

Documentation for the wind turbines including Lua script

Version 1.0 - Stand: 19.02.2024

This documentation provides a concise introduction to the functions of wind turbines and explains how to integrate the supplied Lua script into EEP.

Table of contents

1. Reality / History	1
1.1 Antiquity and the Middle Ages	1
1.2 Period of industrialization	1
1.3 Modern wind turbines	2
2. The models	3
2.1 Overview	3
2.2 Axes and functions	5
2.2.1 Nacelle alignment	5
2.2.2 Blade position	5
2.2.3 Speed	5
2.2.4 Blade bending	5
2.2.5 System_Blink	5
2.3 Sound modules	6
3. The Lua-Script	7
3.1 Functions	8
3.2 Setting options	8
3.2.1 Wind force	10
3.2.2 Wind direction	11
3.2.3 Energy Requirement	12
3.2.4 List of wind turbines	13
3.2.5 List of Sound modules	15
3.2.7 Extra-Infos	15
3.3 Integration into EEP	16
3.3.1 Inserting the Setting Variables into the In-Layout Lua Script	16
3.3.2 Control by count-signal	19
4. Video-Tutorials	20

1. Reality / History

1.1 Antiquity and the Middle Ages

Since ancient times, people have been trying to use the energy of the wind effectively. For this purpose, so-called "resistance rotors" were initially used, which usually had a vertical axis of rotation and converted the resistance force of the wind into torque. Such ancient wind turbines were mainly used to grind grain or to pump water.

Since the Middle Ages, people in Europe have increasingly built so-called "buoyancy rotors", which have a horizontal rotor axis and follow the same principle as modern wind turbines.

With the onset of industrialization, there was a significant increase in wind power from the 19th century onwards. The Netherlands and the USA are the technical pioneers. Even today, the typical Dutch windmills can still be found in numerous places far beyond the borders of the Netherlands. In the USA, well over 600,000 Westernmills were built, which have legitimate appearances in numerous Wild West films.

1.2 Period of industrialization

Nevertheless, these old mills differ significantly from the wind turbines known today. In the case of Dutch mills, the blade depth is almost constant, and in the case of the Westernmills, the blades even become wider towards the outside. This type of design is called "low-speed". On the other hand, the blade depth of modern wind turbines increases steadily towards the tip of the blade. This version is called "high-speed".

In Germany, the pioneer of modern technology is the two-bladed GROWIAN (**large wind power plant**) in the Kaiser-Wilhelm-Koog not far from the mouth of the Elbe. **This wind turbine was built for testing and was** already as large as today's turbines in 1983. However, due to a lack of experience and the associated incorrect design, it had to struggle with operational problems and was demolished after only five years.

The Danes took the concept of the GROWIAN further and developed three-bladed wind turbines under the "Danish concept", on which almost all of today's wind turbines are based.

1.3 Modern wind turbines

Most of today's wind turbines are designed as windwards, which means that the rotor is in *front* of the tower in the direction of the wind. Thus, energy losses due to the resistance of the tower can be reduced.

Since the 1990s, wind turbines have been erected for commercial energy generation and have been continuously developed. Starting with just a few kilowatts, the output of a single plant has now been increased to over 15 megawatts. This is accompanied by a significant increase in the size of the plant. While the first commercial wind turbines had a rotor diameter of 30 meters, modern developments usually had a wingspan of over 100 meters. At the same time, the hub height must also increase, which makes today's wind turbines visible at ever greater distances.

The rotor blades of modern wind turbines are designed to swivel and are automatically controlled to ensure an optimal flow at any wind speed and thus optimize efficiency. Wind turbines are also developed for different wind classes and turbulence intensities. For example, different rotor blade profiles are required near the coast than inland.

As the size increases, wind turbines must be marked with red paint (during the day) and hazard lights (at night) for air traffic. For the most part, the "fire W, red" is used for this purpose, which is the prescribed flashing sequence

1s on – 0,5s off – 1s on – 1,5s off

possesses.

2. The models

2.1 Overview

Selected wind turbines from the three most frequently represented manufacturers in Germany were installed:

- Enercon GmbH (V15NTU10045)
- Nordex SE (V15NTU10046)
- Vestas Wind Systems A/S (V15NTU10047)

All three sets are available in a savings set (V15NTU10048).

The list on the following page shows all implemented wind turbine types and their variants. Some plants have steel lattice towers instead of the usual steel or concrete towers.

Furthermore, individual models are equipped with mobile phone systems.

