Toradex Linux C言語アプリケーション 開発マニュアル

本マニュアルは岡本無線電機株式会社が独自作成したものでありメーカーが保証した内容ではありません。万が一本マニュア ルに間違いがあり、事故が生じたとしても岡本無線電機株式会社は一切の責任を問われないものとさせていただきます。



Ver 2.00

変更履歴

バージョン	更新日付	変更内容
2.00	2020/09/07	BSP3.0.4向けに作成

本マニュアルについて

本マニュアルはトラデックスのCPUモジュール上で動作するC/C++言語アプリケーションを作成する手順を記述しています。

参考;

http://developer.toradex.com/knowledge-base/board-support-package/openembedded-(core) https://developer.toradex.com/knowledge-base/linux-sdks

<u>1.実行環境</u>

本マニュアルの実行環境は下記です。

仮想化ソフト: VMWARE Player v15.5.6 Host OS: Windows 10 1909 Guest OS: Ubuntu Desktop 18.04LTS 64bit(英語版) BSP: v3.0.4 CPUモジュール: Colibri-iMX7D 512MB V1.1C キャリアボード: Colibri 評価ボード Rev 3.2 + アクセサリーキット

本マニュアルとは異なるモジュールや評価ボード以外のキャリアボードを使われても大雑把には同じ操作となります。

インターネット接続環境が必要になります。

<u>2.事前準備</u>

本マニュアルはLinux OSイメージ開発マニュアルの内容をすべて終えた状態で進めています。

3. 前提知識

Linux OSイメージ開発マニュアルの内容をご理解いただいた状態を前提としています。

4.注意点

オープンソース系を利用した開発に共通することですがすべてを理解しようとするときりがなく開発効率を損ないます。必要なタ イミングで必要な知識を身につけるというスタンスで理解することを推奨いたします。

開発環境と実行環境の違いをわかりやすくするためにコマンドの表記の前に下記をつけています。 開発環境(PC)上で入力するコマンド: [ubuntu]\$ 実行環境(モジュール)上で入力するコマンド: [colibri-imx7]#

コピーについて

本マニュアル内のコマンドなどをコピーした場合、改行が入ったり「-」が抜けてしまうことがあるのでご注意ください。一度テキスト エディタなどに張り付けてコピーした内容をご確認ください。

SDKの作成

C言語開発を行うためにOpen Embeddedでターゲットイメージ向けのSDKを作成します。

[ubuntu]\$ cd /work/oe-core

[ubuntu]\$. export

[ubuntu]\$ bitbake console-tdx-image -c populate_sdk

複数のモジュールでSDKを作成している時は下記のようなSDKがすでに存在するというエラーが出るときがあります。

sdkを一度削除してから再度実行してください。

[ubuntu]\$ rm -rf /work/oe-core/build/deploy/sdk/*

user1@ubuntu: /work/oe-core/build	
File Edit View Search Terminal Help	
Initialising tasks: 100% ###################################	****
<pre>ERROR: console-tdx-image-3.0b4-r0 do_populate_sdk: The recipe console-tdx-image is trying to interest into a shared area when those files already exist. Those files and their manifest location as /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-Console-Image-armv7at2hf-neon-toolchain-2 (matched in manifest-colibri_imx6ullx86_64-console-tdx-image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-Console-Image-armv7at2hf-neon-toolchain-2 manifest (matched in manifest-colibri_imx6ullx86_64-console-tdx-image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-Console-Image.armv7at2hf-neon-toolchain-2 manifest (matched in manifest-colibri_imx6ullx86_64-console-tdx-image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-Console-Image.armv7at2hf-neon-toolchain-2 et.manifest (matched in manifest-colibri_imx6ullx86_64-console-tdx-image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-Console-Image.armv7at2hf-neon-toolchain-2 et.manifest (matched in manifest-colibri_imx6ullx86_64-console-tdx-image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-console-Image.populate_sdk) Please verify which recipe should provide the above files.</pre>	stall fil e: 6:4.sh 6.4.host 6.4.targ 6:4.test
The build has stopped, as continuing in this scenario WILL break things - if not now, possibly i ture (we've seen builds fail several months later). If the system knew how to recover from this cally it would, however there are several different scenarios which can result in this and we do which one this is. It may be you have switched providers of something like virtual/kernel (e.g	in the fu automati on't know . from li [

/work/oe-core/build/deploy/sdk配下にSDKが出力されます。

実行してSDKを展開します。(モジュールやBSPのバージョンによってシェルの名前が変わります。) [ubuntu]\$ /work/oe-core/build/deploy/sdk/tdx-x11-glibc-x86_64-Console-Image-armv7at2hf-neon-toolchain-2.6.4.sh

下記のようにSDKのインストールディレクトリを問われるので入力します。 [ubuntu]\$ Enter target directory for SDK (default: /opt/tdx-x11/2.6.4): 本マニュアルではデフォルトの/opt/tdx-x11/2.6.4にインストールしますのでそのままEnterキーを押します。

下記のように問われますので [ubuntu]\$ You are about to install the SDK to "/opt/tdx-x11/2.6.4". Proceed[Y/n]? Yを入力してEnterキーを押します。 管理者権限で実行しますので途中でパスワードを問われます。パスワードを入力してください。

すでに存在していた場合は下記のように上書きするかどうかを問われますがその場合は一度シェルを停止してシェル実行前に ディレクトリを削除しておいたほうが良いです。 If you continue, existing files will be overwritten! Proceed[y/N]?

sudo rm -rf /opt/tdx-x11/2.6.4

最後に下記のような環境変数設定シェルのパスの案内があります。このシェルは後の工程で使用します。 . /opt/tdx-x11/2.6.4/environment-setup-armv7at2hf-neon-tdx-linux-gnueabi

us	er1@ubuntu: /work/oe-core/build	000
File Edit View Search Terminal H	elp	
user1@ubuntu:/work/oe-core/bu 6_64-Console-Image-armv7at2hf TDX X11 SDK installer version	<pre>ild\$ /work/oe-core/build/deploy/sdk/tdx-x11-gl -neon-toolchain-2.6.4.sh 2.6.4 ======</pre>	ibc-x8
Enter target directory for SD You are about to install the Extracting SDK Setting it updone SDK has been successfully set Each time you wish to use the environment setup script e.g. \$. /opt/tdx-x11/2.6.4/envir user1@ubuntu:/work/oe-core/bu	<pre>K (default: /opt/tdx-x11/2.6.4): SDK to "/opt/tdx-x11/2.6.4". Proceed[Y/n]? Y done up and is ready to be used. SDK in a new shell session, you need to source onment-setup-armv7at2hf-neon-tdx-linux-gnueabi ild\$</pre>	e the

