

APPLE TV IP REMOTE V2 FOR CRESTRON

Revision: 2.50

Date: 06 July 2021

SUMMARY

This datasheet relates to the second of Ultamation's Apple TV IP control modules for Crestron control systems. This is a completely new driver, using a different protocol from the original v1.xx driver and there are no overlaps whatsoever. 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 & 4-Series Crestron systems ONLY and has been tested with Apple TVs of generation 4 and 5.



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.

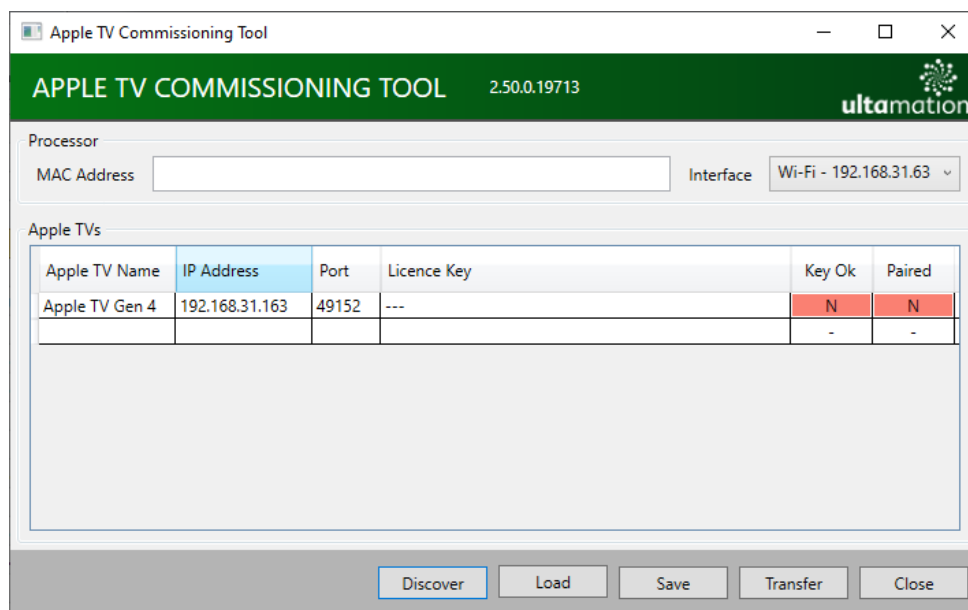
Environment Setup

This new version of the module comes with a separate utility program to ease the Crestron configuration by handling the pairing process on a PC rather than needing to accommodate pairing logic within the host program itself.





Please ensure you read the following document carefully to minimise any potential problems, but if you have trouble, email support@ultamation.com and we will help as best we can (providing you read the documentation!)

The Apple TV Commissioning Tool

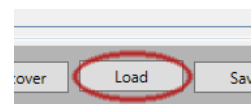
This small Windows application is used to create the configuration file that is used by the module instances running on your Crestron processor. Please ensure you have downloaded the latest version (v2.50 upwards) of the commissioning tool for MAC Address licencing. This can be checked in the green area, at the top of the application.








The configuration file contains the following essential information, and if this file does not exist on the Crestron processor, the Apple TV integration will not work.

-  Crestron processor MAC Address
 This is purely informational and allows the module to check it is running on the expected processor – a warning is generated in the error log if this information is incorrect.
-  Pairing identifiers and keys
 These are generated during the pairing process and are required by the module instances to allow them to authenticate with the paired Apple TVs.
-  Apple TV Instances
 A list of paired Apple TVs – without pairing, there will be no control.
-  A licence key for each Apple TV
 Each module instance is licenced to a specific processor MAC Address/Apple TV name. Without a valid licence, the module will run for an hour and then stop. This allows you to test the module in trial mode.

Each time the application is started, a new pairing identifier and private key is generated for the configuration to use. If you wish to modify an existing configuration and retain the current pairing information, you **MUST** edit an existing configuration file using the Load configuration function



Buttons Explained

-  **Discover**
 The application is always listening for Apple TVs that support the Media Remote protocol to advertise themselves on the network. When a message comes in, the Apple TV will be show in the list above. However, you can force a discovery request by clicking the Discover button. Note: If this doesn't work first time, there is probably a deeper issue with communicating to the Apple TV on your network.
-  **Load**
 As mentioned above, this allows you to load an existing "atvconfig.json" file.
-  **Save**
 This allows you to save a working copy of the Apple TV configuration. The default filename (and what MUST be present on the Crestron control system) is "atvconfig.json", though you can call it whatever you like.
-  **Transfer**
 Although you can manually copy the "atvconfig.json" file to your Crestron processor using a tool such as FileZilla, the commissioning tool also provides a simple FTP mechanism to put the file in the correct place. This does not support authentication.
-  **Close**
 Closes the application – make sure you save your config or transfer it to your processor first!

