APPLE TV IP REMOTE FOR CRESTRON

Revision: 1.14
Date: 20 December 2016

SUMMARY

This datasheet relates to Ultamation's Apple TV IP control module for Crestron control systems. It provides the essential information for integration between an Apple TV and the Crestron control processor, and for programming of the module with a host Crestron program.

This module is compatible with 3-Series Crestron systems ONLY and has been tested with Apple TVs of generation 2, 3 and 4.

INSTALLATION NOTES

The Crestron system communicates with the Apple TV directly via an IP connection. No special wiring or configuration is required outside of that needed for a standalone Apple TV installation.

Equipment Setup

As of version 1.10, the module is able to communicate with an Apple TV by performing Apple's Bonjour device discovery and matching up against the device's configured name. The original mechanism of using a static IP address has been retained for backwards compatibility but is now deprecated.

The Apple TV requires three configuration steps to be made:

1. Set the name of each Apple TV to the unique label you have selected for your installation and matches the name entered at the point of module purchase. We recommend that you test that your selected Apple TV name is valid before purchasing a licence.
2. Turn on 'Home Sharing' on each Apple TV. Each device can use a different Home Sharing Id if desired, but this MUST match the Id used at the point of purchase.
3. Set the Apple TV Sleep setting to 'Never'.

For information on how to make the changes on the Apple TV, please consult Apple's online support site: https://www.apple.com/support/

Module Parameters

AppleId – Enter the Apple Home Sharing Id for the device. This MUST match the same Home Sharing Id that has been used to BOTH set up the Apple TV and was entered at the time of purchase.

Password – You can either 'hard code' the Apple Home Sharing Id password in the module parameters OR pass it to the module via the Password$ serial input. Using the serial input allows the password to be entered programmatically – say – via an on-screen keyboard (this is not included). In this case, one should enter the empty string ("") in parameter field.

A Note on Home Sharing Authentication:

The password is only required for a ONE-TIME authentication with Apple's Home Sharing service. Once this has been completed successfully, the Crestron system will cache an authentication token for each Home Sharing Id in a file \NVRAM\Ultamation\hs.cache. Authentication only requires a valid Internet connection and does not require the Apple TV itself. This cache file can be moved between processors so the authentication can be performed in a development environment and then moved to the production system, even if that system doesn't have an active Internet connection. Subsequently changing the Home Sharing password does not change the authentication token.

Please Note – Apple now provide "Two Factor Authentication" for Home Sharing Ids and this is incompatible with the Crestron integration. However, this authentication step is only required ONCE, the first time the module is used. Two factor authentication should be temporarily disabled until the "hs.cache" (see above) file has been generated.

Some users have reported that they cannot complete Home Sharing authentication and receive a "Home Sharing Failed" message through the diagnostics output. This is usually because the DNS settings on the control processor are not complete or unable to resolve Apple's servers on the Internet. Please ensure your DNS settings are correct.

DeviceName – You must enter the name of the Apple TV to be controlled. This must match the address that was entered at the time of purchase. Entering anything in this field will direct the module to use Bonjour device discovery to find the Apple TV to be controlled, and the DeviceAddr parameter will be ignored.

DeviceAddr – DEPRECATED – If using an old licence, enter the IP address of the Apple TV to be controlled. This must match the address that was entered at the time of purchase and should be in a valid dotted quad format. e.g. 192.168.1.150. Note: DO NOT use zero prefixes in any octet. i.e. 192.168.001.150 is NOT valid. For this parameter to be used, DeviceName must be empty (i.e. "", and NOT 0d)

