

# SOMFY Digital Network Integration Guide



*DOC155888/004 - August, 2024*



|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUCTION .....</b>                  | <b>3</b>  |
| <b>2</b> | <b>SDN FRAME BUILDER .....</b>             | <b>4</b>  |
| <b>3</b> | <b>SDN BASICS.....</b>                     | <b>5</b>  |
| 3.1      | GENERAL INFORMATION .....                  | 5         |
| 3.2      | DEVICE IDENTIFICATION .....                | 5         |
| 3.2.1    | <i>NodeID</i> .....                        | 5         |
| 3.2.2    | <i>NodeType</i> .....                      | 5         |
| 3.3      | GROUPS.....                                | 6         |
| 3.4      | ADDRESSING MODES .....                     | 6         |
| 3.5      | MESSAGE TYPES .....                        | 6         |
| 3.6      | ACKNOWLEDGMENTS .....                      | 7         |
| 3.7      | COMMUNICATION COLLISIONS ON THE BUS .....  | 7         |
| <b>4</b> | <b>SERIAL COMMUNICATION.....</b>           | <b>8</b>  |
| 4.1      | CONFIGURATION.....                         | 8         |
| 4.2      | DATA TRANSMISSION .....                    | 8         |
| 4.3      | TIMINGS.....                               | 9         |
| <b>5</b> | <b>MESSAGES STRUCTURE .....</b>            | <b>10</b> |
| 5.1      | MSG.....                                   | 10        |
| 5.2      | ACK/LEN .....                              | 10        |
| 5.3      | NODE TYPE .....                            | 10        |
| 5.4      | SOURCE@ / DEST@ .....                      | 11        |
| 5.5      | DATA .....                                 | 11        |
| 5.6      | CHECKSUM .....                             | 11        |
| <b>6</b> | <b>MESSAGES .....</b>                      | <b>12</b> |
| 6.1      | DEVICE MANAGEMENT .....                    | 14        |
| 6.1.1    | <i>Device NodeID</i> .....                 | 14        |
| 6.1.2    | <i>Group Configuration</i> .....           | 15        |
| 6.1.3    | <i>Acknowledgement and Errors</i> .....    | 16        |
| 6.2      | DEVICE INFORMATION .....                   | 17        |
| 6.2.1    | <i>Firmware Revision</i> .....             | 17        |
| 6.2.2    | <i>User-defined Text Label</i> .....       | 18        |
| 6.2.3    | <i>Actuator's manufacturing date</i> ..... | 18        |
| 6.3      | DEVICE CONFIGURATION .....                 | 19        |
| 6.3.1    | <i>HMI Management</i> .....                | 19        |
| 6.3.2    | <i>Intermediate Positions</i> .....        | 21        |
| 6.3.3    | <i>Speed Adjustment</i> .....              | 22        |
| 6.3.4    | <i>Lock Network Commands</i> .....         | 23        |
| 6.4      | DEVICE CONTROL.....                        | 25        |
| 6.4.1    | <i>Move to Position</i> .....              | 25        |
| 6.4.2    | <i>Stop</i> .....                          | 25        |
| 6.4.3    | <i>Wink</i> .....                          | 25        |
| 6.5      | DEVICE STATUS .....                        | 26        |
| 6.5.1    | <i>Motor Position</i> .....                | 26        |
| 6.5.2    | <i>Motor Status</i> .....                  | 27        |
| 6.6      | DEVICE RESET .....                         | 28        |
| 6.6.1    | <i>Factory default settings</i> .....      | 28        |





# 1 Introduction

This document describes the “SOMFY Digital Network” (SDN) protocol which is implemented in all SOMFY RS485 products.

The provided content brings all technical information needed to establish a bi-directional communication with SOMFY RS485 devices.

- ⇒ Targeted audience: system integrators / developers
- ⇒ Pre-requisite: devices should already be set (i.e., Up/Down/Tilting limits, rotation direction and application mode already set)

↪ If devices are not configured, please use available tools:

| <b>Product</b>  | <b>Type</b>  | <b>Application Mode</b>  |
|---|--|--|
| <br>RS485 Setting Tool | Handheld<br>Battery powered<br><br>Single motor<br>connection  |  |
| <br>Set Pro by Somfy   | PC Software<br><br>Bus connection<br>with discovery<br>feature |  |

Set Pro by Somfy can be downloaded from the SOMFY Projects website under the “Somfy Digital Network (RS485)/Software” section:



<https://www.somfy.com/projects/downloads>

↪ Limits setting and motor rotation direction are not covered in this document.

For information about bus topology, cable length and other wiring instructions, please refer to DOC114316 “SDN Bus Wiring Guide” available separately.

The following symbols can be found in the document:



**Warning** ⇒ Indicating a key point to be taken in consideration.  
(Usually a common source of problems).

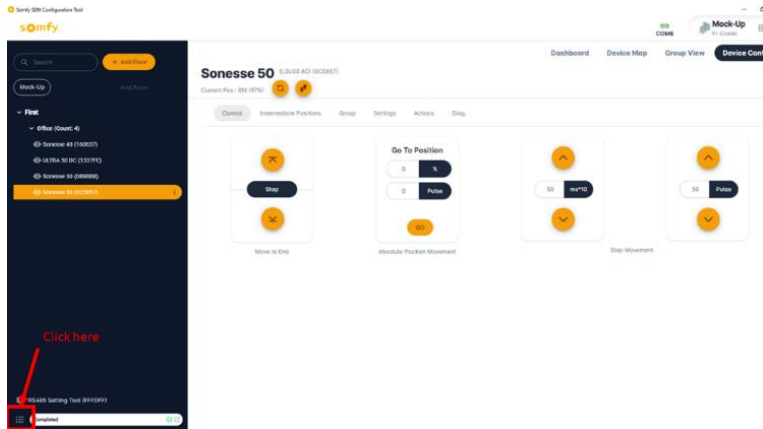


**Example** ⇒ Giving detailed help on how to implement a feature.

## 2 SDN Frame Builder

Inside Set Pro by Somfy, SOMFY provides a tool that helps encoding and decoding messages:

- Automatically generates messages for every device
- Automatically decodes messages (copy/paste from external software/terminal)



| No | Date                  | Source Node Type   | Dest Node Type     | Source | Dest   | Command             | Value     | Act |
|----|-----------------------|--------------------|--------------------|--------|--------|---------------------|-----------|-----|
| 7  | 03/07/24 14:07:14.012 | RS485 Setting Tool | LBU20AC            | 010000 | 013457 | CTRL_MOVE_TO        | 80000000  |     |
| 8  | 03/07/24 14:07:14.176 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | ACK                 |           |     |
| 9  | 03/07/24 14:07:15.068 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | CTRL_STOP           | 01        |     |
| 10 | 03/07/24 14:07:15.074 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | ACK                 |           |     |
| 11 | 03/07/24 14:07:16.047 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | GET_MOTOR_POSITION  |           |     |
| 12 | 03/07/24 14:07:16.166 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 13 | 03/07/24 14:07:17.078 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | GET_MOTOR_POSITION  |           |     |
| 14 | 03/07/24 14:07:17.005 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 15 | 03/07/24 14:07:17.045 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | GET_MOTOR_POSITION  |           |     |
| 16 | 03/07/24 14:07:17.233 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 17 | 03/07/24 14:07:18.074 | RS485 Setting Tool | LBU20AC            | 010000 | 013457 | GET_MOTOR_POSITION  |           |     |
| 18 | 03/07/24 14:07:18.097 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 19 | 03/07/24 14:07:18.856 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | GET_MOTOR_POSITION  |           |     |
| 20 | 03/07/24 14:07:18.223 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 21 | 03/07/24 14:07:19.022 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | GET_MOTOR_POSITION  |           |     |
| 22 | 03/07/24 14:07:19.936 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 23 | 03/07/24 14:07:19.945 | RS485 Setting Tool | LBU20AC            | 010000 | 0C3837 | GET_MOTOR_POSITION  |           |     |
| 24 | 03/07/24 14:07:19.234 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |
| 25 | 03/07/24 14:07:20.033 | RS485 Setting Tool | LBU20AC            | 010000 | 013457 | GET_MOTOR_POSITION  |           |     |
| 26 | 03/07/24 14:07:20.910 | LBU20AC            | RS485 Setting Tool | 0C3837 | 010000 | POST_MOTOR_POSITION | 30354FFFF |     |

Destination Node Type: All Devices  
Addressing Mode: Point to point  
Source Address: 010000  
Destination Address: 013457  
Acknowledgment Request: [ ]  
Command: CTRL\_STOP  
Send [ ] Copy [ ]

## 3 SDN Basics

### 3.1 General information

“SOMFY Digital Network” protocol handles only half-duplex communication between a MASTER and SLAVES:

- MASTER nodes can send commands at any time to one or more SLAVES on the bus.
- SLAVE nodes can only execute commands or report status when requested by a MASTER.
  - There's only one exception to this rule where some devices can send their address when requested by the user (generally using a pushbutton located on the device) without any MASTER request.

### 3.2 Device Identification

#### 3.2.1 NodeID

Every RS485 device has a built-in 3-bytes NodeID which is the address of the device on the bus and is used in every communication to identify the transmitter and the receiver(s).

This NodeID is programmed during product manufacturing and cannot be changed. It can be found on the device labels in both plain text and barcode format.

Addresses are recycled on a 3 to 5 years basis, depending on the product sales.




⇒ On a given installation, the NodeID can be considered as unique.

#### 3.2.2 NodeType

Every RS485 device has a built-in 4-bits value called NodeType which is used to identify the product family.

The NodeType can be used in bus communication to send messages only to a selected range of products.

The following table lists the current reserved NodeType, corresponding devices and supported application modes.

| NodeType | Device                | Supported Application Modes   |   |   |
|----------|-----------------------|---|---|---|
|          |                       |  |  |  |
| 02h      | Ø30 DC Serie RS485    | ✓   |   |   |
| 05h      | RS485 RTS transmitter | ✓   |   |   |
| 06h      | Glydea RS485          |   |   | ✓   |
| 07h      | Ø50 AC Serie RS485    | ✓   |   |   |
| 08h      | Ø50 DC Serie RS485    | ✓   |   |   |
| 09h      | Ø40 AC Serie RS485    | ✓   | ✓   |   |

### 3.3 Groups

Every device can be part of up to 16 groups, defined by a GroupID.

A GroupID has the exact same format as the NodeID and can be:

- The NodeID of an existing device on the bus (generally the case when this device acts as the only controller of the group)
- Any other NodeID, provided it's not used by any other devices on the bus (when more than one device can control the group)

To use group commands, every product belonging to a group must contain the GroupID in one of the 16 group table entries (i.e. GroupIndex).

⇒ See related messages [§6.1.2](#)

### 3.4 Addressing Modes

The protocol provides 3 different addressing modes:

- Point to point: communication from a transceiver to one and only one device
- Group: communication from a transmitter to a group of devices
- Broadcast: communication from a transmitter to all the devices

In addition, messages can be addressed to a dedicated device type using its NodeType.

### 3.5 Message Types

Messages are split into 3 categories:

- Settings:
  - Name = SET\_xxx
  - Change the configuration of the device
- Controls:
  - Name = CTRL\_xxx
  - Send a command to execute
- Status:
  - Name = GET\_xxx
  - Request information from the device (settings values or status)
  - SLAVE(s) will respond with the corresponding POST\_xxx message

### 3.6 Acknowledgments

Every message sent by a MASTER can be associated with an acknowledgment request. This optional acknowledgment will give feedback on message processing, according to the following table:

|          | ACK is sent when...                                | NACK  |
|----------|--|---|
| Settings | Parameters are saved                               | Cannot execute<br>see <a href="#">§6.1.3</a> for a list of errors |
| Controls | Execution is started, but not necessarily finished |   |
| Status   | n/a  |   |

No ACK is sent after a status request, as the feedback is given by the status report itself.

