

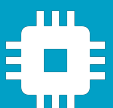
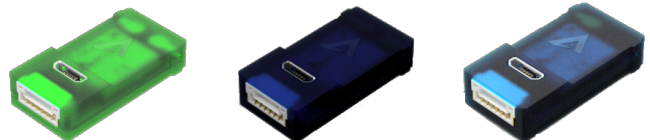


Subsystems for the
UAS integration into
the airspace

idME Remote ID

—

Data sheet - User manual



Contents

1	Introduction	4
1.1	Available variants	4
1.2	Features	4
2	Technical parameters	5
2.1	Basic technical information	5
2.2	Electrical specification	5
2.2.1	Basic electrical parameters	5
2.2.2	PIN definition	5
2.2.3	LED indicators	6
2.3	Mechanical specification	6
2.3.1	Mechanical parameters	6
2.3.2	Dimensions	6
2.3.3	Connectors	6
3	UART configuration	7
4	Principle of operation	8
4.1	States of operation	8
4.1.1	BOOTLOADER state	8
4.1.2	RUN state	8
4.1.3	CONFIGURATION state	8
4.2	Transitions between states	8
4.2.1	BOOTLOADER to RUN transition	8
4.2.2	RUN to CONFIGURATION transition	9
4.2.3	CONFIGURATION to RUN transition	9
4.2.4	CONFIGURATION to BOOTLOADER transition	9
5	System configuration	10
5.1	System settings	10
5.1.1	Write settings	10
5.1.2	Read settings	10
5.1.3	Settings description	10
5.1.4	Errors	11
5.1.5	Command endings	11
5.1.6	Uppercase and lowercase	11
5.1.7	Settings	11
5.1.8	Example	12
5.2	Commands	12
5.2.1	Commands in BOOTLOADER and CONFIGURATION state	12
5.2.2	Commands in CONFIGURATION state	12
5.2.3	Commands in RUN state	15
6	RemoteID transceiver subsystem	16
6.1	Settings	16
6.2	Protocols	17
6.2.1	RemoteID CSV protocol	17
6.2.2	RemoteID MAVLink protocol	18
7	GNSS receiver subsystem	19
7.1	Settings	19
7.2	Protocols	19
7.2.1	GNSS NMEA RAW protocol	19
7.2.2	GNSS JSON DECODED protocol	19
8	Sensors receiver subsystem	21
8.1	Settings	21

8.2	Protocols	21
8.2.1	Pressure CSV protocol	21
8.2.2	Sensor JSON protocol	21
9	Quick start	23
9.1	Configuration	23
9.1.1	Integration with Mavlink	23
10	General information	26
10.1	How to mount the device	26
10.2	Operator number	26
10.3	Status Led	27
10.4	Device status indicator	27
10.5	Troubleshooting	27
10.5.1	Range problem	27
10.5.2	Low Frame rate	27
10.5.3	IdMe still blinking fast	27
10.5.4	After start no message are sending	28
10.6	FCC Statement	28
10.6.1	FCC Compliance Statement	28
10.6.2	FCC Interference Statement	28
10.6.3	FCC Caution	28
11	Disclaimer	29

1 Introduction

The **idME**'s are designed to meet requirements of remote drone identification and localization in **ASTM/ASD-STAN** standard. Using the **BLE** broadcast and **WiFi Nan, Beacon** frames technology the device provides surveillance and drone operator identification capability based on any modern mobile devices such as smartphone or tablet. It is equipped with a high quality **multi-GNSS** receiver and a barometric altitude sensor.

Important:

Each firmware version is documented separately. This document is relevant for firmware version v1.33.2. If your firmware version is different, please find relevant documentation on our website aerobits.pl.

1.1 Available variants

VARIANT	BLE	Wi-Fi	GNSS	Pressure sensor
idME	•			
idME+	•		•	•
idME PROn	•	•		
idME PRO	•	•	•	•

Warning:

idME without GNSS requires an external position source.

1.2 Features

- Capability to work with MAVLINK devices
- WiFi Nan and Beacon frames
- BLE broadcast technology compliant with ASTM and ASD-STAN
- Interfaces: UART, USB
- Supports Bluetooth 4.0 and 5.2
- Free Android application available on Google Play [OpenDroneID OSM](#)
- Integrated GNSS source and pressure sensor
- Simple plug&play integration

For more information please contact support@aerobits.pl.

2 Technical parameters

2.1 Basic technical information

Table 1: General technical parameters

Parameter	Description	Typ.	Unit
First Band	BLE	2400	MHz
Second Band	Wi-Fi	2400	MHz
Third Band	GNSS	1575	MHz
Max. output (BLE)	Maximum output power	+18	dBm
Max. output (Wi-Fi)	Maximum output power	+20	dBm
Sensitivity (GNSS)		-167	dBm
UART	AT commands	921600	bps
USB	AT commands		

2.2 Electrical specification

2.2.1 Basic electrical parameters

Table 2: General electrical parameters

Parameter	Value
Power connector	JST GH 6 PINS or Micro USB Type B
Power supply	5.0 V
Power consumption	< 130 mA

