Wescam Hackathon 2019

Mission Briefing

Steve Lascos, Technical Lead
Software Engineering

Meet The Drones

You will be controlling a team of three Parrot Bebop 2 Drones.

Callsign Alpha

Task - provide Overwatch to maintain situational awareness of the operating area.
Altitude – maintain 15ft
Navigation – IMU / dead reckoning
Notes: maintains constant heading (no rotation)

Meet The Drones

You will be controlling a team of three Parrot Bebop 2 Drones.

Callsign Bravo

Task – general target classification, identification
Altitude – maintain 10 ft
Navigation – command via Overwatch
Notes: medium field of view (FOV)

Meet The Drones

You will be controlling a team of three Parrot Bebop 2 Drones.

Callsign Charlie

Task – up close target classification, identification
Altitude – maintain 5 ft
Navigation – command via Overwatch
Notes: closest to target, narrowest FOV

OPERATION SWARM BRIEFING

You will be taking command of three Bebop 2 class drones as part of a training simulation for operating Unmanned Aerial Vehicles (UAV) in adversarial conditions. UAVs are ideal for situations where flight time, aircraft size, or risk to the pilot prevent use of manned aircraft.

We will be simulating a tactical situation where the enemy is jamming GPS and communications frequencies and the operational risk to the pilots demands used of UAVs.

In the real world, short-range IR laser line-of-sight communication would provide an un-jammable means of communicating between a high-altitude OVERWATCH surveillance drone who visually monitors the entire theater of operation and provides navigation guidance to the low altitude drones in its field of view.

Since conventional communication between the OVERWATCH drone and mission control would be impossible due to jamming, the OVERWATCH drone must operate completely autonomously with its own Mission Computer to

i) Enter the theater of operation
ii) Co-ordinate assets in the theater
iii) Complete mission objectives
iv) Guide the assets home then itself return safely.

Since the Bebop 2 Training Drone doesn't have a sufficiently powerful on-board computer we will use an NVIDIA TX2 to stand in as the Command & Control computer.

Often a company like Wescam is selected to provide the machine vision computing components of the aircraft. The customer often has strict rules on how we interface to their equipment, particularly anything that impacts mission criticality or aircraft safety.

Before being selected, Wescam must first demonstrate basic capabilities to provide confidence to the customer that we will be successful in meeting their requirements.

DEVELOPMENT RULES OF ENGAGEMENT

- You will use the Wescam provided libwscDrone C++ library to communicate with the drones.
- You are not permitted to directly use the Parrot ARSDK3 libraries.
- The drones store photos onboard. You can download these photos at any time for processing to any device. E.g. your own computer, a Raspbery Pi 3, etc.
- Wescam employees will represent the customer, evaluating your progress and monitoring all tests to ensure the safe operation of the aircraft.

- You must first demonstrate situational awareness and effective command and control of all drones.
- Once you have completed the qualifying missions you can move onto the more advanced missions.

WESCAM PROVIDED RESOURCES

Wescam will provide you with a pre-configured NVIDIA TX2 development board (one per group). You will compile your software to run on this board in order to control the drones.

If you have any difficulties with compiling the default provided software on the TX2, ask a Wescam Hackathon representative for a Software rep to come help you.

There is also a VM that was provided via email correspondence before the event. The username and password for both the TX2 and the VM is

Username: hackathon
Password: hack

libwscDrone

libwscDrone is provided on github at:

https://github.com/WescamSW/libwscDrone_TX2

It has been pre-installed on your TX2 at

/home/hackathon/libraries/libwscDrone_TX2

We may need to push out updated versions of this library during the hackathon. If you are requested to update to a new version type the following from the above directory:

git pull

./install_library.sh

Don't forget to cleanly rebuild your application after updating this library.

Hackathon2019 Demo

Provided on github at:

https://github.com/WescamSW/Hackathon2019_Demo

We may put useful updates in this repo. You can update it to the latest on Github by typing in the above directory.

git pull

You can compile the two demo programs by running

make clean

make

BY DEFAULT THE DEMO HAS A FLIGHT GUARD WITH PERMITS CONNECTING TO A DRONE AND STREAMING VIDEO WITHOUT STARTING ANY PILOTING MISSIONS.

YOU MUST COMMENT OUT THE LOCK GUARD IN ORDER TO HAVE THE DEMO FLY THE DRONES.

#define NO_FLIGHT // This means flight guard is on

//#define NO_FLIGHT // This means flight guard is off due to the leading slashes.

Runnning the demo program requires access to TWO drones, alpha and bravo. If a test drone is available in the development area (no props installed), you can power it on and connect to the WiFi network it creates.

