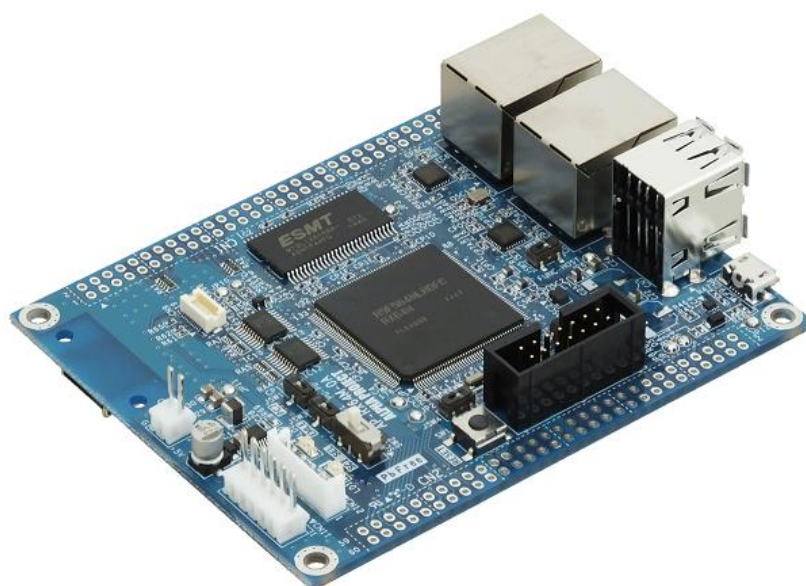

α RPL

対応ボード使用手順

AP-RX64M-0A 編

2.1 版



AP ALPHA PROJECT
株式会社アルファプロジェクト

目 次

1. 概要	1
1.1 概要.....	1
1.2 使用環境.....	1
2. 機能	2
2.1 CPU 初期設定	2
2.2 ボード対応機能一覧.....	6
2.3 ピンアサイン.....	15
2.4 ライブラリ使用メモリ.....	18
3. 動作手順	19
3.1 サンプル動作手順.....	19

1. 概要

1.1 概要

α RPL 対応ボード使用手順 AP-RX64M-0A 編（以下、本マニュアル）では、
「2. 機能」で α RPL のうち AP-RX64M-0A で使用可能なライブラリの一覧、
各ライブラリ機能で設定/取得可能なパラメータの一覧、AP-RX64M-0A のピンアサインを示します。
「3. 動作手順」ではデジタル出力のライブラリ機能を使用した LED（LD2）の点滅プログラムを例に、
動作を確認する手順を紹介します。
なお動作を確認するには、あらかじめ「α RPL User Manual」の「ライブラリ組み込み手順」をお読みいただき、
ライブラリを使用するための準備を行ってください。

1.2 使用環境

α RPL を組み込んで AP-RX64M-0A を開発する際の使用環境を以下に記載します。

名称	バージョン	用途	備考
Windows 10/11		OS	64 ビット OS
統合開発環境 CS+	3.01.00	統合開発環境	旧 CubeSuite+
RX ファミリー用 C/C++コンパイラパッケージ	2.03.00	コンパイル、ビルド	評価版使用可能
E1 エミュレータ / E2 エミュレータ Lite		デバッグ、FlashROM 書き込み	
Renesas Flash Programmer	2.05.01	FlashROM 書き込み	評価版使用可能
PC-USB-04		マイコンのシリアルポートを USB ポートに変換	弊社製品

Table 1.2-1 使用環境(AP-RX64M-0A)

2. 機能

2.1 CPU 初期設定

αRPL ではメイン処理に入る前にあらかじめ CPU の初期設定が行われています。
各動作周波数の設定を以下に示します。

クロック	動作周波数
ICLK	120[MHz]
PCLKA	120[MHz]
PCLKB	60[MHz]
PCLKC	60[MHz]
PCLKD	60[MHz]
FCLK	60[MHz]
BCLK	60[MHz]
BCLK 端子出力	30[MHz]
SDCLK	60[MHz]
UCLK	60[MHz]

Table 2.1-1 各動作周波数の設定

各ピンの初期設定を以下に示します。

ピン	初期設定
P00	汎用入力
P01	汎用入力
P02	汎用入力
P03	汎用入力
P04	-
P05	汎用入力
P06	-
P07	汎用入力
P10	USBA_OVRCURA として使用
P11	USBA_VBUS として使用
P12	SCL0[FM+]として使用
P13	SDA0[FM+]として使用
P14	汎用入力
P15	汎用入力
P16	USB0_VBUS として使用
P17	汎用入力

※、「-」は存在しないピンです。

Table 2.1-2 各ピンの初期設定 1

ピン	初期設定
P20	汎用入力
P21	汎用入力
P22	汎用入力
P23	汎用入力
P24	汎用入力
P25	汎用入力
P26	汎用入力
P27	汎用入力
P30	ET1_MDIO として使用
P31	ET1_MDC として使用
P32	汎用入力
P33	汎用入力
P34	汎用入力
P35	汎用入力
P36	EXTAL として使用
P37	XTAL として使用
P40	汎用入力
P41	汎用入力
P42	汎用入力
P43	汎用入力
P44	汎用入力
P45	汎用入力
P46	汎用入力
P47	汎用入力
P50	外部バスの WR0#/WR# として使用
P51	汎用入力
P52	外部バスの RD# として使用
P53	外部バスの BCLK として使用
P54	-
P55	-
P56	-
P57	-
P60	RMII1_TXD_EN として使用
P61	外部バスの SDCS# として使用
P62	外部バスの RAS# として使用
P63	外部バスの CAS# として使用
P64	外部バスの WE# として使用
P65	外部バスの CKE として使用
P66	外部バスの DQM0 として使用
P67	外部バスの DQM1 として使用

※. 「-」は存在しないピンです。

Table 2.1-3 各ピンの初期設定 2

ピン	初期設定
P70	SDCLK として使用
P71	ET0_MDIO として使用
P72	ET0_MDC として使用
P73	汎用入力 (USB 機能で使用)
P74	RMII0_RXD1 として使用
P75	RMII0_RXD0 として使用
P76	REF50CK0 として使用
P77	RMII0_RX_ER として使用
P80	RMII0_TXD_EN として使用
P81	RMII0_TXD0 として使用
P82	RMII0_TXD1 として使用
P83	RMMIO_CRS_DV として使用
P84	-
P85	-
P86	汎用入力
P87	汎用入力
P90	汎用入力 (USB 機能で使用)
P91	汎用入力 (USB 機能で使用)
P92	RMII1_CRS_DV として使用
P93	汎用入力
P94	RMII1_RXD0 として使用
P95	RMII1_RXD1 として使用
P96	汎用入力
P97	汎用入力
PA0	外部バスの A0 として使用
PA1	外部バスの A1 として使用
PA2	外部バスの A2 として使用
PA3	外部バスの A3 として使用
PA4	外部バスの A4 として使用
PA5	外部バスの A5 として使用
PA6	外部バスの A6 として使用
PA7	外部バスの A7 として使用
PB0	外部バスの A8 として使用
PB1	外部バスの A9 として使用
PB2	外部バスの A10 として使用
PB3	外部バスの A11 として使用
PB4	外部バスの A12 として使用
PB5	外部バスの A13 として使用
PB6	外部バスの A14 として使用
PB7	外部バスの A15 として使用

※. 「-」は存在しないピンです。

Table 2.1-4 各ピンの初期設定 3

ピン	初期設定
PC0	外部バスの A16 として使用
PC1	外部バスの A17 として使用
PC2	外部バスの A18 として使用
PC3	汎用入力
PC4	汎用入力
PC5	汎用入力
PC6	汎用入力
PC7	汎用入力
PD0	外部バスの D0 として使用
PD1	外部バスの D1 として使用
PD2	外部バスの D2 として使用
PD3	外部バスの D3 として使用
PD4	外部バスの D4 として使用
PD5	外部バスの D5 として使用
PD6	外部バスの D6 として使用
PD7	外部バスの D7 として使用
PE0	外部バスの D8 として使用
PE1	外部バスの D9 として使用
PE2	外部バスの D10 として使用
PE3	外部バスの D11 として使用
PE4	外部バスの D12 として使用
PE5	外部バスの D13 として使用
PE6	外部バスの D14 として使用
PE7	外部バスの D15 として使用
PF0	汎用入力（オンチップエミュレータ機能で使用）
PF1	汎用入力（オンチップエミュレータ機能で使用）
PF2	汎用入力（オンチップエミュレータ機能で使用）
PF3	汎用入力（オンチップエミュレータ機能で使用）
PF4	汎用入力（オンチップエミュレータ機能で使用）
PF5	汎用入力
PF6	-
PF7	-
PG0	REF50CK1 として使用
PG1	RMII1_RX_ER として使用
PG2	汎用入力（LED 機能で使用）
PG3	RMII1_TXD0 として使用
PG4	RMII1_TXD1 として使用
PG5	汎用入力
PG6	汎用入力
PG7	汎用入力

※. 「-」は存在しないピンです。

Table 2.1-5 各ピンの初期設定 4

ピン	初期設定
PJ0	-
PJ1	-
PJ2	-
PJ3	汎用入力
PJ4	-
PJ5	汎用入力
PJ6	-
PJ7	-

※、「-」は存在しないピンです。

Table 2.1-6 各ピンの初期設定 5

2.2 ボード対応機能一覧

α RPL のライブラリ機能のうち、AP-RX64M-0A で使用可能であるものを○、使用不可能であるものを×として以下に示します。

ライブラリ機能	クラス	AP-RX64M-0A での使用可否
デジタル入力	DigitalIn	○
デジタル出力	DigitalOut	○
デジタル入出力	DigitalInOut	○
バス入力	BusIn	○
バス出力	BusOut	○
バス入出力	BusInOut	○
ポート入力	PortIn	○
ポート出力	PortOut	○
ポート入出力	PortInOut	○
PWM 出力	PwmOut	○
アナログ入力	AnalogIn	○
アナログ出力	AnalogOut	○
シリアル通信	Serial	○
SPI マスタ通信	SPI	○
SPI スレーブ通信	SPISlave	○
I2C マスタ通信	I2C	○
時間計測タイマ	Timer	○
ワンショットタイマ	TimeOut	○
繰り返しタイマ	Ticker	○
IRQ 割り込み	InterruptIn	○
ウェイト機能	※	○

