

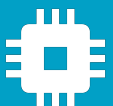


Subsystems for the  
UAS intergration into  
the airspace

# *E-identification idME*



Data sheet & User manual



# E-identification idME



## Introduction

**idME** is designed to meet requirements of remote drone identification and localization in **ASTM/ASD-STAN standard**. Using the BLE broadcast technology the device provides surveillance and drone operator identification capability based on any modern mobile devices such as smartphone or tablet.

For more information please contact: [support@aerobits.pl](mailto:support@aerobits.pl).

# Contents

<b>1</b>	<b>Technical parameters</b>	<b>4</b>
1.1	Basic technical information	4
1.2	Main features	4
<b>2</b>	<b>Electrical specification</b>	<b>5</b>
2.1	Basic electrical parameters	5
2.2	PIN definition	5
2.3	LED indicators	5
<b>3</b>	<b>Mechanical specification</b>	<b>6</b>
3.1	Mechanical parameters	6
3.2	Dimensions	6
3.3	Connectors	6
<b>4</b>	<b>Principle of operation</b>	<b>7</b>
4.1	States of operation	7
4.1.1	BOOTLOADER state	7
4.1.2	RUN state	7
4.1.3	CONFIGURATION state	7
4.2	Transitions between states	7
4.2.1	BOOTLOADER to RUN transition	7
4.2.2	RUN to CONFIGURATION transition	8
4.2.3	CONFIGURATION to RUN transition	8
4.2.4	CONFIGURATION to BOOTLOADER transition	8
<b>5</b>	<b>UART configuration</b>	<b>9</b>
<b>6</b>	<b>Settings</b>	<b>10</b>
6.1	Write settings	10
6.2	Read settings	10
6.3	Settings description	10
6.4	Errors	10
6.5	Command endings	10
6.6	Uppercase and lowercase	10
6.7	Available settings	11
6.8	Example	12
<b>7</b>	<b>Commands</b>	<b>13</b>
7.1	Commands in BOOTLOADER and CONFIGURATION state	13
7.1.1	AT+LOCK	13
7.1.2	AT+BOOT	13
7.2	Commands in CONFIGURATION state	13
7.2.1	AT+CONFIG	13
7.2.2	AT+SETTINGS?	13
7.2.3	AT+HELP	13
7.2.4	AT+SETTINGS_DEFAULT	13
7.2.5	AT+SERIAL_NUMBER	14
7.2.6	AT+FIRMWARE_VERSION	14
7.2.7	AT+REBOOT	14
7.2.8	AT+REBOOT_BOOTLOADER	14
7.3	Commands in RUN state	14
<b>8</b>	<b>Quick start</b>	<b>15</b>
8.1	Configuration	15
8.2	Integration with pixhawk and ardupilot	15
8.2.1	Supported Mavlink	17

<b>9</b>	<b>General information</b>	<b>18</b>
9.1	Operator number	18
9.2	Status Led	18
9.2.1	Device status indicator	19
9.3	Troubleshooting	19
9.3.1	Range problem	19
9.3.2	Low Frame rate	19
9.3.3	IdMe still blinking fast	19

# 1 Technical parameters

## 1.1 Basic technical information

Parameter	Description	Typ.	Unit
Frequency	Bluetooth	2.402 - 2.480	GHz
Max. output	Maximum output power	+8	dBm
ESD protection	All connectors		-
Interface baud	Configuraton or MAVLink	115200	bps
Main connector	SM06B-GHS-TB(LF)(SN)		-
Antenna connector	2x RF-IPX125-1G-AU		-
Dimension		32.0 x 16.7 x 7.5	mm
Weight (with antenna)		4	grams

Table 1: General technical parameters.

## 1.2 Main features

- Capability to work with MAVLINK devices
- 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](#)
- Simple plug&play integration

## 2 Electrical specification

### 2.1 Basic electrical parameters

Parameter	Value
Input voltage	5 V
Current consumption	15 mA

Table 2: General electrical parameters.

