XG-335x Qt+OpenCV の PC 用アプリケーション開発

Rev1.0 2014/11/12



1.	概要		1
	1.1	はじめに1	
	1.2	構成1	
2.	環境構	築	2
	2.1	前提環境2	
	2.2	Qt インストール2	
	2.3	OpenCV インストール3	
3.	プログ	ラミング	6
	3.1	Qt Creator の起動6	
	3.2	プロジェクトの新規作成7	
	3.3	プロジェクトの設定変更10	
	3.4	画像を表示する11	
	3.5	画像処理を行う15	

表記

●コマンドラインの表記

本ドキュメントには、コマンドラインで入力する操作手順が記載されております。 操作は PC で行います。記述について以下に記載します。

プロンプトは、『**\$**』で記載します。

実際のプロンプトには、カレントディレクトリ等が表示されますが、本ドキュメントでは省略します。 なお、省略時には、コマンドプロンプトの前に、 **(省略**)と表記します。

本ドキュメント中での入力では、以下のように表現し、入力の最後には、 🛵 があります。

例:make コマンドを実行する場合の表記

省略 \$ make 入力

コマンドによっては 1 つのコマンドが複数行で記載されている場合もあります。 その場合には、2 行目以降の入力では ENTER キーを押さずに続けて入力し、 (ア) の表記がある行の最後で ENTER キーを 入力してそのコマンドを実行してください。

例:2行続いてコマンド入力がある表記

●ソースコードの表記

ソースコードを表示して説明する場合があります。以下の表記になります。

例:

#include <opencv2/opencv.hpp>

1. 概要

1.1 はじめに

本ドキュメントでは、Qt と OpenCV を利用した PC 上で実行するアプリケーションの開発方法について解説します。

- ・環境構築
- ・プログラミング例

PC 用の Qt+OpenCV アプリケーションは、クロス開発環境で XG シリーズ用に容易に移植することができます。XG シリー ズ用アプリケーションの事前検証として開発することも有用です。そのため、本ドキュメントでは同じバージョン構成を想定 して解説しています。



1.2 構成

ソフトウエア構成

Ubuntu 12.04 LTS Qt Creator 2.4.1 Qt 4.8.1 OpenCV 2.4.9 アプリケーションノート AN5009

2. 環境構築

2.1 前提環境

本章では、次の環境の上に構築する手順を説明しますので、事前にご準備ください。

- Ubuntu 12.04 LTS 💥
- ・ネットワーク設定 : インターネット接続設定。ファイルダウンロードを行うために必要となります。
- ※ Ubuntu 導入の詳細は「Linux 開発インストールマニュアル for XG-335x」を参照してください。

2.2 Qt インストール

Qt および Qt Creator のインストールは、apt-get コマンドにて行います。 ターミナルを開き、以下のコマンドを実行します。

① Qt インストール前にパッケージを更新します。

省略 \$ sudo apt-get update < ^{入力}	
省略 \$ sudo apt-get upgrade 🖓	
、 次の様な問い合わせがありますので、Enter キー または、Y + Enter キーで続行してください。	
続行しますか [Y/n]? 🛛 🐼	

upgrade の実行時に、GRUB (grub-pc)の設定問い合わせがでる場合があります。 ご使用の環境にもよりますが本ドキュメントでの前提環境では GRUB は不要ですので、GRUB パッ ケージをインストールせずに続行してください。(インストールデバイスの選択画面ではデバイスを 選択せずに進んでください。)

② Qt Creator パッケージをインストールします。

省略 \$ sudo apt-get install qtcreator 🛛 🔧

このコマンドにより、Qt 4.8.1 と Qt Creator 2.4.1 がインストールされます。

2.3 OpenCV インストール

sourceforge.net から OpenCV 2.4.9 のソースファイルをダウンロードしてビルド&インストールする手順を説明します。

- YG シリーズ開発キットの OpenCV は 2.4.9 です。 事前検証などの用途で PC 用アプリケーションのビ
 - 事前検証などの用途で PC 用アプリケーションのビルド&実行を行う場合には、PC 動作用の環境も 開発キットと同じ 2.4.9 をインストールします。



