Beware of the Rabbit Hole – A Digital Forensic Case Study of DIY Drones

Samantha Klier[0009−0008−1605−3250] and Harald Baier[0000−0002−9254−6398]

Research Institute CODE, University of the Bundeswehr Munich, Neubiberg, Germany
{samantha.klier,harald.baier}@unibw.de
https://unibw.de/digfor

Abstract. In the past years, Unmanned Aerial Vehicles (UAVs), commonly known as drones, have drawn attention for commercial, military and private use and thus emerged as a significant source of digital evidence. As of today both the research community and commercial software manufacturer focus on UAV mainstream brands such as DJI, which holds a market share of roughly 80% of the civil market. On the other side Do-It-Yourself (DIY) drones are well suited for unconventional and illegal activities due to their technical capabilities and the legal limitations of commercial drones. However, the community currently lacks research of the examination of DIY drones. In this paper we address this research gap by conducting a comprehensive case study of a sample DIY drone. Our case study comprises both the assembly *and* the digital forensic perspective of DIY drones. Our systematic digital forensic examination within our case study follows the well-known process steps, i.e. preparation, acquisition, analysis. We provide insights into the peculiarities of each step and reveal that the identification of the hardware components and the corresponding examination is the most critical step.

Keywords: UAV Forensics · UAS Forensics · DIY drone · Drone Forensics.

1 Introduction

In the present era, Unmanned Aerial Vehicles (UAVs), colloquially known as drones, have become a ubiquitous form of digital evidence. A plethora of research has been conducted on this subject and a multitude of commercial software solutions are available to facilitate the accessibility of digital drone evidence. However, it is notable that the majority of drone forensics research and practice has been concentrated on the Chinese manufacturer DJI, which has indisputably secured the status of market leader with a market share of roughly 80% in the civil area [21].

While the focus on DJI is comprehensible, the construction of Do-It-Yourself (DIY) drones offers distinctive advantages if the user's intentions extend beyond the boundaries of legality or typical use cases. As an illustration, DIY drones

are extensively used in the Russian-Ukrainian War [3, 26] due to their individual configurable compilation. Despite representing a niche area within the broader market, there is now (i) a vast array of components that are readily available and (ii) an active community[1] that provides detailed construction guides and support for anyone who is interested in the assembly of a DIY drone. Consequently, any individual with a technical inclination can construct his own drone today.

Despite their potential and active military use, research on DIY drones from a forensics perspective is almost entirely absent. To address this research gap, we conduct a case study on a self-crafted DIY drone with the objective to run through the entire digital forensic process and to derive a tested procedure and toolchain.

More precisely the contributions of our case study are as follows: we review the diverse universe of hardware and software of DIY drones and point to legal aspects in this scope. We then present a methodology, which comprises two perspectives, i.e. the assembly *and* the digital forensic perspective of DIY drones with a focus on data generation. In the latter, we propose four representative scenarios including both standard scenarios and unexpected signal loss.

As a key result we present and discuss the examination results with respect to the different modules of our case study DIY UAV. First our comprehensive hardware examination reveals eight USB interfaces, some of which are concealed, as well as two external and five internal data carriers. We present details of the respective acquisition process. Second our analysis of the camera data reveals that videos are recorded both on the sending and the receiving end point. Furthermore, we provide instructions to repair recordings that were truncated due to signal loss and interpret the timestamps and file names. Finally an analysis of the remote control shows that in our case it holds only limited data, namely the SSID of the connected WiFi and some timestamps. However, under different premises, telemetry or flight logs may be found.

The rest of the paper is organised as follows: after this introduction we present the related work (being mostly case studies on DJI drones) and important background information on regulation in Section 2. Then we gather key aspects of the world of DIY drones in Section 3, with an emphasis on the technical potential of DIY drones when legal limitations are disregarded. We introduce our methodology in Section 4 followed by the in-depth examination and discussion of our sample DIY drone including all components in Sections 5 to 8. Finally, we conclude our paper in Section 9 and point to future work.

2 Background and Related Work

To ensure clarity and consistency, this section first outlines the terminology relevant to this work. Subsequently, it examines seminal works on commercial drones and drone forensics, providing a useful context for this study.

[1] For example https://oscarliang.com/

2.1 UAS or UAV

When we refer to an Unmanned Aircraft System (UAS), we mean the system of Unmanned Aerial Vehicle (UAV) extended by the equipment used alongside, such as a Radio Control (RC) and First Person View (FPV) goggles, in accordance with the definition of Commission of European Union [4]. Furthermore, the term UAV includes, as the name implies, any kind of *Unmanned Aerial Vehicle*, not just the most recognized copter types, but also e.g. wing types.

2.2 Related Work

UAS forensics is a rather new sub-discipline of digital forensics, as one of the first papers in that field was published just in 2016 [9]. The main goals include the extraction of flight paths and media recorded. Right now, the discipline is primarily based on case studies of selected drone models, primarily from the market leader DJI [2, 10, 19, 20, 27, 28]. Here, the main forensic challenge are proprietary constraints and data formats, as well, as encryption, which are not an issue when handling a basic DIY UAS (see Section 5). However, when these challenges are overcome, DJI UAS are known for their accurate and exuberant data recordings, even among other commercial UAS manufacturers [12].

Despite that, to the best of our knowledge, only one paper included a DIY UAV so far [13] whereas the specific implications imposed by DIY UAVs to digital forensics were not the focus of the work. But, they analyzed the flight path data from the DIY UAV in comparison to a DJI UAV and found that the obtained data sets from the DIY drone are "not very informative and elude towards vague patterns in flight path data" whereas the DJI drone is "more informative and enable the investigator to predict the phases of the flight journey". Indeed, the sparsity of the recorded data per default is also a concern of this paper (see Section 8).

Despite these numerous case studies efforts to systematize drone forensics date back to 2017 when [11] proposed their *Drone forensic framework* which, however, is more concerned of a general forensic procedure and less of the digital forensics part of it. In 2019 [17] proposed a *comprehensive micro UAV/Drone forensic framework* specifically for the digital forensics community. Their digital forensics subprocess comprises of the acquisition of the *Memory Card*, the extraction of the *System Logs* and the subsequent visualization, however, without further discussion of details.

