



**IMST GmbH**

Carl-Friedrich-Gauß-Str. 2-4, D-47475 Kamp-Lintfort

# Wireless M-Bus Range Extender

AN034 - Remote Access Protocol

Version 1.7

## **Document State**

wip

## **Date**

October 2025

## **Document ID**

4000/40140/0165

© 2025 IMST GmbH - All rights reserved

## History

Version	Datum	Comment
1.0	28.09.2020	Initial Version
1.1	15.02.2021	Status Bit 3 for LoRaWAN Activation Procedure added in Application Status
1.2	29.09.2021	Updates for Firmware V1.1 <ul style="list-style-type: none"><li>• <a href="#">3 - Accessible Resources - Application Status</a></li><li>• <a href="#">3 - Accessible Resources - Extras</a></li></ul>
1.3	26.11.2021	Segmented Services for Remote Access removed
1.4	13.01.2022	Notes for unknown Service Requests and unknown Resources added Resource ID for Calendar Items corrected in example code
1.5	March 2024	<a href="#">Application Status</a> updated with respect to Firmware V1.4
1.6	June 2025	<a href="#">Application Status</a> updated with respect to Firmware V1.3
1.7	October 2025	<ul style="list-style-type: none"><li>• Document title changed</li><li>• <a href="#">Application Status</a> updated with respect to Firmware V1.5: Low Battery Detection</li><li>• <a href="#">3 - Accessible Resources - Extras</a> updated with respect to Firmware V1.5: Unconfirmed Packet Upload</li><li>• <a href="#">1 - Remote Access Services - Details</a> Example for <b>Delete</b> service modified</li><li>• Chapter 4: <a href="#">4 - LoRaWAN Downlink Message Sequencing</a> added</li><li>• <a href="#">3 - Accessible Resources</a>: Reject Filter Items added</li><li>• <a href="#">3 - Accessible Resources - LoRaWAN Stack Settings</a> added</li></ul>

## Aim of this document

This document includes a description of the Remote Access Protocol which is supported by the Wireless M-Bus Range Extender. This protocol enables applications to configure the Wireless M-Bus Range Extender over the air from application server side.

### Notation Info

Suffix "b" = binary data

Suffix "h" = hexadecimal data

Without suffix = decimal data

Multi byte / octet fields are considered to be treated as unsigned integers with **Least Significant Byte** first unless explicitly noted

## Content

- [1 - Remote Access Services - Overview](#)
- [2 - Summary of Services](#)
- [3 - Accessible Resources](#)
- [4 - LoRaWAN Downlink Message Sequencing](#)

# 1 - Remote Access Services - Overview

LoRaWAN end nodes like the Wireless M-Bus Range Extender can be accessed through the LoRaWAN network server by means of so called downlink messages.

Many LoRaWAN network servers provide a RESTful api or gRPC interface which allows to enqueue such messages. Initiating a downlink message requires at least three information elements:

1. A mean to identify the LoRaWAN end device like the unique **64-Bit DeviceEUI**
2. A **LoRaWAN Port** number which is part of the LoRaWAN messages
3. The **message payload**

The next sub chapters describe the **message payload** which transports the service requests or corresponding response messages.

Note: LoRaWAN supports a **guaranteed payload capacity** of  $64 - 13 - 15 = 36$  bytes.

## Simple not segmented Services via LoRaWAN Port 20<sub>h</sub>

The following sub chapters describe the supported services in more detail:

Service	Description
<a href="#">Get</a>	Used to read a resource
<a href="#">Get Sub Items Count</a>	Used to read the number of sub items within a resource e.g. the number of configured Calendar Events
<a href="#">Get Sub Item</a>	Used to read a specific sub item within a list of items
<a href="#">Set</a>	Used to configure a resource
<a href="#">Set Sub Item</a>	Used to configure a specific sub item within a list of items
<a href="#">Add Sub Item</a>	Used to append a new sub item to a list
<a href="#">Delete</a>	Used to delete a resource
<a href="#">Delete Sub Item</a>	Used to delete a single sub item within a list

Table 1 : Remote Access Services

[Back to Top](#)

## 1 - Remote Access Services - Details

The following sub chapters include a detailed description of the implemented services:

### Get

This service can be used to read a specific resource. In case of large resources the response might be transmitted as a series of segments which need to be reassembled before further processing.

