MITSUBISHI ELECTRIC
Changes for the Better

三菱電機汎用シーケンサ C言語コントローラユニット クイックスタートガイド

e-F@ctory

はじめよう、C言語コントローラ!

MELSEC iQ-R series

C Controller

目次

CONTENTS

関連マニュアル

本クイックスタートガイドでは，C言語コントローラユニットの基本的な導入手順を紹介しています。
C言語コントローラユニットを十分に活用するために，目的に応じて，下記のマニュアルをお読みください。

マニュアル名称[マニュアル番号]	内容	提供形態	価格
MELSEC iQ-R C言語コントローラユニットユーザーズマニュアル(スタートアップ編) [SH-081366]	C言語コントローラユニットの性能仕様，運転までの手順，トラブルシューティングについて記載しています。	製本物	1,500円
		e-Manual PDF	—
MELSEC iQ-R C言語コントローラユニットユーザーズマニュアル(応用編) [SH-081368]	C言語コントローラユニットの機能，デバイス，パラメータなどについて記載しています。	製本物	3,000円
		e-Manual PDF	—
MELSEC iQ-R C言語コントローラユニットプログラミングマニュアル [SH-081370]	C言語コントローラユニットのプログラミング仕様および専用関数ライブラリについて記載しています。	e-Manual PDF	—
CW Workbench/CW-Sim オペレーティングマニュアル [SH-081372]	CW Workbench/CW-Simのシステム構成，仕様，機能，トラブルシューティングについて記載しています。	e-Manual PDF	—
CW Configurator オペレーティングマニュアル [SH-081381]	CW Configuratorのシステム構成や，パラメータ設定，オンライン機能の操作方法などについて記載しています。	e-Manual PDF	—

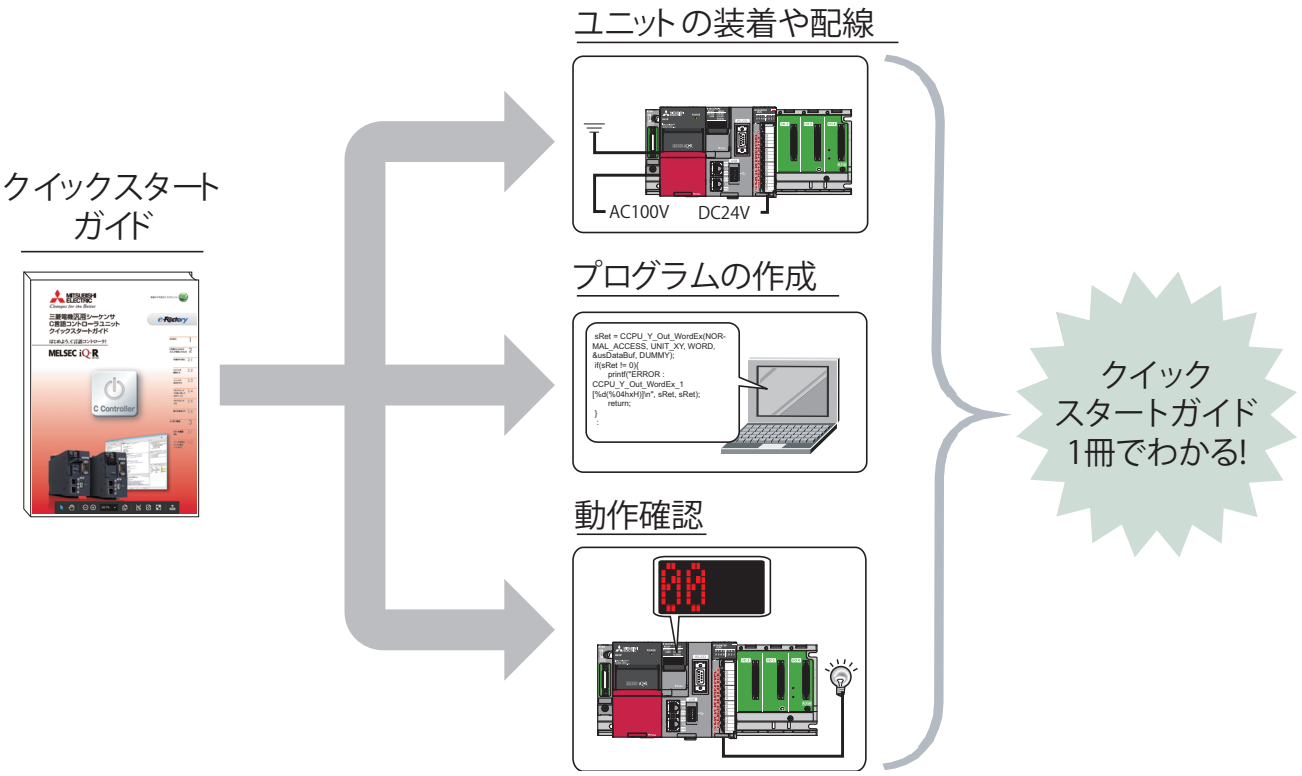
Point

最新のe-ManualおよびマニュアルPDFは，三菱電機FAサイトからダウンロードできます。
www.MitsubishiElectric.co.jp/fa

1 はじめに

本クイックスタートガイドは，三菱電機シーケンサMELSEC iQ-RシリーズC言語コントローラユニットR12CCPU-V(以下C言語コントローラユニット)を初めて使用する場合の基本的な導入手順を，わかりやすく説明しています。
MELSEC iQ-Rシリーズを初めて使用する方で，下記のような方が，C言語コントローラユニットの使い方を簡単に理解することができるようになっています。

- C言語またはC++言語プログラムのプログラミング経験がある
- マイコンボードやパソコン環境からC言語コントローラユニットを使ったシステムへの置換えを考えている



Point

- C言語コントローラユニットを使用するための安全上のご注意は，下記を参照してください。
  (📖MELSEC iQ-R C言語コントローラユニットユーザーズマニュアル(スタートアップ編))
- 本クイックスタートガイドでは，Windows 7での設定例で説明しています。

注意事項

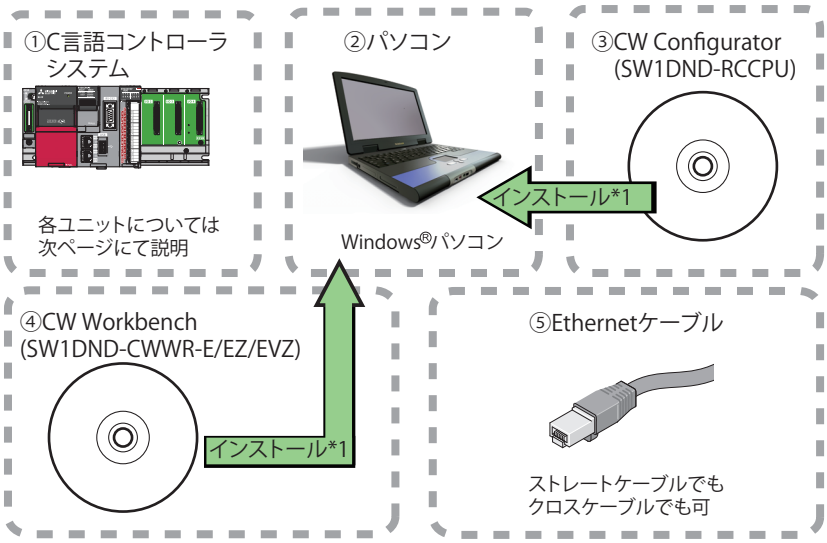
本クイックスタートガイドは，「6ページ システム構成例」に示すシステム構成での操作を前提としています。
実際にシステムを設計/運用する場合には，下記で紹介しているマニュアルを必ずお読みください。
☞ 関連マニュアル

MEMO

2 C言語コントローラユニットを使ってみよう

2.1 作業を行う前に

必要な機材を準備します。



*1 あらかじめ，同じパソコンにCW Configurator，CW Workbenchをインストールしておいてください。

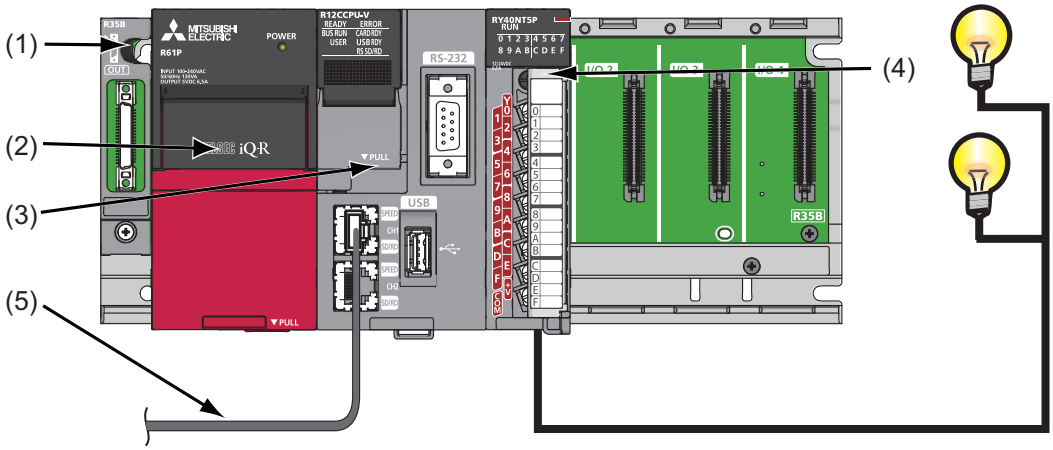
Point

CW Configuratorのインストールについては，下記を参照してください。
CW Configurator インストール手順書
CW Workbenchのインストールについては，下記を参照してください。
CW Workbench/CW-Sim オペレーティングマニュアル

2.2 システムを構築する

システム構成例

本クイックスタートガイドでは，下記のシステム構成を例に挙げて説明します。



No.	名称	形名	説明
(1)	ベースユニット	R35B	電源ユニット，C言語コントローラユニット，入出力ユニットなどを装着するユニットです。
(2)	電源ユニット	R62P	C言語コントローラユニット，入出力ユニットなど各ユニットに電気を供給するユニットです。
(3)	C言語コントローラユニット	R12CCPU-V	C言語コントローラシステムの制御を統括するユニットです。
(4)	出力ユニット	RY40NT5P	—
(5)	接続ケーブル (Ethernetケーブル)	10BASE-T/100BASE-TX/ 1000BASE-Tの規格を満たす Ethernetケーブル	CW Configurator，CW Workbenchをインストールしたパソコンと C言語コントローラユニットを接続します。

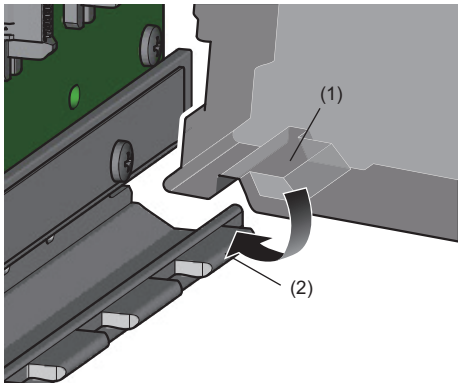
ユニットを装着する

準備したユニットをベースユニットに取り付けます。

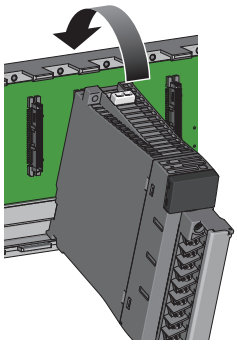
注意事項

ユニットを取り付けるときは，必ず電源を遮断してください。

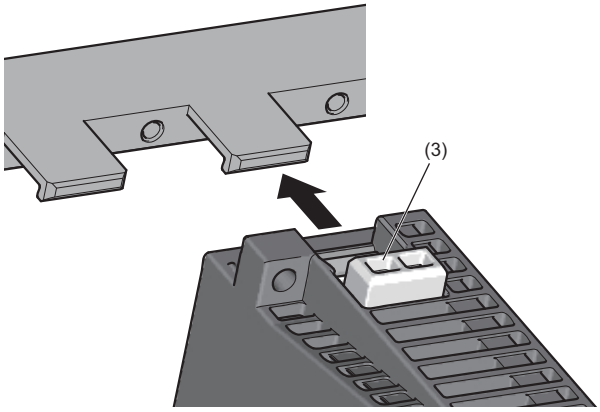
取付け手順



1. ベースユニットのユニットコネクタにキャップが装着されている場合は，キャップを取りはずしてください。
2. ユニット凹部(1)とベースユニットのガイド(2)の先端を合わせます。



3. ユニットは，ガイド(2)を支点とし，ユニット固定用フック(3)が「カチッ」と音がするまで矢印方向に押して，ベースユニットに装着します。



4. ユニットのユニット固定用フック(3)がベースユニットに掛かり，ユニットが確実に装着されていることを確認してください。

ユニットの配線を行う

電源ユニットの配線を行います。

注意事項

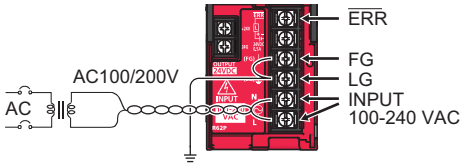
ユニットの配線を行うときは，必ず電源を遮断してください。

Point

配線についての注意事項の詳細については，下記を参照してください。
MELSEC iQ-Rユニット構成マニュアル

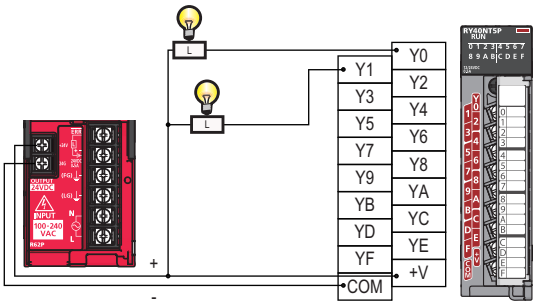
1. 電源ユニットへの配線をする。

電源ユニットへの電源線，接地線の配線例を下記に示します。
接地は，感電，誤動作を防止するために行う配線です。



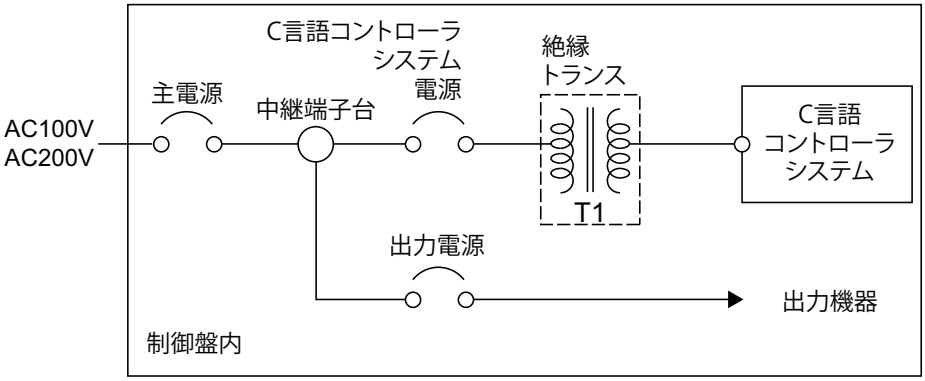
2. 出力ユニットへの配線をする。

出力ユニットRY40NT5Pへの配線例を下記に示します。



Point

出力電源とC言語コントローラシステム電源は，下図のように，系統を分離して配線を行ってください。



電源が正常か確認する

システムの設置，ユニットの取付け，配線をした後に，電源が正常に入ることを確認します。

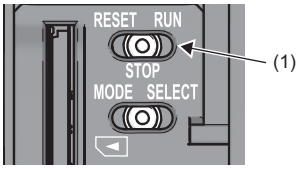
操作手順

1. 電源を入れる前の確認を行います。

- 電源の配線
- 電源電圧

2. C言語コントローラユニットの状態をSTOPにします。

C言語コントローラユニット前面のカバーを開き，RESET/STOP/RUNスイッチ(1)を「STOP」の位置にします。



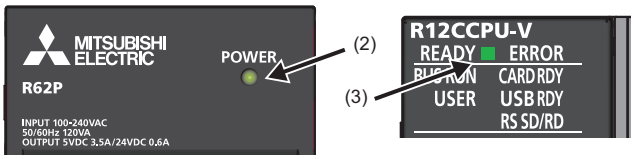
3. 電源をONします。

4. 電源が正常であるかを確認します。

各ユニットの前面LEDを確認します。

正常な状態のときのLED表示は下記のとおりです。

- 電源ユニット: POWER LED(2)が緑色点灯
- C言語コントローラユニット: READY LED(3)が緑色点灯



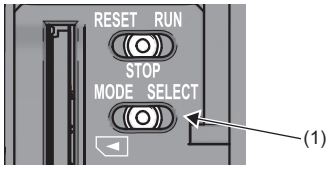
Point

電源を投入しても電源ユニットのPOWER LEDが消灯している場合は，電源の配線，装着が正しく行われているか確認してください。

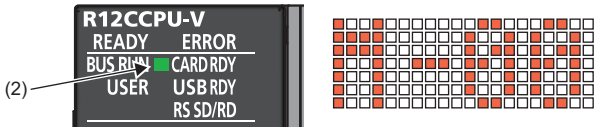
2.3 ユニットの設定をする

C言語コントローラユニットを初期化する

スイッチ操作を行う前に，RESET/STOP/RUNスイッチの位置が中央にあることを確認してください。

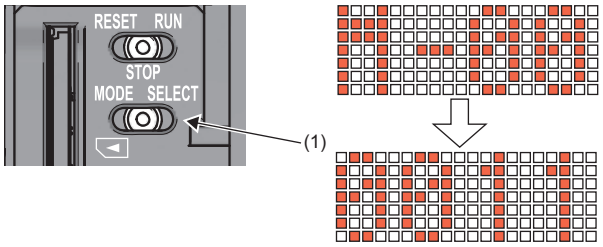


1. MODE/SELECTスイッチ(1)を，MODE側に保持します。

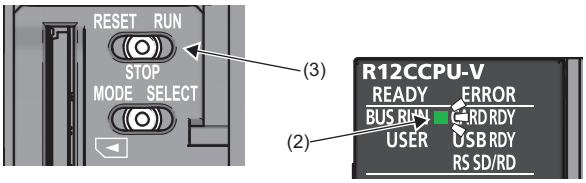


2. C言語コントローラユニットを電源ONします。BUS RUN LED(2)が点灯し，ドットマトリクスLEDに"M-00"が表示されます。

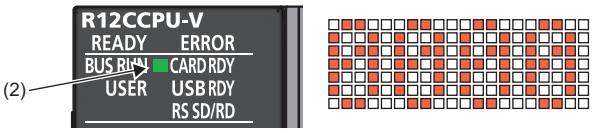
3. MODE/SELECTスイッチ(1)の位置を，中央に戻します。



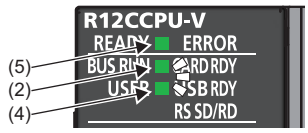
4. MODE/SELECTスイッチ(1)を，SELECT側に倒します。SELECT側に1回倒すごとに，モードが移行し，ドットマトリクスLEDに各モードが表示されます。LEDの表示が"0011"になるまで繰り返してください。



