**Teltonika Data Sending Protocols**

**Introduction**

A codec is a device or computer program for encoding or decoding a digital data stream or signal. Codec is a portmanteau of coder-decoder. A codec encodes a data stream or a signal for transmission and storage, possibly in encrypted form, and the decoder function reverses the encoding for playback or editing.   
  
Below you will see a table of all Codec types with IDs:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Codec 8** | **Codec 8 Extended** | **Codec 16** | **Codec 12** | **Codec 13** | **Codec 14** |
| 0x08 | 0x8E | 0x10 | 0x0C | 0x0D | 0x0E |

Also, there are using two data transport protocols: TCP and UDP. But it is not important which one will be used in Codec.

**Codec for device data sending**

In this chapter, you will find information about every Codec protocol which are used for device data sending and the differences between them.

**Codec 8**

* **Protocol Overview**

Codec8 – a main FM device protocol that is used for sending data to the server.

* **Codec 8 protocol sending over TCP**

TCP is a connection-oriented protocol that is used for communication between devices. The workings of this type of protocol is described below in the **communication with server** section.

* **AVL Data Packet**

The below table represents the AVL Data Packet structure:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **0x00000000 (Preamble)** | **Data Field Length** | **Codec ID** | **Number of Data 1** | **AVL Data** | **Number of Data 2** | **CRC-16** |
| 4 bytes | 4 bytes | 1 byte | 1 byte | X bytes | 1 byte | 4 bytes |

**Preamble** – the packet starts with four zero bytes.   
**Data Field Length** – size is calculated starting from Codec ID to Number of Data 2.   
**Codec ID** – in Codec8 it is always 0x08.   
**Number of Data 1** – a number that defines how many records are in the packet.   
**AVL Data** – actual data in the packet (more information below).   
**Number of Data 2** – a number that defines how many records are in the packet. This number must be the same as “Number of Data 1”.   
**CRC-16** – calculated from Codec ID to the Second Number of Data. CRC (Cyclic Redundancy Check) is an error-detecting code used to detect accidental changes to RAW data. For calculation we are using [CRC-16/IBM](https://wiki.teltonika-gps.com/view/Codec#CRC-16).  
  
**Note:** for [FMB640](https://wiki.teltonika-gps.com/view/FMB640), [FMB641](https://wiki.teltonika-gps.com/view/FMB641), [FMC640](https://wiki.teltonika-gps.com/view/FMC640), and [FMM640](https://wiki.teltonika-gps.com/view/FMM640), minimum AVL record size is 45 bytes (all IO elements disabled). The maximum AVL record size is 255 bytes. Maximum AVL packet size is 512 bytes. For other devices, the minimum AVL record size is 45 bytes (all IO elements disabled). Maximum AVL packet size is 1280 bytes.

* AVL Data

The below table represents the AVL Data structure.

|  |  |  |  |
| --- | --- | --- | --- |
| **Timestamp** | **Priority** | **GPS Element** | **IO Element** |
| 8 bytes | 1 byte | 15 bytes | X bytes |

**Timestamp** – a difference, in milliseconds, between the current time and midnight, January 1970 UTC (UNIX time).   
**Priority** – a field that defines AVL data priority (more information below).   
**GPS Element** – location information of the AVL data (more information below).   
**IO Element** – additional configurable information from the device (more information below).

* Priority

The below table represents Priority values. Packet priority depends on device configuration and records sent.

|  |  |
| --- | --- |
| **Priority** | |
| **0** | Low |
| **1** | High |
| **2** | Panic |

* GPS element

The below table represents the GPS Element structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Longitude** | **Latitude** | **Altitude** | **Angle** | **Satellites** | **Speed** |
| 4 bytes | 4 bytes | 2 bytes | 2 bytes | 1 byte | 2 bytes |

**Longitude** – east-west position.   
**Latitude** – north-south position.   
**Altitude** – meters above sea level.   
**Angle** – degrees from north pole.   
**Satellites** – number of satellites in use.   
**Speed** – speed calculated from satellites.   
  
**Note:** Speed will be 0x0000 if GPS data is invalid.   
  
Longitude and latitude are integer values built from degrees, minutes, seconds, and milliseconds by the formula:   
[](https://wiki.teltonika-gps.com/view/File:GPS.png)  
Where:   
d – Degrees; m – Minutes; s – Seconds; ms – Milliseconds; p – Precision (10000000)   
If the longitude is in the west or latitude in the south, multiply the result by –1.   
  
Note:   
To determine if the coordinate is negative, convert it to binary format and check the very first bit. If it is 0, the coordinate is positive. If it is 1, the coordinate is negative.   
  
Example:   
Received value: 20 9C CA 80 converted to BIN: 00100000 10011100 11001010 10000000 first bit is 0, which means coordinate is positive converted to DEC: 547146368. For more information see two‘s complement arithmetic.