#### Syntax:

Request: < Get > < Resource ID >

Response < Get Response > < Resource ID > < data >

#### Example: Get DateTime

Request

< Get > < DateTime >

< 01<sub>h</sub> > < 01<sub>h</sub> >

Response

< Get Response > < DateTime > < seconds since epoch ( 1.1.1970, midnight UTC/GMT ) as **unsigned** 32-Bit integer, LSB first >

< 02<sub>h</sub> > < 01<sub>h</sub> > < ( 35 6E 7D 5F )<sub>h</sub> >

## Get Sub Item Count

---

This service can be used to retrieve the number of sub items from a resource.

### Syntax:

Request: < Get Sub Items Count > < Resource ID >

Response < Get Sub Items Count Response > < Resource ID > < Count >

### Example: Get Calender Event Count

Request

< Get Sub Items Count > < Calendar >

< 03<sub>h</sub> > < 02<sub>h</sub> >

Response

< Get Sub Items Count Response > < Calendar > < 4 >

< 04<sub>h</sub> > < 02<sub>h</sub> > < 04<sub>h</sub> >

## Get Sub Item

---

This service can be used to read a single sub item with given index. The index is in the range 0 .. Count - 1

### Syntax:

Request: < Get Sub Item > < Resource ID > < Index >

Response < Get Sub Item Response > < Resource ID > < Index > < data >

### Example: Get Calendar Event Item

Request

< Get Sub Item > < Calendar > < Index : 03<sub>h</sub> >

< 05<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> >

Response

< Get Sub Item Response > < Calendar > < 03<sub>h</sub> > < Calendar Event Item Data >

< 06<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < Calendar Event Item Data >

## Set

---

This service can be used to configure a given resource.

For resources which provide a list of sub items the complete list of sub items will be overwritten!

### Syntax:

Request: < Set > < Resource ID > < data >

Response < Set Response > < Resource ID > < Status Code >

#### Example: Set DateTime

Request

< Set > < DateTime > < seconds since epoch ( 1.1.1970, midnight UTC/GMT ) as **unsigned** 32-Bit integer, LSB first >

< 07<sub>h</sub> > < 01<sub>h</sub> > < ( 35 6E 7D 5F )<sub>h</sub> >

Response

< Set Response > < DateTime > < Success >

< 08<sub>h</sub> > < 01<sub>h</sub> > < 00<sub>h</sub> >

## Set Sub Item

---

This service can be used to change a given sub item of a given resource. In case the sub item doesn't exist a new sub item will be created and appended. The returned index will point to the new created item.

#### Syntax:

Request: < Set Sub Item > < Resource ID > < Index > < data >

Response < Set Sub Item Response > < Resource ID > < Index > < Status Code >

#### Example: Set Calendar Event

Request

< Set Sub Item > < Calendar > < 03<sub>h</sub> > < Calendar Event Data >

< 09<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < Calendar Event Data >

Response

< Set Sub Item Response > < Calendar > < Success >

< 0A<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> > < 00<sub>h</sub> >

## Add Sub Item

---

This service can be used to create and append a new sub item of a given resource. The response contains the index of the new created item.

#### Syntax:

Request: < Add Sub Item > < Resource ID > < data >

Response < Add Sub Item Response > < Resource ID > < Index > < Status Code >

#### Example: Add Calendar Event

Request

< Add Sub Item > < Calendar > < Calendar Event Data >

< 0B<sub>h</sub> > < 02<sub>h</sub> > < Calendar Event Data >

Response

< Add Sub Item Response > < Calendar > < Success >

< 0C<sub>h</sub> > < 02<sub>h</sub> > < 04<sub>h</sub> > < 00<sub>h</sub> >

## Delete

---

This service can be used to delete a given resource.

