

# **CBIT System User Guide**

**Version 1.0**

**Ronak Malik  
Cypress Woods High School**

**August 2020**

---

# USER'S MANUAL

## TABLE OF CONTENTS

	<u>Page #</u>
<b>Table of Contents</b>	
General Information.....	3
Bounce Server.....	3
1.1 Installation.....	3
1.1.1 Requirements.....	3
1.1.2 Security.....	3
1.2 Usage.....	4
1.2.1 Command line arguments.....	4
Daemon Mode:.....	4
Max Pairs:.....	4
Max Connections:.....	4
1.2.2 Commands.....	5
Mill-Side Controller Program.....	5
1.1 Installation.....	5
1.1.1 Requirements.....	5
1.2 Usage.....	6
1.2.1 Settings.....	6
1.2.1.1 Connection Settings.....	6
1.2.1.2 Video Settings.....	7
1.2.1.2 LinuxCNC Settings.....	8
User Controller Program.....	8
1.2 Usage.....	9
1.2.1 Startup Screen.....	9
1.2.2 System Menus.....	10
1.2.2.1 File Menu.....	10
1.2.2.2 Connect Menu.....	12
1.2.3 Controlling the Mill.....	13
1.2.3.1 Full Screen Video:.....	13
1.2.4 Exit System.....	14
1.2.4 Running Multiple Clients.....	14
Author.....	14

---

## **GENERAL INFORMATION**

### **1.1 System Overview**

A control system for the Circuit Board Inspection Table (CBIT).

- Allows for remote, responsive control of a 4-axis mill
- Real-time viewing of video from an attached camera
- No port forwarding or firewall configuring necessary
- For technical and architectural details please refer to the CBIT Architecture document

## **BOUNCE SERVER**

### **1.1 Installation**

#### **1.1.1 Requirements**

The Bounce Server only requires a working version of Java 8 and JavaFX 8 or newer. If you have the Oracle version of Java installed, a working version of JavaFX is packaged with it.

The Bounce Server also needs permission to open, write to, and read from sockets.

#### **1.1.2 Security**

The Bounce Server purposefully has few implemented security features to allow for greater extensibility by the user.

Other than the authentication message to prevent unwanted connections, as of yet, the bounce server has no implemented encryption features, so all data is sent in plaintext. This is to allow the user to wrap the bounce server in any security features that are needed.

A good example of this is using an HTTP(S) proxy through [Apache](#). This allows the http server to handle SSL and wraps the data in an HTTP connection making the connection more friendly to some routers/networks.

A tutorial on how to setup an http proxy server with Apache (the given example uses ssh, but can be easily extended to forward to this bounce server) can be found here: <https://geek.co.il/2017/04/18/ssh-over-https-for-fame-profit>

The same tutorial also explains how to setup SSL using [letsencrypt](#)

### **1.2 Usage**

Once the necessary requirements have been met, a user can simply run the jar file provided to start the Bounce Server (either through a system service provider like systemd, or through the command line). If

---

started without a connected terminal (or input/output devices like when started as a service) it is recommended to start the Bounce Server with the "-d" option so it can continue to be monitored/controlled.

### 1.2.1 Command line arguments

Argument	Description	Requires additional parameter
-e	Echo data back to sending client	No
-d	Start in daemon mode	No
-p	Set bind port of bounce server	Yes, integer port to bind to
--max-connections	Set max number of connections allowed within a connection group	Yes, max connection integer
--max-pairs	Set the maximum number of connection groups allowed	Yes, max pairs integer
	Set initial verbosity. 1: Print connections and disconnections 2: Print state changes and full error stack traces 3: Print received packet information (size, type) 4: Print all bytes received	
-v -vv -vvv -vvvv		No

All arguments must be provided separately.

#### *Daemon Mode:*

Starting the bounce server in daemon mode will open a socket on port 32565. Accepted connections will then be presented with a command prompt similar to the one in the default interactive mode. All commands from interactive mode will work in this mode as well. The connection to the control socket can simply be closed to exit the command prompt, or the `exit` command may be given. Only one connection may be connected to the control socket at a time. This running mode is meant to allow this server to be run as a service, then allow a connection to monitor and control the state of the bounce server.

