



# SmartRA 学習キット チュートリアル 1

---

ルネサス エレクトロニクス社 RA マイコン搭載  
HSB シリーズマイコンボード 評価キット

-本書を必ずよく読み、ご理解された上でご利用ください

株式会社 **北斗電子**  
REV.1.0.0.0

—目 次—

注意事項 .....	1
安全上のご注意 .....	2
本書で説明する内容.....	4
マイコンと組み込みプログラム .....	5
1. RA2L1_SMARTKIT_TUTORIAL0 .....	9
取扱説明書改定記録 .....	16
お問合せ窓口 .....	16

## 注意事項

本書を必ずよく読み、ご理解された上でご利用ください

### 【ご利用にあたって】

1. 本製品をご利用になる前には必ず取扱説明書をよく読んで下さい。また、本書は必ず保管し、使用上不明な点がある場合は再読み、よく理解して使用して下さい。
2. 本書は株式会社北斗電子製マイコンボードの使用方法について説明するものであり、ユーザシステムは対象ではありません。
3. 本書及び製品は著作権及び工業所有権によって保護されており、全ての権利は弊社に帰属します。本書の無断複製・複製・転載はできません。
4. 弊社のマイコンボードの仕様は全て使用しているマイコンの仕様に準じております。マイコンの仕様に関しましては製造元にお問い合わせ下さい。弊社製品のデザイン・機能・仕様は性能や安全性の向上を目的に、予告無しに変更することがあります。また価格を変更する場合や本書の図は実物と異なる場合もありますので、御了承下さい。
5. 本製品のご使用にあたっては、十分に評価の上ご使用下さい。
6. 未実装の部品に関してはサポート対象外です。お客様の責任においてご使用下さい。

### 【限定保証】

1. 弊社は本製品が頒布されているご利用条件に従って製造されたもので、本書に記載された動作を保証致します。
2. 本製品の保証期間は購入戴いた日から1年間です。

### 【保証規定】

**保証期間内でも次のような場合は保証対象外となり有料修理となります**

1. 火災・地震・第三者による行為その他の事故により本製品に不具合が生じた場合
2. お客様の故意・過失・誤用・異常な条件でのご利用で本製品に不具合が生じた場合
3. 本製品及び付属品のご利用方法に起因した損害が発生した場合
4. お客様によって本製品及び付属品へ改造・修理がなされた場合

### 【免責事項】

弊社は特定の目的・用途に関する保証や特許権侵害に対する保証等、本保証条件以外のものは明示・黙示に拘わらず一切の保証は致し兼ねます。また、直接的・間接的損害金もしくは欠陥製品や製品の使用方法に起因する損失金・費用には一切責任を負いません。損害の発生についてあらかじめ知らされていた場合でも保証は致し兼ねます。

ただし、明示的に保証責任または担保責任を負う場合でも、その理由のいかんを問わず、累積的な損害賠償責任は、弊社が受領した対価を上限とします。本製品は「現状」で販売されているものであり、使用に際してはお客様がその結果に一切の責任を負うものとします。弊社は使用または使用不能から生ずる損害に関して一切責任を負いません。

保証は最初の購入者であるお客様ご本人にのみ適用され、お客様が転売された第三者には適用されません。よって転売による第三者またはその為になすお客様からのいかなる請求についても責任を負いません。

本製品を使った二次製品の保証は致し兼ねます。

## 安全上のご注意

製品を安全にお使いいただくための項目を次のように記載しています。絵表示の意味をよく理解した上でお読み下さい。

### 表記の意味



取扱を誤った場合、人が死亡または重傷を負う危険が切迫して生じる可能性がある事が想定される



取扱を誤った場合、人が軽傷を負う可能性又は、物的損害のみを引き起こすが可能性がある事が想定される

### 絵記号の意味

	<b>一般指示</b> 使用者に対して指示に基づく行為を強制するものを示します		<b>一般禁止</b> 一般的な禁止事項を示します
	<b>電源プラグを抜く</b> 使用者に対して電源プラグをコンセントから抜くように指示します		<b>一般注意</b> 一般的な注意を示しています

