

三菱電機汎用シーケンサ C言語インテリジェント機能ユニット クイックスタートガイド

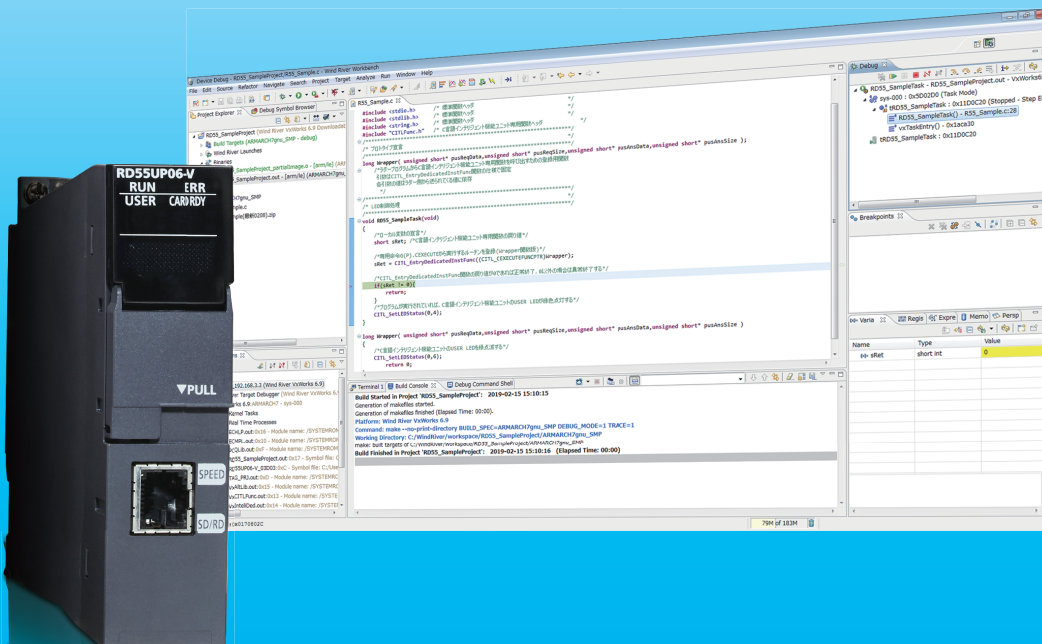
はじめよう、C言語インテリジェント機能ユニット!

MELSEC iQ-R
series

e-Factory



はじめに	1
C言語インテリジェント機能ユニットを使ってみよう	2
作業を行う前に	2.1
システムを構築する	2.2
ユニットの設定をする	2.3
プログラミングする前に知っておきたいこと	2.4
プログラミングする	2.5
動作を確認する	2.6
よく使う機能	3
エラーを確認する	3.1



目次

関連マニュアル.....	2
第1章 はじめに	3
第2章 C言語インテリジェント機能ユニットを使ってみよう	5
2.1 作業を行う前に.....	5
2.2 システムを構築する.....	6
システム構成例.....	6
ユニットを装着する.....	7
ユニットの配線を行う.....	9
電源が正常か確認する.....	9
2.3 ユニットの設定をする.....	10
システム構成を設定する.....	10
C言語インテリジェント機能ユニットを初期化する.....	13
パラメータおよびラダープログラムを設定する.....	15
2.4 プログラミングする前に知っておきたいこと.....	17
C言語インテリジェント機能ユニット専用関数とは.....	17
今回使用するC言語インテリジェント機能ユニット専用関数.....	18
2.5 プログラミングする.....	19
今回作成するプログラムと制御内容.....	19
プロジェクトを作成する.....	20
ユーザプログラムを準備する.....	30
2.6 動作を確認する.....	48
ラダープログラムの実行.....	48
LEDで動作を確認する.....	49
第3章 よく使う機能	50
3.1 エラーを確認する.....	50
エラーが発生した場合の確認方法.....	50
今までに発生したエラー履歴の確認方法.....	52
商標.....	56
ご採用に際してのご注意.....	56
安全にお使いいただくために.....	56

関連マニュアル

本クイックスタートガイドでは、C言語インテリジェント機能ユニットの基本的な導入手順を紹介しています。C言語インテリジェント機能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言語インテリジェント機能ユニット ユーザーズマニュアル(スタートアップ編) [SH-081563]	C言語インテリジェント機能ユニットの仕様、運転までの手順、配線、動作例について記載しています。	製本物	1,500円
		e-Manual PDF	—
MELSEC iQ-R C言語インテリジェント機能ユニット ユーザーズマニュアル(応用編) [SH-081564]	C言語インテリジェント機能ユニットの機能、入出力信号、バッファメモリ、パラメータ設定、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言語インテリジェント機能ユニット プログラミングマニュアル [SH-081565]	C言語インテリジェント機能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	—
GX Works3 オペレーティングマニュアル [SH-081214]	GX Works3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	—

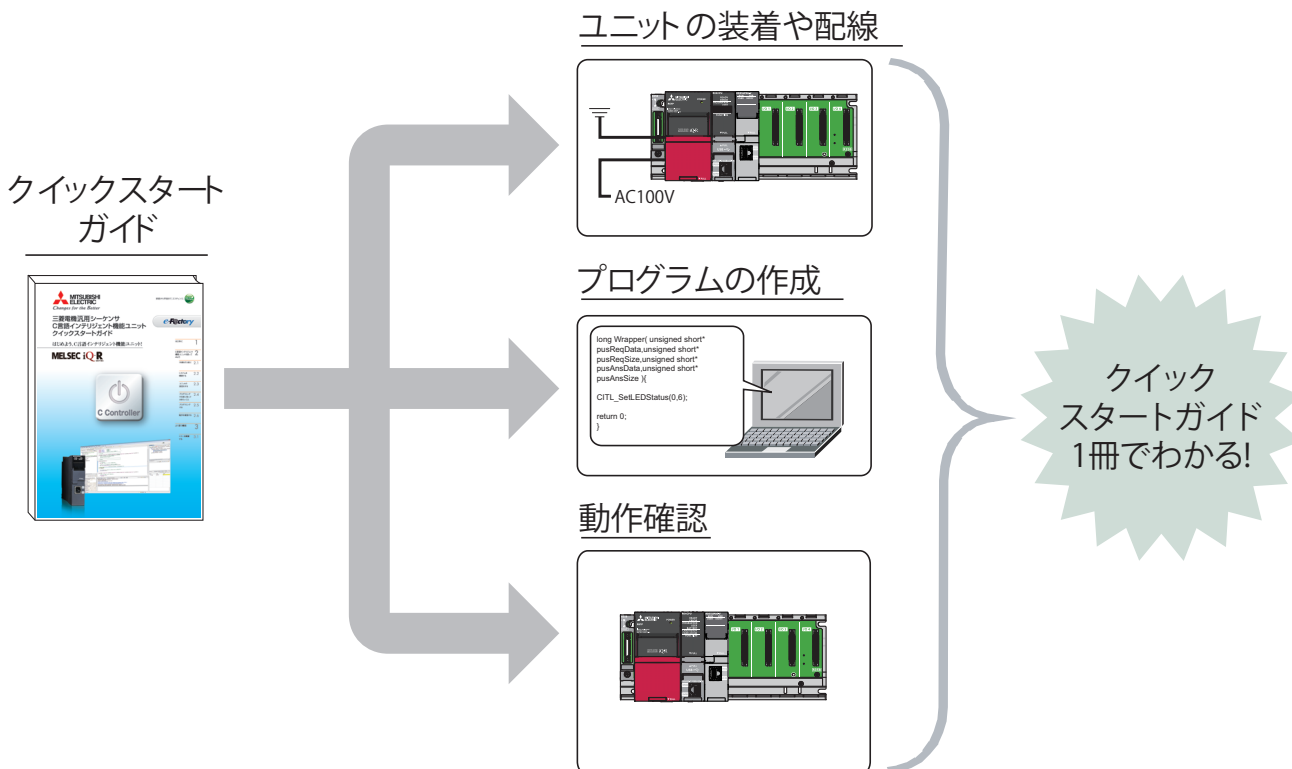
Point

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言語インテリジェント機能ユニットRD55UP06-V(以下C言語インテリジェント機能ユニット)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言語インテリジェント機能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言語インテリジェント機能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言語インテリジェント機能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言語インテリジェント機能ユニットユーザーズマニュアル\(スタートアップ編\)](#))
- CPU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 CPU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。

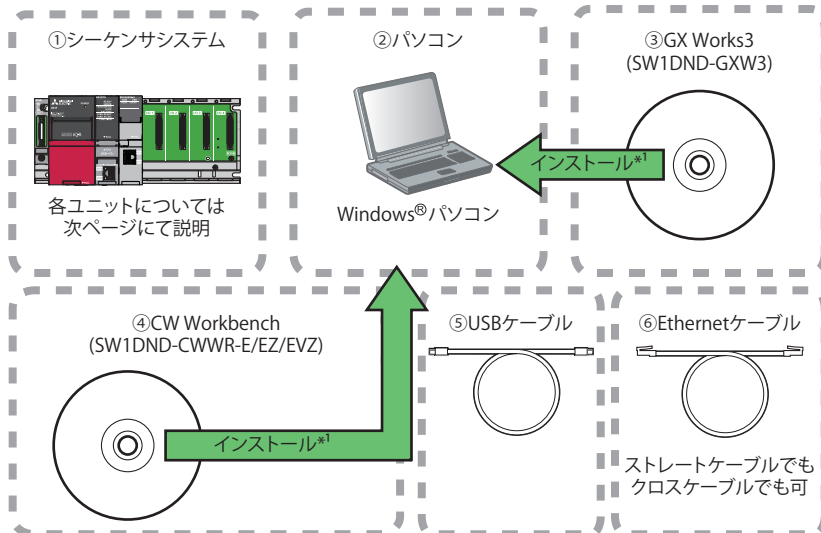
📖 関連マニュアル

MEMO

2 C言語インテリジェント機能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GX Works3、CW Workbenchをインストールしておいてください。

Point

GX Works3のインストールについては、下記を参照してください。

📖GX Works3 インストール手順書

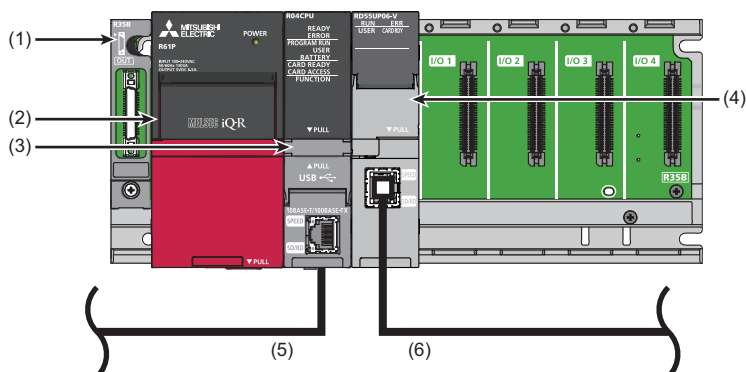
CW Workbenchのインストールについては、下記を参照してください。

📖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、CPUユニット、およびC言語インテリジェント機能ユニットを装着するユニットです。
(2)	電源ユニット	R61P	CPUユニットおよびC言語インテリジェント機能ユニットに電気を供給するユニットです。
(3)	CPUユニット	R04CPU	C言語インテリジェント機能ユニットを管理するCPUユニットです。
(4)	C言語インテリジェント機能ユニット	RD55UP06-V	ユーザが作成したC言語プログラムを制御するインテリジェント機能ユニットです。
(5)	接続ケーブル (USBケーブル)	「USB Aタイプ – USB miniBタイプ」または「USB Bタイプ – USB miniBタイプ」	GX Works3をインストールしたパソコンとCPUユニットを接続します。
(6)	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 WorkbenchをインストールしたパソコンとC言語インテリジェント機能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。

Point

C言語インテリジェント機能ユニットは、ユニット固定用フックなしの手順に従って取り付けてください。
(8ページユニット固定用フックなし)

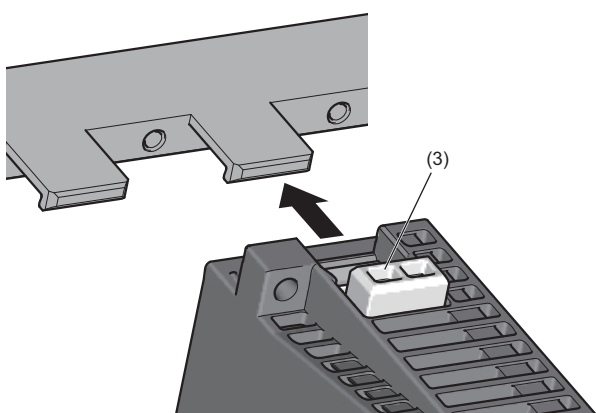
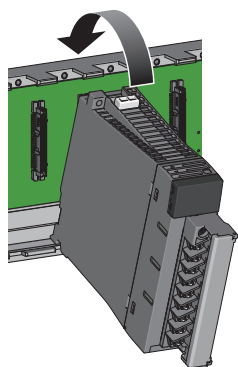
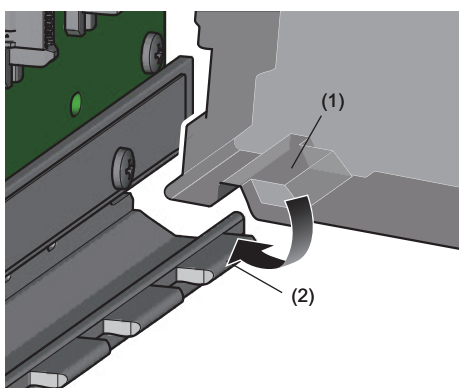
2

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。

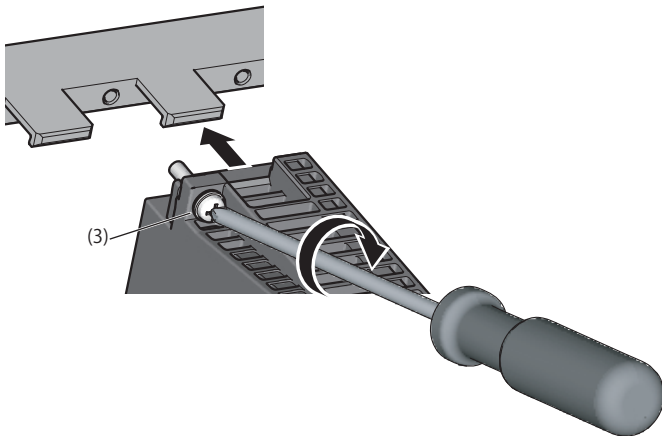
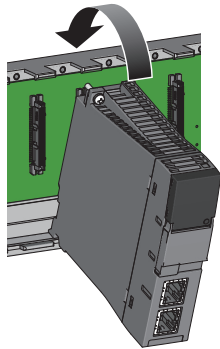
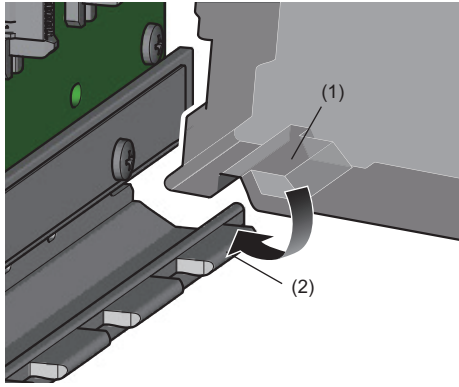
取付け手順

■ユニット固定用フックあり



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、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。

■ユニット固定用フックなし



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、ユニット固定ネジ(3)を締め付けて、ベースユニットに装着します。(締め付けトルク: $0.36\sim 0.48\text{N}\cdot\text{m}$)
4. 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

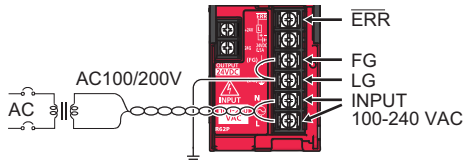
電源ユニットの配線を行います。

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

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。



注意事項

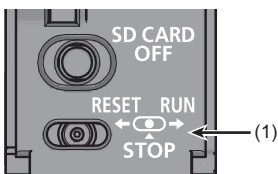
ユニットの配線を行うときは、必ず電源を遮断してください。

電源が正常か確認する

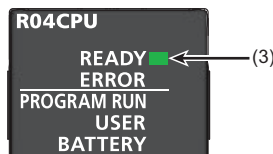
システムの設置、ユニットの取付け、配線をした後に、電源が正常に入ることを確認します。

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. CPUユニットの状態をSTOPにします。
CPUユニット前面のLEDカバーを開き、RUN/STOP/RESETスイッチ(1)を「STOP」の位置にします。



3. 電源を投入します。
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
 - ・ CPUユニット: READY LED(3)が緑色点灯
 - ・ C言語インテリジェント機能ユニット: RUN LED(4)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言語インテリジェント機能ユニットの初期化と、CPUユニットへのパラメータおよびプログラムの書き込みを行います。

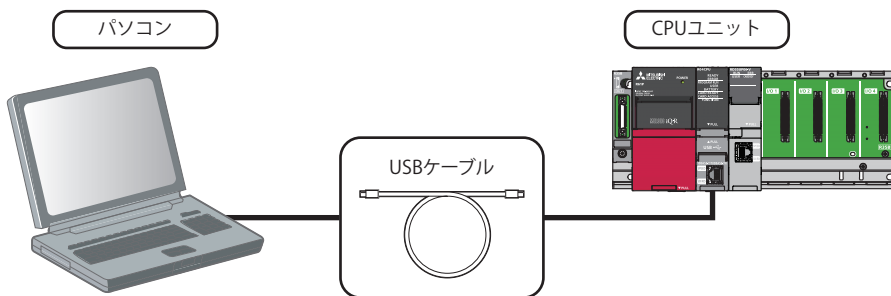
Point

パラメータ:シーケンサ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。GX Works3を使用して、シーケンサ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。