#### *Max Pairs:*

The maximum number of connection groups this server allows. By default, this number is set to 65536 connection groups. It is recommended to set this number to be a power of 2, as it directly affects the `n` in the version string sent to clients.

#### *Max Connections:*

The maximum number of connections allowed in a particular connection group. A connection will be disconnected with the message "Connection Group Full", if it tries to connect to an already filled connection group. Because of certain conditions causing a TCP TIME\_WAIT state, it may be recommended to set this number higher than the desired maximum number of connections. The default number is 2.

---

## 1.2.2 Commands

Command	Description	Parameters
print	Prints information about the current bounce server state	connections, channels
verbose	Sets the verbosity	1, 2, 3, or 4
echo	Sets whether this server should echo back messages to clients	on or off
kill	Closes the given connection	A connection id(found in print connections), or socket address
stop	Stop the bounce server	N/A
exit (daemon only)	Closes the current control connection, but continues listening	N/A
help	Print the help menu	N/A

These commands can be run while either connected to the control socket in daemon mode, or by typing them at the command prompt in the terminal.

## MILL-SIDE CONTROLLER PROGRAM

### 1.1 Installation

#### 1.1.1 Requirements

The mill computer (and optionally an additional single board computer) require a working version of Java 8 and JavaFX 8 or newer. If the Oracle version of Java is installed, a valid version of JavaFX is packaged with it.

The computer encoding and sending video will also require the base version of GStreamer, as well as the “good”, “bad”, and “ugly” plugin packages. It may also be recommended to install GStreamer hardware acceleration libraries such as VAAPI.

The mill computer requires a working version of LinuxCNC, and the "DISPLAY" setting in the .ini file should be set to "linuxcncrsh".

If you choose to use an external video computer separate from the mill computer, it also needs to have an open port on which it can receive connections from the local network. By default the program listens on port "32565".

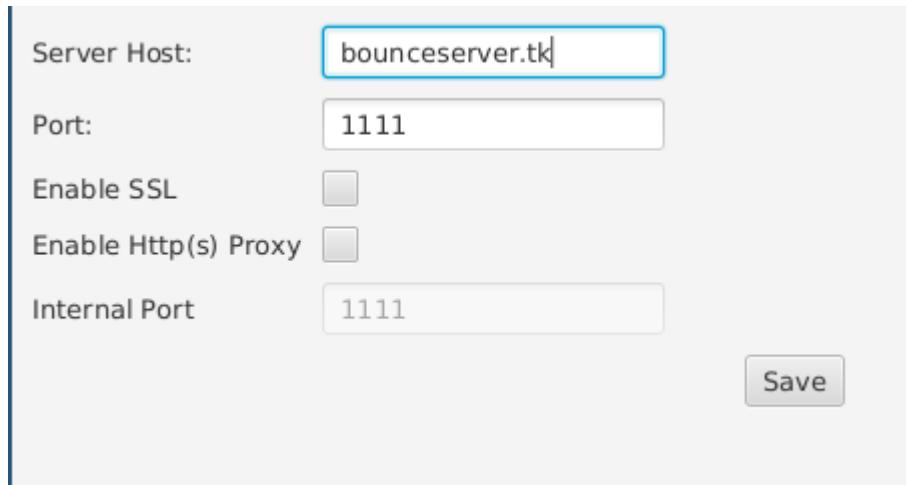
---

## 1.2 Usage

Both the mill program and the optional separated video program can both be run by simply running the jar file with the necessary requirements met. The video program is completely headless, so it either must be run from a terminal window or run as a service.

### 1.2.1 Settings

#### 1.2.1.1 *Connection Settings*



The screenshot shows a configuration dialog titled "Connection Settings". It contains the following fields:

Server Host:	bounceserver.tk
Port:	1111
Enable SSL	<input type="checkbox"/>
Enable Http(s) Proxy	<input type="checkbox"/>
Internal Port	1111

A "Save" button is located at the bottom right of the dialog.

In the connection settings menu, the user can change the settings the network client will use to connect to the bounce server.

