Handbücher/Manuals

VIPA art of automation

VIPA
Gesellschaft für Visualisierung
und Prozessautomatisierung mbH

Ohmstraße 4
D-91074 Herzogenaurach
Tel.: +49-9132-744-0
Fax: +49-9132-744-144
Internet: www.vipa.de
E-Mail: Info@vipa.de

Manual

VIPA System 200V

FM

Order No.: VIPA HB97E_FM
Rev. 06/29

The information contained in this manual is supplied without warranties. The information is subject to change without notice.

Ohmstraße 4, D-91074 Herzogenaurach,

Tel.: +49 (91 32) 744 -0

Fax.: +49 (91 32) 744-144

EMail: info@vipa.de

http://www.vipa.de

Hotline: +49 (91 32) 744-114

Disclaimer of liability

The contents of this manual were verified with respect to the hard- and software.

However, we assume no responsibility for any discrepancies or errors. The information in this manual is verified on a regular basis and any required corrections will be included in subsequent editions.

Suggestions for improvement are always welcome.

Trademarks

VIPA, System 100V, System 200V, System 300V and System 500V are registered trademarks of VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

STEP und S7-300 are registered trademarks of Siemens AG.

Any other trademarks referred to in the text are the trademarks of the respective owner and we acknowledge their registration.

About this manual

This manual describes the System 200V FM modules that are available from VIPA. In addition to the product summary it contains detailed descriptions of the different modules. You are provided with information on the connection and the usage of the System 200V FM modules. Every chapter is concluded with the technical data of the respective module.

Overview

Chapter 1: Basics

This introduction presents the VIPA System 200V as a centralized as well as decentralized automation system.

The chapter also contains general information about the System 200V, i.e. dimensions, installation and operating conditions.

Chapter 2: Assembly and installation guidelines

This chapter provides all the information required for the installation and the hook-up of a controller using the components of the System 200V.

Chapter 3: FM 250S - SSI module

In this chapter you receive every information, which is necessary for the deployment and operation of the SSI module FM 250S.

Chapter 4: FM 250 - Counter module

This chapter deals with the VIPA counter module FM 250. Here information of structure and parameterization can be found. A further part of this chapter is the description of the different counter modes and the respective interfacing.

Chapter 5: FM 253 - MotionControl Stepper

The chapter describes the VIPA MotionControl Stepper module FM 253. It contains information on the assembly, operating modes, data transfer and applications in conjunction with a stepper drive.

Chapter 6: FM 254 - MotionControl Servo

The chapter describes the VIPA MotionControl Servo module. It contains information on the assembly, operating modes, data transfer and applications in conjunction with a servo drive.

Contents

User considerations

Objective and contents

This manual describes the modules that are suitable for use in the System 200V. It contains a description of the construction, project implementation and the technical data.

Target audience

The manual is targeted at users who have a background in automation technology.

Structure of the manual

The manual consists of chapters. Every chapter provides a self-contained description of a specific topic.

Guide to the document

The following guides are available in the manual:

- an overall table of contents at the beginning of the manual
- an overview of the topics for every chapter
- an index at the end of the manual.

Availability

The manual is available in:

- printed form, on paper
- in electronic form as PDF-file (Adobe Acrobat Reader)

Icons Headings

Important passages in the text are highlighted by following icons and headings:

Danger!
Immediate or likely danger.
Personal injury is possible.

Attention!
Damages to property is likely if these warnings are not heeded.

Note!
Supplementary information and useful tips.

Safety information

Applications conforming with specifications

The System 200V is constructed and produced for:

- all VIPA System 200V components
- communication and process control
- general control and automation applications
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle

Danger!
This device is not certified for applications in

- in explosive environments (EX-zone)

Documentation

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

The following conditions must be met before using or commissioning the components described in this manual:

- Modification to the process control system should only be carried out when the system has been disconnected from power!
- Installation and modifications only by properly trained personnel
- The national rules and regulations of the respective country must be satisfied (installation, safety, EMC ...)

Disposal

National rules and regulations apply to the disposal of the unit!

Chapter 1 Basics

Overview

The focus of this chapter is on the introduction of the VIPA System 200V. Various options of configuring central and decentral systems are presented in a summary.

The chapter also contains the general specifications of the System 200V, i.e. dimensions, installation and environmental conditions.

Below follows a description of:

- Introduction of the System 200V
- General information, i.e. installation, operational safety and environmental conditions

Content

Safety information for Users

Handling of electrostatically sensitive modules

VIPA modules make use of highly integrated components in MOS-technology. These components are extremely sensitive to over-voltages that can occur during electrostatic discharges.

The following symbol is attached to modules that can be destroyed by electrostatic discharges:

The symbol is located on the module, the module rack or on packing material and it indicates the presence of electrostatic sensitive equipment.

It is possible that electrostatic sensitive equipment is destroyed by energies and voltages that are far less than the human threshold of perception. These voltages can occur where persons do not discharge themselves before handling electrostatically sensitive modules and they can damage components thereby, causing the module to become inoperable or unusable. Modules that have been damaged by electrostatic discharges may fail after a temperature change, mechanical shock or changes in the electrical load.

Only the consequent implementation of protection devices and meticulous attention to the applicable rules and regulations for handling the respective equipment can prevent failures of electrostatically sensitive modules.

Shipping of electrostatically sensitive modules

Modules have to be shipped in the original packing material.

Measurements and alterations on electrostatically sensitive modules

When you are conducting measurements on electrostatically sensitive modules you should take the following precautions:

- Floating instruments must be discharged before use.
- Instruments must be grounded.

Modifying electrostatically sensitive modules you should only use soldering irons with grounded tips.

Attention!

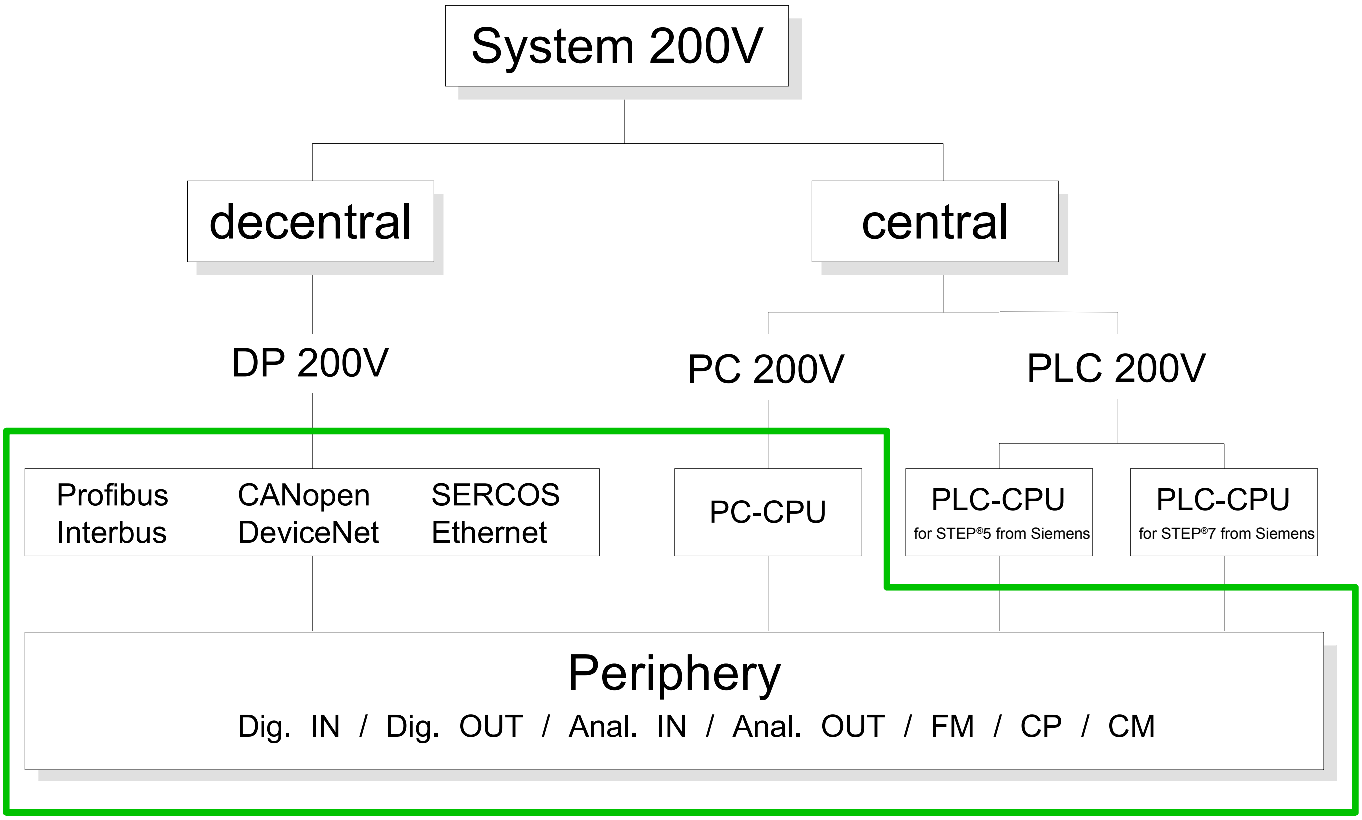
Personnel and instruments should be grounded when working on electrostatically sensitive modules.

Overview

The System 200V

The System 200V is a modular automation system for centralized and decentralized applications requiring low to medium performance specifications. The modules are installed directly on a 35mm DIN rail. Bus connectors inserted into the DIN rail provide the interconnecting bus.

The following figure illustrates the capabilities of the System 200V:



Components

Centralized system

The System 200V series consists of a number of PLC-CPUs. These are programmed in STEP®5 or STEP®7 from Siemens.

CPUs with integrated Ethernet interfaces or additional serial interfaces simplify the integration of the PLC into an existing network or the connection of additional peripheral equipment.

The application program is saved in Flash or an additional plug-in memory module.

The PC based CPU 288 can be used to implement operating/monitoring tasks, control applications or other file processing applications.

The modules are programmed in C++ or Pascal.

The PC 288-CPU provides an active interface to the backplane bus and can therefore be employed as central controller for all peripheral and function modules of the VIPA System 200V.

With the appropriate expansion interface the System 200V can support up to 4 rows.

Decentralized system

In combination with a Profibus DP master and slave the PLC-CPUs or the PC-CPU form the basis for a Profibus-DP network in accordance with DIN 19245-3. The DP network can be configured with WinNCS VIPA configuration tool res. Siemens SIMATIC Manager.

Other fieldbus systems may be connected by means of slaves for Interbus, CANopen, DeviceNet, SERCOS and Ethernet.

Peripheral modules

A large number of peripheral modules are available from VIPA, for example digital as well as analog inputs/outputs, counter functions, displacement sensors, positioners and serial communication modules.

These peripheral modules can be used in centralized as well as decentralized mode.

Integration over GSD File

The functionality of all VIPA system components are available via different GSD-files.

For the Profibus interface is software standardized, we are able to guarantee the full functionality by including a GSD-file using the Siemens SIMATIC Manager.

For every system family there is an own GSD-file. Actual GSD files can be found at ftp.vipa.de/support.

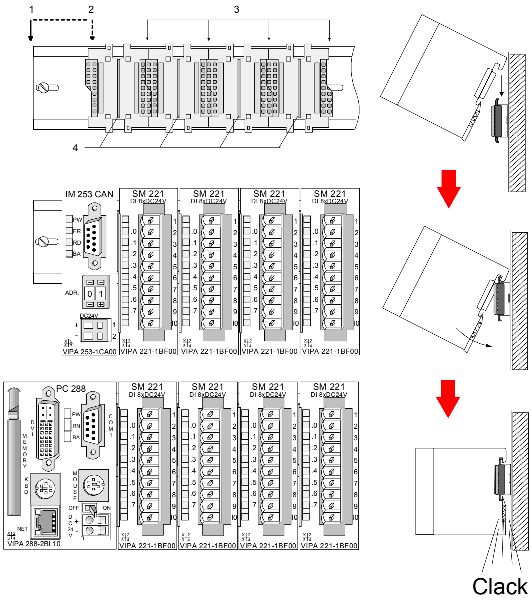
General description System 200V

Structure/ dimensions

- Standard 35mm DIN rail
- Peripheral modules with recessed labelling
- Dimensions of the basic enclosure:
  1tier width: (HxWxD) in mm: 76x25.4x74 in inches: 3x1x3
  2tier width: (HxWxD) in mm: 76x50.8x74 in inches: 3x2x3

Installation

Please note that you can only install header modules, like the CPU, the PC and couplers into plug-in location 1 or 1 and 2 (for double width modules).



[1] Header modules, like PC, CPU, bus couplers (double width)
[2] Header module (single width)
[3] Peripheral module
[4] Guide rails

Note

A maximum of 32 modules can be connected at the back plane bus. Take attention that here the **maximum sum current** of **3.5A** is not exeeded.

Please install modules with a high current consumption directly beside the header module.

Reliability

- Wiring by means of spring pressure connections (CageClamps) at the front-facing connector, core cross-section 0.08...2.5mm^2 or 1.5 mm^2 (18pole plug)
- Complete isolation of the wiring when modules are exchanged
- Every module is isolated from the backplane bus
- ESD/Burst acc. IEC 61000-4-2 / IEC 61000-4-4 (to level 3)
- Shock resistance acc. IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)

Environmental conditions

- Operating temperature: 0 ... +60°C
- Storage temperature: -25 ... +70°C
- Relative humidity: 5 ... 95% without condensation
- Ventilation by means of a fan is not required

Chapter 2 Assembly and installation guidelines

Overview

This chapter contains the information required to assemble and wire a controller consisting of Systems 200V components.

Below follows a description of:

- a general summary of the components
- steps required for the assembly and for wiring
- EMC guidelines for assembling the System 200V

Content

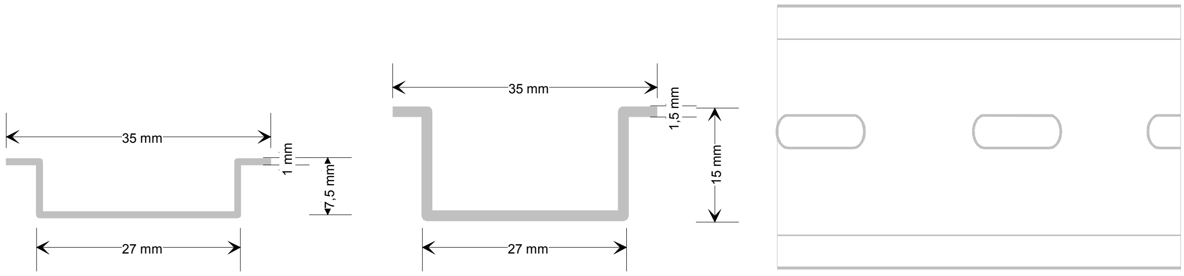
Overview

General

The modules are installed on a carrier rail. A bus connector provides interconnections between the modules. This bus connector links the modules via the backplane bus of the modules and it is placed into the profile rail that carries the modules.

Profile rail

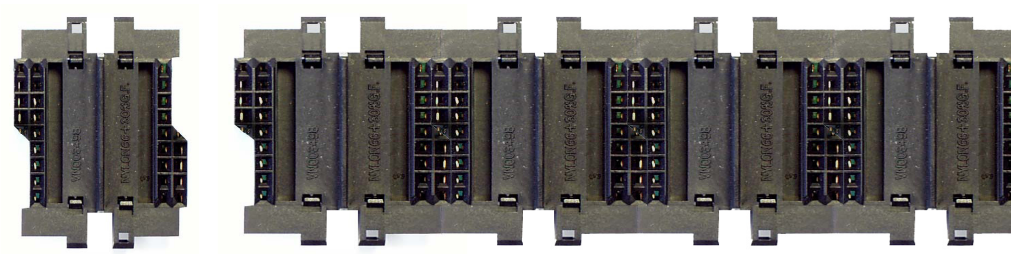
You may use the following standard 35mm profile rail to mount the System 200V modules:



Bus connector

System 200V modules communicate via a backplane bus connector. The backplane bus connector is isolated and available from VIPA in of 1-, 2-, 4- or 8tier width.

The following figure shows a 1tier connector and a 4tier connector bus:

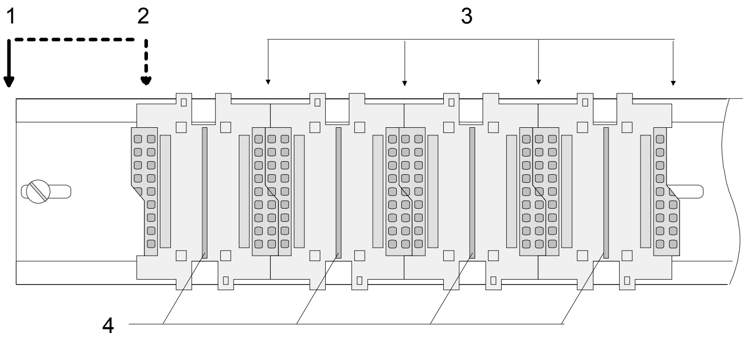


The bus connector is isolated and has to be inserted into the profile rail until it clips in its place and the bus connections protrude from the rail.

Profile rail installation

The following figure shows the installation of a 4tier width bus connector in a profile rail and the plug-in locations for the modules.

The different plug-in locations are defined by guide rails.

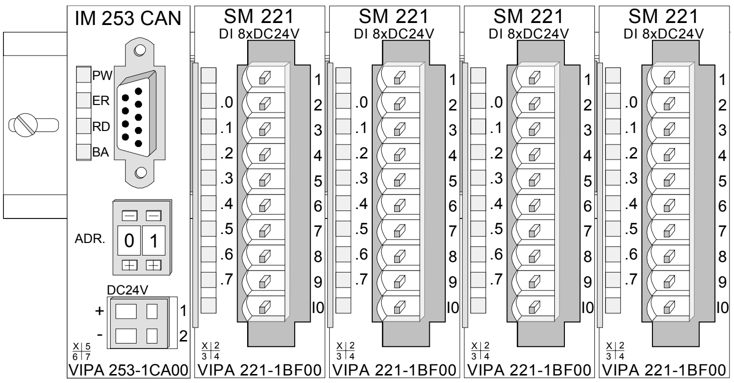


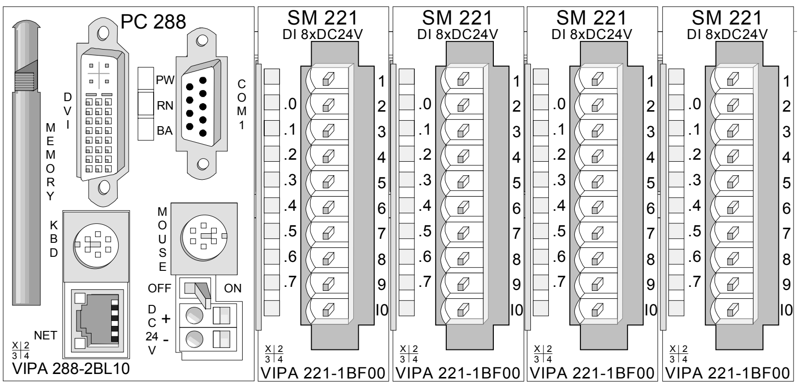
[1] Header module, like PC, CPU, bus coupler, if double width
[2] Header module (single width)
[3] Peripheral module
[4] Guide rails

Note

A maximum of 32 modules can be connected at the back plane bus.

Take attention that here the **maximum sum current** of **3.5A** is not exceeded.





Assembly regarding the current consumption

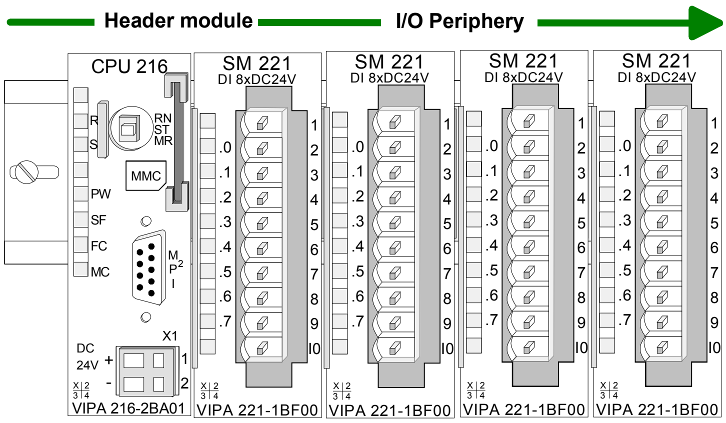
- Use bus connectors as long as possible.
- Sort the modules with a high current consumption right beside the header module. At ftp.vipa.de/manuals/system200v a list of current consumption of every System 200V module can be found.

Assembly horizontal respectively vertical

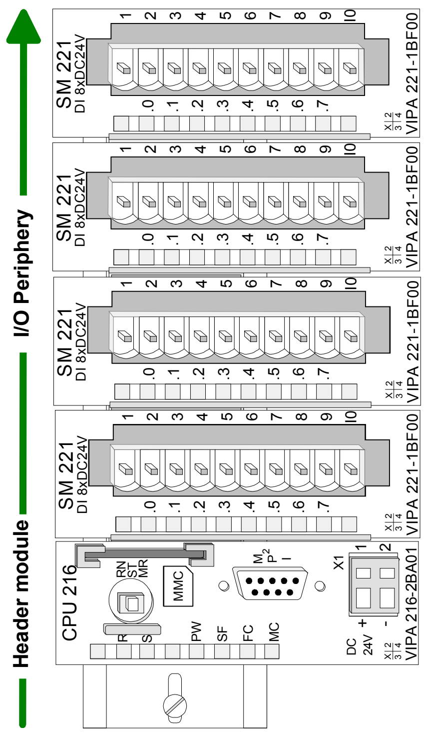
You may install the System 200V as well horizontal as vertical. Please regard the allowed environment temperatures:

- horizontal structure: from 0 to 60°
- vertical structure: from 0 to 40°

The horizontal structure always starts at the left side with a header module (CPU, bus coupler, PC), then you plug-in the peripheral modules beside to the right. You may plug-in maximum 32 peripheral modules.



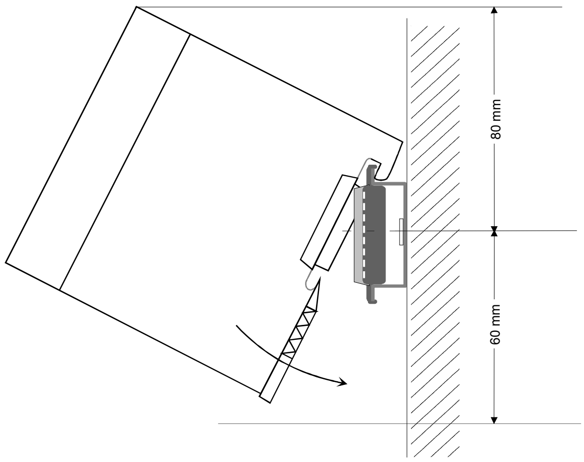
The vertical structure is turned for 90° against the clock.



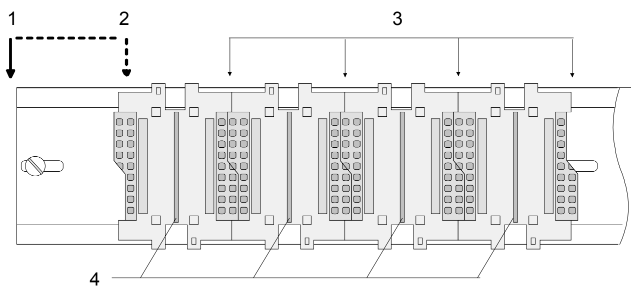
Assembly

Please follow these rules during the assembly!

- Turn off the power supply before you insert or remove any modules!
- Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the bus rail.



- Every row must be completed from left to right and it has to start with a header module (PC, CPU, and bus coupler).



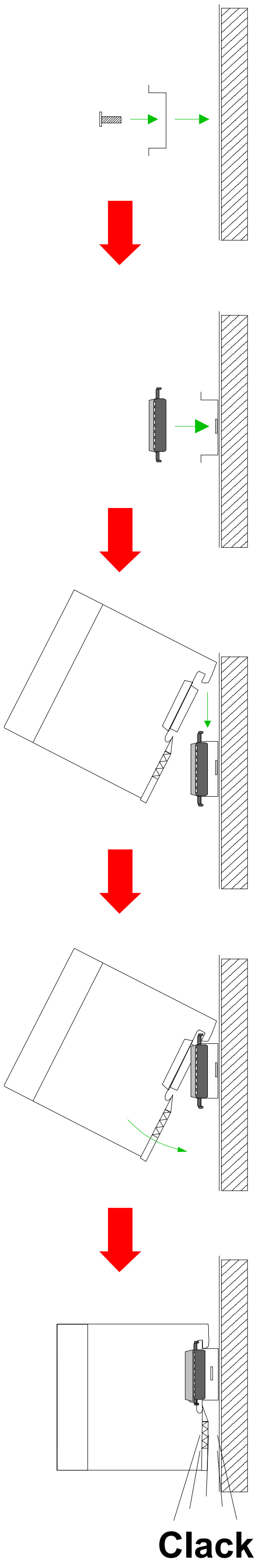
[1] Header module, like PC, CPU, bus coupler, if double width
[2] Header module (single width)
[3] Peripheral module
[4] Guide rails

- Modules are to install adjacent to each other. Gaps are not permitted between the modules since this would interrupt the backplane bus.
- A module is only installed properly and connected electrically when it has clicked into place with an audible click.
- Plug-in locations after the last module may remain unoccupied.

Note!
A maximum of 32 modules can be connected at the back plane bus. Take attention that here the maximum **sum current** of **3.5A** is not exceeded.

Assembly procedure

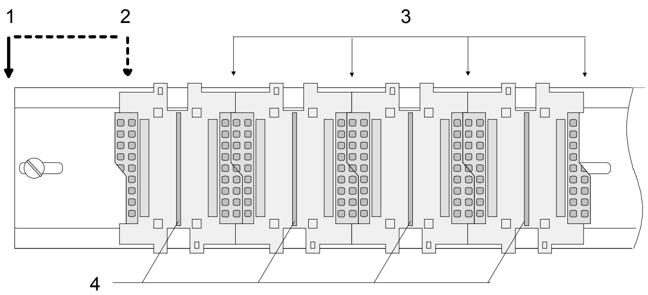
The following sequence represents the assembly procedure as viewed from the side.



- Install the profile rail. Make sure that a clearance of at least 60mm exists above and 80mm below the middle of the bus rail.

- Press the bus connector into the rail until it clips securely into place and the bus-connectors protrude from the profile rail. This provides the basis for the installation of your modules.

- Start at the outer left location with the installation of your header module like CPU, PC or bus coupler and install the peripheral modules to the right of this.



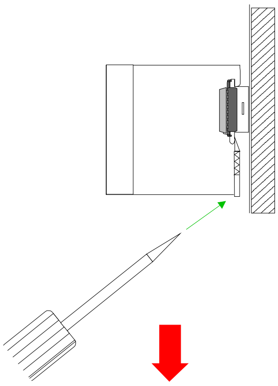
[1] Header module like PC, CPU, bus coupler
[2] Header module when this is a double width or a peripheral module
[3] Peripheral module
[4] Guide rails

- Insert the module that you are installing into the profile rail at an angle of 45 degrees from the top and rotate the module into place until it clicks into the profile rail with an audible click. The proper connection to the backplane bus can only be guaranteed when the module has properly clicked into place.

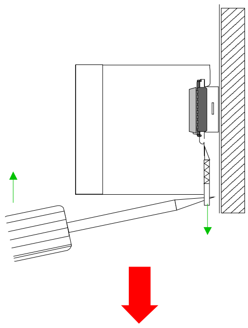
Attention!
Power must be turned off before modules are installed or removed!

Removal procedure

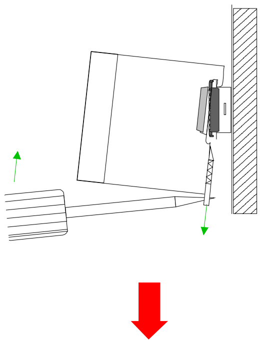
The following sequence shows the steps required for the removal of modules in a side view.



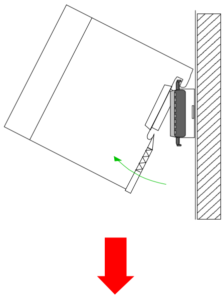
- The enclosure of the module has a spring-loaded clip at the bottom by which the module can be removed from the rail.
- Insert a screwdriver into the slot as shown.

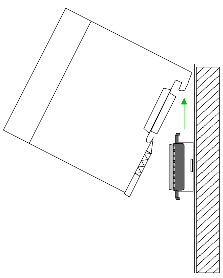


- The clip is unlocked by pressing the screwdriver in an upward direction.



- Withdraw the module with a slight rotation to the top.





Attention!

Power must be turned off before modules are installed or removed!

Please remember that the backplane bus is interrupted at the point where the module was removed!

Wiring

Outline

Most peripheral modules are equipped with a 10pole or an 18pole connector. This connector provides the electrical interface for the signaling and supply lines of the modules.

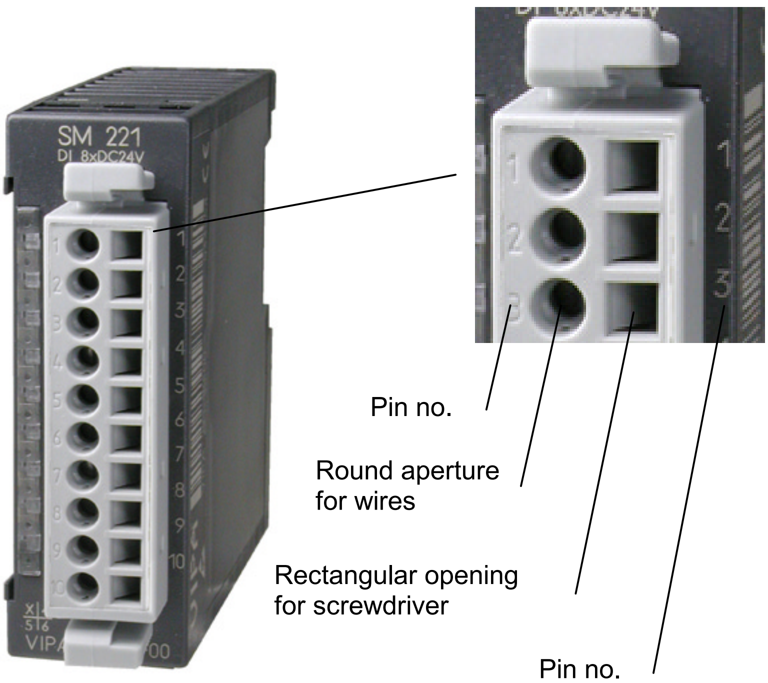
The modules carry spring-clip connectors for the interconnections and wiring.

The spring-clip connector technology simplifies the wiring requirements for signaling and power cables.

In contrast to screw terminal connections, spring-clip wiring is vibration proof. The assignment of the terminals is contained in the description of the respective modules.

You may connect conductors with a diameter from 0.08mm^2 up to 2.5mm^2 (max. 1.5mm^2 for 18pole connectors).

The following figure shows a module with a 10pole connector.

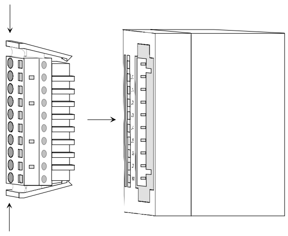


Note!

The spring-clip is destroyed if you insert the screwdriver into the opening for the hook-up wire!

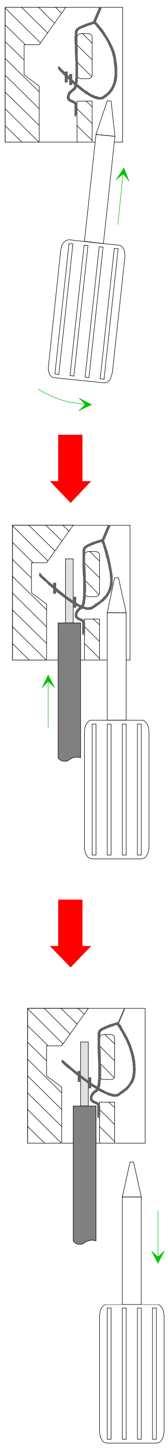
Make sure that you only insert the screwdriver into the square hole of the connector!

Wiring procedure



- Install the connector on the module until it locks with an audible click. For this purpose you press the two clips together as shown.
  The connector is now in a permanent position and can easily be wired.

The following section shows the wiring procedure from above.



- Insert a screwdriver at an angel into the square opening as shown.
- Press and hold the screwdriver in the opposite direction to open the contact spring.

- Insert the stripped end of the hook-up wire into the round opening. You can use wires with a diameter of 0.08mm² to 2.5mm² (1.5mm² for 18pole connectors).

- When you remove the screwdriver, the wire is clipped securely.

Wire the power supply connections first
followed by the signal cables (inputs and outputs).

Assembly dimensions

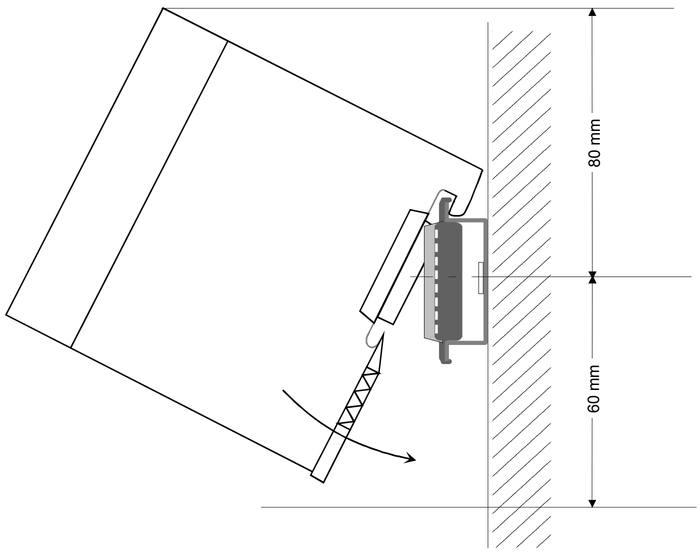
Overview

Here follow all the important dimensions of the System 200V.

Dimensions Basic enclosure

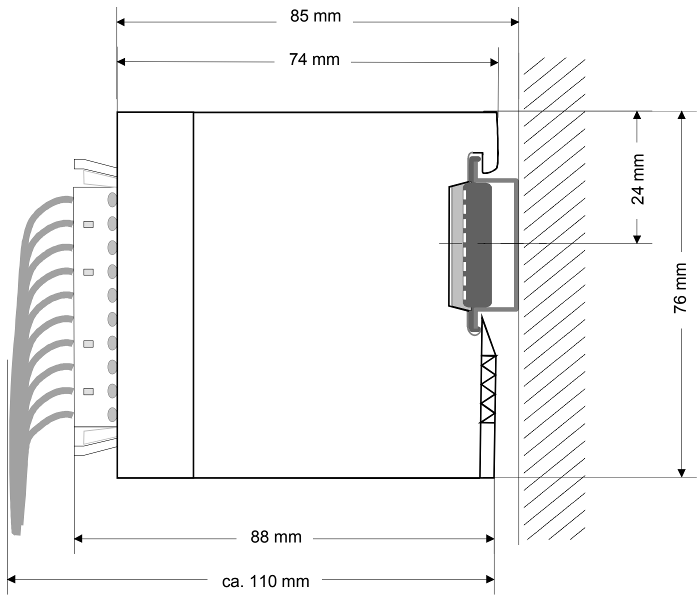
1tier width (HxWxD) in mm: 76 x 25.4 x 74
2tier width (HxWxD) in mm: 76 x 50.8 x 74

Installation dimensions

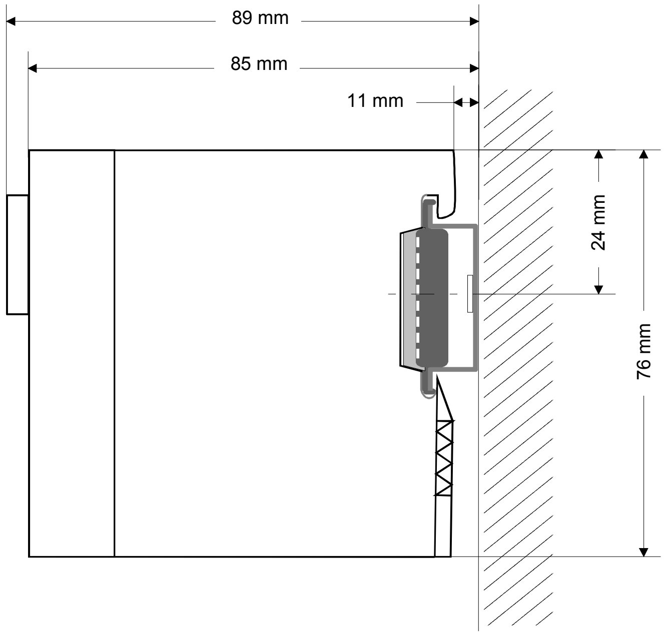


Installed and wired dimensions

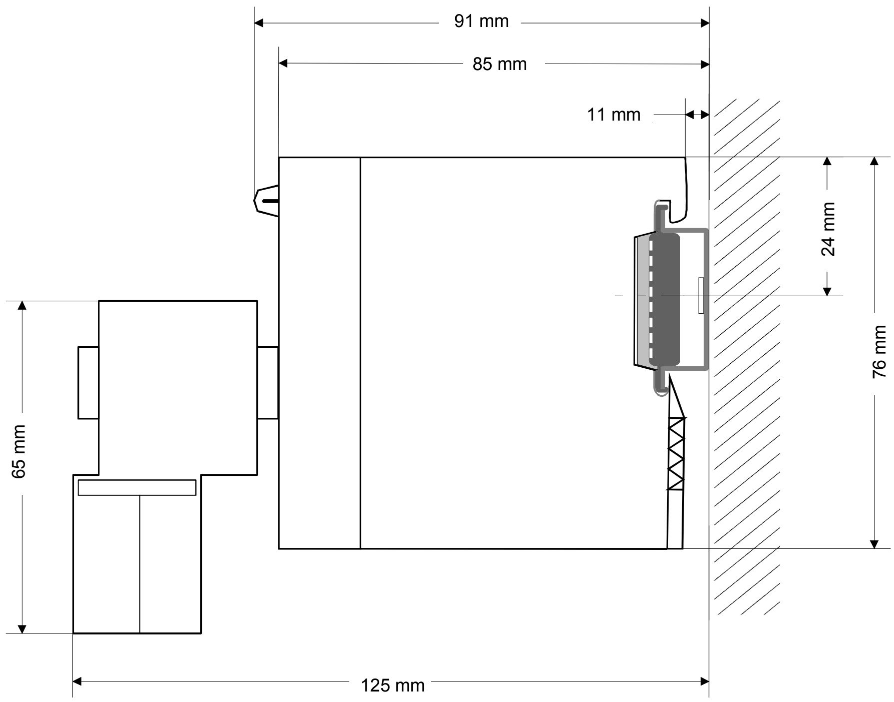
In- / Output modules



Function modules



CPUs here with EasyConn from VIPA



Installation guidelines

General

The installation guidelines contain information on the proper assembly of System 200V. Here we describe possible ways of interference that may disturb the controlling system and how you have to approach shielding and screening issues to ensure the electromagnetic compatibility (EMC).

