

---

# R-IN32M3 Module (RY9012A0)

R30AN0401JJ0102

Rev.1.02

## ファームウェア更新ガイド

---

2023.12.15

### 要旨

本書は、R-IN32M3 Module (RY9012A0)のファームウェア更新方法について説明します。

### 動作確認デバイス

R-IN32M3 Module (RY9012A0)

## 目次

1. 概要	5
1.1 概要	5
1.2 動作環境	5
1.2.1 ハードウェア環境	5
1.2.2 ソフトウェア環境	6
1.2.3 対象プロジェクト	7
2. R-IN32M3 Module ファームウェア更新	8
2.1 ファームウェア更新方法	8
2.1.1 更新方法	8
2.1.2 ファームウェア更新の流れ	9
2.1.3 更新パッケージ	10
2.2 Management Tool による更新手順	11
2.2.1 デバイスの検出	11
2.2.2 ファームウェア更新	14
2.2.3 ファームウェアバージョン確認	15
2.3 その他のファームウェア更新手順	16
2.3.1 HTTP	16
2.3.1.1 HTTP 更新概要	16
2.3.1.2 手順	16
2.3.1.3 複数モジュール更新	18
2.3.1.4 バッチファイル(参考)	19
2.3.2 FoE	20
2.3.2.1 FoE 更新概要	20
2.3.2.2 事前準備	20
2.3.2.3 手順	21
2.3.3 SPI	23
2.3.3.1 SPI 更新概要	23
2.3.3.2 手順	24
2.3.3.3 バッチファイル (参考)	27
3. ホストマイコン・ファームウェア更新	28
3.1 RA マイコン	28
3.1.1 ツール	28
3.1.2 ファームウェア更新準備	30
3.1.2.1 Bootloader プログラム	30
3.1.2.2 アップデート (ユーザプログラム)	32
3.2 RX マイコン	34
3.2.1 ツール	34
3.2.2 ファームウェア更新準備	35
3.2.2.1 Bootloader プログラム	35
3.2.2.2 更新用ユーザプログラムプログラム生成	37
3.2.2.3 アップデート・ユーザプログラム	39

---

改訂記録 .....41

## 用語解説

本書で使用する用語は、以下に示すように定義して使用します。

用語	説明
API	Application Programming Interface
GOAL	Generic Open Abstraction Layer 詳細は、『R-IN32M3 Module ユーザーズマニュアル ソフトウェア編 API 説明 (R17US0002JJ****)』を参照
uGOAL	Micro Generic Open Abstraction Layer, simplified and shrunk memory usage software package of GOAL
HTTP	Hyper-Text Transfer Protocol
FoE	Ethernet over EtherCAT
SPI	Serial Peripheral Interface

## 関連文書

資料名	資料番号
R-IN32M3 Module (RY9012A0) データシート	R19DS0109JJ****
R-IN32M3 Module (RY9012A0) ユーザーズマニュアル ハードウェア編	R19UH0122JJ****
R-IN32M3 Module (RY9012A0) ユーザーズマニュアル ソフトウェア編	R17US0002JJ****
R-IN32M3 Module (RY9012A0) クイックスタートガイド	R12QS0042JJ****
R-IN32M3 Module (RY9012A0) ユーザ実装ガイド	R30AN0386JJ****
R-IN32M3 Module 搭載アダプタボード YCONNECT-IT-I-RJ4501 ユーザーズマニュアル	R12UZ0094JJ****
RA6M3/RA6M4 Sample application	R30AN0398JJ****
Management Tool Instruction Guide	R30AN0390JJ****
ソフトウェアPLC接続ガイド TwinCAT	R30AN0380JJ****

## 1. 概要

### 1.1 概要

本書は、ルネサスエレクトロニクス製 産業イーサネット通信モジュール R-IN32M3 Module (RY9012A0) または、ホストマイコンのファームウェアプログラムを更新する方法について説明します。

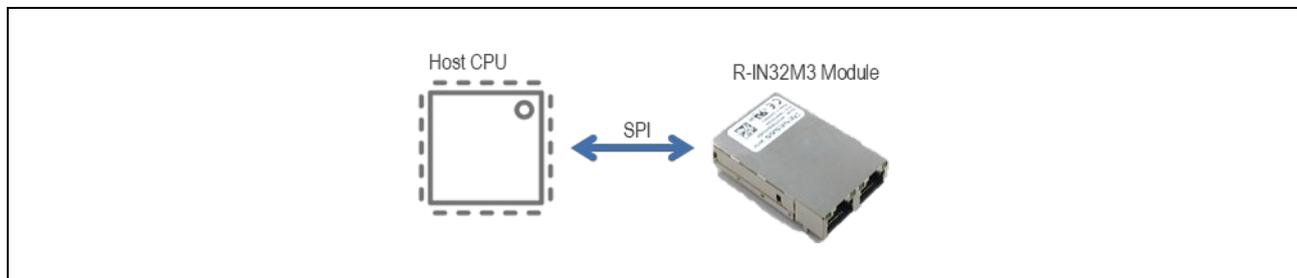


図 1-1 ファームウェアプログラム更新

### 1.2 動作環境

ファームウェア更新を行う動作環境について説明いたします。

ホストマイコンのファームウェア更新を行うサンプルソフトは、プラットフォームにより異なります。また、RL 向けサンプルソフトではホストマイコンのファームウェア更新をサポートしていません。

#### 1.2.1 ハードウェア環境

本書で紹介するファームウェア更新機能は、下記ハードウェア構成にて確認を行っておりますが、本書では、RA6M4 MCU グループ評価キット(EK-RA6M4)に R-IN32M3 Module 搭載アダプタボードを接続したハードウェア環境(1)をつかって説明いたします。

- (1) R-IN32M3 モジュール搭載アダプタボードと EK-RA6M3 または EK-RA6M4 との組合せ
- (2) R-IN32M3 モジュール搭載 アダプタボードと RL78/G14 (RTK5RLG140C00000BJ)との組合せ
- (3) R-IN32M3 モジュール搭載 CPU カード

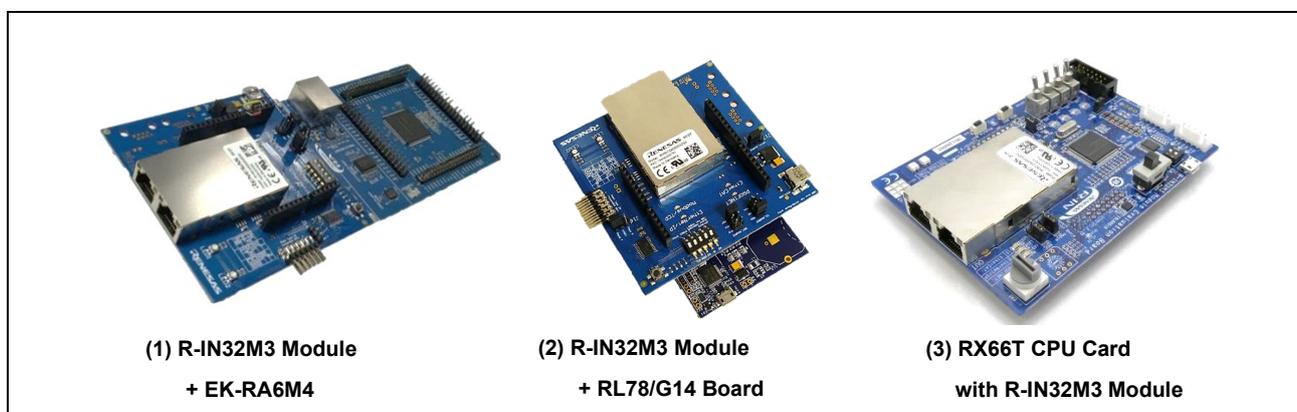


図 1-2 R-IN32M3 モジュール 開発環境

表 1-1 ハードウェア環境

Name	Type Name	Maker	Link	Note
Adapter Board with R-IN32M3 Module	YCONNECT-IT-I-RJ4501	Renesas Electronics Corporation	<a href="#">R-IN32M3-Module-Solution-Kit</a>	
EK-RA6M3	RTK7EKA6M3S00001BU	Renesas Electronics Corporation	<a href="#">RA6M3 MCU Group Evaluation Board</a>	
EK-RA6M4	RTK7EKA6M4S00001BE	Renesas Electronics Corporation	<a href="#">Evaluation Kit for RA6M4 MCU Group</a>	
RL78/G14 Fast Prototyping Board	RTK5RLG140C00000BJ	Renesas Electronics Corporation	<a href="#">RL78/G14 Fast Prototyping Board</a>	
RX66T CPU Card with R-IN32M3 Module	SEMB1320	SHIMAFUJI Electric Incorporated	<a href="https://www.renesas.com/S/EMB1320">https://www.renesas.com/S/EMB1320</a>	

### 1.2.2 ソフトウェア環境

ファームウェア更新を行うサンプルソフトの動作環境を表 1-2 に示します。詳細は、サンプルパッケージに含まれる各ハードウェア環境のアプリケーションノートをご参照ください。

各ソフトウェアは Windows10 (64bit) 環境で確認を行っております。

表 1-2 動作環境

Category	Name	Version	Link	備考
R-IN32M3 モジュール サンプルパッケージ	サンプルパッケージ	—	r18an0064xx01**.zip	<a href="#">製品ページ</a> より最新版をDLしてください
RA ファミリ用 Flexible Software Package	FSP	V4.4.0	<a href="https://github.com/renesas/">github.com/renesas/</a>	
統合開発環境	e2studio	2023-04	-	上記 Link にあるインストーラに FSP と一緒に同梱
GNU Arm Embedded Toolchain	GCC Toolchain	V9.3.1.20200408	-	上記 Link にあるインストーラに FSP と一緒に同梱
Management Tool・簡易 PLC	ICE	V1.3.1 以降	-	port industrial automation GmbH 社製 サンプルパッケージに同梱
ソフトウェア PLC	TwinCAT	V3.1	<a href="https://www.beckhoff.com/">https://www.beckhoff.com/</a>	Beckhoff Automation 社製
RA マイコン ファームウェア更新	Python	V3.8.3 以降	<a href="https://www.python.org/">https://www.python.org/</a>	RA マイコンのファームウェア更新機能の署名データ埋め込みに Python スクリプトを使用
RX マイコン ファームウェア更新	Renesas Secure Flash Programmer	V2.0.2	<a href="#">Renesas Flash Programmer</a>	RX マイコンのファームウェア生成ツール サンプルパッケージに同梱

### 1.2.3 対象プロジェクト

ファームウェアプログラム更新をサポートしているサンプルプロジェクトを示します。

Sample software	Update target	
	R-IN32M3 module	Host MCU
01_pnio	✓	-
02_eip	✓	-
03_ecat	✓	-
04_pnio_largesize	-	-
05_eip_largesize	-	-
06_ecat_largesie	-	-
07_mbus_tcp_server	-	-
10_multi_protocol	-	-
11_pnio_http	-	✓
12_eip_http	-	✓
13_ecat_http	-	✓
17_fwup_bootloader	-	✓

#### [R-IN32M3 Module \(RY9012A0\)のファームウェアプログラム更新](#)

2.3.1 章の HTTP および 2.3.2 章の FoE によるファームウェア更新機能は、サンプルパッケージに含まれる各ハードウェア環境のサンプルアプリケーションに実装されています。

2.3.3 章の SPI 方式については、RA6 用の 01\_pnio, 02\_eip, 03\_ecat プロジェクトにのみ実装されています。

#### [ホストマイコンのファームウェアプログラム更新](#)

ホストマイコンのファームウェア更新をイーサネット(R-IN32M3 Module)経由で実行します。ファームウェア更新機能を実装する場合は、あらかじめ専用の Bootloader をプログラムする必要があります。

## 2. R-IN32M3 Module ファームウェア更新

R-IN32M3 Module のファームウェア更新について説明します。

### 2.1 ファームウェア更新方法

#### 2.1.1 更新方法

R-IN32M3 Module のファームウェア更新を行う手段を大別すると、

- (1) Management Tool (ICE.exe) を使用する手段
- (2) Management Tool (ICE.exe) を使用しない手段

の 2 つがあります。

(1)の Management Tool を使用する手段の場合は、更新パッケージファイルの解凍から、転送、コミットまで一連の手順をツールが自動で行います。

(2)の Management Tool を使用しない場合は、対応しているファームウェア更新時のデータ転送方式によって、HTTP、FoE、SPI による 3 つの方法があります (表 2-1)

表 2-1 ファームウェア更新方式

方式	説明	ファームウェアのコミット設定
HTTP	HTTP-POST 要求を使用したファームウェア更新方法。 Management Tool を使用した手順は 2.2 章、curl コマンドを使った手順は 2.3.1 章を参照。	デフォルト設定ではコミットが必要なため、ファームウェアの更新後に特定の URL を要求する必要があります。 Management Tool を使用する場合はツール内でコミット処理まで実行します。
FoE	EtherCAT FoE 転送を使用したファームウェア更新方法。TwinCAT を使用した手順については 2.3.2 章を参照。	更新されたファームウェアの起動後、自動的にコミットされます。
SPI	アプリケーションコントローラ(AC)による RPC 機能を使用したファームウェア更新方法。ホストマイコンから SPI 経由でファームウェアイメージを転送します。2.3.3 章を参照。	更新されたファームウェアの起動後、自動的にコミットされます。

### 2.1.2 ファームウェア更新の流れ

R-IN32M3 モジュールには、ブートローダとアプリケーション の2つのファームウェアオブジェクトが存在します。ブートローダはユーザによって更新することはできず不変で、基本のスタートアップコード、ファームウェアのフラッシュルーチンとファームウェア更新の署名検証コードが含まれています。

