LoRaWAN® BK-G Pulse Reader CM3061 Payload Structures

This document contains only payload structure definitions for firmware versions 2.3.x, the rest of the information can be found in CM3061 Datasheet .

Table of Contents

1 LoRaWAN Payloads

1.1 LoRaWAN Timed Payload

1.1.1 *usage_packet* and *usage_with_status_packet*

usage_packet and *usage_with_status_packet* are the only regular payloads, containing everything needed for billing purposes. They are essentially the same packet whereas *usage_with_status_packet* contains extra block (with *battery_*, *temperature_*, *radio_* and *meter_serial* parameters). Inclusion of this extra block depends on *device_status_sent* bit.

Example A *usage_packet* (unconfirmed packet, without device status, sent frequently) from fPort 25:
04 C0 00 14 43 3F254E00 01

Example B *usage_with_status_packet* (confirmed packet, with device status, sent at midnight and noon) from fPort 25:
04 C0 00 84 43 3F254E00 752B 03 6B D1 16 4A 33 7C 44983938

<table>
<thead>
<tr><th>Byte</th><th>Type</th><th>Example</th><th>Parameter</th><th>Details</th><th>Example Value</th></tr>
</thead>
<tbody>
<tr><td>0</td><td>uint8</td><td>0x04</td><td>*_packet_type*</td><td>0x04 - *usage_packet*</td><td>*usage_packet*</td></tr>
<tr><td>1</td><td>bits8</td><td>0xC0</td><td>*active_alerts*</td><td colspan="2">
<table>
<tr><th>Bit</th><th>Parameter</th><th>Type</th><th>Example</th></tr>
<tr><td>0-2</td><td>RFU</td><td></td><td>b000</td></tr>
<tr><td>3</td><td>*tamper_pending*</td><td>pending flag</td><td>0</td></tr>
<tr><td>4</td><td>RFU</td><td></td><td>0</td></tr>
<tr><td>5</td><td>*battery*</td><td>instant value</td><td>0</td></tr>
<tr><td>6</td><td>*no_usage*</td><td>instant value</td><td>1</td></tr>
<tr><td>7</td><td>*any_alert_active*</td><td>instant value</td><td>1</td></tr>
</table>
example: [*no_usage*]
example: [*backflow_pending*, *no_usage*]
Pending flag means that alert state is kept pending until next confirmed *usage_packet*. The pending flags can remain pending for hours even though alert's instant value is already off. *any_alert_active_now* indicates if instant value of any pending alert is on.
</td></tr>
<tr><td>2</td><td>uint8</td><td>0x00</td><td>RFU</td><td></td><td></td></tr>
<tr><td>3</td><td>bits8</td><td>0x14 /
0x94</td><td>*[meter]*</td><td colspan="2">
<table>
<tr><th>Bit</th><th>Parameter</th><th>Value</th><th>Example Value A / B</th></tr>
<tr><td>0-2</td><td>*meter_multiplier*</td><td>3 - 0.001 (default)
4 - 0.01
5 - 0.1
6 - 1
7 - 10</td><td>b100 = 0x04 = 4
meter_multiplier = 0.01</td></tr>
<tr><td>3-4</td><td>*meter_medium*</td><td>2 - *gas*</td><td>b10 - *gas*</td></tr>
<tr><td>5-6</td><td>*meter_unit*</td><td>2 - ft3</td><td>b00 -</td></tr>
<tr><td>7</td><td>*privacy_mode_active*</td><td></td><td>b0 / b1</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>uint8</td><td>0x43</td><td>*meter_actuality_duration__minutes*
meter_actuality_duration_formatted
(for *volume)*</td><td>0-59 - *minutes_ago* (if <60)
64-155 - *quarter_hours_ago* (1 h - 23 h 45 min) (if <156)
157-198 - *days_ago* (1 day - 42 days) (if <200)
255 - *n/a*
examples: 53 → 53 (minutes), 67 → 67-60 = 7 * 15 (min) = 1:45, 161 → 161 - 156 = 5 (days)</td><td>0x43 = 67
67 - 60 * 15 (min) =
7 * 15 (min) =
105 (min) = 1:45</td></tr>
<tr><td>5</td><td rowspan="4">uint32</td><td>0x3F</td><td rowspan="4">*meter_accumulated_volume__m3*
meter_accumulated_volume__gal
meter_accumulated_volume__ft3</td><td rowspan="4">convert: **meter_multiplier*
0x7FFFFFFF - *not_available*</td><td rowspan="4">0x004E253F = 5121343
5121343 * *meter_multiplier* =
5121343 * 0.01 =
51213.43 (m³)</td></tr>
<tr><td>6</td><td>0x25</td></tr>
<tr><td>7</td><td>0x4E</td></tr>
<tr><td>8</td><td>0x00</td></tr>
<tr><td>r0</td><td rowspan="2">bits16</td><td>0x75</td><td rowspan="2">*meter_readout_date*
(key-date for *meter_accumulated_volume__L)*</td><td rowspan="2" colspan="2">when: *metering_info.privacy_mode_active*
converting: identical to M-Bus Type E (Compound CP16)
<table>
<tr><th>Bit</th><th>Parameter</th><th>Value</th><th>Example B</th></tr>
<tr><td>0-4</td><td>*day*</td><td>1-31</td><td>b10101 = 0x15 = 21</td></tr>
<tr><td>5-7</td><td>*year_lower*</td><td></td><td>b011 = 0x03 = 3</td></tr>
<tr><td>8-11</td><td>*month*</td><td>1-12</td><td>b1011 = 0x0B = 11</td></tr>
<tr><td>12-15</td><td>*year_upper*</td><td></td><td>b0010 = 0x02 = 2</td></tr>
</table>
year = *year_lower* + 8 * *year_upper* = 3 + 8 * 2 = 3 + 16 = 19
example: 0x75 0x2B → 2019-11-21
</td></tr>
<tr><td>r1</td><td>0x2B</td></tr>
</tbody>
</table>