2.2.2 PIN definition

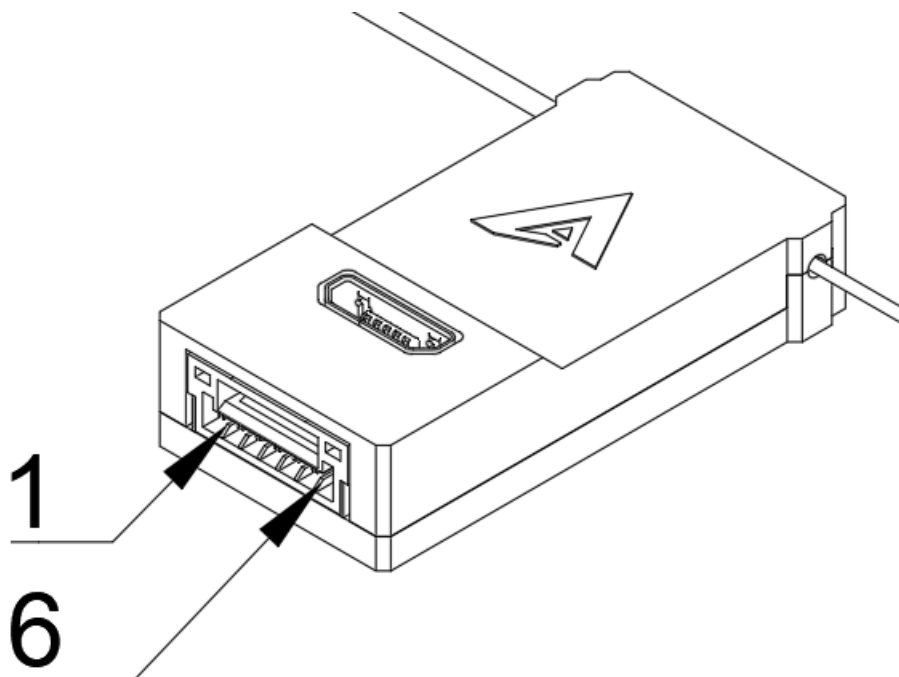


Table 3: Descriptions of idME connector pins

Pin number	Pin Name	Pin Type	Description
1	5V	IN	Power supply input
2	RX	IN	Main UART RXD
3	TX	OUT	Main UART TXD
4	USER_PIN_1	OUT	USER_PIN_1 logical HIGH/LOW set by AT-command
5	NC	N/A	No commercial use (keep floating)
6	GND	PWR	Ground

2.2.3 LED indicators

Table 4: Descriptions of LEDs

LED	Color	Function
POWER	White	Power supply indicator
STATUS	White	Device operation status

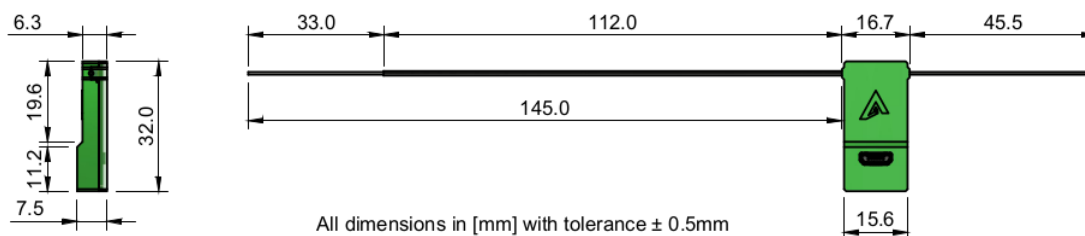
2.3 Mechanical specification

2.3.1 Mechanical parameters

Table 5: Mechanical parameters of the idME's

Parameter	Value
Dimensions	51 mm x 30 mm x 1 mm
Weight	30 g

2.3.2 Dimensions



2.3.3 Connectors

Table 6: Descriptions of used connectors

Description	Type	Function	Mating connector
JST GH	SM06B-GHS-TB(LF)(SN)	Power and Data	GHR-06V-S
Micro USB	USB3160-30-0170-0-C	Power and Data	CUB-100-BK
Antenna connectors	U.F.L socket	RF IN/OUT	U.F.L Plug

3 UART configuration

Communication between module and host device is done using UART interface.

In CONFIGURATION and BOOTLOADER state transmission baud is fixed at 115200bps.

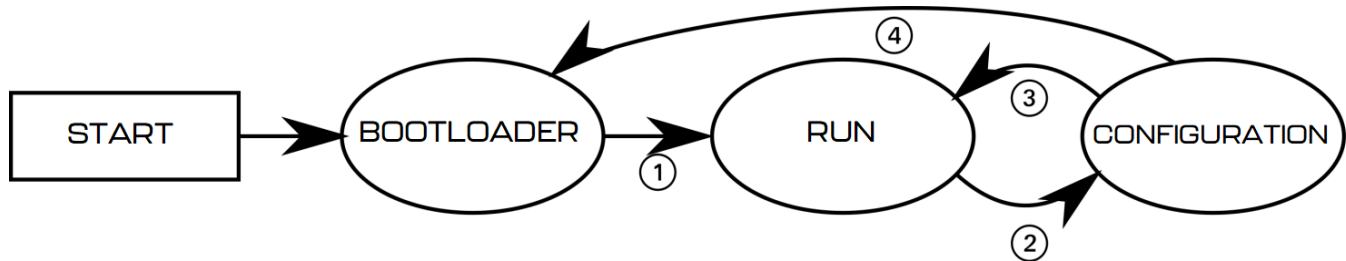
The UART interface uses settings as described in table below:

Table 7: Descriptions of UART settings.

Parameter	Min.	Typ.	Max	Unit
Baud	57600	921600	3000000	bps
Stop Bits Number	—	1	—	—
Flow Control	—	None	—	—
Parity Bit	—	None	—	—

4 Principle of operation

During work module goes through multiple states. In each state operation of the module is different. Each state and each transition is described in paragraphs below.