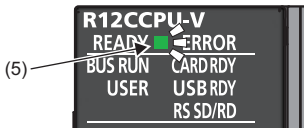
5. RESET/STOP/RUNスイッチ(3)を，RUN側へ倒します。選択したモードが実行されます。初期化実行中は，BUS RUN LED(2)が点滅します。



6. BUS RUN LED(2)の点灯およびドットマトリクスLEDに"0000"が表示されたことを確認したら，C言語コントローラユニットをリセットします。リセット手順は，下記を参照してください。
(☞ 11ページ リセットの操作手順)



7. ユニットのリセットにより，初期化が行われます。初期化中はREADY LED(5)が点灯，BUS RUN LED(2)およびUSER LED(4)が点滅します。

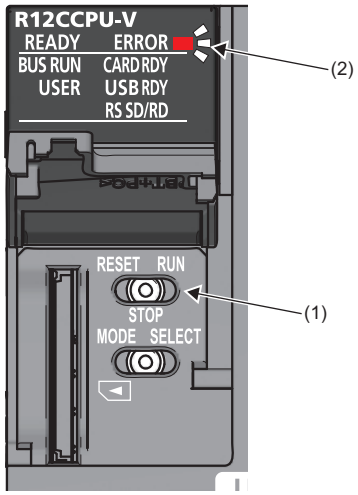


8. 初期化が正常に完了した場合は，BUS RUN LED(2)およびUSER LED(4)が消灯し，READY LED(5)が点滅します。

9. C言語コントローラユニットをリセットします。リセット手順は，下記を参照してください。
(☞ 11ページ リセットの操作手順)

リセットの操作手順

スイッチを下記の手順で操作すると，C言語コントローラユニットをリセットします。



1. RESET/STOP/RUNスイッチ(1)を，RESET側へ保持します。
2. ERROR LED(2)が数回点滅後，すべてのLEDが消灯するのを確認します。
3. RESET/STOP/RUNスイッチ(1)を，STOPの位置に戻します。

Point

C言語コントローラユニットの初期化中は，リセットを行わないでください。
誤ってリセットを行った場合は，再度初期化を行ってください。

パラメータを設定する

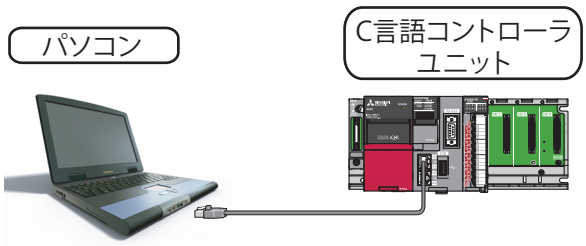
C言語コントローラユニットに，パラメータを設定します。

Point

パラメータ: C言語コントローラシステムを動作させるために必要な設定情報のことです。CW Configuratorを使用して，C言語コントローラシステム内の各ユニットおよびネットワークの設定を行います。

C言語コントローラユニットとパソコンを接続する

C言語コントローラユニットのCH1とパソコンを，Ethernetケーブルで接続します。



注意事項

接続の際には，C言語コントローラユニットとパソコンのIPアドレスを，同一セグメントに設定する必要があります。本クイックスタートガイドでは，C言語コントローラユニットのIPアドレスは初期値(192.168.3.3)を使用するため，パソコン側のIPアドレスを"192.168.3.*"(*: 0，3，255以外)に設定してください。
また，パソコンのサブネットマスクを"255.255.255.0"に設定してください。

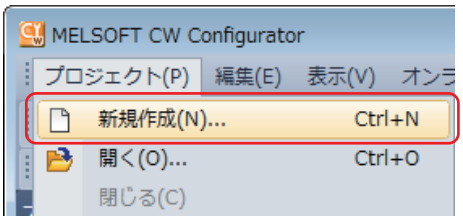
Point

IPアドレスの変更については，下記を参照してください。
MELSEC iQ-R C言語コントローラユニットユーザーズマニュアル(応用編)

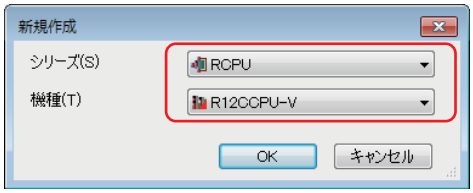
CW Configuratorを起動し，プロジェクトを作成する

操作手順

1. [スタート]⇨[すべてのプログラム]⇨[MELSOFT]⇨[CW Configurator]⇨[CW Configurator]を選択します。
2. [プロジェクト]⇨[新規作成]を選択します。



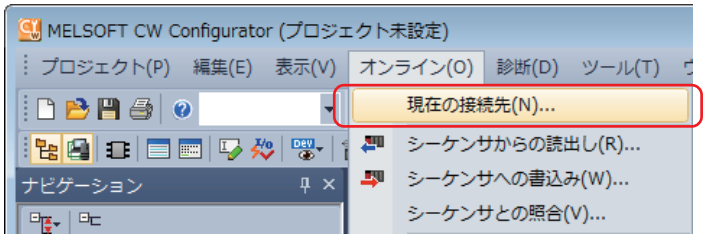
3. "RCPU"，"R12CCPU-V"になっていることを確認して[OK]ボタンをクリックします。



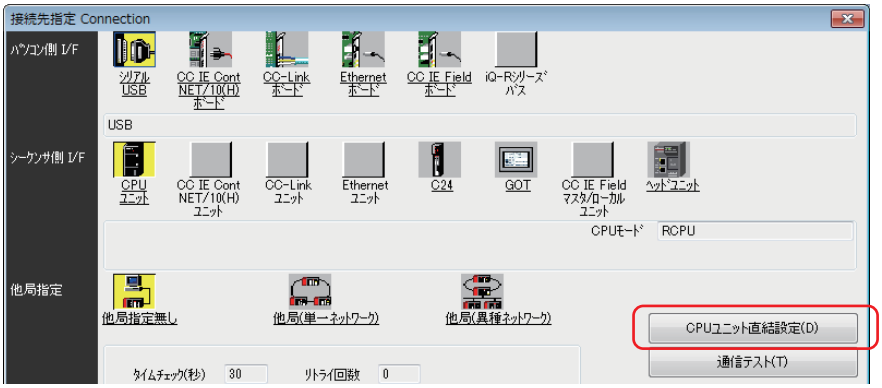
C言語コントローラと通信する

操作手順

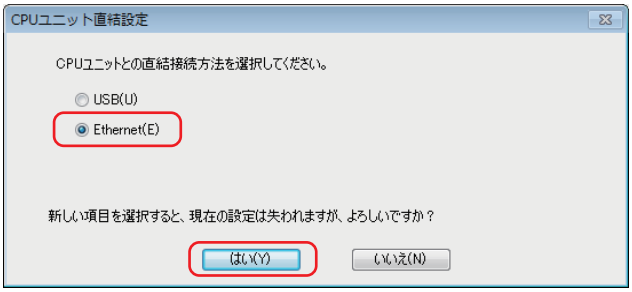
1. CW Configuratorのメニューから，[オンライン]⇨[現在の接続先]を選択します。



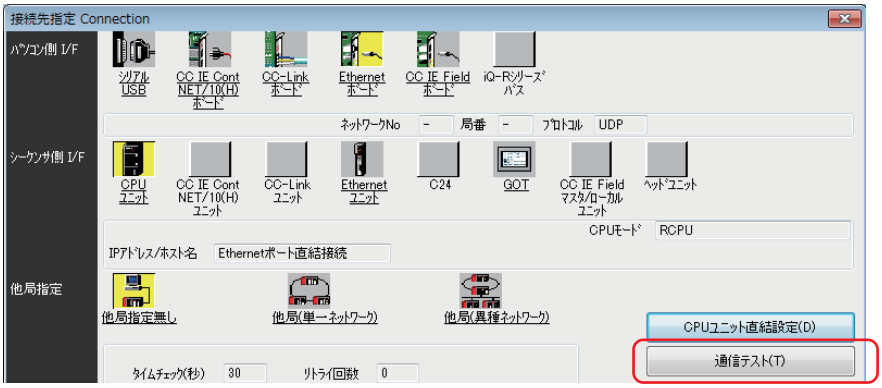
2. "接続先指定"画面で，[CPUユニット直結設定]ボタンをクリックします。



3. "Ethernet"を選択し，[はい]ボタンをクリックします。



4. [通信テスト]ボタンをクリックし，メッセージ"R12CCPU-Vとの接続に成功しました。"が表示されることを確認します。

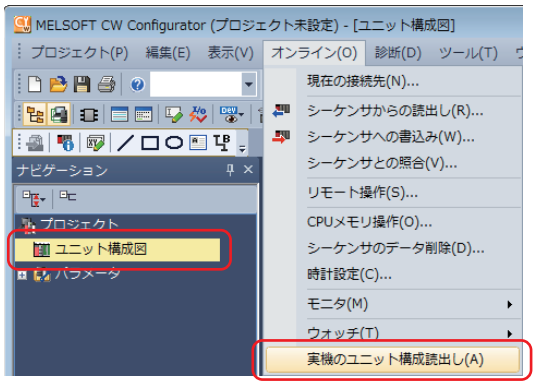


パラメータを設定する

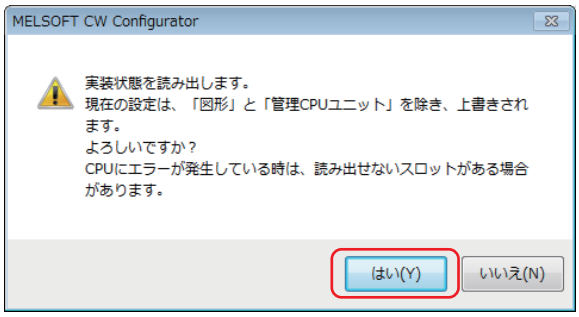
システムのパラメータと各ユニットのパラメータを設定します。

操作手順

1. ナビゲーションウィンドウ上の"ユニット構成図"をダブルクリックして開き，[オンライン]⇨[実機のユニット構成読出し]を選択します。

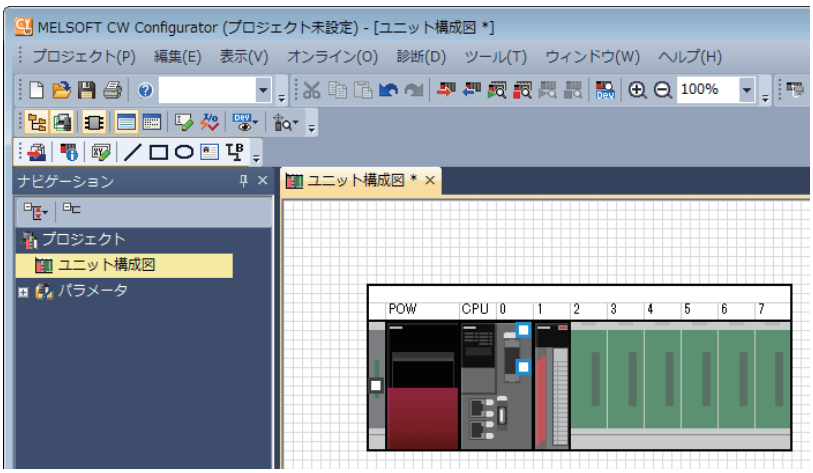


2. 以下のメッセージが表示されたら，[はい]ボタンをクリックします。



3. 読み出しが完了したら，下記のメッセージが表示されるので[OK]ボタンをクリックします。

4. システムパラメータが自動で設定され，実機のシステム構成が"ユニット構成図"ウィンドウに表示されます。

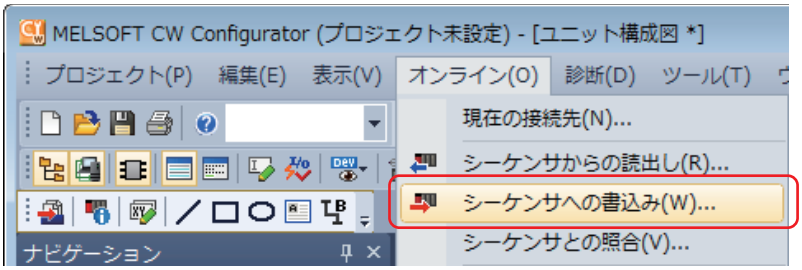


C言語コントローラユニットへパラメータを書込む

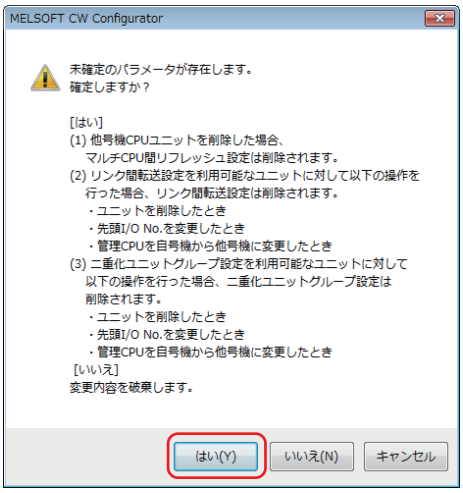
CW Configuratorを使って，C言語コントローラユニットにパラメータを書き込みます。

操作手順

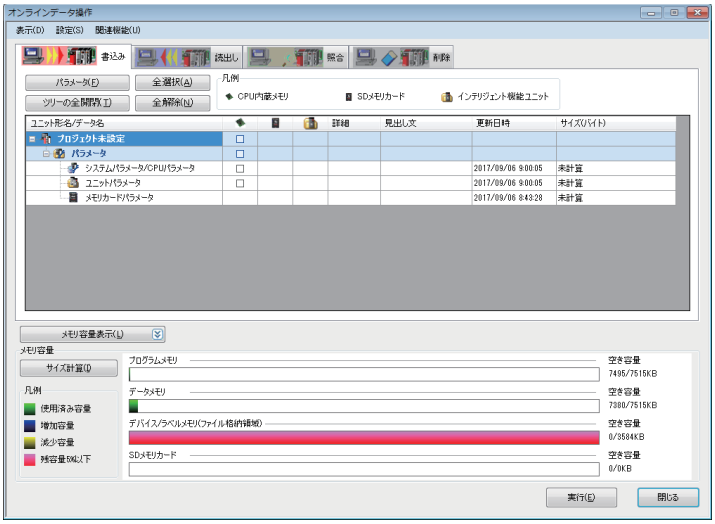
1. [オンライン]⇨[シーケンサへの書込み]を選択します。



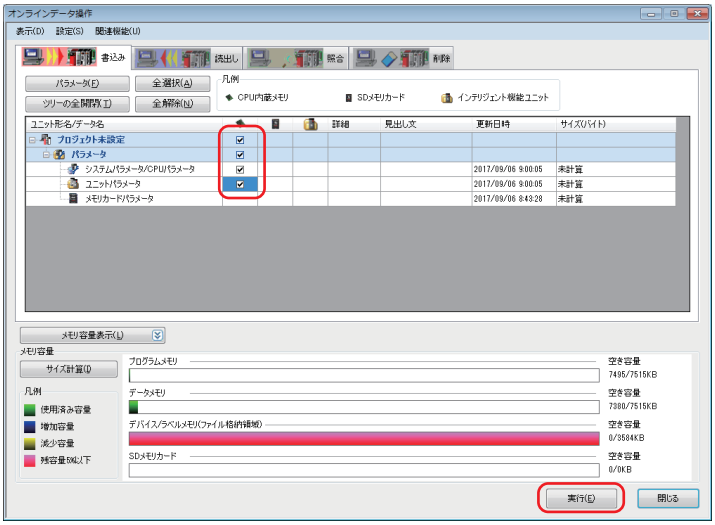
2. [はい]ボタンをクリックします。



3. "オンラインデータ操作"画面が表示されることを確認します。



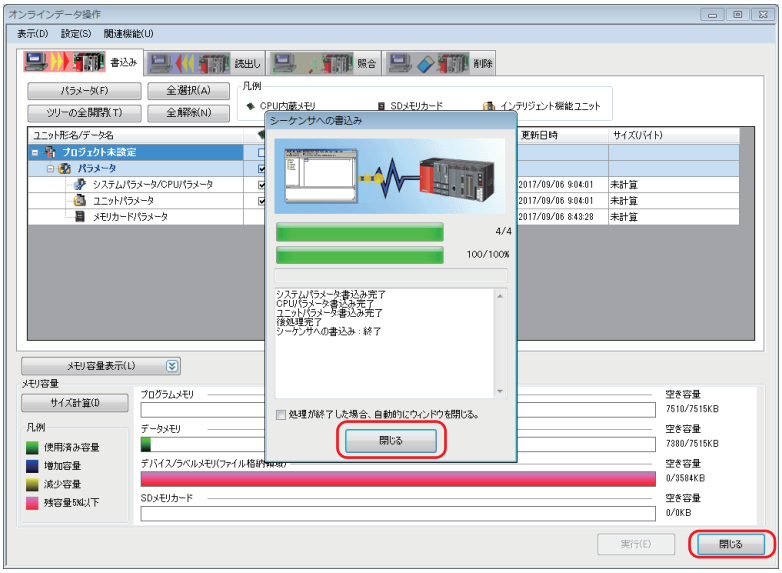
4. "システムパラメータ/CPUパラメータ"，"ユニットパラメータ"を選択し，[実行]ボタンをクリックします。



5. [全てはい]ボタンをクリックします。

書込みが開始されます。

6. C言語コントローラユニットへの書込みが完了したら[閉じる]ボタンをクリックします。



2.4 プログラミングする前に知っておきたいこと

C言語コントローラユニット専用関数とは

C言語コントローラユニット専用関数とは，C言語コントローラの「専用関数ライブラリ」の1つです。ユーザプログラムに使用すれば，MELSEC iQ-Rシリーズの各ユニットを簡単に制御できます。

入出力アクセス

入出力アクセスには，1点単位でのアクセスとワード単位でのアクセスがあります。

■ビット単位アクセス: 1点分の情報(スイッチのON/OFF，ランプの点灯/消灯など)を扱う関数

ビット単位アクセス関数の具体例

関数名	内容
CCPU_X_In_BitEx	入力信号(X)をビット単位(1点)で読み出します。
CCPU_Y_Out_BitEx	出力信号(Y)をビット単位(1点)で出力します。
CCPU_Y_In_BitEx	出力信号(Y)をビット単位(1点)で読み出します。

■ワード単位アクセス: 16点(1ワード)分の情報(数値，文字列など)を扱う関数

ワード単位アクセス関数の具体例

関数名	内容
CCPU_X_In_WordEx	入力信号(X)をワード単位(16点)で読み出します。
CCPU_Y_Out_WordEx	出力信号(Y)をワード単位(16点)で出力します。
CCPU_Y_In_WordEx	出力信号(Y)をワード単位(16点)で読み出します。

ユーザLED制御

ユーザLED制御には，C言語コントローラユニット前面のUSER LED制御とドットマトリクスLED制御があります。

ユーザLED制御関数の具体例

関数名	内容
CCPU_SetLEDStatus	C言語コントローラユニットのLED状態を設定します。
CCPU_SetDotMatrixLED	C言語コントローラユニットのドットマトリクスLEDに表示する値を設定します。

