



Ulyssix Command and Control API

Revision B 12/05/2022

Ulyssix Technologies, Inc
7470 New Technology Way, Suite B, Frederick, MD 21703-9461
Tel: 301-846-4800 ~ Fax: 301-846-0686 ~ www.ulyssix.com

Table of Contents

1. Introduction	4
2. License.....	4
3. Ulyssix Discovery Protocol	4
4. TCP/IP Connection	5
5. Syntax.....	5
6. Error Codes	6
7. Special Commands	7
7.1. Comments.....	7
7.2. Help.....	7
7.3. Version	7
8. Special Options.....	8
8.1. Get.....	8
8.2. Verbose	8
9. Selecting a Ulyssix Card Index and Channel Number.....	8
9.1. Example of Selecting a Ulyssix Card Index and Channel Number.....	8
9.2. Script	9
9.3. Response	10
10. File Pulls and Transfers	11
11. API Command Set	12
11.1. Verbs	12
11.2. ALTAIR	13
11.1. Ulyssix Card	14
11.2. Archive	14
11.3. Bit Sync.....	15
11.4. FrameSync.....	16
11.5. SubFrameSync.....	17
11.6. Decom	18
11.7. Simulator.....	19
11.8. Simulator Frame Sync	19
11.9. Simulator SubFrame Sync	20
11.10. Receiver.....	21
11.11. Transmitter	22

11.1.	TAD UDP.....	22
12.	Client Applications	23
12.1.	PuTTY.....	23
12.2.	Python.....	25
12.2.1.	Python script to read and change the Bit Rate from Ulyssix Card 0	26
12.2.2.	Example Python script to open a Configuration File on the Server	29
12.2.3.	Example Python script using the Ulyssix Discovery Protocol.....	31
12.2.4.	Python Module UlxCmdCtrl.py	32
12.3.	Ulyssix Terminal in C#	34
12.3.1.	Discovery Protocol	35
12.3.2.	TCP Command and Control Connection	35
12.3.3.	Commands	35
12.3.1.	Response	35

1. Introduction

This document describes the Ulyssix Command and Control API. The purpose of this API is to configure, control, and retrieve data from Ulyssix telemetry cards inside of a Ulyssix Altair or Ulyssix Gryphon system. The computer with the Ulyssix cards is referred to as “Remote Computer.” The computer controlling the Remote Computer is referred to as the “Control Computer.”

Each Ulyssix card can be either single or dual channel. Each system can contain one or more cards. And each system has one Command and Control interface. Therefore, to do a simple task like set the Bit Sync’s Bit Rate, the user must go through a series of tasks:

1. Connect to the Ulyssix System.
2. Determine the number of Ulyssix cards in the system and the number of channels in each card.
3. Select the desired Ulyssix card.
4. Select the desired channel on the Ulyssix card.
5. Set the Bit Sync’s Bit Rate.

This document includes the syntax of the API, examples of command structure, and description of the API Command Set.

2. License

The Ulyssix Command and Control Software Development Kit (SDK) includes example Python scripts and a Python Module, `UlxCmdCtrl.py`. These Python scripts are licensed via the MIT License. The Ulyssix Command and Control SDK contains the file “License.txt,” which contains the follow license text:

```
MIT License Copyright (c) 2021 Ulyssix Technologies, Inc. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE
```

3. Ulyssix Discovery Protocol

The Ulyssix Discovery Protocol (UIP) is based on a similar concept as Simple Service Discovery Protocol (SSDP) or Multicast DNS (mDNS). It is a simple protocol where a Ulyssix client broadcasts an UDP command on a specific port across the network and then wait for the response from the Ulyssix servers.

Ulyssix implemented their own protocol because of the numerous documented attack vectors for SSDP and mDNS. A vendor specific protocol is not ideal; however, it does have inherent security advances.

The UIP uses UDP packets on port 1900. The client sends an eight-byte packet that has the ASCII string “ULXIDQRT.” The server responds with a packet that begins with the ASCII string “ULXIDREP” and then contains data formatted in XML for the IP Address, TCP Port, and Ulyssix Cards in the system.

The XML follows the format below. The data is inside of the UlxID tag. Inside that tag, is the Name tag, IP tag, and the UlxCards tag. The Name tag is the name of the computer. Ulyssix sets the name of each computer to be type of system (Altair, Gryphon, etc.) and the serial number. The IP tag contains the IP address for the TCP/IP server. The UlxCards tag contains one or more UlxCard tags, which contains the Slot and Model of the Ulyssix card.

```
<UlxID>
  <Name>Altair0123</Name>
  <IP>192.168.0.10</IP>
  <UlxCards>
    <UlxCard>
      <Slot>0</Slot>
      <Model>232</Model>
    </UlxCard>
    <UlxCard>
      <Slot>1</Slot>
      <Model>202</Model>
    </UlxCard>
  </UlxCards>
</UlxID>
```

Parsing the XML gives the client a list of possible server IP addresses to connect.

4. TCP/IP Connection

The ALTAIR and Gryphon software packages include a TCP/IP Server for use with the Ulyssix Command and Control API. The TCP/IP Server allows multiple connections. Connecting to the TCP/IP server requires knowing the IP Address of the server. The IP Address can be acquired through the user interface on the Ulyssix software on the server or using the Ulyssix Identification Protocol.

The TCP/IP Server is on Port 10611. This is one higher than the port used for the IRIG106-22 Chapter 6 Recorder and Reproducer Command and Control.

Each TCP/IP command should be terminated with the End of Line sequence. The Ulyssix Command and Control API uses the standard Windows OS End of Line sequence of Carriage Return and Line Feed (ASCII hex values 0x0D 0x0A). The response from the TCP/IP server is terminated with an asterisk followed by the End of Line sequence.

5. Syntax

The Ulyssix Command and Control API uses a Unix command line-based syntax. The basic structure of a command is a Verb followed by one or more Options. Each Option can have a Parameter. A Parameter is a value such as a number or a string that is a setting for the Option. An Option has two a Long Name and a Short Name. The Long Name is a string that begins with two dashes "--". The Short Name is a single character that follows a single dash "-".

The Parameters have five types: Integer, Double, String, Enum, or Boolean. These types limit the allowed range and precision of the entry. Strings can optionally be enclosed in either single or double quotes. Enclosing a string in quotes is required when the string contains a dash. This differentiates a dash used for an Option from a dash from a Parameter string. The allowed values for Enum are listed in this document as well as by using the Verb --Help Option. For Boolean Options, including the Options in the

command sets the Boolean to true. Excluding the Option from the Command sets the Boolean to false. You do not use the keywords “True” or “False” in Boolean Options.

Example 1: `BitSync --BitRate 1000000`

The command in Example 1 will set the Bit Sync Bit Rate to 1Mbps. In the command above, BitSync is the verb. The Options BitRate is in the Long Name format. And the Parameter is 1000000

Example 2: `BitSync --b 1000000`

The command in Example 2 has the same functionality of the command in Example1, except that it uses the Short Name for the BitRate Option.

Example 3: `BitSync --BitRate 1000000 --LoopBandwidth 0.1`

The command in Example 3 is an example of a Verb with two Options. The command sets both the Bit Rate and LoopBandwidth for the BitSync.

Syntax Rules:

1. The API is not case sensitive. However, this document will use capitalization for readability.
2. The API use the standard Windows OS End of Line sequence of Carriage Return and Line Feed (ASCII hex values 0x0D 0x0A).
3. Strings can be enclosed with single quotes or double quotes. Strings that include a dash must be in enclosed in single or double quotes so that the parser knows the difference between a string with dashes and a Verb’s Option.
 - a. Optional single quotes ‘My String’.
 - b. Optional double quotes “My String”.
 - c. Required quotes “My-String-With-Dases”.
4. Options have two names.
 - a. The Long Name is a string and must be prepended with double dashes.
 - b. The Short Name is a single character and must be prepended with a single dash.
5. The data for file transfers are enclosed in the File Transfer Tags. This allows the parser to know when the file starts and ends. Please note that file transfers can be complicated because Windows saves many text-based files as Unicode (UTF-16).
 - a. File Transfer Start Tag <filetransfer>.
 - b. File Transfer End tag </filetransfer>.
6. Responses terminate with an asterisk followed by the End of Line sequence.

6. Error Codes

The API has two types of error codes. The first kind of error code is an error code for a syntax error. These errors are generated before the command is parsed.

UnknownOptionError – The Option is not recognized. This could be due to a data entry error.

BadFormatConversionError – The conversion of the command to arguments failed. This could be due to an extraneous character, including non-visible ASCII or Unicode characters.

The second kind of error code is when a command does not parse correctly. The Error Codes follow a set format. They begin with a capital 'E', followed by a space, and then a two-digit number. The error codes are derived from IRIG106-22 Chapter 6 Recorder and Reproducer Command and Control Section 6.2.2 Command Error Codes. Additional codes are added to handle situations that are not applicable to the hardware described in IRIG106-22 Chapter 6.

E 00 – Invalid Command
E 01 – Invalid Range
E 02 – Invalid Mode
E 03 – No Drive
E 04 – Drive Full
E 05 – Command Failed
E 06 – Busy

The error codes provide limited user feedback. For more useful feedback, please use the Verbose Option described in Section 8.2 Verbose.

