

# 開発環境構築ガイド

## RM-92A/RM-92C

LoRa/FSK/GFSK通信モジュール

**RF** **LINK**

Ver2.8

# 目次

- 1.はじめに
- 2.開発の準備
  - 2.1 開発に必要な環境 ※IAR社のICE(i-jet)を使用する場合
  - 2.2 開発キット同梱内容
  - 2.3 開発環境構築の流れ
  - 2.4 方法1の環境構築例
  - 2.5 方法2の環境構築例
  - 2.6 ドライバのインストール
  - 2.7 シリアル通信ソフトの設定
  - 2.8 FTDI(UART/USB変換IC)の設定
- 3.内蔵FlashROMへの書き込み
  - 3.1 書き込みに必要なもの
  - 3.2 プロジェクトからの書き込み手順
  - 3.3 ビルド済みROMデータの書き込み手順
- 4.統合環境(EWARM)を使用した開発方法
  - 4.1 統合開発環境構築までの流れ
  - 4.2 統合開発環境のインストール
  - 4.3 プロジェクトファイルの展開
  - 4.4 プロジェクトのビルド
  - 4.5 プロジェクトのビルド
  - 4.6 実行モジュールのRM-92A/92Cへのダウンロード
  - 4.7 実行モジュールのRM-92A/92Cへのダウンロード
  - 4.8 デバッグ開始
  - 4.9 プログラムの実行～シリアル通信ソフトによる確認
  - 4.10 SimpleMAC-92X-SDK-std3 (中継版セット)の出荷時の設定
- 5.開発ボードの設定
  - 5.1 RM-900EV Rev.200との接続
  - 5.2 RM-900EV Rev.200 SW2 の設定方法
  - 5.3 RM-900EV Rev.200 J1 の設定方法
  - 5.4 RM-24X/9XX\_EV Rev.310との接続
  - 5.5 RM-24X/9XX\_V Rev.310 SW2 の設定方法
  - 5.6 RM-24X/9XX\_EV Rev.310 J5 の設定方法
  - 5.7 RM-24X/92X Rev.4.0との接続
  - 5.8 RM-24X/92X\_Ver Rev.4.0 SW1 の設定方法
  - 5.9 RM-24X/92X\_EV Rev.4.0 J1 の設定方法
  - 5.10 RM-24X/92X Rev.4.1との接続
  - 5.11 RM-24X/92X\_Ver Rev.4.1 SW1 の設定方法
  - 5.12 RM-24X/92X\_EV Rev.4.1 J1 の設定方法
  - 5.13 RM-24X/92X Rev.4.2との接続
  - 5.14 RM-24X/92X\_Ver Rev.4.2 SW1、SW2 の設定方法
  - 5.15 RM-24X/92X\_EV Rev.4.2 J1 の設定方法
  - 5.16 RM-92XUSBを用いた開発方法

# 1.はじめに

本書では、92A/92C開発キット用サンプルソフトウェア(SimpleMACStd92A/92C)を使用した開発環境の構築方法について説明します。

RM-92A/92Cは、STマイクロエレクトロニクス社のSTM32L151(ARM CortexM3)を搭載しているため、ARM環境の開発環境が必要になります。

弊社では、IARシステムズの統合環境(EWARM)、及びJi-Jet(JTAG-ICE)を推奨しており、ご提供するサンプルコードもEWARMに適合するコード体系になっています。

他メーカーの環境や、無償のGNU環境による開発も可能ですが、サンプルコード内のコンパイラ依存部をご使用の環境に合わせてコード変更して頂く必要があります。

## 2.開発の準備

### 2.1 開発に必要な環境 ※IAR社のICE(i-jet)を使用する場合

92A/92Cの開発方法として、2つの方法がありますので、目的と用途に応じて選択して下さい。

(方法1) お客様ご自身で、92A/92C内に書き込むF/Wの開発から着手される場合

(方法2) 弊社でご用意するサンプルソフトウェアを、そのままご使用になる場合

	必要な機材	説明	SDK標準セット	方法1	方法2
1	RM-24X/92X_EV (開発ボード)	RM-92A/92C専用の開発ボード本体 (対向通信用として最低2台必要)	●	●	●
2	RM-92A/92C 通信モジュール	RM-92A/92C通信モジュール本体 (対向通信用として最低2台必要)	●	●	●
3	USBケーブル(mini/マイクロ)	RM-24X/9XX_EVとパソコンとの接続(最低2本) ※開発ボードのバージョンにより、使用するUSBケーブルの種類が異なります。	●	●	●
4	i-Jet、又は ST-Link	ICEによるトレースデバッグ、F/Wの書き込み	オプション	●	
5	IAR統合開発環境 (EWARM)	コンパイル、及びデバッグのソフトウェア開発用	オプション	●	
6	SimpleMACstd92A/92C Projectソースコード	RM-92A/92C用のサンプルプログラム	●	●	
7	シリアル通信ソフト	汎用のフリーソフト等 (teratermなど)	●	●	●

※「RM-92A開発キット」の(方法1)で使用するJTAG-ICE、EWARM(統合開発環境)はオプションになります。  
弊社からご購入頂けます。

# 2. 開発環境の準備

## 2.2 開発キット同胞内容

### 【標準セット内容】

◆RM-RM-92A/92C(13×2ハーフピッチコネクタ)



◆RM-24X/92X\_EV (開発ボード)



モジュールは、内臓アンテナタイプと外部アンテナタイプの2種類あります。  
開発キットをご注文の際、ご指定頂きます。

### 【追加オプション】

◆i-Jet (JTAG-ICE) ※オプション IAR社製品



◆パケットモニタツール RFLINK社製品



◆ST-Link ※オプション STマイクロ社製品



◆コンパイラ・デバッグ統合環境(EWARM) ※オプション IAR社製品



# 2. 開発環境の準備

## 2.3 開発環境構築の流れ

標準的な開発環境を構築するまでの流れを以下に説明します。



(方法2)による開発を行い場合は、④～⑥ は不要です。  
コンフィグレーションを実行した後、直ぐに使用する事が可能です。

③は、i-jetを使用せず、STマイクロエレクトロニクス社専用のJTAツール(ST-Link)を使用する場合に必要になります。

# 2.開発環境の準備

## 2.4 方法1の環境構築例

i-Jet (JTAG-ICE)



RM-24X/92XEV



無線

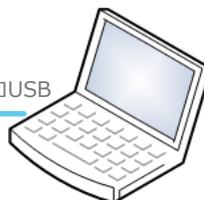
マイクロUSB



RM-24X/92XEV



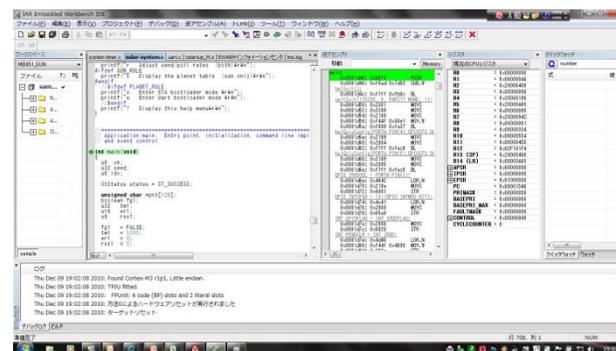
マイクロUSB



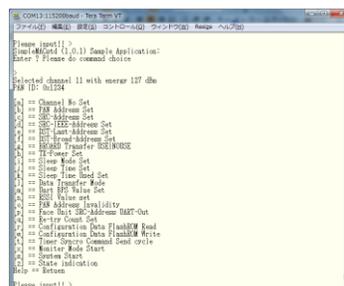
汎用シリアル通信ソフト(TeraTerm)



IAR社 統合環境ツール (Embedded Workbench)



汎用シリアル通信ソフト(TeraTerm)







## 2.開発環境の準備

### 2.6 ドライバのインストール

RM-24X/9XX\_EVを使用する為に、PCにFT232Cドライバのインストールを行います。

手順1 添付CD、又は弊社HPの「ドキュメントダウンロード」ページから、RM-205シリーズの「FTDI-USBドライバ(FT232)」をダウンロードします。(RM-24X9XX\_EVはRM-240EV用のドライバの共通です)  
※この時USB機器はPCに挿入しないで下さい。

手順2 ZIPファイルを任意の場所に解凍します。(システムドライブ(通常はC:ドライブ)のルート上のフォルダを推奨します)

手順3 PCに、RM-24X9XX\_EVをPCとUSB接続します。

手順4 PCのOSのバージョンにより、FT232デバイスをサポートしている場合があります。  
※自動インストールされた場合は、手順7に進みます。

手順5 コントロールパネル→デバイスマネージャーを開きます。

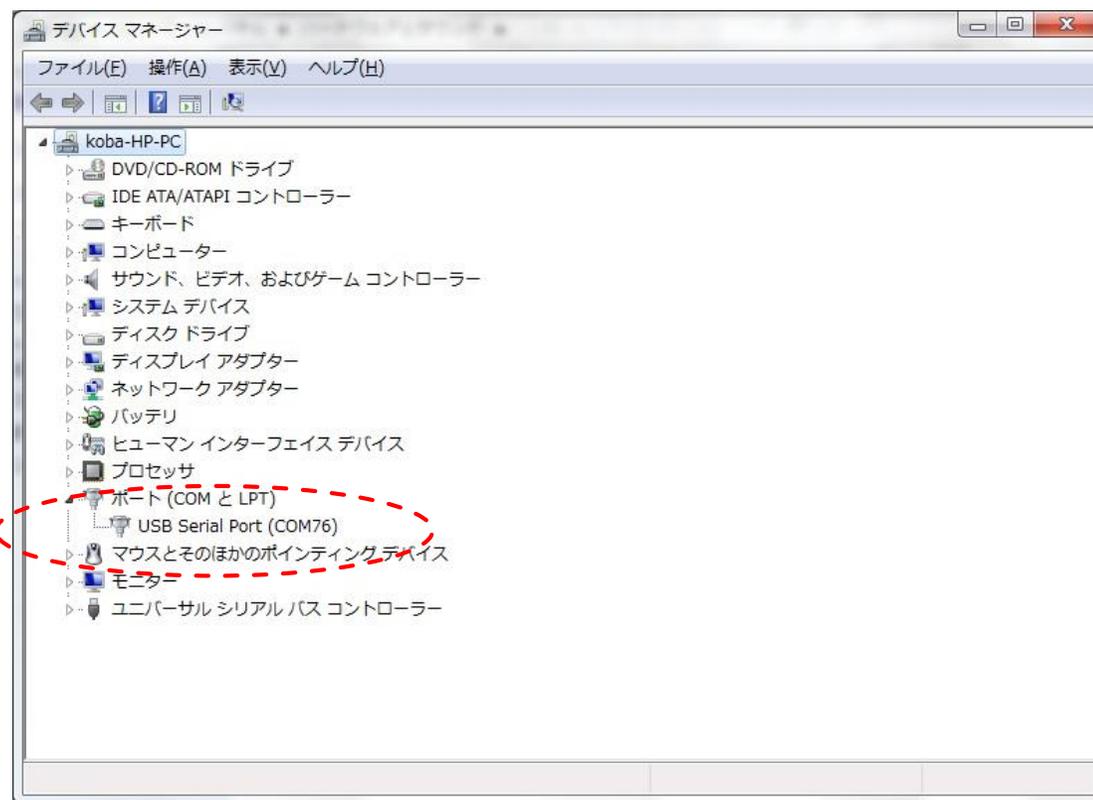
手順6 不明なデバイスとして認識されていた場合、手動設定で、手順2 で指定した場所を指定して、ドライバ設定を完了します。