Eclipseのインストール

作業ディレクトリ作成 [ubuntu]\$ mkdir -p /work/app && cd /work/app

Eclipse起動にはJavaが必要です。Javaをインストールします。 [ubuntu]\$ sudo apt-get -y install openjdk-11-jre

Eclipse(202006バージョン)のサイトからを入手します。 (windows10側のブラウザなどで入手) https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2020-06/R/eclipse-cpp-2020-06-R-linux-gtk-x86_64.tar.gz

app配下にコピーします。

展開します。

[ubuntu]\$ tar -xf ./eclipse-cpp-2020-06-R-linux-gtk-x86_64.tar.gz

Eclipse実行シェル作成

Eclipse起動前にSDK用環境変数を定義する必要があります。またコンパイルオプションも長くなるためSDK用環境変数設定や コンパイルオプションを定義してからEclipseを実行するシェルを作成します。

シェル内にはSDKが出力した環境変数設定シェルの実行も行っています。(長い赤線部分)

(定義する内容は搭載するARMのアーキテクチャやBSPのバージョン、SDK出力ディレクトリなどによって変わります。赤線部分) [ubuntu]\$ cd /work/app/

[ubuntu]\$ gedit ./sdk_eclipse.sh

内容は下記です。

#!/bin/sh SDK_ARM_VER=<u>armv7-a</u> SDK_ARCH=cortex-a7

SDK_ENV_SET=/opt/tdx-x11/2.6.4/environment-setup-armv7at2hf-neon-tdx-linux-gnueabi

```
if [ ! -e ${SDK_ENV_SET} ]; then
    echo "not found: "${SDK_ENV_SET}
    exit
fi
. ${SDK_ENV_SET}
export SDK_INCLUDE=${SDKTARGETSYSROOT}/usr/include/
export SDK_OPTIMIZATION="-march=${SDK_ARM_VER} -fno-tree-vectorize -mthumb-interwork -mfpu=neon
-mfloat-abi=hard -Wno-poison-system-directories"
```

export SDK_LINKER="-mfloat-abi=hard -L\${SDKTARGETSYSROOT}/lib/ -WI,-rpath-link,\${SDKTARGETSYSROOT}/lib/ -L\${SDKTARGETSYSROOT}/usr/lib/ -WI,-rpath-link,\${SDKTARGETSYSROOT}/usr/lib/ --sysroot=\${SDKTARGETSYSROOT}"

./eclipse/eclipse

各々の設定が改行されていないかを確認してください。改行があるとうまくいきません。 geditで行番号を出す設定にするとわかりやすくなります。行番号を表示するにはgedit起動後画面左上のText Editorで Preferencesを開きます。

Activities	🗑 Text Editor 🔫	
٢	New Window	
• 💼	Op Preferences	sdk_eclipse.sh /work/app
. ≻	2 S 3 S 4 Keyboard Shortcuts 5 S Help	2.6.4/environment-setup-armv
9 2	6 7 i About 8 Quit]; then _ENV_SET}
	9 10 f 11 . \${SDK_ENV_SET} 12 export SDK INCLUDE=\${SDKT/	ARGETSYSROOT}/usr/include/
-	13 export SDK_OPTIMIZATION=" mfpu=neon -mfloat-abi=hard	-march=\${SDK_ARM_VER} -fno-t d -Wno-poison-system-directo

Display Line Numbersにチェックをいれて閉じます。



行番号が表示され各exportが一行で記述されているかどうかが分かりやすくなります。

Open 🔻	迅	sdk_eclipse.sh		Save		000
1 #!/bin/	/sh	(A MERCERE				
2 SDK ARM	VER=armv7-a					
3 SDK ARC	H=cortex-a7					
4						
5 SDK_ENV	/_SET=/opt/tdx-x11/2.6.4/environmer	nt-setup-armv7at2	hf-neon-tdx-linux-g	nueabi		
б						
7 if [!	<pre>-e \${SDK_ENV_SET}]; then</pre>					
8 echo	<pre>"not found: "\${SDK_ENV_SET}</pre>					
9 exit						
10 fi						
11 . \${SDH	(_ENV_SET}	entre service automatical entres of				
12 export	SDK_INCLUDE=\${SDKTARGETSYSROOT}/us	sr/include/				
13 export	SDK_OPTIMIZATION="-march=\${SDK_ARM	M_VER} -fno-tree-	vectorize -mthumb-i	.nterwor	'k -	
mfpu=ne	ion -mfloat-abi=hard -Wno-poison-sy	ystem-directories				
14						
15 export	SDK_LINKER="-mfloat-abi=hard -L\${	SDKTARGETSYSROOT }	/lib/ -Wl,-rpath-li	.nk,Ş	N COLOR	
{SDKTAF	<pre>{GETSYSROOT}/lib/ -L\${SDKTARGETSYSF</pre>	ROOT}/USF/lib/ -W	l,-rpath-link,\${SDK	TARGETS	YSRO	OT}/USF/
lib/	sysroot=\${SDKTARGETSYSROOT}"					
10						
17. Jecui	se/eclipse					
18						
		sh 🔻	Tab Width: 8 🔻 🛛 Lr	18, Col 1		INS

作成したファイルに実行権限を付与します。 [ubuntu]\$ chmod +x ./sdk_eclipse.sh

ワークスペースディレクトリ作成 [ubuntu]\$ mkdir ./workspace

Eclipse実行 [ubuntu]\$./sdk_eclipse.sh

以後Eclipseを起動する場合はこのシェルを使ってください。

Eclipseを実行すると下記のようなワークスペースのディレクトリのパスを指定するウィンドウが表示されますのでワークスペースのパスを設定します。本マニュアルでは/work/app/workspaceにしています。 LaunchをクリックしてEclipseを起動します。

