UNI-PRO

DEVELOPMENT ENVIRONMENT FOR PROGRAMMABLE CONTROLLERS

UNI-PRO MANUAL FOR APPLICATION BLOCKS LIBRARIES FOR EEV

CODE 114UPROEBE13

Important

Read this document thoroughly before use of the device and follow all recommendations; keep this document with the device for future consultation.

1. INTRODUCTION

The blocks are almost all independent of the unit of measure type used.
For proper operation, the parameters that set the temperature must be coherent with each other and the pressure parameters must be coherent with each other.
For example, if you wish to work in °C, all of the temperatures must be expressed in °C or K
If, on the other hand, you wish to work in °F, all temperatures will be expressed in °F or R.

The same applies to the pressure.

The only blocks that you need to pay attention to are those for the conversion between pressure and temperature or evaporation and vice versa. In these cases, the pressure is expressed in Barg with precision 2. The temperatures are in °C with precision 1.

Keep in mind that the AIs give the temperature reading in °C with one significant digit after the decimal or voltage or current with two significant digits after the decimal.

All of the percentage positions have two digits after the decimal.

Internally, only the limits (possibly saturated) of the critical inputs are verified (divided by 0, positions above 100.00, value not allowed in configurations)

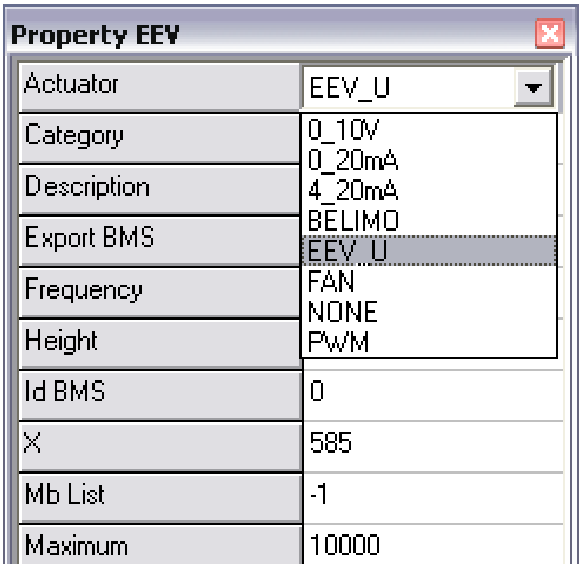
All of the parameters are free. It is the user's responsibility to choose the appropriate limits for the application.

The following tables describe the inputs and outputs on the blocks.

- The min and max columns have the following notations:
    - When the value is between curly brackets {x} it is the recommended minimum/maximum. Its value is not tested and its range is at the user's discretion
    - When a value is not in parentheses, it is the min/max value at which saturation occurs.
    - If nothing is written, the limit is assumed to be the value type (e.g., max CJ_BYTE = 255)
- The opening percentages are in % with precision 2
- The temperatures are in °C with one point of precision.
- The pressure is in Barg with precision 2.

2. AO

The stepper motor is controlled directly using the AO entity configured with the EEV_U type actuator:



The entity controls the positioning, resynchronization and movement method of the stepper motor.

There are two types of movement:

- Forced, which moves the motor in continuous mode until it arrives at the position set-point
- To keep the card from overheating, the motor is driven using a Duty Cycle that alternates pauses and movement.

The resynchronization movement is always forced. It uses the OverdriveSteps parameter as a value and a deceleration ramp around the zero position is activated.

The motor is made to move using the StepRate parameters that set the speed and DrivingMode that sets the mode to set the currents.

The entity value field was used differently than the other AOs:

- The 0 bit is for setting the movement type (forced or according to the duty cycle)
- The 1 to 15 bits set the target position in motor steps (maximum value accepted = 4900 steps) or control the resynchronization if set to the special value 4999