<table>
<tr><td></td><td>r0</td><td>bits8</td><td>0x01 /
0x03</td><td>*[general_parameters]*</td><td colspan="2">
<table>
<tr><th>Bit</th><th>Parameter</th><th>Value</th><th>Example A / B</th></tr>
<tr><td>0</td><td>*app_connected_within_a_day*</td><td>0 - *false,*
1 - *true*</td><td>1 / 1</td></tr>
<tr><td>1</td><td>*device_status_sent*</td><td>0 - *not_sent*
1 - *sent*</td><td>0 / 1</td></tr>
<tr><td>3-5</td><td>RFU *redundant_count*</td><td>0 - not_sent
1 - 3</td><td>0b000</td></tr>
<tr><td>6-7</td><td>RFU *redundant_interval*</td><td>0 - 15_min
1 - 1_h
2 - 6_h
3 - 24_h</td><td>0b00</td></tr>
</table>
</td></tr>
<tr><td rowspan="9"></td><td>r0</td><td>uint8</td><td>0x6B</td><td>*battery_remaining__years*</td><td>when: *general_parameters.device_status_sent*
convert: /12.0</td><td>0x6B = 107 (months) ≈ 8,9 (years)</td></tr>
<tr><td>r1</td><td>uint8</td><td>0xD1</td><td>battery_voltage__V</td><td>when: *general_parameters.device_status_sent*
convert: (/100) + 1.5</td><td>0xD1 = 209
209 / 100 + 1,5 = 3,59 (V)</td></tr>
<tr><td>r2</td><td>int8</td><td>0x16</td><td>*temperature__C*</td><td>when: *general_parameters.device_status_sent*
formatter: %s%s</td><td>0x16 = 22</td></tr>
<tr><td>r3</td><td>bits8</td><td>0x4A</td><td>*[temperature]*</td><td colspan="2">when: *general_parameters.device_status_sent*
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example Value</th></tr>
<tr><td>0-3</td><td>*temperature_min__C* (min_offset)
convert: *-2 + *temperature__C*</td><td>0xA = 10
10 * -2 + *temperature__C = 10 * -2 + 22 = -20 + 22 = 2*</td></tr>
<tr><td>4-7</td><td>*temperature_max__C* (max_offset)
convert: *2 + *temperature__C*</td><td>0x4 = 4
4 * 2 + *temperature__C = 4 * 2 + 22 = 8 + 22 = 30*</td></tr>
</table>
note: extremes are reset with each *status_packet.*</td></tr>
<tr><td>r4</td><td>uint8</td><td>0x33</td><td>*radio_downlink_rssi__dBm*</td><td>when: *general_parameters.device_status_sent*
convert: *-1</td><td>0x33 = 51
51 * -1 = -51 (dBm)</td></tr>
<tr><td>r5</td><td>bits8</td><td>0x7C</td><td>*[radio]*</td><td colspan="2">when: *general_parameters.device_status_sent*
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example Value</th></tr>
<tr><td>0-3</td><td>*radio_downlink_snr__dB* (signal to noise ratio)
convert: *2 - 20</td><td>0xC = 12
12 * 2 -20 = 24 - 20 = 4 (dB)</td></tr>
<tr><td>4-7</td><td>*radio_uplink_power__dBm*
convert: *2</td><td>0x7 = 7
7 * 2 = 14 (dBm)</td></tr>
</table>
example: 0x27 → *radio_downlink_snr* = -6 dB and *radio_uplink_pwr* = 4 dBm</td></tr>
<tr><td>r6</td><td rowspan="4">uint32</td><td>0x44</td><td rowspan="4">*meter_serial*
(sn of meter)</td><td rowspan="4">when: *general_parameters.device_status_sent*
format: hex (BCD - binary coded decimal)
0xFFFFFFFF - *not_available*</td><td rowspan="4">0x44 0x98 0x39 0x38 → 38399844</td></tr>
<tr><td>r7</td><td>0x98</td></tr>
<tr><td>r8</td><td>0x39</td></tr>
<tr><td>r9</td><td>0x38</td></tr>
</table>