Recently, [1] reviewed 32 research papers on drones with regard to their forensic procedure and merged the results in their *Comprehensive Collection Analysis and Forensic Model (CCAFM)*, where the "Evidence Acquisition Stage" incorporates three processes, namely "Live Acquisition", "Post-mortem Acquisition" and "Hybrid Acquisition". Although mapping comprehensively, all processes related to the forensic investigation of UAS, the CCAFM offers no guideline on how sub-processes, such as *Post-Mortem Acquisition*, should be executed.

Consequently, investigators must rely on general acquisition procedures if UAS model specific information is not available. While this seems to be straightforward at first glance, as e.g. every digital forensics practitioner is able to acquire

an SD card, its way more complex than that, especially for DIY drones, as we discuss in Section 5.

2.3 EU Regulations for UAS

We now give an overview on the legal framework which is based on the EU regulations Commission of European Union [4], Commission of European Union [5] and further amendments, as listed in EASA [7]. Furthermore, we assume that offenders refrain from soliciting the endorsement of the regional aviation authority to be able to fly legally in the "specific" or "certified" category, hence, the following remarks are based on the rules of the so-called "open" category.

Remote-ID. Today it is mandatory to officially register a UAVs and the obtained *eID* must be placed on the drone. Consequently, as long as the drone is flying, this information can not be accessed by Law Enforcement Agencies (LEAs), so to remedy this situation, the *Remote-ID* has been introduced, which is consitently sent via radio or network, so that the operator can be identified by LEAs at any time. The idea is similar to the systems used for flight control of airplanes. Therefore, the Remote-ID includes the eID assigned at registration and the serial number, but also its position, the position of the remote control, the take-off point, the flight speed, and the flight direction. Obviously, this is highly disadvantageous for any covert operation of a UAV.

Automatic and Autonomous UAVs. Many UAVs are able to fly automatically which is legal in the EU without strict requirements on the UAS [5]. However, automatic is not autonomous. The first means that a flight mission is planned in advance and the UAS automatically completes the mission while a remote control must be connected to the UAV. Whereas the latter means that the drone is able to react to unforeseen events by itself, hence, is allowed to fly without a connected remote control. Consequently, similar to autonomous cars, autonomous UAS have to be officially certified [5], hence, only few autonomously operating drones for industrial purposes are available on the market right now.

Maximum Range and Altitude. The maximum altitude in the EU is 120m which must be enforced by the drone manufacturer as of 2024. The range is limited by the decree that it is only allowed to fly within sight. However, DJI states that from a technical viewpoint, e.g. the Mavic 3 Classic [6] has a maximum range of 3km and a maximum altitude of 5km while implementing the EU restrictions on the transmission power of the radio (i.a. 25mW). Consequently, from a technical viewpoint UAVs can fly considerably further and higher than the legal restrictions allow, hence, are clearly limited in their capabilities by the current regulations.

3 The Exciting and Chaotic World of DIY UAS

In this section we introduce the infinite possibilities when building their own UAS. Although, they must also comply with applicable regulations, private man-

ufacturers can simply choose not to. Consequently, the capabilities of DIY UASs are only limited by the available components and skills.

3.1 Evading Regulations

Likely, the attractiveness of DIY UAS for e.g. criminal purposes will increase as the technical possibilities expand while the regulations become stricter(see Section 2). Interestingly, no additional skills or special components are required to circumvent most of the regulations when building a UAS. For example, the transmission power of the radio can be configured between 10mW and 2W [23], which is eight times the legal limit in the EU and results in a range of up to 30km. For some regulations, such as the Remote-ID, even an additional component, installation and configuration effort is required to comply.

An autonomous UAV is also easy to build, since the simplest autonomous UAV imaginable is an automatic UAV that can operate without an active connection to a remote controller. Thus, open source UAS software such as Ardupilot can be easily configured to ignore a missing radio link [25] and even provide obstacle avoidance capabilities [24]. Therefore, this offers tempting prospects for covert operations, as an operator does not need to be close to the UAV, and the range of the UAV is limited only by the power supply.

3.2 Hardware of DIY UAS

DIY UAS can be customized for peculiar purposes and to execute a wide range of operations which is only limited by the availability of resources, most notably of qualified technicians.

Basic Example. However, a minimal working UAS can be built cheaply and quickly with just a few components, namely a Radio Control (RC) and the UAV with a Flight Controller (FC), Electronic Speed Controller (ESC) and a Radio Receiver (RX), as well as a frame, motors with propellers and a battery. The RC sends commands from the pilot to the UAV, which are received by the RX and delegated to the FC. The FC, in turn, is the brain of the UAV and translates commands from the RC, based on its sensors such as gyroscopes and accelerometers, into commands for the ESCs (may be included in the FC). The ESCs then regulates the speed of the UAV's motors to achieve the behavior requested by the pilot.

The size of such a minimally equipped UAV can be as small as 90mm and easily up to 800mm, measured diagonally from motor to motor, respectively. These basic UAS can be built based on a plethora of tutorials that are available on the internet, whereas some claim to be as cheap as 150 $ [8]. Furthermore, a UAS can be built very unconcerned and sloppily, for example based on plywood [14]. However, With such a UAS offenders are capable to interfere with air traffic, frighten crowds or execute kamikaze operations. Furthermore, such a UAV can be equipped with any kind of payload, such as explosives, and may also drop the payload over a target area [26].

Endless Possibilities. However, the given minimal example of a UAS can be expanded in numerous ways, most prominently, by adding a GPS and camera system which can be a cheap and lightweight low resolution analogue camera up to a fully fledged camera system with infrared, thermal and 4K resolution. Furthermore, with systems like Pixhawk or BeagleBone which offer i.a. UART, CAN, I2C, USB and WiFi interfaces, by and large any electronically component can be utilized. Furthermore, DIY UAVs can also be built as wing type which are characterized by their very low energy requirements. For instance, it is possible to build a winged UAVs capable of staying in the air for a whole day, since a few mounted solar panels provide enough energy, as demonstrated by several hobby projects [15, 16].