提供されている各ファームウェア更新方法に共通する基本的な流れは以下となります。

- ① 実行中のアプリケーションは、ファームウェアのアップデートデータをモジュールのRAMに保存します。
- ② ファームウェア更新データを受信したら、実行中のアプリケーションの再起動を行います。このステップは、アプリケーション・コントローラで制限することができます。
- ③ ブートローダが起動し、保留中のファームウェア更新を検出します
- ④ ブートローダはファームウェア署名を確認し、パスするとファームウェアをフラッシュROMに保存します。この時点では以前に実行されていたファームウェアは保持されています。
- ⑤ ブートローダは新しいファームウェアを起動します
- ⑥ コミット設定がされている場合(デフォルト設定)、ブート中に実行中のファームウェアをコミットします。それ以外の場合は、コミットプロセスは外部からトリガされます

コミットの実行に応じて、ブートローダは新しいファームウェア、もしくは、以前に実行されたファームウェアを起動します。

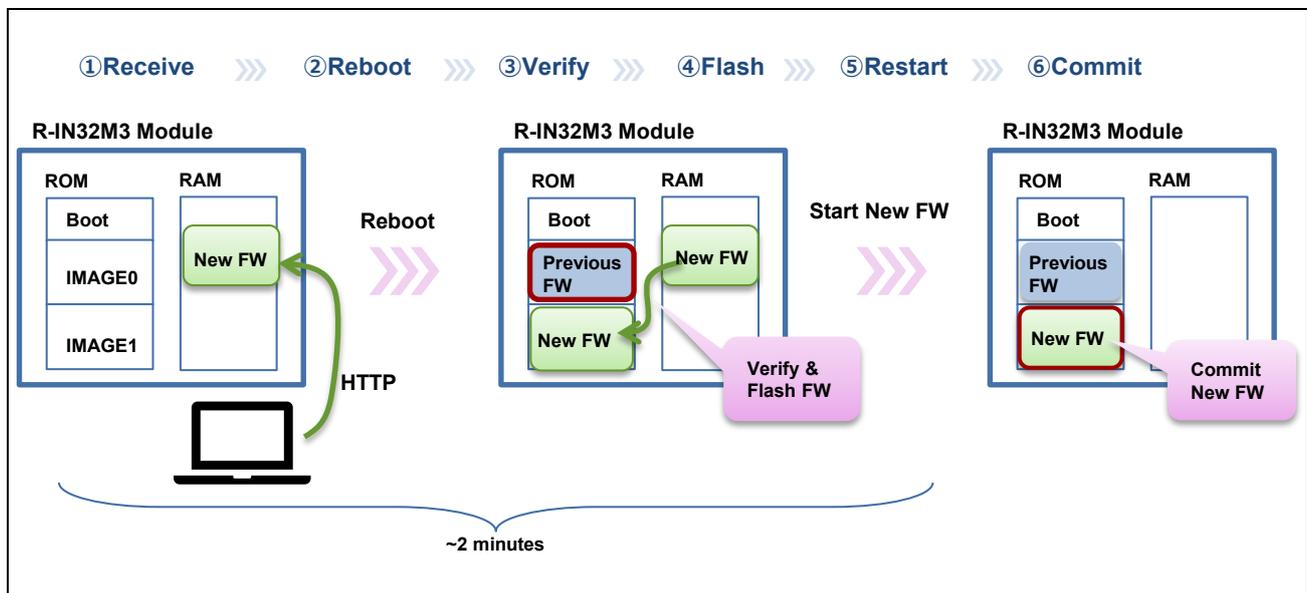


図 2-1 ファームウェア更新イメージ (HTTP)

### 2.1.3 更新パッケージ

ファームウェアはファームウェアパッケージとして提供されます。例えば、『irj45\_2.1.0.0\_ci37.pfw』といったファイル名で、基本的には表 2-2 のファイルを含む zip アーカイブです。

ファームウェアパッケージを解凍すると、/bundle フォルダに表 2-2 のファームウェア更新ファイルがあります。

表 2-2 ファームウェア更新パッケージ内容

File (/bundle)	Description
bundle.xml	バンドル説明
irj45_2port.dat	HTTP 方式によるファームウェア更新ファイル
irj45_2port.bin	FoE/SPI 方式によるファームウェア更新ファイル

(Note) これらのファームウェア更新ファイルは署名されており、編集することを禁じます。

## 2.2 Management Tool による更新手順

Management Tool - ICE – を使って、R-IN32M3 モジュールのファームウェアを更新することができます。ファームウェアファイルは Ethernet 接続を介して R-IN32M3 モジュールへ送信されます。

Management Tool の詳細については、アプリケーションノート(Management Tool Instruction Guide [R30AN0390JJ\*\*\*\*]) を参照してください。

ホストマイコンが未接続の場合でも、R-IN32M3 モジュールは一般的な TCP/IP モードで起動しますので、ファームウェア更新を行うことができます。

### 2.2.1 デバイスの検出

EK-RA6M3 ボード、又は、EK-RA6M4 ボードに R-IN32M3 モジュール搭載アダプタボードを Arduino™ コネクタで接続した後、PC と LAN ケーブル、USB micro ケーブルを接続します(図 2-2)。USB micro B ケーブルを接続する事によって、ボードに電源が供給されます。

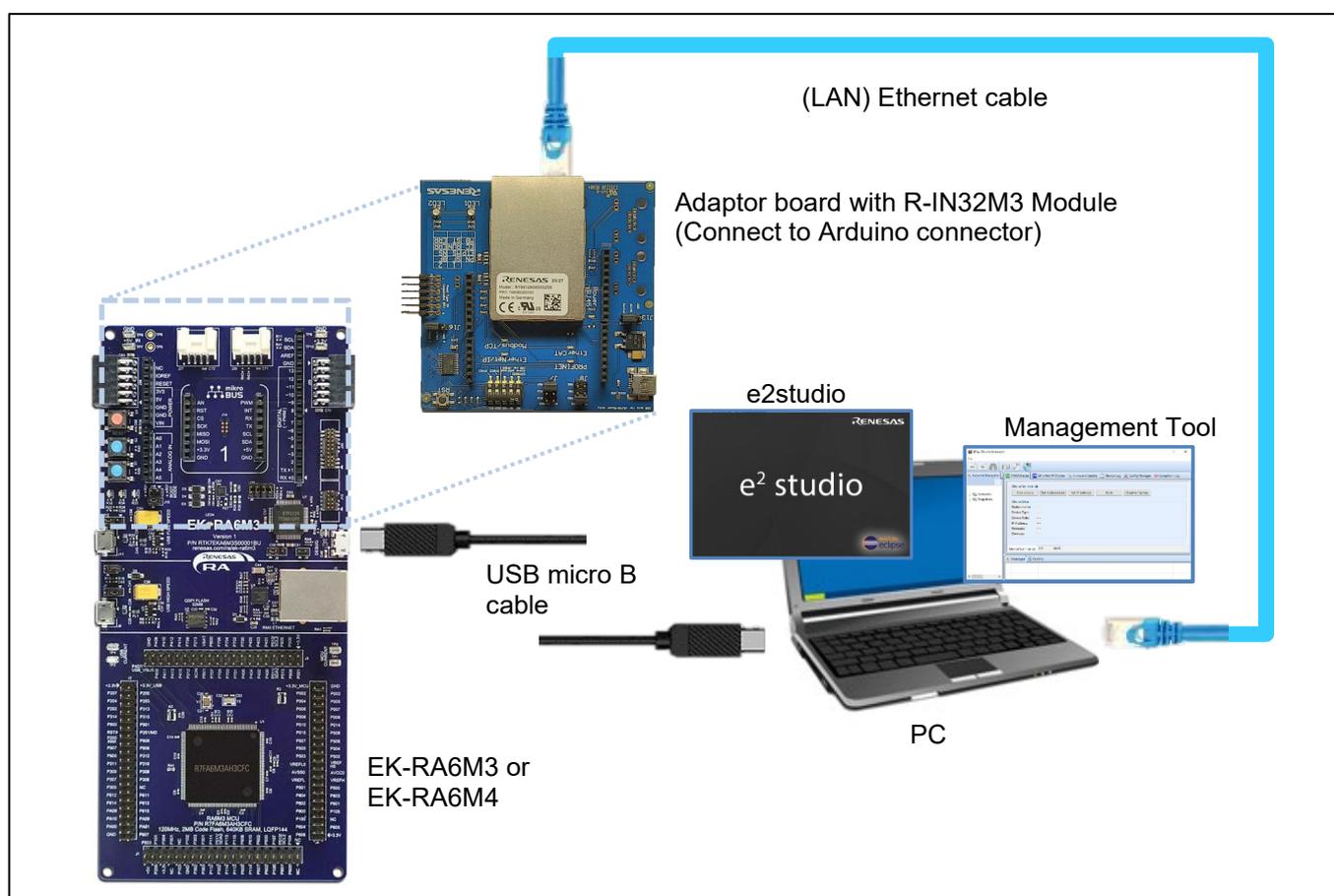


図 2-2 接続構成

ネットワークアダプタの“ネットワークプロパティ”を開き R-IN32M3 モジュールと接続する PC の IP アドレスを設定します。

	PC	R-IN32M3 Module (default)
IP address	192.168.0.1	192.168.0.100
Netmask	255.255.255.0	255.255.255.0

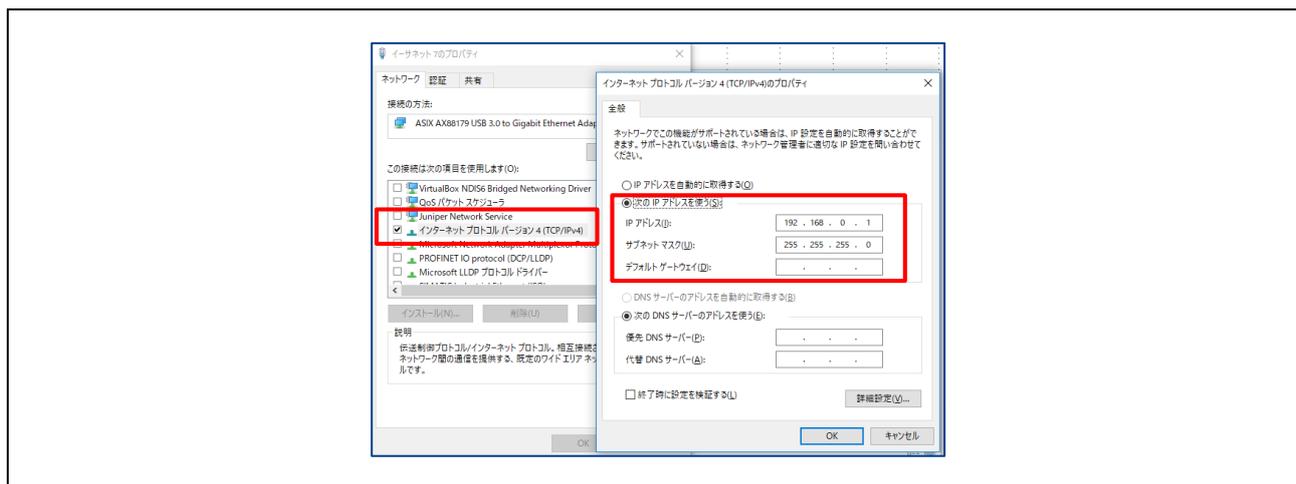


図 2-3 IP アドレス設定

R-IN32M3 モジュールと PC を接続し、Scan Network  を押して実行します。

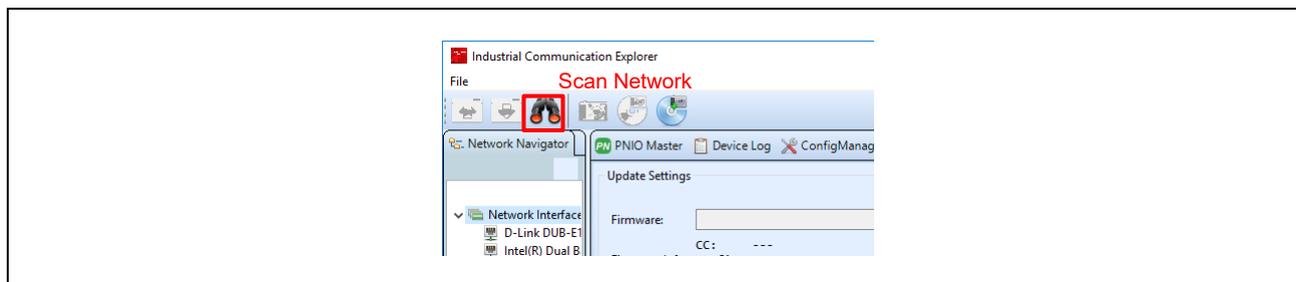


図 2-4 Scan Network

R-IN32M3 モジュールと通信するには、[Network Navigator]の「Networks」リストを開きます。次に、R-IN32M3 モジュールが接続可能なネットワークインタフェースを選択します。ツールバーにある[Scan Network]ボタンを選択してください。

