INSTRUCTION MANUAL

MX33, MX33FU

Preliminary issue

So far:
Pages 1-19 and pages 24-26 revised,
remaining pages taken from MX32

Work in progress

EDITION (MX32)

~~2023 01 15~~

EDITION (MX33)
2024 11 15

TABLE OF CONTENTS

Chapter page

A. The ZIMO command station and controllers 4
B. ZIMO starter sets and commissioning of the system 4
C. ZIMO data structure (**system-wide** object database **SYS DB, local** object data-bases 5
1. Technical data, help, screenshots 7
2. Switch on, first startup, emergency stop (track), switch off 9
3. LOCO IN (short: LO' IN), LOCO MOD (short: LO' MOD) 10
4. GUI Graphical **U**ser **I**nterface (GUI)-Create/modify on the controller) 11
4a. GUI - Handover & Takeover between the system centre and controllers, APPs 16
5. SOFTWARE-UPDATE, uploading ICONS, PICTURES 19
6. THE "E-SCREEN" 21
7. CONFIG Info and Settings for Command Station and Controller 22
8. Operating mode LOCO 25
9. SYSTEM-CONTROLLED CONSIST OPERATION 30
10. STOP, OFF, SHORT CIRCUIT... 33
11. External and Computer Control 34
12. „Trains" Currently not implemented! 35
13. Operating mode **SWI** 36
14. ACC LIST 42
15. StEin LIST (also control and programming of the "old" MX8, MX9 modules) 43
16. Programming in Service Mode: SERV PROG 46
17. Programming in Operational Mode: OP PROG 50
18. ObjectDB... = Object Database 54

MX32 / MX33 INSTRUCTION MANUAL Change log from 2019

2019 03 06
CONFIG: Radio control centre, Home system | CONFIG: This console, Clear data, Deleting vehicle images |
DRIVING TRACTION OPERATION: Setting up traction operation
2019 05 05
WEI: New version of the chapter
2019 07 26
ADdR ... FUMT, SPEEDO, ... : New version of the chapter
OP PROG: various corrections and new images in the sub-chapters OP PROG - A button and OP PROG - □ + A button
DRIVE ON: Reference to A button for activation (under F button)
2019 08 27
IV Small additions
STEIN LISTE HLU and east-west setting - significantly changed
2019 10 15
East-West direction: Shift-Ri first time no Ri change.
2019 11 22
Traction operation: new version
2019 11 25
SWI operating mode: Through pattern switching ...
2019 11 27
FAHR operating status; revised
2019 12 05
WEI operating status: Sample switching in displayed ZUB LIST for inserting parameters in panel fields
IV pages corrected
2019 12 17
Tractions: Tractions in control centre, and corr. in chapter "Tractions"
2020 03 15
Operating status "Switch on...." Addition via "Home system"
CONFIG - Home system
2020 05 01
Operating state WEI: Supplement SWI DEF for semi-automatic insertion of signals from StEin, the corr. in the entire chapter
2020 05 08
Addition Shift-U for transfer without window
2020 05 12
Chap. 9 E+0: Shift-U, Shift-F; Chap. 19: ZUB LISTE: TP > OP PROG; Chap. 12 GUI reference to Home System, Chap. 21 Radio operation: Radio diagnostics info,
Ch 13 LOCO: Description of the progress bar of the background reader, quality of service, description of the screen displays.
2020 05 30:
Chapter CONFIG, LOCO, ...: Add/replace images, small text corr.
2020 06 01
Chapter "DRIVE - TRACTION OPERATION", new section "Moving tractions to the system centre"
2020 10 28
Chapter "ADR ... FUMT, SPEEDO, ...), new NOTE: "GUI design on the computer with ZCS", and
Chapter "GUI ... Saving in MX10... "Loading the GUI from the computer" 3 images modified
2020 12 22
Chapter "SWI operating mode", new text and corrections in the last section (about "SWI DEF for semi-automatic insertion ...)
2021 01 14
Corr in operating state WEI, programming of switch routes
20221 06 24, 2021 07 23
Addition of photos in operating status SWI, ZUB LISTE, StEin LISTE
2022 12 23
In chapter "GUI - Transfer ...": Loading the GUI from decoder via ZIMI file transfer is REQUESTED !
2023 01 01 Chapter "ObjDB" rewritten.

From 2023 08 23 MX33: Preliminary outputs

***SOFTWARE and SOFTWARE UPDATES*:**

The **newest software version** is available on the ZIMO website www.zimo.at as free download from the "**Update & Sound**" ("Update – System") page of the same website.

Information about the **currently installed software version** of the MX33 (version number and date) is shown on the first **CONFIG** page, accessible with **E-Key + 0** (Overview & HW/SW info).

***THE SCOPE of this MANUAL*:**

The MX33 controller is part of the ZIMO digital system, consisting of the (other) important components MX10 or MX10EC and the StEin modules.

***ACCURACY of this MANUAL*:**

The screenshots and the functions described in this manual may differ from the ones in the current software version, because improvements and enhancements are ongoing.

***General information*:**

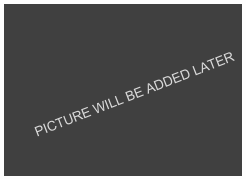
- *This manual reflects (matches) the product- and software version at delivery. The device may show further features after an update. New versions of this manual are published on the ZIMO website www.zimo.at .*
- *ZIMO devices should not be operated in extreme surroundings (heat, humidity). The air circulation into and out of the device should not be restricted (i.e. by covering).*
- *The cable connection should not be squeezed or put under physical tension or twisted. In order to have a faultless power- and data transmission, make sure that all connections have a solid fit.*
- *The device should not sty under voltage unattended. The power supply unit(s) must be disconnected from the mains using a switchable socket strip or pulling the plug(s).*
- *Children under the age of 8 years must be under supervision of an adult when operating the device.*
- *Improper use or opening of the device without consulting ZIMO may lead to danger or loss of warranty.*

In the instructions: Crossed out and not yet available display photos:

As the software of ZIMO products , including that of control panels, is often extended and corrected, display illustrations in the operating instructions often have to be replaced. For organisational reasons, however, these photos are not always available in time.

Therefore, there are markings for images that do not correspond to the current status of the instructions.

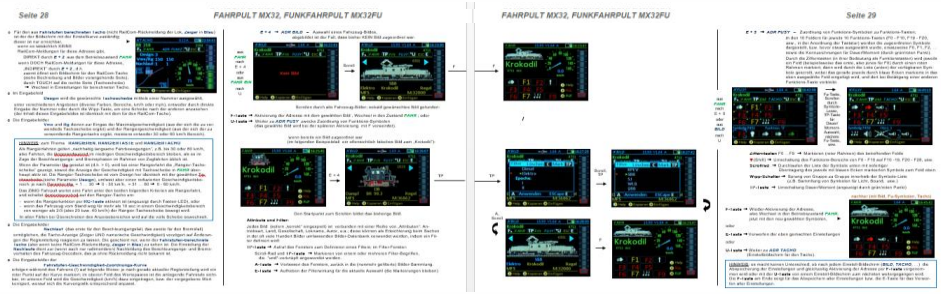
- o Dotted crossed-out image: Image does not match the context; e.g. within a sequence of images with a different locomotive name (because they come from different series of images).
- o Image crossed out in red: outdated representation
- o "Image being added": due to the fact that the software was not yet able to generate the image at the time this section of the operating instructions was written.

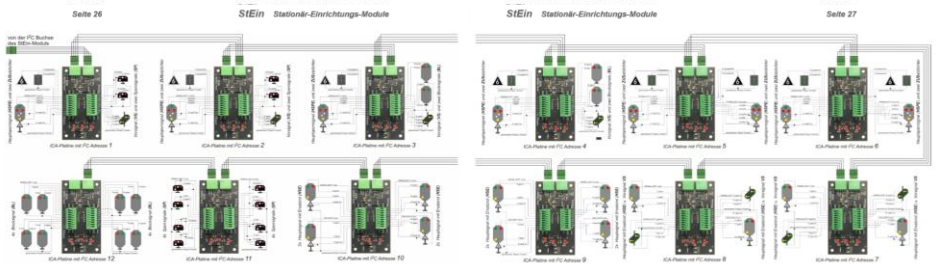


Important NOTE on PRINTING and PRINTING OUT the ZIMO instructions:

The ZIMO operating instructions are written in A4 landscape format; sometimes an even and an odd numbered pages belong together in terms of content and appearance. The printed version (or the .pdf file to be downloaded from www.zimo.at) should therefore contain two subsequent pages next to each other (even page number on the left-hand page). In Adobe Acrobat under "Page view": "Show cover page separately".

Examples (from operating instructions for MX32 control panel or StEin track section module):





A. The ZIMO command station and controllers

To put a new ZIMO system into operation :
*Chapter "A" does **not necessarily have to** be **read**. This is because the user does not need to do much more than type in an address and operate the speed controller.*

*The "Chapters A, - B" are additional information to the actual MX33 manual from "Chapter 1", helpful for **understanding the system architecture** and the ZIMO special features.*

ZIMO ELEKTRONIK has been one of the world's most important suppliers of **decoders** of all kinds for the model railroad industry and selected specialist retailers for many years. With the start of in series delivery of the MX33 in 2023 and the ZIMO app, the **digital systems** product area is once again (as in the past) gaining equal importance.

ZIMO systems comply with the **DCC standards** (according to Railcommunity and NMRA) and **can** therefore **be used with digital vehicles and respectively DCC decoders from a wide range of manufacturers**.

If the ZIMO system and ZIMO decoders used together, the **"ZIMO Specials"** become operational: in this way, the decoders are also upgraded. *) The best known of these - until 2023 - are HLU, East-West, track search, GUI transmission. New ones will be added.

*) Some "ZIMO Specials" are also adopted by other decoder manufacturers, which is by no means legally blocked by ZIMO, but rather encouraged. ZIMO is endeavouring (or in some cases has already achieved) for "ZIMO Specials" are adopted in the standards issued by the industry associations "RailCommunity" (Europe) and NMRA (North America) in order to ensure compatibility between products.

An expanded digital system (decoders are not part of the system) can consist of components:

Command station, booster
ZIMO base units MX10, MX10EC

Controller(s)
ZIMO MX33, MX33FU, MX33WF

Hand controller
Roco WLAN mouse via router

Apps on smartphones/tablets
ZIMO App, Roco Z21 **App**

Stationary equipment modules
ZIMO "StEin": for points. Signals, track sections, audio

The possible minimum configurations of ZIMO systems, which are available in the form of **starter sets**, consist of 1 ZIMO base unit (MX10 or MX10EC, + power supply unit)
1 ZIMO control panel (MX33 or MX33FU) or 1 Roco WLAN mouse,

One for all ..." (Trendy slogan in 2023) is not ZIMO strategy: ZIMO systems (in contrast to ZIMO decoders) are generally not open to third-party products: the above-mentioned operating devices from ZIMO itself and from Roco can be operated on a ZIMO central command station, but not any XPressNet device ZIMO controllers (also MX33) can be operated with Roco Z21 (with limited function). The Roco products mentioned are primarily built for Z21 but are also used with ZIMO. There may be other cases in the future.

B. ZIMO starter sets and commissioning of the system

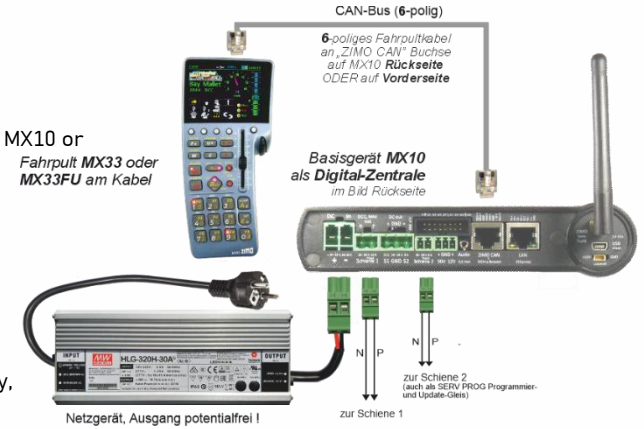
Starter sets are available in several variants, depending on the different types of
base units and control panels:

1 Central command station
MX10 or
MX10EC (Economy)

1 power supply unit NG300 (320 VA) or
NG600 (600 VA)

1 controller MX33 or
MX33FU (radio)

as well as
various plugs, CAN cable,
mains cable (depending on country,
adapter or replacement
may be necessary).



The most popular is the starter set
STARTFU
= MX10 + MX33FU + NG300 (320 VA) *ZIMO STARTFU starter set - system arrangement*

Summary "Initial commissioning" from the MX10 manual:

Other
system arrangements:
see
operating instructions
MX10

Firstly, the necessary **connections** are made using the supplied material:

- The **controller** (at one of its two sockets) via the **CAN bus cab** with **MX10** (ZIMO CAN on the front or rear)
  or with MX10EC (ZIMO CAN on the back).
- The **track system** via self-made cable,
  with **MX10** (double terminal "rail 1" or "rail 2",
  or MX10EC (double terminal "rail").
- The **power supply** with supply socket "DC in" of the **MX10**(EC)

If controller is **new** or "**empty**":
Start operation in ***LOCO IN,***
Enter the desired vehicle address
F button→ Activate, ***LOCO*** status

If controller is **not "empty"** (address from last available)
Start operation immediately in ***LOCO*** or *WEI* state
The displayed vehicle can be driven (slide control, direction button, function buttons, ...).

See chapter 2. Switching on, etc.

C. ZIMO data structure (**system-wide** object database **SYS DB, local** object databases **LOCO DB** and **APP DB**, **loco recall LoR**) **& transmission cycle**

*Similar to chapter "A": chapter "B" does **not necessarily or immediately have to be read.***

However, the ZIMO digital system is designed for both small and large applications. And with around 100 or 200 (or more) digitised vehicles (locomotives and non-traction vehicles) and multiple control devices, a distributed data structure and a complex priority logic is necessary to optimally distribute the available bandwidth on the track.

*The **present chapter** attempts to explain the context, some terms and structures; it is in itself **not a user manual**, with specific steps mentioned only by way of example.*

The ZIMO system is based on **distributed intelligence** and **distributed databases** (digital centre and control units such as controllers and apps). The data (GUI's - Grahical User Interfaces, and current driving data) of the vehicles, trains and accessories are stored centrally and locally. On the one hand, this results in an 'implicit internal data backup' and, on the other hand, it enables you to 'take your data with you' when you transfer your trains to another ZIMO system (layout) using a control device (ZIMO controller, ZIMO App).

*The **system-wide** object database **SYS DB** in the ZIMO command station (= digital centre)*: (currently) located in MX10 or MX10EC; can be seen and edited on the control devices (ZIMO controllers and ZIMO apps); the entries in the ***system-wide object database*** are created...
-1- on **ZIMO control devices** (controllers and apps) when objects are created (see section **'Local...'**),
-2- in the **external sources** (computer programs such as ZCS,..), where objects are also created,
-3- directly from **decoders that** register with the system (according to RCN-218) and transmit their own GUIs.

*Screen **'SYS DB'** on the **controller MX32/MX33** shows the entire system-wide object database SYS DB (i.e. all addresses in the digital centre), with the objects available locally in the controller in **blue/green** line colour, and thus indirectly the **LOCO DB**.*

Green lines** – in own **LOCO DB**, and **LoR
Blue** lines – in own **LOCO DB**, not **LoR
***Turqoise** lines – in **LoR**, and in traction (T1, ...)*

*Grey lines – the other lines, i.e. in systemwide object-database SYS DB (and possibly in **LOCO DBs** and **LoRs** of other devices) but not in your own **LOCO DB** und **LoR**,*

*Further **different** coloured lines as part of the inventory search (registr. accord. RCN-218.)*

*Screen **"SYS DB"** of the **ZIMO App** (Screenshot from August 2024 version):*

*Here too (as in the controller), all addresses of **SYS DB** (i.e. the objects contained in the digital centre) are displayed, but those that are also available for direct use in **APP DB** are displayed in **blue/green** (green=**LoR**).*

*On the screen **"SYS DB"** the addresses for the local **APP DB** are selected: slider, IMPORT*

*Furthermore, the display **"SYS DB"** on the app is particularly useful for viewing and deleting unnecessary entries in the SYS DB in the control centre.*

BACKGROUNDINFORMATION

Local object databases LOCO DB or APP DB in the operator terminals:

Objects of the **LOCO DB** can be seen in the ZIMO MX32/MX33 controllers within the **"SYS DB"** as **blue, green, turquoise** lines, there is no separate window for it. In the ZIMO app, however, there is a separate window **APP DB** for the counterpart, the **"APP DB"**.

*Objects in the local database, such as **LOCO DB** or **APP DB**, are a **subset of** the objects in the **system-wide object database SYS DB**. Objects created locally are automatically accessible system-wide.*
*Objects in **LoR** are in turn a **subset of** the objects in your own local **LOCO DB** or **APP DB**.*

-1- Entries made by typing in a new address and **activating** it in the 'LOCO IN' window also apply to the local **LoR**, the local **LOCO DB** and the **system-wide object database SYS DB**.

-2- Activating a grey line in the **SYS DB** window, i.e. an address that was not previously available locally, creates an entry in both the local **LoR** and the local **LOCO DB** at the same time.

0-3- Self-registering decoders (according to RCN-218) also generate entries.

*In the window **'LOCO IN'** on the **MX32/MX33 controller**, a (new) vehicle address is entered and – if desired – given a name. Further GUI elements can also be activated using F-keys (opening in the screen **LOCO**)*

FAHR

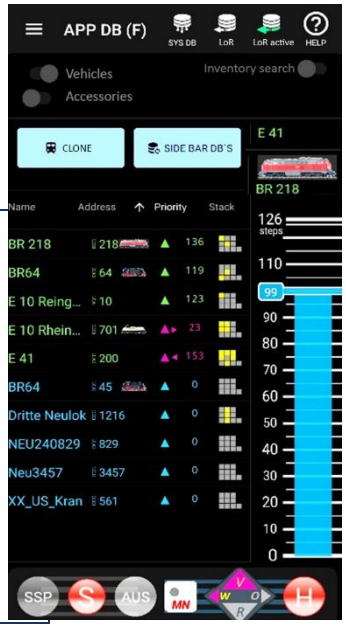
*By activating it (using the A button from the SYS DB, grey line), the screen **LOCO** is opened for the address, and it is also entered into the **LOCO DB** and **LoR** i.e. green line in the screen **'SYS DB'***

DON'T FORGET TO DELETE!

Databases should occasionally be cleared of 'junk' data (addresses that are no longer needed or were entered by mistake, etc.).

DON'T FORGET TO MAKE A BACKUP!

*A certain degree of **system-internal security** is already provided by storing the contents of the local databases in the system-wide database of the ZIMO command station. **Nevertheless,** it is recommended to make copies of the **system-wide** and **local object databases** on USB sticks or on the computer. This is partly done automatically when the software of the controllers is updated.*



Screen **'APP DB'** of the **ZIMO App**

The **APP DB** is the equivalent of the **LOCO DB** of the controller, but here as a separate window **'APP DB'**.

The addresses contained in the **APP DB** have been determined by IMPORTING from the SYS DB (see left).

*From here, you can click on an address (name) to switch directly to the **LOCO** screen.*

In the 'SIDE BAR' on the right (when called up), database manipulations can be carried out: take over, hand over, delete, ...

loco recall = LoR (favourite lists for vehicles) in the operating devices:

'LoR' is a traditional term; the expression was created at a time when the rapid access to important objects was not yet a matter of course. It was about 'retrieving' vehicles that were previously actively controlled (also referred to as 'in the foreground').

The vehicles enter the *LoR* of a control device (ZIMO controller or ZIMO app):

-1- by activating the address by typing it into **LOCO IN** (optionally with GUI creation),

-2- from the **LOCO DB** or **APP DB** by activation or conversion (if already listed there).

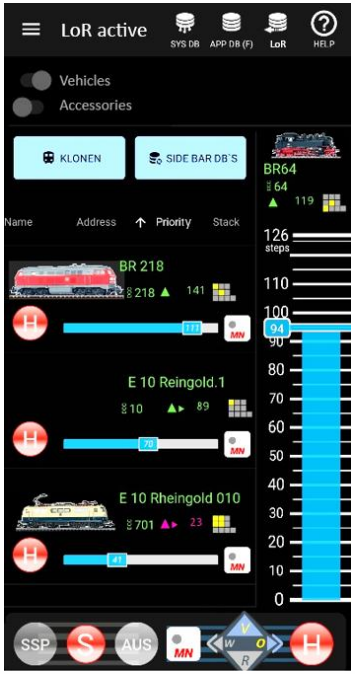
The *LoR* is used to quickly find vehicles among the favourites and then activate them (A button) or just observe the recorded driving and function data (controlled from other control devices). The *LoR* also serves as a platform for forming multiple tractions, among other things.

The content of the *LoR* is always a subset of the **LOCO DB** or **APP DB** because every address in the *LoR* is also part of the **LOCO DB**, but not every address in the **LOCO DB** belongs to the *LoR*.

Usually, the LoR is displayed in the lower half of the screen, while the upper part is still available for the active vehicle, as well as the side bar.
A LoR on the full screen is also possible (by touching the area).

REMEMBER: REMOVE from LoR:
A maximum of 128 addresses (vehicles) can be included in the LoR of a control unit; however, you should not let it reach this number; this would not be beneficial for the overview and system performance. The latter because it would also affect the transmission cycle.

Vehicles that are no longer needed can be removed individually from the LoR using the C key (they remain in the LOCO DB, but in a different colour). The maximum number can (and should) be set in CONFIG from 128 to a lower value.



Screen shot 'LoR active' of the ZIMO app (screenshot from August 2024 version): In addition, there is also a normal list view of the LoR (as on the controller).

But in 'LoR active', it is possible to control up to five vehicles at the same time (more by scrolling): a separate (horizontal, simple) slide controller, stop or direction button for each address – the latter only when stationary – and an enlargeable miniature control panel for functions.

The display "LoR active is also used to form (and dissolve) double and multiple traction.

BACKGROUND INFORMATION

The 'emission cycle' – a brief description:

The decoders (in vehicles, points, signals and other accessories) must be supplied with **operating data** (speed, functions, switching states, etc.) and, if necessary, with **configuration data** (CVs). Control devices (controllers, handheld controllers, apps), computers and other system components determine the data content; changes (speed changes, switching commands, programming commands, etc.) must reach the decoders with priority, i.e. particularly **promptly and reliably.**

A locomotive typically requires 3 to 5 bytes for its quickly changing loco and function data and several hundred to several thousand bytes for its configuration data.

In addition to the rapid transmission of modified operating data, constant **operating data** must also be **continuously refreshed** because it is lost in the historically developed method of operation of digital controllers in the event of power interruptions (at least in older decoders). This does not apply to the configuration data, because this has always been stored in non-volatile memory in decoders.

On a layout where only, a few trains are in motion at the same time, sending data to the decoders is easy to handle; in busier operation (and when there is multiple active control, i.e. the data changes frequently), data backlogs can occur. To avoid poor response times in general and, in particular, delays when stopping, ZIMO uses **a sophisticated priority scheme** to get the data onto the track and into the vehicles.

- The **highest priority** is given to **changes to operating data**, generally and without distinction between speed and functions, if they are caused directly by a control unit, an app, a handheld controller or a PC controller. Also given the highest priority are **CV programming and reading commands**, as well as **switching commands to accessory decoders** (but only 5 times or shortened by RailCom response).

  Linked packages (which contain data associated with one of the above cases) can also be included in the 'highest priority'; this may include, for example, a programming command to shorten the braking time, which precedes an actual braking command.

- In '**normal priority**' (as opposed to 'highest priority'), all other data packets belonging to active addresses are processed; however, the frequency and order of transmission depends on several circumstances and the type of packet; see below 'increased normal priority',...

  Within the 'normal priority', there are weighting factors ('level') that determine the relative frequency of transmissions for the active addresses:

  = 1 (referred to above as 'increased normal priority') - all packet types for an address in the foreground of a controller or the ZIMO app (or in traction with one), or temporarily after power-on (switching on, short circuit) for each address for a complete packet cycle; or on special request ('quasi-foreground' to enable more RailCom feedback). For example, external computer programs can request level 1 for 10 seconds.

  = 2: Address in the background (LoR) of a controller or ZIMO app (or in LoR of several at the same time)
  = 5: Address not in any controller, but according to the database of the control centre, the vehicle is in driving mode.
  = 6: Addresses of accessory in general
  = 10: Address not in any controller, according to the database at the control centre not in driving

The **frequency distribution** of parcel shipments at the 'normal priority' levels follows a specific pattern that depends on the number of addresses at the levels. Foreground and background:
Basic principle: Level 1: 20% (1 address at the level) to 50% (>3 addresses), Level 2... 10: 10% to 30%.

Home and non-home system

Normally, the currently used central unit should be entered as the home system in the CONFIG data of the MX33. The exception to this is when the device is used as a 'guest device' on a different system; in this case, new addresses and GUI changes should not be transferred to the current central unit!

Please refer to *chapter CONFIG*

1. Technical data, help, screenshots

MX33 controller, MX33FU radio controller

Current consumption (with 20 - 30 V on the CAN bus) ..... 70 mA to (while recharging battery) 150mA
Battery in radio MX33FU..... 2200 mAh
Battery runtime in the radio MX33..... 15 min
Rechargeable battery in radio MX33FU..... approx. 5 h
Dimensions front - back x front - back x T..... 175 x 80 x 20-40 mm
Weight MX33 / MX33FU (with battery) ..... 200 g / 250 g

RailCom is a trademark of Lenz Elektronik GmbH

The control panel in driving mode

*DCC / RailCom statistics: Count per sec DCC packets / RailCom responses to them, CV background reader - progress ***)*

"Display Header" *(the header on the screen):*
*Operating status, here **FAHR (LOCO),***
*RailCom display (not related to current address)**)*
Current voltage & Power on track1 (MX10),
"Communication dot) for visualization of data traffic with the central unit (MX10),*
Battery status,Clock (world or model railroad time depending on the situation).

Loco picture *shown here in small format,*
Loco name and address *(including data format):*
Touch on image → large display (full width),
touch on image→ larger font for name,
(graphic display of speedometer is omitted in each case).

Speedometer *here in graphical representation with magenta needle to display the real speed from RailCom feedback (Attention: Adjustment in **GUI SPEEDO**, necessary, i.e. **E + 2**, if not done by automatic GUI loading), or with "blue needle" calculated from speed level (adjustment also in **GUI SPEEDO**).*

*Touch on speedometer → Graphic display is omitted, numerical speed remains, large vehicle image, tableau of functions remains (very popular display of the **LOCO** screen!)*

Functions tableau*, shown here F0 ... F9 in arrangement of the numeric keys. They describe their current meaning and can be operated either by key or touch, duration or momentary.*
↑→Switching to function key levels F10 ... F19, F20 ... F29, etc.
Note: Grouping with theme buttons planned!

Menu and softkeys *M, I, II, II*
The current meaning is shown above the display.

Speed step indicator *on the display (linked with slider):*
Shows, among other things, current speed step, takeover positions, train control.

Numeric and function keys, A *(yes, on, ...) - and **E** (no, end, ESC, off, ..) usable also as SMS keypad for text input (names, etc.)*

X →*Auxiliary button for the scroll wheel, e.g. scroll alignment*

Scroll *in **LOCO** → Fine speed control (+/- 10 steps), locomotive selection, or control for parameters (such as volume).*
*in **OP PROG**, in **SERV PROG*** Change between lines, etc.

The "small" keyblock:

F → *(= Fahr) Change from **LOCO IN** (new address or new vehicle) to **LOCO**, or change between vehicles within **LOCO***
U → *(Übernahme= takeover) Change between vehicles within **LOCO** (other direction than F), takeover of a vehicle from another controller*
TP →*(= traction) Switching between consist locos, or adding or removing locos from a consist*
W → *Switching in and out of the operating mode **SWI***
C → *Clear /* delete characters, vehicles from the ***LoR***, and more.
↑→ *Shift key, various functions, e.g. switching function levels*

Speed step controller: does not control the *speed* directly, *but usually 0 ... 126 steps, adapted to decoder 0 ... 28 or 0 ... 14.*

S → *Collective stop; together with **H** Power off (normal assignment)*
Row → *East-West direction (system-related) Switch & display*
H → *(= Halt) Emergency stop*
RVR → *(Richtung Vorwärts-Rückwärts) Forward-Backwards; switching & display*
MN → *(= manual) HLU and ABC train control Cancel*
RG →*(Rangier) Shunting, Slow speeds*

These control displays are not yet implemented as described.

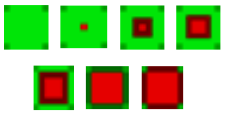
)* The **"communication dot" indicates:
1) how quickly the ZIMO command station (usually ZIMO product) reacts to requests from the controller by lighting up the BLUE SQUARE when, for example, a function button is pressed, and turning it off again when confirmation from the command station is received. The communications dot therefore flashes only briefly; a prolonged illumination indicates a slow transmission (overloaded CAN bus, electrically limited or a poor radio link). A continuously lit dot means that the controller commands were not executed by the command station.
2) when communication errors occur: the EMPTY SQUARE indicates by brief illumination single disturbed transmission processes (which are corrected automatically); continuous illumination means loss of connection.

***)* The **RailCom Logo** in the display header indicates the following:
RailCom messages are sent by the decoder to the command station, where they are processed and passed on to the controller. The flashing rate indicates the percentage of answered DCC data packets:

1:10 (i.e. the RailCom logo flashes briefly) = 1 to 5 % of the sent DCC packets were answered.
1:5 (i.e. the RailCom logo flashes a bit faster) = > 5% to 10%
1:3 = > 10 % to 20 % 3:3 = > 20% to 50%
3:1 = > 50 % to 80 % 5:1 = > 80 % to 90 % permanent = > 90 %

****)* **DCC & RailCom statistics:** Number of DCC packets sent to this address vs. RailCom responses to these packets indicator of transmission/reception and feedback quality.
AND
Progress bar of the CV background reader (the CVs of the decoder with this address are automatically read out in the background so that they are available without further waiting time if required (e.g. for configuration processes such as "Swiss mapping")).

