

## TR-1A

### 1 Watt ADS-B Transceiver

## OVERVIEW

TR-1A belongs to second generation of the smallest ADS-B transceivers on market and has been developed for civil and commercial Unmanned Aircraft Systems. The device operates on 1090 MHz and allows to receive and transmit ADS-B data with defined 0.25, 0.5 or 1 Watt output power. The transceiver does not require external devices to operate. It is equipped with a high quality multi-GNSS receiver and a pressure sensor. The aluminium housing and ESD protection guarantee high resistance of the device to work in difficult conditions. TR-1A opens the way to the implementation of the *Detect and Avoid* algorithms, supporting the integration of UAS into the airspace.

## WARNING

ICAO addresses are used to provide a unique identity normally allocated to an individual aircraft or registration.

**Please do not use random ICAO!**

Address becomes a part of the aircraft's Certificate of Registration and MUST be given by Civil Aviation Authority and registered in aircraft database.

## BASIC FEATURES

- Real-time aircraft tracking on 1090 MHz
- Patented FPGA-In-The-Loop<sup>TM</sup> technology with the capability of receiving thousands of frames per second
- Integrated GNSS source and pressure sensor
- 0.25, 0.5 or 1 Watt RF output power
- Implemented MAVLink and AERO<sup>TM</sup> protocol
- Low-power consumption and low weight design
- Simple plug&play integration
- Programming via AT commands
- Designed to meet MOPS DO-260B (except the output power)
- Dimension: 35.0 x 25.0 x 8.5 mm

# CONTENTS

1	TECHNICAL PARAMETERS	4
2	ELECTRICAL SPECIFICATION	5
2.1	PIN DEFINITION	5
2.2	LED INDICATORS	5
3	MECHANICAL SPECIFICATION	6
4	SPECIFICATION OF USED ANTENNA	7
5	PRINCIPLE OF OPERATION	8
5.1	STATES OF OPERATION	8
5.1.1	BOOTLOADER STATE	8
5.1.2	RUN STATE	8
5.1.3	CONFIGURATION STATE	8
5.2	TRANSITIONS BETWEEN STATES	8
5.2.1	BOOTLOADER TO RUN TRANSITION	8
5.2.2	RUN TO CONFIGURATION TRANSITION	9
5.2.3	CONFIGURATION TO RUN TRANSITION	9
5.2.4	CONFIGURATION TO BOOTLOADER TRANSITION	9
6	UART CONFIGURATION	10
7	SETTINGS	11
7.1	WRITE SETTINGS	11
7.2	READ SETTINGS	11
7.3	SETTINGS DESCRIPTION	11
7.4	ERRORS	11
7.5	COMMAND ENDINGS	12
7.6	UPPERCASE AND LOWERCASE	12
7.7	AVAILABLE SETTINGS	12
7.8	EXAMPLE	13
8	COMMANDS	14
8.1	COMMANDS IN BOOTLOADER AND CONFIGURATION STATE	14
8.1.1	AT+LOCK	14
8.1.2	AT+BOOT	14
8.2	COMMANDS IN CONFIGURATION STATE	14
8.2.1	AT+CONFIG	14
8.2.2	AT+SETTINGS?	14
8.2.3	AT+HELP	14
8.2.4	AT+SETTINGS_DEFAULT	15
8.2.5	AT+SERIAL_NUMBER	15
8.2.6	AT+FIRMWARE_VERSION	15
8.2.7	AT+REBOOT	15
8.2.8	AT+REBOOT_BOOTLOADER	15
8.3	COMMANDS IN RUN STATE	15
9	CSV PROTOCOL (AERO)	16
9.1	CRC	16
9.2	ADS-B AIRCRAFT MESSAGE	16
9.3	STATISTICS MESSAGE	18

10	RAW PROTOCOL	19
10.1	MODE-S RAW FRAMES . . . . .	19
10.2	MODE-AC RAW FRAMES . . . . .	19
11	MAVLink PROTOCOL	20
11.1	ADS-B AIRCRAFT MESSAGE . . . . .	20
12	ASTERIX PROTOCOL	21
13	GDL90 PROTOCOL	22
14	QUICK START WITH TR-1A	23
14.1	CONFIGURATION USING MICRO ADS-B SOFTWARE . . . . .	23
15	GENERAL INFORMATION	25
15.1	MODULE INSTALLATION . . . . .	25
15.2	AERO vs. MAVLink PROTOCOL . . . . .	25
16	REVISION HISTORY	26

