# CMi4140

User manual

elvaco

1.7 Edition

# Table of Contents

# 1. About this manual

This manual covers information needed to mount, install, configure, and use the product. It is intended for installers and system integrators.

To download the latest version of this User manual, visit the Elvaco website, https://www.elvaco.com. There you will also find information about Elvaco's other products and services.

## 1.1. Symbols

The following symbols are used throughout the manual to emphasize important information and useful tips:

**Warning**
Indicates a potentially dangerous situation that could result in severe injuries or serious equipment damage.

**Caution**
Indicates a potentially dangerous situation that could result in minor injuries or equipment damage.

**Note**
Indicates information that is important to take into consideration for safety reasons or to assure correct operation of the product.

**Tip**
Indicates information intended to help you get the most out of your product. For example, it can be used to highlight a possible customization option related to the current section.

## 1.2. Terms and abbreviations

| Abbreviation | Description |
|---|---|
| DIB | Data Information Block |
| DIF | Data Information Field |
| VIF | Value Information Field |
| MCM | Meter Connectivity Module |

### Number representation

- Decimal numbers are represented as a normal number, i.e. 10 (ten)
- Hexadecimal numbers are represented with the prefix 0x, i.e. 0x0A (ten)
- Binary numbers are represented with the prefix 0b, i.e. 0b00001010 (ten)

### European standards

M-Bus standard EN 13757-3:2013: Communication systems for meters and remote reading of meters – Part 3: Dedicated application layer.

# 2. Safety

The following safety precautions must be observed during all phases of the operation, usage, service, or repair of the product. Users of the product are advised to convey the following safety information to users and operating personnel and to incorporate these guidelines into all manuals supplied with the product. Failure to comply with these precautions violates safety standards of design, manufacture and intended use of the product. Elvaco AB assumes no liability for customer's failure to comply with these precautions.

⚠ **Caution**
This is an electrostatic-sensitive product. Observe the necessary ESD protective measures when installing the product.

ⓘ **Note**
The product receives and transmits radio frequency energy while switched on. Remember that interference can occur if the product is used close to TV sets, radios, computers or inadequately shielded equipment. Follow any special regulations and always switch off the product wherever forbidden, or when you suspect that it may cause interference or danger.

ⓘ **Note**
Waste electrical products should not be disposed of with household waste. Please recycle where facilities exist. Contact your Local Authority for recycling advice.

# 3. Product overview

## 3.1. Application description

CMi4140 is a meter connectivity module mounted inside a Kamstrup MULTICAL® 403/603/803 heating/cooling meter to deliver meter data to a server via a LoRaWAN® network. The module is energy-efficient and offers 11 years of battery lifetime with EcoMode enabled. It can be retrofitted into deployed meters and is ideal for applications where long-range wireless communication is required. CMi4140 is easily configured via the Elvaco OTC mobile app, or via downlink commands. With CMi4140, Elvaco offers a meter connectivity module designed for customers that require a user-friendly and cost-effective solution.

## 3.2. Features

CMi4140 offers a combination of battery operation with very long lifetime and versatile use through its many configuration options. Key features of the module include:

- **LoRaWAN Certified$^{CM}$**
  The CMi4140 is certified according to LoRa Alliance®.
  For more information, see https://lora-alliance.org/.
- **Extensive battery lifetime**
  The module's EcoMode feature enables the module to achieve a battery lifetime of at least 11 years of operation plus one year of storage.
- **No manual configuration needed**
  As soon as the meter connectivity module is installed and activated, it will attempt to join a LoRaWAN® network and start delivering meter data.
- **Easy and secure commissioning**
  With Elvaco's One-Touch Commissioning (OTC), deployment, configuration and key transfer can be performed in a secure and flexible way. Use the Elvaco OTC App to enter your desired settings and place your phone next to the meter. New settings will be applied via NFC.
- **Unique and flexible message schemes**
  CMi4140 comes with several different message formats to choose from. This makes it easy to customize the product to the demands of your application.

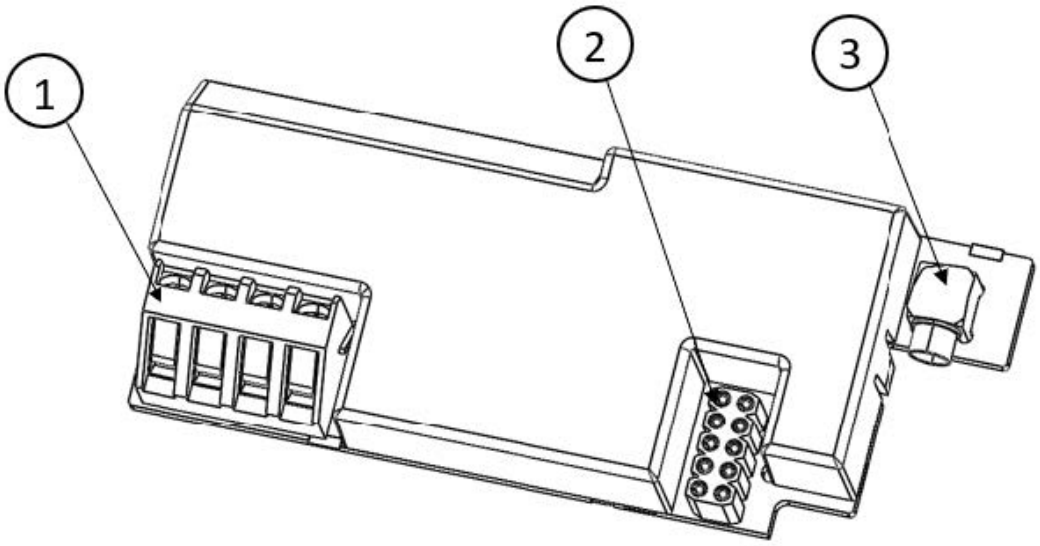
## 3.3. Meter compatibility

CMi4140 is compatible with Kamstrup MULTICAL® 403, MULTICAL® 603 and MULTICAL® 803 heating/cooling meters with a compatible firmware version. Once the module is mounted and activated, a compatibility check is made to ensure that it is compatible with the meter firmware. If the module is mounted in an incompatible meter, this is indicated in the Elvaco OTC App. The table below lists all MULTICAL® meter types that CMi4140 is compatible with.

Table 1. Meter type compatibility

| Meter type | Multical® 403 | Multical® 603 | Multical® 803 |
|---|---|---|---|
| Heat energy | ✓ | ✓ | ✓ |
| Heat/Cool energy | ✓ | ✓ | ✓ |
| Cool energy | ✓ | ✓ | ✓ |
| Volume (cold) | ✓ | ✓ | ✓ |
| Volume (hot) | ✓ | ✓ | ✓ |

## 3.4. Parts overview




1. Pulse inputs In A and In B, used to connect adjacent pulse meters
2. Power connector
3. MCX antenna connector

**Note**
The MCX antenna connector is used to connect either an internal or external antenna.

# 4. Installation

## 4.1. Mounting the product

CMi4140 can be used in a Kamstrup MULTICAL® 403, MULTICAL® 603, or a MULTICAL® 803 heating/cooling meter. If the module is not mounted in the meter upon delivery, it is easily installed:

> ⚠ **Caution**
> This is an electrostatic-sensitive product. Observe the necessary ESD protective measures when installing the product.

1. **Open the meter housing.**
2. **Apply ESD precautions.** Make sure you are grounded (e.g., by touching the meter's metal pipe or a metal part which is electrically connected to it).
3. **Install the device.** Hold the device by the outer edges and gently press it into position in an available slot in the meter.
4. **Attach the LoRaWAN® antenna**. Route the antenna cable using the guiding pins in the meter housing (if using an external antenna). Connect the antenna to the MCX connector on the module (applies to both the internal and external antennas).

   > ⚠ **Caution**
   > When using an external antenna, make sure to mount it at least 0,5 meters away from the meter to avoid interference.

5. **Close the lid of the meter.** When closed, the device will receive power from the meter

## 4.2. Connecting the LoRaWAN® antenna

CMi4140 supports either an internal or external LoRaWAN® antenna. Since both antennas use the MCX connector, they are interchangeable.

1. Connect the internal or external antenna with the MCX connector to the module in the meter.
2. Route the antenna cable (if using an external antenna).
3. Push the connector gently into the socket on the module.

> ⚠ **Caution**
> When using an external antenna, make sure to mount it at least 0,5 meters away from the meter to avoid interference.

