Author	S. Lesage
Verified	
Version	1.2.6
Date	2013/11/29
Document	File Format

PMIP-K8 Keyboard configuration

Table of contents

1. Introduction 3
2. Main Section 3
2.1 Number of pages 3
3. Page Section 4
4. Key Definition 4
4.1 Selection Groups 5
4.2 Radio Groups – mutually exclusive keys 5
5. Vox@net client Functions 6
5.1.1 Select zone(s) 6
5.1.2 Reset zone selection 6
5.1.3 Microphone paging 7
5.1.4 Chime paging 7
5.1.5 Select a message 8
5.1.6 Play message 8
5.1.7 Stop message 8
5.1.8 Play message in predefined zones 9
5.1.9 Stop message in predefined zones 9
5.1.10 Choose music source 9
5.1.11 Increase / Decrease loudspeaker volume 10
5.1.12 Increase / Decrease matrix audio level 10
5.1.13 Increase / Decrease client audio level 10

6. General purpose functions 11
6.1.1 Command selection 11
6.1.2 Execute command (press-to-talk mode) 11
6.1.3 Execute command (latch mode) 12
6.1.4 Example of command with selection 13
6.1.5 Display variable value 14
6.1.1 Toggle Boolean variable value 14

1. Introduction

The PMIP-K8 Keyboard is made of 9 user-configurable keys.

You can edit your Keyboard directly from Vox@Net (from version 2.4.11).

You can also edit the configuration manually with any Text editor and upload it via the embedded web server, Menu Config -> Keyboard. (for example http://192.168.10.159/keyboard.html)

The file is plain text with a classic INI file syntax, cf. http://en.wikipedia.org/wiki/INI_file.

The section and settings names are all case sensitive.
There must be no spaces between the parameter name, = sign and the value.
The last line must be ended by Line-Return.

2. Main Section

The Main section contains global settings.

You can start the main section with:

```
[Main]
```

2.1 Number of pages

The number of pages is specified with:

```
NbPages=1
```

There is always exactly 1 page. It is then defined with its own section.

3. Page Section

The page section contains 9 key definitions.
You start the page section with:

```
[Page1]
```

The keyboard has 8 keys, numbered from 1 to 8 from Top to Bottom and Left to Right:

Key 1	Key 5
Key 2	Key 6
Key 3	Key 7
Key 4	Key 8

There is also a 9th virtual key for the microphone contact input (remote input 1).

4. Key Definition

Each key has 4 settings: Function, Parameter, Selection group and Radio group defined as shown:

```
FuncX=FUNC
ParamX=parameter
GroupX=integer number
RadioX=integer number
```

Where X is the key number from 1 to 9.

`FuncX` sets the function of key X, you MUST define it. It is always written in capitals.

`ParamX`, **`GroupX`** and **`RadioX`** depend on the function.

Next chapter documents all available functions with:

- function description
- used parameters (mandatory or optional)
- parameter syntax
- example

4.1 Selection Groups

By default, selection keys used by mic/gong/message/command actions are global. So you can span the key selection array on multiple sub-pages and put the actions on your root page.

You can modify this behavior by defining selection groups using the `GroupX` parameter.
A function key defined for group N will use only selection keys of the same group N.
So that you can restrict some zones to some actions, or split on different pages.

The `GroupX` parameter can be any integer number (default if not specified is 0).

Functions using ZONESEL of the same group:
MIC, GONG, MSGPLAY, MSGSTOP, RESET, MUSICSOURCE, VOLMATINC/DEC

Functions using COMMANDSEL of the same group:
COMMAND, LATCHCOMMAND

4.2 Radio Groups – mutually exclusive keys

COMMAND keys are independent by default.

But you can optionally make a group of keys have only one key selected at the same time.
This is called a "Radio button" group (think about ancient car stereos with mechanical keys).

To declare mutually exclusive keys, simply assign them the same radio group number with the `RadioX` parameter.

It is used by the following functions:
COMMANDSEL and LATCHCOMMAND.

5. Vox@net client Functions

The following functions are used with a vox@net server.
The keys will be disabled when not connected.

5.1.1 Select zone(s)

Select a zone/group for next call/message.

