RL78/I1Eコード生成学習

2016/6/1　　Rev. 1.00

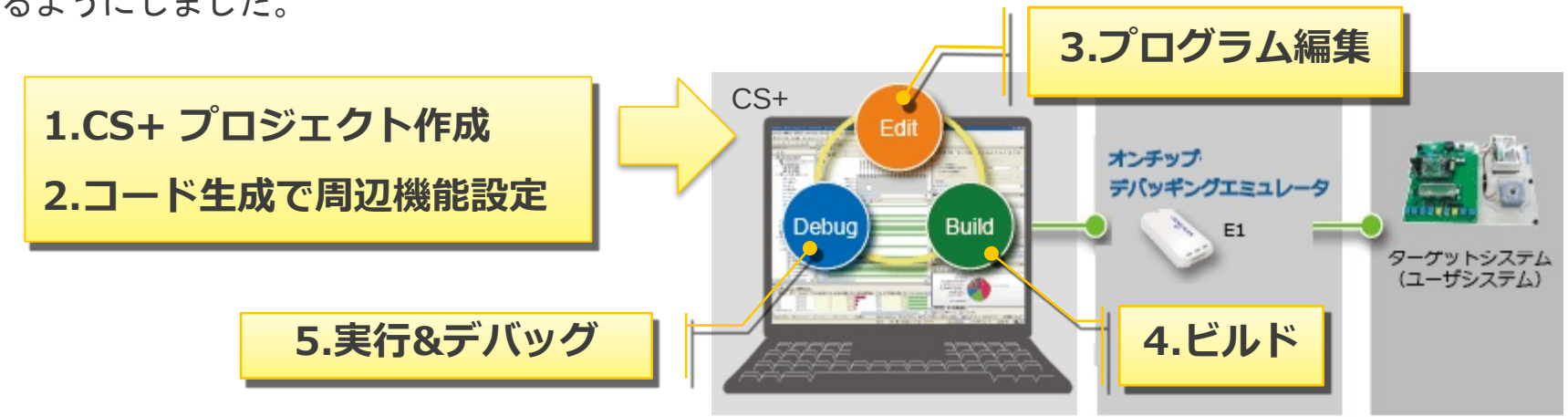
ソフトウエア事業部 ソフトウエア技術部

ルネサスシステムデザイン株式会社

R20UT3809JJ0100

はじめに

この資料はルネサスのマイコンを初めて使うお客様を対象としております。ルネサスの統合開発環境 CS+ for CC とその統合開発環境に含まれるコード生成を使って**プロジェクトの作成～デバッグ**までを解説します。プロジェクトを演習A,B,Cと３つに分けて難易別に学習できるようにしました。



使用ツールとバージョン

・マイコン　　RL78/I1E (R5F11CCC)
https://www.renesas.com/ja-jp/products/microcontrollers-microprocessors/rl78/rl78i1x/rl78i1e.html
ここでは、**FB-R5F11CCC-TB** (株式会社内藤電誠町田製作所製CPUボード)を使用します。

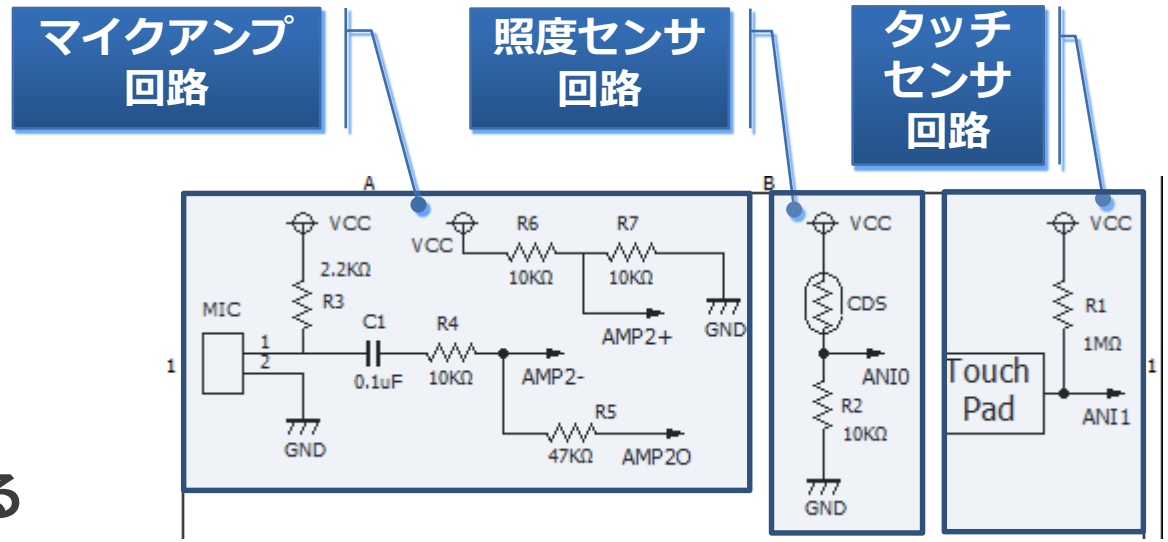
・デバッガ　　E1エミュレータ　　http://japan.renesas.com/e1
・統合開発環境(IDE)　CS+ for CC V4.00　　http://japan.renesas.com/cs+
　　ビルドツール　CC-RL V1.02　　http://japan.renesas.com/rl78_c

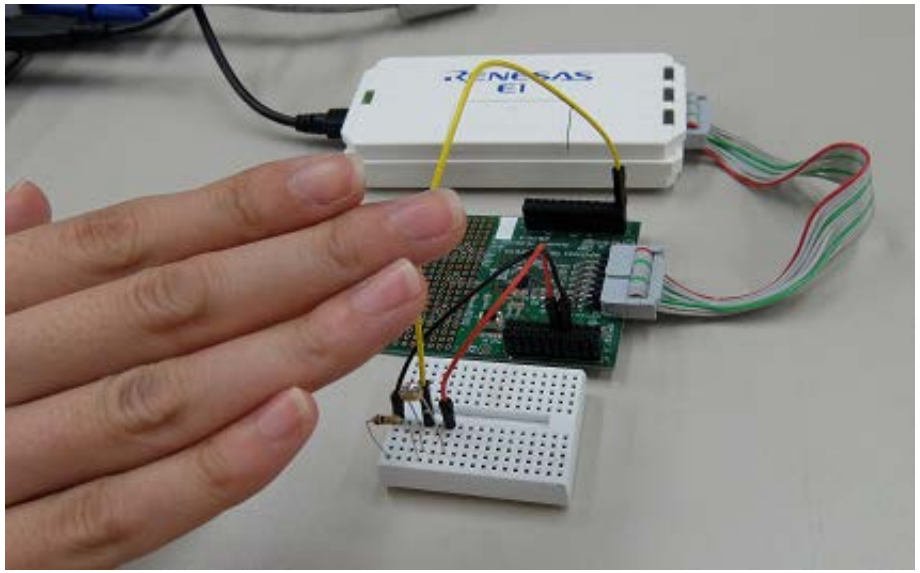
演習内容

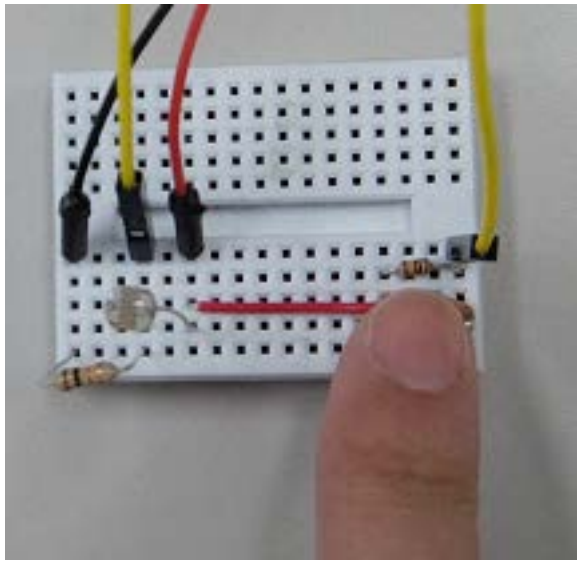
A. 照度センサを使って明るさを検知

B. 簡易タッチセンサでLEDをON/OFF

C. マイクを使って音でLEDの輝度を変える







▪演習A. 照度センサを使って明るさを検知
▪演習B. 簡易タッチセンサでLEDをON/OFF
▪演習C. マイクを使って音でLEDの輝度を変える

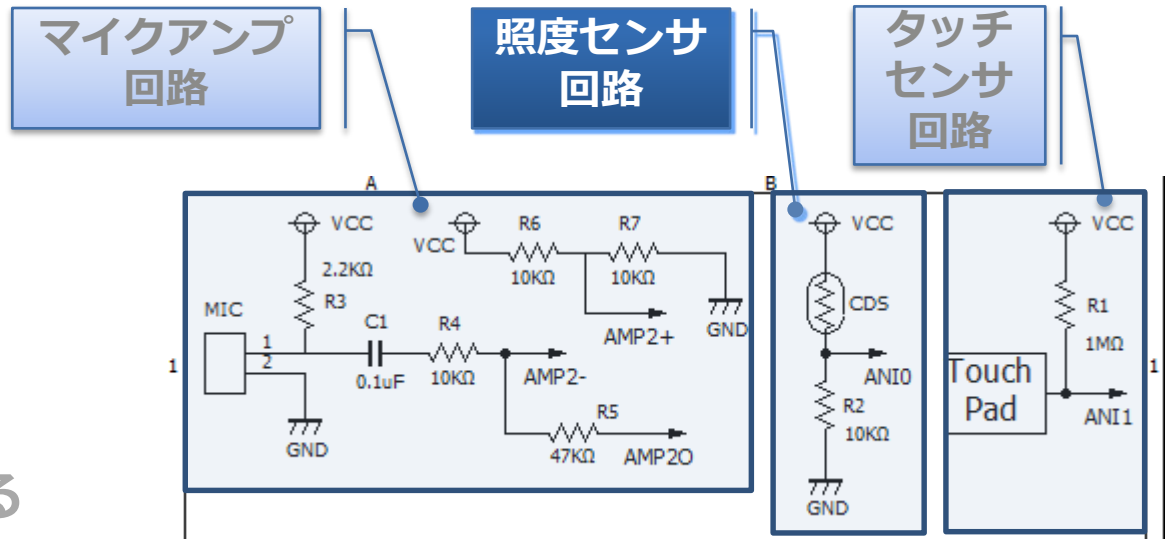
目次

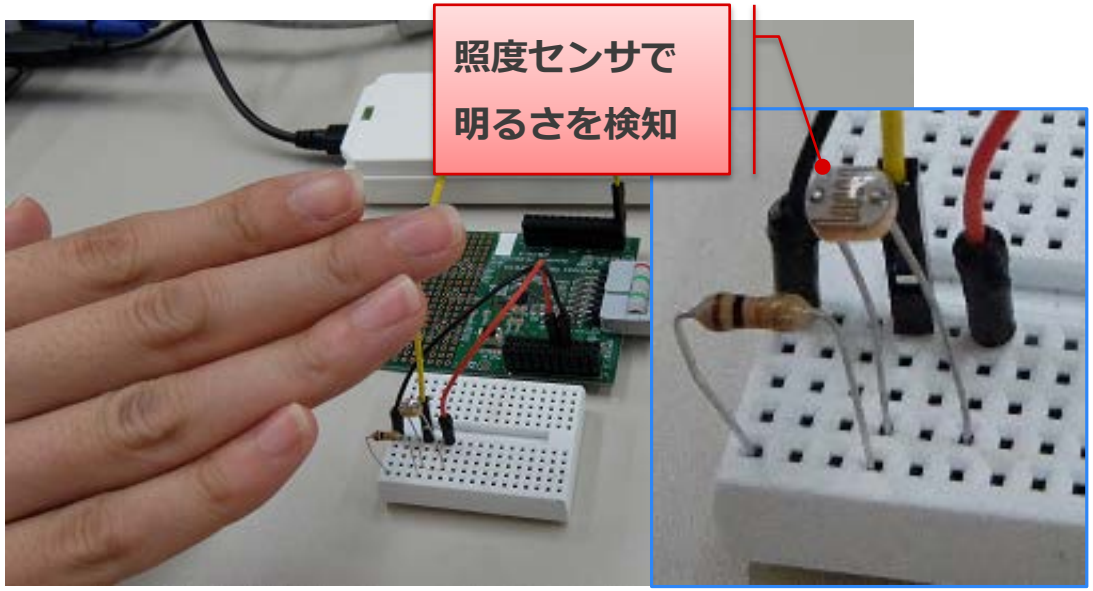
演習A

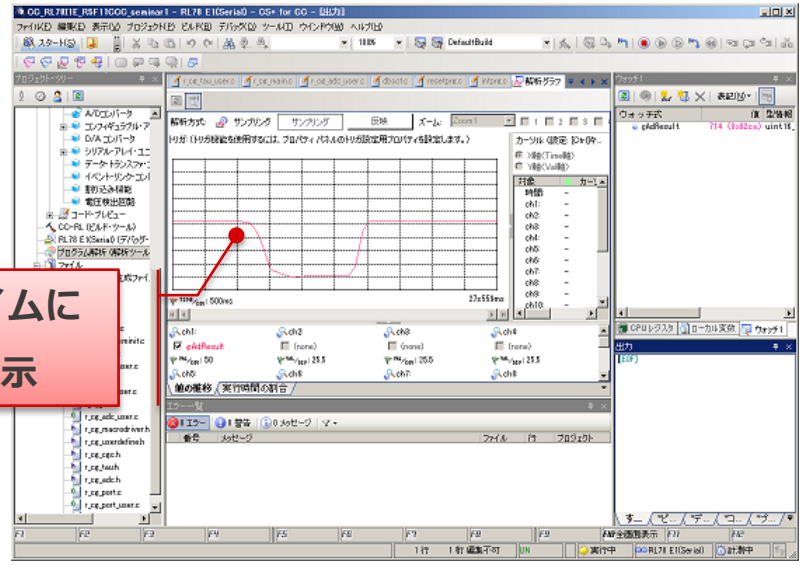
A. 照度センサを使って明るさを検知

B. 簡易タッチセンサでLEDをON/OFF

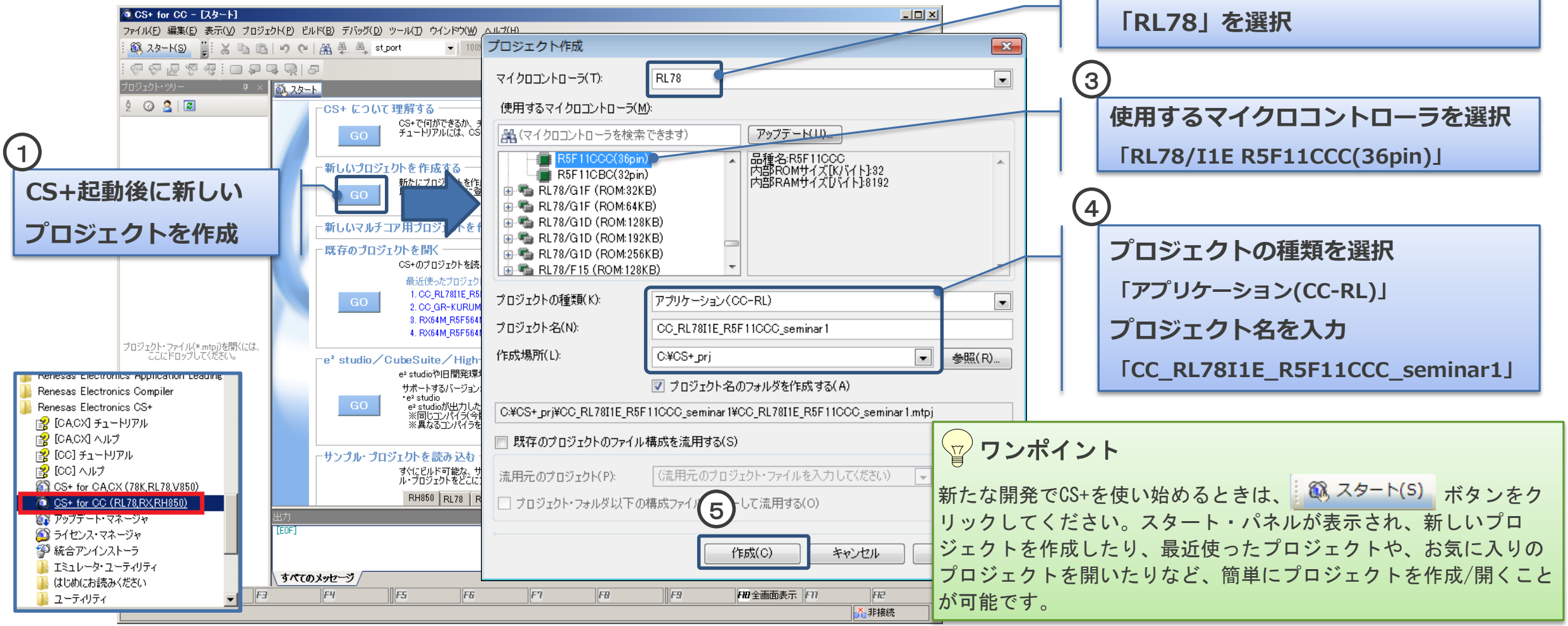
C. マイクを使って音でLEDの輝度を変える







プロジェクト作成



① CS+起動後に新しいプロジェクトを作成

② 「RL78」を選択

③ 使用するマイクロコントローラを選択
「RL78/I1E R5F11CCC(36pin)」

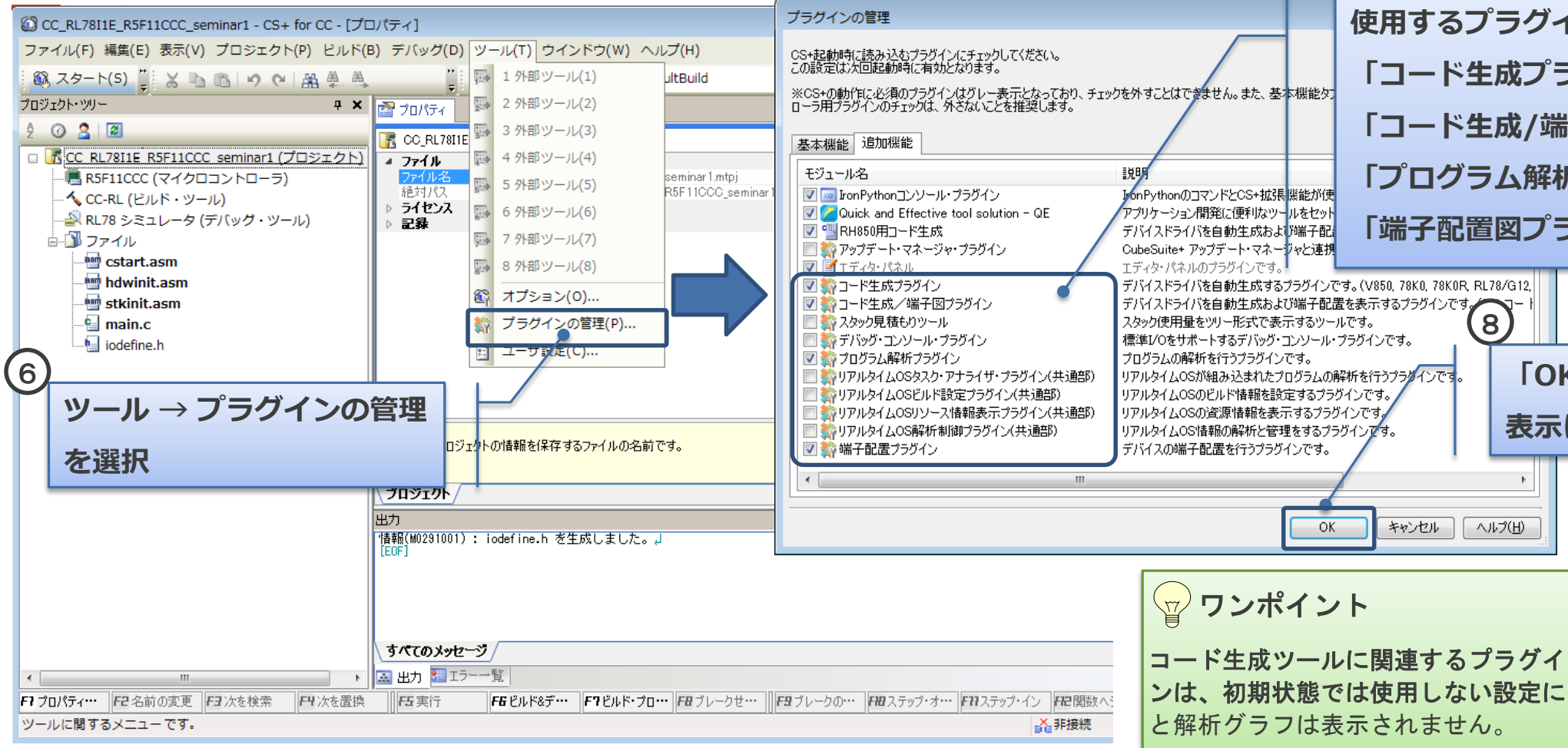
④ プロジェクトの種類を選択
「アプリケーション(CC-RL)」
プロジェクト名を入力
「CC_RL78I1E_R5F11CCC_seminar1」

⑤

ワンポイント

新たな開発でCS+を使い始めるときは、スタート(S) ボタンをクリックしてください。スタート・パネルが表示され、新しいプロジェクトを作成したり、最近使ったプロジェクトや、お気に入りのプロジェクトを開いたりなど、簡単にプロジェクトを作成/開くことが可能です。

プロジェクト作成(プラグインの管理)



⑥ ツール → プラグインの管理を選択

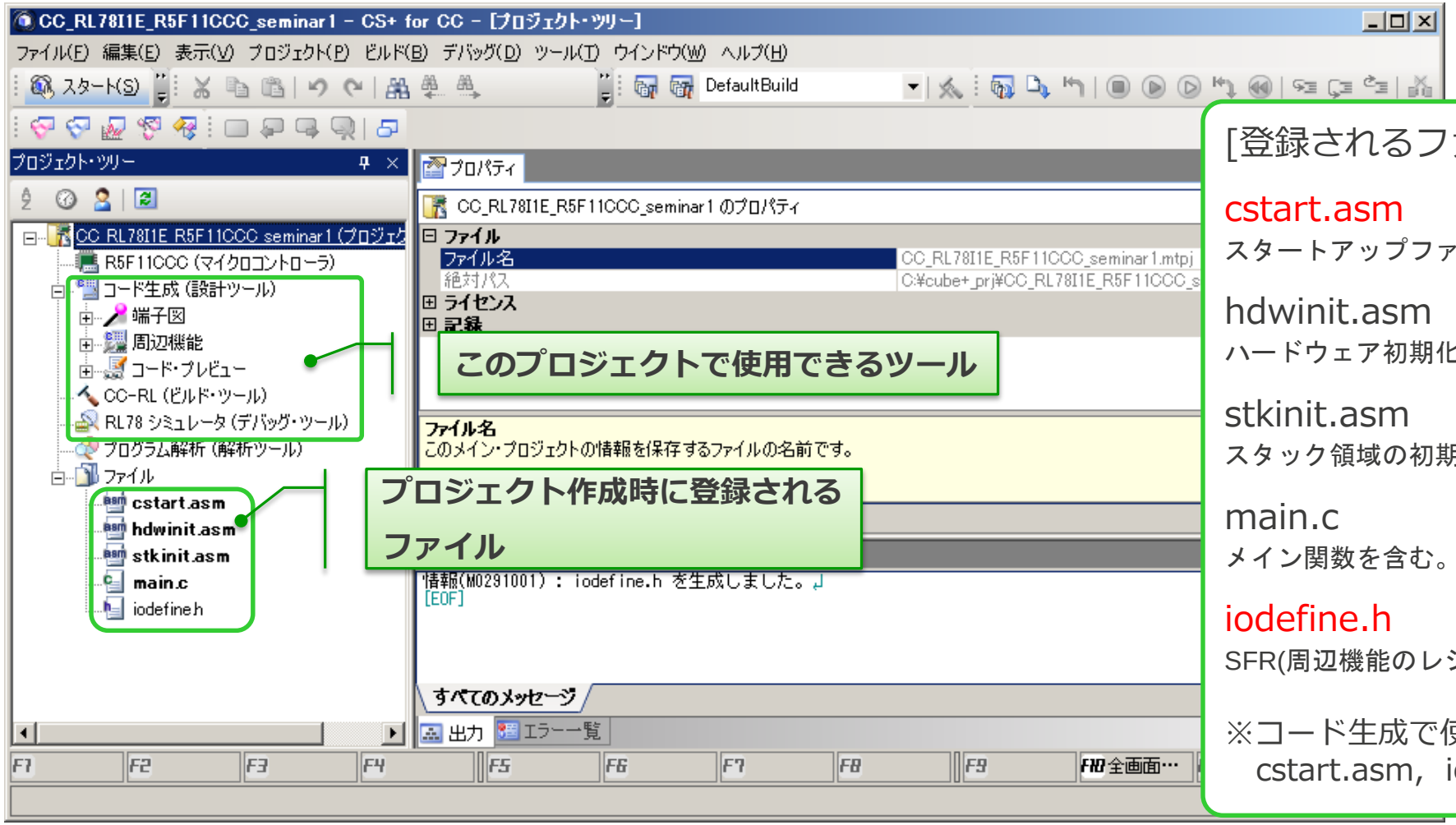
⑦ 使用するプラグインにチェックを入れる
「コード生成プラグイン」
「コード生成/端子図プラグイン」
「プログラム解析プラグイン」
「端子配置図プラグイン」