For resources which provide a list of sub items the complete list of sub items will be deleted!

### Syntax:

Request: < Delete > < Resource ID >

Response < Delete Response > < Resource ID > < Status Code >

### Example: Delete All WM-Bus Device Filter Items

Request

< Delete > < all WM-Bus Device Filter Items >

< 0D<sub>h</sub> > < 06<sub>h</sub> >

Response

< Delete Response > < all WM-Bus Device Filter Items > < Success >

< 0E<sub>h</sub> > < 06<sub>h</sub> > < 00<sub>h</sub> >

## Delete Sub Item

---

This service can be used to delete a single sub item of given resource.

### Syntax:

Request: < Delete Sub Item > < Resource ID > < Index >

Response < Delete Sub Item Response > < Resource ID > < Index > < Status Code >

### Example: Delete single WM-Bus Device Accept Filter Item

Request

< Delete Sub Item > < WM-Bus Device Filter Item> < Index: 3 >

< 0F<sub>h</sub> > < 06<sub>h</sub> > < 03<sub>h</sub> >

Response

< Delete Sub Item Response > < WM-Bus Device Filter Item > < Index > < Success >

< 10<sub>h</sub> > < 06<sub>h</sub> > < 03<sub>h</sub> > < 00<sub>h</sub> >

[Back to 1 - Remote Access Services - Overview](#)

## 2 - Summary of Services

### 2.1 Services Overview

This table gives a summary of the implemented services, their service codes and syntax:

#	Name	Request Code	Response Code	Syntax	
				Request	Response
1	Status Response	n.a.	00 <sub>h</sub>		< Status Response > < Resource ID > < Status Code >
2	Get	01 <sub>h</sub>	02 <sub>h</sub>	< Get > < Resource ID >	< Get Response > < Resource ID > < data >
3	Get Sub Items Count	03 <sub>h</sub>	04 <sub>h</sub>	< Get Sub Items Count > < Resource ID >	< Get Sub Items Count Response > < Resource ID > < Count >
4	Get Sub Item	05 <sub>h</sub>	06 <sub>h</sub>	< Get Sub Item > < Resource ID > < Index: 0..Count -1 >	< Get Sub Item Response > < Resource ID > < Index > < data >
5	Set	07 <sub>h</sub>	08 <sub>h</sub>	< Set > < Resource ID > < data >	< Set Response > < Resource ID > < Status Code >
6	Set Sub Item	09 <sub>h</sub>	0A <sub>h</sub>	< Set Sub Item > < Resource ID > < Index > < data >	< Set Sub Item Response > < Resource ID > < Index > < Status Code >
7	Add Sub Item	0B <sub>h</sub>	0C <sub>h</sub>	< Add Sub Item > < Resource ID > < data >	< Add Sub Item Response > < Resource ID > < Index > < Status Code >
8	Delete	0D <sub>h</sub>	0E <sub>h</sub>	< Delete > < Resource ID >	< Delete Response > < Resource ID > < Status Code >
9	Delete Sub Item	0F <sub>h</sub>	10 <sub>h</sub>	< Delete Sub Item > < Resource ID > < Index >	< Delete Sub Item Response > < Resource ID > < Index > < Status Code >

Table 2.1 : Summary of Services

1. Unknown Service Request Codes will be ignored, i.e. no response will be sent back to the server.
2. Service Requests for unknown Resources IDs will be answered with a corresponding Reponse and Status Code "Resource not found".

