

E2 エミュレータ

ユーザーズマニュアル別冊

(RH850/U2BシリーズDFP連動デバッグのセットアップ)

対象デバイス

RH850ファミリ

RH850/U2Bシリーズ

DFP搭載デバイス

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
 5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
 7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
 8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

1.	概要	4
1.1	本書	4
1.2	マニュアル構成	4
2.	システム構成	6
2.1	ハードウェア構成	6
2.2	ソフトウェア構成	7
3.	G4MH と DFP 連動デバッグ環境セットアップ	8
3.1	RH850/U2Bシリーズデバイスのセットアップ	8
3.2	CS+デバッガのセットアップ	10
3.3	Virtual Boxのセットアップ	12
3.4	OpenOCDのセットアップ	17
3.5	NSI IDEのセットアップ	21
3.6	G4MHとDFPの連動デバッグシステム立ち上げ手順	24
4.	G4MH と DFP 連動デバッグ手法	25
4.1	G4MHとDFP連動実行	25
4.2	G4MHとDFP連動ブレーク	26
5.	問い合わせ先	27

1. 概要

1.1 本書

E2 エミュレータを使用して DFP IP(以下 DFP) 搭載 RH850/U2B シリーズデバイスの G4MH と DFP を連動デバッグする環境のセットアップおよびデバッグ手法を説明します。

1.2 マニュアル構成

E2 エミュレータを使用して RH850/U2B のデバッグを行う場合に必要となるドキュメントを説明します。

E2 エミュレータのマニュアルは、E2 エミュレータユーザズマニュアルと、E2 エミュレータユーザズマニュアル別冊の 2 部で構成されています。E2 エミュレータを使用するにあたっては、両方のユーザズマニュアルを必ずお読みください。

資料名	文書番号
E2エミュレータ RTE0T00020KCE00000R ユーザマニュアル	R20UT3538J
E2エミュレータ ユーザマニュアル別冊 (RH850/U2Bシリーズ接続時の注意事項)	R20UT5052J

DFP 搭載 RH850/U2B シリーズデバイスで G4MH と DFP を連動デバッグする場合には、G4MH のデバッガとして統合開発環境 CS+(以降 CS+デバッガ)を使用します。CS+デバッガを使用するにあたっては、CS+デバッガのオンラインヘルプを必ずお読みください。

資料名	URL
CS+ オンラインヘルプ	CS+ オンラインヘルプ (renesas.com)

DFP 搭載 RH850/U2B シリーズデバイスで G4MH と DFP を連動デバッグする場合には、DFP のデバッガとして NSITEXE 社の DR1000C デバッガ/IDE を使用します。DR1000C デバッガ/IDE を使用するにあたっては、DR1000C のドキュメントを必ずお読み下さい。

資料名
NSITEXE DR1000C Debugger User Manual
NSITEXE DR1000C IDE User Manual

2. システム構成

DFP 搭載 RH850/U2B シリーズデバイスで G4MH と DFP を連動デバッグするためのシステム構成を説明します。

2.1 ハードウェア構成

DFP搭載RH850/U2BシリーズデバイスでG4MHとDFPを連動デバッグするためのハードウェア構成を図 2-1および表 2-1に示します。

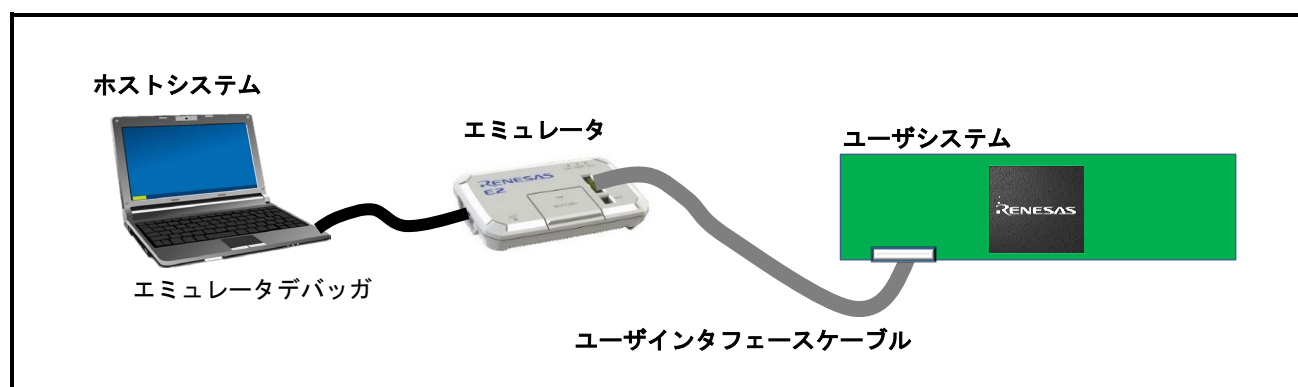


図 2-1 ハードウェア構成図

表 2-1 ハードウェア構成一覧

名称	製品名	備考
エミュレータ デバッグ	ルネサス製 GS+デバッグ	-
	NSITEXE 製 デバッグ/IDE	NSITEXE 社製 DR1000C SDK に含まれる
エミュレータ	E2 エミュレータ	-
ユーザ I/F ケーブル	ユーザ I/F ケーブル	E2 エミュレータ本体付属
デバッグ I/F	JTAG I/F	LPD I/F は不可
ユーザシステム	RH850/U2B Piggyback Board 等	DFP 搭載 RH850/U2B シリーズデバイス搭載

2.2 ソフトウェア構成

DFP搭載RH850/U2BシリーズデバイスでG4MHとDFPを連動デバッグするためのソフトウェア構成を表 2-2に示します。

表 2-2 ソフトウェア構成一覧

名称	製品名	備考
RH850 搭載 G4MH 用 エミュレータデバッガ	ルネサス製 GS+デバッガ	-
RH850 搭載 DFP 用 エミュレータデバッガ	NSITEXE 製 デバッガ/IDE	NSITEXE 社製 DR1000C SDK に含まれる IDE、OpenOCD を使用
仮想 OS 環境	Oracle VM Virtual Box	DR1000C デバッガ/IDE を Linux で使用するため Ubuntu 20.04 LTS 環境を使用

3. G4MH と DFP 連動デバッグ環境セットアップ

G4MH と DFP を連動デバッグする環境のセットアップについて説明します。

3.1 RH850/U2B シリーズデバイスのセットアップ

RH850/U2B シリーズデバイスの詳細については RH850/U2B Group User' s Manual を参照してください。本項では G4MH と DFP の連動デバッグするに当たってのセットアップ内容と注意事項を説明します。