Point

ここでは，最も基本的なC言語コントローラユニット専用関数を紹介しています。
上記のほかにも，各ユニットの制御を行うための便利なC言語コントローラユニット専用関数やMELSEC通信関数があります。各関数については，下記を参照してください。
MELSEC iQ-R C言語コントローラユニットプログラミングマニュアル

今回使用するC言語コントローラユニット専用関数

今回作成するプログラムには，基本的なC言語コントローラユニット専用関数(出力アクセスとドットマトリクスLED制御)を使います。

出力アクセス: CCPU_Y_Out_WordEx関数

■形式

short CCPU_Y_Out_WordEx (short sFlg, unsigned, short usYNo, unsigned short usSize, unsigned short* pusDataBuf, unsigned short usBufSize);

■引数

引数	名称	内容	IN/OUT
sFlg	アクセスフラグ	アクセスフラグを指定します。 ・0: 通常アクセス ・その他: リザーブ	IN
usYNo	先頭出力信号	先頭出力信号(Y)を指定します。 (16の倍数を指定してください。)	IN
usSize	出力サイズ	出力サイズをワード単位で指定します。	IN
pusDataBuf	データ格納先	出力データの格納先を指定します。	IN
usBufSize	データ格納先サイズ	0を指定します。	IN

ドットマトリクスLED制御: CCPU_SetDotMatrixLED関数

■形式

short CCPU_SetDotMatrixLED(unsigned short usLedMode, char* pcData);

■引数

引数	名称	内容	IN/OUT
usLedMode	出力モード	ドットマトリクスLEDへの出力モードを指定します。 (リザーブを指定した場合，関数は無処理で正常終了します。) • 0: ドットモード • 1: ASCIIモード • その他: リザーブ	IN
pcData	LEDデータ	LEDデータを指定します。	IN

• モード1: ASCIIモード時

pcData[0]～pcData[3]に指定された文字列を表示します。

指定可能な文字(ASCIIコード)を示します。

×: 文字指定不可

ビット		上位ビット															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
下位ビット	0	×	×	SP	0	×	P	×	×	×	×	×	×	×	×	×	×
	1	×	×	×	1	A	Q	×	×	×	×	×	×	×	×	×	×
	2	×	×	×	2	B	R	×	×	×	×	×	×	×	×	×	×
	3	×	×	×	3	C	S	×	×	×	×	×	×	×	×	×	×
	4	×	×	×	4	D	T	×	×	×	×	×	×	×	×	×	×
	5	×	×	%	5	E	U	×	×	×	×	×	×	×	×	×	×
	6	×	×	×	6	F	V	×	×	×	×	×	×	×	×	×	×
	7	×	×	×	7	G	W	×	×	×	×	×	×	×	×	×	×
	8	×	×	×	8	H	X	×	×	×	×	×	×	×	×	×	×
	9	×	×	×	9	I	Y	×	×	×	×	×	×	×	×	×	×
	A	×	×	×	×	J	Z	×	×	×	×	×	×	×	×	×	×
	B	×	×	×	×	K	×	×	×	×	×	×	×	×	×	×	×
	C	×	×	×	×	L	×	×	×	×	×	×	×	×	×	×	×
	D	×	×	-	×	M	×	×	×	×	×	×	×	×	×	×	×
	E	×	×	.	×	N	×	×	×	×	×	×	×	×	×	×	×
	F	×	×	/	×	O	×	×	×	×	×	×	×	×	×	×	×

上記以外の文字を指定した場合は，エラーが返ります。

文字列の途中にNULLが入っている場合は，以降のデータは表示せず空白となります。(左詰めで表示します。)

Point

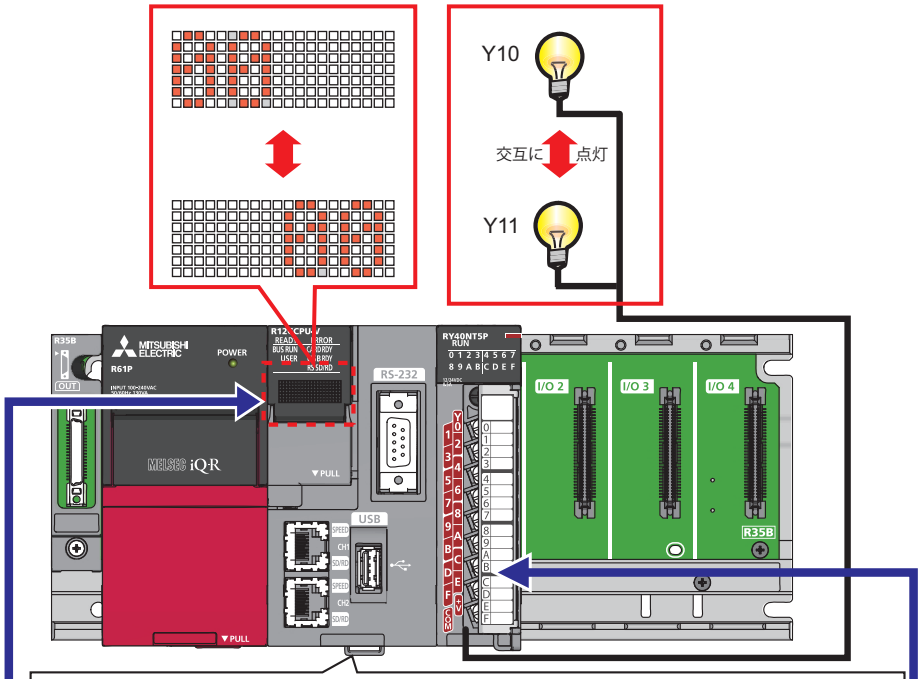
本クイックスタートガイドでは，出力モード(usLedMode)にASCIIモード(1)を指定します。

2.5　プログラミングする

出力ユニットに接続したランプとC言語コントローラユニット前面のドットマトリクスLEDを点滅させるプログラムを作成します。

今回作成するプログラムと制御内容

C言語コントローラユニットをRUNすると，出力ランプY10，Y11が交互に点灯を繰り返します。
また，出力ランプの点灯に同期して，C言語コントローラユニット前面のドットマトリクスLEDが交互に"00__"と"__00"の表示を繰り返します。




```
/* 出力制御 */
sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD, &usDataBuf, DUMMY);
if(sRet != 0){
    printf("ERROR : CCPU_Y_Out_WordEx_1 [%d(%04hxH)]\n", sRet, sRet);
    return;
}

/* ドットマトリクスLED制御 */
sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
if(sRet != 0){
    printf("ERROR : CCPU_SetDotMatrixLED_1 [%d(%04hxH)]\n", sRet, sRet);
    return;
}
```

ソースコード

ソースコードを下記に示します。

```
/*****************************************************************************************/
/* 関数ヘッダ */
/*****************************************************************************************/
#include        <vxworks.h>     /* VxWorks関数ヘッダ                                    */
#include        <taskLib.h>     /* VxWorks関数ヘッダ                                    */
#include        <stdio.h>       /* 標準関数ヘッダ                                       */
#include        <string.h>      /* 標準関数ヘッダ                                       */
#include        "CCPUFunc.h"    /* C言語コントローラユニット関数ヘッダ                   */
/*****************************************************************************************/
/* 定義 */
/*****************************************************************************************/
/* デバッグ用 */
#define UNIT_XY         0x0010 /* ユニットの先頭入出力番号                              */
#define RY_LED          0x5555 /* Y信号の初期出力値（偶数ビット点灯）                   */
#define LED_0           0x30    /* ドットマトリクスLED出力初期値（LED1,2）              */
#define LED_SPACE       0x20    /* ドットマトリクスLED出力初期値（LED3,4）              */

/* C言語コントローラユニット専用関数用                                                  */
#define WORD                    1 /* ワード単位指定                                     */
#define NORMAL_ACCESS           0 /* 通常アクセス指定                                   */
#define DUMMY                   0 /* ダミー                                             */
#define DISPMODE_ASCII          1 /* ドットマトリクスLED出力モード                      */
/*****************************************************************************************/
/* Y出力、ドットマトリクスLED制御処理 */
/*****************************************************************************************/
void R12_SampleTask()
{
        /* ローカル変数の宣言 */
        short           sRet;           /* C言語コントローラユニット専用関数の戻り値    */
        unsigned short  usDataBuf;      /* Y信号ワードアクセス用 */
        unsigned short  usEmptyDataBuf; /* Y信号リセット用 */
        char            pcdata[4];      /* ドットマトリクスLED点灯値 */
        short           i;              /* ループ用                                      */

        /* Y信号の出力値をセット（偶数ビットを点灯）*/
        usDataBuf = RY_LED;

        /* ドットマトリクスLEDの出力値をセット（LED1,2点灯） */
        pcdata[0] = LED_0;
        pcdata[1] = LED_0;
        pcdata[2] = LED_SPACE;
        pcdata[3] = LED_SPACE;

        /* 出力制御、ドットマトリクスLED制御処理を20回ループ */
        for(i = 0; i < 20; i++){
                /* 出力制御 */
                sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD, &usDataBuf, DUMMY);
                if(sRet != 0){
                        printf("ERROR : CCPU_Y_Out_WordEx_1 [%d(%04hxH)]\n", sRet, sRet);
                        return;
                }
```

```
		/* ドットマトリクスLED制御 */
		sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
		if(sRet != 0){
			printf("ERROR : CCPU_SetDotMatrixLED_1 [%d(%04hxH)]\n", sRet, sRet);
			return;
		}

		/* Y信号の出力値を反転する（奇数ビット→偶数ビット→...を点灯） */
		usDataBuf = ~usDataBuf;

		/* ドットマトリクスLEDの出力値を入れ替える（LED1,2点灯→LED3,4点灯） */
		if(i%2 ==0){
			pcdata[0] = LED_SPACE;
			pcdata[1] = LED_SPACE;
			pcdata[2] = LED_0;
			pcdata[3] = LED_0;
		}else{
			pcdata[0] = LED_0;
			pcdata[1] = LED_0;
			pcdata[2] = LED_SPACE;
			pcdata[3] = LED_SPACE;
		}
		/* ウェイト */
		taskDelay(60);
	}

	/* Y信号をリセットする */
	usEmptyDataBuf = 0x00;
	sRet = CCPU_Y_Out_WordEx(NORMAL_ACCESS, UNIT_XY, WORD,&usEmptyDataBuf, DUMMY);
	if(sRet != 0){
		printf("ERROR : CCPU_Y_Out_WordEx_2 [%d(%04hxH)]\n", sRet, sRet);
		return;
	}

	/* ドットマトリクスLEDをリセットする */
	pcdata[0] = LED_SPACE;
	pcdata[1] = LED_SPACE;
	pcdata[2] = LED_SPACE;
	pcdata[3] = LED_SPACE;
	sRet = CCPU_SetDotMatrixLED(DISPMODE_ASCII, &pcdata[0]);
	if(sRet != 0){
		printf("ERROR : CCPU_SetDotMatrixLED_2 [%d(%04hxH)]\n", sRet, sRet);
		return;
	}
}
```

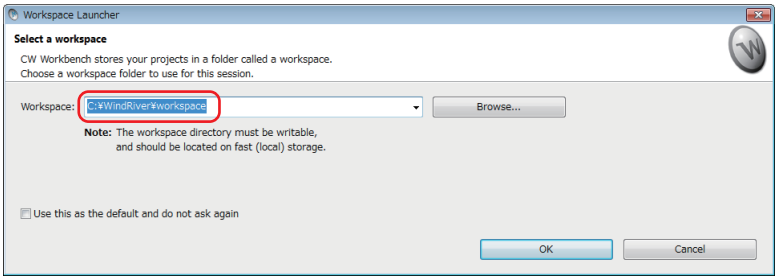
プロジェクトを作成する

CW Workbenchを起動する

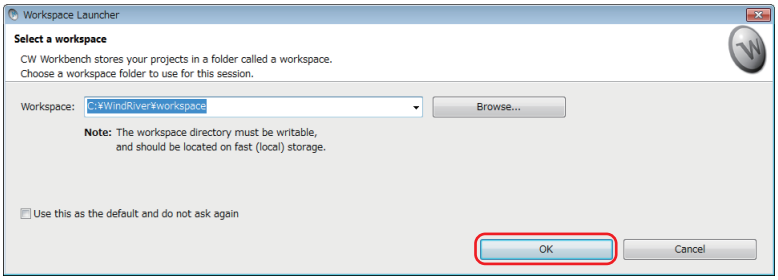
操作手順

1. [スタート]⇨[すべてのプログラム]⇨[Wind River]⇨[CW Workbench 3.3]⇨[CW Workbench 3.3]を選択します。

2. 起動後，プロジェクトの保存先フォルダを入力します。
ここでは，"C:\WindRiver\workspace"とします。



3. [OK]ボタンをクリックします。



CW Workbenchのメイン画面が表示されます。

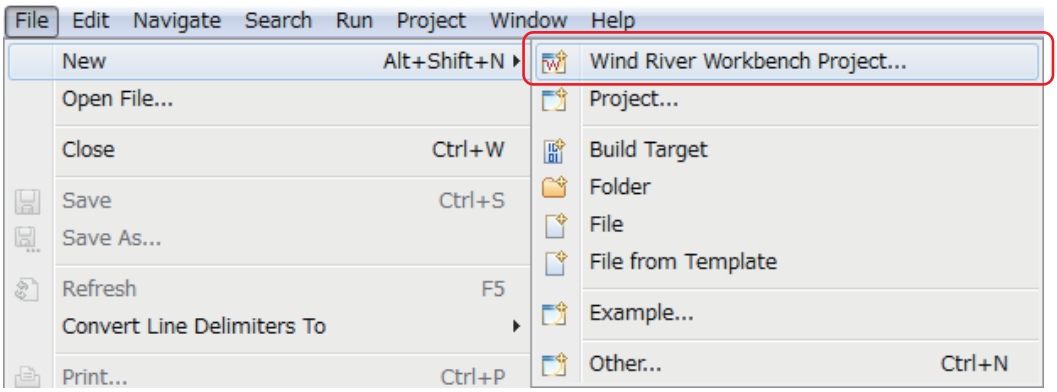
Point

- CW Workbenchの初期化状態の各ウィンドウの大きさやアイコンなどの配置は，お使いのパソコンによって変わります。本クイックスタートガイドに記載している画面と違う場合は，各ウィンドウの大きさを調整してください。
- 拡大したり消したりした各ウィンドウを初期状態に戻すには，[Window]⇨[New Window]を選択してください。

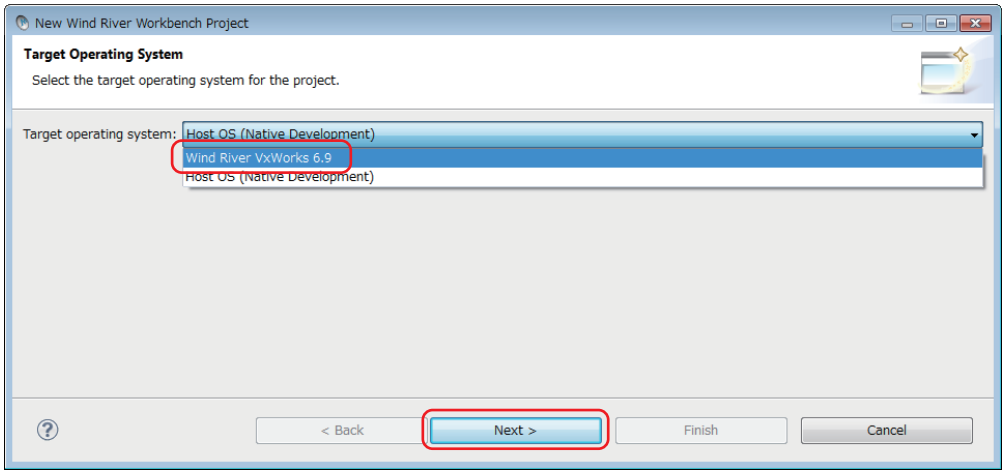
新規プロジェクトを作成する

操作手順

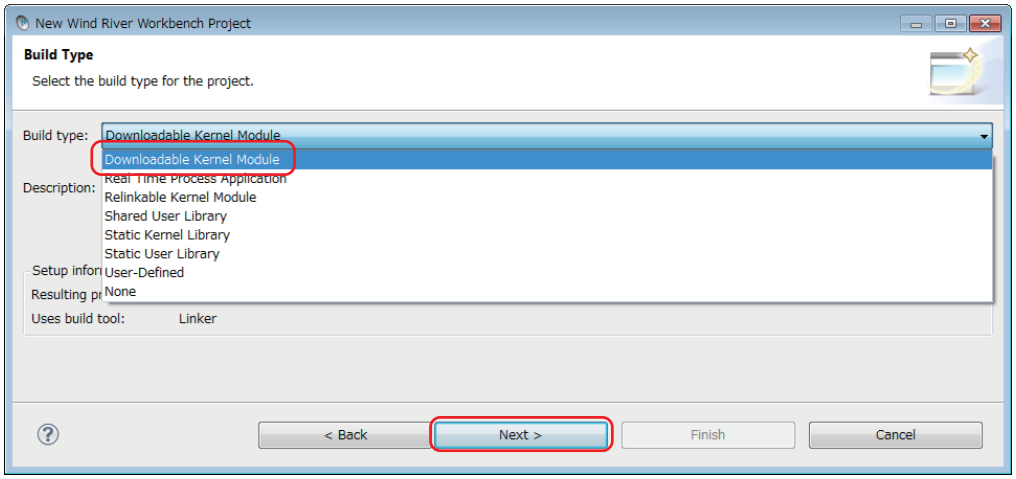
1. [File]⇨[New]⇨[Wind River Workbench Project]を選択します。



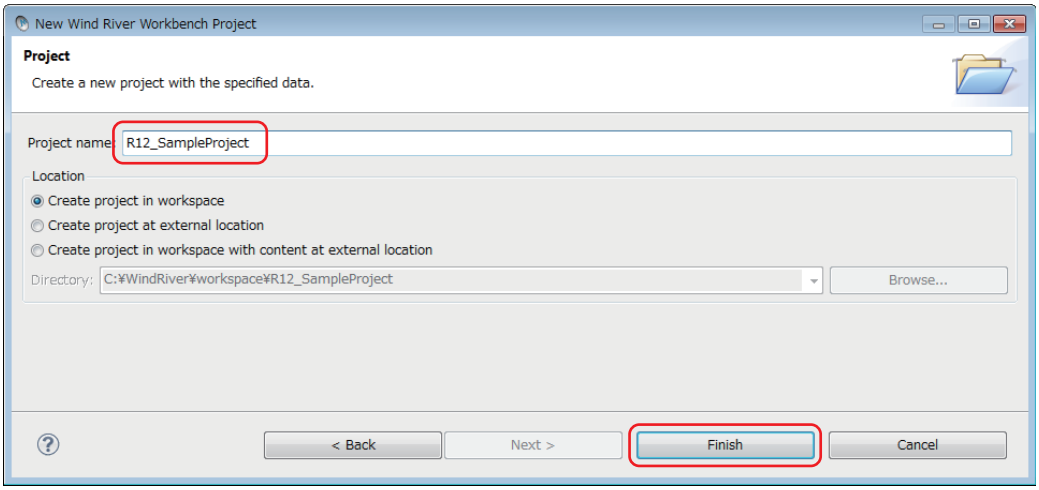
2. "Wind River VxWorks 6.9"を選択し，[Next]ボタンをクリックします。