### 2.2 Status Codes

The following table lists the possible returned **Status Codes**:

#	Status	Code
1	Success	00 <sub>h</sub>
2	Failure	01 <sub>h</sub>
3	Resource not found	02 <sub>h</sub>
4	Sub Item Index not found	03 <sub>h</sub>

Table 2.2 : Status Codes

Back to [Top](#)



## 3 - Accessible Resources

### 3.1 Overview

The following table lists the accessible resources, their corresponding identifier (Resource ID) and the provided services:

#	Resource	Resource ID	Get	Get Sub Item Count	Get Sub Item	Set	Set Sub Item	Add Sub Item	Delete	Delete Sub Item
1	<a href="#">DateTime</a>	01 <sub>h</sub>	yes	no	no	yes	no	no	no	no
2	<a href="#">Calendar Events</a>	02 <sub>h</sub>	yes, not recommended over LoRaWAN	yes	yes	yes, but not recommended over LoRaWAN	yes	yes	yes, but not recommended over LoRaWAN	yes
3	<a href="#">Application Status</a>	03 <sub>h</sub>	yes	no	no	no	no	no	no	no
4	reserved	04 <sub>h</sub>	no	no	no	no	no	no	no	no
5	<a href="#">Extras</a>	05 <sub>h</sub>	yes	no	no	yes	no	no	no	no
6	<a href="#">WM-Bus Accept Filter</a> ( <a href="#">WM-Bus Device Filter Items</a> )	06 <sub>h</sub>	yes, not recommended over LoRaWAN	yes	yes	yes, not recommended over LoRaWAN	yes	yes	yes	yes
7	<a href="#">WM-Bus Reject Filter</a> ( <a href="#">WM-Bus Device Filter Items</a> )	07 <sub>h</sub>	yes, not recommended over LoRaWAN	yes	yes	yes, not recommended over LoRaWAN	yes	yes	yes	yes
8	<a href="#">LoRaWAN Stack Settings</a>	1E <sub>h</sub>	yes	no	no	yes	no	no	no	no

Table 3.1 : Accessible Resources

[Back to Top](#)

## 3 - Accessible Resources - DateTime

### 3.2 Date Time

The DateTime resource enables access to the current date and time of the WM-Bus Range Extender. The DateTime resource can be "Get" and "Set".

The data format looks as follows:

Resource	DateTime
Format	UTC, seconds since 01.01.1970, 00:00:00
Octets	32 Bit, unsigned Integer, LSB first
Example	( 19 9E 64 5F ) <sub>h</sub>
	5F649E19 <sub>h</sub> = 1.600.429.593 seconds since 01.01.1970, 00:00:00
	"2020-09-18 11:46:33"

Table 3.2 : DateTime Resource

#### Example Set DateTime

< Set > < Resource ID > < DateTime >

< 07<sub>h</sub> > < 01<sub>h</sub> > < ( 19 9E 64 5F )<sub>h</sub> >

[Back to 3 - Accessible Resources](#)

## 3 - Accessible Resources - Calendar Events

### 3.3 Calendar Events

The Calendar resource enables access to the configurable Calendar Events. It is possible to get, set and delete the complete list of all sub items by means of the Tiny Transport Protocol (see User Manual) or to set, get and delete only one sub item per request.

The data format of a single sub item looks as follows:

Resource	Calendar Event				
Format	Event ID	Filter Group ID	Repetition Type	Repetition Step Size	Date & Time (UTC)
Octets	8 Bit	8 Bit	8 Bit	8 Bit	32 Bit, LSB first
Example	41 <sub>h</sub> = Receive and record in C/T-Mode see <a href="#">Application Events</a>	FF <sub>h</sub> = always	03h = Daily	00h	( 19 9E 64 5F ) <sub>h</sub> see <a href="#">DateTime</a>