* IO Element

|  |  |  |  |
| --- | --- | --- | --- |
| **Event IO ID** | 1 byte |  | **Event IO ID** – if data is acquired on the event – this field defines which IO property has changed and generated an event. For example, when if the Ignition state changes and it generates an event, the Event IO ID will be 0xEF (AVL ID: 239). If it’s not an eventual record – the value is 0.  **N** – a total number of properties coming with record (N = N1 + N2 + N4 + N8).  **N1** – number of properties, which length is 1 byte.  **N2** – number of properties, which length is 2 bytes.  **N4** – number of properties, which length is 4 bytes.  **N8** – number of properties, which length is 8 bytes.  **N’th IO ID** - AVL ID.  **N’th IO Value** - AVL ID value. |
| **N of Total IO** | 1 byte |
| **N1 of One Byte IO** | 1 byte |
| **1’st IO ID** | 1 byte |
| **1’st IO Value** | 1 byte |
| ... | |
| **N1’th IO ID** | 1 byte |
| **N1’th IO Value** | 1 byte |
| **N2 of Two Bytes** | 1 byte |
| **1’st IO ID** | 1 byte |
| **1’st IO Value** | 2 bytes |
| ... | |
| **N2’th IO ID** | 1 byte |
| **N2’th IO Value** | 2 bytes |
| **N4 of Four Bytes** | 1 byte |
| **1’st IO ID** | 1 byte |
| **1’st IO Value** | 4 bytes |
| ... | |
| **N4’th IO ID** | 1 byte |
| **N4’th IO Value** | 4 byte |
| **N8 of Eight Bytes** | 1 byte |
| **1’st IO ID** | 1 byte |
| **1’st IO Value** | 8 byte |
| ... | |
| **N8’IO ID** | 1 byte |
| **N8’IO Value** | 8 bytes |

* **Communication with server**

First, when the module connects to the server, the module sends its IMEI. First comes a short identifying the number of bytes written and then goes IMEI as text (bytes).   
For example, IMEI 356307042441013 would be sent as 000F333536333037303432343431303133.   
The first two bytes denote IMEI length. In this case 0x000F means, that IMEI is 15 bytes long.   
After receiving IMEI, the server should determine if it would accept data from this module. If yes, server will reply to module 01, if not - 00. Note that confirmation should be sent as a binary packet. I.e. 1 byte 0x01 or 0x00.   
Then the module starts to send the first AVL data packet. After the server receives a packet and parses it, the server must report to the module number of data received as an integer (four bytes).   
If the sent data number and the reported by the server don’t match module resends the sent data.

* Example:

The module connects to the server and sends IMEI:   
000F333536333037303432343431303133   
The server accepts the module:   
01   
The module sends data packet:

|  |  |  |
| --- | --- | --- |
| **AVL Data Packet Header** | **AVL Data Array** | **CRC-16** |
| Four Zero Bytes – 0x00000000,  “AVL Data Array” length – 0x000000FE | Codec ID – 0x08,  Number of Data – **0x02**  (Encoded using continuous bit stream. The last byte is padded to align to the byte boundary) | CRC of “AVL Data Array” |
| 00000000000000FE | 08**02**...(data elements)...**02** | 00008612 |

Server acknowledges data reception (2 data elements): **00000002**

* **Examples**

The hexadecimal stream of AVL Data Packet receiving and response in these examples are given in the hexadecimal form. The different fields of packets are separated into different table columns for better readability and some of them are converted to ASCII values for better understanding.

**1'st example**   
Receiving one data record with each element property (1 byte, 2 bytes, 4 bytes, and 8 bytes).   
  
Received data in the hexadecimal stream:   
000000000000003608010000016B40D8EA30010000000000000000000000000000000105021503010101425E0F01F10000601A014E0000000000000000010000C7CF   
  
Parsed:

|  |  |  |
| --- | --- | --- |
| **AVL Data Packet** | | |
| **AVL Data Packet Part** | | **HEX Code Part** |
|  | Zero Bytes | 00 00 00 00 |
| Data Field Length | 00 00 00 36 |
| Codec ID | 08 |
| Number of Data 1 (Records) | 01 |
| AVL Data | Timestamp | 00 00 01 6B 40 D8 EA 30 (GMT: Monday, June 10, 2019, 10:04:46 AM) |
| Priority | 01 |
| Longitude | 00 00 00 00 |
| Latitude | 00 00 00 00 |
| Altitude | 00 00 |
| Angle | 00 00 |
| Satellites | 00 |
| Speed | 00 00 |
| Event IO ID | 01 |
| N of Total ID | 05 |
| N1 of One Byte IO | 02 |
| 1’st IO ID | 15 (AVL ID: 21, Name: GSM Signal) |
| 1’st IO Value | 03 |
| 2’nd IO ID | 01 (AVL ID: 1, Name: DIN1) |
| 2’nd IO Value | 01 |
| N2 of Two Bytes IO | 01 |
| 1’st IO ID | 42 (AVL ID: 66, Name: External Voltage) |
| 1’st IO Value | 5E 0F |
| N4 of Four Bytes IO | 01 |
| 1’st IO ID | F1 (AVL ID: 241, Name: Active GSM Operator) |
| 1’st IO Value | 00 00 60 1A |
| N8 of Eight Bytes IO | 01 |
| 1’st IO ID | 4E (AVL ID: 78, Name: iButton) |
| 1’st IO Value | 00 00 00 00 00 00 00 00 |
|  | Number of Data 2 (Number of Total Records) | 01 |
| CRC-16 | 00 00 C7 CF |

