tpcontrol AMX Authorized Product **for Smartphone and Tablet devices**

Supported Command List

by Touch Panel Control

These commands are additional to AMX specific commands. AMX commands can be found in AMX PI.

Overview:
The purpose of the document is to provide an overview of commands specific to TPControl.
These commands are *additional* to the AMX specific commands supported.
TPControl is designed to recognise all AMX G4 touch panel commands for applicable supported features, and also includes support for legacy G3 touch panel commands.

Implementation:
For control of any TPControl enabled device, the following commands can be implemented within the NetLinx programming environment, or by parsing in the programming command port (0 - loopback port) of a TP4 design file.

An example TP4 file that demonstrates use of some of the Touch Panel Control API commands can be downloaded from this link:
http://www.touchpanelcontrol.com/guest/tpcontrol/ExampleTP4s/TPC%20API%20command%20examples.TP4

Supported commands:

TPCNotify *(1)*
 	
Notification messages can be presented when the device is running in Background multi-tasking mode.
 	
Apple device:
Presents a notification window when the device is running in background mode, titled "TPControl". If a <text message> is included, the text will be presented within the dialog notification window.
The notification window will provide 2 options; *Close* and *TPControl*.
• Pressing *Close* will close the notification.
• Pressing *TPControl* will return TPCNotify-Accept to the NI Master and the device will relaunch TPControl.
 	
Android device:
Presents a notification alert when the device is running in background mode, with title "TPControl". If a <text message> is included, the text will be provided within the notification list for review.
• Pressing the notification will return to the NI Master TPCNotify-Accept and the device will relaunch TPControl.
	Syntax:
	`TPCNotify-<text message>`
	
	Examples:
	`TPCNotify`
	`TPCNotify-A visitor is at the Front Door`
	`TPCNotify-There is an incoming call`
	
	NOTE: If the *Keep WiFi Active* option within device settings has been disabled, notifications will not be presented in multi-tasking mode, as TPControl disconnects when it loses application focus. Enable Keep WiFi Active to sustain connection when operating multi-tasking mode.
	
	NOTE: iOS5 introduced an enforced 2.5 minute application timeout, which overrides the Multitasking timeout feature. So, if the device is running iOS5+, TPControl will be forced to disconnect after ~2.5 minutes when running in multi-tasking mode.

TPCWarn_Intercom [1]
 	
All incoming or outgoing intercom functionality is indicated via a *Intercom Warning* icon relating to the current mode of Intercom operation.
 	
Modes indicated include:
TX (Outgoing audio only)
i.e. Device is being *Monitored*
 	
RX (Incoming audio only)
i.e. Receiving a *Page-All*
 	
TX/RX (Audio incoming and outgoing)
i.e. A two-way call is in progress
	Syntax:
	`TPCWarn_Intercom-<value>`
	
	Variables:
	`Show` (*default*)
	All Intercom warnings will be shown
	
	`Hide`
	No Intercom warnings will be shown
	
	`Posn,<x>,<y>`
	Updates the top-left draw position for the TPCWarn icons.
	All Intercom warnings will appear from the x,y location.
	
	Examples:
	`TPCWarn_Intercom-Show`
	`TPCWarn_Intercom-Hide`
	`TPCWarn_Intercom-Posn,100,100`

TPCURL and TPCCMD commands [2]***:***

TPCURL
 	
Device specific functionality such as;
- launching URLs
- launching applications
	Examples:
	`TPCURL-http://www.touchpanelcontrol.com`
	`TPCURL-itms-apps://itunes.com/apps/TPControl`
	`TPCURL-tel:+1234567890`
	`TPCURL-sms:+1234567890`
	
	`TPCURL-mailto:support@touchpanelcontrol.com?subject=Test%20subject&body=Great%20work!`
	
	For other schemes that may be supported by your device OS, please refer to:
	http://en.wikipedia.org/wiki/URI_scheme

TPCCMD
 	
TPControl specific configuration commands are prefixed with TPCCMD.
	Syntax:
	`TPCCMD-[<profileID>;]`
	`<cmd>[,<value>;<cmd>,<value>]...`
	
	Variables:
	`<profileID> = 1|2|3|4|5` (*optional*)
	If <profileID> is not declared, the command will be applied to ProfileID "1".
	