## 1 TECHNICAL PARAMETERS

Parameter	Value
Frequency	1090 MHz
Input voltage	5 V
Current consumption	140 mA
Sensitivity ADS-B	-76 dBm
RF Output power	Configurable +30 dBm, +27 dBm or +24 dBm
ESD protection	All lines
MAVLink (baud)	115200 bps
AERO (baud)	115200 bps (AT commands)
Main connector	PXMBNI05RPM04APC
Antenna connector	2 x MMCX
Dimension	35.0 x 25.0 x 8.5 mm
Weight (without cables and antennas)	14 grams

Table 1: Technical parameters.

## 2 ELECTRICAL SPECIFICATION

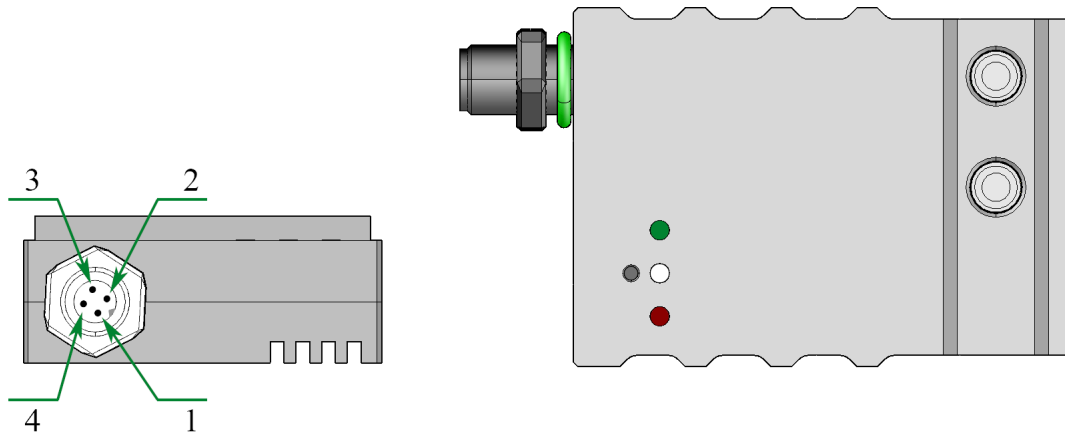


Figure 1: Module overview

### 2.1 PIN DEFINITION

Pin	Wire colour	Name	Function
1	Red	+5 V	Power supply (5 V/ 70 mA)
2	Green	TX	Data from device to host
3	White	RX	Data from host to device
4	Black	GND	Ground

Table 2: Electrical parameters

### 2.2 LED INDICATORS

LED	Function
Green	Power supply indicator
White	Frame detection / receive indicator
Red	ADS-B OUT indicator 1. OFF – Disabled 2. Blink – Wait for FIX 3. ON – Active

Table 3: Electrical parameters

### 3 MECHANICAL SPECIFICATION

All dimensions in mm (tolerances  $\pm 0.1$  mm)