3.3 Software of DIY UAS

Additionally, to the hardware a flight control software is necessary whereas at the moment, the following three are most prominent:

- **PX4** which is supported by the Linux Foundation, provides a professional ecosystem and is also used by commercial drone manufacturers, such as Yuneec.
- **ArduPilot** is licensed under the GPLv3, hence, is subject to the copyleft principle which makes it uninteresting to commercial drone manufacturers, but attractive to hobby UAV builders.
- **Betaflight** is also licensed under the GPLv3, but, in contrast the focus is on piloting and agility which results in a limited range of supported sensors and no capabilities for automatic missions.

However, based on a flight control software, such as ArduPilot, technical capable offenders can build extensive functionality on top. For example, there is a project[2] that shows how a ArduPilot UAV can be combined with the machine learning frameworks OpenCV and YOLO[3]. Consequently, fully autonomous UAVs capable of, e.g. identifying targets to attack, are possible. Although these projects exist, the effort to realize a UAV with sophisticated autonomous functions is significant. However, once successfully built and programmed, manufacturing more UAVs is straightforward.

4 Methodology of Case Study

The methodology used in this case study is defined by two major phases: (i) design, build, configuration and test, and (ii) data generation and forensic examination (incl. acquisition and analysis) which are separated by the shift of perspective.

[2] https://github.com/Intelligent-Quads/iq_tutorials
[3] https://opencv.org/, https://pjreddie.com/darknet/yolo/

4.1 Design, Build, Configuration and Test

This first phase is characterized by a builders and operator's point of view, which means that the sole focus is on building UAS that meet the requirements of a mission. Hence, the goal was not to record as much data, as possible, as digital forensic scientists may be inclined to do, hence, different team members were assigned to the configuration and the examination of the next phase. Thus, our UAS were manufactured realistically.

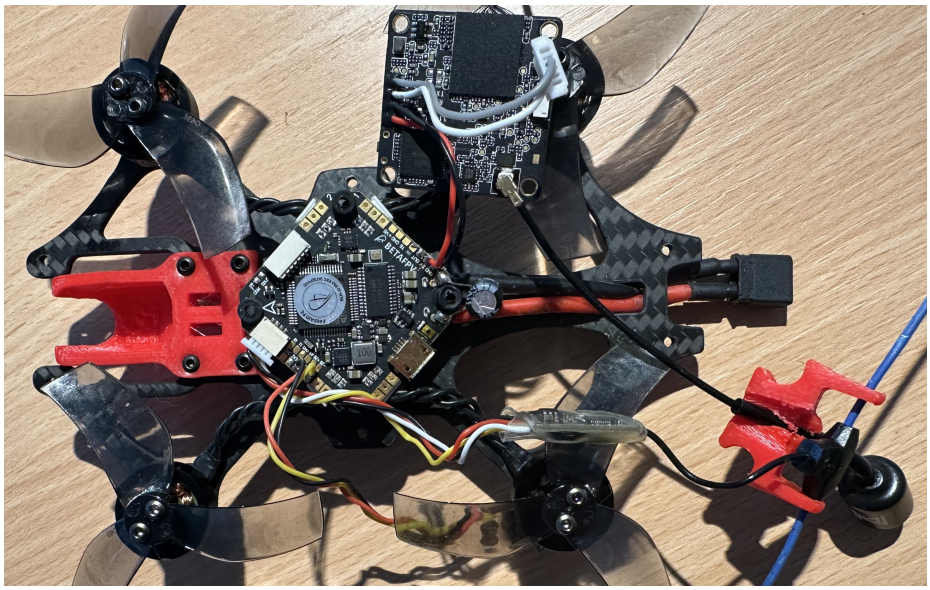

Fig. 1. Half assembled small and cheap DIY UAV with digital HD camera.

However, in this paper, we focus on our first DIY UAS which is easy to setup, small and light, and, as such should provide outdoor and indoor "in the middle of the action" camera footage, e.g. for reconnaissance and may also carry a small amount of explosives. Therefore, the UAV needs to be cheap to make a loss in action bearable while providing a good camera footage quality. Based on the input of several DIY drone communities[4], we designed and built a UAV with a HD camera, for under 200 EUR[5] which is shown in Figure 1.

However, to complete the UAS, a compatible remote control is required and we have also selected FPV goggles as they are advantageous in this scenario. This time, to maximize range, reliability and footage quality we resort to high end components, as in an operation, only the UAV is in danger of damage or loss. However, cheaper compatible components are available on the market. The

[4] https://oscarliang.com/, https://www.reddit.com/r/diydrones/
[5] Market prices fluctuate strongly.

process of building and configuring the UAV, was straightforward, but to tweak the flight performance of the UAV, several cycles of adapting the configuration and testing it, were required.

As the design, build and test phase was completed, we acquired the data of all UAS components and subsequently wiped the data carriers, before we entered the next phase. For the complete technical specification of the UAS see Section A.

4.2 Data Generation and Forensic Examination

We developed four typical scenarios for our experiments with the given UAS, as shown in Table 1. The first scenario, is a standard operation which includes a successful return of the UAV and a recording that was started and stopped with dedicated record button of the FPV goggles. Additionally, we tested loss of connection during recording on the UAV and FPV side, which can occur due to power loss, other technical problems, or during a kamikaze mission. Finally, we have a scenario for the corner case that a recording was started but the UAV was not armed or flying, which can happen when a take off is interrupted or the recording is started accidentally.

Based on these scenarios, we conducted eleven controlled experiments. Finally, the UAS was admitted to our lab and treated as digital forensic evidence.

Table 1. The scenarios tested with the UAS. UAV, FPV Goggles, RC are on and operational in each scenario.

Scenario	Description
STD	Recording of the FPV goggles is started. UAV is armed and started. Flight is executed. UAV is landed and disarmed. Recording is stopped. UAV is turned off.
UAV_LOST	Recording of the FPV goggles is started. UAV is armed and started. Flight is executed. UAV is disconnected from the power source.
FPV_LOST	Recording of the FPV goggles is started. UAV is armed and started. Flight is executed. FPV Goggles are disconnected from the power source.
NO_FLIGHT	Recording of the FPV goggles is started. Recording is stopped.