1.2 LoRaWAN Configuration Payloads

1.2.1 *general_configuration_packet*

Example payload
20 1F0C 17 02 44983938 14 DEBB2A1300000000 0700 01

<table>
<tr><th>Byte</th><th>Type</th><th>Example</th><th>Parameter</th><th>Details</th><th>Example Value</th></tr>
<tr><td>0</td><td>uint8</td><td>0x20</td><td>*_packet_type*</td><td>0x20 - *general_configuration_packet*</td><td>*general_configuration_packet*</td></tr>
<tr><td>1</td><td>bits8</td><td>0x1F</td><td>*configured_parameters*</td><td colspan="2">
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0</td><td>*radio_lorawan_profile_sent*</td><td>1</td></tr>
<tr><td>1</td><td>*radio_wmbus_profile_sent*</td><td>1</td></tr>
<tr><td>2</td><td>*meter_serial_sent*</td><td>1</td></tr>
<tr><td>3</td><td>*meter_unit_medium_multiplier_sent*</td><td>1</td></tr>
<tr><td>4</td><td>*meter_accumulated_volume_sent**</td><td>1</td></tr>
<tr><td>5</td><td>*meter_accumulated_volume_offset_sent**</td><td>0</td></tr>
<tr><td>6-7</td><td>RFU</td><td>b00</td></tr>
</table>
*These can not be configured together, *meter_accumulated_volume_offset__L* will be ignored
</td></tr>
<tr><td>2</td><td>bits8</td><td>0x0C</td><td>*configured_parameters*</td><td colspan="2">
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0-1</td><td>RFU</td><td>b00</td></tr>
<tr><td>2</td><td>*alert_no_usage_sent*</td><td>1</td></tr>
<tr><td>3</td><td>*alert_tamper_sent*</td><td>1</td></tr>
<tr><td>4-7</td><td>RFU</td><td>b0000</td></tr>
</table>
</td></tr>
<tr><td>r0</td><td>uint8</td><td>0x17</td><td>*radio_lorawan_profile*</td><td>when: *configured_parameters.radio_lorawan_profile_sent*

0x00 - *lorawan_disabled*
0x01 - *lorawan_24_h_privacy*
0x02 - *lorawan_24_h*
0x03 - *lorawan_12_h*
0x07 - *lorawan_1_h_static*
0x08 - *lorawan_15_min_static*
0x17 - *lorawan_1_h_dynamic*
0x18 - *lorawan_15_min_dynamic*</td><td>*lorawan_1_h_dynamic*</td></tr>
<tr><td>r0</td><td>uint8</td><td>0x02</td><td>*radio_wmbus_profile*</td><td>when: *configured_parameters.radio_wmbus_profile_sent*

0x00 - *wmbus_disabled*
0x01 - *wmbus_privacy*
0x02 - *wmbus_driveby*
0x03 - *wmbus_fixnet*</td><td>*wmbus_driveby*</td></tr>
<tr><td>r0</td><td rowspan="4">uint32</td><td>0x44</td><td rowspan="4">*meter_serial*

(sn of meter)</td><td rowspan="4">when: *configured_parameters.meter_serial_sent*
type: hex

