



Bluewind s.r.l.
P.I./C.F. 03943330260
I-31033 Castelfranco Veneto (TV)
Borgo Pieve 55/C int. C
Tel: +39 0423 723431
Fax: +39 0423 744738
Mail: info@bluewind.it

VoIP and MMI X11-based demo for Freescale iMX-27 platform.

User's guide

References:

Bluewind: BW070907-SW-FSIMX27

0. Revision

Revision	Notes	Date	Author
1.00	Draft, preliminary: video streaming not yet included	June, 29 th , 2007	Enrico Benetti
2.00	VPU codec integration, H263 streaming enabled	September, 7 th , 2007	Enrico Benetti

1. Introduction

This document provides information on how to integrate into standard LTIB-based iMX27 BSP further software packages included in this application note.

The target is to demonstrate the flexibility of Opensource solutions for implementing Graphical interfaces and Voice / Video over IP solutions on iMX27.

Basically, same audience, introductory notes and conventions of BSP User manual apply here.

A brief package overview lists included items:

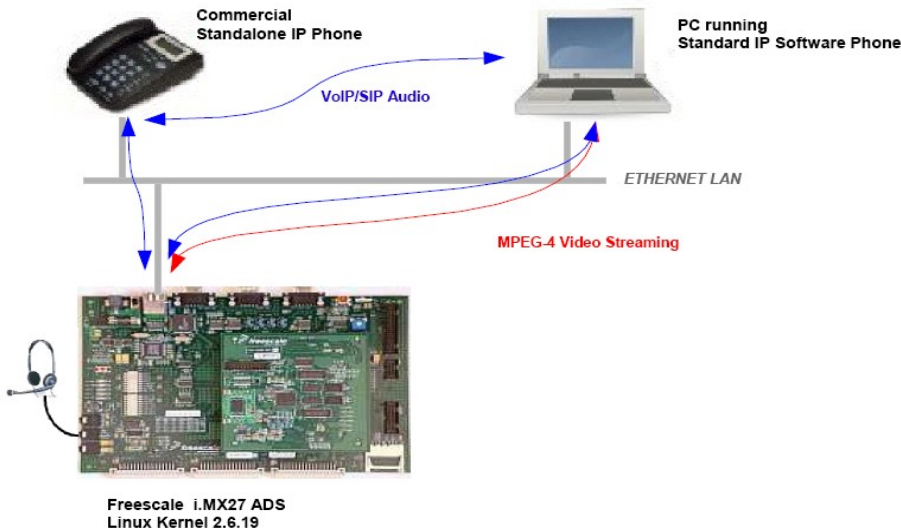
- VoIP application: linphone;
- Window system: Xorg;
- Scripting language interpreter and widget toolkit: Tcl+Tk;
- Demo scripts and auxiliary applications.

Detailed explanations on installation process and provided packages follow on next chapters.

2. Overview

Using Opensource components in VoIP and Video Streaming related applications is a challenge that not all companies want to face with. Unstable results, compatibility issues, time to market needs, deep knowledge about the technologies required when serious problems are found during the development: good reasons for investing on a robust commercial solution for Audio and Video codecs and for SIP / H323 signaling. But when it comes to budget for a new product that will not be developed in big quantities, then Opensource is an option to be taken into account. Several designs carried on in both small and big companies will not serve the huge consumer market of traditional VoIP or Mobile phones; at the same time, the features and compatibility requirements can sometimes be exactly the same for products that will cover niche markets or customized applications.

Bluewind reference demo platform investigates the feasibility of a full Opensource solution for VoIP and Video Streaming over IP. It's based on i.MX27 ADS (development board) from Freescale and a set of ported sources that Bluewind will be glad to distribute free of charge to qualified clients. The result of our tests is a confirmation that the used applications can easily be ported to i.MX27, and the performance obtained with such setup can fit the needs of a wide range of VoIP / Video related products.



Calls between existing commercial VoIP handsets and the Linux-based VoIP software solution running on i.MX27 have been tested, as well as Mpeg4 Video streaming between i.MX27 boards and a standard PC. H263 video streaming has been tested with linphone package between Linux-based PC and ADS board.

3. Installation

Please follow the steps below to install add-on packages:

1. Install Freescale BSP as described in chapter 2 of BSP User Manual.
2. Apply given patch file from LTIB installation directory, assuming `<install_path>` is chosen destination directory on user host's workstation, and `<demo_path>` is the folder where this document and all given stuff is located:

```
patch -p1 -i <demo_path>/ltib-imx27ads-20070522-bw.patch
```

3. Copy all tar.gz and patch files into LTIB packages repository (LPP – Local Package Pool), usually

```
/opt/freescale/pkgs
```

4. Building and running

1. As usual, simply run LTIB with configuration option:

```
cd <install_path>
./ltib --configure --nodeps
```

2. Select Xorg, Tcl/Tk, Tk-demo and Linphone under Package Selection section. Doing this operation, automatic dependencies and conflicts check is performed, as usual with LTIB: be sure to de-select Qtopia, to avoid conflict between different graphic handlers. Automatic selection can be performed selecting `config-bw` with "Load Alternate configuration file".