⑧ 「OK」を選択、
表示に従い製品を再起動する

ワンポイント

コード生成ツールに関連するプラグイン、プログラム解析プラグインは、初期状態では使用しない設定になっているため、コード生成と解析グラフは表示されません。

プロジェクト作成 (プロジェクト作成時に登録されるファイル)



[登録されるファイル一覧]

cstart.asm
スタートアップファイル。リセット後に最初に実行される。

hdwinit.asm
ハードウェア初期化処理を行う。

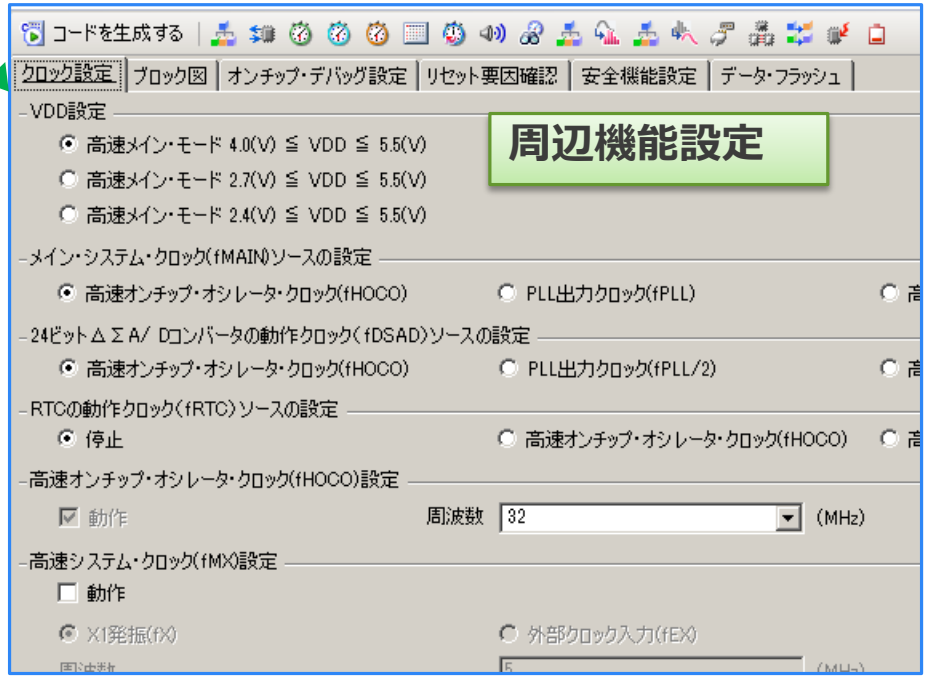
stkinit.asm
スタック領域の初期化を行う。

main.c
メイン関数を含む。

iodefine.h
SFR(周辺機能のレジスタ)を定義したファイル

※コード生成で使うファイルは以下
cstart.asm, iodefine.h

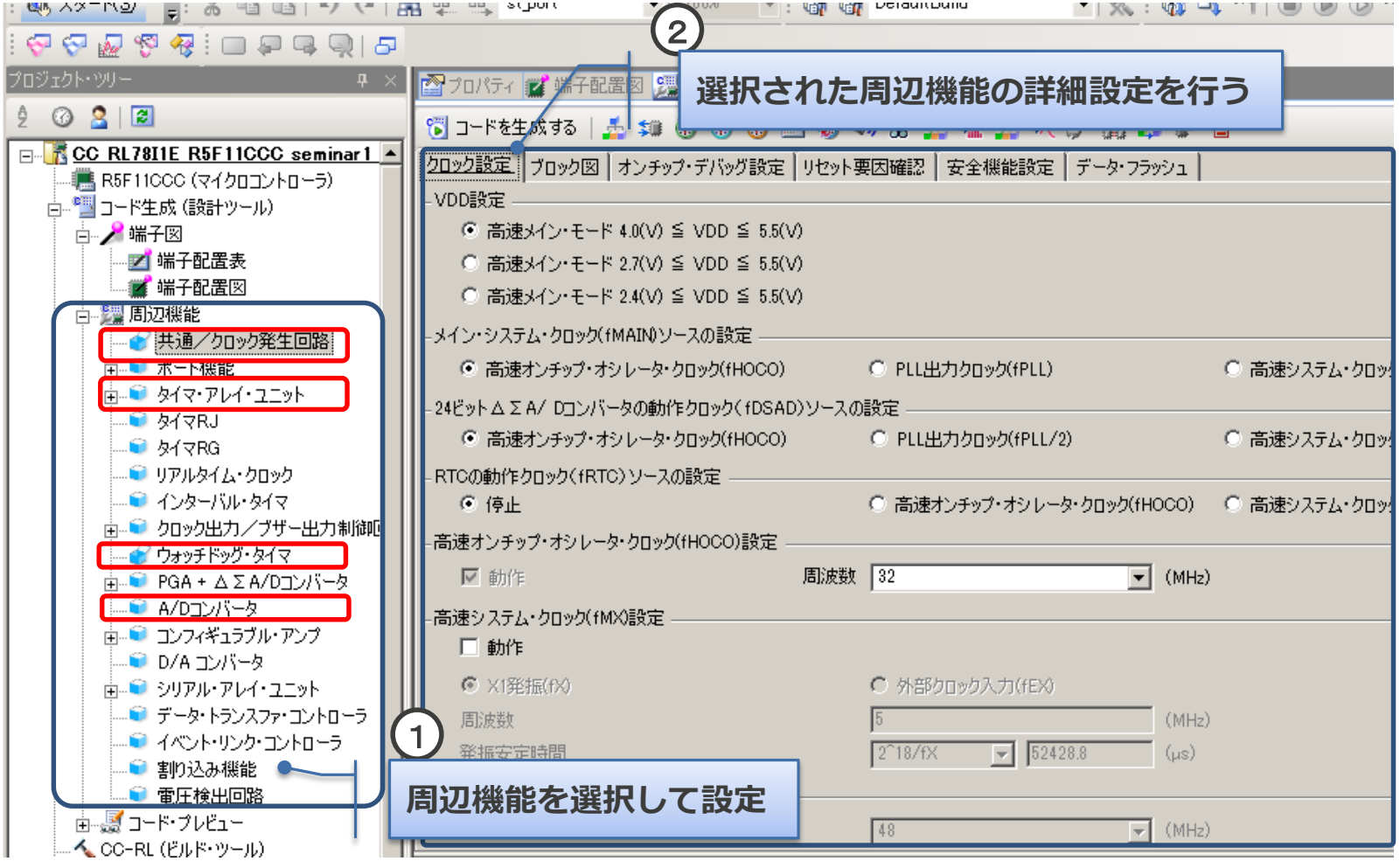
プロジェクト作成 (コード生成の機能について)



ワンポイント

コード生成は、周辺機能のドライバをCソース出力するツールです。
USB、CANなどミドルウェアと組み合わせて使用する周辺機能には、対応していません。

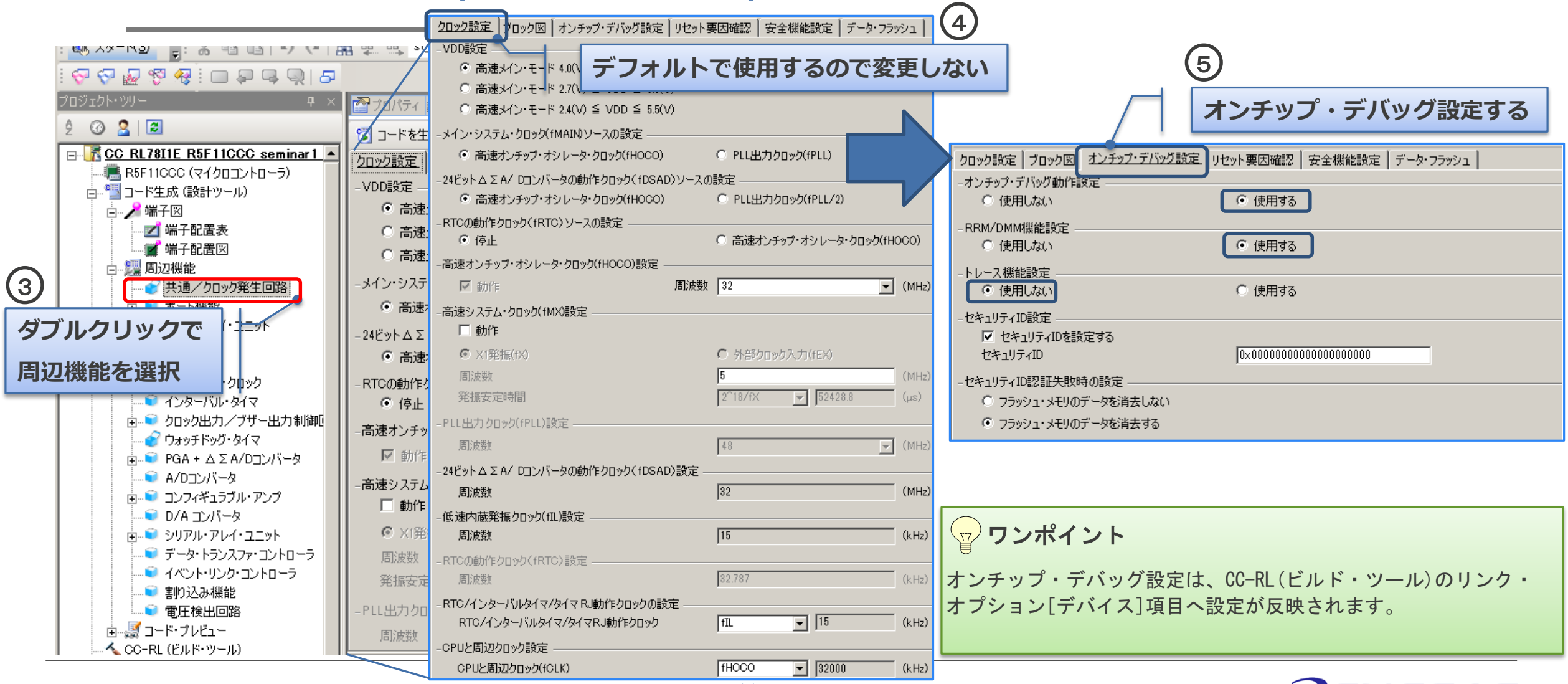
周辺機能設定 (コード生成での設定について)



このデモで設定する周辺機能

- **共通/クロック発生回路(デフォルト設定で使用)**
  → メイン・システム・クロック、高速オンチップ・オシレータ・クロック(fHOCO)
  高速オンチップ・オシレータ(fHOCO) 32MHz
  CPUと周辺クロック fHOCOを使用
- **タイマ・アレイ・ユニット**
  → 1ミリ秒のインターバルタイマを設定
- **ウォッチドッグタイマ**
  → 未使用にする
- **A/Dコンバータ**
  → A0端子のA/D変換を行う

周辺機能設定 (共通/クロック発生回路)

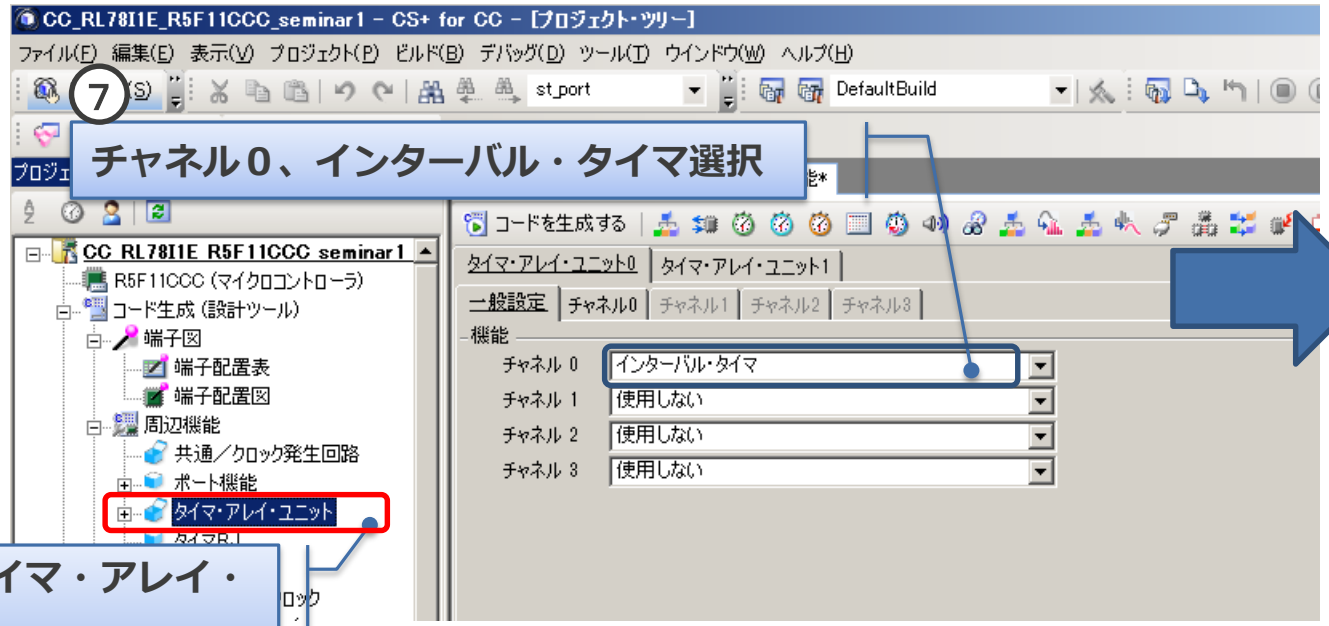


ワンポイント

オンチップ・デバッグ設定は、CC-RL(ビルド・ツール)のリンク・オプション[デバイス]項目へ設定が反映されます。

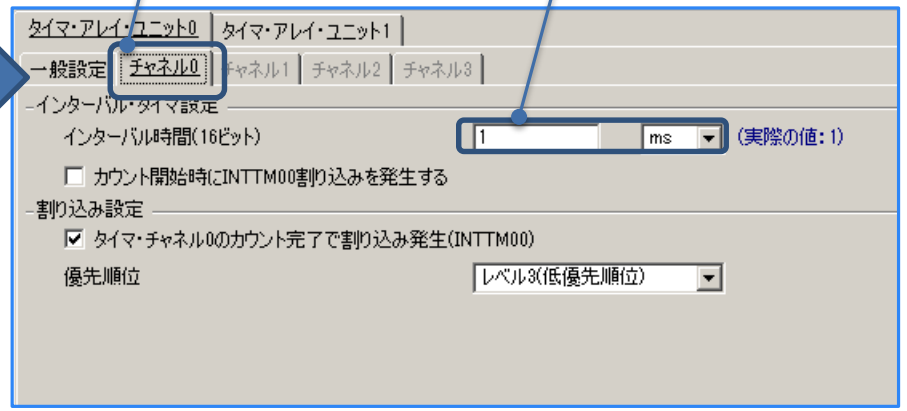
A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

周辺機能設定 (タイマ・アレイ・ユニット)



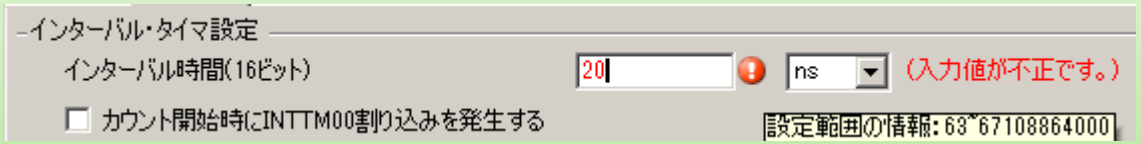
⑧ チャネル0タブを選択

⑨ インターバル時間に1msを設定

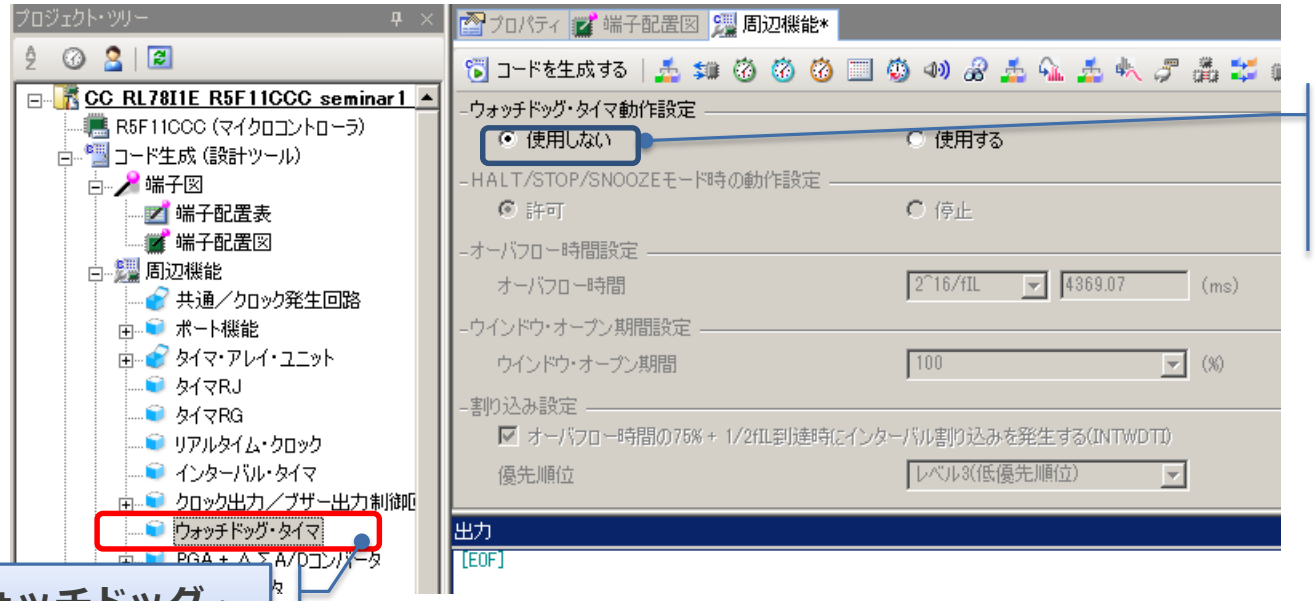


ワンポイント

数値を入力する箇所では、マイコンの設定範囲を超えた場合、エラーを表示します。
例えば、インターバル時間に20nsと入力すると設定範囲外となり、エラーを表示します。



周辺機能設定 (ウォッチドッグ・タイマ)



⑩ ウォッチドッグ・タイマを選択

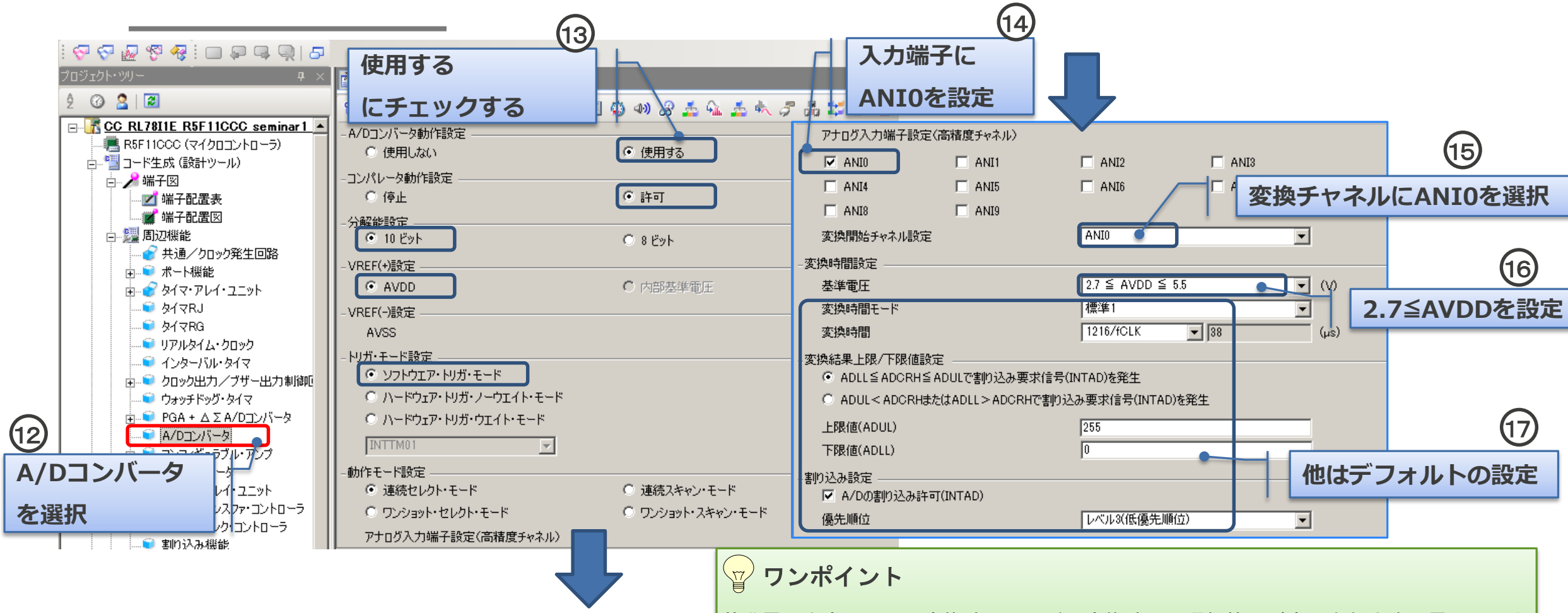
⑪ 使用しない にチェックする

ワンポイント

プロジェクト・ツリー・パネルの各周辺機能のアイコンは、以下を意味しています。

：使用している機能、 ：未使用の機能、 ：設定内容に問題が発生、 ：設定内容の更新が必要

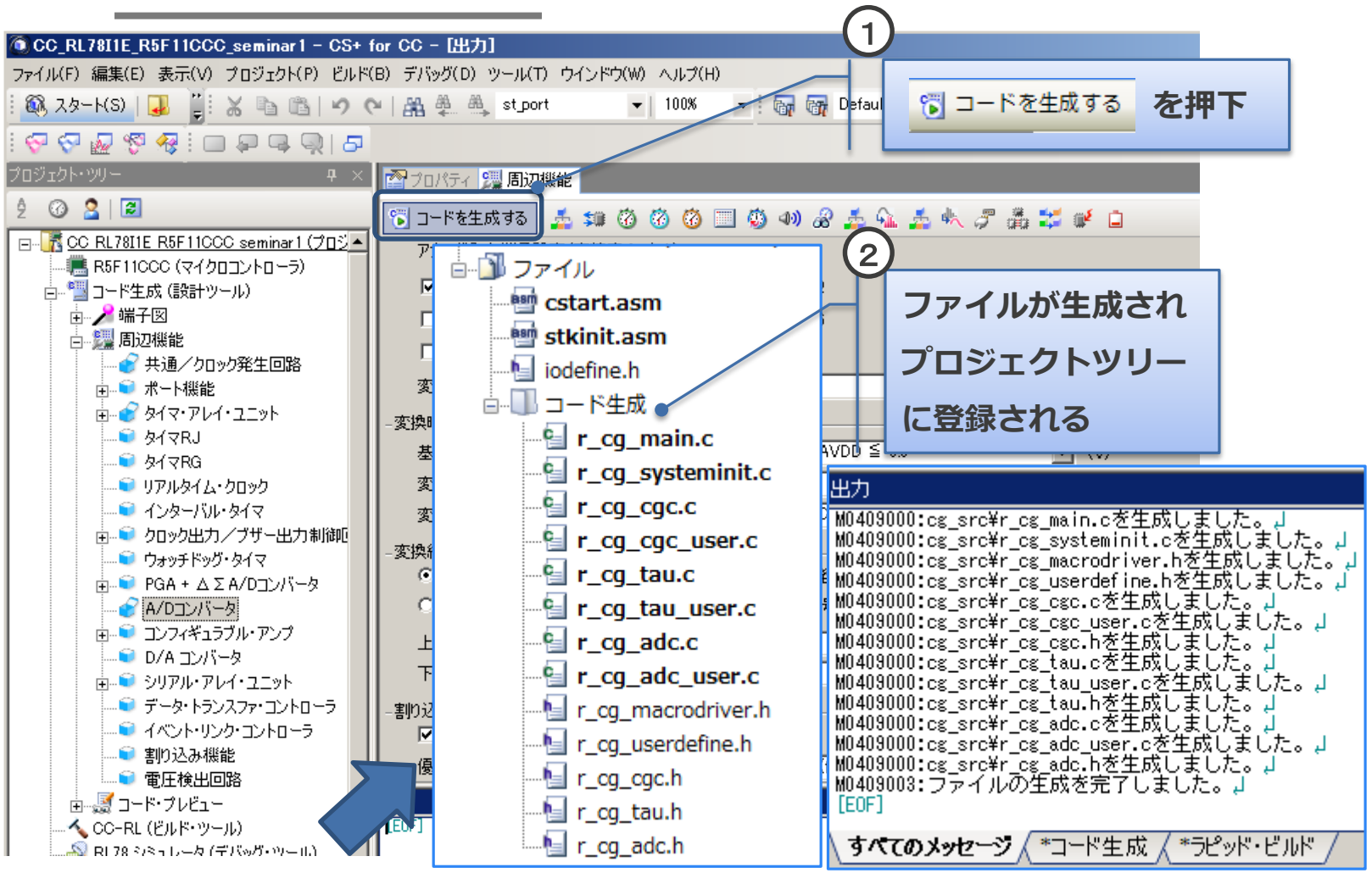
周辺機能設定 (A/Dコンバータ)



