GUI 응용 프로그램 (Qt/Embedded & Qtopia)

임성운

Embedded Linux GUI Toolkit

- Microwindows
 - 소형 임베디드 기기에 탑재할 목적으로 시작된 프로젝트로, 현대적인 그래픽 윈도우 환경 제공
- OpenGUI
 - 오픈 프로젝트로 진행되고 있으며, x86 어셈블리 커널에 GUI를 내장한 형태 갖춤
 - 2D 형태의 화면을 제공하고 있으며, 이벤트 구동 방식의 윈도우 프로그래밍 환경 및 쉬운 어플리케이션 개발 환경을 가짐
- Tiny-X
 - X-window 시스템을 임베디드 시스템에 맞게 소형화시킨 버전
- Qt/Embedded



Qt/Embedded

- Qt란 무엇인가
 - 1996년 초에 노르웨이의 Trolltech에서 발표한 크로스 플랫폼(Cross-platform)을 지원 하는 C++ GUI 프로그램 개발용 라이브러리 또는 Toolkit
 - 어플리케이션 개발자는 Qt의 Widget들을 이용하여 GUI(Graphic User Interface) 환경 을 구축하는데 필요한 기능을 모두 얻을 수 있다.
 - <mark>객체지향</mark>으로 확장이 용이
 - 푸쉬 버튼, 메뉴, 다이얼로그와 같은 그래픽 객체와 함수들의 모임
 - 그래픽 객체들은 Widget이라 하고, Widget에 대응하는 함수들을 합한 것을 Toolkit이라 한다.
- Qt/Embedded 개요
 - 노르웨이의 Trolltech사에서 개발한 리눅스 기반의 Qt 라이브러리로서 C++로 제작 된 Embedded system용 GUI Toolkit
 - 편리한 포팅과 다양한 GUI 기능을 제공
 - 최근 Qt/E 버전은 4.1.2 까지 제공되고 있으며, 적용 버전은 2.3.2 버전 사용
 - X-윈도우 없이 리눅스 커널에서 제공하는 프레임 버퍼를 이용하여 그래픽 장치에 접근 하는 방식 사용



Qt/Embedded의 특징

- 적은 메모리 용량을 요구한다.
 - 1M ~ 5M 사이의 ROM에 구성이 가능
 - X11에 있는 동일한 글꼴보다 작은 글꼴 사이즈
- 여러 플랫폼간 소스 코드 호환
 - ARM, i386, Motorola, MIPS, PowerPC ...
- I386용으로 작성한 프로그램도 Qt/Embedded용으로 쉽게 변경 가능(소스코드 호환성 가짐)
- 풍부한 API 및 안정성
- 트루타입(TrueType)과 BDF 글꼴을 지원한다.
- Qt Designer등의 유용한 Tool 지원
- C++ 기반의 객체지향 클래스 라이브러리이다.
- 비디오 프레임 버퍼에 직접 접근이 가능하다.
- 다국언어 및 유니코드와 국제 언어가 지원된다.
- 표준 GUI를 구성할 만큼의 위젯을 제공한다.
- 소스가 공개된다.



Qtopia

- Qt/Embedded 기반으로 만들어진 Window Manager(WM)로서 콘솔 상에서 실행 하는 Qt/E 어플리케이션을 MS-windows처럼 윈도우 상의 아이콘을 클릭하여 실행 할 수 있는 환경 제공
- WM 기능뿐만 아니라 Qt/E 기반으로 된 여러 가지 어플리케이션을 포함하고 있으므 로 Qtopia를 타겟 보드에 포팅하여 PDA를 위한 WM 기능 구현
- Qtopia에 제공되는 어플리케이션
 - Address Book, Calculator, Calendar, City Time, Clock, File Manager,
 Media Player, Spreadsheet, System Info, Terminal, Text Editor, Todo
 List



Qt/E Source: Trolltech



OS : Redhat Linux 9.0

Source

- qt-x11-2.3.2.tar.gz
- qt-embedded-2.3.2.tar.gz
- qpe-1.4.0.tar.gz
- tmake-1.8.tar.gz
- http://www.trolltech.com



