

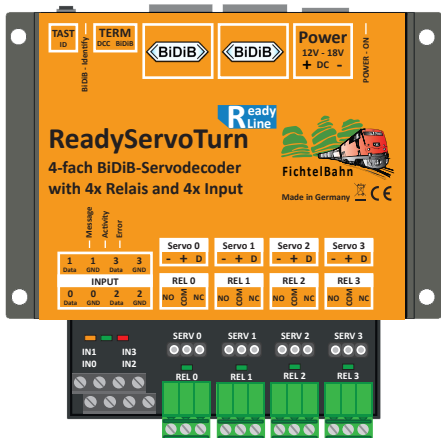


300962



FichtelBahn

Made in Germany



Handbuch / Manual

# ReadyServoTurn



## What is a ReadyServoTurn for?

The ReadyServoTurn is an accessory decoder that is operated by the BiDiBus and provides servo outputs, relay outputs and inputs for the model railway layout.

### Please notice:

The ReadyServoTurn **cannot** be controlled by digital command stations that only send digital turnout commands to the accessory decoders (e.g. DCC format).

**The ReadyServoTurn requires the BiDiBus for operation.**

The ReadyServoTurn allows the position and speed of each servo output to be set. With the help of the relays, the frogs or switch blades can also be polarised.

|  |                                    |
|--|------------------------------------|
| Servo turnouts                           | 4 turnouts                         |
| Servo turnouts with polarisation         | 4 servo turnouts with polarisation |
| Semaphore signals with lighting          | 4 semaphore signals                |
| Barrier crossing with St. Andrew's cross | 2 level crossings                  |
| Double slip switch / three-way turnout   | 2 Multi-way turnouts               |

## Online Documentation

Nowadays, printed manuals can become outdated very quickly.

The most recent version of this manual can be found in the download section of the FichtelBahn webpage. The version number in the footer will show the current version.

New functions and additions are always published in the online version on the webpage first.

Further information on this product can be found also in the BiDiB-Wiki on <https://wiki.fichtelbahn.de> (Until now unfortunately mainly in German)

## Inhaltsverzeichnis

|  |    |
|--|----|
| 01. Safety Instructions .....                    | 3  |
| 02. Introduction.....                            | 4  |
| 03. Technical Data.....                          | 4  |
| 04. Connecting the ReadyServoTurn .....          | 5  |
| 05. Configuration of the module .....            | 11 |
| 06. Module settings for the ReadyServoTurn ..... | 29 |
| 07. LED indication .....                         | 31 |
| 09. Firmware update .....                        | 33 |
| 10. Support case and further help.....           | 34 |
| 11. Warranty Information.....                    | 35 |

## 01. Safety Instructions

To reduce the risk of electric shock and injuries do not touch parts that carry voltage. Do not touch conductive material that might carry voltage in case of a fault, e.g. short circuit, improper input voltage, excessive humidity and accumulation of condensate.

### **To reduce these risks, keep these safety precautions in mind:**

Use this module only indoors and in a clean and dry environment. Avoid moisture and splash water in close proximity.

Switch off the voltage supply before carrying out wiring work. Only use wire with sufficient cross-section. Wait for 2 hours after accumulation of condensate.

## 02. Introduction

This manual explains the basics step by step for using this module. Careful reading and taking note of tips will reduce potential errors and therefore the amount of work to solve failures.

### Designated Use

The normal use of the ReadyServoTurn is for model making especially digital model railways according to this manual. Any improper use will lead to loss of warranty.

### Package Contents

- ReadyServoTurn module with or without housing
- Connection terminal for supplying power
- 4x 3-pole terminal for outputs
- 2x jumper (2,54 mm/1 in grid) for bus termination
- Manual

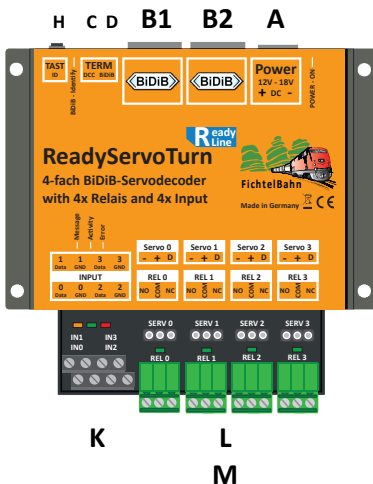
### Required Materials

- Switching power supply with 12V-18V, DC min. 2A current
- RJ45 patch cable for connection to BiDiBus

## 03. Technical Data

|                                       |  |
|---------------------------------------|--|
| Supply voltage                        | 12V - 18V direct current (DC)                                    |
| Power consumption (quiescent current) | 10mA (0,15W)   |
| Servo output                          | 4x servos with 0.5A continuous operation per output (peak = 1A)  |
| Number of relay outputs               | 4x changeover / switching current 2A (4x 3-pole output terminal) |
| Switching times                       | 20ms to continuous operation                                     |
| Inputs                                | 4x inputs with ground reference                                  |
| Protection per output (servo)         | Short-circuit proof with switch-off and error message            |
| Interfaces                            | BiDiBus (RJ45)   |
| Protection class                      | IP 00  |
| Ambient temperature (operation)       | 0 ... +60 °C / 32 ... 140 °F                                     |
| Ambient temperature (storage)         | -10 ... + 80 °C / 14 ... 176 °F                                  |
| Permissible relative humidity         | max. 85 %  |
| Dimensions casing                     | 100mm x 90mm x 34mm / 3.94 in x 3.54 in x 1.34 in                |
| Weight                                | 85 g / 3 oz  |

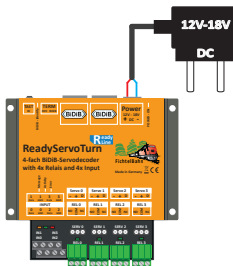
## 04. Connecting the ReadyServoTurn



|    |   |
|----|---|
| A  | Connector for the power supply (DC 12V-18V)                                     |
| B  | <b>BiDiB interface</b><br>connection to command station and further BiDiB nodes |
| B1 | Both sockets are internally connected and can be used equally                   |
| B2 |   |
| C  | Terminating jumper for terminating DCC signal                                   |
| D  | Terminating jumper for terminating BiDiB  |
| H  | Ident- / Bootloader button for system functions                                 |
| K  | 4x inputs against ground (e.g. for position feedback of the servo motors)       |
| L  | 4x servo outputs for servo motors   |
| M  | 4x relay outputs (changeover contact - NO/COM/NC) e.g. for frog polarisation    |

## 04.1 Connecting the power supply

Connect the power supply terminal (A) of the module to a switched 12V-18V DC power supply. Be careful to check the polarity of the module – marked red (+) and blue (-) in the sketch.



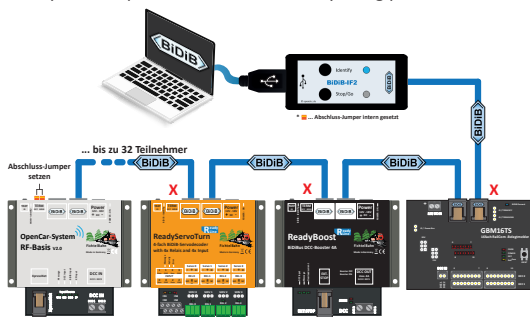
The current consumption of up to max. 2A, depending on the simultaneously active servo motors, should be taken into account when selecting the power supply. Several modules can be connected to one power supply at the same time if the power supply used can provide the necessary output current.

### Please notice:

DO NOT connect a transformer (alternating voltage/AC) or rectified alternating voltage. Doing so will lead to unreparable damage to the module!

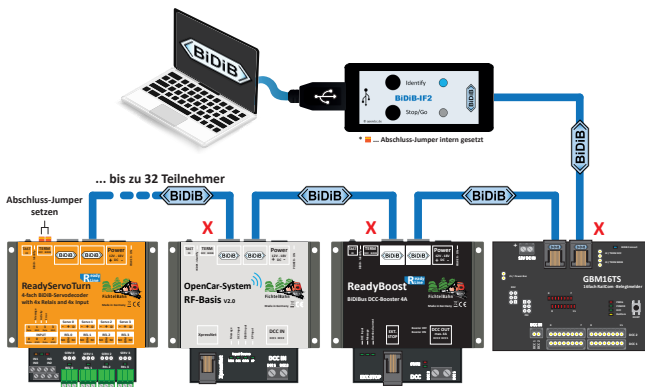
## 04.2 Connecting to the BiDiBus

The ReadyServoTurn has two parallel BiDiBus sockets (B1/B2) which can be used to place the module at any desired position within the bus by using patch cables. .



In the previous figure, the module is placed within the BiDiBus. Therefore, no termination jumper ( X termination) needs to be fitted on the ReadyServoTurn. (For further information on the subject of termination of the BiDiBus, please refer to chapter “08. Background knowledge” on page 32.) A BiDiB-IF2 is shown as symbolic interface for any type (e.g. GBM Master / GBMboost Master)..

