

ekinex
CONTROL YOUR LIVING SPACE

Application manual

**Configurable
KNX Logic module
EK-LM2-TP**

Contents

| | | |
|------|---|----|
| 1. | Scope of the document..... | 4 |
| 2. | Product description | 5 |
| 3. | Switching, display and connection elements | 5 |
| 3.1 | External connections..... | 5 |
| 4. | Configuration | 5 |
| 5. | Commissioning | 6 |
| 5.1 | Factory reset procedure..... | 6 |
| 6. | Function description..... | 7 |
| 6.1 | Common block features | 7 |
| 6.2 | Block type: Logic..... | 8 |
| 6.3 | Block type: Arithmetic / Math | 8 |
| 6.4 | Block type: Range..... | 9 |
| 6.5 | Block type: Mapping..... | 10 |
| 6.6 | Block type: Conversion | 10 |
| 6.7 | Block type: Timer | 10 |
| 6.8 | Block type: Compare..... | 10 |
| 6.9 | Block type: Multiplexer | 10 |
| 6.10 | Power-on behaviour..... | 11 |
| 7. | Device settings | 12 |
| 7.1 | Generic device parameters..... | 12 |
| 7.2 | Parameters for inputs..... | 12 |
| 7.3 | Parameters for outputs | 13 |
| 7.4 | Core parameters for block type: Logic..... | 14 |
| 7.5 | Core parameters for block type: Arithmetic / Math | 14 |
| 7.6 | Core parameters for block type: Range..... | 15 |
| 7.7 | Core parameters for block type: Mapping..... | 15 |
| 7.8 | Core parameters for block type: Conversion | 15 |
| 7.9 | Core parameters for block type: Timer | 16 |
| 7.10 | Core parameters for block type: Compare..... | 16 |
| 7.11 | Core parameters for block type: Multiplexer | 16 |
| 8. | KNX Communication objects table | 17 |
| 8.1 | Generic device comm. objects..... | 17 |
| 8.2 | Comm. objects for block type: Logic..... | 17 |
| 8.3 | Comm. objects for block type: Arithmetic / Math | 17 |
| 8.4 | Comm. objects for block type: Range..... | 18 |
| 8.5 | Comm. objects for block type: Mapping..... | 18 |
| 8.6 | Comm. objects for block type: Convert..... | 18 |
| 8.7 | Comm. objects for block type: Timer | 19 |
| 8.8 | Comm. objects for block type: Compare..... | 19 |
| 8.9 | Comm. objects for block type: Multiplexer | 19 |
| 9. | Block diagrams | 21 |
| 9.1 | Block type: Logic..... | 22 |
| 9.2 | Block type: Arithmetic / Math | 23 |
| 9.3 | Block type: Range..... | 24 |

| | | |
|--------|--|----|
| 9.4 | Block type: Mapping..... | 25 |
| 9.5 | Block type: Conversion | 26 |
| 9.6 | Block type: Timer | 27 |
| 9.7 | Block type: Compare..... | 28 |
| 9.8 | Block type: Multiplexer | 29 |
| 10. | Appendix..... | 30 |
| 10.1 | Warnings..... | 30 |
| 10.2 | Return of defective products..... | 30 |
| 10.2.1 | Devices purchased directly from ekinex®..... | 30 |
| 10.2.2 | Devices purchased through ekinex® resellers..... | 30 |
| 10.3 | Other Information | 30 |

| Revision | Modification | Date | Edited | Verified |
|----------|---------------------------|------------|--------------|----------|
| 0.3 | Changes for Ekinex S.p.A. | 2018-12-05 | G. Schiochet | |
| 0.2 | Minor changes | 2018-11-15 | G. Croci C. | |
| 0.1 | First draft | 2018-09-01 | G.Croci C. | |

DRAFT

1. Scope of the document

This application manual describes application details for the A1.0 release of the ekinex® KNX Logic module EK-LM2-TP.

The document is aimed at the system configurator as a description and reference of device features and application programming. For installation, mechanical and electrical details of the device please refer to the technical description datasheet.

Application manual and application programs for ETS are available for download at www.ekinex.com.

| <i>Item</i> | <i>File name (## = release)</i> | <i>Version</i> | <i>Device rel.</i> | <i>Update</i> |
|---------------------|---------------------------------|----------------|--------------------|---------------|
| Technical datasheet | STEKLM2TP_EN.pdf | - | A1.0 | 01 Sep 2018 |
| Application manual | MAEKLM2TP_EN.pdf | - | | |
| Application program | APEKLM2TP##.knxprod | - | | |

You can access the most up-to-date version of the full documentation for the device using following QR code:

EK-LM2-TP

www.ekinex.com/eklm2tp.cod



2. Product description

The ekinex® Configurable Logic Module **EK-LM2-TP** is a device used to operate on data exchanged on the KNX bus with several processing functions; among these are for instance logical, mathematical, and data conversion functions. The device is not meant to act either as a sensor or as actuator, with the only exception of four auxiliary low-voltage inputs (to be connected with external dry contacts).

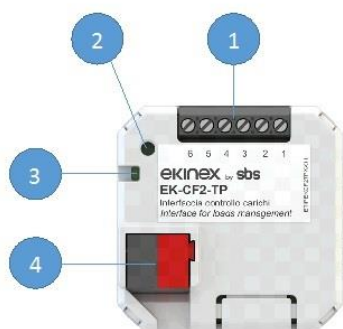
The device is encased in a small module for convenient placement in junction boxes, cabinets or e.g. behind drywall panels. The only required connection is the KNX bus connection.

3. Switching, display and connection elements

The device is equipped with:

- Plug terminal for the KNX bus line connection
- Programming pushbutton and a programming LED
- Screw terminals for auxiliary dry contact, low voltage inputs

3.1 External connections



1. Terminals for the connection of input dry contacts
2. Programming pushbutton
3. Programming mode LED indicator
4. Terminal plug for the KNX bus line

Fig. 1 - Switching, display and connection elements

4. Configuration

The exact functionality of the device depends on the software settings.

In order to configure and commission the device you need ETS4 or later releases and the proper ekinex® application program (named **APEKLM2TP##.knxprod**); this can be downloaded from the ekinex® website www.ekinex.com.

The application program allows the configuration of all working parameters for the device. The device-specific application program has to be loaded into ETS or, as alternative, the whole ekinex® product database can be loaded; at this point, all the instances of the selected device type can be added to the project.

For every single device, ETS allows to set the operating parameters individually for each input as described in detail in the following chapters.

The configuration can, and usually will, be performed completely offline; the actual transfer of the programmed configuration to the device takes place in the commissioning phase as described in the next paragraph.

| Product code | EAN | ETS application software (## = release) | Communication objects (max nr.) | Group addresses (max nr.) |
|--------------|-----|--|---------------------------------------|---------------------------------|
| EK-LM2-TP | | APEKLM2TP##.knxprod | 414 | |

5. Commissioning

After the device has been configured within the ETS project according to user requirements, the commissioning of the device requires the following activities:

- electrically connect the device, as described in the product datasheet, to the bus line on the final network or through a purposely setup network for programming;
- apply power to the bus;
- switch the device operation to programming mode by pressing the programming pushbutton located on the front side of the housing. In this mode of operation, the programming LED is turned on steady;
- upload the configuration (including the physical address) to the device with the ETS program.