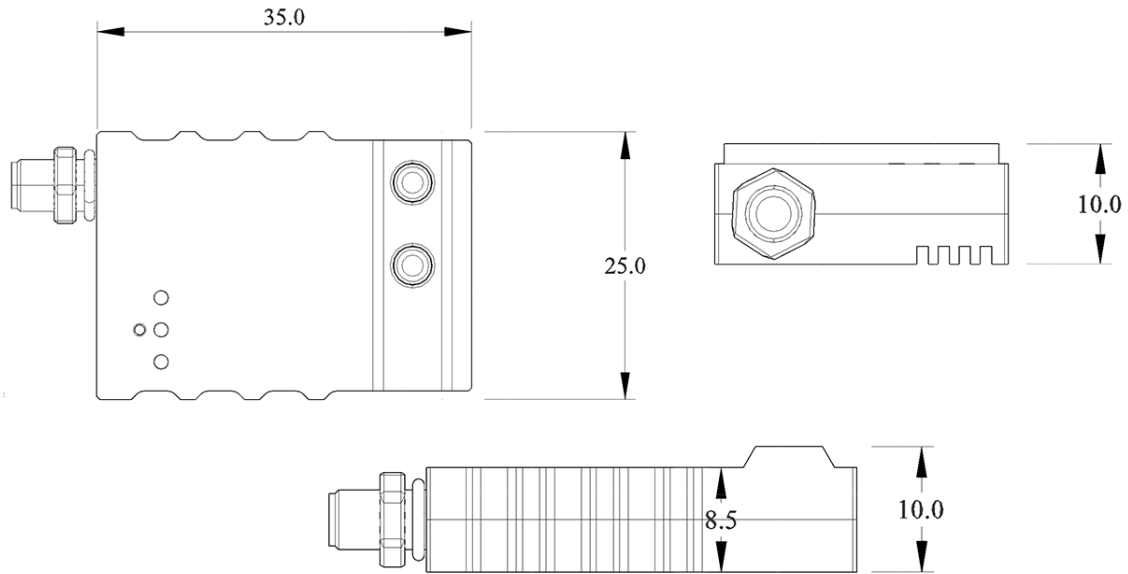


Figure 2: Mechanical drawing

Connector	Type	Example
Main	Installed on board	BULGIN, PXMBNI05RPM04APC
	Mating connector	BULGIN, PXPPVC05FBF04ACL010PVC
Antenna	Installed on board	MMCX, 73415-0961
	Mating connector	ASMK025X174S11

Table 4: Mechanical parameters

## 4 SPECIFICATION OF USED ANNTENA

No.	Part number	Connector type
1 - ADS-B	DELTA1A/X/SMAM/S/S/11	GSM/GPRS, 3G and Ism Stubby Antenna SMA Male
2 - GNSS	TG.08.0113	GPS/LTE Antenna SMA Male

Table 5: Description of commonly used anntena

You can also use **effective alternative** to commonly used anntenas. The following one is proper option to increase performance of your device.

Part number	Connector type
GSM-ANT822	GSM Antenna SMA Male

Table 6: Description of alternative anntena

## 5 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.

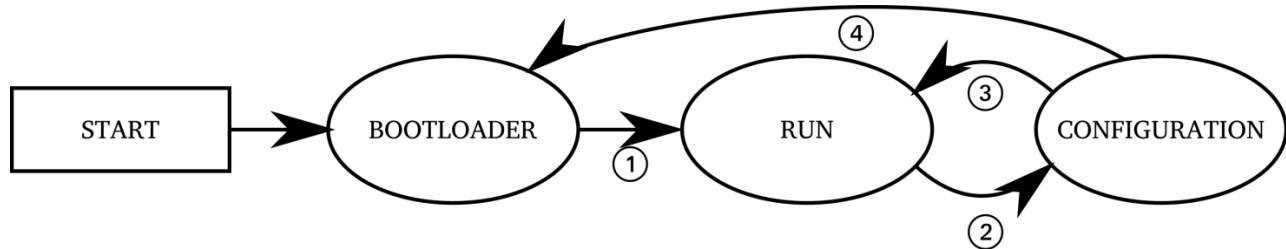


Figure 3: State machine of TT-TR1A module.

### 5.1 STATES OF OPERATION

#### 5.1.1 BOOTLOADER STATE

This is an initial state of TT-TR1A module 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 baudrate is constant and is set to 115200bps. After powering up module, it stays in this state for 1s. If any BOOTLOADER trigger is not present, module will transit to RUN state. Firmware upgrade is possible with use of AEROBITS\_updater.exe program. To acquire this program please contact: [support@aerobits.pl](mailto:support@aerobits.pl).

#### 5.1.2 RUN STATE

In this state module is working and receiving the data from aircraft. 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 baudrate.

#### 5.1.3 CONFIGURATION STATE

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

### 5.2 TRANSITIONS BETWEEN STATES

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

#### 5.2.1 BOOTLOADER TO RUN TRANSITION

BOOTLOADER state is semi-stable, that means the module requires additional action to stay in BOOTLOADER state. This transition will occur automatically if any of the required actions will not be taken. To stay in BOOTLOADER state (prevent this transition) one of following action must be processed:

- Pull BOOT/CONFIG pin low during start of module
- Send AT+LOCK=1 command while device is in BOOTLOADER state (always after power on for approximately 1s)



- Send `AT+REBOOT_BOOTLOADER` command in CONFIGURATION state. This will move to BOOTLOADER state and will lock module in this state.