In the following figure, the ReadyServoTurn was placed as the last module of the BiDiBus. In this case, the two terminating jumpers for the BiDiB and DCC termination must be fitted on the module.



### Please notice:

If the first and last module of the bus are not terminated with the termination jumpers the distortion of the signal might lead to errors in the data transmission. If the termination jumper is fitted on a module within the bus the transmission might be disrupted. **Both cases will not lead to any damage of the modules.**

## 04.3 Connecting a servo

Four servo motors can be connected to the ReadyServoTurn for isolated applications, but also in combination with the relay outputs and the inputs of the module.

### Switch off servos after movement:

The servos can be de-energised after the switching process. This optional setting avoids a possible servo hum. This setting can be activated individually for each servo in the “Servo ports” section.

### Motion profile

The module has 4 different motion profiles to choose from, which can be selected for each movement direction and for each servo in the settings of the “Servo ports”.

There is a “soft movement” for turnout drives and “rocking or kickback” for signals or level crossings.

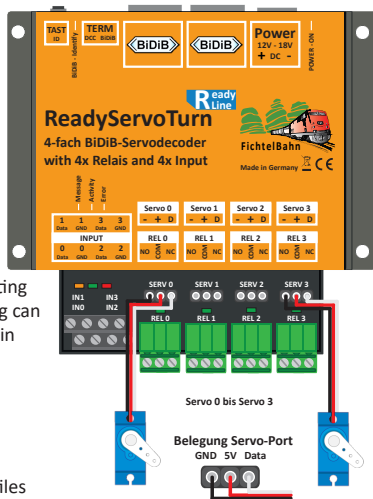
### Wiring length:

The servo motors are controlled with a PWM signal that can interfere with other consumers if the cables are long. Long cable lengths also lead to signal distortions that can be noticed by a twitching servo.

### Please notice:

**It is recommended to keep the cable length between the module and the servo short for safe operation. Cable lengths exceeding 2 metres should always be avoided.**

In the FichtelBahn-Shop you will find a 50cm servo cable extension with the article no. 000830.





### Which servo is suitable - digital or analogue?

One major difference is that a digital servo can move to its position faster and hold the position more accurately. An analogue servo uses an ohmic potentiometer that loses its repetition accuracy with temperature changes (tolerance).

### Both types of servo can be connected to the ReadyServoTurn.

In the case of an effect, the advantage of the digital servo might make a difference. For a turnout or other movements with a mechanical positioning spring and strain relief, the tolerance is unimportant. Here a large travel distance should be chosen and servos with metal gears should be used. In the FichtelBahn-Shop you will find a reliable and inexpensive solution with the article no. 410100

## 04.3.1 Servo protection mechanisms

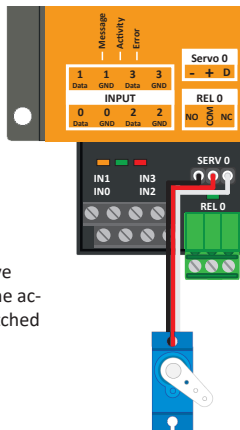
The ReadyServoTurn is designed for operational safety with monitoring functions.

Each servo output has overcurrent and short-circuit detection, which switches off the affected servo output in the event of a fault. The remaining servo outputs can still be used.

The error status (shut-off of a servo output) is signalled by a static **red error LED** on the module.

If a new servo control command is issued for the defective output, the control programme receives a message via the accessory feedback that the servo output could not be switched and therefore the route is not locked.

**The shut-off and error message can only be reset with a power reset of the ReadyServoTurn.**

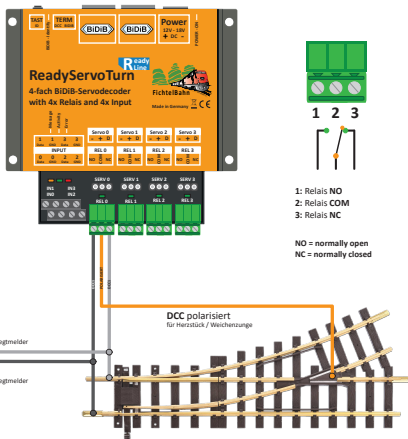


### Please notice:

The overcurrent and short-circuit detection is designed for a continuous load of 500mA / peak = 1A for each servo port.

## 04.4 Connecting a relay output

The ReadyServoTurn has 4 relay outputs (changeover contact) that can be used as a isolated application or with the servo motors and inputs.



The example illustration shows a frog polarisation on relay output 0. Alternatively, loads of up to 2A a continuous current per relay can be switched with the relay outputs.

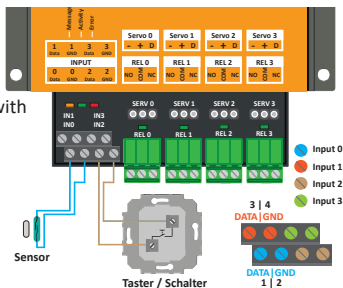
The switched-on state of the relay (1-NO connects to 2-COM) is signalled by the associated LED on the module lighting up.

## 04.5 Connection inputs

The ReadyServoTurn has 4 inputs against ground potential for pushbuttons / sensors for isolated applications or in combination with the outputs of the module.

### Possible applications:

- Position status of a servo with the aid of a Hall sensor or reed contact
- connected pushbuttons can trigger a servo movement and/or relay changeover.



### Please notice:

The inputs may only be connected to the module ground. Connecting an input to the system or another power supply will result in a short circuit that may damage the module.

## 05. Configuration of the module

The BiDiB-Wizard 2 is a Java program for displaying and configuring the BiDiB modules on the BiDiBus. The current version BiDiB-Wizard 2 is available for free download in our BiDiB-Wiki at <https://wiki.fichtelbahn.de> (in the overview tree under „Programme für BiDiB“ / „BiDiB-Wizard“).

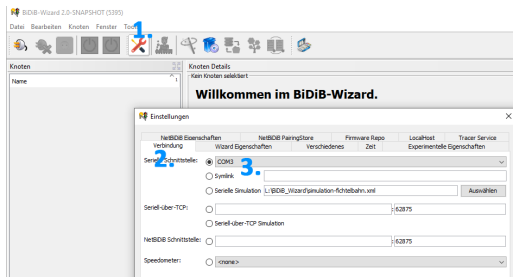
### Please notice:

The new automated BiDiB node configurator for this ReadyLine module is only available with the new BiDiB Wizard 2 version.

The Wizard version 1 is also compatible with this module, but only the classical macro programming is available.

### 05.1 Establishing a connection to the BiDiBus

The BiDiB interface (BiDiB-IF2 or GBMboost Master) is connected to the computer through a virtual serial port (USB). To establish a connection the correct serial port (3.) has to be selected under **Edit (1.) / Preferences (2.)**. In the drop down menu the correct **Serial port** has to be selected.

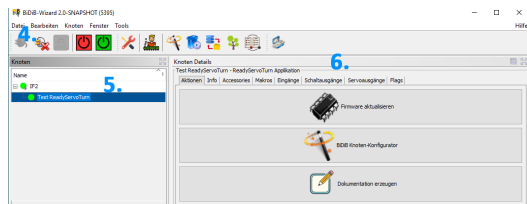


When using a BiDiB-Ethernet interface, e.g. IFnet, you will need the IP address of your interface.

### Please notice:

Only one program can access an active serial port simultaneously. If the railway controlling program is using the serial port this connection has to be terminated before the BiDiB-Wizard can make use of the serial port.

By clicking on the button with the **plug symbol** (4.) the connection will be initiated and all connected nodes will be loaded and shown in the **node tree view** (5.).



By double clicking on a **node in the node tree view** (5.) this node will be loaded and its functions and options will be shown in the **node detail window** (6.).

For the “ReadyServoTurn” module, there are two ways of achieving an operational state:

### 1. Easy to use - Node configurator for beginners

... easy and fast with a simple and intuitive user interface “BiDiB Node Configurator”.  
standard use cases are set up automatically.

| Window  | Description   |
|---------|---|
| Actions | <b>Update firmware</b><br>With this button the firmware can be updated (see Firmware Update on page 33)     |
| Actions | <b>BiDiB Node Configurator</b><br>Application examples can be loaded via this button. (see pageSeite 33 13) |

### 2. Individual macro configuration - for experienced users

...configure the application according to your own wishes individually with the help of the macros.

| Window         | Description   |
|----------------|---|
| Actions        | <b>Update firmware</b><br>With this button the firmware can be updated (see Firmware Update on page 33) |
| Info           | Technical information about the module  |
| Accessories    | Assigning and testing the individual switching outputs or servo outputs, check the feedback indication. |
| Macros         | Memory locations for each individual action, using the macro step chain                                 |
| Switch ports   | 4 relay outputs for switching their states ON / OFF   |
| Servo ports    | 4 servo outputs and their control characteristics, position up / down, speed                            |
| Flags          | Visualisation of internal flags for macro programmings  |
| CV Definitions | Reading and writing of module specific CV settings (see Module settings on page 29)                     |

## 05.2 Node Configurator

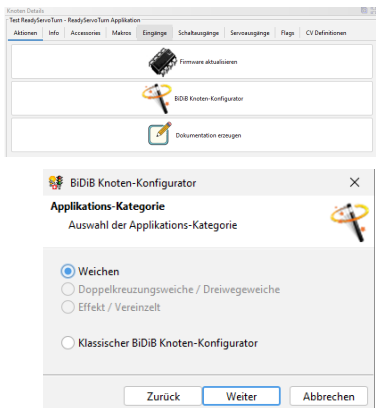
### 05.2.1 Using the assistant

The node configurator is available for simple applications, with the help of which the module is automatically configured.

