IMST GmbH

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

iOKE868_LoRaWAN

AN036 - Remote Access Protocol

Version 1.0

Document State

final

Date

April 2023

Document ID

4000/40140/0165

History

Version	Datum	Comment
1.0	19.04.2023	Initial Version

Aim of this document

This document includes a description of the Remote Access Protocol which is supported by iO881A for LoRaWAN. This protocol enables applications to configure the infrared reading unit 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 **L**east **S**ignificant **B**yte first unless explicitly noted

Content

Remote Access Services - Overview

LoRaWAN end nodes like iO881A 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 specifc 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 >

< 04h > < 02h > < 04h >

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 exits 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 > < Index: 4 > < 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:

> 1. Unknown Service Request Codes will will be answered with a Status Response, corresponding Resource ID and Status Code "Service not available".
> 2. Service Requests for unkown Resources IDs will be answered with a corresponding Reponse and Status Code "Resource not found".

#	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
5	Service not available	04_h

Table : Status Codes

Back to Top

#	Status	Code
1	Success	00_h

Accessible Resources

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

#	Resource	Resource ID	Services							
			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	no	yes	yes	no	yes	yes	yes	yes
3	Application Status	03_h	yes	no	no	no	no	no	no	no
4	OBIS ID Filter	07_h	no	yes	yes	no	yes	yes	yes	yes
5	Reader Configuation	$0A_h$	yes	no	no	yes	no	no	no	no

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	16 Bi, LSB first	8 Bit	8 Bit	32 Bit, LSB first
Example	see Application Events	(01 00)$_h$: 0001$_h$ => Group 1 (02 00)$_h$: 0002$_h$ => Group 2 (04 00)$_h$: 0004$_h$ => Group 3 (08 00)$_h$: 0008$_h$ => Group 4 (09 00)$_h$: 0009$_h$ => Group 1 + Group 4 (06 00)$_h$: 0006$_h$ => Group 2 + Group 3	03$_h$ = Daily	00$_h$	(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 OBIS ID Filters. It defines which OBIS-ID Filter Groups are applied during the meter reading process.
  The Filter Group-IDs are interpreted bitwise and can be logically ored wihtin these Calender Events.
  Note: The value FFFF$_h$ is reserved and means that all configured OBIS-ID Filter Groups should be applied.

- **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), Repetion Step Size = 2 => Repetiton Interval = every 2 + 1 = 3 hours
  Example 2: Repetition Type = 3 (Daily), Repetion Step Size = 0 => Repetiton 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

Request:

< Add Sub Item > < Calendar Event > < Calendar Event Sub Item Data: Single Read, Filter Group 1, Daily, first schedule 2020-09-18 11:46:33 >

< 0B$_h$ > < 02$_h$ > < (51 01 00 03 00 19 9E 64 5F)$_h$ >

Response:

< Add Sub Item Response > < Calendar Event > < Index: 1 > < Success >

< 0Ch > < 02$_h$ > < 01$_h$ > < < 00$_h$ >

Back to Accessible Resources

Accessible Resources - Application Events

The following table lists all application events, which should be used for remote configuration of calendar events

Event Name	Event ID	Description
System Events		
Get Network Time	30_h	Request system time via LoRaWAN MAC command. On response the system time will be synchronized.
Send Application Status	31_h	Transmit Application Status via LoRaWAN
Meter Reader Events		
Reader Off	50_h	Disables the optical reader unit
Single Reading Request	51_h	Enables the optical reading unit for a single reading procedure
Continuous Reading Request	52_h	Enables the optical reading unit for a continuous reading procedure
Scan OBIS-IDs Request	53_h	Enables the optical reading unit for an OBIS-ID scan procedure

Table : Application Events

Back to Accessible Resources - Calendar Events

Accessible Resources - Application Status

The Application Status includes the following information elements:

Resource	Application Status							
Format	Date and Time (UTC)	Firmware Version	Last Sync Time	Reset Counter[1]	Status Bits	Reader Rx Counter[2]	Reader Error Counter[2]	Reader Tx Counter[2]
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
Example	see DateTime	e.g. (07 01)$_h$ V1.7	see DateTime					

- **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 sychronization 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 : reserved

  Bit 6 : 1 = Calendar event list is empty

  Bit 7 : 1 = OBIS ID filter List is empty

- **Reader Rx Counter [2]**
  Number of succesful received meter files

- **Reader Error Counter [2]**
  Number of failed meter readings

- **Reader Tx Counter [2]**
  Number of forwarded meter readings

Info

[1] The Reset Counter is copied to the non-volatile memory earliest 30 seconds after system start.

[2] The Reader Rx7Tx/Error 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.