What is EMC?

The term "electromagnetic compatibility" (EMC) refers to the ability of an electrical device to operate properly in an electromagnetic environment without interference from the environment or without the device causing illegal interference to the environment.

All System 200V components were developed for applications in harsh industrial environments and they comply with EMC requirements to a large degree. In spite of this you should implement an EMC strategy before installing any components which should include any possible source of interference.

Possible sources for disturbances

Electromagnetic interference can enter your system in many different ways:

- Fields
- I/O signal lines
- Bus system
- Power supply
- Protective conductor

Interference is coupled into your system in different ways, depending in the propagation medium (conducted or not) and the distance to the source of the interference.
We differentiate between:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiated power coupling

The most important rules for ensuring EMC

In many cases, adherence to a set of very elementary rules is sufficient to ensure EMC. For this reason we wish to advise you to heed the following rules when you are installing your controllers.

- During the installation of your components you have to ensure that any inactive metal components are grounded via a proper large-surface earth.
  - Install a central connection between the chassis ground and the earthing/protection system.
  - Interconnect any inactive metal components via low-impedance conductors with a large cross-sectional area.
  - Avoid aluminum components. Aluminum oxidizes easily and is therefore not suitable for grounding purposes.
- Ensure that wiring is routed properly during installation.
  - Divide the cabling into different types of cable. (Heavy current, power supply, signal and data lines).
  - Install heavy current lines and signal or data lines in separate channeling or cabling trusses.
  - Install signaling and data lines as close as possible to any metallic ground surfaces (e.g. frames, metal rails, sheet metal).
- Ensure that the screening of lines is grounded properly.
  - Data lines must be screened.
  - Analog lines must be screened. Where low-amplitude signals are transferred, it may be advisable to connect the screen on one side of the cable only.
  - Attach the screening of cables to the ground rail by means of large surface connectors located as close as possible to the point of entry. Clamp cables mechanically by means of cable clamps.
  - Ensure that the ground rail has a low-impedance connection to the cabinet/cubicle.
  - Use only metallic or metallized covers for the plugs of screened data lines.
- In critical cases you should implement special EMC measures.
  - Connect snubber networks to all inductive loads that are controlled by System 200V modules.
  - Use incandescent lamps for illumination purposes inside cabinets or cubicles, do not use fluorescent lamps.
- Create a single reference potential and ensure that all electrical equipment is grounded wherever possible.
  - Ensure that earthing measures are implemented effectively. The controllers are earthed to provide protection and for functional reasons.
  - Provide a star-shaped connection between the plant, cabinets/cubicles of the System 200V and the earthing/protection system. In this way you avoid ground loops.
  - Where potential differences exist you must install sufficiently large equipotential bonding conductors between the different parts of the plant.

Screening of cables

The screening of cables reduces the influence of electrical, magnetic or electromagnetic fields; we talk of attenuation.

The earthing rail that is connected conductively to the cabinet diverts interfering currents from screen conductors to ground. It is essential that the connection to the protective conductor is of low-impedance as the interfering currents could otherwise become a source of trouble in themselves.

The following should be noted when cables are screened:

- Use cables with braided screens wherever possible.
- The coverage of the screen should exceed 80%.
- Screens should always be grounded at both ends of cables. High frequency interference can only be suppressed by grounding cables on both ends.

  Grounding at one end may become necessary under exceptional circumstances. However, this only provides attenuation to low frequency interference. One-sided earthing may be of advantage where:
  - it is not possible to install equipotential bonding conductors.
  - analog signals (in the mV or µA range) are transferred.
  - foil-type shields (static shields) are used.
- Always use metallic or metallized covers for the plugs on data lines for serial links. Connect the screen of the data line to the cover. Do **not** connect the screen to PIN 1 of the plug!
- In a stationary environment it is recommended that the insulation is stripped from the screened cable interruption-free and to attach the screen to the screening/protective ground rail.
- Connect screening braids by means of metallic cable clamps. These clamps need a good electrical and large surface contact with the screen.
- Attach the screen of a cable to the grounding rail directly where the cable enters the cabinet/cubicle. Continue the screen right up to the System 200V module but do **not** connect the screen to ground at this point!

Please heed the following when you assemble the system!

Where potential differences exist between earthing connections it is possible that an equalizing current could be established where the screen of a cable is connected at both ends.

Remedy: install equipotential bonding conductors

Chapter 3 FM 250S - SSI module

Overview

This chapter contains every information on the interfacing, configuration and deployment of the SSI module FM 250S.

The following text describes:

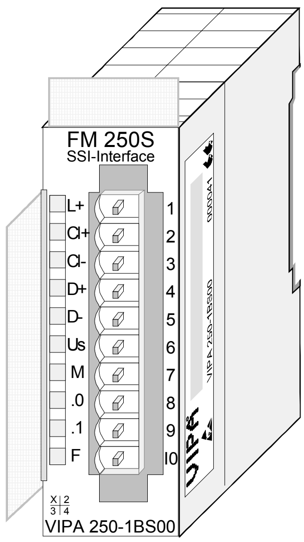
- Structure and deployment
- Technical data

Contents

System overview

Properties

- Wiring does not depend on the length of the data word. The interface always uses 4 wires.
- Maximum security due to the use of symmetrical clock and data signals.
- Secure data acquisition die to the use of single-step gray code (configurable).
- Galvanic isolation of receiver and encoder by means of optocouplers.
- 1 SSI channel
- Direct power supply to the SSI transducer via front-facing connector
- DC 24V power supply
- Baud rate selection between of 100kBaud and 600kBaud
- 2 configurable digital outputs, one may be used as hold input to freeze the current SSI transducer value
- Measured value available in gray or in binary code
- 4Byte of parameter data
- 4Byte of input data
- 4Byte of output data
- Configuration by means of control byte



Order data

Type	Order number	Description
FM 250S	VIPA 250-1BS00	SSI module

Structure

Functionality

SSI is the abbreviation for **S**ynchronous **S**erial **I**nterface. The SSI module provides the connection for transducers with absolute coding and a SSI module.

The module converts the serial information of the transducer into parallel information for the controller. Data can be transferred in gray or in binary code.

Configurable outputs

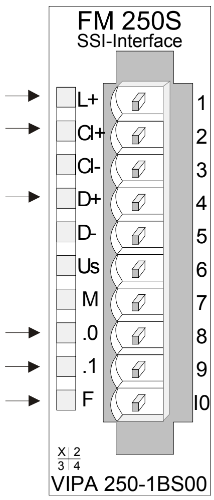
The interface has connections for the SSI signals clock, data and the transducer supply voltage as well as two additional outputs that may be set or reset when a limit value is exceeded.

Output 0 can also be programmed as hold input. This causes the SSI transducer value to be frozen when a 24V high level is applied to output 0. A low level will cause the transducer to transmit the actual SSI values.

You can also configure the outputs that they will remain set if the BASP signal is active.

Status indicator pin assignment

LED	Description
L+	LED (yellow) Supply voltage available
Ci+	LED (green) Clock output
D+	LED (green) Transducer data input
.0	LED (green) Input/output 0
.1	LED (green) Input/output 1
F	LED (red) Error /overload



Pin	Assignment
1	Supply voltage DC +24V
2	CLK+ (Output)
3	CLK- (Output)
4	Data+ (Input)
5	Data- (Input)
6	DC 24V SSI transducer supply voltage
7	Common SSI transducer supply
8	Input/output .0 and hold input
9	Input/output .1
10	Common of supply voltage

LEDs

The SSI module has a number of LEDs. The following table explains the significance of these LEDs:

Name	Color	Description
L+	yellow	Indicates that 24V power is available
C+	green	ON when clock pulses are transmitted OFF when hold function has been activated and 24V at I/O .0
D+	green	ON when data is received from the transducer (wiring test)
.0	green	ON when 24V power is available at I/O .0
.1	green	ON when 24V power is available at I/O .1
F	red	ON when short circuit or overload is detected on one of the two I/O .0/.1

Line distances

The baudrate depends on the length of the communication line and on the SSI transducer. Wiring has to consist of screened twisted pair cables. The specifications below are only intended as a guideline.

< 400m: → 100kBaud

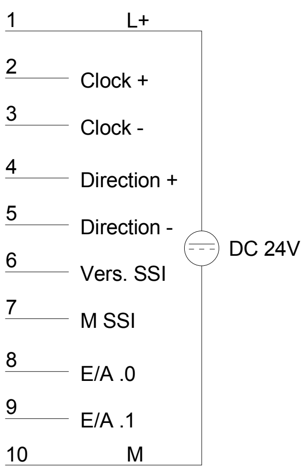
< 100m: → 300kBaud

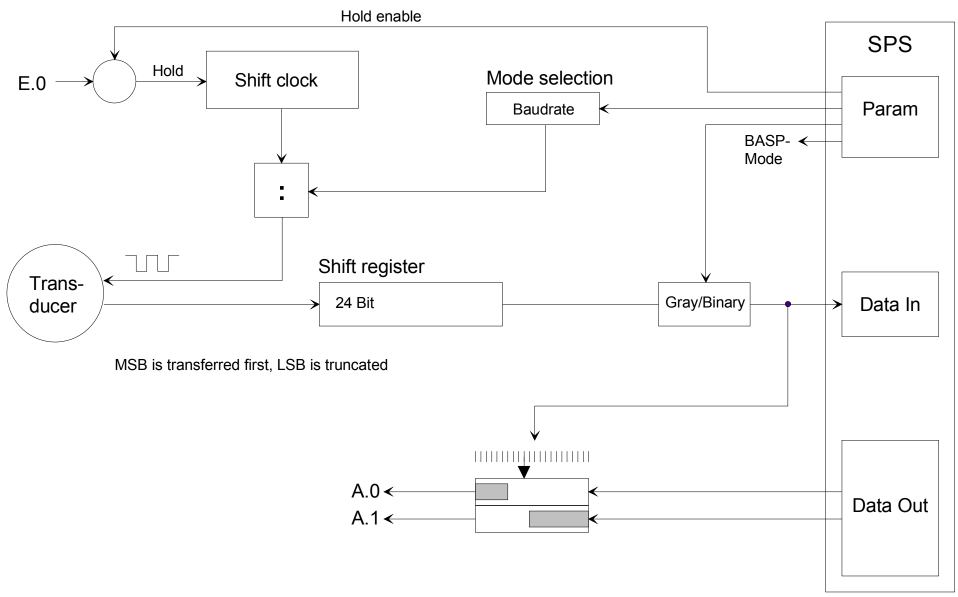
< 50m: → 600kBaud

Wiring diagram schematic diagramm

The SSI module has an internal power supply. This power supply requires a voltage of DC 24V via L+ and M.

The supply voltage provides power to the interface electronics as well as the SSI transducer connected with DC 24V to pin 6 and 7.





Deployment

Access to the SSI module

Input data (Data In)

The input data from the SSI transducer has a length of 4Byte. Byte 0 can be used as an I/O status indicator for the. Data is supplied in binary or in gray code, depending on the selected mode.

Byte	**Data In**
0	Bit 0: Status I/O .0 Bit 1: Status I/O .0 Bit 7 ... 2: reserved
1	SSI transducer value: HB
2	SSI transducer value: MB
3	SSI transducer value: LB

Output data (Data Out)

Data Out provides the option of controlling the 2 I/O ports on the SSI module depending on the value of a transducer input. Output data consists of 4Byte.

The SSI transducer stores 8Byte of output data, i.e. you may define two comparative values along with the respective control byte.

In the control byte you are able to specify how the reference value should affect which output. The status of the I/Os is signaled via the input bytes.

The following table shows the assignment of these output bytes.

Byte	**Data Out**
0	Bit 1 ... 0: set point value 00: no set point value 01: for output 0 10: for output 1 11: for both outputs Bit 2: reserved Bit 3: set conditions for output 0: when actual value exceeds comparison value 1: when actual value is less than comparison value Bit 7 ... 4: reserved
1	Comparison value: HB
2	Comparison value: MB
3	Comparison value: LB

Configuration data

4Byte of configuration data are transferred. In these bytes you define the baudrate, the coding and the analysis of the combined I/O .0 as well as the BASP signal.

The structure of the configuration data is as follows:

Byte	Bit 7 ... 0
0	Bit 7 ... 0: reserved
1	Bit 7 ... 0: reserved
2	Baud rate 0: 300kBaud (default) 1: 100kBaud 2: 300kBaud 3: 600kBaud 4...255: 300kBaud
3	Bit 0: Coding 0: Binary code (default) 1: Gray code Bit 2: SSI format 0: Multiturn (24 bit) 1: Singleturn (12 bit) Bit 4: Hold function 0: deactivate 1: activate Bit 7: BASP signal 0: ignore 1: analyze

Baud rate

The transducer connected to the SSI channel transmits serial data. It requires a clock pulse from the SSI module. The baud rate defines this clock. You may choose a value of 100, 300 or 600kBaud. The default setting is 300kBaud.

Coding

The gray code is a different form of binary code. The principle of the gray code is that two neighboring gray numbers will differ in exactly one single bit.

When the gray code is used, transmission errors can be detected easily as neighboring characters may only be different in a single location.

Table of rules for the gray code:

Decimal	Gray Code
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0

i.e. the last digit of the number results from the vertical repetition of the sequence "0 11 0", the penultimate digit results from the repetition "00 1111 00", the third-last number from the repetition of 4x"0", 8x"1" and again 4x"0", etc. (see columns in the table!).

Hold function

Here you define that I/O .0 should be used as hold input. When you have activated this function, the current transducer value will be stored when I/O .0 is connected to 24V. The transducer value is only updated when the 24V level is removed from I/O .0.

In this case you have to be aware that I/O .0 operates only in input mode.

BASP signal

BASP is a German abbreviation for command output inhibited, i.e. all outputs are reset and inhibited as long as the BASP signal is applied via the backplane bus. You may disable the evaluation of the BASP signals by setting this bit. This means that the outputs will remain set.

Technical data

SSI module
FM 250S

Electrical data	VIPA 250-1BS00
Number of channels	1
Number of outputs	2
Current consumption	120mA via backplane bus
Isolation	yes
SSI module	Transducer supply voltage
Signal cable	RS422, isolated
Clock	RS422, isolated
Baudrate	configurable:
	100 / 300 / 600kBaud (default: 300kBaud)
Signal voltage "0"	-5 ... 7V
Signal voltage "1"	13 ... 36V
Output stage	24V DC high side switch 1A
Ext. power supply	24V DC (18 ... 28.8V)
Status indicator	via LEDs at the front
Programming specifications	
Input data	4Byte
Output data	4Byte, 8Byte buffer in the module
Parameter data	4Byte
Diagnostic data	-
Dimensions and weight	
Dimensions (WxHxD)	25.4x76x78mm
Weight	100g

Chapter 4 FM 250 - Counter module

Overview

This chapter contains every information on the interfacing, configuration and the different counter modes of the counter module FM 250.

The following text describes:

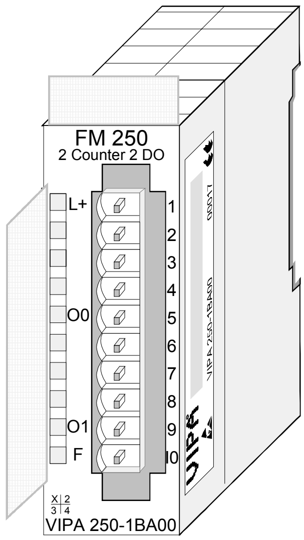
- Structure and deployment
- Counter modes
- Technical data

Contents

System overview

Properties

- two 32Bit channels / four 16Bit channels (depending on the mode)
- DC 24V supply voltage or via backplane bus
- free configurable DC 24V outputs (max.1A)
- Counter and compare registers are loaded by means of a control byte
- Standard up-down counter with a resolution of 32Bit or 16Bit
- Compare and auto-reload functions
- Different modes for encoder pulses
- Pulse-width measurements and frequency measurements



Note!
The following information is only applicable to counter modules with order no.: VIPA 250-1BA00 and a revision level 5 and higher.

Order data

Type	Order number	Description
FM 250	VIPA 250-1BA00	Counter module (2 counter 2 DO)

Structure

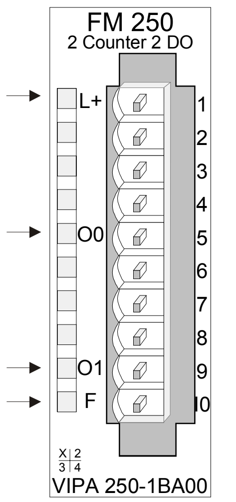
Functionality

The counter module accepts the signals from transducers connected to the module and processes these pulses in accordance with the selected mode of operation. The module has 2/4 channels with a data resolution of 32/16Bit each.

These modules provide 40 counter modes and two 24V outputs they are controlled in accordance with the selected mode.

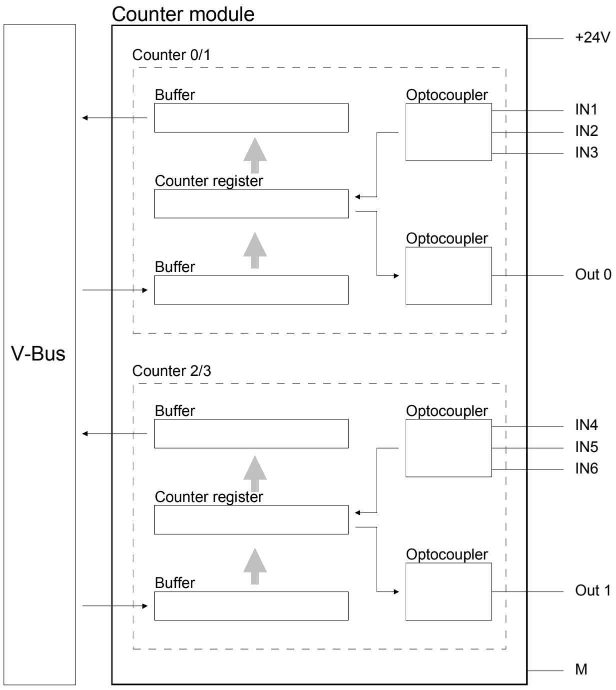
Status indicator pin assignment

LED	Description
L+	LED (yellow) Supply voltage available
O0	LED (green) Output counter 0
O1	LED (green) Output counter 1
F	LED (red) Error /overload

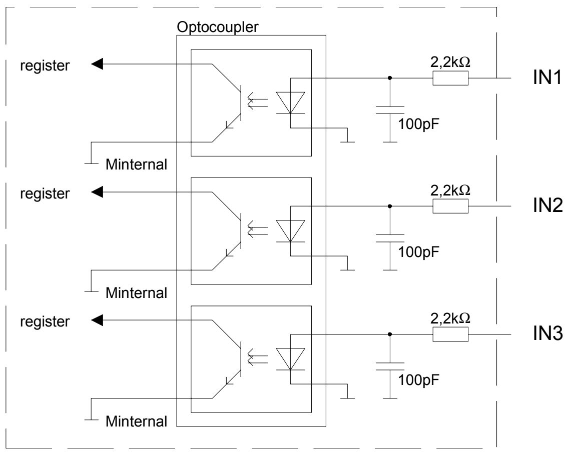


Pin	Assignment
1	Supply voltage +24V DC
2	IN1 input 1 counter 0/1
3	IN2 input 2 counter 0/1
4	IN3 input 3 counter 0/1
5	OUT0 output counter 0/1
6	IN4 input 4 counter 2/3
7	IN5 input 5 counter 2/3
8	IN6 input 6 counter 2/3
9	OUT1 output counter 2/3
10	Common of supply voltage

Block diagram



Input internal circuit



Deployment

Access to the counter module

The module has 2/4 channels with a resolution of 32/16Bit each. You may use parameters to specify the mode for each channel res. channel pair. The pin assignment for the channel depends upon the selected mode (see description of modes).

10 data bytes are required for the data input and output. Data output to a counter channel requires 10Byte, for example for defaults or for comparison values. In the latter case Byte 9 (control) is used to initiate a write operation into the required counter register. The respective values are transferred into the counter registers when they are toggled ($0 \rightarrow 1$).

The 10^{th} byte (status byte) controls the behavior of the counter during a restart of the next higher master module. You may set the counter level to remanent by means of a combination of Bits 0 and 1; i.e. the original counter level will not be reset when the next higher master module restarts.

The following combinations are possible:

Bit 0=1, Bit 1=0	counter value is remanent during restart
Bit 0=x, Bit 1=1	counter value is reset during restart (default)

You may check your settings at any time by reading Byte 10 of the output data.

Data sent to module

00h	DE0	Counter 0/1
01h	DE1	
02h	DE2	
03h	DE3	
04h	DE4	Counter 2/3
05h	DE5	
06h	DE6	
07h	DE7	
08h	Control	
09h	Status	

Data received from module

00h	DA0	Counter 0/1
01h	DA1	
02h	DA2	
03h	DA3	
04h	DA4	Counter 2/3
05h	DA5	
06h	DA6	
07h	DA7	
08h		
09h	Status	

Configuration parameters

The configuration parameters consist of 2Byte. You use these bytes to define the operating mode of each channel by means of a mode number. This chapter contains a detailed description of the different modes further below. The different combinations of the various modes are available from the table on the next page. The procedure for the transfer of parameter bytes is available from the description for the System 200V bus coupler or the master system.

	7							0	bit no.
Parameter byte 1									Mode counter 0/1

	7							0	bit no.
Parameter byte 2									Mode counter 2/3

Summary of counter modes and interfacing

Mode	may be combined	Function	IN1	IN2	IN3	IN4	IN5	IN6	OUT0	OUT1	Auto Re-load	Com-pare Load
			Counter 0/1			Counter 2/3						
0	yes	32bit counter	RES	CLK	DIR	RST	CLK	DIR	=0	=0	no	=0
1	yes	Encoder 1 edge	RES	A	B	RST	A	B	=0	=0	no	=0
3	yes	Encoder 2 edges	RES	A	B	RST	A	B	=0	=0	no	=0
5	yes	Encoder 4 edges	RES	A	B	RST	A	B	=0	=0	no	=0
			Counter 1 counter 0			Counter 3 counter 2						
8	yes	2x16bit counter up/up	-	CLK	CLK	-	CLK	CLK	-	-	no	no
9	yes	2x16bit counter down/up	-	CLK	CLK	-	CLK	CLK	-	-	no	no
10	yes	2x16bit counter up/down	-	CLK	CLK	-	CLK	CLK	-	-	no	no
11	yes	2x16bit counter down/down	-	CLK	CLK	-	CLK	CLK	-	-	no	no
			Counter 0/1			Counter 2/3						
12	yes	32bit counter up + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	no	yes
13	yes	32bit counter down + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	no	yes
14	yes	32bit counter up + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	yes	yes
15	yes	32bit counter down + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	yes	yes
			Combination of counter 0 ... 3									
16	no	Frequency measurement	RES	CLK	Start	Stop	-	-	Meas. active	Meas. compl.	no	yes
17	no	Period measurement	RES	CLK	Start	Stop	-	-	Meas. active	Meas. compl.	no	yes
18	no	Frequency measurement with gate output	RES	CLK	Start	Stop	-	-	Meas. gate	$\overline{\text{Meas. gate}}$	no	yes
19	no	Period measurement with gate output	RES	CLK	Start	Stop	-	-	Meas. gate	$\overline{\text{Meas. gate}}$	no	yes
			Counter 0/1			Counter 2/3						
6	yes	Pulse low, 50kHz with Direction Input	RES	Pulse	DIR	RES	Pulse	DIR	-	-		
20	yes	Pulse low, prog. time base with Direction Input	RES	Pulse	DIR	RES	Pulse	DIR	-	-		
21	yes	Pulse low, up, prog. time base with Gate	RES	Pulse	Gate	RES	Pulse	Gate	-	-		
22	yes	Pulse high, up, prog. time base with Gate	RES	Pulse	Gate	RES	Pulse	Gate	-	-		
			Counter 0/1			Counter 2/3						
23	yes	One Shot, up, Set	RES	CLK	Gate	RES	CLK	Gate	1 if active	1 if active	no	yes
24	yes	One Shot, down, Set	RES	CLK	Gate	RES	CLK	Gate	1 if active	1 if active	no	yes
25	yes	One Shot, up, Reset	RES	CLK	Gate	RES	CLK	Gate	0 if active	0 if active	no	yes
26	yes	One Shot, down, Reset	RES	CLK	Gate	RES	CLK	Gate	0 if active	0 if active	no	yes
			Counter 0/1			Counter 2/3						
27	yes	32 bit counter	Gate/R$^{\uparrow}$	CLK	DIR	Gate/R$^{\uparrow}$	CLK	DIR	=0	=0	no	=0
28	yes	Encoder 1 edge	Gate/R$^{\uparrow}$	A	B	Gate/R$^{\uparrow}$	A	B	=0	=0	no	=0
29	yes	Encoder 2 edges	Gate/R$^{\uparrow}$	A	B	Gate/R$^{\uparrow}$	A	B	=0	=0	no	=0
30	yes	Encoder 4 edges	Gate/R$^{\uparrow}$	A	B	Gate/R$^{\uparrow}$	A	B	=0	=0	no	=0

continued ...

... continue

Mode	may be com-bined	Function	IN1	IN2	IN3	IN4	IN5	IN6	OUT0	OUT1	Auto Re-load	Com-pare Load
			Counter 0/1			Counter 2/3						
31	yes	32Bit counter up+Gate	RES$^{\uparrow}$	CLK	Gate	RES$^{\uparrow}$	CLK	Gate	=comp	=comp	no	yes
32	yes	32Bit counter down+Gate	RES$^{\uparrow}$	CLK	Gate	RES$^{\uparrow}$	CLK	Gate	=comp	=comp	no	yes
33	yes	32Bit counter up+Gate	RES$^{\uparrow}$	CLK	Gate	RES$^{\uparrow}$	CLK	Gate	=comp	=comp	yes	yes
34	yes	32Bit counter down+Gate	RES$^{\uparrow}$	CLK	Gate	RES$^{\uparrow}$	CLK	Gate	=comp	=comp	yes	yes
			Counter 0/1			Counter 2/3						
35	yes	32Bit counter	Gate	CLK	DIR	Gate	CLK	DIR	=0	=0	no	=0
36	yes	Encoder 1 edge	Gate	A	B	Gate	A	B	=0	=0	no	=0
37	yes	Encoder 2 edges	Gate	A	B	Gate	A	B	=0	=0	no	=0
38	yes	Encoder 4 edges	Gate	A	B	Gate	A	B	=0	=0	no	=0
			Counter 0/1			Counter 2/3						
39	yes	32Bit counter up+Gate	RES$^{\uparrow}$	$\overline{\text{Gate}}$		RES$^{\uparrow}$	$\overline{\text{Gate}}$		-	-	-	-
40	yes	32Bit counter down+Gate	RES$^{\uparrow}$	$\overline{\text{Gate}}$		RES$^{\uparrow}$	$\overline{\text{Gate}}$		-	-	-	-
41	yes	32Bit counter up+Gate	RES$^{\uparrow}$	Gate		RES$^{\uparrow}$	Gate		-	-	-	-
42	yes	32Bit counter down+Gate	RES$^{\uparrow}$	Gate		RES$^{\uparrow}$	Gate		-	-	-	-

Due to technical advances the revision level and the functionality of the counter module was continuously expanded. Below follows a list that allocates the different modes to the revision level:

Mode 0-5	revision level 3	Mode 27-30	revision level 8/9
Mode 0-17	revision level 4	Mode 31-38	revision level 10
Mode 0-19	revision level 5	Mode 39-42	revision level 11
Mode 6, 20-26	revision level 6/7		

Terminology:

RES RESET signal that has to be LOW during the measuring process. A HIGH level (level triggered) erases one or both counters, depending on the selected mode.

RES$^{\uparrow}$ The counter is reseted by the rising edge of this signal (edge triggered).

CLK The clock signal from the connected transducer.

Start or Stop A HIGH level starts or stops the counter. When the start level is active, the counter will start with the next CLK pulse that corresponds to the selected mode.

DIR In mode 0 the level of the DIR signal determines the direction of the counting process.

LOW level: count up

HIGH level: count down

Auto Reload

The Auto Reload function transfers a user-defined value into the counter when the counter reaches the number contained in the compare register.

Compare Load

You may use the compare function to specify an stop value for the counter. Depending on the selected mode an output is activated or the counter is re-started when it reaches this value.

Gate

Gate signal enabling the counter.

Gate/R↑

The counter is reseted by the rising edge of this signal. As long as the signal is at "1", the counter is released.

(Gate = level triggered; R↑ = edge triggered)

Measurement gate

Status indicator of the counter activity - is set to a HIGH level after the 1st CLK signal and LOW level after the last CLK signal (mode 18 ... 19).

Pulse

The pulse width of the introduced signal is determined by means of the internal time base.

Fref

Reference or clock frequency that is set permanently to 50kHz in mode 6.

The clock frequency "Fref" for counter modes 20-22, 39-42 is programmable:

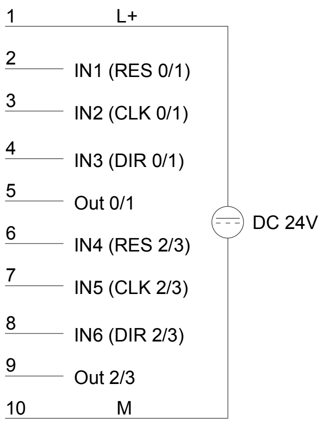
Parameter	Fref
0	10MHz
1	1MHz
2	100kHz
3	10kHz

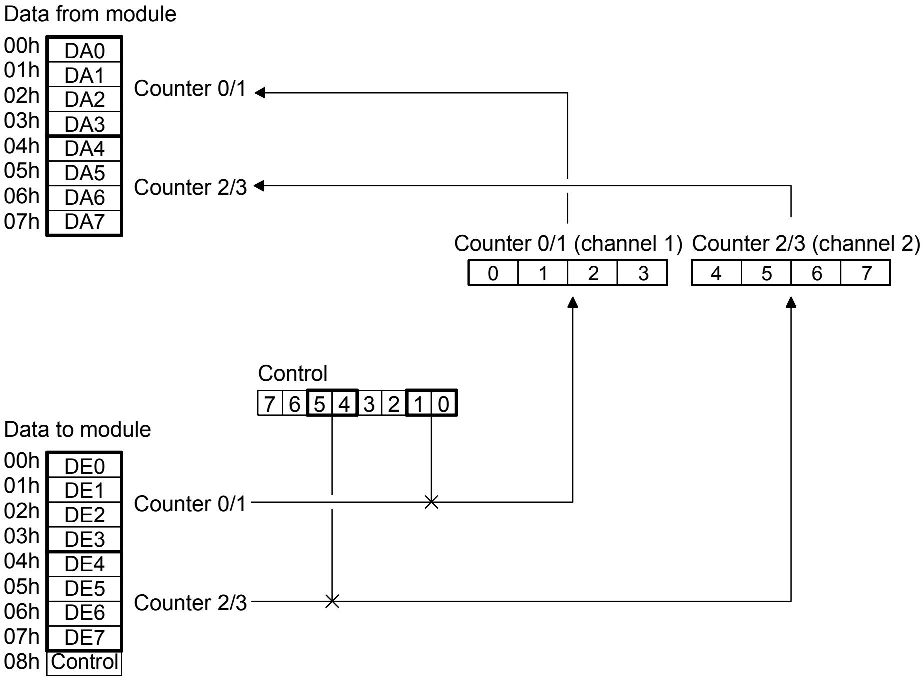
Counter modes

Mode 0
32Bit counter

2x 32Bit Counter. You determine the direction by means of the DIR input (IN3 or IN6). Every rising or falling edge of the input clock signal increments or decrements the counter. During the counting process the RES signal must be at a LOW level. If the RES signal is at a HIGH level, the counter is cleared. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter should continue counting. If the counter stops at zero, the output remains active.

Pin assignment
access to counter

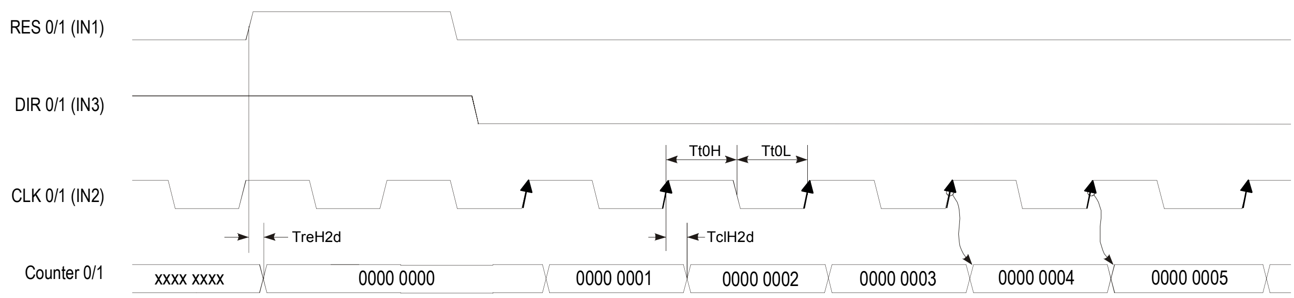




Up counter

In mode 0, a LOW level at the DIR input configures the counter for counting up.

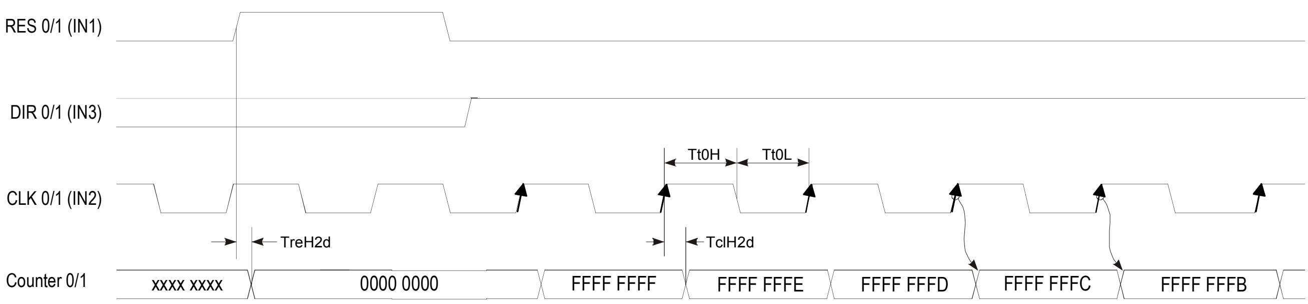
Timing diagram of the counter 0/1 example:



Down counter

In mode 0, a HIGH level at the DIR input configures the counter for counting down.

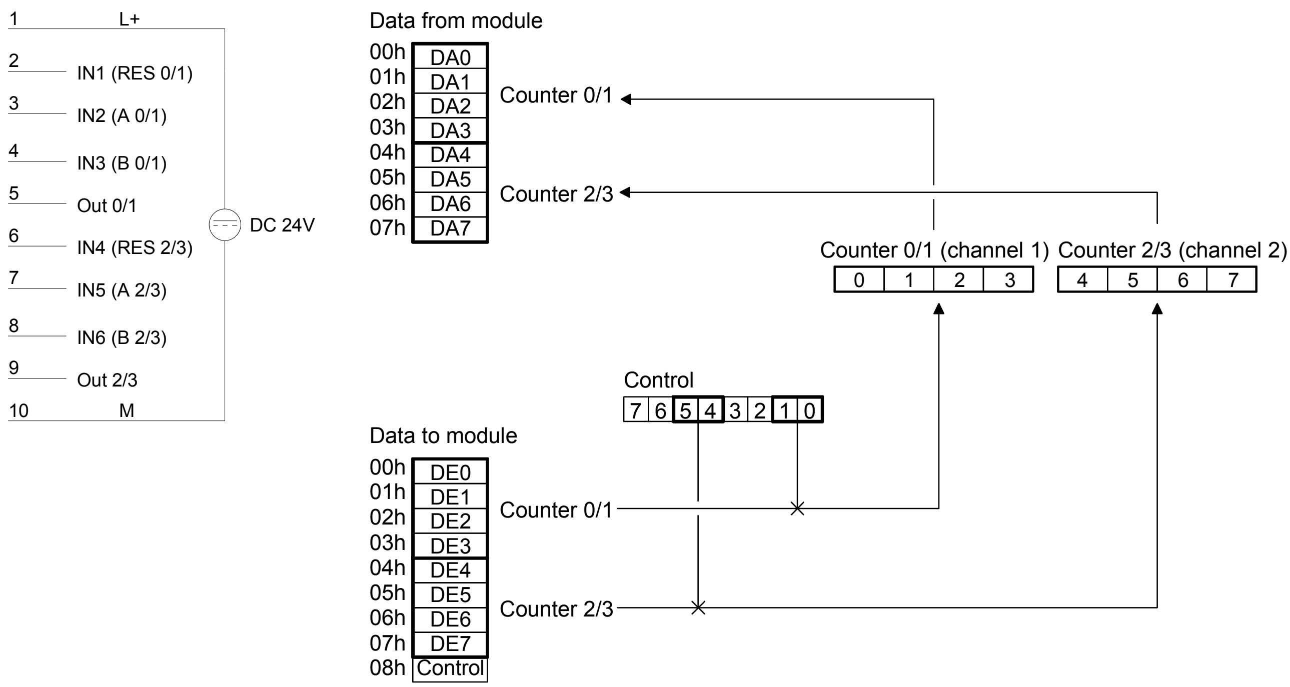
Timing diagram of the counter 0/1 example:



Mode 1 Encoder 1 edge

In mode 1 you may configure an encoder for one of the channels. Depending on the direction of rotation this encoder will increment or decrement the internal counter with every falling edge. The RES input has to be at a LOW level during the counting process. A HIGH level clears the counter. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero the output remains active.