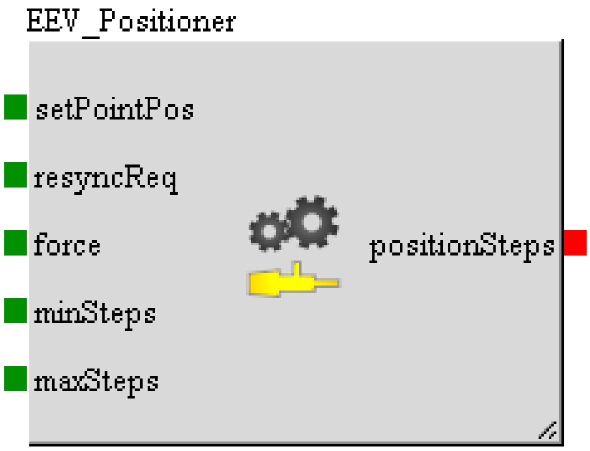
Using only this entity, the stepper motor will move using the parameters set in the controller which can be set using the EEV_ParametrsConfig block for the controllers and the CAN master protocol for the expansions.

The position set by this entity is always activated. Therefore, if you wish to stop the motor, the actual value of the position in steps must be controlled as the set-point position.

Positioning at the next step only occurs if the necessary requirements exist (hold, reverse and ramp out times).:

- If the movement direction changes, you must wait for a reverse time (5-10 ms)
- in order to be able to move the motor if it is in holding (all phases are held at 0V), it must be turned on and you must wait before being able to move (5-10 ms).
- when positioning is complete, before putting the motor in holding (all phases are kept at 0V) you must wait for 5-10 ms.

3. Positioner



This is an essential block for the management of the positioner since it allows the value to be activated to be calculated based on the inputs.

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
setPointPos	CJ_WORD	0	100.00	*	Target position [%]
resyncReq	CJ_BIT	0	1	*	Resynchronization request (uses overdriveSteps), works on the leading edge.
force	CJ_BIT	0	1	*	Movement forcing 0: the movement occurs using the duty cycle 1: the movement is forced (continuous until the end)
minSteps	CJ_ WORD	0	4900	*	Number of steps corresponding to 0.00% opening
maxSteps	CJ_ WORD	minSteps	4900	*	Number of steps corresponding to 100.00% opening

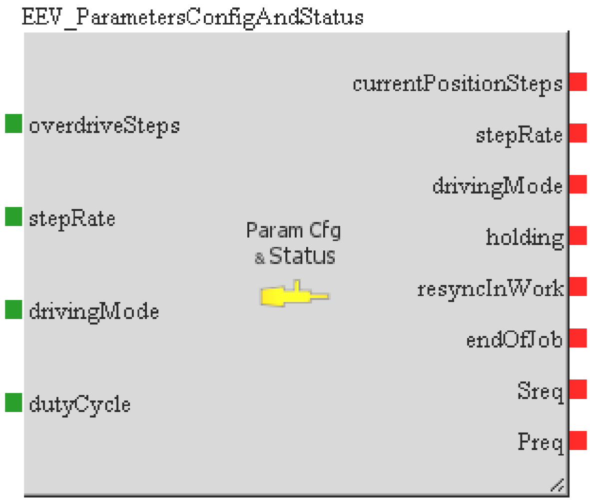
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
positionSteps	CJ_WORD	0	1000 0	The terminal is connected to the EEV_U type AO entity. The value is built as explained in the ***AO*** paragraph.

4. ParametersConfigAndStatus



This is a *single instance* block for writing the parameters and for reading the internal controller states.

If the EEV output is connected to an expansion AO, the configuration of the parameters will be done using the CAN master protocol and the reading of the internal statuses will be done using the CAN master protocol (for drivingMode, stepRate) or dedicated commands 40 and 41 (for currentPositionSteps, holding, resyncInWork, endOfJob, Sreq, Preq) shown in the table below.

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
overdriveSteps	CJ_ WORD	maxSteps	4900	*	Number of steps needed for overdrive. Please Note: writes the related parameter in the controller
stepRate	CJ_WORD	25	1000	*	Step-rate of the stepper motor [step/s] Please Note: writes the related parameter in the controller
drivingMode	CJ_BYTE	0	2	*	Motor piloting method 0: Full step 2 phases on 1: Full step 1 phase on 2: Half step Please Note: writes the related parameter in the controller
dutyCycle	CJ_BIT	10	100	*	Duty cycle for moving the valve: during positioning, the motor moves for a *dutyCycle/10* time in seconds and remains shutdown for (*100-dutyCycle)/10* seconds to prevent the card from overheating. If set at 100, movement is forced. Please Note: writes the related parameter in the controller

* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

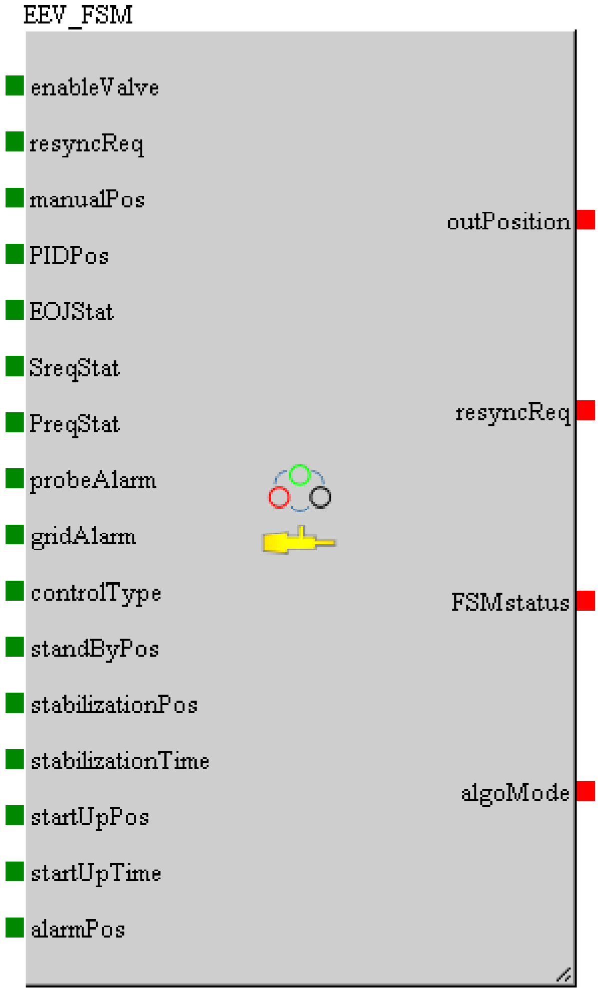
Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
currentPositionSteps	CJ_WORD	0	4900	Actual position of the stepper in FullStep
stepRate	CJ_WORD	1	1000	Step-rate in use by the stepper motor [FullStep/s]
drivingMode	CJ_BYTE	0	2	Piloting method in use by the motor 0: Full step 2 phases on 1: Full step 1 phase on 2: Half step
holding	CJ_BIT	0	1	Motor in holding 0: piloting voltage 1: holding voltage
resyncInWork	CJ_BIT	0	1	Resynchronization in progress 0: resynchronization not activated 1: resynchronization in progress
endOfJob	CJ_BIT	0	1	Positioner operating status 0: positioning in progress 1: positioning ended
Sreq	CJ_BIT	0	1	Resynchronization request scheduled 0: no request 1: Resynchronization request scheduled
Preq	CJ_BIT	0	1	Positioning request scheduled 0: no request 1: positioning request scheduled

The table below shows the expansion commands:

Expansion commands			
Command code	**Command name**	**Variables sent**	**Event**
40	Receive EVCM Current Pos	*Actual stepper motor position in steps (FullStep)*	Every 5 s if unchanged, every 1 s otherwise
41	Receive EVCM Status	*Internal states:* bit 0-10: reserved bit 11: holding bit 12: resyncInWork bit 13: endOfJob bit 14: Sreq	At every variation (100 ms min) or every 5 s if unchanged

		bit 15: Preq	

5. FiniteStateMachine



It is a state machine that calculates the commands for the stepper motor based on external conditions