*****)* The **"Quality of Service"** symbol shows the quality of the messages received from the decoder via RailCom. This is a percentage indicator of how many DCC packets arrived without damage from the decoder. All packets are used in this equation not just the active address– thereby, obtaining a meaningful value as about 80 or 90 events within one second are used as basis.



"Quality of Service" indicators

Important:

A (10 sec) → RESET MX33

This reset option is used sometimes in the event of "software crashes" due to software "bugs".

The MX33 HELP-System

Softkey I → is reserved for calling up the help screen in most operating modes. If this Softkey is pressed, it opens the proper help screen for the current situation.

Each displayed help file is scrollable, if there is more content than the screen can hold. The controller remains operational in driving and switching modes (***LOCO*** And ***SWI***),) while the help file is displayed; but not in the programming modes, which would make little sense without the programming screen present.

Softkey I → closes the help screen and returns to the previous operating mode.
Softkey II, III → in some cases these Softkeys are used to switch directly to other help files.

The help files shown in this manual are part of the explanations!

NOTE: The help system is currently under development and will be continuously updated.

HELP-EXAMPLES
(Nov. 2024) HELP screens for English are a work in progress; set to be released 2025

Screenshots from the MX33

Do not insert the USB stick yet! Take screenshots first, which are automatically saved internally.

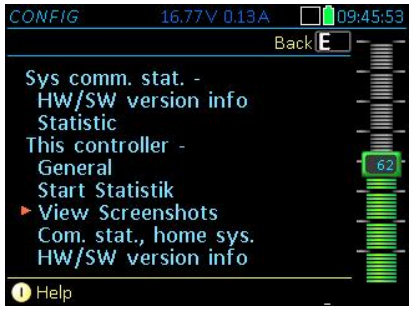
Shift (forward, hold) + **X key** → screenshot is taken and saved internally; you can take (almost) as many screenshots as you like in succession, they are savedinternally and remain there until you copy them to a USB stick; see below.

Display at the top right (instead of the clock for 1 sec) to confirm the recording and indicate the current number for the freshly stored screenshot. ▶

(example)

▼ automatically taken screenshot 'start statistics'

E-key + **0**-key → Enter CONFIG (see CONFIG chapter)
▼ here to view the internally stored screenshots



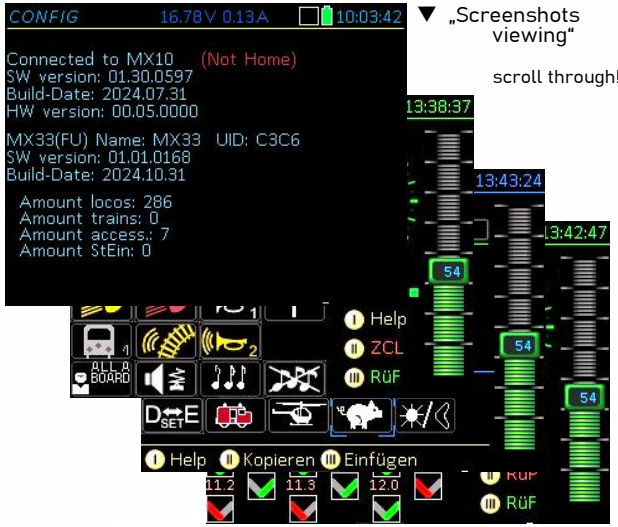
It is not necessary to view the screenhots before saving them on the USB-drive ▲ - ▶ ▼

Insert the **USB-stick** into the **MX33**
→ („forward") **A-key** → menu
Scroll → „save screenshots"

A → Screenshots get copied to usb drive
▼ (new Directory);
Prompt to remove the USB drive.

▼ „Screenshots viewing"

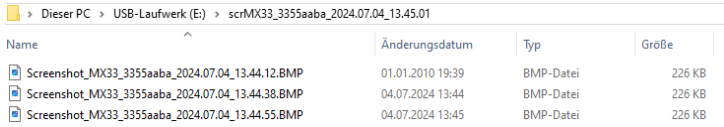
scroll through!



A →

Message that saving is complete, remove USB stick, E (reset)

Connect the **USB stick** to a **computer** open Directory „scrMX33...", Screenshots (.bmp) can be found here.

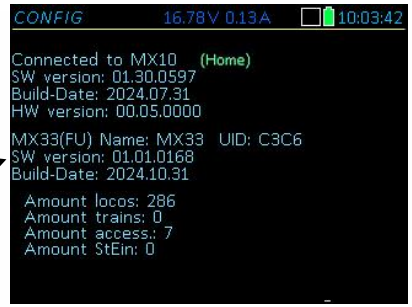


2. Switch on, first startup, emergency stop (track), switch off

More details in the next chapters!

Switching on the system:
Connect the power supply unit to the command station. The controller can be connected before or after, even during operation, via the controller cable (= CAN bus). Controller in **radio** operation: see next chapter!

start-up sequence:
Starting up the command station: **approx. 15 sec:** controller dark, controller starting up **approx. 10 sec:** the protocol of the activities currently being carried out can be seen on the screen, which usually does not need to be observed.



Controller starts with...

... if controller is not a '**blank**' *) (normal case):
LOCO or SWI screen as at last power-off, possibly with a turnout panel in addition.
Controller is **immediately ready to drive and switch**.
if, however, a **different vehicle** is to be driven than the one that was last active:
A-button → switch to LOCO IN screen to enter the desired address (vehicle).

... if controller '**empty**' *) (for example, **newly made**),
LOCO IN screen for entering a vehicle: ADDR: (digits, A),
NAME: (SMS buttons, A),
fill in, possibly drive levels (scroll), **F** → **activate**

Or: just type the first few characters of the address and/or name; the object list (lower window) lists the corresponding vehicles from the *system-wide object database*. Softkey II à Switch to object list, Scroll → Select, A → Apply, **F** → **Activate**

*) 'empty' means that the last time the controller was used, the local LOCO DB was completely deleted (including the vehicle in the foreground), so there is no address to reactivate. This is usually the case with new devices. Nevertheless, valid vehicles (including GUI data) may of course be present in the system-wide object database (in the MX10), valid vehicles (including GUI data) may be present, including those deleted from the local LOCO DB.

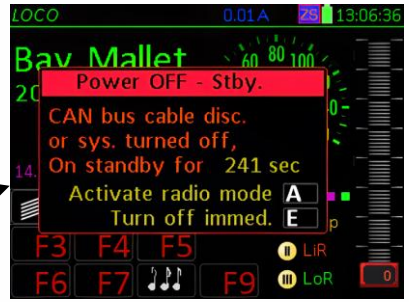
EMERGENCY STOP, and re-start:

H-key **+ S**-key (simultaneously) → **„track 1" OFF**.
In addition to this type of EMERGENCY STOP, the ZIMO system offers numerous other emergency stop variants and settings in the controller menu (via E+0), including application to output 'Track 1' or 'Track 2' or both simultaneously, collective stop or power-off.
See chapter 'STOP, OFF, SHORT CIRCUIT'.
↑ (Shift, hold) **+ S**-key → „track-1" ON

Switching off the system:

Disconnect the plug of the power supply unit from the command station (or the power switch): all devices will turn off.

whereby controller MX32/MX33 include batteries (including non-radio versions); they switch off 'artificially'; however, if there is no data connection, the same screen appears as for individual disconnection:



In wired operation: **Disconnecting the CAN bus cable** (but keeping the command station supplied with power): The controller will initially run in standby mode for a few mins.

E-key → Switch off controller immediately
A-key → (Radio version only) Radio operation on

in radio mode: **E** (in advance) **+ A** (simultaneously) → Switch off the radio controller

Notes on radio operation (MX33FU)

The MX33FU (MX33 with integrated radio and antenna) can be used **either in cable mode** or in **radio mode**. In cable mode, it behaves the same as a non-radio MX33 controller. In cable mode, the battery is charged at the same time: due to the additional charging current, the maximum length of the CAN bus cable may be reduced, but this is only noticeable in rare cases.

I M P O R T A N T: The remote controler must be **registered in** the system (command station)!
Connect the radio controller to the command station using a 6-pole CAN bus cable, **briefly start up** the controller in **cable operation**. The system code number (= identification number of the control centre) is automatically (imperceptibly) transferred and stored in the controller.

Starting the **radio operation** (after the controller has already been registered):

... from the **cable operation :** Pull out the CAN bus cable; for the 'Power off – Standby' window, see above!
A-key → Switch to radio operation

... of the **switched-off** controller: **A**-key **(long)** → Starting up the controller in radio operation

Monitoring and **continuous control displays** regarding the quality of the radio connection:

- **Received signal strength** indicated by the antenna symbol (with bar graph, green... red) on the selected radio channel. The informative value is limited because external networks are also included in the calculation; this is particularly noticeable when several ZIMO systems are in the vicinity (for example, at exhibitions...).
- **Reception number** of messages received from the command station – this transmits 20/sec:
  20, 19 (occasionally 18 or 17): optimal connection
  < 18 (mostly occasionally < 17, 16, ...): critical, the smaller the worse; with good field strength: possible interference from other networks (e.g. WLAN). Possible remedy: **change radio channel!**
- **Radio diagnosis information** in the green line (not necessary for the 'normal' user to observe): the colour sections are used to make radio transmission and reception processes visible.
- **Communication point** indicates delays or loss of radio communication. If it lights up with every change to the MX33, this means a good connection; otherwise, a poorer connection. **Red** when there is interference for 200 ms.
- **Flashing function icons** or **controller knob** when transmission of a new state is delayed (where it is not possible to recognise whether 'only' the feedback is missing, but the execution is working)
- Various notes and **warnings regarding radio restrictions** and battery **charge level** (such as "battery charge below...%)

Restrictions in radio operation: data-intensive applications such as 'StEin LISTE not available!

3. LOCO IN (short: LO' IN), LOCO MOD (short: LO' MOD)

*Address, name, picture, other GUI elements. **IN**put (for new vehicle)*

*or Name, possibly an image, possibly other GUI elements **MOD**ify (for currently active address)*

and on-track search (see next page) = unknown address of a newly on-track or for a short period of time with the vehicle tilted.

*and stock search (see + 2 pages) = Add the system-wide database (the 'stock') with newly registered vehicles (after **RCN-218**)*

LOCO IN can be accessed from several operating states, in particular **LOCO** and **SWI**

A-key → ***LOCO IN*** or if A-key is not conclusive (e.g. in LoR) ↑ + **A** → ***LOCO IN***

Softkey III (LoR)

W (WEI)

Correct marked text during input (e.g. name)
C and each **additional C:** delete text character by character.

A

↑ + **A**

← (shift) ahead because A is used differently in the LoR.

A

The **'normal' input process** for a new vehicle: **see right!**

The **system-wide object database . . .**

is displayed in the **lower half of the screen** and can be scrolled through; it is the same content that can otherwise be seen in the full screen display in ***'SYS DB'*** (after E + 6), i.e. the vehicles already known in the system: all entries in the local device (***LOCO DB*: green, blue, *turquoise*** lines), as well as all entries in the command station (***SYS DB*: grey lines**). When you start typing characters (letters or numbers) into the 'ADDR' and 'NAME' fields, the database is **filtered to** only show the results that include those initial characters.

SK II → Switch cursor in database and back

Scroll → Select a row from the database

A → Transfer data to the fields above

F → Activating the selected address. This enables fast activation and at the same time control of whether an address is already occupied in the system ('FS', ...).

*Address, name, picture, **IN**put other GUI elements of a new vehicle.*

number keys → Enter an **address** in the 'ADDR' field:" (this is the only mandatory entry!)

A or **Scroll** → Switch to the next field: 'Name'

Numeric keys (in SMS mode) → 'Enter **name**

A or **Scroll** → Switch to the next field: 'Group'

X-key → Select/enter **group name** from list

When needed: **Scroll** → List of digital **formats**

If desired (not necessary, because it can also be done later) GUI creation right within **LOCO IN**:

U -key → Go to **1st** GUI element (**PICT**)

Possible at any time (as soon the address has been typed in):

F → **Activate** and switch to **operating mode *LOCO***

Vehicles with entered data come to the foreground, e.g. only with an address but without a name or picture, or with an address and name but without a picture, etc.

Whether the vehicle actually exists (i.e. whether feedback is received from the decoder) is irrelevant for the process; this can only be seen in **LOCO**, among other things, on the magenta speedometer needle.

If you activate (F key) an address that already exists in the **LOCO DB** or **SYS DB**, the name, picture, functions, etc. are copied from there automatically.

F

*Name, possibly a picture, pos. further GUI elements of an active address **MOD** (**Mod**ify):*

LOCO MOD is a **variation** of **LOCO IN**: no new address is entered, but the currently active address is automatically entered in the 'ADDR' field (at the second 'A').

A + A (alternativly ↑ + A + A) → ***LOCO MOD***

The 'Name' field is ready for modifications immediately

A + A

↑ + A + A

Modifying newly selected text (e.g. name)

1) **Without C**, Enter the new, replace the old text.
2) **1 x C**, enter additional text to current text.
3) **2 x C** and **further C**s delete the text character by character.

On-Track search, formerly "ZIMO tilt search" *)

requires ZIMO decoder SW from 37.16, MS decoder from the beginning

*)Due to standardisation by "Railcommunity" (= VHDM, association of manufacturers of digital model railways) at ZIMO's suggestion, the original "ZIMO Tilt Search" is officially called "Rerail Search".

The *on-Track search* is a procedure for determining the previously unknown DCC address of a vehicle with a ZIMO decoder (with a suitable SW version, see above), possibly also other brands, on the layout.

- **A-Key** (or, ↑ **+ A- Key** with ***LoR*** open)
  (it is recommended - although not necessary - to initiate the change to ***LOCO IN*** before tilting/rerailing so that only the TP button needs to be pressed after tilting/rerailing)

Tipping/on-Tracking: Disconnect the vehicle with the address to be determined for **at least 2 seconds** by **tipping off** ("tipping search") or **Track-on** a new vehicle.

-TP (SEARCH) → System starts search for addresses of recently **(**last **30 sec**) new vehicle on the track; all decoders that were placed on the track within the last minute (= power-up) are prompted to send a message via Railcom.

The decoders that have reported are displayed; the **first** (often the only one) in the **report window**, **all** (including the first) in the **on-Track list** (bottom half of the screen). If the addresses are already known from the ***ObjDB*** (object database), the name and driving data are also displayed.

An important additional piece of information (also reported by the decoder) is a message such as "On track 4s ago", i.e. the time that has elapsed since the message. Each line in the on-Track list is also provided with a reporting time. This can be used to estimate which address is probably the correct one.

F → **Activation** of the vehicle in the **message window**
(from text "Addr. xxx found")
and change to **operating mode *LOCO***

If a vehicle from the **rerailing list is** to be activated instead of the vehicle from the message window:

SK II→ Cursor- Change to on-Track list and back

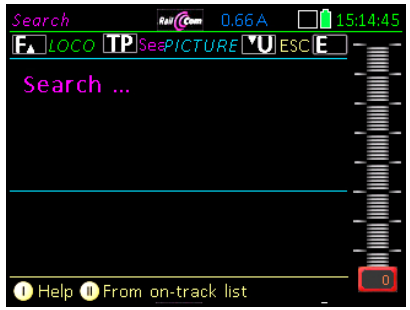
Scroll → Select a line in the on-Track list

F → **Activate** the selected address

or

- **Scroll** and **A** → Select another address from the "Rerail list"; the selected address is placed in the message window as "found".
- **F** → Drive with the selected address.

↓ A, TP



New address found:

NOTE: The *on-Track search* has nothing to do with the *stock search,* which largely corresponds to the first part of the automatic registration according to RCN-218 of the RailCommunity.

*Inventory search – registration according to RCN-218 (DCC-A) *) ZIMO type*

With MX10 and MX33 from software version 01.30.0700 | for RCN-218 capable decoders; ZIMO decoders (MS, MN) from 4.250

*The **RailCommunity standard RCN-218** (also an NMRA standard in the works) for **'automatic registration' (DCC-A)** defines a protocol, i.e. additional DCC commands and RailCom messages for this very purpose. This enables interoperability between digital systems and decoders from different manufacturers in this aspect as well.*

'Registration' refers to a procedure that is designed to fulfil two basic tasks:

1) Registering the vehicles and accessory items found in the system (in particular, newly acquired items) in the digital system, including any re-addressing (to a requested address if it differs from the currently programmed address or, if necessary, due to an address collision) of the decoders in these vehicles and items;

***2)** the **transfer** into the digital system of the **GUI information stored in the decoders,** which is needed for the graphical representation of your vehicles and accessory items (pictures, name, function symbols, etc.), as well as other decoder data in the vehicles, e.g. CV values.*

*The **ZIMO decoders** (from the MS and MN series in certain software versions, see above) can handle the DCC additional protocol for registration according to RCN-218 with corresponding standard-compliant control centres, as can the **ZIMO system products** (from MX10 with the above-mentioned software version) and control units with regard to the registration of compliant decoders.*

*The **'ZIMO inventory search'** corresponds to the first part of the complete RCN-218, i.e. (see above) **1) registration...**; the second part, i.e. **2) . GUI information. is** carried out for ZIMO decoders by the independent 'ZIMO File Transmission', see the chapter 'GUI Graphical User Interface' or, in accordance with RCN-218, will be added later if necessary.*

*The **'ZIMO inventory search'** bears this name because each application is based on the existing 'inventory', namely the decoders of the vehicles and accessory items that are currently in the **system-wide object database** (in short, **'system database'**). The inventory search is designed to **reorganise** this database, and normally not fully automatically, but with the help of the user (who calls it up, monitors it and controls it manually). Among other things, new decoders are to be integrated, address conflicts resolved and (last but not least) redundant addresses deleted.*

*The **'ZIMO inventory search'** considers the practices of typical operation with ZIMO digital systems: therefore, unlike the original design, there are no mandatory registration processes for all vehicles and accessory items, no continuous searches for new objects, and no automatic address selection when re-addressing, but only when requested by the initial initiator of the inventory search.*

*The **'ZIMO inventory search'** uses the 'normal' elements of the RCN-218 registration protocol but is usually started by the user (possible at any time, not just at the start of operation) from any control device. When a system that has not been modified, or only slightly modified, is put back into operation, this is often not necessary at all. ...*

Simplified technical description of the registration process according to RCN-218 or inventory search:

The digital centre (ZIMO MX10 command station) periodically (density depending on the situation) sends out LOGON_ENABLE commands; in response, the decoders (vehicles, accessory decoders) respond with RailCom (Channel 1) ID15 messages (these are the actual logons). This is done using a statistical method to avoid simultaneous responses that would cancel each other out.

*An ID15 message contains the **DID** (decoder unique ID, 12-bit manufacturer code, 32-bit manufacturer serial number);*

*The SELECT command prompts the decoder to transmit the 'Short Info' data space defined in RCN-218. The following RailCom short info message contains the **desired address** of the decoder;*

*with knowledge of **DID** and **desired address** of a reported decoder, the system database is now synchronised.*

*Stock search can be accessed from the operating states **LOCO** and **SWI** (a. o.)*

E-key + 6 → Switch to window **'SYS DB'**, i.e. to the system-wide object database; this shows **green, blue,** turquoise rows (contents of **CTRL DB**), and grey rows (**SYS DB**), i.e. all addresses currently registered in the system.

The individual lines represent both 'real vehicles' and addresses without a real counterpart; some intentional, often forgotten from previous activations. The latter should possibly be deleted, which can be done within the 'inventory search'. The 'real vehicles' are often quickly recognisable by the magenta-coloured direction and speed indicators (RailCom feedback), but not all decoders provide feedback.

TP-key → *Start inventory search* The area on the screen for messages is being cleared.

E+6

*The **system-wide object database** In the **'SYS DB'** window (after E + 6), i.e. the vehicles already known in the system: all entries in your own device (**CTRL DB: green, blue, turquoise** lines), as well as all those that only occur in the command station (SYS DB: grey lines).*

In this illustration or section, address 200 shows a 'RailCom response' (magenta directional arrow and speed); at least this address corresponds to a 'real decoder'.

TP →

An area on the screen is being prepared for future information from the registered decoders and for editing the decoder and system database.

Immediately after starting the inventory search, some settings can be made here; these will be programmed into the decoder when it is registered.

The system-related address stack

Optional feature in decoders: during redirections in the search for the position, entries are made in a list (in the 'address stack') that each contain the current CID (ID of the central unit) and the corresponding address of the decoder.

To use this feature effectively, it is useful to define some settings on the control unit. These are then forwarded to all decoders as part of the inventory search.

The settings are made in the first stock search report window after starting (TP button) the stock search.

***1-key** → in the case of a redirect as part of the stock search, the new (desired) address should be written in the **address stack** of the decoder under CID of the control centre and*
In the event of a system change, the registration with the CID of the control centre should be searched for and used as the desired address of the decoder.

***2-key** → Alternatively, if no address after **1**, the delivery address should apply.*

***TP-key** → Apply! or, (if you are in the Settings section) use the last valid settings again.*

the registration process according to RCN-218

with extension for 'address stack' settings

*LOGON_ENABLE – ID15 (contains **DID**) – SELECT (with „Read ShortInfo" and „Pseudo-CV-query" to set the preferences) – Shortinfo (contains the **desired address**, already determined based on the settings) – LOGON_ASSIGN (with **desired address**).*

In the course of the RCN-218 procedure, all decoders that support the relevant protocol report by *RailCom messages ('ID15')* that each contain the ***DID*** of a decoder. After that, the ***desired addresses*** entered in these decoders are queried.

This results in several **registration cases**, which differ in the extent to which the reported pair of ***DID*** and ***desired address*** is fully or partially matched in the system database.

On the following page, these **registration cases** are described under **1., 2., 3., 4.**

Note: Alternatively In addition to the user options listed, there is always:

E-key → do not register Decoder!

Login case 1. *...a fully matching entry is found in the system database, i.e.* ***DID*** *of the reported decoder and the* ***preferred address*** *match.*

The user has two options:

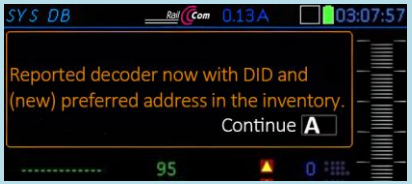
1. ***key A*** → *... almost always the norm, as long as there is no system change:*
Programme the decoder with the preferred address and System-Database confirmation.

Note: ***key A*** *can be* ***automatically*** *'pressed' by making the appropriate settings for the inventory search, so that all confirmed decoders run through without waiting for answers.*

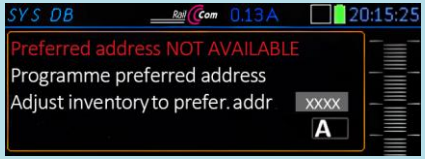
or

2. ***xxxxx + key A*** → *... Readdressing because of a system change with an address collision, or because explicitly desired.*
Program new address (also as preferred address) into the decoder and enter into existing line of the system database.

Result*-messages.*



or
if preferred address not available, i.e. impending address collision: therefore re-addressing necessary!



Login case 2.*, ... A partial match entry is found in the system database; with* ***DID*** *of the reported decoder, but* ***different address*** *(not the desired one).*

It is probably a known decoder that has since been redirected (on this or any other system)

The user has two options:

1. ***key A*** → *... the normal case: Program decoder with preferred address and customise system database to preferred address.*

Note: Key A can be automatically 'pressed' by the corresponding settings for the inventory search, so that all confirmed decoders pass through without waiting for answers.

or

2. ***xxxxx + key A*** → *... Readdressing:*
Programme new address (also as preferred address) in the decoder and enter in existing line of the system database.

Result- messages
as in registration case 1. (see left)

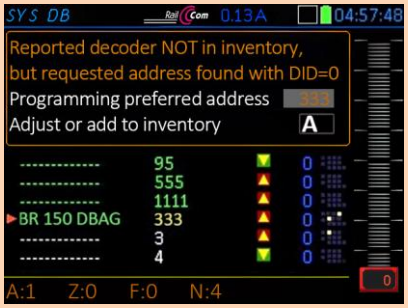
Brief overview of the

registration process 1., 2., 3., 4 :

In practice, the most common cases are **1.** or **4.**: decoders that are already known to the system **are reported,** or **completely new** decoders are.
Reregistration within the inventory search is always possible, and when the message says 'NOT AVAILABLE', it is necessary,

More seldom are cases **2.** and **3. :** these occur if decoders have been redirected 'in the meantime' or have never been registered on this system.

Login case 3. *... a partial match entry is found in the system database, with* ***the preferred address*** *of the reported decoder, but* ***DID=0***



In this case, it is unclear whether it is a decoder that is already known but has not yet been regularly registered (hence DID=0 in the system database), or whether it is a new decoder (which just happens to have the same address as a registered one); in which case it is actually registration case ***4****.*

The user has two options:

1. ***key A*** → *... Assumption that the reported decoder is the one that matches the existing entry: therefore DID add.*

or

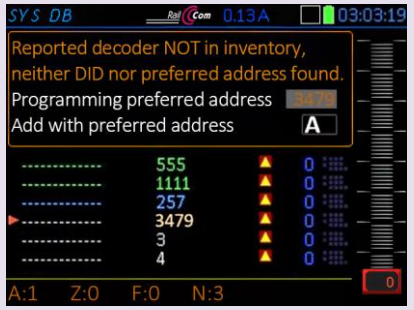
2. ***xxxxx + key A*** → *... Re-addressing, and programme the new address (also as the preferred address) into the decoder and add a new line in the system database, with the DID and address of the reported (obviously new) decoder or vehicle. The previous line (DID=0) remains unchanged.*

Readdressing (which involves adding an additional line to the system database) is necessary if, in the user's view, the reported decoder does not belong to the found entry with the preferred address and DID=0, but is a new decoder whose preferred address would no longer be available (as in registration case ***4****.)*

Result*-messages like in* ***registration case 1. or 4.****. (left or right)*

In the visible section of the system database, the (reported, added, ...) **address** is displayed in the **colour GOLD**.

Login case 4. *... No matching entry is found in the system database, neither the DID nor the preferred address of the reported decoder.*



So it is obviously a actually new decoder, which can be seen (preliminarily) at the end.

The user has two options:

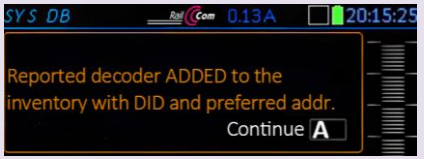
1. ***key A*** → *(... the normal case) : Program the decoder with the preferred address (usually only for formal reasons) and add a new line to the system database, with the DID and address of the reported decoder or vehicle.*

Note: ***key A*** *can be* ***automatically*** *'pressed' by making the appropriate settings for the inventory search, so that all confirmed decoders run through without waiting for answers.*

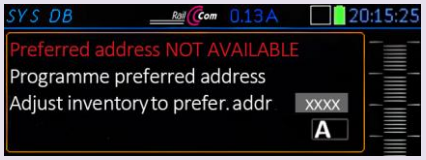
or

2. ***xxxx + key A*** → *... Readdressing, programme new address (also as preferred address) into the decoder and add new line in system database with DID and address of reported decoder.*

Result*-messages*



or



4. GUI Graphical User Interface (GUI)-Create/modify on the controller)

Settings of vehicle image, speedometer, function icons, FUMT data

*also referred to as the individual "**GUI elements**"*

General description (valid principle for all GUI elements):

There are separate ***GUI*** **set up screens** for the individual GUI elements (currently 4, extension possible):

GUI IMAGE | GUI SPEEDO | GUI FUSY | GUI FUMT (each visible in the header)

Each of these set up screens **can** be **accessed and used in** two ways:

- **either** within ***LOCO IN*** or ***LOCO MOD*** (see chapter "***LOCO IN***") by pressing the **U key** and continuing to press it, usually after entering the address and name,
- **or (more frequently in practice)** by calling up ***LOCO*** directly via the ***"E-screen"*** (see chapter ***E-screen)***, i.e. **E** + number key to the respective symbol on the E-screen, i.e. one. Key sequence

E →

The "e-screen"

E + 4 → *GUI IMAGE (G. PICT in short)*
E + 2 → *GUI SPEEDO (G. SPEEDO in short)*
E + 5 →*GUI FUSY (G. FUSY in short)*
E + 1 → *GUI FUMT (G. FUMT in short)*

In most cases, one of these calls is made from the ***LOCO*** or ***SWI*** operating modes; it refers to the active vehicle address in order to create or modify its GUI, or to copy it; see below.

Transferring GUI elements **from vehicle to vehicle** (e.g. the selected vehicle image or the speedometer settings) is made possible by using softkey functions **Copy** and **Paste.**

Softkeys II and III are set up accordingly in each ***GUI*** settings screen and labelled in the foot note.

Here is an example of the ***GUI FUSY*** (i.e. the setting screen for **function icons (FUSY))** ▶

E, 5 →

etc..

Softkey **II** → **Copying** the current data from the settings screen (in this case ***FUSY***) to the **clipboard** ***FUSY*** (i.e. only for this GUI element)

Softkey **III** → **Paste** another address from the ***FUSY*** **clipboard** into the set up screen (in this case ***FUSY***).

