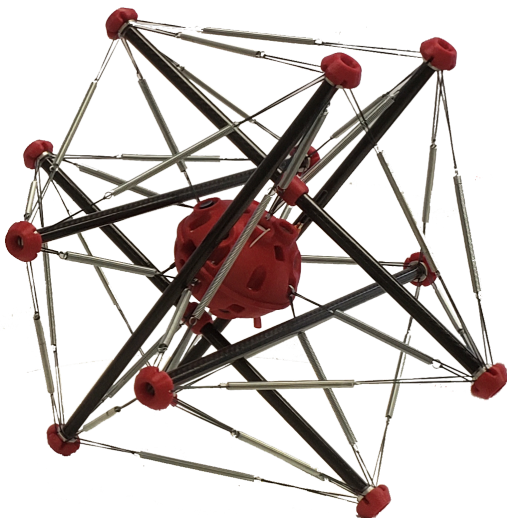




SQUISHY
ROBOTICS



Stationary Robot Streaming Information Guide

Version 1.0

18 July 2024

Squishy Robotics
2600 Tenth Street
Suite 308
Berkeley, CA 94710
Squishy-Robotics.com
info@squishy-robotics.com

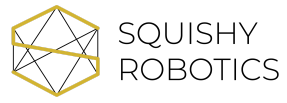


Table of Contents

Document Information	4
Version Number/Document Date	4
Intended Audience	4
Overview	5
Glossary	5
Operating Streaming Services	6
Configuring Data over JSON/KML	10
Overview and Messages/Title UI Behavior	10
ATAK/iTAK/WinTak Streaming	17
Subscribe Device ATAK App to Squishy Data	17
Viewing Squishy Robotics Data via ATAK	21
JSON and Video Streaming	23
Run Instructions	23

Document Information

Version Number/Document Date

Version	Updated Information	Date
Draft 0.1	Stationary Robot Streaming Information Guide DRAFT release	24 April 2023
Version 1.0	Stationary Robot Streaming Information Guide	18 July 2024

Intended Audience

This user guide provides data streaming-related instructions for the Squishy Robotics Stationary Robot and should be read by all individuals responsible for integrating the robot with third-party services. This user guide focuses on streaming services that can be connected to the Squishy Robotics User Interface (UI) and its related robot sensor data and videos. Such streaming connectivity can expand the numbers of viewers as well as the variety of viewing platforms for the reported data.

Overview

This guide focuses only on the operation of streaming add-on services and applications that can connect with Squishy Robotics Data Reporting, Visualization, and Management Software.

For detailed information about the tensegrity robot's deployment, payload sensors, maintenance, and other related operational information, please refer to the latest version of the *Squishy Robotics Stationary Robot User Guide*.

Glossary

Common terms and names used in this document are defined in the following table.

Name/Term	Definition
Tensegrity	A structure made up of isolated components that are positioned within—and keep their shape due to—a net of continuous tension.
Data Reporting, Visualization, and Management Software	The name of the Squishy Robotics software that runs on the Chromebook. Users employ this software when operating the robots.

Operating Streaming Services


The Squishy Robotics User Interface (UI) facilitates the following streaming service applications:

- Transmitting data via JavaScript Object Notation (JSON)
- Transmitting geolocation and robot sensor data through Keyhole Markup Language (KML) files

Streaming data exchange applications require setup and configuration within the Squishy Robotics UI as well as within the receiving software.

Before streaming services can be used, the Squishy Robotics application must be started, an XBee selected, and connectivity with a robot established.

The UI automatically selects an XBee device and establishes robot connectivity when an XBee device is already plugged into the Chromebook after the laptop is

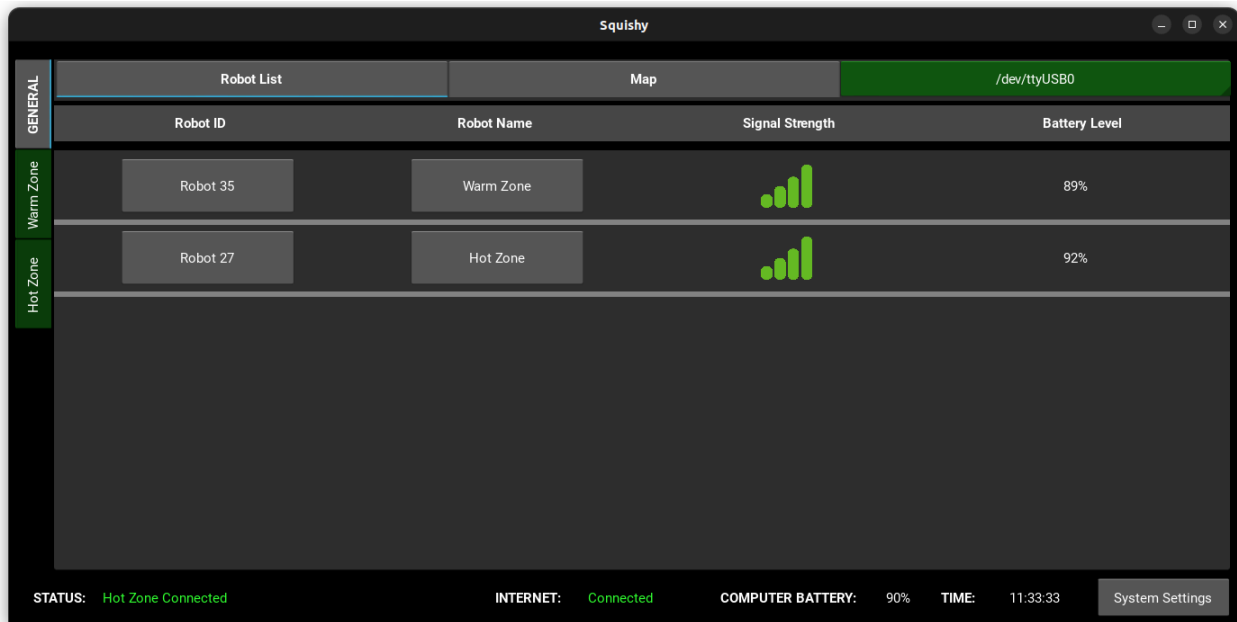
turned on and before the Squishy Robotics GUI icon  is double-clicked to start the Squishy Robotics application.