The Commissioning Process

The first step in the process is to enter your target processor's MAC Address. This is expected to be 6 octets and can be obtained via the System Info tool within Toolbox. It should look something like "00:10:7f:69:d3:4a"

Any Apple TVs on your local network should have appeared in the Apple TVs list below. You *can* enter the information manually if you wish to prepare for on-site commissioning, though you will still need to complete the pairing when on-site using the previously created configuration file. Any manually configured devices will show IP and Port as "unknown/0". The IP address and Port of discovered devices is read only and only provided for information – unlike the previous version of the module, the Media Remote protocol requires Bonjour.

Serial Number 1234567					
Apple TVs					
MANUALLY ENTERED INFORMATION					
Apple TV Name	IP Address	Port	Licence Key	Key Ok	Paired
Other Apple TV	unknown	0	ABCD-EFGH-IJKLM-NOPQ-RSTU	N	N
				-	-

You can also remove unwanted Apple TV definitions from the list by selecting them and pressing the "Delete" key.

When you enter a valid key for the MAC address/Apple TV name combination, the "Key Ok" field will turn GREEN. If you use the configuration when this field is RED, the module will operate in a time limited trial mode for 1 hour following each program reset.

Serial Number

12345678

Apple TVs

Apple TV Name	IP Address	Port	Licence Key	Key Ok	Paired
Other Apple TV	unknown	0		Y	N

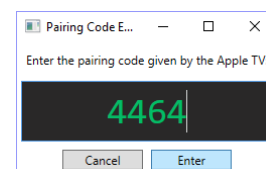
Pairing

In order to communicate with each Apple TV, the device must be paired using the commissioning tool. You do NOT pair the Apple TV from within the Crestron program. This both simplifies the host program and also provides an easy to use commissioning tool that non-technical dealers, and perhaps even some "engaged" clients can use.

To pair each Apple TV, open the context menu (right mouse click) on the Apple TV line.

Note: This will only work if the Apple TV is currently accessible (i.e. an IP address and Port have been discovered). You must also be able to see the Apple TV's video output on a TV.

When you select the "Pair with Apple TV" option, the Apple TV will display a 4-digit code on the display. At the same time, the commission tool will ask you to enter the 4-digit code. Carefully type the code shown on-screen into the code entry box.



When successfully paired, the "Paired" field will turn GREEN and the configuration file will now contain the essential pairing keys for the associated Apple TV.

The Apple TV will similarly now hold pairing information for the control endpoint defined by the configuration file, and you will see "Unknown Device" (we're looking into that!) listed in the paired devices on the Apple TV.

Creating a new configuration file or removing the pairing from the Apple TV will require that you repeat the pairing process.

Please note: The licence key is NOT bound to the pairing process, but it IS bound to the Apple TV's name and Crestron processor's MAC Address. You can therefore pair a different Apple TV (should you need to replace a device) without a new licence key, providing the Apple TV name remains the same.