## 警告



以下の警告に反する操作をされた場合、本製品及びユーザシステムの破壊・発煙・発火の危険があります。マイコン内蔵プログラムを破壊する場合があります。

1. 本製品及びユーザシステムに電源が入ったままケーブルの抜き差しを行わないでください。
2. 本製品及びユーザシステムに電源が入ったままで、ユーザシステム上に実装されたマイコンまたはIC等の抜き差しを行わないでください。
3. 本製品及びユーザシステムは規定の電圧範囲でご利用ください。
4. 本製品及びユーザシステムは、コネクタのピン番号及びユーザシステム上のマイコンとの接続を確認の上正しく扱ってください。



発煙・異音・異臭にお気づきの際はすぐに使用を中止してください。

電源がある場合は電源を切って、コンセントから電源プラグを抜いてください。そのままご使用すると火災や感電の原因になります。

# 注意



以下のことをされると故障の原因となる場合があります。

1. 静電気が流れ、部品が破壊される恐れがありますので、ボード製品のコネクタ部分や部品面には直接手を触れないでください。
2. 次の様な場所での使用、保管をしないでください。  
ホコリが多い場所、長時間直射日光が当たる場所、不安定な場所、衝撃や振動が加わる場所、落下の可能性がある場所、水分や湿気の多い場所、磁気を発するものの近く
3. 落としたり、衝撃を与えたり、重いものを乗せないでください。
4. 製品の上に水などの液体や、クリップなどの金属を置かないでください。
5. 製品の傍で飲食や喫煙をしないでください。



ボード製品では、裏面にハンダ付けの跡があり、尖っている場合があります。

取り付け、取り外しの際は製品の両端を持ってください。裏面のハンダ付け跡で、誤って手など怪我をする場合があります。



CD メディア、フロッピーディスク付属の製品では、故障に備えてバックアップ（複製）をお取りください。

製品をご使用中にデータなどが消失した場合、データなどの保証は一切致しかねます。



アクセスランプがある製品では、アクセスランプの点灯中に電源を切ったり、パソコンをリセットをしないでください。

製品の故障や、データ消失の原因となります。



本製品は、医療、航空宇宙、原子力、輸送などの人命に関わる機器やシステム及び高度な信頼性を必要とする設備や機器などに用いられる事を目的として、設計及び製造されておりません。

医療、航空宇宙、原子力、輸送などの設備や機器、システムなどに本製品を使用され、本製品の故障により、人身や火災事故、社会的な損害などが生じても、弊社では責任を負いかねます。お客様ご自身にて対策を期されるようご注意ください。

## 本書で説明する内容

本書は、最初にマイコンボードのプログラムを行う方を対象にしています。

マイコンチップ自体やボードに搭載されている機能を動かすためにはどうすればよいかを、順を追って解説するマニュアルとなります。

章毎に、解説している内容が異なりますので、順番にお読み頂く必要はありません。興味のある章、使用したい機能を解説している章、チュートリアルを動かしてみてください。

# マイコンと組み込みプログラム

## マイコンとは？

本製品は、マイコンを搭載したボードです。マイコンとは一体何者なのでしょう？

マイコンボードを見ると、黒くて足が出ている部品がありますが、それがマイコンです。

よく見ると、黒くて足が出ている部品は、実はたくさんあります。そのうち、ボードの中央部に配置されており、4方向に細かな足が出ている正方形の部品がマイコンです。

マイコンは、一般的にマイコンと呼ばれていますが、正しくは「マイクロコントローラ」の略です。「小さくて、何かを制御するもの」という意味でしょうか。

マイコンに良く似たものとして、CPU(Central Processing Unit)があります。一般的にパソコンに入っている、銀色の金属製のふたが付いていて、裏面には端子が格子状に並んでいる部品です(2021年現在)。

また、最近のスマートフォンに搭載されているプロセッサチップの事を、SoC(System On Chip)などと呼ぶこともありますが、基本的にこれら、マイコンとCPU、SoCは本質的には何も変わらないと思います。敢えて異なる点を挙げるとすると、「得意分野が異なる」といったところでしょうか。

マイコンは外部の機器を制御するという目的に適しています。

例えば、身近に存在するさまざまな機器(掃除機、洗濯機等の家電や、自動車や航空機等の乗り物、身近にある大多数の機器と言っても良いでしょう)に、マイコンは使用されています。

## 組み込みプログラムとは？

プログラムというのは、コンピュータにとっての予定表のようなもので、「行うこと」が「行う順番」に記載されているものです。コンピュータは、プログラムに沿って動作を行います。