For smaller wind turbines, separate transformer houses are also included, which should be placed next to the wind turbines for design.

All models are available in EEP under the category

Immobilien/Gewerbe und Industrie/Fabriken und Anlagen

The models are named as follows:

"WKA_[turbine type]_[tower height]_[tower type if applicable]_TU1"

The turbine type generally indicates the rotor diameter.

Technical details can be found in the respective model description.

My tip:

With models of larger wind turbines, it is often worth placing them in the background away from the actual turbine to achieve a certain depth effect.

Manufacturer	Plant type	Rotor diameter	Achievement	Hub height	Maximum speed
Enercon	E-33/300	33m	300 kW	35m 41m	$39 \frac{1}{min}$
Enercon	E-40/5.40	40m	540 kW	48m 65m	$38 \frac{1}{min}$
Enercon	E-58	58m	1000 kW	59m 70m	$24 \frac{1}{min}$
Enercon	E-82 E2 2,300	82m	2,3 MW	85m 108m 138m	$18 \frac{1}{min}$
Enercon	E-101	101m	3,05 MW	99m 135m	$14,5 \frac{1}{min}$
Nordex	N27/150	27m	150 kW	30m 30m Lattice 40m	$40 \frac{1}{min}$
Nordex	N43/600	43m	600 kW	40m 60m 78m Lattice	$26,9 \frac{1}{min}$
Nordex	N62/1300	62m	1,3 MW	60m 69m Lattice 80m	$19,2 \frac{1}{min}$
Nordex	N117/2600 Gamma	117m	2,4 MW	91m 120m 140m	$13,2 \frac{1}{min}$
Vestas	V25-200kW	25m	200kW	30m 30m Lattice	$44 \frac{1}{min}$
Vestas	V47-660kW	47m	660kW	40m 45m 55m	$28,5 \frac{1}{min}$
Vestas	V80-2.0MW	80m	2,0 MW	60m 78m 100m	$19 \frac{1}{min}$
Vestas	V112- 3.3MW	112m	3,3 MW	84m 119m 140m	$17,7 \frac{1}{min}$

2.2 Axes and functions

All models can be individually controlled via the familiar axis dialog.

2.2.1 Nacelle alignment

The nacelle can be swivelled 360 degrees around the tower axis via the "Nacelle Orientation" axis to follow the wind.

2.2.2 Blade position

The blades can be rotated from the wind to the so-called flag position (axle position 0%) or to the operating position (axle position 100%) by means of the "blade position" axis. In some models, only the tip of the blade rotates according to the prototype.

2.2.3 Speed

The "Speed" axis determines the rotational speed of the wind turbine (0 to 100%).

Each type of wind turbine has a different, prototypical maximum speed depending on its rotor diameter and the associated blade tip speed.

2.2.4 Blade bending

For the rotor blades of all wind turbines, three different deflections can be set via the "blade bending" axis. These are used to simulate the different wind loads.

- Axle position 0% = switched off **or** hardly any wind
- Axle position 50% = switched on **and** light wind
- Axle position 100% = switched on **and** strong wind

2.2.5 System_Blink

The "System_Blink" axis only exists on models that have hazard lighting. This axis is firmly coupled to the non-stop axis, which simulates flashing. It can and should be ignored, as it is only needed for control via Lua script.

2.3 Sound modules

In addition to the wind turbines, an extensive set of sound material is included. This is divided into 22 wind noises from the rotor blades (for speeds between 4 and 43 revolutions per minute) and three machine sounds per manufacturer.

All sounds can and should be added to the wind turbines via the sound modules designed as real estate. The following sound modules are available:

- WKA_Sound_module_Rotor_langsam_TU1 [a]
- WKA_Sound_module_Rotor_mittel_TU1 [a]
- WKA_Sound_module_Rotor_schnell_TU1 [a]
- WKA_Sound_module_Rotor_Scriptbetrieb_TU1 [b]
- WKA_Sound_module_Machine_Enercon_TU1 [c]
- WKA_Sound_module_Machine_Nordex_TU1 [c]
- WKA_Sound_module_Machine_Vestas_TU1 [c]

The sound modules for the rotor are divided into three speed ranges, which are intended for manual operation. [a] Each of these modules has three axes, which reproduce a speed corresponding to the respective axis labeling in terms of sound.

(For example, "13_RPM" stands for 13 revolutions per minute.)

For the most realistic sound possible, it is recommended to use the rotor sound module for script operation. [b] to use. It must be as described in chapter "3.2.5 List of Sound module" can be included in the Lua script.