Server response: 00000001

**2'nd example**   
Receiving one data record with one or two different element properties (1 byte, 2 bytes).   
  
Received data in the hexadecimal stream:   
000000000000002808010000016B40D9AD80010000000000000000000000000000000103021503010101425E100000010000F22A   
  
Parsed:

|  |  |  |
| --- | --- | --- |
| **AVL Data Packet** | | |
| **AVL Data Packet Part** | | **HEX Code Part** |
|  | Zero Bytes | 00 00 00 00 |
| Data Field Length | 00 00 00 28 |
| Codec ID | 08 |
| Number of Data 1 (Records) | 01 |
| AVL Data | Timestamp | 00 00 01 6B 40 D9 AD 80 (GMT: Monday, June 10, 2019, 10:05:36 AM) |
| Priority | 01 |
| Longitude | 00 00 00 00 |
| Latitude | 00 00 00 00 |
| Altitude | 00 00 |
| Angle | 00 00 |
| Satellites | 00 |
| Speed | 00 00 |
| Event IO ID | 01 |
| N of Total ID | 03 |
| N1 of One Byte IO | 02 |
| 1’st IO ID | 15 (AVL ID: 21, Name: GSM Signal) |
| 1’st IO Value | 03 |
| 2’nd IO ID | 01 (AVL ID: 1, Name: DIN1) |
| 2’nd IO Value | 01 |
| N2 of Two Bytes IO | 01 |
| 1’st IO ID | 42 (AVL ID: 66, Name: External Voltage) |
| 1’st IO Value | 5E 0F |
| N4 of Four Bytes IO | 00 |
| N8 of Eight Bytes IO | 00 |
|  | Number of Data 2 (Number of Total Records) | 01 |
| CRC-16 | 00 00 F2 2A |

Server response: 00000001

**3'rd example**   
Receiving two or more data records with one or more different element properties.   
  
Received data in the hexadecimal stream:   
000000000000004308020000016B40D57B480100000000000000000000000000000001010101000000000000016B40D5C198010000000000000000000000000000000 101010101000000020000252C   
  
Parsed:

|  |  |  |
| --- | --- | --- |
| **AVL Data Packet** | | |
| **AVL Data Packet Part** | | **HEX Code Part** |
|  | Zero Bytes | 00 00 00 00 |
| Data Field Length | 00 00 00 43 |
| Codec ID | 08 |
| Number of Data 1 (Records) | 02 |
| AVL Data  (1'st record) | Timestamp | 00 00 01 6B 40 D5 7B 48 (GMT: Monday, June 10, 2019, 10:01:01 AM) |
| Priority | 01 |
| Longitude | 00 00 00 00 |
| Latitude | 00 00 00 00 |
| Altitude | 00 00 |
| Angle | 00 00 |
| Satellites | 00 |
| Speed | 00 00 |
| Event IO ID | 01 |
| N of Total ID | 01 |
| N1 of One Byte IO | 01 |
| 1’st IO ID | 01 (AVL ID: 1, Name: DIN1) |
| 1’st IO Value | 00 |
| N2 of Two Bytes IO | 00 |
| N4 of Four Bytes IO | 00 |
| N8 of Eight Bytes IO | 00 |
| AVL Data  (2'nd record) | Timestamp | 00 00 01 6B 40 D5 C1 98 (GMT: Monday, June 10, 2019 10:01:19 AM) |
| Priority | 01 |
| Longitude | 00 00 00 00 |
| Latitude | 00 00 00 00 |
| Altitude | 00 00 |
| Angle | 00 00 |
| Satellites | 00 |
| Speed | 00 00 |
| Event IO ID | 01 |
| N of Total ID | 01 |
| N1 of One Byte IO | 01 |
| 1’st IO ID | 01 (AVL ID: 1, Name: DIN1) |
| 1’st IO Value | 01 |
| N2 of Two Bytes IO | 00 |
| N4 of Four Bytes IO | 00 |
| N8 of Eight Bytes IO | 00 |
|  | Number of Data 2 (Number of Total Records) | 02 |
| CRC-16 | 00 00 25 2C |

Server response: 00000002