./bebop2Swarm_arm64 lone_wolf

The 'lone_wolf' parameter tells the program to connect directly to the drones WiFi network.

The provided demo also has the ability to demonstrate control of two drones simultaneously. When working with the 'propped' drones, you must use their proper callsigns. In this mode, the drones will connect to the Wifi network we have setup in the drone flying area. You pass the drone callsigns as parameters.

./bebop2Swarm_arm64 alpha bravo

Frequently Asked Questions

Can I use my own computers or devices to control the drones?

Yes, but you must network them to the TX2. The application you run on the TX2 must be the one directly communicating with the drones.

The exception is the photos taken by the drone. These can be accessed via FTP and download directly to your own device such as another computer, a Raspberry Pi, etc.

Can I write my software in Python, Rust or some other language?

Yes, but only the C++ wscDrone library running on the TX2 can directly control the drones so at a minimum you will need a C++ application similar to the Hackthon2019_Demo which the other software communicates with.

You could write additional programs in any language on any device that can communicate with your C++ application.

For example, consider the following: employing a 'mailbox' approach using the TX2 filesystem might be useful. Other applications can copy text files containing commands into the "inbox". The C++ Application could return data by placing files in an "outbox". This could be used to have two different applications on the TX2 communicate with each other, or an application running on another computer on the network.

How do I download photos or video from a drone?

Drones have static IP addresses based on their callsign.

Lone Wolf – 192.168.42.1 (only on the non-propped drones when you connect to their Wifi network)

The flying drones will be on one of two networks:

Test Area A SSID: 'Hack2019A'

Test Area B SSID: 'Hack2019B'

Wifi password: 'hack1337'

The drones on these networks will be:

Alpha – 192.168.1.101

Bravo – 192.168.1.102

Charlie – 192.168.1.103

Video is automatically recorded and stored in the same location as photos. The drone can take several different types of photos, refer to the libwscDrone documentation for the CameraControl class.

Media is stored on the drones ftp drive at:

/internal_000/Bebop_2/media

How can I test my software when not flying on the drones?

Access to the drones will be via taking turns during 15-minute "slots" during which you can fly up to a maximum of 8 minutes to ensure time to download your video/photos before the next team comes in.

Taking photos and video of the actual test area during your first runs provides realistic test data for you to use while you further test and develop your software.

How do I know if I've completed a mission objective?

The Wescam representative operating the drone area will monitor your mission and evaluate based on the video displayed by either your TX2 application, and/or the application running on other devices. See the individual missions for further details.

What if I crash a drone, hit something or lose control?

The Wescam representative will be monitoring your drones while flying. Your software must run autonomously, but will provide some user keyboard control on the TX2 in order for us to stop the test. SPACEBAR must land all drones immediately. In an emergency F12 must cut the motors (the drones will fall). See the Hackathon2019_Demo for an example on how to do this.

If I fail to complete a mission, do I get to retry?

Yes. You may need several test flights in order obtain photos or video in order to develop your software algorithms. You will can run multiple test flights during your ‘slot’ on the Drone Flight Deck.

RULES OF THE FLIGHT DECK

All missions will rely on providing the Wescam operator the ability to land drones safely. The Hackathon2019_Demo provides keyboard controls to make manual flight adjustments to the drone (CURSOR KEYS), as well as land all drones immediately (SPACEBAR). These will be checked before each mission as a preflight check.

Pre-Mission Flight Control Checklist

1. Check to make sure you have the latest version of libwscDrone installed using the check_version.sh script.
2. Have the volunteer power on the drones and connect to the Wifi network.
3. After launching the application, is the video feed displayed for all 3 drones on the TX2 output?
4. When ready, does pressing 't' cause all drones to take off to their designated altitudes?
5. ~~Can each drone me manually controlled with the cursor keys by selecting the with the '1', '2', or '3' keys at the top of the keyboard?~~
6. Does pressing SPACEBAR cause all drones to land?

If these checks pass, the Team is cleared to fly their mission.

Mission is a 'GO' for FLIGHT

Once the mission is a go, the drones will be put back to their starting positions. The Wescam operator will use the 'g' key to commence takeoff and let the computer take control. Once finished the mission, the drones return to their landing locations (while still hovering). The Wesam operator will make any necessary manual corrections and press SPACEBAR to land all drones.

Post-flight Debrief

Contestants will likely be downloading their photos during their mission, however the mission videos must wait until landing and the video stream is stopped. They can then download their mission videos while time permits before the next team arrives.

PLEASE DELETE YOUR VIDEO AND PHOTOS OFF EACH DRONE BEFORE LEAVING THE FLIGHT DECK.