Table 2.2-1 AP-RX64M-0A 対応ライブラリ機能

※、ウェイト機能にクラスは存在しません。

2.2.1 対応クラス使用可能ピン

各ライブラリ機能に対応するクラスにおいて、使用可能なピンの一覧を以下に示します。

クラス	コンストラクタ	使用可能ピン	補足
DigitalIn	DigitalIn [インスタンス名](PinName pin)	P00～P03, P05, P07	-
	DigitalIn [インスタンス名](PinName pin, PinMode mode)	P14, P15, P17 P20～P27 P32～P35 P40～P47 P51 P86, P87 P93, P96, P97 PC3～PC7 PF5 PG2, PG5～PG7 PJ3, PJ5	
DigitalOut	DigitalOut [インスタンス名](PinName pin)	P00～P03, P05, P07 P14, P15, P17	-
	DigitalOut [インスタンス名](PinName pin, int value)	P20～P27 P32～P34 P40～P47 P51 P86, P87 P93, P96, P97 PC3～PC7 PF5 PG2, PG5～PG7 PJ3, PJ5	
DigitalInOut	DigitalInOut [インスタンス名](PinName pin)	デジタル入力時は DigitalIn、デジタル出力時は DigitalOut の使用可能ピンに従います。	-
	DigitalInOut [インスタンス名](PinName pin, PinDirection direction, PinMode mode, int value)		
BusIn	BusIn [インスタンス名](PinName p0, PinName p0=NC, … , PinName p15=NC)	DigitalIn と同様 (ただし、最大 16 ピンまで設定可能)	-
BusOut	BusOut [インスタンス名](PinName p0, PinName p0=NC, … , PinName p15=NC)	DigitalOut と同様 (ただし、最大 16 ピンまで設定可能)	-
BusInOut	BusInOut [インスタンス名](PinName p0, PinName p0=NC, … , PinName p15=NC)	バス入力時は BusIn、バス出力時は BusOut の使用可能ピンに従います。	-

Table 2.2-2 使用可能ピン一覧 1

クラス	コンストラクタ	使用可能ピン	補足
PortIn	PortIn [インスタンス名](PortName port, int mask)	PORT_0 PORT_1 PORT_2 PORT_3 PORT_4 PORT_5 PORT_8 PORT_9 PORT_C PORT_F PORT_G PORT_J	DigitalIn の使用可能ピンをご確認頂き、mask を正しく設定して使用して下さい。
PortOut	PortOut [インスタンス名](PortName port, int mask)	PortIn と同様	DigitalOut の使用可能ピンをご確認頂き、mask を正しく設定して使用して下さい。
PortInOut	PortInOut [インスタンス名](PortName port, int mask)	PortIn と同様	ポート入力時は DigitalIn、ポート出力時は DigitalOut の使用可能ピンをご確認頂き、mask を正しく設定して使用して下さい。
PwmOut	PwmOut [インスタンス名](PinName pin)	P14/MTIOC3A P26/MTIOC2A P34/MTIOC0A PJ3/MTIOC3C	-
AnalogIn	AnalogIn [インスタンス名](PinName pin)	P40/AN000～P47/AN007	-
AnalogOut	AnalogOut [インスタンス名](PinName pin)	P03/DA0 P05/DA1	-
Serial	Serial [インスタンス名](PinName tx, PinName rx)	PF0/TXD1, PF2/RXD1 P00/TXD6, P01/RXD6	シリアル IF 用コネクタにより SCI1 と SCI6 を切り替えることが可能です。
SPI	SPI [インスタンス名](PinName mosi, PinName miso, PinName scl)	PC6/MOSIA, PC7/MISOA, PC5/RSPCKA	-
SPISlave	SPISlave [インスタンス名](PinName mosi, PinName miso, PinName scl, PinName ssel)	PC6/MOSIA, PC7/MISOA, PC5/RSPCKA, PC4/SSLA0	-
I2C	I2C [インスタンス名](PinName sda, PinName scl)	P13/SDA0, P12/SCL0	-
InterruptIn	InterruptIn [インスタンス名](PinName pin)	PF5/IRQ4 P02/IRQ10 P07/IRQ15	-

Table 2.2-3 使用可能ピン一覧 2

2.2.2 対応クラス設定可能パラメータ

各クラスにおいて設定可能なパラメータの一覧を以下に示します。

クラス	コンストラクタ/メンバ関数	設定可能パラメータ	補足
DigitalIn	DigitalIn [インスタンス名](PinName pin, PinMode mode)	PullNone PullUp OpenDrain PullDefault (PullNone)	-
	void mode(PinMode pull)		
DigitalInOut	DigitalInOut [インスタンス名](PinName pin, PinDirection direction, PinMode mode, int value)	PullNone PullUp OpenDrain PullDefault (PullNone)	-
	void mode(PinMode pull)		
BusIn	void mode(PinMode pull)	PullNone PullUp OpenDrain PullDefault (PullNone)	-
BusInOut	void mode(PinMode pull)	PullNone PullUp OpenDrain PullDefault (PullNone)	-
PortIn	PortIn [インスタンス名](PortName port, int mask)	0x00～0xFF	-
	void mode(PinMode pull)	PullNone PullUp OpenDrain PullDefault (PullNone)	-
PortOut	PortOut [インスタンス名](PortName port, int mask)	0x00～0xFF	-
	void write(int value)	0x00～0xFF	-
PortInOut	PortInOut [インスタンス名](PortName port, int mask)	0x00～0xFF	-
	void mode(PinMode pull)	PullNone PullUp OpenDrain PullDefault (PullNone)	-
	void write(int value)	0x00～0xFF	-
PwmOut	void period(float seconds)	0.000001～0.05	1[us]～50[ms]で設定可能です。
	void period_ms(int ms)	1～50	
	void period_us(int us)	1～50000	

Table 2.2-4 対応クラス設定可能パラメータ 1

クラス	コンストラクタ/メンバ関数	設定可能パラメータ	補足
AnalogOut	void write_u16(unsigned short value)	0x0000～0x0FFF	DAC は 12 ビット分解能です。
Serial	void baud(int baudrate)	120～1875000 ※	以下はボーレートの設定例です。 「120, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200, 230400, 460800, 921600, 1875000」

Table 2.2-5 対応クラス設定可能パラメータ 2

※.シリアル通信のボーレート設定を、補足にある設定例以外のパラメータで設定する場合、
 以下の手順を行い、使用可能なボーレートであることを確認してから設定するようにしてください。

1. 設定予定のボーレートを用いて以下の計算式①から値域に収まるような n, N の値を求めます。

$$N = \frac{PCLK}{64 \times 2^{2n-1} \times B} - 1 \quad \dots \textcircled{1}$$

(B=ビットレート[bps]、PCLK=60[MHz]、 $0 \leq n \leq 3$ 、 $0 \leq N \leq 255$)

2. 次に、①の計算式から得られた n, N の値を用いて、以下の計算式②から誤差を求めます。

$$\text{誤差} = \left\{ \frac{PCLK}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100 \quad \dots \textcircled{2}$$

3. ②の計算式で得られた誤差が 3%以内（目安）であれば、指定のボーレートは使用できる設定値であると判断します。
 実際に通信を行って問題がないことを確認してください。

クラス	コンストラクタ/メンバ関数	設定可能パラメータ	補足
SPI	void frequency(int hz)	29300～30000000 ※	以下はビットレートの設定例です。 「29300, 1000000, 1250000, 2500000, 5000000, 10000000, 15000000, 30000000」
SPISlave	void frequency(int hz)	29300～15000000 ※	相手デバイスの AC スペックを考慮のうえ、電気的特性を満足するビットレートを設定して下さい。 以下はビットレートの設定例です。 「29300, 1000000, 1250000, 2500000, 5000000, 10000000, 15000000」

Table 2.2-6 対応クラス設定可能パラメータ 3

※.SPI、SPISlave 通信のビットレート設定を、補足にある設定例以外のパラメータで設定する場合、以下の計算式で、n, N がそれぞれ値域に収まるようなビットレートを設定してください。

$$B = \frac{PCLK}{2 \times (n+1) \times 2^N}$$

(B=ビットレート[bps]、PCLK=120[MHz]、 $0 \leq n \leq 255$ 、 $0 \leq N \leq 3$)

上の計算式を満足するビットレートであっても誤差による影響で通信が正常に行えないことがあります。設定をしようとしているビットレートで実際に通信を行ってみて問題がないかどうかを確認してください。

詳細は「RX64M グループ ユーザーズマニュアル ハードウェア編」をご参照下さい。

クラス	コンストラクタ/メンバ関数	設定可能パラメータ	補足
I2C	void frequency(int hz)	7300~400000 ※	SCL クロックのデューティ比は50%です。 以下は、転送速度の設定例です。 「7300, 10000, 50000, 100000, 400000」

Table 2.2-7 対応クラス設定可能パラメータ 4

※.I2C 通信の転送速度を、補足にある設定例以外のパラメータで設定する場合、
以下の計算式で、n, N がそれぞれ値域に収まるような転送速度を設定してください。

$$B = \frac{1}{\frac{2 \times (n + 1)}{IIC_{\phi}} + tr + tf}$$

$$IIC_{\phi} = \frac{PCLK}{2^N}$$

(B=ビットレート[bps]、PCLK=60[MHz]、tr=tf=300[ns]、0≤n≤31、0≤N≤7)