At the end of the upload, the operation of the device automatically returns to normal mode; in this mode the programming LED is turned off. Now the device is programmed and ready for use on the bus.

5.1 Factory reset procedure

If, for any reason, a reset of the device to factory parameters is required, this procedure should be followed:

- Disconnect the KNX bus by removing the connector from its seat;
- Reinsert the KNX connector *while keeping the programming button pressed*. The programming LED should start flashing quickly;
- release the programming button and switch off the device by disconnecting the KNX bus once again.

At this point the device will be in its original factory condition.



Warning: the factory reset procedure erases all parameter configuration and address settings.

In order to be deployed again, the device must first be reprogrammed as required through ETS.

6. Function description

The device has a number of configurable blocks of different type. Each block has a number of inputs, outputs and configuration parameters; it performs a certain operation on its inputs (communication objects), according to the configuration set through the parameters, yielding the corresponding number and type of outputs (which are other communication objects).

The behaviour of the block might be influenced by values set through other communication objects depending on block type and configuration.

There are eight different types of blocks, listed in the table below; for each block, more than one instance is available if enabled. The number of available instances is listed on the corresponding line.

Blocks also have additional features, such as e.g. the possibility of "locking" the status of inputs or outputs.

The operation of each block, as well as additional features, will be described in the following sections.

| <i>Block type</i> | <i>Summary description</i> | <i>Number of available instances</i> | <i>Number of INPUTS</i> | <i>Number of OUTPUTS</i> |
|-------------------|---|--------------------------------------|-------------------------|--------------------------|
| Logic | Combines a number of boolean inputs according to logic operations; yields a single boolean output as a result. | 8 | 2..8 | 1 |
| Arithmetic / Math | Performs mathematical computations on one or more input values (according to selected operation) | 4 | 1 to 4 | 1 |
| Range | Performs a scaling of the input value to a different range of values, or a boundary limit | 8 | 1 | 1 |
| Mapping | Translates a set of input values (up to 8 points) to a different set of output values | 2 | 1 | 1 |
| Conversion | Converts a CO to another CO with a different KNX data type (DPT) | 8 | 1 | 1 |
| Timer | Implements a one-shot (monostable) or free-running (astable) timer; depending on configuration, the timer can be reset, retriggered or require rearming | 8 | 1/2 | 1 |
| Compare | Returns the result of a comparison (equal, larger, smaller etc.) between two input values | 8 | 2 | 1 |
| Multiplexer | Copies the value of one input object to either 1 of N output objects, or to all objects selected in the output set. | 2 | 1 | 1..8 |

6.1 Common block features

With a few exceptions (described in the respective paragraphs), every block has following features:

- One or more inputs, depending on the type of block and on the "enable" parameter. Each input has an associated CO; the type (DPT) of this CO can be selected from a wide range of DPTs having different sizes. Depending on the block operation, input (and output) objects might have each a different assigned DPT.

The behaviour of an input object can be chosen for both the "Bus On" and the "After download" events; for each of these events, if the proper option is specified, a value can be assigned as default for the object.

A "lock" object can also be assigned to each input; a locked input appears effectively disabled.

- One or more outputs, depending on the type of block and on the "enable" parameter. Output objects may have different DPTs, like just described for inputs.

The transmission behaviour of an output object can be chosen for both the "Bus On" and the "After download" events.

A "lock" object can also be assigned to each output; a locked output maintains its last value until the lock is waived and a new output value is produced by a change in the inputs (or, in the case of timers, by the variation of the internal status).

The above features will not be listed again in the detailed description of each block (unless there is some difference that needs be highlighted).

6.2 Block type: Logic

The Logic block can have at least two, at most 8 inputs (additional inputs are enabled individually); all input types are generic 1-bit.

In addition to standard features, each input has a "negation" parameter which inverts the value of corresponding input.

There is only one output, which also has a negation parameter.

The available operations are the following:

- OR - the output is the logical (boolean) OR of all enabled inputs
- AND - the output is the logical (boolean) AND of all enabled inputs
- XOR - the output is the logical (boolean) exclusive OR of all enabled inputs (the output is "1" only if an odd number of inputs is "1")
- 1 of N - the output is "1" only if exactly one input is "1"
- Selector - the output has the same state as the input whose number is selected through a communication object. If the selector value is invalid, i.e. is either out of range or corresponds to a disabled input, the output is "0".

6.3 Block type: Arithmetic / Math

The Arithmetic / Math block can have at least two, at most 4 inputs (additional inputs are enabled individually).

In addition to standard features, each input has:

- an "abs" parameter which takes the absolute value of the corresponding input
- a "negation" parameter which inverts the sign of the corresponding input

There is only one output, which also has an "abs" and a negation parameter.

The Arithmetic block is actually composed by two operations, an N-operand operation (applied to inputs) and a unary operation (applied in cascade to the former); a further scaling operation is then applied.

The available N-operand operations are the following:

- + (addition, or subtraction): the inputs are all added, each with their own sign (thus implementing subtraction)

- * (multiplication) - the first two (mandatory) inputs are multiplied
- / (division) - the first input is divided by the second one
- DIV (integer division) - the first input is divided by the second one, and the integer part of the result is taken
- MOD (modulus or remainder) - the first input is integer-divided by the second one, and the remainder of the result is taken
- AVG (average) - the arithmetic average of all enabled inputs is produced
- Selector - the output has the same value as the input whose number is selected through a communication object. If the selector value is invalid, i.e. is either out of range or corresponds to a disabled input, the output is "0".

The output of this block can be subjected to the following unary operations:

- none (i.e. the unary block is skipped)
- SQR() - the output is the square of the intermediate input
- SQRT() - the output is the square root of the intermediate input
- SIN() - COS() - TAN() - the output is the sinus/cosinus/tangens of the intermediate input (in radians)
- DSIN() - DCOS() - DTAN() - the output is the sinus/cosinus/tangens of the intermediate input (in degrees)

Finally, if the corresponding option is selected, the output of the unary operation block can be multiplied with and summed (or subtracted) to two constants specified as parameters.

Notes:

- For the *Division* and *Integer Division* operations, a default value can be specified for the case of division by 0; alternatively, a division by 0 does not trigger the operation (the last result is unchanged).
- Internal conversions make sure that the data type used for calculations has sufficient precision for the specified operation. If different DPTs are specified for inputs or outputs, and they are not carefully chosen, overflow / underflow errors may occur; these are neglected without reporting an error, and therefore the result might be different than expected.
- If the block is required for a unary operation only (i.e. only one input is required), the user should choose a neutral setting for the N-operand operation, e.g. choosing operation "+" and assigning the value "0" to the second mandatory operand.

6.4 Block type: Range

The Range block type accepts one input and one output (of possibly different types).