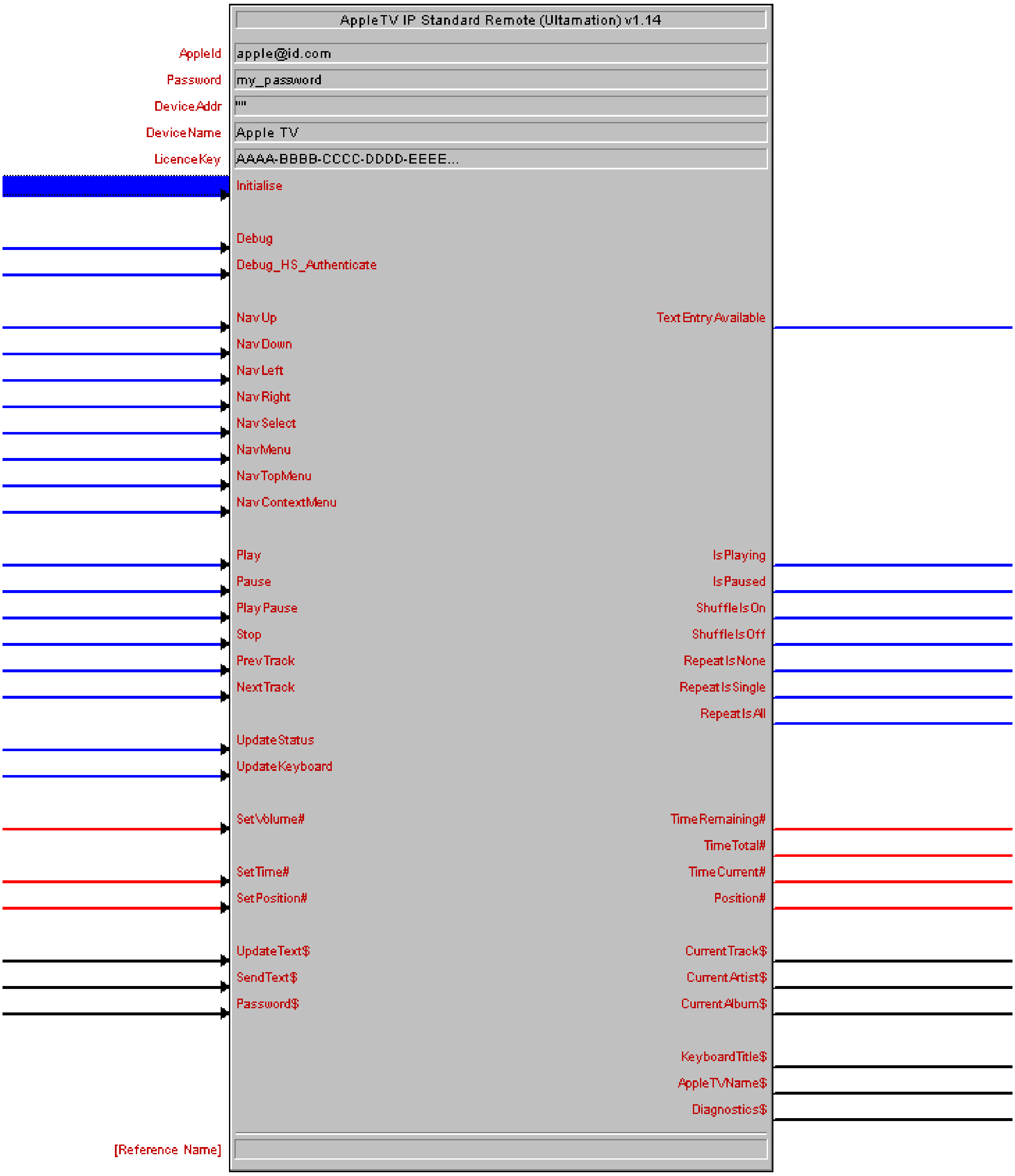
LicenceKey – The module will function, without any restrictions, for a limited time without a licence key so that integrators can "try before you buy". For continued use, the module requires a licence key that is generated at the time of purchase from the Ultamation Store and is linked, at that time, to the information provided for Home Sharing Id and Device Name. The licence key will be delivered via email to the address linked to the account used at checkout. Only a valid 'Home Sharing Id – Name (or IP Address) - Licence Key' combination will enable the module. NOTE: This should not be confused with adding the module to your cart in Crestron's Application Market, which is a free download of the "try before you buy" module.

PROGRAMMING NOTES

Each of the module files (see below) should be placed either in the host program's project folder, or to make the Apple TV control module available to all Crestron programs, in the SIMPL Windows installation's User SIMPL+ (for .usp, .ush and .clz files) directories. This pdf should be placed in the same directory for SIMPL's F1 help function to work properly.

