

Sound Creator

Sound Editor

Valid from software version
ZIMO Sound Tools V2.00

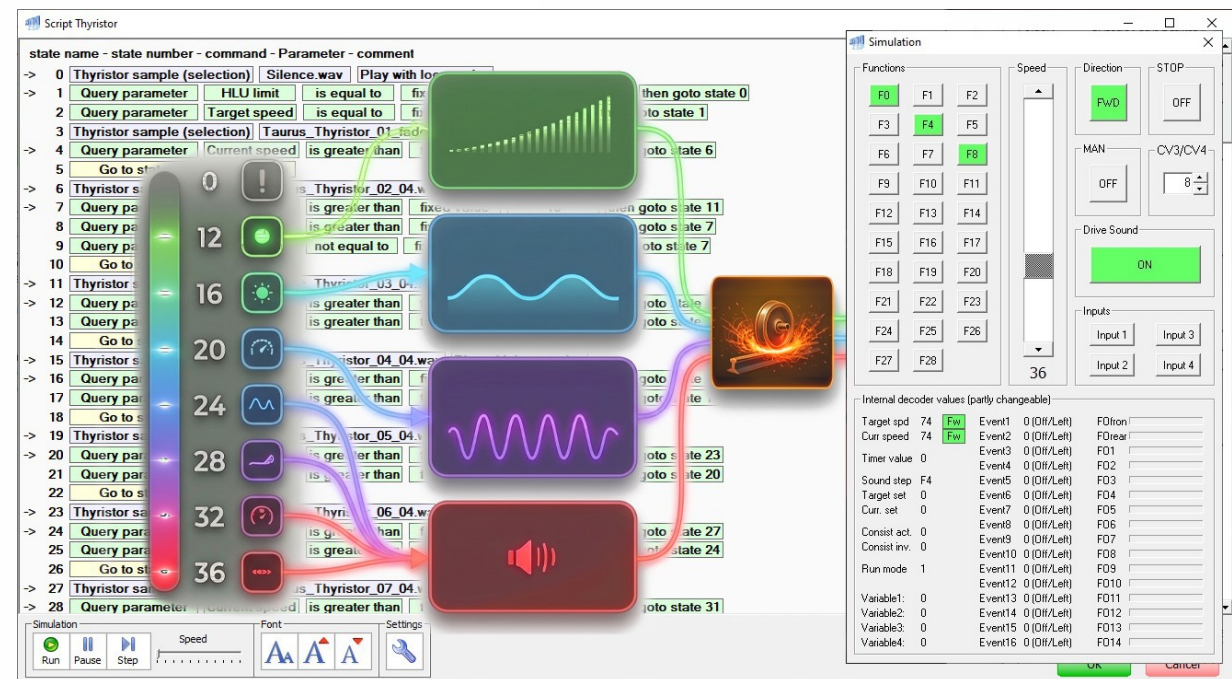
RELEASES:

October 2022
March 2024
June 2025
July 2026

Scripts

OPERATING INSTRUCTIONS

Introduction to programming of Scripts
in ZIMO Sound Creator
and ZIMO Sound Editor



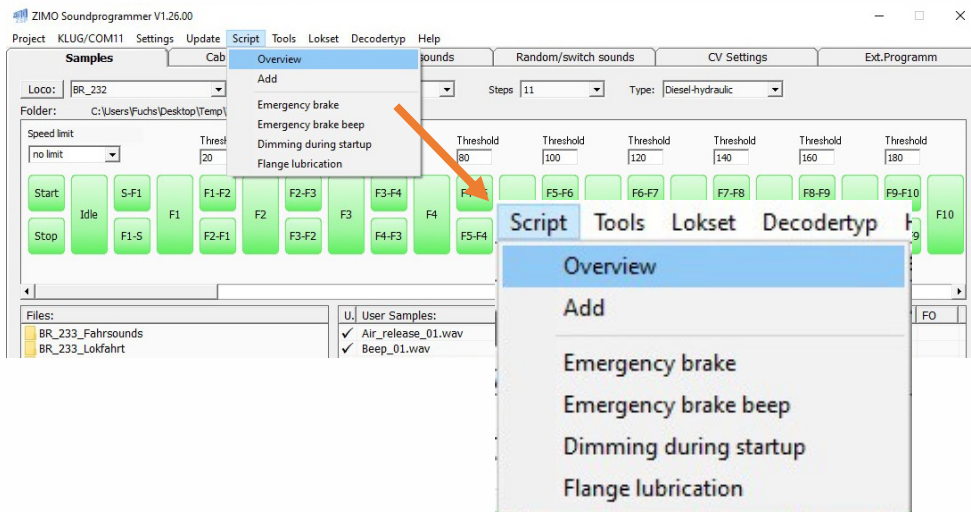
As of July 2026: The **ZIMO Sound Editor** replaces the previous **ZPP Konfig** application. It is still available for download as a combined installation package. During installation, you can choose whether to install only the **ZIMO Sound Editor**, only the **ZIMO Sound Creator**, or both programmes. The functionality and appearance remain largely unchanged; however, the ZIMO Sound Editor has been enhanced with new features, including support for **ZIMO stereo decoders**. All existing **ZIMO sound projects** can still be opened and edited as usual.