手順7 正常にドライバ認識がされると、仮想COMポートとして認識されます ※次頁参照

# 2.開発環境の準備

## 2.6 ドライバのインストール

正しくインストールが出来た状態のコントロールパネル表示

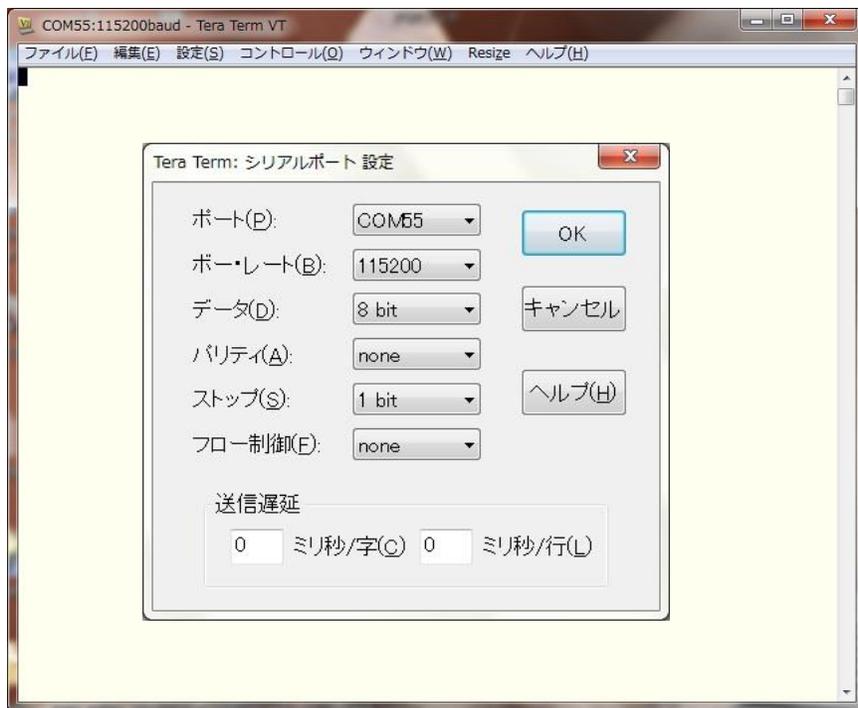


# 2. 開発環境の準備

## 2.7 シリアル通信ソフトの設定

PCとRM-92A/92Cとはシリアル通信でコンフィグレーションを行います。  
通信ソフトウェアは、Windows標準ツール(ハイパーターミナル等)や、LinuxのminiCOMなどの他、フリーソフトなどで前章で設定したCOMポート番号で接続します。

下記例は、フリーソフトのTeratermの画面です。



シリアル通信パラメータは、左図の様に設定して下さい。

※通信速度は、コンフィグレーションで変更可能ですが、工場出荷時は、左図の設定になっています。

# 2.開発環境の準備

## 2.8 FTDI(UART/USB変換IC)の設定

弊社の製品で、開発ボード(RM-24X/92X\_EV)や、RM-92XUSB、RM-240USBなど、PCと接続してシリアル通信を行う機能を持つ製品は、FTDI社のFT232RLデバイスを使用しています。

このデバイスは、UART信号をUSB信号に変換する機能を持ち、専用ドライバがインストールされているPCやスマートフォンに接続した時に、仮想COMポートとしてシリアル通信を行う事が出来ます。

弊社の製品をPC等に接続した時に、COMポートとして表示されない場合や、COMポートとして認識されているにもかかわらず入出力が出来ない場合などに、以下の方法で再設定を行う事が出来ます。

【手順1】 弊社HPの、ダウンロードサイトから、「FTDI設定ツール」カテゴリから、FT\_Prog\_v2.4.2.zip をダウンロードします。(バージョンは随時更新します)

【手順2】 ダウンロードしたzipファイルを任意のフォルダに解凍します。

【手順3】 解凍すると、FT\_Prog\_v2.4.2 というフォルダが現れますので、フォルダ内の FT\_PROG.EXE ファイルを実行します。

【手順4】 PCに、開発ボード(RM-24X/92X\_EV)や、RM-92XUSB、RM-240USBなどの対象機器をUSB接続します。

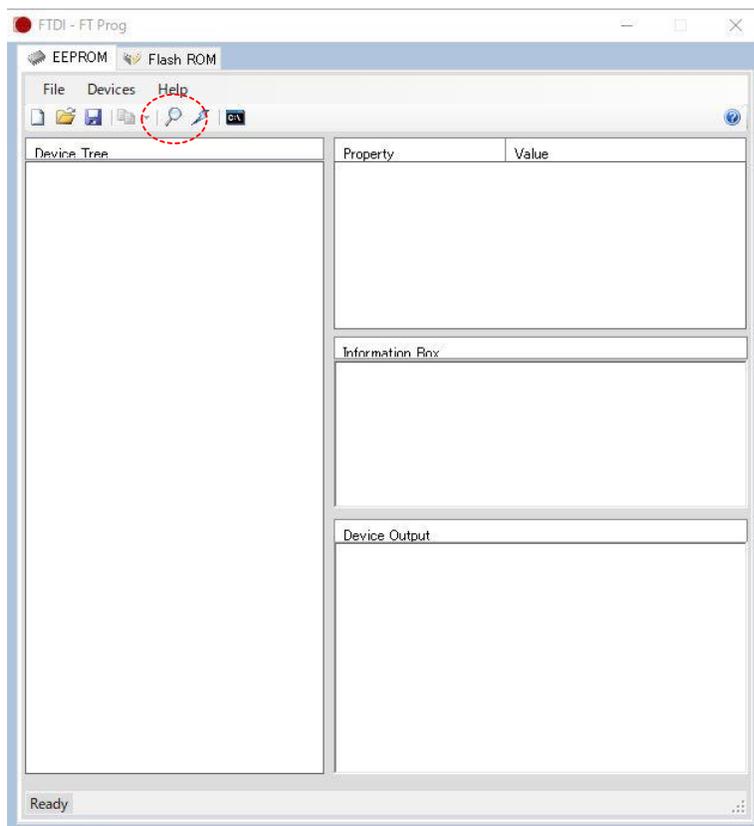
以下、次ページに続きます。

## 2.開発環境の準備

### 2.8 FTDI(UART/USB変換IC)の設定

【手順4】 ツールが起動すると、図1 の画面が表示されますので、対象機器がUSB接続されている事を確認してから、 のアイコンをクリックします。

図1 ツール起動時初期画面

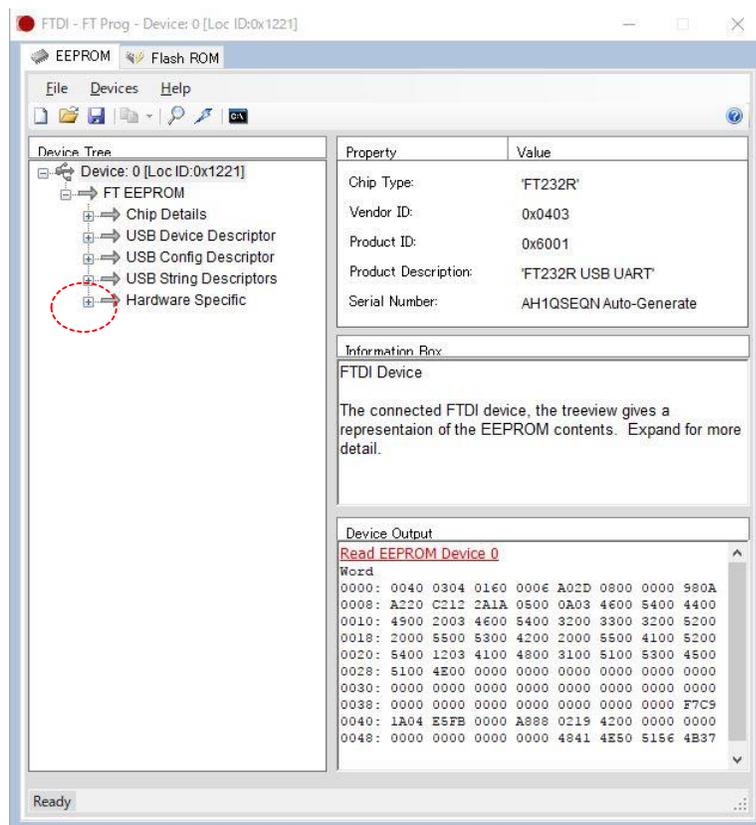


# 2.開発環境の準備

## 2.8 FTDI(UART/USB変換IC)の設定

【手順5】 手順4の操作により、下記画面が表示されますので、の部分をクリックします。

図2 FT232RT 検索結果画面

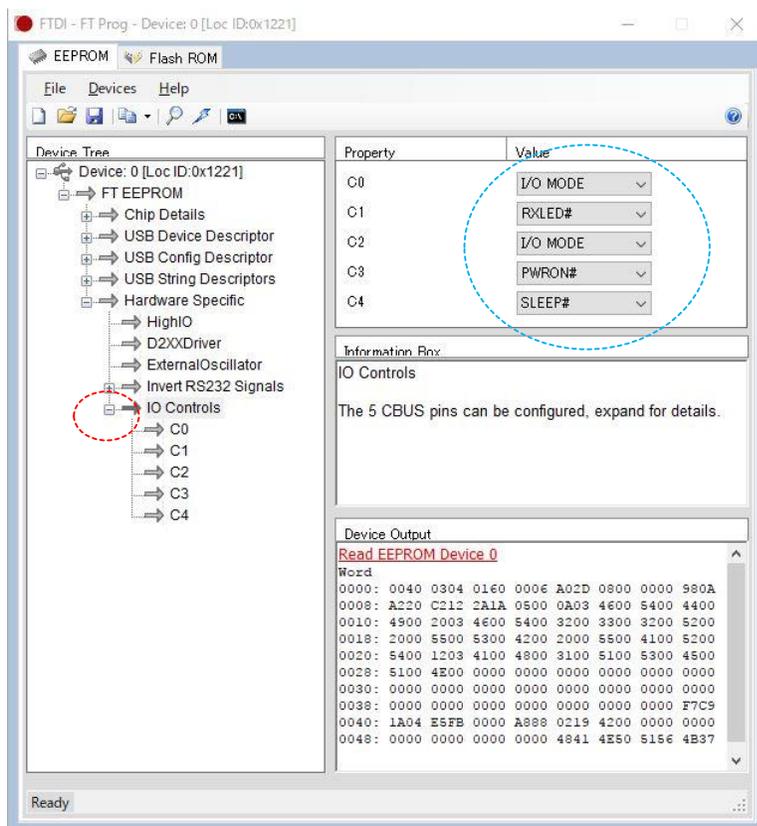


# 2.開発環境の準備

## 2.8 FTDI(UART/USB変換IC)の設定

【手順6】 手順5の操作により、下記画面が表示されますので、 の部分をクリックして、IO Controlsの詳細を表示し、 の部分を、図3の表示内容と同じ設定にします。

