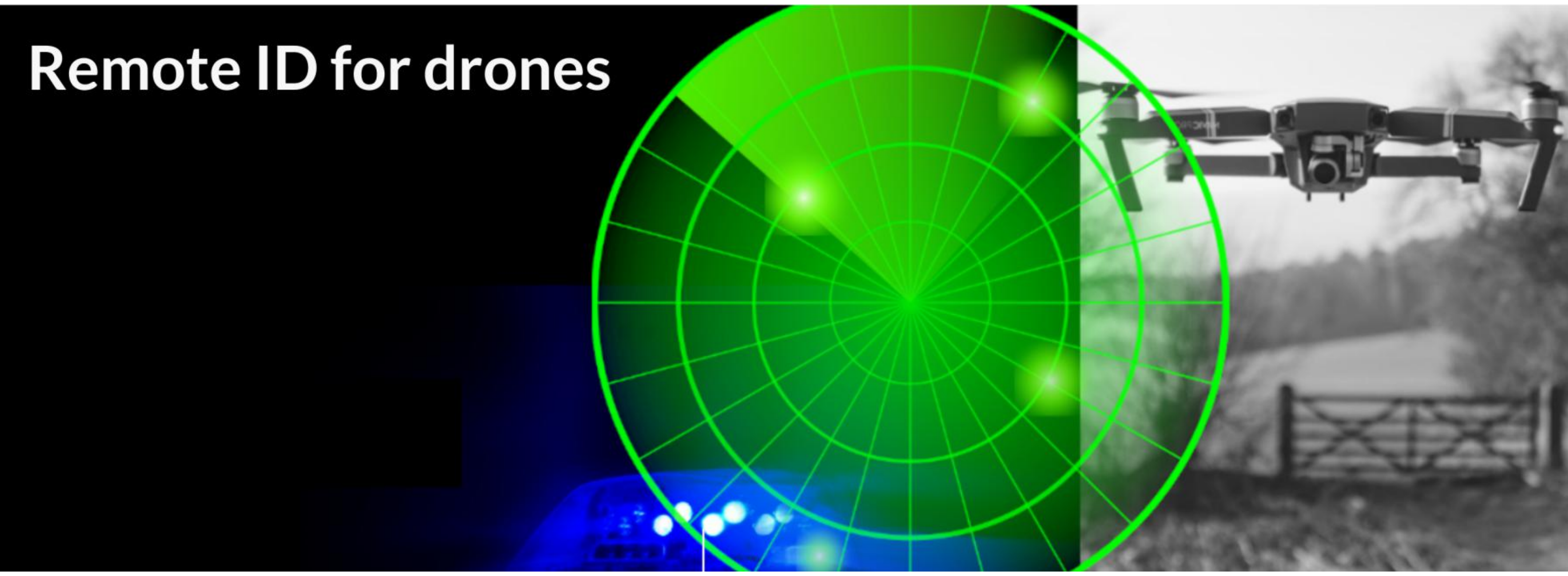
DroneScout - Receiver Manual
230 and 240-series

September 2023 - version 1.6



The latest version of this manual is located here:
https://download.bluemark.io/ds230.pdf

Intended audience: system integrators

Disclaimer: we are not responsible or liable for errors or incomplete information in this document.

Version history

version	date	description
1.0	May 2022	● Initial release
1.1	November 2022	● Added FCC information ● How to install a MQTT broker on the ds230
1.2	December 2022	● Added transmit_mode and aggregate_data options ● Improved update and reboot function
1.3	January 2023	● Describe how to solve DHCP issues in chroot mode ● Add typical RSSI values at 1 meter distance
1.4	March 2023	● Added transmit_mode 2 description
1.5	June 2023	● Added potential risks about transmit_mode 1 ● Describe how to configure a static IP address in section 1.6
1.6	September 2023	● Added the ds240 received to the manual

Contents

1 INTRODUCTION

Thank you for purchasing and using DroneScout products!

The latest version of this user manual may be downloaded at the following link, where the most up-to-date version will be found:
https://download.bluemark.io/ds230.pdf

(Direct/Broadcast) Remote Identification (Remote ID) adds “beacon” capability to drones to broadcast basic information of airborne drones, such as the operator's registration number, drone serial number and current position. The EU and USA are planning new rules that make Remote ID mandatory for drones over 250 grams weight. The beacon information can be used by general public, law enforcement and drones to give better situation awareness of the airspace around them.

BlueMark Innovations BV offers Remote ID transponders and receivers. DroneBeacon is an add-on (transponder) for drones which broadcasts Remote ID beacon signals. DroneScout is a receiver that detects Remote ID signals of nearby drones up to several km distance (in open space). See https://dronescout.co for more information about our products.

1.1 Audience

This document is intended for system integrators that want to use the *DroneScout 230 or DroneScout 240* receiver in their own product. This product is not intended for end users!

1.2 Receiver

The DroneScout receiver family consists of three models:

- *ds230* - Cost-effective basic Remote ID receiver
- *ds240* - Long Range Remote ID receiver, 3x more detection range compared to the ds230.
- *ds240 barebone* - Long Range Remote ID receiver (receiver only). No antennas, antenna cables and surge protectors are provided.

ds230

The DroneScout ds230 receiver consists of an embedded system and several radio-interfaces to collect remote ID signals.

Key specifications:

- Quad-Core Cortex-A53 ARM CPU 1.8 GHz
- 2 GByte RAM
- 8 GByte eMMC flash storage
- 10/100M/1000M Ethernet interface
- Compliant with international standards
    - EU ASD-STAN DIN EN 4709-002
    - USA ASTM Remote ID Standard ASTM F3411-22a-RID-B

- 1x Bluetooth (LE and BLE-Long Range) radio
    - Sensitivity:
        - BLE -97 dBm
        - BLE Long Range -105 dBm
- 2x triple-band WiFi radio: 2.4, 5.2 and 5.8 GHz
    - Sensitivity:
        - WiFi Beacon + WiFi NaN: -85 dBm
- PoE (Power over Ethernet): 802.3af/at
    - connectivity and power
- Power consumption: < 5 W
- Outdoor enclosure IP67
    - 1x Bluetooth antenna connector (N-type)
    - 2x WiFi antenna connector (N-type)
    - omni-directional antennas with 5 dBi gain
- Practical detection area: 80 km^2 / 5 km radius, see also section 1.9. The actual detection area can be bigger or smaller as it depends on the installation location, such as height and obstruction of large buildings.
- Size: 27.2 x 27.6 x 9.6 cm (without antennas).
- Operating temperature: -20°C to +40°C
- Weight: around 1.4 kg (with mast mount 1.9 kg

ds240

The DroneScout ds240 receiver consists of an embedded system and several radio-interfaces to collect remote ID signals.

Key specifications:

- Quad-Core Cortex-A53 ARM CPU 1.8 GHz
- 2 GByte RAM
- 8 GByte eMMC flash storage
- 10/100M/1000M Ethernet interface
- Compliant with international standards
    - EU ASD-STAN DIN EN 4709-002
    - USA ASTM Remote ID Standard ASTM F3411-22a-RID-B
- 1x Bluetooth (LE and BLE-Long Range) radio
    - Sensitivity:
        - BLE -97 dBm
        - BLE Long Range -105 dBm
- 2x 2.4 GHz WiFi radio
- 1x : 5 GHz band WiFi radio
    - Sensitivity:
        - WiFi Beacon + WiFi NaN: -85 dBm
- PoE (Power over Ethernet): 802.3af/at
    - connectivity and power
- Power consumption: < 5 W
- Outdoor enclosure IP67
    - 1x Bluetooth antenna connector (N-type)
    - 3x WiFi antenna connector (N-type)
- Antennas (the ds240 barebone receiver does not include these items):
    - 4x surge protectors 0 - 6 GHz
    - 3x 15 dBi omni-directional 2.4 GHz antenna: dimensions 1485*68 mm / 910 gram
    - 1x 15 dBi omni-directional 5 GHz antenna: dimensions 740*68 mm / 440 gram
    - 4x 50 cm N-connector antenna cables
- Practical detection area: 700 km^2 / 15 km radius, see also section 1.9
- Size: 27.2 x 27.6 x 9.6 cm (without antennas).
- Operating temperature: -20°C to +40°C

- Weight: around 1.4 kg (with mast mount 1.9 kg (excluding antennas) Antennas, RF cables, surge protectors are roughly 5 kg in total.