上の計算式を満足する転送速度であっても、誤差による影響で通信が正常に行えないことがあります。
設定をしようとしている転送速度で実際に通信を行ってみて問題がないかどうかを確認してください。

詳細は「RX64M グループ ユーザーズマニュアル ハードウェア編」をご参照下さい。

クラス	コンストラクタ/メンバ関数	設定可能パラメータ	補足
TimeOut	void attach(void (*fptr)(void), float t)	0.00001～	<p>10[us]から設定可能です。 1[us]単位の設定はできません。 例: 5[us], 15[us]等は設定不可。</p> <p>メンバ関数を実行するタイミングによって、指定関数の発生タイミングが最大 10us 前後します。 ※1</p>
	void attach_us(void (*fptr)(void), unsigned int t)	10～	
Ticker	void attach(void (*fptr)(void), float t)	0.00001～	<p>10[us]から設定可能です。 1[us]単位の設定はできません。 例: 5[us], 15[us]等は設定不可。</p> <p>メンバ関数を実行するタイミングによって、指定関数の初回発生タイミングが最大 10us 前後します。※1</p>
	void attach_us(void (*fptr)(void), unsigned int t)	10～	
InterruptIn	void mode(PinMode pull)	PullNone PullUp OpenDrain PullDefault (PullNone)	-

Table 2.2-8 対応クラス設定可能パラメータ 5

※1. 確実に 10us 以上待機してから指定関数を実行したい場合は、20us 以上を設定するようにしてください。

2.2.3 対応クラス取得可能パラメータ

各クラスにおいて取得可能なパラメータの一覧を以下に示します。

クラス	コンストラクタ/メンバ関数	取得可能パラメータ	補足
PortIn	int read()	0x00～0xFF	-
PortOut	int read()	0x00～0xFF	-
PortInOut	int read()	0x00～0xFF	-
AnalogIn	unsigned short read_u16()	0x0000～0x0FFF	ADC は 12 ビット分解能です。
Timer	float read(void)	0.00001～	10[us]から取得可能です。 1[us]単位の取得はできません。 例: 5[us], 15[us]等は取得不可。
	int read_ms(void)	1～	
	int read_us(void)	10～	

Table 2.2-9 対応クラス取得可能パラメータ

2.3 ピンアサイン

AP-RX64M-0A 上で α RPL の各ライブラリ機能を動作させる際のピンアサインを以下に示します。

α RPL を使用し動作を行う際は、以下のピンアサインを参考に使用ピンを指定して下さい。

No.	信号名	対応機能	No.	信号名	対応機能
1	+5V	-	2	+5V	-
3	VCC	-	4	VCC	-
5	GND	-	6	GND	-
7	D15	-	8	D14	-
9	D13	-	10	D12	-
11	D11	-	12	D10	-
13	D9	-	14	D8	-
15	D7	-	16	D6	-
17	D5	-	18	D4	-
19	D3	-	20	D2	-
21	D1	-	22	D0	-
23	GND	-	24	GND	-
25	PA0	-	26	A1	-
27	A2	-	28	A3	-
29	A4	-	30	A5	-
31	A6	-	32	A7	-
33	A8	-	34	A9	-
35	A10	-	36	A11	-
37	A12	-	38	A13	-
39	A14	-	40	A15	-
41	A16	-	42	A17	-
43	A18	-	44	PC3	デジタル入力/出力/入出力
45	PC4/SSLA0	デジタル入力/出力/入出力 SPI マスタ/スレーブ	46	PC5/ RSPCKA	デジタル入力/出力/入出力 SPI マスタ/スレーブ
47	PC6/MOSIA	デジタル入力/出力/入出力 SPI マスタ/スレーブ	48	P61/SDCS#	-
49	P62/RAS#	-	50	P63/CAS#	-
51	P64/WE#	-	52	P65/CKE	-
53	P66/DQM0	-	54	P67/DQM1	-
55	P50	-	56	P51	デジタル入力/出力/入出力
57	P52	-	58	P53/BCLK	-
59	CPU_VBAT	-	60	P70/SDCLK	-

Table 2.3-1 拡張コネクタ CN1 ピンアサイン

No.	信号名	対応機能	No.	信号名	対応機能
1	P10/USBA_OVRCURA	-	2	P11/USBA_VBUS	-
3	P12/SCL0	I2C	4	P13/SDA0	I2C
5	P14/MTIOC3A	デジタル入力/出力/入出力 PWM 出力	6	P15	デジタル入力/出力/入出力
7	P16/USB0_VBUS	-	8	P17	デジタル入力/出力/入出力
9	P90	-	10	P91	-
11	P93	デジタル入力/出力/入出力	12	P96	デジタル入力/出力/入出力
13	P97	デジタル入力/出力/入出力	14	VCC	-
15	PG7	デジタル入力/出力/入出力	16	PG6	デジタル入力/出力/入出力
17	PG5	デジタル入力/出力/入出力	18	PG2	デジタル入力/出力/入出力
19	P73	-	20	nRESET	-
21	PF5/IRQ4	デジタル入力/出力/入出力 IRQ	22	PC7/MISOA	デジタル入力/出力/入出力 SPI マスタ/スレーブ
23	PJ3/MTIOC3C	デジタル入力/出力/入出力 PWM 出力	24	PJ5	デジタル入力/出力/入出力
25	P26/MTIOC2A	デジタル入力/出力/入出力 PWM 出力	26	P27	デジタル入力/出力/入出力
27	P24	デジタル入力/出力/入出力	28	P25	デジタル入力/出力/入出力
29	P22	デジタル入力/出力/入出力	30	P23	デジタル入力/出力/入出力
31	P20	デジタル入力/出力/入出力	32	P21	デジタル入力/出力/入出力
33	P87	デジタル入力/出力/入出力	34	P86	デジタル入力/出力/入出力
35	P34/MTIOC0A	デジタル入力/出力/入出力 PWM 出力	36	P35/UPSEL	デジタル入力
37	P32	デジタル入力/出力/入出力	38	P33	デジタル入力/出力/入出力
39	RES#	-	40	EXRES	-

Table 2.3-2 拡張コネクタ CN2 ピンアサイン 1

No.	信号名	対応機能	No.	信号名	対応機能
41	GND	-	42	GND	-
43	P00/TXD6	デジタル入力/出力/入出力 シリアル	44	P01/RXD6	デジタル入力/出力/入出力 シリアル
45	P02/IRQ10	デジタル入力/出力/入出力 IRQ	46	P03/DA0	デジタル入力/出力/入出力 アナログ出力
47	P05/DA1	デジタル入力/出力/入出力 アナログ出力	48	P07/IRQ15	デジタル入力/出力/入出力 IRQ
49	VREFH0	-	50	VREFL0	-
51	P40/AN000	デジタル入力/出力/入出力 アナログ入力	52	P41/AN001	デジタル入力/出力/入出力 アナログ入力
53	P42/AN002	デジタル入力/出力/入出力 アナログ入力	54	P43/AN003	デジタル入力/出力/入出力 アナログ入力
55	P44/AN004	デジタル入力/出力/入出力 アナログ入力	56	P45/AN005	デジタル入力/出力/入出力 アナログ入力
57	P46/AN006	デジタル入力/出力/入出力 アナログ入力	58	P47/AN007	デジタル入力/出力/入出力 アナログ入力
59	AVCC	-	60	AGND	-

Table 2.3-3 拡張コネクタ CN2 ピンアサイン 2

No.	信号名	対応機能	No.	信号名	対応機能
1	PF0/TXD1 P00/TXD6 ※	シリアル	2	PF2/RXD1 P01/RXD6 ※	シリアル
3	-	-	4	-	-
5	VCC	-	6	GND	-

※.DIPSW で接続端子を切替可能

Table 2.3-4 シリアル IF 用コネクタ CN6 ピンアサイン

2.4 ライブラリ使用メモリ

α RPL を AP-RX64M-0A で動作させる際は、以下のメモリ領域を α RPL 用に確保する必要があります。
必要なメモリ領域が別の用途で使用されていた場合、 α RPL は正しく動作を行うことはできません。

アドレス	用途	備考
0x0001F000～ 0x0001F3FF	割り込みベクタテーブル	内蔵 RAM 領域
0xFFFFF80～ 0xFFFFF8B	例外ベクタテーブル	内蔵 ROM 領域
0xFFFFF8C～ 0xFFFFFFF	リセットベクタ	内蔵 ROM 領域

Table 2.4-1 α RPL AP-RX64M-0A ライブラリ使用領域

また、 α RPL では、スタック領域を、以下のサイズ分確保しています。
アプリケーションを作成する際は、下記のスタック領域サイズを超過しないように注意して下さい。

名称	サイズ	セクション	内容
ユーザスタック領域	0x1800	SU	プログラム実行に必要な領域
割り込みスタック領域	0x1800	SI	割り込み実行に必要な領域

