

<b>BMXHA7E_15.lib</b>	<b>Shortinfo</b>
HA7E_DebugOut_String	<i>Send debug messages via UDP</i>
HA7E_Init	<i>include HA7E and prepare for eventual debugging</i>
HA7E_Sensor	<i>includes a sensor</i>
<b>CHGLibrary_FritzBox.lib</b>	<b>Shortinfo</b>
FritzBoxCallMonitor	<i>FritzBoxCallMonitor-Makro - please read macro description.</i>
<b>EnerterxCodingContest.lib</b>	<b>Shortinfo</b>
Alarmer	<i>Alarmer-Makro - please read makro description.</i>
AlarmMelder	<i>Alarm system monitoring contacts, movements and sabotage inputs</i>
FlashSetAndSaveF16	<i>Float16 remanent speichern</i>
FlashSetAndSaveF32	<i>Float32 remanent speichern</i>
FlashSetAndSaveS16	<i>Signed16 remanent speichern</i>
FlashSetAndSaveS32	<i>Signed32 remanent speichern</i>
FlashSetAndSaveU32	<i>Float32 remanent speichern</i>
Irrigation_over_MoistureSensor	<i>Control irrigation with moisture sensor in the ground and the EibPC</i>
Irrigation_over_WeatherStation	<i>Control irrigation with weather station and the EibPC</i>
PB_Triggered_ArmingReq	<i>Alarm system arming request generation via push-button triggers</i>
PrepareFlashF32	<i>Float32 remanent speichern</i>
PrepareFlashS16	<i>Signed16 remanent speichern</i>
PrepareFlashS32	<i>Signed32 remanent speichern</i>
PrepareFlashSystem	<i>PrepareFlashF16</i>
PrepareFlashU32	<i>Unsigned32 remanent speichern</i>
SendSMSViaClickatell	<i>SendSMSViaClickatell</i>
ShutterVentilationLogic	<i>shutter ventilation logic</i>
UnixTime	<i>Aktuelle Werte für Zeit und Datum ermitteln und als UnixTime zurückgeben</i>
UnixTimeToVar	<i>ToUnixTime</i>
<b>EnerterxCodingContest.lib</b>	<b>Shortinfo</b>
CommandFusion	<i>Connecting the EibPC with Command Fusion.</i>
Join2Command	<i>Join to execute commands with short and long buttonpush.</i>
Join2DimmerAlternate	<i>Join to execute commands with short and long button press, to operate a dimmer gradually.</i>
Join2Rollo	<i>Join to execute commands with short and long button press. There are two GAs and two sepearate commands are given.</i>
Join2Var	<i>Join "pushbutton" for any variable, e.g. 1-Bit variable: "push ON release OFF" with special Statusobjekt</i>
JoinCommand	<i>Join to execute commands.</i>
JoinDimmer	<i>Join for a dimmer</i>
JoinMinMax2GA	<i>Join for a analog value with Min-Max Declaration - with StatusGA</i>
JoinMinMax	<i>Join for a analog value with Min-Max Declaration</i>
JoinPushOff	<i>Join "pushbutton" for a 1-Bit GA, "push OFF"</i>
JoinPushOn	<i>Join "pushbutton" for a 1-Bit GA, "push ON"</i>
JoinStatus	<i>Join for the display of a GA or variable</i>
JoinToggle2GA	<i>Join "toggle" for a 1 Bit GA, with special status object</i>
JoinToggle2Var	<i>Join "toggle" for a 1 Bit variable</i>
JoinToggleGA	<i>Join "toggle" for a 1-Bit GA</i>
JoinToggleVar	<i>Join "toggle" for a 1-Bit variable</i>
JoinVar	<i>Join "pushbutton" for any variable, e.g. 1-bit variable "push ON release OFF"</i>

<b>EnertexENG.lib</b>	<b>Shortinfo</b>
ClockSynchronisation	<i>Synchronise daily at 3:00am</i>
Dewpoint	<i>Dewpointcalculation</i>
DoubleClick	<i>One Key with DoubleClick</i>
DoubleClickUM	<i>One Key with DoubleClick (toggled)</i>
FloatToPercent	<i>Converts a float value to percent</i>
ForLoop	<i>For-loop</i>
GoogleWeather	<i>Query the Google weather forecast.</i>
kWh	<i>simple electricity counter from mA actor signal</i>
kWh_Simple	<i>simple electricity counter from mA actor signal, voltage 230V and Cosinus-PHI = 1</i>
Longkeypress	<i>Evaluating short and long keypress</i>
LongkeypressGA	<i>Evaluating short and long keypress</i>
Online	<i>Displays how long the EibPC is online.</i>
PercentToFloat	<i>Converts a percent value to a float</i>
PIHeatRuler	<i>PIruler for Heating with percent value</i>