## 4.3. Activating the product

> ⓘ **Note**
> Before activating the device and trying to join a LoRaWAN® network, make sure the LoRaWAN® network server in use is prepared with necessary device- and security-related information. Security keys for Elvaco devices can be accessed in Elvaco EVO, Elvaco EVO Login.

Upon delivery, the device is set to inactive mode, which means no messages will be transmitted until it is activated. It can be activated in two ways: via the buttons on the meter or via the Elvaco OTC App:

### Activation via the meter buttons

- **On MULTICAL® 403**, press and hold both buttons on the front of the meter until "CALL" is displayed.
- **On MULTICAL® 603 and MULTICAL® 803**, press and hold the two arrow buttons on the front of the meter until "CALL" is displayed.

### Activation via the Elvaco OTC App

1. Open the Elvaco OTC app (available for Android and iOS).
2. Scan the device (make sure NFC is enabled on the phone).
3. Go to **Apply mode**.
4. Set the power mode to **Active**.
5. Select **Apply settings**.
6. Scan the device to apply new settings.
   To verify that it has been activated, go to **Inspect**, scan the device, and make sure that power mode is set to **Active**.

**NFC Scanning Position**
**MULTICAL® 403**: Place the phone in the center of the raised section on the back of the meter.

**MULTICAL® 603**: The meter has two slots:

- Left slot: Middle of the raised back section
- Right slot: Right side of the raised back section

**MULTICAL® 803:** Open the meter and place the phone close to the device.

## 4.4. LoRaWAN® end-device activation

Before being able to transmit data over a LoRaWAN® network, the device must be personalized and activated. This can be done in two ways: Over-the-air activation (OTAA) or Activation by Personalization (ABP). For security reasons, Elvaco strongly recommends using OTAA, where all network keys are generated each time the device joins the LoRaWAN® network. In contrast, for ABP, all keys are set manually and stay constant over time. The tables below list the required information depending on the activation mode.

### In OTAA mode

| | |
|---|---|
| Device EUI | 16-digit unique identification number (not configurable). |
| Application key | The application key of each device is generated by Elvaco and used in OTAA mode to generate network keys when it joins the LoRaWAN® network. |
| JoinEUI | Sets the identification number of the join server. The identification number is set to a default value in all devices. The default value is presented in Elvaco OTC configuration options. |

### In ABP mode

| | |
|---|---|
| Device EUI | 16-digit unique identification number (not configurable). |
| Application session key | Used for payload encryption and decryption. |
| Network session key | Used by both the device and the network server. |

## 4.5. Reading the module status via meter display

In addition to the Elvaco OTC App, the current status of the module can be read via the meter display. To retrieve the status of the module via the meter display, enter the tech loop of the meter display and request the module status information. For detailed information, see the Kamstrup MULTICAL® Technical description.

The following table provides a description of how the current status of the module is read from the meter display.

As an example, if the display shows “5311111”, that means that the module has been activated and has successfully joined the LoRaWAN® network.

Table 2. Module status indication on meter display

| Character, MSB -> LSB | Status |
|---|---|
| 1-2 | 53 = Module identification number (CMi4140) |

| Character, MSB -> LSB | Status |
|---|---|
| 3 | 1 = All following characters are set to “1”<br>0 = One or more of the following characters are set to “0” |
| 4 | 1 = Meter has granted active mode for the module<br>0 = Meter has requested the module to be idle |
| 5 | 1 = Module has joined the LoRaWAN® network<br>0 = Module has not joined the LoRaWAN® network |
| 6 | 1 = Module SuperCap has been charged<br>0 = Module SuperCap is charging |
| 7 | 1 = Module is active<br>0 = Module is inactive |

# 5. Operation

## 5.1. Configuration

CMi4140 can be configured in two ways, either via the Elvaco OTC App, or via downlink commands. Configuration via the Elvaco OTC App is highly flexible, and all configuration options available in the module can be set. In contrast, downlink configuration comes with some limitations, as not all options are available. For a detailed view of which options are available, see Downlink configuration options [16].

**Important**
Before relying on downlink configuration, make sure you fully understand its accessibility limitations and restricted option set.

### Via Elvaco OTC app

The module is configured via the Elvaco OTC app. The app uses NFC to transfer settings to the module. It is compatible with iPhone or Android phones with Android 5.0 or later. The OTC app can be downloaded from Google Play for Android and App Store for iOS.

1. Open the Elvaco OTC app on your smartphone.
2. Place your smartphone close to the device and press **Scan**.
3. After a successful scan, the current settings will be displayed on the **Inspect** tab.
4. To change settings, go to the **Apply** tab. Use the toggle button to activate an option, and make your changes.
5. When done with changes press **Apply** at the bottom of the screen, then apply settings. Repeat step 2 to change the settings in the device.
   A successful change is acknowledged by a vibration.

**Tip**
Ensure secure and stable commissioning by using the profiling feature in the Elvaco OTC App. Create a configuration profile once and effortlessly apply it to all your devices, ensuring consistency and minimizing errors.

**Tip**

- If the module is mounted inside a MULTICAL® 403, the NFC field is easiest to access by placing your phone in the center of the raised section on the back.
- MULTICAL® 603 has two module slots. If mounted in the left module slot, the NFC field is easiest to access by placing your phone in the center of the raised section on the back. If the module is mounted in the right module slot, place your phone to the right of the higher part of the back.
- If the module is mounted in MULTICAL® 803, the meter needs to be open to access the NFC field. Place your phone close to the CMi4140 when making the NFC scan.

### Via downlink

The product supports configuration via downlink, i.e. sending commands to an end-device via the LoRaWAN® network. Communication via downlink is sent on LoRa port 2 and can only be made in a short window after an uplink transmission from module to server. Therefore, time-critical communication should not be performed over downlink.

Downlink commands are structured according to the following format: *“0x00” "TLV" "Number of bytes in configuration" "Configuration"*.

**Note**
The downlink feature should only be used sparingly due to bandwidth considerations.

## 5.2. Transmit interval

The transmit interval is used to set how frequently the module should transmit data on the LoRaWAN® network. The parameter can be set to a value between 10 and 1440 minutes (i.e. from 10 minutes to once per day). It is possible to fix the transmit interval, but for battery-operated devices where battery lifetime is critical, Elvaco recommends using the EcoMode feature, which adapts the transmit interval depending on prevailing radio conditions.

### 5.2.1. Transmit interval using EcoMode

When EcoMode is active, a battery lifetime of at least 11+1 years is guaranteed for the module. The module is able to achieve this by using a table of allowed transmit interval settings for each data rate. When radio conditions are poor (and data rate is low), the module will be able to send data less frequently in order to conserve battery life. When signal conditions are good, the module will be able to send data more frequently. When EcoMode is enabled, the module will continuously check if the set transmit interval is “allowed” by the EcoMode table. If a longer transmit interval is needed for the specific data rate in order to achieve 11 years of battery life, the module will adjust the parameter accordingly.

The following table shows the transmit intervals that the module will use for different data rates in order to achieve an 11-year battery life.

Table 3. Transmit intervals for different data rates in EcoMode

| Data rate | Transmit interval |
|---|---|
| DR0 | 180 min |
| DR1 | 120 min |
| DR2 | 60 min |
| DR3 | 30 min |
| DR4-DR5 | 15 min |

The following table shows the transmit interval for different data rates when using a message format with two telegrams, such as the Scheduled Extended+. Sending two telegrams instead of one will double the possible transmit interval using EcoMode.

Table 4. Transmit intervals for different data rates in EcoMode using a two-telegram message format

| Data rate | Transmit interval |
|---|---|
| DR0 | 360 min |
| DR1 | 240 min |
| DR2 | 120 min |
| DR3 | 60 min |
| DR4-DR5 | 30 min |

**Note**
For applications where battery lifetime is critical, EcoMode should be enabled. If EcoMode has been disabled, battery life guarantees no longer apply, even if EcoMode is activated later on.

## 5.3. Time handling

The module relies on the meter’s clock to keep time. The meter time is assumed to be standard local time (no DST). When synchronizing time in the meter using the Elvaco OTC App, local standard time is always used, even if DST is in effect. The timestamped meter data sent from the module can be adjusted to be sent in UTC by specifying the “UTC offset” configuration parameter. The UTC offset will be subtracted from the timestamp prior to transmission. If the meter is in Sweden, which uses CET (Central European Time), it should have UTC offset set to +60 (+1h). In this case at time 12:00 a telegram is sent with timestamp 11:00 as this is the corresponding UTC time. A meter in New York (USA) should have a UTC offset of "-300" (-5h) etc. A UTC offset of "0" means the meter time is used as-is.