システム構成を設定する

CPUユニットとパソコンを接続する

CPUユニットとパソコンを、USBケーブルで接続します。



Point

パソコンとCPUユニットの接続の詳細については、下記を参照してください。

📖 GX Works3 オペレーティングマニュアル

GX Works3を起動し、プロジェクトを作成する

本クイックスタートガイドでは、専用のサンプルプロジェクトを使用します。

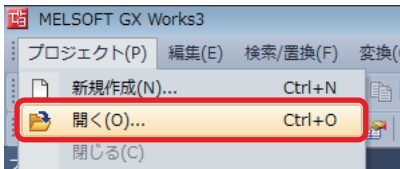
Point

本クイックスタートガイド専用のサンプルプロジェクトは、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

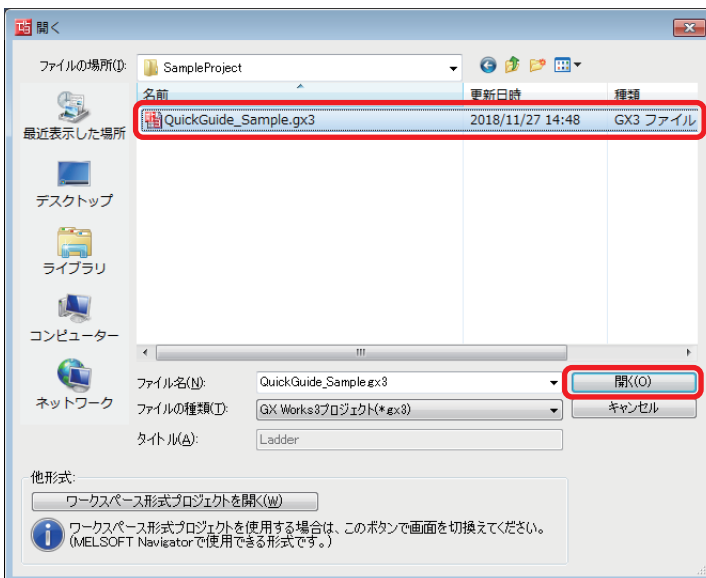
2

操作手順

1. [スタート]⇒[すべてのプログラム]⇒[MELSOFT]⇒[GX Works3]⇒[GX Works3]を選択します。
2. [プロジェクト]⇒[開く]を選択します。



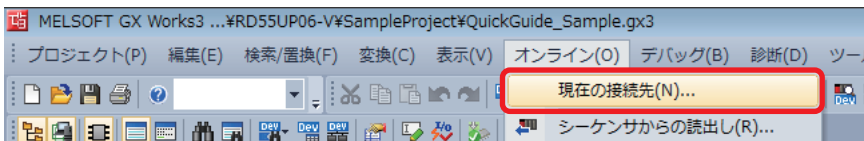
3. サンプルプロジェクトを選択し、[開く]ボタンをクリックします。



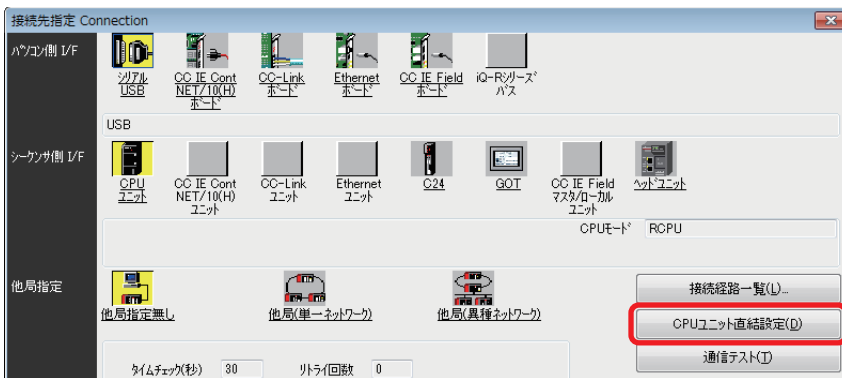
CPUユニットと通信する

操作手順

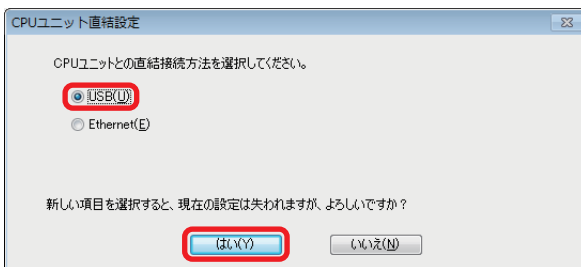
1. GX Works3のメニューから、[オンライン]⇒[現在の接続先]を選択します。



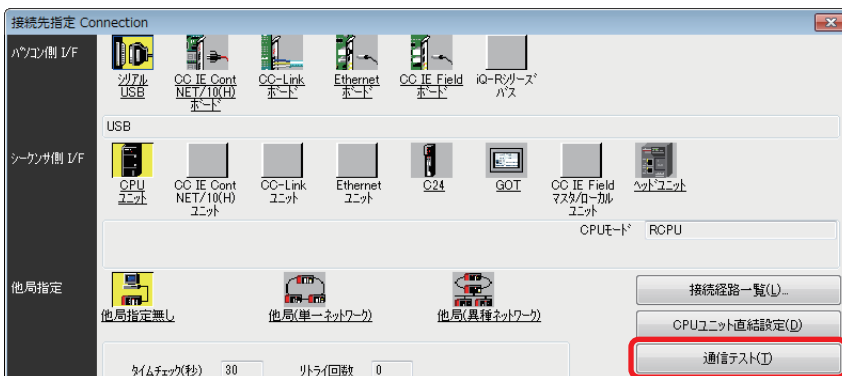
2. "接続先指定"画面で、[CPUユニット直結設定]ボタンをクリックします。



3. "USB"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"R04CPU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。

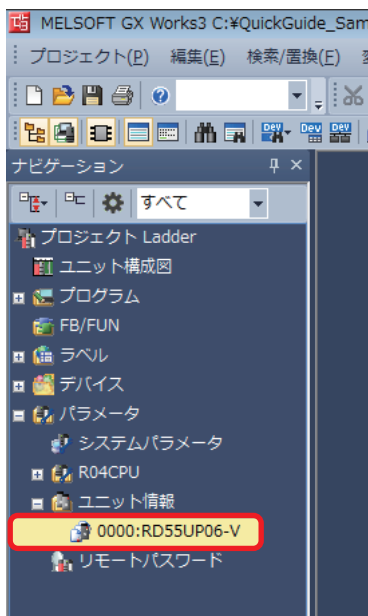


C言語インテリジェント機能ユニットを初期化する

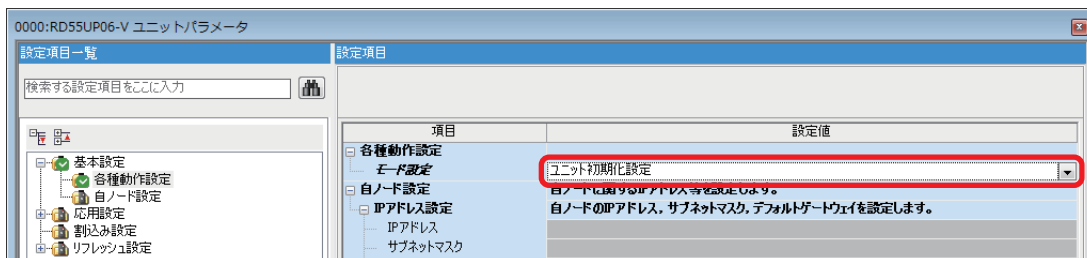
初期化する前に、CPUユニットのRUN/STOP/RESETスイッチがSTOPになっていることを確認してください。

操作手順

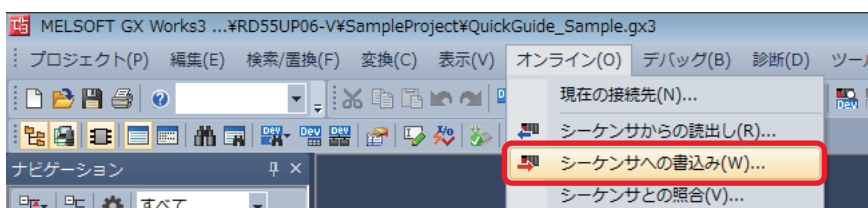
1. C言語インテリジェント機能ユニットのユニットパラメータを開きます。
[ナビゲーションウィンドウ]⇒[パラメータ]⇒[ユニット情報]⇒対象ユニットをダブルクリックします。



2. C言語インテリジェント機能ユニットのユニットパラメータ内の[基本設定]⇒[各種動作設定]⇒[モード設定]を、"ユニット初期化設定"に変更します。

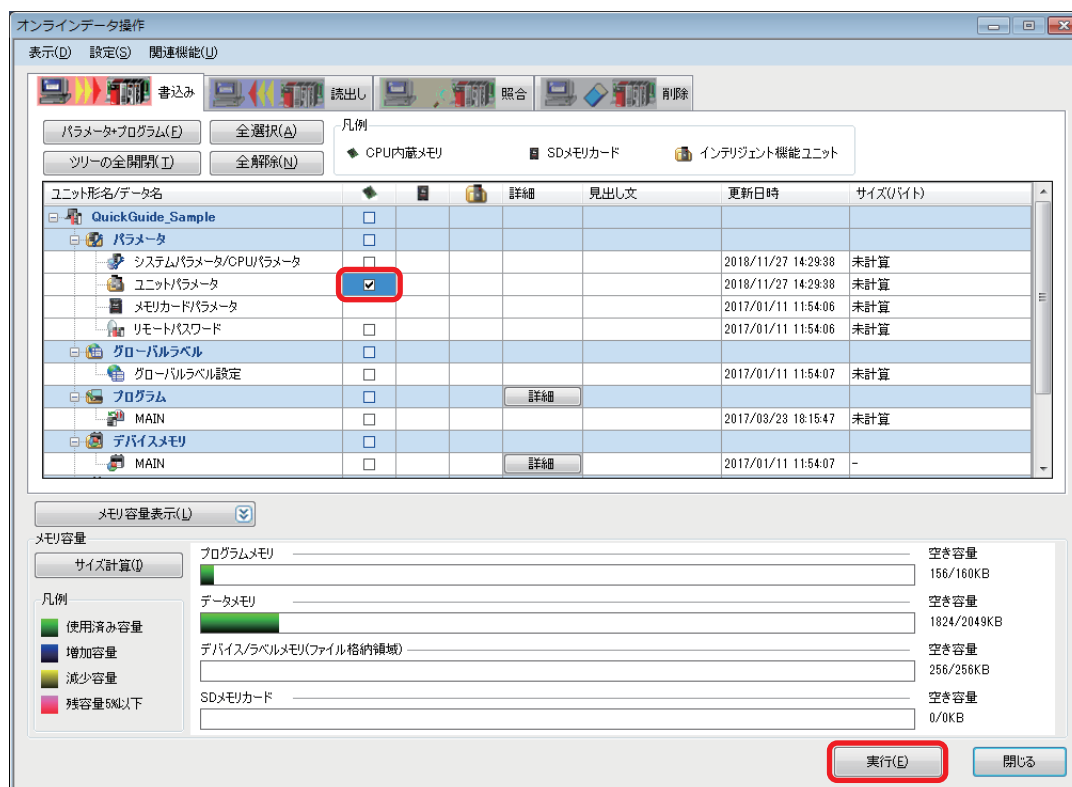


3. [オンライン]⇒[シーケンサへの書き込み]を選択します。



"オンラインデータ操作"画面が表示されます。

4. "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. CPUユニットをリセットします。

CPUユニットのリセット後、自動的にC言語インテリジェント機能ユニットの初期化が実行されます。初期化実行時のC言語インテリジェント機能ユニットのLED状態を示します。

状態		RUN LED状態	ERR LED状態
初期化中		点滅(低速)	消灯
初期化完了	正常完了	点灯	消灯
	異常完了	点灯	点灯

異常完了した場合は、下記を参照してください。

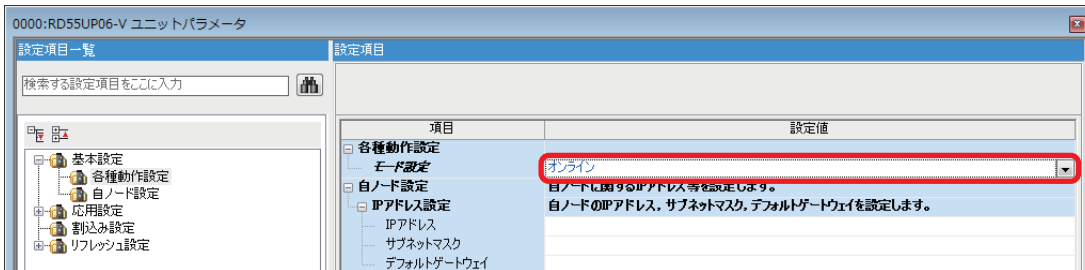
📖 MELSEC iQ-R C言語インテリジェント機能ユニットユーザーズマニュアル(応用編)

パラメータおよびラダープログラムを設定する

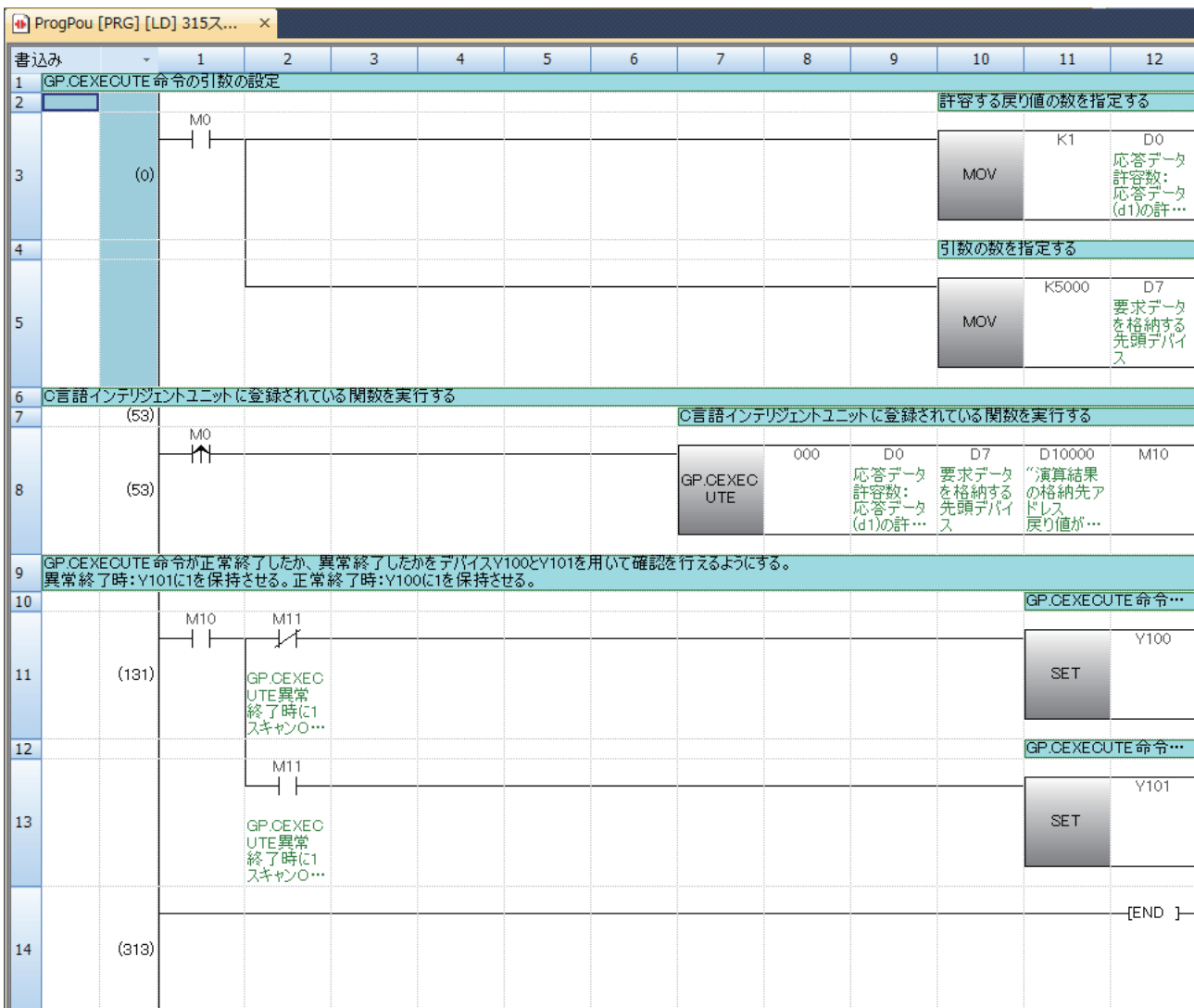
パラメータおよびラダープログラムを書込む前に、CPUユニットのRUN/STOP/RESETスイッチがSTOPになっていることを確認してください。

操作手順

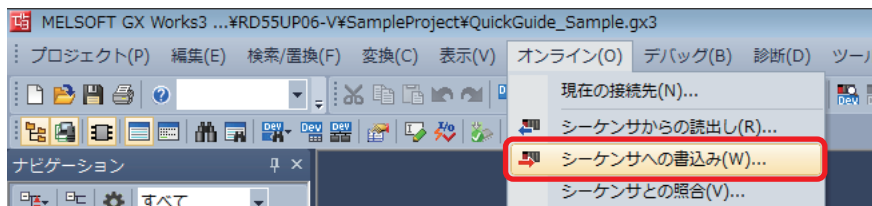
1. C言語インテリジェント機能ユニットのユニットパラメータ内の[基本設定]⇒[各種動作設定]⇒[モード設定]を、"オンライン"に変更します。