次のようなダイアログが表示され、検出された 1 台のデバイスが報告されます。

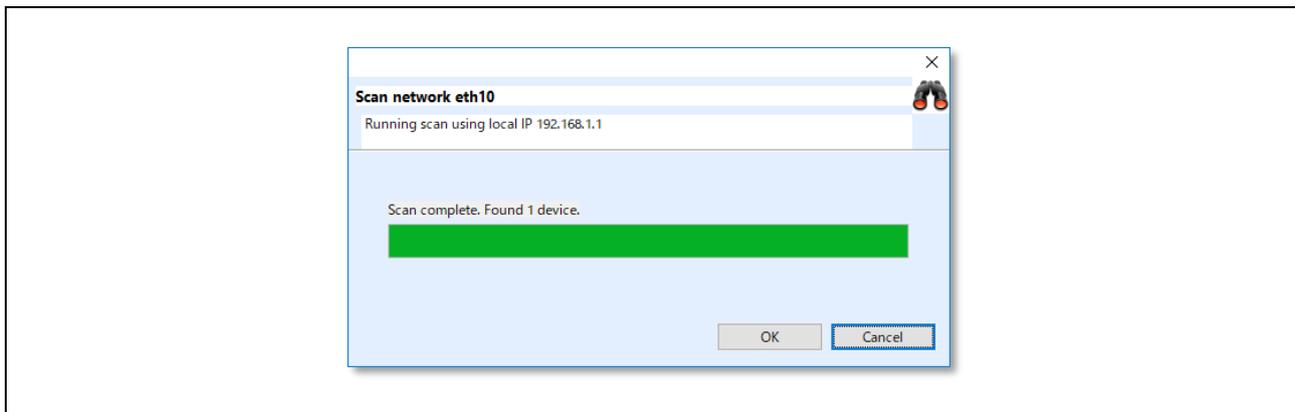


図 2-5 ネットワークスキャンダイアログ

その結果、R-IN32M3 モジュールは、スキャンされたネットワーク内の[Network Navigator]に、新しいデバイスとして表示されます。

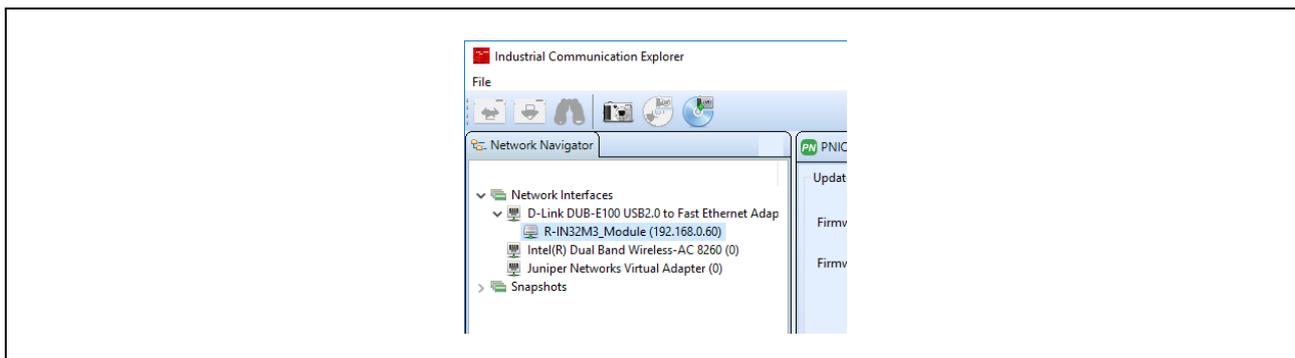


図 2-6 検出された R-IN32M3 モジュール

## 2.2.2 ファームウェア更新

"Firmware Update" タブを開きます。

"Select FW bundle" から、ファームウェアファイル lrij45\_\*\*\*\*.pfw を指定します。

"Start update" を押し、ファームウェア更新を行います。完了するまで2分程度かかります。Phase に "FW Update completed successfully" が表示されればファームウェア更新は全て完了したことを示します。

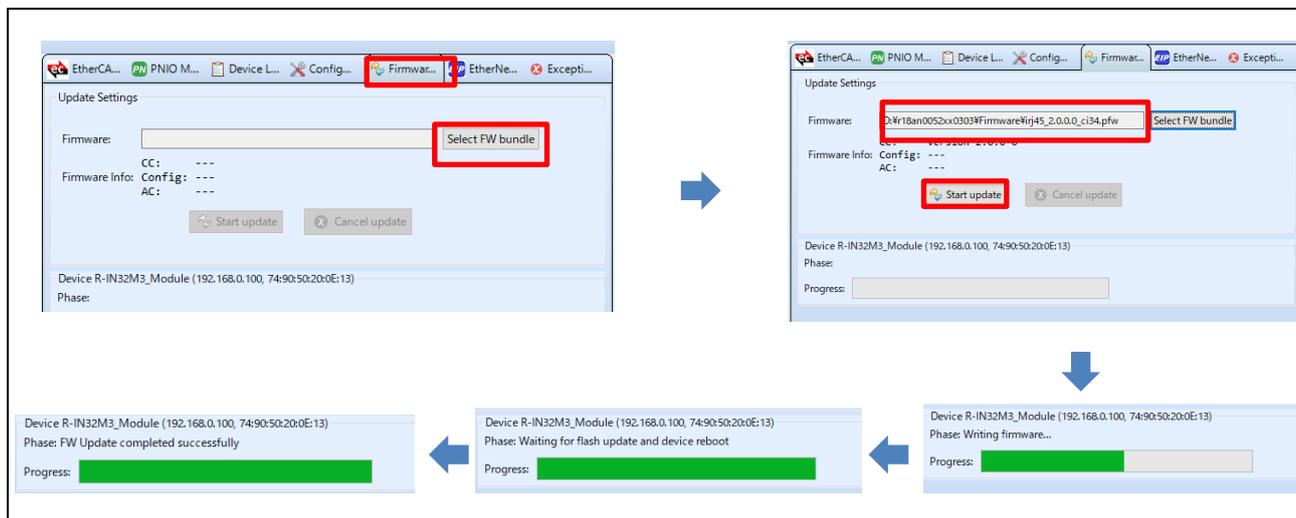


図 2-7 R-IN32M3 モジュールファームウェアの更新

### 2.2.3 ファームウェアバージョン確認

現在のファームウェアのバージョンを確認する際は、[ConfigureManager]機能パネルを選択した後、[Read configuration]ボタン を選択します(図 2-8)。

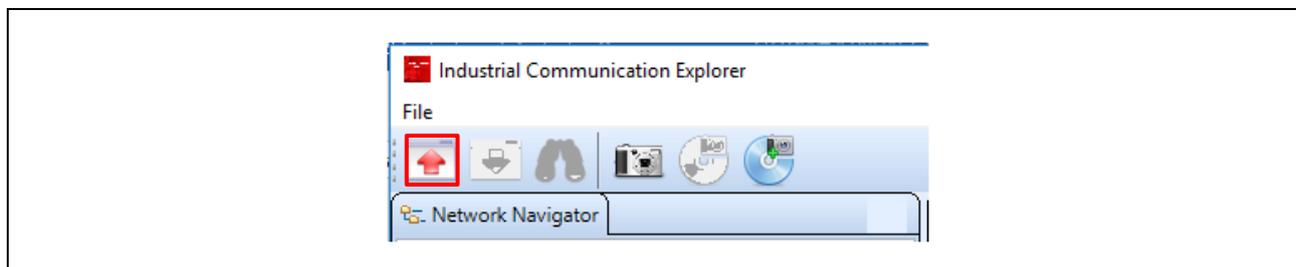


図 2-8 構成の読み出し

コンフィグ設定の"FWVERSION"の値が現在のファームウェアのバージョンになります。

Module	Variable	Action	Type	Temp	Read	Write	Value
GOAL_ID_BOOT	SIGNATURE		Generic	X	X	X	0xccb51c0d1e62c4fec86292d5085ebd46
GOAL_ID_BOOT	BLVERSION		String	X	X	X	1.3.0.0
GOAL_ID_BOOT	FWVERSION		String	X	X	X	2.1.0.0
GOAL_ID_BOOT	RESET_CAUSE		uint8	X	X	X	0x00
GOAL_ID_BOOT	IMAGE_NUMBER		uint8	X	X	X	0x01
GOAL_ID_BOOT	IMAGE_COUNTER		uint8	X	X	X	0xab

図 2-9 R-IN32M3 モジュールファームウェアのバージョン情報

関連する変数パラメータを表 2-3 に示します。

ファームウェアプログラムは、図 2-1 にありますように Flash ROM 内に 2 つのイメージを保存する領域が用意されており、コミットすることでブートするイメージを設定します。"IMAGE\_NUMBER" はブートするイメージ番号を示しており、ファームウェア更新時にコミットする度に切り替わります。

"IMAGE\_NUMBER" はファームウェアイメージの更新回数を表します。

表 2-3 Module Id = GOAL\_ID\_BOOT (37)

Variable Name	Variable ID	Type	Max. Size	Long description
SIGNATURE	0	GENERIC	16	Signature of booted image
BLVERSION	1	STRING	16	Bootloader Version
FWVERSION	2	STRING	16	Firmware Version
RESET_CAUSE	1000	UINT8	1	Reset cause:0, Unspecified1, Firmware Update Requested2, Watchdog3, Firmware Commit Required4, Reserved
IMAGE_NUMBER	1001	UINT8	1	Booted image number
IMAGE_COUNTER	1002	UINT8	1	Booted image age counter

## 2.3 その他のファームウェア更新手順

本章では、Management Tool を使用せずにファームウェアを更新する手順について説明します。

### 2.3.1 HTTP

#### 2.3.1.1 HTTP 更新概要

Windows10 では標準搭載されている curl コマンドを使って HTTP プロトコルでファームウェアを更新する手順を示します。

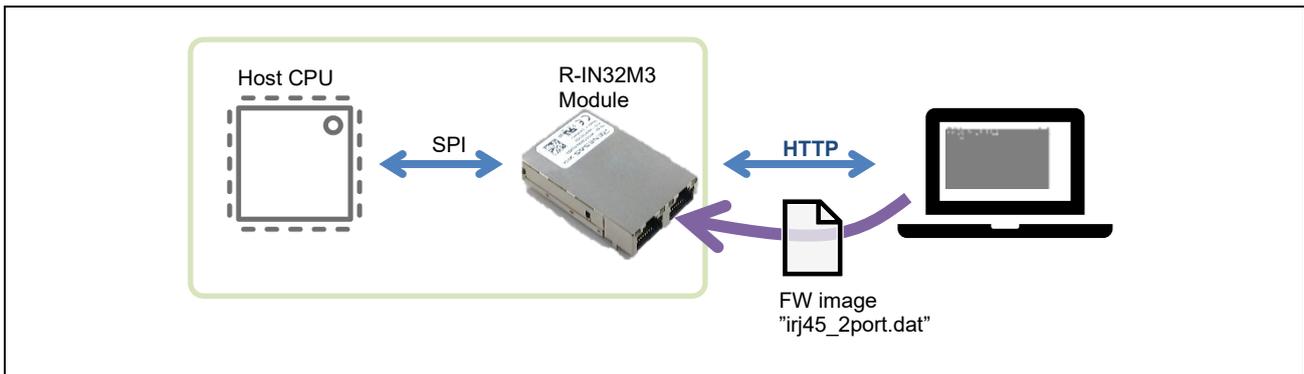


図 2-10 ファームウェアダウンロード(HTTP)

#### 2.3.1.2 手順

(1) プロジェクトをビルドし、RA6 ボードに書き込みます。

(2) コマンドプロンプトを起動し、任意のフォルダに移動します。

移動後のフォルダで、ファームウェア更新パッケージ (irj45\_2.1.0.0\_ci37.pfw 等)を配置し、zip 解凍します。(解凍すると得られる irj45\_2port.dat ファイルを書き込みます。)

(3) 以下のコマンドを実行します。(ここで、IP アドレスは R-IN32M3 モジュールに格納されているアドレスとしてください。以下の例では、192.168.0.100 としています。ユーザ名 : admin, パスワード : password で固定です。)

```
curl -v http://192.168.0.100:8080/fw/firmware.html -F file=@bundle/irj45_2port.dat --basic -u admin:password
```

```
C:\Work\IEM\test>curl -v http://192.168.0.100:8080/fw/firmware.html -F file=@bundle/irj45_2port.dat --basic -u admin:password
* Trying 192.168.0.100...
* TCP_NODELAY set
* Connected to 192.168.0.100 (192.168.0.100) port 8080 (#0)
* Server auth using Basic with user 'admin'
> POST /fw/firmware.html HTTP/1.1
Host: 192.168.0.100:8080
Authorization: Basic YWRtaWw46cGFzc3dvcmQ=
User-Agent: curl/7.55.1
Accept: */*
Content-Length: 1681849
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----0514caf0d53874f3
< HTTP/1.1 100 Continue
< Connection: close
< HTTP/1.1 204 No Content
< Connection: close
* Closing connection 0
```

図 2-11 成功時の Log

- (4) IEM 側の更新が終了するまで、2 分程度待ちます。(2.1.1 章の⑤完了まで)

(Note) 更新が完了するまで IEM は応答できません。ping 等で接続が復帰したのを確認したのち次へ進んでください。

- (5) 以下のコマンドを実行 (IP アドレスは IEM に格納されているアドレスとしてください)

```
curl -v http://192.168.0.100:8080/fw/commit.html -F commit=commit --basic -u admin:password
```

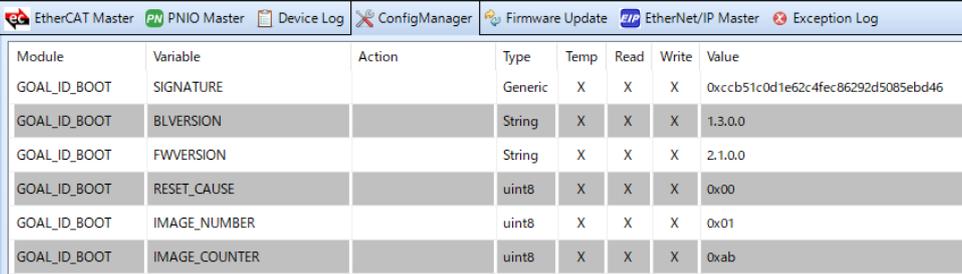
```
C:\Work\IEM\test>curl -v http://192.168.0.100:8080/fw/commit.html -F commit=commit --basic -u admin:password
* Trying 192.168.0.100...
* TCP_NODELAY set
* Connected to 192.168.0.100 (192.168.0.100) port 8080 (#0)
* Server auth using Basic with user 'admin'
> POST /fw/commit.html HTTP/1.1
Host: 192.168.0.100:8080
Authorization: Basic YWRtaW46cGFzc3dvcmQ=
User-Agent: curl/7.55.1
Accept: */*
Content-Length: 147
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----d4b3aa8067b97a3a

< HTTP/1.1 100 Continue
< Connection: close
< HTTP/1.1 204 No Content
< Connection: close
* Closing connection 0
```