5 Digital Forensic Examination and Acquisition

In the lab, we conducted a thorough examination of the hardware to identify relevant data carriers and interfaces for the next step, the acquisition. The results are illustrated in Figure 2 which are now discussed in detail.

5.1 Hardware Examination

UAV. The hardware examination of the UAV revealed two USB interfaces to the FC, the micro USB interface on the top is easily accessible, in contrast to, the

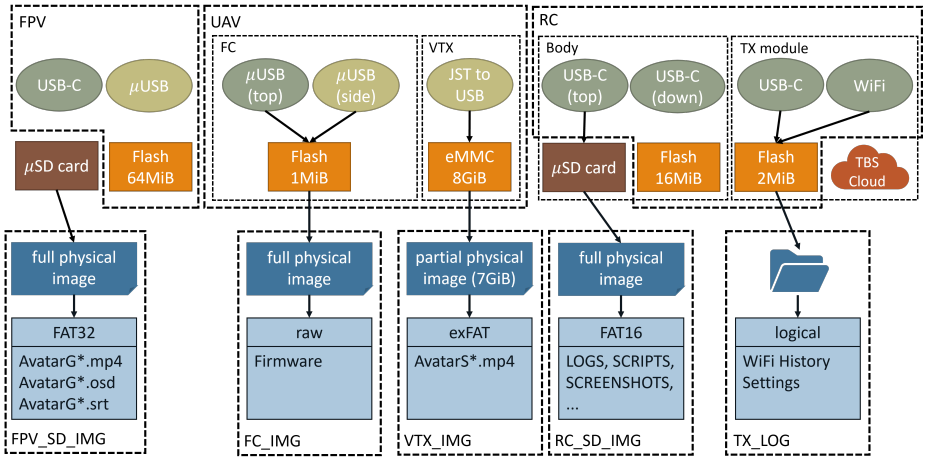

Fig. 2. Overview of (sub-)components, interfaces (light color indicates being hidden), data carriers (external: brown, internal: orange) and the acquired data whereas the arrows indicate the connection between these entities.

semi hidden micro USB on the side. However, no internal storage was identified besides the memory of the `STM32F405RG` chip.

Another semi-hidden interface on the side, of type JST, belongs to the VTX which core piece is a chip `KMFN60012B-B214` for mobile devices which includes 8GiB of Embedded Multi Media Card (eMMC) storage. Interestingly, the manual of the VTX shows that the JST interface translates to a 4-pole standard USB interface, hence, a compatible cable can be easily manufactured e.g. by cutting a standard 4-pole USB cable and connecting it to the respective pins.

FPV Goggles. Here, the FPV goggles provide a micro SD card slot and a USB-C interface, for "HDMI output" as stated by the user manual. Additionally, dismantling the FPV goggles revealed a concealed mini USB connection to the mainboard, as shown in Figure 3 which is not accessible externally, and unusually well marked JTAG connector pads. However, only one built-in flash memory of 64MiB was found which strongly indicates that video recordings are not saved internally.

Remote Control. The remote control consists of the body and the inserted TX module. The body has two USB-C ports, one for charging and one to access the micro SD card, which must be inserted for the body to function. For this reason, we decided not to disassemble the body and simply removed the micro SD card. However, the TX module does have its own USB-C interface, which powers the module when it is removed from the body. A WiFi module and a 2MiB flash memory were also found during disassembly.

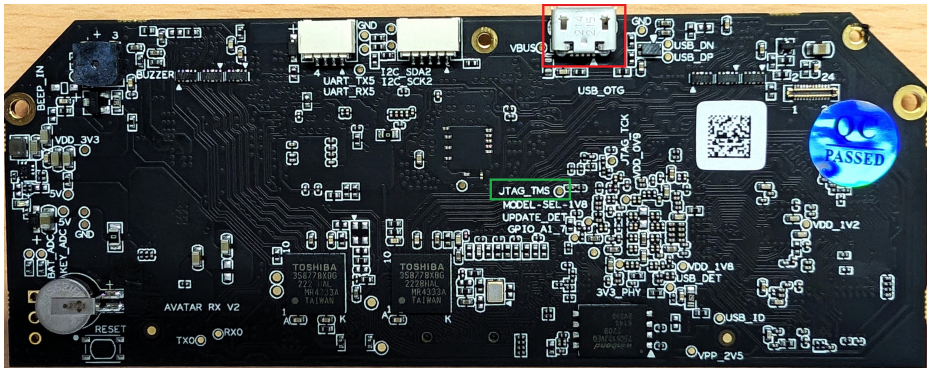

Fig. 3. Backside of the FPV goggles mainboard with concealed mini USB interface (red) and one exemplary JTAG connector (green).

5.2 Acquisition

UAV. Surprisingly, the internal storage of the VTX can easily be acquired physically via the "JST to USB" interface even with the use of a write blocker, as it is recognized as an ordinary mass storage device. However, following the standard procedure, we only acquired a partial physical image with 7GiB of 8GiB to which we refer as `VTX_IMG` (see Figure 2). However, it is of utmost importance to ensure sufficient cooling (e.g. with a cool pad or external fan) while acquiring the VTX, as it overheats quickly when not flying, and consequently, shuts down to prevent damage. Additionally, it is worth mentioning, that the VTX can only be acquired if the UAV is connected to its battery, as the power supply from the USB is not sufficient.

In contrast, the FCs internal storage can be acquired physically and completely when booted into its Device Firmware Update (DFU) mode, which is initiated by holding the so-called *boot* button on the board, when plugging the USB cable in. However, the usage of a write blocker, such as *Tableau T8u*[6], is not possible, but, the DFU mode inherently impedes data changes and enables the acquisition of a complete physical image of the internal memory by, e.g. using the firmware development software *STM32CubeProgrammer*[7]. We refer to this image as `FC_IMG`. Oddly, both USB interfaces (see Figure 2) provide access to the same storage, and yield hash identical images.

FPV Goggles. First off, the micro SD card was extracted from its slot and acquired straightforward, yielding the image `FPV_SD_IMG`. Although, the SD card will be the main source of usage data, there is a 64MiB flash storage built-in which may not just contain the goggles firmware. Therefore, we examined the

[6] "T8u supports USB devices that conform to the USB Mass Storage Bulk-Only specification", as stated by its manual.