図3 FT232RTデバイス状態表示画面

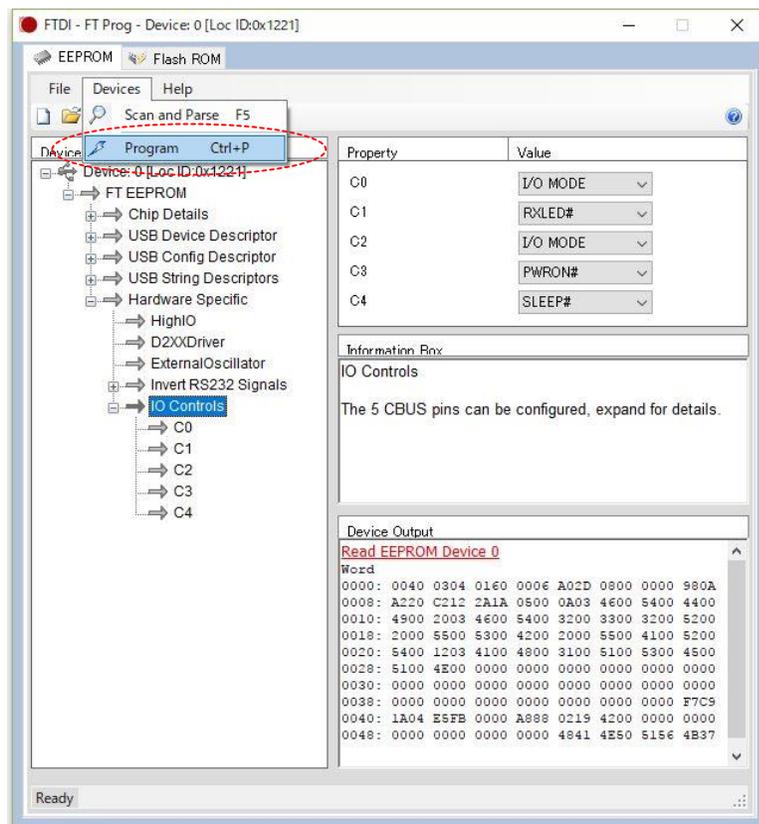


# 2.開発環境の準備

## 2.8 FTDI(UART/USB変換IC)の設定

【手順7】 Devicesメニューから  のProgram を実行します。

図4 FT232RT Program実行前画面



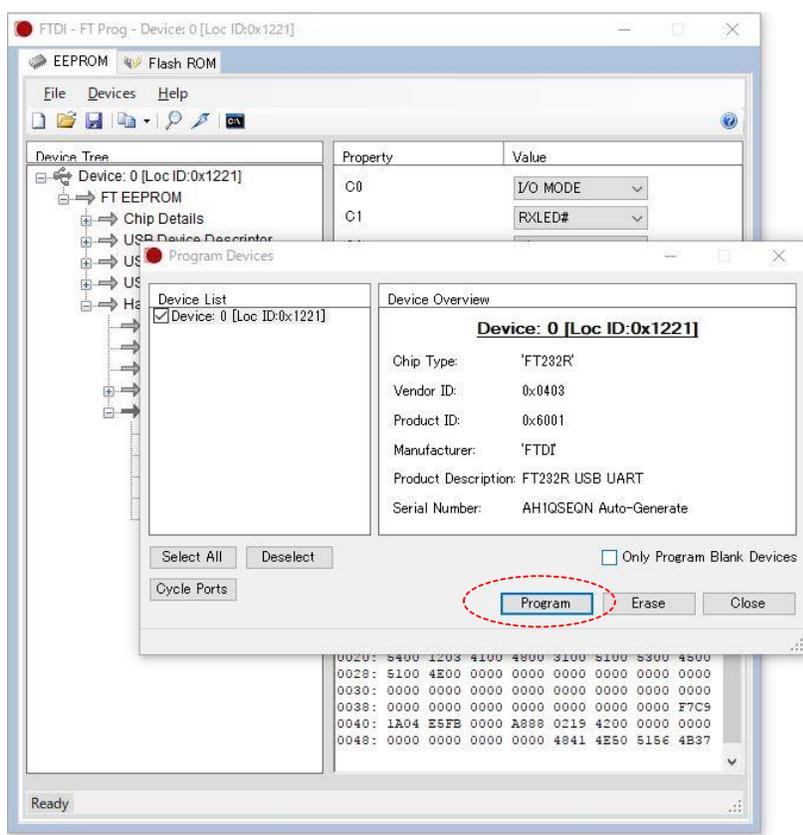


# 2.開発環境の準備

## 2.8 FTDI(UART/USB変換IC)の設定

【手順8】 手順7の操作により、下記のダイアログ画面が表示されますので、の部分の Program をクリックします。この操作によりFT232RTの設定が更新されます。

図5 FT232RT 設定更新実行画面

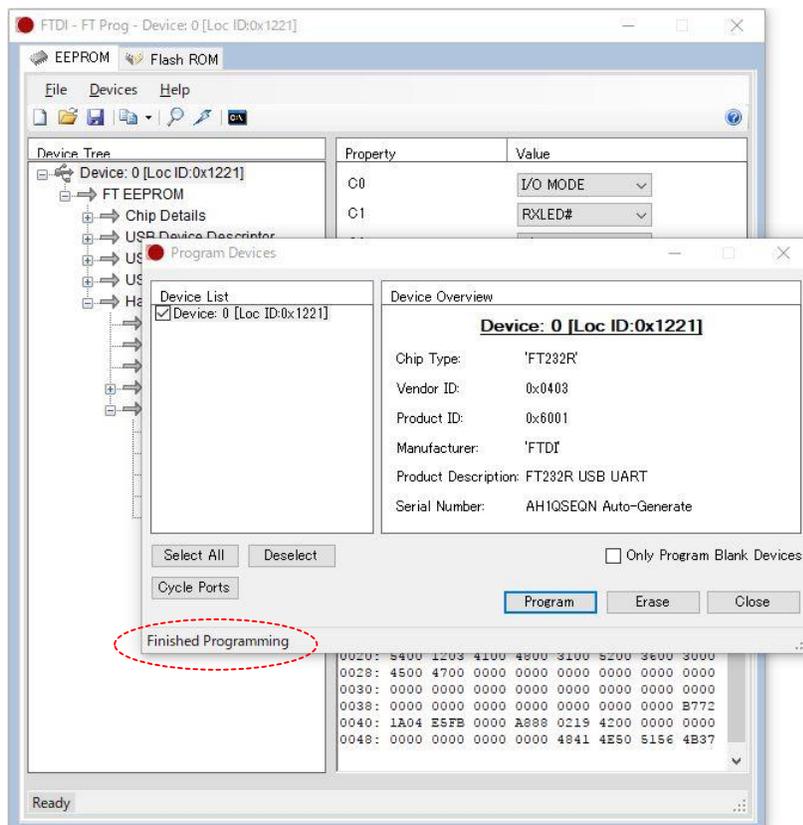


# 2.開発環境の準備

## 2.8 FTDI(UART/USB変換IC)の設定

【手順9】 手順8の操作により、更新が終了すると  部に、Finished Programming と表示されたら設定完了です。

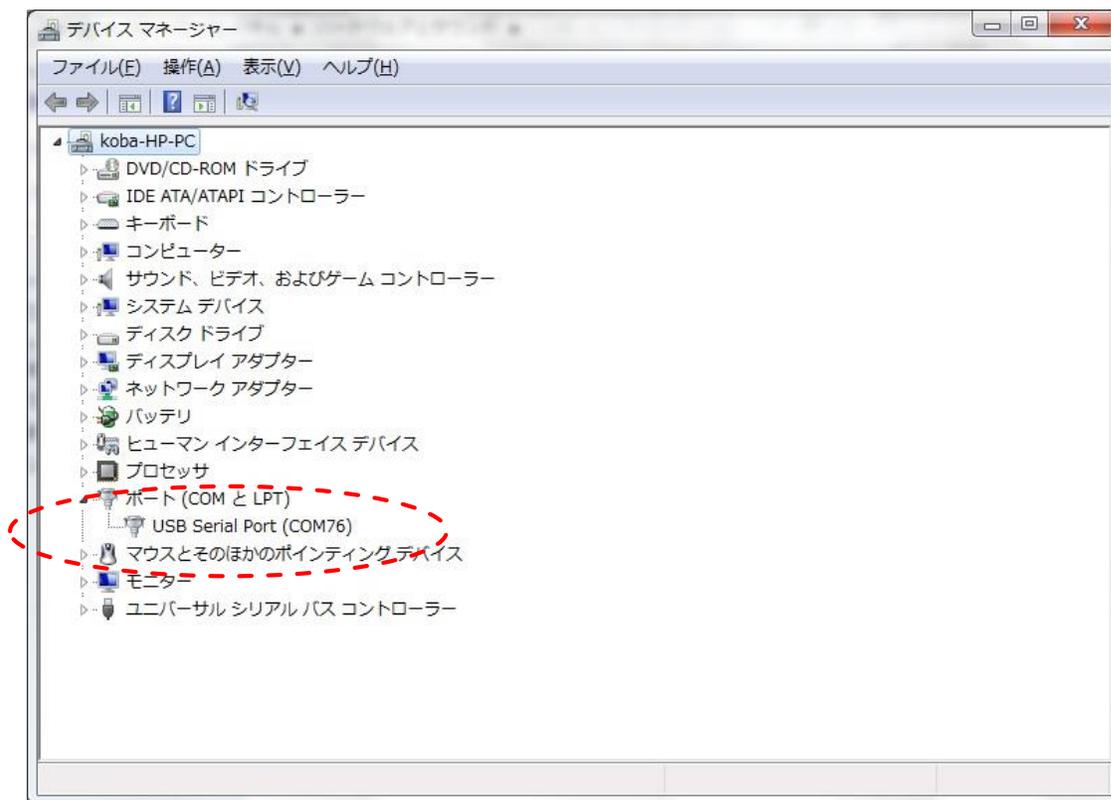
図6 FT232RT 設定更新実行完了画面



## 2. 開発環境の準備

### 2.8 FTDI(UART/USB変換IC)の設定

【最終手順】 対象機器をUSBから一度外して、再度挿入します。この時に、デバイスマネージャーにより、COMポートとして認識されている事を確認して作業終了です。



# 3.内蔵FlashROMへの書込み

## 3.1 書込みに必要なもの

RM-92A/92Cの内蔵FlashROMに、プログラムを書き込む為には、以下のツールが必要になります。

	ツール名	説明
1	RM-24X/92X_EV (開発ボード)	RM-92A/RM-92C専用の開発ボード
2	RM-92A/RM-92C	RM-92A/RM-92C本体
3	マイクロUSBケーブル	RM-24X/920_EVとパソコンとを接続するのに必要になります。
4	i-Jet、ST-Link	IAR社製JTAG-ICE 又は STマイクロ社製 JTAGツール
5	IAR統合開発環境 (EWARM)	IAR社製統合環境

# 3.内蔵FlashROMへの書込み

## 3.2 プロジェクトからの書き込み手順

以下の手順で書き込み作業を行います。

手順1 お客様のPCに、IAR社のIAR Embedded Workbench統合環境をインストールします

手順2 SimpleMACstd92A/92Cのプロジェクトファイルをクリックして、統合環境を起動します。  
※¥RM92x\_SimpleMACstd\_Project\_vx.x.xxx¥EWARM¥SimpleMACstd.eww

手順3 統合環境により、「リビルド」を実行します。(詳細な説明は、統合環境のマニュアルを参照下さい。)

手順4 統合環境により、「ダウンロードしてデバッグ」を実行します。  
※この操作によりRM-92A/92CのFlashROMへの書込みが実行されます。

# 3.内蔵FlashROMへの書込み

## 3.3 ビルド済みROMデータの書き込み手順

以下の手順で書き込み作業を行います。

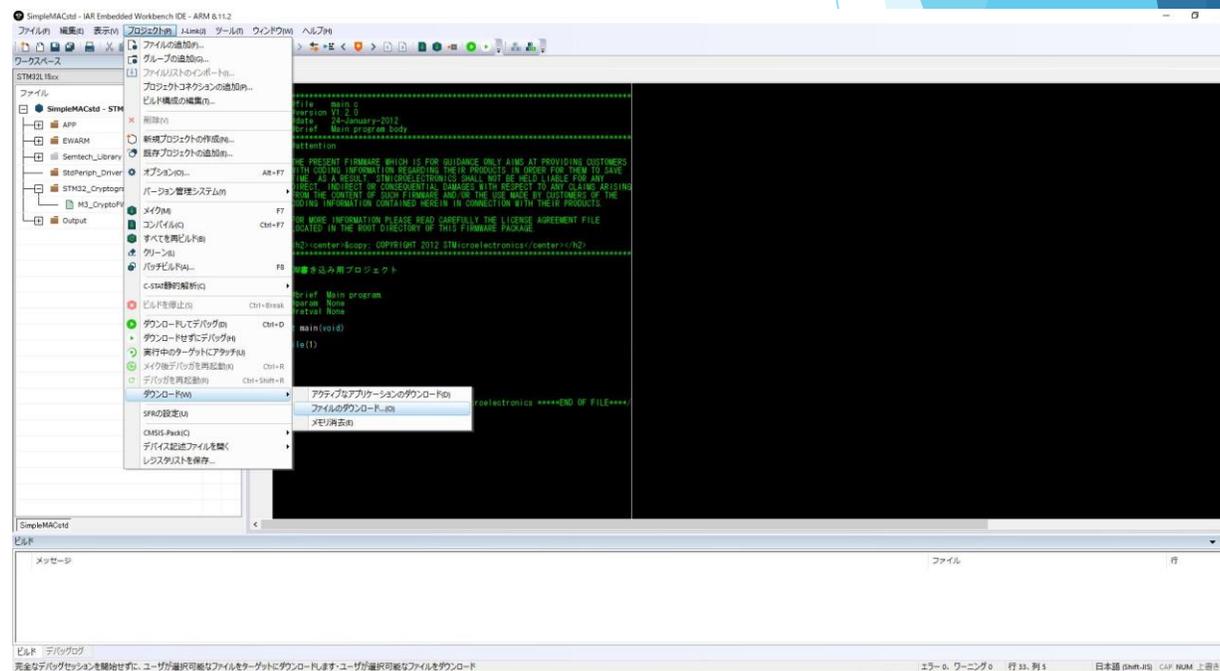
手順1 お客様のPCに、IAR社のIAR Embedded Workbench統合環境をインストールします