`FuncX`	**`ZONESEL`**
`ParamX`	mandatory, syntax is `/MxCy,y,…,y/MxCy` where x is a matrix number and y is a code number of matrix x
`GroupX`	optional
`RadioX`	not used

Example:

```
Func1=ZONESEL
Param1=/M1C1,2/M2C5/M4C1,5,6
Group1=3
```

Select zones 1 and 2 of Matrix 1, zone 5 of Matrix 2 and zones 1,5 and 6 of Matrix 4

5.1.2 Reset zone selection

Deselect all zones (in specified selection group)

`FuncX`	**`RESET`**
`ParamX`	not used
`GroupX`	optional
`RadioX`	not used

Example:

```
Func1=RESET
Group1=3
```

5.1.3 Microphone paging

Start/stop paging with the microphone. You must select the target zones first.

`FuncX`	**`MIC`**
`ParamX`	not used
`GroupX`	optional
`RadioX`	not used

Example:

```
Func1=MIC
Group1=3
```

Call using zones selected in group 3.

5.1.4 Chime paging

Start microphone paging with after a pre-announcement chime. You must select the target zones first. The key is blinking while the chime is playing. Then **`MIC`** key becomes active when you can talk.

`FuncX`	**`GONG`**
`ParamX`	not used
`GroupX`	optional
`RadioX`	not used

Example:

```
Func1=GONG
```

Call using zones of the default selection group.

5.1.5 Select a message

Select a message for play/stop/listen, as well as optional predefined zones.
Only one message can be selected at a time.

`FuncX`	**MSGSEL**
`ParamX`	msgname=zones=chime=prio
`GroupX`	optional
`RadioX`	not used

`msgname` is the message name on the vox@net server (mandatory)
`zones` are the predefined zones with the same syntax as the **ZONESEL** key, or nothing.
`chime` is 0 for no chime, or 1 to use the default chime (defined in the bank of the message)
`prio` is the priority to use (see **PRIO** key) or nothing for default (message bank or client if not defined)

Examples:

`Func1=MSGSEL`	
`Param1=test`	only message name
`Param1=evac=/M1C1,2`	with predefined zones
`Param1=attention==1=75`	with chime and priority
`Param1=alarm=/M1C1/M2C2=0`	with zones and NO chime

5.1.6 Play message

Start playing the selected message. You must select the message and the target zones first.

`FuncX`	**MSGPLAY**
`ParamX`	not used
`GroupX`	not used
`RadioX`	not used

5.1.7 Stop message

Stop playing a message:

- with zone selection: stop all messages in the selected zones
- without zone selection: stop the playback on the integrated loudspeaker

`FuncX`	**MSGSTOP**
`ParamX`	not used
`GroupX`	not used
`RadioX`	not used

5.1.8 Play message in predefined zones

Start playing a predefined message into predefined zones.

FuncX	**MSGZONEPLAY**
ParamX	msgname=zones=chime=priority
GroupX	not used
RadioX	not used

5.1.9 Stop message in predefined zones

Stop playing a predefined message into predefined zones.

FuncX	**MSGZONESTOP**
ParamX	msgname=zones=chime=priority
GroupX	not used
RadioX	not used

5.1.10 Choose music source

Choose music source for currently selected zones.

FuncX	**MUSICSOURCE**
ParamX	client number in vox@net server configuration or 0 (none) to make a “stop music” key
GroupX	optional
RadioX	not used

Example:

```
Func1=MUSICSOURCE
Param1=1

Func2=MUSICSOURCE
Param2=2

Func3=MUSICSOURCE
Param3=0
```

5.1.11 Increase / Decrease loudspeaker volume

`FuncX`	**`VOLINC / VOLDEC`**
`ParamX`	not used
`GroupX`	not used
`RadioX`	not used

Example:

```
Func1=VOLINC
Func2=VOLDEC
```

5.1.12 Increase / Decrease matrix audio level

`FuncX`	**`VOLMATINC / VOLMATDEC`**
`ParamX`	not used
`GroupX`	not used
`RadioX`	not used

The parameter is the predefined matrix number in the vox@net server configuration.
Without parameter, you must select the target zones first.

Example:

```
Func1=VOLMATINC
Func2=VOLMATDEC
```

5.1.13 Increase / Decrease client audio level

`FuncX`	**`VOLCLIINC / VOLCLIDEC`**
`ParamX`	client number in the vox@net server configuration
`GroupX`	not used
`RadioX`	not used

Example:

```
Func1=VOLCLIINC
Param1=1

Func2=VOLCLIDEC
Param2=1
```

6. General purpose functions

6.1.1 Command selection

Select an additionnal command for execution by a **COMMAND** or **LATCHCOMMAND** key of the same selection group. You can define multiple keys to be exclusive using the same `RadioX` parameter.