Back to Accessible Resources

Accessible Resources - OBIS-ID Filter Items

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

Resource	OBIS-ID Filter Item							
Format	OBIS-ID Groups						OBIS-ID Mask	Filter Group-ID
	Group A	Group B	Group C	Group D	Group E	Group F		
Octets	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 bit	8 Bit	16 Bit, LSB first
Example 1	1	0	1	8	0	255	FC_h => all 6 Groups applied	(01 00)$_h$
Example 2	0	0	1	8	0	0	30_h => only Group C,D	(01 00)$_h$

- **OBID-ID Groups**
  A sequence of 6 bytes in total which identifies the requested OBIS-ID to read.

- **OBID-ID Mask**
  This mask defines which of the OBIS-ID Group fields are used for comparison with every decoded meter record.
  Bit 7 = Group A
  Bit 6 = Group B
  Bit 5 = Group C
  Bit 4 = Group D
  Bit 3 = Group E
  Bit 2 = Group F
  Bit 1 .. 0 = reserved

- **Filter Group-ID**
  This element is only used to group several filter items per Calender Event. Filter Group-IDs are interpreted bitwise and can be logically ored in a Calender Event (see Example Accessible Resources - Calendar Events).
  Note: The value $FFFF_h$ is reserved and means that this filter item should be always applied.

Example: Add OBIS ID Filter Item

Request

< Add Sub Item > < OBIS-ID Filter > < OBIS-ID Filter Item Data >

< $0B_h$ > < 07_h > < (01 00 01 08 00 FF FC 01 00)$_h$ >

Response

< Add Sub Item Response > < OBIS-ID Filter > < Index: 1 > < Success >

< $0C_h$ > < 07_h > < 01_h > < 00_h >

Back to Accessible Resources

Accessible Resources - Reader Configuration

This resource provides some configuration parameters which control the behaviour of the optical reader unit and the radio protocol payload format which is decribed in document **iOKE868_LoRaWAN_AN035_RadioProtocol**.

Resource	Reader Configuration			
Format	Options Bits	Protocol	Baudrate	Paired Meter-ID (only for Get Service)
Octets	32 Bit, LSB first	8 Bit	8 Bit	n * 8 Bit
Example	(06 00 00 00)$_h$ => 0000 0006$_h$: include Meter ID in Meter Data Messages + Status Field	0 = Auto Detection	5 = 9600 bps	

- **Options Bits**
  This field includes several configuration bits:

  Bit 0 : reserved

  Bit 1 : Include optional Meter-ID Field in Meter Data Messages
      0 = Meter-ID Field not added
      1 = Meter-ID Field is added in uplink messages

  Bit 2 : Include optional Status Field in Meter Data Messages
      0 = Status Field not added
      1 = Status Field is added in uplink messages

  Bit 3 : Include optional Time Field in Meter Data Messages
      0 = Time Field not added
      1 = Time Field is added in uplink messages

  Bit 4 : Omit Unit Information in Meter Object Fields
      When it's possible to derive the unit by means of the OBIS-ID itself this option migh be helpful to reduce the uplink payload.
      0 = Unit Information Field is present in Meter Object Fields
      1 = Unit Information Field is not added in Meter Object Fields

  Bit 5: IEC Number Conversion
      IEC numbers are usually given in ASCII format which requires a single byte per digit.
      This configuation option enables a conversion function from ASCII to integer which helps to reduce the uplink payload.
      0 = Optimization off
      1 = Optimization on

  Bit 6 : IEC Wake-up Procedure (IEC Mode only)
      This option can be used to configure the fast wake-up procedure which is needed for battery powered meter devices which operate in IEC mode.
      0 = Fast Wake-up procedure off
      1 = Fast Wake-up procedure on

  Bits 7 ..31 : reserved

- **Reader Protocol**
  This parameter defines the applied reader protocol.
  0 = Auto Detection, the reader unit tries to detect one of the implemented uni- and bidirectional protocols
  1 = IEC Mode A (bidirectional, 300 bps)
  2 = IEC Mode B (bidirectional, 300 bps + dynamic baudrate change defined by meter)
  3 = IEC Mode C (bidirectional, 300 bps + dynamic baudrate change defined by meter)
  4 = IEC Mode D (unidirectional, baudrate manufacturer specific e.g.: 9600 bps)
  5 = reserved
  6 = SML (default, unidirectional, typical with baudrate of 9600 bps)

- **Baudrate**
  This parameter can be used to select a specific reader baudrate for meter operating in unidirectional IEC Mode D or SML.
  0 = 300 bps
  1 = 600 bps

2 = 1200 bps
3 = 2400 bps
4 = 4800 bps
5 = 9600 bps (default)
6 = 19200 bps
7 = 38400 bps
8 = 57600 bps
9 = 115200 bps

- **Paired Meter-ID**
  This information field is only present in the "Get Response" message. It includes the Meter-ID to which the reader is paired.

Back to Accessible Resources