USER MANUAL

A COMPREHENSIVE GUIDE TO OPERATION AND USABILITY

TABLE OF CONTENTS

Table of Figures

Introduction

Surveilia is an autonomous drone capable of human detection. It can be operated both manually and autonomously. Surveilia was designed to be repairable and requires several interfaces to establish connectivity. As such, it is expected that users have some level of technical ability.

Automation is handled in the Kakute F7 flight controller using ArduPilot Mission Planner software. Mission planner allows users to configure patrols suited to the site's requirements. Surveilia can also be manually controlled using the FS-i6 2.4 GHz RC transmitter and FS-A8S 2.4 GHz receiver.

Human recognition is handled by the Raspberry Pi Zero Wireless and communicated back to ground station software by UDP communications. Surveilia makes use of the HOG descriptor algorithm for identifying humans in the image frame. When the Raspberry Pi recognizes that a human is in its image, an image is sent to the ground station with an alert so the user can inspect the encounter.

Tools and Equipment

Fasteners:

- M3 Screws (x48)
- M3 Nuts (x8)
- M3 Standoffs (x8)
- M5 Screws (x4)
- Zap Straps (x7)

Tools:

- Socket Set (2.5mm and 2mm)
- Wrench (8mm)
- LiPo Charger (1s-6s)
- Soldering Iron
- Soldering Essentials (Solder Paste, Alcohol, K-Wipes, Third Hand)
- Micro USB Cable
- Micro SD Card 32GB+
- FS-i6 RC Transmitter (2.4GHz)
- 3D Printer (Ultimaker 3 and/or Ender 3)
- Filament (PETG and/or PLA)

Electronics:

- Kakute F7 V1.5 Flight Controller

- Skystars KO 4 – in – 1 ESC 45A
- Flysky FS-A8S 2.4GHz RC receiver
- BN-220 GPS
- 1750KV T-MOTOR VELOX VELOCE V2207.5 V2 (x4)
- Raspberry Pi Zero W
- Raspberry Pi 8 Megapixel Camera
- DHT11 Humidity and Temperature Sensor
- MPU6050 Gyroscope and Accelerometer
- Personal Computer running Windows 10 OS
- 1500mAh 6s LiPo Battery
- 18650 3000mAh Battery x 2
- DC Voltage Regulator 3A 5V Output

GROUND STATION

The Ground station is a desktop application used for users of Surveilia to track their mission's information. It displays updated information regarding the environment of the drone mid-flight, including the temperature, and the gyroscope's information. This data can be used to fine tune the drone's stabilization through corners, wind interference, and differences in ground elevation.

DOWNLOADING AND INSTALLING GROUND STATION

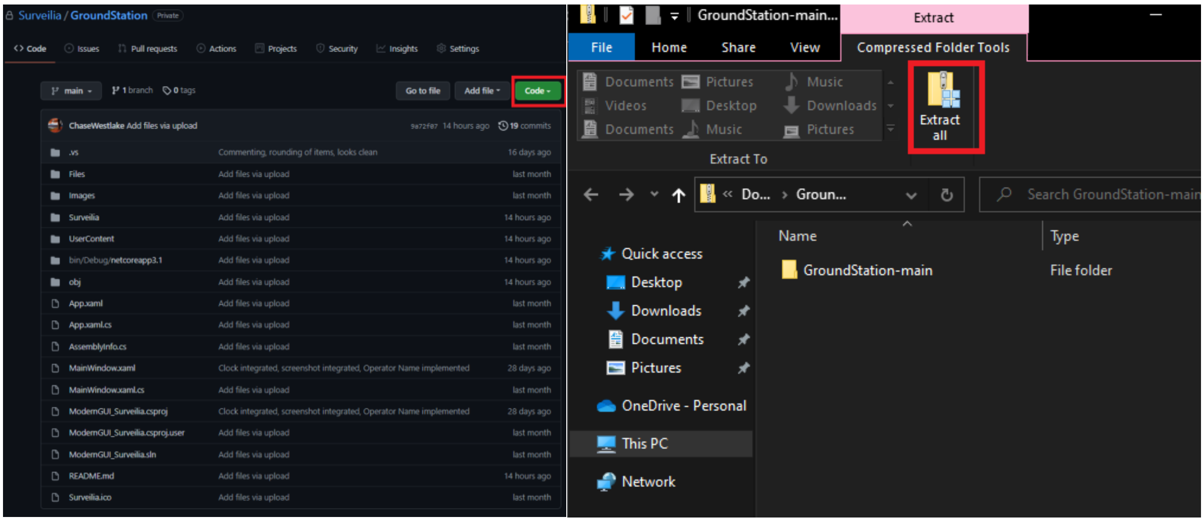

Figure 1 Download and extract Ground station software from GitHub

1. Download the ground station at https://github.com/Surveilia/GroundStation.
2. Ground station can be downloaded by selecting **Code** followed by "Download ZIP" when the drop-down menu appears.