Qt/E 라이브러리 구성

• QObject 클래스는 대부분의 클래스 상속 관계





시그널(Signal) & 슬롯(Slot)

- 모든 QT 클래스(QObject로부터 상속받 은 클래스)는 시그널과 슬롯을 가질 수 있다.
- Signal 어떤 일이 발생했다는 것을 알린 다.(예, 마우스 클릭 혹은 키보드 누를 경우 발생)
- Slot 어떤 시그널에 응답을 하는 함수
- 특정 위젯이 자신의 상태가 변경되면 시그널
 을 발생하고 특정 슬롯에서 시그널을 받아 처리한다.
- connect 함수를 사용하여 작성한다.
- connect(시그널송신자, SIGNAL(시그 널), 시그널수신자, SLOT(슬롯));
- ex) connect(bt, SIGNAL(clicked()), aApp, SLOT(quit()));





Qt/E 유틸리티

- uic
 - Qt Designer에서 만들어낸 .ui 파일에서 .cpp와 .h 소스를 추출해낼 때 사용하는 유 틸리티
- progen
 - tmake를 이용하여 Makefile을 만들 때 사용하는 프로젝트 파일(.pro)을 자동 생성하 는 유틸리티
 - \$ progen hello.cpp hello.h -o hello.pro
- tmake
 - tmake는 .pro 파일을 가지고 make에서 사용 가능한 Makefile을 만드는 유틸리티
 - \$ tmake hello.pro -o Makefile



- Moc(Meta Object Compiler)
 - Qt에서 사용하는 Signal과 Slot을 포함한 소스를 C++ 컴파일러에서 컴파일 할 수 있도 록 일종의 Meta Object 코드를 생성한다.
- Qt Designer(사용)
 - GUI를 구성하기 위해서 위젯을 모눈종이에 그려 넣고 눈금의 위치를 이용하여 위젯의 위 치와 크기를 계산했던 단순 작업을 좀 더 쉽게 해주는 툴



Qt Designer 실행화면





Qt/E 개발환경

개발용 리눅스 PC

타겟 보드





Qtopia 컴파일

- Qt-X11
 - qt-x11-2.3.2.tar.gz
- Qt/E 라이브러리
 - qt-embedded-2.3.7.tar.gz
- Qtopia
 - qpe-1.4.0.tar.gz
- Tmake- Trolltech에서 개발한 소프트웨어 프로젝트 툴
 - -tmake-1.8.tar.gz





Pre-compiled Qtopia 설치

- Pre-compiled Qtopia 사용 방법
 - pre-compiled Qtopia를 압축하여 제공하므로 압축 파일을 해제하고, PATH 설정만으로 사용할 수 있다.

```
      source 명령은 bash의 내부 명
      # c

      령어이며, 명령어가 따로 존재하는
      # c

      것이 아니라, bash에 포함되어 있
      # c

      는 명령어이기 때문에 bash 쉘이
      # c

      작동 중일 때만 동작. 만약 다른 쉘
      # .

      을 사용하면 사용 불가능.
      # 1
```

# mount /dev/cdrom		
# cd /mnt/cdrom/Lecture		
# cp qt.tar.gz /root		
# cp qtinstall.sh /root		
# cd /root		
# ./qtinstall.sh		
# 1s		
Qt-application	Qtinstall	•••
# cd Qtinstall		
# source qtearm_env		



Qtopia 프로그램 작성

- 1. Qtopia Program
- 2. Qtopia WM 에서 실행하기
- 3. Qt-Designer를 이용한 Qtopia Program



Qtopia Program(1)

- main.cpphello.pro
- Makefile





abc 🥒 🔺

٩

0:02

Qtopia Program(2)

• "Hello World" 출력 프로그램 (main.cpp)

```
#include<qpeapplication.h> //qpe 제작에 필요한 기본 헤더, 기본폰트/커서와 같은 응용프로
  그램 자원 관리 역할.
#include<qwidget.h> //위젯의 생성 및 사용에 관한 헤더, 눈에 보이는 사용자 인터페이스를
  갖는 기반 클래스
class MyWidget : public QWidget // 하나의 위젯을 상속 받음
public:
  MyWidget();
                     // QWidget을 상속받아 MyWidget 클래스를 정의
};
MyWidget::MyWidget() // MyWidget의 생성자
  setCaption( "Hello World" ); // 타이틀 바에 글자를 출력하는 메소드 호출
```