PIHeatRulerGA	<i>PIruler for Heating with percent valueand cycle output to the group address</i>
PIRoomControllerGA	<i>Heating controller with flash memory</i>
ReadFlag	<i>set readflag at EibPC</i>
round	<i>a float number is rounded off or an</i>
TranslateSettimeENG	<i>Translates settime() into an english string, it's saved in the variable you declare</i>
Valve	<i>pulse-width modulation of a valve controlling through RTR</i>
WhileLoop	<i>While-Loop</i>
WOL	<i>Wake on Lan</i>
<b>EnertexFlashENG.lib</b>	<b>Shortinfo</b>
FlashCycleGA	<i>Use of flash memory for data recording remanente</i>
VisuFlash	<i>Visualization of data from recording data remantenter</i>
VisuFlashHistory	<i>Visualization of stored data from the flash at remantenter data recording.</i>
<b>EnertexLight.lib</b>	<b>Shortinfo</b>
ComfortDimmer	<i>Dimmer, switchable through motion sensor or switcher. The value of the dimmer can be set up differently for day and night.</i>
DawnSwitcher	<i>Dawn switcher, which switches on a lightactor at darkness and off at brightness</i>
DawnSwitcherTimer	<i>Dawn switcher which controls a Lightactor depending on the time and brightness</i>
MotionSensorlockable	<i>Motion sensor, manually bridgeable.</i>
StairsLight	<i>Stairslight with ON-pushbutton</i>
Wakeuplight	<i>To dim the dimmer to a certain value in a certain time at a certain time</i>
<b>EnertexLogic.lib</b>	<b>Shortinfo</b>
AND2	<i>AND-link with 2 inputs</i>
AND3	<i>AND-link with 3 inputs</i>
AND4	<i>AND-link with 4 inputs</i>
AND8	<i>AND-link with 8 inputs</i>
NOT_AND2	<i>NOT_AND-link with 2 inputs(inversion of AND)</i>
NOT_AND3	<i>NOT_AND-link with 3 inputs(inversion of AND)</i>
NOT_AND4	<i>NOT_AND-link with 4 inputs(inversion of AND)</i>
NOT_AND8	<i>NOT_AND-link with 8 inputs(inversion of AND)</i>
NOT_OR2	<i>NOT_OR-link with 2 inputs(inversion of OR)</i>
NOT_OR3	<i>NOT_OR-link with 3 inputs(inversion of OR)</i>
NOT_OR4	<i>NOT_OR-link with 4 inputs(inversion of OR)</i>
NOT_OR8	<i>NOT_OR-link with 8 inputs(inversion of OR)</i>
OR2	<i>OR-link with 2 inputs</i>
OR3	<i>OR-link with 3 inputs</i>
OR4	<i>OR-link with 4 inputs</i>
OR8	<i>OR-link with 8 inputs</i>

<b>EnertexOneWireENG.lib</b>	<b>Shortinfo</b>
HA7E	<i>Embedding the HA7E adapter</i>
Initialise	<i>Initialising the HA7E adapter</i>
OWextender	<i>Sensor query with OW-Extender</i>
OWextenderGA	<i>Sensor query with OW-Extender</i>
Temperature	<i>Reading periodically the temperature from a sensor</i>
TemperatureGA	<i>Reading periodically the temperature from a sensor and sending it to a group address</i>
TemperatureGASync	<i>Reads cyclically the temperature from a sensor and send it to a group address.</i>
TemperatureSync	<i>Reads cyclically the temperature from a sensor.</i>
<b>EnertexPhyMonitorENG.lib</b>	<b>Shortinfo</b>
PhyGAMonitor	<i>\$device group address monitor</i>
PhyGAMonitorTime	<i>\$Actor watchdog</i>
PhyMonitor	<i>\$device monitor</i>
<b>EnertexPresence.lib</b>	<b>Shortinfo</b>
InternalScheduler	<i>only for internal use</i>
InternalSchedulerD	<i>only for internal use</i>
InternalSchedulerH	<i>only for internal use</i>
InternalSchedulerW	<i>only for internal use</i>
Rec_GA	<i>record group address in simulation</i>
Scheduler_2Weeks	<i>presence simulation - basismacro</i>
<b>EnertexRusssoundENG.lib</b>	<b>Shortinfo</b>
GetBalance	<i>Get current balance and write it to variable Name^GetBalance</i>
GetBass	<i>Get current bass and write it to variable Name^GetBass</i>
getChecksumForGet	<i>Internal macro to compute a checksum.</i>
getChecksumForGetExtended	<i>Internal macro to compute a checksum.</i>
getChecksumForSet	<i>Internal macro to compute a checksum.</i>
getChecksumForSetExtended	<i>Internal macro to compute a checksum.</i>