3. When the software is downloaded, extract the contents into a suitable directory

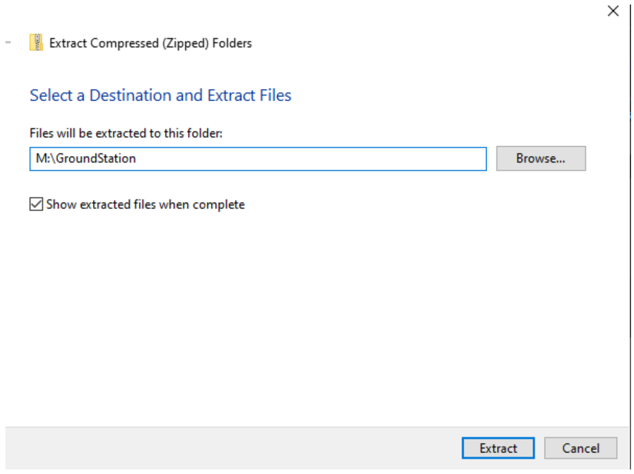

Figure 2 Extract contents of the ground station zip file into an appropriate

4. Once the file has been extracted, find the "Surveilia" Folder and copy it into the C: drive

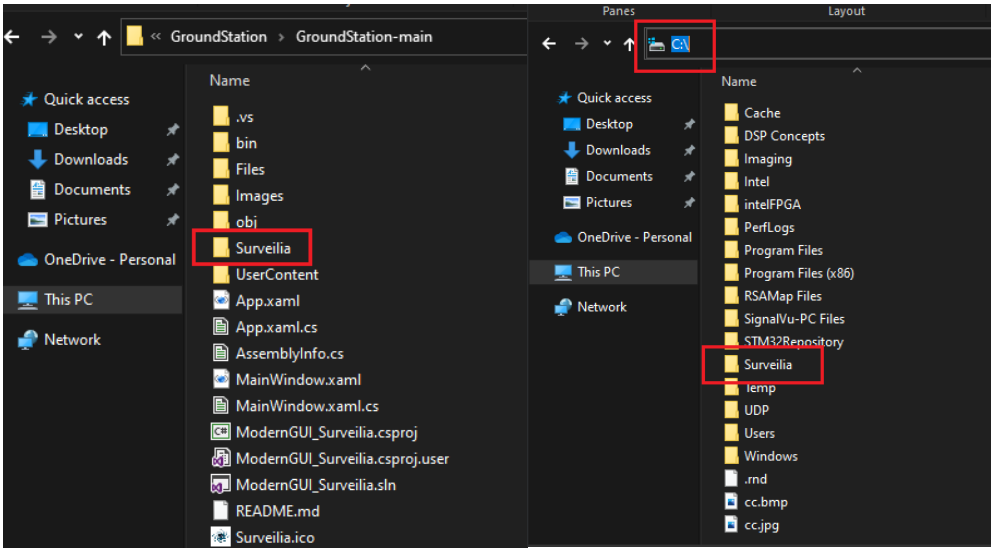

Figure 3 Move the Surveilia file over to the C drive

Navigating the Ground Station

The ground station is intuitive software that helps the user of Surveilia track flight data and receive imaging when a person is detected on the flight. The ground station has 4 primary tabs: Home, Stats, Help, and User menu.