Ubuntu 12.04 LTS の apt-get コマンドによってインストールされる OpenCV パッケージは 2.3.1 です。

OpenCV 2.4.9 のインストール

① ホームディレクトリに OpenCV という名前の作業ディレクトリを作成します。

省略	\$	cd	入力	
省略	\$	mkdir	OpenCV	入力
省略	\$	cd Op	enCV 🔹	5
省略	\$	pwd	入力	
/home	e/s	guest/	OpenCV	

② インストール済みの ffmpeg と x264 パッケージを削除します。

省略 \$ sudo apt-get -qq remove ffmpeg x264 libx264-dev 🛛 💀

③ OpenCV パッケージのビルドと実行に必要なパッケージを追加します。

\$ sudo apt-get -qq install libopencv-dev build-essential checkinstall cmake pkg-co nfig yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc 1394-22-dev libxine-dev libgstreamer0. 10-dev libgstreamer-plugins-base0. 10-dev libv41-d ev python-dev python-numpy libtbb-dev libqt4-dev libgtk2. 0-dev libfaac-dev libmp31ame-d ev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcoredev x264 v41-utils ffmpeg

④ OpenCV 2.4.9 のソースを「sourceforge.net」からダウンロードします。

⑤ ソースの zip を解凍します。

省略 \$ unzip OpenCV-2.4.9.zip 🗘

⑥ 解凍したディレクトリに build ディレクトリを作成します。



⑦ cmake コマンドでビルドファイルを作成します。

 \$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB

 =ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON -D INSTALL_P

 YTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON ...

⑧ make コマンドでビルドを実行します。(実行環境によりますが、ビルドには時間がかかります)

省略 \$ make -j2 🔨

(多くの cc1plus の警告が表示されますが、そのまま続行してください。)

⑨ checkinstall コマンドでインストールします。

省略 \$ sudo checkinstall 入2

途中で問い合わせがありますので、以下のように回答して進めます。

The package documentation directory ./doc-pak does not exist. Should I create a default set of package docs? [y]: n

このパッケージの説明を書いてください 説明の末尾は空行かEOFにしてください。 >>

```
*****
**** Debian package creation selected ***
*****
このパッケージは以下の内容で構成されます:
0 - Maintainer: [ guest@guest-VirtualBox ]
1 - Summary: [ Package created with checkinstall 1.6.2 ]
         [ build ]
2 - Name:
3 - Version: [ 20141028 ]
4 - Release: [1]
5 - License: [ GPL ]
6 - Group: [checkinstall]
7 - Architecture: [ i386 ]
8 - Source location: [ build ]
9 - Alternate source location: [ ]
10 - Requires: [ ]
11 - Provides: [ build ]
12 - Conflicts: [ ]
13 - Replaces: [ ]
変更するものの番号を入力してください。Enterで続行します: 🗤
```

You probably don't want them to be included in the package. それらを表示しますか? [n]: �� それらをパッケージから除外しますか?(yesと答えることをおすすめします) [n]:y �� ⑩ Idconfig を実行して共有ライブラリの依存関係情報を更新します。

省略 \$ sudo sh -c 'echo "/usr/local/lib" > /etc/ld. so. conf. d/opencv. conf' 本
省略 \$ sudo ldconfig 本

以上で OpneCV 2.4.9 のインストールは完了です。

ファイルは次のディレクトリにインストールされています。

ライブラリファイル /usr/local/lib インクルードファイル /usr/local/include

OpenCV 2.3 のアンインストール

「OpenCV 2.4.9 のインストール」の③で行った libopencv-dev パッケージのインストールによって、OpenCV 2.3.1 のパッケージがインストールされています。これを削除します。