0xFFFFFFFF - *not_set*</td><td rowspan="4">0x44 0x98 0x39 0x38 → 0x38399844 → 38399844</td></tr>
<tr><td>r1</td><td>0x98</td></tr>
<tr><td>r2</td><td>0x39</td></tr>
<tr><td>r3</td><td>0x38</td></tr>
<tr><td>r0</td><td>uint8</td><td>0x14</td><td>*[meter_mode]*</td><td colspan="2">when: *configured_parameters.meter_unit_medium_multiplier_sent*
<table>
<tr><th>Bit</th><th>Parameter</th><th>Value</th><th>Example</th></tr>
<tr><td>0-2</td><td>*meter_multiplier*</td><td>3 - 0.001 (default)
4 - 0.01
5 - 0.1
6 - 1
7 - 10</td><td>b100 = 0x04 = 4
meter_multiplier = 0.01</td></tr>
<tr><td>3-4</td><td>*meter_medium*</td><td>2 - *gas*</td><td>b10 - *gas*</td></tr>
<tr><td>5-6</td><td>*meter_unit*</td><td>2 - ft3 (ft^3)</td><td>b00 -</td></tr>
<tr><td>7</td><td>RFU</td><td></td><td>0</td></tr>
</table>
</td></tr>
<tr><td>r0</td><td rowspan="2">uint64</td><td>0xDE</td><td rowspan="2">*meter_accumulated_volume__m3*
meter_accumulated_volume__gal
meter_accumulated_volume__ft3</td><td rowspan="2">when: *configured_parameters.meter_accumulated_volume_sent*</td><td rowspan="2">0x00000000132ABBDE = 321567710

321567710 / 1000 = 321567.710(m^3)</td></tr>
<tr><td>r1</td><td>0xBB</td></tr>
</table>

r2		0x2A	(absolute reading)	note: last decimal places will be floored with larger *meter_multiplier* convert: /1000.0	
r3		0x13			
r4		0x00			
r5		0x00			
r6		0x00			
r7		0x00			
r0	int32	0xBE	*meter_accumulated_volume_offset__m3* *meter_accumulated_volume_offset__gal* *meter_accumulated_volume_offset__ft3* (relative reading)	when: *configured_parameters. meter_accumulated_volume_offset_sent* note: last decimal places will be floored with larger *meter_multiplier* convert: /1000.0	0xFFFFE7BE = -6210 -6210 / 1000 = -6.210 (m³)
r1		0xE7			
r2		0xFF			
r3		0xFF			
r0	uint16	0x07	*alert_no_usage_interval__days*	when: *configured_parameters.alert_no_usage_sent* min: 1, max: 400 0 - *disabled*	0x0007 = 7 (days)
r1		0x00			
r0	uint8	0x01	*alert_tamper_enabled*	when: *configured_parameters.alert_tamper_sent*	*true*

1.2.2 *location_configuration_packet*

In EU868 region DR0 has maximum payload length of 51 bytes which is less than maximum data in *location_configuration_packet*. Therefore the packet may be sent or received in two parts, the longest address field and all the rest of the fields. *address*, *id_customer* and *id_location* all contain utf-8 string where single unicode symbol can take several bytes.

Example payload to/from fPort 50:
21 1F AC5E6D23 0017C10E F4 12 c396c3b662696b75205374722e2032 2d3136 10 32365630303030303030303337313734 05 3132414236

