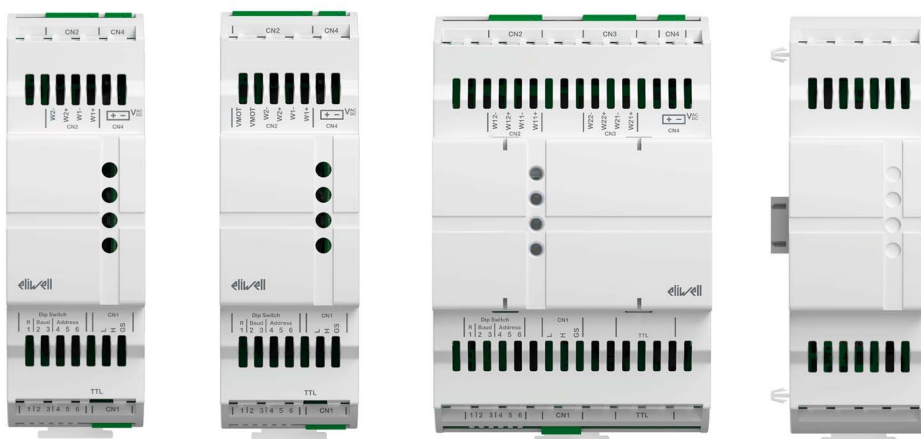


FREE EVEVD Electronic Expansion Valve Driver User Guide

Original instructions

12/2020



The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor Eliwell nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric or Eliwell, given in writing. You also agree not to establish any hypertext links to this document or its content. Either Schneider Electric or Eliwell does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Eliwell software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2020 Eliwell. All rights reserved.

Table of Contents



	Safety Information	5
	About the Book	7
Part I	Overview	11
Chapter 1	FREE EVEVD Range Overview	13
	FREE EVEVD Electronic Expansion Valve Driver Offer Overview	14
	Valve Driver Range Overview	15
	Battery Backup Overview	16
	Accessories	17
Part II	Global Features	19
Chapter 2	Before Starting	21
	Before Starting	21
Chapter 3	Wiring Best Practices	25
	Wiring Best Practices	25
Chapter 4	Installation	29
	EVEVD••000500 Electronic valve drivers Mounting Positions	30
	EVEVDBA000000 Battery Backup Installation	31
	Mounting Clearances	32
	Top Hat Section Rail (DIN Rail)	33
	Electronic Valve Drivers and Battery Backup Installation	36
Part III	Electronic Valve Drivers and Battery Backup	39
Chapter 5	Environmental Characteristics	41
	Environmental Characteristics	41
Chapter 6	EVEVD••000500 Electronic Valve Drivers Description	43
	EVEVD1U000500	44
	EVEVD1B000500	46
	EVEVD2B000500	48
Chapter 7	EVEVDBA000000 Battery Backup Description	51
	EVEVDBA000000	51
Chapter 8	Electrical Characteristics and Wiring Diagrams	53
8.1	Power Supply	54
	Power Supply	54
8.2	Valve Output	56
	Unipolar Valve Output	57
	Bipolar Valve Output	58
8.3	Communication	59
	CAN Expansion Bus Port	60
	TTL Ports	63
	Battery Backup Port	65
Chapter 9	User Interface	67
	EVEVD••000500 User Interface	67
Part IV	Parameters	69
Chapter 10	Parameters	71
	Overview	72
	Valve Parameters in Step Unit	73
	Electronic Expansion Valve Driver Parameters Table	75
	I/O Values Folder	77
	Unipolar Valve Configuration Parameters	80
	Bipolar Valve Configuration Parameters	83

Part V	Commissioning	87
Chapter 11	Commissioning	89
	Commissioning	90
	Add a EVEVD••000500 to your project	91
	Configure the EVEVD••000500	93
	Advanced Configuration and Programming the EVEVD••000500	97
	PC Connection	98
Part VI	Maintenance	99
Chapter 12	Maintenance	101
	Maintenance	101
Chapter 13	BIOS Update	103
	Device BIOS Update	103
Part VII	Valve Drivers Application Function Blocks	105
Chapter 14	EVEVD••000500 Application Function Block Overview	107
	Application Function Block Library	108
	Regulation Overview	110
Chapter 15	EVEVD••000500 Settings	
	Management: EEVDSettingsU/EEVDSettingsB	111
	Functional Overview	112
	EEVDSettingsU/B Application Function Blocks Details During Commissioning	113
	Input Pin Description	115
	Output Pin Description	117
	Troubleshooting	118
	Implementation	119
Chapter 16	EVEVD••000500 Application Control: EEVDCntrl	121
	Functional Overview	122
	EEVDCntrl Application Function Block Details During Commissioning	123
	Input Pin Description	125
	Output Pin Description	128
	Troubleshooting	129
	Implementation	130
	Advanced Use	132
Chapter 17	EVEVD••000500: CAN Expansion Bus Monitoring CheckComAlarm	133
	CAN Expansion Bus Monitoring CheckComAlarm	133
Chapter 18	EVEVD••000500: Standard Application Control Algorithm EEVSHCntrl ..	135
	Standard Application Control Algorithm (EEVSHCntrl)	136
	Evolved Superheat Algorithm (EEVSHCntrl)	139
	Input Pin Description	141
	Output Pin Description	146
	Parameters Description	148
	Troubleshooting	153
	Implementation	155
Chapter 19	EVEVD••000500 Evaporator Temperature and Superheat	
	Calculation: SuperHeatCalc	157
	Evaporator Temperature and Superheat Calculation (SuperHeatCalc)	158
	Input Pin Description	159
	Output Pin Description	161
Chapter 20	EVEVD••000500 Alarm Management: EEVDAlarmMgmt	163
	Alarm Management (EEVDAlarmMgmt)	164
	Input Pin Description	165
	Output Pin Description	167
Glossary	169
Index	171



Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric and Eliwell for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

Qualification of Personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical, electrical, or electronic equipment. The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

Intended Use

The products described or affected by this document, together with software, accessories, and options, are controllers, intended for commercial HVAC machines according to the instructions, directions, examples, and safety information contained in the present document and other supporting documentation.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements, and the technical data.

Prior to using the product, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety-related measures must be implemented.

Since the product is used as a component in an overall machine or process, you must ensure the safety of persons by means of the design of this overall system.

Operate the product only with the specified cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted is prohibited and can result in unanticipated hazards.

Prohibited Use

Any use other than that expressed above under Permitted use is strictly prohibited.

The relay contacts supplied are of an electromechanical type and subject to wear. Functional safety protection devices, specified in international or local standards, must be installed externally to this device.

Liability and Residual Risks

The liability of Schneider Electric and Eliwell is limited to the proper and professional use of this product under the guidelines contained in the present and other supporting documents, and does not extend to damages caused by (but not limited to):

- Unspecified installation/use and, in particular, in contravention of the safety requirements of established legislation or specified in this document;
- Use on equipment which does not provide adequate protection against electrocution, water and dust in the actual installation conditions;
- Use on equipment in which dangerous components can be accessed without the use of specific tools;
- Installation/use on equipment which does not comply with established legislation and standards.

Disposal

The appliance (or the product) must be disposed of separately in compliance with the local standards in force on waste disposal.

About the Book



At a Glance

Document Scope

This document describes the FREE EVEVD Electronic Expansion Valve Driver and accessories, including installation, wiring, commissioning, and programming information.

NOTE: Read and understand this document and all related documents (*see page 7*) before installing, operating, or maintaining your device.

Validity Note

This document has been updated for the release of FREE Studio Plus V1.3.

For product compliance and environmental information (RoHS, REACH, PEP, EOLI, etc.), go to <https://www.eliwell.com/en/Support/Green-Premium.html>.



The technical characteristics of the devices described in the present document also appear online. To access the information online, go to the Eliwell home page: www.eliwell.com.

Related Documents

Title of documentation	Reference number
FREE Studio Plus software - Operating Guide	9MA10256 (ENG)
FREE EVEVD Electronic Expansion Valve Driver - Preconfigured Valves Parameters Guide	9MA10304 (ENG)
FREE EVEVD Electronic Expansion Valve Driver - Instruction Sheet	GDF42244
FREE Advance 7/18 IO – Instruction Sheet	9IS54609
FREE Advance 28/42 IO – Instruction Sheet	9IS54473
FREE Advance 28/42 IO isolated – Instruction Sheet	9IS54655
FREE EVE6000 / EVE10200 Expansion module	9IS54578

You can download these technical publications and other technical information from our website at www.eliwell.com.

Product Related Information

  DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

- Disconnect all power from all equipment including connected devices prior to removing any covers or doors, or installing or removing any accessories, hardware, cables, or wires except under the specific conditions specified in the appropriate hardware guide for this equipment.
- Always use a properly rated voltage sensing device to confirm the power is off where and when indicated.
- Replace and secure all covers, accessories, hardware, cables, and wires and confirm that a proper ground connection exists before applying power to the unit.
- Use only the specified voltage when operating this equipment and any associated products.

Failure to follow these instructions will result in death or serious injury.

This equipment has been designed to operate outside of any hazardous location, and exclusive of applications that generate, or have the potential to generate, hazardous atmospheres. Only install this equipment in zones known to be free, at all times, of hazardous atmospheres.

DANGER

POTENTIAL FOR EXPLOSION

- Install and use this equipment in non-hazardous locations only.
- Do not install and use this equipment in applications capable of generating hazardous atmospheres, such as those applications employing flammable refrigerants.

Failure to follow these instructions will result in death or serious injury.

For information concerning the use of control equipment in applications capable of generating hazardous materials, consult your local, regional, or national standards bureau or certification agency.

WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.¹
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Eliwell for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Care must be taken and provisions made for use of this product as a control device to avoid inadvertent consequences of commanded machine operation, controller state changes, or alteration of data memory or machine operating parameters.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- You must have a complete understanding of the application and the machine before attempting to control the application remotely.
- Isolate your industrial network from other networks inside your company.
- Take the precautions necessary to assure that you are operating remotely on the intended machine by having clear, identifying documentation within the application and its remote connection.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Eliwell and Schneider Electric adhere to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

Consult the [Schneider Electric Cybersecurity Best Practices \(https://www.eliwell.com/download/downloader.php?cat=prd_docs&id=1938\)](https://www.eliwell.com/download/downloader.php?cat=prd_docs&id=1938) for additional information.

The controller can be upgraded only with authenticated Eliwell files. In case the authenticity check fails the controller regulates with current version.

NOTICE
UNINTENDED EQUIPMENT OPERATION
Use authenticated Eliwell files only.
Failure to follow these instructions can result in equipment damage.

Electronic Expansion Valve

Before connecting the valve, carefully configure the EVEVD••000500 driver by selecting the type of valve from the valves list.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
<ul style="list-style-type: none"> • Verify the correct selection of valve model (refer to dE00 parameter). • Verify the valve parameters and data provided by the valve manufacturer before using the valve.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Always disconnect the equipment's power supply before carrying out any maintenance on the electrical connections.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety, safety function, safe state, fault, fault reset, malfunction, failure, error, error message, dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2015	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.

Standard	Description
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive (2006/42/EC)* and *ISO 12100:2010*.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Part I

Overview

Chapter 1

FREE EVEVD Range Overview

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
FREE EVEVD Electronic Expansion Valve Driver Offer Overview	14
Valve Driver Range Overview	15
Battery Backup Overview	16
Accessories	17

FREE EVEVD Electronic Expansion Valve Driver Offer Overview

General Description

EVEVD••000500 are Electronic Expansion Valve Drivers (EEVD) which are designed as an expansion to the FREE Advance Logic Controller (FREE Advance).

The EEVD are designed to control Electronic Expansion Valves (EEVs).

The EEVD communicate with the controller using the CAN Expansion Bus.

FREE EVEVD Electronic Expansion Valve Driver devices are designed to control simple or complex machines:

- Air/water-cooled chiller
- Rooftop unit
- Heat pump
- Compressor rack
- Air conditioner

Preconfigured Electronic Valves

The valve drivers are provided with a set of parameters corresponding to a set of preconfigured valve references, both unipolar and bipolar models.

For more details, refer to the [FREE EVEVD Preconfigured Electronic Expansion Valves list](#).

Programming Software

In association with the controllers hardware, the FREE Studio Plus development tool is available to program and customize applications.

You can download FREE Studio Plus - Programming Software for FREE Advance Logic Controllers from [Eliwell web site download center](#).

The use of several programming languages in accordance with IEC 61131-3 regulations (programming standard for industrial control), makes it possible to develop new algorithms or entire programs easily, which can then be uploaded to the FREE Advance controllers via a PC and a Programming cable, helping to provide confidentiality with appropriate security.

For more information, refer to PC Connection ([see page 98](#)).

Valve Driver Range Overview

Type Code

Valve Driver type code:

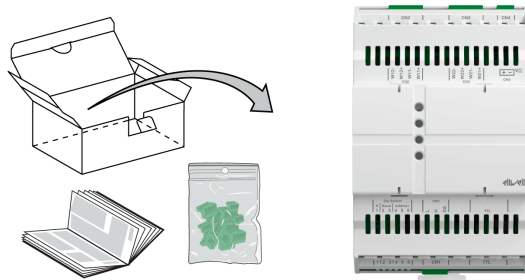
Type code description						
EVEVD1U000500	EVEVD	1	U	000	5	00
Product family	E = Expansion Electronic Valve Driver					
Number of drivable valves	1 2					
Valve type	U = Unipolar B = Bipolar					
Not relevant					000	
Power Supply					5 = 24 Vac/dc	
Not relevant						00

Valve Driver References

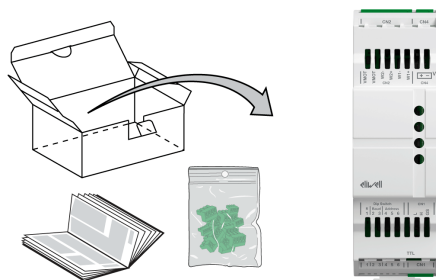
Reference	Description	Details	Power Supply	DIP Switch	Communication	
					1 TTL (1)	1 CAN Exp. bus
EVEVD1U000500	EEVD Expansion 1 EEV Unipolar (<i>see page 44</i>)	1 Unipolar (up to 6 wires)	24 Vac ± 10% 50/60 Hz / 20...38 Vdc	✓	✓	✓
EVEVD1B000500	EEVD Expansion 1 EEV Bipolar (<i>see page 46</i>)	1 Bipolar (4 wires)				
EVEVD2B000500	EEVD Expansion 2 EEV Bipolar (<i>see page 48</i>)	2 Bipolar (4 wires + 4 wires)				
(1) For service only.						

EVEVD••000500 Delivery Content

EVEVD2B000500:



EVEVD1U000500 / EVEVD1B000500:



Battery Backup Overview

Type Code

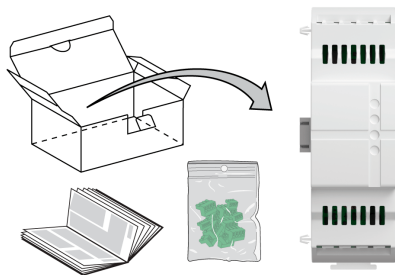
Battery backup type code:

Type code description					
EVEVDBA000000	EVEVD	BA	000	0	00
Product family	E = Expansion Electronic Valve Driver				
Battery backup	BA = Battery backup				
Not relevant			000		
Not relevant				0	
Not relevant					00

Battery Backup References

Reference	Description	Details	Power Supply	DIP Switch	Communication	
					1 TTL	1 CAN Exp. bus
EVEVDBA000000	EEVD Battery backup	-	-	-	-	-

EVEVDBA000000 Delivery Content



Accessories

Overview

This section describes the accessories.

Mounting and Wiring Accessories References

Description	Use	Reference
12 clips-on lock	To install the EVEVD••000500 Electronic Expansion Valve Drivers and the EVEVDBA000000 battery backups on a panel surface <i>(see page 36)</i>	AVA00PMCL0000
Programming cable	To connect the PC to the device ("Yellow cable" is needed)	DMI
USB-USB cable		
"Yellow cable": cable (USB connector / 5-pin connector) with one yellow wire	To connect PC (with DMI programming cable) to the device.	
"Blue cable": cable (USB connector / 5-pin connector) with one blue wire	Not used	

Part II

Global Features

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
2	Before Starting	21
3	Wiring Best Practices	25
4	Installation	29

Chapter 2

Before Starting

Before Starting

Before Starting

Read and understand this chapter before beginning the installation of your system.

Pay particular attention in conforming to any safety information, different electrical requirements, and normative standards that would apply to your machine or process in the use of this equipment.

The use and application of the information contained herein require expertise in the design and programming of automated control systems. Only you, the user, machine builder or integrator, can be aware of all the conditions and factors present during installation and setup, operation, and maintenance of the machine or process, and can therefore determine the automation and associated equipment and the related safeties and interlocks which can be effectively and properly used. When selecting automation and control equipment, and any other related equipment or software, for a particular application, you must also consider any applicable local, regional or national standards and/or regulations.

WARNING

REGULATORY INCOMPATIBILITY

Ensure that all equipment applied and systems designed comply with all applicable local, regional, and national regulations and standards.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Disconnecting Power

All options and modules should be assembled and installed before installing the control system on a mounting rail, onto a mounting plate or in a panel. Remove the control system from its mounting rail, mounting plate or panel before disassembling the equipment.

DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

- Disconnect all power from all equipment including connected devices prior to removing any covers or doors, or installing or removing any accessories, hardware, cables, or wires except under the specific conditions specified in the appropriate hardware guide for this equipment.
- Always use a properly rated voltage sensing device to confirm the power is off where and when indicated.
- Replace and secure all covers, accessories, hardware, cables, and wires and confirm that a proper ground connection exists before applying power to the unit.
- Use only the specified voltage when operating this equipment and any associated products.

Failure to follow these instructions will result in death or serious injury.

Programming Considerations

The products described in this manual have been designed and tested using Eliwell programming, configuration, and maintenance software products.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Eliwell for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Operating Environment

This equipment has been designed to operate outside of any hazardous location, and exclusive of applications that generate, or have the potential to generate, hazardous atmospheres. Only install this equipment in zones known to be free, at all times, of hazardous atmospheres.

⚠ DANGER
POTENTIAL FOR EXPLOSION
<ul style="list-style-type: none">● Install and use this equipment in non-hazardous locations only.● Do not install and use this equipment in applications capable of generating hazardous atmospheres, such as those applications employing flammable refrigerants.
Failure to follow these instructions will result in death or serious injury.

For information concerning the use of control equipment in applications capable of generating hazardous materials, consult your local, regional, or national standards bureau or certification agency.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Install and operate this equipment according to the conditions described in the Environmental Characteristics.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Installation Considerations

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
<ul style="list-style-type: none">● Use appropriate safety interlocks where personnel and/or equipment hazards exist.● Install and operate this equipment in an enclosure appropriately rated for its intended environment and secured by a keyed or tooled locking mechanism.● Use the sensor and actuator power supplies only for supplying power to the sensors or actuators connected to the module.● Power line and output circuits must be wired and fused in compliance with local and national regulatory requirements for the rated current and voltage of the particular equipment.● Do not use this equipment in safety-critical machine functions unless the equipment is otherwise designated as functional safety equipment and conforming to applicable regulations and standards.● Do not disassemble, repair, or modify this equipment.● Do not connect any wiring to unused connections, or to connections designated as No Connection (N.C.).
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The FREE Advance electronic expansion valve drivers are intended for Top Hat Section Rail (DIN rail) mounting, panel mounting, or wall mounting.

Care must be taken to avoid damage from electrostatic sources when handling this equipment. In particular exposed connectors and, in some cases, exposed printed circuit boards are exceptionally vulnerable to electrostatic discharge.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION DUE TO ELECTROSTATIC DISCHARGE DAMAGE
<ul style="list-style-type: none">● Keep equipment in the protective conductive packaging until you are ready to install the equipment.● Only install equipment in approved enclosures and / or locations that prevent casual access and provide electrostatic discharge protection.● Use a conductive wrist strap or equivalent field force protective device attached to an earth ground when handling sensitive equipment.● Always discharge yourself by touching a grounded surface or approved antistatic mat before handling the equipment.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

For more information about enclosures, refer to the definition found in IEC 1000-4-2.


Chapter 3


Wiring Best Practices

Wiring Best Practices

Wiring Best Practices

The following information describes the wiring guidelines and associated best practices to be respected when using a FREE EVEVD Electronic Expansion Valve Driver.

 DANGER
HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH <ul style="list-style-type: none">• Disconnect all power from all equipment including connected devices prior to removing any covers or doors, or installing or removing any accessories, hardware, cables, or wires except under the specific conditions specified in the appropriate hardware guide for this equipment.• Always use a properly rated voltage sensing device to confirm the power is off where and when indicated.• Replace and secure all covers, accessories, hardware, cables, and wires and confirm that a proper ground connection exists before applying power to the unit.• Use only the specified voltage when operating this equipment and any associated products. Failure to follow these instructions will result in death or serious injury.

 WARNING
LOSS OF CONTROL <ul style="list-style-type: none">• The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.• Separate or redundant control paths must be provided for critical control functions.• System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.• Observe all accident prevention regulations and local safety guidelines.¹• Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service. Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

Wiring Guidelines

The following rules must be applied when wiring FREE Advance offer product range:

- I/O and communication wiring must be kept separate from the power wiring. Route these two types of wiring in separate cable ducting.
- Verify that the operating conditions and environment are within the specification values.
- Use proper wire sizes to meet voltage and current requirements.
- Use copper conductors (required).
- Use twisted pair, shielded cables for analog, and/or fast I/O.
- Use twisted pair, shielded cables for networks, and fieldbus.

Use shielded, properly grounded cables for all analog and high-speed inputs or outputs and communication connections. If you do not use shielded cable for these connections, electromagnetic interference can cause signal degradation. Degraded signals can cause the controller or attached modules and equipment to perform in an unintended manner.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Use shielded cables for all fast I/O, analog I/O and communication signals.
- Ground cable shields for all analog I/O, fast I/O and communication signals at a single point¹.
- Route communication and I/O cables separately from power cables.

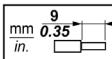









Failure to follow these instructions can result in death, serious injury, or equipment damage.

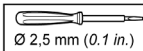

¹Multipoint grounding is permissible if connections are made to an equipotential ground plane dimensioned to help avoid cable shield damage in the event of power system short-circuit currents.

NOTE: Surface temperatures may exceed 60 °C (140 °F). Route primary wiring (wires connected to power mains) separately and apart from secondary wiring (extra low voltage wiring coming from intervening power sources). If that is not possible, double insulation is required such as conduit or cable gains.

Rules for Screw Terminal Block

The following table presents the cable types and wire sizes for a 3.50 mm (0.14 in.) pitch screw terminal block:

									
	mm ²	0.14...1.5	0.14...1.5	0.25...1.5	0.25...0.5	2 x 0.08...0.5	2 x 0.08...0.75	2 x 0.25...0.34	2 x 0.5
	AWG	26...16	26...16	22...16	22...20	2 x 28...20	2 x 28...20	2 x 24...22	2 x 20

		N·m	0.22...0.25
		lb-in	1.95...2.21

The use of copper conductors is required.

⚡ ⚠ DANGER

LOOSE WIRING CAUSES ELECTRIC SHOCK

Tighten connections in conformance with the torque specifications.

Failure to follow these instructions will result in death or serious injury.

⚠ WARNING

FIRE HAZARD

Use only the recommended wire sizes for the current capacity of the I/O channels and power supplies.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Special Handling Considerations

Care must be taken to avoid damage from electrostatic sources when handling this equipment. In particular exposed connectors and, in some cases, exposed printed circuit boards are exceptionally vulnerable to electrostatic discharge.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION DUE TO ELECTROSTATIC DISCHARGE DAMAGE

- Keep equipment in the protective conductive packaging until you are ready to install the equipment.
- Only install equipment in approved enclosures and / or locations that prevent casual access and provide electrostatic discharge protection.
- Use a conductive wrist strap or equivalent field force protective device attached to an earth ground when handling sensitive equipment.
- Always discharge yourself by touching a grounded surface or approved antistatic mat before handling the equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Electronic Expansion Valve

Before connecting the valve, carefully configure the EVEVD••000500 driver by selecting the type of valve from the valves list.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify the correct selection of valve model (refer to dE00 parameter).
- Verify the valve parameters and data provided by the valve manufacturer before using the valve.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Always disconnect the equipment's power supply before carrying out any maintenance on the electrical connections.

For a correct connection, adhere to the following:

- Separate the cables of probes and digital inputs from inductive loads and dangerous voltage connections to prevent any electromagnetic interference. Do not place the probe cables near other electrical equipment (switches, meters, etc.)
- Make connections as short as possible and do not wind them around electrically connected parts.

Chapter 4

Installation

What Is in This Chapter?

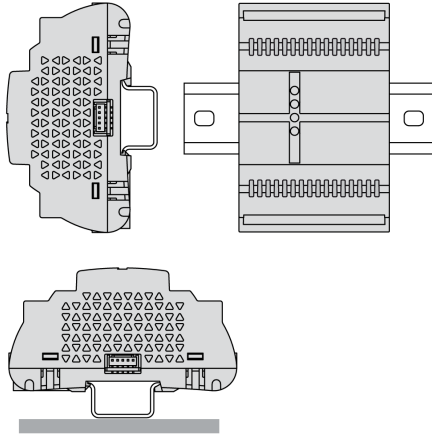
This chapter contains the following topics:

Topic	Page
EVEVD••000500 Electronic valve drivers Mounting Positions	30
EVEVDBA000000 Battery Backup Installation	31
Mounting Clearances	32
Top Hat Section Rail (DIN Rail)	33
Electronic Valve Drivers and Battery Backup Installation	36

EVEVD••000500 Electronic valve drivers Mounting Positions

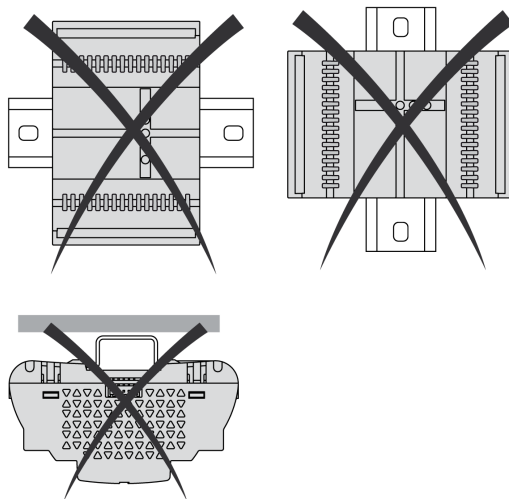
Correct Mounting Position

EVEVD••000500 Electronic valve drivers must be mounted horizontally on a vertical plane or horizontally upward as shown in the figure below:



Incorrect Mounting Position

EVEVD••000500 Electronic valve drivers cannot be mounted neither vertically, nor horizontally backward:



EVEVDBA000000 Battery Backup Installation

Overview

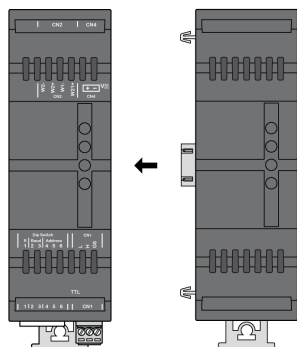
The EVEVDBA000000 battery backup must be plugged to a EVEVD**000500 Electronic Expansion Valve Driver.