The machine sound modules [c] play different operating noises of the nacelle via three axes. They can also be controlled via the script.

All sound modules should be placed inside the respective wind turbine tower. It is a good idea to use the modules for the rotor sound at the level of the lower blade tip, i.e. about half a rotor diameter below hub height. The machine sound modules should be placed directly in the nacelle. This is the only way the Lua script can recognize the modules!

In addition, an unavoidable faltering of the rotor sound in particular can occur during a speed change in script mode!

3. The Lua-Script

In addition to the models, each set also includes two Lua script files, which can be used to control several wind turbines.

The files can be found in Explorer starting from the EEP root directory under

LUA\TU1\WKA_Steuermodul_TU1.lua und

LUA\TU1\WKA_Einstellungsvariablen_TU1.lua

The control module contains all the executing code for controlling the wind turbines and does not need to be viewed by the user. All of the variables discussed on the following pages are instead in the

WKA_Einstellungsvariablen_TU1.moon

It should always be opened and edited with a common editor (such as the Windows Editor or Notepad++).

Hint:

No previous programming knowledge is necessary to use the script!

All required entries are explained on the following pages.

Important programming knowledge is taught in other such programming basics.

mediates. All pseudo-data is enclosed in square brackets.

3.1 Functions

The script is designed to manage multiple models and control them randomly.

It serves the following basic functions:

- Synchronization of the hazard lights of all models equipped with it
- Wind tracking with specification of the main wind direction and a variance
- Random selection of systems to be switched on or off, determined by the specification of the energy demand
- Simulation of changing wind speeds
- Prototypical control of blade position and blade bending

Over a longer period of time, the script calculates the values specified in chapter "**Fehler! V erweisquelle konnte nicht gefunden werden.**" is continuously redefined in order to simulate the impression of changing wind conditions and fluctuating energy demand.

3.2 Setting options

In the following, the setting options in the script are documented. The reference lines of the respective sub-items are marked in the following screenshot of the code.

Changes to lines of code other than those discussed here are made at your own risk!

Programming Basics:

A single variable (used in points 3.2.1 to 3.2.4 and 3.2.6) ALWAYS has the following structure:

```
local [variable name] = [assigned value]
```

Programming Basics:

Comments in the code are used to document this. They are always started with a double hyphen in Lua. The text that follows in the respective line is ignored by Lua.

Programming Basics:

If you want to specify a number with a decimal place, you must always use the period as the comma. Using the usual comma will lead to errors!

```
1  -- =====
2  -- Den folgenden Block in der Anlagen-internen Lua-Datei >>>VOR<<< der Funktion EEPMain() einfüegen!!!
3  -- Werte wie in der Dokumentation beschrieben auf eigene Wunschwerte abändern
4  -- =====
5
6  WKA_Windstaerke = 5          -- Mittelwert der Windstaerke von 0 (windstill) bis 10 (Sturm)
7  [1] WKA_Windstaerke_Sig = -1 -- Mittelwert der Windstaerke ueber Zaehelersignal (10) einstellen
8  WKA_Windstaerke_Varianz = 0 -- Varianz der Windstaerke zwischen 0 und 3 in Schritten von 0.1
9
10 WKA_Windrichtung = 0       -- Mittelwert der Windrichtung von 0 bis 360 Grad, 0 Grad ist (aktuell noch) rechter Anlagenrand
11 [2] WKA_Windrichtung_Sig = -1 -- Mittelwert der Windrichtung ueber Zaehelersignal (40) einstellen
12 WKA_Windrichtung_Varianz = 0 -- Varianz der Windrichtung zwischen 0 und 180 Grad in Schritten von 0.5
13
14 WKA_Energiebedarf = 5      -- Mittelwert des Energiebedarfs von 0 (niedrig) bis 10 (hoch)
15 [3] WKA_Energiebedarf_Sig = -1 -- Mittelwert des Energiebedarfs ueber Zaehelersignal (10) einstellen
16 WKA_Energiebedarf_Varianz = 0 -- Varianz des Energiebedarfs zwischen 0 und 5 in Schritten von 0.1
17
18 WKA_Windraeder = {
19   -- Schema: (Windrad-Lua-Name, Modellwinkel in EEP (Ab EEP16.1 bitte 0 eintragen)),
20   [4] -- Beispiel: {"#1_WKA_Vestas_V112-3.3MW_84m_TU1",0},
21 }
22
23 WKA_Soundmodule = { -- Auflistung der Soundmodul-Lua-IDs durch Kommata getrennt
24   [5] -- Beispiel: "#13", "#14", "#15"
25 }
26
27
28 [6] WKA_extra_info = false
29
30 require("TU1/WKA_Steuermodul_TU1")
31
32 -- =====
33 -- Den folgenden Block in der Anlagen-internen Lua-Datei >>>INNERHALB<<< der Funktion EEPMain() einfüegen!!!
34 -- =====
35
36 [7] WKA_Ausfuehren()
37
```