HACKATHON 2019 MISSIONS

Operation Green Triangle GT-A – Qualification

Objective – Demonstrate OVERWATCH capability by flying a single high altitude drone (Alpha) such that the entire Area of Operation (AOO) is within the FISHEYE Camera Field of View (FOV).

Use machine vision techniques to identify and track the corners of the test area, use them to steer the drone such that it maintains heading (the direction the drone is facing) as well as position (it doesn't drift away).

While Bravo and Charlie will rely on Alpha to track their position within the AOO in future missions, Alpha must determine its own absolute position using dead reckoning. E.g., if takeoff is at (0,0), and the drone moves 1 metre to the right and 2 metres forward, it's current position is at (1,2). This information is needed to guide Alpha back to its launch site at the end of a mission.

When returning home, the drone will move to the landing position and hold position. The Wescam operator will assess if it is safe to land and make any manual corrections if necessary.

SUCCESS CRITERIA

- Does the Team visually show evidence they are tracking the reference using graphics or symbols on the video or photos?
- Does the drone maintain its orientation and position in the middle of the test area for at least 2 minutes?
- Did the drone return to a safe landing position without major corrections?

Operation Green Triangle GT-B – Qualification

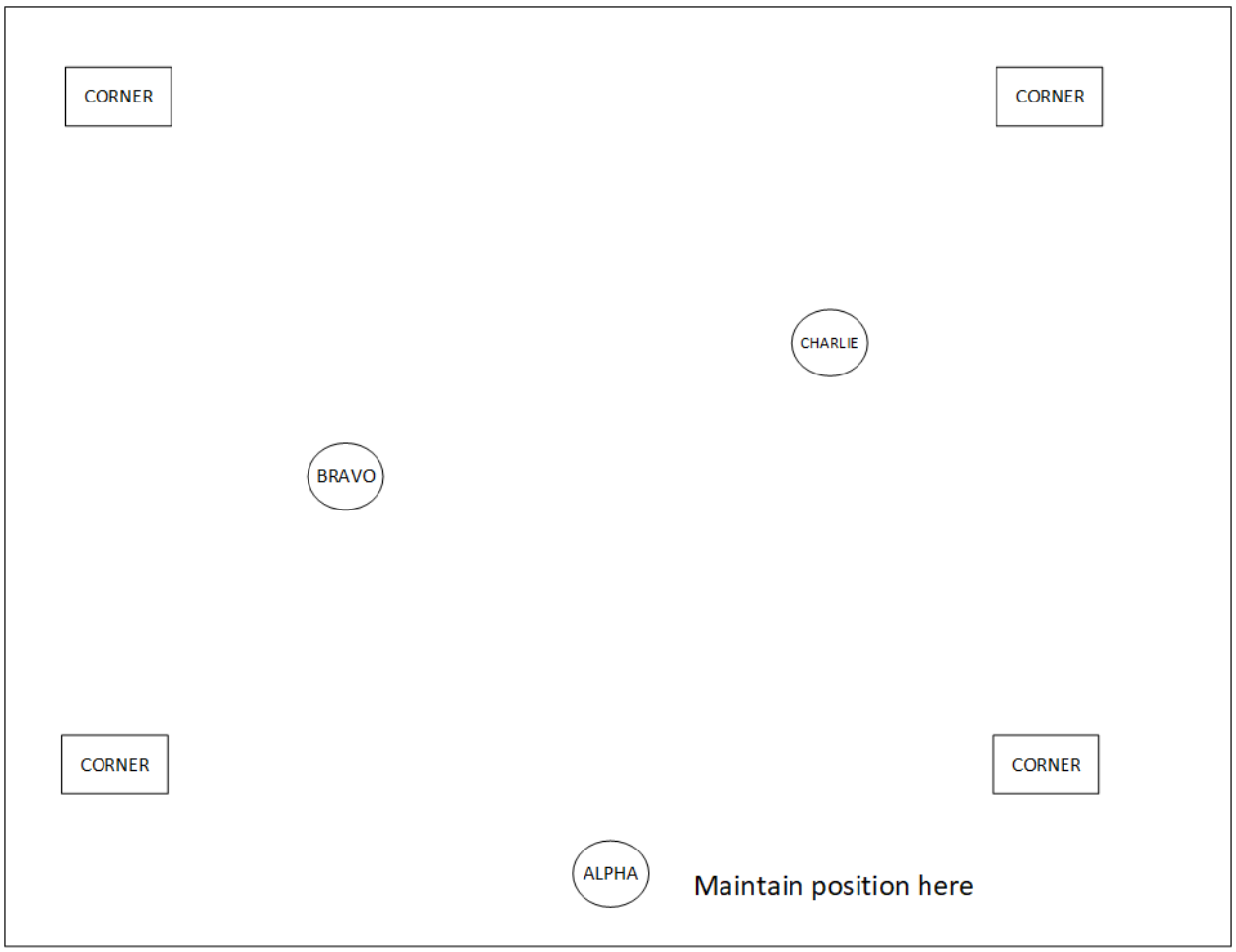
This mission adds upon GT-A. Now that you have demonstrate you can hold callsign Alpha over the AOO without drifting, it's time to control the other drones.