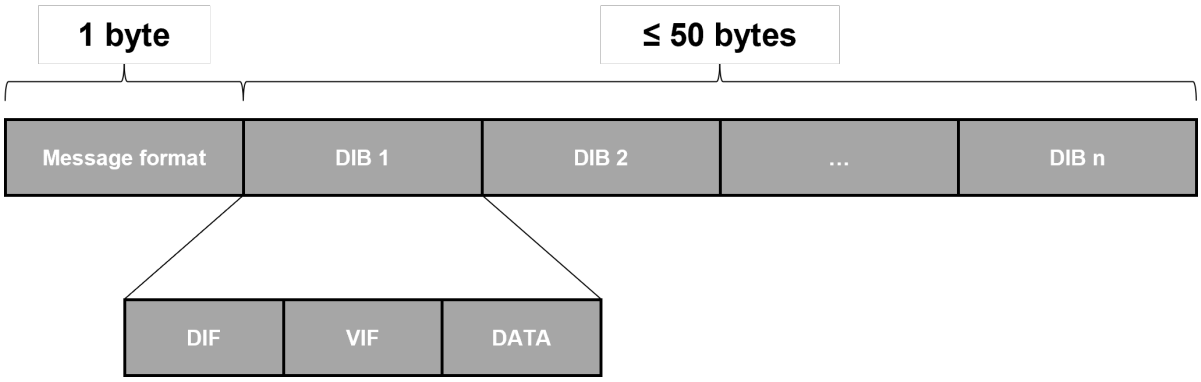
If the meter is set to use DST this is ignored by the module and the standard time is used. Thus, the time on the meter’s display may not match the time in the telegram or in the Elvaco OTC App.

## 5.4. Adaptive data rate (ADR)

ADR is part of the LoRaWAN® standard where the network server determines the optimal rate for the module based on current signal conditions. In the best radio conditions, the module will use its highest data rate (DR5) in order to be as energy-efficient as possible. When signal conditions are poor, the network server will incrementally lower the data rate until it is able to receive the message. When the data rate is low, the energy consumption per telegram will increase.

## 5.5. Message encoding

All message formats encoded to M-Bus standard will have the following structure. Each telegram begins with one byte specifying the message format. Then follows a sequence of data information blocks (DIBs). The data and structure of the DIBs depends on the message type set. Each DIB contains a data information field (DIF), a value information field (VIF) and a data field (DATA), where the actual payload is stored.




*M-Bus message structure*

> **Note**
> When using the message format JSON, the data will be presented as plain text, and not according to the M-Bus structure as presented above.

## 5.6. Security and access control

The product has a configuration lock feature, which prevents unauthorized access to the module. When the configuration lock has been enabled, a device-specific Product Access Key (PAK) will be needed to access the device. Keys are managed securely using Elvaco's OTC solution which includes the mobile application for configuration.

> **Note**
> For more information about security and access control for the product, refer to the One-Touch Commissioning (OTC) documentation, available on the Elvaco website.

## 5.7. Reset procedures

### 5.7.1. Rebooting the module

The module can be rebooted using the Elvaco OTC App:

**Via the Elvaco OTC app**

1. Open the Elvaco OTC app.
2. Scan the module.
3. Go to **Apply mode**.
4. Enable the **reboot switch** and **Apply changes**.

### 5.7.2. Switching off the module

To switch off the module, use the Elvaco OTC app.

1. Open the Elvaco OTC app.
2. Scan the module.
3. Go to **Apply mode**.
4. Set **Power mode** to Inactive and then **Apply changes**.

> **Note**
> This function is restricted to the registered owner of the product in Elvaco OTC App and will not be visible/available for other users.

## 5.8. Meter communication error messages

Payloads using M-Bus encoded data use the Function field of the DIF to indicate errors. In this case, it is set to “value during error state” (M-Bus standard EN 13757-3:2013) and the value sent should not be used. A typical case for this is when the module is unable to communicate with the meter and retrieve meter values, in which case all the fields in the payload have the DIF indicating “value during error state”. In case of erroneous data or if the module is unable to communicate with the meter, bits 4–5 of the DIF code (first byte of each index field) will be set to 11b.

Example 1. Example

A DIF code of 0x02 will be set to 0x32 in case of error. (This applies to all message formats except JSON.) For message format JSON, the actual value will be replaced with “null” in case of error state. If no meter communication is possible at all, all fields have this error indication set.

# 6. Configuration options

## 6.1. Elvaco OTC App configuration options

The table below lists all settings that can be made using the Elvaco OTC App. If the device is locked, and the app user does not have access to the Product Access Key (PAK), available settings are only readable.

Table 5. All configuration options

| Field name (Abbr.) | Description | Default value | Locked device & correct PAK or open device | Locked device & no PAK | Downlink |
|---|---|---|---|---|---|
| Meter ID | Meter identification number of the meter. Not configurable. | N/A | Readable | Readable | N/A |
| Power mode | Used to activate/deactivate the module. | Inactive | Readable / Writable | Readable | N/A |
| Message format | The message format determines the structure and payload of the telegram sent from the module. | 0x15 (Standard) | Readable / Writable | Readable | Writable |
| EcoMode | When activated, 11+1 years of battery life is guaranteed by adapting the transmit interval of the module to current signal conditions (using Kamstrup's D-cell meter battery). | On | Readable / Writable | Readable | Writable |
| Transmit interval | Sets the number of minutes between each transmission from the module | 60 min | Readable / Writable | Readable | Writable |
| Date & Time | Date and time set for the meter. | N/A | Readable | Readable | N/A |
| Set absolute time | Sets the time of the meter. | N/A | Writable | N/A | N/A |
| Set Time Relative | Adjusts the time of the meter relative to the current time | N/A | N/A | N/A | Writable |
| Set UTC offset | Sets the UTC offset of the meter. | N/A | Readable / Writable | Readable | Writable |
| Configuration Lock | Locks the module to prevent unauthorized access. | Open | Readable / Writable | Readable | Writable |
| **LoRaWAN® settings** | | | | | |
| Device EUI | Unique module identification number. Not configurable. | 94193A010BXXXXXX (Retrofitted devices)<br>94193A0111XXXXXX (Factory-mounted devices) | Readable | Readable | N/A |
| Activation type | Sets the way the device joins the LoRaWAN® network. | OTAA | Readable / Writable | Readable | N/A |
| Network join | Displays whether the module has joined the LoRaWAN® network or not. | N/A | Readable | Readable | N/A |
| JoinEUI | Identifies the device's OTAA join context. | 94193A030B000001 (Retrofitted devices)<br>94193A0311000001 (Factory-mounted devices) | Readable / Writable | Readable | N/A |
| Device address | 32-bit address used by the module to identify itself on the LoRaWAN® network. | N/A | Readable / Writable | N/A | N/A |

| Field name (Abbr.) | Description | Default value | Locked device & correct PAK or open device | Locked device & no PAK | Downlink |
|---|---|---|---|---|---|
| Current data rate | The current data rate used for the module. | N/A | Readable | Readable | N/A |

## 6.2. Downlink configuration options

The table below lists the settings that can be configured using LoRaWAN® downlink commands.

Table 6. Downlink commands

| Field name | TLV | Number of bytes in configuration | Configuration | Example |
|---|---|---|---|---|
| Configuration lock | 0x05 | 0x01 | 0x00 = Locked<br>0x01 = Open | 0x00050101<br>(Disables the configuration lock) |
| Transmit interval | 0x06 | 0x02 | 0xNumber of minutes between transmission<br>(lsByte -> msByte) | 0x0006021E00<br>(Sets the Tx interval to 30 minutes) |
| Message format | 0x07 | 0x01 | 0x15 = Message format Standard<br>0x16 = Message format Compact<br>0x17 = Message format JSON<br>0x18 = Message format Scheduled-daily redundant<br>0x19 = Message format Scheduled-Extended<br>0x1A = Message format Combined heat/cooling<br>0x1B = Message format Heat Intelligence<br>0x3B* = Message format Scheduled Extended+<br>0x1C* = Message format Pulse<br>0x4D* = Message format Pulse Extended<br>**Note**<br>*Message formats built up by 2 telegrams. only the first Message ID should be included in the downlink command) | 0x00070116<br>(Sets the message format to compact) |
| EcoMode | 0x0F | 0x01 | 0x00 = Disable EcoMode<br>0x01 = Enable EcoMode | 0x000F0100<br>(Disables EcoMode) |
| Set Time Relative | 0x13 | 0x04 | 0xNumber of seconds* (lsByte -> msByte)<br>*Negative numbers supported. | 0x0013043C000000<br>(Adds 60 seconds to the current time)<br>0x0013043C000080<br>(Subtracts 60 seconds from the current time) |
| UTC offset | 0x17 | 0x02 | 0xNumber of minutes*<br>(lsByte -> msByte)<br>*Negative numbers supported. | 0x17023C00<br>(Sets the UTC offset to +60 minutes)<br>0x17023C80<br>(Sets the UTC offset to -60 minutes) |
| Reboot | 0x22 | 0x02 | 0x759E is used to reboot device (note endianness, LSB first). | 0x0022029E75<br>(Reboots the device) |

## 6.3. Message formats

To allow a flexible output from the device, CMi4140 has several message formats. Some are scheduled, meaning the meter is read at the top of every hour, and a clock message is sent once per day, see Scheduled message formats [20].