マイコンもコンピュータ(計算機)の一種ですので、プログラムに従った動作を行うという点は変わりません。

プログラムはパソコンやスマートフォンを動かしたり、至る所で使われていますが、マイコン向けのプログラムを「組み込みプログラム」と呼ぶことがあります。「組み込みプログラム」というと、何か通常のプログラムと別なものになるのでしょうか？

パソコン上で動いているアプリケーションプログラムと組み込みプログラムは、本質的な違いはないと思います。どちらもコンピュータ上で実行する命令を書き下したものです。

組み込みプログラムは、機器に組み込まれたプログラムという意味で、機器(ハードウェア)に近い立ち位置のプログラム(ソフトウェア)となるかと思います。外から見ると、ハードウェアの一部に見えるかも知れません。

パソコンやスマートフォンで動作するプログラムに慣れた方でも、マイコン向けの組み込みプログラムというと、ちょっと(かなり?)勝手が違うのは事実で、「組み込み」というキーワードが付いただけで、異世界の存在に感じてしまう方も少なくないかも知れません。

本キットは、マイコンや組み込みプログラムが身近ではないという方をターゲットにしていますので、あまり身構えずに、マイコンを自由に動かす事を楽しんで頂ければと思います。

## マイコンのプログラムの特徴って？

マイコン向けのプログラム(組み込みプログラム)は、何か特別な点があるわけではありませんが、いくつか特徴のようなものはあるかも知れません。

・ハードウェアとソフトウェアを切り離して考える事ができない

マイコンは、ハードウェアを制御する目的で使用されることが多いため、ソフトはソフト、ハードはハードのような切り分けができないのかと思います。

パソコンのアプリケーションプログラム開発のときに、マザーボードや CPU の回路図を見ながらというケースはまず無いと思いますが、マイコンのプログラムでは、ボードの回路図を見ながら行うというのが普通です。

要は回路図が読めないとプログラムが書けないといった側面があるのです。

・マイコンの中にタイマや通信モジュール等のハードウェアが備わっており、それらを制御する必要がある

マイコンの中には、種々のモジュールが内蔵されており、マイコンを使いこなすためにはそれらのモジュールを制御する必要があります。

この時は、マイコンの「ハードウェアマニュアル」という資料を読む事になります。文字通り、「ハードウェア」について記載されているマニュアルですので、純粋なソフト屋さんにはピンと来ない点があるかも知れません。

- ・メモリにデータを書き込むとランプが点灯する？

マイコンには特定のアドレスにデータを書き込むと、端子(マイコンの足)の電圧が0→3.3Vに変化したりします。

通常メモリにデータを書き込むといった処理は、メモリ中のデータが書き換わるというイメージですが、マイコンのハードウェアを制御する場合もメモリのアクセスと同じような手法となります。

- ・プログラムを間違えると全体が止まってしまう

パソコンのアプリケーションですと、プログラムに致命的なエラーがあった場合、「ハードディスクのブートセクタを壊す」「他のアプリケーションやOSの使用しているメモリ領域を壊す」といった様な処理はOSがブロックしてくれたりもします。OSの動作を含め完全に止まってしまうケースはそう多くないかと思います。

それに対し、組み込み系のプログラムでは、暴走時の保護機構等がありますが、基本的にはプログラムで「何でも」できてしまいます。ブートセクタに相当する領域を上書きする事も容易にできます。場合によっては、縛りがなく自由という考え方もできるかと思います。

- ・プログラム全体を手の内に

パソコンのアプリケーション作成時は、OSやAPIが提供する機能を使わずにプログラムを行うという事は現実的ではないかと思います。自分以外の方が書いたコードと協調して動かす事が少なからず求められるかと思います。

それに対し、組み込みプログラムは、本当の意味での「フルスクラッチ」「ゼロベース」で、プログラミングできるということが言えるかと思います。

- ・メモリが潤沢には使用できない

SmartRA 学習キットの搭載マイコンのメモリ(RAM)は、32KBです。プログラムを格納するコード領域(ROM)は、256KBです。

一昔前の組み込み環境から見ると、RAMもROMも潤沢にあり、メモリを節約したプログラムコードを書く事に専念しなくても良いレベルですが、パソコンやスマートフォンから見るとかなり少ないですので、この点は、節約の意識が必要でしょうか。

