

K
R
M O D E L L
I
S

CAR  SYSTEM

HANDBOOK

VERSION 1.0.1

24. FEBRUARY 2021

car.krois-modell.at

1 INTRODUCTION

This document describes the components and features of the Krois-Modell Car-System. Besides the detailed instructions on how to wire the decoder and its peripherals it also contains all the information needed to configure the hardware using the PC-software.

We strive to constantly improve and extend this user manual – in case you find any errors or want to suggest improvements of any kind please do not hesitate to contact us under the email support@krois-modell.at.

2 CHANGELOG

- ❖ Version 0.9.5 (03.05.2018)
 - | First published version (work in progress)
- ❖ Version 0.9.6 (21.05.2018)
 - | Full release
- ❖ Version 1.0.1 (24.02.2021)
 - | Updated ACC-schematic

3 TABLE OF CONTENTS

1	INTRODUCTION.....	1
2	CHANGELOG.....	1
3	TABLE OF CONTENTS.....	2
4	DECODER.....	4
4.1	HARDWARE.....	4
4.1.1	COMPONENTS.....	4
4.1.2	TRANSCEIVER (TOP PCB).....	4
4.1.3	PERIPHERALS (BOTTOM PCB).....	4
4.2	STATUS-LEDs.....	5
4.2.1	LED-STATES (DESCRIPTION).....	5
4.3	SUPPLY (BATTERY).....	5
4.3.1	VOLTAGE LEVELS.....	5
4.3.2	SOLDER INSTRUCTIONS.....	5
4.3.3	POWER SWITCH.....	5
4.3.4	IMPORTANT INSTRUCTIONS.....	5
4.4	PROPULSION (MOTOR).....	6
4.4.1	POLARITY.....	6
4.5	FUNCTION-LEDs.....	6
4.5.1	POLARITY.....	6
4.5.2	MAPPING.....	6
4.5.3	OUTPUT-CURRENT/BRIGHTNESS CONFIGURATION.....	6
4.5.4	WIRING.....	6
4.5.5	IMPORTANT INSTRUCTIONS.....	7
4.6	INFRARED-LEDs (DISTANCE CONTROL – COMPATIBILITY MODE).....	8
4.6.1	PURPOSE.....	8
4.6.2	INFRARED DIODES.....	8
4.6.3	SOLDER INSTRUCTIONS.....	8
4.7	DISTANCE CONTROL (KROIS-MODELL).....	8
4.7.1	INTRODUCTION.....	8
4.7.2	MODULE (OVERVIEW).....	8
4.7.3	PREPARATIONS.....	9
4.7.4	SOLDER INSTRUCTIONS.....	10
4.7.5	INSTALLATION GUIDE.....	10

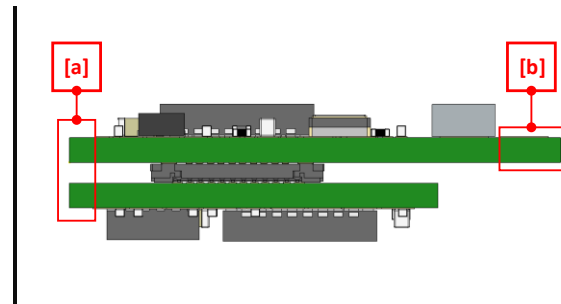
4.8	SCHEMATICS	13
4.8.1	TRANSCEIVER (TOP PCB).....	13
4.8.2	PERIPHERALS (BOTTOM PCB).....	13
5	MASTER	14
5.1	HARDWARE	14
5.1.1	USB-CONNECTOR.....	14
5.1.2	DCC-INPUTS.....	14
5.1.3	EXPANSION-PORTS.....	14
6	SOFTWARE.....	15
6.1	GENERAL	15
6.1.1	ICONS.....	15
6.1.2	START-UP PROCEDURE	16
6.2	GARAGE	17
6.2.1	OVERVIEW	17
6.2.2	CAR-TILE	17
6.2.3	CONTEXT-MENU	1
6.2.4	ADDING A CAR (DECODER).....	2
6.2.5	CHANGING THE DCC-ADDRESS.....	1
6.2.6	DECODER-UPDATE	2
6.3	DASHBOARD	1
6.3.1	OVERVIEW	1
6.3.2	TELEMETRY.....	2
6.3.3	CONTROL-MODE.....	2
6.3.4	TACHOMETER.....	1
6.4	DECODER-SETTINGS.....	3
6.4.1	OVERVIEW	3
6.4.2	GENERAL	4
6.4.3	PORT-MAPPING.....	1
6.4.4	CAR-LIGHTS.....	2
6.4.5	FUNCTIONS	1
6.4.6	MOTOR-CHARACTERISTICS	1
6.4.7	MOTOR-MAPPING.....	2
6.4.8	MOTOR-PROFILE	3

4 DECODER

4.1 HARDWARE

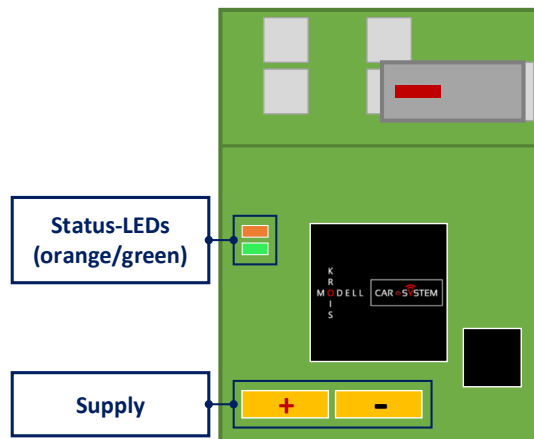
4.1.1 COMPONENTS

The decoder consists of two independent small PCB that are held together through subminiature connectors on the inner side. On delivery, the hardware comes always pre-assembled and is ready to use. However, if it makes integrating the decoder into a vehicle easier, the PCBs can be separated by the user and put back together later. When assembling the decoder, it is important to get the orientation of the PCBs right, this means, that the lower edges should align as shown in the picture [a].

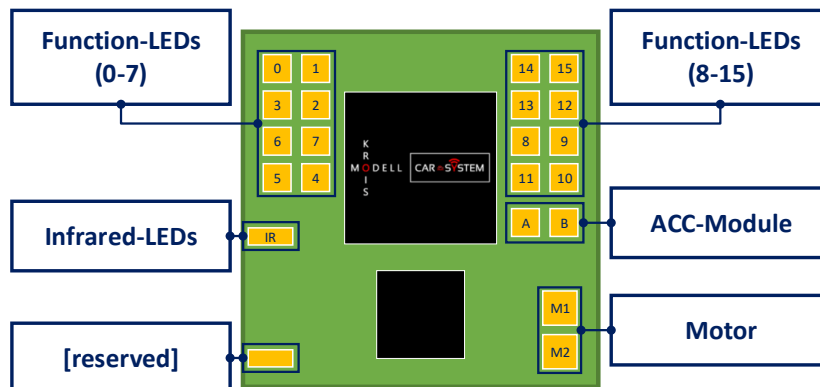


Mechanical modifications in any form (e.g. cutting or milling away the upper part of the PCB [b]) can cause irreparable damage to the decoder! Problems, that arise from such modifications are not covered by our warranty and must therefore be avoided.

4.1.2 TRANSCIEVER (TOP PCB)







4.1.3 PERIPHERALS (BOTTOM PCB)



4.2 STATUS-LEDs

The decoder accommodates two small LEDs (green and orange). Depending on the mode of operation these LEDs are used to signal information about the decoder status to the user. There is – of course – also the possibility to deactivate the LEDs to save energy and prevent unwanted lighting in the car.

4.2.1 LED-STATES (DESCRIPTION)

State	Meaning
	The Decoder is operational and ready to receive commands from the master
	A new firmware version is currently being transmitted
	No connection to the master is available (timeout)
	(currently not in use – should not occur)

4.3 SUPPLY (BATTERY)

4.3.1 VOLTAGE LEVELS

The decoder was designed to be used with lithium polymer (LiPo) batteries. The minimum voltage required by the decoder is therefore 3.8V. To prevent damage to the decoder, the voltage input must never exceed 5V. According to these limitations, it is not necessary to disconnect the battery from the decoder while charging (as the charging voltage of a single cell LiPo-battery should never exceed the allowed maximum voltage)

4.3.2 SOLDER INSTRUCTIONS

Both solder pads for the battery are located on lower edge of the top PCB (transceiver). They are – compared to all the other solder pads – rather large to allow the soldering of more than one wire if necessary. Directly above the right pad (negative input) there are five small capacitors – special care should be taken to make sure that these components are not desoldered or bridged with solder when connecting the wire(s).

4.3.3 POWER SWITCH

Although the decoder automatically enters a power saving mode after a certain period of inactivity, it is advisable to electrically disconnect the battery when it is no longer in use. The simplest possibility to for this is a switch, that is to be connected between the positive output of the battery and the positive input of the decoder.

4.3.4 IMPORTANT INSTRUCTIONS

Polarity

When connecting the battery, it is very important to make sure that the polarity is correct. Reversing the polarity can lead to irreparable damage and must therefore be avoided at all costs!

4.4 PROPULSION (MOTOR)

4.4.1 POLARITY

As the polarity of the motor can easily be switch trough the configuration software, there are no requirements regarding the connection between the motor and the corresponding pads on the decoder.

4.5 FUNCTION-LEDS

4.5.1 POLARITY

All LED-outputs are “open drain”-outputs, this means, that they are either connected to ground or left floating. Therefore, the anodes of the LEDs (positive) can be connected directly to the supply voltage of the decoder (i.e. the battery). The cathodes of the LEDs (negative) must be connected to the corresponding pads of the decoder. It is recommended to combine all LED-anodes to one single wire that is than connected directly to the battery as this greatly simplifies the wiring in the car.

4.5.2 MAPPING

The LED-outputs can be tested, configured and mapped at any time in the configuration software. It is therefore not necessary to write down or remember to which port or pad an LED is connect (although it can be handy at times) – just connect all LEDs and user the graphical interface to set the up as required.

4.5.3 OUTPUT-CURRENT/BRIGHTNESS CONFIGURATION

Each LED-output has a fixed current limit that is set to 20mA. This limit cannot be set by the user, but as the brightness of every output can be controlled through PWM (in 255 steps) anyways, it is also not necessary to change it. The only thing that should be noted is, that “low current” LEDs (that have a current rating of 5mA or less) can be damaged when they run on a high brightness setting for an extended period.

4.5.4 WIRING

As the outputs are constant current sources, there are different possibilities to wire the LEDs (depending on the color/forward voltage and the available supply voltage). The following descriptions reference the configurations shown in the “Schematics”-Section of the document.

❖ One LED (simplest layout) [a]

If a single LED is connected to the decoders as shown in [a], the diode current will be 20mA. The forward voltage of the LED is irrelevant in this case, as even blue LEDs (with a forward voltage of 3V or more) can be operated without problems.

❖ Two (or more) LEDs in parallel [b]

In respect to the forward voltage, this configuration is comparable to [a]. In general, the LEDs that are put in parallel should be of the same type (regarding color/voltage), as different forward voltages can cause unexpected variations in brightness. The current per LED can be calculated easily by dividing the total current (20mA) by the number of LEDs in parallel. If “low current” LEDs are to be used, this configuration should be the way to – by putting more of these LEDs in parallel, the current through a single LED can be held low enough to prevent damage to the at higher brightness levels.

❖ Two (or more) LEDs in series [c]

Depending on the LED color (forward voltage), it is also possible to put two (or more) LEDs in series. In this configuration, the current is still 20mA (maximum output current). Special care should be taken to ensure that the required total forward voltage of the LEDs in series (i.e. the sum of the individual forward voltages) is lower than the available supply voltage. If – for example – two red LEDs (with a typical forward voltage of 1.6-2.2V) are used, the required total voltage would be about 3.2V. This is of course lower than the minimum supply voltage of the decoder (3.8V) and will therefore work without problems. However, the same configuration with blue or white LEDs (that have a typical forward of about 3V) will not work, as the required total voltage lies far beyond the allowed maximum supply voltage of the decoder (5V).

❖ Two (or more) LEDs in series and parallel [d]

This configuration is basically a combination of [b] and [c] – the total current is divided by the number of LED branches in the circuit and the maximum forward voltage of a single branch must not exceed the available supply voltage. It is recommended (but not necessary) to use only LEDs of the same color/voltage for this configuration.

❖ Special case: Two different supply voltages

Depending on the available space in the car and the required lighting configuration, it is possible to include a second, higher voltage supply net to power to LEDs. In this case, the anodes (“+”) of the LEDs can be connect to the higher supply voltage (e.g. a 2-cell lipo battery) while the cathodes (“-“) are still connected directly to the decoder. Of course, the negative lines of both supply nets must be connected for this configuration to work correctly (the positive supply lines must remain separate). With the higher LED voltage, it is then possible to have more LEDs in series, leading to reduced current consumption and even higher efficiency due to the higher total forward voltage.

4.5.5 IMPORTANT INSTRUCTIONS

Safety Features

Short circuits, inversely poled LEDs and connected outputs lead to the automatic deactivation of the effected outputs to prevent damage to the decoder. Nevertheless, special care should be taken to prevent such problems from happening in the first place, as these safety features might not always be enough to protect the hardware.

4.6 INFRARED-LEDs (DISTANCE CONTROL – COMPATIBILITY MODE)

4.6.1 PURPOSE

The infrared (IR) LEDs are only required if our system is run in combination with another, IR-based system. By mounting IR-LEDs at the back of a car that runs with our decoder and connecting them to the appropriate pad, an IR-signal is generated that causes cars with other decoders to slow down or stop when approaching. The diodes should be mounted in the same height as the phototransistors on the other cars to allow a reliable transmission of the IR-signal.

