### XG-335x Qt (4.8.5)プログラミング

Rev1.0 2014/11/12



1.	概要		1
	1.1	はじめに1	
	1.2	開発言語について1	
	1.3	QtのGUI開発環境について1	
2.	Qt Cre	ator の概要	2
	2.1	システム概要2	
	2.2	Qt システム構成2	
3.	Qt Cre	ator のインストール	3
4.	Qt Cre	ator によるサンプルアプリ作成	4
	4.1	Qt Creatorの起動4	
	4.2	XG シリーズのためのデフォルト設定5	
	4.3	プロジェクトの作成10	
	4.4	プロジェクトの設定13	
	4.5	サンプル GUI アプリ作成16	
	4.6	サンプルの実行	
5	リモー	トデバッガ	27
	5.1	リモートデバッガの設定27	
	5.2	リモートデバッグ	
6.	Qtプロ	]グラミング Tips	32
	6.1	 フォントファミリの指定32	
7.	関連情	報	34

# 1. 概要

### 1.1 はじめに

Qt は主に Unix/Linux で利用されている GUI ツールキットです。Windows でも利用することができ Linux と Windows 両方 に対応するアプリケーション開発などでしばしば利用されています。また、組込み系のシステムの GUI としても利用されてい ます。

本ドキュメントでは、XG シリーズをターゲットとした Qt のアプリケーション開発方法を説明します。



### 1.2 開発言語について

Qt は C++によって開発されており、多くのアプリケーションが C++によって開発されています。また、多くのコミュニティによって Qt バインディングが開発されており、多くの言語で Qt アプリケーションを開発することが可能です。本アプリケーションノートでは Qt Creator を使った、XG シリーズ用のクロスアプリ開発について解説します。

Qtはバインディングを利用することによって他の多くの言語で開発することも可能です。各種言語でQtを利用するときはその言語で必要となるバインディングのためのライブラリ等をターゲットにインストールすることが必要となります。

本ドキュメントでは、Qt Creator および C++での開発を前提としたアプリケーション開発について説明します。 その他の言語での詳細については、一般書籍やインターネットから多くの情報を得られますので、それらを参考にしてください。

### 1.3 QtのGUI 開発環境について

Qt Creator は Qt 用の GUI 開発環境です。Qt Creator は Ubuntu ソフトウェアセンターに登録されています。 Ubutntu12.04LTS の場合 Qt Creator2.4.1(Qt 4.8.0)をインストールすることができますが、最新版を利用したい場合は別 途インストール作業が必要となります。

# 2. Qt Creator の概要

### 2.1 システム概要

Qt アプリを実行するには、バックエンドとして X.org、Kdrive(Xfbdev)、DirectFB 等の組み合わせ構成や、framebuffer のみの構成があります。

これらのどのバックエンドと組み合わせるかは、アプリケーションの稼動環境に応じて選択することになります。

### 2.2 Qt システム構成

Qt を動かすための構成は幾つかありますが、ここでは framebuffer のみの構成を例にして説明します。アプリの稼動環境が すべて Qt による独自開発のアプリケーションの場合は framebuffer のみの構成の方が GUI の動作が速くなります。X11 を前 提としたアプリを移植する場合は Xfbdev をバックエンドにした方が有益ですので、必要に応じて環境を構築します。



Fig 2.2-1 Qt システム例

# 3. Qt Creator のインストール

Qt および Qt Creator をインストールするにはターミナルを開き以下のコマンドを入力します。

省略 \$ sudo apt-get install qtcreator 🗘



Ubuntu12.04LTS でインストールされる Qt Creator は 2.4.1(Qt 4.8.0)です。XG シリーズの Qt は 4.8.5 です。4.8.0 以降に追加された機能についての互換性が必要でしたら、Ubuntu 側を最新の Qt にする必要があります。

# 4. Qt Creator によるサンプルアプリ作成

### 4.1 Qt Creator の起動

Ubuntu Dash ランチャーに Qt Creator のアイコンがある場合は、それをクリックして、Qt Creator を起動します。ランチャーに Qt Creator のアイコンが無い場合は Dash ホームアイコンをクリックし「検索」項目に Qt と入力して Qt 関連アプリを検索し「Qt Creator」をクリックしてください。

	ジェクトを開く		フィードバック	最新ニュース
1	Address Book Example	Building and Running an Example		Welcome to Qt Creator 2.3
сх Энд 1	99: tutorial, parti, tutorial, addressbook           Non-information         Imple Advects Data         Imple Advects Data           Matrix         Address         Imple Advects Data         Imple Advects Data	91: at creator, build, compile		The Qt Creator Team This release adds lots of new features as well as a great amount of bug fixe • Example and tutorial browsing
Г f		Image: A set of the s		with descriptive texts, and filtering for examples matching a keywoi • Enhanced C++ coding style option with indent setting
	This first part covers the design of the basic graphical user interface (GUI) for our address book application.	You can test that your installation is successful by opening an existing example application project.		and alignment settings split up for the different use cases depending or element, including preview and separation betwee
	Creating a Qt Quick Application Using Qt	Creating a Qt Quick Application		global and project specific settings
	979: at auick, aml. components, symbian, visual designer; at creator			<ul> <li>Support for deployment and running to a more general "remote Linux"</li> <li>Support for compiling projects with the Clang compiler</li> <li>Code completion doesn't block the editor any more</li> </ul>
<u>]</u>	This tutorial describes how to use Qt Creator to create a small Qt application, Battery Status, that uses the System Information Mobility API to fetch battery information from the device. The user interface for the application is designed using Qt Oxide Generate the for on his designed using Qt	This tutorial uses basic elements and illustrates basic concepts of Qt Quick.		<ul> <li>Profiling now has i own "Analyze" mode.</li> <li>Symbian got COD/ support, allowing for deployment vi WiFi</li> <li>Support for mode</li> </ul>

Dash ランチャーに Qt Creator が残っていない場合、Qt Creator 起動中にアイコンを右クリックし、 ポップアップメニューにて「ランチャーに固定」をクリックすると Qt Creator が固定されます。

## 4.2 XG シリーズのためのデフォルト設定

XG-3358, XG-BBEXT 用のクロス開発用のデフォルト設定をします。デフォルト設定ですので一度設定するだけです。

① メニュー「ツール」-「オプション」にてオプションダイアログを開きます。

🔞 🗐 🗐 mainwindow.cpp - qt-test1 - Qt Creator	🌢 クイックアクセス		
ブロジェクト 💠 🖓 😌 🗄 🗙 🌵 🔶	C++( <u>C</u> )	Window::~MainWindow()	≑ 行番号: 15, 列位置: 1 🖬 🖌
Ot Test1	ブックマ <i>ー</i> ク( <u>B</u> )		-
a qt-test1.pro	コード貼り付け( <u>C</u> )	k/qt-test1/mainwindow.h	
Image: 1 = 1	マクロ( <u>M</u> )	•	
main.cpp		•	
■ mainwindow.cpp	GIE( <u>G</u> )		
79702 II V	VPAP(L)		
	オブション( <u>0</u> )		
デバッグ 15			
プロジェクト			
F3478			
解析			
0			
~n.7			
			÷
聞いているドキュメント + H ×			
mainwindow.cpp			
qt-test1			
Qt 4lease			
	2 検索結果 マブリケーショ		E
			10