Qtopia Program(2)

• "Hello World" 출력 프로그램 (main.cpp)

```
int main(int argc, char **argv)
{
 QPEApplication app(argc, argv);// QPE 라이브러리를 사용하는 어플리케이션(app) 생성
 MyWidget mw; // 보여줄 위젯의 객체를 설정
 app.setMainWidget(&mw); // 어플리케이션의 기본(메인) 위젯으로 mw를 지정
 mw.resize(150,100); // 화면사이즈를 지정
 mw.show(); // 화면에 보여주는 함수를 호출
 return app.exec(); // 제어를 어플리케이션에게 맡긴다.
```



Qtopia Program(3)

• Makefile 생성방법

progen main.cpp –o hello.pro (hello.pro에 추가 사항)

INCLUDEPATH += /root/Qtinstall/qpearm-1.4.0/library

DEPENDPATH += /root/Qtinstall/qpearm-1.4.0/library

LIBS += -lqpe ----- -l: Library link option

tmake hello.pro –o Makefile # make – 실행파일 생성

직접 실행 시키기 위해서는 타겟 보드에 업로드 후 다음과 같이 실행한다. #ps 명령어 사용하여 현재 실행 중인 프로세서 정보를 확인 #killall qpe 혹은 #kill _PID번호 qpe # ./hello -qws &

Target과 Host System이 독립적으로 동작하기 위한 Option



Qtopia WM에서 실행하기

• Window Manager에서 실행하기 위한 설정 파일 다운로드

	File	Directory
Photoshop과 같은 프로그램을	hello.desktop	\$QPEDIR/apps/Applications/
이용하여 작업 (32X32	hello.png	\$QPEDIR/pics/
pixel)	hello	\$QPEDIR/bin/

그래픽 -> 추가그래픽응용프로그램 -> Icon Editor 프로그램 사용하여도 됨.

• hello.desktop 파일의 내용

[Desktop Entry]	- 타겟 보드의 \$QPEDIR = /jffs/Qtopia	
Comment=Hello Program		
Exec=hello		
Type=Application		
Name=hello		
Icon=hello		



Qt-Designer를 이용한 Qtopia Program(1)

- Qt Designer -> New File/Project -> Widget
- 새로운 위젯을 생성한다.

	🖌 Qt Designer by Trolltech 📃 🗃 🕷
	<u>Eile Edit Project S</u> earch <u>I</u> ools <u>L</u> ayout <u>P</u> review <u>W</u> indow <u>H</u> elp
프로그래밍 -> 추가 프로그래 밍 도구 -> QT Designer	Elle Edit Project Search Iools Layout Preview Window Help
	ButtonGroup ListBox Buttons Containers Views Database nput Display Display Display Display C++ Header File Wiews (KDE) C++ Header File Uiews (KDE) Graphics (KDE) Create a new project, form or source file



Qt-Designer를 이용한 Qtopia Program(2)

• Toolbox -> PushButton, PushButton을 생성한다.

♥Qt Designer by Trollte	ech		- 8 ×
Eile Edit Project Sea	rch <u>I</u> ools <u>L</u> ayout <u>P</u> review <u>W</u> indov	/ Help	
🖹 🕒 🖏 🔚 <no pro<="" td=""><td>oject> 🛃 🜈 💁 🐹 🛱 🗈</td><td>1 B</td><td></td></no>	oject> 🛃 🜈 💁 🐹 🛱 🗈	1 B	
	😤 🐻 🗖 🚺 📷 🚳	₩?	
Toolbox 🔀	Form1	Property Editor/Signal Handlers 🗵	Object Explorer 🗵
Common Widgets		Properties Signal Handlers	Objects Memb < >
Buttons	••••••••••••••••••••••••••••••••••••••	Property Value	Name Class -
🕶 PushButton	pushButton1	name JshButton1	Eorm10Widget
J ToolButton		enabled True	
I TOOIBULLOIT	<u></u>	geometry [40, 30, 121	C. DK P QPusnB
RadioButton		sizePolicy Minimum/Fixed	
CheckBox		minimumSize [0, 0]	
CHCCKDOX		maximumSi [32767, 327]	*
		🗈 paletteFor	
Containers		paletteBa	ProjectOverview 🛛
Views		paletteBa	
Database		palette	🔄 <no project=""> 🗖</no>
Input		backgroun WidgetOrigin	Eorm1: upp
Display		I font Sans-10	E- Former unit
Display (KDE)		cursor Arrow	unnamed
Buttons (KDE)		mouseTra False	
Input (KDE)		focusPolicy TabFocus	
Views (KDE)		acceptDro False	
Container (KDE)		autoMask False	
Graphics (KDE)		text pushButton1	
Custom Widgets	<		
Ready	prost theory		