図 2-12 成功時の Log

- (6) 再起動すると更新したファームウェアで起動します。

Management tool の ConfigManager で書き込まれている FW バージョン等を確認できます。(2.2.3 章)



Module	Variable	Action	Type	Temp	Read	Write	Value
GOAL_ID_BOOT	SIGNATURE		Generic	X	X	X	0xccb51c0d1e62c4fec86292d5085ebd46
GOAL_ID_BOOT	BLVERSION		String	X	X	X	1.3.0.0
GOAL_ID_BOOT	FWVERSION		String	X	X	X	2.1.0.0
GOAL_ID_BOOT	RESET_CAUSE		uint8	X	X	X	0x00
GOAL_ID_BOOT	IMAGE_NUMBER		uint8	X	X	X	0x01
GOAL_ID_BOOT	IMAGE_COUNTER		uint8	X	X	X	0xab

図 2-13 ファームウェアのバージョン情報

### 2.3.1.3 複数モジュール更新

スイッチングハブを使って接続することで、複数の R-IN32M3 Module のファームウェアを同時並行で行うことが可能です。図 2-14 は 2 台の R-IN32M3 Module のファームウェアを更新するためのネットワーク接続例です。PC からそれぞれの IP アドレスに対して、2.3.1.2 章のファームウェア更新手順でコマンドを発行します。

更新の過程で通信不可状態になるタイミングがあるため、ディジー接続での同時更新はできません。

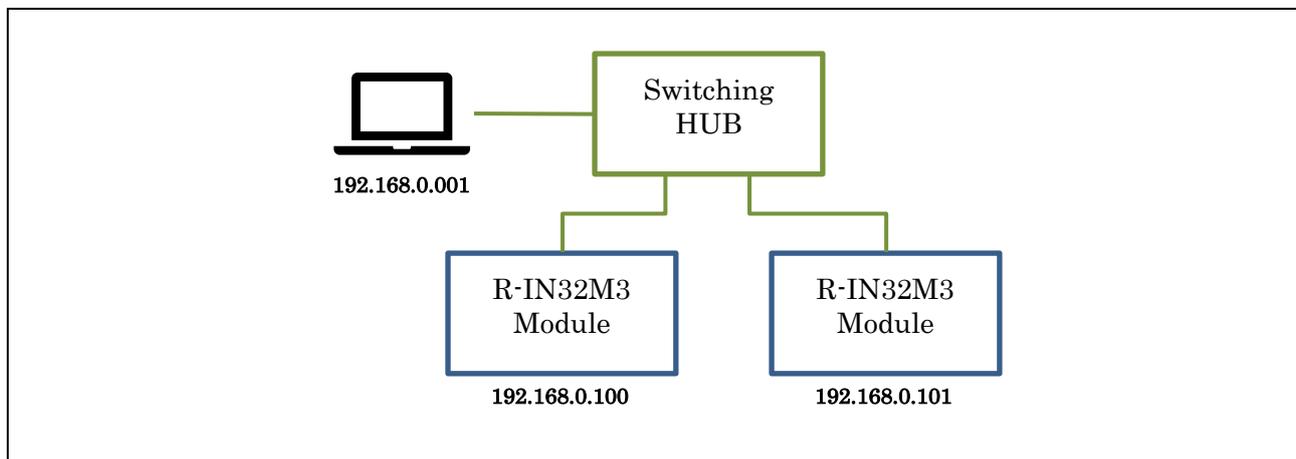


図 2-14 複数モジュール同期更新

### 2.3.1.4 バッチファイル(参考)

HTTP でファームウェア更新を行うためのバッチファイル例を示します。

更新ファイル irj45\_2port.dat と同じフォルダにバッチファイルを置き、引数として IP アドレスを指定して起動すると、対象に対してダウンロードを開始します。

1 つ目のコマンド発行後、ping で接続できるかを確認して、ping が通ったら 2 つ目のコマンドを発行してコミットを行います。 ping は最大 2 分まで待ち、接続できない場合はエラー終了します。

#### <バッチプログラム例>

```
@echo off

set count=0
ping -n 1 %1 | find "ms TTL=" > NUL
if not errorlevel 1 goto upload
echo host stopped or not found
goto end

:upload

@echo on
curl -v http://%1:8080/fw/firmware.html -F file=@irj45_2port.dat --basic -u admin:password

@echo off
echo R-IN32M3_Module connecting ...
:retry
set /a count=count+1
ping -n 1 -w 1000 %1 | find "ms TTL=" > NUL
if not errorlevel 1 goto commit
if "%count%" == "120" (
    echo timeout error
    exit /b
)
goto retry

:commit
@echo on
curl -v http://%1:8080/fw/commit.html -F commit=commit --basic -u admin:password
:end
```

## 2.3.2 FoE

## 2.3.2.1 FoE 更新概要

R-IN32M3 モジュールが EtherCAT で動作しているときは、HTTP は使えませんが、FoE によりファームウェアを更新することができます。本章では、TwinCAT3 を使用して EtherCAT の FoE 通信で R-IN32M3 モジュールのファームウェアを更新する手順について説明します。

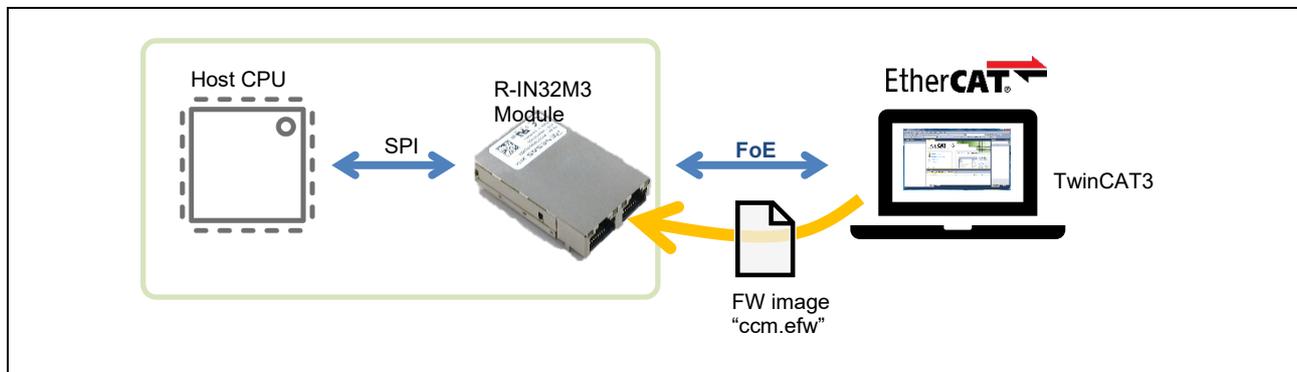


図 2-15 FoE によるファームウェア更新

## 2.3.2.2 事前準備

## (a) コンフィグ設定確認

R-IN32M3 Module のコンフィグ設定は、Management Tool を使って確認することができます。(2.2.3 章) FoE によるファームウェア更新に関連するコンフィグ設定を確認します。本説明では変更せずデフォルト設定のままとします。

表 2-4 FoE 設定

GOAL_ID_CCM	Variable ID	Type	Default value	Description
FOE_FILENAME	9	STRING	ccm.efw	EtherCAT FoE update file name
FOE_PASSWORD	10	UINT32	0x00000000	EtherCAT FoE update password
FOE_UPDATE_REQUIRES_BOOT	11	UINT8	0x01	EtherCAT FoE update required state
FOE_FILENAME_MATCH_LEN	12	UINT8	0x00	EtherCAT FoE update file name match length

## (b) ファームウェア更新ファイル

更新パッケージ(irj45\_2.0.0.0\_ci34.pfw 等)を解凍した中に含まれる、"irj45\_2port.bin" ファイルを上記の FOE\_FILENAME に設定する名称にリネームします。(本説明では、デフォルトの" ccm.efw" とします。)

## 2.3.2.3 手順

TwinCAT3 を使用して FoE によるファームウェア更新を行う手順について説明します。

- (1) プロジェクトをビルドし、RA6 ボードに書き込みます。
- (2) TwinCAT で EtherCAT 接続した後、プロジェクトツリーから Device を展開し、Online タブの Renesas Module の右クリックメニューから Firmware Update を選択します。

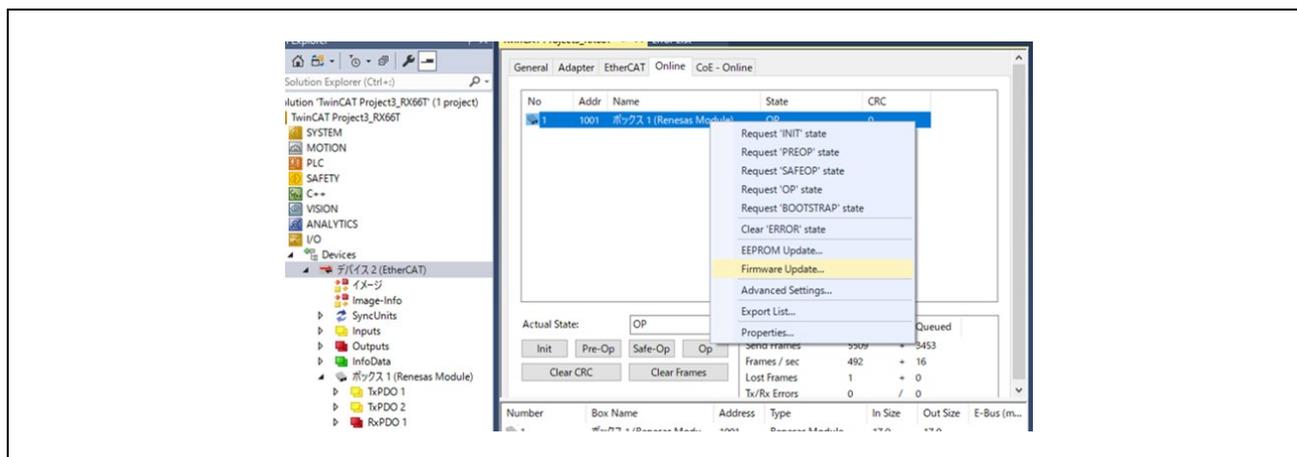


図 2-16 オンラインスレーブ画面のデバイス

- (3) 書き換え対象のファームウェアイメージファイルを選択します。("ccm.efw")

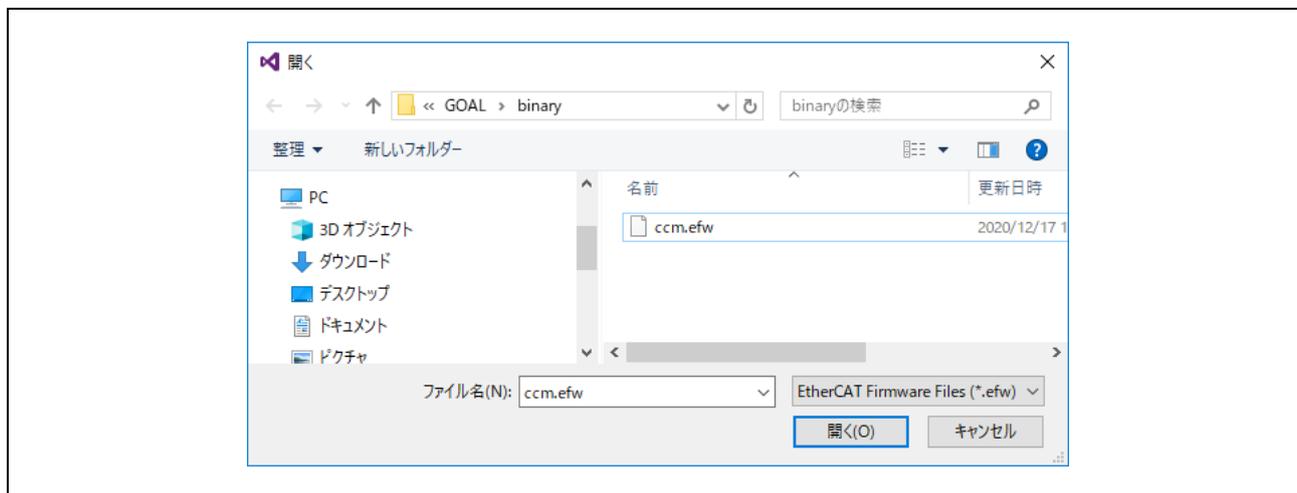


図 2-17 ファイル選択

- (4) FoE の指定パラメータを入力して、OK を押すとダウンロードを開始します。

(Note) ファイル選択だけでは "String" に表示されるファイル名に拡張子が含まれませんので、手動で入力してください。

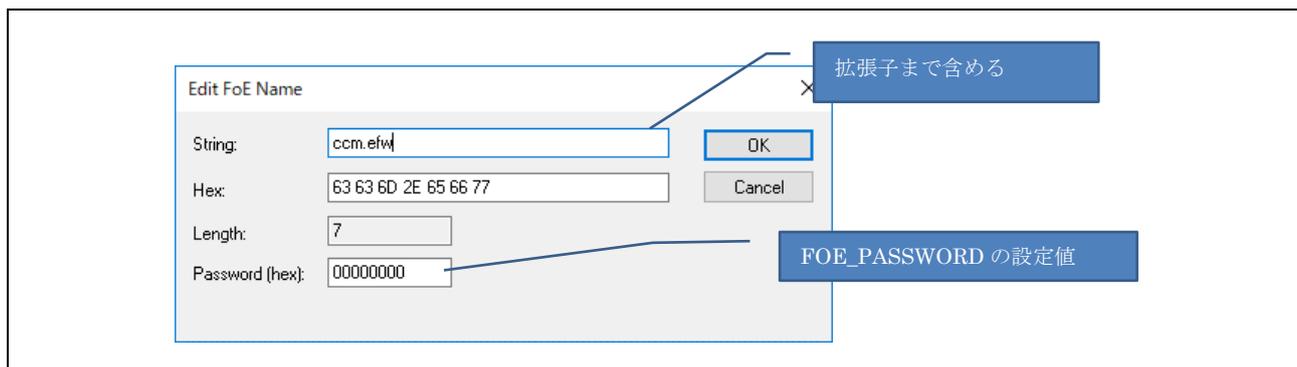


図 2-18 Edit FoE Name

## (5) 更新完了待ち

ダウンロードが行われたらデバイスステータスが自動的に INIT NO\_COMM に切り替わります。この間 FlashROM へ書き込みが行われ、この間 1 分ほど待ちます。その後ステータスが自動的に OP に切り替わり、ファームウェアの更新が完了します。