4.6.2 INFRARED DIODES

The decoder requires two IR-LEDs in series (!) to be connected to its IR-output if this feature is used. Connecting only a single LED will cause damage to either the decoder or the LED, as the passive components are not scaled for the low forward voltage! The size of the used LEDs is of no relevance if the forward voltage (of a single diode) is about 1.4V (resulting in a total forward voltage of about 3V when put in series).

4.6.3 SOLDER INSTRUCTIONS

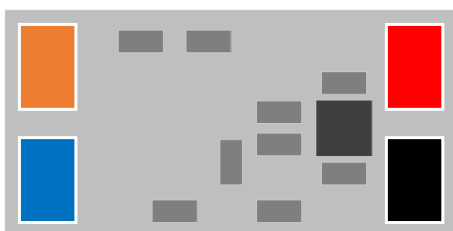
On the bottom-left side of the peripherals-PCB there is another, very small PCB soldered on top. The purpose of the PCB is to filter unwanted noise from the signal output and to amplify the output-current of the decoder itself. There are two solder pads located on it as well: on the left (IR-output) and one on the right (reserved) edge. When soldering a wire to one of these pads, special care must be taken to prevent short circuits between the solder pad(s) and the components on the PCB.





4.7 DISTANCE CONTROL (KROIS-MODELL)

4.7.1 INTRODUCTION

The decoder comes with a little module that replaces the known infrared-technology that has long been the only option for in-car distance control. With this module, no modifications or components on the rear of car are necessary, as it only must be integrated into to front. All steps necessary to do this as well as some “best practice” tips for achieving the best performance are presented on the following pages.

4.7.2 MODULE (OVERVIEW)



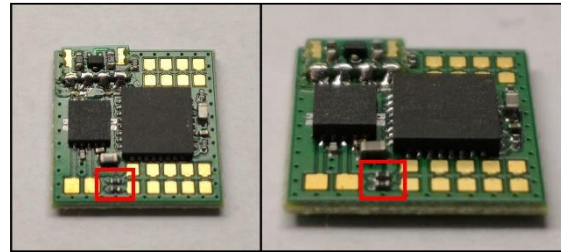
Pad	Function
	Supply („+“) – maximum voltage: 5V
	Ground („-“) – negative battery terminal
	Signal [A]
	Signal [B]

4.7.3 PREPARATIONS

Before the ACC-module can be connected/soldered to the decoder, two resistors on the bottom-PCB need to be removed (de-soldered) first. This task should be performed with some caution and a relatively fine soldering tip for precise soldering. The modification itself isn't very difficult. However, if one doesn't want to do it himself for whatever reason, please don't hesitate to contact us – we will be happy to help.

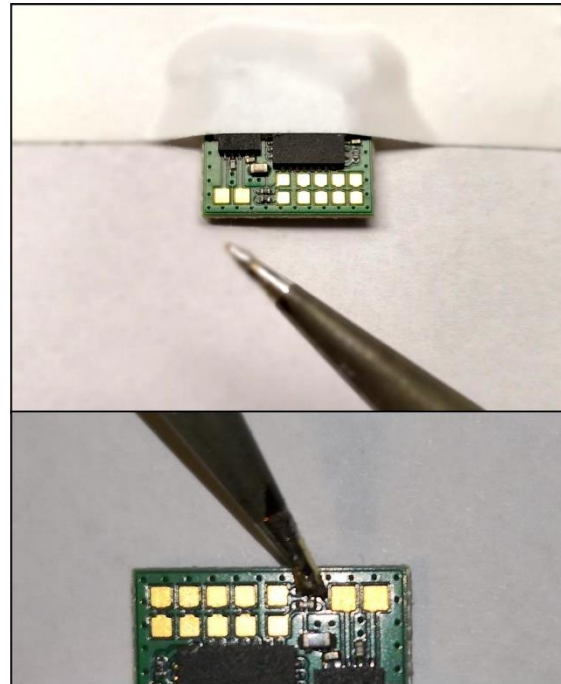
❖ Locating the resistors

Both resistors are located on the right edge of the peripheral-PCB (bottom). They lie parallel to each other and are easy to reach as they are not surrounded by other components.



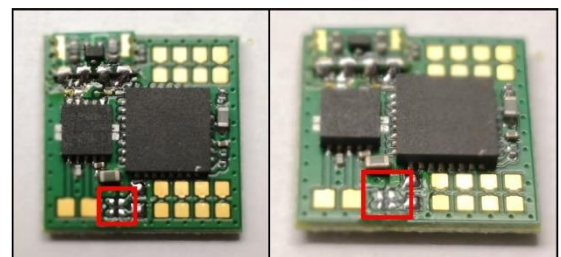
❖ Removal using a soldering iron

Before starting the removal process, it is recommended to use a temperature resistive tape to keep the PCB in place. As the de-soldering of components requires enough flux to be available, start by putting some solder on the tip of your soldering iron. With all these preparations in place, the removal of the resistors should be straightforward: Put the tip of the soldering iron onto the components and wait for a few seconds until enough heat has been transferred. As soon as the resistors begin to move, carefully pull them to the edge of the PCB (using the soldering iron) to remove them.



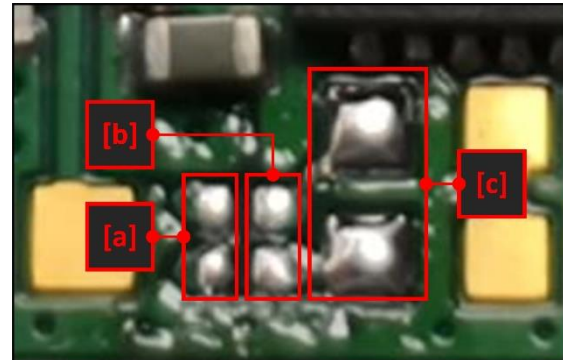
❖ Examination of the pads

After the resistors have been removed, the underlying solder pads should be visible. Using bright light and a magnifier (if available), make sure that no unwanted solder bridges have been created during the de-soldering process.



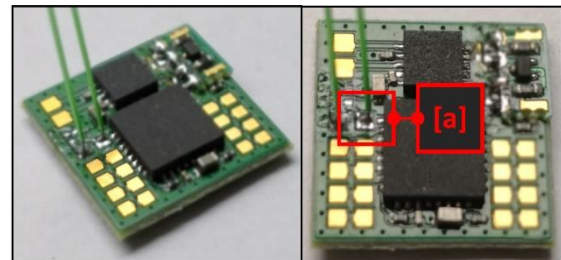
❖ Possible Error Sources

The highlighted solder pads in [a] are internally connected – a solder bridge between them doesn't necessarily need to be removed. This doesn't apply to the pads in [b] and [c], however. If a solder bridge exists between those pads, it must be removed before the module can be connected. In most cases, reheating the pads using a soldering iron is enough to remove unwanted bridges.



4.7.4 SOLDER INSTRUCTIONS

When connecting the signal lines of the module to the decoder, it is advisable to solder the wires "vertically" (as shown in the picture). Using this soldering technique, the probability of short circuits, unwanted solder bridges or other possible complications is greatly reduced. Especially the area [a] should be inspected properly to make sure that there is no connection between the pad and the IC-pins.



4.7.5 INSTALLATION GUIDE

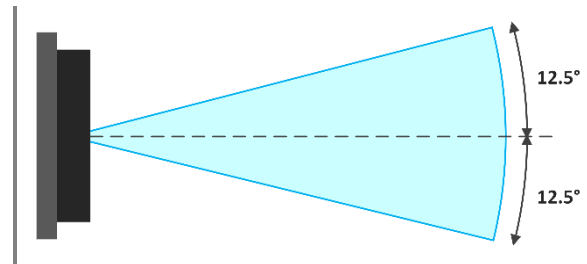
For the distance control and the ACC-module to work as expected, certain requirements must be met when installing the sensor: As the distance control not only detects other cars but every kind of obstacle, the alignment of the module is very important. If the angle is too low (i.e. the sensor looks "downwards"), the road will be detected as an obstacle and the car will stop moving or won't move at all. If, however, the angle is too high, obstacles will be out of the sensors field-of-view and therefore the car may not stop quickly enough to avoid a collision. Finding a reasonable compromise in terms of the installation point and angle is crucial to guarantee a seamless operation of the distance control.

Final Mounting

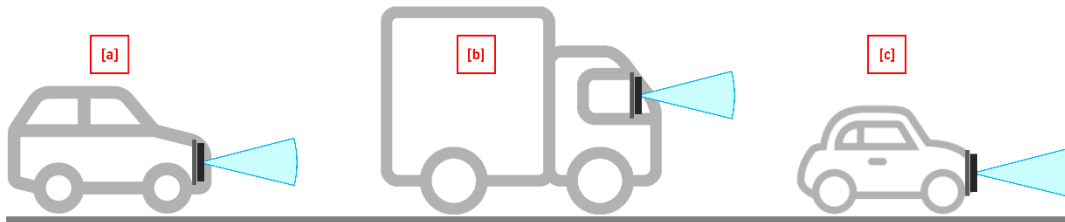
Because of the requirements mentioned above, the module should not be fixed to the car before the distance control has been tested! In other words: Never expect the module to be perfectly aligned without first testing the function. Most of the time it will be necessary to slightly re-adjust the orientation. In addition, it should be noted that the installation requirements differ from car to car and must be worked out individually.

❖ **Module Field-of-View**

The emitted signal from the ACC-module spreads conically with angle of 12.5° (in every direction). As a result, the total field-of-view is around 25° and is further limited by the maximum ranging distance of around 200m.

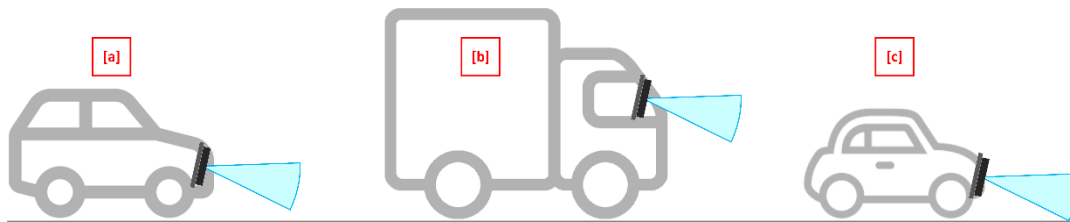


❖ **Mounting Angle = 0°**



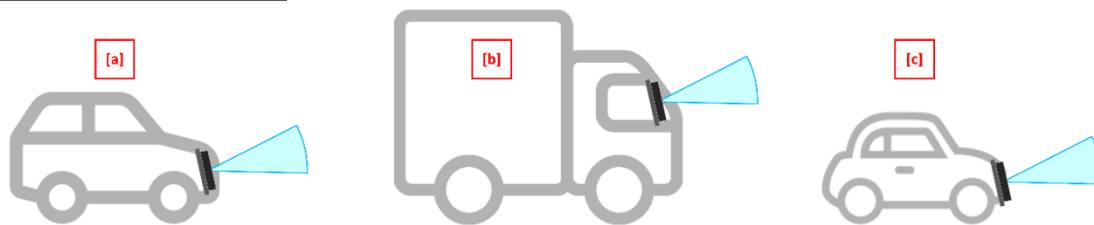
Depending on the road surface and the vehicle type, a mounting angle of 0° can cause malfunctioning of the distance control. As illustrated in the examples [a] and [c], the cone emitted from the module will read the road surface after a relatively short distance. By mounting the module higher above the surface (as in [b]), this problem can be avoided to a certain degree.

❖ **Mounting Angle = -12.5°**



A negative mounting angle is unsuitable for most smaller vehicles as the effective distance is further reduced (see examples [a] and [c]). This doesn't necessarily apply to taller vehicles, however. Depending on the exact sensor position, it might be necessary for the module to be mounted in a slightly negative angle to allow the detection of smaller vehicles in front (see [b]).

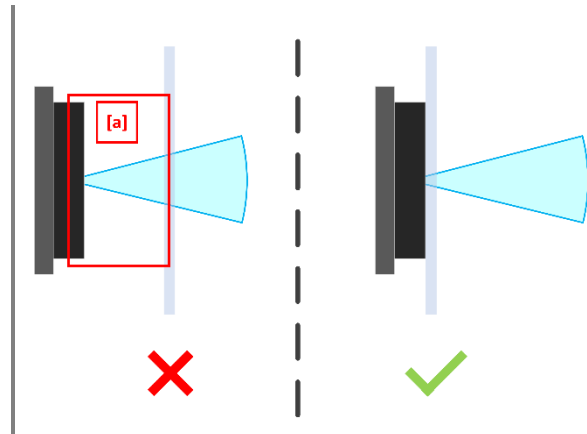
❖ **Mounting Angle = 12.5°**



For smaller vehicles ([a] and [c]), a positive mounting angle is the way to go. By pointing the module slightly upwards, the road can no longer be detected as an obstacle even if it runs relatively close below the sensor. This, however, might not be the most ideal way for taller vehicles ([b]) as smaller targets in front might be missed or detected too late.

❖ Mounting behind Windscreen

The module can be placed inside a car as well, allowing a nearly invisible integration. When placing the sensor behind cover glass, it is important to make sure that the air gap (highlighted in [a]) is kept as small as possible. In general, the maximum distance should not exceed 1-2mm. It is, of course, not required that the sensor is mounted completely parallel to the glass if the air gap is held small enough.