[7] https://www.st.com/en/development-tools/stm32cubeprog.html

USB interfaces to complete the data acquisition, which, was not successful, but we were able to gather some information.

Despite the fact, that the USB-C interfaces supplies the goggles with power which the micro USB interface does not, no major difference was ascertainable and we were not able to acquire the internal storage by any of the two. However, kernel messages of our forensic workstation[8], could be observed when the device is connected (via any USB interface) first, and then powered on, whereas the device is not recognized when its powered on and connected afterwards.

Therefore, as shown in Listing 1.1 we are provided with a serial number, a product name and the dedicated manufacturer (*Artosyn*) which is known for supplying DJI with components [22]. Furthermore, the device registers as a RNDIS host which provides a virtual Ethernet over USB, hence, is registered as network interface (see Listing 1.3 in Section B for the bash records). However, an IP address must be manually assigned, but unfortunately, a port scan showed no relevant open ports. Due to the small size of the internal memory, it presumably contains only the firmware of the goggles, as they show a GUI when booted, we decided to let it be. But, as stated, a JTAG acquisition of the internal NOR storage is a possibility if the firmware is of interest in a given case.

Listing 1.1. Kernel messages retrieved for the FPV goggles. Serial number is obscured. Interface name changes on each connection.

```
$ sudo dmesg
[   63.590203] usb 1-2: new high-speed USB device number 4 using xhci_hcd
[   63.759012] usb 1-2: New USB device found, idVendor=1d6b, idProduct=0104, bcdDevice= 3.10
[   63.759094] usb 1-2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   63.759111] usb 1-2: Product: Sirius
[   63.759124] usb 1-2: Manufacturer: Artosyn
[   63.759135] usb 1-2: SerialNumber: ZBBM5***MP
[   63.830255] hid-generic 0003:1D6B:0104.0002: hiddev0,hidraw1: USB HID v1.01 Device [
     ↪ Artosyn Sirius] on usb-0000:00:14.0-2/input2
[   63.830346] usbcore: registered new interface driver usbhid
[   63.830349] usbhid: USB HID core driver
[   63.832721] usbcore: registered new interface driver cdc_ether
[   63.836540] rndis_host 1-2:1.0 usb0: register 'rndis_host' at usb-0000:00:14.0-2, RNDIS
     ↪ device, c2:af:08:37:e5:5d
[   63.836583] usbcore: registered new interface driver rndis_host
[   63.846985] rndis_host 1-2:1.0 enxcaddb23e0d15: renamed from usb0
```

Remote Control. Again, the micro SD card of the remote control's body was simply extracted and handled typically. Furthermore, the body inhabits a 16MiB internal flash memory which is not addressable with the given interfaces, hence, an acquisition would require intrusive measures.

However, the data stored in the internal flash memory of the TX module can be viewed, but also not acquired. On the one hand, the TX module provides a WiFi access point and a WebUI to connected devices at `192.168.4.1`. On the other hand, the manufacturers software *TBS Agent*[9] can be used to connect to the device via USB-C.

[8] Kubuntu 24.04 LTS, 6.8.0-31 generic x86_64, auto mount disabled
[9] https://www.team-blacksheep.com/products/prod:agentx

5.3 Acquisition Results

Finally, the acquired images were reviewed, as shown in Table 2. To sum up, the SD cards are conventionally partitioned with a DOS partition scheme and can be analyzed with standard tools due to their FAT file systems. Unusually, the SD card, as formatted by the RC, has a FAT16 file system, which is, however, no challenge for standard forensic tools. In contrast, the mass storage of the VTX has no partition scheme which is also unusual but no hindrance for an analysis with standard tools, as there is direct access to an exFAT file system. In contrast, the image of the flight controller, is a firmware image which is not changed during operation, hence, will not be considered any further in this work.

Table 2. The acquired images with initial assessment.

Storage	Size (B)	Partition Scheme	File System	Operational Data
FPV_SD_IMG	31,267,487,744	DOS	FAT32	yes
FC_IMG	1,048,76	none	none	no
VTX_IMG	7,503,068,672	none	exFAT	yes
RC_SD_IMG	504,365,056	DOS	FAT16	yes

6 Analysis of UAV and FPV goggles

As there is no GPS module, the most interesting data will be the video files and corresponding data which can be found on FPV side (`FPV_SD_IMG`) and UAV side (`VTX_IMG`), respectively, in the root directory of the dedicated file system.

Generally, the recorded data that can be found on both sides differs. Most importantly, on FPV side there are not only video files (in MP4 format) saved, but also additional information from the On Screen Display (OSD) which are missing on the VTX side[10]. Now, we will discuss the available metadata of the saved files, before addressing the primary data.

6.1 Filenames, Timestamps and Metadata

The metadata will be analyzed on the example of the second controlled flight of which all generated files are shown in Table 3.

Table 3. File system information of the files created for the second controlled flight.

Image	Path	Filename	Modified Time	Access Time	Created Time	Size (B)
VTX_IMG	/	AvatarS0014.mp4	2000-01-01 00:06:24	2000-01-01	2000-01-01 00:06:24	208,586,788
FPV_SD_IMG	/	AvatarG0001.mp4	2000-01-01 00:06:26	2000-01-01	2000-01-01 00:06:26	204,418,187
FPV_SD_IMG	/	AvatarG0001.osd	2000-01-01 00:06:26	2000-01-01	2000-01-01 00:06:26	450,328
FPV_SD_IMG	/	AvatarG0001.srt	2000-01-01 00:06:26	2000-01-01	2000-01-01 00:06:26	26,089

[10] A detailed list of all camera related files can be downloaded from our cloud storage.

Filenames. On the FPV side MP4, SRT and OSD files always appear together and adhere to the naming scheme `^AvatarG[0-9]{4}.(mp4|srt|osd)$`. However, on VTX side only a MP4 file of this flight is saved and instead of a `G` an `S` is present in the filename. On both sides, the four digits act as a counter, which start by 0 and is incremented by one with each recording. Interestingly, the empty SD card on FPV side leads to the counter to be reset, with the first controlled flight. In contrast, the data storage on VTX side was also erased, but, the counter still remembers that 13 videos have been recorded before.