Normally, connectivity with the robot(s) is established before or just as the dashboard appears, but occasionally the dashboard appears with the upper right appearing gray and the **STATUS** message displaying **Searching for Robots**. The green vertical tab appears and the status bar indicates when a robot is found by displaying **Robot Found!** and then naming the connected robot. Such screen activity may happen so quickly that it can take place almost unnoticed.

In this documentation, the dashboard example screenshots show two meshing robots (named **Hot Zone** and **Warmn Zone**); the **Hot Zone** robot is meshing with the **Warm Zone** robot that then forwards and transmits data to the UI.

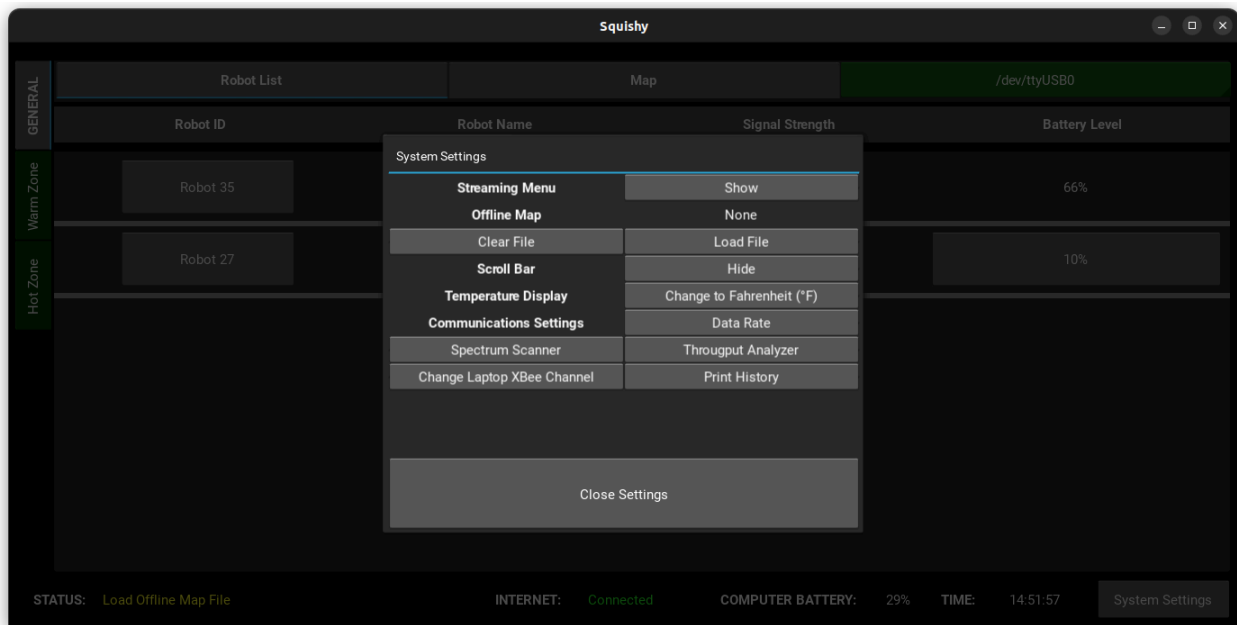
When the **GENERAL** vertical tab is active, the two horizontal tabs, **Robot List** and **Map**, appear in the top row along with the green XBee tab.

The **System Settings** button (lower bottom right in the dashboard's **STATUS** bar) opens a pop-up menu that enables system-wide settings and options that include streaming.



1. Click dashboard's System Settings () button.

The System Settings pop-up menu appears.

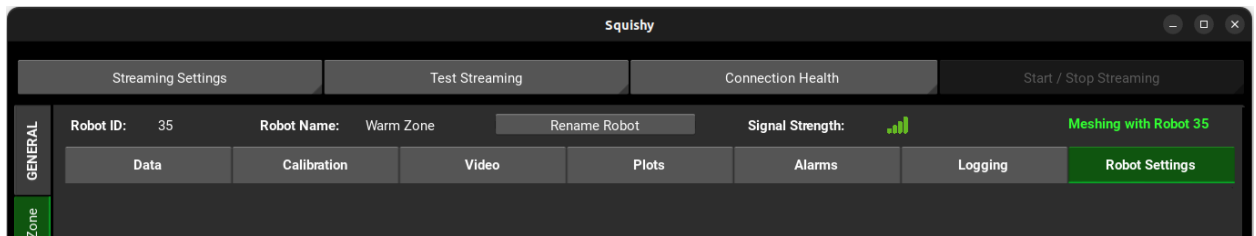


The Streaming Menu Show/Hide button displays the tools that allow data collected via the Squishy Robotics UI to be communicated to third-party services.