3.1.1 DFP 関連オプションバイト

RH850/U2B シリーズデバイスに搭載された DFP をデバッグする場合には、以下のオプションバイトを設定する必要があります。

- S_OPBT0 : MPSELECT (DFP Chain mode に設定すること)

また、パワーオン時の DFP の状態およびリセットベクタを以下のオプションバイトで設定する必要があります。

- OPBT38 : DFP_init_boothart : 0000 0000_H (設定値)
- OPBT39 : DFP_resetvec : 0030 0000_H (設定値)

3.1.2 RH850/U2B シリーズデバイス搭載 DFP のアドレスマップ

RH850/U2B シリーズデバイスに搭載された DFP をデバッグするに当たってのアドレスマップを説明します。

表 3-1 DFP アドレスマップ

Start Address	End Address	Area
0030 0000 _H	003F FFFF _H	External ROM
F000 0000 _H	F000 7FFF _H	CCU Local Memory
F004 0000 _H	F004 1FFF _H	SPU Local Memory0
F004 4000 _H	F004 5FFF _H	SPU Local Memory1
F004 8000 _H	F004 9FFF _H	SPU Local Memory2
F004 C000 _H	F004 DFFF _H	SPU Local Memory3
F005 0000 _H	F005 005F _H	Mutual Exclusion Register Valiable (MERV)
F010 0000 _H	F013 FFFF _H	VLM Local Memory
F080 0000 _H	F0BF FFFF _H	DFP Peripheral
F800 0000 _H	F81F FFFF _H	DFP DEBUG
FE04 0000 _H	FE07 FFFF _H	External RAM

3.2 CS+デバッガのセットアップ

CS+デバッガの詳細についてはCS+ オンラインヘルプを参照してください。本項ではG4MHとDFPの連動デバッグするに当たってのセットアップ内容と注意事項を説明します。

3.2.1 G4MHとDFP連動デバッグ時の設定

RH850/U2B シリーズデバイスに搭載されたDFPをデバッグする場合には、CS+デバッガのE2エミュレータプロパティを設定する必要があります。また、G4MHとDFPを連動デバッグするには追加の設定をする必要があります。設定内容を表3-3および表3-4に、参考画面を図3-1および図3-2に示します。

表 3-2 DFPをデバッグする場合のCS+デバッガ E2エミュレータプロパティ設定

設定項目	設定内容
通信方式	JTAG
DFPをデバッグする	はい
サーバのIPアドレス	OpenOCDで指定したGDBサーバのIPアドレス 例：192.168.56.1
サーバのポート番号	OpenOCDで指定したGDBサーバのポート番号 例：9824

表 3-3 G4MHとDFPの連動デバッグする場合のCS+デバッガ E2エミュレータプロパティ設定

設定項目	設定内容
DFP連動Resume/連動Halt機能を使用する	はい
DFP実行要求待ちタイムアウト時間[s]	G4MH実行要求後、DFP実行要求を出すまでの タイムアウト時間 例：20
停止時に周辺エミュレーションを停止する	はい



図 3-1 G4MHとDFPの連動デバッグのCS+ E2エミュレータプロパティ設定参考(接続用設定タブ)

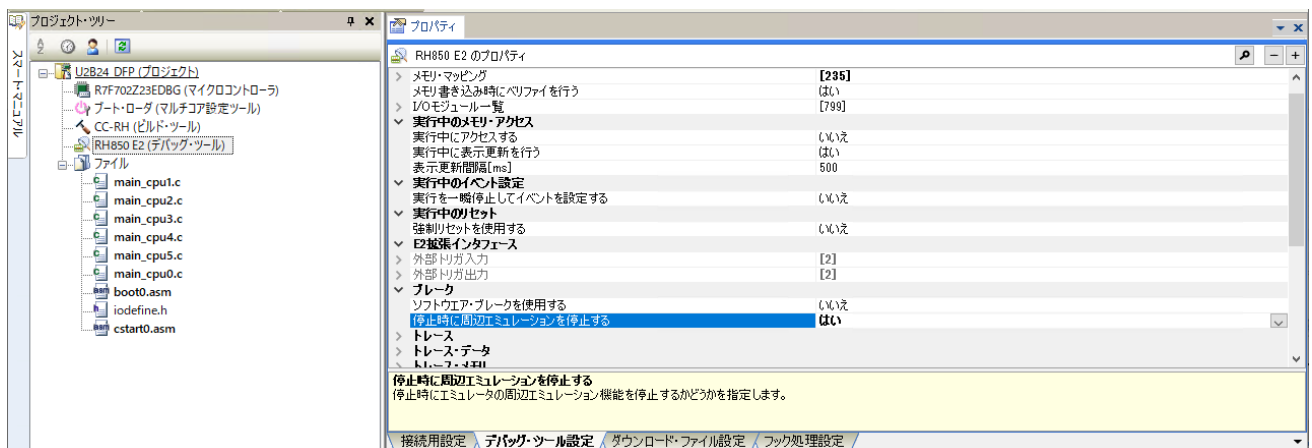


図 3-2 G4MHとDFPの連動デバッグのCS+ E2エミュレータプロパティ設定参考(デバッグ・ツール設定タブ)

3.3 Virtual Box のセットアップ

DFP のデバッガである DR1000C デバッガ/IDE を Linux 環境で使用するため、Virtual Box での仮想 Linux 上で DR1000C デバッガ/IDE をセットアップします。

3.3.1 Virtual Box のインストール

Oracle 社のホームページより Oracle VM Virtual Box のインストーラをダウンロードします。インストーラを起動し、インストーラに従って標準インストールします。

3.3.2 Virtual Box の起動と仮想ハードディスクの設定

Virtual Box を起動します。まず[ツール]→[環境設定]→[一般]で仮想環境用の仮想ハードディスク(以下仮想 HD)の格納先を指定します。その後、[ツール]→[新規]で仮想 HD を作成します。この時の推奨設定を表 3-5 に、参考画面を図 3-3~図 3-4 に示します。

表 3-4 G4MHとDFPの連動デバッグする場合のVirtual Box 仮想HD推奨設定

設定項目	設定内容
メモリーサイズ	4GB (デフォルトでは 1GB)
ファイルサイズ	30GB (デフォルトでは 10GB だが足りないため)
ハードディスクのファイルタイプ	VHD
物理ハードディスクにあるストレージ	固定サイズ

