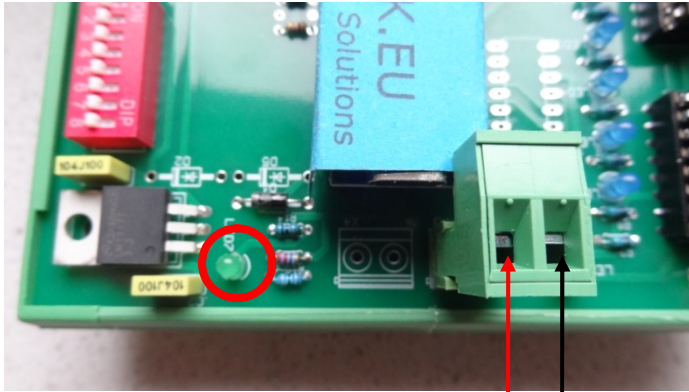


Set up and connect the Switch ID Reader

Standard Switch ID Reader:

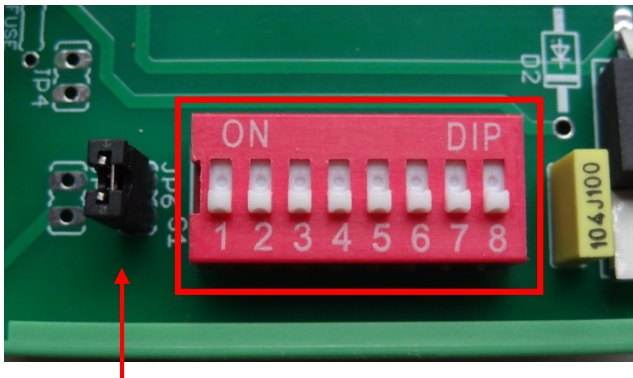
6x Outputs (ID 1...6) suitable for switching up to 50mA.
Power is supplied via the track connection.

The track voltage is connected here.



the **green led** light up
The connection is protected against incorrect connection

Left the Plus, right the mines (mass)



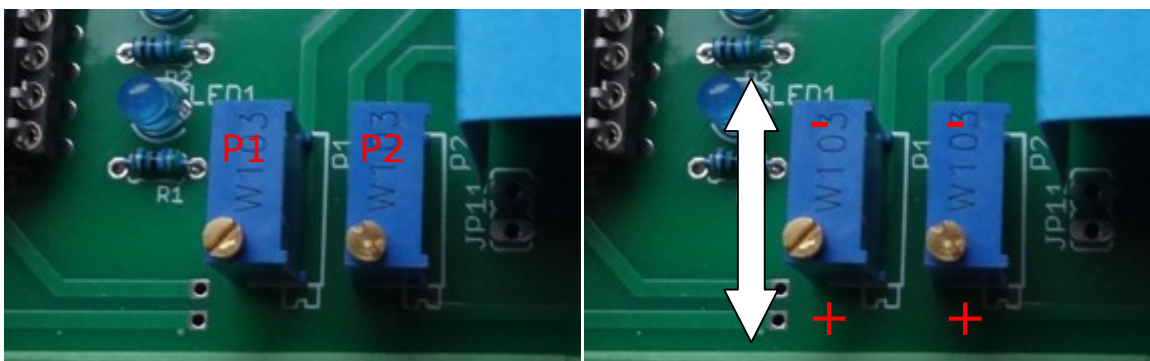
Dipswitch:

- S1 = External reset, standard off
- S2 = External release on/ off
- S3 = Auto reset on/ off
- S4 = Chaos*
- S5 = Auto Start
- S6 = Time setting 10 S
- S7 = Time setting 20 S
- S8 = Time setting 40 S

Jumper is standard

Speed setting for 2 presses:

- P1** = Minimum time setting for 2 consecutive presses
- P2** = Maximum time setting for 2 consecutive presses



S1 = External Reset On/Off

Off => External reset possible

On => Reset is always active, do not use

S2 = External Enable On/Off

Off => External control possible to block outputs (ID 1...6)

On => No blocking

S3 = Auto Reset or On/Off

Off => Output remains on after Chaos is activated. Chaos can be cancelled by pressing twice again or by performing a reset (see external reset).