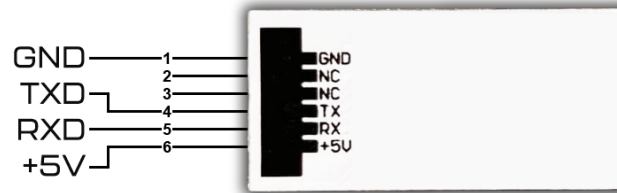


Figure 1: Appendant drawing of E-identification idME bottom.

### 2.2 PIN definition

PIN	Color	Name	Function
1	-	GND	Ground
2	-	NC	Not connected
3	-	NC	Not connected
4	-	TX	MAVLink, AERO TXD
5	-	RX	MAVLink, AERO RXD
6	-	+5 V	Power supply (5 V/ 70 mA)

Table 3: Pin definition.

### 2.3 LED indicators

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

Table 4: LED indicators.

## 3 Mechanical specification

### 3.1 Mechanical parameters

Parameter	Value
Dimensions	32.0 x 16.7 x 7.5mm
Weight	4 g

Table 5: Mechanical parameters of E-identification idME

### 3.2 Dimensions

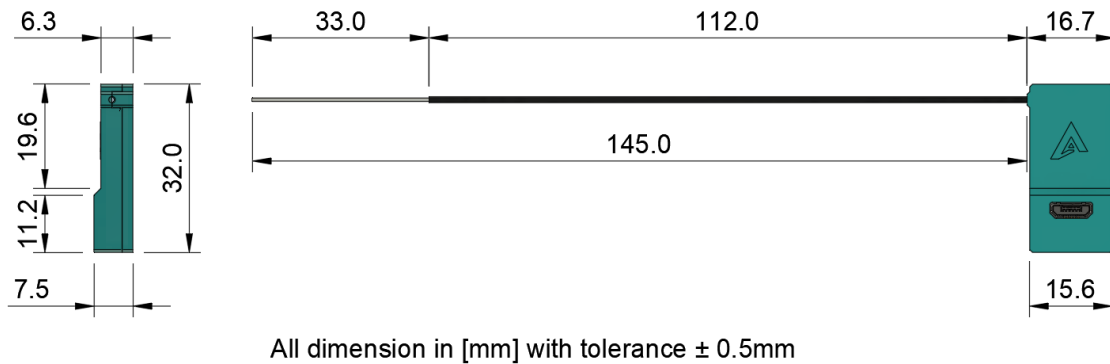


Figure 2: Dimensions of E-identification idME

### 3.3 Connectors

Connector	Type	Example
Main	Installed on board	SM06B-GHS-TB(LF)(SN)
	Mating connector	GHR-06V-S
	Pins	SSHL-002T-P0.2
Antenna	Installed on board	RF-IPX125-1G-AU
	Mating connector	GSM-IPX or GSM-IPX/SMA-1G-150

Table 6: Connectors

## 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.

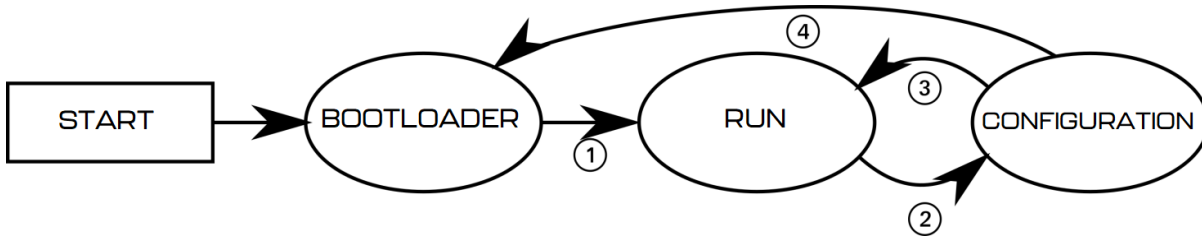


Figure 3: State machine of E-identification idME

### 4.1 States of operation

#### 4.1.1 BOOTLOADER state

This is an initial state of E-identification idME after restart. Firmware update is possible here. Typically module transits 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 up to 3 seconds. If no BOOTLOADER trigger is present, module will transit 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](mailto:support@aerobits.pl).

