PAYLOAD DESCRIPTION - GENERIC DE

1. VERSION HISTORY

Version	Date	Revision
1.0.0	06.01.2023	Created
1.0.1	13.04.2026	• Structure standardized • Content updated

TABLE OF CONTENTS

2. WHAT IS A PAYLOAD DESCRIPTION?

A **payload description** in LoRaWAN is a documented specification that describes how the payload of a LoRaWAN device is structured. It specifies what information is contained in the data packets, how it is encoded, and how it can be decoded.

2.1 CONTENTS OF A PAYLOAD DESCRIPTION

1. **Field structure**:
   - The payload description specifies how the data fields are organized within the payload.
   - Each field represents a specific piece of information, such as temperature, humidity, GPS coordinates, or battery level.

   Example:

   ```
   Byte 0-1: Temperature (in 1/100 °C, big-endian)
   Byte 2: Humidity (in %, integer)
   Byte 3: Battery level (in 0.1 V)
   ```

2. **Data encoding**:
   - Information is often stored in binary or hexadecimal format to save space.
   - It is specified whether the data is big-endian or little-endian (byte order).
   - The units of the values are specified (e.g., °C, %, volts).
3. **Data format**:
   - Specifies whether the data is encoded as an integer, float, string, or another type.
   - Some devices use scaled values that must be adjusted by multiplication or division (e.g., battery level in 0.1 V instead of directly in volts).
4. **Optional fields**:
   - Some payloads contain optional fields that are only sent under certain conditions. The description specifies when and how these fields appear.
5. **Sample data**:
   - A payload description often includes examples of an encoded data packet as well as the corresponding decoded values.

   Example of a binary payload:

   ```
   Hexadecimal: 0x07D0123A
   Description:
   - Bytes 0-1 (0x07D0): Temperature = 2000 / 100 = 20.00 °C
   - Byte 2 (0x12): Humidity = 18%
   - Byte 3 (0x3A): Battery level = 58 / 10 = 5.8 V
   ```

2.2 WHY IS A PAYLOAD DESCRIPTION IMPORTANT?

- **Decoder creation**: Developers use the description to write payload decoders that automatically interpret the raw data.

- **Interoperability**: When multiple systems or platforms communicate with the same device, the payload description helps ensure that the data is interpreted correctly.
- **Troubleshooting**: The description is essential for diagnosing problems when the data does not look as expected.

2.3 TYPICAL DOCUMENTATION

The payload description is usually provided by the manufacturer of the LoRaWAN device. It may be included in a user manual, a technical specification, or as a separate document.

An example of documentation:

```
Device: LoRa Sensor XY123
Payload Structure:
  Byte 0-1: Temperature (°C * 100)
  Byte 2: Humidity (%)
  Byte 3: Battery level (V * 10)
```

3. PAYLOAD DESCRIPTION SENTINUM

This chapter describes the structure of the telemetry data. The number of bytes in the payload depends on the sensor configuration. The data structure depends on which sensors are present in your product. In principle, every version has a header. This header contains information about the version and status of the sensor and also includes a master reading. This reading can simply be assumed to be the current reading. The header is followed by an additional payload section that contains further information, such as specific parameters for the measurement principle or settings for position and opening detection. Not every sensor has an additional payload section.

3.1 PAYLOAD EXAMPLE

The following payload example is provided for the Helios pressure sensor:

```
11 11 FE 1A D5 95 06 03 00 23 BE
```

Bytes	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
HEX	11	11	FE	1A	D5	95	06	03	00	23	BE
Description	Module key	Module Key	Uplink Counter	Battery Voltage	Battery Voltage	Temperature	Alarm Flag	Alarm Flag	Measurement Status	Pressure mbar	Pressure mbar