Eclipse IDE Launcher

Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

<u>W</u> orkspace:	/work/app/workspace	•	Browse	
🗆 Use this a	s the defaul <mark>t and do</mark> not ask again			
		Ca	incel	Launch

Welcomeと表示されますが必要ないので×ボタンをクリックして消します。

右下のAlways show Welcome at start upのチェックをはずいておけば毎回起動時に表示されることがなくなります。



プロジェクトの作成&ビルド

メニューからFile > New > C++Projectを選択します。

Eclipse -			Thu 1634 •					
workspace - https://www.eclipse.org	g/setups/donate/	7500	pe=Eclipse%20IDE%20For%20C/	C%28%28%	20Developers	%20(includes%2.		
File Edit Source Refactor Navigat	e Search Projec	t Ru	in Window Help					
New	A second	E	Arduino Project			() #1(, #1		
Open File		1	Makefile Project with Existing Co	ode.			0	10.03
🗣 Open Projects from File System.	••	Ē	C/C++ Project			Dec. 14 44 100 10	~	E3 40
Recent Files	N.		Project		-	Se Outl 24 UP 6	iuil	20
Close Editor	Ctrl+W	. 24	Convert to a C/C++ Autotools Pro	oject		a - 8 - 68		
Close All Editors	shift+ctrl+W	2-	Convert to a C/C++ Project (Add	s C/C++ Natu	re)	There is no activ that provides an	e ed	line
Save			Source Folder			char provides at	100	Contract.
			Folder					
Save All		G	Source File					
		h	Header File		1			
			File from Template					
d Rename		Ø	Class					
Refresh	+5		Example					
Convert Line Delimiters To	•		Other		CERI+N			
🌒 Print		the s	ite. You can set your browser to r	notify you be	efore you			
import		ikie o	or turn off cookies. If you do so, h	owever, son	ne areas			
🖬 Export		may	not function properly. To read Er	clipse Foun	dation			
Properties	Alt+Enter	v cli	ck here.					
Switch Workspace		d lent						
Restart		Allo	w cookies					
Exit		1944.00	and the second					
	Problems 22	E Tar		Call Graph			7	-
	û items		the second second second second	and an application of the				
	Description		Res	ource	Path	Location	1	Туре

CもしくはC++言語のプログラムを作成するため「C++ Managed Build]を選択してNextをクリックします。

a,

at.	Arduino C++ Sketch
rduino Mako	A single C++ file with empty setup() and loop() functions.
lake leson	C Managed Build A C Project build using the CDT's managed build system.
t	C++ Managed Build A C++ Project build using the CDT's managed build system.
	CMake Project A CMake project with a Hello World executable to get started.
	Empty or Existing CMake Project Create a CMake project with no files. Can be used to create one over existing content.
	Empty or Existing Meson Project Create a Meson project with no files. Can be used to create one over existing content.
	Makefile Project (Experimental) Create a new project that builds with the'make' build tool using CDT's new Core Build System.
	Meson Project A Meson project with a Hello World executable to get started.
	Qt C++/QML Application

Project Nameを入力します。本マニュアルではTestにしています。 Project typeにEmpty Project、ToolchainsにCross GCCを選択します。 Nextをクリックします。

a.			0 😣
C++ Projec Create C++ p	t project of selected type		Ď
Project nam	ne: Test		
Location:	/work/app/workspace/Test hoose file system: default]	Browse
Project type Control Control	e: Autotools table o <mark>ty Project</mark> o World C++ Project d Library Library ile project	Toolchains: Cross GCC Linux GCC	
Show pr	oject <mark>ty</mark> pes an <mark>d</mark> toolchains only	if the <mark>y a</mark> re supported on the	e platform

ここではデフォルト設定のままにします。DebugとRelease設定をもつProjectになります。 Nextをクリックします。

	, , , , , , , , , . , . , . , . , . , . , . , . , . , . , . , . , . , . , . , .	<u>1</u>
roject type: oolchains: onfigurations:	Executable Cross GCC	
🛛 🛞 Debug		Select all
🌌 🧐 Release		Deselect all
		Advanced settings
se "Advanced s	settings" button to edit project's properties.	

Cross Compiler prefix [CTARGET_PREFIX]

Cross Compiler pathに「\${OECORE_NATIVE_SYSROOT}/usr/bin/arm-angstrom-linux-gnueabi」を入力してFinishをクリックします。

画面を大きくしないとCross Compiler pathの設定がすべて見えないので画面を最大化して確認してください。



作成したプロジェクトを右クリックしてPropertiesを開きます。



C/C++ Build > Settings > Cross GCC Compiler > Includesで+のアイコンをクリックして下記を設定します。 \${SDK_INCLUDE}



設定追加のアイコン

同画面のままOptimizationを選択しOther optimization flagsに下記を入力します。 \${SDK_OPTIMIZATION}

	Propertie	es for Test		1
type filter text	Settings		Q+0	Ŧ
Resource Builders	Configuration: Debug [Active]		Manage Configuratio	ns.
Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Linux Tools Path Project Natures Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Cross Settings Container S Cross Settings Cross GCC Compiler Cross GCC Compiler Cross GCC Compiler Cross GCC Compiler Preprocessor Cost General Cross G++ Compiler Cross G++ Compiler Compiler Cross G++ Compiler Cross G++ Co	Settings Pould Steps Optimization Level Other optimization flags	Build Artifact Binary Parse None (-00) \${SDK_OPTIMIZATION}	•
201			Restore Defaults Apply	y.

