

---

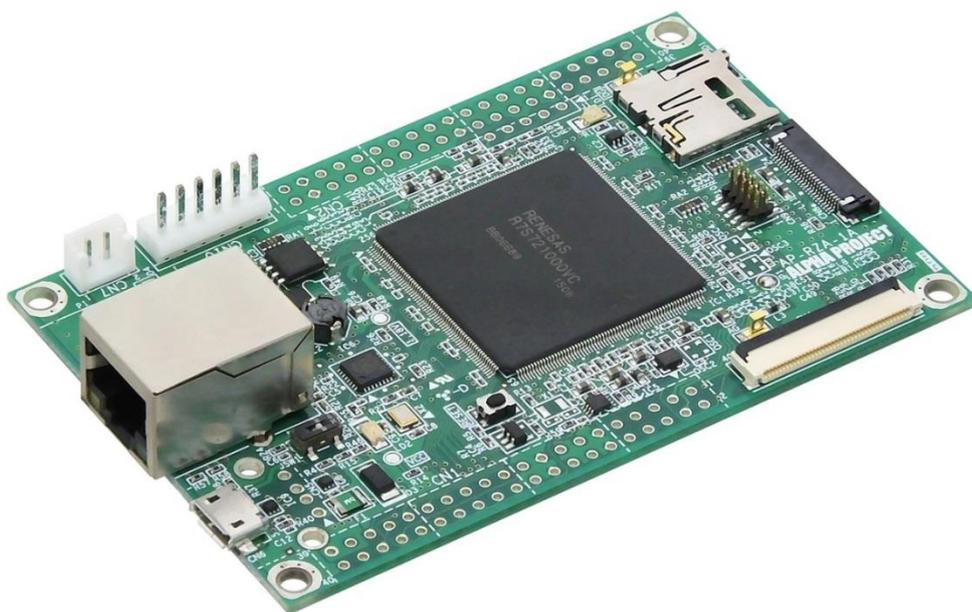
# α RPL

---

対応ボード使用手順

AP-RZA-1A 編

2.1 版



## 目次

<b>1. 概要</b>	<b>1</b>
1.1 概要.....	1
1.2 使用環境.....	1
<b>2. 機能</b>	<b>1</b>
2.1 CPU 初期設定 .....	2
2.2 ボード対応機能一覧.....	7
2.3 ピンアサイン.....	16
2.4 ライブラリ使用メモリ.....	18
<b>3. 動作手順</b>	<b>18</b>
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-RZA-1A 編（以下、本マニュアル）では、

「2. 機能」で α RPL のうち AP-RZA-1A 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-RZA-1A 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（LD1）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「4. ライブラリ組み込み手順(RZ 編)」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下さい。

## 1.2 使用環境

α RPL を組み込んで AP-RZA-1A 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 e <sup>2</sup> studio	4.0.1.007	統合開発環境	ルネサス エレクトロニクス 株式会社製品
KPIT GNUARM-NONE-EABI Toolchain	14.02	コンパイル、ビルド	
J-Link		デバッグ	SEGGER 社製品
FlashWriter EX / RZ Limited Edition	1.0	FlashROM 書き込み	弊社サイトより入手可能

Table 1.2-1 使用環境(AP-RZA-1A)

## 2. 機能

### 2.1 CPU 初期設定

α RPL 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
CPU クロック(Iφ)	384[MHz]
画像処理クロック(Gφ)	256[MHz]
内部バスクロック(Bφ)	128[MHz]
周辺クロック 1(P1φ)	64[MHz]
周辺クロック 0(P0φ)	32[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
JP0_0	-
JP0_1	-
P0_0	-
P0_1	-
P0_2	-
P0_3	-
P0_4	-
P0_5	-
P1_0	-
P1_1	-
P1_2	-
P1_3	-
P1_4	-
P1_5	-
P1_6	ポートモード/入力
P1_7	ポートモード/入力
P1_8	ポートモード/入力
P1_9	ポートモード/入力
P1_10	ポートモード/入力
P1_11	ポートモード/入力
P1_12	ポートモード/入力
P1_13	ポートモード/入力
P1_14	ポートモード/入力
P1_15	ポートモード/入力