Table 7. CMi4140 message formats

| Message format name | Message ID | Scheduled | Introduced in FW |
|---|---|---|---|
| Standard | 0x15 | No | 1.0.1 |
| Compact | 0x16 | No | 1.0.1 |
| JSON | 0x17 | No | 1.0.1 |
| Scheduled Daily Redundant | 0x18 | Yes | 1.0.1 |
| Scheduled Extended | 0x19 | Yes | 1.0.1 |
| Combined heating/cooling | 0x1A | Yes | 1.0.1 |
| Heat Intelligence | 0x1B | Yes | 1.0.1 |
| Scheduled Extended+ | 0x3B telegram 1<br>0x3C telegram 2 | Yes | 1.0.3 |
| Pulse | 0x1C telegram 1<br>0x1D telegram 2 | Yes | 1.0.4 |
| Pulse Extended | 0x4D telegram 1<br>0x4E telegram 2 | Yes | 1.0.4 |
| Scheduled Monthly | 0x4F telegram 1<br>0x50 telegram 2 | Yes | 1.0.5 |
| Scheduled Daily | 0x51 telegram 1<br>0x52 telegram 2 | Yes | 1.0.5 |
| Maximum Flow | 0x53 | No | 1.0.5 |

**Note**
Message formats with more than one message ID, such as Pulse Extended, have been divided into several telegrams due to size limitations.

### 6.3.1. Unscheduled message formats

#### 6.3.1.1. Standard

Table 8. Payload, message format Standard [0x15]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x15 (Standard) |

| Field | Size | Data type | Description |
|---|---|---|---|
| Heat energy E1 / Cooling energy E3 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |
| Volume | 6 bytes | INT32 | Volume ( m³ )<br><br>0411xxxxxxxx = xxxxxxxx * 0.00001 m³<br>0412xxxxxxxx = xxxxxxxx * 0.0001 m³<br>0413xxxxxxxx = xxxxxxxx * 0.001 m³<br>0414xxxxxxxx = xxxxxxxx * 0.01 m³<br>0415xxxxxxxx = xxxxxxxx * 0.1 m³<br>0416xxxxxxxx = xxxxxxxx m³<br>0417xxxxxxxx = xxxxxxxx * 10 m³ |
| Power | 4 bytes | INT16 | Power (W)<br><br>022Bxxxx = xxxx W<br>022Cxxxx = xxxx * 10 W<br>022Dxxxx = xxxx * 100 W<br>022Exxxx = xxxx kW<br>022Fxxxx = xxxx * 10 kW |
| Flow | 4 bytes | INT16 | Flow ( m³/h)<br><br>023Bxxxx = xxxx * 0.001 m³/h<br>023Cxxxx = xxxx * 0.01 m³/h<br>023Dxxxx = xxxx * 0.1 m³/h<br>023Exxxx = xxxx m³/h<br>023Fxxxx = xxxx * 10 m³/h |

| Field | Size | Data type | Description |
|---|---|---|---|
| Fw temp | 4 bytes | INT16 | Forward temperature (°C)<br><br>0258xxxx = xxxx * 0.001 °C<br>0259xxxx = xxxx * 0.01 °C<br>025Axxxx = xxxx * 0.1 °C<br>025Bxxxx = xxxx °C |
| Rt temp | 4 bytes | INT16 | Return temperature (°C)<br><br>025Cxxxx = xxxx * 0.001 °C<br>025Dxxxx = xxxx * 0.01 °C<br>025Exxxx = xxxx * 0.1 °C<br>025Fxxxx = xxxx °C |
| Meter ID | 6 bytes | According to M-Bus EN13757-3 identification field | Meter ID<br><br>0C78xxxxxxxx |
| Info bits | 7 bytes | INT32 | Error and warning flags<br><br>04FD17xxxxxxxx<br><br>For further information about Info bits please refer to the meter's manual |

### 6.3.1.2. Compact

Table 9. Payload, message format Compact [0x16]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x16 (Compact) |
| Heat energy E1 / Cooling energy E3 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |
| Meter ID | 6 bytes | According to M-Bus EN13757-3 identification field | Meter ID<br><br>0C78xxxxxxxx |

| Field | Size | Data type | Description |
|---|---|---|---|
| Info bits | 7 bytes | INT32 | Error and warning flags<br><br>04FD17xxxxxxxx<br><br>For further information about Info bits please refer to the meter's manual |

#### 6.3.1.3. JSON

For message format JSON, the data is presented in plain text rather than M-Bus format, unlike the other message formats. The table below contains a description of all fields included in the JSON telegram.

Table 10. Payload, message format JSON [0x17]

| Field | Description |
|---|---|
| Message format identifier | 0x17 (Message format JSON) |
| Energy | Energy consumption |
| Unit | Unit of energy consumption (Wh, kWh, MWh, GWh, J, kJ, MJ, GJ, Cal, kCal, MCal or GCal) |
| Meter ID | Identification number of the meter in which the module is mounted. |

See the following example of a telegram for message format JSON:

**{"E":12345678,"U":"kWh","ID":87654321}**

#### 6.3.1.4. Maximum flow

The Maximum flow message format includes the maximum flow from the previous month, as well as the date on which it occurred. See the details of the content in the table below.

Table 11. Payload, message format Maximum flow [0x53]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x53 |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field. Example: 0C7896979072 (Meter 72909796) |
| Energy E1 or E3 | 6-7 | INT32 | Example: 040612340000 (13330 kWh), 040812340000 (13330 J) |
| Max. flow rate in the last completed month | 5 | INT16 | Example: D2013Bxxxx. Maximum flow last month. Storage nr 3 is used (DIF=0x44, DIFE=0x01). Function field of DIF set to 0b01 for "maximum" value. |
| Date of max flow | 5 | INT16 | Date (Type G), Example: D2016C1A36 (2024-06-26).<br><br>Storage nr 3 is used (DIF=0x44, DIFE=0x01).<br><br>Function field of DIF set to 0b01 for "maximum" value.<br><br>This is the date for the maximum flow, typically different from the due date timestamp.<br><br>Note: This date is not compensated to be sent in UTC since it is only a date. It is the date stored in the meter, which may be local time, not UTC. |
| Energy E1 or E3 at reporting date | 7-8 | INT32 | Example: 840106xxxxxxxx. Monthly energy value, either heat or cooling. Storage nr 2 is used for monthly value. |
| Rt temp | 4 | INT16 | Example: 425Exxxx. Storage nr 1 is used for daily values in DIF. |
| Info codes | 7 | INT32 | Error and warning flags. Example: 44FD17xxxxxxxx or 42FD170000. Note: 4 bytes in MC603/803 and 2 in MC403. This register is read and updated prior to transmission and not taken from the monthly log. |

### 6.3.2. Scheduled message formats

For scheduled message formats, such as Scheduled Daily Redundant and Pulse Extended, two types of messages will be transmitted from the module - a clock message and a data message. Scheduled message formats will read the meter on top of the hour, making received telegrams more predictable. The difference between a clock message and a data message is described in the following table.

Table 12. Clock message and data message

| Message | Time interval | Description |
|---|---|---|
| Clock message | Once per day | The clock message presents the current time of the meter. It can be used to verify that the clock is correct and has not drifted more than accepted. |
| Data message | Determined by Transmit interval parameter | The actual meter data collected from the meter. |

The following table contains a detailed description of the payload of the clock message.