Table 2.4-2 α RPL AP-RX64M-0A スタック使用領域

3. 動作手順

3.1 サンプル動作手順

ライブラリ機能のデジタル出力(DigitalOut クラス)のサンプルコードを例に、
α RPL を使用したファームウェア開発の手順を解説します。

3.1.1 動作モード

AP-RX64M-0A では、使用する動作モードに応じてスイッチを設定する必要があります。以下に動作モードの設定例を示します。

① オンチップデバッグエミュレータ(E1 エミュレータ/E2 エミュレータ Lite)を使用する場合

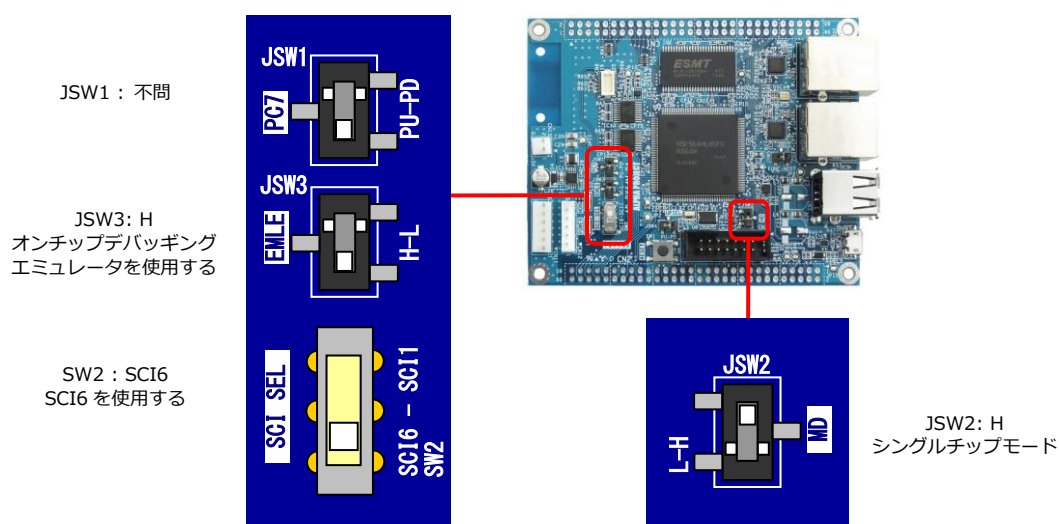


Fig 3.1-1 オンチップデバッグエミュレータを使用する場合の設定

② シリアル経由でプログラムを書き込む場合(Renesas Flash Programmer 等を使用する場合)

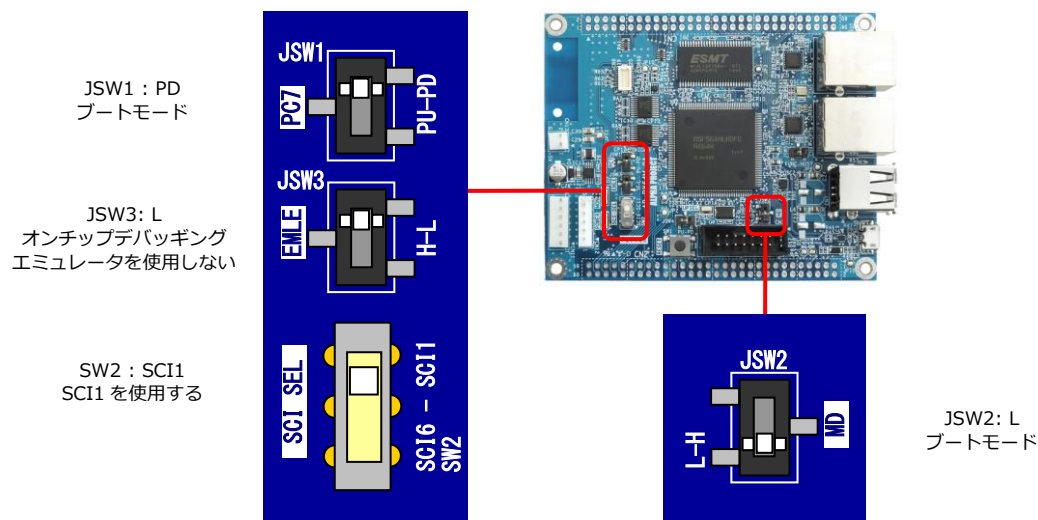


Fig 3.1-2 シリアル経由でプログラムを書き込む場合の設定

③ プログラムを動作させる場合

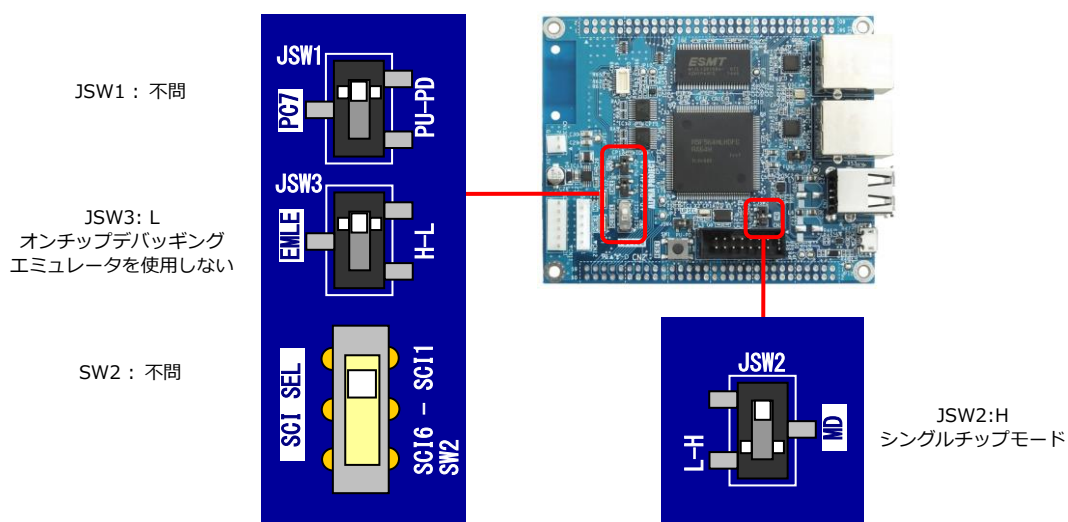


Fig 3.1-3 プログラムを動作させる場合の設定

④ USB ブートモードでプログラムを書き込む場合(Renesas Flash Programmer を使用する場合)

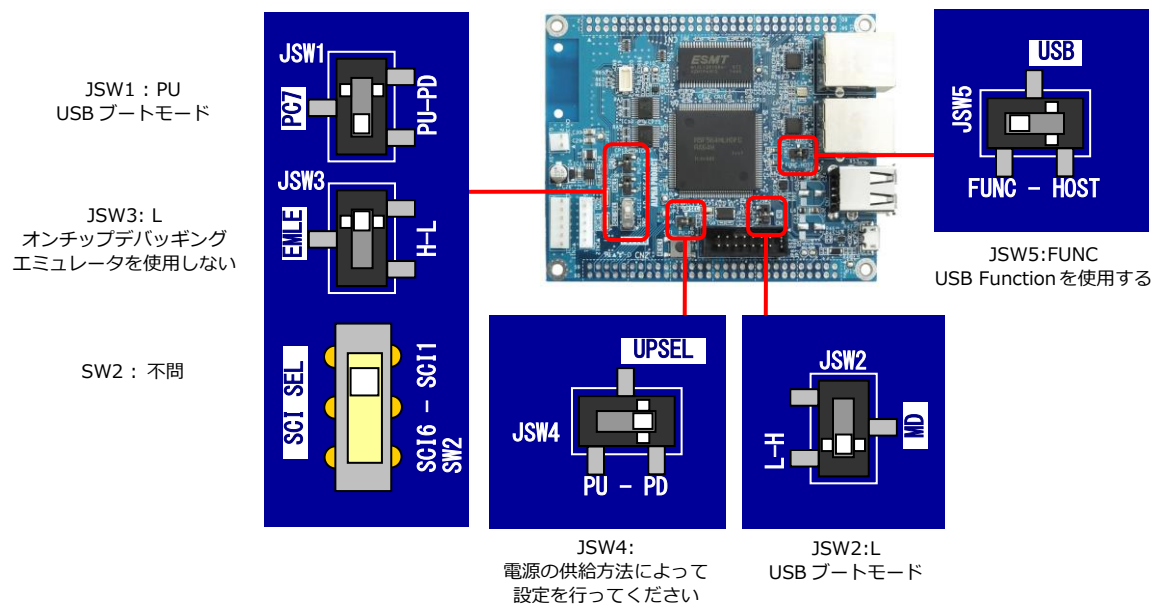


Fig 3.1-4 USB ブートモードでプログラムを書き込む場合の設定

3.1.2 デジタル出力のサンプルコード

サンプルコードは「1 秒周期で LD2 を点滅させるプログラム」です。

以下のサンプルコードを参考に「main.cpp」にプログラムを記述して下さい。

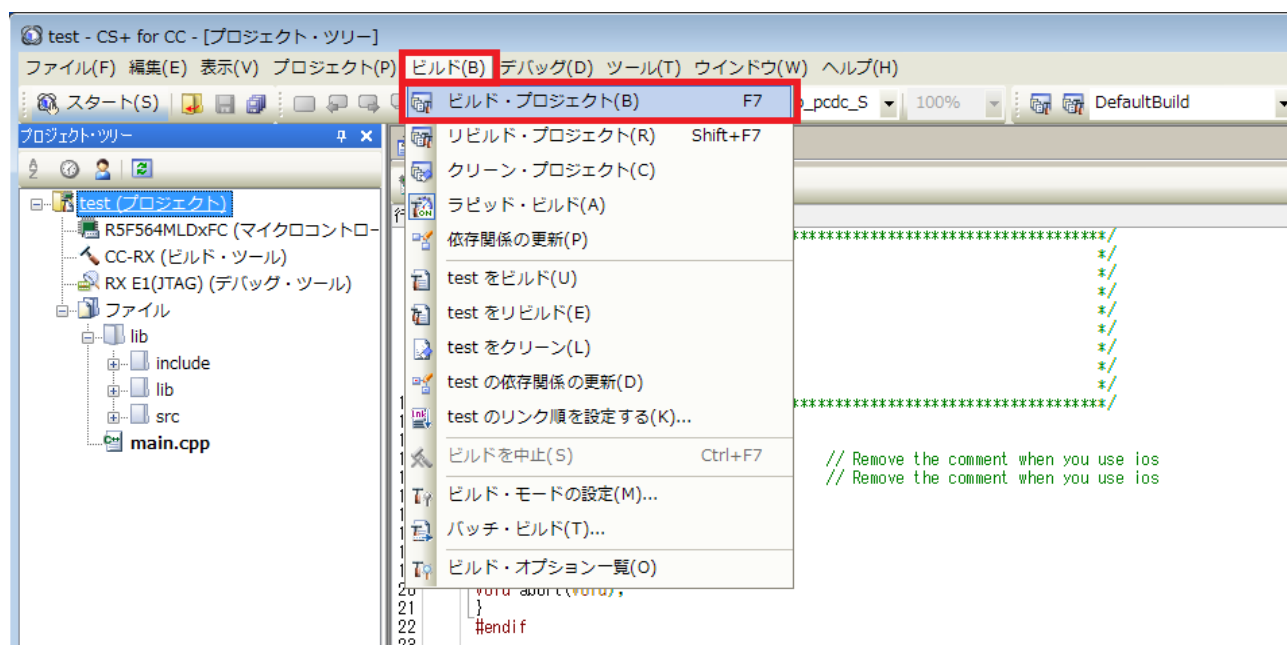
```
void main( void )          /* メインルーチン */
{
    DigitalOut led(PG2);    /* デジタル出力初期化(LD2) */
    while(1)
    {
        led = ! led;        /* LED の点滅 */
        wait(0.5);          /* 0.5 秒ウェイト */
    }
}
```