In case of EVEVD**000500 power supply outage, to be able to close valves:

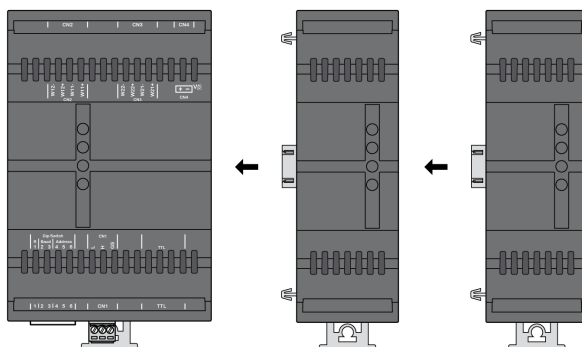
- Install one EVEVDBA000000 with the EVEVD1U000500 or the EVEVD1B000500
- Install two EVEVDBA000000 with the EVEVD2B000500

For more details about installation, refer to Electronic Expansion Valve Drivers and Battery Backup Installation (*see page 36*).

Installation of one EVEVDBA000000



Installation of two EVEVDBA000000



Commissioning

Once installed, you have to indicate the number of installed battery backup module on your EVEVD**000500 electronic expansion valve driver. To do this, you must update the value of the parameter n36 (*see page 80*).

Mounting Clearances

Minimum Clearances

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Place devices dissipating the most heat at the top of the cabinet and ensure adequate ventilation.
- Avoid placing this equipment next to or above devices that might cause overheating.
- Install the equipment in a location providing the minimum clearances from all adjacent structures and equipment as directed in this document.
- Install all equipment in accordance with the specifications in the related documentation.

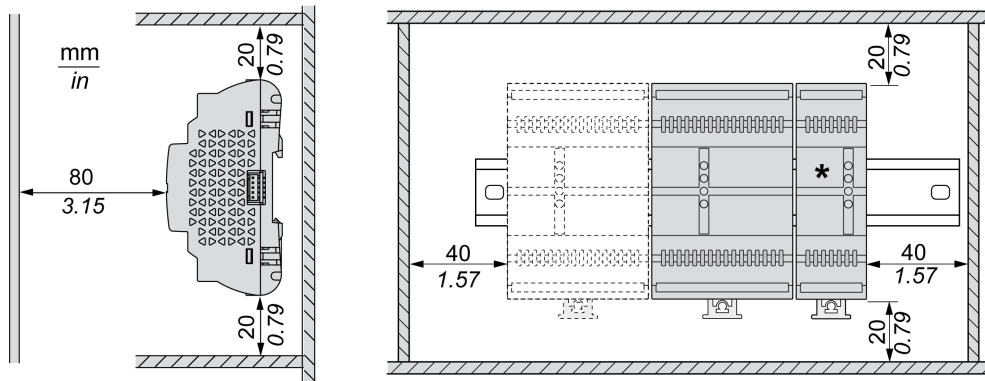
Failure to follow these instructions can result in death, serious injury, or equipment damage.

EVEVD••000500 electronic valve drivers and EVEVDBA000000 battery backup have been designed as IP20 products and must be installed in an enclosure appropriately rated for its intended environment and secured by a keyed or tooled locking mechanism .

There are 3 types of clearances between:

- The FREE Advance device and the sides of the cabinet (including the panel door).
- The FREE Advance device terminal blocks and the wiring ducts. This distance reduces electromagnetic interference between the device and the wiring ducts.
- The FREE Advance device and other heat generating devices installed in the same cabinet.

The following figure shows the minimum clearances that apply to EVEVD••000500 references:



*: EVEVDBA000000

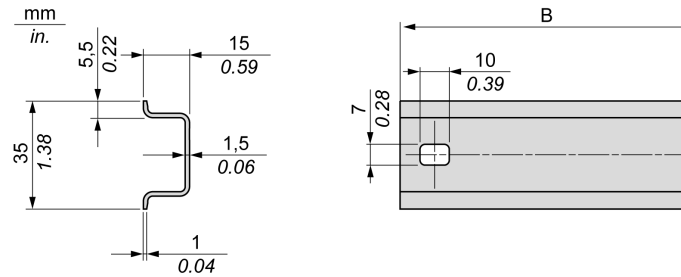
Top Hat Section Rail (DIN Rail)

Dimensions of Top Hat Section Rail (DIN Rail)

You can mount the controller and valve driver on a 35 mm (1.38 in.) top hat section rail (DIN rail). It can be attached to a smooth mounting surface or suspended from a EIA rack or mounted in a NEMA cabinet.

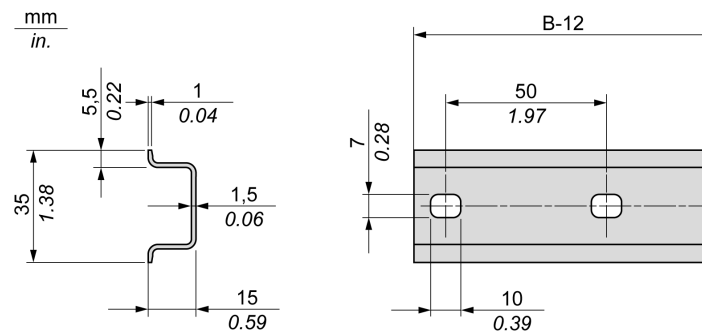
Symmetric Top Hat Section Rails (DIN Rail)

The following illustration and table show the references of the top hat section rails (DIN rail) for the wall-mounting range:



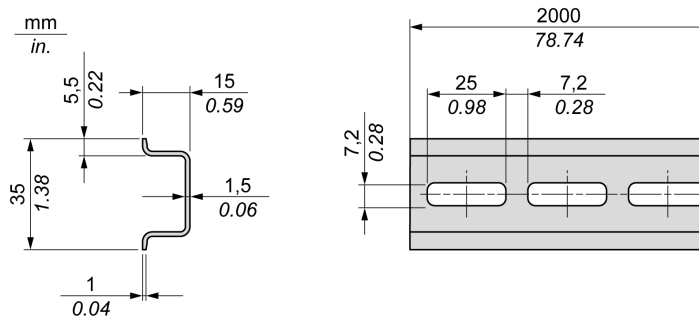
Schneider Electric Reference	Type	Rail length (B)
NSYSDR50A	A	450 mm (17.71 in.)
NSYSDR60A	A	550 mm (21.65 in.)
NSYSDR80A	A	750 mm (29.52 in.)
NSYSDR100A	A	950 mm (37.40 in.)

The following illustration and table show the references of the symmetric top hat section rails (DIN rail) for the metal enclosure range:



Schneider Electric Reference	Type	Rail length (B-12 mm)
NSYSDR60	A	588 mm (23.15 in.)
NSYSDR80	A	788 mm (31.02 in.)
NSYSDR100	A	988 mm (38.89 in.)
NSYSDR120	A	1188 mm (46.77 in.)

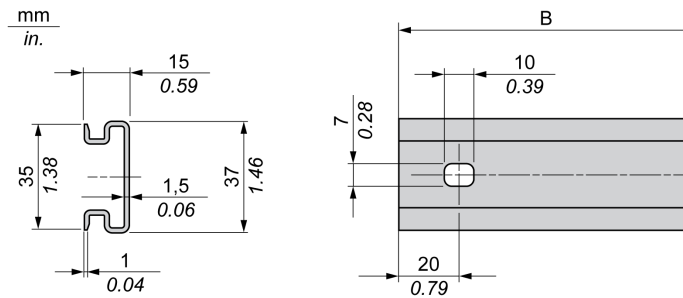
The following illustration and table shows the references of the symmetric top hat section rails (DIN rail) of 2000 mm (78.74 in.):



Schneider Electric Reference	Type	Rail length
NSYSDR200 ¹	A	2000 mm (78.74 in.)
NSYSDR200D ²	A	
1 Unperforated galvanized steel 2 Perforated galvanized steel		

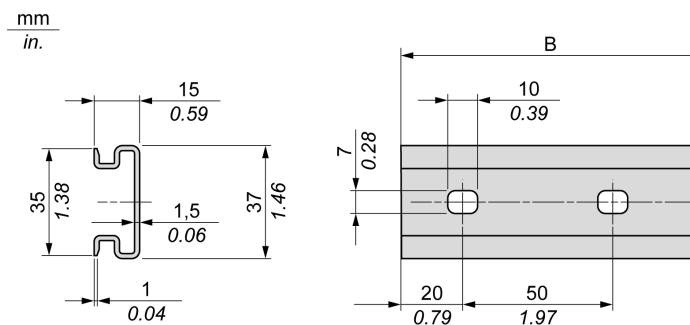
Double-Profile Top Hat Section Rails (DIN Rail)

The following illustration and table show the references of the double-profile top hat section rails (DIN rails) for the wall-mounting range:



Schneider Electric Reference	Type	Rail length (B)
NSYDPR25	W	250 mm (9.84 in.)
NSYDPR35	W	350 mm (13.77 in.)
NSYDPR45	W	450 mm (17.71 in.)
NSYDPR55	W	550 mm (21.65 in.)
NSYDPR65	W	650 mm (25.60 in.)
NSYDPR75	W	750 mm (29.52 in.)

The following illustration and table show the references of the double-profile top hat section rails (DIN rail) for the floor-standing range:



Schneider Electric Reference	Type	Rail length (B)
NSYDPR60	F	588 mm (23.15 in.)
NSYDPR80	F	788 mm (31.02 in.)
NSYDPR100	F	988 mm (38.89 in.)
NSYDPR120	F	1188 mm (46.77 in.)

Electronic Valve Drivers and Battery Backup Installation

Overview

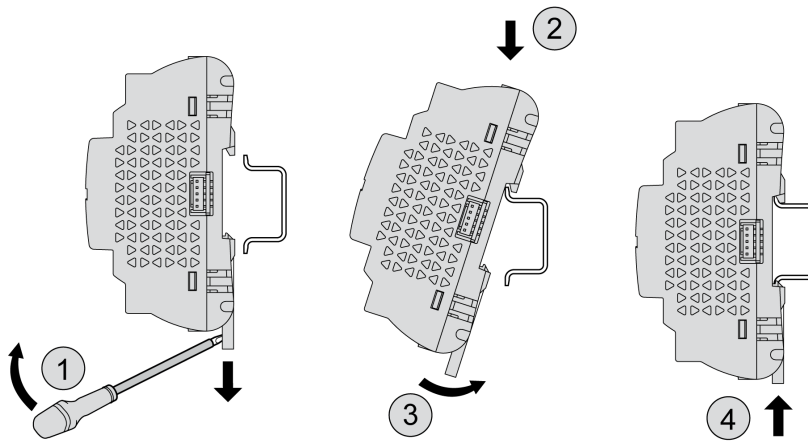
This section describes how to install and remove a EVEVD••000500 electronic valve driver and EVEVDBA000000 battery backup from a top hat section rail (DIN rail).

If a EVEVDBA000000 battery backup is needed, you must first plug the EVEVDBA000000 battery backup to a EVEVD••000500 Electronic Expansion Valve Driver (*see page 31*).

Installing on a Top Hat Section Rail (DIN Rail)

The following procedure describes how to install an electronic valve driver or a battery backup on a top hat section rail (DIN rail):

Step	Action
1	Move the two spring docking devices to their standby position (use a screwdriver to press against the relative compartments).
2	Position the top groove of the device on the top edge of the Top Hat Section Rail (DIN rail).
3	Press the assembly against the Top Hat Section Rail (DIN rail).
4	Press the spring docking devices to put them into the locked position.



Removing from a Top Hat Section Rail (DIN Rail)

The following procedure describes how to remove an electronic valve driver or a battery backup from a top hat section rail (DIN rail):

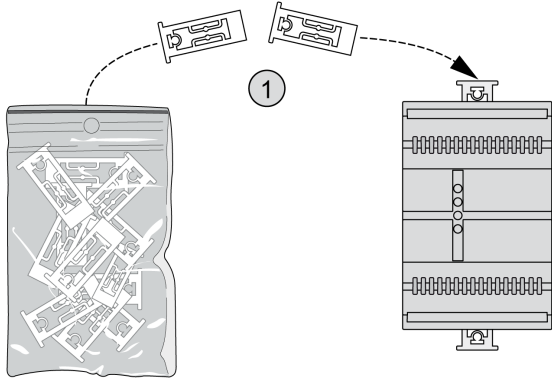
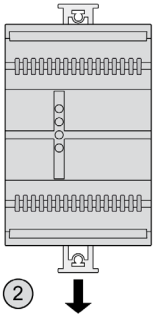
Step	Action
1	Remove the power from the device.
2	Insert a flat screwdriver into the spring docking devices.
3	Pull down the spring docking device to move it to its standby position.
4	Pull the device from the top hat section rail (DIN rail) from the bottom.

Panel Installation

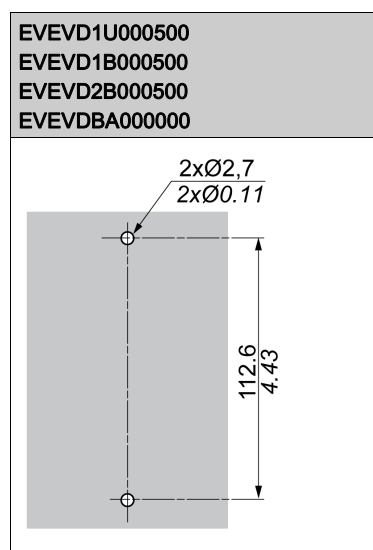
To install the electronic valve drivers or a battery backup on a panel, you must use clip-on locks.

NOTE: Upper clip-on locks are not provided with the devices and must be ordered separately (*see page 17*).

The following procedure shows how to install a EVEVD2B000500 electronic valve driver on a panel using the clip-on locks. The same procedure shall be followed for the EVEVD1U000500 / EVEVD1B000500 / EVEVDBA000000:

Step	Action
1	<p>Install the 2 upper clip-on locks:</p> 
2	<p>Move the 2 lower clip-on locks to their standby position:</p> 
3	<p>Secure the device in position with 2 screws. Refer to the mounting holes layout (see page 37).</p>

Mounting Holes Layout



Part III

Electronic Valve Drivers and Battery Backup

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
5	Environmental Characteristics	41
6	EVEVD**000500 Electronic Valve Drivers Description	43
7	EVEVDBA000000 Battery Backup Description	51
8	Electrical Characteristics and Wiring Diagrams	53
9	User Interface	67

Chapter 5

Environmental Characteristics

Environmental Characteristics

Technical Data

The FREE EVEVD Electronic Expansion Valve Driver offer components meet European Community (CE) requirements for open equipment. You must install them in an enclosure or other location designed for the specific environmental conditions and to minimize the possibility of unintended contact with hazardous voltages. Use metal enclosures to improve the electromagnetic immunity of your FREE EVEVD Electronic Expansion Valve Driver system. This equipment meets CE requirements as indicated in the following tables.

⚠ WARNING	
UNINTENDED EQUIPMENT OPERATION	
Do not exceed any of the rated values specified within this chapter.	
Failure to follow these instructions can result in death, serious injury, or equipment damage.	

Electronic Valve Drivers and Battery Backup Specifications

Characteristics	Specification				
		EVEVD1U000500	EVEVD1B000500	EVEVD2B000500	EVEVDBA000000
The product complies with the following harmonized Standards	EN 60730-1 / EN 60730-2-9	✓			
Construction of control	Electronic automatic Incorporated Control	✓			
Purpose of control	Operating control (non-safety-related)	✓			
Mounting	Top Hat Section Rail (DIN rail)	✓			
	Optional panel mounting (with accessories)	✓			
Type of action	1	✓			
Pollution degree	2 (normal)	✓			
Over-voltage category	I	✓			
Rated impulse voltage	330 V	✓			
Power supply	24 Vac (+/- 10 %) 50 Hz / 60 Hz 20...38 Vdc (UL/CSA) 24 Vdc (IEC)	✓	-		
Power Consumption	12 VA / 9 W	✓	-		
	23 VA / 18 W	-	✓	-	
	35 VA / 30 W	-	-	✓	-
Ambient operating temperature	-20...65 °C (-4...149 °F)	✓		-	
	-20...60 °C (-4...140 °F)	-		✓	
Ambient operating humidity (non-condensing)	5...95 %	✓			
Ambient storage temperature	-30...70 °C (-22...158 °F)	✓			
Ambient storage humidity (non-condensing)	5...95 %	✓			
Temperature for ball pressure test	125 °C (257 °F)	✓			
Software class and structure	A	✓			
Degree of protection by enclosure	IP20	✓			

Chapter 6

EVEVD**000500 Electronic Valve Drivers Description

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
EVEVD1U000500	44
EVEVD1B000500	46
EVEVD2B000500	48

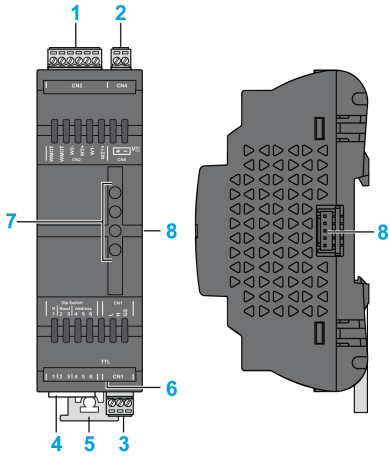
EVEVD1U000500

Overview

Reference	Description
EVEVD1U000500	EEVD Expansion 1 EEV Unipolar

Physical Description

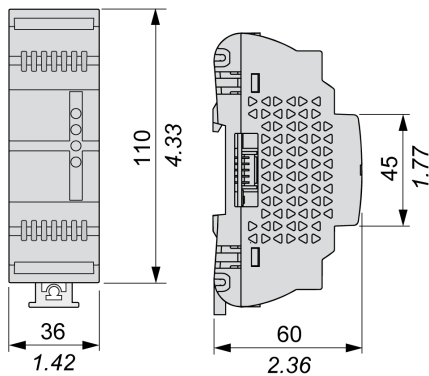
The following illustration presents the EVEVD1U000500 electronic valve driver:



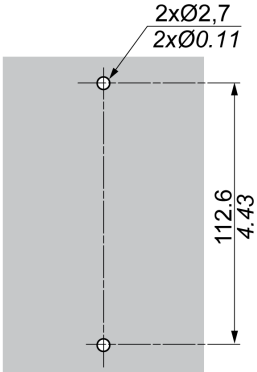
Number	Name	Description
1	CN2	Unipolar Valve Output (<i>see page 57</i>)
2	CN4	Power supply (isolated from CAN expansion bus - Functional Insulation) (<i>see page 54</i>)
3	CN1	CAN expansion bus (isolated from Power supply - Functional Insulation) (<i>see page 60</i>)
4	-	Dip Switch (<i>see page 60</i>)
5	-	Clip-on lock (<i>see page 36</i>)
6	TTL	TTL port (used for service) (<i>see page 63</i>)
7	-	User interface - LEDs (<i>see page 67</i>)
8	-	Battery Backup female connector (<i>see page 65</i>)

NOTE: The electronic valve drivers are delivered with removable screw terminal blocks (*see page 15*). For more information about the wiring, refer to wiring best practices (*see page 25*).

Dimensions



Mounting Holes Layout



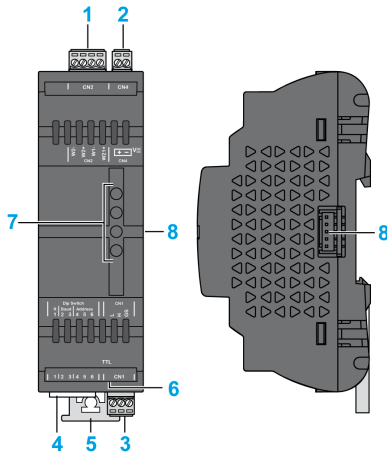
EVEVD1B000500

Overview

Reference	Description
EVEVD1B000500	EEVD Expansion 1 EEV Bipolar

Physical Description

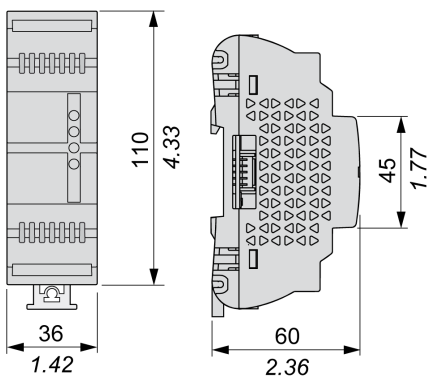
The following illustration presents the EVEVD1B000500 electronic valve driver:



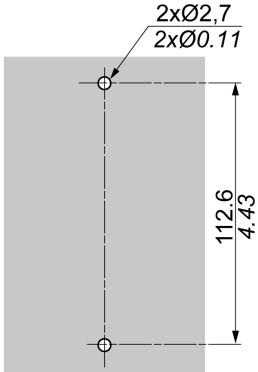
Number	Name	Description
1	CN2	Bipolar Valve Output (<i>see page 58</i>)
2	CN4	Power supply (isolated from CAN expansion bus - Functional Insulation) (<i>see page 54</i>)
3	CN1	CAN expansion bus (isolated from Power supply - Functional Insulation) (<i>see page 60</i>)
4	-	Dip Switch (<i>see page 60</i>)
5	-	Clip-on lock (<i>see page 36</i>)
6	TTL	TTL port (used for service) (<i>see page 63</i>)
7	-	User interface - LEDs (<i>see page 67</i>)
8	-	Battery Backup female connector (<i>see page 65</i>)

NOTE: The electronic valve drivers are delivered with removable screw terminal blocks (*see page 15*). For more information about the wiring, refer to wiring best practices (*see page 25*).

Dimensions



Mounting Holes Layout



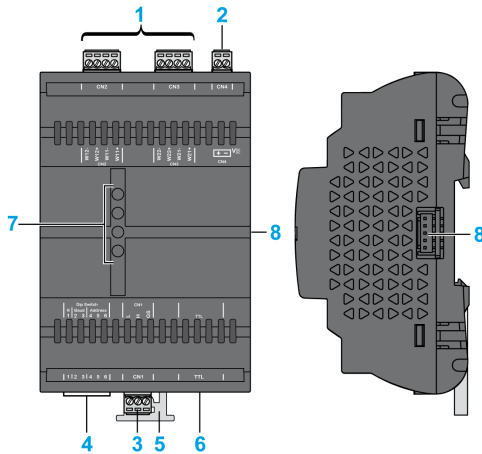
EVEVD2B000500

Overview

Reference	Description
EVEVD2B000500	EEVD Expansion 2 EEV Bipolar

Physical Description

The following illustration presents the EVEVD2B000500 electronic valve driver:

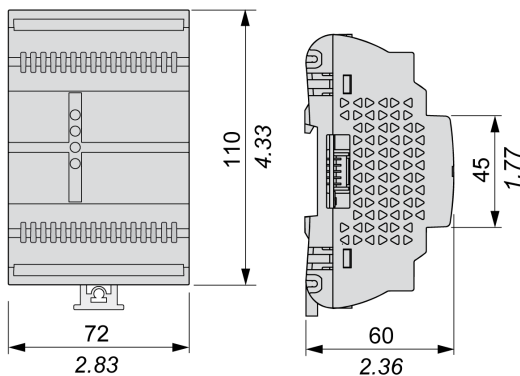


Number	Name	Description
1	CN2	Bipolar Valve 1 Output (<i>see page 58</i>)
	CN3	Bipolar Valve 2 Output (<i>see page 58</i>)
2	CN4	Power supply (isolated from CAN expansion bus - Functional Insulation) (<i>see page 54</i>)
3	CN1	CAN expansion bus (isolated from Power supply - Functional Insulation) (<i>see page 60</i>)
4	-	Dip Switch (<i>see page 60</i>)
5	-	Clip-on lock (<i>see page 36</i>)
6	TTL	TTL port (used for service) (<i>see page 63</i>)
7	-	User interface - LEDs (<i>see page 67</i>)
8	-	Battery Backup female connector (<i>see page 65</i>)

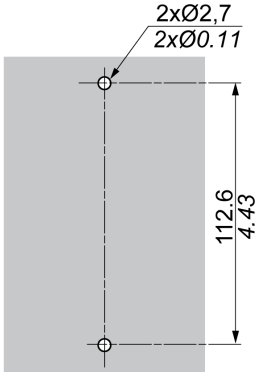
NOTE: The electronic valve drivers are delivered with removable screw terminal blocks (*see page 15*).

For more information about the wiring, refer to wiring best practices (*see page 25*).

Dimensions



Mounting Holes Layout



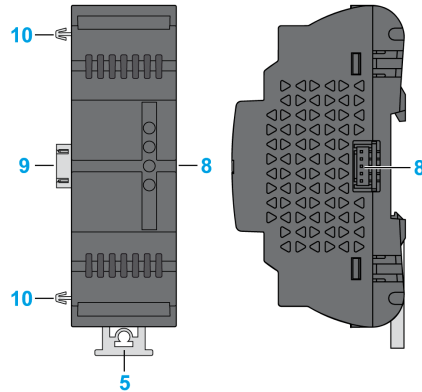
Chapter 7

EVEVDBA000000 Battery Backup Description

EVEVDBA000000

Physical Description

The following illustration presents the EVEVDBA000000 Battery Backup:



Number	Name	Description
5	-	Clip-on lock (see page 36)
8	-	Battery Backup female connector (see page 65)
9	-	Battery Backup male connector (see page 65)
10	-	Battery Backup side mounting clips (see page 65)

The battery backup female connector (9) permits you to connect the battery backup:

- To one Electronic Expansion Valve Driver
- To a second battery backup already connected to a EVEVD2B000500 Electronic Expansion Valve Driver.

⚠️ ⚠️ DANGER

HAZARD OF ELECTRIC SHOCK OR ARC FLASH

Do not short-circuit the connector pins.

Failure to follow these instructions will result in death or serious injury.

For more information about the installation, refer to EVEVDBA000000 Battery Backup installation ([see page 31](#)).

Battery Backup Charge Time

The battery backup devices are powered by the valve driver.

If one fully discharged battery pack is connected, the charge time can take up to 3 minutes.

If two fully discharged battery packs are connected, the charge time can take up to 6 minutes.

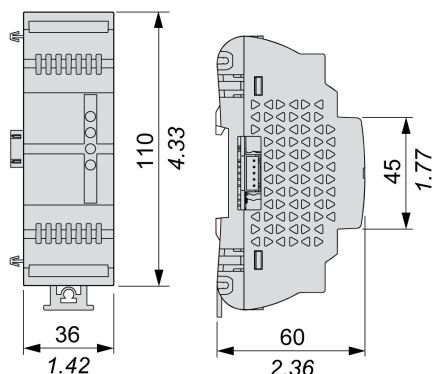
At power on, the valve driver does not move the valve for a timeout depending on the n36 parameter value, meant to allow the battery packs become fully charged:

- n36=0, 0 battery pack connected: No timeout.
- n36=1, 1 battery pack connected: 3 minutes timeout.
- n36=2, 2 battery pack connected: 6 minutes timeout.