・処理速度がコントロールできる

パソコンやスマートフォン用のアプリは、搭載 CPU のクロック速度もまちまちで、他のタスクとの関係で、自分のアプリケーションにどの程度の CPU リソースが割り振られるかも決まっています。

それに対し、組み込み系では、クロック周波数を(ある程度)プログラマが指定できますので、命令の実行速度はプログラマの手の内にあります。また、外からの影響のないプログラムであれば、一連のルーチンの処理速度はほぼ決まるので、1 秒以内に必ず処理を終わらせたいといった要求に応える事ができます。

・開発言語は？

組み込みのプログラムで使用されるのは、

C 言語(C++)

アセンブリ言語

が主です。もちろん、他の言語向けのコンパイラやインタプリタを用意しているマイコンもありますが、基本はこの 2 種の言語となると思います。本チュートリアルでは、C 言語をベースにします。

# 1. RA2L1\_SMARTKIT\_TUTORIAL0

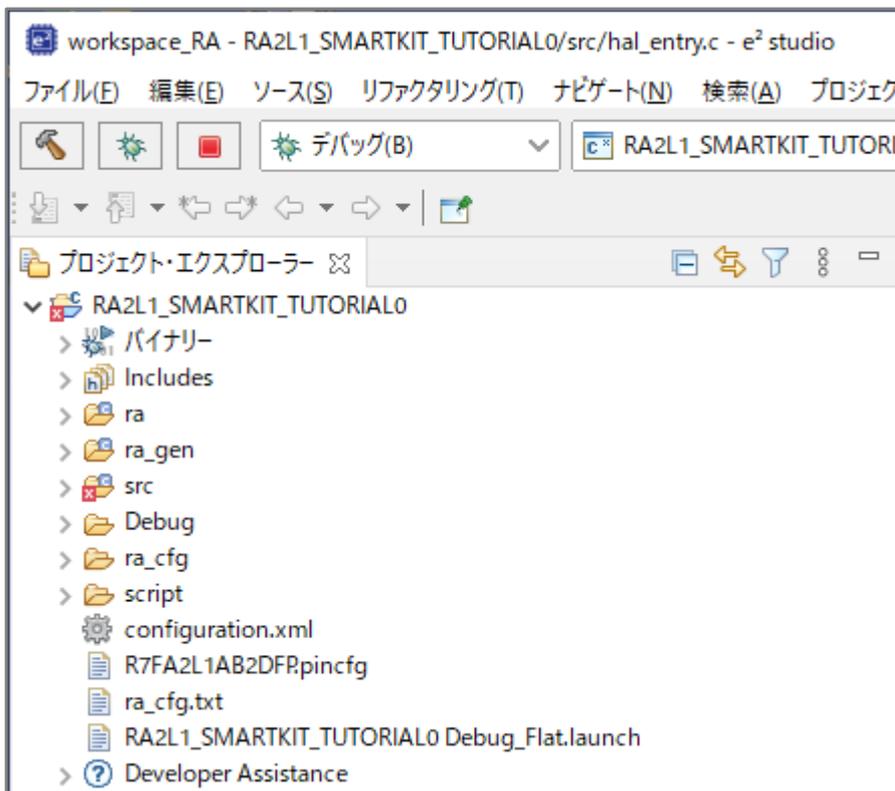
最初のチュートリアルとなります。

CD 内の、TUTORIAL\RA2L1\_SMARTKIT\_TUTORIAL0.zip

を、e2studio の環境にインポートしてください。インポートの手順は、スタートアップマニュアルにハードコピー付きで詳細に記載がありますので、そちらを参照してください。

インポート後、RA2L1\_SMARTKIT\_TUTORIAL0 プロジェクトを開いた状態としてください。

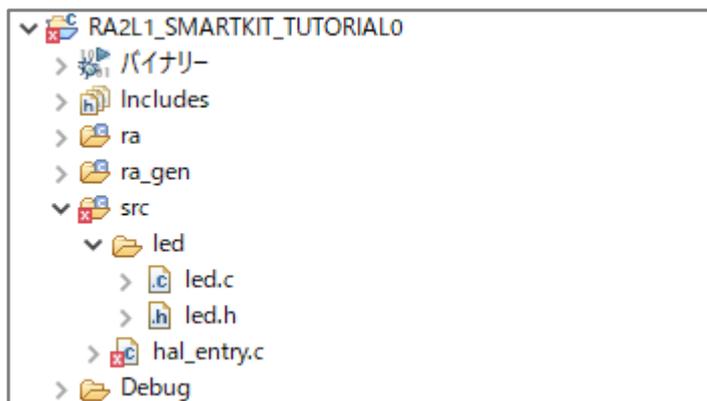
e2studio を使うのが初めてという方は、スタートアップマニュアルに、e2studio の基本的な使い方等を付録にまとめていますので、目を通してみてください。