Lua code of the script in its delivery state

3.2.1 Wind force

Wind force [2] defines how fast the wind turbines rotate when they are switched on. The variable `WKA_Windstaerke` indicates the basic wind strength (0 = no wind to 10 = storm), while the variable `WKA_Windstaerke_Varianz` defines the value (0 to 3) by which the basic wind direction may deviate randomly in positive and negative directions.

The values can be specified in increments of 0.1, but at best they are integer.

Important: All wind turbines switch off automatically under a wind force of 1! This should be taken into account when specifying the variance!

Example:

```
WKA_Windstaerke = 4
```

```
WKA_Windstaerke_Varianz = 1.3
```

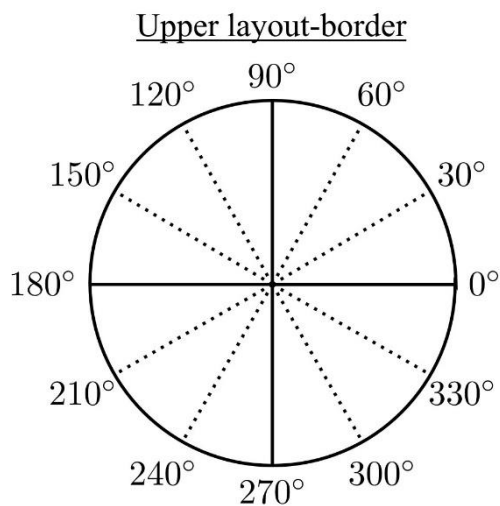
→ The wind strength will be between 2.7 and 5.3 degrees.

3.2.2 Wind direction

The wind direction [3] defines the direction in which the wind turbines should be oriented when they are switched on. The variable *WKA_Windrichtung* specifies the basic wind direction from 0 to 360 degrees, while the variable *WKA_Windrichtung_Varianz* defines the angle by which (0 to 180 degrees in steps of 0.5) the basic wind direction may deviate randomly in the positive and negative directions.

The values can be specified in increments of 0.5 degrees, but at best they are integer.

Important: The wind direction is analogous to the model angle in EEP:



Example:

`WKA_Windrichtung = 30`

`WKA_Windrichtung_Varianz = 20`

→ The wind direction will be between 10 and 50 degrees.

3.2.3 Energy Requirement

The energy demand [4] defines how many wind turbines are switched on at the same time. The variable *WKA_Energiebedarf* specifies the basic energy requirement. The value range extends from 0 (= 0%) to 10 (= 100%) and is converted by the script so that the corresponding percentage of wind turbines is switched on. The variable *WKA_Energiebedarf_Varianz* defines by which value (0 to 5) the energy demand may deviate.

The values can be specified in increments of 0.1, but at best they are integer.

Example:

WKA_Energiebedarf = 3

WKA_Energiebedarf_Varianz = 2

→ The energy demand will be between 10% and 50%.

Programming Basics:

A table, also known as an array, is made up of several variables. It is defined as follows and filled with values (in this case value1 and value2):

```
local [table name] = {[value1] , [value2]}
```

Programming Basics:

Tables can also be nested inside each other to create multidimensional structures. A two-dimensional table is defined and populated as follows:

```
local [table name] = {[value1a] , [value1b] },{ [value2a] , [value2b]}
```

When defining tables, it is important to set curved brackets correctly (see colored markings)!

In this example `{[Value1a] , [Value1b]}` is a small table , together with the second small table `{[Value2a] , [Value2b]}` it is sorted into the large main table `{ , }`.

3.2.4 List of wind turbines

The list of wind turbines [5] contains all the models to be controlled by the script. The outer brackets must always remain unchanged. Only the lines in between (in this example lines 31 to 33) may be changed and, of course, supplemented by additional lines.