ds240 antenna patterns

2.4 GHz (Bluetooth & 2.4 GHz WiFi)

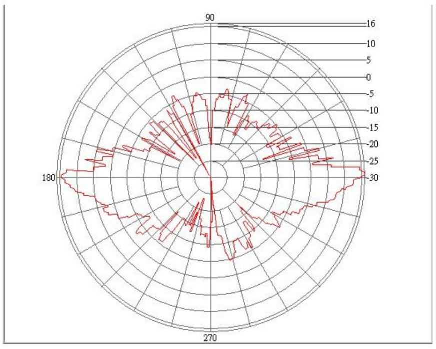
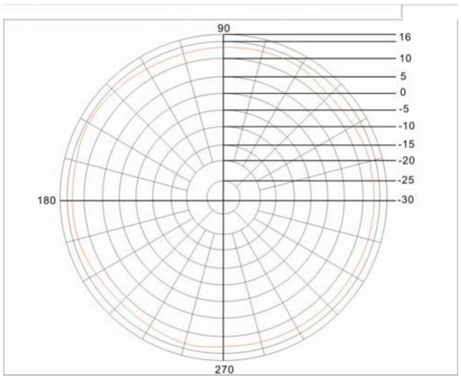
Frequency:2.4-2.5Ghz	
E-plane co-pol ------ 3-dB beam-with=6 Deg	H-plane co-pol ------ 3-dB beam-with=360 Deg
	
Vertical Pattern	**Horizontal Pattern**

Figure 1 - DroneScout 240 antenna pattern 2.4 GHz

5 GHz

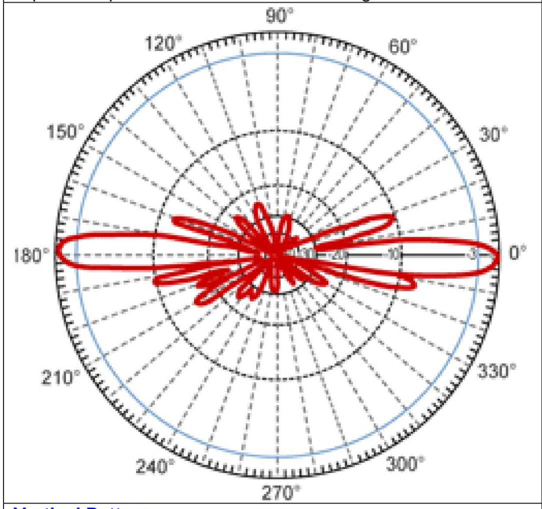
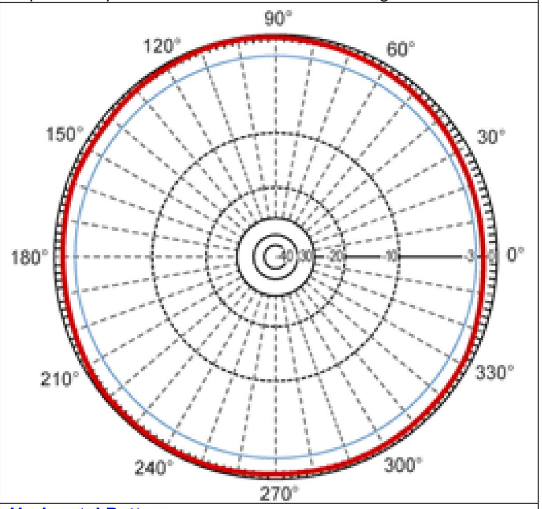
Frequency:5.1-5.9Ghz	
E-plane co-pol ------ 3-dB beam-with=7 Deg	H-plane co-pol ------ 3-dB beam-with=360 Deg
	
Vertical Pattern	**Horizontal Pattern**

Figure 2 - DroneScout 240 antenna pattern 5 GHz

Bluetooth scanning

The Bluetooth radio scans continuously for Bluetooth LE and Bluetooth LE Long-Range packets with Remote ID payload.

WiFi scanning

Remote ID signals can be broadcast on several frequencies. This means that the WiFi radio interface will hop every second to a new channel. If no Remote ID signals are detected, both radio interfaces will keep sensing for Remote ID signals on all WiFi channels. If there is a Remote ID signal found, radio 1 will keep hopping and scanning for new Remote ID signals. Radio 2 on the other hand will permanently tune to the channel where a Remote ID signal has been found.

In case of multiple Remote ID signals and multiple WiFi channels, radio 2 will hop every second to another channel where Remote ID signals have been found. If no Remote ID signals are found for 60 seconds, the channel will be removed from the list.

ds240 receiver
The ds240 receiver has a dedicated radio for 5 GHz signals. This means that both 2.4 GHz radio work as above. In case a Remote ID signal is found in the 5 GHz, the 5 GHz antenna will dedicate 50% of the time to tracking the 5 GHz signal and the other 50% scanning for new Remote ID signals. Compared to the ds230, the ds240 has faster detection of Remote ID signals and has a 3x larger detection range.

Remote ID can be broadcast in two WiFi formats: WiFi Beacon and WiFi NaN. WiFi NaN is also called Wi-Fi Aware and those signals can only be found on WiFi channel 6 (2.4 GHz), 44 (5 GHz) and 149 (5 GHz). For this reason, these channels are more frequently scanned for Remote ID signals. WiFi Beacon on the other hand is a format that is similar to the signals that a regular Access Point transmits. Those Remote ID signals can be found on all WiFi channels.

There is a configuration file where you can specify which WiFi channels will be scanned. See the next chapter for more details.

Antenna connectors

ds230

The ds230 uses 3 antennas. Below the N-connector there is a marking: ANT1/2/3/4

ANT 2 - WLAN 1
ANT 3 - WLAN 2
ANT 4 - Bluetooth

ds240
The ds240 uses 4 antennas. Below the N-connector there is a marking: ANT1/2/3/4

ANT 1 - WLAN 1 - 2.4 GHz
ANT 2 - WLAN 2 - 2.4 GHz
ANT 3 - WLAN 3 - 5 GHz
ANT 4 - Bluetooth