当該プロジェクトを開くと、プロジェクト・エクスプローラ上では、上記の様なツリーとなります。「>」のアイコンは展開可能(そのフォルダの下にファイル等がある)事を示しています。

上記の中で、基本的にユーザが編集するのは、src 以下です。それ以外のフォルダは、ツールが生成したり使用するものです。

src フォルダを展開すると、次のようになります。



このプロジェクトでは、src の下に

led		サブフォルダ
	led.c	LED 制御プログラムソース
	led.h	LED 制御プログラムヘッダ
hal_entry.c		メイン処理

となっています。led のフォルダ以下はこちらで追加したものです。hal\_entry.c は空のプロジェクトを作成した段階で生成されるファイルです。

led 以下のファイルは、次のチュートリアル LED\_SW で説明します。

hal\_entry.c をダブルクリックすると、右側のペイン(表示領域)に、ファイルの中身が表示されます。

```

hal_entry.c
1  #include "hal_data.h"
2
3  #include "led/led.h"
4
5  FSP_CPP_HEADER
6  void R_BSP_WarmStart(bsp_warm_start_event_t event);
7  FSP_CPP_FOOTER
8
9  * main() is generated by the RA Configuration editor and is used to generate threads if an RTOS is used. This function
10
11 void hal_entry(void)
12 {
13     /* TODO: add your own code here */
14     unsigned char i;
15
16     while(1)
17     {
18         //LEDを
19
20         //全消灯
21         //1つON
22         //2つON
23         //...
24         //8つON
25         //先頭に戻る
26
27         //の動作を繰り返します
28
29         for(i=0; i<9; i++)
30         {
31             //指定された個数のLEDを点灯
32             led_num_on(i);
33
34             //0.5秒(500ms)待つ
35             R_BSP_SoftwareDelay(500U, BSP_DELAY_UNITS_MILLISECONDS);
36         }
37     }
38 }
39
40

```

ここで、枠線で囲んだ部分が追加した部分です。その他の部分は、プロジェクトを作成した時点で生成されたものです。

### void hal\_entry(void)

この、hal\_entry という関数が、いわゆるメイン関数 main()に相当します。

正しくは、別途 main()は定義されていますが、ユーザ向けのプログラムのエントリーポイント(実行開始)が、この hal\_entry()という関数で、この部分にユーザプログラムを書くとお考えください。