※、「-」は拡張コネクタに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
P2_0	-
P2_1	-
P2_2	-
P2_3	-
P2_4	-
P2_5	-
P2_6	-
P2_7	-
P2_8	ポートモード/出力(Low 出力)
P2_9	ポートモード/出力(Low 出力)
P2_10	ポートモード/出力(Low 出力)
P2_11	ポートモード/出力(Low 出力)
P2_12	ポートモード/出力(Low 出力)
P2_13	ポートモード/出力(Low 出力)
P2_14	ポートモード/出力(Low 出力)
P2_15	ポートモード/出力(Low 出力)
P3_0	-
P3_1	-
P3_2	-
P3_3	ポートモード/入力
P3_4	ポートモード/入力
P3_5	ポートモード/入力
P3_6	ポートモード/入力
P3_7	-
P3_8	-
P3_9	-
P3_10	-
P3_11	-
P3_12	-
P3_13	-
P3_14	-
P3_15	-

※. 「-」は拡張コネクタに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
P4_0	-
P4_1	-
P4_2	-
P4_3	-
P4_4	-
P4_5	-
P4_6	-
P4_7	-
P4_8	-
P4_9	-
P4_10	-
P4_11	-
P4_12	-
P4_13	-
P4_14	-
P4_15	-
P5_0	兼用モード/第5兼用機能/出力
P5_1	兼用モード/第5兼用機能/出力
P5_2	ポートモード/出力(Low 出力)
P5_3	ポートモード/出力(Low 出力)
P5_4	ポートモード/出力(Low 出力)
P5_5	ポートモード/出力(Low 出力)
P5_6	ポートモード/出力(Low 出力)
P5_7	ポートモード/出力(Low 出力)
P5_8	-
P5_9	-
P5_10	-

※. 「-」は拡張コネクタに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
P6_0	兼用モード/第1兼用機能/出力
P6_1	兼用モード/第1兼用機能/出力
P6_2	兼用モード/第1兼用機能/出力
P6_3	兼用モード/第1兼用機能/出力
P6_4	兼用モード/第1兼用機能/出力
P6_5	兼用モード/第1兼用機能/出力
P6_6	-
P6_7	-
P6_8	兼用モード/第1兼用機能/出力
P6_9	兼用モード/第1兼用機能/出力
P6_10	兼用モード/第1兼用機能/出力
P6_11	兼用モード/第1兼用機能/出力
P6_12	兼用モード/第1兼用機能/出力
P6_13	兼用モード/第1兼用機能/出力
P6_14	兼用モード/第1兼用機能/出力
P6_15	兼用モード/第1兼用機能/出力
P7_0	-
P7_1	-
P7_2	-
P7_3	-
P7_4	-
P7_5	-
P7_6	-
P7_7	-
P7_8	-
P7_9	-
P7_10	-
P7_11	-
P7_12	-
P7_13	-
P7_14	-
P7_15	-

※. 「-」は拡張コネクタに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
P8_0	-
P8_1	-
P8_2	兼用モード/第 1 兼用機能/出力
P8_3	兼用モード/第 1 兼用機能/出力
P8_4	兼用モード/第 1 兼用機能/出力
P8_5	兼用モード/第 1 兼用機能/出力
P8_6	兼用モード/第 1 兼用機能/出力
P8_7	-
P8_8	ポートモード/出力(Low 出力)
P8_9	ポートモード/出力(Low 出力)
P8_10	ポートモード/出力(Low 出力)
P8_11	ポートモード/出力(Low 出力)
P8_12	ポートモード/出力(Low 出力)
P8_13	ポートモード/出力(Low 出力)
P8_14	ポートモード/出力(Low 出力)
P8_15	ポートモード/出力(Low 出力)
P9_0	ポートモード/入力
P9_1	ポートモード/入力
P9_2	-
P9_3	-
P9_4	-
P9_5	-
P9_6	-
P9_7	-