Timestamps and Metadata. In the file system, all files have a created, last modified and last access timestamp which inexplicable points to the `2000-01-01`, also, the flights were not conducted shortly after midnight. The same is the case for the timestamps of the metadata of the MP4 files, as extracted by ExifTool[11]. Although, we have no explanation for the date, the time on both sides, represents the uptime of the FPV goggles. Therefore, from the metadata of the video files the approximate uptime when the recording started (i.e. `5:54`) can be obtained, and, from the file system, when the recording stopped (i.e. `6:24`). This information can be verified with the duration of the video (i.e. `0:32`s).

Table 4. Timestamps of the files created for the second controlled flight, based on the metadata embedded in the video files.

Filename	CreateDate	ModifyDate	Duration
AvatarG0001.mp4	2000:01:01 00:05:54	2000:01:01 00:05:54	0:00:32
AvatarS0014.mp4	2000:01:01 00:05:53	2000:01:01 00:05:53	0:00:32

6.2 Primary Data

The primary data recorded by the UAS is the camera footage and the corresponding OSD information.

Video Files. When the user starts a recording, a video is usually saved on VTX and FPV side, respectively. However, these recordings are not identical, e.g. the resolution on VTX side (i.a. 1920x1080px) is higher than on the FPV side (i.a. 1280x720px). Despite that fact, further difference are shown in Table 5, respectively, in relation to the executed scenario (see Table 1). Due to the counting differences, as discussed in Section 6.1, the video files have been joined by executed operation, as identified by the footage, rather than the filenames.

First off, the recordings of the two flights of the standard scenario differ only slightly, in contrast to the flights of the `UAV_LOST` or `FPV_LOST` scenario, as in these case, the MP4 files are truncated on the "lost" side. Fortunately, the video files can be repaired, e.g. with `untrunc`[12] which, uses a reference file from the

[11] https://exiftool.org/, version used: 12.65
[12] https://github.com/anthwlock/untrunc

same device to rework the container structure. But, the videos repaired in this way, are approximately 10 seconds shorter than their untruncated counterparts. Therefore, in the event of an abrupt interruption of a recording, a significant amount of data will be missing and a more sophisticated approach may be required.

However, this suggests, in concordance to the fact that no deleted files can be found on the storage, that the MP4 files are written to the storage without an intermediate temporary file or buffering.

Table 5. Juxtaposition of video files found on FPV side and UAV side.

Cr./Mod. Timestamp	Dur.	Trunc.	File of VTX_IMG	Scenario	File of FPV_SD_IMG	Trunc.	Dur.	Cr./Mod. Timestamp
2000-01-01 00:03:34	02:11	yes	AvatarS0013.mp4	UAV_LOST	AvatarG0000.mp4	no	02:14	2000-01-01 00:06:26
2000-01-01 00:06:24	00:32	no	AvatarS0014.mp4	STD	AvatarG0001.mp4	no	00:31	2000-01-01 00:06:26
2000-01-01 00:09:40	00:08	yes	AvatarS0015.mp4	UAV_LOST	AvatarG0002.mp4	no	00:15	2000-01-01 00:09:46
			-	NO_FLIGHT	AvatarG0003.mp4	no	00:02	2000-01-01 00:10:30
2000-01-01 00:11:26	00:48	no	AvatarS0016.mp4	STD	AvatarG0004.mp4	no	00:48	2000-01-01 00:11:26
2000-01-01 00:11:44	00:11	yes	AvatarS0017.mp4	UAV_LOST	AvatarG0005.mp4	no	00:21	2000-01-01 00:11:56
			-	NO_FLIGHT	AvatarG0006.mp4	no	00:02	2000-01-01 00:16:58
2000-01-01 00:17:58	00:37	yes	AvatarS0018.mp4	UAV_LOST	AvatarG0007.mp4	no	00:46	2000-01-01 00:18:10
			-	NO_FLIGHT	AvatarG0008.mp4	no	00:06	2000-01-01 00:19:38
2000-01-01 00:20:54	01:14	no	AvatarS0019.mp4	FPV_LOST	AvatarG0009.mp4	yes	01:06	2000-01-01 00:20:48
2000-01-01 00:06:06	01:36	yes	AvatarS0020.mp4	UAV_LOST	AvatarG0010.mp4	no	01:43	2000-01-01 00:06:16

Flight Information. Additionally, to every MP4 file on FPV side there is a corresponding SRT and OSD file (see Table 3) which are plain text and binary files, respectively. The SRT files provide subtitles which can be replayed alongside the dedicated MP4 files, e.g. with the VLC player[13], as shown in Figure 4. Also, the SRT files can be analyzed quantitatively due to their very simple structure, as stated by Rodriguez-Alsina et al. [18]: "Each subtitle entry consists of the subtitle number, the time at which the subtitle should appear on screen, the subtitle itself, and a blank line to indicate the subtitle's end".

For example, in Listing 1.2 an excerpt of `AvatarG0001.srt` and the saved flight information is shown. Therefore, even without GPS we are provided with an accurate approximation of the distance between VTX and FPV goggles, in this case, 11m, as well, as the duration of the flight independently from the duration of the recording. Furthermore, we can check if e.g. a crash of the UAV was due to low battery. However, the transmitted information depends on the particular UAV and its configuration.

Listing 1.2. Excerpt of AvatarG0001.srt

```
208
00:00:31,049 --> 00:00:31,199
Signal:4 CH:1 FlightTime:22 SBat:5.0V GBat:16.5V Delay:25ms Bitrate:25.0Mbps Distance:11m

209
00:00:31,199 --> 00:00:31,349
Signal:4 CH:1 FlightTime:22 SBat:5.0V GBat:16.5V Delay:25ms Bitrate:25.0Mbps Distance:11m
```

[13] Menu: Subtitle - Add subtitle file..., https://www.videolan.org/

Fig. 4. AvatarG0000.mp4 with overlaid subtitle, as displayed by VLC.