デジタル出力サンプルコード

3.1.3 ビルド・ダウンロード

(1) ビルド

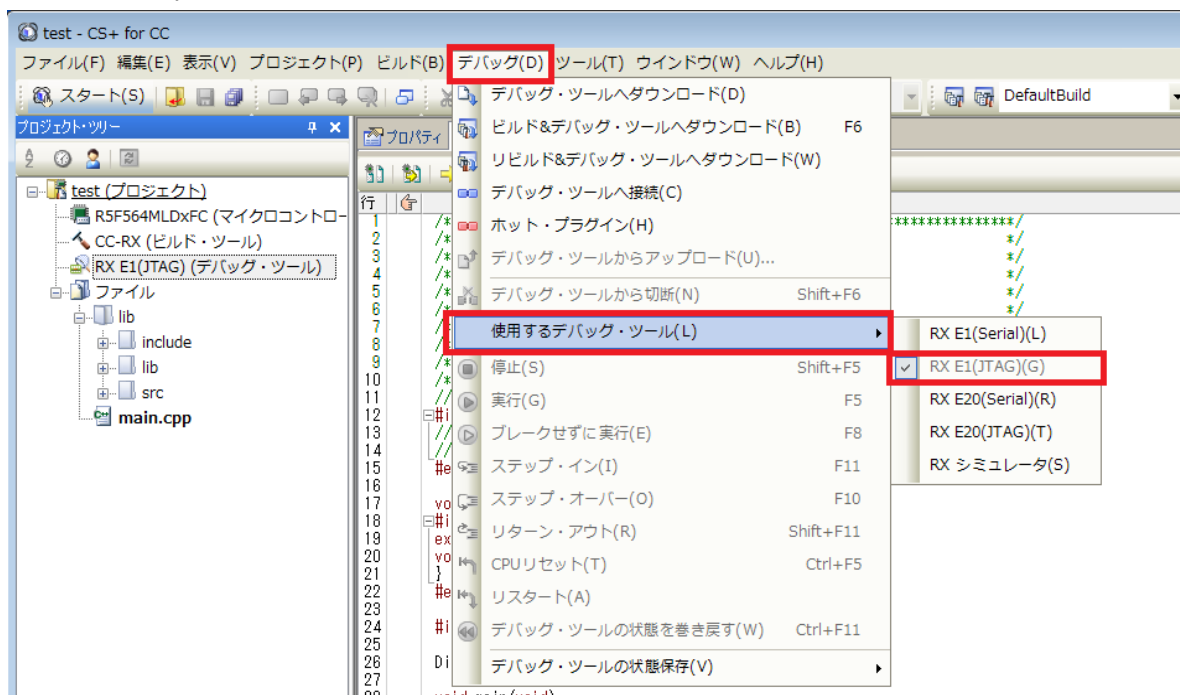
- ① CS+を起動し、ユーザで使用するプロジェクトファイル（.mtpj）を読み込みます。
- ② プログラムをビルドします。CS +のメニューから「ビルド(B)」 → 「ビルド・プロジェクト(B)」を実行して下さい。



- ③ デフォルトのビルド・モードは「DefaultBuild」となっており、¥DefaultBuild ワークフォルダ内にモトローラファイル(.mot)、アブソリュートファイル(.abs)、マップファイル(.map) が出力されます。
ビルド・モードの設定はメニュー「ビルド(B)」 → 「ビルド・モードの設定(M)」からビルド・モードの設定ウィンドウで変更できます。

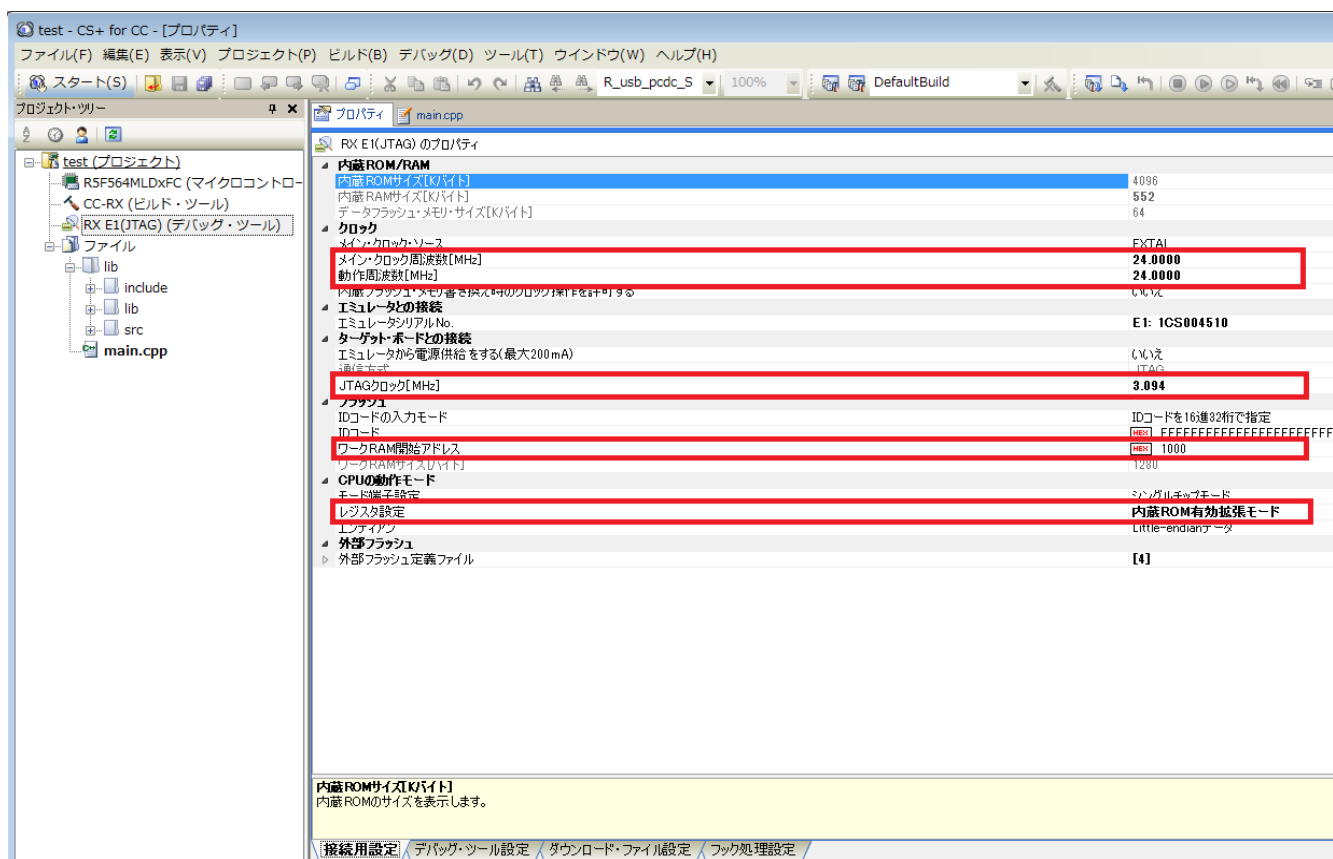
(2-1) E1 エミュレータ/E2 エミュレータ Lite を使用したダウンロード

- ① 3.1.1 動作モードの①「オンチップデバッグエミュレータ (E1 エミュレータ/E2 エミュレータ Lite) を使用する場合」を参考に AP-RX64M-0A の設定を行い、PC と AP-RX64M-0A を E1 エミュレータ、もしくは、E2 エミュレータ Lite を介して接続します。
- ② CS+を起動し、プロジェクトファイルを読み込みます。
- ③ メニュー「デバッグ(D)」→「使用するデバッグ・ツール(L)」→「RX E1(JTAG)(G)」を選択します。(E1 エミュレータを使用した場合)



- ④ CPU 依存部分の設定を行います。デバッグ・ツールのプロパティを開き、「接続用設定タブ」から以下の設定値を参考に設定して下さい。

- ・メイン・クロック・ソース：24MHz
- ・JTAG クロック：3.094MHz
- ・ワーク RAM 開始アドレス：1000
- ・レジスタ設定：内蔵 ROM 有効拡張モード



- ⑤ AP-RX64M-0A へ電源を投入し、メニュー「デバッグ(D)」→「デバッグ・ツールへ接続(C)」を選択し、続けて「デバッグ(D)」→「デバッグ・ツールへダウンロード(D)」を選択します。



- ⑥ プログラムのダウンロードが完了しました。E1 エミュレータ、もしくは、E2 エミュレータ Lite でデバッグを行い、1 秒周期で LED が点滅することを確認して下さい。

