

Моя работа основывалась на совместном использовании описания методов применения патчера, указанных тут - https://dji.retroroms.info/howto/fc_patcher_tool и в оригинальном посте, - https://github.com/o-gs/DJI_FC_Patcher. Автор описания, приведённого в первой ссылке, в общих чертах всё объяснил, но, как мне кажется, намеренно упустил несколько нюансов. Так что, пришлось искать ответы в тексте оригинального поста, хотя он под Win64 не адаптирован.

В настоящем мануале приводится пример с Mavic Pro, прошивка 01.04.0300. Работа будет проводиться в среде Windows и Linux-песочнице. Для начала, подготовим рабочий каталог для размещения необходимых нам файлов. При параллельном использовании среды ОС Win и Unix, определение путей к папкам и их синтаксис очень важны. Итак, начнем.

ШАГ 1. ПОДГОТОВКА

1. Создаем каталог C:\fcpatcher;
2. Скачиваем следующие инструменты и размещаем их указанным далее образом:
 - A)** сам патч - https://github.com/o-gs/DJI_FC_Patcher/archive/master.zip. Размещаем его в корне созданной нами ранее папки. То есть, путь к файлам патчера будет выглядеть так - "C:\fcpatcher\DJI_FC_Patcher\";
 - B)** утилиты, необходимые для работы с патчером - dji_firmware_tools - <https://github.com/o-gs/dji-firmware-tools/archive/master.zip>. Размещаем папку по тому же пути - "C:\fcpatcher\dji-firmware-tools\". Удаляем "-master" из названия папки;
 - C)** скрипт для экстракции файлов конфигурации и *.sig файлов - image.py - https://github.com/fvantienen/dji_rev/raw/master/tools/image.py; Помещаем файл в C:\fcpatcher\;
 - D)** дешифратор файлов прошивки полётного контроллера - dji_mvfc_fwpack.py - https://github.com/o-gs/dji-firmware-tools/raw/master/dji_mvfc_fwpack.py. ; Помещаем файл в C:\fcpatcher\;
 - E)** скрипт для изменений лётных параметров файлов прошивки полётного контроллера - dji_flyc_param_ed.py - https://github.com/o-gs/dji-firmware-tools/raw/master/dji_flyc_param_ed.py. Помещаем файл в C:\fcpatcher\;
 - F)** дистрибутивы Python2(2.7.16) и Python3 - <https://www.python.org/downloads>. Важно отметить галочкой пункты меню установщика, задающие путь системы к данным компиляторам – SET PATH. Проверить правильную установку можно через ввод в терминале командной строки: **python**. Должен появиться текст с версией Python. Если по каким-то причинам пропустили, задаете вручную через Win-утилиту sysdm.cpl. Гуглите, всё популярно расписано;

Г) последнюю рабочую с патчером версию прошивки - 01.04.0300, используя сервис DankDroneDownloader - <https://github.com/cs2000/DankDroneDownloader>; Помещаем файл в C:\fcpatcher\;

Н) утилиту для прошивки и предоставления root-доступа Dumldore v3.20 Tool - <https://github.com/jezzab/DUMldore/releases/download/v3.20/DUMldoreV3.zip>; Помещаем папку в C:\fcpatcher\;

И) пакет для работы с Android-устройствами в Win-среде Android SDK for Windows и ADB интерфейсом - <https://developer.android.com/studio/releases/platform-tools>. Содержимое папки platform-tools копируем в корень папки fcpatcher;

Л) Linux-песочницу Cygwin64 для работы с .sh скриптами - <https://www.cygwin.com/>. **На сайте запускаем: setup-x86_64 -> Install from Internet -> All Users -> Use system proxy server -> выбираем первое зеркало для скачивания -> указываем какие пакеты скачивать (в нашем случае это bash, python2 и python3). В диалоге "View" выбираем "Full", а в меню "Search", - поочередно "bash", потом "python2" и "python37". Следим за тем, чтобы каждый раз в меню "View" стояло значение "Full". Нам нужны именно этих три пакета. В соседней колонке напротив названия нажимаем на стрелку меню «вниз» и выбираем последние версии. Нажимаем «далее», скачиваем и выходим.**

К) архиватор WinZip или WinRar;

Л) python-модуль pycrypto. Для этого в командной строке Windows PowerShell(администратор) введите: **`python -m pip install pycryptodome`** и ждите до окончания установки модуля.