Table 13. Payload, clock message

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0xFA (=Clock message) |
| Date/Time | 6 bytes | 32 bit binary integer M-Bus type F | 046Dxxxxxx = Valid date/time message<br>346Dxxxxxx = Invalid date/time message |

The clock message will be transmitted once every day and the data message at least (regulated by transmit interval parameter or EcoMode) once every day. The transmit interval can only be set to the values listed in Table 14, “Transmit interval options ” [21].

Although the meter readout will occur on top-of-the-hour, the data message will not necessarily be transmitted at that exact time. The LoRaWAN® transmission will occur after a random delay of 0-15 minutes to decrease the risk of collisions. The readout for the clock message occurs at a random hour (00:00-23:00) at a random minute in the 35-45 minute interval and will be transmitted immediately after the readout.

**Note**
When using scheduled message formats, the transmit interval cannot be set to higher than 1440 (24 hours).

Table 14. Transmit interval options

| Parameter | Values | Unit |
|---|---|---|
| Transmit interval | 60, 120, 180, 240, 360, 480, 720, 1440 | Minutes |

#### 6.3.2.1. Scheduled Daily Redundant

Message format *Scheduled Daily Redundant* contains an accumulated daily energy field, which is updated at 24:00 each day. In other words, depending on transmit interval settings and data rate, the same field will be included in 1 to 24 telegrams per day. This will increase the probability of the value being received. For example, if the transmit interval is set to “120”, the accumulated energy read at 24:00 will be transmitted 12 times during the next 24 hours (every second hour).

Table 15. Payload, message format Scheduled Daily Redundant [0x18]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x18 (Scheduled-daily redundant) |

| Field | Size | Data type | Description |
|---|---|---|---|
| Heat energy E1 / Cooling Energy E3 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |
| Volume | 6 bytes | INT32 | Volume ( m³ )<br><br>0411xxxxxxxx = xxxxxxxx * 0.00001 m³<br>0412xxxxxxxx = xxxxxxxx * 0.0001 m³<br>0413xxxxxxxx = xxxxxxxx * 0.001 m³<br>0414xxxxxxxx = xxxxxxxx * 0.01 m³<br>0415xxxxxxxx = xxxxxxxx * 0.1 m³<br>0416xxxxxxxx = xxxxxxxx m³<br>0417xxxxxxxx = xxxxxxxx * 10 m³ |
| Meter ID | 6 bytes | According to M-Bus EN13757-3 identification field | Meter ID<br><br>0C78xxxxxxxx |
| Info bits | 7 bytes | INT32 | Error and warning flags<br><br>04FD17xxxxxxxx<br><br>For further information about Info bits please refer to the meter’s manual |

| Field | Size | Data type | Description |
|---|---|---|---|
| Meter date/time | 6 bytes | INT32 | Meter date and time (YY-MM-DD HH:MM)<br><br>046Dxxxxxxxx<br><br>Bit 31-28 = Year-high*<br>Bit 27-24 = Month<br>Bit 23-21 = Year-low*<br>Bit 20-16 = Day<br>Bit 15 = Summertime flag**<br>Bit 12-8 = Hour<br>Bit 7 = Error flag***<br>Bit 6 = Reserved for future use***<br>Bit 5-0 = Minute<br><br>*The year is read by combining the year-high and year-low field. For example, year-high = 0010 and year-low = 010 => year = 0010010<br><br>**0 = standard time, 1= daylight-saving time<br><br>***0 = timestamp is valid, 1 = timestamp is not valid |
| Accumulated heat / cooling energy at 24:00 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>4400xxxxxxxx = xxxxxxxx,xxx Wh<br>4401xxxxxxxx = xxxxxxxx,xx Wh<br>4402xxxxxxxx = xxxxxxxx,x Wh<br>4403xxxxxxxx = xxxxxxxx Wh<br>4404xxxxxxxx = xxxxxxxx *10 Wh<br>4405xxxxxxxx = xxxxxxxx *100 Wh<br>4406xxxxxxxx = xxxxxxxx kWh<br>4407xxxxxxxx = xxxxxxxx * 10 kWh<br>440Exxxxxxxx = xxxxxxxx MJ<br>440Fxxxxxxxx = xxxxxxxx * 10 MJ<br>44FB0Dxxxxxxxx = xxxxxxxx MCal<br>44FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>44FB0Fxxxxxxxx = xxxxxxxx * 100 MCal<br><br>Note: Before a midnight reading has been performed the Function field of the DIF is set to “value during error state” to indicate that the value is not valid. |

### 6.3.2.2. Scheduled Extended

The data message format *Scheduled Extended* contains all the data fields from message format *Standard*. In addition to these, it also includes the meter date/time to transmit messages at the top of the hour. As for all scheduled message formats, the transmit interval can only be set to the values included in Table 14, “Transmit interval options ” [21].

The following table contains a detailed description of the payload of message format *Scheduled Extended*.

Table 16. Payload, message format Scheduled Extended [0x19]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x19 (Scheduled-Extended) |
| Heat energy E1 / Cooling Energy E3 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |
| Volume | 6 bytes | INT32 | Volume ( m³ )<br><br>0411xxxxxxxx = xxxxxxxx * 0.00001 m³<br>0412xxxxxxxx = xxxxxxxx * 0.0001 m³<br>0413xxxxxxxx = xxxxxxxx * 0.001 m³<br>0414xxxxxxxx = xxxxxxxx * 0.01 m³<br>0415xxxxxxxx = xxxxxxxx * 0.1 m³<br>0416xxxxxxxx = xxxxxxxx m³<br>0417xxxxxxxx = xxxxxxxx * 10 m³ |
| Power / Flow / Fw temp / Rt temp | 12 bytes | INT64 | Byte 0-2 = DIF/VIF codes, 0x07FFA0<br>Byte 3 = VIFE defining scaling of Power/Flow<br>-Bit 6.4 (n), 10n-3 W, n = 0..7<br>-Bit 2..0 (m), 10m-3 m³/h, m = 0..7<br>Byte 4-5 = Fw temp (lsByte -> msByte), °C, 2 decimals<br>Byte 6-7 = Rt temp (lsByte -> msByte), °C, 2 decimals<br>Byte 8-9 = Flow (lsByte -> msByte), 10m-6 m³/h<br>Byte 10-11 = Power (lsByte -> msByte), 10n-3 W |
| Meter ID / Info bits | 11 bytes | INT96 | Byte 0-2 = DIF/VIF codes, 0x07FF21<br>Byte 3-6 = Info bits (lsByte -> msByte)<br>Byte 7-10 = Meter ID (lsByte -> msByte)* *Sent in binary format |

| Field | Size | Data type | Description |
|---|---|---|---|
| Meter date/time | 6 bytes | INT32 | Meter date and time (YY-MM-DD HH:MM)<br><br>046Dxxxxxxxx<br><br>Bit 31-28 = Year-high*<br>Bit 27-24 = Month<br>Bit 23-21 = Year-low*<br>Bit 20-16 = Day<br>Bit 15 = Summertime flag**<br>Bit 12-8 = Hour<br>Bit 7 = Error flag***<br>Bit 6 = Reserved for future use***<br>Bit 5-0 = Minute<br><br>*The year is read by combining the year-high and year-low field. For example, year-high = 0010 and year-low = 010 => year = 0010010<br><br>**0 = standard time, 1= daylight-saving time<br><br>***0 = timestamp is valid, 1 = timestamp is not valid |