4.1 States of operation

4.1.1 BOOTLOADER state

This is an initial state of after restart. Firmware update is possible here. Typically module transitions automatically to RUN state. It is possible to lock module in this state (prevent transition to RUN state) using one of BOOTLOADER triggers. UART baud is constant and is set to 115200bps. After powering up module, it stays in this state for 3 seconds. If no BOOTLOADER trigger is present, module will transition to RUN state. Firmware upgrade is possible using Micro ADS-B App software. For automated firmware upgrading scenarios, aerobits_updater software is available. To acquire this program, please contact: support@aerobits.pl.

4.1.2 RUN state

In this state module is broadcasting drone identification data. In this state module is working and receiving the data from aircrafts. It uses selected protocol to transmit received and decoded data to the host system. In this state of operation module settings are loaded from non-volatile internal memory, including main UART interface's baud.

4.1.3 CONFIGURATION state

In this mode change of stored settings is possible. Operation of the module is stopped and baud is set to fixed 115200bps. Change of settings is done by using AT-commands. Changes to settings are stored in non-volatile memory on exiting this state. Additional set of commands is also available in this state, allowing to e.g. reboot module into BOOTLOADER state, check serial number and firmware version. It is possible to lock module in this state (similarly to BOOTLOADER) using suitable command.

4.2 Transitions between states

For each state transition, different conditions must be met, which are described below. Generally, the only stable state is RUN. Module always tends to transition into this state. Moving to other states requires host to take some action.

4.2.1 BOOTLOADER to RUN transition

BOOTLOADER state is semi-stable: the module requires additional action to stay in BOOTLOADER state. The transition to RUN state will occur automatically after a short period of time if no action is taken. To prevent transition from BOOTLOADER state, one of following actions must be taken:

- Send AT+LOCK=1 command while device is in BOOTLOADER state (always after power on for up to 3s)
- Send AT+REBOOT_BOOTLOADER command in CONFIGURATION state. This will move to BOOTLOADER state and will lock module in this state.

If none of above conditions are met, the module will try to transition into RUN state. Firstly it will check firmware integrity. When firmware integrity is confirmed, module will transition into RUN state, if not, it will stay in BOOTLOADER state.

To transition into RUN state:

- If module is locked, send `AT+LOCK=0` command

When module enters RUN mode, it will send `AT+RUN_START` command.

4.2.2 RUN to CONFIGURATION transition

To transition from RUN into CONFIGURATION state:

- Send `AT+CONFIG=1` (using current baud).

When module leaves RUN state, it sends `AT+RUN_END` message, then `AT+CONFIG_START` message on entering CONFIGURATION state. The former is sent using baud from settings, the latter always uses 115200bps baud.

4.2.3 CONFIGURATION to RUN transition

To transition from CONFIGURATION into RUN state:

- Send `AT+CONFIG=0` command.

When module leaves CONFIGURATION state, it sends `AT+CONFIG_END` message, then `AT+RUN_START` message on entering RUN state. The former is always sent using 115200bps baud, the latter uses baud from settings.

4.2.4 CONFIGURATION to BOOTLOADER transition

To transit from CONFIGURATION into BOOTLOADER state, host should do one of the following:

- Send `AT+REBOOT_BOOTLOADER` command.
- Send `AT+REBOOT` and when module enters BOOTLOADER state, prevent transition to RUN state.

When entering the bootloader state, the module sends `AT+BOOTLOADER_START`.

5 System configuration

In RUN state, operation of the module is determined based on stored settings. Settings can be changed in CONFIGURATION state using AT-commands. Settings can be written and read.

Note:

New values of settings are saved in non-volatile memory when transitioning from CONFIGURATION to RUN state.

Settings are restored from non-volatile memory during transition from BOOT to RUN state. If settings become corrupted due to memory fault, power loss during save, or any other kind of failure, the settings restoration will fail, loading default values and displaying the AT+ERROR (Settings missing, loaded default) message as a result. This behavior will occur for each device boot until new settings are written by the user.

5.1 System settings

5.1.1 Write settings

After writing a new valid value to a setting, an AT+OK response is always sent.

AT+SETTING=VALUE

For example AT+SYSTEM_STATISTICS=1

Response: AT+OK

5.1.2 Read settings

AT+SETTING?

For example: AT+SYSTEM_STATISTICS?

Response: AT+SYSTEM_STATISTICS=1

5.1.3 Settings description

AT+SETTING=?

For example: AT+SYSTEM_STATISTICS=?

Response:

```
Setting: SYSTEM_STATISTICS
Description: System statistics protocol(0:none, 1:CSV, 2:JSON)
Access: Read Write
Type: Integer decimal
Range (min.): 0
Range (max.): 2
Preserved: 1
Requires restart: 0
```

5.1.4 Errors

Errors are reported using following structure:

AT+ERROR (DESCRIPTION)

DESCRIPTION is optional and contains information about error.

5.1.5 Command endings

Every command must be ended with one of the following character sequences: "\n", "\r" or "\r\n". Commands without suitable ending will be ignored.

5.1.6 Uppercase and lowercase

All characters (except preceding AT+) used in command can be both uppercase and lowercase, so following commands are equal:

AT+SYSTEM_STATISTICS?

AT+sYSTEM_staTISTICS?

Note:

This statement is true in configuration state, not in bootloader state. In bootloader state all letters must be uppercase.