#### Warning



It's highly recommended to use acknowledgements to ensure messages are received and properly processed. In this case, a retry strategy can be implemented in the controller when:

- NACK is received (message as received but cannot be processed)
- No ACK is received after a period of time (message was not received)

⇒ See related messages [§6.1.3](#)

### 3.7 Communication collisions on the bus



Due to the nature of RS485 bus, collisions may happen. As a result, some messages will not reach their destination.

To lower the risk of collisions:

- Avoid requesting feedback in group or broadcast addressing mode
- Avoid requesting acknowledgment in group or broadcast mode

## 4 Serial Communication

### 4.1 Configuration

SDN uses an asynchronous serial communication:

|                  |                 |
|------------------|-----------------|
| Baud Rate        | 4800            |
| Data bits        | 8               |
| Parity           | Odd             |
| Start Bit        | Logical level 0 |
| Stop Bit         | Logical level 1 |
| Character coding | NRZ             |

### 4.2 Data transmission

- Least significant bit is always sent first



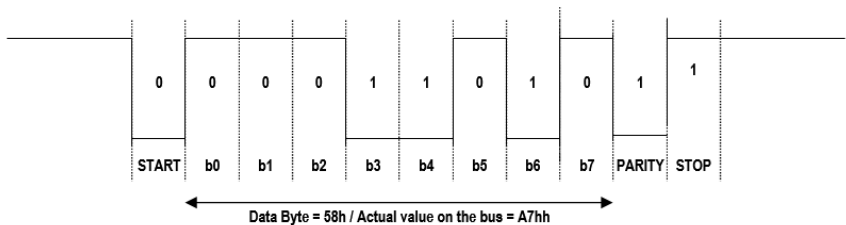
**Warning:**

To ensure backward compatibility with earliest versions of the protocol, all data bits need to be inverted before transmission.



**Example:**

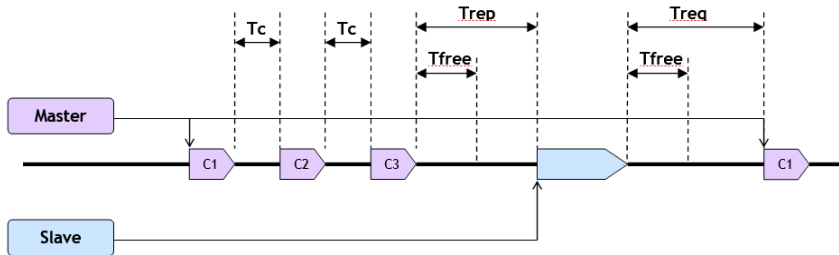
Transmission of data byte 58h => actual value on the bus = NOT (58h) = A7h





## 4.3 Timings

The following diagram shows a typical MASTER / SLAVE communication with timing requirements:



| Timing           | Min      | Typical | Max      | Description  |
|------------------|----------|---------|----------|--|
| Tc               | N/a      |         | ≤ 1ms    | Maximum time between two consecutive characters  |
| Tfree            | > Tc Max | -       | < 3ms    | Bus free timeout   |
| Trep (Unicast)   | ≥ 5ms    | -       | ≤ 255 ms | Bus inactivity delay for a slave node before sending its reply<br>Unicast = point to point communication<br>Multicast = Group or Broadcast |
| Trep (Multicast) | 30ms     | n/a     | 280ms    |  |
| Treq             | 25ms     | N/a     |          | Bus inactivity delay for a master before it can send a new request   |

No synchronization byte is defined in the protocol to detect the beginning or end of a message.

A message is seen as a bundle of bytes ended with bus inactivity.

To guarantee proper communication, the following rules apply:

### MASTER requirements

- Before transmitting data:
  - The MASTER device ensures that at least **Treq** has elapsed since the last bus activity.
- While transmitting data:
  - The MASTER device ensures that **Tc** is not exceeded between 2 characters

### SLAVE behaviors

- Before transmitting data:
  - Any SLAVE device ensures **Trep** has elapse since the last bus activity.
  - **Trep** is not a fixed value and can vary within the range indicated in the table above
- While transmitting data:
  - The SLAVE device ensures that **Tc** is not exceeded between 2 characters

## 5 Messages Structure

All SDN messages are formatted as follow:

| Byte 1     | Byte 2         | Byte 3           | Byte 4          | Byte 5 | Byte 6 | Byte 7        | Byte 8 | Byte 9 | ...         | Byte n-1        | Byte n |
|------------|----------------|------------------|-----------------|--------|--------|---------------|--------|--------|-------------|-----------------|--------|
| <i>MSG</i> | <i>ACK/LEN</i> | <i>NODE TYPE</i> | <i>SOURCE @</i> |        |        | <i>DEST @</i> |        |        | <i>DATA</i> | <i>CHECKSUM</i> |        |

- Minimum length = 11 bytes (i.e., without any DATA)
- Maximum length = 32 bytes (i.e., maximum 21 bytes of DATA)

Following sections give detailed explanation for each field.

### 5.1 MSG

| Byte 1 |    |    |    |    |    |    |    |
|--------|----|----|----|----|----|----|----|
| b7     | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| MSG    |    |    |    |    |    |    |    |

- **MSG** = Message identifier

Refer to messages list for supported MSG values (see §6)

### 5.2 ACK/LEN

| Byte 2 |     |    |     |    |    |    |    |
|--------|-----|----|-----|----|----|----|----|
| B7     | b6  | b5 | b4  | b3 | b2 | b1 | b0 |
| ACK    | EXT |    | LEN |    |    |    |    |

- **ACK** = Acknowledge request – Set to 1 to request acknowledge from the SLAVE(s)
- **EXT** = Always 0 (Reserved)
- **LEN** = Frame length (0 to 31)

### 5.3 NODE TYPE

| Byte 3           |    |    |    |                |    |    |    |
|------------------|----|----|----|----------------|----|----|----|
| b7               | b6 | b5 | b4 | b3             | b2 | b1 | b0 |
| SOURCE Node Type |    |    |    | DEST Node Type |    |    |    |