GetLoudness	<i>Get current loudness and write it to variable Name^GetLoudness</i>
GetTreble	<i>Get current treble and write it to variable Name^GetTreble</i>
GetTurnOnVolume	<i>Get current turn on volume and write it to variable Name^TurnOnVolume</i>
GetVolume	<i>Get current volume and write it to variable Name^GetVolume</i>
InitRusssound	<i>Integrate Russsound via Moxa</i>
ReadAnswer	<i>internal macro to read UDP telegrams.</i>
ReadData	<i>Macro for debug</i>
SendCommand	<i>Internal macro to send commands</i>
SendQuery	<i>Internal macro to send queries.</i>
SetBalanceRusssound	<i>Change balance step by step</i>
SetBassRusssound	<i>Change bass step by step</i>
SetLoudnessRusssoundToggle	<i>Turns loudness on and off (toggle)</i>
SetStateRusssoundToggle	<i>Set the state of a zone for a particular controller</i>
SetTrebleRusssound	<i>Change treble step by step</i>
SetTurnOnVolumeRusssound	<i>Change turn on volume step by step</i>
SetVolumeRusssound	<i>Change volume step by step</i>

<b>EnergexSceneENG.lib</b>	<b>Shortinfo</b>
Scene3Module	<i>Define a scene module with 3 group addresses or variables.</i>
Scene3PresetNr	<i>Predefine a scene number of a scene module</i>
Scene5Module	<i>Define a scene module with 5 group addresses or variables.</i>
Scene5PresetNr	<i>Predefine a scene number of a scene module</i>
Scene10Module	<i>Define a scene module with 10 group addresses or variables.</i>
Scene10PresetNr	<i>Predefine a scene number of a scene module</i>
Scene20Module	<i>Define a scene module with 20 group addresses or variables.</i>
Scene20PresetNr	<i>Predefine a scene number of a scene module</i>
<b>EnergexShadowing.lib</b>	<b>Shortinfo</b>
ShadowingBladeBlindEast	<i>Blind and slat shade of an east window. Blade parameterization: 0%: slats completely closed.</i>
ShadowingBladeBlindEastInverse	<i>Blind and slat shade of an east window. Blade parameterization: 100%: slats completely closed.</i>
ShadowingBladeBlindSouth	<i>Blind and slat shade of an south window. Blade parameterization: 0%: slats completely closed.</i>
ShadowingBladeBlindSouthInverse	<i>Blind and slat shade of an south window. Blade parameterization: 100%: slats completely closed.</i>
ShadowingBladeBlindSouthWest	<i>Blind and slat shade of an south-west window. Blade parameterization: 0%: slats completely closed.</i>
ShadowingBladeBlindSouthWestInverse	<i>Blind and slat shade of an south-west window. Blade parameterization: 100%: slats completely closed.</i>
ShadowingBladeBlindWest	<i>Blind and slat shade of an west window. Blade parameterization: 0%: slats completely closed.</i>
ShadowingBladeBlindWestInverse	<i>Blind and slat shade of an west window. Blade parameterization: 100%: slats completely closed.</i>
ShadowingBlind	<i>Blind-shadowing of a window. The orientation in degree has to be known.</i>
ShadowingBlinddegreeTime	<i>Blind-shadowing of a window with after run time. The orientation in degree has to be known.</i>
ShadowingBlindEast	<i>Blind-shadowing of an East-window. The actor has a group address saving the stored shadowing level.</i>
ShadowingBlindEastTime	<i>Blind-shadowing of a East-window with adjustable run-up time for light incidence at shadowing.</i>
ShadowingBlindSouth	<i>Blind-shadowing of a south-window. The actor has a group address saving the stored shadowing level.</i>
ShadowingBlindSouthEast	<i>Blind-shadowing of a south-East-window. The actor has a group address saving the stored shadowing level.</i>
ShadowingBlindSouthEastTime	<i>Blind-shadowing of a south-East-window with adjustable run-up time for light incidence at shadowing.</i>
ShadowingBlindSouthTime	<i>Blind-shadowing of a south-window with adjustable run-up time for light incidence at shadowing.</i>