General:

With **ZIMO Sound Creator** and **ZIMO Sound Editor**, it is possible to write pre-programmed sequences and dependencies with the help of **scripts**, and thereby significantly expand the functions and possibilities of ZIMO decoders. Both programmes allow to create these sequences in an input mask. For each sound project or CV set, it is possible to add (or import) up to 16 scripts on MS/FS decoders from MS-FW 4.207 onwards, and up to 8 scripts on MX sound decoders from MX-FW 39.0 onwards. However, MX Sound decoders are subject to certain limitations compared to the MS decoder family; for example, only 6 independent sound channels are available, and two samples can only be played at the same time in specific cases in a script. The following applies to both decoder families: up to **256 states** (commands or lines) can be programmed per script. Individual scripts can be deactivated via the corresponding bits of CVs **#837** and **#843** (CV #837 Bit0-7 for Scripts 1-8, CV #843 Bit0-7 for Scripts 9-16).

Many values and parameters can be implemented as **Script-CVs (#980 to #1019)**, which can be changed later in the decoder without the need to reload the entire sound project. Common use for Script-CVs include sample volumes, timer durations or speed steps. Of course, the permitted values of the respective CVs must be taken into consideration – please refer to the decoder manuals for [MX](#) and [MS decoders](#).

From MS-FW 4.250 onwards, scripts can also be used in **CV sets** in **MN decoders**. CV sets containing scripts can only be created in **ZIMO Sound Editor** and loaded directly into the decoder via **KLUG**. Another option is to use the **CV-Set Creator**, which allows you to make a special .zpp file and load the CV-set onto the MN decoder via a USB stick using MXULF. However, please note that scripts with sound samples are not possible with MN decoders, or more clearly, the corresponding commands (queries for sound samples, diesel steps, thyristor and motor samples, etc.) are ignored. It makes more sense to influence function outputs or driving behaviour of MN decoders via scripts. Although, in certain cases, a timer set to the length of a sound sample can be used instead of the actual sample.



Writing and editing scripts:

From version V1.21.22 onwards, scripts can be created and edited in both ZIMO Sound Creator and ZIMO Sound Editor. In ZIMO Sound Editor, however, existing scripts in coded projects cannot be viewed, edited or deleted, as these are specifically made by the sound provider to suit the project's characteristic sound sequence. It is only possible to deactivate them here via the corresponding bits in CVs **#837** and **#843**, which can significantly restrict the functionality of the sound project. But it is possible to create or import your own scripts into coded projects.

The scripts can be found in the **ZIMO Sound Creator** under the **'Script'** menu. Here, you can either select **'Overview'** or **'Add'**, or choose one of the existing scripts directly for editing. Clicking on 'Overview' opens the separate main window, which provides a list of existing scripts and editing options.

In the **ZIMO Sound Editor**, the scripts can be found under the **'Scripts'** tab. The editing methods described here are the same in both programmes.

To create a new script, click the **Add** button and a field labelled "Script1" (or the next available script number) will appear. When a script is selected in the overview (highlighted in blue), you can press the F2 key to give the script a distinctive name, making it easier to identify it later on.

Double-clicking on a script opens the script editor.

Add Import from file Export to file					
Active	Name	Keys	Outputs	CVs	Sounds
☒	Fan-2016	F12		981,982	Rh2016_Lüfter_Stand_02.wav, Rh2016_Lüfter_F0-F2_Start_10.wav, Rh2016_Lüfter_F1-1
☒	Spring brake	F19		988,989,990	Rh2016_Federspeicherbremse_01.wav, Rh2016_Federspeicher-anlegen_01.wav, Rh2016
☒	Emergency brake	F20		983,984	Rh2016_SiFa-Zwangsbremsung_01.wav, Rh2016_Störung_01.wav, Rh2016_SiFa_01.wav,
☒	Cab light timer			981	

Export to file: A script can be exported to any location on the computer, for example to use it in another sound project or the CV-Set Creator, or to share it with someone else. It is saved in .zsc file format with a name of your choice.

Import from file: Once a script has been saved or exported, it can be imported into another sound project or CV-Set. IMPORTANT: Generally, only the index numbers of sound samples within the current sound project are saved, and these may vary from project to project. Therefore, sound samples must be manually checked and adjusted if necessary after importing into another project!

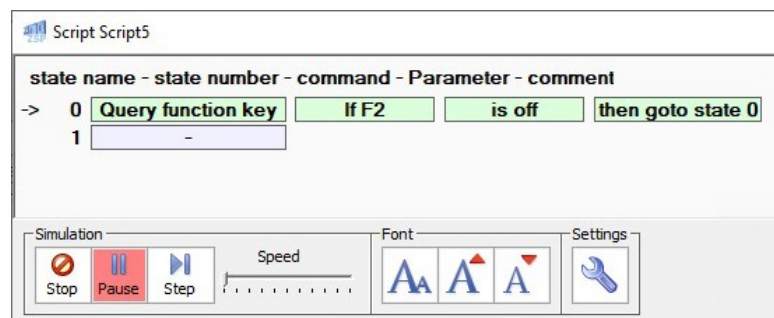
Window size: The size of the Script Editor window can be adjusted and it can also be maximised. With long and complex scripts, this offers the advantage of being able to see more states at once.