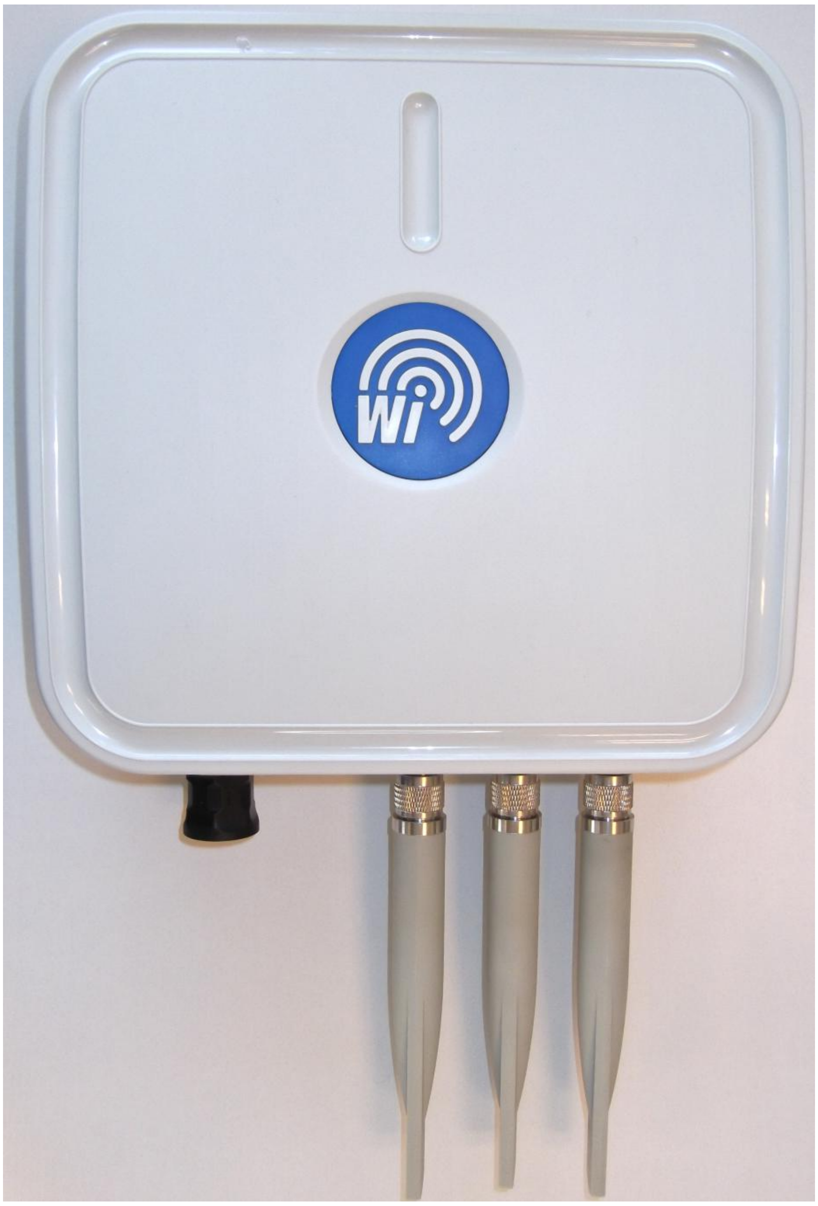


Figure 3 - DroneScout 230 receiver

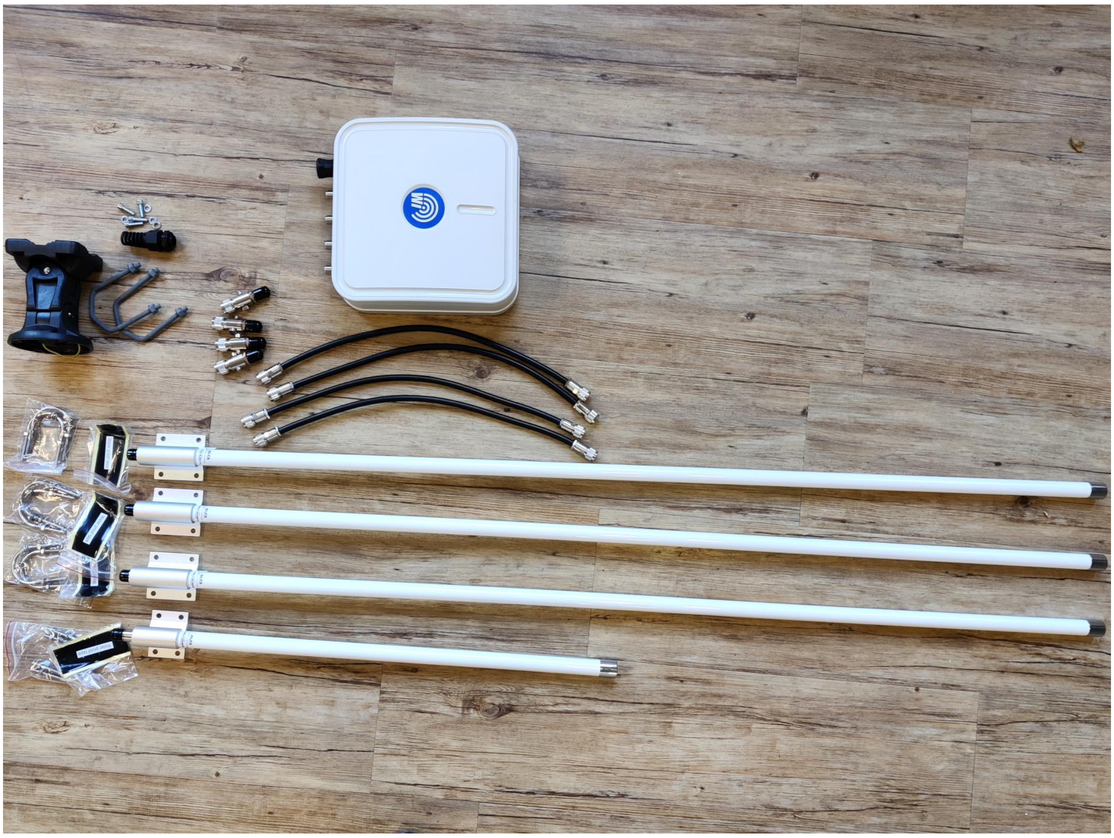


Figure 4 - DroneScout 240 receiver

1.3 Installation

Summary

- *Install the receiver to a wall or pole mast using the included mast mount.*
- *Connect/screw the 3 antennas to the antenna connectors on the enclosure.*
    - *Powering up the receiver **without antennas** may **damage** the Bluetooth and WiFi radios.*
    - ***Removing or attaching antennas** when the **receiver is powered up**, will **damage** the Bluetooth and WiFi radios.*
- *Connect the receiver to Ethernet and for outdoor installations make sure this connection is waterproof (by using the included waterproof accessories). This Ethernet cable needs to have power (PoE 802.3af/at) and also connectivity. The receiver acts as a DHCP client in the network.*

Location - The receiver detects remote ID signals from all directions, it is *omnidirectional*. For installation, it is therefore important not to install receivers near the border of the detection area, but instead in the center. It also depends a bit on the situation. If you have nearby WiFi networks, you don't want the receiver nearby it, as (busy) WiFi networks will reduce the detection range. Basically, install the receiver away from areas where there are signals in the 2.4/5 GHz band or large nearby objects (house) that can block detection of signals/drones from that direction.

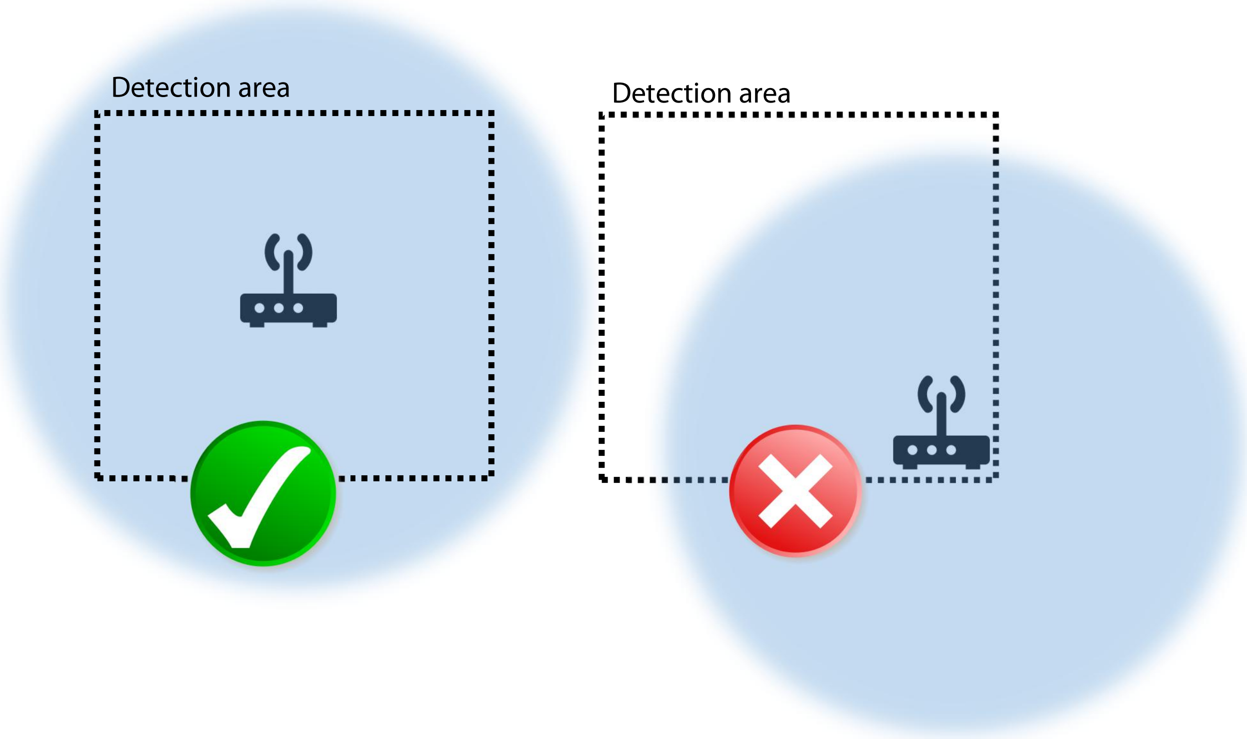


Figure 2 - install the receiver in the center of the detection area.

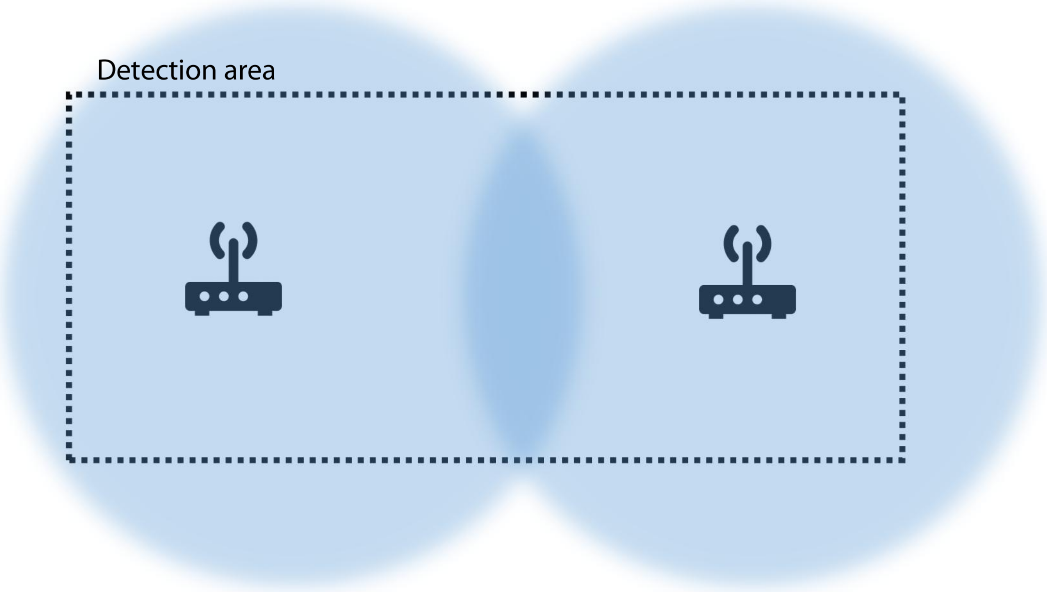


Figure 3 - use multiple receivers to cover the detection area in case the detection area is not square, or circle shape.