2. ラダープログラムを作成します。
サンプルプロジェクトには、下記のラダープログラムが準備されています。

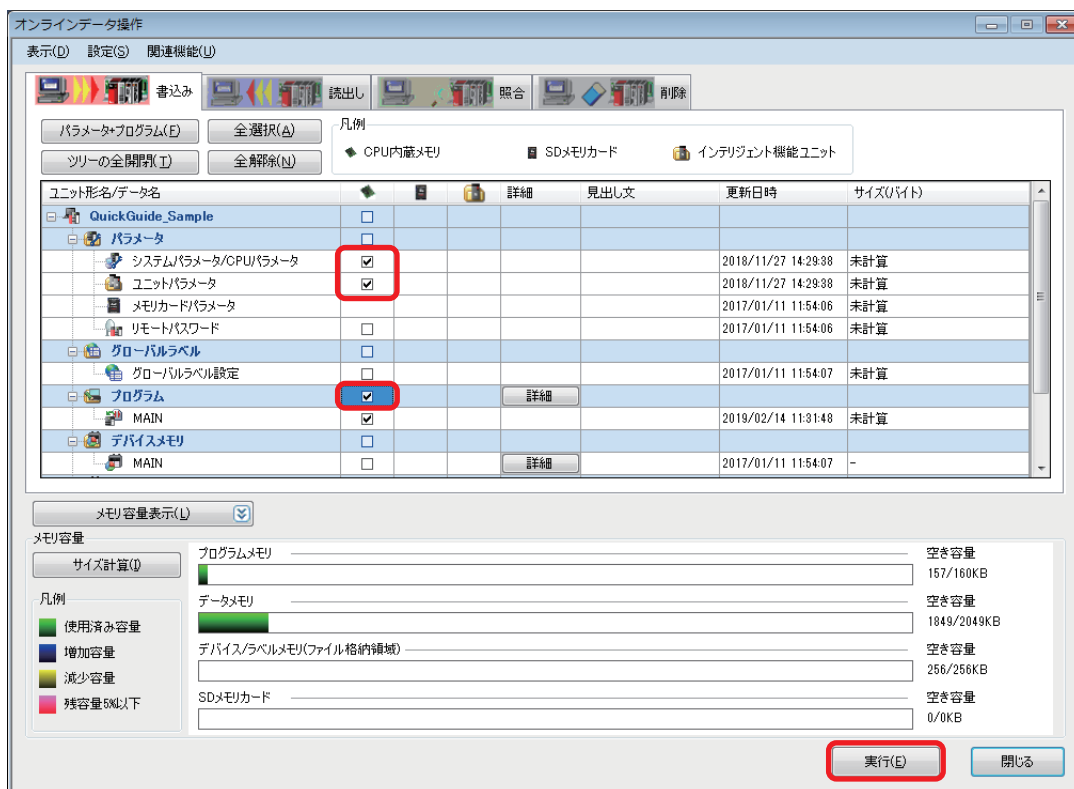


3. [オンライン]⇒[シーケンサへの書込み]を選択します。



"オンラインデータ操作"画面が表示されます。

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", および"プログラム"にチェックを入れ, [実行]ボタンをクリックします。



5. CPUユニットをリセットします。

2.4 プログラミングする前に知っておきたいこと

C言語インテリジェント機能ユニット専用関数とは

C言語インテリジェント機能ユニット専用関数とは、C言語インテリジェント機能ユニットの「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。

Point

入出力アクセスで使用する関数は、C言語インテリジェント機能ユニットの状態読出しや割込み機能などで使用します。

■ビット単位アクセス:1点分の情報を扱う関数

ビット単位アクセス関数の具体例

関数名	内容
CITL_X_In_Bit	入力信号(X)をビット単位(1点)で読み出します。
CITL_X_Out_Bit	入力信号(X)にビット単位(1点)で書き込みます。
CITL_Y_In_Bit	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
CITL_X_In_Word	入力信号(X)をワード単位(16点)で読み出します。
CITL_X_Out_Word	入力信号(X)にワード単位(16点)で書き込みます。
CITL_Y_In_Word	出力信号(Y)をワード単位(16点)で読み出します。

実行ルーチン登録


関数名	内容
CITL_EntryDedicatedInstFunc	専用命令(G(P).CEXECUTE)から実行するルーチンを登録します。

ユーザLED制御

関数名	内容
CITL_SetLEDStatus	C言語インテリジェント機能ユニットのLED状態を設定します。

Point

ここでは、最も基本的なC言語インテリジェント機能ユニット専用関数を紹介しています。上記のほかにも、制御を行うための便利なC言語インテリジェント機能ユニット専用関数やMELSEC iQ-Rシリーズ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言語インテリジェント機能ユニットプログラミングマニュアル

今回使用するC言語インテリジェント機能ユニット専用関数

今回作成するプログラムには、基本的なC言語インテリジェント機能ユニット専用関数(実行ルーチン登録およびユーザLED制御)を使用します。

CITL_EntryDedicatedInstFunc

■形式

short CITL_EntryDedicatedInstFunc (CITL_CEXECUTEFUNCPTR pCEXECUTEFuncPtr)

■引数

引数	名称	説明	IN/OUT
pCEXECUTEFuncPtr	登録ルーチン	登録するルーチンを指定します。 (NULLを指定すると、ルーチンの登録を解除します。)	IN

登録ルーチン(pCEXECUTEFuncPtr)のデータ型は、ヘッダファイル(CITLFunc.h)で下記のとおり定義されています。

- void (*CITL_CEXECUTEFUNCPTR) (unsigned short* pusReqData, unsigned short* pusReqSize, unsigned short* pusAnsData, unsigned short* pusAnsSize)

引数	名称	説明	IN/OUT
pusReqData	要求データ	専用命令で指定された要求データを受け取ります。	IN
pusReqSize	要求データサイズ	専用命令で指定された要求データのサイズを受け取ります。	IN
pusAnsData	応答データ	専用命令に応答データを返します。	OUT
pusAnsSize	応答データサイズ	専用命令に応答データのサイズを返します。	OUT

CITL_SetLEDStatus

■形式

short CITL_SetLEDStatus (long lLed, unsigned short usLedInfo)

■引数

引数	名称	説明	IN/OUT
lLed	対象LED	対象LED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: USER LED ・ その他: リザーブ	IN
usLedInfo	LED状態情報	LED状態を指定します。	IN

LED状態情報(usLedInfo)の指定方法を示します。

格納情報	LED状態
0	消灯
1	点灯(赤色)
2	低速点滅(赤色)
3	高速点滅(赤色)
4	点灯(緑色)
5	低速点滅(緑色)
6	高速点滅(緑色)

2.5 プログラミングする

C言語インテリジェント機能ユニット前面のUSER 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言語インテリジェント機能ユニット専用関数(CITL_EntryDedicatedInstFunc)で登録したルーチン(ユーザLED制御)が、ラダープログラム上の専用命令(G(P).CEXECUTE)で実行され、C言語インテリジェント機能ユニットのUSER LEDが緑色点滅します。

ソースコード

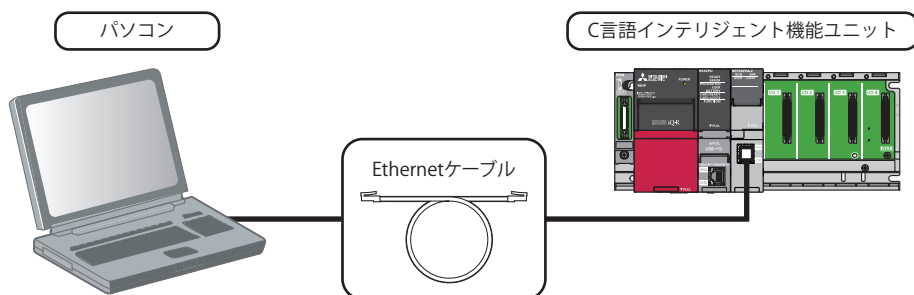
ソースコードを下記に示します。

```
/******  
/* 関数ヘッダ */  
/******  
#include <stdio.h> /* 標準関数ヘッダ */  
#include <stdlib.h> /* 標準関数ヘッダ */  
#include <string.h> /* 標準関数ヘッダ */  
#include "CITLFunc.h" /* C言語インテリジェント機能ユニット専用関数ヘッダ */  
/******  
/* プロトタイプ宣言 */  
/******  
long Wrapper( unsigned short* pusReqData,unsigned short* pusReqSize,unsigned short* pusAnsData,unsigned short* pusAnsSize );  
/*ラダープログラムからC言語インテリジェント機能ユニット専用関数を呼び出すための登録用関数  
引数はCITL_EntryDedicatedInstFunc関数の仕様で固定  
各引数の値はラダー側から送られてくる値に依存  
*/  
/******  
/* LED制御処理 */  
/******  
void RD55_SampleTask(void)  
{  
    /* ローカル変数の宣言 */  
    short sRet; /* C言語インテリジェント機能ユニット専用関数の戻り値 */  
  
    /* 専用命令G(P).CEXECUTEから実行するルーチンを登録(Wrapper関数版) */  
    sRet = CITL_EntryDedicatedInstFunc((CITL_CEXECUTEFUNCPTR)Wrapper);  
  
    /* CITL_EntryDedicatedInstFunc関数の戻り値が0であれば正常終了、0以外の場合は異常終了する */  
    if(sRet != 0){  
        return;  
    }  
    /* プログラムが実行されていれば、C言語インテリジェント機能ユニットのUSER LEDが緑色点灯する */  
    CITL_SetLEDStatus(0,4);  
}  
  
long Wrapper( unsigned short* pusReqData,unsigned short* pusReqSize,unsigned short* pusAnsData,unsigned short* pusAnsSize )  
{  
    /* C言語インテリジェント機能ユニットのUSER LEDを緑色点滅する */  
    CITL_SetLEDStatus(0,6);  
    return 0;  
}
```

プロジェクトを作成する

C言語インテリジェント機能ユニットとパソコンを接続する

C言語インテリジェント機能ユニットとパソコンを、Ethernetケーブルで接続します。



注意事項

接続の際には、C言語インテリジェント機能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言語インテリジェント機能ユニットの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

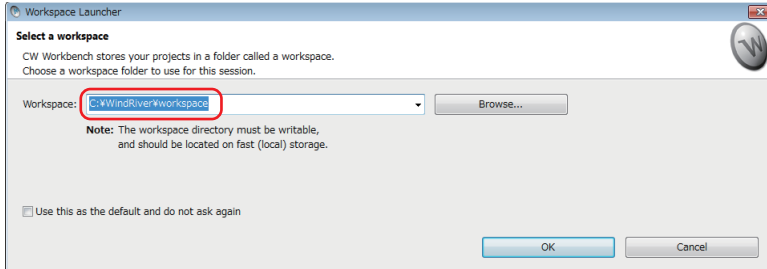
IPアドレスの変更については、下記を参照してください。

📖 MELSEC iQ-R C言語インテリジェント機能ユニットユーザーズマニュアル(応用編)

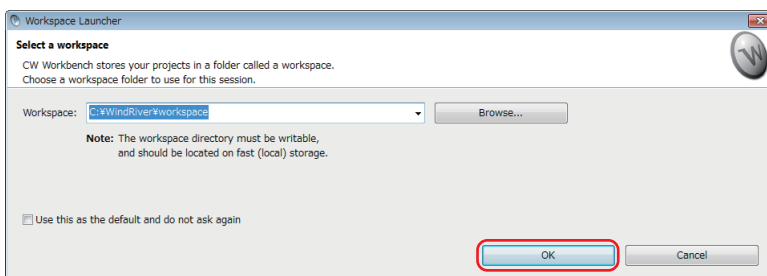
CW Workbenchを起動する

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。

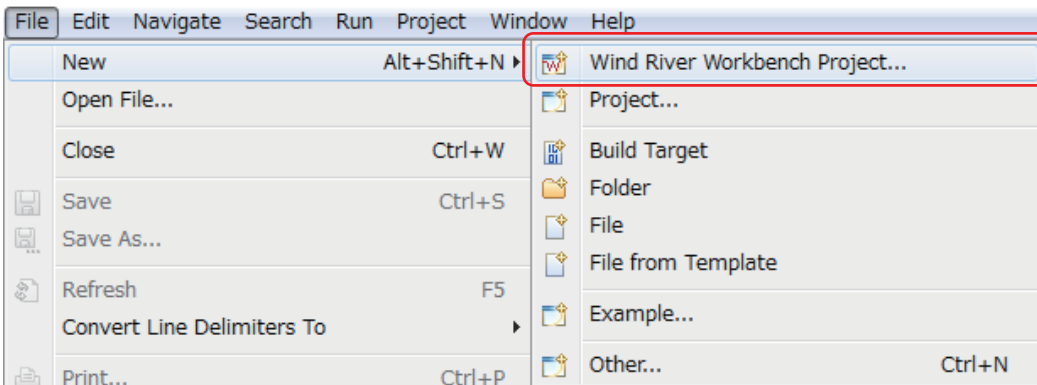
Point

- 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。

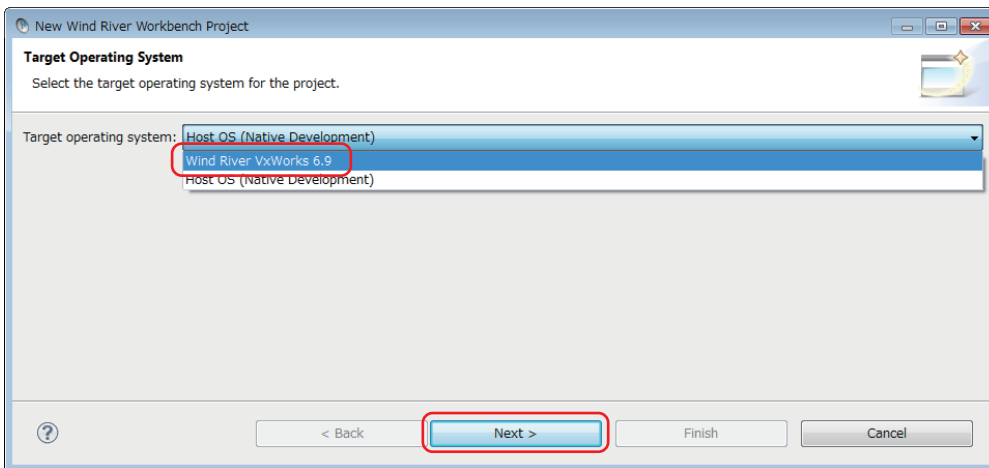
新規プロジェクトを作成する

操作手順

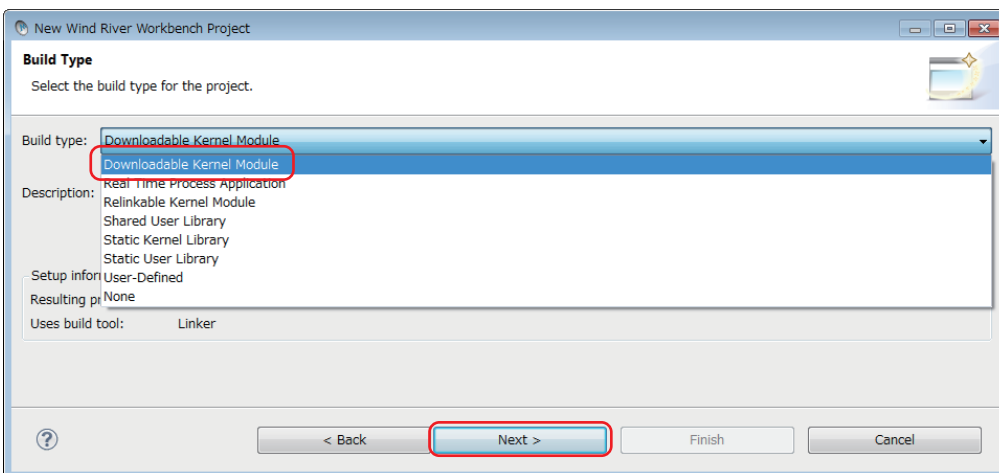
1. [File]⇒[New]⇒[Wind River Workbench Project]を選択します。



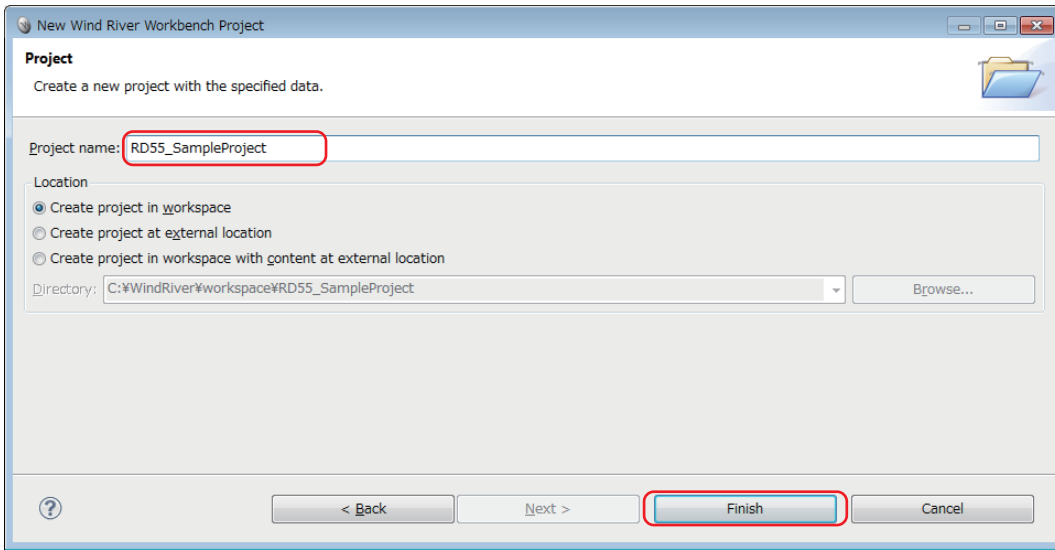
2. "Wind River VxWorks 6.9"を選択し, [Next]ボタンをクリックします。



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"RD55_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言語インテリジェント機能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。