#### 4.1.2 RUN state

In this state module is broadcasting drone identification data.

#### 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 of state transitions, different conditions must be met, which are described below. Generally, the only stable state is RUN. Module always tends to transit 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 short period of time if no action will be taken. To prevent transition from BOOTLOADER state, one of following actions must be processed:

- 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 transit into RUN state. Firstly it will check firmware integrity. When firmware integrity is confirmed, module will transit into RUN state, if not, it will stay in BOOTLOADER state.



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

- 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 transit from CONFIGURATION into RUN state, host should do one of the following:

- 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 UART configuration

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

The UART interface uses settings as described in table 7.

UART Settings				
Parameter	Min.	Typ.	Max	Unit
Baud	115200	115200	3000000	bps
Stop Bits Number	-	1	-	-
Flow Control	-	None	-	-
Parity Bit	-	None	-	-

Table 7: UART settings.

## 6 Settings

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 do 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.

### 6.1 Write settings

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

```
AT+SETTING=VALUE  
For example AT+PROTOCOL=1  
Response: AT+OK
```

### 6.2 Read settings

```
AT+SETTING?  
For example: AT+PROTOCOL?  
Response: AT+PROTOCOL=1
```

### 6.3 Settings description

```
AT+SETTING=?  
For example: AT+PROTOCOL=?  
Response:
```

```
Setting: PROTOCOL  
  Description: Selected protocol (0: NONE, 2: CSV, 3: MAVLINK)  
  Type: Integer decimal  
  Range (min.): 0  
  Range (max.): 5  
  Is preserved: 1  
  Is restart needed: 0
```

### 6.4 Errors

Errors are reported using following structure:

```
AT+ERROR (DESCRIPTION)  
DESCRIPTION is optional and contains information about error.
```

### 6.5 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.

### 6.6 Uppercase and lowercase

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

AT+PROTOCOL?

AT+pRoToCoL?

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

## 6.7 Available settings

Setting	Min	Max	Def	Comment
BAUDRATE	0	2	0	Baudrate in RUN state 0 - 115200bps 1 - 921600bps 2 - 3000000bps
DRONE_ID_BASIC_BROADCAST_PERIOD	100	3000	2900	Basic frame broadcast period in [ms]
DRONE_ID_LOCALIZATION_BROADCAST_PERIOD	100	1000	900	Localization frame broadcast period in [ms]
DRONE_ID_ADVERTISING_ENABLE	0	1	1	Enable Bluetooth advertising
DRONE_ID_SCAN_ENABLE	0	1	0	Enable Bluetooth scan
DRONE_ID_HEIGHT_TYPE	0	1	0	Device Height type 0 - Above Takeoff 1 - AGL
MAVLINK_CONNECTION_TIMEOUT	0	99	5	Mavlink connection timeout in seconds
DRONE_ID_OPERATOR_ID	-	-	-	Operator ID (20 bytes )
DRONE_ID_OPERATOR_ID_TYPE	0	255	0	Operator ID type 0 - Operator ID 201-255 - Available for private use
DRONE_ID_OPERATIONAL_STATUS	0	2	0	Operation status 0 - Undeclared 1 - Ground 2 - Airborne
DRONE_ID_SELF_ID	-	-	-	Self ID (20 bytes )
DRONE_ID_SELF_ID_TYPE	0	255	0	Self ID type 0 - Text Description 201-255 - Available for private use
DRONE_ID_STANDALONE	0	1	0	Ignore streams from Mavlink protocol

DRONE_ID_TYPE	0	3	0	UAS ID type 0 - None 1 - Serial Number 2 - CAA Assigned Registration ID 3 - UTM Assigned UUID
DRONE_ID_UAS_TYPE	0	15	0	Specification of the type of UAS 0 - None 1 - Aeroplane 2 - Helicopter or Multicopter 3 - Gyroplane 4 - Hybrid Lift 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
DRONE_ID_UAS_ID	-	-	Device serial number	UAS ID (20 bytes)

Table 8: Settings

## 6.8 Example

As an example, to set parameter `DRONE_ID_OPERATOR_ID` for E-identification idME device, one should send following commands. “<<” indicates command sent to module, “>>” is a response.