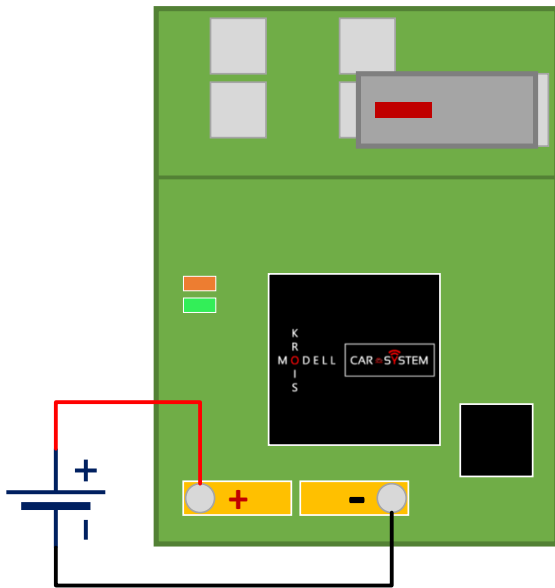


Handling and Mounting Instructions

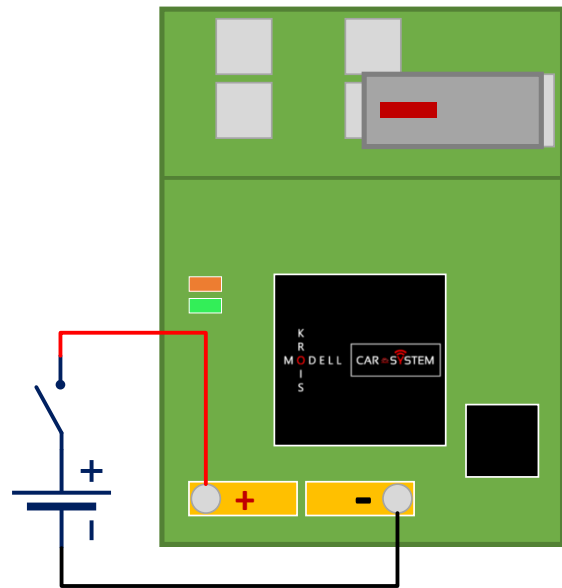
The ACC-Module contains sensitive optical components that must be handled with the appropriate amount of caution and care. This especially applies to the front side of the module: As the sensor openings are not protected or sealed against fluids or particles of any sort, glue and other chemicals must never get in touch with it. The coating of the module with paint is also not recommended, as the paint will most likely make the optics unusable. If the sensor is to be mounted using glue, it must only be applied to the backside of the module. The PCB is not designed to withstand mechanical forces – modifications to the module like milling or cutting of the PCB can result in irreparable damage that is not covered by our warranty!

4.8 SCHEMATICS

4.8.1 TRANSCEIVER (TOP PCB)

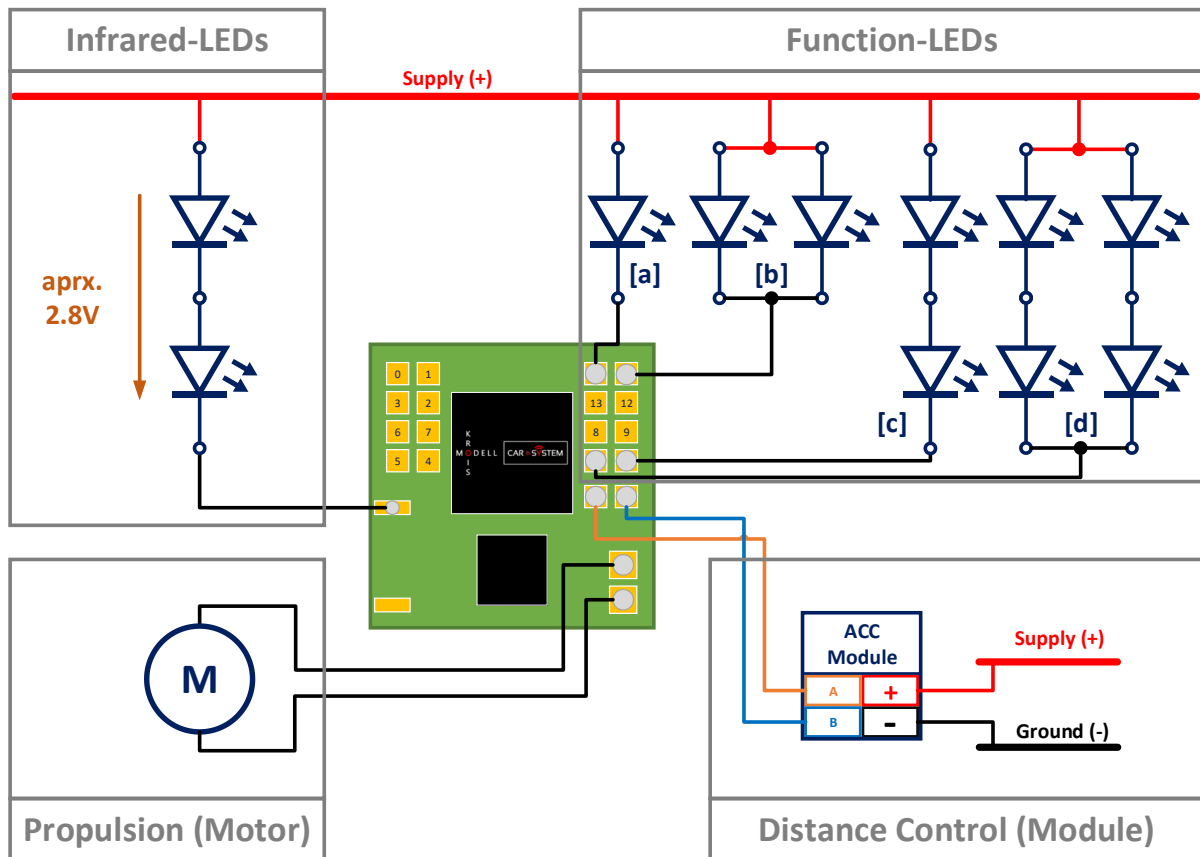


Direct Supply
(decoder always on)



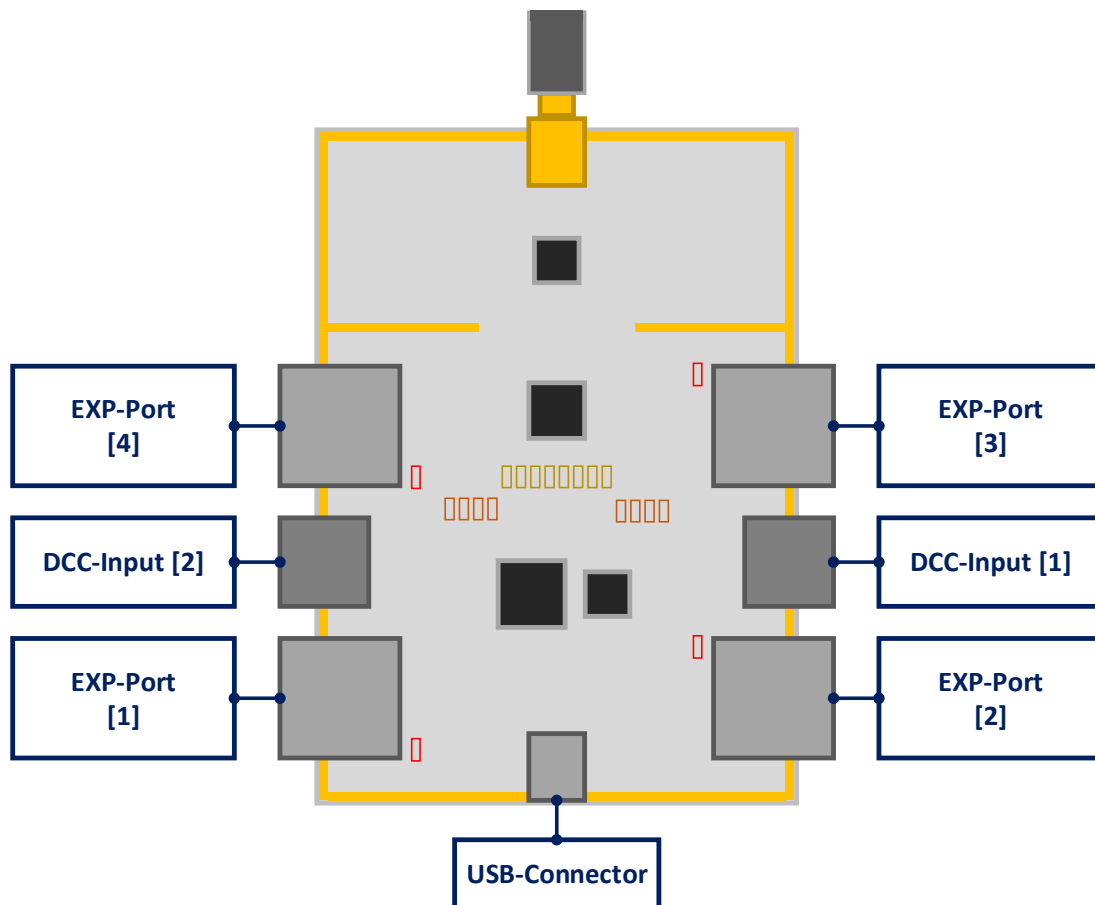
Power Switch Configuration
(recommended)

4.8.2 PERIPHERALS (BOTTOM PCB)



5 MASTER

5.1 HARDWARE



5.1.1 USB-CONNECTOR

The primary purpose of the USB-connector USB-Anschluss (type Mini-B) is to power the device. As the master doesn't use more than 500mA when operational, it can be powered from a standard USB 2.0 port. If the device is to be used in "standalone"-mode, a generic USB-charger (e.g. from a smartphone) is recommended.

5.1.2 DCC-INPUTS

Both inputs are electrically isolated from the sensitive electronics on the device. This means, that voltage spikes or short circuits on the DCC-signal line do not affect the master. As a result, powering the master through the DCC-signal is not possible! The numbering of the inputs is only for identification purposes – both inputs can be used independently (or simultaneously) at any time.



















5.1.3 EXPANSION-PORTS

The four RJ45-connectors allow future expansions of the hardware without replacing the device itself. Every connector has two LEDs for status indication and can be turned on or off through software. Please note that although the connector uses the same standard, these ports do not support the ethernet protocol and can therefore not be connected to a network!

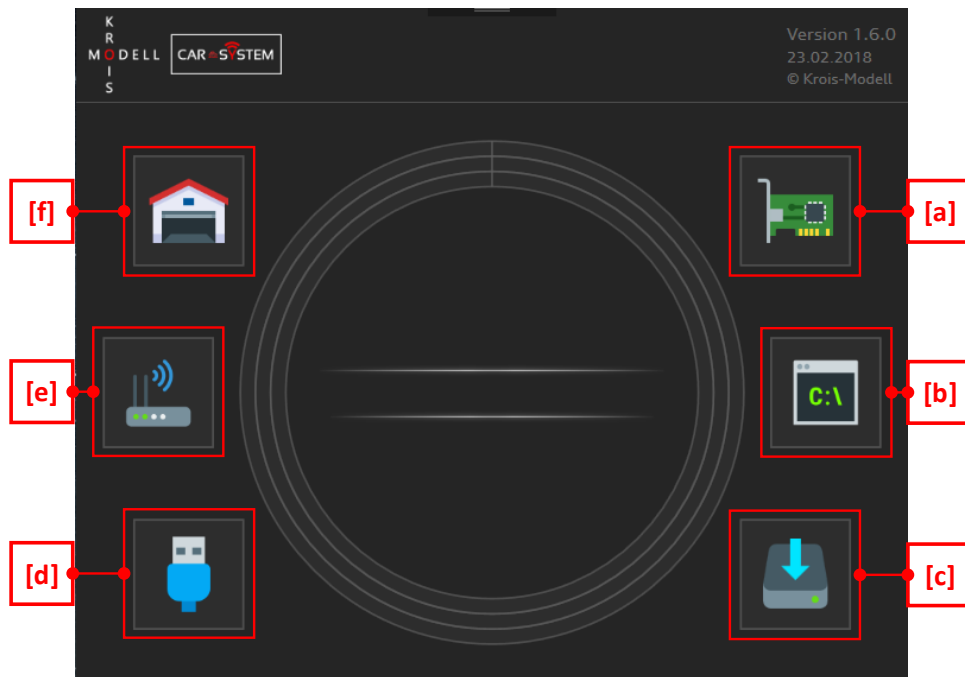
6 SOFTWARE

6.1 GENERAL

6.1.1 ICONS

	distance control		motor voltage		settings
	address change		erase memory		internet connection
	backwards (driving direction)		transmit firmware		O.K. / success
	forwards (driving direction)		start update		color selection
	back (navigation)		verfiy update		expansion ports
	battery		abort / stop		add / create
	documentation		reset / delete		reset
	car lights		characteristic curve		master
	fog-light		motor settings		save
	hazard-lights		warning / error		search / read
	high beam		feedback		settings
	indicator (left)		vehicle / decoder		motor characteristics
	indicator (right)		garage		emergency stop
	parking-lights		lights (F0) - ON		daytime running lights
	rear fog-lights		lights (F0) - OFF		general settings
	outputs / mapping		help		speed step speedometer
	software		action in progress		temperature
	PC-control (mode)		infrared		update
	dashboard		function-preset(s)		USB-connection
	temperature (decoder)		state = OFF PWM-OFF value		information
	voltage (decoder)		state = ON PWM-ON value		

6.1.2 START-UP PROCEDURE



❖ [a] Internet (Verify Connection)

First, the software checks if there is an active internet connection available. This is only required on the first run of the program as it must download all the required resources. All consecutive starts can then happen without internet connection (offline-mode).

❖ [b] Software (Check for Updates)

If an active internet connection has been found, the software automatically checks for updates and starts the installation of a new version if available.