ШАГ 2. РАСПАКОВЫВАЕМ ФАЙЛ ПРОШИВКИ ПОЛЕТНОГО КОНТРОЛЛЕРА

Открываем файл V01.04.0300_Mavic_dji_system.bin с помощью архиватора. Распаковываем файлы: wm220.cfg.sig; wm220_0305..... и wm220_0306..... в папку fcpatcher.

ШАГ 3. ИЗВЛЕКАЕМ ИЗ ФАЙЛА wm220.cfg.sig ФАЙЛЫ wm220.cfg.ori и wm220.cfg_0000.bin

Открываем командную строку (**ПКМ на значке Пуск – выполнить - cmd**) и вводим:

```
cd\fcpatcher <Enter>
```

в результате получаем заголовок командной строки: PS C:\fcpatcher>
далее вводим:

```
python image.py wm220.cfg.sig
```

```
move wm220.cfg_0000.bin wm220.cfg.ori
```

в результате в папке C:\fcpatcher\ появятся два новых файла - wm220.cfg.ori и wm220.cfg_0000.bin

ШАГ 4. СНИМАЕМ ЦИФРОВУЮ ПОДПИСЬ С ФАЙЛА

wm220_0306_v03.02.44.07_20171116.pro.fw.sig, в котором содержатся данные полетного контроллера

Подключаем дрон. Через утилиту DumIdore v3.20 Tool получаем root-доступ (кнопка "Enable ADB"). Далее, в командной строке вводим команду:

```
adb shell
```

В тексте начала диалога командной строки вместе с идентификатором дрона должно появиться слово root. Далее делаем раздел /vendor открытым для записи и создаем папку во внутренней памяти дрона:

```
mount -o remount,rw /vendor
```

ПОТОМ:

```
mkdir /vendor/bin
```

И ВЫХОДИМ:

```
exit
```

Перемещаем файл wm220_0306_v03.02.44.07_20171116.pro.fw.sig в новосозданную папку:

```
adb push wm220_0306_v03.02.44.07_20171116.pro.fw.sig /vendor/bin/
```

```
exit
```

Обходим верификацию и в результате получаем файл 0306.unsig. Извлекаем его с папки vendor/bin/ дрона:

```
adb pull /vendor/bin/0306.unsig
```

файл сохранится в папке C:\fcpatcher

Удаляем его и файл wm220_0306_v03.02.44.07_20171116.pro.fw.sig с дрона и возвращаем раздел /vendor к исходному состоянию:

```
adb shell
```

```
cd /vendor/bin/
```

```
rm 0306.unsig
```

```
rm *.fw.sig
```

```
cd /
```

```
sync
```

```
mount -o remount,ro /vendor
```

После этого шага у Вас снята подпись с файла `wm220_0306_v03.02.44.07_20171116.pro.fw.sig` параметров контроллера, но его еще нужно расшифровать.

ШАГ 5. ДЕКРИПТИМ `0306.unsig` с помощью скрипта `dji_mvfc_fwpack.py`.

Не забываем начинать работу с каталога `C:\fcpatcher`. Вводим:

```
cd \fcpatcher
```

Далее дешифруем файл `0306.unsig`:

```
python dji_mvfc_fwpack.py dec -i 0306.unsig
```

В папке `C:\fcpatcher` должен появиться файл `0306.decrypted.bin`

Далее перемещаем данные этого файла в файл `wm220_0306_v03.02.44.07_20171116.pro.fw_0306.decrypted.bin`, вводим одной строкой:

```
move 0306.decrypted.bin wm220_0306_v03.02.44.07_20171116.pro.fw_0306.decrypted.bin
```

ШАГ 6. Извлекаем из вышеполученного файла таблицу полётных параметров в виде файла `flyc_param_infos`.

Вводим одной строкой:

```
python dji_flyc_param_ed.py -vv -x -b 0x420000 -m  
wm220_0306_v03.02.44.07_20171116.pro.fw_0306.decrypted.bin
```

В результате данной операции, в папке `C:\fcpatcher` появится файл - `flyc_param_infos`.

ШАГ 7. Модифицируем полётные параметры.