手順2 ROM書き込み専用プロジェクトを開きます  
※ROM\_write\_project¥EWARM¥SimpleMACstd.eww

手順3 統合環境の、  
projectメニュー → ダウンロード → ファイルのダウンロード  
を選択します

手順4 手順3の操作で、書き込むROMデータを指定すると、ROMデータの書き込みが開始  
されます。

※ビルド操作は絶対に行わないでください。



# 4.統合環境(EWARM)を使用した開発方法

## 4.1 統合開発環境構築までの流れ

RM-92A/RM-92CのF/W開発を行う場合の開発ツールとして、IAR社の開発環境を推奨しています。本章では、ツールのダウンロードからビルド後のデバッグまでの流れについて説明します。

手順1 弊社HPの「ドキュメントダウンロード」ページから、AM-900シリーズの「ソフトウェアツール」→「EWARM(コンパイラ+統合環境)」をクリックし、IAR社専用サイトに接続します。

手順2 IAR社サイトから、「ARM用 30日間期間限定版」を選択して、サイトの指示に従ってユーザー登録を行います。

手順3 ユーザー登録後に、登録したメールアドレスにIAR社からのメールが届きますので、指示に従って環境のダウンロードを行います。

手順4 ダウンロードした実行ファイルを実行します。 ※次頁参照

手順5 SDK添付のCD内から、Projectフォルダ内のプロジェクトサンプルをEWARMにより開きます。

手順6 RM-24X/9XXEV(開発ボード)とJTAG-ICE(i-Jet)を接続します。また、PCとUSB接続を行い、シリアル通信ソフトを起動します。

手順7 ダウンロードしたプロジェクトをビルとして、エラーが無い事を確認します。

手順8 RM-24X/92X\_EVに、実行モジュールをダウンロードしてデバッグできる事を確認します。

# 4.統合環境(EWARM)を使用した開発方法

## 4.2 統合開発環境のインストール

<手順4>の説明

ダウンロードした実行ファイルを実行すると、下記が表示されますので、「IAR Embedde Workbenchのインストール」を選択します。  
後は指示に従ってインストール作業を完了して下さい。(GUI画面は随時変更されます)