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
enableValve	CJ_BIT	0	1	*	Valve enabling 0: valve not enabled 1: valve enabled
resyncReq	CJ_BIT	0	1	*	Resynchronization request (active on the leading edge) 0 → 1: activate request
manualPos	CJ_WORD	0	100.00	*	Forced position (use positioner) [%]
PIDPos	CJ_WORD	0	100.00	*	Position calculated by the regulation algorithm
EOJStat	CJ_BIT	0	1	*	Actuator operating state
SreqStat	CJ_BIT	0	1	*	Actuator resynchronization request status
PreqStat	CJ_BIT	0	1	*	Actuator positioning request status
probeAlarm	CJ_BIT	0	1	*	Probe alarm request status (forces to alarmPOS)
gridAlarm	CJ_BIT	0	1	*	A network outage activates the automatic valve closure procedure
controlType	CJ_BYTE	0	2	*	Valve control type 0: positioner (use forcedPos) 1: regulation algorithm (use PIDPos)
standByPos	CJ_WORD	0	100.00	{0}	Resting position [%]
stabilizationPos	CJ_WORD	0	100.00	{100.00}	Position for the stabilization ramp [%]
stabilizationTime	CJ_BYTE	0	255	{0}	Stabilization position permanence [s]
startUpPos	CJ_WORD	0	100.00	{50.00}	Position for the stabilization ramp [%]
startUpTime	CJ_BYTE	0	255	{5}	Start-up position permanence [s]
alarmPos	CJ_WORD	0	100.00	{0.00}	Probe alarm position [%]

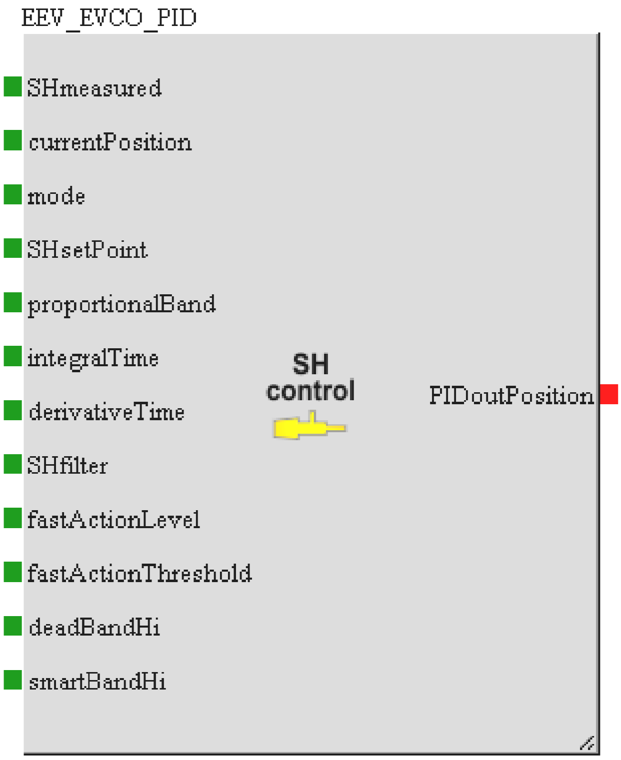
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	Type	Min.	Max.	Description
outPosition	CJ_WORD	0	100.00	Output position [%]
resyncReq	CJ_BIT	0	1	Resynchronization command
FSMStatus	CJ_BYTE	0	60	Machine status 0: initialization 1: resynchronization in progress 2: alarmPos positioning in progress 3: probe alarm status 9: standByPos positioning in progress 10: stand-by status 11: operating mode selection 30: positioner (forcedPos) 40: stabilization ramp initialization 41: start-up ramp initialization 43: awaiting stabilPos positioning 44: awaiting stabilTime 45: awaiting startUpPos positioning 46: awaiting startUpTime 60: regulation algorithm active
algoMode	CJ_BYTE	0	2	Command for PID regulation algorithm 0: algo stopped 1: algo initilization 2: algo run

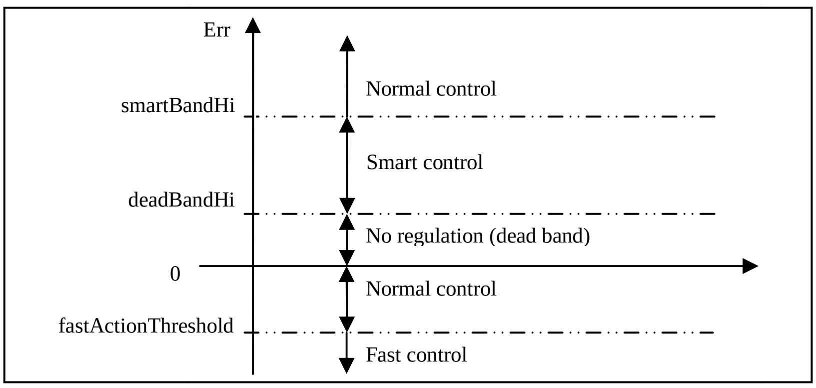
6. SH_Control_PID



It is the regulation algorithm that calculates the motor position in order to reach the desired SH set-point.