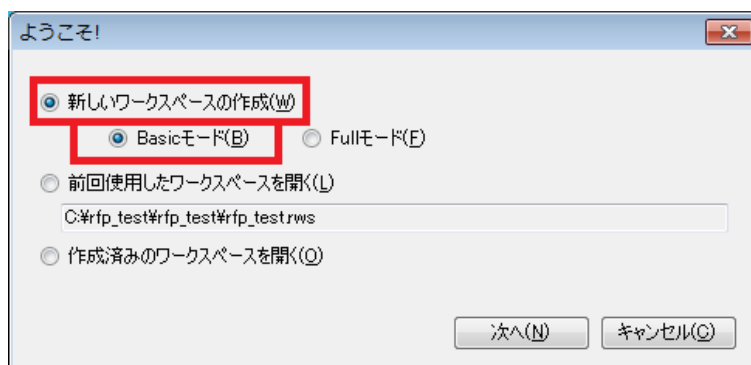
ビルド方法および AP-RX64M-0A にプログラムをダウンロードする詳細な方法については「AN1526 RX 開発環境の使用方法 (CS+、Renesas Flash Programmer)」を参照して下さい。

また、E1 エミュレータのデバッグ機能の詳細などに関しては「E1/E20 エミュレータユーザズマニュアル」を参照して下さい。

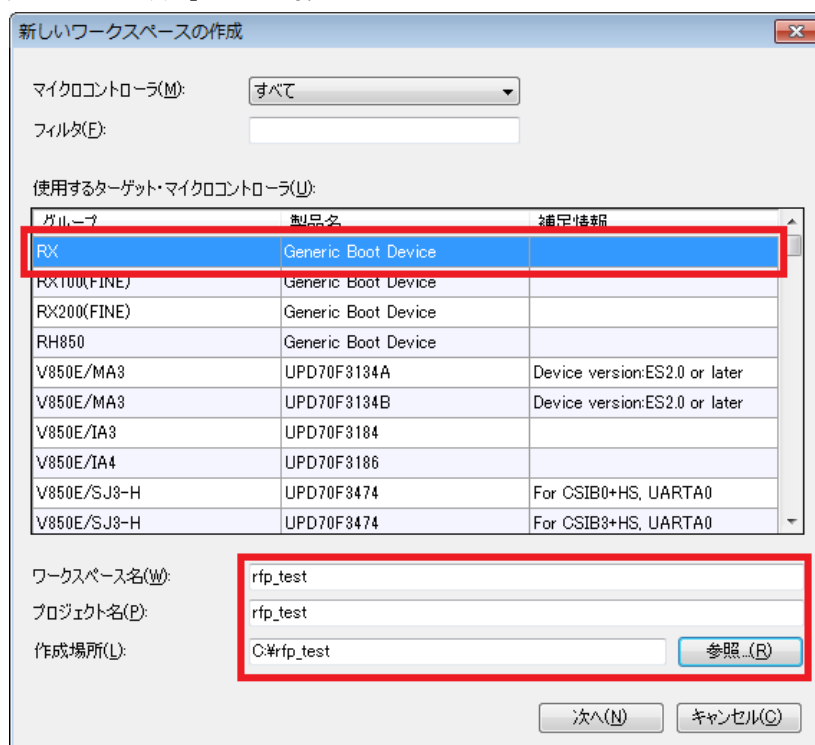
E2 エミュレータ Lite のデバッグ機能の詳細などに関しては「E2 エミュレータ Lite ユーザズマニュアル」を参照して下さい。

(2-2) Renesas Flash Programmer を使用したダウンロード

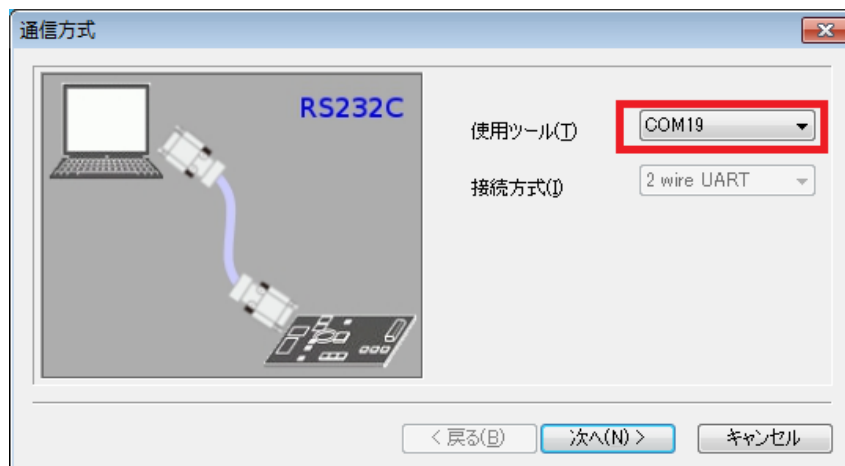
- ① 3.1.1 動作モードの②「シリアル経由でプログラムを書き込む場合」を参考に AP-RX64M-0A の設定を行い、PC と AP-RX64M-0A を PC-USB-04 を介して接続します。
- ② Renesas Flash Programmer (以下、RFP) を起動します
- ③ RFP を起動すると、以下のようなウィンドウが表示されますので、「新しいワークスペースの作成」と「Basic モード」を選択し、「次へ」ボタンをクリックします。



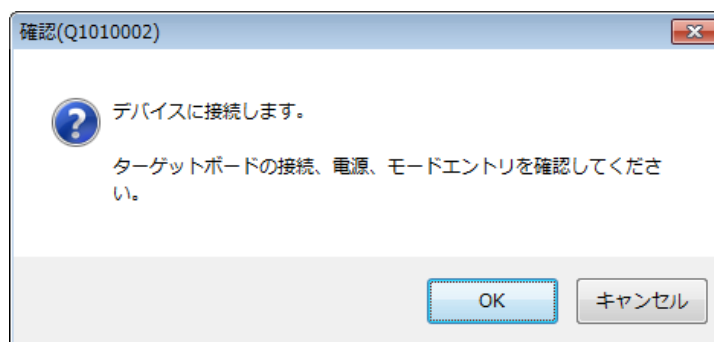
- ④ 新しいワークスペースの作成ウィンドウが表示されますので、使用するターゲット・マイクロコントローラを RX に設定します。次に、ワークスペース名及びプロジェクト名を設定します。その後、「参照」ボタンを押し、作成場所を選択します。
(※ ワークスペース名、プロジェクト名および作成場所は、ユーザの環境に合わせて任意で設定可能です)
全ての設定が完了したら 「次へ」 ボタンを押します。



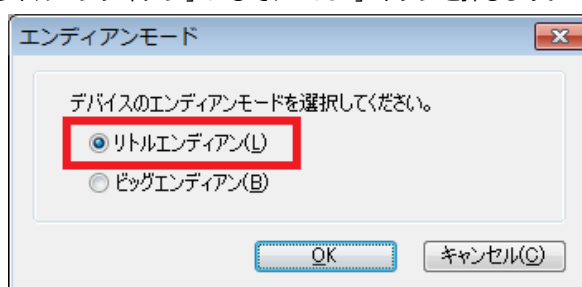
- ⑤ 通信方式のウィンドウが表示されますので、AP-RX64M-0A で使用する通信ポートに合わせて使用ツール(通信ポート)を選択します。選択したら、「次へ」ボタンを押します。



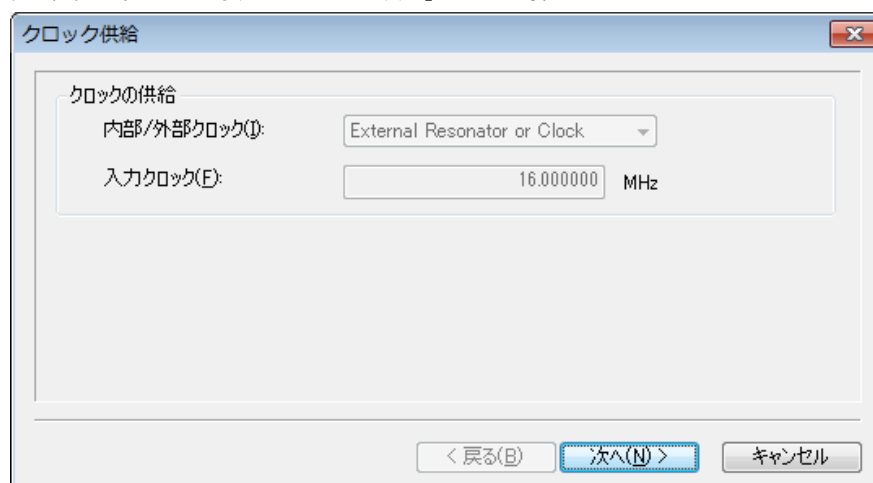
- ⑥ 確認ウィンドウが表示されたら、AP-RX64M-0A へ電源を投入します。
電源を投入したら、「OK」ボタンを押します。



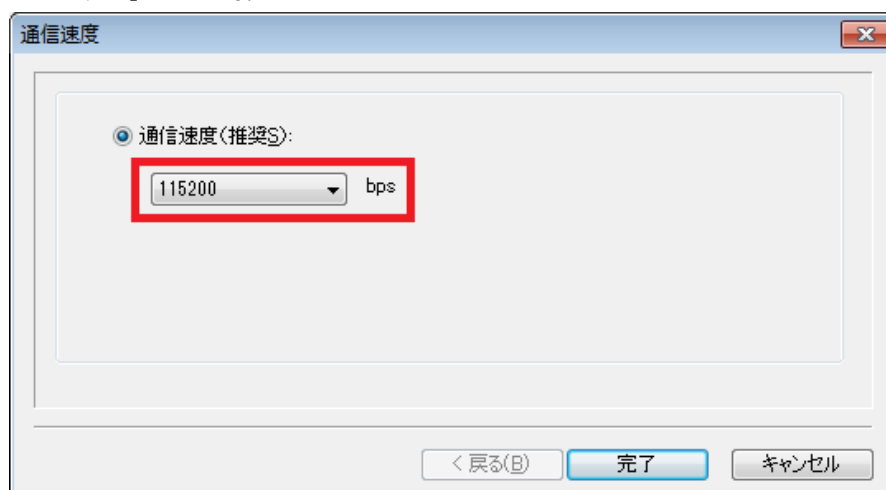
- ⑦ デバイス確認が開始されると、エンディアンモードウィンドウが表示されます。
エンディアンモードを「リトルエンディアン」にして、「OK」ボタンを押します。