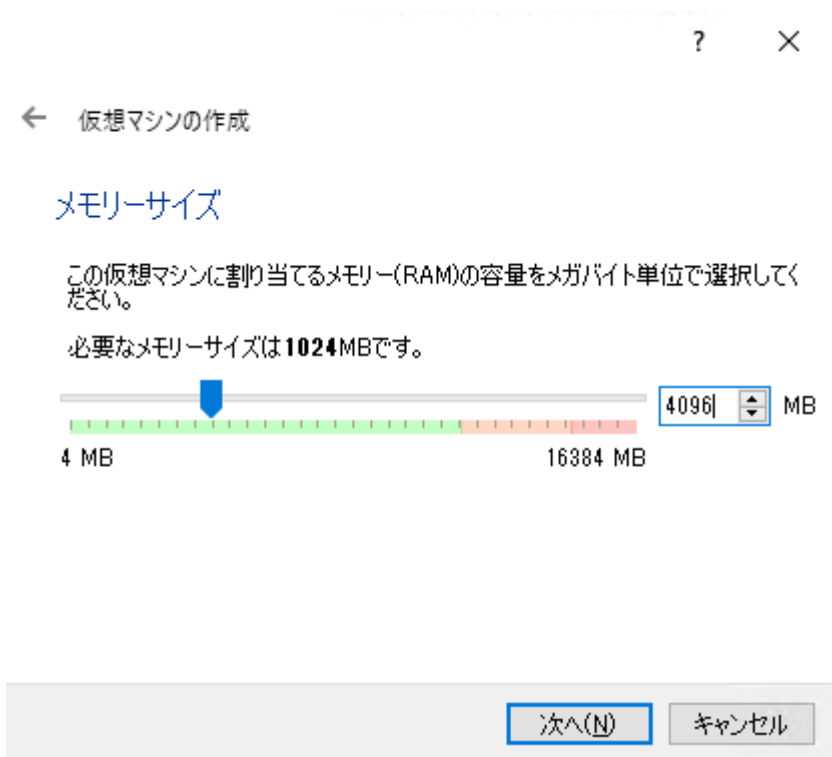


図 3-3 Virtual Box 仮想 HD 推奨設定参考 (メモリーサイズ)

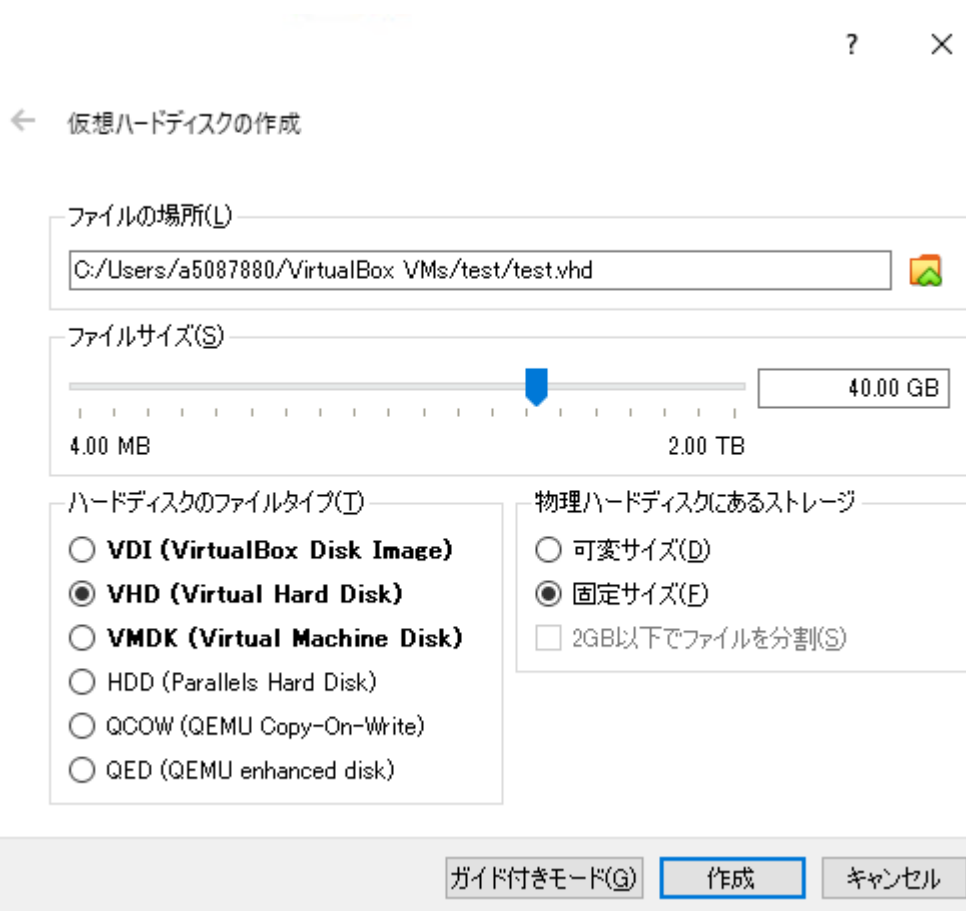


図 3-4 Virtual Box 仮想 HD 推奨設定参考 (仮想 HD 作成時のエキスパートモード画面)

3.3.3 ubuntu イメージファイルの入手

ubuntu のホームページより ubuntu 20.04 LTS の ISO イメージをダウンロードします。

3.3.4 仮想マシンの作成準備

作成した仮想 HD を選択し、[設定]→[ストレージ]を選択、ストレージデバイスの空となっているディスクアイコンをクリックします。属性項目の光学ドライブ選択欄で、ディスクアイコンをクリックするとディスクファイルを選択が出るためクリックします。そこで ubuntu のイメージファイル(iso)をマウントし、OK ボタンをクリックします。参考画面を図 3-5 に示します。