3. Confirm choices and save configuration: building process is automatically started. Be sure to setup right network parameters.
4. **Please note that checking dependencies of X.org server by rpm could report failure, due to linux-gate.so.1: run ltib with --nodeps flag to avoid this.**
5. Follow BSP user's guide to setup a deployment session on the target (usually with NFS filesystem).

If building and deployment process goes fine, user can see X.org windows server running on iMX27ADS board.

The first time X.org is running it's necessary to run a touchscreen calibration application, called `./calcare-evdev` and located into `/opt/demo/bin`: running X server must be stopped.

5. Filesystem details

In this section we have a brief explanation of root filesystem structure, focused on demo add-on's: everything belongs to `/opt` folder.

`/opt/xorg/` : X11R7.2 Window System.
`/opt/xorg/bin/Xfbdev` : X server.

`/opt/tcltk/` : Tcl+Tk rev.8.3 libraries and interpreters.

`/opt/demo/` : tk demo root folder. In main, setup `PHONE_DEST_1/2` variables according to network parameters.

`/opt/demo/bin` : tk example scripts.

`/opt/demo/bin/calcare-evdev` : touch screen calibration application. This must be run whenever `/etc/calib.data` file is missing or corrupted with `/dev/input/event0` parameter.

`/opt/lin/` : linphone VoIP SIP-compliant client (console version linphonec).
 Tk demo script using linphone is `/opt/demo/bin/demo-phone`.

Folder `/etc/` holds Linphone and Tck/Tk dedicated scripts to setup right environment path and variables: `linphone_env`, `tcltk_env`. Exec these if you want to run applications by hand, outside the provided demo.

To run demo application, execute following steps:

1. Check touch screen calibration: if pointer shows wrong behavior, open a serial console, kill Xfbdev process getting right PID (ps, then kill pid), run
`/opt/demo/bin/calcare-evdev /dev/input/event0`
 and follow screen requests.
2. Check audio settings running alsamixer from console.
 A working configuration examples is:
 Playback: MASTER 96, MASTER_B 100, MASTER_M 67, MASTER_O 87
 Capture: MASTER 40, MASTER_I 29
3. `cd /opt/demo`
4. `./main`

Next chapter gives some more details on X Window System, Tcl/Tk language and Linphone SIP client.

To run manually linphonec from console:

1. Execute configuration script:
`. /etc/linphone_env`
2. Execute linphonec with V option to enable video capabilities:
`linphonec -V`
3. Following help command output, make a call to the desired client with:
`call sip:ip_dest`

6. Xorg and Tcl/tk: advanced graphics system

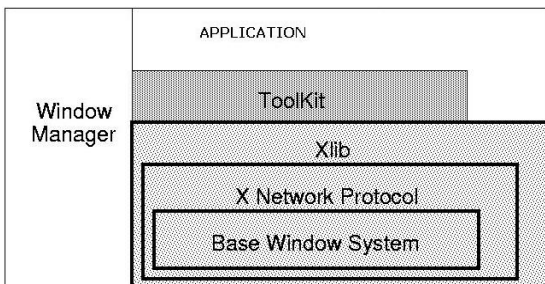
The Tcl/Tk package contains a scripting language Tcl, widely used to build attractive graphics through its graphical extension Tk. Instead of using a specific graphical library (i.e. QT, uWindows..), the developer uses Tcl scripts, which are interpreted run time.

The main advantages of this approach are the noticeable reduction in development time and portability of the script code. Interoperability is also a plus: it is possible to synchronize events of the Tcl execution to other static processes, written for example in C/C++ or Java.

The language extension Tk and its wish (window shell) interpreter is dedicated to the development of graphical applications requiring graphical windows.

In the proposed environment, the direct handling of the display or framebuffer is not a concern. Instead, a graphics server X.org (Open Source Implementation of The X Window System) has been ported.

The port of these elements (X, Tcl/Tk) has been performed on an i.MX21 platform, and used for the i.MX27: this is the advantage of using Linux Standard applications.

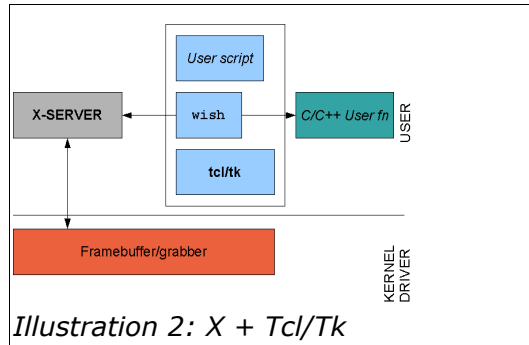
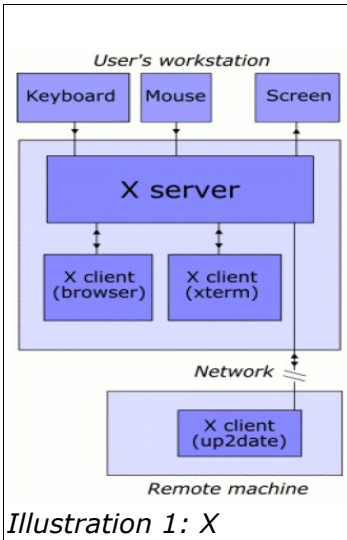


Left picture shows a generic overview of the X and Tcl/Tk Toolkit architecture

X Server

The X server acts as an interface between the applications (clients) and peripherals. In this MMI application example, the wish interpreter is a possible client, exploiting the X graphical libraries to implement the graphical items (i.e windows).