In contrast, the binary OSD file can be used to replicate the display of this information as it was displayed during flight, e.g. by the Walksnail OSD Tool[14]. However, this re-rendered video may not include all available information, as the OSD is normally configured to show only an excerpt of the transmitted information. Therefore, from a forensics perspective, the re-rendering of the video file with the OSD file is only reasonable when it is important to determine which information the operator of the UAV had while flying.

7 Analysis of Remote Control

The body and the TX module are independent systems from different manufacturers which will be analyzed separately.

7.1 The Body's SD card

The SD card of the body contains twelve directories, i.a. named `LOGS` and `SCREENSHOTS`[15]. Due to the fact, that we used the remote control in standard configuration and only for controlling the UAV, our tests did not generate files of interest in these directories. However, we had to store our configuration for our UAV in the RC which is saved in the `MODELS` directory as `model16.yml`. Despite the name and a customizable icon, only the sensitivity of the joysticks

[14] https://github.com/avsaase/walksnail-osd-tool
[15] A detailed list of all files and directories can be downloaded from our cloud storage.

and such parameters are saved. But, the body has an internal clock, and the timestamps of the file and directory are updated, even when the configuration is only activated which can be a hint to the time of the last operation.

7.2 The TX Module

The configuration of the TX Module can be viewed with the "TBS Agent Desktop" software which includes a Telemetry and a Log Viewer and allows to configure the manufacturers cloud service which, however, were empty or disabled here. Therefore, the most interesting data that was saved in the TX Module is the SSID of the WiFi it was connected to, which is accessible in the WiFi category of the WebUI. Furthermore, e.g. the serial number, MAC address an firmware version can be viewed.

8 Discussion and Practical Implications

The aim of our case study was to investigate DIY UAS from a forensics perspective and our results reveal that the challenges are indeed distinct to popular consumer market UAS. One important result is, that DIY UAS can be incredibly sparse in the data they record even if they appear to be data-rich, as seen with the eight USB interfaces in our example. Therefore, if we opted to not record videos, no data of an operation could have been found on the UAV. Therefore, in practice the hardware examination is tremendously important to avoid falling down a rabbit hole.

Consequently, the first step should be the identification of components and sub-components, such as cameras, VTXs and GPS modules, focusing on those that may provide valuable evidence. Moreover, the identification process should guide the forensic strategy toward relevant interfaces for data acquisition. However, simply relying on SD cards or USB sockets may be fatal, as the UAV's VTX demonstrates. This component, although seemingly peripheral, and with an unusual JST socket, might hold key information that other interfaces do not capture. Finally, it must be kept in mind that a DIY UAS could be built completely different, so that any kind of data may be present. These findings underscore the necessity for a structured approach in the forensic examination of DIY UAS.

9 Conclusion and Future Work

In summary, DIY UAS provide unique opportunities for operations beyond the legal scope, while the existing body of research on consumer market UAS is not applicable. Moreover, each DIY UAS is an original, ranging in complexity from bare firmware to full-fledged AI-based IT system. Therefore, to enable a successful forensic examination, we pioneer by building a unique DIY UAS to present a

complete case study. As a result, we point out the identified challenges and recommend a general yet rudimentary approach. However, the main characteristic of DIY UAS is their diversity and imponderability.

Therefore, our next step is to build further models with emphasis on market coverage and introduction of more complex software functionalities. Additionally, we will perform JTAG and chip-off procedures for otherwise not addressable data carriers to conclude the data acquisition. Finally, our aim is the proposal of a process model and a tool chain for the forensic examination of DIY UAS, based on empirical research.

Acknowledgements. We would like to express our sincerest gratitude to Mario Winkler for his invaluable contributions to the construction and operation of the drone, as well as for his unwavering support throughout the course of this research.

This work has been developed in the project FOCUS. FOCUS (reference number: 13N16510) is partly funded by the German ministry of education and research (BMBF) within the research programme "Anwender– Innovativ: Forschung für die zivile Sicherheit II".

A Configuration of the DIY UAS

Table 6: Configuration of the DIY UAS, separated by main component.

UAV	
Flight Controller	F405 AIO 20 A Toothpick V4
ESC	incl. in FC
BEC	incl. in FC
GPS	none
Compass	none
Video Transmitter	Walksnail Avatar HD Mini 1s Kit
Motors	1404 4500KV Brushless Motors
Propellers	Gemfan D63 3-Blade Propellers 1.5mm
Radio Receiver	TBS Crossfire Nano Receiver RX SE
Battery	Tattu 4s 450mAh 75C Lipo XT30
Flight Control Software	Betaflight 4.4.3
Remote Control	
Body	RadioMaster TX16S MAX MKII Hall 4.0 4in1
Transmission Module	TBS CrossfireTX V2
FPV	
FPV Goggles	Walksnail Avatar HD

B Details of the FPV's Mainboard Acquistion Procedure

Listing 1.3. Acquisition procedure of the FPV mainboard due to the concealed micro USB interface.

```
$ ifconfig
[...]
enxcaddb23e0d15: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.122.176  netmask 255.255.0.0  broadcast 169.254.255.255
        ether ca:dd:b2:3e:0d:15  txqueuelen 1000  (Ethernet)
```

```
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 292  bytes 58407 (58.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
[...]
$ sudo nmap -A 169.254.122.176
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-02 14:15 CEST
Nmap scan report for * (169.254.122.176)
Host is up (0.00010s latency).
All 1000 scanned ports on * (169.254.122.176) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops
Nmap done: 1 IP address (1 host up) scanned in 1.84 seconds
[...]
$ sudo nmap -sU 169.254.122.176
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-02 14:16 CEST
Nmap scan report for * (169.254.122.176)
Host is up (0.0000040s latency).
Not shown: 998 closed udp ports (port-unreach)
PORT     STATE         SERVICE
631/udp  open|filtered ipp
5353/udp open|filtered zeroconf
Nmap done: 1 IP address (1 host up) scanned in 1.37 seconds
```

C List of Acronyms

CCAFM	Comprehensive Collection Analysis and Forensic Model
RC	Radio Control
RX	Radio Receiver
DFU	Device Firmware Update
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aircraft System
LEAs	Law Enforcement Agencies
eMMC	Embedded Multi Media Card
FC	Flight Controller
ESC	Electronic Speed Controller
FPV	First Person View
OSD	On Screen Display
DIY	Do-It-Yourself

