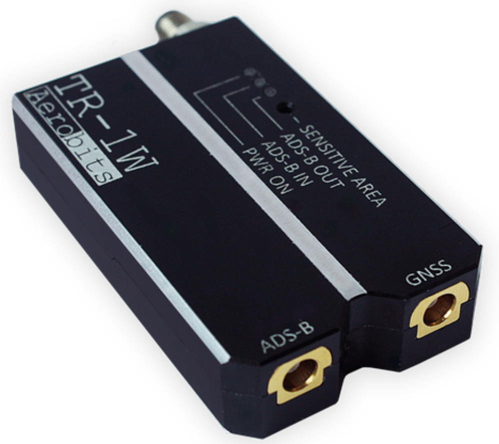


TR-1W

1 Watt ADS-B Transceiver

Overview

TR-1W belongs to the class 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 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-1W 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^{TM} technology with the capability of receiving thousands of frames per second
- Integrated GNSS source and pressure sensor
- 1 Watt RF output power
- Implemented MAVLink and AERO^{TM} 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: 45.5 x 28.0 x 10.0 mm

Contents

1 Technical parameters

Parameter	Value
Frequency	1090 MHz
Input voltage	5 V
Current consumption	130 mA
Sensitivity	-80 dBm
RF Output power	+30 dBm
ESD protection	All lines
MAVLink (baud)	115200 bps
AERO (baud)	115200 bps (AT commands)
Main connector	PXMBNI05RPM04APC
Antenna connector	2 x MCX
Dimension	45.5 x 28.0 x 10.0 mm
Weight (without cables and antennas)	30 grams

Table 1: Technical parameters.

2 Electrical specification

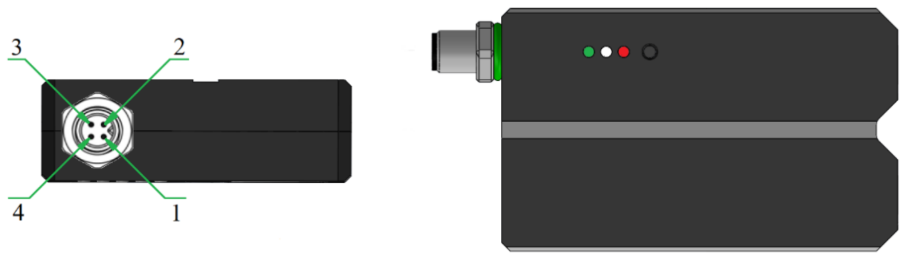


Figure 1: Module overview

2.1 Pin definition

Pin	Wire colour	Name	Function
1	Red	+5 V	Power supply (5 V/ 130 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 ±0.1 mm)

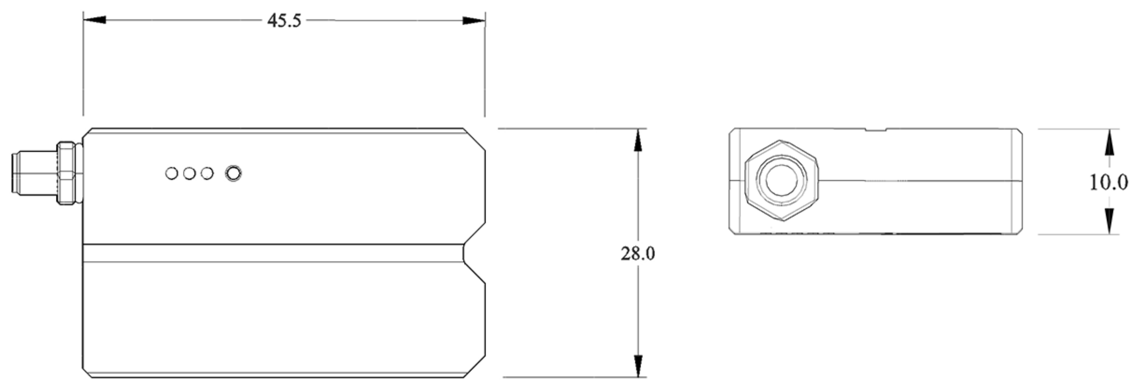


Figure 2: Mechanical drawing

Connector	Type	Example
Main	Installed on board	BULGIN, PXMBNI05RPM04APC
	Mating connector	BULGIN, PXPPVC05FBF04ACL010PVC
Antenna	Installed on board	MCX, 73415-1061
	Mating connector	MOLEX, 73366-0010

Table 4: Mechanical parameters

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.