On => Output is switched off again after a fixed time after Chaos activation

S4 = Chaos Activation

Off => Chaos activation only works when the controller is pressed

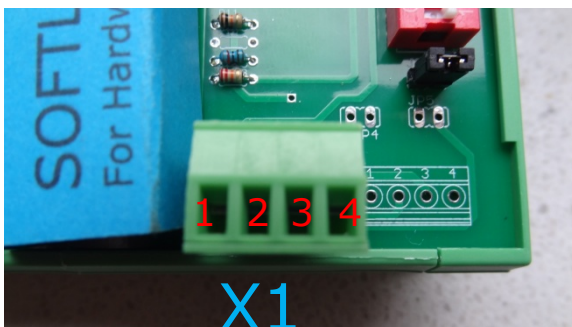
On => Chaos activation only works when the controller is not pressed

S5 = Auto Start*

Off => No automatic cancellation of Chaos

On => Automatic cancellation of Chaos with start light

*** If an external relay card is connected, Chaos can also be cancelled without the start light.**



from left to right for both

1 = N.C

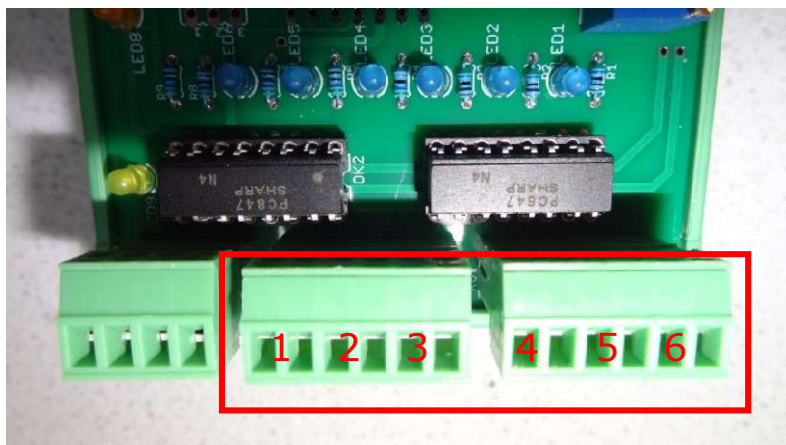
2 = N.C

3 = Relay via output 7 or 8*

4 = - (mass)

* has no function without relay extension

Connecting the 6 outputs:

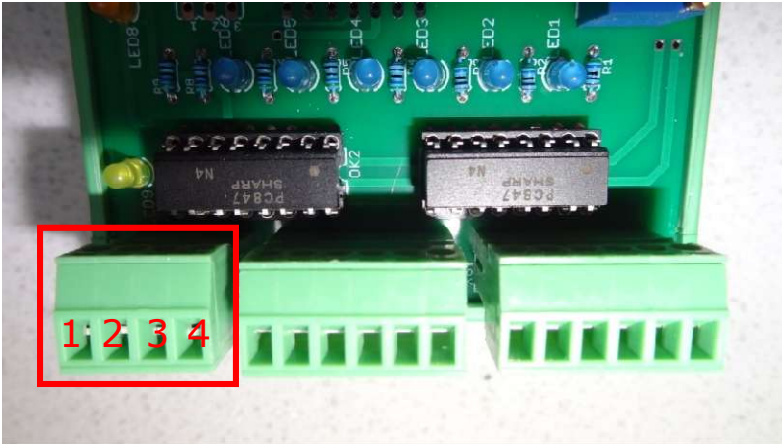


From left to right ID1... ID6

Each ID has two connections, which can be connected to, for example, the inputs of a USB box or Arduino.