If none of preceding conditions are meet, 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:

- Release or pull high BOOT/CONFIG pin
- If module is locked, send `AT+LOCK=0` command

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

### 5.2.2 RUN TO CONFIGURATION TRANSITION

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

- Pull BOOT/CONFIG pin low
- Send `AT+CONFIG=1` (using current baudrate). This method is not recommended, because module will support multiple protocols in future and Aerobits Sp. z o.o. cannot ensure that this command will be present in all protocols.

When module leaves RUN state it will send `AT+RUN_END` command and then `AT+CONFIG_START` command, on entering CONFIGURATION state. The former will be send using baudrate from settings, the latter will be send using 115200bps baudrate.

### 5.2.3 CONFIGURATION TO RUN TRANSITION

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

- Release or pull high BOOT/CONFIG pin
- Send `AT+CONFIG=0` command.

When module leaves CONFIGURATION state it will send `AT+CONFIG_END` command and then `AT+RUN_START` command, on entering RUN state. The former will be send using 115200bps baudrate, the latter will be send using baudrate from settings.

### 5.2.4 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`.

## 6 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
Baudrate	-	115200	-	bps
Stop Bits Number	-	1	-	-
Flow Control	-	None	-	-
Parity Bit	-	None	-	-

Table 7: UART settings.

## 7 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.

### 7.1 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

### 7.2 READ SETTINGS

AT+SETTING?

For example: AT+PROTOCOL?

Response: AT+PROTOCOL=1

### 7.3 SETTINGS DESCRIPTION

AT+SETTING=?

For example: AT+PROTOCOL=?

Response:

Setting: PROTOCOL

Description: Selected protocol: 0 NONE, 1 - RAW HEX, 2- CSV, 3 - MAVLINK,  
4 - ASTERIX, 5 - GLD90

Type: Integer decimal

Range (min.): 0

Range (max.): 5

Is preserved: 1

Is restart needed: 0

### 7.4 ERRORS

Errors are reported using following structure:

AT+ERROR (DESCRIPTION)

DESCRIPTION is optional and contains information about error.

## 7.5 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.

## 7.6 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.**

## 7.7 AVAILABLE SETTINGS

Setting	Min value	Max value	Default value	Comment
BAUDRATE	0	2	0	Baudrate in RUN state 0 - 115200bps 1 - 921600bps 2 - 3000000bps
GNSS_LOG	0	2	0	GNSS NMEA forwarding 0 - No forwarding 1 - RMC Messages only 2 - All
ICAO	-	-	0	ICAO number broadcasted by this device
SQUAWK	-	-	0	SQUAWK broadcasted by this device
EMITTER_CAT	-	-	0	Emitter category broadcasted by this device
ADSB_TX_ENABLED	0	1	1	ADSB broadcasting enabled
ADSB_TX_ON_BOOT	0	1	1	Device broadcasting ADSB on boot enabled
ADSB_TX_SURFACE	0	1	0	Device broadcasting ADSB when on surface enabled
ADSB_POWER	0	2	2	Trasmiting power of ADS-B 0 - 0.25W 1 - 0.5W 2 - 1W
PROTOCOL	0	6	2	Selected protocol. Not all values are valid for all devices. 0 - None 1 - RAW HEX 2 - CSV (AERO) 3 - MAVLink 4 - ASTERIX 5 - GDL90
SUBPROTOCOL	0	0	0	Reserved for future use