F → End (exit) of the set up screen **with** saving of the entered GUI element

E → First **a security question** as to whether discarding the GUI element is really meant; if again **E** → End (exit) of the settings screen **without** saving

LOCO → **E + 4** → ***GUI IMAGE*** *(**G.PICT** in short)* ← **U**← ***LOCO IN***

Selection of a **vehicle image**; shown below is the case where NO image was previously assigned (similar function if a previously assigned image is to be replaced)

from ***LOCO*** → **E + 4**

or from ***LOCO IN*** → **U**

F → to ***LOCO*** **E** →

U → to ***GUI SPEEDO***

Scroll

Scroll through all vehicle images; as soon as the desired image is found:

F → Activation of the address with the selected image , change to the ***LOCO*** mode ; or

E →Security question as to whether the selected image should really be discarded;
E (again) → Really discard the input, return to ***LOCO IN*** (without image or old image)
F → (just as if **F** had already been pressed instead of E) do not discard the selected image, but switch to the ***LOCO*** state with the selected image ;

U → Continue to the next GUI element, in this case ***GUI TACHO*** for the purpose of assigning the speedometer design and parameters (usually used if it has already come from ***LOCO IN***)

Attributes and filters:

The vehicle images (if fully imported into the database) are provided with a number of "attributes": Drive type, country, company, locomotive name, author, etc.; these attributes can be filtered on when searching...:

TP → Call up the filter window to define a filter;
in the filter window: scroll wheel and
TP → Select one or more filter terms that are applied *and* linked to.

A → **Exit** the filter window and return to the (now filtered) image collection.

E → **Cancels** the filter effect for the current selection (the markings remain)

DESCRIPTION OF the FILTER TOOL, etc.

will be added after the creation of a new database!

LOCO → E + 2 → *GUI SPEED (G.SPEEDO in short)* ← U← *GUI IMAGE*

Selection of the **speedometer display** (design, colour, km/h maximum value, km/h shunting speedometer value); and **two setting screens** in connection with the speed display:

1) to determine the **speed in km/h** from the **RailCom feedback of** the decoder, i.e. definition and **CV programming** of the "RailCom factor", i.e. the multiplier to be used to determine the actual speed in km/h from the original RailCom feedback of the decoder (which is formed without taking the vehicle mechanics into account),
2) for creating the "blue needle" curve for calculating the **speed in km/h** from the **speed step** , which is needed for all cases in which no RailCom feedback is available.

ad 1) a manually measured or estimated km/h value is typed into the "target km/h" input field, whereupon CV #136 (the RailCom factor, which only exists in ZIMO decoders) is automatically reprogrammed so that in future the "correct" feedback comes from the decoder and directly controls the pointer - magenta needle - in contrast to the blue needle, see below).

ad 2) for third-party decoders or ZIMO decoders, if there is no RailCom reception; in this case the speed display in the speedometer can only be shown by a pointer = "blue needle" calculated from the speed step. To do this, the (usually non-linear) curve for calculating the km/h value from the speed step (0 - 126) must first be defined using several pairs of values (speed step | km/h).

from *LOCO*
E + 2
or from *G.PICT*
U

"RailCom factor" setting screen (1)

Touch on blue ...

Touch on RailCom ...

"Blue needle" setting screen (2)

F → to *LOCO*
E
U → to *G.FUSY*

The correct setting of the "blue needle" curve is highly recommended (e.g. due to the future necessity for **ETCS operation**). This is done quickly and easily by moving to the individual support points using the slider (marking) and then transferring the current RailCom-reported speed **to the centre of the speedometer disc by touch** (after "settling").

The speedometer settings (input fields in the set up screen; currently active field is marked in grey):
Scroll → to the desired input field (grey background - i.e. ready for input - from field to field)
Design number for a speedometer disc from the collection, in various designs and colours
Vmx/Vrg "Vmx" is the permitted maximum speed (does not limit the "blue needle" curve). "Rg" (manoeuvring speed) defines the manoeuvring speedometer to which system automatically switches.
Tracking currently not in use
Value pairs of the "blue needle" curve; there are 4 value pairs, each consisting of the speed level specified by the slider and the km/h value (to be entered or transferred from RailCom)
F → Activation of the address with selected tacho parameters , change to *LOCO* status ; or
E → Security question as to whether the selected image should really be discarded.
E (again) → Really discard selected speedometer, back to *LOCO IN*
F → Save and use selected speedometer settings , also back to *LOCO IN*

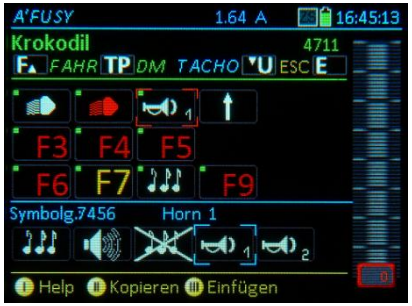
U → Continue to the next GUI element, in this case *GUI FUSY*

LOCO → E + 5 → *GUI FUSY (G.FUSY in short)* ← U← *GUI SPEEDO*

Assignment of **function icons** and parameters to the **function keys**; 10 functions (F0 - F10, F19 - F20, etc.) are displayed in the upper part of the screen as in the *LOCO* operating mode, and can also be operated using the function keys. After pressing, a symbol can be selected (and thus assigned) from the list in the lower part of the screen; the selection is made by scrolling or by jumping from group to group with the X button.

from *LOCO*
E + 5
or from *G.SPEED*
U

Press the Fu. button - Scroll icons bar - TP keys for duration/ moment – Next Fu. key, etc.



F → to *LOCO*
E
U → to *G.FUMT*

Number keys F0 ...F9→ Highlight (red frame) the relevant field
↑ (Shift) → Switches the function range from F0 - F10 to F10 -19, F20 - F28, etc.
Scroll wheel→ Scroll through the list of icons at the bottom with immediate transfer of the symbol marked with blue corners to the field at the top
X → Jump from group to group within the symbol list
(e.g. Collections oficons for light, sound, etc.)
TP→ Duration/moment switch-over (indicated by green/red dot in the assigned symbol)

F → Activation of the address with the newly selected function icons
E → Security question as to whether the selected image should really be discarded;
E (again) → Really discard selected icons, back to *LOCO IN*
F → (just as if **F had** already been pressed instead of E) do not discard selected icons , but change to the *LOCO* state with the selected icons
U → Continue to the next GUI element, in this case *GUI FUSY*

LOCO → E + 1 → *GUI FUSY* ← U← *GUI TACHO*

The specification of some address-related data often plays little role:

Format (track format): Number of speed steps (14, 28, 126); generally "126" applies
MAN mode: a relic of earlier products; for a long time (at least 2023 and later) "ZIMO " always corresponds;
Number of functions; a specification other than "28" (or in future "68" is hardly meaningful)
AZ, BZ, ABK; the system-controlled acceleration and braking times as well as braking sequences, which only become relevant when
the corresponding features are recognised in the system!

4a. GUI *(Graphical User Interface) - Handover & Takeover between the system centre (command station) and operating devices (controllers, APPs).*

and Loading the GUI from the decoder using "ZIMO file transfer"

(December 2022 trial version only with MX sound decoders)

The settings made on a MX33 controller in ***ADDR FUMT, ADDR TACHO, ADDR IMAGE, ADDR FUSY*** (the "Graphical User Interface" = GUI) should also be able to be used by other remotes. However, there are also cases where an automatic transfer of the GUI is not desired, for example when using the remote as a guest device in a system other than the home system.

Saving *the GUI in the MX10 command station & import the GUI from MX10 :*

Automatic GUI saving from the controller into the system centre (MX10 command station) :

Every change to the GUI (i.e. the ***name*** and settings in ***ADDR FUMT, ADDR TACHO, ADDR PICTURE, and ADDR FUSY)*** made on an MX33 control panel is not only stored in its local object database (***ObjDb***) , but is also transmitted to the *system centre (i.e. MX10, MX10EC, MX33ZL, ...)* , where it is stored in the Command station's database in the **data block** for this address, is saved as **"System GUI"** (there are also spaces for "Computer GUI" and "Decoder GUI" for each address)

I.e.: each "ADDR" procedure for setting a GUI element (processed by E + 1, E + 2 ...), i.e. ***ADDR FUMT, ADDR TACHO, ADDR BILD, ADDR FUSY,...*** is completed with

F button → Re-activation of the address, i.e. change to the operating state ***LOCO,*** now with the newly set GUI.

The new GUI for this address is transmitted to the command station (system centre) at the same time. *However, this only happens if* the control centre is the ***"Home System".***

"Home system" can only be MX10 or MX10EC command station or later ZIMO system control centres (see also chapter *CONFIG*, Control centre, ***Home system***), but not Roco Z21 or an old ZIMO base unit MX1.

Automatic import of GUI or GUI elements from the MX10 into the controller:

If an address that previously had NO GUI or an INCOMPLETE GUI (i.e. only a part of the 5 GUI elements) is activated on a controller, regardless of whether

LOCO IN or

from the ***LoR*** recall memory or from the ***ObjDB*** object database, or

from "External control" or "Assign address" with the U key or automatically.

the GUI or GUI ELEMENTS * saved in the MX10 are transferred to this controller.

*) As of December 2022: The 5 "GUI ELEMENTS" are: Name, FUMT values, speedometer, image, function icons.

If, for example, a name is defined for an address on controller A but no function icons, and yet for the same address on controller B both name and function icons are defined, the function icons will appear on controller A also when the address is recalled, the name however is not changed.

Special provision for function icons: Those function keys which were still represented with a "substitute symbol" (F1, F2, F3, etc.) will, when the address is activated again, receive the function icons stored in the MX10 for that address, even if some of the function keys already had icons assigned.

ATTENTION: A GUI Handover / Takeover while the MX33/32 operates in wireless mode can take a very long time (**large amount of data**).

Only the IDs of locomotive images or speedometers are transferred, not the image itself.

Menu-controlled GUI Handover in MX10 and Takeover (import fromMX10 to controller):

For all cases in which the automatic save & import described above is not sufficient (for example, because existing GUI elements are to be overwritten, which is not provided for by the automatic rules), there is the option of starting via the menu,

M key →Entry to the menu of the operating status ***LOCO***
Scroll to the desired action →

A button,
for example after selecting "Load GUI from MX10 system GUI"

In the MX10 command station (or MX10EC or future system centres), there are **3 GUI memory locations** for each address in the system objectDB

- The memory location for the **"System GUI":**
The GUIs for ZIMO operating devices (control panels, apps) are saved and retained here for all (up to 1000) registered vehicles (addresses), even if they are deleted from the operating device itself. It is the central location for exchanging between the devices.

- the storage space for the **"computer GUI":**
here, external programmes (Windows software such as ZPP, ZCS, ...) can add defined GUIs, which are then loaded by operating devices on request (menu, second option) - from there, they can, in turn, be saved to the "system GUI" on request.

- the memory space for the **"decoder GUI"**
is used if the GUI for an operating device (e.g. from the sound project) is to be used in the operating device; or (vice versa) a device-defined GUI is stored in the decoder. Both are done (2022/23 in progress) with ZIMO decoders by "file transfer", with third-party decoders if equipped for "automatic logon" according to RCN-218 of the "Railcommunity" (VHDM).

A button → Execution of the selected line of the menu.

Save and import directly from the decoder GUI
See the next sections in this chapter!

NOTE: the administration of the GUIs (Graphical User Interfaces) is a **critical issue**, because the GUI creation (type in name, select icons, ...) can be done in different ways: on operating devices (control panels, apps), in external computer software (ZCS, ...) or
by loading the integrated GUI of a decoder project or sound project.
At the same time, however, it should be prevented that GUIs that have already been created are inadvertently and irretrievably lost by overwriting: this is why there are the 3 GUI storage locations described above.

BUT: In practice, the user **only** has to **deal with this a little**, for example a GUI from ZCS automatically ends up in the correct memory location (in this case "Computer GUI"); the "System GUI" is not overwritten so that it can be retrieved again if required (if the ZCS GUI has failed).

The ZIMO **MS sound** and **MN** non-sound **decoders** (as opposed to MX decoders) are supported:

- as mfx-capable decoders (together with Märklin mfx digital systems): Märklin "mfx"
- in their capacity as DCC standard-compliant decoders (together with fitting DCC digital systems) the "automatic registration" according to RCN-218 of the "RailCommunity" or its NMRA counterpart.
- *In a **ZIMO environment*** (i.e. together with a ZIMO or compatible digital system) a procedure that uses the means of the RCN-218 for address registration (finding new addresses), but goes its own way when transferring ("fetching") the GUI from the decoder: the ***"ZIMO file transfer".***

Characteristic features of the "ZIMO file transmission:

- Compatibility with the traditional address orientation of DCC technology, which is in contrast to the "session" reference of mfx or RCN-218.
- Freedom of choice with regard to different operating concepts thanks to an unlimited type and number of vehicle images, function icons or control elements. Different operating devices receive the fitting GUI data from ZIMO decoders or the loaded sound projects.
- Integration of the GUI transmission into the general DCC and RailCom-Communication, the driving and switching operation continues (almost) unhindered. With ZIMO GUI transmission it needs...
- Not "Register first, then drive"; but "Drive off immediately, pick up GUI sometime later",

Technical functionality and features of the "ZIMO file transmission:

- A data block of up to 1000 bytes to be transmitted is divided into numerous small pieces (2 to 3 usable bytes each), which are sent from the decoder to the command station as a sequence of RailCom datagrams, each following the "normal" DCC packets.
- Note: The ZIMO file transmission is not only used for GUIs, but also for the transmission of data blocks with other contents: Texts (messages, alarm messages), train data such as type, weight, sound of each car; route profile from sensor data such as gradients, curves.

Summary of the most important steps in file transfer:

- **Requesting** the fitting GUI from the **operating device**. For example, an MX33 requires the GUI of the vehicle for display on the MX33, the ZIMO app requires the GUI for display on the smartphone, the Roco app requires the GUI for display on the tablet. This request is first sent to the control centre (via the system bus cable).
- **DCC request packet** with the request of the control centre to the relevant decoder
- **RailCom file start datagram** from the decoder to the command station. This tells the decoder that the following DCC packets (which are addressed to it) will be answered with RailCom datagrams, each containing a few bytes of the file

DCC confirmation packet to the decoder. Notification that the control centre is ready to receive File-C

M key → Menu

Context menu for the LOCO operating state:

*Scroll → **Get GUI from decoder (ZIMO**[*])*

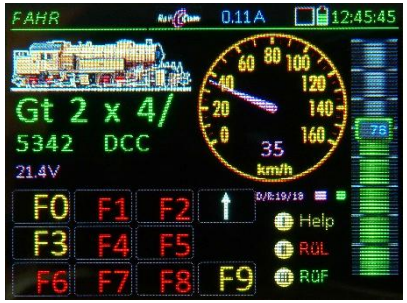
This means that the base unit should request the GUI for this device[*] (here: an MX32 remote) from the decoder, which will then provide the GUI using "ZIMO file transfer".

*A key → **Start** data transmission*

In the example

an address that is already in driving mode (recognisable by the magenta speedometer needle) is to be equipped with the GUI that the controlled decoder or the loaded sound project makes available for download.

***After approx. 0.2 seconds** (with undisturbed transmission, otherwise slower), the first data has already arrived: Information for image, selected speedometer disc and name; however, the function icons are still missing, which - depending on the number - account for the majority of the data volume to be transmitted.*



***A short time later,** the information for the first function icons is available and these icons are displayed immediately.*

..... more, now on the second level.

The GUI transfer is complete,

Brief technical explanation:

Requesting the fitting GUI from the **operating device**. For example, an MX33 requires the GUI of the vehicle for display on the MX33, the ZIMO app requires the GUI for display on the smartphone, the Roco app requires the GUI for display on the tablet. This request is first sent to the control centre (via the system bus).

- **DCC request packet** with the request from the control centre to the relevant decoder.
- **RailCom file start datagram** from the decoder to the command station. This tells the decoder that the following DCC packets (which are addressed to it) will be answered with RailCom datagrams, each containing a few bytes of the file.
- **DCC confirmation packet** to the decoder. Notification that the control centre is ready to receive file content datagrams from the decoder.
- **RailCom file content datagrams** from the decoder to the control centre (each with sequence numbers, content bytes and CRC redundancy check) in response to ANY DCC packets received on the own address until the entire file has been sent.

  This technical feature (no need for special DCC recall commands) is one of the main differences compared to the recall of data rooms according to RCN-218: the other control and signalling functions of DCC and RailCom are hardly restricted. In the context of the ZIMO control concept, this (together with other characteristics) outweighs the mathematically lower efficiency of the procedure.
- These content datagrams are NOT answered individually by the control centre.
- **File end datagram** After the entire content (the file) has been transmitted,
  a CRC redundancy check (over the entire file) is carried out for control purposes.
- **DCC claim packets** request incomplete or not-arrived content datagrams - these datagrams are then transmitted repeatedly.

[*] *"(ZIMO)" means that the "ZIMO file transmission" method (i.e. data content in a chain of RailCom packets via the rail) is to be used.*

[*] *"this device" indicates that ZIMO decoders can provide GUIs for several operating devices, e.g. for MX33 (extended MX32), or for ZIMO App or for Roco App.*

Clone" vehicles *(i.e. create a new vehicle with the GUI of an existing one)*

"Cloning" means that the GUI (name, image, function icons, ...) of an address is copied to another address so that there are two vehicles in the database of the remote that are identical in all attributes but differ in the address.

When "**cloning**", there is always an "**origin**" (= the address with an already defined GUI) and a "**clone**" (= the vehicle with the address to which this GUI is copied).

Cloning" saves a lot of work when creating GUIs by benefiting the fact that there are often several copies of a specific or similar model on the system. If there is little difference between the GUIs of two models, it makes sense to " clone" the second model from the first and then modify the GUI elements of the clone,

There are **two different cloning processes** that can be recalled from the menu (using the **softkey M**) in the ***LOCO*** **operating mode** ; the cloning window then has the relevant title in each case

Cloning from active address

Origin Address 765 Clone .... Address 765 is an example

The **GUI** (name, image, ...) **of the active address** ("origin") is copied **to another address** ("clone"). In this case, the original address (the active address) is specified; the other address, which is referred to as the "clone", is typed in; it may be new (not yet in the database of the remote) or "in use" (already in the database of the controller).

Or Cloning from origin in ObjDB

Origin Address ..... Clone 567 Address 567 is an example

The **GUI** (name, image, ...) is copied **from an existing address** ("origin") **to the active address**. In the second line of the window, the existing address is typed in as the source address. The destination address, which is referred to as the "clone" (not entirely correct, as it already exists), is, however, predefined (***LOCO*** address).

In both cases, a name for the "clone" is suggested in the clone window, this is a variant of the original name with a (2; or consecutive number), which can be changed immediately (still in the window) to the desired name (therefore labelled as an input line), i.e. the lines still follow within the clone window:

Origin Name Crocodile "Crocodile" is only an examp

Clone Name Crocodile (2

Cloning is available from the ***LOCO*** operating mode via the menu:

Softkey M→ Enters the **Menu** of the operating mode ***LOCO***

Scroll wheel → Selection in the list of menu items, in this case one of the two lines "Clone ..."

A-Key → Start the selected cloning procedure

M →

In the opened window, the **desired address** (mandatory) and a name (optional) for the clone is entered (by overwriting suggestion).

A possible address conflict (address already exists in the system) is carried out automatically and a corresponding alert appears in the window. However, the **cloning process** is **not blocked** because it may be desirable to have a duplicate address.

F-Key → activates the clone, operating mode ***LOCO***

In this case, the name of the clone has not been changed, so is suggested as "Crocodile (2)", however all GUI elements are identical to the "Crocodile" origin. The picture already shows the driving mode. ▶

Cloning also takes over the individual function states (i.e. headlights and sound are turned on immediately).

All of the clone's GUI elements are **totally independent from its origin** and can be changed at any time with the usual E-procedures ***ADDR ... FUMT, SPEEDO etc.*** or with **A+A-Key** in the ***LOCO MOD*** screen (Name, Group).

Cloning in the "opposite direction", that is "Cloning from ObjDB Origin" is often used to quickly and easily find a **fitting GUI for a new engine** or for using it as a template.

For this, the same kind or a similar vehicle on the layout is used (which already has a GUI).

In the picture on the right, it is assumed that the address of a new engine is unknown and therefore first read-out with the track-On search feature (see chapter "LOCO IN and LOCO MOD"). Then:

F-Key →switches to the driving mode ***LOCO,*** in this case only to get to the menu.

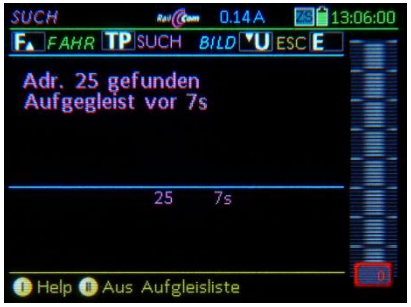
Softkey M → enters the menu

Scroll, then A-Key → opens the clone window

In the clone window (here: „ Cloning from ObjDB Origin") enter the address of the origin from which the clone (the new engine) takes over the GUI; the destination address (the clone) is however fixed.

F- Key → the clone is activated, operating mode ***LOCO***

U- Key → optional: step by step search of the ObjDB entries for a suitable origin (for a name or address with the correct GUI).



5. SOFTWARE-UPDATE, *uploading ICONS, PICTURES as well as: Saving GUI and other object data, **Exporting** GUI*

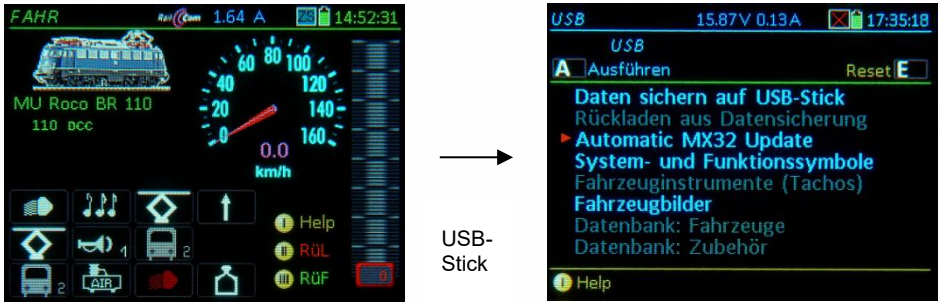
A USB stick with USB-C connection is required for these processes; possibly a stick with USB-C and USB-A (if the computer only has the "large" socket) is required; FAT32 file system; in contrast to MX32, there is no restriction on the cluster size for the MX33. which should fulfil the following criteria:

Software-Update und/oder Laden von Daten (Symbole, Bilder, usw.):

From the ZIMO website www.zimo.at, section "Update & Sound" (menu option Update - System) the **latest zip file** is downloaded and unpacked: the new MX33 software is contained in a container file, which usually (not always the same) also contains software for MX10, MX10EC and MX33. This must be loaded into the **root directory** of the USB stick.

The same USB stick (if suitable USB-A and USB-C connections are available) can then be used successively for both the MX10 command station and for all existing control panels (MX32/33).

The flash drive is then **plugged into the running MX33**, which may be done in any operating mode. MX33FU must be connected with a CAN bus cable to the MX10. The stick is automatically recognized and read, while some information (size, name...) is being displayed in the flash drive window:



A-Key → the "table of contents" is displayed: a list of entries (in order): "DATA BACK UP to flash drive", "Restore Data", "Automatic MX33 Update" and, after that, the names of expected data (not the actual file names), such as data currently present on the stick (blue) as well as expected that is not on the stick (gray) but could be.

Scroll-Rad → select the desired line from the list
A-Taste → starts the selected list position;

Selecting the **"Automatic MX33 Update"** installs the following:
1) the actual software update, so load the new software version, and
2) the data that is usually part of the new software version (but not data the user created himself or downloaded at his own request. This is being preserved throughout the update), such as

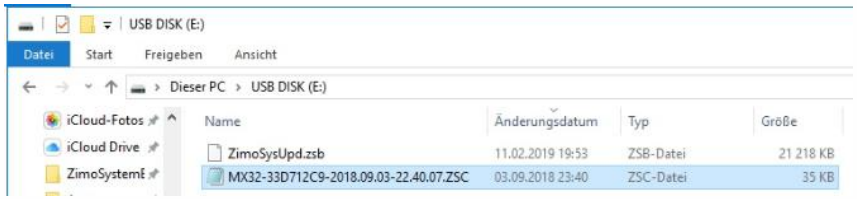
- the fonts and texts in the key operator languages German and English
- the system, function and switch panel icons, speedo dials and instrument graphics
- the device sound files (Controller sound accessible in ***LOCO IN*** with address 16382)
- the help screens

Displayed below that, a list of **data** which is available **either in addition** to the "Automatic Update" (for the new software version) or alone (without new software version) on the stick: the detailed composition of these files may change from version to version (new finished features...), but essentially, they are the following:

- System and function icons for functions, speedo, speed slider etc.
- Vehicle instruments (speedos): various speedo designs, amp meter and more
- Switch panel elements: for switch panel displays of the MX33,
- Device sound files: among others warning signals, ambient sound for the MX33 sound (.wav-files),
- Vehicle pictures: lists the picture collections on the flash drive, from which one or the other can be loaded into the controller,
- Databases: vehicles, accessories, switch panels - CV lists and sets

Automatic data backup on USB stick after the "Automatic MX33 update"

If the USB stick is reconnected to the computer AFTER the update has been carried out and checked, there will also be a .ZCS file with the vehicle database on the stick.



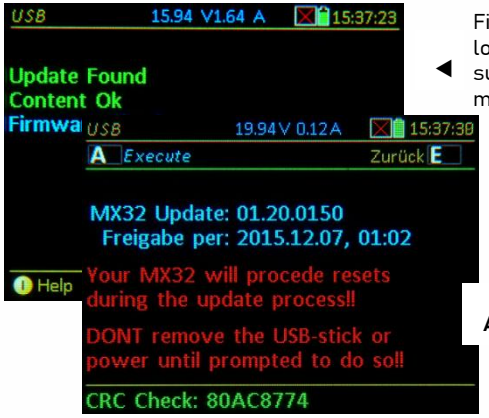
This file can be used to restore the vehicle database in the event of data loss due to the update; see section below "(Manually called up) data backup ..."

Importing data (after making a selection, i.e. "System and function symbols" etc.):

TP-Key → Marks the position to be imported
A-Key → Displays the details of this data line, **scroll** again, **TP- Key→** Select
- Last line: **„"Import all marked (selected) data"** **A-Key →** Starts importing data

The actual update process (after selecting the "Automatic MX33 Update"):

After starting the **"Automatic MX33 Update"** (with the **A- Key** or making a selection and "Import all marked ...") first, an information screen is displayed (current and new software versions as well as other data). From there, a decision has to be made if the update shall really be made:



◄ Firmware Check: The MX32 tests the software to be loaded for bad or corrupted data and shows the result. After this presentation, the control panel automatically goes to the update information screen.

Update Information Screen:

◄ Listing of the software version to be loaded, and its creation date. In addition it shows the note: *"The MX33 makes several restarts during the update! "*

A-Key (Execute)

The software update and installation of all data belonging to the “Automatic MX33 Update” is now in progress. This will take a few minutes.

After the update, the **prompt to remove the USB stick** is displayed, followed (always) by an **automatic reset** and start-up of the MX33.

(manually called up) Data backup on flash drive / restore from flash drive:

It is best (for clarity's sake) to use an empty flash drive for this purpose (See chapter Software-Update about formatting); although it may contain updates or other data.

As mentioned above, the first table entry that becomes visible after the flash drive is plugged in, is: **"Save data to flash drive"**. If this is selected and started (**A- Key**), a list of securable data such as "Object Database", "Image Database"... appears. This list may change depending on the software version.

Scroll-Rad → Select a position
TP-Key → Mark the position

Last line: **„Back-up marked data“**
A-Key → Starts back-up

If there is a **backup** on the USB-Stick - this can be from the device itself or from another controller! - the second entry **"Restoring the data backup"** becomes visible and - after selecting it - the list of saved data.

Each position is presented in two forms: with **"... replace"** and **"... add"**, depending on whether the previously contained data of the respective type in the controller should stay or be deleted.

Exporting the GUI data of an address to the flash drive:
only usable together with suitable versions of the decoder and the sound programmer ZSP

Unlike the data backup described above, the purpose of exporting is NOT to secure the data, but to **provide the GUI data for sound project expansions.** The exported data such as name, image, function icons, and some other data are then integrated into existing or future sound projects, from where they can be retrieved by the controller as needed; this saves the tedious "manual" input of the "GUI" on the controller.

For this purpose, the entry **"Export GUI to flash drive "** is displayed in the table of contents. Upon this, a file is created for the GUI of the currently active address (currently located in the foreground) and stored on the flash drive. This is a memory-optimized format, so that a future transmission from the decoder does not take too long.

Loading self-made vehicle images:

A "configuration file" (as a ".zsc" file; a sample is available from www.zimo.at, MX33 update page) must be created first and then copied together with the images to a flash drive.

The configuration file must contain at least the red mandatory lines: [FileInfo], Content =Image.Loco, Version=03.02.0002, [/FileInfo] , [Image], Id=... (the loco pictures) and the links File = 1|... and File = 2|... of the pictures (which must also be on the stick). The Id=... (max. 4-digit 1-9999) should be unique, otherwise a picture with the same ID would be replaced.