<table>
<tr><th>Byte</th><th>Type</th><th>Example</th><th>Parameter</th><th>Details</th><th>Example Value</th></tr>
<tr><td>0</td><td>uint8</td><td>0x21</td><td><i>_packet_type</i></td><td>0x21 - <i>location_configuration_packet</i></td><td><i>location_configuration_packet</i></td></tr>
<tr><td>1</td><td>bits8</td><td>0x1F</td><td><i>configured_parameters</i></td><td colspan="2"><table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0</td><td><i>gps_position_sent</i></td><td>b1</td></tr>
<tr><td>1</td><td><i>time_zone_sent</i></td><td>b1</td></tr>
<tr><td>2</td><td><i>address_sent</i></td><td>b1</td></tr>
<tr><td>3</td><td><i>id_customer_sent</i></td><td>b1</td></tr>
<tr><td>4</td><td><i>id_location_sent</i></td><td>b1</td></tr>
<tr><td>5-7</td><td>RFU</td><td>b000</td></tr>
</table></td></tr>
<tr><td>r0</td><td>int32</td><td>0xAC</td><td><i>gps_position_latitude__deg</i></td><td>when: <i>configured_parameters.gps_position_sent</i>
converter: /10e7
0x7FFFFFFF - <i>not_configured</i></td><td>0x236D5EAC = 594370220
594370220 / 10000000 = 59.4370220 = 59°26'13.3"N</td></tr>
<tr><td>r1</td><td></td><td>0x5E</td><td></td><td></td><td></td></tr>
<tr><td>r2</td><td></td><td>0x6D</td><td></td><td></td><td></td></tr>
<tr><td>r3</td><td></td><td>0x23</td><td></td><td></td><td></td></tr>
<tr><td>r4</td><td>int32</td><td>0x00</td><td><i>gps_position_longitude__deg</i></td><td></td><td>0x0EC11700 = 247535360
247535360 / 10000000 = 24.7535360 = 24°45'12.7"E</td></tr>
<tr><td>r5</td><td></td><td>0x17</td><td></td><td></td><td></td></tr>
<tr><td>r6</td><td></td><td>0xC1</td><td></td><td></td><td></td></tr>
<tr><td>r7</td><td></td><td>0x0E</td><td></td><td></td><td></td></tr>
<tr><td>r0</td><td>int8</td><td>0xF4</td><td><i>time_zone__h</i>
(winter time recommended)</td><td>when: <i>configured_parameters.time_zone_sent</i>
converter: /4
min: -12 h
max: 14 h</td><td>0xF4 = -12
-12 / 4 = -3.0 (h) = UTC-3</td></tr>
<tr><td>r0</td><td>uint8</td><td>0x12</td><td><i>address_len</i>
(length in bytes)</td><td>when: <i>configured_parameters.address_sent</i>
values: 1 - 38
0x00 - <i>not_configured</i></td><td>0x12 = 18 (bytes)</td></tr>
<tr><td>r1</td><td>string</td><td>0xc3</td><td><i>address</i></td><td>when: <i>address_len</i> not 0
length: <i>address_len</i>
type: utf-8 string
note: does not end with '\0'</td><td>0xc3 0x96 0xc3 0xb6 0x62 0x69 0x6b 0x75 0x20 0x53 0x74 0x72 0x2e 0x20 0x32 0x2d 0x31 0x36 →
"Ööbiku Str. 2-16"</td></tr>
<tr><td>r2</td><td></td><td>0x96</td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td>0xc3 0xb6 0x62 0x69 0x6b 0x75 0x20 0x53 0x74 0x72</td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

	r0+ *_address_len*		0x2e 0x20 0x32 0x2d 0x31 0x36			
	r0	uint8	0x10	*id_customer_len* (length in bytes)	when: *configured_parameters.id_customer_sent* values: 1-16 0x00 - *not_configured*	0x10 = 16 (bytes)
	r1	string	0x32	*id_customer*	when: *id_customer_len* not 0 length: *id_customer_len* type: utf-8 string note: string does not end with '\0'	0x32 0x36 0x56 0x30 0x30 0x30 0x30 0x30 0x30 0x30 0x30 0x33 0x37 0x31 0x37 0x34 → "26V0000000037174"
	r2		0x36			
	...		0x56 0x30 0x30 0x30 0x30 0x30 0x30 0x30 0x30 0x33 0x37 0x31 0x37 0x34			
	...					
	r0+ *_id_customer_len*					
	r0	uint8	0x05	*id_location_len* (length in bytes)	when: *configured_parameters.id_location_sent* values 1-16 0x00 - *not_configured*	0x05 = 5 (bytes)
	r1	string	0x31	*id_location*	when: *id_location_len* not 0 length: *id_location_len* type: utf-8 string note: string does not end with '\0'	0x31 0x32 0x41 0x42 0x36 → "12AB6"
	r2		0x32			
	...		0x41			
	...		0x42			
	r0+ *_id_location_len*		0x36			

1.3 LoRaWAN Command Payloads

There are two types of commands: configuration requests (fPort 49) and commands (fPort 60).

1.3.1 *general_configuration_request* and *location_configuration_request*

The response is sent to same port in the same format as configuration message for this package.

Example payload to fPort 49: 20

Byte	Type	Example	Parameter	Details	Example Value
0	uint8	0x20	*_packet_type*	*0x20 - general_configuration_request* *0x21 - location_configuration_request*	*general_configuration_request*

1.3.2 *local_time_request* and *enter_dfu_command*

enter_dfu_command response is *shutdown_packet* with reason *enter_dfu*, sent only if duty allows it. local_time_request is responded with local_time_response.