② 「ビルドして実行」を選択し「Qt バージョン」ページを開きます。

XG-3358 または XG-BBEXT の Qt4.8.5 が登録されていなければ「追加ボタン」をクリックします。 登録済みの場合は、手順⑤に進みます。

フィルタ	ビルドして実行				
環境	概要 Qt バージョン	ツールチェイン CMa	ke		
テキストエディタ	名前		qmake のパス		直加
FakeVim	自動検出 ▼ 手動				削除
ヘルプ	PATH (システ.	ム) に含まれる Qt 4.8.1	/usr/bin/qmake-qt4		
<b>}</b> C++					
Qt Quick					
ビルドして実行					
デバッガ					
				20	ーンアッフ
6 7 7 1 7					
■ アナ <del>ラ</del> イザ					
<ul> <li>▼ ナライザ</li> <li>バージョン管理システム</li> </ul>	パージョン名: P	ATH (システム) に含まれ	ເວັ Qt <mark>4.8.1</mark>		
<ul> <li>アナライザ</li> <li>パージョン管理システム</li> <li>コード貼り付け</li> </ul>	パージョン名: P qmake のパス: /u	ATH (システム) に含まれ sr/bin/qmake-qt4	13 Qt 4.8.1	Bro	wse
<ul> <li>アナライザ</li> <li>バージョン管理システム</li> <li>コード貼り付け</li> <li>Linux デバイス</li> </ul>	パージョン名: P qmake のパス: /u デスクトップ 向け (	ATH (システム) に含まれ ısr/bin/qmake-qt4 Qt バージョン 4.8.1	າວັ Qt 4.8.1	Bro	owse
アナライザ アナライザ ゴード貼り付け Linux デバイス	バージョン名: F qmake のパス: /u デスクトップ 向け ( ヘルパ: QML ダンプ	ATH (システム) に含まれ sr/bin/qmake-qt4 Qt バージョン 4.8.1	₩ã Qt 4.8.1	Bro	owse 詳細 、

③ 「qmake 実行ファイルの選択」ダイアログにて、XG-3358 または XG-BBEXT の Buildroot システムのホスト qmake を選択します。

機種(開発キット)	qmake のパス
XG-3358(LK-3358-A01)	~/xg3358-lk/buildroot-2013.11-xg3358-X.X/output/host/usr/bin/qmake
XG-BBEXT	~/xgbbext-lk/buildroot-2013.11-xgbbext-X.X/output/host/usr/bin/qmake

🥘 qmake 実行ファ	マイルの選択						
🖋 🔺 🗟 guest	xg3358-lk	buildroot-2013.11-xg3358-1.0	output	host usr	bin		
場所(P)	名前					サイズ	最終変更日
Q 検索 ∅ 最近開いた…	🔷 qmake					11.8 MB	水曜日
◙ guest ■ デスクトップ ■ ファイル・シス							
<ul> <li>□ ドキュメント</li> <li>■ ミュージック</li> <li>□ ピクチャ</li> <li>□ ビデオ</li> </ul>							
÷							qmake ;
					キャン	セル(C)	開<(0)

④ バージョン名を分かりやすいように「Qt 4.8.5(XG-3358)」または「Qt 4.8.5(XG-BBEXT)」に変更します。
 「この Qt バージョンのコードを生成できるツールチェインがありません。」と表示されますので、ページを「ツールチェイン」に切替えます。

	ビルドして実行	
環境	概要 Qt パージョン ツールチェイン CMake	
] テキストエディタ	名前 qmake のパス	追加
FakeVim	自動検出 ▼ 手動	削除
) ヘルプ	PATH (システム) に含まれる Qt 4.8.1 /usr/bin/qmake-qt4	
<mark>}</mark> C++	Qt 4.8.5 (XG-3358) /home/guest/xg3358-lk/buildroot-	-2013.11-xg3358-1.0/(
Qt Quick		
ビルドして実行		
デバッガ		
デバッガ デザイナ		
デバッガ デザイナ アナライザ	(·())	クリーンアッ
デバッガ デザイナ アナライザ バージョン管理システム	(•())	(グリーンアッ
<ul> <li>デバッガ</li> <li>デザイナ</li> <li>アナライザ</li> <li>バージョン管理システム</li> <li>コード貼り付け</li> </ul>	(()) パージョン名: [Qt 4.8.5 (XG-3358)	クリーンアッ
<ul> <li>デバッガ</li> <li>デザイナ</li> <li>アナライザ</li> <li>パージョン管理システム</li> <li>コード貼り付け</li> <li>Lipux デバイス</li> </ul>	(-()) パージョン名: Qt 4.8.5 (XG-3358) gmake のパン /home/guest/xg3358-lk/buildroot-2013.11-xg3358-1.0/outp	クリーンアッ ut/host/usr/bin/gmake Browse
デバッガ デザイナ マナライザ		

⑤ 「ツールチェイン」ページにて追加ボタンをクリックし「gcc」を追加します。手動に追加された GCC 行を選択し、 コンパイラのパス、デバッガを「参照」ボタンをクリックして入力します。「適用」ボタンをクリックしツールチェ インの登録をし「Qt バージョン」のページに戻ります。

#### [XG-3358(LK-3358-A01)]

項目	コンパイラのパス
コンパイラ	~/xg3358-lk/buildroot-2013.11-xg3358-X.X/output/host/usr/bin/arm-linux-gnueabihf-gcc
デバッガ	~/xg3358-lk/buildroot-2013.11-xg3358-X.X/output/host/usr/bin/arm-linux-gnueabihf-gdb

#### [XG-BBEXT]

項目	qmake のパス
コンパイラ	~/xgbbext-lk/buildroot-2013.11-xgbbext-X.X/output/host/usr/bin/arm-linux-gnueabihf-gcc
デバッガ	~/xgbbext-lk/buildroot-2013.11-xgbbext-X.X/output/host/usr/bin/arm-linux-gnueabihf-gdb

イルタ	ビルドして実行		
環境	概要 Qt バージョン	ソールチェイン CMake	
テキストエディタ	名前	タイプ	追加,
🚡 FakeVim	▼ 自動検出	自動検出	Clone
シーヘルプ	GCC (x86 32bit)		削除
} C++	▼ 手動	手動	133154
Ot Quick	ucc		
レルドして実行			
デバッガ			
▲ デザイナ			
アナライザ			
バージョン管理システム			
	コンパイラのパス( <u>C</u> ):	/buildroot-2013.11-xg3358-1.0/output/host/usr/bin/arm-linux-gnueabihf-gcc	参照
	ABI( <u>A</u> ):	arm-linux-g( 2) arm 2 - linux 2 - generic 2 - elf 2 - 2	zbit ;
LINUX FAILX	デバッガ( <u>D</u> ):	/buildroot-2013.11-xg3358-1.0/output/host/usr/bin/arm-linux-gnueabihf-gdb	参照

 ⑤ 「Qt バージョン」ページにて先ほどのエラーメッセージが消え「qmlviewer がインストールされていません」という メッセージだけが表示されていることを確認し、次に「Linux デバイス」を選択します。

フィルタ	ビルドして実行		
環境	概要 Qt バージョン ツールチェイン CM	ake	
テキストエディタ	名前	qmake のパス	追加
FakeVim	自動検出		削除
シー ヘルプ	PATH (システム) に含まれる Qt 4.8.1	/usr/bin/qmake-qt4	
} C++	A Qt 4.8.5 (XG-3358)	/home/guest/xg3358-lk/buildroot-2013.11-xg3358-1.0/	
Qt Quick			
Qt Quick J ビルドして実行			
Qt Quick ビルドして実行 デバッガ			クリーンアッ
Qt Quick レルドして実行 デバッガ デザイナ	(		「クリーンアッ
<ul> <li>Qt Quick</li> <li>ビルドして実行</li> <li>デバッガ</li> <li>デザイナ</li> <li>アナライザ</li> </ul>	·· (		クリーンアッ
<ul> <li>Qt Quick</li> <li>ビルドして実行</li> <li>デバッガ</li> <li>デザイナ</li> <li>アナライザ</li> <li>パージョン管理システム</li> </ul>	·・・ · · · · · · · · · · · · · · · · · ·	buildroot-2013.11-xq3358-1.0/output/host/usr/bin/qmak	クリーンアッ を Browse
<ul> <li>Qt Quick</li> <li>ビルドして実行</li> <li>デバッガ</li> <li>デザイナ</li> <li>アナライザ</li> <li>バージョン管理システム</li> <li>コード貼り付け</li> </ul>	<ul> <li>・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</li></ul>	buildroot-2013.11-xg3358-1.0/output/host/usr/bin/qmake	クリーンアッ e Browse
<ul> <li>Qt Quick</li> <li>ビルドして実行</li> <li>デバッガ</li> <li>デザイナ</li> <li>アナライザ</li> <li>バージョン管理システム</li> <li>コード貼り付け</li> <li>Linux デバイス</li> </ul>	・・ バージョン名: Qt 4.8.5 (XG-3358) qmake のパン /home/guest/xg3358-lk// デスクトップ 向け Qt バージョン 4.8.5 qmlviewer がインストールされていません	buildroot-2013.11-xg3358-1.0/output/host/usr/bin/qmak	クリーンアッ c Browse