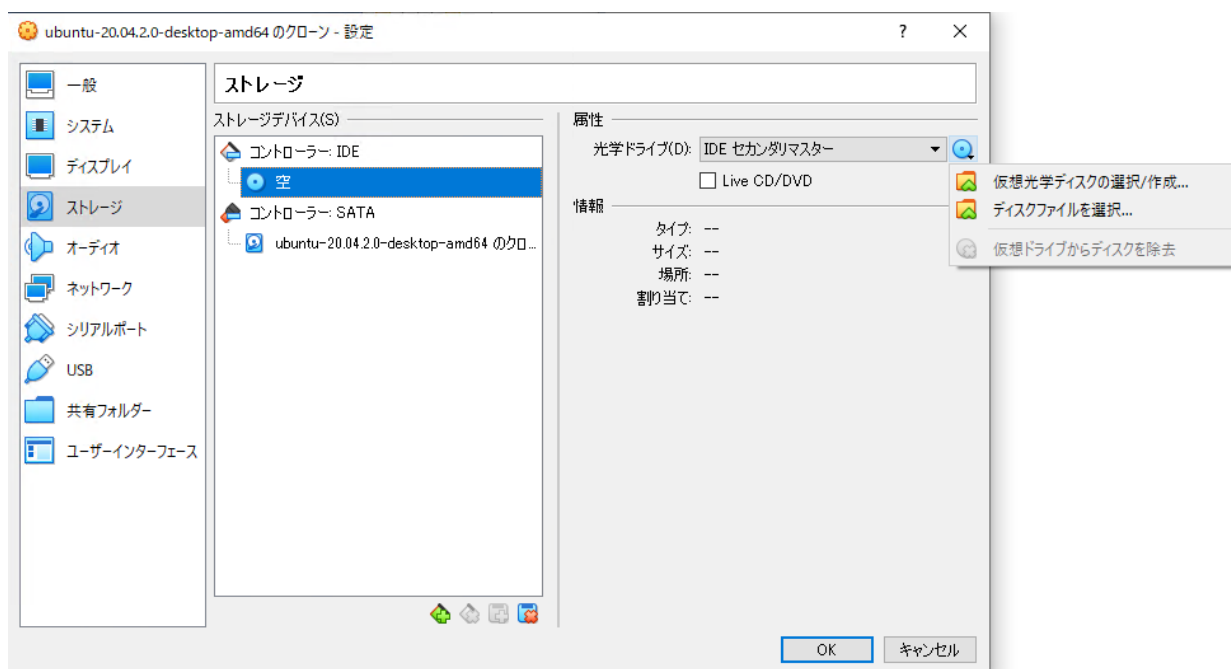


図 3-5 Virtual Box 仮想マシンのイメージファイルマウント参考

3.3.5 ubuntu のインストール

Virtual Box 上から[起動]をクリックすると、ubuntu のインストールを開始します。仮想環境上で言語を選択し、[Ubuntu をインストール]をクリックします。以降、インストール画面に従ってインストールを進めます。

インストール完了時点での参考画面を図 3-6 に示します。インストール完了後は仮想環境を再起動し、ユーザ名とパスワードを入力して仮想環境にログインします。ログイン後、ソフトウェアアップデートを実施し、仮想環境を再起動します。

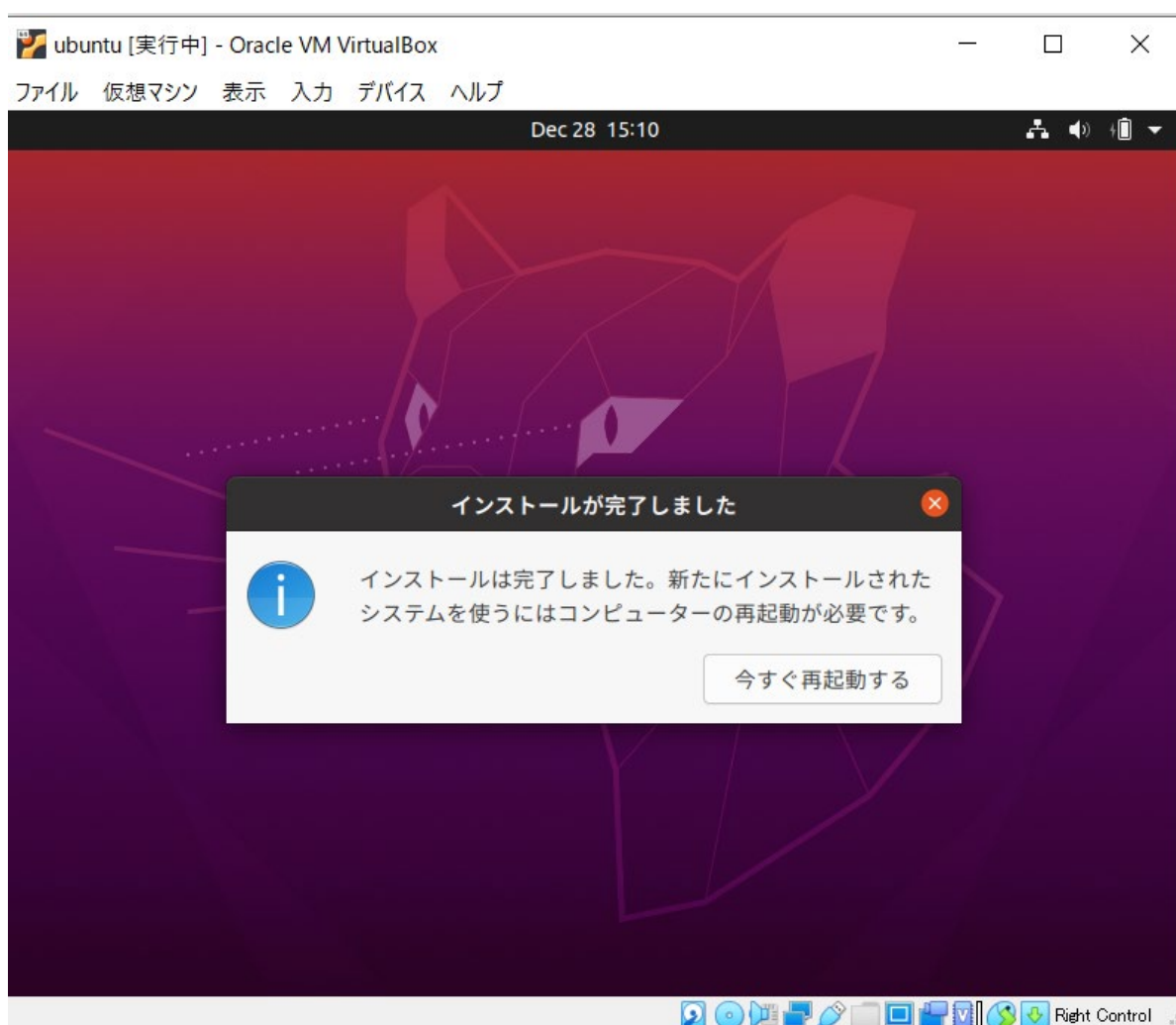


図 3-6 Virtual Box 仮想マシン上 ubuntu インストール完了参考

3.3.6 仮想環境の構築

[端末で開く]でターミナルを開き、ubuntu に以下のソフトをインストールします。

- build-essential : GCC, MAKE, GDB 等を使用可能にするため

3.3.7 PC と仮想マシンの接続設定

PC (Windows) と仮想マシン (ubuntu) を接続するため、PC と仮想マシンの IP を確認し、Virtual Box に設定します。設定箇所は以下となります。設定内容を表 3-5 に、参考画面を図 3-7 に示します。

[設定]→[ネットワーク]→[高度]を展開→[ポートフォワーディング]→[ルール追加]