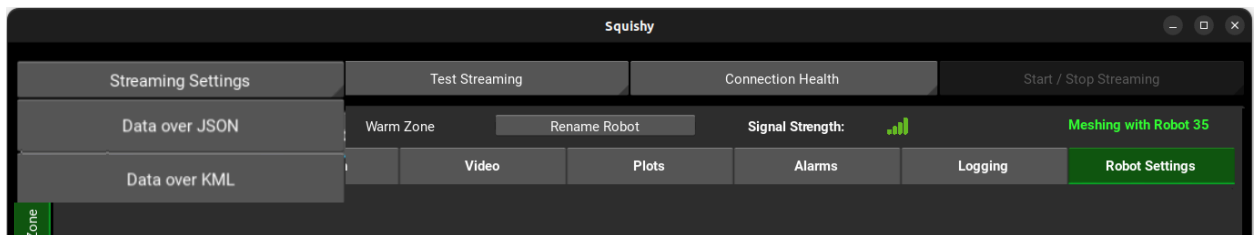
2. Click the Show button (this button toggles between Show and Hide).

3. Click the **Close Settings** button.

The pop-up menu disappears and the dashboard reveals a new top row of streaming-related buttons: **Streaming Settings**, **Test Streaming**, **Connection Health**, and **Start/Stop Streaming** (which is blacked-out/inactive until a streaming service is correctly configured).



4. Click **Streaming Settings** and select one option from the drop-down menu to select how data should be exchanged: **Data over JSON** or **Data over KML**. (Note that JSON and KML can each be separately selected and configured and can then run concurrently.)



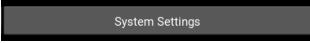
The **Test Streaming** and **Connection Health** options offer the same two pull-down menu choices (**Data over JSON** and **Data over KML**). Once the configuration settings (covered in the following subsection) for a data exchange application have been loaded and connectivity is successful, these two options provide a way to check that a streaming data exchange application is running correctly. Clicking the **Start/Stop Streaming** button begins or terminates streaming communications; the **STATUS**: message is updated accordingly.

Click the **Test Streaming** button and select a drop-down choice to check if the streaming data exchange application is operational. The **Connection Health** menu automatically opens when a test is run (**Connection Health** can also be opened with a manual click). Each drop-down choice also displays a green background (connected), red background (not connected), or gray background (not running or untested).

Based upon connectivity, the **STATUS** displays **Streaming Connection Test Successful** (in green) or **Streaming Connection Test Unsuccessful** (in red).



The **System Settings** pop-up can be used to return the dashboard to normal display after streaming activities are no longer needed.

1. Click dashboard's **System Settings** () button.
2. Click the **Hide** button on the **System Settings** pop-up menu.
3. Click **Close Settings**.

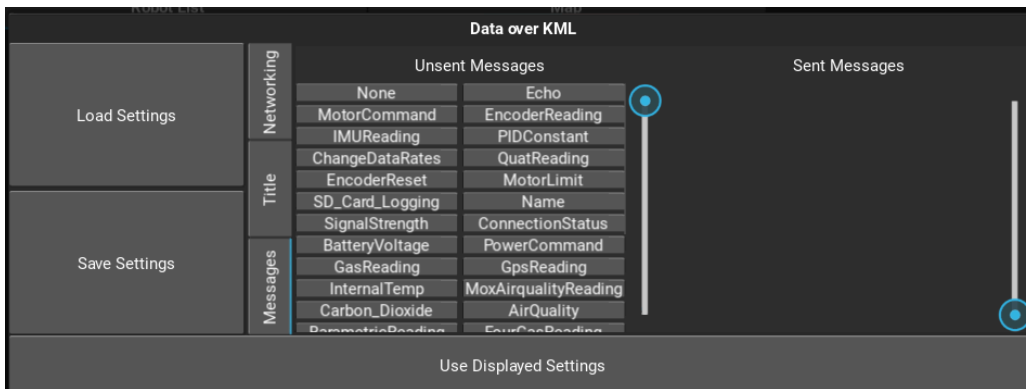
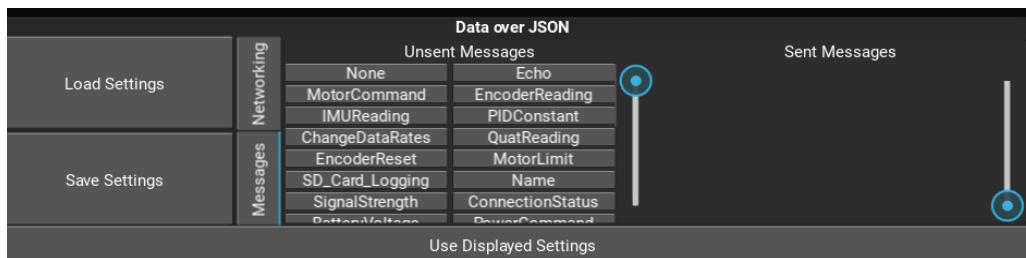
The dashboard no longer displays the row of streaming-related buttons.

Configuring Data over JSON/KML

After JSON or KML is selected (from the **Streaming Settings** button pull-down) as the data exchange application, various configuration settings (such as message, location, and authentication information) must be entered and loaded. Setup and software flow operate similarly for both data exchange applications.

Overview and Messages/Title UI Behavior

When JSON is the selected data exchange, the UI displays the **Data over JSON** pop-up (top image). When KML is the selected data exchange, the UI displays the **Data over KML** pop-up (bottom image).



The UI for both data exchange applications makes use of and presents an identical list of built-in messages when the vertical **Messages** tab is clicked. Additionally, a built-in title choice list is available for KML data exchange when its **Title** tab is clicked. **Messages** consist of various sensor data; **Title** consists of exactly one sensor data. **Sent Messages** and **Title** are streamed from the robot to the Chromebook to the third-party service.