⑦ 「Linux デバイス」の「デバイス構成」のページにて「追加」ボタンをクリックします。

	Linux デバイス		
■ 環境	デバイス構成 MeeGo Qemu	設定	
テキストエディタ	構成( <u>C</u> ):		\$ 追加( <u>A</u> )
FakeVim	27 mil (bil)		削除( <u>R</u> )
② ヘルプ	-===( <u>N</u> )-		デフォルトに設定
() c++	デバイス種類:		SSH 鍵を生成(G)
Qt Quick	認証方法: ⑧ パスワー	-F 🔘 (#(K)	
🕓 ビルドして実行	ホスト名(H): デバイスの)	Pアド SSH ポート(S):	
🀊 デバッガ	空きボート:	タイムアウト: 秒	
🖌 デザイナ	ユーザー名( <u>U</u> ):		
アナライザ	バスワード( <u>P</u> ):	■ パスワードを表示	
1 バージョン管理システム	秘密鍵ファイル:	参照 デフォルトに設定	
コード貼り付け			
Linux デバイス			
KDiff3			

⑧ 「デバイス構成ウィザードを選択」ダイアログが表示されたら「一般的な Linux デバイス」を選択し「ウィザード開始」 ボタンをクリックします。



⑤ 「新しい一般的な Linux デバイス構成の設定」ダイアログにて接続データを入力し「次へ」ボタンをクリックします。
 「設定が完了しました」と表示されたら「完了」ボタンをクリックし「デバイス構成」画面に戻ります。

項目	qmake のパス
この構成を識別する為の名前	XG-3358 または XG-BBEXT
デバイスのホスト名か IP アドレス	ターゲットボードの IP アドレス「192.168.128.200」
デバイスにログインする為のユーザ名	root
認証方式	パスワード
パスワード	default に設定したパスワード「xg3358&bbext」

	[
この構成を識別する為の名前:	XG-3358
デバイスのホスト名かIPアドレス:	192.168.128.200
デバイスにログインする為のユーザー名	: root
忽証方法:	◉ パスワード ○ 鍵
ペスワード:	*****
ユーザーの秘密鍵を含んだファイル:	/home/ouest/.ssh/id_rsa 参照

⑩ ウィザードを閉じると自動的にターゲットボートとの接続確認をします。

[コネクションができたとき]



#### [コネクションができなかったとき]

ホストに接続中 SSH 接続失敗: サーバーからの応答を待機中にタイムアウトしました。 デパイステストが失敗しました。
デバイステストが失敗しました。
開活

「デバイス構成」の「テスト」ボタンをクリックすれば再度接続テストができます。ターゲットのXGボードとの接続が確認されましたら「OK」ボタンを押してオプション設定を終了します。



初めて接続するときなどは接続に時間が掛かることもあります。デバイステストが失敗した場合、他に 原因が見当たらないようでしたら、タイムアウトの設定を大きな値にして再度テストしてください。

## 4.3 プロジェクトの作成

Qt アプリケーションを作成するには、最初に Qt プロジェクトを以下の手順で作成します。

 メニュー「ファイル」–「ファイル/プロジェクトの新規作成」を選択して「新しいプロジェクト」ダイアログを開きます。 プロジェクトの中の「Qtウィジェットプロジェクト」「Qt GUI アプリケーション」を選択し「選択(C)…」ボタンを クリックします。

プロジェクト	🔜 QEGUI アプリケーション
Qt Quick プロジェクト	◎ モバイル Qt アブリケーション
QLウィジェットプロジェクト	
他のプロジェクト	
バージョン管理からインポート	
	Qt Designer ベースのメインウィンドウを含んだ デスクトップ向け Ot アプリケーションを作成し
	ます。
	可能であれば、あらかじめデスクトップ用アプリ
	ケーションとして設定されます。

 ② 名前に「qt-test1」、パスは参照ボタンをクリックしてディレクトリ「/home/guest/qt-work」を作成し 「次へ」ボタンをクリックします。プロジェクトは「/home/guest/qt-work/qt-test1」に作成されます。

Qt GUI アプリ	リケーション
\$ 152	プロジェクト名とパス
ターゲット 詳細 サマリ	Qt4 GUIアプリケーションプロジェクトを生成するウィザードで す。QApplicationを派生し、空のウィジェットを含んだアプリケーションが 生成されます。
	名前: qt-test1
	パス: /home/guest/qt-work 参照 ダプロジェクトのデフォルトパスとして使用
	次へ(N)> キャンセル

③ ターゲットの設定をします。 デスクトップのチェックを確認し、ビルド構成の作成は「Qt バージョンごとに Debug と Release を1つずつ」を選択します。

<ul> <li>         Qt GUI アプリ パス         パス     </li> </ul>	<b>リケーション</b> ターゲット設定 Qt Creator はプロジェクト <b>qt-test1</b> を以下のターゲットに設定できます:	
詳細 サマリ	<ul> <li>✓ ■ デスクトップ</li> <li>ビルド構成の作成: Qt パージョン毎に Debug と Release を1つずつ こ</li> <li>✓ シャドウビルドを使用</li> </ul>	詳細 🔺
	<戻る(B) 次へ(N)>	キャンセル

④ クラス情報にて、クラス名、ヘッダーファイル、ソースファイル、フォームファイルを設定します。ここではデフォルトのまま「次へ」ボタンをクリックします。。

ターゲット 詳細	生成するソースコードフ <sup>-</sup>	ァイルのクラスについての基本情報を指定して下さい。
≥ 詳細		
サマリ	クラス名(C):	MainWindow
i	基底クラス( <u>B</u> ):	QMainWindow
	ヘ <mark>ッダーフ</mark> ァイル( <u>H</u> ):	mainwindow.h
1	ソースファイル( <u>S</u> ):	mainwindow.cpp
	フォームを生成する( <u>G</u> ):	<b>S</b>
	フォームファイル(F):	mainwindow.ui

 ④ プロジェクトに作成されるファイル一覧が表示されます。確認の上「完了」ボタンをクリックし、プロジェクトの作成を 完了します。

バージョン管理システムを使用する場合は「バージョン管理システムに追加」	項目でバージョン管理システムを選択して
ください。	

🔞 Qt GUI アプリ	ケーション		
パス	プロジェクト管理		
ターゲット 詳細	サブプロジェクトとしてプロジェクトに追加:	$\left \langle \hat{\tau}_{\phi}^{*} _{\omega}  ight>$	÷
▶ サマリ	バージョン管理システムに追加( <u>V</u> ):	<なし>	\$ 管理
	ファイルの追加先		
	/home/guest/qt-work/qt-test1:		
	main.cpp mainwindow.cpp mainwindow.h mainwindow.ui qt-test1.pro		
		< 戻る(B) 完了	(F) キャンセル

⑤ qt-test1 プロジェクト 開発用に変わります。

000	mainwindow.cpp - qt-test1 - Qt Ci	reator		
2025 2025 2025 2025 2025 2025 2025 2025	プロジェクト ÷ ⑦	<pre></pre>	↓ <シンボルの選択> *parent):	÷ 行蕃号:1,列位置:1
qt-test1	開いているドキュメント キ 日 X mainwindow.cpp			
Qt 4lease	<ul> <li>A・キーを入力して下さ</li> </ul>	1 問題 2 検索結果 3 アプリケ	ーション出力 4 コンパイル出力	

## 4.4 プロジェクトの設定

Qt アプリケーションプロジェクトの設定をします。

① プロジェクトをクリックし「ビルド設定」ページを選択します。

<ul> <li>マニノアト・RX AC</li> <li>ビルド構成を編集: Qt 4.8.5 (システム) Release</li> <li>⇒ 道加 ▼ 削除 名前を変更</li> <li>概要</li> </ul>				
Qt バージョン: 不正な Qt バージョン	: 管理			
ッールチェイン: ペッールチェイン未通状> シャドウビルド: 図 ビルドディレクトリ: /home/guest/qt-work/qt-test1-build-desktop-Qt_4_8_5Release	2 <b>管理</b> 参照			
ビルドステップ				
<b>qmake:</b> Qt パージコンが設定されていません。qmake を実行できません。	目¥#田 →			
ビルドステップを追加 ▼ クリーンステップ *****				
<ul> <li>クリーンステップを追加 ▼</li> <li>ビルド時の環境変数</li> </ul>	нтим *			
<b>システム環境変数</b> を使用	▼ 昭祥第			