Each line contains information about exactly **one wind turbine**. The informations always look as follows:

```
{ [Lua model name], [model angle in EEP] },
```

The Lua name can be read from the dialog box of the respective model in EEP. It is important to make sure that the name in the script is given with double goose feet. In principle, only the number after the diamond (but including the diamond) is required, but the complete name can also be specified.

The model angle must be equal to the "rotation Z" of the model in EEP.

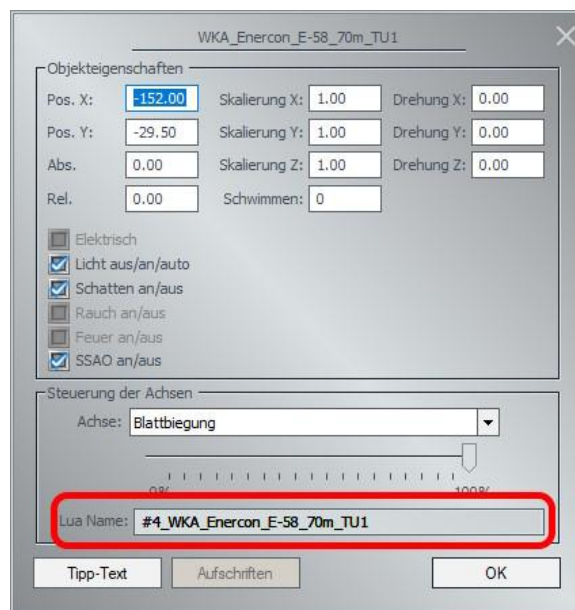
This value is only required if an EEP version **less than EEP 16.1** is used. Otherwise, any number (preferably 0) must be entered there.

Example:

```
WKA_Windraeder = {  
  {"#798_WKA_Enercon_E-40/5.40_48m_TU1",10},  
  {"#235_WKA_Vestas_V112-3.3MW_119m_TU1",120},  
}
```

→ The model with the ID 798 has an angle of 10 degrees, the model with the ID 235 has an angle of 120 degrees.

In any case, care must be taken to ensure that there is a comma after each of the inner brackets!



The required Lua name is highlighted in red here.

3.2.5 List of Sound modules

If sound modules are to be used, they must first be set as described in chapter "2.3 Sound module".

Both the rotor sound module for script operation and the machine sound modules can be addressed via the Lua script!

To do this, the Lua-names of the suitable models must be entered in a simple list in the table `Soundmodule` [6]. Again, pay attention to the strict use of quotation marks and the separation by commas!

The full Lua names should be given aswell.

Example:

```
WKA_Soundmodule = {  
    "#352_WKA_Soundmodul_Rotor_Skriptbetrieb_TU1",  
    "#728_WKA_Soundmodul_Maschine_Enercon_TU1",  
}
```

3.2.7 Extra-Infos

The variable `extra_info` [7] defines how much information about current events should be displayed in the script in the EEP event window. If the variable is set to `false`, only accessible information is provided, after which the script runs "silently" and controls the wind turbines. If the value is set to `true`, the script will deliver further messages, for example when a wind turbine is switched on or off.

3.3 Integration into EEP

3.3.1 Inserting the Setting Variables into the In-Layout Lua Script

In order for the script to work, the discussed code must be inserted into the plant-specific Lua script. The button for the plant-specific Lua script can be found in the top menu bar of EEP on the far right.

So the code area between the two multi-line comment blocks must be copied from the file

WKA_Einstellungsvariablen_TU1.lua

and pasted into the Lua script before the *EEPMain()* function call.

Analogous to the previous chapters, all wind turbines and sound modules can now be entered here and the parameters for wind strength, wind direction and energy demand as well as their variances can be set.