3. "Downloadable Kernel Module"を選択し，[Next]ボタンをクリックします。



4. プロジェクト名を入力し，[Finish]ボタンをクリックします。
(プロジェクト名: 本章では"R12_SampleProject"とします。)



プロジェクトのプロパティを設定する

作成したプロジェクトを，C言語コントローラユニットで実行可能なモジュールに変換(ビルド)するための設定を行います。

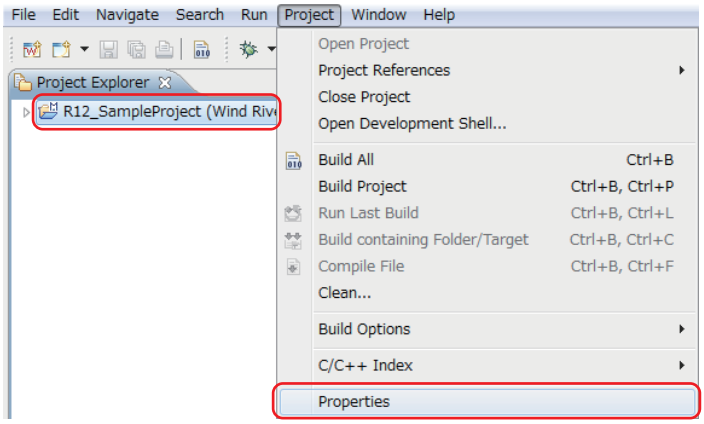
Point

ビルド: プロセッサに応じたソースコードのコンパイル，インクルードファイルとのリンクを行います。

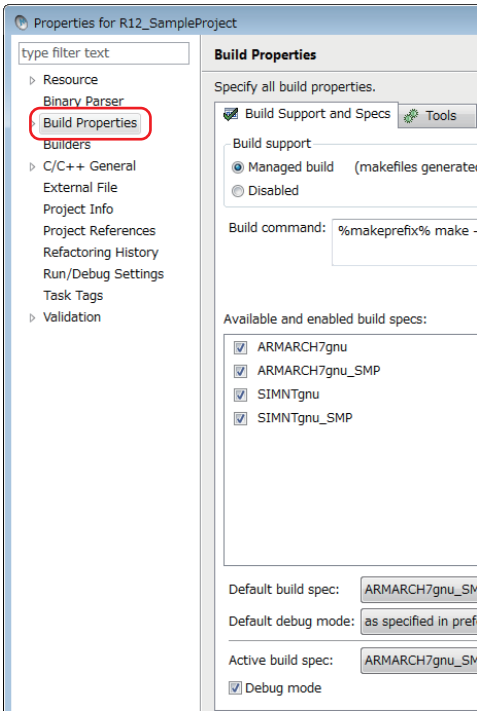
■使用するプロセッサを設定

操作手順

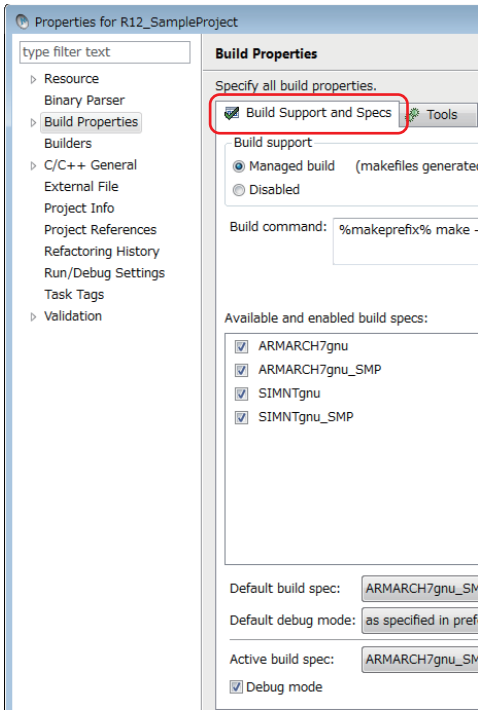
1. "Project Explorer"ウィンドウから作成したプロジェクトを選択し，[Project]⇨[Properties]を選択します。



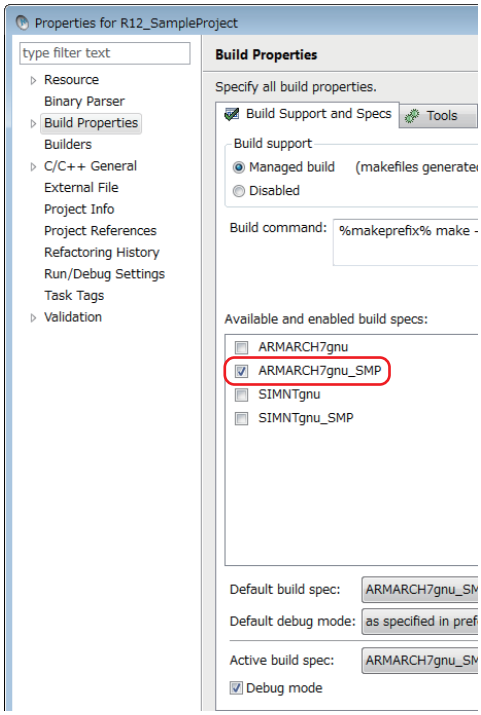
2. 画面左側のツリーから"Build Properties"を選択します。



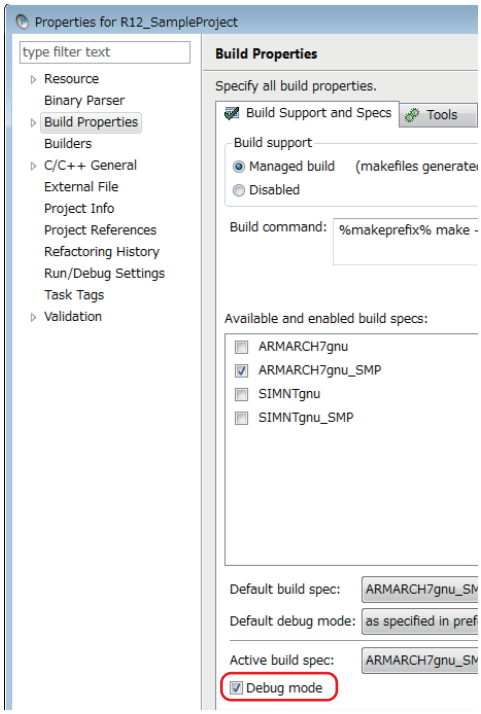
3. [Build Support and Specs]タブを選択します。



4. "Available and enabled build specs"で"ARMARCH7gnu_SMP"のみにチェックを入れます。



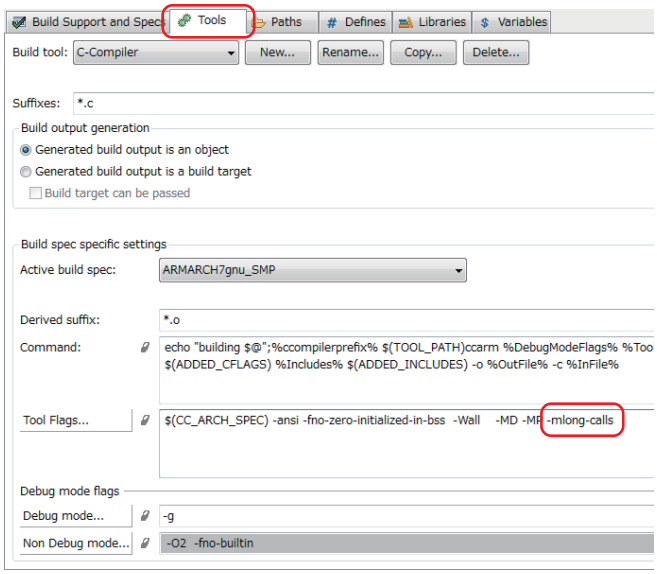
5. "Debug mode"にチェックを入れます。



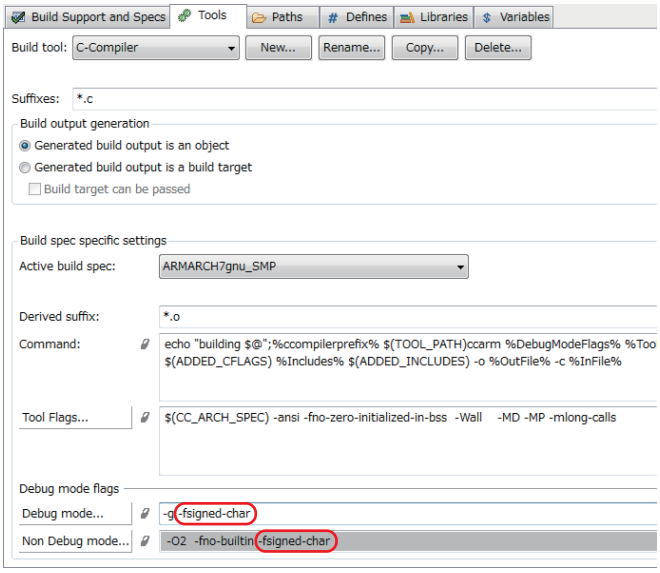
Point

正式なシステム運用，稼動時は，"Debug mode"のチェックを外してください。

6. [Tools]タブを選択し，[Tool Flags]に"-mlong-calls"を入力します。



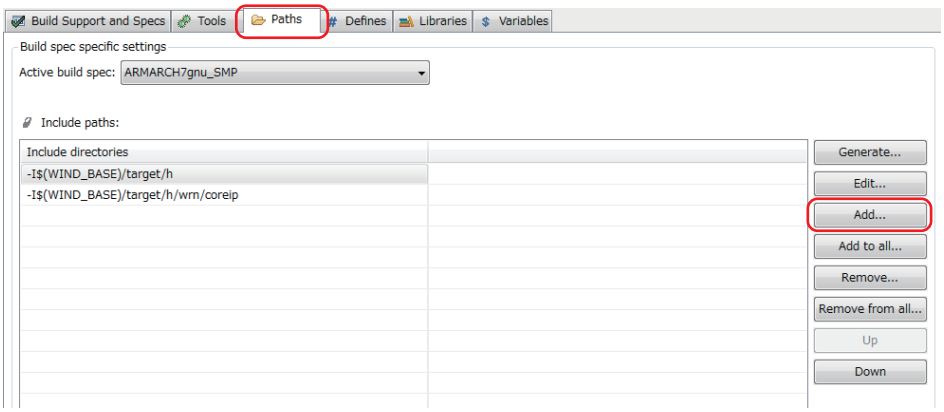
7. 同様に，"Debug mode flags"の[Debug mode]および[Non Debug mode]に"-fsigned-char"を入力します。



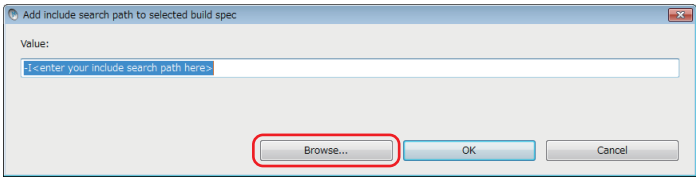
■インクルードファイルを設定

操作手順

1. [Paths]タブを選択し，[Add]ボタンをクリックします。

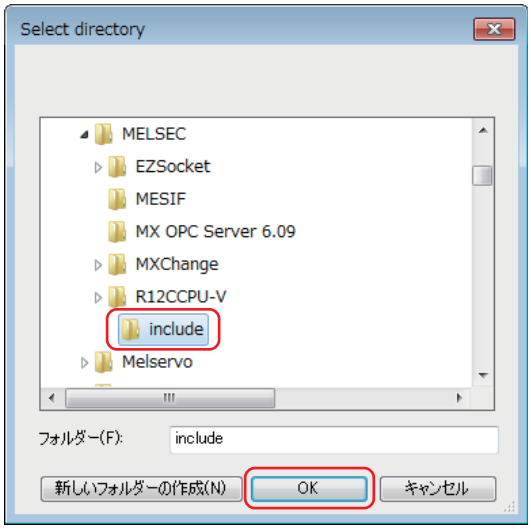


2. [Browse]ボタンをクリックします。



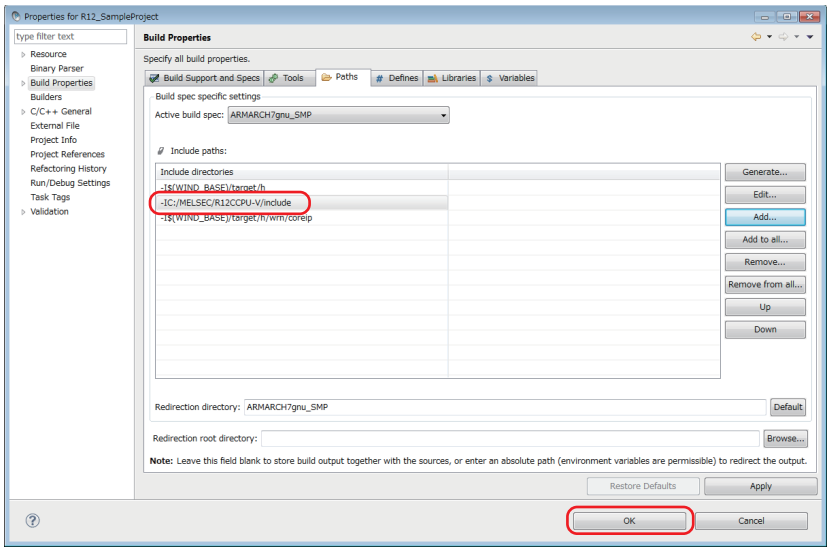
3. "Select directory"画面からC言語コントローラユニット専用のインクルードフォルダを選択し，[OK]ボタンをクリックします。

ここでは"C:\MELSEC\R12CCPU-V"とした場合のフォルダになります。

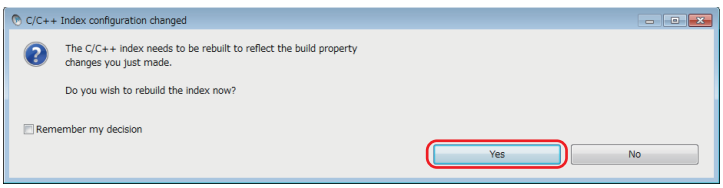


4. "Add include search path to selected build spec"画面で選択したフォルダが指定されていることを確認し，[OK]ボタンをクリックします。

5. 追加したインクルードパスが"Include paths"に表示されていることを確認し，[OK]ボタンをクリックします。



6. [OK]ボタンをクリック後，下記メッセージが表示された場合は，[Yes]ボタンをクリックします。



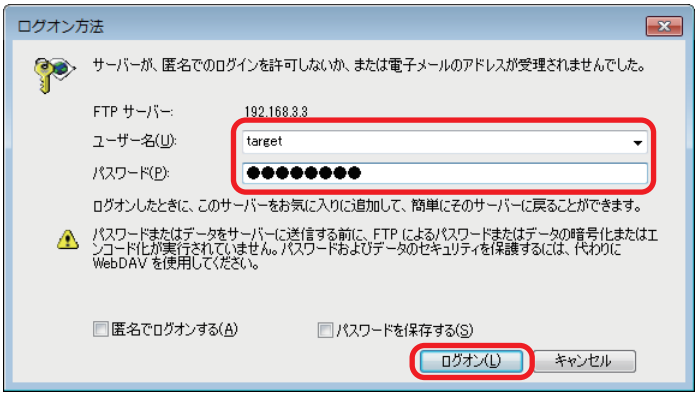
7. インクルードファイルを手順1～5で追加したインクルードフォルダに追加します。
C言語コントローラに格納されているインクルードファイルを取得するために，エクスプローラを起動し，アドレス欄に下記の形式で入力します。
ftp://192.168.3.3/SYSTEMROM/INCLUDE/

"ログオン方法"画面が表示されます。

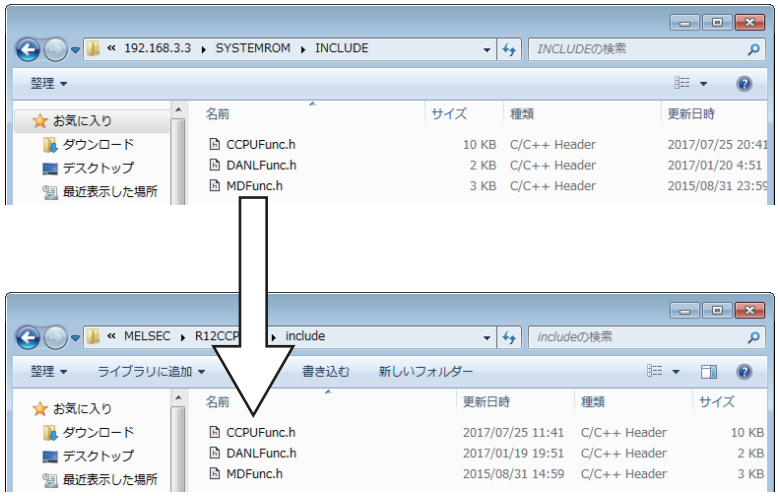
8. "ログオン方法"画面で下記のユーザ名とパスワードを入力します。

- ユーザ名: target
- パスワード: password

9. [ログオン]ボタンをクリックします。



10. ヘッダファイルを，手順1～5で追加したインクルードフォルダにコピーします。



ユーザプログラムを準備する

C言語コントローラシステムの制御を行うユーザプログラムを準備します。
本クイックスタートガイドでは，専用のサンプルプログラムを使用します。

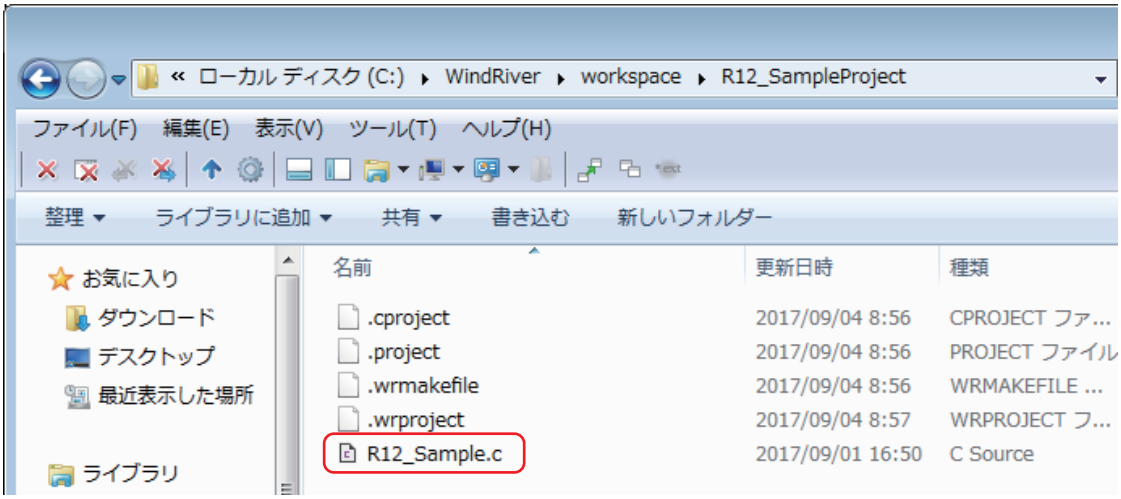
Point

本クイックスタートガイド専用のサンプルプログラムは，三菱電機FAサイトからダウンロードできます。
www.MitsubishiElectric.co.jp/fa

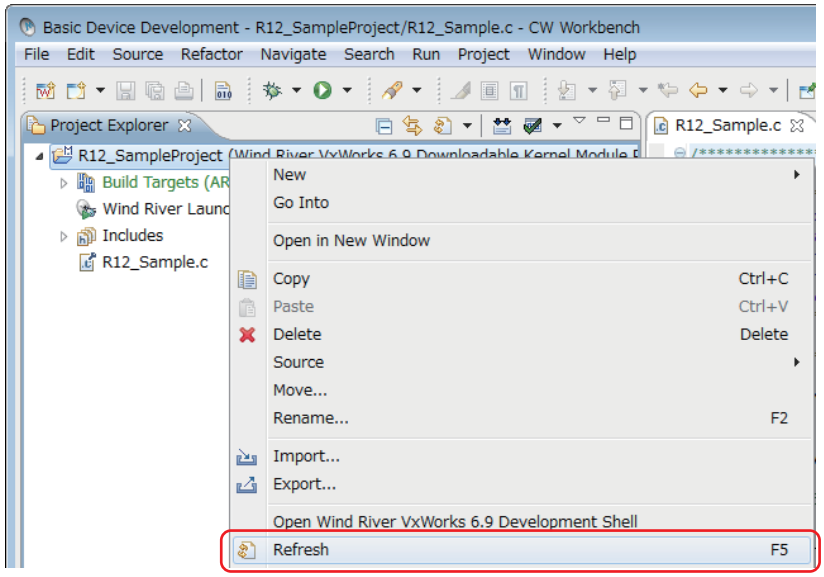
サンプルプログラムを追加する

操作手順

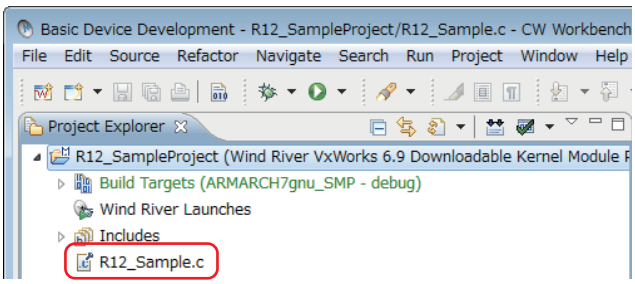
1. 今回作成したプロジェクトフォルダ直下に，本クイックスタートガイド専用のサンプルプログラムを格納します。
"C:\WindRiver\workspace\R12_SampleProject"