The "Host" and "Port" field change the host address on the internet that this client will connect to (this can be an IP address or a hostname). These parameters should be the address and listening port # of the bounce server.

The "Enable SSL" checkbox turns on SSL capabilities currently based on OpenSSL 1.1. This ensures that transferred data is encrypted and secure. The bounce server defined in the "Host" and "Port" fields must support SSL if this checkbox is selected, otherwise the connection will fail. Enabling SSL can help hide the data this connection is transferring in case of snooping or router deep packet inspection feature blocking unknown protocols.

The "Enable Proxy" checkbox selects whether to wrap the data stream in an HTTP proxy. At the moment, this program only supports using a proxy to connect to a bounce server on the same computer (loopback adapter) as the proxying computer. Similar to SSL, the bounce server must support HTTP proxies, otherwise the connection will fail. Selecting this option also enables the "Internal Port" field, which must be filled out for the connection to succeed. Using an HTTP proxy can help mask the connection this program makes on particularly picky firewalls.

The "Internal Port" field defines the local port the bounce server is listening on that the proxy should forward this connection to.

### 1.2.1.2 Video Settings

<input checked="" type="radio"/> Default Device	
<input type="radio"/> Connected Webcam	
Device Path:	/dev/video0
<input checked="" type="radio"/> IP Camera	
IP Camera URL:	rtsp://192.168.2.88:554
<input type="radio"/> TCP Source	
TCP Source Host:	127.0.0.1
TCP Source Port:	
<input checked="" type="checkbox"/> Is the incoming video H.264 encoded?	
<input checked="" type="checkbox"/> Is the streaming device remote?	
Remote Streaming Host:	192.168.2.67
Remote Streaming Port:	32565
<input checked="" type="checkbox"/> Start video when client connects?	
<input type="button" value="Save"/>	

The mill side program gives the option to get video from several different sources.

**Note: If you choose to use an external computer for streaming video separate from the mill computer, the video configurations (device paths, IPs, etc.) are in reference to the video computer, NOT the mill computer, even though the configuration is done through the interface on the mill computer.**

The "Default Device" option tries to automatically find and select a video capture device attached to the computer, whether it be a USB camera, embedded camera, or other capture device that creates an entry in /dev/.

The "Connected Webcam" option allows the user to select the device path of the capture device.

The "IP Camera" option allows the user to specify the URI of a connected IP camera. The URI given must be a valid address. The “PTZ Optics” inspection camera included as part of the CBIT controller is such a source.

The "TCP Source" option tells the program to read from a specified TCP source. If the source does not exist, the program will timeout and display an error.

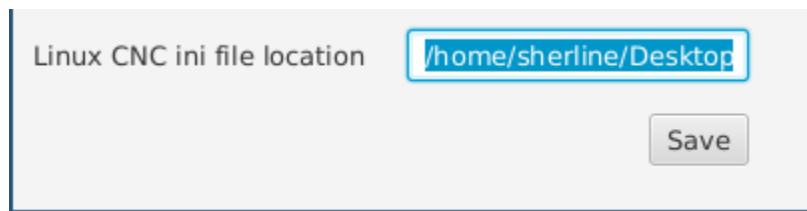
---

It is highly recommended to accurately select if the video being received is already H.264 encoded. This allows the program to simply repackage the video before sending it, saving on CPU power and decreasing video latency.

The next section, "Remote video device", allows the user to specify the location of the separate remote streaming computer. The video computer needs to have opened the specified port and have the video program listening on that port.

The last checkbox tells the program whether to start the video stream or tell the remote video computer to start the video stream when it receives a connection from the user controller program. If this checkbox is selected, then the user does not need to manually start the video in the Video menu.