7. Special Commands

The API includes some special use case commands that warrant discussion outside of the API Command Set section.

7.1. Comments

The API includes two types of comments: REM and COM. Any text between comment command and an End of Line is ignored. REM is an abbreviation for Remove. The REM comment is never echoed back; its purpose to help document scripts. COM is an abbreviation for Comment. The COM comment can be echoed back as part of the response. The comment commands do not require dashes. The COM and REM commands are not case sensitive, but this document will show them as capitalized to follow the standard formatting from other scripting languages.

Example 1: `REM The following code sets the Bit Rate.`

Example 2: `COM The following code sets the Frame Sync Pattern.`

7.2. Help

The Help command returns a documentation on a command based on its usage. Help is used like an Option, but it only had a Long Name. The string "--help" can be used on its own to get a list of the verbs. Or it can be used as an Option after a verb.

The command "--help" returns the list of Verbs.

The command "BitSync -help" will return the list of Options for the Verb BitSync.

7.3. Version

The Version command displays the version of ALTAIR and the copyright information. The Version information is included at the top of the response for the Help Command. Version is used like an Option, but it only has a long name. The proper usage is "--Version".

8. Special Options

8.1. Get

Get is an Option inside of some of the Verbs that returns the current value of another Option. The Get Options requires a Value of a string, where the string is another Option without any dashes. The value for the Get Option can be the long name or short name of the desired Option. The Short Name for the Get Option is -g.

Example 1: `BitSync --Get BitRate`

Example 2: `BitSync -g BitRate`

Example 2: `BitSync -g b`

The three examples above will return the current value of the Option BitRate inside of the Verb BitSync.

8.2. Verbose

Verbose is an Option inside every Verb. The Verbose Option requests a more detailed response. The typical response is a simple asterisk or an error code. The Verbose Option is useful when developing and debugging a series of commands. The Short Name for the Verbose Option is -v.

9. Selecting a Ulyssix Card Index and Channel Number

Each Ulyssix System will have one or more Ulyssix cards. Each Ulyssix card will have one or two channels depending on the card model. A channel is a data path through the hardware. For example, a Ulyssix Tarsus3-02 Dual Bit Sync, Frame Sync, and Decom card has two channels. Before changing a setting, like the Bit Sync's Bit Rate, you must select the Ulyssix Card Index and the Channel Number using the Verb UlyssixCard and its associated Options.

9.1. Example of Selecting a Ulyssix Card Index and Channel Number

The first step is to list the Ulyssix cards in the system. Use the Verb "UlyssixCard" and the Option "--List". The response will be the Ulyssix Card Index followed by the description of the card. In the example below, there are two Ulyssix BaldEagleRF-02 dual channel cards in the system. By definition, a dual channel card has two channels.

```
Command:    UlyssixCard --List
Response:   0 Bald Eagle RF Dual Channel
            1 Bald Eagle RF Dual Channel
            *
```

The second step is to select the Ulyssix card. Use the Verb "UlyssixCard" and the Option "--Index" to select the desired Ulyssix card. The Verbose Option gets a detailed response.

```
Command:    UlyssixCard --Index 0 -v
Response:   Ulyssix Card Index 0 selected.
            *
```

The third step is to select the Channel in the Ulyssix card. Use the Verb "UlyssixCard" and Option "--Channel" to select the desired Channel. The Verbose Option gets a detailed response.

```
Command:    UlyssixCard --Channel 0 -v
```


Response: Ulyssix Card Channel 0 selected.
*

The final step is to verify the selected Ulyssix card and Channel. Use the Verb “UlyssixCard” and the Option “--Selection” to get the current selected Ulyssix Card Index and Channel. The Verbose Option is not needed.

Command: UlyssixCard --Selection
Response: Index 0 Channel 0
*

After selecting a Ulyssix Card Index and Channel Number, you can use commands for polling current settings, changing settings, and polling status.

To get the current Bit Sync Status for the selected Ulyssix Card Index and Channel, use the Verb “BitSync” and the Option “--Status”. The response includes the Measured Bit Rate (Bits per Second), the Signal Strength (Percent), and Bit Sync Lock Status (Lock or Loss). Status responses are delimited with a semicolon between the name and the value.

Command: BitSync --Status
Response: BitRate:20000000.0 bps
SigStrength:100.0 %
BitSync:Lock
*

To get the current Bit Sync Bit Rate Setting for the selected Ulyssix Card Index and Channel Use the Verb “BitSync”, the Option “--Get”, and the Value “BitRate”. The response is the current setting using engineering units (Mbps, kbps, or bps). Get responses are delimited with an equal sign between the name and the value.

Command: BitSync --Get BitRate
Response: BitRate=20Mbps
*

To set the Bit Sync Bit Rate Setting for the selected Ulyssix Card Index and Channel Use the Verb “BitSync”, the Option “--BitRate”, and the Value of the new Bit Rate settings in units of Bits per Second. The Verbose Option gets a detailed response.

Command: BitSync --BitRate 1000000 -v
Response: BitRate=10Mbps
*

The previous command changed the Bit Sync Bit Rate setting in the System but did not download the setting to the selected Ulyssix card. To download the Bit Sync Settings to the Ulyssix card, use the Verb “BitSync” and the Option “--Download”. The Verbose Option gets a detailed response.

Command: BitSync --Download -v
Response: Download Bit Sync 0 in Ulyssix Card 0
*

9.2. Script

Below is a script that includes the above commands and comments:

```
COM Turn on echo
Altair -e on

COM Select the card and index then verify selection
UlyssixCard --List
UlyssixCard --Index 0 -v
UlyssixCard --Channel 0 -v
UlyssixCard --Selection

COM BitSync Status
BitSync --Status

COM Get Bit Sync Settings
BitSync --Get BitRate

COM Set Bit Sync Settings
BitSync --BitRate 10000000 -v

COM Download Bit Sync Settings
BitSync --Download -v
```

9.3. Response

Below is the response of the script:

```
COM Turn on echo
Altair -e on
*
COM Select the card
UlyssixCard --List
0 Bald Eagle RF Dual Channel
*
UlyssixCard --Index 0 -v
Ulyssix Card Index 0 selected.
*
UlyssixCard --Channel 0 -v
Ulyssix Card Channel 0 selected.
*
UlyssixCard --Selection
Index 0 Channel 0
*
COM BitSync Status
BitSync --Status
BitRate:20000000.0 bps
SigStrength:100.0 %
BitSync:Lock
*
COM Get Bit Sync Settings
BitSync --Get BitRate
BitRate=10Mbps
*
COM Set Bit Sync Settings
BitSync --BitRate 10000000 -v
Set BitRate=10Mbps
*
COM Download Bit Sync Settings
BitSync --Download -v
Download Bit Sync 0 in Ulyssix Card 0
*
```

10. File Pulls and Transfers

The Ulyssix Command and Control API includes the functionality to transfer configuration files and archive files. The file transfer begins with the Start File Transfer Tag <filetransfer>, then the data in the file encoded as raw binary (UTF-8), and finally the End File Transfer Tag </filetransfer>. The File Transfer Tags are required for the receiving computer to identify the end of the file.

The term “Pull” is used when moving a file from the Remote Computer to the Control Computer. The term “Transfer” is used when moving a file from the Control Computer to the Remote Computer.

The Ulyssix Command and Control API includes the Option Checksum for the Verb Altair and the Verb Archive. The SHA256 algorithm is used for the checksum. Please note that the encoding used for the SHA256 is Unicode (UTF-16). The response to the Checksum command is a 32-byte number in hexadecimal with dashes between each byte. The response terminates with an asterisk and the End of Line sequence.

Example Checksum: 75-09-79-26-68-7E-93-FF-48-50-44-25-39-22-DC-FA-AD-D7-43-47-0C-1B-60-51-99-7C-D1-3F-EE-D0-C0-C9*

Process for pulling a file:

1. Determine the file to pull.
2. Send the command to get the SHA256 Checksum for the file.
3. Send the Pull command for the file name inside double quotes.
 - a. Altair --Pull “MyFile.xml”
4. Wait for the server to respond with the file data enclosed in the file transfer tags.
5. Save the file to the local drive.
6. Calculate the SHA256 Checksum for the local file.
7. Compare the checksums to evaluate the success of the Pull command.

Process for Transferring a file:

1. Determine the file to transfer.
2. Calculate the SHA256 Checksum for the file using Unicode (UTF-16) encoding.
3. Send the Transfer command with the file name inside double quotes.
 - a. Altair --Transfer “MyFile.xml”
4. Wait for the response from the server and verify the message.
 - a. * End Of Line
5. Send Start File Transfer Tag, the file data encoded as Unicode (UTF-16), and the End File Transfer Tag.
6. Wait for the response from the server and verify the message.
 - a. </filetransfer>* End of line
7. Send the command to get the checksum.
 - a. Altair --Checksum MyFile.xml
8. Wait for the response from the server with the SHA256 Checksum.
9. Compare the checksums to evaluate the success of the Transfer.

11. API Command Set

The Ulyssix Command and Control API Command set is generated by using the Special Command --Help. Each section contains the response from the Help command. The response begins with the Version information. After the Version, there is a table with commands. For the Verbs, the table has two columns. The Verb name is in the first column and the description in the second column. All other tables have three columns. The Option Short Name is in the first column, the Option Long Name is the second column, and the description in the third column. After the table are a list of notes with information about the commands.