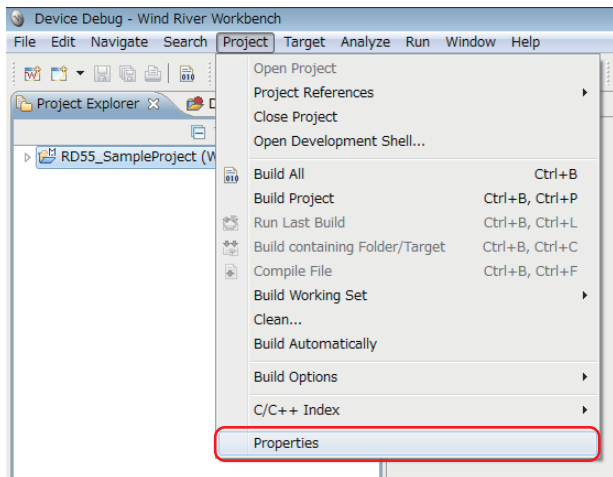
Point

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。

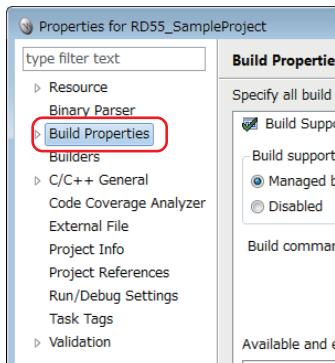
■使用するプロセッサを設定

操作手順

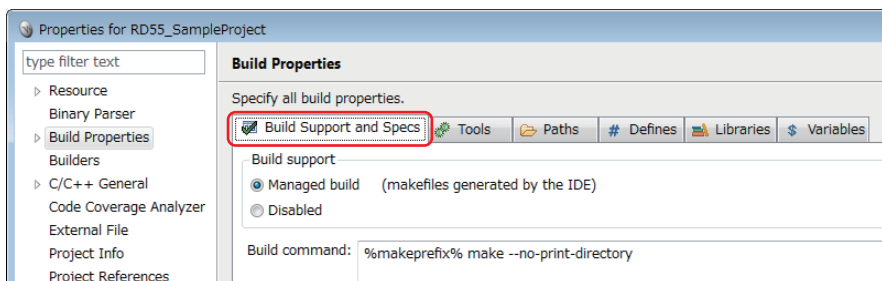
1. "Project Explorer"ウィンドウから作成したプロジェクトを選択し、[Project]⇒[Properties]を選択します。



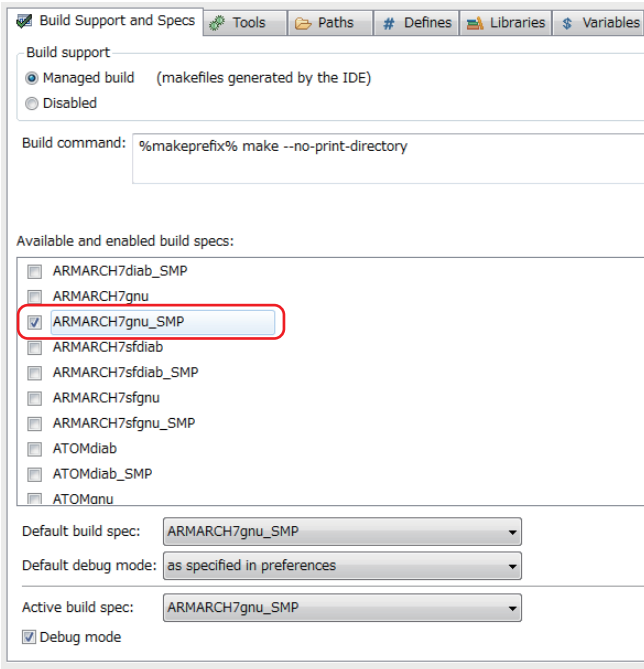
2. 画面左側のツリーから"Build Properties"を選択します。



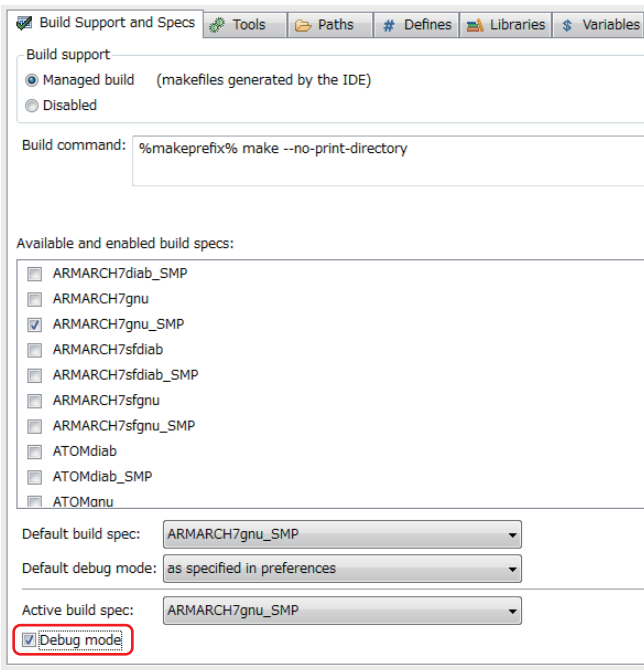
3. [Build Support and Specs]タブを選択します。



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



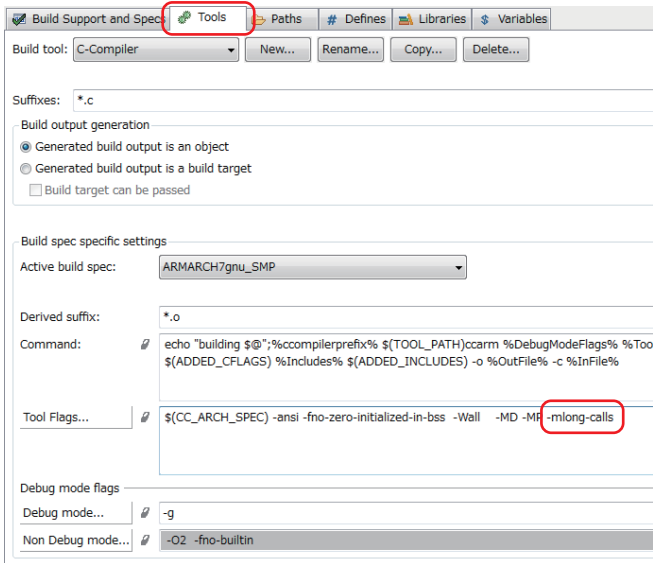
5. "Debug mode"にチェックを入れます。



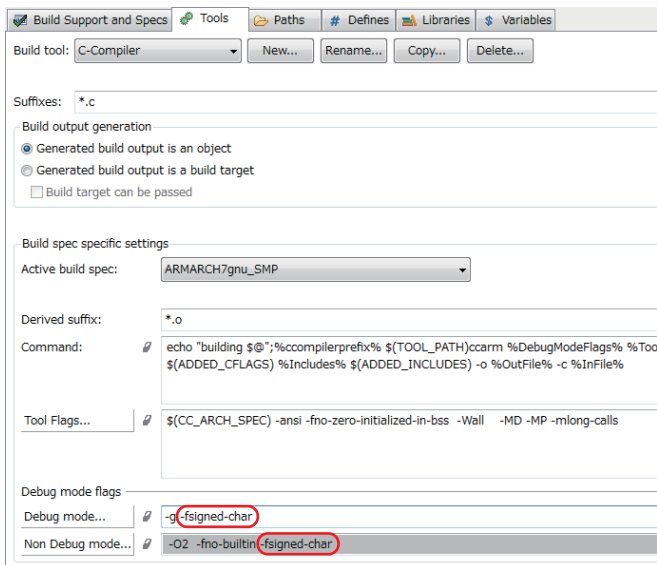
Point

正式なシステム運用、稼働時は、"Debug mode"のチェックをはずしてください。

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



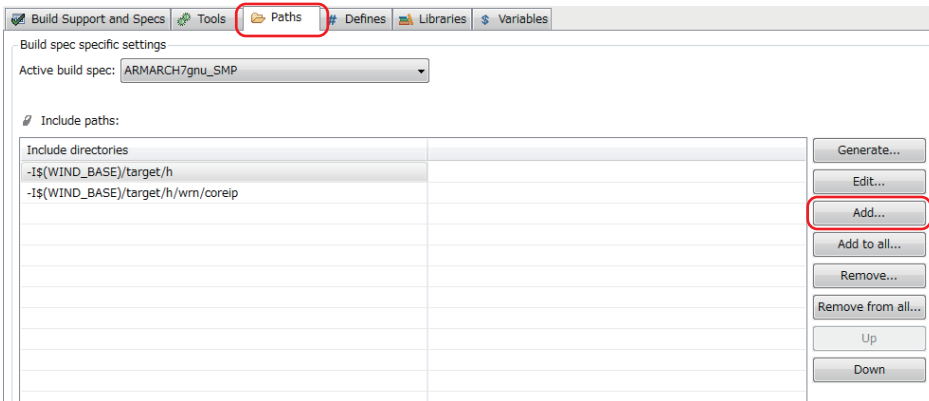
7. 同様に、"Debug mode flags"の[Debug mode]および[Non Debug mode]に"-fsigned-char"を入力します。



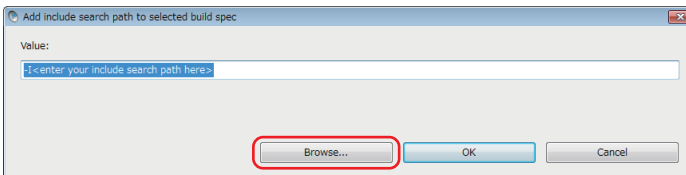
■インクルードファイルを設定

操作手順

1. [Paths]タブを選択し、[Add]ボタンをクリックします。

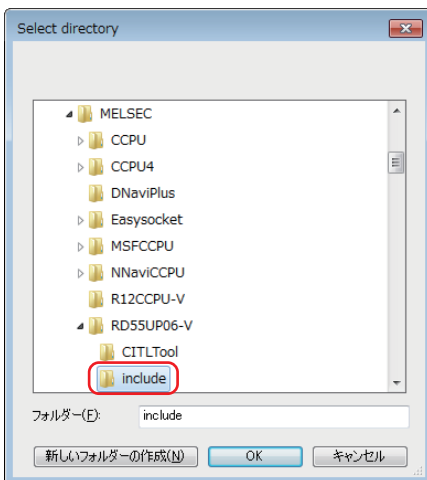


2. [Browse]ボタンをクリックします。

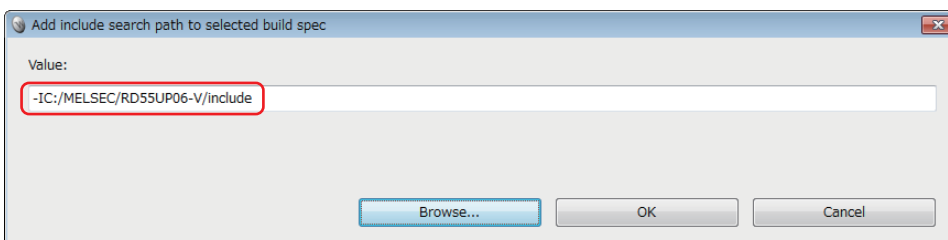


3. "Select directory"画面からC言語インテリジェント機能ユニット専用のインクルードフォルダを選択し、[OK]ボタンをクリックします。

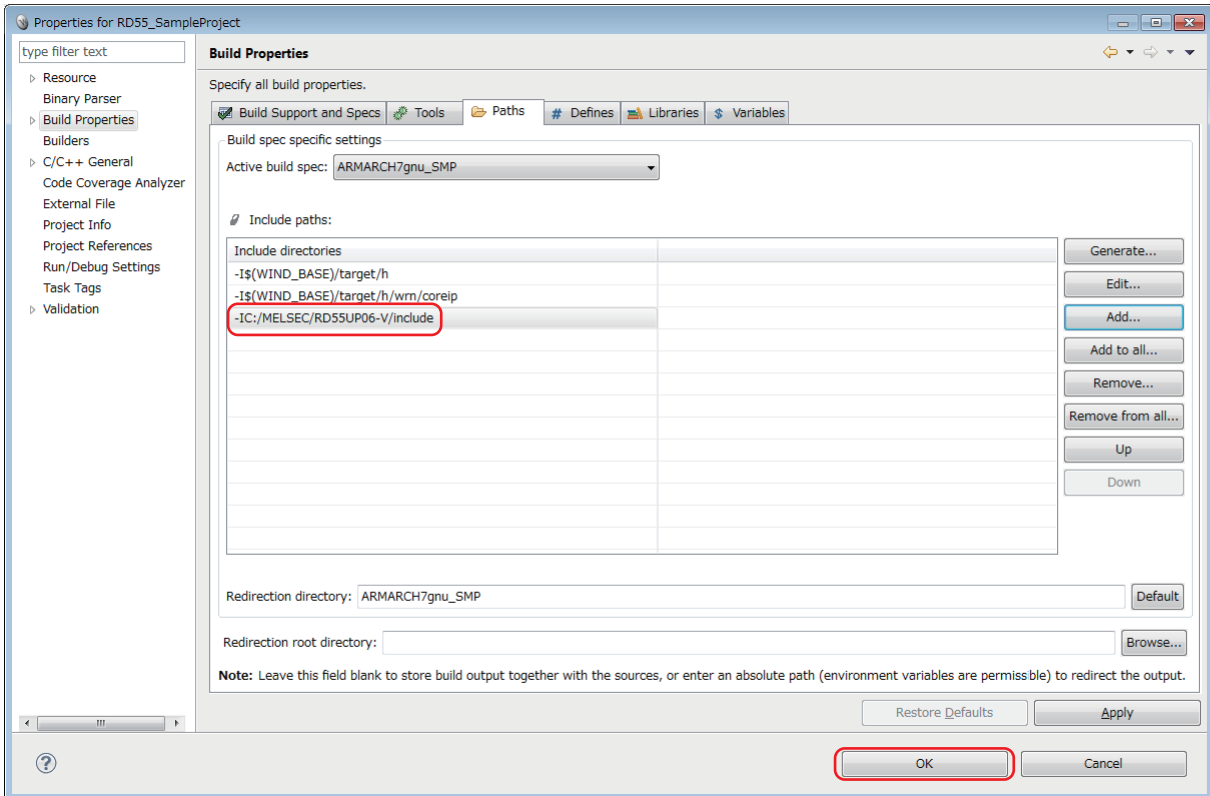
ここでは"C:\MELSEC\RD55UP06-V"とした場合のフォルダになります。



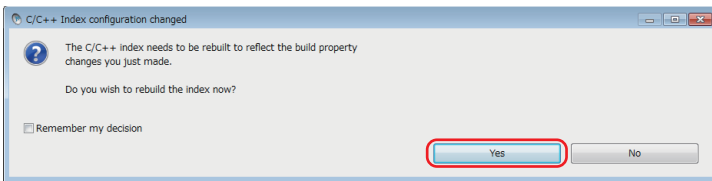
4. "Add include search path to selected build spec"画面で選択したフォルダが指定されていることを確認し、[OK]ボタンをクリックします。



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言語インテリジェント機能ユニット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/

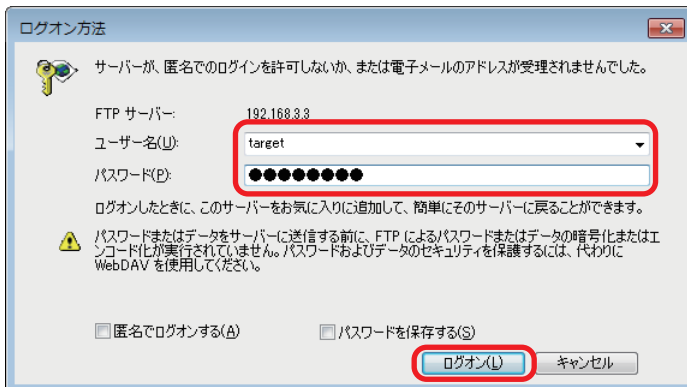


"ログイン方法"画面が表示されます。

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言語インテリジェント機能ユニット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。

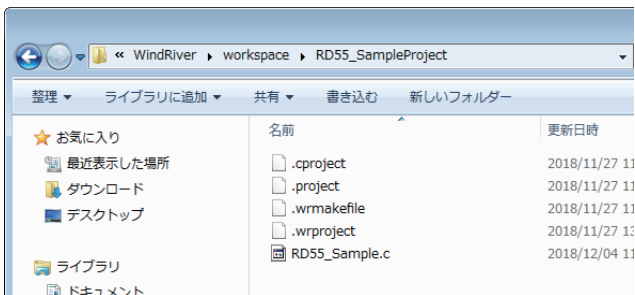
Point

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

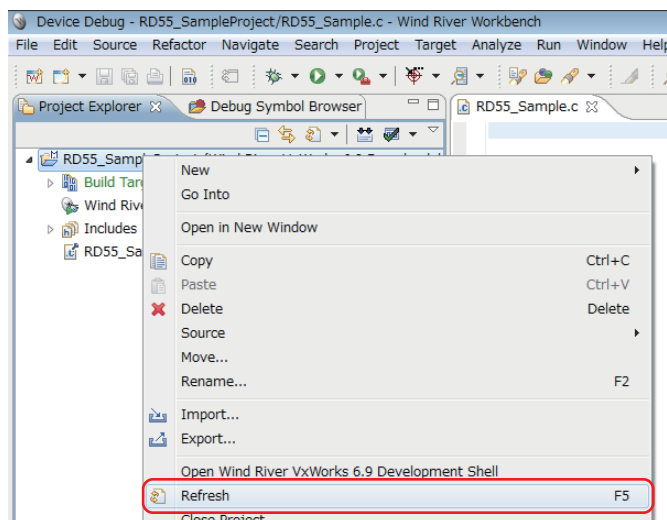
サンプルプログラムを追加する

操作手順

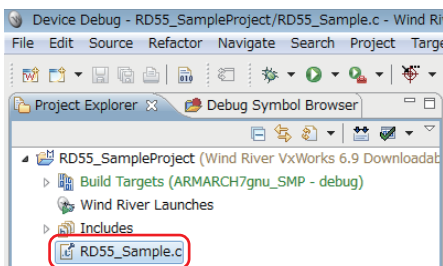
1. 今回作成したプロジェクトフォルダ直下に、本クイックスタートガイド専用のサンプルプログラムを格納します。
"C:\WindRiver\workspace\RD55_SampleProject"