	`<cmd>`
	Refer to the Command List.
	
	`<value>`
	Refer to the Command List for specific values for each command.

TPCCMD: Command List	

Verify Verify/refresh the TPControl device licence based on information stored on Touch Panel Control servers.	Syntax and example: `TPCCMD-Verify` *NOTE*: The device must have unrestricted access to the Internet in order to complete the command successfully. A "Device successfully verified" message will be presented upon successful completion of the verification process.
Update This option provides the ability to update the TPControl TP4 file and/or Settings via the Internet.	Syntax and example: `TPCCMD-Update` *NOTE*: The device must have unrestricted access to the Internet in order to complete the command successfully. Management of device Settings and TP4 file uploads is provided via the token-associated account login at www.touchpanelcontrol.com/customer/account/tokens/
TPCCMD: Profile related options	

LocalHost
 	
Set the NetLinx Master Connection IP/ URL
	Syntax:
	`TPCCMD-[<profileID>;]`
	`LocalHost[:PortNumber],<IPAddress>`
	
	Examples:
	`TPCCMD-LocalHost,192.168.10.11`
	`TPCCMD-1;LocalHost,192.168.10.11:1319`
	`TPCCMD-2;LocalHost,192.168.1.101`

LocalPort
 	
Set the ICSP port number value (Default Port value is: 1319)
	Syntax:
	`TPCCMD-[<profileID>;]LocalPort,<PortNumber>`
	
	Example:
	`TPCCMD-LocalPort,1319`

eICSPu
 	
Set the Encrypted ICSP Username
	Syntax:
	`TPCCMD-[<profileID>;]eICSPu,<username>`
	
	Example:
	`TPCCMD-1;eICSPu,User 1`

eICSPp
 	
Set the Encrypted ICSP Password
	Syntax:
	`TPCCMD-[<profileID>;]eICSPp,<password>`
	
	Example:
	`TPCCMD-1;eICSPp,Pass1`

eICSP
 	
Enable or Disable Encrypted ICSP connection method
	Syntax:
	`TPCCMD-[<profileID>;]eICSP,<true|false>`
	
	Example:
	`TPCCMD-2;eICSP,true`

UseMobileData (*)
 	
Options available determine whether to use WiFi and/or MobileData to establish a connection with the NetLinx Master.
 	
(UseMobileData replaces the legacy Use3G API command)
	Syntax:
	`TPCCMD-[<profileID>;]UseMobileData,<false|withWiFi|withoutWiFi>`
	
	Variables:
	`false`
	Device will only use the WiFi adapter to establish a connection
	
	`withWiFi`
	Device will use the WiFi adapter to establish a connection if a connection is present. If no WiFi is available, MobileData will be used to establish a connection
	
	`withoutWiFi`
	Device will use only MobileData to establish a connection
	NOTE: MobileData may not be supported while an active WiFi connection is present.
	
	Example:
	`TPCCMD-1;UseMobileData,false`
	`TPCCMD-2;UseMobileData,withWiFi`
	`TPCCMD-2;UseMobileData,withoutWiFi`

DeviceID
 	
Set the Device ID number used upon connection to the NetLinx master.
	Syntax:
	`TPCCMD-[<profileID>;]DeviceID,<value>`
	
	Example:
	`TPCCMD-1;DeviceID,11001`

DeviceName
 	
Set the Device Name
	Syntax:
	`TPCCMD-[<profileID>;]DeviceName,<value>`
	
	Example:
	`TPCCMD-1;DeviceName,Cinema`

ProfileName (*)
 	
Provides the ability to name profiles
	Syntax:
	`TPCCMD-[<profileID>;]ProfileName,<value>`
	
	Variables:
	`<profileID> = 1|2|3|4|5` (*optional*)
	If <profileID> is not declared, the command will be applied to ProfileID "1".
	
