Human-Robot Interface: Efficient Control and Flight Data Display of a Quadcopter through Google Glass

User Guide

June 6, 2015

Prepared By:
Daniel Chen
Laura Demesa
Steven Rodriguez
Justin Salanga

Prepared For:
ECS193A/B

Client:
Dr. Ilias Tagkopoulos

Preface

The Human-Robot Interface project was a part of a University of California Davis Department of Computer Science undergraduate senior design project. A team of four undergraduate students designed and implemented this solution to a problem posed by an assigned client. The team consulted with the faculty mentor and client to identify requirements and improve the implementation through feedback.

The purpose of the Human-Robot Interface project is to provide a convenient process for operating a quadcopter and storing its media files. The ideal solution was a completely hands-free system. However, time constraints, group ability, and practical use were all considered and led to this implementation. In short, the reader should not consider this implementation ideal.

Acknowledgements

This work was supported by the NSF CAREER award CCF-1254205, Center for Information Technology Research in the Interest of Society (CITRIS) seed grant #2469085 to Dr. Ilias Tagkopoulos and the Department of Computer Science, University of California, Davis.

Thank you Dr. Ilias Tagkopoulos and Professor Xin Liu for your support and guidance on this project.

Table of Contents

1.0 Introduction 6

1.1 User Guide Formatting 6

1.1.1 Sections Formatting 6

1.1.2 Instructions Formatting 7

2.0 Definition of Terms 8

3.0 Project Overview 9

3.1 Phantom 2 Vision 10

3.2 Controller 10

3.3 Google Glass 10

3.4 Website 10

4.0 Installation and Setup 11

4.1 Google Glass Setup 11

4.1.1 Obtaining Glass Drivers 13

4.1.2 Accessing the ADB 13

4.1.3 App Launcher Installation 13

4.1.4 Locating the .apk File 14

4.2 Phantom 2 Vision Setup 14

4.3 Phantom Controller Setup 14

4.4 Website Environment Setup 15

5.0 General Use Guide 17

5.1 Data Management 17

5.2 Google Glass 18

5.2.1 How to Connect to Phantom 18

5.2.2 How to Take a Picture 19

5.2.3 How to Record Video 20

5.2.4 How to Clear Phantom SD Card 21

5.2.5 How to Transfer Data from Phantom to Glass 22

5.2.6 How to Transfer Data from Glass to Website 23

5.3 Website 24

5.3.1 How to View Media 24

5.3.2 How to Pair Google Glass 25

5.3.3 How to Manage Users, Uploads, and Settings 25

6.0 Troubleshooting..26

6.1 Google Glass 26

6.1.1 Problem: Glass is not connecting to the quadcopter. 26

6.1.2 Problem: The voice command is not working. 26

6.1.3 Problem: The video stream is not showing. 27

6.1.4 Problem: The application crashed or froze suddenly. 27

6.1.5 Problem: I took a photo/video but I don’t see it in the media list. 27

6.1.6 Problem: When synchronizing Glass with a username the app takes me back to the launcher. 27

6.1.7 Problem: I see the following screen. 28

6.2 Website 29

6.3 Phantom Vision 2 29

7.0 FAQ 30

8.0 Contact Information and Support 31

9.0 Appendix 32

9.1 Design Document 33

9.2 Testing Document 48

1.0 Introduction

This user guide is written with accessibility in mind and covers the following: the project overview, installation, general use, and troubleshooting. Additional product support information is provided, but the user guide lacks in-depth technical details. For technical details about the Human-Robot Interface (HRI) project, please reference the design document attached in the appendix section of this user guide. In addition, this guide assumes the reader has a good understanding of using Google Glass, computers, and the Phantom 2 Vision.

1.1 User Guide Formatting

There are three types of formatting in this user guide: sections, instructions, and lists. Each formatting style is designed to allow easy navigation.

1.1.1 Sections Formatting

Sections provide information about topics that do not require sequencing. A section can be further divided into topic specific subsections. This allows the table of contents to provide easy and quick navigation. In addition, each subsection provides a greater level of detail. Figure 1 below details the design of the section numbering system used in this user guide.

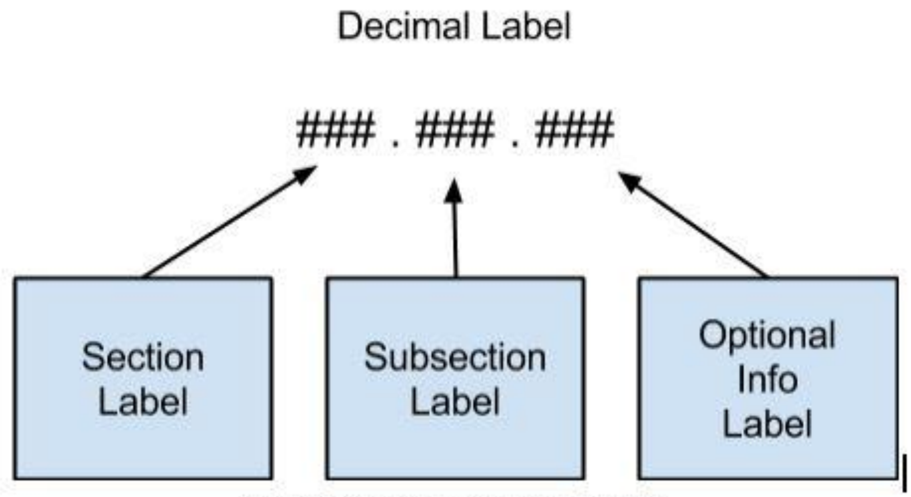


Figure 1: Decimal Label Description

Information under the section and subsection labels is important for understanding the topic. Information under an optional info label provides further details to help readers who would like further explanation. Readers familiar with this project will likely want to skip these sections.

1.1.2 Instructions Formatting

Instructions provide a sequence of steps that lead to a specific outcome. This guide numbers steps starting at one. A step may be further broken down to help readers who would like further explanation. Figure 2 below details the design of the instruction numbering system.

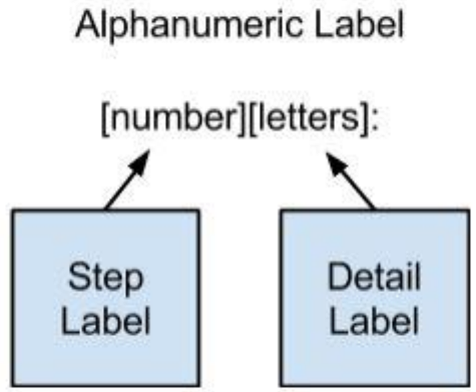
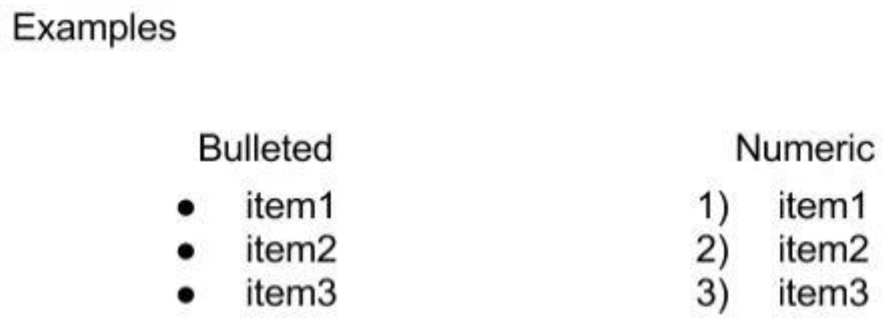


Figure 2: Alphanumeric Label Description

Every step will have a numeral value in its label. The presence of a letter indicates a detail label. Readers familiar with this project will likely want to skip detail steps.

1.1.3 Lists Formatting

Lists provided brief segments of text that relate to a specific topic. Lists will typically be bullet points, but may be numerals when appropriate.



Examples

Bulleted

- item1
- item2
- item3

Numeric

1) item1
2) item2
3) item3

Figure 3: Examples of List Formats

Figure 3 illustrates the two types of formatting used for lists.

2.0 Definition of Terms

1. **ADB**: The android debug bridge is a special user interface for Google Glass that requires a computer. The ADB allows greater access to Google Glass.
2. **Django**: A web application framework written in Python. Follows MVC architectural pattern. Used to create project's website.
3. **Driver**: A computer program that allows the control over a device connected to a computer.
4. **Flight Menu**: The Google Glass interface shown when connected to the Phantom and video stream is shown.
5. **Google Glass**: A wearable computer that is similar to a pair of glasses. Often referred to as Glass in this user guide.
6. **HTTP verbs**: The hypertext transfer protocol uses keywords, such as PUT, to communicate data requests and data transfers.
7. **Media Menu**: The Google Glass interface shown when connected to the Phantom and viewing its files.
8. **MVC**: Model-view-controller is an architectural pattern that details how data should be stored, viewed, and changed.
9. **Phantom 2 Vision**: The quadcopter used in this project, which is made by DJI.
10. **Quadcopter**: An aerial vehicle with four propellers, used interchangeably with UAS and UAV in this user guide.
11. **REST**: Representational state transfer is a software architecture style that typically uses HTTP verbs, which are often used in internet communication.
12. **Sideloading**: The process of installing software to Google Glass through the ADB.
13. **UAS**: An unmanned aircraft system, used interchangeably with UAV and quadcopter in this user guide.
14. **UAV**: An unmanned aerial vehicle, used interchangeably with UAS and quadcopter in this user guide.

3.0 Project Overview

The HRI project's main purpose was convenience. The standard method of controlling a UAS and transferring flight data from the UAS to a computer involved clunky controllers and transferring physical memory cards. The HRI project simplifies the use of a UAS, without sacrificing precision control.

The HRI project has four main components: the Phantom Vision 2, Google Glass, the website, and the Phantom controller. Figure 4 below illustrates the communication of the components. All communication in the project is through Wi-Fi. However, there are two Wi-Fi networks: one for the Phantom connection and one for an internet connection.

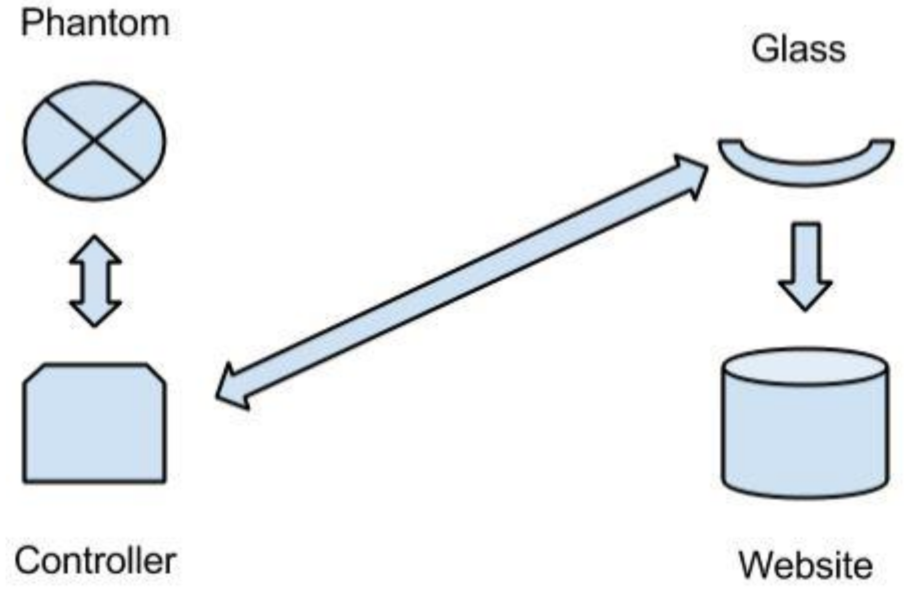


Figure 4: Project Communication Diagram

In general, users will fly the quadcopter using the controller and view the camera feed on Google Glass's display. When the user desires to take a picture or video recording, the user will use Glass to issue the command. Then the user can issue a Glass command to store the media files from the quadcopter to Glass, and finally the user will connect to the internet via Wi-Fi to transfer the media files from Glass to the website.