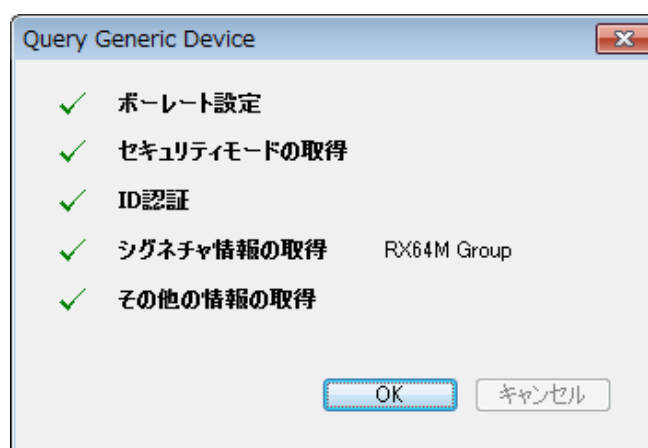
- ⑧ 続いて、クロック供給ウィンドウが表示されます。「次へ」ボタンを押します。



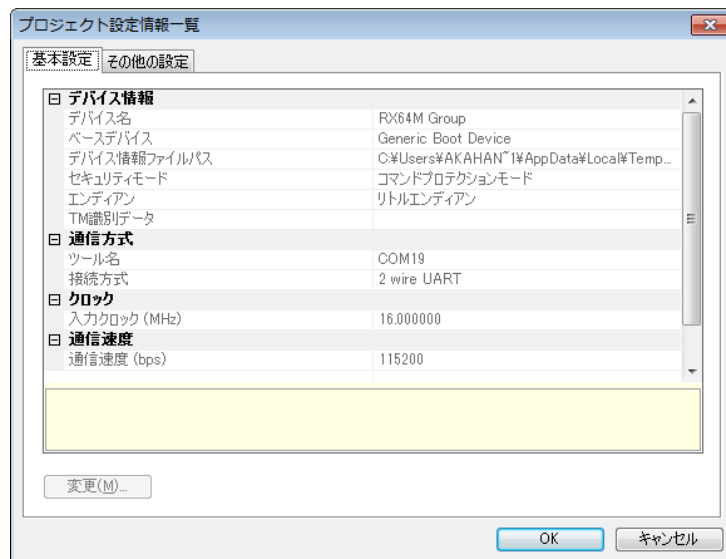
- ⑨ 次に、通信速度ウィンドウが表示されます。「115200」bps を選択します。
選択が完了したら、「完了」ボタンを押します。



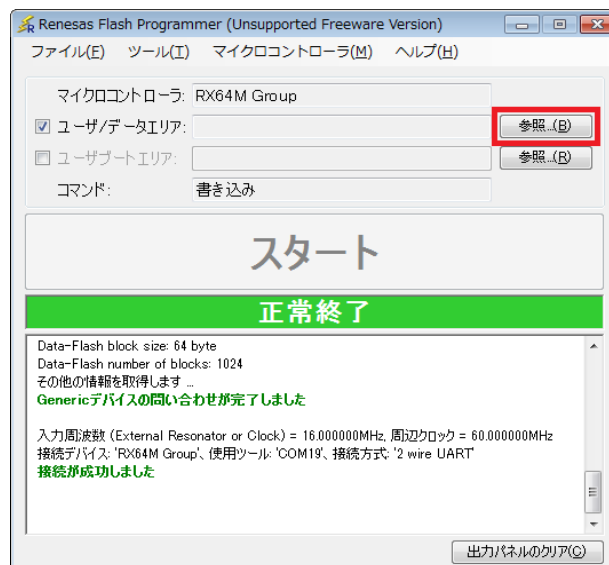
- ⑩ デバイス確認が完了すると、Query Generic Device ウィンドウで「OK」ボタンが押せるようになりますので、「OK」ボタンを押します。
デバイスの確認中にエラーが発生した場合は、AP-RX64M-0A の設定や通信ポートの接続などを確認し、再度デバイスの確認を行って下さい。



- ⑪ 次に、プロジェクト設定情報一覧ウィンドウが表示されます。「OK」ボタンを押します。



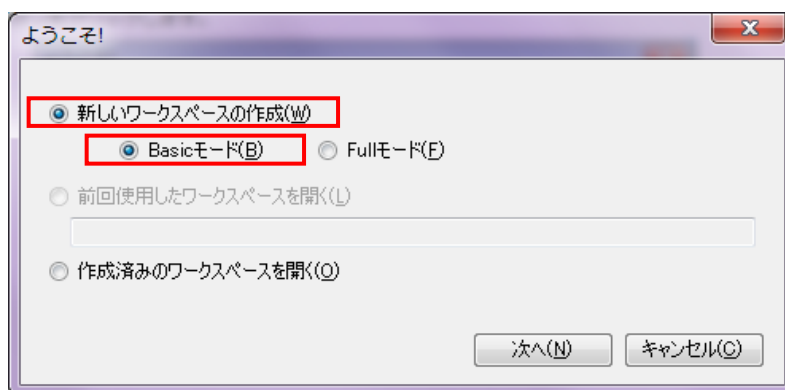
- ⑫ 設定が完了すると、デバイスへの接続が開始されます。正常に接続が完了すると、RFP のログに「接続が成功しました」と表示されます。
- ⑬ 接続が成功したら、ツールヘダダウンロードするファイルを選択します。「参照」ボタンからダウンロードファイルを指定します。ダウンロードファイルは、ビルド時に作成したモトローラファイル (.mot) を指定します。指定ができれば「スタート」ボタンを押します。



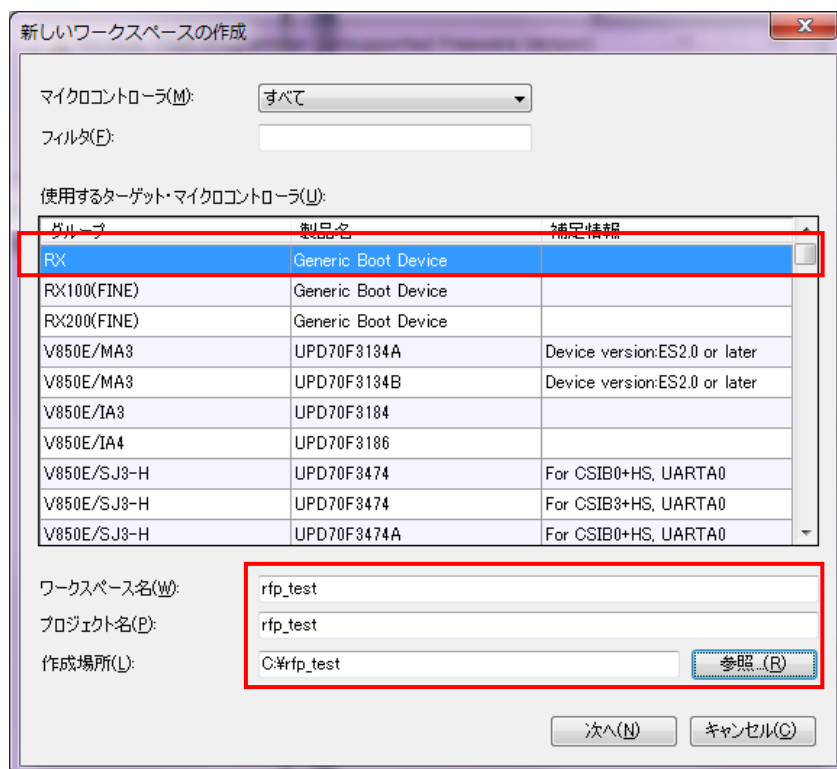
- ⑭ 正常に書き込みが完了すると、RFP のログに「書き込みが完了しました」と表示されます。
AP-RX64M-0A の電源を切ります。その後 RFP を終了します。以上でプログラムのダウンロードは完了です。
- ⑮ 3.1.1 動作モードの③「プログラムを動作させる場合」を参考に AP-RX64M-0A の設定を変更し、再度電源を投入して、1 秒周期で LED が点滅することを確認します。

(2-3) USB ブートモードで Renesas Flash Programmer を使用したダウンロード

- ① 3.1.1 動作モードの④「USB ブートモードでプログラムを書き込む場合」を参考に AP-RX64M-0A の設定を行い、PC と AP-RX64M-0A を USB micro ケーブルを用いて接続します。
- ② Renesas Flash Programmer (以下、RFP) を起動します。
- ③ RFP を起動すると、以下のようなウィンドウが表示されますので、「新しいワークスペースの作成」と「Basic モード」を選択し、「次へ」ボタンをクリックします。



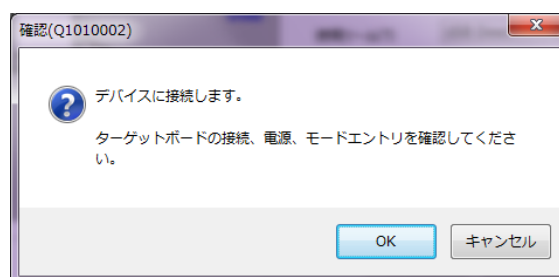
- ④ 新しいワークスペースの作成ウィンドウが表示されますので、使用するターゲット・マイクロコントローラを RX に設定します。次に、ワークスペース名およびプロジェクト名を設定します。その後「参照」参照ボタンを押し、作業場所を選択します。
(※ ワークスペース名、プロジェクト名および作成場所は、ユーザの環境に合わせて任意で設定可能です)
すべての設定が完了したら「次へ」ボタンを押します。