Delete: A script can be selected by simply clicking on it (highlighted in blue) and then deleted using the **Delete** key.

The Overview window also displays all elements linked to the respective script. These include function keys, function outputs, script CVs used and assigned sound samples.

Enable / Disable: In the Overview window, you can enable or disable a script ticking the box at the left side. This also directly affects the values of CVs **#837** and **#843**.

Script Editor



Add or delete states:

Clicking on a new empty state opens the selection menu for the various commands:

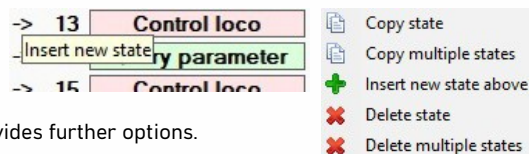
Sound	Queries	Set	Miscellaneous
Start sound	Query function key	Control loco	Go to state
Stop sound	Query parameter	Load timer	Generate random value
Wait for end of sound	Query reed input	Set diesel step	I2C command
Thyristor sample (selection)	Query event	Switch output on	
Wait for end of thyristor sample	Query variable	Set event	
Thyristor pitch	Query sound active	Set effect	
Suppress brake squeal	Read CV (only MS)	Set variable	
EMotor sample	Read CV bit (only MS)	Change variable/event	
Wait for end of E-Motor sample		Set parameter	
Take over diesel sequence		Set PWM group	
Set volume (only MS)		Set CV (only MS)	
		Set CV bit (only MS)	Delete state

By **left-clicking** on the number of an existing state, you can insert a new state above it, or delete it by clicking on an empty state.

Right-clicking on a state's number provides further options.

At the bottom left of the script editor is the **'Simulation'** field. This will be explained later in more detail, as it is used to test a finished script.

You can also adjust the font and font size of the text to suit your preferences.



Writing and reading scripts:

Scripts are best 'read' like a text: you start at the top left and read line by line down to the bottom right. This is exactly how the ZIMO programmes, and later the decoder, handle the scripts when running them. Depending on the type of **command**, several **parameters** may be selected. In principle, you can generate a wide variety of sequences using scripts. However, there are a few basic rules to bear in mind when writing them.

The commands check for a specific state: if its condition is met, the programme jumps to the state set in the last parameter and continues the sequence from that point. However, if the check returns a negative result, the sequence continues in the next following state. In the following example, the sequence remains in the same state until the F2 key is pressed:

-> 0 Query function key If F2 is off then goto state 0

The **"Query function key"** command checks whether the **F2** function key is **off**. If this is the case, the script jumps to **State 0**, i.e. remains on the current line.

If **F2 is activated** – either on the controller of the digital command or via the online controller of the ZIMO Sound Creator or ZIMO Sound Editor – the query returns a negative result – and the script automatically proceeds to the next line, i.e. to the next state.

In the next state any action should be defined that is to be executed when the function key is activated – for example, the start of a sound, a further sequence or query.

Once the action has been completed, the key's state must be checked again. In this example, this is done by checking the status as **ON**:

state name - state number - command - Parameter - comment

-> 0 Query function key If F2 is off then goto state 0

1 Start sound Cl 66_Horn-low_mid_01.wav Volume = -3 dB (71%) Play sample once Play all if sound ends Start sound now Speaker 1

2 Wait for end of sound

-> 3 Query function key If F2 is on then goto state 3

4 Go to state 0

Whatever is started in a script must also be terminated there!

Whether it involves sound samples, function outputs, driving behaviour, or intervention in the sequence: if an action is triggered in a script, you must ensure that it is also terminated there.

The end of a script must not remain open!

A closing command (e.g. 'Go to State X') is essential to ensure a continuous script sequence.

If this is missing, the script will stop and cannot be continued. Therefore, the last line of a script should always contain a closing command. Most of the time, the script loops back to the beginning.

However, it is not strictly necessary to jump from the last state to State 0; it is merely important that the script forms a closed loop and does not remain open or stop at any point.

Parameters set as fixed values in the script can be entered and changed by clicking in the respective field.

Sound volumes are set either as a fixed value via the slider, or by selecting a Script-CV. If a Script-CV is selected, the volume can also be changed later once the project has already been loaded onto the decoder.

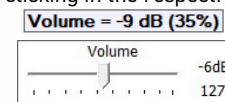
This applies to all parameters that can be selected as CV #980–#1019 and is also recommended by ZIMO. Changes can be made much more easily on the decoder without having to reload entire projects.

Checking for the opposite

Scripts work on a similar principle to the if/else query in programming. For example: if pressing an F key should trigger an action, the 'off' state of the key should be queried.

