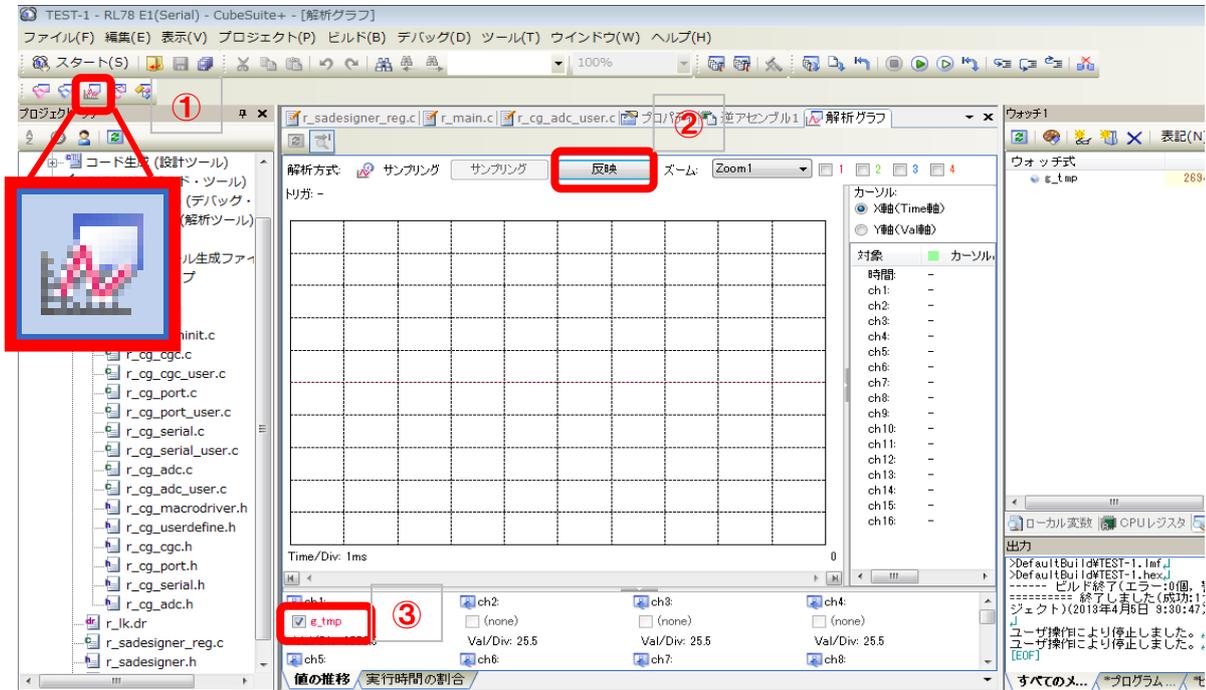
ここで、センサの動きをより分かり易くするためにグラフ表示をさせてみます。
一旦プログラムを停止します。下記の赤枠で囲ったアイコンをクリックしてください。