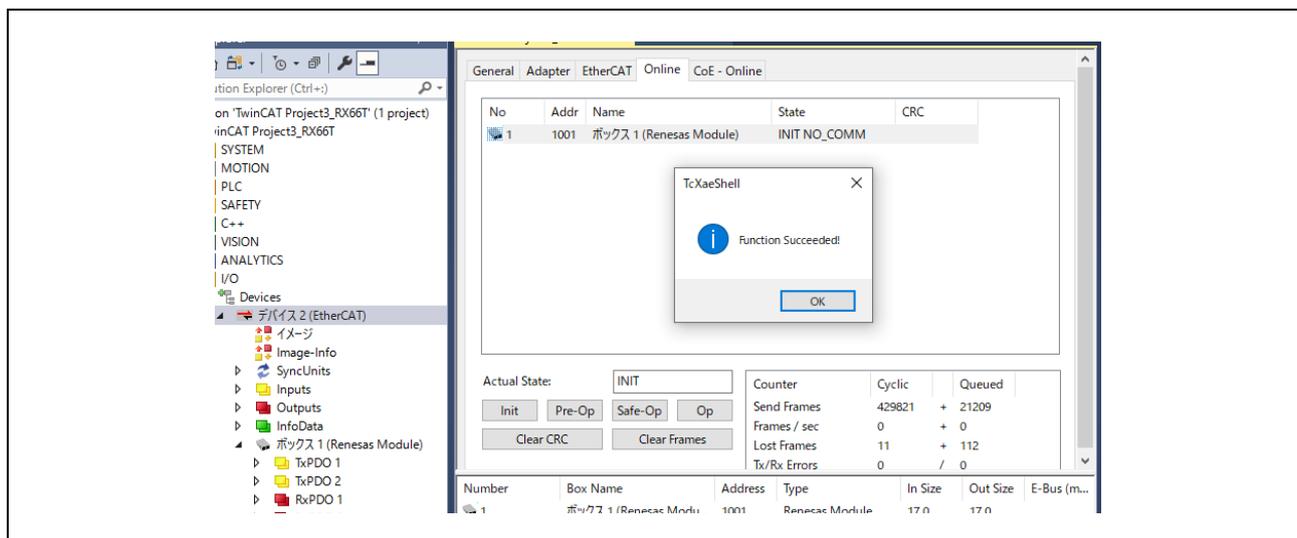


図 2-19 INIT NO\_COMM

### 2.3.3 SPI

#### 2.3.3.1 SPI 更新概要

ホストマイコンから SPI インタフェース経由で R-IN32M3 モジュールのファームウェアを更新する手順を説明します。

本機能は、RA6 ホストマイコン用のサンプルソフトウェアに実装されています(/01\_pnio, /02\_eip, /03\_ecat)。コンパイルオプション(GOAL\_CONFIG\_FW\_UPDATE\_SPI) を 1 にすることで機能が有効化されます。

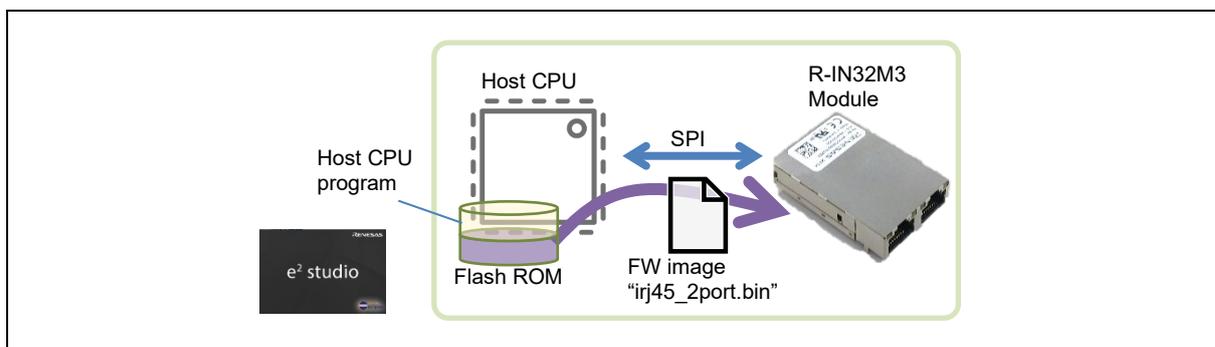


図 2-20 SPI 経由ファームウェア更新

サンプルプログラムでは、ホストマイコンの実行用プログラムを保存する Flash ROM 内に R-IN32M3 モジュールのファームウェアイメージも保存しておき、EK-RA6M4 および EK-RA6M3 ボード上のユーザスイッチ(S1)を押しながらホストマイコンを起動すると SPI 経由でファームウェアイメージが R-IN32M3 モジュールにダウンロードされ、ファームウェアが更新されるようになっております。

(Note) 本サンプルソフトを実行するためには、R-IN32M3 Module のファームウェアイメージを保存するための ROM 容量がホストマイコン側に必要となりますのでご注意ください。

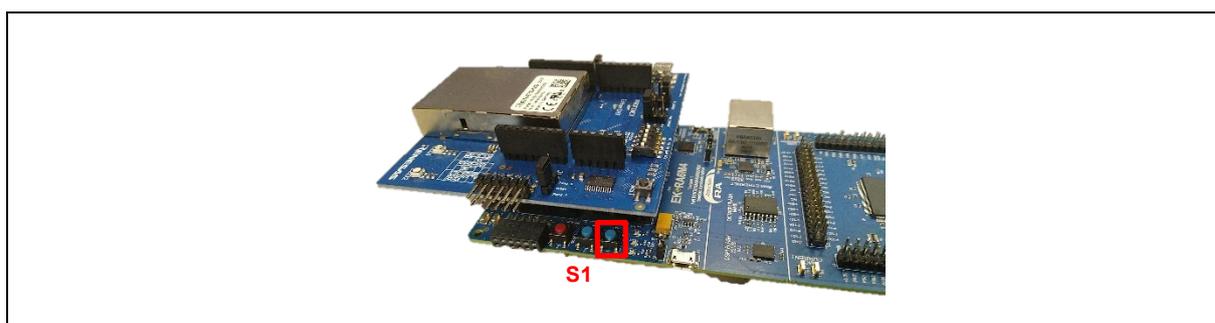


図 2-21 User Switch S1 on EK-RA6M4

## 2.3.3.2 手順

(1) F/W イメージのバイナリデータを用意します。

ファームウェア更新パッケージ (irj45\_2.1.0.0\_ci37.pfw 等) を zip 解凍し、内容物の irj45\_2port.bin を binary フォルダに配置します。binary フォルダがない場合は、/uGOAL 配下 (GOAL サンプルソフトの場合は /GOAL 配下) に作成して下さい。

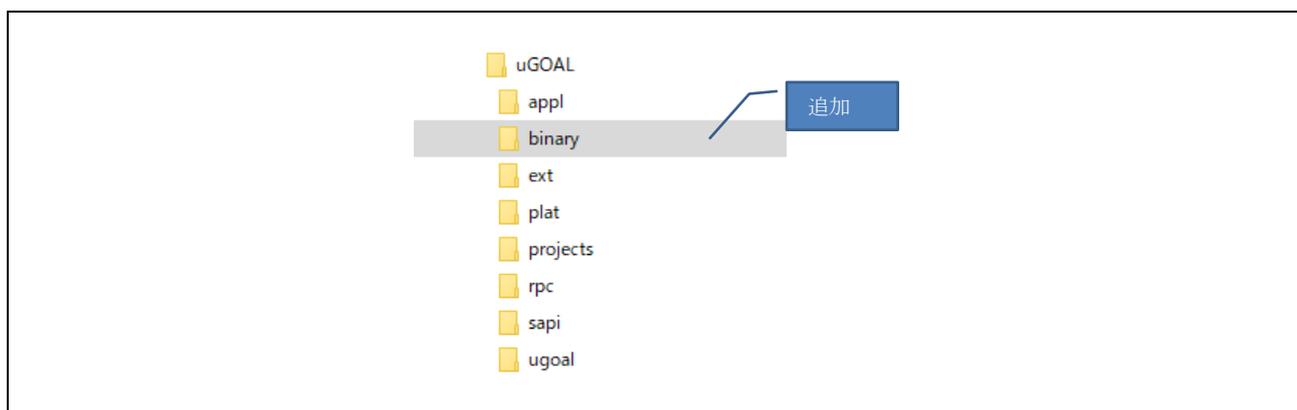


図 2-22 binary フォルダ

(2) コンパイルオプション設定

e2studio の該当プロジェクト (mirror\_sample) のプロパティを開き、コンパイルオプションを FW ダウンロード用のコードを有効にする設定に変更します。(GOAL\_CONFIG\_FW\_UPDATE\_SPI=1)

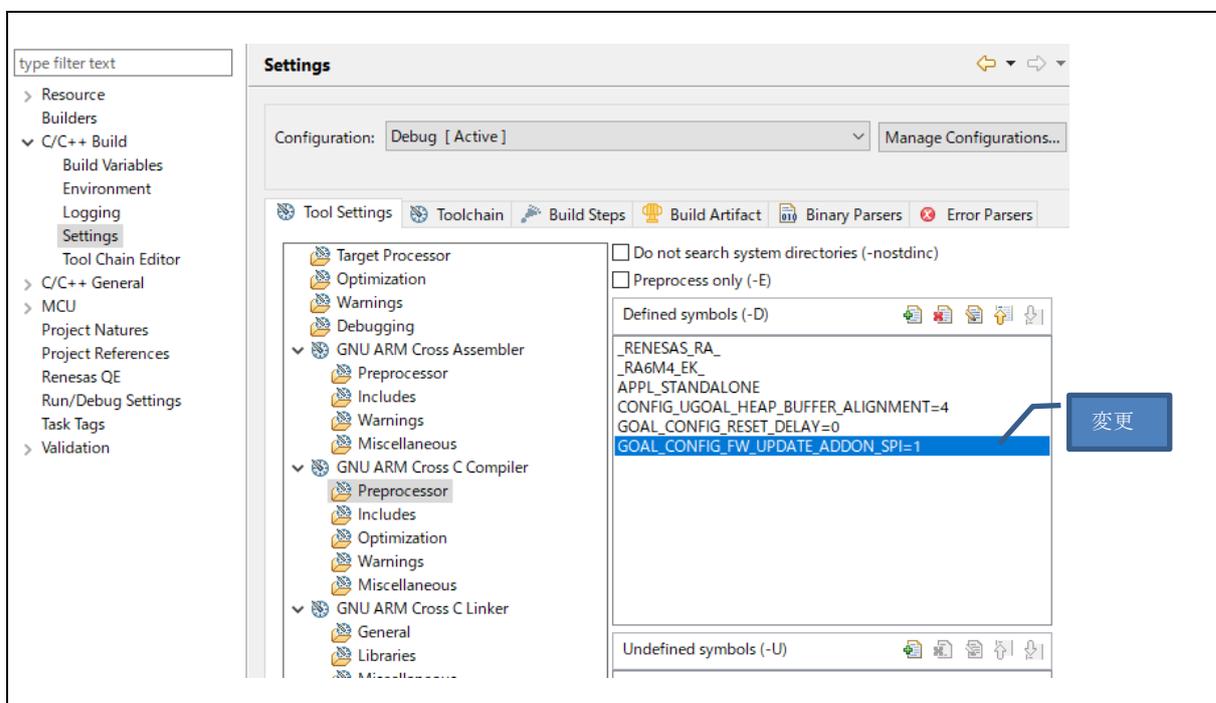


図 2-23 SPI 経由ファームウェア更新コード有効化

- (3) F/W イメージのバイナリデータをオブジェクトファイルに変換します。

F/W イメージをリンク可能な形式に objcopy を使って変換するように、下記コマンドをプレビルドに追加します。

```
cd ${GoalDirPath}/binary && arm-none-eabi-objcopy -I binary -O elf32-littlearm -B arm --rename-section .data=.rodata,alloc,data,readonly,load,contents irj45_2port.bin irj45_2port.o
```