Bibliography

[1] Alotaibi, F.M., Al-Dhaqm, A., Al-Otaibi, Y.D., Alsewari, A.A.: A comprehensive collection and analysis model for the drone forensics field. Sensors **22**(17), 6486 (2022)

[2] Barton, T.E.A., Hannan Bin Azhar, M.: Forensic analysis of popular UAV systems. In: 2017 Seventh International Conference on Emerging Security Technologies (EST), pp. 91–96, IEEE, IEEE Press, Canterbury, UK (2017), https://doi.org/10.1109/EST.2017.8090405

[3] Boenke, M., Eckstein, C., Ehmann, A., Friederichs, H., Kireev, M., Luther, C., Prost, A., Tröger, J., Vooren, C., Gutheil, B.: Sie kreisen, sie jagen, sie töten. Die ZEIT (2024), URL https://www.zeit.de/politik/ausland/2024-02/ukraine-krieg-drohnen-truppengattung-fpv

[4] Commission of European Union: Regulation (EU) 2019/945 (2019), URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019R0945

[5] Commission of European Union: Regulation (EU) no 2019/947 (2019), URL https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R0947

[6] DJI: Specs – DJI Mavic 3 Classic (2024), URL https://www.dji.com/global/mavic-3-classic/specs

[7] EASA: Easy Access Rules for Unmanned Aircraft Systems (2022), URL https://www.easa.europa.eu/document-library/easy-access-rules/online-publications/easy-access-rules-unmanned-aircraft-systems

[8] FPV, D.: Build This Quality Freestyle FPV Drone For $150 (2024), URL https://www.youtube.com/watch?v=Ti9qMJ4LNiI

[9] Horsman, G.: Unmanned aerial vehicles: A preliminary analysis of forensic challenges. Digital Investigation **16**, 1–11 (2016)

[10] Husnjak, S., Forenbacher, I., Peraković, D., Cvitić, I.: UAV forensics: DJI mavic air noninvasive data extraction and analysis. In: 5th EAI International Conference on Management of Manufacturing Systems, pp. 115–127, Springer, Springer Cham, Cham (2022), ISBN 978-3-030-67241-6, https://doi.org/10.1007/978-3-030-67241-6_10

[11] Jain, U., Rogers, M., Matson, E.T.: Drone forensic framework: Sensor and data identification and verification. In: 2017 IEEE Sensors Applications Symposium (SAS), pp. 1–6, IEEE, IEEE Press, Glassboro, NJ, USA (2017), https://doi.org/10.1109/SAS.2017.7894059

[12] Kumar, R., Agrawal, A.K.: Drone GPS data analysis for flight path reconstruction: A study on DJI, Parrot & Yuneec make drones. Forensic Science International: Digital Investigation **38**, 301182 (2021)

[13] Mekala, S.H., Baig, Z.: Digital Forensics for Drone Data – Intelligent Clustering Using Self Organising Maps. In: Future Network Systems and Security: 5th International Conference, FNSS 2019, Melbourne, VIC, Australia, November 27–29, 2019, pp. 172–189, Springer, Springer Cham, Cham (2019), ISBN 978-3-030-34353-8, https://doi.org/10.1007/978-3-030-34353-8_13

[14] Racing, X.C.D.: X Class budget build for under $400 (2024), URL https://www.youtube.com/watch?v=YyD9GRPMviw

[15] rctestflight: RC Solar Plane Flight Duration Test. https://www.youtube.com/watch?v=1OGrDvInUAY (2021), accessed: 2024-06-21

[16] rctestflight: Solar Plane V4 Cross-Country Waypoint Mission. https://www.youtube.com/watch?v=vYeYZpBE51I (2021), accessed: 2024-06-21

[17] Renduchintala, A., Jahan, F., Khanna, R., Javaid, A.Y.: A comprehensive micro unmanned aerial vehicle (UAV/Drone) forensic framework. Digital Investigation **30**, 52–72 (2019)

[18] Rodriguez-Alsina, A., Talavera, G., Orero, P., Carrabina, J.: Subtitle synchronization across multiple screens and devices. Sensors **12**(7), 8710–8731 (2012)

[19] Salamh, F.E., Mirza, M.M., Karabiyik, U.: UAV forensic analysis and software tools assessment: DJI Phantom 4 and Matrice 210 as case studies. Electronics **10**(6), 733 (2021)

[20] Stanković, M., Mirza, M.M., Karabiyik, U.: UAV forensics: DJI mini 2 case study. Drones **5**(2), 49 (2021)

[21] Statista: Drones – Worldwide (2024), URL https://www.statista.com/outlook/cmo/consumer-electronics/drones/worldwide#key-players

[22] sUASNews: A history of DJI wireless system, is Walksnail using DJI technology? https://www.suasnews.com/2022/06/a-history-of-dji-wireless-system-is-walksnail-using-dji-technology/ (2022), accessed: 2024-07-16

[23] TBS: TBS CROSSFIRE R/C System. TBS (aug 2022), URL https://www.team-blacksheep.com/media/files/tbs-crossfire-manual.pdf

[24] Team, A.D.: Ardupilot – Object Avoidance (2024), URL https://ardupilot.org/copter/docs/common-object-avoidance-landing-page.html

[25] Team, A.D.: Ardupilot – Radio Failsafe (2024), URL https://ardupilot.org/copter/docs/radio-failsafe.html

[26] Times, T., Times, T.S.: Inside Ukraine's deadly drone war (2024), URL https://www.youtube.com/watch?v=Cmv1frnURHA

[27] Yousef, M., Iqbal, F.: Drone forensics: A case study on a DJI Mavic Air. In: 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), pp. 1–3, IEEE, IEEE Press, Abu Dhabi, United Arab Emirates (2019), https://doi.org/10.1109/AICCSA47632.2019.9035365

[28] Zhao, Z., Wang, Y., Liao, G.: Digital Forensic Research for Analyzing Drone and Mobile Device: Focusing on DJI Mavic 2 Pro. Drones **8**(7), 281 (2024)