5.1.7 Settings

Table 8: Descriptions of system settings.

Setting	Min	Max	Def	Comment
BAUDRATE	0	3	0	Baudrate in RUN state 0 - 115200bps 1 - 921600bps 2 - 3000000bps 3 - 57600bps
SYSTEM_LOG	0	1	0	System logs 0 - disable 1 - enable
SYSTEM_STATISTICS	—	—	None	System statistics protocol: None CSV
USER_PIN_1	0	1	0	Set the logical level high 0 - disable 1 - enable

Note:

USER_PIN_1 in the MP1 product series turns on power to the active GNSS antenna.

5.1.8 Example

As an example, to switch the Aerobits device to CSV protocol, one should send following commands: “<<” indicates command sent to module, “>>” is a response.

```
<< AT+CONFIG=1\r\n
>> AT+OK\r\n
<< AT+ADSB_RX_PROTOCOL_DECODED=1\r\n
>> AT+OK\r\n
<< AT+CONFIG=0\r\n
>> AT+OK\r\n
```

5.2 Commands

Apart from settings, module supports a set of additional commands. Format of these commands is similar to those used for settings, but they do not affect operation of module in RUN state.

5.2.1 Commands in BOOTLOADER and CONFIGURATION state

AT+LOCK

AT+LOCK=1 - Set lock to enforce staying in BOOTLOADER or CONFIGURATION state

AT+LOCK=0 - Remove lock

AT+LOCK? - Check if lock is set

AT+BOOT

AT+BOOT? - Check if module is in BOOTLOADER state

Response:

AT+BOOT=0 - module in CONFIGURATION state

AT+BOOT=1 - module in BOOTLOADER state

5.2.2 Commands in CONFIGURATION state

AT+CONFIG

AT+CONFIG=0 - Transition to RUN state.

AT+CONFIG? - Check if module is in CONFIGURATION state.

Response:

AT+CONFIG=0 - module in RUN state

AT+CONFIG=1 - module in CONFIGURATION state (baudrate 115200)

AT+CONFIG=2 - module in CONFIGURATION state (baudrate as set)

AT+SETTINGS?

AT+SETTINGS? - List all settings. Example output:

```
AT+BAUDRATE=0
AT+BOOT=0
AT+CONFIG=1
AT+DEVICE=TR-1F
AT+FIRMWARE_VERSION=2.72.1.0 (Jun 17 2024)
AT+LOCK=0
AT+SERIAL_NUMBER=22-0000309
AT+SYSTEM_LOG=0
AT+SYSTEM_STATISTICS=0
AT+ADSB_RX_PROTOCOL_DECODED=1
AT+ADSB_RX_PROTOCOL_INC=0
AT+ADSB_RX_PROTOCOL_RAW=0
AT+ADSB_STATISTICS=1
AT+ADSB_TX_EMITTER_CAT=0
AT+ADSB_TX_ENABLED=1
AT+ADSB_TX_ICAO=000000
AT+ADSB_TX_IDENT=
AT+ADSB_TX_ON_BOOT=1
AT+ADSB_TX_PWR=2
AT+ADSB_TX_SQUAWK=0000
AT+ADSB_TX_SURFACE=0
AT+ADSB_TX_TRANSPONDER_PRESENT=0
AT+FLARM_INFO=LIBFLARM-2.03, expires: 2025-03-01, status: OK
AT+FLARM_RX_PROTOCOL_DECODED=1
AT+FLARM_STATISTICS=0
AT+FLARM_TX=1
AT+FLARM_TX_AIRCRAFT_TYPE=13
AT+GNSS_RX_PROTOCOL_RAW=0
AT+SENSOR_PROTOCOL_DECODED=0
AT+ASTERIX_SAC=1
AT+ASTERIX_SIC=129
```

AT+HELP

AT+HELP - Show all settings and commands with descriptions. Example output:

```
SETTINGS:
SYSTEM:
  AT+BAUDRATE=0 [Baudrate of serial interface (0:115200, 1:921600, 2:3000000,
  ↪ 3:57600) ]
  AT+BOOT=0 [Is firmware in bootloader mode]
  AT+CONFIG=1 [CONFIG mode (0:disable, 1:baudrate 115200, 2:baudrate as set)]
  AT+DEVICE=IDME-PRO [Device type's name]
  AT+LOCK=0 [Device in CONFIG mode (0:no lock, 1:lock)]
  AT+SERIAL_NUMBER=18099300000323 [Device's serial number]
  AT+SYSTEM_LOG=0 [System logs (0:disable, 1:enable)]
  AT+SYSTEM_STATISTICS=0 [System statistics protocol(0:none, 1:CSV, 2:JSON)]
  AT+FIRMWARE_VERSION=1.22.5.0 (Aug 7 2024) [Device's firmware version]
GNSS:
  AT+GNSS_RX_PROTOCOL_RAW=0 [GNSS_RX RAW protocol (0:none, 5:NMEA)]
```

(continues on next page)

(continued from previous page)

```
SENSORS:
  AT+SENSORS_PROTOCOL_DECODED=0 [SENSORS decoded protocol (0:none, 1:CSV, 3:JSON)]
COMMANDS:
  AT+3RD_PARTY_LICENSES [Displays licenses of third party software]
  AT+BLUETOOTH_MAC [Bluetooth device mac address]
  AT+DRONE_ID_OPERATOR_ID [Operator message payload]
  AT+HELP [Show this help]
  AT+INFO [Display device information]
  AT+REBOOT [Reboot system]
  AT+REBOOT_BOOTLOADER [Reboot to bootloader]
  AT+SETTINGS_DEFAULT [Loads default settings]
  AT+TEST [Responds "AT+OK"]
  AT+WIFI_MAC [WiFi device mac address]
```

AT+SETTINGS_DEFAULT

AT+SETTINGS_DEFAULT - Set all settings to their default value.