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。

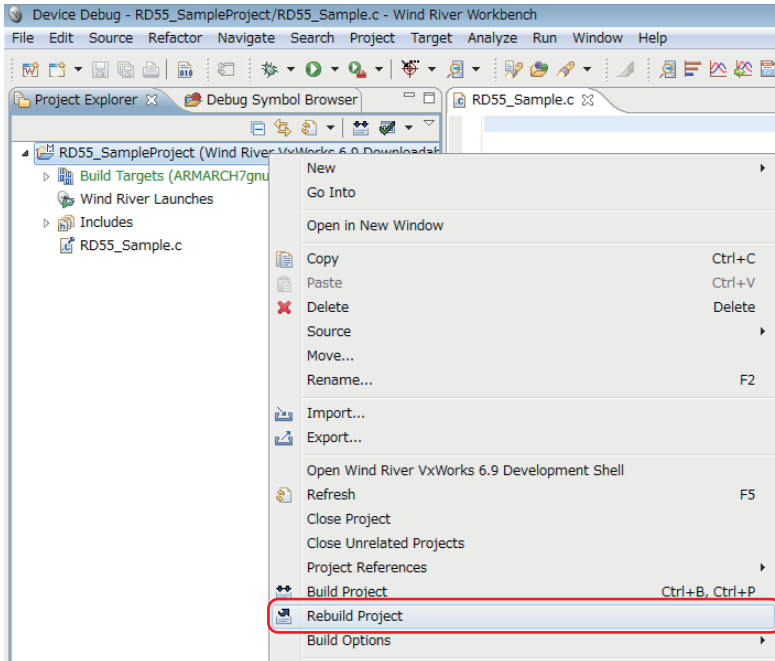


ユーザプログラムから実行モジュールを生成する

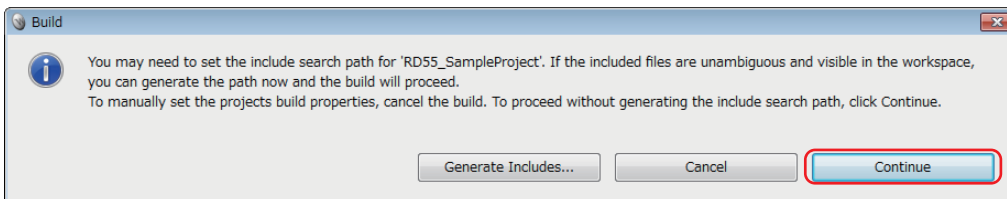
作成したプログラムを、C言語インテリジェント機能ユニットで実行可能なモジュールに変換(ビルド)します。

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。

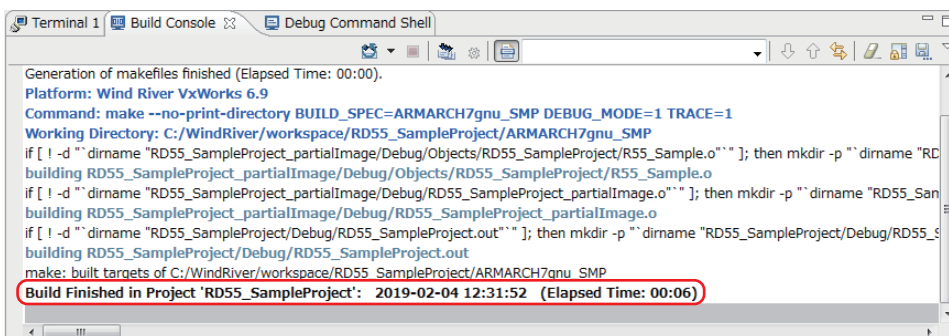


2. [Rebuild Project]を選択後、下記のメッセージが表示された場合は、[Continue]ボタンをクリックします。



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「31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。

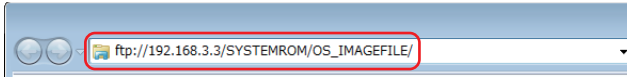
C言語インテリジェント機能ユニットとCW Workbenchを接続する

CW Workbenchでデバッグを行うために、C言語インテリジェント機能ユニットのEthernetポートとCW Workbenchを接続します。

操作手順

1. C言語インテリジェント機能ユニットから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。

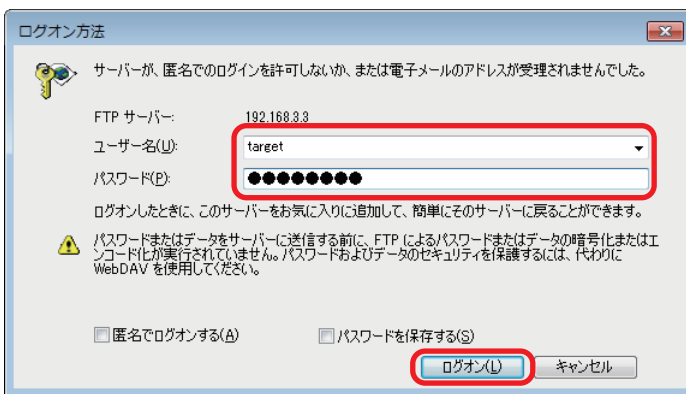
Point

C言語インテリジェント機能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

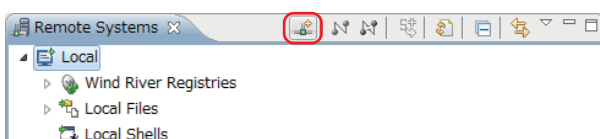
3. [ログオン]ボタンをクリックします。



4. "C:\MELSEC\RD55UP06-V\CITLTool"フォルダを作成し、C言語インテリジェント機能ユニットに格納されているVxWorksイメージファイルを"C:\MELSEC\RD55UP06-V\CITLTool"にコピーします。

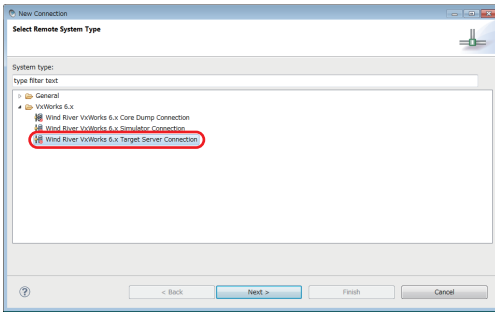


5. "Remote Systems"ウィンドウ内のをクリックします。

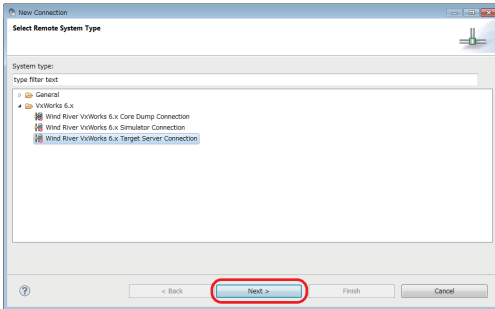


"New Connection"画面が表示されます。

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

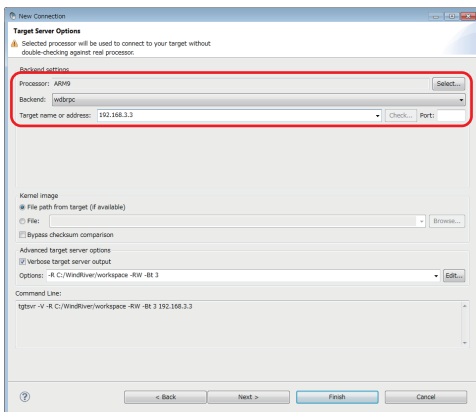


7. [Next]ボタンをクリックします。

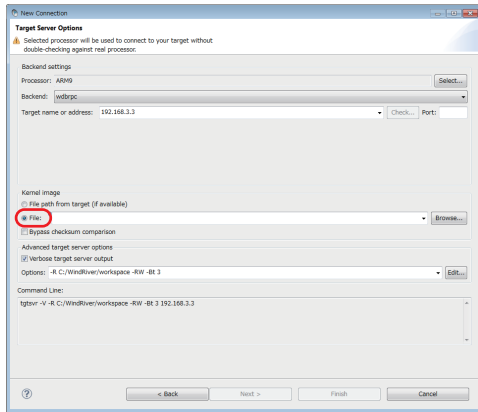


8. "Backend settings"の設定項目で、下記の内容を設定します。

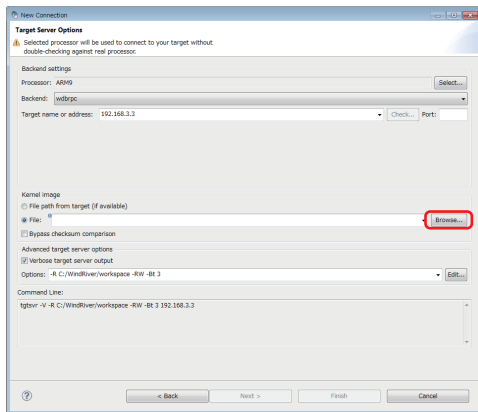
- Processor: ARM9([Select]ボタンをクリックし、ツリーから選択)
- Backend: wdbrpc
- IP Address: 192.168.3.3(デフォルト)
- Port: 空欄



9. "Kernel image"で"File"を選択します。

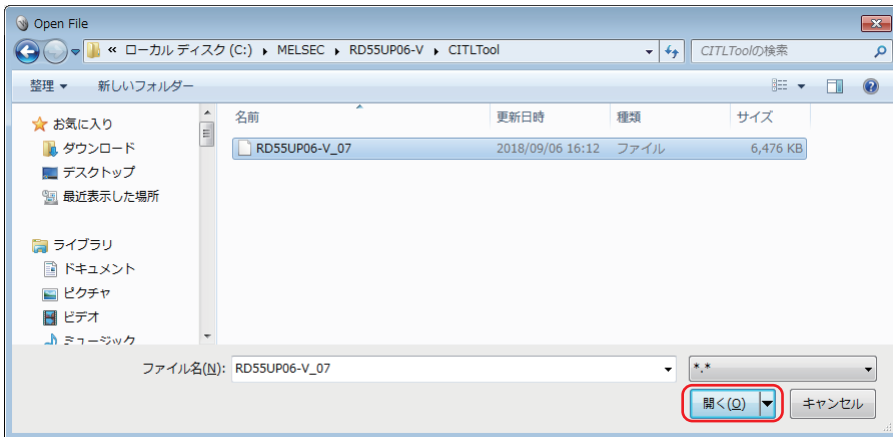


10. [Browse]ボタンをクリックします。

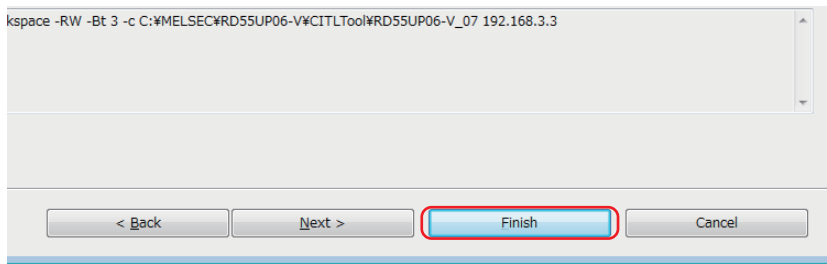


"Open File"画面が表示されます。

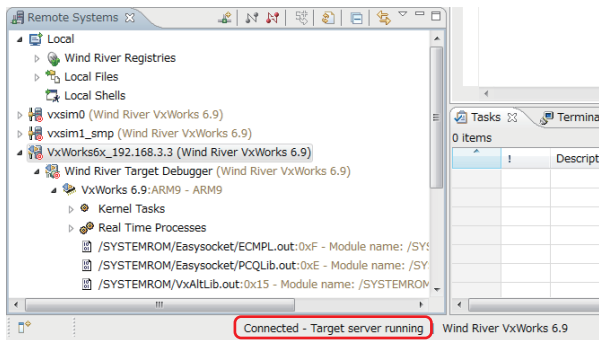
11. 手順4で"C:\MELSEC\RD55UP06-V\CITLTool"にコピーしたVxWorksイメージファイルを選択し、[開く]ボタンをクリックします。



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言語インテリジェント機能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「32ページC言語インテリジェント機能ユニットと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言語インテリジェント機能ユニットにダウンロードする

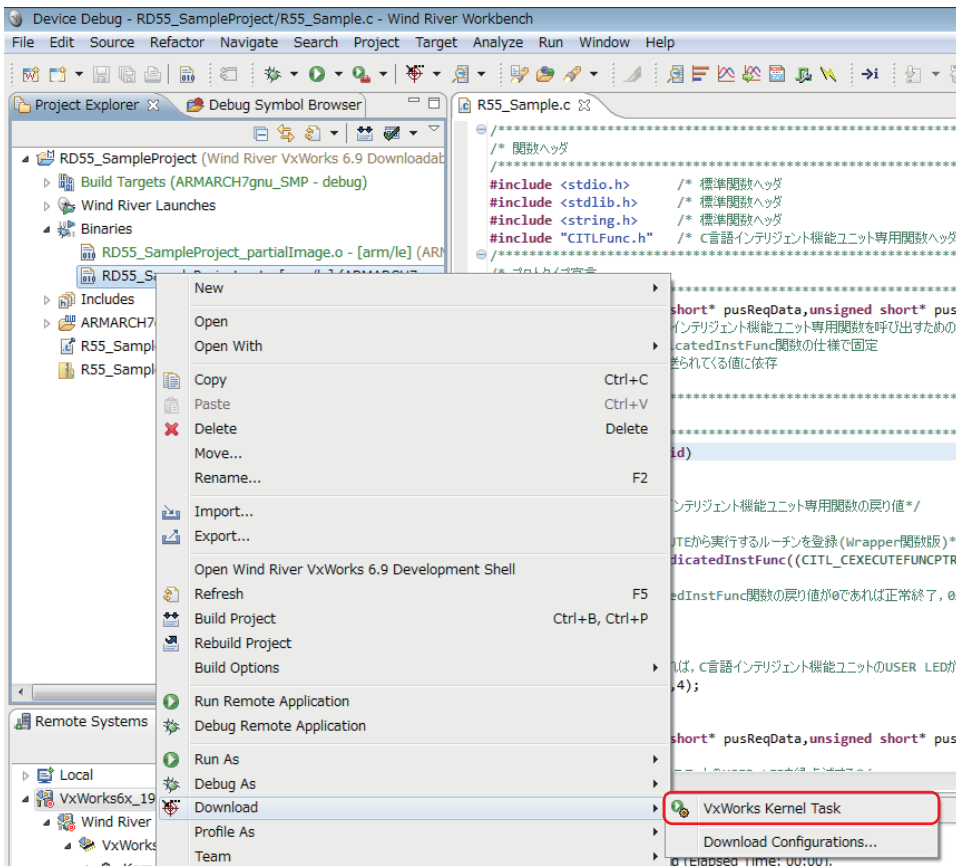
デバッグを行うために、実行モジュールをC言語インテリジェント機能ユニットのメモリにダウンロードします。ダウンロードを行うと、スクリプトファイルがなくてもプログラムが実行できます。

Point

スクリプトファイル: C言語インテリジェント機能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。

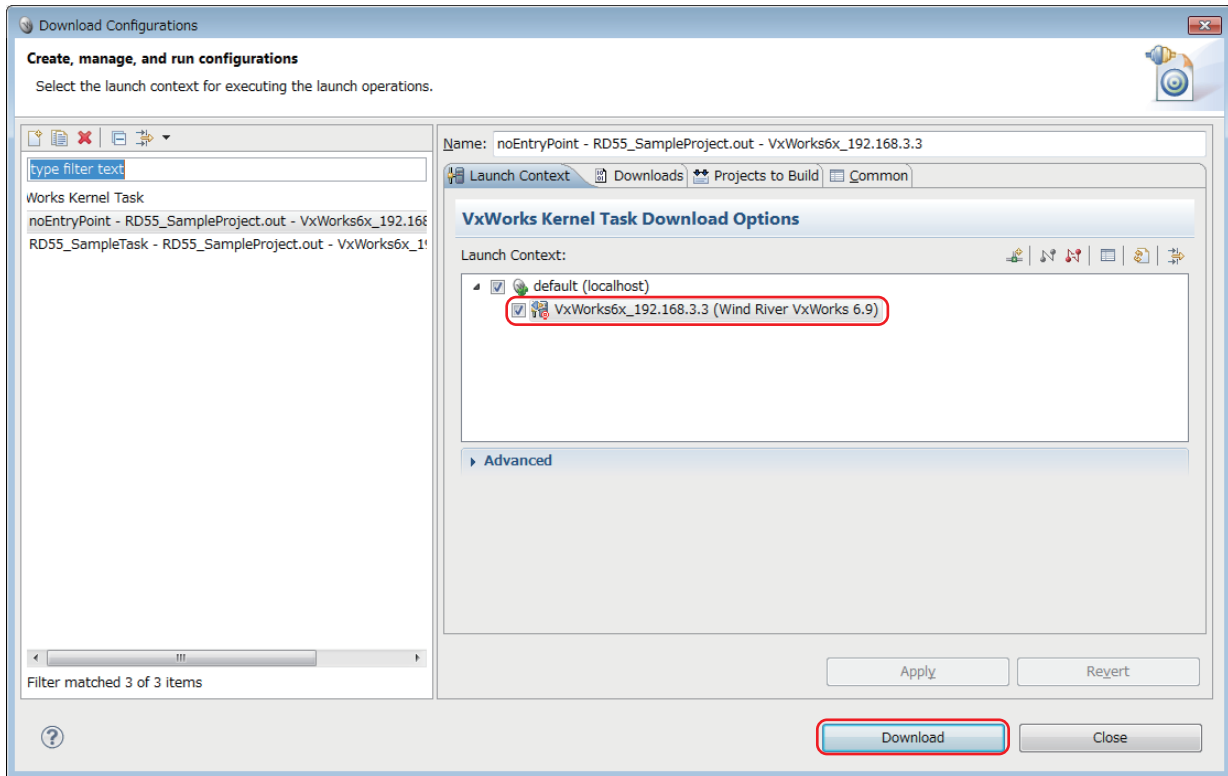
操作手順

1. "Project Explorer"ウィンドウから作成したモジュールファイル"RD55_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。

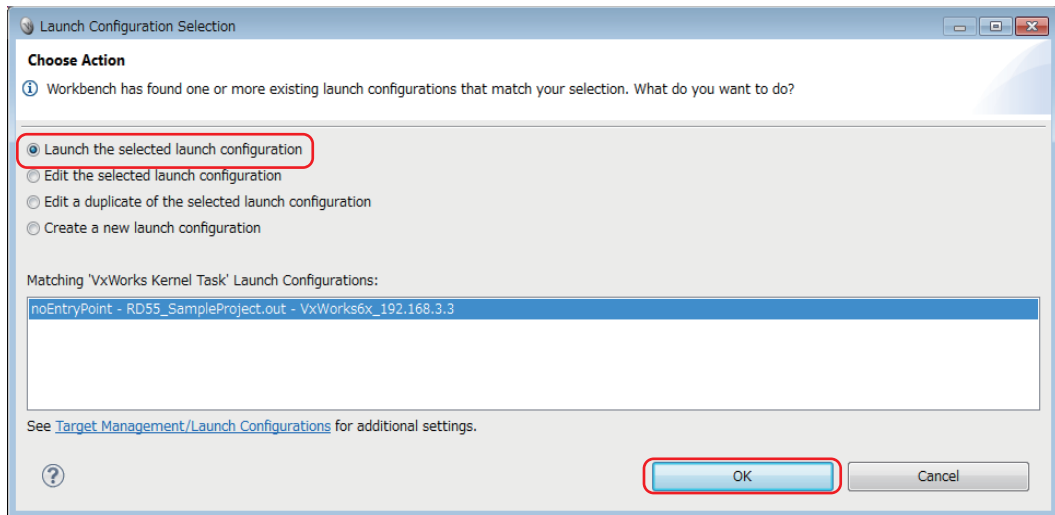
2. [Launch Context]タブにて"VxWorks6x_192.168.3.3(Wind River VxWorks 6.9)"のみにチェックを入れ, [Download]ボタンをクリックします。