	Example:
	`TPCCMD-ProfileName,Home`
	`TPCCMD-1;ProfileName,Home`
	`TPCCMD-2;ProfileName,Office`
	`TPCCMD-3;ProfileName,Home while in Office`

ApplyProfile (*)
 	
Provides the ability to recall the Settings stored within the declared profile.
If the profile is different from the current active profile, TPControl will disconnect the active connection, and attempt to connect using the new profile settings.
	Syntax:
	`TPCCMD-[<profileID>;]ApplyProfile`
	
	Variables:
	`<profileID> = 1|2|3|4|5` (*optional*)
	If <profileID> is not declared, the command will be applied to ProfileID "1".
	
	Example:
	`TPCCMD-ApplyProfile`
	`TPCCMD-1;ApplyProfile`
	`TPCCMD-3;ApplyProfile`

DefaultProfile (*)
 	
Provides the ability to set the default start-up profile for TPControl, which will be retained between sessions. Default is profileID "1".
 	
NOTE: Returning to TPControl from Home-screen or multi-tasking mode, does not represent restarting TPControl. As such, TPControl retains the previous active profile when returning.
	Syntax:
	`TPCCMD-[<profileID>;]DefaultProfile`
	
	Variables:
	`<profileID> = 1|2|3|4|5` (*optional*)
	If <profileID> is not declared, the command will be applied to ProfileID "1".
	
	Example:
	`TPCCMD-DefaultProfile`
	`TPCCMD-1;DefaultProfile`
	`TPCCMD-2;DefaultProfile`

QueryProfile (*)
 	
Returns a STRING including related profile data.
The current default profile and active profile information is included in the returned data.
	Syntax:
	`TPCCMD-[<profileID>;]QueryProfile`
	
	Variables:
	`<profileID> = 1|2|3|4|5` (*optional*)
	If <profileID> is not declared, the command will be applied to ProfileID "1".
	
	Response format:
	`ProfileInfo-<profileID>;`
	`LocalHost,<host_ip>:<local_port>;`
	`DeviceID,<device_id>;`
	`eICSP,<0|1>;`
	`DeviceName,<device_name>;`
	`UseMobileData,<false|withWiFi|withoutWiFi>;`
	`ProfileName,<profile_name>;`
	`DefaultProfile,<default_profile_number>;`
	`ActiveProfile,<current_active_profile_number>`
	
	Response Example:
	`ProfileInfo-3;`
	`LocalHost,192.168.10.12:1319;`
	`DeviceID,11001;`
	`eICSP,1;`
	`DeviceName,Galaxy Nexus;`
	`UseMobileData,withoutWiFi;`
	`ProfileName,Home while in Office;`
	`DefaultProfile,1;`
	`ActiveProfile,1`

Example concatenated commands
 	
Commands and their associated values i.e. <cmd>,<value> can be concatenated, so that multiple commands can be applied by sending one command expression.
Concatenate commands must be separated by semi-colons ';'
e.g. <cmd>,<value>;<cmd>,<value>
	Examples:
	`TPCCMD-1;LocalHost,192.168.10.2;DeviceID,11006`
	
	`TPCCMD-1;LocalHost,192.168.10.2:1319;DeviceID,11006`
	
	`TPCCMD-2;LocalHost,192.168.5.51:1319;DeviceID,11004`
	
	`TPCCMD-2;LocalHost,192.168.5.51;eICSP,true;eICSPu,User 1;eICSPp,Pass`

TPCCMD: Settings related options	

KeepWiFiActive
 	
When Enabled, TPControl will continue to keep a connection live with the NI Master when the device goes to sleep or another application takes device focus e.g. the device Home screen.
	Syntax:
	`TPCCMD-KeepWiFiActive,<true|false>`
	
	Example:
	`TPCCMD-KeepWiFiActive,true`

Gestures
 	
Enable or disable standard AMX gesture recognition.
 	