Table 3.3 : Calendar Event

- **Event ID**  
The event type defines the kind of action to be performed. A list of possible Event Types is given here: [Application Events](#)
- **Filter Group ID**  
This element is only used in combination with Wireless M-Bus reception / recording types. It defines the group of WM-Bus Filter Items which should be applied during a Wireless M-Bus reception / recording phase.  
Note: The value 255 ( FF<sub>h</sub> ) is reserved and means that all configured Wireless M-Bus Filters should be applied independent of their own configured [Filter Group ID](#)
- **Repetition Type**  
The repetition type defines the periodicity of an event:  
0 = No repetition, single event, can be used for test purpose  
1 = Every Minute  
2 = Hourly  
3 = Daily  
4 = Weekly  
5 = Monthly
- **Repetition Step Size**  
The repetition step size is a second parameter which defines the periodicity of an event:  
Example 1: Repetition Type = 2 ( Hourly ), Repetition Step Size = 2 => Repetition Interval = every 2 + 1 = 3 hours  
Example 2: Repetition Type = 3 ( Daily ), Repetition Step Size = 0 => Repetition Interval = every 0 + 1 = 1 days
- **Date & Time**  
The date / time element defines when the event should be scheduled for the first time.

#### Example Add Calendar Event Subitem

< Add Sub Item > < Resource ID > < Calendar Event Sub Item >

< 0B<sub>h</sub> > < 02<sub>h</sub> > < ( 41 FF 03 00 19 9E 64 5F )<sub>h</sub> >

[Back to 3 - Accessible Resources](#)

### 3 - Accessible Resources - Application Events

#### 3.3.1 Application Events

The following table lists all application events

Event Name	Event ID	via Calendar	via HCI Interface	Description
None	00 <sub>h</sub>	no	no	Invalid event
UI Events				
Show Status	01 <sub>h</sub>	yes	yes	Output of internal status on LED
Push Button	02 <sub>h</sub>	yes	yes	Simulates the push button function: performs LoRaWAN Activation per OTAA or ABP or if already activated displays the status on LED
LED Off	03 <sub>h</sub>	yes	yes	Set LED off
LED Red	04 <sub>h</sub>	yes	yes	Set LED red color
LED Green	05 <sub>h</sub>	yes	yes	Set LED green color
LED Yellow	06 <sub>h</sub>	yes	yes	Set LED yellow color
LED Red Blinking	07 <sub>h</sub>	yes	yes	Set LED red blinking
LED Green Blining	08 <sub>h</sub>	yes	yes	Set LED green blinking
LED Yellow Blinking	09 <sub>h</sub>	yes	yes	Set LED yellow blinking
LoRaWAN Events				
LoRaWAN Activate	20 <sub>h</sub>	not recommended	yes	Activate LoRaWAN Stack per OTAA or ABP
LoRaWAN Deactivate	21 <sub>h</sub>	not recommended	yes	Deactivate LoRAWAN Stack
System Events				
Get Network Time	30 <sub>h</sub>	yes	yes	Request system time via LoRaWAN MAC command. On response the system time will be synchronized.
Send Application Status	31 <sub>h</sub>	yes	yes	Transmit Application Status via LoRaWAN
WM-Bus Events				
Receive in S-Mode and record	40 <sub>h</sub>	yes	yes	Enable receiver for Wireless M-Bus S-Mode, received messages will be filtered and stored in non-volatile flash memory.
Receive in C/T-Mode and record	41 <sub>h</sub>	yes	yes	Enable receiver for Wireless M-Bus C/T-Mode, received messages will be filtered and stored in non-volatile flash memory
Receiver Off	42 <sub>h</sub>	yes	yes	Disable receiver
Start Upload	43 <sub>h</sub>	yes	yes	Disable receiver and start upload of stored WM-Bus messages via LoRaWAN

Receive S-Mode and output via HCI	44 <sub>h</sub>	yes	yes	Enable receiver for Wireless M-Bus S-Mode, received messages will be forwarded via HCI
Received C/T-Mode and output via HCI	45 <sub>h</sub>	yes	yes	Enable receiver for Wireless M-Bus C/T-Mode, received messages will be forwarded via HCI

Table 3.3.1 : Application Events

[Back to 3 - Accessible Resources - Calendar Events](#)

## 3 - Accessible Resources - Application Status

### 3.4 Application Status

The Application Status includes the following information elements:

Resource	Applicati on Status								Firmware V1.1 ff.		Firmware V1.3 ff.	Firmware V1.4 ff.
Format	Date and Time (UTC)	Firmware Version	Last Sync Time	Reset Counter <sup>1)</sup>	Status Bits	WM-Bus Rx Counter <sup>2)</sup>	WM-Bus Stored Counter <sup>2)</sup>	WM-Bus Tx Counter <sup>2)</sup>	Battery Voltage in mV	Firmware Type	Minimum Battery Voltage in mV	Reserved Info
Octets	32 Bit, LSB first	16 Bit, Minor version first	32 Bit, LSB first	32 Bit, LSB first	16 Bit, LSB first	32 Bit, LSB first	32 Bit, LSB first	32 Bit, LSB first	16 Bit, LSB first	8 Bit	16 Bit	8 Bit
Example	see <a href="#">DateTime</a>	e.g. ( 07 01 ) h V1.7	see <a href="#">DateTime</a>						e.g. 3450 mV	0 = Release 1 = Beta	e.g. 2920 mV	

Table 3.4 : Application Status

- **Date and Time**  
Contains the current date and time in seconds since 01.01.1970 00:00:00
- **Firmware Version**  
Minor and major firmware version
- **Last Sync Time**  
Contains the time stamp of the latest synchronization via local or air interface
- **Reset Counter<sup>1)</sup>**  
Contains the number of device resets
- **Status / Error Bits**  
This field includes several Status Bits:
  - Bit 0: LoRaWAN<sup>®</sup> Activation State  
1 = LoRaWAN<sup>®</sup> Stack is not activated  
0 = Stack is activated
  - Bit 1: Network Time Synchronization State  
1 = No synchronization via LoRaWAN<sup>®</sup>  
0 = Synchronized via LoRaWAN<sup>®</sup>
  - Bit 2: System Time Synchronization State  
1 = RTC not synchronized at all  
0 = RTC synchronized ( via local serial interface or LoRaWAN<sup>®</sup> )
  - Bit 3: LoRaWAN Activation Procedure State  
1 = Activation Procedure active  
0 = Procedure not active
  - Bit 4: LoRaWAN Configuration State  
1 = Configuration is invalid Activation not possible  
0 = Configuration is valid
  - Bit 5: Wireless M-Bus Address Filter List Configuration State  
1 = Whitelist is empty no recording possible  
0 = Whitelist contains at least one item
  - Bit 6: Calendar Event List Configuration State  
1 = List is empty, note: from Firmware 1.5 onwards, an hourly 'Get Network Time' event is automatically issued when the Calendar Event List is empty.  
0 = List contains at least one item
  - Bit 7 : Reserved
  - Bit 8: Flash Memory Full State  
1 = A flash memory full condition has been detected during capture phase this bit will be automatically cleared during the

next recording phase  
0 = No error

- Bit 9: Flash Memory CRC Error State  
1 = A file CRC error has been detected during read & upload operation this bit will be automatically cleared during the next recording phase  
0 = No error

- Bit 12: Low Battery detected <sup>5)</sup> ( see also User Manual, chapter Battery Voltage Measurement )  
1 = a voltage below 3000 mV was measured once.  
0 = no warning

- **WM-Bus Rx Counter <sup>2)</sup>**

Total received WM-Bus packets before any packet filtering since last counter reset

- **WM-Bus Stored Counter <sup>2)</sup>**

Number of stored WM-Bus packets after packet filtering

- **WM-Bus Tx Counter <sup>2)</sup>**

Number of uploaded WM-Bus packets

- **Battery Voltage <sup>3) 6)</sup>**

The battery voltage is measured just before transmitting this status message. The value is returned in Millivolts. A value of 0 or FFFFh is considered invalid because no measurement has been performed so far.

- **Firmware Type <sup>3)</sup>**

This element indicates different types of firmware version: e.g. official released version or field test beta version:

0 = Release  
1 = Field Test Beta Version  
X = Reserved

- **Minimum Battery Voltage ( Firmware 1.3 ff.)**

This field contains the minimum measured battery voltage. The voltage is periodically measured during WM-Bus Receiving / WM-Bus Recording windows. The value is returned in Millivolts. A value of 0 or FFFFh is considered invalid because no measurement has been performed so far.