省略 \$ sudo apt-get remove libopencv-core2.3 <²27

opencv-core に依存する他のパッケージも削除されます。

アプリケーションノート AN5009

3. プログラミング

Qt Creator で OpenCV ライブラリを利用するアプリケーションを開発する手順を説明します。

3.1 Qt Creator の起動

 Dash ホームアイコンをクリックします。(Dash ホームは、Ubuntu 画面の左側にあるパネルの一番上のアイコンです。) 検索の文字入力部に"Qt"と入力すると、"Qt Creator"が表示されます。



Fig 3.1-1 Qt Creator 検索

Qt Creator をクリックします。



Fig 3.1-2 Qt Creator 画面



3.2 プロジェクトの新規作成

新規作成

メニューの ファイル -> ファイル/プロジェクトの新規作成… を選択します。

プロジェクト	🛋 Qt GUI アプリケーション						
Qt Quick プロジェクト	◎ モハイル QE アノリケーション						
Qt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	N						
ファイルとクラス C++	L3						
Qt							
QML	Qt Designer ペースのメインウィンドウを含んだ						
GLSL	テスクトップ回け QL アプリゲーションを作成し ます						
その他	6. 7 0						
	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。						

Fig 3.2-1 新規作成

ここでは「Qt ウィジェットプロジェクト」の「Qt GUI アプリケーション」を指定し、選択ボタンを押します。

プロジェクト名とパス

> パス	ノロシェクト石とハス
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 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: opencv01 パス: /home/guest/projects 参照
	□ プロジェクトのデフォルトパスとして使用

Fig 3.2-2 プロジェクト名とパス

プロジェクト名とパスを設定します。例えば、

・名前:opencv01

・パス : /home/guest/projects/

と入力します。この場合、/home/guest/projects/opencv01/フォルダが作成されて、フォルダ内にプロジェクトファイル 群が入ります。パスには、存在するフォルダを指定してください。

ターゲッ	ト設定
------	-----

パス	ターゲット設定	
ターゲット 詳細 サマリ	Qt Creator はプロジェクト opencv01 を以下のターゲットに設定できます: Image: State St	
	ビルド構成の作成: Qt パージョン毎に Debug と Release を1つずつ	
	<戻る(B) 次へ(N) > キャン	12

Fig 3.2-3 ターゲット設定

既定値のまま[次へ]進みます。

クラス情報

🥘 QE GUI アプリ	「ケーション	
パス	クラス情報	
ターゲット	生成するソースコードフ	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	0 0.00 C	
サマリ	クラス名(<u>C</u>):	Widget
	基底クラス(B):	QWidget ‡
	ヘ <mark>ッダーフ</mark> ァイル(<u>H</u>):	widget.h
	ソースファイル(<u>s</u>):	widget.cpp
	フォームを生成する(<u>G</u>):	S
	フォームファイル(<u>F</u>):	widget.ui
		<戻る(B) 次へ(N)> キャンセル

Fig 3.2-4 クラス情報

基底クラス=QWidgetに変更します。[次へ]進みます。

プロジェクトの管理

🥝 Qt GUI アプリ	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:	$< \tilde{t}_{d}^{+} \lfloor_{d} >$	4. T	
> - + -	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/projects/opencv0	1:		
	main.cpp widget.cpp			
	widget.h widget.ui opencv01.pro			
	[< 戻る(<u>B</u>)	完了(F)	キャンセル

Fig 3.2-5 プロジェクト管理

バージョン管理システムを設定できますが、今回は <なし>のままとします。 作成されるフォルダとファイル名の情報が表示されていますので、確認します。 [完了]ボタンを押します。

プロジェクトファイルとトップウィジェット(=ベースウインドウ)となる Widget クラスが作成され、opencv01 プロジェ クトを開いた画面に切り替わります。

3.3 プロジェクトの設定変更

Qt プロジェクトファイルに、OpenCV ライブラリのリンク設定を行います。

① プロジェクトファイル opencv01 / opencv01.pro をダブルクリックして開きます。



Fig 3.4-1 プロジェクトファイル

② OpenCV ライブラリリンクの定義を追加します。

```
LIBS += -lopencv_core
LIBS += -lopencv_imgproc
LIBS += -lopencv_highgui
```