Or, to switch an LED to indicate who has activated Chaos.

Connecting the two inputs (functions S1 and S2):



From left to right:

- 1 = + 5V
- 2 = Enable outputs 1...6
- 3 = Reset Chaos output
- 4 = - (ground)

To activate the input, connect terminals 2 and/or 3 to terminal 4 (ground).

S6...S8 see table

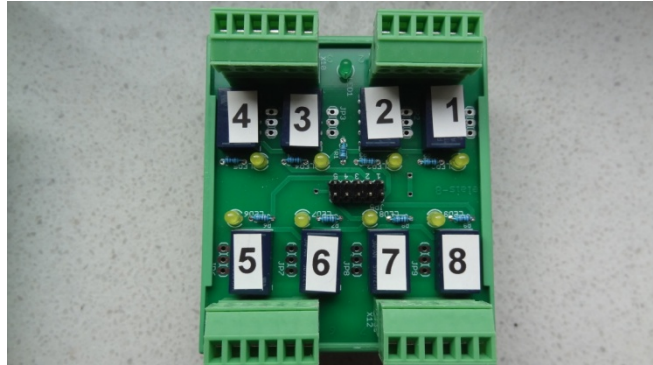
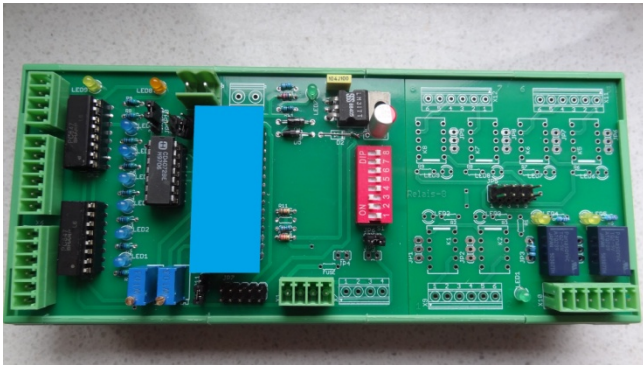
| S6 | S7 | S8 | Time (S) |
|-----------|-----------|-----------|----------|
| Off | Off | Off | 10 |
| On | Off | Off | 20 |
| Off | On | Off | 30 |
| On | On | Off | 40 |
| Off | Off | On | 50 |
| On | Off | On | 60 |
| Off | On | On | 70 |
| On | On | On | 80 |

Jumpers:

- J1 = Power supply external or via USB-C
- J2 = Not present
- J3 = Not present
- J4 = Pull-up resistor for relay selection
- J5 = Not present
- J6 = Not present
- J7 = Debug on/off (removed by default)

Extended Switch ID Reader:

- 6x Outputs (ID 1...6) suitable for switching max. 50mA
- 7x Outputs (ID 1...6) suitable for switching max. 2.0A
- 1x output as a shared output (for max. 2.0A) for ID 1...6

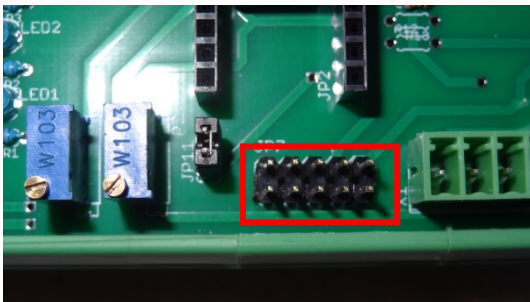


Expansion only with relays 7 & 8.

Expansion with 8 relays.

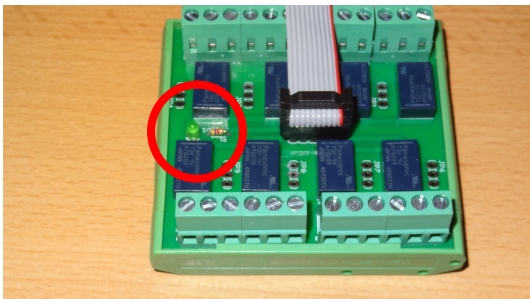
Extended ID Reader with the option to connect a relay module.