※. 「-」は拡張コネクタに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-RZA-1A 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-RZA-1A 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-RZA-1A 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)	P1_6~P1_15	-
	DigitalIn [インスタンス名](PinName pin, PinMode mode)	P2_8~P2_15 P3_3~P3_6 P5_0~P5_7 P6_0~P6_5, P6_8~P6_15 P8_2~P8_6, P8_8~P8_15 P9_0~P9_1	
DigitalOut	DigitalOut [インスタンス名](PinName pin)	P1_6~P1_7 P2_8~P2_15	-
	DigitalOut [インスタンス名](PinName pin, int value)	P3_3~P3_6 P5_0~P5_7 P6_0~P6_5, P6_8~P6_15 P8_2~P8_6, P8_8~P8_15 P9_0~P9_1	
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_1 PORT_2 PORT_3 PORT_5 PORT_6 PORT_8 PORT_9	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)	P5_5/TIOC0C P2_11/TIOC1A P6_2/TIOC2A P8_12/TIOC3C	-
AnalogIn	AnalogIn [インスタンス名](PinName pin)	P1_8/AN0~P1_15/AN7	-
Serial	Serial [インスタンス名](PinName tx, PinName rx)	P6_6/TxD5, P6_7/RxD5	-
SPI	SPI [インスタンス名](PinName mosi, PinName miso, PinName sclk)	P2_14/MOSI0, P2_15/MISO0, P2_12/RSPCK0	-
SPIslave	SPIslave [インスタンス名](PinName mosi, PinName miso, PinName sclk, PinName ssel)	P2_14/MOSI0, P2_15/MISO0, P2_12/RSPCK0, P2_13/SSL00	-
I2C	I2C [インスタンス名](PinName sda, PinName scl)	P1_3/SDA1, P1_2/SCL1	-
InterruptIn	InterruptIn [インスタンス名](PinName pin)	P9_1/IRQ0 P8_3/IRQ1 P6_10/IRQ2 P6_11/IRQ3 P6_1/IRQ4 P6_0/IRQ5 P1_6/IRQ6 P1_7/IRQ7	-

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 PullDefault (PullNone)	-
	void mode(PinMode pull)		
DigitalInOut	DigitalInOut [インスタンス名](PinName pin, PinDirection direction, PinMode mode, int value)	PullNone PullDefault (PullNone)	-
	void mode(PinMode pull)		
BusIn	void mode(PinMode pull)	PullNone PullDefault (PullNone)	-
BusInOut	void mode(PinMode pull)	PullNone PullDefault (PullNone)	-
PortIn	PortIn [インスタンス名](PortName port, int mask)	0x0000~0xFFFF	-
	void mode(PinMode pull)	PullNone PullDefault (PullNone)	-
PortOut	PortOut [インスタンス名](PortName port, int mask)	0x0000~0xFFFF	-
	void write(int value)	0x0000~0xFFFF	-
PortInOut	PortInOut [インスタンス名](PortName port, int mask)	0x0000~0xFFFF	-
	void mode(PinMode pull)	PullNone PullDefault (PullNone)	-
	void write(int value)	0x0000~0xFFFF	-
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
Serial	void baud(int baudrate)	300~2000000 ※	以下はボーレートの設定例です。 「300, 2400, 9600, 14400, 19200, 38400, 57600, 115200, 250000, 1000000, 2000000」

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から値域に収まるような n, N の値を求めます。

$$N = \frac{P1\phi}{64 \times 2^{2n-1} \times B} - 1$$

B : ビットレート[bps], P1φ : 64[MHz], 0 ≤ n ≤ 3, 0 ≤ N ≤ 255

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

$$\text{誤差} = \left\{ \frac{P1\phi}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$$

3. 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)	16000~32000000 ※	以下はビットレートの設定例です。 「16000, 100000, 1000000, 4000000, 8000000, 16000000, 32000000」
SPISlave	void frequency(int hz)	16000~8000000 ※	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。 「16000, 100000, 1000000, 4000000, 8000000」

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。

$$B = \frac{P1\phi}{2 \times (n + 1) \times 2^N}$$

B : ビットレート[bps], P1φ : 64[MHz], 0 ≤ n ≤ 255, 0 ≤ N ≤ 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「RZ/A1Hグループ、RZ/A1M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)	5000~400000 ※	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。 「5000, 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。

$$B = \frac{P0_{\phi}}{2 \times (n + 3) \times 2^N}$$

B : ビットレート[bps], P0 $\phi$  : 32[MHz], 0 $\leq$ n $\leq$ 31, 0 $\leq$ N $\leq$ 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「RZ/A1H グループ、RZ/A1M 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)	0.001～	1[ms]から設定可能です。  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 1[ms]前後します。※1
	void attach_ms(void (*fptr)(void), unsigned int t)	1～	
	void attach_us(void (*fptr)(void), unsigned int t)	設定禁止 AP-RZA-1A では attach_us 関数は使用できません。	
Ticker	void attach(void (*fptr)(void), float t)	0.001～	1[ms]から設定可能です。  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 1[ms]前後します。※1
	void attach_ms(void (*fptr)(void), unsigned int t)	1～	
	void attach_us(void (*fptr)(void), unsigned int t)	設定禁止 AP-RZA-1A では attach_us 関数は使用できません。	
InterruptIn	void mode(PinMode pull)	PullNone 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 1[ms]以上待機してから指定関数を実行したい場合は、2[ms]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()	0x0000~0xFFFF	-
PortOut	int read()	0x0000~0xFFFF	-
PortInOut	int read()	0x0000~0xFFFF	-
AnalogIn	unsigned short read_u16()	0x0000~0x0FFF	ADCは12ビット分解能です。
Timer	float read(void)	0.001~	1[ms]から取得可能です。
	int read_ms(void)	1~	
	int read_us(void)	取得禁止 AP-RZA-1Aでは read_us関数は 使用できません。	