7.16 解析グラフパネルでセンサの動きをグラフ表示する。

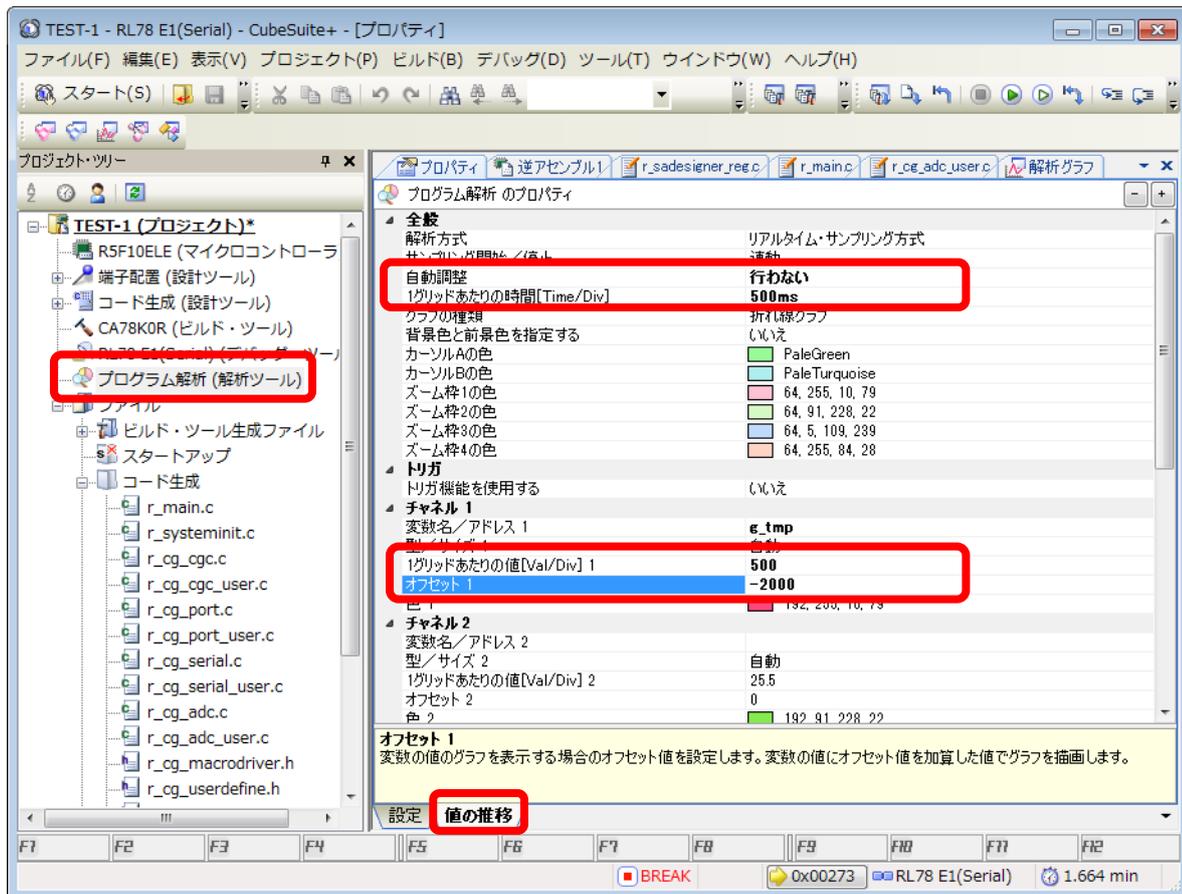
センサの動きをグラフ表示させる設定をします。

下記の①～③の設定をしてください。



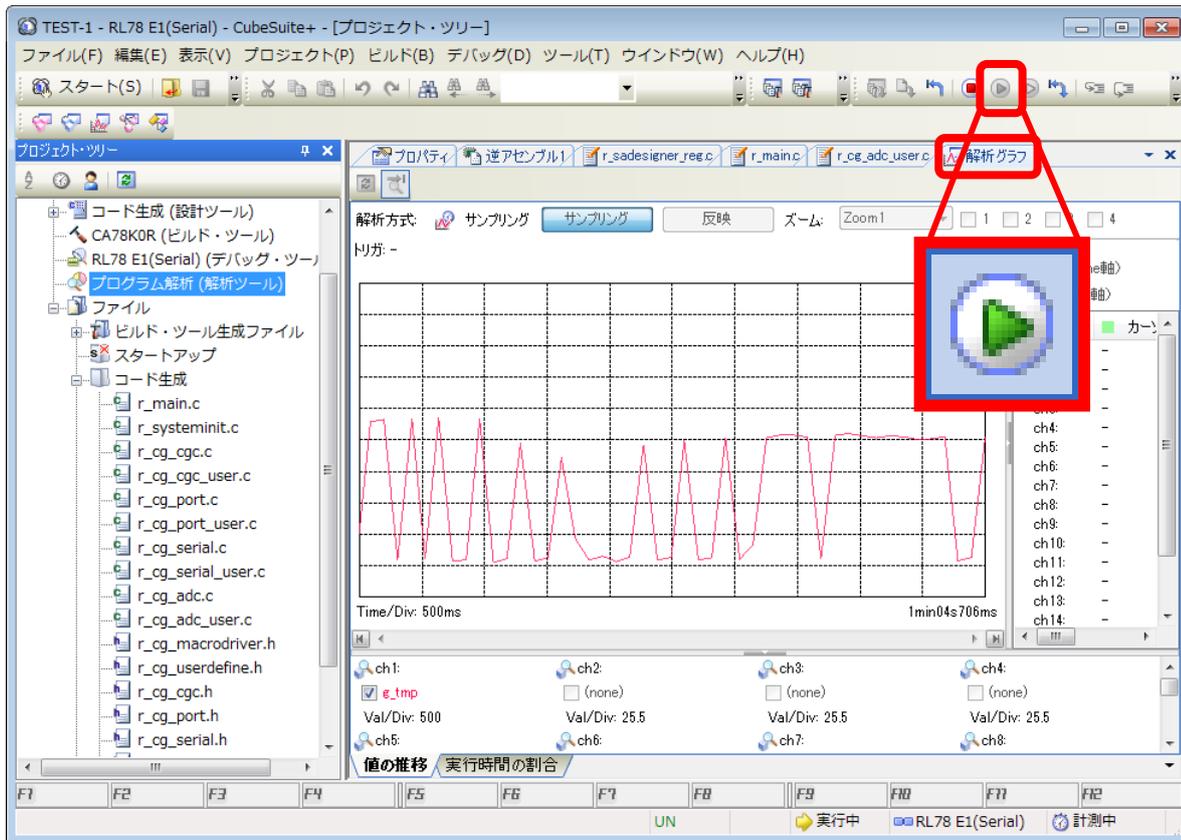
- ① 面左上のアイコンをクリックします。
- ② ラフウィンドウが表示されますので、その上の「反映」ボタンをクリックします。
- ③ ラフ下画面に「g_tmp」が追加されたことを確認します。

次にプロジェクトツリーの「プログラム解析 (解析ツール)」をダブルクリックします。



- 「値の推移」タブの「全般」 → 「自動調整」 → 「行わない」
 → 「1グリッドあたりの時間[Time/Div]」 → 「500ms」
 → 「チャンネル 1」 → 「1グリッドあたりの値[Val/Div]1」 → 500
 → 「オフセット 1」 → -2000