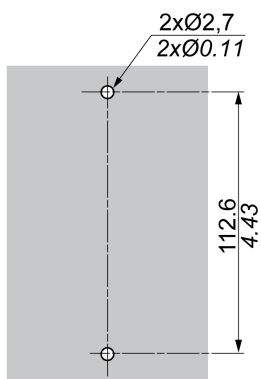
For example, if n36=1 and 1 battery pack is connected, the valve driver starts to move the valve 3 minutes after the valve driver power on.

After the timeout, the valves are still moved by the driver even if a battery pack alarm occurs (see page 77).
 Independently of the n36 value, if a power loss occurs, an already connected battery pack is used by the valve driver to close the valve.

Dimension



Mounting Holes Layout



Chapter 8

Electrical Characteristics and Wiring Diagrams

From time to time, new input modules, output modules, or other devices are made available that are not documented in the present documentation. For information on new devices, contact your local Eliwell representative.

NOTICE

INOPERABLE EQUIPMENT

Update the controller firmware to the latest version every time you install a newly released Input/Output expansion module or other device to this equipment.

Failure to follow these instructions can result in equipment damage.

NOTE: For more information on how to update the controller firmware, contact your local Eliwell representative.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
8.1	Power Supply	54
8.2	Valve Output	56
8.3	Communication	59

Section 8.1

Power Supply

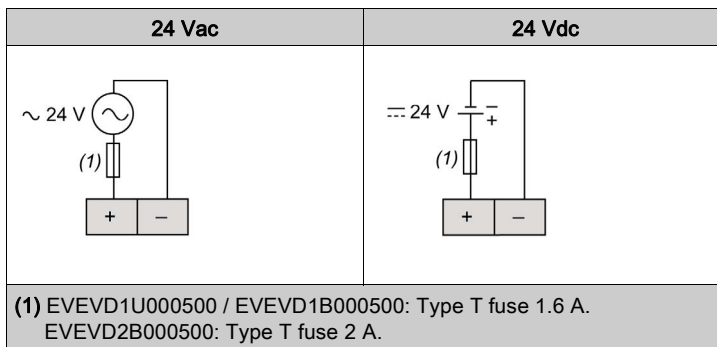
Power Supply

Electronic Expansion Valve Drivers Power Supply

Electronic Expansion Valve Drivers	Power supply characteristics		Maximum power draw
EVEVD1U000500	Isolated	24 Vac (+/- 10 %) - 50/60 Hz	12 VA
		20...38 Vdc	9 W
EVEVD1B000500		24 Vac (+/- 10 %) - 50/60 Hz	23 VA
		20...38 Vdc	18 W
EVEVD2B000500		24 Vac (+/- 10 %) - 50/60 Hz	35 VA
		20...38 Vdc	30 W

Wiring Diagram Example

Power supply wiring diagram:



Pitch of the terminal block	Wiring length
3.50 mm (0.14 in.)	10 m (32.8 ft)

NOTE: The negative terminal of the power supply connection and the signal reference for CAN Expansion Bus (indicated as GS connection) are not internally connected. For more details, refer to CAN Expansion bus wiring ([see page 61](#)).

<i>NOTICE</i>
<p>INOPERABLE EQUIPMENT</p> <p>Do not connect a power cable longer than 10 m (32.8 ft).</p> <p>Failure to follow these instructions can result in equipment damage.</p>

For more information about the wiring, refer to Best wiring practices ([see page 25](#)).

Generalities

The power supplies for the EVEVD••000500 must be rated Safety Extra Low Voltage (SELV) according to IEC 61140. These sources of power are isolated between the electrical input and output circuits of the power supply as well as simple separation from ground (earth), PELV, and other SELV systems.

⚠ DANGER**GROUND LOOP CAUSING ELECTRIC SHOCK AND/OR INOPERABLE EQUIPMENT**

- Do not connect any valve outputs to any external ground (earth) connection.
- Do not connect the 0 V power supply/transformer connection supplying this equipment and the signal reference for CAN Expansion Bus to any external ground (earth) connection.

Failure to follow these instructions will result in death or serious injury.

In all cases, if the specified voltage range is not maintained, the products may not function as intended. Use appropriate safety interlocks and voltage monitoring circuits.

⚠ WARNING**POTENTIAL OF OVERHEATING AND FIRE**

- Do not connect the equipment directly to line voltage.
- Use only isolating SELV, Class 2 power supplies / transformers to supply power to this equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Section 8.2

Valve Output

Valve Output

This table presents the valve outputs of electronic expansion valve drivers:

Description	EVEVD1U000500	EVEVD1B000500	EVEVD2B000500
Unipolar Valve Output (<i>see page 57</i>)	1	-	-
Bipolar Valve Output (<i>see page 58</i>)	-	1	2

What Is in This Section?

This section contains the following topics:

Topic	Page
Unipolar Valve Output	57
Bipolar Valve Output	58

Unipolar Valve Output

Characteristics

The table indicates the Unipolar Valve outputs characteristics:

Labels	Characteristic	Value
VMOT	Maximum voltage	12 Vdc
	Maximum current	800 mA
W±	Maximum current	400 mA
-	Built-in protection	Against overload & short-circuit (current off)

Wiring Diagram Example

EVEVD1U000500 valve output:

VMOT	VMOT	W2-	W2+	W1-	W1+
------	------	-----	-----	-----	-----

Pitch of the terminal block	Wiring length
3.50 mm (0.14 in.)	10 m (32.8 ft)

For more information about the wiring, refer to Best wiring practices (*see page 25*).

Related Devices and Connectors

The table indicates the related devices and connectors

Related Device	Connector	Label	Description										
EVEVD1U000500	CN2	<table border="1"> <tr> <td>VMOT</td> <td>VMOT</td> <td>W2-</td> <td>W2+</td> <td>W1-</td> <td>W1+</td> </tr> </table>				VMOT	VMOT	W2-	W2+	W1-	W1+	VMOT	Common
		VMOT	VMOT	W2-	W2+	W1-	W1+						
		VMOT	Common										
		W1+	W1 terminals for connection to valve first coil winding.										
		W1-											
W2+	W2 terminals for connection to valve second coil winding.												
W2-													

Wiring Description for various common valves

Refer to Wiring Description for unipolar common valves (*see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide*).

WARNING

UNINTENDED EQUIPMENT OPERATION

Before wiring valve, verify the wiring in the valve manufacturer Technical Data.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Bipolar Valve Output

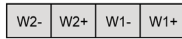
Characteristics

The table indicates the Bipolar Valve outputs characteristics:

Labels	Characteristic	Value
W+±	Maximum current	850 mA
-	Built-in protection	Against short-circuit (current off)

Wiring Diagram Example

EVEVD1B000500 valve output:



Pitch of the terminal block	Wiring length
3.50 mm (0.14 in.)	10 m (32.8 ft)

For more information about the wiring, refer to Best wiring practices (*see page 25*).

Related Devices and Connectors

The table indicates the related devices and connectors

Related Device	Connector	Label	Description	
EVEVD1B000500	CN2		W1+	Valve 1 W1 terminals for connection to valve first coil winding.
			W1-	
			W2+	W2 terminals for connection to valve second coil winding.
			W2-	
EVEVD2B000500	CN2		W11+	Valve 1 W1 terminals for connection to valve first coil winding.
			W11-	
			W12+	W2 terminals for connection to valve second coil winding.
			W12-	
	CN3		W21+	Valve 2 W1 terminals for connection to valve first coil winding.
			W21-	
			W22+	W2 terminals for connection to valve second coil winding.
			W22-	

Wxy : x = valve number, y = coil number

Wiring Description for various common valves

Refer to Wiring Description for bipolar common valves (*see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide*).

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Before wiring valve, verify the wiring in the valve manufacturer Technical Data.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Section 8.3

Communication

What Is in This Section?

This section contains the following topics:

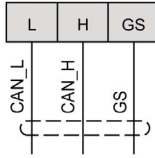
Topic	Page
CAN Expansion Bus Port	60
TTL Ports	63
Battery Backup Port	65

CAN Expansion Bus Port

Overview

The EVEVD••000500 electronic valve driver must be connected through the CAN expansion bus.

Connector



NOTE: The signal reference for CAN Expansion Bus (indicated as GS) and the negative terminal of power supply connection are not internally connected.

Pitch of the terminal block
3.50 mm (0.14 in)

Wiring

Use a twisted pair shielded cable with two conductors with section 0.5 mm² (AWG 20), plus a sheath (characteristic impedance 120 Ω) with PVC sleeve, nominal capacity between conductors 36 pF/m, nominal capacity between conductor and shield 68 pF/m.

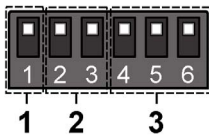
For laying wires, comply with the indications given in standard EN 50174 on information technology wiring. Extra care must be taken in separating data transmission circuits from power lines.

The network must have a bus daisy chain topology and must have 120 Ω 1/4 W termination resistances between the CAN_H and CAN_L terminals on each of the two ends of the bus or enable those embedded on the Electronic Expansion Valve Driver modules via its DIP switches (*see page 60*).

The maximum cable length depends on the communication speed set in baud:

Baud rate	Maximum network length using:
50 kBd	1000 m (3280.83 ft)
125 kBd	500 m (1640.41 ft)
250 kBd	200 m (656.17 ft)
500 kBd	30 m (98.42 ft)

EVEVD••000500 Port Configuration Using DIP Switch



The 6-position DIP switches on valve drivers are used to:

- Assert the 120 Ω termination (1)
- Set the CAN baud rate (2)
- Set the CAN address (3)

DIP switch function		DIP switch number					
		1	2	3	4	5	6
120 Ω termination	Disabled (Default setting)						
	Enabled						
1 The CAN address of the device is equal to the value of parameter Addr_CAN_OB (<i>see page 75</i>) plus the composite value of the 6-position DIP switches DIP4...DIP6 (8 addresses).							

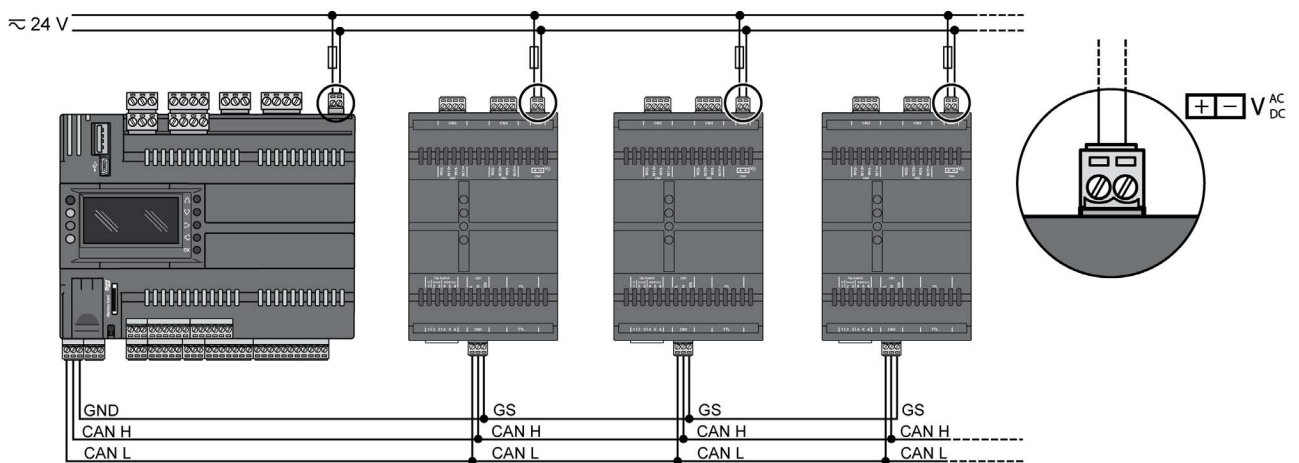
DIP switch function		DIP switch number					
		1	2	3	4	5	6
CAN baud rate	500 kBd (Default value)	-		-	-	-	-
	250 kBd	-		-	-	-	-
	125 kBd	-		-	-	-	-
	50 kBd	-		-	-	-	-
CAN address ⁽¹⁾	Address = Addr_CAN_OB + 0 (Default value)	-	-	-		-	-
	Address = Addr_CAN_OB + 1	-	-	-		-	-
	Address = Addr_CAN_OB + 2	-	-	-		-	-
	Address = Addr_CAN_OB + 3	-	-	-		-	-
	Address = Addr_CAN_OB + 4	-	-	-		-	-
	Address = Addr_CAN_OB + 5	-	-	-		-	-
	Address = Addr_CAN_OB + 6	-	-	-		-	-
	Address = Addr_CAN_OB + 7	-	-	-		-	-
1 The CAN address of the device is equal to the value of parameter Addr_CAN_OB (<i>see page 75</i>) plus the composite value of the 6-position DIP switches DIP4...DIP6 (8 addresses).							

CAN Expansion Bus (Field) Network Restriction

A CAN Expansion Bus (Field) network connection can be constituted by:

- Maximum 1 AV.....6•500 / AV.....5•500 functioning as master
- Maximum 7 EVE.....0500 expansion modules functioning as slaves. This number can be reduced, depending on the number of inputs and outputs that the master can control.
- Maximum 4 EVEVD••000500 electronic expansion valve drivers functioning as slaves.

CAN Expansion Bus (Field) Network Connection Example



Follow the resistance termination recommendations (*see page 60*).

Power Supply Restriction

If you connect the GND connection of the network across multiple devices with power supply inputs that are non-isolated (AV•84•••60500, AV•126•••0500, or EVE•••••0500), use separate, isolated power supplies. Alternatively, if you are interconnecting the equipment to a single power supply, do not connect the GND signal. Pay special attention when connecting serial lines. Miswiring may lead to inoperable equipment.

For more details, refer to Power Supply description ([see page 54](#)).

TTL Ports

Overview

Each EVEVD••000500 electronic valve driver is equipped with one TTL port.

This port permits you to communicate between FREE Studio Plus Installer and the device to:

- Download the device firmware (*see page 103*)
- Directly read/write parameters in the device.

⚠ WARNING

UNINTENDED EQUIPEMENT OPERATION

Before modifying parameters directly, verify that these parameters are not downloaded by the controller.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The controller can be upgraded only with authenticated Eliwell files. In case the authenticity check fails the controller regulates with current version.

NOTICE

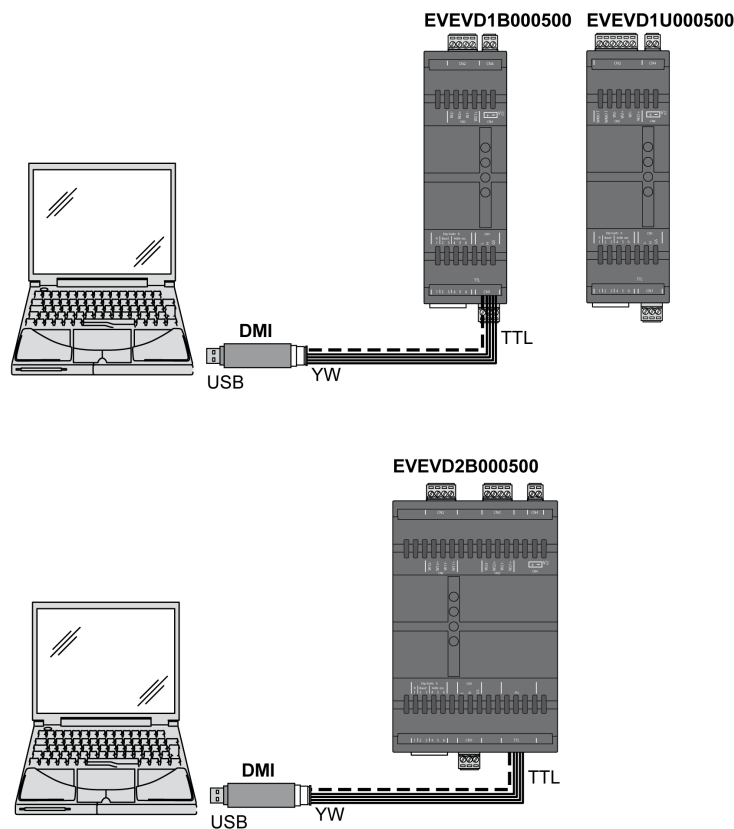
UNINTENDED EQUIPMENT OPERATION

Use authenticated Eliwell files only.

Failure to follow these instructions can result in equipment damage.

Connect EVEVD••000500 to a PC via TTL Port

To directly connect EVEVD••000500 to a PC, supply the valve driver via its 24 Vac/dc power supply connector CN4 and then connect the DMI programming cable with the yellow cable to its TTL port:



NOTE: Do not apply voltage via 24 Vac/dc while the device is already connected to a PC.

Before applying power via 24 Vac/dc power supply connection:

- Disconnect the programming cable.
- Supply the EVEVD••000500 via its 24 Vac/dc power supply connector CN4.
- Reconnect the programming cable.

Battery Backup Port

Overview

Each EVEVD••000500 electronic expansion valve driver is equipped with one battery backup port.

This female connector allows connection between EVEVD••000500 and EVEVDBA000000 battery backup module.

In case of EVEVD••000500 power supply outage, a EVEVDBA000000 battery backup allows the valve driver to close one valve.

Connection

For more details about installation, refer to the EVEVDBA000000 battery backup installation (*see page 31*).

Compatibility

In case of EVEVD••000500 power supply outage, to be able to close valves:

- Install one EVEVDBA000000 with the EVEVD1U000500 or the EVEVD1B000500
- Install two EVEVDBA000000 with the EVEVD2B000500

NOTE: Depending on the system, you could install only one EVEVDBA000000 with the EVEVD2B000500. For more details, contact your local technical support.

Commissioning

Once installed, you have to indicate the number of installed battery backup modules on your EVEVD••000500 electronic expansion valve driver. To do this, you must update the value of the parameter n36 (*see page 80*).

DANGER

HAZARD OF ELECTRIC SHOCK OR ARC FLASH

Do not short-circuit the connector pins.

Failure to follow these instructions will result in death or serious injury.

Chapter 9

User Interface

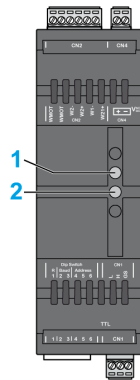
EVEVD••000500 User Interface

Overview

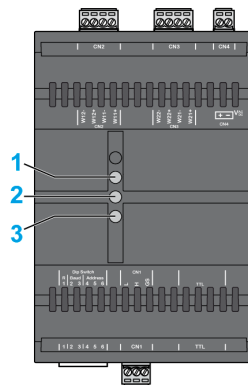
The user interface of the EVEVD••000500 electronic valve drivers has several LEDs.

LEDs

EVEVD1B000500/EVEVD1
U000500



EVEVD2B000500



Number	LED	Color	On	Flashing (1)	Off
1	Alarm	Red	-	Flashing SLOW = alarm present or the handling of an alarm is in progress on Valve 1 or Valve 2 ⁽²⁾	No detected alarm on valve(s)
2	Valve 1	Green	Power supply present and driver ready to be operative	Communication established with controller, driver operating: <ul style="list-style-type: none"> Flashing SLOW = valve is close (0%) Flashing MID = valve is open > 0% and < 100% Flashing FAST = valve fully open (100%) 	No power supply, driver is off
3 ⁽²⁾	Valve 2	Green			
<p>1 SLOW: 500 ms ON / 500 ms OFF, MID: 250 ms ON / 250 ms OFF, FAST: 125 ms ON / 125 ms OFF</p> <p>2 Valve 2 only for EVEVD2B000500.</p>					

Part IV

Parameters

Chapter 10

Parameters

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Overview	72
Valve Parameters in Step Unit	73
Electronic Expansion Valve Driver Parameters Table	75
I/O Values Folder	77
Unipolar Valve Configuration Parameters	80
Bipolar Valve Configuration Parameters	83

Overview

Overview

Parameters are used to configure a FREE EVEVD Electronic Expansion Valve Driver.

There are two ways to configure the valve driver:

Connection type	Connection Software Accessories	Description
FREE Studio Plus software with Application Function Blocks (Recommended) or with ADVANCED SETTINGS tab (<i>see page 94</i>).	<ul style="list-style-type: none"> ● PC->controller->valve driver ● FREE Studio Plus software ● Cables: <ul style="list-style-type: none"> ○ Ethernet or USB cable to connect the PC to the controller. ○ CAN cable to connect the controller to the valve driver. 	<p>The parameter modification is sent by the controller to the valve driver via the CAN expansion bus.</p> <p>For more details, refer to:</p> <ul style="list-style-type: none"> ● Commissioning (<i>see page 87</i>). ● Maintenance (<i>see page 99</i>).
Direct connection. (For expert maintenance only)	<ul style="list-style-type: none"> ● PC->valve driver ● FREE Studio Plus Installer ● DMI programming cable 	<p>With FREE Studio Plus Installer, you can modify the parameters "on the field". Pay attention, these new values may be overwritten by the controller through the CAN Expansion bus.</p> <p>For more details about installation, refer to TTL Port description (<i>see page 63</i>).</p>

Parameters Table

The following table list the information required to read, write, and decode the accessible resources in the device: Electronic Expansion Valve Driver Parameters table (*see page 75*).

Description of columns:

Column	Description
LABEL	Indicates the label of the parameter
ADDRESS	Indicates the address of the Modbus register containing the resource to access.
DATA TYPE	Indicates the size of the data in bits.
R/W	Indicates whether the resources are read/write, read only, or write only: <ul style="list-style-type: none"> ● R = read only resource. ● W = Write only resource. ● RW = Read/write resource.
CPL	<p>When the field indicates "-1", the value read by the register requires conversion because the value represents a number with a sign. In the other cases, the value is always positive or null.</p> <p>To carry out the conversion, proceed as follows:</p> <ul style="list-style-type: none"> ● If the value in the register is from 0 to 32767, the result is the value itself (zero and positive values). ● If the value in the register is from 32768 to 65535, the result is the value of the register - 65536 (negative values).
RESET	<p>Indicates whether the controller must be rebooted after the parameter has been modified.</p> <ul style="list-style-type: none"> ● Y = the controller must be rebooted to modify the parameter. ● Empty "-" = the controller does not need to be rebooted to modify the parameter.
DESCRIPTION	Description of the parameter usage.
RANGE	Describes the interval of values that can be assigned to the parameter. It can be correlated with other equipment parameters (indicated in the parameter label).
DEFAULT	Indicates the factory setting for the reference of the device.
U.M.	<p>Indicates the unit of measurement for values converted according to the rules indicated in the CPL column. The unit of measurement shown is for example purposes only, as it may change depending on the application (for example, parameters with a U.M. in °C/bar could also have %RH).</p> <p>For parameters indicated in Step unit, refer to Valve Parameters in Step Unit description (<i>see page 73</i>).</p>

Valve Parameters in Step Unit

Overview

There are three different driving modes for standard valves.

The $dE07$ parameter defines the valve driving mode:

- 0: FULL STEP
- 1: HALF STEP
- 2: MICRO STEP

Depending on the $dE07$ value, the value of the parameters which are in Steps unit is automatically internally converted by the valve driver.

In the valve driver, the parameters indicated in **Step** unit are always expressed in FULL STEP. By this, you must set them in FULL STEP even though they are given by the datasheet in HALF STEP or MICRO STEP.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Pay particular attention when you enter the value of valve parameters which are indicated in Step unit.

Failure to follow these instructions can result in equipment damage.

Usually, the data in valve datasheets are given in the corresponding valve driving modes:

Data indicated in the valve DataSheets		Parameter Value to be entered in the valve driver	
Valve driving modes	Data in Step/Pulse unit	$dE07$	Parameters in FULL STEP unit or Steps/s
FULL STEP	X Steps	0: FULL STEP	X Steps
HALF STEP	X Steps	1: HALF STEP	X / 2 Steps
MICRO STEP	X Steps	2: MICRO STEP	X / n [*] Steps
*: n to be checked in valve datasheet.			

NOTE: Unipolar valves are typically driven in HALF STEPS, usually called "1-2 phase excitation" in the valve datasheets.

Example for Unipolar Valves

Example of data found in a unipolar valve datasheet:

Pulse	Steps rotated	Phase			
		Orange	Red	Yellow	Black
1	1	Zero	HI	HI	HI
2	2	Zero	Zero	HI	HI
3	3	HI	Zero	HI	HI
4	4	HI	Zero	Zero	HI
5	5	HI	HI	Zero	HI
6	6	HI	HI	Zero	Zero
7	7	HI	HI	HI	Zero
8	8	Zero	HI	HI	Zero

Resolution: 500 Pulses

Step rate: 30~80 PPS

For correspondence between wires color of the valve and terminal pins of the valve driver, refer to valve datasheet or to the Preconfigured Unipolar Valve Configuration (*see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide*).

Unipolar valve characteristics	Comments	FREE EVEVD parameter
Unipolar Step Logic = 1-2 Phase excitation/Half step	The table shows that the recommended driving mode is Half Step.	dE07 = 1
Resolution = 500 Steps	These Steps are Half steps and not Full steps because driving mode is Half step. 1 Half Step = 0.5 Full Step 500 Half Step = 250 Full Step	dE02 = 250 (Full)Steps
Step Rate =30...80 Steps/s	These Steps are Half steps and not Full steps because driving mode is Half step. 1 Half Step = 0.5 Full Step Maximum value is usually used: 80 Half Step = 40 Full Step	dE01 = 40 (Full)Steps/s

Example for Bipolar Valves

Example of data found in a bipolar valve datasheet:

	Bipolar drive sequence					
	Step	Black	White	Red	Green	
← Open	1	12 volts	0 volts	12 volts	0 volts	Close ↑
	2	0 volts	12 volts	12 volts	0 volts	
	3	0 volts	12 volts	0 volts	12 volts	
	4	12 volts	0 volts	0 volts	12 volts	
	1	12 volts	0 volts	12 volts	0 volts	

Number of steps: 2500

Recommended step rate: 200/second (L/R), up to 400/second (current limited)

For correspondence between wires color of the valve and terminal pins of the valve driver, refer to valve datasheet or to the Preconfigured Bipolar Valve Configuration (see *FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide*).

Bipolar valve characteristics	Comments	FREE EVEVD parameter
Bipolar drive sequence = 2-Phase excitation/Full step	The table shows that the recommended driving mode is Full Step.	dE07 = 0
Resolution = 2500 Steps	These Steps are Full steps because driving mode is Full step.	dE02 = 2500 (Full)Steps
Step Rate = 200...400 Steps/s	These Steps are Full steps because driving mode is Full step. Maximum value is usually used: 400 Full Steps/s	dE01 = 400 (Full)Steps/s

Electronic Expansion Valve Driver Parameters Table

Folders

NOTE: Not all parameters listed are available depending on the accessible resources in the device.