 
 ② 最初に Qt バージョンは「Qt 4.8.5(XG-3358)」を選択します。Qt バージョンの選択を終えると「ビルド構成を編集」は 「Qt 4.8.5(XG-3358) Debug」または「Qt 4.8.5(XG-3358)Relase」になりますので「Qt 4.8.5(XG-3358) Debug」 を選択します。

シャドウビルドが選択されていると、構成によってビルドされるディレクトリが別々になります。

ビルド構成を編集: Qt 4.8.5 (XG-3358) Debug 3 道加 7 削除 名前7	5変更				
· 教史					
0F /55227/- 0F 4.8.5 (YC.3358)	等神				
	告·在				
ビルドディレクトリ: /home/guest/gt-work/gt-test1-build-desktop-Qt 4 8 5 Debug	参照				
既存ビルド構成のインボート					
(An Inc					
ENFXFU					
qmake: qmake qt-test1.pro -r -spec /home/guest/xg3358-lk/buildroot-2013.11-xg3358-1.0/out;	詳細 ▼				
Make: make -w in /home/guest/qt-work/qt-test1-build-desktop-Qt_4_8_5Debug	詳細 👻				
独自プロセスステップ: cp ./qt-test1 /nfs	詳細 👻				
ビルド ステップを追加 🔻					
クリーンステップ					
Make: make clean -w in /home/guest/qt-work/qt-test1-build-desktop-Qt_4_8_5Debug	5半 <b>4日</b> 👻				
クリーン ステップを追加 🔹					
ビルド時の環境変数					
<b>システム環境変数</b> を使用	詳細 マ				

③ ビルド後、出来上がった実行可能ファイルを/nfs ディレクトリに自動的にコピーするように設定します。 「ビルドステップを追加」ボタンをクリックすると、メニューが表れますので「独自プロセスステップ」を選択します。

ビルド構成を編集:	Qt 4.8.5 (XG-3358) Debug 🗧 🗧	追加 ▼ 削除 名前を	変更	
概要				
Qt バージョン:	Qt 4.8.5 (XG-3358)	:)[	管理	
ツールチェイン:	GCC	:	管理	
シャドウビルド:	<b>S</b>			
ビルドディレクトリ:	/home/guest/qt-work/qt-test1-build-desktop-Qt_	4_8_5Debug	参照	
	成社にかり相応の	-1×4		
ビルドステップ				
<b>qmake:</b> qmake qt-t	test1.pro -r -spec /home/guest/xg3358-lk/buildroot	-2013.11-xg3358-1.0/out;	言¥#田 👻	
Make: make -w in /	/home/guest/qt-work/qt-test1-build-desktop-Qt_4_	8_5Debug	詳細 マ	
ビルド ステップを追	• D.C.			
Make 独自プロセスステ				
Make: make clean	-w in /home/guest/qt-work/qt-test1-build-desktop-	Qt 4 8 5 Debug	話羊糸田 🐨	
クリーン ステップを追				
ビルド時の環境変	≤⇒数			
システム環境変数を	と使用		11年1月11日 11日	

 ④ 独自プロセスの「独自プロセスステップの有効化」にチェックを入れ、コマンドに「cp」、コマンド引数に「./qt-test1 /nfs」 と入力します。

~	ビルド設定				
	ビルド構成を編集: Qt 4.	8.5 (XG-3358) Debug	⇒ 追加 ▼ 削除 名前を	変更	
	概要				
	Qt パージョン: Qt	4.8.5 (XG-3358)	:	管理	
	ツールチェイン: GC	C	*	管理	
	シャドウビルド: 🧧				
	ビルドディレクトリ: /ht	ome/guest/qt-work/qt-test1-build-desk	ktop-Qt_4_8_5Debug	参照	
		既存ビル	ド構成のインボート		
	ビルドステップ				
	<b>qmake:</b> qmake qt-test	1.pro -r -spec /home/guest/xg3358-lk/b	ouildroot-2013.11-xg3358-1.0/out;	■ 日本	
	Make: make -w in /hon	ne/guest/qt-work/qt-test1-build-deskto	pp-Qt_4_8_5Debug	計細 マ	
	独自プロセスステップ: c	p ./qt-test1 /nfs		▲ 田林洋路	
	独自プロセスステップの	D有効化 🗹			
	コマンド:	ср		参照	
	作業ディレクトリ:	%{buildDir}		参照	
	コマンド引数:	./qt-test1 /nfs			
	ビルドステップを追加	*			
	クリーンステップ				
	Make: make clean -w in	n /home/guest/qt-work/qt-test1-build-	desktop-Qt_4_8_5Debug	詳細 👻	

⑤ 編集ボタンをクリックして、編集モードに戻します。

## 4.5 サンプル GUI アプリ作成

GUI のデザインは次のような手順にて行います。

① 「qt-test1」-「フォーム」を展開し「mainwindo.ui」をダブルクリックし GUI デザイン用に切り替えます。



② Display Widgets から Label をドラッグ&ドロップでフォームに配置します。

800	mainwindow.ui - qt-test1	- Qt Creator							
12	mainwindow.ui	÷ 🖬 🗳	🦄 🚟 III		: ::: =\$ <b>=</b>				ní ×
Ot	フィルタ	-				-		オブジェクト	クラス
QL.	Spin Box	ここに入力						▼ MainWindow	QMainWindow
よっこそ	Double Spin Box	753 5773 577						🐻 centralWidg	et 🔟 QWidget
10	🕓 Time Edit	10.0 1000 1000						menuBar	QMenuBar
編集	🛗 Date Edit	met Street Street						statusBar	OStatusBar
4	📆 Date/Time Edit								
デザイン	🧿 Dial	nas Seres Ser							
	Horizontroll Bar					1			
-a	Vertical Scroll Bar								
デバッグ	Horizontal Slider								
プロジェクト	T Vertical Sider								
1777									
New York	V Laber SxtLa	bel							
	Text Browser								
	Graphics View								
ヘルプ	12 Calendar								
	LCD Number								
	Progress Bar							フィルタ	+ /-
	Horizontal Line							MainWindow : QMain	Window
	IIII Vertical Line							プロパティ	值
	Qt QDeclarativeView							QObject	
	QWebView							objectName	MainWindow
	Arthur Wits [Demo]							QWidget	
	Qt PathDefdererEx							windowModality	NonModal
	01 XFormRendererEx							enabled	
	Ot GradientEditor							▶ geometry	[(0, 0), 400 × 300]
	Ot GradiendererEx		0 0.			フィルタ		► sizePolicy	[Preferred, Preferred,
	01 PathStr. dererEv	名前	使用中	テキスト	ショートカット	チェック可	ツールチップ	→ ▶ minimumsize	U × U
	Composi onderer	in the	Dero 1			,		MaximumSize	10///215 × 10///215
	Dicelay yamples							<ul> <li>baseSize</li> </ul>	0×0
qt-test i	MultiDagoMidgot							nalette	朝から継承
								▼ font	Δ [Ubuntu 11]
Qt 4Debug								ファミリー	Ubuntu
	TicTacToe							ポイントサイズ	11
4	WorldTimeClock							ボールド	
	Qt 3 Support							斜体	
	Q3IconView	アクションエデ	(タ シグナル	/スロットエディ	9			下線	
1	■ 2.キーを入力  ア下さ	1 230	图 2 検索結	果 マアプリケ	ーション出力	コンパイル思力			
Avenue 1	CAMPOCES				m / J				