`FuncX`	**`COMMANDSEL`**
`ParamX`	commandON|commandOFF
`GroupX`	optional
`RadioX`	optional

The parameter must contain the "ON" command and an optional "OFF" command separated by the pipe character. Refer to ATEIS VNB command framework document for a list of available commands.

`commandON` will be executed before the `commandON` of the **COMMAND** or **LATCHCOMMAND.**

`commandOFF` will be executed after the `commandOFF` of the **COMMAND** or **LATCHCOMMAND.**

6.1.2 Execute command (press-to-talk mode)

Execute a command when pressed.
Execute another command when released.

`FuncX`	**`COMMAND`**
`ParamX`	commandON|commandOFF
`GroupX`	optional
`RadioX`	not used

The parameter must contain the "press" command and an optional "release" command separated by the pipe character. Refer to ATEIS VNB command framework document for a list of available commands.

Pressing the key will:

- execute additional `commandON` of selected **COMMANDSEL** keys
- execute `commandON` of the key
- change color according to Success or Failure

Releasing the key will:

- execute `commandOFF` of the key
- execute additional `commandOFF` of selected **COMMANDSEL** keys

Example:

```
Func16=COMMAND
Param16=talkstart|talkstop
```

6.1.3 Execute command (latch mode)

Latch command key.
When pressed, a command is executed according to current selection state.
If execution is successful, the key changes and stays to the new state.

`FuncX`	**`LATCHCOMMAND`**
`ParamX`	commandON|commandOFF
`GroupX`	optional
`RadioX`	optional

The parameter must contain the "ON" and "OFF" commands separated by the pipe character.
Refer to ATEIS VNB command framework document for a list of available commands.

Pressing the key when current state is OFF:

- execute additional `commandON` of selected **COMMANDSEL** keys
- execute `commandON` of the key
- change color according to Success or Failure
- if successful, the color will switch to ON state when the key is released

Pressing the key when current state is ON:

- execute `commandOFF` of the key
- execute additional `commandOFF` of selected **COMMANDSEL** keys
- change color according to Success or Failure
- if successful, the color will switch to OFF state when the key is released

You can define multiple keys to be exclusive using the same `RadioX` parameter.
If another key of the same radio group is ON, then pressing a new key will first execute OFF commands of the previously selected key, before executing ON commands of the new key.

Example:

```
Func2=LATCHCOMMAND
Param2=udpack://192.168.10.220:19761/%02WC0011%03%5C%0D
       |udpack://192.168.10.220:19761/%02WC0010%03%5B%0D
```

We have a UAPg2 at address 192.168.10.220.
We defined COO1 parameter of Third Party Protocol to control input 1 mute.

Starting with in OFF state (key not selected)
Press the key to send the ON command "Write 1 to C001" via UDP with acknowledge on port 19761.
If successful, the key will turn ON when released. So we know UAP Input 1 is muted.

Now the we are is ON state (key selected)
Press the key to send the OFF command "Write 0 to C001" via UDP with acknowledge on port 19761.
If successful, the key will turn OFF when released. So we know UAP Input 1 is not muted.

6.1.4 Example of command with selection

```
Func1=COMMANDSEL
Param1=vnb://192.168.10.151/decode?channel=1&mode=1&addr=225.1.2.3&port=8000
       |vnb://192.168.10.151/decode?channel=1&mode=0

Func2=COMMANDSEL
Param2=vnb://192.168.10.152/decode?channel=1&mode=1&addr=225.1.2.3&port=8000
       |vnb://192.168.10.152/decode?channel=1&mode=0

Func3=COMMAND
Param3=encode?channel=1&mode=1&addr=225.1.2.3&port=8000
       |encode?channel=1&mode=0
```

First select a zone with key 1 and/or key 2.

Pressing key 3 will:

- send decoder start command(s) via VNB protocol to use multicast address 225.1.2.3:8000
- start local encoder of the PMIP-D / Vox@D to stream this multicast address

Releasing key 3 will:

- stop local encoder of the PMIP-D / Vox@D
- send decoder stop command(s) via VNB protocol

6.1.5 Display variable value

Set the light ON or OFF according the a variable value, with automatic refresh.

FuncX	**VARIABLE**
ParamX	varid
GroupX	not used
RadioX	not used

The parameter is the variable identifier or ID (shortname).

Examples:

```
Func1=VARIABLE
Param1=volout1
```

6.1.1 Toggle Boolean variable value

When pressed, execute toggle command on the variable.
Display is the same as the **VARIABLE** function.

FuncX	**VARTOGGLE**
ParamX	varid
GroupX	not used
RadioX	not used

Examples:

```
Func1=VARTOGGLE
Param1=remout1
```