#### 1.2.1.2      ***LinuxCNC Settings***



The user needs to tell the program the location of the LinuxCNC ini file so control of the mill can be successfully started. Once the location is saved, the user can start LinuxCNC by going to the "LinuxCNC Menu" > "Start LinuxCNC"

## **USER CONTROLLER PROGRAM**

### **1.1 Installation**

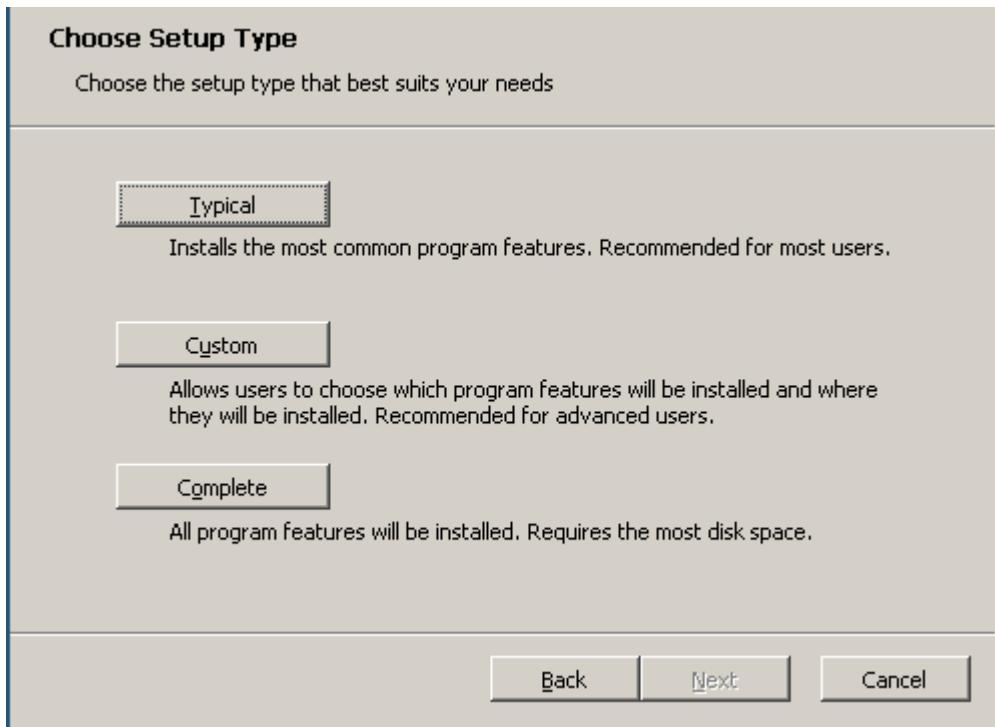
Version 1.0 of this program requires a complete version of GStreamer to be installed on the user's system for the program to function.

GStreamer can be downloaded here:

<https://gstreamer.freedesktop.org/download/>

Ensure that the version of GStreamer (32-bit or 64-bit) that is downloaded and installed matches the version of Java on the computer.

During GStreamer installation select the "Complete" option when prompted on this screen show below:



## 1.2 Usage

### 1.2.1 Startup Screen

The user is greeted by the main screen when the program starts as shown below.



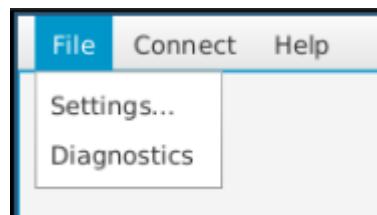
---

This is the main control screen with rectangles drawn on for ease of explanation. The blue rectangle encloses the button controls, which are disabled on startup because the program does not have a connection to the mill. The red rectangle on the right surrounds the area where the video will be displayed. The green rectangle in the lower left surrounds the program's status messages, notifying the user of recent events and operations. This status message can be clicked to open an event log which contains all the status messages throughout the running time of the program.