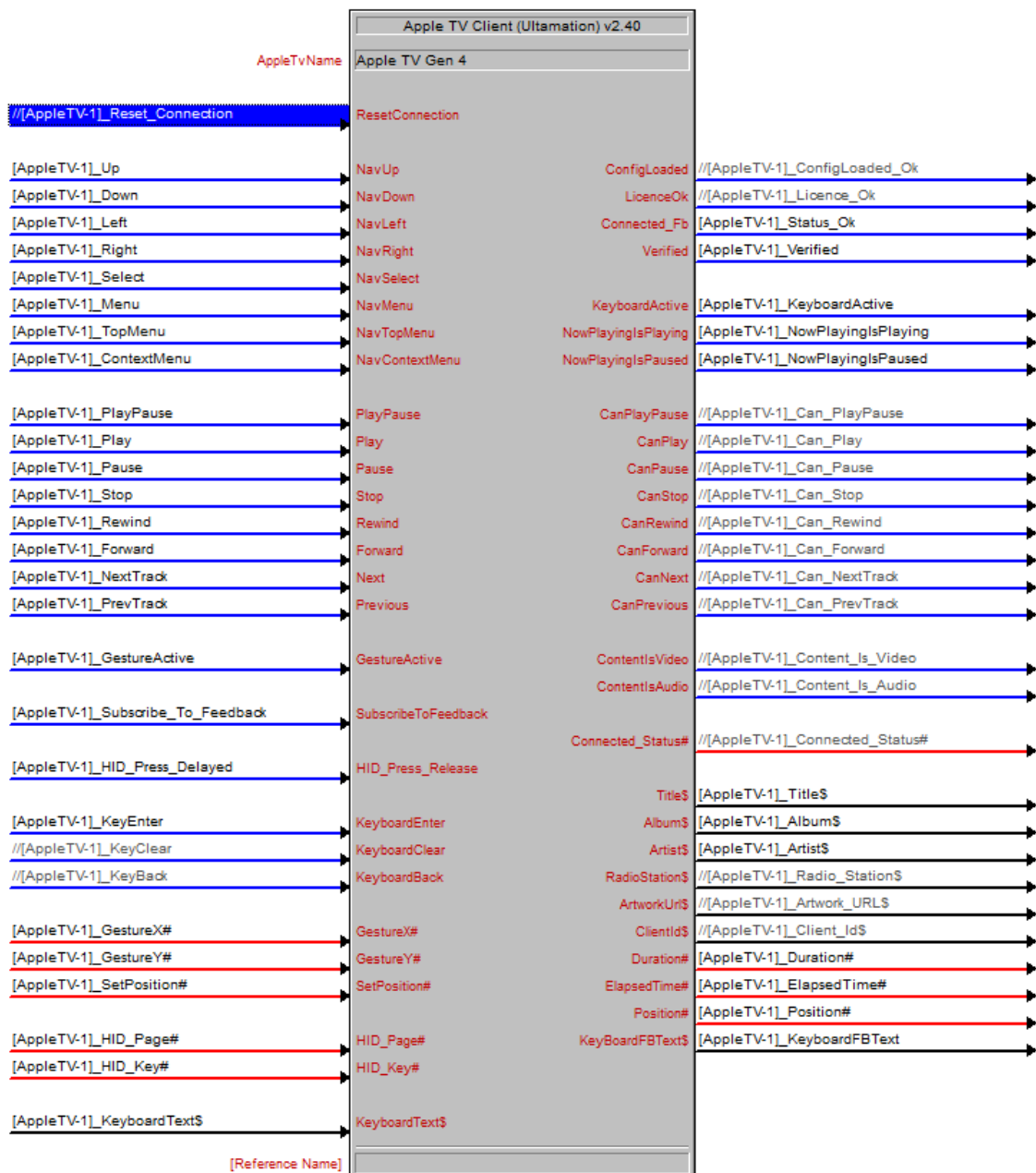
Please also note: The atvconfig.json file will be transferred to /NVRAM/Ultamation in the processors file structure, if you wish to transfer the file manually via a tool such as Filezilla then you MUST use the same file path. The processor slot which holds the program for the Apple TV Module would also need to be rebooted if an atvconfig file is uploaded/transferred to the processor while the program is live.

The Apple TV Client Module

Usage - Multiple Program Slots

Spreading Apple TV module instances across program slots must be avoided. A separate instance of the MDNS Listener would be created for each program slot where an Apple TV module is placed, but only the FIRST will receive Bonjour updates. The result is that Apple TVs in subsequent slots will never receive notification of their IP addresses.

Please set your Apple TV to never go to sleep via the Settings/General/Sleep After.



Module Parameters

AppleTVName – Enter the Apple TV's Name, This can be found under Settings/General/About on the Apple TV, if this is entered incorrectly or changed afterwards on the Apple TV then the module will stop functioning. The module does not

use Mac address nor the IP Address to communicate with the Apple TV therefore no option given to use them as parameters.

Inputs

ResetConnection – NOTE: This signal is NOT required for normal operation and is provided only to recover from an inconsistent state.

This signal will force a disconnection for a current connection and internally clean up the connection state, and then attempt a new connection. It will also re-transmit the Bonjour device discovery packet to attempt to solicit the current Apple TV settings in case they have changed (e.g. due to a reboot). The signal should only be pulsed and has no effect if the Apple TV has not already been discovered via Bonjour.

Select, NavL, NavR – These signals provide navigation control. A rising edge will send a single command. Holding these signals high will repeat until released. 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 Skip Forward/Back have no function if a video clip does not contain chapter index marks (Nav left and right can also be used to Rewind and Fast Forward.)

SubscribeToFeedback – This signal enables or disables feedback subscription. We have found that subscribing to feedback can degrade performance on the control system in some cases so this signal provides you with a mechanism to selectively enable feedback when desired.