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-RZA-1A 上で  $\alpha$ 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。

$\alpha$ 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	P1_10/AN2	デジタル入力 アナログ入力	2	P1_11/AN3	デジタル入力 アナログ入力
3	P1_12/AN4	デジタル入力 アナログ入力	4	P1_13/AN5	デジタル入力 アナログ入力
5	P1_14/AN6	デジタル入力 アナログ入力	6	P1_15/AN7	デジタル入力 アナログ入力
7	P1_8/AN0	デジタル入力 アナログ入力	8	P1_9/AN1	デジタル入力 アナログ入力
9	AVCC	-	10	AGND	-
11	GND	-	12	GND	-
13	P3_6	デジタル入力/出力/入出力	14	P3_5	デジタル入力/出力/入出力
15	P3_4	デジタル入力/出力/入出力	16	P3_3	デジタル入力/出力/入出力
17	GND	-	18	GND	-
19	P8_2	デジタル入力/出力/入出力	20	P8_3/IRQ1	デジタル入力/出力/入出力 IRQ
21	P8_4	デジタル入力/出力/入出力	22	P8_5	デジタル入力/出力/入出力
23	P8_6	デジタル入力/出力/入出力	24	P8_8	デジタル入力/出力/入出力
25	P8_9	デジタル入力/出力/入出力	26	P8_10	デジタル入力/出力/入出力
27	P8_11	デジタル入力/出力/入出力	28	P8_12/TIOC3C	デジタル入力/出力/入出力 PWM 出力
29	P8_13	デジタル入力/出力/入出力	30	P8_14	デジタル入力/出力/入出力
31	P8_15	デジタル入力/出力/入出力	32	P9_0	デジタル入力/出力/入出力
33	P9_1/IRQ0	デジタル入力/出力/入出力 IRQ	34	CPU_nRESET	-
35	CKIO	-	36	NMI	-
37	GND	-	38	GND	-
39	VCC	-	40	+5V	-

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	+5V	-	2	VCC	-
3	GND	-	4	GND	-
5	P6_15	デジタル入力/出力/入出力	6	P6_14	デジタル入力/出力/入出力
7	P6_13	デジタル入力/出力/入出力	8	P6_12	デジタル入力/出力/入出力
9	P6_11/IRQ3	デジタル入力/出力/入出力 IRQ	10	P6_10/IRQ2	デジタル入力/出力/入出力 IRQ
11	P6_9	デジタル入力/出力/入出力	12	P6_8	デジタル入力/出力/入出力
13	P6_5	デジタル入力/出力/入出力	14	P6_4	デジタル入力/出力/入出力
15	P6_3	デジタル入力/出力/入出力	16	P6_2/TI0C2A	デジタル入力/出力/入出力 PWM 出力
17	P6_1/IRQ4	デジタル入力/出力/入出力 IRQ	18	P6_0/IRQ5	デジタル入力/出力/入出力 IRQ
19	GND	-	20	GND	-
21	P5_7	デジタル入力/出力/入出力	22	P5_6	デジタル入力/出力/入出力
23	P5_5/TI0C0C	デジタル入力/出力/入出力 PWM 出力	24	P5_4	デジタル入力/出力/入出力
25	P5_3	デジタル入力/出力/入出力	26	P5_2	デジタル入力/出力/入出力
27	P5_1	デジタル入力/出力/入出力	28	P5_0	デジタル入力/出力/入出力
29	GND	-	30	GND	-
31	P1_7/IRQ7	デジタル入力/出力/入出力 IRQ	32	P1_6/IRQ6	デジタル入力/出力/入出力 IRQ
33	P2_15/MISO0	デジタル入力/出力/入出力 SPI マスタ/スレーブ	34	P2_14/MOSI0	デジタル入力/出力/入出力 SPI マスタ/スレーブ
35	P2_13/SSL00	デジタル入力/出力/入出力 SPI マスタ/スレーブ	36	P2_12/RSPCK0	デジタル入力/出力/入出力 SPI マスタ/スレーブ
37	P2_11/TI0C1A	デジタル入力/出力/入出力 PWM 出力	38	P2_10	デジタル入力/出力/入出力
39	P2_9	デジタル入力/出力/入出力	40	P2_8	デジタル入力/出力/入出力