The "configuration file" is written in the Windows editor and saved as "...zsc" file (File / Save as / (file type "All files") / "....zcs" / Encoding "ANSI" / Save.

```
// Vehicle Images              * Optional commentary lines
//
[FileInfo]
Content=Image.Loco             * * Content description of the following database
Version=03.02.0002             * Version of the loco data bank, 03.02.0002 requires SW 01.20.0150 in the MX33/FU
Date=16.12.2016                * Date the dataset was created
Name=My loco pictures          * Name the loco database (all black entries are strings!)
[/FileInfo]
// ----------------------  P 8          * Optional comment line: Vehicle description
[Image]                * Image opens
Id=1004                * Unique image number (.bmp file), which must also be included in the image name.
Name=P8                * Name (Type...) of the vehicle (is also shown in the MX33)
Engine=S               * Attribute "Propulsion", for display and as a filter for image selection in the MX33
                                                  (S=Steam, E=Electro, D=Diesel)
Country=DE             * Attribute "Country", for display and as a filter for image selection in the MX33
Company=KPEV           * Attribute "RR-Company", for display and as a filter for image selection in the MX33
Epoch=1                * Attribute "Era", for display and as a filter for image selection in the MX33
Gauge=N                * Attribute "Gauge", for display and as a filter for image selection in the MX33
Author=Peter M.        * Attribute "Author", for display and as a filter for image selection in the MX33
PartNum=M37031         * Attribute "Manufacturer and article number" (M: Märklin, R: Roco ...)
File=1|Loco 1004.01 P8 [M37031].bmp  * Link to small image (.bmp-File 150 x 50 px, 24 bit)
File=2|Loco 1004.02 P8 [M37031].bmp  * Link to large image (.bmp-File 279 x 92 px, 24 bit)
[/Image]               * Image is closed
//
[/Image]               more pictures...
```

6. THE "E-SCREEN"

The "E-screen" is the menu page for different applications.

E-Key (from the ***LOCO*** or ***SWI*** mode)

→ ***E-Screen*** ▶

The meaning of the **E- Key** in this instance is not for "Escape" but for "**E**instellungen", which is the German word for "Settings".

The E-screen symbolically shows which "setting" each key leads to (application, operating status).

The key sequences E + ... have the following meaning:

E + F → ***OP PROG*** ,
Entry into Operational Mode programming for CV and address programming on the main track. See chapter "OP PROG".

E + MN → ***SERV PROG***,
Entry into Service Mode programming for CV and address programming on the programming track. See chapter "SERV PROG".

E + 1 → ***ADDR FUMT,***
Various settings for the GUI of each vehicle (this is NOT decoder programming), in particular the number of function keys. See chapter "ADDR: FUMT...".

E + 2 → ***ADDR SPEEDO,***
Settings for the display speedo (Design, Range etc.). See Chapter "*ADDR*: *FUMT...*".

E + 3 → ***ACC LIST,***
Displays a list of accessory addresses where turnouts and signals, connected to accessory decoders, can be operated from. See chapter "ACC LIST".

E + 4 → ***ADDR IMAGE*** ,
Select or change a vehicle image. See chapter "*ADDR*: *FUMT...*".

E + 5 → ***ADDR FUSY,***
Assignment of function symbols to the function keys, See chapter "*ADDR*: *FUMT...*".

E + 7 → ***ROUTES,***
Setting of routes in connection with suitable dispatching programs (ESTWGJ, STP...).
No description in this manual.

E + 8 → ***StEin LIST*** ,
Displays a list of existing StEin modules for monitoring and actuating their connections, i.e. track sections (HLU, occupancy detection, short circuits), turnouts, inputs, signals...
See chapter "StEin LIST".

E + 9 → ***AOS's,***
not yet implemented and therefore no description.

E + 0 → ***CONFIG***
SW-Version information, as well as viewing and changing of the command station settings and the "own device" settings (i.e. the present controller): language, time & date, stop mode modifications, operating details, statistics and much more. See chapter "CONFIG" (formerly "CAB CONF").

For details go to chapter ***ADDR ... FUMT, SPEED, IMAGE, FUSY, OP PROG, SERV PROG, SWI***

7. CONFIG *Info and Settings for Command Station and Controller*

E-Key (to the E-Screen) + **0-Key** (number key 0) → Entry to ***CONFIG*** from ***LOCO*** or ***SWI***

NOTE: the screen content shown here may not match exactly the actual appearance on the controller display, because it is permanently developed.

E →

Key 0 = CONFIG →

Scrolling wheel →Move to a specific line in the ***CONFIG*** menu
A-Key → Enter the selected page

E-Key → Back from the menu to the main list
E-Key → Return to the operating mode ***LOCO*** or ***SWI***

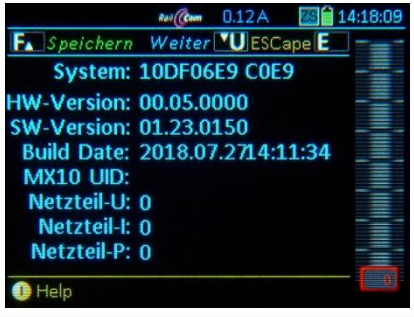
Switch between the detail pages with:

U- Key → continuous to the next ***CONFIG*** screen
↑ (Shift) + **U- Key** → returns to the previous page
↑ (Shift) + **F-Key** → back to the previous ***CONFIG*** screen

Within a page:
Scrolling wheel → move the cursor to the desired CONFIG line
Rocker switch→ Select from available options
Number Keys → Enter a value

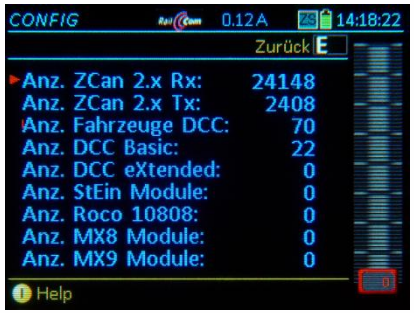
The new value will be save das soon as you switch to another line with the A-, E-, U- Key or Scrolling wheel.

Command station – HW/SW Version Info



HW-Version: : Version of the MX10 (MX10EC) board
SW-Version: Software Version installed
Build Date, ... Time: Software development date
MX10 UID: Factory serial number
Power supply

Command station – Statistics



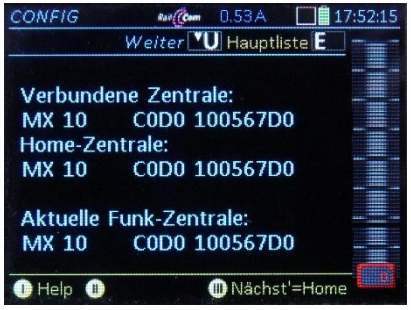
Various current data regarding communication and devices connected to the command station.

CONFIG with the "old" command station MX1 (MX1EC; MX1HS)

<<< via the Menu (M_Key) to total memory erase Hard Reset

<<< CV-Programming of station MX1

Command Station, Home System



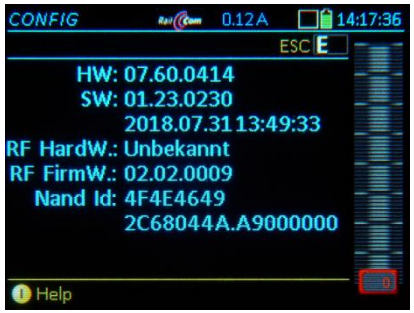
Connected station: MX10 (also applies to MX10EC), MX1, Z21, ... (are not suitable as a home centre)
ID is displayed
Home-Station: MX10 (also applies to MX10EC),
ID is displayed
Current radio control centre: MX10 (also applies to MX10EC) which communicates with the control panel via MiWi
ID is displayed (this should be the same as the last control centre connected by cable)
Softkeys for selecting the home system:

II (verb' =) Currently connected system should become Home system (if it is not already)

III (Next = Home) Not the currently connected control centre (e.g. before delivery, during service or demonstration) should remain the home system, but the one connected at the next Power-ON.

Changes to the **GUI** and **FUMT** data on the controller are automatically written to the central memory of the **Home System** (see "GUI" section) When the address is called up, data is transferred from the **Home System** (Both applicable if connected).

This Controller - HW/SW Version Info



Name: default "MX32"; a new unique name for the controller can be entered here

Language: Selection of the languages available in this device (loaded from the flash drive or already present in the controller). Currently German / English.

Style: System and function symbols come in four different styles (see chapter "Styles"); two can be loaded in to the controller and are available here for selection

UID: factory-assigned serial number (Read-only)

HW: Hardware Version (Read Only)

SW: Software Version with Date (Read Only)

RF HW/FW: SW Version Radio (Read Only)

This Controller – Date, Time, Timeouts

Date (Y.M.D): The date for the internal calendar in the controller (NOT the system's clock, see MX10 manual).

Time (H.M.S): Internal clock setting (NOT the system's clock, see MX10 manual).

Fast Clock: Acceleration factor compared to real time.

E-Timeout: The time the controller remains in the E menu page after pressing the **E-Key** (from the ***LOCO*** or ***SWI*** mode) before the screen returns to the previous page if no further selection is made.

Screen Saver: A **screen saver** is activated after the controller sits idle for the time entered here: Name and address number moves randomly around the screen.

Idle Standby: Only useful when the controller is operated in **radio mode**: if the controller sits idle for the time specified here, a special **stand-by screen** opens with a clock counting down the seconds until shutdown (see "Stand-by time").

Standby-time: Time in seconds for the timer in the stand-by screen *) to count down to 0. During this time the controller can be returned to normal operation (by pressing the **A-Key**) without going through the boot-up sequence.

NOTE: The (local) time set in the MX33 is not transferred to the MX10 and is no longer displayed after a restart of the MX33. The parameters set in the MX10 are sent to the MX33 via the CAN2 bus.

*) The Stand-by screen appears when power is lost (i.e. cable unplugged) or when the cab sits idle while in the radio mode (in order to conserve battery power)

This Controller – STOP&OFF Management

BCS Mode Type of Broadcast Stop (BCS)
= BCS emergency: Emergency broadcast stop
= BCS SS0: Normal broadcast stop **)

S-Key short = Single stop: Emergency stop for "your" engine
= Broadcast stop: Stops all engines per BCS mode

S-Taste long = Broadcast stop: Stops all engines per BCS mode
= Track power OFF: Track 1 or track 1+2

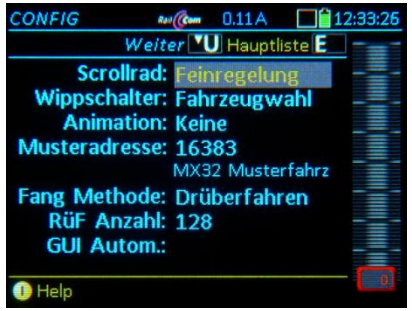
R-Key = Direction: Changes direction at any speed
= Single stop: Direction key initiates an emergency stop
for "your" running engine; direction change only at stand-still (useful when S-Key = BCS).

Above speed step = The effect of the R-Key described above as a single stop key applies only above the speed step defined here (e.g. changes direction immediately at lower speeds).

NOTE: There is no track power "OFF" with the Roco Z21, only "STOPP" (as in broadcast stop per ZIMO terminology).

**) BCS (= Broadcast stop) applies to all engines but can be issued as emergency stops or "normal" stops (as programmed in CV #4). "Emergency" stops trains faster but may also cause derailments in tight curves; in such cases a non-emergency BCS would be better suited.

This Controller – Operating Mode LOCO



Scrolling wheel = Fine control: the speed (which is normally set with the speed slider) can be fine-tuned in small steps of +/- 10% of the selected speed steps (14/28/128).
= Vehicle selection: scrolling through the recall memory (***LoR***) with immediate activation even if the memory window is NOT visible *)
= No function.

*) The above selections are not valid when the recall memory (LoR) window is open, because the scrolling wheel is then used exclusively to scroll through the list (= for selecting a vehicle, but in this case also requires the **A-Key** to activate the address).

Wipp-Schalter= No Function.
= Fine control: the speed (which is normally set with the speed slider) can be fine-tuned in small steps of +/- 10% of the selected speed steps (14/28/128).
= Vehicle selection; scrolling through the recall memory (***LoR***) with immediate activation even if the memory window is NOT visible.

Sample Addr: Default 16383, although this is not a valid DCC address, it nevertheless serves as a sample address. Any (real) address can be used instead.

The settings of this sample address (function symbols, speedo etc.) are taken over for every new address that is activated via ***LOCO IN.***
If the sample settings are to be changed, activate the sample address like a real address and make the changes in the ***ADDR*** ....

Animation = none: the function icons remain motionless
yes: the function icons are shown animated provided such images exist.

This Controller – Operating Mode SWI

Display = Sequential numbers: a sequential number is shown above each switch icon (press the ↑ (Shift)-Key to see the corresponding address)

= Addresses: the accessory address is shown above each switch icon (press the ↑-Key to see the sequential numbers).

Switching = Keys + **Touch**: The accessories can be operated with the keyboard or the touch screen

= **Touch**: Accessories with touch screen only.
The function keys are available for normal loco functions. The shift key only works if a switch panel is active.

This Controller – Object Groups

Some "groups" (e.g., steam and diesel engines...) are
– already pre-defined (as shown), or
- can be defined here
and assigned to the objects (mostly vehicles), usually in the ***LOCO IN*** mode while a new address is assigned or modified.

This Controller – Info / Statistics

Battery status (mV): Zeigt die aktuelle Akku-
= > 3,8 V (3800 mV): charging
= < 3,6 V (3600 mV): discharging
Life Time: Total operating hours since its first start-up.
Power On Time: in seconds; current operating time since the last power up
Number of vehicles, panels, texts etc.: Each line lists the number of articles stored in the MX32 (i.e. locomotives, accessories, texts, labels, function icons, loco images...

Various parts of the memory or data content can be deleted; this is usually not necessary;

This Controller – Clear Data

An important point in Clear Data:

Deleting vehicle images

When the **"Vehicle images"** line is selected, either

TP-Key → Mark ALL images for deletion and then

A-Key → Delete ALL images, whereby the security enquiry (window) must be answered with the A key beforehand.

... or ...

A-Key → Switch to the main list of locomotive names in order to select the images to be deleted or to apply attribute filters.

Scroll + TP-Key → Set markers (red dots) in the main list of locomotive names

(Shift) + **TP-Key →** Switch to the list of attributes (drive, era, ...)

Scroll + TP-Key → Select certain attributes (e.g. "Diesel" drive and era "III") (red dots) to delete all images that fulfil these criteria.

A-Key → Back to the main list of locomotives; additional images can be marked for deletion there, or the deletion of individual images provided for by the attribute marking can be cancelled.

E-Taste → Delete markings from the list of attributes and return to the main list.

A-Taste → Delete all MARKED images, whereby the confirmation prompt (window) must be answered with the **A-Key** beforehand.

E-Key → End, exit *Clear Data, CONFIG* (several times depending on the situation)

ab SW 01.25:

Various parts of the memory or data content can be deleted; this is usually not necessary;

ATTENTION: Deleting certain items may **negatively affect** or **destroy** the functionality of the controller.

8. Operating mode LOCO

In ***LOCO*** you can
to exit from most operating states,
also from ***LOCO IN***:

F-Key → *LOCO* (new vehicle)

LOCO is the main status of the control panel - where vehicles (and trains) are driven

NOTE: Consist operation in the next chapter.

There are various display options in the ***LOCO*** operating state:
the possibilities depend on which **GUI-elements** (GUI = "Graphical User Interface") are available for the active vehicle (i.e. are defined, which is NOT necessarily always "visible"):

- always present: the (vehicle) **address**,
  where a speedometer is usually automatically included from the start
  (usually with a green disc, but this can be set differently by the sample vehicle at address 16383),
- optionally (to be entered in ***LOCO IN***, when entering the address or via A + A) a name,
- optional (to be selected in ***ADR SPEEDO***, via E + 2) a speedometer (various discs, or "no speedometer"),
- optional (select in ***ADR IMAGE***, after entering the address or via E + 4) a picture
- always available (to be "filled" in ***ADR FUMT***, via E + 5) the tableau of function icons,
  the GUI elements (if available) can be made visible (or hidden) on the screen as desired or arranged differently.
  You can choose between these display types using TOUCH (alternatively also via the ***LOCO*** menu); the following applies in principle:

TOUCH on GUI-Element →
→ THIS element is displayed alternately larger <> smaller.

Text field (name & address) larger - smaller, image larger - smaller, speedometer large disc - small digital display
(other GUI elements are displayed in a suitable way (or not at all if there is no space)

The only GUI element that CANNOT be changed by **TOUCH** is the "function icons" panel (**TOUCH** is used to switch the functions themselves); the "function icons" panel is only displayed if "address & name" AND "picture" are made small OR if a large picture AND the digital display are shown instead of the speedometer disc.

***TOUCH** in SPEEDO* ↑↓ ***TOUCH** in IMAGE*

EXAMPLES

for modifying the display using TOUCH:

◀ "Normal view II": Display with **large picture**, name and address, NO disc speedometer, only **digital speed display**). Also visible in the picture is the magenta- coloured **"progress bar of the background reader",** which reads out important CVs unnoticed and without prompting, so that the values are available without waiting time if required, and the **"Quality of Service" dot** in different colours (green is very good, etc.).

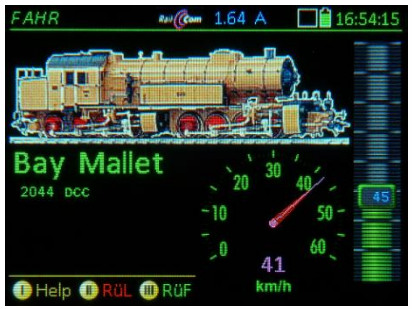
TAP the image (in normal view II)→ Image becomes small,,
Digital speed display is replaced by speedometer disc

◀ „Normal view I": Display with **small picture**,
Name and address, windscreen speedometer.

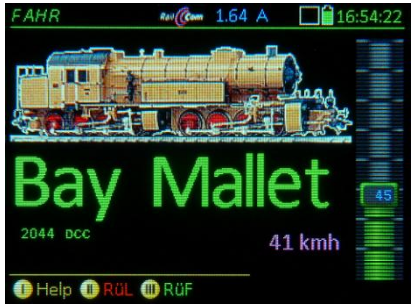
TAP the speedometer (in normal view I)→ Image becomes large, Speedometer disc is replaced by digital speed display

TAP the speedometer (in normal view I)→large image and speedo below. There are NO function icons in this view.

Display with large image and speedometer below ▶



TAP the speedometer → Returns to "normal view"



TAP the text field (in normal view; Text field = Name & Address or just address without name) → Speedo is removed to make room for larger image and name & address

◀ Display with large text for name & address but without speedo and function icons

(TAP the image + Speedometer→ return to "normal view I")

TAP the image + text (in "normal view I") → Digital speed display, large text and function icons.

(TAP the text + Image + Speedo → Return to "normal view")

The MN-Key ("Manual") and the RG-Key ("Shunting")

MN- ***Key*** **– "MANual"** – MN-LED flashing red: Cancels the HLU speed reduction (Signal controlled speed influence through MX9 or StEin modules) or ABC (Stop on sections with asymmetrical track signal – see decoder instruction manual).

RG- ***Key*** **– Shunting** → Press 1 x: RG-LED yellow, speed range is cut in half (64 instead of 128 steps).

Press 2 x: 1/3 of actual speed (42 instead of 128 steps). Press again: Shunting mode OFF; RG-LED dark.

Press **RG-Key** again → Switch off shunting status; RG-LED dark

HINWEIS: NOTE: The volume of sounds that can be triggered with function keys can be changed (by turning the scrolling wheel): keep **the function key pressed, turn the scrolling wheel** ("up" = **louder**, "down" = **softer**).

The East-West Direction MX33-Umstellung

The ZIMO system offers the possibility for a vehicle **equipped with ZIMO decoder** (types from approx. 2005 and later, with up-to-date software) displaying or setting the system-related direction in addition to the (usual) vehicle-related direction, i.e. the direction of travel relative to the track system or, if the train is stationary, the expected direction of travel when the train starts moving.

1. On the operating device (ZIMO controller, ZIMO APP, ...), the **direction arrows "East"** and **"West"** indicate which system-related direction is currently valid. This information comes - if possible - from a RailCom feedback directly from the decoder, otherwise it can be assigned manually - distinguishable by the colour of the arrows.
2. From the control unit (MX33 controller, ZIMO APP, ...), a special control element (key or touch field) can be used to drive in the **desired direction (east or west)** or to switch the lighting to the desired direction when stationary.

The terms "east" and "west" do not really indicate the direction of the compass; they could also be interpreted as "left" and "right", for example; the decoder actually detects the polarity of the DCC track signal, which is symmetrical at first glance, but has a measurable (i.e. system-related) phase polarity.

The display of the current "east-west direction" (in addition to the "forwards-backwards direction")

on the ZIMO MX33 control panel is done by the horizontal direction arrows "East" and "West", integrated in the east-west direction key "ROW", and automatically switched on if the so-called "forward-backward direction" is activated for the active address the so-called "direction status byte" arrives often enough via RailCom for the active address (on the MX32, "East-West" is switched on manually on the MX32). The "normal" vertical direction arrows "Forwards" and "Backwards", integrated in the "RVR" forward/reverse button, are always switched on (because they convey the direction information contained in the "normal" DCC drive command)
The colours of the illuminated direction arrows have a certain standardised meaning (MX32, MX33, ZIMO APP); therefore, the exact sequence when changing direction is not described here, but rather the meaning of the lighting.

Green *arrows "Forwards" and "Backwards"* = usually barely visible because changing quickly from **green** to **yellow**; new direction not (yet) adopted by the central command station after pressing the direction button "RVR"; longer visible only when changing direction from travelling AND active system-controlled braking time (BT) until zero crossing or error situation in remote control operation (**red** opposite arrow).

Yellow *arrows "Forwards" and "Backwards"* = central **command station confirmed** of the **DCC drive commands** with direction bit in the desired direction.
Note: The yellow colour of the "Forward" or "Reverse" arrow does NOT mean that (or that not) a (RailCom) confirmation of the displayed direction of travel by the decoder has taken place. This can only be seen from the colour (*magenta*) of the "East" or "West" arrow! *Magenta* "East" or "West" means "incidentally" that forwards-backwards is also signalled. The colour logic, that "Forwards" and "Backwards" always remain *yellow* (although *magenta* would be logical), serves to the similarity of the controller types.

Green *arrows "East" and "West"* = direction of **expected travel** after pressing the direction key, but NOT (YET) reported back by the decoder; the colour ***green*** or *magenta* of the arrows "East", "West" shows for the forward-reverse direction AND for the "East-West" direction if RailCom confirmation is received.

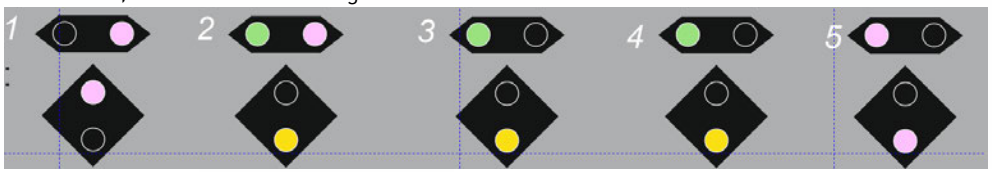
Magenta *arrows "Forward", "Reverse" "East" and "West"* = **direction of travel reported back** by the decoder; the *magenta* colour indicates for the forward/reverse direction AND for the "East/West" direction that RailCom confirmation has taken place.

Blue *arrows "East" and "West"* = NO feedback from the decoder via RailCom (not even via radio); the **east-west direction key "ROW"** can be used to manually switch the "east-west direction" back and forth, independently of the **"RVR" forwards/backwards key.** The vehicle follows the last key that is pressed.

Function of the direction buttons ***"RVR" and "ROW",***

if RailCom feedback is available, i.e. arrow "East" or "West" in magenta: Since the feedback from forwards/backwards and east/west is sent in the same RailCom message set, both directional information is always sent simultaneously (or simultaneously not).

This also has the consequence that in this case both direction buttons **"RVR"** and **"ROW"** have the same effect, in other words a change of direction.



Drive in previous direction: *Vehicle with direction "Forward" and "East", signalled via RailCom direction byte, therefore "RVR" and "ROW" in both buttons.Arrows in magenta.*

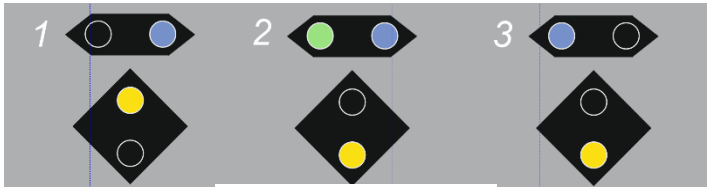
Direction key *"RVR" or "ROW" (same effect) →*
VR display (in "RVR" button) changes immediately, in the example to "Backwards" in yellow, OW display (in "ROW" button) shows the target direction in green, the previous, still current direction magenta - often seen very briefly, only as long as no DCC commands have been sent in the new direction.

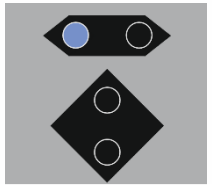
Direction change *is ongoing:*
DCC commands are sent out in the new direction, i.e. "backwards".
It is not visible whether the decoder has already initiated the direction change, therefore no arrow in magenta; so far no RailCom message for reaching the new direction (zero crossing)

Drive in new direction:
The feedback from the decoder about the change of direction (at least zero crossing) has been received.
The vehicle is now driving in "reverse" and "west" direction, signalled via RailCom direction byte, therefore in both buttons:
Arrows in magenta.

NOTE: To drive in a specific east-west direction, the **"East"** and **"West"** arrows are monitored; the direction button is only pressed if the "wrong" arrow is found.

if no RailCom feedback, i.e. arrow "East" or "West" in blue (or dark)
↑ (Shift) **– ROW-key** simultaneously (↑ forward) → Switching **on** and manual **reversal** of the Assignment of the blue arrows **"East"** and **"West";** only possible "on sight" due to lack of feedback.
↑ (Shift) **- 3 x ROW-**key in succession (↑ forward) → Switch off the blue arrows **"East"** and **"West".**





Drive in previous direction:
Vehicle with direction "Forward" and - manually assigned - "East".
therefore
"East" arrow in blue.

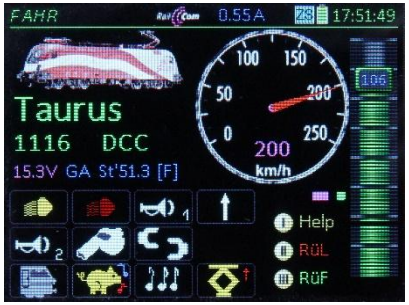
Direction key *"RVR"→*
VR display (in "RVR" button) changes immediately, for example to "Reverse". OW display (in "ROW" button) shows the destination direction in green, the previous direction in blue - often visible for a very short time, only as long as no DCC commands have been sent in the new direction.

Drive in new direction:
DCC commands are sent out in a new direction, i.e. "backwards".
In the absence of feedback, the east-west direction is not confirmed, hence the "West" arrow in blue.

Direction key *"ROW"→*
VR display (in "RVR" button) goes dark until the "RVR" button is pressed again.
The vehicle only receives east-west direction commands, it is therefore controlled in relation to the system.

Other cases: Reverse loops, hub
HLU-Direction
ABC-Direction
Internal shuttle operation

The INFO line under Name/Address



Here " is normally" displayed:

- the driving voltage measured by the decoder (useful information on the status of the system cabling,
- the track section number (StEin), where the vehicle is currently in blue colour because it is from the system.

M-Key → (menu) Here you can switch off the info line or add further information (to the alternating display; e.g. decoder type, error messages from the decoder (overtemp, etc.)

The LOWER SCREEN HALF

can also be used differently than in the "normal" LOCO presentations.
This will usually replace the function icons with the desired content

Softkey III → Opens the ***LoR*** (= Loco Recall memory) in the lower screen half
repeat (while ***LoR*** is open) **Softkey III → Close**

Softkey II → Opens the ***ACC LIST*** (Accessory addresses),
repeat (while ***ACC LIST*** is open) **Softkey II → Close**
Repeat (while panel is open) **W- Key → closes** the panel, returns to ***LOCO***

Tap on the ***LoR*** or a ***LIST*** → to toggle between full screen and half screen

E-Key + 8 → Opens the ***STEIN LIST*** (Stationary Equipment Module "StEin")
repeat (while ***STEIN LIST*** is open) **Softkey II → Close**

W- Key → Opens a switch panel; and switches to the operating mode ***SWI***

Softkey III → *LoR* in the lower half of the display, ▶
see description in this chapter

The ***LoR*** (= Loco Recall memory) provides an overview of and quick access to all vehicles (addresses) and trains that are already in this controller. 4 lines can be seen at once, the rest is made visible by scrolling.
Each line contains name, address, traction affiliation (T1, ...), speed (blue: speed level in 1000 steps), magenta: km/h reported back via RailCom), mini tableau functions F0 ... F9.

A-Taste → activates the marked vehicle.▲
Softkey II → *ACC LIST* opens,
◀ (= List of accessory addresses) shows the addresses of accessory decoders as well as MX8 and MX9 modules registered in this cab. The accessories belonging to the marked address can be operated by the number keys.
A-Key at the line >NEW< → Enter another address as single/paired function.

E-Key + 8 → *STEIN LIST* opens
◀ (= StEin Module List) shows the track sections, turnouts and other inputs / outputs of the StEin modules.

W-Key → *SWI* in the lower half of the display ▶
also see chapter *Operating mode **SWI***

In the operating mode ***SWI*** (= switching turnouts, but actually operates all types of accessories) "panels" or track diagrams are filled with icons for turnouts, signals, etc., which are used for operation on the layout.

See chapter ***SWI*** – for object-oriented control of

The ZIMO system is based on sophisticated data management. This can largely be ignored for small applications (i.e. the garden railroad with 10 or 20 engines), you can "just start driving". If a large layout (i.e. with 100 or several 100 engines) is to be operated, it is advisable to know the system's philosophy.
Especially important are the two data structures in the MX33 controller:

*Loco Recall = **LoR** & Object Database = **ObjectDb***

Each "Driving Object" (Address, vehicle, train) that is activated and driven will remain in the ***LoR*** (recall memory) and the ***ObjectDb*** (object database) even after it is deactivated; the ***LoR*** however is limited to a maximum of 128 objects.