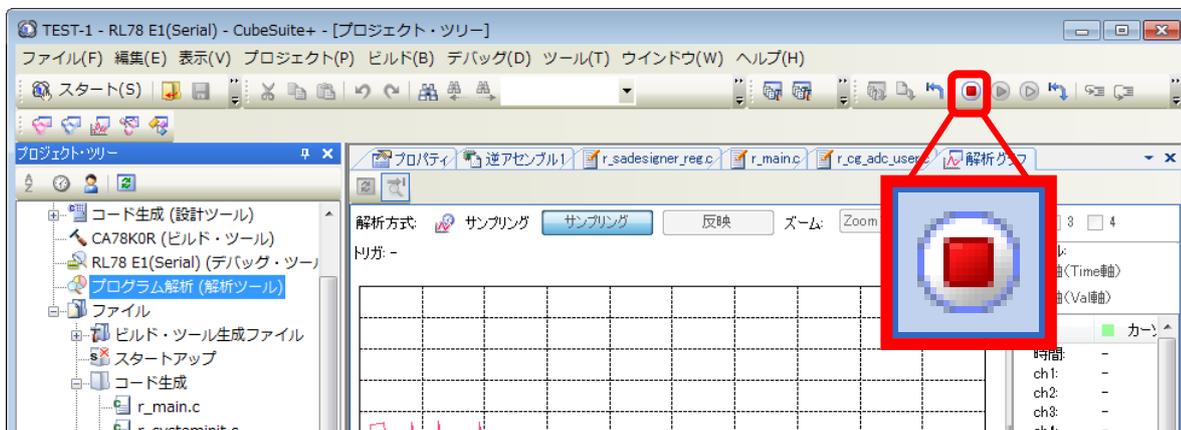
ここまでの設定が終わったら、下記赤枠のアイコンをクリックしてプログラムを再度動作させます。



「解析グラフ」のタブでセンサの動きが波形で表示されることを確認してください。
(先ほどと同様にセンサを手で覆ったり、蛍光灯などの光源に近づけたりしてみてください。)
以上でプログラムは完成です。