ここでは、

```

unsigned char i;

while(1)
{
    for(i=0; i<9; i++)
    {
        //指定された個数のLEDを点灯
        led_num_on(i);

        //0.5秒(500ms)待つ
        R_BSP_SoftwareDelay(500U, BSP_DELAY_UNITS_MILLISECONDS);
    }
}

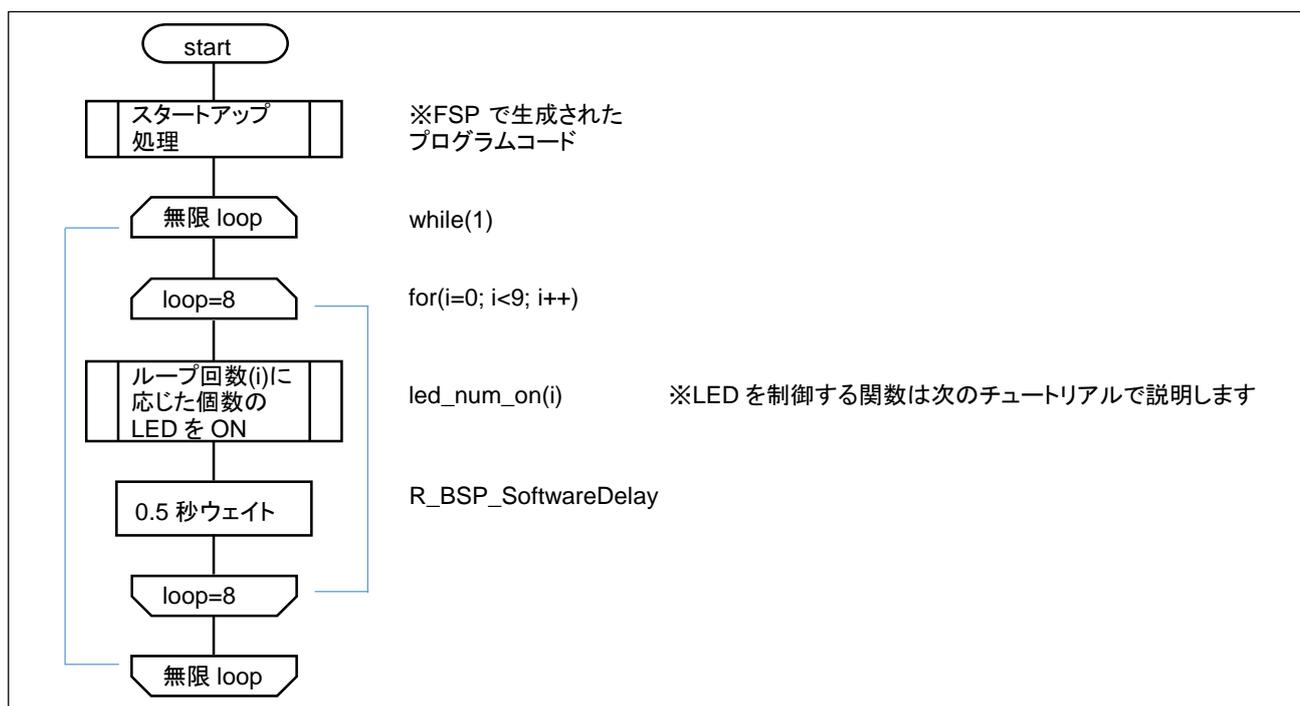
```

コメント一部省略

- ・while(1){ }で囲まれた部分は無限ループ
- ・for 文は、0~8 の 8 回ループします。LED を 0 個 (全消灯)、1 個、2 個と点灯させていき、8 個点灯すると、for 文を抜けます
- ・そうすると、while ループで再度同じ事を繰り返します
- ・R\_BSP\_SoftwareDelay は、LED の変化を目で見て判るようにウェイトを掛けています

プログラムの流れとしては上記となります。

#### フローチャート



ここで、本チュートリアルは意図的に構文エラーを入れ込んでいます。

- ・プロジェクトのインポート
- ・プロジェクトを開く
- ・ソースファイルの修正
- ・プロジェクトのビルド
- ・生成した ROM イメージ (sreq ファイル) のボードへの書きこみ

のチュートリアルなので、実際に一連の操作を行って頂きたいので、最初はエラーがある状態となっています。

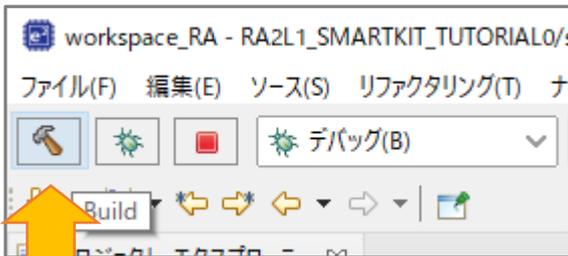
```

33 //指定された個数のLEDを点灯
34 led_num_on(i);
35