The EVCO regulation algorithm activates different regulations based on the operating point:
If the error measured is less than the FAThreshold, a more aggressive regulation algorithm is used in agreement with the level chosen for FALevel.

- If the error measured is greater than zero but less than the deadZoneHi threshold, no regulation is made
- If the error measured is greater than deadZoneHi but less than the propConstBand threshold, a "smart" correction algorithm is used.
- In other cases, a "normal" algorithm is used



The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
SHmeasured	CJ_SHORT			*	Current superheat (SH) value
currentPosition	CJ_WORD	0	100.00	*	Current position [%]
mode	CJ_BYTE	0	2	*	Command for PID regulation algorithm 0: algo stopped 1: algo initilization 2: algo run
SHsetPoint	CJ_WORD	{3.0}	{25.0}	{6.0}	SH [K] set-point
propBand	CJ_WORD	0.1 {1.0}	{100.0}	{40.0}	Proportional band [K]
integralTime	CJ_WORD		{1000}	{120}	Integral time [s] 0: disabled
derivativeTime	CJ_WORD		{1000}	{40}	Derivative time [s] 0: disabled
SHfilter	CJ_BYTE	0	255	{10}	SH filter time constant [100ms]
fastActionLevel	CJ_BYTE	1	100	{100}	Fast action level 100: disabled 1: maximum level
fastActionThreshold	CJ_SHORT	{-10.0}	0	{-1.0}	Fast action zone threshold [°C]
deadBandHi	CJ_WORD	0	{25.0}	{1.0}	Dead band threshold [°C]
smartBandHi	CJ_WORD	deadBandHi	{25.0}	{3.0}	Smart control threshold [°C]

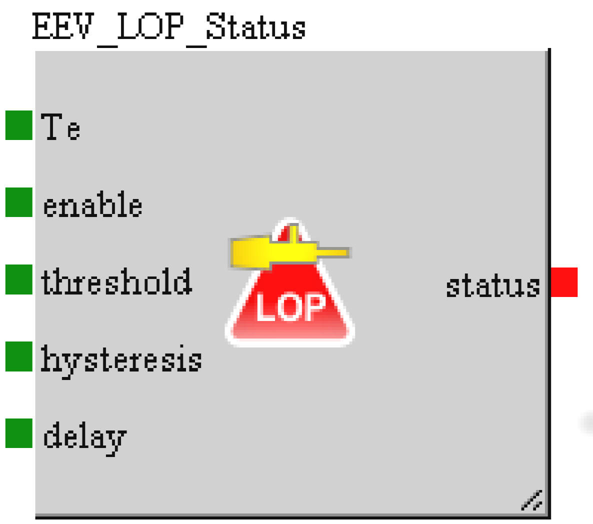
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
PIDOutPosition	CJ_WORD	0	100.00	Calculated Position [%]

7. LOP_Status



Calculates the alarm status due to low evaporation temperature

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
Te	CJ_ANALOG	-	-	*	Measures the evaporation temperature [°C]
enable	CJ_BIT	0	1	*	Enables the status calculation 0: calculation disabled (=> status = OK) 1: calculation enabled
threshold	CJ_SHORT	{-40.0}	{40.0}	{-40.0}	Alarm threshold [°C]
hysteresis	CJ_WORD	{0.0}	{10.0}	{1.0}	Hysteresis [K]
delay	CJ_BYTE	0	255	{3}	Delay [seconds]

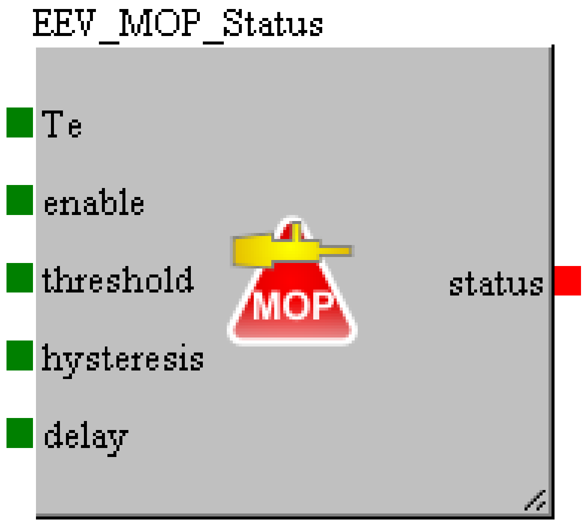
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
status	CJ_BYTE	0	2	Status: 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm

8. MOP_Status



Calculates the alarm status due to high evaporation temperature

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
Te	CJ_ANALOG	-	-	*	Measures the evaporation temperature [°C]
enable	CJ_BIT	0	1	*	Enables the status calculation 0: calculation disabled (=> status = OK) 1: calculation enabled
threshold	CJ_SHORT	{-40.0}	{40.0}	{40.0}	Alarm threshold [°C]
hysteresis	CJ_WORD	{0.0}	{10.0}	{1.0}	Hysteresis [K]
delay	CJ_BYTE	0	255	{3}	Delay [seconds]

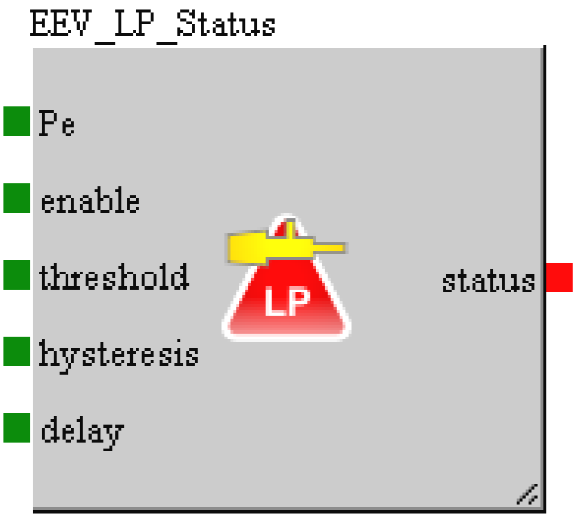
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
status	CJ_BYTE	0	2	Status: 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm

9. LP_Status



Calculates the alarm status due to low evaporation pressure

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
Pe	CJ_ANALOG	-	-	*	Measures the evaporation pressure [Barg]
enable	CJ_BIT	0	1	*	Enables the status calculation 0: calculation disabled (=> status = OK) 1: calculation enabled
threshold	CJ_SHORT	{0.00}	{45.00}	{0.00}	Alarm threshold [Barg]
hysteresis	CJ_WORD	{0.20}	{1.00}	{0.30}	Hysteresis [Bar]
delay	CJ_BYTE	0	255	{3}	Delay [seconds]

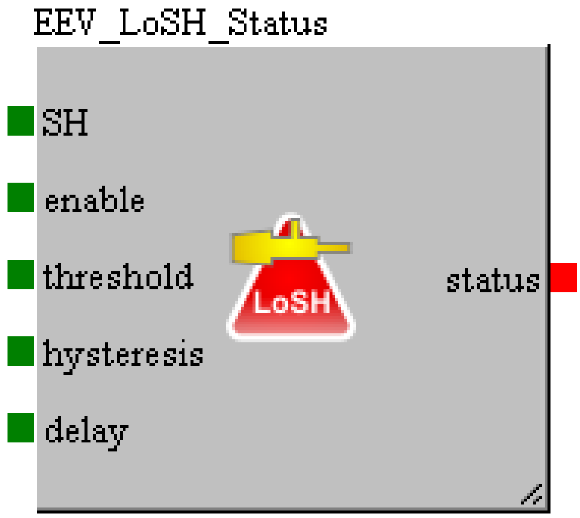
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
status	CJ_BYTE	0	2	Status: 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm

10. LoSH_Status



Calculates the alarm status due to low superheat

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
SH	CJ_ANALOG	-	-	*	SH measurement [K]
enable	CJ_BIT	0	1	*	Enables the status calculation 0: calculation disabled (=> status = OK) 1: calculation enabled
threshold	CJ_SHORT	{1.0}	{3.0}	{2.0}	Alarm threshold [K]
hysteresis	CJ_WORD	{0.0}	{25.0}	{0.5}	Hysteresis [K]
delay	CJ_BYTE	0	255	{3}	Delay [seconds]