## 1.2.2 System Menus

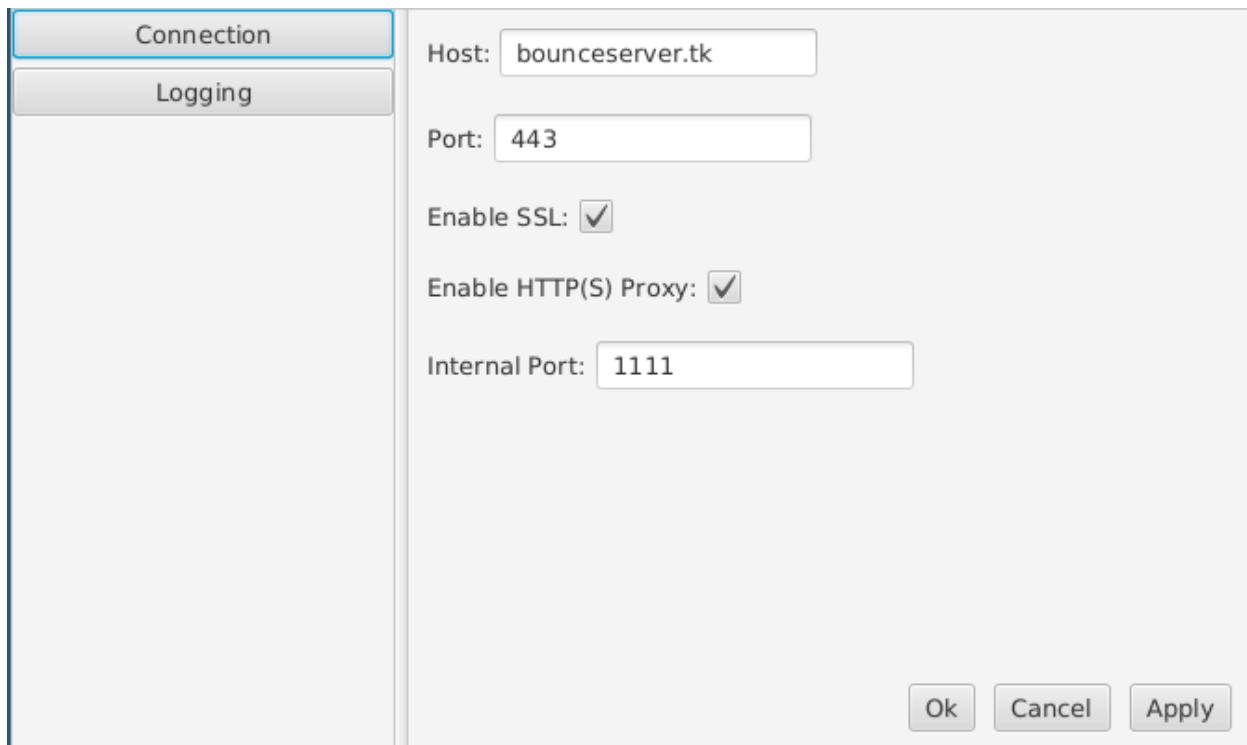
### 1.2.2.1 File Menu

The file menu contains the buttons to open the settings and diagnostic windows.



### 1.2.2.1.a Settings

#### 1.2.2.1.a.1 Connection Settings



In the connection settings menu, the user can change the settings the network client will use to connect to the bounce server.