- **SOURCE NodeType** = NodeType of the transmitter
  - ▶ Always 0h for MASTER devices
- **DEST NodeType** = NodeType of the receiver(s)
  - ▶ Used to implement NodeType filtering which allows to address only selected devices within a broadcast/group message

|                  | DEST Node Type               | Description   |
|------------------|------------------------------|---|
| Any devices      | 0000                         | All devices will process the message                          |
| Selected devices | See §3.2.2<br>NodeType table | Only devices with specified NodeType will process the message |

## 5.4 SOURCE@ / DEST@

- **SOURCE@** = NodeID of the transmitter
- **DEST@** = NodeID of the receiver

Different addressing modes are available, depending on the values of SOURCE@ and DEST@ fields.

|                       | Bytes 4 to 6      | Bytes 7 to 9   | Description  |
|-----------------------|-------------------|----------------|--|
| <b>Point to Point</b> | SOURCE@ = NodeID  | DEST@ = NodeID | Only the device with NodeID = DEST@ will execute the message   |
| <b>Group</b>          | SOURCE@ = GroupID | 000000h        | All nodes belonging to the group will execute the message (i.e. all nodes with GroupID present in their group table) |
| <b>Broadcast</b>      | SOURCE@ = NodeID  | FFFFFFh        | All nodes on the bus will execute the message  |



### Warning:

SOURCE@ and DEST@ are LSBF



### Example:

If a Point-to-Point communication is initiated by NodeID = 05:04:03 (as shown on the device label) to NodeID = 00:01:02, then the SOURCE@ and DEST@ fields are coded as follow:

| Byte 4  | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Byte 9 |
|---------|--------|--------|--------|--------|--------|
| SOURCE@ |        |        | DEST@  |        |        |
| 03h     | 04h    | 05h    | 02     | 01     | 00     |

## 5.5 DATA

| Byte 10 | ... | Byte n-2 |
|---------|-----|----------|
| DATA 0  | ... | DATA x   |

Chapter §6 gives the DATA structure and length for every message.

Some DATA fields are marked as “*Reserved*”: they must be present in the message and should be set to 00h or FFh.



### Warning:

When receiving a message from a device, the actual DATA length can be longer than the value in the “DATA length” cell.

⇒ The “DATA length” should be considered as the *minimum* DATA length.

## 5.6 CHECKSUM





The checksum is only a basic error detection algorithm, without any error correction capabilities. It's calculated by adding the complement of every byte in the frame.

| Byte n-1 | Byte n |
|----------|--------|
| CHECKSUM |        |





$$\text{CHECKSUM} = (\text{Byte } 1 + \dots + \text{Byte } n-2)$$

## 6 Messages

When a MASTER / SLAVE communication is initiated, the following table lists the messages that can be sent by each device:

|                   | SET_XXX   | CTRL_XXX  | GET_XXX   | POST_XXX  | ACK / NACK                |
|-------------------|---|---|---|---|---------------------------|
| Used for...       | Configuration   | Control   | Information Request   | Information Report  | Acknowledgment and Errors |
| Represented by... |  |  |  |  | n/a                       |
| MASTER            | ✓   | ✓   | ✓   | ✗   | ✗                         |
| SLAVE             | ✗   | ✗   | ✗   | ✓   | ✓                         |

Summary of available messages for the MASTER:

|   |  |  |  +  | Messages           |
|---|---|---|---|--------------------|
| <b>Device Management</b>                      |   |   |   |                    |
| <a href="#">Device NodeID</a>                 | ✗   | ✗   | ✓   | NODE_ADDR          |
| <a href="#">Group Configuration</a>           | ✓   | ✗   | ✓   | GROUP_ADDR         |
|   |   |   |   |                    |
| <b>Device Information</b>                     |   |   |   |                    |
| <a href="#">Firmware Revision</a>             | ✗   | ✗   | ✓   | NODE_APP_VERSION   |
| <a href="#">User Defined Text Label</a>       | ✓   | ✗   | ✓   | NODE_LABEL         |
| <a href="#">Actuator's manufacturing date</a> | ✗   | ✗   | ✓   | NODE_SERIAL_NUMBER |
|   |   |   |   |                    |
| <b>Device Configuration</b>                   |   |   |   |                    |
| <a href="#">HMI management</a>                | ✓   | ✗   | ✓   | LOCAL_UI           |
| <a href="#">Intermediate Positions</a>        | ✓   | ✗   | ✓   | MOTOR_IP           |
| <a href="#">Speed Adjustment (DC motors)</a>  | ✓   | ✗   | ✓   | MOTOR_SPEED        |
| <a href="#">Lock Network Commands</a>         | ✓   | ✗   | ✓   | NETWORK_LOCK       |
|   |   |   |   |                    |
| <b>Device Control</b>                         |   |   |   |                    |
| <a href="#">Move to Position</a>              | ✗   | ✓   | ✗   | MOVETO             |
| <a href="#">Stop</a>                          | ✗   | ✓   | ✗   | STOP               |
| <a href="#">Wink</a>                          | ✗   | ✓   | ✗   | WINK               |

|  |   |   |   |                 |
|--|---|---|---|-----------------|
| <b>Device Status</b>                     |   |   |   |                 |
| <a href="#">Motor Position</a>           | ✘ | ✘ | ✓ | MOTOR_POSITION  |
| <a href="#">Motor Status</a>             | ✘ | ✘ | ✓ | MOTOR_STATUS    |
|  |   |   |   |                 |
| <b>Device Reset</b>                      |   |   |   |                 |
| <a href="#">Factory default settings</a> | ✓ | ✘ | ✘ | FACTORY_DEFAULT |

## 6.1 Device management

### 6.1.1 Device NodeID

#### GET\_NODE\_ADDR (40h)

| MSG | Name          | DATA Length |
|-----|---------------|-------------|
| 40h | GET_NODE_ADDR | 0           |



**Warning:**

When a lot of devices are available on the bus, there is no guarantee that replies from all devices will be received.

#### POST\_NODE\_ADDR (60h)

| MSG | Name           | DATA Length |
|-----|----------------|-------------|
| 60h | POST_NODE_ADDR | 0           |

**Remark:**

No data needed; address is included in message header.