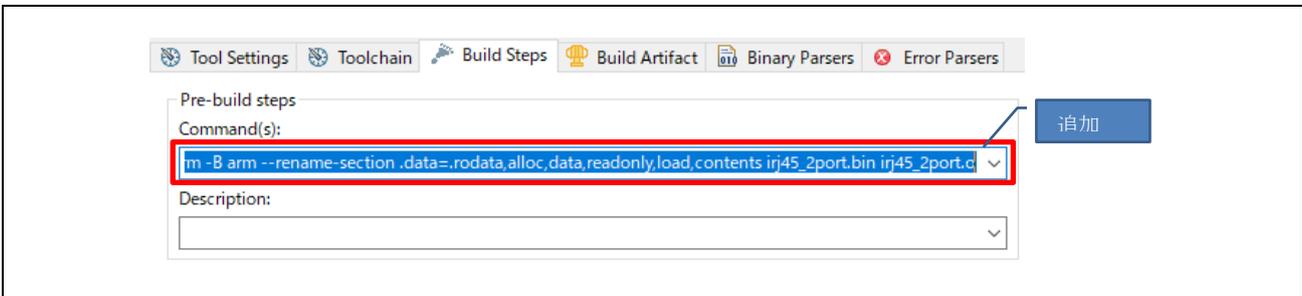


図 2-24 Pre-Build steps 追加

- (4) (3)で作成したファイルをリンク時にマージされるよう設定します。

リンカ設定の追加ファイルに以下の設定を追加します。

```
"${GoalDirPath}/binary/irj45_2port.o"
```

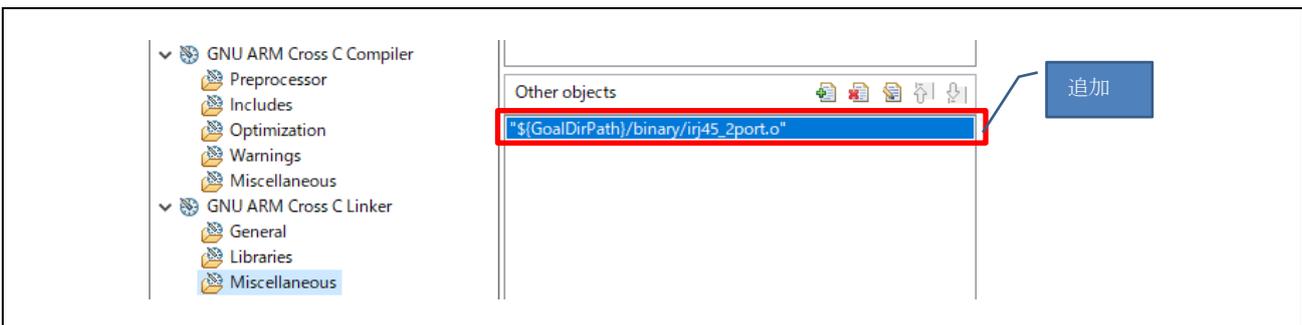


図 2-25 Linker 設定追加

- (5) プロジェクトをビルドし、RA6 ボードに書き込みます。

(Note) 評価ボードおよびプログラムサイズによっては書き込み時のコンペアに時間がかかり、e2studio の起動が失敗する可能性があります。その際は一旦終了して、再度立ち上げを行うことでコンペアチェックを回避することで書き込むことができます。

- (6) ファームウェア更新実行

RA6 ボードの S1 スイッチを押しながら、電源 ON（デバッグ起動中の場合はリセットスタート）すると、SPI 通信でファームウェアイメージの転送および書き込みが始まります。

データ転送が開始されると、LED1,LED2 を交互に点滅し、転送終了すると点滅が停止します。同時に動作中のプロトコル LED（PROFINET 等）が消灯します。（データ転送には 3 分程度かかります。）

SPI のデータ転送が終了すると R-IN32M3 モジュール側の更新が行われますので、1 分程度待ちます。（この間、R-IN32M3 モジュールは応答しません。）モジュール側のファームウェア更新が終わるとホストマイコン側を自動で再起動しますので、そこで更新完了となります。

デバッグ起動中の場合はプログラムの先頭に戻るので、続行ボタンを押してください。

## 2.3.3.3 バッチファイル (参考)

2.3.3.2 章で説明したように、プログラムのダウンロードに時間がかかりすぎると、e2studio がエラーとなる場合があります。本章では、e2studio を介せず、直接 J-Link を起動してホストマイコン RA6M3 のファームウェアを書き換えるバッチファイル例を示します。

## &lt;環境&gt;

- ・ RA6M3 サンプルソフト(uGOAL)
- ・ /01\_pnio プロジェクト
- ・ J-Link V6.3 以降が、C:/Program Files (x86)/SEGGER/JLink にインストールされていること。

## &lt;使い方&gt;

- (1) ダウンロードしたいプロジェクトは予めビルドします。
- (2) 以下のバッチプログラムをコピーした .bat ファイルを /uGOAL フォルダ直下に配置します。  
例では RA6M3 用の 01\_pnio プロジェクトをダウンロードしています。他プロジェクトで行う場合はパスおよび J-Link に対するデバイス指示を適宜変更してください。
- (3) バッチファイルをダブルクリックで実行する。
- (4) 成功すると図 2-26 のようなメッセージが出ます。
- (5) 電源を入れなおして完了になります。

```

Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via ATRCR.SYSRESETREQ.

Downloading file [.\%projects%\mirror_sample\01_pnio\%e2studio%\ra6m3ek%\Debug%\mirror_sample__01_pnio__ra6m3_osless.srec]...
J-Link: Flash download: Bank 0 @ 0x0100A150: Skipped. Contents already match
J-Link: Flash download: Bank 1 @ 0x00000000: 1 range affected (40960 bytes)
J-Link: Flash download: Total: 1.025s (Prepare: 0.116s, Compare: 0.121s, Erase: 0.210s, Program & Verify: 0.566s, Restore: 0.010s)
J-Link: Flash download: Program & Verify speed: 70 KB/s
O.K.

Script processing completed.

C:\Work\IEM\SVN\ikeda\IEM_uGOAL\%uGOAL>pause
続行するには何かキーを押してください . . .

```

図 2-26 バッチファイル実行 Log

## &lt;バッチプログラム例&gt;

```

if exist .%ProgramLoader.Cmd (
    del ProgramLoader.Cmd
)
echo h>> ProgramLoader.Cmd
echo r>> ProgramLoader.Cmd
echo
loadfile .%projects%\mirror_sample\01_pnio\%e2studio%\ra6m3ek%\Debug%\mirror_sample__01_pnio__ra6m3_osless.srec,0x00000000>> ProgramLoader.Cmd
echo q>> ProgramLoader.Cmd
"C:\Program Files (x86)\SEGGER\JLink\JLink.exe" -speed 4000 -if SWD -device R7FA6M3AH
-autoconnect 1 -CommanderScript ProgramLoader.Cmd
pause

```

### 3. ホストマイコン・ファームウェア更新

ホストマイコンのファームウェア更新について説明します。ここで説明するホストマイコンのファームウェア更新は、イーサネット(R-IN32M3 Module)を経由して実行する手法になります。

更新手法は、プラットフォームにより異なります。

(Note) RL 向けサンプルソフトではホストマイコンのファームウェア更新をサポートしていません。

#### 3.1 RA マイコン

RA マイコンのファームウェア更新について説明します。

RA サンプルアプリケーションノート [R30AN0398JJ\*\*\*\*]を参照し、開発環境をセットアップしてください。

(Note) RA マイコンのファームウェア更新は、Bootloader 領域に専用プログラムが書き込まれていることを前提としています。イーサネット経由での Bootloader の書き換えは不可のため、製品生産時の Flash Writer で Initial Firm[Boot Program + User Program]を書き込む必要があります。

**Bootloader サンプルプロジェクト: 17\_fwup\_bootloader**

##### 3.1.1 ツール

ファームウェアアップデート機能を実装する場合、署名データの埋め込みに Python スクリプトを使用しています。そのため、Python をインストールする必要があります。下記の手順により Python 環境を構築し、署名データの埋め込みんでください。

- 1. [Python](#) 3.8.3 以降をインストール
- 2. e2studio へインポートした Bootloader プロジェクト\* のスマートコンフィグレータを起動し、FSP のソースコードを生成します。 \* 17\_fwup\_bootloader\_<デバイス名>\_osless

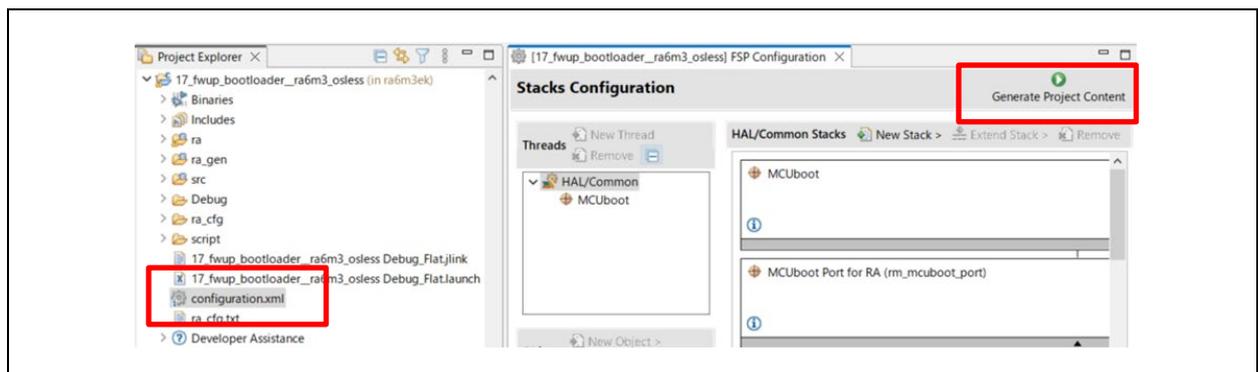


図 3-1 Bootloader コード生成

- 3. MCUboot フォルダを選択し、[コマンド プロンプト] を起動

```
projects¥ 17_fwup_bootloader¥e2studio¥<デバイス名>¥ra¥mcu-tools¥
```

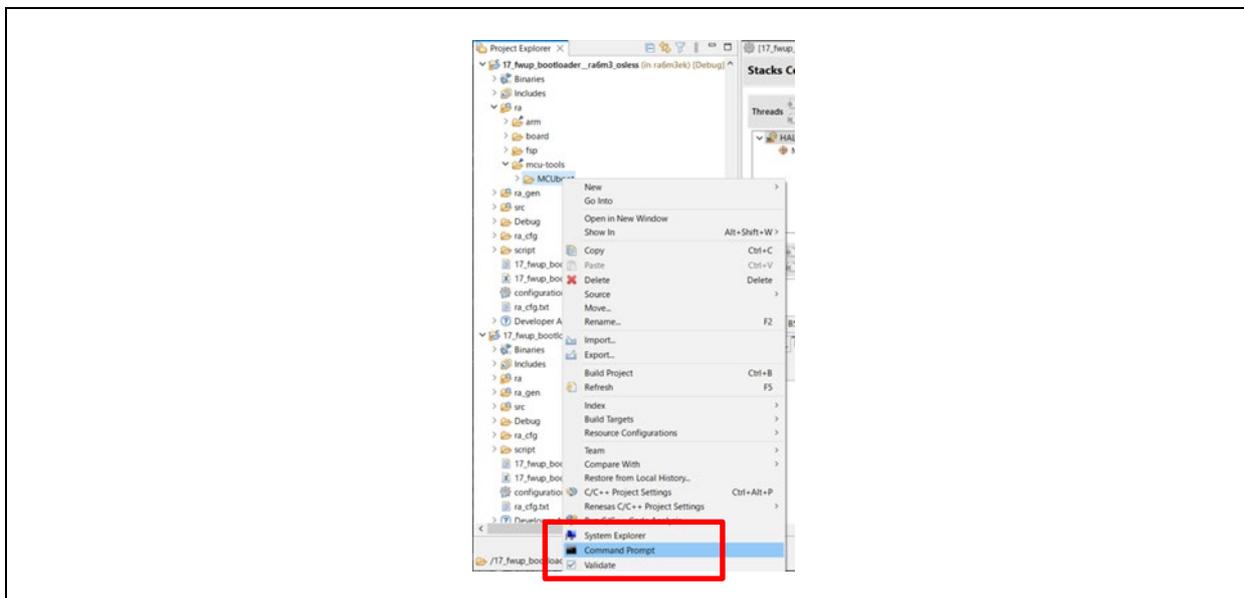


図 3-2 コマンド プロンプト

- 4. pip を更新

依存関係をインストールする前に、pip をアップグレードすることをお勧めします。 次のコマンドを入力して、pip を更新します。

```
python -m pip install --upgrade pip
```

- 5. 次のコマンド ラインを入力して、すべての MCUboot 依存関係をインストール

次に、コマンド ウィンドウで次のコマンド ラインを入力して、すべての MCUboot 依存関係をインストールします。

```
pip3 install --user -r scripts/requirements.txt
```

複数の Python バージョンがインストールされている場合は、Python バージョンがバージョン 3.8.3 以降であることを確認してください。

### 3.1.2 ファームウェア更新準備

#### 3.1.2.1 Bootloader プログラム

##### -1. Bootloader プロジェクトのビルド実行

##### -2. ユーザプログラムのプロジェクトのビルド実行

Bootloader には書き込み機能がないため、初期状態（メインプログラムが書き込まれていない状態）ではメインプログラムをフラッシュローダ等で書き込む必要があります。

各デバイス/プロトコルに対応するプロジェクトは以下になります。

Device	Sample software	Project Name
RA6M3	11_pnio_http	11_pnio_http__ra6m3_osless
	12_eip_http	12_eip_http__ra6m3_osless
	13_ecat_http	13_ecat_http__ra6m3_osless
RA6M4	11_pnio_http	11_pnio_http__ra6m4_osless
	12_eip_http	12_eip_http__ra6m4_osless
	13_ecat_http	13_ecat_http__ra6m4_osless