Example payload to fPort 60: FF

Byte	Type	Example	Parameter	Details	Example Value
0	uint8	0xFF	*command_type*	0x03 - *local_time_request* 0xFF - *enter_dfu_command*	enter_dfu_command

1.3.3 *local_time_response*

Example payload from fPort 60: 03 34546A5F

Byte	Type	Example	Parameter	Details	Example Value
0	uint8	0x03	*_packet_type*	0x03 - *local_time_response*	*local_time_response*
1	uint32_t	0x34	*device_local_time__s* *device_local_time_formatted* (CM3061 local time)	note: treat as unix epoch in local time	0x5F6A5434 = 1600803892 → 2020-09-22T19:44:52Z (local time)
2		0x54			
3		0x6A			
4		0x5F			

1.3.4 *faulty_downlink_packet*

If any configuration or command packet parsing fails, an error code is sent back.

Example payload from fPort 99: 13 32 05

Byte	Type	Example	Parameter	Details	Example Value
0	uint8	0x13	*_packet_type*	0x13 - *faulty_downlink_packet*	*faulty_downlink_packet*
1	uint8	0x32	*packet_fport*	original fPort where the invalid packet arrived	0x32 = 50 (fPort)
2	uint8	0x05	*packet_error_reason*	0x00 - *n/a* 0x01 - *n/a* 0x02 - *unknown_fport* 0x03 - *packet_size_short* 0x04 - *packet_size_long* 0x05 - *value_error* 0x06 - *protocol_parse_error* 0x07 - *reserved_flag_set* 0x08 - *invalid_flag_combination* *unsupported_header* *internal_error*	*value_error*

1.4 LoRaWAN Other Payloads

1.4.1 *boot_packet*

Sent after boot and every rejoin.

Example payload from fPort 99: 00 2B001650 020312 80 A0 FF 0000 371100

<table>
<tr><th>Byte</th><th>Type</th><th>Example</th><th>Parameter</th><th>Details</th><th>Example Value</th></tr>
<tr><td>0</td><td>uint8</td><td>0x00</td><td>*_packet_type*</td><td>0x00 - *boot_packet*</td><td>*boot_packet*</td></tr>
<tr><td>1</td><td rowspan="4">uint32</td><td>0x2B</td><td rowspan="4">*device_serial*
(sn of CM3061)</td><td rowspan="4">formatting: hex</td><td rowspan="4">0x5016002B → 5016002B</td></tr>
<tr><td>2</td><td>0x00</td></tr>
<tr><td>3</td><td>0x16</td></tr>
<tr><td>4</td><td>0x50</td></tr>
<tr><td>5</td><td>uint8</td><td>0x02</td><td rowspan="3">*device_firmware_version*</td><td>*hardware*</td><td rowspan="3">0x02 0x03 0x12 → 2.3.18</td></tr>
<tr><td>6</td><td>uint8</td><td>0x03</td><td>*major*</td></tr>
<tr><td>7</td><td>uint8</td><td>0x12</td><td>*minor*</td></tr>
<tr><td>8</td><td>bits8</td><td>0x80</td><td>*wakeup_reason*</td><td colspan="2">note: all fields are cleared if *general_info: lorawan_rejoin*
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0</td><td>RFU</td><td>0</td></tr>
<tr><td>1</td><td>*watchdog_reset*</td><td>0</td></tr>
<tr><td>2</td><td>*soft_reset   (e.g. from dfu_mode)*</td><td>0</td></tr>
<tr><td>3</td><td>RFU</td><td>0</td></tr>
<tr><td>4</td><td>*magnet_wakeup*</td><td>0</td></tr>
<tr><td>5-6</td><td>RFU</td><td>b00</td></tr>
<tr><td>7</td><td>*nfc_wakeup*</td><td>1</td></tr>
</table>
example: [*nfc_wakeup*]</td></tr>
<tr><td>9</td><td>bits8</td><td>0xA0</td><td>*general_info*</td><td colspan="2">
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0-4</td><td>RFU</td><td>b00000</td></tr>
<tr><td>5</td><td>*wakeup_from_dfu*</td><td>1</td></tr>
<tr><td>6</td><td>*lorawan_rejoin*</td><td>0</td></tr>
<tr><td>7</td><td>*configuration_restored*</td><td>1</td></tr>
</table></td></tr>
<tr><td>10</td><td>uint8</td><td>0xFF</td><td>*hardware_configuration*</td><td>0x20 - *BK_G*
0xFF - *n/a* (will never occur)</td><td>*n/a*</td></tr>
<tr><td>11</td><td rowspan="2">uint16</td><td>0x00</td><td rowspan="2">RFU</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>12</td><td>0x00</td></tr>
<tr><td>13</td><td rowspan="3">uint24</td><td>0x37</td><td rowspan="3">*device_uptime_accumulated__days*</td><td rowspan="3">convert: /24.0</td><td rowspan="3">0x001137 = 4407 (h)
4407 / 24.0 ≈ 183,6 (days)</td></tr>
<tr><td>14</td><td>0x11</td></tr>
<tr><td>15</td><td>0x00</td></tr>
</table>