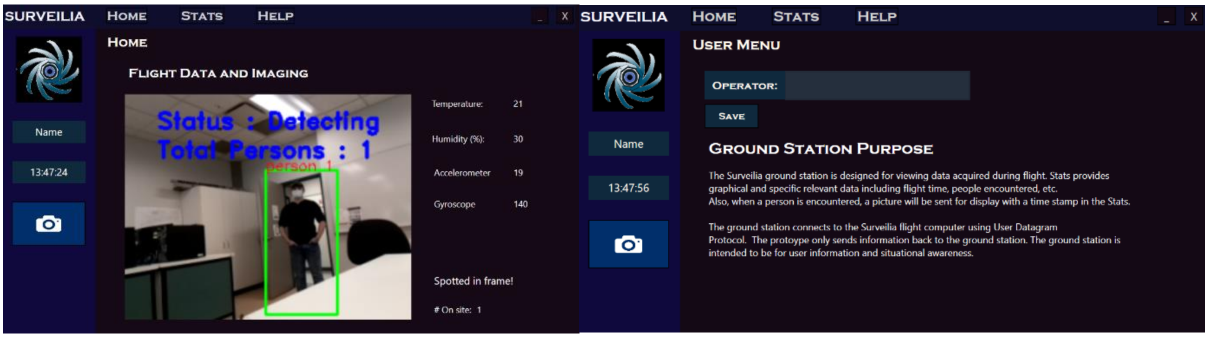

Figure 4 Ground station home and user menu

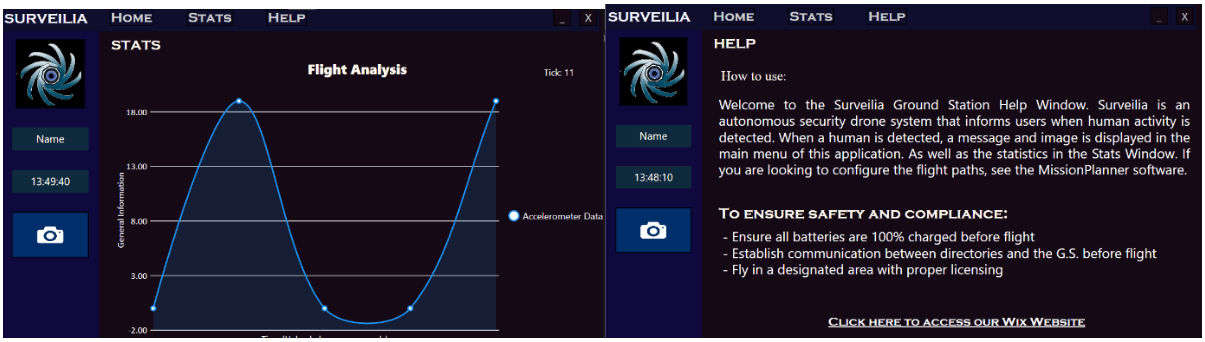

Figure 5 Ground station stats and help

- Home: Contains images and numerical flight data
- Stats: Displays accelerometer data graphically
- Help: Provides information on how to use the Ground Station
- User Menu: Defines purpose of the software and changeability of the operator's name

HOW IS DATA DISPLAYED AND WHAT DOES IT MEAN?

Temperature & Humidity: Environmental data for the current temperature and humidity

Accelerometer: Accelerometer data is a force of acceleration. On board Surveilia is an MPU6050 which senses accelerometer and gyroscope data, sending information back to the Pi.

Gyroscope: Gyroscope data reflects the axis of tilt that the drone is experiencing. This data lends insight into the drone's roll, pitch, and yaw.

Imaging: Alongside the image displayed, the packet involved in communicating flight data also displays a warning of someone in the frame and the number of people seen in the frame.

Figure 6 Data on ground station

BETA-FLIGHT AND MISSION PLANNER

BetaFlight is a software capable of programming your Kakute F7 to be the as precise as possible. BetaFlight is great for configuring the controller, but we primarily use it to flash the ArduPilot software so that we can connect to Mission Planner. Mission Planner is the open-source software we use to automate our drone to fly autonomously through written paths. Follow the following procedures to ensure that you have a successful initial flight.

CONNECTING THE TX/RX (BINDING)

1. Hold down the bind key on the FS-i6 and slide on the power switch
2. Hold down the binding button on the FS-A8S as its being powered up
3. Once the light on the receiver is a solid red then they have been bound together

DOWNLOADING/INSTALLING APPLICATIONS

1. Download BetaFlight found at
   a. https://github.com/betaflight/betaflight-configurator/releases
2. Open BetaFlight
3. Download the first FC driver:
   a. https://www.st.com/en/development-tools/stsw-stm32102.html.
4. Download the 4-in-1 ESC firmware at:
   a. https://drive.google.com/drive/folders/1Y1bUMnRRolmMD_lezL0FYd3aMBrNzCig
5. BetaFlight download for windows can be found at:
   a. https://github.com/betaflight/betaflight-configurator/releases
6. Once downloaded and installed, open beta flight and plug in the flight controller
7. STM32 drives can be found at Second STM Driver
8. The KO 4-in-1 ESC has its own firmware that can be installed from this link:
   a. https://drive.google.com/drive/folders/1Y1bUMnRRolmMD_lezL0FYd3aMBrNzCigs

BETAFLIGHT SETTINGS:

Thanks to the Kakute F7 for arriving pre-flashed with the BetaFlight software, beginning the process for programming your Surveilia Drone is seamless.