Table 2.3-2 拡張コネクタ CN2 ピンアサイン

## 2.4 ライブラリ使用メモリ

$\alpha$ RPL を AP-RZA-1A で動作させる際は、以下のメモリ領域を  $\alpha$ 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、 $\alpha$ 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
0x20100000~0x2010001C	ベクタテーブル	内蔵 RAM ブート時のアドレス
0x18000000~0x1800001C	ベクタテーブル	シリアル FlashROM ブート時のアドレス
0x20102000~0x2010201C	ベクタテーブル	内蔵 RAM/シリアル FlashROM からブート後、ベクタテーブルはこのアドレスに変更される。

Table 2.4-1  $\alpha$ RPL AP-RZA-1A ライブラリ使用領域

また、 $\alpha$ 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	領域	内容
システムモード スタック領域	0x4000	ARM_LIB_STACK	システムモードを処理するために必要な領域
IRQ モード スタック領域	0x1000	IRQ_STACK	IRQ 割り込みを処理するために必要な領域
FIQ モード スタック領域	0x1000	FIQ_STACK	FIQ 割り込みを処理するために必要な領域
SVC モード スタック領域	0x1000	SVC_STACK	OS 等のシステムを処理するために必要な領域
アボートモード スタック領域	0x1000	ABT_STACK	メモリアクセス違反を処理するために必要な領域

Table 2.4-2  $\alpha$ RPL AP-RZA-1A 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、 $\alpha$ 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-RZA-1A 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 シリアル FlashROM からブートする場合



Fig 3.1-1 シリアル FlashROM からブートする場合の設定

##### ② SD ホストインターフェースからブートする場合



Fig 3.1-2 SD ホストインターフェースからブートする場合の設定

### 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 LD1 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。

```
DigitalOut led(P1_4, 1)      /* デジタル出力初期化(LD1) */  
  
int main( void )            /* メインルーチン */  
{  
    while (1)  
    {  
        led = ! led;        /* LED の点滅 */  
        wait(0.5);         /* 0.5 秒ウェイト */  
    }  
  
    return 0;  
}
```