表 3-5 PC と仮想マシンの接続設定内容

設定項目	設定内容
名前	任意
プロトコル	TCP
ホスト IP	PC の IPv4 アドレス
ホストポート	ゲストポート番号や予約・使用済みと 被らないポート番号
ゲスト IP	仮想マシンの IPv4 アドレス
ゲストポート	予約・使用済みと被らないポート番号

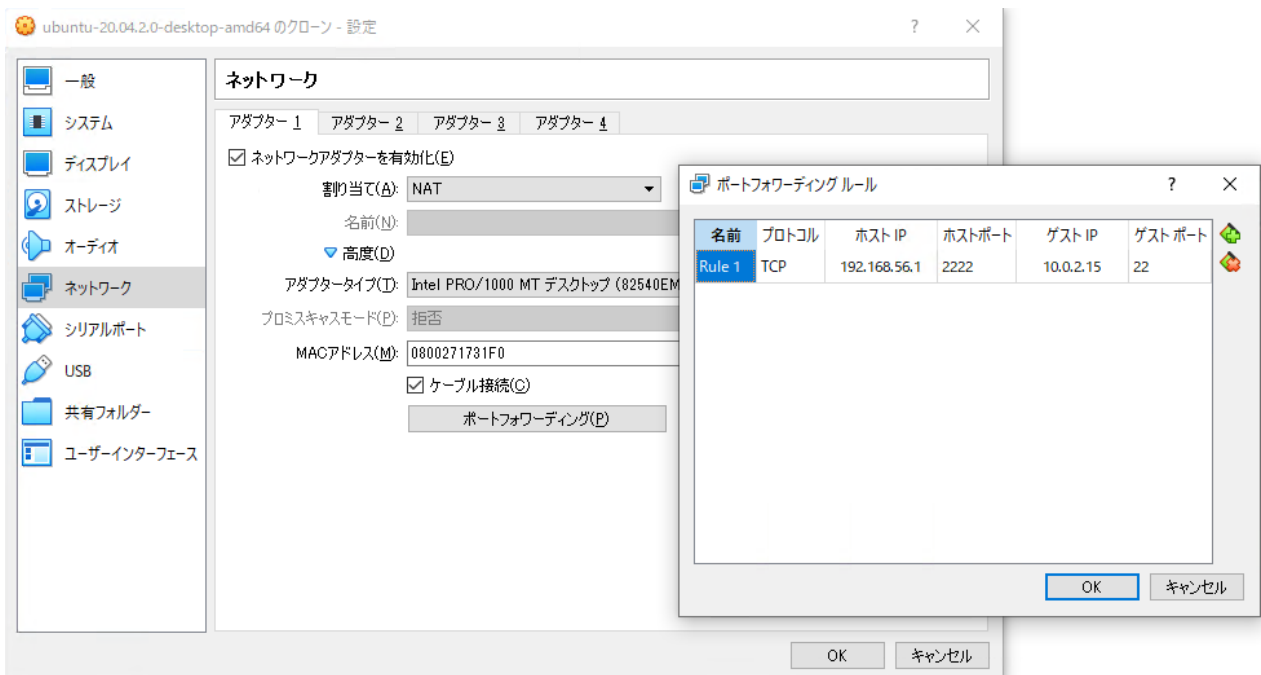


図 3-7 Virtual Box PC と仮想マシンの接続設定参考

3.4 OpenOCD のセットアップ

OpenOCD の詳細については NSITEXE DR1000C Debugger User Manual を参照してください。本項では G4MH と DFP の連動デバッグするに当たってのセットアップ内容と注意事項を説明します。

なお、OpenOCD は次期アップデートプレリリースである 1.1.1 を使用してください。

3.4.1 IP アドレス、ポート番号、デバッグ対象の hart 設定

OpenOCD に設定する IP アドレスおよびポート番号は、CS+デバッガで指定した IP アドレスおよびポート番号と合わせてください。IP アドレスおよびポート番号が異なる場合、DFP をデバッグできません。

以下は 5 つの hart および 9 つの hart で DFP をデバッグするときの OpenOCD コンフィグファイルの記載例となります。必要に応じて、デバッグ対象の hart の指定を変更してください。デバッグ対象の hart ID は 0 および 16~31 から選択可能です。ただし、同時にデバッグ可能な最大 hart 数は 3.5.1 に記載した数になります。また、hart0 は必ずデバッグ対象としてください。9 つの hart でデバッグするコンフィグファイルの例に、5 つの hart でデバッグする設定との差分を記載しますので、デバッグ対象の hart の指定を変更する場合の参考にしてください。

OpenOCD コンフィグファイル (5hart デバッグの例)

```
# Debug Adapter Configuration
interface remote_bitbang
remote_bitbang_host 192.168.56.1 ★ CS+デバッガで指定した[サーバの IP アドレス] を指定します
remote_bitbang_port 9824 ★ CS+デバッガで指定した[サーバのポート番号] を指定します

# Tap Declaration
set _CHIPNAME riscv
set _DAP_TAPID 0x100039df
set _ENDIAN little

jtag newtap $_CHIPNAME dap -irlen 5 -ircapture 0x01 -irmask 0x03 -expected-id $_DAP_TAPID

set _TARGETNAME $_CHIPNAME.cpu0
set _TARGETNAME_16 $_CHIPNAME.cpu16
set _TARGETNAME_20 $_CHIPNAME.cpu20
set _TARGETNAME_24 $_CHIPNAME.cpu24
set _TARGETNAME_28 $_CHIPNAME.cpu28

target create $_TARGETNAME riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 0 -rtos
hwthread

target create $_TARGETNAME_16 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 16
```

```

target create $_TARGETNAME_20 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 20
target create $_TARGETNAME_24 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 24
target create $_TARGETNAME_28 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 28

target smp $_TARGETNAME $_TARGETNAME_16 $_TARGETNAME_20 $_TARGETNAME_24
$_TARGETNAME_28

$_TARGETNAME configure -event gdb-attach { halt }

riscv set_reset_timeout_sec 1200
riscv set_command_timeout_sec 1200
riscv set_enable_virt2phys off

puts "Before init"
# Server Configuration
gdb_port 3333                ★GDBからの接続待ちを行うポート番号
gdb_report_data_abort enable
poll_period 500
init

# General Commands
bindto 0.0.0.0
targets $_TARGETNAME