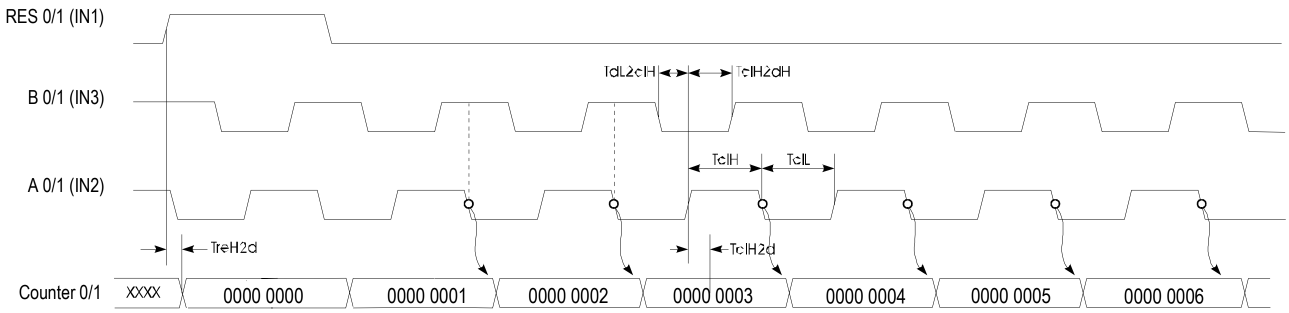
Pin assignment access to counter



Up counter

Every falling edge of the signal at input A increments the counter if input B is at HIGH level at this moment.

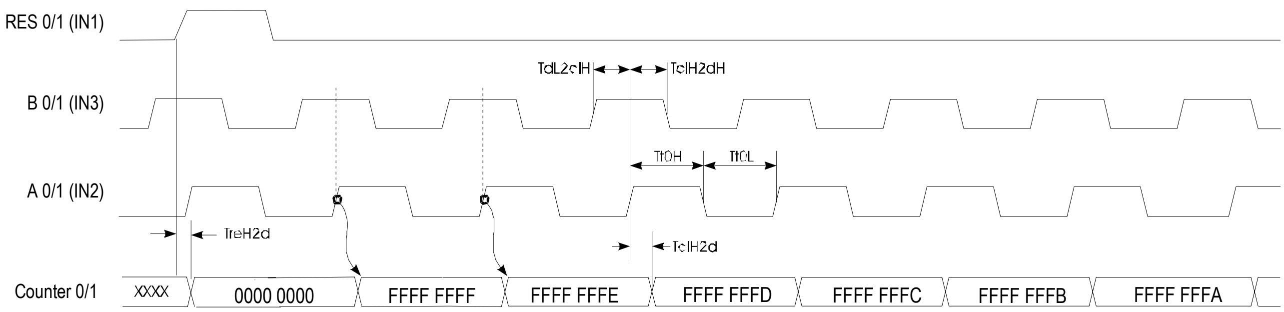
Timing diagram for the counter 0/1 example:



Down counter

Every rising edge of the signal at input A decrements the internal counter if input B is at HIGH level at this moment.

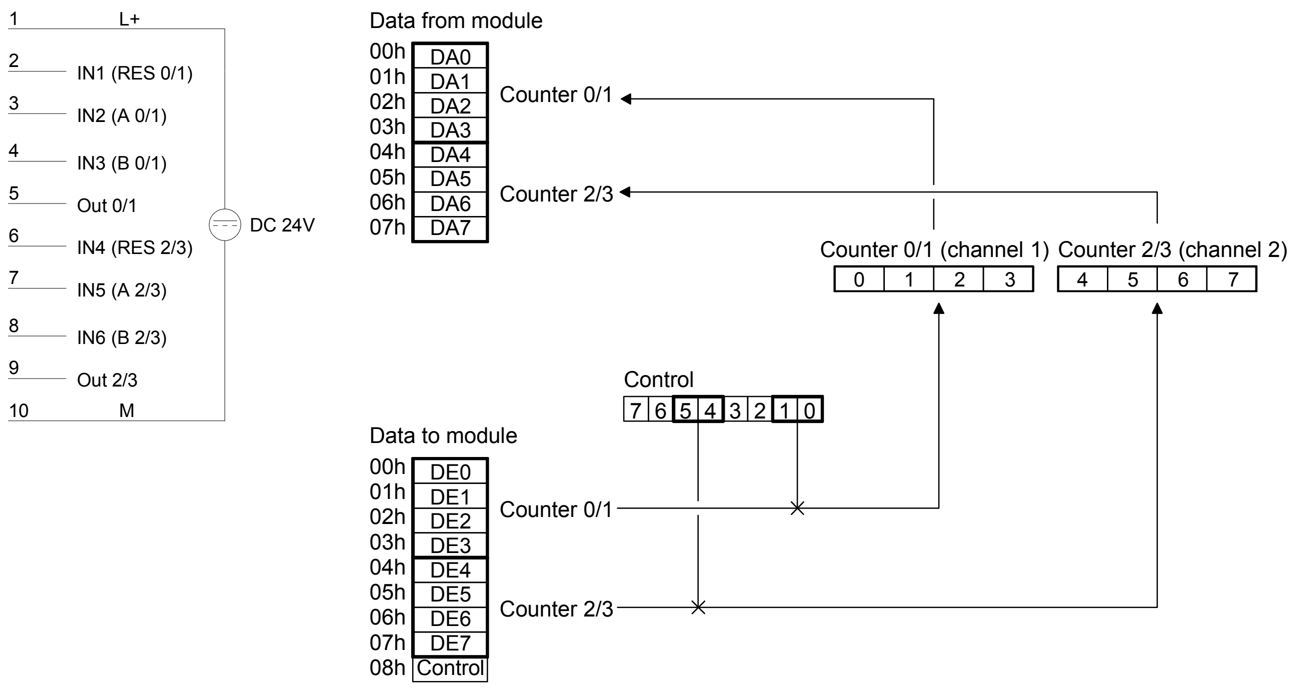
Timing diagram for the counter 0/1 example:



Mode 3
Encoder 2 edges

Every rising or falling edge of the signal at input A changes the counter by 1. The direction of the count depends on the level of the signal applied to input B. RES has to be at a LOW level during the counting process. A HIGH level clears the counter. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero the output remains active.

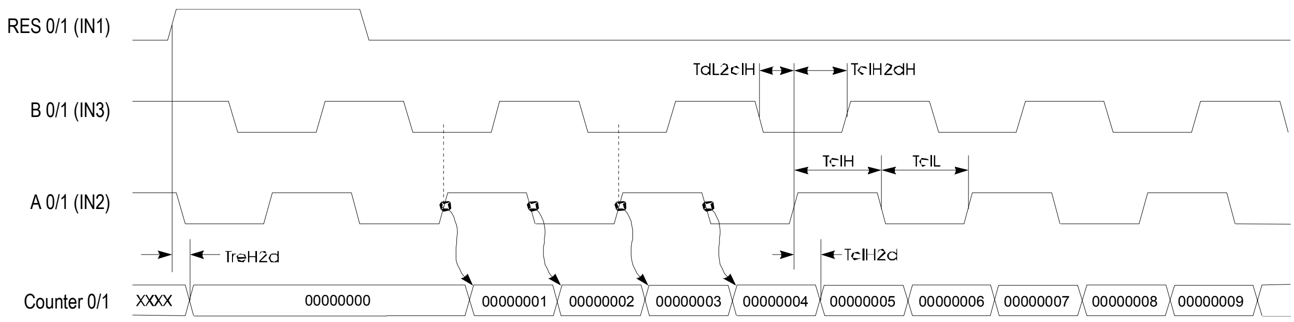
Pin assignment
access to counter



Up counter

The counter is incremented by the rising edge of signal A if input B is at a LOW level or by the falling edge of input A when input B is at a HIGH level.

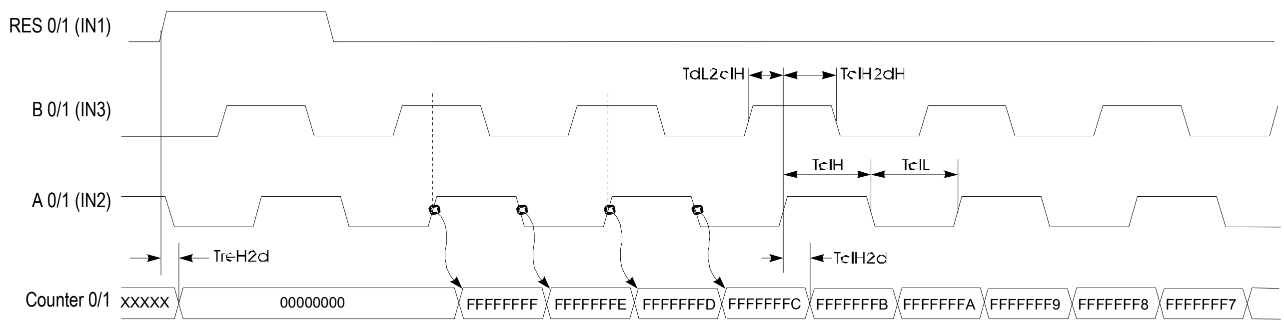
Timing diagram for the counter 0/1 example:



Down-counter

The counter is decremented by the rising edge of signal A if input B is at a HIGH level or by the falling edge of input A when input B is at a LOW level.

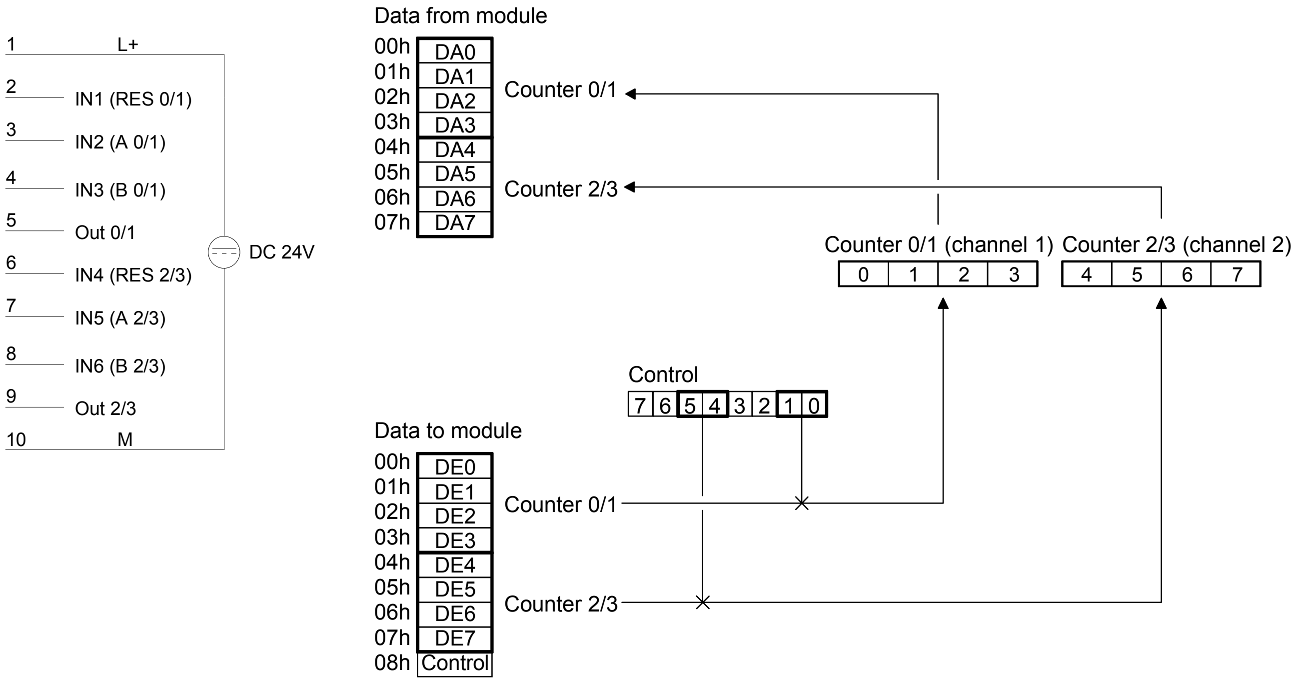
Timing diagram for the counter 0/1 example:



Mode 5
Encoder 4 edges

Every rising or falling edge at inputs A or B increments or decrements the counter. The direction depends on the level applied to the other input (B or A). RES has to be at a LOW level during the counting process. A HIGH level clears the counter. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero, the output remains active.

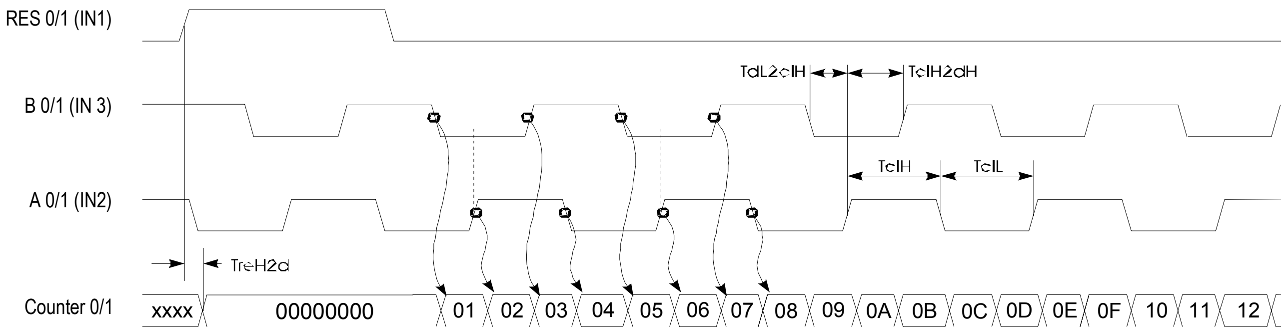
Pin assignment access to counter



Up counter

The counter is incremented when a rising edge is applied to B while input A is at a HIGH level or if a falling edge is applied to B when input A is at a LOW level. Alternatively it is also incremented when a rising edge is applied to A when input B is at a LOW level or by a falling edge at A when input B is at a HIGH level.

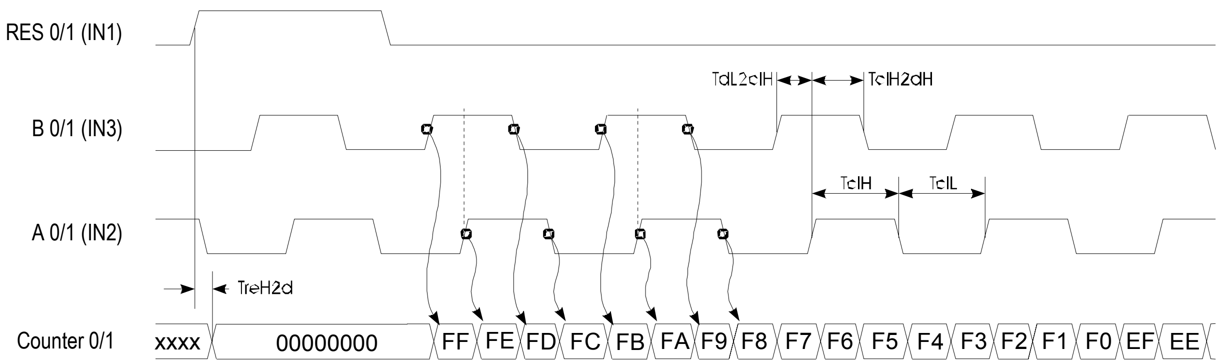
Timing diagram for the counter 0/1 example:



Down counter

The counter is decremented when a rising edge is applied to B while input A is at a LOW level or if a falling edge is applied to B when input A is at a HIGH level. Alternatively it is also decremented when a rising edge is applied to A when input B is at a HIGH level or by a falling edge at input A when input B is at a LOW level.

Timing diagram for counter 0/1 example:



Mode 8 ... 11 two input counter function

In this mode each channel provides 2 counters of 16Bit each. The rising edge of the input clock CLK x increments or decrements the respective counter. In this mode each counter can also be preset to a certain value by means of a control bit. Outputs are not available. A RESET is also not available. The following combinations are possible for every channel:

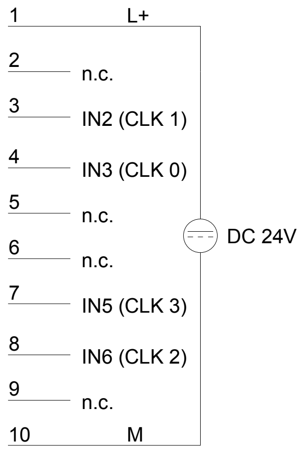
Mode 8 - counter 0/1 up, counter 2/3 up

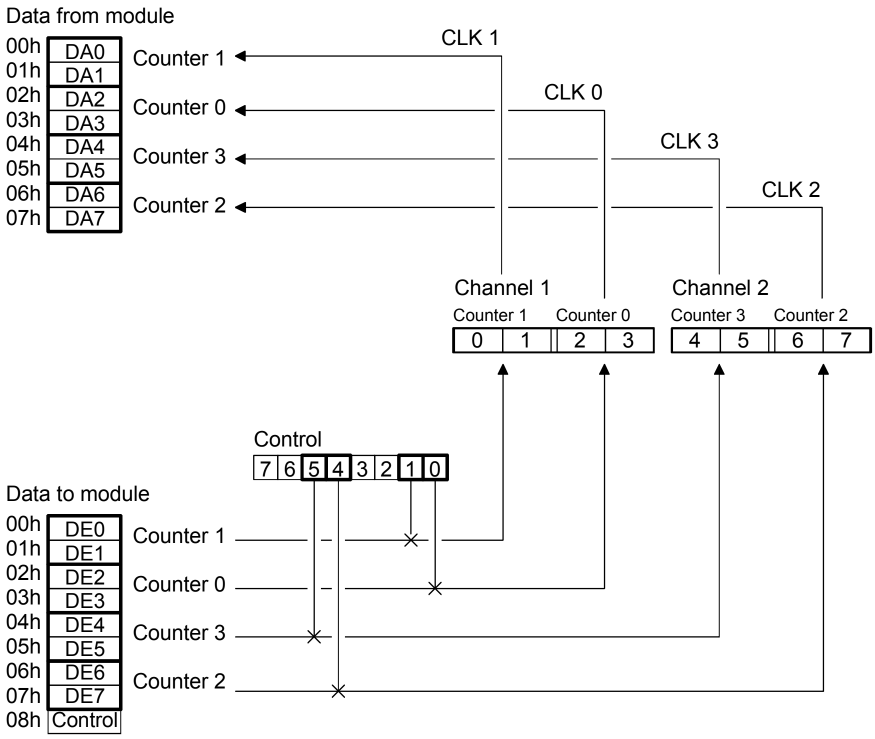
Mode 9 - counter 0/1 down, counter 2/3 up

Mode 10 - counter 0/1 up, counter 2/3 down

Mode 11 - counter 0/1 down, counter 2/3 down

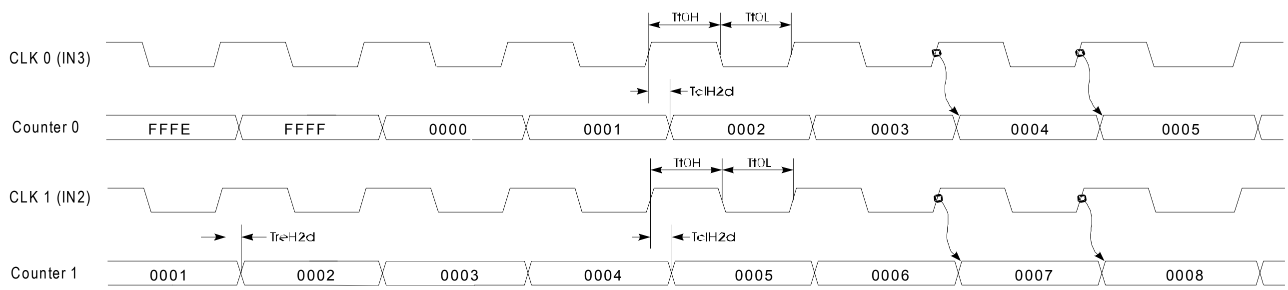
Pin assignment access to counter





Timing diagram

Below follows a timing diagram depicting an example of counter 0 and counter 1 in mode 8:



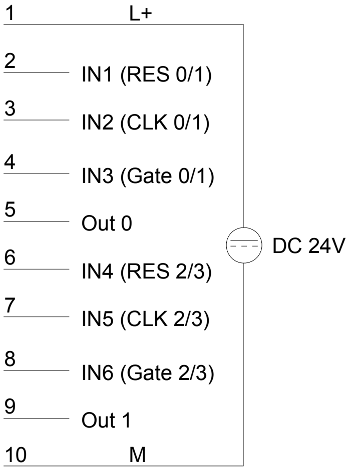
Mode 12 and 13 32bit counter with gate

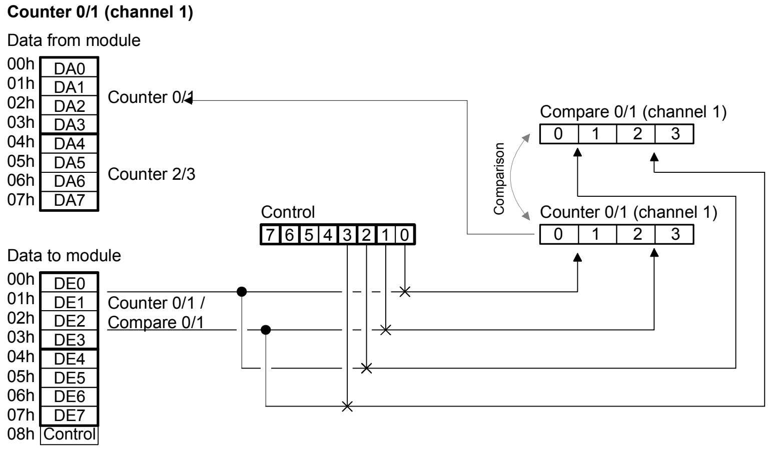
In mode 12 and mode 13 you can implement a 32Bit counter that is controlled by a gating signal (Gate). The direction of counting depends on the selected mode. Every rising edge of the input signal increments or decrements the counter provided that the GATE signal is at HIGH level. RES has to be LOW during the counting process. A HIGH level clears the counter. When the counter reaches the value that was previously loaded into the compare register, output OUT is set active for a minimum period of 100ms while the counter continues counting.

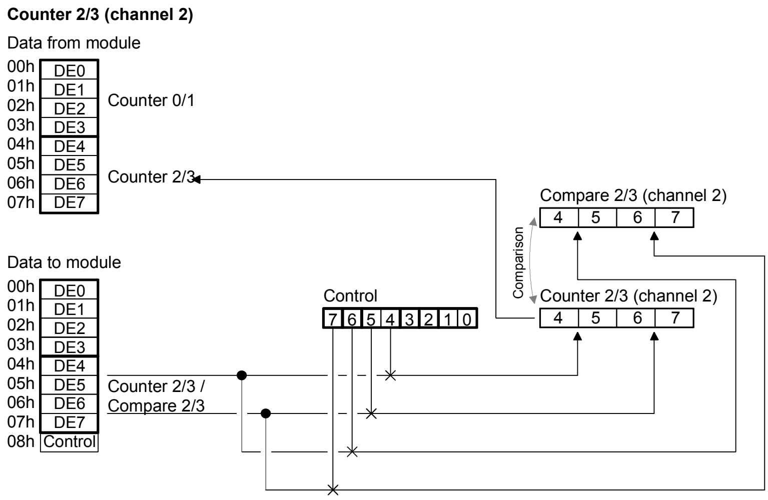
Mode 12 - 32Bit counter up + gate with compare

Mode 13 - 32Bit counter down + gate with compare

Pin assignment access to counter

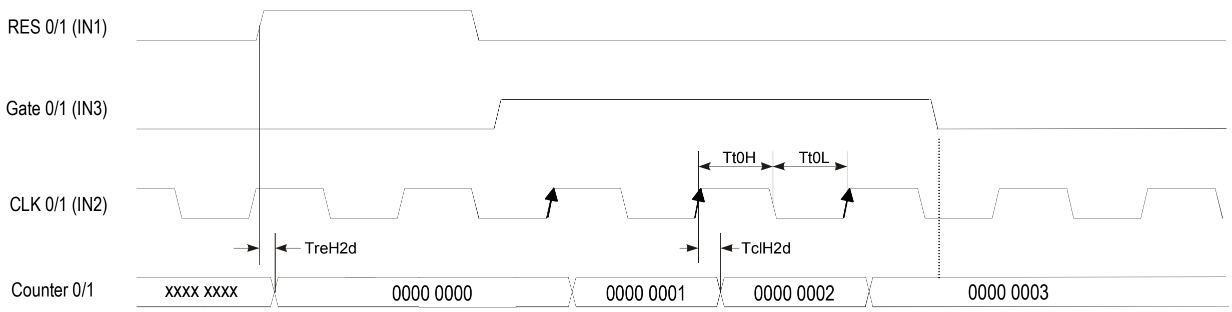






Timing diagram

Below follows an example of a timing diagram of counter 0/1 in mode 12:



Mode 14 and 15 32Bit counter with gate and auto reload

Modes 14 and 15 operate in the same manner as mode 12 and 13 with the addition of an Auto Reload function. The "Auto Reload" is used to define a value in the load register that is used to preset the counter automatically when it reaches the compare value.

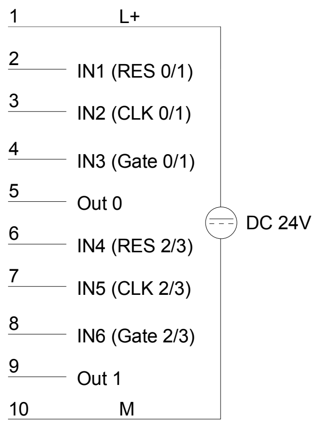
A HIGH pulse applied to RES clears the counter to 0000 0000. A HIGH level applied to GATE enables the counter so that is incremented/decremented by every rising edge of the CLK signal. As long as GATE is HIGH, the counter will count every rising edge of the signal applied to CLK until the count is one less than the value entered into COMPARE. The next pulse overwrites the counter with the value contained in the load register. This process continues until GATE is set to a LOW level. When an auto reload occurs, the status of the respective output changes.

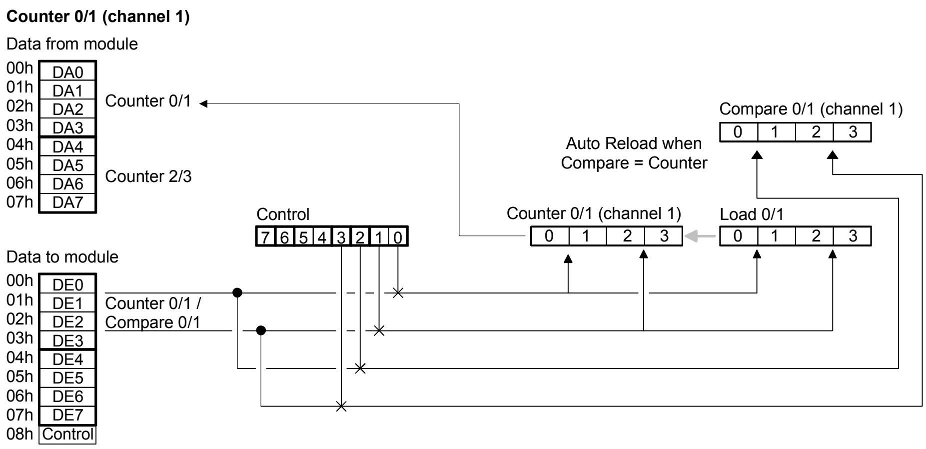
The RES signal only resets the counter but not the outputs.

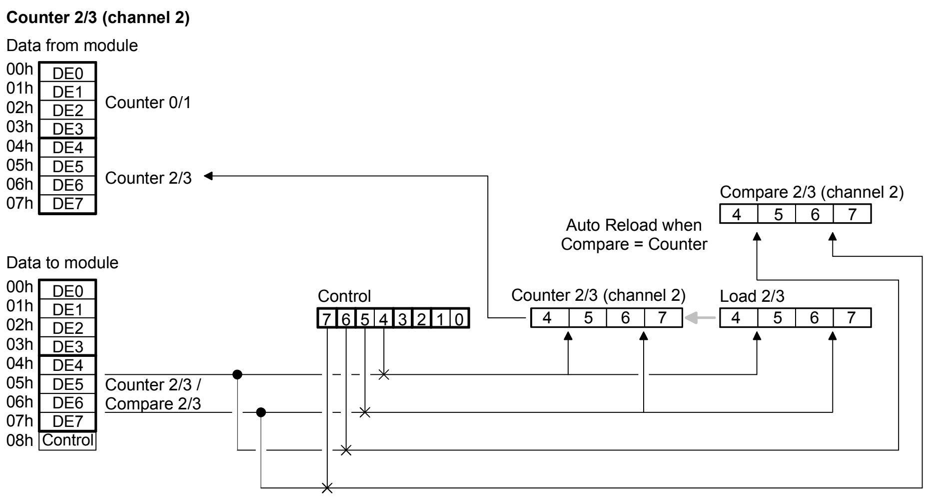
Mode 14 - 32Bit counter up + gate with compare and auto reload

Mode 15 - 32Bit counter down + gate with compare and auto reload

Pin assignment access to counter







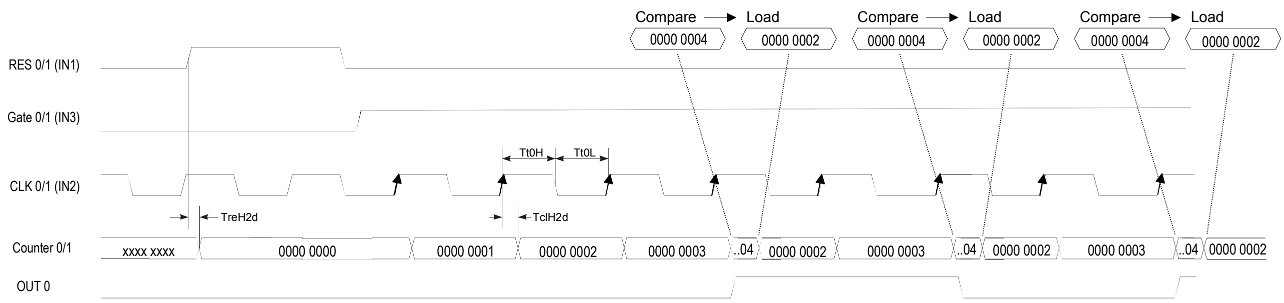
Example

This example is intended to explain the operation of the counters in mode 14 and 15.

A HIGH pulse applied to RES clears the counter to 0000 0000. A HIGH level applied to GATE enables the counter. As long as GATE is HIGH the counter will count every rising edge of the signal applied to CLK until the count is one less than the value entered into COMPARE. In this example the counter counts to 0000 0004 followed immediately by an auto reload, i.e. the counter is preset to the contents of the load register (in this case 0000 0002). The state of output OUT 0 changes every time an auto reload is executed.

In this example the counter counts from 0000 0002 to 0000 0004 as long as the GATE input is at a HIGH level.

Every load operation changes the status of output OUT 0.



Mode 16 frequency measurement

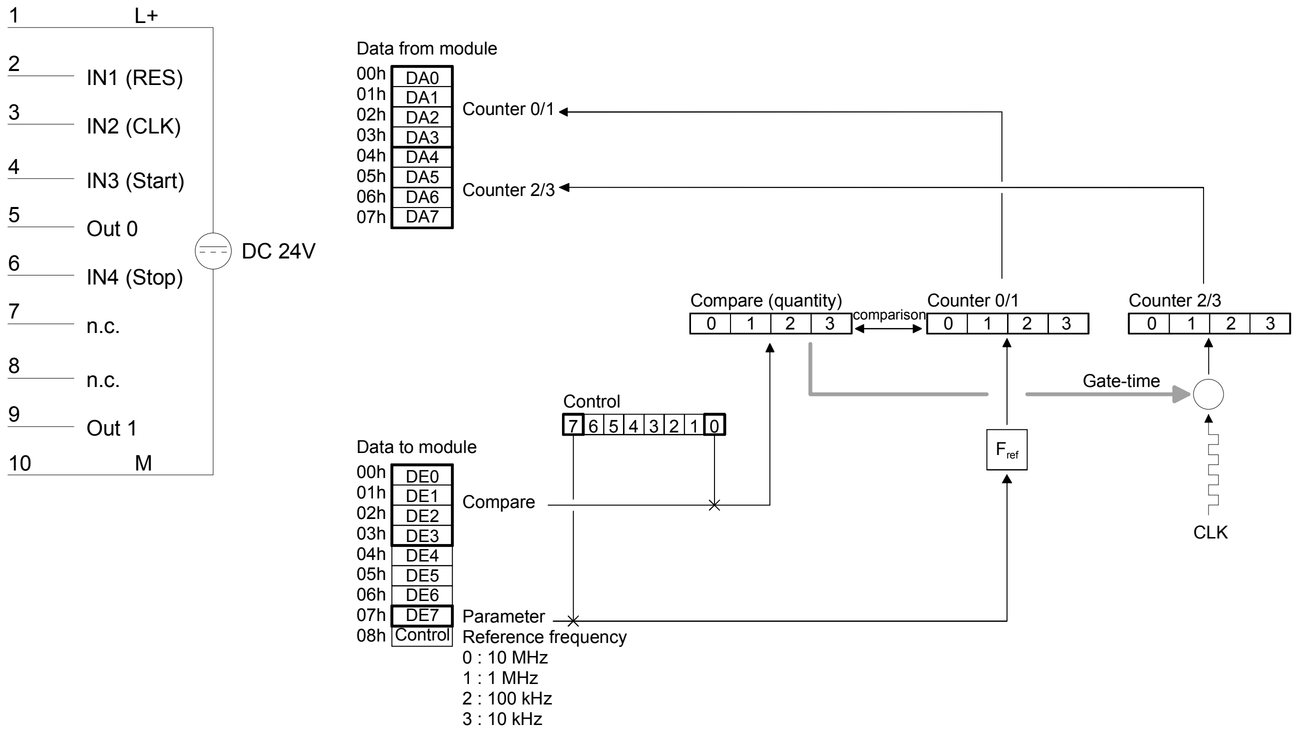
In this mode it is possible to determine the frequency of the signal that is applied to the CLK input. Counter 0/1 is provided with a reference signal by means of DE7 and a gate time that is controlled indirectly by the value n to determine the duration for which counter 2/3 is enabled. The value of n has a range from 1 to 2^{32}-1 and it is loaded into the COMPARE register.

When enabled by the rising edge of the signal applied to Start, counter 0/1 counts reference pulses of the reference clock generator from the first rising edge of the CLK signal.

During this time counter 2/3 counts every rising edge of the CLK signal. Both counters are stopped when counter 0/1 reaches the COMPARE value or when a HIGH level is applied to Stop. You may calculate the frequency by means of the formula shown below.

This mode can not be combined with other modes!

Pin assignment access to counter



Frequency calculation

When the measurement has been completed you may calculate the frequency as follows:

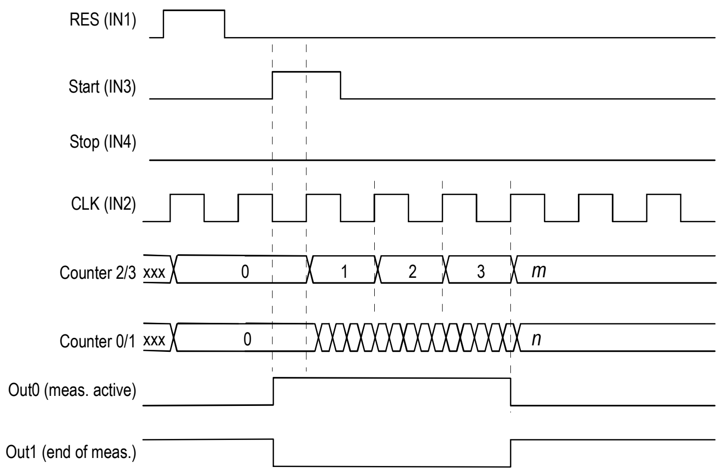
$$Frequency = \frac{Fref \cdot m}{n}$$

where *Fref*: reference frequency (supplied in DE7 with control bit 7)

m: counter 2/3 contents (number of CLK pulses)

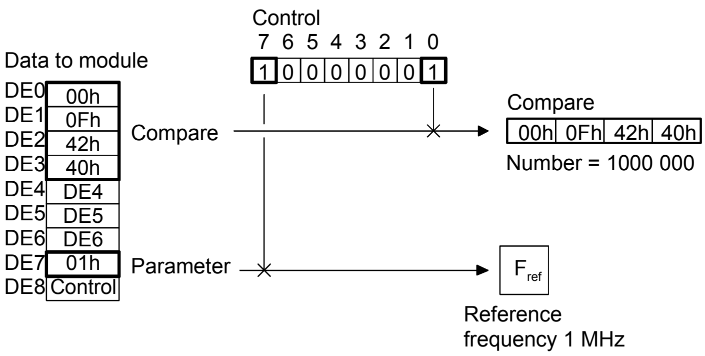
n: number of reference frequency pulses in counter 0/1 (equal to COMPARE, if the operation was not terminated prematurely by means of Stop)

Timing diagram



Example

Quantity = 1 000 000 pulses
Reference frequency = 1MHz



Using a frequency of 1MHz and 1 000 000 pulses will return 1Hz, i.e. when the measurement is completed, counter 2/3 contains the frequency directly - no conversion is required.