Construction materials - Construction materials (wood, concrete) attenuate wireless signals. This means that the detection area is reduced, if a receiver is installed behind or in such an object. This is especially true for the 5 GHz band. Also, it may introduce *blind spots* in the detection area where a drone is not detected. Installing multiple receivers is a solution to avoid blind spots.

For optimal performance install the receiver in open space, not surrounded by nearby objects. As a reference please find below a table describing the RF attenuation by various construction materials.

For instance, if a drone is detected in the 5 GHz band with a 114 mm wooden fence between receiver and drone, the signal strength is 13 dB less compared to no fence.

Material and thickness	2.4 GHz	5 GHz
Red brick (hollow), 89 mm	5	15
Window glass (uncoated), 6 mm	1	1
Plasterboard, 13 mm	1	0
Wood dry, 114 mm	7	13
Plywood dry, 13 mm	1	0
Bricks (concrete, hollow), 203 mm	11	15
Concrete (C8 mix), 203 mm	35	56
Reinforcing steel mesh (19 mm Ø, 70-mm-grid)	10	3
Reinforced concrete (C8, 19 mm Ø, 70-mm-grid)	37	58

Table 1: RF attenuation by various construction materials. (source c't 9/2021, page 139 using data from William C. Stone, Electromagnetic Signal Attenuation in Construction Materials, 1997)

Height - Preferred installation height is 2 to 40 meters. Installation lower, near the ground, will reduce the detection area as objects in the detection area will block wireless signals more. Installing the receiver higher on the other hand will increase the detection area, but may prevent detecting remote ID signals very nearby.

Angle - The receiver has omnidirectional antennas. Install the receiver with zero angle (vertical plane). This means that the receiver should looks straight ahead. Not down or up under an angle.

Power - The receiver needs power and is powered via Power over Ethernet (PoE), 802.11af. Connect the Ethernet port of the receiver to an PoE capable switch/router to have both power and connectivity.

Connectivity - Connect the Ethernet port of the receiver to your router. The receiver needs Ethernet to upload data to the MQTT broker. It can also be used for management purposes. The network name is the serial number that is printed on back of the receiver (dsxxxxxxxxxxxx) e.g. ds220300000101.

Mast mount

For each receiver a mast mount is provided. It can be used to install the receiver to a mast or directly to a wall. See details below.

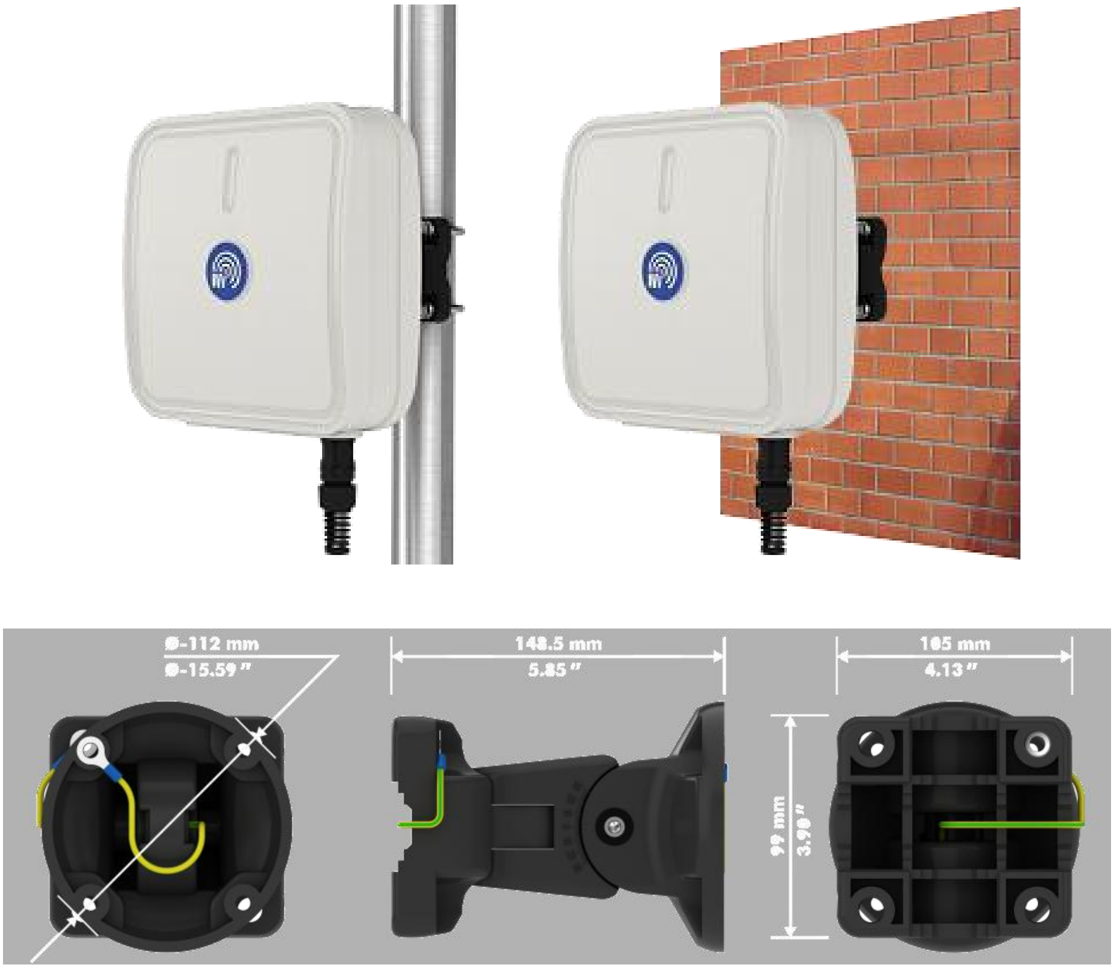
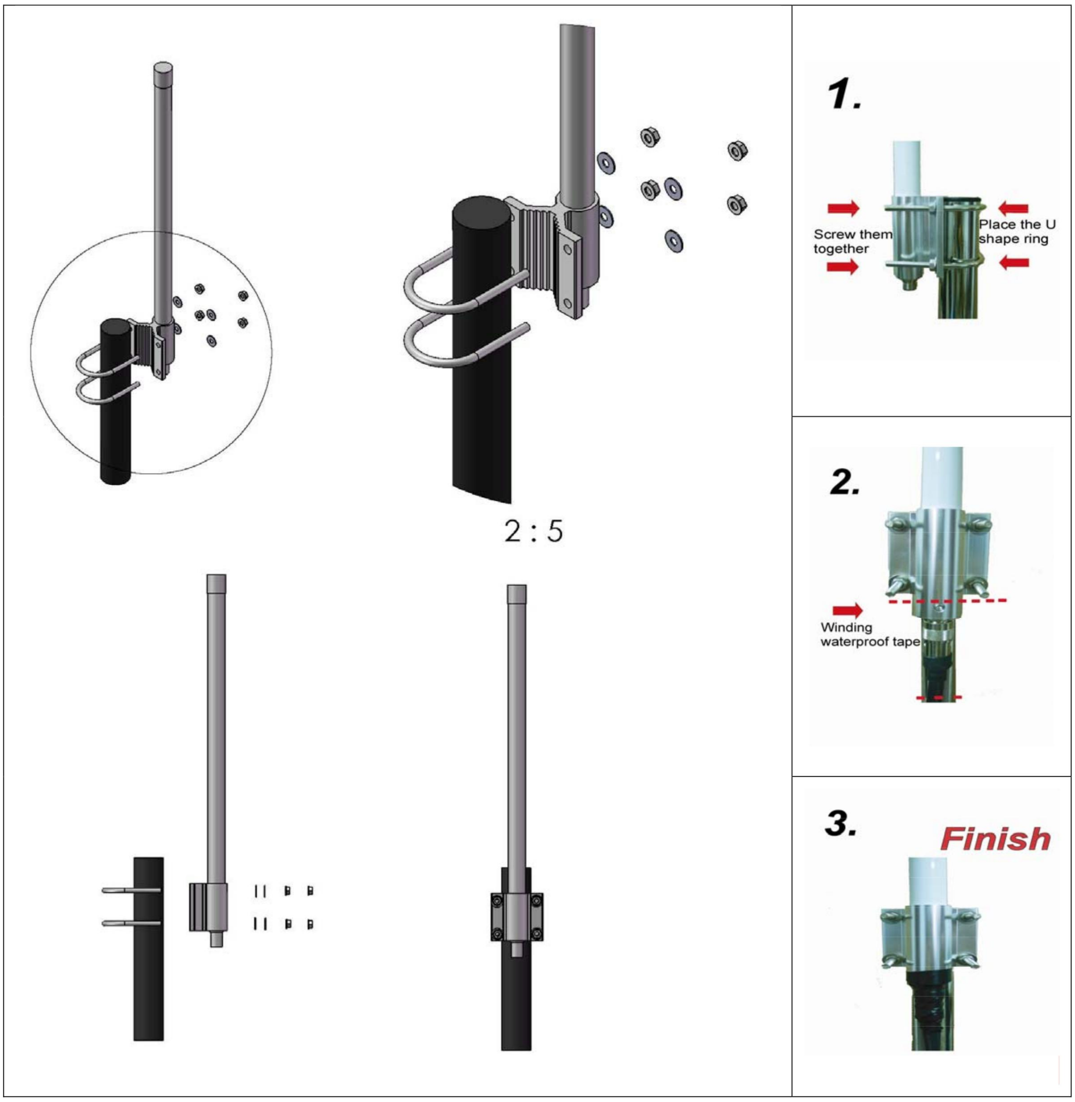
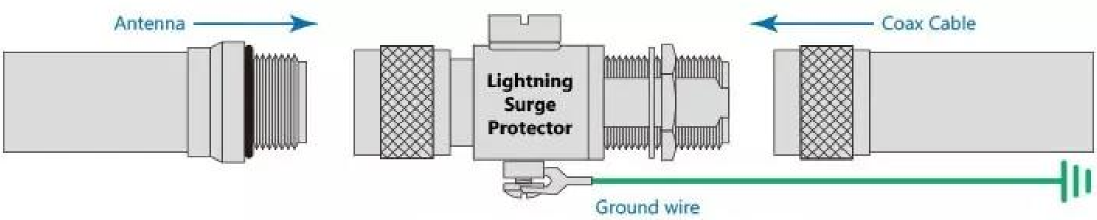


Figure 4 - mast mount to install the receiver to a mast or directly to a wall.