以上の3つのライブラリを使用します。



3.4 画像を表示する

OpenCV 機能を利用した画像ファイルの読込/表示の手順を説明します。

ラベル定義 (ui ファイル編集)

描画先のラベルを追加定義します。以下の手順で行います。

① opencv01 / フォームを展開し、widget.ui をダブルクリックします。



Fig 3.4-1 widget.ui ファイル

② 編集モードからデザインモードに移行し、widget.uiのデザイン画面が表示されます。

	widget.ui*	÷ 🖬 🏭	🛤 🔤 III		- III 🛤 🖪				×
Ot	フィルタ]•		•			オブジェクト	クラス	ĥ
ようこそ	Layouts						👼 Widget 📗	QWidget	
	📑 Vertical Layout	1122112							
	III Horizontal Layout								
副集	Grid Layout								
	Form Layout	110001000100							
デザイン	Spacers	, 2000 2000 B							
	Horizontal Spacer								
デバッグ	Vertical Spacer	a have have h							
	▼ Buttons	11000100000							
	Push Button								2
プロジェクト	Tool Button						フィルタ	+ /	6
	Radio Button	1122312231					Widget : QWidge	et	
儲析	🗹 Check Box						プロパティ	値	E
2	CommanButton		2.4				QObject		
ヘルプ	Button Box						objectName	Widget	
opencv01	Item Viel-Based)		0 8.		フィルタ		QWidget		
	List View	名前(申 	テキスト	ショートカット	チェック可	enabled		
	Tree View	- H02 (2/01	2 1 2 11	2 4 1 7 2 7 1	1-11-1	▶ geometry	[(0, 0), 400	4
PATHlease	Table View						sizePolicy	[Preferred	4
							▶ minimumSize	0 × 0	
							▶ maximumSize	16777215	>
	List Widget	<u>ec</u>		.00		X	▶ sizeIncremen	t 0×0	-
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Stor Trace Luideach	アクションエディ	ターシグナル/	スロットエディタ			▶ DaseSize	0×0	30
1	■ ₽・キーを入力して下	さ 1問	图 2 検索結	果 3 アプリケ	-ション出力 4	コンパイル出す	]		
									_

Fig 3.4-2 デザインモード

③ 左側のウィジェットボックス内の 'Label' をドラッグ&ドロップでウインドウに配置します。
 表示文字の"TextLabel"は不要ですので削除します。

	widget.ui*	÷ 🖬 🗳 🖬	• 🌆 III	E H I	II III 🖽 🖪			ní ×
lot	フィルタ						オブジェクト	クラス
tar#	🔋 Vertical Scroll Bar 🖹						🔻 🐻 Widget	2 QWidget
	Horizontal Slider						label	🛇 QLabel
	🕈 Vertical Slider							
##	Display Widgets		•					
	📎 Label		• •					
デザイン	AI Text Browser							
	Sraphics View							
デバッグ	12 Calendar							
5	LCD Number							
プロジェクト	Progress Bar							e e e e e e e e e e e e e e e e e e e
							フィルタ	+ /-
	Wertical Line						label : QLabel	
解析							プロパティ	値
2							QObject	2
ヘルプ	Qwebview						objectName	label
opencv01	Arthur Ws [Delilo]						QWidget	
			) 💠	N	フィルタ		enabled	
E laure	An A	使用	用中	テキスト	ショートカット	チェック可	geometry	[(90, 90), 66
PAINease	GradientEditor						sizePolicy	[Preferred,
	GradiendererEx						minimumSize	e 0×0
×	Ot PathStrdererEx						maximumSiz	e 16777215 >
	Ot Composnderer					<u>) ))</u>	▶ sizeIncremer	
X	TDisplay xamples] レアクロ	ノヨンエディタ	シグナル	/スロットエディ	9		DaseSize	U×U ()
	P・キーを入力して下さ	1 問題	2 検索編	吉果 3 アプリ	ケーション出力 4	コンパイル出力		

