



Author	S. Lesage
Verified	
Version	1.2.5.2
Date	2013/11/04
Document	File Format

PMIP-D and Vox@D GUI configuration

Table of contents

1. Introduction	3
2. Main Section	3
2.1 Number of pages	3
2.2 Global Options	3
2.3 Text Fonts	3
2.4 Language	4
2.5 Colors	5
3. Page Sections.....	6
4. Key Definition.....	6
4.1 Selection Groups.....	7
4.2 Radio Groups – mutually exclusive keys	7

5.	Vox@net client Functions	8
5.1.1	Select zone(s)	8
5.1.2	Reset zone selection.....	8
5.1.3	Select the priority	9
5.1.4	Microphone paging.....	9
5.1.1	Vu-meter.....	10
5.1.2	Chime selection.....	10
5.1.3	Chime paging.....	10
5.1.1	Open program list window	11
5.1.2	Open message list window	11
5.1.3	Select a message	12
5.1.4	Play message.....	12
5.1.5	Stop message	12
5.1.6	Play message in predefined zones	13
5.1.7	Stop message in predefined zones	14
5.1.8	Choose music source.....	15
5.1.9	Listen message	16
5.1.10	Stop listening.....	16
5.1.11	Listen to zone	17
5.1.12	Reset listening	17
5.1.13	Display loudspeaker volume.....	18
5.1.14	Increase / Decrease loudspeaker volume.....	18
5.1.15	Increase / Decrease matrix audio level	18
5.1.16	Increase / Decrease client audio level.....	19
6.	General purpose functions	20
6.1.1	Label.....	20
6.1.2	Jump to another page	20
6.1.3	Command selection	21
6.1.4	Execute command (press-to-talk mode)	21
6.1.5	Execute command (latch mode).....	22
6.1.6	Example of command with selection.....	23
6.1.7	Display variable value.....	24
6.1.1	Toggle Boolean variable value.....	25

1. Introduction

The PMIP-D or Vox@D Graphical User Interface is made of user-configurable pages of 16 keys.

You can edit your GUI directly from Vox@Net (from version 2.4.7).

You can also edit the configuration manually with any Text editor and upload it via the embedded web server, Menu Config -> GUI. (for example <http://192.168.10.159/simplegui.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=3
```

Then each page is defined with its own section.

2.2 Global Options

Options are enabled with 1 and disabled with 0:

```
DisplaySourceName    display the source name inside Zone Selection Keys
Style                graphic style
                    0 for flat style (default)
                    1 for 3D style
```

2.3 Text Fonts

You can change the predefined text size (height in pixels):

```
FontLabel            for all labels (default is 18 on PMIP-D, 24 on Vox@D)
FontSource           for source name (default is 12 on PMIP-D, 18 on Vox@D)
```

2.4 Language

Language FR for French, EN or anything else for English

Regardless of the general language setting, all strings can be customized:

MsgCancel	Cancel
MsgClose	Close
MsgListen	Listen
MsgPlay	Broadcast
MsgStart	Start
MsgStop	Stop
MsgDelete	Delete
MsgListBanks	Banks
MsgListSelectBank	Select Bank
MsgListToManage	Go to Management
MsgListToBroadcast	Go to Broadcast
MsgListNewProgram	New Program
MsgListNewRecord	New Record
MsgListRecordOver	Record Over
MsgListConfirmDelete	Are you sure you want to delete this message ?
ProgListNewProgram	New Program
ProgListActivate	Activate
ProgListDeactivate	Deactivate
ProgListEdit	Edit
ProgListValidate	Validate
ProgListConfirmDelete	Are you sure you want to delete this program ?
ProgActive	Active
ProgMessage	Message
ProgStart	Start
ProgStop	Stop
ProgPeriod	Period
ProgInterval	Interval
ProgTimes	Number of broadcasts
ProgRepeat	Message repetition
MsgRecNewRecord	Record a new message
MsgRecRecordOver	Record over message ?
MsgRecRecording	Recording message...
MsgRecRemaining	%d:%02d remaining
InstantRecBusy	The zones are currently busy. Record your message ?
InstantRecReady	Recording finished OK. Message Ready.
InstantRecListening	Listening message
MsgStateTesting	Testing message broadcast
MsgStatePlaying	Message is now broadcasting
MsgStatePartial	Message broadcasting partially
MsgStateQueued	Message queued
MsgStateDenied	Message broadcast denied
MsgStateListening	Listening message

2.5 Colors

There are a lot of colors definitions, at least one for each kind of key.
All these colors are predefined, but you can customize them to your needs.

A color is encoded with values between 0 and 255 for each Red, Green and Blue component.

Examples:

```
ColorGongON=#ff0000      hexadecimal value start with a sharp
ColorMsg=0 255 255      decimal values separated by any character
ColorBusy=128;128, 128
```

Global color definitions:

```
ColorListenWin    background color of message listening window
ColorRecordWin    background color of message recording window
ColorWarning      background color of warning window

ColorPress        for all keys, while pressed
ColorDisabled     for all keys, when disabled
```

Function specific color definitions are further described for each function.

2.6 Scheduler programs

You can customize the defaults value used when creating a new scheduled program:

```
NewProgStart      program start time (empty/undefined for current time)
NewProgStop       program stop time (default is 23:59:59)
NewProgPeriod     program period (default is 10 minutes)
NewProgDestination matrix/zones destination (empty/undefined to use current selection)
                  syntax is /MxCy,y,...,y/MxCy
                  where x is a matrix number and y is a code number of matrix x
```

This setting is used when editing any program:

```
ProgTimeStep     step size used when editing start/stop times
                  (default is 10 minutes)
```

Examples:

```
NewProgStart=
NewProgStop=21:00:00
NewProgPeriod=00:12:34
ProgTimeStep=00:06:00
NewProgDestination=/M1C1,2/M3C4
```

3. Page Sections

The page sections contain key definitions for each page.
You start a page section with:

```
[PageX]
```

Where X is a decimal number from 1 to NbPages.

Each page can display 16 keys, numbered from 1 to 16 from Left to Right and Top to Bottom:

Key 1	Key 2	Key 3	Key 4
Key 5	Key 6	Key 7	Key 8
Key 9	Key 10	Key 11	Key 12
Key 13	Key 14	Key 15	Key 16
Status Bar			

4. Key Definition

Each key has 5 settings: Function, Label, Parameter, Selection group and Radio group defined as shown:

```
FuncX=FUNC
LabelX=This is a label
ParamX=parameter
GroupX=integer number
RadioX=integer number
```

Where X is the key number from 1 to 16.

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

LabelX is just a string to be displayed inside the key rectangle.
It is generally mandatory with the exception of display functions.

You can write on multiple lines using the pipe character to jump to next line.

Example:

```
Label1=Matrix 1|zone 1
```

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
- applicable colors

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, LISTENZONE, 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
LabelX	mandatory
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
Label1=West|Corridor
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
```

Applicable Colors:

ColorUnavailable	when zone is unavailable (matrix absent/not working)
ColorFault	when zone has an immediate fault (loudspeaker line, amplifier fault no backup)
ColorDeferredFault	when zone has a deferred fault (backed-up amplifier, leak on loudspeaker)
ColorFree	when zone status is OK and not occupied
ColorMusic	when zone is occupied by a music source
ColorBusy	when zone is occupied by a microphone or message
ColorLocal	when zone is occupied by a local source (directly on to the matrix)
ColorEvac	when zone is occupied by an evacuation source
ColorSelect	when the key is selected
ColorActive	when zone is occupied by the PMIP-D / Vox@D itself

5.1.2 Reset zone selection

Deselect all zones (in specified selection group)

FuncX	RESET
LabelX	mandatory
ParamX	not used
GroupX	optional
RadioX	not used

Example:

```
Func1=RESET
Label1=Reset
Group1=3
```

Applicable Colors:

ColorResetOFF	no zone selected, nothing to reset
ColorResetON	there is a zone selection, you can press to clear it

5.1.3 Select the priority

Select the priority to use for next call/message.
Only one priority key is selected at a time.

FuncX	PRIO
LabelX	mandatory
ParamX	priority, number from -2 to 100
GroupX	not used
RadioX	not used

Parameter can be:

- 0, High Priority preset
- -1, Medium Priority preset
- -2, Low Priority preset
- any number from 1 to 100

If no **PRIO** key is selected, then the default priority of the client is used.

Example:

```
Func1=PRIO
Label1=Urgent
Param1=0
```

```
Func2=PRIO
Label2=Normal
Param2=-1
```

Applicable Colors:

ColorPrio	when key is not selected
ColorSelect	when key is selected

5.1.4 Microphone paging

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

FuncX	MIC
LabelX	mandatory
ParamX	not used
GroupX	optional
RadioX	not used

Example:

```
Label1=Talk
Func1=MIC
Group1=3
```

Call using zones selected in group 3.