2. "Project Explorer"ウィンドウから作成したプロジェクトを選択して右クリックし，[Refresh]を選択します。



3. 手順1で格納したサンプルプログラムがプロジェクトに追加されます。

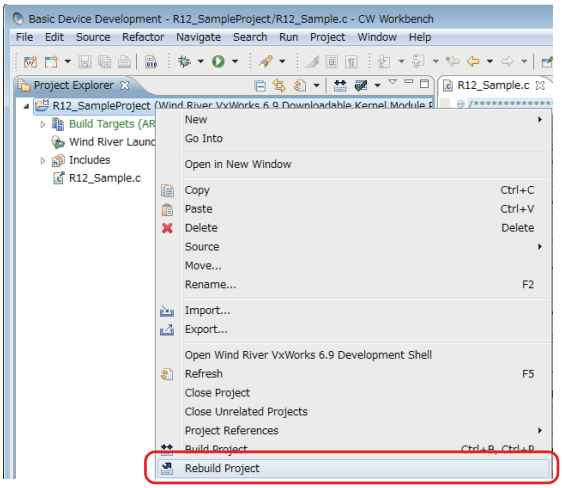


ユーザプログラムから実行モジュールを生成する

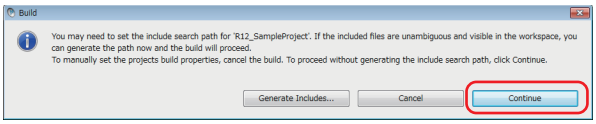
作成したプログラムを，C言語コントローラユニットで実行可能なモジュールに変換(ビルド)します。

操作手順

1. "Project Explorer"ウィンドウから作成したプロジェクトを選択して右クリックし，[Rebuild Project]を選択します。

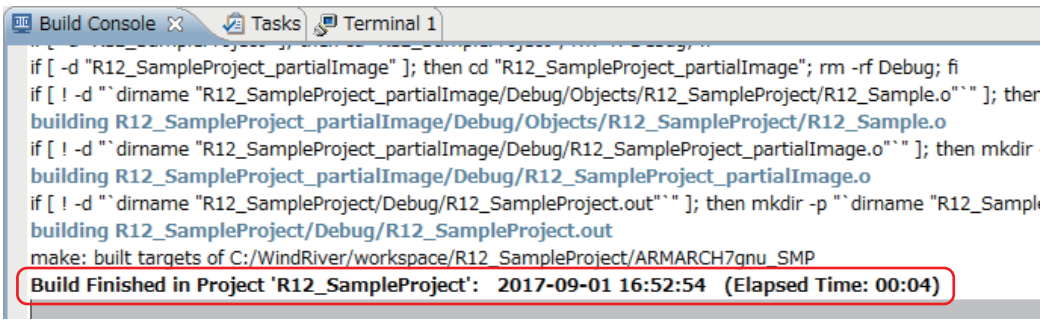


2. [Rebuild Project]を選択後，下記のメッセージが表示された場合は，[Continue]ボタンをクリックします。



プロジェクトのビルドが開始され，"Build Console"ウィンドウにビルドの処理過程が表示されます。

3. "Build Console"ウィンドウで，"Build Finished..."が表示されることを確認してください。



Point

"Build Finished..."が表示されずエラーが発生した場合は，エラー内容を確認してプログラムを修正してください。
プログラムを修正後，再度「36ページ ユーザプログラムから実行モジュールを生成する」から実施してください。

C言語コントローラユニットとCW Workbenchを接続する

CW WorkbenchでデバッグをCW Workbenchを行うために，C言語コントローラユニットのユーザEthernetポートCH1とCW Workbenchを接続します。

操作手順

1. C言語コントローラユニットからVxWorksイメージファイルを取得するために，エクスプローラを起動し，アドレス欄に下記の形式で入力します。

ftp://192.168.3.3/SYSTEMROM/OS_IMAGEFILE/

"ログオン方法"画面が表示されます。

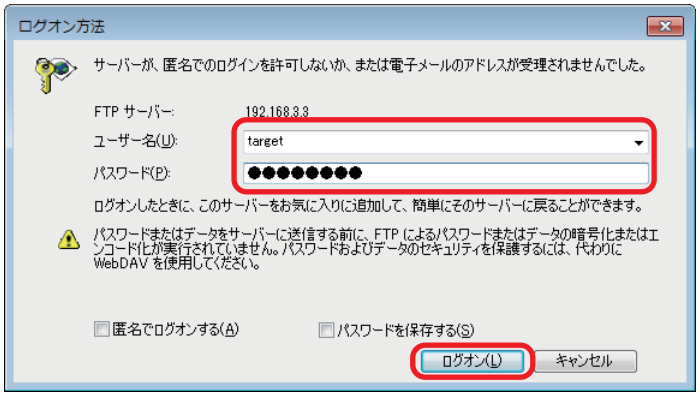
Point

C言語コントローラユニットとパソコンで通信する場合は，双方で同じVxWorksイメージファイルを指定する必要があります。

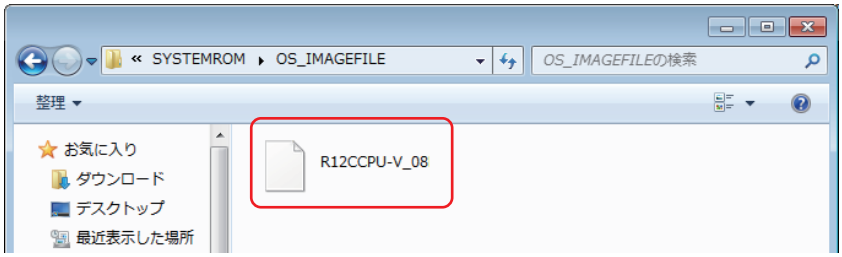
2. "ログオン方法"画面で下記のユーザ名とパスワードを入力します。

- ユーザ名: target
- パスワード: password

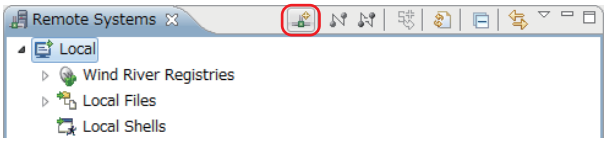
3. [ログオン]ボタンをクリックします。



4. "C:\MELSEC\R12CCPU-V\CCPUTool"フォルダを作成し，C言語コントローラユニットに格納されているVxWorksイメージファイルを"C:\MELSEC\R12CCPU-V\CCPUTool"にコピーします。

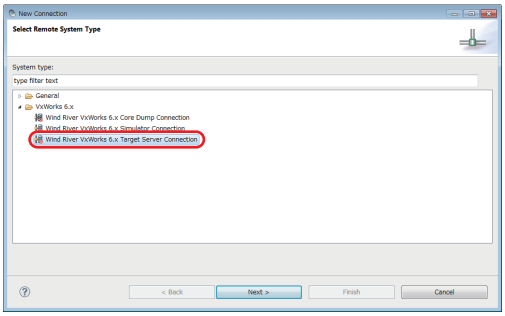


5. "Remote Systems"ウィンドウ内の  をクリックします。

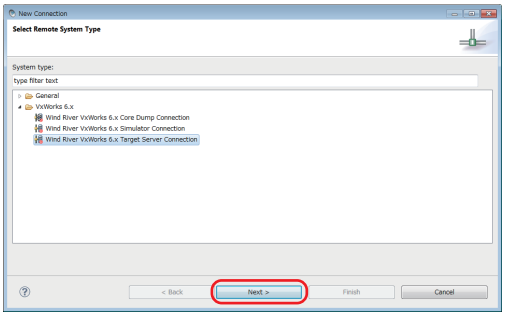


"New Connection"画面が表示されます。

6. "New Connection"画面にて"Wind River VxWorks 6.x Target Server Connection"を選択します。

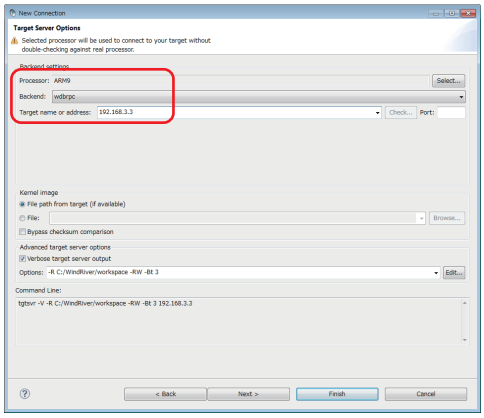


7. [Next]ボタンをクリックします。

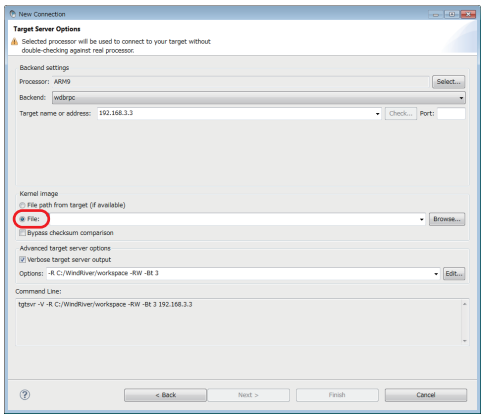


8. "Backend settings"の設定項目で，下記の内容を設定します。

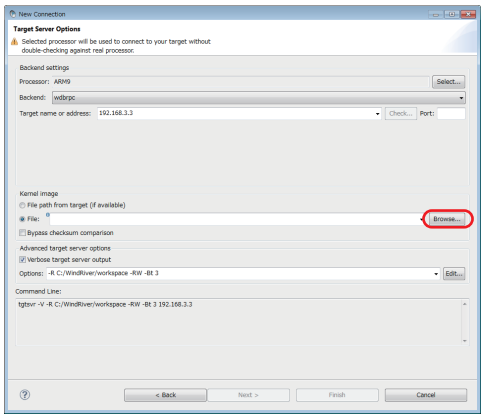
- Processor: ARM9([Select]ボタンをクリックし，ツリーから選択)
- Backend: wdbrpc
- IP Address: 192.168.3.3(デフォルト)
- Port: 空欄



9. "Kernel image"で"File"を選択します。

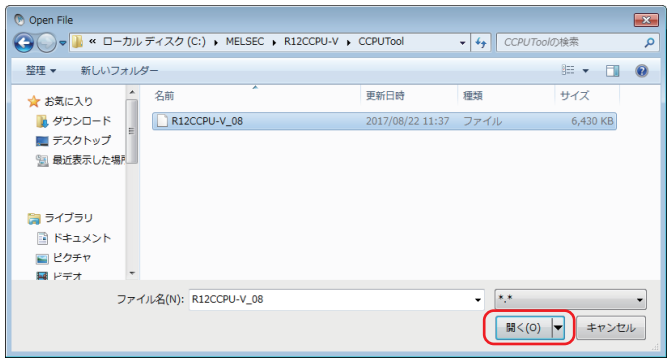


10. [Browse]ボタンをクリックします。

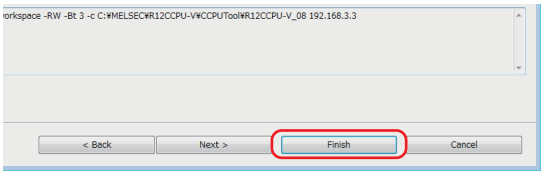


"Open File"画面が表示されます。

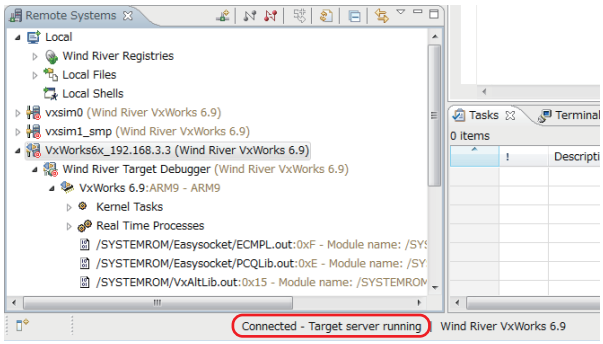
11. 手順4で"C:\MELSEC\R12CCPU-V\CCPUTool"にコピーしたVxWorksイメージファイルを選択し，[開く]ボタンをクリックします。



12. [Finish]ボタンをクリックします。



13. "Remote Systems"ウィンドウの下部に"Connected - Target server running" が表示されると接続完了です。



Point

"Connected - Target server running"が表示されない場合は，C言語コントローラユニットの電源が正常に投入されていることを確認し，再度「37ページ C言語コントローラユニットとCW Workbenchを接続する」から実施してください。

ユーザプログラムをデバッグする

作成したプログラムが正しく動作するか確認します。

■ユーザプログラムをC言語コントローラユニットにダウンロードする

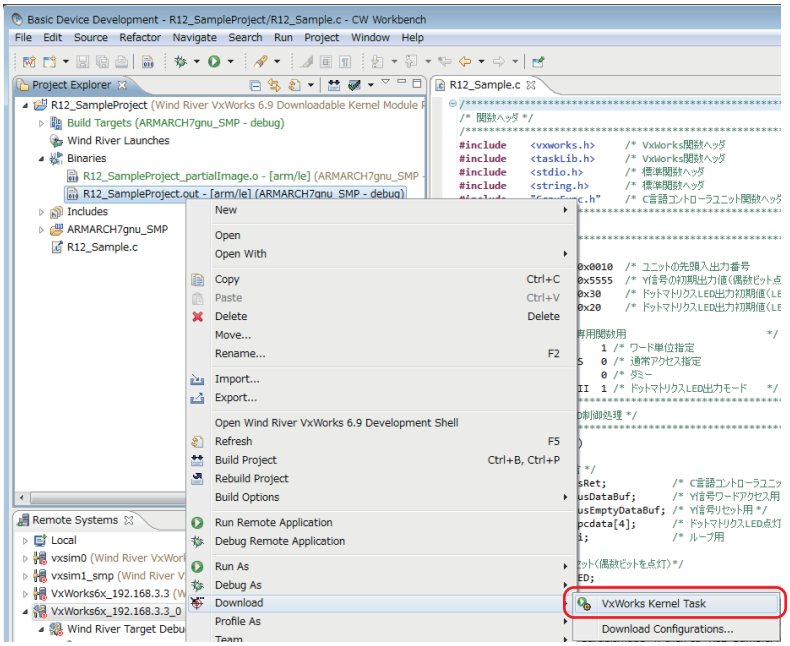
デバッグを行うために，実行モジュールをC言語コントローラユニットのメモリにダウンロードします。
ダウンロードを行うと，スクリプトファイルがなくてもプログラムが実行できます。

Point

スクリプトファイル: C言語コントローラユニットを立上げ時に起動するユーザプログラムのロード先，起動順序などを記述するファイルです。

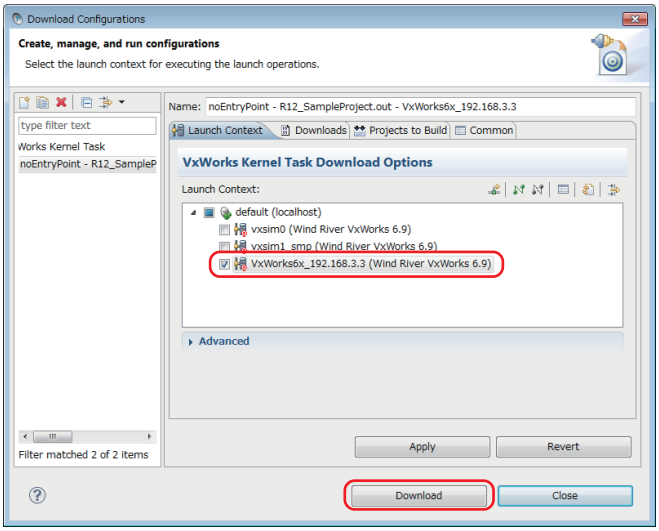
操作手順

1. "Project Explorer"ウィンドウから作成したモジュールファイル"R12_SampleProject.out"を選択して右クリックし，[Download]⇨[VxWorks Kernel Task]を選択します。