The input / source has two associated boundary values, upper and lower; this is also true for the output / destination value, depending on the operation.

Possible operations are following:

- Range - Source values in the interval defined by upper and lower source limits are mapped to the values defined by the upper and lower destination limits. Values outside that range are consequently computed (extrapolation);
- Limited range - same as for "Range", except values exceeding the two limits in both directions are "clipped" to the limit value;
- Limit - values outside the two source limits "clipped" to the corresponding limit value; otherwise values are not mapped to different ranges.

6.5 Block type: Mapping

The Mapping block type accepts one input and one output (of possibly different types).

The input / source has up to eight associated values; if the input matches the n-th of these values, the output is set to the n-th value of a corresponding output value set.

Two types of operation are possible:

- Discrete: whenever the input value does not match any of the enabled entries in the table, the output assumes a value specified as default;
- Continuous: If the input is between two values in the input table, the output value is scaled between the two corresponding values in the output table (similarly to what happens for the "Range" block). This requires that the input and output tables are correctly assigned in growing order, otherwise the result might not be the expected one.

6.6 Block type: Conversion

The Conversion block type accepts one input and one output of different types; it simply converts the input value to the same value in a different DPT.

6.7 Block type: Timer

The timer block implements different types of timers; its input(s) are control inputs and therefore are always of boolean type. The timer has these features:

- can be a one-shot timer (monostable) or a "blinker" (astable)
- Has an associated on-delay and an off-delay;
- Timing starts with a trigger input (which is always present). The timer can be configured to be retriggerable (a new trigger restarts the count) or not (a new trigger is ineffective until the count is over);
- An additional "Reset" input can be used for two purposes, both configurable:
 - Actual "reset" - a "1" on the reset input stops the count and halts the timer;
 - Rearm - once a count cycle has ended (for one-shot), the Rearm object must receive a "1" value before a new trigger can be issued.

6.8 Block type: Compare

The Compare block type accepts two inputs of the same type; the result is a boolean object that carries the result of the comparison between the two inputs.

Available comparison operations are:

- equal (=)
- Different (<>)
- higher / higher or equal (> / >=)
- lower / lower or equal (< / <=)

6.9 Block type: Multiplexer

The Multiplexer block type accepts one input and issues up to 8 outputs of the same type.

It basically mirrors the input value to one of the outputs (operation: 1 of n) or several of them (operation: Individual selection). In the former case, one single object carries the number of the only output that must be kept active; in the latter case, each output has its own "enable" boolean object.

6.10 Power-on behaviour

After switching on the bus, which also acts as the power supply for logics, the device becomes fully functional after a very short time (some milliseconds) needed for reinitialization. A further delay is programmable for the device to become active on the bus in order to avoid a bus traffic overload during the first moments of start-up of the whole network.

In case of a bus power failure (voltage lower than 19 V for 1 s or more), the device is switched off; on power-off, all current working values are saved.

As soon as the bus voltage is restored, the device will resume operation in its previous, unless different initialization settings are programmed.

The status of device inputs after either a power-up (Bus On) event or after a download can be set through configuration parameters.

DRAFT

7. Device settings

This section lists all configurable parameters, i.e. all parameters that can be set through the ETS application during commissioning.

The default values for each parameter are printed in **bold**.

In order to make the description more concise, the available parameters will be listed in the same fashion used for the description of block features: the parameters concerning any generic input and output will be described first, followed by the specific parameters for each type of block.

7.1 Generic device parameters

| Parameter name | Conditions | Values |
|---------------------------------|------------|------------------------------|
| External input enabled [#1..#4] | - | No Yes |
| External input negated [#1..#4] | - | No Yes |
| Delay at bus on | - | hh:mm:ss [00:00:02] |

7.2 Parameters for inputs

| Parameter name | Block type | | | | | | | | Values |
|-----------------------|------------|-----|-----|-----|-----|-----|-----|-----|--|
| | Log | Mat | Rng | Map | Cnv | Tim | Cmp | Mpx | |
| Status at bus on | x | | | | | - | | | 0 1 Previous value Wait for telegram Force read value |
| Status at bus on | | x | x | x | x | - | x | x | Assigned value Previous value Wait for telegram Force read value |
| Status after download | x | | | | | | | | 0 1 No change Wait for telegram Force read value |
| Status after download | | x | x | x | x | - | x | x | Assigned value No change Wait for telegram Force read value |
| Value at bus on | | x | x | x | x | - | x | x | (value according to chosen DPT) |
| Value after download | | x | x | x | x | - | x | x | (value according to chosen DPT) |

| Parameter name | Block type | | | | | | | | Values |
|----------------|------------|-----|-----|-----|-----|-----|-----|-----|--|
| | Log | Mat | Rng | Map | Cnv | Tim | Cmp | Mpx | |
| DPT | | x | x | x | x | - | x | x | 8bit unsigned [5.x] 8bit signed [6.x] 16bit unsigned [7.x] 16bit signed [8.x] Float 16b [9.x] 32bit unsigned [12.x] 32bit signed [13.x] Float 32b [14.x] |
| Absolute value | | x | | | | - | | | No Yes |
| Negate | x | x | | | | - | | | No Yes |
| Enabled | x | x | | | | - | | | No Yes |
| Lock enabled | x | x | x | x | x | - | x | x | No Yes |

7.3 Parameters for outputs

| Parameter name | Block type | | | | | | | | Values |
|-----------------------------|------------|-----|-----|-----|-----|-----|-----|-----|--|
| | Log | Mat | Rng | Map | Cnv | Tim | Cmp | Mpx | |
| Transmission at bus on | x | x | x | x | x | x | x | x | None Only if valid Anyway (Previous value) |
| Transmission after download | x | x | x | x | x | x | x | x | None Only if valid Anyway (Previous value) |
| DPT | | x | | x | x | | | | 8bit unsigned [5.x] 8bit signed [6.x] 16bit unsigned [7.x] 16bit signed [8.x] Float 16b [9.x] 32bit unsigned [12.x] 32bit signed [13.x] Float 32b [14.x] |
| Absolute value | | x | | | | | | | No Yes |
| Negate | x | x | | | | x | | | No Yes |
| Enabled | | | | | | | | x | No Yes |
| Lock enabled | x | x | x | x | x | x | x | x | No Yes |

7.4 Core parameters for block type: Logic

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|----------------|---------------------|---|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | OR AND XOR 1 of N Selector |

7.5 Core parameters for block type: Arithmetic / Math

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|-------------------|---|---|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | +/- * / DIV MOD AVG Selector |
| Unary operation | Block enabled = Yes | none SQR() SQRT() SIN() COS() TAN() DSIN() DCOS() DTAN() |
| Scaling | Block enabled = Yes | No Yes |
| Scaling - offset | Block enabled = Yes Scaling = Yes | [numeric value] |
| Scaling - factor | Block enabled = Yes Scaling = Yes | [numeric value] |
| On divide by 0 | Block enabled = Yes | No Change Fixed value |
| Divide by 0 value | Block enabled = Yes On divide by 0 = Fixed value | [numeric value] |