同画面のままCross G++ Compiler > Includesにも同様に下記を設定します。設定を追加しただけで前回入力したものが入力 されていますのでその場合は入力の必要はありません。 \${SDK INCLUDE}



設定追加のアイコン

同画面のままOptimizationを選択しOther optimization flagsに下記を入力します。 \${SDK_OPTIMIZATION}

Type filter text Settings Resource Builders Configuration: Debug [Active] Manage Configurations C (/C++ Build Build Variables Environment Logging Source Build Status Container Settings Build Steps Build Artifact Binary Parsers Settings Cool Settings Container Settings Optimization Level None (-00) Image Configurations Settings Settings Optimization Level None (-00) Image Configurations Settings Settings Optimization Level None (-00) Image Configurations C/C++ General Linux Tools Path Project References Run/Debugstetings Brepository Build Mascellaneous Optimization Sigs Sigs Sigs Sign Sign Sign Sign Sign Sign Sign Sign		Properti	es for Test		9
 Resource Builders C/C++ Build Build Variables Environment Logging Settings Constance Settings Pool Settings Container Settings Optimization Level None (-00) Solution Level None (-00) Solution Level Solution Hags Solution Solution Solution Project Natures Project Natures Bobugging Warnings Walidation Debugging Warnings Miscellaneous Solutions Solutions Solutions Solutions Solutions Solutions Solutions Solution Level Solution Hags Solution Hags	type filter text	Settings			Q+0+
Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Linux Tools Path Project Natures Run/Debug Settings Task Repository Task Repository Task Repository Task Repository Validation WikiText Berostory WikiText Berostory Build Steps Build Steps Build Artifact Binary Parsers Optimization Level None (-00) Other optimization flags Stoppinization Bobugging Warnings Bincludes Debugging Warnings Bobugging Warnings Bincludes Debugging Warnings Bincludes Debugging Warnings Bincludes Debugging Warnings Bincludes Debugging <td>Resource Builders C/C++ Build</td> <td>Configuration: Debug [Active]</td> <td></td> <td>▼ Manag</td> <td>ge Configurations</td>	Resource Builders C/C++ Build	Configuration: Debug [Active]		▼ Manag	ge Configurations
Bross GCC Assembler	Build Variables Environment Logging Settings: Tool Chain Editor C/C++ General Linux Tools Path Project Natures Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Cross Settings Container Cross Settings Cross GCC Compiler Dialect Preprocessor Includes Warnings Warnings Warnings Miscellaneous Cross G++ Compiler Bialect Preprocessor Includes Cotimization Debugging Warnings Dialect Debugging Warnings Miscellaneous Cotimization Cotimizatio Cotimization Cotimizatio Cotimization Cotimization C	Settings Puild Steps Optimization Level Other optimization flags	Build Artifact None (-00) \$[SDK_OPTIMIZA	Binary Parsers
Restore Defaults Apply				Restore Defau	lts Apply

同画面のままCross G++ Linker > MiscellaneousのLinker flagsに下記を設定します。 \${SDK_LINKER}

type filter text	Settings	¢•⇒•
Builders	Configuration: Debug [Active]	Manage Configurations
Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Linux Tools Path Project Natures Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Cross Settings Cross Settings Cross Settings Cross GCC Compiler Dialect Preprocessor Includes Coptimization Debugging Warnings Miscellaneous Cross G++ Compiler Dialect Preprocessor Includes Optimization Debugging Warnings Miscellaneous Cross G++ Linker Cross G++ Linker Coss G++ Linker Cross G++ Linker Coss G++ Linker Cross G++ Cross G++ Linker Cross G++ Cross G++ Linker Cross G++ Cross G++ Linker Cross G++ Cross G++	Settings Build Steps Linker flags \$[SDK_LINKER]] Other options (-Xlinker [option]) Other objects

ConfigurationをReleaseにしてReleaseの設定にもIncludesやOptimization、Miscellaneousに同様のパラメータを入力します。 すべての設定入力後Apply and Closeをクリックします。

	Properti	es for Test	
type filter text	Settings		⇔ +⊖
Resource Builders	Configuration: Release		Manage Configuration
Growthead and a second and a second a s	 Tool Settings Cross Settings Cross GCC Compiler Dialect Preprocessor Includes Optimization Debugging Warnings Miscellaneous Cross G++ Compiler Dialect Preprocessor Includes Optimization Debugging Warnings Miscellaneous Cross G++ Linker General Libraries Miscellaneous Cross G++ Linker Shared Library Settings Cross GCC Assembler 	Settings Puild Steps Build Ste	ild Artifact 前Binary Parsers 配配面子

下記のようなリビルドするまで変更が反映されないという警告が出ます。

基本的にプロジェクトの設定変更後はリビルドして反映させますのでRemember my decisionにチェックをいれてYesをクリックします。

	Settings	8
?	Changes made will not be reflected in the index until it is rebut to rebuild it now?	ilt. Do you wi <mark>sh</mark>
R	emember my decision	
	No	Yes

プロジェクトを右クリックしNew > Source Fileをクリックしてソースファイルを追加します。



Source fileを入力します。拡張子を必ずつけてください。

Templateはソースコードに付加されるコメントですが本マニュアルではNoneとしています。Finishをクリックします。

	New Source Fil	e 💿 😣
Source File		
Create a new s	ource file.	<u> </u>
Source folder:	Test	Browse
Source file:	Test.c	
Template:	<none></none>	▼ Configure
	<u> </u>	

```
ソースコードを入力します。本マニュアルではHello Worldと出力するだけのソースコードを入力しています。
#include <stdio.h>
int main(int argc, char ** argv)
{
    printf("Hello World");
    return 0;
}
```



ビルドボタンを押してビルドを行います。Consoleにログが出力されます。正常に終わった場合、青字でログが出力されます。



ビルドログは下記のようになります。

もしエラーが出た場合にはなにかしらの設定ミスがあります。差異に注意してください。

15:20:46 **** Build of configuration Debug for project Test ****

make all

Building file: ../Test.c

Invoking: Cross GCC Compiler

arm-tdx-linux-gnueabi-gcc -l"/opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi/usr/include/" -O0 -march=ar mv7-a -fno-tree-vectorize -mthumb-interwork -mfpu=neon -mfloat-abi=hard -Wno-poison-system-directories -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"Test.d" -MT"Test.o" -o "Test.o" "../Test.c"

Finished building: ../Test.c

Building target: Test

Invoking: Cross G++ Linker

arm-tdx-linux-gnueabi-g++ -mfloat-abi=hard -L/opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi/lib/ -WI,-rp ath-link,/opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi/lib/ -L/opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neo n-tdx-linux-gnueabi/usr/lib/ -WI,-rpath-link,/opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi/usr/lib/ --sysroo t=/opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi -o "Test" ./Test.o