Scroll bars can assist the viewing and manipulating of the unset and sent messages lists (a scroll bar is also available for the KML **Title** choices). When a message is clicked, each "list builder" (sometimes called a "dual listbox") moves the

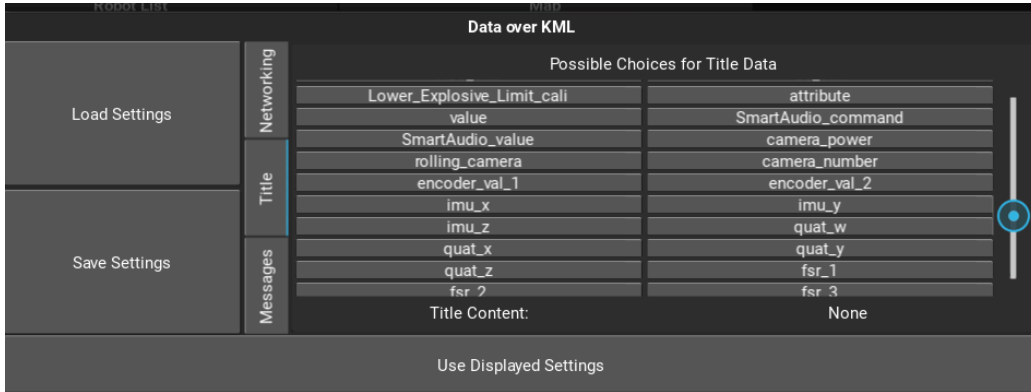
message from the **Unsent Messages** column to the **Sent Messages** column (and vice versa). Multiple messages can be selected.

The following table lists all the message choices for both JSON and KML; the table color codes messages to rank their relevance and value for use with the Squishy Robotics 4-Gas^{PLUS} sensor payload and robot.

Possible Message Choices	
None	Echo
MotorCommnd	EncoderReading
IMUReading	PIDConstant
ChangeDataRates	QuatReading
EncoderReset	MotorLimit
SD_Card_Logging	Name
SignalStrength	ConnectionStatus
BatteryVoltage	PowerCommand
GasReading	GpsReading
InternalTemp	MoxAirqualityReading
Carbon_Dioxide	AirQuality
BarometricReading	FourGasReading
ChemCalibration	ChemConsts
SmartAudio	CameraControl
EnclmuQuatReading	FSRReading
MasterBroadcast	SoftwareVersion
Notification	AlarmNotification
AlarmSetting	LoggingStatus
DropTest	CalibrationTime

	Primary messages for 4-Gas ^{PLUS} operations
	Secondary messages for 4-Gas ^{PLUS} operations
	Unrelated messages for 4-Gas ^{PLUS} operations

When the **Title** vertical button is clicked, the **Data over KML** pop-up displays the **Possible Choices for Title Data**.



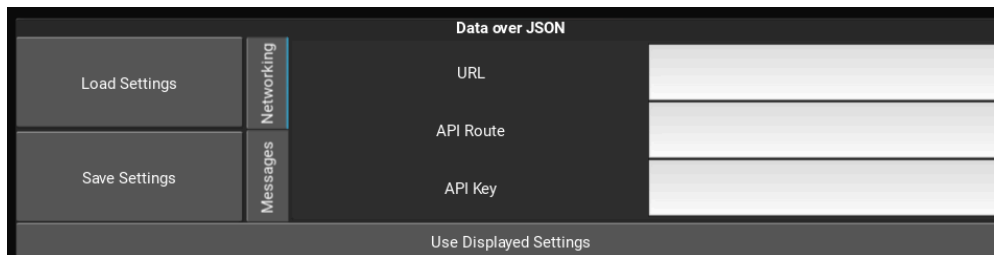
The following table lists all these KML **Title** choices; color coding marks the four most relevant titles. Only one **Title** can be selected for transmission.

Data over KML Title Choices			
None	verification number	press	O ₂
length_motor_commd	motor_commands	LEL	CO
length_encoder_reading	encoder_readings	H ₂ S	calibration_command
imu_x	imu_y	CO_cali	H ₂ S_cali
imu_z	imu_id	O ₂ _cali	Lower_Explosive_Limit_cali
kp	ki	attribute	value
kd	enc_rate	SmartAudio_command	SmartAudio_value
imu_rate	quat_w	camera_power	rolling_camera
quat_x	quat_y	camera_number	encoder_val1
quat_z	quat_id	encoder_val2	fsr_1
enc_val	enable	fsr_2	fsr_3
POS_LIMIT	NEG_LIMIT	fsr_4	fsr_5
cmd	msg	fsr_6	fsr_7
change	name_holder	fsr_8	fsr_9
rss	connected	fsr_10	fsr_11
Is_Unsafe	Volts	fsr_12	robot_id
Amps	Percent_Remaining	robot_type	branch
gas	longitude	hash	compile_date
latitude	internal_temp_celsius	compile_time	notification_code
co2equiv	tvoc	alarm_message_id	alarm_name
temp	humid	alarm_command	upper_bound
CO2	PM_1	lower_bound	LOGGING_STATE
PM_2.5	PM_10	BYTES_WRITTEN	Calibration_Timestamp