ワンポイント

基準電圧を変えると、変換時間モードと変換時間の選択範囲が変更されます。電圧の入力が必要なのは、設定時間に影響するためです。

コード生成とプログラム編集



生成されるファイル一覧

必ず生成されるファイル

- r_cg_main.c
- r_cg_systeminit.c
- r_cg_macrodriver.h
- r_cg_userdefine.h
- r_cg_cgc.c /h
- r_cg_cgc_user.c

使用する周辺機能に応じて生成されるファイル

以下は、タイマ・アレイ・ユニット、A/Dコンバータを設定すると出力されるファイル

- r_cg_tau.c /h
- r_cg_tau_user.c
- r_cg_adc.c/h
- r_cg_adc_user.c

コード生成とプログラム編集 (生成されるファイルの説明)

生成ファイルのルール

r_cg_main.c	メイン関数
r_cg_systeminit.c	hdwinit()を含む周辺機能の初期化
r_cg_macrodriver.h	コード生成で使うマクロ/定数定義
r_cg_userdefine.h	ユーザが使うマクロ/定数定義
r_cg_cgc.c /h	
r_cg_cgc_user.c	周辺機能、クロックに関する設定
r_cg_周辺機能.c / .h	周辺機能の初期化と制御するAPIを含む
r_cg_周辺機能_user.c	周辺機能の割り込み関数を含む。 ユーザーは割り込み関数内に処理を追加する。

ワンポイント

周辺機能を制御するAPIには、周辺機能の開始/停止や、結果の取得、データの送受信などがあります。

例えば r_cg_tau.c には、タイマを開始/停止するAPIが含まれます。

- タイマ開始：R_TAU0_Channel0_Start()
- タイマ停止：R_TAU0_Channel0_Stop()

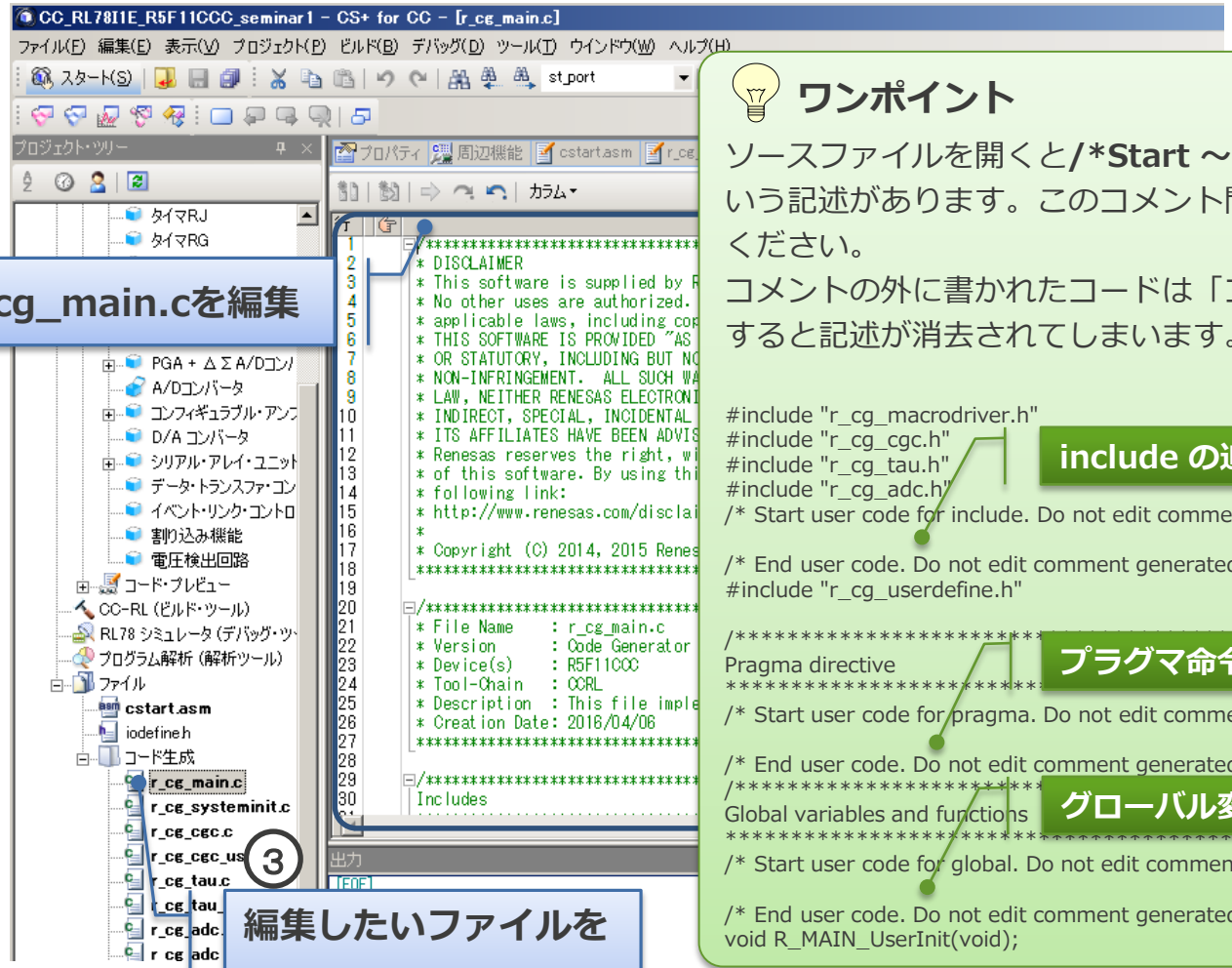
ユーザーが R_TAU0_Channel0_Start() を呼び出すとタイマが開始します。周期割り込みで呼び出される割込みハンドラ内に処理を記述します。

- カウント完了： r_tau0_channel0_interrupt()

このプロジェクトでは1ミリ秒ごとに上記関数が呼ばれます。

コード生成ツールが出力するAPIの詳細は、下記のユーザーズマニュアルをご参照ください。
http://japan.renesas.com/products/tools/coding_tools/coding_assistance/cg_p/Documentation.jsp
CS+ コード生成ツール 統合開発環境 ユーザーズマニュアル RL78 API リファレンス編（R20UT3102JJ0102）

コード生成とプログラム編集 (ソースファイルの編集)



ワンポイント

ソースファイルを開くと/*Start ～*/，/*End～*/という記述があります。このコメント間にコードを書いてください。
コメントの外に書かれたコードは「コード生成」を実行すると記述が消去されてしまいます。

```
#include "r_cg_macrodriver.h"
#include "r_cg_cgc.h"
#include "r_cg_tau.h"
#include "r_cg_adc.h"
/* Start user code for include. Do not edit comment generated here */

/* End user code. Do not edit comment generated here */
#include "r_cg_userdefine.h"

/***********************************************************************
Pragma directive
***********************************************************************/
/* Start user code for pragma. Do not edit comment generated here */

/* End user code. Do not edit comment generated here */
/***********************************************************************
Global variables and functions
***********************************************************************/
/* Start user code for global. Do not edit comment generated here */

/* End user code. Do not edit comment generated here */
void R_MAIN_UserInit(void);
```

include の追加を書く場合

プラグマ命令を書く場合

グローバル変数を書く場合

```
/***********************************************************************
* 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 */
    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 */
}

/* Start user code for adding. Do not edit comment generated here */

/* End user code. Do not edit comment generated here */
```

mainの処理を記述

ユーザの初期化処理を記述

ユーザ関数を新規に追加するときに記述

プログラム追記部分

コード生成とプログラム編集 (r_cg_main.c)

```
46  /***********************************************************
47  Global variables and functions
48  ***********************************************************/
49  /* Start user code for global. Do not edit comment generated here */
50  uint16_t gAdResult;                    // AD変換結果
51  /* End user code. Do not edit comment generated here */
52
53  void R_MAIN_UserInit(void);
54  /***********************************************************
55  * Function Name: main
56  * Description  : This function implements main function.
57  * Arguments    : None
58  * Return Value : None
59  ***********************************************************/
60  void main(void)
61  {
62      R_MAIN_UserInit();
63      /* Start user code. Do not edit comment generated here */
64      while (1U)
65      {
66          ;
67      }
68      /* End user code. Do not edit comment generated here */
69  }
```

```
70  /***********************************************************
71  * Function Name: R_MAIN_UserInit
72  * Description  : This function adds user code before implementing main function.
73  * Arguments    : None
74  * Return Value :None
75  ***********************************************************/
76  void R_MAIN_UserInit(void)
77  {
78      /* Start user code. Do not edit comment generated here */
79      R_TAU0_Channel0_Start();           // タイマチャネル０開始
80      EI();
81      /* End user code. Do not edit comment generated here */
82  }
```

ワンポイント

コード生成で使用する変数のデータ型は、r_cg_macrodriver.hに定義されています。

```
typedef signed char        int8_t;
typedef unsigned char      uint8_t;
typedef signed short       int16_t;
typedef unsigned short     uint16_t;
typedef signed long        int32_t;
typedef unsigned long      uint32_t;
typedef unsigned short     MD_STATUS;
```

プログラム追記部分

コード生成とプログラム編集 (r_cg_tau_user.c)

```
29  /***********************************************
30  Includes
31  ***********************************************/
32  #include "r_cg_macrodriver.h"
33  #include "r_cg_tau.h"
34  /* Start user code for include. Do not edit comment generated here */
35  #include "r_cg_adc.h"
36  /* End user code. Do not edit comment generated here */
37  #include "r_cg_userdefine.h"
38
39  /***********************************************
40  Pragma directive
41  ***********************************************/
42  #pragma interrupt r_tau0_channel0_interrupt(vect=INTTM00)
    .
    .
    .
52  /***********************************************
53  * Function Name: r_tau0_channel0_interrupt
54  * Description  : This function INTTM00 interrupt service routine.
55  * Arguments    : None
56  * Return Value : None
57  ***********************************************/
58  __interrupt static void r_tau0_channel0_interrupt(void)
59  {
60      /* Start user code. Do not edit comment generated here */
61      R_ADC_Start();
62      /* End user code. Do not edit comment generated here */
63  }
```

A/DのAPIを使うための宣言

A/D変換の開始

プログラム追記部分

コード生成とプログラム編集 (r_cg_ad_user.c)

```
45  /***********************************************************
46  Global variables and functions
47  ***********************************************************/
48  /* Start user code for global. Do not edit comment generated here */
49  extern uint16_t gAdResult;
50  /* End user code. Do not edit comment generated here */
51
52  /***********************************************************
53  * Function Name: r_adc_interrupt
54  * Description  : None
55  * Arguments    : None
56  * Return Value : None
57  ***********************************************************/
58  __interrupt static void r_adc_interrupt(void)
59  {
60      /* Start user code. Do not edit comment generated here */
61      R_ADC_Stop();
62      R_ADC_Get_Result( &gAdResult );
63      /* End user code. Do not edit comment ger
64  }
65
66  /* Start user code for adding. Do not edit comment generated here */
67  /* End user code. Do not edit comment generated here */
```

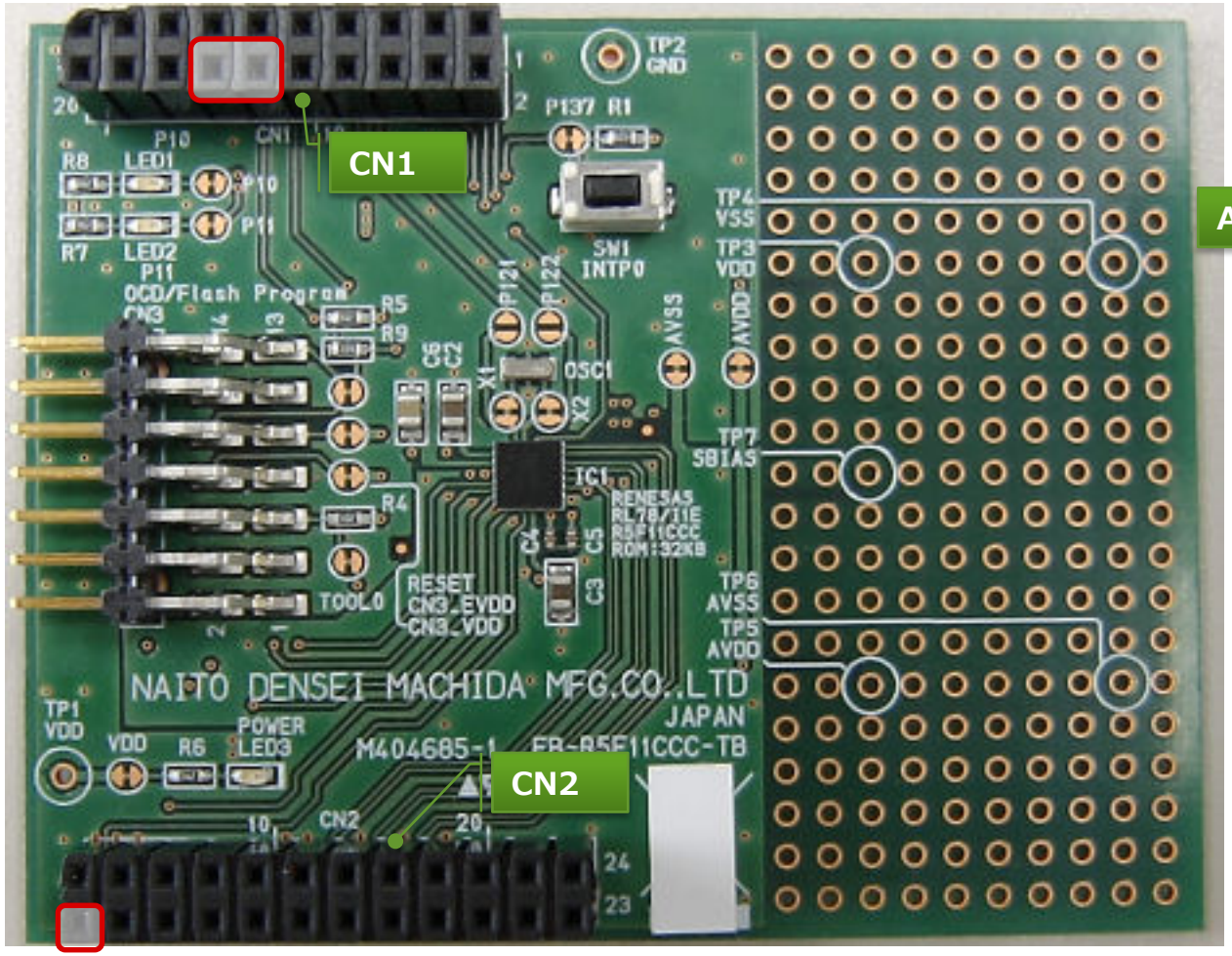
A/D変換が終了するごとに呼ばれる割り込み関数

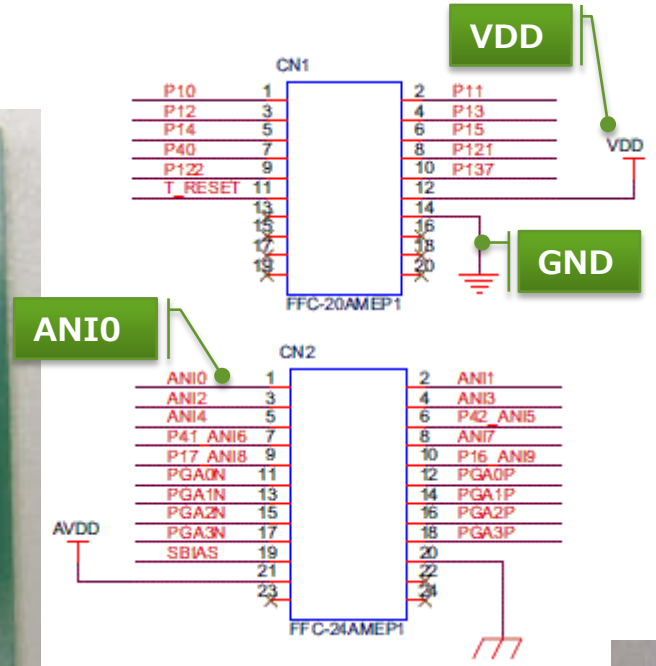
A/D変換を停止

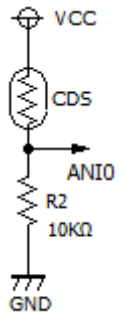
A/D変換結果を変数へ代入

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

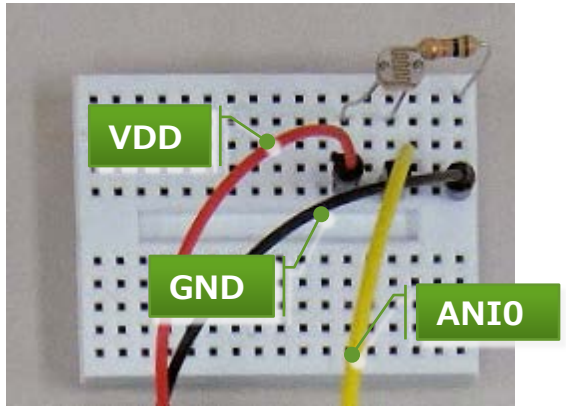
ハードウェア設定



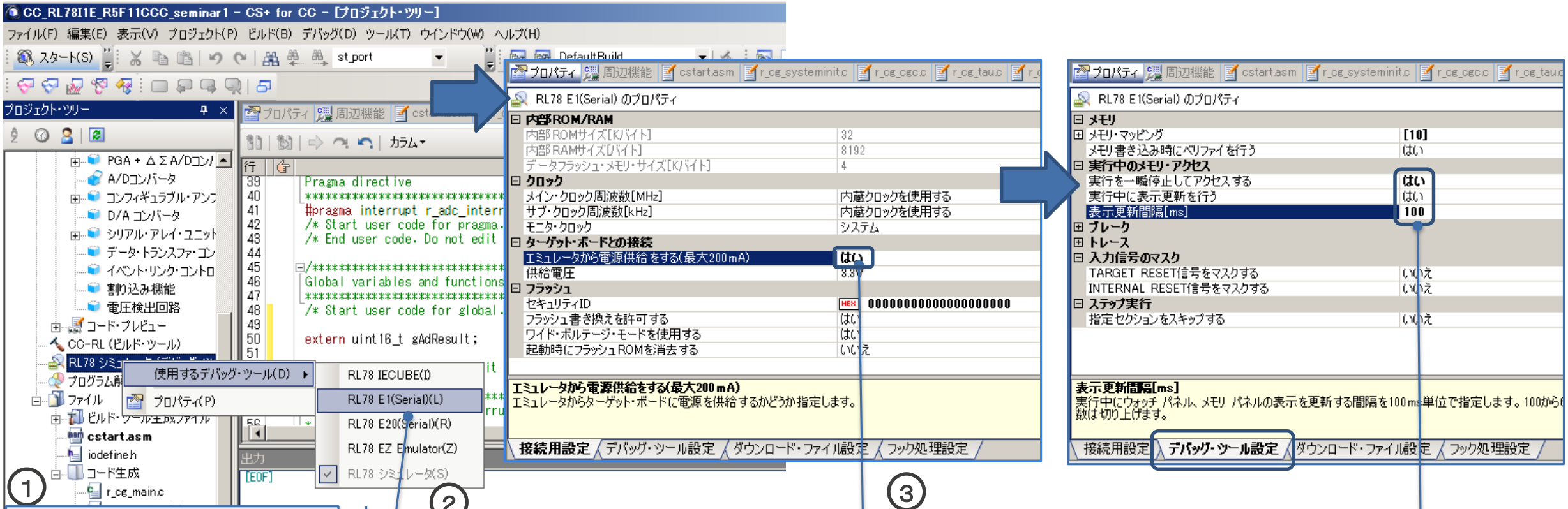




CN1 12pin VDD
CN1 14pin GND
CN2 1pin ANI0
上記をボードに接続



プログラムの実行とデバッグ (デバッグツールの設定)



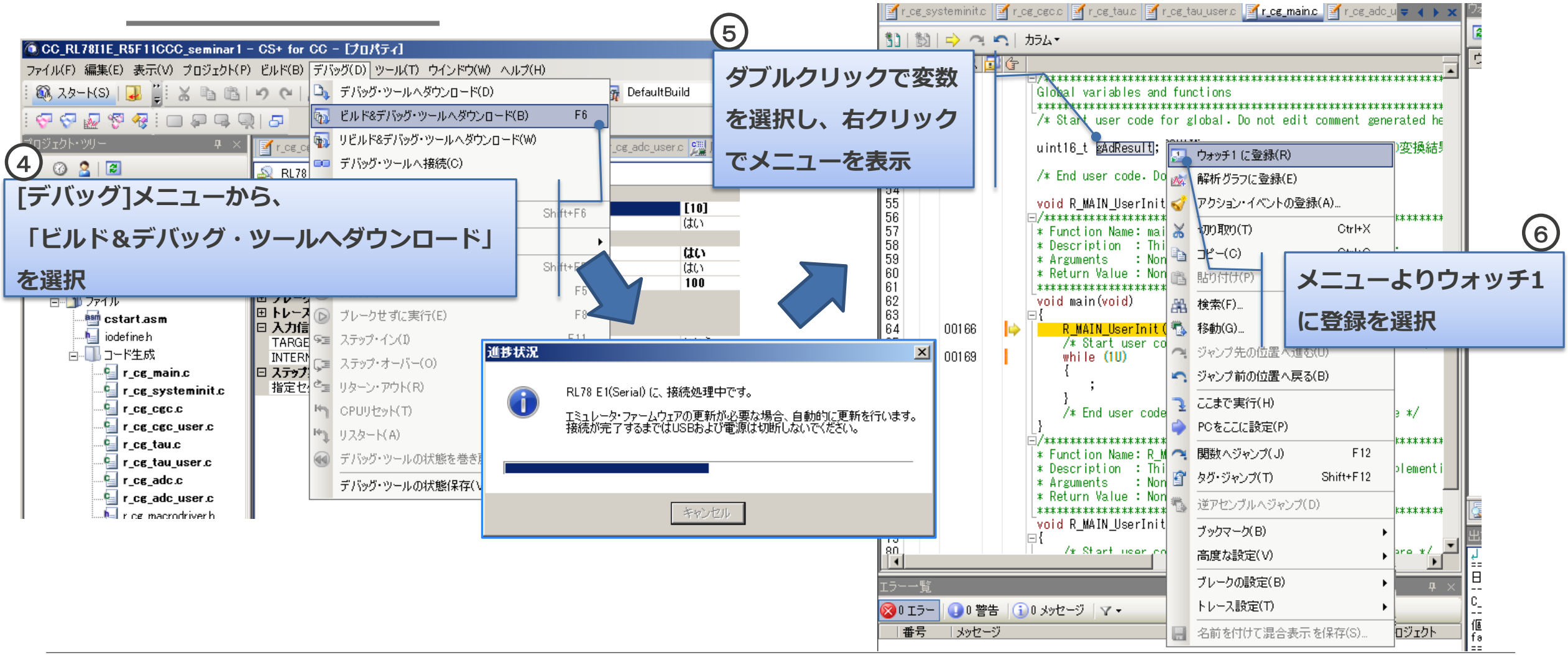
① デバッグ・ツール上で右クリック、
「使用するデバックツール」
→RL78 E1(Serial)　を選択