This project has two code repositories, one for the website and one for the Glass application. The website code provides long-term storage for media files, and a user interface to view the stored files.

The Glass code is a heavily modified DJI SDK "Get Started" Android app. This modified app provides controls for the camera, transfer of media files, and display of live video stream.

3.1 Phantom 2 Vision

Due to potential slow upload speeds, the recording parameters are hardcoded to 1280x720_30p. This is done through the Glass application.

3.2 Controller

There are no modifications made to the stock Phantom 2 Vision controller in this project.

3.3 Google Glass

Google Glass has a sideloaded .apk file that allows communication with the website and the quadcopter. The app provides the following features:

- Voice Commands
- Gimbal Control
- Camera Controls
- Hands Free Media Transfer Phantom to Glass
- Hands Free Media Transfer Glass to Website
- Live Video Stream Display
- Flight Data Display

In addition, we recommended the user install a launcher application on Glass to provide a convenient way to run the HRI app. This is not required, but Glass does not fully support sideloaded apps. For example, the app uses wakelocks to keep Glass from sleeping, because if Glass sleeps the application closes.

3.4 Website

The website implementation is Django and MySQL. A server must be setup to host the website. Once setup the server provides the following features:

- Database for Media Storage
- Web Interface to View and Navigate Media
- Admin Interface
- User Account Support
- Management of active user, user accounts, and files.

4.0 Installation and Setup

The project requires the setup or installation of the four components. However, the setup of the website is only important for data storage. In addition to the setup and installation procedures, this guide assumes the user has access to a Wi-Fi connection at some point for Google Glass to connect to the website.

An individual who is experienced with using computers should do the installation and setup process. For individuals with limited computer experience, many resources are available online that can provide details. A web-browser search can answer many questions beyond the scope of this guide.

4.1 Google Glass Setup

The following instructions will prepare Google Glass for use. It is assumed the user already has an account activated on Google Glass and has gone through the initial Glass setup process.

For more information about the initial Glass setup process, please visit the Google Glass website <support.google.com/glass>.

WARNING: Individuals wanting to setup Google Glass should be comfortable using a command prompt.

Required Materials

- Google Glass
- Traditional Computer
- Micro USB to USB Cable
- Internet Connection
- Application's .apk File (http://dchen93.github.io/profile/media/HRI_v1.0.1.apk)

Instructions

1: Install Android ADB on the computer.

1a: You can find this on the internet as a single file or install Android Studio.

1b: Keep track of the ADB folder location.

2: Move the .apk file into the ADB folder.

3: Install Google Glass Drivers on PC.

3a: Depending on the drivers you get you may need to edit the android_winusb.inf file. This should not be necessary with the drivers you get from Android Studio.

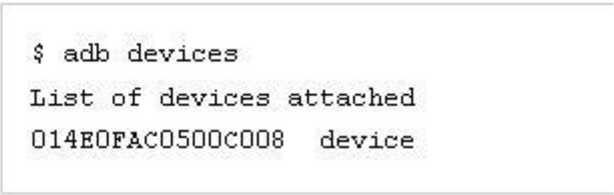
3b: Open the file android_winusb.inf and paste the following text under [Google.NTamd64] and [Google.NTx86] :

```
;Google Glass
%SingleAdbInterface% = USB_Install, USB\VID_18D1&PID_4E11&REV_0216
%CompositeAdbInterface% = USB_Install, USB\VID_18D1&PID_4E11&MI_01

%SingleAdbInterface% = USB_Install, USB\VID_18D1&PID_9001&REV_0216
%CompositeAdbInterface% = USB_Install, USB\VID_18D1&PID_9001&MI_01
```

Note: See this guide for further details about editing android_winusb.inf.
http://glassappz.com/ten-easy-steps-to-side-load-an-app-to-google-glass/

4: Put Google Glass into debug mode.
5: Connect Google Glass to the computer with the usb cable.
5a: On Google Glass, accept the connection if necessary.6: Navigate to ADB folder in a command prompt.
7: Test if the computer recognizes Google Glass.
7a: Make sure no other android devices are connected to the computer.
7b: Type “adb devices” into the command prompt. You should see the results below; however, the device id will likely be different.



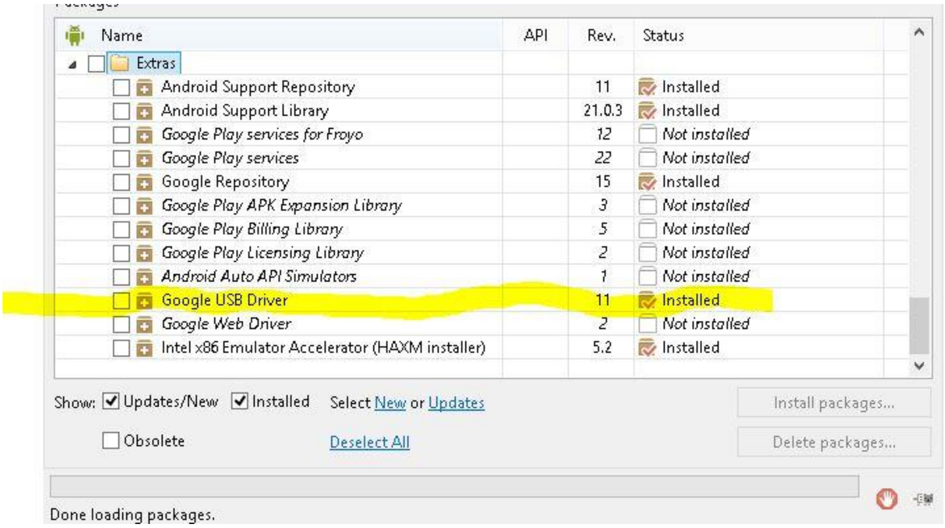
```
$ adb devices
List of devices attached
014E0FAC0500C008   device
```

7c: If you do not see the device listed, please make sure the Glass driver is properly installed.
8: Install the apk file to Glass by typing “adb install ApkName.apk”.
8a: The command prompt should confirm the installation.
9: Google Glass is ready! You can disconnect Glass and exit debug mode.

TIP: Use “adb shell pm list packages -f ” to view already installed apps on Glass. Use “adb uninstall com.packageName.app” to uninstall an already installed app on Glass.

4.1.1 Obtaining Glass Drivers

If you cannot find the Glass drivers online as a standalone file, an alternative is to install Android Studio or Eclipse. Each IDE will have a menu for packages you can install. Below is the package name for Android Studio that has the Glass driver.



Name	API	Rev.	Status
☐ Extras			
☐ Android Support Repository		11	Installed
☐ Android Support Library		21.0.3	Installed
☐ *Google Play services for Froyo*		*12*	*Not installed*
☐ *Google Play services*		*22*	*Not installed*
☐ Google Repository		15	Installed
☐ *Google Play APK Expansion Library*		*3*	*Not installed*
☐ *Google Play Billing Library*		*5*	*Not installed*
☐ *Google Play Licensing Library*		*2*	*Not installed*
☐ *Android Auto API Simulators*		*1*	*Not installed*
☐ Google USB Driver		11	Installed
☐ *Google Web Driver*		*2*	*Not installed*
☐ Intel x86 Emulator Accelerator (HAXM installer)		5.2	Installed

Done loading packages.

4.1.2 Accessing the ADB

The initial installation of the ADB does not add an environment variable to the computer. We recommend adding the ADB folder path to the list of local variables. This will allow the user the ability to access the ADB in any directory while using the command prompt.

Always remember that Google Glass must be in debug mode to access the ADB. In addition, the first time Glass connects to a system, Glass prompts the user to allow the connection. If this is not approved the ADB cannot be accessed.

4.1.3 App Launcher Installation

Google Glass does not fully support sideloaded applications. To help run the project's app, we recommend installing a launcher app. This will provide an easy way of using the app. During development, we used Glass Launcher. However, there are several launcher apps available and this project does not require a specific one. It is highly likely you will need to sideload the launcher app onto Glass.

Launcher apk: http://dchen93.github.io/profile/media/glasslauncher.apk

4.1.4 Locating the .apk File

Please visit https://humanrobotinterface.wordpress.com/ for many of the project's resources.

4.2 Phantom 2 Vision Setup

The phantom 2 Vision has no additional setup for this project. Users should follow the instructions provided by DJI to complete a standard setup. Please consult DJI's Phantom website for instructions about this process. You can find the website information in the contact section of this user guide.

We highly recommend recalibrating the Phantom 2 Vision before every flight.

4.3 Phantom Controller Setup

The Phantom 2 Vision controller has no additional setup for this project. Users should follow the instructions provided by DJI to complete a standard setup. Please consult DJI's Phantom website for instructions about this process. You can find the site information in the contact section of this user guide.

4.4 Website Environment Setup

Setting up the website varies from system to system. Covering all possible configurations is beyond the scope of this user guide. The following setup described below allows the user to modify, run, and debug the website. Readers should use the instructions as a general guideline rather than a systematic description.

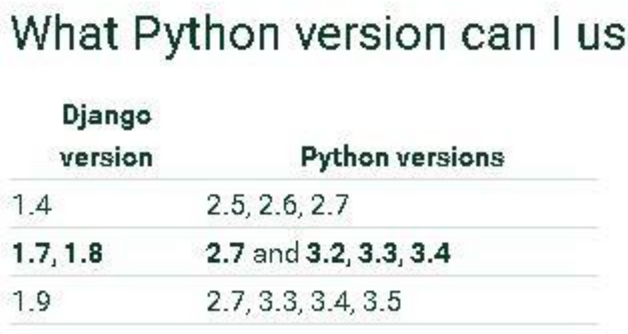
NOTE: The server is already deployed. The website environment setup is for developing the website further. If you are developing on the project's deployed-website server, the environment is already setup.

Instructions

1: Install Python on your system.

1a: You can find these files at python.org.

WARNING: See the below image for compatibility. Visit Django site for latest compatibility table.



What Python version can I use with Django?

Django version	Python versions
1.4	2.5, 2.6, 2.7
1.7, 1.8	**2.7** and **3.2, 3.3, 3.4**
1.9	2.7, 3.3, 3.4, 3.5

2: Install Python Package Managers easy_install, then pip.

2a: You must install easy_install before pip.