Connecting the relay module:



Connect it to the PCB header using a 10-pin flat cable (included with the relay module).

Pin number 1 is shown in the top left corner.



If the cable is connected correctly, **the green LED** on the relay module will light up.

Chaos control with the CU 30352:

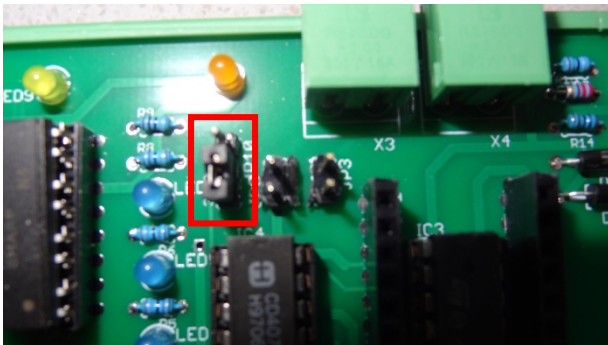
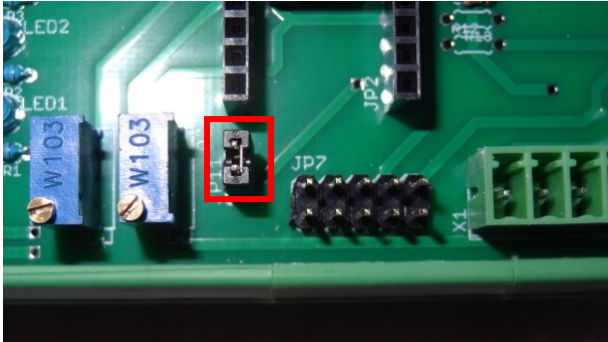
Relay 7 can be used to control the "Start" button on the CU 30352, and relay 8 can be used to control the "ESC" button on the CU 30352.

Relays 1...6 follow the outputs ID1...6.

Additional jumper settings:

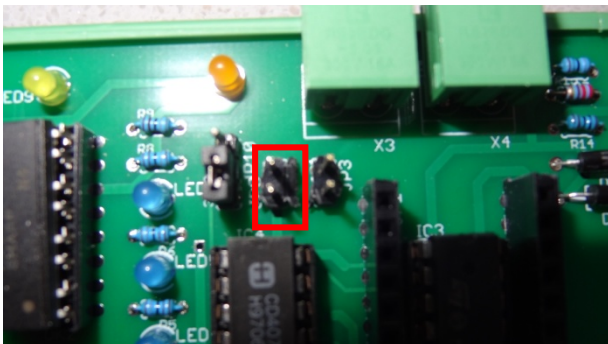
Next to the pin header for the relay module is a jumper; this must be present when the track voltage is connected.

USB-C must not be connected in this situation!



This jumper selects whether output 8 also switches relay 7. Relay 7 becomes active when outputs 1...6 are activated.

In the photo, this jumper is in the wrong position!



This jumper must not be inserted!
(It has no function in this application).



This jumper allows you to control the outputs (**blue LEDs**) of the Switch ID reader. As soon as the ID reader is powered (jumper inserted), outputs 1...6 light up one by one (5 seconds on, 0.5 seconds off).

Remove jumper for normal operation!