For all the sections other than the Verbs, the description column includes at least two lines. The first line is the description of the Option. The second line defines the Value. Some Options do not use Values; they are labeled as “No parameter.” Other Options have Values defined as Strings, Integers, Doubles, or Enums. An Option with a Value of type Enum includes the allowed Enum values inside of curly brackets. For an example, see the CodeType Option in the BitSync Verb.

Please note that not all Verbs are used on all Ulyssix cards. The Tarsus3 card does not have a Receiver or a Transmitter and therefore the response for using those Verbs will be an error. If the Verb includes the Verbose Option, then the response will include text like “Selected Ulyssix card does not have a receiver.”

11.1. Verbs

The Help for Verbs lists of the Verb commands as well as the two comments (COM and REM), and the commands for Help and Version.

```
--help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.

--Altair           Configure the ALTAIR Software.
--Archive          Configure the ALTAIR Archive.
--UlyssixCard      Configure the Ulyssix Card settings.
--BitSync          Configure the Bit Sync settings.
--FrameSync        Configure the Frame Sync settings.
--SubFrameSync     Configure the SubFrame Sync settings.
--Simulator        Configure the Simulator settings.
--SimFrameSync     Configure the Simulator Frame Sync settings.
--SimSubFrameSync Configure the SubFrame Sync settings.
--Receiver         Configure the Receiver settings.
--Transmitter      Configure the Transmitter settings.
--TadUdp           Configure the UDP TAD settings.
--COM              Comment that is echoed to the return text. COM echo
                  occurs even if Echo is set to Off.
--REM              Comment that is never in the returned text. REM is
                  ignored even if Echo is set to On.
--help            Display this help screen.
--version         Display version information.
```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.2. ALTAIR

The Help for Altair lists the Options for the Altair Verb. The Altair Options include commands for opening, saving, and transferring configuration files as well as system level Options like Delay and Echo.

Transferring a file from the Remote Computer to the Control Computer is the Option Pull. Transferring a file from the Control Computer to the Remote Computer is the Option Transfer. The Altair Verb includes the Option Checksum to calculate the SHA256 Checksum for a specified file on the Remote Computer. After the transfer, the SHA256 Checksum should be calculated on the received file and the two checksums compared to verify a successful file transfer.

```
Altair --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

```
-v, --Verbose          Return a verbose response from the command followed by
                        an end of line.
                        No parameter.
-a, --SaveAs           Save As ALTAIR Configuration File with user specified
                        ALTAIR Configuration File name on the remote computer. File
                        name can be in single quotes or double quotes.
                        String.
-c, --CurrentConfig    Returns current ALTAIR Configuration file name.
                        No parameter.
-d, --Delay            Delay script by user specified time in milliseconds.
                        Integer.
-e, --Echo             Echo command back with responses.
                        String Enum. {On, Off}.
-k, --Checksum         Checksum SHA256 calculation user specified ALTAIR
                        Configuration File. Specify file name or use shortcut
                        'Current' for the current ALTAIR Configuration File.
                        File name can be in single quotes or double quotes.
                        String.
-l, --Log              Log the commands and responses to a text file.
                        String Enum. {On, Off}.
-n, --New              Create new ALTAIR Configuration File on the remote
                        computer.
                        No parameter.
-o, --Open             Open user specified ALTAIR Configuration File on the
                        remote computer. File name can be in single quotes or
                        double quotes.
                        String.
-p, --Pull             Pull user specified ALTAIR Configuration File from
                        remote computer. Specify file name or use shortcut
                        'Current' for the current ALTAIR Configuration File.
                        File name can be in single quotes or double quotes.
                        String.
-s, --Save             Save ALTAIR Configuration File on the remote computer.
                        No Parameter.
-t, --Transfer         Configure transfer of user specified ALTAIR
                        Configuration File to the remote computer. File name
                        can be in single quotes or double quotes.
                        String.
-x, --ConfigFiles     List ALTAIR Configuration Files on the remote computer.
                        No parameter.
--help                Display this help screen.
--version              Display version information.
```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.

3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.1. Ulyssix Card

The Help for UlyssixCard lists the Options for the UlyssixCard Verb. The UlyssixCard Options include commands for selecting and displaying the Ulyssix Card Index and the Ulyssix Card Channel as well as setting the Archive File.

Setting the Archive File might seem out of place in the UlyssixCard verb. It is located here instead the Archive because the Ulyssix Card Verb addresses a specific Ulyssix Card Index and Channel, while the Archive Verb applies to the entire system.

```
UlyssixCard --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

```
-v, --Verbose           Return a verbose response from the command followed by
                        an end of line.
                        No parameter.
-a, --CurrentArchive   Get current Archive File Name for the selected Ulyssix
                        Index and Channel.
                        No parameter.
-c, --Channel          Select Channel Index for selected Ulyssix Card.
                        Integer. Zero based
-d, --Description     Get the description of the selected Ulyssix Card.
                        No parameter.
-e, --EnableArchive   Enable Archive for selected Ulyssix Index and Channel.
                        String Enum. {On, Off}
-f, --ArchiveFile     Set Archive File Name for Ulyssix Index and Channel. File
                        name can be in single quotes or double quotes.
                        String.
-i, --Index           Select the Ulyssix Card Index.
                        Integer. Zero based.
-l, --List            List the Ulyssix Cards in the computer.
                        No parameter.
-s, --Selection       Return the current Ulyssix Card Index and Channel
                        Number.
                        No parameter.
--help               Display this help screen.
--version            Display version information.
```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.2. Archive

The Help for Archive lists the Options for the Archive Verb. The Archive Options include commands for configuring the archive file, starting archive recording, stopping archive recording, and transferring archive files.

Please note that setting the Archive File Name for a specific Ulyssix Card Index and Channel is in the UlyssixCard Verb. It is located there instead the Archive because the Ulyssix Card Verb addresses a specific Ulyssix Card Index and Channel, while the Archive Verb applies to the entire system.

Transferring a file from the Remote Computer to the Control Computer is the Option Pull. Transferring a file from the Control Computer to the Remote Computer is the Option Transfer. The Altair Verb includes the Option Checksum to calculate the SHA256 Checksum for a specified file on the Remote Computer. After the transfer, the SHA256 Checksum should be calculated on the received file and the two checksums compared to verify a successful file transfer.

```
Archive --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-a, --ArchiveFiles	List Archive Files. No parameter.
-c, --CurrentFiles	List of currently enabled Archive Files. No parameter.
-k, --Checksum	Checksum SHA256 calculation for the user specified Archive file. File name can be in single quotes or double quotes. String.
-r, --ArchvieRec	Start Archive Recording. No parameter.
-p, --Pull	Pull user specified Archive file from remote computer. File name can be in single quotes or double quotes. String.
-s, --ArchiveStop	Stop Archive Recording. No parameter.
-t, --Transfer	Configure transfer of user specified Archive File to the remote computer. File name can be in single quotes or double quotes. String.
--help	Display this help screen.
--version	Display version information.

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.3. Bit Sync

The Help for BitSync lists the Options for the BitSync Verb. The BitSync Options include commands for configuring the Bit Sync Settings, getting the current Bit Sync Settings, downloading the Bit Sync Settings to the Ulyssix Card, polling the Bit Sync Status, and configuring the transferring Eye Pattern data via UDP.

```
BitSync --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-g, --Get	Get the value for a current setting for a specified option. Use the either the short name (without dash) or long name (without dashes). String Enum. {BitRate, EyePattern, CodeType, Input, LoopBandwidth, Polarity}

```

-b, --BitRate          Set the Bit Rate.
                        Integer.
-c, --CodeType         Set the code type for the Bit Sync.
                        String Enum. {NRZ-L, NRZ-M, NRZ-S, Bi-L, Bi-M, Bi-S,
                        RNRZ(11)-F, RNRZ(15)-F}
-d, --Download         Download Bit Sync Settings to the selected Ulyssix Card
                        Index and Channel.
                        No parameter.
-e, --EyePattern       Enable Eye Pattern Data via UDP.
                        String Enum. {On, Off}.
-i, --Input            Set the input source for the Bit Sync.
                        String Enum. {Input BNC, Input Diff., Input RX}.
-l, --LoopBandwidth   Set the Loop Bandwidth.
                        Double.
-p, --Polarity         Set Polarity for the Bit Sync.
                        String Enum. {Normal, Inverted}
-t, --Status           Get the Bit Sync Status.
                        No parameter.
--help                Display this help screen.
--version              Display version information.

```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.4. FrameSync

The Help for FrameSync lists the Options for the FrameSync Verb. The FrameSync Options include commands for configuring the Frame Sync Settings, getting the current Frame Sync Settings, downloading the Frame Sync Settings to the Ulyssix Card, and polling the Frame Sync Status. The settings for the SubFrame Sync are in the SubFrameSync Verb.

```

FrameSync --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.