NOTE: Recommend disabling gestures when device Accessibility mode is enabled, due to gesture specific operation of Accessibility functions.
	Syntax:
	`TPCCMD-Gestures,<true|false>`
	
	Example:
	`TPCCMD-Gestures,true`

AutoLock
 	
When enabled, <true>, this will allow the Device to run the OS screen lock feature as set within the device settings. If AutoLock is set to <false> the screen will stay active until the Power button is pressed.
	Syntax:
	`TPCCMD-AutoLock,<true|false>`
	
	Example:
	`TPCCMD-AutoLock,true`

LockRotation
 	
Enable or disable screen rotation.
	Syntax:
	`TPCCMD-LockRotation,<true|false>`
	
	Example:
	`TPCCMD-LockRotation,false`

DarkScreen
 	
When enabled, provides a dark background loading image for TPControl as opposed to the default white background loading image. Avoid the potential for a bright white loading screen when in dark rooms by enabling this option.
	Syntax:
	`TPCCMD-DarkScreen,<true|false>`
	
	Example:
	`TPCCMD-DarkScreen,true`

DisableMultitouch
 	
Multitouch functionality (if supported) can be disabled/enabled with this option.
	Syntax:
	`TPCCMD-DisableMultitouch,<true|false>`
	
	Example:
	`TPCCMD-DisableMultitouch,false`

AccessibilityIncludeNoTextBtns
 	
When enabled, buttons with no text will be included when navigating during Accessibility VoiceOver mode.
	Syntax:
	`TPCCMD-AccessibilityIncludeNoTextBtns,<true|false>`
	
	Example:
	`TPCCMD-AccessibilityIncludeNoTextBtns,true`

AccessibilityLevelIncrement
 	
Defines the level increment/decrement percentage when a bargraph/level is selected during Accessibility VoiceOver mode.
Gesture Up/Down defines increment/ decrement action for the selected level.
	Syntax:
	`TPCCMD-AccessibilityLevelIncrement,<0-100>`
	
	Example:
	`TPCCMD-AccessibilityLevelIncrement,10`

InactivityTimeout
 	
TPControl will flip to the Inactivity page that has been defined within the properties of the TP4 file, based on the timeout defined.
	Syntax:
	`TPCCMD-InactivityTimeout,<Never|0|1|2|5|10|15|30|60|120|180|240>`
	
	Example:
	`TPCCMD-InactivityTimeout,Never`

ScreenResize
 	
Options allow the TP4 project file to be presented in various formats: No-scaling (None), Scale-to-fit (Scale), and Stretch-to-fit (Stretch).
	Syntax:
	`TPCCMD-ScreenResize,<None|Scale|Stretch>`
	
	Variables:
	`None`
	TP4 project will be presented in the original TP4 resolution
	
	`Scale`
	Upsizes/downsizes to maintain original aspect ratio of the TP4 file to fully extend to fill at least one dimension of the device display. Scale-to-fit.
	
	`Stretch`
	Upsizes/downsizes to fill the usable native screen area of the device display. Stretch-to-fit.
	
	Example:
	`TPCCMD-ScreenResize,Scale`

ProfilePrompt
 	
When enabled, TPControl will present a dialog requesting confirmation of the connection profile to use whenever returning from the Home-screen or multi-tasking mode. This is independent of whether an active profile connection exists or not.
 	
NOTE: Within supported versions of TPControl, the "Not Connected" dialog presents any *named* profiles for selection.
The following named or unnamed profiles will always be shown:
• the current active profile
• the current default profile
• any named profile
	Syntax:
	`TPCCMD-ProfilePrompt,<true|false>`
	
	Example:
	`TPCCMD-ProfilePrompt,false`

ButtonHit
 	
When enabled, Button Hit produces a "Beep" sound when a valid button area is pressed within the touch panel design file.
	Syntax:
	`TPCCMD-ButtonHit,<true|false>`
	
	Example:
	`TPCCMD-ButtonHit,true`

ButtonMiss
 	
When enabled, Button Miss produces a "Double Beep" sound when any area outside of a valid button area is pressed within the touch panel design file.
	Syntax:
	`TPCCMD-ButtonMiss,<true|false>`
	
	Example:
	`TPCCMD-ButtonMiss,true`

BeepLevel
 	
Sets the level at which the volume for the Beep will be announced.
	Syntax:
	`TPCCMD-BeepLevel,0-100`
	