Finished building target: Test

15:20:47 Build Finished. 0 errors, 0 warnings. (took 269ms)

SSH接続設定作成

デバッグに使用するGDBはEthernetで接続してSSHプロトコルを利用してデバッグを行います。

デバッグを行うためにモジュールとSSHで接続できるようにしておく必要があります。

モジュール側の設定を行います。

モジュールにEthernetケーブルを挿入し開発パソコンと同じサブネットに接続します。

Teratermを起動してモジュールを起動します。rootでログインしpasswdコマンドでパスワードを設定します。

本マニュアルではセキュリティを一切気にせず利便性のよいパスワード認証を使い、rootでSSHにログインできるようにします。 あくまでデバッグ目的で設定するだけです。

[colibri-imx7]# passwd

パスワードを2回入力するとパスワードが設定されます。(2回目は確認用)

COM1:115200bps - Tera Term VT	-		×
ファイル(E) 編集(E) 設定(<u>S</u>) コントロール(<u>O</u>) ウィンドウ(<u>W</u>) 漢字コード(<u>K</u>) ヘルプ(<u>H</u>)			
 OK] Reached target Network. 7.958869] IPv6: ADDRCONF(NETDEV_UP): usb0: link is not ready OK] Started NFS status monitor for NFSv2/3 locking Starting Permit User Sessions Starting Lightning Fast Webserver With Light System Requ OK] Started Permit User Sessions. OK] Started Hostname Service. OK] Started Lightning Fast Webserver With Light System Requi OK] Started Service. OK] Started Lightning Fast Webserver With Light System Requi OK] Started Lightning Fast Webserver With Light System Requi OK] Started Service. OK] Started Lightning Fast Webserver With Light System Requi OK] Started Service OK Started Service OK	i remen rement	ts s.	^
TDX X11 2.6-snapshot colibri-imx7 ttymxc0			
Colibri-iMX7_Console-Image 3.0b4 20200824			
colibri-imx7 login: [58.489749] fec 30be0000.ethernet eth0: Lin bps/Full - flow control rx/tx [58.497703] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes root root@colibri-imx7:~# passwd New password: Reture new password:	nk is ready	Up - 11	DOM
passwd: password updated successfully			
root@colibri-imx7:~#			~

次にモジュールのIPアドレスの設定を行います。何も設定していない場合はDHCPとなります。 設定ファイル/etc/systemd/network/wired.networkを新規作成します。

[colibri-imx7]# vi /etc/systemd/network/wired.network

eth0はインターフェイス名です。モジュールやBSPのバージョンによって異なりますのでifconfigコマンドで調べてください。

固定IPの場合

[Match] Name=eth0

[Network] Address=192.168.100.10/24 Gateway=192.168.100.254

192.168.100.10/24はIPアドレス192.168.100.10 サブネットマスク255.255.255.0を意味します。

下記コマンドでネットワークマネージャーを再起動します。 [colibri-imx7]# systemctl restart systemd-networkd

ifconfigで設定が反映されているのを確かめます。 [colibri-imx7]# ifconfig

COM1:115200bps - Tera Term VT × ファイル(E) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) 漢字コード(K) ヘルプ(H) root@colibri-imx7:~~# vi /etc/svstemd/network/wired.network root@colibri-imx7:~# systemctl restart systemd-networkd root@colibri-imx7:~# ifconfig Link encap:Ethernet HWaddr 00:14:2D:2B:66:2F leth0. inet addr:192.168.100.10 <u>Bcast:192.168.100.255</u> <u>Mask:255.255.255.0</u> inet6 addr: fe80::214:2dff:fe2b:662f/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:70 errors:0 dropped:0 overruns:0 frame:0 TX packets:116 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:8259 (8.0 KiB) TX bytes:17923 (17.5 KiB) Link encap:Local Loopback lo inet addr:127.0<u>.0.1</u> Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host UP LOOPBACK RUNNING MTU:65536 Metric:1 RX packets:306 errors:0 dropped:0 overruns:0 frame:0 TX packets:306 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:23692 (23.1 KiB) TX bytes:23692 (23.1 KiB) usb0 Link encap:Ethernet HWaddr 00:14:2D:FF:FF:FF UP BROADCAST MULTICAST MTU:1500 Metric:1

SSH接続確認

開発パソコンから接続できているか確認します。VMwareの設定がデフォルトのNATになっている場合、ホストOSのWindows10 で接続できていればUbuntu側で接続できます。接続できない場合は何かしらの設定が間違っている可能性があります。 [ubuntu]\$ ping 192.168.100.10

デフォルト設定のOSイメージではSSHサーバーが起動しています。 Teratermの新しい接続でSSH接続をして接続できるかどうか確かめてみます。 Teratermのメニューからファイル > 新しい接続を選択します。

<u>vi</u> 1	「era Term - [未掛	妾続] VT					_	×
ファイル	レ(F) 編集(E)	設定(S)	コントロール(0)	ウィンドウ(W)	漢字コード(K)	ヘルプ(H)		
i	新しい接続(N)		Alt+N					^
	セッションの複製((U)	Alt+D					
1	Cygwin接続(G)		Alt+G					
	ログ(L)							
1	ログにコメントを付	加(0)						
	ログを表示(V)							
1	ログダイアログを表	表示(W)						
	ファイル送信(S)							
i	転送(T)		>					
1	SSH SCR							
	ディレクトリを変更	Ē(C)						
1	ログを再生(R)							
	TTY Record							
	TTY Replay							
	印刷(P)		Alt+P					
	接続断(D)		Alt+I					
i	終了(X)		Alt+Q					
	Tera Termの全総	§了(A)						

ホスト(T)にはモジュールに設定したIPを入力、サービスはSSHを選択してOKをクリックします。

Tera Term: 新しい接続	×
● TCP/IP	ホスト(<u>T</u>): 192.168.100.10 ~ 「ビヒストリ(O) サービス: O Telnet TCPポート#(<u>P</u>): 22 ● SSH SSHバージョン(<u>V</u>): SSH2 ~ O その他 プロトコル(<u>C</u>): UNSPEC ~
Οシリアル(E)	ポート(R): COM1: PCIe to High Speed Serial Port (C \sim
	OK キャンセル ヘルプ(<u>H</u>)