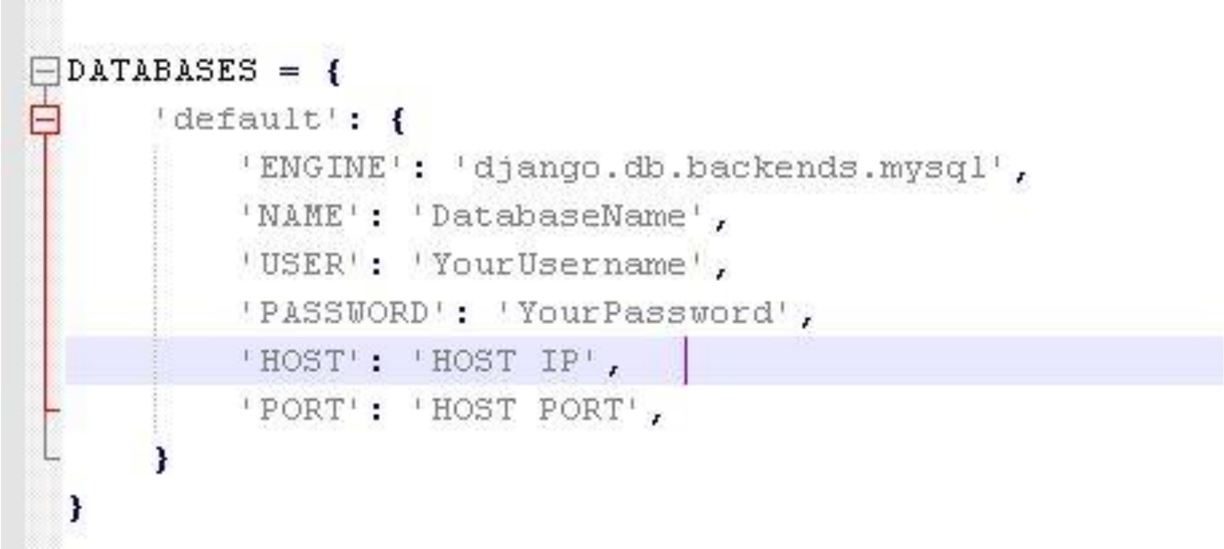
NOTE: Some versions of Python already have easy_install and pip.

3: Setup Git version control. This will be used to obtain the website source code.

3a: Git clone: https://github.com/dchen93/Human-Robot-Interface-Web

4: Setup a database. We used MySQL.

4a: In the project folder, fill out the following in hrisite/settings.py. This should match the setup of your database setup.