### 6.3.2.3. Combined heat/cooling

**Note**
Message format Combined heat/cooling is only meant to be used in combined heat/cooling meters.

Table 17. Payload, message format Combined heat/cooling [0x1A]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x1A (Combined heat/cooling) |
| Heat energy E1 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |

| Field | Size | Data type | Description |
|---|---|---|---|
| Cooling Energy E3 | 8-9 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0483FF02xxxxxxxx = xxxxxxxx Wh<br>0484FF02xxxxxxxx = xxxxxxxx * 10 Wh<br>0485FF02xxxxxxxx = xxxxxxxx * 100 Wh<br>0486FF02xxxxxxxx = xxxxxxxx kWh<br>0487FF02xxxxxxxx = xxxxxxxx * 10 kWh<br>048EFF02xxxxxxxx = xxxxxxxx MJ<br>048FFF02xxxxxxxx = xxxxxxxx * 10 MJ<br>04FB8DFF02xxxxxxxx = xxxxxxxx MCal<br>04FB8EFF02xxxxxxxx = xxxxxxxx * 10 MCal<br>04FB8FFF02xxxxxxxx = xxxxxxxx * 100 MCal |
| Volume | 6 bytes | INT32 | Volume ( m³ )<br><br>0411xxxxxxxx = xxxxxxxx * 0.00001 m³<br>0412xxxxxxxx = xxxxxxxx * 0.0001 m³<br>0413xxxxxxxx = xxxxxxxx * 0.001 m³<br>0414xxxxxxxx = xxxxxxxx * 0.01 m³<br>0415xxxxxxxx = xxxxxxxx * 0.1 m³<br>0416xxxxxxxx = xxxxxxxx m³<br>0417xxxxxxxx = xxxxxxxx * 10 m³ |
| Fw temp | 4 bytes | INT16 | Forward temperature (°C)<br><br>0258xxxx = xxxx * 0.001 °C<br>0259xxxx = xxxx * 0.01 °C<br>025Axxxx = xxxx * 0.1 °C<br>025Bxxxx = xxxx °C |
| Rt temp | 4 bytes | INT16 | Return temperature (°C)<br><br>025Cxxxx = xxxx * 0.001 °C<br>025Dxxxx = xxxx * 0.01 °C<br>025Exxxx = xxxx * 0.1 °C<br>025Fxxxx = xxxx °C |
| Meter ID | 6 bytes | According to M-Bus EN13757-3 identification field | Meter ID<br><br>0C78xxxxxxxx |
| Info bits | 7 bytes | INT32 | Error and warning flags<br><br>04FD17xxxxxxxx<br><br>For further information about Info bits please refer to the meter's manual |

### 6.3.2.4. Heat intelligence

This telegram will be adapted to the type of meter in which the module is mounted. That means that it will look slightly different depending on whether the module is mounted in a heat meter, a cooling meter or a combined

heat/cooling meter. Please note for instance that DIB 2 in the table below is only included in combined heat/cooling meters, i.e. not in pure heat meters or pure cooling meters.

Table 18. Payload, message format Heat Intelligence [0x1B]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x1B (Heat intelligence) |
| Heat energy E1 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |
| Cooling energy E3 | 8-9 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0480 FF02xxxxxxxx = xxxxxxxx,xxx Wh<br>0481 FF02xxxxxxxx = xxxxxxxx,xx Wh<br>0482 FF02xxxxxxxx = xxxxxxxx,x Wh<br>0483FF02xxxxxxxx = xxxxxxxx Wh<br>0484FF02xxxxxxxx = xxxxxxxx * 10 Wh<br>0485FF02xxxxxxxx = xxxxxxxx * 100 Wh<br>0486FF02xxxxxxxx = xxxxxxxx kWh<br>0487FF02xxxxxxxx = xxxxxxxx * 10 kWh<br>048EFF02xxxxxxxx = xxxxxxxx MJ<br>048FFF02xxxxxxxx = xxxxxxxx * 10 MJ<br>04FB8DFF02xxxxxxxx = xxxxxxxx MCal<br>04FB8EFF02xxxxxxxx = xxxxxxxx * 10 MCal<br>04FB8FFF02xxxxxxxx = xxxxxxxx * 100 MCal |

| Field | Size | Data type | Description |
|---|---|---|---|
| Volume | 6 bytes | INT32 | Volume ( m³ )<br><br>0411xxxxxxxx = xxxxxxxx * 0.00001 m³<br>0412xxxxxxxx = xxxxxxxx * 0.0001 m³<br>0413xxxxxxxx = xxxxxxxx * 0.001 m³<br>0414xxxxxxxx = xxxxxxxx * 0.01 m³<br>0415xxxxxxxx = xxxxxxxx * 0.1 m³<br>0416xxxxxxxx = xxxxxxxx m³<br>0417xxxxxxxx = xxxxxxxx * 10 m³ |
| Meter ID / Info bits | 16 bytes | INT96 | Byte 0-2 = DIF/VIF codes, 0x07FF21<br>Byte 3-6 = Info bits (lsByte -> msByte)<br>Byte 7-10 = Meter ID (lsByte -> msByte)*<br><br>*Sent in binary format |
| Energy E8 | 7 bytes | INT32 | Energy ( m³ * °C)<br><br>04FF07xxxxxxxx = xxxxxxxx m³ * °C |
| Energy E9 | 7 bytes | INT32 | Energy ( m³ * °C)<br><br>04FF08xxxxxxxx = xxxxxxxx m³ * °C |

### 6.3.2.5. Scheduled Extended+

The data message of message format *Scheduled Extended+* contains three tariffs and all the meter data included in the Standard telegram. In addition to these, a timestamp from the meter (meter date/time) is included in each telegram. It is divided into two telegrams to keep each telegram below 45 bytes.

Table 19. Telegram 1 - Payload, message format Scheduled Extended+ [0x3B]

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x3B (Scheduled Extended+, telegram 1) |
| Heat energy E1 / Cooling energy E3 | 6-7 bytes | INT32 | Energy consumption (Wh, J, Cal)<br><br>0400xxxxxxxx = xxxxxxxx,xxx Wh<br>0401xxxxxxxx = xxxxxxxx,xx Wh<br>0402xxxxxxxx = xxxxxxxx,x Wh<br>0403xxxxxxxx = xxxxxxxx Wh<br>0404xxxxxxxx = xxxxxxxx * 10 Wh<br>0405xxxxxxxx = xxxxxxxx * 100 Wh<br>0406xxxxxxxx = xxxxxxxx kWh<br>0407xxxxxxxx = xxxxxxxx * 10 kWh<br>040Exxxxxxxx = xxxxxxxx MJ<br>040Fxxxxxxxx = xxxxxxxx * 10 MJ<br>04FB0Dxxxxxxxx = xxxxxxxx MCal<br>04FB0Exxxxxxxx = xxxxxxxx * 10 MCal<br>04FB0Fxxxxxxxx = xxxxxxxx * 100 MCal |
| Energy Tariff 2 | 8 | INT32 | 840203xxxxxxxx = xxxxxxxx Wh |
| Energy Tariff 3 | 8 | INT32 | 840303xxxxxxxx = xxxxxxxx Wh |

| Field | Size | Data type | Description |
|---|---|---|---|
| Meter ID | 6 bytes | According to M-Bus EN13757-3 identification field | Meter ID<br><br>0C78xxxxxxxx |
| Meter date/time | 6 bytes | INT32 | Meter date and time (YY-MM-DD HH:MM)<br><br>046Dxxxxxxxx<br><br>Bit 31-28 = Year-high*<br>Bit 27-24 = Month<br>Bit 23-21 = Year-low*<br>Bit 20-16 = Day<br>Bit 15 = Summertime flag**<br>Bit 12-8 = Hour<br>Bit 7 = Error flag***<br>Bit 6 = Reserved for future use***<br>Bit 5-0 = Minute<br><br>*The year is read by combining the year-high and year-low field. For example, year-high = 0010 and year-low = 010 => year = 0010010<br><br>**0 = standard time, 1= daylight-saving time<br><br>***0 = timestamp is valid, 1 = timestamp is not valid |

Table 20. Telegram 2 - Payload, message format Scheduled Extended+

| Field | Size | Data type | Description |
|---|---|---|---|
| Message format identifier | 1 byte | - | 0x3C (Scheduled Extended+) |
| Volume | 6 bytes | INT32 | Volume ( m³ )<br><br>0411xxxxxxxx = xxxxxxxx * 0.00001 m³<br>0412xxxxxxxx = xxxxxxxx * 0.0001 m³<br>0413xxxxxxxx = xxxxxxxx * 0.001 m³<br>0414xxxxxxxx = xxxxxxxx * 0.01 m³<br>0415xxxxxxxx = xxxxxxxx * 0.1 m³<br>0416xxxxxxxx = xxxxxxxx m³<br>0417xxxxxxxx = xxxxxxxx * 10 m³ |
| Power | 4 bytes | INT16 | Power (W)<br><br>022Bxxxx = xxxx W<br>022Cxxxx = xxxx * 10 W<br>022Dxxxx = xxxx * 100 W<br>022Exxxx = xxxx kW<br>022Fxxxx = xxxx * 10 kW |