❖ [c] Load Resources

The third step of the start-up is the download of various software resources like hardware updates, presets and the in-app documentation.

❖ [d] Check USB-Connection

For the software to start it is required to have the master connected to the PC via USB. In addition, the correct USB-driver must be installed (available on our website) for the hardware to be discovered by the program.

❖ [e] Initialize Master

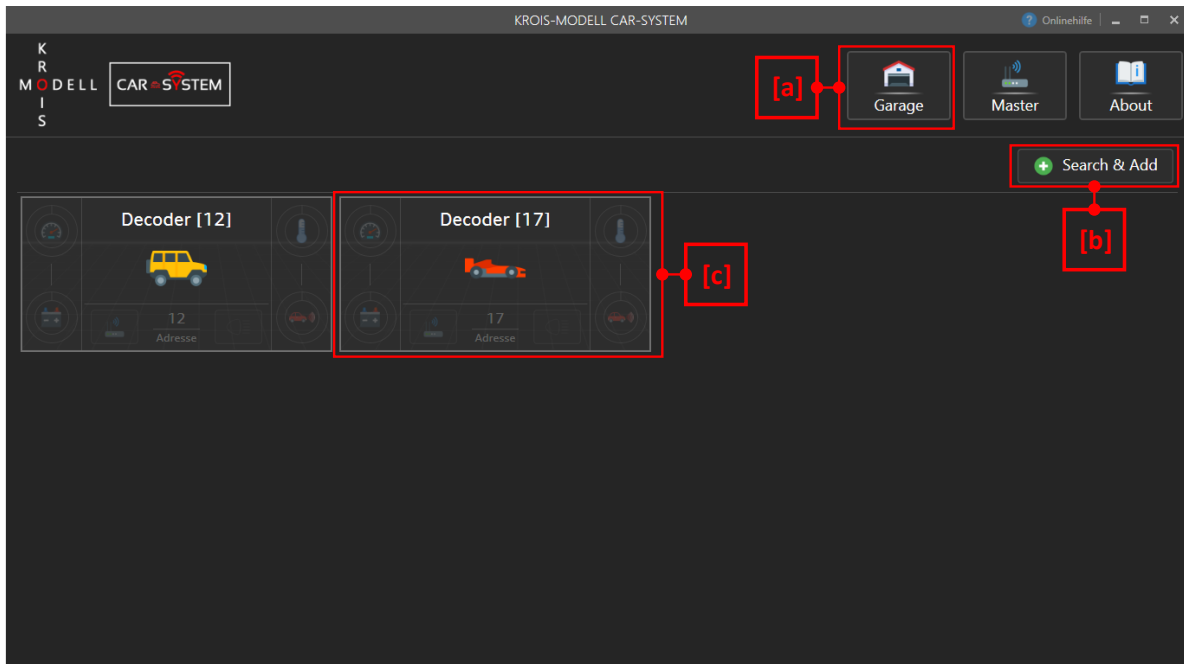
If the hardware has been found, the software performs some health checks on the master and – if available – transmits and installs the new firmware to the device. This step can take some time (depending on the system configuration).

❖ [f] Load Garage

After all the previous steps have been completed, the locally stored cars are loaded, the DCC-inputs of the master are re-activated again, and the main window of the program opens. The master then scans for active decoders and the system is ready to use.

6.2 GARAGE

6.2.1 OVERVIEW

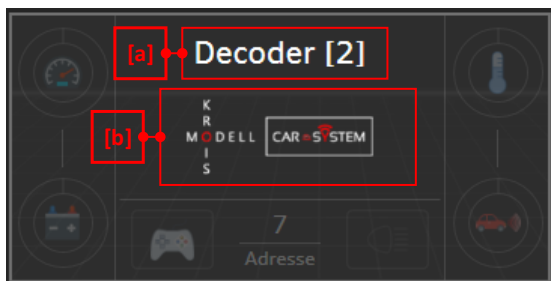


The garage is essentially the landing page of the program – it enables the user to manage (add/delete) cars and monitor the status of all connected decoders. To return to the garage-view, a simple click on the corresponding button in the navigation bar [a] is sufficient. The button labeled “Add & Search” [b] opens a wizard that can be used to find active decoders and subsequently add them to the garage. All cars that are stored on the PC are displayed as a tile [c] in the garage. The appearance and the functionality of the tile change depending on the operational state.

6.2.2 CAR-TILE

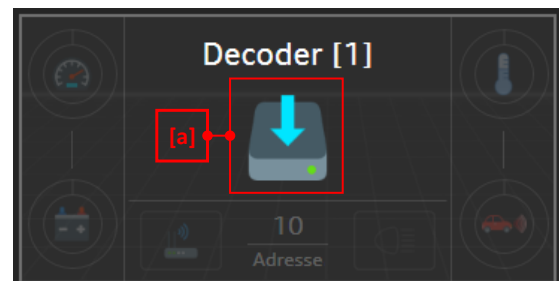
The tiles display information about the state of the decoders/cars and they also offer different functionalities regarding management, control and decoder settings.

❖ Offline



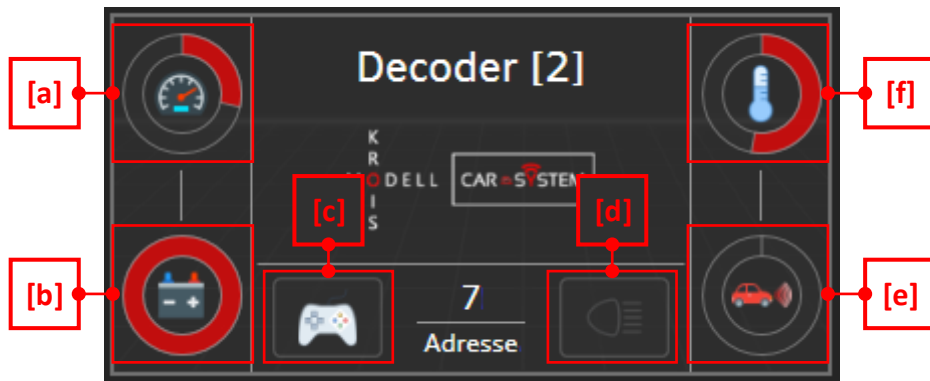
When a decoder is offline, the corresponding tile only shows the name [a] as well as the icon [b] of the car. All options in the context menu, except for “delete”, are deactivated in this state.

❖ Update



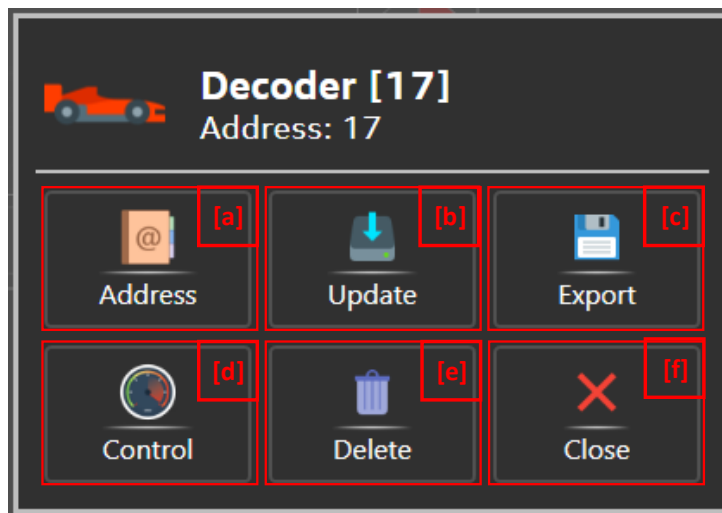
When a decoder is online but has an outdated firmware installed, the car icon is replaced with the update icon [a]. To start the firmware update, simply click on the tile (using the left or right mouse button).

❖ Online



When a decoder is online and has the latest firmware installed, the tile switches to the “active” view. The four circular bars show a quick summary of the most important decoder parameters: [a] shows the set speed step (0-28), [b] shows the battery voltage (ranging from 3 - 5V), [e] shows the measured distance from the ACC-module (ranging from 0 - 255mm) and [f] shows the decoder temperature (ranging from 0-70°C). The left button [c] can be used to switch the control mode between DCC- (“router”) or PC- (“gamepad”) mode. In DCC-mode, the right button [d] is read only and displays the state of function F0 (lights). In PC-mode, it can additionally be used to change the state of function F0 (lights). By clicking on the tile (using the left mouse button and outside of [c] and [d]), the dashboard of the corresponding car is loaded.

6.2.3 CONTEXT-MENU



The context menu of a car can be opened by right-clicking on the corresponding tile in the garage. If the tile is in “offline”-mode, only the buttons [e] (deleting the car from the garage and the master) and [f] (closing the context menu) are activated. When a firmware update is pending, the context menu isn’t displayed as the update wizard [b] is automatically started as soon as the tile is clicked (opening the update wizard when the firmware is up-to-date simply causes it to display the currently installed version). The buttons [a], [c] and [d] require the decoder to be online and up-to-date: [a] opens the address-change wizard (that is used to change the DCC-address of a car/decoder), [c] opens the car-export wizard (that allows the export/backup of car settings) and [d] switches to the dashboard view (that is used to control and configure the car).

6.2.4 ADDING A CAR (DECODER)

❖ Overview

The wizard contains three important areas that serve different purposes:

[a] Scan

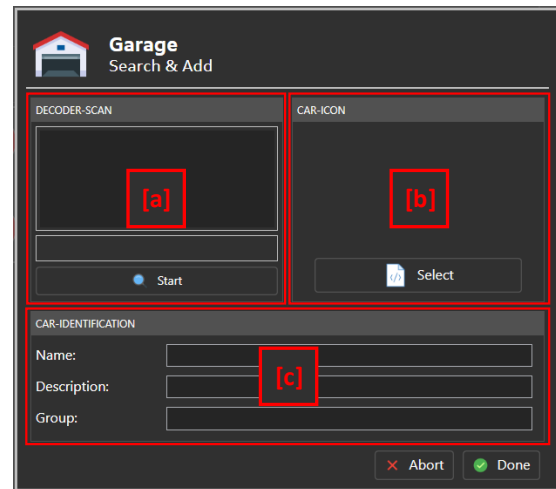
Is used to scan the address range (3-4094) for active decoders.

[b] Car-Icon

Is used to choose/add an icon for the car (displayed in the garage).

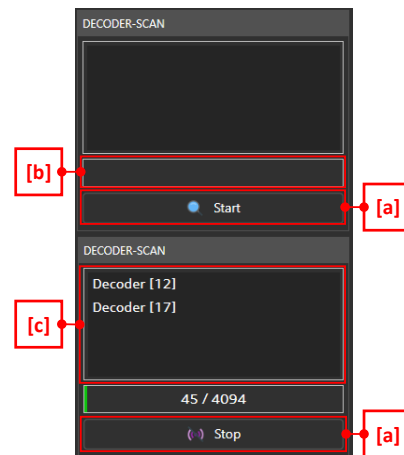
[c] Identification

Is used to specify identification data for the car (display in the garage as well as in the dashboard).



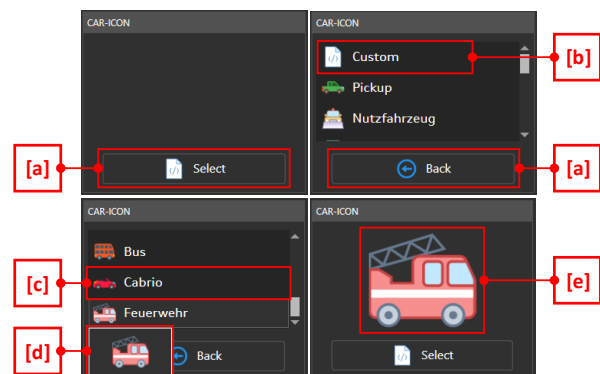
❖ Search & Select a Decoder

To start searching for decoders, simply click on the “start”-button [a] below. The scan progress is displayed in the bar [b] above the button, found decoders are listed as “Decoder [X]” (where “X” is replaced by the DCC-address). Clicking on an entry [c] automatically selects the address and stops the scan. Alternatively, the scan can be stopped at any time by clicking on the control button [b] again.



❖ Choosing an Icon

The “Select”-button [a] opens the icon list (that can be left at any time by clicking [a] once more). If a custom icon file should be loaded, select the first entry [b] in the list and navigate to the desired file. There are also a few standard icons that can be used to get started (the icon can of course also be changed in the car settings later). To select a pre-defined icon, choose an icon group [c] and select the symbol [d] by clicking on it. A preview of the selected icon is shown in [e] after the list has been closed again.



❖ Fill in Identification-Data

Parameter	Description
name	display name of the car (e.g. on the car tile)
description	details about the car (only visible in the dashboard)
group	classification of the car (used for grouping in the garage)

Decoder-Supervision

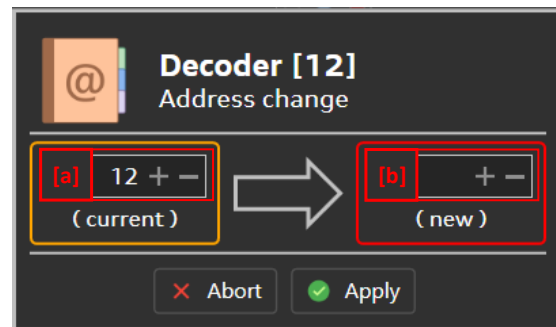
As soon as the wizard is loaded, the supervision of the connected decoders is no longer active. This means, that DCC-commands are no longer transmitted to the cars to allow for faster scanning. As a result, all active cars should be in a safe state (ideally stopped) when a new decoder is to be added to the garage. This state is automatically left when the wizard is closed.