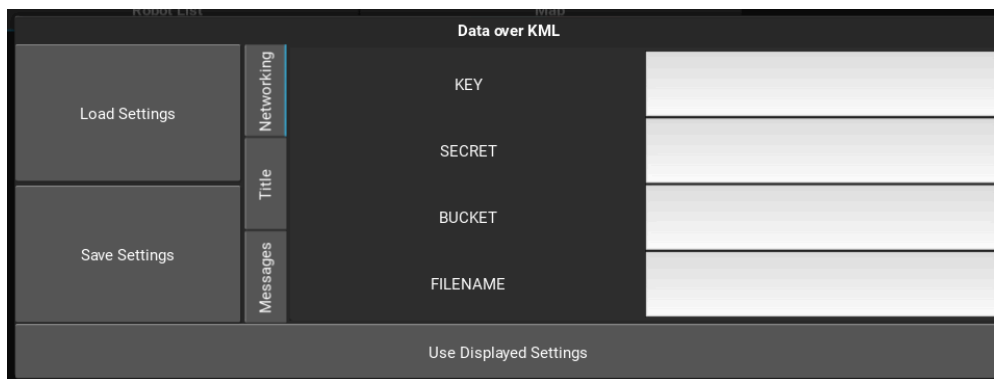
Primary title for 4-Gas ^{PLUS} operations	Secondary title for 4-Gas ^{PLUS} operations	Unrelated title for 4-Gas ^{PLUS} operations
--	--	--

Networking Overview

Depending on which data exchange application is being configured, clicking the **Networking** vertical tab displays either the JSON networking pop-up (top image) or the KLM networking pop-up (bottom image).



Data over JSON	
Load Settings	Networking
	URL
Save Settings	API Route
	API Key
Use Displayed Settings	



Data over KML	
Load Settings	Networking
	KEY
	SECRET
Save Settings	Title
	BUCKET
	Messages
	FILENAME
Use Displayed Settings	

The following tables describe the networking-related information that must be manually typed or be copied and pasted in the text boxes for each data exchange application. (The abbreviations for application programming interface (API) and for uniform resource locator (URL) are used in the pop-ups.)

The content of the **URL**, **API Route**, and **API Key** that are used with JSON networking are best set in consultation with your system administrator or with the developer of the system that is ingesting the data from the 4-Gas^{PLUS} sensor robot. JSON aggregates data from all chosen sent messages as an individual JSON payload contained in an Hypertext Transfer Protocol (HTTP) packet to a specified Internet protocol (IP) URL and/or API address. Data is sent once every five (5) seconds.

The content for the **KEY**, **SECRET**, **BUCKET**, and **FILENAME** that are used with KML networking are settings/credentials/labels that are used with Amazon Web Services (AWS). The KML application uploads KML files to an AWS server; users or services with the required credentials and information can then download such files to their phones or tablets.

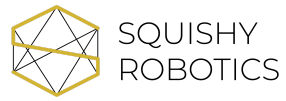
Data over JSON Networking		Examples
URL	A unique identifier for a specific location on the Internet; either a human-readable URL or IP address, as appropriate	http://192.168.0.1:3000
API Route	A specific path to use to transfer data	/api/squishy/sendData
API Key	An authentication code to identify a user or other application	password

Data over KML Networking		Examples
KEY	An authentication code to identify a user or other application	AKIA5H724RGZ5NRWRTEX
SECRET	A password, pass code, or private key	QWER3Vwnvx50oiRZjz3zehMRrlExl/UTfx9wOQds
BUCKET	A container for objects stored in Amazon S3	squishy-atak
FILENAME	Filename of the KML file stored on AWS and downloaded by end users	data.kml

Button Functionality

Three (horizontal) buttons, **Load Settings**, **Save Settings**, and **Use Displayed Settings**, are included in every configuration pop-up. Clicking these buttons implements the following actions:

- **Load Settings:** load the configuration choices and text box information from the software configuration file into the user interface; loaded changes replace any existing settings in the configuration menu.
- **Save Settings:** save the configuration choices and text box information to a permanent file; such saved configuration information may be quickly loaded when restarting the system.



- **Use Displayed Settings:** use the currently displayed configuration choices and text box information and close the currently open pop-up; such configuration settings are not saved to the permanent configuration file.

KML and JSON each have one permanent configuration file.

ATAK/iTAK/WinTak Streaming

ATAK (Android Team Awareness Kit) is a geospatial infrastructure and situational awareness app for civilian or government use. In addition to providing geospatial information, the app's software enables distributed user collaboration without extensive pre-deployed communications infrastructure.

The software was originally developed for military use (the ATAK abbreviation stands for Android Tactical Assault Kit) by the Air Force Research Laboratory (AFRL), and is now maintained by the TAK Product Center (TPC).

While first designed to run on Android systems, the application has versions that run on Apple iOS and on Microsoft Windows.

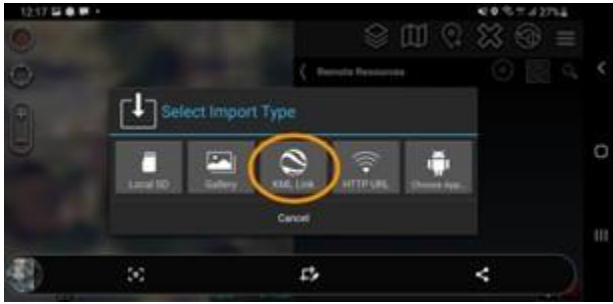
The application (called ATAK or ATAK-CIV) must first be downloaded and installed on a device (phone or tablet) before subscribing the device to AWS to receive Squishy Robotics data.