```

```

-v, --Verbose          Return a verbose response from the command followed
                        by an end of line.
                        No parameter.
-g, --Get              Get the value for a current setting for a specified
                        option. Use the either the short name (without
                        dash) or long name (without dashes).
                        String Enum. {BitsPerMinorFrame, SyncErrors,
                        NumMinorFrames, DataInSearchMode, NumSyncBits,
                        SyncPattern, BitSlips}
-b, --BitsPerMinorFrame Set the number of Bits in the Minor Frame (includes
                        the Frame Sync Pattern).
                        Integer.
-d, --Download         Download Frame Sync Settings to the selected
                        Ulyssix Card and Channel.
                        No parameter.
-e, --SyncErrors       Set the number of Sync Errors allowed.
                        Integer.
-f, --NumMinorFrames   Set the number of Minor Frames.
                        Integer.
-m, --DataInSearchMode Set Data in Search Mod
                        String Enum. {On, Off}
-n, --NumSyncBits      Set the number of Minor Frames.
                        Integer.
-p, --SyncPattern       Set Frame Sync Pattern in Hex.

```


	String Hex. Example: FE6B2840.
-s, --BitSlips	Set the number of Bit Slips. Integer.
-t, --Status	Get the Frame Sync Status. No parameter.
--help	Display this help screen.
--version	Display version information.

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.5. SubFrameSync

The Help for SubFrameSync lists the Options for the SubFrameSync Verb. The SubFrameSync Options include commands for configuring the SubFrame Sync Settings, getting the current SubFrame Sync Settings, downloading the SubFrame Sync Settings to the Ulyssix Card, and polling the SubFrame Sync Status. The settings for the Frame Sync are in the FrameSync Verb.

```
SubFrameSync --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-g, --Get	Get the value for a current setting for a specified option. Use the either the short name (without dash) or long name (without dashes). String Enum. {BitOrder, SfidEnd, BitsToLSB, NumSfidBits, SubFrameEnable, SfidStart}
-b, --BitOrder	Set the SFID Bit Order. String Enum. {MSB, LSB}
-d, --Download	Download Frame Sync Settings to the selected Ulyssix Card Index and Channel. No parameter.
-e, --SfidEnd	Set the end value for the SubFrame ID Counter. Integer.
-l, --BitsToLSB	Set the number of bits from the end of the Frame Sync Pattern to the LSB of the SFID. Integer.
-n, --NumSfidBits	Set the number of bits in the SFID. Integer.
-o, --SubFrameEnable	Turn the SubFrame Sync On or Off. String Enum. {On, Off}
-s, --SfidStart	Set the start value for the SubFrame ID Counter. Integer.
-t, --Status	Get the SubFrame Sync Status. No parameter.
--help	Display this help screen.
--version	Display version information.

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.6. Decom

The Help for Decom lists the Options for the Decom Verb. The Decom Options include commands for configuring the Decom Frame Size, configuring the Variable Bits per Word Table, downloading the Decom Settings to the Ulyssix Card.

Setting or reading the Variable Bits per Word Table is accomplished by using the SetWordNumber Options with either the ReadBitsForWordNumber Option or the SetBitsForWordNum Option in the same line.

Example 1: `Decom -SetWordNumber 1 -SetBitsForWordNum 10`

Example 2: `Decom -SetWordNumber 1 -ReadBitsForWordNum`

In Example 1, the number of bits for Word 1 is set to 10-bits in the Variable Bits Per Word Table. In Example 2, the number of bits in Word 1 is returned as “Word 2 size is 10-bits.”

```
Decom --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

```
-v, --Verbose          Return a verbose response from the command
                        followed by an end of line.
                        No parameter.
-g, --Get              Get the value for a current setting for a
                        specified option. Use the either the short name
                        (without dash) or long name (without dashes).
                        String Enum. {CommonWordSize, StreamName,
                        WordsPerMinorFrame}
-c, --CommonWordSize  Set the Common Word Size for the frame.
                        Integer.
-d, --Download        Download Decom Settings to the selected Ulyssix
                        Card Index and Channel.
                        No parameter.
-f, --WordsPerMinorFrame
                        Set the Words Per Minor Frame.
                        Integer.
-n, --StreamName      Name for the PCM Stream. Stream name can be in
                        single quotes or double quotes.
                        String.
-r, --ReadBitsForWordNum
                        Read the number of bits for the specified Word
                        Number for the Variable Bits Per Word Table.
                        No Parameter.
-s, --SetBitsForWordNum
                        Set the number of Bits for the selected Word
                        Number for Variable Bits Per Word Table. Will
                        return error if the Word Number is not selected.
                        Integer.
-w, --SetWordNumber  Set the Word Number for uses with setting the
                        number bits in the Variable Bits Per Word Table.
                        Used with either ReadBitsForWordNum or
                        SetBitsForWordNum.
                        Integer.
--help                Display this help screen.
--version              Display version information.
```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.7. Simulator

The Help for the Simulator lists the Options for the Simulator Verb. The Simulator Options include commands for configuring the Simulator including the PCM Output and Simulator Data source. The other settings required to configure the Simulator are in the verbs SimFrameSync and SimSubFrameSync.

The PCM Output Options include the Simulator Enable, Bit Rate, Code Type, and Polarity. The Simulator data source include the Archive Mode (Major Frame Simulator or Archive File Playback) and the Archive File Name.

```
Simulator --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.

-v, --Verbose          Return a verbose response from the command followed by an
                        end of line.
                        No parameter.
-g, --Get              Get the value for a current setting for a specified
                        option. Use the either the short name (without dash) or
                        long name (without dashes).
                        String Enum. {ArchiveMode, BitRate, CodeType, Enable,
                        ArchiveFile, Polarity}
-a, --ArchiveMode     Archive Mode On plays a TAD file. Archive Mode Off plays
                        major frame simulator.
                        String Enum. {On, Off}
-b, --BitRate         Set the Simulator Bit Rate.
                        Integer.
-c, --CodeType        Set the code type for the Simulator.
                        String Enum. {NRZ-L, NRZ-M, NRZ-S, Bi-L, Bi-M, Bi-S,
                        RNRZ(11)-F, RNRZ(15)-F}
-d, --Download        Download Frame Sync Settings to the selected Ulyssix Card
                        Index and Channel.
                        No parameter.
-e, --Enable          Set Simulator Enable.
                        String Enum. {On, Off}
-f, --ArchiveFile     Set Archive File with file extension of .TAD. File name can
                        be in single quotes or double quotes.
                        String.
-p, --Polarity        Set Simulator Polarity.
                        String Enum. {Normal, Inverted}
--help                Display this help screen.
--version             Display version information.
```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.8. Simulator Frame Sync

The Help for the Simulator Frame Sync lists the Options for the SimFrameSync Verb. The SimFrameSync Options include commands for configuring the Simulator Frame.

```
SimFrameSync --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-g, --Get	Get the value for a current setting for a specified option. Use the either the short name (without dash) or long name (without dashes). String Enum. {CommonBitsPerWord, NumMinorFrames, NumSyncBits, SyncPattern, UnusedWordValue, WordsPerMinorFrame}
-c, --CommonBitsPerWord	Set the number of Common Bits per Word. Integer.
-d, --Download	Download Frame Sync Settings to the selected Ulyssix Card and Channel. No parameter.
-f, --NumMinorFrames	Set the number of Minor Frames. Integer.
-n, --NumSyncBits	Set the number of Minor Frames. Integer.
-p, --SyncPattern	Set Frame Sync Pattern in Hex. String Hex. Example: FE6B2840.
-u, --UnusedWordValue	Set value for the Unused Word. String Hex. Example: 5555.
-w, --WordsPerMinorFrame	Set the number of Words per Minor Frame. Integer.
--help	Display this help screen.
--version	Display version information.

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.9. Simulator SubFrame Sync

The Help for the Simulator SubFrame Sync lists the Options for the SimSubFrameSync Verb. The SimSubFrameSync Options include commands for configuring the Simulator SubFrame.

```
SimSubFrameSync --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.
```

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-g, --Get	Get the value for a current setting for a specified option. Use the either the short name (without dash) or long name (without dashes). String Enum. {SfidEnd, NumSfidBits, SubFrameEnable, SfidStart, WordNumber}
-d, --Download	Download Frame Sync Settings to the selected Ulyssix Card Index and Channel. No parameter.
-e, --SfidEnd	Set the end value for the SubFrame ID Counter. Integer.
-n, --NumSfidBits	Set the number of bits in the SFID. Integer.
-o, --SubFrameEnable	Turn the SubFrame Sync On or Off. String Enum. {On, Off}
-s, --SfidStart	Set the start value for the SubFrame ID Counter. Integer.
-w, --WordNumber	Set the word number for the SFID.

```

Integer.
--help          Display this help screen.
--version       Display version information.

```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.10. Receiver

The Help for Receiver lists the Options for the Receiver Verb. The Receiver Options include commands for configuring the Receiver Settings, getting the current Receiver Settings, downloading the Receiver Settings to the Ulyssix Card, polling the Receiver Status, and configuring the transferring RF Waveform data via UDP.