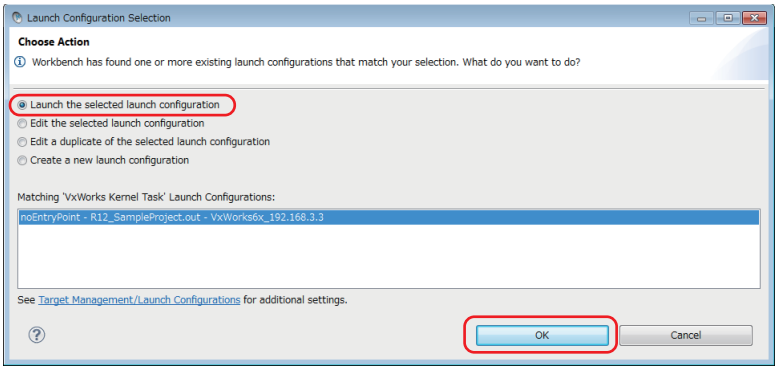
"Download Configurations"画面が表示されます。

2. [Launch Context]タブにて"VxWorks6x_192.168.3.3(Wind River VxWorks 6.9)"のみにチェックを入れ，[Download]ボタンをクリックします。



Point

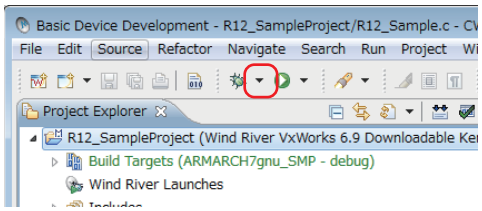
2回目以降の2の操作では，"Launch Configuration Selection"画面が表示されます。
"Launch the selected launch configuration"を選択し，[OK]ボタンをクリックしてください。



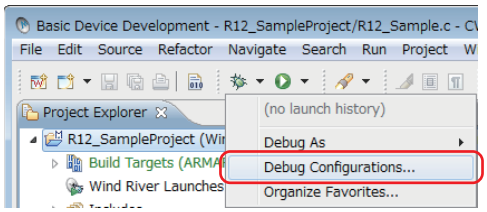
■ユーザプログラムをデバッグする

操作手順

1. "Project Explorer"ウィンドウから，作成したプロジェクトを選択し，ツールバーの右脇の[▼]ボタンをクリックします。

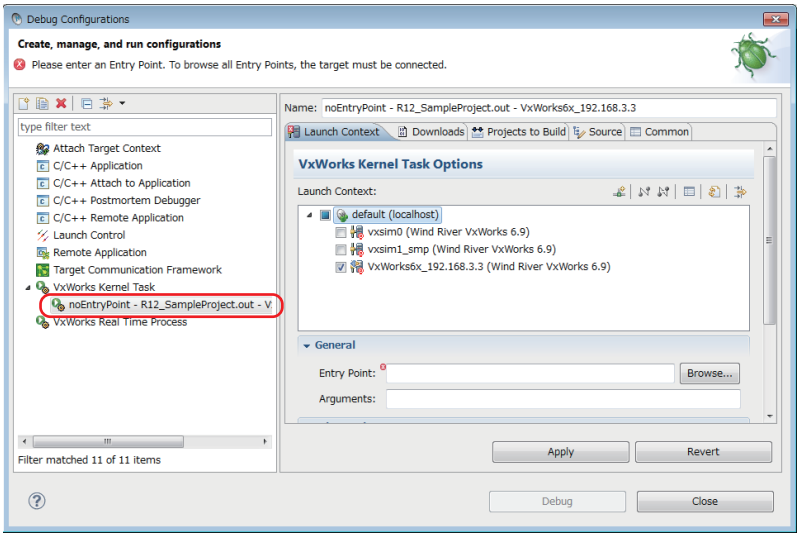


2. [Debug Configurations]を選択します。

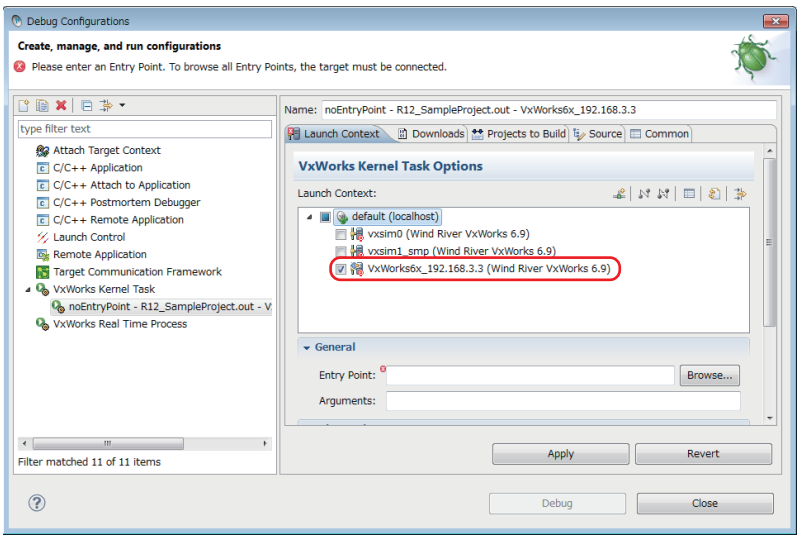


"Debug Configurations"画面が表示されます。

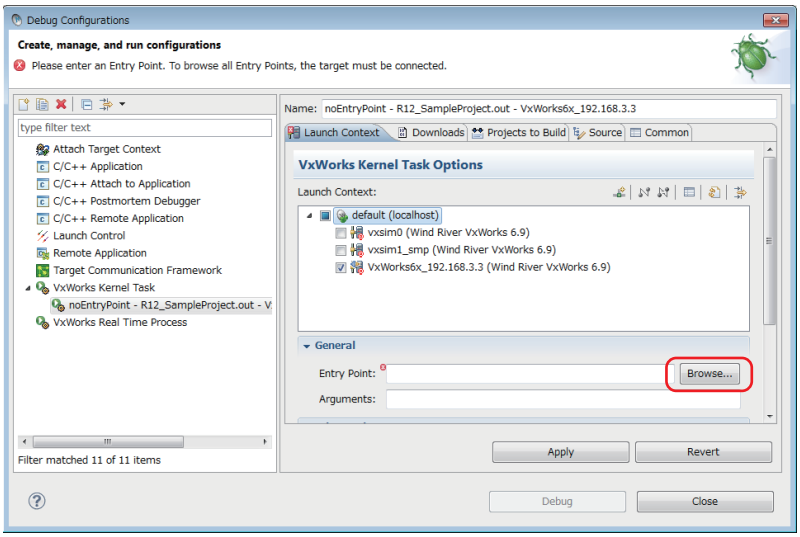
3. "VxWorks Kernel Task"からダウンロードしたモジュール"R12_SampleProject.out"を選択します。



4. C言語コントローラユニットとの接続を示すターゲットサーバにチェックを入れます。

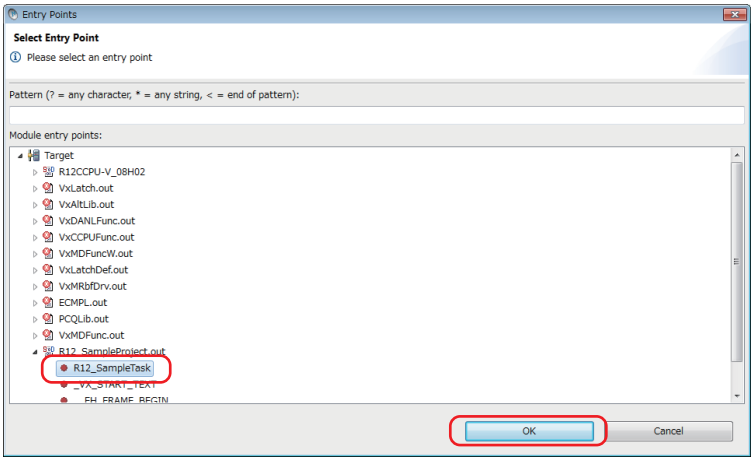


5. [Browse]ボタンをクリックします。

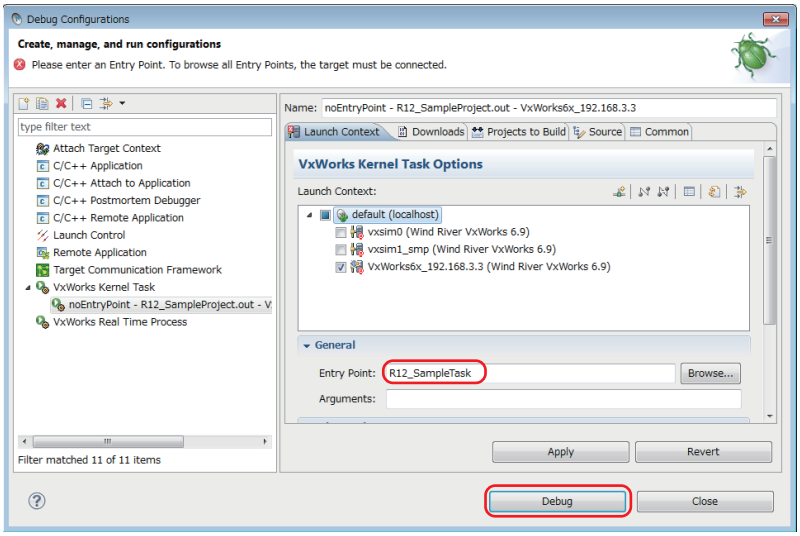


"Entry Points"画面が表示されます。

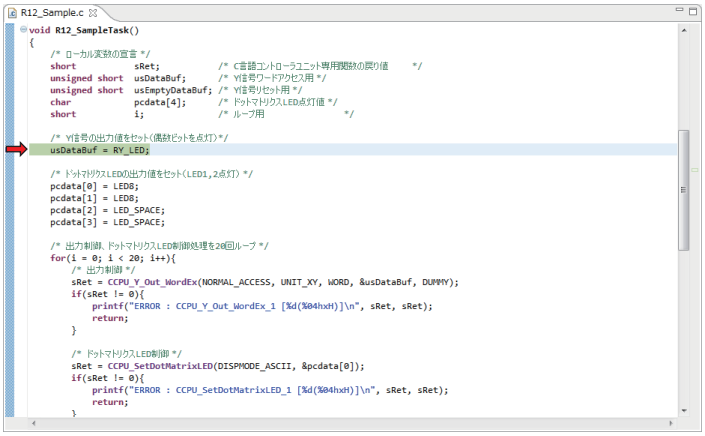
6. デバッグを開始する関数"R12_SampleTask"を選択し，[OK]ボタンをクリックします。



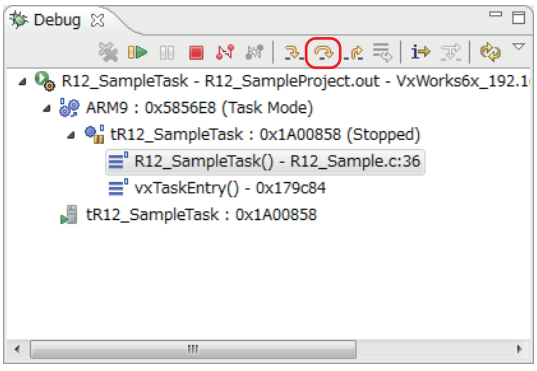
7. "Entry Point"に手順6で選択した関数名が設定されていることを確認し，[Debug]ボタンをクリックします。



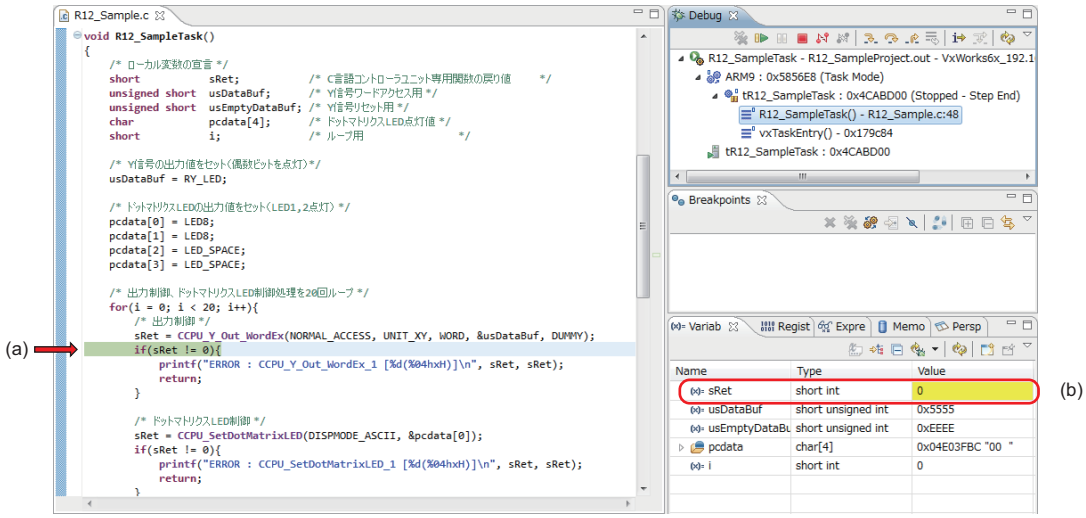
8. デバッグが開始され，"Entry Point"で指定した関数の先頭でプログラムの実行が止まります。



9. "Debug"ウィンドウので，1ステップごとにデバッグを行います。



10. 画面右下の"Variables"ウィンドウで，各変数の値の確認や変更を行うことができます。
ここではCCPU関数の戻り値である"sRet"が0(正常値)であることを確認します。
(a)手順9のステップ実行で「→」の行まで実行する。
(b)[Variables]タブで，sRetの値が0(正常値)であることを確認する。



手順9，10を繰り返し，作成したプログラム全体をデバッグします。

Point

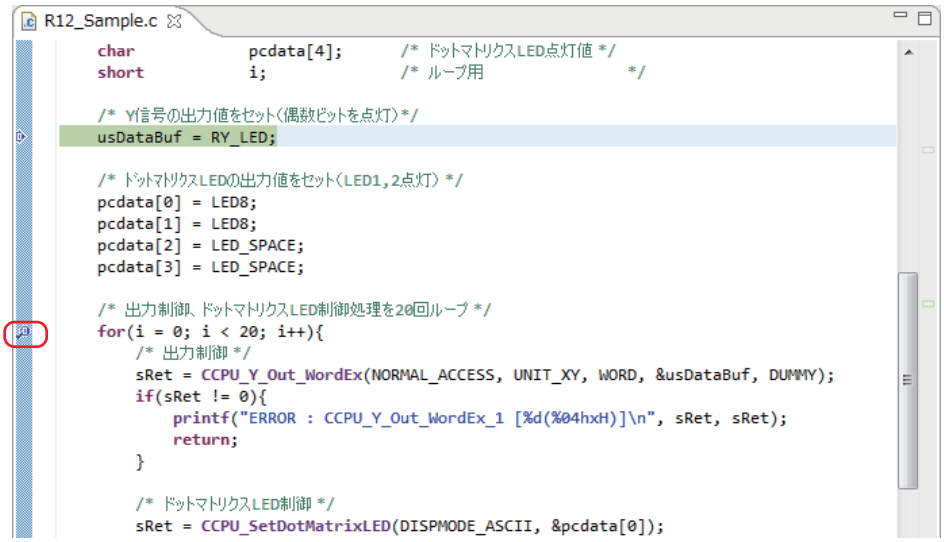
C言語コントローラユニット専用関数の戻り値が0以外の場合は，下記を参照してトラブルシューティングしてください。
📖MELSEC iQ-R C言語コントローラユニットプログラミングマニュアル

■Breakpointを使用したデバッグ方法

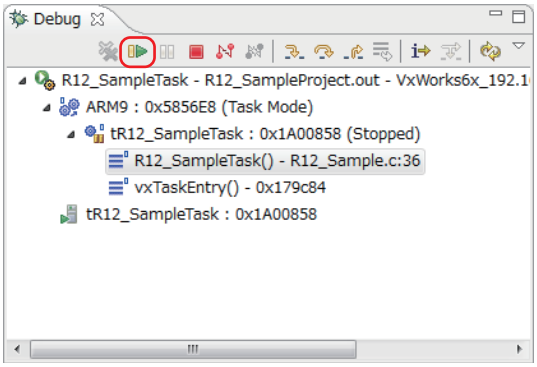
手順9の1ステップごとのデバッグではなく，プログラム内の任意の位置にBreakpointを指定してデバッグを進めることができます。

操作手順

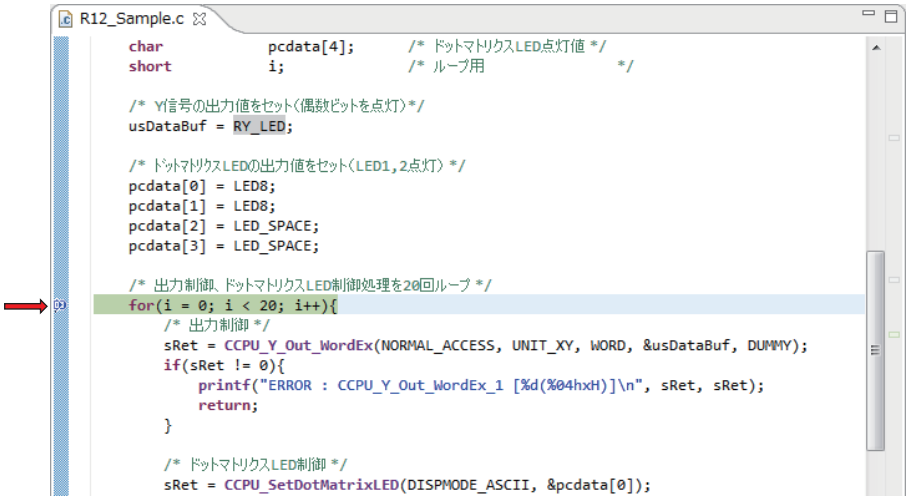
1. ソースファイルの左端をダブルクリックし，Breakpoint挿入します。



2. をクリックします。



指定したBreakpointの位置までプログラムが実行されます。

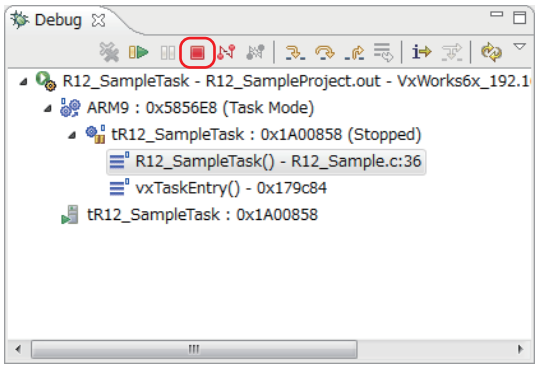


Point

各アイコンの内容は下記のとおりです。

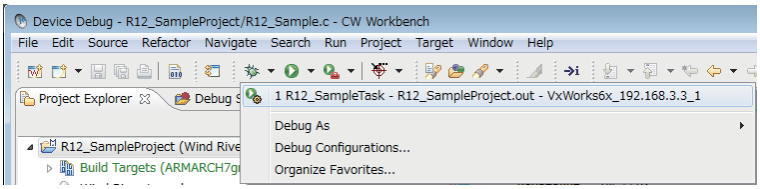
- : 1ステップ実行
  1ステップ単位で実行し，関数の場合は当該関数の中に入りステップ実行を続けます。
- : 1関数単位実行
  1ステップ単位で実行し，関数の場合は当該関数の中へは入らず関数単位でステップの実行を続けます。
- : 関数の最後(Return)まで実行します。
- : プログラム実行
- : プログラム停止
- : デバッグ終了