8 Go to state 0

9 -



Description of commands:

Sound:

Start sound: Plays a **sound sample** that is selected in the parameter 'Samples', which lists all the samples of the sound project. As long as no sound has been selected, the field is displayed in red. The next parameter determines the **volume** at which the sound is played. In general, the volume can only be reduced, not augmented. By clicking into the volume field, you can either reduce the volume using the slider, or select one of the CVs #980-#1019. Then, you can set whether the sound should be played once or in a loop, or how many times it should be repeated. With **"Play sample once"**, the sound is played entirely from start to finish without any further commands. The next parameter, **"Skip forward at sound end..."**, has no effect, when the sample is only played once. If a sound has loop markers and is started with **"Loop until Stop sound command"**, there are two ways to end it: **"Play all if sound ends"** plays the current loop section in full until the end, whilst **"Jump to loop end marker if sounds end"** skips the current loop and jumps to the end marker in the sample. If a sound sample continues beyond an end marker, this section is also played. The loop markers for the sound samples must be edited in the Sound Creator or set in the ZIMO Sound Editor when adding your own samples. Sounds or loops can also be started with a fixed **"loop count"** (2-8). In this case, either the entire sound sample is repeated x times, or the loop section is repeated if a loop is set. As a final parameter, you can specify to **"Start sound now"** or if the end of the previously started sound should be waited for. **"Start sound after last sound finished"** replaces the "wait for end of sound" command and so you can save one additional state of the script. Finally, it can be select from which **speaker** (channel) the sound is to be played.

From MS decoder firmware version 5.29.0 and ZIMO Sound Tools version 2.00 onwards, regardless of the decoder type within the MS and FS families, all sounds can be assigned to different 'speakers' and each can be filtered separately. In particular, decoders with two speaker outputs – the **MS420**, **MS950** and **MS990** – can thus reproduce a more prototypical sound. However, even with decoders having only one speaker output, this offers the advantage of being able to apply different filters to the sounds assigned to it.

Stop sound: If the sound was started with a loop, this command must be set to stop it. If it was started to play once or with a specific loop count, this command is not necessary; the sound will stop automatically.

Wait for end of sound: Before a new sound sample is started, this option allows you to wait until the current sound sample has played completely (including all loops or fixed loop counts). The command also ensures an audio transition to the next sound sample – thus avoiding gaps in the sound. However, this command has no meaning for the sequence of the script; the subsequent queries continue to run!

Thyristor sample (selection): Selects a sound sample for the thyristor effect. The pitch can be controlled either via the corresponding CVs #289-292 or via the script command "Thyristor Pitch". The dynamic volume is set via CVs #293-295, #357-358. The programme does not distinguish which kind of sound project is currently being edited, which is why this function is also visible in steam and diesel projects. Thyristor samples are usually started with **'Play with loop active'**. If manual transitions are desired, for example between individual thyristor stages, **'Play sample once'** can also be activated. Each thyristor sample started in the script replaces a previous one (including the one under "Autom. sounds"), so there is no need to manually terminate a playing sample.

Sound
Start sound
Stop sound
Wait for end of sound
Thyristor sample (selection)
Wait for end of thyristor sample
Thyristor pitch
Suppress brake squeal
EMotor sample
Wait for end of E-Motor sample
Take over diesel sequence
Set volume (only MS)

Kanal
All channels
Speaker 1
Speaker 2
Speaker 1/Fw. 2/Bw
Speaker 2/Fw. 1/Bw
1/2 Random

Wait for end of thyristor sample: If a thyristor sample is started with "Play sample once", this command can be used to control the sequence of the script to wait for the sample to play in full before starting the next Thyristor sample. Without this waiting function, the playing sample would be immediately replaced by the newly started one, which could lead either to premature termination or, in the worst case, to an audible gap in the sound. Please note that Thyristor samples started with 'Loop' cannot be waited for in this way.

Thyristor pitch: This command controls the pitch of the thyristor, linked to specific speed steps. In parameter 1, you define the pitch of the thyristor at medium drive step, and in parameter 2 the maximum pitch of the thyristor at maximum drive step (0-255 corresponds to the pitch in %). The medium drive step is defined in parameter 3, and the maximum drive step in parameter 4 (0-255 correspond to the internal drive step). The settings here correspond to CVs #291-293 and #838. Once used in a script, the effect of these CVs cannot be returned to the decoder's automatic CVs, and any further changes must be made in the script.

Suppress brake squeal: Determines whether the brake squeal should be suppressed during deceleration.

EMotor Sample: This allows you to select a sound sample for the electric motor effect. The dynamic settings for the electric motor are made via the corresponding CVs (#296-299, #372, #373, #844) and it can be played either **'Play with loop active'** or **'Play sample once'** just like thyristor samples. Also, each newly started electric motor sample replaces the previous one.

Wait for end of E-Motor sample: If an electric motor sample has been started with "play once", this command waits until it has finished playing completely.