Please note some apps supply very large packets when supplying feedback. This was slowing down the processors and causing the CPU load to reach 100%. For this reason we have set a limit to the size of packet that we de-crypt and therefore some apps will no longer show feedback,

TextToSiri – You must have Apple TVos 14.5 and have turned on "Settings / Accessibility/ Text To Siri" to make use of this function..

This signal brings up Siri function on the Apple TV and can be used to send serial text which can then trigger any Siri command for the Apple TV. Please check the example program, to see how to set up sending a Siri Command within the program.

See link below for commands that Siri can execute.

<https://support.apple.com/en-us/HT205300>

HID_Press_Release – This allows you to send any HID message to the Apple TV. The HID messages supported by Apple TV are undocumented, but we have found that – for example – keyboard keys will jump to content in menus. In combination with HID_Page# and HID_Key# signals, setting the signal to HIGH will "PRESS" the HID key, and setting it LOW will "RELEASE" the HID key. Long pulses act as press and hold. For a list of HID

commands, please refer to:

https://www.usb.org/sites/default/files/documents/hut1_12v2.pdf

We would be interested to hear any exciting HID commands you may discover!

KeyboardText\$ – When the Apple TV is in a state where it can accept text entry (Output state KeyboardActive is high) 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, every time the text is changed the string will be sent to the Apple TV and every few seconds the KeyboardFBText\$ will be updated with what the Apple TV has currently got in the text field (See example Program and UI.)

KeyboardEnter – digital signal is use as a 'Search' button While in keyboard state.

KeyboardClear – While in KeyboardActive state this signal will clear the text in the Apple TV text field (Which will be then returned back to KeyboardFBText\$.)

KeyboardBack – This is an alternative to backspace, normal backspace on the touch panel keyboard will resend the text string with one character less, while KeyboardBack will send a backspace command to the Apple TV, then the new Apple TV Text String will be sent back to the Processor with KeyboardFBText\$.

GestureActive – Required for gesture control to be implemented; this indicates whether the gesture canvas is being used (finger is on and ether moving or not). GestureActive pulsed acts as a select, held high acts as a select and hold and held high with GestureX# and/or GestureY# changing is movement (Please see example program and touch panel for more information).

GestureX#, GestureY# – Required for gesture control to be implemented; the signal requires a full analog range (0 – 65535) from the 'X/Y Touch Feedback Analog Join' on the gesture canvas (Please see example program and touch panel for more information).

SetPosition# – New to version v2.01, This allows you to set the position of the playbar, with analog between 0 – 65535 where 0 being 0:00 and 65535 being full duration of the currently playing media.

Notes on Gesture Control

The module itself is capable of controlling the Apple TV with touch events in a very similar way to the Apple Bluetooth remote. Unfortunately, this behaviour is very difficult to reproduce with the current Crestron UI elements for two reasons:

- a) The Gesture Canvas presents the component updates to the control system discretely. i.e. The gesture position values (x, y) are not valid at the time when the "touch" event goes high. This leads to erratic behaviour as the x, y values "jump" to catch up.
- b) Positional updates are throttled to limit the load on the processor but this has the impact that the movement becomes very jerky.

We have implemented slight delays, and a smoothing algorithm to mitigate these issues, but the result is that the gesture control may appear "sluggish". This aspect of the implementation will hopefully improve as Crestron's UI technology develops.

Outputs

Most signals are self-explanatory, so we will only detail those that require special mention.

Connected_Fb – This signal will go high, when the system has successfully received a TCP Connection with the Apple TV. NOTE: This does not mean that the Apple TV is paired and ready to be used it simply indicates whether the processor has an open connection with an Apple TV. This DOES indicate successful Bonjour device discovery.

Connected_Status# - This analog signal mirrors the TCP status of the SIMPL TCP client symbol and can be used in conjunction with the Connected_Fb signal.

Verified – Indicates whether the module has successfully verified with the Apple TV. If this signal is high then this indicates that the Apple TV module has successfully paired and verified with the atvconfig.json file resident on the processor. This therefore means that the Apple TV setup was correct and the module has control of the specified Apple TV.

KeyboardActive – Unlike our previous Apple TV IP control module, this new version will be notified by the Apple TV when it is in keyboard mode through the KeyboardActive signal (see below), hence why there is no longer an "UpdateKeyboard" or "UpdateStatus" polling input signal. This output signal should be used to activate a touch panel text entry (see touch panel example).