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'DatabaseName',
        'USER': 'YourUsername',
        'PASSWORD': 'YourPassword',
        'HOST': 'HOST IP',
        'PORT': 'HOST PORT',
    }
}
```

5: Use pip to Install virtualenv, this is used to make further development simpler.

5a: Command example: pip install virtualenv

6: Enter virtualenv for project.

7: Make sure the following is installed via pip:

- Django
- Django Rest Framework
- MySQL-python

8: Create database tables.

8a: Navigate to manage.py file.

8b: Use command: python manage.py makemigrations

8c: Use command: python manage.py migrate

9: Environment setup! Use command: python manage.py runserver to run the server.

WARNING: In the settings.py file, make sure debug mode is turned off when not developing.

TIP: Don’t forget to leave your virtualenv when no longer using the website.

5.0 General Use Guide

This user guide assumes the user is familiar with Google Glass and operating the Phantom Vision 2. Many of the interfaces have on screen instructions that are easy to follow. However, for completeness we have included several use cases.

5.1 Data Management

In order to avoid duplicate file creation, the app has a nuanced data management system. Figure 5 below demonstrates the optimal use of the app when considering data management. This allows the most choice in choosing what files to save.

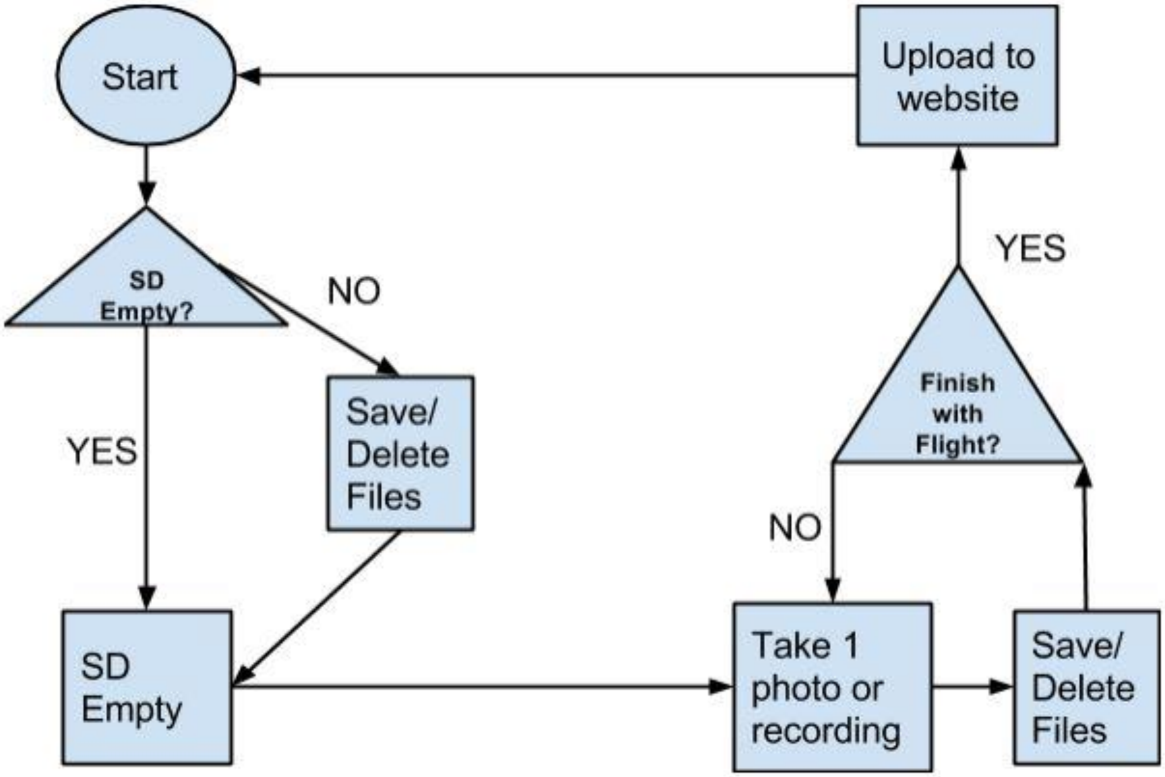


Figure 5: Optimal Data Management Flow Chart

However, it may be tedious to delete/save after every media file. You can create multiple media files at a time, but delete/save will apply to all the files created. If the user saves an unwanted file to Glass, then you will need to wait until uploading to the website to delete this file. There may be a way to navigate the files on Glass to delete the file, but this is not a part of the project.

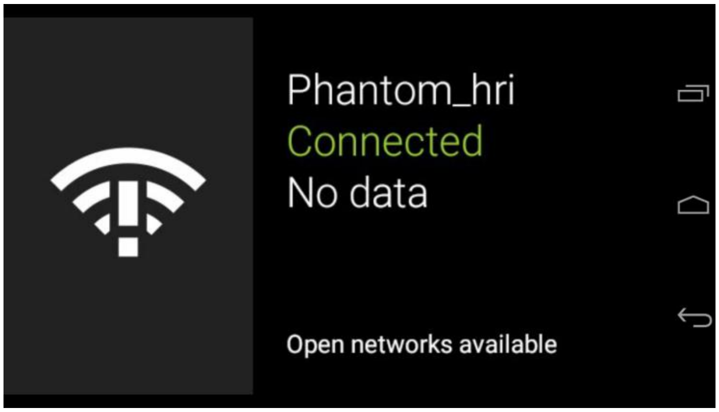
5.2 Google Glass

TIP: A user will want to pair Glass to a user on the website before flights.

5.2.1 How to Connect to Phantom

1: Turn the phantom on.
2: Turn the Wi-Fi extender on.
3: Turn Google Glass on.
4: Navigate Glass UI to settings.
    4a: Navigate to Wi-Fi menu.
    4b: Select Phantom Wi-Fi network.

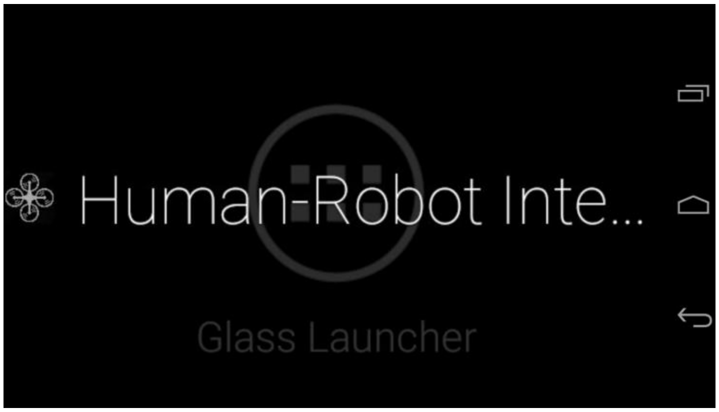
Note: There is no internet connectivity when connected to Phantom.



Phantom_hri
Connected
No data

Open networks available

5: Launch HRI App (Note: This screenshot shows the optional launcher app)



Human-Robot Inte...

Glass Launcher

5.2.2 How to Take a Picture

1: Connect Glass to Phantom
2: Navigate to Flight Menu

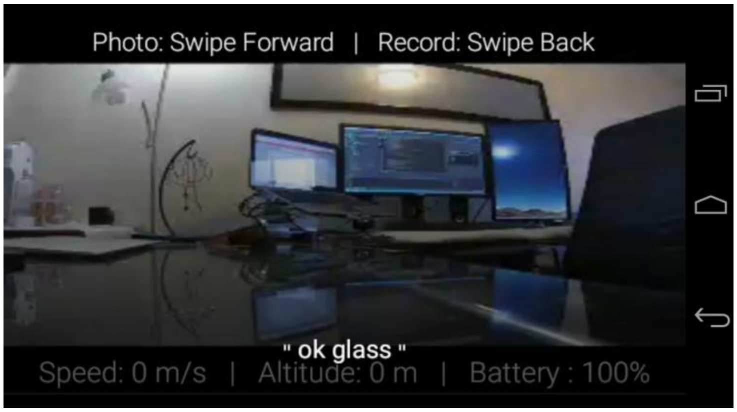


Photo: Swipe Forward | Record: Swipe Back

" ok glass "

Speed: 0 m/s | Altitude: 0 m | Battery : 100%

3: Use voice command “Ok Glass” to activate Glass listener.
4: Use voice command “Take a Picture”
5: Alternatively, swipe forward to take a photo.

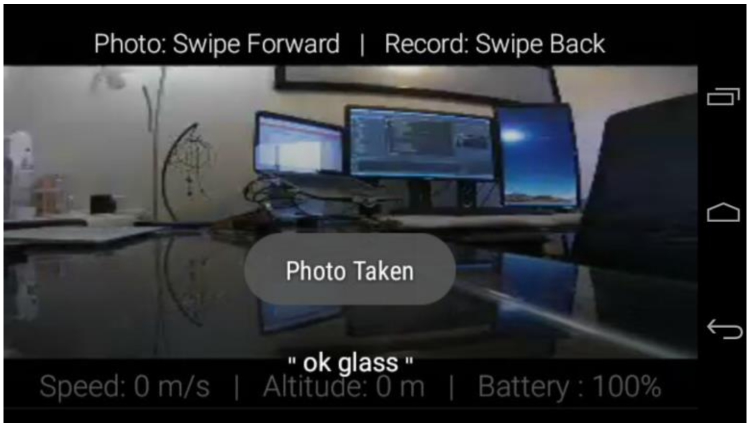


Photo: Swipe Forward | Record: Swipe Back

Photo Taken

" ok glass "

Speed: 0 m/s | Altitude: 0 m | Battery : 100%

NOTE: The app has only saved the photo to the Phantom SD card at this point.

5.2.3 How to Record Video

1: Connect Glass to Phantom
2: Navigate to Flight Menu
3: Use voice command “Ok Glass” to activate Glass listener.
4: Use voice command to start recording.
    4a: A red dot indicates a recording is in progress.

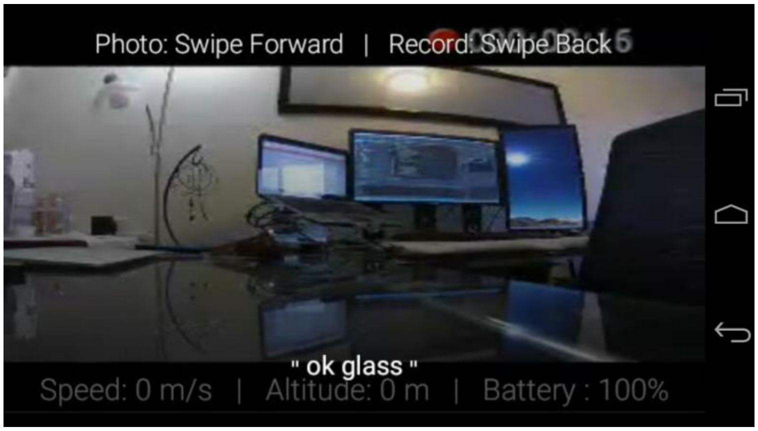


Photo: Swipe Forward | Record: Swipe Back

" ok glass "

Speed: 0 m/s | Altitude: 0 m | Battery : 100%

5: When finished, use voice command “Ok Glass” to activate Glass listener.
6: Use voice command to stop recording.

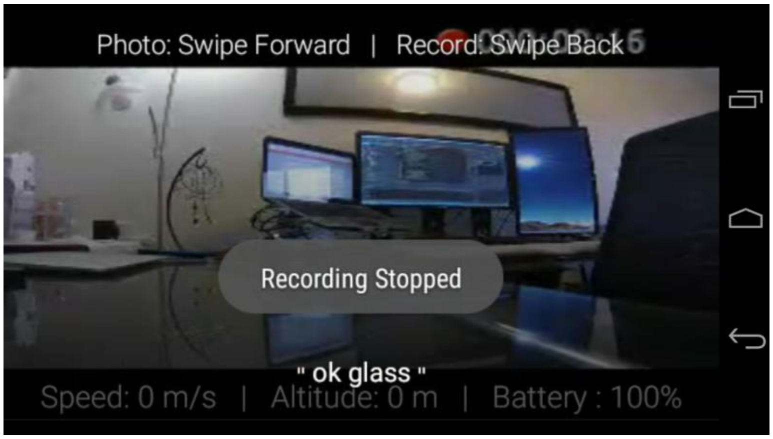


Photo: Swipe Forward | Record: Swipe Back

Recording Stopped

" ok glass "

Speed: 0 m/s | Altitude: 0 m | Battery : 100%

NOTE: The app has only saved the video to the Phantom SD card at this point.

5.2.4 How to Clear Phantom SD Card

WARNING: The application can only clear the whole SD card – you cannot delete individual files. See the data management section for tips about file transferring.

1: Connect Glass to Phantom
2: Navigate to Flight Menu
3: Use voice command “Ok Glass” to activate Glass listener.
4: Use voice command “Clear Storage”

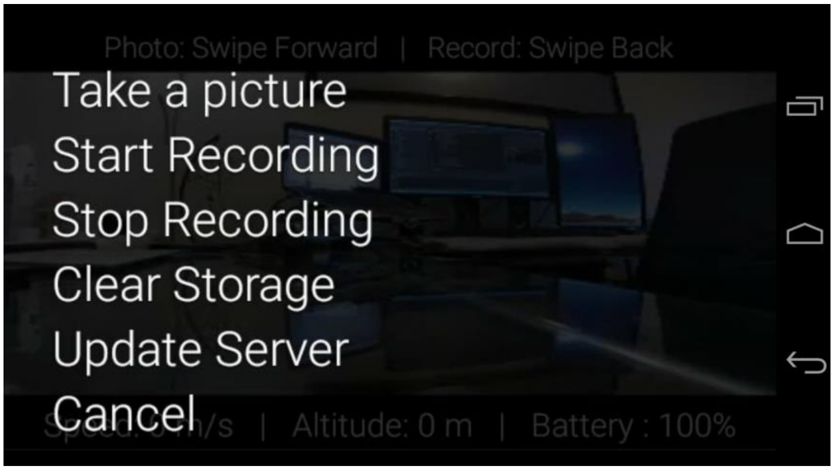


Photo: Swipe Forward | Record: Swipe Back
Take a picture
Start Recording
Stop Recording
Clear Storage
Update Server
Cancel
Speed: 0 m/s | Altitude: 0 m | Battery : 100%

5: Use voice command “Erase Media”

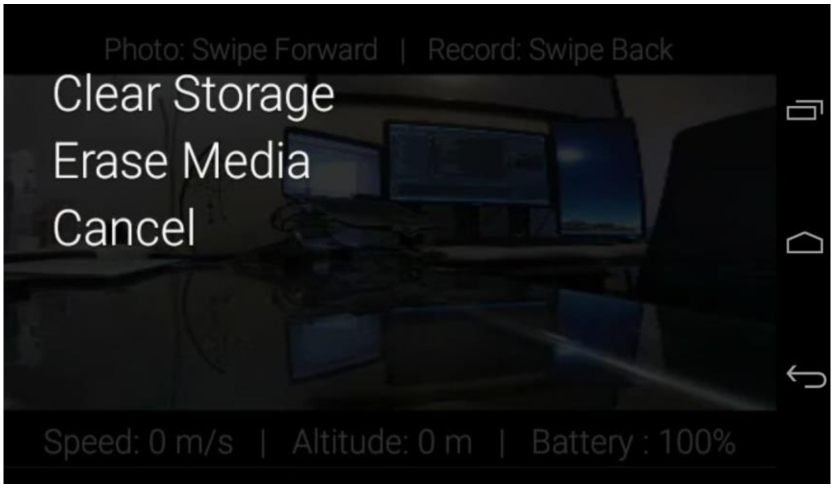


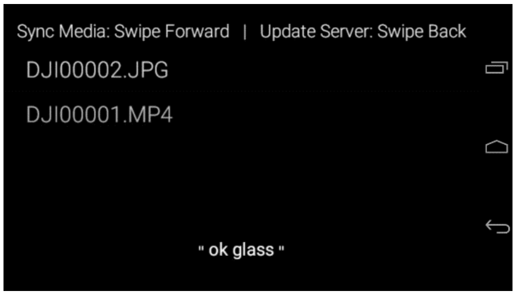
Photo: Swipe Forward | Record: Swipe Back
Clear Storage
Erase Media
Cancel
Speed: 0 m/s | Altitude: 0 m | Battery : 100%

NOTE: The app has only cleared the files on the SD card at this point.

5.2.5 How to Transfer Data from Phantom to Glass

Warning: Data transfer from Phantom to Glass will erase Phantom SD data.

1: Connect Glass to Phantom
2: Tap to go to Media Menu



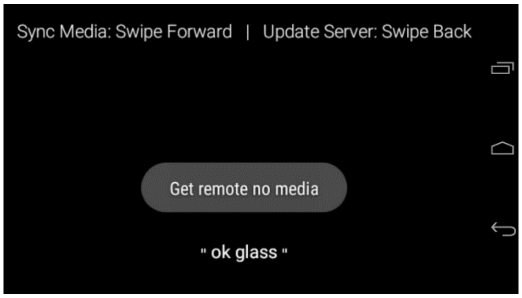
Sync Media: Swipe Forward | Update Server: Swipe Back

DJI00002.JPG

DJI00001.MP4

" ok glass "

Media Menu w/ Files



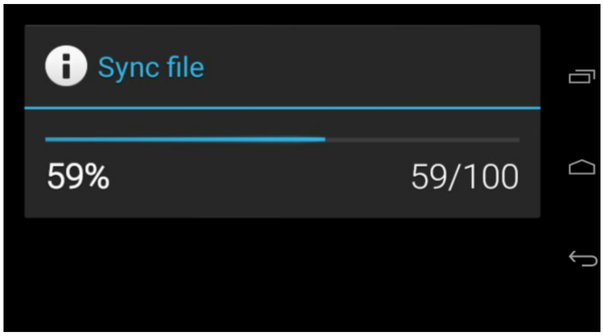
Sync Media: Swipe Forward | Update Server: Swipe Back

Get remote no media

" ok glass "

Media Menu w/o Files

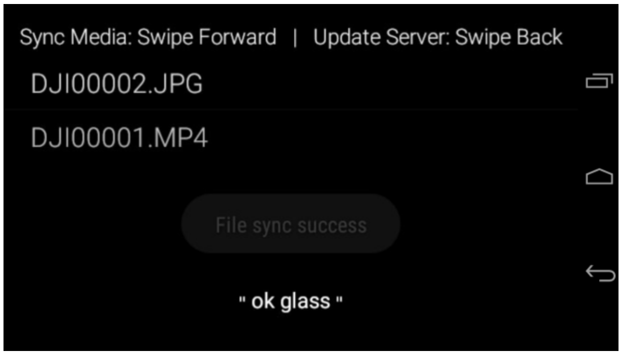
3: Swipe forward to begin file transfer.



Sync file

59% 59/100

4: Wait until file sync confirmation message.



Sync Media: Swipe Forward | Update Server: Swipe Back

DJI00002.JPG

DJI00001.MP4

File sync success

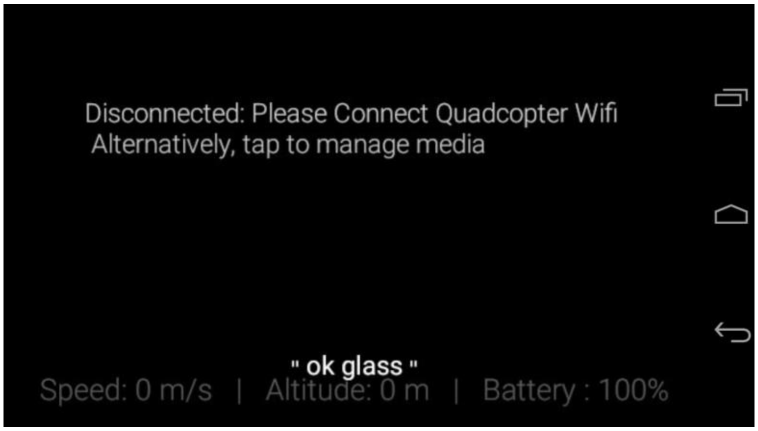
" ok glass "

5.2.6 How to Transfer Data from Glass to Website

Warning: Once a file successfully transfers from Glass to the website, the file is deleted locally on Glass. In addition, files uploaded to the sight are viewable by anyone who visits it!

1: Connect Glass to the Internet via Wi-Fi.

2: Tap to go to Media Menu



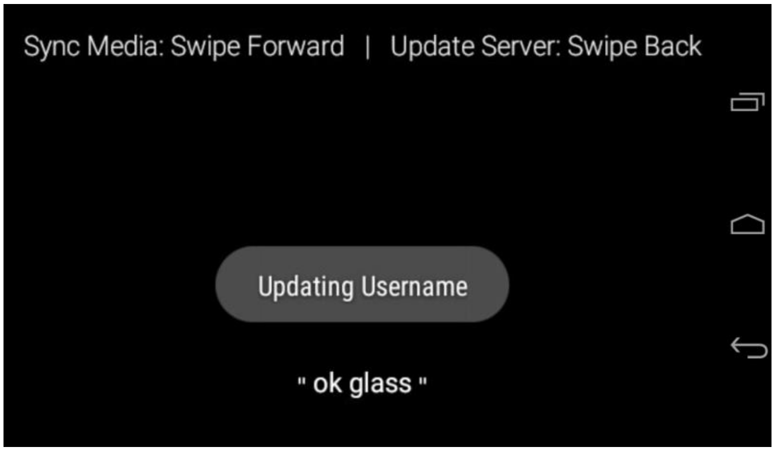
Disconnected: Please Connect Quadcopter Wifi
Alternatively, tap to manage media

" ok glass "

Speed: 0 m/s | Altitude: 0 m | Battery : 100%

3: Pair Glass with a user on the website if it isn't already.

3a: After selecting user on website, two finger tap.



Sync Media: Swipe Forward | Update Server: Swipe Back

Updating Username

" ok glass "

4: Files will be uploaded via Wi-Fi. If you lose connection then you will get an error, and need to start the upload process again.

5.3 Website

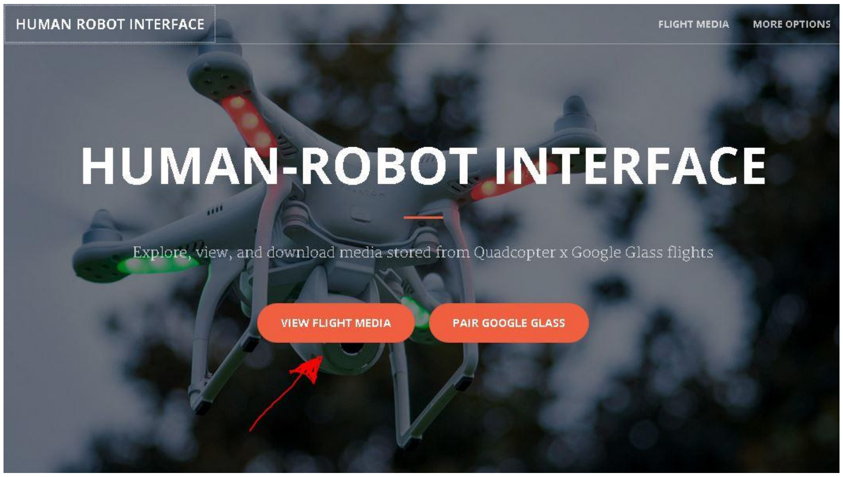
Navigating the site and using its features is very straightforward. Most users can simply follow the on screen menu options to reach their desired outcomes. For completeness, we have included several use case guides.

5.3.1 How to View Media

1: Connect to website. (You can setup your own website)

1a: Website URL: http://taglab3.genomecenter.ucdavis.edu/hri/

2: Click "View Flight Media" button.



3: Select a user.

4: Select the desired media you want to view.

4a: Right click to download the file.

5.3.2 How to Pair Google Glass

1: Connect to website. (You can setup your own website)

  1a: Website URL: http://taglab3.genomecenter.ucdavis.edu/hri/

2: Click “Pair Google Glass” Button

3: Select the desired user to pair with.

  3a: See “How to Manage Users...” to create a new user.

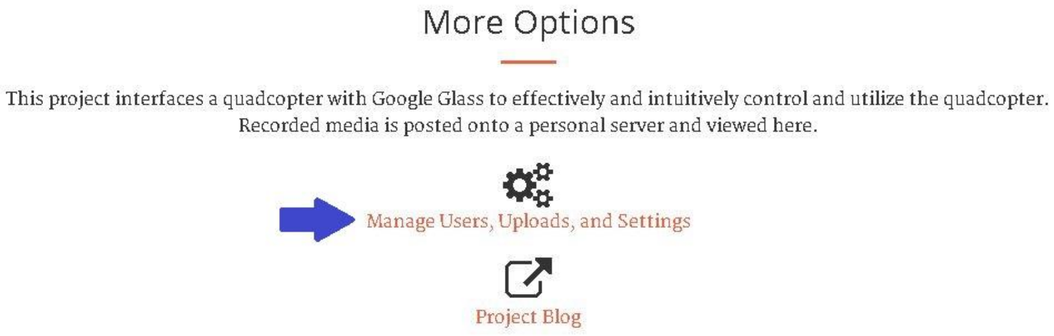
4: Click the “Update” button.

5: Launch the HRI App on Glass while connected to internet.

