SET UP GUIDE FOR CUSTOM GCS – CAV GCS with PCB

1. GroundUnit and AirUnit Setup

1A GroundUnit setup

1. Upgrade AirUnits to the latest firmware.(Make sure toggle both switches at down position)
2. Power off the GroundUnit by disconnecting the power cable (7v-12v).
3. Remove the screws from the chassis.
4. Herelink AirUnit v1.0 (without ethernet port) Toggle the switches as shown below to activate GroundUnit. (1 up, 2 down) **Note: For Herelink AirUnit v1.1 (with ethernet port) GroundUnit activated by toggle switch (1 down, 2up)**
5. Power on GroundUnit by reconnecting the power cable (7v-12v).

A close-up of a circuit board

Description automatically generated

1B AirUnit setup

Upgrade AirUnits to latest firmware.(Make sure toggle both switches down position)

2. Pairing

Press and hold the pairing button for 3s on both AirUnit and GroundUnit within 30s

\*Once pairing is completed LED will turn solid GREEN.

4. PC setup

1. Please switch off PC’s WIFI

2. Go to control panel > Network & Internet > Network connection

3. Right click and choose properties

4. Choose “internet (TCP/IPv4)”,then click properties

5. Click “use the following IP address” and enter 192.168.144.3”

6. Click “Subnet mask” and it will give you a Subnet mask.

A screenshot of a computer screen

Description automatically generated

Mavproxy connection (Reference) :

Open command terminal and type the command which is below. For connecting the vehicle through mavproxy .

Command to connect through Com port:-

mavproxy --master="com54",57600 --out 127.0.0.1:14560 --out 127.0.0.1:14561 --out 127.0.0.1:14562 --out 127.0.0.1:14563"

// udpout:192.168.144.11:14552 –main vehicle connection

// out 127.0.0.1:14560 – localhost connection to MP.

// out 127.0.0.1:14561 – localhost connection to python.

// out 127.0.0.1:14562 – localhost connection to QGC.

// out 127.0.0.1:14563 – localhost connection to UGCS.

Command to connect through ethernet :-

mavproxy --master=udpout:192.168.144.11:14552 --out 127.0.0.1:14560 --out 127.0.0.1:14561 --out 127.0.0.1:14562 --out 127.0.0.1:14563

Pymavlink connection through python script :

vehicle = utility.mavlink\_connection('udp:127.0.0.1:14561')  -- connection to vehicle

Command to connect through BotBlox Switch:-

mavproxy --master=udp:192.168.144.3:14552 --streamrate=-1 --out 127.0.0.1:14560 --out 127.0.0.1:14561 --out 127.0.0.1:14562 --out 127.0.0.1:14563

**Wiring and connection with PCB.**

Connecting Gimbal potentiometer with Teensy Board to act as joystick

Pinout :

* Analogue pin – A0 for roll
* Analogue pin – A1 for pitch
* Analogue pin – A2 for throttle
* Analogue pin – A3 for yaw

Connecting Buttons and camera control to Arduino.

* Pinout :
* const int buttonPin = 2;      //Safetyoff-d2
* const int buttonPin11 = A6;   //safetyon-a6
* const int buttonPin1 = 7;     //Arm -d7
* const int buttonPin2 = 3;     //Disarm -d3
* const int buttonPin3 = 8;     //Althold -d8
* const int buttonPin4 = 4;     //Loiter -d4
* const int buttonPin5 = 9;     //Auto -d9
* const int buttonPin6 = 5;     //RTL -d5
* const int buttonPin7 = 6;     //Land -d6
* const int buttonPin8 = 10;    //Cameraautocenter -d10
* const int Switch = A3;   //flip switch
* const int buttonPin9 = 11;    //payload Arm and open- d11
* const int buttonPin10 = 12;   //payload disarm and close -d12
* float pot1Value; // Declare variable to store value from potentiometer 1
* float pot2Value; // Declare variable to store value from potentiometer 2
* float pot3Value; // Declare variable to store value from potentiometer 3
* const int pot1 = A0; // Define the pin connected to potentiometer 1
* const int pot2 = A1; // Define the pin connected to potentiometer 2
* const int pot3 = A2; // Define the pin connected to potentiometer 3
* float prevPot1Value = 0; // To store the previous value of pot1
* float prevPot2Value = 0; // To store the previous value of pot2
* float prevPot3Value = 0; // To store the previous value of pot2

Uploading Code :

* Upload Teensy code from “Teensy” folder for AG01 gimbal to act as joystick.
* Upload Arduino code from “Arduino” Folder to connect buttons.