Clicking on Actions / BiDiB Node Configurator opens a wizard that guides you through all settings and queries.

In the following window you can choose between the four main categories.

A detailed functional description with a potential connection diagram is described at the end of this chapter.

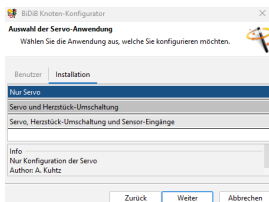


| Category                        | Application examples   |
|---------------------------------|--|
| Turnouts                        | 4x servo turnouts with / without frog polarisation or with / without position feedback           |
| Turnouts with external input    | 4x servo turnouts with/without frog polarisation that can also be operated via an external input |
| Doubleslip / three-way turnouts | Identical to the category "Turnouts" only with 2 drives on one turnout                           |
| Effect / Parts                  | 4x individual servos and 4x individual relays  |
| Classic BiDiB Node Configurator | Configurator for own scripts (import of additional scenarios)                                    |

After selecting the basic category, the required application must be chosen.  
e.g. only 4 servo motors

**Please notice:**

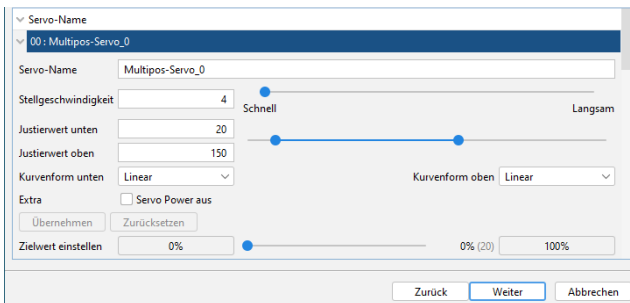
In addition to the configurator, the macros for a customised configuration can always be used. The wizard automatically creates macros, which can be changed after completion.



## 05.2.2 Servo settings

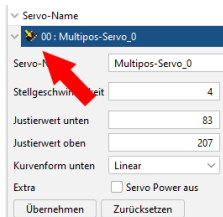
The following settings for servo 0 are exemplary for all 4 servo motors.

The window is part of the wizard, but can also be displayed at any time via the “Servo ports” tab.



| Setting                   | Explanation   |
|---------------------------|---|
| Servo name                | Field for entering a user-defined name  |
| Speed                     | Definition of the servo's orbital speed   |
| Trim value bottom and top | Definition of the working range of the rotary movement (down = 0% / up = 100%). Key commands: down = arrow right and left / up = CTRL + arrow right and left) |
| Curve form bottom and top | Servo motion profile: Linear, Soft, Bounce, Rebound, User   |
| Servo power off           | The servo is de-energised after reaching the end position. This function prevents servo hum, but the servo loses its holding force.                           |
| Adjust destination value  | The 0% / 100% buttons and the slider can be used to test the servo settings. <b>Clicking on “Apply” is necessary for the setting to be applied!</b>           |

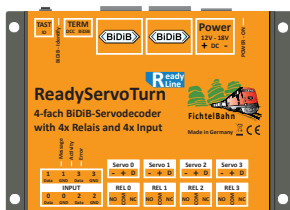
Changes to the settings are made locally and are not yet saved on the module. These changes are only saved on the module with a click on “Apply”. The “Destination value” function test can only demonstrate the new changes if they have been saved beforehand. The last saved values can be read with “Revert”.



## 05.2.3 Servo with relay and feedback

In this set-up, the module can operate 1-4 servos, switch 1-4 relays and report back the position with the 4 inputs for each servo.

The BiDiB Node Configurator sets up the macros and accessories according to this illustration. The servos, relay outputs and inputs must be connected in the same way.



### Konfiguration und Anschlusskonzept

#### Accessory 0

**Begriff 0** Servo 0 = 0%  
Relais 0 = OFF / DCC1  
IN 0 = OFF

**Begriff 1** Servo 0 = 100%  
Relais 0 = ON/ DCC2  
IN 0 = ON

#### Accessory 1

**Begriff 0** Servo 1 = 0%  
Relais 1 = OFF / DCC1  
IN 1 = OFF

**Begriff 1** Servo 1 = 100%  
Relais 1 = ON/ DCC2  
IN 1 = ON

#### Accessory 2

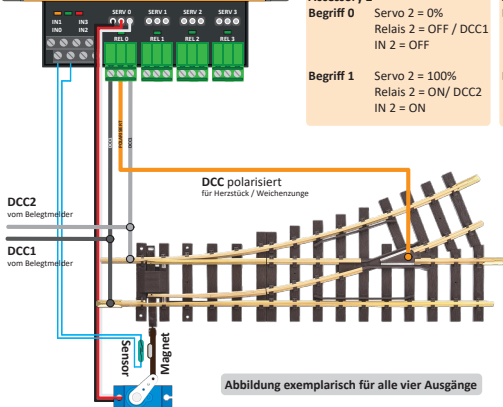
**Begriff 0** Servo 2 = 0%  
Relais 2 = OFF / DCC1  
IN 2 = OFF

**Begriff 1** Servo 2 = 100%  
Relais 2 = ON/ DCC2  
IN 2 = ON

#### Accessory 3

**Begriff 0** Servo 3 = 0%  
Relais 3 = OFF / DCC1  
IN 3 = OFF

**Begriff 1** Servo 3 = 100%  
Relais 3 = ON/ DCC2  
IN 3 = ON



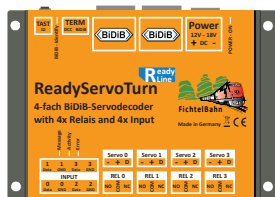
- Input 0
- Input 1
- Input 2
- Input 3

| Node configurator                       | Application examples                                       |
|---|--|
| Turnouts • Servo, frog relay and sensor | Servo turnout with frog polarisation and position feedback |

## 05.2.4 Servo with relay

In this set-up, the module can operate 1-4 servos and switch 1-4 relays.

The BiDiB node configurator creates the macros and accessories according to this illustration. The servos and relay outputs must be connected in the same way.



### Konfiguration und Anschlusskonzept

#### Accessory 0

**Begriff 0** Servo 0 = 0%  
Relais 0 = OFF / DCC1

**Begriff 1** Servo 0 = 100%  
Relais 0 = ON/ DCC2

#### Accessory 1

**Begriff 0** Servo 1 = 0%  
Relais 1 = OFF / DCC1

**Begriff 1** Servo 1 = 100%  
Relais 1 = ON/ DCC2

#### Accessory 2

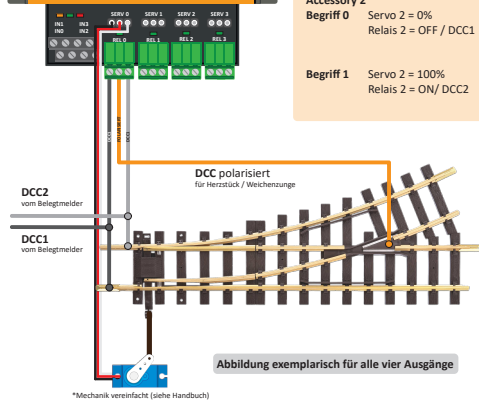
**Begriff 0** Servo 2 = 0%  
Relais 2 = OFF / DCC1

**Begriff 1** Servo 2 = 100%  
Relais 2 = ON/ DCC2

#### Accessory 3

**Begriff 0** Servo 3 = 0%  
Relais 3 = OFF / DCC1

**Begriff 1** Servo 3 = 100%  
Relais 3 = ON/ DCC2



| Node configurator               | Application examples                 |
|---------------------------------|--------------------------------------|
| Turnouts • servo and frog relay | Servo turnout with frog polarisation |

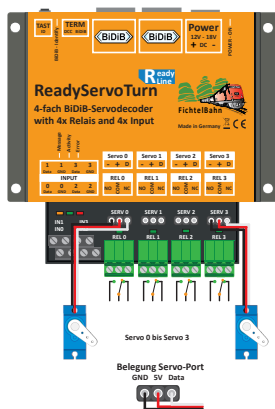




## 05.2.6 Servo and relay separated (effects)

In this set-up, the module can move 1-4 servos and, separately, the relays 1-4. The servos and the relays are divided between separate accessories 0-7.