2

Point

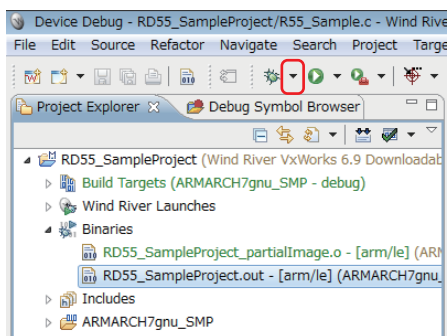
2回目以降の2の操作では, "Launch Configuration Selection"画面が表示されます。
"Launch the selected launch configuration"を選択し, [OK]ボタンをクリックしてください。



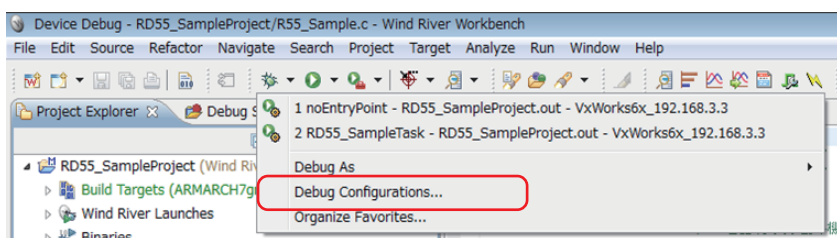
■ユーザプログラムをデバッグする

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。

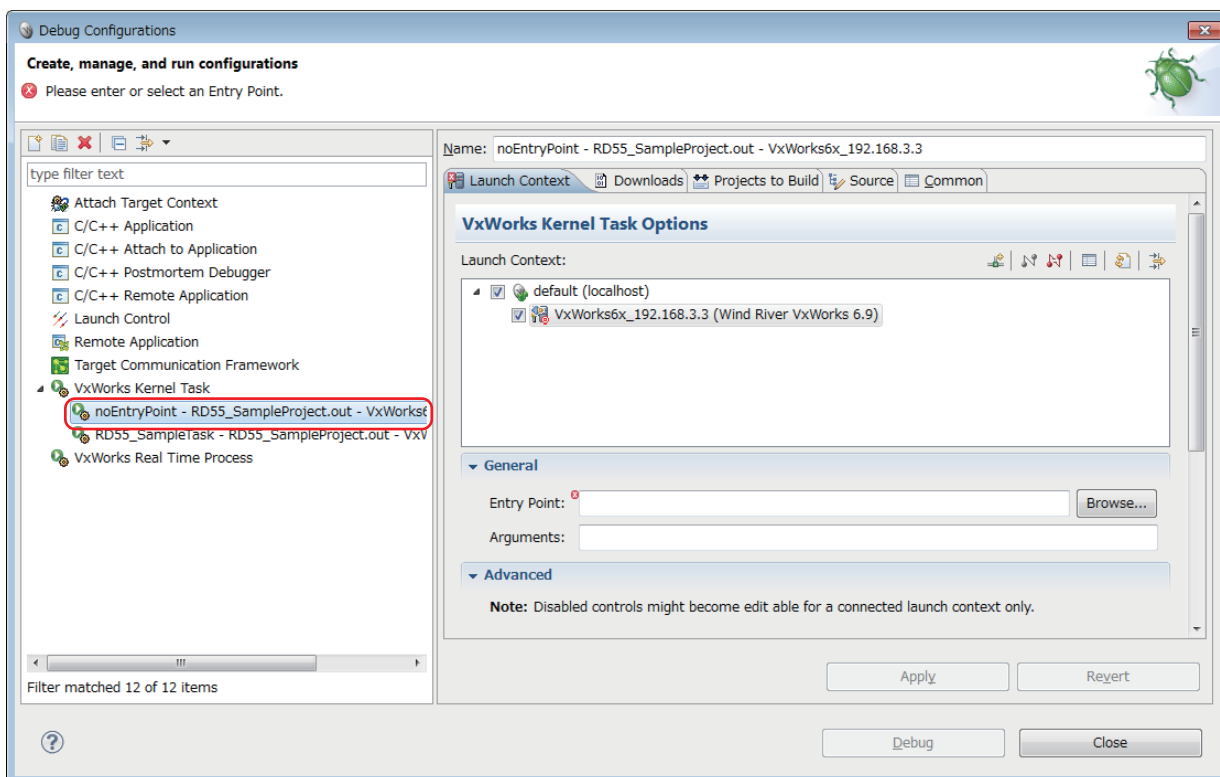


2. [Debug Configurations]を選択します。

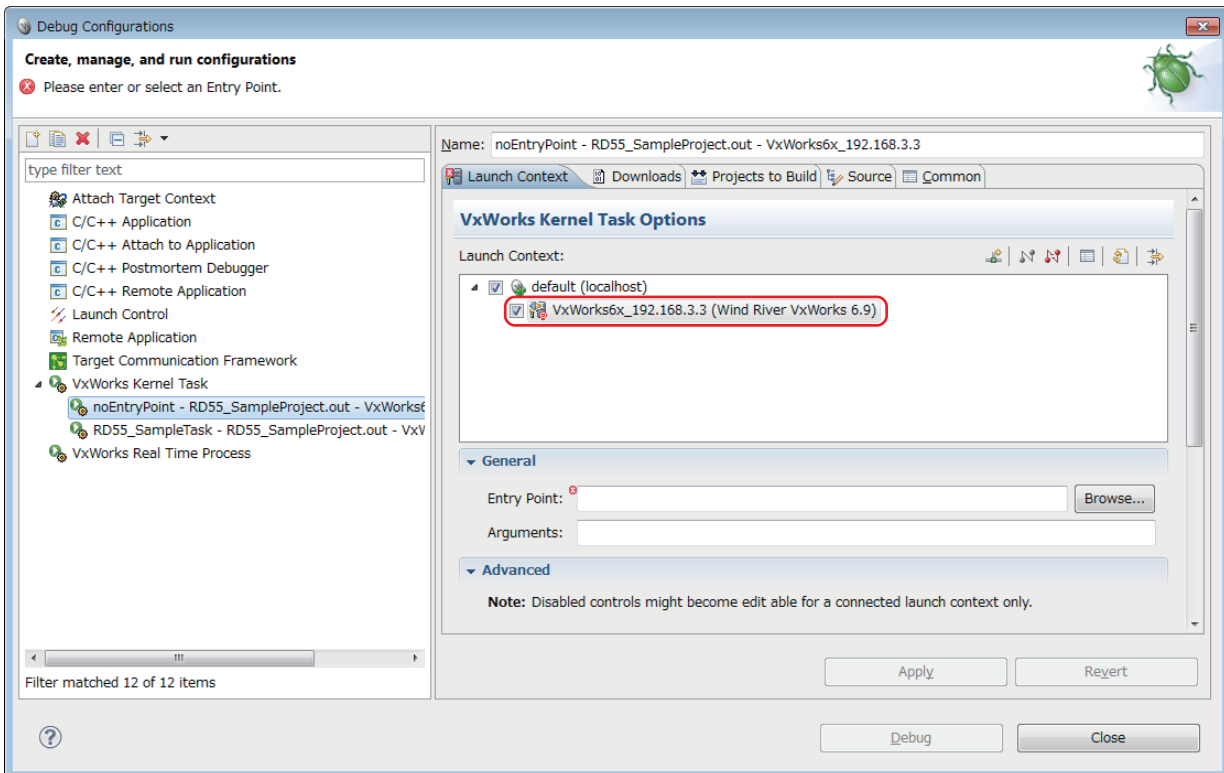


"Debug Configurations"画面が表示されます。

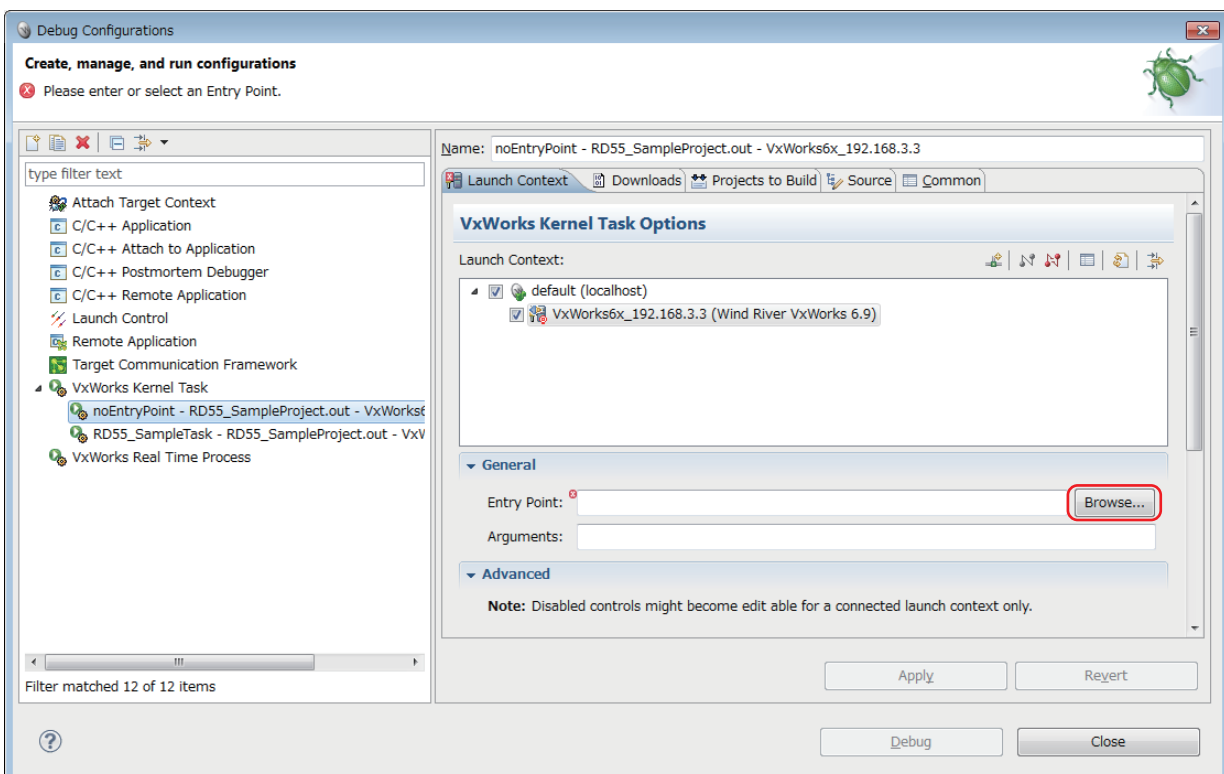
3. "VxWorks Kernel Task"からダウンロードしたモジュール"RD55_SampleProject.out"を選択します。



4. C言語インテリジェント機能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。

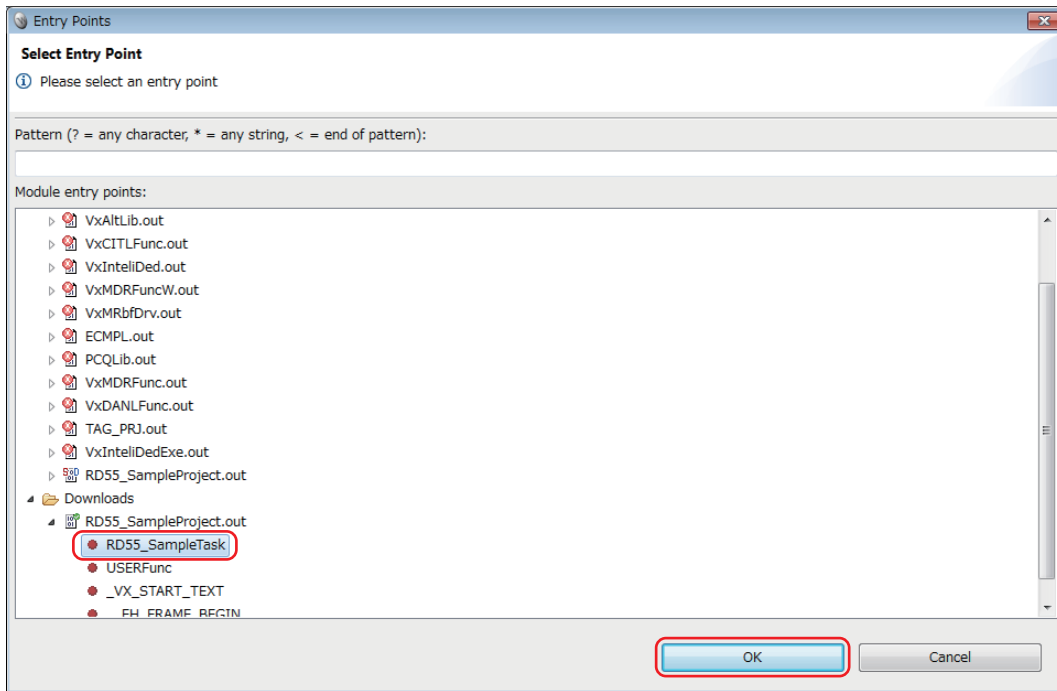


5. [Browse]ボタンをクリックします。

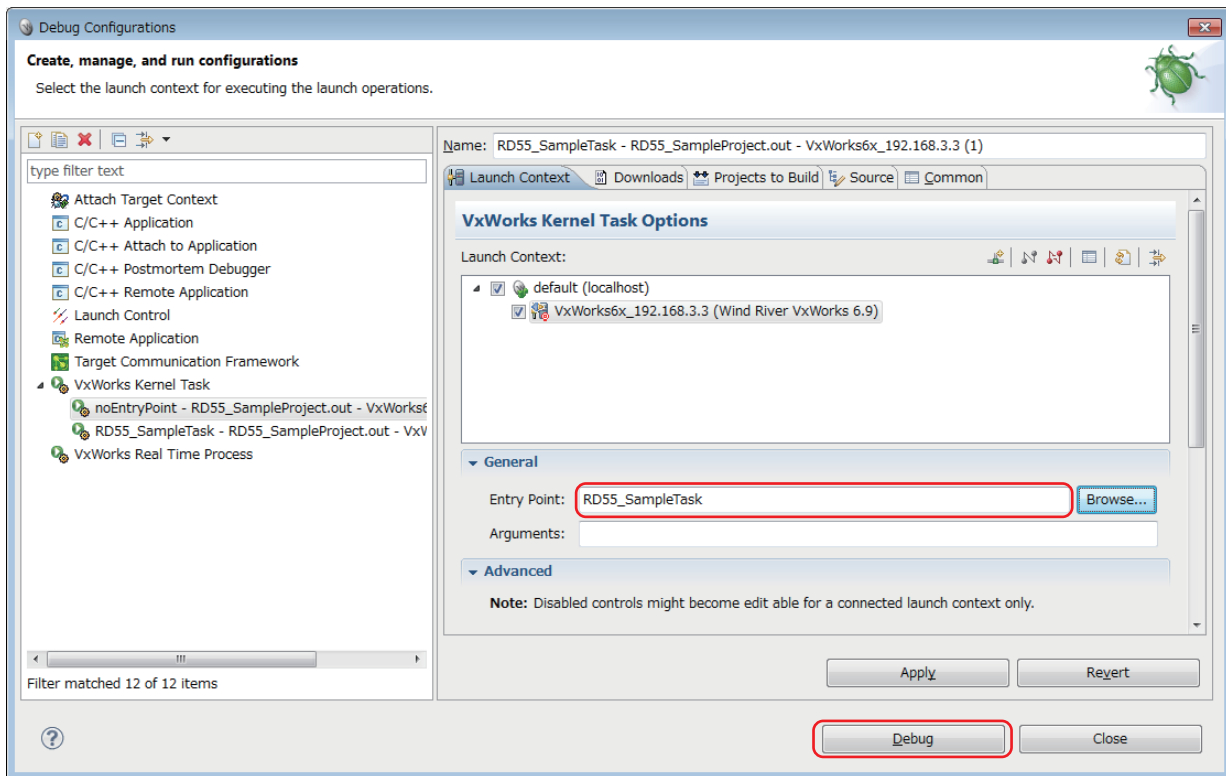


"Entry Points"画面が表示されます。

6. デバッグを開始する関数"RD55_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。




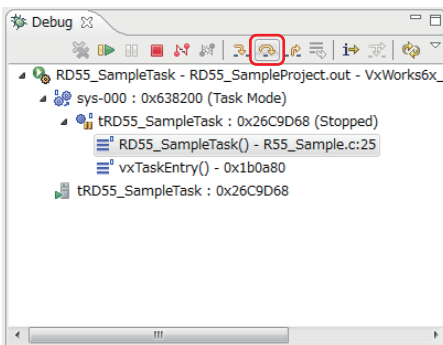
8. デバッグが開始され, "Entry Point"で指定した関数の先頭でプログラムの実行が止まります。

```

R55_Sample.c
/* 関数ヘッダ
*/
#include <stdio.h> /* 標準関数ヘッダ
#include <stdlib.h> /* 標準関数ヘッダ
#include <string.h> /* 標準関数ヘッダ
#include "CITLFunc.h" /* C言語インテリジェント機能ユニット専用関数ヘッダ
*/
/* プロトタイプ宣言
*/
long Wrapper( unsigned short* pusReqData,unsigned short* pusReqSize,unsigned short* pusAnsData,unsigned short* pusAnsSize );
/*マザープログラムからC言語インテリジェント機能ユニット専用関数を呼び出すための登録用関数
引数はCITL_EntryDedicatedInstFunc関数の仕様で固定
各引数の値はラダー側から送られてくる値に依存
*/
/* LED制御処理
*/
void RD55_SampleTask(void)
{
/*ローカル変数の宣言*/
short sRet; /*C言語インテリジェント機能ユニット専用関数の戻り値*/
/*専用命令G(P).CEXECUTEから実行するルーチンを登録 (Wrapper関数版)*/
sRet = CITL_EntryDedicatedInstFunc((CITL_CEXECUTEFUNCPTR)Wrapper);
/*CITL_EntryDedicatedInstFunc関数の戻り値が0であれば正常終了, 0以外の場合は異常終了する*/
if(sRet != 0){
return;
}
/*プログラムが実行されているば, C言語インテリジェント機能ユニットのUSER LEDが緑色点灯する*/
CITL_SetLEDStatus(0,4);
}
long Wrapper( unsigned short* pusReqData,unsigned short* pusReqSize,unsigned short* pusAnsData,unsigned short* pusAnsSize )
{
}