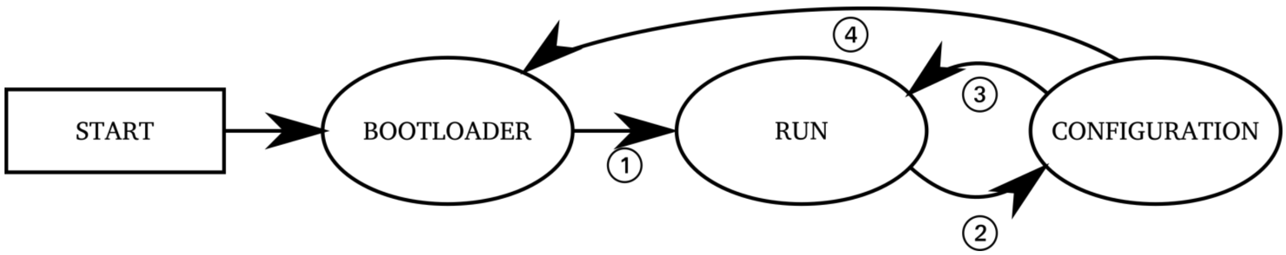


Figure 3: State machine of TR-1W module.

4.1 States of operation

4.1.1 BOOTLOADER state

This is an initial state of TR-1W 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 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.

4.1.2 RUN state

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 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.

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

The UART interface uses settings as described in table 5.

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

Table 5: 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 value	Max value	Default value	Comment
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
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 6: Settings

6.8 Example

As an example, to switch TR-1W 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
```

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=1
AT+SUBPROTOCOL=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:

- For older devices: `AT+SERIAL_NUMBER=0202041E43`
- For newer devices: `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 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.

8.1 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);
}
```

8.2 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,ECAT,CRC\r\n
```

#A	Aircraft message start indicator	Example value
ICAO	ICAO number of aircraft (3 bytes)	`3C65AC`
FLAGS	Flags bitfield, see table 8	`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 feets	`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 9 (v2.6.0+)	`31B`
ALT_GEO	Geometric altitude, in feets (v2.6.0+)	`5000`
ECAT	Emitter category, see table 10 (v2.7.0+)	`14`
CRC	CRC16 (described in CRC section)	`2D3E`

Table 7: 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 8: ADS-B message Flags description.

The NIC/NAC bitfield is transmitted in big endian hexadecimal format without leading zeros. Table 9 describes its bitfield layout. The meaning of NIC/NAC indicators is exactly the same as described in ED-102A.

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Reserved</td><td colspan="4">NAC_p</td><td colspan="3">NAC_v</td><td>NIC_{baro}</td><td colspan="4">NIC</td></tr>
</table>

Table 9: Structure of NIC/NAC bitfield in CSV protocol.

Below is a list of emitter category values returned in ECAT field.

ECAT value	Description
0	Unknown.
1	Light (below 15500 lbs.).
2	Small (15500 - 75000 lbs.).
3	Large (75000 - 300000 lbs.).
4	High-Vortex Large (aircraft such as B-757).
5	Heavy (above 300000 lbs.).
6	High performance (above 5g acceleration and above 400 knots).
7	Rotorcraft.
8	Reserved.
9	Glider, Sailplane.
10	Lighter-Than-Air.
11	Parachutist, Skydiver.
12	Ultralight, hang-glider, paraglider.
13	Reserved.
14	Unmanned Aerial Vehicle.
15	Space, Trans-atmospheric Vehicle.
16	Reserved.
17	Surface Vehicle - Emergency Vehicle.
18	Surface Vehicle - Service Vehicle.
19	Point Obstacle (includes Tethered Ballons).
20	Cluster obstacle.
21	Line obstacle.

Table 10: ADS-B emitter category values 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.

8.3 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 11: Statistics message fields.

9 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,TS)\r\n
```

Var.	Description	Example
SIGS	Signal strength in mV	`840`
SIGQ	Signal quality in mV	`72`
TS	Timestamp for multilateration. Time from last PPS pulse in hex format.	`20CB3`

Table 12: Extended messages description.

NOTE: To use multilateration, TS value must be calibrated using calibration value from statistics message.

NOTE: TS field is available when precise PPS signal from GNSS source is applied to module to 1PPS pin.

9.1 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;(899,58,20CB3)\r\n`
- Long frame: `*8D4CA7E858B9838206BA422BBD7B;(995,164,20CB3)\r\n`

9.2 MODE-AC RAW FRAMES

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

Starting with firmware 2.7.0, each frame is interpreted as squawk and formatted as 4 octal digits. They can also be read as binary frame with 4 hexadecimal digits, with bits being set as shown in table below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A4	A2	A1		B4	B2	B1		C4	C2	C1		D4	D2	D1

Table 13: Description of bits in raw Mode-A/C frames in new protocol version.