Both, current driving data and the so-called "GUI data" are stored there. The latter determines the appearance of the vehicle on display screens, such as name, image, function icons, etc.

The task of the ***LoR*** (recall memory) is to quickly find, observe and activate vehicles (trains...) that were active on this cab earlier, but were moved to the background (inactive) due to the activation of a different vehicle; the ***LoR*** is also a kind of "Favorites list".

It is recommended to delete entries that are likely no longer needed from the ***LoR*** (C-Key in the ***LoR*** table display); this will NOT delete them from the ***ObjectDb*** (object database), so that the data can be retrieved by the ***LoR*** if needed.

Objects stored in the ***LoR*** (loco recall memory) are also ALWAYS stored in the ***ObjectDb*** (object database) and identified there by the ***green*** font. The reverse is NOT the case; there are usually many more entries in the ***ObjectDb*** (blue font, if the address is not in the ***LoR***).

There is also access to the ***ObjectDb*** - with the possibility of transfer - to the object databases of other devices (control panels, control center, app's, if integrated). Deleting an object from the object database is also possible (C-Key...) but should be proceeded with caution (create a back-up first).

See chapter ***ObjctDb (Object Database)***

Another important data structure in the system is the **object database of the MX10 command station**, which is primarily used to organize the data packets (DCC, MM...) that are to be sent via tracks to the vehicles and accessories.

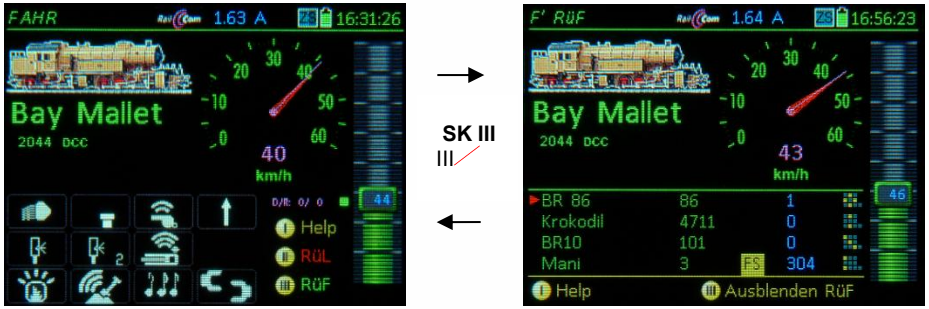
There, copies of object database entries of all cabs and other input devices are created automatically. As already briefly mentioned above, entries can also be transferred back to the cabs, or just parts of the data, i.e. for the purpose of the so-called "GUI acquisition" (if, for example, the name and the image for a particular address from one controller should be copied to another controller).

The loco recall memory (*LoR*) is **always** available in the operating mode *LOCO* to activate vehicles, whether it is visible or not:

F- and **U-Key** → **switches directly** from address to address (more precisely: from driving object to driving object, which may also be consists or trains), ascending or descending in the loco recall memory (*LoR*). If required, the object list containing names and addresses can be made visible in the lower half of the screen ("Show"):

Softkey III *LoR* → **displays** the loco recall memory (*LoR*), instead of the function keys.

Softkey III *LoR* → **hides** the loco recall memory (*LoR*), the function keys are shown again.



TAP on *LoR* - → **Full-screen display** ▶

TOUCH on *LoR*

In Vollbilddarstellung gleiche Funktionsweise wie in Halbbilddarstellung; aber mit mehr Zeilen.

At the end of the *LoR* list is always the line >NEW<, for an alternative way to get to the address input screen (instead of using the A-Key or ↑ + A):

TAP the large *LoR* - field → returns to the **Half-screen representation**

The **individual rows** of the *LoR* contain in the **first level** (see below for second level):

* First column: Vehicle name (if one is given, otherwise substitute-dashes)
  green: "normal" single vehicle
  turquoise: vehicle that is part of a consist (see third column)
* Second column: Vehicle address (without further identification, if it's a DCC address)
* Third column: External control, (External) Consist *)
  "FS": Vehicle controlled by another controller
  "T1, T2...: Vehicle is part of a consist
  "FT(..)": Consist belongs to another controller
  "FS(..)": Consist controlled by another controller
* Fourth column: Speed
  magenta: km/h-value from RailCom feedback, or
  blue: Speed in 1024 speed-step - scale (NOT in km/h)
* Fifth column: Direction (stand-still / driving)
* Sixth column: Small function table F0 ... F9 (to see which functions are turned on, i.e. headlight, sound...).

Rocker switch → **to second level of the** *LoR* display.

The **individual rows** of the second level contain:

* First column to fourth column: same as the first level.
* Fifth column: **Group** info (if available)

The arrangement of the symbols is not yet final, depending on the development level of consisting!

Activating a vehicle from the LoR and other actuations

Softkey III → Open or close the loco recall memory (also see above)

F- and **U-Key** → **switches directly** to the next address in memory; ascending or descending, regardless of whether the *LoR* is visible (open) or invisible,

Scrolling wheel →Scrolling through the recall memory and **marking** with the cursor (regardless of the scrolling wheel setting in *CONFIG*),

A-Key → ***activation from the LoR*** (bringing the marked address to the foreground).

Softkey II →Sort (Change the order of objects in the recall memory),

C-Key → Delete marked object from the recall memory (remains in the Object Database!).

Activating a vehicle from the Object Database

E-Key (to the E-Screen) + 6-Key → switches to the *ObjectDb*

Scrolling wheel → scroll through object database,

A-Key → **activates** the selected objects and returns to the operating mode *LOCO.*

For more info about the object database: see chapter *ObjectDb.*

▶

9. SYSTEM-CONTROLLED CONSIST OPERATION

In the "DCC world" there are two ways of forming a VEHICLE CONSTITUTION: ("consist"):

- Decoder-controlled consist operation, because of historical reasons also known as "advanced consist":

  By programming CV #19 (and possibly also #20) in several (two, three, ...) vehicles to the same address, i.e. the "consist address": then the speed and direction control in these vehicles can only be controlled via this address and is therefore synchronised, while the functions can be switched via the respective individual address (in CV #1 or CV's 17,18) or via the consist address (in CV's 19, 20), depending on the bits in CV's #21, 22.

System-controlled consist operation, a special ZIMO method; the term "consist" (double traction, triple traction, ...) is used for this type of connection, as opposed to "consist for decoder-controlled operation".

The control of several (two, three, ...) cars as participants in a consist is done via their own addresses; the system ensures that the driving commands for speed and direction are the same in each case and are sent quasi-simultaneously (in close succession).

This is ZIMO's preferred consist operation for its greater flexibility and is described in this chapter.

-Upcoming (not yet realised at the time of this description):

The Train control

The entire train (locomotive(s) and coaches) is controlled via a single address in the form of a single decoder, which acts as a real or virtual "train control computer". This means that there are not several decoders of the same rank with the same "Consist address". The other vehicles in the train are connected to the train control computer via a real or virtual train bus and receive their control commands from there.

➢ *Forming a new Consist*

A consist is a combination of locomotives with jointly controlled speed, direction, MAN and shunting function. The starting point is the operating mode ***LOCO***; the addresses intended for the consist must already be in the ***LoR*** (loco recall memory).

Softkey III → ***LoR*** opens (if not already open); the third column (after the address) gives information about existing consists.

Scrolling wheel → select the first vehicle to be part of the consist and with the (this can - but does not have to - be the currently active address)

NOTE: The addresses intended for the new traction to be set up must already be in the ***LoR*** (retrieval memory) and must not be actively controlled in another operating device (i.e. NO FS marking) or already belong to another traction (i.e. NO "T..)" or "FT)").

Show ***LoR***; and scroll in the ***LoR*** to "BR10", which will be the first consist locomotive in the example

Scrolling wheel → select the first vehicle to be part of the consist and with the

TP-Key + Number key → **assign** the selected vehicle to the consist, i.e. "T1", "T2"..." etc. ▶

(ATTENTION: 7 sec timeout). The consist designation ("T1", " T2"...) is visible in the ***LoR*** line AND in the upper main field (if this engine is in the foreground); the line in the lower half AND the main text above appear in the color turquoise.

The assignment to a consist causes the relevant address in the ***LoR*** of other controllers to be marked with "FT (...)", and is therefore no longer available for consist building in those controllers.

Scrolling wheel → select second vehicle for consist and start with

TP-Key + Number Key → i.e. add "T1", "T2", ..." to the first vehicle or to the already started consist. ▶

As soon as a traction consists of at least two addresses that are all marked with the same "T1", (or "T2", "T3", . . .), are **automatically shifted up** in the ***LoR*** and the respective vehicles are **controlled together.**

Scrolling wheel → third vehicle etc.

➢ *Drive existing (e.g. newly formed) consist:*

TP-Key (only when ***LoR*** is **off**)→ ▶

Cyclical change to the next address within the consist if a consist vehicle is in the foreground (i.e. turquoise text and marking "T1", "T2"...)

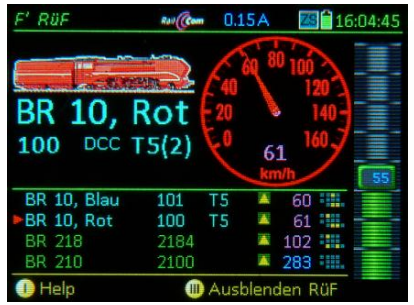
NOTE: It is of course also possible to switch among the consist addresses with ***LoR*** **open**, but requires an extra step: by scrolling (**with the Scrolling wheel**) to the address and activating it (**with the A-Key**).

The TP key has a different meaning in this situation has a different meaning; see below

Softkey III → **Open/Close** the ***LoR*** is always possible, in order to see the function icons for the active vehicle in the foreground, for example. (The functions themselves can also be switched using the number keys when the symbols are not visible.)

TP, 5, to mark BR10 as the first consist locomotive of the new "T5" consist

Scroll in the ***LoR*** to "BR10 red" which second consist locomotive, TP, 5, to mark it with "T5"



TP to switch between consist locomotives

INSERTION: BASIC INFORMATION on DOUBLE and MULTI-FOLD CONSISTS in the ZIMO SYSTEM

(Instructions for setting and removing consists follow on the next pages)

- Double-headers or consists, means linking together two or more vehicles (= addresses). This causes all engines in the consist to react the same when a new speed, direction, MAN or RG command is sent to any one of them. Functions on the other hand are still controlled individually for each decoder in the consist.

- A **consist** in the ZIMO system generally does NOT "reside" (= is created, processed and primarily stored) in the central command station, but in an "operating device", i.e. an MX33 controller (or in ZIMO App) where the vehicles (addresses) of a consist are marked with "Tx" ("x" is a number 1 ... 9 among the consists residing in the device, e.g. "T5", both in the main window and in the loco recall memory "*LoR*" (if visible) and in the object database.

(2) Here, a double traction, in the LoR you can see which vehicle still belongs to "T5", in the example this is "BR 10 Blau"

- However, the consists residing in the controllers are "mapped" **in the central command station** and available **as a copy**. This means that a consist continues to exist even if the "resident device" is removed from the system (or the radio connection is lost).

- A vehicle (an address) that is located in a consist that resides in a specific controller is marked with "FT" ("foreign traction/consist") in LoR; in other controllers; this means: it cannot be activated immediately, but only after a takeover procedure for the entire traction, which automatically includes all vehicles contained therein.

-This takeover procedure consists of the "External traction" window and confirmation of the U key to confirm the takeover of the traction including all vehicles involved, whereby the complete traction resides in the "new" controller.

On another control panel, i.e. non-resident device of traction "T5" of the above example. When trying to activate the second traction locomotive "BR 10 Blau", the message "External consist" and the marking "FT" appears...

- After transferring a consist to a new control panel, this becomes the resident device; there (and only there) the consist can now be cancelled or modified (i.e. vehicles can be added or removed).

-All processes such as transferring consists or deleting and modifying are simultaneously "mapped" in the command station so that it can take its place when the resident device is absent.

➢ *Direction and speed synchronisation among vehicles in a consist*

Softkey III → Show ***LoR*** (if not already displayed); matching only possible with ***LoR***

Scrolling wheel → Select the vehicle to be synchronised

TP-Key on this address → "Consist settings" window with option "... remove" and matching options. ▶

Scrolling wheel → Select the desired matching option

If directional alignment option is selected:

A-Key → Switch identical <> inverted

If **speed calibration** option selected (not yet implemented):

Speed slider → Setting a value between 50% and 150% for the speed of the selected speed of the selected vehicle relative to the speed of the consist itself.

While this adjustment is being carried out, the consist must be travelling at a speed of between 30 and 80.

E-Key → Exit and **save** the settings

➢ *Removing a vehicle from a consist or cancelling a consist:*

Softkey III → Show ***LoR*** (if not already shown); editing and removing a traction is only possible with the ***LoR*** shown.

Scrolling wheel→ Select the vehicle to be removed (i.e. one with T markings and turquoise text).

TP-Key on this address → Window "Consist settings" with option "Remove from consist" and matching options.

When the window is opened, the cursor is normally already on "Remove from consist": otherwise use the scrolling wheel!

TP-Key (again), i.e. a total of **2 x TP-Key →** **Remove this address** from the existing consist, delete the T marker. The ***LoR*** line turns green again. If it was the second last participant in the traction, the entire traction is cancelled.

Possible at any time:

Rocker switch→ switches to the **second level** of the ***LoR*** (with direction and position indicators)

Softkey III, Scroll, TP

Scroll to BR10 Red

TP

TP

"BR 10 Red" deleted from consist, green again
"BR 10 Blue" automatically deleted from consist

➢ *Handover/Takeover of consist between controllers:*

◀ If an address in the *LoR* is marked with "FT (..)", the vehicle can not be activated directly because it is part of a consist in another controller *), which is NOT ACTIVE at the moment (not in that controller's foreground). If, on the other hand, this consist is currently ACTIVE in the other controller (i.e. one of the vehicles involved is in the foreground on that controller), "FS (..)" is displayed in the *LoR* (= external control with the number of consisted vehicles).

*) This is also the case if this other controller, the "owner" of the consist, is no longer connected to the system or the radio communication is lost.

These "FT (..)" or "FS (..)" markers are also shown in the main address field (after the vehicle is brought to the foreground with the **A-Key** from a *LoR* line or via the **F-** and **U-key**) together with the corresponding note
"External Consist" or "External Control" in the header.

If a takeover attempt of such a vehicle is made (by pressing a function key or changing the speed), the header closes and in its place the WINDOW

"External Consist" or "External Control" opens

◀ with these options:

U-Key → the consist (= with all participating engines) is taken over and controlled by the new controller. At the same time the **ownership** **) of the consist changes from the old to the new controller, and all vehicles involved *) receive a mark such as "T1", "T2" ... ", while on the old controller these addresses are changed to "FS (..)".

*) Addresses that belong to the consist but have not yet existed on the new controller will also automatically be included in the *LoR* (with the markers "T1", "T2" ... ").

**) A consist can only belong to one user; this one can be changed; the original creator does NOT matter.

A- Key → the consist will NOT be taken over, the controller once again displays the header as before.

◀ After the consist has been taken over, all the addresses involved are located in the *LoR* of the "new" controller and marked with "T1", "T2" etc., whereby the system attempts to reassign the previously used T-number (if free).

➢ *"Move" consists to the system center (central command station)":*

Consists that were created in a controller (see previous pages) and reside there, can be "moved" to the central command station, i.e. the "image" of the consists that was already present in the command station becomes the actual residence of the traction (which was previously a controller).

This is useful in some cases, for example to free up one of the consist numbers (T1 ... T9) in the controller and thus create space (there is only T1 ... T9) for new consists to be generated.

On the MX10 / MX10EC command station, the existing consists (both those in connected controller and those residing in the command station) can be monitored and deleted via the menu item "Object database consists", but not edited. To make settings to consists, the consist can (must) be recalled from a controller (see below).

Softkey III → *LoR* (if not already displayed)
displayed); we recommend this procedure in the *LoR*
Scrolling wheel → Select (any) vehicle from the consist that needs to be moved to the command station.
C-Key on this address → "Move consist" window,
options A key = yes / E key = no

If move is initiated: all vehicles in this consist disappear from *LoR*; the previous consist number becomes available for new consists.
The addresses labelled TZ remain in the *ObjDB* object database. The addresses can be deleted from the object database as usual (each one individually), which does not change anything operationally; the deleted addresses can then be found under the grey lines (because they exist in the command station).
The consist remains in the central command station with all the matching data and is operated from there.

➢ *Recall of tractions that have been "moved" to the central command station:*

The process is very similar to taking over a consist from another controller, i.e:
Activate one of the addresses of the "moved" traction
in one of the usual ways, i.e. using *LOCO IN* or from
the *ObjDB*
→ Window „Consist in Command station ",

Options A-key = Hide (no transfer),
U-key = Accept ALL vehicles

➢ *Connecting or reconnecting a controller with consist:*

If an operating device is not permanently in contact with "its" base device ("home system") (cable or radio), but is temporarily disconnected or is only connected to the system later, there may be inconsistencies between the consists stored in the controller and in the command station.
A consist can be taken over by another remote during the absence of the resident device; this becomes the new resident device without the previous one knowing about it. In itself, it is irrelevant whether the consist is modified or deleted by the new remote: at most, it would have to be taken over again from the original resident device (possibly modified in the meantime).

➢ *Connecting a controller with consists to a non-home system:*

If a controller is connected to a command station that is NOT the home system, the consists created or present are NOT PERMANENTLY mapped in this command station even after successful activation, i.e. copies from the consists of the "non-home device" are only present in the command station as long as the contact to this device is not lost for more than 10 minutes.

10. STOP, OFF, SHORT CIRCUIT...

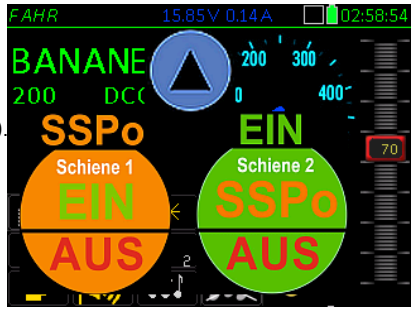
- ***SINGLE STOP*** (also known as "Emergency Stop "); only stops the engine (or consist, train) currently under direct cab control and is meant to stop the engine rapidly, if the normal braking would take too long due to the set momentum in the decoder (CV #4...) and/or system.

- **BCS = BROADCAST STOP** (a standardized NMRA-DCC command on address 0); this command stops all vehicles but the decoders retain the last received data (i.e. lights remain lit) and it is also possible to switch them on and off.

- ***OFF (Track Power)*** this is the fastest and safest type of a broadcast stop (independent of data reception). However, the decoders lose all data and the stopping distance is heavily dependent on possible built-in capacitors (GoldCaps), the stops are sudden (which often leads to derailments) and restarting a larger layout may take some time.

NOTE: in ***CONFIG*** (accessible with ***E+0***), (menu "STOP&OFF") there are selection options, whether SSP (S button) or OFF (H-S) is effective on a track connection 1 or 2 or on both, whether BCS mode (S button) should mean **"Emergency Stop"** (quick stop without braking curve) or **"SS 0"** (= normal braking curve with CV #4 etc.). ATTENTION: a sudden "Emergency Stop" can lead to derailment.

H ("Halt"-Stop) → SINGLE STOP, FAST STOP ▶
(*=emergency stop for the active engine (train), the green speed step indicator turns red; the speed slider must be moved to 0 in order to restart the engine later*)
and simultaneously → *displays the* "**STOP Bullets**"
(remain for 3 sec)



S → BCS (Broadcast Stop) on Track-1 or -2 ▶
Stopping of all trains on track 1 or -2 or
track -1 and -2 (depending on the specification in ***CONFIG***).
The STOP balls are displayed at the same time.

Touch one of the ball halves ON, BCS, OFF

→ rail 1 or rail 2 of the base unit is set respectively.

The STOP-balls appear equally on all travelling devices (if at least one track connection is **BCS**, **OFF**, **OVC**)

Touching the blue arrow icon (pointing up)

→ **Removes the "STOP Bullets"** ▶
i.e. the blue arrow now appears in the header

When the STOP balls are hidden, the screen content behind them (partly hidden) becomes visible and usable.

However, the background is not active when the STOP ball is displayed.

When the STOP Bullets are replaced by the small blue arrow in the header (pointing down):
Touching the small blue arrow (in the header)
→ **reopens the "STOP Bullets"**

H ("Stop") -S (press simultaneously or in quick succession) → **OFF** Track-1 or -2 or both
Switch off the track voltage on Track-1 or -2 or both Track -1 and -2 (depending on the entries in ***CONFIG***).
H - S (simultaneously) is the fastest way to switch off the track voltage, however, the same result can also be achieved by pressing S (alone) and then tapping OFF the STOP ball in this case.

↑ (Shift) -S (press simultaneously or in quick succession)
→**ON** rail-1 or - 2 or both
↑ S is the direct method for switching back on; the same result can be achieved with the STOPP balls.

OVERCURRENT (SHORT CIRCUIT & UNDERVOLTAGE:

- **"OVC" = OVERCURRENT (Short);** the "OVC" message indicates that the maximum current defined in the command station (identified as "SHORT threshold" in the MX10) has been reached for the specified output, and the track output (1 or 2) has been turned off; therefore the message "1:SHORT" or "2:SHORT" is shown in the stop bullet (top).

- ***"UND" = UNDERVOLTAGE***; the "UND" message is displayed, not because the overcurrent threshold was reached, but because the desired track voltage could not be maintained and the track power had to be turned off. This usually happens because of a weak power supply and the current thresholds for the track outputs are set too high, which causes the power supply to overload or overheat after reaching a specific current level, and therefore lowers the output voltage (= primary voltage for the command station).

Overcurrent (Short circuit) on track-1 ▶
(Track output of the command station is turned off; there is NO automatic power-on) at the same time opens the "**STOP Bullets**" with **OVC** on top.

Tap on "ON"
of track 1 (left)

If both tracks are ON again :
StopTouch field disappears after 1 sec.

The overcurrent handling of the MX33 is principally the same as for STOPP & OFF: the two track outputs (if two are present, i.e. MX10) can be turned on again independently by touching the corresponding fields **ON** (turns track power back on) or **BCS** (=Broadcast stop), provided there is no longer a short circuit present.

Not available in the case above is ***"OFF"***, because "OFF" has the same effect as "OVC", both means "Track switched off".

Undervoltage at track-1
(Track connection is turned off; there is NO automatic power-on) at the same time opens the "**STOP Bullets**" with **UND** on top.

As in the "UES" case ... (see above)

11. External and Computer Control

- ***External Control*** ***or*** **Assign address** (appears in the header or window with take-over option with the **U-Key**) means that the vehicle (train...) in the display is currently controlled by another controller (hence external control); The speed slider and function keys have no effect but the screen and LED's record changes made by the other controller.
  A TAKEOVER can be forced from the "externally controlled" control panel: if the "External control" window (or "Address in use") is visible, by pressing the U key; if "External control" is only visible in the upper bar, the "External control" window must first be opened by pressing the RG key (or any other key), or by activating the controller, the "External control window" must be opened beforehand
- ***Computer control*** appears in the header (or the name of the computer program may be shown instead, i.e. **"STP"** or **"ESTWGJ"**), indicating that the current vehicle is controlled by a computer (speed, direction, functions) via USB or LAN or from an App...
  In contrast to an "external controller" (external control), a computer does NOT have to go through a takeover procedure in order to send commands to the vehicle; such commands will always be executed and the changes are shown on the cabs, along with the message "Computer control". **Neither** does the controller have to go through takeover procedure to take control over the vehicle again. BUT: although a vehicle is under computer control, an operator can still send commands directly from the controller: function commands can simply be added (without losing computer control); when speed and/or direction changes are made, the message "Computer control" will be deleted; the computer recognizes when the changes are made and can take over control again if needed.

If an attempt is made to activate a vehicle address from the ***LOCO IN*** page or from the ***LoR*** memory **and this** address is already active on another controller, the

"Address in use" window appears. ▶

U-Key→ address takeover; "steals" the address from the other controller, which in turn displays "External control" in its header.

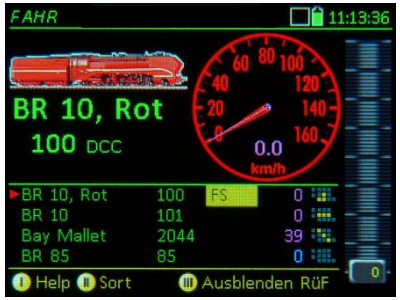
A- Key → closes the "Address in use" window without taking over the address. The controller then moves the label "External control" to the header.

The latter (**A-Key** action) also happens automatically after a few seconds (Time-out).

A

U

↑ - U



◀ If "External control" is displayed in the header bar the takeover is NOT possible with the U-Key

↑ *(Shift) - U-key* → *Direct transfer of the address*
(↑ forward, hold) address (without window)
the U key acts contrast to the case of a "Remote control" or "Assign address" window give address" window, it activates a vehicle from the ***RüF***.
However, if (for example accidentally) displayed "External control" of the speed sliders or a function key is pressed, or MN or RG

any key *)→ Transfer window opens

*) Pressing the speed slider has the same effect!
In this window, the keys (as described above) can be pressed, i.e.

U key → Accept address, i.e. "take away" from the other controller; this other controller
itself changes to "external control" (indicated shown in the upper bar).

A key→ the address should NOT be taken over
the address should NOT be taken over, but the own remote control" (but "only" in the upper bar).

A special case when pressing the **S-Key** with "External Control" or "Address in use" displayed;
In order to initiate an emergency stop even for such an "inactive" address (without going through a takeover procedure first), the **S-Key** is effective nonetheless and does a takeover at the same time:

With "External Control": S-Key→ EMERGENCY (SINGLE) STOP & TAKEOVER

S

The **S-Key** can also be used as a quick alternative to take over a loco when "External Control" is shown in the header, but the **train IS NOT moving** (i.e. speed slider at 0): Press the **S-Key** (has no effect since the train is already stopped) but the "External Control" is removed and the controller is fully functional (the address is deactivated on the other controller and "External Control" is being displayed there).

Programs (or Apps) on computers, tablets, smartphones connected via USB or LAN can intervene at any time in the control of vehicles (and, of course, accessories). No takeover procedure is necessary even if the address is actively in use by a controller. The control activities (changes in speed, functions, turnout positions etc.) are shared with each other and displayed.

To inform the user that a computer, cell phone etc. takes temporary control of a loco address, the MX32 displays "Computer control" (in the header) or – if known – the name of the application. For example Roco Multimaus or App (X-Net Device):

This image remains until a new command is sent by the controller.

12. „Trains"

Currently not implemented!

13. Operating mode SWI

Object-oriented control of turnouts, signals…

The following description of the SWI operating mode has in principle already been valid for many SW versions; a general revision and extension (e.g. to StEin and switch routes) in accordance with these instructions will take place in SW versions 1.27.x and 1.28.x.

SWI is one of two operating modes for accessory items, known as the object-oriented control of **turnouts** and **signals**, and also for forming and operating **turnout routes** (the last of these from a SW version in 2020).

W-Key → *SWI* operating status (i.e. upper half of the screen remains vehicle, lower half SWI panel)

See also:

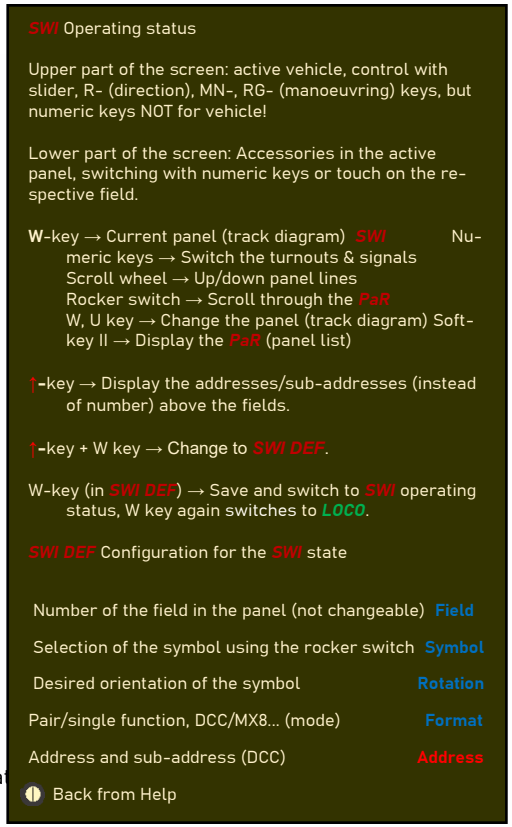
E + 3 → ***ACC LIST*** *(top half of the screen vehicle, bottom half accessory addresses)*

….. or **Softkey II** (only from ***LOCO***)

Die ***ACC LIST*** is the alternative operating state for turnout setting (see the chapter ***"ACC LIST"***) - specifically the "classic" address-orientated control (selection of accessory addresses, switching of sub-addresses, which can also be used temporarily within SWI to insert panel addresses).