The following tables present the valve driver parameters, divided into categories (folders):

Folder label
Acknowledgment (<i>see page 75</i>)
Valve Folder: <ul style="list-style-type: none"> • Unipolar Valve Configuration Parameters (<i>see page 80</i>) • Bipolar Valve Configuration Parameters (<i>see page 83</i>)
ON Board CAN Expansion Bus Folder (<i>see page 75</i>)
I/O Values Folder (<i>see page 77</i>)
DIP Switch Values Folder (<i>see page 75</i>)
LED Values (<i>see page 76</i>)

Acknowledgment Folder

LABEL	ADDRESS	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
Par_TAB	15716	WORD	R	-	Y	Map code	0...65535	0	Num
Par_POLI	15717	WORD	R	-	Y	Model Code	0...65535	1025	Num
Par_PCH	15719	WORD	R	-	-	Device model <ul style="list-style-type: none"> • 439 for unipolar valve drivers • 438 for bipolar valve drivers 	0...65535	-	Num
DW0(*)	9000	DWORD	R	-	-	Serial Number: First Double Word	0...FFFFFFFF _h	-	Hex
DW1(*)	9002	DWORD	R	-	-	Serial Number: Second Double Word	0...FFFFFFFF _h	-	Hex
DW2(*)	9004	DWORD	R	-	-	Serial Number: Third Double Word	0...FFFFFFFF _h	-	Hex
DW3(*)	9006	DWORD	R	-	-	Serial Number: Fourth Double Word	0...FFFFFFFF _h	-	Hex

*: Serial number of the Microcontroller. For Internal use only.

ON Board CAN Expansion Bus Folder

LABEL	ADDRESS	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
Addr_CAN_OB	15780	-	R/W	-	Y	On-board CAN Expansion Bus serial address	1...127	20	Num

DIP Switch Values Folder

LABEL	ADDRESS	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
SW6	8320	WORD	R	-	-	Dip switches state: CAN Expansion Bus Address	0...1	0	Flag
SW5	8321	WORD	R	-	-		0...1	0	Flag
SW4	8322	WORD	R	-	-		0...1	0	Flag
SW3	8323	WORD	R	-	-	Dip switches state: CAN Expansion Bus Baud rate	0...1	0	Flag
SW2	8324	WORD	R	-	-		0...1	0	Flag

LED Values Folder

LABEL	ADDRESS	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
LED1	8640	WORD	R	-	-	LED 1 state: <ul style="list-style-type: none">● 0: Off● 1: On● 2: Blink	0...2	0	Num
LED2	8641	WORD	R	-	-	LED 2 state: <ul style="list-style-type: none">● 0: Off● 1: On● 2: Blink	0...2	0	Num
LED3	8642	WORD	R	-	-	LED 3 state: <ul style="list-style-type: none">● 0: Off● 1: On● 2: Blink	0...2	0	Num

I/O Values Folder

Overview

The valve driver has 6 parameters per valves to control the valves.

These parameters can be read/write automatically by the controller once dedicated variables are configured in COMMAND tab (*see page 93*) and STATUS tab (*see page 94*).

I/O Values Folder

LABEL	ADDRESS for 1st valve	ADDRESS for 2nd valve	DATA TYPE	RW	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
Open_at	8336	8340	WORD	R	-	-	0: Closed ... 1000: Open	0...1000	0	‰
Status	8337	8341	WORD	R	-	-	Status code: <ul style="list-style-type: none"> ● Bit 0: Command received ● Bit 1: Command completed ● Bit 2: Command completed successfully ● Bit 3: Action in progress cannot be interrupted ● Bit 4: Valve position is known ⁽¹⁾ ● Bit 8: Error detected 	0...65535	0	Num
Diagnostic	8338	8342	WORD	R	-	-	Diagnostic code (<i>see page 77</i>)	0...65535	0	Num
Info	8339	8343	WORD	R	-	-	Reserved	0...65535	0	Num
Open_at_wr	8448	8450	WORD	R/W	-	-	Opening set point	0...1000	0	‰
Command	8449	8451	WORD	R/W	-	-	Write hardware configuration parameters command: <ul style="list-style-type: none"> ● 0: Rearm ● 1: Activate and synchronize ⁽²⁾ ● 2: Reserved 	0...2	0	Num

(1): If Status bit 4 is reset, it is not possible to drive the valve. The bit is reset after power-on and it is set after an “Activate and synchronize” command and the synchronize (zeroing) procedure end. The command is managed and sent by the `EEVDSettingsU/B` AFBs

(2): After “Activate and synchronize” command, use Rearm (`Command = 0`), just after Status bit 2 = 1 (Command completed successfully) or after a timeout of 600 s (command not completed). This procedure is implemented by `EEVDSettingsU/B` AFBs.

Diagnostic Parameter and Internal Alarm Management

Diagnostic Parameter		Alarm	EVEVD•000500			Valve Driver Action	Rearm Condition
Bit	Description		1U	1B	2B		
0	Chip does not respond	Detected error on bipolar driver chip: chip does not respond.	No	Yes	Valve stops in current position	Automatic	
1	Thermal shutdown						
2	Predriver detected error ^(*)						
3	Undervoltage Lockout ^(*)						
4	Reserved						
5	Overcurrent	Valve coil in short-circuit	Yes		Valve stops in current position	Manual: Alarm cause must be solved and a new “Activate and synchronize” command must be sent. Automatic rearm is managed by the AFB <code>EEVDAlarmMgmt</code> .	
6	Reserved						

(*) Contact your local support.

Diagnostic Parameter		Alarm	EVEVD••000500			Valve Driver Action	Rearm Condition
Bit	Description		1U	1B	2B		
7	Max number of valve direction changing (only if n12 > 0)	Max number of valve direction changing achieved	Yes			Moves the valve to the emergency position n30. Then any activity on the valve is stopped.	Automatic
8	Quantity of EVEVDBA000000 connected <> parameter n36 value or EVEVDBA000000 exceeds charging time timeout	EVEVDBA000000 inoperable	Yes			Non-blocking alarm	Automatic
9	Power supply outage If EVEVDBA000000 connected and charged with enough energy	Power supply not detected	Yes			Movement towards the emergency position n30 is executed only if n31=1 otherwise no action is done. If n17=0, the value of the "maximum speed in emergency closing" is equal to dE01. If n17 >0 the movement towards the emergency position n30 is done using n17 speed.	Automatic
10	EVEVDBA000000 degraded due to: <ul style="list-style-type: none"> the charge is too fast or if the last time that there has been an emergency movement followed by power fail, this has not been completed. The value has persisted in EEPROM so an emergency movement must be repeated successfully to cancel it.	EVEVDBA000000 degraded (*)	Yes			Non-blocking alarm	Automatic
11	Configuration error n16 value not compatible with EVEVD••000500 model. or: <ul style="list-style-type: none"> Unipolar: n32, n33, n34, and n35 value combination not allowed Bipolar: dE04<0 and at least one of n27, n28, n29 not 0 	Configuration error	Yes			Valve stops in current position	Manual: Alarm cause must be solved and a new "Activate and synchronize" command must be sent. Automatic rearm is managed by the AFB EEVDSettingsU or EEVDSettingsB after sending new parameter values.
12	Disconnection on W1+ or W1-	Valve disconnected Detected only when valve is in synchronization phase.	Yes			Driver moves the EEV in any case. By default, the AFB EEVDCntrl considers this alarm as a stopping condition and try to move the valve to the "Alarm" position, where the EEV rest until the alarm disappear.	Manual: The alarm is detected and/or reset only in synchronization phase. If detected, it is maintained active until the next synchronization, as the disconnection is checked again and alarm is confirmed or reset. Automatic rearm is managed by the AFB EEVDAlarmMgmt that by default is set to periodically drive a synchronization phase to check again the alarm when present.
13	Disconnection on W2+ or W2-						
(*) Contact your local support.							

Diagnostic Parameter		Alarm	EVEVD••000500			Valve Driver Action	Rearm Condition
Bit	Description		1U	1B	2B		
14	CAN not working, communication with master lost (not readable via CAN)	Loss of communication on CAN expansion bus Detected only after first connection with CAN master controller	Yes			Moves the valve to the emergency position n30. Then any activity on the valve is stopped.	Automatic
15	Reserved						

(*) Contact your local support.

The alarm rearm can be:

- **Manual:** The unit does not clear the alarm or resume operation automatically. Alarm cause must be solved and a new "Activate and synchronize" command must be sent.
- **Automatic:** The unit clears the alarm and resume operation (if the alarm triggered a stop of the movement or a movement to an emergency position) automatically. Consider that the system may wait up to 20 seconds (starting from the end of the error handling, in case of movement to an emergency position this starts at the end of the movement) to update the alarm status and resume operation.

Unipolar Valve Configuration Parameters

Overview

The EVEVD**000500 devices can use preconfigured valve parameters for a set of Electronic valves.

The possible preconfigured valve parameters are stored in the FREE Advance Logic Controller non-volatile memory through the `EEVDSettingsU` AFB. The EVEVD**000500 is configured by its FREE Advance Logic Controller master to work with the chosen valve.

dE00

The valve parameters depend on the `dE00` value:

LABEL	ADDRESS	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
dE00	15800	WORD	R/W	-	-	Valve selection: <ul style="list-style-type: none"> dE00 = 0: The valve parameters <code>dE01...n40</code> are customizable. Refer to Customizable Unipolar Valve Configuration Parameters (<i>see page 80</i>). dE00 ≠ 0: The valve parameters <code>dE01...n40</code> are preconfigured. Refer to Preconfigured Unipolar Valve Configuration Parameters (<i>see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide</i>). 	0...255	0	Num

NOTE:

The `dE00` parameter read/write access depends on the software used:

- with FREE Studio Plus, `dE00` is in R/W by using `EEVDSettingsU/B` AFB.
- with FREE Studio Plus Installer, `dE00` is in Read only.

Customizable Unipolar Valve Configuration Parameters

Valve parameters if `dE00 = 0`

LABEL	ADDRESS Valve 1	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
dE01	15801	UINT	R/W	-	-	Maximum speed. Defines the maximum valve motor speed to allow step precision and integrity.	0...999	40	Steps/s
dE02	15802	UINT	R/W	-	-	Full opening. Defines the maximum number of valve steps. The total travel refers to the FULL STEP mode (<code>dE07=0</code>). The valve opening is complete when this value is reached.	0...9990	250	Steps
dE03	15803	UINT	R/W	-	-	Extra movement in full closure. Defines the number of extra valve steps beyond the total closing position to allow a correct total closure. A total closure command implies the valve positioned to zero and a further number of steps <code>dE03</code> .	0...999	0	Steps
dE04	15804	INT	R	-1	-	Reserved	-560 ... 850	260	mA
dE05	15805	UINT	R	-	-	Reserved	0...999	46	Ohm
dE06	15806	UINT	R	-	-	Reserved	0...850	0	mA

LABEL	ADDRESS Valve 1	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
dE07	15807	UINT	R/W	-	-	Type of stepper motor control. Defines the driving modes: <ul style="list-style-type: none"> ● 0: FULL STEP ● 1: HALF STEP ● 2: Reserved For more details, refer to the technical documentation of the electronic valve.	0...2	1	Num
dE08	15808	UINT	R/W	-	-	Duty cycle. In the case of valve superheat, reduce the enabling duty cycle to allow it to cool down.	0...100	100	%
dE09	15809	UINT	R/W	-	-	Acceleration/deceleration Defines the acceleration/deceleration in motor start/stop. The time between one step and the next is reduced by dE09 at each step until dE01 is reached. If dE09 = 0 acceleration is not applied.	0...999	0	Num
dE80	15810	UINT	R/W	-	-	Minimum motor speed for acceleration/deceleration To be modified only if dE09 >0	0...999	0	Steps/s
n10	15811	UINT	R/W	-	-	Pause time.	0...999	0	ms
n11	15812	UINT	R/W	-	-	Extra movement in full closure every n15 hours working.	0...9990	30	Steps
n12	15813	UINT	R/W	-	-	Change direction counter limit.	0...9990	0	Num
n13	15814	UINT	R/W	-	-	Extra movement in full opening. Related to bit 7 of Diagnostic Parameters (see page 77).	0...9990	0	Steps
n14	15815	UINT	R/W	-	-	Duty cycle period of activation/deactivation.	0...9990	0	Sec/10
n15	15816	UINT	R/W	-	-	Period of periodical synchronization. <ul style="list-style-type: none"> ● 0 = Function is disabled 	0...9990	0	Hours
n16	15817	UINT	R	-	-	Unipolar/Bipolar valve selection. <ul style="list-style-type: none"> ● 1 = Bipolar ● 2 = Unipolar 	1...2	2	Num
n17	15818	UINT	R/W	-	-	Maximum speed in emergency closing. If set at 0, referred value is dE01	0...999	40	Steps/s
n18	15819	UINT	R	-	-	Reserved	0...1	0	Num
n19	15820	UINT	R	-	-	Reserved	0...4095	3072	Num
n20	15821	UINT	R	-	-	Reserved	0...2047	256	Num
n21	15822	UINT	R	-	-	Reserved	0...512	50	Num
n22	15823	UINT	R	-	-	Reserved	0...512	288	Num
n23	15824	UINT	R	-	-	Reserved	0...2047	1296	Num
n24	15825	UINT	R	-	-	Reserved	0...4095	2562	Num
n25	15826	UINT	R	-	-	Reserved	0...4095	240	Num
n26	15827	UINT	R/W	-	-	Periodical override mode: <ul style="list-style-type: none"> ● 0=after n15 period with Open_at_wr = 0 ● 1=after n15 period) 	0...1	0	Flag
n27	15828	UINT	R/W	-	-	Reserved	0...850	0	mA
n28	15829	UINT	R/W	-	-	Reserved	0...1000	0	‰
n29	15830	UINT	R/W	-	-	Reserved	0...3	0	Num
n30	15831	UINT	R/W	-	-	Emergency Opening percentage.	0...1000	0	‰
n31	15832	UINT	R/W	-	-	Behaviour on power fail: <ul style="list-style-type: none"> ● 0 = no action, an alarm is generated Refer to bit 9 of Diagnostic Parameters (see page 77) <ul style="list-style-type: none"> ● 1 = close 	0...1	1	Num

LABEL	ADDRESS Valve 1	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
n32	15833	UINT	R/W	-	-	Unipolar W1+ allocation. <ul style="list-style-type: none"> ● 0=no redirection ● 1=redirect to W1+ ● 2=redirect to W1- ● 3=redirect to W2+ ● 4=redirect to W2- 	0...4	0	Num
n33	15834	UINT	R/W	-	-	Unipolar W1- allocation. <ul style="list-style-type: none"> ● 0=no redirection ● 1=redirect to W1+ ● 2=redirect to W1- ● 3=redirect to W2+ ● 4=redirect to W2- 	0...4	0	Num
n34	15835	UINT	R/W	-	-	Unipolar W2+ allocation. <ul style="list-style-type: none"> ● 0=no redirection ● 1=redirect to W1+ ● 2=redirect to W1- ● 3=redirect to W2+ ● 4=redirect to W2- 	0...4	0	Num
n35	15836	UINT	R/W	-	-	Unipolar W2- allocation. <ul style="list-style-type: none"> ● 0=no redirection ● 1=redirect to W1+ ● 2=redirect to W1- ● 3=redirect to W2+ ● 4=redirect to W2- 	0...4	0	Num
n36	15837	UINT	R/W	-	-	Number of Battery Backup modules. NOTE: This parameter is overwritten by <code>i_usi_battery</code> . This is a parameter of the driver, not of the valve. Input of the FB settings is preponderant respect to parameter value.	0...2	0	Num
n37	15838	UINT	R/W	-	-	Valve energization time at startup.	0...65535	0	ms
n38	15839	UINT	R/W	-	-	Valve energization time at stop.	0...65535	0	ms
n39	15840	UINT	R	-	-	Reserved	0...65535	0	Num
n40	15841	UINT	R	-	-	Reserved	0...65535	0	Num

Bipolar Valve Configuration Parameters

Overview

The EVEVD••000500 devices can use preconfigured valve parameters for a set of Electronic valves. The possible preconfigured valve parameters are stored in the FREE Advance Logic Controller non-volatile memory through the `EEVDSettingsB` AFB. The EVEVD••000500 is configured by its FREE Advance Logic Controller master to work with the chosen valve.

WARNING

FIRE HAZARD

Only use bipolar valves that doesn't exceed 9 W as nominal input power per phase.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

dE00

The valve parameters depend on the `dE00` value:

LABEL	ADDRESS Valve 1	ADDRESS Valve 2	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
dE00	15800	16000	WORD	R/W	-	-	Valve selection: <ul style="list-style-type: none"> dE00 = 0: The valve parameters <code>dE01...n40</code> are customizable. Refer to Customizable Bipolar Valve Configuration Parameters (<i>see page 83</i>). dE00 ≠ 0: The valve parameters <code>dE01...n40</code> are preconfigured. Refer to Preconfigured Bipolar Valve Configuration Parameters (<i>see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide</i>). 	0...255	0	Num

NOTE:

The `dE00` parameter read/write access depends on the software used:

- with FREE Studio Plus, `dE00` is in R/W by using `EEVDSettingsU/B` AFB.
- with FREE Studio Plus Installer, `dE00` is in Read only.

Customizable Bipolar Valve Configuration Parameters

Valve parameters if `dE00 = 0`

LABEL	ADDRESS Valve 1	ADDRESS Valve 2	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
dE01	15801	16001	UINT	R/W	-	-	Maximum speed. Defines the maximum valve motor speed to allow step precision and integrity.	0...999	200	Steps/s
dE02	15802	16002	UINT	R/W	-	-	Full opening. Defines the maximum number of valve steps. The total travel refers to the FULL STEP mode (<code>dE07=0</code>). The valve opening is complete when this value is reached.	0...9990	2500	Steps
dE03	15803	16003	UINT	R/W	-	-	Extra movement in full closure. Defines the number of extra valve steps beyond the limit switch to allow a correct total closure. A total closure command implies the valve positioned to zero and a further number of steps <code>dE03</code> .	0...999	0	Steps

LABEL	ADDRESS Valve 1	ADDRESS Valve 2	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
dE04	15804	16004	INT	R/W	-1	-	Winding maximum current. Defines the maximum current per phase utilized by the valve (maximum torque). Negative current value: the maximum current is set to the value with no sign (absolute) dE04 with an extra 50% with the valve movement command (starting or end point) within 5% of total opening, to a value equal to the absolute value of dE04 for the other movements.	-560 ... 850	90	mA
dE05	15805	16005	UINT	R	-	-	Reserved	0...999	100	Ohm
dE06	15806	16006	UINT	R/W	-	-	Winding holding current. Defines the phase circulating current in the valve stop condition (minimum torque).	0...850	0	mA
dE07	15807	16007	UINT	R/W	-	-	Type of stepper motor control. Defines the driving modes: <ul style="list-style-type: none"> ● 0: FULL STEP ● 1: HALF STEP ● 2: MICRO STEP For more details, refer to the technical documentation of the electronic valve.	0...2	0	Num
dE08	15808	16008	UINT	R/W	-	-	Duty cycle. In the case of valve superheat, reduce the enabling duty cycle to allow it to cool down.	0...100	100	%
dE09	15809	16009	UINT	R/W	-	-	Acceleration/deceleration Defines the acceleration/deceleration in motor start/stop. The time between one step and the next is reduced by dE09 at each step until dE01 is reached. If dE09 = 0 acceleration is not applied.	0...999	0	Num
dE80	15810	16010	UINT	R/W	-	-	Minimum motor speed for acceleration/deceleration To be modified only if dE09 >0	0...999	0	Steps/s
n10	15811	16011	UINT	R/W	-	-	Pause time.	0...999	25	ms
n11	15812	16012	UINT	R/W	-	-	Extra movement in full closure every n15 hours working.	0...9990	100	Steps
n12	15813	16013	UINT	R/W	-	-	Change direction counter limit.	0...9990	0	Num
n13	15814	16014	UINT	R/W	-	-	Extra movement in full opening. Related to bit 7 of Diagnostic Parameters (see page 77).	0...9990	0	Steps
n14	15815	16015	UINT	R/W	-	-	Duty cycle period of activation/deactivation.	0...9990	0	s/10
n15	15816	16016	UINT	R/W	-	-	Period of periodical synchronization. <ul style="list-style-type: none"> ● 0 = Function is disabled 	0...9990	0	Hours
n16	15817	16017	UINT	R	-	-	Unipolar/Bipolar valve selection. <ul style="list-style-type: none"> ● 1 = Bipolar ● 2 = Unipolar 	1...2	1	Num
n17	15818	16018	UINT	R/W	-	-	Maximum speed in emergency closing. If set at 0, referred value is dE01	0...999	0	Steps/s
n18	15819	16019	UINT	R	-	-	Reserved	0...1	0	Num
n19	15820	16020	UINT	R/W	-	-	Reserved	0...4095	3072	Num
n20	15821	16021	UINT	R/W	-	-	Reserved	0...2047	256	Num
n21	15822	16022	UINT	R/W	-	-	Reserved	0...512	50	Num
n22	15823	16023	UINT	R/W	-	-	Reserved	0...512	288	Num
n23	15824	16024	UINT	R/W	-	-	Reserved	0...2047	1296	Num
n24	15825	16025	UINT	R/W	-	-	Reserved	0...4095	2562	Num
n25	15826	16026	UINT	R/W	-	-	Reserved	0...4095	240	Num

LABEL	ADDRESS Valve 1	ADDRESS Valve 2	DATA TYPE	R/W	CPL	RESET	DESCRIPTION	RANGE	DEFAULT	U.M.
n26	15827	16027	UINT	R/W	-	-	Periodical override mode: <ul style="list-style-type: none"> ● 0=after n15 period with <code>Open_at_wr = 0</code> ● 1=after n15 period) 	0...1	0	Flag
n27	15828	16028	UINT	R/W	-	-	Winding maximum current during boosting phase.	0...850	0	mA
n28	15829	16029	UINT	R/W	-	-	Boosting windows.	0...1000	0	‰
n29	15830	16030	UINT	R/W	-	-	Boosting mode: <ul style="list-style-type: none"> ● 0=no ● 1=open ● 2=close ● 3=both 	0...3	0	Num
n30	15831	16031	UINT	R/W	-	-	Emergency Opening percentage.	0...1000	0	‰
n31	15832	16032	UINT	R/W	-	-	Behaviour on power fail: <ul style="list-style-type: none"> ● 0 = no action, an alarm is generated Refer to bit 9 of Diagnostic Parameters (see page 77) ● 1 = close 	0...1	1	Num
n32	15833	16033	UINT	R/W	-	-	Reserved	0...4	0	Num
n33	15834	16034	UINT	R/W	-	-	Reserved	0...4	0	Num
n34	15835	16035	UINT	R/W	-	-	Reserved	0...4	0	Num
n35	15836	16036	UINT	R/W	-	-	Reserved	0...4	0	Num
n36	15837	-	UINT	R/W	-	-	Number of Battery Backup modules. NOTE: Parameter value for valve 2 is not used. NOTE: This parameter is overwritten by <code>i_usi_batterynr</code> . This is a parameter of the driver, not of the valve. Input of the FB settings is preponderant respect to parameter value.	0...2	0	Num
n37	15838	16038	UINT	R/W	-	-	Valve energization time at startup.	0...65535	25	ms
n38	15839	16039	UINT	R/W	-	-	Valve energization time at stop.	0...65535	25	ms
n39	15840	16040	UINT	R/W	-	-	Reserved	0...65535	0	Num
n40	15841	16041	UINT	R/W	-	-	Reserved	0...65535	0	Num

Part V

Commissioning

Chapter 11

Commissioning

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Commissioning	90
Add a EVEVD••000500 to your project	91
Configure the EVEVD••000500	93
Advanced Configuration and Programming the EVEVD••000500	97
PC Connection	98

Commissioning

Overview

EVEVD••000500 valve drivers do not embed program. To be used, the settings and control commands must be done by the FREE Advance controllers.

To do it, you must use in the application the 3 dedicated Application Function Blocks:

- `EEVDSettingsU` to manage and monitor parameters and settings of the unipolar EEVD EVEVD1U000500.
- `EEVDSettingsB` to manage and monitor parameters and settings of the bipolar EEVD EVEVD1B000500 and EVEVD2B000500.
- `EEVDCntrl` to control the EEV actuator for the application purpose, providing the communication framework with the EEVD EVEVD••000500.

First Commissioning Procedure

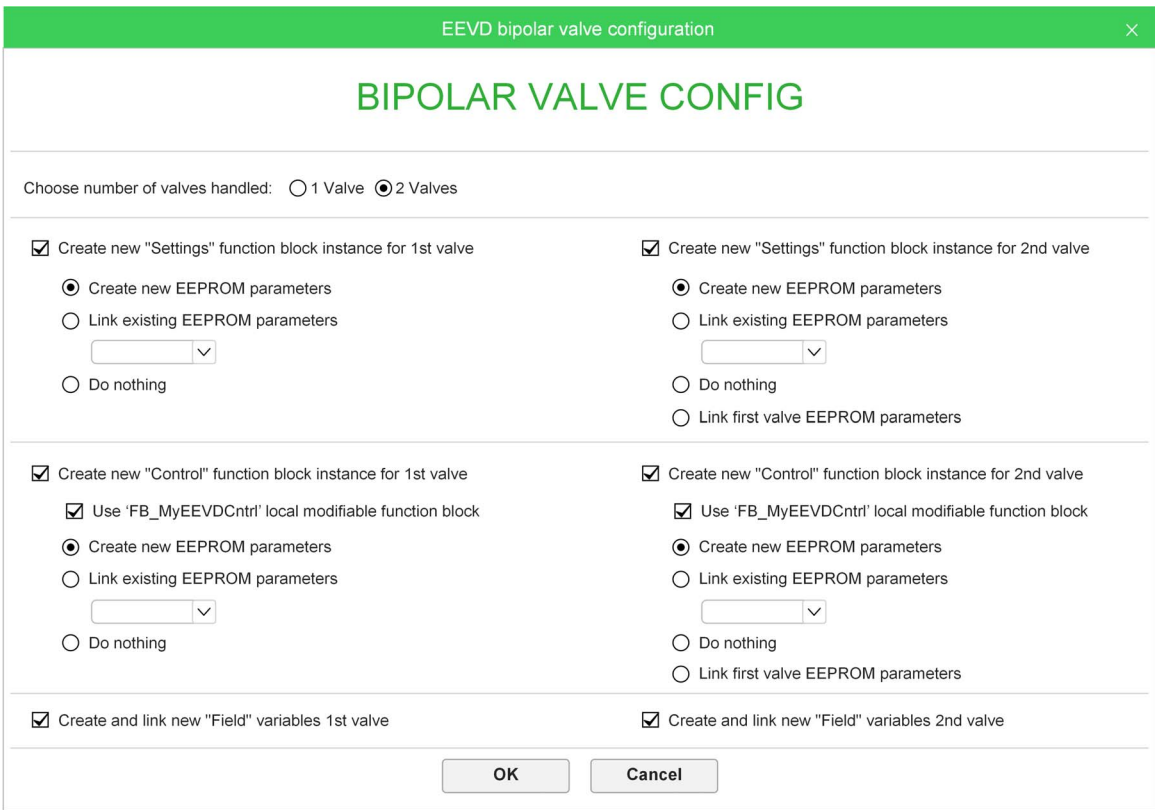
First commission procedure for a EVEVD••000500:

Step	Action
1	Create a project with your FREE Advance controller (<i>see FREE Studio Plus, Operating Guide</i>).
2	Add a EVEVD••000500 to your project (<i>see page 91</i>).
3	Configure the EVEVD••000500 (<i>see page 93</i>).
4	Advanced configuration and Program your application (<i>see page 97</i>).
5	Connect your FREE Advance controller to FREE Studio Plus software. Refer to FREE Advance hardware Guide (<i>see FREE Advance Logic Controller, Hardware Guide</i>)
6	Load the application in the FREE Advance controller (<i>see FREE Studio Plus, Operating Guide</i>)
7	Run the application.