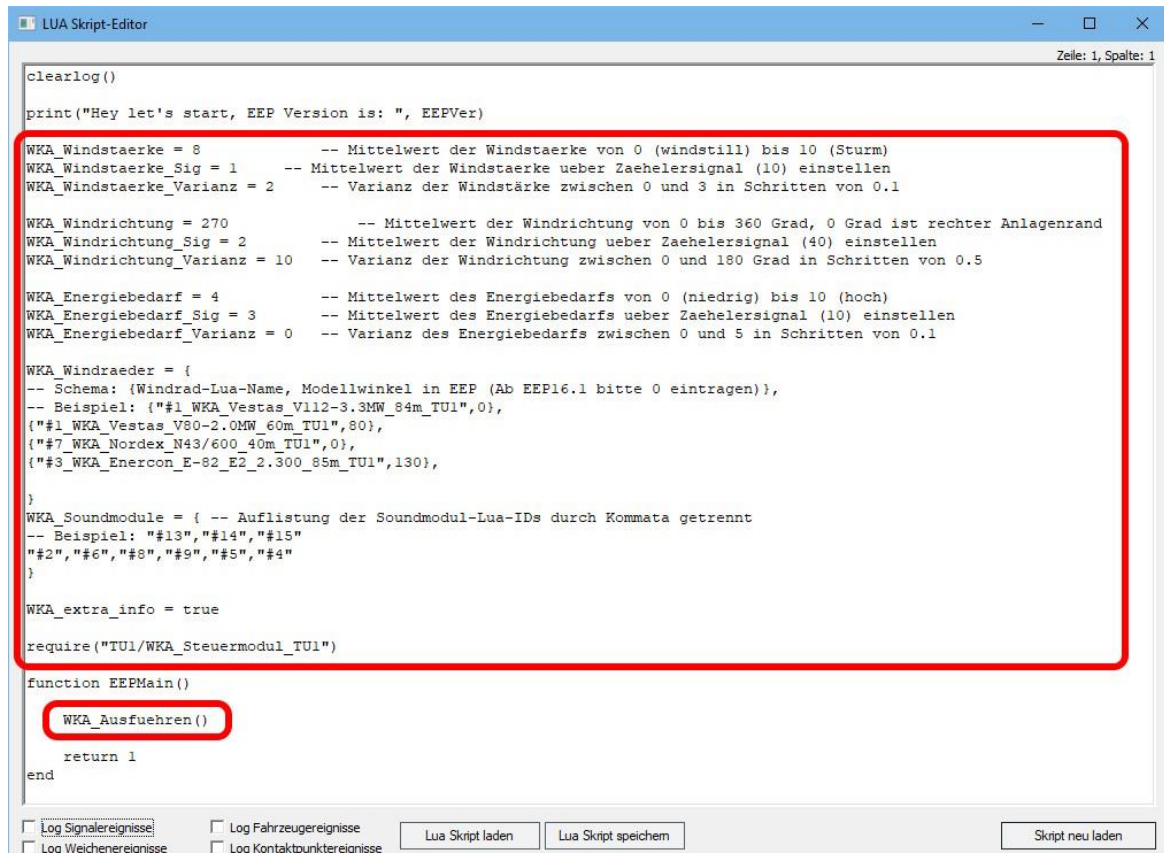
A confirmation via the "Reload script" button saves the system's internal Lua script. If you now look into the EEP event window, which can be activated in the program settings, you will notice a direct response from the script if all data is entered correctly.

To start the routine for controlling the wind turbines, the line below the second comment block must be copied from the file

WKA_Einstellungsvariablen_TU1.moon

within the *EEPMain()* function.

For a new plant, the code will look like this. The added lines of code are color-coded and correspond to those described in chapter "4. Video-Tutorial" linked video tutorials.



```
clearlog()
print("Hey let's start, EEP Version is: ", EEPVer)

WKA_Windstaerke = 8      -- Mittelwert der Windstaerke von 0 (windstill) bis 10 (Sturm)
WKA_Windstaerke_Sig = 1  -- Mittelwert der Windstaerke ueber Zaehlersignal (10) einstellen
WKA_Windstaerke_Varianz = 2  -- Varianz der Windstaerke zwischen 0 und 3 in Schritten von 0.1

WKA_Windrichtung = 270  -- Mittelwert der Windrichtung von 0 bis 360 Grad, 0 Grad ist rechter Anlagenrand
WKA_Windrichtung_Sig = 2  -- Mittelwert der Windrichtung ueber Zaehlersignal (40) einstellen
WKA_Windrichtung_Varianz = 10  -- Varianz der Windrichtung zwischen 0 und 180 Grad in Schritten von 0.5

WKA_Energiebedarf = 4    -- Mittelwert des Energiebedarfs von 0 (niedrig) bis 10 (hoch)
WKA_Energiebedarf_Sig = 3  -- Mittelwert des Energiebedarfs ueber Zaehlersignal (10) einstellen
WKA_Energiebedarf_Varianz = 0  -- Varianz des Energiebedarfs zwischen 0 und 5 in Schritten von 0.1

WKA_Windraeder = {
  -- Schema: {Windrad-Lua-Name, Modellwinkel in EEP (Ab EEP16.1 bitte 0 eintragen)},
  -- Beispiel: {"#1_WKA_Vestas_V112-3.3MW_84m_TU1",0},
  {"#1_WKA_Vestas_V80-2.0MW_60m_TU1",80},
  {"#7_WKA_Nordex_N43/600_40m_TU1",0},
  {"#3_WKA_Enercon_E-82_E2_2.300_85m_TU1",130},
}

WKA_Soundmodule = { -- Auflistung der Soundmodul-Lua-IDs durch Kommata getrennt
  -- Beispiel: "#13", "#14", "#15"
  "#2", "#6", "#8", "#9", "#5", "#4"
}

WKA_extra_info = true

require("TU1/WKA_Steuermodule_TU1")

function EEPMain()
  WKA_Ausfuehren()
  return 1
end
```