Examples of raw MODE-A/C frames using this format are as follows:

- `*0363;(979,151,20CB3)\r\n`
- `*7700;(995,167,20CB3)\r\n`

In firmwares prior to 2.7.0, each frame consists of 2 data bytes formatted as 4 hexadecimal digits. Individual bits directly correspond to pulses transmitted in Mode-A/C frame.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C1	A1	C2	A2	C4	A4	B1	D1	B2	D2	B4	D4				SPI

Table 14: Description of bits in raw Mode-A/C frames in old protocol version.

Examples of raw MODE-AC frames using this format are shown below.

- `*6611;(979,151,20CB3)\r\n`
- `*E3E0;(995,167,20CB3)\r\n`

10 MAVLink protocol

TR-1W 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 standarized format, which is well described on official protocol webpage (mavlink.io/en/messages).

10.1 ADS-B Aircraft message

Aircrafts are encoded using ADSB_VEHICLE message (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 15: 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 wil be preferred.

11 ASTERIX PROTOCOL

TR-1W 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 the device sends a heartbeat message using I023 ver. 1.2 format in Ground Station Status variant.

For further reference of parsing ASTERIX frames, please see relevant official 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

12 GDL90 PROTOCOL

TR-1W 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 wil be preferred.

13 Quick start with TR-1W

TR-1W 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.

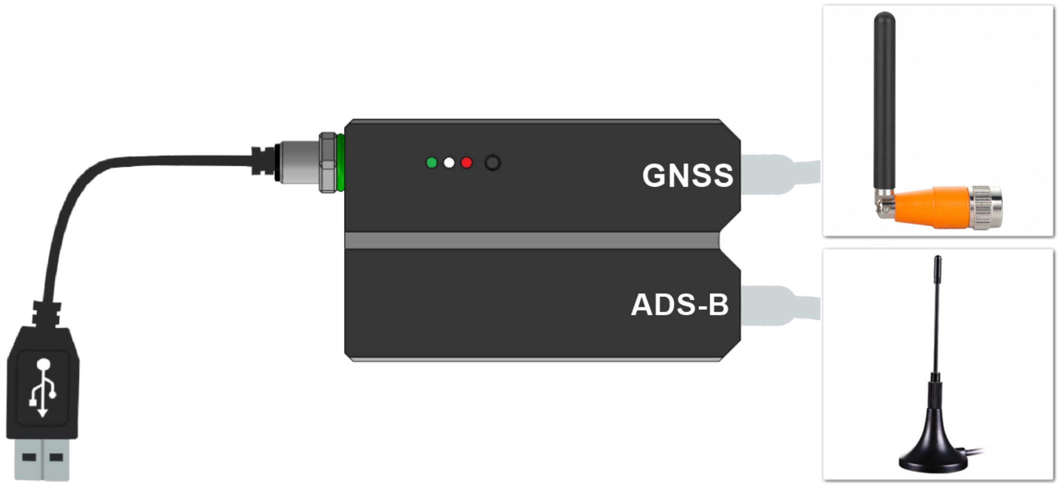


Figure 4: Combination overview

13.1 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. 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”.