---

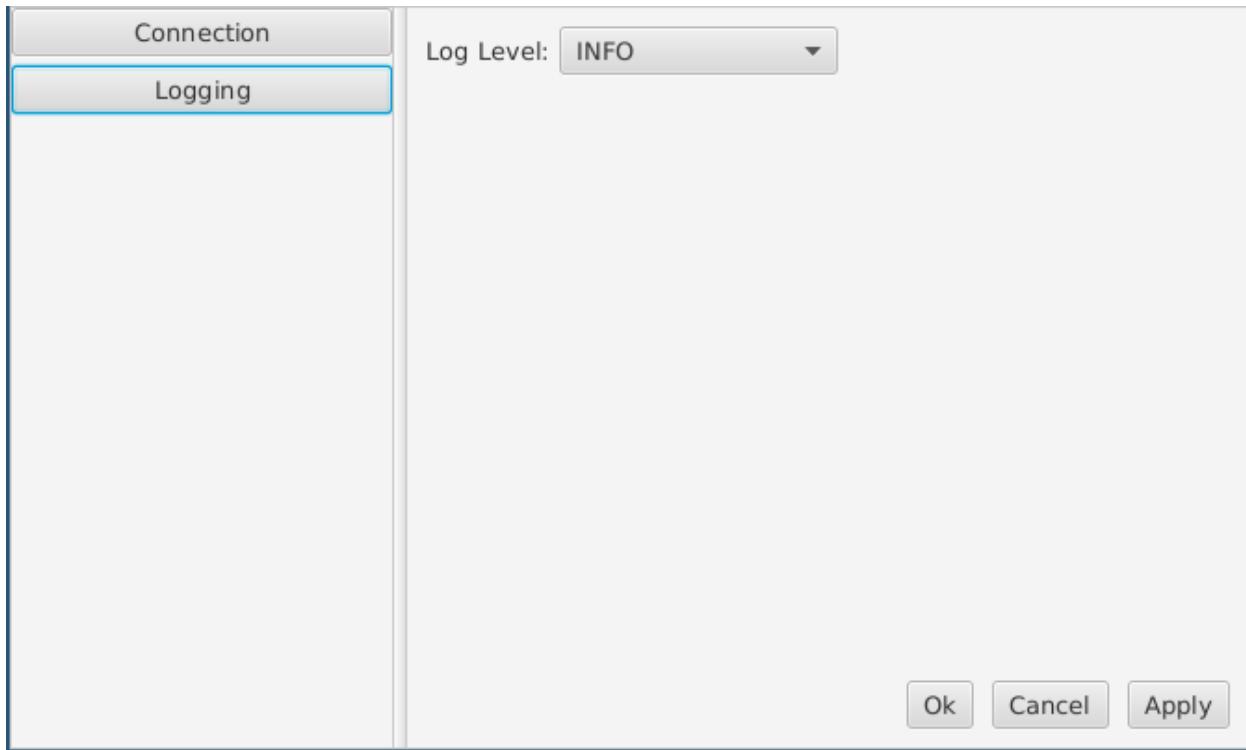
The "Host" and "Port" field change the location on the internet that this client will connect to. These parameters should be the IP address or hostname of the bounce server.

The "Enable SSL" checkbox turns on SSL capabilities currently based on OpenSSL 1.1. This ensures that transferred data is encrypted and secure. The bounce server defined in the "Host" and "Port" fields must support SSL if this checkbox is selected, otherwise the connection will fail. Enabling SSL can help hide the data this connection is transferring in case of snooping or router deep packet inspection blocking unknown protocols.

The "Enable Proxy" checkbox selects whether to wrap the data stream in an HTTP proxy. At the moment, this program only supports using a proxy to connect to a bounce server on the same computer (loopback adapter) as the proxying computer. Like SSL, the bounce server must support HTTP proxies, otherwise the connection will fail. Selecting this option also enables the "Internal Port" field, which must be filled out for the connection to succeed. Using an HTTP proxy can help mask the connection this program makes on particularly picky firewalls.

The "Internal Port" field defines the localport the bounceserver is listening on that the proxy should forward this connection to.