```

OpenOCD コンフィグファイル (9hart デバッグの例)

```

# Debug Adapter Configuration
interface remote_bitbang
remote_bitbang_host 192.168.56.1
remote_bitbang_port 9824

# Tap Declaration
set _CHIPNAME riscv
set _DAP_TAPID 0x100039df
set _ENDIAN little

jtag newtap $_CHIPNAME dap -irlen 5 -ircapture 0x01 -irmask 0x03 -expected-id $_DAP_TAPID

set _TARGETNAME $_CHIPNAME.cpu0
set _TARGETNAME_16 $_CHIPNAME.cpu16
set _TARGETNAME_17 $_CHIPNAME.cpu17 ★ 5hart との差分 : hart 17 を追加
set _TARGETNAME_20 $_CHIPNAME.cpu20
set _TARGETNAME_21 $_CHIPNAME.cpu21 ★ 5hart との差分 : hart 21 を追加

```

```
set _TARGETNAME_24 $_CHIPNAME.cpu24
set _TARGETNAME_25 $_CHIPNAME.cpu25 ★ 5hart との差分 : hart 25 を追加
set _TARGETNAME_28 $_CHIPNAME.cpu28
set _TARGETNAME_29 $_CHIPNAME.cpu29 ★ 5hart との差分 : hart 29 を追加

target create $_TARGETNAME riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 0 -rtos
hwthread

target create $_TARGETNAME_16 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 16
target create $_TARGETNAME_17 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 17 ★
5hart との差分 : hart 17 を追加
target create $_TARGETNAME_20 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 20
target create $_TARGETNAME_21 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 21 ★
5hart との差分 : hart 21 を追加
target create $_TARGETNAME_24 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 24
target create $_TARGETNAME_25 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 25 ★
5hart との差分 : hart 25 を追加
target create $_TARGETNAME_28 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 28
target create $_TARGETNAME_29 riscv -endian $_ENDIAN -chain-position $_CHIPNAME.dap -coreid 29 ★
5hart との差分 : hart 29 を追加

target smp $_TARGETNAME $_TARGETNAME_16 $_TARGETNAME_20 $_TARGETNAME_24
$_TARGETNAME_28 $_TARGETNAME_17 $_TARGETNAME_21 $_TARGETNAME_25
$_TARGETNAME_29 ★ 5hart との差分 : hart 17、21、25、29 を追加

$_TARGETNAME configure -event gdb-attach { halt }

riscv set_reset_timeout_sec 1200
riscv set_command_timeout_sec 1200
riscv set_enable_virt2phys off

puts "Before init"
# Server Configuration
gdb_port 3333
gdb_report_data_abort enable
poll_period 500
init

# General Commands
bindto 0.0.0.0
targets $_TARGETNAME
```

3.4.2 OpenOCD 起動

仮想環境 (ubuntu) から” 端末” を開いて以下のコマンドを実行します。

```
$ openocd -f (OpenOCD コンフィグファイル)
```

3.4.3 OpenOCD 起動時の注意事項

OpenOCD を起動する時は、CS+デバッガで G4MH デバッグを開始した状態（「デバッグ・ツールへ接続」した状態）であることを確認してください。CS+デバッガが G4MH デバッグを開始する前に OpenOCD を起動した場合、失敗となります。

3.5 NSI IDE のセットアップ

NSI IDE の詳細については NSITEXE DR1000C IDE User Manual を参照してください。本項では G4MH と DFP の連動デバッグするに当たってのセットアップ内容と注意事項を説明します。

3.5.1 デバッグ可能 hart 数

RH850/U2B 搭載 DFP をデバッグする時、デバッグ可能な hart は 9 つまでとなります。

3.5.2 DFP プログラム

NSI IDE のプロジェクト新規作成時に、対象ボードを指定する項目で [RH850U2B] を設定します。

新規作成されたプロジェクトをビルドすると DFP プログラム (*.mot) ファイルが生成されます。

NSI IDE の新規作成、ビルド手順については NSITEXE DR1000C IDE User Manual を参照してください。

生成した DFP プログラム (*.mot) を連動デバッグするために、CS+デバッガから以下の手順でダウンロードするファイルとして指定します。設定箇所は以下となります。参考画面を図 3-8 に示します。

[プロジェクトツリー]→[RH850 E2(デバッグ・ツール)]→[ダウンロード・ファイル設定]
→[ダウンロードするファイル]→[[1]]

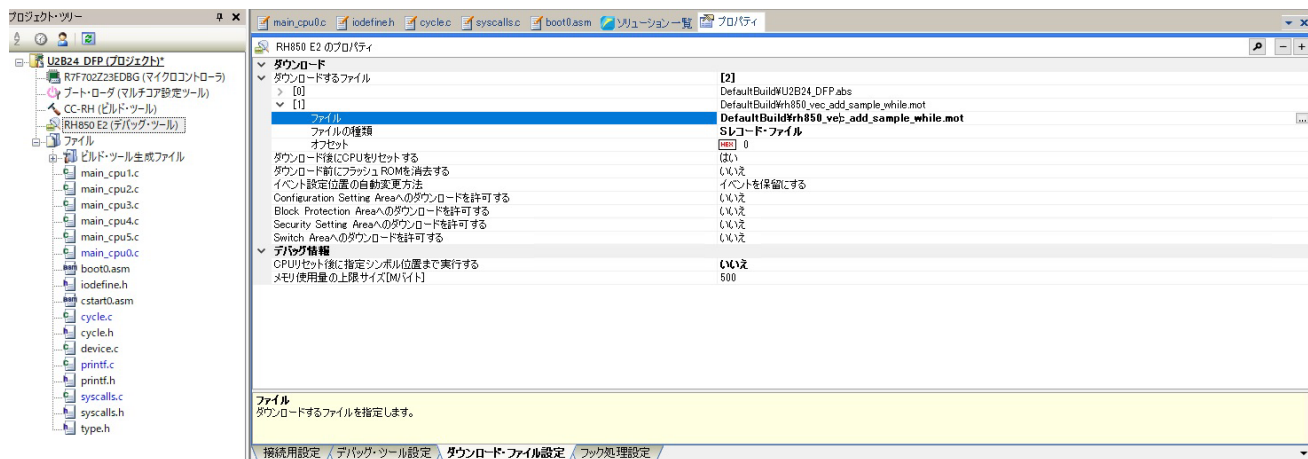


図 3-8 G4MHとDFPの連動デバッグのCS+ E2エミュレータプロパティ設定参考（ダウンロード・ファイル設定）

3.5.3 OpenOCD-GDB 間の接続

NSI IDE は OpenOCD を起動したサーバの IP アドレスと GDB からの接続待ちを行うポート番号を指定して OpenOCD との接続を行います。

設定画面の表示方法は以下となります。設定内容を表 3-6 に、参考画面を図 3-9 に示します。

- 1 NSI IDE の Project Explorer ビューから対象のプロジェクトを右クリック
→[Debug As]→[Debug Configurations...] で Debug Configurations ウィンドウを表示
- ② [GDB Hardware Debugging]→[(プロジェクト名)_Debug_TCPIP]→[Debugger]