```

Receiver --help
ALTAIR v22.25
Copyright (c) 2022 Ulyssix Technologies, Inc.

```

```

-v, --Verbose          Return a verbose response from the command followed
                        by an end of line.
                        No parameter.
-g, --Get              Get the value for a current setting for a specified
                        option. Use the either the short name (without
                        dash) or long name (without dashes).
                        String Enum. {Bands, BitRate, DiversityCombiner,
                        RfFrequency, IfBandwidth, Modulation, OutputFilter,
                        Source}
-a, --AutoCalc         Auto calculate the IF Bandwidth and Output Filter
                        based on the bit rate and modulation.
                        No parameter.
-b, --BitRate          Set the Receiver Bit Rate.
                        Integer.
-c, --DiversityCombiner Turn the Diversity Combiner On or Off. Only for
                        Receiver Channel 0.
                        String Enum. {Best Source, Optimal Ratio, Off}
-f, --RfFrequency     Set RF Frequency in MHz.
                        Double.
-i, --IfBandwidth     Set the IF Bandwidth in kHz.
                        Double.
-m, --Modulation       Set the Receiver Modulation
                        String Enum. {FM, SQPSK, BPSK, QPSK, GMSK}
-o, --OutputFilter     Set the Output Filter Deviation Ratio.
                        String Enum. {Auto DR1, Auto DR2, Auto DR3, Off}
-d, --Download         Download Frame Sync Settings to the selected
                        Ulyssix Card Index and Channel.
                        No parameter.
-s, --Source           Set the Receiver Waveform Source.
                        String Enum. {RF Spectrum, DemodOutFiltered}
-t, --Status           Get the SubFrame Sync Status.
                        No parameter.
-w, --Waveform         Enable Receiver Waveform data via UDP.
                        String Enum. {On, Off}
--help                Display this help screen.
--version              Display version information.

```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.

4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.11. Transmitter

The Help for Transmitter lists the Options for the Transmitter Verb. The Transmitter Options include commands for configuring the Transmitter Settings, getting the current Transmitter Settings, and downloading the Transmitter Settings to the Ulyssix Card.

Transmitter --help

ALTAIR v22.25

Copyright (c) 2022 Ulyssix Technologies, Inc.

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-g, --Get	Get the value for a current setting for a specified option. Use the either the short name (without dash) or long name (without dashes). String Enum. {Bands, BitRate, Enable, RfFrequency, Modulation, Source, Power}
-b, --BitRate	Set the Transmitter Bit Rate. Integer.
-e, --Enable	Turn on the Transmitter. String Enum. {On, Off}
-f, --RfFrequency	Set Transmitter RF Frequency in MHz. Double.
-m, --Modulation	Set the Transmitter Modulation String Enum. {FM, SOQPSK, BPSK, QPSK, GMSK, CW}
-s, --Source	Set the Transmitter Source. String Enum. {Freq Translator, Simulator, TTL Input}
-p, --Power	Set Transmitter RF Power in dBm with resolution of 0.25 dBm. The max setting is 0dBm. The min setting is -110dBm. Double.
-d, --Download	Download Frame Sync Settings to the selected Ulyssix Card Index and Channel. No parameter.
--help	Display this help screen.
--version	Display version information.

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

11.1. TAD UDP

The Help for TadUdp lists the Options for the TadUdp Verb. The TadUdp Options include commands for enabling the TadUdp and getting the current Transmitter Settings.

TadUdp --help

ALTAIR v22.25

Copyright (c) 2022 Ulyssix Technologies, Inc.

-v, --Verbose	Return a verbose response from the command followed by an end of line. No parameter.
-g, --Get	Get the value for a current setting for a specified option. Use the either the short name (without dash) or long name (without dashes). String Enum. {Enable}
-e, --Enable	Enable the TAD via UDP.

```
String Enum. {On, Off}
--help          Display this help screen.
--version       Display version information.
```

NOTES:

1. All Verbs Commands and Option Commands are case insensitive.
2. End of line is carriage return then new line.
3. File names optionally enclosed in single quotes or double quotes.
4. File names with dashes must be enclosed in single quotes or double quotes.

*

12. Client Applications

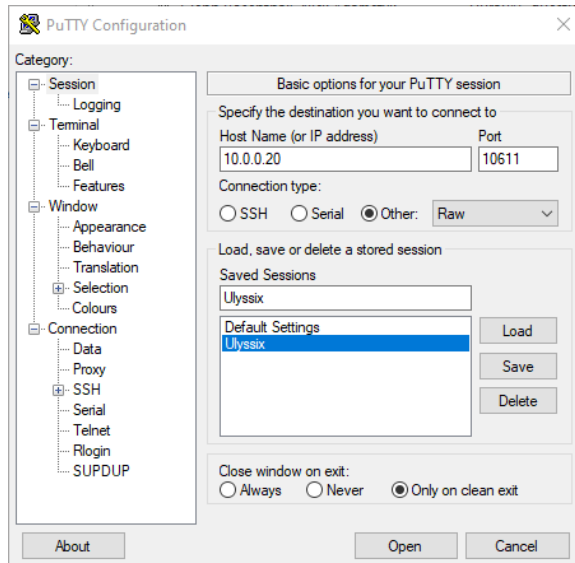
The Ulyssix Command and Control Server processes incoming data as UTF-8 characters. There are many ways to make a TCP connection to the Ulyssix Command and Control Server. A simple terminal application, like PuTTY (<https://www.putty.org/>), sends characters encoded as UTF-8 characters. Python scripting language includes a simple TCP client library that will communicate with the Ulyssix Command and Control Server. Other programming languages that support TCP clients (C++, C#, etc.) will also work, however the difficulty of implanting a TCP client varies.

This document includes directions for how to use PuTTY and example Python scripts. The Ulyssix Command and Control SDK includes additional Python scripts and a Python module that includes custom functions. Examples in other languages are not available, however functions like those found in UlxCmdCtrl.py Python Module can be implemented in any language given sufficient knowledge of the TCP client.

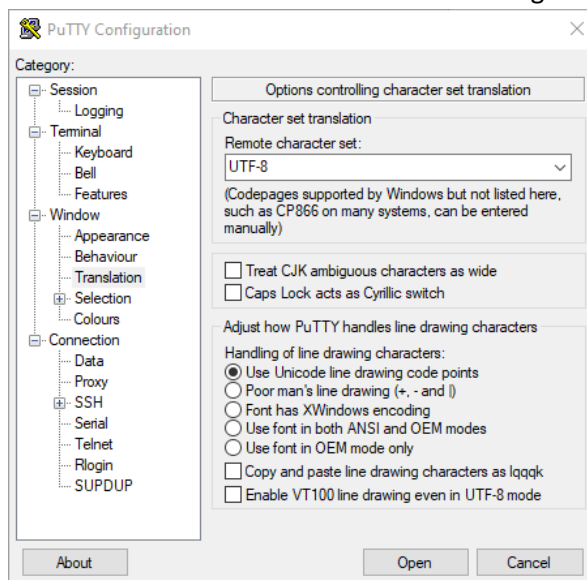
12.1. PuTTY

PuTTY is a free and open-source terminal application for Windows and other operating systems. It has many features, but only the raw binary TCP client configuration is discussed here. To connect to the Ulyssix Command and Control Server, you must know the IP address of the server. PuTTY include the Plink command line tool to run a script.

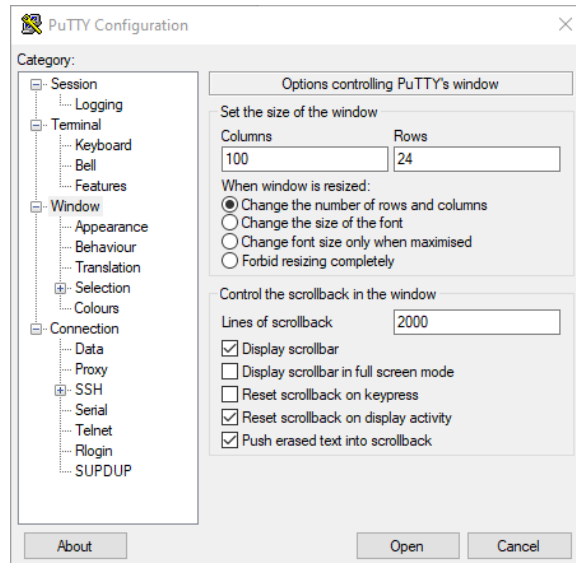
1. Launch PuTTY. The PuTTY Configuration window appears.
2. In the PuTTY Configuration window, select Session from the list of Categories on the left side of the window. In the Session settings:



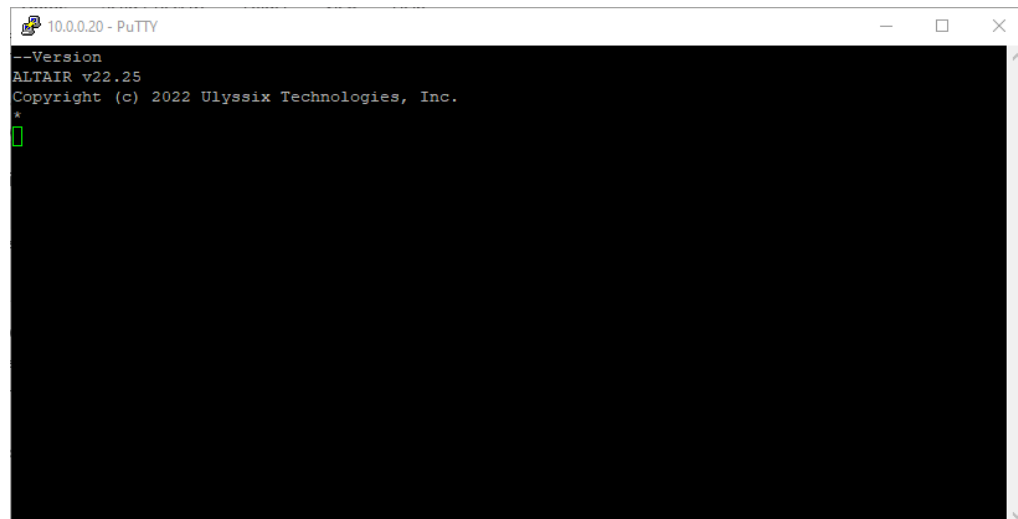
- a. Enter the IP address into the Host Name entry box.
 - b. Enter the 10611 into the Port entry box.
 - c. Set the Connection Type to Other and then select Raw from the drop-down menu.
3. In the PuTTY Configuration window, select Window \ Translation from the list of Categories on the left side of the window. In the Translation settings:



- a. Ensure that the Remote Character set drop-down menu is set to UTF-8.
4. Optional. In the PuTTY Configuration window, select Window from the list of Categories on the left side of the window. In the Window settings:



- a. Set the Columns to 100 (default is 80).
5. In the PuTTY Configuration window, select Session from the list of Categories on the left side of the window. In the Session settings:
 - a. Enter a name into the Saved Sessions and then click the Save button to store the settings.
6. Click the Open button at the bottom right of the window to open the connection.
7. Enter the command `--Version` and press the enter key to send the Version command. The response should be:



12.2. Python

Python is a scripting language with many high-level libraries and online support. Python includes a TCP library named Socket. Python is a good choice for a simple TCP script because the TCP socket implementation is simple. The examples below are not exhaustive Python scripts. Due to the power of Python and its libraries, the possibilities for advanced scripts are endless.

The Python scripts examples in this document were written and tested with Python 3.11.0 in Windows 10. They may work in other versions of Python 3.x; however, the scripts contain commands that will not

work in Python 2.x. The example scripts use commands for file paths that are Windows OS specific. There are also some networking commands used with Sockets that are Windows OS specific.

The example Python scripts below include a hard coded IP Address and Port. In your example, you will need to change the IP Address variable to the current IP Address of your Ulyssix Command and Control Server.

The example scripts include a function, `SendCmd()`, that takes an input command string and sends it via the TCP Socket. The function manages adding the End of Line (EOL) string and the options to display the command and the response. The function returns the response string for processing. The `SendCmd()` function includes two Boolean parameters. The first, `echo`, prints the command to the screen. The second, `resp`, prints the response from the server to the screen. These parameters are useful in getting feedback and debugging scripts.

In the example Python scripts, the code inside of the `WITH Socket` statement runs while the TCP socket is connected to the server. The Python Input command waits for the user to hit the Enter key before continuing. Using the Input command as the last line inside of the `WITH Socket` ensures that the TCP socket stays connected while waiting for the server to respond to the last command. This is important for slow commands like opening a Configuration File, saving a Configuration File, creating a new Configuration File, starting an Archive, stopping an Archive, or transferring a file.

The Ulyssix Command and Control SDK includes the Python script examples included in this document as well as examples for transferring files, calculating SHA255 checksums, and calculating the number of bit errors between two SHA256 checksums. These examples are based on a series of Python functions written by Ulyssix. These functions are included in a Python module named `UlxCmdCtrl.py`. This module is a `.PY` file and is saved as plain text. The `UlxCmdCtrl.py` module is included in a Python `.PY` file using the command: `import UlxCmdCtrl`. To use a function included in the library, you must prepend the module name and a period. For example, the aforementioned `SendCmd()` function is called using: `UlxCmdCtrl.SendCmd()`.

12.2.1. Python script to read and change the Bit Rate from Ulyssix Card 0

The first example Python script sends a series of commands to the Ulyssix Command and Control Server to select Channel 0 in Ulyssix Card 0 and then read the current Bit Rate, get the Bit Sync Status, change the Bit Rate, download the Bit Sync Settings, and then read the new Bit Rate. After the code, a screen shot of the IDLE window shows the results of the commands.

Please note that after you change settings that you must use the Download option to send those settings to the Ulyssix Card. A Download option sends all the settings for that Verb to the card. For example, a script can change multiple Bit Sync setting and then call the Download option once to download all the setting once.

This example Python script uses the Verbose option (`-v`) in several commands. The Verbose option is useful because it provides more feedback from a command. For example, the command `UlyssixCard – Index 0` without the Verbose option simply returns an asterisk followed the end of line string.