## 6.1.2 Group Configuration

### SET\_GROUP\_ADDR (51h)

| MSG | Name           | DATA Length |
|-----|----------------|-------------|
| 51h | SET_GROUP_ADDR | 4           |

| DATA       | TYPE    | MIN | MAX | Description              |
|------------|---------|-----|-----|--------------------------|
| GroupIndex | 8-bits  | 0   | 15  | Entry in the group table |
| GroupID    | 24-bits | N/a | N/a | Associated group address |

### GET\_GROUP\_ADDR (41h)

| MSG | Name           | DATA Length |
|-----|----------------|-------------|
| 41h | GET_GROUP_ADDR | 1           |

| DATA       | TYPE   | MIN | MAX | Description              |
|------------|--------|-----|-----|--------------------------|
| GroupIndex | 8-bits | 0   | 15  | Entry in the group table |

### POST\_GROUP\_ADDR (61h)

| MSG | Name            | DATA Length |
|-----|-----------------|-------------|
| 61h | POST_GROUP_ADDR | 4           |

| DATA       | TYPE    | MIN | MAX | Description              |
|------------|---------|-----|-----|--------------------------|
| GroupIndex | 8-bits  | 0   | 15  | Entry in the group table |
| GroupID    | 24-bits | n/a | n/a | Associated group address |

### 6.1.3 Acknowledgement and Errors

#### ↩ ACK (7Fh)

| MSG | Name | DATA Length |
|-----|------|-------------|
| 7Fh | ACK  | 0           |

#### Remark

This message is only sent when ACK bit is set to 1 in the request (CTRL, GET or SET message).

#### ↩ NACK (6Fh)

| MSG | Name | DATA Length |
|-----|------|-------------|
| 6Fh | NACK | 1           |

| DATA      | TYPE   | MIN | MAX | Description |
|-----------|--------|-----|-----|-------------|
| ErrorCode | 8-bits | 01h | FFh |             |

| ErrorCode | Description                   | Remarks  |
|-----------|-------------------------------|--|
| 01h       | Data out of range             | DATA fields values are not within expected range |
| 10h       | Unknown message               | MSG identifier is unknown                        |
| 11h       | Message Length Error          | Message length is below minimum value            |
| FFh       | Busy – Cannot process message |  |

#### Remarks

- This message is sent when ACK is requested but an error is detected during data analysis.
- Above mentioned NACK values are implemented in all products.



## 6.2 Device Information

### 6.2.1 Firmware Revision

#### GET\_NODE\_APP\_VERSION (74h)

| MSG | Name                 | DATA Length |
|-----|----------------------|-------------|
| 74h | GET_NODE_APP_VERSION | 0           |

#### POST\_NODE\_APP\_VERSION (75h)

| MSG | Name                  | DATA Length |
|-----|-----------------------|-------------|
| 75h | POST_NODE_APP_VERSION | 6           |

| DATA            | TYPE         | MIN | MAX | Description             |
|-----------------|--------------|-----|-----|-------------------------|
| App_Reference   | 24-bits      | n/a | n/a | Firmware Part Number    |
| App_IndexLetter | 8-bits ASCII | 41h | 5Ah | Firmware major revision |
| App_IndexNumber | 8-bits       | N/a | N/a | Firmware Revision       |
| Reserved        | 8-bits       | N/a | N/a |                         |

#### **Example**

Application software reference 5063486A02 is coded as follows:

| App_Reference | App_IndexLetter | App_IndexNumber |
|---------------|-----------------|-----------------|
| 4Dh 43h 3Eh   | 41h             | 02h             |

## 6.2.2 User-defined Text Label

- A text label can be assigned to each device for identification purpose. It has no effect on the behavior of products or communications on the bus.

### SET\_NODE\_LABEL (55h)

| MSG | Name           | DATA Length |
|-----|----------------|-------------|
| 55h | SET_NODE_LABEL | 16          |

| DATA  | TYPE   | MIN | MAX | Description |
|-------|--------|-----|-----|-------------|
| Label | String | n/a | n/a |             |



#### **Warning:**

The DATA length is always 16 characters. Fill with space if the actual string is shorter.

### GET\_NODE\_LABEL (45h)

| MSG | Name           | DATA Length |
|-----|----------------|-------------|
| 45h | GET_NODE_LABEL | 0           |

### POST\_NODE\_LABEL (65h)

| MSG | Name            | DATA Length |
|-----|-----------------|-------------|
| 65h | POST_NODE_LABEL | 16          |

| DATA  | TYPE   | MIN | MAX | Description |
|-------|--------|-----|-----|-------------|
| Label | String | n/a | n/a |             |

## 6.2.3 Actuator's manufacturing date

- Each device contains a serial number made of the node ID, a manufacturer ID and the year and week of the year of production

### GET\_NODE\_SERIAL\_NUMBER (4Ch)

| MSG | Name                   | DATA Length |
|-----|------------------------|-------------|
| 4Ch | GET_NODE_SERIAL_NUMBER | 0           |

### POST\_NODE\_SERIAL\_NUMBER (6Ch)

| MSG | Name                    | DATA Length |
|-----|-------------------------|-------------|
| 6Ch | POST_NODE_SERIAL_NUMBER | 12          |

| DATA         | TYPE   | MIN | MAX | Description |
|--------------|--------|-----|-----|-------------|
| SerialNumber | String | n/a | n/a |             |

Coded in ASCII as follows :

| Byte 0  | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6          | Byte 7 | Byte 8 | Byte 9 | Byte 10 | Byte 11 |
|---------|--------|--------|--------|--------|--------|-----------------|--------|--------|--------|---------|---------|
| Node ID |        |        |        |        |        | Manufacturer ID |        | Year   |        | Week    |         |

## 6.3 Device Configuration

### 6.3.1 HMI Management

- HMI stands for any local user interface items such as buttons located on motor head, LEDs, DCT inputs and Bluetooth feature.