② E1のプロパティを開き、
「エミュレータから電源を供給する」
→ ”はい”

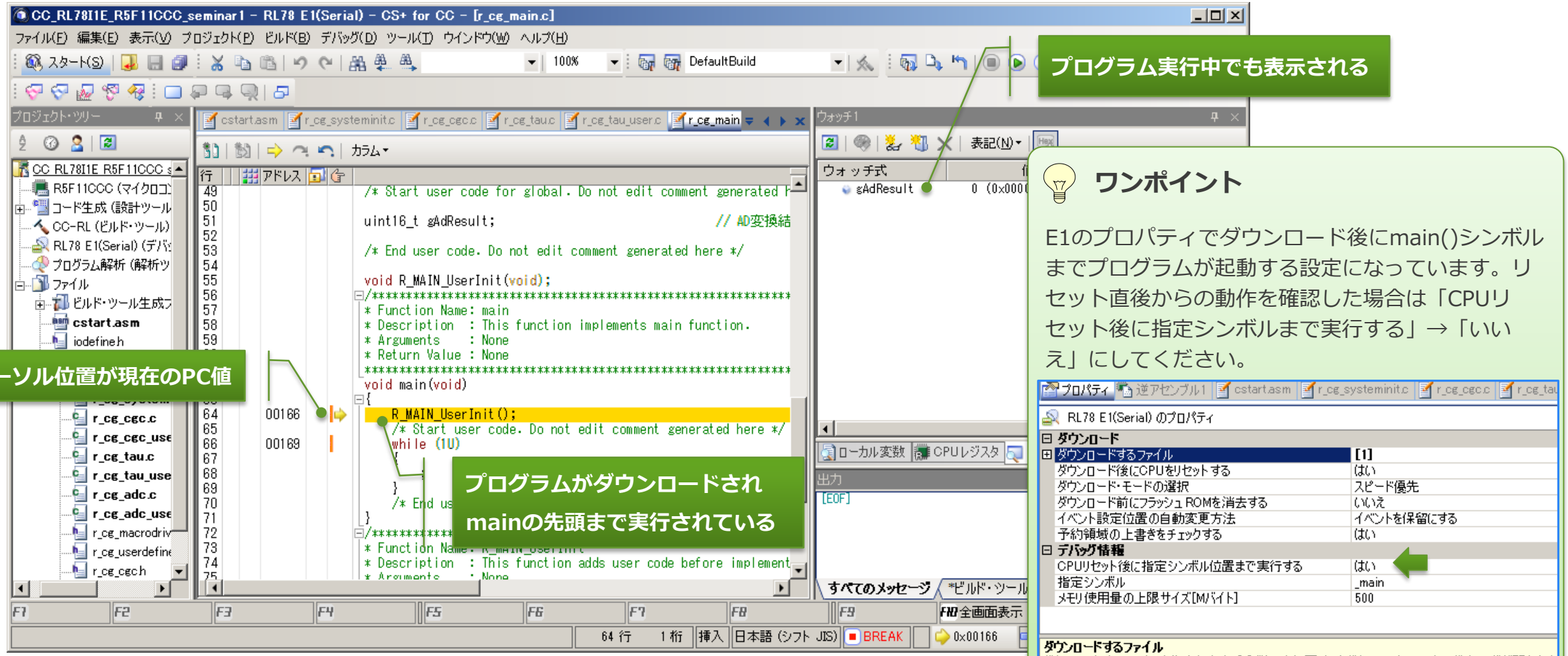
③ デバッグ・ツール設定タブを選択し、
「実行を一瞬してアクセスする」 → ”はい”
「表示更新間隔(ms)」 → ”100”

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

プログラムの実行とデバッグ (プログラムのビルド&ダウンロード)



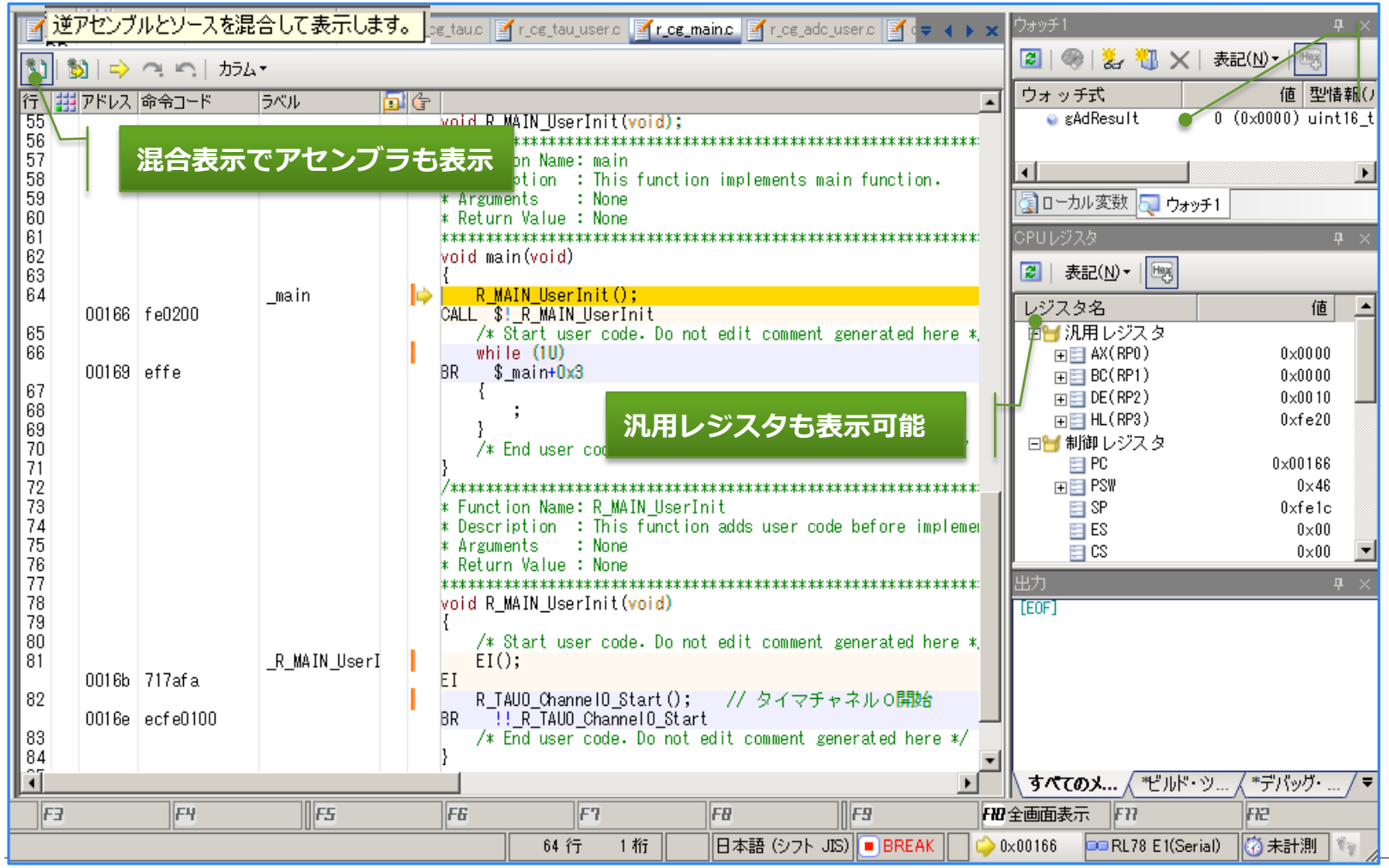
プログラムの実行とデバッグ (デバッグ画面の説明)



ワンポイント

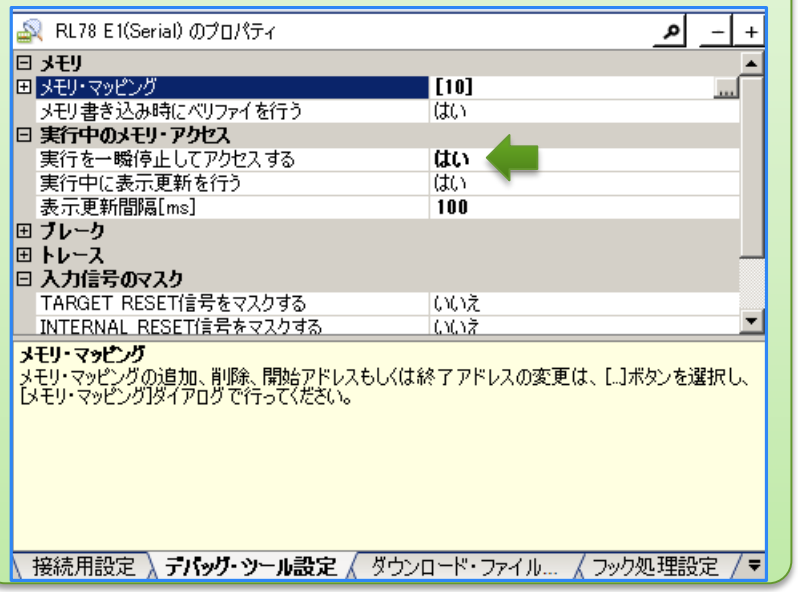
E1のプロパティでダウンロード後にmain()シンボルまでプログラムが起動する設定になっています。リセット直後からの動作を確認した場合は「CPUリセット後に指定シンボルまで実行する」→「いいえ」にしてください。

プログラムの実行とデバッグ (デバッグ画面の説明)

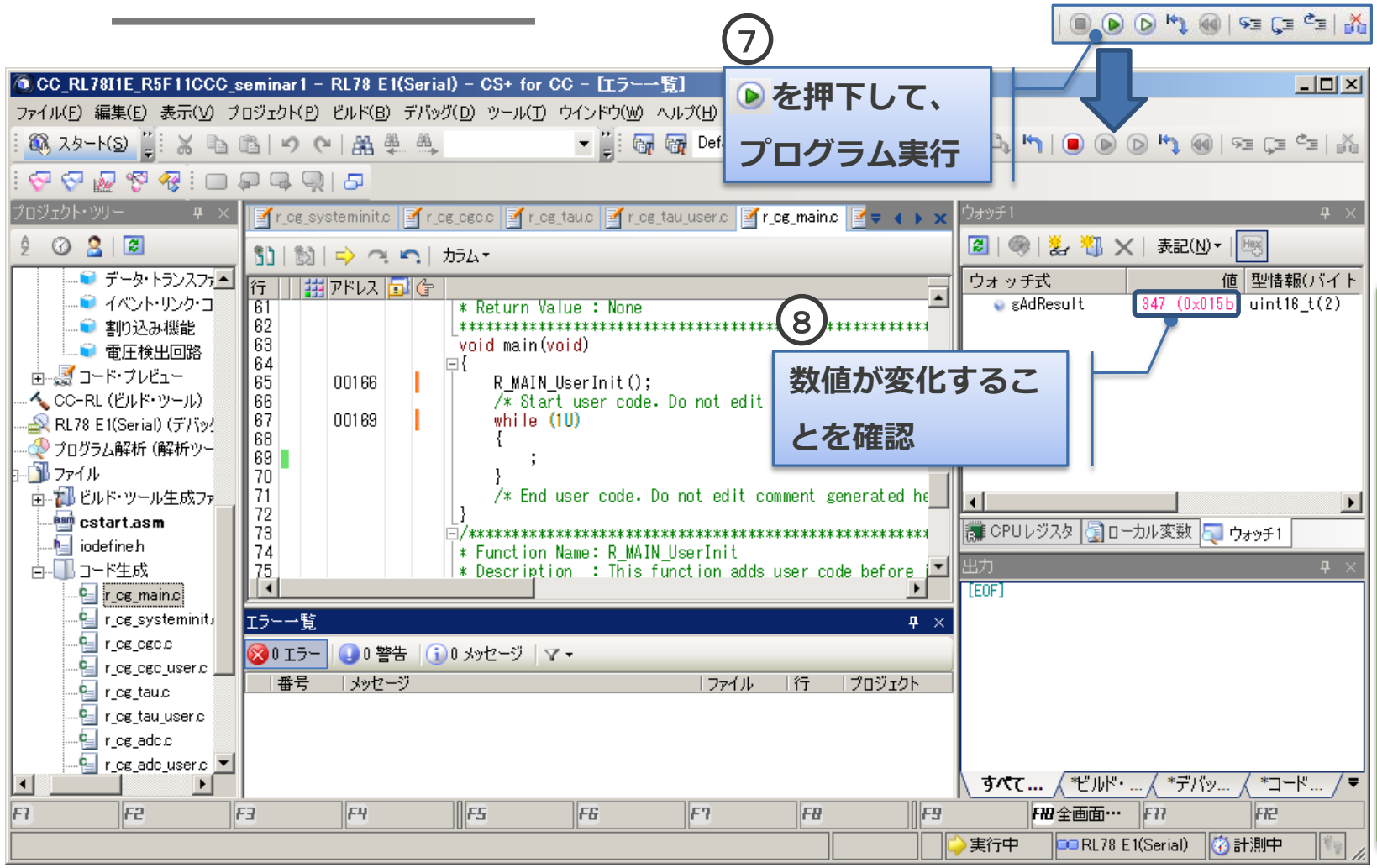


ワンポイント

プログラム実行中に変数を表示させるためには、E1のデバッグツール設定で「実行を一瞬停止してアクセスする」”はい”にします。この一瞬という時間はウォッチに登録する変数の数で変わります。変数が1つなら、停止する時間はおよそ**3μ秒**です。

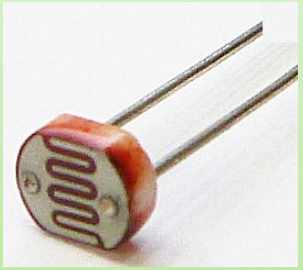


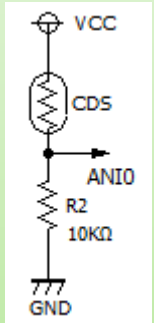
プログラムの実行とデバッグ (ウォッチパネルで確認)



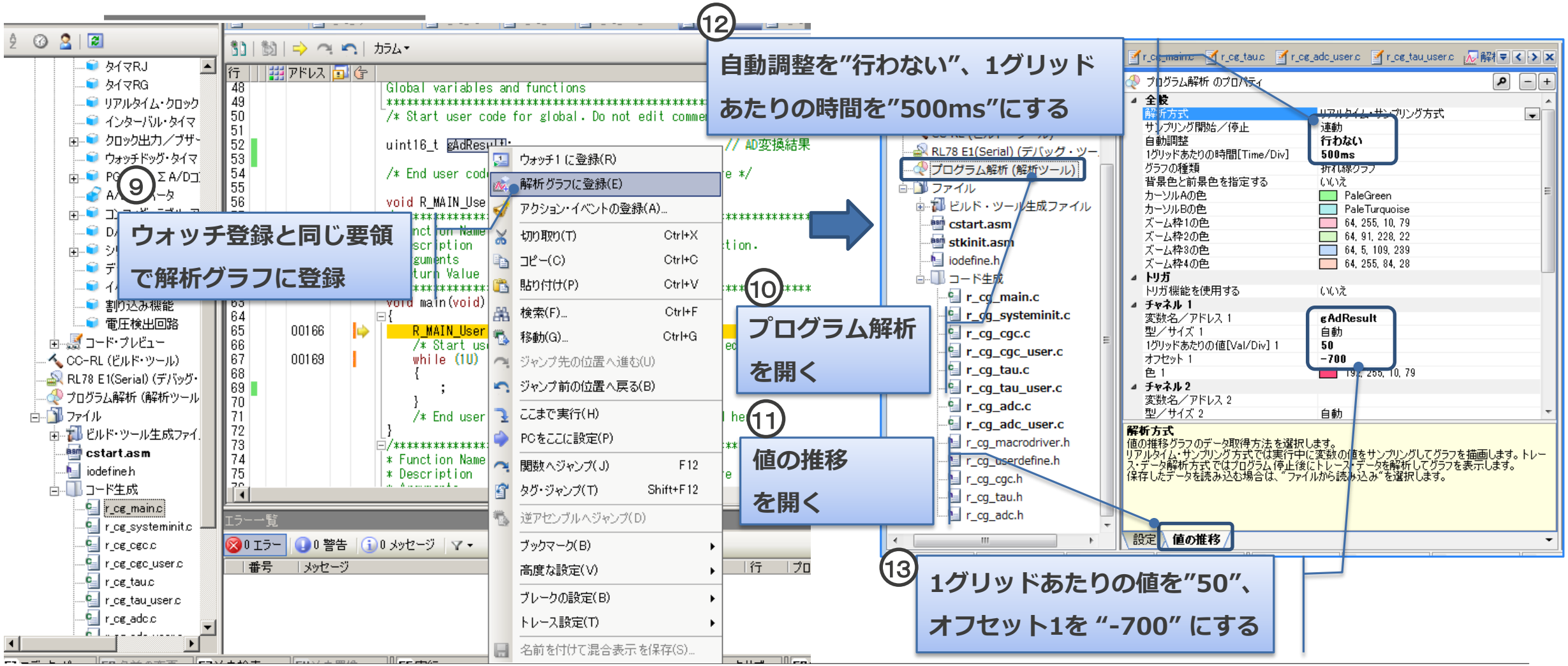
ワンポイント

CDSは暗いと抵抗値が高くなり、明るいと抵抗値が低くなります。回路は抵抗分圧しています。
そのため、CDSに光を当てると抵抗値が下がりANI0の電圧が上がります。逆に手をかざして暗くすると
ANI0の電圧が下がります。

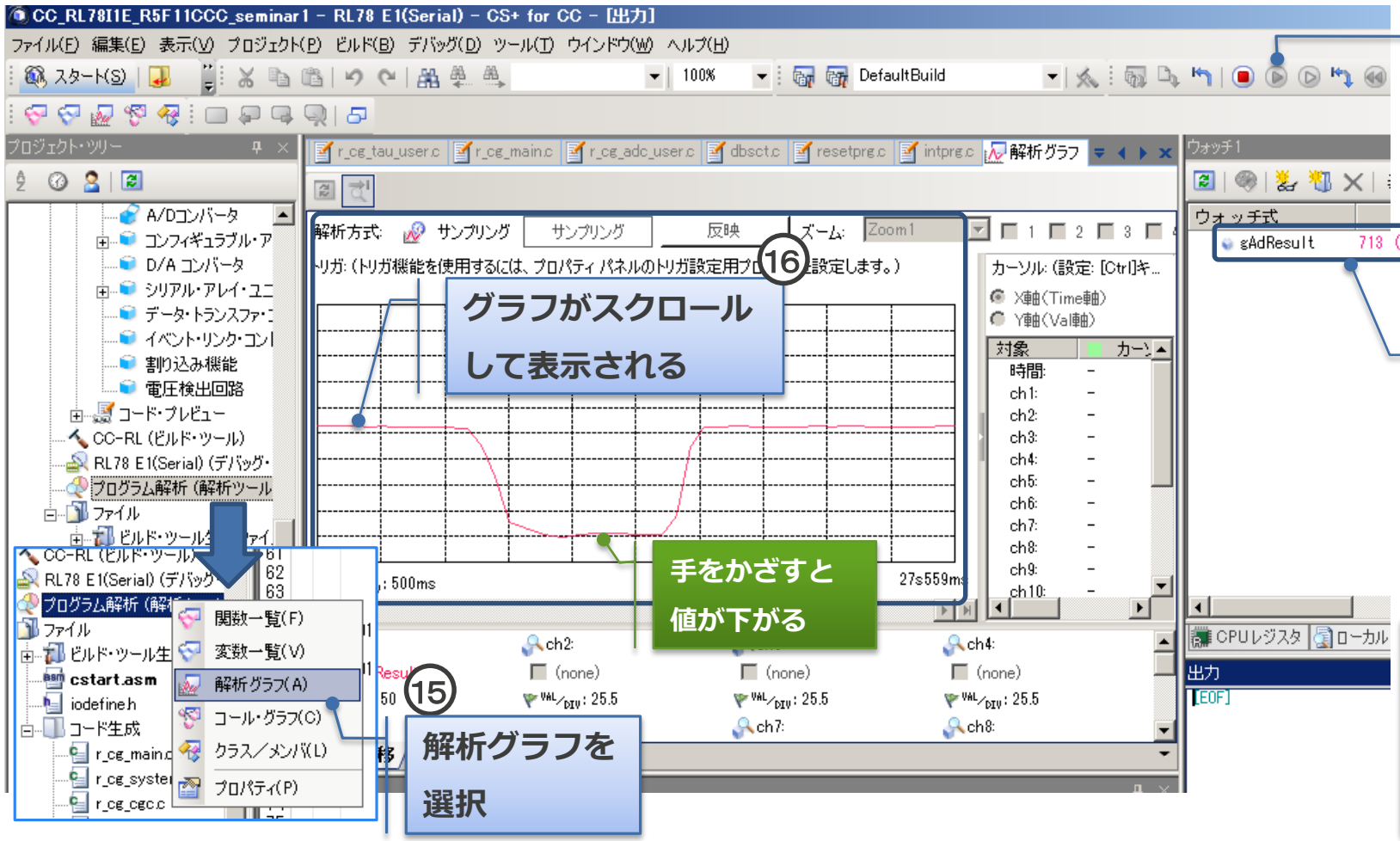




プログラムの実行とデバッグ (解析グラフ表示)



プログラムの実行とデバッグ (解析グラフ表示)

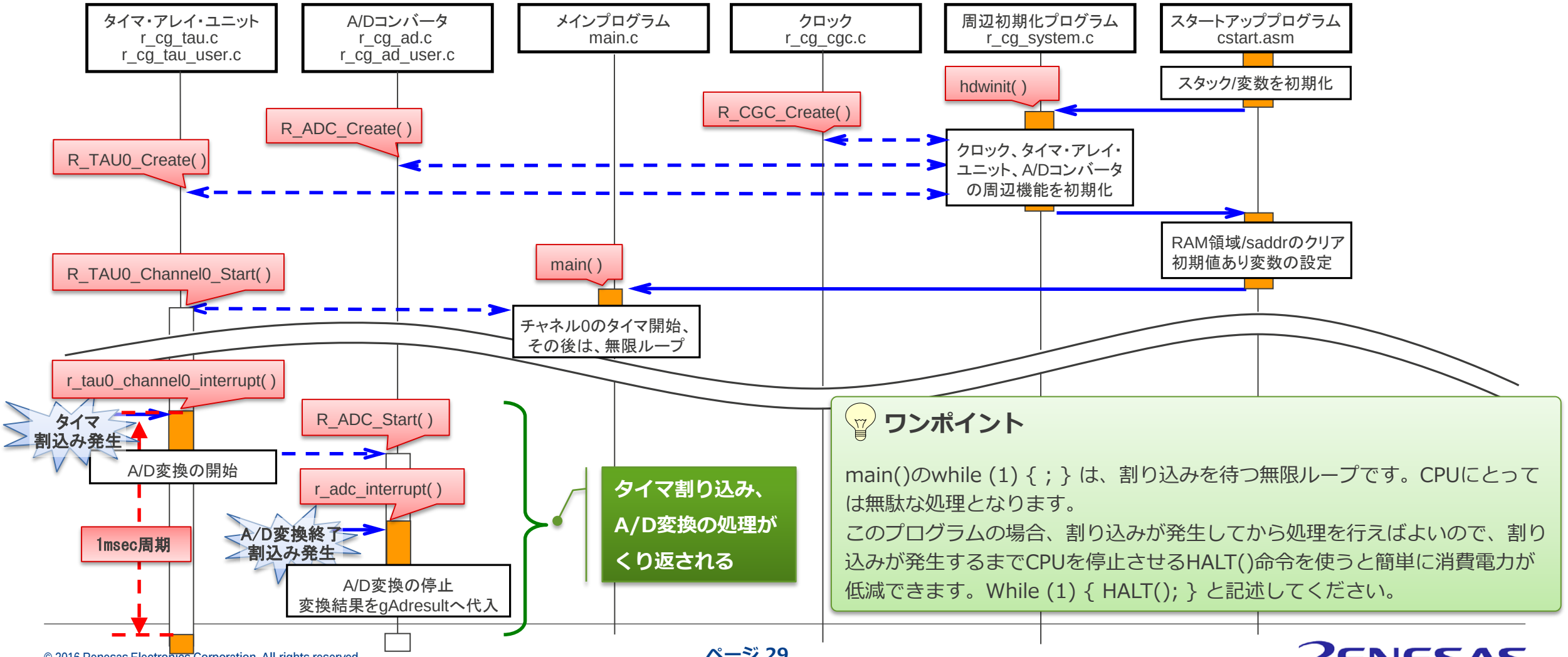


ワンポイント

表示の更新は100msが最小値です。そのため、オシロスコープのようにリアルタイムに波形を取得することはできません。

CPUを占有する処理

プログラムの解説 (リセットからmain()の処理について)