| Field | Size | Data type | Description |
|---|---|---|---|
| Flow | 4 bytes | INT16 | Flow ( m³/h)<br><br>023Bxxxx = xxxx * 0.001 m³/h<br>023Cxxxx = xxxx * 0.01 m³/h<br>023Dxxxx = xxxx * 0.1 m³/h<br>023Exxxx = xxxx m³/h<br>023Fxxxx = xxxx * 10 m³/h |
| Fw temp | 4 bytes | INT16 | Forward temperature (°C)<br><br>0258xxxx = xxxx * 0.001 °C<br>0259xxxx = xxxx * 0.01 °C<br>025Axxxx = xxxx * 0.1 °C<br>025Bxxxx = xxxx °C |
| Rt temp | 4 bytes | INT16 | Return temperature (°C)<br><br>025Cxxxx = xxxx * 0.001 °C<br>025Dxxxx = xxxx * 0.01 °C<br>025Exxxx = xxxx * 0.1 °C<br>025Fxxxx = xxxx °C |
| Meter ID | 6 bytes | According to M-Bus EN13757-3 identification field | Meter ID<br><br>0C78xxxxxxxx |
| Meter date/time | 6 bytes | INT32 | Meter date and time (YY-MM-DD HH:MM)<br><br>046Dxxxxxxxx<br><br>Bit 31-28 = Year-high*<br>Bit 27-24 = Month<br>Bit 23-21 = Year-low*<br>Bit 20-16 = Day<br>Bit 15 = Summertime flag**<br>Bit 12-8 = Hour<br>Bit 7 = Error flag***<br>Bit 6 = Reserved for future use***<br>Bit 5-0 = Minute<br><br>*The year is read by combining the year-high and year-low field. For example, year-high = 0010 and year-low = 010 => year = 0010010<br><br>**0 = standard time, 1= daylight-saving time<br><br>***0 = timestamp is valid, 1 = timestamp is not valid |
| Info bits | 7 bytes | INT32 | Error and warning flags<br><br>04FD17xxxxxxxx<br><br>For further information about Info bits please refer to the meter's manual |

### 6.3.2.6. Pulse

Message format *Pulse* includes data from the pulse inputs In A and In B of the module. How the pulses should be interpreted can be configured in the meter interface. Depending on configuration, the pulses can represent either volume or energy. Due to size, the message format is divided in two telegrams.

The following table contains a detailed description of the payload of message format *Pulse*.

Table 21. Telegram 1 - Payload, message format Pulse [0x1C]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x1C |
| Date and time | 6 | INT32 | Date and Time (M-Bus Type F)<br>Example: 046Dxxxxxxxx |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field<br>Example: 0C78xxxxxxxx |
| Energy | 6-7 | INT32 | Example: 0406xxxxxxxx, 040Fxxxxxxxx |
| Volume | 6 | INT32 | Example: 0413xxxxxxxx |
| Power | 4 | INT16 | Example: 022Bxxxx |
| Flow | 4 | INT16 | Example: 023Bxxxx |
| Fw temp | 4 | INT16 | Example: 025Axxxx |
| Rt temp | 4 | INT16 | Example: 025Exxxx |

Table 22. Telegram 2 - Payload, message format Pulse [0x1D]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x1D |
| Date and time | 6 | INT32 | Date and Time (M-Bus Type F)<br>Example: 046Dxxxxxxxx |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field<br>Example: 0C78xxxxxxxx |
| Pulse In A | 7 | INT32 | Reads the registers corresponding to the In A connector on the module.<br>Sub-unit 1 is used in DIFE for In A.<br>Examples:<br>Volume: 844014B1EB0100 = 1258,73 $m^3$<br>Energy: 84400725120000 = 46450 kWh |
| Pulse In B | 8 | INT32 | Reads the registers corresponding to the In B connector on the module<br>Sub-unit 2 is used in DIFE for In B<br>Examples:<br>Volume: 848040144E1E0100 = 732,94 $m^3$<br>Energy: 8480400601230000 = 8961 kWh |
| Operating hours | 6 | INT32 | Reads the Operating Hours register of the meter.<br>Examples:<br>042238220000 = 8 760h (~1 year)<br>042280230200 = 140 160h (~16 years) |
| Info codes | 7 | INT32 | Error and warning flags.<br>Example: 04FD17xxxx<br>Note that this field is always 32 bits, even when using MULTICAL® 403 which only reports 16 bits. |

### 6.3.2.7. Pulse Extended

Message format *Pulse Extended* is similar to message format Pulse and includes data from the pulse inputs In A and In B of the module. Besides other data points, this message format also includes energy tariffs. How the pulses should be interpreted can be configured in the meter interface. Depending on configuration, the pulses can represent either volume or energy. Due to size, the message format is divided in two telegrams.

The following table contains a detailed description of the payload of message format *Pulse Extended*.

Table 23. Telegram 1 - Payload, message format Pulse Extended [0x4D]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x4D |
| Date and time | 6 | INT32 | Date and Time (M-Bus Type F)<br>Example: 046Dxxxxxxxx |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field<br>Example: 0C78xxxxxxxx |
| Energy | 6-7 | INT32 | Example: 0406xxxxxxxx, 040Fxxxxxxxx |
| Volume | 6 | INT32 | Example: 0413xxxxxxxx |
| Power | 4 | INT16 | Example: 022Bxxxx |
| Fw temp | 4 | INT16 | Example: 025Axxxx |
| Rt temp | 4 | INT16 | Example: 025Exxxx |
| Info codes | 7 | INT32 | Error and warning flags<br>Example: 04FD17xxxx<br>Note that this field is always 32 bits, even when using MULTICAL® 403 which only reports 16 bits. |

Table 24. Telegram 2 - Payload, message format Pulse Extended [0x4E]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x4E |
| Date and time | 6 | INT32 | Date and Time (M-Bus Type F)<br>Example: 046Dxxxxxxxx |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field<br>Example: 0C78xxxxxxxx |
| Energy Tariff 2 | 7-8 | INT32 | 842003xxxxxxxx |
| Energy Tariff 3 | 7-8 | INT32 | 843003xxxxxxxx |
| Pulse In A | 7 | INT32 | Reads the registers corresponding to the In A connector on the module.<br>Sub-unit 1 is used in DIFE for In A.<br>Examples:<br>Volume: 844014B1EB0100 = 1258,73 $m^3$<br>Energy: 84400725120000 = 46450 kWh |
| Pulse In B | 8 | INT32 | Reads the registers corresponding to the In B connector on the module<br>Sub-unit 2 is used in DIFE for In B<br>Examples:<br>Volume: 848040144E1E0100 = 732,94 $m^3$<br>Energy: 8480400601230000 = 8961 kWh |

### 6.3.2.8. Scheduled Monthly Extended

The Scheduled Monthly Extended message format is developed to meet the requirements of customers with high data privacy requirements. Unlike several other message formats, current values are not sent. Instead, the meter log is used to get values from the last day or month. For example, this means that the same monthly value will be repeated throughout a month. Due to size limitations, this message format is divided into two telegrams.

**Note**

The availability of the log data is dependent on correct configuration of the meter. If the meter is not configured to store the needed log data it will not be part of the payload.

This is an important prerequisite for using this message format. Monthly values are read from the monthly log in the meter (using storage number 2 when sent as M-Bus). Storage number bit in DIF unset and bit 0 in DIFE set means storage number 2.

There are three timestamps included in this format, separated by storage number. Date of monthly value (storage number 2), Date of max flow value last month (storage number 3), and Date and time of info codes (storage number 0)

Table 25. Telegram 1 - Payload message format Scheduled Monthly Extended [0x4F]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x4F |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field. Example: 0C7896979072 (Meter 72909796) |
| Date part of reporting date (monthly log) | 5 | INT32 | Date (M-Bus Type G), Example: 82016C1A36 (2024-06-26). Storage number 2 used for monthly value. Note: this date is not compensated to be sent in UTC since it is only a date. It is the date stored in the meter, which may be local time, not UTC. |
| Energy E1 or E3 at reporting date (monthly log) | 7-8 | INT32 | Example: 840106xxxxxxxx. Monthly energy value, either heat or cooling. Storage number 2 is used for monthly value. |
| Volume at the reporting date (monthly log) | 7 | INT32 | Example: 840111xxxxxxxx. Monthly volume value. Storage number 2 is used for monthly value. |
| Power at reporting date (monthly log) | 5 | INT16 | Example: 82012Bxxxx. Storage number 2 is used for monthly value. |
| Date and Time | 6 | INT32 | Date and Time (Type F), Example: 046D00262332 (2025-02-03 06:00). Note: this is the time for the info codes! |
| Info codes | 5-7 | INT32 | Error and warning flags. Example: 04FD17xxxxxxxx or 02FD170000. Note: 4 bytes in MC603/803 and 2 in MC403. This register is read and updated prior to transmission and not taken from the monthly log. |

