



Technical Document

GSM PROTOCOL STACK

GPF

xPANEL – MMI TEST APPLICATION (PC)

DEVELOPER DESCRIPTION

Document Number:	06-03-36-SLL-003
Version:	0.3
Status:	Draft
Approval Authority:	
Creation Date:	2000-Oct-26
Last changed:	2015-Mar-08 by Ronny Kiessling
File Name:	Xpan_description.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third-party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, including photocopying and recording, for any purpose without the express written permission of TI.

Table of Contents

0	DOCUMENT CONTROL	4
0.1	CHANGE HISTORY.....	4
0.2	LIST OF FIGURES AND TABLES.....	4
0.3	LIST OF REFERENCES.....	4
0.4	ABBREVIATIONS.....	5
0.5	TERMS.....	5
1	INTRODUCTION:	6
2	SOLUTION.....	7
2.1	GENERAL PROCEDURE.....	7
2.1.1	<i>Environment</i>	7
2.1.2	<i>Build process</i>	8
2.2	DETAILED DESCRIPTIONS - UNDER CONSTRUCTION -	8
2.2.1	<i>Tcl/Tk-Frontend</i>	8
2.2.2	<i>C-Entity-Backend</i>	9
2.2.3	<i>Display update method</i>	9
2.3	TEST	9
3	INTERFACE SPECIFICATIONS	10
3.1	XPANEL – DISPLAY-DRIVER COMMUNICATION - UNDER CONSTRUCTION -	10
3.2	TEST-INTERFACE USAGE - UNDER CONSTRUCTION -	11
4	LAYOUT DEVELOPMENT	12
4.1	BASICS	12
4.2	EDITING THE TEMPLATE.....	12
5	COMMAND EXECUTION	13
5.1	AT-COMMANDS	13
5.2	CONFIG-PRIMITIVES.....	13
6	KNOWN PROBLEMS AND FUTURE TASKS	14
6.1	KNOWN BUGS.....	14
6.2	„SOON IMPLEMENTED“.....	14
6.3	„NICE TO HAVE “	14

0 Document Control

0.1 Change History

Date	Changed by	Approved by	Version	Status	Notes
2000-Oct-26	RK et al.		0.1		1
2003-May-21	XINTEGRA		0.2	Draft	
2003-Aug-18	RK		0.3	Draft	2

Notes:

1. Initial version
2. New Document ID introduced

0.2 List of Figures and Tables

0.3 List of References

[ISO 9000:2000]	International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000
[GSM 2.30]	ETS 300 511: July 1995 (GSM 02.30 version 4.13.0) Man-Machine Interface (MMI) of the Mobile Station (MS), ETSI
[PANEL]	8415.014.00.105, February 7, 2000, Panel – PC Test Application
[XPAN_UG]	06-03-36-UDO, xPanel – MMI Test Application (PC) (xpan_userguide.doc)
[XPAN_MC]	xpan_mainconcept.ppt
[XPAN+]	xPanel_plus_design_spec.doc
[MOAN]	06-03-53-UDO, MoanBtn – Instant GUI-problem Informer (mbtn_userguide.doc)
[PCO2]	06-03-35-UDO, PCO2 – Tracing Environment (pco_userguide.doc)

0.4 Abbreviations

ACI	Application Control Interface (AT Commands)
G23	The Condat implementation of Layers 2 and 3 of the GSM Protocol Stack
G23 Target System	Hardware which executes G23
LCD	Liquid Crystal Display
MM	Mobility Management
MMI	Man Machine Interface
MOC	Mobile Originated Call
MTC	Mobile Terminated Call
PC	Personal Computer
PCO	Point of Control and Observation
PIN	Personal Identification Number
RS232	Serial Communication Standard
Target System	Shortened form of 'G23 Target System'

0.5 Terms

Entity	Program which executes the functions of a layer
Message	A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.
Primitive	A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.
Service Access Point	A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

1 Introduction:

LGIC would like to have their own customer specific PANEL solution. They have provided the keyboard layout and the LCD display size and expect a panel with that design and corresponding functionality: Key-Code connection and graphical display response.

To avoid another proprietary „isolated solution“ only one new panel will be implemented with the functionality of the former „Panel“ and the „Gameboy“ insight. This merged and extended „xPanel“ will have a configurable layout as well. The customer himself should be able to do configuration and creation of new layouts.

At the same time the two existing mechanisms for transmission of graphical display data (MMITAP, Gameboy) will be merged into one solution.