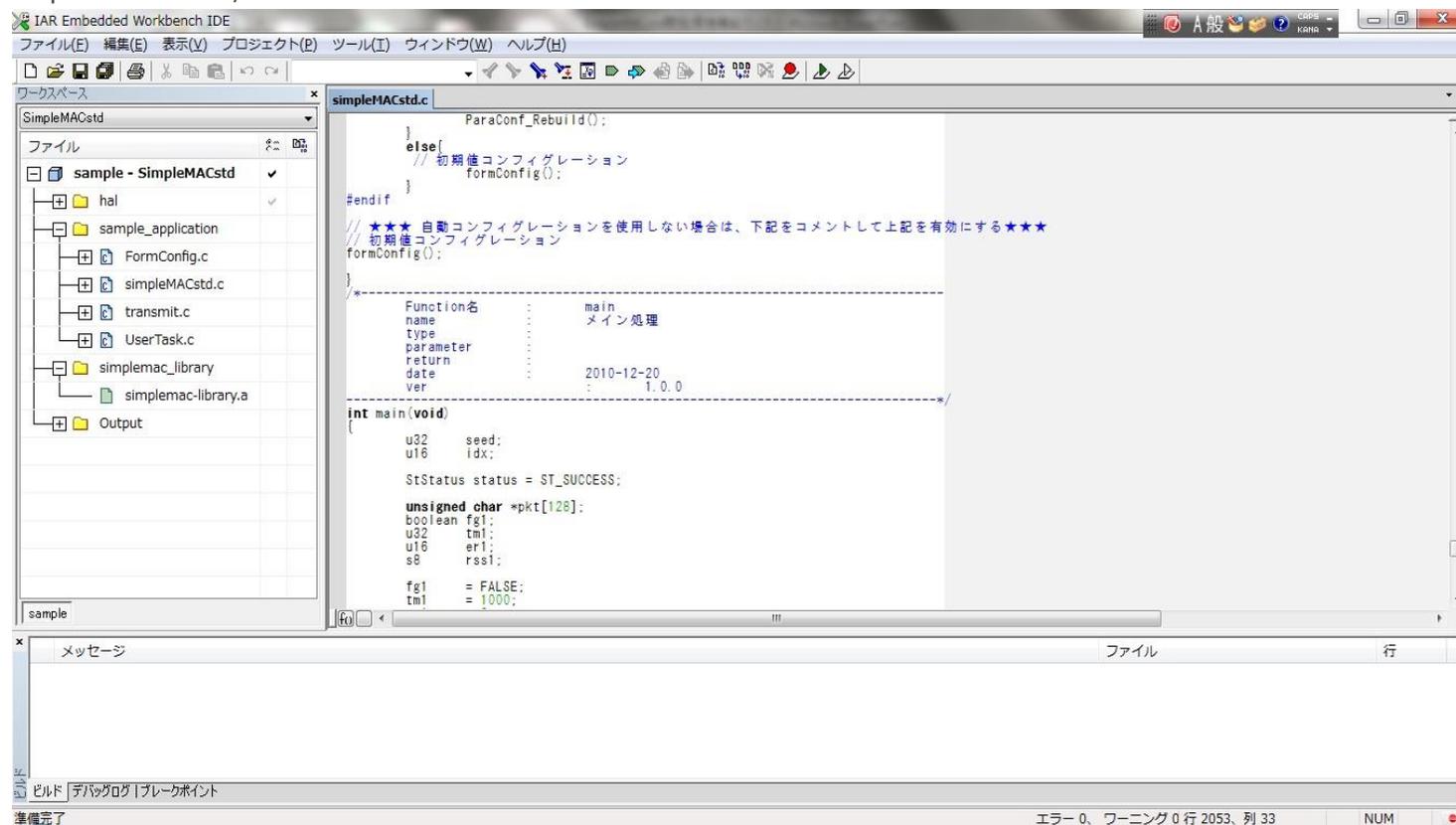


# 4. 統合環境(EWARM)を使用した開発方法

## 4.3 プロジェクトファイルの展開

<手順7>の説明

SimpleMACStd92A/92Cワークスペースを開いた時の表示例



ビルド | デバッガログ | ブレークポイント

準備完了

エラー 0、ワーキング 0 行 2053、列 33

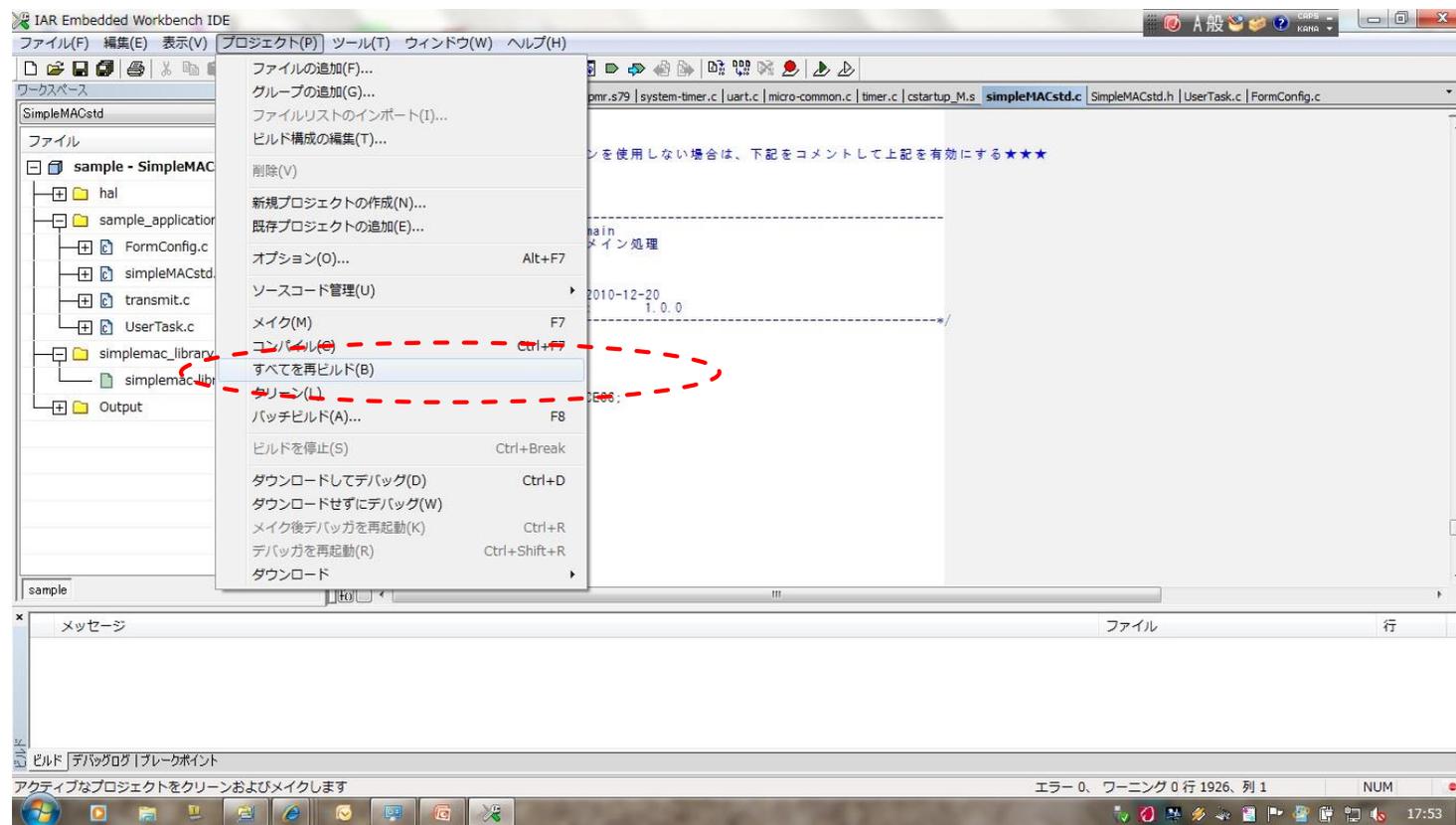
NUM

# 4.統合環境(EWARM)を使用した開発方法

## 4.4 プロジェクトのビルド

<手順7>の説明

プロジェクトのビルドを実行します。下図の通り、「プロジェクト」→「全てを再ビルド」を実行します。

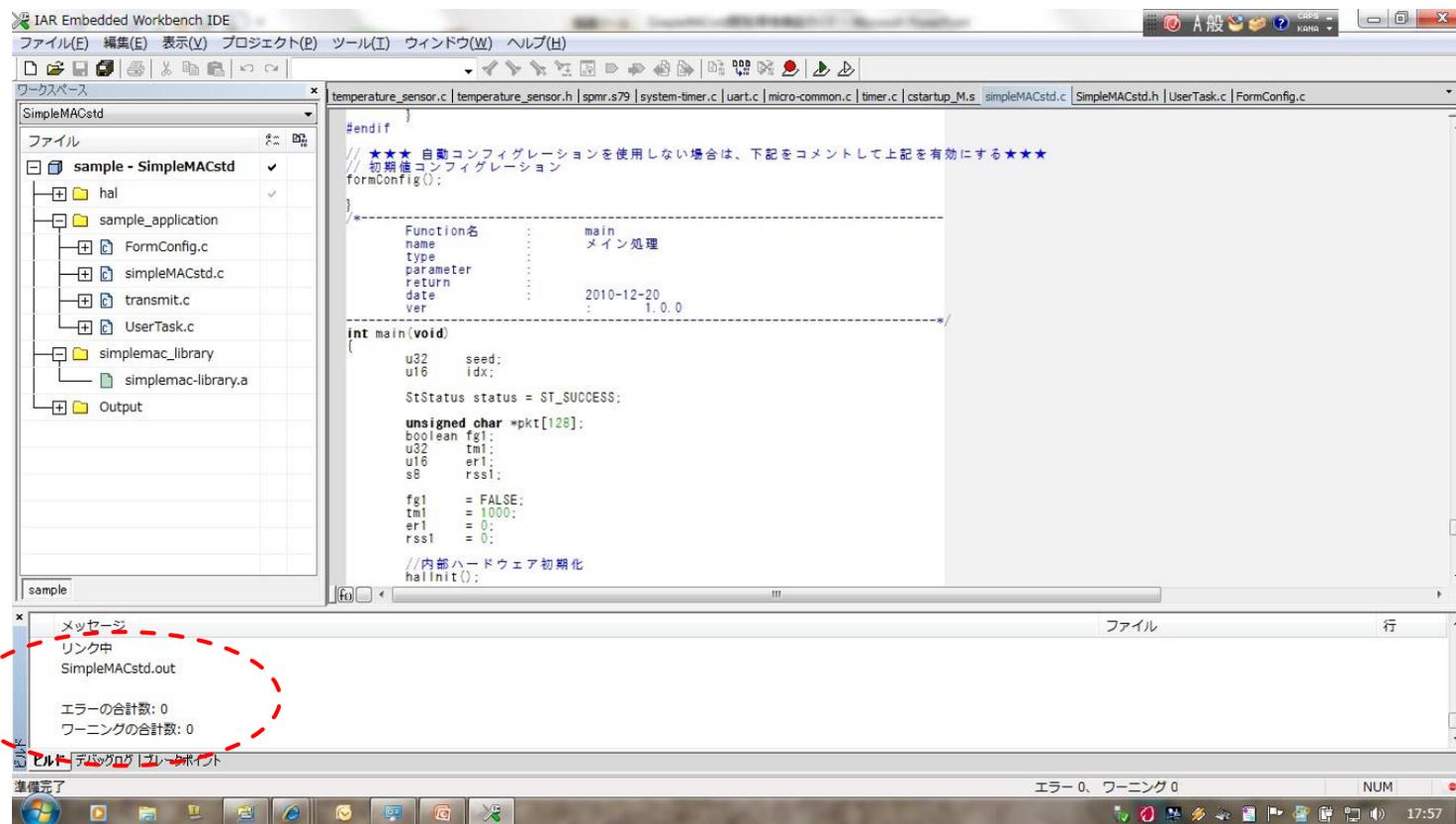


# 4. 統合環境(EWARM)を使用した開発方法

## 4.5 プロジェクトのビルド

<手順7>の説明

ビルド終了後に、エラー、ワーニング、が無い事を確認します。

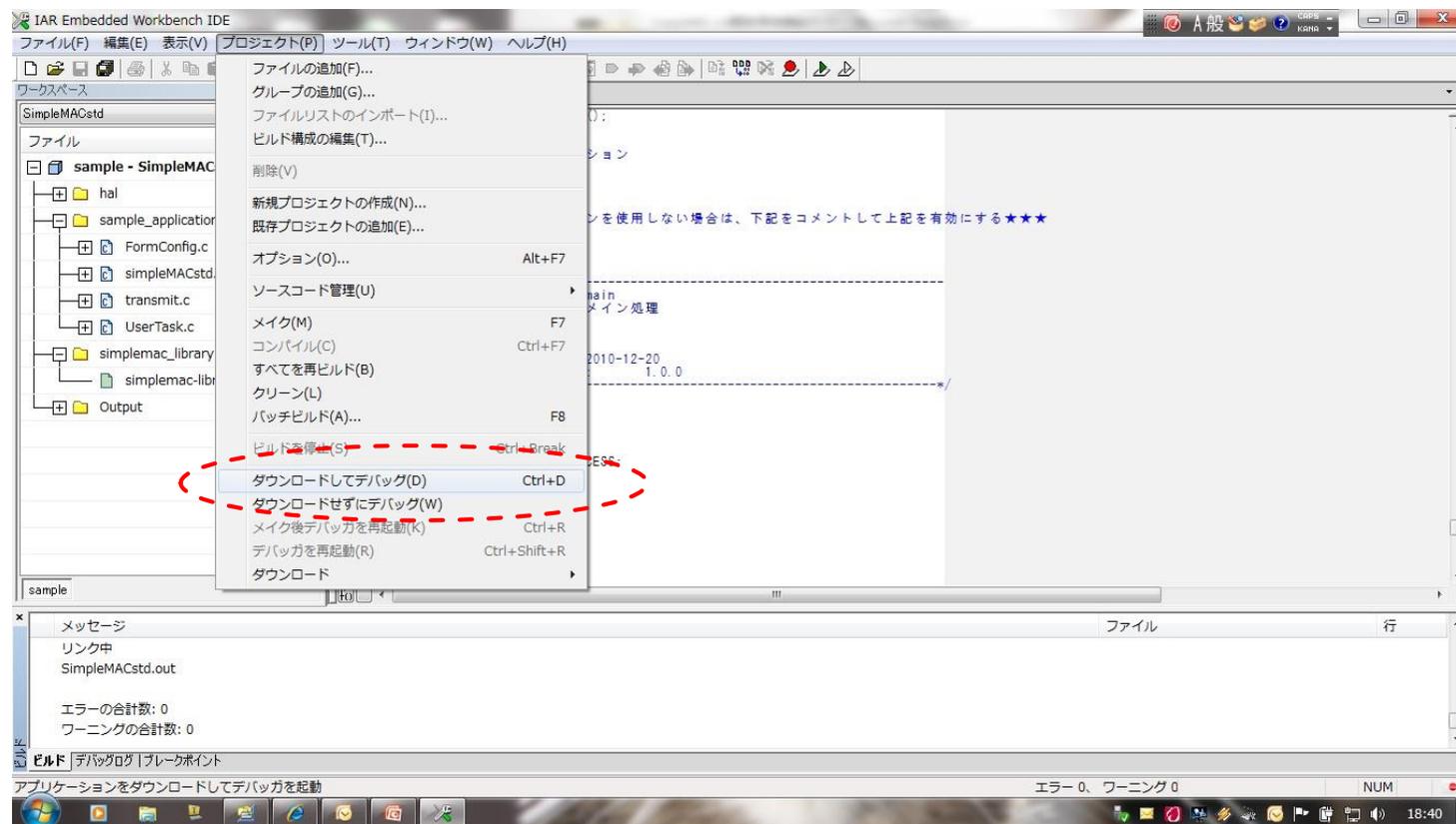


# 4.統合環境(EWARM)を使用した開発方法

## 4.6 実行モジュールのRM-92A/92Cへのダウンロード

<手順8>の説明

「プロジェクト」→「ダウンロードしてデバッグ」を選択します。→RM-92A/RM-92Cの内蔵FROMに実行ファイルがダウンロードがされます。



# 4. 統合環境(EWARM)を使用した開発方法

## 4.7 実行モジュールのRM-92A/92Cへのダウンロード

<手順8>の説明

「プロジェクト」→「ダウンロードしてデバッグ」を選択します。→RM-920の内蔵FROMに実行ファイルがダウンロードがされます。

The screenshot displays the IAR Embedded Workbench IDE interface. The main window shows the source code for `simpleMACStd.c`. A red dashed oval highlights the `int main(void)` function signature. Below the code, a table lists function details:

Function名	main
name	メイン処理
type	
parameter	
return	
date	2010-12-20
ver	1.0.0

To the right, the 'レジスタ' (Registers) window shows the state of various CPU registers, including R0 through R14, xPSR, APSR, IPSPR, EPSR, PC, and control registers like MSP, PSP, PRIMASK, BASEPRI, and FAULTMASK.

At the bottom, the 'ログ' (Log) window shows the following output:

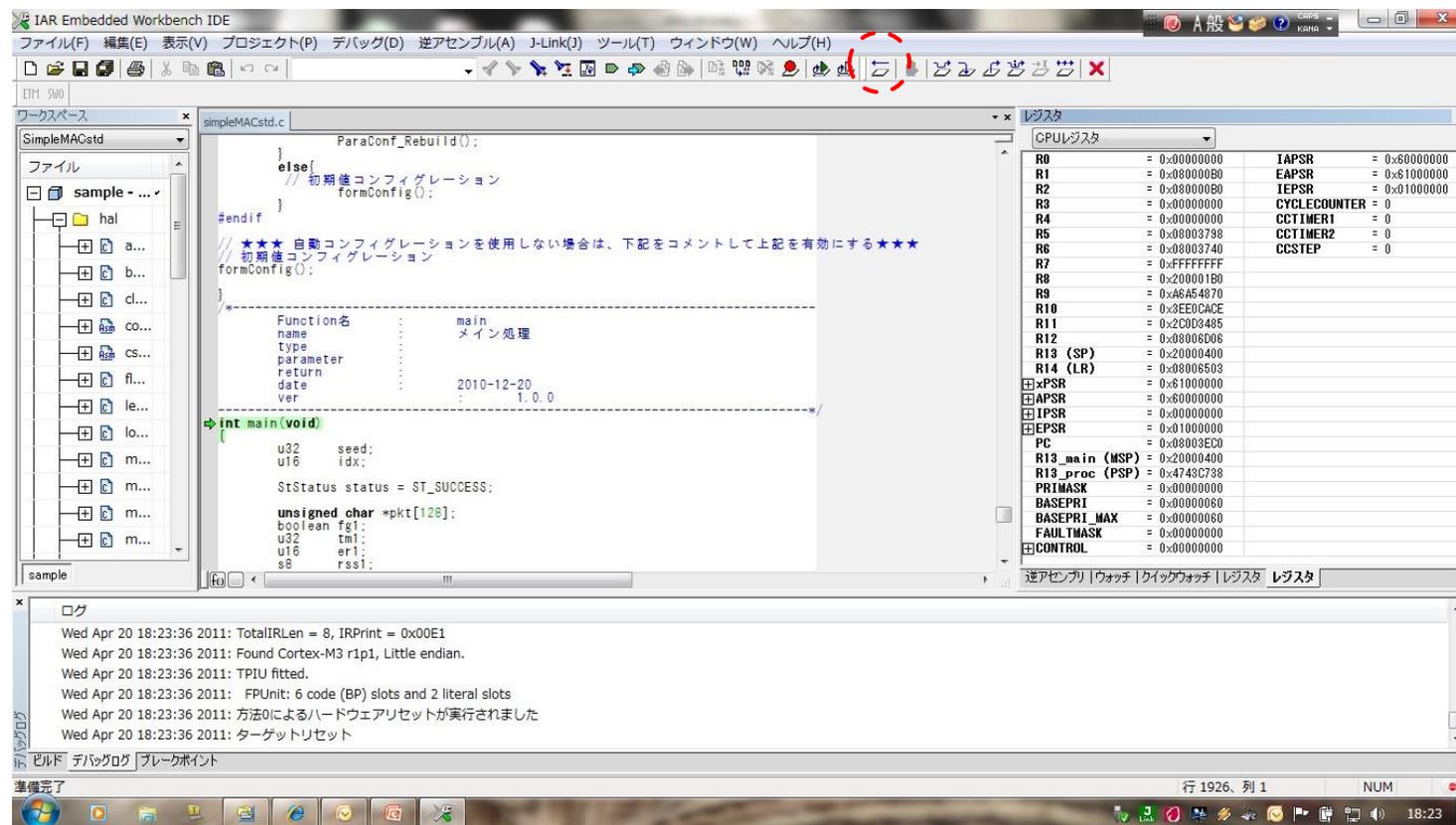
```
Wed Apr 20 18:23:36 2011: TotalIRLen = 8, IRPrint = 0x00E1
Wed Apr 20 18:23:36 2011: Found Cortex-M3 r1p1, Little endian.
Wed Apr 20 18:23:36 2011: TPIU fitted.
Wed Apr 20 18:23:36 2011: FPUUnit: 6 code (BP) slots and 2 literal slots
Wed Apr 20 18:23:36 2011: 方法0によるハードウェアリセットが実行されました
Wed Apr 20 18:23:36 2011: ターゲットリセット
```