Note!
Counter 2/3 will indicate the exact frequency if you choose *Fref* and *n* so that the formula returns 1Hz precisely.

Mode 17 period measurement

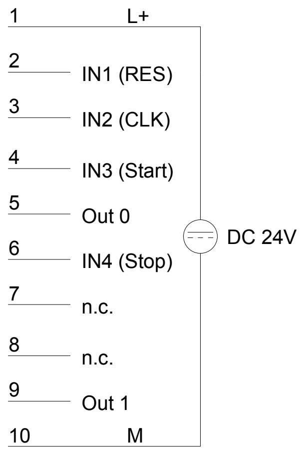
This mode is used to determine the average period of n measuring intervals of a signal that is connected to the CLK input. For this purpose you supply a reference clock to counter 2/3 by means of DE7 and indirectly a gate time defined by the value of n for which counter 2/3 is enabled. The value of n has a range from 1 to 2^{32}-1 and it is loaded into the COMPARE register.

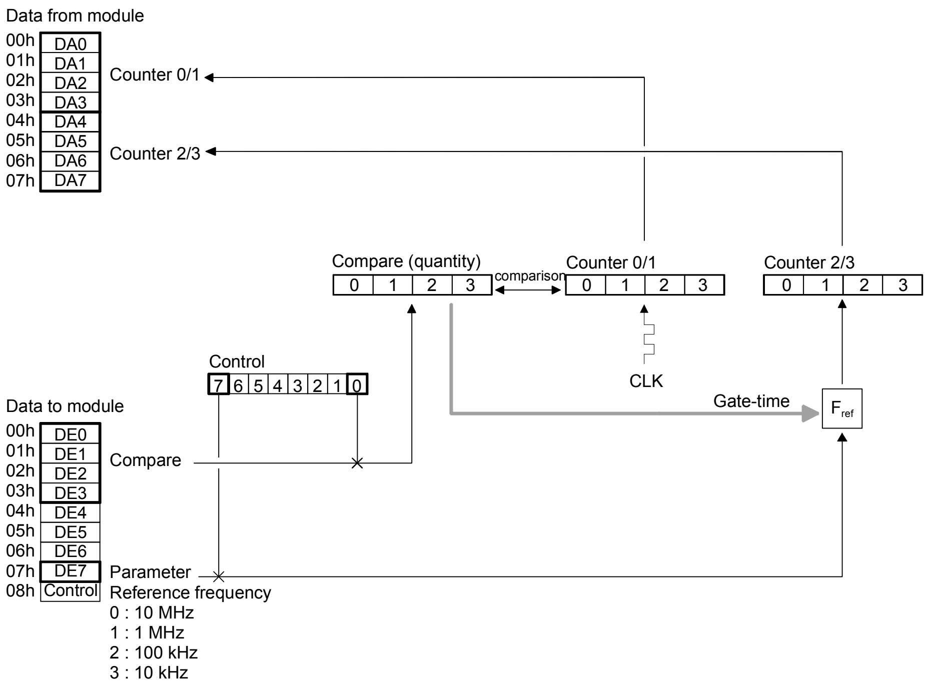
The measurement period begins when a rising edge is applied to Start. During this period counter 2/3 counts reference pulses from the reference clock generator starting with the first rising edge of the CLK signal.

In the mean time counter 0/1 counts every rising edge of the CLK signal. Both counters are stopped when the count in counter 0/1 reaches the Compare value or when Stop is set to a HIGH level. You may then calculate the average period by means of the formula shown below.

This mode can not be combined with other modes!

Pin assignment access to counter





Period calculation

When the measurement has been completed, you may calculate the period as follows:

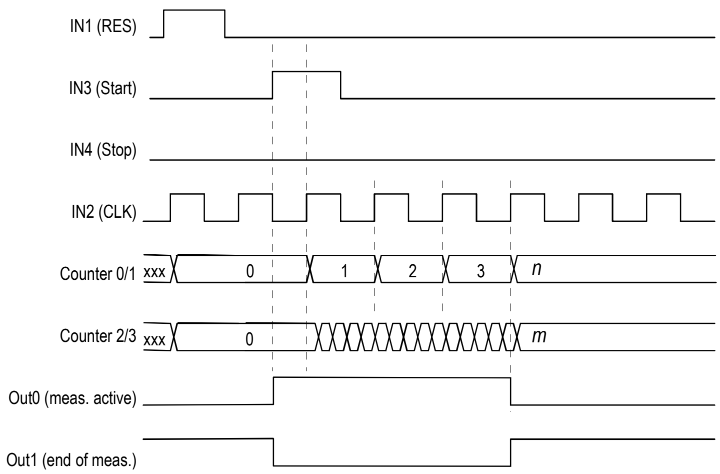
$$Period = \frac{m}{Fref \cdot n}$$

where *Fref*: reference frequency (supplied in DE7 with control bit 7)

m: contents of counter 2/3 (counts reference clock pulses)

n: number of CLK pulses in counter 0/1 (corresponds to COMPARE, provided it was not terminated prematurely by Stop)

Timing diagram:

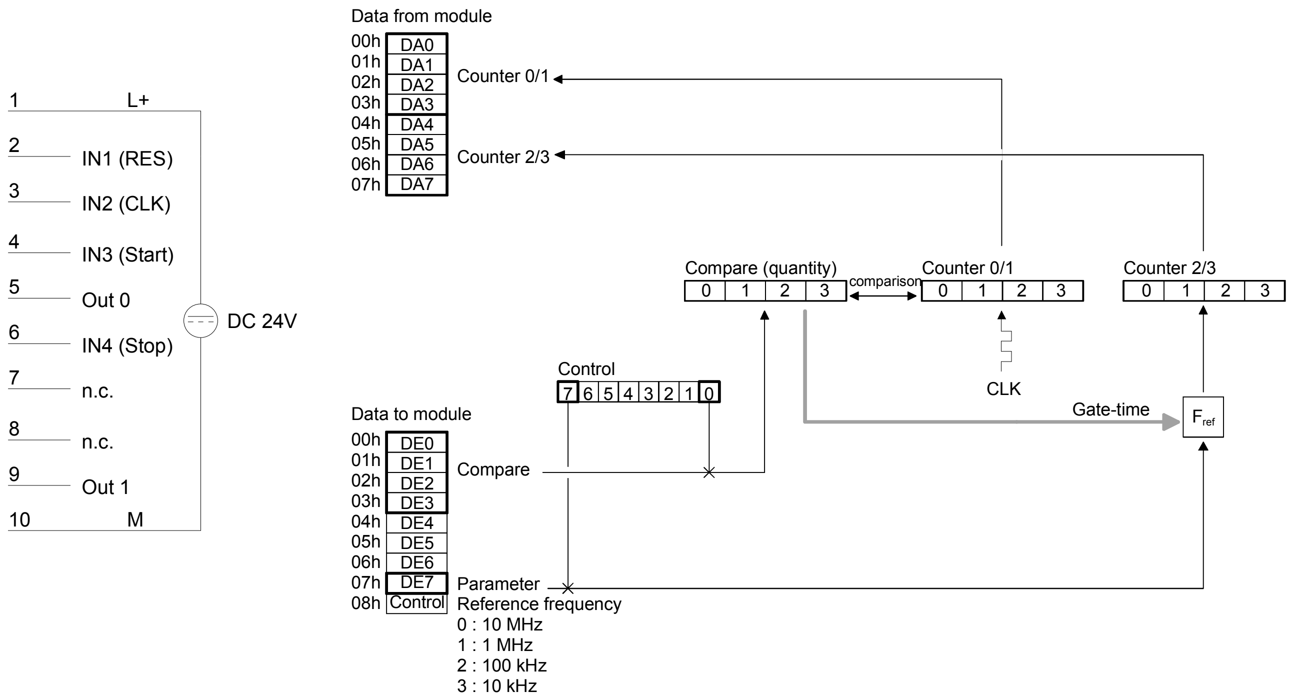


Mode 18 frequency measurement with gate output

The operation of mode 18 is similar to mode 16. The only difference is the manner in which OUT 0 and OUT 1 are controlled. In this case OUT 0 is only activated when the counting operation starts and it is deactivated when counting ends, i.e. OUT 0 provides an indication of the internal gate. OUT 1 provides the inverted status of the gate.

This mode can not be combined with other modes!

Pin assignment access to counter



Frequency calculation

When the measurement has been completed, you may calculate the frequency as follows:

$$Frequency = \frac{Fref \cdot m}{n}$$

where *Fref*: reference frequency (supplied in DE7 with control bit 7)

m: contents of counter 2/3 (CLK pulse count)

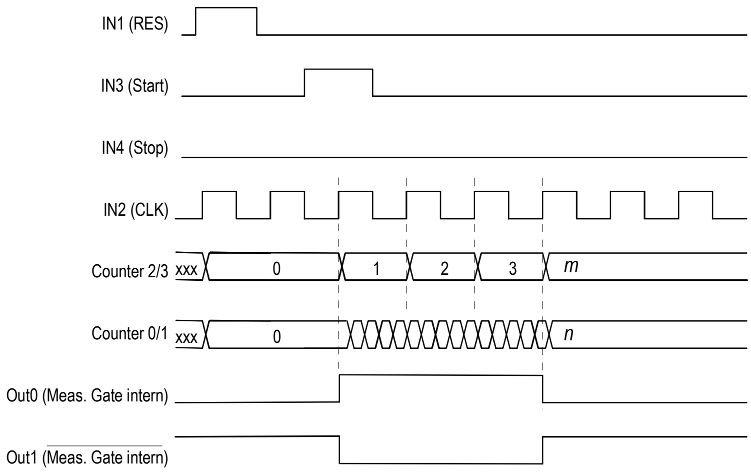
n: number of pulses of the reference frequency in counter 0/1 (corresponds to COMPARE provided it was not terminated prematurely by Stop)

Note!

Counter 2/3 will indicate the exact frequency if you choose *Fref* and *n* so that the formula returns 1Hz precisely.

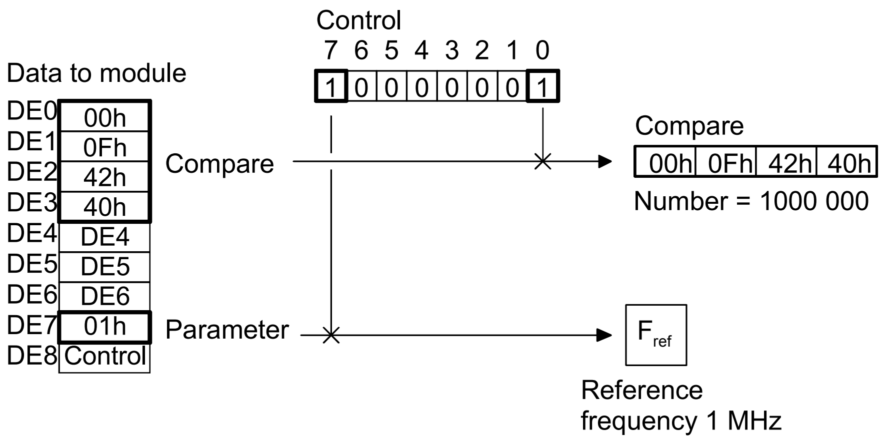
For example when the applied frequency is 1MHz and the number of pulses is 1 000 000 the result will be 1Hz, i.e. counter 2/3 contains the precise frequency after the measurement - this does not require further conversion.

Timing diagram:



Example

Pulse count = 1 000 000
Reference frequency = 1MHz

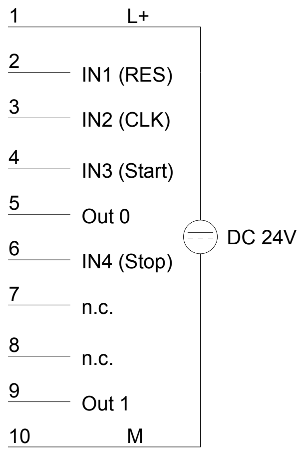


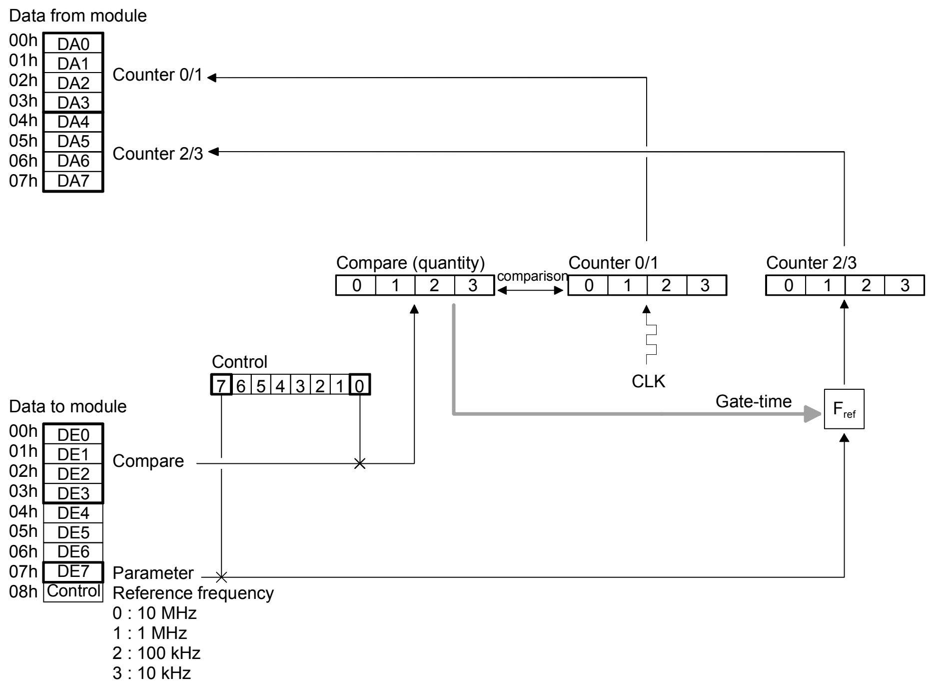
Mode 19 period measurement with gate output

The operation of mode 19 is identical to mode 17. The only difference is the manner in which OUT 0 and OUT 1 are controlled. Other than for mode 17, OUT 0 is only activated when the counting operation starts and it is deactivated when counting ends, i.e. OUT 0 provides an indication of the internal gate. OUT 1 provides the inverted status of the gate.

This mode can not be combined with other modes!

Pin assignment access to counter





Period calculation

When the measurement has been completed you may calculate the mean period as follows:

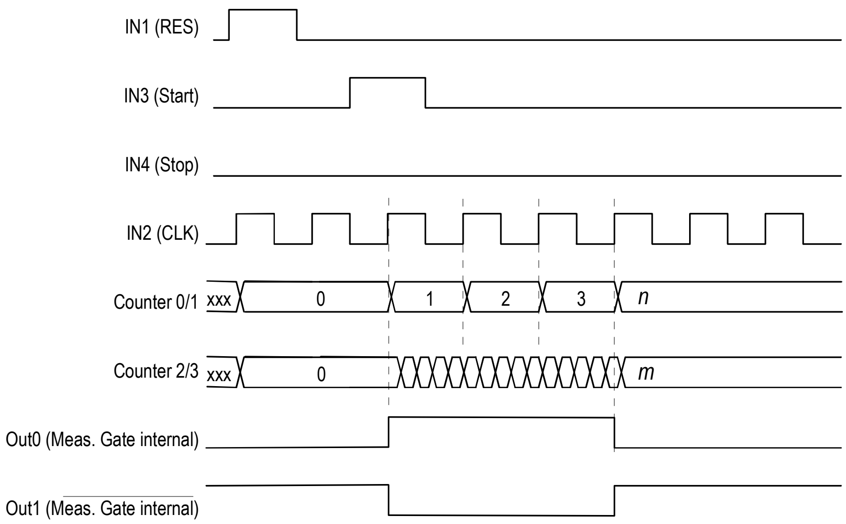
$$Period = \frac{m}{Fref \cdot n}$$

where *Fref*: reference frequency (supplied in DE7 with control bit 7)

m: contents of counter 2/3 (reference clock pulse count)

n: number of CLK pulses in counter 0/1 (corresponds to COMPARE, provided it was not terminated prematurely by Stop)

Timing diagram:



Mode 6 pulse measuring, Pulse LOW, 50kHz with direction control

The pulse width of a signal connected to the CLK input is determined by means of an internal time base and saved. The measurement is started with the falling edge of the input signal and it is stopped by the rising edge of the input. This saves the value in 20µs units in a buffer from where it may be retrieved (corresponds to Fref = 50kHz).

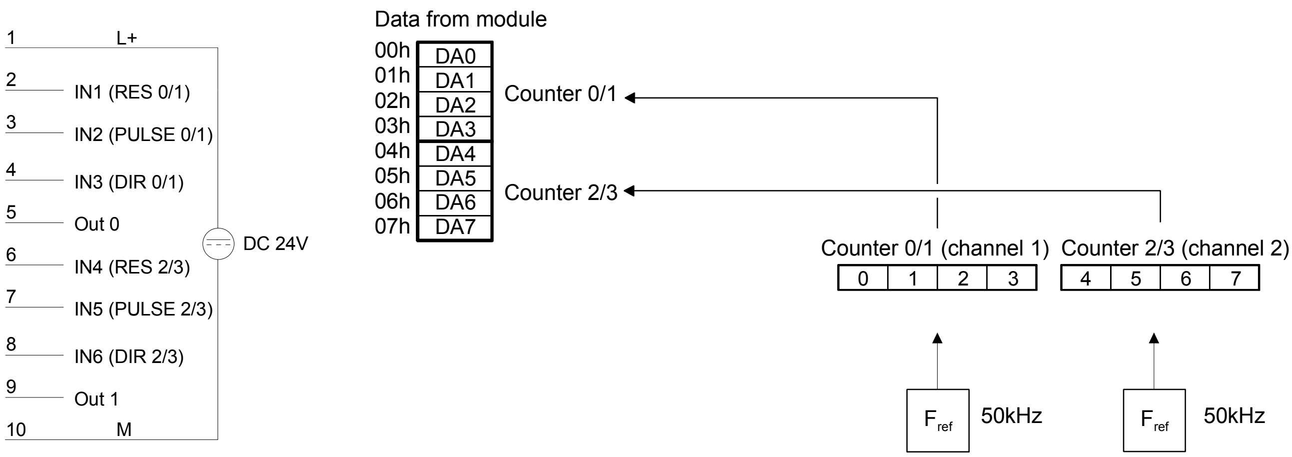
Input DIR determines the counting direction of the counter. If DIR is at a LOW level the counter counts up. A HIGH level lets the counter count down.

The input RES has to be at a LOW level. A HIGH at this input would clear the counter.

With the rising edge of the signal pulse, a result is transferred into the DA area; the result remains available until it is overwritten by the next new result.

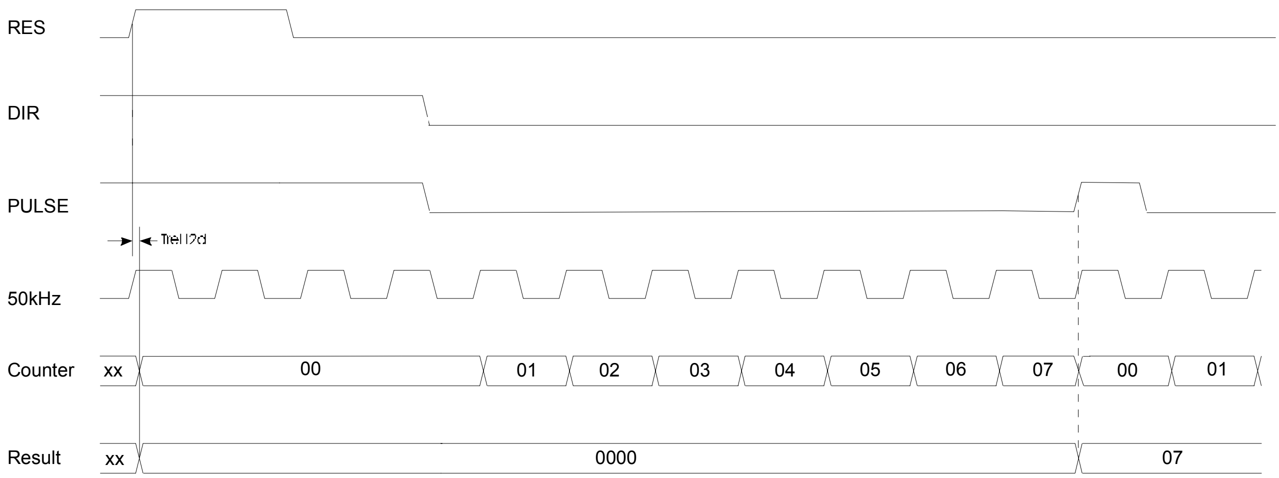
Signals Out 0 or Out 1 are not modified.

Pin assignment access to counter



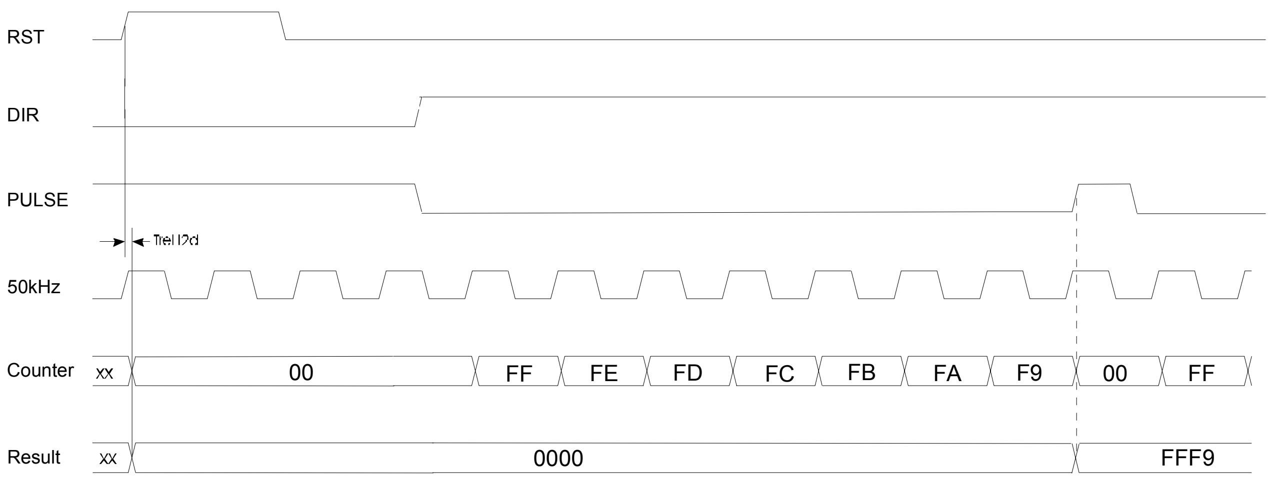
Up counter

The RES signal and the DIR signal are reset. The measurement is started by the falling edge at input PULSE and the counter is clocked up by the 50kHz clock. The rising edge of the signal at input PULSE terminates the count operation and the result is transferred into the result register. The result is available to the PLC. The value remains in the result register until a new measurement has been completed which overwrites the register.



Down counter

The RES signal is reset and the DIR signal is placed at a HIGH level. The measurement is started by the falling edge at input PULSE and the counter is clocked down by the 50kHz clock. The rising edge of the signal at input PULSE terminates the count operation and the result is transferred into the result register. The result is available to the PLC. The value remains in the result register until a new measurement has been completed which overwrites the register.



Mode 20 pulse measurements, pulse down

prog. time base with direction control

The pulse width of a signal that is applied to the PULSE input is determined by means of an internal time base. The measurement is started by the falling edge of the input signal and ends with the rising edge. The rising edge of the measured signal stores the resulting pulse width in units of 1/Fref, that may me retrieved again.

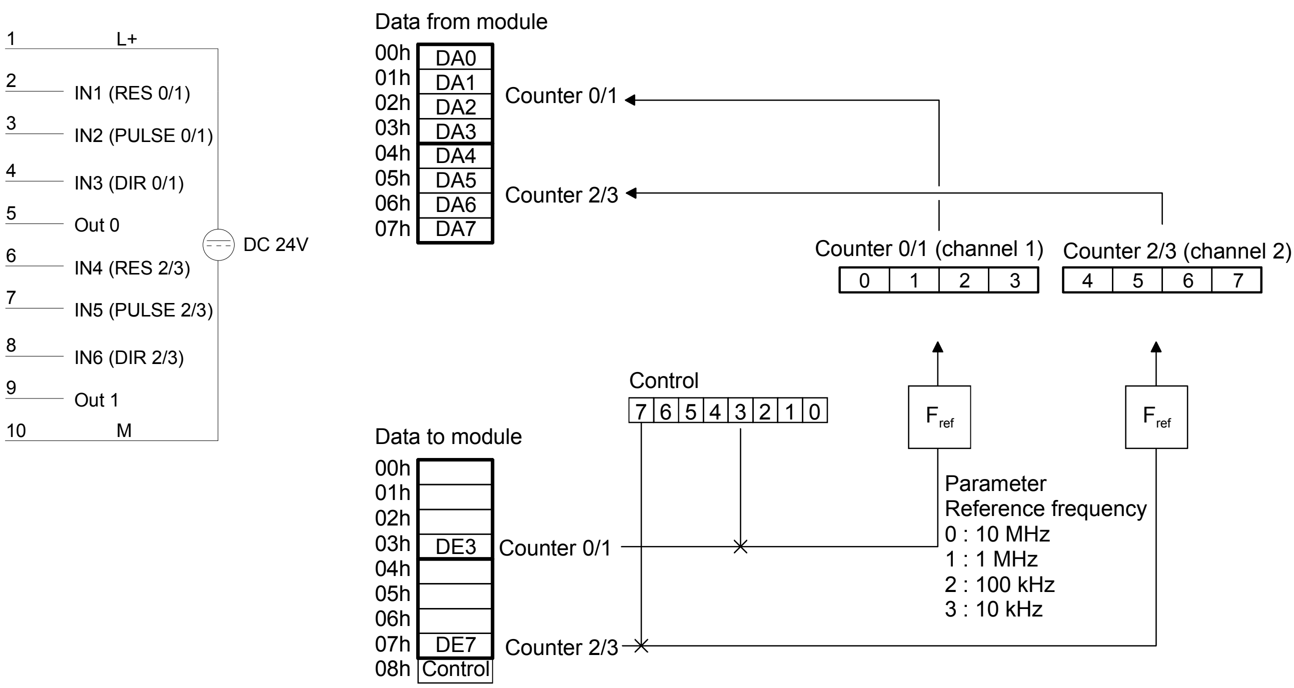
Input DIR controls the direction of the count. When DIR is held at a LOW level the counter counts up. When DIR is at a HIGH level the counter counts down.

RES has to be held at LOW during the counting operation. A HIGH level clears the counter.

Fref is programmable.

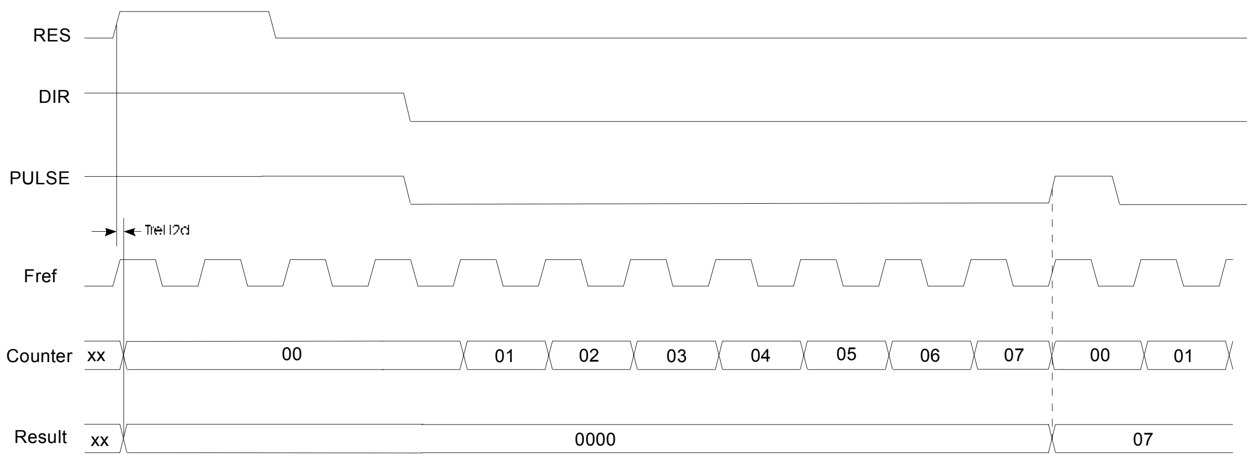
The OUT signal is not changed.

Pin assignment access to counter



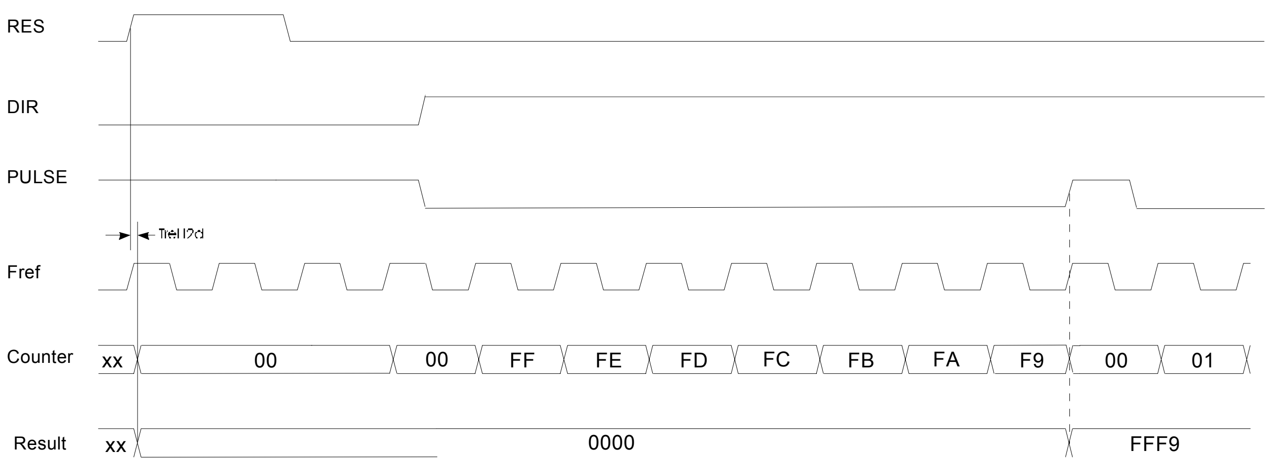
Up counter

The RES signal and the DIR signal are set to LOW. Subsequently the measurement is started with the falling edge of PULSE and the counter counts up in accordance with the selected time base. A rising edge at PULSE terminates the counting operation and the accumulated count is transferred into the result register. The result register is available to the PLC. The value remains in the result register until a new measurement has been completed and the register is changed by the new result.



Down counter

The RES signal is set to LOW and the DIR signal to HIGH. Subsequently the measurement is started with the falling edge of PULSE and the counter counts down in accordance with the selected time base. A rising edge at PULSE terminates the counting operation and the accumulated count is transferred into the result register. The result register is available to the PLC. The value remains in the result register until a new measurement has been completed and the register is changed by the new result.



Mode 21 pulse measurement, pulse low

Direction up, prog. time base, with release

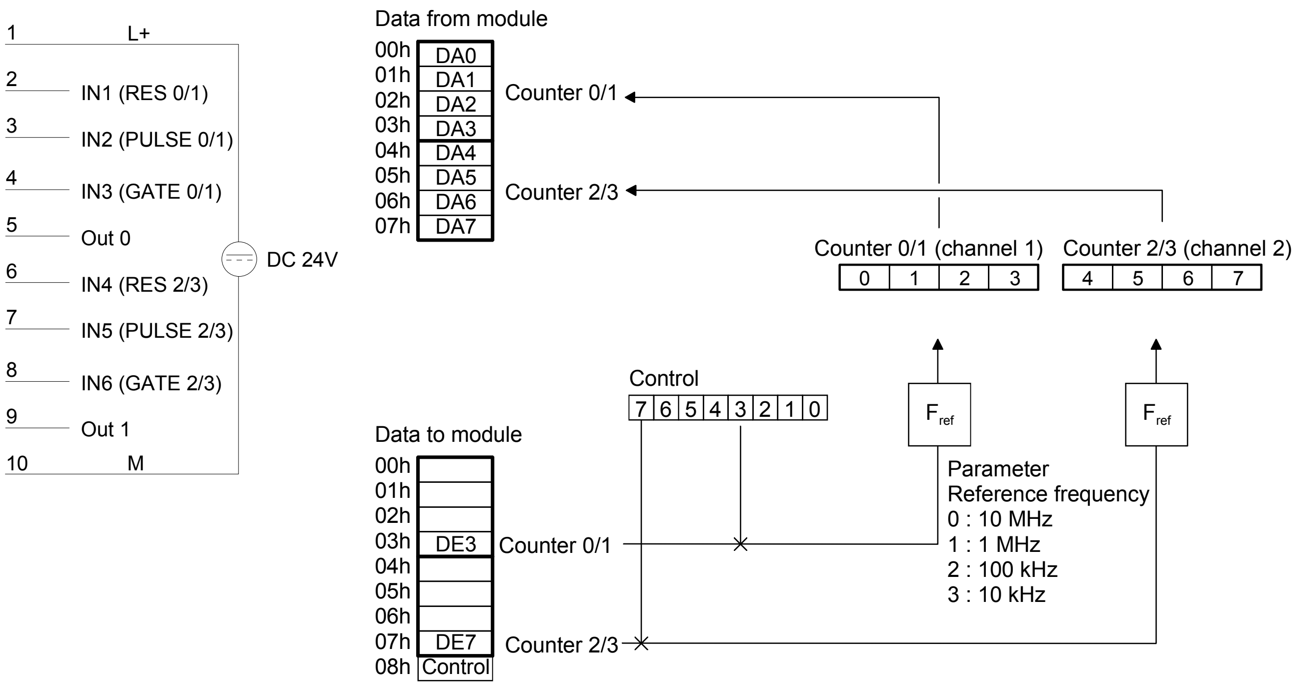
The pulse width of a signal applied to the PULSE input is determined by means of a programmable time base (Fref). The measurement starts with the falling edge of the input signal and it is stopped by the rising edge of the input signal. The rising edge of the input signal saves the resulting pulse width in units of 1/Fref. This is available to other devices.

A condition for the function is that a HIGH level is applied to the GATE input.

Input RES must be at a LOW level. A HIGH level at this input would clear the counter.

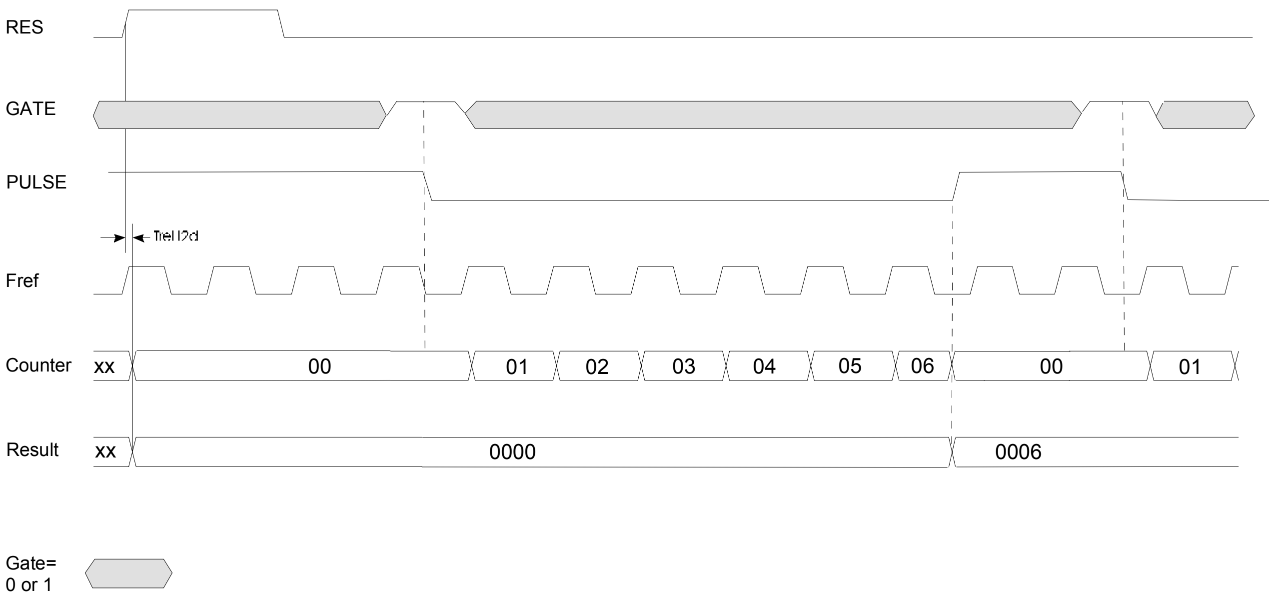
The OUT signal is not modified.

Pin assignment access to counter



Up counter

The RES signal is set to zero. The measurement can only be started when the GATE signal is at a HIGH level. The measurement is started with the falling edge of PULSE and the counter counts up in accordance with the selected time base. A rising edge at PULSE terminates the counting operation and the accumulated count is transferred into the result register. The result register is available to the PLC. The value remains in the result register until a new measurement has been completed and the register is changed by the new result.



Mode 22 pulse measurement, pulse high

Direction up, prog. time base, with release

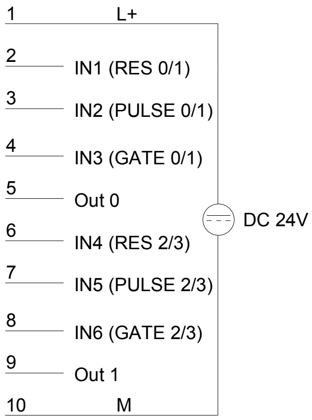
The pulse width of a signal applied to the PULSE input is determined by means of a programmable time base (Fref). The measuring starts with the rising edge of the input signal and ends with the falling edge. The rising edge of the input signal saves the resulting pulse width in units of 1/Fref. This is available to other devices.