Table 26. Telegram 2 - Payload message format Scheduled Monthly Extended [0x50]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x50 |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field. Example: 0C7896979072 (Meter 72909796) |
| Date part of reporting date (daily log) | 4 | INT32 | Date (M-Bus Type G), Example: 426C1A36 (2024-06-26). Storage number 1 used for daily value. Note: this date is not compensated to be sent in UTC since it is only a date. It is the date stored in the meter, which may be local time, not UTC. |
| Flow at reporting date (daily log) | 4 | INT16 | Example: 423Bxxxx. Storage number 1 is used for daily value. |
| Fw temp at reporting date (daily log) | 4 | INT16 | Example: 425Axxxx. Storage number 1 is used for daily value. |
| Rt temp at reporting date (daily log) | 4 | INT16 | Example: 425Exxxx. Storage number 1 is used for daily value. |
| Max. flow rate in the last completed month | 5 | INT16 | Example: D2013Bxxxx. Maximum flow last month. Storage number 3 is used (DIF=0xD2, DIFE=0x01). Function field of DIF set to 0b01 for “maximum” value. Note: this date is not compensated to be sent in UTC since it is only a date. It is the date stored in the meter, which may be local time, not UTC. |
| Date of max flow | 5 | INT32 | Date and Time (Type G), Example: C2016C1A36 (2024-06-26). Storage number 3 is used (DIF=0xC2, DIFE=0x01). This is the date for the maximum flow, typically different from the due date timestamp. |

### 6.3.2.9. Scheduled Daily Extended

The Scheduled Daily Extended message format is developed to meet the requirements of customers with high data privacy requirements. Unlike several other message formats, current values are not sent. Instead, the meter log is utilized to get values from the last day or month. For example, this means that the same daily value will be repeated throughout the day if sent multiple times (which in turn will increase the probability of receiving the message). Due to size limitations, this message format is divided into two telegrams.

**Note**

The availability of the log data is dependent on correct configuration of the meter. If the meter is not configured to store the needed log data it will not be part of the payload.

This is an important prerequisite for using this message format. Monthly values are read from the monthly log in the meter (using storage number 2 when sent as M-Bus). Storage number bit in DIF unset and bit 0 in DIFE set means storage number 2.

There are three timestamps included in this format, separated by storage number. Date of monthly value (storage number 2), Date of max flow value last month (storage number 3), and Date and time of info codes (storage number 0)

Table 27. Telegram 1 - Payload message format Scheduled Daily Extended [0x51]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x51 |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field. Example: 0C7896979072 (Meter 72909796) |
| Date part of reporting date | 4 | INT32 | Date (M-Bus Type G), Example: 426C1A36 (2024-06-26). Storage number 1 used for daily value. Note: this date is not compensated to be sent in UTC since it is only a date. It is the date stored in the meter, which may be local time, not UTC. |
| Energy (E1 or E3 depending on meter type) | 6-7 | INT32 | Example: 4406xxxxxxxx, 440Fxxxxxxxx. Storage number 1 is used for daily values in DIF. |
| Volume | 6 | INT32 | Example: 4413xxxxxxxx. Storage number 1 is used for daily values in DIF. |
| Power | 4 | INT16 | Example: 422Bxxxx. Storage number 1 is used for daily values in DIF. |
| Flow | 4 | INT16 | Example: 423Bxxxx. Storage number 1 is used for daily values in DIF. |

Table 28. Telegram 2 - Payload message format Scheduled Daily Extended [0x52]

| Field | Size (bytes) | Data type | Description |
|---|---|---|---|
| Message ID | 1 | UINT8 | Always 0x52 |
| Meter ID | 6 | BCD8 | According to M-Bus EN13757-3 identification field. Example: 0C7896979072 (Meter 72909796) |
| Date part of reporting date | 4 | INT32 | Date (M-Bus Type G), Example: 426C1A36 (2024-06-26). Storage number 1 used for daily value. Note: this date is not compensated to be sent in UTC since it is only a date. It is the date stored in the meter, which may be local time, not UTC. |
| Fw temp | 4 | INT16 | Example: 425Axxxx. Storage number 1 is used for daily values in DIF. |
| Rt temp | 4 | INT16 | Example: 425Exxxx. Storage number 1 is used for daily values in DIF. |
| Date and Time | 6 | INT32 | Date and Time (Type F), Example: 046Dxxxxxxxx. |
| Info codes | 5-7 | INT32 | Error and warning flags. Example: 04FD17xxxxxxxx or 02FD170000. Note: 4 bytes in MC603/803 and 2 in MC403. This register is read and updated prior to transmission and not taken from the daily log. |

# 7. Technical specifications

| Mechanics | Value |
|---|---|
| Dimensions (W x H x D) | 90 x 12 x 35 mm |
| Weight | 33 g |
| Mounting | In a module slot of Kamstrup MULTICAL® 403/603/803 |
| External antenna connector | MCX |

| Electrical connections | Value |
|---|---|
| Supply voltage | Internal meter battery (up to 11 years lifetime, having EcoMode enabled, using Kamstrup's D-cell meter battery). |

| Electrical characteristics | Value |
|---|---|
| Nominal voltage | 3.0 VDC |
| Power consumption (max) | 50 mA |
| Power consumption (sleep mode) | 2.5 µA |

| Environmental specifications | Value |
|---|---|
| Operating temperature | +5 - +55 ºC |
| Operating humidity | 0 - 93 % RH, no condensation |
| Operating altitude | 2000 m |
| Pollution degree | Degree 1 |
| Usage environment | Indoors |
| Storage temperature | -20 - 60 ºC |

| Radio characteristics | Value |
|---|---|
| Frequency | 868 MHz |
| Output power | 14 dBm |
| Receiver sensitivity | -135 dBm |

| LoRaWAN® characteristics | Value |
|---|---|
| Device class | Class A, bi-directional |
| LoRaWAN® version | 1.0.2 |
| Activation | OTAA or ABP |
| Data rate | DR0 - DR5 (250-5470 bit/s) |
| JoinEUI | 94193A030B000001 (Retrofitted devices)<br><br>94193A0311000001 (Factory-mounted devices) |

**Tip**
The JoinEUI is determined by the DevEUI series. DevEUIs in the 94193A010BXXXXXX series use JoinEUI 94193A030B000001, while DevEUIs in the 94193A0111XXXXXX series use JoinEUI 94193A0311000001. The DevEUI series identifier is 01; the corresponding JoinEUI series identifier is 03.

| User interface | Value |
|---|---|
| Push button | Start-up / reboot / switch off the module |
| Configuration | NFC via the Elvaco OTC app or via downlink data |

| Approvals | Value |
|---|---|
| LoRa Alliance® | LoRaWAN Certified$^{CM}$ |
| EMC | EN 301489-1, EN 301489-3 |

# 8. Simplified Declaration of Conformity

Hereby, Elvaco declares that CMi4140 is in compliance with the following directives:

| EU | UK |
|---|---|
| 2014/53/EU (RED) | 2017 No. 1206 (RED) |
| 2014/30/EU (EMC) | 2016 No. 1091 (EMC) |
| 2014/35/EU (LVD) | 2016 No. 1101 (LVD) |
| 2011/65/EU + 2015/863 (RoHS) | 2012 No. 3032 (RoHS) |

See EU Declaration of Conformity for a complete EU DoC.

See UK Declaration of Conformity for a complete UK DoC

# 9. Document history

Table 29. Versions

| Version | Date | Description |
|---|---|---|
| 0.1 | 2019-03 | PoC version |
| 0.2 | 2019-10 | Evaluation samples version |
| 1.0 | 2019-12 | Commercial release version |
| 1.1 | 2020-06 | Commercial release v1.1 |
| 1.2 | 2021-05 | Commercial release v1.2 |
| 1.3 | 2021-11 | Commercial release v1.3 |
| 1.4 | 2023-12 | Minor layout changes were made. |
| 1.5 | 2024-09 | Adaptation to release of FW 1.0.4. Added two new message formats, Pulse and Pulse Extended. |
| 1.6 | 2025-05 | Adaptation to release of FW 1.0.5. Added three new message formats, Maximum flow, Scheduled Daily extended, and Scheduled monthly extended. |
| 1.7 | 2026-05 | General clarifications and document structure updates. |

**Manufacturer:**

Elvaco AB / Kabelgatan 12 / 434 37 Kungsbacka / SWEDEN

info@elvaco.com

elvaco.com

**Support:**

support.elvaco.com

support@elvaco.se