Applicable Colors:

ColorMicOFF	when key not used
ColorMicTest	while processing and testing
ColorMicON	when microphone is active

5.1.1 Vu-meter

Vu-meter of microphone signal.

FuncX	VUMETER
LabelX	not used
ParamX	not used
GroupX	not used
RadioX	not used

5.1.2 Chime selection

Select pre-announcement chime for next chime paging.
Pressing on the key selects the next available chime.

FuncX	GONGSEL
LabelX	not used
ParamX	not used
GroupX	optional
RadioX	not used

Example:

```
Func1=GONGSEL
```

Applicable Colors:

```
ColorGongSel    unique color
```

5.1.3 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
LabelX	mandatory
ParamX	not used
GroupX	optional
RadioX	not used

Example:

```
Label1=Talk
Func1=GONG
```

Call using zones of the default selection group.

Applicable Colors:

```
ColorGongOFF    when key not used
ColorGongTest   while processing and testing
ColorGongON     when blinking or when microphone is active and there is no MIC key
```

5.1.1 Open program list window

FuncX	PROGLIST
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Example:

```
Func1=PROGLIST
Label1=Programs
```

Applicable Colors:

```
ColorProgList    unique color
```

5.1.2 Open message list window

FuncX	MSGLIST
LabelX	mandatory
ParamX	not used
GroupX	optional, for play/stop actions
RadioX	not used

Example:

```
Func1=MSGLIST
Label1=Messages
Group1=5
```

Play/stop actions in message list window will use selection group 5.

Applicable Colors:

```
ColorMsgList    unique color
```

5.1.3 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
LabelX	mandatory
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	
Label1=Message1	
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

Applicable Colors:

ColorMsg	when key is not selected
ColorSelect	when key is selected

ZONESEL colors are also used when the parameter defines a zones selection.

5.1.4 Play message

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

FuncX	MSGPLAY
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Applicable Colors:

ColorMsgPlay	unique color
--------------	--------------

5.1.5 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
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Applicable Colors:

ColorMsgStop	unique color
--------------	--------------

5.1.6 Play message in predefined zones

Start playing a predefined message into predefined zones.

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

For parameter syntax, refer to 0

Select a message.

Applicable Colors:

ColorMsgZoneON	when message is currently playing
ColorMsgZoneOFF	when message is not playing

5.1.7 Stop message in predefined zones

Stop playing a predefined message into predefined zones.

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

For parameter syntax, refer to 0

Select a message.

Applicable Colors:

ColorMsgZoneON when message is currently playing
 ColorMsgZoneOFF when message is not playing

5.1.8 Choose music source

Choose music source for currently selected zones.

FuncX	MUSICSOURCE
LabelX	mandatory
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
 Label1=Radio
 Param1=1

Func2=MUSICSOURCE
 Label2=CD
 Param2=2

Func3=MUSICSOURCE
 Label3=Music OFF
 Param3=0

Applicable Colors:

ColorMusicSource unique color

5.1.9 Listen message

Playback a message on the integrated loudspeaker. You must select a message first.

FuncX	LISTENMSG
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Example:

```
Func1=LISTENMSG
Label1=Listen|Message
```

Applicable Colors:

ColorListenMsg	unique color
----------------	--------------

5.1.10 Stop listening

FuncX	LISTENSTOP
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Example:

```
Func1=LISTENSTOP
Label1=Listen|Stop
```

Applicable Colors:

ColorListenStop	unique color
-----------------	--------------

5.1.11 Listen to zone

Listen to the source playing in a zone. You must select one zone first.

FuncX	LISTENZONE
LabelX	mandatory
ParamX	not used
GroupX	optional
RadioX	not used

Example:

```
Func1=LISTENZONE
Label1=Listen|Zone
```

Applicable Colors:

```
ColorListenZone    unique color
```

5.1.12 Reset listening

Reset to automatic listening function (last routed source).