Adjust settings P1 and P2:

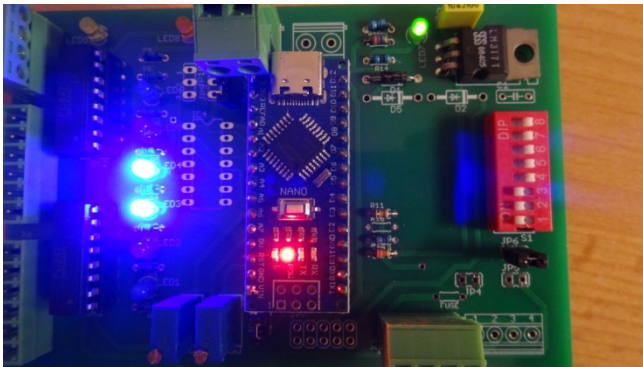
| Dipswitch | Reset (externally) | Release output | Auto Reset | Chaos | Auto Start |
|-----------|-----------------------|----------------|-----------------------|-----------|------------|
| | S1 | S2 | S3 | S4 | S5 |
| S1 | ✓ P1 P2 | ✓ P2 | ✗ P1 P2 | ✓ | ✗ |
| S2 | ✗ | ✓ | ✓ | ✓ | ✓ |
| S3 | ✗ | ✓ | ✓ | ✓ | ✓ |
| S4 | ✗ | ✓ | ✓ | ✓ | ✓ |
| S5 | ✗ | ✓ | ✓ | ✓ | ✓ |

Note: S1 and S3 should never be set to "On" simultaneously.

To simplify setting the click speed (double-clicking the slider), the Switch ID Reader has a "Test" mode.

Set P1 (minimum setting) to a time window of 100...250 ms.

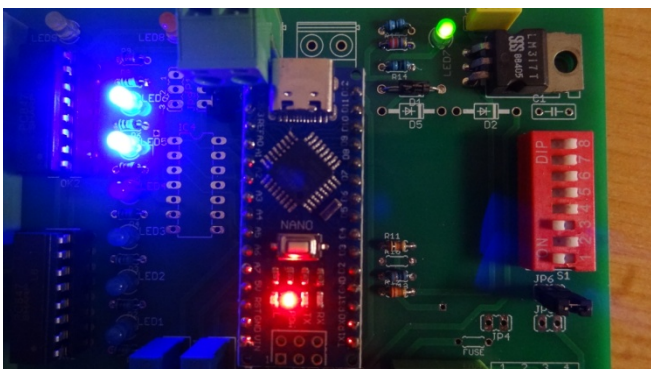
Set DIP switches **S1**, **S2**, and **S3** to **On** (S4...S8 have no effect).



Now turn **P1** to the correct setting; the photo shows the setting at approximately 180 ms. Turning it clockwise (-) allows for a faster double-click. The blinking frequency is 1 second.

Set P2 (maximum setting) to a time window of 300...450 ms.

Set DIP switches **S1** and **S3** to **On** and **S2** to **Off** (S4...S8 have no effect).