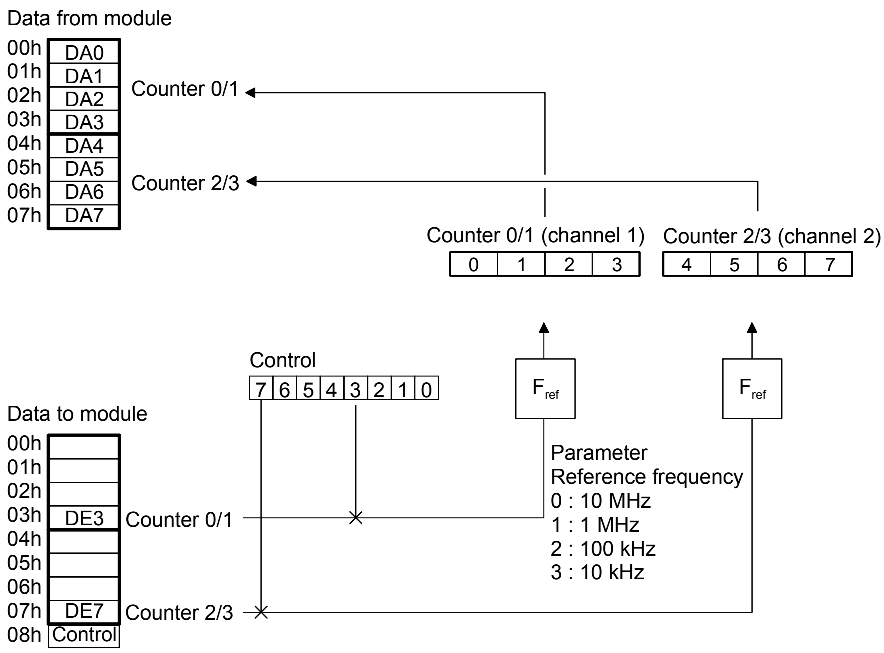
A condition for the function is that a HIGH level is applied to the GATE.

Input RES must be at a LOW level. A HIGH level at this input would clear the counter.

The OUT signal is not modified.

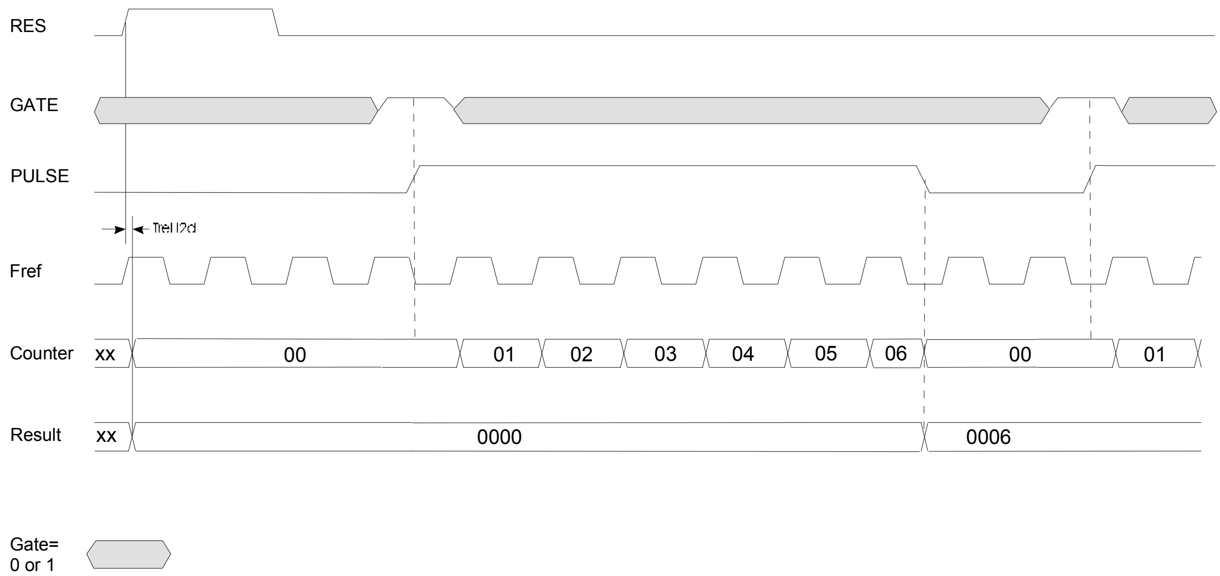
Pin assignment access to counter





Up counter

The RES signal is set to zero. The measuring only starts if the GATE signal is set to HIGH with the rising edge at PULSE. A falling edge at PULSE terminates the counting operation and the accumulated count is transferred into the result register. The result register is available to the PLC. The value remains in the result register until a new measurement has been completed and the register is changed by the new result.

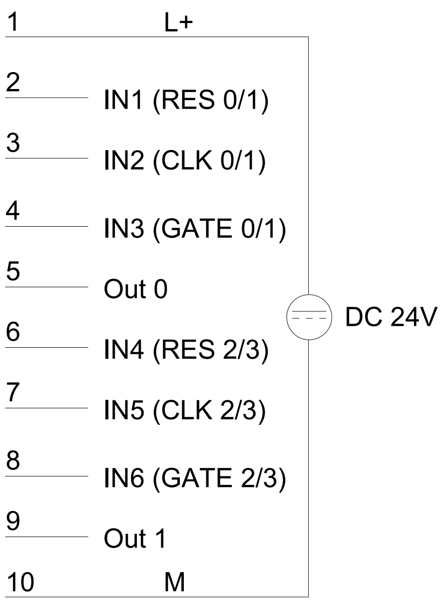


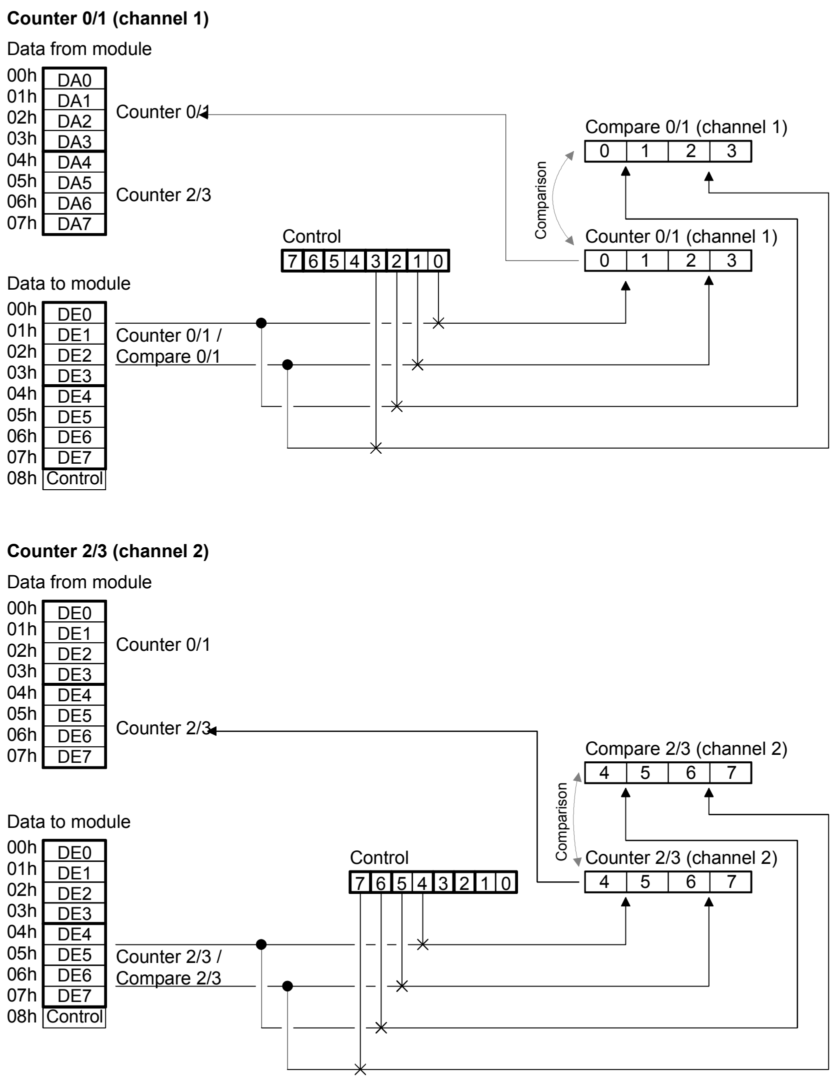
Mode 23
One Shot, count up, with release, output signal

In mode 23 you may implement one 32Bit counter per channel, each one controlled by a GATE signal. Every rising edge of the input clock increments the counter by 1 as long as the signal applied to GATE is HIGH. RES must be at a LOW level. A HIGH level clears the counter. The counter is started by loading. Starting the counter, the output OUT is set active (HIGH). OUT is cleared when the value entered into COMPARE is reached. The counter will continue the count operation after the value in COMPARE was reached.

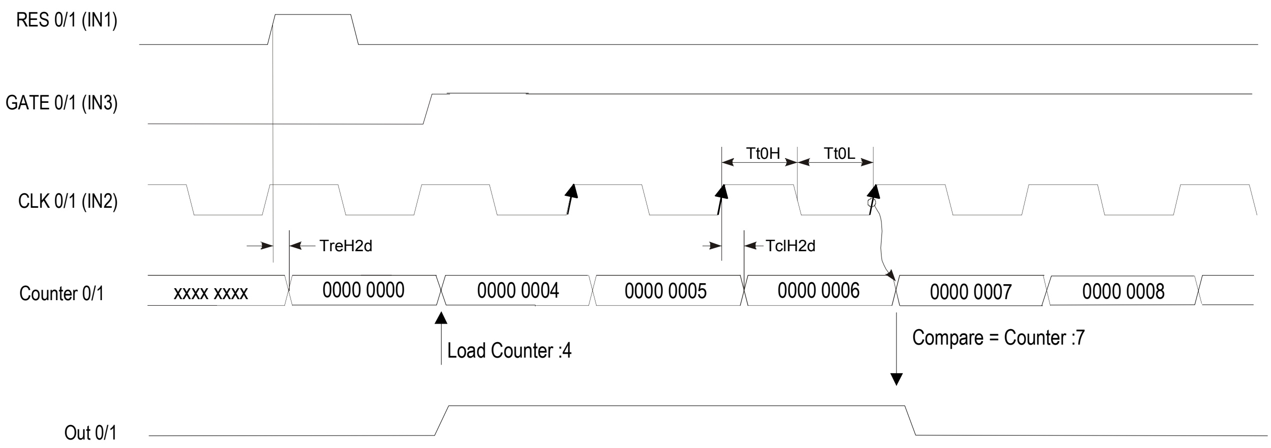
Mode 23 - One Shot, up with Gate input, Output set

Pin assignment access to counter





Timing diagram Example of counter 0/1 in mode 23:

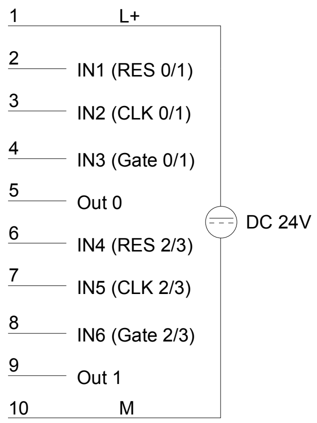


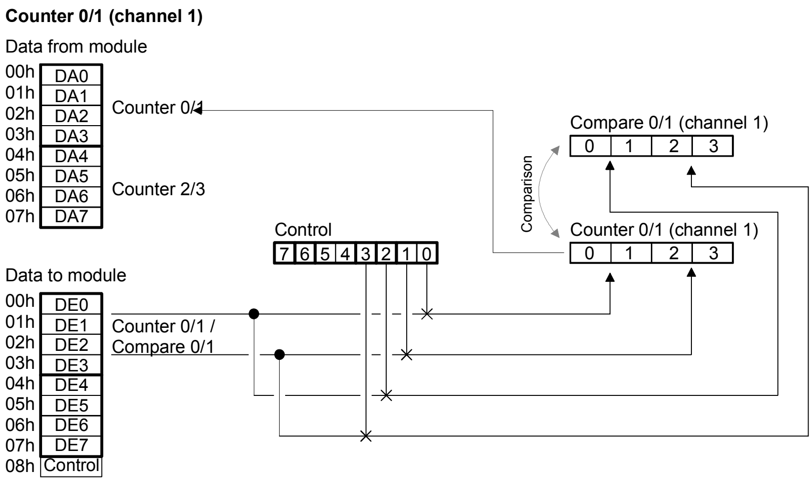
Mode 24
One Shot, count down, with gate, output signal

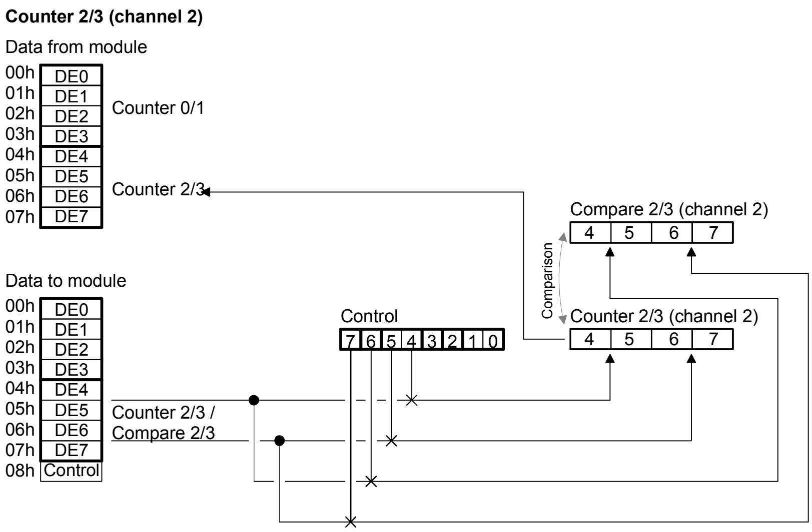
In mode 24 you may implement one 32Bit counter per channel, each one controlled by the signal applied to the GATE input. Every rising edge of the input clock decrements the counter by 1 as long as the signal applied to GATE is HIGH. RES must be at a LOW level. A HIGH level at this input would clear the counter. The counter is started by loading. Starting the counter, the output OUT is set active (HIGH). OUT is cleared when the value entered into COMPARE is reached. The counter will continue the count operation after the value in COMPARE was reached.

Mode 24 - One Shot, down with Gate-Input, Output set

Pin assignment access to counter

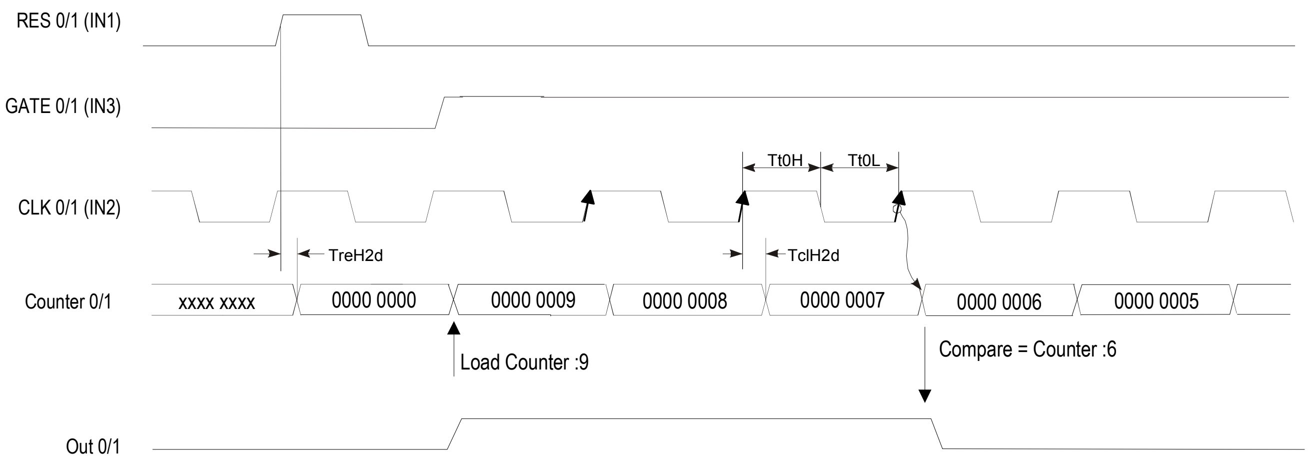






Timing diagram

Example of counter 0/1 in mode 24:

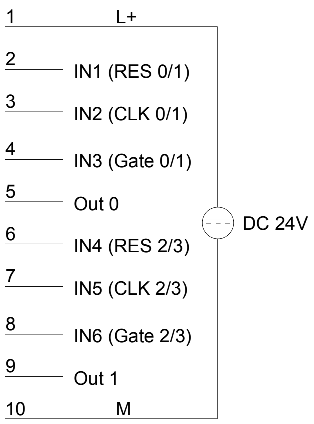


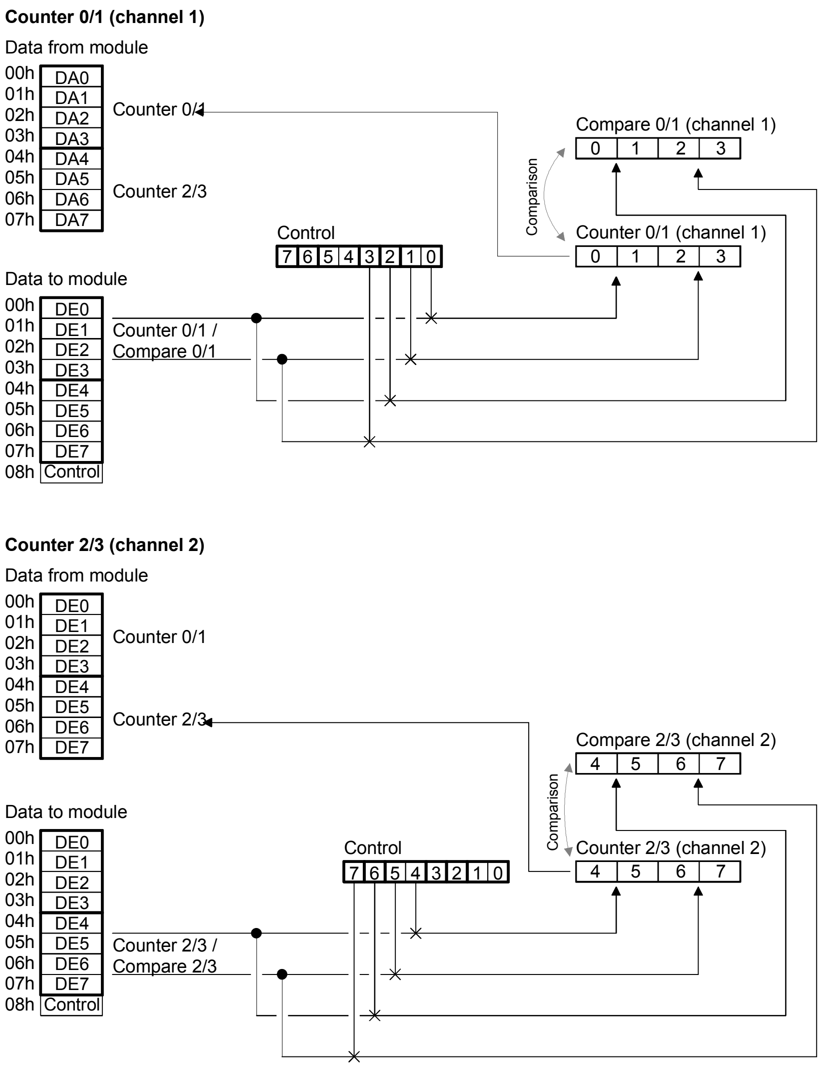
Mode 25
One Shot, count up, with reset signal

In mode 25 you may implement one 32Bit counter per channel, each one controlled by the signal applied to the GATE input. Every rising edge of the input clock increments the counter by 1 as long as the signal applied to GATE is HIGH. RES must be at a LOW level. A HIGH level at this input would clear the counter. The counter is started by loading. Starting the counter, the output OUT is set active (LOW). OUT becomes HIGH when the value entered into COMPARE is reached.

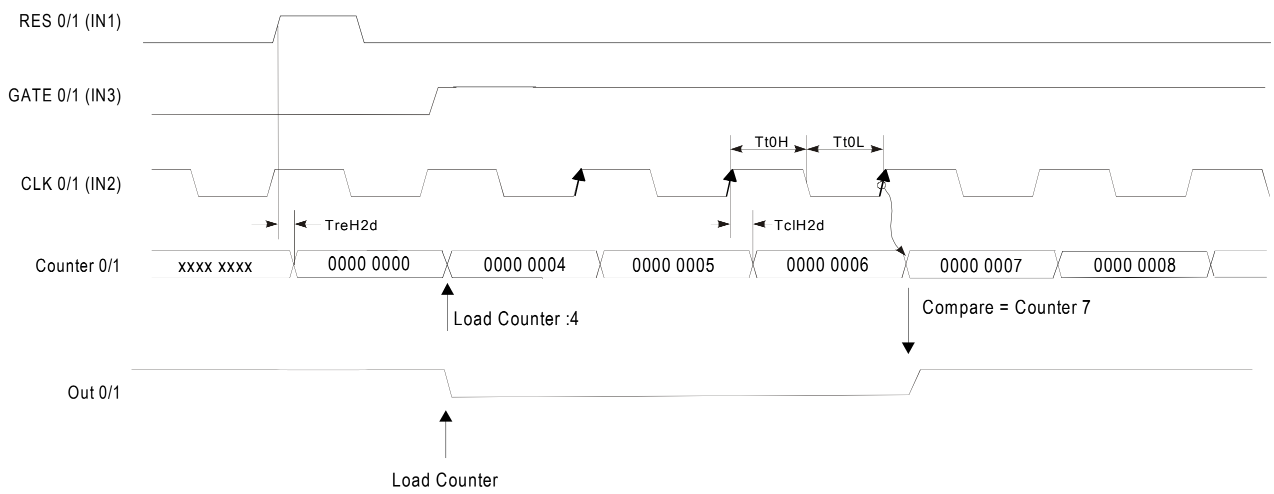
Mode 25 One Shot, count up, Reset

Pin assignment access to counter





Timing diagram Example of counter 0/1 in mode 25:

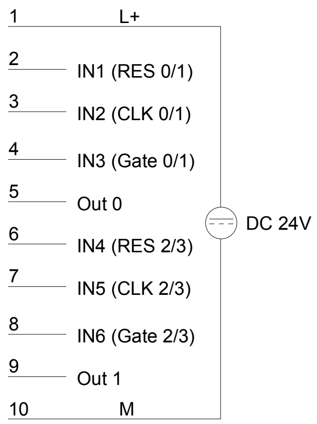


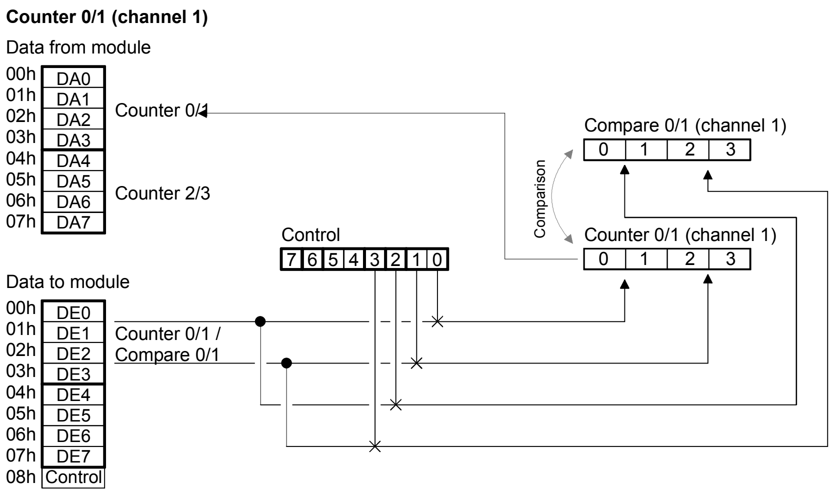
Mode 26 One Shot, count down, with reset signal

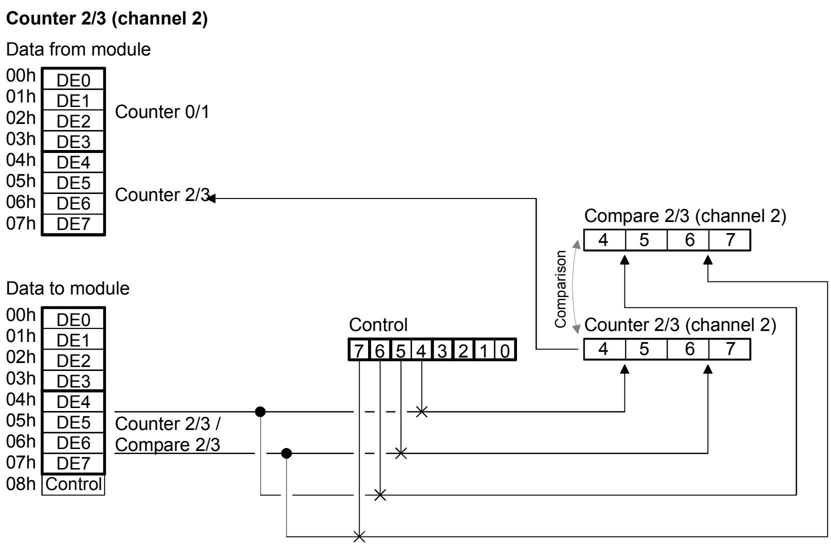
In mode 26 you may implement one 32Bit counter per channel, each one controlled by the signal applied to the GATE input. Every rising edge of the input clock decrements the counter by 1 as long as the signal applied to GATE is HIGH. RES must be at a LOW level. A HIGH level at this input would clear the counter. The counter is started by loading. Starting the counter, the output OUT is set active (LOW). OUT becomes HIGH when the value entered into COMPARE is reached.

Mode 26 - One Shot, down, Reset

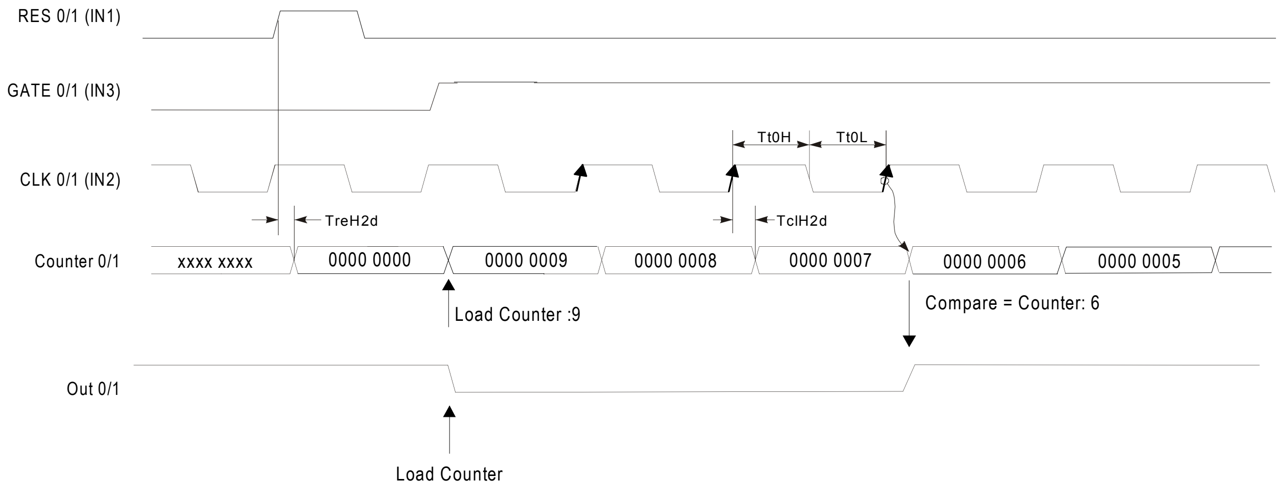
Pin assignment access to counter







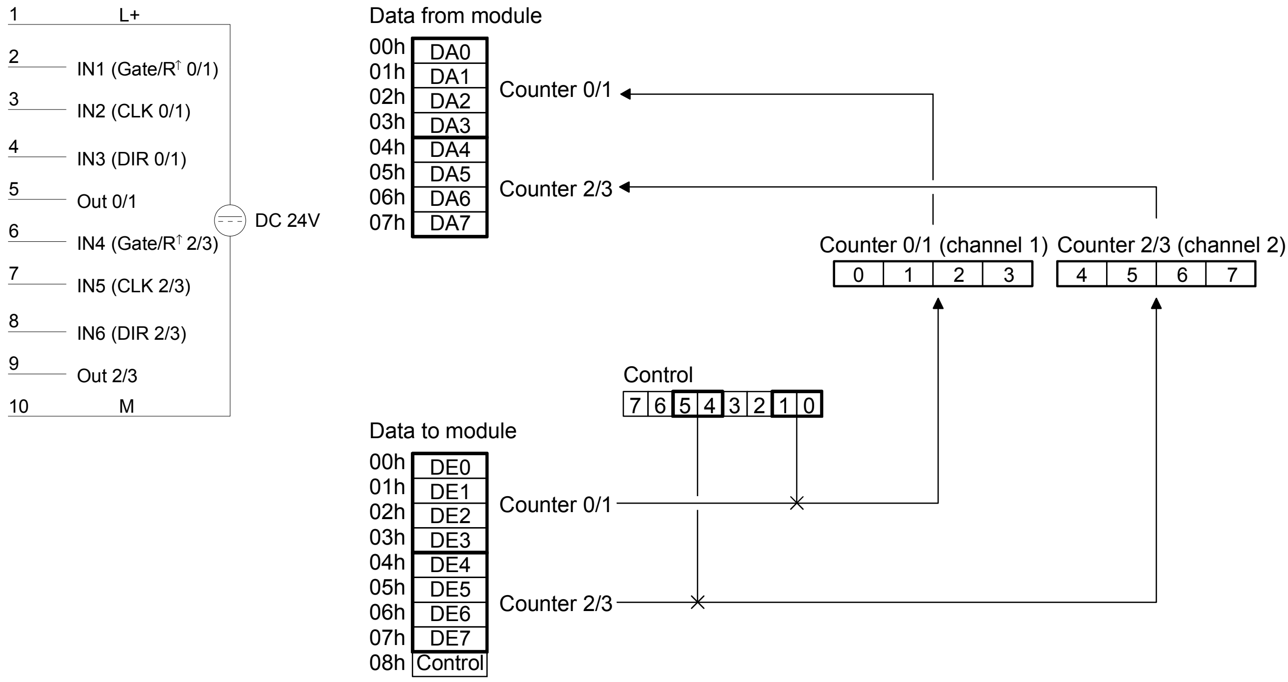
Timing diagram Example of counter 0/1 in mode 26:



Mode 27
32Bit counter

You determine the direction by means of the DIR input (IN3 or IN6). Every rising or falling edge of the input clock signal increments or decrements the counter. The rising edge of the signal Gate/R↑ resets the counter. During the count process, the signal Gate/R↑ has to be HIGH. When the signal Gate/R↑ becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter should continue counting. If the counter stops at zero, the output remains active.

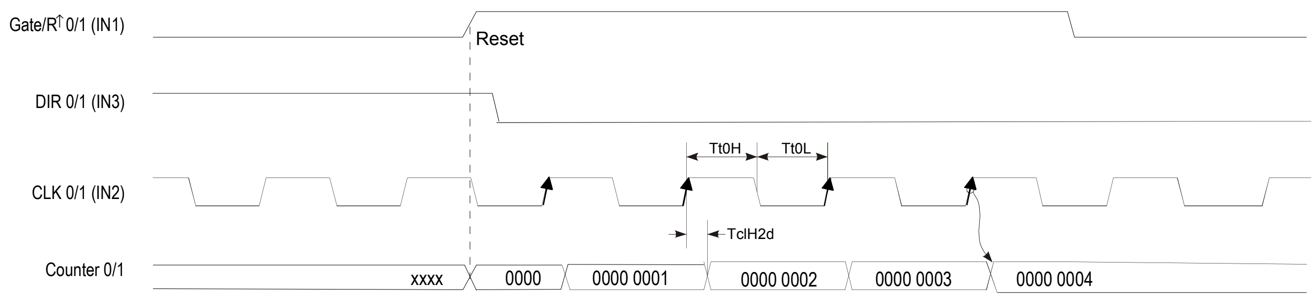
Pin assignment
access to counter



Up counter

In mode 27, a LOW level at the DIR input configures the counter for counting up.

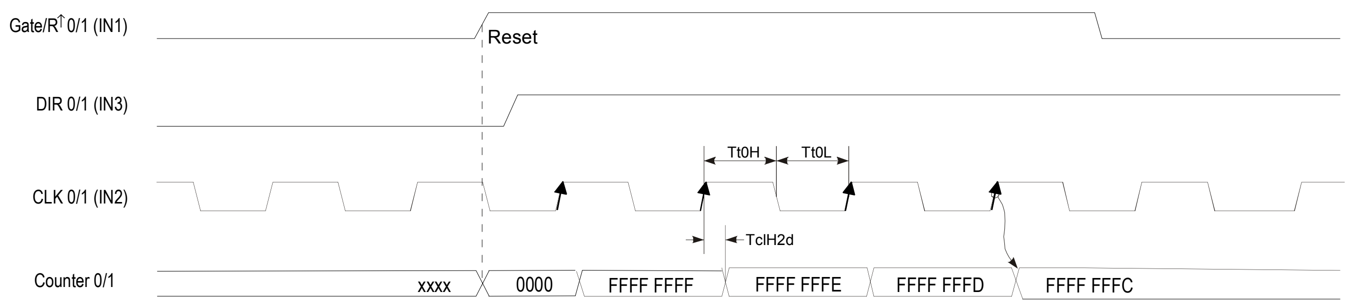
Timing diagram of the counter 0/1 example:



Down counter

In mode 27, a HIGH level at the DIR input configures the counter for counting down.

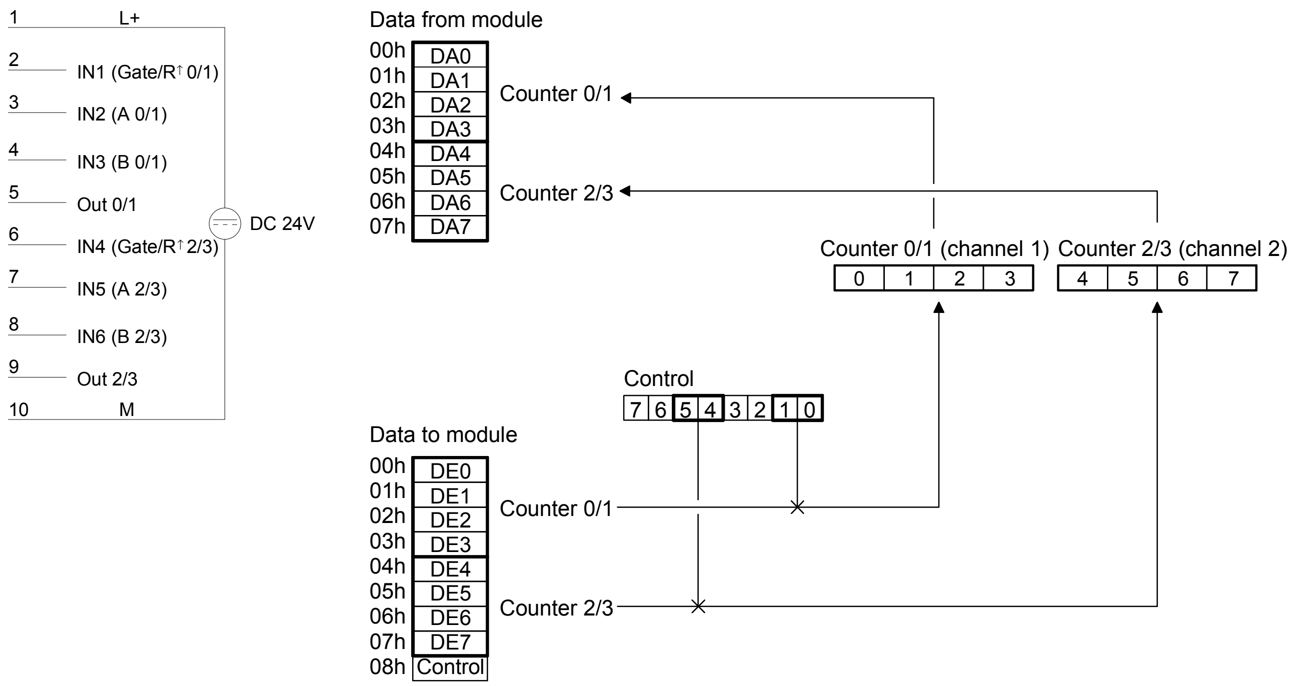
Timing diagram of the counter 0/1 example:



Mode 28
Encoder 1 edge

In mode 28 you may configure an encoder for one of the channels. Depending on the direction of rotation this encoder will increment or decrement the internal counter with every falling edge. The rising edge of the signal Gate/R↑ resets the counter. During the count process, the signal Gate/R↑ has to be HIGH. When the signal Gate/R↑ becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero the output remains active.