```
1. # Socket.py
2. # Written in Python 3.11.0
```

```

3. # in dir: C:\Users\Wade\AppData\Local\Programs\Python\Python311
4.
5. #Command to run script from Python IDLE
6. #exec(open('SocketBitRate.py').read())
7.
8. import socket          #for TCP Socket
9.
10. #IP Address and Port of the Ulyssix CmdCtrl Server
11. HOST = '10.0.0.20'
12. PORT = 10611
13.
14. #Function to take a command, add EOL, send socket, wait for response up to 8192 char,
    and print results
15. #Parameters:
16. #   s:   Socket - already connected
17. #   cmd: String - command without EOL
18. #   echo: Boolean - true to echo the command
19. #   resp: Boolean - true to print the response
20. #Return:
21. #   the TCP response string
22. def SendCmd(s, cmd, echo, resp):
23.     #Add the EOL string to the command
24.     cmd = cmd + '\r\n'
25.
26.     #Send the command via the TCP socket
27.     #The UTF-8 Encoding is each character is an 8-bit number
28.     s.sendall(cmd.encode('utf-8'))
29.
30.     #Wait for a response up to 8192 characters
31.     data = s.recv(8192)
32.
33.     #echo the command back
34.     if(echo):
35.         print(f'Sent:\r\n' + cmd)
36.
37.     #Print the response.
38.     #The UTF-8 Encoding is each character is an 8-bit number
39.     if (resp):
40.         print(f'Received:\r\n' + data.decode('Utf-8'))
41.
42.     return data.decode('Utf-8')
43.
44.
45. #Open the socket and send the command
46. with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
47.     #Connect to the Ulyssix CmdCtrl Server
48.     s.connect((HOST, PORT))
49.
50.     #Send a comment
51.     SendCmd(s, 'COM Example Code', False, True)
52.
53.     #Select the Ulyssix Card Index - use the Verbose flag for detailed response
54.     SendCmd(s, 'UlyssixCard --Index 0 -v', True, True)
55.     #Select the Ulyssix Card Channel - use the Verbose flag for detailed response
56.     SendCmd(s, 'UlyssixCard --Channel 0 -v', True, True)
57.     #Print the selected UlyssixCard Index and Channel
58.     SendCmd(s, 'UlyssixCard --Selection', True, True)
59.
60.     #Get the current Bit Sync BitRate setting
61.     SendCmd(s, 'BitSync --Get BitRate', True, True)
62.

```

```
63. #Get the BitSync Status
64. SendCmd(s, 'BitSync --Status', True, True)
65.
66. #Set the Bit Rate setting in bps - use the Verbose flag for detailed response
67. #NOTE: Use the Download command to download all BitSync settings
68. SendCmd(s, 'BitSync --BitRate 1230000 -v', True, True)
69. #Download the BitSync settings - use the Verbose flag for detailed response
70. SendCmd(s, 'BitSync --Download -v', True, True)
71.
72. #Get the current Bit Sync BitRate setting
73. SendCmd(s, 'BitSync --Get BitRate', True, True)
74.
75. #Wait until the user presses Enter to close the connection
76. input("Press Enter to close connection...")
```

IDLE result from Python script to get the Bit Rate from Ulyssix Card 0:

```
*IDLE Shell 3.11.0*
File Edit Shell Debug Options Window Help
>>> exec(open('SocketBitRate.py').read())
Received:
COM Example Code

Sent:
UlyssixCard --Index 0 -v

Received:
Ulyssix Card Index 0 selected.
*

Sent:
UlyssixCard --Channel 0 -v

Received:
Ulyssix Card Channel 0 selected.
*

Sent:
UlyssixCard --Selection

Received:
Index 0 Channel 0
*

Sent:
BitSync --Get BitRate

Received:
BitRate=1Mbps
*

Sent:
BitSync --Status

Received:
BitRate:1000000.0 bps
SigStrength:100.0 %
BitSync:Lock
*

Sent:
BitSync --BitRate 1230000 -v

Received:
Set BitRate=1.23Mbps
*

Sent:
BitSync --Download -v

Received:
Download Bit Sync 0 in Ulyssix Card 0
*

Sent:
BitSync --Get BitRate

Received:
BitRate=1.23Mbps
*

Press Enter to close connection...