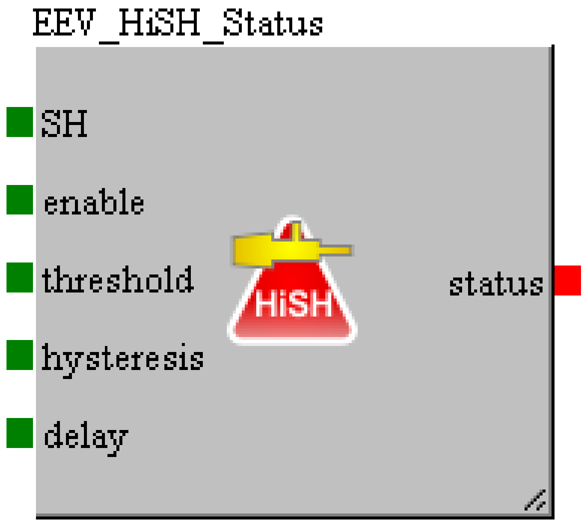
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
status	CJ_BYTE	0	2	Status: 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm

11. HiSH_Status



Calculates the alarm status due to high superheat
The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
SH	CJ_ANALOG	-	-	*	SH measurement [K]
enable	CJ_BIT	0	1	*	Enables the status calculation 0: calculation disabled (=> status = OK) 1: calculation enabled
threshold	CJ_SHORT	{10.0}	{40.0}	{15.0}	Alarm threshold [K]
hysteresis	CJ_WORD	{0.0}	{25.0}	{1.0}	Hysteresis [K]
delay	CJ_BYTE	0	255	{3}	Delay [seconds]

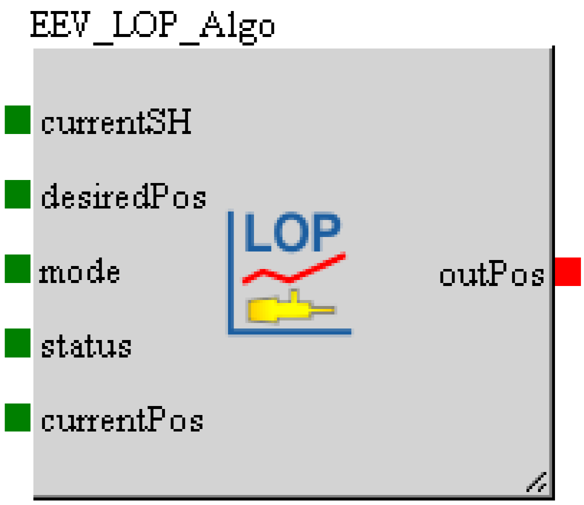
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
status	CJ_BYTE	0	2	Status: 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm

12. LOP_Algo



Calculates the correction in position due to the LOP status.

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
currentSH	CJ_ANALOG	-	-	*	SuperHeat current [K]
desiredPos	CJ_WORD	0	100.00	*	Position calculated by the machine at statuses [%]
mode	CJ_BYTE	0	2	*	Command for the LOP algorithm: 0: no correction 1: correction in start-up ramp 2: correction in algorithm
status	CJ_BYTE	0	2	*	LOP status 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm
currentPos	CJ_WORD	0	100.00	*	Current valve position [%] Please note: if an expansion is used this information must be updated once every 1 s (or every 5 s if the position does not change)

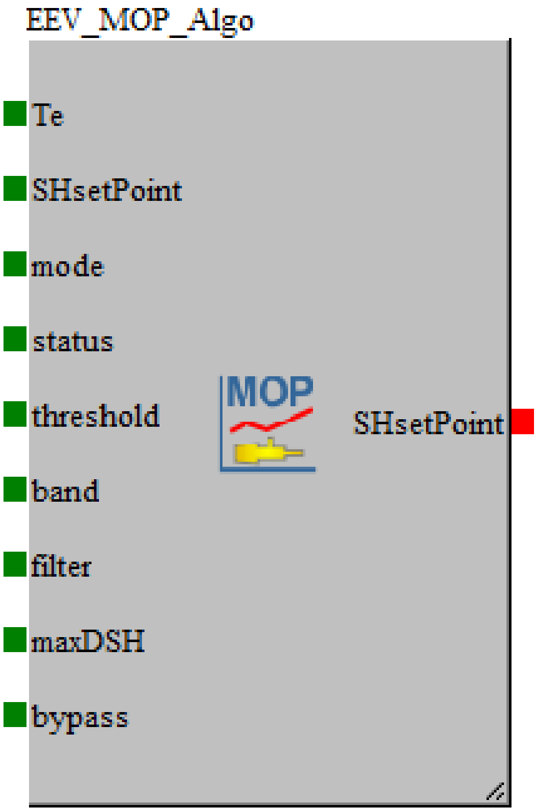
* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
outPos	CJ_WORD	0	100.00	Position with any correction due to LOP [%]