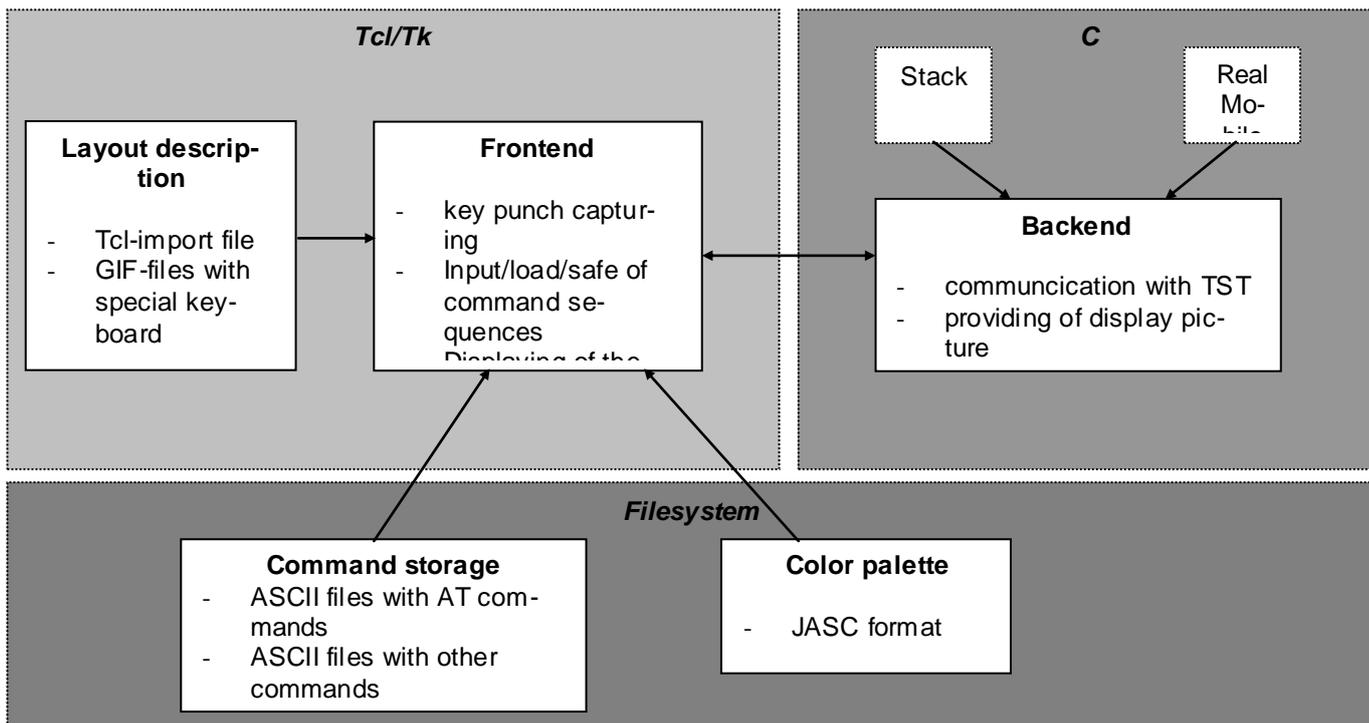
This documentation is dedicated to interested developers. For user/customer specific documentation see [XPAN_UG].

2 Solution

2.1 General Procedure

There are two main parts of the documented software: the TCL/Tk-Frontend and the C-Entity-Backend. While the first will provide the user interface and will handle in- and output the latter communicates with real mobile hardware or a dedicated stack program.

The following picture shows the dependencies between the components:



2.1.1 Environment

To use xPanel some environmental constraints have to be taken into account.

For use under Windows:

At first you'll have to make sure that several DLL-files are available to the system. In the Condat development directory structure you can find them in „<View>/GPF/Bin” and „<View>/GPF/tools/bin”:

- Frame-DLLs (frame.dll, tif.dll, misc.dll)
- cms.dll, ipc.dll
- tk84.dll, tcl84.dll

So just make sure „<View>/GPF/Bin” and “<View>/GPF/gnu/bin” are in your PATH-variable. (use e.g. the “initvars.bat” in „<View>/GPF”)

From the MS-Explorer you can directly double click on the “<View>/GPF/bin/xpan.bat”, which will solve the PATH-problem for you.

Furthermore the Tcl-interpreter expects some files in subdirectories of a lib-folder relative to the start-directory -> “./lib”. At Condat all the necessary files can be found in “<View>/GPF/Lib” :

- “tk8.4”-directory
- “tcl8.4”-directory

At least you need to have xpanel.tcl in the same directory as the executable.

For your special needs you may have to change the default ini-file of xPanel (“<view>/GPF/cfg/xpan.ini”). An explanation of the options is given in **Error! Reference source not found.**

In “<view>/GPF/cfg” you can find some example ini-s like one for the usage with the TI-Multiplexer (“<view>/GPF/cfg/xpan.ini.ti_mux”).

To start meaningful work you additionally need a layout-file –adapted to your needs (see 4 for details). You should use PCO (see [PCO2]) together with xPanel, to watch traces and redirected primitives.