DS240 antenna mounting



DS240 lightning surge protectors



Unit provides protection to both connector equaly.

1.4 Login

The receiver runs on Armbian Linux distribution (ARM64); https://www.armbian.com/. It can be accessed via SSH on the local network.

Login details

User name: root
Password: bluemark
Service: SSH
Port: 22
IP address: DHCP client in local network

Note: please change the password in production deployments!

1.5 Read-only file system

The receiver uses a so-called *overlayroot* file system. The file system is mounted read-only and there is a read-write file system in memory on top of it. This means that the system can be used normally, but after a reboot all changes to the file system are lost. Using a read-only file system prevents for instance corrupt file systems after an unexpected power loss.

To make changes permanent:

- Enter in the SSH console: `overlayroot-chroot`
  - After this command you can make changes to the filesystem. Enter `exit` to exit this mode.
  - If you need full access with internet, using apt etc, **execute prior to overlayroot-chroot, the following commands:**

```
mount -t sysfs none /media/root-ro/sys
mount -t proc none /media/root-ro/proc
mount --bind /dev/ /media/root-ro/dev
mount --bind /tmp/ /media/root-ro/tmp
mount --bind /dev/pts /media/root-ro/dev/pts
mount -o bind /etc/resolv.conf /media/root-ro/etc/resolv.conf
```

- Or mount the read-only partition as read-write instead. I.e enter `mount -o remount,rw /media/root-ro`
  - Make the changes in the folder /media/root-ro
  - Remount as read-only: `mount -o remount,rw /media/root-ro`

Reboot afterwards.

1.6 Change IP address of the sensor

To change the IP address of the DroneScout receiver from DHCP client to a static IP address, use the following commands (modify where needed). **Do not execute the commands in section 1.5.**

```
nmcli dev status #available network devices
DEVICE TYPE STATE CONNECTION
enx00e04c680aaa ethernet connected Wired connection 1
eth0 ethernet unavailable --
lo loopback unmanaged –

#set to 192.168.100.144 with gateway 192.168.100.1 and dns server 1.1.1.1
#modify to your settings
nmcli con mod "Wired connection 1" ipv4.addresses 192.168.100.144/24
nmcli con mod "Wired connection 1" ipv4.gateway 192.168.100.1
nmcli con mod "Wired connection 1" ipv4.dns 1.1.1.1
nmcli con mod "Wired connection 1" ipv4.method manual
nmcli con up "Wired connection 1" #apply new settings and save

#copy the config to the permanent partition
mount -o remount,rw /media/root-ro
cp NetworkManager/system-connections/Wired\ connection\ 1.nmconnection
/media/root-ro/etc/NetworkManager/system-connections/
sync
mount -o remount,ro /media/root-ro
reboot -f #
```

1.7 Open Drone ID

DroneScout uses the Open Drone ID framework to encode Remote ID signals. The framework can be found on this page:

https://www.opendroneid.org/

1.8 Global architecture

The receiver has a binary, dronescout, in the root-folder (/root) to sense for Remote ID signals. There is also a configuration file *dronescout.conf* to configure MQTT and other settings. The file *wlan_channels.conf* is used to specify the WiFi channels that will be scanned. Finally, there are some auxiliary files that are discussed in the next chapter.

1.9 Maximum detection range

The maximum detection range of the ds230/ds240 receiver depends on several factors:

- **Effective radiated power** (ERP) of the transponder (transmit power, antenna design)
- **Antenna height** of the receiver and transponder. Detection range increases with higher height as it converges to free-space propagation.
- **Antenna gain and directivity** of the receiver. The maximum ERP power of the transponder is limited by the WiFi/Bluetooth technology standard.
- **Line of sight versus non-line of sight** to the transponder due to buildings, trees, hills etc.
- **Weather**: rain, fog or other “wet” weather will limit the range.

- **Moving drone**: if the drones moves, it will impact negatively on the range. The average RSSI will vary more (signals may be lost more easy). Also if the drones move fast, typically the drone will tilt. In such a case, the drone may block partly the signals towards the receiver.
- **Probability threshold** of detection[1].

No guarantees can be made about the detection range, due to the complex nature of wireless propagation, as shortly described above. Typically, the detection range is at least multiple kilometers.

The ds240 receiver has 3x larger detection range compared to the ds230 receiver due to the use of high gain antennas that result in a 10 dB extra increase in the sensitivity.

The detection range below is calculated based on internal measurements, that have been extrapolated using the "best-case" Free space propagation model[2].

In general, free-space is considered the most ideal situation. In real-life situation there is more attenuation between the receiver and beacon. Typically, how higher the receiver is installed, how more the propagation mimics free-space. Hence, the detection range increases. This is supported for instance by publications like: *"Characterization of Radio Path Loss in Seaport Environment for WiMAX Applications" - Ming-Tuo Zhou , Joe Jurianto , Jaya Shankar , M. Fujise*[3]

The free-space path loss model is:

PL= $20log_{10}\left(\frac{d}{d_0}\right) + c$

Where d is the distance, d_0, a reference distance and c a constant value that among other depends on the frequency. If the distance d doubles i.e. $d = 2d$, the path loss will increase by 6 dB and if d would be 10 times larger, the path loss increases by 20 dB.

Measurement setup (ds230 receiver):

- receiver antenna height 2.5 m
- drone/transponder height 10 m
- transponder ERP power 20 dBm.
- RSSI measurements based on a *slow* moving drone equipped with the DroneBeacon transponder.
- flat agricultural land
  - nearby trees > 15 meter (behind antenna), nearby building > 25 meter (behind antenna)
  - nearby building can act as reflector, so results could be (slightly) too optimistic.
- sunny weather

BLE legacy

- Average RSSI at 500 m: -78 dBm
- Sensitivity radio -97 dBm

[1] The received signal varies both in time and location due to the reception of multiple radio signals (like direct path and ground wave). In time, typically the average signal strength is used. With respect to locations, paths can add up constructively or destructively, which depend on the path difference(s). As a result it means that at distance N there is a variation in received signal strength depending on the location. If you are on the edge of the detection range, you won't always detect the transponder due this variation. In this document we use 50% of the area locations at distance N.

[2] https://en.wikipedia.org/wiki/Free-space_path_loss

[3] http://ap-s.ei.tuat.ac.jp/isapx/2006/pdf/3B1b-4.pdf (Table 1).

So 19 dB (-78 - -97) above sensitivity level. Assuming free space propagation (6 dB loss per doubling of the distance), the maximum detection range is: 500*10^(19/20) = **4.4 km (ds230 receiver)**. This results in an detection area of **61 km^2.**

The ds240 receiver has 10 dB more sensitivity, the maximum detection range is **13.2 km (ds240 receiver)**. This results in an detection area of **547 km^2**.

BLE Long Range

- Average RSSI at 500 m: -78 dBm
- Sensitivity radio -105 dBm

So 27 dB (-78 - -105) above sensitivity level. Assuming free space propagation (6 dB loss per doubling of the distance), the maximum detection range is: 500*10^(27/20) = **11.2 km (ds230 receiver)**. This results in an detection area of **394 km^2.**

The ds240 receiver has 10 dB more sensitivity, the maximum detection range is **33.6 km (ds240 receiver)**. This results in an detection area of **3545 km^2.**

WiFi NaN 2.4 GHz

- Average RSSI at 500 m: -61 dBm
- Sensitivity radio -85 dBm

So 19 dB (-61 - -85) above sensitivity level. Assuming free space propagation (6 dB loss per doubling of the distance), the maximum detection range is: 500*10^(19/20) = **7.9 km (ds230 receiver)**. This results in an detection area of **196 km^2.**

The ds240 receiver has 10 dB more sensitivity, the maximum detection range is **23.7 km (ds240 receiver)**. This results in an detection area of **1764 km^2.**

WiFi Beacon 2.4 GHz

- Average RSSI at 500 m: -60 dBm
- Sensitivity radio -85 dBm

So 25 dB (-60 - -85) above sensitivity level. Assuming free space propagation (6 dB loss per doubling of the distance.) So maximum detection range is: 500*10^(25/20) = **8.9 km (ds230 receiver)**. This results in an detection area of **249 km^2.**

The ds240 receiver has 10 dB more sensitivity, the maximum detection range is **26.7 km (ds240 receiver)**. This results in an detection area of **2238 km^2.**

WiFi Beacon 5.2 GHz

- Average RSSI at 500 m: -68 dBm
- Sensitivity radio -85 dBm

Typically, the 5.2 and 5.8 GHz frequency band have lower detection range due to the higher frequency.