The module consists of a single SIMPL+ wrapper module to the core SIMPL# driver:

- AppleTV IP Standard Remote (Ultamation) v1.14.usp & .ush
- AppleDACPClient.clz (DLL version 1.14.0.*)
- AppleTV IP Control Module Datasheet (Ultamation) v1.14.pdf (this file)



AppleTV IP Standard Remote (Ultamation) v1.14

AppleId apple@id.com

Password my_password

DeviceAddr ""

DeviceName Apple TV

LicenceKey AAAA-BBBB-CCCC-DDDD-EEEE...

Inputs

Initialise – For installations with a single Apple TV, you can place a 1 on this signal to initialise the module. For larger installations, we suggest you pace the initialisation of the modules by a short (<1s) time.

Debug – In situations where the module is not authenticating with Home Sharing, or not communicating with an Apple TV, asserting "Debug" high will enable extended debug information to be sent to the Crestron Console. This will detail most of the important events as they occur, such as device discovery, authentication, login and command success or failure. DO NOT LEAVE THIS ENABLED IN A CLIENT PROGRAM.

Debug_HS_Authenticate – This digital signal SHOULD not be used under normal conditions. The module will automatically attempt to authenticate the Home Sharing credentials with Apple if the authentication token is missing from the hs.cache file. On a rising edge, the driver will clear any existing authentication token for the specified Home Sharing Id and attempt to retrieve a new one. In general, this can be commented out with '//'. Note: This signal was previously called "ForceAuthentication" but we have renamed as it appeared to cause some confusion when debugging connection issues.

Nav[*] - These signals provide navigation control. A rising edge will send a single command. Holding these signals high will not repeat. The ContextMenu provides, as the name suggest, a context specific menu depending on where you are in the device's UI and, at times, does nothing.

Play, Pause, etc. – These signals provide simple transport control. Again, the effect will depend on the current state of the device. i.e. Play will only function if the device has a track or video clip queued, and SkipFwd/Back have no function if a video clip does not contain chapter index marks.

UpdateStatus – A rising edge on this signal will poll the device for basic status information. The module provides limited feedback in the form of playing state and track position.

Feedback can be polled manually using this signal, and prior to version 1.10 polling had to be handled by the host program.

As of version 1.10 the module will poll the device following select/transport operations automatically, removing the burden from the host program. Polling will continue while the Apple TV is playing content, and then stop when the Apple TV is idle. If this behaviour is insufficient, you can continue to use UpdateStatus manually.

UpdateKeyboard – A rising edge on this signal will poll the device for the current state for text entry which is required prior to sending data via 'UpdateText$'. Please refer to the "Notes on Text Entry" below.

SetVolume# - On 2nd and 3rd generation Apple TVs, the output volume can be controlled using this signal (analog, unsigned 0-65535, or 0-100%). Care should be taken not to initialise this value to 0, as it may appear that the Apple TV is muted. Currently, this appears to have no effect on the 4th generation device.

SetTime# - This analog provide random access to a position within the currently playing audio track or video clip. The analog value should be expressed in 'seconds' – i.e. 3 minutes and 15 seconds would be expressed as 195 (3*60 + 15). Requesting a time beyond the end of the playing media will snap to the end of the media. Requesting a time when no media is selected will have no effect.

SetPosition# - This analog signal provides random access to a position within the currently playing audio track or video clip. The signal should ONLY be set AFTER a successful UpdateStatus message, as the track/clip length is required prior to requesting a new position. Performance varies between the Apple TV generations, but we have tested this signal connected directly to a SmartGraphics^{TM} slider and the results, particularly on the 4th generation, are excellent. The input range is specified as a percentage of total track length (0-65535 or 0-100%).

UpdateText$ - When the Apple TV is in a state where it can accept text entry (See 'Notes on Text Entry' below) you can send the current contents of the text entry box using this signal. Note this should be the complete string, and not character-by-character.

SendText$ - When the Apple TV is in a state where it can accept text entry (See 'Notes on Text Entry' below) you can send a complete string, and 'submit' using this serial input. What 'submitting' means will vary depending on the function being used – e.g. Searching for content usually results in the text box being dismissed and the search results brought into focus.

Password$ - The password serial is only used when the password parameter is left empty (""). Passing the complete Home Sharing password to this signal in a single logic wave will do the same as the ForceAuthentication signal but using this input as the Apple Home Sharing password.

Notes on Text Entry:

Some care is required when using the text entry feature as the Apple TV will only accept text when in specific states – i.e. the text entry box has focus. The module provides a mechanism to 'initialise' text entry when required, via the 'UpdateKeyboard' digital signal.