3. "Debug"ウィンドウの をクリックし，デバッグを終了します。



Point

再度デバッグを開始する場合は，ツールバー の右端の[▼]ボタンをクリックし，表示されたポップアップメニュー上部にある生成済みデバッグ構成を選択することでデバッグを開始できます。
また，上記の操作により「42ページ ユーザプログラムをデバッグする」の手順1～8を省くことができます。



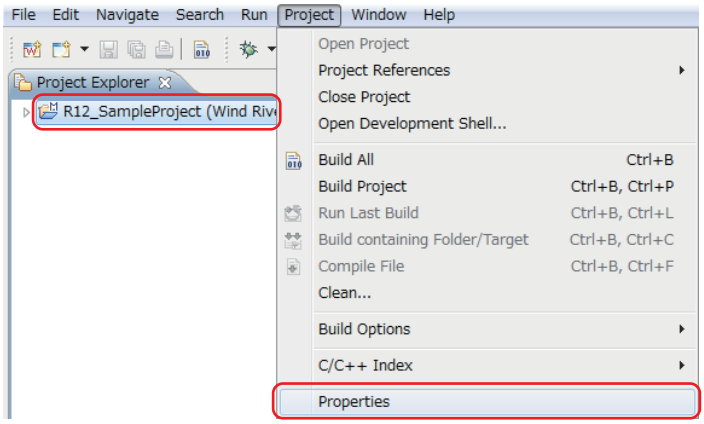
実行モジュールを登録する

作成したプログラムを稼動用にビルドし，C言語コントローラユニットに格納します。

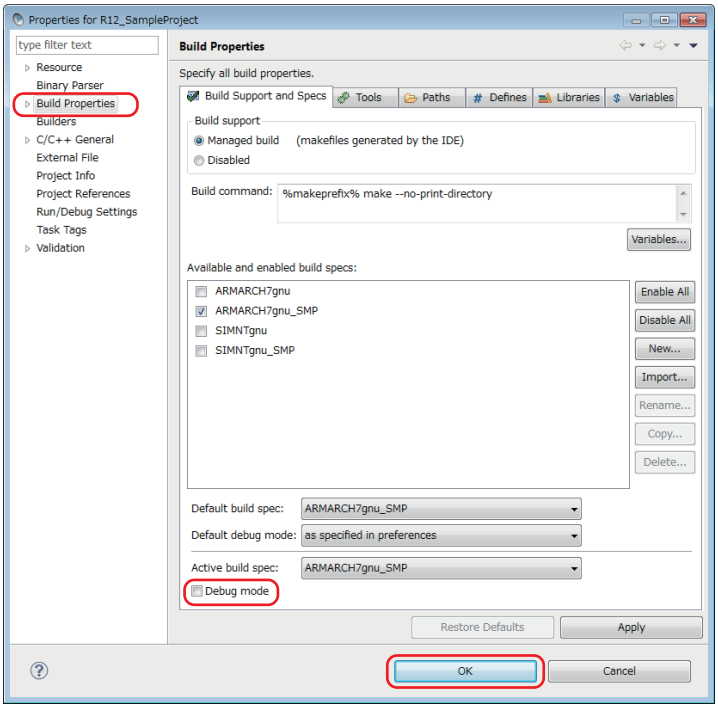
■ユーザプログラムをビルドする

操作手順

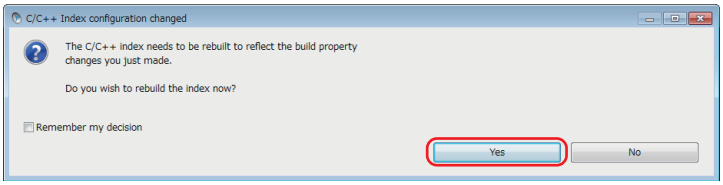
1. "Project Explorer"ウィンドウから作成したプロジェクトを選択し，[Project]⇨[Properties]を選択します。



2. 画面左側のツリーから"Build Properties"を選択し，"Debug mode"のチェックを外し，[OK]ボタンをクリックします。



3. [OK]ボタンをクリック後，下記メッセージが表示された場合は，[Yes]ボタンをクリックします。



4. 「36ページ ユーザプログラムから実行モジュールを生成する」に従ってビルドします。

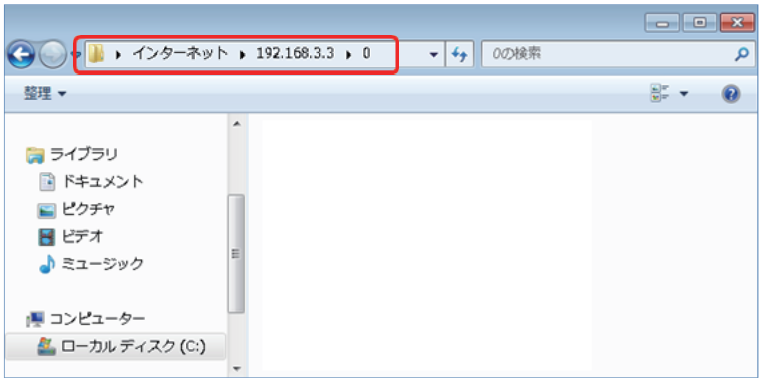
■ユーザプログラムを格納する

操作手順

1. エクスプローラを起動し，アドレス欄に下記の形式で入力します。

ftp://192.168.3.3/0

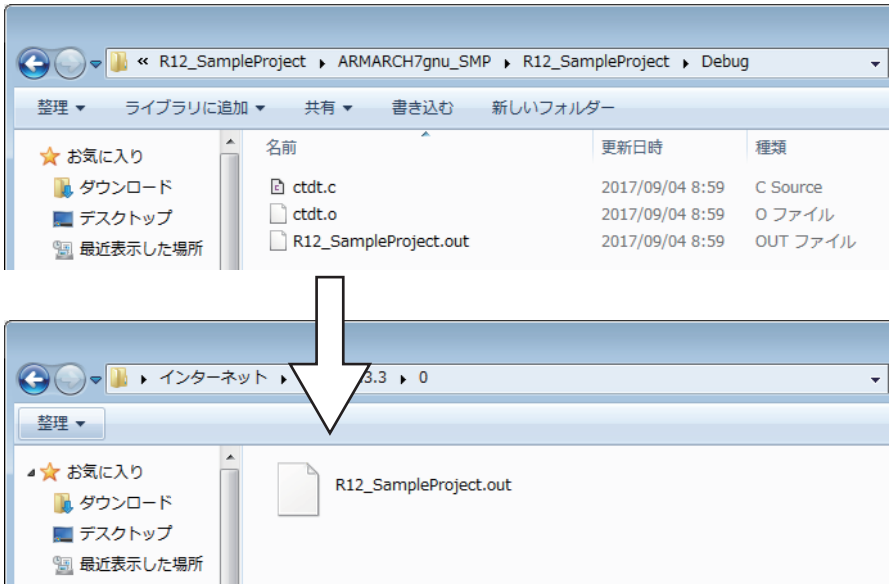
C言語コントローラユニットへログインすると，下記のように表示されます。



2. 作成したユーザプログラム"R12_SampleProject.out"を，ドラッグ&ドロップでC言語コントローラユニットのプログラムメモリ「0」へコピーします。

本クイックスタートガイドで作成したユーザプログラムは，下記に格納されます。

C:\WindRiver\workspace\R12_SampleProject\ARMARCH7gnu_SMP\R12_SampleProject\Debug

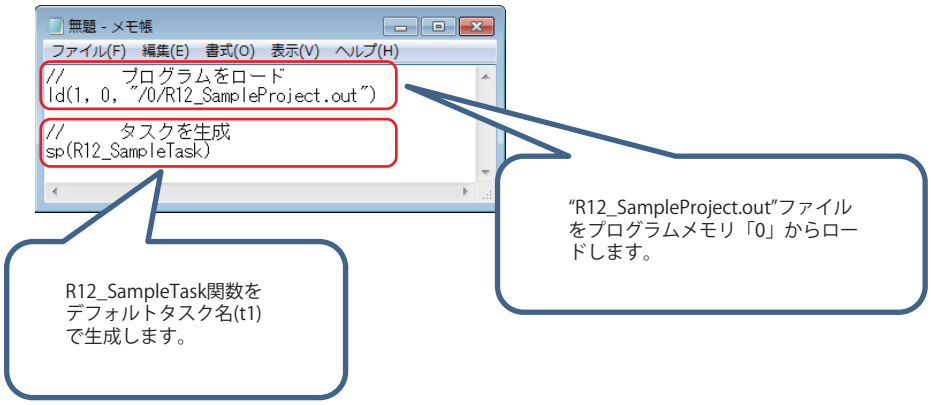


■スクリプトファイルを作成・格納する

C言語コントローラユニットを起動時に，実行モジュールを自動的にロードするためのスクリプトファイルを作成します。

操作手順

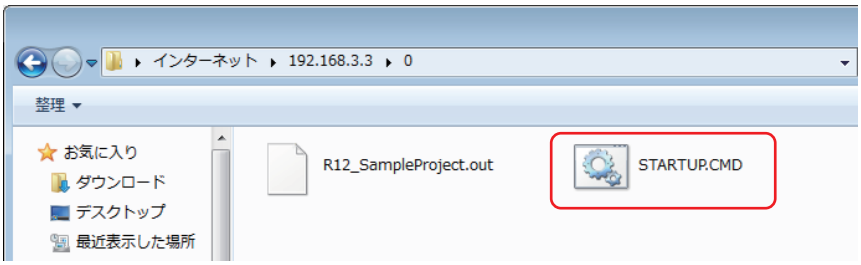
1. テキストファイルを開き，下記のようにユーザプログラムのロード，タスク生成を行うスクリプトファイルを記述します。



2. ファイル名を"STARTUP.CMD"として保存します。

3. 作成したスクリプトファイルを，C言語コントローラユニットのプログラムメモリへコピーします。

ftp://192.168.3.3/0



これでスクリプトファイルの作成・格納は完了です。

Point

ユーザプログラム，スクリプトファイルはプログラムメモリ以外に，SDメモリカードにも格納することができます。
スクリプトファイルを両方に格納した場合は，SDメモリカード内のスクリプトファイルが優先的に起動します。

2.6 動作を確認する

C言語コントローラユニットへ登録したプログラムを実行し，動作を確認します。
操作にはC言語コントローラユニット前面のRESET/STOP/RUNスイッチを使用します。
RESET/STOP/RUNスイッチの用途は下記のとおりです。

- RUN: ユーザプログラムからの出力(Y)，バッファメモリ書込み許可状態
- STOP: ユーザプログラムからの出力(Y)，バッファメモリ書込み禁止状態
- RESET: ユニットをリセット

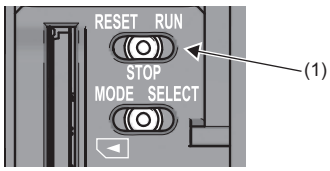
Point

- C言語コントローラユニットのプログラムの演算は，スイッチの状態がRUN/STOPにかかわらず実行されます。
- RUN/STOP/RESETスイッチの詳細については，下記を参照してください。
  (MELSEC iQ-R C言語コントローラユニットユーザーズマニュアル(スタートアップ編))

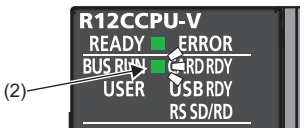
ユーザプログラムからの出力(Y)を許可

操作手順

1. C言語コントローラユニット前面のRESET/STOP/RUNスイッチ(1)をRUN側に倒します。



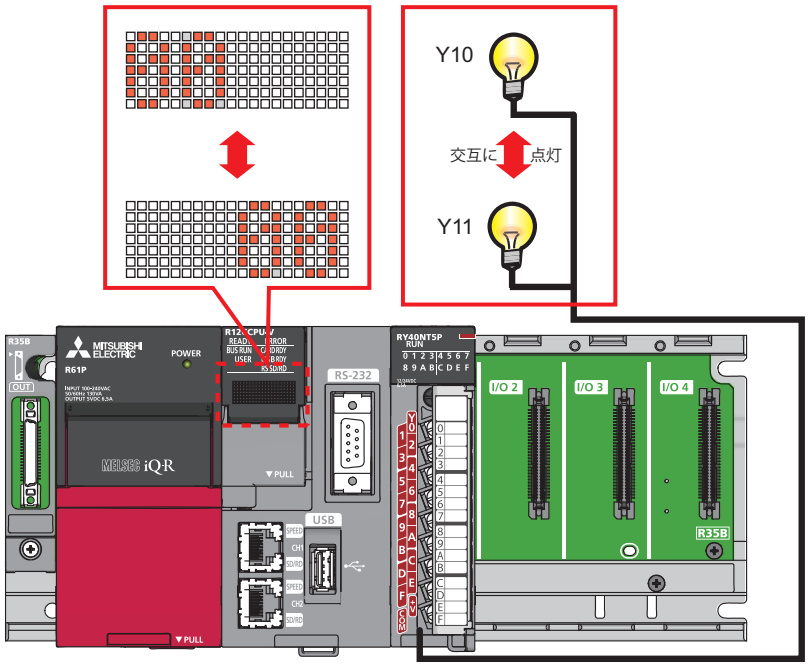
2. BUS RUN LED(2)が緑色点灯すれば，正常にプログラムが実行されています。



ドットマトリクスLED，ランプを使って，動作を確認する

C言語コントローラユニット前面のドットマトリクスLEDと出力ランプが，下記のように点灯します。

1. C言語コントローラユニット前面のドットマトリクスLEDの表示が，交互に10回，切り替わります。
2. ドットマトリクスLEDに同期し，出力ランプY10とY11が交互に点灯を繰り返します。



3. 再度確認したい場合は，C言語コントローラユニットをリセットします。

3 よく使う機能

C言語コントローラユニットで，システム立上げ時や運用・稼動後の保守などによく使う機能を紹介します。

3.1 エラーを確認する

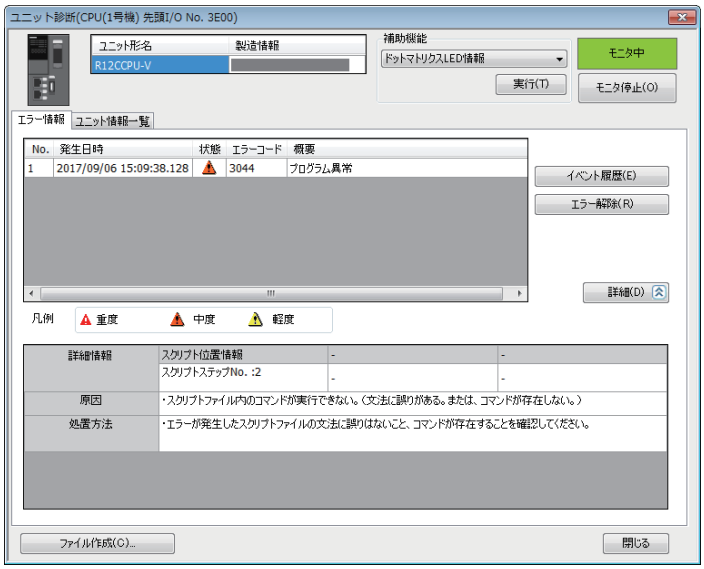
CW Configuratorを使って，発生したエラーの内容を確認できます。

エラーが発生した場合の確認方法

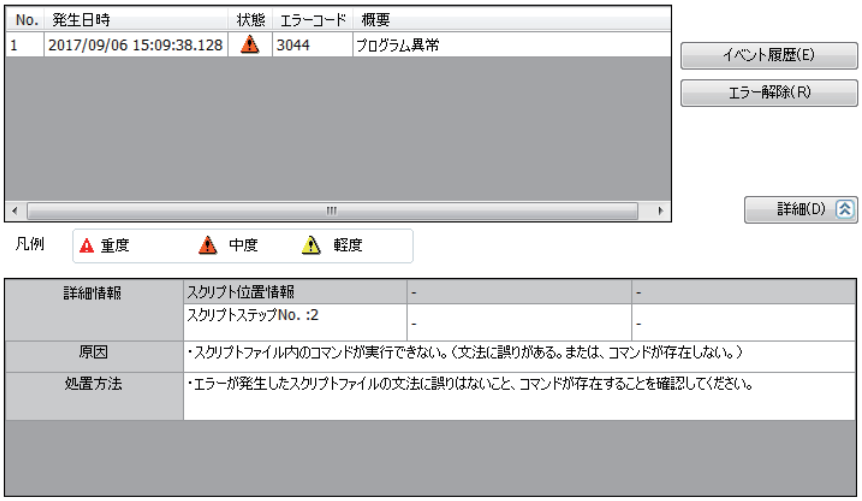
操作手順

1. CW Configuratorを起動します。
2. [診断]⇨[ユニット診断(CPU診断)]を選択します。

3. "ユニット診断"画面が表示されます。



4. 画面にエラーコードが表示されます。原因，処置方法など詳細な情報を確認できます。

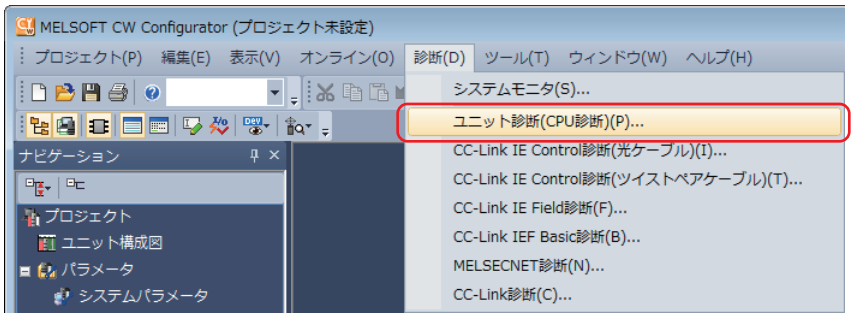


今までに発生したエラー履歴の確認方法

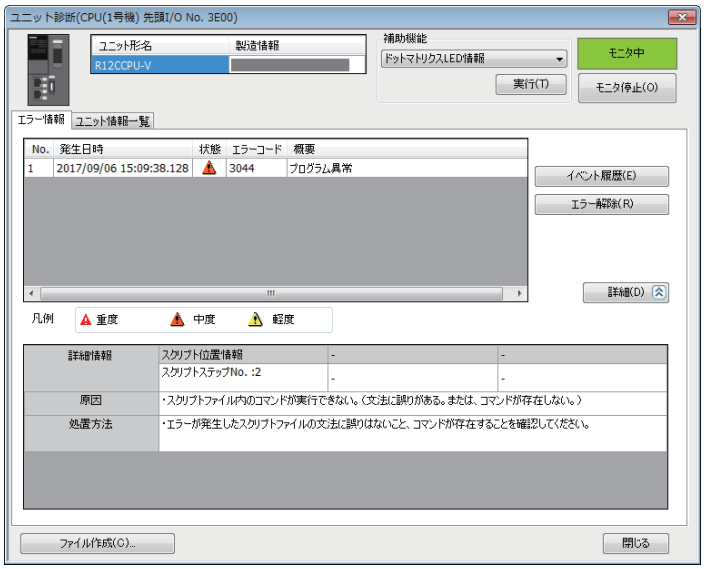
今までに発生したエラーの履歴と詳細情報を確認することができます。
いつ，どのようなエラーが発生したかを知ることができ，トラブルの解析に役立ちます。