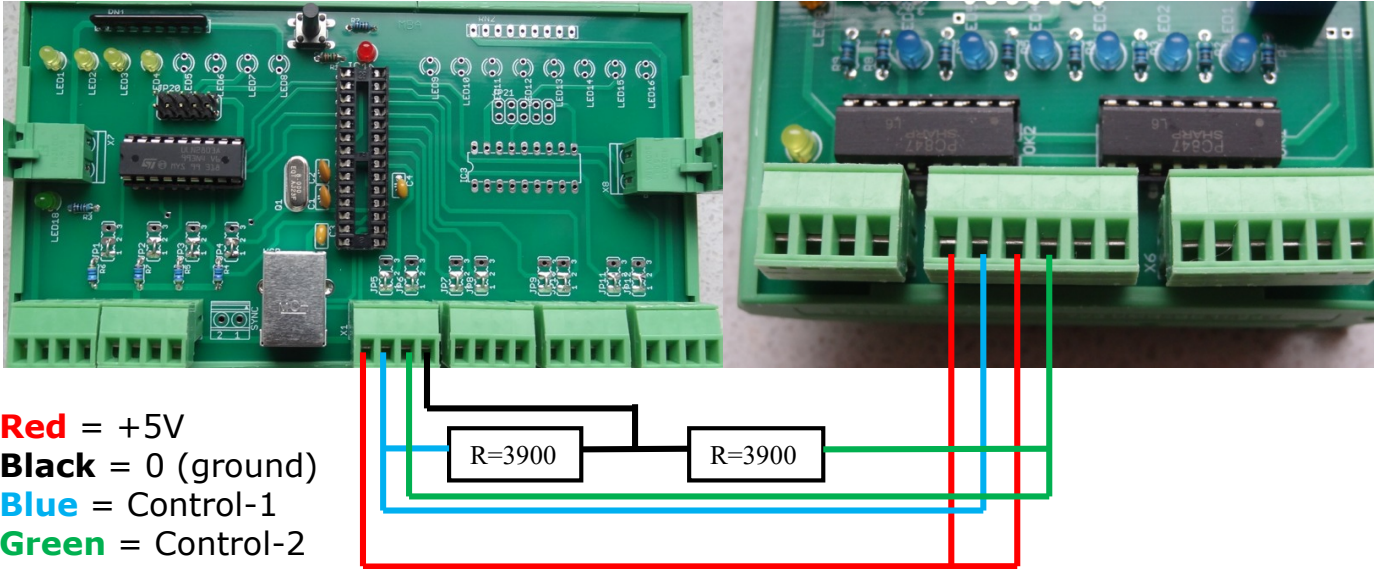


Now turn **P2** to the correct setting. The setting in the photo is approximately 325 ms. Turning it clockwise (-) allows you to click twice more quickly. The blinking frequency is 2 seconds.

After the test, turn off S1 or S3!

Connecting the Switch ID reader to a US Box/Arduino (Cockpit-XP)

The Switch ID reader can be easily connected to an input module that is read by CPX. The 6 outputs of the Switch ID reader are then connected as follows:
(Connection diagram for Inputs 1 and 2 (left) and Outputs 1 and 2 (right))



In this way, all 6 outputs are connected to the inputs. You can freely choose which inputs you want.

Because CPX works with contacts that are closed to 0, i.e., the input of the USB Box/Arduino is connected by temporarily making the input "low."

This gives an incorrect display in CPX:

Contacts that are not connected to the USB Box are displayed as "**Ein**"?



The first 3 inputs are connected to 0; the rest are not connected.

Input 1 is a pushbutton with a "make" contact.

Inputs 2 and 3 are sensors; it doesn't matter whether they are analog or digital. Is the indication correct, because the sensors are not active?

How can a pushbutton with a normally open contact be connected to the 0? See the drawing: the resistor ensures that if the contact is not closed, the input is connected to the 0, which causes the message to read "**Aus.**"

So far, the hardware is connected correctly. However, if you also want to read these inputs to activate Chaos, some further adjustments are required.

If you're going to use the Switch ID reader with CPX, you don't need a relay card, because CPX activates and deactivates the CU 30352 in Chaos, with or without the start light. Note that this only works with CUs with the correct firmware.

In addition to these settings, you also need an Add-On to automatically give Chaos a penalty lap to the correct offender.

To do this, go to [Data Center] and select the racetrack you want to modify.