#### Info

<sup>1)</sup> The Reset Counter is copied to the non-volatile memory earliest 30 seconds after system start.

<sup>2)</sup> The WM-Bus packet counters are written into the non-volatile memory earliest 30 seconds after last increment. These counters can be reset by means of the WS Configurator PC - Tool.

<sup>3)</sup> Battery Voltage and Firmware Type are available from Firmware V1.1.

<sup>4)</sup> The Minimum Battery Voltage is available from Firmware V1.3.

<sup>5)</sup> The Low Battery Warning is available from Firmware V1.5.

<sup>6)</sup> No new measurement is triggered when this status is requested via the remote access protocol.

[Back to 3 - Accessible Resources](#)

## 3 - Accessible Resources - Extras

### 3.5 Extras

This resource provides some extra configuration parameters which control the behaviour of the Wireless M-Bus Range Extender.

Resource	Extras
Format	Option Bits
Octets	32 Bit, LSB first
Example	< 03 00 00 00 > <sub>h</sub>

Table 3.5 : Extras

- **Options Bits for Firmware Version <= 1.4**

This field includes several configuration bits:

Bit 0 : Duplicate WM-Bus Packet Filter:

0 = disabled

1 = enabled

Bit 1 : Duplicate WM-Bus Packet Filter with CRC :

0 = Verification of WM-Bus Header bytes only

1 = Verification of WM-Bus Header bytes and additional Packet CRC

Note: Bit 0 must be enabled too

Bit 2 - 3 : reserved for future

Bit 4 : LED usage for status signalling:

0 = disabled

1 = enabled

Bit 5 : WM-Bus Messages with RSSI ( Firmware Version 1.1ff )

0 = disabled

1 = enabled

Note: WM-Bus Message including RSSI will be uploaded on dedicated LoRaWAN Ports.

Bit 6 - 31 : reserved for future

- **Options Bits for Firmware Version 1.5ff.**

Bit 0 - 1 : New Coding for Duplicate Filter

0 = Filter disabled

1 = Simple Filter:

Verification of 10 WM-Bus Header bytes:

L-Field(1), C-Field(1), ManID(2), DeviceID(4), Version(1), Type(1)

2 = Single Station Packet Filter:

Verification of 8 WM-Bus Header bytes:

ManID(2), DeviceID(4), Version(1), Type(1)

3 = Advanced Filter:

Verification of 10 WM-Bus Header bytes and generated packet hash/crc value

Bit 6 : Unconfirmed WM-Bus Packet Uploads

0 = Use confirmed LoRaWAN uplink packets for upload session

1 = Use unconfirmed LoRaWAN uplink packets to reduce server side downlink traffic

Other Bits : as before

[Back to 3 - Accessible Resources](#)



### 3 - Accessible Resources - WM-Bus Device Filter Items

#### 3.6 WM-Bus Device Filter Items for Accept and Reject Filter

WM-Bus Device Filter Items are used to configure the Accept Filter (Ressource ID = 6) and Reject Filter ( Ressource ID = 7 ) of the WM-Bus Packet Filter. A single WM-Bus Device Filter Item includes the following information elements:

Resource	WM-Bus Device Filter Item					
Format	WM-Bus Address Fields				Address Field Mask	Filter Group ID
	Manufacturer ID <sup>1)</sup>	Device ID <sup>1)</sup>	Version	Type		
Octets	16 Bit	32 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Example						

Table 3.6 : WM-Bus Device Filter Item

**Note**

<sup>1)</sup> The byte ordering of multi byte fields is the same as in the Wireless M-Bus packets transmitted over the air.