Lua script editor of the attachment with entered and configured lines of code

With another click on the "Reload script" button, the control of the wind turbines will begin.

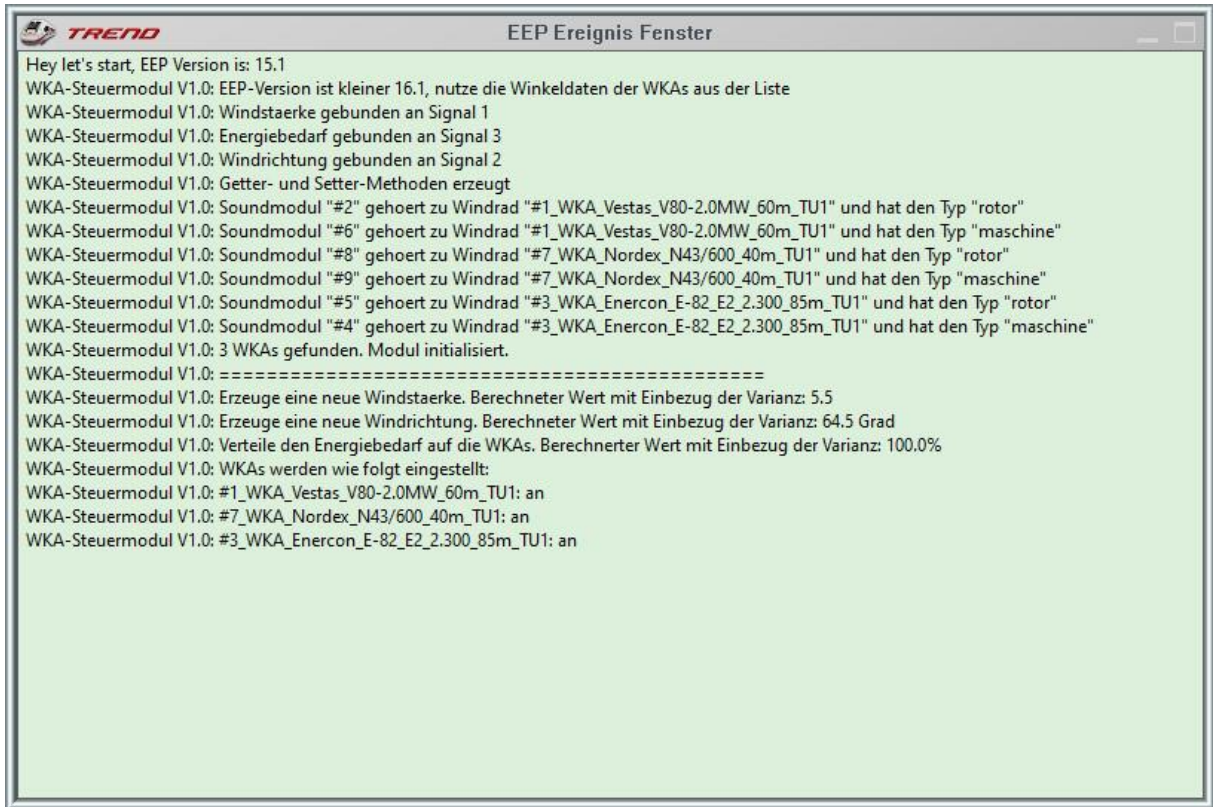
The script now randomly recalculates the values for wind strength, wind direction and energy demand from the specified base values and variances at longer intervals (approx. 10 minutes). This wait time is skipped when the attachment script is reloaded.

Depending on the variance entered, corresponding changes to the wind turbines will now become apparent over time. The greater the variance, the more random these changes will be.

If sound modules have been placed and entered, only the sounds of the individual wind turbines can be heard. Unfortunately, the transitions between the individual speeds can be harsh, but this cannot be avoided.