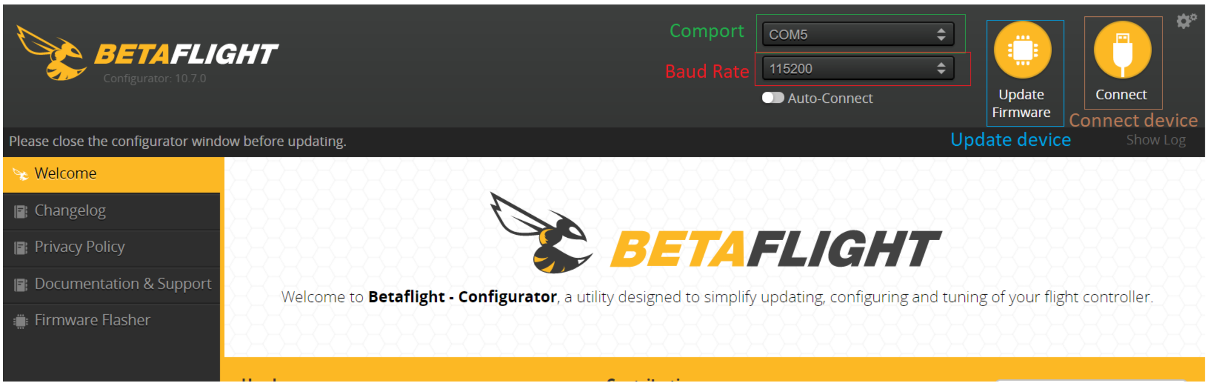

Figure 7 Beta flight application

1. Connect your micro-USB to USB between the Kakute F7 and your computer
2. Ensure you are connected (see graphic in top right of BetaFlight)
3. Once connected, navigate to the Setup tab.
4. Configure the UART serial ports for the Receiver and GPS (found under ports tab)
5. Click the Serial Rx option for the UART receiver pin (hardwired connection)
6. Click the “Disabled” drop down menu under the “Sensor Input” section for the UART pin that the GPS was soldered to
7. Select GPS and press the “Save and Reboot”

CONFIGURING BETAFLIGHT:

1. Navigate to the Configuration tab.
2. Set the ESC/motor protocol to ONESHOT 123 or DSHOT600
3. Set “Receiver Mode” to “Serial-based receivers”
4. Set the “Serial Receiver Provider” to “SBUS”
5. Lastly, other features such as Telemetry, OSD, Air Mode, and Dynamic Filter can be enabled, however; these do not need to be turned on to get the drone in the air
6. Click Save & Reboot
7. Move down to the “Receiver” tab to begin testing the controller and the receiver.
8. Turn on the controller transmitter to see the values for the roll, pitch, yaw, and throttle
9. If the values are not in the centre at 1500 for the pitch, yaw, and roll, add dead band to compensate

Note: In this tab, you can check what values your specific controller’s Aux switches are using. If you feel inclined, you can add additional features of functionality to the drone’s operation using those switches.

BETAFLIGHT MODES:

1. Navigate to the *modes* tab

2. Select ARM mode and set it to AUX 1 to set power on and off
3. Set top limit to 2100 and the bottom limit to 1700
4. If unsuccessful, investigate which switches are being used as AUX ports on your controller
5. Turn on "Angle mode" using another AUX port

BetaFlight and Motors:

1. Navigate to the motors tab
2. Test the rotation of the motors. DO NOT test the motors with propellers
3. Ensure all motors are properly soldiered to the correct pad.

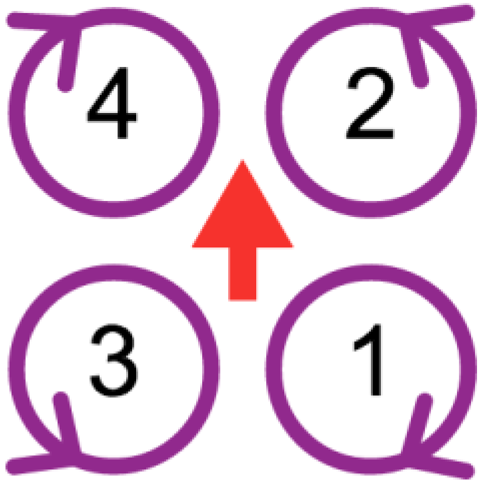

Figure 8 Motor Spin Direction

Setting up Mission Planner:

1. Download Mission Planner at http://firmware.ardupilot.org/Copter/latest/KakuteF7/
    a. Arducopter_with_bl.hex is the *target file*
2. Plug the Kakute F7 into the computer while holding bootloader button on the F7 (*See Figure 9 Kakute F7*)
3. Load the *target file* by clicking the "Load Firmware [Local]"
4. Press Flash

Note: This may take some time. Once completed, load Mission Planner and connect to the Kakute F7. You will no longer be able to connect via BetaFlight.

Figure 9 Kakute F7 Bootloader Button

Manufacturing the Drone Body

Downloading STL files

All drone components are available on the Surveilia GitHub under *DroneBody*:

- https://github.com/Surveilia/DroneBody/tree/main/STL%20Files.

The STL files are acquired to process slicer settings for the 3D printer.

Slicing

Slicing is the process of breaking down an STL file such that it can be understood by a 3D printer. The slicer has settings which users adjust to accommodate the needs of the print. Slicing can be achieved using *Cura. Cura* is specialized software designed to break down STL files into G-Code.