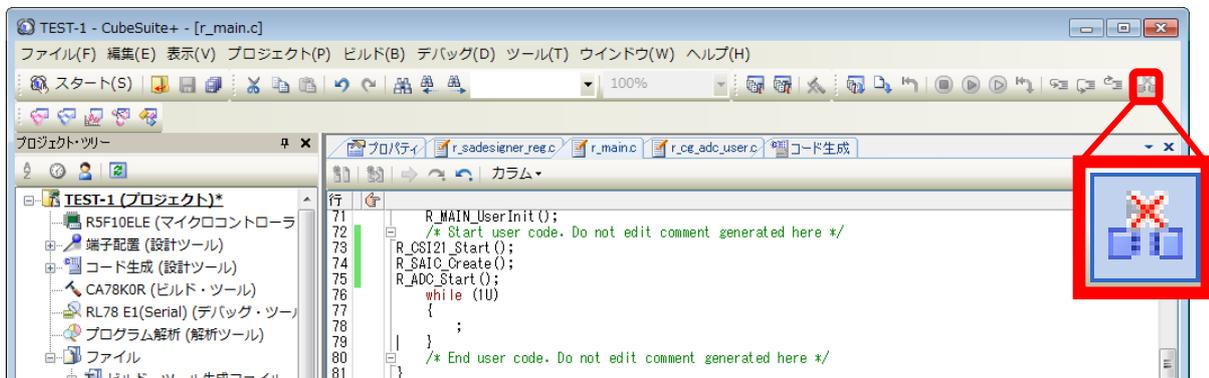
プログラムの動作を停止させます。

下記の赤枠で囲ったアイコンをクリックしてください。

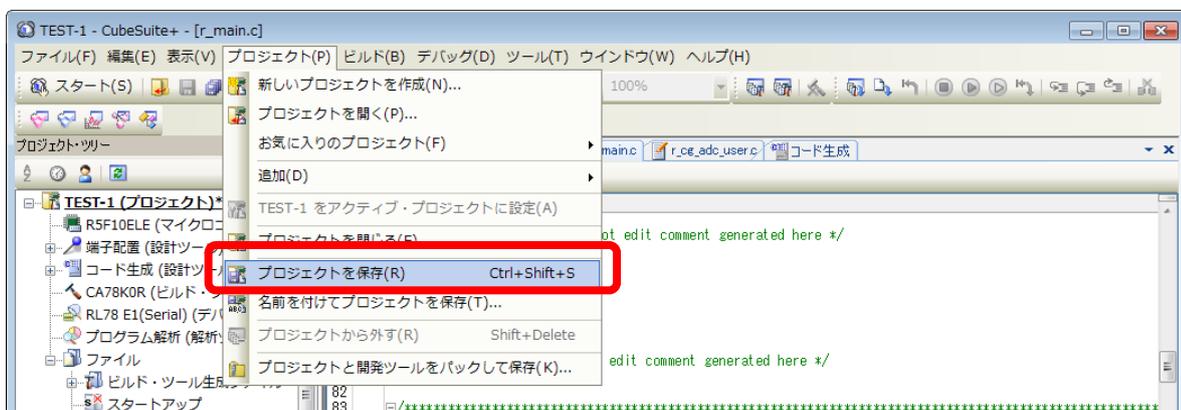


デバッグ・ツールを停止させて、デバッグを終了します。

下記の赤枠のアイコンをクリックしてください。

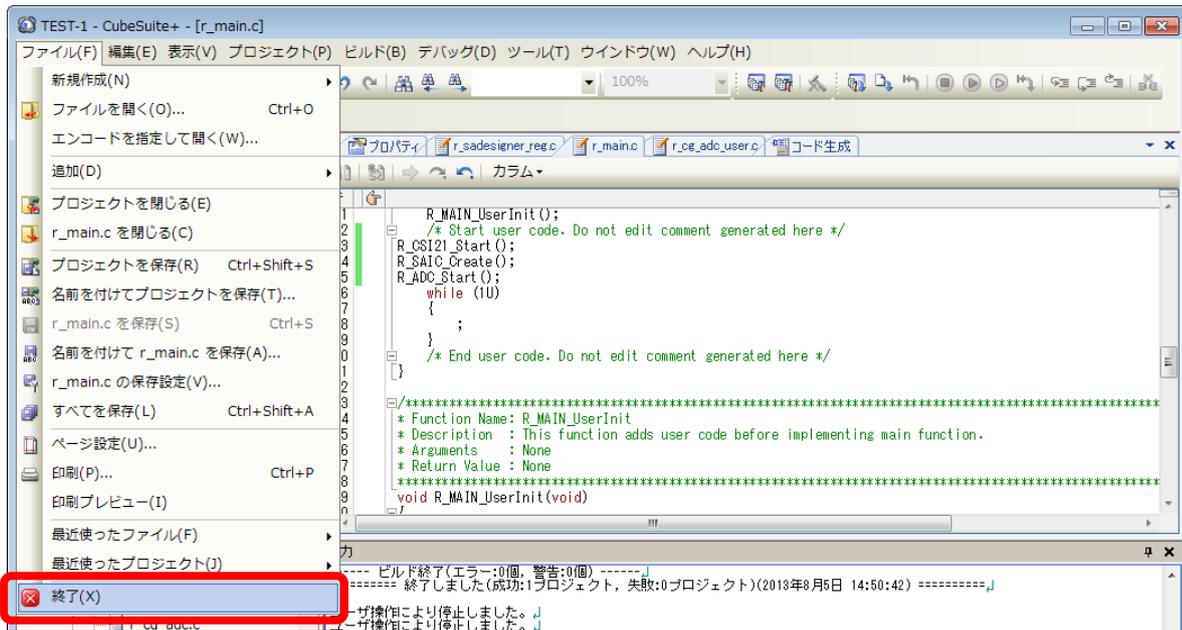


プロジェクトを保存して、CubeSuite+を終了します。

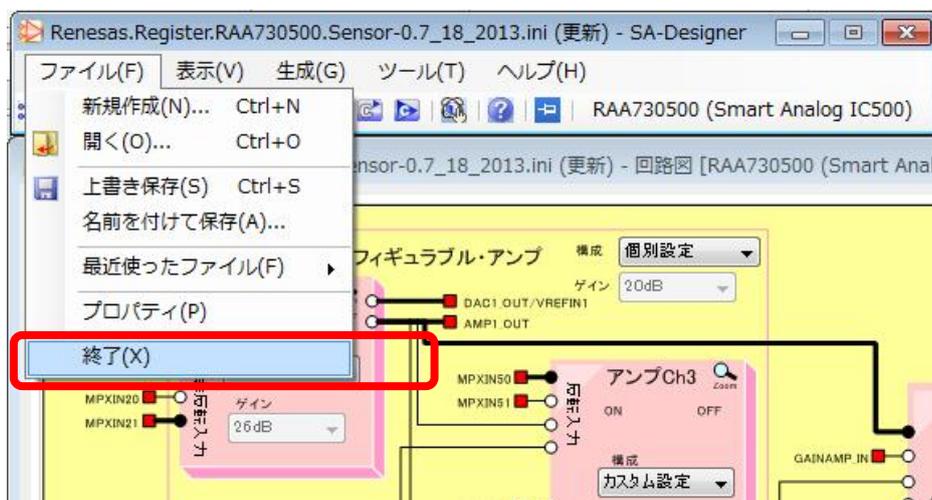


<参考> 次回 CubeSuite+を起動して本プロジェクトを呼び出せば、上記で設定した各内容が自動で呼び出されます。

「ファイル」タブから「終了」を選択して、CubeSuite+を終了させてください。



続いて SA-Designer を終了します。



「ファイル」タブから「終了」を選択して、SA-Designer も終了させてください。

今回使用した Renesas VA の AFE レジスタファイルの保存を聞かれますので、必要に応じて保存してください。

この後今回作成したプログラムをマイコンに書き込みますので、評価ボード、E1、PC の接続はそのままにしてください。

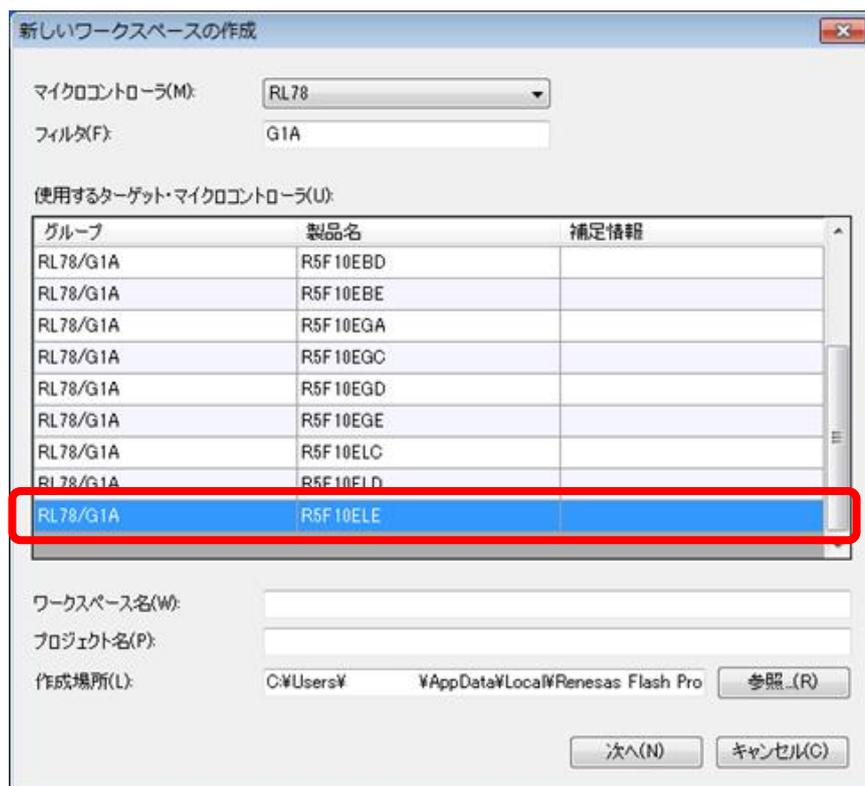
7.17 プログラムをマイコンに書込む

続いて完成したプログラムを RL78/G1A へ書き込んでみましょう。まずは書き込みツールを起動させます。Windows のスタートメニューから書き込みツールを起動します。