```

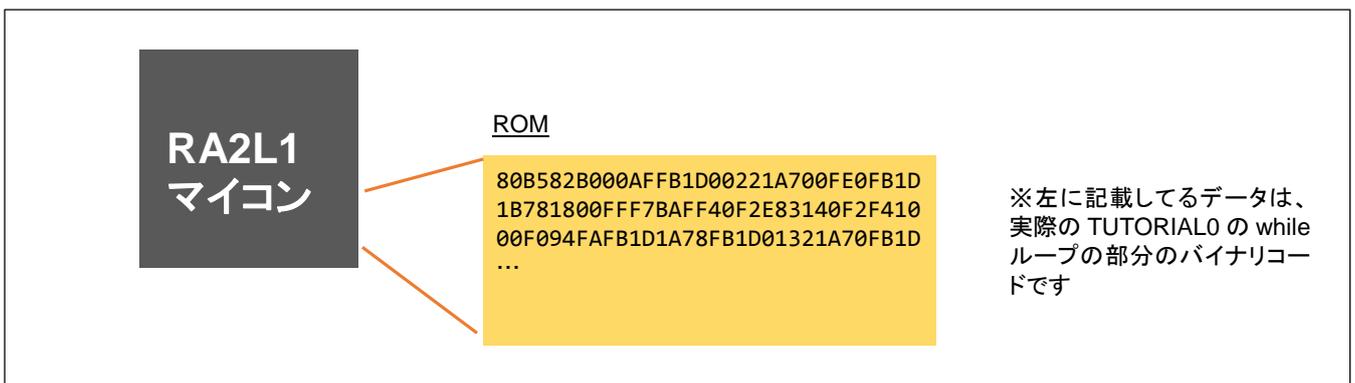
エラー(バグ)がある場所は一目瞭然で、虫(バグ)がで示されている行です。

この行がセミコロン(;)で終わっていないためにエラーとなっています。ソースファイルの修正としては、この行(34 行目)の行末に、;を追加してください(半角のセミコロン;)。



修正後、Build を実行すると、エラーが解消されプログラムのコンパイル、リンクが完了し、マイコンボードに書き込む ROM のイメージファイル(Debug¥RA2L1\_SMARTKIT\_TUTORIAL0.srec)が生成されます。

このファイルは、RX マイコン等では、.mot の拡張子が付いていた(モトローラ S 形式)ファイルです。RA の環境では、拡張子.sreq となっていますが、中身は mot ファイルと同じです。



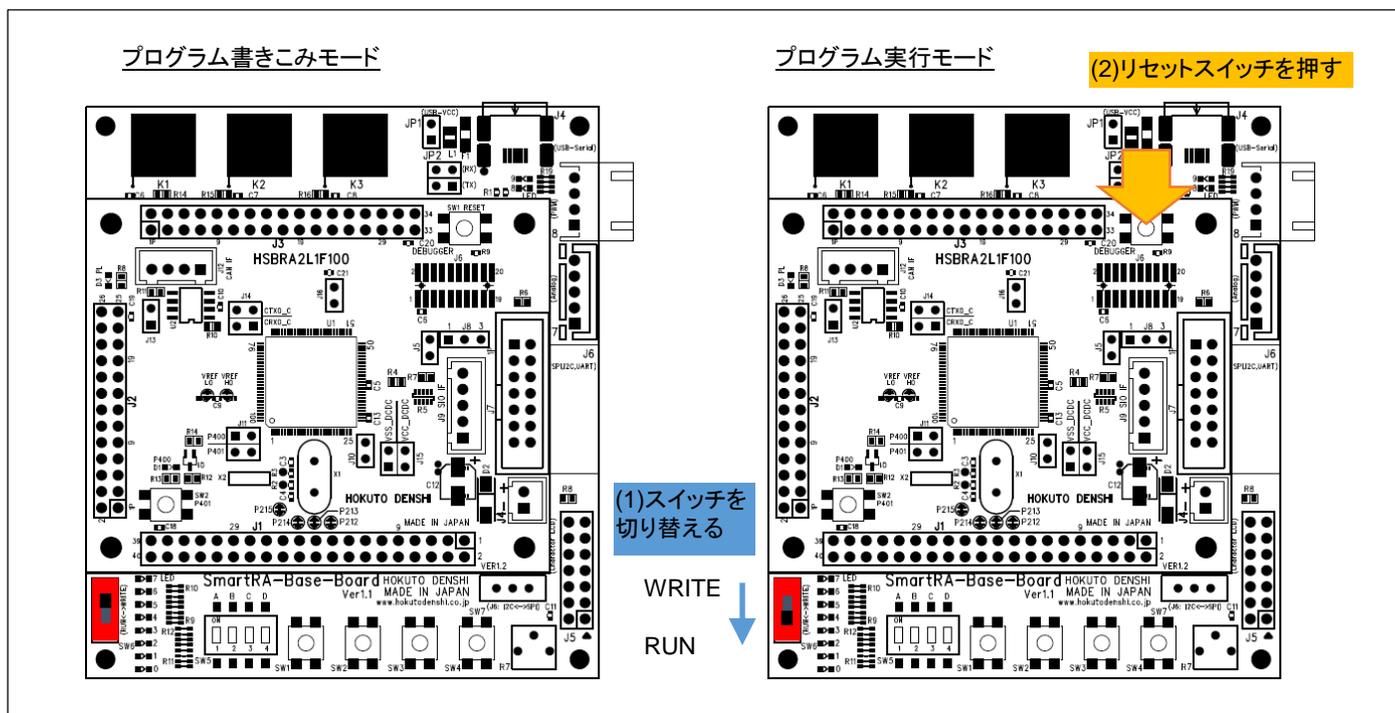
マイコンの中には、ROM(ReadOnlyMemory, 読み出し専用メモリ)があり、マイコンは ROM に格納されているプログラム(正しくはプログラムを機械語に変換したもの)に従い処理を実行します。