Add a EVEVD••000500 to your project

Add a EVEVD••000500 to your project

Add a EVEVD••000500 to your project:

Step	Action
1	In the configuration tab, in the Resources window, select CAN exp bus .
2	Configure FREE Advance controller CAN Expansion Bus in Master (for field) mode.
3	Configure the Baud rate and the Master Settings .
4	<p>From the Catalog window, select one valve driver node and drop it on the CAN Expansion bus node. You can choose between two valve driver type:</p> <ul style="list-style-type: none"> • Expansion EVEVD1U: for EVEVD1U000500 that can control one unipolar valve. • Expansion EVEVD1B/2B: for EVEVD1B000500 and EVEVD2B000500 that can control respectively 1 or 2 bipolar valves. <p>Result: A window appears:</p> 
5	<p>Check Create new "Settings" function block instance: creates the <code>EEVDSettingsUEEV•</code> / <code>EEVDSettingsBEEV•</code> function blocks instance (usable in the Programming tab). You can in addition manage associated EEPROM parameters by choosing options. For more information, refer to <code>EEVDSettingsU/B</code> details during commissioning (see page 113).</p>
6	<p>Check Create new "Control" function block instance: creates the <code>EEVDCntrlEEV•</code> function blocks instance (usable in the Programming tab). You can in addition manage associated EEPROM parameters by choosing options. For more information, refer to <code>EEVDCntrl</code> details during commissioning (see page 123).</p>

Step	Action
7	<p>Check Create and link new "Field" variables to creates the variables:</p> <ul style="list-style-type: none"> • iEEVDAlarmsEEVx • iEEVDOpeningReqEEVx • iEEVDCommandsEEVx • iEEVDInfoEEVx • iEEVDOpeningCurrEEVx • iEEVDStatusEEVx <p>NOTE: x corresponds to the valve number.</p> <p>These variables are created in:</p> <ul style="list-style-type: none"> • Resources → I/O Mapping → Field in Configuration tab • IO_Mapping Global shared variables in the Programming tab <p>These variables automatically fill PLC Var fields of COMMAND tab (<i>see page 93</i>) and STATUS tab (<i>see page 94</i>) to read/write the I/O value parameters (<i>see page 77</i>).</p>
8	<p>Click OK to validate.</p> <p>If you click Cancel, the valve driver is created but none function block are instantiated nor parameters are created or assigned.</p> <p>Result: a valve driver node is created into the CAN Exp. bus node.</p>

Reminder of nodes, function blocks, and parameters naming:

- Nodes name are incremented by type of valve driver (starting from 1).
- Function block and parameter are incremented independently of the type of valve driver (starting from 0).

Namings

Examples of nodes, function blocks, and parameters naming:

Added Valve Driver	Type	Names
Expansion EVEVD1B/2B with 2 valves	Node	Expansion EVEVD1B/2B_1
	Function blocks	EEVDSettingsBEEV0 EEVDCntrlEEV0 EEVDSettingsBEEV1 EEVDCntrlEEV1
	Parameter	dE00: E2_uidE00EEV0 dE00: E2_uidE00EEV1
Expansion EVEVD1U	Node	Expansion EVEVD1U_1
	Function blocks	EEVDSettingsUEEV2 EEVDCntrlEEV2
	Parameter	dE00: E2_uidE00EEV2
Expansion EVEVD1B/2B with 1 valve	Node	Expansion EVEVD1B/2B_2
	Function blocks	EEVDSettingsBEEV3 EEVDCntrlEEV3
	Parameter	dE00: E2_uidE00EEV3

For example: the first parameter dE00 is created as E2_uidE00EEV0:

- E2_: for EEPROM Parameters
- ui: the IEC type of variable (in the example, ui for unsigned integer)
- dE00: parameter label, refer to the parameter table (*see page 75*)
- EEV: Electronic Expansion Valve
- 0: number of valve added.

Configure the EVEVD••000500

Overview

In this configuration example, you want to use EVEVD••000500 valve driver as expansion of a FREE Advance controller.

In the **Configuration** tab, in the **Resources** window, under the **CAN exp bus** node, select the valve driver.

Expansion EVEVD1B/2B configuration is divided into four tabs:

- **GENERAL**: to configure the network settings (*see page 93*).
- **COMMAND**: to configure the command variables (*see page 93*).
- **STATUS**: to configure the status and diagnostic variables (*see page 94*).
- **ADVANCED SETTINGS**: to add and remove variables (*see page 94*).

GENERAL Tab

GENERAL tab permits you to configure the network settings.

GENERAL tab of **Expansion EVEVD1B/2B**:

EXPANSION EVEVD1B/2B GENERAL CONFIGURATION

GENERAL COMMAND STATUS ADVANCED SETTINGS

Network settings

Node number (1...122)

Advanced >>>

USER DEFINED Mode

SYNC Mode

EVENT Mode

CYCLIC Mode ms

In the **GENERAL** tab, you can configure:

- Node number (1...122).
- **CAN Mode**

COMMAND Tab

COMMAND tab permits you to configure the command variables.

This corresponds to the commands sent by the controller to the valve driver.

COMMAND tab of **Expansion EVEVD1B/2B**:

EXPANSION EVEVD1B/2B COMMAND

GENERAL **COMMAND** STATUS ADVANCED SETTINGS

Configuration 1 Valve 2 Valves

First valve


	PLC Var	DataBlock
Command	<input type="text" value="iEEVDCommandsEEV0"/>	<input type="text" value="QW10.0"/>
Open at [%]	<input type="text" value="iEEVDOpeningReqEEV0"/>	<input type="text" value="QW10.1"/>


Second valve

	PLC Var	DataBlock
Command	<input type="text" value="iEEVDCommandsEEV1"/>	<input type="text" value="QW10.2"/>
Open at [%]	<input type="text" value="iEEVDOpeningReqEEV1"/>	<input type="text" value="QW10.3"/>

NOTE: These field are automatically filled by selecting **Create and link new "Field" variables** when you Add a EVEVD••000500 to your project (*see page 91*).

FREE Studio Plus software knows the EVEVD••000500 valve drivers dictionary. Each object can be assigned to a **PLC variable**.

Step	Action
1	Click  Assign button.
2	Select the PLC variable that you want to assign to the PLC object: NOTE: It is possible to apply a filter to the choice list by entering a string of characters.
3	Click OK button to validate.
4	The PLC Var name is added and its address is displayed in the DataBlock field.

NOTE: Click  **Unassign** button to remove the assignment.

STATUS Tab

STATUS tab permits you to configure the status and diagnostic variables.

This corresponds to the status and diagnostic of the valve driver read by the controller.

STATUS tab of **Expansion EVEVD1B/2B**:

EXPANSION EVEVD1B/2B STATUS

GENERAL
COMMAND
STATUS
ADVANCED SETTINGS

Configuration 1 Valve 2 Valves

First Valve

	PLC Var	↕ ↖	DataBlock
Status	<input type="text" value="iEEVDStatusEEV0"/>		<input type="text" value="IW11.1"/>
Open at [%]	<input type="text" value="iEEVDOpeningCurrEEV0"/>		<input type="text" value="IW11.0"/>
Info	<input type="text" value="iEEVDInfoEEV0"/>		<input type="text" value="IW11.2"/>
Diagnostic	<input type="text" value="iEEVDAlarmsEEV0"/>		<input type="text" value="IW11.3"/>

Second Valve

	PLC Var	↕ ↖	DataBlock
Status	<input type="text" value="iEEVDStatusEEV1"/>		<input type="text" value="IW11.5"/>
Open at [%]	<input type="text" value="iEEVDOpeningCurrEEV1"/>		<input type="text" value="IW11.4"/>
Info	<input type="text" value="iEEVDInfoEEV1"/>		<input type="text" value="IW11.6"/>
Diagnostic	<input type="text" value="iEEVDAlarmsEEV1"/>		<input type="text" value="IW11.7"/>

NOTE: These field are automatically filled by selecting **Create and link new "Field" variables** when you Add a EVEVD••000500 to your project (*see page 91*).

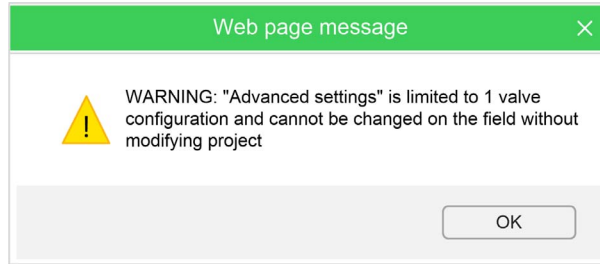
As in the **COMMAND** tab, you can assign or unassign variables to Datablocks.

ADVANCED SETTINGS Tab

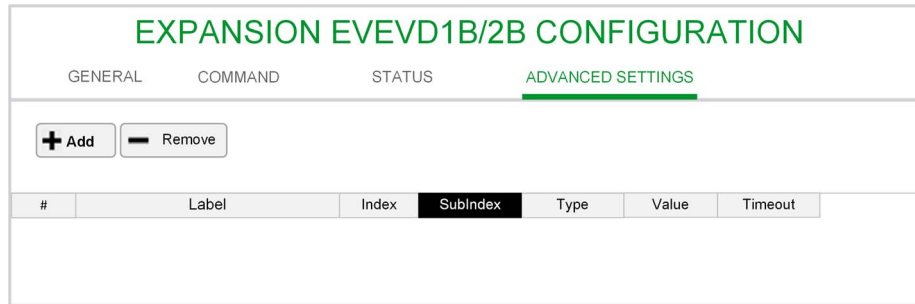
ADVANCED SETTINGS tab permits you to add and remove additional variables.

This is an alternative way of setting valve driver parameters instead of using the `EEVDSettingsU/B` Application Function Blocks.

Before displaying **ADVANCED SETTINGS** tab, a message window appears:



ADVANCED SETTINGS tab of **Expansion EVEVD1B/2B**:



To add a variable:

Step	Action														
1	Click Add .														
2	<p>Select a variable in the list:</p>														
3	<p>Click OK. Result: The variable is added:</p> <table border="1"> <thead> <tr> <th>#</th> <th>Label</th> <th>Index</th> <th>SubIndex</th> <th>Type</th> <th>Value</th> <th>Timeout</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>V1.dE01</td> <td>3db9</td> <td>0</td> <td>UINT</td> <td></td> <td>100</td> </tr> </tbody> </table>	#	Label	Index	SubIndex	Type	Value	Timeout	1	V1.dE01	3db9	0	UINT		100
#	Label	Index	SubIndex	Type	Value	Timeout									
1	V1.dE01	3db9	0	UINT		100									

Step	Action
4	Click in the field to modify: <ul style="list-style-type: none">• The Value of the variable.• The Timeout of the variable.

To remove a variable:

Step	Action
1	Select a variable.
2	Click Remove .

For more details, refer to Configuring Using the Advanced Settings tab (*see page 101*).

Advanced Configuration and Programming the EVEVD••000500

Overview

Once the first configuration is done (*see page 93*), you have to realize the advanced configuration and the programming of your application.

You have to:

- identify the valve(s)
- Configure the valve
- Program your application

Advanced Configuration

You have to identify the valve to be controlled by the valve driver.

To do it, you should use the `EEVDSettingsU/EEVDSettingsB` Application Function Blocks.

In standard use, you can use the preconfigured parameters. For this, you just have to identify the value of the `de00` parameter.

For more details about preconfigured parameter value, refer to:

- Preconfigured Unipolar Valve Configuration Parameters (*see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide*)
- Preconfigured Bipolar Valve Configuration Parameters (*see FREE EVEVD Electronic Expansion Valve Driver, Preconfigured Valves Parameters Guide*)

For advance users, you can customize the valve parameters.

For this, you have to put the parameter `de00 = 0` and to modify the parameters:

- Customizable Unipolar Valve Configuration Parameters (*see page 80*)
- Customizable Bipolar Valve Configuration Parameters (*see page 83*)

Programming

Once the valve driver is added in the application, the valve(s) identified and their parameters configured, you have to program your application. To do it, you should use the `EEVDCntrl` application function block (*see page 121*).

PC Connection

Overview

There are two ways to configure the valve driver:

Connection type	Connection Software Accessories	Description
FREE Studio Plus software with Application Function Blocks (Recommended) or with ADVANCED SETTINGS tab (<i>see page 94</i>).	<ul style="list-style-type: none"> ● PC->controller->valve driver ● FREE Studio Plus software ● Cables: <ul style="list-style-type: none"> ○ Ethernet or USB cable to connect the PC to the controller. ○ CAN cable to connect the controller to the valve driver. 	The parameter modification is sent by the controller to the valve driver via the CAN expansion bus. For more details, refer to: <ul style="list-style-type: none"> ● Commissioning (<i>see page 87</i>). ● Maintenance (<i>see page 99</i>).
Direct connection. (For expert maintenance only)	<ul style="list-style-type: none"> ● PC->valve driver ● FREE Studio Plus Installer ● DMI programming cable 	With FREE Studio Plus Installer, you can modify the parameters “on the field”. Pay attention, these new values may be overwritten by the controller through the CAN Expansion bus. For more details about installation, refer to TTL Port description (<i>see page 63</i>).

FREE Studio Plus software Software Component

The FREE Studio Plus software development tool makes it possible to create and customize IEC 61131-3 programs for various types of application. You can download FREE Studio Plus software from [Eliwell web site download center](#). It is intended for applications in HVAC&R.

FREE Studio Plus software permits to:

- Create and manage libraries, applications, and diagnostics.
- Manage previously developed applications, upload/download applications, and modify device parameters from a serial port.

PC Connection with FREE Studio Plus software

As in a standard installation, the EVEVD••000500 is connected to a FREE Advance Logic Controller (FREE Advance).

You can connect the PC to the controller with USB or Ethernet cable.

For more details, refer to FREE Advance Hardware Guide.

Once connected, you can use FREE Studio Plus software software to:

- modify parameters in the application and download the new application in the controller. The new parameters will be sent to the valve driver depending on the application.
- modify the valve driver parameters.

Part VI

Maintenance

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
12	Maintenance	101
13	BIOS Update	103

Chapter 12

Maintenance

Maintenance

Overview

EVEVD••000500 valve drivers do not embed program. To be used, the settings and control commands must be done by the FREE Advance controllers. To do it, the EVEVD••000500 valve drivers include 2 function blocks to manage their settings and to control them.

EEVD Maintenance Use Cases

Description	Description	Connection Software Accessories
Update the program and download	Update the program and download in FREE Advance controller. According to the configuration of the <code>EEVDSettingsU/B</code> AFB, new parameters can be sent to the valve driver by the controller.	<ul style="list-style-type: none">● PC->controller● FREE Studio Plus software● Ethernet or USB cable
Customized valve driver parameters	Configuring Using the Commissioning tab <i>(see page 101)</i> .	<ul style="list-style-type: none">● PC->controller->valve driver● FREE Studio Plus software● Ethernet or USB cable
	Configuring Using the Advanced Settings tab <i>(see page 101)</i> .	
	Configuring Using The Programming Cable <i>(see page 101)</i> .	<ul style="list-style-type: none">● PC->valve driver● FREE Studio Plus Installer● DMI programming cable
Bios update	Bios update <i>(see page 103)</i>	

Configuring Using the Commissioning tab

To perform the configuration of the parameters of the EVEVD••000500, it is possible to use the **Commissioning** tab of FREE Studio Plus software.

The `EEVDSettingsU/B` AFB must have been programmed before to start the configuration.

Once connected, it allows you to modify the parameters in the EVEVD••000500.

Configuring Using the Advanced Settings tab

This is an alternative way of setting valve driver parameters instead of using the `EEVDSettingsU/B` Application Function Blocks.

To perform the configuration of the parameters of the EVEVD••000500, it is also possible to use the **Advanced Settings** in **Configuration** tab of FREE Studio Plus

The values of the parameters are downloaded to the controller with **Download all** command and sent to the EVEVD••000500 at each restart of the controller. To modify them, you must recompile the controller application. In any case, the set of values would always be one.

This configuration has a lower priority compare to the `EEVDSettingsU/EEVDSettingsB` Application Function Block. So if Application Function Block is implemented and enabled, it overwrites **Configuration** tab settings.

If you set parameters through the **Configuration** tab, a message is displayed if an Application Function Block is compiled at the same time: "settings from Advance Settings in **Configuration** tab are overwritten if Application Function Block enabled."

Configuring Using The Programming Cable

It is possible to set the parameters with DMI connected to the EVEVD••000500. This has lower priority compare to `EEVDSettingsU/EEVDSettingsB` Application Function Blocks, the Application Function Blocks must be disabled before to use this type of configuration.

EEVDSettingsU/EEVDSettingsB Application Function Block generates an alarm if the EEVD valve configuration parameters are modified after the controller restart.

NOTE: The controller detects that the EEVD configuration parameters have been modified with a method different than Application Function Block. It is necessary to program how the machine reacts in such a case (for example: stop the machine or set an alarm).

NOTE: If EEVDSettingsU/B are not used in application, an “Activate and synchronize” command (*see page 77*) must be implemented in the application.

Chapter 13

BIOS Update

Device BIOS Update

Overview

There is one way to update the EVEVD••000500 valve driver BIOS:

- Downloading into the EVEVD••000500 from PC with FREE Studio Plus Installer.

⚠ WARNING
UNINTENDED EQUIPEMENT OPERATION
Prior to update the EVEVD••000500 valve driver BIOS, disconnect the valve driver from the controller. Failure to follow these instructions can result in death, serious injury, or equipment damage.

Care must be taken and provisions made for use of this product as a control device to avoid inadvertent consequences of commanded machine operation, controller state changes, or alteration of data memory or machine operating parameters.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
<ul style="list-style-type: none">• Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.• You must have a complete understanding of the application and the machine before attempting to control the application remotely.• Isolate your industrial network from other networks inside your company.• Take the precautions necessary to assure that you are operating remotely on the intended machine by having clear, identifying documentation within the application and its remote connection. Failure to follow these instructions can result in death, serious injury, or equipment damage.

Consult the [Schneider Electric Cybersecurity Best Practices \(https://www.eliwell.com/download/downloader.php?cat=prd_docs&id=1938\)](https://www.eliwell.com/download/downloader.php?cat=prd_docs&id=1938) for additional information.

Eliwell and Schneider Electric adhere to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

The controller can be upgraded only with authenticated Eliwell files. In case the authenticity check fails the controller regulates with current version.

NOTICE
UNINTENDED EQUIPMENT OPERATION
Use authenticated Eliwell files only. Failure to follow these instructions can result in equipment damage.

Download BIOS from PC

Steps to download BIOS from PC:

Step	Action
1	Supply the EVEVD••000500 via its 24 Vac/dc power supply connector.
2	Connect EVEVD••000500 to a PC using the DMI with the yellow cable.

Step	Action
3	Open FREE Studio Plus Installer software. Click Scan . Wait until the list of scanned devices appears. Click Select of one device founded.
4	Add, for instance, an EVEVD**000500 target to the project. Select the correct target device.
5	Select the name of the target.
6	Click BIOS download . Click Browse... to select the BIOS file. The BIOS files links is: <C:\Programs>\Eliwell\free Studio Plus\Catalog\TM172EVEVxx\Firmware
7	Open the .bin file you want to download.
8	Click Download button. The operation may take a few minutes. If the download terminates successfully, a confirmation is displayed.
9	Disconnect the EVEVD**000500 from the PC.

Part VII

Valve Drivers Application Function Blocks

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
14	EVEVD**000500 Application Function Block Overview	107
15	EVEVD**000500 Settings Management: <code>EEVDSettingsU/EEVDSettingsB</code>	111
16	EVEVD**000500 Application Control: <code>EEVDCntrl</code>	121
17	EVEVD**000500: CAN Expansion Bus Monitoring <code>CheckComAlarm</code>	133
18	EVEVD**000500: Standard Application Control Algorithm <code>EEVSHCntrl</code>	135
19	EVEVD**000500 Evaporator Temperature and Superheat Calculation: <code>SuperHeatCalc</code>	157
20	EVEVD**000500 Alarm Management: <code>EEVDAlarmMgmt</code>	163

Chapter 14

EVEVD••000500 Application Function Block Overview

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Application Function Block Library	108
Regulation Overview	110

Application Function Block Library

Functional Architecture

This Application Function Block library is aimed to provide the necessary control and management routines for Electronic Expansion Valves (EEVs) in a refrigeration circuit.

The typical usage of the EEV is to control superheat after evaporator. But other variables may be also regulated in parallel or standalone, for example Maximum Operating Pressure (MOP) in suction line or air/water temperature in the refrigerated environment, or subcooling.

This library is designed to control superheat with MOP and secondary fluid (air/water) temperature.

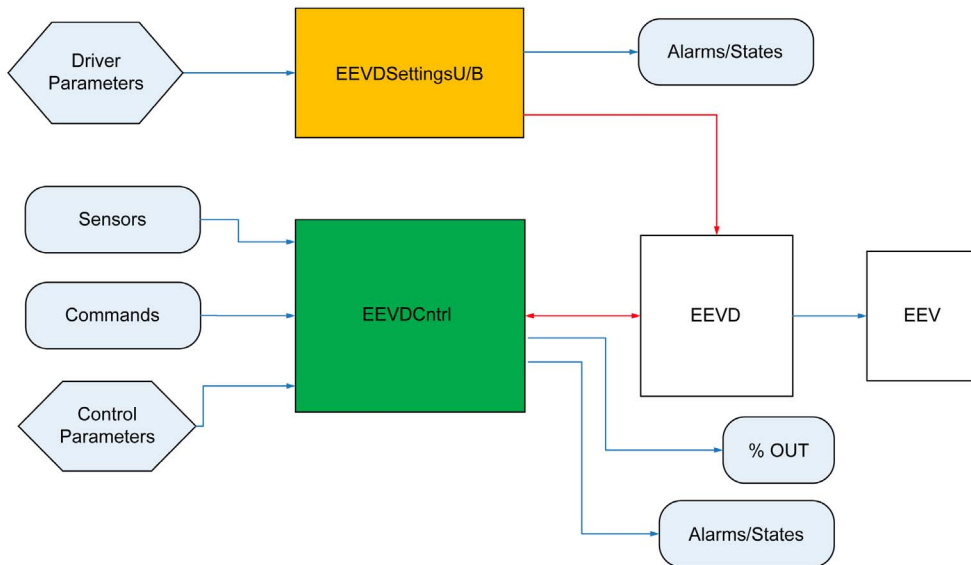
The EEV can be a pulse (ON/OFF) valve or a stepper motor valve. In both cases, the control algorithm provides an opening degree (percentage) with range from 0.0 % to 100.0 %.

- Pulse valve: the valve is modulated activating a digital output with pulse width modulation (PWM) signal. The control algorithm calculates the percentage of the period (parametrized, usually 6 s) during which the valve is open.
- Stepper valve: the control algorithm calculates the percentage of opening of the valve (0% = valve close, zero step; 100 % = valve open, maximum number of steps admitted). In this case, a driver is necessary to operate the valve electrically.

3 main Application Function Blocks are developed:

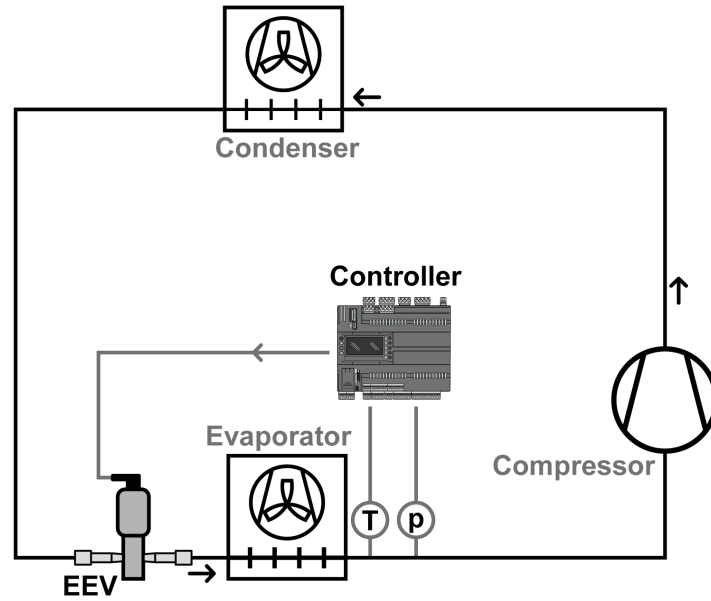
- `EEVDSettingsU` to manage and monitor parameters and settings of the unipolar EEVD `EVEVD1U000500`.
- `EEVDSettingsB` to manage and monitor parameters and settings of the bipolar EEVD `EVEVD1B000500` and `EVEVD2B000500`.
- `EEVDCntrl` to control the EEV actuator for the application purpose, providing the communication framework with the EEVD `EVEVD••000500`.

It is possible to represent the control diagram and the Application Function Blocks as follows (red lines for data exchanged through CAN serial line):



This schematic must be replicated in different instances for every EEV that have to be controlled in the application.

The following graphic is an example of a refrigeration circuit and a superheat control loop with EEV:



Regulation Overview

Refrigeration and Superheat Control Application

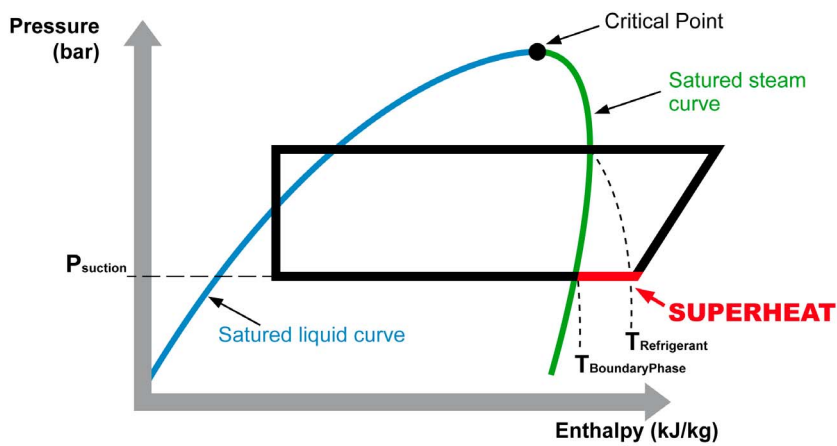
In refrigeration circuits before evaporators, an Expansion Valve (EEV or thermostatic) is installed which permits to the refrigerant liquid to expand and reduce its pressure by using a narrowing (enthalpy is kept constant).

This valve also controls the quantity of refrigerant that is evaporated inside the evaporator and is aspirated from the compressor. The degree of valve opening creates an equilibrium in the circuit, this is dependent on the desired superheat.

The superheat indicates that there is complete evaporation inside the evaporator.