 ③ 配置したラベルを選択し、プロパティの「QWidget」 – 「geometry」を展開し、Xに50、Yに50、幅に300、 高さに60をセットします。

	mainwindow.ui* 🗧 🛱	🐴 📑 🔜 I	II ≡ 14 II		55 📕					ní X
Int	フィルタ		- 10 T -						オブジェクト	クラス
LAT T	Item Widgets (Item-Based)	19	こに人力				_		▼ MainWindow	QMaindov
*)CC	List Widget	1.1.1							centralWidget	QWidget
	™i 3 Tree Widget	1000							menuBar	OMenuBar
<b>11</b> .9.	Table Widget	100							mainToolBar	QToolBar
1	<ul> <li>Containers</li> </ul>		Fexti abel			1000			statusBar	QStatusBar
デザイン	Group Box									
	Scroll Area		194111111							
=15mH	Tool Box	1.53								
~	Tab Widget	-								
	Stacked Widget	1 223								
プロジェクト	Erame									
1246	1 utidaet	1.00								
解析	widget	0								
0	MolArea									
ヘルプ	Dock Widget									
	Combo Box									
	Font Combo Box									
	AND LINE Edit									
	All Text Edit								C (	
	All Plain Text Edit								JANY	<b>-</b> - /-
	1. Spin Box								label : QLabel	. 6
	Double Spin Box								70/71	<u>a</u> -
	🕓 Time Edit								QObject	
	Date Edit								objectName l	abel
	📷 Date/Time Edit								Qwidget	7
	🥥 Dial								Enabled	(50, 50) 300
	📻 Horizontal Scroll Bar			Ø.	_	74	ルタ		x geometry	0
	Vertical Scroll Bar	名前	使用	中	テキスト	ショートカット	チェック可	ツールラ	Y	0
	Horizontal Slider								幅	00
abitest1	🔶 Vertical Slider								高さ (	i0
	Display Widgets								▶ sizePolicy	Preferred, Pr
	🚫 Label								▶ minimumSize 0	) × 0
Qt 4Debug	<sup>9</sup> All Text Browser								▶ maximumSize 1	6777215 × 1
	Craphics View								▶ sizeIncrement 0	)×0
No.	Calendar								▶ baseSize	× 0
		ac			.00.			) ))	palette 3	見から離承
2	CC Number	- アク:	ションエディタ	シグナル	/スロットエディ	9			* ront	A [Ubuntu, 11]
-	P・キーを入力して下さ	問題 2 検索	- 新果 3 アプ	リケーショ	ン出力 4 コ	ンパイル出力				

④ プロパティで「QLabel」 – 「text」に「Hello, world」と入力します。

800	mainwindow.ui - qt-test1 - Qt Creato	r						
	mainwindow.ui* 💠 🖷 🗳	I H I H I H I	0 00 55 🛝			_		uí X
Ot	フィルタ				-		オブジェクト	クラス
Q.	Item Widgets (Item-Based)	ここに人力					▼ MainWindow	QMai
aret	List Widget	ford sand sand sand s					▼ iii centralWidg	et 🛛 QW
L <sub>D</sub> h	* 8 Tree Widget						menuBar	OMenu
総第	Table Widget						mainToolBar	QToolE
1	Containers	Hello world					statusBar	QStatu
デザイン	Group Box							
	Scroll Area							
715417	Tool Box							
	Tab Widget							
	Stacked Widget							
プロジェクト	Frame							
	Widget							
解析	MdiAraz							
2	Dack Wideot							
ヘルプ	Dock Widget							
	T Seet Combo Box							
	Font Combo Box							
	AT Taut Falls						(+( iii	) )))
							フィルタ	+ /-
	Au Plain lexc Edic						label : QLabel	
	Spin Box						プロパティ	值
	Double Spin Box						autoFillBackgro	
	U Time Edit						styleSheet	
	Date Edit						▶ locale	Japanese, J
	Date/Time Edit				7110		InputMethodHints	ImhNone
	🥥 Dial	2** #P#		See 1 deal	ブールクテ	N	QFrame	Matan
	Horizontal Scroll Bar	名刑 使用中	TTAN	ショートカット	テエックリ	ワールチップ	frameShadow	Plain
	🔋 Vertical Scroll Bar						lineWidth	1
qt-test1	- Horizontal Slider						midLineWidth	0
	🕈 Vertical Slider						- OLabel	10 N
Qt 4Debug	Display Widgets						text	Hello world
	🏷 Label						textFormat	AutoText
	AI Text Browser						pixmap	
	🚯 Graphics View						scaledContents	
	12 Calendar	アクションエディタ シグナ	ル/スロットエディ	9			alignment	左端揃え,中・
	■ ₽ キーを入力して下さ 11	問題 2 検索結果 ミアブリ	ケーション出力	4 コンパイル出力			N.S.	

⑤ プロパティの「QWidget」-「font」-「ポイントサイズ」を 32 にします。



⑥ 次にアプリケーションの終了ボタンを追加します。Buttonsの「Push Button」をフォームの右下端にドラッグ&ドロップで配置します。

mainwindow.ut* the	2 4
2       2       77.2       77.2         Image: Second Seco	
Layouts     Vertical Layout     Vertical Layout     Vertical Layout     Vertical Layout     Vertical Layout     Vertical Layout     Spacers     Spacers     Vertical Spacer     Vertical Vertical     Vertical Spacer     Vertical Spacer     Vertical Vertical     Vertical     Vertical Vertical     Vertical Vertical     Vert	6
Vertical Layout     Horizontal Layout     Horizontal Layout     Horizontal Layout     Horizontal Layout     Hello, world     Hello, world     Hello, world     StatusBar     QMenuBa     statusBar     QStatusB     Soft     StatusBar     QStatusB     Soft     So	ndow
Horizontal Layout     Horizontal Layout     General Layout     Hello, world     Hello, world     Hello, world     Spacers     Wertical Spacer     Workical Spacer	jet
Grid Layout     Grid Layout     Spacers     Soft Layout     Spacers     Subton     Subton     Subton     CommandButton     CommandButton     Soft     S	r
Image: StatusBar     StatusBar     QStatusBar       Spacers     Hello, world     statusBar     QStatusBar       Image: Spacers     StatusBar     QStatusBar     QStatusBar       Image: Spacers     StatusBar     StatusBar     QStatusBar       Image: S	
PHY       Spacers         PHY       Spacers         PHY       Mil Horizontal Spacer         PHY       Buttons         Buttons       Push Button         Push Button       Push Button         Push Button Box       Push Button         Push Push Push Push Push Push Push Push	ar
Image: Space Spac	
Prov       Eventical Spacer         Buttons       Buttons         Push Button       SushButton         Radio Button       Galde Button         Button Sox       SushButton         Button Box       SushButton         List View       Zalk-2         Button Box       Itabel: QLabel         Ibit View       Zalk-2	
Buttons  PushButton  PushButton  Padio Button  Padio Button Padio Button  Padio Button Padio Butt	-
Image: Push Button       PushButton         Image: Push Button       Image: PushButton         Image: PushButton       Image: PushButton	
22/221       W Tool Button         Image: Badio Button         I	
Radio Button     Section 24/02     CommandButton     CommandButton     Section 24/02     List View     Section 24/02     List View     Iabel: QLabel     Jabe View     Jabe View     Jabe View	
Image: Second secon	
™ Uster Viewel-Based)           □ List View           ™IS Tree View           ™IS Tree View           □ Ide /: QLabel           □ Ide /: QLabel           □ Ide /: QLabel	
List View         2×1№9           %3 Tree View         Iabei: QLabel           Table View         7µ7+	
	+- /-
Table View	
	6
Column View 高さ 60	
ttem Widm-Based)	referred,
List Widget   V 0 × 0	
Tree Widget 16777215 × 1	6777215
Table Widget 0 × 0	-
Containers 0×0	
「Group Box P F F F O A F F F F O A F F F F F O A F F F F	
Scroll Area	32]
Statked widget	
Qt 4. Debug Trane Do the second descent and t	
Widget UNAW	
MdiArea フレート MdiArea フレーチェイリアス デフォルト係め	
Dock Widget	
Input Widgets アクションエディタ シグナル/スロットエディタ mouse Tracking	
■ P キーを入力して下さ… 1 問題 2 検索結果 3 アプリケーション出力 4 コンパイル出力	0