Table 8: Settings

## 7.8 EXAMPLE

As an example, to switch TR1A module 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+PROTOCOL=2\r\n
>> AT+OK\r\n
>> AT+OK\r\n
<< AT+CONFIG=0\r\n
```

## 8 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.

### 8.1 COMMANDS IN BOOTLOADER AND CONFIGURATION STATE

#### 8.1.1 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

#### 8.1.2 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

### 8.2 COMMANDS IN CONFIGURATION STATE

#### 8.2.1 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

#### 8.2.2 AT+SETTINGS?

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

AT+PROTOCOL=2

AT+SUBPROTOCOL=0

AT+BAUDRATE=0

#### 8.2.3 AT+HELP

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

## SETTINGS:

AT+PROTOCOL=2 [Selected protocol: 0 - NONE, 1 - RAW HEX, 2- CSV, 3 - MAVLINK,  
4 - ASTERIX, 5 - GLD90]

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]

## 8.2.4 AT+SETTINGS\_DEFAULT

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

## 8.2.5 AT+SERIAL\_NUMBER

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

Response:

AT+SERIAL\_NUMBER=0202041E43

## 8.2.6 AT+FIRMWARE\_VERSION

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

Response:

AT+FIRMWARE\_VERSION=10101017 (May 11 2018)

## 8.2.7 AT+REBOOT

AT+REBOOT - Restart module.

## 8.2.8 AT+REBOOT\_BOOTLOADER

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

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

## 8.3 COMMANDS IN RUN STATE

AT+CONFIG=1 - transition to CONFIGURATION state.

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

**NOTE: This command is deprecated and may not work in every future protocol. It is better to use BOOT/CONFIG pin to make module transit to CONFIGURATION state.**

## 9 CSV PROTOCOL (AERO)

CSV protocol is simple text protocol, that allows fast integration and analysis of tracked aircrafts. CSV messages start with '#' character and ends with "\r\n" characters. There are following types of messages:

1. ADS-B Aircraft message,
2. Statistics message.

**NOTE: In future versions, additional comma-separated fields may be introduced to any CSV protocol message, just before CRC field, which is guaranteed to be at the end of message. All prior fields are guaranteed to remain in same order.**

### 9.1 CRC

Each CSV message includes CRC value for consistency check. CRC value is calculated using standard CRC16 algorithm and its value is based on every character in frame starting from '#' to last comma ',' (excluding last comma). After calculation, value is appended to frame using hexadecimal coding. Example function for calculating CRC is shown below.

```
uint16_t crc16(const uint8_t* data_p, uint32_t length){
    uint8_t x;
    uint16_t crc = 0xFFFF;
    while (length--){
        x = crc>>8 ^ *data_p++;
        x ^= x>>4;
        crc = (crc<<8) ^ ((uint16_t)(x<<12)) ^ ((uint16_t)(x<<5)) ^ ((uint16_t)x);
    }
    return swap16(crc);
}
```

### 9.2 ADS-B AIRCRAFT MESSAGE

This message describes state vector of aircraft determined from ADS-B messages and is sent once per second. The message format is as follows:

```
#A: ICAO, FLAGS, CALL, SQ, LAT, LON, ALT_BARO, TRACK,
VELH, VELV, SIGS, SIGQ, FPS, NICNAC, ALT_GEO, CRC\r\n
```



#A	Aircraft message start indicator	Example value
ICAO	ICAO number of aircraft (3 bytes)	3C65AC
FLAGS	Flags bitfield, see table 10	1
CALL	Callsign of aircraft	N61ZP
SQ	SQUAWK of aircraft	7232
LAT	Latitude, in degrees	57.57634
LON	Longitude, in degrees	17.59554
ALT_BARO	Barometric altitude, in feet	5000
TRACK	Track of aircraft, in degrees [0,360)	35
VELH	Horizontal velocity of aircraft, in knots	464
VELV	Vertical velocity of aircraft, in ft/min	-1344
SIGS	Signal strength, in mV	840
SIGQ	Signal quality, in mV	72
FPS	Number of raw MODE-S frames received from aircraft during last second	5
NICNAC	NIC/NAC bitfield, see table 11	31B
ALT_GEO	Geometric altitude, in feet	5000
CRC	CRC16 (described in CRC section)	2D3E

Table 9: Descriptions of ADS-B message fields.

Value	Flag name	Description
0x0001	PLANE_ON_THE_GROUND	The aircraft is on the ground
0x0002	PLANE_IS_MILITARY	The aircraft is military object
0x0100	PLANE_UPDATE_ALTITUDE_BARO	During last second, barometric altitude of this aircraft was updated
0x0200	PLANE_UPDATE_POSITION	During last second, position (LAT & LON) of this aircraft was updated
0x0400	PLANE_UPDATE_TRACK	During last second, track of this aircraft was updated
0x0800	PLANE_UPDATE_VELO_H	During last second, horizontal velocity of this aircraft was updated
0x1000	PLANE_UPDATE_VELO_V	During last second, vertical velocity of this aircraft was updated
0x2000	PLANE_UPDATE_ALTITUDE_GEO	During last second, geometric altitude of this aircraft was updated

Table 10: ADS-B message Flags description.

The NIC/NAC bitfield is transmitted in big endian hexadecimal format without leading zeros. Table 11 describes its bitfield layout. The meaning of NIC/NAC indicators is exactly the same as described in ED-102A.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NAC <sub>p</sub>				NAC <sub>v</sub>			NIC <sub>baro</sub>	NIC			

Table 11: Structure of NIC/NAC bitfield in CSV protocol.

If data of any field of frame is not available, then it is transmitted as empty. For example:

```
#A:4CA948,300,,2122,52.99750,13.76526,37000,169,442,0,814,72,3,,6F1C\r\n
```

```
#A:424313,,2362,52.43431,14.84535,37000,65,456,0,806,61,0,,6843\r\n
```

**NOTE: SIGS and SIGQ fields are updated based on raw MODE-S frames. They are calculated from frames received in last second. If there were no receiver frames (FPS=0), those fields will not be updated.**

**NOTE: SIGS is measured based on analog RF signal. This signal has DC offset of about 700mV.**

### 9.3 STATISTICS MESSAGE

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

#S:CPU,RES,RES,FPSS,RES,RES,CRC

#S	Statistics message start indicator	Example
CPU	CPU load in %	12.1
RES	Reserved for future use	-
RES	Reserved for future use	-
FPSS	Number of MODE-S frames received in last second	3
RES	Reserved for future use	-
RES	Reserved for future use	-
CRC	CRC16 (described in CRC section)	2D3E

Table 12: Statistics message fields.

## 10 RAW PROTOCOL

This protocol is dedicated for raw Mode-A/C/S frames acquisition. In this special mode of operation, output frames are not processed, nor validated in any way. All processing, checksum validation, etc. must be done on user's side. All raw frames, regardless of type, start with '\*' and end with ';' ASCII characters, whereas their content is encoded in hexadecimal format, MSB first. At the end, extended fields are appended to frame.

\*RAW\_FRAME; (SIGS, SIGQ, RES) \r\n

Var.	Description	Example
SIGS	Signal strength in mV	840
SIGQ	Signal quality in mV	72
RES	Reserved for future use.	

Table 13: Extended messages description.

### 10.1 MODE-S RAW FRAMES

Short and long frames consist accordingly of 7 or 14 data bytes. Examples of raw MODE-S frames:

- Short frame: \*5D4B18FFFC710B; (899, 58,) \r\n
- Long frame: \*8D4CA7E858B9838206BA422BBD7B; (995, 164,) \r\n

### 10.2 MODE-AC RAW FRAMES

Each frame consists of 2 data bytes. Bits of first 3 nibbles represent data pulses of MODE-A/C frames in the following order:

C1, A1, C2, A2, C4, A4, B1, D1, B2, D2, B4, D4

The fourth nibble has least significant bit set, which indicates presence of SPI pulse:

- 0 - SPI pulse not present in frame
- 1 - SPI pulse present in frame

Examples of raw MODE-AC frames:

- \*6610; (979, 151,) \r\n
- \*E3E0; (995, 167,) \r\n

**NOTE: It is impossible to reliably distinguish between MODE-A and MODE-C frames based only on received signal on 1090MHz.**

## 11 MAVLINK PROTOCOL

TT-TR1A can be switched to use MAVLink protocol. This can be achieved by altering PROTOCOL setting. When MAVLink protocol is used, module is sending list of aircrafts every second. MAVLink messages have standardized format, which is well described on official protocol webpage ([mavlink.io/en/messages](http://mavlink.io/en/messages)).

### 11.1 ADS-B AIRCRAFT MESSAGE

Aircrafts are encoded using ADSB\_VEHICLE message ([mavlink.io/en/messages/common.html#ADSB\\_VEHICLE](http://mavlink.io/en/messages/common.html#ADSB_VEHICLE)). MAVLink message contains several data fields which are described below.