Fig 3.4-3 ラベル配置

④ ベースウィジェットの"Widget"を選択します。

上部のアイコンリストの中の「水平に並べる」アイコンをクリックしてレイアウト設定します。

							× 🕶 🖌							
	widg	get.ui*		÷ 5	SL 1	s 🔜	III =	нπ		E 🖪				mí X
Ot	フィル	レタ	]•										フシェクト クラ	<b>7</b> , <b>1</b>
ようこそ	ABI)	Line Edit	î.								•	•	🗏 Widget 🜌	QWidget
	AI	Text Edit												
	AI	Plain Text Edit												
01	1	Spin Box												
サイン	1.2	Double Spin Box												1
	0	Time Edit												
Server .		Date Edit												
~ ~	1	Date/Time Edit												
	۵ 🥥	Dial	~											) =
コジェクト		Horizontroll Bar										5	イルタ	+ /-
	<b>)</b>	Vertical Scroll Bar										W	idget : QWidget	
鮮析	- I	Horizontal Slider										7	ロパティ	值
2	۴V	Vertical Slider										⊳	minimumSize	0×0
ノレプ	<b>V</b>	Display Widgets	-									•	maximumSize	16777215
encv01	01	Label		a -	1 6	9 4				フィルタ			sizeIncrement	0 × 0
- F	AI	Text Browser	名	前	使	用中	5	テキスト	ショ	ートカット	チェック可	•	baseSize	0 × 0
Hlease	10	Graphics View											palette	親から継承
	12	Calendar										P	CUISOF	A [Ubuntu
	45 1	LCD Number											mouseTracking	
		Progress Bar	01	6			2.00	0			7		focusPolicy	NoFocus
17-18		Horizontal Line	7	7クションご	エディタ	シグ	ナル/スロ	コットエディ	9				contextMenu	DefaultCor
And a second														

Fig 3.4-4 レイアウト設定



Fig 3.4-5 「水平に並べる」アイコン

Label を選択して、レイアウトの変更を確認します。
 下図のようなサイズに変更されています。

widget.ui*	÷ 5	🕰 📑 🌉 I	∥ ≡ H I	H H 😽 🖬		and the second	aí X
フィルタ Layouts C Vertical Layout						オブジェクト ク グ III Widget 図 label  、	QWidget QLabel
Grid Layout							11-
Horizontal Spacer Vertical Spacer Buttons						N	
Push Button     Tool Button     Radio Button						レオ フィルタ label:QLabel	+ /-
Check Box CommanButton					-	ブロパティ QObject objectName	值 label
Item Viel-Based) List View To Tree View	名前	■ [◎] �- 使用中	テキスト	フィルタ ショートカット	チェック可	QWidget     enabled     peometry     sizePolicy	✓ [(9, 9), 382 [Preferred,
Table View Column View						<ul> <li>minimumSize</li> <li>maximumSize</li> </ul>	0×0 16777215>
	widget.ui*         7 11.49         ▼ Layouts         ↓ Vertical Layout         ↓ Horizontal Layout         ☆ Grid Layout         ☆ Form Layout         ♥ Form Layout         ♥ Pron Layout         ♥ Buttons         ♥ Dush Button         ֎ Check Box         ② CommanButton         ♥ Button Box         ♥ List View         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **         **	widget.ui*	widget.ui* ・ Layouts ・ Vertical Layout ・ Vertical Layout ・ Vertical Layout ・ Form Layout ・ Spacers M Horizontal Spacer ・ Buttons ・ Push Button ② Tool Button ③ Radio Button ③ Check Box ② CommanButton ※ Button Box ・ List View ・ 法 New ・ 法 New * * * * * * * * * * * * * * * * * * *	widget.ui* → ■ 単 ■ ■ ■ ■ H = H = 27ルタ ✓ Layouts ✓ Vertical Layout Ⅲ Horizontal Layout ☑ Form Layout ※ Form Layout ※ Form Layout ※ Push Button @ Push Button @ Radio Button @ Radio Button @ Check Box @ CommanButon ※ Button Box Item VieI-Based) ■ List View ※ Tree View 圖 Table View	widget.ui*     ● 唱 唱 唱 唱 調 川 三 村 王 詳 詳 型       7/ルタ     ·       Vertical Layout     ·       Wertical Layout     ·       Wertical Layout     ·       Grid Layout     ·       Wertical Layout     ·       Wertical Layout     ·       Wertical Spacer     ·       Push Button     ·       Push Button     ·       O Radio Button     ·       Check Box     ·       CommanButon     ·       List View     ·       *B Tree View     Table View	widget.ui*     ● 唱 唱 唱 唱 調 調 目 日 工 詳 詳 型       27ルタ       Y Layouts       Y Vertical Layout       Horizontal Layout       Grid Layout       Wertical Layout       Buttons       Push Button       Posh Button       Omage: CommanButton       List View       List View       List View       Tote View	widget.ui*     ・ 私 私 風 川 三 村 工 前 前 認 型       27ルタ       Y Layouts       Wridget Iayout       Wridget Iayout       Wridget Iayout       Wridget Iayout       Wridget Iayout       Form Layout       Spacers       Widget Iabel       Push Button       Posh Button       Check Box       CommanButton       List View       List View       Atim 使用中 テキスト ショートカット チェック可       Pable View       Table View