So, before using text entry, you must pulse the 'UpdateKeyboard' signal high. This will then query the Apple TV for the current keyboard state and the module will bring the 'TextEntryAvailable' signal high if, and only if, the Apple TV is currently in a mode which will accept text entry. In addition, the 'KeyboardTitle$' serial output will provide – if possible – a title for the currently active keyboard. If the Apple TV is not in a suitable text entry state, the 'TextEntryAvailable' signal will be low. The 'UpdateKeyboard' signal must be pulsed before sending the first string on 'UpdateText$', but need not be pulsed before subsequent 'UpdateText$' updates while the text entry box on the Apple TV is still in focus.

An example use case of this might be; On a TSW panel, the UI displays a page of controls for the Apple TV and a 'keyboard' icon.

Pressing the keyboard icon would trigger the 'UpdateKeyboard' signal. If 'TextEntryAvailable' goes high, a popup text entry field could be displayed and focus given to the field, which – in turn – brings up the TSW built-in keyboard. This also initialises the text entry feature for the current on-screen Apple TV keyboard.

The text entry field string output is then sent to the 'UpdateText$' input signal. The Apple TV will dynamically update to show any search results etc.

Additionally, pressing the 'Enter' key on the on-screen keyboard could send the text field string to the 'SendText$' signal which instructs the Apple TV to terminate the current text entry session. You should close the keyboard at this point by clearing focus from the text entry field.

NOTE: There is also a limitation in that the module cannot send text for 'secure' data, such as passwords. Therefore, you must use the on-screen navigation for such situations. Warning! The text WILL be sent, and appear as a series of dots in the text box, but the data itself is NOT valid.

TECHNICAL NOTE: The 'SendText$' signal is a little more forgiving than 'UpdateText$' since the module internally processes a KeyboardUpdate before sending the completed text. This means that the module will always attempt to initialise the text entry session before sending the text – it does NOT guarantee that the text will be accepted by the Apple TV though!

Outputs

A small amount of status information is requested from the Apple TV device when the UpdateStatus signal is triggered. Most signals are self-explanatory, so we will only detail those that require special mention.

TextEntryAvailable – This signal provides the result from the 'UpdateKeyboard' signal. It is polled, so care must be taken not to assume that the reported state is valid. Immediately following an 'UpdateKeyboard' pulse, this signal will be high if and only if the Apple TV is ready to take text entry updates.

Track/Video Clip Timings – This information is only valid when a track or clip is playing or paused. When the Apple TV has no content queued, these values will all report zero.

TimeCurrent#, Remaining#, Total# - are all represented in seconds. i.e. a value of 195 would represent 3m 15s.

Position# - This is a full-scale (0-65535 or 0-100%) value representing the current position in a track or video clip and can be used to directly drive a gauge or slider.

CurrentTrack$/Artist$/Album$ –FOR AUDIO PLAYBACK ONLY. These signals provide information about the currently playing track. This information is not reported for video playback.

*KeyboardTitle$* - Following an 'UpdateKeyboard' pulse, if the Apple TV is ready to take 'UpdateText$' or 'SendText$' serial data, the current keyboard title (e.g. "Podcasts") will be returned.

AppleTVName$ - When the module initially connects to the Apple TV, it queries the device for its name. This can then be used in the Crestron UI. If the Apple TV name is changed, the module will not reflect this change until either the program is rebooted, or the module disconnects from the Apple TV (which can happen during normal communications, but is not deterministic).

Diagnostics$ - This is a special serial signal which will provide limited diagnostics information about the Home Sharing/Session Authentication status. It provides nothing useful in a working context and there is no reason to present this to the UI. If you need additional information for debugging, please refer to the Debug digital input.

Licence

This module (including software, images and any and all other associated assets distributed as part of the purchased download package) is licenced on a PER DEVICE basis.

A licence key is generated at the point of purchase and is linked at that time to specific information that MUST be provided at the time of purchase. A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

The licence key for each device will be delivered via email along with links to download the module. There is no physical delivery.

The module is provided without any warranty with respect to Apple's Home Sharing authentication mechanism or remote control protocol. We will endeavour, through best efforts, to maintain the module's functionality and any bug fixes will be provided free-of-charge. Additional functionality may be released as a variation of this module and this will be a separate, purchasable, product.