[Renesas Electronics Utilities] → [書き込みツール] → [Renesas Flash Programmer]



「新しいワークスペースの作成」「Basic モード」を選択し次へボタンをクリックしてください。



新しいワークスペースの作成画面でグループ:「RL78/G1A」製品名:「R5F10ELE」を選択し、任意の「ワークスペース名」を入力して、「次へ」ボタンをクリックしてください。

(プロジェクト名はワークスペース名と同じ名前が自動入力されます。)



使用ツールが「E1」になっていることを確認して「次へ」をクリックしてください。



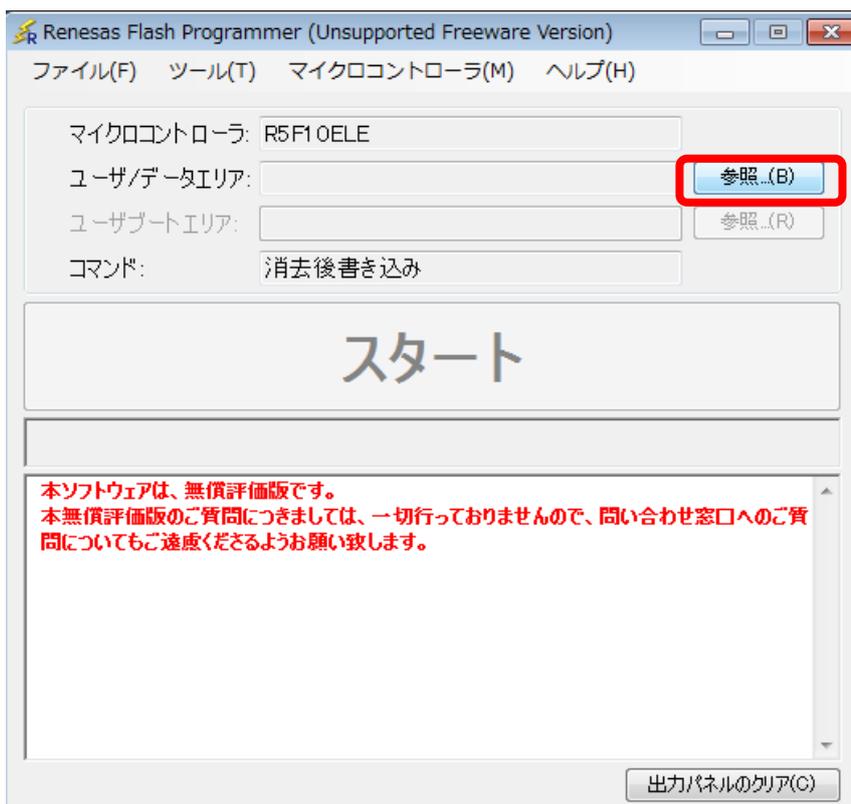
通信速度はデフォルトで選択された速度をそのままお使いください。「次へ」をクリックしてください。