#### ✳ SET\_LOCAL\_UI (17h)

| MSG | Name         | DATA Length |
|-----|--------------|-------------|
| 17h | SET_LOCAL_UI | 3           |

| DATA     | TYPE   | MIN | MAX       | Description                              |
|----------|--------|-----|-----------|--|
| Function | 8-bits | 00h | 01h       |  |
| UI_Index | 8-bits | 00h | See Table |  |
| Priority | 8-bits | 00h | FFh       | Greater number indicates higher priority |

| Function | Description     | Remarks                          |
|----------|-----------------|----------------------------------|
| 00h      | Enable / Unlock | Enables or Unlocks the feature   |
| 01h      | Disable / Lock  | Disables or Locks the feature    |
| Others   | Invalid         | Returns NACK (DATA_OUT_OF_RANGE) |

| UI_Index | Description                                    | Remarks                          |
|----------|--|----------------------------------|
| 00h      | All Local controls and feedbacks               |                                  |
| 01h      | DCT input                                      |                                  |
| 02h      | Local stimuli (e.g., radio pairing pushbutton) |                                  |
| 03h      | Local Radio access (e.g., Bluetooth)           |                                  |
| 04h      | Touch Motion feature                           |                                  |
| 05h      | LEDs   |                                  |
| Others   | Invalid  | Returns NACK (DATA_OUT_OF_RANGE) |

#### Remarks

- When an item is disabled, all actions/feedback related to this item are ignored and/or switched off until it's enabled again.
- Each UI can be locked with a different priority level
- Lock may be re-set or removed by another SET\_LOCAL\_UI message:
  - When UI\_Index = 00h, priority shall be equal to or higher than the highest of all lock levels otherwise it shall return a NACK (LOW\_PRIORITY).
  - For other UI\_Index values, priority shall be equal to or higher than the corresponding lock level otherwise it shall return a NACK (LOW\_PRIORITY).
- LOCAL\_UI status upon power failure:
  - DCT / Local Stimuli => Not saved / Not restored after power-up
  - All other items => Always saved then restored after power-up.
- Default status (=Factory Default): all UI enabled

① *GET\_LOCAL\_UI (27h)*

| MSG | Name         | DATA Length |
|-----|--------------|-------------|
| 27h | GET_LOCAL_UI | 1           |

| DATA     | TYPE   | MIN | MAX    | Description                      |
|----------|--------|-----|--------|----------------------------------|
| Ui_Index | 8-bits | 01h | UI_MAX | Refer to UI list in SET_LOCAL_UI |

② *POST\_LOCAL\_UI (37h)*

| MSG | Name          | DATA Length |
|-----|---------------|-------------|
| 37h | POST_LOCAL_UI | 5           |

| DATA        | TYPE    | MIN     | MAX     | Description                                     |
|-------------|---------|---------|---------|---|
| Status      | 8-bits  | 00h     | 01h     |   |
| Source_Addr | 24-bits | 000000h | FFFFFFh | NodeID of the device that sent the lock command |
| Priority    | 8-bits  | 00h     | Ffh     | Greater number indicates higher priority        |

| Status | Description        | Remarks |
|--------|--------------------|---------|
| 00h    | Enabled / Unlocked |         |
| 01h    | Disabled / Locked  |         |
| Others | Ignored            |         |

**Remarks**

- When Device is enabled / unlocked, both "Source\_Addr" and "Priority" are reset to 0x000000 and 0x00 respectively

## 6.3.2 Intermediate Positions

### SET\_MOTOR\_IP (15h)

| MSG | Name         | DATA Length |
|-----|--------------|-------------|
| 15h | SET_MOTOR_IP | 4 to 6      |

| DATA           | TYPE    | MIN   | MAX                  | Description   |
|----------------|---------|-------|----------------------|---|
| Function       | 8-bits  | 00h   | 0Bh                  |   |
| IP_Index       | 8-bits  | 1     | 16                   |   |
| Value_Position | 16-bits | 0000h | See Function Remarks |   |
| Value_Tilting  | 16-bits | 0000h | See Function Remarks | Can be omitted if application is not a venetian blind |

| Function | Description  | Remarks   |
|----------|--|---|
| 00h      | Delete IP  | <b>Value_xxx</b> are ignored<br>If IP doesn't exist, returns NACK (IP_NOT_SET)                      |
| 01h      | Set IP at current position                                     | <b>Value_xxx</b> are ignored  |
| 03h      | Set IP at the specified position (in %)                        | <b>Value_Tilting</b> is ignored   |
| 04h      | Divide the Full range with the given number of IPs             | <b>Value_Position</b> contains the IP count<br><b>IP_Index</b> and <b>Value_Tilting</b> are ignored |
| 05h      | Set IP at current position and angle                           | <b>Value_xxx</b> are ignored  |
| 0Ah      | Set IP at the specified position (in %) and angle (in %)       |   |
| 0Bh      | Set IP at the specified position (in %) and angle (in degrees) |   |

#### Remarks

- Default value for all IPs is FFFFh, meaning they are not yet set.
- Setting an IP out of limits range is not allowed.
- Function 04h set the 'x' first IPs to equally separated positions within limits range from top position to bottom position.
  - Example:
    - Set 2 IPs => IP1 to 33% - IP2 to 66%.
    - Set 3 IPs => IP1 to 25% - IP2 to 50% - IP3 to 75%.
  - Existing IPs are overwritten

### GET\_MOTOR\_IP (25h)

| MSG | Name         | DATA Length |
|-----|--------------|-------------|
| 25h | GET_MOTOR_IP | 1           |

| DATA     | TYPE   | MIN | MAX | Description |
|----------|--------|-----|-----|-------------|
| IP_Index | 8-bits | 1   | 16  |             |

### POST\_MOTOR\_IP (35h)

| MSG | Name          | DATA Length |
|-----|---------------|-------------|
| 35h | POST_MOTOR_IP | 4 to 9      |

| DATA                   | TYPE    | MIN | MAX | Description                |
|------------------------|---------|-----|-----|----------------------------|
| IP_index               | 8-bits  | 1   | 16  |                            |
| Reserved               | 16-bits | n/a | n/a |                            |
| IP_position_percentage | 8-bits  | 0   | 100 | FFh if IP position not set |
| Reserved               | 16-bits | n/a | n/a |                            |
| Reserved               | 8-bits  | n/a | n/a |                            |
| IP_angle_degree        | 16-bits | 0   | 180 | 8000h if IP angle not set  |

### 6.3.3 Speed Adjustment

- Speed adjustment is only available on DC motors.