Cura can be downloaded at:

- https://ultimaker.com/software/ultimaker-cura

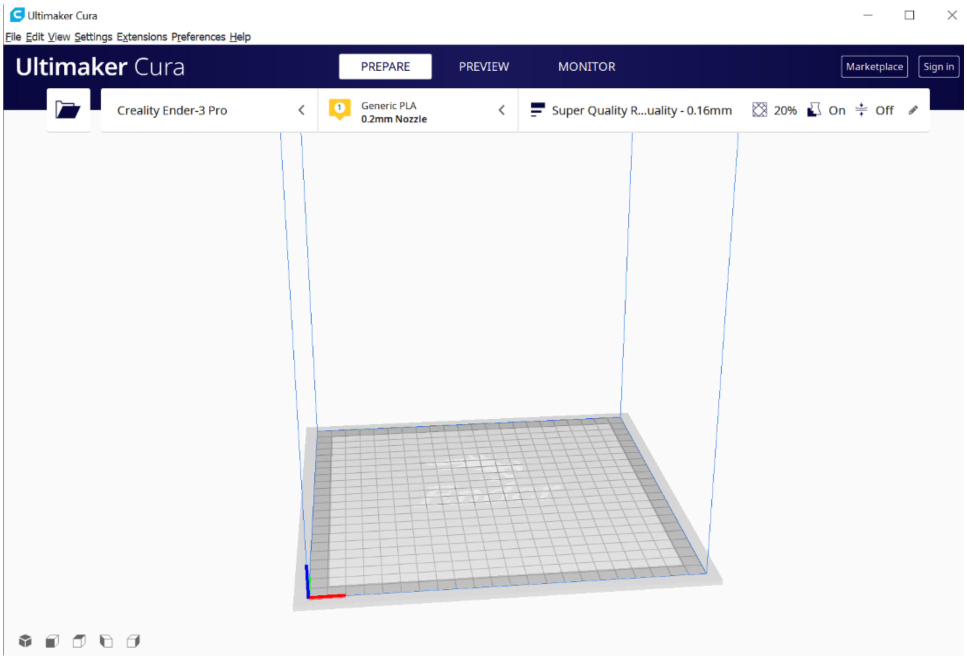

Figure 10 Cura Slicer Software

Different filaments also require different settings. It is recommended that the drone arms be printed with PETG filament for superior strength and rigidity.

Note: PETG requires high heat and cannot print multiple components at once.

Recommended PLA & PETG Slicer Settings

To achieve a successful print, the slicer must have the appropriate settings for the material being used. Please follow the recommendations provided for the chosen material.

For PETG with a standard 0.4mm Nozzle

- Cures deflate PETG setting work well by default.
- 0.2 mm layer height
- 50% infill
- 240 Celsius print temperature
- 80 Celsius bed temperature
- 4 wall layers
- Add a brim
- Slow first layer
- Add a thin layer of glue stick

PLA with a standard 0.4mm Nozzle

- 0.2 mm layer height
- 75% infill
- 200 Celsius print temperature
- 60 Celsius bed temperature

PRINTER

The Ender 3 was used to manufacture the drone body. There are several important components on the printer that users need to be aware of. All of which are labelled in Figure 9.

1. *Extruder:* Feeds plastic filament to the hot end for printing
2. *Hot end:* The hot end heats the plastic so it can be moulded for the print. The hot end contains the nozzle
3. *Build Plate:* The build plate is the structure that the parts are built on.
4. *LCD Screen:* The LCD screen is the printer interface for software tuning and is used to start a print
5. *Power Supply:* The power supply has an on/off switch

Figure 11 Ender 3 printer

PRINTER TUNING

One of the most important parts of printer tuning is bed levelling. Bed levelling is achieved by adjusting bed level screws to maintain a bed-nozzle alignment. G-Code is available to help in levelling the bed quickly https://www.thingiverse.com/thing:3235018. Be aware, bed levelling is a largely subjective process and troubleshooting is required.

Level the bed with

- a sheet of printer paper
- G-Code bed level

ONGOING INSPECTION

During the printing process, failures can occur for a variety of reasons.
When conducting a print job, the user should pay close attention to:

1. Bed adhesion
   a. Make sure that the nozzle is not too close, and not too far from the build plate when printing is occurring
   b. Ensure that the first few layers are secure to the bed and oozing doesn't occur
2. Stringing of filament
   a. If the extruder becomes blocked, or if bed adhesion did not occur, stringing may happen
   b. Characterized from a lack of consistent texture in the part
   c. Will not have solid walls

d. Note: PLA and PETG will have some excess stringing. This is normal and will look like a cobweb

AFTER PRINT – ASSEMBLY

Once all the drone components have been successfully printer, they must be assembled. You will need:

- 4 x 16mm M3
- 4 x 10mm M3
- 8 x 8mm M3
- 16 x Motor Mount bolts
- 12 x M3 nuts
- Lock tight
- M3 x 0.5mm tap
- 4 x Motors
- 4 x Propellers
- 1 x ESC
- 1 x Flight controller
- 1 x Drone top plate
- 1 x Drone bottom plate
- 4 x Drone arms
- 4 x Drone body standoffs

STEPS TO ASSEMBLING THE SURVEILIA DRONE:

1. Begin by assembling arms and bottom plate
2. Use 16mm M3 bolts to secure arms to bottom plate
3. Fasten with nuts on bolts
4. Fasten stand offs to bottom plate with 8mm M3 screws
5. Attach ESC and F7 stack to bottom plate using 10mm M3 screws
6. Mount motors to each arm, using M3 screws
7. Fasten top plate to stand offs using M3 screws