So 17 dB (-68 - -85) above sensitivity level. Assuming free space propagation (6 dB loss per doubling of the distance), the maximum detection range is: 500*10^(17/20) = **3.5 km (ds230 receiver)**. This results in an detection area of **39 km^2.**

The ds240 receiver has 10 dB more sensitivity, the maximum detection range is **10.5 km (ds240 receiver)**. This results in an detection area of **346 km^2.**

Extending the detection range

The detection range can be extended in the following ways:

- Install the receiver at a higher place. In this case the received signal will be stronger as propagation is more similar to free-space propagation.
- Replace the antennas with a higher antenna gain. Rule of thumb is that every 6 dB increase, will result in doubling of the detection range. Hence, a 15 dBi antenna would triple (3x) the maximum detection distance. High-gain antennas and/or high receiver location may prevent detection of nearby drones. Experiments with the 5 dBi antennas and flying 50 m above the receiver, did not give any problem detecting the transponder. (The signal was strong received.) The same holds for a 50 m distance experiment, where the drone was moved from 3m to 50 meter height. In all cases the drones was received with a strong signal.

1.10 RSSI values at 1 meter distance

The following RSSI values were measured at 1 meter distance using a db120 transponder in an office environment and a ds230 receiver. If you measure much weaker RSSI signals as the ones stated below with the db120 transponder (larger as 10 dB), typically the ds230 receiver is broken or the antennas are not mounted properly.

Using a transponder of a different brand can result in different (lower RSSI values), for instance if the transmit power of that transponder is lower or a different antenna is used. In general the RSSI values also depend on the environment and position of the transponder. This can result in RSSI values that are 3 dB higher or lower.

WLAN transmission protocols: on average -20 dBm +/- 3 dB.
Bluetooth transmission protocols: ~ -37 dBm +/- 3 dB.

2 CONFIGURATION

The firmware is protected by a license/device key. In case the receiver does not work anymore due to license errors, please contact support.

The ds230/ds240 needs a MQTT broker for uploading data. In the default configuration none is configured. For test purposes you can also install a MQTT broker on the sensor it self. This is described in section 2.5.

2.1 root password

For production environments it is **strongly** advised to change the default password.

```
overlayroot-chroot
passwd #interactive tool to change the password
exit
reboot # new password will work after a reboot
```

2.2 dronescout.conf

The file /root/dronescout.conf is the main configuration file.

The default contents are shows below.

```
#
# Configuration file
# (c) Bluemark Innovations BV 2022 - 2023

[global]
sensorID = ds220500000100 ; receiver ID up to 256 characters

[mqtt]
host = myserver ; MQTT host
port = 8883 ; MQTT port
topic = ; leave empty to use default topic based on receiverID
QoSlevel = 1 ; QoS level 0, 1 or 2
username = ; leave empty if not used
password = ; leave empty if not used
keepalive = 60 ; keep alive period in seconds.
clientID =  ; set clientID, leave empty for default setting
ssl = 1 ; 0 disable SSL in MQTT connection 1, enable SSL
ssl_verify = 0 ; disable/enable SSL verification
CAfile = /root/certs/ca.crt ; location to CA file for SSL connection
CRTfile = /root/certs/client.crt ; location to CRT file for SSL connection
KEYfile = /root/certs/client.key ; locaton to KEY file for SSL connection
compression = lzma ; none or lzma. In case of lzma, payload is compressed.
retain = 0 ; set to 1 in order to retain messages on mqtt broker
```

```
transmit_mode = 2; 0 send MQTT message for each new message, 1 every second
#                  (truncate), 2 combine all and send at transmit mode 2
#                  interval ms period
transmit_mode_2_interval_ms = 250; set the transmission interval (ms) in
#                  transmit mode 2 (valid range: 50 to 60000)
aggregate_data = 1; 0 send MQTT message with latest received message only,
1 send all information about this RemoteID device

[interface]
WLAN_USB_1 = wlan1 ; interface for WLAN 1 adapter
WLAN_USB_2 = wlan2 ; interface for WLAN 2 adapter
WLAN_USB_3 = wlan3 ; ds240 only interface for WLAN 3 adapter
BT_UART_1 = /dev/ttyUSB0 ; interface for USB UART adapter

[threshold]
WLAN_USB = -200 ; signals weaker as the threshold won't be processed.
BT_UART = -200 ; signals weakers as the threshold won't be processed.
```

Use the following commands to edit file /root/dronescout.conf:

```
overlayroot-chroot
nano /root/dronescout.conf  # use nano (or vi) as editor
exit
reboot # to apply changes
```

sensorID

The sensorID is a string up to 256 characters. It is used to identify the DroneScout receiver and is also used in the MQTT payload.

MQTT

The receiver uses internally the MQTT mosquitto library (https://mosquitto.org/). Settings in this MQTT section relate to this library.

For non-encrypted MQTT brokers, only set the *host* and *port*. Make sure that *ssl* is set to 0. If no *topic* is specified, the receiver will use the topic: */sensor/<sensorID>/upload*

If *compression* is set to *none*, the receiver will publish JSON payload in plain text. If compression is set to *lzma* the entire JSON payload will be compressed with LZMA. Typically, LZMA achieves over 80% compression ratio. In plain text mode, the payload is typically around 1400 bytes, with LZMA compression, it is around 260 bytes.

For production deployments, it is **strongly** advised to enable SSL encrypted communication! In this case set *ssl* to 1. Also, set the related file locations: *CAfile*, *CRTfile* and *KEYfile* to the files needed for SSL-encrypted communication to the MQTT broker. If case of self-generated SSL keys, set *ssl_verify* to 0.

For more security also a *username*, *password* can be configured, if the MQTT broker requires this.

In firmware *20221208-1250* and higher, two new options are available: transmit_mode and aggregate_data. The default value for both variables is 1. If transmit_mode is 1, roughly 2 times per second (2 Hz) the sensor will generate a MQTT message for each detected Remote ID device (if new signals are detected for that device). If transmit_mode is 0, a MQTT message is generated every time a Remote ID signal is detected. (No throttling.) In case transmit_mode is 0, the MQTT broker may be overloaded, if a lot of signals are detected and a lot of sensors are uploading data to the broker. In such cases new MQTT messages will get stalled at the sensor. In most smaller setups, it is safe to use transmit_mode is 0.

In firmware *20230329-1042* and higher transmit mode 2 has been introduced. In this mode all received MQTT messages are combined into one payload. At the interval *transmit_mode_2_interval_ms* the combined payload (in milliseconds) is transmitted via MQTT. The valid range is between 250 ms and 60000 (1 minute). Use this transmit mode if you want to publish all received Remote ID signals, but at the same time want to limit the message rate to prevent overloading of the MQTT broker/publishing capacity. Note for transmit mode 2 you also need the latest MQTT subscriber application (Chapter 4).

Potential risks of transmit mode 1 Although transmit mode 1 prevents overloading of a MQTT broker, there is a small probability that an attacker could use this throttling, to broadcast a similar malicious Remote ID signal where some values have been changed (like location data). If the attacker uses the correct timing, it can prevent the ds230 to receive the original Remote ID signal. For that reason newer ds230 receiver will use transmit mode 2 by default. Also for existing deployments we recommend to switch from transmit mode 1 to 2.

The other option is aggregate_data. In transmission mode BT4 legacy, only one part of the Remote ID signal is broadcast per time. E.g. one message for the Basic ID, another one for the Location data etc. For other transmission modes, the messages will typically contain all Remote ID information. If aggregate_data = 1, the sensor will save in memory the latest contents of the Remote ID device. So if a BT4 Basic ID message is received, that information element is updated. Other elements are unchanged. The sensor will output an MQTT signal with all available information elements. In case of aggregate_data = 0, all information elements are set to zero and only the latest received information will be saved to memory and the sensor will output an MQTT signal accordingly.

interfaces

This section configures the location of the Bluetooth and WiFi radios. Leave to default settings.

threshold

Advanced setting that you typically don't need to change. All radios sense with maximum sensitivity. In case you want to reduce the detection range, you can specify here a threshold. Signals lower as the threshold won't be processed.

2.3 remote SSH login

When receivers are placed behind a router, they can't be accessed remotely. Reverse SSH is a method to be able to login remotely without changing router or firewall settings. See https://www.howtogeek.com/428413/what-is-reverse-ssh-tunneling-and-how-to-use-it/ for background information.