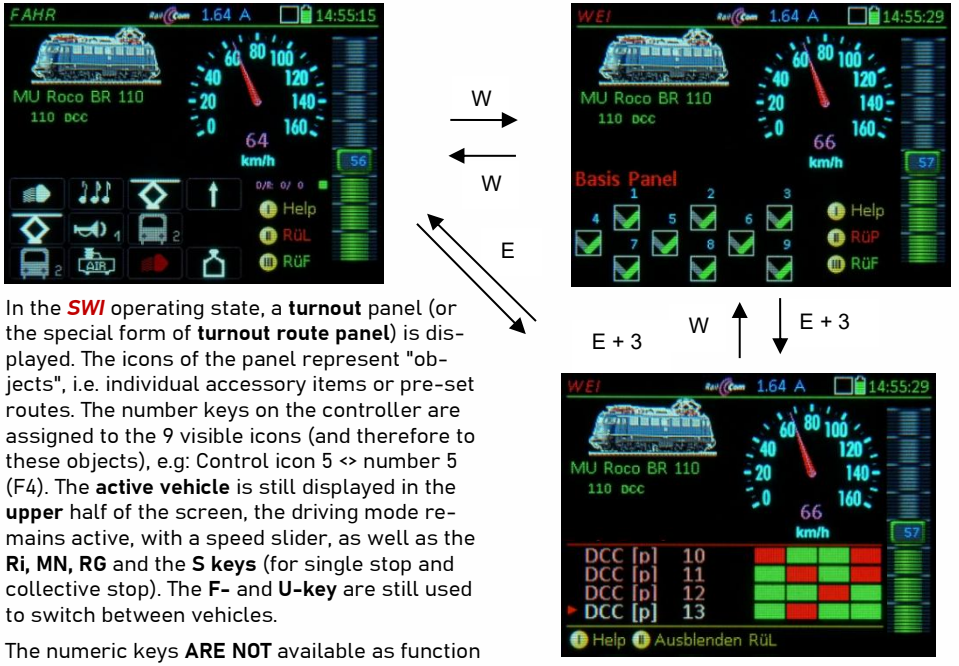
The Help-File for ***SWI*** ▶

No complete help file available.



W-Key (among other things from the operat

When you first enter ***SWI*,** the lower half of the screen shows the **"basic panel"** consisting of 18 icons (depicted with "V-icons" *), of which the first 9 icons are visible.
The "V- icons" used as standard can be interpreted as general left-right switches or as red-green signals. They are NOT the real prototypical switch panel field.
In the ***SWI DEF*** definition screen, the V-icons can be replaced by other icons; see below.



In the ***SWI*** operating state, a **turnout** panel (or the special form of **turnout route panel**) is displayed. The icons of the panel represent "objects", i.e. individual accessory items or pre-set routes. The number keys on the controller are assigned to the 9 visible icons (and therefore to these objects), e.g: Control icon 5 <> number 5 (F4). The **active vehicle** is still displayed in the **upper** half of the screen, the driving mode remains active, with a speed slider, as well as the **Ri, MN, RG** and the **S keys** (for single stop and collective stop). The **F-** and **U-key** are still used to switch between vehicles.

The numeric keys **ARE NOT** available as function keys with this display because they will be used to switch the turnouts and signals.
However, switching to normal driving mode is quickly possible using the W key.

◤ ▲ Reversing routes between the "normal driving mode" ***LOCO***, operating mode ***SWI*** and ***ACC LIST*** states.

Number Keys **1 … 9** → Switching the visible icons.
The 18 icons of the "basic panel" are assigned the addresses 10.0, 10.1, 10.2, 10.3, 11.0, 11.1, 11.2, … i.e. the accessory item addresses from 10, each with all 4 sub-addresses (0 …3).

↑ (Shift) Displays the addresses of the accessory items instead of the field numbers (next page image)
Scrolling wheel → Scroll within the panel (here: to see more than the first 9 icons)

The "Basic Panel" is often sufficient for **smaller applications** (for up to about 30 turnouts), by adapting it o the actual accessory items on the layout. In the simplest case it is only necessary to change the decoder addresses in the respective fields and to extend the panel by adding additional icons (see ***SWI DEF***).

For a better overview over **larger layouts,** it is often better to distribute the accessory items to several newly created panels with meaningful names, rather than using the "Basic Panel", i.e., : "Hidden station", "Station 1"…etc.

↑

HINWEIS: in ***CONFIG*** ("this controller) under "Operating status WEI", the permanent display of the address can be set instead of the number.

Display of addresses and ▲ Sub-addresses instead of field numbers

➢ ***SWI DEF*** *for directly typing in the addresses for accessory items of the "Basic Panel": Other parameters (in addition to the addresses): see section "**SWI DEF***

➢ *for general insertion..."*

The aim here is to use the standard "Basic Panel" to put turnouts or two-letter signals into operation. The standard "V icons" and their orientation should therefore be retained. Only the accessory addresses assigned by default (10.0, 10.1, ...) must be replaced with the desired ones and/or supplemented (additional fields).

↑ (Shift) **+ W +** ↑ (Shift) → Enter ***SWI DEF***, whereby the second ↑ (Shift) immediately jumps to the line for entering the addresses (and sub-addresses) and the "Scroll line" option is activated (whereby "Scroll all" is displayed for the U key, for - if desired - switching back to "Scroll... all"). Now the addresses and sub-addresses for all panel fields can be typed in one after the other, with the A key in between; or scroll to a more distant field. So typical input sequence:

Address - A – Sub-address - A - Address - A – Sub-address - A - Address - .....

↑, W, ↑

23, A, 0, A, 23, A, 1, A, 24, A, 3, A, ...

W (here in the picture are the transferred new addresses NOT visible seen, because numbers)

After entering ***SWI DEF*** the input blocks for the first three fields (turnouts) are displayed,

Scroll wheel→ Continue scrolling through the fields (in the panel AND in input blocks)

Termination of ***SWI DEF***:

W-key (Save) → End,
Take-over of the newly entered data

E (ESC) → Security question with selection of whether Discard data or save after all

➢ ***WEI DEF*** *with displayed **ACC LIST** for semi-automatic insertion via pattern-Switching the addresses of turnouts (NOT signals) in the panel field:*

Instead of directly typing in addresses, sub-addresses, module numbers, etc., the data format and other information on the connection point can be learnt via pattern switching, i.e. the points (or other objects) are switched in the ***ACC LIST*** display and thus automatically assigned to a specific panel field. This works for turnouts that are connected to accessory decoders or to StEin-modules.

TP-Key → Display of an ***ACC LIST*** in the lower half of the screen, which contains all accessory addresses (in their format) and StEin modules (in the StEin list format, see chapter 20) registered in the controller. The turnouts (and other objects) can be switched there. The ***SWI DEF*** input blocks remain in the upper half of the screen.

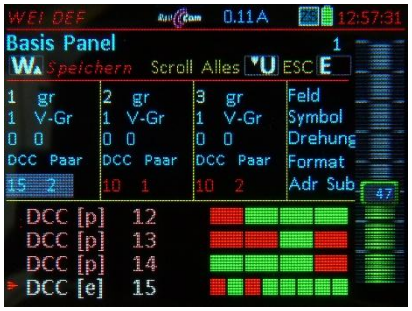
NOTE: the temporary display is limited to the lower half of the screen (NO tap to full screen)

Number keys, scroll wheel→ are now (after TP key) NOT active for typing in the parameters, but for switching the turnouts (or similar) in the ***ACC LIST***. However, the data (format, address, ...) of the switched turnout is immediately entered in the corresponding fields in ***SWI DEF***.

A-Key → is still assigned to the ***SWI DEF*** area, i.e. the upper half of the screen: this allows you to switch from input block to input block, which means that several turnouts can be assigned to a panel very quickly.

E-Key → Return to the "normal" screen layout for ***WEI DEF***, i.e. three input blocks in the upper half of the screen and the panel itself in the lower half.

TP

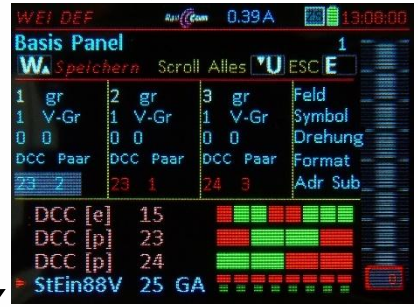


Scroll

Number keys:

Switch the turnouts, A,

Switch the turnouts, A,

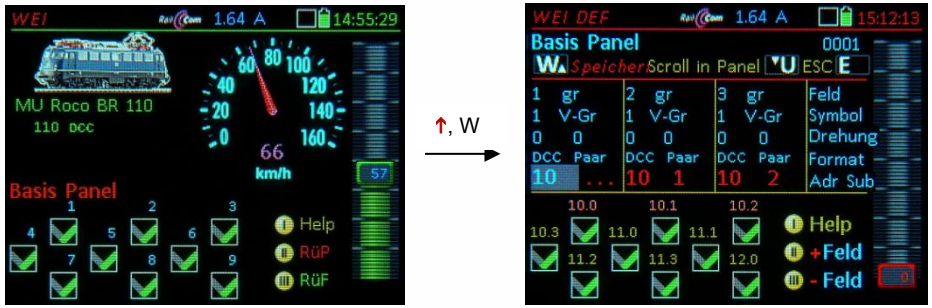


At the end of the ***ACC LIST*** - Entries from the StEin list StEin modules.

SWI DEF *for* ***the general setting of ALL parameters*** *for the panel fields:*

While the two previous sections ("to type in ..." and "... to insert ...") deal exclusively with assigning the addresses of accessory objects, the following describes the insertion of all parameters in panel fields. The addresses are also part of the parameters, i.e. the procedure described in the previous sections can be used more efficiently in many cases, but is otherwise included in " ...general insertion ..".

↑ (Shift) **+ W** → Entry in ***SWI DEF***



In ***SWI DEF***, three input blocks are assigned to three panel fields (i.e. one line of the panel). The number at the top left of the input block shows the field number, i.e. the panel field for which changes are currently being made.

In each of the input blocks, the symbols can be selected, their rotation (= orientation), the format (DCC, StEin, ...), the associated addresses / subaddresses of the decoders (or numbers of modules) can be entered or edited in the respective lines (column on the far right).

Scrolling wheel → Select the individual input fields for the parameters; when the ends of the input blocks are reached (bottom right or top left), a jump is made to the next block or the following or preceding panel line.
At the end of the panel, 3 fields are automatically added when **scrolling over it!**

U-Key → Change the scroll path to "in-line mode": when scrolling, the marker then moves from field to field in the current line, for example in the address line: this allows the decoder address and sub-address to be entered easily for each field without having to scroll across all other parameters.

Back also with the "U key"→ **„scroll in block“**

Softkey II - + Field → adds an additional icon field before the currently marked field; the marked and subsequent fields are moved down automatically.

Softkey III - - Field → deletes the marked field (icon) from the panel; the subsequent elements move up automatically.

Number keys, C-Key → deletes entries or individual characters in the value entry fields.

Rocker switch → select from several fixed values for input fields that don't require continuous numerical values, rather predefined values; in the cases “Icon” and “Rotate”, also numbers are placed ahead of the value field, so that numerical values (with number keys) or predefined values can be entered (i.e. 0, 1, 2, 3.... or 0, 90, 180, 270 degrees for icon rotation). If the number (e.g. of a icon) is already known from previous entries, the entry is of course faster.

After scrolling through the respective input fields: Input via number keys or selection (rocker switch):
The meaning of the individual parameters in the input blocks:

Number of the icon in the panel (Object) belonging to this block (this number cannot be changed).

`1 gr | 2 rg | 3 gr FIELD`

Color for the **function key LEDs** that indicates the turnout or signal position: r = red, g = green, y = yellow (the color is predetermined for these icons and don't need to be changed in most cases); Select with rocker switch.

Icon design number (if the number is known for the desired icon it can be entered directly to this field, using the number keys).

`5 WLA-li | 2 V-inv | 6 WSF-re | Icon`

ONLY ONE PARAMTER at a time, which can be entered using the number keys or selection (rocker).

Description of the desired **icon** for the panel field (permanently linked to the number on top); immediately visible in the corresponding panel field; Select with the rocker switch.

Rotation angle (icon orientation, the numerical entry of the desired angle is also possible).

`2 180 | 1 90 | 0 0 | Rotate`

ONLY ONE PARAMTER at a time (as above)

Description of the desired **rotating angle** (for certain icons also **colour**) for the icon; immediately visible in the corresponding panel field; select with the rocker switch.

Data format (DCC, MM, MX8, StEin, Q-Dec, Z-mode0, Z-mode4); Select with rocker switch (if already implemented in version).

`DCC Paired | DCC Single | StEin Paired | Format`

In this line TWO SEPARATE PARAMETERS (not number for selection as in upper lines)

Paired, Single function or **Sequence**, for is turnouts or red-green signals (paired functions), light bulb (single function) or multi-aspect signals (to call up sequences of 3, 4, 5... signal aspects one by one with one key). Select with rocker switch (“Paired”, “Single”, “Seq-3”, “Seq-4”...”Seq-8”).

For this input field an automatic suggestion is made on the basis of the icon selected above, a change is usually not necessary.

Enter the **accessory decoder address** (1...511), MX8 address (0 ... 63), or StEin address

`10 0 | 10 1 | 12 3 | Sub Addr`

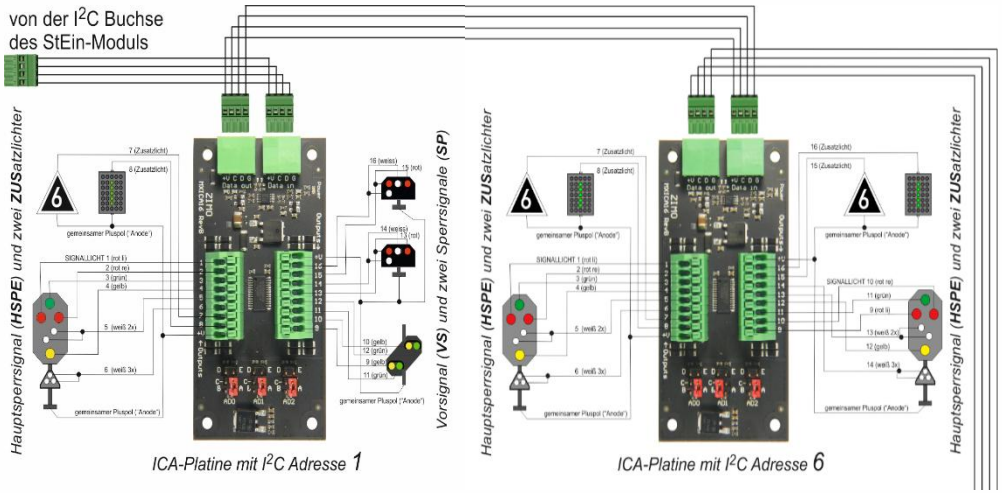
In this line TWO SEPARATE PARAMETERS (not number for selection as in upper lines)

Enter the **sub-address** of the accessory decoder address or the respective MX8.

➢ ***SWI DEF*** *for* ***manual*** *or* ***semi-automatic setting*** *of parameters for* ***signals****, which are connected to* ***StEin modules*** *via* ***ICA boards*****:**
Semi-automated not yet implemented (December 2020)

Light signals from signal boards (=ICA boards) are operated on the StEin, which are supplied and controlled by an I²C bus of the StEin; please see the StEin operating instructions:

Based on the loaded "parameter sheet", each StEin module knows where which signals are connected, which signal patterns exist, etc. In addition to manual input, it therefore makes sense to retrieve this information directly from the Stein modules via the CAN-Bus cable.



Either all lines can be filled in completely (starting with "Symbol") or only the connection location of the signal can be specified, which is agreed by specifying the "first wire" - this is generally the red light (or the "strongest" stop, e.g. the two yellow lights of a distant signal. The ***SWI DEF*** lines "Symbol" are optional, the lines "Format" and "Addr Sub" are to be filled in as follows (slightly different from the information on the previous page):

Data format (Select with rocker switch): **StEin**

DCC Paired	StEin ICA	StEin ICA	Format	In this line TWO SEPARATE PARAMETERS

Second Parameter (Select with rocker switch): **ICA**

first and second Parameter:
StEin-Number (1 ... 99), ICA-Boards-Number (1 ... 12),
Input via number keys

10 0	25 6 13	28 11 5	Adr Sub	In this line THREE SEPARATE PARAMETERS

Third Parameter:
Connection point "first wire" of the signal on the ICA board (1...16),
Input via number keys.

If "Symbol" is not specified, the control panel automatically queries the StEin modules and receives the applicable symbol and the signal images from there, which are to be displayed by the respective number key (icon number) when switching the signal (sequentially one after the other). *)

*) The prerequisite for this is, of course, that **the relevant StEin modules are present** and loaded with the correct configuration; whether the signals are actually connected, however, is irrelevant. Otherwise, "???" is automatically written in the "Symbol" field.
Each Stein module **automatically** reports the data of the signals at power-on (i.e. at the same time as busy signals, switch positions, etc.) or when the configuration is changed.

Automatic entries in panels or creation of entire panels from the controller:
In the SIG object class - object lines for signals - there is a PANSYMB field where the name of the icon to be displayed in the controller, can be entered. This controller icon (see list below) is NOT identical to SIGTYP because only a limited number of icons are available on the controller. SIG objects also have the parameters PANEL and PANFELD, which are used to specify a specific position in the designated controller via the StEin configuration.

The list of icons shown below contains not only the signals (these start with no. 19), but also all icons (including turnouts and single lights) that are available for selection on the controller or that can be requested from StEin. It does NOT have to match the icons actually available in the controller, because ongoing development is under way.

	Symbol	Beschreibung	Anschlussfolge	Vorgaben	
1	LR-sw	Leerfeld-schwarz	0	-	
2	LR-gr	Leerfeld-grau	0	-	
3	V-Gr	V-Symbol „Grün“	2	Paar	
4	V-Rt	V-Symbol „Rot“	2	Paar	
5	W-Li	Einzelweiche links	2	Paar	
6	W-Re	Einzelweiche rechts	2	Paar	
7	W3	Dreiwegweiche	3	3 Stell	
8	WV-Li	Gleisverbindung links	2	Paar	
9	WV-Re	Gleisverbindung rechts	2	Paar	
10	WHS	Hosenträger	2	Paar	
11	EKW	EKW	2	Paar	
12	DKW	DKW	4	4 Stell	
13	ENTK	Entkuppler	1	Einzel	
14	WSTR	Weichenstraße	-	-	
15	L1	Einzellicht	ein Anschluss	Einzel	
16	S2	Rot-grün Signal	rot - grün	Paar	
17	S3	Rot-Grün-gelb Signal	rot - grün - gelb	3 Bild	
19	DEBL	HV Blocksignal	rot - grün	Paar	
19	DEHS	HV Hauptsignal	rot - grün – gelb	3 Bild	
20	DEHSP	HV Hauptsperrsignal	rot li - rot re - gn - gb - ws	4 Bild	
21	DEVS	HV Vorsignal	gb li - gb re - gn li - gn re	3 Bild	
22	DESP	Sperrsignal (Rang, Zwerg)	ws Halt - ws Fahrt	Paar	
23	CHSM	Signal, schmaler Schirm	4-5 L	5 Bild	
24	CHBR	Signal, breiter Schirm	6 L	6 Bild	
25	CHVS	Vorsignal	5 L	4 Bild	
26	CHKS	Kombinationssignal	8 L	8 Bild	
27	CHZW	Zwergsignal	2 L	Paar	
28	ATBL	Blocksignal	2 L	Paar	
29	ATEF1	Einfahrtssignal-1	3 L	3 Bild	
30	ATEF2	Einfahrtssignal-2	4 L	4 Bild	
31	ATAF	Ausfahrtssignal	5 L	5 Bild	
32	ATVS	Vorsignal	4 L	4 Bild	
33	ATRG	Rangiersignal	2 L	2 Bild	

Display of the icons in the panel: in miniature form within the panel; a **large display of the icons** in the area (instead of) the softkey icons is shown in the following cases:

- in ***SWI DEF***: when assigning to a field, in each case after pressing the rocker switch to select or enter the field number numerically, until the next time the rocker switch is pressed or after 3 seconds have passed.

- during operation: after switching the turnout, signal or other accessory item, for 3 seconds or until the next switching operation.

Example of a modified panel with different symbols; in the first image panel lines 1 to 3 (fields 1 to 9), in the second image scrolled down by two lines to display lines 3 to 5 (fields 7 to 15).

NOTE (not yet implemented):

Signals at the light outputs of the MX820, MX821 decoders in control mode = 4

The CV values in the decoder that match the icon should be programmed automatically. In control module 4, however, the programming goes to the decoder address, which is not the same as the object address of the signal, although the object address is also programmed. Up to 8 signal patterns can be defined in the MX820, the sequence must be specified based on signals in the StEin (ready configurations), as no communication between MX32 and MX820 is possible for configuration.

Selection between panels and new panels in PaR (recall memory panels)

(LIR is actually a panel list; "recall ..." is an analogy to the recall memory for vehicles)

Softkey II (*PaR* FROM *SWI,* NICHT aus *LOCO*) → Displays the list of existing panels, Scroll, select (A), delete (C), see below for details.

Tap on the small PaR list to switch to the full-screen view.

Softkey II →

Tap on *LIR* List

Scrolling wheel → Select a line of the *PaR* list

A-Key (or **W**) → Activate the selected panel,

C-Key → Delete the selected panel from the the *PaR* (not from the object database!),

U-Key → Edit the panel name,

Softkey III → Sort: move the selected line with the scroll wheel to the desired position, again

Softkey III → to position the line.

Creating of a new panel in the PaR:

Scrolling wheel → on the **">NEW<"** line at the end of the panel list,

A-Key → Activate the procedure for new panel, automatically switching to the full screen view. The name is typed alphanumerically in the empty field: Number keys as SMS keyboard, with ↑ (Shift) character switching ABC, abc, 123.

The new panel initially consists of standard "V icons" like the "basic panel" in the initial design. Other icons (in the example WLA - switch lanterns) have already been set for the image on the right.

A "?" next to the icon indicates a new undefined field that can be filled and edited in the definition screen *SWI DEF*; see section above *"SWI DEF general modification of the parameters of the panel fields"*.

Programming of turnout routes for the "basic rail panel"

Not yet implemented

Softkey II *PaR* → Show the "List Recall memory" (list of existing panels),

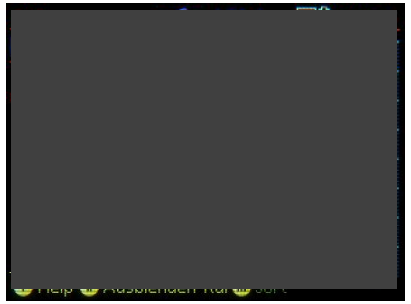
Scrolling wheel → to the "Basic Rail Panel",

A-Key → Activate the selected panel (in this case rail panels)

The rail panel looks like a "normal" panel (for turnouts and signals), but is assigned "rail symbols" by default. A turnout route can be programmed for each of these "rail objects".

However, "WEISTR" rail icons can also be defined in each panel field (regardless of the name of the panel), even mixed with individual signals or turnouts.

Scroll to the **„Icon" Line + W-Key**
→ Enter *STRA DEF* and open a list of turnout addresses (partially prefilled as an example) for a turnout route.



1	DCC	Pair	10	0	300	ms
2	DCC	Pair	10	1	300	ms
3	DCC	Pair	10	2	300	ms

etc.

This list can now EITHER

- can be edited directly (by overwriting the prefilled values with your own)

OR

- by test switching as with the objects:

TP-Key → Temporary display of a combination of *ACC LIST* and *StEin LIST*

There, the respective objects (turnouts, signals, ...) are switched as required using the number keys and scroll wheel (to change between the lines) (also several times back and forth, on changing addresses, ...) until the correct turnout is found. Then

A-Key → Transfer of the last switched turnout (or accessory item) to the rail by automatically entering the address/sub-address (or number) in the list,

TP-Key → *End* of the temporary display *ACC LIST / StEin LIST*, back to *STRA DEF*

The procedure for extending the road can be repeated.

OR

- by test switching from an already completed panel:

Softkey II PaR → **Temporarily show** *PaR* list (the existing panels)

Scrolling wheel → to the desired panel (panel with objects only or mixed panel),

A-Key →Activate the selected panel

Switch any number of turnouts within the panel, only the last turnout and signal activations are noted for the rail

TP-Key → End of the temporary display of the last panel, back to *STRA DEF*

The procedure for extending the rail can be repeated

The resulting list can then be edited again directly, for example the automatically assigned waiting times (usually 300 ms) can be changed.

C-Taste→ on the corresponding line → Delete the line

14. ACC LIST

Address-oriented control of turnouts, signals … *and* *OP PROG*

… is the alternative for operating the accessory items (formerly magnetic items) to ***SWI***, based on the usual control in earlier system generations. This display also provides access to *OPERATIONAL PROGRAMMING* (i.e. CVs on the main track, POM).

The ***ACC LIST*** can only be reached from the driving mode ***LOCO*** (NOT from the ***SWI*** mode) with

Softkey II LiR→ Opens the ***ACC LIST*** in the lower screen half,

o r (from ***LOCO*** and also from ***SWI***)
E-Key + 3 → ***ACC LIST***

*) **LiR**: List (Acessory) Recall Memeory

Accessory decoder addresses can be entered in the operating mode ***ACC LIST*** and associated turnouts and signals can be operated, or existing addresses (entered automatically, see below) monitored and operated when needed.

The function keys do NOT operate loco functions as long as the ***ACC LIST*** is displayed. However, the active vehicle in the upper screen half can still be controlled (Speed, direction…).

All addresses assigned to ***SWI*** panels are automatically added to the ***ACC LIST,*** but can be removed from the list if they are not desired here without changing the functionality of the panel.
To add a **new** (additional) **address** it is necessary to scroll to the line **›New‹**, then:

- **Scrolling wheel:** Move the cursor to the line **›NEW‹**
- **A-Key:** "DCC [p]" (DCC with 4 paired functions per address (=p)) is the standard definition, if changes are required such as single functions proceed with the
- **Rocker switch** to change to:
  "DCC [e]"= (8 single functions per address), often for double-slip switches
  "DCC [x]" =(extended)
  "MM1 [e]" = MM (also Motorola data format; 8 single functions per address)
  "MM1 [p]" = MM (also Motorola data format; paired)

- **A-Key→** confirms and opens the accessory address input field, then use
- **Number keys→** to enter the new address and confirm with the
- **A-Key→** saves the new entry (address) and adds it to the ***ACC LIST***.

Already **existing addresses** can be **edited** the same way (scroll to line, A-Key…)

Operating accessories (only if ***ACC LIST*** is open):

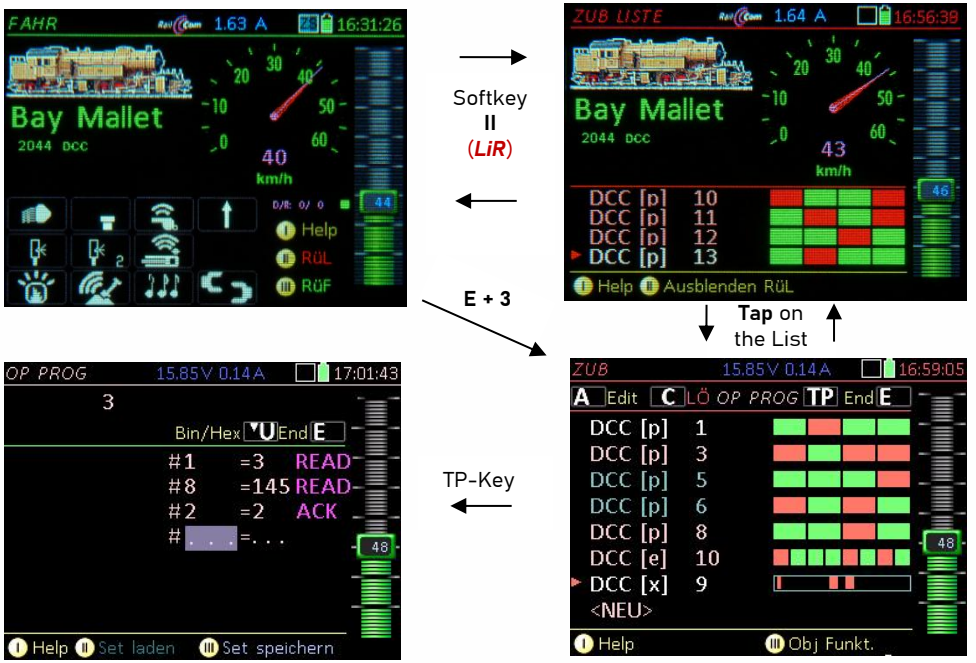
- **Scrolling wheel→** scrolling up/down in the accessory list
- **Number keys** 1 – 4 or 1 – 8→ operate 4 turnouts (or 8 single functions) …

DELETE entries and **EXIT** this operating mode:

- **C-Key→** deletes the highlighted line from the ***ACC LIST***
- **Soft-Key II→** Closes the ***ACC LIST*** and returns to ***LOCO***.

IF ***ACC LIST*** ONLY in full screen display (open by touching the bottom half of the screen):

- ***TP-Key →*** *OPERATIONAL PROGRAMMING (POM)* for CV programming on the main track



Alternative, quick method of entering a new accessory address:
Directly from the operating screen ***LOCO***:

A- Key → ***LOCO IN***

"Misuse" of the ***LOCO IN*** screen:

screen: Enter the desired address, then

W- Key → confirms and jumps directly to the address in the ***ACC LIST*** = operate the accessory or use the

A- Key → for editing (i.e. to change to "DCC [e]" or change the address), then

Softkey II → ***LOCO***

15. StEin LIST *(also control and programming of the "old" MX8, MX9 modules)*

Monitoring and operating stationary equipment, that are connected to StEin-Modules (MX8, MX9: see the next page)

ZIMO "StEin" modules used for stationary equipment of the model railroad layouts, especially **track sections** (with occupancy and overcurrent detection, HLU, RailCom), **accessory items** like turnouts, signals, uncouplers, inputs (track contacts, reed switches, photocells...), as well as **point detector** (contact tracks, switch tracks, light sensors) and **speakers.** (which play the integrated sound generator).