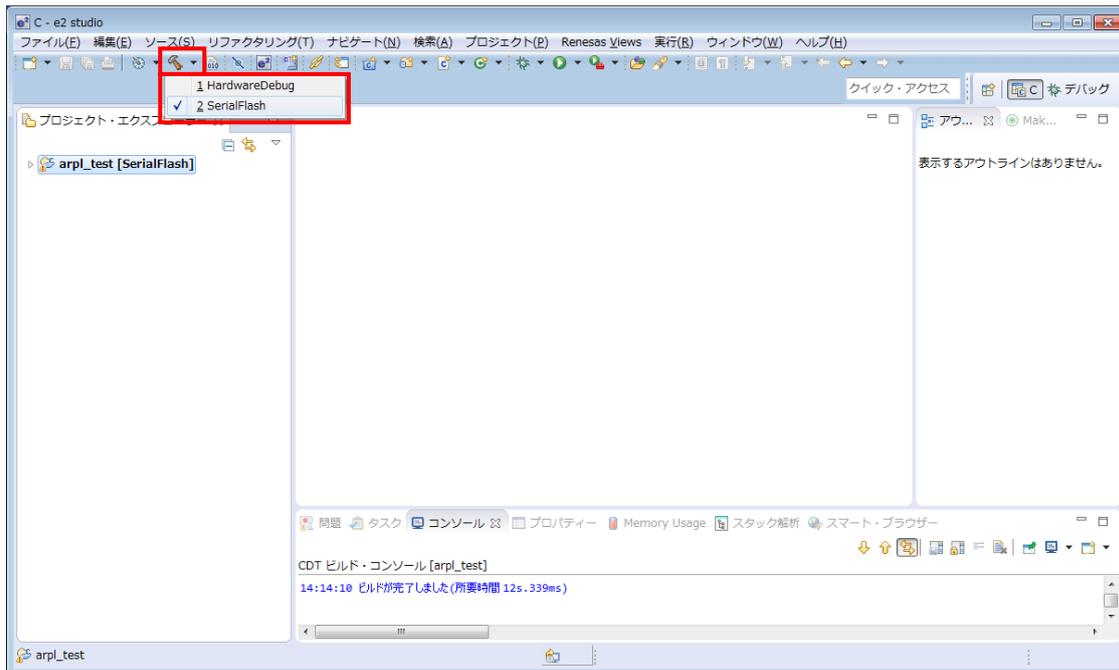
デジタル出力サンプルコード

## 3.1.3 ビルド・ダウンロード

### (1) ビルド

- ①あらかじめ「 $\alpha$ RPL User Manual」の「4. ライブラリ組み込み手順(RZ 編)」を参照し、e<sup>2</sup> studio のワークスペースに  $\alpha$ RPL を組み込んで下さい。組み込みが完了したらビルド構成を作成し、ビルドが成功することを確認して下さい。
- ②e<sup>2</sup> studio を起動し、 $\alpha$ RPL を組み込んだワークスペースを開いて下さい。
- ③ツールバーからビルドアイコンを選択します。

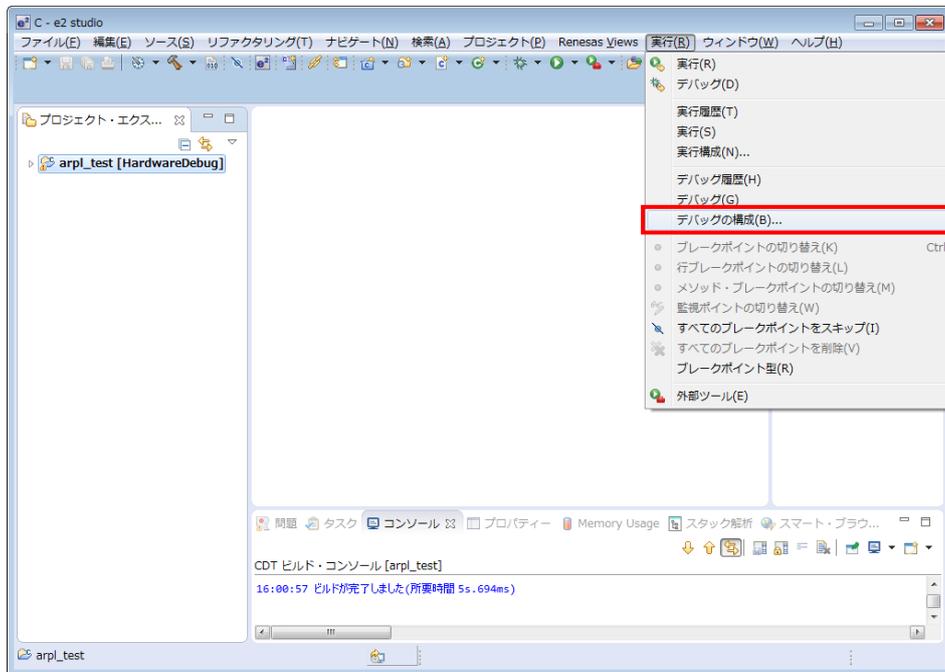
アイコン横の下矢印をクリックすることでコンフィグレーションごとのビルドが可能です。