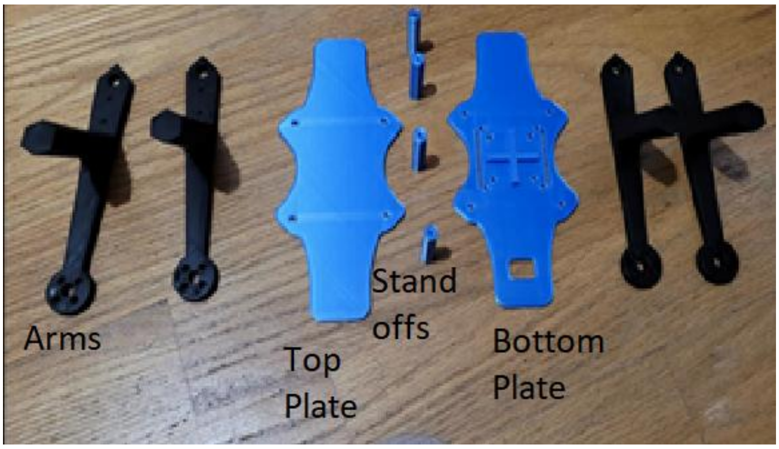

Figure 12 Drone body components

UDP Communication

Setup & Executing UDP

UDP is the protocol used to communicate between the drone and the ground station software. All transmitted files are named using the date/time that they were sent. This makes it easy to match a particular iteration of data with an image to be displayed concurrently.

Find the UDP files from Github at https://github.com/Surveilia/UDP/tree/main/UDP

1. Download client, packet and server from Github
2. Install client and packet on the flight computer
3. Install server on computer hosting ground station
4. Find IP address of flight computer with the console command *hostname -I*
5. Execute communication on flight computer and host computer
    a. Type *python SurveiliaClient.py* on flight computer console
    b. Type *python SurveiliaServer.py* on host computer console

Human Recognition

For our human detection algorithm, we are going to need to install OpenCV 4. OpenCV 4 is a real-time optimized computer vision software that includes libraries, tools, and modules that are open source and easy to use.

Because PyPi does not contain pre-compiled OpenCV 4 binaries which can be installed via pip, we will need to compile OpenCV 4 from source.

Recommended: Before doing any installs the first thing you should do is update your system:

Steps to Installing OpenCV on Raspberry Pi Zero W:

1. Enter the terminal interface on the Raspberry Pi Zero
2. Run the following command
    $ sudo apt-get update && sudo apt-get upgrade
3. Then install the relevant developer tools:
    $ sudo apt-get install build-essential cmake unzip pkg-config
4. Next, install a selection of image and video libraries
    $ sudo apt-get install libjpeg-dev libpng-dev libtiff-dev
    $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
    $ sudo apt-get install libxvidcore-dev libx264-dev
5. From there, install GTK and the GUI backend:
    $ sudo apt-get install libgtk-3-dev

6. And then install a package that will reduce GTK warnings:

    $ sudo apt-get install libcanberra-gtk*

    Note: The asterisk is needed to grab the ARM specific GTK

7. This is followed by installing two packages which contain numerical optimizations for OpenCV:

    $ sudo apt-get install libatlas-base-dev gfortran

8. Finally, install the Python 3 development headers:

    sudo apt-get install python3-dev

OpenCV Download

1. First, download both the "OpenCV" and "opencv_contrib" code:

    $ cd ~
    $ wget -O opencv.zip
    https://github.com/opencv/opencv/archive/4.0.0.zip
    $ wget -O opencv_contrib.zip
    https://github.com/opencv/opencv_contrib/archive/4.0.0.zip

2. From there, unzip the archives:

    $ unzip opencv.zip
    $ unzip opencv_contrib.zip

3. OPTIONAL – Rename the directories

    $ mv opencv-4.0.0 opencv
    $ mv opencv_contrib-4.0.0 opencv_contrib

Configure Python3 environment for OpenCV (Recommended)

1. First, install pip which is a python package manager:

    $ wget https://bootstrap.pypa.io/get-pip.py
    $ sudo python3 get-pip.py

2. Secondly, install "virtualenv" and "virtualenvwrapper" to allow for python virtual environments:

    $ sudo pip install virtualenv virtualenvwrapper
    $ sudo rm -rf ~/get-pip.py ~/.cache/pip

3. Install these tools, we will need to update our "~/.profile" file.
4. Using a terminal text editor such as "nano" or "vim", add the following lines to your ~/.profile file:

    ```
    # virtualenv and virtualenvwrapper
    export WORKON_HOME=$HOME/.virtualenvs
    export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
    source /usr/local/bin/virtualenvwrapper.sh
    ```

5. Next, source the ~/.profile file:

    $ source ~/.profile

CREATE VIRTUAL ENVIRONMENT FOR OPENCV 4 AND PACKAGES

Virtual environments allow you to run different versions of Python software in isolation on your system. Any python interpreter, library, or script will be installed only on the environment that you are inside. While this step isn't required, it is highly recommended.