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

## 7 Commands

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

### 7.1 Commands in BOOTLOADER and CONFIGURATION state

#### 7.1.1 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

#### 7.1.2 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

### 7.2 Commands in CONFIGURATION state

#### 7.2.1 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

#### 7.2.2 AT+SETTINGS?

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

AT+PROTOCOL=2

AT+SUBPROTOCOL=0

AT+BAUDRATE=0

#### 7.2.3 AT+HELP

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

SETTINGS:

AT+PROTOCOL=2 [Selected protocol (0: NONE, 2: CSV, 3: MAVLINK)]

AT+SUBPROTOCOL=0 [Subprotocol of selected protocol]

COMMANDS:

AT+HELP [Show this help]

AT+TEST [Responds "AT+OK"]

AT+SETTINGS\_DEFAULT [Load default settings]

AT+REBOOT [Reboot system]

#### 7.2.4 AT+SETTINGS\_DEFAULT

AT+SETTINGS\_DEFAULT - Set all settings to their default value.

### 7.2.5 AT+SERIAL\_NUMBER

AT+SERIAL\_NUMBER? - Read serial number of module.

Response:

```
AT+SERIAL_NUMBER=07-0001337
```

### 7.2.6 AT+FIRMWARE\_VERSION

AT+FIRMWARE\_VERSION? - Read firmware version of module.

Response:

```
AT+FIRMWARE_VERSION=10101017(May 11 2018)
```

### 7.2.7 AT+REBOOT

AT+REBOOT - Restart module.

### 7.2.8 AT+REBOOT\_BOOTLOADER

AT+REBOOT\_BOOTLOADER - Restart module to BOOTLOADER state.

**NOTE: This command also sets lock.**

## 7.3 Commands in RUN state

AT+CONFIG=1 - transition to CONFIGURATION state.

**NOTE: This command also sets lock.**

## 8 Quick start

### 8.1 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.

### 8.2 Integration with pixhawk and ardupilot

1. Connect IdMe 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 swiched).

**NOTE: In mavlink dependent mode DRONE\_ID\_STANDALONE parameter should be configure to 0.**

2. Using Mission Planner software enable Mavlink V2 protocol on TELEM 1 port.
3. Connect Pixhawk to Mission Planner.

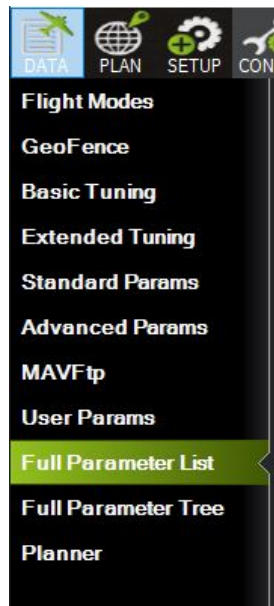


4. Select CONFIG tab.



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_RTSCSTS		0		0:Disabled 1:Enabled 2:Auto
Basic Tuning	BRD_SER2_RTSCSTS		0		0:Disabled 1:Enabled 2:Auto
Extended Tuning	RSSI_TYPE		0		0:Disabled 1:AnalogPin 2:RCChannelPwmV 3:ReceiverProtocol 4:PWMInputPin
Standard Params					
Advanced Params					
MAVFtp					
User Params	SERIAL1_BAUD		115		1:1200 2:2400 4:4800 9:9600 19:19200 39:38400 57:57600 111:111100 115:115200 256:256000 500:500000 921:321600 1500:1500000
Full Parameter List					
Full Parameter Tree					

7. Find SERIAL1\_PROTOCOL parameter and set a value to 2.

Flight Modes	Command	Δ	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](#)

### 8.2.1 Supported Mavlink

As a standard remote id idMe should be connected to the flight controller to obtain the necessary data. idMe provides heartbeat messages and can be used to read the device's status.