2.1.2 Build process

MS Developer Studio 5.0 must be installed and it's bin-directory has to be in the PATH. (CONDAT internal you should use /GPF/initvars.bat)

ClearCase : make sure all project dependencies are solved by your config spec (see /GPF/assist/readme_xpan.txt)

no ClearCase : make sure that you get all files on which xPanel depends (see /GPF/assist/readme_xpan.txt)

Change to xpanel directory and call 'gnumake' (or 'gnumake help' for more options).

Hint: remove error.log before running make if you want to track current ClearCase errors.

2.2 Detailed Descriptions - under construction -

2.2.1 Tcl/Tk-Frontend

2.2.1.1 Framework

The implementation of the so called frontend “framework” resides in the source file “xpanel.tcl”. The realised functions are:

- General layout construction
- Panel screen update
- Graphical mode: PPM-image file, modified by C-Entity - reread and displayed by this framework (in function “displ_img”)
- Text mode: a multi lined label string is updated in function “displ_text” which will be called by C-Entity

```
set KEY_0      0
set KEY_1      1
set KEY_2      2
set KEY_3      3
set KEY_4      4
set KEY_5      5
set KEY_6      6
set KEY_7      7
set KEY_8      8
set KEY_9      9
set KEY_STAR   10
set KEY_HASH   11
set KEY_VOLUP  12
set KEY_VOLDOWN 13
set KEY_MNUUP  14
set KEY_MNUDOWN 15
set KEY_LEFT   16
```

```
set KEY_RIGHT 17
set KEY_CALL 18
set KEY_HUP 19
set KEY_OK 20
set KEY_CLEAR 21
set KEY_ABC 22
set KEY_POWER 23
set KEY_F1 24
set KEY_F2 25
set KEY_F3 26
set KEY_F4 27
set KEY_MAX 28
```

2.2.1.2 Layout Template

2.2.1.3 Layout For LGIC

The Layout for LGIC is a specially modified version of the layout template -> „LGIC_lo.tcl“

2.2.2 C-Entity-Backend

2.2.3 Display update method

This is how the display updating works:

The Tcl/Tk-frontend periodically calls a procedure *displ_update* which gets the current display information via a C-function call and updates the display window.

For text mode this C-function is *c_get_displtext(xpan_tcl_get_displtext)* which returns the current text to be displayed.

In the graphical mode the C-function is *c_get_displimg(xpan_tcl_get_displimg)* which returns the name of the image file to use.

2.3 Test

- Tcl/Tk-layout generation test
- C-Entity test
- integration test

3 Interface specifications

The next paragraphs contain specific informations about used and provided interfaces.

3.1 XPanel – Display-Driver Communication - under construction -

To redirect a text display output to xpanel a “CONFIG MMI_MODEL=<1,1,1>” primitive is send to PL. Now primitives of the following type reach xpan_pei :

MMI_DISPLAY_REQ ...

Elements:

long name	short name	Type
Attributes	Attrib	STRUCT: - content - control
y-coordinate	c_y	USHORT
x-coordinate	c_x	USHORT
service data unit	Sdu	STRUCT elements: - l_buf - u_buf - buf

Description:

MMI_DISPLAY_REQ requests the output of a given string or a whole bitmap into the display of, e.g. the xPanel. (see 3.1)
 The receiver has to know the type of data (in xpan_pei.c controlled by xpan_displytp):

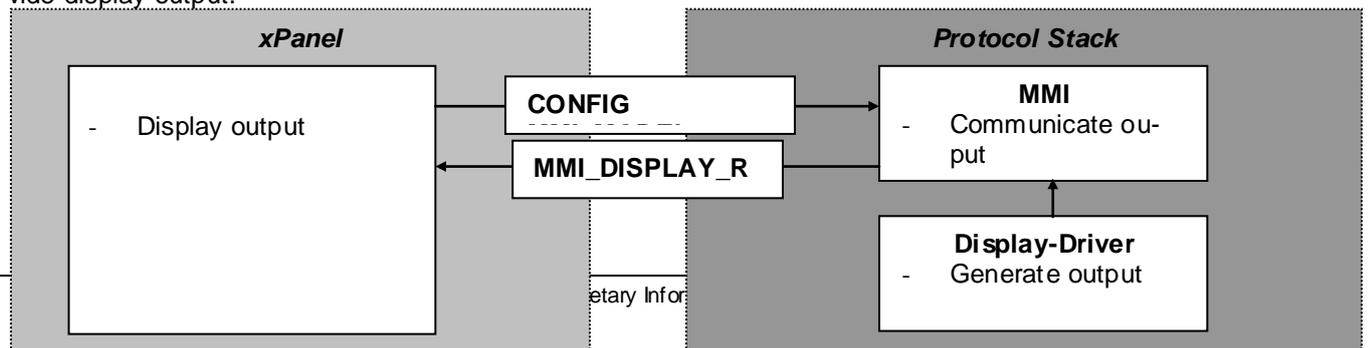
Text-Display: The appearance of the Text is controlled by attributes. The Text is stored in the SDU structure to avoid the exchange of fixed size primitives. The attributes can contain the following values:

Name	value	c-macro	Comment
Content	1	CON_TEXT	Sdu contains text
	2	CON_CURSO R	Positioning the cursor
Control	0	CTL_NORMA L	Normal operation
	1	CTL_HIDDEN	Hidden attribut (used for cursor control)

Name	Comment
C_x. c_y	position of the cursor

Name	value	c-macro	Comment
Content	1	CON_TEXT	Sdu contains text
	2	CON_CURSO R	Positioning the cursor
Control	0	CTL_NORMA L	Normal operation
	1	CTL_HIDDEN	Hidden attribut (used for cursor control)

The following picture gives an overview about the communication process which takes place to provide display output.



To receive graphical display the new external display interface is used. See [XPAN+] for detailed information.

3.2 Test-Interface Usage - under construction -

To communicate with the protocol stack the so called test interface of the Condat-FRAME is used (see [XPAN_MC]).

send_syscmd

4 Layout Development

4.1 Basics

To develop a special mobile layout you have to get detailed informations about all the keys – their location, the text/graphic on them and the associated function.

The next step is to modify the template tcl-file (xpan_lo_tmpl.tcl) and rename it to <name>_lo.tcl where <name> stands e.g. for the name of the company you develop the layout for.

Now this name can just be passed to xPanel as a parameter (e.g. xpan -l <name>) or specified in your ini-file (see **Error! Reference source not found.**) and the new layout will be used. Another option is to load the layout from the menu of xPanel (file/load layout ...).

4.2 Editing the template

After copying the „xpan_lo_tmpl.tcl” to “<name>_lo.tcl” you can modify the constants in the so called part of the layout file:

- Display colors

Here you can specify the fore-and background colors of the text display.

- Key count

Here you can specify the count of keys in a row/column of your mobile.

- Key IDs

Here you can specify the mapping of key names to IDs. Usually no changes are necessary.

- key distribution

Here you can specify how the key are distributed on your mobile. Where no key exists use \$KEY_NOKEY.

- Key labels

Here you can specify strings as labels of your keys. Where no key exists or where you want to provide an image use an empty string.

- Optional Key images

Here you can specify names of image files (Bitmaps) which will be uses instead of a string as labels of your keys. Where no key exists or where you want use a label string specify an empty string.

- Key sizes

Here you can specify the sizes of your keys. Just look at the given examples and “trial and error” until you are satisfied. Where no key exists use “0” (it will be ignored anyway).

- Key positions

Here you can specify the relative positions of your keys using sticky-strings. You can choose from “w” (west), “e” (east), “n” (north) and “s” (south) to indicate that a key should stick at a side, or you can center it by using “”. Where no key exists use an empty string (it will be ignored anyway).

- Key reliefs

Here you can specify the relief type of the keys. One of the following constants may be used: raised, sunken, flat, ridge, solid, groove. Just try and decide what you like.

5 Command execution

It is possible to execute several commands from the xPanel application. Either by directly input them or by loading them from properly formatted ASCII-files.

Description for all supported formats will be provided in the following sections:

5.1 AT-Commands

An AT-command is just a string like "AT+...".

You can input such a line directly by choosing the Menu entry „Cmd/Input AT command ...“.

Under „Cmd/Load AT commands ...“ it is possible to load AT-lines from an ASCII-file formatted like this:

```
<at-string>
<at-string>
<at-string>
:
```

For example:

```
AT+CSCA=491710760000*145
AT+CMGS=491712002400*145*Test__das_ist_MO_SMS
```

5.2 CONFIG-Primitives

A CONFIG-primitive consists of a receiver and a primitive body. You can input both directly by choosing the Menu entry „Command/Input primitive ...“.

Under „Command/Load primitive ...“ it is possible to load primitives from an ASCII-file formatted like this:

```
<receiver>
<prim body>
<receiver>
<prim body>
:
```

For example:

```
MMI
CONFIG KEY_SEQUENCE= LEFT
MMI
CONFIG KEY_SEQUENCE= DOWN
```

6 Known problems and future tasks

This paragraph is meant to show which bugs are already found (but not removed yet) and to provide an impression of future plans concerning this product.

6.1 Known bugs

- XPanel does not remember its position/size correctly sometimes

6.2 „Soon implemented“

- the file dialogs should „remember“ the last chosen path
- a recording of button-presses will be possible
- Online-Help

6.3 „Nice to have “

- more extensive possibilities of command execution – more automatization

Appendices

A. Acronyms

DS-WCDMA Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000) Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: <http://www.imt-2000.org/>>