6.2.5 CHANGING THE DCC-ADDRESS

❖ Overview

Parameter	Description
address	DCC-address of the car/decoder. The allowed address range spans from 3 to 4094 . In delivery condition, the address is always set to 3. As two decoders with the same address must never be active at the same time, it is recommended to change the address right after the decoder has been added to the garage.

The interface used for changing the address is designed with simplicity in mind: the current address is displayed in the left box [a] and the new (target-) address can be set in the right box [b]. The new address can either be entered directly or using the “+”-“–”-buttons on the right edge of the field.



❖ Setting a new Address

Every address entered in the „new“-field is automatically checked for its validity. The result of this verification is indicated using the border around the “new“-field. An **orange** border means that entered address is identical to the current address of the decoder, **red** shows that the entered address is already in use (i.e. there is already a car with that address in the garage). When the border turns **green**, the entered address is valid, and the requested change can be performed by clicking the “save“-button below. Please note that the “save“-button is only enabled when the border around the “new“-field is green.

Duration

As the address change is a relatively complex process, it might take a few seconds until the software has finished all necessary steps. During this period, the software will not react to any kind of user input and the interface will be temporarily disabled. This state is automatically left as soon as the address change has been completed.

Decoder-Supervision

Due to some internal requirements the master will be reset during the address change. As this can cause temporary loss of some DCC-packets, it is advisable to make sure that all cars are in a safe state (ideally stopped) prior to changing the address of a decoder.

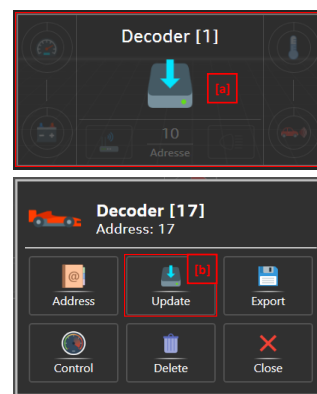
(Re-)Initialization

After the address change has been completed, the affected car tile (and maybe some others as well) might change to the “offline”-state for a few seconds. As soon as the master rediscovers these decoders, the tiles will switch to “online”-mode again and the decoders are ready to use.

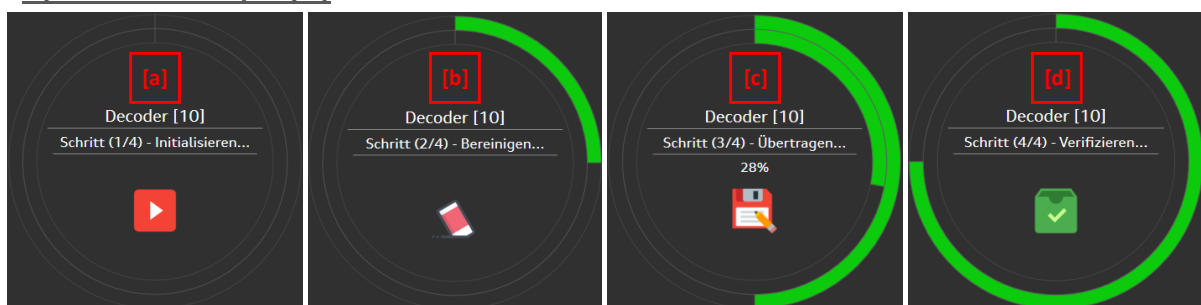
6.2.6 DECODER-UPDATE

❖ Overview

The update wizard can be started/opened in different ways: if a car tile shows the update icon [a] a simple click on the tile (with either the left or the right mouse button) opens the wizard and starts the pending firmware update. If no update is pending, the wizard can be opened through the context menu of a tile by clicking on the update button [b].



❖ Update-Process (Steps)



Every update consists of four steps that are processes automatically and cannot be influenced by the user. Nevertheless, these steps are explained in a little more detail for better understanding and troubleshooting purposes. As every update comes with certain risks special care needs to be taken to minimize these risks as much as possible.

❖ **Step 1: Initialize**

The update mode on the decoder is started, followed by different checks regarding the decoder state and various other requirements.

❖ **Step 2: Clean & Prepare**

The current settings are loaded to the backup space and the old firmware is cleared from the decoder.

❖ **Step 3: Transmit**

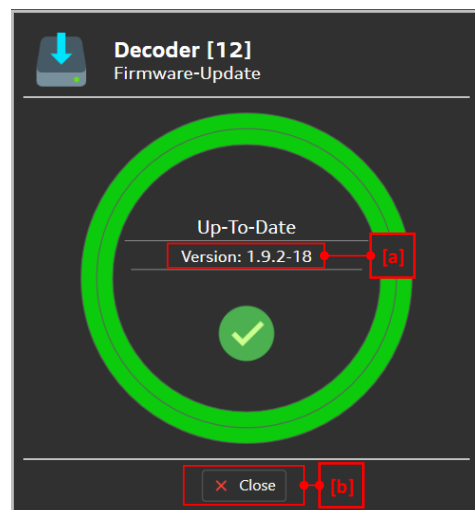
The new firmware file is now transmitted to the decoder. During this process, the progress is displayed in the wizard.

❖ **Step 4: Verify**

The last step is the verification of the new firmware – if everything has been transmitted correctly, the update mode is left, and the decoder will be restarted automatically.

❖ **Up-To-Date**

After the update has been completed successfully, the wizard shows the current version number [a]. To close the windows, simply click the button [b] on the bottom edge. If the firmware version needs to be checked at a later point in time, simply load the update wizard from the context menu.



Decoder-Supervision

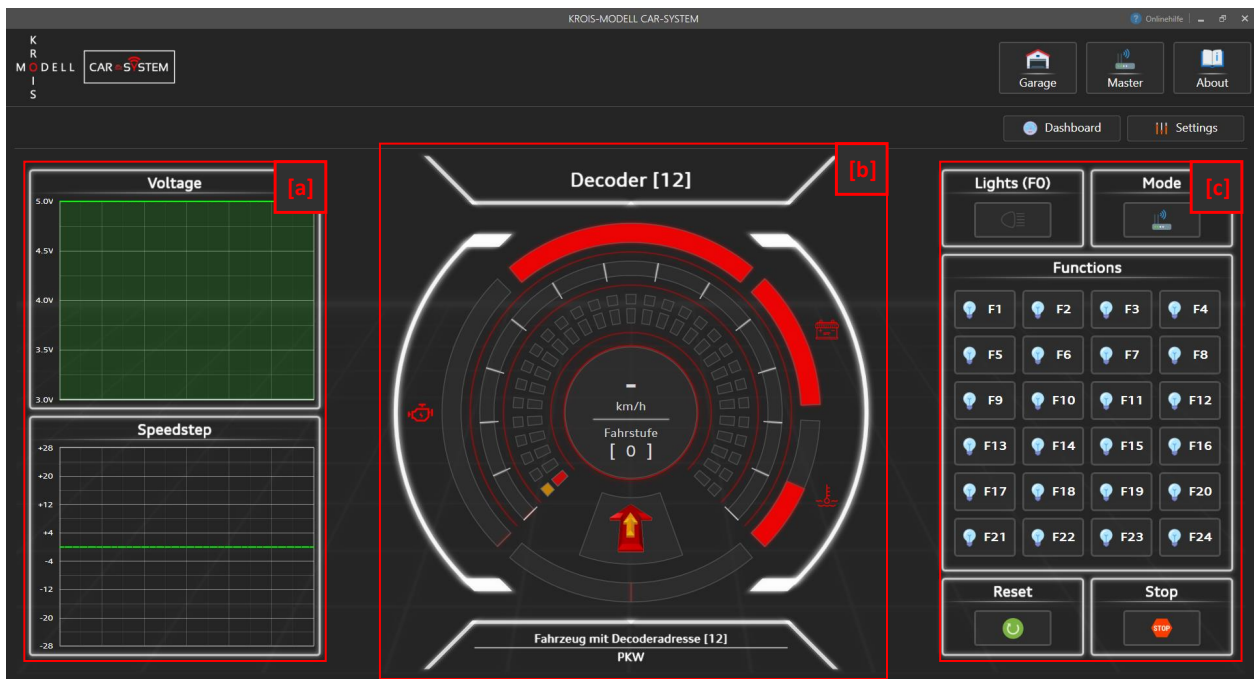
For faster transmission speeds (i.e. to reduce the total update duration) the decoder supervision is disabled during an update. This means, that DCC-packets and other commands are not transmitted to the cars/decoders during this period! Before starting an update, it is therefore necessary to make sure that all cars are in a safe state (ideally stopped) to avoid unexpected problems. The supervision is automatically reactivated as soon as the update is finished.

During the Update

Before starting an update, the user must make sure that the battery is sufficiently charged to avoid a loss of power during the update process. It is highly recommended to connect the car/decoder to a charger prior to starting an update as this eliminates the danger of losing power during critical update steps. This especially applies to the fourth step (“verify”) – during the verification, the supply to the decoder must never be cut off, as this can lead to irreparable damage to the decoder!

6.3 DASHBOARD

6.3.1 OVERVIEW



When a car is selected (either by directly clicking on the corresponding tile or by selecting “control” in the context menu) the dashboard view is loaded. The dashboard can be left at any time by selecting an item (e.g. “Garage”) in the navigation bar.

The dashboard itself can be divided in three separate areas that all serve a different purpose:

❖ [a] Telemetry / Parameter-History

On the left side there are two plots that show the history of different decoder parameters. The latest value is always the one on the right edge. While the scaling in y-direction is fixed, the scaling in x-direction (time) is dependent on various parameters.

❖ [b] Tachometer

The center of the dashboard consists of a large tachometer that has different functionalities depending on the current control mode. Above and below the tachometer identification data such as the name, description and group of the selected car/decoder is displayed.

❖ [c] Mode & Functions

The functionalities of the buttons on the right side are also dependent on the current control mode. Generally, these controls provide means to enable or disable the various functions of the selected car.

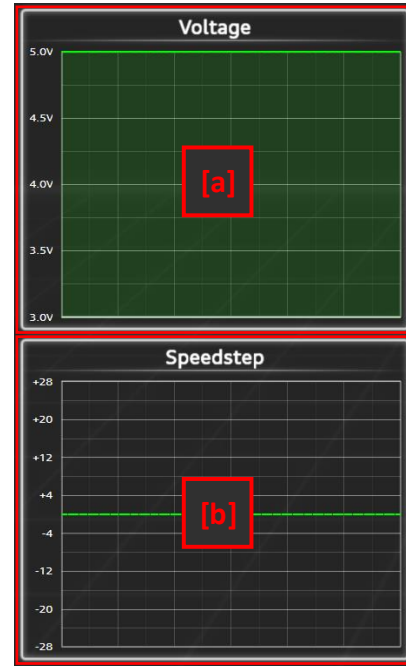
6.3.2 TELEMETRY

❖ [a] Voltage

For the best resolution, the scaling on the y-axis ranges from 3.0V to 5.0V. This results from the fact, that the minimum supply voltage for the decoder lies above 3V – therefore it makes no sense to start the scaling of the y-axis at lower voltages (or even 0V).

❖ [b] Speed Step

The center of the y-axis corresponds to speed step 0/stop (in both directions). Positive speeds (forwards) are plotted above the center line, negative speeds (backwards) are plotted below. As the plot shows the current speed, changes to the target speed might not be visible immediately (depending on the acceleration/braking settings).



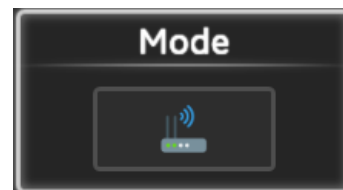
The scaling of the x-axis is variable, as the time between two samples is not exactly predictable and will depend of different factors. Therefore, the axis labeling has been omitted.

6.3.3 CONTROL-MODE

Using the configuration software, it is possible to control cars directly from the PC without the need for any additional hardware. Right after the start, decoders “listen” only to commands from the DCC signal input(s). When the master is connected to a PC, the control mode can be changed for every car/decoder independently through the “mode”-button in the dashboard.

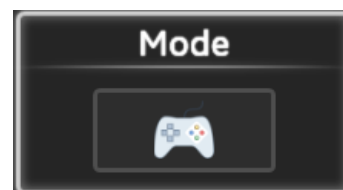
❖ DCC-Mode

In this mode, the car only reacts to commands issued over one of the DCC-inputs. The software displays the currently set parameters (speed, direction, functions) in the dashboard but cannot influence the car’s behavior.



❖ PC-Mode

In this mode, the car can (only) be controlled through the PC-software. Commands that are transmitted through DCC are ignored until the mode is switched by the user or the master is reset.



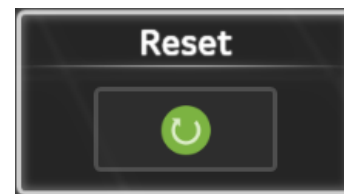
❖ **Lights (F0)**

The “Lights“-button corresponds to the F0-key on any other (manual) controller. When the function is activated, the icon inside the button is shown in green, otherwise it is greyed out. Depending on the control mode, the button is either read-only (DCC-mode) or can be used to control the function (PC-mode).



❖ **Reset**

This button resets the decoder to its default state: the speed step is set to 0 and all functions (F0-F24) are deactivated. In addition, the control mode is automatically set to PC-mode if necessary as the decoder parameters cannot be changed in DCC-mode.



❖ **Stop**

This button resets the speed step to 0, causing the car to stop immediately. All functions keep their current state. In addition, the control mode is automatically set to PC-mode if necessary as the decoder parameters cannot be changed in DCC-mode.