ShadowingBlindSouthWest	<i>Blind-shadowing of a south-West-window. The actor has a group address saving the stored shadowing level.</i>
ShadowingBlindSouthWestTime	<i>Blind-shadowing of a south-West-window with adjustable run-up time for light incidence at shadowing.</i>
ShadowingBlindWest	<i>Blind-shadowing of a West-window. The actor has a group address saving the stored shadowing level.</i>
ShadowingBlindWestTime	<i>Blind-shadowing of a West-window with adjustable run-up time for light incidence at shadowing.</i>
ShadowingRoofSlat	<i>slats-shadowing of a roofwindow at declaration of the shadowing angle.</i>
ShadowingRoofSlatEast	<i>slats-shadowing of a East-roof-window.</i>
ShadowingRoofSlatSouth	<i>slats-shadowing of a south-roof-window.</i>
ShadowingRoofSlatSouthEast	<i>slats-shadowing of a east-roof-window.</i>
ShadowingRoofSlatSouthWest	<i>slats-shadowing of a south-West-roof-window.</i>
ShadowingRoofSlatWest	<i>slats-shadowing of a West-roof-window.</i>
ShadowingSlat	<i>slats-shadowing of a window at declaration of the shadowing angle</i>
ShadowingSlatEast	<i>slats-shadowing of an East-window.</i>
ShadowingSlatEastInverse	<i>slats-shadowing of an East-window. Slat parameterization: 100%: slates fully closed.</i>
ShadowingSlatInverse	<i>slats-shadowing of a window at declaration of the shadowing angle</i>
ShadowingSlatSouth	<i>slats-shadowing of a south-window.</i>
ShadowingSlatSouthEast	<i>slats-shadowing of a south-East-window.</i>
ShadowingSlatSouthEastInverse	<i>slats-shadowing of a south-East-window. Slat parameterization: 100%: slates fully closed.</i>
ShadowingSlatSouthInverse	<i>slats-shadowing of a south-window. Slat parameterization: 100%: slates fully closed.</i>
ShadowingSlatSouthWest	<i>slats-shadowing of a south-west-window.</i>
ShadowingSlatSouthWestInverse	<i>slats-shadowing of a south-West-window. Slat parameterization: 100%: slates fully closed.</i>
ShadowingSlatWest	<i>slats-shadowing of a West-window.</i>
ShadowingSlatWestInverse	<i>slats-shadowing of a West-window. Slat parameterization: 100%: slates fully closed.</i>

<b>EnertexSonosBetaENG.lib</b>	<b>Shortinfo</b>
GetBalanceSonos	<i>Get current balance</i>
GetBassSonos	<i>Get current bass</i>
GetTrebleSonos	<i>Get current treble</i>
GetVolumeSonos	<i>Get current volume</i>
SetBalanceSonos	<i>Change balance step by step.</i>
SetBassSonos	<i>Change bass step by step.</i>
SetChangeSongSonos	<i>Switch to next or previous song</i>
SetMuteSonosToggle	<i>Change between mute ON and OFF</i>
SetPlaylistSonos	<i>Play the specified playlist station.</i>
SetRadioSonos	<i>Play the specified radio station.</i>
SetStateSonosToggle	<i>Change between PLAY and STOP</i>
SetTrebleSonos	<i>Change treble step by step</i>
SetVolumeSonos	<i>Change volume step by step</i>
Sonos	<i>Sonos initialisation</i>
<b>EnertexSqueezeboxENG.lib</b>	<b>Shortinfo</b>
CommandGA	<i>Sending an arbitrary command through an actor to the squeezebox</i>
CommandVAR	<i>Sending an arbitrary command through a variable to the squeezebox</i>
DisplayStringGA	<i>Sends a string to the display through an actor</i>
DisplayStringVar	<i>Sends a string through a variable to the display</i>
DisplayValueCycle	<i>Sends a value periodically to the display</i>
DisplayValueGA	<i>Sends a value through an actor to the display.</i>
DisplayValueVar	<i>Sends a value through a variable to the display.</i>
ForwardGA	<i>"Forward" is sent to the squeezebox through an actor</i>
ForwardVAR	<i>"Forward" is sent to the squeezebox through a variable</i>
IndexGA	<i>Through an actor a given song will be played</i>
IndexVAR	<i>Through a variable a given song will be played</i>
PauseGA	<i>"Pause" is sent to the squeezebox through an actor</i>