The BiDiB Node Configurator creates the macros and accessories according to this illustration. The servos or relay outputs must be connected the same way.



### Konfiguration und Anschlusskonzept

#### Servo 0-3:

##### Accessory 0

Begriff 0 Servo 0 = 0%

Begriff 1 Servo 0 = 100%

##### Accessory 1

Begriff 0 Servo 1 = 0%

Begriff 1 Servo 1 = 100%

##### Accessory 2

Begriff 0 Servo 2 = 0%

Begriff 1 Servo 2 = 100%

##### Accessory 3

Begriff 0 Servo 3 = 0%

Begriff 1 Servo 3 = 100%

#### Relais 0-3:

##### Accessory 4

Begriff 0 Relais 0 = OFF

Begriff 1 Relais 0 = ON

##### Accessory 5

Begriff 0 Relais 1 = OFF

Begriff 1 Relais 1 = ON

##### Accessory 6

Begriff 0 Relais 2 = OFF

Begriff 1 Relais 2 = ON

##### Accessory 7

Begriff 0 Relais 3 = OFF

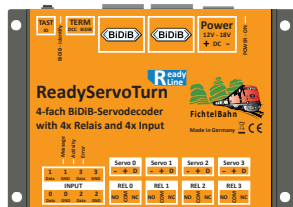
Begriff 1 Relais 3 = ON

| Node configurator                          | Application examples  |
|--|---|
| Effect / Parts • servo only                | Effect applications e.g. shed gates, waving pedestrians                           |
| Effect / Parts • Servo and relay separated | Effect applications for servo movements and separate relay for switching of loads |

## 05.2.7 Input operates servo and relay

With this set-up, the inputs can also trigger the operation of servo and relay in addition to the accessory. With input state 0, the servo moves to the 0% position. When the state is changed to 1, the servo moves to 100% and switches the relay on. After configuration over the BiDiBus, the module can be used standalone (without BiDiBus connection).

The BiDiB Node Configurator sets up the macros and accessories according to this illustration. The servos, relay outputs and inputs must be connected the same way.



### Konfiguration und Anschlusskonzept

#### Accessory 0

##### Begriff 0

oder Input 0 = 0  
Servo 0 = 0%  
Relais 0 = OFF / DCC1

##### Begriff 1

oder Input 0 = 1  
Servo 0 = 100%  
Relais 0 = ON / DCC2

#### Accessory 1

##### Begriff 0

oder Input 1 = 0  
Servo 1 = 0%  
Relais 1 = OFF / DCC1

##### Begriff 1

oder Input 1 = 1  
Servo 1 = 100%  
Relais 1 = ON / DCC2

#### Accessory 2

##### Begriff 0

oder Input 2 = 0  
Servo 2 = 0%  
Relais 2 = OFF / DCC1

##### Begriff 1

oder Input 2 = 1  
Servo 2 = 100%  
Relais 2 = ON / DCC2

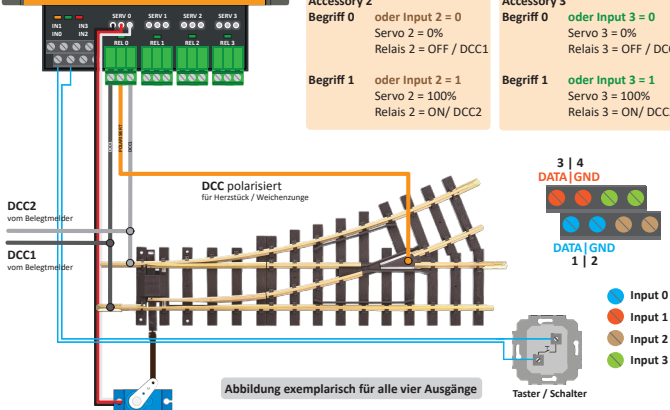
#### Accessory 3

##### Begriff 0

oder Input 3 = 0  
Servo 3 = 0%  
Relais 3 = OFF / DCC1

##### Begriff 1

oder Input 3 = 1  
Servo 3 = 100%  
Relais 3 = ON / DCC2



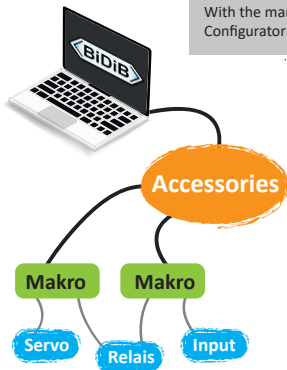
| Node configurator                        | Application examples  |
|--|---|
| Turnouts • servo, frog relays and inputs | Input operates servo with frog polarisation e.g. external control panel or push-button at the edge of the layout / manual control |

## 05.3 Manual Configuration

In addition to the predefined configurations provided by the node configurator, you can also create your own configurations and sequences using the macros and accessories.

**Please notice:**

With the manual configuration even configurations created by the Node Configurator can be amended.



### Correlation of output, input, macros and accessories:

The specific parameters of the outputs and inputs can be defined in the hardware settings (servo, switching outputs and inputs).

These ports are linked in the macros as sequences and thus result in the desired effect, movement sequence or switching process. A macro describes the sequence of an accessory aspect: e.g. servo movement from turnout position “straight” to turnout position “diverging”.

### Example:

A turnout with two aspects has 2 macros. A signal with seven aspects has 7 individual macros.

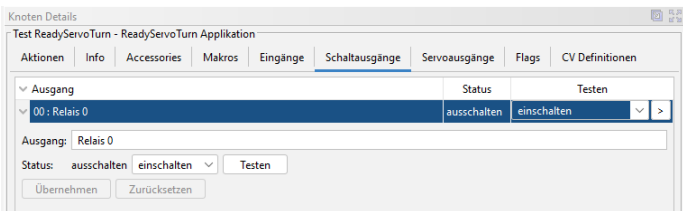
The top level **accessory** is the linking element and creates the link between aspects and an action. This link is called an **accessory** and can be controlled by the control program.

In the following chapters, the individual windows (Ports, Macro and Accessory) are explained in more detail, with the example application “Servo, Relay and Feedback” from the Node Configurator. For other applications, individual steps can be omitted or configured differently.

### 05.3.1 Servo ports

The “Servo ports” window is identical to the “Adjust the servos” window of the Node Configurator. See chapter: 05.2.2 Servo settings.

### 05.3.2 Switch ports (relay)



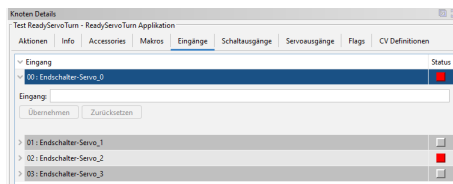
In our example, in addition to the servo movement, a polarisation of the frog is needed, which should be done with the first relay.

The 4 relays are directly connected to the 4 outputs of this module.

There are very few configuration parameters, except for a “turn on / turn off” function test. The switching status “relay ON” is indicated by the lighting of the corresponding LED 0-3.

| Switch port (relay) | Description   |
|---------------------|---|
| Output              | Field for entering a user-defined name - e.g. relay 0. The name is not permanently saved until the “Apply” button is pressed.   |
| Status              | Display of the current switching status of the relay  |
| Test                | This can be used to switch the relay on or off for test purposes. With the “Test toggle” setting, the port is switched on and off automatically until the function is deactivated or power to the module is cycled. |

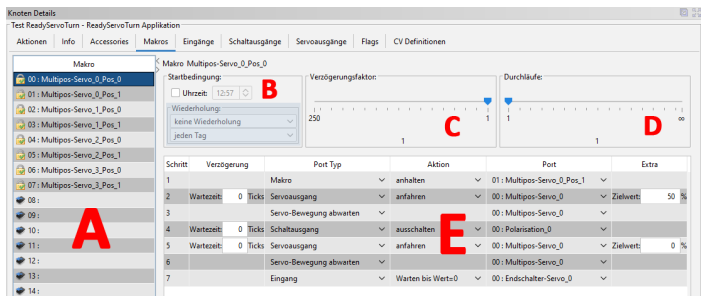
### 05.3.3 Input ports



In the “Input ports” window, the states of the 4 inputs can be monitored. A user-defined name can be assigned in the “Input” field. A red box indicates an active input, a grey box indicates an inactive input.

## 05.3.4 Macros

The actual configuration of the sequence takes place in window "Macro".