下記のような警告が出てきます。意図する接続先に間違いありませんので続行をクリックします。

セキュリティ警告	\times
known hostsリストにサーバ 192.168.100.10 のエントリはありません。 悪意を持ったホストが、 接続しようとしているサーバのふりをしている可能性もありますので、 十分注意してください!	
known hostsリストのこのホストを追加して続行すると、次回からこの警告は出な くなります。	
サーバ側のホスト鍵指紋: 鍵指紋ハッシュアルゴリズム: <u>M</u> D5 <u>●SHA256</u> SHA256:EWftPM6I/0fC3+mi75kilmtbi0OUFVnigQEdPilycPI	
+[ECDSA 256]+ o o+o*=+. = .*oo+ . E.+o+ o .+. + . S o = . + o . +o *0+.+ o+=o==ooo +[SHA256]+	
✓このホストをknown hostsリストに追加する(A) 続行(C) 接続断(D)	

ユーザ名はroot、パスフレーズはpasswdコマンドで設定したパスワードを入力します。 OKをクリックして接続します。

SSH認証			_		\times
ログイン中: 192.168.100.10					
認証が必要です。					
ユーザ名(<u>N</u>): root		•			
パスフレーズ 🛛 🌢 🔴 🌢	••••	-]		
☑バスワードをメモリ上に記憶	きする(<u>M</u>)				
□エージェント転送する(0)					
Authentication methods					
● ブレインバスワードを使う(」	i				
O <u>R</u> SA/DSA/ECDSA/ED255	19鍵を使う				_
秘密鍵(<u>K)</u> :					
니는/JJ////J/드카워(U):					
ホスト鍵(<u>F)</u> :					
○ キーボードインタラクティブ	認証を使う(1)				
○ Pageantを使う					
			_		
		OK		接続断	<u>D</u>

接続できた場合は下記のような画面になります。接続確認ができたらTeratermを終了します。

			· · · · · · · · · · · · · · · · · · ·					
💻 192.1	68.100.10	- Tera Tern	n VT				_	\times
ファイル(E)	編集(<u>E</u>)	設定(<u>S</u>)	בארכ (<u>ס</u>)ע-ם	ウィンドウ(<u>W</u>)	漢字コード(<u>K</u>)	ヘルプ(田)		
Last Io	gin: Fr	ri Nov (29 04:19:34	2019				^
root@co	libri-i	ım×7:~#						
								~

Ubuntuに戻ります。EclipseのメニューのWindow > Show View > Otherを選択します。



Connectionsを選択してOpenをクリックします。



+マーク(赤枠)のアイコンをクリックしてConnection設定を追加します。



1	c	onnection Type		0 😣
Connection Ty	pe			
Select type of c	connection to create.			
Serial Port				
Arduino				
🔠 Telnet				
SSH				
1				1
?	< Back	Next >	Cancel	Finish

Connection nameにわかりやすい名前を付けます。

Host, User, PasswordにそれぞれモジュールのIPアドレス、ユーザー名、パスワードを入力してFinishをクリックします。

lew Connec	tion
Specify prop	erties of a new connection
Connection r	ame: Colibri-iMX7
lost informa	ation
Host:	192.168.100.10
User:	root
O Public k	ey based authentication Keys are set at <u>Network Connections, SSH2</u>
Passphrase	:
O Passwo	rd based authentication
Password:	

下記のようなウィンドウが表示されます。Ubuntuのパスワードを入力してUnlockをクリックします。



パスワードの保存を行った場合、パスワード復元用のヒントを作るかどうかを問われます。本マニュアルでは使用しないのでNo をクリックします。

	Secure Scorage - Password Hint Needed	
?)	A master password was created in Eclipse Secure Storage. If you ever n recover your password, you need a password hint. Do you want to prov password hint?	eed to ide a

作成したConnection設定を右クリックしてOpen Command Shellを選択します。



モジュールと接続できると下記のような接続先が正しいホストかどうかの警告がでますがYesをクリックします。



既知のホストリストを保存するファイルがないので作成するかどうかを問われます。 Yesをクリックします。



モジュールと接続できるとConsoleでコマンドが打てるようになります。下記ではpwdコマンドを実行しています。



デバッグ設定作成

GDBの初期処理を記述するgdbinitファイルを作成します。本マニュアルでは/work/app/gdbinit に作成します。 [ubuntu]\$ gedit /work/app/gdbinit

内容は下記です。(環境変数は使用できません。)

set sysroot /opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi

set auto-load safe-path /opt/tdx-x11/2.6.4/sysroots/armv7at2hf-neon-tdx-linux-gnueabi

上記2つのコマンドで指定するパスはSDKが出力した環境変数設定シェル内の下記の内容になります。 SDKTARGETSYSROOT

下記コマンドで見ることができます。

[ubuntu]\$ cat /opt/tdx-x11/2.6.4/environment-setup-armv7at2hf-neon-tdx-linux-gnueabi | grep SDKTARGETSYSROOT=

user1@ubuntu: /work/app	e 🛛 😣
File Edit View Search Terminal Help	
<pre>user1@ubuntu:/work/app\$ cat /opt/tdx-x11/2.6.4/environm -tdx-linux-gnueabi grep SDKTARGETSYSROOT= export SDKTARGETSYSROOT=/opt/tdx-x11/2.6.4/sysroots/arm ueabi user1@ubuntu:/work/app\$</pre>	ent-setup-armv7at2hf-neon v7at2hf-neon-tdx-linux-gn