Take over diesel sequence: As soon as the diesel sequence is taken over with "deactivate automatic and start diesel sounds manually in the script", the audio channel of the diesel sequence (diesel engine or fan in electro projects) is transferred to the script. Any sequence sounds still playing should be terminated using "Stop sound", and the desired new sequence must then be recreated using sound samples in the script. The command "Diesel process transferred back to automatic at sound step" returns the diesel sequence back to automatic. Here, a diesel step can be defined to which the sequence should be linked. To ensure that control is transferred the correct diesel stage, the command "set diesel step" can be used beforehand.

Set Volume (MS/FS only): Sets the general volume of the entire sound project to a value between minus 36dB (3%) and plus 6dB (200%). +/- 0dB (100%) corresponds to the value =64 in **CV #266**, 200% corresponds to the value =128. Mute (0%) can also be selected as well as a Script-CV for the volume. The fade time from the current volume of the sound project to the desired volume in the script can be set as a fixed value in 0.1s increments (comparable to **CV #314** [Mute fade-in/fade-out time]), or selected as a Script-CV.

Queries:

Query function key: Checks whether a function key is on or off and specifies where to go if the check is positive. Clicking in the "then go to state" field specifies a target state, confirmed by pressing ENTER. The desired F-key can also be specified using one of the script CVs #980-#1011; values =0 to =28 correspond to keys F0 to F28. In addition, the MAN key (fixed value only) can be queried.

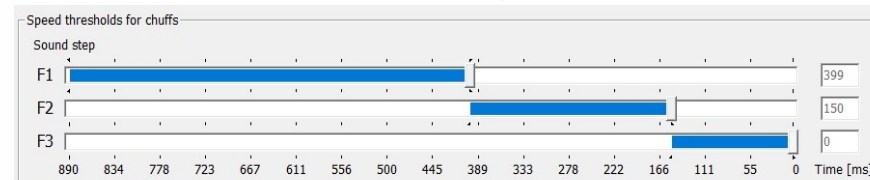
Query parameter: Checks whether the queried **parameter** (see detailed description below) is equal to, not equal to, greater than or less than the defined value (either fixed value or from CV). It can also be compared with another parameter, e.g. whether the current speed is equal to the target speed. Just like the "Query function key", in the last parameter you define where the script should go to if this query is positive. If this query is negative, the sequence continues in the following state.

Queries
Query function key
Query parameter
Query reed input
Query event
Query variable
Query sound active
Read CV (only MS)
Read CV bit (only MS)

The following parameters can be set or queried:

- **Travel speed:** Current internal speed level (0–255¹).
- **Target speed:** Speed level (0–255) set on the controller to which acceleration or deceleration is changing to.
- **Sound on/off:** Checks whether the sound is on or off. This query is preferable when starting other sounds, rather than querying the sound key, e.g. F8.
- **Target set:** Checks which loco set will be switched to next (e.g. when change of sets with switch key via CV #345). The set index is defined in the script as follows: Set 1 = Index 0, Set 2 = Index 1, etc. When working with transition sets in a diesel or electric project (e.g. for dual-mode vehicles, CV #346 Bit2), the first set has an index of 0 and the fourth complete set has an index of 1. The transition sets cannot be queried, the index changes at the end of each transition set.
- **Current set:** Checks which loco set is currently being played. In steam projects, a set change takes place immediately. The current set and the target set are therefore the same, and only one query is needed.
- **Diesel step:** Checks which speed step the diesel sequence sound is currently in. The term 'diesel' step is used in the context of the sound sequence and may also include different fan speeds of an electric locomotive.
- **Timer value:** Counting down from the value set in 'Load timer' to zero. This query allows you to trigger various actions whilst the timer is running.
- **Acceleration value:** Number of speed levels by which acceleration occurs. If the vehicle is travelling at a constant speed or decelerating, the script returns the value 0.
- **Deceleration value:** Number of speed levels by which the vehicle decelerates.
- **Current direction of travel:** Checks the current direction of travel (0= forward, 1=reverse).
- **Target direction of travel:** Checks which direction the locomotive is supposed to be travelling in according to the control (e.g. locomotive still decelerating in the 'previous' direction before the change of direction).
- **Consist active:** Checks (Yes, No) whether the decoder is currently in Consist mode.
- **Consist inverted:** Checks whether CV #19, bit 7 = 1 is set (inverts the direction of travel in Consist mode) and whether Consist mode is active.

- **HLU limit:** Current limit of the HLU brake section. The values in the script CVs are: =0: HLU emergency stop (braking without delay), =2: Stop, =3: UH, 4= U, =5: LU, =6: L, =7: FL, =8: Driving.
- **ABC limit:** Current limit of the ABC brake section. The values in the script CVs are: =2: Stop, =5: Slow, =8: Driving.
- **DC limit:** Current limit of the DC brake section. The values in the script CVs are: =2: Stop, =8: driving.
- **Current Limit:** Lowest limit of the three previous parameters. To avoid having to query all three in the script, this parameter checks for the lowest value found.
- **Chuff interval (MS/FS only):** Queries the time in milliseconds between individual chuffs (see the [ZIMO Sound Creator](#) manual for further details).