Considering the pressure/enthalpy diagram, the refrigerant is evaporated when it is in the saturated steam curve. The heat absorbed from the secondary fluid in the evaporator is used for the phase transition and there is no temperature increase (latent heat). After this point, the refrigerant is superheated and there is a temperature increase; this is necessary to permit a certain margin in the aspiration of compressors to be sure that no liquid goes inside.

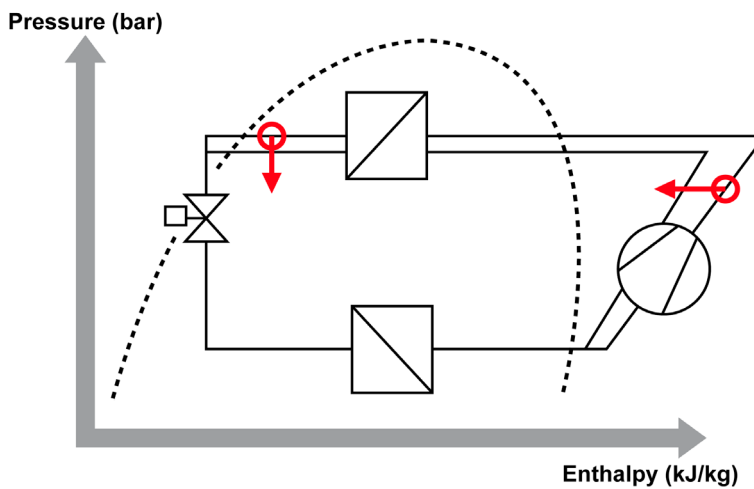
Then superheat (SH) is defined as the difference between the temperature outside the evaporator and the temperature of the saturated steam.



$$SH = T_{Refrigerant} - T_{BoundaryPhase} = T_{Refrigerant} - GasP2T(P_{suction})$$

Superheat and EEV

Changing EEV opening means continuously acting on the superheat and changing evaporator capacity, depending on current demand. With maximum load, the evaporator must be filled with the maximum quantity of liquid refrigerant to have the minimum superheat and this permit to obtain the best COP.



Chapter 15

EVEVD••000500 Settings

Management: `EEVDSettingsU/EEVDSettingsB`

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Overview	112
<code>EEVDSettingsU/B</code> Application Function Blocks Details During Commissioning	113
Input Pin Description	115
Output Pin Description	117
Troubleshooting	118
Implementation	119

Functional Overview

EEVDSettingsU/EEVDSettingsB Overview

EVEVD••000500 are expansions for AVC-AVD controllers, exchanging data through CAN expansion bus. The EEVDSettingsU Application Function Block is specific for EVEVD1U000500 Electronic Expansion Valve Drivers (EEVD) for unipolar stepper valves.

The EEVDSettingsB Application Function Block is specific for EVEVD1B000500 and EVEVD2B000500 Electronic Expansion Valve Drivers (EEVD) for bipolar stepper valves.

EEVDSettingsU/EEVDSettingsB Application Function Blocks are in charge of sending to the valve drivers the right settings.

The EEVDSettingsU/EEVDSettingsB Application Function Blocks are instantiated as many times as there are valves.

The EEVDSettingsU/EEVDSettingsB Application Function Blocks must be put inside the Background task of the PLC.

The Application Function Blocks can only be used with the FREE Advance Logic Controller (FREE Advance) and support the following valve drivers:

- EVEVD1U000500
- EVEVD1B000500
- EVEVD2B000500

Why Use the EEVDSettingsU/EEVDSettingsB Application Function Blocks?

The EEVDSettingsU/EEVDSettingsB Application Function Blocks are used to manage the following EVEVD••000500 functionality:

- Sending to the EVEVD••000500 associated to each valve the right settings selecting them from presets (internal or defined in a file), or using EEPROM parameters.
- Monitoring the parameters integrity and modification.

Features of the EEVDSettingsU/EEVDSettingsB Application Function Blocks

The EEVDSettingsU/EEVDSettingsB Application Function Block facilitate the following features:

- Pushes the settings automatically at restart or set on controller side during runtime.
- Writes the parameters via CAN.
- Send an “Activate and Synchronize” command if Status bit 4 “Valve position is known” is reset (*see page 77*).

Error Avoidance Features

The EEVDSettingsU/EEVDSettingsB Application Function Block provides the following error avoidance features to help you avoid the potentials of certain sources of machine detected error:

Error Avoidance Feature	Description
Input range validation	Input and parameter ranges are validated to help prevent out of range data.
Alarm/alert notification	An alarm is generated if: <ul style="list-style-type: none"> ● an invalid value is entered or has not been successfully written, ● the Application Function Block is not in the correct task.

EEVDSettingsU/B Application Function Blocks Details During Commissioning

Adding a EVEVD••000500 Device in the Project

Once a EVEVD••000500 device is added from the **Catalog** window to the **CAN Exp bus** node in the **Configuration** tab of FREE Studio Plus, a dialog window is displayed, where several choices are possible.

<input checked="" type="checkbox"/> Create new "Settings" function block instance for 1st valve	<input checked="" type="checkbox"/> Create new "Settings" function block instance for 2nd valve
<input checked="" type="radio"/> Create new EEPROM parameters	<input checked="" type="radio"/> Create new EEPROM parameters
<input type="radio"/> Link existing EEPROM parameters	<input type="radio"/> Link existing EEPROM parameters
<input type="radio"/> Do nothing	<input type="radio"/> Do nothing
<input type="radio"/> Do nothing	<input type="radio"/> Link first valve EEPROM parameters

EEVDSettingsU/B Application Function Blocks are in charge of sending the settings to the valve drivers.

Once added the EVEVD••000500 in the **Configuration** tab, you have to add the EEVDSettingsU/B instance inside a program in the **Programming** tab and to connect its I/O. This program must be added in the **Background** or **Init** task of the controller.

NOTE: If a list of parameters defined inside the `EEVsPar.csv` file is selected through `E2_uidE00` parameter and the EEVDSettingsU/B is not running in the **Background** or **Init** task, an alarm is generated and parameters are not set.

Create New "Settings" Function Block Instance

If checked, one of the following choices must be selected for EEPROM Parameters:

Choice	Description
Create New EEPROM Parameters	<ul style="list-style-type: none"> Creates the <code>EEVD Settings EEVx</code> menu in the Configuration Tab, which includes the EEPROM parameters. Creates the <code>EEV EEPROM</code> parameters that are added in the <code>EEPROM_Parameters</code> list in the Programming tab. Initializes the pointers to EEPROM parameters in the <code>EEVDSettingsU/BEEVx</code> Global Variable. Creates the <code>EEVD Settings EEVx PLC Application</code> menu in the Commissioning Tab, which includes the EEPROM parameters.
Link Existing EEPROM Parameters	In the case of an AFB is already instantiated and valid, it is possible to link the parameters of the new valve to an existing valve previously created. This is useful if you have several identical valves in your project. By this, you have to manage only one parameter list
Link first valve EEPROM parameters ^(*)	This link the parameters of the second valve to the new parameters created for the first valve. By this, you have to manage only one parameter list.
Do Nothing	In that case, the pointers to EEPROM parameters are initialized to 0. You have to link the pointers to the EEPROM parameters created by your own if you intend to use EEPROM parameters to configure the valve driver.
*: Only for the second valve of a EVEVD2B000500 valve driver.	

Once the valve configuration is validated, EEVDSettingsU/B Application Function Block is automatically instantiated as global variable in the `EEV_group` in the **Programming** tab.

The name of the instance and of parameters have the suffix indicating the number of the instance, for example `EEVDSettingsUEEV1`, `E2_dE01EEV1`, ..., for **EEV 1** related items.

It initializes the values for:

- The `usiNodeID` internal variable with the correspondent EEVD CAN address (default 20).
- The `usiEEVID` with the correspondent EEV index (default 1),
- The pointers to EEPROM parameters.
- The pointers to the following field variables (if **Create And link New "Field" Variables** is checked):
 - `iEEVDCommandsEEVx`
 - `iEEVDInfoEEVx`
 - `iEEVDOpeningCurrEEVx`
 - `iEEVDStatusEEVx`

NOTE: x corresponds to the valve number.

The link between AFB and global variables or EEPROM parameters is implemented through pointers which can be found in the field **Init value** of the AFB instance (see page 114).

Nothing Checked

Once the valve configuration is validated, the valve driver is added in the hardware structure in **Configuration** Tab. Nothing else is added to the project. It has the same effect as clicking **Cancel** button. In that case, you have to import from the library the instance for the valve driver.

You have also to initialize:

- The `usiNodeID` internal variable with the correspondent EEVD CAN address (default 20).
- The `usiEEVID` with the correspondent valve driver index (default 1),
- The pointers to field variables.

NOTE: This pointers by default point to generic variables (`iEEVDCommands`, `iEEVDOpeningCurr`, `iEEVDStatus`), you have to create your own and point to them to avoid compiling errors.

- The pointers to EEPROM parameters.

NOTE: These pointers by default point to NULL. You have to create your own variables or EEPROM parameters and point to them to use custom parameters, otherwise define only `dE00` and initialize the pointer.

You have to create a list of custom EEPROM parameters or to use only preconfigured values (`E2_uidE00` must be > 0).

The file `ParEEV.csv` must also be added in the `filesystem` if it is intended to be used.

In case of the valve driver is a EVEVD2B000500, you have to instantiate a second Application Function Block for the second valve driver and its related parameters.

ParEEV.csv File

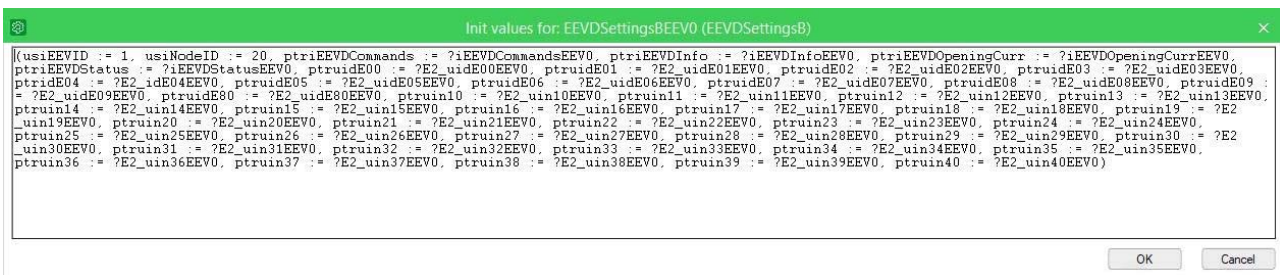
At the first instantiation of the first `EEVDSettingsU/B` Application Function Block, FREE Studio Plus automatically includes a `ParEEV.csv` file in the application folder. This file contains predefined sets of parameters for specific type of valves. It is downloaded to the controller together with the application (**Download all** command).

Init value

To modify the pointers initial value, select the `EEV_Group` node in **Project Window**:

	Name	Type	Address	Array	Init value
1	EEVDSettingsBEEV0	EEVDSettingsB	Auto	No	(usiEEVID := 1, usiNodeID := 2...
2	EEVDCntrlEEV0	EEVDCntrl	Auto	No	(usiNodeID := 20, ptriEEVD...

Select a function block and click the button in **Init value** cell. The pointers initial values are displayed:



You can modify the values. Click **OK** to valid.

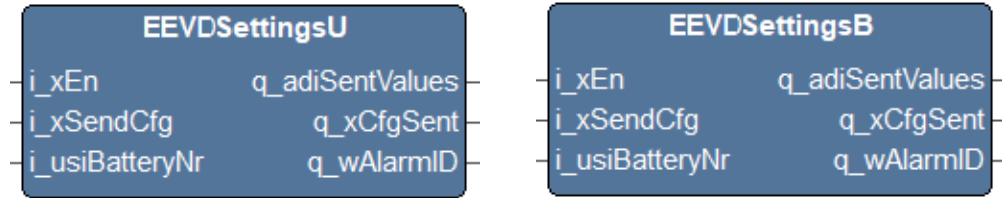
Input Pin Description

Pin Diagram

The following picture presents the pin diagrams of:

EEVDSettingsU

EEVDSettingsB



Input Pin Description

Input	Data type	Range	Scaling/Unit	Description
i_xEn	BOOL	TRUE / FALSE	–	Enables the function block
i_xSendCfg	BOOL	TRUE / FALSE	–	Send configuration Refer to i_xSendCfg description.
i_usiBatteryNr	USINT	0...2	–	Number of battery backup modules connected

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that there is only one master controller configured on the network or remote link.
- Verify that the devices have unique addresses.
- Confirm that the address of the device is unique before placing the system into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

i_xEn

If the input i_xEn is FALSE:

- The Application Function Block is disabled,
 - parameters loading and sending are disabled
 - alarm monitoring is disabled
- alarms that are not active at the moment are reset.

i_xSendCfg

The configuration parameters are sent to the EEVD on each transition from FALSE to TRUE of i_xSendCfg.

When i_xSendCfg=FALSE and E2_uidE00 becomes 0, the EEPROM configuration parameters are overwritten by the last selected E2_uidE00 configuration parameters.

NOTE: If the parameters to be sent are already at the correct value in the EVEVD••000500, the sending operation is not realized.

E2_uidE00 values:

The value of E2_uidE00 indicates which list of parameters to be sent to the EVEVD••000500 to control the EEV:

- 0: list of customized configuration parameters defined in EEPROM (Refer to Internal Variables description (*see page 119*)).
- 1...100: range for preconfigured configuration parameters internally defined (Refer to Valve configuration Parameters (*see page 80*)).
- 101...255: range for customized configuration parameters defined in `ParEEV.csv` file to be found in internal file system memory.

Values outside the range 0...255 determine no action and can be used to force a resending with `i_xSendCfg` maintained at TRUE, for example if some EEPROM parameters are modified.

`i_usiBatteryNr`

Indicates the number of battery backup modules connected to the driver. This input overrides all settings of `n36` parameter. If this input is out of range, the setting related to `E2_uidE00` is considered.

Default value of input is 0. When EVEVD1B000500/EVEVD2B000500 is used, only the setting of `n36` of EEV1 is considered (`n36` of EEV2 is ignored).

Output Pin Description

Output Pin Description for EEVDSettingsU/EEVDSettingsB

Output	Data type	Range	Scaling/Unit	Description
q_adiSentValues	DINT	ARRAY OF 42 ELEMENTS	–	Array of last sent parameters
q_xCfgSent	BOOL	TRUE / FALSE	–	Configuration successfully sent
q_wAlarmID	WORD	0 _h ...FFFF _h	–	Alarm flags

q_adiSentValues

It represents the last list of parameters sent to EVEVD••000500. If an error is detected during the sending of a single parameter, the correspondent parameter is set to 2147483647 (7FFFFFFF_h).

q_xCfgSent

Becomes TRUE when the list of parameters is successfully sent to EVEVD••000500.

Becomes FALSE when xSendCfg = FALSE or just after E2_uidE00 is changed if xSendCfg = TRUE.

This output can be used for monitoring or to reset the xSendCfg command with proper programming.

q_wAlarmID

Refer to Troubleshooting (*see page 118*).

Troubleshooting

Alarm ID Description

The `q_wAlarmID` output represents a value from 0 to 65535, whereby each bit represents an alarm. The bits and their description are described in the following table:

Alarm bit	Alarm Cause	Effect
0	Communication interruption	Parameters list not sent
1	Parameters not set	Some parameters are not correctly set
2	File not found	Parameters list not loaded nor sent
3	Invalid file format: header not found	
4	Invalid file format: invalid values	
5	Invalid file format: EEV not found	
6	Invalid task	Parameters list not sent
7	Pointers to EEPROM not initialized (only if <code>E2_uidE00=0</code>)	Parameters list not sent (only if <code>E2_uidE00=0</code>), EEPROM parameter not updated
8...10	Reserved	–
11	Writing in EEPROM not possible (only if <code>E2_uidE00=0</code>)	EEPROM parameter not updated
12	Parameters externally modified	Event to be externally managed
13...15	Reserved	–

Alarm ID Probable Cause

The probable cause linked to each alarm bit is presented in the following table:

Alarm bit	Alarm Details
0	Indicates a communication interruption with EVEVD••000500.
1	Indicates that some parameters are not correctly sent or set to EVEVD••000500. This can happen for communication issues or for a not valid value sent. It is possible to identify the incorrect parameters looking at <code>q_adiSentValues</code> array. This bit is raised if a send command is given.
2	Indicates that file <code>ParEEV.csv</code> is not found in <code>nor0</code> memory or cannot be opened. This applies if <code>E2_uidE00</code> is in the range 101...200 and a send command is given and/or the value of <code>E2_uidE00</code> is changed.
3	Indicates that the header in file <code>ParEEV.csv</code> has not the right format. To be valid, the file must have a row with an entry 'Selector' and an entry 'Description'. Between columns indicated with these two tags are assumed to be defined columns containing different parameters. This applies if <code>E2_uidE00</code> is in the range 101...200 and a send command is given and/or the value of <code>E2_uidE00</code> is changed.
4	Indicates that in file <code>ParEEV.csv</code> some values are not valid. For example: some index of parameter is beyond the maximum number of parameters or the length of index rows is different that of value rows. This applies if <code>E2_uidE00</code> is in the range 101...200 and a send command is given and/or the value of <code>E2_uidE00</code> is changed.
5	In file <code>ParEEV.csv</code> , the row containing the list of parameters is selected looking for a value equal to <code>E2_uidE00</code> in the column corresponding to the tag 'Selector'. This bit indicates that the value was not found. This applies if <code>E2_uidE00</code> is in the range 101...200 and a send command is given.
6	Indicates that the Application Function Block has been added to the timed task.
7	Indicates that some pointer to EEPROM parameters has been initialized to 0. This applies if <code>E2_uidE00</code> is 0 and a send command is given.
8...10	Reserved
11	Indicates that writing in EEPROM was not correctly executed. This applies if <code>E2_uidE00</code> is in the range 1...200 and a send command is given and/or the value of <code>E2_uidE00</code> is changed.
12	Indicates that a tool (FREE Studio Plus Installer connected with a DMI or other supervisor) other than the Application Function Block has modified parameter values of EVEVD••000500.
13...15	Reserved

Implementation

Internal Variables Description

These variables are automatically initialized to correct values if automatic instantiation is used. Otherwise you must provide your own values.

It is possible to define a parameter as a variable in RAM, changing the pointer initialization to address those variables.

NOTE: ptruidE00...ptruin40 are pointers to EEPROM parameters, used with E2_dE00 = 0.

Variable Name	Data type	Description
usiNodeID	USINT	CAN expansion bus address of EVEVD••000500
usiEEVID	USINT	EEV to be addressed inside EVEVD••000500
ptruiEEVDOpeningCurr	@USINT	Pointer to field variable EEVDOpeningCurr
ptruiEEVDStatus	@USINT	Pointer to field variable EEVDStatus
ptruiEEVDInfo	@USINT	Pointer to field variable EEVDInfo
ptruiEEVDCommands	@USINT	Pointer to field variable EEVDCommands
ptruidE00 ... ptruidE03	@UINT	Pointer to parameter E2_uidE00...E2_uidE03
ptridE04	@INT	Pointer to parameter E2_idE04
ptruidE05 ... ptruidE09	@UINT	Pointer to parameter E2_uidE05...E2_uidE09
ptruidE80	@UINT	Pointer to parameter E2_uidE80
ptruin10 ... ptruin40	@UINT	Pointer to parameter E2_uin10...E2_uin40

Parameters Description

Following the list of parameters automatically generated during slave device creation in **Configuration** tab.

Parameter	Data type	Range	Scaling/Unit	Description
E2_uidE00...E2_uidE03				Refer to Customizable Valve Configuration Parameters (see page 80)
E2_idE04				
E2_uidE05...E2_uidE09				
E2_uidE80				
E2_uin10...E2_uin40				

Chapter 16

EVEVD**000500 Application Control: `EEVDCntrl`

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Overview	122
<code>EEVDCntrl</code> Application Function Block Details During Commissioning	123
Input Pin Description	125
Output Pin Description	128
Troubleshooting	129
Implementation	130
Advanced Use	132

Functional Overview

Functional Overview

The `EEVDCntrl` Application Function Block is specific for EVEVD••000500 Electronic Expansion Valve Drivers (EEVD) for stepper Electronic Expansion Valves (EEVs). EVEVD••000500 acts as an expansion for FREE Advance controllers, exchanging data through CAN expansion bus.

The `EEVDCntrl` Application Function Block is instantiated as many times as there are valves. `EEVDCntrl` is in charge of controlling the EVEVD••000500 EEV depending on some application variables and sensors.

The algorithms implemented inside the Application Function Block decide the opening degree of the EEV considering some measures like refrigerant superheat, refrigerant pressure, or refrigerated fluid temperature.

Why Use the `EEVDCntrl` Application Function Blocks?

The `EEVDCntrl` Application Function Blocks are used to manage the following EVEVD••000500 functionality:

- Controlling the EVEVD••000500 associated to each valve depending on some application variables and sensors.
- Monitoring the parameters integrity.
- Monitoring the proper running of the sensor.

Features of the `EEVDCntrl` Application Function Blocks

The `EEVDCntrl` Application Function Block facilitates the following features:

- Create a set of EEPROM parameters automatically.
- Add the Field variables automatically.

Error Avoidance Features

The `EEVDCntrl` Application Function Block provides the following error avoidance features to help you avoid the potentials of certain sources of machine detected error:

Error Avoidance Feature	Description
Input range validation	Input and parameter ranges are validated to help prevent out of range data.
Alarm/alert notification	An alarm is generated if: <ul style="list-style-type: none"> ● an invalid value is entered or has not been successfully written. An alert is generated if: <ul style="list-style-type: none"> ● an invalid value is received from the probe.

EEVDCntrl Application Function Block Details During Commissioning

Adding a EVEVD••000500 Device in the Project

Once a EVEVD••000500 device is added from the **Catalog** window to the **CAN Exp bus** in the **Configuration** tab of FREE Studio Plus, a dialog window is displayed, where several choices are possible.

<input checked="" type="checkbox"/> Create new "Control" function block instance for 1st valve	<input checked="" type="checkbox"/> Create new "Control" function block instance for 2nd valve
<input checked="" type="checkbox"/> Use 'FB_MyEEVDCntrl' local modifiable function block	<input checked="" type="checkbox"/> Use 'FB_MyEEVDCntrl' local modifiable function block
<input checked="" type="radio"/> Create new EEPROM parameters	<input checked="" type="radio"/> Create new EEPROM parameters
<input type="radio"/> Link existing EEPROM parameters	<input type="radio"/> Link existing EEPROM parameters
<input type="radio"/> Do nothing	<input type="radio"/> Do nothing
<input type="radio"/> Link first valve EEPROM parameters	<input type="radio"/> Link first valve EEPROM parameters

EEVDCntrl Application Function Blocks is in charge of computing the opening degree of the EEV.

EEVDCntrl is an open Application Function Block, in the sense that you can go inside and see the structure in order to modify some parts and create your own algorithms, maintaining general structure and interface towards EVEVD••000500 EEVD. At the same time, you can take the function block inside and use them for example with another EEVD.

To do this, select **Use "FB_MyEEVDCntrl" local modifiable function block**. This creates a function block named `FB_MyEEVDCntrl` with the same structure of `EEVDCntrl` but editable. The function block is created only once (but instantiated many times) and is not be canceled when removing the EVEVD••000500 objects from the **Configuration** tab.

The EEPROM Parameters are used to set the configuration of the algorithms inside the Application Function Block

Once added the EVEVD••000500 in the **Configuration** tab, you have to add the `EEVDCntrl` or `FB_MyEEVDCntrl` instance inside a program in the **Programming** tab and to connect its I/O. This program must be added in the **Timed** task of the controller.

NOTE: If the `EEVDCntrl` is not running in the **Timed** task, an alarm is generated and the control is not performed.

Create New "Control" Function Block Instance

If checked, one of the following choices must be selected for EEPROM Parameters:

Choice	Description
Create New EEPROM Parameters	<ul style="list-style-type: none"> Creates the <code>EEVDCntrl Parameters EEVx</code> menu in the Configuration Tab, which includes the EEPROM parameters. Creates the EEV EEPROM parameters that are added in the <code>EEPROM_Parameters</code> list in the Programming tab. Initializes the pointers to EEPROM parameters in the <code>EEVDCntrlEEVx</code> Global Variable. Creates the <code>EEVDCntrl Paramters EEVx PLC Application</code> menu in the Commissioning Tab, which includes the EEPROM parameters.
Link Existing EEPROM Parameters	In the case of an AFB is already instantiated and valid, it is possible to link the parameters of the new system to an existing system previously created. This is useful if you have several identical systems in your project. By this, you have to manage only one parameter list.
Do Nothing	In that case, the pointers to EEPROM parameters are initialized to 0. You have to link the pointers to the EEPROM parameters created by your own if you intend to use EEPROM parameters to configure the valve driver.
Link first valve EEPROM parameters ^(*)	This link the parameters of the second valve to the new parameters created for the first valve By this, you have to manage only one parameter list.
*: Only for the second valve of a EVEVD2B000500 valve driver.	

Once the valve configuration is validated, `EEVDCntrl` or `FB_MyEEVDCntrl` Application Function Block is automatically instantiated as global variable in the `EEV_group` in the **Programming** tab.

The name of the instance and of parameters have the suffix indicating the number of the instance, for example EEVDCntrlEEV1, E2_uiPbAlarmValueEEV1, ..., for **EEV 1** related items.

It initializes the values for:

- The `usiNodeID` internal variable with the correspondent EEVD CAN address (default 20).
- The pointers to EEPROM parameters.
- The pointers to the following field variables (if **Create And link New "Field" Variables** is checked):
 - `iEEVDCCommandsEEVx`
 - `iEEVDOpeningCurrEEVx`
 - `iEEVDStatusEEVx`
 - `iEEVDAlarmsEEVx`
 - `iEEVDOpeningReqEEVx`

NOTE: x corresponds to the valve number.

The link between AFB and global variables or EEPROM parameters is implemented through pointers which can be found in the field **Init value** of the AFB instance (see page 124).

Nothing Checked

Once the valve configuration is validated, the valve driver is added in the hardware structure in **Configuration** tab. Nothing else is added to the project. It has the same effect as clicking **Cancel** button.

In that case, you have to import from the library the instance for the valve driver.

You have also to initialize:

- The `usiNodeID` internal variable with the correspondent EEVD CAN address (default 20).
- The pointers to field variables.
 - `iEEVDCCommands`
 - `iEEVDOpeningCurr`
 - `iEEVDStatus`
 - `iEEVDAlarms`
 - `iEEVDOpeningReq`

NOTE: This pointers by default point to generic variables (`iEEVDCCommands`, `iEEVDOpeningCurr`, `iEEVDStatus`, `iEEVDAlarms`), `iEEVDOpeningReq`), you have to create your own and point to them to avoid compiling errors.

- The pointers to EEPROM parameters.

NOTE: These pointers by default point to generic variables (`iSuperHeatSetp`, `uiPbAlarms`,...) You have to create your own variables or EEPROM parameters and point to them to avoid compiling errors.

In case of the valve driver is a EVEVD2B000500, you have to instantiate a second Application Function Block for the second valve driver and its related parameters.