デバッグ

プロジェクトエクスプローラに戻り(赤枠をクリック)、プロジェクトを右クリックしDebug As > Debug Configurationsを選択します。

s 🗧 Eclipse 🔻				F
				workspace - Tes
File Edit Source Re	efactor Navigate Search P	roject Run Windo	w Help	
<u>≪</u> ⊙ ■	✓ No Launch C	Configurations	✓ on: —	✓ ☆ ★
ြဲ Project 🛛 🛓 Cor	nnect 🖓 🗖 🚺 Test.c 🛛			
	E∰7 8 1 #include	<stdio.h></stdio.h>		
🕆 🖨 Test	New	Plat Proc Char	* argv)	
Binaries	Go Into);		
Includes	Open in New Window			
Debug	Show In	Shift+Alt+W ▶		
Test.c	Show in Local Terminal	Þ		
	📔 Сору	Ctrl+C		
	💼 Paste	Ctrl+V		
	🞽 Delete			
	Source	•		
	Move			
	Rename	FZ		
	import			
	Build Project			
	Refrech	FS	DATE: NOT THE REPORT OF	
	Close Project	ble ble	🖾 Properties 🚻 Cal	l Graph
	Close Unrelated Project			
	Build Targets	► Ig	iracion Debug for pro	lect lest ****
	Index	×		
	Build Configurations	► <mark>1</mark> -	/opt/tdx-x11/2.6.4/sv	sroots/armv7at2hf-neon-tdx-linu
	Profiling Tools	▶ <mark>c</mark>		
	Run As	₩.		
	🎋 Debug As		1 C/C++ Container Ap	plication
	Profile As	2	<u>2</u> Local C/C++ Applica	tion 6.4/sysroots/armv7at2
	Restore from Local History	<i>[</i>	De <u>b</u> ug Configurations	
	Run C/C++ Code Analys	is		

C/C++ Remote Application(青枠)をダブルクリックしデバッグ設定を追加します。赤枠内の設定を行います。 Projectと C/C++ Applicationは自動で設定されるままで問題ありません。 Connectionは先ほど作成した設定を選択します。

	Debug Configurations	0 😣
Create, manage, and run configura S Remote executable path is not specif	fied.	Ť
Image: Second system type filter text Image: C/C++ Application Image: C/C++ Attach to Application Image: C/C++ Container Launcher Image: C/C++ Postmortem Debugger Image: C/C++ Remote Application Image: C/C++ Remote Application Image: C/C++ Remote Application Image: C/C++ Unit Image: CDB Hardware Debugging Image: Launch Group Image: Launch Group (Deprecated)	Name: Test Debug Main Main Project: Test Test C/C++ Application: Debug/Test Variables Build (if required) before launching Build Configuration: Select Automatically Disable auto build Image: Disable auto build Onfigure Workspace Settings Configure Workspace Settings Connection: Colibri-iMX7 Remote Absolute File Path for C/C++ Application: New	<u>B</u> rowse B <u>r</u> owse ▼
Filter matched 10 of 10 items	Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select other</u> Revert	Apply
1	Close	Debug

4	Debug Configurations 💿 🛞
Create, manage, and run configura Remote executable path is not speci	fied.
Image: Second system Image: Second system <th>Name: Test Debug Main Arguments Main Arguments Build (if required) before launching Build Configuration: Select Automatically Enable auto build Disable auto build Use workspace settings Configure Workspace Settings Connection: Colibri-iMX7 Remote Absolute File Path for C/C++ Application: Browse Commands to execute before application Browse Skip download to target path. Skip download to target path.</th>	Name: Test Debug Main Arguments Main Arguments Build (if required) before launching Build Configuration: Select Automatically Enable auto build Disable auto build Use workspace settings Configure Workspace Settings Connection: Colibri-iMX7 Remote Absolute File Path for C/C++ Application: Browse Commands to execute before application Browse Skip download to target path. Skip download to target path.
Filter matched 10 of 10 items	Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select other</u> Revert Apply
•	Close Debug

接続ができていればモジュール内のホームディレクトリが見えます。(ファイルは何もありません。)そのままOKをクリックします。

Select Remote C/C++ Application File
Selected resource: /home/root
Show hidden files
Cancel OK

自動的にRemote Absolute File Path for C/C++ Applicationに実行パスが設定されます。

	Debug Configurations	0				
Create, manage, and run con	igurations	Ś				
	Name: Test Debug					
type filter text	B Main ⋈= Arguments 券 Debugger 💱 Source 🔲 Common					
C/C++ Application C/C++ Attach to Applicatic C/C++ Container Launcher	Variables Search Project Build (if required) before launching Search Project	t B <u>r</u> owse				
C/C++ Postmortem Debug	Build Configuration: Select Automatically					
▼ C/C++ Remote Application Test Debug Cii C/C++ Unit	Configure Workspace Settings					
🖸 GDB Hardware Debugging 🛱 Launch Group	Connection: Colibri-iMX7 New Edit	Properties				
Launch Group (Deprecated	Remote Absolute File Path for C/C++ Application:					
	/home/root/Test	Browse				
	Commands to execute before application					
	Skip download to target path.					
Filter matched 10 of 10 items	Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select other</u>	Apply				
0	Close	Debug				

Debuggerのタブを開きます。GDB debuggerにgdbのパスを指定します。 本マニュアルでは下記になります。(環境変数は使用できません。) /opt/tdx-x11/2.6.4/sysroots/x86 64-tdxsdk-linux/usr/bin/arm-tdx-linux-gnueabi/arm-tdx-linux-gnueabi-gdb

上記で指定するパスはSDKが出力した環境変数設定シェル内の下記の設定によります。 OECORE_NATIVE_SYSROOT 上記設定配下の/usr/bin/arm-tdx-linux-gnueabi/arm-tdx-linux-gnueabi-gdbになります。

OECORE_NATIVE_SYSROOTの内容は下記コマンドで見ることができます。

cat /opt/tdx-x11/2.6.4/environment-setup-armv7at2hf-neon-tdx-linux-gnueabi | grep OECORE_NATIVE_SYSROOT=



GDB command fileには先ほど作成したgdbinitのパスを指定します。本マニュアルでは下記になります。 /work/app/gdbinit

Applyをクリック後Debugをクリックします。

	Debug Configurations	0				
Create, manage, and run con	figurations	Ť				
┇╔╬╔ <mark>╳</mark> ╘┦╶	Name: Test Debug					
type filter text	Main 🕪 Arguments 🕸 Debugger 🖕 Source 🔲 Common					
C/C++ Application C/C++ Attach to Applicatic	Stop on startup at: main					
C/C++ Container Launcher	Main Shared Libraries Coloranues Settings					
 C/C++ Remote Application Test Debug 	GDB debugger: dk-linux/usr/bin/arm-tdx-linux-gnueabi/arm-tdx-linux-gnueabi-gdb]					
Cü C/C++ Unit						
© GDB Hardware Debugging ♣ Launch Group ▶ Launch Group (Deprecatec	(Warning: Some commands in this file may interfere with the startup operation of the d example "run".)	ebugger, for				
	Enable Reverse Debugging at startup using: Software Reverse Debugging (detailed	d but slower)				
	 Force thread list update on suspend Automatically debug forked processes (Note: Requires Multi Process GDB) 					
Filter matched 10 of 10 items	Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select other</u>	Apply				
?	Close	Debug				

Debugが始まると下記のようなDebugウィンドウを開くかどうか問われますのでSwitchをクリックします。 この操作を覚える場合はRemember my decisionにチェックを入れておきます。

Confirm Perspective Switch



This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Do you want to switch to this perspective now?