- **Chuff/revolution (MS/FS only):** Queries the current single chuff (1–16). The number of chuffs is set exclusively in the ZIMO Sound Creator (under 'Chuffs, 2–16'). Please note that the index, not the number of the chuff, applies here, i.e. chuff 1 = value 0, chuff 2 = 1, etc...
- **Chuff speed index (MS/FS only):** Queries the speed level currently being played in the sequence ('horizontal' chuff level 1–10).
- **Chuff load index (MS/FS only):** Queries the current load level H, M or L.
- **Drain active (MS/FS only):** Checks the status of the cylinder drainage function.
- **Current operating mode (MS/FS/MN only):** Queries the current mode (digital or analogue). "None"=value 0, "Digital mode"=1, "Analogue mode"=2.



Query reed input: The status of an input (1–4) is queried. This status can either be "Active", "Inactive", or "Event occurred" or "No event occurred". If the status of a reed input changes, an "event" is triggered; if the status remains unchanged (on or off), no event occurs. The input can be selected via a CV, with values 0–3 for inputs 1–4.

Query event: Queries the status of an event (1–16). Events can be set in sound samples or triggered in another script. The following values apply as comparison values in scripts: 0 = off/left, 1 = on/right, 2 = centre. Any higher values up to 255 can also be queried.

¹ Calculate based on the selected drive step mode. With 128 drive steps, the value must be multiplied by approximately 2

Query variable: Queries the value of a variable (1–4). Variables are used to count specific actions or to 'store' specific information. As an example: after every third time an F key is pressed (also queried in the script), an additional sound should be triggered. The variable is set to the value of 3 for this example, and after each press (and release) of the key, the value of the variable is reduced by 1, using the "Change Variable" command. As soon as the value reaches 0, the sound will be played. Generated random values, CV values or CV bits can also be stored in variables.

Sound active query: This allows you to determine whether the sound most recently started within the script is currently playing or has finished.

Read CV (MS/FS/MN only): Reads the value of a CV within the set range and stores it in a variable 1–4.

Read CV bit (MS/FS/MN only): Reads a bit from the selected CV and stores it in a variable 1–4.

Set:

Control loco: Sets parameters for controlling the locomotive, and the type can be selected under "Control". The following control commands are available:

- **No influence:** is used to cancel all previous control commands.
- **Set drive step:** this sets a fixed drive step regardless of the currently set drive step of the loco. This can also be drive step 0.
- **Limit drive step:** this sets a maximum drive step that cannot be exceeded.
- **Scale down drive step:** reduces the internal drive step (of 255) in proportion to the set drive step, e.g.: a value of 128 reduces it to about half, a value of 64 to about a quarter.
- **Set direction:** forces a direction of travel (forward, reverse, east, west, opposite to [current] direction of travel).
- **Emergency Stop:** this initiates an emergency stop. The deceleration for the emergency stop can be set in CV #111.
- **Inhibit acceleration:** when this command is active, you can only decelerate or travel at a constant speed, but not accelerate.

Every new command under **Control loco** overrides the previous one! This also applies to Control loco commands in other scripts: if the drive level is set to =0 in one script but "No influence" is selected in another, only the last command is executed.

The internal drive step 0–255 is defined in "**Drive Step**". The final parameter defines the direction in which the control applies (in direction of travel = current direction of travel, against direction of travel = opposite to the current direction of travel, Forward = locomotive's forward direction of travel, Reverse = locomotive's reverse direction of travel, East, West). If the drive step is selected via CV, one of the script CVs #980–#1019 must be used.

Load timer: Here, a timer is set which is counted down in the background. Depending on the selected time unit, the timer counts down from the set value to zero. The units are 1ms, 10ms, 100ms, 1s, 10s and 100s for MS/FS/MN decoders. For MX decoders, only 100ms (0.1s) can be selected. If an action is

Set
Control loco
Load timer
Set diesel step
Switch output on
Set event
Set effect
Set variable
Change variable/event
Set parameter
Set PWM group
Set CV (only MS)
Set CV bit (only MS)

to be triggered after a certain time, the "Timer value" parameter is checked in "Query parameters". For MS decoders, a timer can be set to a maximum value of 255 with the unit 100s (i.e. 25,500 seconds, approximately 7 hours). Multiple timers can be set up successively with different units. However, they do not operate simultaneously, and the script queries the most recently started timer. The time value can also be selected as a script CV.

Set diesel step: Activates a fixed diesel step (value 0 = idle, value 20 = diesel step F20 [not to be confused with the F20 key]). This allows the script to take control of the diesel sequence², i.e. to fix a specific step in the sequence regardless of the actual sequence corresponding to the current speed. The decoder passes through all steps up to the selected one, and it is recommended to query this 'target' diesel step in a later query if this is relevant to the script's execution. If you deactivate the set diesel step with 'back to automatic' in one of the later states, the decoder's sequence will find its way back to the proper diesel step.