Field Name	Type	Description
ICAO_address	uint32_t	ICAO address
lat	int32_t	Latitude, expressed as degrees * 1E7
lon	int32_t	Longitude, expressed as degrees * 1E7
altitude_type	uint8_t	Type from ADSB_ALTITUDE_TYPE enum
altitude	int32_t	Barometric/Geometric Altitude (ASL), in millimeters
heading	uint16_t	Course over ground in centidegrees
hor_velocity	uint16_t	The horizontal velocity in centimeters/second
ver_velocity	uint16_t	The vertical velocity in centimeters/second, positive is up
callsign	char[9]	The callsign, 8 chars + NULL
emitter_type	uint8_t	Type from ADSB_EMITTER_TYPE enum
tslc	uint8_t	Time since last communication in seconds
flags	uint16_t	Flags to indicate various statuses including valid data fields
squawk	uint16_t	Squawk code

Table 14: MAVLink ADSB\_VEHICLE message description

The ADS-B vehicle may transmit barometric, as well as geometric altitude. The SUBPROTOCOL setting allows for toggling altitude transmit priority:

- When set to 0, altitude field will be filled with geometric altitude first. If not available, barometric altitude will be used.
- When set to 1, barometric altitude will be preferred.

## 12 ASTERIX PROTOCOL

TT-TR1A can be switched to use ASTERIX binary protocol. This can be achieved by altering PROTOCOL setting. When ASTERIX protocol is used, module is sending list of aircrafts every second. Aircrafts are encoded using I021 ver. 2.1 message. Also, once per second device sends I023 Ground Station Status frame as a heartbeat.

For further reference of reading ASTERIX frames see relevant documentation:

- I021 messages: [CAT021 - EUROCONTROL Specification for Surveillance Data Exchange Part 12: Category 21](#)
- I023 messages: [CAT023 - EUROCONTROL Specification for Surveillance Data Exchange Part 16: Category 23](#)

## 13 GDL90 PROTOCOL

TT-TR1A can be configured to use GDL90 binary protocol. This can be achieved by altering PROTOCOL setting. When GDL90 protocol is used, module is sending list of aircrafts every second. Aircrafts are encoded using Traffic Report (#20) message. Also, once per second device sends Heartbeat (#0), Ownship Report (#10) and Ownship Geometric Altitude (#11) messages.

For further reference of parsing GDL90 frames see relevant documentation: [GDL90 Data Interface Specification](#).

The ADS-B vehicle may transmit barometric, as well as geometric altitude. The SUBPROTOCOL setting allows for toggling Traffic Report altitude transmit priority:

- When set to 0, altitude field will be filled with geometric altitude first. If not available, barometric altitude will be used.
- When set to 1, barometric altitude will be preferred.

## 14 QUICK START WITH TR-1A

TR-1A is a stand-alone device and in the simplest case of its operation requires only a power supply. However during the first start-up, you must configure the device. That can be performed in the few steps described below. First install the antennas using the MCX-> SMA adapters included in the kit. Also connect the configuration cable that will help you set the device parameters. The following figure shows the installation method.