1. You can now create a OpenCV 4 + Python3 virtual environment:
   $ mkvirtualenv cv -p python3

This command creates a Python 3 virtual environment named "cv". If you prefer more detail, I recommend using a name like "py3cv4". For the rest of this guide, we will use "cv".

2. To verify that the virtual environment is working as intended, use the command:
   $ workon cv

Note: If the verification process was successful, you are now working inside the CV virtual environment.

3. Next, you can install the first Python package and only OpenCV requirement NumPy:
   $ pip install numpy

MAKING AND COMPILING OPENCV 4

Following the build of your virtual environment, you will want to register your build with your OpenCV.

1. Navigate back to your OpenCV repo and create + enter a build directory:
   $ cd ~/opencv
   $ mkdir build
   $ cd build
2. Now run CMake to configure the OpenCV 4 build:
   **$ cmake -D CMAKE_BUILD_TYPE=RELEASE **

   **-D CMAKE_INSTALL_PREFIX=/usr/local **

   **-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules **

   **-D ENABLE_NEON=ON **

   **-D ENABLE_VFPV3=ON **

   **-D BUILD_TESTS=OFF **

   **-D OPENCV_ENABLE_NONFREE=ON **

   **-D INSTALL_PYTHON_EXAMPLES=OFF **

   -D BUILD_EXAMPLES=OFF ..
3. Following the running of this command, scroll down until you see "OpenCV modules". Under those lines you should see "non-free algorithms".

4. Make sure that this says “Yes”.
5. Open your “/etc/dphys-swapfile” file using a text editor like “nano”:
   $ sudo nano /etc/dphys-swapfile
6. Then edit the “CONF-SWAPSIZE” variable:
   # CONF_SWAPSIZE=100
   CONF_SWAPSIZE=2048

Note: In the case above, the swap size will be changed from 100MB to 2048MB. While this step is optional, not performing it will likely cause your system to hang.

7. From there, restart the swap service:
   $ sudo /etc/init.d/dphys-swapfile stop
   $ sudo /etc/init.d/dphys-swapfile start

Compiling OpenCV 4

Steps to compiling OpenCV 4:

1. Run command:
   $ make –j4
2. Install OpenCV 4 with two additional commands:
   $ sudo make install
   $ sudo ldconfig
3. Next, go back to your “/etc/dphys-swapfile” file and:
   Reset CONF_SWAPSIZE to 100MB from 2048MB
4. Restart the swap service

Linking OpenCV 4 into Python3 virtual environment

Steps to linking OpenCV 4 Into Python 3 Virtual Environment:

1. Create a symbolic link from the OpenCV install in the system “site-packages” directory to our virtual environment (THE PYTHON VERSION MAY BE DIFFERENT IN YOUR FILE PATHS):
   $ cd ~/.virtualenvs/cv/lib/python3.5/site-packages/$ ln -s /usr/local/python/cv2/python-3.5/cv2.cpython-35m-arm-linux-gnueabihf.so cv2.so
   $ cd ~

Note: This step is critical. If you don’t create a symbolic link, you won’t be able to import OpenCV in your scripts.

Test your OpenCV 4 install on your system:

To test if your OpenCV 4 install succeeded:

1. Open the terminal
2. Perform the following:
   $ workon cv
   $ python
   >>> import cv2

>>> cv2.__version__
'4.x.x'
>>> exit()

Note: The first command activates our virtual environment. We then run the Python interpreter associated with the environment. If the version displayed is 4.x.x, your installation was successful.

Pi Camera Setup

All current models of Raspberry Pi have a port for connect a PiCamera module. The camera port can be seen below surrounded by a red rectangle.

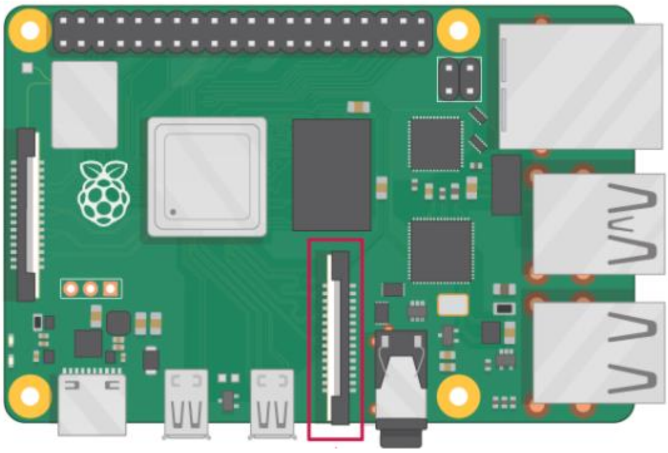

Figure 13 Raspberry Pi 4 Module

If you're using a Raspberry Pi Zero (or Zero W), you need a camera module ribbon cable that fits the RPi Zero's smaller camera module port.