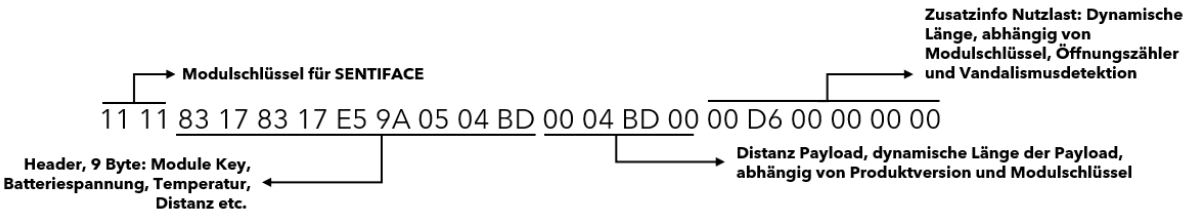
The module key is required for the downlink.

Terms	Description
Byte No.	Byte number starting at 1
Alias	Descriptive name of the variable
Description	Description of the variable
Label	Designation in the data converter

3.2 STRUCTURE OF THE MODULE KEY

Byte 1		Byte 2	
Bits 7–4	Bits 3–0	Bits 7–4	Bit 3 – 0
Module Base ID, e.g., Sentiface, Senticom, Sentivisor	Major Version (SW/HW Version	Minor Version (SW/HW Version)	Product Version (Sensors) (Sensors, e.g., TH, THL, ACC, …)

The module key of the SENTIFACE module can be extracted from the first 2 bytes of each uplink.
Uplink example:



4. PORTS

Ports	Channel	Description
0	Uplink	Join
1	Uplink	Telemetry Uplink
2	Uplink	Information Uplink
3	Uplink	Response to a downlink
4	Downlink	Port for downlinks to configure settings
5	Downlink	All commands marked with "EXEC", e.g., all commands in the Supervisor module, Reboot, Reset, Start Scan, etc.
6	Downlink	Downlink command for reading configurations
192	Uplink	GNSS scan data
197	Uplink	Wi-Fi SSID scan data

5. WHAT IS A PAYLOAD DECODER

A **LoRaWAN payload decoder** is a piece of software or code used to decode and interpret the raw data transmitted by a LoRaWAN device into a comprehensible and useful format.

Background:

- **LoRaWAN devices** typically transmit data in compressed and binary form to save bandwidth and energy.
- The transmitted "payloads" are often cryptic and contain raw data that must be decoded to extract the actual measured values (e.g., temperature, humidity, GPS coordinates).

How a Payload Decoder Works:

1. **Input data**:
   - A gateway receives the LoRaWAN data packets and forwards them to a network server (e.g., The Things Network - TTN).
   - This data is typically binary-encoded and may contain additional metadata.
2. **Decoder logic:**
   - The payload decoder is used to interpret the data according to the structure of the specific device or sensor.
   - This often includes:
     - Unpacking the binary data.
     - Converting the data into understandable units (e.g., Celsius, meters, volts).
     - Possibly also decrypting the data if it is encrypted.
3. **Output data:**
   - The decoder returns the decoded data in a readable format, e.g., as a JSON object.
   - Example:

```
json
CopyEdit
{
  "temperature": 22.5,
  "humidity": 60,
  "battery": 3.7
}
```

Applications:

- **Network servers**: Payload decoders can be implemented directly on servers such as TTN.
- **Data processing**: The decoded data is often stored and visualized in an IoT platform or database.
- **Debugging**: Developers use decoders to verify that their devices are sending the correct data.

Example of a payload decoder (e.g., in JavaScript for TTN):

```
JavaScript

function decodeUplink(input) {
  const bytes = input.bytes;
  return {
    data: {
      temperature: (bytes[0] << 8 | bytes[1]) / 100,
      humidity: bytes[2],
      battery: bytes[3] / 10
    }
  };
}
```

6. WHICH PAYLOAD DECODERS DOES SENTINUM PROVIDE?

Sentinum provides payload decoders for LoRaWAN in .js for TTN and for chirpstack.