6: Use the two finger tap gesture in media menu to finish pairing.

5.3.3 How to Manage Users, Uploads, and Settings

1: Click the manage link at the bottom of the page.



More Options

This project interfaces a quadcopter with Google Glass to effectively and intuitively control and utilize the quadcopter. Recorded media is posted onto a personal server and viewed here.

2: Enter admin information and login.
3: Select desired option.
4: Follow on screen instructions.

5.4 Phantom 2 Vision

Turn on all components. Consult the Phantom 2 vision manual to understand how to operate the controller and quadcopter.

6.0 Troubleshooting

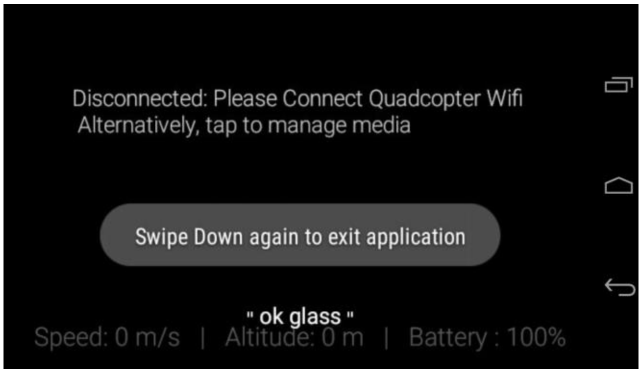
This troubleshooting section assumes each components setup was successfully completed.

General Troubleshooting Checklist

- Make sure all batteries are charged. (Glass, Quadcopter, Controller, etc)
- Make sure all components are on. (Glass, Quadcopter, Wi-Fi extender, etc)
- Make sure Glass is connected to the proper network.
    - Connect Glass to Phantom’s Wi-Fi when interacting with quadcopter.
    - Connect Glass to internet via Wi-Fi when interacting with database.
- Make sure the quadcopter is calibrated properly.

6.1 Google Glass

6.1.1 Problem: Glass is not connecting to the quadcopter.



Disconnected: Please Connect Quadcopter Wifi
Alternatively, tap to manage media

Swipe Down again to exit application

" ok glass "

Speed: 0 m/s | Altitude: 0 m | Battery : 100%

- Swipe down twice consecutively to exit the app.
- Check that Glass’s network settings is set to the quadcopter.
- Make sure the Wi-Fi extender is turned on.

6.1.2 Problem: The voice command is not working.

- Make sure you say “Ok Glass” before saying a voice command.
- Check voice command list for valid commands.
- Glass’s voice commands can be sensitive to accents. Try a different tone of speech. If the problem persists, try another control method.

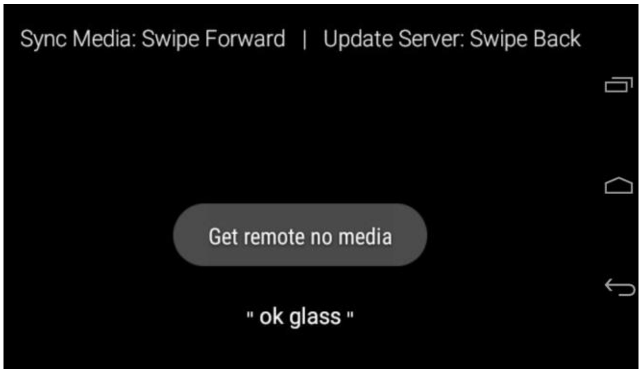
6.1.3 Problem: The video stream is not showing.

- Try navigating to the media menu, then return to the flight menu.
- Make sure Glass is connected to the Phantom Wi-Fi.
- Check that the range extender is on.
- Restart the application.

6.1.4 Problem: The application crashed or froze suddenly.

- Close unnecessary applications on Glass. Then, launch the app again.
- Do not spam commands during use.
- Report the issue to the technical support contact.

6.1.5 Problem: I took a photo/video but I don't see it in the media list.



Sync Media: Swipe Forward | Update Server: Swipe Back

Get remote no media

" ok glass "

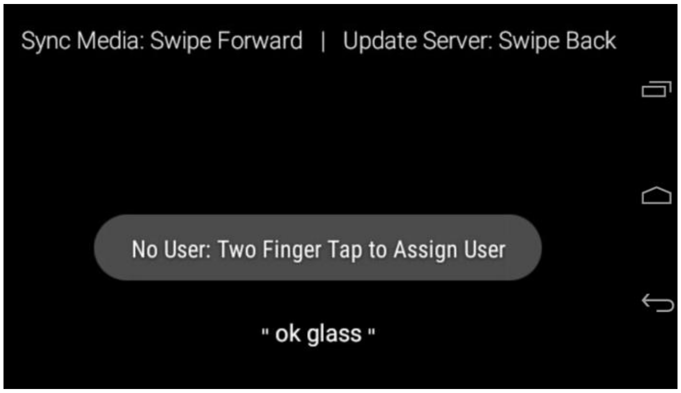
- If you uploaded to the website after creating the file, then the file was deleted from Glass.
- If you transferred the file from the Phantom to Glass, then the file was deleted on the Phantom SD card.
- Check that the SD card is properly connected.

6.1.6 Problem: When synchronizing Glass with a username the app takes me back to the launcher.

This occurs because Glass has no internet connectivity. Please connect via Wi-Fi to the internet on Glass before launching the app, and try again.

6.1.7 Problem: I see the following screen.

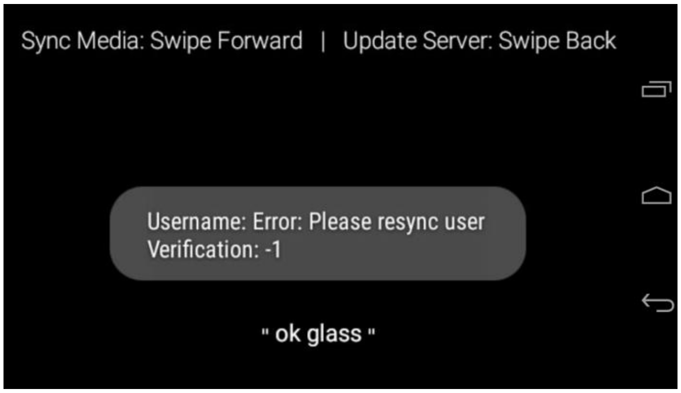
Screen 1



No User: Two Finger Tap to Assign User

Issue: This screen appears when you did not pair Glass with a user on the website before uploading to the website. To correct this, pair a user on the website with Glass and then try uploading again.

Screen 2



Username: Error: Please resync user
Verification: -1

Issue: This screen appears when the Wi-Fi connection to the internet is loss during upload to the website. Simply retry the uploading process. If it occurs again, check your Wi-Fi connection.

6.2 Website

The website is straightforward from a user point of view; any issues will display a proper message. However, the host server may run into issues, which may affect the user experience. If the on screen messages do not resolve the user's issue and the user suspects that the server is the issue, please contact technical support for the server. This is not the technical support contact for the project.

Alternatively, an individual can turn debug mode on in the settings.py file on the server and get a detailed description of the error. However, only advanced users familiar with the server, Django, and have permission will be able to do this.

For help understanding the website code and settings, please contact technical support in the contact section of this project.

6.3 Phantom Vision 2

For troubleshooting procedures for the Phantom Vision 2 quadcopter and controller, please consult the manuals DJI provides. You can find these at their website dji.com/product/phantom-2-vision/download.

7.0 FAQ

Q: Does the HRI project work with other UAS?
A: No, the HRI project currently only works with the Phantom Vision 2 and Phantom Vision 2 +.

Q: Do I need anything special to develop the Glass app?
A: Yes, you will need to apply for a Developer license with DJI. Visit their developer website for details and different licenses offered.

Q: How long does the battery life of ____ last?
A: Glass has the lowest battery duration with around 10 minutes. For information about the other products and factors, please visit the appropriate product website.

Q: What happens if the connection between the controller and quadcopter is interrupted?
A: The Phantom automatically enters a hover state and drops to a set altitude upon lost connection.

Q: What are the laws about flying UAS?
A: This varies depending on location. Please consult your government's statutes and guidelines for details..

Q: Is it acceptable to lose site of the UAS since I can see its video feed with Glass?
A: In general, no. You must maintain a line of site with the UAS. Please consult your government's statutes and guidelines for details.

Q: How can I further develop this project?
A: There are two code repositories for this project. The website code is available through GitHub. Simply clone this repository and follow the environment setup section to get started.

The Glass app code is currently private due to terms of use with DJI's SDK. Contact a project member to find out more about obtaining the Glass app source code. Once the source code is obtained, you will need to get a developer's license from DJI. DJI offers several tiers that unlock more features of the SDK the higher the tier.

DJI's SDK and Google Glass do not provide emulators for testing. Therefore, only individuals with both Google Glass and a supported Phantom product will be able to properly test.

8.0 Contact Information and Support

When contacting the project support contacts, please identify the project as Human-Robot Interface.

General Questions (Project)
Laurademesa@gmail.com

jmsalanga@ucdavis.edu

danielchen93@gmail.com

sgrodriguez@hotmail.com

Technical Support (Project)
danielchen93@gmail.com

DJI Support Site
www.dji.com/support

Phantom 2 Vision Setup Support
wiki.dji.com/en/index.php/Phantom_2_Vision-Setup_Wizard

Google Glass Support Site
support.google.com/glass/?hl=en#topic=4363345

Project Blog (has links to resources)
https://humanrobotinterface.wordpress.com/

9.0 Appendix

Included items (inorder):

- Design Document
- Testing Document

9.1 Design Document