FuncX	LISTENRESET
LabelX	mandatory
ParamX	not used
GroupX	optional
RadioX	not used

Example:

```
Func1=LISTENRESET
Label1=Listen|Reset
```

Applicable Colors:

```
ColorListenReset   unique color
```

5.1.13 Display loudspeaker volume

FuncX	VOL
LabelX	not used
ParamX	not used
GroupX	not used
RadioX	not used

Example:

```
Func1=VOL
```

Applicable Colors:

ColorVol	unique color
----------	--------------

5.1.14 Increase / Decrease loudspeaker volume

FuncX	VOLINC / VOLDEC
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Example:

```
Func1=VOLINC
Label1=Vol +
```

```
Func2=VOLDEC
Label2=Vol -
```

Applicable Colors:

ColorVolInc	unique color
ColorVolDec	unique color

5.1.15 Increase / Decrease matrix audio level

FuncX	VOLMATINC / VOLMATDEC
LabelX	mandatory
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
Label1=Matrix +
```

```
Func2=VOLMATDEC
Label2=Matrix -
```

Applicable Colors:

ColorVolMatInc	unique color
ColorVolMatDec	unique color

5.1.16 Increase / Decrease client audio level

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

Example:

```
Func1=VOLCLIINC  
Label1=Music +  
Param1=1
```

```
Func2=VOLCLIDEC  
Label2=Music -  
Param2=1
```

Applicable Colors:

ColorVolCliInc	unique color
ColorVolCliDec	unique color

6. General purpose functions

6.1.1 Label

No function. This is just a label.

FuncX	NONE OR LABEL
LabelX	mandatory
ParamX	not used
GroupX	not used
RadioX	not used

Example:

```
Func1=LABEL
Label1=This is a label
```

Applicable Colors:

ColorLabel unique color

6.1.2 Jump to another page

FuncX	JUMP
LabelX	mandatory
ParamX	page number
GroupX	not used
RadioX	not used

Example:

```
Func16=JUMP
Label16=Reset
Param16=3
```

Applicable Colors:

ColorJump unique color

6.1.3 Command selection

Select an additional 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.

<code>FuncX</code>	COMMANDSEL
<code>LabelX</code>	mandatory
<code>ParamX</code>	<code>commandON commandOFF</code>
<code>GroupX</code>	optional
<code>RadioX</code>	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**.

Applicable Colors:

<code>ColorCmdSel</code>	color when not active
<code>ColorSelect</code>	color when selected
<code>ColorCmdSuccess</code>	color when command succeeded
<code>ColorCmdFail</code>	color when command failed

6.1.4 Execute command (press-to-talk mode)

Execute a command when pressed.

Execute another command when released.

<code>FuncX</code>	COMMAND
<code>LabelX</code>	mandatory
<code>ParamX</code>	<code>commandON commandOFF</code>
<code>GroupX</code>	optional
<code>RadioX</code>	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
Label16=Press-To-Talk
Param16=talkstart|talkstop
```

Applicable Colors:

<code>ColorCmd</code>	color when not active
<code>ColorCmdTest</code>	color before sending commands
<code>ColorCmdSuccess</code>	color when command succeeded
<code>ColorCmdFail</code>	color when command failed

6.1.5 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
LabelX	mandatory
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
Label2=Mute UAP|Input 1
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.

Applicable Colors:

<code>ColorCmd</code>	color in "OFF" state
<code>ColorSelect</code>	color in "ON" state
<code>ColorCmdTest</code>	color before sending commands
<code>ColorCmdSuccess</code>	color when command succeeded
<code>ColorCmdFail</code>	color when command failed

6.1.6 Example of command with selection

```
Func1=COMMANDSEL
Label1=Zone 1
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
Label2=Zone 2
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
Label3=Press|to talk
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

6.1.7 Display variable value

Display a variable value in the label with automatic refresh.

FuncX	VARIABLE
LabelX	optional prefix or integer/string, label or for Booleans
ParamX	varid
GroupX	not used
RadioX	not used

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

For integers or strings, LabelX is optional.

If present, the variable value is displayed on the next line.

For Booleans, there are two display modes:

- unique label, the color of the key is used to reflect the Boolean value
- label1|label0, the first label is used when the value is 1 and the second label when the value is 0

Examples:

```
Func1=VARIABLE
Label1=volout1=
Param1=volout1
```

```
Func2=VARIABLE
Label2=Input 1
Param2=remin1
```

```
Func3=VARIABLE
Label3=Active|Inactive
Param3=remin1
```

Applicable Colors:

ColorVarOFF	color for integer/strings variable or when Boolean is 0
ColorVarON	color when Boolean variable is 1

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
LabelX	optional
ParamX	varid
GroupX	not used
RadioX	not used

Refer to 6.1.7 Display variable value for **ParamX** and **LabelX**.

Examples:

```
Func1=VARTOGGLE
Label1=Output 1
Param1=remout1
```

```
Func2=VARTOGGLE
Label2=Active|Inactive
Param2=remout1
```

Applicable Colors:

ColorVarOFF	color when variable is 0
ColorVarON	color when variable is 1