AT+SERIAL_NUMBER

AT+SERIAL_NUMBER? - Read serial number of module.

Response:

AT+SERIAL_NUMBER=07-0001337

AT+FIRMWARE_VERSION

AT+FIRMWARE_VERSION? - Read firmware version of module.

Response:

AT+FIRMWARE_VERSION=2.73.1.0 (Jun 27 2024)

AT+REBOOT

AT+REBOOT - Restart module.

AT+REBOOT_BOOTLOADER

AT+REBOOT_BOOTLOADER - Restart module to BOOTLOADER state.

Note:

NOTE: This command also sets lock.

5.2.3 Commands in RUN state

AT+CONF IG=1 - transition to CONFIGURATION state (baudrate 115200). AT+CONF IG=2 - transition to CONFIGURATION state (baudrate as set).

Note:

NOTE: This command also sets lock.

6 RemoteID transceiver subsystem

6.1 Settings

Table 9: Descriptions of RemoteID settings.

Setting	Min	Max	Def	Comment
DRONE_ID_ADVERTISING_ENABLE	0	1	1	Advertising enable
DRONE_ID_BASIC_BROADCAST_PERIOD	200	3000	1500	Basic messages broadcast period in [ms]
DRONE_ID_BROADCAST_BLUETOOTH_4	0	1	1	Enable Bluetooth 4.0 broadcast
DRONE_ID_BROADCAST_BLUETOOTH_5	0	1	1	Enable Bluetooth 5.0 broadcast
DRONE_ID_BROADCAST_WIFI_BEACON	0	1	1	Enable Wifi Standard Beacon broadcast
DRONE_ID_BROADCAST_WIFI_NAN_BEACON	0	1	0	Enable WiFi NaN Beacon broadcast
DRONE_ID_DRONE_CATEGORY_CLASS	0	7	0	Drone category class: 0 – None 1 – C0 2 – C1 3 – C2 4 – C3 5 – C4 6 – C5 7 – C6
DRONE_ID_HEIGHT_TYPE	0	1	0	Height type: 0 – Relative to take-off location 1 – Relative to ground
DRONE_ID_LOCALIZATION_BROADCAST_PERIOD	100	1000	500	Location message broadcast period in [ms]
DRONE_ID_MAVLINK_CONNECTION_TIMEOUT	2	30	5	Mavlink connection timeout in [s]
DRONE_ID_MODE	0	1	1	Determines Mavlink reception: 0 - Full MAVLink support 1 - Ignore all MAVLink messages 2 - Hybrid MAVLink mode
DRONE_ID_OPERATIONAL_STATUS	0	2	0	Operational status: 0 – Undeclared 1 – Ground 2 – Airborne
DRONE_ID_OPERATION_CATEGORY	0	3	0	Operation category: 0 – None 1 – Open 2 – Specific 3 - Certified
DRONE_ID_OPERATOR_ID	—	—	—	Operator message payload

continues on next page

Table 9 – continued from previous page

Setting	Min	Max	Def	Comment
DRONE_ID_OPERATOR_ID_TYPE	0	255	0	Operator ID type: 0 - Operator ID 1 – 200 Reserved 201 – 255 Available for private use
DRONE_ID_SELF_ID	–	–	–	Self message payload
DRONE_ID_SELF_ID_TYPE	0	255	0	Self ID type: 0 - Text description 1 – 200 Reserved 201 – 255 Available for private use
DRONE_ID_CAA_REG_ID	–	–	–	CAA Assigned Registration ID, empty not sent
DRONE_ID_UTM_UUID	–	–	–	UTM Assigned UUID (128-bits), input must be 32 signs HEX string, empty not sent
DRONE_ID_SESSION_ID	–	–	–	Specific Session ID (20-bytes), input must be 40 signs HEX string, empty not sent
DRONE_ID_UAS_TYPE	0	15	0	Specification of the type of UAS: 0 – None 1 – Aeroplane 2 - Helicopter or Multirotor 3 – Gyroplane 4 - Hybrid Lif 5 - Ornithopter 6 – Glider 7 – Kite 8 - Free Balloon 9 - Captive Balloon 10 – Airship 11 - Free Fall 12 – Rocket 13 - Tethered Powered Aircraft 14 - Ground Obstacle 15 – Other

6.2 Protocols

6.2.1 RemoteID CSV protocol

Statistics message

This message contains some useful statistics about operation of module. Format of that frame is shown below:

```
#SR:FPS,FPB4_0S,FPB4_1S,FPB4_3S,FPB4_4S,FPB4_5S,FPB5S,FPNS,FPBS,CRC\r\n
```

Table 10: Descriptions of RemoteID Statistic frame.