Human-Robot Interface: Efficient Control and Flight Data Display of a Quadcopter through Google Glass

Design Document v3

June 1, 2015

Prepared By:	Prepared For:
Daniel Chen	ECS193A/B
Laura Demesa	Client:
Steven Rodriguez	Dr. Ilias Tagkopoulos
Justin Salanga	

Table of Contents

Introduction **2**

Technology Background **2**

Requirements 2

Google Glass 2

DJI Phantom Vision 2 3

Design Goals **4**

Design Overview **5**

Project Tasks and Flow **6**

Communications **7**

UAV and Glass 7

Glass and Database 8

Data **9**

Data Transfer 9

Data Storage 10

Controls **11**

Gimbal Controls 11

Camera Controls 11

Interface **12**

Glass UI 12

Database UI 12

Additional Notes **13**

Introduction

The Human-Robot Interface project address the desire to have a more hands-free experience when controlling an unmanned aerial vehicle (UAV). Modern UAV technology requires the use of physical devices to control not only the UAV but also attachments such as a camera. This project reduces the need for UAV operators to use their hands when controlling the camera and storing flight information to a database. The project accomplishes this by using Google Glass's motion controls and wireless connectivity.

This document details the technology required and software design. The project's design consist of four categories: communications, data, controls, and interface. Each segment explains the design choices, decisions, and implementation.

Technology Background

Requirements

The human-robot interface project has two technology requirements. First, the device to interface with the UAV is Google Glass. Second, the UAV must be a programmable quadcopter capable of providing flight data and connecting with Google Glass.

Google Glass

Google Glass is a wearable electronic device that fits similar to a pair of glasses. Google Glass's display screen rest slightly above eye level in front of one eye. Google Glass has these additional tech specs:

- Microphone
- Accelerometer
- Wi-Fi - 802.11 b/g 2.4GHz
- Bluetooth
- 16 GB Flash Memory
- GPS
- Can be paired with an Android (4.0.3 or later) or iOS (7 or later) device.

Programming applications for Google Glass requires the Glass Development Kit, which is an additional package for Android Studio. Developing for Glass is similar to Android development, which is a combination of Java and XML. However, Glass's user interface has a specific structure that differs from the typical Android device.

Glass introduces the concept of timeline, which manages processes called cards that can be live or static. Timeline allows the user to switch between different live cards, which update frequently and have access to low-level sensor data, and static cards, which rarely, if at all, update. Switching between processes in the timeline is analogous to tabbing through program windows on typical operating systems.

In order for a process to have full control, Glass uses the concept immersion, which runs a process outside the timeline and allows developers the ability to create their own UI and process the entire user input.

In addition to running native applications on Glass, users can install regular android software in a process known as sideloading. This requires access to the Glass debug bridge, where a user can manually load files to Glass, and the compiled .apk Android program. However, sideloading software does not guarantee compatibility with Glass. Therefore, undesired results may occur when using this method.

DJI Phantom 2 Vision

The Human-Robot Interface project uses the Phantom 2 Vision quadcopter made by DJI as its UAV. We selected the Phantom because it met the programmability and connectivity requirements of the project. We evaluated alternative quadcopters, but many did not meet the programmability requirement. Prior to selecting the Phantom, we used Parrot's Bebop quadcopter; however, the quadcopter did not perform as described and lacked critical developer documentation. DJI offers a software development kit (SDK) to help developers.

The Phantom 2 Vision transmits its own Wi-Fi signal to a controller, which then extends the signal for other devices to connect. This Wi-Fi extender is how Google Glass and other devices can connect.

Additional Phantom specs:

- 14 Megapixel camera
- 4 GB SD card
- GPS
- 3-axis gimbal
- 25 Mins flight time

DJI restricts their SDK by application approval and offers several tiers of access. The first tier offers video uplink, gimbal control, camera controls, flight OSD, and battery. The second tier offers all the features of the first tier and flight controls. The second tier may require payment to use. However, this project only uses the first tier.

Design Goals

The Human-Robot Interface project has the following design goals:

1. Establish a connection between the Phantom quadcopter and Google Glass.
2. Transfer video from the UAV to the Google Glass display.
3. Transfer flight data from the UAV to Google Glass.
4. Store the flight data from Google Glass to a database.
5. Implement Google Glass head tracking to control the UAV’s gimbal.
6. Implement voice commands that Google Glass will transfer to the UAV.
7. Create an easy to use user-interface.

Design goals are listed in order of significance. Goals 6 and 7 are considered low priority and not critical to the core project requirements. However, the addition of these features creates a better application overall. In addition, goal four’s design is tentative and likely to change.

Design Overview

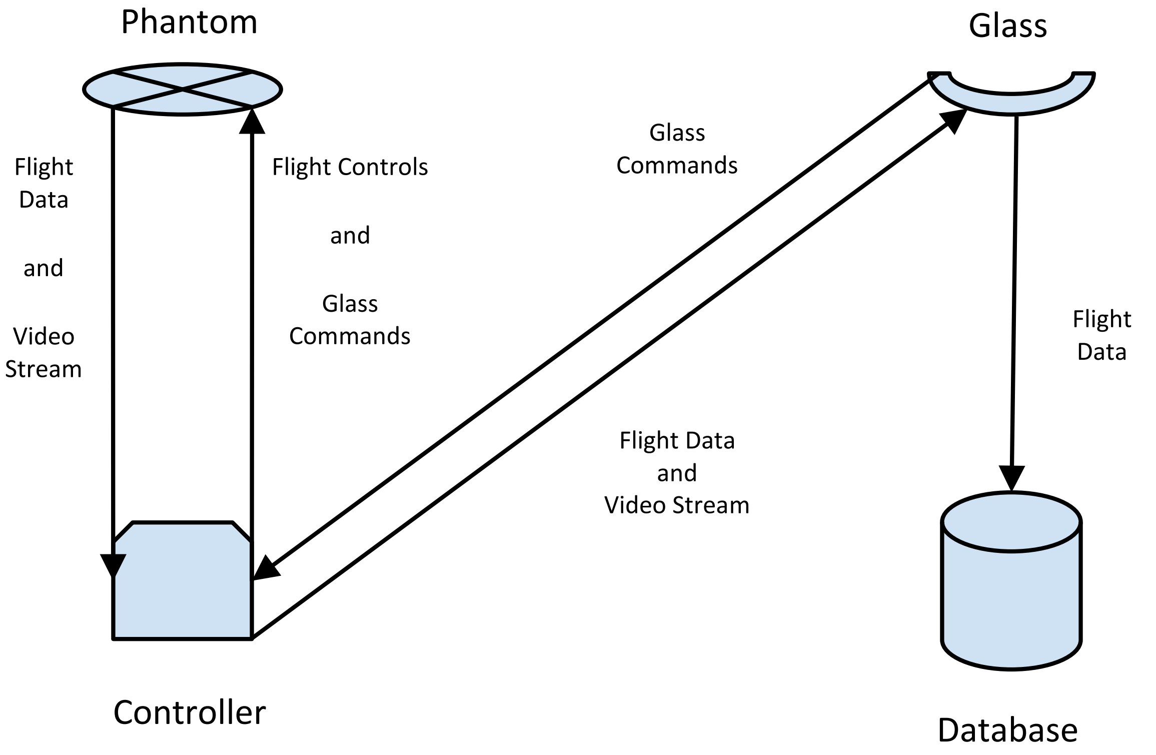

Phantom
Glass
Flight Data and Video Stream
Flight Controls and Glass Commands
Glass Commands
Flight Data and Video Stream
Flight Data
Controller
Database

Figure 1: Design Overview

Figure 1 illustrates the overall communication and data transfer for the project. Note that the controller is an intermediate to the communication between the Phantom and Glass, which is done through the Wi-Fi extender. All communication is through wireless connections.

The project application is a native android application (i.e. .apk) that is sideloaded onto Glass, due to the limitations of the GDK.

Project Tasks and Flow

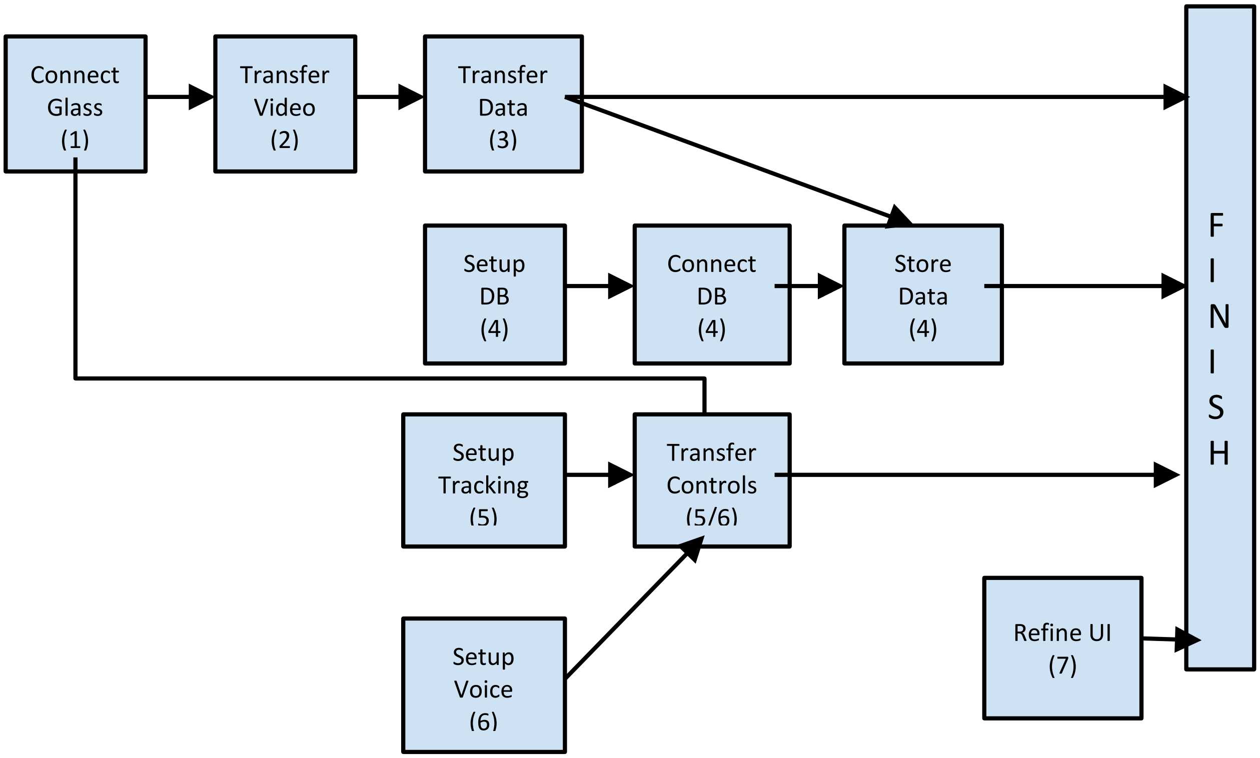


Figure 2: Project Flow

Figure 2 illustrates the project's tasks and order of completion. The numbers represent the order of completion. The arrows show dependencies in the project — a task began once all prior tasks before it were completed.

Communications

This project requires establishing communication between the several devices involved. The connection between the Phantom and the controller is a part of the UAV product and does not need modifying. However, the connection between the Phantom drone and Glass is a project task. In addition, a connection between Glass and a database system is another task for this project.

Drone and Glass

Our project connects Google Glass to the Phantom drone through the controller transmitter. The connection type between the drone and controller is 5.8 GHz Wi-Fi, which is then transmitted to Glass through 2.8Ghz Wi-Fi. Figure 3 illustrates this architecture.

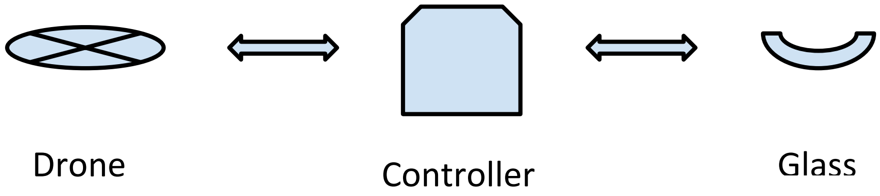


Figure 3: Drone/Glass Communication

Glass must detect the controller to connect Glass to the drone; Google already implements this requirement. On Glass navigating to the Wi-Fi settings and selecting the Phantom’s Wi-Fi establishes a connection.