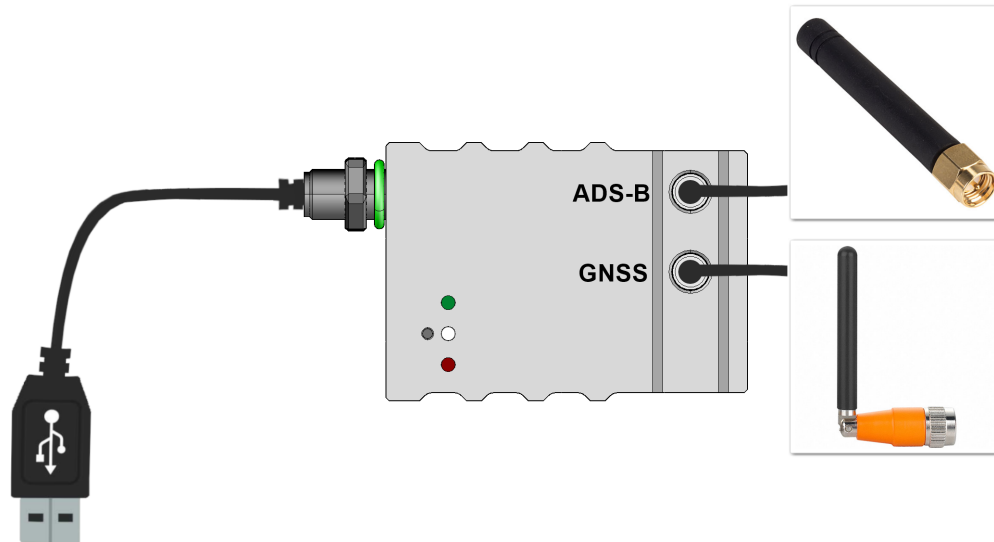


Figure 4: Combination overview

### 14.1 CONFIGURATION USING MICRO ADS-B SOFTWARE

1. Connect the device to the PC. The converter is supplied with the FTDI chip. In this case, the installation of the controller takes place automatically.
2. Download the latest Micro ADS-B software from [www.aerobits.pl](http://www.aerobits.pl). Install Micro ADS-B on your Windows computer. If the device is connected to a PC, it should be found automatically after clicking the "Connect" button. The connection window should look similar to the one in the picture. Select the device found and press "OK".

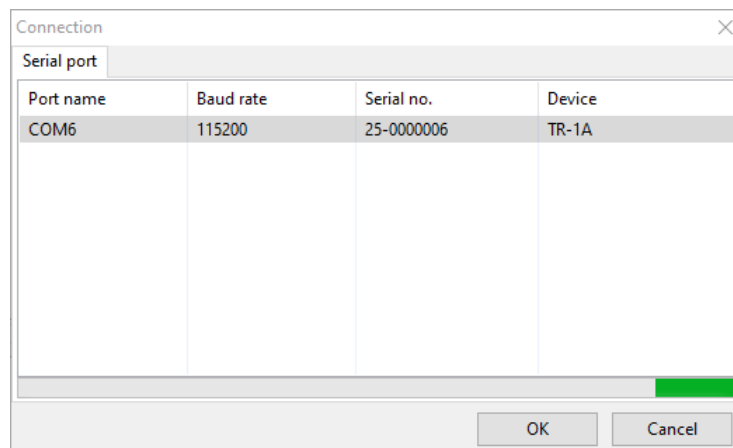


Figure 5: Port select window

3. Press "Settings" to enter the parameterization mode of the module. After setting the parameters, press the "Ok" button to save the settings. TR-1A is ready to work.

	Value	Description
<b>PROTOCOL</b>	2	Selected protocol: 0 - NONE, 1 - RAW HEX, 2- CSV, 3 - MAVLink, 4 - ASTERIX, 5 - GDL90
<b>SUBPROTOCOL</b>	0	Subprotocol of selected protocol
<b>GNSS_LOG</b>	0	GNSS printing - 0 - NONE, 1 - RMC message, 2 - all NMEA
<b>ICAO</b>	0	ICAO number broadcasted by this device
<b>SQUAWK</b>	0	SQUAWK broadcasted by this device
<b>EMITTER_CAT</b>	0	Emitter category broadcasted by this device
<b>IDENT</b>		Ident broadcasted by this device
<b>ADSB_TX_ENABLED</b>	1	Is ADSB broadcasting enabled?
<b>ADSB_TX_ON_BOOT</b>	1	Is this device broadcasting ADSB on boot?
<b>ADSB_TX_SURFACE</b>	0	Is this device broadcasting ADSB when on surface?
<b>ADSB_POWER</b>	2	Trasmiting power of ADS-B: 0-0.25W; 1-0.5W; 2-1W
<b>PRESSURE_LOG</b>	0	Log data from pressure sensor

Serial number: 25-0000006      Firmware version: 20010598(Jul 17 2020)

OK    Restore Defaults

Figure 6: Settings window



## **15 GENERAL INFORMATION**

### **15.1 MODULE INSTALLATION**

There is a high concentration of various electronic systems on a small area at UAS. Try to keep as much separation between TR-1A and other devices, especially radio ones. Despite the high robustness of TR-1A to jamming, try to install the antenna away from other on-board systems.

### **15.2 AERO vs. MAVLINK PROTOCOL**

TR-1A is based on OEM TT-SF1a module. The default is in AERO protocol mode, which is an ASCII protocol. If you want to use the module to work with MAVLink system, it is possible to switch the protocol to MAVLink, which has the binary representation. Details of the module programming can be found on the website.

## 16 REVISION HISTORY

Date	Revision	Changes
09-March-20	1	Initial release.
23-July-20	2	Update of mechanical parameters and new software version user manual.
22-October-20	3	Sections related to used protocols and description of settings added.

Table 15: Document revision history.

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.

© 2020 Aerobits - All rights reserved