13. MOP_Algo



Calculates the correction, manipulating the superheat setpoint due to the MOP status.

The table below shows the inputs:

Inputs					
Name	**Type**	**Min.**	**Max.**	**Def.**	**Description**
Te	CJ_ANALOG	-	-	*	Measures the evaporation temperature [°C]
SHsetPoint	CJ_SHORT			{6.0}	Set-point of the SH (parameter)
mode	CJ_BYTE	0	2	*	Command for the MOP algorithm: 0: algo stopped 1: algo initilization 2: algo run
status	CJ_BYTE	0	2	*	MOP status 0 : OK (even if the probe is in error or algorithm disabled) 1: Warning 2: Alarm
threshold	CJ_SHORT	{-40.0}	{40.0}	{40.0}	Alarm threshold [°C]
band	CJ_WORD	0.1	{25.0}	{8.0}	Band for the calculation of the MOP correction [K]
filter	CJ_BYTE	1	255	{15}	Time constant for MOP filtering [10 s]
maxDSH	CJ_WORD	{0.0}	{25.0}	{7.0}	Maximum applicable correction [K]
delay	CJ_BYTE	0	255	{10}	Delay time [seconds]

* Inputs to connect

{x} Input to connect, recommended default

The table below shows the outputs

Outputs				
Name	**Type**	**Min.**	**Max.**	**Description**
SHSetPoint	CJ_WORD			SH set-point with any correction due to MOP [K]

UNI-PRO MANUAL FOR APPLICATION BLOCKS LIBRARIES FOR EEV
Version 1.3 January 2019
CODE 114UPROEBE13
File 114UPROEBE13.pdf

Every Control Group

Registered Office
Evco
Via Mezzaterra 6, 32036 Sedico Belluno ITALIA
Tel. 0437-852468
Fax 0437-83648
info@evco.it
www.evco.it

FOREIGN OFFICES
Control France
155 Rue Roger Salengro, 92370 Chaville Paris FRANCE
Tel. 0033-1-41159740
Fax 0033-1-41159739
control.france@wanadoo.fr

Evco Latina
Larrea, 390 San Isidoro, 1609 Buenos Aires ARGENTINA
Tel. 0054-11-47351031
Fax 0054-11-47351031
evcolatina@anykasrl.com.ar

Evco Pacific
59 Premier Drive Campbellfield, 3061, Victoria Melbourne, AUSTRALIA
Tel. 0061-3-9357-0788
Fax 0061-3-9357-7638
everycontrol@pacific.com.au

Evco Russia
111141 Russia Moscow 2-oy Proezd Perova Polya 9
Tel. 007-495-3055884
Fax 007-495-3055884
info@evco.ru

Every Control do Brasil
Rua Marino Félix 256, 02515-030 Casa Verde São Paulo SÃO PAULO BRAZIL
Tel. 0055-11-38588732
Fax 0055-11-39659890
info@everycontrol.com.br

Every Control Norden
Cementvägen 8, 136 50 Haninge SWEDEN
Tel. 0046-8-940470
Fax 0046-8-6053148
mail2@unilec.se

Every Control Shangai
B 302, Yinhai Building, 250 Cao Xi Road, 200235 Shangai CHINA
Tel. 0086-21-64824650
Fax 0086-21-64824649
evcosh@online.sh.cn

Every Control United Kingdom
Unit 19, Monument Business Park, OX44 7RW Chalgrowe, Oxford, UNITED KINGDOM
Tel. 0044-1865-400514
Fax 0044-1865-400419
info@everycontrol.co.uk