ワンポイント

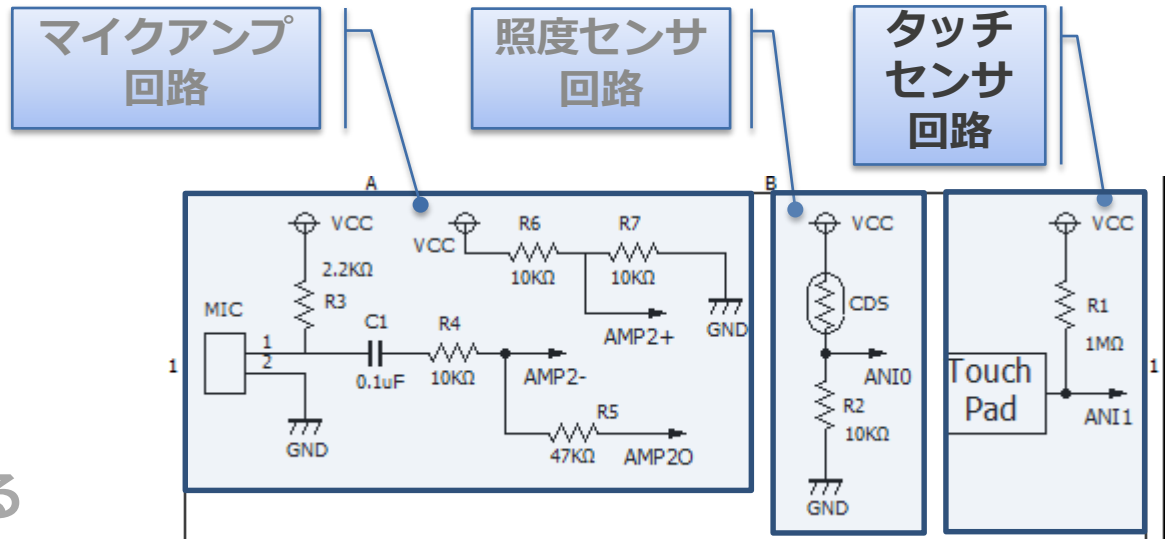
main()のwhile (1) { ; } は、割り込みを待つ無限ループです。CPUにとっては無駄な処理となります。
このプログラムの場合、割り込みが発生してから処理を行えばよいので、割り込みが発生するまでCPUを停止させるHALT()命令を使うと簡単に消費電力が低減できます。While (1) { HALT(); } と記述してください。

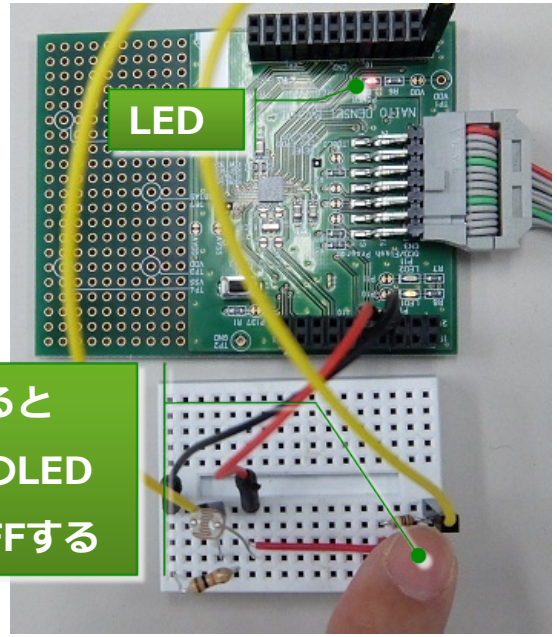
演習B

A. 照度センサを使って明るさを検知

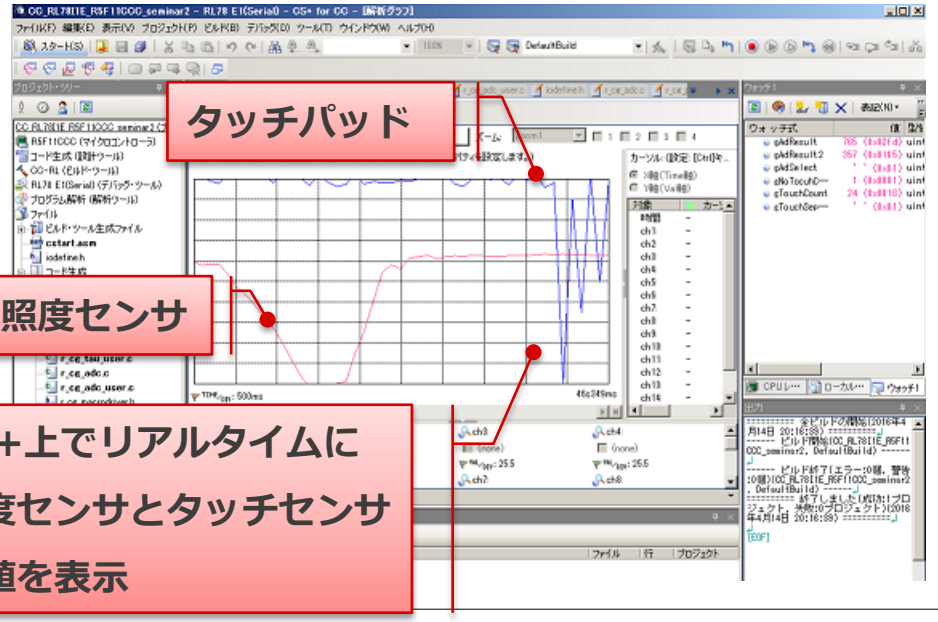
B. 簡易タッチセンサでLEDをON/OFF

C. マイクを使って音でLEDの輝度を変える





タッチすると
ボード上のLED
がON/OFFする



CS+上でリアルタイムに
照度センサとタッチセンサ
の値を表示

処理のポイント

接続は下記の通り
ANI0:照度センサ
ANI1:タッチパッド
Port10: LED

タッチパッドに触るとボード上のLED1(P10)がON/OFFする

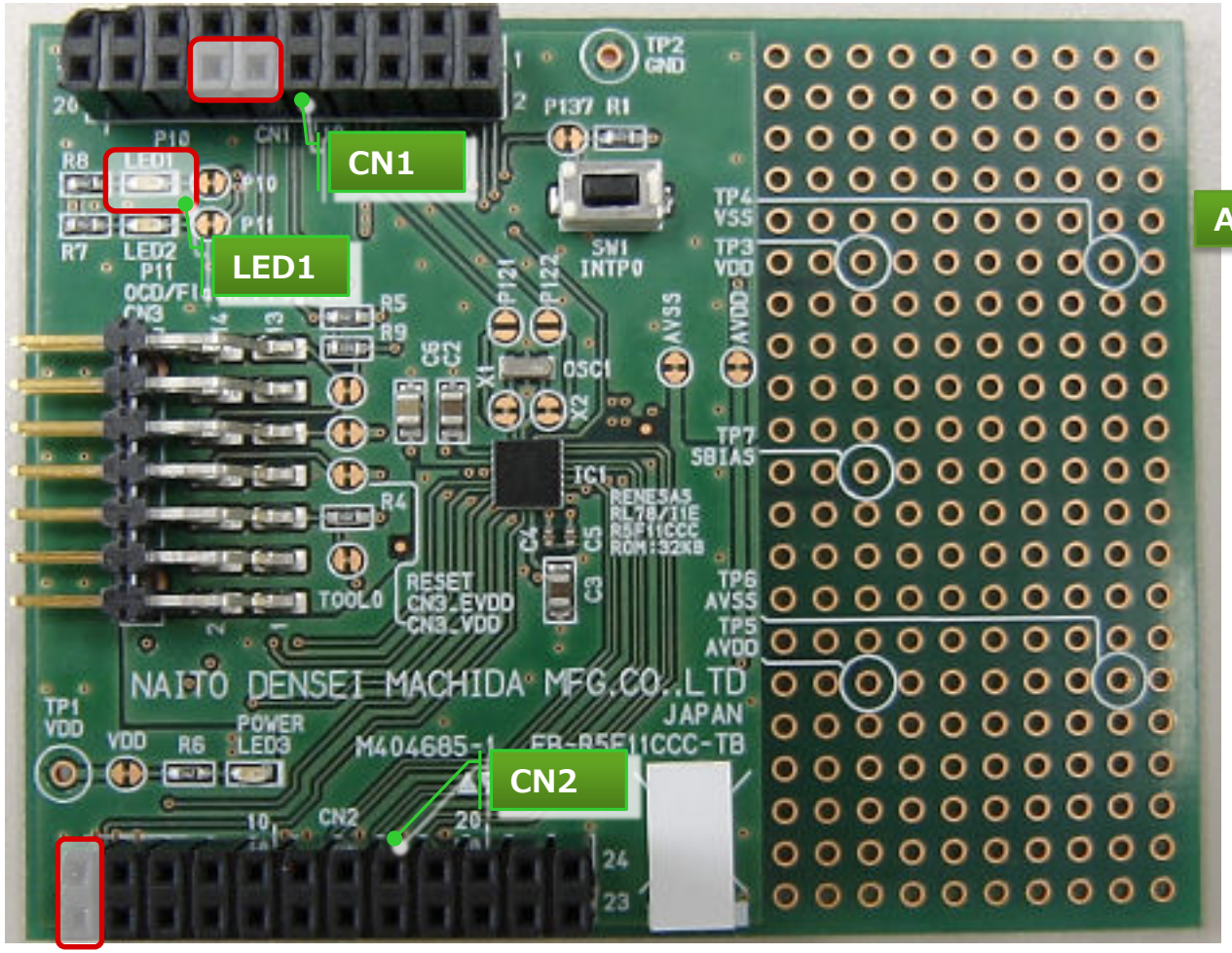
1msecごとにA/D変換を行い、ANI0とANI1を1msecごとに切り替える

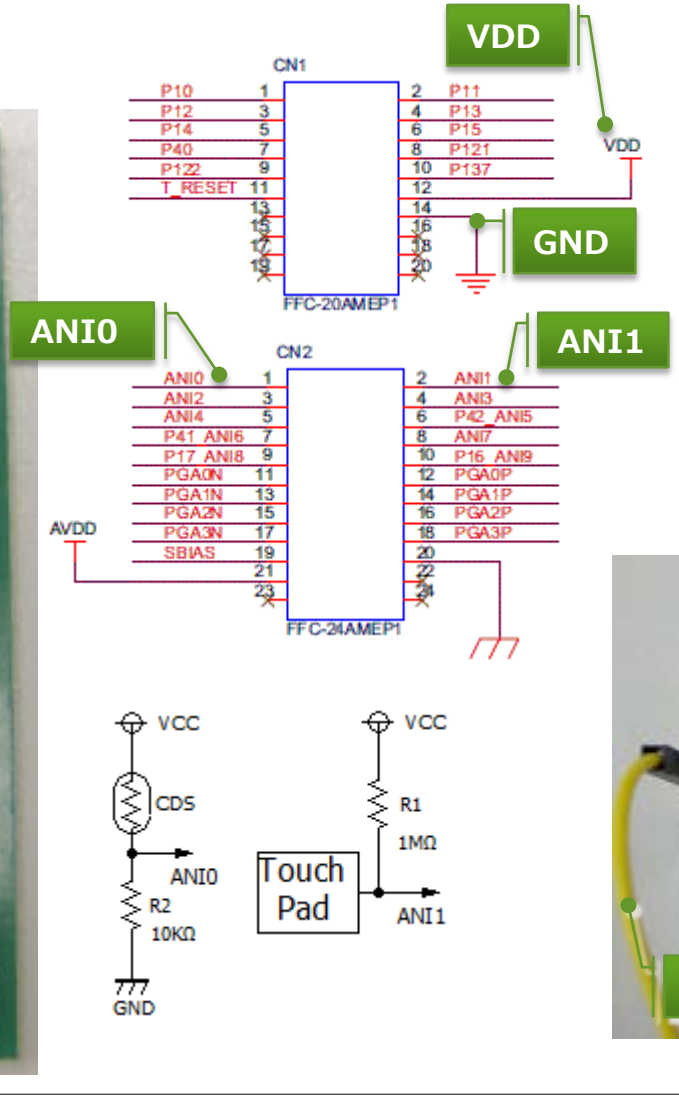
A. 照度センサを使って明るさを検知

B. 簡易タッチセンサでLEDをON/OFF

C. マイクを使って音でLEDの輝度を変える

ハードウェア設定





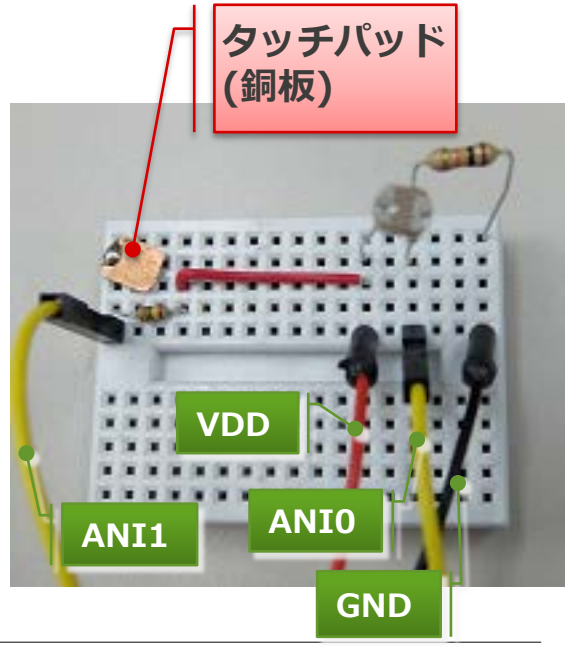
CN1 12pin VDD

CN1 14pin GND

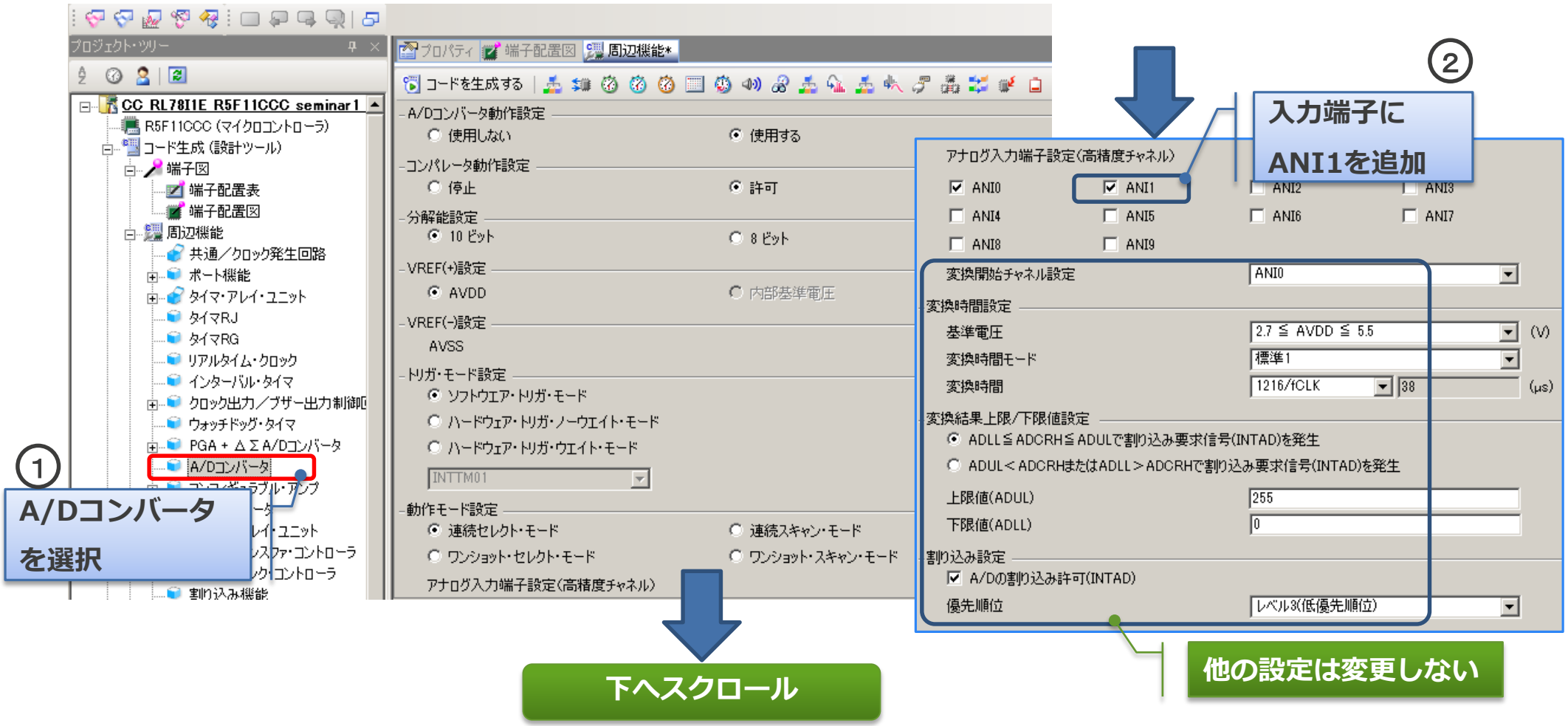
CN2 1pin ANI0

CN2 2pin ANI1

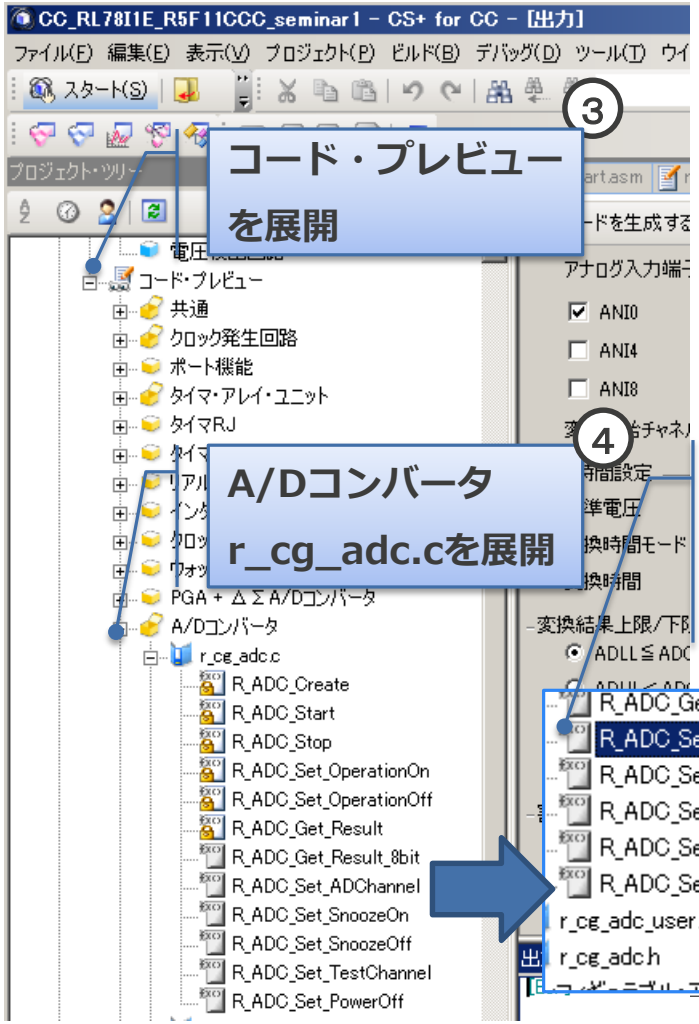
ANI1へ接続する回路を追加

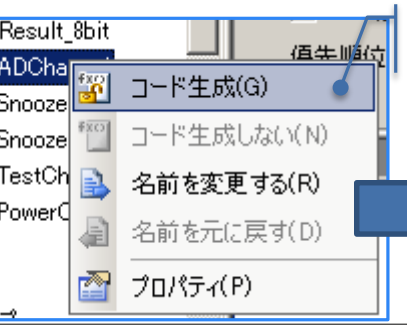


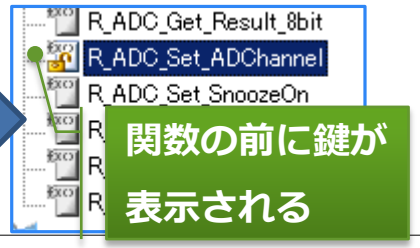
周辺機能設定 (A/Dコンバータ)

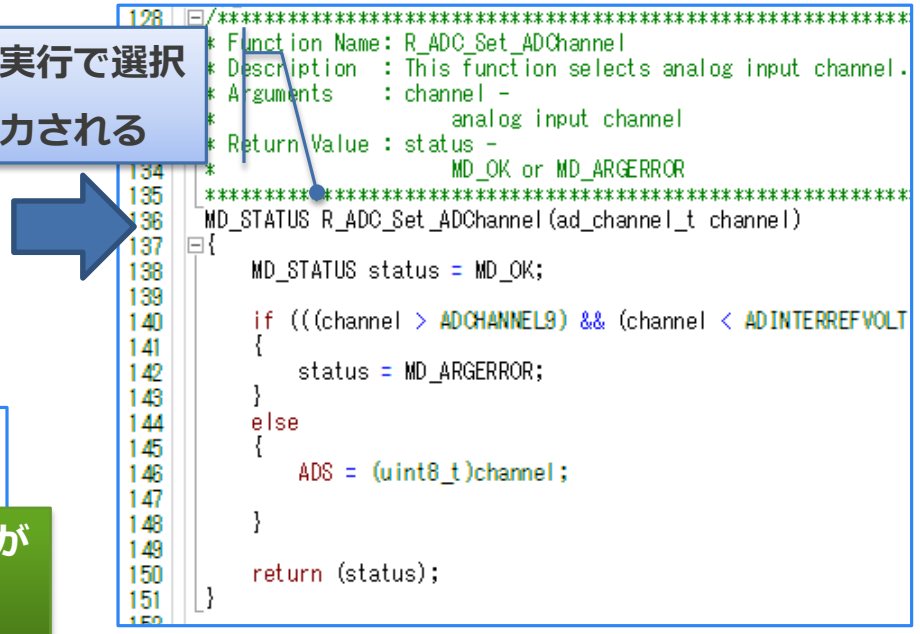


周辺機能設定 (A/Dコンバータが出力するAPIを追加)







