



IMST GmbH

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

Wireless M-Bus Range Extender

Remote Access Protocol for Application Server

Version 1.3

Document State

final

Date

November 2021

Document ID

4000/40140/0165

© 2021 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">• Accessible Resources - Application Status• Accessible Resources - Extras
1.3	26.11.2021	Segmented Services for Remote Access removed

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

-
- [Remote Access Services - Overview](#)
 - [Summary of Services](#)
 - [Accessible Resources](#)
-

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_h

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

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

Table : Remote Access Services

Back to [Top](#)

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_h > < 01_h >

Response

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

< 02_h > < 01_h > < (35 6E 7D 5F)_h >

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_h > < 02_h >

Response

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

< 04_h > < 02_h > < 04_h >

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_h >

< 05_h > < 02_h > < 03_h >

Response

< Get Sub Item Response > < Calendar > < 03_h > < Calendar Event Item Data >

< 06_h > < 02_h > < 03_h > < 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_h > < 01_h > < (35 6E 7D 5F)_h >

Response

< Set Response > < DateTime > < Success >

< 08_h > < 01_h > < 00_h >

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 Item

Request

< Set Sub Item > < Calendar > < 03_h > < Calendar Event Item Data >

< 09_h > < 02_h > < 03_h > < Calendar Event Item Data >

Response

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

< 0A_h > < 02_h > < 03_h > < 00_h >

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 Item

Request

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

< 0B_h > < 02_h > < Calendar Event Item Data >

Response

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

< 0C_h > < 02_h > < 04_h > < 00_h >

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 Calendar Events

Request

< Delete > < Calendar all Events >

< 0D_h > < 02_h >

Response

< Delete Response > < Calendar all Events > < Success >

< 0E_h > < 02_h > < 00_h >

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 Calendar Event Item

Request

< Delete Sub Item > < Calendar > < Index: 3 >

< 0F_h > < 02_h > < 03_h >

Response

< Delete Sub Item Response > < Calendar > < Success >

< 10_h > < 02_h > < 03_h > < 00_h >

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

Summary of Services

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

Table : Summary of Services

Status Codes

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

#	Status	Code
1	Success	00 _h
2	Failure	01 _h

3	Resource not found	02 _h
4	Sub Item Index not found	03 _h

Table : Status Codes

[Back to Top](#)

Accessible Resources

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	DateTime	01 _h	yes	no	no	yes	no	no	no	no
2	Calendar Events	02 _h	yes	yes	yes	yes	yes	yes	yes	yes
3	Application Status	03 _h	yes	no	no	no	no	no	no	no
4	reserved	04 _h	no	no	no	no	no	no	no	no
5	Extras	05 _h	yes	no	no	yes	no	no	no	no
6	WM-Bus Device Filter Items	06 _h	yes	yes	yes	yes	yes	yes	yes	yes

Table : Accessible Resources

[Back to Top](#)



Accessible Resources - DateTime

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) _h 5F649E19 _h = 1.600.429.593 seconds since 01.01.1970, 00:00:00 "2020-09-18 11:46:33"

Table : DateTime Resource

Example Set DateTime

< Set > < Resource ID > < DateTime >

< 07_h > < 01_h > < (19 9E 64 5F)_h >

Back to [Accessible Resources](#)

Accessible Resources - 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 _h = Receive and record in C/T-Mode see Application Events	FF _h = always	03h = Daily	00h	(19 9E 64 5F) _h see DateTime

Table : 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_h) 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_h > < 01_h > < (41 FF 03 00 19 9E 64 5F)_h >

Back to [Accessible Resources](#)