7.6 Core parameters for block type: Range

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|--------------------|---|--|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | Range Limited range Limit |
| From / lower value | Block enabled = Yes | [numeric value] |
| From / upper value | Block enabled = Yes | [numeric value] |
| To / lower value | Block enabled = Yes Operation = Range, Limited Value | [numeric value] |
| To / upper value | Block enabled = Yes Operation = Range, Limited Value | [numeric value] |

7.7 Core parameters for block type: Mapping

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|-------------------------------|---------------------|-------------------------------|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | Discrete Continuous |
| Number of points | Block enabled = Yes | 1..8 [2] |
| Source value [#1..#8] | Block enabled = Yes | [numeric value] |
| Destination value [#1..#8] | Block enabled = Yes | [numeric value] |
| Default value | Block enabled = Yes | [numeric value] |

7.8 Core parameters for block type: Conversion

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|----------------|------------|------------------|
| Block enabled | - | No Yes |

7.9 Core parameters for block type: Timer

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|----------------|---------------------|---|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | One-shot (monostable) Blinker (astable) |
| Rearmable | Block enabled = Yes | No Yes |
| Retriggerable | Block enabled = Yes | No Yes |
| Resettable | Block enabled = Yes | No Yes |
| Output negated | Block enabled = Yes | No Yes |
| ON delay | Block enabled = Yes | hh:mm:ss [00:00:10] |
| OFF delay | Block enabled = Yes | hh:mm:ss [00:00:10] |

7.10 Core parameters for block type: Compare

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|----------------|---------------------|---|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | = <> > >= < <= |

7.11 Core parameters for block type: Multiplexer

This set of parameters is repeated for each available block instance.

| Parameter name | Conditions | Values |
|-----------------|---------------------|---------------------------------------|
| Block enabled | - | No Yes |
| Operation | Block enabled = Yes | 1 of N Individual selection |
| No-select value | Block enabled = Yes | Default value No change |

| Parameter name | Conditions | Values |
|-------------------------------|---------------------|----------|
| Number of outputs [#1..#8] | Block enabled = Yes | 1..8 [2] |

8. KNX Communication objects table

This section lists a summary of all KNX Communication Objects (CO) and corresponding Data Point Types (DPT).

Size and DPT, if not specified, depend on the the settings chosen in the ETS application during commissioning.

For the same reason, according to chosen parameters, not all objects may be visible at all times in the CO listing for the device.

The object numbers listed are those addressing the first block of each individual type; other blocks of the same type will have subsequent CO numbers.

8.1 Generic device comm. objects

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|---------------------|------------|-------|-------|--------------------|--------------|
| Hardware pin status | - | 1 bit | CR-T- | [1.001] DPT_Switch | 1..4 |

8.2 Comm. objects for block type: Logic

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|---------------------|------------|-------|-------|--------------------------------|--------------|
| Input [#1..#8] | - | 1 bit | C-W-U | [1.001] DPT_Switch | 5..12 |
| Input lock [#1..#8] | - | 1 bit | C-W-U | [1.011] DPT_State | 13..20 |
| Output | - | 1 bit | CR-T- | [1.001] DPT_Switch | 21 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 22 |
| MPX selector | - | 8 bit | C-W-U | [5.010] DPT_Value _1_Ucount | 23 |

8.3 Comm. objects for block type: Arithmetic / Math

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|---------------------|------------|-------|-------|-------------------|--------------|
| Input [#1..#4] | - | * | C-W-U | * | 157..160 |
| Input lock [#1..#4] | - | 1 bit | C-W-U | [1.011] DPT_State | 161..164 |

| | | | | | |
|-------------|---|-------|-------|--------------------------------|-----|
| Output | - | * | CR-T- | * | 165 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 166 |
| Selector | - | 8 bit | C-W-U | [5.010] DPT_Value _1_Ucount | 167 |

8.4 Comm. objects for block type: Range

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|-------------|------------|-------|-------|-------------------|--------------|
| Input | - | * | C-W-U | * | 201 |
| Input lock | - | 1 bit | C-W-U | [1.011] DPT_State | 202 |
| Output | - | * | CR-T- | * | 203 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 204 |

8.5 Comm. objects for block type: Mapping

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|-------------|------------|-------|-------|-------------------|--------------|
| Input | - | * | C-W-U | * | 233 |
| Input lock | - | 1 bit | C-W-U | [1.011] DPT_State | 234 |
| Output | - | * | CR-T- | * | 235 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 236 |

8.6 Comm. objects for block type: Convert

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|-------------|------------|-------|-------|-------------------|--------------|
| Input | - | * | C-W-U | * | 241 |
| Input lock | - | 1 bit | C-W-U | [1.011] DPT_State | 242 |
| Output | - | * | CR-T- | * | 243 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 244 |

8.7 Comm. objects for block type: Timer

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|---------------|------------|-------|-------|---------------------|--------------|
| Trigger Input | - | 1 bit | C-W-U | [1.017] DPT_Trigger | 273 |
| Input lock | - | 1 bit | C-W-U | [1.011] DPT_State | 274 |
| Output | - | 1 bit | CR-T- | [1.001] DPT_Switch | 275 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 276 |
| Reset Input | - | 1 bit | C-W-U | [1.017] DPT_Trigger | 277 |

8.8 Comm. objects for block type: Compare

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|---------------|------------|-------|-------|--------------------|--------------|
| Input #1 | - | * | C-W-U | * | 313 |
| Input #2 | - | * | C-W-U | * | 314 |
| Input lock #1 | - | 1 bit | C-W-U | [1.011] DPT_State | 315 |
| Input lock #2 | - | 1 bit | C-W-U | [1.011] DPT_State | 316 |
| Output | - | 1 bit | CR-T- | [1.001] DPT_Switch | 317 |
| Output lock | - | 1 bit | CR-T- | [1.011] DPT_State | 318 |

8.9 Comm. objects for block type: Multiplexer

| Object name | Conditions | Size | Flags | DPT | CO number(s) |
|------------------------|------------|-------|-------|-------------------|--------------|
| Input | - | * | C-W-U | * | 361 |
| Input lock | - | 1 bit | C-W-U | [1.011] DPT_State | 362 |
| Input default value | - | * | C-W-U | * | 363 |
| Output [#1..#8] | - | * | CR-T- | * | 364..371 |
| Output lock [#1..#8] | - | 1 bit | CR-T- | [1.011] DPT_State | 372..379 |
| Output select [#1..#8] | - | 1 bit | C-W-U | [1.011] DPT_State | 380..387 |

| | | | | | |
|----------|---|-------|-------|--------------------------------|-----|
| Selector | - | 8 bit | C-W-U | [5.010] DPT_Value _1_Ucount | 388 |
|----------|---|-------|-------|--------------------------------|-----|