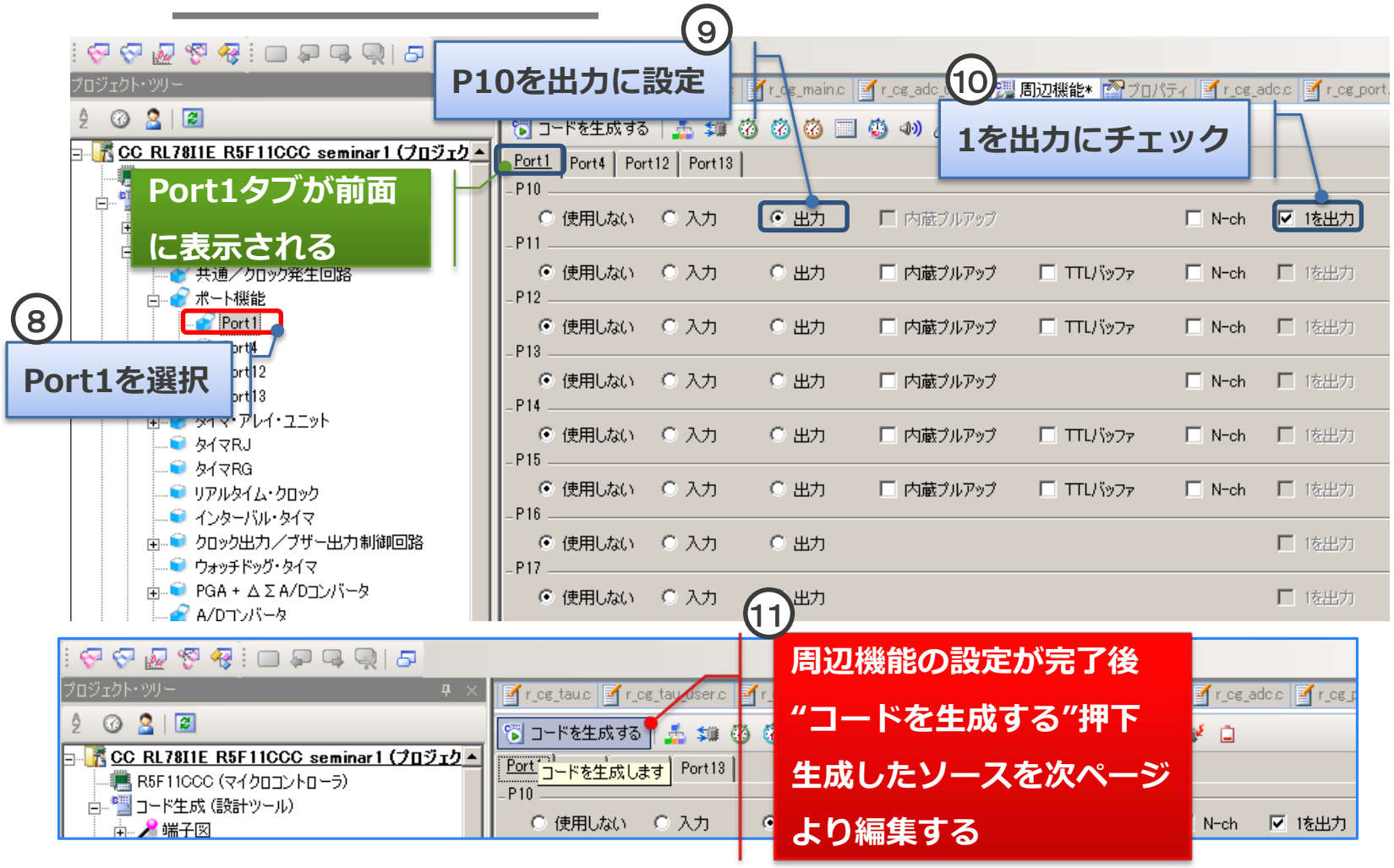


ワンポイント

R_ADC_Set_ADChannelのように、デフォルトではコード生成されない関数があります。コード・サイズ削減のため、一部のAPIは、コード生成の要否を選択式としています。
プロジェクト・ツリー・パネルのコード・プレビューにあるアイコンは下記を意味します。

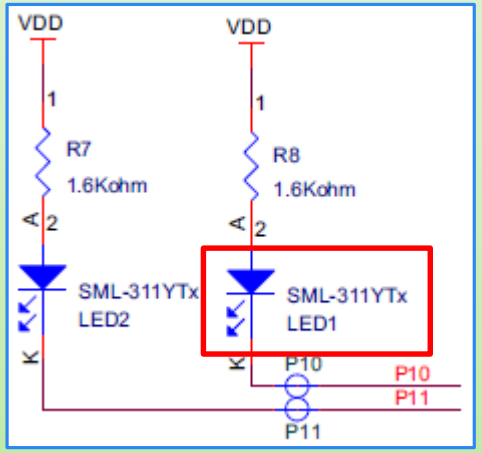
: コード生成する、　　: コード生成しない

周辺機能設定 (ポート)



ワンポイント

RL78/I1EボードはP10, P11にLEDが接続されています。LEDを点灯させるには、P10, P11をLowにします。Lowで動作することをアクティブ・ローと呼びます。1ポートにつき電流は、ローレベル出力で最大40mAです。抵抗を介してLEDが接続されているので、LEDが点灯しても5mA程度しか流れません。LEDを接続するとき、ドライバICを必ず使わなくてもよいのです。



プログラム追記部分
赤字は演習Aとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _cg_main.c)

```
47  /***********************************************************
48  Global variables and functions
49  ***********************************************************/
50  /* Start user code for global. Do not edit comment generated here */
51
52  uint16_t gAdResult;         // AD変換結果
53  uint16_t gAdResult2;        // AD変換結果2
54  uint8_t  gAdSelect;         // 変換するA/Dチャネル 0 = ANI0, 1= ANI1
55  uint16_t gTouchCount;       // タッチパネルに触れていると判断した数
56  uint16_t gNoTocuhCount;     // タッチパネルに触ってない回数
57  uint8_t  gTouchSeparate;    // タッチパネルに触っている状態 0=OFF, 1=ON
58
59
60  /* End user code. Do not edit comment generated here */
61
62  void R_MAIN_UserInit(void);
63  /***********************************************************
64  * Function Name: main
65  * Description  : This function implements main function.
66  * Arguments    : None
67  * Return Value : None
68  ***********************************************************/
69  void main(void)
70  {
71      R_MAIN_UserInit();
72      /* Start user code. Do not edit comment generated here */
73      while (1U)
74      {
75          ;
76      }
77      /* End user code. Do not edit comment generated here */
78  }
```

```
79  /***********************************************************
80  * Function Name: R_MAIN_UserInit
81  * Description  : This function adds user code before implementing main function.
82  * Arguments    : None
83  * Return Value : None
84  ***********************************************************/
85  void R_MAIN_UserInit(void)
86  {
87      /* Start user code. Do not edit comment generated here */
88
89      R_TAU0_Channel0_Start(); // タイマチャネル０開始
90      gAdSelect = 0;
91      gTouchCount = 0;
92      gNoTocuhCount =0;
93      EI();
94
95  /* End user code. Do not edit comment generated here */
96  }
```

プログラム追記部分
赤字は演習Aとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _tau_user.c)

```
29  /***********************************************************
30  Includes
31  ***********************************************************/
32  #include "r_cg_macrodriver.h"
33  #include "r_cg_tau.h"
34  /* Start user code for include. Do not edit comment generated here */
35  #include "r_cg_adc.h”
36  /* End user code. Do not edit comment generated here */
37  #include "r_cg_userdefine.h“
38
39  /***********************************************************
40  Pragma directive
41  ***********************************************************/
42  #pragma interrupt r_tau0_channel0_interrupt(vect=INTTM00)
43  /* Start user code for pragma. Do not edit comment generated here */
44  extern uint8_t  gAdSelect;
45  /* End user code. Do not edit comment generated here */
46
47  /***********************************************************
48  Global variables and functions
49  ***********************************************************/
50  /* Start user code for global. Do not edit comment generated here */
51  /* End user code. Do not edit comment generated here */
52
```

```
53  /***********************************************************
54  * Function Name: r_tau0_channel0_interrupt
55  * Description  : This function INTTM00 interrupt service routine.
56  * Arguments    : None
57  * Return Value : None
58  ***********************************************************/
59  static void __near r_tau0_channel0_interrupt(void)
60  {
61      /* Start user code. Do not edit comment generated here */
62      ad_channel_t ch;
63
64      switch ( gAdSelect )
65      {
66          case 0:
67              ch = ADCHANNEL0;
68              break;
69
70          case 1:
71              ch = ADCHANNEL1;
72              break;
73
74          default:
75              ch = ADCHANNEL0;
76              break;
77      }
78      R_ADC_Set_ADChannel( ch );
79      R_ADC_Start();
80
81  /* End user code. Do not edit comment generated here */
82  }
83
```

A/D変換を開始するときに
チャネル番号を切り替える

プログラム追記部分
赤字は演習Aとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _adc_user.c)

```
20  /***********************************************************
21  * File Name    : r_cg_adc_user.c
22  * Version      : Code Generator for RL78/I1E V1.02.00.06 [12 Aug 2015]
23  * Device(s)    : R5F11CCC
24  * Tool-Chain   : CCRL
25  * Description  : This file implements device driver for ADC module.
26  * Creation Date: 2016/04/08
27  ***********************************************************/
                          .
                          .
                          .
45  /***********************************************************
46  Global variables and functions
47  ***********************************************************/
48  /* Start user code for global. Do not edit comment generated here */
49  extern uint16_t gAdResult;
50  extern uint16_t gAdResult2;
51  extern uint8_t  gAdSelect;
52  extern uint16_t gTouchCount;
53  extern uint16_t gNoTocuhCount;
54  extern uint8_t gTouchSeparate;
55  /* End user code. Do not edit comment generated here */
56
57  /***********************************************************
58  * Function Name: r_adc_interrupt
59  * Description  : None
60  * Arguments    : None
61  * Return Value : None
62  ***********************************************************/
63  static void __near r_adc_interrupt(void)
64  {
```

```
65  /* Start user code. Do not edit comment generated here */
66      uint16_t result;
67
68      R_ADC_Stop();
69      R_ADC_Get_Result( &result );
70      switch ( gAdSelect )
71      {
72          case 0:
73              gAdResult = result;
74              break;
75
76          case 1:
77              gAdResult2 = result;
78
79              if ( result > D_TOUCH_AD)
80              {
81                  gTouchCount++;
82                  gNoTocuhCount = 0;
83              }
84              else if ( result > D_NOTOUCH_ADMIN )
85               {
86                  gNoTocuhCount++;
87              }
88
89              if ( gNoTocuhCount > D_NOTOUCH_COUNT )
90              {
91                  gTouchCount = 0;
92                  gTouchSeparate = 0;
93              }
```

A/D変換結果をテンポラリ変数へ代入

照度センサの結果を代入

タッチパッドの結果を代入

次ページへ

プログラム追記部分
赤字は演習Aとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _adc_user.c)

前ページから

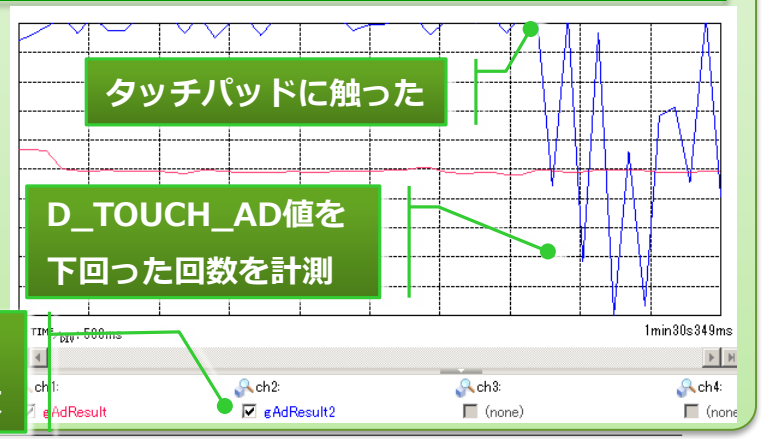
```
94
95          if ( (gTouchCount > D_TOUCH_COUNT) &&( gTouchSeparate == 0) )
96          {
97              // LED点灯と消灯を繰り返す
98              P1_bit.no0 ^= 1;
99              gTouchSeparate = 1;
100         }
101
102         break;
103
104     default:
105         gAdResult = result;
106         break;
107 }
108
109 gAdSelect++;
110 if ( gAdSelect >= D_MAXAD_CHANNEL )
111 {
112     gAdSelect = 0;
113 }
114
115 /* End user code. Do not edit comme
116 }
117
118
```

A/D変換の完了の最後の処理で次に変換するにチャネル番号を指定する

ワンポイント

タッチパッドの処理が複雑になっていますが、理由があります。下図は、タッチパッドを触った時の解析グラフの波形です。触った瞬間に電圧降下し、充電と放電を繰り返します。人間はキャパシタ(コンデンサ)の役割と同等なので、このような波形になります。

触ったと判断するのは、A/D値がD_TOUCH_AD(100)を下回った回数をカウントする変数 **gTouchCount** が一定の大きさを超えた場合です。また、タッチパッドに触ったままの状態では、何もしないようにしています。



プログラム追記部分
赤字は演習Aとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _cg_userdefine.h)

```
20  /***********************************************************
21  * File Name    : r_cg_userdefine.h
22  * Version      : Code Generator for RL78/I1E V1.02.00.06 [12 Aug 2015]
23  * Device(s)    : R5F11CCC
24  * Tool-Chain   : CCRL
25  * Description  : This file includes user definition.
26  * Creation Date: 2016/04/08
27  ***********************************************************/
28  #ifndef _USER_DEF_H
29  #define _USER_DEF_H
30
31  /***********************************************************
32  User definitions
33  ***********************************************************/
34
35  /* Start user code for function. Do not edit comment generated here */
36
37  #define D_MAX_AD_CHANNEL        2           // A/D変換するチャネル数
38
39  #define D_TOUCH_AD              100         // タッチパッドに触ったと判断する値
40  #define D_NOTOUCH_ADMIN         900         // タッチパッド非接触時を判断するA/D最小値
41  #define D_TOUCH_COUNT           10          // タッチパッドを触ったと判断する回数
42  #define D_NOTOUCH_COUNT         100         // タッチパッドを触っていないと判断する回数
43
44  /* End user code. Do not edit comment generated here */
45  #endif
46
```

ワンポイント

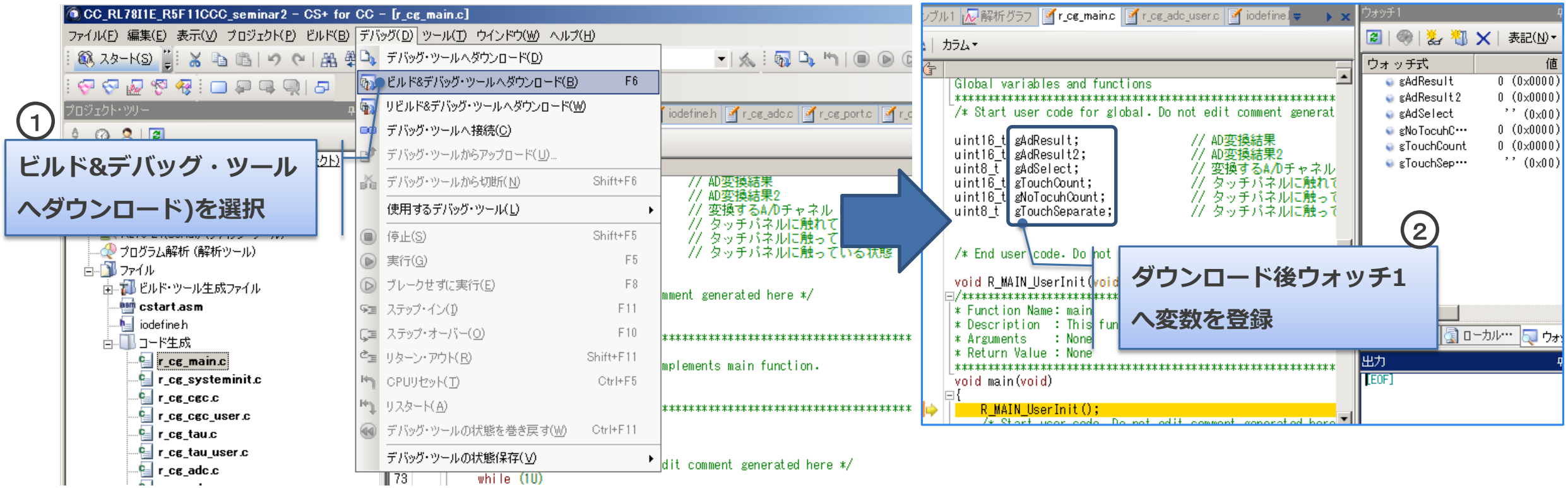
r_cg_userdefine.hは、コード生成されるファイルには必ず includeされます。プログラムを記述するときにマジックナンバーを使わないためにユーザ専用に定義するファイルを用意しました。
define だけでなく、マクロ命令、enum、typedef などを、ここへ書いてください。

A. 照度センサを使って明るさを検知

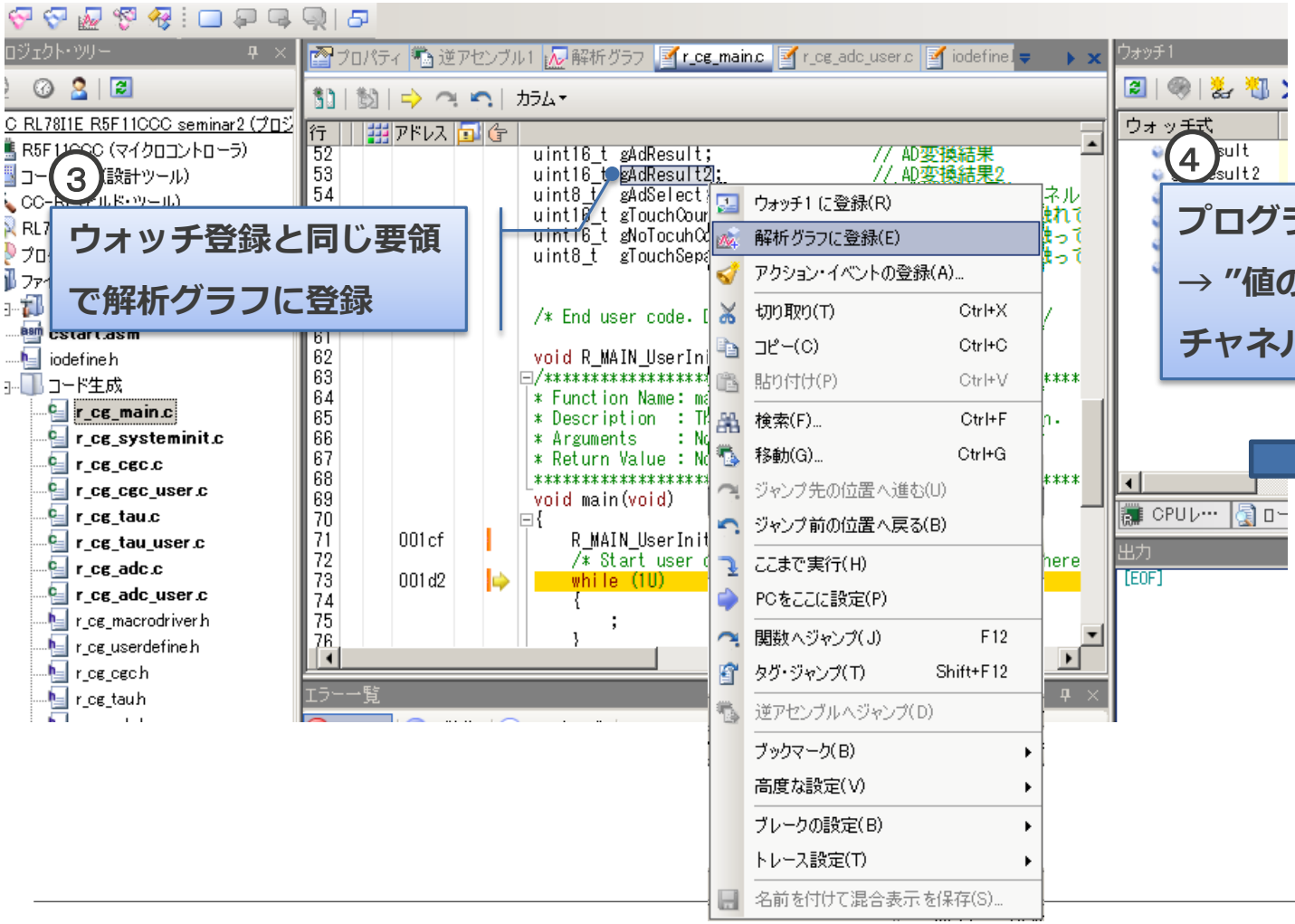
B. 簡易タッチセンサでLEDをON/OFF

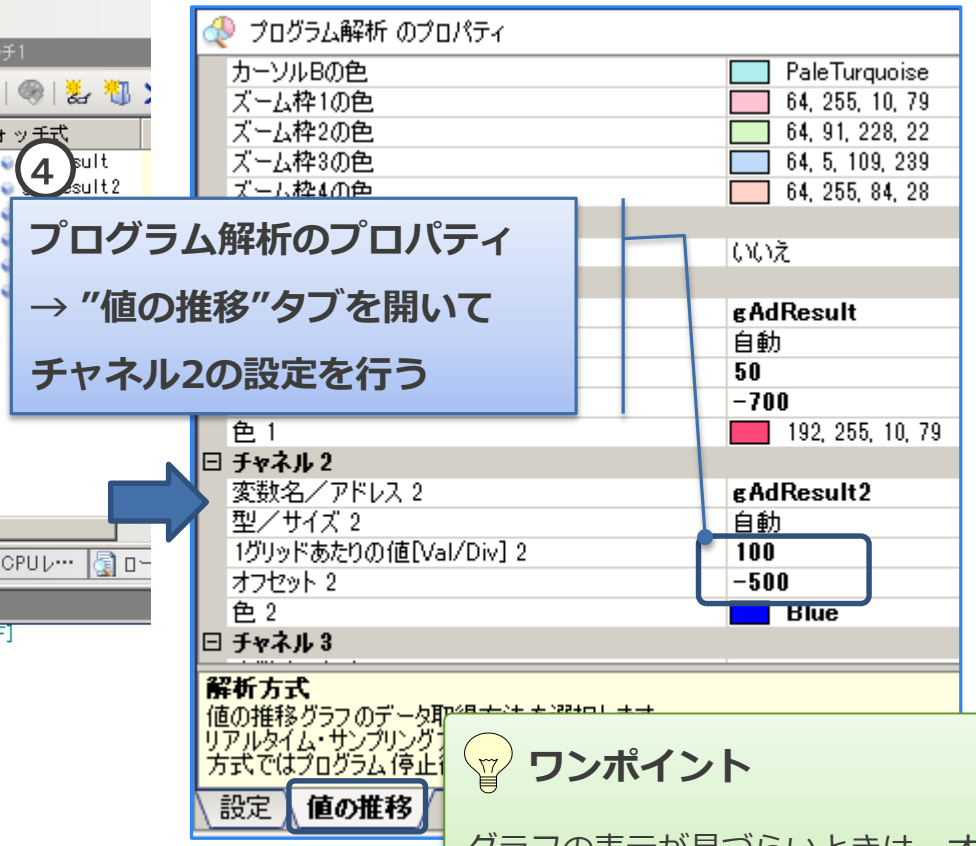
C. マイクを使って音でLEDの輝度を変える

プログラムの実行とデバッグ (プログラムのビルド＆ダウンロード)



プログラムの実行とデバッグ (解析グラフに登録)





ワンポイント

グラフの表示が見づらいときは、オフセット値や1グリッドあたりの値を大きくすると表示を調整できます。

A. 照度センサを使って明るさを検知

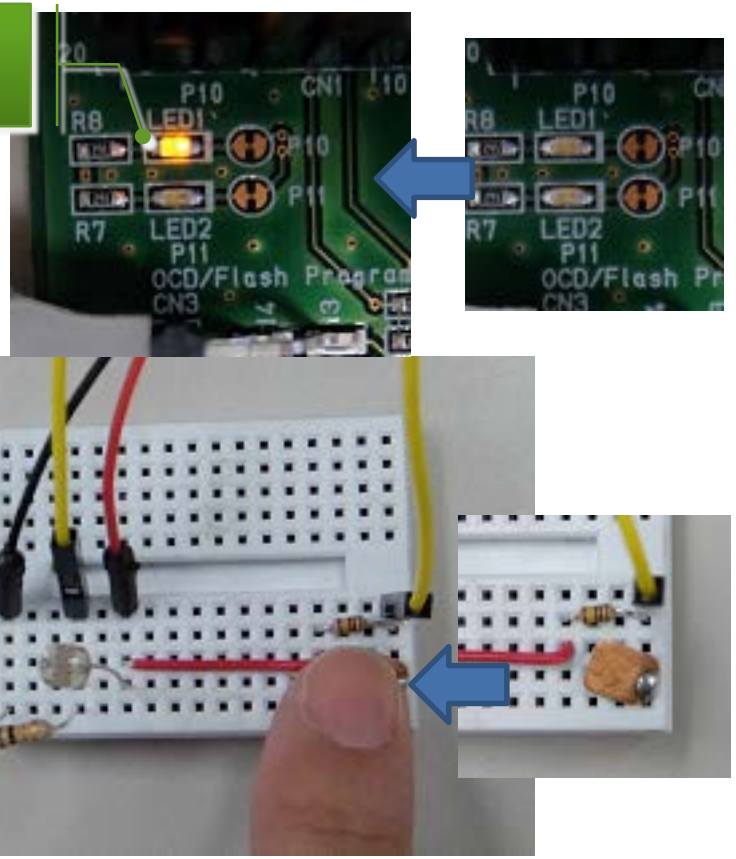
B. 簡易タッチセンサでLEDをON/OFF

C. マイクを使って音でLEDの輝度を変える

プログラムの実行とデバッグ (解析グラフ表示)