Открываем файл `flyc_param_infos` с помощью Блокнота и правим ИСКЛЮЧИТЕЛЬНО параметр `"defaultValue"` согласно данных из таблицы:

Parameter Name	Value	Description
<code>g_config.airport_limit_cfg.cfg_disable_airport_fly_limit</code>	1	
<code>g_config.airport_limit_cfg.cfg_limit_data</code>	20250910	
<code>g_config.flying_limit.driver_license_limit_enable</code>	0	
<code>g_config.flying_limit.viechle_license_limit_enable</code>	0	
<code>g_config.flying_limit.height_limit_enabled</code>	2	
<code>g_config.airport_limit_cfg.cfg_search_radius</code>	1	
<code>g_config.airport_limit_cfg.cfg_enable[FLY_LIMIT_TYPE_AIRPORT]</code>	0	
<code>g_config.airport_limit_cfg.cfg_enable[FLY_LIMIT_TYPE_SPECIAL]</code>	0	
<code>g_config.flying_limit.limit_height_rel</code>	10000	
<code>g_config.flying_limit.limit_height_abs</code>	10000	
<code>g_config.flying_limit.limit_height_abs_without_gps</code>	10000	
<code>g_config.mode_normal_cfg.tilt_atti_range</code>	33	
<code>g_config.mode_normal_cfg.vert_vel_up</code>	7	
<code>g_config.mode_normal_cfg.vert_vel_down</code>	-6	
<code>g_config.mode_normal_cfg.vert_acc_up</code>	7	
<code>g_config.mode_normal_cfg.vert_acc_down</code>	-6	
<code>g_config.mode_sport_cfg.tilt_atti_range</code>	50	
<code>g_config.mode_sport_cfg.vert_vel_up</code>	8	
<code>g_config.mode_sport_cfg.vert_vel_down</code>	-8	
<code>g_config.mode_sport_cfg.vert_acc_up</code>	8	
<code>g_config.mode_sport_cfg.vert_acc_down</code>	-8	
<code>g_config.fw_cfg.max_speed</code>	20	
<code>g_config.avoid_cfg.avoid_atti_range</code>	23	
<code>g_config.control.avoid_atti_range</code>	50	
<code>bat_level_2_action</code>	0	
<code>bat_cap_v2_prot_type</code>	0	
<code>g_config.bat_config.level2_smart_battert_land</code>	0	

Тут указаны стандартные модификаторы, если нужно что-то дополнительно изменить – ищем нужный Вам параметр и так же меняем только значение "defaultValue".

Сохраняем файл, не меняя при этом названия и следим за тем, что б у данного файла не появилось расширения .dat или любого другого.

ШАГ 8. ПЕРЕНОСИМ ФАЙЛЫ С МОДИФИЦИРОВАННЫМИ ПАРАМЕТРАМИ НА ДРОН.

Теперь нужно внедрить изменённые полётные данные в файл прошивки дрона. Скрипт по внесению изменений написан под Bash и имеет формат .sh. Скриптовый язык Bash работает в Unix-среде, потому лучше всего использовать Linux-песочницу. Как говорилось раньше, мы будем работать в Cygwin. В моем случае, для корректной работы программы нужно было перепрописать дефолт-путь рабочего каталога, что и Вам советую. В файле параметров программы – C:\cygwin64\etc\profile с помощью блокнота после строк: \${ORIGINAL_PATH=\${PATH}}, вместо следующих 4-х строчек:

```
if [ ${CYGWIN_NOWINPATH-addwinpath} = "addwinpath" ] ; then
PATH="/usr/local/bin:/usr/bin${PATH:+:${PATH}}"
else
PATH="/usr/local/bin:/usr/bin"
fi
```

пропишите следующие:

```
PATH="/usr/local/bin:/usr/bin:/bin:/usr/sbin:/usr/local/sbin:${PATH}"
unset DOSDRIVE
unset DOSDIR
unset TMPDIR
unset TMP
```

8.1. Запускаем Cygwin64. Переходим в скопированную ранее папку (помним, что в Unix-среде пути прописываются через backslash):

```
cd c:/fcpatcher/
```

8.2. Указываем путь для папки со скриптом и другими необходимыми для него утилитами. В первоисточнике указана следующая команда:

```
set PATH_TO_TOOLS=C:\fcpatcher
```

В моем случае она не сработала. Я использовал другую:

```
export PATH_TO_TOOLS=C:/fcpatcher
```

8.3. Запускаем скрипт FC_patch_sequence_for_dummy_verify.sh, используя следующие параметры - Mavic 03.02.44.08. В окне терминала Cygwin вводим одной строкой:

```
c:/fcpatcher/DJI_FC_Patcher/FC_patch_sequence_for_dummy_verify.sh Mavic 03.02.44.08
```

Работа скрипта заключается в изменении полётных параметров (тех что в файле flyc_param_infos плюс несколько модификаций в режим Waypoint), внесение в конфигурацию GNSS ресивера Ublox модифицированных параметров для работы с спутниками системы Galileo и внесении перечисленных изменений в файл конфигурации прошивки - wm220.cfg.sig. Далее скрипт собирает файл dji_system_wm220_0306_03.02.44.08_dummy_verify.bin из модифицированных и ранее растаренных для последующей прошивки его в дрон.