#SR	Statistics message start indicator	Example
FPS	Number of frame received in last second	1
FPB4_0S	Number of Basic ID message Bluetooth 4.0 frame send in last second	1
FPB4_1S	Number of Location message Bluetooth 4.0 frame send in last second	1
FPB4_3S	Number of Self ID message Bluetooth 4.0 frame send in last second	1
FPB4_4S	Number of System message Bluetooth 4.0 frame send in last second	1
FPB4_5S	Number of Operator ID message Bluetooth 4.0 frame send in last second	1
FPB5S	Number of Message Pack Bluetooth 5.0 frame send in last second	1
FPNS	Number of Message Pack Wi-Fi NaN frame send in last second	1
FPBS	Number of Message Pack Wi-Fi Beacon frame send in last second	1
CRC	CRC16 (described in CRC section)	2D3E

6.2.2 RemoteID MAVLink protocol

RemoteID has MAVLink protocol autodetect mode, if input data will be in MAVLink mode then device automatically switch to this protocol. This option will be available only when standalone mode is not enabled. For fully detailed information about MAVLink protocol take a look [here](#)

List of supported MAVLink messages:

- HEARTBEAT (0)
- SYSTEM_TIME (2)
- GPS_RAW_INT (24)
- SCALED_PRESSURE (29)
- GLOBAL_POSITION_INT (33)
- OPEN_DRONE_ID_BASIC_ID (12900)
- OPEN_DRONE_ID_LOCATION (12901)
- OPEN_DRONE_ID_SELF_ID (12903)
- OPEN_DRONE_ID_SYSTEM (12904)
- OPEN_DRONE_ID_OPERATOR_ID (12905)

7 GNSS receiver subsystem

7.1 Settings

Table 11: Descriptions of GNSS settings

Setting	Min	Max	Def	Comment
GNSS_RX_PROTOCOL_RAW	NONE	NMEA	NMEA	GNSS_RX RAW protocol select NONE NMEA
GNSS_RX_PROTOCOL_DECODED	NONE	JSON	NONE	GNSS_RX Decoded protocol select NONE JSON

7.2 Protocols

7.2.1 GNSS NMEA RAW protocol

Note:

For more information about all NMEA GNSS fields go to [docs](#).

7.2.2 GNSS JSON DECODED protocol

The *gnss* section contains basic GNSS information. This message is sent once per second. The example JSON message with “gnss” section fields described:

```
{
  "src": "ID-0000001",
  "ts": 69061337,
  "ver": 1,
  "gnss": {
    "fix": 1,
    "lat": 53.42854,
    "lon": 14.55281,
    "altWgs84": 499.6,
    "altMsl": 508.6,
    "track": 127.3,
    "hVelo": 10.5,
    "vVelo": 25,
    "gndSpeed": [
      5.2,
      2.1
    ],
    "acc": {
      "lat": 5.2,
      "lon": 2.1,
      "alt": 3.6
    },
    "nacp": 12,
    "nacv": 2,
```

(continues on next page)

(continued from previous page)

```

    "nic": 12
  }
}

```

Table 12: Descriptions of JSON GNSS section fields.

JSON Field	Unit	Example	Description
gnss			Type of message
fix	—	1	Set to 1 if onboard GNSS currently has fix, otherwise 0.
lat	—	53.42854	Last known latitude. Omitted if there was no GNSS fix since device boot.
lon	—	14.55281	Last known longitude. Omitted if there was no GNSS fix since device boot.
altWgs84	—	499.6	Last known WGS-84 Altitude, in meters. Omitted if there was no GNSS fix since device boot.
altMsl	—	508.6	Last known MSL Altitude, in meters. Omitted if there was no GNSS fix since device boot.
track	—	127.3	Track angle, 0°..360°, relative to true north. Omitted if unknown.
hVelo	—	10.5	Horizontal velocity, in knots. Omitted if unknown.
vVelo	—	25	Vertical velocity, in m/s. Positive value is upwards. Omitted if unknown.
gndSpeed	knots	[5.2,2.1]	Ground speed in east-west and north-south axes respectively, in knots. Positive value is East and North. Derived from track / hVelo values. Omitted if unknown.
acc	m/s ²	struct	Acceleration in all 3 dimensions
lat	—	5.2	Accuracy of latitude, in meters. Omitted if unknown.
lon	—	2.1	Accuracy of longitude, in meters. Omitted if unknown.
alt	—	3.6	Accuracy of altitude, in meters. Omitted if unknown.
nacp	—	12	Navigational Accuracy Category for Position value, as defined in ED-282. Omitted if unknown.
nacv	—	2	Navigational Accuracy Category for Velocity value, as defined in ED-282. Omitted if unknown.
nic	—	12	Navigation Integrity Category as defined in ED-282. Omitted if unknown.

8 Sensors receiver subsystem

8.1 Settings

Table 13: Descriptions of Sensors settings.

Setting	Min	Max	Def	Comment
SENSORS_RX_PROTOCOL_RAW	—	—	None	Sensors decoded protocol: None CSV JSON

8.2 Protocols

8.2.1 Pressure CSV protocol

This message describes state vector of sensor determined from SENSORS messages and is sent once per second. The message format is as follows:

```
#SP:CALIB,PRESS,TEMP,CRC
```

Table 14: Descriptions of SENSORS fields.

#SP	Sensors message start indicator	Example value
CALIB	Pressure sensor calibration value	1
PRESS	Current pressure value	1002.213742
TEMP	Current temperature value	56.420123
CRC	CRC16 (described in CRC section)	2D3E

8.2.2 Sensor JSON protocol

The *sensor* section contains values acquired from miscellaneous sensors present in Aerobits device hardware and consists of fields shown below. This message is sent once per second. All fields are optional - they are sent only if appropriate sensor is enabled.

```
{
  "ver": 1,
  "sensor": {
    "pressure": 1006.87,
    "temp": 39.8
  },
  "HumiditySensor": {
    "Temperature": 36.9,
    "Humidity": 19,
  }
}
```

Table 15: Descriptions of JSON Sensor section fields.