PauseVAR	<i>"Pause" is sent to the squeezebox through a variable</i>
PlayGA	<i>"play" is sent to the squeezebox through an actor</i>
PlayVAR	<i>"Play" is sent to the squeezebox through a variable</i>
PowerGA	<i>The squeezebox is switched on and off through an actor</i>
PowerSwitch	<i>Switching the squeezebox with switch</i>
PowerVAR	<i>The squeezebox is switched on and off through a variable</i>
RewindGA	<i>"Rewind" is sent to the squeezebox through an actor</i>
RewindVAR	<i>"Rewind" is sent to the squeezebox through a variable</i>
Ruler	<i>You control the volume of the squeezebox through a ruler.</i>
SignalGA	<i>A signal is played through an actor</i>
SignalVar	<i>Die Squeezebox spielt über eine Variable ein Signal ab</i>
Squeezebox	<i>Initialising the squeezebox</i>

<b>EnertexTimeswitchesV2.lib</b>	<b>Shortinfo</b>
At_Sunrise	<i>Switching the actor at sunrise</i>
At_Sunrise_Capped	<i>Switching the actor at sunrise or the latest at a determined time</i>
At_Sunrise_Capped_withRelease	<i>Switching the actor at sunrise or the latest at a determined time</i>
At_Sunset	<i>Switching the actor at sunset</i>
At_Sunset_Capped	<i>Switching the actor at sunset or the latest at a determined time</i>
At_Sunset_Capped_withRelease	<i>Switching the actor at sunset or the latest at a determined time</i>
DayTimeswitch	<i>Switching a group address every day at a determined time.</i>
DayTimeswitch_2GA	<i>Switching two group addresses every day at a determined time</i>
SunriseEarliest	<i>Switching an actor at sunrise but not before a given time.</i>
TimeswitchMidnight	<i>Timeswitch with which you can switch an actor on and off, as well over midnight</i>
WeekTimeswitch	<i>Switching a group address every week at a determined time.</i>
WeekTimeswitch_2GA	<i>Switching two group addresses every week at a determined time</i>
<b>EnertexTimingElements.lib</b>	<b>Shortinfo</b>
switch_delay	<i>switching an actor with any value, timer isn't able to restart</i>
switch_delay_with_restart	<i>switching an actor with any value, timer is able to restart</i>
switch_off_delay_1	<i>switch-off delay with one actor, timer isn't able to restart</i>
switch_off_delay_2	<i>switch-off delay with two actors, timer isn't able to restart</i>
switch_off_delay_with_restart_1	<i>switch-off delay with one actor, timer is able to restart</i>
switch_off_delay_with_restart_2	<i>switch-off delay with two actors, timer is able to restart</i>
switch_on_delay_1	<i>switch-on delay with one actor, timer isn't able to restart</i>
switch_on_delay_2	<i>switch-on delay with two actors, timer isn't able to restart</i>
switch_on_delay_with_restart_1	<i>switch-on delay with one actor, timer is able to restart</i>
switch_on_delay_with_restart_2	<i>switch-on delay with two actors, timer is able to restart</i>
<b>EnertexWebENG.lib</b>	<b>Shortinfo</b>
MultiButton2	<i>To implement a mbutton or mshifter with two GA</i>
MultiButton3	<i>To implement a mbutton or mshifter with three GA</i>
MultiButton4	<i>To implement a mbutton or mshifter with four GA</i>
MultipageButton2	<i>To implement a mpbutton or mpshifter with two GA</i>
MultipageButton3	<i>To implement a mpbutton or mpshifter with three GA</i>
MultipageButton4	<i>To implement a mpbutton or mpshifter with four GA</i>
Roomcontroller	<i>Roomcontroller</i>
SliderDimmer	<i>Controlling a dimmer with a Slider</i>
ToggleButton	<i>To implement a toggle</i>
ToggleShifter	<i>To implement a toggle</i>
TwoBlinds	<i>Controlling two blinds</i>
Webtimer	<i>to control a timer through the webserver</i>

<b>EnertexWebENGv2.lib</b>	<b>Shortinfo</b>
Blind	<i>Controlling a blind</i>
BlindsWithSlats	<i>to control blinds with slat via webserver</i>
DateDisplay	<i>date display via webserver</i>