Switch output on: Switches a function output on (can also be set via Script-CV; F00f = 0, F00r = 1, F01 = 2, F02 = 3, etc.) and defines the dimming (the PWM value in %) for this output in the following parameter. Also, the function output can be switched off here, and dimming values can be selected via Script-CV or a variable. However, outputs switched on via mapping or triggered in samples cannot be switched off via a script; these would have to be managed entirely via script. The only option is to assign such an output to a PWM group in the Function Mapping (for further details, see the MX and MS Decoder manuals, chapter 'Swiss Mapping', or the manuals for ZIMO Sound Creator and ZIMO Sound Editor) and set this group to 0 via Set PWM Group. The configuration of the effects is described below.

Set event: When using this command, you should first ensure that an event is not already in use elsewhere. Events apply globally throughout the entire project, regardless of where they are triggered. For the sound samples, it is advised to take a look at the overview. In Scripts, events can primarily be used to allow multiple scripts to communicate with one another. When a specific action is reached in a script, you can set an event (1–16) with any value between 0 and 255 and query this in another script.

If you wish to trigger something with a sound, you must set event markers in the samples. To do this, select and edit the sound sample in the '**User Samples**' field in ZIMO Sound Creator or in the '**Samples**' window in ZIMO Sound Editor. In ZIMO Sound Creator, you can open the sample by right click and selecting 'Change Loop Markers' (see the [manual](#) for details); in ZIMO Sound Editor, this is only possible via 'Position', except for a sample you have added yourself, where you can set the markers during import.

Set effect: This allows you to apply effects to the selected function output. This function output does not necessarily have to be enabled via a script; the command overrides the effect set via CVs #125–#132, #159, #160, #195–#200, #205–#207.

Set variable: This allows you to set any variable 1–4 and assign a value between 0 and 255. Variables only take effect within the script in which they were set (unlike events).

Change variable/event: Adds or subtracts the value of 1 from a variable (1–4) or an event (1–16). If you wish to add or subtract more than 1, simply execute this command multiple times in succession.

² Diesel sequence: When performing acceleration or braking, the diesel engine – or, in the case of electric locomotives, the fan – passes through various stages, i.e. idle, F1, F2, etc., and transition samples (F1–F2, F2–F3, or F4–F3, F2–F1, etc.). See the Sound Creator manual for further details.

Set parameter: The following parameters are available (as of MS-FW 5.21.1 and MX-SW 40.22):

- **SET-Offset:** sets a diesel set. Value 0 = diesel set 1, value 1 = diesel set 2... (MS and MX)
- **Chuff rate CV828:** optionally uses the steam chuff rate of CV #828 (not yet implemented).
- **Cylinder valves open:** activates or deactivates 'cylinder drain' (not yet implemented).
- **Activate mute:** activates or deactivates the mute function including fade time CV #314 (MS and MX).
- **Activate pulse output:** applies the smoke fan pulse to a function output (not yet implemented).
- **Deactivate momentum (shunting key):** enables or disables acceleration deactivation according to CV #124 (currently only on MS).
- **Invert motor direction:** inverts the direction of the model's motor (only on MS/FS/MN).
- **Drive sound on/off:** regardless of the current sound status, this forces the loco sound to OFF or ON. This command can be compared to 'Fix diesel step' described above. Any intermediate diesel steps are played in sequence. (MS/FS only).
- **Invert lights direction:** Inverts the direction of all direction-dependent lights (MS/FS/MN only).
- **Invert loco direction:** Inverts the feedback of the protocol direction (MS/FS/MN only).

Set PWM group: This allows you to set or change the settings of the PWM groups in the Function Mapping. The PWM (dimming) value can be set in 3% increments and, just like in the Function Mapping, you can set up a flashing effect or suppress an effect.

Set CV (only MS/FS/MN): Sets any CV to a specified value (the permissible CV values are specified in the manual for MS decoders and must be followed!). This value can either be set as a fixed value or applied to a variable or an event.

Set CV Bit (MS/FS/MN only): Here, bits 0-7 in any CVs can be set to 0 or 1.

Miscellaneous:

Go to state: Simply sets a path to another state.

Generate random value: Generates a random value between "Value from -" and "to -", which is either defined as a fixed value in the script or executed via script CVs and can be changed later on the decoder.

The random value is applied either to a variable (1-4) or to a timer, which is started simultaneously. The time unit can be selected from a range of units (see "load timer"). Immediately afterwards, a timer value can be queried with "Query Parameter".

I²C Command: ZIMO internal test function for MX645 for output signals to I²C extensions; currently not implemented.