1) ビルド実行前にコンフィグレーションで FSP のコード生成を実行します。



図 3-3 コード生成

2) ビルド構成を”DebugFwup”に変更します。

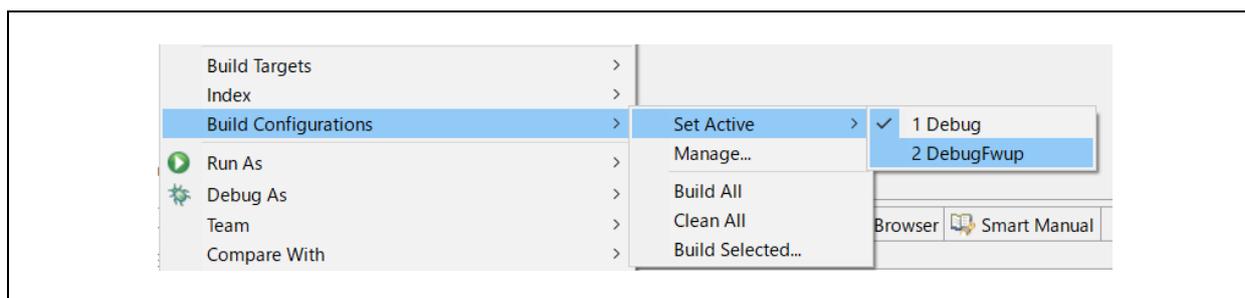


図 3-4 ビルド構成変更

- 3) ビルドに成功すると project フォルダ下に ユーザプログラムのダウンロードファイル <プロジェクト名>.bin.signed が生成されます。  
ex: projects¥11\_pnio\_http¥e2studio¥ra6m3ek¥DebugFwup

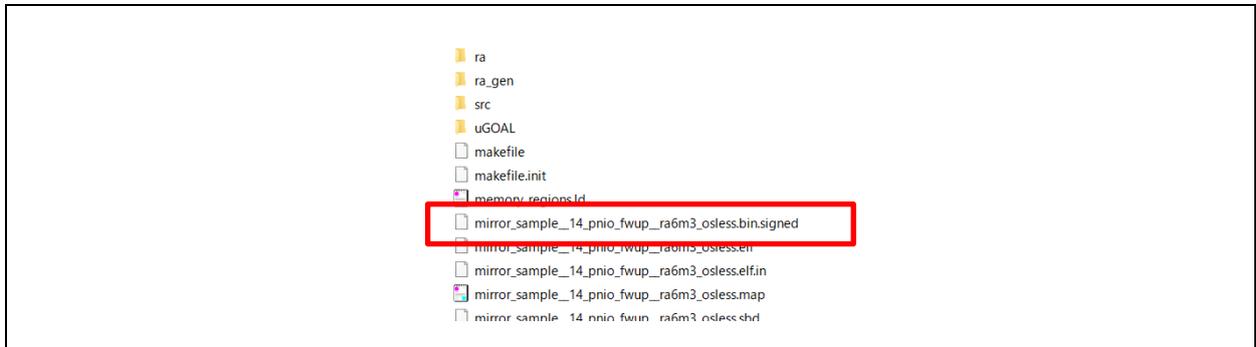


図 3-5 bin.signed ファイル

### -3. デバッグ実行

当該プロジェクトをデバッグする場合、Bootloader とユーザプログラムを同時にダウンロードする必要がありますのでデバッグ実行前に以下の設定をしてください。

#### 1) デバッガ設定に Bootloader をロードする設定を追加\*

\* 本設定はサンプルプロジェクトのデバッガ設定ファイルに設定済です

メニューから、[Run]→[Debug Configurations...]を選択し、Debug Configurations ダイアログを表示します。

- ・ユーザプログラムの Load type を Symbols only に変更
- ・同じデバイス名の Bootloader を追加
- ・ユーザプログラムのバイナリデータをユーザプログラム用領域 H'18000~に配置

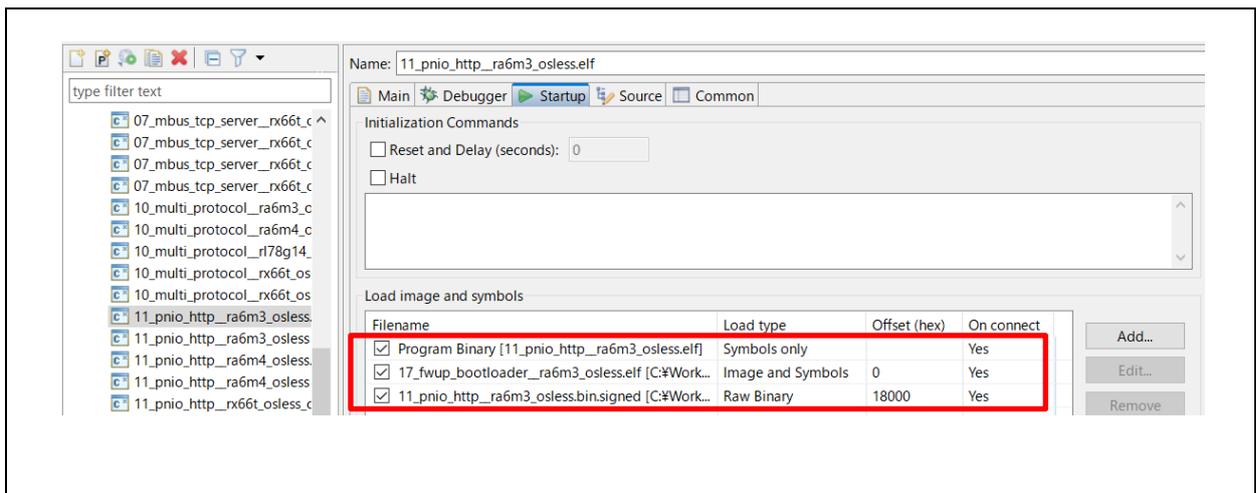


図 3-6 デバッガ設定

#### 2) デバッグ実行

以上で、Bootloader とユーザプログラムがプログラムされます。

### 3.1.2.2 アップデート（ユーザプログラム）

ユーザプログラムをネットワーク経由で書き換えます。

ユーザプログラムを書き換えるにはあらかじめ 3.1.2.1 Bootloader プログラムで説明している Bootloader とユーザプログラムが書き込まれている必要があります。

- 1. ブラウザ経由でターゲットのボードへ接続します。  
サンプルアプリケーションでは IP アドレスを "192.168.0.100" に設定していますので <http://192.168.0.100> で接続できます。

なお、EtherCAT の場合は EtherCAT マスタでの設定も必要となります。RA サンプルアプリケーションノート [R30AN0398JJ\*\*\*\*]の Web サーバ機能の章を参照し設定してください。

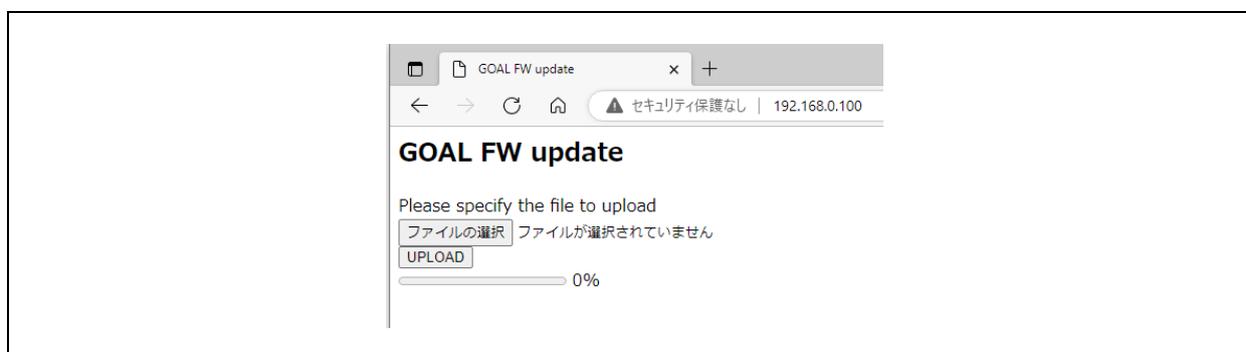


図 3-7 ブラウザアクセス

- 2. ロードモジュールの選択  
[ファイルの選択] からロードモジュール "\*.bin.signed" ファイルを指定します。
- 3. アップロード開始  
[UPLOAD] を押すとアップデートを開始します。  
アップロード完了までに 1 分程度かかります。

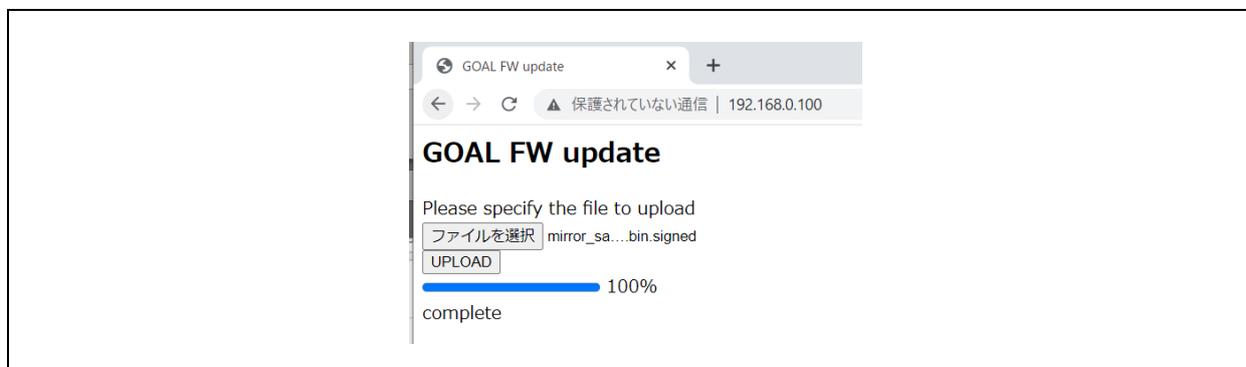


図 3-8 アップデート

-4. 再起動

アップロード完了時は、旧ファームウェアで動作しているため再起動します。再起動後は書き込んだ新ファームウェアで動作します。

## 3.2 RX マイコン

RX マイコンのファームウェア更新について説明します。

RX サンプルアプリケーションノート [R30AN0399JJ\*\*\*\*]を参照し、開発環境をセットアップしてください。

(Note) RX マイコンのファームウェア更新は、Bootloader 領域に専用プログラムが書き込まれていることを前提としています。イーサネット経由での Bootloader の書き換えは不可のため、製品生産時の Flash Writer で Initial Firm[Boot Program + User Program]を書き込む必要があります。

**Bootloader サンプルプロジェクト: 17\_fwup\_bootloader**

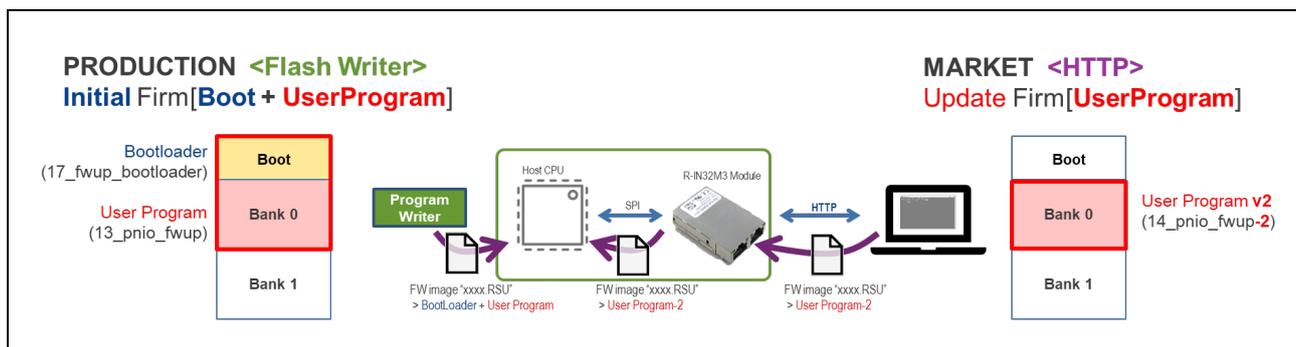


図 3-9 RX Firmware update

### 3.2.1 ツール

イーサネット経由でアップデートする場合、アプリケーションをビルドして生成された mot ファイルをダウンロード可能な RSU ファイル形式に変換する必要があります。ファイル形式の変換ツールとして、サンプルパッケージ[r18an0064xx01\*\*] に同梱されている Renesas Image Generator を使用します。

(Note) サンプルパッケージに同梱している Renesas Image Generator は、SEMB1320 R-IN32M3 モジュール搭載 CPU カードで採用されているホストマイコン(R5F566TKADFP)を公開している Renesas Image Generator に追加したものです。

オリジナルは、[Convert MOT file to RSU | \(github.com/renesas\)](https://github.com/renesas) で公開されています。

### 3.2.2 ファームウェア更新準備

RX マイコンのファームウェア更新は、Bootloader 領域に専用プログラムが書き込まれていることを前提としています。ここでは、更新ファームウェアの生成手法と更新手順を説明します。

更新ファームウェアのサンプルプロジェクトはプロトコル毎に用意されています。

表 3-1 ブートローダと更新ファームウェア対応サンプルプロジェクト

Protocol	Sample project
PROFINET	11_pnio_http
EtherNet/IP	12_eip_http
EtherCAT	13_ecat_http
Function	Sample project
Bootloader	17_fwup_bootloader

#### 3.2.2.1 Bootloader プログラム

Bootloader とユーザプログラムのプログラム手順を示します。

##### (1) Tinycrypt ライブラリの追加

更新ファームウェアの完全性を確保するためにデジタル署名(ECDSA+SHA256)にて検証を行います。検証を行うために下記のコードをサンプルプロジェクトに追加する必要があります。

[Github | \(github.com/renesas/\)](https://github.com/renesas/) より取得し、ブートローダプロジェクト配下にある lib フォルダに追加します。追加後のフォルダ構成例を下図に示します。