In the meantime, it is not possible to control the properties manually, the script will always force the models into the position it specifies.

In the EEP event window, when the WKA_extra_infos is enabled, you can read continuous information about the current status:



```
Hey let's start, EEP Version is: 15.1
WKA-Steuermodul V1.0: EEP-Version ist kleiner 16.1, nutze die Winkeldaten der WKAs aus der Liste
WKA-Steuermodul V1.0: Windstaerke gebunden an Signal 1
WKA-Steuermodul V1.0: Energiebedarf gebunden an Signal 3
WKA-Steuermodul V1.0: Windrichtung gebunden an Signal 2
WKA-Steuermodul V1.0: Getter- und Setter-Methoden erzeugt
WKA-Steuermodul V1.0: Soundmodul "#2" gehoert zu Windrad "#1_WKA_Vestas_V80-2.0MW_60m_TU1" und hat den Typ "rotor"
WKA-Steuermodul V1.0: Soundmodul "#6" gehoert zu Windrad "#1_WKA_Vestas_V80-2.0MW_60m_TU1" und hat den Typ "maschine"
WKA-Steuermodul V1.0: Soundmodul "#8" gehoert zu Windrad "#7_WKA_Nordex_N43/600_40m_TU1" und hat den Typ "rotor"
WKA-Steuermodul V1.0: Soundmodul "#9" gehoert zu Windrad "#7_WKA_Nordex_N43/600_40m_TU1" und hat den Typ "maschine"
WKA-Steuermodul V1.0: Soundmodul "#5" gehoert zu Windrad "#3_WKA_Enercon_E-82_E2_2.300_85m_TU1" und hat den Typ "rotor"
WKA-Steuermodul V1.0: Soundmodul "#4" gehoert zu Windrad "#3_WKA_Enercon_E-82_E2_2.300_85m_TU1" und hat den Typ "maschine"
WKA-Steuermodul V1.0: 3 WKAs gefunden. Modul initialisiert.
WKA-Steuermodul V1.0: =====
WKA-Steuermodul V1.0: Erzeuge eine neue Windstaerke. Berechneter Wert mit Einbezug der Varianz: 5.5
WKA-Steuermodul V1.0: Erzeuge eine neue Windrichtung. Berechneter Wert mit Einbezug der Varianz: 64.5 Grad
WKA-Steuermodul V1.0: Verteile den Energiebedarf auf die WKAs. Berechneter Wert mit Einbezug der Varianz: 100.0%
WKA-Steuermodul V1.0: WKAs werden wie folgt eingestellt:
WKA-Steuermodul V1.0: #1_WKA_Vestas_V80-2.0MW_60m_TU1: an
WKA-Steuermodul V1.0: #7_WKA_Nordex_N43/600_40m_TU1: an
WKA-Steuermodul V1.0: #3_WKA_Enercon_E-82_E2_2.300_85m_TU1: an
```

3.3.2 Control by count-signal

The main values (not the variances!) for wind strength, wind direction and energy demand can also be linked to suitable meter signals in order to be able to control these guideline values directly from EEP.

The following counter signals should be used:

- Wind speed: Zählersignal_10_BH2
- Windrichtung: Zählersignal_40_BH2
- Energy Requirement: Zählersignal_10_BH2

In order to connect these signals, the previously uncommented variables exist

- WKA_Windstaerke_Sig
- WKA_Windrichtung_Sig
- WKA_Energiebedarf_Sig

If a value of -1 is assigned to these variables, the script uses the values described in the chapters "3.2.1 Wind force" to "3.2.3 Energy Requirement" assigned values.

If you replace the -1 with the appropriate signal ID of the count-signals, their signal positions will be used as a default from now on. The values for energy demand and wind speed correspond directly to the display of the meter signals (0 to 10), the value of the 40-count meter signal for the wind direction must be multiplied by ten to arrive at the angle (e.g. the display 06 corresponds to the angle value 60 degrees).,

The count signals can be downloaded from the website of my colleague Benjamin Högl (BH2) at the following link:

[EMAPS-EEP](#)

4. Video-Tutorials

I have visualized all the settings presented and the integration into the system in two videos on YouTube.

The main video can be accessed via the following link:

[Full-length video tutorial \(YouTube\)](#)

Due to the length of the main video, I have also created a short version of the video, which only shows the most important aspects of integrating the script into EEP and is only a fraction of the length of the full video tutorial. This short version can be viewed via the following link:

[Video-Tutorial in Kurzversion \(YouTube\)](#)