Test ReadyServoTurn - ReadyServoTurn Applikation

Aktionen Info Accessories Makros Eingänge Schaltausgänge Servoausgänge Flags CV Definitionen

Makro Multipos-Servo\_0\_Pos\_0

Startbedingung:  Uhrzeit: 12:57 **B** Wiederholung: keine Wiederholung jeden Tag

Verzögerungsfaktor: 250 **C**

Durchläufe: 1 **D**

| Schritt | Verzögerung        | Port Typ                | Aktion               | Port                       | Extra          |
|---------|--------------------|-------------------------|----------------------|----------------------------|----------------|
| 1       |                    | Makro                   | anhalten             | 01: Multipos-Servo_0_Pos_1 |                |
| 2       | Wartezeit: 0 Ticks | Servoausgang            | anfahren             | 00: Multipos-Servo_0       | Zielwert: 50 % |
| 3       |                    | Servo-Bewegung abwarten |                      |                            |                |
| 4       | Wartezeit: 0 Ticks | Schaltausgang           | ausschalten <b>E</b> | 00: Polarisation_0         |                |
| 5       | Wartezeit: 0 Ticks | Servoausgang            | anfahren <b>E</b>    | 00: Multipos-Servo_0       | Zielwert: 0 %  |
| 6       |                    | Servo-Bewegung abwarten |                      |                            |                |
| 7       |                    | Eingang                 | Warten bis Wert=0    | 00: Endschalter-Servo_0    |                |

| Term | Description     |
|------|-----------------|
| A    | Macro           |
| B    | Start condition |
| C    | Slowdown factor |
| D    | Cycles          |
| E    | Macro step      |

Macro 00 to macro XX is a memory location on the module. In a macro (e.g. macro 00), a function sequence (step chain) can be stored, e.g. for an aspect "Set turnout from diverging (100%) to straight (0%)". The macro can be given a user-defined name.

If the time is not activated, the macro is only executed if it is started via an accessory or macro command. If the time is active, the macro is started via the internal model railway time of the BiDiBus at the set time once per day or repeatedly.

The basic clock of a macro tick is 20ms with a delay factor of 1. With this factor, the standard delay time of the macro (waiting time of 1 tick = 20ms) can be stretched (up to max. 1 tick = 5 seconds). This value only has to be changed if a waiting time of more than 5 seconds per macro step is necessary. This factor only affects this macro locally and has no global effect.

The default setting is 1, which means that the macro will run from step 1 to step X and ends with the last step. With a value greater than 1, the macro is repeated by this factor. Such a macro can be stopped with a second macro and the macro command "stop".

The sequence of the step chain (readable from left to right)

A macro step is comparable to a step chain and consists of:  
 Step -> Delay -> Select Port Type -> Action -> Port -> Extra / Target Value.  
 A step is read and also executed by the module in this order.

| Column | Description  |
|--------|--|
| Step   | Index number - no further function   |
| Delay  | Waiting time in ticks (1 tick = 20ms with a slowdown factor of 1). After the waiting time has elapsed, the action of the macro step is executed. |

| Schritt | Verzögerung        | Port Typ                | Aktion            | Port                     | Extra          |
|---------|--------------------|-------------------------|-------------------|--------------------------|----------------|
| 1       |                    | Makro                   | anhalten          | 00 : Multipos-Servo_0    |                |
| 2       | Wartezeit: 0 Ticks | krit. Bereich           | anfahren          | <kein>                   | Zielwert: 50 % |
| 3       |                    | Flag                    | anfahren          | 00 : Multipos-Servo_0    |                |
| 4       | Wartezeit: 0 Ticks | Eingang                 | ausschalten       | 01 : Multipos-Servo_1    |                |
| 5       | Wartezeit: 0 Ticks | Makro                   | anfahren          | 02 : Multipos-Servo_2    | Zielwert: 0 %  |
| 6       |                    | Servoausgang            |                   | 03 : Multipos-Servo_3    |                |
| 7       |                    | Servo-Bewegung abwarten |                   | 00 : Multipos-Servo_0    |                |
|         |                    | Schaltausgang           |                   |                          |                |
|         |                    | Accessory notify        |                   |                          |                |
|         |                    | Verzögerung             |                   |                          |                |
|         |                    | zufällige Verzögerung   |                   |                          |                |
|         |                    |                         | Warten bis Wert=0 | 00 : Endschalter-Servo_0 |                |

| Column           | Description  |
|------------------|--|
| Port Typ         | This selection window distinguishes between a hardware action (e.g. servo movement) or a software function (e.g. stop macro). Depending on the selection of this port type, the following menu entries for action, port and extra change.                |
| Crit. section    | With action "Start" and "End" a macro area can be marked. In this area, the macro cannot be interrupted by other macros or actions.  |
| Flag             | With action "query" and "set" a macro can react to a flag and pause the further process until this condition is fulfilled with a value = 1 or value = 0, e.g. for dependencies between different macros.   |
| Input port       | Query of a condition (value = 1 or value = 0) at an input of the module. The further execution of the macro is paused until this condition is fulfilled.   |
| Macro            | Any macro from the macro list can be started or stopped.   |
| Servo port       | A servo can be moved from target position X to target position Y   |
| Await servo move | Depending on the set rotation time (speed) of the servo, a movement can take shorter or longer. If the next macro step is to be executed only when the servo has reached the target position, this macro step must be inserted after the servo movement. |
| Switch port      | The switching output X (relay) is "switched on / off" depending on the selected action.  |
| Accessory notify | After each successful completion of an accessory aspect, a status feedback is sent to the host (successfully completed / error). By inserting this action, the status message can be sent before the complete macro has been executed.                   |
| Delay            | Delay step that delays the execution of the next step by the value X (1 tick = 20ms with a delay factor of 1).   |
| Random delay     | Random delay in the value range of 1 tick to x ticks (1 tick = 20ms with delay factor 1)   |
| Action           | Depending on the selected port type, the action to be performed can be selected  |
| Port             | Selection of the port (e.g. servo 0-3)   |
| Extra            | Special functions and settings e.g. target value 0-100% for servos   |

**Please notice:**

Macro steps can be added and deleted with a right click.

### Macro example:

Servo 0 is moved from target position 100% to target position 0%, the associated frog polarisation is reversed by relay 0 and the servo movement is monitored via input 0. As an example, the following illustration shows the associated macro generated by the configurator. The sequence consists of 4 actions marked with A, B, C, D.

| Schritt    | Verzögerung        | Port Typ                | Aktion            | Port                        | Extra          |
|------------|--------------------|-------------------------|-------------------|-----------------------------|----------------|
| 1 <b>A</b> |                    | Makro                   | anhalten          | 01 : Multipos-Servo_0_Pos_1 |                |
| 2 <b>B</b> | Wartezeit: 0 Ticks | Servoausgang            | anfahen           | 00 : Multipos-Servo_0       | Zielwert: 50 % |
| 3          |                    | Servo-Bewegung abwarten |                   | 00 : Multipos-Servo_0       |                |
| 4 <b>C</b> | Wartezeit: 0 Ticks | Schaltausgang           | ausschalten       | 00 : Polarisation_0         |                |
| 5 <b>B</b> | Wartezeit: 0 Ticks | Servoausgang            | anfahen           | 00 : Multipos-Servo_0       | Zielwert: 0 %  |
| 6          |                    | Servo-Bewegung abwarten |                   | 00 : Multipos-Servo_0       |                |
| 7 <b>D</b> |                    | Eingang                 | Warten bis Wert=0 | 00 : Endschalter-Servo_0    |                |

| Function | Description   |
|----------|---|
| A        | Conflict protection<br>The first macro step terminates the opposing macro, which is responsible for the opposite movement of the turnout from 0% to 100%. This step is very useful if an executed switch command is followed by an immediate counter command and the servo has a long turnaround time. In this case, the opposing macro is aborted and not executed to the end.   |
| B        | Servo movement<br>In the second step, servo 0 is moved from its current position to the target position of 50% with the servo settings from servo 0. The 4th macro step is only started when the servo has reached the calculated target position of 50%. The function in macro step 3 is responsible for this delay (wait). Similarly, macro steps 5 and 6 continue the servo movement to the 0% end position.   |
| C        | Switch relay<br>Macro step 4 stops the servo movement exactly at the middle position of the servo movement and switches the state from relay ON to relay OFF. This intermediate step at 50% of the servo movement is important because it ensures that the point blades have no contact with one side of the rail and can be switched over safely.  |
| D        | Feedback<br>The last macro step ensures that the macro is not terminated if the position feedback is not set to "true". If no movement occurs due to a technical error in the servo or its mechanics, then the current value = 1, is not changed to the new value = 0. In this case the macro step waits for the value = 0 and the macro cannot be terminated. The host that triggered the switching command via an accessory does not receive an accessory feedback "Macro completed" from the module and therefore the route could not be reserved or set. An accessory could also report back an error condition. (see feedback in the Accessories category) |

The opposing macro from the target position 0% to 100% has the same structure, except that it has inverted settings.