The ***StEin LIST*** is accessible from the operating mode ***LOCO*** or ***SWI*** with the:

E- Key + 8 → ***StEin LIST***

When the ***StEin LIST*** is called up, all ***StEin modules*** connected to the CAN bus are searched for and listed, each represented by one line. They are sorted according to module numbers, which were previously entered on the ***StEin*** modules themselves (to be seen there with illuminated numbers or later small display).
Presentation either in the lower half of the screen or in full screen display (see right)

The **AOS inputs and outputs** of the MX10 are "also" shown in the ***StEin LIST***.

↑ (Shift) – **Key** (Short) → **Switch module line** between the displays for
GA: Track sections | **WE:** Switches & signals | **IN:** Switching inputs.
The displayed elements are activated by the numeric keys.

↑ (Shift) - **Key** LONG → shows **software versions** of ALL StEin modules in the list;

TS (GA) – Track sections:

*)

For each of the 8 track connections is shown:
- the **active HLU statel** lluminated dots in colour gradation (similar to the red-green LED on the module),
either in **rectangular** (without east-west) or **arrow** shape (if HLU east or west bit is set).
- the occupancy detection: yellow dot (corresponding to the yellow occupied-LED on the module, but without "twitching", which indicates the RailCom messages received on the module)
- **overcurrent** and **short circuit** notifications (blue dot, similar to the blue LED on the module).

Setting the**HLU statuses** of the **track sections** and **switching back on** from the ***StEin LIST***

Number key (short) GA number 1 ... 8 **"** → switch one HLU level higher (i.e. H >> UH >> U >> LU, etc.) cyclically rotation, visible by the colour change of the HLU light point ("track OFF" not in the cycle)

- ↑ (Shift - press and hold continuously) - Number key→ make East-West assignment, cyclically rotation (without east-west >> HLU west >> HLU east, etc.) visible on the arrow shape of the HLU light point.

Number key (long) GA number 1 ... 8 **"** →Displays the "narrow list" of HLU states for alternative (numerical) setting of the HLU level:
Number key 0 ... 7 **"** HLU level (with "Rail OFF" !)
Select east-west by pressing **h** (shift),
3 sec Non-activation of the "narrow list" **"**
Change to "wide list" (same function, but with descriptions of the HLU states).

Number key GA number after "overcurrent" or "short circuit" (blue LED) → **Switch on again**!

Above example of a GA line: **Track sections 1 - 3** show a typical entry sequence from full speed to stop (colours green to red); can be a fixed setting or a current state in operation. Track sections 5 - 7 form a typical setting for shuttle operation: the end sections 5, 7 are directed "inwards" (shuttle train should turn back), track section 6 without directional specification; entire route at slow speed.

WE – Turnouts or single outputs: is shown...

- **the current turnout setting** is displayed by an arrow, which depending on feedback, is either coloured or blank; Flashes until the desired position is reached (i.e. with motorized turnouts) or when malfunctions occur.

The turnouts can be switched from the controller with the:
- corresponding number keys → switches turnout right and left

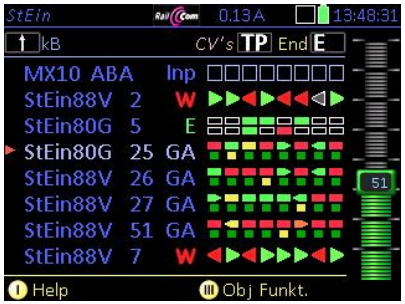
IN – Switch inputs: for each of the 16 switch inputs:

- the current state (green dot means ON).

E + 8

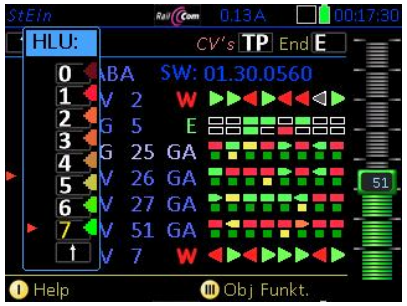
In this example, there are several StEin modules in the list, two of which just show the turnout positions and the other track sections with different HLU and east-west states.

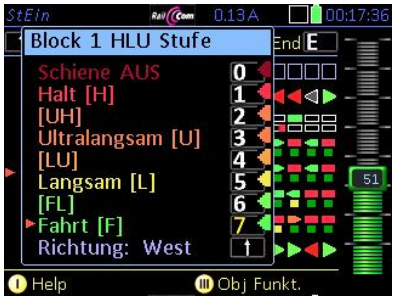
Tap on the list to switch from half to full screen display



Numeric key (long) TS number 1 ... 8→ for lower half or full screen
Switching the HLU states using the numeric key of the number from the "narrow list".

Numeric key (long) 1 ... 8, wait 3 sec.→ Automatic switchover to "wide list" with description of the HLU states Press the number key (long) and wait 3 seconds.





Inserted in the chapter StEin LIST (continuation of chapter 15)

Monitoring, switching, HLU setting with the "old" MX8 accessory modules and MX9 track section modules, CV programming and addressing of the MX8 and MX9 modules.

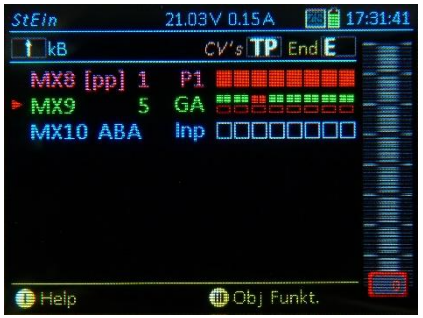
From SW-Version (MX10, MX32, MX33) 1.29.0600

MX8 and MX9 are modules from the MX1 / MX2 - MX31 system generation, available on the market in DCC versions from around 2000 to 2010. The MX10 (MX32 - MX33) system generation supports these "old" modules, as they are present in many systems.

MX8 and MX9 are also displayed in the ***StEin LIST***;
This can be accessed from the ***LOCO*** or ***SWI*** operating states via

E-Key + 8 → *StEin LIST*

If MX8 and/or MX9 modules are connected to MX10 via CAN bus cable (see MX10 operating instructions regarding precautions for the CAN bus for "old" modules), the following appears in the „StEin List“ ▶



This display therefore shows the elements to be monitored or switched for each module found: in the case of the MX8 the 16 turnouts in two groups (switchable with ↑ (Shift) key), in the case of the MX9 the occupancy statuses of the 16 track sections and the HLU statuses of the 8 main sections (combining two track sections in each case). The turnouts or the HLU states of the track sections can be selected using the number keys.
Softkey III → Opening the “**Object menu**” for the (cursor) module that selected in the “StEin list”.
This menu is used to initiate all processes relating to addressing and CV programming of the MX8 and MX9 modules.

Re-addressing an MX9

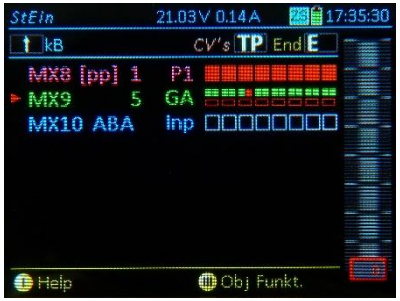
To re-address an MX9, this module must first be reset, which returns the MX9 to address 00 (often also referred to as "900"), i.e.
RESET in the object menu

Scroll to „MX8/MX9 Reset“ →

Address 0 ("900") should NOT be appointed for operational purposes!!
To actually **address the MX9 (from 0 to the target address)** again:
Softkey III → Open the "Object menu" for the same MX9

Scroll to „Adressing“ A-Key →

↓ Fill in the "New address" field, A-Key



The easiest way to re-address is of course to have only one MX8 or MX9 module connected to the CAN bus in addition to MX10 and MX32/MX33; other CAN bus devices such as control panels are irrelevant in this respect.

However, this is not always necessary;
For example, a "brand new" MX9 (which always has the address 0 or "900") can be also connected in addition to other modules already in the system and can be selected in the "StEin list" and set from 0 to the target address.

Incidentally, this is the reason (comment above) why address 0 ("900") should not be used for operational purposes.

Re-addressing an MX8

The procedure that must be carried out to address an MX8 is NOT completely identical to that for an MX9! There is no separate "reset command" for MX8 in the object menu, but the address 0 (also known as "800") must be set (which corresponds to a reset)
In the case of MX8 addressing (as opposed to MX9), this MX8 must be the only MX8 connected to the CAN bus.
To re-address an MX8, this module must first be reset, which is done as follows; "0" must be entered in the field for the "Old address"!

Scroll to "Addressing", A-Key, under "old address" Enter "0" A-key →

The MX8 is then NO LONGER displayed in the "StEin list".

Then press the

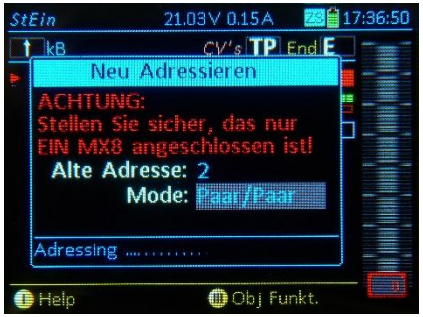
M-Key (MX33-Menu) → „MX8 Adressing" must be called up!

Scroll, A-Key

Enter the desired address ▶

IMPORTANT: Restart MX10 now!

→ MX8 now with new address in the "StEin list"



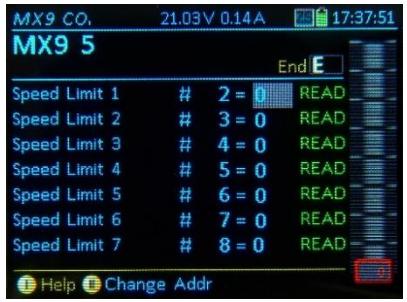
CV-Programming in MX8 or MX9

Select module by scrolling from the StEin list (cursor)

TP-Key → Open the relevant program screen

Alternatively, it is also possible to access the "Object menu" via softkey III and call up "MX8/MX9 Prog" therein Programming/reading the CVs in the same way as for decoders.

respectively



16. Programming in Service Mode: SERV PROG

= „Service Mode Programming", that is, programming on the programming track

SERV PROG can be reached with the

E-Key (to the E-Screen) + MN-Key (pressed in quick succession)

SERV PROG - ***the „Service Mode"***- was the only way of programming decoders in the early days of digital multi-train control with DCC and remained predominant for a long time. Today (for some time now) CVs are usually written and read in ***OP MODE - the "Operational Mode"***. Only when addressing (setting the vehicle address) is ***SERV PROG*** still common (although not often necessary).

The ***SERV PROG*** Help-File accessible with ▶ **Softkey I**.

SERV PROG

Programming and reading out CVs (Configuration Variables) on the programming track.
(Service Mode). Prerequisite: a track section separate from the main track ("output loop 2).

(E+)...MN-; TP-; U-; A-key, number keys

E+ MN key combination starts the SERV program. Pressing the TP- key identifies the decoder
(version, manufacturer, if ZIMO,
then type, serial number and loading code). This is not obligatory:
A+ number key → Programming the 'Address, that initiated by F key.
A+A → Read out the address,
U > Enter CV programming mode.
Enter the CV number, confirm with the A-key...

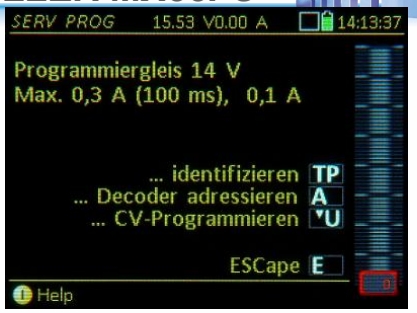
Other controllers indicate that the command station station's track output 2 is in Service Mode (provided the command station has two track outputs, otherwise the device itself). ▶

E-Key → Exits the ***SERV PROG*** mode

If the E-Key is pressed within the work area "Addressing" or "CV-Programming", it returns first to the programming menu screen, a SECOND E-Key actuation exits the ***SERV PROG*** mode.

E

MN



The first screen after entering the ***SERV PROG*** mode (with E + MN keys) is the programming menu, which also functions as a "security question" to prevent accidental deletion of decoder addresses or data.

Here, the currently applicable parameters of the command station's programming track are also displayed (if this is a MX10) such as maximum track voltage, maximum current etc., and that they correspond with the standard of the VHDM ("Rail Community") standard. However, these settings can be changed in the MX10 if needed (under "VOLT & AMP DETAIL, section " SERV"). This is necessary when a decoder or a vehicle with installed decoder consumes more quiescent current during programming than the "Rail Community" intended.

The keys displayed in the programming menu lead to the actual working areas:

TP

TP-Key identifies the decoder on the programming track with its address and displays it with name and image, if present in the object database. Additionally, some other important information such as manufacturer ID, decoder type, SW version, decoder ID, possibly loading code, and active sound project is read and displayed (the latter points only if it is a decoder manufactured by ZIMO)

If the decoder is identified as a ZIMO product, advanced programming support is available by displaying descriptions for the special ZIMO CVs (otherwise only DCC standard CV descriptions).

F-Key → (now that the address is known by decoder identification or address read-out) activates the newly recognized address and switches over to the driving mode ***LOCO***

TIP: This is a quick way to test a vehicle on the programming track if the address is not known.

NOTE: The identification procedure (which may take several seconds) **can be aborted** at any time with **the E-Key, U-Key or F-Key**. This is useful if only the first few lines of information are of inter-

The decoder identification with the TP-Key is, however, NOT necessary for changing the address or CV programming; the A-Key for address programming (or A+A for reading the address) or the U-Key for CV programming can be used immediately (without knowing the address). All this is possible in the service mode without actually knowing the address, which of course carries a certain risk.

A-Key → starts the working area **Address Programming**; Here, either press the A-Key again,

Therefore A+A → to **read** the current decoder address, or **enter a new address** and press the

F-Key or **W-Key** → to **write** (program) the new address into the decoder; see below.

U-Key → starts the working area **CV-Programming**; where CVs can be programmed or read-out; details see further down in this chapter!

E-Key → **Exit** service mode programming, switch to selection.

E-Key → Pressing the E button again causes a transition to the ***LOCO*** operating mode with the address that was active before programming (in contrast to the **F-Key**, which activates the address of the identified decoder)

*SERV PROG – Decoder address programming (after **A- Key**)*

If the step "...identify decoder" (with **TP-Key**) was performed before calling for "...program new address" (with **A-Key**), the current address and the other data are already known by the controller and displayed next to the address input field (until now: ...) ▼

NOTE: During the actual addressing procedure the necessary data is always read out (especially CV #29), even if previously "identified", because the system can only recognize in this way whether the decoder on the programming track has possibly been replaced with another one.

The foregoing "identification" is therefore not really necessary for programming an address but does offer more information. If only the "old" address is of interest, it can be read out directly (by pressing the A-Key again, see below); this is also faster than the decoder "identification".

A

(in this example AFTER identification with TP)

Type a new address in the entry field and press the

F-Key → programs the decoder to the new address as **a vehicle** decoder
W-Key → programs the decoder to the new address as **an accessory** decoder

In the case shown here (where the decoder was first "Identified" with the TP key) only one of these options (F or W-key) appears because the type of decoder (vehicle or accessory decoder) on the programming track is already known, due to a successful "identifying" or "address readout" procedure.

As for the other case, see further down in the text: "Program decoder WITHOUT "Identification".

→

F

After programming an address (or attempted programming) with the F or W-Key, one of the following confirmation messages appears:

ACK for a successful (= confirmed) programming step, or

NACK in cases where no confirmation was returned by the decoder. The associated alert "ATTENTION: Addressing not confirmed" means: the decoder did not send acknowledgement pulses, but the address MAY have been programmed successfully anyway, BUT: IT IS ALSO POSSIBLE that the address was programmed correctly but the previous CV #29 values were changed to ZIMO default values; this could affect the speed step mode (ZIMO default: 128), analog control (ZIMO default: yes) and RailCom (ZIMO default: yes).

ERR the decoder has rejected the (address) programming step

In addition, the MX33 screen shows which CVs have been set or attempted to be set to which values in the past addressing process, each with **ACK** / **NACK**

NOTE Replacing the decoder or vehicle on the programming track with another one without exiting and re-entering the ***SERV PROG*** mode is permitted and can be quite useful for saving time during "mass" address programming. In such a case, take the decoder (or vehicle) from the programming track after a successful programming process (confirmed with ***ACK***, see above), connect the next decoder to (or place the next vehicle on) the programming track and continue with the A-Key, type in the new address and press the F-key.

*Programming decoders WITHOUT preceding "Identification" (**A- Key** w/o preceding **TP- Key**)*

In this case, the usual list of data is not shown, instead the warning: "ATTENTION: Unidentified Decoder – Address programming is not safe ".

This means: it is NOT (yet) CERTAIN whether

1) whether there is a decoder (vehicle with decoder) on the programming track, or whether there is a decoder (vehicle with decoder) on the programming track.
2) the decoder is of the proper type (for a vehicle or accessory) that can be programmed with the **F-Key** or **W-Key**
3) it is a ZIMO or third-party decoder (important in relation to CV #29 default values)
4) it is even possible to communicate (read/write, with or w/o acknowledgment) with the decoder on the programming track.

However, the warning "not identified..." does NOT mean that programming or read-outs with acknowledgments are not possible, but only that there is an uncertainty in this regard.

The "deficiency" of a missed identification can either be rectified with the

A-Key (which in turn results in A+A) → to **read out** the address, (which is sufficient for a "minimal identification" that allows for a secure address programming) *or*
not rectified by immediately programming the new address unsecured, at one's own risk by typing in the new address and confirming it with the **F** or **W-Key**.

If the decoder type is not known, it is up to the user to press the CORRECT key (F or W). The WRONG button either prevents a successful address programming (namely when the decoder is able to acknowledge it, is read-out within the addressing process and thereby recognizing the key as the wrong one) or a failed programming (if the decoder cannot acknowledge - e.g. because no load is connected).

*SERV PROG – CV – Programming (after pressing the **U-Key**)*

After entering "CV programming" (with the **U-Key**), **the first input line** appears in which the first CV editing is carried out. The example shown illustrates the case where decoder identification was NOT carried out in this session, hence the message "... not identified".

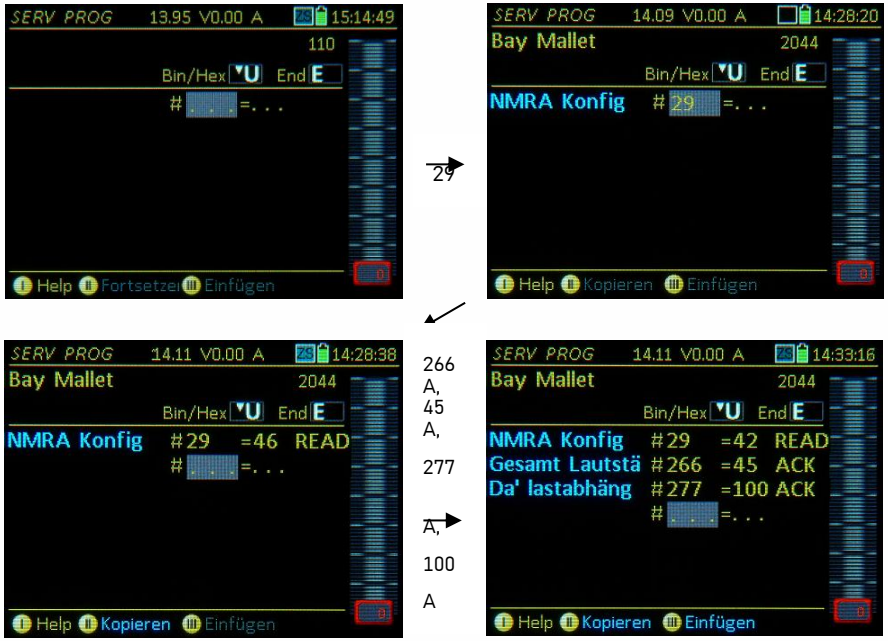
... Program a CV by

Typing a CV Number – **A-Key** – Enter a CV value – **A-Key**

... Read-out a CV by

Typing a CV Number – **A-Key** – (again) **A-Key**

U (in this example AFTER identification with TP or A+A)



CV programming success (or failure) report at the end of each line after completion with the **A-Key**:

ACK successful CV programming (acknowledged)
READ CV value read out successfully
NACK programmed without acknowledgement
NO-R CV read-out failed

An automatic line feed takes place after each programming or readout process. A new process can then be carried out with the

Scrolling wheel → can be used to mark both values and CV numbers in previously edited lines,

C-Key → to delete the highlighted number (CV number or value), then enter a new value in the field followed again with the A-Key; the process of correcting a line is the same as entering to a new line (see above).

The following actions can be taken at any time during CV programming

F-Key → **activates** the vehicle and switches to the ***LOCO*** mode (only when the address is known, i.e. if the decoder was identified with the TP-Key or address read-out with A+A, before starting the CV programming).

F (from the CV-Prog window)

E-Key → **returns** to the programming menu screen and pressing the E-Key again returns to the screen the controller was in before entering ***SERV PROG***, which is usually the ***LOCO*** or ***SWI*** mode.

U-Key → opens an editing window for the CV currently worked on, to turn individual bits ON/OFF, some important CVs come with a description and a hexadecimal representation.

▼ *Examples*

U (on CV 1)

U (on CV 112)

NOTE: In many cases, the individual bits have no standardized meaning and can be used differently by the decoder manufacturers. Therefore, full names of some CVs are only available if the decoder was previously identified (**TP-Key**) as ZIMO decoder.

SERV PROG – Creation and use of CV – Sets

…. Using the Clipboard:

Softkey II = **Continue** → (appears when entering the CV programming mode after a previous programming session with the same address).

Displays the programmed and read-out CV lines from the last session again (i.e. CV 29 = 10 READ, etc.

> **NOTE**: there is no automatic control whether the values still match what is in the decoder. CV writing and reading were already carried out in the previous session and are now listed again for overview and correction.

Softkey II = **Copy** → copies all CV lines (i.e. CV #29 = 10, etc.) from the current screen view to the clipboard. Copying serves primarily to transfer a group of CVs (a "CV-Set") from the decoder currently worked on to another.

Softkey III = **Paste** → The CV lines from the clipboard (i.e. CV #29 = 10, etc.) are pasted to the current screen. See notes above for Soft-Key II = Copy.

> **ATTENTION**: With the "Paste" command, the CV-lines are only added to the list but NOT yet sent to the decoder. This must be carried out afterwards for each individual line; that is, in practice:

A-Key (after scrolling to the first line) → repeat for each CV number and value (other than pressing the A-Key, no further actions are required, except when a line is to be skipped), then the confirmation ACK appears in each line if the programming step was successful.

…from and within the CV-Set-Memory:

In contrast to the clipboard (for one single CV set) an (almost) limitless number of CV-sets is managed by the CV-set memory, stored by theme with freely definable names and descriptions. A number of CV-sets are included from the outset in the controller and more can be added via flash drive (as part of the general controller software updates or independent thereof)

↑ (Shift) → switches the Soft-Keys II and III from the above clipboard application (II = copy, III = paste) to the CV-set application; recognizable by changing to a blue font:

II = copy CV-Set III = paste CV-Set

An active shunting mode (R-key engaged) is automatically terminated when calling the CV-programming screen (OP PROG and SERV PROG)!

Soft-Key II = Copy CV-Set → Saves the CV rows of the current screen to the CV-Set memory:

A CV-Set input window opens.

Theme: Select one of the suggested themes (with rocker switch).

Name: Enter a name of your choice

Info: Free choice of text.

↑ (Shift) + + A-Key → SAVE (Attention: otherwise the CV-Set is not saved).

↑, A

Soft-Key III = Paste…→ Search for the desired CV-Set in the CV-Set database and paste the CV-set rows to the current display screen.

II → A

> ATTENTION: With the "Paste" command the CV-lines are only listed but NOT yet sent to the decoder. This must be carried out afterwards for each individual line; that is, in practice:

A-Key (after scrolling to the first row) for each CV number and value (other than A, no further actions are to be taken, except when a row is to be skipped), then the confirmation ACK appears in each row if the programming step was successful.

CV-Liste vs.CV-Sets:

CV-Set: Each line consists of the CV-number and the CV value (e.g. CV 29 = 10)

CV-List: The CV-lines are stored without values and are therefore also inserted without values; they must be filled line by line and serve as a template for the complete coverage of all CVs for a specific topic (e.g. parameters necessary for motor control).

BILD wird ERSETZT

17. Programming in Operational Mode: OP PROG

= „Operational Mode Programming“
also known as „programming on-the-main “, “PoM” or „Programming on-the-fly”.

OP PROG is started from the driving mode ***LOCO*** or ***SWI*** with
E- Key (to E-screen) + **F- Key**

In contrast to ***SERV PROG***, the ***OP PROG*** always refers to the currently active address displayed on the controller and therefore it is not necessary to enter or read-out the address first.

The Help-File for ***OP PROG*** can be reached ▶ with Softkey I:

In the early days of digital multi-train control, ***SERV PROG*** - *the* ***"Service Mode"*** - was the only way of programming decoders and remained predominant for a long time. Today, CVs are written and read in ***OP PROG*** - *the* ***"Operational Mode"***. Only for addressing (setting the vehicle address) is ***SERV PROG*** still widely used (although not often necessary).

E-Key ➞ Exits the ***OP PROG*** mode.

If the E-Key is pressed within the working area of address or CV programming, the program switches back to the programming menu first; only a SECOND **E-Key** actuation exits ***OP PROG*** completely.

Programming with ***OP PROG***

For programming and reading CVs on the main track (Operational Mode = PoM, Programming on the Main); change also decoder address or enter Consist address. i.e in the operational mode ***LOCO***, also E + F

The decoder is identified (manufacturer, type, SW.Version,,..., serial number and load code for Sound project).

A ➞ Changing the vehicle address
A ➞ Enter a consist address,
U ➞ Getting started with CV programming
Enter the CV number – Complete with A
Enter CV value – complete with A – or
A again (for CV readout via Railcom),

Decoder responses during programming/reading:
ACK Acknowledgement Prog (Acknowledgement)
READ Read and displayed (by RailCom)
SENT Prog command sent (NO response)
NACK Programming attempt without ACK
NO-R Unsuccessful read attempt

E➞ End of the CV programming procedure
E (again) ➞ ***LOCO*** screen
Further operating procedures during programming:
Scrolling wheel - mark already edited lines, and thus, edit again.
C ➞ Delete (CV number or value).
U key ➞ opens window: Set / delete bits individually delete ink description and heximal representation.
TP Key- opens window: Test operation of the F-keys; controller and R key are assigned to the address being processed.

Back from Help

E

F

▲

The first screen when entering ***OP PROG*** is a menu screen; ABOVE is the display shown of a decoder with working **RailCom feedback**, which makes identification possible: Information such as manufacturer, software version and decoder ID are read out and displayed automatically (in magenta, the color for RailCom);

BELOW is the display of a decoder **WITHOUT** working **RailCom feedback** and is therefore unidentifiable (i.e. the decoder is not equipped with RailCom, disrupted transmission etc.): the message “not identifiable” is displayed but the CV programming can still be started with the **U-Key** (although reading decoder CVs won't work).

▼

E+ F

NOTE: ***OP PROG*** is possible while driving or at standstill. The vehicle can be driven normally while programming or reading out data. Of course, data transmission tends to be worse while driving and it is therefore recommended to drive only when RailCom feedback is available (or the programming changes can be seen instantly by the vehicle's behavior).

Use the displayed function keys from the menu to switch to the actual working area:

TP-Key ➞ the decoder is once again identified (manufacturer, SW version... same as during entry)

A-Key ➞ the decoder input screen is displayed where a new address can be written to the decoder (provided the decoder allows programming a decoder address on the main track, which the NMRA R&R does not provide for).

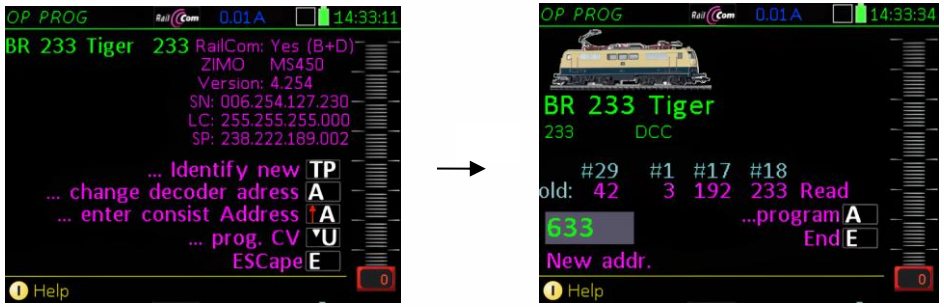
↑ (Shift) + **A-Key** ➞ Screen for entering a consist address (CVs #19, 20) including supplementary info

U-Key ➞ opens the CV-Programming page, where CVs can be read-out and programmed; see below for details.

OP PROG – A-Key → Decoder Address change

NOTE: The address programming via "PoM" is only possible for decoders that support it, for ZIMO decoders from SW version 37.8 (non-sound) or 37.28 (sound decoder). Older versions and many thirdparty decoders can only be programmed in SERV PROG (i.e. on the programming track).

A-Key → Enters the procedure "Change decoder address"
then all CVs relevant to the address are read, i.e. #1, #17, #18, #29; the values are normally not important for the user, but useful in case of problems. ONLY IF this reading procedure works (ACK), re-addressing is possible.



An input field appears under the "Old data", in which ▲ the new address is entered.

A-Key (... programming) → The address entered in the input field is programmed in decoder

ACK after programming a CV and confirmation via "RailCom" (or other feedback)
SENT after programming a CV without a feedback system (shouldn't actually happen),
NACK after a programming attempt without acknowledgement (even though a feedback system is present).

The active engine is automatically switched to the new address and can be driven immediately! (Exception after "NACK": the old address remains active).