```

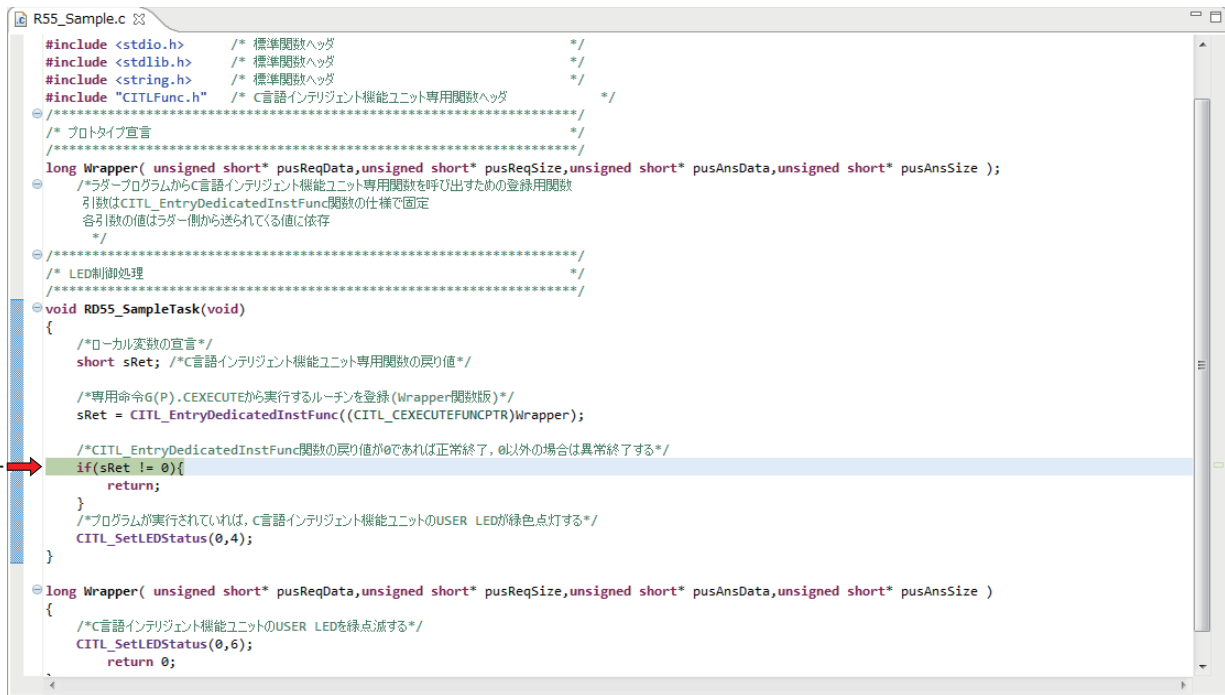
9. "Debug"ウィンドウので, 1ステップごとにデバッグを行います。



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CITL関数の戻り値である"err"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。



```
#include <stdio.h> /* 標準関数ヘッダ */
#include <stdlib.h> /* 標準関数ヘッダ */
#include <string.h> /* 標準関数ヘッダ */
#include "CITLFunc.h" /* C言語インテリジェント機能ユニット専用関数ヘッダ */

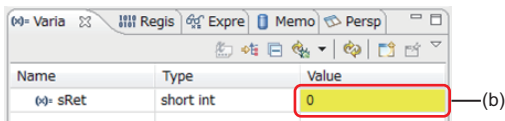
/* プロトタイプ宣言 */
/* ===== */
long Wrapper( unsigned short* pusReqData, unsigned short* pusReqSize, unsigned short* pusAnsData, unsigned short* pusAnsSize );
/* ラダープログラムからC言語インテリジェント機能ユニット専用関数を呼び出すための登録用関数
   *戻りはCITL_EntryDedicatedInstFunc関数の仕様で固定
   *各戻り値の値はラダー側から送られてくる値に依存
   */
/* ===== */
/* LED制御処理 */
/* ===== */
void RD55_SampleTask(void)
{
    /*ローカル変数の宣言*/
    short sRet; /*C言語インテリジェント機能ユニット専用関数の戻り値*/

    /*専用命令G(P).CEXECUTEから実行するルーチンを登録(Wrapper関数版)*/
    sRet = CITL_EntryDedicatedInstFunc((CITL_CEXECUTEFUNCPTR)Wrapper);

    /*CITL_EntryDedicatedInstFunc関数の戻り値が0であれば正常終了, 0以外の場合は異常終了する*/
    if(sRet != 0){
        return;
    }
    /*プログラムが実行されていれば, C言語インテリジェント機能ユニットのUSER LEDが緑色点灯する*/
    CITL_SetLEDStatus(0,4);
}

long Wrapper( unsigned short* pusReqData, unsigned short* pusReqSize, unsigned short* pusAnsData, unsigned short* pusAnsSize )
{
    /*C言語インテリジェント機能ユニットのUSER LEDを緑点滅する*/
    CITL_SetLEDStatus(0,6);
    return 0;
}
```

(b)[Variables]タブで、errの値が0(正常値)であることを確認する。



手順9, 10を繰り返して、作成したプログラム全体をデバッグします。

Point

C言語インテリジェント機能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。

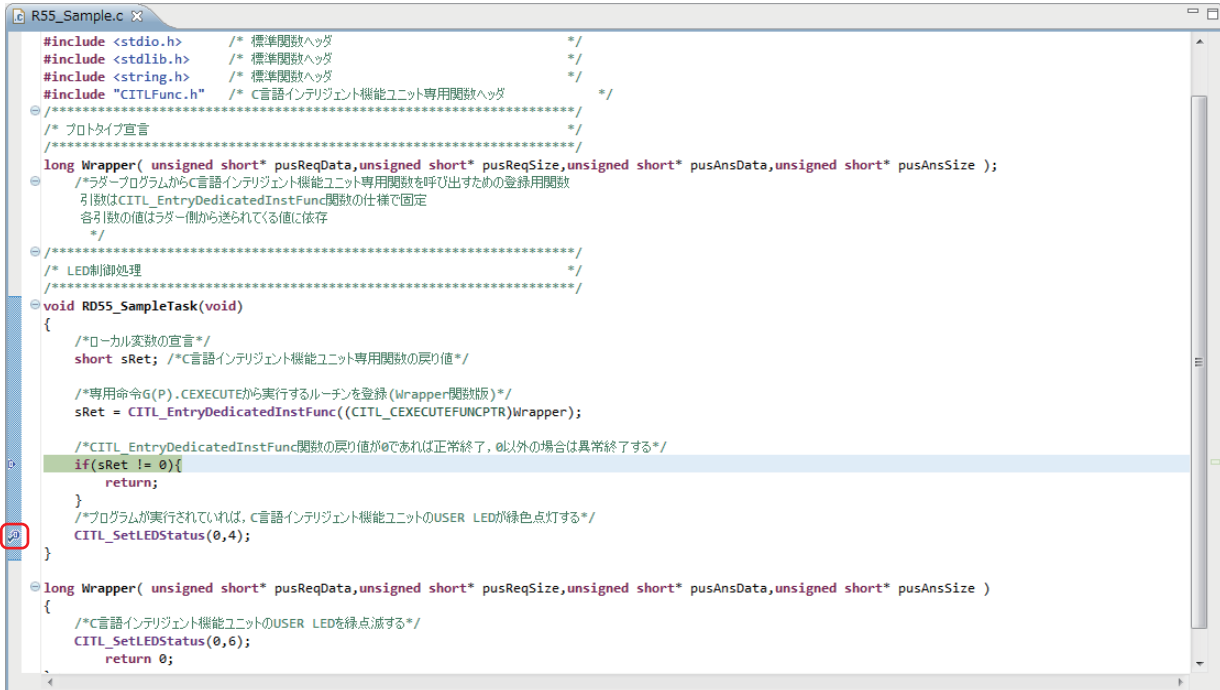
📖 MELSEC iQ-R C言語インテリジェント機能ユニットプログラミングマニュアル


■Breakpointを使用したデバッグ方法

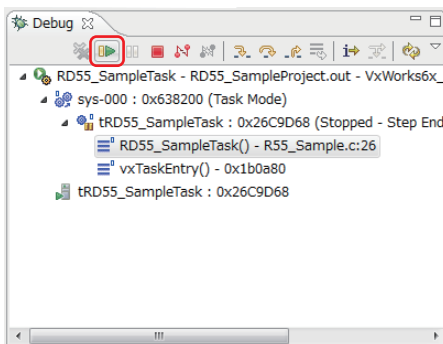
手順9の1ステップごとのデバッグではなく、プログラム内の任意の位置にBreakpointを指定してデバッグを進めることができます。

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。

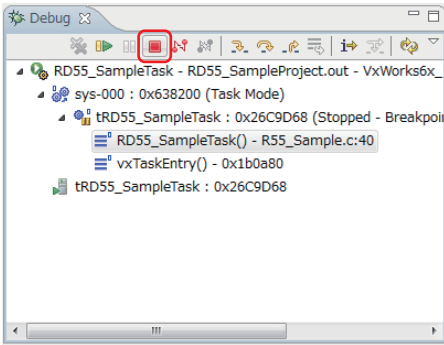


Point


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

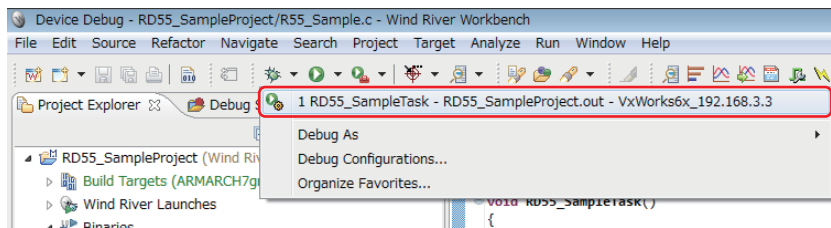
3. "Debug"ウィンドウのをクリックし、デバッグを終了します。



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「38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。



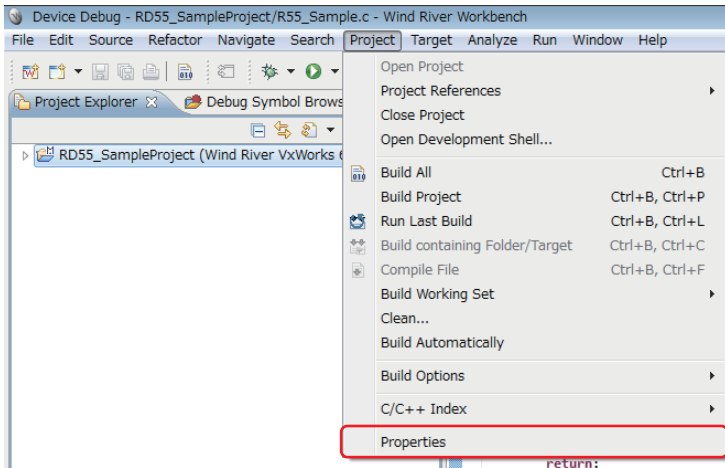
実行モジュールを登録する

作成したプログラムを稼動用にビルドし，C言語インテリジェント機能ユニットに格納します。

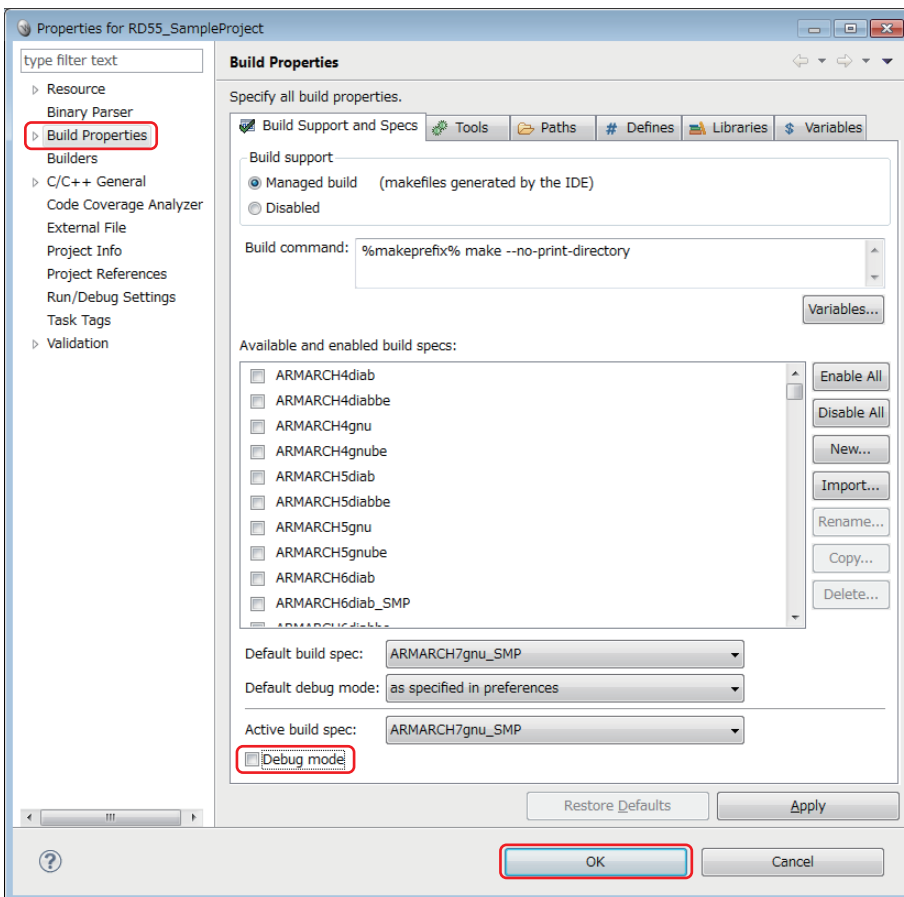
■ユーザプログラムをビルドする

操作手順

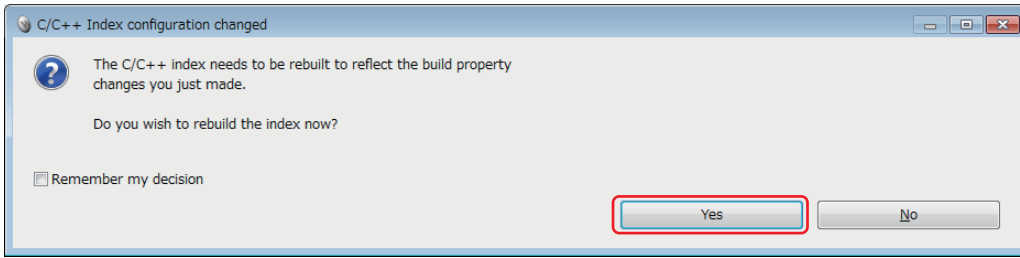
1. "Project Explorer"ウィンドウから作成したプロジェクトを選択し，[Project]⇒[Properties]を選択します。



2. 画面左側のツリーから"Build Properties"を選択し，"Debug mode"のチェックをはずし，[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. 「31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。

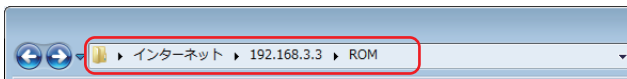
■ユーザプログラムを格納する

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/ROM

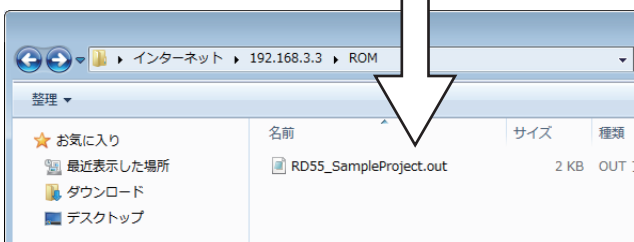
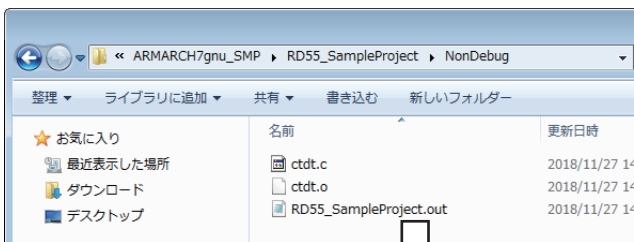
C言語インテリジェント機能ユニットへログインすると、下記のように表示されます。



2. 作成したユーザプログラム"RD55_SampleProject.out"を、ドラッグ&ドロップでC言語インテリジェント機能ユニットの標準ROM「ROM」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\RD55_SampleProject\ARMARCH7gnu_SMP\RD55_SampleProject\NonDebug