Init value

To modify the pointers initial value, select the **EEV_Group** node in **Project Window**:

	Name	Type	Address	Array	Init value
1	EEVDSettingsBEEV0	EEVDSettingsB	Auto	No	(usiEEVID := 1, usiNodeID := 2...
2	EEVDCntrlEEV0	EEVDCntrl	Auto	No	(usiNodeID := 20, ptriEEVD...

You can modify the values. Click **OK** to valid.

Select a function block and click the button in **Init value** cell. The pointers initial values are displayed:

```

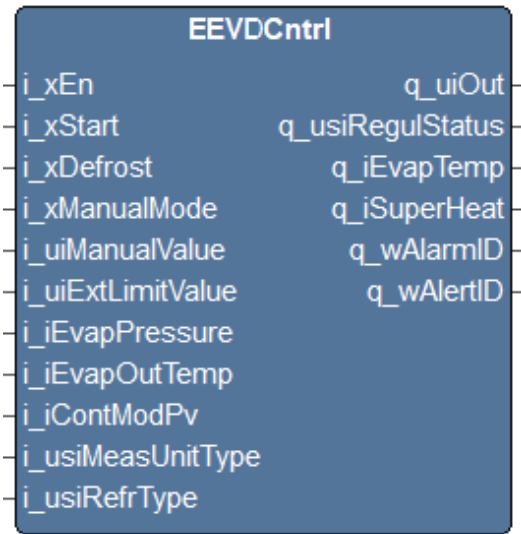
Init values for: EEVDCntrlEEV0 (EEVDCntrl)
(usiNodeID := 20, ptriEEVDCCommands := ?iEEVDCCommandsEEV0, ptriEEVDOpeningCurr := ?iEEVDOpeningCurrEEV0, ptriEEVDStatus := ?iEEVDStatusEEV0, ptriEEVDAlarms := ?iEEVDAlarmsEEV0, ptriEEVDOpeningReq := ?iEEVDOpeningReqEEV0, ptriSuperHeatSetp := ?E2_iSuperHeatSetpEEV0, ptruiPbAlaraValue := ?E2_uiPbAlaraValueEEV0, ptruiExtAlaraValue := ?E2_uiExtAlaraValueEEV0, ptruiComAlaraValue := ?E2_uiComAlaraValueEEV0, ptruiUnitType := ?E2_uiUnitTypeEEV0, ptriEnMOP := ?E2_xEnMOPEEV0, ptriMOPSetp := ?E2_iMOPSetpEEV0, ptruiContModType := ?E2_usiContModTypeEEV0, ptriContModSetp := ?E2_iContModSetpEEV0, ptriMaxValue := ?E2_uiMaxValueEEV0, ptruiDeadZoneValueMin := ?E2_uiDeadZoneValueMinEEV0, ptruiDeadZoneValueMax := ?E2_uiDeadZoneValueMaxEEV0, ptruiMaxVarOut := ?E2_uiMaxVarOutEEV0, ptruiValueOnStart := ?E2_uiValueOnStartEEV0, ptruiTimeOnStart := ?E2_uiTimeOnStartEEV0, ptruiValueOnStop := ?E2_uiValueOnStopEEV0, ptruiTimeOnStop := ?E2_uiTimeOnStopEEV0, ptruiValueOnDefrost := ?E2_uiValueOnDefrostEEV0, ptruiHighLoadDelay := ?E2_uiHighLoadDelayEEV0, ptruiMOPAlaraDelay := ?E2_uiMOPAlaraDelayEEV0, ptruiOpenAlaraDelay := ?E2_uiOpenAlaraDelayEEV0, ptruiSuperHeatPb := ?E2_uiSuperHeatPbEEV0, ptruiSuperHeatTi := ?E2_uiSuperHeatTiEEV0, ptruiSuperHeatTd := ?E2_uiSuperHeatTdEEV0, ptriEnSuperHeatEvo := ?E2_xEnSuperHeatEvoEEV0, ptruiSuperHeatDz := ?E2_uiSuperHeatDzEEV0, ptruiMOPPb := ?E2_uiMOPPbEEV0, ptruiMOPTi := ?E2_uiMOPTiEEV0, ptruiMOPTd := ?E2_uiMOPTdEEV0, ptruiContModPb := ?E2_uiContModPbEEV0, ptruiContModTi := ?E2_uiContModTiEEV0, ptruiContModTd := ?E2_uiContModTdEEV0, ptriEnDinSetpCalc := ?E2_xEnDinSetpCalcEEV0, ptruiDinSetpMaxOffset := ?E2_uiDinSetpMaxOffsetEEV0, ptruiSetpIncrStep := ?E2_uiSetpIncrStepEEV0, ptruiSetpIncrTime := ?E2_uiSetpIncrTimeEEV0, ptriLastValue := ?E2_iLastValueEEV0)
    
```

You can modify the values. Click **OK** to valid.

Input Pin Description

Pin Diagram

The following picture presents the pin diagram of EEVDCntrl:



Input Pin Description

Input	Data type	Range	Scaling/Unit	Description
i_xEn	BOOL	TRUE / FALSE	-	Enables the function block
i_xStart	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> TRUE: enables the regulation FALSE: closing procedure
i_xDefrost*	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> TRUE: abrupt close of the valve to permit the defrost FALSE (default value): restart from defrost
i_xManualMode*	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> TRUE: commands directly the output with uiManualValue FALSE (default value): regulation active
i_uiManualValue*	UINT	0...1000	0.1 %	Manual value to set the output Default value: 0
i_uiExtLimitValue*	UINT	0...1000	0.1 %	External value limitation of EEV opening Default value: 1000
i_iEvapPressure	INT	-32760...32760	0.01 bar/psi	Evaporator pressure Values lower than -32760 and greater than 32760 are invalid
i_iEvapOutTemp	INT	-32760...32760	0.1 °C/°F	Evaporator output temperature Values lower than -32760 and greater than 32760 are invalid
i_iContModPv*	INT	-32760...32760	0.1 °C/°F/bar/psi	Continuous modulation process value Values lower than -32760 and greater than 32760 are invalid Default value: -32768
i_usiMeasUnitType	USINT	0...1	-	Measurement unit type: <ul style="list-style-type: none"> 0 = International System (SI) 1 = Imperial System
i_usiRefrType	USINT	0...255	0.1 °C/°F	Refrigerant type: 255 = custom refrigerant
*: If not connected, the default value is used.				

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify that there is only one master controller configured on the network or remote link.
- Verify that the devices have unique addresses.
- Confirm that the address of the device is unique before placing the system into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

i_xEn

If the input `i_xEn` is FALSE, the outputs are set to zero.

This command has the highest priority above `i_xManualMode`, `i_xStart`, and `i_xDefrost`.

i_xStart

If `i_xEn = TRUE` and `i_xStart` is set to TRUE, the function block enables the regulation with a start procedure (see description of the parameter `E2_uiValueOnStart` (see page 151)).

The PID integral is loaded and the output is set to `E2_uiValueOnStart` or to last saved value, just after the startup.

When it is reset to FALSE, the stop procedure is enabled (refer to the description of the parameter `E2_uiValueOnStop` (see page 151)).

`i_xStart` has less priority than `i_xEn` and `i_xManualMode` so if `xEn = FALSE` the start procedure cannot take place as well as if `xManualMode = TRUE` a closure routine cannot be performed.

i_xDefrost

It enables the Defrost procedure which consists on an abrupt closing procedure when the defrost is enabled and in a start from defrost procedure when `i_xDefrost` is reset (see description of the parameter `E2_uiValueOnDefrost` (see page 151)).

The Start from defrost is different than simple Start procedure: a different opening value is used. Function used in refrigeration.

i_xManualMode, i_uiManualValue

If the function block is enabled and `i_xManualMode` is set to TRUE, the function block commands an opening of the valve at `i_uiManualValue` (`q_uiOut = i_uiManualValue`).

When the pin is reset to FALSE, and `i_xEn` and `i_xStart` are enabled, the output goes to the regulated value with a bump-less pass.

If `xEn = TRUE`, `xStart = FALSE`, and `xManualMode` is reset to FALSE, the valve abruptly goes to zero.

If `xEn = FALSE`, the opening command is not performed cause the function block is disabled.

i_uiExtLimitValue

This input can be used to force the EEV to close at this value if other regulators drive a greater value and at the same time to keep other regulators ready to take control if they require a minor opening degree.

i_iEvapPressure

The value of evaporating pressure. It comes from an evaporator pressure sensor in the suction line of the refrigeration circuit.

If an invalid (outside range) value on the pressure sensor is detected, the system sets a flag managing the situation how application requires.

This value is internally used to calculate `q_iEvapTemp` and `q_iSuperHeat`, depending on `i_usiRefrType`.

i_iEvapOutTemp

The value of evaporator output temperature. It comes from temperature sensor placed in the pipe of the suction line of the refrigeration circuit between the evaporator and the compressor usually.

If an invalid value on the temperature probe is detected, a flag is set managing the situation how application requires.

This value is internally used to calculate $q_{iSuperHeat}$, depending on *i_usiRefrType*.

i_iContModPv

The value of continuous modulation process variable. This could be for example the temperature of the air in the cabinet to be regulated or the temperature of the water to be chilled, or the pressure in secondary line in CO2 cascade system.

i_iMeasUnitType

Indicates which measurement units must be considered for inputs and outputs representing pressures, temperatures, or temperature difference.

A value of 0 indicates the International System (SI) that means bar, °C, and K; a value of 1 indicates the Imperial System that means psi, °F, °R.

i_usiRefrType

Indicates the refrigerant to be considered for pressure/temperature conversion. The valid range of values is indicated in *Press2Temp* Application Function Block of Application library. A value of 255 permits to you to implement custom refrigerant.

Output Pin Description

Output	Data type	Range	Scaling/Unit	Description
q_uiOut	UINT	0...1000	0.1 %	Output value representing the opening level of the EEV
q_usiRegulStatus	USINT	0 ...9	-	Current regulation status
q_iEvapTemp	INT	-32760...32760	0.1 °C/°F	Current evaporation temperature calculated -32768 means that i_iEvapPressure is invalid
q_iSuperHeat	INT	-32760...32760	0.1 K/°R	Current super heat calculated -32768 means that i_iEvapPressure or i_iEvapOutTemp is invalid
q_wAlarmID	WORD	0 _h ...FFFF _h	-	Alarm flags
q_wAlertID	WORD	0 _h ...FFFF _h	-	Alert flags

q_uiOut

Output value representing the opening level (%) of the EEV.

If the EEVD is not activated and synchronized and if the position is not known (Status bit 4 = 0), the driven opening level is ignored from EEVD. This could happen if EEVDSettingsU/B is not properly running and enabled in your application.

q_usiRegulStatus

Indicates the current regulation status:

- 0= OFF
- 1=SH
- 2= MOP
- 3=CONTINUOUS MODULATION
- 4=EXTERNAL LIMITATION
- 5=START
- 6=STOP
- 7=DEFROST
- 8=MANUAL
- 9=ALARM

Refer to the algorithms description (*see page 110*).

q_iEvapTemp

Value of current evaporation temperature calculated; it is the saturation temperature in the suction line of the refrigeration circuit, calculated depending on evaporation pressure and refrigerant (see description of Press2Temp (*see EcoStruxure Machine Expert - HVAC, Application Function, HVAC Library Guide*)).

q_iSuperHeat

Value of current superheat calculated; it is equal to the difference between i_iEvapOutTemp and q_iEvapTemp.

q_wAlarmID

Refer to Troubleshooting (*see page 129*).

q_wAlertID

Refer to Troubleshooting (*see page 129*).

Troubleshooting

q_wAlarmID

The `q_wAlarmID` output represents a value from 0 to 65535, whereby each bit represents an alarm. Because alarms are directly coming from `FB_EEVSHCntrl` Application Function Block, it is possible to find the description in the related paragraph (*see page 153*).

q_wAlertID

The `q_wAlertID` output represents a value from 0 to 65535, whereby each bit represents an alert. Because alerts are directly coming from `FB_EEVSHCntrl` Application Function Block, it is possible to find the description in the related paragraph (*see page 154*).

Implementation

Internal Variables Description

These variables are automatically initialized to right values if automatic instantiation is used. Otherwise you must provide your own values or eliminate unused inputs. It is possible to define a parameter as a variable in RAM, changing the pointer initialization to address those variables.

Variable Name	Data type	Description	Range
usiNodeID	USINT	Node ID	1...255
ptriEEVDOpeningCurr	@INT	Pointer to field variable iEEVDOpeningCurr	-
ptriEEVDStatus	@INT	Pointer to field variable iEEVDStatus	-
ptriEEVDAlarms	@INT	Pointer to field variable iEEVDAlarms	-
ptriEEVDInfo	@INT	Pointer to field variable iEEVDInfo	-
ptriEEVDOpeningReq	@INT	Pointer to field variable iEEVDOpeningReq	-
ptriEEVDCommands	@INT	Pointer to field variable iEEVDCommands	-
ptriSuperHeatSetp	@INT	Pointer to field variable E2_iSuperHeatSetp	-
ptruiPbAlarmValue	@UINT	Pointer to field variable E2_uiPbAlarmValue	-
ptruiExtAlarmValue	@UINT	Pointer to field variable E2_uiExtAlarmValue	-
ptruiComAlarmValue	@UINT	Pointer to field variable E2_uiComAlarmValue	-
ptrusiUnitType	@USINT	Pointer to field variable E2_usiUnitType	-
ptrxEnMOP	@BOOL	Pointer to field variable E2_xEnMOP	-
ptriMOPSetp	@INT	Pointer to field variable E2_iMOPSetp	-
ptrusiContModType	@USINT	Pointer to field variable E2_usiContModType	-
ptriContModSetp	@INT	Pointer to field variable E2_iContModSetp	-
ptruiMaxValue	@UINT	Pointer to field variable E2_uiMaxValue	-
ptruiDeadZoneValueMin	@UINT	Pointer to field variable E2_uiDeadZoneValueMin	-
ptruiDeadZoneValueMax	@UINT	Pointer to field variable E2_uiDeadZoneValueMax	-
ptruiMaxVarOut	@UINT	Pointer to field variable E2_uiMaxVarOut	-
ptruiValueOnStart	@UINT	Pointer to field variable E2_uiValueOnStart	-
ptruiTimeOnStart	@UINT	Pointer to field variable E2_uiTimeOnStart	-
ptruiValueOnStop	@UINT	Pointer to field variable E2_uiValueOnStop	-
ptruiTimeOnStop	@UINT	Pointer to field variable E2_uiTimeOnStop	-
ptruiValueOnDefrost	@UINT	Pointer to field variable E2_uiValueOnDefrost	-
ptruiHighLoadDelay	@UINT	Pointer to field variable E2_uiHighLoadDelay	-
ptruiMOPAlarmDelay	@UINT	Pointer to field variable E2_uiMOPAlarmDelay	-
ptruiOpenAlarmDelay	@UINT	Pointer to field variable E2_uiOpenAlarmDelay	-
ptruiSuperHeatPb	@UINT	Pointer to field variable E2_uiSuperHeatPb	-
ptruiSuperHeatTi	@UINT	Pointer to field variable E2_uiSuperHeatTi	-
ptruiSuperHeatTd	@UINT	Pointer to field variable E2_uiSuperHeatTd	-
ptrxEnSuperHeatEvo	@BOOL	Pointer to field variable E2_xEnSuperHeatEvo	-
ptruiSuperHeatDz	@UINT	Pointer to field variable E2_uiSuperHeatDz	-
ptruiMOPPb	@UINT	Pointer to field variable E2_uiMOPPb	-
ptruiMOPTi	@UINT	Pointer to field variable E2_uiMOPTi	-
ptruiMOPTd	@UINT	Pointer to field variable E2_uiMOPTd	-
ptruiContModPb	@UINT	Pointer to field variable E2_uiContModPb	-
ptruiContModTi	@UINT	Pointer to field variable E2_uiContModTi	-
ptruiContModTd	@UINT	Pointer to field variable E2_uiContModTd	-
ptrxEnDinSetpCalc	@BOOL	Pointer to field variable E2_xEnDinSetpCalc	-
ptruiDinSetpMaxOffset	@UINT	Pointer to field variable E2_uiDinSetpMaxOffset	-

Variable Name	Data type	Description	Range
ptruiSetpIncrStep	@UINT	Pointer to field variable E2_uiSetpIncrStep	-
ptruiSetpIncrTime	@UINT	Pointer to field variable E2_uiSetpIncrTime	-
ptriLastValue	@INT	Pointer to field variable E2_iLastValue	-

Parameters and relative pointers are entirely managed in the FB_EEVSHCntrl Application Function Block.

This AFB could eventually be used standalone (for example with a third party driver). The AFB I/O is open so you can delete the CAN interface to indicate that you do not want to control the EEVD via CAN Expansion Bus.

usiNodeID

This represents the CAN expansion bus address of EVEVD••000500.

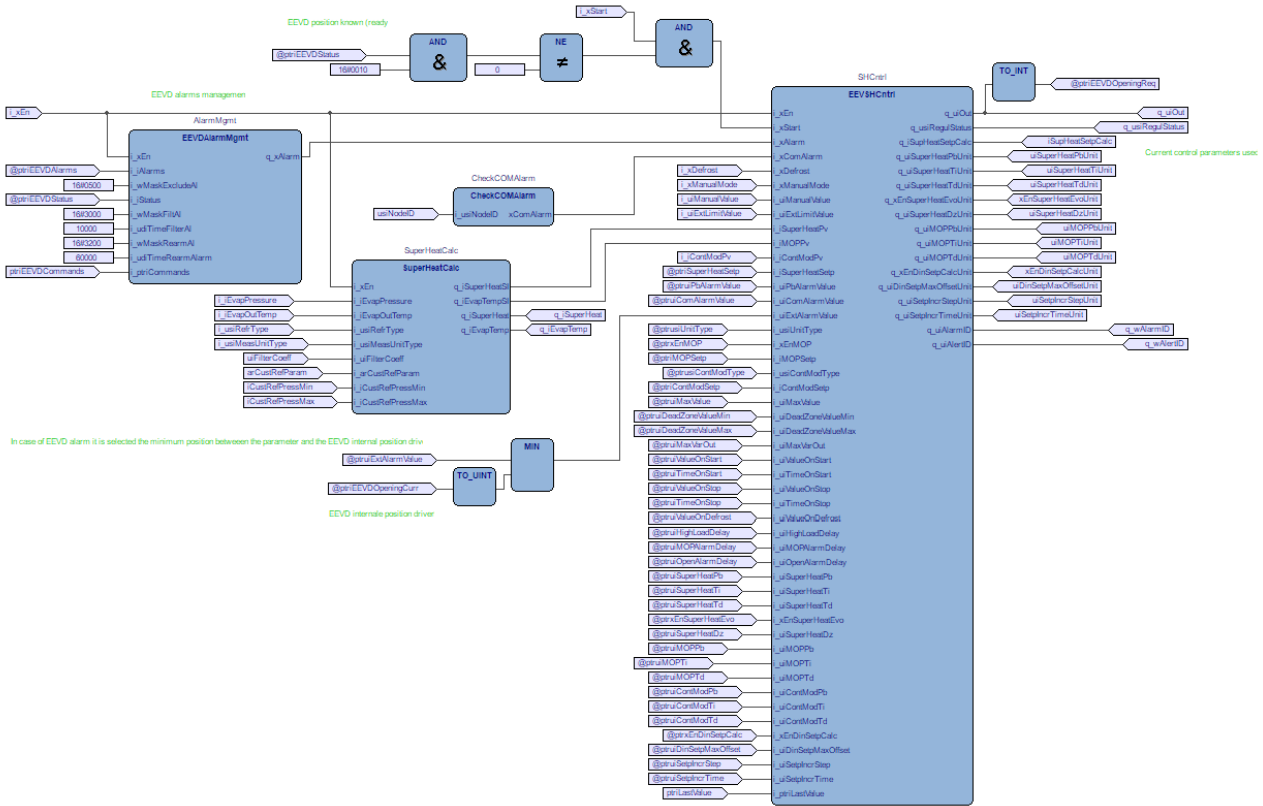
Pointers

The others variables are pointers to the interface global variables automatically created during procedure in **Configuration** tab.

Advanced Use

Internal Structure of EEVDCntrl

The following graphic indicates how EEVDCntrl is internally organized in Functions Blocks and how you can interface to implement your own control or customize some functions.



It is possible to see the variables defined as input having the prefix `i_`, those defined as outputs with prefix `q_`.

Also `ptruiEEVDAIarms`, `ptriEEVDOpeningCurr`, and `ptriEEVDOpeningReq` are pointers to field variables used to interface with EVEVD••000500 and `usiNodeID` indicates the CAN address of the EVEVD••000500 associated.

The other variables are for internal use and debug but you can take them outside as I/O modules or create your own. To implement your modification, you can use the **FB_myEEVDCntrl** function block (see page 125).

EEVSHCntrl Application Function Block and SuperHeatCalc are described in the next paragraphs.

For more information about:

- EEVSHCntrl: refer to EVEVD••000500: EEVSHCntrl chapter (see page 135).
- SuperHeatCalc: refer to EVEVD••000500 Evaporator Temperature and Superheat Calculation: SuperHeatCalc chapter (see page 157).

Chapter 17

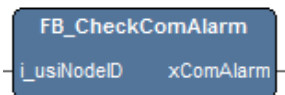
EVEVD**000500: CAN Expansion Bus

Monitoring CheckComAlarm

CAN Expansion Bus Monitoring CheckComAlarm

Functional Overview

The CAN Expansion Bus communication is continuously monitored through this function block. If a communication interruption is detected, a flag is set. This flag is managed by the following Application Function Block (here the `EEVSHCntrl`):



Input Pin Description

Input	Data type	Range	Scaling/Unit	Description
i_usiNodeID	USINT	1...255	-	Node ID CAN address

Output Pin Description

Input	Data type	Range	Scaling/Unit	Description
xComAlarm	BOOL	TRUE / FALSE	-	Communication alarm detected TRUE if a communication interruption is detected

Chapter 18

EVEVD••000500: Standard Application Control Algorithm `EEVSHCtrl`

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Standard Application Control Algorithm (<code>EEVSHCtrl</code>)	136
Evolved Superheat Algorithm (<code>EEVSHCtrl</code>)	139
Input Pin Description	141
Output Pin Description	146
Parameters Description	148
Troubleshooting	153
Implementation	155

Standard Application Control Algorithm (EEVSHCtrl)

Overview

This Application Function Block manages the EEV opening degree to control the superheat and keep the evaporator pressure under MOP (maximum operating pressure) in an evaporator or equivalent system.

It implements the same algorithm as implemented inside EVEVD••000500 driver adding the continuous modulation feature.

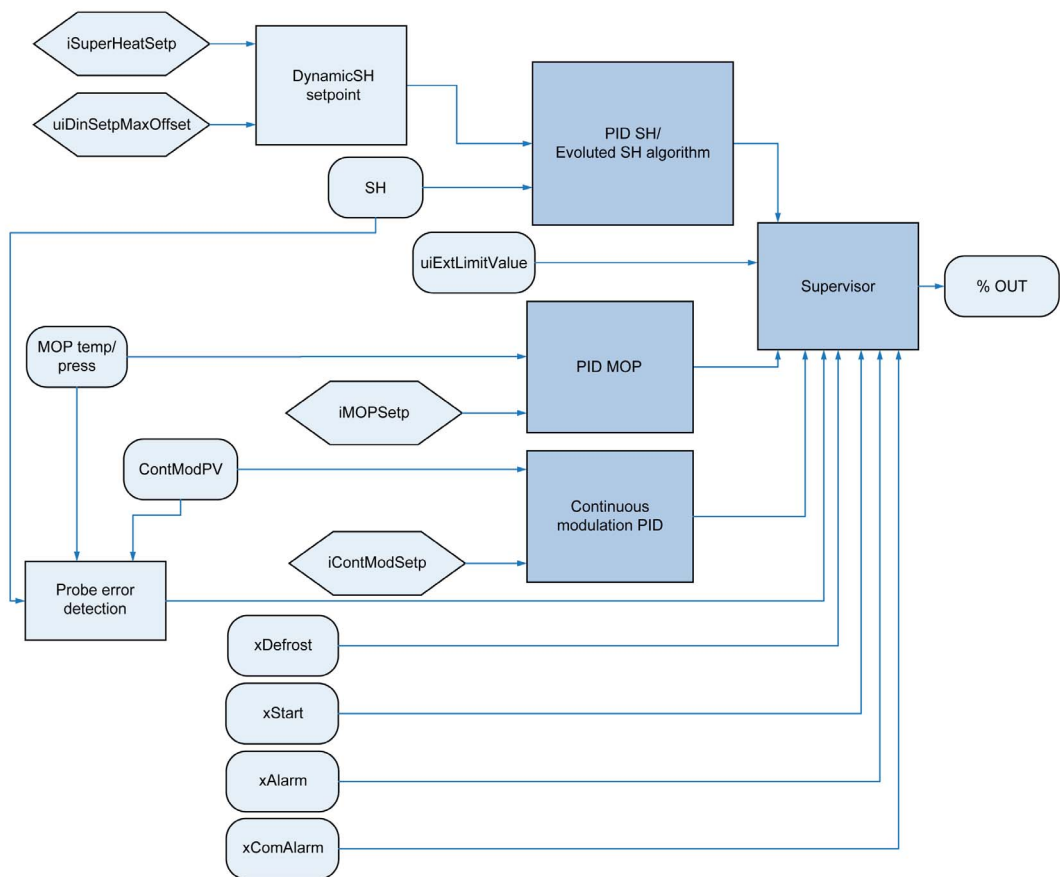
The inputs are some logic commands and the analogic quantities superheat and MOP, which normally corresponds to operating suction pressure or equivalent saturated vapor temperature. These quantities (superheat and saturated vapor temperature) must be calculated through SuperHeatCalc function block or equivalent.

It is possible also to manage a third analogic quantity, named Continuous Modulation Process Value, to be regulated in parallel. This could be for example the temperature of the secondary fluid to be refrigerated (air or water) or the pressure in a secondary circuit as for example CO2 in cascade systems.

The output is the opening degree for the EEV, status of regulation, and alarms.

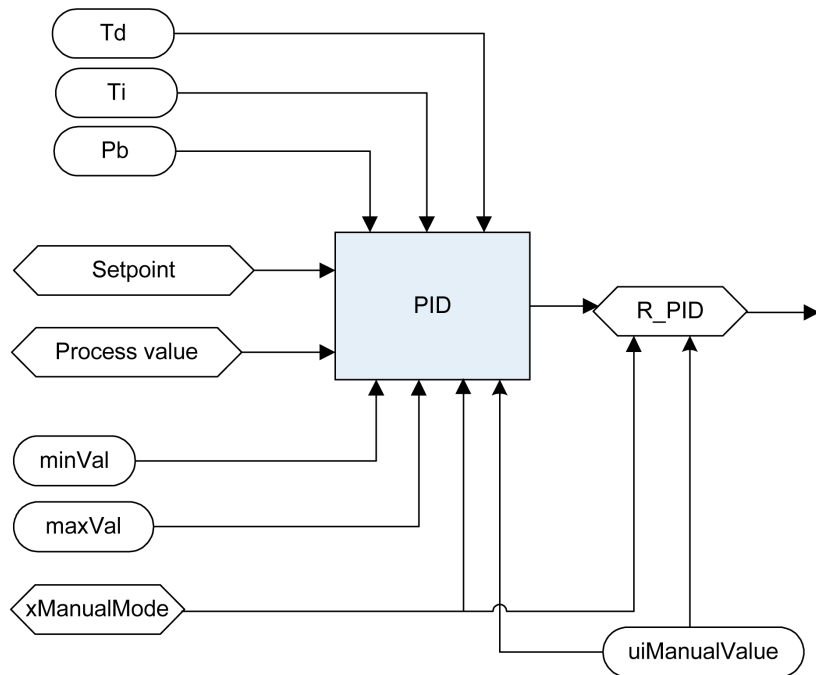
These outputs are sent to the EVEVD••000500 through CAN expansion bus or may be used with other drivers in order to pilot the EEV (pulse or stepper).