#### SET\_MOTOR\_ROLLING\_SPEED (13h)

| MSG | Name                    | DATA Length |
|-----|-------------------------|-------------|
| 13h | SET_MOTOR_ROLLING_SPEED | 3           |

| DATA       | TYPE   | MIN                     | MAX | Description                          |
|------------|--------|-------------------------|-----|--------------------------------------|
| UP_Speed   | 8-bits | See Technical Datasheet |     | Speed during UP movement (rpm)       |
| DOWN_Speed | 8-bits | See Technical Datasheet |     | Speed during DOWN movement (rpm)     |
| Slow_Speed | 8-bits | See Technical Datasheet |     | Speed for adjustment movements (rpm) |

#### Remarks

- Default Speed and speed range are not the same for all motors. Refer to device technical datasheet.

#### GET\_MOTOR\_ROLLING\_SPEED (23h)

| MSG | Name                    | DATA Length |
|-----|-------------------------|-------------|
| 23h | GET_MOTOR_ROLLING_SPEED | 0           |

#### POST\_MOTOR\_ROLLING\_SPEED (33h)

| MSG | Name                     | DATA Length |
|-----|--------------------------|-------------|
| 33h | POST_MOTOR_ROLLING_SPEED | 3           |

| DATA       | TYPE   | MIN                     | MAX | Description                     |
|------------|--------|-------------------------|-----|---------------------------------|
| UP_Speed   | 8-bits | See Technical Datasheet |     | Speed during UP movement        |
| DOWN_Speed | 8-bits | See Technical Datasheet |     | Speed during DOWN movement      |
| Slow_Speed | 8-bits | See Technical Datasheet |     | Speed for adjustments movements |

### 6.3.4 Lock Network Commands

- Lock feature gives the ability to prevent any commands from the network to operate a device. A priority level is available to allow high-priority controls to execute commands even if devices are locked.

#### SET\_NETWORK\_LOCK (16h)

| MSG | Name             | DATA Length |
|-----|------------------|-------------|
| 16h | SET_NETWORK_LOCK | 2           |

| DATA     | TYPE   | MIN | MAX | Description                              |
|----------|--------|-----|-----|--|
| Function | 8-bits | 00h | 04h |  |
| Priority | 8-bits | 00h | FFh | Greater number indicates higher priority |

| Function | Description                               | Remarks                          |
|----------|---|----------------------------------|
| 00h      | Unlock                                    | Unlock device                    |
| 01h      | Lock                                      | Lock device at current position  |
| 03h      | Save NETWORK_LOCK upon power cycle        | Priority is ignored              |
| 04h      | Do not save NETWORK_LOCK upon power cycle | Priority is ignored              |
| Others   | Invalid                                   | Returns NACK (DATA_OUT_OF_RANGE) |

#### Remarks

- When network is locked, only CTRL\_NETWORK\_LOCK control message with equal or higher priority level will be accepted.
  - All other messages involving a movement or changing the limits are rejected: CTRL\_XXX functions, SET\_MOTOR\_LIMITS and SET\_TILT\_LIMITS
  - NACK (NODE\_IS\_LOCKED) is returned
- Lock may be re-set or removed by another SET\_NETWORK\_LOCK or CTRL\_NETWORK\_LOCK message with equal or higher priority level.
- When SET\_NETWORK\_LOCK with function 03h (Save) is received by a motor:
  - The highest NETWORK\_LOCK (if any) is saved at power off.
  - At power on, the highest NETWORK\_LOCK will be restored.
- When SET\_NETWORK\_LOCK with function 04h (Do not save) is received by a motor:
  - NETWORK\_LOCK is not saved at power off
  - At power-on, no NETWORK\_LOCK will be restored → CTRL\_XXX are always enabled after power-on.
- Default value (=Factory Default): Do Not Save
- Source\_Addr is only saved when function 01h (Lock) is received.

① *GET\_NETWORK\_LOCK (26h)*

| MSG | Name             | DATA Length |
|-----|------------------|-------------|
| 26h | GET_NETWORK_LOCK | 0           |

↳ *POST\_NETWORK\_LOCK (36h)*

| MSG | Name              | DATA Length |
|-----|-------------------|-------------|
| 36h | POST_NETWORK_LOCK | 6           |

|     | DATA        | TYPE    | MIN     | MAX     | Description  |
|-----|-------------|---------|---------|---------|--|
| Req | Status      | 8-bits  | 00h     | 01h     |  |
| Req | Source_Addr | 24-bits | 000000h | FFFFFFh | NodeID of the device that sent the lock command              |
| Req | Priority    | 8-bits  | 00h     | FFh     | Greater number indicates higher priority                     |
| Req | Saved       | 8-bits  | 00h     | 01h     | Indicate if lock will be saved and restored upon power cycle |

| Status | Description | Remarks |
|--------|-------------|---------|
| 00h    | Unlocked    |         |
| 01h    | Locked      |         |
| Others | Ignored     |         |

| Saved  | Description                              | Remarks |
|--------|--|---------|
| 00h    | Lock will not be restored on power cycle |         |
| 01h    | Lock will be restored on power cycle     |         |
| Others | Ignored                                  |         |

**Remarks**

- When Device is unlocked, both “Source\_Addr” and “Priority” are reset to 0x000000 and 0x00 respectively