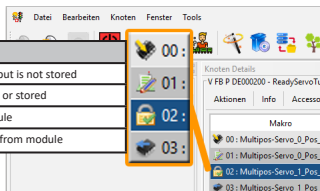
#### Please notice:

The sequences with the node configurator could have discrepancies with the connected inputs, polarity or servo direction. In this case, you could change the mechanics or the polarity of the connection. However, it is much easier if you adjust the settings in the created macro in such a case:

- the relay switches the wrong polarity** --> swap action "Turn off" to "Turn on".
- the servo moves in the wrong direction** --> change the target position from 0% to 100%.
- the input reacts to the wrong state** --> swap action "Wait until value=0" to value=1



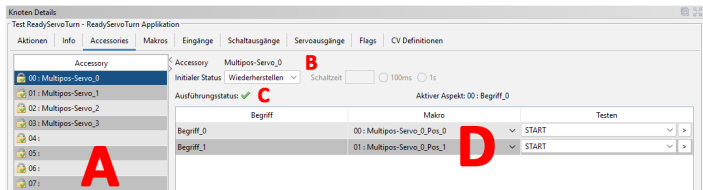
| Example at | Function                         | Description                            |
|------------|----------------------------------|--|
| 00         | Macro transferred but not stored | Macro can be tested, but is not stored |
| 01         | Unsaved changes                  | Macro not transferred or stored        |
| 02         | Permanently stored               | Macro stored on module                 |
| 03         | Macro not yet loaded             | Macro not yet loaded from module       |



### 05.3.5 Accessories

The “Accessories” window is the top level and acts as a link between the host program and the macro. Example:

Servo turnout with two aspects: Aspect 0: Movement 100% to 0%  
Aspect 1: Movement 0% to 100%



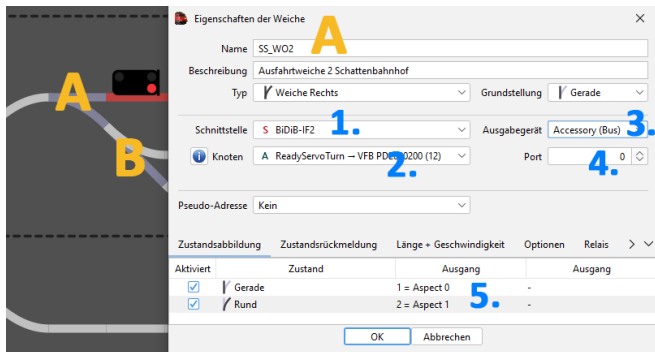
|           | Function  | Description   |           |  |         |   |          |  |
|-----------|---|---|-----------|--|---------|---|----------|--|
| A         | Accessory   | Accessory 00 to accessory XX are storage locations for sets of one or more aspects. In the current example, accessory 00 is the listing of the aspects for a turnout with two possible positions (straight/diverging).<br><b>An alternative example: a light signal with 7 states requires an accessory with 7 aspects.</b>   |           |  |         |   |          |  |
| B         | Initial state   | In the drop down menu “initial state” you can define the start-up behaviour for the accessory when the module is switched on. <table border="1" data-bbox="256 953 953 1085"> <tbody> <tr> <td>Unchanged</td> <td>No aspect is executed when the module is switched on</td> </tr> <tr> <td>Restore</td> <td>The module repeats the last sent aspect when it is switched on with this setting enabled.</td> </tr> <tr> <td>Aspect X</td> <td>The defined aspect X is executed when the module is switched on.</td> </tr> </tbody> </table> | Unchanged | No aspect is executed when the module is switched on | Restore | The module repeats the last sent aspect when it is switched on with this setting enabled. | Aspect X | The defined aspect X is executed when the module is switched on. |
| Unchanged | No aspect is executed when the module is switched on                                      |   |           |  |         |   |          |  |
| Restore   | The module repeats the last sent aspect when it is switched on with this setting enabled. |   |           |  |         |   |          |  |
| Aspect X  | The defined aspect X is executed when the module is switched on.                          |   |           |  |         |   |          |  |
| C         | Execution state   | If the associated macro is completed successfully, an “Accessory finished” message is automatically sent to the host. If an error occurs, an error is transmitted. The transmission is sent only once per execution and is signalled in the BiDiB Wizard with a symbol.   |           |  |         |   |          |  |
| D         | Accessory aspects   | Aspects can be added to the accessory here, each of which are assigned a macro.   |           |  |         |   |          |  |

**Please notice:**

Macro steps can be added and deleted with a right click

## 05.3.6 Integration into the control program

Finally, the accessory must be connected to the turnout symbol in the control program. In the following explanation, this procedure is explained using iTrain as an example. The procedure is identical to other programs, like RocRail, WinDigiPet or Modellstellwerk...



The screenshot shows the 'Eigenschaften der Weiche' dialog box. The 'Schnittstelle' field is set to 'S BiDiB-IF2', 'Knoten' to 'A ReadyServoTurn - VFB PD... 0200 (12)', and 'Ausgabegerät' to 'Accessory (Bus)'. The 'Port' is set to '0'. The 'Ausgang' table shows two aspects: '1 = Aspect 0' and '2 = Aspect 1', both with a '-' in the 'Ausgang' column.

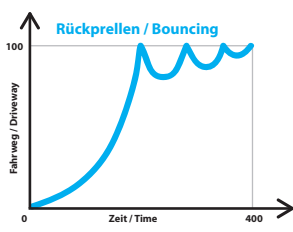
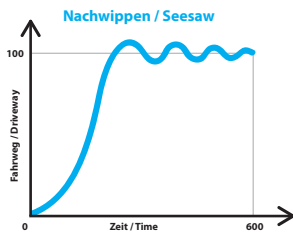
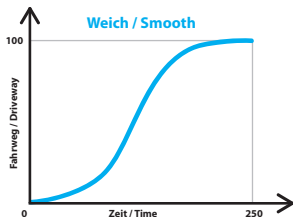
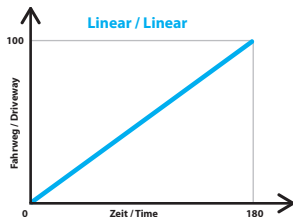
Turnout A is set in the “Properties of the turnout” with the following settings:

| Function        | Description   |
|-----------------|---|
| 1 Interface     | The BiDiB interface (command station) must be selected here.  |
| 2 Node          | In the node window, the correct BiDiB module to which the turnout is connected must be selected. If there are many modules, the displayed UID might help with the selection.                |
| 3 Output device | Depending on the control program, different protocols are supported - “Accessory” is required here.   |
| 4 Port          | The ReadyServoTurn could be used to control 4 turnouts that can be addressed via the accessories 0-3, for example. This accessory is selected in the port selection menu.                   |
| 5 Aspect        | Our example turnout has 2 aspects in the wizard, each of which has been assigned a macro. If the mapping is not correct, the assignment of straight and diverging can also be swapped here. |

For turnout B, port 0 is replaced by another value in the properties. e.g. 1 for Accessory 1.

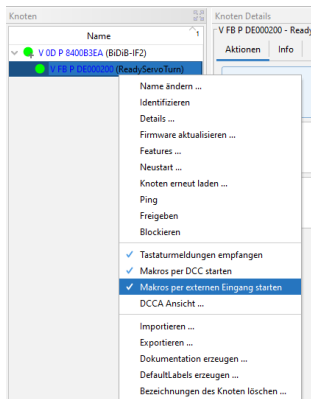
## 05.4 Servo motion profiles

The ReadyServoTurn offers predefined curves for typical model railway motion sequences. These curves can be changed for every servo movement in the node configurator or in the servo settings (tab: Servo ports).



| Motion profile | Typical usage  |
|----------------|--|
| Linear         | Straight movement with rough beginning and end             |
| Smooth         | Straight movement with soft beginning and end e.g. turnout |
| Seesaw         | Rocking of a semaphore signal                              |
| Bouncing       | Rebounding of a railway barrier                            |

## 05.5 Extended start conditions of macros



In addition to the known start conditions *Execution from host, time and start behaviour at module start*, a macro can be started from the 4 inputs of the module.

To achieve this, the function “Start macros from external input port” must be activated by right-clicking on the module name.

Now the 4 inputs are permanently linked with the 4 first macros. If input 0 changes state from 0 to 1, macro 0 is started.

The assignment is visible in the following table. If the function is activated, all 4 dependancies are created. If not all inputs are required for this function, the macros affected must be left without a function.

### Please notice:

It is advisable to not to use macros 0 to 3 for normal sequences. The normal program sequence should start with macro 4. If an input is required, a start instruction with “Start macro X” can be add to macro 0 to 3.