表 3-6 GDBとOpenOCDの接続設定内容

設定項目	設定内容
JTAG Device	OpenOCD (via socket)
Host name or IP address	OpenOCD を起動したサーバの IP アドレス NSI IDE を起動したサーバと同じであれば “localhost” の指定も可能
Port number	OpenOCD で指定した GDB からの接続待ちを行うポート番号

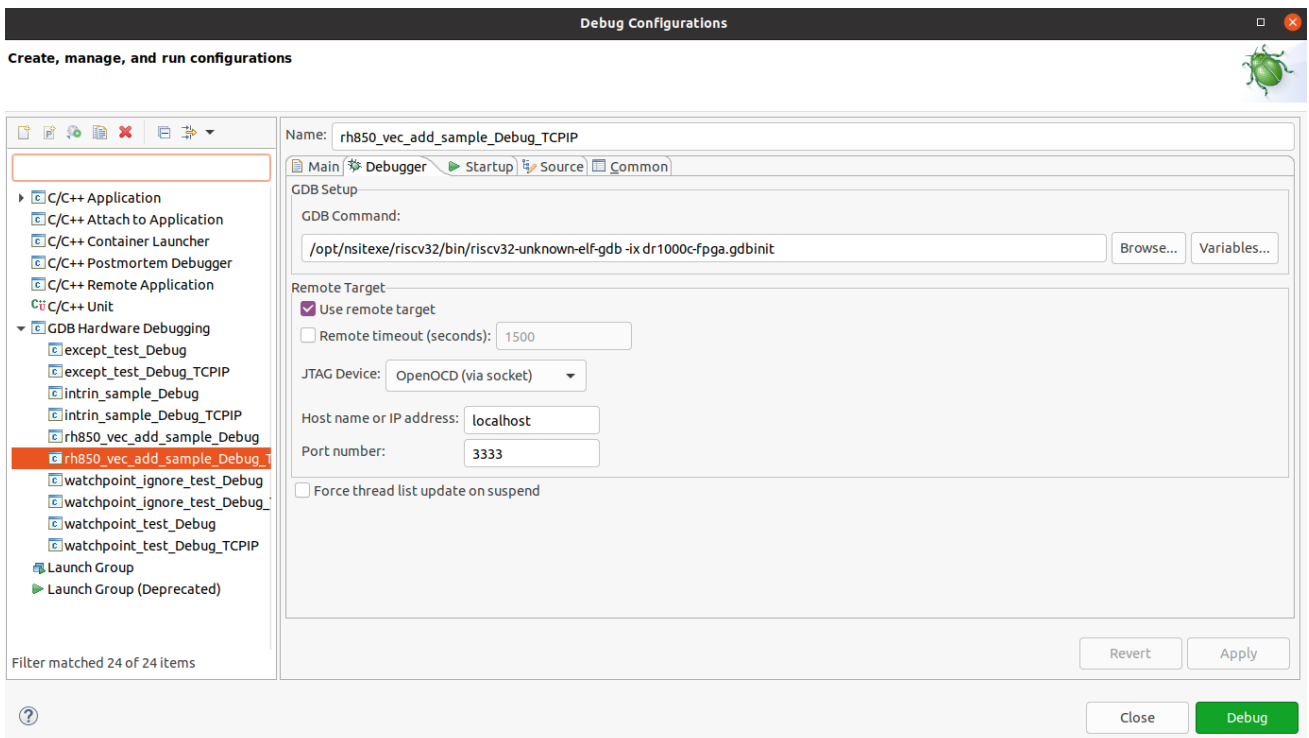


図 3-9 NSI IDE Debug Configurations設定参考(Debugger)

3.5.4 DFP デバッグの開始

OpenOCD-GDB 間の接続の設定を行ったあと、同画面 (Debug Configurations ウィンドウ) から左下の [Debug] をクリックすることでデバッグが開始されます。

デバッグ開始前に、画面から以下の設定を確認してください。

設定画面の表示方法は以下となります。設定内容を表 3-7 に、参考画面を図 3-10 に示します。

Debug Configurations ウィンドウから、[GDB Hardware Debugging] → [(プロジェクト名)_Debug_TCPIP] → [Main]

表 3-7 プロジェクトデバッグの設定内容

設定項目	設定内容
C/C++ Application	<p>ELF ファイル</p> <p>NSI IDE は ELF ファイルとして、.run ファイルを生成する。 .run ファイルは DFP プログラム生成時に合わせて生成される。 この項目の設定では、.mot ファイルではなく .run ファイルを設定することに注意。</p>