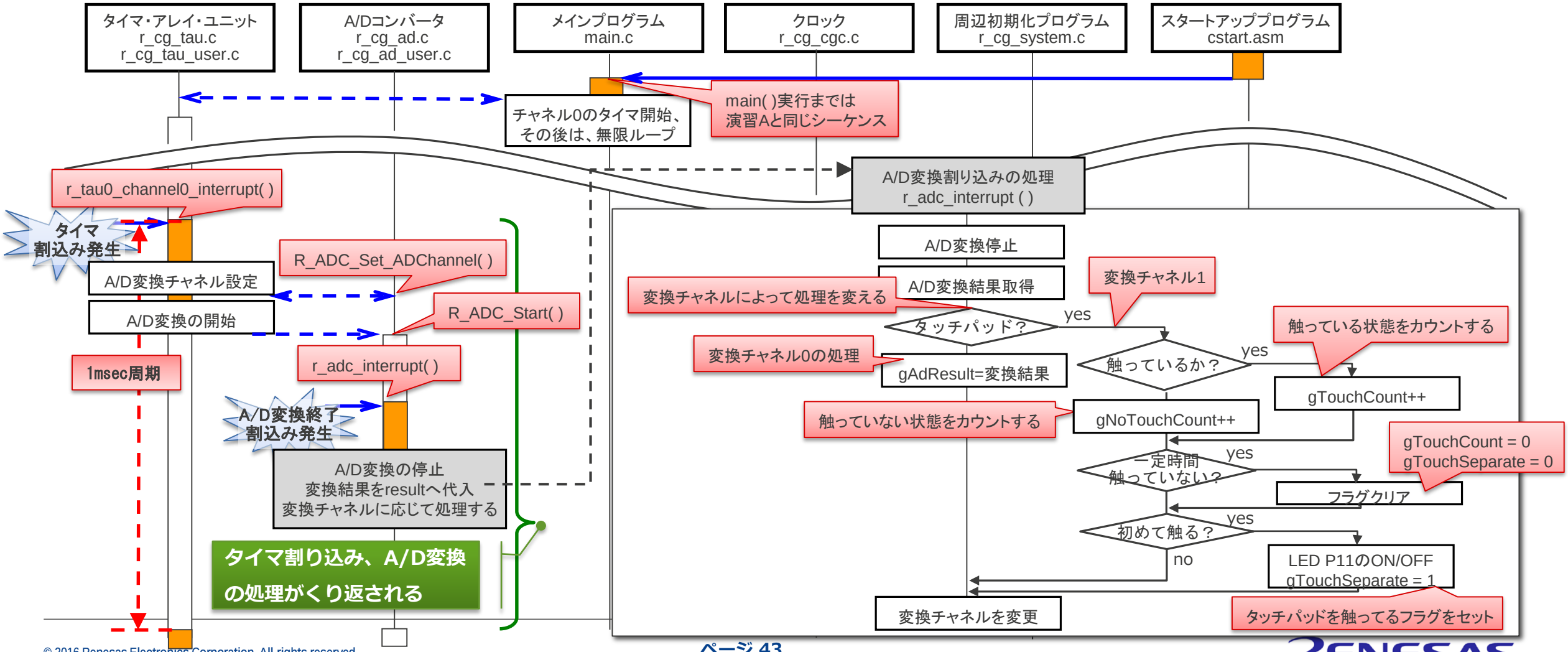
タッチパッドを触ると
LEDが点灯/消灯する

タッチパッドを触ると
グラフ表示が変化



CPUを占有する処理

プログラムの解説 (リセットからmain()の処理について)

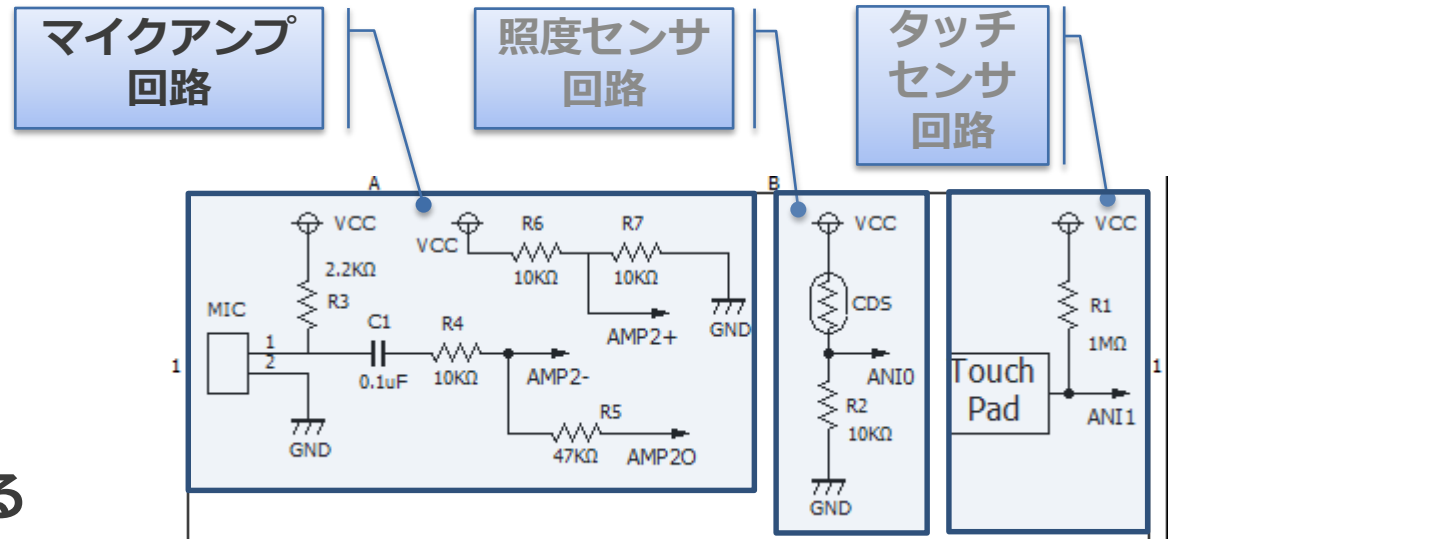


演習C

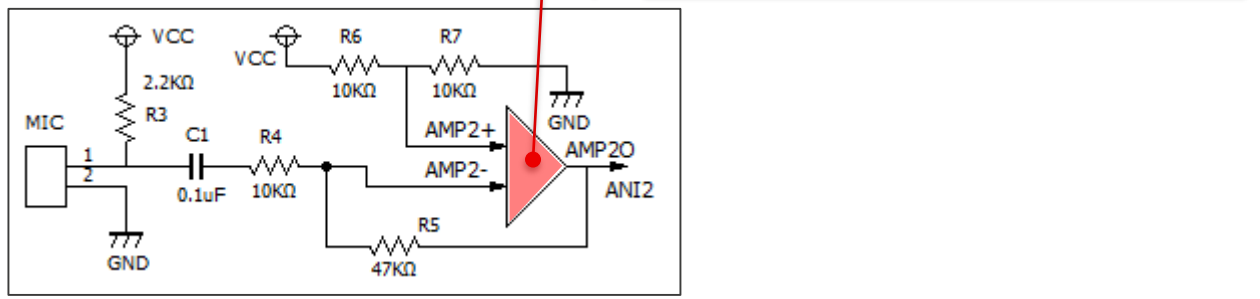
A. 照度センサを使って明るさを検知

B. 簡易タッチセンサでLEDをON/OFF

C. マイクを使って音でLEDの輝度を変える

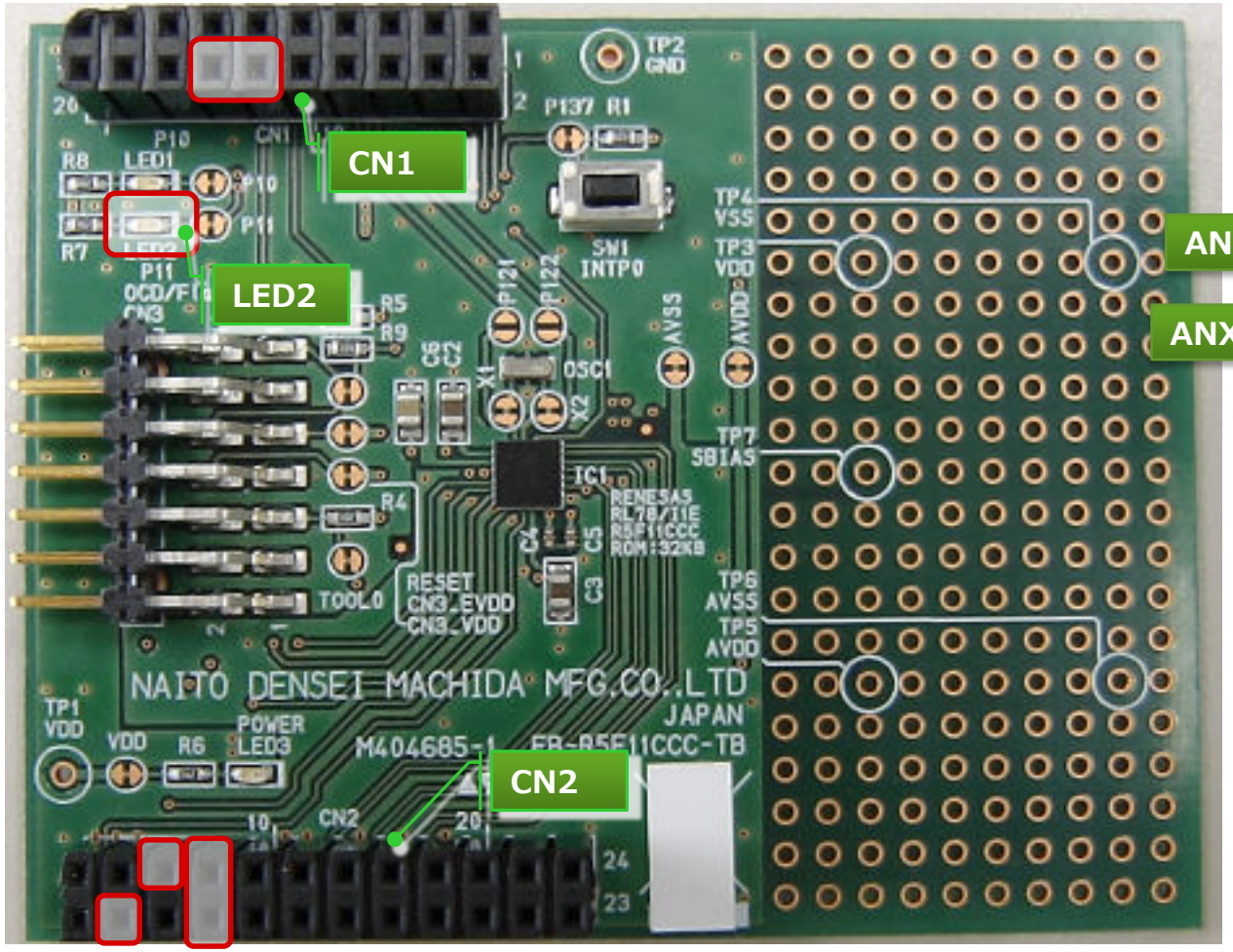


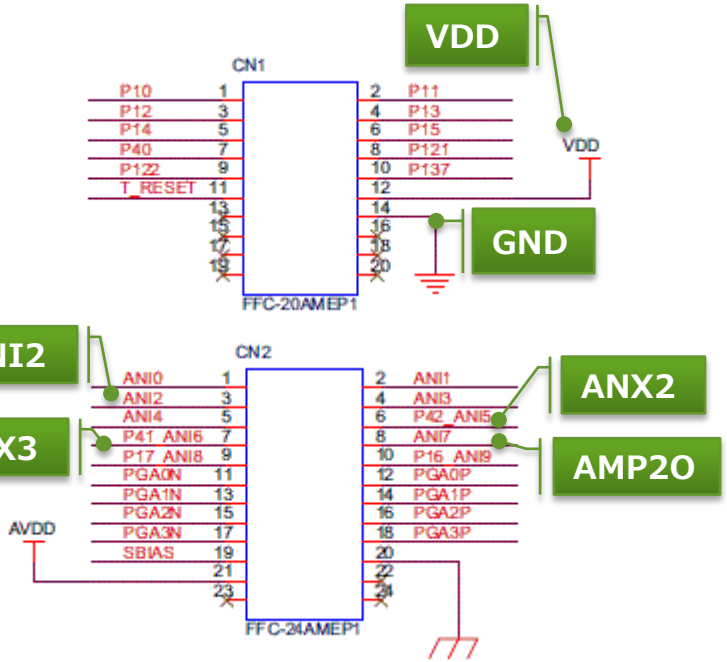
RL78/I1Eを含めた等価回路



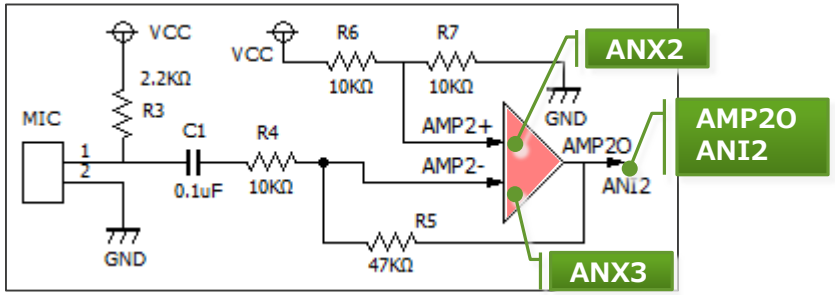
A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

ハードウェア設定



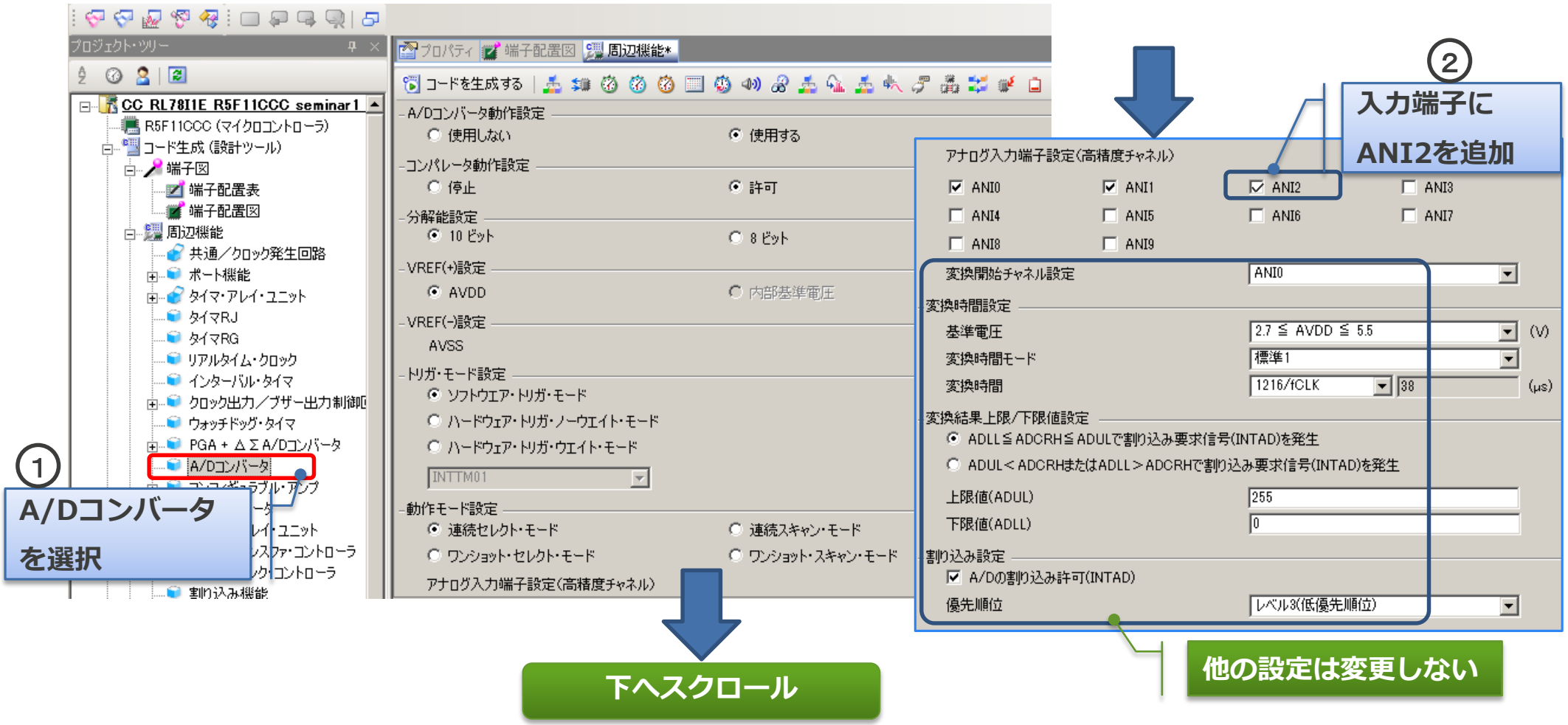


CN1 12pin	VDD
CN1 14pin	GND
CN2 3pin	**ANI2**
CN2 6pin	**ANX2**
CN2 7pin	**ANX3**
CN2 8pin	**AMP2O**



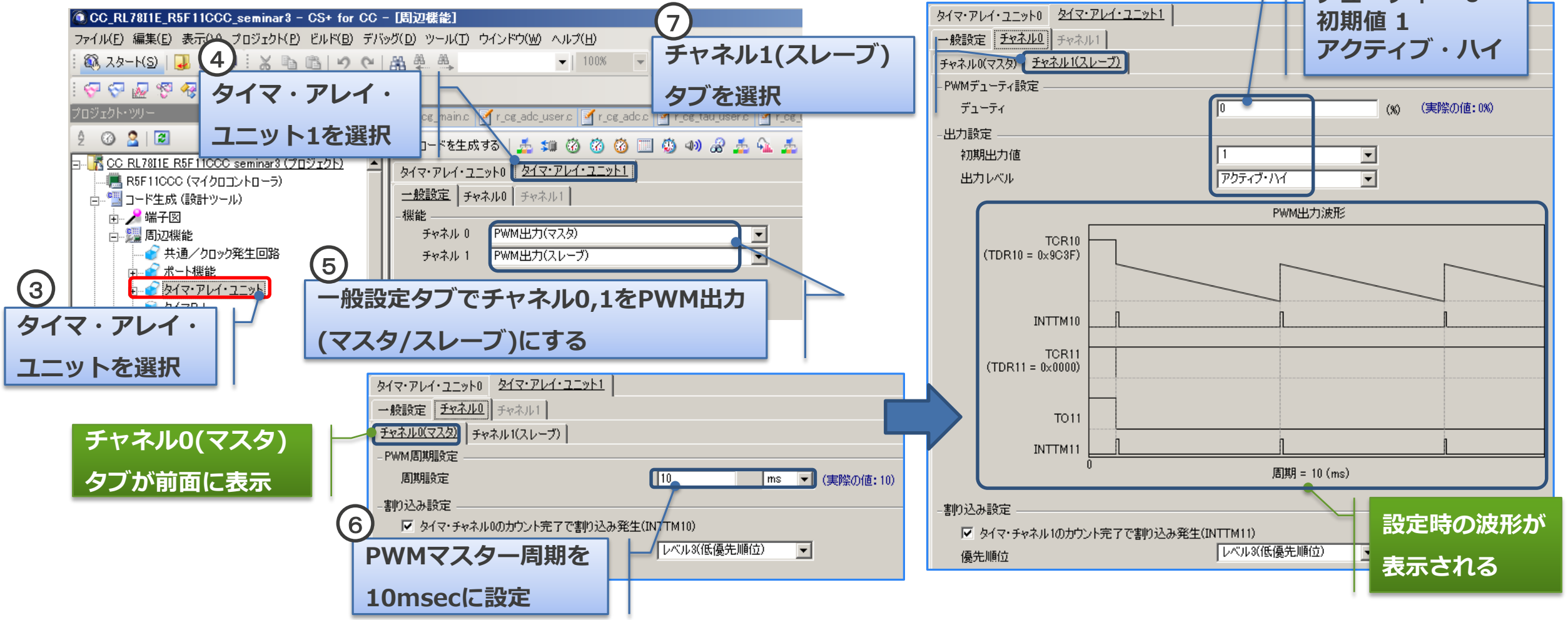
A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

周辺機能設定 (A/Dコンバータ)



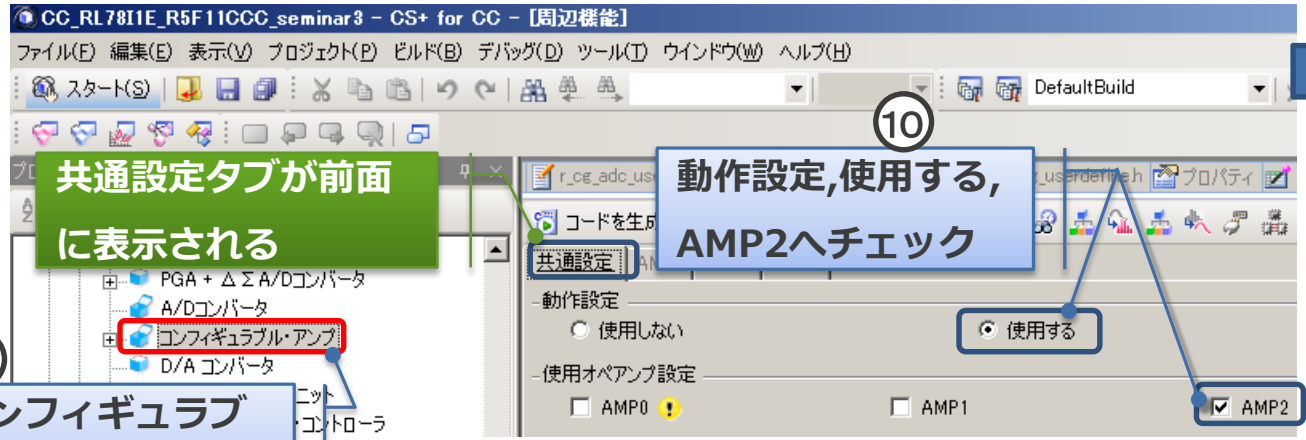
A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

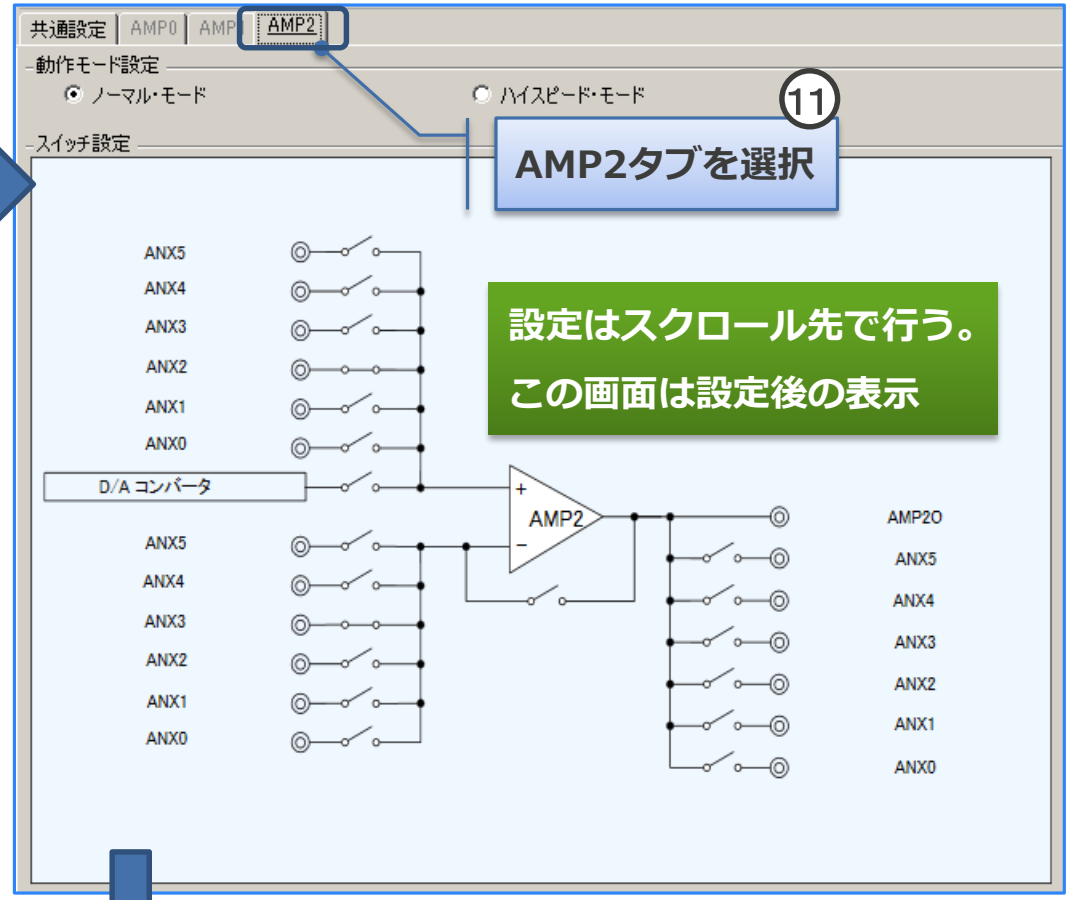
周辺機能設定 (タイマ・アレイ・ユニット)