RA2L1 マイコンの ROM は、フラッシュメモリとなっているため、(読み出し専用といいつつ)何回でも書き換えることが可能です。

ビルド操作で生成されたの.sreq ファイルをマイコンの ROM に書き込むと、マイコンはプログラムした通りに動きます。

.sreq ファイルをマイコンボードに書込む操作は、スタートアップマニュアルを参照してください。

プログラム(.sreq ファイル)書きこみ後



ボードのスイッチをプログラム実行モード(RUN 側)に変更し、ボードのリセットスイッチを押すと LED が順次点灯する動作となりましたでしょうか。

マイコンボードは、「プログラム書きこみモード」と「プログラム実行モード」があり、それらを切り替えるのが、ベースボード上の SW6(B-SW6)です。モードを切り替えた後は、一度マイコンをリセットしないと設定したモードには切り替わりません。マイコンをリセットするのが、マイコンボード上の SW1(M-SW1)です。

本チュートリアルは、一連の操作に慣れて頂くためのチュートリアルですので、マイコンボードにプログラムを書き込んで動作させる事が出来れば目的は達成しています。

プロジェクトツリーには、色々なファイルがありますが、

hal\_entry.c

が、メインの処理を含むファイルとなります。それ以外にも重要なファイルはいくつかあるのですが、とりあえず他のファイルは「おまじない」とここでは考えてください。

hal\_entry.c[抜粋]

```
unsigned char i;

while(1)
{
    for(i=0; i<9; i++)
    {
        //指定された個数のLEDを点灯
        led_num_on(i);

        //0.5秒(500ms)待つ
        R_BSP_SoftwareDelay(500U, BSP_DELAY_UNITS_MILLISECONDS);
    }
}
```

上記を例えば、

```
while(1)
{
    led_num_on(0);

    //0.5秒(500ms)待つ
    R_BSP_SoftwareDelay(500U, BSP_DELAY_UNITS_MILLISECONDS);

    led_num_on(8);

    //0.5秒(500ms)待つ
    R_BSP_SoftwareDelay(500U, BSP_DELAY_UNITS_MILLISECONDS);
}
```

の様に変更すれば、8個のLEDが点滅する動作となるはずですが。プログラムソースを変更、ビルド、書きこみを行いLEDの点灯のパターンが変わる(制御できている)事を確かめてみてください。

本チュートリアルは、「ソースコードの変更」「プログラムのビルド」「マイコンボードへの書きこみ」「プログラムの実行」を行い、LEDの点灯のパターンの制御が実際に行える事を理解するという所までとなります。実際に、どのようなプログラムを書けばLEDが制御できるかは、次のチュートリアルとなります。

## 取扱説明書改定記録

バージョン	発行日	ページ	改定内容
REV.1.0.0.0	2021.5.6	—	初版発行

## お問合せ窓口

最新情報については弊社ホームページをご活用ください。

ご不明点は弊社サポート窓口までお問合せください。

株式会社 **北斗電子**

〒060-0042 札幌市中央区大通西 16 丁目 3 番地 7

TEL 011-640-8800 FAX 011-640-8801

e-mail: support@hokutodenshi.co.jp (サポート用)、order@hokutodenshi.co.jp (ご注文用)

URL: <http://www.hokutodenshi.co.jp>

商標等の表記について

- ・ 全ての商標及び登録商標はそれぞれの所有者に帰属します。
- ・ パーソナルコンピュータを PC と称します。

---

ルネサス エレクトロニクス RA マイコン搭載  
HSB シリーズマイコンボード 評価キット

# SmartRA 学習キット チュートリアル 1

株式会社 **北斗電子**

©2021 北斗電子 Printed in Japan 2021 年 5 月 6 日改訂 REV.1.0.0.0 (210506)

---