e2studio 環境で動作する RL78_G24_101GA ボード用サンプルプログラムの使い方 2025.03.04/03.13

あらかじめパソコン (PC) と RL78_G24_101GA ボードをケーブルで接続しておいてください。 ケーブルは USB A+C タイプです。

e2studioを立ち上げ、ダウンロード、解凍したホルダを適当な所に配置します。

1番はじめはプログラムをインポートします。



一般→既存プロジェクトをワークスペースへ



ルートディレクトリの選択で、ダウンロードしたファイルのディレクトリを指定。プロジェクト、プロジェクトをワークスペースにコピーをチェック。終了をクリック。

● ル <mark>-ト・ディレクトリーの選択</mark> (T):	C:¥Users¥	hirosi¥OneDriv	ve¥ドキュメント	¥F (参照(R)	ł)
○ アーカイブ・ファイルの選択(A):	1				参照(R)	ð
プロ <u>ジ</u> ェクト(P):						
RL78_G24_101GA_test1	C:¥Users¥hi	rosi¥OneDrive	ŧ¥ドキュメント¥I	ha 📑	べて選択((S)
				選択を	すべて解	¥除(D)
					更新(E)	
٤			5	>		
< オジョン コネストしたプロジェクトを検索(ト フリジェクトをワークスペースにご 完了次葉、新しくインボートし; ワークスペースに既に存在する; ワーキング・セット	4) ビー(C) たプロジェクトを プロジェクトを≬	を閉じる(o) 寝す(i)	2	>		
< オブション オブション ネストしたプロジェクトを検索() アリジェクトをワークスペースにご 一売了次第、新しくインボートし ワークスペースに既に存在する ワーキング・セット ロワーキング・セットにプロジェクリ	4) ビー(C) ビブロジェクトを ・を追加(T)	を閉じる(o) 厚す(i)		▶ 	新規(W)	

RL78_G24_101GA_test1 のプロジェクトが表示されます。ソースファイルは拡張子. cpp です。 クリックで右画面に表示されます。

workspace_RL78 - RL78_G24_101GA_test1/src/RL78_G24_101GA_test1.cpp - e² studio ファイル(F) 編集(E) ソース(S) リファクタリング(T) Source ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) Renesas AI ウィンドウ(W) ヘルプ(H) 📳 🛞 • 🔦 • 💷 🔌 🏘 • 💁 • 🌽 🕅 ■ C/C++ プロジェクト × (> ⇒) Q | □ ≤ 8 □ □ Q RL78_G24_101GA_test1.cpp × ✓ → RL78_G24_101GA_test1 ⊕ * DISCLAIMER. 2 19 > 場 バイナリー 21 26 27 ⊕ * File Name : RL78_G24_101GA_test1.cpp. > 🗊 Includes v 🕮 src 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 #cnult
#include "r_smc_entry.h"
© #ifdef CPPAPP > C RL78_G24_101GA_test1.cpp > 👝 HardwareDebug } #endif > 🗁 trash RL78_G24_101GA_test1.scfg int cnt; RL78_G24_101GA_test1 HardwareDebug.launch void lwait(volatile long wtime) { 0 while(wtime != 0) { wtime--; } } int main(void); デバックをクリック。

通常、1回目は接続できないと表示されます。これは製作者の COM ポートと利用者の COM ポ ートが異なるからです。

■ デバ	ッグ接続に失敗しました	×
	ターゲットに接続できません。 GDBサーバコンソールの表示で詳細をご確認ください。	
	考えられる原因: 1: プロジェクトで指定されているデバイスが、ターゲットハードウェアの指定と 致しない 2: デバイスとエミュレータ、および、ターゲットボードとエミュレータ間の接続の 題 3. デバッグオプションがハードウェア要件と一致しない 4: ターゲットボードに電源が供給されていない 5: 間違った電源がターゲットボードに供給されている 6: 必要な dll や so ファイルが見つからない 7: サボート対象外のデバイスに接続している	:一 問
	注:エミュレータの 'RUN' LED が ON もしくは、一度接続に失敗した場合 エミュレータの切断、再接続が必要な場合があります。	à,
	ОК	