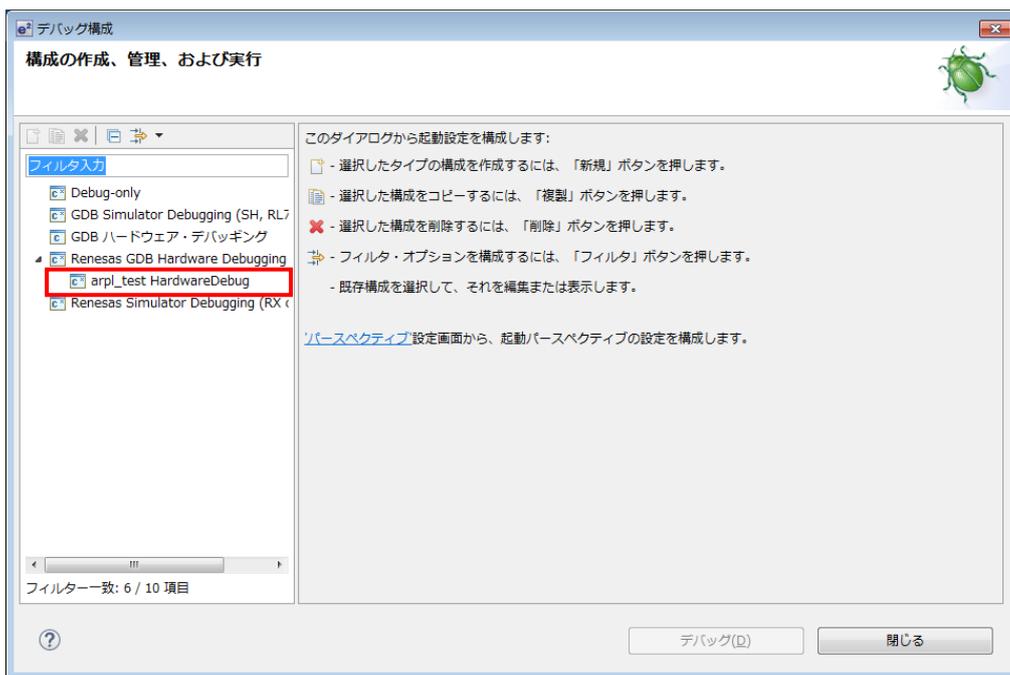
構成が HardwareDebug の場合、¥HardwareDebug ワークフォルダに RAM 動作のオブジェクトファイルが、SerialFlash の場合、¥SerialFlash ワークフォルダ内に ROM 動作のオブジェクトファイルが生成されます。

## (2-1) J-Link を使用したダウンロード

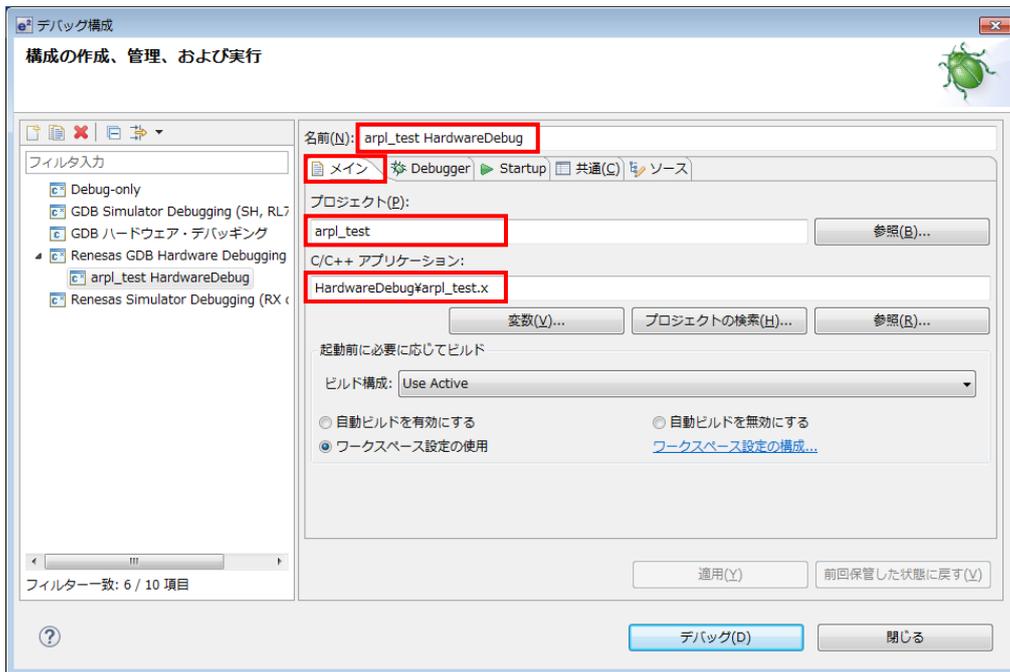
- ① e<sup>2</sup> studio を起動し、 $\alpha$  RPL を組み込んだワークスペースを開いて下さい。
- ② 「3.1.3 ビルド・ダウンロード」の「(1)ビルド」を参考に HardwareDebug 構成でビルドして下さい。
- ③ 「3.1.1 動作モード」の「②SD ホストインタフェースからブートする場合」を参考に AP-RZA-1A の設定を行い、PC と AP-RZA-1A を J-Link を介して接続します。
- ④ 「実行」→「デバッグの構成」を選択して下さい。