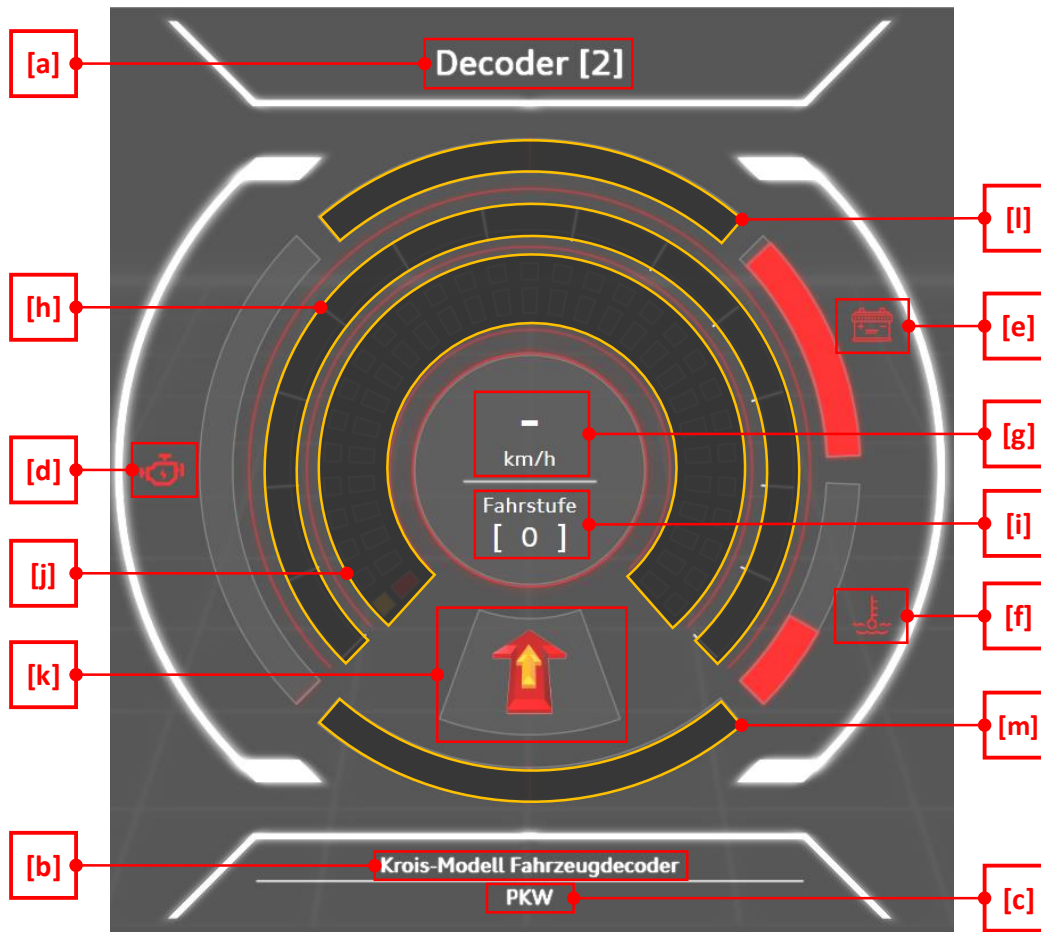
❖ **Function-Keys**

The „Functions“-area contains 24 buttons that are used to control the customizable decoder functions. In DCC-mode, the light bulbs [a] show the current state of the corresponding function [b], in PC-mode this state can also be toggled by clicking on the button. If the user has provided a description for the function (optional), it is shown as a tooltip [c] when hovering over the button(s).



6.3.4 TACHOMETER

❖ Overview



❖ [a-c] Identification-Data

Above and below the actual tachometer controls the name [a], description [b] and the group [c] of the selected car are displayed. These controls mainly serve identification purposes and do not influence the decoder operation in any way.

❖ [d-f] Status-Bars

The bars on the left and right side show different parameters of the decoder. On the left side, the current motor voltage [d] (as set in the motor mapping) is displayed. The bars that show the battery voltage [e] and the decoder temperature [f] are located on the left.

❖ [g-h] Speed

If a valid motor profile is set for the decoder, the center of the tachometer [g] shows the current speed of the car (calculated according to the set scale) in kilometers per hour. Additionally, this speed value is also displayed using the outer arc [h].

❖ [i-j] Speed Step

The two innermost arcs [i] show the current speed step (inner arc) as well as the set target speed step (outer arc). They are split into 29 segments (stop + 28 speeds). In addition to the arcs, the current speed step is also shown in the center of the tachometer [j] for better readability.

The current speed step can differ from the set target speed for multiple reasons: depending on the set defined curves, the acceleration and deceleration can take some time. While the car is in the process of changing its speed, the target speed is already set to the final value whereas the current speed rises or falls in rather small steps. Another reason can be the distance control: when an obstacle is recognized, the car may slow down to avoid possible collisions. In this situation, the car keeps the requested target speed, but due to the distance control the actual speed might be slower.

If the control mode is set to "PC", the target speed can be changed by clicking on the outer arc segments. The first segment (left) represents step 0, the last one (right) corresponds to step 28.

Key	Description
▲ (arrow up)	Increments the target speed step (+1)
▼ (arrow down)	Decrements the target speed step (-1)
← (backspace)	Sets the target speed to zero (stop)

❖ [k] Direction

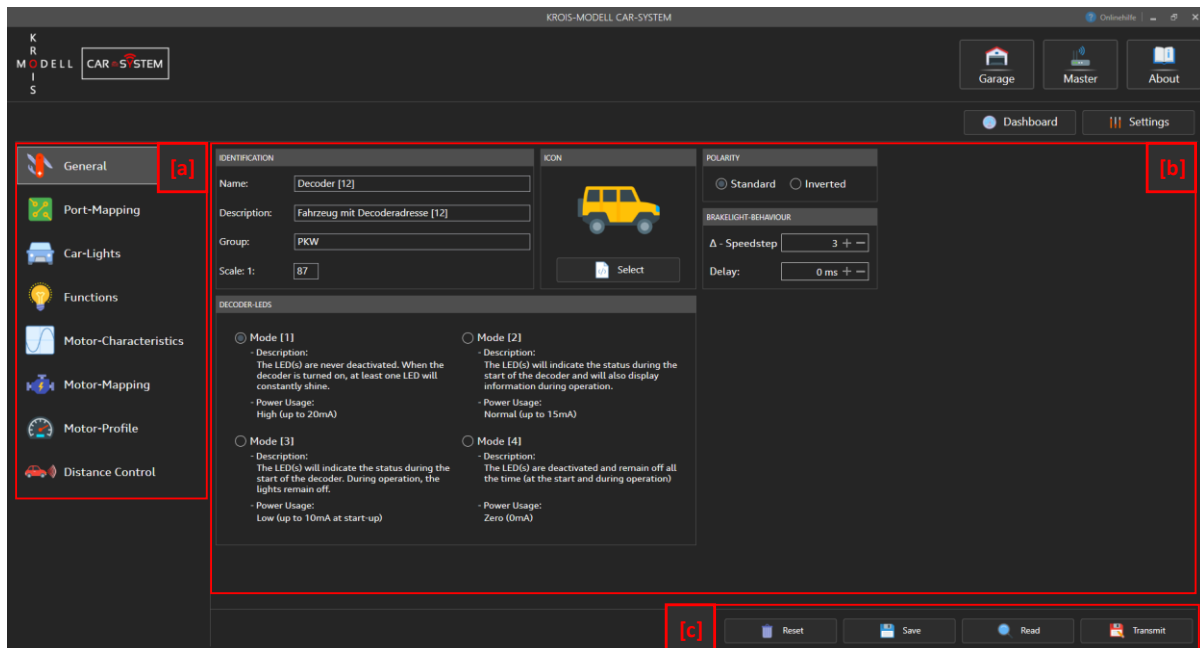
As with the speed step, there are also two direction settings: the current direction (red arrow) and the target direction (yellow arrow). If, for example, a car is driving forwards with speed 14 and the user suddenly decides that the car should drive backwards with speed 4, the decoder must not change the motor direction instantly. Instead, it must first start decelerating until speed 0 is reached before actually reversing the motor polarity. During this period, the two arrows (red and yellow) will point in different directions.

If the control mode is set to "PC", the target direction can be changed by clicking on the smaller (yellow) arrow arc segments. If the arrow points "inside" the screen, the set direction is forwards, otherwise (when the arrow points towards the user) backwards.

Key	Description
⌘ (space)	Toggles the target direction

6.4 DECODER-SETTINGS

6.4.1 OVERVIEW



❖ [a] Groups





The list on the left side of the window shows the available settings. By selecting an entry from the list, the settings are loaded and displayed.

❖ [b] Configuration

This part of the window changes dynamically depending on the selection. It shows the different settings contained in the group and provides means to change them accordingly.

❖ [c] Management

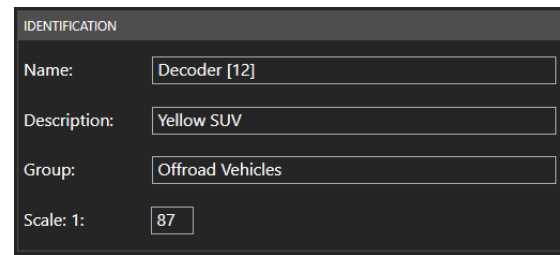
The buttons on the lower edge are used for storage management (on the decoder as well as on the PC). The actions described below always apply only to the selected group.

Command	Meaning
 Reset	The locally stored settings are reset to their default values – the settings stored on the decoder remain unchanged.
 Save	The settings are verified, and changes are saved locally – the settings stored on the decoder remain unchanged.
 Read	The settings contained in the selected group are read from the decoder and displayed in the interface – the locally stored settings are not overwritten (be can be using the “save”-command)
 Transmit	The settings are verified, changes are saved locally and transmitted to the decoder (where they are saved as well).

6.4.2 GENERAL

❖ Identification-Data

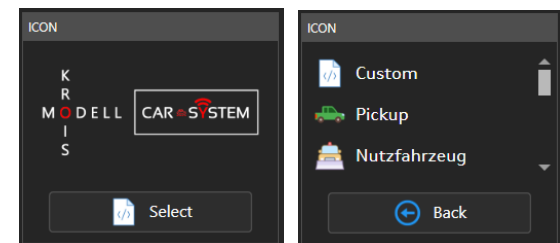
The settings listed here are mainly used for identification purposes in the software – they do not affect the behavior of the decoder in any way. Nevertheless, even these settings are stored on the decoder and can be recovered at any time.



Parameter	Description	Chars (min)	Chars (max)
name	The (unique) ID auf the car/decoder (displayed in the garage and on the dashboard)	1	64
description	Additional details about the car (displayed on the dashboard only)	0	128
group	Classification of the car (needed for sorting) (displayed on the dashboard only)	1	64
scale	The scale of the car (e.g. 1:87) (used for internal calculations)	1	3

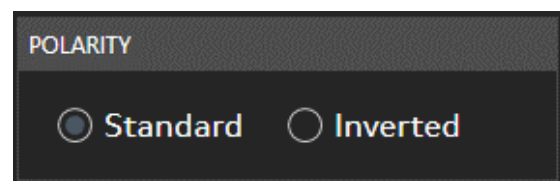
❖ Icon

The icon is only displayed on the car tiles in the garage. It is an easy and fast way to distinguish between the different cars. Besides the standard (preset) icons, a custom icon file can be loaded by selecting the first entry in the list.



❖ Polarity

By switching the polarity setting, the turning direction of the dc motor can be reversed.



❖ Braking lights

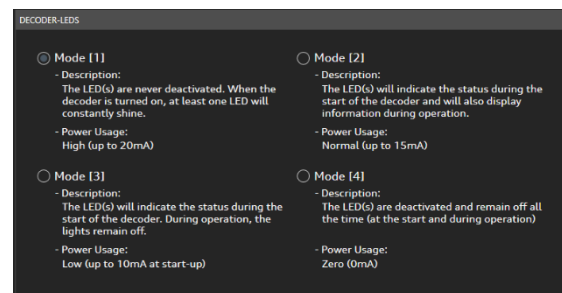
The decoder can switch the braking lights on and off without user intervention. These settings are used to customize and tune the behavior of the braking lights.