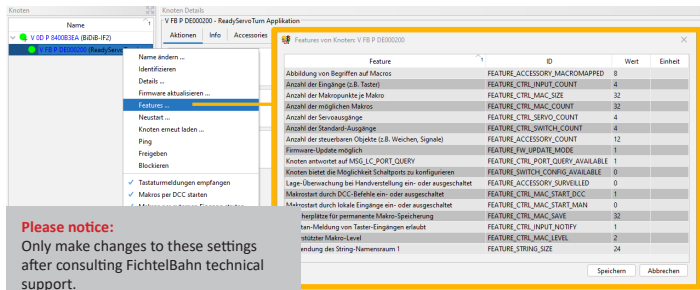
| Input   | Assigned macro |
|---------|----------------|
| Input 0 | macro 00       |
| Input 1 | macro 01       |
| Input 2 | macro 02       |
| Input 3 | macro 03       |

## 06. Module settings for the ReadyServoTurn

All settings that can be adjusted for the ReadyServoTurn can be accessed via the configuration tool “BiDiB-Wizard 2”. It is not possible to change the settings at the unit itself.

### 06.1 ReadyServoTurn features

Right-clicking on the ReadyServoTurn in the node list opens a window with further options. The “Features” window can be opened here.

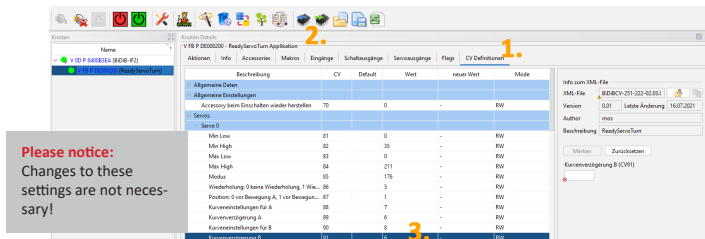


**Please notice:**  
Only make changes to these settings after consulting FichtelBahn technical support.

| Feature   | ID                                | Wert | Einheit |
|---|-----------------------------------|------|---------|
| Abbildung von Begriffen auf Macros                            | FEATURE_ACCESSORY_MACROMAPPED     | 0    |         |
| Anzahl der Eingänge (z.B. Tester)                             | FEATURE_CTRL_INPUT_COUNT          | 4    |         |
| Anzahl der Makropunkte je Makro                               | FEATURE_CTRL_MAC_SIZE             | 32   |         |
| Anzahl der möglichen Makros                                   | FEATURE_CTRL_MAC_COUNT            | 32   |         |
| Anzahl der Servoausgänge                                      | FEATURE_CTRL_SERVO_COUNT          | 4    |         |
| Anzahl der Standard-Ausgänge                                  | FEATURE_CTRL_SWITCH_COUNT         | 4    |         |
| Anzahl der steuerbaren Objekte (z.B. Weichen, Signale)        | FEATURE_ACCESSORY_COUNT           | 12   |         |
| Firmware-Updates möglich                                      | FEATURE_FW_UPDATE_MODE            | 1    |         |
| Knoten antwortet auf MDG_IC_PORT_QUERY                        | FEATURE_CTRL_PORT_QUERY_AVAILABLE | 1    |         |
| Knoten bietet die Möglichkeit Schaltschritte zu konfigurieren | FEATURE_SWITCH_CONFIG_AVAILABLE   | 0    |         |
| Lage-Überwachung bei Handverstellung ein- oder ausgeschaltet  | FEATURE_ACCESSORY_SURVEILLED      | 0    |         |
| Makrostart durch DCC-Befehle ein- oder ausgeschaltet          | FEATURE_CTRL_MAC_START_DCC        | 1    |         |
| Makrostart durch lokale Eingänge ein- oder ausgeschaltet      | FEATURE_CTRL_MAC_START_MAN        | 0    |         |
| herplätze für permanente Makro-Speicherung                    | FEATURE_CTRL_MAC_SAVE             | 32   |         |
| ben-Meldung von Tester-Eingängen erlaubt                      | FEATURE_CTRL_INPUT_NOTIFY         | 1    |         |
| spätester Makro-Level   | FEATURE_CTRL_MAC_LEVEL            | 2    |         |
| Verlängerung des String-Namensraum 1                          | FEATURE_STRING_SIZE               | 24   |         |

### 06.2 CV register of the ReadyServoTurn

With a click on the „CV Definitions“ tab (1), the CV registers become visible in the BiDiB Wizard. The current value can be read individually (right click on the individual CV, then read CVs) or all CVs of the module (2). Writing back new values is done with the same procedure (write CVs).



**Please notice:**  
Changes to these settings are not necessary!

| Bezeichnung                                  | CV | Default | Wert | neuer Wert | Mode |
|--|----|---------|------|------------|------|
| <b>Algemeine Daten</b>                       |    |         |      |            |      |
| <b>Algemeine Einstellungen</b>               |    |         |      |            |      |
| Accessory beim Einschalten wieder herstellen | 70 | 0       | -    | -          | RW   |
| <b>Servo</b>                                 |    |         |      |            |      |
| <b>Servo 0</b>                               |    |         |      |            |      |
| Min Low                                      | 01 | 0       | -    | -          | RW   |
| Min High                                     | 02 | 20      | -    | -          | RW   |
| Max Low                                      | 03 | 0       | -    | -          | RW   |
| Max High                                     | 04 | 211     | -    | -          | RW   |
| Modus  | 05 | 176     | -    | -          | RW   |
| Wiederholung: 0 keine Wiederholung, 1 Wic... | 06 | 5       | -    | -          | RW   |
| Position: 0 von Bewegung A, 1 von Bewegun... | 07 | 1       | -    | -          | RW   |
| Kurveneinstellungen für A                    | 08 | 7       | -    | -          | RW   |
| Kurveneinstellung A                          | 09 | 6       | -    | -          | RW   |
| Kurveneinstellungen für B                    | 90 | 8       | -    | -          | RW   |
| Kurveneinstellung B                          | 91 | 6       | -    | -          | RW   |

## 06.3 Settings for experienced users

The most important settings and configurations have been described in the previous chapters and are automatically configured with the port settings and macros (e.g. node configuration) by BiDiB Wizard 2.

**Changes to the features or CV registers are not necessary and should not be made without consulting FichtelBahn technical support.**

### 06.3.1 PowerOFF behaviour of the servos

#### Servo x: PowerOFF

Using **CV95 for servo 0** (see register for servo 1 to servo 3), the servo drive can be de-energised after servo movement, e.g. to prevent servo hum. Via the wizard settings of the servo port, this switch-off can only be activated with the delay of 20ms (value = 1).

If a higher delay time is necessary, this value must be changed in this register > 1..

| CV95 | Value | Description   |
|------|-------|---|
|      | 0     | Servo is not de-energised (default)   |
|      | 1-255 | Servo is switched off (delay with n*20ms).<br>Servo is de-energised after the delay (n*20ms) has elapsed. |

#### Servo x: PrePower

By using **CV96 for servo 0** (see register for servo 1 to servo 3), the time can be selected when the motion is started after switching on the servo power. This setting cannot be changed via the wizard settings of the servo port. (Default = 0)

| CV96 | Value | Description                                  |
|------|-------|--|
|      | 0     | PowerON is performed with movement (default) |
|      | 1-255 | PowerOn is performed n*5ms before movement   |

### 06.3.2 User-defined motion profiles

In the "Servo ports" tab, it is possible to select as motion profile the option "User-defined 1 and 2". These curves can be stored individually as vectors (position / time) in the CV register as points.

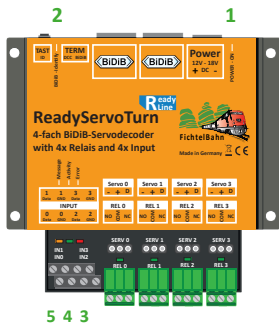
| CV  | Value range | Point | Value | Description                  |
|-----|-------------|-------|-------|------------------------------|
| 329 | 0-255       | 0     | 0     | Point 0 - Unit Time (n*20ms) |
| 330 | 25-230      | 0     | 25    | Point 0 - Unit Position      |

The upper table shows an example of the first point 0 of the user-defined curve. After the last curve point x, a point with the value = 0 in position and time must be added. With this, the module detects the end of the curve.