# 4. 統合環境(EWARM)を使用した開発方法

## 4.8 デバッグ開始

<手順8>の説明

下図の赤丸部をクリックして実行します。



# 4.統合環境(EWARM)を使用した開発方法

## 4.9 プログラムの実行～シリアル通信ソフトによる確認

<手順8>の説明

前頁の「実行」操作により、正常に実行されると、PCのシリアル通信ソフト(以下はTeraterm)に、SimpleMACStdから起動メッセージが表示されます。

最初に表示される画面 → 変調モードで 1:LoRa を入力すると → 次に表示される画面 (コンフィギュレーションの基本メニュー)

```
COM73 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)
/*****
COPYRIGHT 2016 RF-Link. All rights reserved.
STM32L151xB CortexM3
RM-92A SimpleMAC Project [Ver.1.1.11b]
*****/
Transmit RF Mode LORA or FSK or GFSK? [1:LORA 2:FSK 3:GFSK]= []
```

```
COM73 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)
*****
* RM92A SimpleMACStd Command List [LORA Mode]
*****
[a] : Channel No Set [24 - 61 ]
[b] : PAN Address Enable [0:Not Use 1:Use ]
[c] : SRC-Address Set [1 - 65534 ]
[d] : DST-Address Set [1 - 65535 ]
[e] : Unit Mode Set [0:Parent1:Child]
[f] : Routing Mode [0:Fixation 1:AutoRouting 2:NonRouting]
[g] : RF Settings-----
      [1:TX-Power Set 2:Bandwidth Set ]
      [3:Factor(SF) Set 4:Error Coding Set ]
      [5:Optimize Set ]
[h] : Ack Request Set [0:Not Use 1:Use ]
[i] : Data Transfer Mode [0:Discharge 1:Frame 2:TimerSend ]
      [3:SleepTimerSend(Non Routing Only) ]
[j] : Sleep Mode [0:Not Use 1:Use]
[k] : UART BaudRate Set
      [0:4800 1:9600 2:14400 3:19200 4:38400 ]
      [5:57600 6:115200 7:230400 8:460800 9:921600 ]
[l] : Recv Packet Output Set
      [1:RSSI Output Set 2:Transfer(SRC) Address Output Set]
      [3:CR+LF OutPut]
[m] : Carrier Sense Set [0:Not Use 1:Use ]
[n] : RF-Data AES KEY [0:Not Use 1:Use ]
[o] : RTC Clock Source [0:LSI 1:LSE ]
[p] : Transmit-Time-Total Count Set (Test Only, Default is <Use>)
      [0:Not Use 1:Use ]
[q] : Low-level noise filter function [0:Not Use 1:Use ]
[r] : System Start
[s] : Software Reset
[x] : Setting Data EEPROM Save
[y] : Setting Data EEPROM Read
[z] : EEPROM Configuration Data Default Set (Reset it)
[?] : State indication
Help : Return
Please input >[]
```

# 4.統合環境(EWARM)を使用した開発方法

## 4.10 SimpleMAC-92X-SDK-std3 (中継版セット)の出荷時の設定

中継版セットの場合、3つのモジュールに対しての出荷時設定は以下の様になっています。

自動スタート設定はOFFにして出荷していますので、yコマンドで設定内容をFlashROMから読み出して sコマンドでネットワークスタートして下さい。

Yコマンドで読み出した後で、?コマンドで設定内容をteraterm確認ができます。

親機の設定画面

```
COM74 - Tera Term VT
?
***** [Settings] *****
[*]RF Mode : [LORA]
[a]RF-Channel : [24](Frequency[920600000Hz])
[b]PAN-ID : [Enable] PAN-ID[0x1234]
[c]SRC-ID : [0x0000]
[d]LAST-DST-ID : [0xFFFF]
[e]Unit Mode : [Parent]
[f]Routing Mode : [Auto-Routing]
[g]RF Settings : Tx-Power [13dBm]
: Bandwidth [125kHz]
: Factor [SF10]
: Error Coding [2: 4/6]
: Optimize [ON]
[h]Ack Request : [Disable]
[i]Data Mode : [Discharge]
[j]Sleep Mode : [Not Use]
[k]UART BaudRate : [115200bps]
[l]Recv Packet Output : RSSI [Enable] SRC-Address[Disable]
: CR+LF [Enable]
[m]Carrier Sense : [Use] Retry Count[0]
[n]RTC : [LSE]
[p]Transmit Total Time Set: [Not Use]
[q]Low Level Noise Filter : [Not Use]
[r]AES KEY : [Not Use]
EEPROM Data :
0A 00 06 AE 01 12 34 00 00 00 18 FF FF 00 00 2B
7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C 01
00 00 00 00 00 00 00 00 00 0B E8 02 00 00 00
03 E8 00 00 00 00 00 0A 00 00 00 0A 01 01 00
00 0D 36 DF 3D C0 00 01 C2 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 C3 50 01 00 00 02 01 00 01
*****
Please input >[]
```

子機1の設定画面

```
COM50 - Tera Term VT
?
***** [Settings] *****
[*]RF Mode : [LORA]
[a]RF-Channel : [24](Frequency[920600000Hz])
[b]PAN-ID : [Enable] PAN-ID[0x1234]
[c]SRC-ID : [0x0001]
[d]LAST-DST-ID : [0x0000]
[e]Unit Mode : [Child]
[f]Routing Mode : [Auto-Routing]
[g]RF Settings : Tx-Power [13dBm]
: Bandwidth [125kHz]
: Factor [SF10]
: Error Coding [2: 4/6]
: Optimize [ON]
[h]Ack Request : [Enable] TimeOut[3sec] NoAck Retry[3]
[i]Data Mode : [Discharge]
[j]Sleep Mode : [Not Use]
[k]UART BaudRate : [115200bps]
[l]Recv Packet Output : RSSI [Enable] SRC-Address[Disable]
: CR+LF [Enable]
[m]Carrier Sense : [Use] Retry Count[0]
[n]RTC : [LSE]
[p]Transmit Total Time Set: [Not Use]
[q]Low Level Noise Filter : [Not Use]
[r]AES KEY : [Not Use]
EEPROM Data :
0A 00 06 AE 01 12 34 00 01 00 18 00 00 00 00 2B
7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C 01
00 00 00 00 01 01 00 00 00 00 0B E8 03 00 00 00
03 E8 00 00 00 00 00 0A 00 00 00 0A 01 01 00
00 0D 36 DF 3D C0 00 01 C2 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 C3 50 01 00 00 02 01 00 01
*****
Please input >[]
```

子機2の設定画面

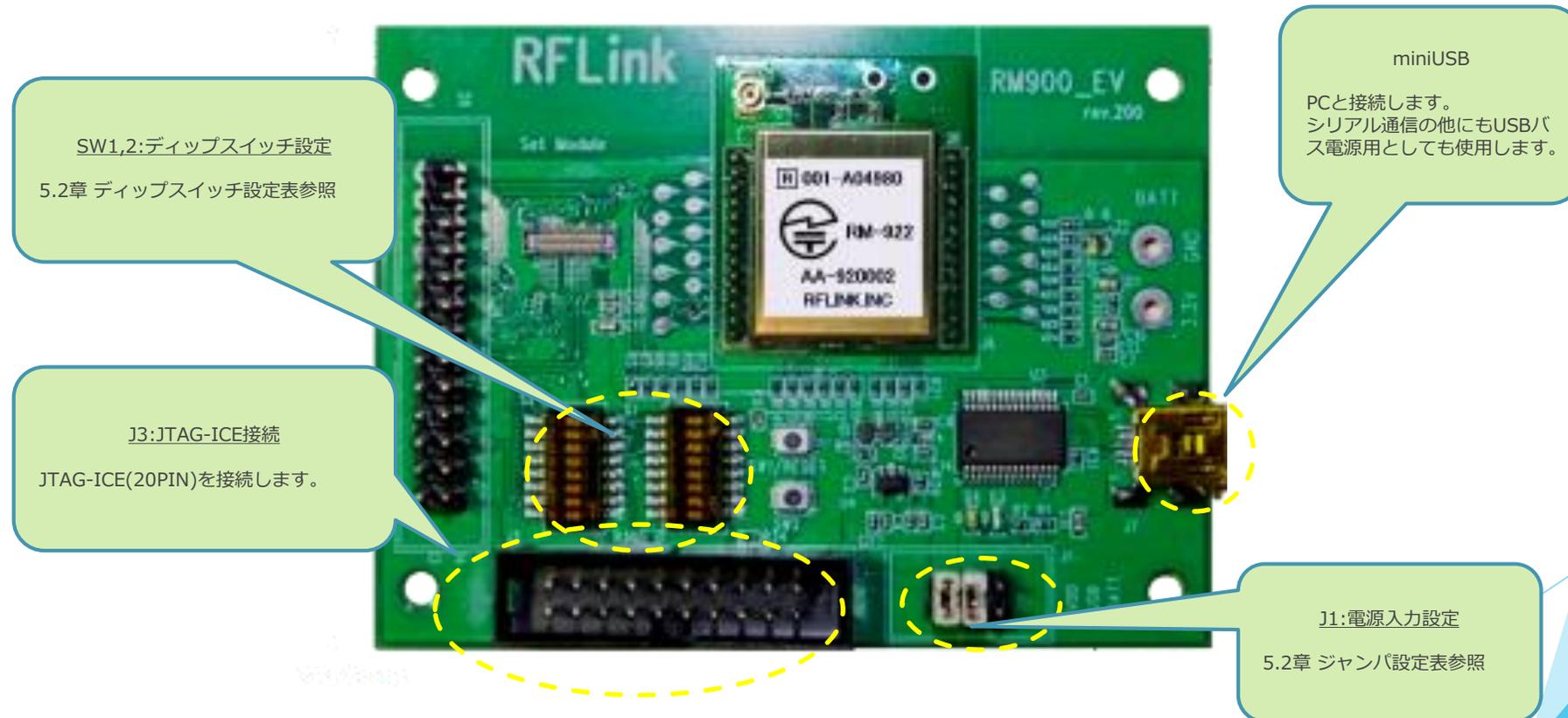
```
COM73 - Tera Term VT
?
***** [Settings] *****
[*]RF Mode : [LORA]
[a]RF-Channel : [24](Frequency[920600000Hz])
[b]PAN-ID : [Enable] PAN-ID[0x1234]
[c]SRC-ID : [0x0002]
[d]LAST-DST-ID : [0x0000]
[e]Unit Mode : [Child]
[f]Routing Mode : [Auto-Routing]
[g]RF Settings : Tx-Power [13dBm]
: Bandwidth [125kHz]
: Factor [SF10]
: Error Coding [2: 4/6]
: Optimize [ON]
[h]Ack Request : [Enable] TimeOut[3sec] NoAck Retry[3]
[i]Data Mode : [Discharge]
[j]Sleep Mode : [Not Use]
[k]UART BaudRate : [115200bps]
[l]Recv Packet Output : RSSI [Enable] SRC-Address[Disable]
: CR+LF [Enable]
[m]Carrier Sense : [Use] Retry Count[0]
[n]RTC : [LSE]
[p]Transmit Total Time Set: [Not Use]
[q]Low Level Noise Filter : [Not Use]
[r]AES KEY : [Not Use]
EEPROM Data :
0A 00 06 AE 01 12 34 00 02 00 18 00 00 00 00 2B
7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C 01
00 00 00 00 01 01 00 00 00 00 0B E8 03 00 00 00
03 E8 00 00 00 00 00 0A 00 00 00 0A 01 01 00
00 0D 36 DF 3D C0 00 01 C2 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 C3 50 01 00 00 02 01 00 01
*****
Please input >[]
```