 ⑦ 配置したラベルを選択し、プロパティの「QWidget」 – 「geometry」を展開し、Xに280、Yに200、幅に90、 高さに40をセットします。



⑧ プロパティにて QObject の Object Name を「btnQuit」に、QAbstructButton の Text プロパティを「終了」に します。

800	mainwindow.ui - qt-test1	- Qt Creator		
	mainwindow.ui*	• ¶ ¶ ¶ N II = H II = H II + H		
Ot	フィルタ		オブジェクト	クラス
ようごそ	Layouts	- 22к. ХЛ =	<ul> <li>MainWindow</li> <li>CentralWidge</li> </ul>	QMainWindow t 🔟 QWidget
1.00	III Horizontal Layout		btnQuit label	QPushButton QLabel
	Grid Layout		menuBar	QMenuBar
	Borm Layout	Hello world	statusBar	OStatusBar
デザイン	Spacers	ricito, world		
	Horizontal Spacer			
デバッグ	Vertical Spacer			
	Buttons			
プロジェクト	Push Button			
	Tool Button	• 終了 •		
	Radio Button			
2010T	Check Box			
8	Ocommand Button			
	X Button Box			
	Item Viewel-Based)			1.
	List View		フィルタ	+ /-
	Tree View		btnQuit: QPushButtor	
	Table View		プロパティ	值
	Column View		autoFillBackgro	
	Item Widm-Based)		styleSheet	
	List Widget		▶ locale	Japanese, Japan
	Tree Widget		InputMethodHints	ImhNone
	Table Widget		QADSCI ACCOULCON	87 N
	Containers		▶ icon	40 7
	Group Box	😼 🖥 🛍 🛇 🍄-	▶ iconSize	16 × 16
	Scroll Area	名前 使用中 テキスト ショートカット チェック可 ツールチップ	shortcut	
	Tool Box		checkable	
	🛅 Tab Widget		checked	
	Stacked Widget		autoRepeat	
Ot 4Debug	💹 Frame		autoExclusive	
	🕖 Widget		autoRepeatDelay	300
	📑 MdiArea		autoRepeatInte	100
	Dock Widget		QPushButton	-
the second	Input Widgets	アクションエディタ シグナル/スロットエディタ	autoDerault	
1	Combo Pox		derault	
	・ キーを人力して下さ			

	mainwindow.ui - qt-test1	- Qt Creator		
	mainwindow.ui*	• 6 4 5 5 5 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
Ot	フィルタ		オブジェクト	クラス
QL.	Layouts )	ССЕХЛ	▼ MainWindow	QMainWindow
a./	📑 Vertical Layout		centralWidge	et W QWidget
1.1	III Horizontal Layout		label	OLabel
<b>#1.9</b> .	🗱 Grid Layout		menuBar	QMenuBar
1	Borm Layout	Hollo world	mainToolBar	QToolBar
デザイン	Spacers	riello, world	scacusbar	QStatusBar
	Horizontal Spacer			
₹15mH	Vertical Spacer			
-	Buttons			
	Push Button			
プロジェクト	Tool Button	1/27		
	Radio Button	1/2° J		
解析	Check Box			
2	Command Button			
ヘルプ	Button Box			
	Item Viewel-Based)			
	List View		7114	+ - /
	Tree View		btnOuit : OPushButto	n
	Table View		プロパティ	値
	Column View		sizeIncrement	0×0
	Item Widm-Based)		baseSize	0×0
	List Widget		palette	親から継承
	Tree Widget		▼ font	A [Ubuntu, 24]
	Table Widget		ファミリー	Ubuntu
	Containers		ポイントサイズ	24
	Group Box	R 1 1 0 12.	ボールド	24
	Scroll Area	名前 使用中 テキスト ショートカット チェック可 ツールチップ	- 斜体	
	Tool Box		下标	
at tart1	Tab Widget		収用線	
	Stacked Widget		アンチェイリアフ	デフォルト優先
<u></u>	Frame		CUISOL	シームの「つん」の
Qt 4Debug	Widget		mouseTracking	
	MdiArea		focusPolicy	StrongFocus
	Dock Widget		contextMenuPo	DefaultContextMenu
	La Jock Widget		acceptDrops	
N		アクションエディタ シグナル/スロットエディタ	▶ toolTip	
	. キーを入力して下さ	1 問題 2 検索結果 3 アプリケーション出力 4 コンパイル出力		

⑨ プロパティの「QWidget」−「font」−「ポイントサイズ」を 24 にします。

画面上で同じフォントを利用する場合、デフォルトとして登録しておくと便利です。

MainWindow の font プロパティにフォントを設定しておけば、MainWindow 上に配置された font プロパティを持った widgets のデフォルトは Mainwindow の font プロパティが用いられます。

 Quit ボタンをタップしたときに、qt-test1 アプリケーションが終了するようします。そのためには btnQuit にてマウス 右ボタンをクリックしポップアップメニューの「スロットへ移動…」を選択します。

800	mainwindow.ui - qt-test1	i - Qt Creator				
	mainwindow.ui*	* <b>% % % % </b> # = H I # # #	8			ní X
(Ot	フィルタ				オブジェクト	クラス
QL	Layouts	ここに入力			▼ MainWindow	QMainWindow
anct	Vertical Layout	The local local local local local local local local			centralWidge	et 🔟 QWidget
	III Horizontal Lavout				btnQuit	QPushButton
編集	333 Grid Layout	and Same Same Same Same Same Same Same Same			menuBar	OMenuBar
4	B Form Lavout	Halla world			mainToolBar	QToolBar
デザイン	Spacers	nello, world			statusBar	QStatusBar
	Bill Horizontal Spacer					
- A	Wastical Cases					
デバッグ						
	Buccons					
プロジェクト		45				
1200	Tool Button	• 於				
No. 10		11 <b>-</b>	ナキストを変更			
-	Check Box		オブジェクト名を変更			
	Ormand Button		他のクラスへ変更	*		
ヘルプ	X Button Box		ツールチップを変更			
	Item Viewel-Based)		トントを変更			E Contraction of the second se
	List View		スタイルシートを変更		フィルタ	+ /-
	Tree View				btnQuit: QPushButto	n
	Table View		限界サイス	,	プロパティ	值
	Column View		格上げ先を指定		autoFillBackgro	
	Item Widm-Based)		スロットへ移動		styleSheet	
	List Widget		2		Iocale	Japanese, Japan
	* Tree Widget		背面へ移動( <u>B</u> )		inputMethodHints	ImhNone
	Table Widget		前面へ移動(F)		QAbstractButton	
	Containers		切り取り(T)	Ctrl+X	▶ text	終了
	Group Box	<b>k h h O O</b> .	コピー(C)	Ctrl+C	▶ icon	
	Scroll Area	名前 使用中 テキスト ショ	貼り付け(P)	Ctrl+V y	▶ iconSize	16 × 16
	Tool Box		すべてを選択(A)	Ctrl+A	▶ snortcut	
-	Tab Widget		削除(D)		спескаріе	
qt-test?					checked	
			レイアワト	*	autoRepeat	
Qt 4Debug	Frame				autoExclusive	200
	Widget				autoRepeatDelay	100
	MdiArea				oBuchButton	100
	Dock Widget				putoDefault	-
	Input Widgets	アクションエディタ シグナル/スロットエディタ			default	
1	0. +- 57 11 7 T+	1 問題 2 検索結果 2 アプリケーション中	カークコンパイル出力		deredit	
and the second value of th	P. +- & XJUCLE		10 コンパイル田川			

⑪ ここでは単にタップ (クリック) するだけなので、「clicked()」を選択し「OK」ボタンをクリックします。

グナルを選択	
clicked()	QAbstractButton
clicked(bool)	QAbstractButton
pressed()	QAbstractButton
released()	QAbstractButton
togaled(bool)	OAbstractButton

≑ 行番号: 18, 列位置: 14 🛛 🖬 🗙

clo	ose () ;		
800	mainwindow.cpp - qt-test1 - Qt	Creator	. And
	Image: Second secon	<pre>inclinVinDovAcppi include "maintendow," include "maintendow," include "maintendow," include "maintendow," int include "laintendow" (Waintindowce Lisintendow) u-&gt;setupUi(this); u-&gt;setupUi(this); u-&gt;setupUi(this); delate u;; }</pre>	<ul> <li>Mainwindow-on_btriguic_citcked(boo)</li> <li>t sparent) :</li> </ul>