Accessible Resources - Application Events

The following table lists all application events

Event Name	Event ID	via Calendar	via HCI Interface	Description
None	00 _h	no	no	Invalid event
UI Events				
Show Status	01 _h	yes	yes	Output of internal status on LED
Push Button	02 _h	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 _h	yes	yes	Set LED off
LED Red	04 _h	yes	yes	Set LED red color
LED Green	05 _h	yes	yes	Set LED green color
LED Yellow	06 _h	yes	yes	Set LED yellow color
LED Red Blinking	07 _h	yes	yes	Set LED red blinking
LED Green Blining	08 _h	yes	yes	Set LED green blinking
LED Yellow Blinking	09 _h	yes	yes	Set LED yellow blinking
LoRaWAN Events				
LoRaWAN Activate	20 _h	not recommended	yes	Activate LoRaWAN Stack per OTAA or ABP
LoRaWAN Deactivate	21 _h	not recommended	yes	Deactivate LoRAWAN Stack
System Events				
Get Network Time	30 _h	yes	yes	Request system time via LoRaWAN MAC command. On response the system time will be synchronized.
Send Application Status	31 _h	yes	yes	Transmit Application Status via LoRaWAN
WM-Bus Events				
Receive in S-Mode and record	40 _h	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 _h	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 _h	yes	yes	Disable receiver
Start Upload	43 _h	yes	yes	Disable receiver and start upload of stored WM-Bus messages via LoRaWAN
Receive S-Mode and output via HCI	44 _h	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 _h	yes	yes	Enable receiver for Wireless M-Bus C/T-Mode, received messages will be forwarded via HCI
--------------------------------------	-----------------	-----	-----	--

Table : Application Events

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



Accessible Resources - Application Status

The Application Status includes the following information elements:

Resource	Application Status								Firmware V1.1 ff.	
Format	Date and Time (UTC)	Firmware Version	Last Sync Time	Reset Counter ¹⁾	Status Bits	WM-Bus Rx Counter ²⁾	WM-Bus Stored Counter ²⁾	WM-Bus Tx Counter ²⁾	Battery Voltage in mV	Firmware Type
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
Example	see DateTime	e.g. (07 01) h V1.7	see DateTime						z.B. 3450 mV	e.g. 0 = Release

- **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 ¹⁾**
Contains the number of device resets
- **Status Bits**
This field includes several Status Bits:
 - Bit 0 : 1 = LoRaWAN Stack is not activated
 - Bit 1 : 1 = Network Time is not synchronized
 - Bit 2 : 1 = System Time is not synchronized
 - Bit 3 : 1 = LoRaWAN Activation Procedure is ongoing
 - Bit 4 : 1 = LoRa Configuration is invalid
 - Bit 5 : 1 = WM-Bus Device Filter list is empty
 - Bit 6 : 1 = Calendar event list is empty
 - Bit 7 : Reserved
 - Bit 8 : 1 = Flash Memory full condition detected
 - Bit 9 : 1 = Flash Memory CRC error detected
- **WM-Bus Rx Counter ²⁾**
Total received WM-Bus packets before any packet filtering since last counter reset
- **WM-Bus Stored Counter ²⁾**
Number of stored WM-Bus packets after packet filtering
- **WM-Bus Tx Counter ²⁾**
Number of uploaded WM-Bus packets
- **Battery Voltage ³⁾**
The battery voltage is measured just before transmitting this status message. The value is returned in Millivolts.
- **Firmware Type ³⁾**
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

Info

- 1) The Reset Counter is copied to the non-volatile memory earliest 30 seconds after system start.
- 2) 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 an [HCI message](#).
- 3) Battery Voltage and Firmware Type are available in Firmware V1.1

[Back to Accessible Resources](#)

Accessible Resources - 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
Exampel	< 03 00 00 00 > _h

- **Options Bits**

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 1.1)

- 0 = disabled
- 1 = enabled

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

Bit 6 - 31 : reserved for future

Back to [Accessible Resources](#)

Accessible Resources - WM-Bus Device Filter Items

A single WM-Bus Device Filter Item includes the following information elements:

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

Note

¹⁾ 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_h) is reserved and means that this filter item should be applied always, independently of the configuration of an [Calendar Event](#)

[Back to WM-Bus Range Extender Services](#)