# 5.開発ボードの設定

## 5.1 RM-900EV Rev.200との接続

<手順6>の説明  
RM-900EVとIAR社のi-Jetを接続します。

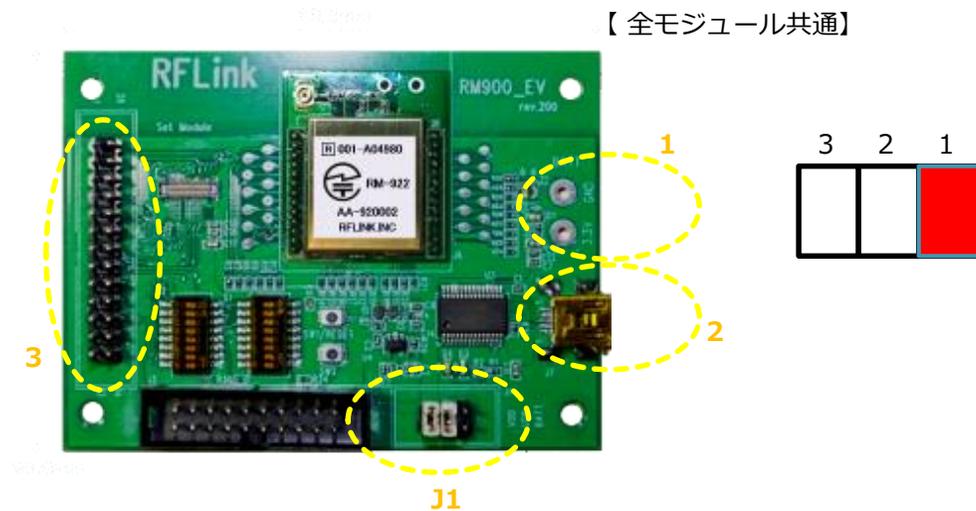




# 5.開発ボードの設定

## 5.3 RM-900EV Rev.200 J1 の設定方法

◆RM-900EVは、電源の供給元に応じて、J1の設定を切り替えて使用します。

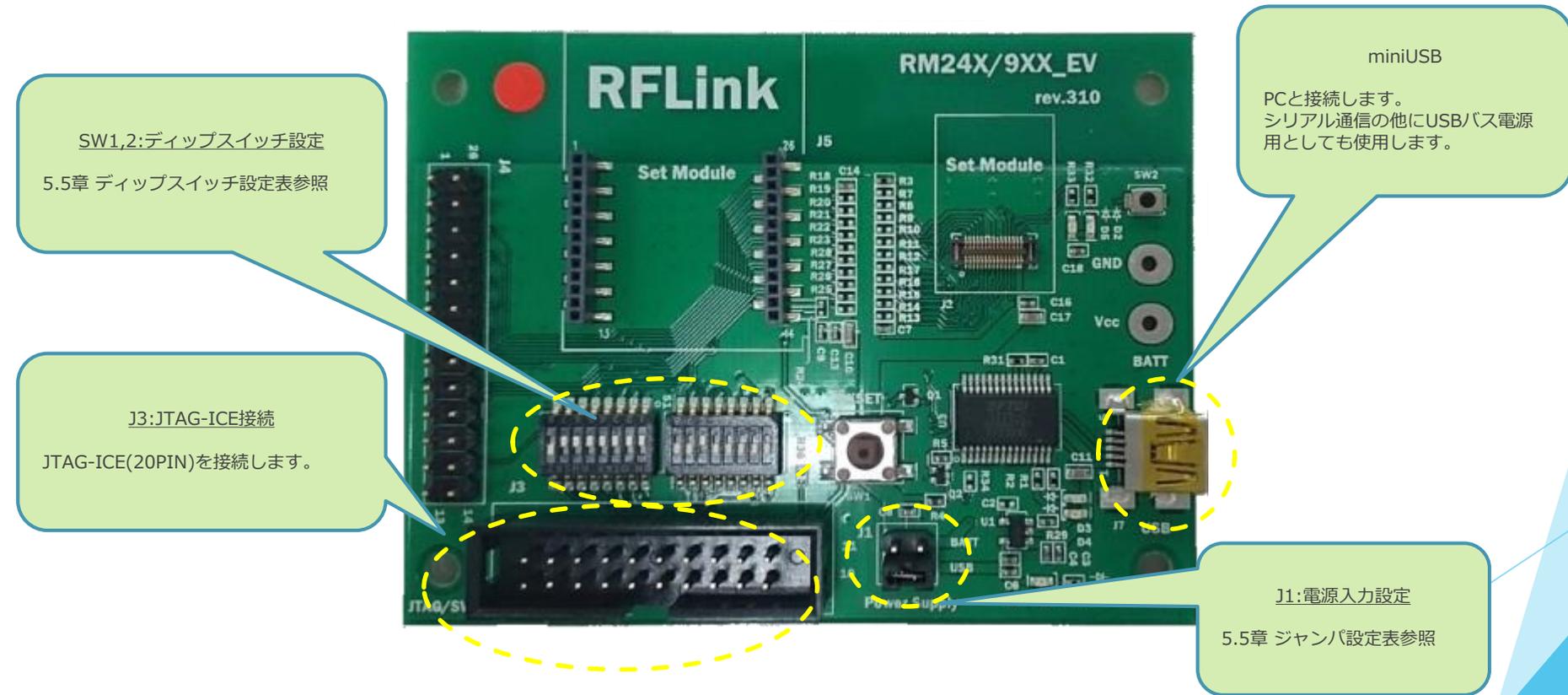


# 5.開発ボードの設定

## 5.4 RM-24X/9XX EV Rev.310との接続

<手順6>の説明

RM-24X/9XX\_EV Ver310とJTAG-ICEを接続します。(接続はJTAG接続になります) (RM-92Cは使用できません)



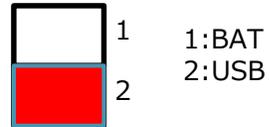
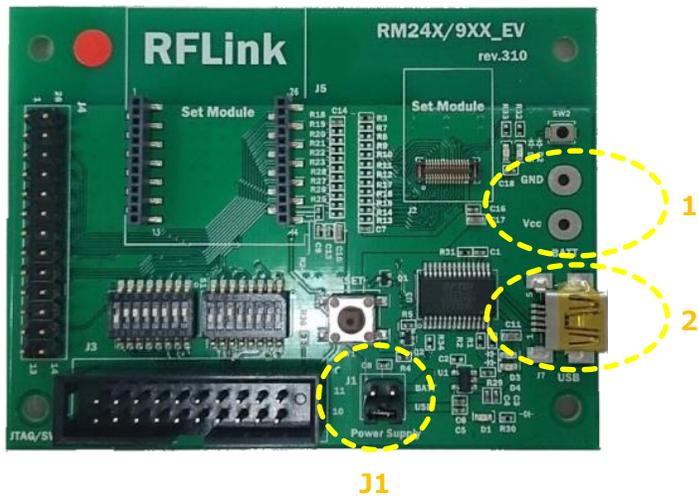


# 5.開発ボードの設定

## 5.6 RM-24X/9XX EV Rev.310 J5 の設定方法

◆RM-24X/9XX\_EV Ver310は、電源の供給元に応じて、J1の設定を切り替えて使用します。

【全モジュール共通】



# 5.開発ボードの設定

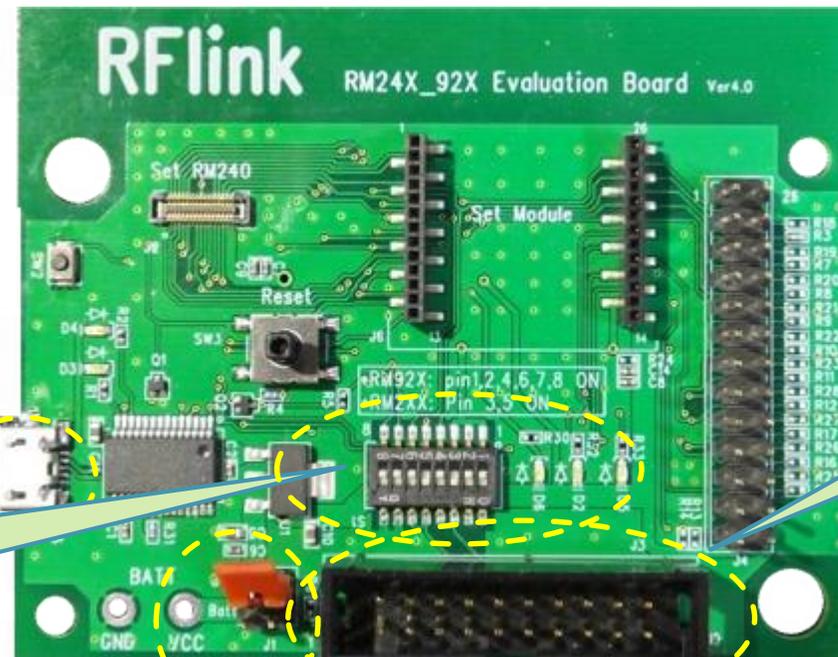
## 5.7 RM-24X/92X Rev.4.0との接続

<手順6>の説明

RM-24X/92X\_EV Ver4.0 とJTAG-ICEを接続します。(接続はSWD接続です)

マイクロUSB

PCと接続します。  
シリアル通信の他にも  
USBバス電源用としても  
使用します。



J13:JTAG-ICE接続

JTAG-ICE(20PIN)を接続します。

SW1:ディップスイッチ設定

5.8章 ディップスイッチ設定表参照

J1:電源入力設定

5.8章 ジャンパ設定表参照

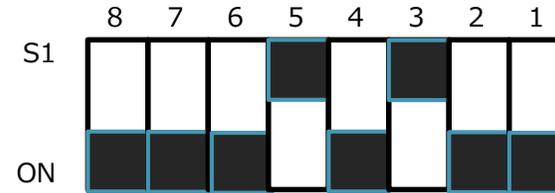
# 5.開発ボードの設定

## 5.8 RM-24X/92X Ver Rev.4.0 SW1 の設定方法

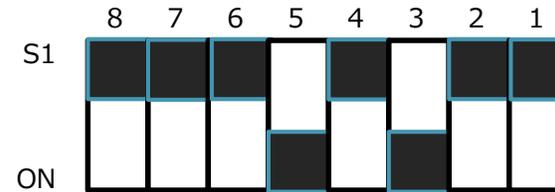
◆RM-24X/92X\_EV Ver4.0は、920MHz通信モジュール(RM-92A/RM-92C)と、2.4GHz通信モジュール(RM-240/241)と共通に使用する事が出来ます。使用するモジュールに応じて、SW2のディップスイッチを切り替えて使用します。



◆RM-92A/92C(920MHz)モジュールで使用する場合



◆RM-240/241(2.4GHz)モジュールで使用する場合



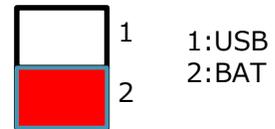


# 5.開発ボードの設定

## 5.9 RM-24X/92X EV Rev.4.0 J1 の設定方法

◆RM-24X/92X\_EV Ver4.0 は、電源の供給元に応じて、J1の設定を切り替えて使用します。

【全モジュール共通】



# 5.開発ボードの設定

## 5.10 RM-24X/92X Rev.4.1との接続

<手順6>の説明

RM-24X/92X\_EV Ver4.1 とJTAG-ICEを接続します。(接続はSWD接続です)

マイクロUSB

PCと接続します。  
シリアル通信の他にも  
USBバス電源用としても  
使用します。



SW1,2:ディップスイッチ設定

5.11章 ディップスイッチ設定表参照

J13:JTAG-ICE接続

JTAG-ICE(20PIN)を接続します。

J11:電源入力設定

5.11章 ジャンパ設定表参照

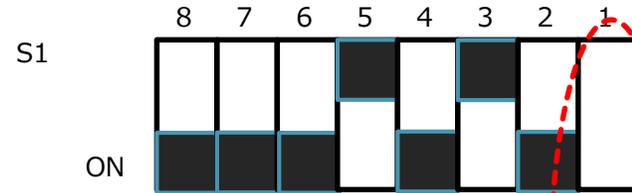
# 5.開発ボードの設定

## 5.11 RM-24X/92X Ver Rev.4.1 SW1 の設定方法

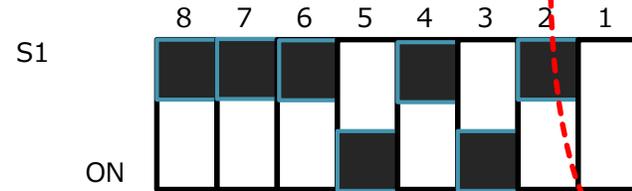
- ◆RM-24X/92X\_EV Ver4.1は、920MHz通信モジュール(RM-92A/RM-92C)と、2.4GHz通信モジュール(RM-240/241)と共通に使用する事が出来ます。使用するモジュールに応じて、SW2のディップスイッチを切り替えて使用します。プログラムの自動スタート機能が利用できます。デバッグI/Fは、SWD接続のみとなります。



- ◆RM-92A/92C(920MHz)モジュールで使用する場合(有効ビット1~7)



- ◆RM-240/241(2.4GHz)モジュールで使用する場合(有効ビット1~7)