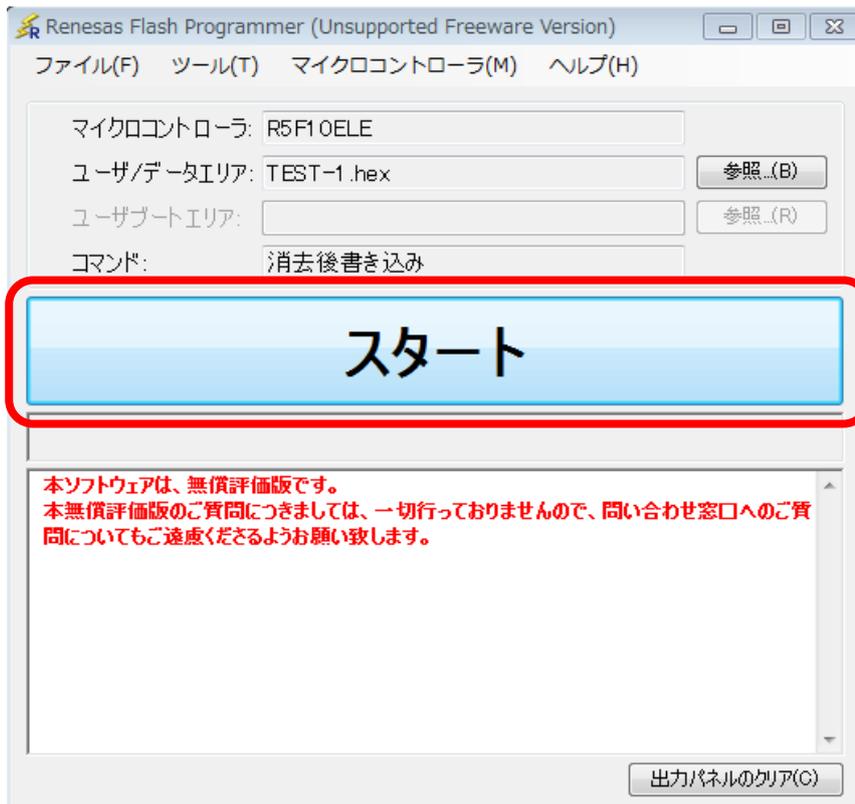
何も選択せず、「次へ」をクリックしてください。



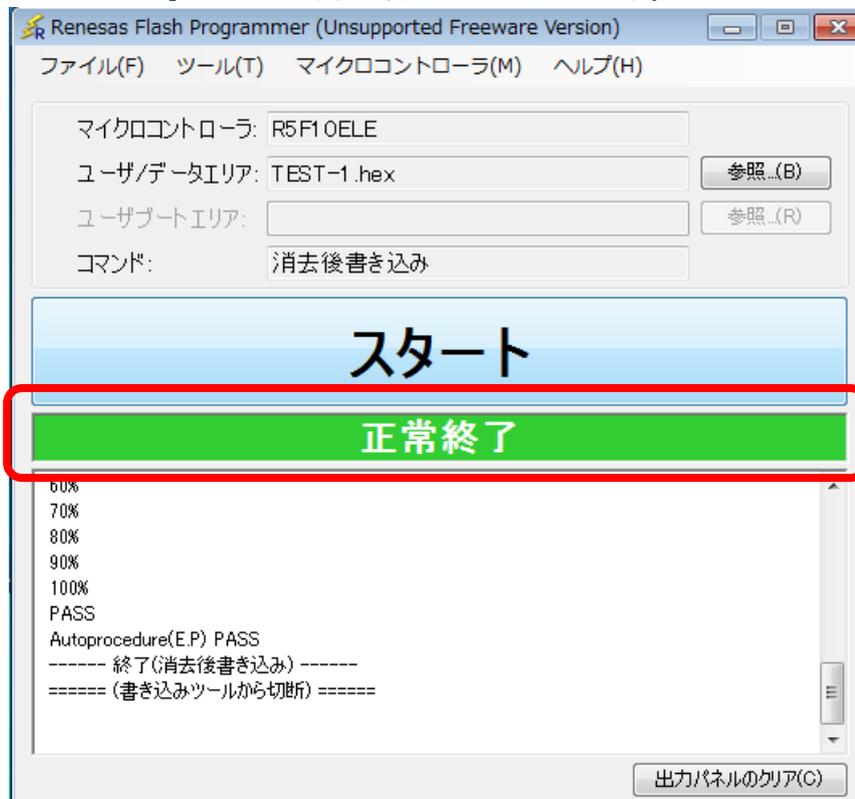
「完了」をクリックしてください。



「参照」ボタンをクリックして、今回作成した CubeSuite+ のプロジェクトファイルの中にある「ファイル名.hex」を指定してください。



「スタート」ボタンを押すと書き込みが始まります。



「正常終了」が表示されたら、書き込みは成功です。



「ファイル」タブから「終了」を選択して終了させてください。

以上でプログラムの書込みは完了です。

PC → E1 → 評価ボード(TSA-IC 500)の各配線を外してください。

PC → 評価ボード(TSA-IC 500)の USB ケーブルは評価ボードへの電源供給に使用しますのでそのまま外さないでください。

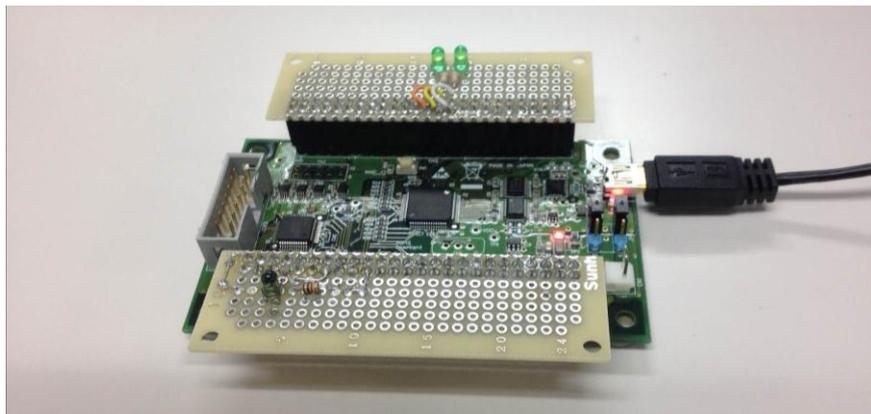
以上で SA-Designer～CubeSuite+でのプログラム作成は完了です。

最後に動作を確認します。

8. 実機で動作確認してみよう！

プログラムが無事完成したら、実機で動作確認してみましょう。

評価ボード(TSA-IC 500)にある RESET(SW2)スイッチを押して、評価ボード上の IC を一度リセットしてください。



実物のイメージ

センサ(手前側)を手で翳したりして LED の点灯が明るさに応じて変わるのを確認してください。

以上で Smart Analog の開発は完了です。

余力があれば、以下を試してみてください。

- Renesas VA でセンサのゲインを変更して感度を調整してみる。
- LED の個数を増やしてみる。
- LED の点灯する方法も変更してみる。

< うまく動かない方へ >

うまく動かない方は以下を確認してみてください。

- 7の内容をもう一度確認して設定、プログラムの内容が間違っていないか？
- 評価ボードとセンサ基板、LED 基板の接続がずれていないか？

9. 最後に

この資料では、Smart Analog を初めてお使い頂く皆様を対象に、センサのシミュレーションからマイコンのプログラム開発、実機動作確認までを解説してきました。