③ Qt アプリケーションのビルドは、Qt Creator メニュー「ビルド」 – 「プロジェクト"qt-test1"をビルド」を選択し ビルドを開始します。

🛞 🗐 💿 mainwing		Shift+Ctrl+B		
プロジェク	☆ すべてリビルド		AainWindow::on_btnQuit_clicked(bool)	≑ 行番号: 20, 列位置: 1 🛛 🖬 ×
0t. 🔻 🖾 qt-t	すべてテプロイ			-
4)524 p 🙀	<b>y y x cyy</b> = <b>y</b>		et *parent) :	
	→ プロジェクト "qt-test1" をビルド	Ctrl+B		
編集	フロジェクト "qt-test1" をリビルド			
× × 📈	プロジェクト quetesti をデフロイ プロジェクトを公開			
79A2	プロジェクト "qt-test1" をクリーン			
	象 qmake 実行			
デバッグ			licked( <mark>bool</mark> checked)	
	▶ 実行	Ctrl+R		
	ビルド/実行ターゲットセレクタを開く	Ctrl+T		
新新 新新				
0				
ヘルプ				
				8
開いている	レント キ 日 ×			
mainwind	ow.cpp*			
mainwind	ow.ui*			
qt-test1				
Qt 4Debug				
				U N
P+ #	を入力して下さ 1 問題 2 検	索結果 🗈 アプリク	アーション出力 4 コンパイル出力	
$\sim$				
	ビルドボタンを使う	っこともつ	し しょう	
L				

保存してないファイルがあるときは、以下のように問い合わせがあります。その場合は「すべて保存」をクリックします。 「ビルド前にすべてのファイルを保存する」にチェックを入れておけば自動的に保存されてからビルドされるようにな ります。



ビルドが正常に終了すると、ディレクトリ「/nfs」に「qt-test1」がコピーされます。

ビルドが始まるとビルド進行インジケータが表示されます。ビルドの結果エラーが無いときは次のように数秒間画面 左下のビルドの部分が緑色になります。

問題に「unused parameter 'checked'[-Wunused-paramter」とウォーニングが出ていますが、これは終了ボタンを 押したときの処理「on\_btnQuit\_clicked」で check 引数を使用してないためです。





エラーがある場合は、ビルド進行インジケータが赤色になり、次のようにエラーメッセージが表示されます。

### 4.6 サンプルの実行

ビルドが正常に終わると開発 Linux サーバの/nfs ディレクトリに実行可能なアプリケーションファイルをコピーされています。 ターゲット側の XG ボードで NFS マウントしてあればそのまま、/mnt/nfs ディレクトリに移動して実行する事ができます。

① 開発用 PC の NFS サーバに接続してないときは、mount します。

/etc/init.d に「K95debug」スクリプトがあ	5る場合は
<pre># /etc/init d/K95debug start</pre>	入力

コマンドで直接 NFS マウントする場合は以下のようにします。

# mount -t nfs -o nolock 192.168.128.210:/nfs /mnt/nfs 🛛 🐲

② qt-test1 が存在することを確認します。

<pre># cd /mnt/nfs</pre>	入力			
# ls qt-test1	入力			
qt-test1				

③ qt-test1 を実行します。

# ./qt-test1 -qws

④ 次のような画面が表示され「終了」ボタンをタッチすると終了します。

Qt	MainWin	dow	- • ×
<u></u>	Hello w	orld	
			終了

# 5 リモートデバッガ

## 5.1 リモートデバッガの設定

XG-3358 または XG-BBEXT 上で稼動する Qt アプリケーションをデバッグできるようにするための設定を行います。

 「プロジェクト」-「実行時の設定」ページに切替えます。初期状態ではデプロイ、実行ともリモートでバッグ用に 設定されていません。

	実行時の設	定	
100	デプロイ		
	メソッド: デブロ	1イしない 🛟 追加 🔻 削除 名前を変更	
	デプロイステップな	al.	
	デプロイステップ	を追加 マ	
	実行		
	実行構成: qt-tes	it1	
	実行ファイル:	/home/guest/qt-work/qt-test1-build-desktop-Qt_4_8_5Debug/qt-test1	
	引数:	Lease and an and a second s	
	作業ティレクトリ:	/home/guest/qt-work/qt-test1-build-desktop-Qt_4_8_5Debug 参照	
		☐ mm,k/3 ( ≠1)	
	デバッガ:	□ QML デバッグボート: 3768 〕 前提条件は?	
	実行時の環境到	変数	
	ビルド時の環境多	<b>変数</b> を使用	詳細 マ
	アナライザ設定	Valorind	1944年 -
	0.10 -3 80-9-82 AC	- segure	MT 374

② デプロイ(配布)の追加をします。

「デプロイ」-「メソッド」の「追加」をクリックして「リモート Linux ホストにデプロイ」を選択します。 「名前を変更」をクリックして名前を「XG-3358 にデプロイ」または「XG-BBEXT にデプロイ」にします。



デプロイ設定をするとビルドが成功すればターゲットボードに実行ファイルを自動的に配布すること ができます。

「4.5 サンプルの実行」で行った Ubuntu 側の/nfs ディレクトリに実行ファイルをコピーして、ター ゲットボード側で NFS マウントして実行する必要はなくなります。 実行ファイル以外に沢山の画像、サウンドファイルやデータファイルなどを配布する必要がある場合 は、NFS を利用した方が便利ですので、デバッグ環境にあわせて NFS の使用は検討してください。

③ ターゲットボードの配布先ディレクトリを指定します。

「<ターゲットパス未設定>」となっておりマウスカーソルを近づけると「項目の追加・削除にはプロジェクトファイル を編集して下さい。」とヒントが表示されます。「編集」にてプロジェクトファイル qt-test1.pro をクリックしてプロジ ェクトファイルを開き、ターゲットにデプロイするパスを指定するために以下の2行を追加します。



プロジェクトファイルの編集が終わりましたらメニュー「ファイル」-「"qt-test1.pro"の保存」でプロジェクトファイル を保存します。



プロジェクトに戻ると先ほど赤字で出ていた「<ターゲットパス未設定>」は「/root」に変更されています。

④ リモートでの実行環境の設定をします。

「実行構成」の「追加」をクリックし「qt-test1(リモートの一般的な Linux ホスト上)」を選択します。 「名前を変更」をクリックして「qt-test1 (リモートデバイス上)」を「XG-3358」または「XG-BBEXT」に変更します。 Qt アプリケーションは qws サーバが無いと実行できませんので、ターゲットボードですでに qws サーバが動いている 場合を除き、引数に「-qws」を追加します。