自動スタート設定

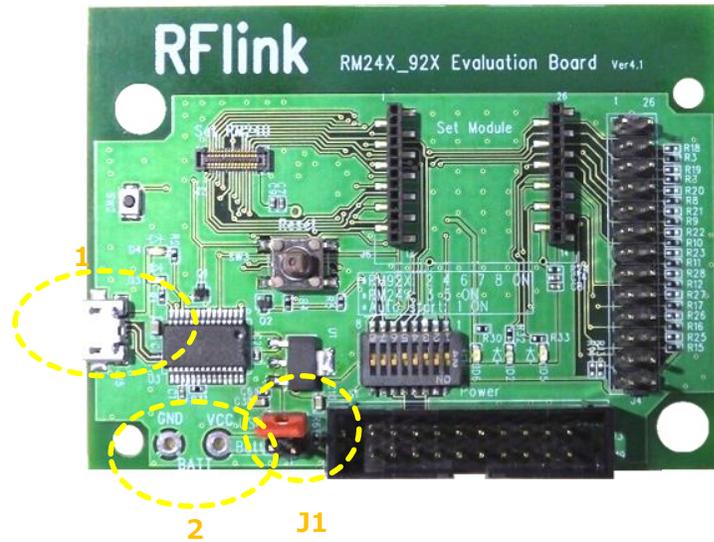
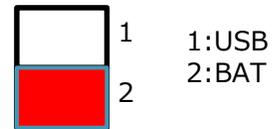
ON : 電源投入後・自動スタート  
OFF : 電源投入後・手動スタート

# 5.開発ボードの設定

## 5.12 RM-24X/92X EV Rev.4.1 J1 の設定方法

◆RM-24X/92X\_EV Ver4.1 は、電源の供給元に応じて、J1の設定を切り替えて使用します。

【全モジュール共通】



# 5.開発ボードの設定

## 5.13 RM-24X/92X Rev.4.2との接続

<手順6>の説明

RM-24X/92X\_EV Ver4.1 とJTAG-ICEを接続します。(接続はJTAG設定とSWD設定が可能です)



マイクロUSB

PCと接続します。  
シリアル通信の他にも  
USBバス電源用としても  
使用します。

SW2:ディップスイッチ設定

5.14章ディップスイッチ設定表参照

SW1:ディップスイッチ設定

接続するICEのタイプ合わせ  
て  
設定します(JTAG/SWD)  
4.17ディップSW設定表参照

J3:JTAG/SWD-ICE接続

JTAG-ICE(20PIN)を接続します。

J6:電源入力設定

5.14章 ジャンパ設定表参照

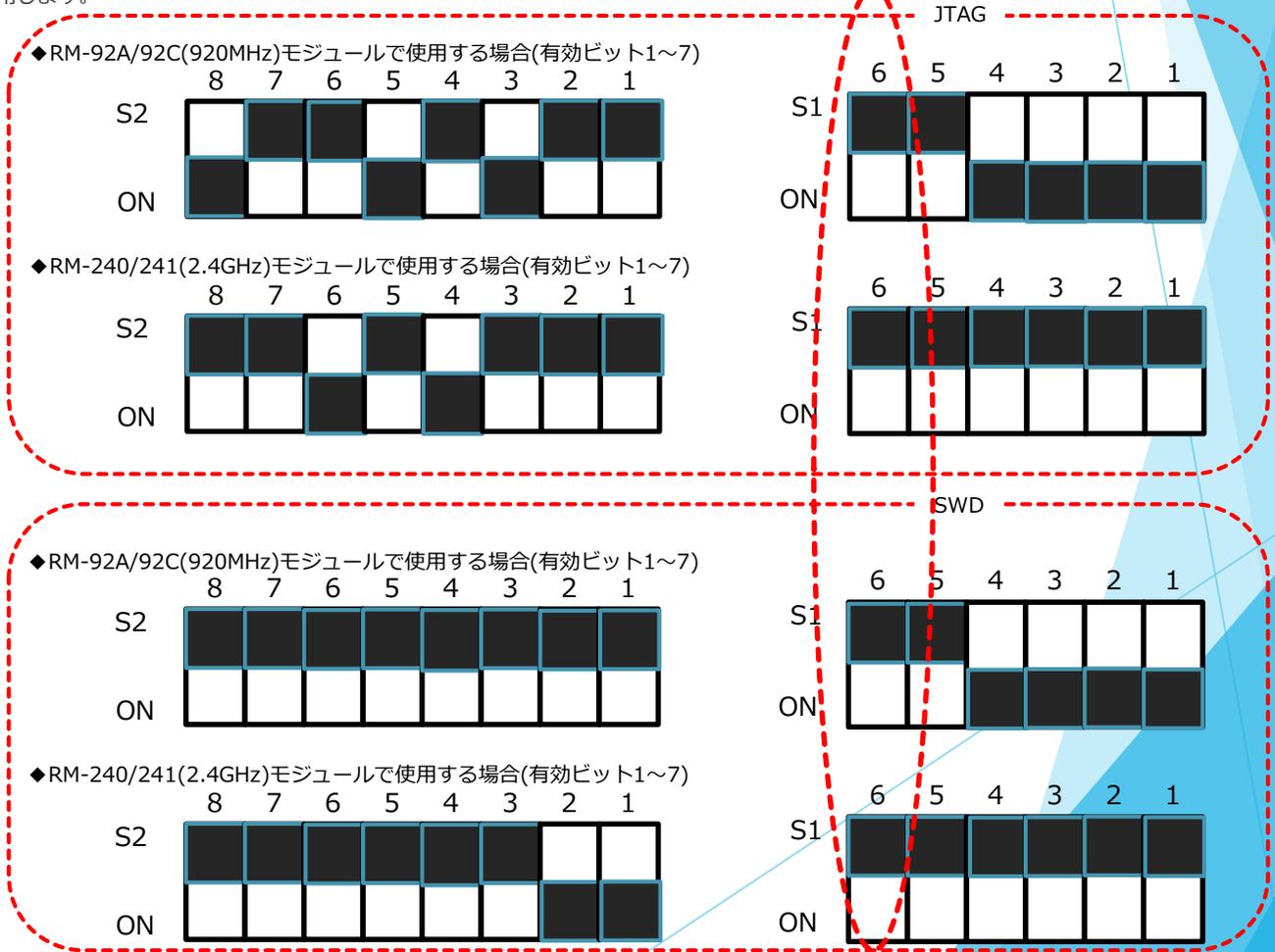
# 5.開発ボードの設定

## 5.14 RM-24X/92X Ver Rev.4.2 SW1、SW2 の設定方法

自動スタート設定  
 ON : 電源投入後・自動スタート  
 OFF : 電源投入後・手動スタート

◆RM-24X/92X\_EV Ver4.2は、920MHz通信モジュール(RM-92A/RM-92C)と、2.4GHz通信モジュール(RM-240/241)と共通に使用する事が出来ます。使用するモジュールに応じて、SW2のディップスイッチを切り替えて使用します。

■部分が、可動SW部です

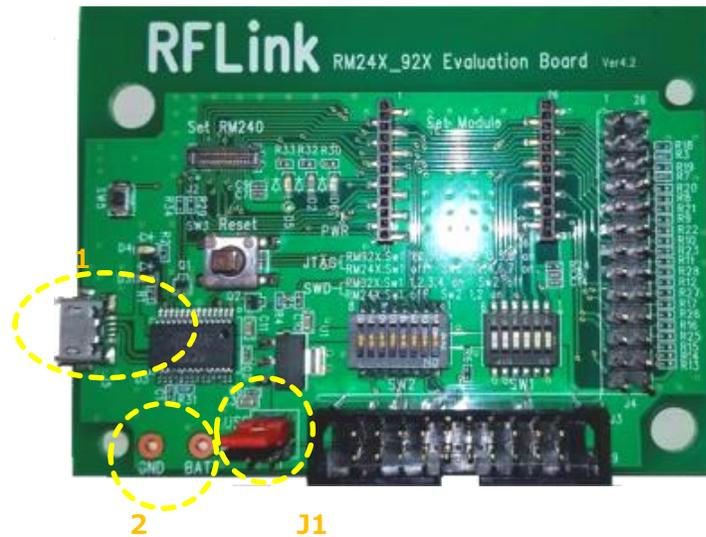
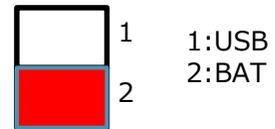


# 5.開発ボードの設定

## 5.15 RM-24X/92X EV Rev.4.2 J1 の設定方法

◆RM-24X/92X\_EV Ver4.2 は、電源の供給元に応じて、J1の設定を切り替えて使用します。

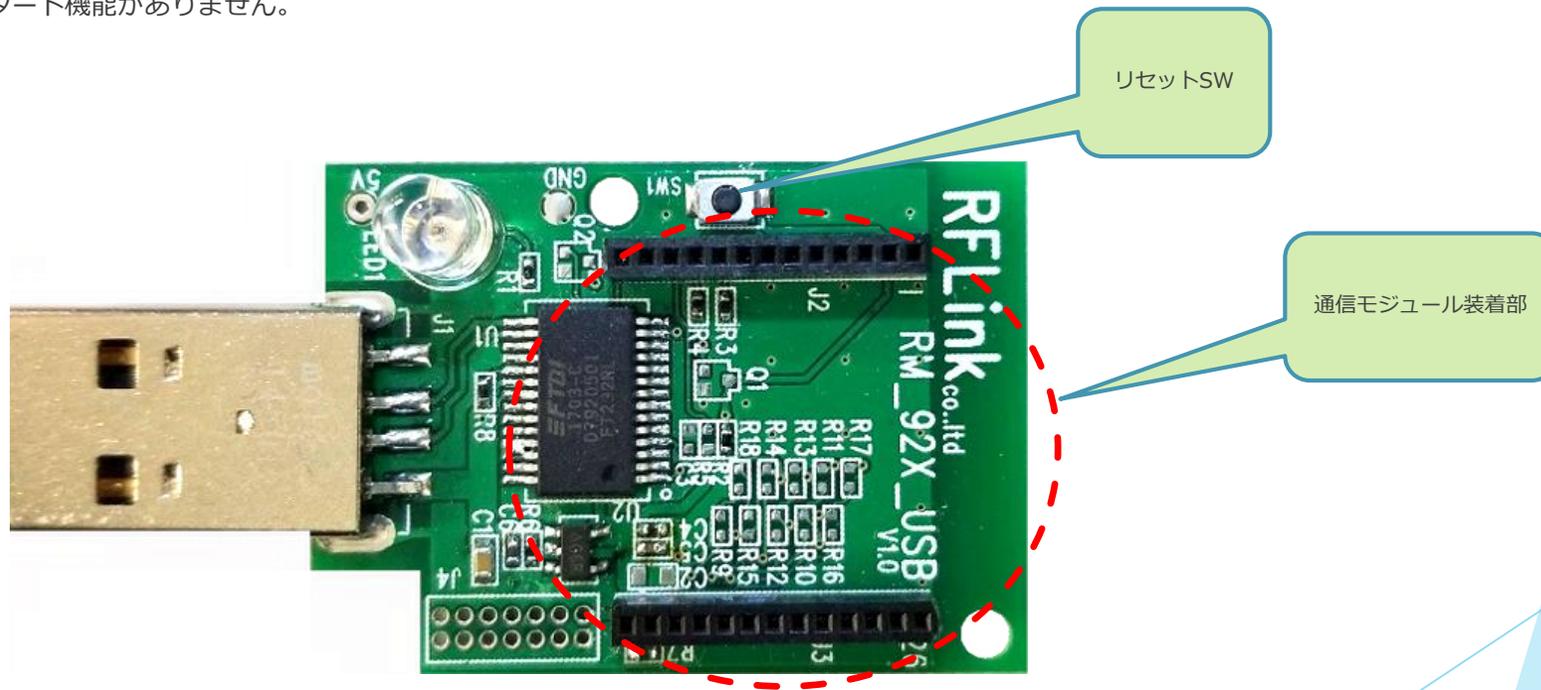
【全モジュール共通】



# 5.開発ボードの設定

## 5.16 RM-92XUSBを用いた開発方法

◆RM-92XUSBは、RM-92Xシリーズ(RM-92A/RM-92C)を装着してPCなどとUSB接続して使用します。  
開発ボードと同様に、FTDIドライバのインストールが必要です。正常に接続されるとCOMポートとして認識されますので、teraterm等のシリアルソフトウェアで設定が可能です。  
※本ボードには自動スタート機能がありません。





# 開発環境構築ガイド

◆ Release version

Version 2.8.0 2018-01-10