Once the program establishes a connection with the drone, Glass communicates to Phantom through application programming interface (API) calls. These are functions determined by DJI and returns data values, check values, and initiate commands. The program uses API calls to control the gimbal, transfer video and flight data, and issue camera commands.

There are no other practical ways to connect the Phantom and Glass.

Glass and Database

The setup of a database on a server and connecting Glass to the database is a task for this project. The database will store flight data transferred from Glass. The database software used is MySQL, and the Web Page uses Django.

Figure 4 illustrates the current design for database communication.

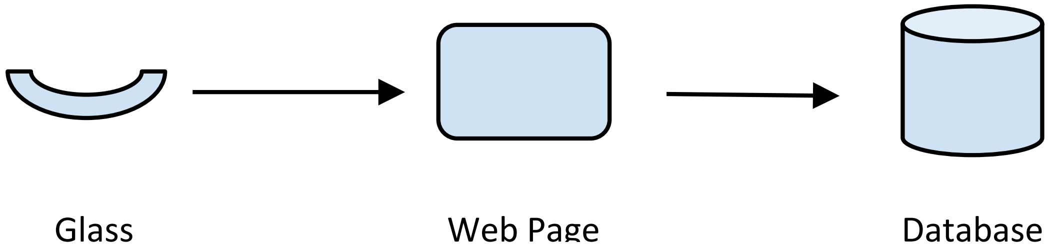


Figure 4: Glass/Database communication

Django is a python based web framework that follows the MVC architecture model. For this project, we used the RESTful Framework package to allow the website to handle HTTP keyword requests.

The model-view-controller architecture segregates the data representations (model), display methods (view), and alteration methods (controller). Django has its models in models.py, views are templates.py, and controllers are views.py. Django has a high level of abstraction in its framework to allow quick and easy programming.

Data

There are several types of data involved in this project: flight data, video data, and image data. Flight data consist of GPS coordinates, drone speed, altitude, and error codes. Video data is 1080p/30fps HD video in the ".MP4" file format. Image data is RAW or JPEG file formats. The transferring and storing of this data is a project task. In addition, a database must be set up for long-term storage.

Data Transfer

The program transfers data from the quadcopter to Glass through API calls. Once transferred to Glass, the data will remain locally on Glass's memory until Glass connects to the internet, but the transfer deletes the files on the Phantom's SD card. Then upon connecting to the internet, Glass will automatically upload the data to the database, and delete the file on Glass upon successful upload. Figure 5 illustrates this design.

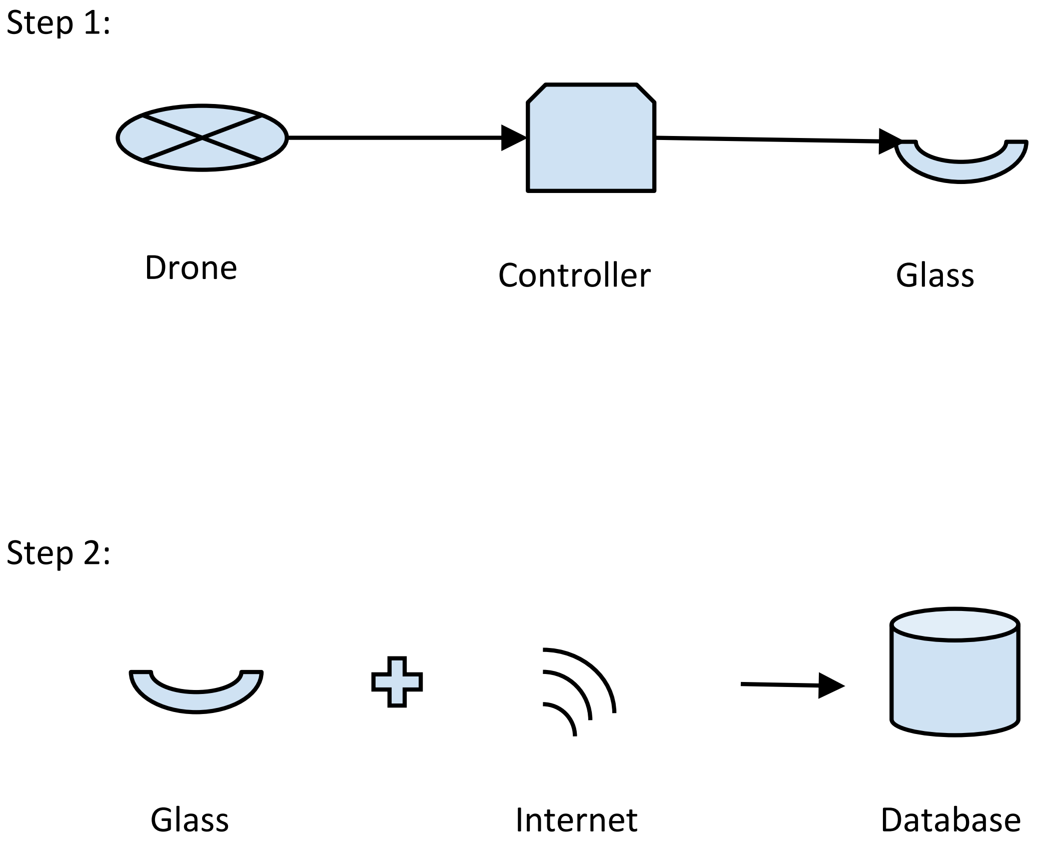


Figure 5: Data Transfer

Data transfer is designed in a two-step process because Glass can only connect to one network at a time. Glass has about 10GB of storage, so a single flight should be completely storable until Glass is able to connect to the internet. However, a user must consider memory limitations when using the drone for multiple flights before transferring data to the database.

To transfer the data to the database the user must use a traditional computer to connect to the website and setup a paring between Glass and a user. Once this is done, Glass can sync its media files with the database. The administrator interface for the website has the ability to manage users and files.

Data Storage

When the data is stored on Glass the data is in appropriate files. For example, the video data is a ".MP4" file.

The database is a relational database with the following entry formats.

User data format: (ID, username, active)
File data format: (ID, user, uploaded_file)

Django automatically adds the ID key to each model. The user field in File is a foreignkey link to a specific User. This allows a one to many relation from User to File.

Controls

There are two types of controls that this project implements. The gimbal controls for the camera and the camera controls. The controller that comes with the drone handles the controls for the drone.

Gimbal Controls

Google Glass offers head tracking, which provides values based on the tilt and positioning of a person's head. Glass's sensors determine the values and have little to do with a person's actual head. The program translates the values obtained from Glass to values for the API calls for the gimbal. For example, if a person tilts there head down 5 degrees then the corresponding gimbal API call may be for a tilt of 2 degrees. Controlling the gimbal of Glass is just a matter of collecting the sensor data and translating it into gimbal values for the API call.

Camera Controls

To create a more hands free experience voice commands can make taking pictures easy. This requires setting up a phrase command, which recognizes spoken words and executes its corresponding code. Figure 6 illustrates how camera controls work.

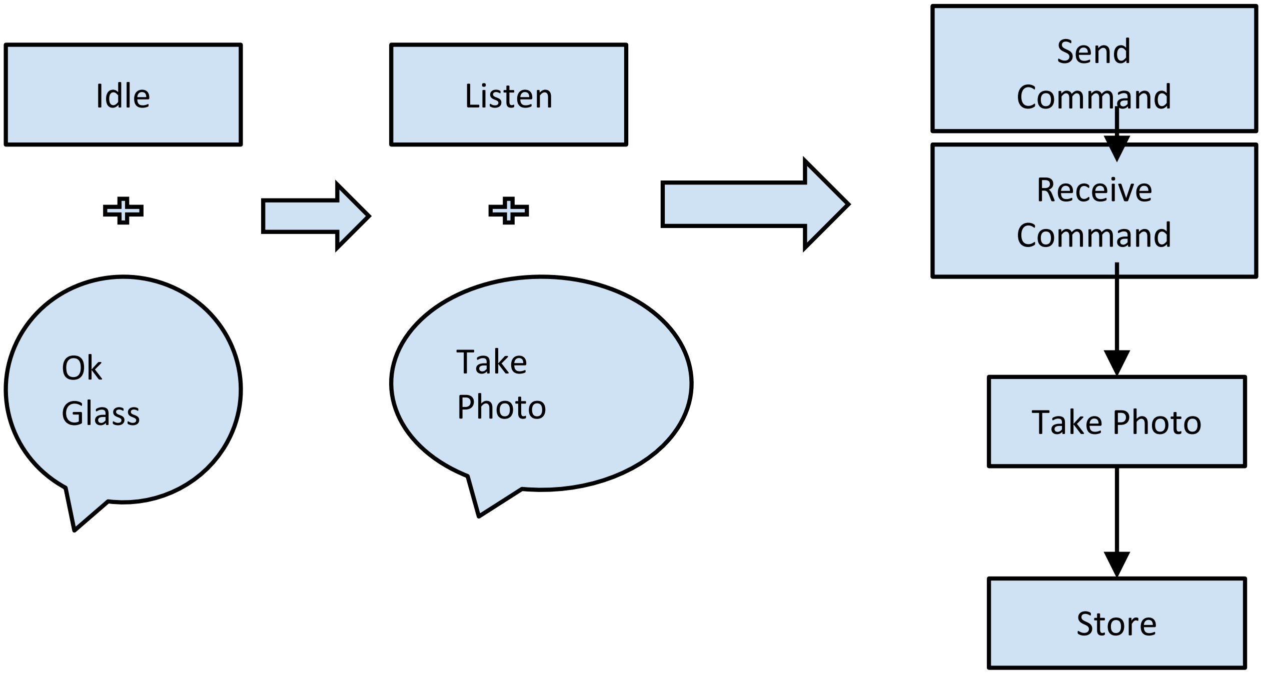


Figure 6: Photo Command Outline

Interface

There are three main interaction points for the user: Google Glass, the database, and the drone controller. Each has a distinctive interface preference. This project requires the creation of the database and Glass user interfaces.

Glass UI

Google suggest a simplistic user interface for Glass programs. The following images illustrate the current interface design.

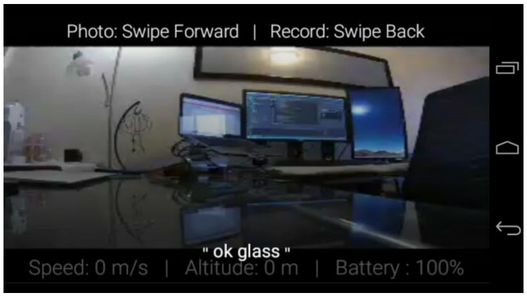


Photo: Swipe Forward | Record: Swipe Back
" ok glass "
Speed: 0 m/s | Altitude: 0 m | Battery : 100%

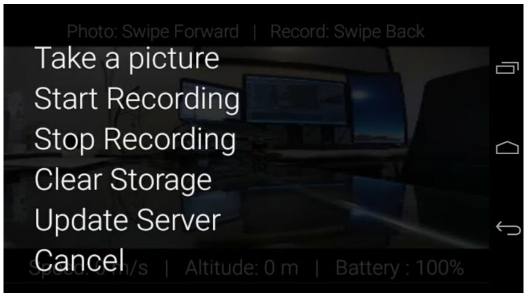
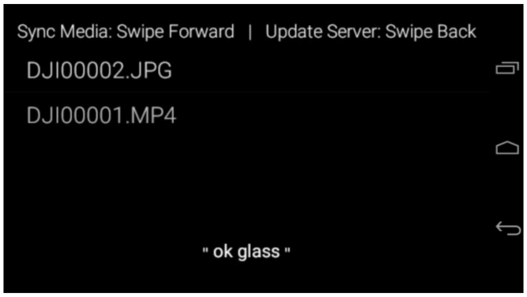
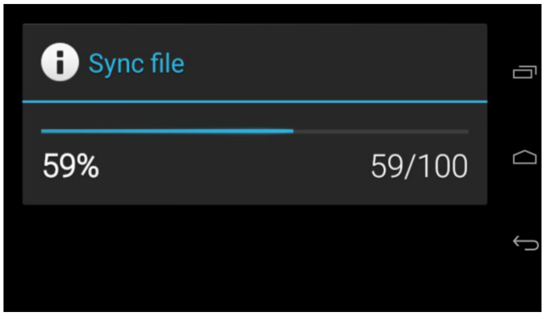


Photo: Swipe Forward | Record: Swipe Back
Take a picture
Start Recording
Stop Recording
Clear Storage
Update Server
Cancel
Speed: 0 m/s | Altitude: 0 m | Battery : 100%





The Glass UI breaks down into two activities. The first is the flight menu, where a user can take photos, record video, and view the live stream. The second is the media menu, where a user transfers data from the Phantom to Glass and from Glass to the website.

Depending on which network Glass is connected to, the type of transfer will differ. A Phantom connection will allow Phantom to Glass, and an internet connect will allow Glass to the website.

In order to mimic a Glass like UI, several overrides were made to allow swipes and taps.

In addition, to prevent Glass from sleeping, wakelocks keep Glass active until the application cloes.

Database UI

The database is accessed through the website. The website has no authentication for viewing the content. This is because Glass does not allow an easy way to enter passwords. However, to manage users and settings of the website, admin login information is required.

When viewing the website media files are linked to the user that uploaded them. First selecting a user is necessary to view their files. Next, selecting a file allows the user to view photos and videos. The media files are sorted alphabetically.

In addition, to viewing the database, the website provides a pairing option for Glass. Paring tells the server which user is uploading from Glass. The paring process requires a traditional computer to connect to the website side of the paring, and Glass to confirm the paring. Once paired, Glass can upload to the website.

Additional Notes

- The Phantom has flight zone restrictions such as major airports. Entering these zones is not possible. You can find more information at the DJI website.

- Sideloaded android applications on Glass do not run as expected. Returning to the timeline does not put the application in the background rather it exits the program. In addition, launching a program does not always bring the program into view. We recommend a special android application launcher to resolve these issues.