Enabling the camera module

Once your PiCamera is installed, you need to enable it via the command line.

1. Open the command line and execute the following command:
   - **$ sudo raspi-config**
2. Using the arrow keys on your keyboard to scroll down to Option 5
3. Enable camera and press enter (If "Enable Camera" isn't your 5th option, look around for it as it was probably moved to a different path).
4. After it is enabled, arrow down to the "Finish" button and press enter again.
5. Reboot your Raspberry Pi for the configuration to take effect.

Testing the camera module

To test if everything is working as intended:

1. Open your terminal
2. Execute the following command:

   $ raspistill -o output.jpg

Note: This command activates your Raspberry Pi camera module. After waiting a few seconds, it snaps a picture and saves it your current working directory as “output.jpg”. If the photo came out as expected, your Raspberry Pi camera module is working as intended.

Installing Pi Camera

Now that you know that the camera module is working properly, it is time to configure your Raspberry Pi module to interface with the camera module using Python.

Steps to integrating the Raspberry Pi:

1. To do this, we will install the PiCamera module (Before doing this, make sure you are inside the “cv” virtual environment):

```
$ pip install "picamera[array]"
```

Note: If your install finished without any errors, you now have the PiCamera module with NumPy array support installed.

Trouble shooting

Battery Charging and Discharge:

The drone runs off a 6s LiPo which means there are 6 cells all at a voltage of 3.7V.

- The battery's total voltage is 22.2V.
- LiPo battery's need to be charged one cell at a time.
- If you push voltage into the battery cell, it will get overcharged and blow up.
- Using a smart balance charger is key when using LiPos.

Here is a charger that is capable of charging 1s – 6s battery's:

- https://www.amazon.ca/Battery-Charger-Discharger-Chargers-Connectors/dp/B07V5CPYRS

Note: While charging or discharging a LiPo, a fireproof charging bag is a necessary safety measure for preventing injuries.

Choosing a Battery:

1. If you have higher KV, a smaller 4s or 5s LiPo can be used.
2. The ESC and flight controller can handle 4s up to 6s batteries.
3. Additionally, they do not need to have a huge capacity, around 1200mAh to 2000mAh is plenty for this drone to fly for a few minutes.
4. A discharge rating of about 100c to 130c is ideal for proper current draw.

Motor Safety:

1. When testing the motors on a bench, do not have any propellers attached. (As the drone will not be flying in the air, there is no significant cooling for the motors.)

2. If the motors run for too long, or go too fast without cooling, they will overheat. Not only will you burn the motors, but possibly also yourself.

Note: You can check the motor coils to see if there is any black charring on them. If there is, then the motor has gotten too hot and may need to be replaced if it is significantly damaged. If the motors are not spinning at the same rate, then one or more motors might be burnt out.

CONCLUSION

Surveilia was designed to be used in the security industry or by residences. We built and included a large suite of systems and functionality, with repairability in mind. By following the above steps and instructions, you should now have your very own complete Surveilia prototype. Ensure if you decide to build this drone, you follow all proper legalities for flying in your area.

Thank you for taking your time to understand the Surveilia operations and functionality. We are proud of our accomplishments this term and hope to have provided you with the necessary information to seamlessly begin building a drone of your own.

Contact us at surveiliadrone@gmail.com

REFERENCES

Mission Planner. (Aug 1, 2021). ArduPilot. Accessed September 20th, 2021. [Software application]. Available: https://firmware.ardupilot.org/tools/missionplanner/

A. Rosebrock, "Install OpenCV 4 on your Raspberry Pi," 26 Sept, 2018. [Online]. Available https://www.pyimagesearch.com/2018/09/26/install-opencv-4-on-your-raspberry-pi/. [Accessed: Sept. 23, 2021].

Iottrends, "What is Raspberry Pi Compute Module 4 – A Complete Guide," 16 Jan, 2021 [Online]. Available:https://www.iottrends.tech/blog/raspberry-pi-compute-module-4-everything-you-need-to-know/ [Accessed: Oct 12, 2021].

Paladin Security, "Security Guard Salary in Canada by Province in 2021," Nov 14,2021. [Online] Available: https://paladinsecurity.com/security-careers/security-guard-salary/. [Accessed: Sept 16, 2021].

Wikipedia, "Histogram of Oriented Gradients," 01 Feb, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients. [Accessed: September 24, 2021].

Y. Xu, G. Yu, Y, Wang and X. Wu, "A Hybrid Vehicle Detection Method Based on Viola-Jones and HOG + SVM from UAV Images", ResearchGate, Sensors. 16. 1325., p.6, Aug, 2016. [Online]. Available: https://www.researchgate.net/publication/306333664_A_hybrid_vehicle_detection_method_based_on_viola-jones_and_HOG_SVM_from_UAV_images. [Accessed: Oct 12, 2021].

"Feature descriptor: Hog descriptor tutorial," *Analytics Vidhya*, 10-May-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/. [Accessed: 12-Nov-2021].