The value for position must not be <25 and >230.

| Servo-Benutzerkurven         |     |  |     |
|------------------------------|-----|--|-----|
| Servo Kurve 0                |     |  |     |
| Punkt 0: Zeit (Einheit 20ms) | 329 |  | 0   |
| Punkt 0: Position (0..255)   | 330 |  | 25  |
| Punkt 1: Zeit (Einheit 20ms) | 331 |  | 1   |
| Punkt 1: Position (0..255)   | 332 |  | 33  |
| Punkt 2: Zeit (Einheit 20ms) | 333 |  | 4   |
| Punkt 2: Position (0..255)   | 334 |  | 77  |
| Punkt 3: Zeit (Einheit 20ms) | 335 |  | 6   |
| Punkt 3: Position (0..255)   | 336 |  | 128 |
| Punkt 4: Zeit (Einheit 20ms) | 337 |  | 10  |
| Punkt 4: Position (0..255)   | 338 |  | 128 |
| Punkt 5: Zeit (Einheit 20ms) | 339 |  | 15  |
| Punkt 5: Position (0..255)   | 340 |  | 230 |
| Punkt 6: Zeit (Einheit 20ms) | 341 |  | 0   |
| Punkt 6: Position (0..255)   | 342 |  | 0   |

## 07. LED indication



### 07.1 Operating Modes

|                     |   |
|---------------------|---|
| <b>1</b>            | <b>Power-ON LED</b>   |
| fast flicker        | ReadyServoTurn is operating   |
| Double flashing     | Registration at the bus was rejected  |
| <b>2</b>            | <b>BiDiB-Identify LED</b>   |
| OFF                 | No connection to the BiDiBus  |
| continuous lighting | Connected to the BiDiBus  |
| fast flashing       | Identify - function active  |
| Double flashing     | Registration at the bus was rejected  |
| <b>3</b>            | <b>Error LED</b>  |
| continuous lighting | Short circuit or overload on one of the 4 servo ports. The relevant servo port has been switched off. |
| <b>4</b>            | <b>Activity LED</b>   |
| flashing            | There is a movement of one of the 4 servos.   |
| <b>5</b>            | <b>Message LED</b>  |
| continuous lighting | Module is in update mode  |

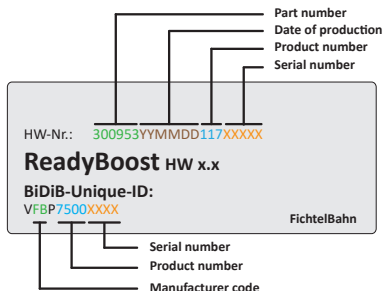
### 07.2 Error states at module start up

|   |  |
|---|--|
| <b>10x fast flashes (1, 2, 3)</b>   | <b>Power-ON LED, BiDiB-Identify LED, Message LED</b> |
| No bootloader found / no firmware update possible (contact the FichtelBahn-Support) |  |
| <b>Continuous Flashing (1, 2, 3)</b>  | <b>Power-ON LED, BiDiB-Identify LED, Message LED</b> |
| EEPROM faulty (conduct a firmware update, see "09. Firmware Update" on page 33)     |  |
| <b>Continuous Flashing (1, 2)</b>   | <b>Power-ON LED, BiDiB-Identify LED</b>              |
| No BiDiB-Unique-ID found (contact the FichtelBahn-Support)                          |  |

## 8. Background knowledge

### 8.1 What is a BiDiB-Unique-ID?

For working with the BiDiBus all BiDiB modules must have an Unique-ID which can be found as a label on the casing of the ReadyBoost.

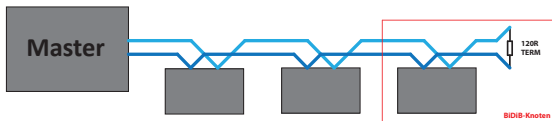


The Unique-ID is a unique identifier. This identifier enables the system to find the module regardless of its position within the BiDiBus. The BiDiB-System has a kind of „telephone book“ where to find a module.

Memorisable names can be given to each module with a host program (=computer railroad controlling program). The Unique-ID is the link between the given name on the computer and the module.

### 8.2 What is needed for bus termination?

The BiDiBus is a RS485 two wire connection which has been especially developed for high speed data transfer over long distances. This type of connection is used in an increasing number of industrial installations. Due to the properties high data rates can be achieved over a length of up to 200 m.



To guarantee an error free communication at this high data rates it is necessary to terminate the BiDiBus to avoid reflexions.

The terminating resistor of 120 Ohms is part of every BiDiB module and is activated by fitting the corresponding jumper.

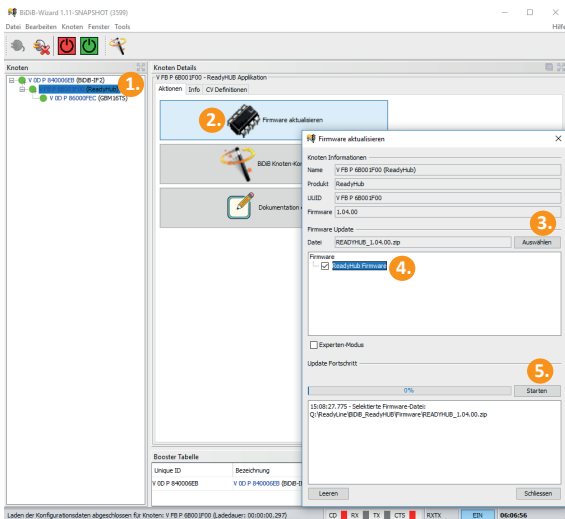


## 09. Firmware update

### 09.1 Function update

To keep the module compatible with new developments it is possible to conduct a firmware update through the BiDiBus. Start the Tool “BiDiB-Wizard” and double click on the entry „ReadyServoTurn“ (1.) in the tree view.

The ReadyServoTurn node will be loaded and displayed in the right window. (The illustration shows a different module name, but the function is identical).



Click on the button “Update firmware” (2.). In the new window choose the firmware zip file (3.) which can be loaded from the FichtelBahn website. Tick the check box (4.) for the desired firmware and proceed by clicking “Start” (5.). During the update process the Message-LED of the ReadyBoost module is lit.

## 09.2 Update under error conditions

In case of a faulty FLASH or EEPROM or a failed firmware update the module can be started in the bootloader mode manually. With the bootloader mode it is possible to redo an update with the tool “BiDiB-Wizard”.

To do so disconnect the module from the power supply (A) and press the button (H) while reconnecting the power supply (A).

In the tree view appears a module with the name “ReadyServoTurn Bootloader” (1.). This is a safety function to be able to redo the function update (see “09 Firmware update on page 33).

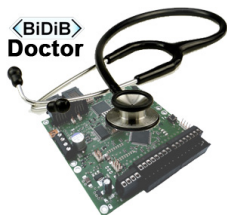
## 10. Support case and further help

For any further questions please contact our support center: <https://doctor.fichtelbahn.de>

A defective device can be sent in for repair with ticket number and / or error description.

In case of warranty you will receive a replacement or we will repair it for free.

If the damage does not fall under the product warranty, we charge a maximum of 50% of the current sales price for the costs of the repair. The lump sum for a review or repair is at least 20 euros. We reserve the right to refuse the repair of an assembly if this is not technically possible or uneconomical, there are no additional costs.



## 11. Warranty Information

We voluntarily grant a two year warranty period starting with the purchase date of the original buyer. This period ends also three years after manufacturing. The warranty provided doesn't affect the consumer's statutory rights. This warranty covers manufacturing defects in materials and workmanship at no charge. We reserve the right to repair, replace or refund the selling price. Any further claims shall be excluded. Claims for consequential damages or product liability shall only be accepted according to the statutory regulations. Following this operating instructions is a prerequisite for the warranty to be valid. Warranty claims become void under the following circumstances: modification of the circuit, repair attempts, incorrect operation or damage by negligent treatment or misuse.

## 12. Declaration of Conformity

Hiermit erklärt FichtelBahn, dass die Baugruppe „ReadyMagnet“ der Richtlinie 2014/30/EU entspricht. Der vollständige Text der EU-Konformitätserklärung ist unter der folgenden Internetadresse verfügbar: [www.fichtelbahn.de/declaration.html](http://www.fichtelbahn.de/declaration.html)

## 13. WEEE directive and packaging regulations

This product is in compliance with the requirements of EU directive regarding waste from electronic and electrical equipment (WEEE).

**WEEE registration number:** DE 52732575

Do not dispose this products with domestic waste. Local regulations may provide for separate collection of electrical products from the household or at municipal waste sites.

This product is in compliance with the requirements of the German packing regulations "VerpackG" from 01/01/2019.

**VerpackG number:** DE2189339488295



RailCom® ist das eingetragene Warenzeichen von:  
Lenz Elektronik GmbH | Vogelsang 14 | DE-35398 Gießen  
Zur Erhöhung der Lesbarkeit des Textes haben wir darauf verzichtet,  
bei jeder Verwendung des Begriffes darauf zu verweisen.



WEEE-Reg.-Nr. DE 52732575

Made in Germany



**FichtelBahn**

## FichtelBahn

Christoph Schörner  
Am Dummersberg 26  
D-91220 Schnaittach

Tel.: +49 9153 9703051  
support@fichtelbahn.de

© 2022 FichtelBahn®

Alle Rechte, insbesondere das Recht der  
Vervielfältigung und Verbreitung sowie  
der Übersetzung vorbehalten.

Vervielfältigungen und Reproduktionen  
in jeglicher Form bedürfen der  
schriftlichen Genehmigung durch FichtelBahn.

Technische Änderungen vorbehalten.