Subscribe Device ATAK App to Squishy Data

1. Open the ATAK application on a tablet or smartphone.
2. Open the Tools Menu and select Import.

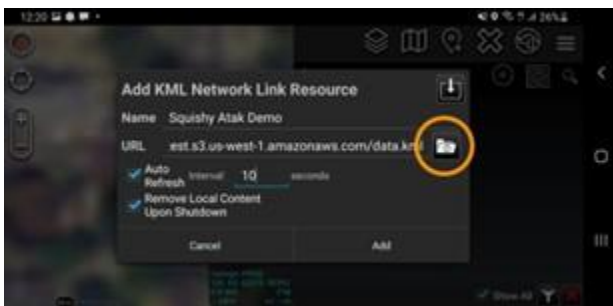


3. Select KML-Link.

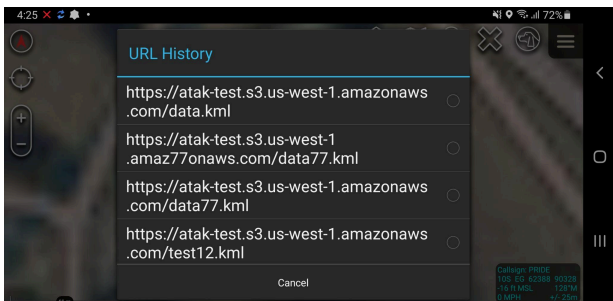


4. In the pop-up that appears, fill in the Name and URL text boxes.

Name is required by ATAK and cannot be left blank. Any name/word(s) that works well for an operation may be used, for example, **Squishy** or **SR Streaming**. The URL must be manually typed or be copied and pasted in the text box (be aware that the *entire* URL path information must be correct).



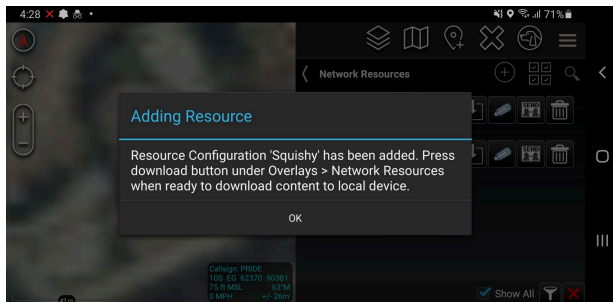
The folder (circled in the previous screenshot) stores the URL History. The URL History (shown in the following screenshot) lists previously used URLs that can be selected (via a click or press) for entry.



5. Select **Auto Refresh** to direct ATAK to continuously download this file (thereby getting the latest data) and enter a two-digit number (**10** is suggested) for the interval time. Be aware that ATAK limits this interval to no shorter than 10 seconds.
6. Click **Add**.
7. Click/press OK to exit.

8. From the Main Menu, navigate to Overlay Manager > Remote Resources.
9. Click/press OK

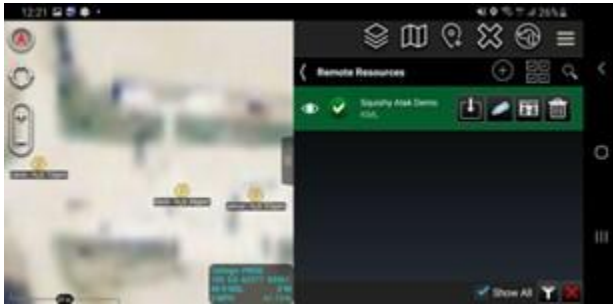
This affirms that Squishy Robotics data is to be downloaded as a Remote Resource.



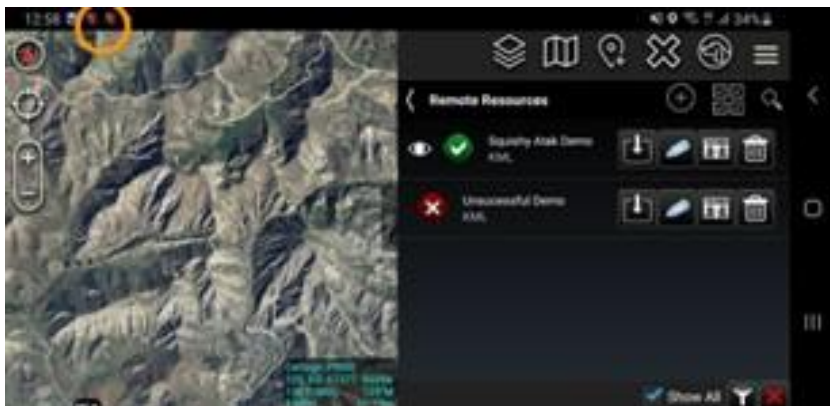
10. When the small popup labeled Verify Download appears, click Stream to begin the download.



When the download is successful, a green checkmark and message appear.



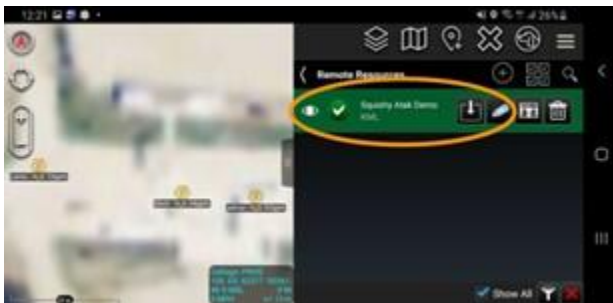
If the download is not successful, the top bar displays a red ATAK notification. Additionally, a pop-up may appear requesting login credentials. The following screenshots illustrate failed downloads. Both errors are typically caused by incorrectly typing the URL. At present, the data file is unsecured and no credentials are required.