## 6.4 Device Control

### 6.4.1 Move to Position

#### ▶ CTRL\_MOVE\_TO (03h)

| MSG | Name         | DATA Length |
|-----|--------------|-------------|
| 03h | CTRL_MOVE_TO | 6           |

| DATA     | TYPE    | MIN             | MAX | Description  |
|----------|---------|-----------------|-----|--|
| Function | 8-bits  | 00h             | 04h |  |
| Position | 16-bits | See table below |     |  |
| Reserved | 8-bits  | n/a             | n/a |  |
| Angle    | 16-bits | -90             | 90  | Can be omitted if application mode is not Venetian Blind |

| Function | Description                                 | Remarks                              |
|----------|---|--------------------------------------|
| 00h      | ... DOWN limit                              | Position is ignored                  |
| 01h      | ... UP limit                                | Position is ignored                  |
| 02h      | ... Intermediate Position                   | Position contains IP index (0 to 15) |
| 04h      | ... Position (in %)                         | Position contains % value (0 to 100) |
| 0Ch      | ... Position (in %) and Angle (in %)        |                                      |
| 0Dh      | ... Position (in %) and Angle (in degrees)  |                                      |
| 0Fh      | ... Current Position and Angle (in %)       | Position is ignored (tilt only)      |
| 10h      | ... Current Position and Angle (in degrees) | Position is ignored (tilt only)      |

### 6.4.2 Stop

#### ▶ CTRL\_STOP (02h)

| MSG | Name      | DATA Length |
|-----|-----------|-------------|
| 02h | CTRL_STOP | 1           |

| DATA     | TYPE   | MIN | MAX | Description |
|----------|--------|-----|-----|-------------|
| Reserved | 8-bits | n/a | n/a |             |

#### Remarks

- Motor is immediately stopped without speed ramp-down

### 6.4.3 Wink

#### ▶ CTRL\_WINK (05h)

| MSG | Name      | DATA Length |
|-----|-----------|-------------|
| 05h | CTRL_WINK | 0           |

#### Remarks

- In Venetian mode, the final angle can't be guaranteed

## 6.5 Device Status

### 6.5.1 Motor Position

#### GET\_MOTOR\_POSITION (0Ch)

| MSG | Name               | DATA Length |
|-----|--------------------|-------------|
| 0Ch | GET_MOTOR_POSITION | 0           |

#### POST\_MOTOR\_POSITION (0Dh)

| MSG | Name                | DATA Length |
|-----|---------------------|-------------|
| 0Dh | POST_MOTOR_POSITION | 5 to 11     |

| DATA                | TYPE    | MIN      | MAX        | Description |
|---------------------|---------|----------|------------|-------------|
| Position_pulse      | 16-bits | UP_LIMIT | DOWN_LIMIT |             |
| Position_percentage | 8-bits  | 0        | 100        |             |
| Tilting_Percentage  | 8-bits  | 0        | 100        |             |
| IP                  | 8-bits  | 01h      | IP_MAX     |             |
| Reserved            | 16-bits |          |            |             |
| Tilting_Degrees     | 16-bits | 0        | 180        |             |
| Reserved            | 16-bits |          |            |             |

#### Remarks

- The position is sent even if the motor is running
- If the position does not correspond to any IP, returned value for IP is FFh. Motor may consider to be at an IP position even if it is above or below of a few pulses. Tolerance around this point may be variable and depends on the motor.
- If the position corresponds to several IP, first IP of the matching value on the list is returned.

## 6.5.2 Motor Status

- Motor status indicates the current state of the motor, the direction of current or previous movement, the origin of the last command and some additional information explaining the status.

### ① GET\_MOTOR\_STATUS (0Eh)

| MSG | Name             | DATA Length |
|-----|------------------|-------------|
| 0Eh | GET_MOTOR_STATUS | 0           |

### ① POST\_MOTOR\_STATUS (0Fh)

| MSG | Name              | DATA Length |
|-----|-------------------|-------------|
| 0Fh | POST_MOTOR_STATUS | 4           |

| DATA      | TYPE   | MIN             | MAX | Description             |
|-----------|--------|-----------------|-----|-------------------------|
| Status    | 8-bits | See table below |     | Status of the motor     |
| Direction | 8-bits |                 |     | Last rotation direction |
| Source    | 8-bits |                 |     | Origin of the command   |
| Cause     | 8-bits |                 |     | Additional information  |

| Status | Description | Remarks                                    |
|--------|-------------|--|
| 00h    | Stopped     |  |
| 01h    | Running     | During movement                            |
| 02h    | Blocked     | Cannot move (thermal protection, obstacle) |
| 03h    | Locked      | Locked by another device (NETWORK_LOCK)    |

| Direction | Description | Remarks   |
|-----------|-------------|---|
| 00h       | Going DOWN  | If motor is stopped, last movement direction is indicated |
| 01h       | Going UP    |   |
| FFh       | Unknown     |   |

| Source | Description     | Remarks  |
|--------|-----------------|--|
| 00h    | Internal        | Limit/IP/Position reached, Over-current, obstacle detection, thermal protection, ... |
| 01h    | Network message | Any message received from the SDN bus  |
| 02h    | Local UI        | DCT, Local stimulus, local wireless  |

| Cause | Description             | Remarks   |
|-------|-------------------------|---|
| 00h   | Target reached          | Reached limit or IP or already there                                    |
| 01h   | Explicit command        | Network or Local UI command   |
| 02h   | Wink                    |   |
| 20h   | Obstacle detection      |   |
| 21h   | Over-current protection |   |
| 22h   | Thermal protection      |   |
| 30h   | Run time exceeded       | Continuous runtime exceeded limit                                       |
| 32h   | Timeout exceeded        | When using CTRL_MOVE and more than 2min. elapsed => canceled adjustment |
| FFh   | Reset / PowerUp         | Power recycled / No command after startup                               |

## 6.6 Device Reset

### 6.6.1 Factory default settings

- Recall factory default settings.

#### SET\_FACTORY\_DEFAULT (1Fh)

| MSG | Name                | DATA Length |
|-----|---------------------|-------------|
| 1Fh | SET_FACTORY_DEFAULT | 1           |

| DATA     | TYPE   | MIN | MAX | Description |
|----------|--------|-----|-----|-------------|
| Function | 8-bits | 00h | 17h |             |

| Function | Description                        | Remarks                  |
|----------|------------------------------------|--------------------------|
| 00h      | All settings to factory default    |                          |
| 01h      | Clear all Group addresses          | Group address = 00:00:00 |
| 15h      | Delete all IPs                     | IPs = FFFFh              |
| 17h      | Clear all locks + Lock Save Status |                          |