1.4.2 *shutdown_packet*

If LoRaWAN is not duty-limited, shutdown packet is sent out to indicate the shutdown reason

Example payload from fPort 99:
01 33 04C00054 433F254E00752B036BD1164A337C44983938

Byte	Type	Example	Parameter	Details	Example Value
0	uint8	0x01	*_packet_type*	0x01 - *shutdown_packet*	*shutdown_packet*
1	uint8	0x33	*_shutdown_reason*	0x31 - *magnet_shutdown* 0x32 - *enter_dfu* 0x33 - *app_shutdown* 0x34 - *switch_to_wmbus*	*app_shutdown*
2	rest of bytes	0x04	*[usage_with_status_packet]*	comment: full *usage_with_status_packet* payload structure	see *usage_with_status_packet* example
3		0xC0			
4		0x00			
...		...			
...		...			

2 wM-Bus Payload

Example payload:
2F2F 0374 201C00 0414 AE0A8D01 4414 180A8D01 426C 752B 02FD74 BD0C 0F01 8088

Byte	Type	Example	Parameter	Example Value
0	uint8	0x2F	*aes_verification*	
1	uint8	0x2F	*aes_verification*	
2	uint8	0x03	*meter_actuality_duration__s* - DIF (int24)	
3	uint8	0x74	*meter_actuality_duration__s* - VIF (sec)	
4	int24	0x20	*meter_actuality_duration__s* (for *meter_accumulated_volume__L*)	0x001C20 = 7200 (sec)
5		0x1C		
6		0x00		
7	uint8	0x04	*meter_accumulated_volume__L* - DIF 0x04 (int32)	
8	uint8	0x14 (0x13...0x17)	*meter_accumulated_volume__L* - VIF	0x14, *meter_multiplier* = 10
9	int32	0xAE	*meter_accumulated_volume__L* note: not sent if *privacy_mode_active*	0x018D0AAE = 26020526 26020526 * *meter_multiplier* = 26020526 * 10 = 260205260 (L)
10		0x0A		
11		0x8D		
12		0x01		
13	uint8	0x44	*meter_key_date_accumulated_volume__L* - DIF (int32)	
14	uint8	0x14 (0x13...0x17)	*meter_key_date_accumulated_volume__L* - VIF	0x14, *meter_multiplier* = 10
15	int32	0x18	*meter_key_date_accumulated_volume__L*	0x018D0A18 = 26020376 26020376 * *meter_multiplier* = 26020376 * 10 = 260203760 (L)
16		0x0A		
17		0x8D		
18		0x01		
19	uint8	0x42	*meter_key_date* - DIF (int16)	
20	uint8	0x6C	*meter_key_date* - VIF	
21	int16	0x75	*meter_key_date* (for *meter_key_date_accumulated_volume__L*)	0x75 0x2B → 21.11.2019
22		0x2B		
23	uint8	0x02	*remaining_battery__days* - DIF (int16)	
24	uint8	0xFD	*remaining_battery__days* - VIF	
25	uint8	0x74	*remaining_battery__days* - VIFE	
26	int16	0xBD	*remaining_battery__days*	0x0CBD = 3261 (days)
27		0x0C		
28	uint8	0x0F	*manufacturer_specific* - DIF	
29	uint8	0x01	*manufacturer_specific* - Type (decimal 1)	
30	bits8	0x80	*manufacturer_specific*	

Bit	Parameter	Example
0	*alert_tamper*	0
1	RFU	0
2	RFU	0
3	RFU	0
4	RFU	0
5	*alert_battery_low*	0
6	RFU	0
7	*alert_no_usage*	1

<table>
<tr><td></td><td>31</td><td>bits8</td><td>0x88</td><td><em>manufacturer_specific</em>
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0-2</td><td>RFU</td><td>b000</td></tr>
<tr><td>3-7</td><td><em>battery_remaining__semesters</em> (half years)</td><td>b10001 = 17
17 / 2 = 8.5 (years)</td></tr>
</table>
</td></tr>
</table>

3 Annex: Understanding Payload Structures

3.1 Example Payload Structure

Every Payload structure description comes in following structure: example payload and then table with similar columns.