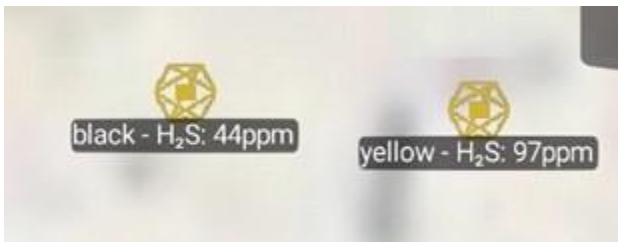
Viewing Squishy Robotics Data via ATAK

Two data viewing options exist: a label-style data view (that is natively shown as a georeferenced pin) and a detailed data view (this view provides robot sensor data as metadata). Both views are described in this subsection.


1. From the **Overlay Manager**, click **Remote Resource** to center the ATAK map on the robots.



2. View the label-style data by examining the text associated with each robot's icon. The format is: **robot name - sensor name: value**. (The **sensor name: value** is the KML Title value that was selected as described in [Configuring Data over JSON/KML](#).)



Robots are typically named after their color, however, the Squishy Robotics UI enables user-settable robot names. (The Chromebook must be used for this robot naming feature.)

3. View detailed data by clicking on a robot icon and then selecting the “more information” icon  (shown in the following screenshot).



A second screen that shows more data appears. The displayed data consists of the **Messages** that were chosen during the KML application **Messages** configuration that was performed via the Chromebook and Squishy Robotics UI (described in [Configuring Data over JSON/KML](#)).



Be aware that this screen does not automatically refresh as additional data files are downloaded. To refresh the data, exit by using the vertical tab in the center of the screen (the tab is to the left of and slightly below the text with the white background) and then reopen to view the most recent data.

JSON and Video Streaming

At present, Blueforce Development Corporation is the only Squishy Robotics partner running JSON data exchanges with the Squishy Robotics UI.

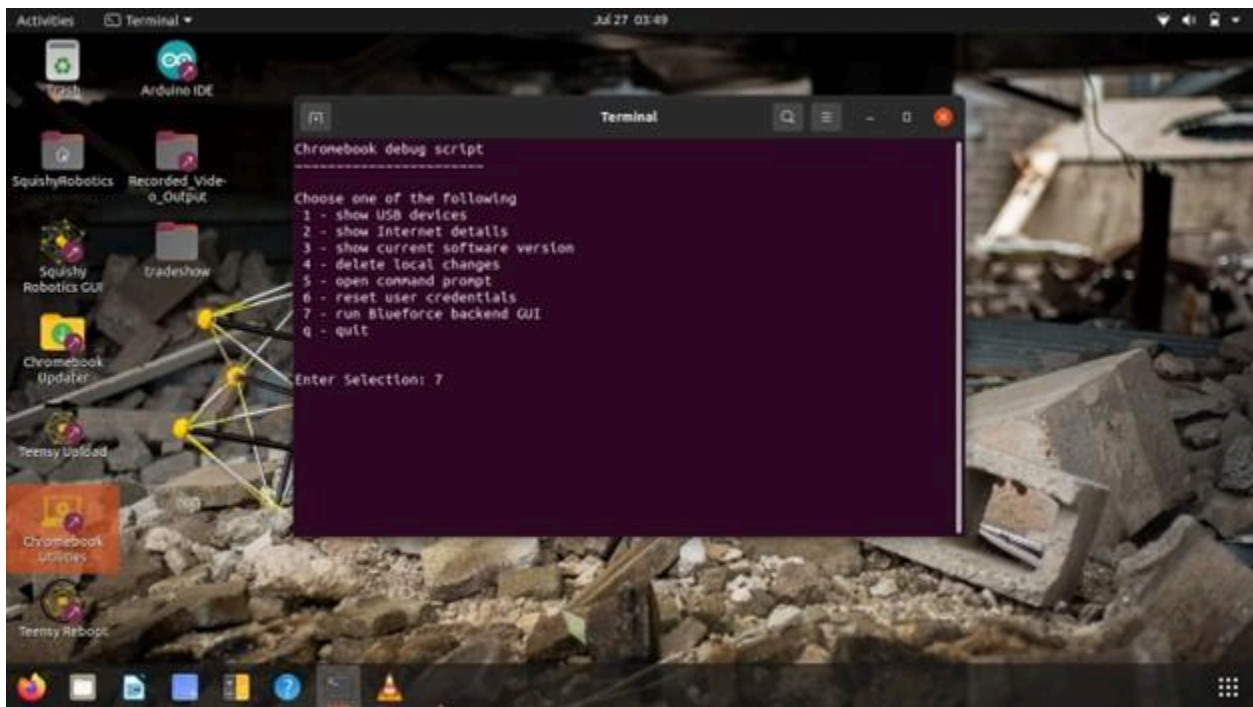
Blueforce-specific instructions and information are provided in this section.

Run Instructions

Both the XBee and the video receiver dongle should be plugged into the Chromebook. Perform the following steps to run the Blueforce software.