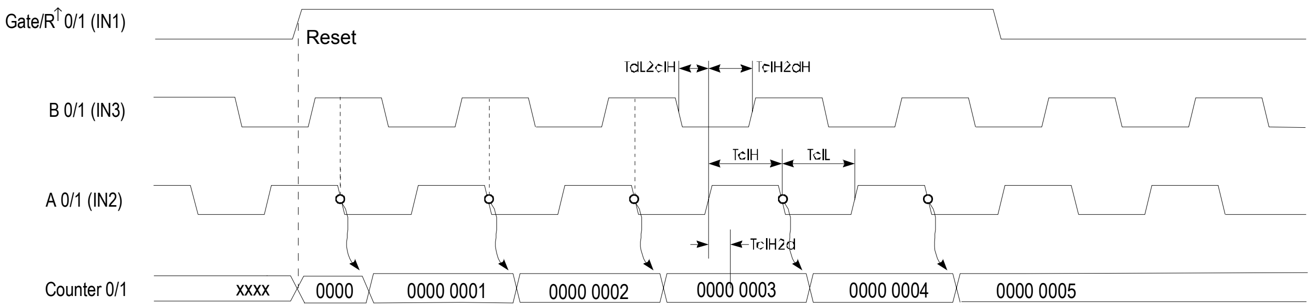
Pin assignment
access to counter



Up counter

Every falling edge of the signal at input A increments the counter if input B is at HIGH level at this moment.

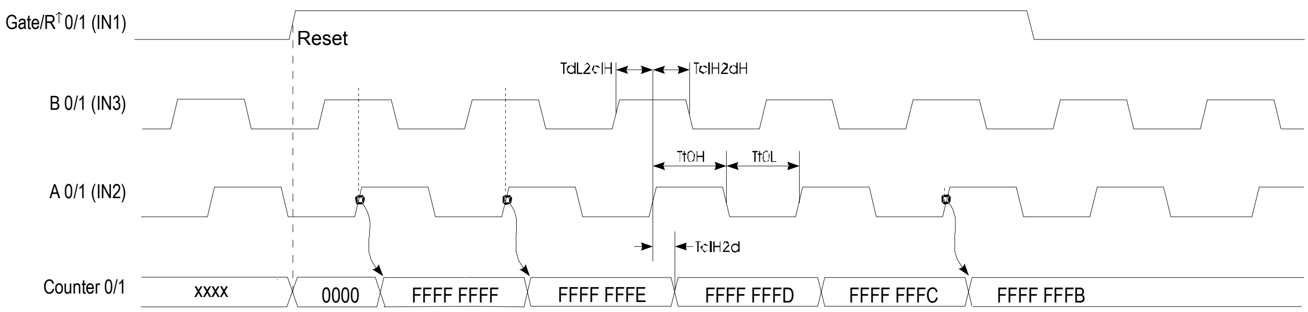
Timing diagram for the counter 0/1 example:



Down counter

Every rising edge of the signal at input A decrements the internal counter if input B is at HIGH level at this moment.

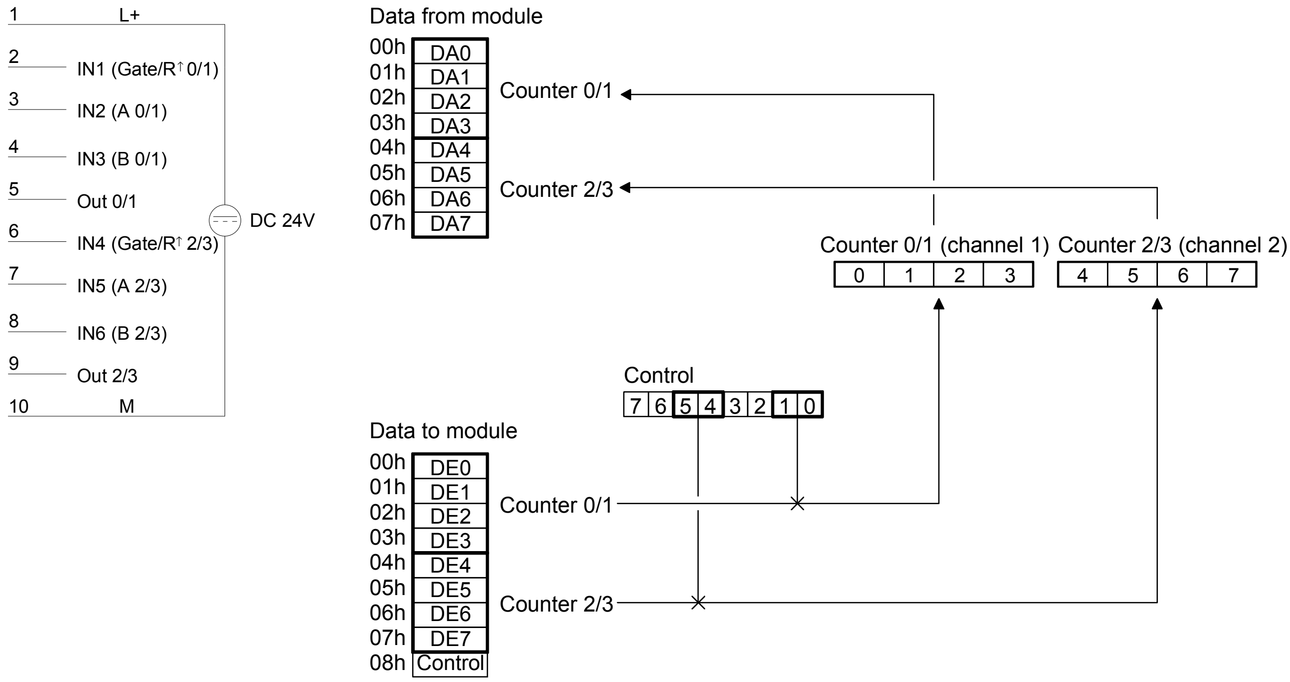
Timing diagram for the counter 0/1 example:



Mode 29
Encoder 2 edges

Every rising or falling edge of the signal at input A changes the counter by 1. The direction of the count depends on the level of the signal applied to input B. The rising edge of the signal Gate/R$^{\uparrow}$ resets the counter. During the count process, the signal Gate/R$^{\uparrow}$ has to be HIGH. When the signal Gate/R$^{\uparrow}$ becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero the output remains active.

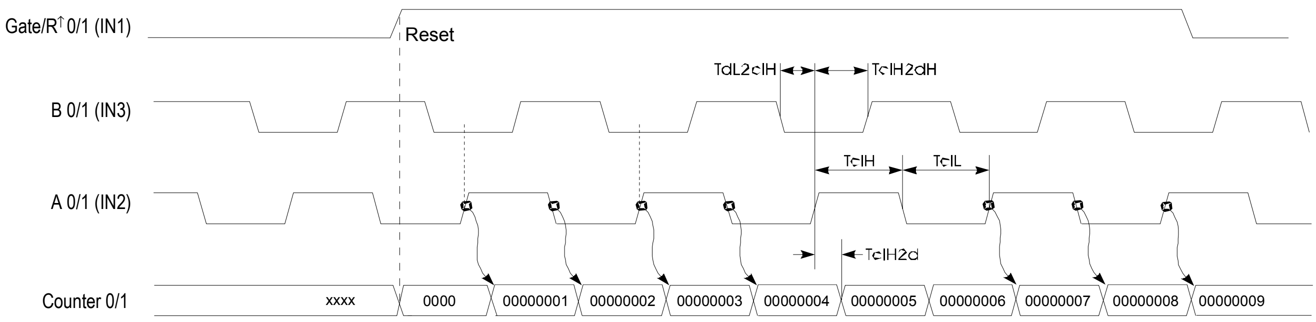
Pin assignment
access to counter



Up counter

The counter is incremented by the rising edge of signal A if input B is at a LOW level or by the falling edge of input A when input B is at a HIGH level.

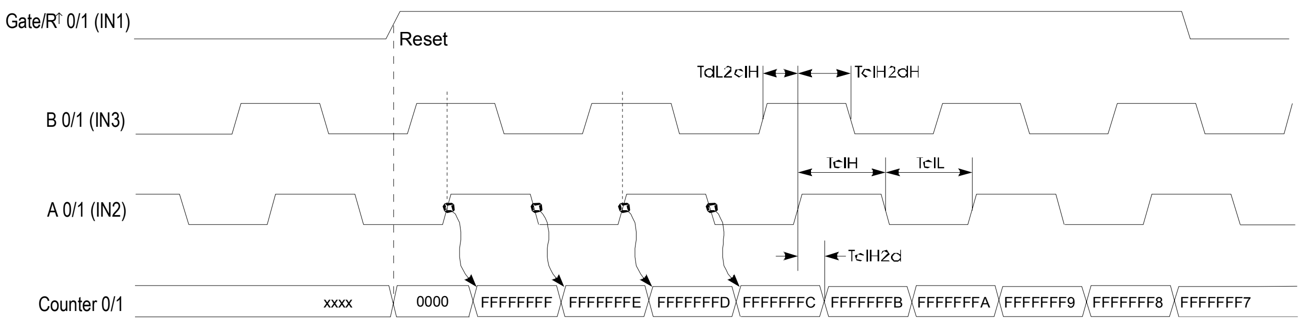
Timing diagram for the counter 0/1 example:



Down-counter

The counter is decremented by the rising edge of signal A if input B is at a HIGH level or by the falling edge of input A when input B is at a LOW level.

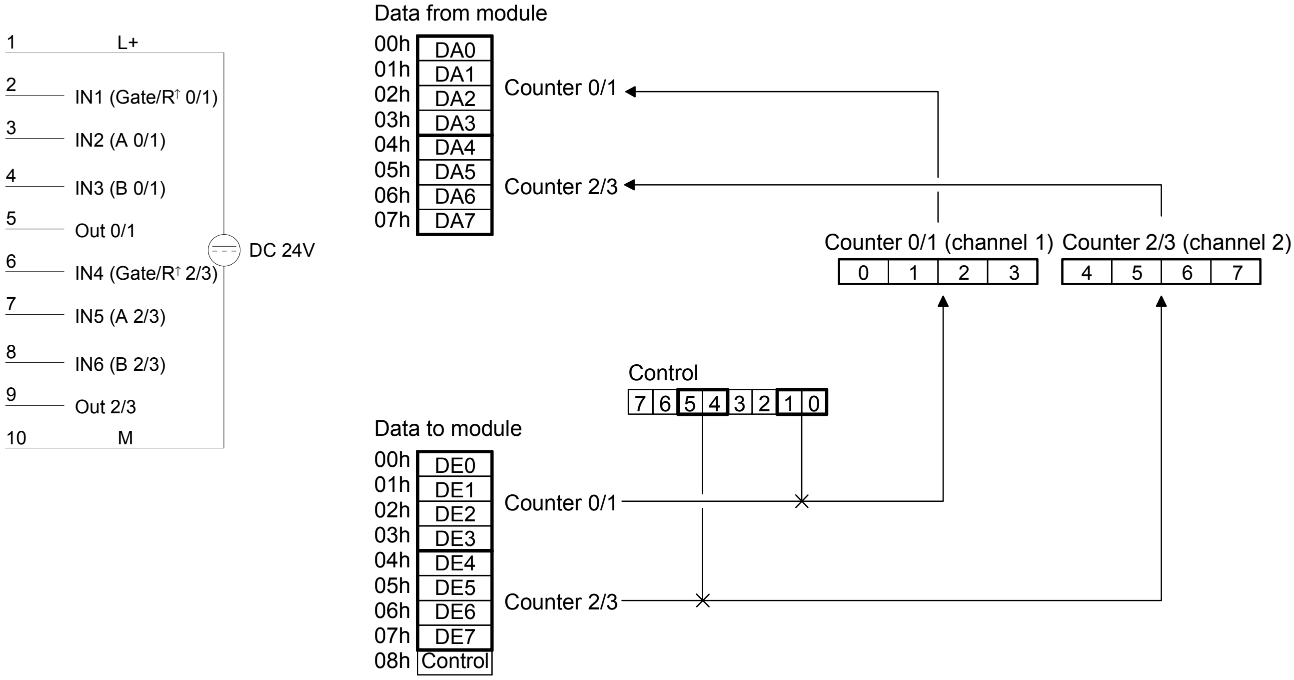
Timing diagram for the counter 0/1 example:



Mode 30
Encoder 4 edges

Every rising or falling edge at inputs A or B increments or decrements the counter. The direction depends on the level applied to the other input (B or A). The rising edge of the signal Gate/R$^{\uparrow}$ resets the counter. During the count process, the signal Gate/R$^{\uparrow}$ has to be HIGH. When the signal Gate/R$^{\uparrow}$ becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero, the output remains active.

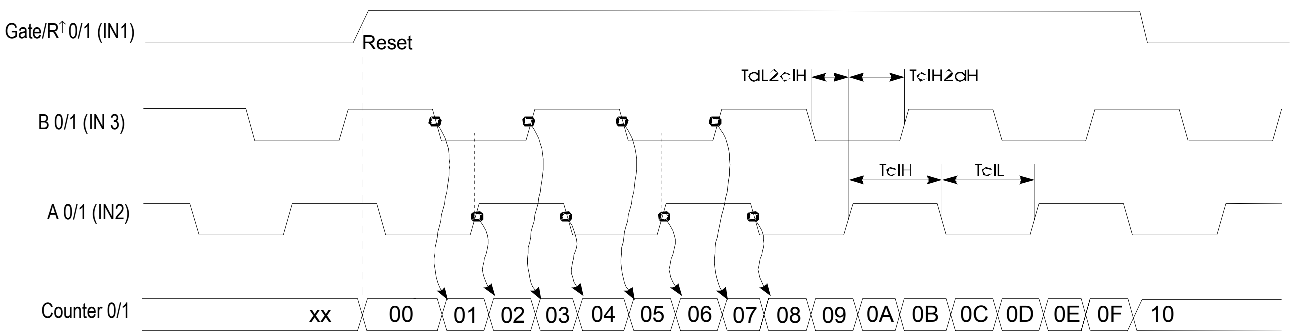
Pin assignment
access to counter



Up counter

The counter is incremented when a rising edge is applied to B while input A is at a HIGH level or if a falling edge is applied to B when input A is at a LOW level. Alternatively it is also incremented when a rising edge is applied to A when input B is at a LOW level or by a falling edge at A when input B is at a HIGH level.

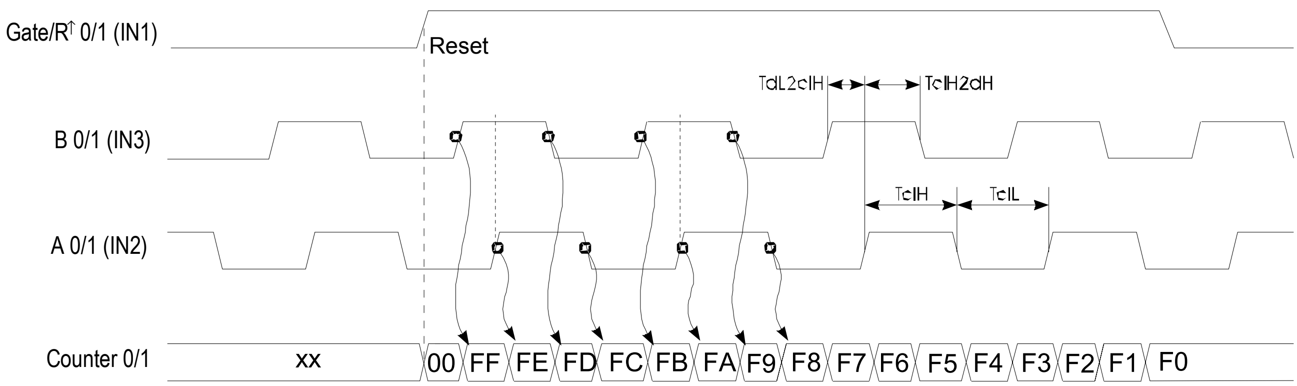
Timing diagram for the counter 0/1 example:



Down counter

The counter is decremented when a rising edge is applied to B while input A is at a LOW level or if a falling edge is applied to B when input A is at a HIGH level. Alternatively it is also decremented when a rising edge is applied to A when input B is at a HIGH level or by a falling edge at input A when input B is at a LOW level.

Timing diagram for counter 0/1 example:



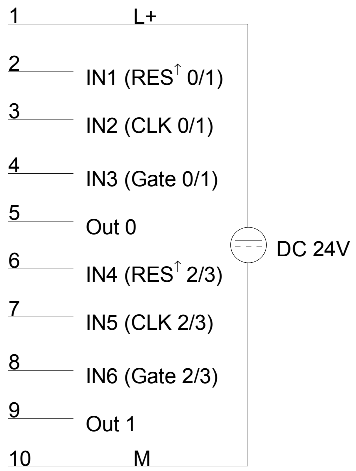
Mode 31 and 32 32bit counter with gate

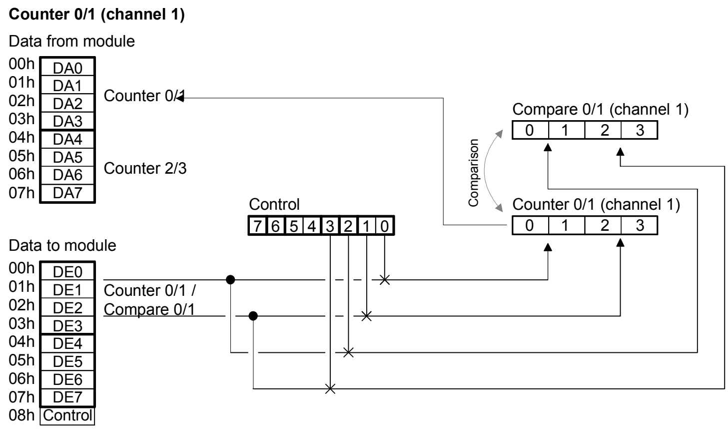
In mode 31 and mode 32 you can implement a 32Bit counter that is controlled by a gating signal (Gate). The direction of counting depends on the selected mode. Every rising edge of the input signal increments or decrements the counter provided that the GATE signal is at HIGH level. A rising edge of RES↑ clears the counter. When the counter reaches the value that was previously loaded into the compare register, output OUT is set active for a minimum period of 100ms while the counter continues counting.

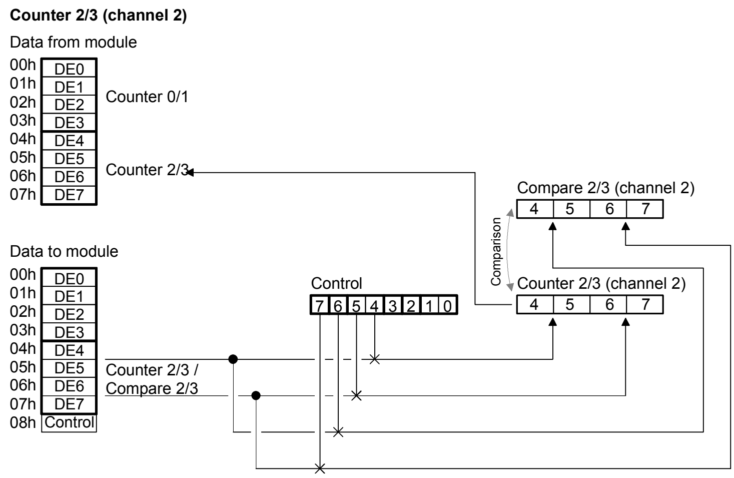
Mode 31 - 32Bit counter up + gate with compare

Mode 32 - 32Bit counter down + gate with compare

Pin assignment access to counter

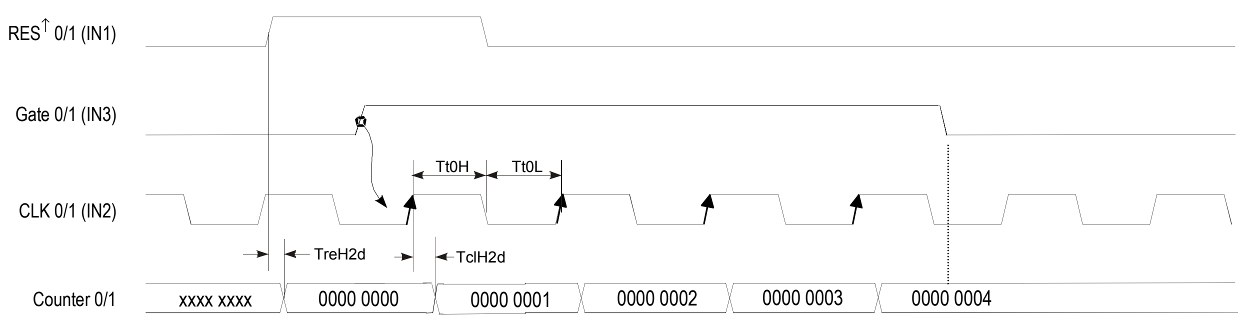






Timing diagram

Below follows an example of a timing diagram of counter 0/1 in mode 31:



Mode 33 and 34 32Bit counter with gate and auto reload

Modes 33 and 34 operate in the same manner as mode 31 and 32 with the addition of an Auto Reload function. The "Auto Reload" is used to define a value in the load register that is used to preset the counter automatically when it reaches the compare value.

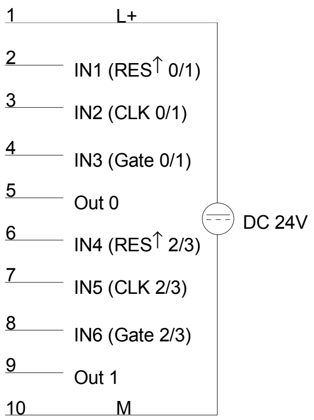
A rising edge of RES↑ clears the counter to 0000 0000. A HIGH level applied to GATE enables the counter so that is incremented/decremented by every rising edge of the CLK signal. As long as GATE is HIGH, the counter will count every rising edge of the signal applied to CLK until the count is one less than the value entered into COMPARE. The next pulse overwrites the counter with the value contained in the load register. This process continues until GATE is set to a LOW level. When an auto reload occurs, the status of the respective output changes.

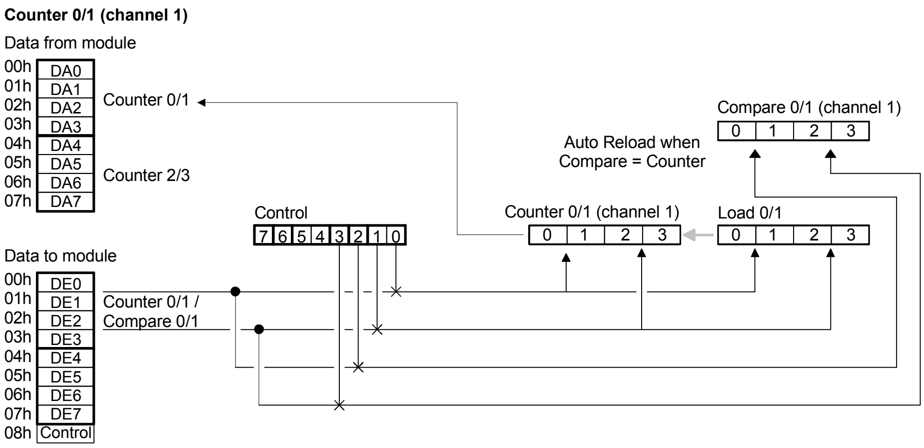
The RES↑ signal only resets the counter but not the outputs.

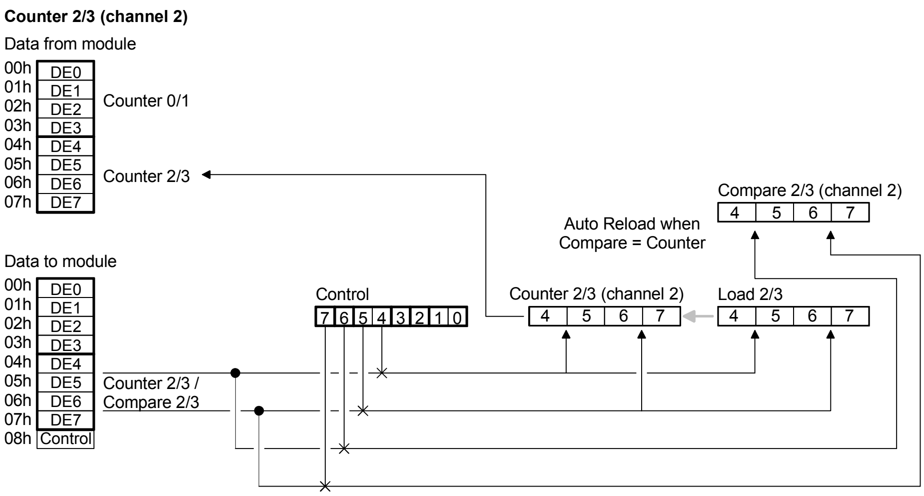
Mode 33 - 32Bit counter up + gate with compare and auto reload

Mode 34 - 32Bit counter down + gate with compare and auto reload

Pin assignment access to counter







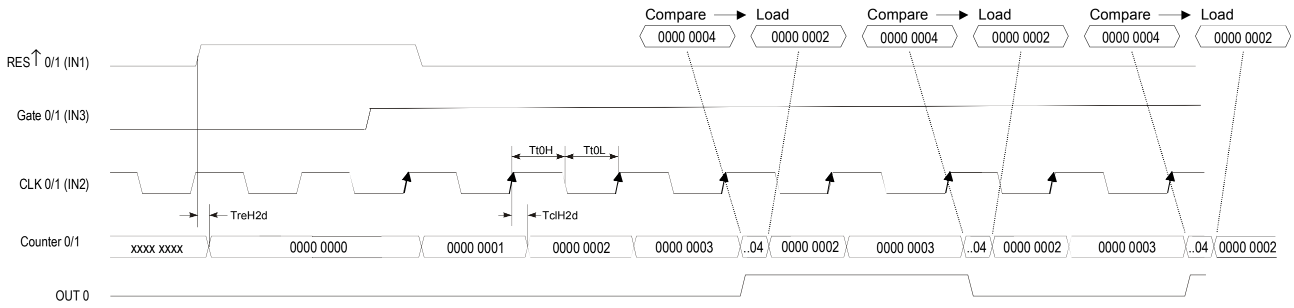
Example

This example is intended to explain the operation of the counters in mode 33 and 34.

A rising edge of RES↑ clears the counter to 0000 0000. A HIGH level applied to GATE enables the counter. As long as GATE is HIGH the counter will count every rising edge of the signal applied to CLK until the count is one less than the value entered into COMPARE. In this example the counter counts to 0000 0004 followed immediately by an auto reload, i.e. the counter is preset to the contents of the load register (in this case 0000 0002). The state of output OUT 0 changes every time an auto reload is executed.

In this example the counter counts from 0000 0002 to 0000 0004 as long as the GATE input is at a HIGH level.

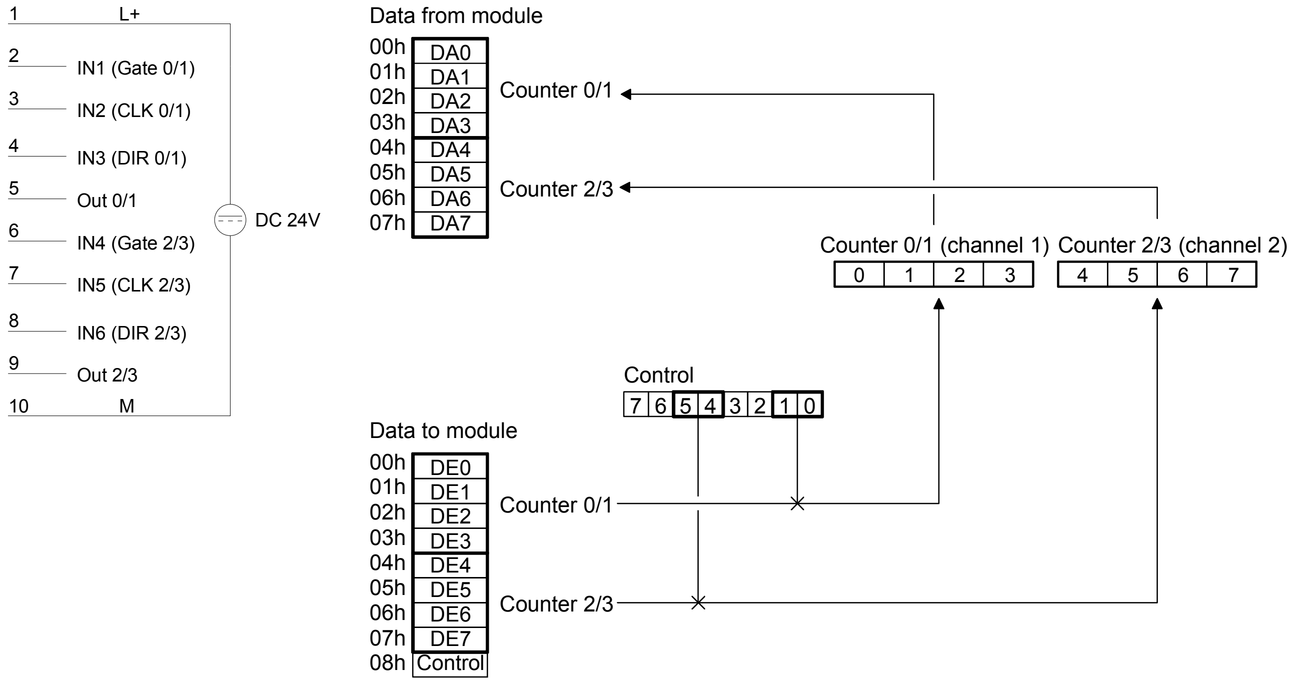
Every load operation changes the status of output OUT 0.



Mode 35
32Bit counter

You determine the direction by means of the DIR input (IN3 or IN6). Every rising or falling edge of the input clock signal increments or decrements the counter. During the count process, the signal Gate has to be HIGH. When the signal Gate becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter should continue counting. If the counter stops at zero, the output remains active.

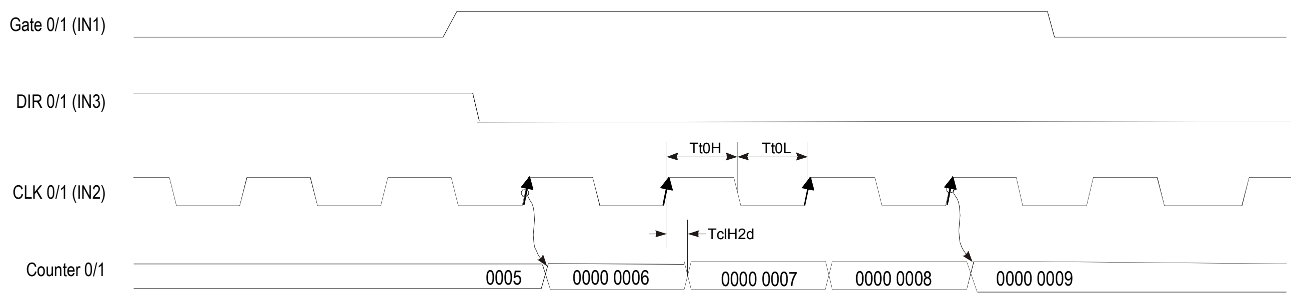
Pin assignment
access to counter



Up counter

In mode 35, a LOW level at the DIR input configures the counter for counting up.

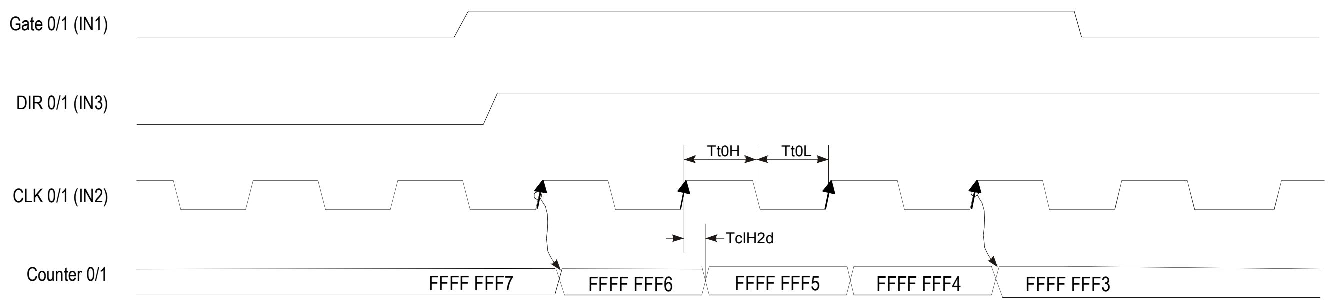
Timing diagram of the counter 0/1 example:



Down counter

In mode 35, a HIGH level at the DIR input configures the counter for counting down.

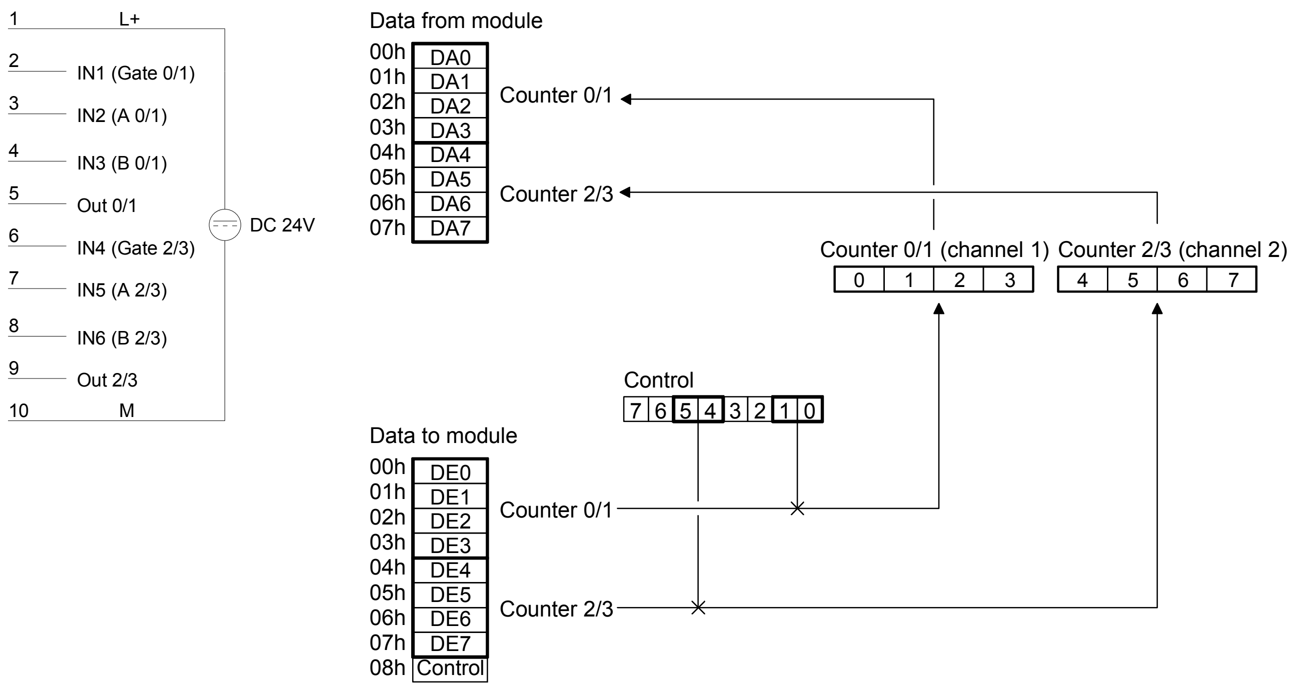
Timing diagram of the counter 0/1 example:



Mode 36
Encoder 1 edge

In mode 36 you may configure an encoder for one of the channels. Depending on the direction of rotation this encoder will increment or decrement the internal counter with every falling edge. During the count process, the signal Gate has to be HIGH. When the signal Gate becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero the output remains active.

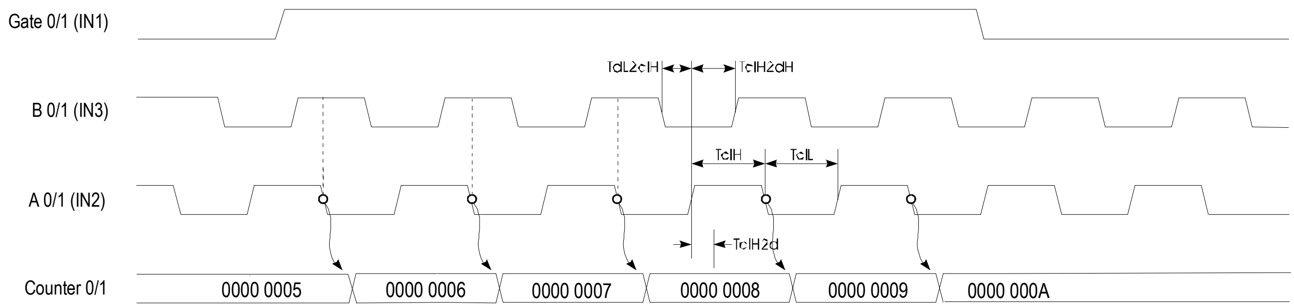
Pin assignment access to counter



Up counter

Every falling edge of the signal at input A increments the counter if input B is at HIGH level at this moment.

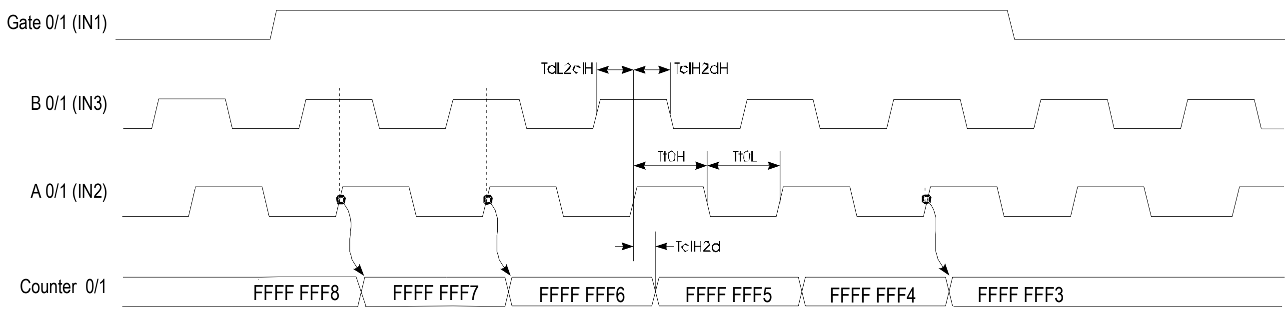
Timing diagram for the counter 0/1 example:



Down counter

Every rising edge of the signal at input A decrements the internal counter if input B is at HIGH level at this moment.