Ln: 607 Col: 0
```

12.2.2. Example Python script to open a Configuration File on the Server

The second example Python script sends the command to get the list of Configuration Files on the Ulyssix Command and Control Server. The list of files is a single string that is delimited by the EOL string Carriage Return and New Line. The script then checks if the desired Configuration File name is in the list of Configuration Files. If the Configuration File is found, the scrip sends the file to open the file. If the file

is not found, then the string "File Not Found." is displayed. Please note that the Server can take up to 30 seconds to open a Configuration File. The Python TCP client will not receive the response string until the Server completes the file open operation.

The code inside of the WITH Socket statement runs while the TCP socket is connected to the server. As previously mentioned, the --Altair Open command can take up to 30 seconds to execute. The Python Input command waits for the user to hit the Enter key before continuing. Using the Input command as the last line inside of the WITH Socket ensures that the TCP socket stays connected while waiting for the server to respond to the --Altair Open command.

```
1. # SocketSelectFile.py
2. # in dir: C:\Users\Wade\AppData\Local\Programs\Python\Python311
3.
4. #Command to run script from Python IDLE
5. #exec(open('SocketSelectFile.py').read())
6.
7.
8. import socket      #for TCP Socket
9.
10.
11. #IP Address and Port of the Ulyssix CmdCtrl Server
12. HOST = "10.0.0.20"
13. PORT = 10611
14.
15.
16. #Function to take a command, add EOL, send socket, wait for response up to 8192 char,
    and print results
17. #Parameters:
18. #   s:   Socket - already connected
19. #   cmd: String - command without EOL
20. #   echo: Boolean - true to echo the command
21. #   resp: Boolean - true to print the response
22. #Return:
23. #   the TCP response string
24. def SendCmd(s, cmd, echo, resp):
25.     #Add the EOL string to the command
26.     cmd = cmd + '\r\n'
27.
28.     #Send the command via the TCP socket
29.     #The UTF-8 Encoding is each character is an 8-bit number
30.     s.sendall(cmd.encode('utf-8'))
31.
32.     #Wait for a response up to 8192 characters
33.     data = s.recv(8192)
34.
35.     #echo the command back
36.     if(echo):
37.         print(f'Sent:\r\n' + cmd)
38.
39.     #Print the response.
40.     #The UTF-8 Encoding is each character is an 8-bit number
41.     if (resp):
42.         print(f'Received:\r\n' + data.decode('Utf-8'))
43.
44.     return data.decode('Utf-8')
45.
```

```

46.
47. #Define the file that want to open
48. configFile = 'LiveDemo.xml'
49.
50.
51. #Open the socket and send the command
52. with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
53.     #Connect to the Ulyssix CmdCtrl Server
54.     s.connect((HOST, PORT))
55.
56.     #Get list of files
57.     files = SendCmd(s, 'Altair --ConfigFiles', False, False)
58.
59.     #If the list of file includes the desired file name then open the file
60.     if (files.find(configFile) > 0):
61.         SendCmd(s, 'Altair --Open ' + configFile + ' -v', True, True)
62.     else:
63.         print(f'File not found.')
64.
65.     #Wait until the user presses Enter to close the connection
66.     input("Press Enter to close connection...")

```

IDLE result from Python script to check if a Configuration File exists on the server and open it:

```

IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> exec(open('SocketSelectFile.py').read())
Sent:
Altair --Open LiveDemo.xml -v

Received:
File Open: LiveDemo.xml

Press Enter to close connection...
>>>
Ln: 11 Col: 0

```

12.2.3. Example Python script using the Ulyssix Discovery Protocol

Included in the SDK is the file `UdpDiscovery.py`. It is an example of using the Ulyssix Discovery Protocol to find the Ulyssix Remote Servers on your network. The Ulyssix Discovery Protocol uses Multicast UDP packets. Configuring Python sockets for sending and receiving Multicast UDP packets more complicated than TCP packets. Due to the length and complication of the script, its text is not shown in this section.

The socket to send the Ulyssix Discovery Query requires the following settings:

1. Time To Live (TTL) is the number of routers that the packet is allowed to traverse before reaching its destination. This value should be set based on your network architecture. In the example, the TTL value is 32 and this value might be too high for some networks.
2. Reuse Address is a Boolean to determine if the address can be reused after it is closed. This value is set to true.

3. The socket binding is set to the user selected adapter. When the computer is connected to multiple networks, it is important to set the Adapter. Please note that the Loop Address (127.0.0.1) and any VPNs count as adapters.

Once configured, the socket sends a packet that contains the eight-character Ulyssix Query string: "ULXIDQRY." Python does not require you to define the encoding, but for sake of completion the encoding is UTF-8.

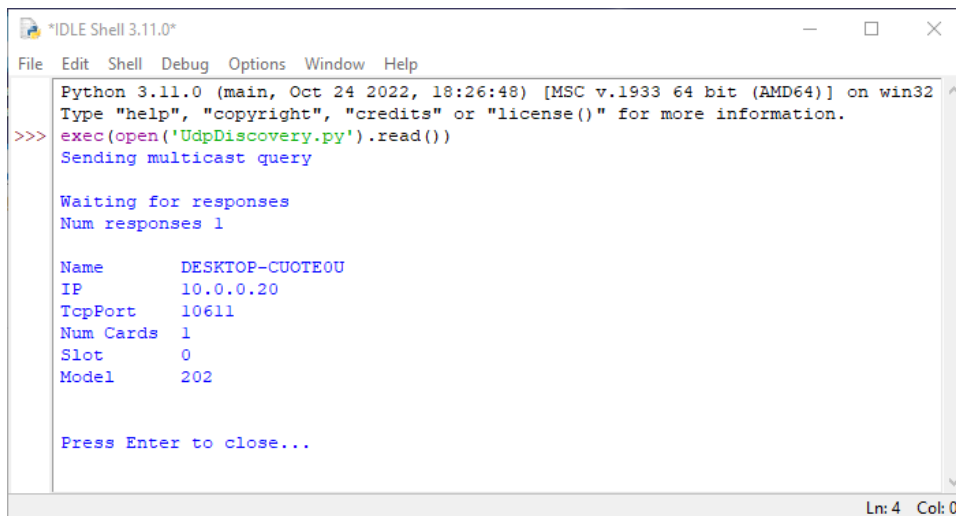
The socket to receive the Ulyssix Discover Query requires the following settings:

1. Reuse Address is a Boolean to determine if the address can be reused after it is closed. This value is set to true.
2. Multicast Membership subscribes the socket to the multicast address of the specified adapter.
3. The socket timeout is set to 250mS.

Once configured, the socket waits to receive packets. A Try / Except allows the socket to timeout without the error stopping the script. The socket receive function runs a fixed number of times defined by the variable count (set to 16 in the example). The socket receives data as a byte array, the script converts the byte array to a string. Then the string is compared to the Ulyssix Response string; the string must begin with the Ulyssix Response string to be a valid message. If the message is valid, it is added to a list of strings. Please note that if there are multiple Ulyssix Remote Servers on your network, then there should be multiple responses saved in list of strings.

After the socket processes its responses, the list of strings is processed. The processing begins by parsing each string into XML. Then the information in the XML is formatted and printed to the display.

Response from UdpDiscovery.py



```
*IDLE Shell 3.11.0*
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> exec(open('UdpDiscovery.py').read())
Sending multicast query

Waiting for responses
Num responses 1

Name      DESKTOP-CUOTE0U
IP        10.0.0.20
TcpPort   10611
Num Cards  1
Slot      0
Model     202