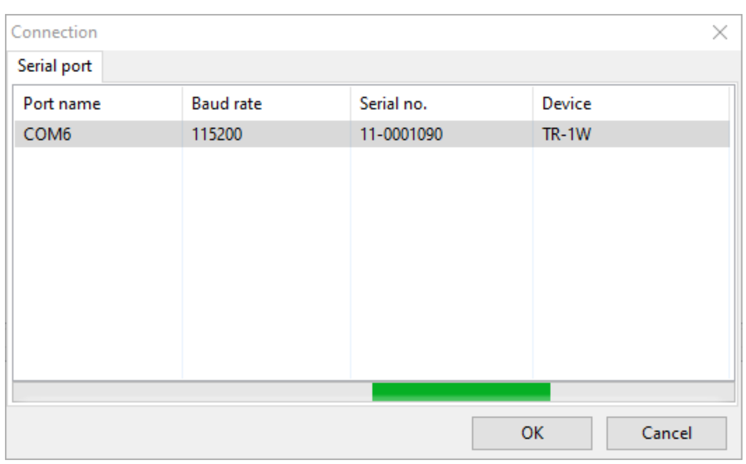
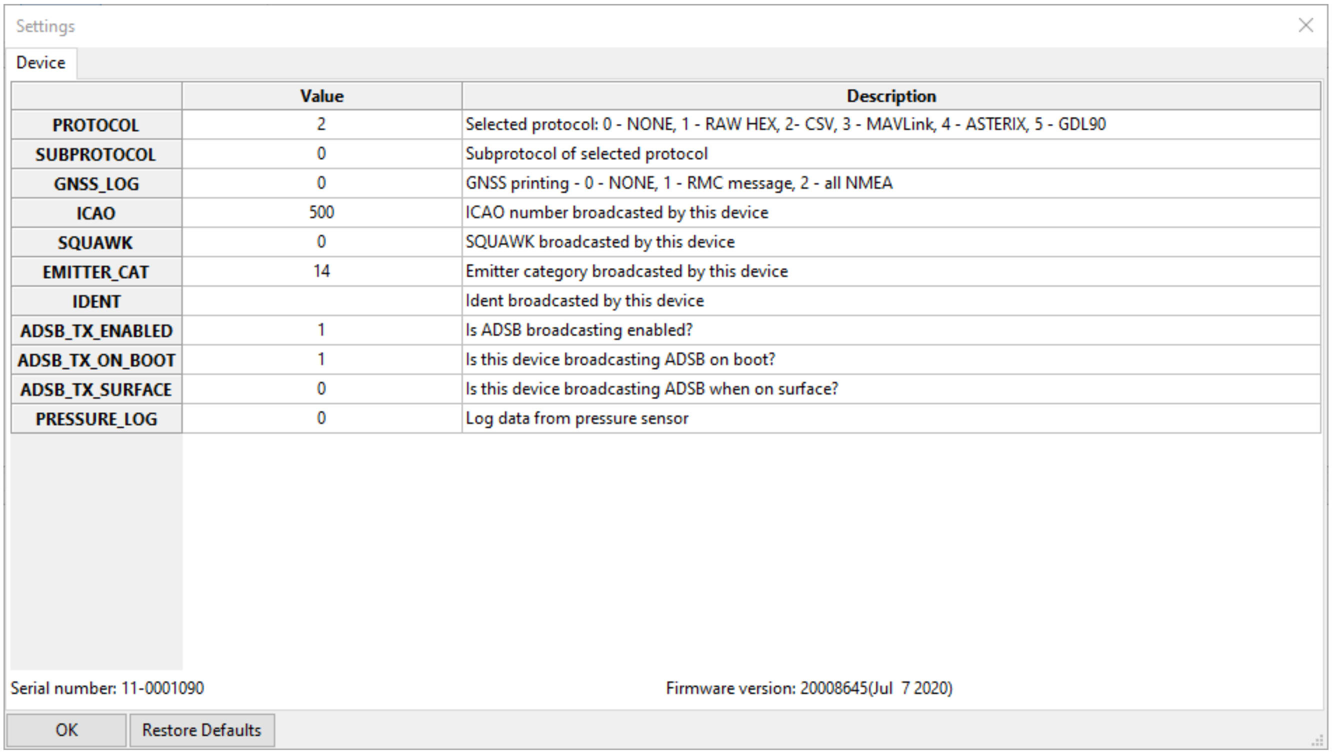


Figure 5: Port select window

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


Settings

Device

	Value	Description
PROTOCOL	2	Selected protocol: 0 - NONE, 1 - RAW HEX, 2- CSV, 3 - MAVLink, 4 - ASTERIX, 5 - GDL90
SUBPROTOCOL	0	Subprotocol of selected protocol
GNSS_LOG	0	GNSS printing - 0 - NONE, 1 - RMC message, 2 - all NMEA
ICAO	500	ICAO number broadcasted by this device
SQUAWK	0	SQUAWK broadcasted by this device
EMITTER_CAT	14	Emitter category broadcasted by this device
IDENT		Ident broadcasted by this device
ADSB_TX_ENABLED	1	Is ADSB broadcasting enabled?
ADSB_TX_ON_BOOT	1	Is this device broadcasting ADSB on boot?
ADSB_TX_SURFACE	0	Is this device broadcasting ADSB when on surface?
PRESSURE_LOG	0	Log data from pressure sensor

Serial number: 11-0001090 Firmware version: 20008645(Jul 7 2020)

OK Restore Defaults

Figure 6: Settings window

14 General information

14.1 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-1W and other devices, especially radio ones. Despite the high robustness of TR-1W to jamming, try to install the antenna away from other on-board systems.

14.2 AERO vs. MAVLink protocol

TR-1W is based on OEM TT-SC1a 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.

15 Revision history

Date	Revision	Changes
25-February-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 16: Document revision history.