DRAFT

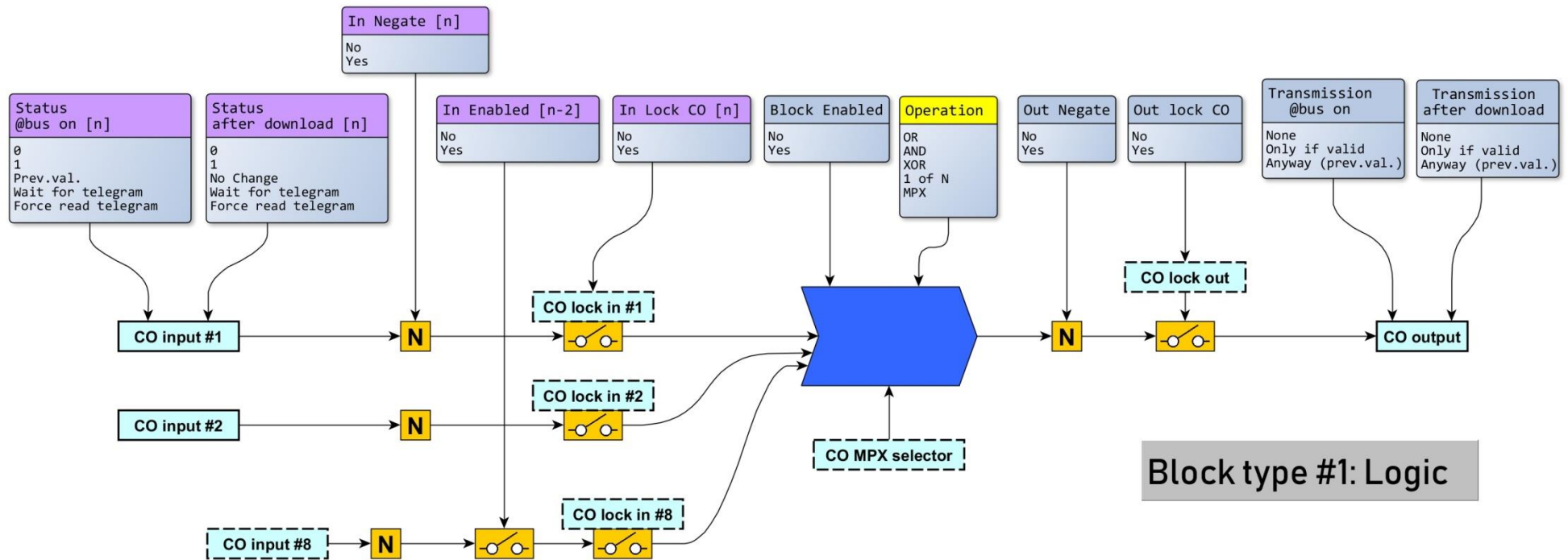
9. Block diagrams

Following sections contain the diagrams for all available block types; parameters and communication objects are also listed.

The purpose of the diagrams is to give a better overview of the block structure and operation; for more exact details please refer to the paragraphs with the textual description of block features.