Miscellaneous
Go to state
Generate random value
I²C command

Example:

The easiest way to explain the use of scripts is an example (here one of the sample scripts, 'Directional whistle', which can be downloaded [here](#) from the [ZIMO Sound database](#)):

state name - state number - command - Parameter - comment
-> 0 Query parameter Sound on/off is equal to fixed value off then goto state 0
1 Query parameter Current speed is equal to fixed value 0 then goto state 0
2 Query function key If F4 is off then goto state 0
3 Query parameter Current direction of travel (0-fw, 1-rev) is equal to fixed value Backwards then goto state 8
4 Start sound Cl 66_Horn-low_mid_01.wav Volume = -9 dB (35%) Play sample once Play all if sound ends Start sound now All channels
5 Wait for end of sound
-> 6 Query function key If F4 is on then goto state 6
7 Go to state 0
-> 8 Start sound Cl 66_Horn-low_short_01.wav Volume = -9 dB (35%) Play sample once Play all if sound ends Start sound now All channels
9 Wait for end of sound
-> 10 Query function key If F4 is on then goto state 10
11 Go to state 0
12 -

How this script is "read":

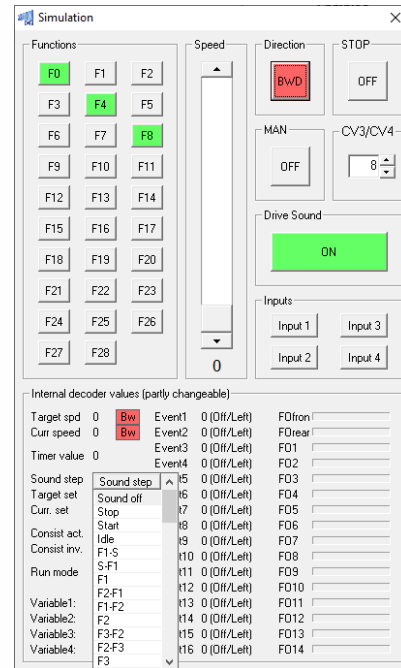
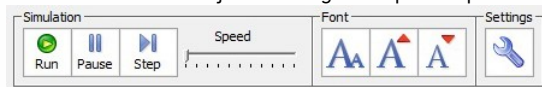
0. Is the driving sound off?
 - a. Yes?→ remain in State 0
 - b. No?→ proceed to State 1
1. Is driving speed 0?
 - a. Yes?→ go back to 0
 - b. No?→ proceed to the next State 2
2. Is function key F4 off?
 - a. Yes?→ go back to 0
 - b. No?→ proceed to State 3
3. Which direction of travel is currently set?
 - a. Reverse?→ go to State 8
 - b. Forward?→ proceed to the next State 4
4. Play the forward whistle sound.
5. Wait until the whistle has finished playing.
6. Is the F4 function key still pressed?
 - a. Yes?→ stay in State 6
 - b. No?→ proceed to the next state, and then to 0.

The query sequence for States 4-7 is repeated in States 8-11 for the whistle in reverse direction. States 4-7 could be copied like this: right click on state 4, select "copy multiple state", enter "from 4 to 7" in the following window, and insert copy of states at state 8. The query for the driving sound could also be placed immediately before each 'Start sound' command.

Scripts simulation:

Once the script has been programmed, it can be tested for errors, both in ZIMO Sound Creator and ZIMO Sound Editor. At any stage, it is possible to test and check a script by running a simulation. This is done using the **“Simulation Run/Pause/Step”** buttons at the bottom left side of the script editor window.

Currently (Sound Tools V2.0), the simulation does not work with all sequences and parameters, as ZIMO Sound Creator and ZIMO Sound Editor in some cases are unable to replicate certain functions (such as servo movements, queries for diesel steps, etc.). However, it is possible, for example, to select a diesel step yourself, or specify sets, events or variables. In general, however, any script can be tested to check whether it gets stuck at any point or runs smoothly in a loop. If the script states are executed too quickly and the steps cannot be followed or an error occurs whose cause cannot be precisely determined, it is possible to run the script step by step. To do this, click on **“Pause”** to halt the simulation. You can then use **“Step”** to go through each state individually. The speed of the simulation can also be adjusted using the **“Speed”** option.



The acceleration and deceleration behaviour can be set in the CV3/CV4 panel. If the driving sound is being queried in a script, “Drive Sound ON” or the corresponding F key can be activated in the simulation. It is possible to test Inputs 1–4, and variables, events and function outputs triggered in this script are displayed. For Target set and Cur(rent). set, you can click in the field next to it and enter the respective values.

Only the script currently opened is tested – potential interactions with other scripts cannot be tested.

Next steps once the script is working successfully:

Clicking ‘OK’ in the bottom right-hand corner saves the script and returns to the overview window. Clicking “Cancel” closes the script editor and does not save the script! All changes will be lost.

If the script does NOT work as intended:

Working with scripts quickly shows that they can become very extensive and complex. The simulation is a useful tool for error analysis. If an error cannot be identified this way, the script could be saved, a new script created, and only the faulty section recreated using as few states as possible. It can also be helpful to write down or draw the desired behaviour a script is intended to do.





ZIMO Elektronik GmbH

ZIMO ELEKTRONIK GmbH, Schönbrunner Straße 188, 1120 Vienna, Austria | www.zimo.at | Subject to change without notice.

RailCom is a trademark of Lenz GmbH; mfx is a trademark of Märklin & Cie GmbH