TBR 700 RT - Protocol

Communication interfaces

TBR 700 RT: RS-485 (half duplex), Bluetooth and USB

TBR 700: Bluetooth and USB

Receiver communication protocol

TBR 700 and TBR 700 RT prints real time tag data and receiver sensor readings (temperature and background noise) to all active communication interfaces. For real time applications, we recommend using the RT-version with RS-485 for robust communication.

Receiver output format

Tag detections

When the receiver detects signals from acoustic transmitters the result is printed out in a standardized csv-format. One line for each successful received transmit:

```
$001234,1527073474,945,S256,241,150,32,69,2467
```

`$`	= Real time data
`001234`	= Receiver serial number (000001 -> 999999)
`1527073474`	= UTC UNIX timestamp (0000000000 -> 4294967295)
`945`	= Millisecond timestamp (0 -> 999)
`S256`	= Transmit protocol (R256, R04K, R64K, R01M, S256, S64K, HS256, DS256)
`241`	= Tag id number (1 -> 1048576)
`150`	= Tag raw data (",," blank if R256, R04K, R64K or R01M) (0 -> 65535 or blank)
`32`	= SNR (Signal to Noise Ratio) (0 ->63)
`69`	= Signal frequency [kHz] (63->77)
`2467`	= Receiver memory address (1->1506408, 1506409 = memory full)
`\r`	= end of line character

Transmit protocols

R256 (6 pings): - tag id = 1 -> 256
R04K (7 pings): - tag id = 1 -> 4096
R64K (8 pings): - tag id = 1 -> 65536
R01M (10 pings): - tag id = 1 -> 1048576 **- extra strong CRC**
S256 (8 pings): - tag id = 1 -> 256 & data = 0 -> 255 [8-bit data]
S64K (11 pings): - tag id = 1 -> 65536 & data = 0 -> 255 [8-bit data] **- extra strong CRC**
HS256 (11 pings): - tag id = 1 -> 256 & data = 0 -> 65535 [16-bit data] **- extra strong CRC**
DS256 (11 pings): - tag id = 1 -> 256 & data = 0 -> 65535 [8-MSB data1, 8-LSB data2] **- extra strong CRC**

$$SNR[dB] = 10 \cdot log_{10}\left(\frac{Average\ peak\ signal\ power\ in\ pulse\ train}{Average\ noise\ power}\right) = 20 \cdot log_{10}\left(\frac{Average\ peak\ signal\ amplitude\ in\ pulse\ train}{\boldsymbol{Average\ noise\ amplitude}}\right)$$

Typical values for SNR will be from 7 to 55 depending on signal strength and noise conditions. Where 7 is very week signals and 30 and above is strong signals. The scale is logarithmic.

TBR Sensor Readings

```
$001234,1527073500,TBR Sensor,323,10,16,69,2469
```

`$`	= Real time data
`001234`	= Receiver serial number (000001 -> 999999)
`1527073500`	= UTC UNIX timestamp (0000000000 -> 4294967295)
`TBR Sensor`	= receiver sensor readings (TBR 700 has a factory calibrated temperature chip inside the housing for water temperature documentation, and monitors background noise with the hydrophone) (fixed text)
`323`	= temperature ((data-50)/10 -> °C) (0->550)
`10`	= Average noise level (0-255)
`16`	= Peak noise level (0-255)
`69`	= Noise logging frequency [kHz] (63->77)
`2469`	= Receiver memory address (1->1506408, 1506409 = memory full)
`\r`	= end of line character

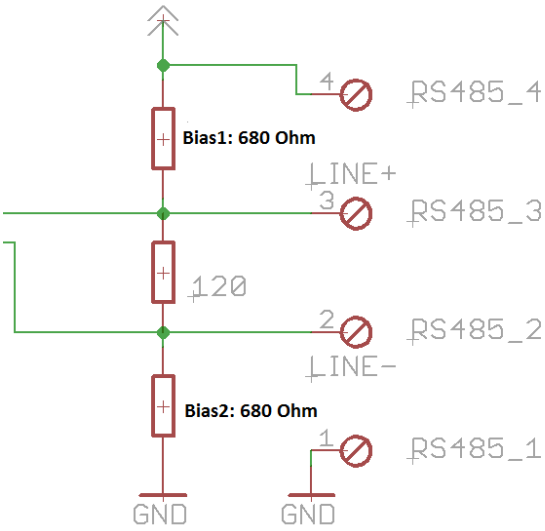
Commands

When TBR 700 is in listening mode (normal mode) it responds to the following commands:

1. Request: “?” -> The TBR responds with its serial number: “SN=001234 ><>” Can be used to test communication.
2. Clock sync. designed for very precise timestamps in positioning systems: (only supported by TBR 700 RT):
    a. When “(+)” is sent from connected sync. unit, the receivers’ UTC UNIX timestamp is rounded to closest 10. The internal real time clock is also reset so that the timestamp is incremented after exactly 500 ms. The time is therefore in practice synced to XXXXXXXXX0.5. The receiver responds with “ack01” if ok.
    b. “(+)123456789X” where X is Luhn’s verification number. The 9 first digits is the new UTC UNIX timestamp and it’s implied that the least significant digit is 0 (so the actual time given is 1234567890). If the verification number is correct, the timestamp is updated. When the time first is set, the clock should stay synchronized if “(+)” is given regularly. The receiver responds with “ack02” if ok. Note: when (+) is received the action of point a. takes place also here.

Bias Network – RS-485

To save power when TBR RT works as standalone, there is no bias network on the receiver side. To make sure the lines are in a defined state when no device transmits we recommend using bias resistors on connected devices. The bias resistors should be chosen so that the offset voltage between Line+ and Line- is >200 mV. This also makes less noise on the transitions between transmit and receive mode:



Connectors – RS-485

Wet connector on receiver:

BH-4-MP Connector, Seacon, WetCon, Bulkhead, Male

1. Gnd
2. RS485+
3. Vcc
4. RS485-

Dry side of the cable:

AMPHENOL TUCHEL C016 30D006 110 10 Circular Connector, Ecomate C016 Series, Pins/male, 6, Socket, Solder, Cable

Pinout:

1. Vcc (5-12VDC), can be left unconnected, the receiver will in that case use the internal battery.
2. Not in use
3. Not in use
4. RS485+
5. RS485-
6. Not in use
7. Gnd