OP PROG – ↑ + A-Key → Decoder-connected consist address entry

NOTE I: described here is the decoder-based consisting, which is not the same as the system-based consisting as explained in the chapter "LOCO AND CONSIST CONTROL"!
NOTE II: The ZIMO controller procedure for the assignment of a consist address partly refers to settings which only exist in ZIMO decoders, for example the use of the CV #20 for consist addresses >127 (to 9999) and the consist automation (**automatic switching between single engine and consist driving**, when the individual address of a (consist) vehicle is addressed).
NOTE III: Further setting options in Consist mode (auto. control reduction, auto. light switch-off, etc.) are defined in the ZIMO decoders by individual CV programming.

↑ + A-Key→ starts the procedure "Consist address entry
The currently valid consist data (the consist address, direction and consist automation) is read out of the decoder and displayed in the input fields,where new data can be entered: the new consist address for the vehicle, the desired direction (same as the consist or reversed) and the consist automation (ON / OFF), see above for details.
A-Key (... programming) → new data is sent to the decoder

↑ + A

OP PROG – U-Key → CV-Programming

After entering "CV programming" (with U-**Key**), the first input line appears for:
... **Programming** a CV by typing in a
CV-Number – **A-Key** – CV-Value – **A-Key** , or
... **Read-out** of a CV (if RailCom or other feedback system is available) by typing in a
CV-Number – **A-Key** – (again) **A-Key**.

A report about success (or failure) of a program or read-out process, at the end of each line after completion with the **A key**, appears automatically:

ACK after programming a CV and receiving confirmation via "RailCom" (or other feedback)
READ after reading out a CV value and confirmation through "RailCom" (or other feedback)
SENT after programming a CV without a feedback system
NACK programmed without acknowledgement (even though a feedback system is present)
NO-R after unsuccessful read-out (usually because of no feedback)

Scrolling wheel → is used to mark previously edited lines (values or CV numbers) and overwrite the values as needed

C-Key → deletes the marked value (CV number or value)

CV-Programming **without** feedback: ▶
Values are sent (SENT),
programming success cannot by checked, read-out is not possible (therefore always NO-R),
only NMRA standardized CV descriptions will be shown because the decoder manufacturer is not known.

CV programming, read-out with feedback (RailCom):
Programming is acknowledged (ACK), read-out is possible (READ), and all CVs of ZIMO decoders have descriptions. ▼ ◢

E-Key → Returns to the menu and with a second E-Key to the ***LOCO*** or ***SWI*** mode
TP-Key → **Testing** (with function key window) without leaving the programming screen.
U-Key → CV-Bit-Programming (via a special pop-up window for turning individual bits on or off)
The Softkeys : **II** = **Continue with or copy** the last edited CV-List , **III** = **Paste**
"Copy" and "Paste": to or from the clipboard for CV-sets. This allows CV sets (these are the CVs in the current list) to be passed from one decoder to another.

Das CV-Programming in „Scroll Prog“ Mode

This makes it possible to place a CV on the scroll wheel and adjust the CV value quasi-continuously via the scroll wheel (by incrementing and decrementing The programming is carried out immediately and the reaction of the vehicle can be observed immediately, for example the adjustment of the volume as with a classic rotary knob.

In ***"Scroll Prog"*** mode, you can also use the F and U keys to conveniently switch back and forth between two or more CV values in the list in order to make related or codependent settings together, e.g. acceleration and braking, P and I values of the motor control, high and low pass filters of the sound, etc.

Introduction and operation principle of the "Scroll Prog" mode:

First enter the desired CV number as described on the previous page and then read out the current value or enter your own CV value (field must be marked); then:

Press the rocker switch → Switch the ***„Scroll Prog“*** mode;
corresponding indication on the display; see image above ▲▲▲

Scrolling wheel → Set the desired CV value; setting is transferred immediately (**SENT**),
and - if possible - confirmed (**ACK**)

F- und U-Key → Jump to the line above or below, where the corresponding CV can then be set immediately using the scroll wheel (however, this must already have been opened in this line when preparing to work in "Scroll Prog").

Exit the „Scroll Prog“ Mode:

Any number key, A or E key → Return to "normal" ***OP PROG.***

CV numbers and values can only be typed in again and/or CVs can be read out specifically in the "normal" ***OP PROG.***

OP PROG – Anwenden und Erstellen von CV - Sets

Currently not implemented!

.... **Via the clipboard:**

Softkey II = **Continue** → reloads the CV list from the last editing session

Softkey II = **Copy** → copies the current CV lines to the clipboard.

Softkey III = **Paste** → pastes the CV lines from the clipboard to the currently displayed list

.... **from and within the CV set memory:**

↑-Key (Shift) → switches the Soft-Keys II and III from the above clipboard application to the CV-set application (II = copy, III = paste); visible by changing to a blue font: II = copy CV-Set III = paste CV-Set

Softkey II = **copy CV-Set** → Saving the CV rows of the current screen to the CV-Set memory:
Theme: Select one of the suggested themes (with rocker switch).
Name: Enter a name of your choice. / Info: Free choice of text.

↑-Key + A-Key → SAVE (Attention: the CV-Set is not saved without this key sequence)

Softkey III = ... **Paste** → search for the desired CV-Set in the CV-Set database and paste the CV-set rows to the current display screen.

The "Paste" command only lists the CV-lines in the display but are NOT yet sent to the decoder. This must be carried out thereafter for each individual line; that is, in practice press the;

A-Key (after scrolling to the first row) → continue with each subsequent line for each CV number and value (other than pressing the A-Key, no further actions are necessary, except when a row is to be skipped), then the confirmation ACK appears in each row if the programming step was successful.

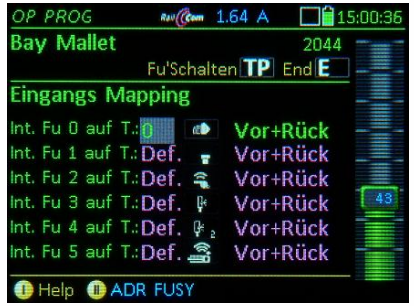
OP PROG – Special configuration procedures only for ZIMO decoder

M-Taste → the first MENU lines ▶

Scrolling wheel → scroll down to the special procedures for ZIMO decoders (last menu section)
A-Key → starts the selected procedure

ZIMO LOCO Settings (not yet available) ZIMO HLU Settings (not yet available) ZIMO Input Mapping ZIMO Swiss Mapping ZIMO Speaker volume for function sounds ZIMO Smoke functions (not yet available)

The **ZIMO Input Mapping** allows you to change the function key assignments ("external" functions to "internal" functions, as per sound project or default setting) . ▶



The “internal function" for example, is assigned to the respective **F-Key** in the sound project: this function is to be assigned to another function key. If this new function key is already activated used for another function, it must be moved first.

Directional dependencies of functions can also be defined in this assignment window.

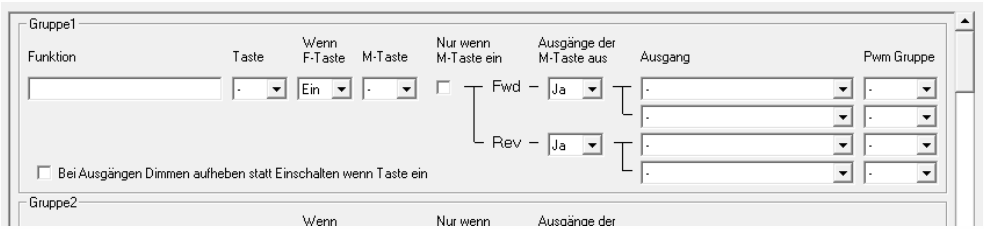
E-Key → returns to the menu screen

ZIMO Swiss Mapping

It makes easy to set up very complex lighting or function (dependent) switchings. For example, a free function key is assigned for (light) functions, without having to know the required CVs. ▶

(F-Tst): the selected function key should switch the function
(Inv): yes/no, if yes: the function switching is inverted when the function key is activated.
(M-Tst): Defines a key as the “Master Key” (i.e. F0 Light).
(F&M): outputs only on if both F and M-Key are on
(A1v, A1r; A2v, A2r): outputs to be activated, direction dependent.
(PWM): pulse-width modulation groups can be defined.

The logic is the same as in the "ZIMO Sound Programmer" (ZSP) available as a free download from the ZIMO Website (http://www.zimo.at/web2010/products/zspdownload.htm). The picture below is a screen copy to compare with the MX32 display. Text in magenta is from the MX32 Swiss Mapping:



The ZIMO Function Sound Volume screen, allows to quickly and easily assign function sounds to (free) function keys and adjust the volume at the same time. ▶

The sounds can also be set to endless playback or short playback duration (sound breaks off and plays the end part of the loop). To simplify the use of this setting, it is of advantage to know the sound sample number, which is published by the "Sound Providers".

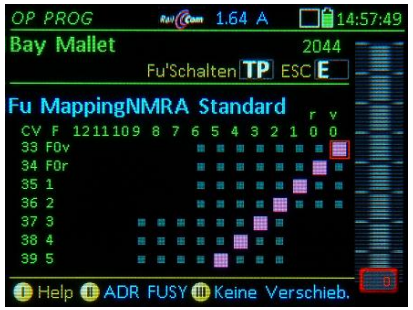
OP PROG – NMRA Function mapping

Accessible in ***OP PROG*** with:

M-Key → the first menu lines appear. ▶

Scrolling wheel → scroll to the desired line

A-Key → starts the selected procedure



◀ **NMRA Function Mapping (standard)**

This is the basic function mapping of ZIMO decoders, with certain additional capabilities, compared to the NMRA defined procedures.

Scrolling wheel → Scroll in the column

Rocker switch → Scroll in the row

Number keys → select the function keys

◀ **NMRA Function Mapping without left-shift**

This special ZIMO feature is accessible with CV #61 = 97 and allows "higher" F-keys to also control "lower" functions

ZIMO SOUND configuration in real time and ZIMO "Sound Collections"

A distinctive feature of ZIMO sound projects is the ability to adapt the sound through the controller to the user's requirements during operation.

The user is free to change acoustics of a locomotive to his/her own taste by combining for example a chuff sound from 5 different chuff samples and one or several whistles; equally select bells, compressors, coal shovels, oil burners or break squeals etc.

The "CV #300 - Procedures" can be used with any digital system. It is however much more convenient with a ZIMO MX31, MX32 or MX33.

Some ZIMO sound projects are organized as "Sound Collections" containing "sound groups", with each group (e.g. chuff sets, whistles, air pump sounds ...) having different variants (recordings from different prototype locomotives) stored in the decoder, which can be put together individually

The following sound groups (selectable in the field "Type") within the sound configuration are available:

- Driving sound groups (only in sound-collections, not in "normal"single-engine projects) - Groups: Chuff sets, boiling, brake squeal, thyristor sounds ...
- Function sound selection - the predefined allocation of whistles, horns, pumps etc. can be changed for each function key F1 – F20, both within a sound group (i.e. whistles, if several types are available which is typical for sound collections) and between groups (i.e. moving a whistle to another function key).
- Random sound selection - for each random sound generator Z1... Z8
- Switch input sound selection - for each switch input S1...S2

It is also possible to adjust the volume of the selected sound file within the sound selection procedure (not to be confused with the total volume).

18. ObjectDB... = Object Database

The storage organisation of the vehicle addresses in the ZIMO system:

The ZIMO system is designed for installations of any size, i.e. also for an almost unlimited number of vehicles. This requires a sophisticated database structure to manage an almost unlimited number of vehicles. The so-called **"object database for vehicle addresses"**, abbreviated to **"ObjektDB"** or **"ObjDB"**, plays a role here, possibly with a new name in 2023 (see also below) for the purpose of standardization with the ZIMO APP.

BUT: In the case of SMALL applications - with 10 or 20 locomotives - it is NOT NECESSARY to really deal with the entire ZIMO storage organization; it is usually sufficient to activate the addresses as required or to take them from the retrieval memory =*LoR*.

NOTE: The ZIMO digital system is - like other DCC systems - ADDRESS-ORIENTED; i.e. the vehicle address is the organising principle for vehicles, which usually retain their address once assigned (programmed into the decoder) in the long term. This is a fundamental difference to mfx systems, and also remains so in the case of automatic registration according to RCN-218 and/or ZIMO GUI transmission, although there is a certain similarity to mfx.

The central ObjectDB in the MX10 or MX10EC (or in a future command station):

All vehicles (addresses) known in the system are listed here, together with the current driving data (speed, direction, ...) and GUI data. If the driving or GUI data is changed on a controller (ZIMO controller, ZIMO APP, compatible third-party device such as Roco WLAN Maus), the *central ObjectDB* will automatically adopt this change, provided that the controller recognises this digital command station as a home system.

The list of vehicles (addresses) stored in the ***central ObjectDB*** is displayed ...

- on the display of the digital command station, but limited by the small size of the display,
- on the ZIMO MX32/33 controller by accessing (E+6) the ***ObjectDB***, where *blue*, *green*, *turquoise* *) AND *grey* lines are displayed, whereby the grey lines do NOT reside in the operating device, but only in the central ObjectDB (and possibly also in other operating devices).
- on the ZIMO APP by selecting the ***SYS DB (F)***, which is a copy of the ***central ObjectDB***.

*)*Blue Text* represent vehicles (addresses) that are NOT in the *LoR* and do NOT belong to a consist, but only to the object database of the displaying operating device (MX32/33 or ZIMO APP).
Green Text stands for vehicles (addresses) that are contained in *LoR = loco recall memory* of this operating device,
Turquoise Text stand for vehicles (addresses) that belong to a consist (thus automatically also in the *LoR*),
Grey Text stand for vehicles (addresses) that are NOT contained in the object database of the displaying operating device, but only in the digital command station (MX10, ...) and possibly also in other operating devices.

The *local ObjectDB in the operating device (i.e. ZIMO controller or ZIMO APP):*

All vehicles (addresses) known to the operating device itself are contained here. If a change is made to the driving or GUI data on another operating device (ZIMO controller, ZIMO APP, third-party device such as Roco WLAN Maus), the own ***local ObjectDB*** also takes over.

The list of vehicles (addresses) contained in the ***local ObjectDB*** is displayed ...

- in the case of a ZIMO MX32/33 controller by selecting (E+6, i.e. as above) the ***ObjectDB***, where the *blue*, *green* and *turquoise* *) lines form the ***local ObjectDB***, while the *grey* lines are visible (because they are selected with E+6 as above), but do not belong to it.
- in the case of the ZIMO APP by calling the ***APP DB (F).***

The *LoR (=Loco Recall Memory) in the operating device (i.e. ZIMO controller or ZIMO APP)*

Favourites list of the ***local ObjectDB***, where all activated addresses are automatically entered; i.e. no own database, but entries of the ***local ObjectDB*** marked as *LoR* member.

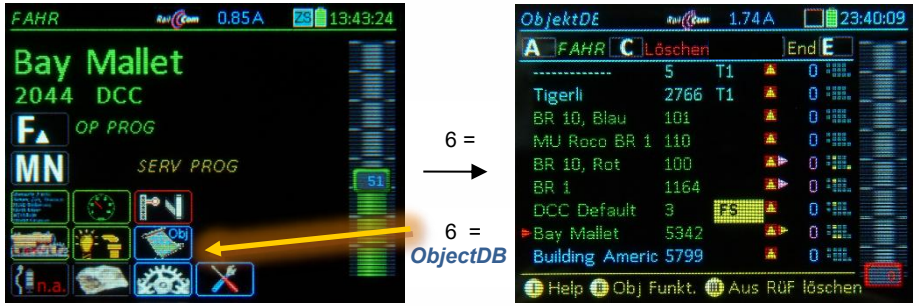
The list of vehicles (addresses) contained in the ***LoR (=Loco Recall Memory)*** is displayed ...

- in the case of a ZIMO MX32/33 remote by calling (softkey III), after which the *green* and *turquoise*[*)] lines from the ***local ObjectDB*** are displayed, initially in the lower part of the screen.
- in the case of the ZIMO APP by accessing the *LoR*.

The *LoR* that is active in the *ZIMO APP*,

Special display of the *LoR (=Loco recall memory)*, which retains the most important operating elements (speed controller, direction key, function keys) for each entry, which allows the immediate and simultaneous control of all vehicles (addresses) contained, without individual activations.

E-Key + 6 → Entry into ***ObjectDB*** from the ***LOCO*** or ***SWI*** operating states



Wipp

ObjectDB will be shown *after E+6*
all texts are taken into account, i.e. the blue, *green*, *turquoise*, *grey* texts *)
leads to the following results in ObjectDB ▶
(= all vehicles in the digital command station)

The grey texts are not taken into account, so only the *blue*, *green* and *turquoise* texts in the ***local ObjectDB,*** ▶
(= the vehicles in your own device)

The data displayed is very similar to the LoR display (in content and sequence), i.e. name, address, any external controls or consists, current speed level or speed, function tableau for F0 - F9.
Fields, similar to LoR ...

Wipp

Level with GA's

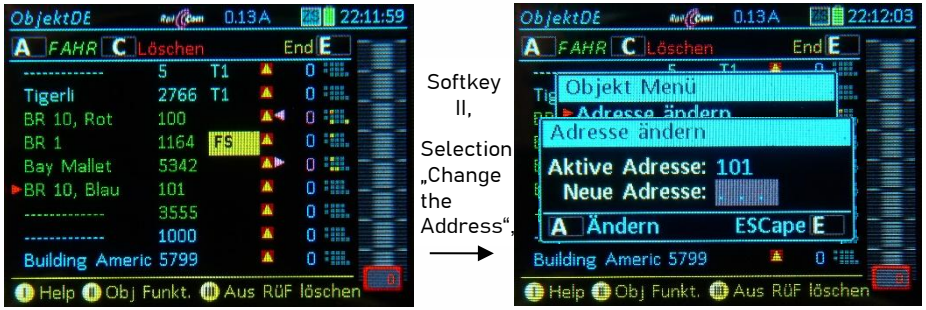
The levels of the ObjectDB display...

Rocker switch → Display of further levels of the displayed ***ObjectDB***, where for example the track sections (GA's, position of the decoders) or the DID's (Decoder Unique ID's) can be seen. The data actually displayed can vary from SW version to SW version!

ObjectDB level with DID's

Softkey II → opens the „**object-functions**“ in the box with the options.
„change the address“ or
„change the name“

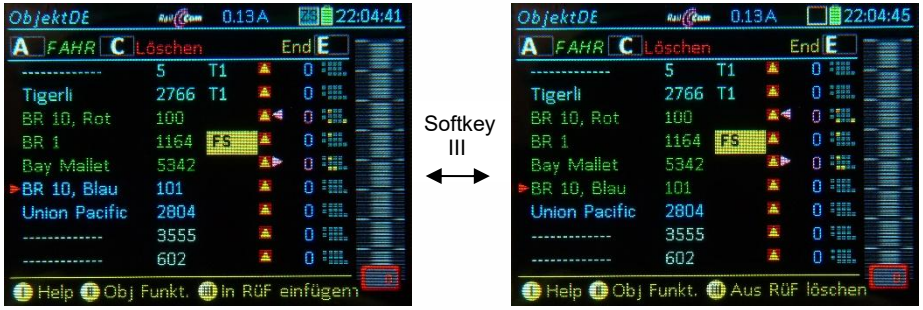
NOTE: these options are of course also available elsewhere, but here in the table is clearer for larger "change requests"!



Softkey III → Insert into the ***LoR*** / remove from the ***LoR*** - leave in Local ***ObjectDB***.

Insert address that was previously NOT in the RüF (but in ***local ObjectDB***), i.e. *blue text*, into the ***LoR*** (*loco recall memory*) (line turns *green*)

remove address that was previously in the ***LoR*** (i.e. *green* text) from the ***LoR*** (line turns *blue* (because it is still in the ***local ObjectDB***).



*Deleting addresses from **local ObjectDB** or **central ObjectDB***
… is always initiated by the **C-Key**

If C-key is clicked on *blue* or *green* text = deletion from ***local ObjectDB*** (operating device),
However, the address remains in the ***central ObjectDB*** (i.e. in MX10), so it becomes a *grey* text

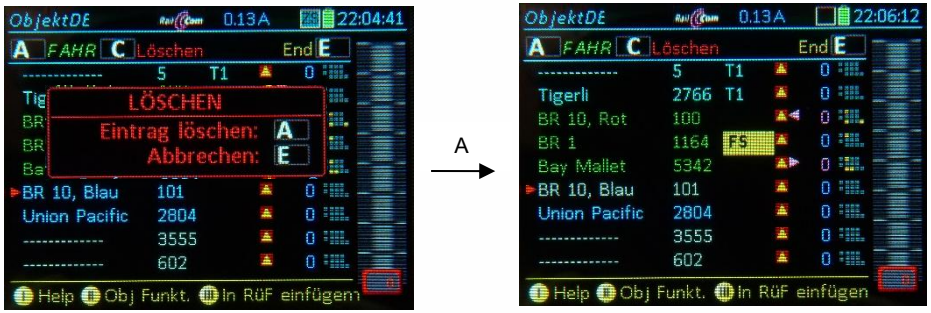
If C-key clicked on *grey text* = delete from ***central ObjectDB*** (i.e. from MX10), *grey text* remains, marked with "Recycle bin" (no longer in the display when ***ObjectDB*** is displayed again).

NOTE: Simultaneous deletion from the ***local*** and ***central ObjectDB*** is NOT possible; only two-stage deletion is possible!

therefore:

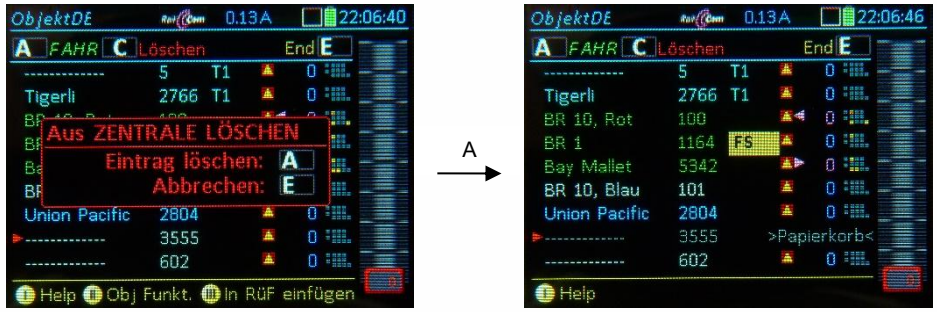
C-Key on *blue* or *green* text (i.e. address in ***local ObjectDB***, regardless of whether in ***LoR*** or not):
NOT possible on *turquoise* line; the traction would have to be cancelled first - in ***LoR***.

→ **Delete** from the ***ObjectDB*** (ATTENTION: active vehicle can NOT be deleted), text becomes *grey*, so no longer in ***local ObjectDB*** (i.e. no longer in the operating device), but still in ***central ObjectDB*** (i.e. only in MX10, ...)



C-Key on *gray text*
(i.e. address in ***central ObjectDB*** (in MX10), NOT (any more) in **local ObjectDB** (operating device):

→ **Delete** from the ***central ObjectDB***, in MX10, so in the system)
actually: Move to the **recycle bin**
Text remains grey, marked >recycle bin

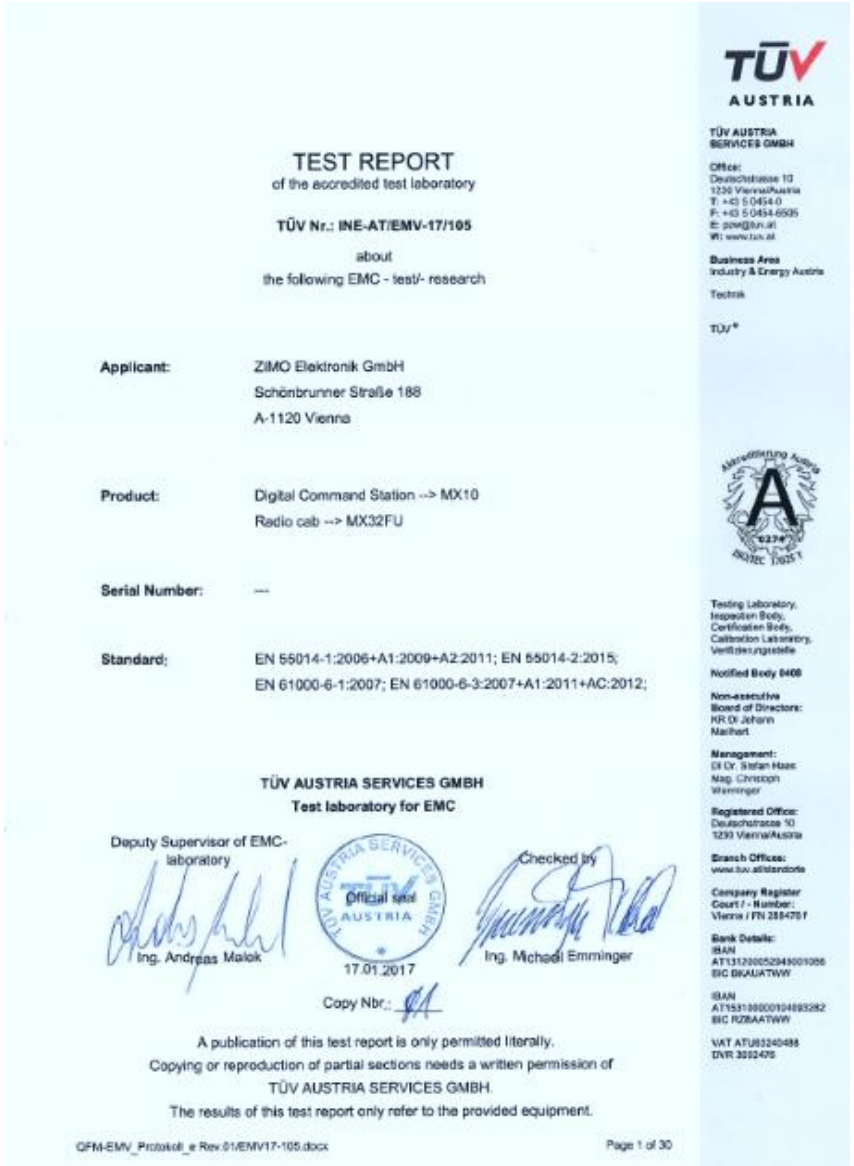


Can still be reactivated in this state with the A key!
*After leaving the **ObjectDB** display is no longer available!*

A-Key → Activate the selected address and open in LOCO; the (previous) blue or grey line is transferred to its own LoR (recall memory), i.e. it becomes a green text.

Anhang

Betreffend Basisgerät MX10 und Fahrpult MX32.



TÜV AUSTRIA

TÜV AUSTRIA SERVICES GMBH

Office:
Deutschstrasse 10
1230 Vienna/Austria
T: +43 5 0454 0
F: +43 5 0454 6505
E: psw@tuv.at
W: www.tuv.at

Business Area
Industry & Energy Austria

Technik

TÜV®

Testing Laboratory,
Inspection Body,
Certification Body,
Calibration Laboratory,
Verifikationsstelle

Notified Body 0408

Non-executive
Board of Directors:
KR DI Johann
Marihart

Management:
DI Dr. Stefan Haas
Mag. Christoph
Wareninger

Registered Office:
Deutschstrasse 10
1230 Vienna/Austria

Branch Offices:
www.tuv.at/standorte

Company Register
Court / - Number:
Vienna / FN 288470 f

Bank Details:
IBAN
AT131200052949001066
BIC BKAUATWW

IBAN
AT153100000104693282
BIC RZBAATWW

VAT ATU63240488
DVR 3002476

TEST REPORT
of the accredited test laboratory

TÜV Nr.: INE-AT/EMV-17/105

about
the following EMC - test/- research

Applicant:	ZIMO Elektronik GmbH Schönbrunner Straße 188 A-1120 Vienna
Product:	Digital Command Station --> MX10 Radio cab --> MX32FU
Serial Number:	---
Standard:	EN 55014-1:2006+A1:2009+A2:2011; EN 55014-2:2015; EN 61000-6-1:2007; EN 61000-6-3:2007+A1:2011+AC:2012;

TÜV AUSTRIA SERVICES GMBH
Test laboratory for EMC

Deputy Supervisor of EMC-laboratory
Ing. Andreas Malok

Official seal
17.01.2017

Checked by
Ing. Michael Emminger

Copy Nbr.: 01

A publication of this test report is only permitted literally.
Copying or reproduction of partial sections needs a written permission of
TÜV AUSTRIA SERVICES GMBH.
The results of this test report only refer to the provided equipment.

QFM-EMV_Protokoll_e Rev.01/EMV17-105.docx Page 1 of 30

***Declaration of Conformity*:**

ZIMO Elektronik GmbH hereby declares that the product MX10 bears the EC mark and built in accordance with the provisions of Directives 88/378 / EEC; 89/336 / EEC; 73/23 / EEC.

***24 months warranty*:**

Our products are technically sophisticated and are manufactured and tested with utmost care, therefore, ZIMO Elektronik GmbH guarantees their products for 24 months from the date of purchase (with proof of purchase from a ZIMO contractor).

The warranty covers the repair or replacement of defective parts. ZIMO Elektronik GmbH reserves the right to proceed at its own discretion only if the damage is proven to be the result of a design, manufacturing, material or transport fault. A repair does not extend the warranty. Warranty claims can be made with a ZIMO contract partner or ZIMO Elektronik GmbH. Proof of purchase is required.

The warranty does not apply:

- with normal wear and tear
- if devices are not used for the purpose intended by ZIMO Elektronik GmbH and in accordance with its operating instructions
- in case of modifications or alterations not performed by ZIMO Elektronik GmbH.

ZIMO Elektronik GmbH
Schönbrunner Straße 188
1120 Wien
Österreich