JSON Field	Unit	Example	Description
ver	—	1	JSON protocol version.
sensor	—	type of sensor	

continues on next page

Table 15 – continued from previous page

JSON Field	Unit	Example	Description
pressure	hPa	1006.87	Current pressure sensor value in hPa.
temp	°C	39.8	Current temperature sensor value in °C.
HumiditySensor	—	type of sensor	
Temperature	°C	36.9	Current temperature sensor value in °C.
Humidity	%	19	Current humidity sensor value in %.

9 Quick start

9.1 Configuration

1. configure the device settings and assign an operator number using Micro ADS-B software via USB or UART interface.
2. mount the device on the drone.
3. connect the power supply through the JST connector directory of the flight controller or MicroUSB.
4. observe the LED indicating that the device is ready for flight - the LED blinks slowly

Note:

Baro Altitude is computed based on pressure measured when IdMe is started up.

9.1.1 Integration with Mavlink

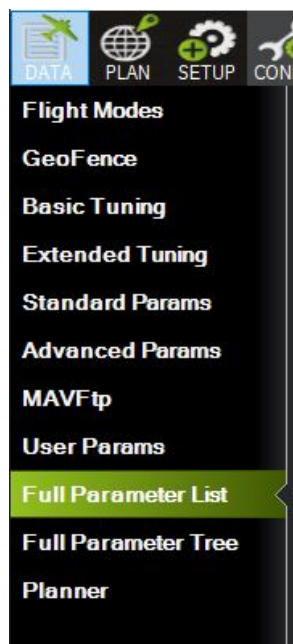
1. Connect IdMe PRO to device supports Mavlink V2 protocol. In this example IdMe+ will be connected to Pixhawk TELEM1 port using JST connectors with not crossed wire(TX and RX are not switched)
2. Using Mission Planner software enable Mavlink V2 protocol on TELEM 1 port
3. Connect Pixhawk to Mission Planner



4. Select CONFIG tab



5. Select Full Parameter List



6. Find SERIAL1_BAUD parameter and set a value to 115

Flight Modes	Command	Δ	Value	Units	Options
GeoFence	BRD_SER1_RTSCS		0		0:Disabled 1:Enabled 2:Auto
Basic Tuning	BRD_SER2_RTSCS		0		0:Disabled 1:Enabled 2:Auto
Extended Tuning	RSSI_TYPE		0		0:Disabled 1:AnalogPin 2:RCChannelPwmV 3:ReceiverProtocol 4:PWMInputPin
Standard Params	SERIAL1_BAUD		115		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 256:256000 500:500000 921:921600 1500:1500000
Advanced Params					
MAVFtp					
User Params					
Full Parameter List					
Full Parameter Tree					

7. Find SERIAL1_PROTOCOL parameter and set a value to 2

Flight Modes	Command	Value	Units
GeoFence			
Basic Tuning			
Extended Tuning			
Standard Params			
Advanced Params			
MAVftp			
User Params			
Full Parameter List	SERIAL1_PROTOCOL	2	
Full Parameter Tree			
Planner			

8. Reboot Pixhawk

After configuration device is ready to work when status led starts blinking slowly (once every second). For more information visit: [Mission Planner documentation](#).

10 General information

IdMe is an add-on device. This means that it does not need any additional components to work, it is equipped with a high-quality multi-GNSS receiver and a barometric altitude sensor. Using BLE and WiFi transmission technology, the device provides surveillance and the ability to identify the drone operator based on any modern mobile device such as a smartphone or tablet.

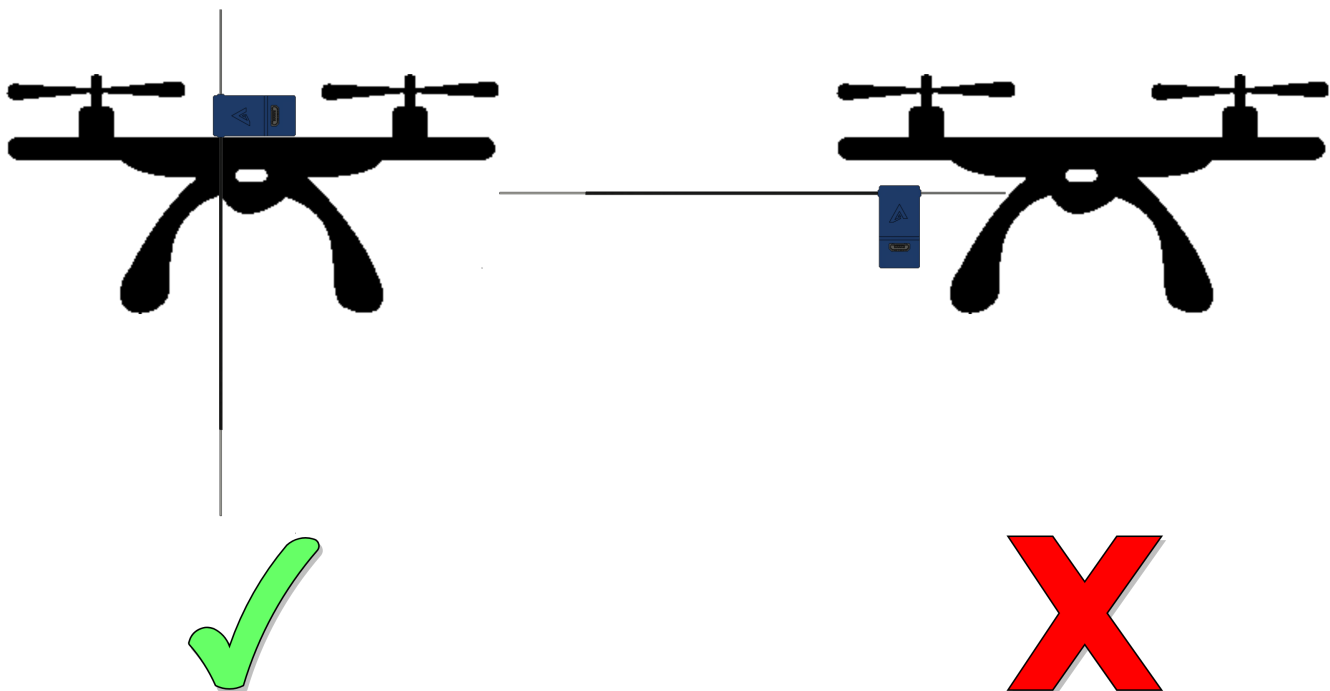
The device automatically detects the drone's start and immediately starts transmitting a broadcast until the drone is turned off.