- DJI might update both its SDK and drone firmware. This may introduce bugs into the application.

- DJI issues an application key for developers to associate with their program. If DJI invalidates this key then developing this program further may not be possible without obtaining a new key.

9.2 Testing Document

Human-Robot Interface with Phantom Vision Quadcopter Test Plan.

ECS193ABeasts

Prepared by Justin Salanga, Daniel Chen, Laura Demesa, and Steven Rodriguez
For our client: Dr. Ilias Tagkopoulos

Summary of Product(s) to be Tested:

- Interface a Quadcopter with Google Glass to aid in the search for missing children/animals with aerial live-feed.
- Quadcopter: Phantom 2 Vision
- Wearable: Google Glass
- Additional: Phantom 2 Controller, Android Device, Phantom SD Card
- Developers:
  - Justin Salanga: jmsalanga@ucdavis.edu
  - Daniel Chen: danielchen93@gmail.com
  - Laura Demesa: laurademesa@gmail.com
  - Steven Rodriguez: sgrodriguez@ucdavis.edu

Resources Required for Testing:

- Hardware:
  - Phantom 2 Vision and Phantom 2 Controller
  - Google Glass
  - Android Device
- Software:
  - Get_Start-release.apk
  - Android SDK (included with Android Studio)
  - Android Debug Bridge
- Connections: Google Glass and Android device connected to Phantom personal network.
- Estimated Hours for Testing: 2 hours.
- Resources can be obtained from Dr. Tagkopoulos, or one of the developers (See above for Contact Information)

Packaging, Building, Configuration

- Bitbucket was used for this project’s development: https://bitbucket.org/dchen93/ecs193code
    - Please contact one of the developers to gain access.
- For the apk: pull from ECS193code/Daniel-Sideloaded autoplaying android app/DJI_SDK_Android_GetStart/Demo/Get_Start/build/outputs/apk

Installing the APK to Google Glass From Windows:

- Once the apk is downloaded, move the file to your ADB directory
- Open cmd.exe and change directory to where the apk is saved to.
- Make sure no other Android device is connected to your computer.
- Connect the Google Glass with the appropriate USB cable.
- To test if your Glass is connected, enter “adb devices” and you should see your Glass ADB number and the status “device” as an output.
- To install the application to the glass, input the following command: “adb install <application.apk>”
    - Be sure to replace the name with the most current APK release.

Process for Defect Reporting & Repair

Bug Reports from the Testing Group will be responded to within 5 hours between the hours of 8:00am and 10:00pm. The response will be an acknowledgement of the problem, with a variable amount of time until the problem is fixed (depending on the severity of the bug).

When a Bug Report is fixed, the developers will push the newly updated program on bitbucket and contact the Testing Team, who can now pull the new apk described above.

Functional Testing Plan

User Identification	User Goal	Test ID	Estimated Time (Dev)	Estimated Time (Test)
Admin/Sub User	Connect Google Glass to Phantom			≤ 1 Hour
Admin/Sub User	Take Photo	1		≤ 1 Hour
Admin/Sub User	Take Video	2		≤ 1 Hour
Admin/Sub User	Save to Database	3		≤ 1 Hour
Admin/Sub User	View Database	4		1-2 Hours
Admin User	Change Database	5		1-2 Hours

Non-Functional Testing Plan

Category	Subcategory	Specific Goal	Use Case Index	Est. Time (Dev)	Est. Time (Test)
Security	Access Control	Success on saving or viewing photos	1, 2		
Performance	Photo Quality	How is the photo quality while stationary, in motion, or hovering?	1, 2		
Reliability	[1] Database Reliability	Is the Glass successfully saving onto the database?	3, 4		
Reliability	[2] Database Reliability	Can you successfully retrieve saved pictures and videos from database?	4		

Use Cases

Use Case ID	1
Goal	Take a Picture
Scope	External
Pre-Conditions	Google Glass has installed APK Phantom Vision Quadcopter and Controller functional.
Success State	Video Live-Feed takes a screenshot, Image saved in SD Card
Failed State	Nothing is saved in SD Card
Start Action	User connects Glass to Phantom Network
Description	1. User opens Get Start Application 2. App transmits Live Feed from Quadcopter to Google Glass UI 3. User swipes forward on touchpad 4. App screenshots Live Feed. 5. App saves image on SD Card. 6. App transmits message “Photo Taken”wd
Variations	1. SD Card Full/Missing/Error Read 2. Connection distance too far
Extensions	1. Phantom can be in flight or stationary.

Use Case ID	2
Goal	Take a Video
Scope	External
Pre-Conditions	Google Glass has installed APK Phantom Vision Quadcopter and Controller functional.
Success State	Video Live-Feed takes a starts and stops, saved in SD Card
Failed State	Nothing is saved in SD Card
Start Action	User connects Glass to Phantom Network
Description	1. User opens Get Start Application 2. App transmits Live Feed from Quadcopter to Google Glass UI 3. User double taps on touchpad 4. App starts video recording. 5. User double taps on touchpad. 6. App ends video recording. 7. App saves video on SD Card. 8. App transmits message “Video Saved”
Variations	1. SD Card Full/Missing/Error Read 2. Connection distance too far
Extensions	1. Phantom can be in flight or stationary.

Use Case ID	3
Goal	Save to Database
Scope	Internal
Pre-Conditions	Google Glass has installed APK Database running, not under maintenance.
Success State	Data on Application moved to database. Application Data is deleted.
Failed State	No change in Application Data or database.
Start Action	User connects Glass to WiFi with Internet.
Description	1. User opens Get Start Application 2. User swipes back on touchpad 3. Application checks internet connection. 4. Application checks database status. 5. Application copies pictures and videos to database in a new folder. 6. Application clears successfully saved pictures and videos. 7. Application transmits message: “Images and Videos successfully uploaded”
Variations	1. Corrupted images/videos. 2. Internet connection lost during saving.
Extensions	1. Select which to upload and which to not.

Use Case ID	4
Goal	View Database
Scope	External
Pre-Conditions	User has credentials to access database. Database running, not under maintenance.
Success State	User can view pictures and videos
Failed State	User cannot view pictures or videos.
Start Action	User connects to the database.
Description	1. User enters credentials. 2. Application pulls images and videos from database 3. Application displays successfully pulled data.
Variations	3. Corrupted images/videos. 4. Internet connection lost during saving.
Extensions	1. Select which to upload and which to not.

Use Case ID	5
Goal	Change Database
Scope	External
Pre-Conditions	User has admin credentials to access database. Database running, not under maintenance.
Success State	User can move/delete/rename pictures and videos
Failed State	User cannot move/delete/rename pictures or videos.
Start Action	Case 4 success.
Description	Move: 1. User drags file from one folder to another. 2. Application copies file to new location. Delete: 1. User chooses file(s)/folder(s). 2. Selects Delete. 3. Application removes/flags selected for deletion. Rename: 1. User chooses file/folder. 2. Selects Rename. 3. User enters new name. 4. Application changes names and references to selected name.
Variations	1. Corrupted images/videos. 2. Internet connection lost during changing.
Extensions	1. Select which to upload and which to not.