- **WM-Bus Address Fields**  
A sequence of 8 bytes in total which are transmitted in the header of each Wireless M-Bus packet.
- **Address Field Mask**  
This mask defines which of the single WM-Bus Address Field Bytes is used for comparison with every received WM-Bus packet.  
Bit 0 = Type  
Bit 1 = Version  
Bit 2 .. 5 = Device ID Bytes  
Bit 6 .. 7 = Manufacturer ID Bytes
- **Filter Group ID**  
This element is only used to group several filter items.  
Note: The value 255 ( FF<sub>h</sub> ) is reserved and means that this filter item should be applied always, independently of the configuration of an [Calendar Event](#)

Back to [3 - Accessible Resources](#)



## 3 - Accessible Resources - LoRaWAN Stack Settings

---

### 3.7 LoRaWAN Stack Settings

This resource provides configuration parameters which control the behaviour of the integrated LoRaWAN Stack.

Resource	
Format	Number of Tx Retries
Octets	8 Bit
Example	< 04 > <sub>h</sub>

Table 3.7 : LoRaWAN Stack Settings

- **Number of Tx Retries**  
Defines the maximum number of confirmed uplink retries.  
The configurable range is 4 to 255.

[Back to 3 - Accessible Resources](#)

## 4 - LoRaWAN Downlink Message Sequencing

### 4.1 Importance of Complete Downlink Message Sequences

In LoRaWAN systems, it is critical to ensure that a **complete sequence** of individual LoRaWAN downlink messages is placed into the **downlink queue** of the gateway/server.

This is especially important for our Remote Access Protocol, because after a single downlink message is processed, the following occurs:

1. **Uplink Response Generation** – The end device sends an uplink response, which in turn opens new downlink reception slots.
2. **LoRaWAN Pending Bit Mechanism** – More importantly, the gateway/server sets the **LoRaWAN downlink pending bit** in each downlink message if additional messages remain in the queue.  
As a result, a Class A LoRaWAN end node will **automatically generate further uplinks** (sometimes referred to as “empty packets”) to retrieve the remaining queued downlink messages.

### 4.2 Example: Calendar Event List Update

1. Downlink - MessageDelete Calendar List

< 0D<sub>h</sub> > < 02<sub>h</sub> >

2. Downlink - Message : Set Sub Item ( Index = 0, data = ...):

< 09<sub>h</sub> > < 02<sub>h</sub> > < 00<sub>h</sub> > < data >

3. Downlink - Message : Set Sub Item ( Index = 1, data = ...):

< 09<sub>h</sub> > < 02<sub>h</sub> > < 01<sub>h</sub> > < data >

4. Downlink - Message : Set Sub Item ( Index = 2, data = ...):

< 09<sub>h</sub> > < 02<sub>h</sub> > < 02<sub>h</sub> > < data >

5. ... ( set further sub-items if needed )

### 4.3 Plan B: Incremental Update Without Full List Deletion

If, for any reason, the platform or LoRaWAN server/gateway cannot handle sending the complete sequence in one operation, use the following fallback procedure:

#### First Sequence:

1.1 Use **Get Sub Item Count** to read the current number of calendar events:

< 03<sub>h</sub> > < 02<sub>h</sub> >

1.2. - 1.n Use **Set Sub Item** to write/overwrite the desired **N<sub>new</sub>** events as shown in the example above.

During this process, an asynchronous response for *Get Sub Item Count* will be received:

< 04<sub>h</sub> > < 02<sub>h</sub> > < **N<sub>old</sub>** >

#### Second Sequence:

If  $k = N_{new} - N_{old}$  and  $k > 0$ , delete the unwanted trailing items.

Example:

If  $N_{new} = 3$  and  $N_{old} = 5$ , delete index 3 and 4:

2.1. Delete Sub Item (Index = 3):

< 0F<sub>h</sub> > < 02<sub>h</sub> > < 03<sub>h</sub> >

2.2. Delete Sub Item (Index = 4):

< 0F<sub>h</sub> > < 02<sub>h</sub> > < 04<sub>h</sub> >

#### 4.4 Final Verification Step

After writing the new data, all updated items should be read back for verification to ensure that the configuration matches the intended state.

[Back to Top](#)