Press Enter to close...
Ln: 4 Col: 0
```

12.2.4. Python Module UlxCmdCtrl.py

This Python Module contains functions useful for writing Python scripts. The module includes the following functions:

1. **SendCmd (s, cmd, echo, resp)** – This function takes a user specified Ulyssix Command and Control command, formats it, and sends it via the connected TCP socket. The function returns the response from the server.
 - a. Parameters:
 - i. s: Socket – Connected TCP socket.
 - ii. cmd: String - Command without EOL.
 - iii. echo: Boolean - True to print the command to the window.
 - iv. resp: Boolean - True to print the response to the window.
 - b. Return: String - TCP response.
2. **TransferFile (s, filePath)** – This function takes the absolute path to a file and sends the file to the remote computer via the connected TCP socket.
 - a. Parameters:
 - i. s: Socket - Connected TCP socket.
 - ii. filePath: String - Absolute path to file to send to the remote computer.
 - b. Return: Null – Nothing is returned.
3. **PullFile (s, filePath)** – This function takes the absolute path to a file, uses the connected TCP socket to ask the server to send the file name, and then saves the file to the user specified file path. Note, the function extracts the file name from the file path parameter.
 - a. Parameters:
 - i. s: Socket - Connected TCP socket.
 - ii. filePath: String - Absolute path to file for the location to save the file. The file name is extracted from the file path and sent to the remote computer as the requested file to transfer.
 - b. Return: Null – Nothing is returned.
4. **CalcSha256 (filePath, resp)** – This function calculates the SHA256 checksum for specified absolute file path on the local computer. The checksum is formatted in as a hex string in the Ulyssix format: capital letters with dashes between bytes.
 - a. Parameters:
 - i. filePath: String - Absolute path to file to calculate the checksum.
 - ii. resp: Boolean - True to print the hex string to the window.
 - b. Return: String - The SHA256 check sum as a hex string in Ulyssix format (capitol letters and dashes between bytes).
5. **CompareHexString (one, two)** – This function calculates the number of bit differences between two SHA256 Checksums hex string in the Ulyssix format: capital letters with dashes between bytes.
 - a. Parameters:
 - i. one: String - Hex string in Ulyssix format (capital letters and dashes between bytes).
 - ii. two: String - Hex string in Ulyssix format (capital letters and dashes between bytes).
 - b. Return: Integer – Number of bit differences between the two hex strings. Value of zero for when the two hex strings are identical.

The Python scrip example files SocketFilePull.py and SocketFileTransfer.py in the SDK use the UlxCmdCtrl.py module. These two files are examples of transferring files between the local computer

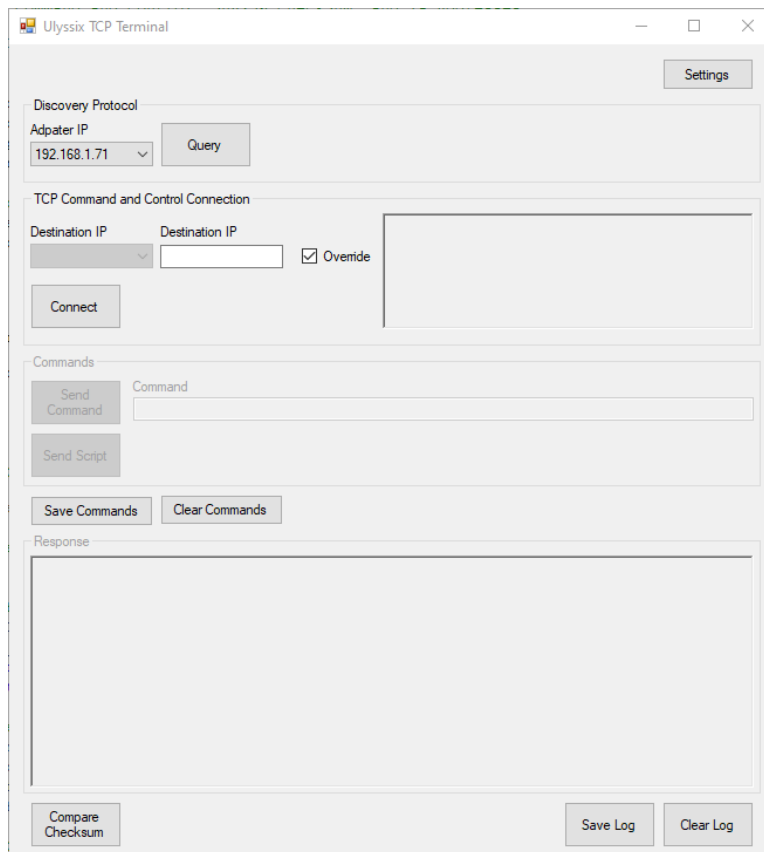
and the remote server. The best way to verify the success of the transfer is to compare the SHA256 checksum of the local file and the SHA256 checksum from the remote server.

The Altair verb and the Archive verb both contain the option—Checksum that returns the SHA256 checksum in the Ulyssix format hex string. The UlxCmdCtrl function CalcSha256() calculates the SHA256 in the Ulyssix format hex string checksum for a file on the local computer. The function ComareHexString() calculates the number bits that are different between the two input hex strings. The expected result for a successful file transfer is zero bit differences.

12.3. Ulyssix Terminal in C#

The Ulyssix Terminal is a C# Winforms application written in Visual Studio 2022 using .NET 4.8 Framework. The Ulyssix Terminal implements the Ulyssix Discovery Protocol and TCP Command and Control. The example C# project uses synchronous UDP sockets because they are simple to implement and understand. The tradeoff is that any network communication that takes longer than a few seconds will freeze the application until the communication finishes. Both the application and Microsoft Visual Studio solution are part of the SDK.

The graphical user interface (GUI) is divided into four sections: Discovery Protocol, TCP Command and Control Connection, Commands, and Response. The Settings button is in the upper right corner. It launches a window to edit the settings stored in an XML file. Settings include the local paths for Archive and Configuration files.

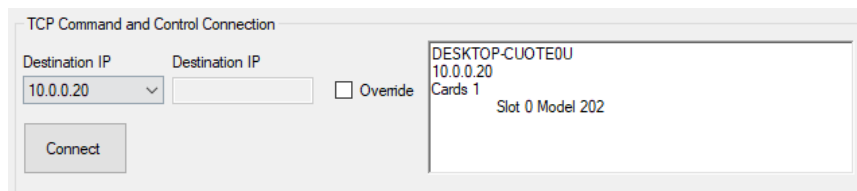


12.3.1. Discovery Protocol

The Discovery Protocol section has two controls: Adapter IP combo box and the Query button. First, select the desired Adapter IP Address from the combo box. This determines the adapter to send the query. Then click the Query button to send the Ulyssix Discovery Protocol query packet over the network. The query process takes about five seconds. When it completes, the Destination IP combo box in TCP Command and Control Connection section will populate with the IP Addresses for the Ulyssix Remote Servers on the network.

12.3.2. TCP Command and Control Connection

The TCP Command and Control Connection connects to the Ulyssix remote server specified by user selected the IP address. There are two ways to select the IP address. First, run the Discovery Protocol to populate the Destination IP combo box and then select the from the combo box. When the Destination IP combo box is changed, the text box displays information about the selected Ulyssix remote server. The information includes the server's name, IP address, and information about the Ulyssix cards in the server.



Second, click the Override check and enter the Ulyssix remote server IP address in the Destination IP textbox. The override method is required in cases when the network security blocks the multicast address used for the Ulyssix Discovery Protocol.

Finally, click the Connect button to connect to the selected Ulyssix remote server. Once connected, the Command and Response controls become enabled. When the system is connected, the Connect button turns blue. When disconnected, the Connect button is gray.

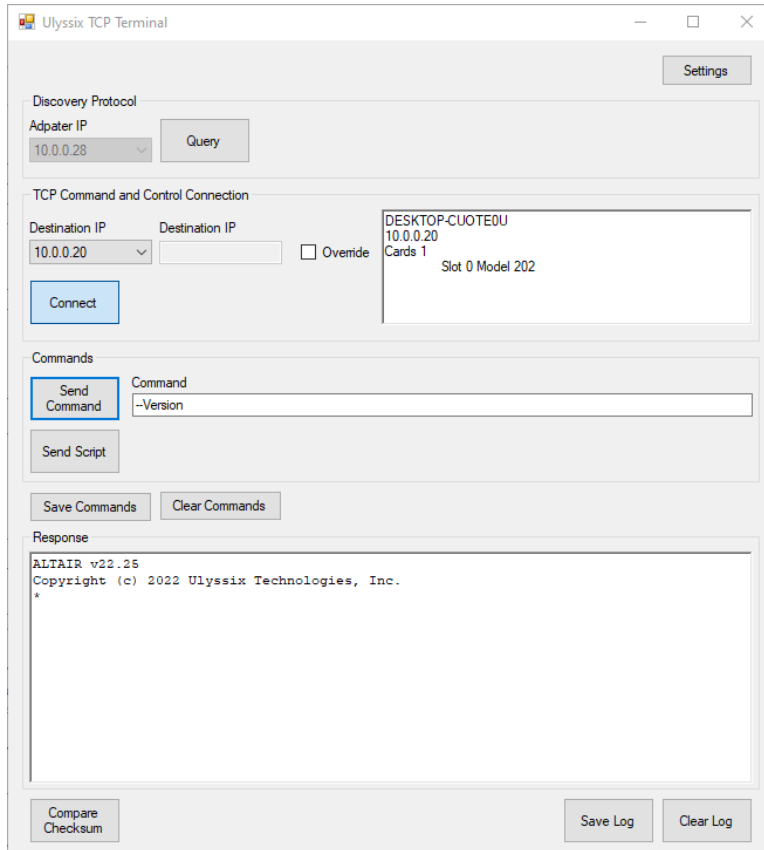
12.3.3. Commands

The Commands section sends commands to the Ulyssix remote server in two ways. First, type a command into the Command textbox and then click the Send Command button. Second, click the Send Script button and select a text file with multiple commands. The Ulyssix TCP Terminal will send each line in the file as a separate command. The server's response for each command appears in the Response section in the large textbox.

The Ulyssix TCP Terminal keeps a log of every command sent. The Save Commands button saves this log to a text file for future use as a script. The Clear Commands button clears the Command Log.

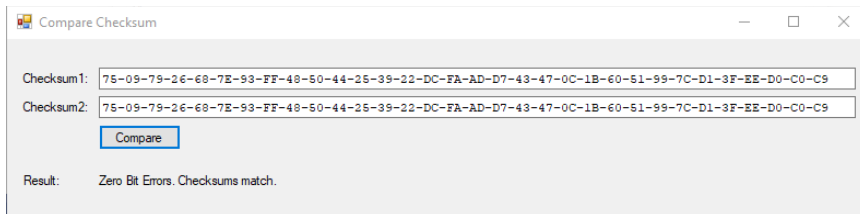
12.3.1. Response

The Response section includes a text box that logs the responses from the server. The Save Log button saves the responses to a text file. The Clear Log button clears the log.



The Compare Checksum button launches a calculator where the user enters two SHA256 checksums in Ulyssix format. The Compare Checksum calculator determines the number of bits that are different between the two checksums.

Zero Bit Errors



Two Bit Errors