Default remote SSH login is disabled.

If remote SSH is enabled (by setting *enabled to* "1" in */root/remote.conf*), the receiver will try to connect to the configured SSH server (using *server*, *port* and *user* in *remote.conf*). It will open a port (*remote_port* in *remote.conf*) that can be used to remotely login. Each receiver needs an unique port, otherwise those receivers will compete for the same port..

In addition, passwordless login by using SSH keys is assumed. This means that the receiver SSH key (/root/.ssh/id_rsa) needs to be accepted by the remote SSH server. More information can be found in the mentioned background article.

Use the following commands to edit file /root/remote.conf:

```
overlayroot-chroot
nano /root/remote.conf  # use nano (or vi) as editor
exit
reboot # to apply changes
```

Login from the SSH server

The receiver can be accessed from the SSH server by entering the following command: `ssh root@localhost -p10230`

The number 10230 is the *remote_port* in *remote.conf.*

2.4 wlan_channels.conf

The /root/wlan_channels.conf file specifies which WiFi channels are scanned by the ds230/ds240 receiver. The ds240 will automatically filter this list and direct the 5 GHz channels to the 5 GHz radio interface and the 2.4 GHz to both 2.4 GHz radios. The default contents is show below:

```
0,1
1,2
2,3
3,4
4,5
5,6
6,44
7,149
8,7
9,8
10,9
11,10
12,11
13,13
14,6
15,149
16,44
17,36
18,40
19,48
20,52
21,56
22,60
23,149
24,6
25,44
26,13
27,11
28,10
29,9
30,8
31,7
32,149
```

```
33,44
34,6
35,5
36,4
37,30
38,2
39,1
40,64
41,161
42,165
43,153
44,157
```

Use the following commands to edit file /root/remote.conf:

```
overlayroot-chroot
nano /root/wlan_channels.conf  # use nano (or vi) as editor
exit
reboot # to apply changes
```

The first number on the row is the sequence number. The second number is the channel number. Up to 256 channels can be configured. If the WiFi radio reaches the end of the sequence it will start with the first element. The second radio will scan the same channels, but with half period delay. Typically leave this file to default settings.

WiFi NaN (also called Wi-Fi Aware) signal can only be found on WiFi channel 6 (2.4 GHz), 44 (5 GHz) and 149 (5 GHz). WiFi Beacon signals can be found on all WiFi channels.

The following channels can be configured (capabilities of WiFi radio. The number in brackets [ ] is the channel number:

- 2412 MHz [1]
- 2417 MHz [2]
- 2422 MHz [3]
- 2427 MHz [4]
- 2432 MHz [5]
- 2437 MHz [6]
- 2442 MHz [7]
- 2447 MHz [8]
- 2452 MHz [9]
- 2457 MHz [10]
- 2462 MHz [11]
- 2467 MHz [12]
- 2472 MHz [13]
- 2484 MHz [14]
- 5075 MHz [15]
- 5080 MHz [16]
- 5085 MHz [17]
- 5090 MHz [18]
- 5100 MHz [20]
- 5120 MHz [24]
- 5140 MHz [28]
- 5160 MHz [32]
- 5180 MHz [36]
- 5200 MHz [40]
- 5220 MHz [44]
- 5240 MHz [48]
- 5260 MHz [52]
- 5280 MHz [56]
- 5300 MHz [60]
- 5320 MHz [64]
- 5340 MHz [68]
- 5360 MHz [72]
- 5380 MHz [76]
- 5400 MHz [80]
- 5420 MHz [84]
- 5440 MHz [88]
- 5460 MHz [92]
- 5480 MHz [96]
- 5500 MHz [100]
- 5520 MHz [104]
- 5540 MHz [108]
- 5560 MHz [112]
- 5580 MHz [116]
- 5600 MHz [120]
- 5620 MHz [124]
- 5640 MHz [128]
- 5660 MHz [132]

- 5680 MHz [136]
- 5700 MHz [140]
- 5720 MHz [144]
- 5745 MHz [149]
- 5765 MHz [153]
- 5785 MHz [157]
- 5805 MHz [161]
- 5825 MHz [165]
- 5845 MHz [169]
- 5865 MHz [173]
- 5885 MHz [177]

2.5 MQTT broker on the DroneScout receiver

For test purposes you can also install a MQTT broker (mosquitto) on the ds230/ds240 sensor. (In the default ds230/ds240configuration none is configured.) Installing a MQTT broker on the sensor may result in network vulnerabilities. For that reason we don't advise this for production environments.

There are two options for installing the MQTT broker: a) permanent b) temporarily (lost after a reboot). For option a) see Section 1.5 to make permanent changes to the file system.

- First install the mosquitto MQTT broker:

  ```
  apt update; apt install -y mosquitto
  ```

- Add these lines to the mosquitto configuration (`/etc/mosquitto/mosquitto.conf`)

  ```
  echo "listener 1883" >> /etc/mosquitto/mosquitto.conf
  echo "allow_anonymous true" >> /etc/mosquitto/mosquitto.conf
  ```

- Restart mosquitto MQTT broker

  ```
  service mosquitto restart
  ```

- The default configuration of the dronescout receiver (file /root/dronescout.conf) publishes messages to the local MQTT broker. Make sure that host, port and ssl are set correctly as shown below.

```
#
# Configuration file
# (c) Bluemark Innovations BV 2022 - 2023

[global]
sensorID = ds220500000100 ; sensor ID

[mqtt]
host = localhost ; MQTT host
port = 1883 ; MQTT port
topic = ; leave empty to use default topic based on sensorID
QoSlevel = 1 ; QoS level 0, 1 or 2
username = ; leave empty if not used
password = ; leave empty if not used
keepalive = 60 ; keep alive period in seconds.
```

```
clientID =  ; set clientID or to random to generate random ID, leave empty
for default setting
ssl = 0 ; 0 disable SSL in MQTT connection 1, enable SSL
ssl_verify = 0 ; disable SSL verification of SSL key of MQTT broker (useful
for self-generated keys)
CAfile = /root/certs/ca.crt ; location to CA file for SSL connection
CRTfile = /root/certs/client.crt ; location to CRT file for SSL connection
KEYfile = /root/certs/client.key ; locaton to KEY file for SSL connection
compression = lzma ; none or lzma. In case of lzma, payload is compressed.
retain = 0 ; set to 1 in order to retain messages on mqtt broker
transmit_mode = 2; 0 send MQTT message for each new message, 1 every second
transmit_mode_2_interval_ms = 250; set the transmission interval (ms) in
transmit mode 2 (valid range: 50 to 60000)
aggregate_data = 1; 0 send MQTT message with latest received message only,
1 send all information about this RemoteID device

[interface]
WLAN_USB_1 = wlan1 ; interface for WLAN 1 adapter
WLAN_USB_2 = wlan2 ; interface for WLAN 2 adapter
BT_UART_1 = /dev/ttyS3 ; interface for Bluetooth adapter

[threshold]
WLAN_USB = -200 ; signals weakers as the threshold won't be processed,
-200 is all packets are processed.
BT_UART = -200 ; signals weakers as the threshold won't be processed, -200
is all packets are processed.
```

- Restart the dronescout application
  `pkill dronescout`

- (Optional) point the mqtt subscriber application to the ds230/ds240 sensor:
  https://github.com/BluemarkInnovations/RemoteID-MQTT-subscriber/blob/main/mqtt_sub.py
  - Change `broker` to the IP address of the sensor
  - Change `port` to 1883
  - Comment (#) the line starting with client_pem:
    `#client_pem = "./certs/client.pem"`

The python script now looks like (line 19 to 31):

```
...
broker = 'IP_address_of_sensor'
port = 1883
topic = "#"

# generate client ID with pub prefix randomly
client_id = f'mqtt-subscriber-{random.randint(0, 100)}'

#optional user/password for connecting to the MQTT broker, uncomment if
used.
#username = 'myusername'
#password = 'mypassword'

#file containing full SSL chain, uncomment when MQTT broker uses encrypted
messages [preferred]
#client_pem = "./certs/client.pem"
...
```

You should now be able to receive ds230/ds240 MQTT messages.

For option a) permanent installation, enter the command `exit` and reboot the sensor to apply the changes: `reboot -f`.

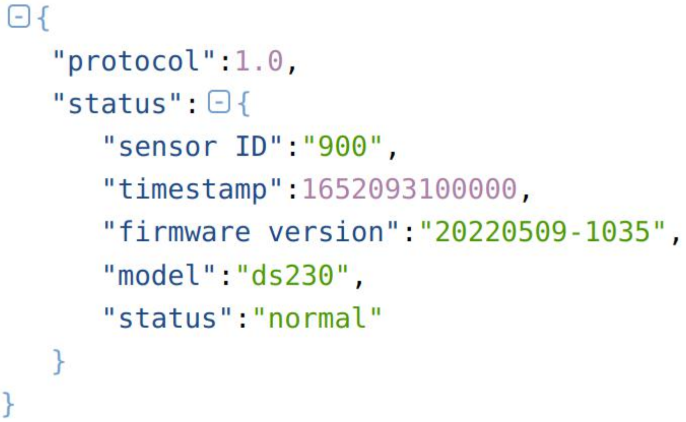
3 MQTT MESSAGES

The receiver generates two types of MQTT messages (JSON format):

- *Status messages* - every minute, the receiver will send a status message
- *Data messages* - data messages with Remote ID data.