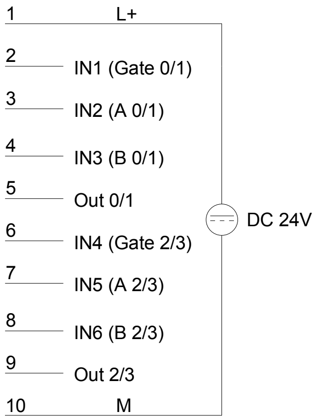
Timing diagram for the counter 0/1 example:

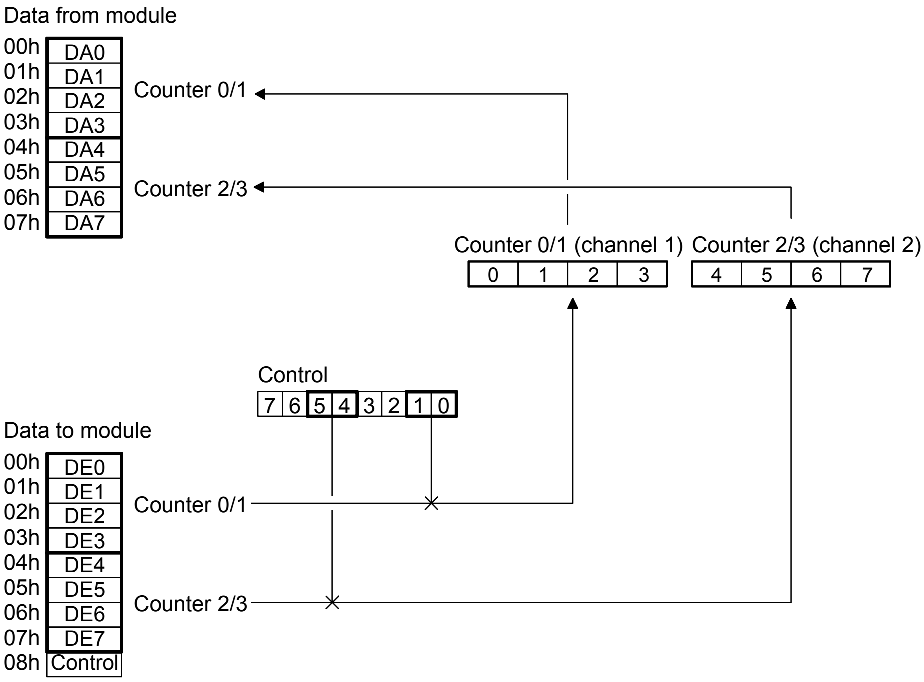


Mode 37
Encoder 2 edges

Every rising or falling edge of the signal at input A changes the counter by 1. The direction of the count depends on the level of the signal applied to input B. During the count process, the signal Gate has to be HIGH. When the signal Gate becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero the output remains active.

Pin assignment
access to counter

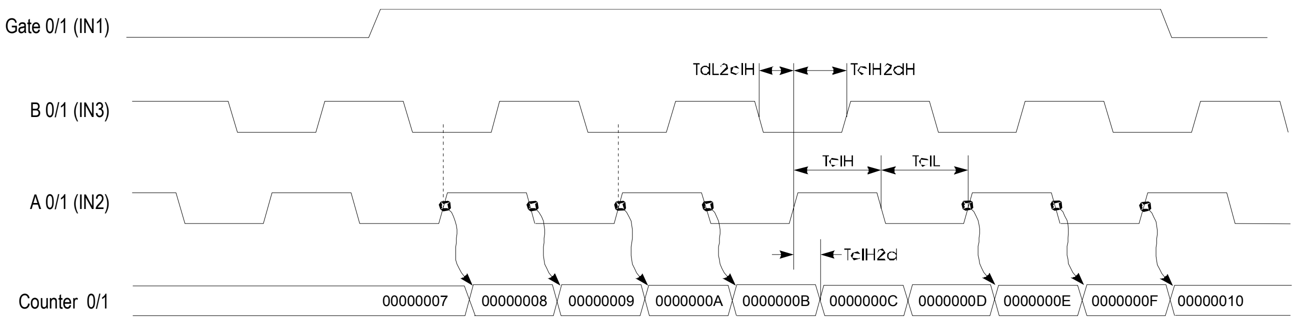




Up counter

The counter is incremented by the rising edge of signal A if input B is at a LOW level or by the falling edge of input A when input B is at a HIGH level.

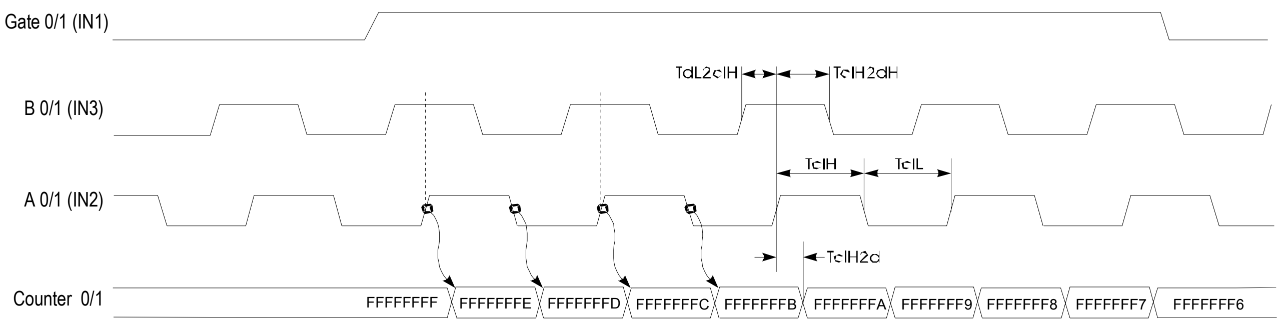
Timing diagram for the counter 0/1 example:



Down-counter

The counter is decremented by the rising edge of signal A if input B is at a HIGH level or by the falling edge of input A when input B is at a LOW level.

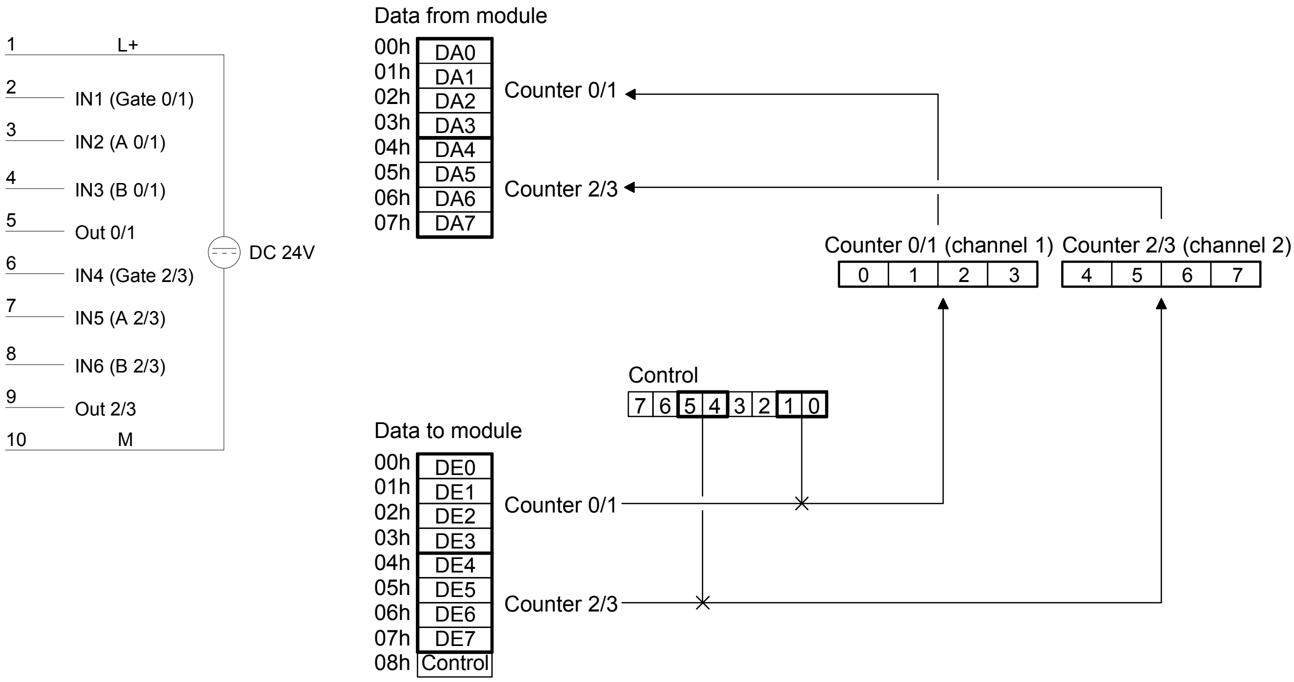
Timing diagram for the counter 0/1 example:



Mode 38
Encoder 4 edges

Every rising or falling edge at inputs A or B increments or decrements the counter. The direction depends on the level applied to the other input (B or A). During the count process, the signal Gate has to be HIGH. When the signal Gate becomes "0", the counter value remains valid. When the counter reaches zero, output OUT of the respective counter is active for a minimum period of 100ms, even if the counter continues counting. If the counter stops at zero, the output remains active.

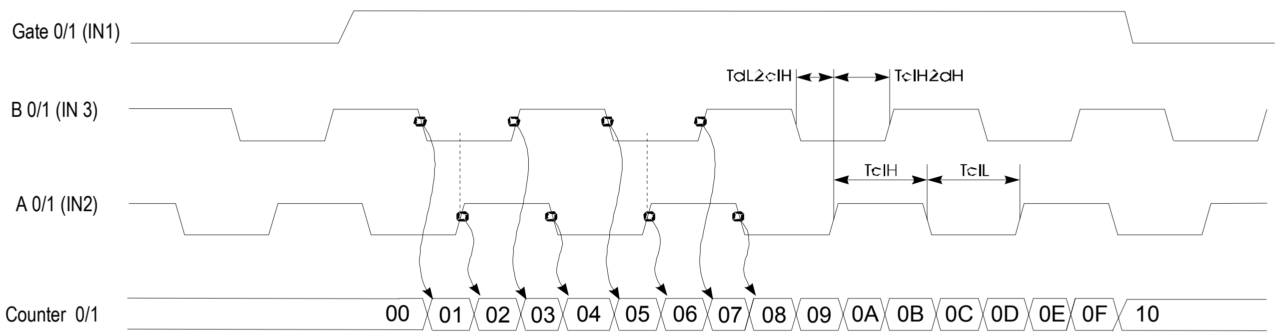
Pin assignment access to counter



Up counter

The counter is incremented when a rising edge is applied to B while input A is at a HIGH level or if a falling edge is applied to B when input A is at a LOW level. Alternatively it is also incremented when a rising edge is applied to A when input B is at a LOW level or by a falling edge at A when input B is at a HIGH level.

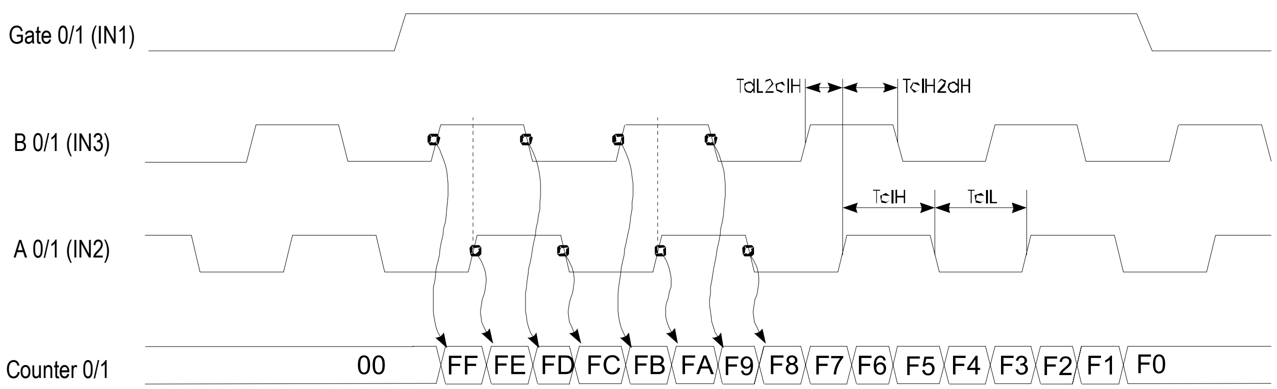
Timing diagram for the counter 0/1 example:



Down counter

The counter is decremented when a rising edge is applied to B while input A is at a LOW level or if a falling edge is applied to B when input A is at a HIGH level. Alternatively it is also decremented when a rising edge is applied to A when input B is at a HIGH level or by a falling edge at input A when input B is at a LOW level.

Timing diagram for counter 0/1 example:



Mode 39 ... 42

Mode 39 - 32Bit Counter up + Gate low active

Mode 40 - 32Bit Counter down + Gate low active

Mode 41 - 32Bit Counter up + Gate high active

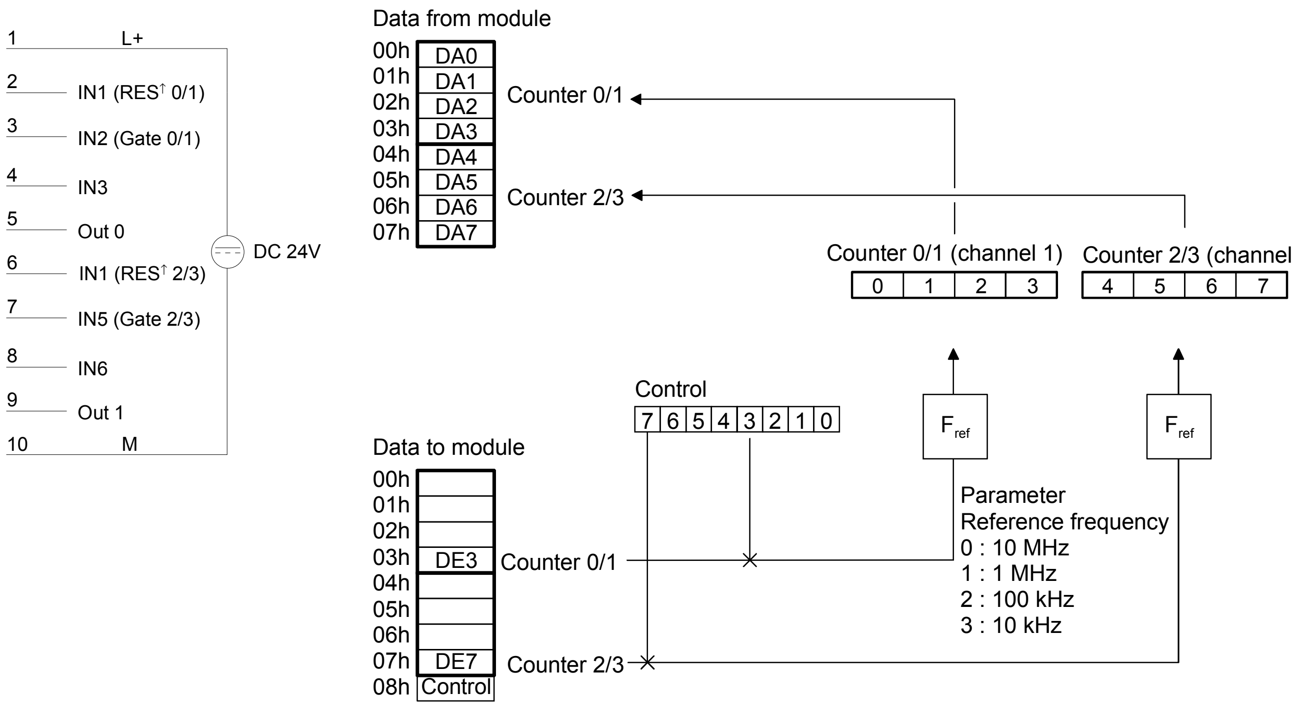
Mode 42 - 32Bit Counter down + Gate high active

The modes 39 to 42 allow you to realize a 32Bit counter for each channel that is controlled via a low or high active gate signal (gate) and counts with an internal reference frequency.

The direction of counting depends on the selected mode. With rising edge of the pulse frequency the counter is incremented res. decremented for 1.

The rising edge of the signal RES↑ resets the counter.
Fref is programmable.
The OUT signal is not changed.

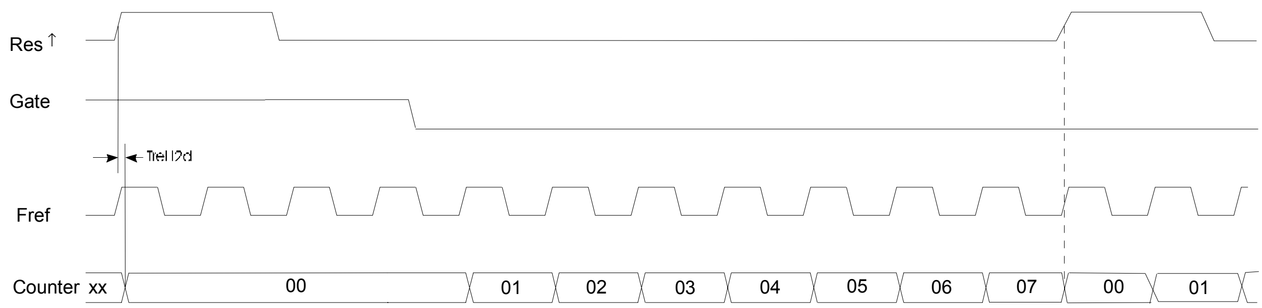
Pin assignment access to counter



Mode 39

32Bit Counter up + Gate low active

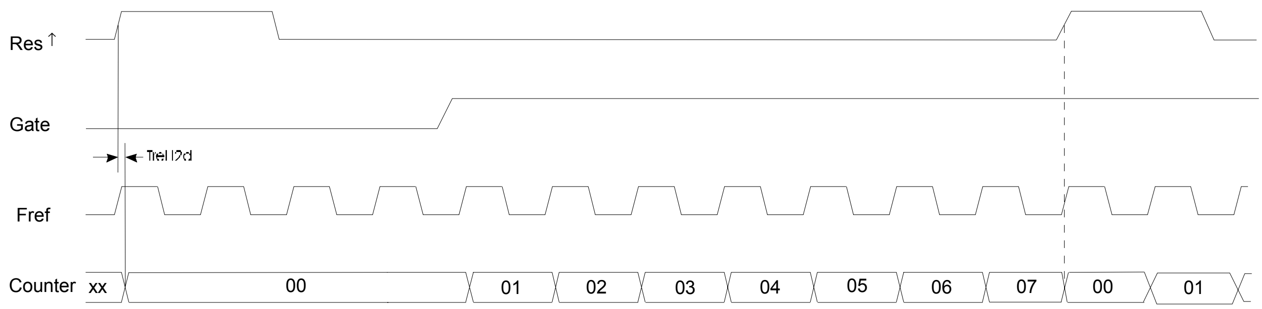
The rising edge of the signal RES↑ resets the counter. With the signal gate "0", the counter counts up with Fref.



Mode 41

32Bit Counter up + Gate high active

The rising edge of the signal RES↑ resets the counter. With the signal gate "1", the counter counts up with Fref.



Technical data

Counter module
FM 250

Electrical data	VIPA 250-1BA00
Number of counters	2 res. 4
Counter resolution	32Bit res. 16Bit
Number of operating modes	43
Counter frequency	max. 1MHz
Current consumption	80mA via backplane bus
Isolation	yes
Output stage	DC 24V high side switch 1A
Ext. power supply	DC 24V (18 ... 28.8V)
Signal voltage "0"	-30 ... 5V
Signal voltage "1"	13 ... 30V
Status indicator	via LEDs at the front
Programming specifications	
Input data	10Byte
Output data	10Byte
Parameter data	2Byte
Diagnostic data	-
Dimensions and weight	
Dimensions (WxHxD)	25.4x76x78mm
Weight	100g

Chapter 5 FM 253 - MotionControl Stepper

Outline

This chapter contains information about the installation, the data transfer and the operating modes of the MotionControl Stepper module for stepper motors.

The following text describes:

- Installation
- Parameterization
- Data transfer
- Technical data

Content

System overview

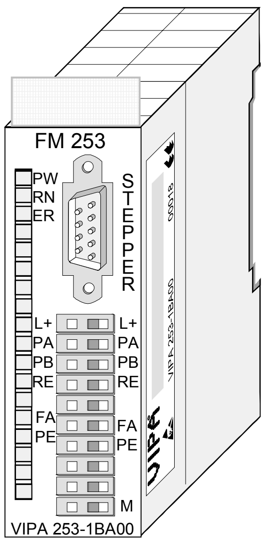
Properties

The FM 253 is a positioning module for controlling a stepper motor. The modules may be used for point-to-point positioning as well as for complex drive outlines with need for a high level of precision, dynamics and speed.

Stepper motors are employed where a maximum torque at low rotational speed is required and the target position shall be reached and held without overshoot.

The module works independently and is controlled via an according user application at the CPU. The module has the following characteristics:

- Microprocessor controlled positioning module for controlling a 1axis drive with stepper motor.
- Operating round and linear axis
- different operating modes
- the parameterization data is stored in the internal Flash memory. There is no battery required.
- the module contains 3 inputs for connecting end switches and is able to control 2 outputs. The states of the in-/outputs are additionally shown via LEDs.

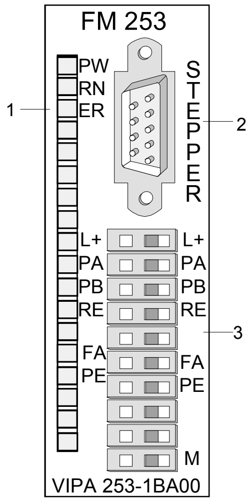


Order data

Type	Order number	Description
FM 253	VIPA 253-1BA00	MotionControl Module Stepper

Structure

Front view



[1] LED Status monitor
[2] Plug for drive
[3] Connection for supply voltage, end switch and outputs

Components

LEDs

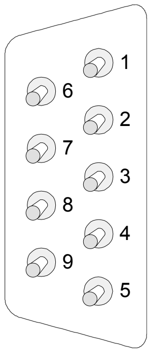
The FM 253 has some LEDs at the front used for status monitoring.

The usage and the according colors of these LEDs are shown in the following table:

Label	Color	Description
PW	Yellow	DC 24V supply voltage is applied
RN	Green	RUN: control active
ER	Red	Internal error
L+	Yellow	DC 24V supply voltage for outputs is applied
PA	Green	Limit value A overrun, input PA is set
PB	Green	Limit value B overrun, input PB is set
RE	Green	Reference point overrun
FA	Green	Drive in run
PE	Green	Drive reached position

Stepper interface

Via this interface your stepper motor is connected. The interface appears as 9pin D-type-plug and works with RS422 level. It has the following pin assignment:

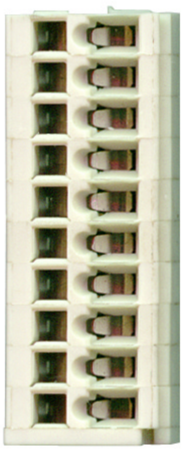


9pin D-type-plug

Pin	Assignment
1	PULSE_P: (+) pulse output
2	DIR_P: (+) direction signal
3	reserved
4	reserved
5	GND: ground
6	PULSE_N: (-) pulse output
7	DIR_N: (-) direction signal
8	reserved
9	reserved

Control interface

The control interface provides connection possibilities for end switches and output elements. The interface has the following pin assignment:



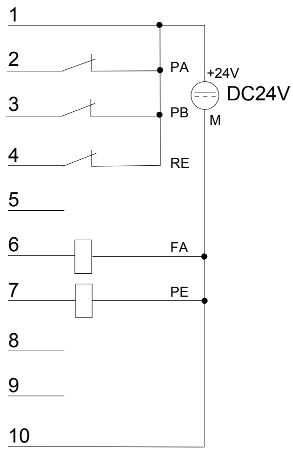
Pin	Assignment
1	Supply voltage DC 24V for outputs
2	Input: end switch PA
3	Input: end switch PB
4	Input: reference switch
5	reserved
6	Output: axis in motion
7	Output: position reached
8	reserved
9	reserved
10	Ground 24V

Connecting a drive

Connection stepper motor

The connection of a stepper motor is exclusively via the stepper interface.

Connection of supply voltage, end switch and output units



Voltage supply

The module itself is provided via the back plane bus. The deployment of the integrated digital outputs requires an additional voltage supply. The connection of an additional DC 24V supply voltage takes place via the clamps 1 and 10 of the control interface.

Inputs for end switches

You may connect up to 3 end switches (opener) to the module.

At terminals 2 and 3 (PA and PB) you connect the end switches with which you limit the distance. As soon as one of these switches is operated, the drive is stopped immediately and may only be driven into the other direction.

Terminal 4 is for the connection of the reference switch which is responsible for the tuning with the FM 253 module.

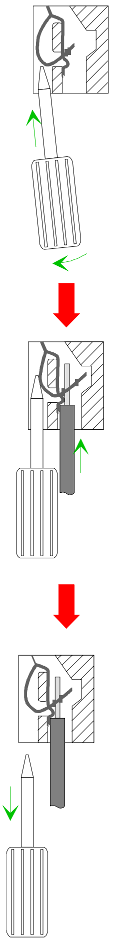
Outputs

The module contains 2 outputs that are only controlled by the module:

- FA - drive in run (clamp 6)
- PE - drive reached position (clamp 7)

The states of the outputs are shown via the according LEDs.

Cabling



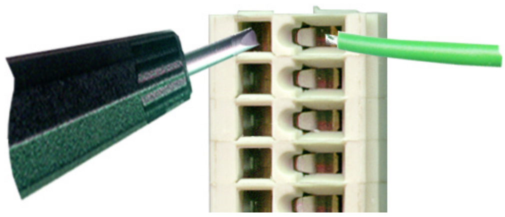
The end switches and the outputs are to connect at the control interface. Herefore a 10pin plug with CageClamp technology from WAGO is used. The cabling with CageClamps is very fast and in opposite to screw connections vibration secure.

You may connect cores with a core cross-section from 0.08mm^2 up to 1.5mm^2.

The cabling is analog to the big CageClamps of the System 200V.

Push the spring in the square opening with a fitting screwdriver more inside and insert the core into the rectangular opening.

By releasing the screwdriver the core is securely fixed.



Data transfer >> FM 253

Drive data

The MotionControl Stepper module fetches a data block from the CPU cyclically and analyzes it.

The data block has a length of 16Byte and the following structure:

Byte No.	Content	Length
0-3	Scheduled position	4Byte
4-7	Scheduled frequency	4Byte
8-9	Reserved	
10	Mode	1Byte
11	Index	1Byte
12-15	Variable parameters	4Byte

Via the Mode-Byte the contents of the data block are specified. The following functions may be initiated via the MODE-Byte:

Mode (Byte 10)

Bit 7 ... 0	Preset in Byte	Response in Byte
00: Idle-Mode - no status change of the drive, serves for parameter changes	-	-
01: Positioning relative - driving the preset number of steps	0-3: rel. set position	-
02: Reference run - calibration of the drive	15: Parameter bits	-
03: Permanent run axis - drive runs with scheduled frequency	4-7: set frequency	-
04: Read inputs - responds with the end switches states	-	15: State
05: Motor parameters - transmits parameters depending on index	11: Index, 12-15: Parameter	-
06: Set position - sets the recent position in the module without moving the drive	0-3: Set position	-
07: Delete error - deletes the error bit activated with 1	14-15: Error bit	-
08: Positioning absolute - drive to scheduled position	0-3: abs. set position	-

Parameter transfer (Mode = 05h)

Via **Index (Byte 11)** you set the parameter which value may be predefined via **Byte 12-15**. The value is transferred to the module by setting the **Mode 05h in Byte 10**.

More detailed information follows below.

Parameterization

Overview

The parameter data is transferred to the module together with the drive data in the 16Byte sized data block. For the parameterization you type the parameter to change in the **Index-Byte (Byte 11)** via the **Index-No.**.

The new value is fixed in **Byte 12-15**.

As soon as you set the **Mode-Byte (Byte 10)** to **05h**, the parameter is transferred to the module.

Please regard, that new parameters are only taken over when there has been a mode change before. For this you switch into the IDLE-Mode (MODE-Byte 10 = 00h) after every parameter transfer.

Store parameters in the Flash

The parameters that you transfer to the module are stored in the RAM. As long as the module is supplied with voltage, the parameters are preserved. Via the index no. 61h you also have the possibility to store the parameters in the internal Flash.

So the parameters are available again after PowerOn.

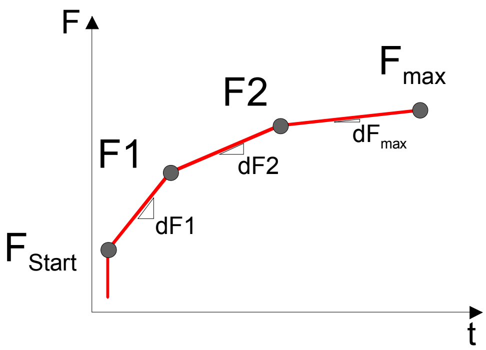
Parameterization via FCs

You get FCs from VIPA that should make the deployment of the FM 253 easier. For example you may parameterize your module via the FCs 201 and 202.

The control of the drive functions via FC 200. Via this FC you may access all modes except "Set parameters".

Context of the parameters

The following illustration shows the important contexts of the parameters. The assignment of the according index no. is to find in the table below.



Set index at parameter

Via the index no. you fix the parameter in Byte 11, where the value may be preset in Byte 12-15.

Index	Parameter	Unit	Value range	Default	Description
00h	Fstart	Hz	UINT32	200	Start frequency (min. 125Hz)
01h	F1	Hz	UINT32	4000	Limit frequency 1
02h	dF1	Hz	UINT32	100	Acceleration of Fstart ⇒ F1
03h	F2	Hz	UINT32	10000	Limit frequency 2
04h	dF2	Hz	UINT32	60	Acceleration of F1 ⇒ F2
05h	Fmax	Hz	UINT32	15000	Maximum drive frequency (max. 25 000Hz)
06h	dFmax	Hz	UINT32	40	Acceleration of F2 ⇒ Fmax
07h	Fpos	Hz	UINT32	15000	Frequency at positioning
08h	Fref	Hz	UINT32	1000	Frequency for reference run
09h	steps	Hz	UINT32	10	Steps between calculation frequency (min. 10)
0Ah	Fist	Hz	UINT32	-	Recent motor frequency (read only)
0Bh	Fsoll	Hz	UINT32	-	Recent set frequency (read only)
0Dh	FTarget	Hz	UINT32	-	Target frequency (read only)
61h				-	Store parameters in Flash
62h				-	Read parameters from Flash (State like after PowerON)
63h				-	Load default parameters

Note!

When setting parameters for the drive, you should remember the following rules:

- **dF1** should always be **smaller** than **Fstart**
- **dF2** should be the **half** of **dF1**
- **dFmax** should be the **half** of **dF2**

For this the following context appears:

$$\boxed{4 \cdot dF_{max} = 2 \cdot df2 = dF1 < F_{Start}}$$

Wrong inputs are partly corrected by the firmware of the module.

Operating modes

Overview

By setting according bits in the "MODE"-Byte you may set the following operating modes described below:

- IDLE-Mode
- Positioning relative / absolute
- Permanent run
- Set position
- Reference run

IDLE-Mode

Default: Byte 10 = 00h

In the IDLE-Mode no state change of the drive occurs. For new data is only taken over by the module after an state change, you may initiate a mode change by jumping into the IDLE-Mode and back again.

Via the IDLE-Mode you may e.g. start a new order, for a mode change is recognized by the jump into the IDLE-Mode.

The operating mode IDLE should always be called when no action shall be initiated. For initiating an action you normally branch into another mode only for a short time and switch then back to the IDLE-Mode.

Positioning relative

Default: Byte 10 = 01h, Byte 0-3 = relative set position

At the relative positioning a predefined number of steps is added to the recent position and then approached.

Herefore you have to predefine the position offset (number of steps) as relative scheduled position in Byte 0-3 and then set the **Mode (Byte10) to 01h.**

By setting the Byte 10 to 01h the relative positioning starts.

For acceleration and frequency of the drive, the values set in the parameters are used. If there are no presetting, the default values are used.

As long as the drive is operating, the output **"Axis in run"** is set. After reaching the position this output is cleared and the output **"Position reached"** is set.

Positioning absolute

Default: Byte 10 = 08h, Byte 0-3 = absolute set position

At the absolute positioning an absolute scheduled position is approached.

Herefore you have to predefine the position (number of steps) as absolute scheduled position in Byte 0-3 and then set the **Mode (Byte 10) to 08h**.

By setting the Byte 10 to 08h the absolute positioning starts.

continued ...

... continue

For acceleration and frequency of the drive, the values set in the parameters are used. If there are no presetting, the default values are used. As long as the drive is operating, the output **"Axis in run"** is set. After reaching the position this output is cleared and the output **"Position reached"** is set.

Permanent run

Default: Byte 10 = 03h, Byte 4-7 = Scheduled frequency (125 ... 25 000Hz)

If the nominal frequency is less than 125Hz, it will be reset. The frequency Fmax limits the maximum of nominal frequency.

At permanent run the axis rotates with the set frequency until it is changed.

Herefore you have to predefine the rotational speed as set frequency in Byte 4-7 and then set **Mode (Byte10) to 03h.**

By setting Byte 10 to 03h the drive starts and rotates with the given frequency until a new frequency value is set.

A new frequency is only taken over at mode change. This is reachable by changing into the IDLE-Mode (Byte 10 = 00h) after the start-up of the drive. Now type the new scheduled frequency and set Byte 10 back to 03h. The drive is set to the new frequency immediately.

For acceleration of the drive, the values set in the parameters are used. If there are no presetting, the default values are used.

As long as the drive is operating, the output **"Axis in run"** is set. By presetting 00h as scheduled frequency (mode change required) the drive stops and the output is set back.

Stop drive by permanent run and set frequency = 00h

By setting a scheduled frequency of 00h in Byte 4-7 and the mode 03h in Byte 10 you may stop the drive at any time.

Note!

Please regard, that a frequency change is only recognized by the module via a mode change. This is also valid for stopping the drive. For a mode change, use the short time jump to the IDLE-Mode.

Set position

Default: Byte 10 = 06h, Byte 0-3: Position value

In the operating mode "Set position" you may assign a new value to the recent actual value.

Herefore you predefine the new value in Byte 0-3 and then set the Mode-Byte 10 to 06h.

Reference run

Default: Byte 10 = 02h, Byte 15 = Control bits for reference run

The reference run supports the calibration of your drive system. The reference point should be inside the drive outline.

Before starting a reference run you have to specify the type of the reference run and the direction to run to in Byte 15.

By setting Byte 10 to 02h, the drive starts with its reference run.

As frequency the reference frequency set in the parameters are used. If there are no parameters, the default values are used.

There are 6 different possibilities for the reference drive that are predefined via Byte 15:

- Reference run to reference switch and delete position counter
- Reference run to reference switch and keep position counter
- Reference run to end switch B and delete position counter
- Reference run to end switch B and keep position counter
- Reference run to end switch A and delete position counter
- Reference run to end switch A and keep position counter

Control bits for the reference run

The control bits in Byte 15 have the following assignment:

Byte 15	Parameter
Bit 0	1: Direction forward 0: Direction backward
Bit 1	1: delete position after reference run 0: keep position after reference run
Bit 2	Reference run to reference switch
Bit 3	Reference run to end switch B
Bit 4	Reference run to end switch A

Note!
When starting a reference run, please regard, that you always have to set a direction via Bit 0 and that you may set only one bit in the Bits 2 ... 4!

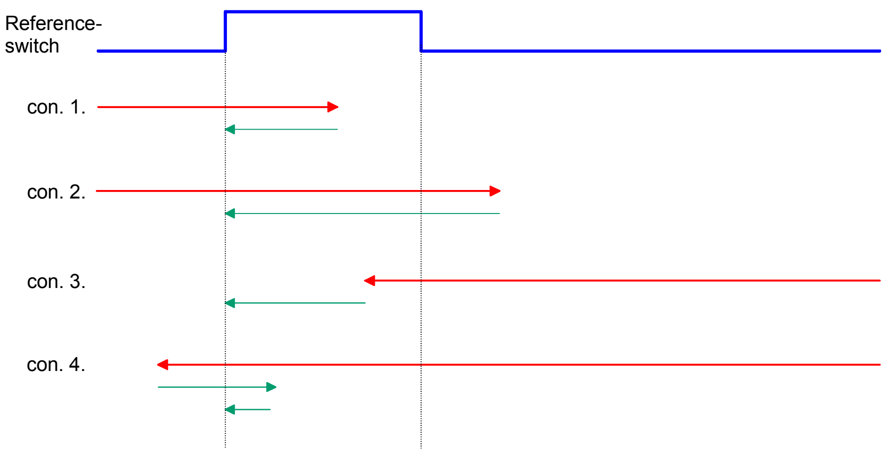
Reference run to reference switch

The reference run starts always with the speed predefined in FREF. The direction has to be preset in the variable parameter (Byte 15, Bit 0). As soon as the ascending edge of the reference switch is recognized, the motor slows down to FSTART.

Depending on the reference speed the drive may overrun the reference switch or not during slow down.

The following 4 drives to the reference switch are possible:

1. Motor comes from the left side, slows down inside the reference switch and drives backward with FSTART until the descending edge of the reference switch is recognized.
2. Motor comes from the left side, overruns the reference switch during slow down and drives backward with FSTART over the ascending edge until the descending edge of the reference switch is recognized.
3. Motor comes from the right side, slows down inside the reference switch and drives with FSTART until the descending edge of the reference switch is recognized.
4. Motor comes from the left side, overruns the reference switch during slow down, it changes the rotational direction and drives with FSTART until the ascending edge of the reference switch is recognized, switches the direction again and drives on until the descending edge of the reference switch is recognized.



Reference run to end switch

You may limit your distance via the end switches A and B.