Parameter	Description	Value (min)	Value (max)
Δ - Speed	The speed step-difference that is required for the braking lights to become active. As it is not realistic to turn on the braking lights when decrementing to set speed step by one, the required threshold can be configured using this parameter. Example: Δ - Speed = 3 When decrementing the speed from 14 to 12 ($\Delta = 2$) the braking lights remain off; when decrementing the speed from 8 to 4 ($\Delta = 4$) the braking lights are activated.	1	14
Delay	When the deceleration is completed, the braking lights are turned off automatically. Using this parameter, a delay between the completion of the braking process and the deactivation of the braking lights can be defined.	0	5000

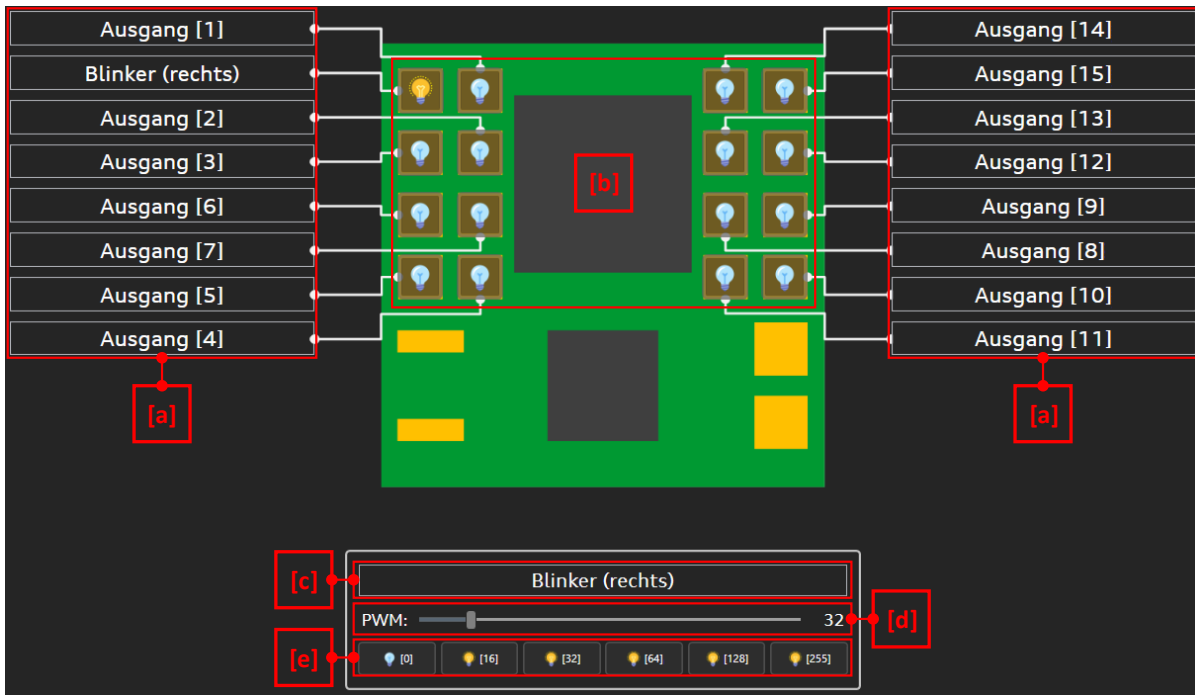
❖ Status-LEDs

Depending on the car and the installation of the decoder, it might be necessary to disable the LEDs on the decoder to prevent unwanted lighting. In addition, disabling the LEDs reduces the used current and therefore leads to longer battery life. There are four different modes that define the behavior of the status-LEDs:



Mode	Description	Current Consumption
[1]	The LED(s) are never deactivated. When the decoder is turned on, at least one LED will constantly shine.	High (up to 20mA)
[2]	The LED(s) will indicate the status during the start of the decoder and will also display information during operation.	Normal (up to 15mA)
[3]	The LED(s) will indicate the status during the start of the decoder. During operation, the lights remain off.	Low (up to 10mA at start-up)
[4]	The LED(s) are deactivated and remain off all the time (at the start and during operation)	Zero (0mA)

6.4.3 PORT-MAPPING



This group is used to test and name the 16 LED-outputs of the decoder. Although the port names are stored on the decoder and can be read out at any time, they do not affect the behavior of the decoder in any way – the user-defined names are meant to simplify the configuration of the different light functions later.

Every output in the graphic is connected to a textbox [a]. By clicking inside the textbox, the name of a port can be changed. If an output is selected, its name is also displayed in the large textbox [c] at the bottom (where it can be changed in the same manner). To select an output, simply click one of the lightbulbs (inside [b]). As only one output can be active at a time, all other outputs are turned off as soon as a new port is selected. To change the brightness (PWM setting) of the selected port, simply drag the slider [d] using the mouse pointer or click on of the quick-select buttons [e] below. A selected output can be de-selected by clicking on the corresponding light bulb again. If no output is selected, the control [c], [d] and [e] are disabled but the names can still be changed in the textboxes [a].

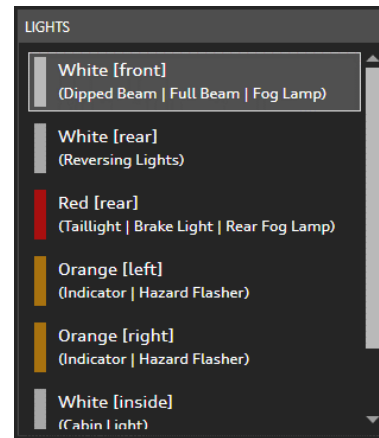
Please note that all functions and car-lights are automatically disabled as soon as this settings group is loaded. They are, however, enabled and restored as soon as the settings are left again.

Parameter	Description	Chars (min)	Chars (max)
name	The (unique) name or description of an output (used in various settings)	1	64

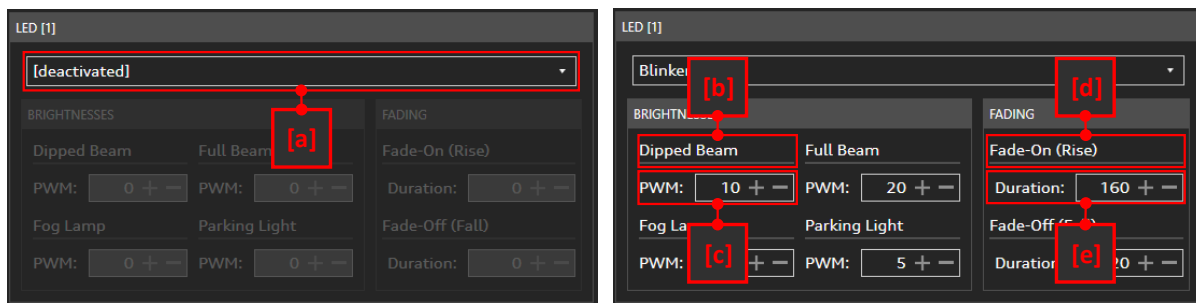
6.4.4 CAR-LIGHTS

❖ Lights (Selection) (LEDs / Colors)

The lights are grouped into LED-colors (i.e. their position in the car). Every color (entry) can be linked to up to four LED-outputs and every output can have up to four different brightness settings (depending on the number of functions associated with the color/LED). For example: the white led(s) mounted at the front can be used as the parking lights (low), headlights (normal), fog lights (high) as well as the high beam (very high).



❖ Lights (Configuration)

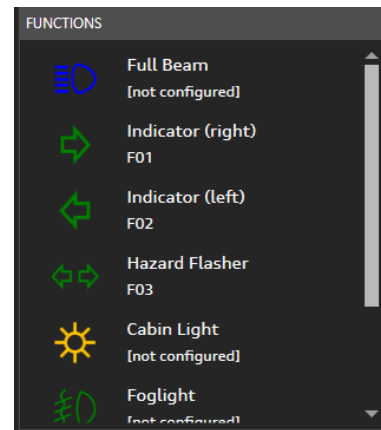


To link an output to the selected LED (color), simply select the port name from the dropdown menu [a]. After an output has been selected, the other controls are enabled. The PWM-settings are listed in the left box, each function name [b] has a textbox [c] linked to it that can be used to set the desired brightness. The same principle applies to the box on the right side where the fading durations can be specified. Every fading direction [d] has a textbox [e] linked to it again.

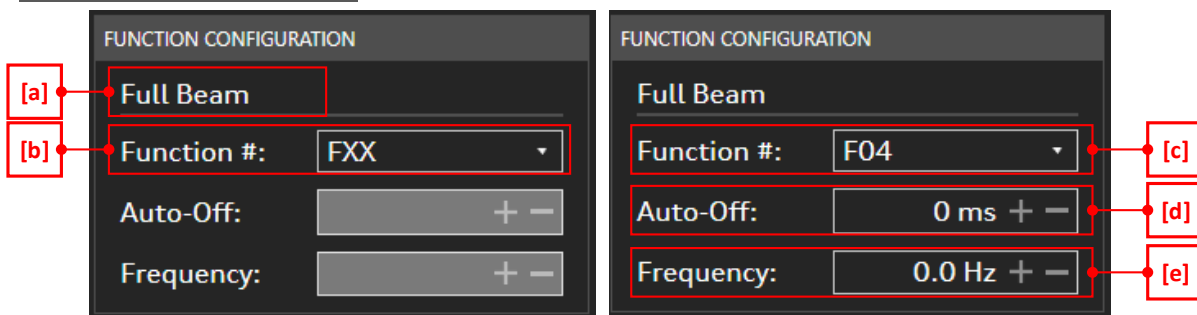
Parameter	Description
port	The port that is to be used for the light. Please note that every output can only be used once in the car light configuration.
brightness (PWM)	The brightness that is set when the corresponding function is enabled (0 = off, 255 = maximum)
fading	The duration (in milliseconds) that it takes until the LED reaches the desired brightness (if the value is greater than zero, the PWM value is slowly ramped rather than set instantly)

❖ **Functions (Selection)**

All lights that have been enabled in the previous configurations step have to be linked to a function key to be usable (except for the braking and the reversing lights, of course). To do this, first select the required function from the list. If a function has already been linked to a key, the key number is displayed below the title (e.g. "F12").



❖ **Functions (Configuration)**

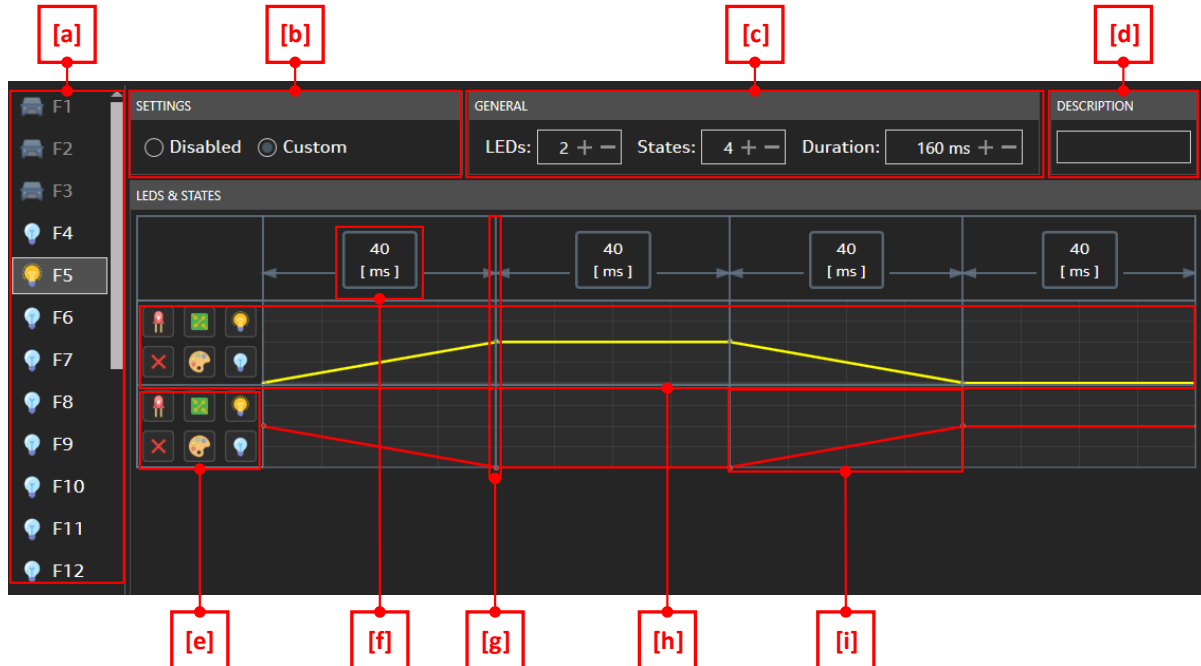


When a light function [a] is selected, the configuration parameters are loaded into the settings window. If the function has not been linked to a key yet, the first dropdown menu displays "FXX" [b]. As soon as a valid function key [c] is selected from the menu, the other parameters ([d] and [e]) are enabled and can be changed. To un-link the function, simply select "FXX" from the dropdown menu again.

Parameter	Description
function #	The function key that is linked to the selected light function. If the entry "FXX" is selected, the light function is currently inactive (not configured). Only keys that have no other function (custom or car lights) linked to them can be selected (functions that are already in use are greyed out).
auto-off	Defines a time interval after which the function is deactivated automatically. If the set value is equal to zero, the function remains on until the key is deactivated again.
frequency	Sets the blinking frequency of the light function (primarily intended for indicator lights and hazard flasher).




6.4.5 FUNCTIONS

❖ Overview



The function editor allows the custom configuration of light animations and complex light operations incorporating the 16 LED-outputs of the decoder. For every function key, 8 LEDs can be used that can each have up to 12 individually configurable states.

❖ [a] Function-Key

Symbol	Description
	<u>Disabled</u> The function key is currently not in use and can either be configured as a light function or a custom animation.
	<u>Enabled (Mode: Custom)</u> The function key is currently configured as a custom animation and can therefore not be used for a light function.
	<u>Enabled (Mode: Car light)</u> The function key is currently linked to a light function and can therefore not be used for a custom animation.

❖ [b] Function-Mode

Mode	Description
Disabled	The function key is currently disabled (not in use)
Custom	The function key can be configured using the editor.







❖ [c] Function-Configuration (General)

Parameter	Description	Value (min)	Value (max)
LEDs	The number of outputs that are used by the function (corresponds to the number of lines in the editor).	1	8
states	The number of states that are used by the function (corresponds to the number of columns in the editor)	1	12
duration	The total duration of the animation (the sum of all state durations)	40ms per LED	16000 ms

❖ [d] Function-Description

Parameter	Description	Chars (min)	Chars (max)
description	The (optional) description for the function. (display as a tooltip in the dashboard)	0	128

❖ [e] LED-Configuration

Symbol	Description
	<u>Presets</u> Opens a list of preset function configurations. Depending on the preset, the number of states and the total duration may be changed automatically – this can therefore also affect other LEDs of the function.
	<u>Delete/Remove</u> Deletes the corresponding LED from the configuration if the total number of LEDs is greater than 1. If there is only one LED in the configuration, this command has no effect.
	<u>Port/Output</u> Opens the port list to allow the selection of an output that should be linked to the LED in the editor. Every port can only be used once per function (key).
	<u>Color</u> For easier identification and an improved workflow, the LED/line color can be changed. This setting has no effect on the function itself and is not stored on the decoder.
	<u>PWM [on]</u> Sets the LED-brightness for the “on”-state.
	<u>PWM [off]</u> Sets the LED-brightness for the “off”-state. If this value is greater than zero, the LED is never turned off while the function is enabled. This setting can be used to toggle the LED brightness between two different levels.