Qt-Designer를 이용한 Qtopia Program(3)

• Tools -> Connect Signal/Slots, 시그널과 슬롯을 생성한다.

♥ Qt Designer by Trolltech				
Eile Edit Project Search Iools Layout Preview Window Help				
📱 🖻 🖏 🔚 <no project=""> 🛃 🜈 💊 🖾 🛍 🛍 🚳</no>				
Toolbox 🗵 🖬 Form1 🔺 Property Editor/Signal Handlers 🗵 Object Explorer 🗵				
Common Widgets Common Widgets Objects Memb < >				
Buttons Name Class A				
PushButton pushButton1 name Form1 Form1QWidget				
ToolButton				
View and Edit Connections				
Connections:				
Chec Sender Signal - Receiver Slot New				
pushButton1 clicked() pushButton1				
Container Delete Ject Overview X				
Views				
Database (No Project)				
Input Eorm1: upp				
Buttons (
Views (KDE)				
Container (KDE)				
Graphics (KDE)				
Custom Widgets				
Edit connections				



Qt-Designer를 이용한 Qtopia Program(4)

- form (form1.ui) 파일 생성
- 디렉토리를 생성하고 form1.ui 파일을 저장한다.





Qt-Designer를 이용한 Qtopia Program(5)

• uic 프로그램을 이용하여 (form1.h, form1.cpp) 파일 생성

```
# uic form1.ui > form1.h
```

uic -impl form1.h form1.ui > form1.cpp

• 시그널 & 슬롯을 위한 form1.cpp 파일 수정

```
#include <qmessagebox.h>
. . .
Form1::Form1()
   // signals and slots connections
   connect( pushButton1, SIGNAL( clicked() ), this, SLOT( newSlot() ) );
void Form1::newSlot()
   QMessageBox mb(tr("Message Box"), tr("Signal and Slot Test"),
   QMessageBox::Information, QMessageBox::Ok, 0, 0, this);
   mb.exec();
```



Qt-Designer를 이용한 Qtopia Program(6)

- Button 위젯을 호출할 main() 함수 구현 • 새로운 슬롯을 form1.h 파일에 등록 #include <qpeapplication.h> ... #include <qimage.h> **QPushButton* pushButton1;** #include "form1.h" **Protected:** int main(int argc, char **argv) **Protected slots:** virtual void languageChange(); **QPEApplication** app(argc, argv); virtual void newSlot(); Form1 button; **};** app.setMainWidget(&button); button.resize(640, 480); . . . button.show(); return app.exec();
- 자동툴을 이용한 Makefile 작성

progen main.cpp form1.cpp form1.h –o button.pro (button.pro 파일에 Qtopia 관련 옵션 추가) **INCLUDEPATH** += /root/Qtinstall/qpearm-1.4.0/library # tmake button.pro –o Makefile **DEPENDPATH** += /root/Qtinstall/qpearm-1.4.0/library LIBS += -lqpe # make



Qtopia WM에서 실행하기

• Window Manager에서 실행하기 위한 설정 파일 다운로드

File	Directory
button.desktop	\$QPEDIR/apps/Applications/
button.png	\$QPEDIR/pics/
button	\$QPEDIR/bin/

• button.desktop 파일의 내용

```
[Desktop Entry]- 탄켓 보드의 $QPEDIR = /jffs/QtopiaComment=PushButton ProgramExec=buttonType=ApplicationName=buttonIcon=button
```



Qt-Designer를 이용한 Qtopia Program(7)

• PushButton Program 실행 예



Message Box	8 (K)	
Siggnal a	nd Slot test!	
4		
ОК		



QWidget

• 인터페이스를 구성하는 눈에 보이는 위젯들은 QWidget의 상속을 받는데 QWidget 역시 QObject을 상속을 받는다. QWidget의 기본적인 메소드는 다음과 같다.