Picture shows system architecture.



Tck/Tk

This section presents some basic examples to give a first glance to this scripting language. User can run wish8.3 and interactively execute commands: on target board, run

```
cd /opt/tcltk/bin
./wish8.3
```

Wish presents its prompt to the user:

```
%
```

Example 1:

Definition of a window with two buttons, one with 'quit' function, the other with user defined procedure.

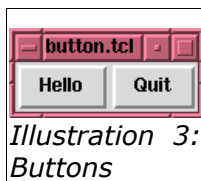
```
#The first button calls a procedure (a callback) that switches the
#buttons name between Hello and Goodbye. The second button
#executes a command that destroys the window.
```

```
set text Hello
proc doIt {widget} {
    global text
    if {$text == "Hello"} {
        set text "Goodbye"
    } else {
        set text "Hello"
    }
    $widget configure -text $text
}
```

```
%button .b1 -text "Hello" \
    -command "doIt .b1"
%button .b2 -text "Quit" \
    -command "destroy ."
```

```
# Put them in the window in row order
%grid .b1 -row 0 -column 0
%grid .b2 -row 0 -column 1
```

Command execution results in button.tcl illustration: basic graphics elements can vary from used libraries.



Example 2

Another useful graphic element is the 'entry', an input text box:

```
# Creates an entry that you can type in.
# focus puts the cursor in the entry, and he button clears it
```

```

label .l -text "Enter:"
entry .e -width 40 -relief sunken -bd 2 -textvariable name
focus .e
button .b -text Clear -command {set name ""}

grid .l -row 0 -column 0 -sticky e
grid .e -row 0 -column 1 -sticky w
grid .b -row 1 -column 0 -columnspan 2
    
```

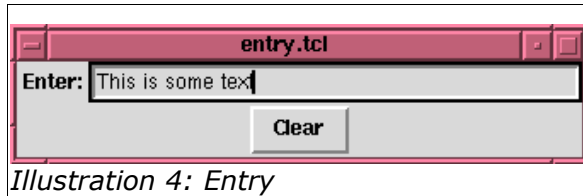


Illustration 4: Entry

7. Linphone

The application chosen to integrate audio/video streaming functionalities is *linphone*, an example of open source implementation based on SIP.

See more at www.linphone.org.

The *linphonec* command line engine allows handling of signalling processes according to SIP standard.

Picture shows the used software structure, and the relationship with the underlying hardware blocks.

Video

Linphone is capable of handle CIF and QCIF MPEG4 and H263+ video streams: acquisition is done interfacing with V4L and V4L2 cameras.

Achievement of input and output video streaming is based on the integration of *Linphone* along with *ffmpeg* codec/application library.

In order to exploit the i.MX27 hardware functionalities, the used codec relies on the H264 hw decoding block implemented in the Multimedia Accelerator of the iMX27. The current Freescale BSP also includes an interface to this hardware block.

At the moment, only H263 VPU codec has been integrated into *ffmpeg libavcodec* : interface to VPU MPEG4 and H264 capabilities will be developed.

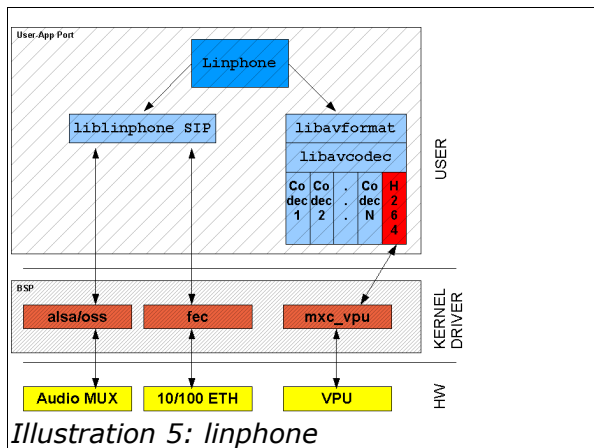


Illustration 5: linphone

8. References

This section reports documentation and packages references:

- LTIB-based BSP release 2 – 20070522
- Linux BSP for the Freescale i.MX27ADS - User's Guide - Rev. 1.6 – 05/2007
- i.MX27 Fact Sheet – Rev. 1
- Linux BSP for the Freescale i.MX27 ADS – Device Drivers - Rev.2 – 05/2007

Development phase was done on a x86 workstation running Ubuntu 6.06:

```
Linux version 2.6.17-10-generic (root@vernadsky) (gcc version 4.1.2
20060928 (prerelease) (Ubuntu 4.1.1-13ubuntu5)) #2 SMP Fri Oct 13
18:45:35 UTC 2006 (Ubuntu 2.6.17-10.33-generic)
```

and tested also over a FEDORA CORE 5 based system.