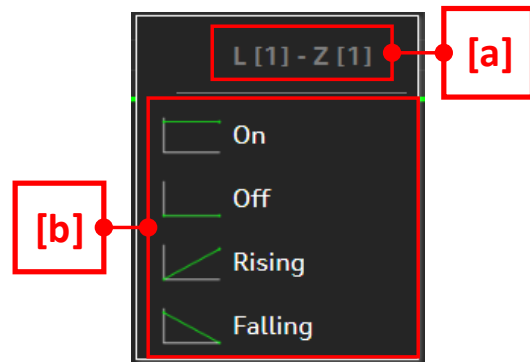
❖ **[f-g] State-Duration**





The duration of the individual states can be set by the user (within certain limitations, of course). Every state must be at least 40ms long and the total duration cannot exceed 16 seconds. The set duration is displayed in the top row [f] (if there is enough space) or as a tooltip when hovering the grey area. To change the duration of a state, simply drag the state delimiters [g] while holding down the left mouse button. All durations are automatically rounded to a multiple of 10ms.

❖ **[h] State-Configuration**

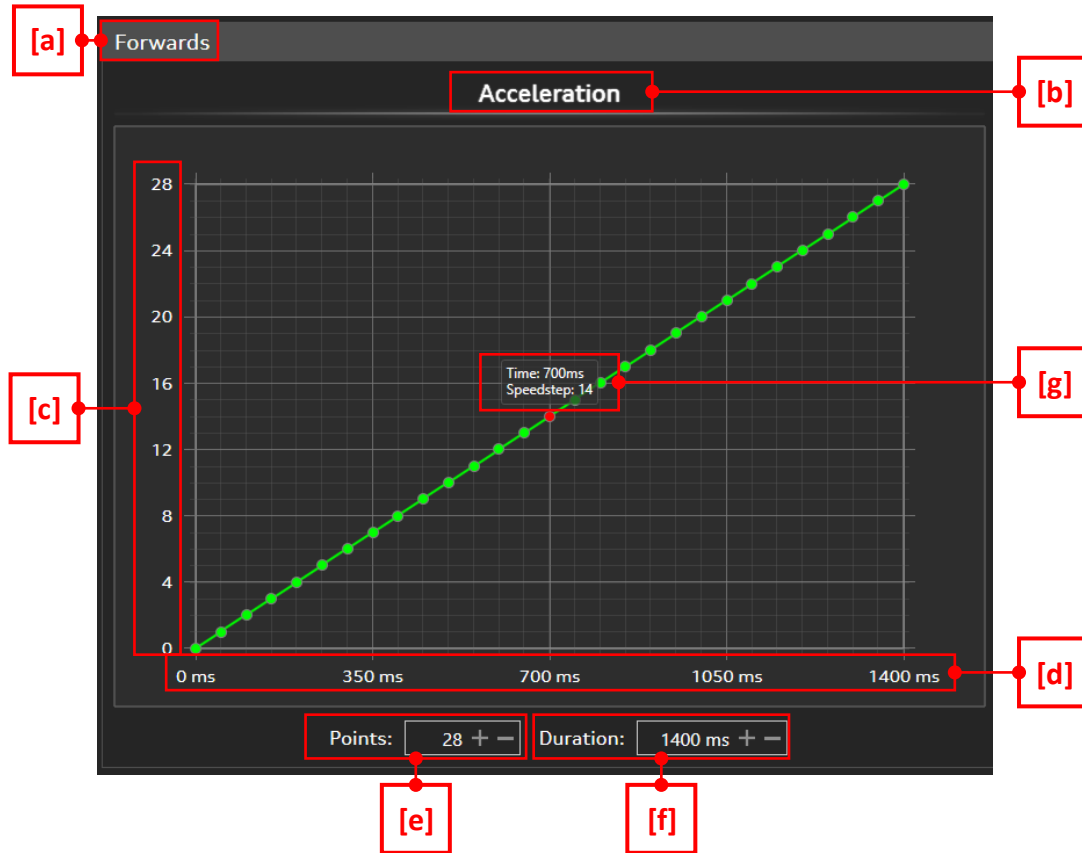
Every row in the editor represents an LED and its states. Before the configuration of the individual states is started, the general parameters [c] of the functions should be set (as the configuration might be reset when the general parameters are changed).

For each LED state there are four different actions that can be set. The selection menu can be opened by right-clicking on the desired state. When opened, the title [a] shows the index of the LED (“L”) as well as the state number (“Z”). All available actions for the selected state are listed below the title [b].



Symbol	Description
	<u>On</u> The LED is constantly on (brightness = PWM [on]). If the previous state is “off” or “falling”, the brightness instantly jumps from the PWM [off]- to the PWM [on]-value.
	<u>Off</u> The LED is constantly off (brightness = PWM [off]). If the previous state is “on” or “rising”, the brightness instantly jumps from the PWM [on]- to the PWM [off]-value.
	<u>Rising</u> The brightness rises gradually from the PWM [off]- to the PWM [on]- value. If the previous state is “on” or “rising”, the brightness instantly jumps from the PWM [on]- to the PWM [off]- value and then starts to rise again.
	<u>Falling</u> The brightness falls gradually from the PWM [on]- to the PWM [off]- value. If the previous state is “off” or “falling”, the brightness instantly jumps from the PWM [off]- to the PWM [on]- value and then starts to fall again.

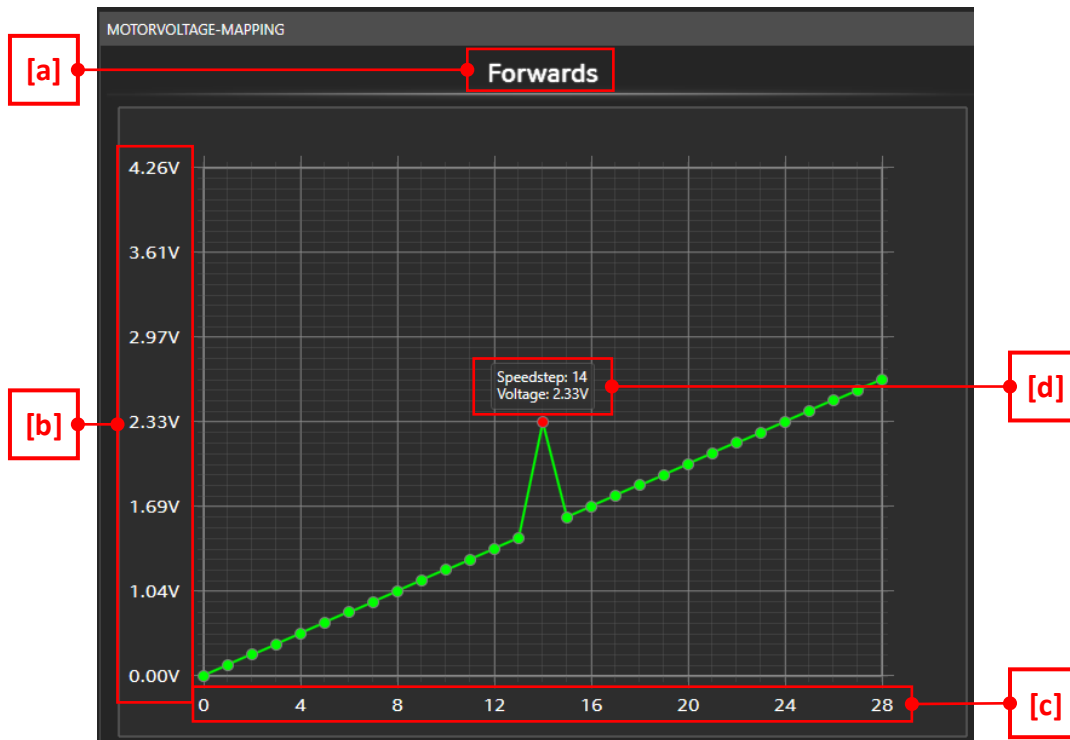
6.4.6 MOTOR-CHARACTERISTICS



Using the four interactive plots in this settings group, the acceleration and deceleration behavior of the decoder can be set and customized. Every plot has a title [b] that shows whether it controls acceleration or deceleration. As there are two driving directions [a], four plots are needed in total (the picture above shows the acceleration curve in the forwards direction). The y-axis [c] of the plots lists the 28 speed steps while the x-axis [d] shows the total duration. To adjust the time base, a textbox [f] is provided below each plot that sets the maximum value for the x-axis (and therefore also controls the timespan between the individual lines/steps). To other textbox [e] is used to set that number of steps for the acceleration or deceleration curve. Every point in the plots can be moved using the mouse pointer, enabling easy and fast changes to the motor characteristics. When a point is selected, the tooltip [e] shows its current position and settings.

Parameter	Description	Value (min)	Value (max)
points	The total number of points that are run through when accelerating from speed zero to the maximum speed (or when decelerating from the maximum speed down to zero, respectively).	8	44
duration	The total duration of the acceleration or deceleration process (can be used to vary the time between two steps/points).	200 ms	19200 ms

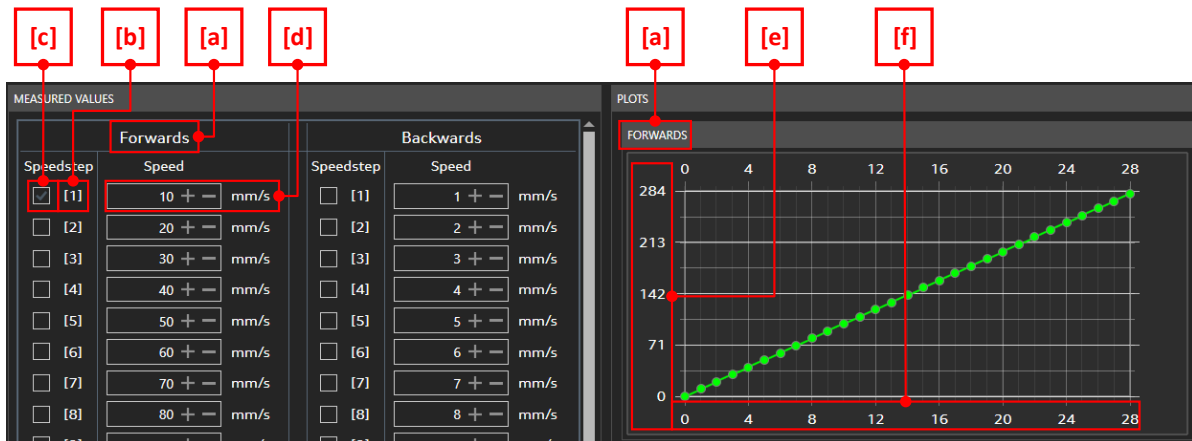
6.4.7 MOTOR-MAPPING



Although the interfaces might look quite similar, these settings have nothing to do with the motor characteristics described in the previous chapter. The mapping controls the voltage that is applied to the dc motor at a given speed step. Especially when using a gearbox or gears that need a little more torque before they begin to turn, it might be necessary to run the motor with a higher voltage. Of course, it is also possible to limit the maximum speed of the car as well, although these settings should rather be made using the motor characteristics. The mapping can be set independently for both directions [a] using the two diagrams. This time, the y-axis [b] shows all available voltages while the x-axis [c] represents the 28 speed steps. To change the set voltage for a given speed, simply drag the corresponding point in plot along the vertical (y-) axis. When a point is selected, a tooltip [d] opens and shows the current voltage setting as well as the selected speed step.

Parameter	Description	Value (min)	Value (max)
voltage	The (PWM-controlled) voltage that is applied to the motor at a given speed step. As the voltage is constantly monitored, the motor driver automatically reacts to load changes and adjusts accordingly. If the battery voltage falls below the set motor voltage, the driver can of course no longer apply the requested voltage and instead simply uses the maximum available voltage (i.e. the battery voltage).	0.00V	4.26V

6.4.8 MOTOR-PROFILE



The motor profile is used to calculate the real-world speed based on the provided car scale. In addition, the profile is required by the distance control to allow the calculation of the speed difference between the cars. When at least one value is provided by the user, the software automatically calculates the remaining, non-user defined values. Of course, the profile can be set for both directions independently as indicated by the column headers [a]. To enter a value, checkbox [c] for the corresponding speed step [b] needs to be checked first. After the speed step has been enabled, the measured speed value can be entered in the textbox [d]. The unit of the measured speed is always “millimeters per second”. Every change in the column on the left side automatically leads to an update of the plots on the right side. The y-axis [e] of the plots shows the speed (in [mm/s]) and is updated dynamically while the x-axis [f] simply shows the 28 speed steps. As the plots are read-only, they are only used to visualize the values from the table - dragging of the points is not supported.

Parameter	Description	Value (min)	Value (max)
user-defined	Specifies if the value is measured (and defined) by the user or should be calculated by the software. (set by the checkbox besides the speed step label)	-	-
speed	A measured value that specifies the real-world speed of a car at a given speed step. The unit of measurement is “millimeters per second”. To conduct a measurement, let the car drive along a ruler or another object with a defined (and known) length while stopping the time it takes the car to cover the distance. Divide the distance (in millimeters) by the total driving time (in seconds) to get the required value for the motor profile. This measurement/calculation should be performed for at least three different speed steps to get an accurate representation of the motor profile.	1 mm/s	999 mm/s