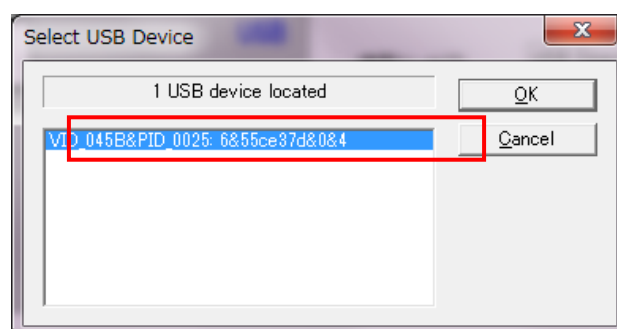
- ⑤ 通信方式のウィンドウが表示されますので、使用ツールに「USB Direct」を選択します。
選択したら、「次へ」ボタンを押します。



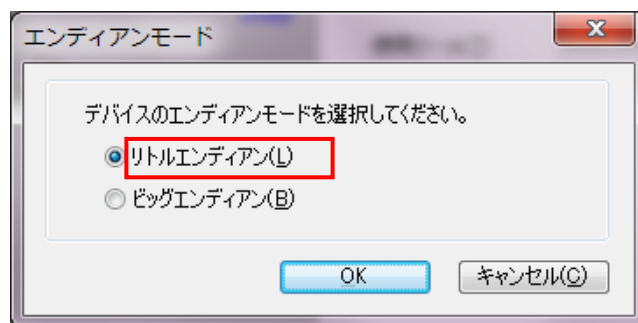
- ⑥ 確認ウィンドウが表示されたら、AP-RX64M-0A へ電源を投入します。
電源が投入されていたら「OK」ボタンを押します。



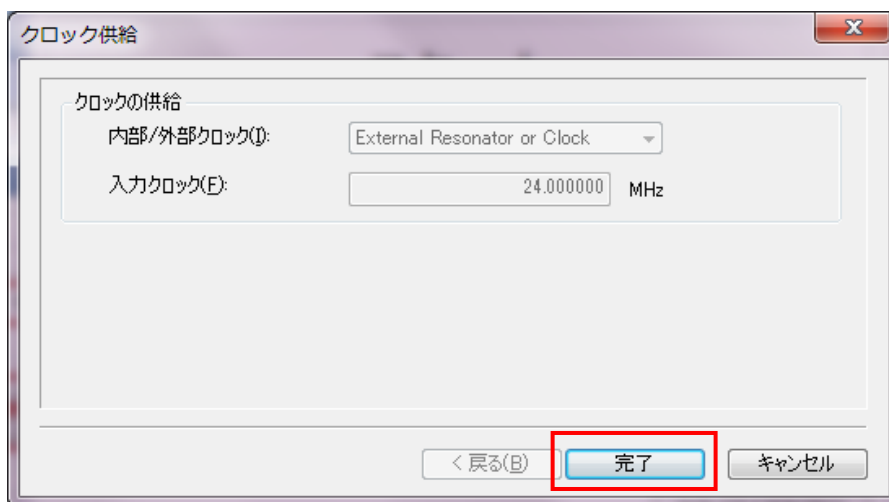
- ⑦ ⑥の確認ウィンドウの次に Select USB Device ウィンドウが表示されることがあります。
PC と AP-RX64M-0A とを結ぶ USB micro ケーブルを選択して、「OK」ボタンを押します。



- ⑧ デバイス確認が開始されると、エンディアンモードウィンドウが表示されます。
ラジオボタンを「リトルエンディアン」に設定して、「OK」ボタンを押します。



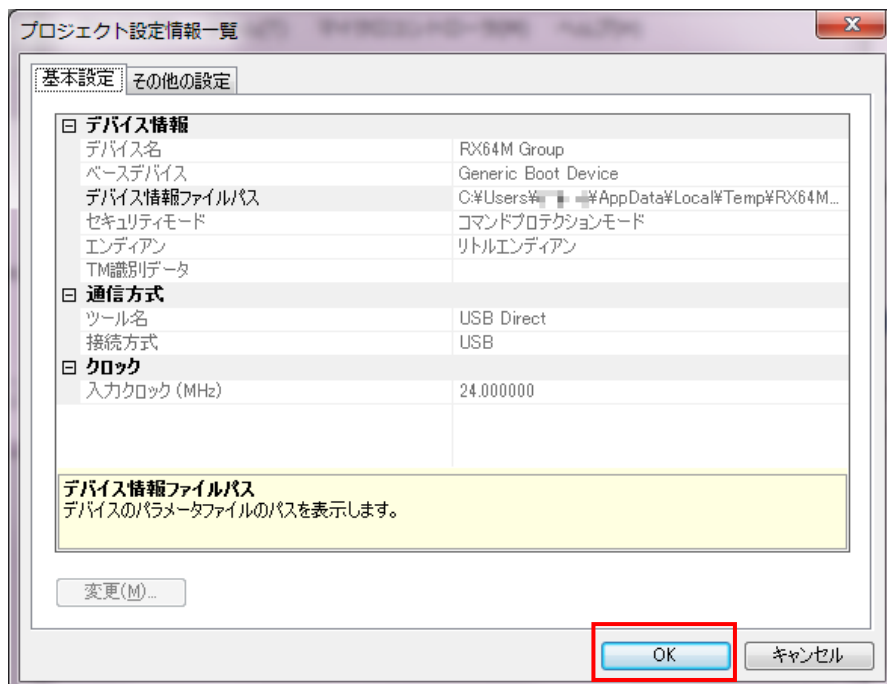
- ⑨ クロック供給ウィンドウが表示されます。「完了」ボタンを押します。



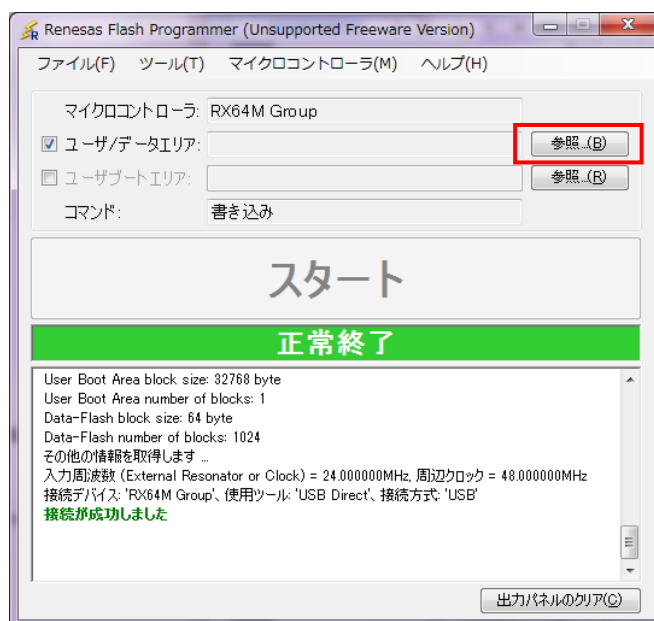
- ⑩ デバイス確認が完了すると、Query Generic Device ウィンドウで「OK」ボタンが押せるようになりますので、「OK」ボタンを押します。
デバイスの確認中にエラーが発生した場合は、AP-RX64M-0A の設定や通信ポートの接続などを確認し、再度デバイスの確認を行ってください。



- ⑪ 次にプロジェクト設定情報一覧ウィンドウが表示されます。「OK」ボタンを押します。



- ⑫ 設定が完了すると、デバイスへの接続が開始されます。
正常に接続が完了すると、RFP のログに「接続が成功しました」と表示されます。
- ⑬ 接続が成功したら、ツールヘダダウンロードするファイルを選択します。
ユーザ/データエリア右の「参照」ボタンからダウンロードファイルを指定します。
ダウンロードファイルは、ビルド時に作成したモトローラファイル (.mot) を指定します。
指定ができれば画面中央の「スタート」ボタンが押せるようになりますのでボタンを押します。



- ⑭ 正常に書き込みが完了すると、RFP のログに「書き込みが完了しました」と表示され、接続が切断されます。
AP-RX64M-0A の電源を切り、その後 RFP を終了します。
以上でプログラムのダウンロードは完了となります。

- ⑮ 3.1.1 動作モードの③「プログラムを動作させる場合」を参考に AP-RX64M-0A の設定を変更し、
再度電源を投入して、1 秒周期で LED が点滅することを確認します。

RFP の詳細などに関しては「[RH850 RX700(RX64M 含む)]Renesas Flash Programmer V2.05 ユーザーズマニュアル」を参照して下さい。

改定履歴

版数	日付	改定内容
1 版	2015/08/04	新規作成
2 版	2021/01/27	自社ホームページの URL を更新
2.1 版	2023/10/02	Table 1.2-1 使用環境(AP-RX64M-0A) 更新 (1 章) 動作手順更新を更新(3 章) 住所の更新

参考文献

「RX64M グループ ユーザーズマニュアル ハードウェア編」
その他 各社データシート

ルネサス エレクトロニクス株式会社

本文書について

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
- ・本文書の内容に基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

商標について

- ・RX および RX64M は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
 - ・CS+（旧CubeSuite+）は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
 - ・E1エミュレータは、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
 - ・E2エミュレータLiteは、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
 - ・Renesas Flash Programmerは、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
-
- ・Windows®の正式名称は Microsoft®Windows®Operating System です。
 - ・Microsoft、Windowsは、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
 - ・Windows®10、Windows®11は、米国 Microsoft Corporation.の商品名称です。
本文書では下記のように省略して記載している場合がございます。ご了承ください。
Windows®10は Windows 10もしくは Win10
Windows®11は Windows 11もしくは Win11
-
- ・その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市中央区積志町 834
<https://www.apnet.co.jp>
E-Mail : query@apnet.co.jp