Автор патчера рекомендует проверять размер и хэш MD5 полученных после работы скрипта файлов перед их прошивкой на дрон. Выглядит это следующим образом, - переименовываем dji_system_wm220_0306_03.02.44.08_dummy_verify.bin в dji_system_wm220_0306_03.02.44.08_dummy_verify.tar (ПКМ – Переименовать). Далее разархивируем данный .tar файл (внутри должно быть 3 файла!!!) и сравниваем размер и [хэш файла](#) wm220_0306_v03.02.44.08_20171116.pro.fw.sig с указанными в wm220.cfg.sig. И если всё совпало, переходим к следующему шагу.

ШАГ 9. ПРОШИВКА ДРОНА МОДИФИЦИРОВАННЫМ ФАЙЛОМ КОНФИГУРАЦИИ ПОЛЁТНОГО КОНТРОЛЛЕРА С ПОМОЩЬЮ СКРИПТА - dummy_verify.sh

9.1. Здесь используется метод замены файла верификации dji_verify на dummy_verify.sh, после чего прошивка dji_system_wm220_0306_03.02.44.08_dummy_verify.tar становится возможной.

Подключаем дрон. Активируем root-доступ через Dumldore v3.20 Tool – Активировать ADB). В командной строке Windows (уже Wincyg не нужен) вводим:

```
adb shell
```

```
mount -o remount,rw /vendor
```

```
exit
```

```
adb push dummy_verify.sh /vendor/bin/
```

```
adb shell
```

```
cd /vendor/bin/
```

```
chown root:root dummy_verify.sh
```

```
chmod 755 dummy_verify.sh
```

```
cp /sbin/dji_verify /vendor/bin/original_dji_verify_copy  
sync
```

```
cd /
```

```
mount -o remount,ro /vendor
```

После выключаем дрон.

9.2. ПРОШИВАЕМ dji_system_wm220_0306_03.02.44.08_dummy_verify.tar.

Снова включаем дрон. В командной строке Windows вводим:

```
adb shell
```

```
mount -o bind /vendor/bin/dummy_verify.sh /sbin/dji_verify
```

Затем открываем Dumldore v3.20 Tool, выбираем файл dji_system_wm220_0306_03.02.44.08_dummy_verify.tar (Load firmware) и прошиваем (Flash firmware). Время прошивки – до 5 мин.. Дрон самостоятельно перезагрузиться и по концовке отключиться.

9.3. Далее следует проверить результат работы скрипта уже на прошивке дрона. Включите дрон снова. В меню Dumldore v3.20 Tool выберите пункт “Pull the upgrade logs”. В результате выгрузятся логи, из которых Вам необходим один - upgrade00.log. Откройте данный файл и в меню блокнота – «Правка» – «Найти», и введите, опустившись в конец файла и выбрав соответственно в меню поиска «вверх», номер версии 03.02.43.21 или без нуля 3.2.43.21. Найдите рядом с этими цифрами параметры – Current ver.Fw и Target ver.Fw.. Суть проверки следующая. В логе есть два места, где упоминается номер версии. В той, что ближе к концу файла, параметры Current ver.Fw и Target ver.Fw НЕ ДОЛЖНЫ БЫТЬ ОДИНАКОВЫМИ. В моем случае, - Current ver.Fw 03.02.44.07 а Target ver.Fw - 03.02.44.08. Перед этим, значением упомянутые выше параметры была равны. Это и есть подтверждение позитивного исхода нашей работы. Конечно, главная проверка будет в полете, но эта – уже пол дела)))

Сразу после проделанного я поставил патч jkson_fcc_mod. - https://github.com/jkson5/jkson_fcc_mod. Отдельно хочу поделиться наблюдением, что после прошивки патча, БЕЗ NDL-ВЕРСИИ DJI-GO4 у меня стало ловит 25-28 спутников. Когда ранее было не больше 17-ти. Пользуюсь русифицированной

версией 4.1.22. DJI-GO-4, скачанной с этого форума. Думаю, что патч и без изменений в джиге активирует ресивер под спутники Galileo.

Надеюсь, что этим мануалом помог нашему сообществу воспользоваться своим девайсом по полной. Хотя мне больше понравился сам процесс модификации...Удачи!