Example payload from/to fPort 50: 02 4A 10 0E 32

<table>
<tr><th>G</th><th>Byte</th><th>Type</th><th>Example</th><th>Parameter</th><th>Details</th><th>Example Value</th></tr>
<tr><td rowspan="2"></td><td>0</td><td>uint8</td><td>0x02</td><td>*_packet_type*</td><td>0x02 - *configuration_packet*</td><td>*configuration_packet*</td></tr>
<tr><td>1</td><td>bits8</td><td>0x4A</td><td>*configured_parameters*</td><td colspan="2">Example: 0x4A = b0100 10 1 0
<table>
<tr><th>Bit</th><th>Parameter</th><th>Example</th></tr>
<tr><td>0</td><td>*secondary_interval_sent*</td><td>0</td></tr>
<tr><td>1</td><td>RFU</td><td>1</td></tr>
<tr><td>2-3</td><td>*some_2bit_parameter*</td><td>b10 = 2</td></tr>
<tr><td>4-7</td><td>*some_4bit_parameter*</td><td>b0100 = 4</td></tr>
</table>
</td></tr>
<tr><td></td><td>2</td><td rowspan="2">int16</td><td>0x10</td><td rowspan="2">*main_interval__s*</td><td rowspan="2">converter: *2</td><td rowspan="2">0x0E 10 = 3600 * 2 = 7200 (s)</td></tr>
<tr><td></td><td>3</td><td>0x0E</td></tr>
<tr><td></td><td>r0</td><td>uint8</td><td>0x32</td><td>*secondary_interval*</td><td>when: *configured_parameters:secondary_interval_sent*
0xFF - *not_configured*</td><td>0x32 = 50</td></tr>
</table>

The payloads are described in top to bottom structure. Bitwise least significant bit comes first (lsb), bytewise least significat byte comes first (LSB).

Human readability of hex arrays introduces confusion into this. E.g. decimal value of 1000000 in hexadecimal is 0x000F4240 (LSB on the right) but in hexadecimal payload string it is usually printed LSB on the left 40 42 0F 00 .

Example payload hex has LSB on the left. Different parameter portions in hex are separated with whitespace. Packet fPort and direction (in reference to NAS module) are also signified with to (downlink) and from (uplink).

3.2 Payload Structure Columns Explanation

Column: G

Optional group column to help to identify which blocks are inseparable.

Column: Byte

Byte column expresses byte position. Always starting with 0 due to array first member being 0. The sequence is broken with the start of every optional block. From the first optional block onwards the positions of the bytes are relative, therefore e.g. r0, r1.

Column: Type

Type column determines encoding and length of the data.

Type	Length in bytes	Encoding
uint8 / int8	1	8 bit unsigned / signed integer
uint16 / int16	2	16 bit unsigned / signed integer
uint32 / int32	4	32 bit unsigned / signed integer
uint64 / int64	8	64 bit unsigned / signed integer
bits8 / bits16	1 / 2	8 bit / 16 bit bitfield (flags and/or decimals)
string	N	N bytes of string (byte array), encoding utf-8

Column: Example

Example bytes to help to understand how bytes from example payload are mapped. Hex values on Example column hex values can be matched top-down with Example payload hex left to right. Grey hex means that this value is constant.

Column: Parameter

parameter_name is descriptive name of parameter, formatted in *italic_snake_case*. They are used consistently throughout the documentation and intended to be used in parsers.
RFU - reserved for future use
grey_parameter - should be hidden in the end parser to declutter the results.
x_parameter_sent - sent keyword indicates that this bit controls following block denoted like when: *x_parameter_sent*.
main_interval__s - the unit is always behind double underscores '__'. Unit L/h would be __L_h. Unit °C would be '__C'.

Column: Details

defined_options or *special_values* are italic and indigo blue.
n/a - not available / not applicable

Under detailed bitfields tables Bit 0 is 0x01 and Bit 7 is 0x80.

Used Keywords:
when: defines the conditions when the block will be active. When shows the relationships between flags and existence of optional blocks.
converter: defines how to convert the value to extract the intended value by e.g. adding offset or multiplying with something.
formatter: defines how to format the value.

Column: Example Value

Shows how to convert the Example column's hex into useful output value.