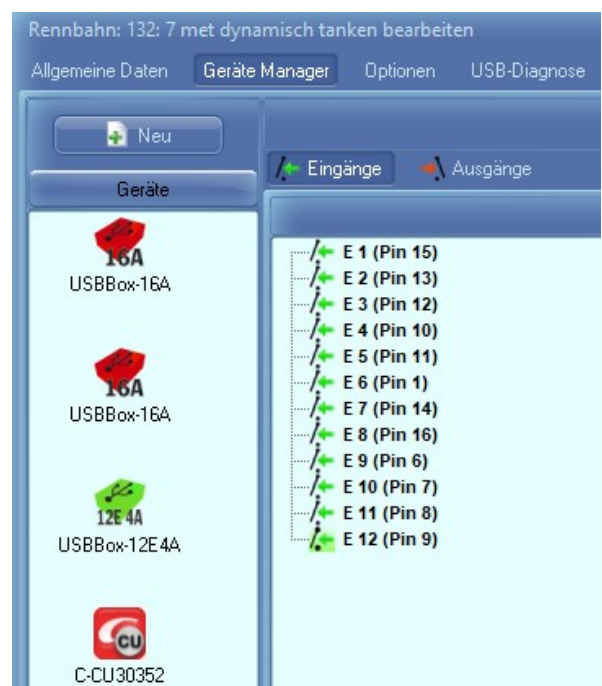


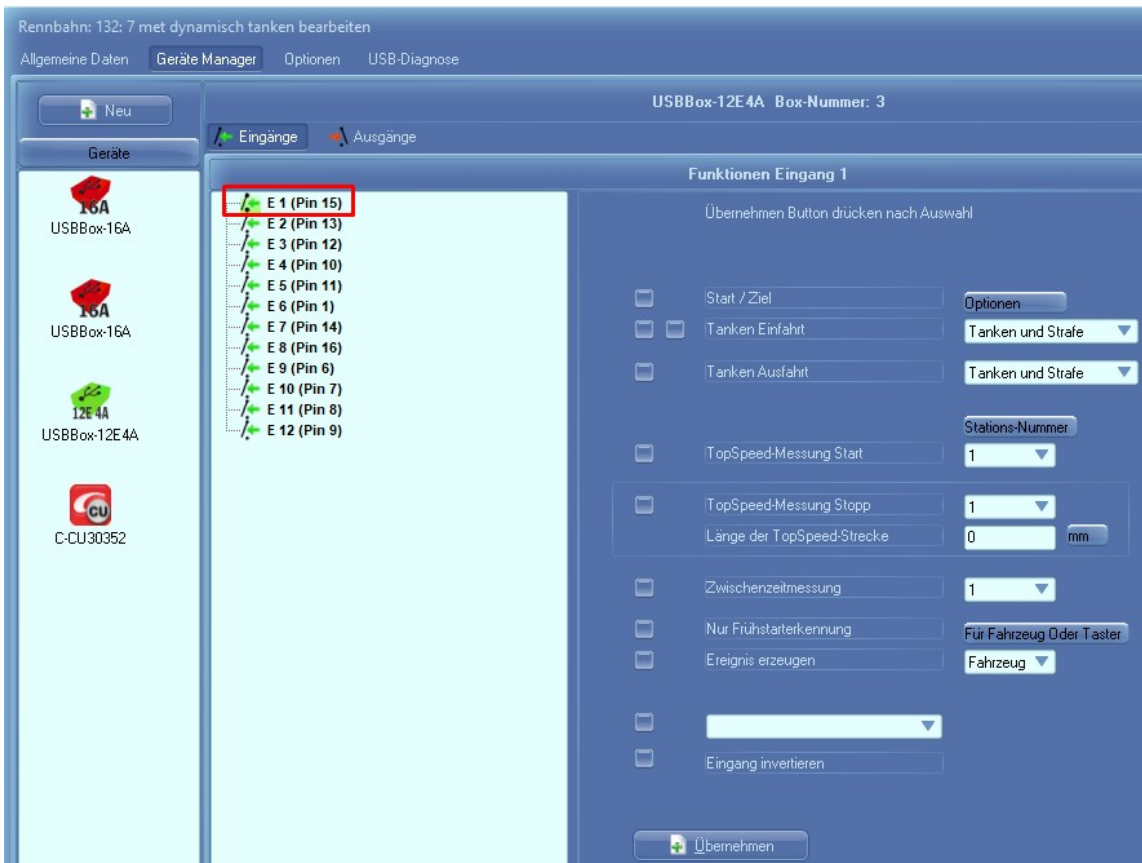
Then select [Device Manager].