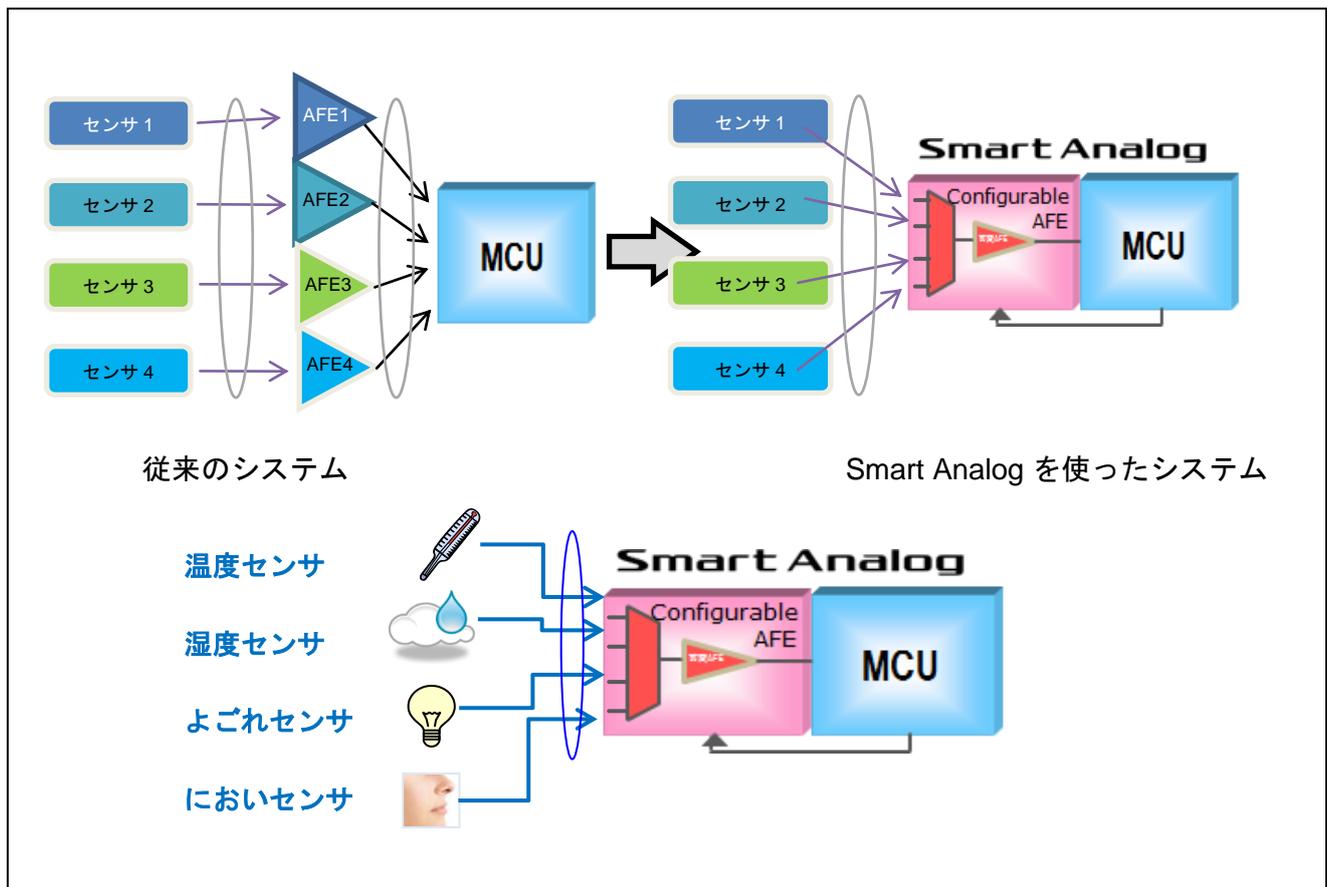
この資料で取得して頂いた知識を元に、今後は別のセンサの制御や Smart Analog 内蔵の機能(フィルタや同期検波など)の利用、などを是非チャレンジしてください。

なお、Smart Analog には他にも便利な機能を沢山搭載していますので、代表例を3つご紹介します。

9.1 マルチセンサ制御

Smart Analog には入力段に差動対で 6ch のマルチプレクサを搭載しています。

これと内蔵 AMP のリコンフィギュラブル機能(反転・非反転・差動・I/V 等を動的に変更できる)を利用して制御方法の異なる複数のセンサを同時に制御する「マルチセンサ制御」が実現できます。