DisplayState	<i>Displays state over WebServer</i>
GetInfoButton	<i>Controlling a page button. With push the button, the value is displayed.</i>
HeatingPChart	<i>XY-Diagram for webserver, scaling optimized for heating values.</i>
Holidaycalendar	<i>Visualizing an Holidaycalendar</i>
HolidayExecute	<i>Realising the Holidaycalendar</i>
LightDisplayButton	<i>Display the current light value.</i>
MinMaxTemperatureDisplayButton	<i>display the min and max temperature of a day</i>
MultiButton2	<i>To implement a mbutton or mshifter with two GA - OBSOLETE next generation: ToggleMultiButton2</i>
MultiButton3	<i>To implement a mbutton or mshifter with three GA - OBSOLETE next generation: ToggleMultiButton3</i>
MultiButton4	<i>To implement a mbutton or mshifter with four GA - OBSOLETE next generation: ToggleMultiButton4</i>
MultipageButton2	<i>To implement a mpbutton or mpshifter with two GA - OBSOLETE next generation: ToggleMultiPageButton2</i>
MultipageButton3	<i>To implement a mpbutton or mpshifter with three GA - OBSOLETE next generation: ToggleMultiPageButton3</i>
MultipageButton4	<i>To implement a mpbutton or mpshifter with four GA - OBSOLETE next generation: ToggleMultiPageButton4</i>
OnlineDisplayButton	<i>Dislay the EibPC online time.</i>
Roomcontroller	<i>Roomcontroller</i>
SliderDimmer	<i>Controlling a dimmer with a Slider</i>
SliderDimmerButton	<i>Controlling a dimmer with a Slider and its toggle button</i>
SliderTemperature	<i>Changing a temperature group address with a slider</i>
SliderTemperatureVar	<i>Changing a temperature group address with a slider</i>
SunriseSunsetRainSignalDisplayButton	<i>Display state for sunreise, sunset and rain.</i>
TemperaturePChart	<i>XY-Diagram for webserver.</i>
ThreeBlindsWithSlats	<i>to control three blinds with slat via webserver</i>
ThreeWindowContacts	<i>Displays state of three window contact via webserver</i>
TimeDisplay	<i>time display via webserver</i>
ToggleButton	<i>To implement a toggle</i>
ToggleButtonState	<i>To implement a toggle with declaring the states of the icon</i>
ToggleButtonVar	<i>To implement a toggle with declaring the states of the icon</i>
ToggleMultiButton2	<i>To implement a mbutton or mshifter with two GA</i>
ToggleMultiButton3	<i>To implement a mbutton or mshifter with three GA</i>
ToggleMultiButton4	<i>To implement a mbutton or mshifter with four GA</i>
ToggleMultipageButton2	<i>To implement a mpbutton or mpshifter with two GA</i>
ToggleMultipageButton3	<i>To implement a mpbutton or mpshifter with three GA</i>
ToggleMultipageButton4	<i>To implement a mpbutton or mpshifter with four GA</i>
ToggleShifter2	<i>Controlling a pshifter with two elements</i>
ToggleShifter	<i>To implement a toggle showing the date</i>
ToggleShifterState	<i>To implement a toggle showing the date and with declaring the states of the icon</i>
TwoBlinds	<i>Controlling two blinds</i>
TwoBlindsWithSlats	<i>to control two blinds with slat via webserver</i>
TwoWindowContacts	<i>Displays state of two window contact via webserver</i>
WebSettingValueGA	<i>Controls a value change via WebServer</i>
WebTimer	<i>to control a timer through the webserver</i>
WebTimerDaySelection	<i>To implement a day selection for a timer</i>
WebTimerExecuting	<i>A webtimer controls a GA</i>
WindDisplayButton	<i>Display the current wind speed.</i>
WindowContact	<i>Displays state of a window contact via webserver</i>
<b>EnertexWigaENG.lib</b>	<b>Shortinfo</b>
awning_outside	<i>to control the awning in the winter garden depending on the outside temperature</i>
awning_outside_and_inside	<i>to control the awning depending on the outside and inside temperature</i>
ventilation_dewpoint	<i>winter garden ventilation depending on the dewpoint</i>