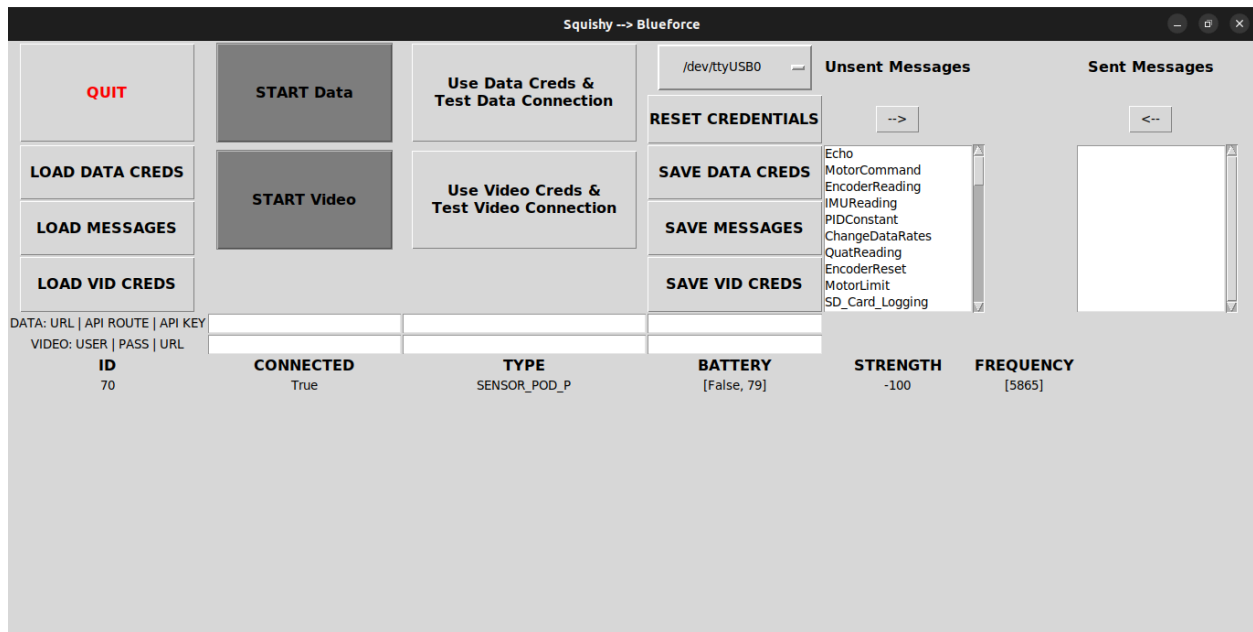
1. Turn on the Chromebook.
2. Double-click the Chromebook Utilities icon  (located on the desktop).

A terminal window appears.

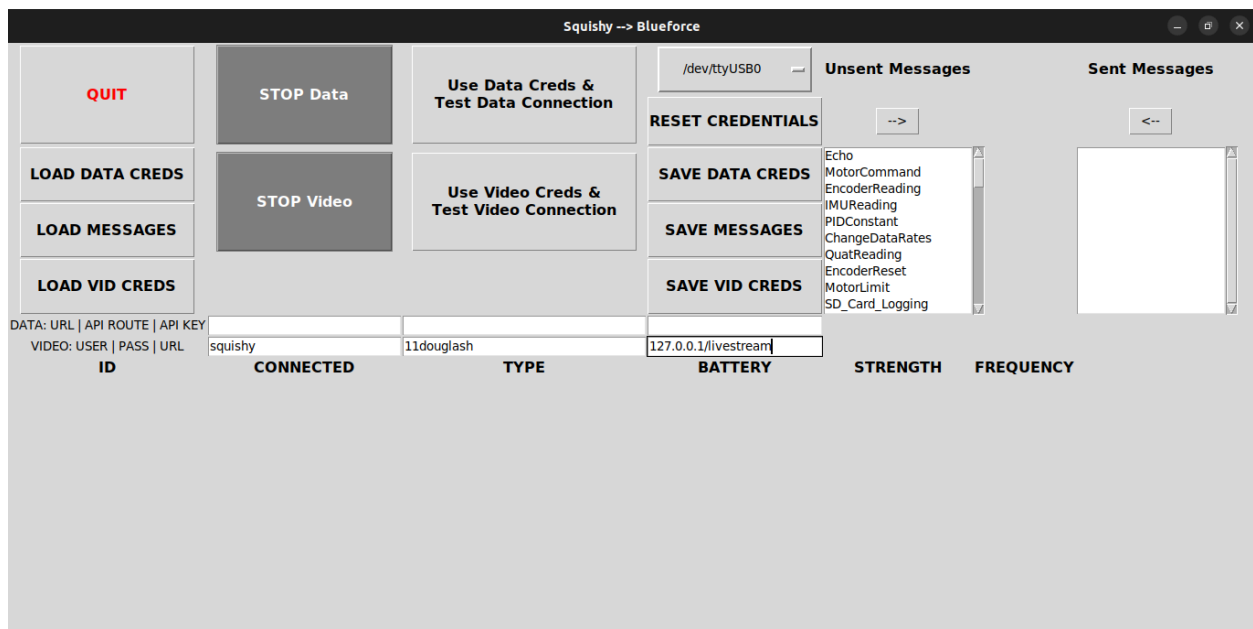


3. Enter 7 as the Selection; this selection option is the run Blueforce backend GUI.

The Squishy --> Blueforce UI appears.



4. Type in the appropriate streaming destination in the bottom text boxes that are labeled VIDEO: USER |PASS|URL.



The above screenshot shows data being streamed to squishy 11douglassh@127.0.0.1/live/stream.

5. Click the **SAVE VID CREDS** button to save the credential information to a file so that the credentials can be loaded with the **LOAD VID CREDS** button next time the Blueforce UI is run.
6. Click the **Use Video Creds & Test Video Connection** button.

The button turns green to indicate that the UI has located the video receiver; the button turns red to indicate that there is a connection-related error.

7. Click the **START/STOP Video** button.

The video begins.

The button turns red to indicate an error, such as the video receiver is not properly plugged in, the URL is invalid, or there is an issue with Internet reception/connectivity. A gray button indicates the video stream has closed; this behavior can happen when a shutdown command has been issued.

8. Use the video transmitter to select the correct channel by pressing and holding down one button until the transmitter starts scanning all channels. The transmitter will auto-select the strongest channel. Stop pressing the button when scanning begins.