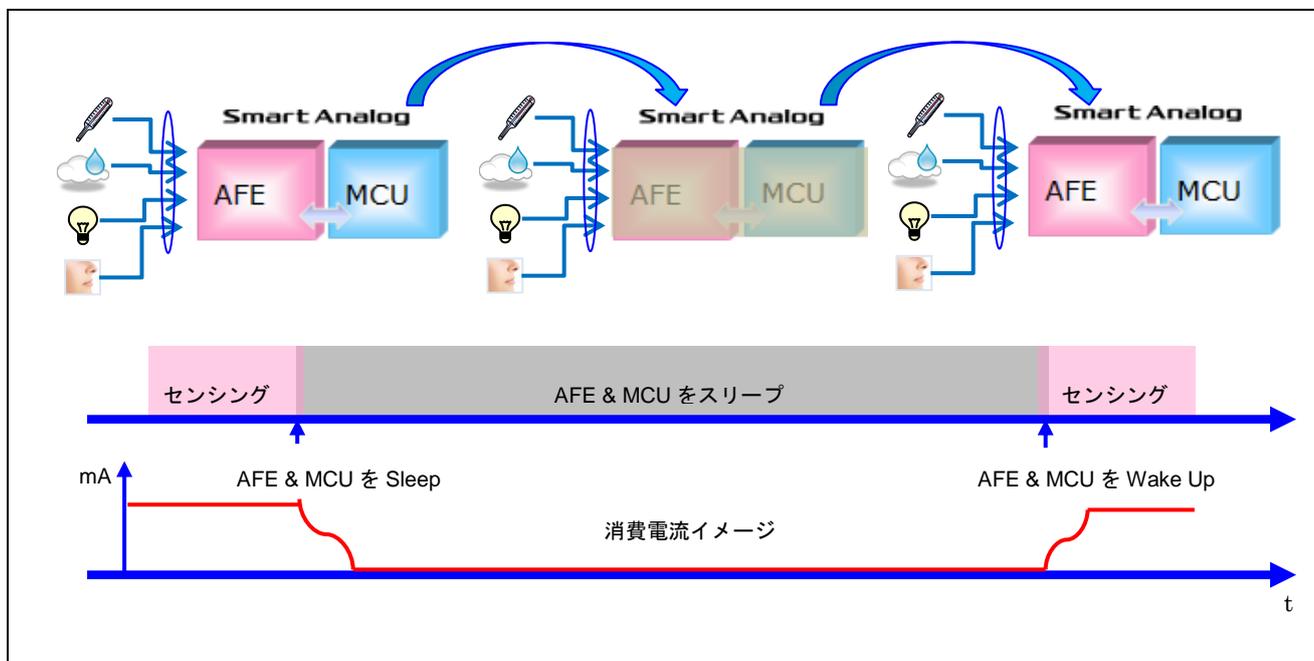


Smart Analog を使ったマルチセンサ制御の例

9.2 間欠動作による低消費電流動作

Smart Analog には内蔵する個々のアナログ回路ブロックを個別に ON/OFF させることができます。

この機能を利用して、センシングしていない時はアナログ回路ブロックを OFF、センシングするときに ON というように制御すれば消費電流を削減することができます。

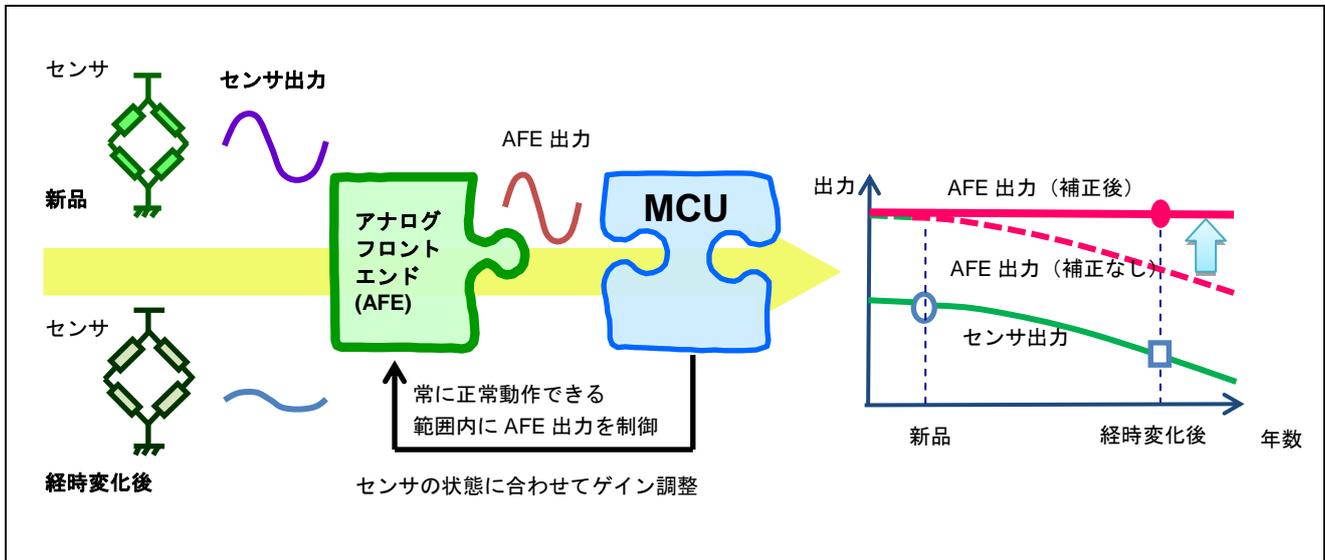


Smart Analog を使った低消費電流動作の例

9.3 自動補正制御

Smart Analog は内蔵アナログ回路のゲインやオフセットなどのパラメータを動的に変更することが出来ます。

この機能を利用して、汚れや経年によるセンサの劣化を自動で補正することができます。



Smart Analog を使った自動補正制御の例

これら以外にも様々な付加機能の実現が可能です。

是非みなさんのアイデアを Smart Analog で試してみてください。

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

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
Rev.0.01	2013.04.18	---	初版発行
Rev.1.00	2013.07.19	1	動作確認デバイスの説明を追加
		2	概説、開発環境の説明を追加
		—	Renesas VA Version 2.0.0.0 のリビジョンアップによるドキュメントの改訂
		—	SA-Designer V1.02.00 のリビジョンアップによるドキュメントの改訂
		—	CubeSuite+ V2.00.00 のリビジョンアップによるドキュメントの改訂
		38	CubeSuite+起動時のプラグインの設定の説明を追加
		—	誤記訂正

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットにかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

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、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じて、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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、当社製品は耐放射線設計については行っていません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、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。お客様が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. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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。

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とその総株主の議決権の過半数を直接または間接に保有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。



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販売株式会社 〒100-0004 千代田区大手町 2-6-2（日本ビル）

■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：<http://japan.renesas.com/contact/>