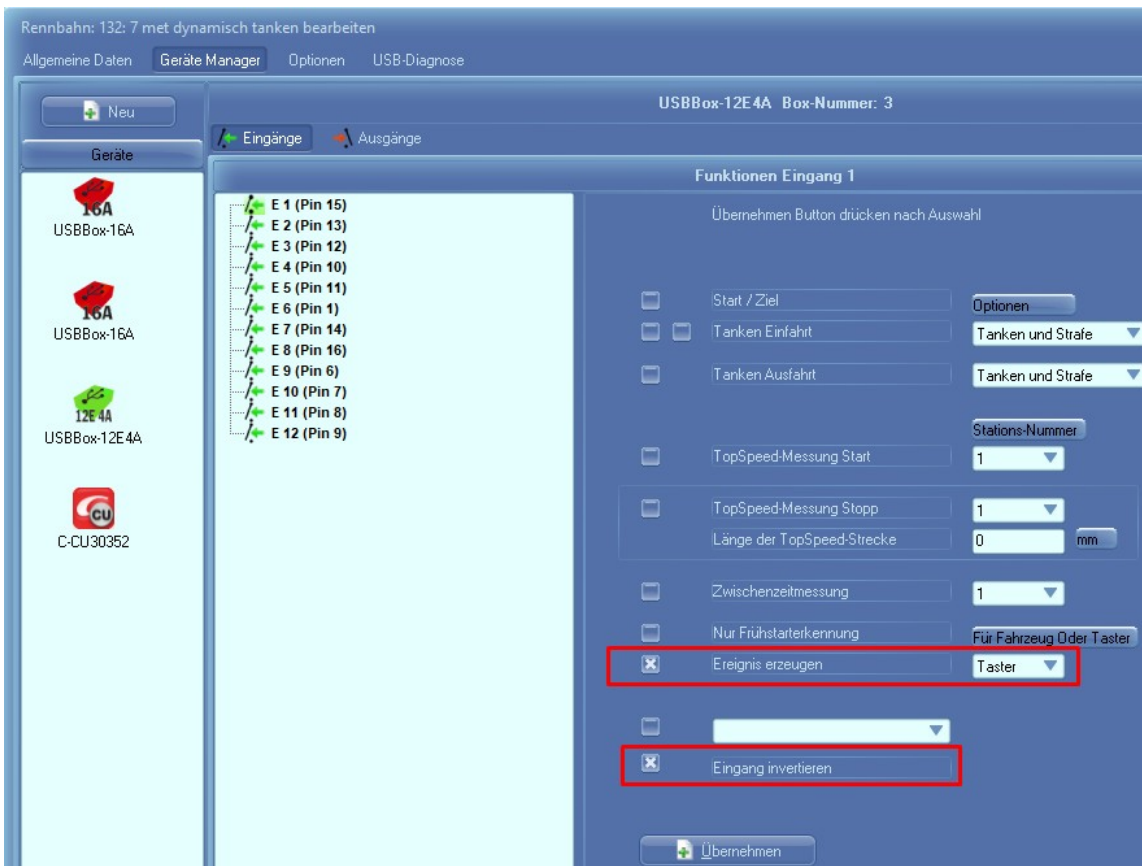
Click on USB Box 12E/4A

Because in our example, input 1 is a pushbutton, it still needs to be modified for this function.





Select input 1 and check the following boxes:



Click "Übernehmen" and do the same for all other inputs connected to the Switch ID reader.

The reason you need to invert the inputs is that the contact is already closed (connected to 0). If you don't do this, the "Ein" and "Aus" indications will no longer be correct!

The sensors in this case work the other way around, so they don't need to be inverted.

0 (Low) = contact closed (Lock sensor/IR sensor)
1 (High) = contact open (Lock sensor/IR sensor)

The Switch ID reader works with outputs that are only closed when the output is activated. The resistor now ensures that when the output is not active, the input of the USB Box/Arduino remains at 0 (low).

Add-On:

I'm offering this for free Available.

You must have a USB Box or Arduino with a license!