実行→デバックの構成

実行	(R) Renesas AI ウィンドウ	v(W) ヘルプ(H)
	Renesas Debug Tools	>
	再開(M)	
	中断(S)	
	終了(T)	
14	切断	
Э.	ステップイン(I)	
Q	ステップ・オー/(-(O)	
.Ē	ステップ・リターン(U)	
=	Set Next Statement	Ctrl+Alt+R
⇒]	指定行まで実行(L)	
ক্ট	ステップ・フィルタの使用(F)	シフト+F5
Q,	実行(R)	Ctrl+F11
徳	デバッグ(D)	F11
	実行履歴(T)	>
0	実行(S)	>
	実行構成(N)	
	デバッグ履歴(H)	>
林	デバッグ(G)	>
	デバッグの構成(B)	

Debugger→ターゲット・ボードとの接続 で COM 番号を利用者のものに合わせて下さい。 OK であれば「適用」をクリック。

デバック構成			- 🗆
成の作成 管理 おけ(実行			15
			2
起動時にノフツシュを消去 オノショノが有効に	- なうています。正常に接続しためと、このオノジョンを無効にしてくたさい。	3	25
	名前(N): RL78_G24_101GA_test1 HardwareDebug(2)		
イルタ入力	📄 メイン 🕸 Debugger 🕨 Startup 🧤 ソース 🔲 共通		
++ アプリケーション			
++ リモート・アプリケーション	Debug hardware: COM Port (RL78) V Target Device:	R7F101GAE	
Script			
Simulator Debugging (RH850)	GDB Settings Connection Settings デバッグ・ツール設定	マルチコア設定 FAA Core Settings	
ハードウェア・デバッギング	✓ クロック		
RL78_G24_101GA_test1 HardwareDeb	メイン・クロック周波数 [MHz]	内部クロックの使用	~
Python Run	サブ・クロック周波数 [kHz]	内部クロックの使用	~
Python unittest	モニター・クロック	システム	~
on run	✓ ターゲット・ボードとの接続		
on unittest	COM#-F	COM37	
ev Django	リセット制御端子	OTA	~
ev Google App Run	ボーレート [bps]	Auto	~
on Run	> フラッシュ		
on unittest	現在のセキュリティー ID (HEX)	000000000000000000000000000000000000000	
esas GDB Hardware Debugging	現在のシリアル・プログラミング・セキュリティID (HEX)	FFFFFFFFFFFFFFFFFFFFFFFFFF	
L78_G24_101GA_test1 HardwareDebu	シリアル・プログラミング・セキュリティIDの書き換えを許可	いいえ	~
RL78_G24_101GA_test1 HardwareDeb	フラッシュ書き込みを許可する	はい	~
esas Simulator Debugging (RX, RL78)	広域電圧モードの使用	はい	~
クループ	起動時にフラッシュ ROM を消去	はい	~
	フラッシュのセルフ・プログラミングを行う	いいえ	~
	フラッシュメモリー書き換え時に未使用領域をOxFFで埋める	いいえ	~

再度、デバックをクリック。以下、パースペクティブ切り替えの確認が出たら接続は成功 しています。「切り替え」をクリック。



今まで見えなかったボタンが出てきます。実行をクリック。

workspace_RL78 - RL78_G24_101GA_test1/src/RL78_G24_101GA_test1.cpp - e² studio
 ファイル(F) 編集(E) ソース(S) リファクタリング(T) Source ナビゲート(N) 検索(A) プロジ



1回目はカーソルがmain()の後で止まります。メインの前の初期化等は正常に実行されました。もう1回、実行をクリックするとwhile(1)の無限ループ動作に入り、 LED D1が点滅すれば動作成功です。

	00		
	51	⊖int	main(void)
	52	{	
l	53 00000331		EI();
	54 00000334		cnt = 0;
	55		
	56 00000355	Θ	while(1)
	57		{
	58 00000338		cnt++;
	59 0000033b		P3_bit.no0 = 1;
	60 0000033f		lwait(1000000);
	61 00000348		P3_bit.no0 = 0;
	62 0000034c		lwait(1000000);
	63		<pre>// TODO: add application code here</pre>
	64		}
	65		return 0;
	66	}	
	67		

LED D1の点滅が目で確認できると思います。変数窓の変数、cntがLEDが点滅するたびに+1されるのが確認できると思います。

			(٦	1	6	C/C++ 🗳	スマート
(x)= 変数	• ブ	/	20	ý	िं में	X	e ^e イベン	品 F
						K.	** 🖂 🕯	₽ %
式		型		値	I		アドレス	
es c	nt	int		76	5		0xfd700	
🕂	FLUIT						1 1	

有限会社ビーリバーエレクトロニクス info@beriver.co.jp