Duration# – Returns the total length of the currently playing film, song or program expressed in seconds.

ElapsedTime# – Indicates how far through, in seconds, that particular piece of media is.

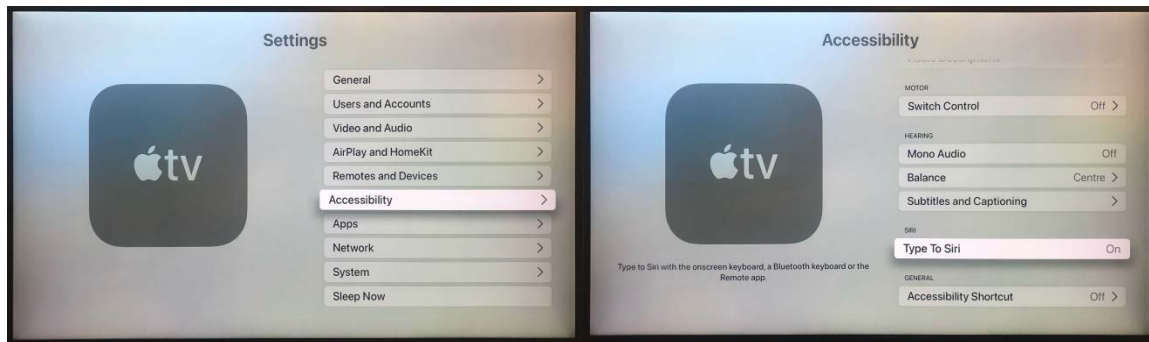
Position# – Analog signal returns full range (0 – 65535), this value has been pre calculated to indicate how far through the particular playing media is.

KeyboardFBText\$ – Returns periodically (every few seconds) the current text string on the Apple TVs search box, this is information that the Apple TV sends the processor after every change to its search box text.

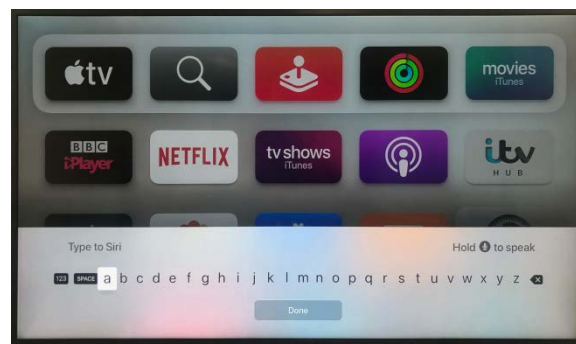
LAUNCHING APPS FROM CRESTRON

With the release of TVOs 14.5, the Apple TV now allows a keyboard function to appear when using the Apple TV Siri function. Our module enables you to open the Siri function and insert a Command, which then make the Apple TV jump to an app form anywhere within the UI.

For this feature to work you need to turn on "Type to Siri" To do this go to Setting/Accessibility/TypeToSiri with in the Apple TV UI.]



Within the example program you can find a Stepper and a Serial IO, which shows the process needed to open the Siri Feature, send the Siri command and then trigger the done button.



There are many other commands which Siri can control and can be found in the link below.

<https://support.apple.com/en-gb/guide/tv/atvbb96a3e3f/tvos>

PROGRAMMING NOTES – CRESTRON STUDIO & CRESTRON HOME

We are currently working on making the module compatible with Crestron Certified Drivers which will allow for integration with Studio, and the Crestron Home OS3 system for quick, easy and reliable programming for small projects. This will hopefully be ready for release in the near future.

MODULE ELEMENTS

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

- 🌟 Apple TV Client (Ultamation) v2.42.usp
- 🌟 MediaRemoteClientV2.clz
- 🌟 AppleTV IP Control Module Datasheet (Ultamation) v2.42.pdf (this file)

REVISION HISTORY

- 🌟 2.44 – Added signal to bring up Siri function on the Apple TV serial text
- 🌟 2.42 – Fixed crash on re-connection, which would require a progress/reboot to re-connect
- 🌟 2.42 – Set a limit to the size of feedback packets, the module would de-crypt and read. Large feedback packets causing Creston processor to slow down and reach 100% CPU load.
- 🌟 2.40 – Added manual connection reset capability, fixed fatal crash on connection recovery
- 🌟 2.32 – Reworked connection automatic recovery logic
- 🌟 2.30 – BETA version of re-worked connection logic
- 🌟 2.20 – Initial release of Database 200 version (same as 2.12)
- 🌟 2.12 – Improved stability with large messages and added feedback indicators