At the reference run to end switch the drive starts and drives with the preset speed FREF and the predefined rotational direction until the according end switch gets active, stops abruptly, changes its rotational direction and drives with FSTART until the end switch is inactive again.

Note!

If you use the reference run to end switch, you have to regard, that there is enough space behind the end switch for the motor to slow down!

Data transfer >> CPU

Respond message

The MotionControl Stepper module sends a data block to the CPU cyclically that contains several information about the recent state of the drive. The data block has a length of 16Byte and the following structure:

Byte no.	Content	Length
0-3	actual position	4 Byte
4-7	actual frequency	4 Byte
8-9	error messages	2 Byte
10	actual mode	1 Byte
11	state	1 Byte
12-15	data of variables	4 Byte

Actual position, actual frequency

Via this two parameters the actual position and frequency of your drive is always shown.

Error messages

The recently recognized errors are monitored via the error bits of Byte 8-9. The errors remain active until the according Bits are set back.

As long as an error is still valid, the according error bit is set again after the reset.

The following error messages are used:

Error byte (Byte 8-9)

Byte 9	Description
Bit 0	Error in the internal state administration
Bit 1	System has been booted (always after PowerON)
Bit 2	Error at proofing Flash parameters, motor parameters not valid
Bit 3	This function is not permitted during motor run
Bit 4	Motor is recently blocked
Bit 5	Error at positioning the motor
Bit 6	End switch is/was active
Bit 7	Frequency has been limited to FMAX
Byte 8	
Bit 0	General error at the motor
Bit 1	Fstart < 125Hz
Bit 2	Fmax > 25 000Hz
Bit 3	Value of steps < 10

Set back error messages

For deleting an active error (Byte 8-9) you have to set the according error bit in the variables parameter (Byte 14-15) to "1".

As soon as you set the **Mode (Byte 10**) to **7**, the according errors in the module are set back. You may also set back several error messages at the same time. FFFFh in Byte 14-15 for example sets back all errors.

Recent mode

Here you always find the mode that your FM 253 has at the moment.

The following modes may be shown:

Mode (Byte 10)

Byte	Mode
10	00h: IDEL 01h: Positioning relative 02h: Reference run 03h: Permanent run axis 04h: Read inputs 05h: Change motor parameters 06h: Set position 07h: Delete error 08h: Positioning absolute

State

The STATE-Byte shows you the state of the drive.

The following state messages may be shown:

State (Byte 11)

Byte 11	State
Bit 0	1: Drive in run 0: Drive in stop
Bit 1	1: Direction forward 0: Direction backward
Bit 2	1: Drive in position 0: Drive not in position

Read inputs

For reading the inputs, the **Mode (Byte 10)** is set to **4** and now the module shows the state of the end switches and the reference switch in the variables data (Byte 15).

Inputs (Byte 15)

Byte 15	Input
Bit 0	State PA end switch (1: operated, 0: not operated)
Bit 1	State PB end switch (1: operated, 0: not operated)
Bit 2	State RE reference switch (1: operated, 0: not operated)

Handling blocks

Overview

There are different handling blocks available with the FM 253 to make the usage of the module more comfortable. The following handling blocks are available for the FM 253 at this time:

Block	Description
FC 200	Control drive
FC 201	Adjustment of a parameter
FC 202	Adjustment of all drive parameters (Index 0...9)

FC 200 Control drive

This FC serves the control of your drive by transferring the drive data to the module through setting the according mode.

With this FC you may transfer all modes except "Set parameters" and the according parameters to the module.

Data transfer to FM 253 with SET_MODE = 1

- Set the mode.
- Give data to the according parameters.
- Start the transfer by setting SET_MODE to 1. When the mode is started, the module SET_MODE is set back at the next cycle and shows the actual data of the FM 253.

Data transfer to CPU with SET_MODE = 0

At the call of the FC 200 with SET_MODE = 0, the actual data of the FM 253 is shown via the labels ACT_POSITION, ACT_FREQUENCY, ACT_MODE, ERROR, STATE and VAR_DATA.

It is convenient to store the single values in a data block. In the following example we used DB5 for this purpose.

Parameters

Parameter	Declaration	Data Type	Description
ADDRESS	IN	INT	Set basic address
SOLL_POSITION	IN	DINT	Transfer position values
SOLL_FREQUENCY	IN	DINT	Transfer frequency at permanent run
VARIABLES	IN	DWORD	Transfer variables at reference run
MODE	IN	INT	Transfer mode to change
ACT_POSITION	OUT	DINT	Response actual position
ACT_FREQUENCY	OUT	DINT	Response actual frequency
ERROR	OUT	INT	Error word
ACT_MODE	OUT	INT	Response actual mode
STATE	OUT	BYTE	Response status bits
VAR_DATA	OUT	DWORD	Response variables
SET_MODE	IN_OUT	BOOL	Start function

ADDRESS

Start address from where on the FM 253 is stored in the CPU.

SET_POSITION

In mode 01, 06 and 08 you fix the scheduled position for the drive here.

SET_FREQUENZ

In mode 03 you fix the scheduled rotational speed as set frequency.

VARIABLES

Fix here the control bits for the reference run (MODE = 02) and for setting the errors back (MODE = 07).

The control bits for the reference run have the following assignment:

Control bits

VARIABLE-Byte	Parameter
Bit 0	1: Direction forward 0: Direction backward
Bit 1	1: after reference run delete position 0: after reference run keep position
Bit 2	Reference run to reference switch
Bit 3	Reference run to end switch B
Bit 4	Reference run to end switch A

An overview over the error-bit-assignment follows below.

MODE

With this parameter you transfer the mode to the FM 253. The following modes are possible:

Mode

Value	Description	Default in	Response in
00	Idle-Mode - no status change of the drive, serves for parameter changes	-	-
01	Positioning relative - driving the preset number of steps	SET_POSITION	-
02	Reference run - calibration of the drive	VARIABLES	-
03	Permanent run axis - drive runs with scheduled frequency	SET_FREQUENCY	-
04	Read inputs - responds with the end switches states	-	VAR_DATA
06	Set position - sets the recent position in the module without moving the drive	SET_POSITION	-
07	Delete error - deletes the error bit activated with 1	VARIABLES	-
08	Positioning absolute - drive to scheduled position	SET_POSITION	-

ACT_POSITION, ACT_FREQUENCY

Via those parameters the recent actual position and actual frequency of your drive is shown.

ERROR

Here you may find error messages if occurred. The errors remain active until the error cause is removed and the according bits are set back. The following error messages may occur:

Error messages

ERROR-Byte 1	Description
Bit 0	Error in the internal state administration
Bit 1	System booted (always after PowerON)
Bit 2	Error at validating the Flash parameters, motor parameters not valid
Bit 3	Function is not available during motor run
Bit 4	Motor is blocked
Bit 5	Error at positioning the motor
Bit 6	End switch is/was active
Bit 7	Frequency has been limited to FMAX
ERROR-Byte 0	
Bit 0	General error at the motor
Bit 1	Fstart < 125Hz
Bit 2	Fmax > 25 000Hz
Bit 3	Value of steps < 10

The clearing of the error messages takes place via MODE = 07 and VARIABLE = Error bytes.

ACT_MODE

Responds the mode in which the module is at this moment.

STATE

The STATE-Byte shows you information about the state of the drive. The following state messages may occur:

State

STATE-Byte	State
Bit 0	1: Drive in run 0: Drive in stop
Bit 1	1: Direction forward 0: Direction backward
Bit 2	1: Drive in position 0: Drive not in position

VAR_DATA

In VAR_DATA the state of the inputs is returned after you requested this by MODE = 04. For reading the inputs the **Mode 4** is set and now the module shows the state of the end switches and the reference switch in the variables data (Byte 15).

Inputs

VAR_DATA-Byte	Input
Bit 0	State PA end switch (1: operated, 0: not operated)
Bit 1	State PB end switch (1: operated, 0: not operated)
Bit 2	State RE reference switch (1: operated, 0: not operated)

SET_MODE

After you defined the according parameters the data is transferred to your module via SET_MODE = 1.

When the mode has been started, the module sets back again the SET_MODE in the next cycle and returns the actual data of the FM 253.

Example

```
DB 5
DBD  0 Position        DINT    L#0      Position value
DBD  4 Frequency       DINT    L#0      Frequency for permanent run
DBW  8 reserve         WORD    W#16#0
DBW 10 MODE            INT     0        Mode
DBW 12 Index           INT     0        Index default
DBD 14 Variable_PARAM  DWORD   DW#16#0  Var. for Ref.run/Param...
DBW 18 Reserve1        WORD    W#16#0
DBD 20 Act_Position    DINT    L#0      actual position
DBD 24 Act_Frequency   DINT    L#0      actual frequency
DBW 28 Error           INT     0        error monitor
DBW 30 ACT_Mode        INT     0        actual mode
DBW 32 State           BYTE    B#16#0   State response
DBD 34 VAR_DATA        DWORD   DW#16#0  Return parameter/data

CALL  FC 200                  //FC for Stepper module
    ADDRESS      :=128        //Module address
    SET_POSITION :=DB5.DBD 0  //DBD with position for abs/rel
    SET_FREQUENCY:=DB5.DBD 4  //DBD with frequency for permanent run
    VARIABLES    :=DB5.DBD14  //Delete data for Ref_Run/Del error
    MODE         :=DB5.DBW10  //Mode default for new order
    SET_MODE     :=M1.0       //Start order
    ACT_POSITION :=DB5.DBD20  //actual position
    ACT_FREQUENCY :=DB5.DBD24 //actual frequency
    ERROR        :=DB5.DBW28  //Monitor error
    ACT_MODE     :=DB5.DBW30  //actual mode
    STATE        :=DB5.DBW32  //State bits from module
    VAR_DATA     :=DB5.DBD34  //Return of values
                                e.g. read inputs
```

FC 201 - set a parameter

With the FC 201 it is possible to set a parameter at the FM 253.

Parameter

Parameter	Declaration	Data Type	Description
ADDRESS	IN	INT	Fixed basic address
INDEX	IN	INT	Transfer INDEX for parameters
PARAMETER	IN	DWORD	Transfer parameter value
	OUT		
SET_PARA	IN_OUT	BOOL	Start parameter transfer

ADDRESS

Start address from where on the FM 253 is stored in the CPU.

INDEX

Via INDEX you fix the parameter where the value is set in PARAMETER.

Index	Parameter	Unit	Value range	Default	Description
00h	Fstart	Hz	UINT32	200	Start frequency
01h	F1	Hz	UINT32	4000	Limit frequency 1
02h	dF1	Hz	UINT32	100	Acceleration from Fstart ⇒ F1
03h	F2	Hz	UINT32	10000	Limit frequency 2
04h	dF2	Hz	UINT32	60	Acceleration from F1 ⇒ F2
05h	Fmax	Hz	UINT32	15000	Maximum drive frequency (max. 25 000Hz))
06h	dFmax	Hz	UINT32	40	Acceleration from F2 ⇒ Fmax
07h	Fpos	Hz	UINT32	15000	Frequency at positioning
08h	Fref	Hz	UINT32	1000	Frequency for reference run
09h	steps		UINT32	10	Steps between calculation frequency (min. 10)
0Ah	Fist	Hz	UINT32	-	Recent motor frequency (read only)
0Bh	Fsoll	Hz	UINT32	-	Recent set frequency (read only)
0Dh	FTarget	Hz	UINT32	-	Target frequency (read only)
61h				-	Store parameter in Flash
62h				-	Read parameter from Flash (state like after PowerON)
63h				-	Load default parameters

PARAMETER

Here you type the value of the parameter specified via INDEX.

SET_PARA

After you filled the according parameters, the parameter is transferred to your module via SET_PARA = 1. After the transfer SET_PARA is set back automatically.

Parameterize FC 202 - FM 253

Via the FC 202 you may adjust all relevant parameters of the FM 253.

Parameter

Parameter	Declaration	Data Type	Description
DATA_DB	IN	BLOCK_DB	Data block with parameters
ADDRESS	IN	INT	Module address
	OUT		
START	IN_OUT	BOOL	Start parameter transfer
RUN	IN_OUT	BOOL	Transfer single runs

DATA_DB

Please fix here the data block where your parameters are stored.

The DB has the following structure:

```
DBD  0 Fstart        DINT   L#0  Start frequency
DBD  4 F1            DINT   L#0  Limit frequency 1
DBD  8 F2            DINT   L#0  Limit frequency 2
DBD 12 Fmax          DINT   L#0  Maximum drive frequency
DBD 16 dF1           DINT   L#0  Acceleration Fstart --> F1
DBD 20 dF2           DINT   L#0  Acceleration F1 --> F2
DBD 24 dFmax         DINT   L#0  Acceleration F2 --> Fmax
DBD 28 Fpos          DINT   L#0  Frequency at positioning
DBD 32 Fref          DINT   L#0  Frequency at reference run
DBD 36 StepRepeat    DINT   L#0  Step between frequency calculation
```

ADDRESS

Start address from where on the FM 253 is stored in the CPU.

START

After you created the DB you may transfer your parameters to your module via START = 1.

As soon as all parameters are transferred, START is set back again.

RUN

This variable stores one cycle spreading state and it is responsible for the single parameter transfer.

Technical data

MotionControl Stepper FM 253

Electrical data	VIPA 253-1BA00
Number of axis	1
Voltage supply	DC 24V (20.4 ... 28.8) via front from ext. power supply
Current consumption backplane bus	typ. 330mA, max. 500mA
Status monitor	via LEDs at the frontside
Connectors / Interfaces	
"Drive"-Interface	Output for pulse, direction and release with RS422
Digital inputs	
Number	3
Function	2 end switch, reference switch
Signal voltage "0"	0 ... 5V
Signal voltage "1"	15 ... 28.8V
Digital outputs	
Number	2
Function	"axis in motion", "position reached"
Output current	1A protected against sustained short circuits
Potential separation	yes
Max. Impulse frequency	25 000Hz
Min. Impulse frequency	125Hz
Width of pulse	
125 ... 10 000Hz	>10µs
10 001 ... 25 000Hz	>5µs
Programming data	
Input data	16Byte
Output data	16Byte
Dimensions and weight	
Dimensions (WxHxD) in mm	25.4x76x78
Weight	80g

Chapter 6 FM 254 - MotionControl Servo

Outline

This chapter contains information about the installation, the data transfer and the operating modes of the MotionControl modules for stepper motors or servo drives.

The following text describes:

- Installation
- Parameterization
- Data transfer
- Technical data

Content

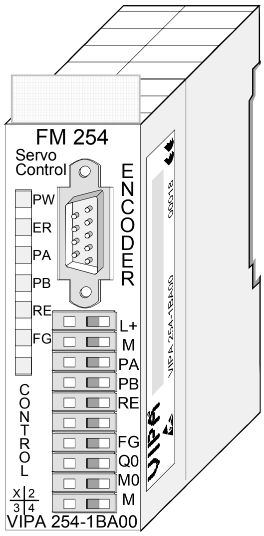
System overview

Properties

The FM 254 is a positioning module for controlling a servo drive. The modules may be used for point-to-point positioning as well as for complex drive outlines with need for a high level of precision, dynamics and speed.

The module works independently and is controlled via an according user application at the CPU. The module has the following characteristics:

- Microprocessor controlled positioning module for drives with an analog set point interface (±10V control voltage)
- different operating modes
- closed-loop position control
- the module contains 3 inputs for connecting end switches and is able to control 2 outputs. The states of the in-/outputs are additionally shown via LEDs.



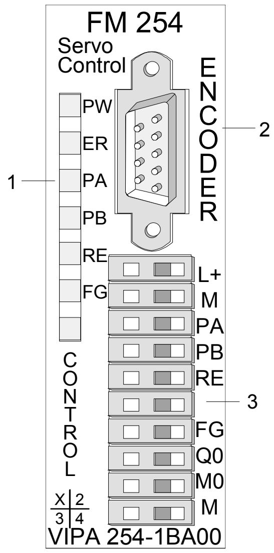
Note!
The explanations on this module refer to a firmware version beginning at 111. You can find the firmware version on a label on the backside of the module.

Order data

Type	Order number	Description
FM 254	VIPA 254-1BA00	MotionControl Module Servo

Structure

Front view



[1] LED status indicators
[2] Encoder interface
[3] Connector for supply voltage, drive, end switch and outputs

Components

LEDs

The positioning module FM 254 has status indicator LEDs.

The following table contains the description and the respective color of these LEDs.

Label	Color	Description
PW	Yellow	24V DC supply voltage is applied
ER	Red	internal error
PA	Green	Limit value A overrun, input PA is set
PB	Green	Limit value B overrun, input PB is set
RE	Green	Reference point overrun
FG	Green	Drive released
	Yellow	Blinking always when supply voltage is applied (Heartbeat - 2Hz)

Note!

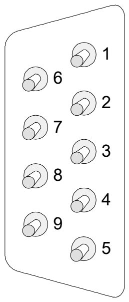
If the PW-LED is not on during operation, this may depend on a short circuit in the DC 24V voltage supply.

Please control also the connections of the encoder plug.

If the LED remains off even when you disconnect the encoder plug, the module has a defect.

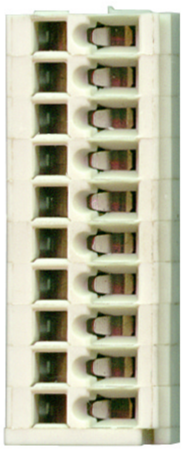
Encoder interface

9pin D-type plug



Pin	Assignment
1	+24V encoder power
2	+5V encoder power
3	R+ clock input null pulse
4	B+ clock input
5	A+ clock input
6	Ground encoder power
7	R- clock input null pulse
8	B- clock input
9	A- clock input

Control interface



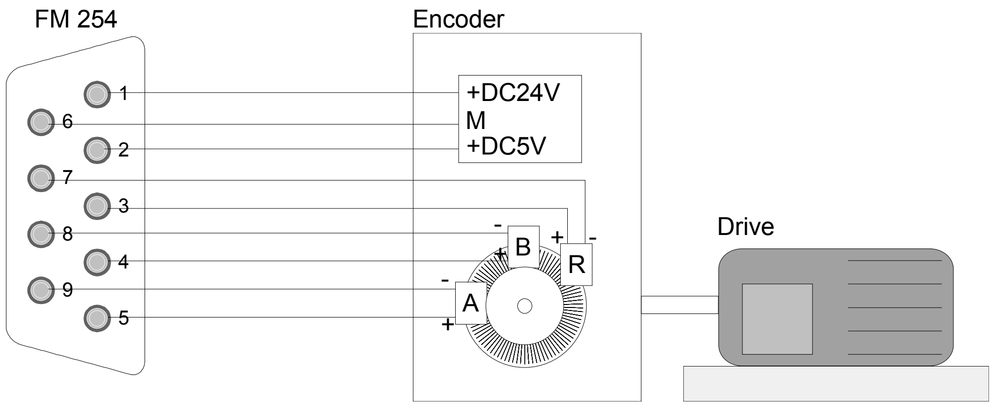
Pin	Assignment
1	DC 24V supply voltage
2	Ground 24V
3	Input for start switch (low active)
4	Input for end switch (low active)
5	Input for reference switch (low active)
6	reserved
7	Output regulator release
8	Analog output ground
9	Analog output +
10	Screen

Connecting a drive with encoder

Connection of an encoder via encoder interface

The encoder is wired to the 9pin D-type connector located at the front. The module supplies the encoder with the required DC 24V and DC 5V voltages.

The following figure shows the connection of an encoder:



Attention!
Please regard you only may attach encoder with 5V signal level (RS422).

Connection of supply voltage, drive, end switch and outputs via control interface

Power supply

The module requires a power supply of DC 24V via pins 1 and 2.

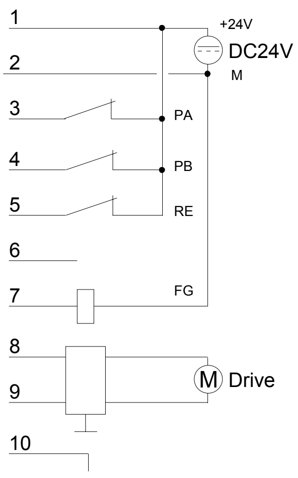
End switches

You may connect up to 3 end switches (opener) to the module.

The end switches for the extremes of the distance are connected to terminals 3 and 4. The drive will be stopped immediately as soon as one of these switches is operated. In this situation may only be driven into the opposite direction.

The reference switch is connected to terminal 5. This is required to tune the drive to the positioning module.

The end switch that stops the drive in the mode hardware-controlled run is also connected to terminal 5.



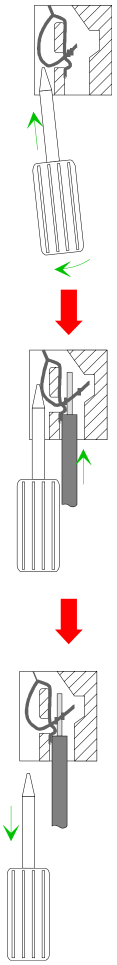
Outputs

The module has 2 outputs that are controlled directly by the module. At present, however, only the output "Controller Enable" (pin 7) is available. The second output is intended for future expansion. You enable the output by setting bit 0 in the traversing data.

Drive

Pin 8 and 9 supply an analog signal for ±10V regulator control.

Cabling



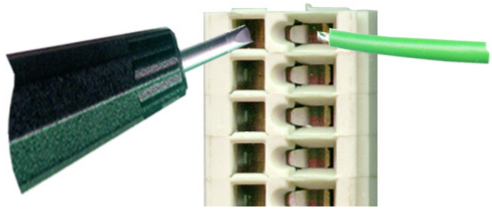
The drive and the end switches are to connect at the control interface. Herefore a 10pin plug with CageClamp technology from WAGO is used. The cabling with CageClamps is very fast and in opposite to screw connections vibration secure.

You may connect cores with a core cross-section from 0.08mm^2 up to 1.5mm^2.

The cabling is analog to the big CageClamps of the System 200V.

Push the spring in the square opening with a fitting screwdriver more inside and insert the core into the rectangular opening.

By releasing the screwdriver the core is securely fixed.



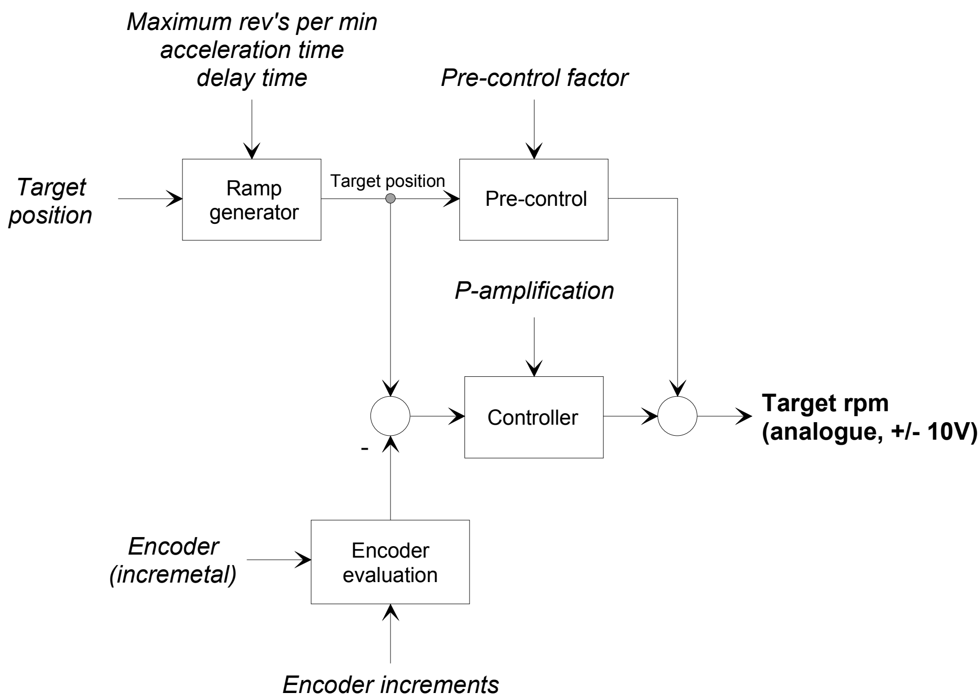
Summary of parameters and transfer values

The following table lists all the parameters and transfer values. A block diagram depicts the interaction between the parameters.

Overview

Value	Size	Unit	Physical range
Destination position Set position Actual position	32Bit	1	32 Bit Integer
Maximum rpm.	16Bit	1/min	100 ... 6000 1/min
Acceleration time Delay time	16Bit	10ms	20ms ... 30s
P-amplification	16Bit	0.1	0.0 ... 1000.0
Pre-control factor	16Bit	0.1	0.0 ... 1.0
Encoder increments	16Bit	1	10 ... 10000
Operating mode	16Bit	binary coding	

Block diagram



Parameterization

When commissioning the MotionControl Servo module it requires 16Byte of parameter data. These have the following structure:

Parameter data (write only)

Byte no.	Name	Length	Range	Unit
1, 0	Maximum rotat. speed	2Byte	10 ... 6000	1/min
3, 2	reserved	2Byte	-	-
5, 4	reserved	2Byte	-	-
7, 6	P-amplification	2Byte	0.0 ... 1000.0	0.1
9, 8	Pre-control factor	2Byte	0.0 ... 1.0	0.1
11, 10	Encoder increments	2Byte	10 ... 10000	1
13, 12	Reference rot. speed	2Byte	1 ... 6000	1/min
14	Pos. reached window	1Byte	0 ... 255	1INK
15	Drag distance	1Byte	0 ... 1020	4INK

Parameter description

Maximum rotational speed

Defines the maximum rotations for your drive.
If the product of maximum rotational speed and encoder increments is exceeding the value 3 000 000, the maximum rotational speed will be limited to the value of 3 000 000/encoder increments.

P-amplification, Pre-control factor

These values control the regulation properties.

Encoder increments

This parameter matches your MotionControl Servo module to the encoder.

Reference rotational speed

This value for the rotational speed is used for the reference run that is required by the MotionControl Servo module to re-acquire parameters for the control path.
If the rotational speed in the input byte 6 and 7 is smaller than the reference rotational speed, the rotational speed of the input byte 6 and 7 will be used as reference rotational speed.

Pos-reached-window

When the target position has been reached, this position is maintained by continuous control of the drive. The drive is never stopped.

You can specify a window by entering certain increments into the *Pos-reached-window.* These define the tolerance by which the actual value may differ from the target position before the drive is controlled, i.e. when the drive is stationary.

Drag distance

This parameter defines the drag error or the difference between the actual and the set value, which causes the drive to be stopped.
If the value is 0, the drag error supervision will be switched off.

Data transfer >> FM 254

Traversing data

The CPU can control the MotionControl Servo module by writing the following values into the FM 254 module:

Byte no.	Name	Length	Range	Unit
3, 2, 1, 0	Target position	4Byte	32 Bit Integer	Encoder increments
5, 4	Control bytes	2Byte		
7, 6	Rot. speed	2Byte	1 ... 6000	1/min
9, 8	Acceleration time	2Byte	2 ... 3000	10ms
13, 12 ,11, 10	Parameter field	4Byte		
15, 14	Field identifier	2Byte		

Control bytes (Byte 4 and Byte 5)

Byte	Bit 7 ... 0
4	Bit 0: Enable Bit 1: Operating mode reference run positive Bit 2: Operating mode reference run negative Bit 3: Operating mode hardware-controlled run positive Bit 4: Operating mode hardware-controlled run negative Bit 5: Operating mode incremental dimension Bit 6: Operating mode infinite incremental dimension Bit 7: Taking over target position
5	Bit 0: Reset counter at non-maintained command mode (edge 0 after 1 sets back the actual position to zero) Bit 1: Non-maintained command mode direction of rotation pos. Bit 2: Non-maintained command mode direction of rotation neg. Bit 3: Stop bit Bit 7 ... 4: reserved

Attention!

Bit Enable (Control byte 4, Bit 0)

At digital and analog outputs of FM 254 “0“ will be read-out if the release bit is reset during transmission.

After repeated setting of the release bit (Byte 4, Bit 0) the module is continuing the interrupted order if no new order is available.

Stop bit: (Control byte 5, Bit 3)

If the stop bit is set during transmission, the drive is stopped with the specified delay time. If the stop bit is reset during the delay, the stop order will be deleted and the order previously defined will be carried out. If the drive comes to rest, the target position will be equated with the actual position and the order will be finished.
Prior to deleting the stop bit, control bits (Byte 4 and 5) must be cancelled. Otherwise a new order will be placed.

Note!

Reset: (Control byte 5, Bit 0)

The reset of the counter may only be executed in the non-maintained command mode. During positioning mode the regulator would throw a drag error because of the jumping actual value.

Traversing data

Parameter field and Field identifiers (Byte 10 ... Byte 15)

You can send additional parameters with the traversing data to the MotionControl Servo module by specifying a field identifier. The parameters for the respective field identifier must be entered into the parameter field (Byte 10...13).

The FM 254 will use the default settings shown below if you do not transfer any field identifiers.

Field ident.	Description	Range	Unit	Default setting
FF01h	Software end switch (+)	32Bit Integer	Encoder increments	7FFF.FFFF
FF02h	Software start switch(-)	32Bit Integer	Encoder increments	8000.0001
FF03h	Rot. speed at non-maintained command mode	1 ... 6000 [1)]	1/min	Reference rotational speed
FF04h	Delay time	1 ... 10000	10ms	Acceleration time

1) If the rotational speed in control byte 6, 7 is smaller than the non-maintained mode rotational speed, the rotational speed of control byte 6, 7 will be used as non-maintained rotational speed.

Count frequency

If the counter frequency considerably exceeds the value of 200 000 pulse/s respectively 50 000 increments longer than 1ms, the drive will be stopped.

In operational state byte 11 bit 3 will be set for internal error.

The LED “internal module error” at the front side of the module will lighten.

By resetting the release bit you can delete the “internal error-bit”. The setpoint position will be equated with the actual position.

The operating modes are described below.

Operating modes

Overview

The following operating modes can be selected by setting the respective bit in the control byte:

- Positioning operation (positioning to an absolute target position)
- Reference run (system calibration)
- Hardware run (drive to reference switch)
- Incremental run (use addition to approach a relative target)
- Infinite incremental run (relative traversal without counter overflow)
- Non-maintained command mode

Positioning mode

Operation

During the positioning operation the absolute target position is only transferred to the FM 254, if the bit "Taking over target position" is set.

If a new position is specified with the enable bit set, the drive moves to the respective position ± POS-REACHED-WINDOW with the values that were previously specified for the rotational speed and the acceleration/delay and sets the "Position reached"-Bit. After transferring the parameters for the traversal, you can start the drive by setting the enable bit. During the traversal the module indicates the direction of rotation by setting bit 1 or 2 in Operating state byte 10.

Should the deviation between set and actual position exceed the window specified for the drag error, the positioning operation is terminated and the motor is stopped. The program is notified by means of an active drag error bit 0 in Operating state byte 11. You can clear the drag error bit by resetting the enable bit. This also sets the set position to the actual position.

The drive is also stopped if soft- or hardware switches are passed that terminate the traversal distance.

The operation can be continued at any time by setting the enable bit.

The acceleration/delay time can be modified before a new command is issued.

It is always possible to specify a new value for the rotational speed by modifying the traversing data. If the rotational speed is changed while movement is taking place, the new value is attained respecting the current acceleration/delay times.

Control bytes

The control bytes that you use to specify this operating mode are an integral part of the traversal data.

A general description of the traversal data is available on page 6-9.

Byte	Bit 7 ... 0
4	Bit 0: enable (drive is started) Bit 6 ... 1: 0 Bit 7: irrelevant
5	Irrelevant

Reference run

Operation

The reference run calibrates your drive system. The point of reference should be located on the path of traversal.

Start the reference run:

- Set the enable bit.
- Release the reference run by means of the bit "Reference run positive" or "Reference run negative".
  - → The drive will travel to the point of reference using the reference rotational speed specified in the parameter set.
  - → As soon as the point of reference is passed, the reference switch is operated (LED RE is turned off).
  - → The position of the point of reference is recorded in memory.
  - → The drive is reversed up to the next encoder zero pulse.

This concludes the reference run and the bit "Reference detected" is set.

Note!

Please remember that a set position is not required for operating mode "Reference run". The set position is ignored.

Control bytes

The control bytes that you use to select this operating mode are included in the traversing data.

A general description of the traversal data is available on page 6-9.

Byte	Bit 7 ... 0
4	Bit 0: enable (drive is started) Bit 2 ... 1: 01: reference run positive 10: reference run negative Bit 7 ... 3: 0
5	Irrelevant

Operating mode Hardware run

Operation

This mode is only used to approach a target position until the drive is stopped by an overrun end switch. The end switch must be connected to the reference switch input.

The traversal is governed by the values that were specified for rotational speed and acceleration or delay times. After the end switch is reached the respective position the drive is stopped with the specified delay time.

The acceleration/delay time can be modified before a new job is initiated.

If the rotational speed is altered when during the traversal, the new value is achieved by means of the current acceleration/delay time values.

Note!
Please remember that a set position is not required for operating mode "Hardware run". The set position is ignored.

Control bytes

The control bytes that you use to select this operating mode are included in the traversing data.

A general description of the traversal data is available on page 6-9.

Byte	Bit 7 ... 0
4	Bit 0: enable (drive is started) Bit 2 ... 1: 0 Bit 4 ... 3: 01: Hardware run positive 10: Hardware run negative Bit 7 ... 5: 0
5	Irrelevant

Operating mode Incremental run

Operation

The incremental mode makes use of relative positions, i.e. the value supplied as set position is added to the actual position.

When the enable bit is set, the drive travels in a positive or negative direction for the specified relative value. The drive uses the predefined values for rotational speed and acceleration to travel to the new position. If the position is negative the drive will be reversed.

You can modify the acceleration/delay time before you initiate a new job.

If the rotational speed is altered when during the traversal, the new value is achieved by means of the current acceleration/delay time values.

Control bytes

The control bytes that you use to select this operating mode are included in the traversing data.

A general description of the traversal data is available on page 6-9.

Byte	Bit 7 ... 0
4	Bit 0: enable (drive is started) Bit 4 ... 1: 0 Bit 5: 1 (Incremental run) Bit 7 ... 6: 0
5	Irrelevant

Operating mode Infinite incremental mode

Operation

In this mode the position supplied as a value is approached as a absolute position when enabled. When the position is reached, the set and the actual position are set to zero. You can use this mode to move the drive in one direction without counter overflow condition.

You can modify the acceleration/delay time before you initiate a new job.

You may specify a new value for the rotational speed at any time. If the rotational speed is altered during the traversal, the new value is achieved by means of the current acceleration/delay time values.

Control bytes

The control bytes that you use to select this operating mode are included in the traversing data.

A general description of the traversal data is available on page 6-9.

Byte	Bit 7 ... 0
4	Bit 0: enable (drive is started) Bit 5 ... 1: 0 Bit 6: 1 (Infinite incremental run) Bit 7: 0
5	Irrelevant

Operating mode Non-maintained command mode

Operation

The drive is released by setting Bit 0 in Byte 4 (enable) with before opposed rotational speed and acceleration. By setting Bit 1 or Bit 2 in Byte 5, a rotation direction is given and the drive starts. The drive stops as soon as Bit 1 or Bit 2 of Byte 5 is set back.

Control bytes

The control bytes that you use to select this operating mode are included in the traversing data.

A general description of the traversal data is available on page 6-9.

Byte	Bit 7 ... 0
4	Bit 0: enable (drive is started)
5	Bit 0: Reset counter at non-maintained command mode (edge 0 after 1 sets back the actual position to zero) [1)] Bit 1: 1 direction of rotation positive Bit 2: 1 direction of rotation negative

1)

Note!

The reset of the counter may only be executed in the non-maintained command mode. During positioning mode the regulator would throw a drag error because of the jumping actual value.

Data transfer >> CPU

The following values are transferred cyclically by the MotionControl Servo module to the CPU and stored.

Byte no.	Name	Length	Range	Unit
3, 2, 1, 0	Set position	4Byte	32Bit Integer	Encoder increments
7, 6, 5, 4	Actual position	4Byte	32Bit Integer	Encoder increments
9, 8	Set rotational speed (Value at input of A/D transducer)	2Byte	16Bit Integer	1
11, 10	Operating mode	2Byte	binary coded	
13, 12	reserved	2Byte	-	-
15, 14	Reply field identifier	2Byte		hex

Operating state

Byte	Bit 7 ... 0
10	Bit 0: enable issued Bit 1: clockwise rotation Bit 2: anticlockwise rotation Bit 3: position reached Bit 4: HW start switch operated Bit 5: HW end switch operated Bit 6: HW reference switch operated Bit 7: Reference detected
11	Bit 0: Drag error detected Bit 2 ... 1: reserved Bit 3: Internal Error Bit 4: SW end switch anticlockwise rotation Bit 5: SW end switch clockwise rotation Bit 6: Unacceptable mode Bit 7: reserved

Example

If the MotionControl Servo module was addressed starting at peripheral address PY128 in your CPU, you may obtain the "set position" from PY128 to PY131.

Other values follow these values in the peripheral area in accordance with the list above.

For example, the 2Byte for the "Operating state" are located at PY138...PY139.

Technical data

MotionControl Servo module FM 254

Electrical data	VIPA 254-1BA00
Voltage supply	DC 24V (20.4 ... 28.8) via front from ext. power supply
Current consumption Control interface	200mA
Current consumption backplane bus	200mA
Status indicator	via LEDs on the frontside
Connectors / interfaces	
Encoder	Incremental encoder
Signal voltages	5V as per RS 422
Supply voltage	5.2V / 300mA
	24V / 300mA
Count frequency	200 000Pulse/s = 50 000Increments/s
Control	
Scanning interval	2ms
Set point output	-10 ... +10V
Digital inputs	
Number	3
Supply voltage	DC 24V
Digital outputs	
Number	1
Potential separation	no
Output current	0.5A
Lamp load	5W
Programming data	
Input data	16Byte
Output data	16Byte
Parameter data	16Byte
Diagnostic data	-
Dimensions and Weight	
Dimensions (WxHxD) in mm	25.4x76x78
Weight	80g

Appendix

A Index

M.Stich