Table of Contents

RM module work description

Jooby Development

- Message documentation
- Message encoders/decoders
- Message encoders/decoders GUI (Demo)
- API for interacting with Jobby devices. Other documentation

- Command and Parameter Examples

Beginning of work

It is necessary to know the module factory parameters to instal module in NS correctly (they are available in specification):

- Activation method in the LoraWAN network - **ABP** (Activation by personalization) or **OTAA** (Over the Air Activation)
- Keys **APPSKEY[16]**, **NETSKEY[16]** - **ABP** or **APPKEY[16]** - **OTAA**

Time and date

> ⚠️ The Radio module and server times need to be synchronized. It is necessary to correct the modulating time when it leaves the server. Possible deviation no more than 30 seconds.

Device time management

The time is transmitted as a 4-byte counter. This is the time in seconds since the year 2000. The module time must be synchronized with the server time. The time can be set using the commands: **SET_TIME** and **CORRECT_TIME**. The **SET_TIME** command is for initial time setting and for adjustments greater than +/-128 seconds. Adjusting the time backwards may result in the loss of hourly data. Further, the server must correct the module time if it deviates from the server time. The recommended allowable time deviation is no more than 30 seconds; if it is exceeded, the time is corrected using the **CORRECT_TIME** command.

Battery

Once a day, the module measures the battery parameters: battery voltage without load and battery voltage under resistor load (150 Ohm for modules on WLE). These data and the calculated internal resistance value are transmitted once a day also. Based on the measurement results, it is possible to start the battery depassivation process. In this case, the **DEPASS_DONE** event is logged.

Magnetic field sensor

A magnetic field sensor is installed in the module. This sensor is used to send current pulse counter to server. In case of short magnetic influence (more than 2 sec and less than 20 sec), the current pulse counter data **GET_CURRENT** will be sent to the server.

Activation/Deactivation

- [Installation. Activation](#)

The module is activated after connecting the **Jooby Splitter 1x4PU/1x2PU** to the module's port. The pulse counter in the module is continuously polled. Activation means that the module starts transmitting data frames via the radio channel.

Upon the occurrence of the **INSERT** and **ACTIVATE** events, the module will send **NEW_EVENT** frames containing the **INSERT** and **ACTIVATE** events and the corresponding event timestamps. The **NEW_EVENT** frame will also include a **CURRENT** message containing the current pulse counter data.

If the **Jooby Splitter 1x4PU/1x2PU** is disconnected from the module, the module will be deactivated after 30 seconds. A **NEW_EVENT** frame will be sent to the Network Server with the **REMOVE** and **DEACTIVATE** events, along with the **CURRENT** data. The event time is transmitted as part of the event data.

Consumption data

Once activated, the module sends consumption data regularly. By default, the module transmits hourly consumption data at an interval of about 4 hours + a pseudo-random component ranging from 0 to 17 minutes. The frequency of sending data can be changed, but the period cannot be less than 10 minutes and more than 36 hours. You can set the type of data to be transferred. It is possible to transfer current, hourly, daily data. It is also possible to transmit hourly and daily data simultaneously (in one frame). To select the type of data to be transmitted, use parameter 5 (Type of data to be output). Hourly data is stored in the archive, the maximum archive depth is 6 months. Daily consumption data is also stored in the archive, the maximum depth is 2 years. The depth of the archives does not change configurationally.

Absolute data

The device supports Absolute data mode. In Absolute data you can put real meters value.

First of all you have to setup and enable Absolute Data on module with commands:

Example:

- **Setup absolute data** - `030a17000028c28200000b072f`. That command setup absolute data with next fields

```
"id": 23,
"value": 2823,
"meterValue": 10434,
"pulseCoefficient": 10
```

`id` - Parametr ID

`value` - Counter value on module. You can get it with command 070052, or module send it after activate/deactivate events

`meterValue` - Real meter value on module multiplied by pulseCoefficient. In that example the meter has readings 104,34 cubic meters

`pulseCoefficient` - Pulse Coefficient of meter. If pulse coefficient of meter is 0,01 then that valuer should be 10, if pulse coefficient of meter is 0,1 then that valuer should be 100

- **After setup you have to enable Absolute Data with command** - `030218014d`

In response you will get `0302170142` - Means "Absolut Data" enabled

For example module get 3 impulse. With command `1f0f0045` possible to get current Absolute Data. in response will be `1803018a161f0f040182c551d0`.

Current counter value of module "CurrentMC"

```
"value": 2826,
```

```
"index": 1
```

and Absolute Data "ExAbsCurrentMC"

```
"pulseCoefficient": 10,
"value": 10437,
"index": 1
```

Here we can see Meter Readings 10437. That means 104,37 cubic meters

Module send Absolute Data in Reporting Data Interval

Events

Device events

All events are recorded in the archive. Archive capacity - not less than 256 events and not more than 512 events. The event log records the time of the event, the event ID, and the **LAST_EVENT** sequence number. The event number is incremented with each entry. The event number is passed to the server with the **LAST_EVENTS** command, which is added to every UPLINK frame sent without a request. There is no priority for messages in the archive. Events are recorded in a cycle. Archives can be requested from the most recent to the oldest and from a specific time.

Commands

There are 2 types of command direction:

- downlink - request (sent from server to device)
- uplink - response or event (sent from device to server)

  - Downlink commands
  - Uplink commands

Parameter types

Description of parameter types

From:
https://wiki.jooby.eu/ -

Permanent link:
https://wiki.jooby.eu/playground/rm_description_commands_parameters

Last update: **2024/10/22 14:41**