	Example:
	`TPCCMD-BeepLevel,30`

DeveloperMode
 	
When enabled, this will allow TPControl to communicate with TPTransfer.
	Syntax:
	`TPCCMD-DeveloperMode,<true|false>`
	
	Example:
	`TPCCMD-DeveloperMode,true`

TransferPort
 	
The port used by TPControl to communicate with TPTransfer. (Default port value is: 10700).
	Syntax:
	`TPCCMD-TransferPort,<integer>`
	
	Example:
	`TPCCMD-TransferPort,10700`

IntercomCallNotify
 	
When enabled, if TPControl is running but does not currently have application focus, an alert notification will be presented on the device.
 	
NOTE: this notification occurs only if corresponding intercom call initiation commands are received by TPControl. Also *KeepWiFiActive* must be enabled within Settings to maintain connection with the NI Master when running in Background multi-tasking mode.
	Syntax:
	`TPCCMD-IntercomCallNotify,<true|false>`
	
	Example:
	`TPCCMD-IntercomCallNotify,true`

STTAutoHide (*)
(Speech-to-Text, AutoHide)
 	
When disabled, TPControl will hide the Speech-to-Text recording dialog only after a successful result is processed.
 	
When enabled, TPControl will hide the Speech-to-Text recording dialog once processing of recorded audio commences, irrespective of the result.
	Syntax:
	`TPCCMD-STTAutoHide,<true|false>`
	
	Example:
	`TPCCMD-STTAutoHide,true`

STTDisplayResult (*)
(Speech-to-Text, DisplayResult)
 	
The result of Speech-to-Text analysis can be displayed via a brief notification on-screen.
 	
NOTE: Errors in processing will still solicit a notification.
	Syntax:
	`TPCCMD-STTDisplayResult,<0|1|2|3>`
	
	Variables:
	0: Do Not Display
	1: 1 result will be displayed (default)
	2: Up to 2 results will be displayed
	3: Up to 3 results will be displayed
	
	Example:
	`TPCCMD-STTDisplayResult,1`

TTSOfflineMode (*)
(Text-to-Speech, OfflineMode)
(Android only)
 	
When enabled, TPControl will utilise device defined language packs to determine text-to-speech translation.
 	
When disabled, TPControl will utilise online resources to determine text-to-speech translation.
 	
NOTE: Results are cached with each completed translation when operating in online mode.
	Syntax:
	`TPCCMD-TTSOfflineMode,<true|false>`
	
	Example:
	`TPCCMD-TTSOfflineMode,true`

TTSClearCache (*) Clears the text-to-speech cache which is created during online operation.	Syntax and example: `TPCCMD-TTSClearCache`
RestoreAllSettings This will restore all settings within the device Settings page back to defaults	Syntax and example: `TPCCMD-RestoreAllSettings`
ClearUserPages The design file will be removed and the original Demo Pages will be loaded back onto the device.	Syntax and example: `TPCCMD-ClearUserPages`
DownloadDemo Downloads the most recent TP4 demonstration file from Touch Panel Control.	Syntax and example: `TPCCMD-DownloadDemo` *NOTE*: The device must have unrestricted access to the Internet in order to complete the command successfully.
ReprocessTP4 Clears any caching, and reprocesses the installed TP4 file. This is the same process that runs whenever a file is transferred to the device.	Syntax and example: `TPCCMD-ReprocessTP4`

Speech-to-Text and Text-to-Speech commands (*):

Speech-To-Text
 	
LISTEN
 	
When *Speech-to-Text* is activated, a recording window will appear on the device prompting you to speak.
 	
An option for selecting supported languages is provided.
 	
At the first discernible pause during speech, recording will automatically end, and processing will commence.
 	
The result of the speech analysis will be parsed to the NetLinx master in string format. The string is based on the language definition and will include up to three (3) closely matched results.
 	
NOTE: The primary result <result1> will be displayed on the device display.
	Syntax:
	`TPCSTT`
	`TPCSTT-LISTEN`
	
	Example:
	`TPCSTT`
	
	Response format:
	`TPCSTT-<result1>;<result2>;<result3>`
	
	Response Example:
	`TPCSTT-lights on;light song;like song`
	
	NOTE: The device must have unrestricted access to the Internet in order to complete the command successfully.