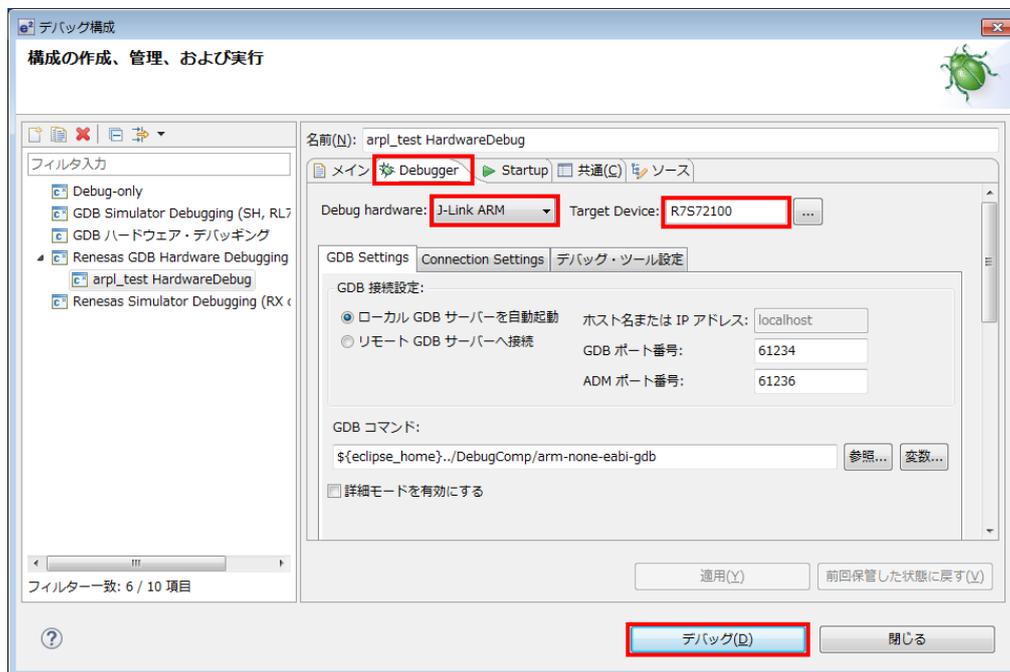
- ⑤ 「Renesas GDB Hardware Debugging」を選択し、新規にデバッグ構成を追加して下さい。  
既に存在している場合、追加する必要はありません。



- ⑥ 追加したデバッグ構成の「メイン」を選択し、名前を「arpl\_test HardwareDebug」、プロジェクトを「arpl\_test」、C/C++アプリケーションを「HardwareDebug¥arpl\_test.x」に設定して下さい。



- ⑦ 「Debugger」を選択し、Debug hardware を「J-Link ARM」に、Target Device を「R7S72100」に設定して下さい。設定が完了したらボードに電源を投入し「デバッグ」ボタンを押して下さい。



- ⑧ プログラムを実行し、1秒周期でLD1が点滅することを確認して下さい。

### (2-2) Flash Writer EX/RZ を使用したダウンロード

- ① e<sup>2</sup> studio を起動し、 $\alpha$  RPL を組み込んだワークスペースを開いて下さい。
- ② 「3.1.3 ビルド・ダウンロード」の「(1)ビルド」を参考に SerialFlash 構成でビルドして下さい。
- ③ 「AN1608 AP-RZA-1A (RZ/A1H CPU BOARD) シリアル FlashROM の書き込み方法」を参考に  
②で生成されたモトローラファイルをボードにダウンロードして下さい。
- ④ 「3.1.1 動作モード」の「①シリアル FlashROM からブートする場合」を参考に AP-RZA-1A の設定を行い、  
ボードに電源を投入します。
- ⑤ プログラムが実行され、1 秒周期で LD1 が点滅すること確認して下さい

改定履歴

版数	日付	改定内容
1 版	2016/01/12	新規作成
2 版	2021/02/10	URL を更新
2.1 版	2023/10/02	Table 1.2-1 使用環境(AP-RZA-1A) 更新 (1 章) 住所の更新

## 参考文献

「RZ/A1H グループ、RZ/A1M グループ ユーザーズマニュアル ハードウェア編」ルネサス エレクトロニクス株式会社  
その他 各社データシート

## 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。万が一不審な点、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、万が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

- ・RZ および RZ/A1H、A1M 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。
  - ・e<sup>2</sup> studio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。
  - ・J-Linkは、SEGGER Microcontroller GmbH & Co. KG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](mailto:query@apnet.co.jp)