QV	QWidget			
Public Members				
	void setStyle(const QString & style)	위젯에 사용되는 스타일을 정 의한다.	style()	
	void setFont(const QFont &)	위젯에 사용되는 폰트를 정의 한다.	font()	
	void resize(int w, int h)	위젯의 크기를 재조정한다.		
	void setBaseSize(int basew, int baseh)	위젯의 기본 크기를 정의한다.	baseSize()	
	void setFixedSize(int w, int h)	위젯의 고정된 크기를 정의한다.		
	virtual void setMininumSize(int minw, int minh)	위젯의 최소크기를 정의한다. 사용자가 위젯의 크기를 조정할때 이 이하의 값으로 조정할 수 없다.	mininumSize()	
	virtual void setMaxinumSize(int maxw, int maxh)	위젯의 최대크기를 정의한다. 사용자가 위젯의 크기를 조정할때 이 이상의 값으로 저정할 수 없다.	maxinumSize()	



QPushButton

- PushButton이라는 위젯은 일반적인 윈도우 프로그래밍에서 많이 사용되는 것으 로 사용자가 마우스를 클릭하면 어떠한 반응을 하도록 하기 위한 위젯이다.
- QPushButton은 기본적으로 QButton을 상속받는다. QButton은 기본적인 버튼의 표현을 위한 추상화된 클래스이다. QButton은 토글버튼(Toggle Button)과 푸시버튼(Push Button)을 모두 제공하며 버튼에 표시되는 정보는 문자나 이미지 모두 가능하다.



QPushButton

QF	PushButton		
Pub	olic Members		
	void setToggleButton(bool b)	버튼의 형태를 결정한다. 참이면 토 글, 거짓이면 푸시버튼이다.	
	void setFlat(bool)	버튼이 모양이 평평한지를 결정한다.	void isFlat()
	virtual void setPixmap(const QPixmap &)	버튼에 표시되는 이미지를 정의한 다.	
	virtual void setText(const QString &)	버튼에 표시되는 글자를 정의한다.	QString text()
Sig	nals		
	void pressed()	마우스로 버튼을 누르는 상태	
	void released()	마우스로 버튼을 누르는 상태를 해제	
		할 때	QButton으로
	void clicked()	마우스로 버튼을 클릭할때	부터 상속받는
	void toggled(bool on)	마우스로 버튼을 한 번 누르면 들어가	다.
		고 다시 한 번 누르면 버튼이 나온다.	
	void stateChanged(int state)	버튼의 상태가 변화할 때	



QApplication

- QApplication은 QT 어플리케이션의 세팅과 프로그램의 흐름을 관리한다. QT 애플리케이션에 공통적으로 필요한 것을 자동으로 알아서 처리해, 애플리케이션 개 발자는 QApplication에 대해서 자세한 지식이 없어도 된다. 애플리케이션마다 공통적으로 필요로 하는 부분들은 이 클래스에 포함되어 있다.
- QApplication 객체는 qApp라는 전역변수(Global Variable)를 통해서 접 근할 수 있다.

QApplication			
Puk	Public Members		
	int exec ()	프로그램이 종료되거나 메인 위젯이 닫힐때까지 루프를 돌면서 이벤트를 처리한다.	
	virtual void setMainWidget (QWidget * mainWidget)	메인 위젯을 정의한다.	
Signals			
	void lastWindowClosed ()	제일 마지막 윈도우가 닫힌다.	
Slots			
	void quit ()	프로그램을 종료한다.	void exit ()
	void closeAllWindows ()	프로그램의 모든 윈도우를 닫는다.	