END Ends recording, and processes the recorded result when applicable. The recording window will be removed from the UI.	Syntax: `TPCSTT-END` Example: `TPCSTT-END`

Text-To-Speech
 	
Android:
Text to Speech will operate in one of two modes; offline or online, as determined by *TTSOfflineMode*.
 	
iOS:
Text to Speech will operate in online mode only.
 	
Where applicable, the device will present the audio, and the result of processing will be parsed to the NetLinx master in string format.
	Syntax:
	`TPCTTS-<text>;<locale>`
	
	Variables:
	`<text>`
	The text to convert to speech
	
	`<locale>`
	The target language for the device to speak. Refer to appendix for suggested <locale> list.
	
	Example:
	`TPCTTS-The lights are at 65%;en`
	
	Response format:
	`TPCTTS-<OK|Unsupported Language/Locale|`
	`Error_tts|Error_player>`
	
	NOTE: When operating in online mode, the device must have unrestricted access to the Internet in order to complete the command successfully.
	
	NOTE: When operating in offline mode, a supported language pack (locale) must be installed on the device in order to function as intended.

TPCLockOut commands supported in TPControl for Android (^):
TPCLockOut functionality may vary between different Android devices. Factors that can affect the intended operation can relate to the operating system version installed, and manufacturer or service-provider software operating on the device.

LockOut
 	
Enabling *LockOut* will restrict usage of the "Home", "Back", "Menu/Setting", and "Search" soft-or-external button functions on the Android device. TPControl retains application focus, and if TPControl detects that application focus has been lost, will attempt to automatically regain focus.
 	
Restricted access to TPControl *Settings* is provided through a pin-code (see *LockOutPin*) when *LockOut* is enabled.
	Syntax:
	`TPCCMD-LockOut,<true|false>`
	
	Example:
	`TPCCMD-LockOut,true`

LockOutPin
 	
When the *LockOut* functionality is enabled, access to TPControl Settings is restricted by entry of a pin-code. The LockOutPin pin-code can be updated using this command. There is no restriction on pin-code length.
	Syntax:
	`TPCCMD-LockOutPin,<ascii-numeric>`
	
	Example:
	`TPCCMD-LockOutPin,1234567890`
	
	NOTE: Non-numeric characters parsed will invalidate the command.

AutoLaunch
 	
Option to automatically launch TPControl when the device boots up.
	Syntax:
	`TPCCMD-AutoLaunch,<true|false>`
	
	Example:
	`TPCCMD-AutoLaunch,true`

(1) Added in: v2.0.0.0
(2) Added in: v2.2.0.0
(^) Added in: Android v2.2.0.11 (TPCLockOut pre-release)
(*) Added in: v2.3.0.0

APPENDIX

Text-to-Speech language and Locale information

Language	Locale
Afrikaans	af
Albanian	sq
Arabic	ar
Armenian	hy
Catalan	ca
Chinese	zh
Croatian	hr
Czech	cs
Danish	da
Dutch	nl
English	en
Esperanto	es
Finnish	fi
French	fr
German	de
Greek	el
Haitian Creole	ht
Hindi	hi
Hungarian	hu
Icelandic	is
Indonesian	id
Italian	it
Japanese	ja
Korean	ko
Latin	la
Latvian	lv
Macedonian	mk
Norwegian	no
Polish	pl
Portuguese	pt
Romanian	ro
Russian	ru
Serbian	sr
Slovak	sk
Spanish	es
Swahili	sw
Swedish	sv
Tamil	ta
Thai	th
Turkish	tr
Vietnamese	vi
Welsh	cy

Problem Reporting

We aim to make your device integration as seamless as possible within your AMX environment. If you encounter any difficulties using the product or any of its facilities, please let us know and we will be happy to help you.

The helpdesk on our website at support.touchpanelcontrol.com operates from:

- Monday to Friday; 09:00 to 18:00 (GMT)
- Monday to Friday; 09:00 to 17:00 (AEST)

Touch Panel Control Team.