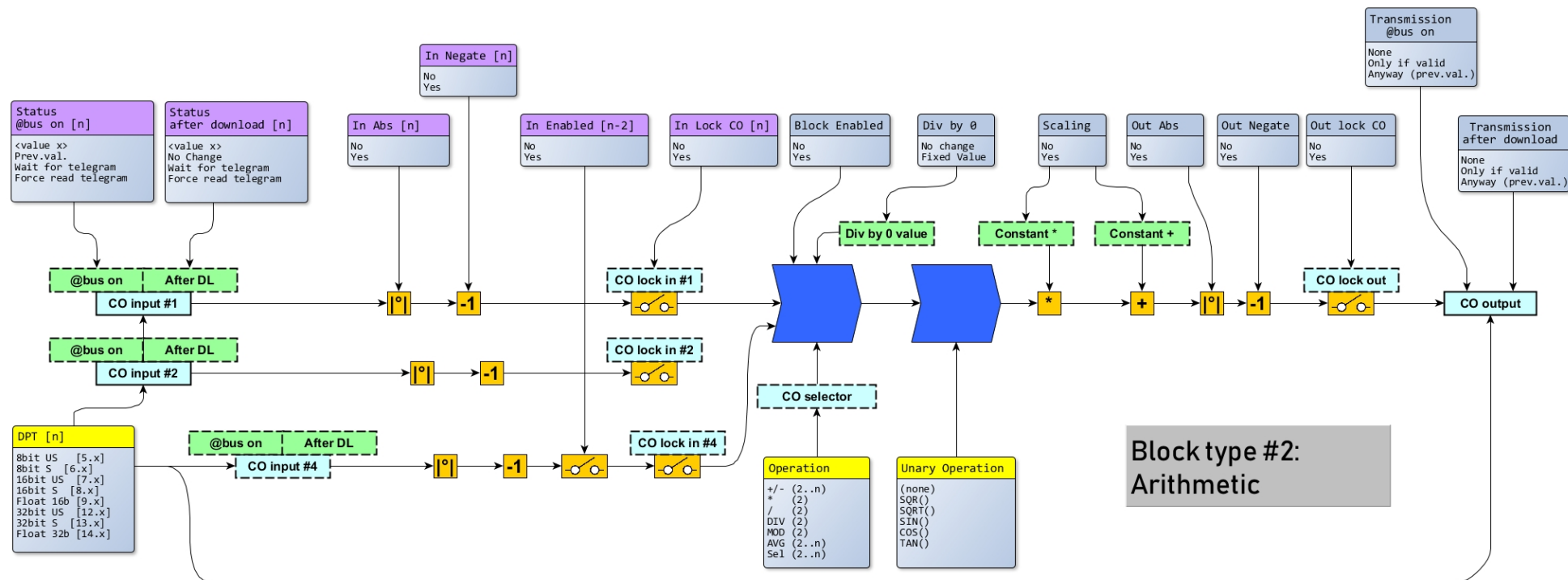
DRAFT

9.1 Block type: Logic

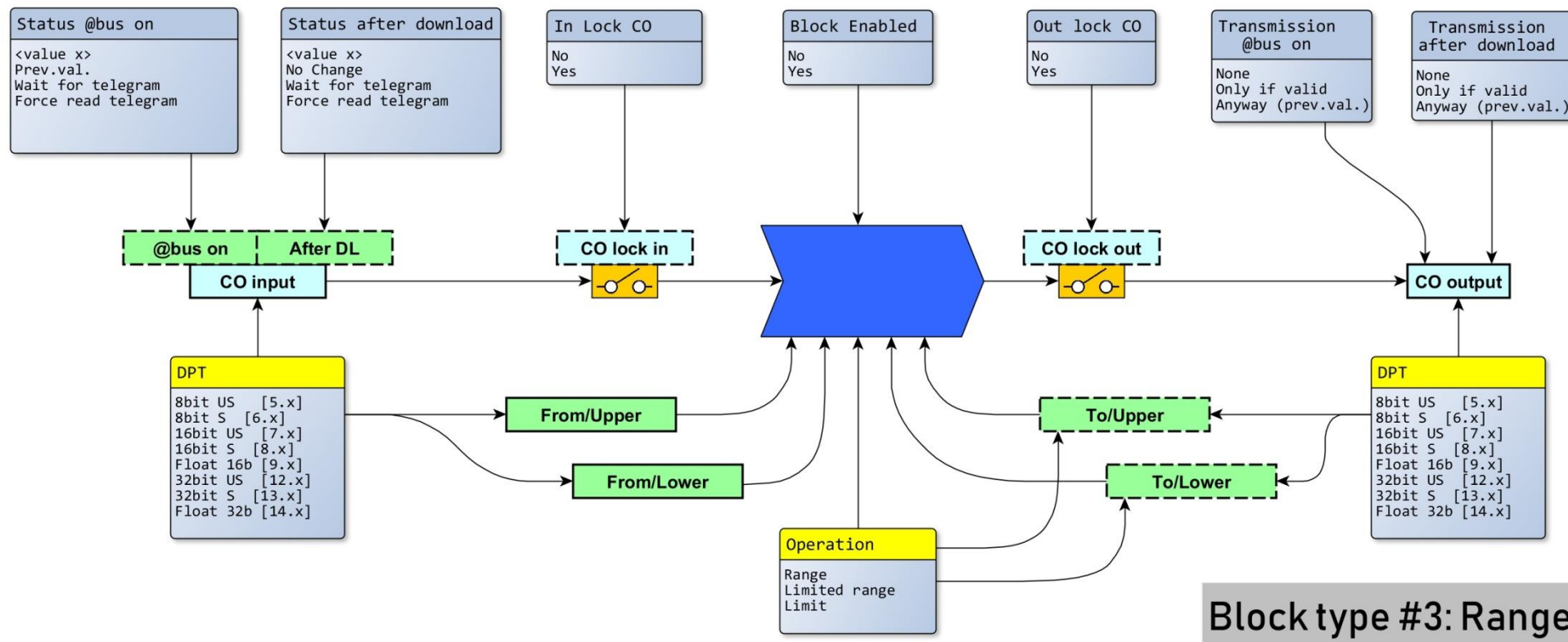


Block type #1: Logic

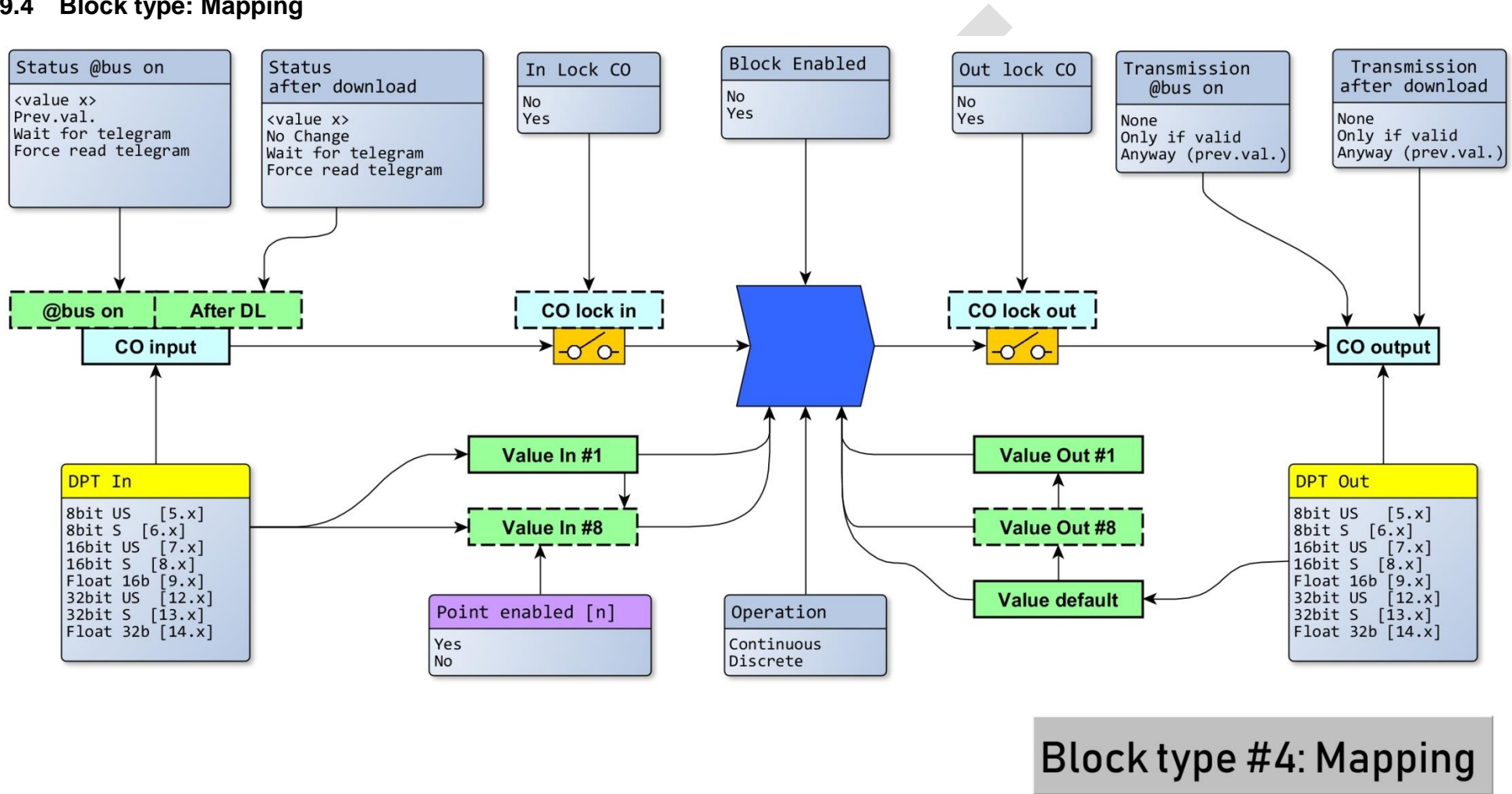
9.2 Block type: Arithmetic / Math



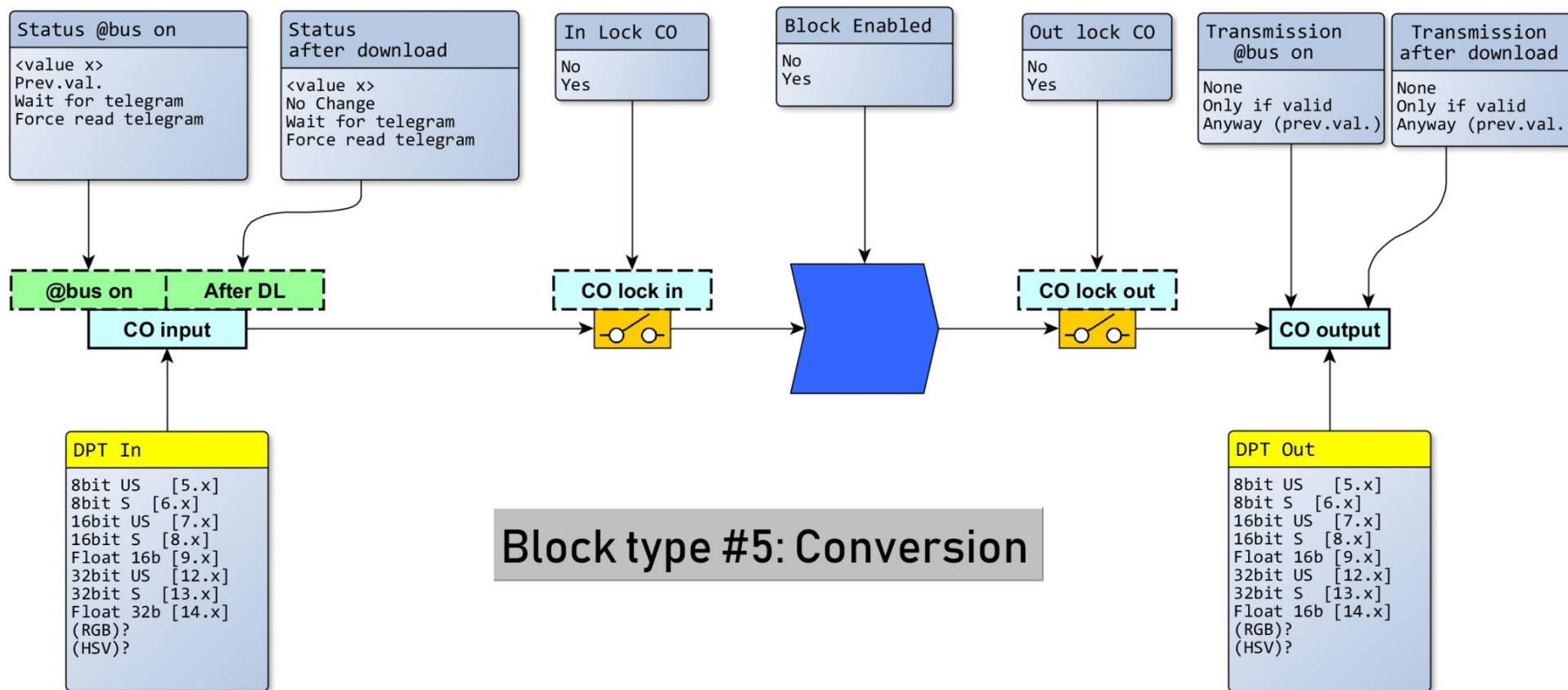
9.3 Block type: Range



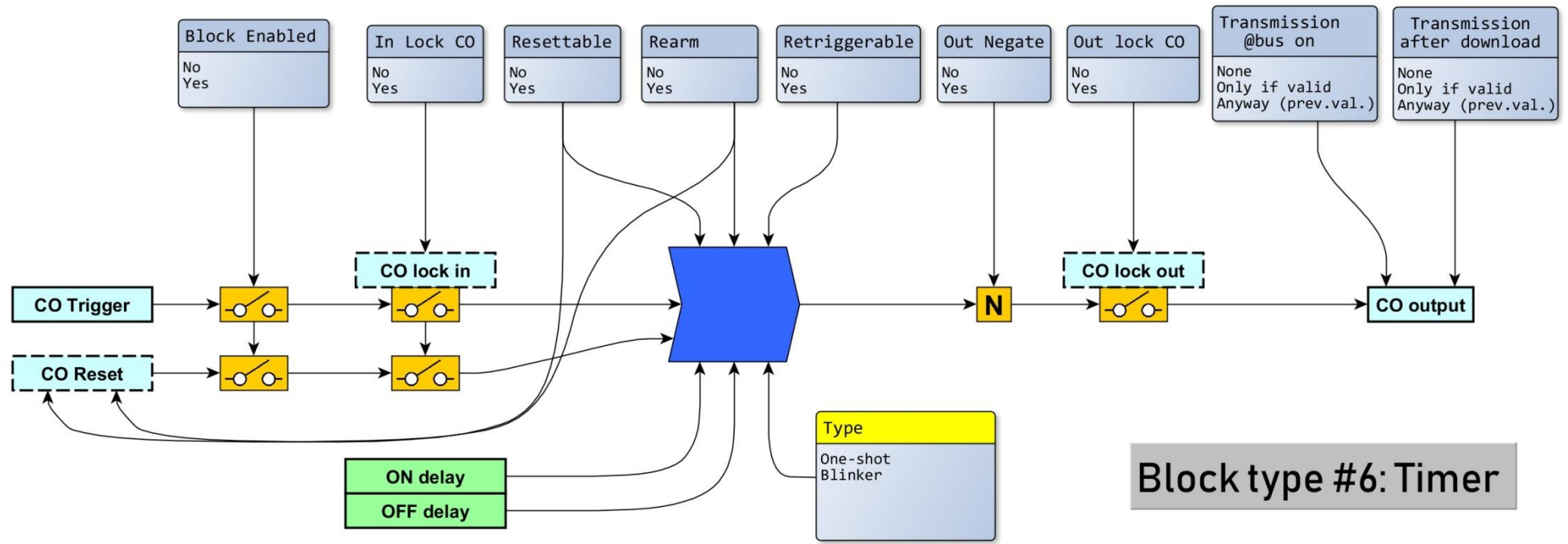
9.4 Block type: Mapping



9.5 Block type: Conversion

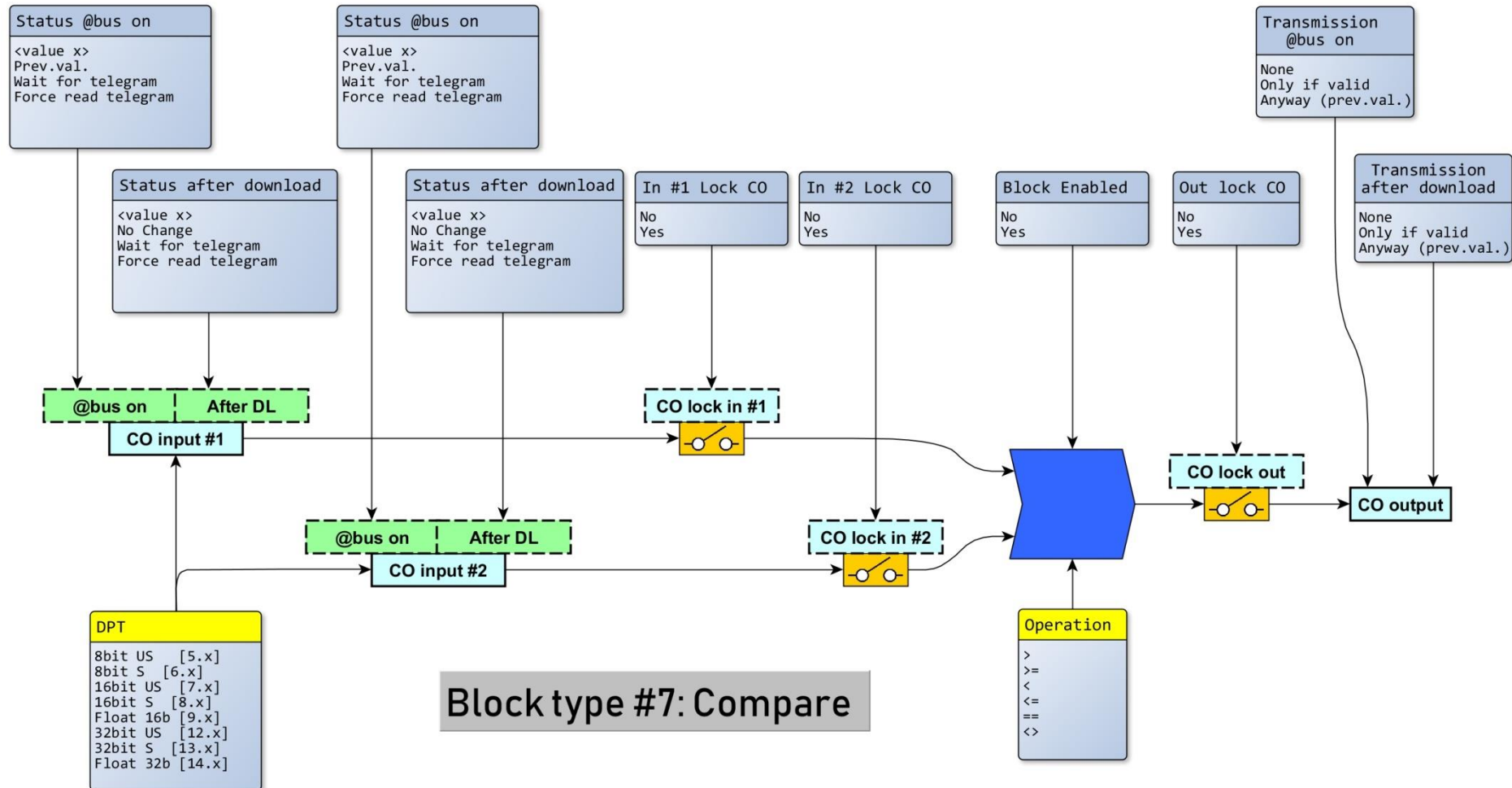


9.6 Block type: Timer

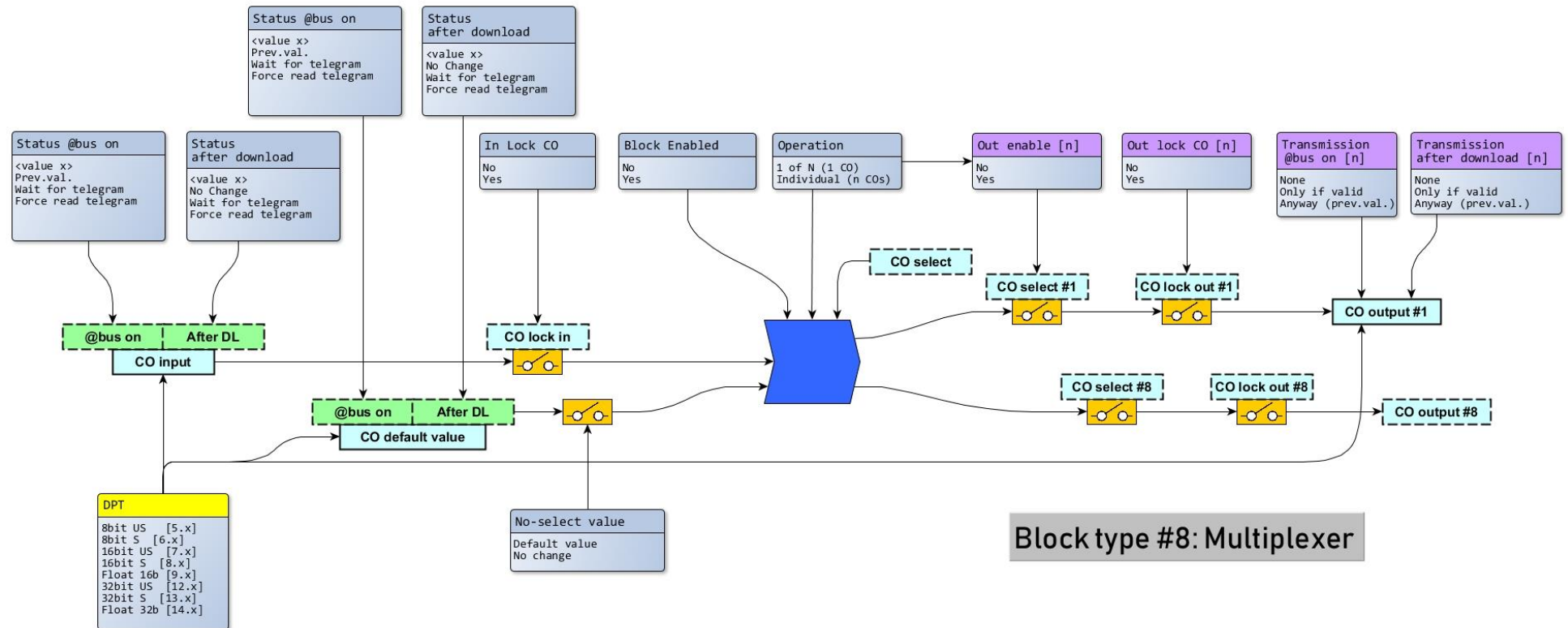


Block type #6: Timer

9.7 Block type: Compare



9.8 Block type: Multiplexer



10. Appendix

10.1 Warnings

- Installation, electrical connection, configuration and commissioning of the device can only be carried out by qualified personnel
- Opening the housing of the device causes the immediate end of the warranty period

10.2 Return of defective products