操作手順

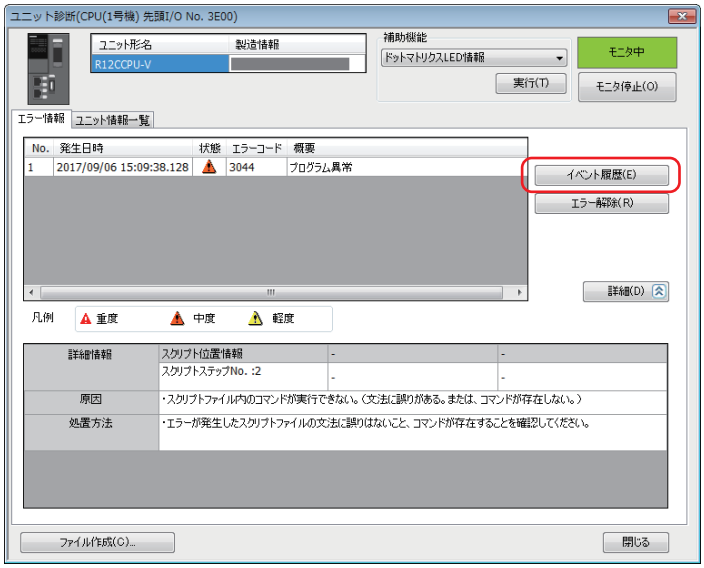
1. CW Configuratorを起動します。
2. [診断]⇨[ユニット診断(CPU診断)]を選択します。



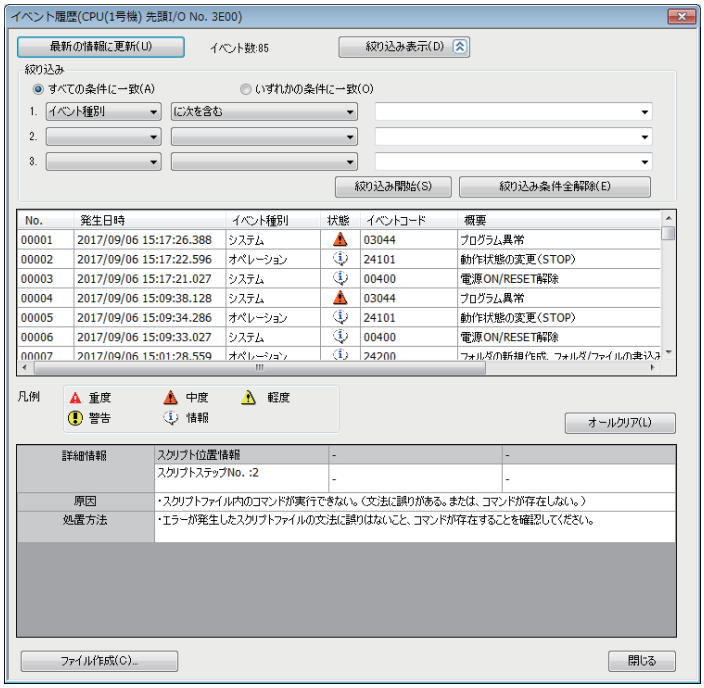
3. "ユニット診断"画面が表示されます。



4. [イベント履歴]ボタンをクリックします。

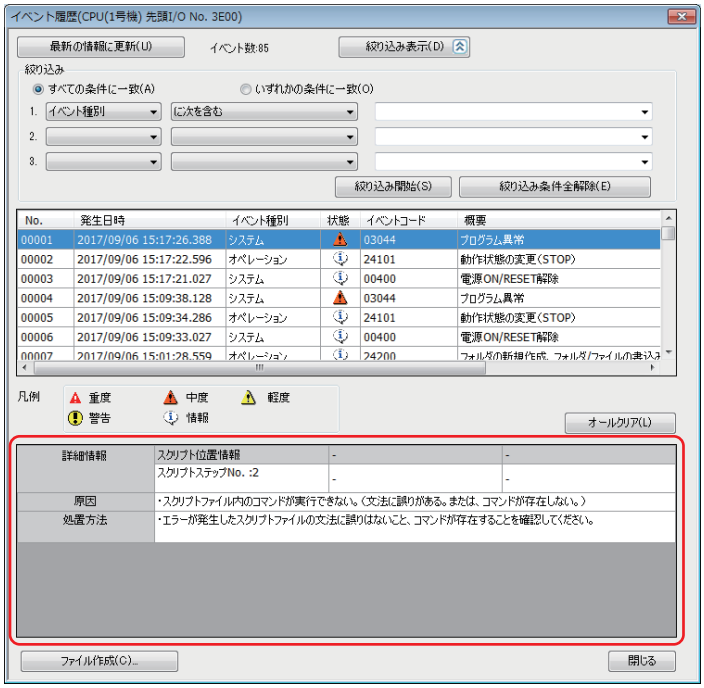


5. 今までに発生したエラーの履歴と詳細情報が表示されます。



6. 詳細を知りたいエラーの行をクリックします。

7. 詳細情報が表示されます。



3.2 ユニット状態のモニタと動作テストを行う

CW Configuratorを使って，ユニットの入出力やバッファメモリの状態が確認できます。
また，立上げ時や保守時などに，一時的な入出力の確認や動作テストができます。

ユニット状態のモニタ

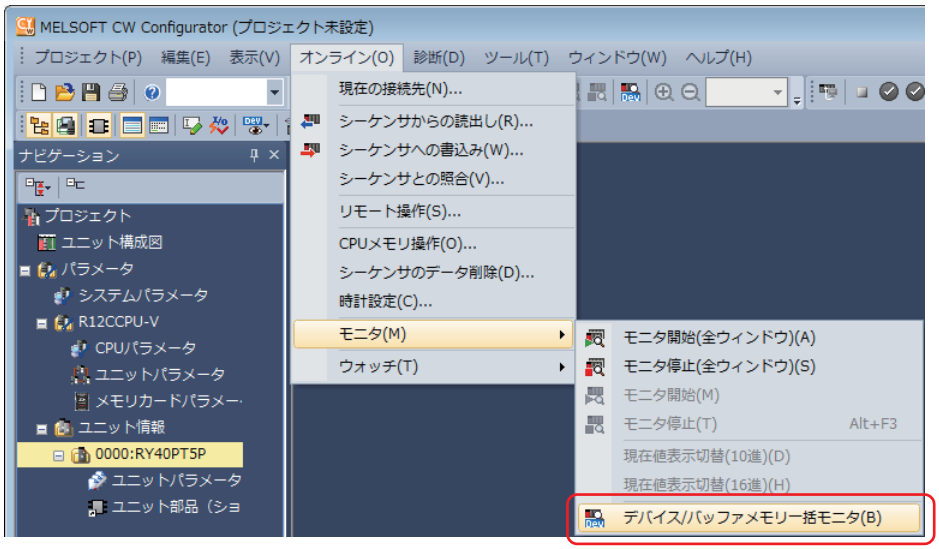
ユニットの入力(X)，出力(Y)，バッファメモリの状態をモニタできます。

Point

バッファメモリ: インテリジェント機能ユニット(A/D，D/A変換ユニットなど，入出力以外の機能を持つユニット)内部の記憶領域で，C言語コントローラユニットと受け渡しするデータ(設定値，モニタ値など)が格納されます。

操作手順

1. CW Configuratorを起動します。
2. [オンライン]⇨[モニタ]⇨[デバイス/バッファメモリ一括モニタ]を選択します。



3. "デバイス/バッファメモリ一括モニタ"画面が表示されるので，"デバイス名"に確認対象(先頭)を入力し，[モニタ開始]ボタンをクリックします。

ここでは，"D0"を指定した場合の例です。

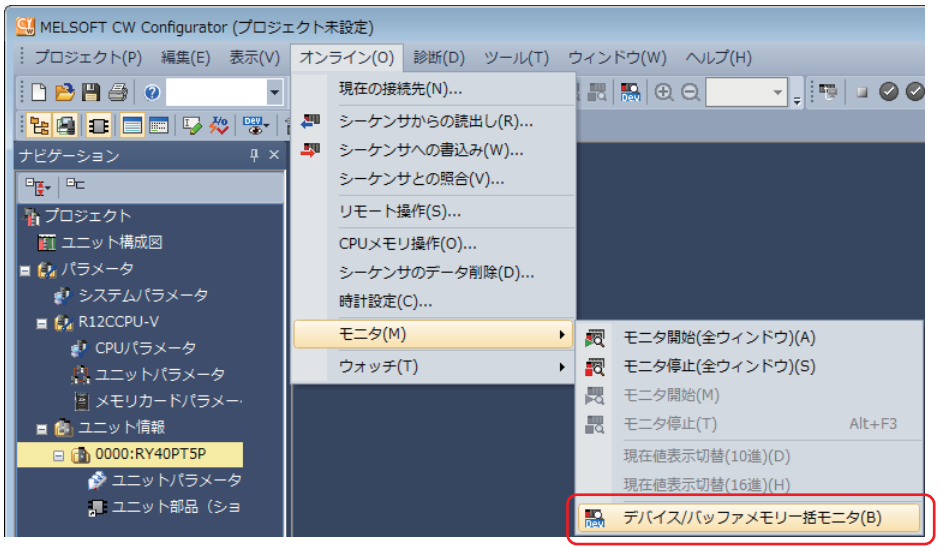
"D0"の状態が確認できます。

強制出力による動作テストの実施方法

出力(Y)の強制出力を行うことで，ユニットの動作テストを実施できます。
ここでは，出力(Y)の強制出力の手順を示します。

操作手順

1. CW Configuratorを起動します。
2. [オンライン]⇨[モニタ]⇨[デバイス/バッファメモリー括モニタ]を選択します。



3. "デバイス/バッファメモリー括モニタ"画面が表示されるので，"デバイス名"に確認対象(先頭)を入力し，[モニタ開始]ボタンをクリックします。

ここでは，"Y0"を指定した場合の例です。

4. Y0の0ビット目をダブルクリックします。

Y0の0ビット目「0」→「1」に変わり，出力(Y)の強制出力が実行されます。

Point

バッファメモリへの強制書込みによる動作テストも，同様の手順で行うことができます。

商標

VxWorksおよびWind Riverは，Wind River Systems, Inc.の登録商標または商標です。
Windowsは，米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
本文中における会社名，システム名，製品名などは，一般に各社の登録商標または商標です。
本文中で，商標記号(™，®)は明記していない場合があります。

ご採用に際してのご注意

この資料は，製品の代表的な特長機能を説明した資料です。使用上の制約事項，ユニットの組合わせによる制約事項などがすべて記載されているわけではありません。ご採用にあたりましては，必ず製品のマニュアルをお読み頂きますようお願い申し上げます。
当社の責に帰することができない理由から生じた損害，当社製品の故障に起因するお客様での機会損失，逸失利益，当社の予見の有無を問わず特別の事情から生じた損害，二次損害，事故補償，当社製品以外への損傷およびその他の業務に対する保証については，当社は責任を負いかねます。

⚠安全にお使いいただくために

- このガイドに記載された製品を正しくお使いいただくために，ご使用の前に必ずマニュアルをお読み下さい。
- この製品は一般工業等を対象とした汎用品として制作されたもので，人命にかかわるような状況下で使用される機器あるいはシステムに用いられることを目的として設計，製造されたものではありません。
- この製品を原子力用，電力用，航空宇宙用，医療用，乗用移動体用の機器あるいはシステムなど特殊用途への適用をご検討の際は，当社の営業担当窓口までご照会ください。
- この製品は厳重な品質管理体制の下に製造しておりますが，この製品の故障により重大な事故または損失の発生が予測される設備への適用に際しては，バックアップやフェールセーフ機能をシステム的に設置してください。

三菱電機 汎用 シーケンサ
C言語コントローラユニット クイックスタートガイド

三菱電機株式会社 〒100-8310 東京都千代田区丸の内2-7-3 (東京ビル)

お問い合わせは下記へどうぞ

本社機器営業部	〒110-0016	東京都台東区台東1-30-7 (秋葉原アイマークビル)	(03) 5812-1450
北海道支社	〒060-8693	札幌市中央区北二条西4-1 (北海道ビル)	(011) 212-3794
東北支社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア)	(022) 216-4546
関越支社	〒330-6034	さいたま市中央区新都心11-2 (明治安田生命さいたま新都心ビル)	(048) 600-5835
新潟支店	〒950-8504	新潟市中央区東大通1-4-1 (マルタケビル)	(025) 241-7227
神奈川支社	〒220-8118	横浜市西区みなとみらい2-2-1 (横浜ランドマークタワー)	(045) 224-2624
北陸支社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル)	(076) 233-5502
中部支社	〒450-6423	名古屋市中村区名駅3-28-12 (大名古屋ビルヂング)	(052) 565-3314
豊田支店	〒471-0034	豊田市小坂本町1-5-10 (矢作豊田ビル)	(0565) 34-4112
関西支社	〒530-8206	大阪市北区大深町4-20 (グランフロント大阪タワーA)	(06) 6486-4122
中国支社	〒730-8657	広島市中区中町7-32 (ニッセイ広島ビル)	(082) 248-5348
四国支社	〒760-8654	高松市寿町1-1-8 (日本生命高松駅前ビル)	(087) 825-0055
九州支社	〒810-8686	福岡市中央区天神2-12-1 (天神ビル)	(092) 721-2247

三菱電機 FA 検索

www.MitsubishiElectric.co.jp/fa

メンバー登録無料!

インターネットによる情報サービス「三菱電機FAサイト」

三菱電機FAサイトでは、製品や事例などの技術情報に加え、トレーニングスクール情報や各種お問い合わせ窓口をご提供しています。また、メンバー登録いただくとマニュアルやCADデータ等のダウンロード、eラーニングなどの各種サービスをご利用いただけます。

三菱電機FA機器電話

●電話技術相談窓口 受付時間※1 月曜～金曜 9:00～19:00、土曜・日曜・祝日 9:00～17:00

対象機種			電話番号	自動窓口案内選択番号※7
自動窓口案内			052-712-2444	-
エッジコンピューティング製品		産業用PC MELIPC Edgecross対応ソフトウェア(NC Machine Tool OptimizerなどのNC関連製品を除く)	052-712-2370※2	8
シーケンサ	MELSEC iQ-R/Q/Lシーケンサ (CPU内蔵Ethernet機能などネットワークを除く)		052-711-5111	2→2
	MELSEC iQ-F/FXシーケンサ全般		052-725-2271※3	2→1
	ネットワークユニット(CC-Linkファミリー/MELSECNET/Ethernet/シリアル通信)		052-712-2578	2→3
	MELSOFTシーケンサエンジニアリングソフトウェア	MELSOFT GXシリーズ (MELSEC iQ-R/Q/L/QnAS/AnS)	052-711-0037	2→2
	MELSOFT統合エンジニアリング環境	MELSOFT Navigator/ MELSOFT Update Manager	052-799-3591※2	2→6
	iQ Sensor Solution			
	MELSOFT通信支援ソフトウェアツール	MELSOFT MXシリーズ	052-712-2370※2	2→4
	MELSECパソコンボード	Q80BDシリーズなど		
	C言語コントローラ/C言語インテリジェント機能ユニット			
	MESインタフェースユニット/高速データロガーユニット/高速データコミュニケーションユニット/OPC UAサーバユニット		052-799-3592※2	2→5
	システムレコーダ			
	MELSEC計装/iQ-R/Q二重化	プロセスCPU/二重化機能 SIL2プロセスCPU (MELSEC iQ-Rシリーズ)	052-712-2830※2※3	2→7
		プロセスCPU/二重化CPU (MELSEC-Qシリーズ)		
		MELSOFT PXシリーズ		
	MELSEC Safety	安全シーケンサ (MELSEC iQ-R/QSシリーズ)	052-712-3079※2※3	2→8
		安全コントローラ (MELSEC-WSシリーズ)		
	電力計測ユニット/絶縁監視ユニット	QEシリーズ/REシリーズ	052-719-4557※2※3	2→9
FAセンサ MELSENSOR		レーザ変位センサ	052-799-9495※2	6
		ビジョンセンサ		
		コードリーダ		

対象機種		電話番号	自動窓口案内選択番号※7
表示器 GOT	GOT2000/1000シリーズ	052-712-2417	4→1
	MELSOFT GTシリーズ		4→2
SCADA GENESIS64™		052-712-2962※2※6	-
サーボ/位置決めユニット/モーションユニット/シンプルモーションユニット/モーションコントローラ/センシングユニット/組込み型サーボシステムコントローラ	MELSERVOシリーズ	052-712-6607	1→2
	位置決めユニット (MELSEC iQ-R/Q/Lシリーズ)		1→2
	モーションユニット (MELSEC iQ-R/iQ-Fシリーズ)		1→1
	モーションソフトウェア		1→1
	シンプルモーションユニット (MELSEC iQ-R/iQ-F/Q/Lシリーズ)		1→2
	モーションCPU (MELSEC iQ-R/Qシリーズ)		1→1
	センシングユニット (MR-MTシリーズ)		1→2
	シンプルモーションボード/ポジションボード		1→2
	MELSOFT MTシリーズ/MRシリーズ/EMシリーズ		1→2
センサレスサーボ	FR-E700EX/MM-GKR	052-722-2182	3
インバータ	FREQROLシリーズ	052-722-2182	
三相モータ	三相モータ225フレーム以下	0536-25-0900※2※4	-
産業用ロボット	MELFAシリーズ	052-721-0100	5
電磁クラッチ・ブレーキ/テンションコントローラ		052-712-5430※5	-
データ収集アナライザ	MELQIC IU1/IU2シリーズ	052-712-5440※5	-
低圧開閉器	MS-Tシリーズ/MS-Nシリーズ	052-719-4170	7→2
	US-Nシリーズ		
低圧遮断器	ノーヒューズ遮断器/漏電遮断器/MDUブレーカ/気中遮断器 (ACB) など	052-719-4559	7→1
電力管理用計器	電力量計/計器用変成器/指示電気計器/管理用計器/タイムスイッチ	052-719-4556	7→3
省エネ支援機器	EcoServer/E-Energy/検針システム/エネルギー計測ユニット/ B/NETなど	052-719-4557※2※3	7→4
小容量UPS (5kVA以下)	FW-Sシリーズ/FW-Vシリーズ/FW-Aシリーズ/FW-Fシリーズ	052-799-9489※2※6	7→5

お問い合わせの際には、今一度電話番号をお確かめの上、お掛け間違いのないようお願いいたします。
※1：春季・夏季・年末年始の休日を除く ※2：土曜・日曜・祝日を除く ※3：金曜は17:00まで ※4：月曜～木曜の9:00～17:00と金曜の9:00～16:30
※5：受付時間9:00～17:00 (土曜・日曜・祝日・当社休日を除く) ※6：月曜～金曜の9:00～17:00
※7：選択番号の入力は、自動窓口案内冒頭のお客様相談内容に関する代理店、商社への提供可否確認の回答後にお願いいたします。

L(名)08509-D 1903(KWIX)

2019年3月作成