Fly drones Bravo and Charlie into the test area, find and track them using Alpha's field of view, and use this information along with the position of the corner markers to move them where you want, and keep them from leaving the AOO unintentionally.

Fly them home back to the landing area.

SUCCESS CRITERIA

- Did Alpha maintain OVERWATCH by visually demonstrating tracking of the corners and the drones?
- Did all drones return to the safe landing area without major correction?



Operation Red Comet RC-A – Text Recognition

Intelligence is picking up local chatter in the Area of Operation (AOO). Read text in the AOO so intelligence can match up their intel with specific locations.

Detect and read all three English words.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- The text must be printed as text graphics on a video or photo showing the text object in the scene.
- 'Noise characters' and false positives are permitted. Eg. If the word is 'WELCOME', the following would be acceptable. "89ewr WELCOME |_(* ||"
-

Operation Red Comet RC-B – Text Recognition

Intelligence has requested better accuracy. Read text cleanly with no false positives or noise characters.

Detect and read all three English words.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- The text must be printed as text graphics on a video or photo showing the text object in the scene.
- 'Noise characters' and false positives are NOT permitted.

Operation Red Comet RC-C – Text Recognition

Intelligence has requested the location of specific areas of interest. Read the alpha numeric text cleanly with no false positives.

Detect and read all three alpha-numeric codes.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- The text must be printed as text graphics on a video or photo showing the text object in the scene.
- 'Noise characters' and false positives are NOT permitted.

Operation Golden Angel GA-A – Target Identification

Intelligence has determined that a high value target is in the Area of Operation.

Find the **Tonka Truck** carrying bomb making equipment.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- At the end of the mission, display to the operator two cropped photos containing the truck.
- One of the two candidates must contain the truck. The truck should fill more than half the field of view.

Operation Golden Angel GA-B – Target Identification

Intelligence has determined that a high value target is in the Area of Operation.

A **Fire Truck** has been stolen as part of a plan to gain access to a secure area by causing a fire alarm. Find the Fire Truck.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- Use Charlie to get a photo of the Fire Truck.
- Only one candidate may be displayed as a cropped photo where the truck fills more than half the field of view.

Operation Golden Angel GA-C – Target Identification

Intelligence has discovered the enemy is constructing a concealed barracks.

Find the **Bulldozer**. Pinpoint it's location on using the FISHEYE view from Alpha and the SNAPSHOT view from Bravo. Provide a cropped image from Charlie.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- Provide 3 photos at the end of the mission containing the Bulldozer including
    - Mark the location of the bulldozer from Alpha's FISHEYE.
    - Mark the location of the bulldozer from Bravo's SNAPSHOT.
    - Provide a cropped image from Charlie where the Bulldozer fills at least half the field of view.

Operation Bronze Fire BF-A – Target Classification

The enemy has infiltrated a de-militarized zone (DMZ).

They are hiding improvised explosive devices as **Balls**. Final all balls in the Area of Operation.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- Identify all balls in the scene. Provide croped photos of each candidate ball at the end of the mission.
- The number of correct positives must exceed the sum of missed balls and false positives.

Operation Bronze Fire BF-B – Target Classification

The enemy has infiltrated a de-militarized zone (DMZ).

Soldiers have illegally entered the DMZ. Find all **Human Dolls** in the Area of Operation.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- Identify all human dolls in the scene. Provide croped photos of each candidate doll at the end of the mission.
- The number of correct positives must exceed the sum of missed balls and false positives.

Operation Bronze Fire BF-C – Target Classification

The enemy has infiltrated a de-militarized zone (DMZ).

Arms are being illegally supplied to the DMZ. Find all **Toy Guns** in the Area of Operation.

SUCCESS CRITERIA

- Did all previous requirements for pre-flight and safe return succeed?
- Identify all human dolls in the scene. Provide croped photos of each candidate doll at the end of the mission.
- The number of correct positives must exceed the sum of missed balls and false positives.