The following MAVlink messages are required:

- MAVLINK\_MSG\_ID\_VFR\_HUD (#74)
- MAVLINK\_MSG\_ID\_SYSTEM\_TIME (#2)
- MAVLINK\_MSG\_ID\_GLOBAL\_POSITION\_INT (#33)
- MAVLINK\_MSG\_ID\_SCALED\_PRESSURE (#29)
- MAVLINK\_MSG\_ID\_GPS\_RAW\_INT (#24)

### Heartbeat

The device delivers heartbeat messages about once a second, describing the status of the device.

HEARTBEAT ( #0 )	
<b>type</b>	MAV_TYPE_ODID
<b>system_status</b>	MAV_STATE_CALIBRATING
	MAV_STATE_ACTIVE
	MAV_STATE_EMERGENCY

**MAV\_STATE\_CALIBRATING** - idMe awaits on-board sensor calibration

**MAV\_STATE\_ACTIVE** - idMe is running, waiting for launch or the device is in the air

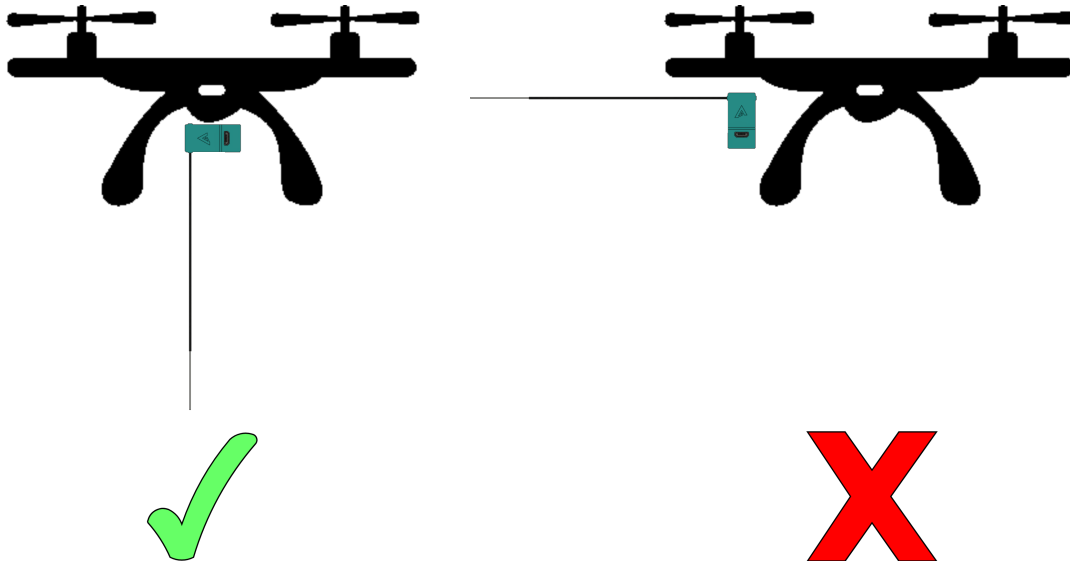
**MAV\_STATE\_EMERGENCY** - idMe has infernal error

## 9 General information

IdMe is an standard remote id device. This means need to be controller to flight controller to works properly. Using BLE 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.



### 9.1 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.

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.

### 9.2 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.

DEVICE STATUS	STATUS LED	POWER LED
boot	light	light
configuration	light	light
error or calibration	blink fast	light
connection with flight controller	blink slowly	light

### 9.2.1 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.

## 9.3 Troubleshooting

### 9.3.1 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

### 9.3.2 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

### 9.3.3 IdMe still blinking fast

Failed to establish connection with flight controller

- Check idMe baudrate is exaly the same like in flight controller
- Check jst-jst cable to ensure, all pins are connected
- Check flight controller telemetry port settings are correct

Please read carefully

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.

© 2022 Aerobits - All rights reserved