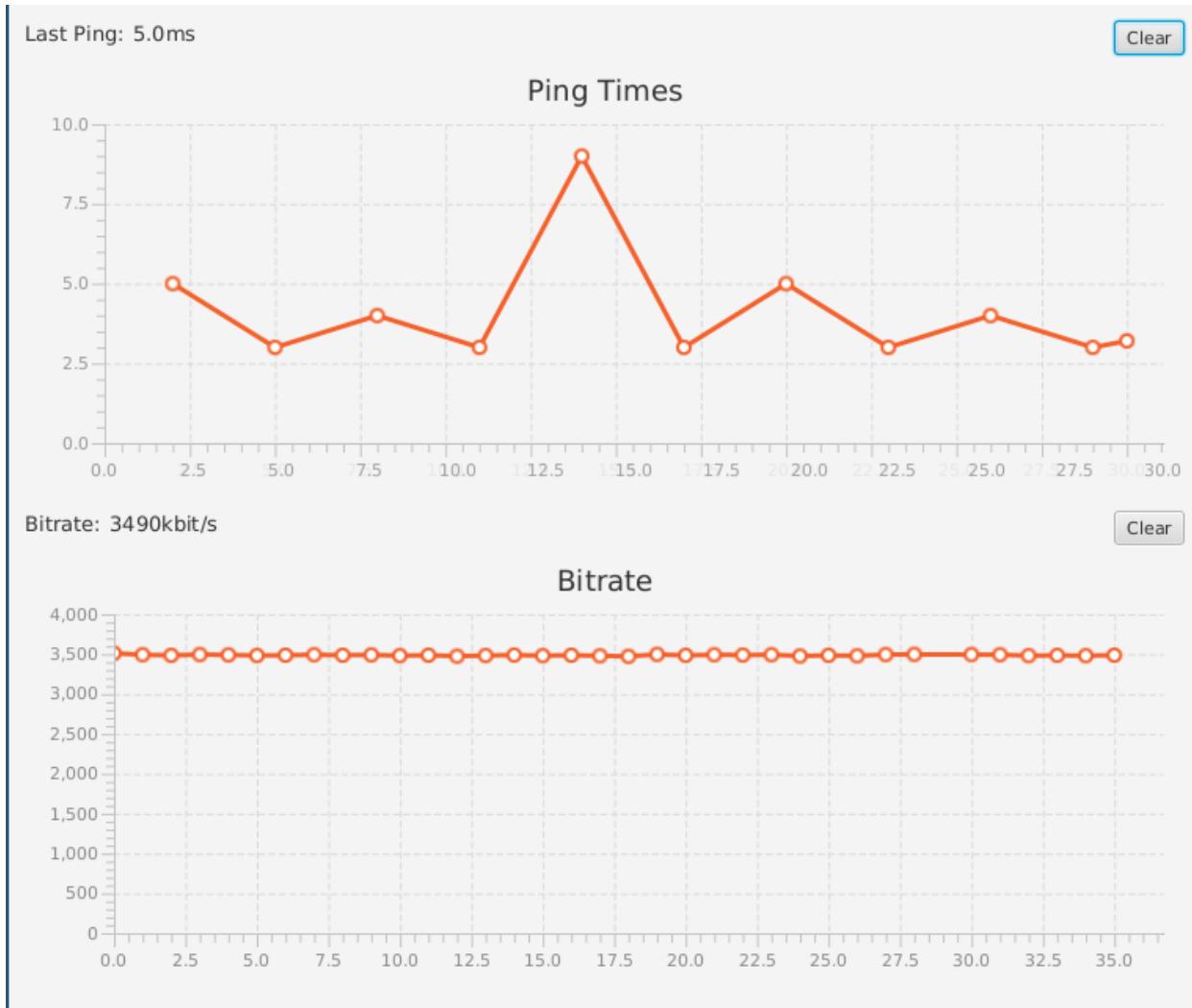
#### **1.2.2.1.a.2      *Logging Settings***



In the logging settings menu, the user can change the granularity of the log messages printed and stored in the event log to assist with connection debugging or other run time issues.

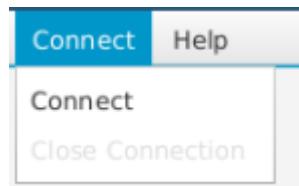
#### **1.2.2.1.b      *Diagnostics***

The diagnostics menu contains information about the current state of the program. The top section displays the time taken to send and receive the last ping to the mill, as well as a graph that displays the ping times throughout the running time of the program. The lower section displays a similar graph, but of video bitrate while the video is active. The X-Axis of the two graphs are aligned based on program running time for ease of comparison.



### 1.2.2.2 Connect Menu

The connect menu has one of two options depending on the current state of the program. If a connection has not yet been made, the Connect button will be available, and clicking it will attempt to make a connection based on the settings defined in **Section 3.2.1.a.1**. If a connection has already been established, then the Close Connection button will be available, and clicking will disconnect this program from the bounce server.



### 1.2.3 Controlling the Mill

Once the connection has been established to the bounce server and the mill, the controls will be enabled, and this program will start receiving and displaying video.

The user can then control the mill by either pressing the buttons in the left panel, or by pressing the key bindings corresponding to the movement buttons.

For convenience, the mill control buttons are also bound to keys as follows:

Jog X Forward: Right Arrow  
Jog X Backward: Left Arrow  
Jog Y Forward: Up Arrow  
Jog Y Backward: Down Arrow  
Jog Z Up: Page Up  
Jog Z Down: Page Down  
Rotate A Clockwise: X  
Rotate A Counterclockwise: Z  
Speed Up: Ctrl + Movement Key

The live HD video is displayed in the right panel and can be used for inspection purposes.