Fig 3.4-6 レイアウト確認

#### コード編集

画像ファイル読込/表示の処理を定義します。widget.cpp ファイルを以下のように変更します。

```
#include "widget.h"
#include "ui_widget.h"
#include <opencv2/opencv.hpp>
Widget::Widget(QWidget *parent) :
   QWidget(parent),
   ui(new Ui::Widget)
{
   ui->setupUi(this);
   const char* filename="/usr/local/share/OpenCV/samples/c/fruits.jpg";
   cv::Mat img = cv::imread(filename, 1);
   cv::cvtColor(img, img, CV_BGR2RGB);
   QImage qimg(img.data, img.cols, img.rows,
                img.step, QImage::Format_RGB888);
   ui->label->setPixmap(QPixmap::fromImage(qimg));
}
Widget::~Widget()
{
   delete ui;
}
```

#### 実行

 Qt Creator 左下の実行ボタン(緑の三角アイコン)を押して実行スタートします。 初回の実行では、ビルドしてから実行されます。



#### Fig 3.4-7 「実行」ボタン

② PC(Ubuntu)上でプログラムが起動します。

下図のように、"Widget"ウインドウのラベルに fruits.jpg 画像が表示されます。



Fig 3.4-8 画像表示結果

uiファイルのラベルサイズ/ウィンドウサイズは小さいですが、画像サイズに合わせて大きく表示されます。

#### 解説

- · 画像ファイル fruits.jpg は OpenCV サンプルフォルダにあるファイルです。
- · imread()は 画像ファイルを読み込んで、cv::Mat 型の画像データにしています。
- cvtColor()は画像データの変換を行うメソッドです。CV_RGB2BGRを指定してデータの並びを BGR から RGB の順 にスワップしています。これは、OpenCV が BGR の並びになっているのに対し、表示に使う Qt は RGB であるため です。
- · QImage は Qt の画像データクラスです。コンストラクタで cv::Mat のデータを渡して作成しています。

## 3.5 画像処理を行う

次は、読込画像に対して、ガウシャンフィルタを用いたぼかし(平滑化)と Canny 法のエッジ検出を行います。 本節では、前節の画像表示処理からの変更部分のみを記載します。

コード編集

widget.cpp のコンストラクタ部分を次のように変更します。

#### 実行



Fig 3.5-1 エッジ検出結果

#### 解説

GaussianBlur() でぼかし処理をしています。 Canny()で Canny 法のエッジ検出を行っています。 Canny()は 8 ビットシングルチャンネルの入力画像である必要があるため、cvtColor()でグレースケール変換しています。

## 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

- ・VirtualBox は、Oracle Corporation の登録商標、商標または商品名称です。
- ・Linux は、Linus Torvaldsの米国およびそのたの国における登録商標または商標です。
- ・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。
- ・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