A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

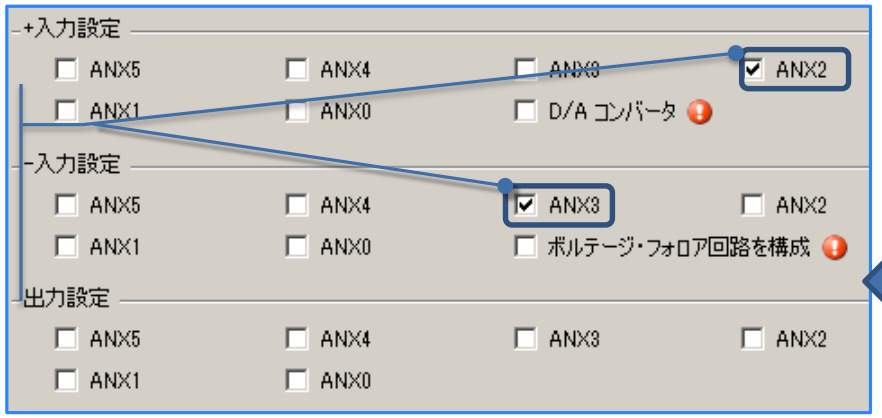
周辺機能設定 (コンフィギュラブル・アンプ)



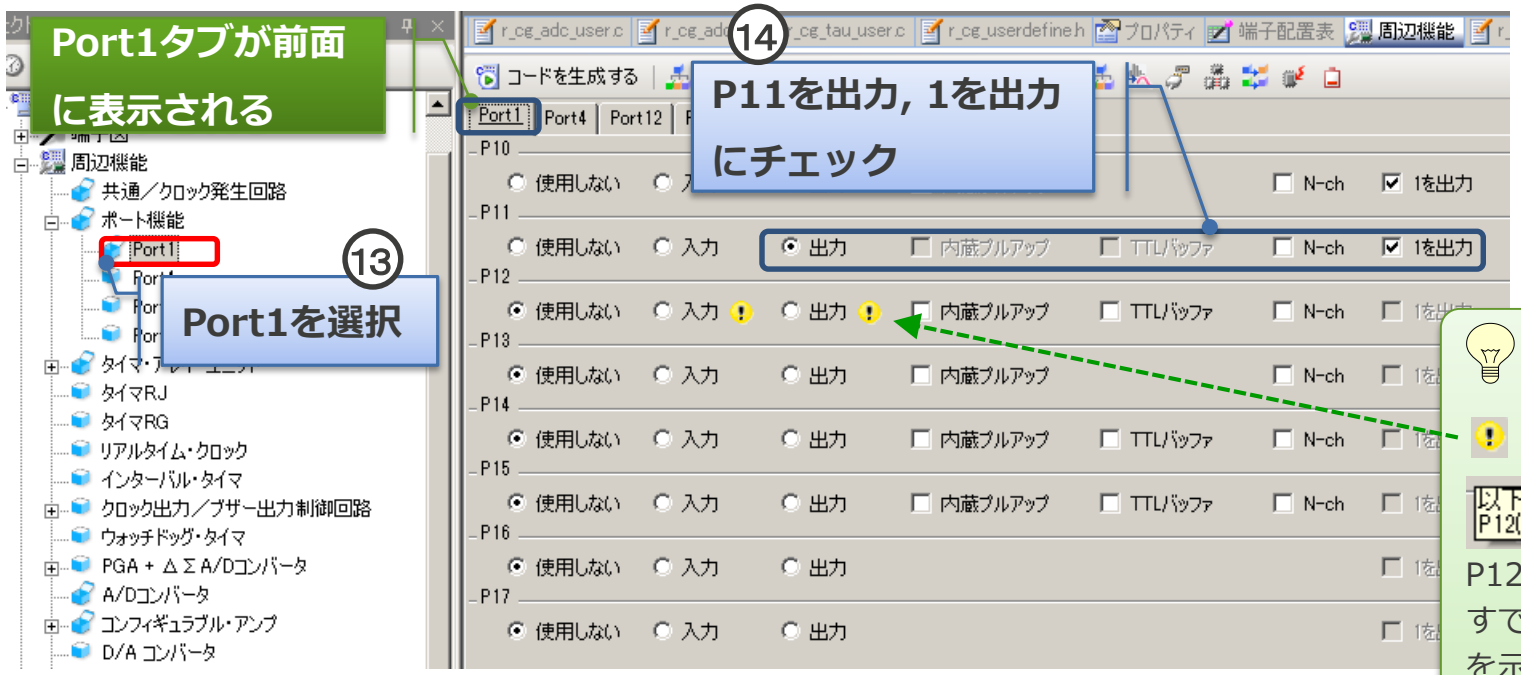


下へスクロール

⑫
+入力を"ANX2"
-入力を"ANX3"
それぞれチェックする



周辺機能設定 (ポート)

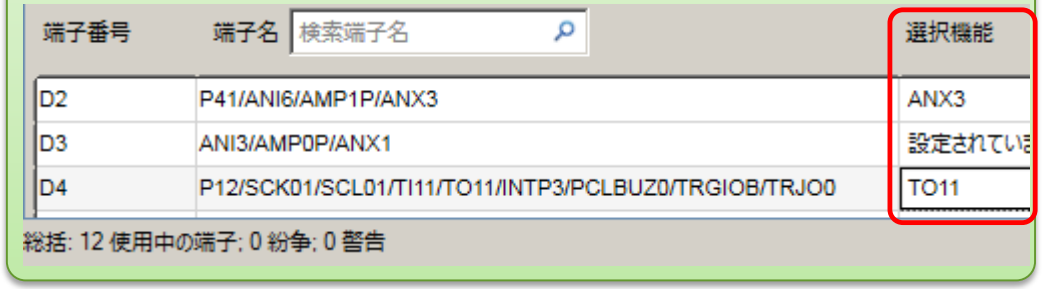


ワンポイント

にマウスカーソルを合わせると下記の文言が表示されます。

以下の端子と競合しています。この機能を使用する場合は競合する機能の設定を無効にしてください。
P12はTO11で使われています。

P12は兼用端子として多くの機能が割り当てられています。今は、すでにTO11機能が使われているため ”P12” としては使えないことを示しています。



プログラム追記部分
赤字は演習Bとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _cg_main.c)

```
48  /***********************************************************
49  Global variables and functions
50  ***********************************************************/
51  /* Start user code for global. Do not edit comment generated here */
52
53  uint16_t gAdResult;         // AD変換結果
54  uint16_t gAdResult2;        // AD変換結果2
55  uint16_t gAdResult3;        // AD変換結果3
56  uint8_t  gAdSelect;         // 変換するA/Dチャネル 0 = ANI0, 1= ANI1
57  uint16_t gTouchCount;       // タッチパネルに触れていると判断した数
58  uint16_t gNoTocuhCount;     // タッチパネルに触ってない回数
59  uint8_t  gTouchSeparate;    // タッチパネルに触っている状態 0=OFF, 1=ON
60  uint16_t  gTmCounter        // タイマカウンタ
61
62  /* End user code. Do not edit comment generated here */
63
64  void R_MAIN_UserInit(void);
65  /***********************************************************
66  * Function Name: main
67  * Description  : This function implements main function.
68  * Arguments    : None
69  * Return Value : None
70  ***********************************************************/
71  void main(void)
72  {
73      R_MAIN_UserInit();
74      /* Start user code. Do not edit comment generated here */
75      while (1U)
76      {
77          ;
78      }
79      /* End user code. Do not edit comment generated here */
80  }
```

```
81  /***********************************************************
82  * Function Name: R_MAIN_UserInit
83  * Description  : This function adds user code before implementing main function.
84  * Arguments    : None
85  * Return Value : None
86  ***********************************************************/
87  void R_MAIN_UserInit(void)
88  {
89      /* Start user code. Do not edit comment generated here */
90
91      R_TAU0_Channel0_Start(); // タイマチャネル0開始
92      gAdSelect = 0;
93      gTouchCount = 0;
94      gNoTocuhCount =0;
95      gTmCounter = 0;
96      R_CAMP2_Start();
97      EI();
98
99  /* End user code. Do not edit comment generated here */
100 }
```

アンプは最初から使えるようにする

プログラム追記部分
赤字は演習Bとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _tau_user.c)

```
29  /***********************************************************
30  Includes
31  ***********************************************************/
32  #include "r_cg_macrodriver.h"
33  #include "r_cg_tau.h"
34  /* Start user code for include. Do not edit comment generated here */
35
36  #include "r_cg_adc.h”
37
38  /* End user code. Do not edit comment generated here */
39  #include "r_cg_userdefine.h“
40
41  /***********************************************************
42  Pragma directive
43  ***********************************************************/
44  #pragma interrupt r_tau0_channel0_interrupt(vect=INTTM00)
45  #pragma interrupt r_tau1_channel0_interrupt(vect=INTTM10)
46  #pragma interrupt r_tau1_channel1_interrupt(vect=INTTM11)
47  /* Start user code for pragma. Do not edit comment generated here */
48
49  extern uint8_t  gAdSelect;
50  extern uint16_t gTmCounter;
51
52  /* End user code. Do not edit comment generated here */
53
54  /***********************************************************
55  Global variables and functions
56  ***********************************************************/
57  /* Start user code for global. Do not edit comment generated here */
58  /* End user code. Do not edit comment generated here */
```

```
60  /***********************************************************
61  * Function Name: r_tau0_channel0_interrupt
62  * Description  : This function INTTM00 interrupt service routine.
63  * Arguments    : None
64  * Return Value : None
65  ***********************************************************/
66  static void __near r_tau0_channel0_interrupt(void)
67  {
68      /* Start user code. Do not edit comment generated here */
69      ad_channel_t ch;
70      switch ( gAdSelect )
71      {
72          case 0:
73              ch = ADCHANNEL0;
74              break;
75
76          case 1:
77              ch = ADCHANNEL1;
78              break;
79
80          case 2:
81              ch = ADCHANNEL2;
82              break;
83
84          default:
85              ch = ADCHANNEL0;
86              break;
87      }
88      R_ADC_Set_ADChannel( ch );
89      R_ADC_Start();
90      gTmCounter++;
91  /* End user code. Do not edit comment generated here */
92  }
```

チャネル番号2の変換結果
処理を追加

プログラム追記部分
赤字は演習Bとの差分

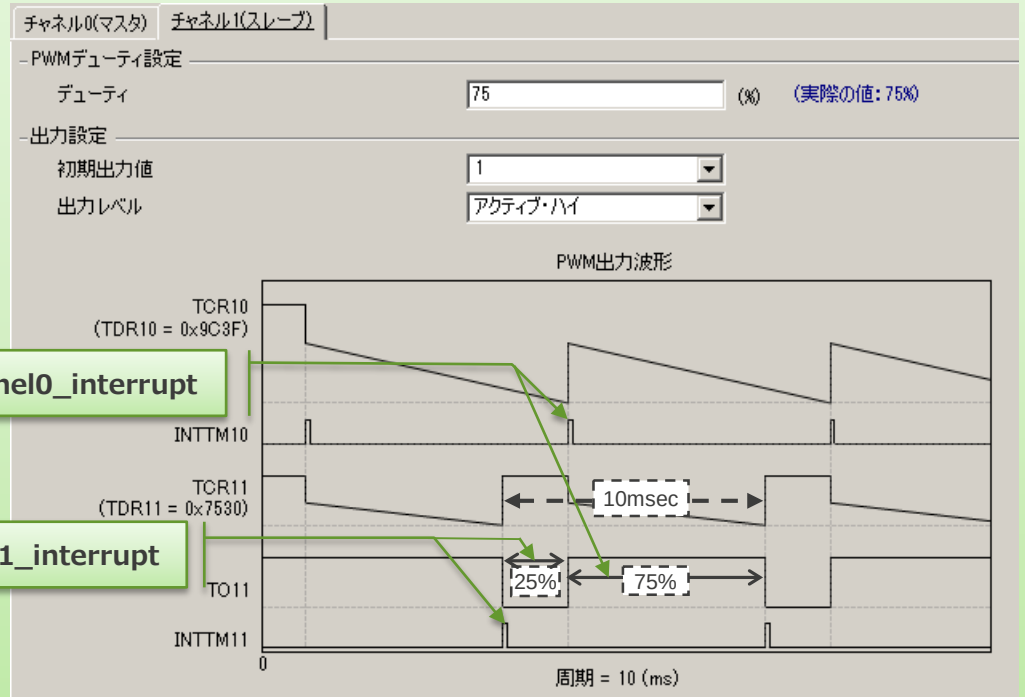
A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _tau_user.c)

```
95  /***********************************************************
96  * Function Name: r_tau1_channel0_interrupt
97  * Description  : This function INTTM10 interrupt service routine.
98  * Arguments    : None
99  * Return Value : None
100 ***********************************************************/
101 static void __near r_tau1_channel0_interrupt(void)
102 {
103     /* Start user code. Do not edit comment generated here */
104
105     // LED P11点灯
106     P1_bit.no1 = 0;
107
108     /* End user code. Do not edit comment generated here */
109 }
110 /***********************************************************
111 * Function Name: r_tau1_channel1_interrupt
112 * Description  : This function INTTM11 interrupt service routine.
113 * Arguments    : None
114 * Return Value : None
115 ***********************************************************/
116 static void __near r_tau1_channel1_interrupt(void)
117 {
118     /* Start user code. Do not edit comment generated here */
119
120     // LED P11消灯
121     P1_bit.no1 = 1;
122
123 /* End user code. Do not edit comment generated here */
124 }
125
126 /* Start user code for adding. Do not edit comment generated here */
127 /* End user code. Do not edit comment generated here */
```

ワンポイント

コード生成で10msecのPWMを作成しました。そのPWMに対する割り込みは2度発生します。下図はデューティ比75%の場合です。75%がLED点灯時間、25%がLED消灯時間となります。プログラム実行中にデューティ比を変更してLEDの輝度を変えています。



プログラム追記部分
赤字は演習Bとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _adc_user.c)

```
20  /***********************************************************
21  * File Name    : r_cg_adc_user.c
22  * Version      : Code Generator for RL78/I1E V1.02.00.06 [12 Aug 2015]
23  * Device(s)    : R5F11CCC
24  * Tool-Chain   : CCRL
25  * Description  : This file implements device driver for ADC module.
26  * Creation Date: 2016/04/08
27  ***********************************************************/
28
29  /***********************************************************
30  Includes
31  ***********************************************************/
32  #include "r_cg_macrodriver.h"
33  #include "r_cg_adc.h"
34  /* Start user code for include. Do not edit comment generated here */
35  #include "r_cg_tau.h"
36  /* End user code. Do not edit comment generated here */
37  #include "r_cg_userdefine.h"
 :
46  /***********************************************************
47  Global variables and functions
48  ***********************************************************/
49  /* Start user code for global. Do not edit comment generated here */
50  extern uint16_t gAdResult;
51  extern uint16_t gAdResult2;
52  extern uint16_t gAdResult3;
53  extern uint8_t  gAdSelect;
54  extern uint16_t gTouchCount;
55  extern uint16_t gNoTocuhCount;
56  extern uint8_t gTouchSeparate;
57  extern uint16_t gTmCounter;
58  /* End user code. Do not edit comment generated here */
```

```
60  /***********************************************************
61  * Function Name: r_adc_interrupt
62  * Description  : None
63  * Arguments    : None
64  * Return Value : None
65  ***********************************************************/
66  static void __near r_adc_interrupt(void)
67  {
68   /* Start user code. Do not edit comment generated here */
69      uint16_t result;
70
71      R_ADC_Stop();
72      R_ADC_Get_Result( &result );
73      switch ( gAdSelect )
74      {
75          case 0:
76              gAdResult = result;
77              break;
78
79          case 1:
80              gAdResult2 = result;
81
82              if ( result > D_TOUCH_AD)
83              {
84                  gTouchCount++;
85                  gNoTocuhCount = 0;
86              }
87              else if ( result > D_NOTOUCH_ADMIN )
88               {
89                  gNoTocuhCount++;
90              }
```

次ページへ

プログラム追記部分
赤字は演習Bとの差分

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _adc_user.c)

前ページから

```
92          if ( gNoTocuhCount > D_NOTOUCH_COUNT )
93          {
94              gTouchCount = 0;
95              gTouchSeparate = 0;
96          }
97
98          if ( (gTouchCount > D_TOUCH_COUNT) &&( gTouchSeparate == 0) )
99          {
100             // LED点灯と消灯を繰り返す
101             P1_bit.no0 ^= 1;
102             gTouchSeparate = 1;
103         }
104         break;
105
106     case 2:
107         gAdResult3 = result;
108         if ( gAdResult3 > D_MIC_ADTYP )
109         {
110             comp = ( gAdResult3 - D_MIC_ADTYP )*1000;
111             if ( comp > D_MIC_LEDMAXPWM )
112             {
113                 comp = D_MIC_LEDMAXPWM;
114             }
115             if ( ( gTmCounter > D_MIC_LEDCHANGE ) | ( comp > TDR11 ) )
116             {   // LEDの明るさを100msecごとに変更する
117                 TDR11 = comp;
118                 gTmCounter = 0;
119             }
120         }
```

```
121             else
122             {
123                 if ( gTmCounter > D_MIC_LEDOFFDEALY )
124                 {
125                     TDR11 = 0;
126                     gTmCounter = 0;
127                 }
128             }
129             break;
130
131         default:
132             gAdResult = result;
133             break;
134     }
135
136     gAdSelect++;
137     if ( gAdSelect >= D_MAXAD_CHANNEL )
138     {
139         gAdSelect = 0;
140     }
141
142     /* End user code. Do not edit comment generated here */
143 }
144
145 /* Start user code for adding. Do not edit comment generated here */
146 /* End user code. Do not edit comment generated here */
```

プログラム追記部分
赤字は演習Bとの差分

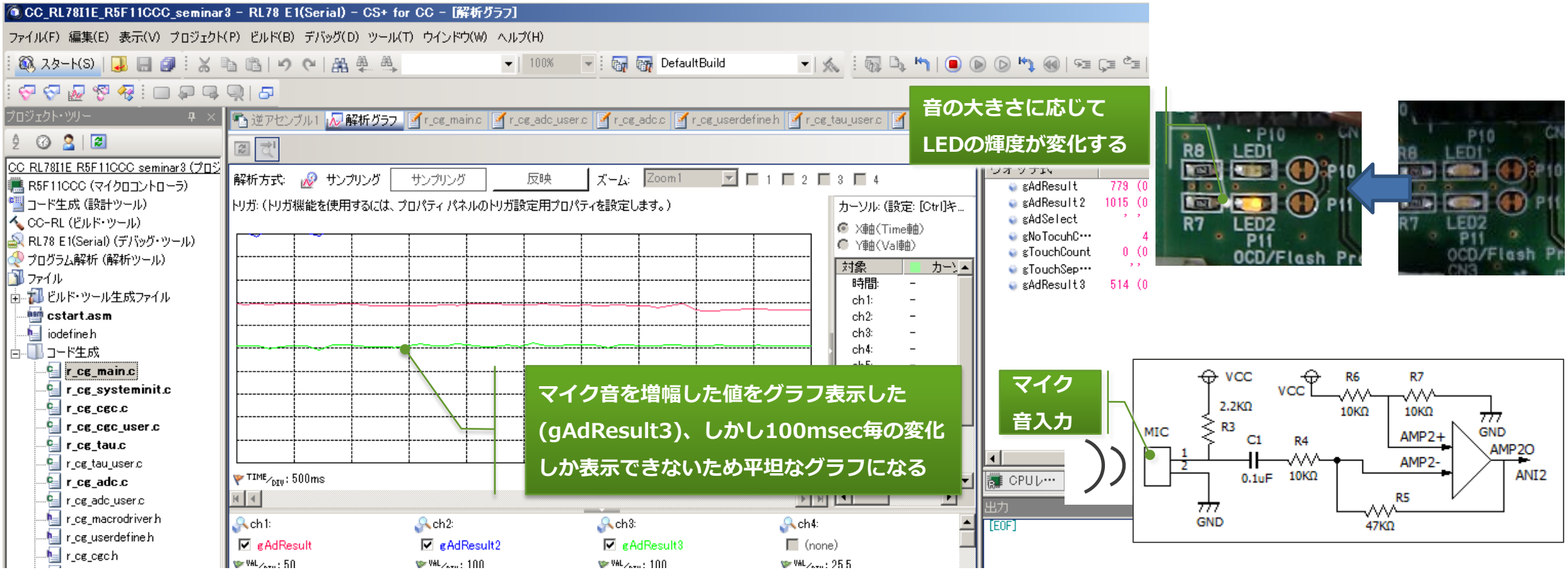
A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

コード生成とプログラム編集 (r _cg_userdefine.h)

```
20  /***********************************************************
21  * File Name    : r_cg_userdefine.h
22  * Version      : Code Generator for RL78/I1E V1.02.00.06 [12 Aug 2015]
23  * Device(s)    : R5F11CCC
24  * Tool-Chain   : CCRL
25  * Description  : This file includes user definition.
26  * Creation Date: 2016/04/08
27  ***********************************************************/
28  #ifndef _USER_DEF_H
29  #define _USER_DEF_H
30
31  /***********************************************************
32  User definitions
33  ***********************************************************/
34
35  /* Start user code for function. Do not edit comment generated here */
36
37  #define D_MAX_AD_CHANNEL        3           // A/D変換するチャネル数
38
39  #define D_TOUCH_AD              100         // タッチパッドに触ったと判断する値
40  #define D_NOTOUCH_ADMIN         900         // タッチパッド非接触時を判断するA/D最小値
41  #define D_TOUCH_COUNT           10          // タッチパッドを触ったと判断する回数
42  #define D_NOTOUCH_COUNT         100         // タッチパッドを触っていないと判断する回数
43
44  #define D_MIC_ADTYP             550         // マイクA/D入力の無音状態値
45  #define D_MIC_LEDOFFDEALY       500         // LEDを消去するまでの時間 msec
46  #define D_MIC_LEDCHANGE         100         // LEDの輝度を変えるタイミング msec
47  #define D_MIC_LEDMAXPWM         40000       // LEDの輝度を変えるPWMの最大値
48
49  /* End user code. Do not edit comment generated here */
50  #endif
```

A. 照度センサを使って明るさを検知
B. 簡易タッチセンサでLEDをON/OFF
C. マイクを使って音でLEDの輝度を変える

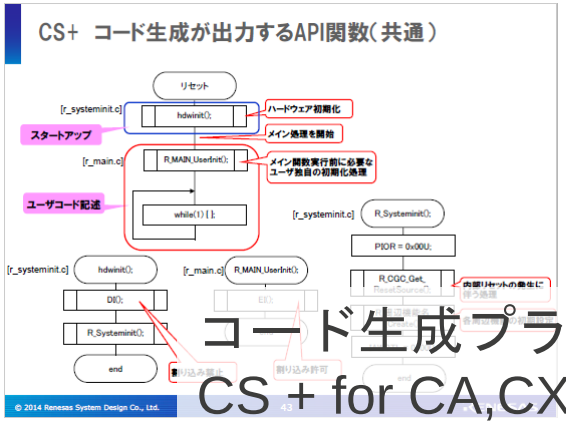
プログラムの実行とデバッグ (解析グラフ表示)



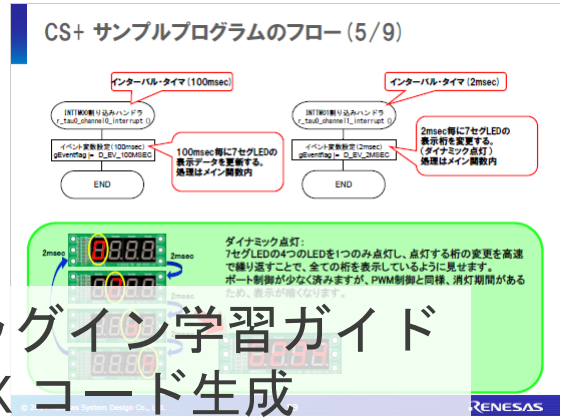
最後に

他にもコード生成のガイドがあるのでご活用ください。

https://www.renesas.com/ja-jp/products/software-tools/tools/code-generator/code-generator-plug-in.html



コード生成プラグイン学習ガイド
CS + for CA,CX コード生成
RL78/G13



RL78/I1Eコード生成　脈拍センサデモ
コンフィギュラブル・アンプ使用例
CS + for CA,CX コード生成
e² studio コード生成