**lib フォルダ:** 17\_fwup\_bootloader¥tinycrypt¥lib¥include¥tinycrypt

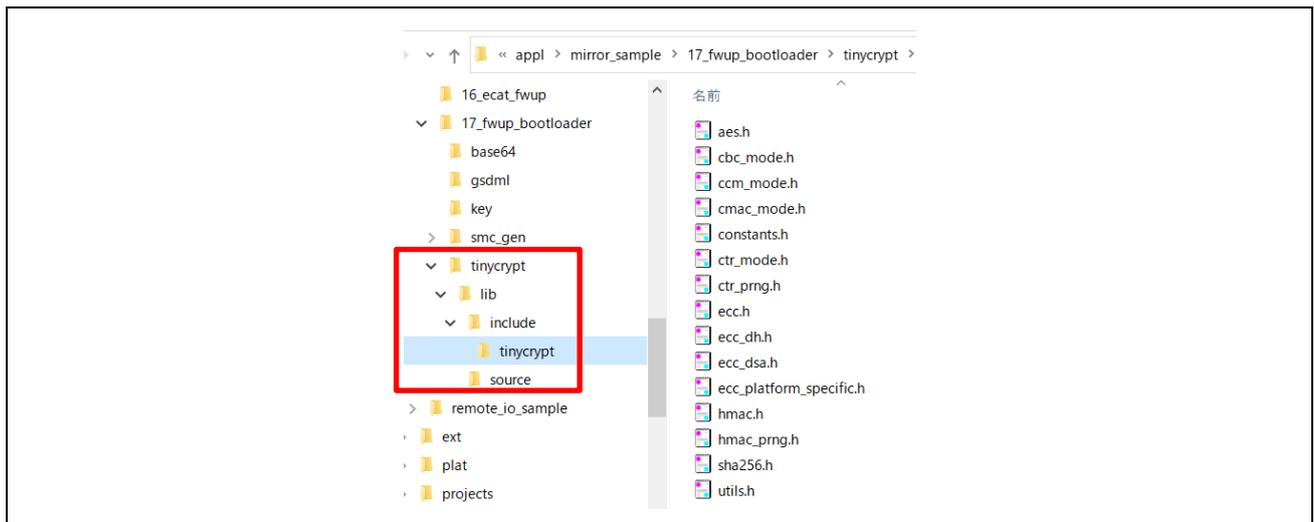


図 3-10 Library

## (2) Key ファイルの公開鍵情報入力

署名検証用の公開鍵情報をサンプルアプリケーションのヘッダファイル [code\_signer\_public\_key.h] に設定します。

ファームウェア検証に使用する鍵を OpenSSL で作成

作成方法については下記解説を参照ください。

[OpenSSL での ECDSA+SHA256 用の鍵ペア生成方法 | \(github.com/renesas/\)](https://github.com/renesas/openssl/blob/master/doc/man1/openssl-keygen.1.rst)

作成後、公開鍵は secp256r1.publickey、秘密鍵は secp256r1.privatekey というファイルでそれぞれ生成されます。

```
#define CODE_SIGNER_PUBLIC_KEY_PEM \  
"-----BEGIN PUBLIC KEY-----\  
"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEwiA1aCQGEgIKoP+qk7Uqc/ME/hjw\  
"amq1v/z/LWx15CKig59Pd3+ar2RF01MM0hIfkYgS+Ha7tH+w0gggnKDrUug==\  
"-----END PUBLIC KEY-----"
```

図 3-11 Key pair

## (3) ビルド実行

Bootloader プロジェクトをビルドします。

## (4) ブートローダ書き込み

デバッガで起動します。引き続きユーザプログラムを書き込むため、一旦デバッグを停止ボタン等で停止させ事項に進みます。

## (5) ユーザプログラム起動

ユーザプログラムをデバッガで起動します。デバッグ画面が立ち上がり、プログラムのダウンロードが完了したら、[再開]ボタンを選択することで、プログラムが実行されます。

## 3.2.2.2 更新用ユーザプログラムプログラム生成

ユーザプログラムの更新ファームウェア RSU ファイルの生成について説明します。

## 1. Mod ファイル生成

更新ファームウェアを生成するには、プロジェクトのビルド時にモトローラ形式(\*.mot) でファイル生成します。

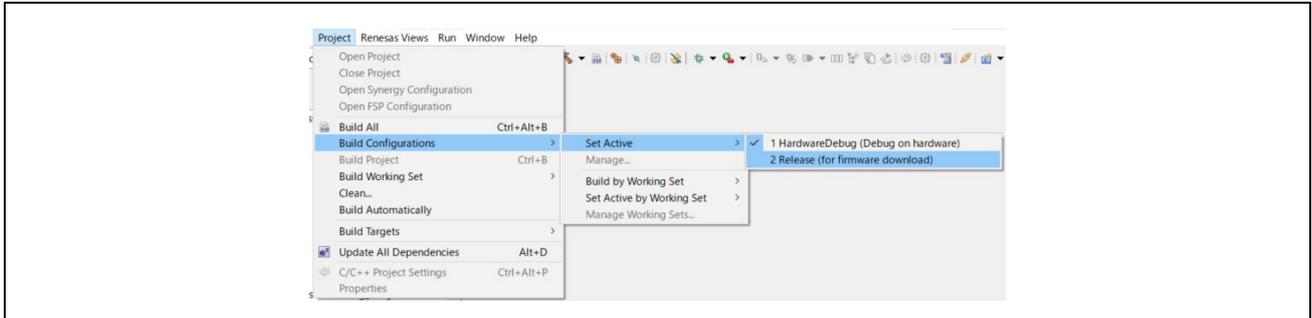


図 3-12 Build mot file

## 2. RSU ファイル変換

Renesas Image Generator を使用して、生成した mot ファイルを RSU ファイルに変換します。

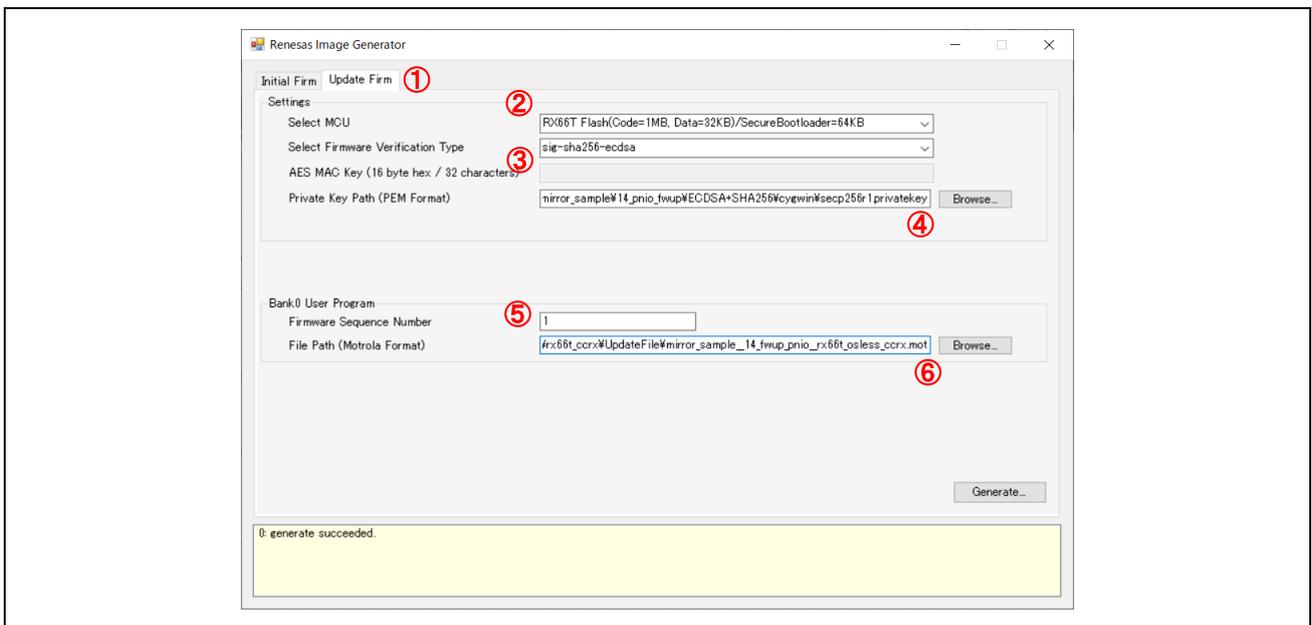


図 3-13 RSU conversion [Renesas Image Generator]

- ① [Update Firm] 選択
- ② [RX66T Flash(Code=1MB, Data=32KB)/SecureBootloader=64KB] 選択
- ③ [sig-sha256-ecdsa] 選択
- ④ [3.2.2.1](#) で生成した secp256r1.privatekey を指定
- ⑤ 任意の数値を設定
- ⑥ -1. で生成した mod ファイルを指定

[Generate]ボタン押下後、RSU ファイルの生成に成功すると "generate succeeded"と表示されます。

### 3.2.2.3 アップデート・ユーザプログラム

ユーザプログラムをネットワーク経由で書き換えます。

ユーザプログラムを書き換えるにはあらかじめ [3.2.2.1 Bootloader プログラム](#) で説明している Bootloader とユーザプログラムが書き込まれている必要があります。

- 1) ブラウザ経由でターゲットのボードへ接続します。  
サンプルアプリケーションでは IP アドレスを "192.168.0.100" に設定していますので `http://192.168.0.100` で接続できます。

なお、EtherCAT の場合は EtherCAT マスタでの設定も必要となります。RX サンプルアプリケーションノート [R30AN0399JJ\*\*\*\*]の Web サーバ機能の章を参照し設定してください。

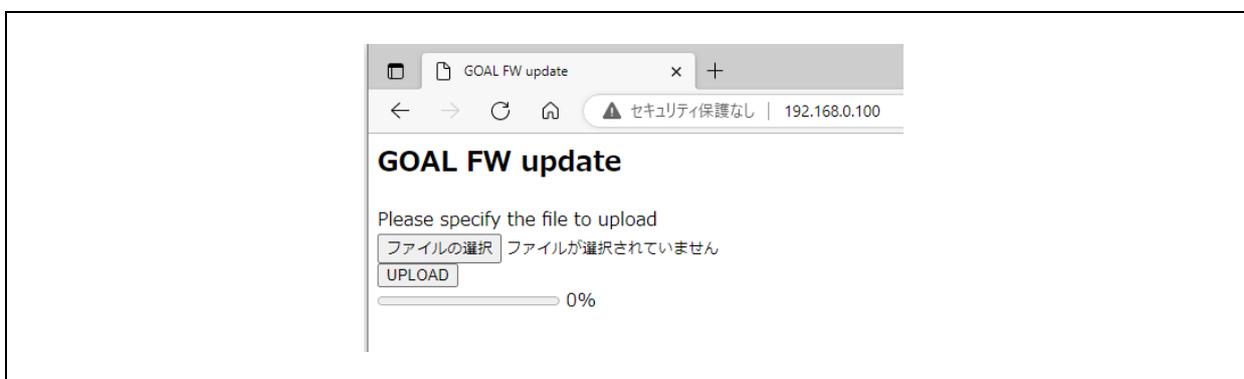


図 3-14 ブラウザアクセス

- 2) RSU ファイル選択  
“ファイルの選択” から Renesas Image Generator で生成した RSU ファイルを指定します。
- 3) アップロード開始  
“UPLOAD” を押すとアップデートを開始します。  
アップロード完了までに 1 分程度かかります。

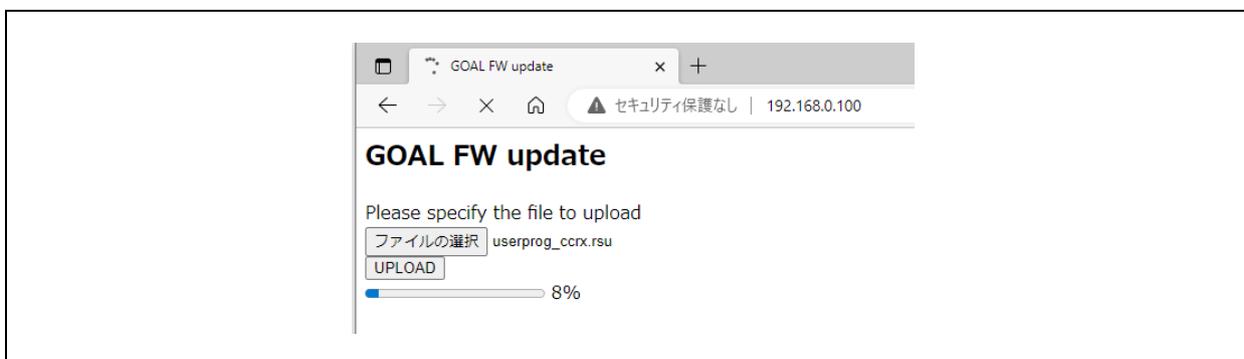


図 3-15 アップデート

## 4) 再起動

アップロード完了時は、旧ファームウェアで動作しているため再起動します。再起動後は書き込んだ新ファームウェアで動作します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2021/10/15	-	新規作成
1.01	2023/05/31	28	ホストマイコン・ファームウェア更新機能追加
1.02	2023/12/15	22	TwinCAT での Firmware 更新方法について内容差し替え

## 商標

- \* Arm および Cortex は、Arm Limited（またはその子会社）の EU またはその他の国における登録商標です。
- \* Ethernet およびイーサネットは、富士ゼロックス株式会社の登録商標です。
- \* EtherCAT は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。
- \* その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です

## 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、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

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の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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「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。

### 5. クロックについて

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。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）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。

### 8. 製品間の相違について

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。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品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生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

- 当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。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. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
  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。
  12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 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が直接的、間接的に支配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。

(Rev.4.0-1 2017.11)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。