🖾 Remember my decision

No

Switch

デバッグウィンドウが表示されデバッグを行うことができます。

а,	wor	kspace - Test/Te	st.c - Eclip	se IDE			0	
File Edit Source Refactor	Navigate Searc	h Project Run	Window	Help				
🐔 🕸 🔳 🎋 Debug	∨ 💽 Test	Debug	~ \$	k on: -		~	*	010
🕒 🔌 🕪 🛛 🔳 🕅 🧟 😨 . 🖬 🗮		- 00	491 - 51	* (0 * C) *	- 2	1	۹ 😰	₽
☆ Debu ⋈ Image: blue black Image: black	I #include	e <stdio.h></stdio.h>				(x)= V ☎ 🎭 E	3 ∰ E "1 1 00 🖻 📬	
▼ C Test Debug [C/C++ Remot	2⊖int main 3 {	n(int argc, cha	r ** argv	()		Name	Туре	
 Test [362] [cores: 1] Thread #1 [Test] 362 main() at Test.c:40 Remote Shell /opt/tdx-x11/2.6.4/syst 	<pre> 4 print 5 return 6 } 7 </pre>	f{"Hello World" n 0;);			⇔=argc ▶ ⇒ argv	int char **	
	⊑ Console ¤	📲 Registers 🔝	Problems	Executabl	Ę	Debugger	Memory	- 8
	Test Debug [C/0	C++ Remote Appli	cation]			XXB		- 1
	gdbserver : root@colibri Process /hom Listening on Remote debug	2345 /home/root -imx7:~# gdbser e/root/Test cre port 2345 ging from host	:/Test;exi ver :234 ated; pic 192.168.1	it 45 /home/roo d = 362 100.106	t/Te	est;exit		
				4 4 4				10

デバッグを行うデバッグウィンドウとソースコードを編集するプロジェクトエクスプローラを切り返すには右上のボタンで切り替える ことができます。

4	wor	kspace - Test/	Test.c - Eclipse I	IDE		00	8
File Edit Source Refactor	Navigate Sear	ch Project Ru	ın Window Hel	P			
🐔 🕸 🔳 🎋 Debug	V C Tes	t Debug	~ 🌩 o	n: 🗕	2	× 🌣 📑 🔹 🖓	010
🕒 🔍 🗈 🗆 🛤 🗷 🧐 . iə 🗮	**0-0	• • •	1.3191+61+1	20 + 0 + <mark>-</mark>	3	Q 🖻 🖬	
* Debu X Proje □ i i	© Test.c ⊠ 1 #includ	e <stdio.h></stdio.h>		- 8	(x)= V ⊠ ⁰₀	B ∰ E *1 -	
▼ C Test Debug [C/C++ Remot	2⊖ int mai 3 {	n(int argc, c	: har ** argv)		Name	Туре	•
 Test [362] [cores: 1] Thread #1 [Test] 362 main() at Test.c:40 Remote Shell Jopt/tdx-x11/2.6.4/syst 	<pre> 4 print 5 retur 6 } 7 </pre>	<pre> 4</pre>		(≫=argc)	int char **		
	Console X Test Debug [C/	It Registers	Problems 🔾	Executabl 🖬	B Debugger	0 Memory " 	9 •
	gdbserver : root@colibri Process /hom Listening on Remote debug	2345 /home/ro -imx7:~# gdb e/root/Test o port 2345 ging from hos	bot/Test;exit server :2345 / created; pid = st 192.168.100.	/home/root/T 362 106	est;exit		
			1				10

Releaseビルドで実行ファイルを作成

デバッグを行いプログラムを完成させた後はリリースビルドで最適化を行ったバージョンの実行ファイルを作成します。 プロジェクトを右クリック Build Configurations > Set Active > Releaseを選択してReleaseに変更後ビルドボタンを押しビルドを 行います。ビルドした実行ファイルは/work/app/workspace/Test/Releaseに格納されています。この実行ファイルをSDカードな どでモジュールに移動し実行することができます。

以上でアプリケーションの開発は終了です。

		workspace - Test/Test.c - Eclipse IDE	_
File Edit	Source Refactor Navigate	Search Project Run Window Help	
*	New	▶ ~ 奈 on: ~ ~ 奈	* • 6
≥ 0≥000	Go Into	-1.71 +7 + CQ + C + M	
🅸 Debug	Show In	Shift+Alt+W >	(x)= Varia 🖾
-	Show in Local Terminal	<pre>tdio.h></pre>	
🕶 😂 Test	🗎 Сору	ctrl+C	Name
► 🐇 Bir	Paste	Ctrl+v Hello World");	
Inc 🕞	💥 Delete		
De De	Source		
, <u>s</u> ic.	Rename	F2	
	🚵 Import		
	👍 Export		
	Build Project		
	Clean Project	9.22	
	Refresh	F5	
	Close Uncelated Project	Problems 🜔 Executables 📓 Debugger Console 👘	**
	Build Targets	Debug [C/C++ Remote Application] Remote Shell (Term	inated Sep 9,
	Index	<pre>> /nome/root/lest;exit</pre>	
	Build Configurations	Set Active ▶ ✓ <u>1</u> Debug ot/Test;exit	
	Profiling Tools	Manage <u>2 Release</u>	
	🜔 Run As	▶ Build All B.100.106	
	🐞 Debug As	Clean All	
🔒 Test	Profile As	Build Selected	