위젯 종류 (1)

	위 젯	설 명
OK	PushButton	사용자가 '눌러서' 어떤 동작(Action 또는 command)이 발생하게 하는 위젯을 말한다. PushButton에 문자나 그림을 이용해서 어떤 역할을 하는지 시각적인 정보를 제공할 수 있다.
	Tool Button	특별한 명령이나 옵션을 빠르게 접근하는 것을 제공하는 버튼으로 보통 QToolBar 내부에서 사용된다.
۲	Radio Button ^[4]	동그란 박스와 설명을 위한 레이블을 가지고 있다. 보통 두개이상을 버튼그룹박스로 그룹을 이룬 후에 사용하는데 여 러개의 선택사항 중 하나만을 선택할 때 사용한다.
⊠	Checkbox	Radio Button과 기본적인 사용은 같지만 동그란 박스가 아닌 네모 난박스를 가지고 있다. 사용자가 네모박스를 클릭해서 체크표시를 하거나 없앨 수 있다. 특 정 값을 표시할 때 TRUE/FALSE의 값을 리턴한다.
	Groupbox	서로 관련된 위젯을 묶는데 사용되며 제목을 붙여서 시각적인 효과 를 제공한다.
	ButtonGroup	보통 Push Button이나 Radio Button을 묶기 위해서 사용된다.
	Frame	제목을 갖지 않고 서로 관련된 위젯을 시각적으로 묶는데 사용된다.



위젯 종류 (2)

	위 젯	설 명
	Tab Dialog	Tab 다이얼로그를 구현할 때 사용한다.복수의 폼을 단일 위젯의 하 나의 폼에 구성할려고 할 때 이용된다.
ī	Listbox	문자열로 구성된 정보를 표시, 각 항목을 마우스를 이용하여 선택이 가능하다.
	List View	Listbox가 문자열로 구성된 항목만을 나열하는데에 비해서 ListView 는 여러 컬럼으로 구성된 세부적인 정보와 다양한 기능을 제공한다.
•	Icon View	아이콘과 같은 Pixmap을 나열하기 위해서 사용된다.
	Table View	항목을 스프레드쉬트에서 사용하는 테이블 형식으로 나열한다.
abi	Line Edit	사용자에게 어떤 문자를 입력하거나 수정하게 할 때 사용된다.
•	Spin Box	Label 위젯 또는 다른 위젯에 표시되는 숫자값을 증가시키거나 감소 시키는데 사용된다.
cde abĭ	MultiLine Edit	여러 줄로 입력되는 값을 받거나 수정하고자 할 때 사용된다.
1	Combo Box ^[5]	Dropdown List Box라고도 불리우는데 에디터를 할 수 있는 영역과 리스트 영역(여러개의 값중 하나를 선택할 수 있는 영역)으로 구성 되어 있다. 사용자에게 어떤 값을 선택하게 하고 싶을 때 사용된다.



위젯 종류 (3)

	위 젯	설 명
	Slider	슬라이더와 틱(Tick)마크를 포함하며, 사용자가 마우스를 이용하여 슬라이더를 끌거나 좌우측으로 이동할 수 있다.
\odot	Dial	스핀박스와 기본적인 기능은 같지만 보다 가식적인 형태로 입력값 과 출력값을 표시한다.
A	Text Label	사용자에게 정보를 제공하는 단순한 텍스트를 나타내기 위해서 사 용된다. 단지 읽기를 위한 용도이지만 프로그램 내부적으로 나타나 는 문자를 변경하게 할 수 있다.
	Pixmap Label	사용자에게 이미지로 된 정보를 나타내는데 사용된다. 기본적인 기 능은 Text Label과 같다.
45	LCD Number	전자시계에서 사용하는 LCD Display같은 형태로 숫자를 표시한다.
	Line	화면에 선을 나타내기 위해 사용된다. 항목간의 구분이나 시각적인 효과를 위해서 사용한다.
	Progress bar	박스 안의 Bar의 크기를 조절하여 시각적인 양의 % 를 나타낸다. 보 통 작업의 진행사항을 표시할 때 사용한다.
text	TextView	한 줄의 텍스트가 아닌 여러 줄의 텍스트를 표시하기 위해서 사용된 다.
text	Text Browser	TextView에 간단한 탐색기능을 추가한 문자 표시기이다.