実行時の設定			
デプロイ			
メソッド: XG-3358にデプロイ :	追加 ▼ 削除 名前	を変更	
	[		
ナハイス構成:	XG-3358 2 7/	1人構成の管理	
サノフロシェクトにインストールする	(1) 167	17 1 2 1 8 1	
/home/guest/gt-work/gt-tes 8	5 Debug/gt-test1 /home.	/root	
/			
			- P
っこ SFTP 経由でファイルをアップロード	で使用中のデバイス: XG-3358		>)
○ (	で使用中のデパイス: XG-3358	)	))) 詳細 v
○て SFTP 経由でファイルをアップロード デブロイステップを追加 ★	 で使用中のデパイス: XG-3358	)	) 詳細
C:て SFTP 経由でファイルをアップロード デブロイステップを追加 ▼ 実行	 で使用中のデバイス: XG-3358	)	)) 8¥40 v
○:C SFTP 経由でファイルをアップロード デブロイステップを追加 ★ 実行 実行構成: XCC3358 ◆	で使用中のデバイス: XG-3358 	·	)) 27年4日 - 平
<ul> <li>SFTP 経由でファイルをアップロード</li> <li>デブロイステップを追加 *</li> <li>実行構成: XG-3358</li> </ul>	で使用中のデバイス: XC-3358 追加 * 削除 名前	· · · · · · · · · · · · · · · · · · ·	2) 2740 - v
cr     sFTP 経由でファイルをアップロード     デプロイステップを追加 *      安行構成: XG-3358 :      デバイス構成:	で使用中のデバイス: XC-3358 追加 * 削除 名前 XC-3358	を変更 デバイス構成の管理 デバ:	) 詳細 * ガの設定
CC     SFTP 経由でファイルをアップロード     デプロイステップを追加 ▼     P     安行構成: XCr3358 :     デバイス構成:     ホスト上の実行可能ファイル:     デバイス構成:     ホスト上の実行可能ファイル:     デバイス構成:	で使用中のデバイス: XC-3358 道加 ▼ 削除 名前 XC-3358 /home/cost/obtext1-buil	を変更 デバイス構成の管理 デバ: d-desktop-Qt_4_8.5Debug/q	) 詳細 マ ガの設定 btest1
	で使用中のデバイス: XG-3358 道加 v 削除 名前 XG-3358 /home/guest/qt-work/qt-test1-buil /home/root/qt-test1	を変更 デバイス構成の管理 デバ d-desktop-Qt_4_8_5Debug/q	〕 詳細 ▼ /ガの設定 btest1 //5を使う
C     C     C     SFTP 経由でファイルをアップロード     デブロイステップを追加 ▼     ア     ズク     て     ズク     ズク	で使用中のデバイス: XG-3358 道加 v 削除 名前 XG-3358 /home/guest/qt-work/qt-test1-buil /home/root/qt-test1	を変更 デバイス構成の管理 デバラ d-desktop-Qt_4_8.5Debug/q □ 代わりにこのコマン	♪) 詳細 ♥ // <b>/の設定</b> t-test1 //Fを使う
<ul> <li>・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</li></ul>	で使用中のデバイス: XC-3358 道加 * 創除 名前 XC-3358 /home/guest/qt-vork/qt-test1-buil /home/root/qt-test1	を変更 デバイス構成の管理 デバジ d-desktop-Qt_4_8_5Debug/q □ 代わりにこのコマン	♪ 詳細 ▼ がの設定 bbest1 バを使う
SFTP 縦由でファイルをアップロード デブロイステップを追加 * 実行構成: XC-3358 デバイス構成: 水C-3358 デバイス構成: ホスト上の実行可能ファイル: デバイス上の実行可能ファイル: デバイス上の実行可能ファイル: 引数: デバッグタイプ:	で使用中のデパイス: XC-3358 道加 ▼ 削除 名前 XC-3358 /home/guest/qt-work/qt-test1-buil /home/guest/qt-test1 	を変更 デバイス構成の管理 デパ d-desktop-Qt_4_8_5Debug/q ロー代わりにこのコマン ++と QML	〕 詳細 マ <b>ソガの設定</b> btest1 バを使う

以上でリモートデバッグの設定が完了しました。

## 5.2 リモートデバッグ

① デバッグを開始します。

デバッグを開始するには以下の方法があります。

- a. メニューの「デバッグ」-「デバッグ開始」-「デバッグ開始」でデバッグを開始します。
- b. 「デバッグ」画面に切替えて、デバッグ開始ボタンをクリックします。
- c. 画面左下のデバッグ開始ボタンをクリックします。



デバッグ関係のスピードボタンには次のものがあります。



左から、

実行/続行 デバッガを終了 ステップオーバー ステップイン ステップアウト 命令操作モード切替 となっています。 またデバッグメニューにはスピードボタンにある機能以外のものも登録されています。

ブレークポイントの設定/解除は行番号の左側をクリックします。

# 6. Qt プログラミング Tips

### 6.1 フォントファミリの指定

Qt Creator を使って GUI デザインするときにフォントファミリ指定で困ることがあります。たとえば XG-3358 また XG-BBEXT ターゲットボードに IPA の明朝とゴシックフォントをインストールした場合、GUI デザイン設計においてウィジ ェットのフォントごとに明朝、ゴシック、明朝(プロポーショナル)、ゴシック(プロポーショナル)を指定したいことがあ ります。

通常フォントファミリは、プロパティの「font-ファミリ」で設定できますが、Ubuntu 上の QtCreator では選択するフォントファミリー覧には日本語のファミリ名しか列挙されません。そしてそのまま日本語のフォントファミリ名を選択、XG ターゲット用の実行ファイルを生成して XG ボードで動かすと期待したフォントで表示されないことがあります。

なぜならば XG ボード側ではフォントファミリは英字のみで検索されるからです。そのため日本語フォントファミリ名で指定 してあるとフォントが見つからないのでデフォルトフォントが使用されます。困ったことに QtCreator では「font-ファミリ」 にはフォントファミリ名を直接入力することができません。したがって回避方法としては一旦日本語のフォントファミリ名を 選択し Qt Creator を終了した後、プロジェクト内の「\*.ui」ファイルを編集します。

「IPA P 明朝」を指定したときの mainwindow.ui ファイル

<property name="font"> <font> <family>IPA P 明朝</family> <pointsize>36</pointsize> </font> </property>

XG-3358、XG-BBEXT のフォントファミリ名に変えるため以下のように編集します。

<property name="font"> <font> <family>IPAPMincho</family> <pointsize>36</pointsize> </font> </property>

再度、QtCreator を起動しプロジェクトを開くと、Ubuntu 側に該当フォントがインストールされていれば指定されたフォントで表示されます。フォントファミリ名は英字が基本ですので QtCreator や LibreOffice で表示される日本語のフォント名で 指定しなくても、QtCreator 上では同じフォントで表示されます。



フォントファミリ名を知りたいときは、フォントファイルをクリックするとフォントビューワーが開きフォントの表示例と明細が表示されます。フォントファミリ名は Name 欄に記載されています。



フォントファイルは、Qtの場合、次のディレクトリに保存されています。 ~/xg3358-lk/buildroot-2013.11-xg3358-X.X/output/target/usr/lib/fonts または

~/xgbbext-lk/buildroot-2013.11-xgbbext-X.X/output/target/usr/lib/fonts

Ubuntu にこのフォントがインストールされていない場合は、右下の「Install Font」ボタンをクリックすることによってインストールができます。

# 7. 関連情報

Qt Creator に関しての情報は以下の書籍および、サイトが参考になります。

名前	著者	出版社
実践 Qt プログラミング	Mark Summerfield	オライリージャパン
入門 Qt 4 プログラミング	Jasmin Blanchette	オライリージャパン
	他	
Qt Quick ではじめる	折戸 孝行	アスキー書籍
クロスプラットフォーム UI プログラミング		

#### Table 7-1 参考書籍

サイト名	URL
digia	http://www.digia.com
Qt	http://qt.digia.com
Qt wiki(日本語)	http://qt-project.org/wiki/Wiki_Home_Japanese
SRA Qt	http://www.sra.co.jp/qt/licence/
Sitara Linux Training:	http://processors.wiki.ti.com/index.php/Sitara_Linu
Hands on with QT	x_Training:_Hands_on_with_QT

Table 7-2 参考 Web サイト

#### ご注意

- ・本文書の著作権は、株式会社アルファプロジェクトが保有します。
- 本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は、株式会社アルファプロジェクトが保有します。
- ・本文書に記載されている内容およびサンプルプログラムについての技術サポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。
- ・本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
- ・本文書で記載されているプログラム、データ等は執筆時点のライセンス規定をもとに解説してあります。ライセンス規定は変更になる場合もありますので
   それらソフトウェア、データをご利用の都度ライセンス規定をご確認ください。
- ・本文書の内容は、将来予告なしに変更されることがあります。

#### 商標について

・Windows®の正式名称は、Microsoft®Windows®Operating System です。
 Microsoft、Windows、Windows NTは、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
 Windows®7、Windows®Vista、Windows®XPは、米国 Microsoft Corporation.の商品名称です。
 本文書では下記のように省略して記載している場合がございます。ご了承ください。
 Windows®7は、Windows 7もしくはWin7
 Windows®Vistaは、Windows VistaもしくはWinVista
 Windows®XPは、Windows XPもしくはWinXP

・その他の会社名、製品名は、各社の登録商標または商標です。

### ALPHA PROJECT CO.,LTD.

株式会社アルファプロジェクト 〒431-3114 静岡県浜松市東区積志町 834 http://www.apnet.co.jp E-MAIL : query@apnet.co.jp