Its small size and low power consumption allow it to be used in ultralight drones. AT commands provide the ability to configure the messages to be transmitted, such as the drone's identification number, aircraft type, etc. Additional authentication mechanisms are also available.

10.1 How to mount the device

The device should be mounted as far as possible from devices that generate RF interference, such as built-in antennas, speed controllers and motors. Pay attention to moving parts, as well as propellers or the camera. Since the device is lightweight, you can attach it with tape or glue. It is a good idea to leave room to access the device for software updates or configuration.

The device should be mounted in such an orientation that the antenna from the GPS(the shorter one) is vertically up, while the transmitting antenna is vertically down. This configuration allows you to get the best omnidirectional transmission, and reduce the time you have to wait for the GNSS position to be determined.



10.2 Operator number

The operator number can be obtained from the country's state registration system. This number must be entered into the device using the AT command or the Micro ADS-B software. The writing process requires 3 additional extra digits used to check integrity or provide temperament.

Table 16: Operator number

OPERATOR NUMBER	SEPARATOR	SECURE CHARACTERS
AAABBBBBBBBBBBBC	—	DDD

Example command: `AT+DRONE_ID_OPERATOR_ID=AAABBBBBBBBBBBBC-DDD`

If the Operator ID contains any error, the message “Operator ID not correct!” will appear.

By default, the device broadcasts the serial number assigned to the device in the manufacturing process, this number cannot be overwritten.

10.3 Status Led

When the device is in bootloader or configuration mode the led diode lights up continuously. In boot mode or when the device has an error the led blinks very fast. If the device is ready to fly the led blinks slowly, once per second.

Table 17: Status LED

DEVICE STATUS	STATUS LED	POWER LED
boot	light	light
configuration	blink fast	light
error or calibration	blink fast	light
ready to fly, airborne	blink slowly	light

10.4 Device status indicator

If an error occurs, it can be easily detected by observing the STATUS LED. The device automatically changes its Remote ID status to emergency. Additional information is described in the SelfID message, which can be easily identified by other airspace users.

10.5 Troubleshooting

10.5.1 Range problem

Most coverage problems occur when the device is mounted in the wrong place or with the wrong orientation.

- check the antenna and ufl connector
- change the position and orientation of the device

10.5.2 Low Frame rate

In long range or radio interference environments, some frames may be missing.

- decrease period between broadcasts using DRONE_ID_BASIC_BROADCAST_PERIOD and DRONE_ID_LOCALIZATION_BROADCAST_PERIOD parameters in configuration mode

10.5.3 IdMe still blinking fast

In most cases, the problem is the lack of a GNSS fix.

- Check GNSS antenna (the shorter one) is well connected to ufl connector
- Device is placed in wrong orientation
- Device is placed in heavy noisy environment.

10.5.4 After start no message are sending

The device does not receive GNSS corrections before launch. It is very important to wait until the device is ready for launch.

- Reset the module and wait for the GNSS position to be fixed, then start again.

10.6 FCC Statement

10.6.1 FCC Compliance Statement

This device complies with part 15 of the FCC rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference,
2. This device must accept any interference received, including interference that may cause undesired operation.

10.6.2 FCC Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna,
- Increase the separation between the equipment and receiver,
- Connect the equipment into an outlet on a circuit different from which the receiver is connected,
- Consult the dealer or an experienced radio/TV technician for help.

10.6.3 FCC Caution

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

11 Disclaimer

Information contained in this document is provided solely in connection with Aerobits products. Aerobits reserves the right to make changes, corrections, modifications or improvements to this document, and to products and services described herein at any time, without notice. All Aerobits products are sold pursuant to our own terms and conditions of sale. Buyers are solely responsible for the choice, selection and use of the Aerobits products and services described herein, and Aerobits assumes no liability whatsoever, related to the choice, selection or use of Aerobits products and services described herein. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services, it shall not be deemed a license granted by Aerobits for use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering use, in any manner whatsoever, of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN AEROBITS TERMS AND CONDITIONS OF SALE, AEROBITS DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO USE AND/OR SALE OF AEROBITS PRODUCTS INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED AEROBITS REPRESENTATIVE, AEROBITS PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE.

Information in this document supersedes and replaces all previously supplied information.

© 2024 Aerobits - All rights reserved