Status messages

The receiver will publish every minute a status message. An example message is shown below.

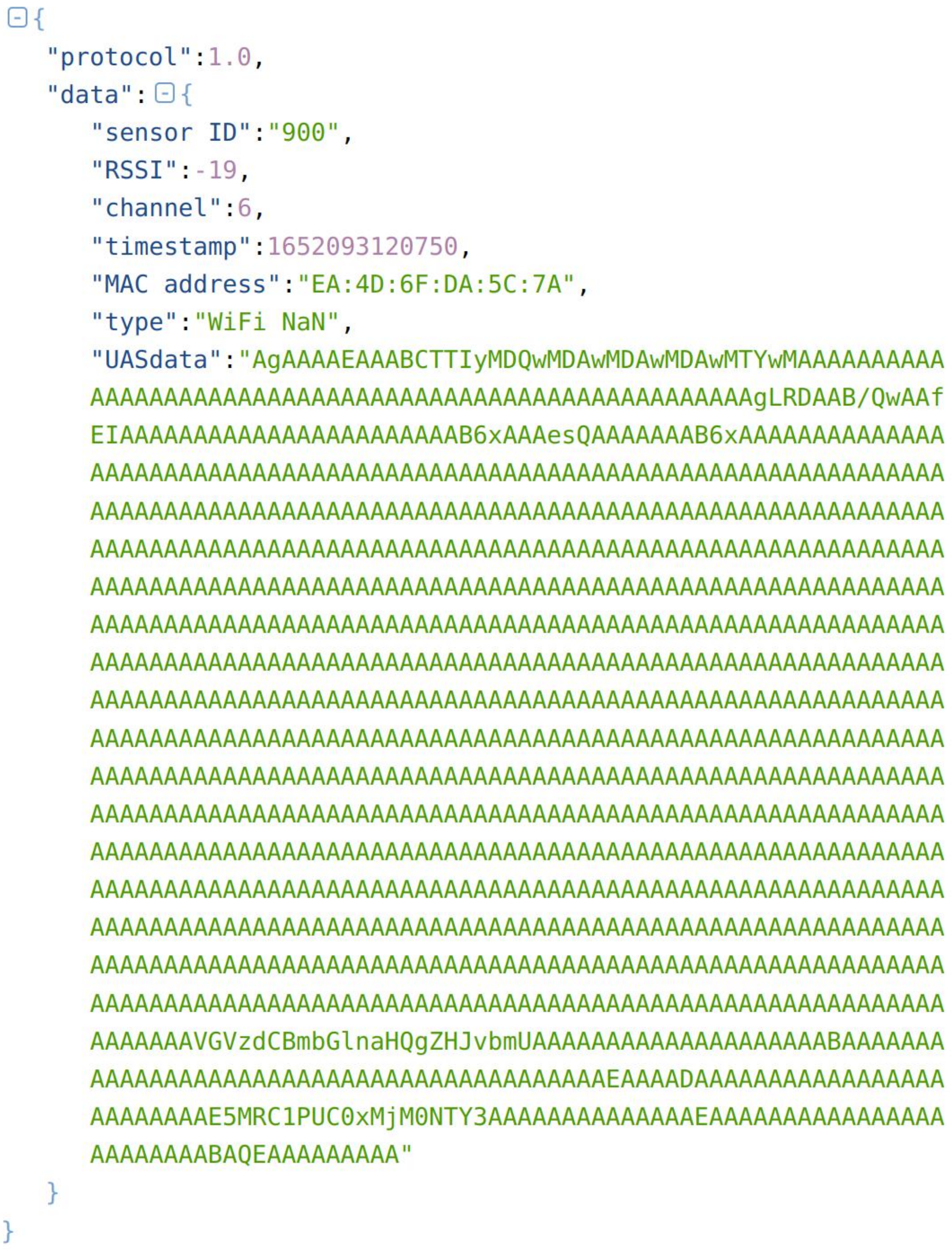


```
{
   "protocol":1.0,
   "status":{
      "sensor ID":"900",
      "timestamp":1652093100000,
      "firmware version":"20220509-1035",
      "model":"ds230",
      "status":"normal"
   }
}
```

Figure 4 - Example receiver status message.

This message contains several sections:

- *protocol* - indicates the protocol version. Currently only protocol 1.0 exists.
- *status* - a status message contains a section status.
- *sensor ID* - the sensor ID set in dronescout.conf
- *timestamp* - the epoch time stamp (in milliseconds)
- *firmware version* - the current firmware version of the receiver
- *model* - the model of the receiver: can be ds230 or ds240.
- *status* - the status, can be “normal” or “invalid license”. If the license is invalid, no data will be published.



```
{
   "protocol":1.0,
   "data":{
      "sensor ID":"900",
      "RSSI":-19,
      "channel":6,
      "timestamp":1652093120750,
      "MAC address":"EA:4D:6F:DA:5C:7A",
      "type":"WiFi NaN",
      "UASdata":"AgAAAAEAAABCTTIyMDQwMDAwMDAwMDAwMTYwMAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgLRDAAB/QwAAf
      EIAAAAAAAAAAAAAAAAAAAAAAAB6xAAAesQAAAAAAAAB6xAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
      AAAAAAAVGVzdCBmbGlnaHQgZHJvbmUAAAAAAAAAAAAAAAAAAAABAAAAAAA
      AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAAADAAAAAAAAAAAAAAAAA
      AAAAAAAAE5MRC1PUC0xMjM0NTY3AAAAAAAAAAAAAAEAAAAAAAAAAAAAAAA
      AAAAAAAABAQEAAAAAAAAA"
   }
}
```

Figure 5 - Example remote ID data message.

This message contains several sections:

- *protocol* - indicates the protocol version. Currently only protocol 1.0 exists.
- *data* - a data message contains a section data.
- *sensor ID* - the sensor ID set in dronescout.conf
- *RSSI* - the RSSI of the received Remote ID packet.
- *channel* - the channel on which the Remote ID packet is received. It is zero for BLE Remote ID signals, otherwise it is the WiFi channel number.
- *timestamp* - the epoch time stamp of the message in milliseconds.
- *MAC address* - the MAC address that broadcast the Remote ID packet

- *type* - the remote ID type. It can be BLE legacy, BLE long range, WiFi NaN or WiFi beacon.
- *UASdata* - this contains the Remote ID data. It is *base64*-encoded. The binary data itself is the Open Drone ID structure defined on **line 401** in file opendroneid.h: https://github.com/opendroneid/opendroneid-core-c/blob/master/libopendroneid/opendroneid.h

It is shown below:

```
typedef struct ODID_UAS_Data {
    ODID_BasicID_data BasicID[ODID_BASIC_ID_MAX_MESSAGES];
    ODID_Location_data Location;
    ODID_Auth_data Auth[ODID_AUTH_MAX_PAGES];
    ODID_SelfID_data SelfID;
    ODID_System_data System;
    ODID_OperatorID_data OperatorID;

    uint8_t BasicIDValid[ODID_BASIC_ID_MAX_MESSAGES];
    uint8_t LocationValid;
    uint8_t AuthValid[ODID_AUTH_MAX_PAGES];
    uint8_t SelfIDValid;
    uint8_t SystemValid;
    uint8_t OperatorIDValid;
} ODID_UAS_Data;
```

The data structures used in ODID_UAS_Data are also defined in opendroneid.h file.

The current firmware version uses git commit *4785de4570e2ecd418543d130d16147108181d0e* of the Open Drone ID project.

In the next chapter, reference Python source code is described to subscribe to the broker and parse these MQTT messages.

4 MQTT SUBSCRIBER (REFERENCE CODE)

You need to write your own MQTT subscriber application to process the MQTT messages published by the DroneScout receivers.

Reference Python3 source code is available to process MQTT messages of the ds230 or ds240 receiver. It can be found here:
https://github.com/BluemarkInnovations/RemoteID-MQTT-subscriber

5 FIRMWARE UPDATE

The firmware can be updated to the latest version by executing the following update-script.

```
sh /root/update.sh
```

Note, this update script does not remove files, nor does it overwrite the file remote.conf.

Firmware history
The firmware history can be found at:
https://download.bluemark.io/dronescout/firmware/history.txt

6 WARRANTY

The product has a two-year warranty period, starting at the date of receiving the product. Outside warranty are issues like crash damage, improper use, (extreme) weather conditions that damages the product. The product is eligible for future firmware updates as described in the Chapter Firmware update.

7 MORE INFORMATION

If you need more information, please contact us at info@bluemark.io or by phone: +31 53 711 2104.

All contact information can be found at the *DroneScout* contact page:
https://dronescout.co/contact/