Setup Guide for GCS with Vehicle Integration using GCS laptop:

1. **Download and Install Arduino IDE (Optional)**   
   Download and install the Arduino IDE, which is optional but may be needed for checking the output of Arduino program. Upload the code to Arduino board

**File name: GCSMAIN.ino**

1. **Download and Install Teensyduino (Optional)**   
   Download and install Teensyduino, an optional add-on for the Arduino IDE to read stick input values.

**File name:** **joystickGCSTeensy.ino**

1. **Download and Install Python**   
   Ensure that Python is installed on your system. Download python for windows.
2. **Download MavProxy**   
   Download and install MavProxy, a command-line-based ground control station software for MAVLink.
3. **Install Required Python Libraries**
4. Install **PySerial**: A Python library for serial communication.

Open command prompt -> open pymavlink directory-> paste “pip install pyserial”

1. Install **Pymavlink**: A Python implementation of the MAVLink protocol.

Open command prompt -> open pymavlink directory-> paste “pip install pymavlink”.

1. **Configure Network Settings (IPv4)**   
   Modify your network's IPv4 settings as necessary to ensure proper connectivity with the system.
2. Go to control panel > Network & Internet > Network connection
3. Right click and choose properties
4. Choose “internet (TCP/IPv4)”,then click properties
5. Click “use the following IP address” and enter 192.168.144.3”
6. Click “Subnet mask” and it will give you a Subnet mask.
7. Check if Teensy Code and Arduino code is uploaded on respective MCU.
8. **Calibrate Joystick – GCS Box**
9. Open **Setup USB Game Controller** in the Control Panel.’
10. Select serial+keyboard+mouse+joystick.
11. Calibrate the joystick to ensure accurate control input.

1. **Joystick Mapping in Mission Planner**
2. Open **Mission Planner** and map the control sticks to the appropriate channels:
3. Channel 1: Roll
4. Channel 2: Pitch
5. Channel 3: Throttle
6. Channel 4: Yaw
7. Ensure the **MIN** and **MAX PWM ranges** are correct according to the calibration results.
8. **Download Python Script**
9. Download the Python script named **GCSMAIN.py**.
10. Save the script to the following directory: C:\Users\GCS\pymavlink\GCSmain.py.
11. If the pymavlink folder does not exist, create it and then copy the downloaded Python file into this folder.

1. **Create/Import a Batch File (autorun.bat)**
2. Create or import a batch file named **Autorun.bat** to run both MavProxy and the Python program simultaneously.
3. change the batch file mavproxy command according to connection through ethernet or com port.

1. **Download QGroundControl (QGC)**
2. Download QGroundControl for telemetry and video feed.

1. **Download and Configure UGCS**
2. Download UGCS for telemetry and video streaming.
3. Modify the vsm-ardupilot.conf file in UGCS to connect to the vehicle and enable telemetry viewing.

# Procedure to connect to drone

A screenshot of a cloud

Description automatically generated

Step1 : There are 2 cases present use according to systems.

* Case1: if there is **no drone net** present on Drone

Double click on this “Autorun.bat” file saved on desktop and wait for it to run.

* Case2 : if there is **drone net** present on Drone

Double click on this “CAVAutoBOTBLOX.bat” file saved on desktop and wait for it to run.

Step 2:If you want to use multiple GCS Software on same GCS

select UDP connect and port number respectively.

14560 – localhost connection to MP.

14561 – localhost connection to python.

14562 – localhost connection to QGC.

14563 – localhost connection to UGCS.

Step 3: To view camera feed

Open VLC and go to network stream and insert RTSP stream for camera specified by manufacturer.

Open QGround Control enter RTSP Stream.