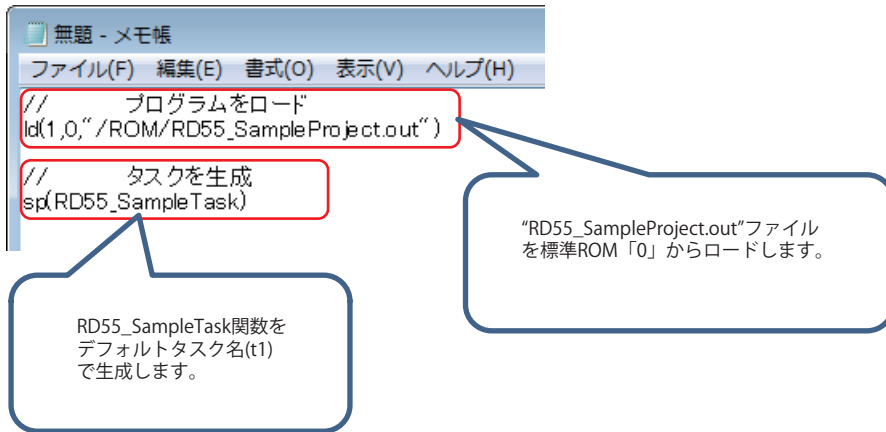


■スクリプトファイルを作成・格納する

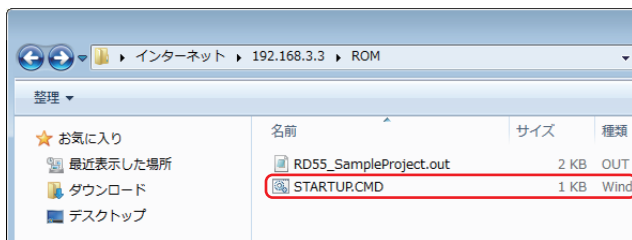
C言語インテリジェント機能ユニットを起動時に、実行モジュールを自動的にロードするためのスクリプトファイルを作成します。

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言語インテリジェント機能ユニットの標準ROMへコピーします。
ftp://192.168.3.3/ROM



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標準ROM以外に、SDメモ리카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리카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

CPUユニットのラダープログラムを実行し、C言語インテリジェント機能ユニットの動作を確認します。
操作にはCPU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: プログラムを実行する。
- STOP: プログラムを停止する。
- RESET: CPUユニットをリセットする。

Point

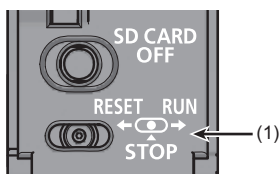
CPUユニットのRUN/STOP/RESETスイッチの詳細については、下記を参照してください。

📖 MELSEC iQ-R CPUユニットユーザーズマニュアル(スタートアップ編)

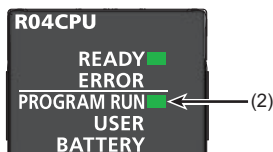
ラダープログラムの実行

操作手順

1. CPUユニット前面のRUN/STOP/RESET(1)をRUN側に倒します。



2. PROGRAM 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。

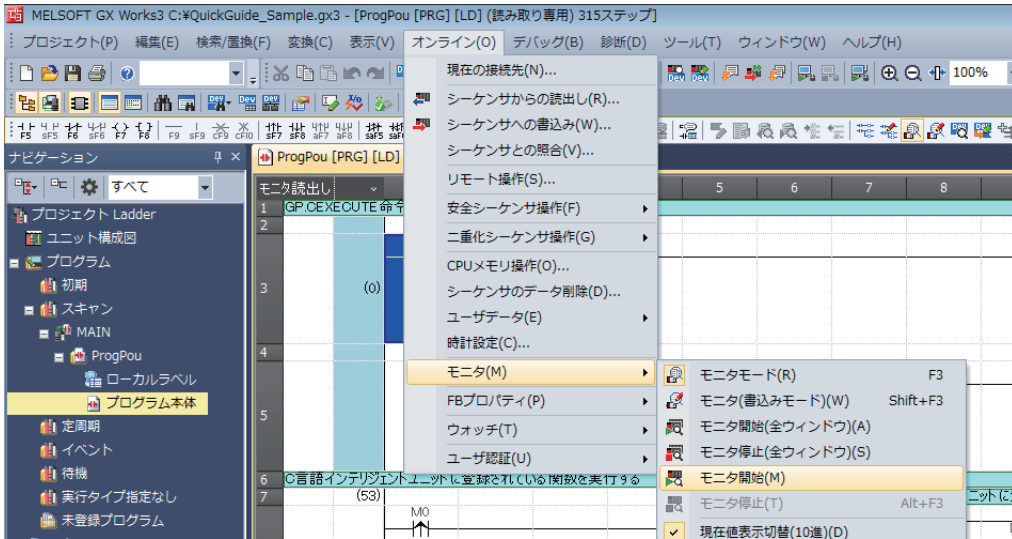


LEDで動作を確認する

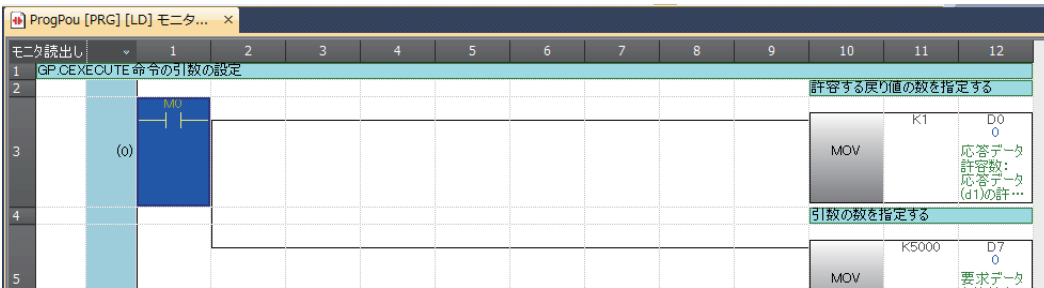
ユーザプログラムで登録したルーチンがラダープログラムの専用命令(G(P).CEXECUTE)で実行され、C言語インテリジェント機能ユニットのUSER LEDが緑点滅します。

操作手順

1. GX Works3のメニューから、[オンライン]⇒[モニタ]⇒[モニタ開始]を選択します。



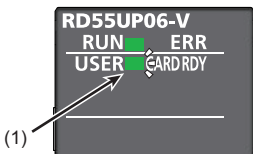
2. "MO"を選択し、**Shift**+**Enter**を押します。



MOがONし、専用命令が実行されます。

3. C言語インテリジェント機能ユニットのUSER LEDを確認します。

専用命令によって登録したルーチンが正常に実行された場合は、USER LED(1)が緑点滅します。



専用命令による登録ルーチンの実行が正常完了されなかった場合、USER LEDが点滅しません。"デバイス/バッファメモリー括モニタ"画面などで、専用命令の完了状態を確認してください。

完了状態	デバイス	
	Y100	Y101
正常完了	1	0
異常完了	0	1

3 よく使う機能

C言語インテリジェント機能ユニットで、システム立上げ時や運用・稼働後の保守などによく使う機能を紹介します。

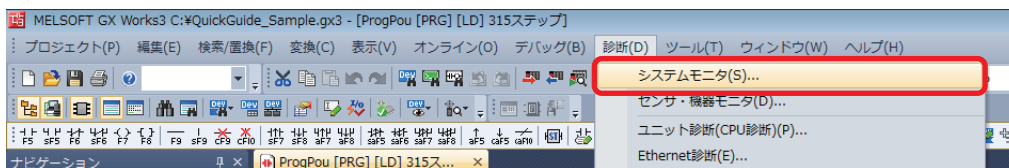
3.1 エラーを確認する

GX Works3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. GX Works3を起動します。
2. [診断]⇒[システムモニタ]を選択します。



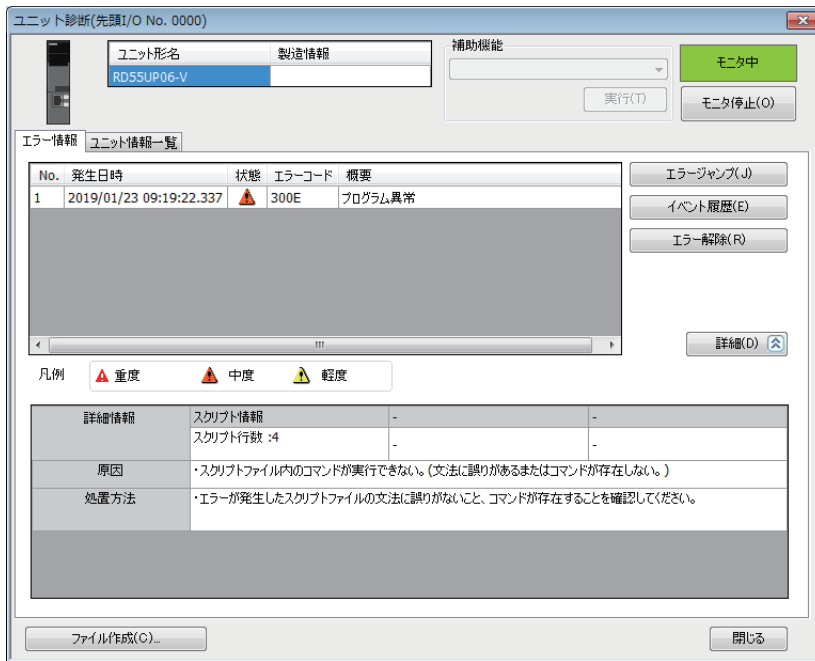
"システムモニタ"画面が表示されます。

3. エラーが発生しているユニットをダブルクリックします。



"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。また、原因、処置方法などの詳細な情報を確認できます。

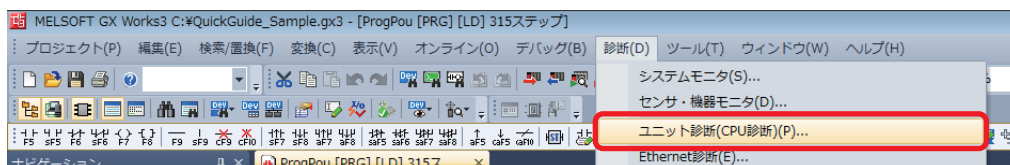


今までに発生したエラー履歴の確認方法

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。

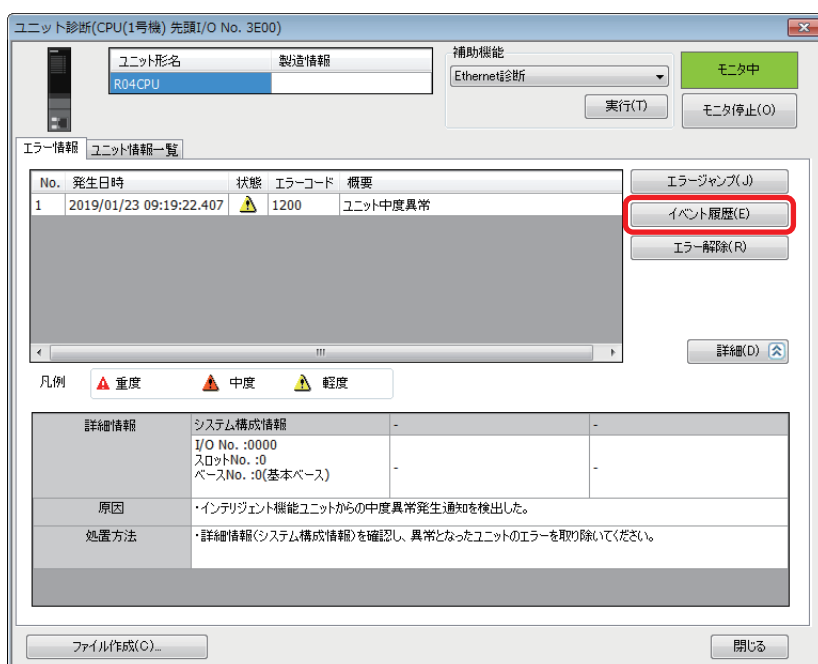
操作手順

1. GX Works3を起動します。
2. [診断]⇒[ユニット診断(CPU診断)]を選択します。



"ユニット診断"画面が表示されます。

3. [イベント履歴]ボタンをクリックします。



4. CPUユニットおよびC言語インテリジェント機能ユニットで、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。

The screenshot shows a software window titled "イベント履歴(CPU1号機) 先頭I/O No. 3E00". It contains a search filter section, a table of event logs, a legend for severity levels, and a detailed view of a specific event.

検索条件: イベント数: 1023

絞り込み: すべての条件に一致(A) いずれかの条件に一致(O)

絞り込み開始(S) 絞り込み条件全解除(E)

No.	発生日時	イベント種別	状態	イベントコード	概要
00001	2019/01/23 9:19:22.407	システム	⚠	01200	ユニット中度異常
00002	2019/01/23 9:19:22.337	システム	⚠	0300E	プログラム異常
00003	2019/01/23 9:19:12.546	オペレーション	ℹ	24101	動作状態の変更(STOP)
00004	2019/01/23 9:18:58.082	システム	ℹ	00400	電源ON/RESET解除
00005	2019/01/23 9:18:56.447	システム	⚠	01000	電源断発生
00006	2019/01/23 9:18:52.122	オペレーション	ℹ	20100	エラー解除
00007	2019/01/23 9:18:51.709	オペレーション	ℹ	20100	エラー解除

凡例: ⚠ 重度 ⚠ 中度 ⚠ 軽度
! 警告 ℹ 情報

ジャンプ(J) オールクリア(L)

詳細情報	システム構成情報	-	-
	I/O No. :0000		
	スロットNo. :0		
	ベースNo. :0(基本ベース)		
原因	・インテリジェント機能ユニットからの中度異常発生通知を検出した。		
処置方法	・詳細情報(システム構成情報)を確認し、異常となったユニットのエラーを取り除いてください。		

ファイル作成(C) 開じる

5. 詳細を知りたいエラーの行をクリックします。

No.	発生日時	イベント種別	状態	イベントコード	概要
00001	2019/01/23 9:19:22.407	システム	⚠	01200	ユニット中度異常
00002	2019/01/23 9:19:22.337	システム	🔥	0300E	プログラム異常
00003	2019/01/23 9:19:12.546	オペレーション	ℹ	24101	動作状態の変更(STOP)
00004	2019/01/23 9:18:58.082	システム	ℹ	00400	電源ON/RESET解除
00005	2019/01/23 9:18:56.447	システム	⚠	01000	電源断発生
00006	2019/01/23 9:18:52.122	オペレーション	ℹ	20100	エラー解除
00007	2019/01/23 9:18:51.709	オペレーション	ℹ	20100	エラー解除

詳細情報が表示されます。

イベント履歴(CPU(1号機) 先頭I/O No. 3E00)

最新の情報に更新(U) イベント数: 1023 絞り込み表示(D)

絞り込み

すべての条件に一致(A) いずれかの条件に一致(O)

1. イベント種別 以下を含む

2.

3.

絞り込み開始(S) 絞り込み条件全解除(E)

No.	発生日時	イベント種別	状態	イベントコード	概要
00001	2019/01/23 9:19:22.407	システム	⚠	01200	ユニット中度異常
00002	2019/01/23 9:19:22.337	システム	🔥	0300E	プログラム異常
00003	2019/01/23 9:19:12.546	オペレーション	ℹ	24101	動作状態の変更(STOP)
00004	2019/01/23 9:18:58.082	システム	ℹ	00400	電源ON/RESET解除
00005	2019/01/23 9:18:56.447	システム	⚠	01000	電源断発生
00006	2019/01/23 9:18:52.122	オペレーション	ℹ	20100	エラー解除
00007	2019/01/23 9:18:51.709	オペレーション	ℹ	20100	エラー解除

凡例 ⚠ 重度 ⚠ 中度 ⚠ 軽度
 警告 ℹ 情報

詳細情報	スクリプト情報	-	-
	スクリプト行数: 4	-	-
原因	スクリプトファイル内のコマンドが実行できない。(文法に誤りがあるまたはコマンドが存在しない。)		
処置方法	エラーが発生したスクリプトファイルの文法に誤りがないこと、コマンドが存在することを確認してください。		

Point

"絞り込み"に条件を設定することで、表示するユニットを絞り込めます。

イベント履歴(CPU(1号機) 先頭I/O No. 3E00)

最新の情報に更新(U) イベント数: 1023 絞り込み表示(D)

絞り込み

すべての条件に一致(A) いずれかの条件に一致(O)

1. 発生元ユニット 以下と一致する RD55UP06-V

2.

3.

絞り込み開始(S) 絞り込み条件全解除(E)

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系統的に設置してください。

三菱電機 汎用シーケンサ C言語インテリジェント機能ユニット 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
関東支社	〒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	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	-	表示器 GOT	052-712-2417	4→1	
エッジコンピューティング製品	産業用PC MELIPC	8	MELSOFT GTシリーズ	052-712-2962*2*8	4→2	
	Edgecross対応ソフトウェア(NC Machine Tool OptimizerなどのNC関連製品を除く)		SCADA GENESIS64™		-	
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	MELSERVOシリーズ	052-712-6607	1→2	
MELSEC iQ-F/FXシーケンサ全般	052-725-2271*3	2→1	位置決めユニット (MELSEC iQ-R/Q/Lシリーズ)		1→2	
ネットワークユニット(CC-Linkファミリ/ MELSECNET/Ethernet/シリアル通信)	052-712-2578	2→3	モーションユニット (MELSEC iQ-R/Q/Fシリーズ)		1→1	
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		モーションソフトウェア	1→1
MELSOFT統合 エンジニアリング環境	MELSOFT Navigator/ MELSOFT Update Manager	052-799-3591*2	2→6		シンプルモーションユニット (MELSEC iQ-R/Q/F/Q/Lシリーズ)	1→2
iQ Sensor Solution					モーションCPU (MELSEC iQ-R/Qシリーズ)	1→1
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		センシングユニット (MR-MTシリーズ)	1→2
MELSEC/パソコンボード	Q80BDシリーズなど					シンプルモーションボード/ ボジションボード
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					MELSOFT MTシリーズ/ MRシリーズ/FEMシリーズ	1→2
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	センサレスサーボ		FR-E700EX/MM-GKR	052-722-2182
システムレコーダ			インバータ	FREQROLシリーズ	052-722-2182	
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	三相モータ	三相モータ225フレーム以下	0536-25-0900*2*8
	プロセスCPU/二重化CPU (MELSEC-Qシリーズ) MELSOFT PXシリーズ				産業用ロボット	MELFA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	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	-
	安全コントローラ (MELSEC-WSシリーズ)				データ収集アナライザ	MELQIC IU1/IU2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	低圧開閉器	MS-Tシリーズ/MS-Nシリーズ US-Nシリーズ	052-719-4170
FAセンサ MELSENSOR	レーザ変位センサ	052-799-9495*2	6	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
	ビジョンセンサ			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
	コードリーダー			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
				小容量UPS (5kVA以下)	FW-Sシリーズ/FW-Aシリーズ/ FW-Aシリーズ/FW-Fシリーズ	052-799-9489*2*6

お問い合わせの際は、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をお願いいたします。