#### 1.2.3.1 Full Screen Video:

For maximum screen real-estate, the video may be popped out of the panel into a separate window by pressing the "Popout Video" button. It may then be maximized or moved independently from the main window. The key bindings can still be used to control the mill while the separate video view is in focus. The video can be returned to the main window by either closing the video window, or by pressing the "Popout Video" button again.

---

#### **1.2.4 Exit System**

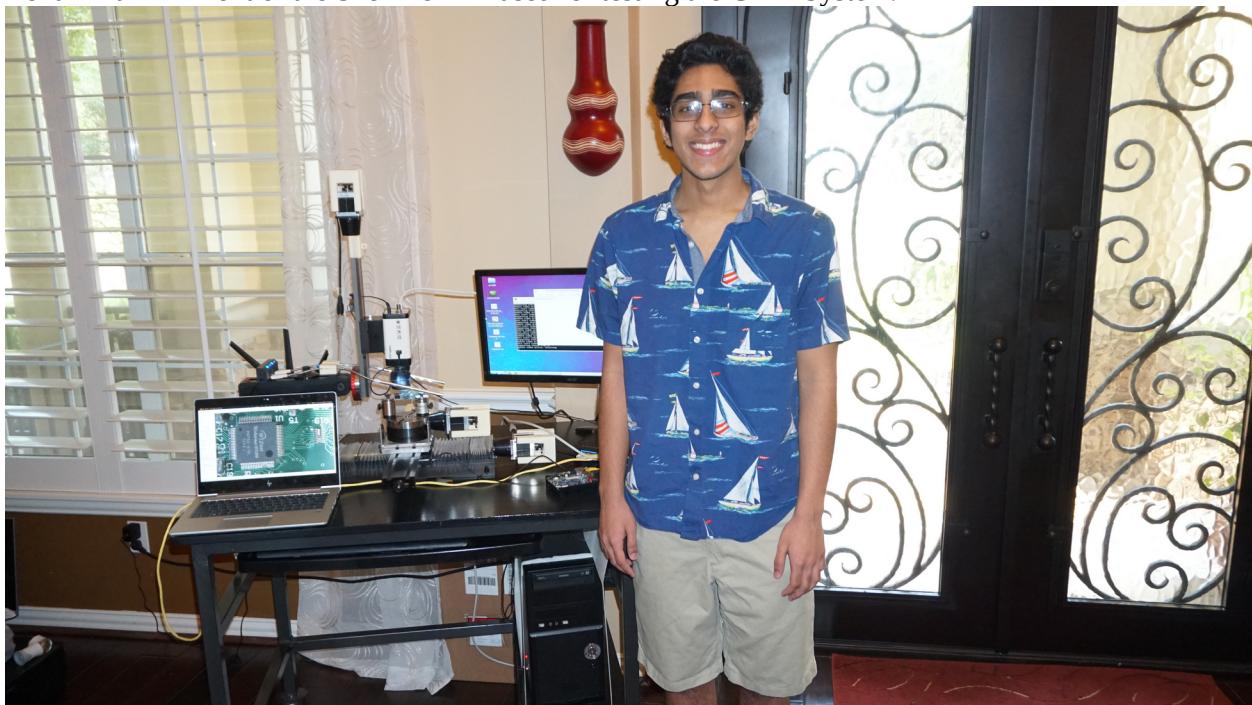
The program can be safely closed at any time during the operation period. If the mill program does not receive a “keep alive” message from the client program it will stop the mill and also the video streaming. It is recommended to close this program if you wish to save network bandwidth being used by the HD video streaming.

#### **1.2.4 Running Multiple Clients**

The system supports connecting multiple user-control clients to the bounce server at one time. Each client will be able to control the mill as well as receiving real-time video.

### **AUTHOR**

Ronak Malik in front of the Sherline Mill used for testing the CBIT System.



Check out my GitHub!

<https://github.com/BeyondPerception>