Defective ekinex® KNX devices can be returned for repair / replacement following the procedure detailed below.

10.2.1 Devices purchased directly from ekinex®

Request an RMA number by sending an E-Mail to the address support@ekinex.com with following mandatory information:

- Exact device model
- Device serial number (can be found on the product label)
- Date of purchase / Order reference
- Detailed description of the fault or issue

The technical assistance team will contact you as quickly as possible to either investigate the problem further, suggest possible solutions or authorize the return of the device for replacement or repair.

If the device should be returned, it should be shipped to the following address:

Ekinex - Via Novara, 37 / SP229 - I-28010 Vaprio d'Agogna (NO) - Italy.

making sure that the assigned RMA number is listed both on the package and on the accompanying documentation.

Further arrangements will be made with the technical support team, according to the type of issue and device.

10.2.2 Devices purchased through ekinex® resellers

If the device has been purchased through a reseller, please refer to the reseller's technical support contact. Depending on the issue and other factors, at the decision of ekinex® and after agreement with the reseller, the customer might be instructed to contact ekinex® directly according to the procedure above.

10.3 Other Information

This application manual is aimed at installers, system integrators and planners

For further information on the product, please contact the ekinex® technical support at the e-mail address: support@ekinex.com or visit the website www.ekinex.com

KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels

© Ekinex S.p.A. 2018. The company reserves the right to make changes to this documentation without notice.