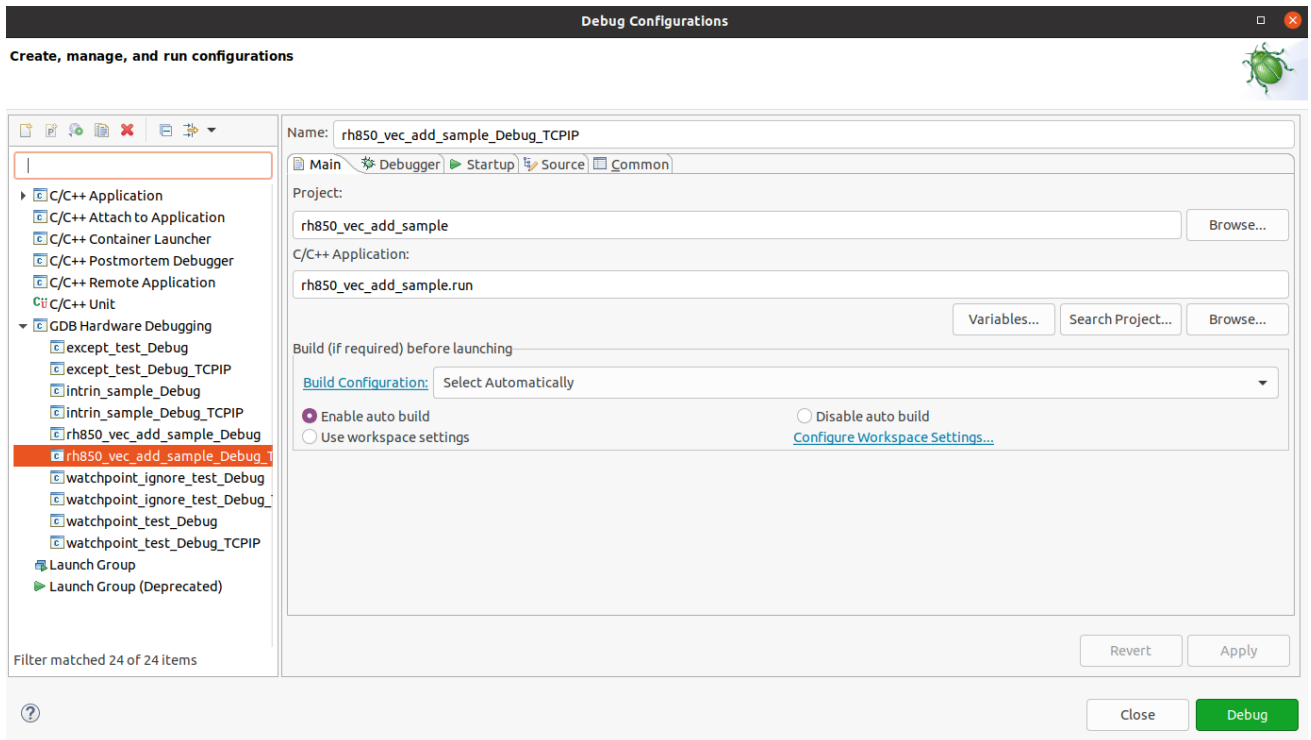


図 3-10 NSI IDE Debug Configurations 設定参考 (Main)

3.6 G4MH と DFP の連動デバッグシステム立ち上げ手順

G4MH と DFP の連動デバッグ環境を立ち上げる手順について説明します。

- ① DFP 搭載 RH850/U2B シリーズデバイスのオプションバイトを設定 (参照 : 3.1.1)
- ② CS+デバッガで G4MH と DFP 連動デバッグ設定 (参照 : 3.2.1)
- ③ CS+デバッガでデバッグする DFP プログラムを設定 (参照 : 3.5.2)
- ④ CS+デバッガで「デバッグ・ツールへ接続」を行い、G4MH のデバッグを開始
- ⑤ DR1000C デバッガ/IDE を Linux 環境で使用する場合、Virtual Box で ubuntu 環境を起動
- ⑥ DR1000C デバッガ/IDE の OpenOCD で IP アドレスとポート番号を CS+デバッガの指定値と同じに設定 (参照 : 3.4.1)
- ⑦ DR1000C デバッガ/IDE の OpenOCD を起動 (参照 : 3.4.2)
- ⑧ DR1000C デバッガ/IDE の NSI IDE を起動し、OpenOCD-GDB 間の接続設定 (参照 : 3.5.3)
- ⑨ DR1000C デバッガ/IDE の NSI IDE から DFP デバッグを開始 (参照 : 3.5.4)

以上で CS+デバッガで G4MH デバッグ、NSI IDE で DFP デバッグできる状態になります。

4. G4MH と DFP 連動デバッグ手法

4.1 G4MH と DFP 連動実行

G4MH と DFP を連動実行させてデバッグする操作方法と注意事項について説明します。

4.1.1 G4MH と DFP 連動実行手順

G4MH と DFP を連動デバッグするとしており、G4MH と DFP が共にブレーク (DB-Halt) 状態の時、G4MH と DFP を同時に連動実行させるデバッグ操作手順は以下になります。

- ① CS+デバッガで「実行」をクリック
- ② DFP 実行要求待ちタイムアウト時間以内に NSI IDE で「Resume」をクリック

4.1.2 G4MH と DFP 連動実行の注意事項

No.1 NSI IDE の「Resume」のみをクリック

G4MH と DFP を連動デバッグする時、CS+デバッガで「実行」をクリックせず、NSI IDE の「Resume」をクリックした場合、G4MH はブレーク状態のまま、DFP のみプログラム実行を開始します。

DFP のみプログラム実行状態から、NSI IDE の「Suspend」クリックや、ブレークポイントでの DFP Halt によって DFP は DB-Halt 状態に遷移します。この状態から G4MH と DFP 連動実行手順を行った場合、G4MH と DFP は連動実行し、G4MH も DFP もプログラム実行を開始します。

No.2 DFP 実行状態での CS+デバッガの「実行」クリック

G4MH と DFP を連動デバッグする時、DFP がプログラム実行中に CS+デバッガで「実行」をクリックした場合、エラーとなります。この時、G4MH はプログラム実行を開始しません。

No.3 CS+デバッガで「実行」クリック後、NSI IDE で「Resume」をクリックしない

G4MH と DFP を連動デバッグする時、CS+デバッガで「実行」をクリックした後、NSI IDE の「Resume」がクリックされず、DFP 実行要求待ちタイムアウト時間を過ぎた場合、エラーとなります。この時、G4MH はプログラム実行を開始しません。

4.2 G4MH と DFP 連動ブレーク

G4MH と DFP を連動ブレークさせてデバッグする操作方法と注意事項について説明します。

4.2.1 G4MH と DFP 連動ブレーク動作

G4MH と DFP を連動デバッグするとしており、G4MH と DFP が連動実行している状態の時、以下のようなブレークによって G4MH または DFP がブレーク (Halt) した場合、G4MH と DFP は連動ブレークし G4MH と DFP は共にブレーク (Halt) します。

- CS+デバッガの「停止」をクリック (G4MH 強制ブレーク)
- NSI IDE の「Suspend」をクリック (DFP 強制 Halt)
- CS+デバッガで設定したブレークポイントで G4MH がブレーク (G4MH イベントブレーク)
- NSI IDE で設定したブレークポイントで DFP がブレーク (DFP イベントブレーク)

5. 問い合わせ先

DFP 搭載 RH850/U2B シリーズデバイスをデバッグする時、トラブルが発生した場合の問い合わせ先は以下になります。どちらに問い合わせれば良いかわからない場合は、Renesas Electronics に問い合わせください。

- ・ CS+デバッガを使用した G4MH デバッグ中のトラブルや利用法に関して : Renesas Electronics
- ・ NSI IDE を使用した DFP デバッグ中のトラブルや利用法に関して : NSITEXE

Rev.	発行日	改訂内容	
		ページ	ポイント
0.50	2022.04.25	-	初版
1.00	2022.07.20	17	3.4.1 タイトルを修正
		17	3.4.1 9hart デバッグの説明を追加
		19	3.5.1 デバッグ可能 hart 数を 9 つに修正

E2エミュレータ ユーザーズマニュアル別冊
(RH850/U2BシリーズDFP連動デバッグのセットアップ)

発行年月日 2022年07月20日 Rev. 1.00

発行 ルネサス エレクトロニクス株式会社
〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

E2エミュレータ ユーザーズマニュアル別冊

(RH850/U2BシリーズDFP連動デバッグのセットアップ)