Algorithm Description



Standard PID

The PID regulators in the previous scheme are implemented in the following way:



Ti (uiSuperHeatTi, uiMOPTi) Integral Time

Td (uiSuperHeatTd, uiMOPTd) Derivative Time

Pb (uiSuperHeatPb, uiMOPPb) Proportional band

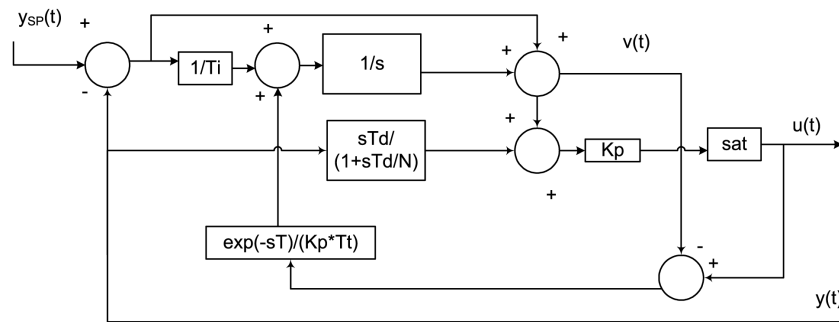
Setpoint Current regulation setpoint; for MOP it corresponds to iMOPSetp, in superheat regulation it is dynamic setpoint described in the next section

Process value Variable to be regulated (superheat, vapor saturated temperature, ...)

minVal, maxVal Maximum and minimum saturation of regulator output (see *sat* in the following diagram), they are calculated in the supervisor block every time a block is selected

R_PID Regulator output, the opening degree percentage of EEV which is passed to supervisor

xManualMode If TRUE, it is possible to drive R_PID directly at uiManualValue opening degree.



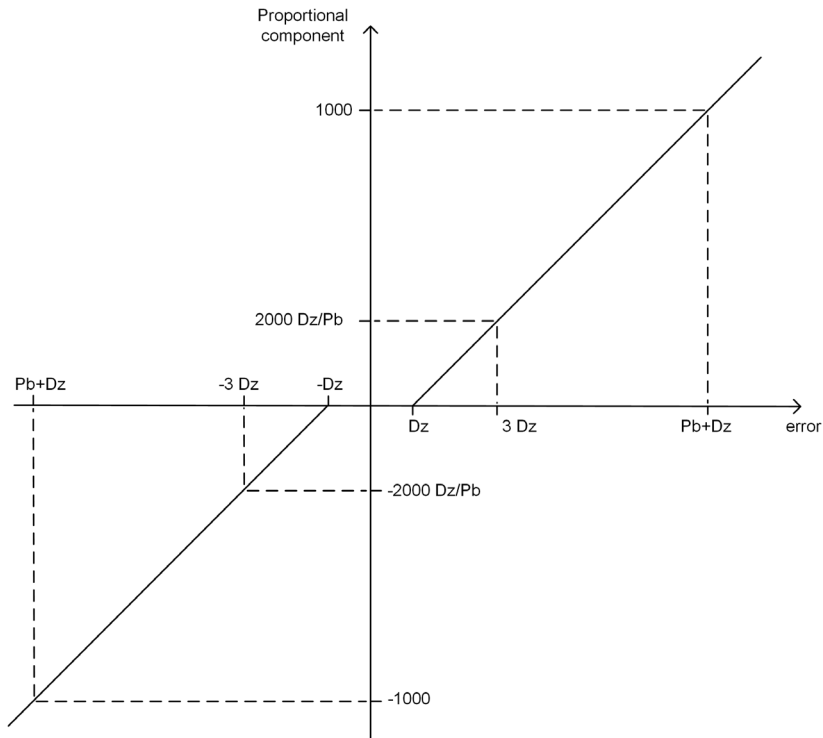
Regulation Algorithm Description

The proportional component is calculated as a function of the error and the dead zone of the regulation:

$ProportionalComponent = 0, \text{ if } ABS(error) \leq uiSuperHeatDz;$

$ProportionalComponent = (error - uiSuperHeatDz) * 1000 / uiSuperHeatPb, \text{ if } ABS(error) > uiSuperHeatDz;$

If $u_{iSuperHeatDz} > 0$, a non-linearity in the proportional part is introduced which can be represented as in following graphic:



The algorithm, in continuous time, is the following (standard form ISA):

$$u(t) = K_p \cdot \left((y_{sp1} - y(t)) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + \frac{1}{K_p \cdot T_t} \int_0^t (u(\tau) - v(\tau)) d\tau - T_d \frac{d(y(t))}{dt} \right)$$

PID Tuning

As an example only, here is a PID control tuning method to be performed with a constant load:

1. Set a sufficiently high proportional band value (Pb) and integral time (Ti).
2. Energize the device and verify the superheat progress:
 - If it approaches the setpoint very slowly, decrease the integral time value (Ti) and repeat the step.
 - If it goes far below the setpoint and stays there for a significantly long time, increase the integral time value (Ti) and repeat the step.
3. When the result of the previous step is sufficiently accurate, you can increase the system reactivity, reducing the proportional band (Pb).
4. Typically derivative part should not be used (Td=0) because of the great noise in thermodynamics systems.

Evolved Superheat Algorithm

If $x_{EnSuperHeatEvo} = \text{TRUE}$, the superheat control is performed with an advanced algorithm, instead of the standard PID.

For more details, refer to Evolved Superheat Algorithm (*see page 139*).

Evolved Superheat Algorithm (EEVSHCtrl)

Overview

If `xEnSuperHeatEvo = TRUE`, the superheat control is performed with an advanced algorithm, instead of the standard PID.

Calculation Time

The control is performed every `T` milliseconds. `T` depends on the entity of the error:

`T = 100` if `error > 2*uiSuperHeatDz` or `error < - uiSuperHeatDz`

`T = 200` if `error <= 2*uiSuperHeatDz` and `error >= - uiSuperHeatDz`

Regulation Algorithm Description

The output of the algorithm is the sum of a proportional (not linear) component and an integral (not linear) component.

The proportional component is calculated as a function of the error and the dead zone of the regulation:

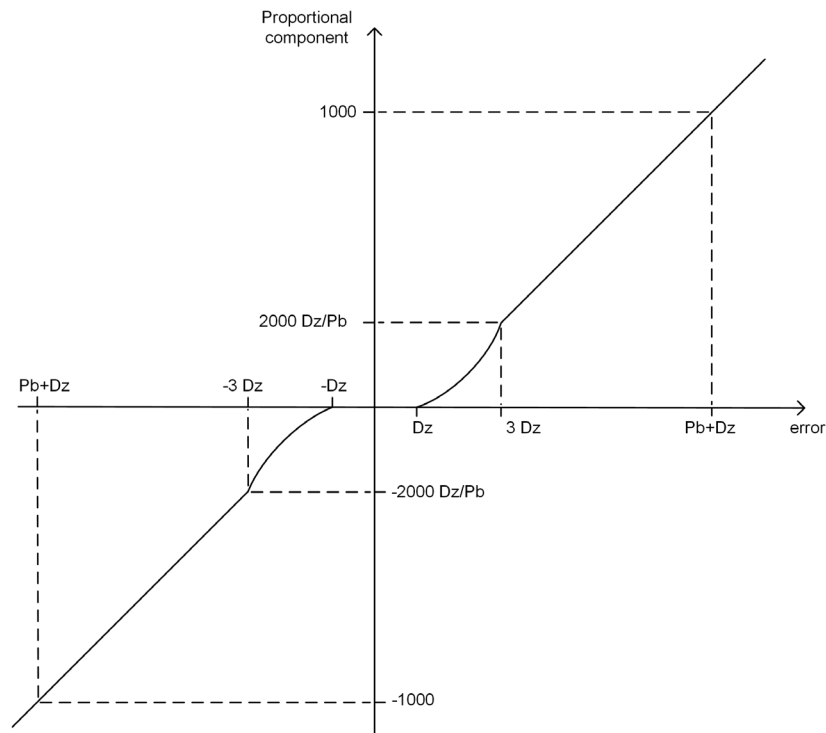
`ProportionalComponent = 0` if `ABS(error) <= uiSuperHeatDz`;

`ProportionalComponent = [(error - sign(error) * uiSuperHeatDz) * 2 * uiSuperHeatDz / (4 * uiSuperHeatDz - ABS(error - uiSuperHeatDz))] * (1000 / uiSuperHeatPb)` ,

if `uiSuperHeatDz < ABS(error) < 3*uiSuperHeatDz`;

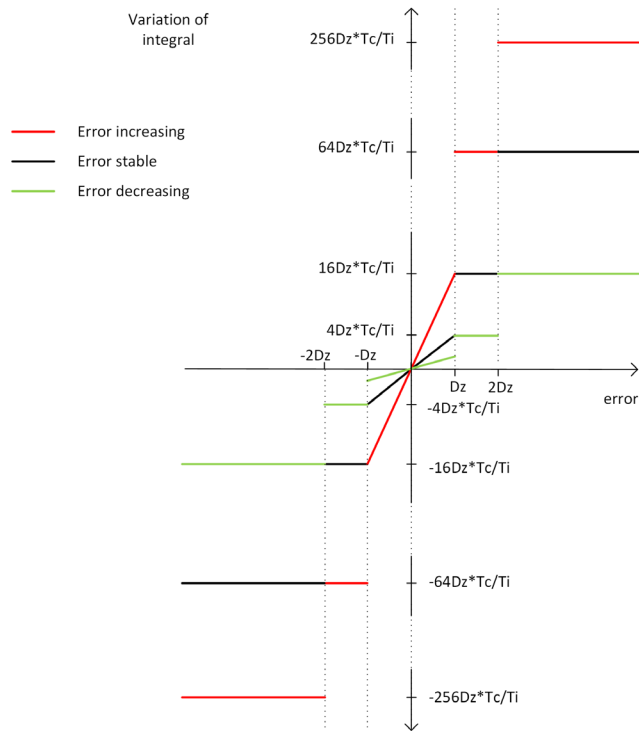
`ProportionalComponent = (error - sign(error) * uiSuperHeatDz) * 1000 / uiSuperHeatPb`, if `ABS(error) >= 3*uiSuperHeatDz`;

The proportional component is calculated as a function of the error and the dead zone of the regulation:



The integral component is based on the direction of the error, on its sign and on the parameters `uiSuperHeatDz`, `uiSuperHeatTi` ($T_c = 0.1$ s). The routine is calculated at every sample time (T_c).

The following graphic presents the variation of integral:



Tuning

Starting from a properly tuned PID leaving the same P_b and T_i parameters and setting a proper D_z , the Evoluted algorithm introduces some non-linearities, in particular:

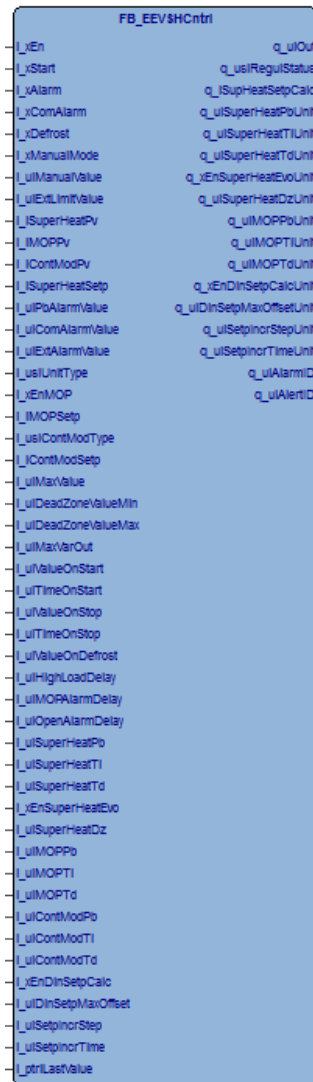
- A proportional part which is absent inside the dead zone and slightly increase until becoming linear between D_z and $3 \cdot D_z$
- An integral part which has the same behaviors as linear PID only inside dead zone with a decreasing (in absolute value) error, so in general more powerful in particular outside dead zone

In general, this non-linearities improves the behavior of the properly tuned linear PID maintaining the same parameters, but a T_i increase could be useful in some situations to reduce a too powerful integral action.

Input Pin Description

Pin Diagram

The following picture presents the pin diagram of EEVSHCNtrl:



Input Pin Description

Input	Data type	Range	Scaling/Unit	Description
i_xEn <i>(see page 144)</i>	BOOL	TRUE / FALSE	-	Enables the function block
i_xStart <i>(see page 144)</i>	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> ● TRUE: enables the regulation ● FALSE: closing procedure
i_xAlarm * <i>(see page 144)</i>	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> ● TRUE: alarm ongoing, go to fixed value ● FALSE (default value): normal regulation
i_xComAlarm <i>(see page 144)</i>	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> ● TRUE (default value): communication alarm ongoing goes to fixed value ● FALSE: normal regulation
*: If not connected, the default value is used.				

Input	Data type	Range	Scaling/Unit	Description
i_xDefrost * (<i>see page 144</i>)	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> TRUE: abrupt close of the valve to permit the defrost FALSE (default value): restart from defrost
i_xManualMode * (<i>see page 144</i>)	BOOL	TRUE / FALSE	-	TRUE: commands directly the output with uiManualValue FALSE (default value): regulation active
i_uiManualValue *	UINT	0...1000	0.1 %	Manual value to set the output Default value: 0
i_uiExtLimitValue * (<i>see page 144</i>)	UINT	0...1000	0.1 %	External value limitation of EEV opening Default value: 1000
i_iSuperHeatPv (<i>see page 144</i>)	INT	-32760...32760	0.1 K/°R	Super heat process value Values lower than -32760 and greater than 32760 are invalid
i_iMOPpv * (<i>see page 145</i>)	INT	-32760...32760	0.1 °C/°F/bar/psi	MOP process value Values lower than -32760 and greater than 32760 are invalid Default value: -32768
i_iContModPv * (<i>see page 145</i>)	INT	-32760...32760	0.1 °C/°F/bar/psi	Continuous modulation process value Values lower than -32760 and greater than 32760 are invalid Default value: -32768
l_iSuperHeatSetp	INT	0...1000	0.1 K/°R	Super heat set point
l_uiPbAlarmValue *	UINT	0...1000	0.1 %	Output in case of super heat or continuous modulation process value not valid Default value: 500
l_uiComAlarmValue *	UINT	0...1000	0.1 %	Output in case of communication alarm Default value: 0
l_uiExtAlarmValue *	UINT	0...1000	0.1 %	Output in case of generic external alarm Default value: 0
l_usiUnitType	USINT	0...9	-	Type of unit to be controlled Default value: 5 (HVAC unit with plate heat exchanger (slow reaction)) Refer to E2_usiUnitType description. (<i>see page 150</i>)
l_xEnMOP *	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> TRUE: enables the MOP regulation FALSE: disable the MOP control Default value: FALSE
l_iMOPSetp *	INT	0...1000	0.1 °C/°F/bar/psi	MOP setpoint Default value: 320
l_usiContModType *	USINT	0...2	-	Continuous modulation type 0=disable, 1=cool, 2=heat Default value: 0
l_iContModSetp *	INT	-1000...1000	0.1 °C/°F/bar/psi	Continuous modulation setpoint Default value: 50
l_uiMaxValue *	UINT	0...1000	0.1 %	Maximum opening value Default value: 1000
l_uiDeadZoneValueMin *	UINT	0...1000	0.1 %	Dead zone minimum opening value Default value: 0
l_uiDeadZoneValueMax *	UINT	0...1000	0.1 %	Dead zone maximum opening value Default value: 1000
l_uiMaxVarOut *	UINT	0...1000	0.1 %	Maximum variation of output Default value: 0 (no limitation)

*: If not connected, the default value is used.

Input	Data type	Range	Scaling/Unit	Description
I_uiValueOnStart *	UINT	0...1000	0.1 %	Fixed output value after restart from stop, hold for the time uiTimeOnStart Default value: 0 If 0, use value stored in E2_uiLastValue to restart
I_uiTimeOnStart *	UINT	0...65535	s	Time with uiValueOnStart before starting the regulation Default value: 0
I_uiValueOnStop *	UINT	0...1000	0.1 %	Fixed value on stop, hold for the time uiTimeOnStop (if 0 abrupt closing) Default value: 0
I_uiTimeOnStop *	UINT	0...65535	s	Time with the fixed value before closing Default value: 0
I_uiValueOnDefrost *	UINT	0...1000	0.1 %	Fixed value on restart from defrost, hold for the time uiTimeOnStart Default value: 0
I_uiHighLoadDelay *	UINT	0...65535	s	Time to wait for MOP control/alarm and dynamic setpoint Default value: 0
I_uiMOPAlarmDelay *	UINT	0...65535	s	Time to trigger the MOP alarm Default value: 180
I_uiOpenAlarmDelay *	UINT	0...65535	s	Time to trigger the full opening alert Default value: 3600
I_uiSuperHeatPb *	UINT	0...1000	0.1 K/°R	Super heat PID proportional band Default value: 100
I_uiSuperHeatTi *	UINT	0...1000	s	Super heat PID integral time constant Default value: 40
I_uiSuperHeatTd *	UINT	0...1000	s	Super heat PID derivative time constant Default value: 0
I_xEnSuperHeatEvo *	BOOL	TRUE / FALSE	-	Enable evolved superheat algorithm Default value: FALSE
E2_uiSuperHeatDz *	UINT	0...1000	0.1 K/°R	Super heat regulation dead band Default value: 5
I_uiMOPPb *	UINT	0...1000	0.1 °C/°F/bar/psi	MOP PID proportional band Default value: 10
I_uiMOPTi *	UINT	0...1000	s	MOP PID integral time constant Default value: 10
I_uiMOPTd	UINT	0...1000	s	MOP PID derivative time constant Default value: 0
I_uiContModPb *	UINT	0...1000	0.1 °C/°F/bar/psi	Continuous modulation PID proportional band Default value: 10
I_uiContModTi *	UINT	0...1000	s	Continuous modulation PID integral time constant Default value: 10
I_uiContModTd *	UINT	0...1000	s	Continuous modulation PID derivative time constant Default value: 0
I_xEnDinSetpCalc *	BOOL	TRUE / FALSE	-	Enable the dynamical super heat set point calculation Default value: FALSE
I_uiDinSetpMaxOffset *	UINT	0...1000	0.1 K/°R	Maximum offset in the dynamical set point calculation Default value: 100
I_uiSetpIncrStep *	UINT	0...1000	0.1 K/°R	Incremental step in the dynamical set point calculation Default value: 1

*: If not connected, the default value is used.

Input	Data type	Range	Scaling/Unit	Description
I_uiSetplncrTime *	UINT	0...1000	s	Step time in the dynamical set point calculation Default value: 20
I_iLastValue	INT	0...1000	0.1 %	Last stored value
*: If not connected, the default value is used.				

i_xEn

If the input `i_xEn` is FALSE, the outputs are set to zero.

This command has the highest priority above `i_xManualMode`, `i_xStart`, and `i_xDefrost`.

i_xStart

If `xEn` = TRUE and `xStart` is set to TRUE, the function block enables the regulation with a start procedure (see description of the parameter `E2_uiValueOnStart` (see page 151)).

When it is reset to FALSE, the selected stop procedure (see description of the parameter `E2_uiValueOnStop` (see page 151)).

`xStart` has less priority than `xEn` and `xManualMode`. So if `xEn` = FALSE, the start procedure cannot take place as well as if `xManualMode` = TRUE a closure routine cannot be performed.

i_xAlarm

If `xAlarm` = TRUE, an alarm flag is set and the open value (`uiOut`) is set at `uiExtAlarmValue` until `xAlarm` = FALSE when there is a bumpless restart of regulators from this value.

i_xComAlarm

If `xComAlarm` = TRUE an alarm flag is set. The open value (`uiOut`) is set at `uiComAlarmValue` until `xComAlarm` = FALSE when there is a bumpless restart of regulators from this value.

This input is differentiated from previous in order to manage differently the communication interruption with the EEV driver.

i_xDefrost

It enables the Defrost procedure which consists on an abrupt closing procedure when the defrost is enabled and in a Start from defrost procedure when `xDefrost` is reset (see description of the parameter `E2_uiValueOnDefrost` (see page 151)).

i_xManualMode, i_uiManualValue

If the function block is enabled and `xManualMode` is set to TRUE, the function block commands an opening of the valve at `uiManualValue` (`uiOut` = `uiManualValue`).

When the pin is reset to FALSE, and `xEn` and `xStart` are enabled, the output goes to the regulated value with a bump-less pass.

i_uiExtLimitValue

This input can be used to force the EEV to close at this value if other regulators drive a greater value. At the same time, the input can be also used to keep other regulators ready to take control if they require a minor opening degree.

If `xEn` = TRUE, `xStart` = FALSE and `xManualMode` is reset to FALSE, the valve abruptly goes to zero. If `xEn` = FALSE the opening command is not performed because the function block is disabled.

i_iSuperHeatPv

Super heat measure from the external calculation.

If an invalid value on the superheat probe is detected, a flag is set and the super heat PID regulator gives as output the constant value `uiPbAlarmValue`.

i_iMOPPv

The value of evaporating pressure or the saturated vapor temperature (a value calculated from pressure). If an invalid value on the pressure or the saturated temperature probe is detected, the system sets a flag and sets the output to zero by disabling the regulation.

i_iContModPv

The value of continuous modulation process variable. This could be for example the temperature of the air in the cabinet to be regulated or the temperature of the water to be chilled, or the pressure in secondary line in CO2 cascade system.

If an invalid value on this process value is detected when the continuous modulation is enabled through `usiContModType`, a flag is set and the continuous modulation PID regulator gives as output the constant value `uiPbAlarmValue`.

Other Inputs

For information on other inputs, refer to Parameter Description (*see page 148*).

Output Pin Description

Output	Data type	Range	Scaling/Unit	Description
q_uiOut	UINT	0...1000	0.1 %	Output value representing the opening level of the EEV
q_usiRegulStatus	USINT	0...9	-	Current regulation status
q_iSupHeatSetpCalc	INT	0...1000	0.1 K/°R	Superheat current setpoint calculated
q_uiSuperHeatPbUnit	UINT	0...1000	0.1 K/°R	Super heat PID proportional band for selected unit type
q_uiSuperHeatTiUnit	UINT	0...1000	s	Super heat PID integral time constant for selected unit type
q_uiSuperHeatTdUnit	UINT	0...1000	s	Super heat PID derivative time constant for selected unit type
q_xEnSuperHeatEvoUnit	BOOL	TRUE / FALSE	-	Enable evolved superheat algorithm for selected unit type
q_uiSuperHeatADzUnit	UINT	0...1000	s	Super heat regulation dead band for selected unit type
q_uiMOPPbUnit	UINT	0...1000	0.1 °C/°F/bar/psi	MOP PID proportional band for selected unit type
q_uiMOPTiUnit	UINT	0...1000	s	MOP PID integral time constant for selected unit type
q_uiMOPTdUnit	UINT	0...1000	s	MOP PID derivative time constant for selected unit type
q_xEnDinSetpCalcUnit	BOOL	TRUE / FALSE	-	Enable the dynamical super heat set point calculation for selected unit type
q_uiDinSetpMaxOffsetUnit	UINT	0...1000	0.1 K/°R	Maximum offset in the dynamical set point calculation for selected unit type
q_uiSetpIncrStepUnit	UINT	0...1000	0.1 K/°R	Incremental step in the dynamical set point calculation for selected unit type
q_uiSetpIncrTimeUnit	UINT	0...1000	s	Step time in the dynamical set point calculation for selected unit type
q_wAlarmID	WORD	0 _h ...FFFF _h	-	Alarm flags
q_wAlertID	WORD	0 _h ...FFFF _h	-	Alert flags

q_uiOut

Output value representing the opening level (%) of the EEV.

q_usiRegulStatus

Indicates the current regulation status:

- 0=OFF
- 1=SH
- 2=MOP
- 3=CONTINUOUS MODULATION
- 4=EXTERNAL LIMITATION
- 5=START
- 6=STOP
- 7=DEFROST
- 8=MANUAL
- 9=ALARM

Refer to the algorithms description (*see page 110*).

q_iSupHeatSetpCalc

Value of current superheat setpoint; it is equal to `iSuperHeatSetp` if `xEnDinSetpCalc = FALSE` otherwise it is dynamically calculated.

Others Outputs

q_uiSuperHeatPbUnit, q_uiSuperHeatTiUnit, q_uiSuperHeatTdUnit, q_xEnSuperHeatEvoUnit, q_uiSuperHeatDzUnit, q_uiMOPpbUnit, q_uiMOPTiUnit, q_uiMOPTdUnit, q_xEnDinSetpCalcUnit, q_uiDinSetpMaxOffsetUnit, q_uiSetpIncrStepUnit, q_uiSetpIncrTimeUnit

These values are selected from preset values if E2_usiUnitType is different from 0 or are equal to correspondent input values (E2_uiSuperHeatPb, E2_uiSuperHeatTi, E2_uiSuperHeatTd, E2_xEnSuperHeatEvo, E2_uiSuperHeatDz, E2_uiMOPpb, E2_uiMOPTi, uiMOPTd, E2_xEnDinSetpCalc, E2_uiDinSetpMaxOffset, uiSetpIncrStep, E2_uiSetpIncrTime) if E2_usiUnitType = 0.

q_wAlarmID

Refer to troubleshooting (*see page 153*).

q_wAlertID

Refer to troubleshooting (*see page 153*).

Parameters Description

The following parameters are automatically generated during device creation in **Configuration** tab. They are linked through correspondent pointers to inputs of the Application Function Block. It is possible to define a parameter as a variable in RAM, changing the pointer initialization to address those variables.

Parameter	Data type	Range	Scaling/Unit	Description
E2_iSuperHeatSetp	INT	0...1000	0.1 K/°R	Super heat set point
E2_uiPbAlarmValue*	UINT	0...1000	0.1 %	Output in case of super heat or continuous modulation process value not valid Default value: 500
E2_uiExtAlarmValue*	UINT	0...1000	0.1 %	Output in case of generic external alarm Default value: 0
E2_uiComAlarmValue*	UINT	0...1000	0.1 %	Output in case of communication alarm Default value: 0
E2_usiUnitType	USINT	0...9	-	Type of unit to be controlled Refer to E2_usiUnitType description. (see page 150) Default value: 5
E2_xEnMOP*	BOOL	TRUE / FALSE	-	<ul style="list-style-type: none"> TRUE: enables the MOP regulation FALSE: disable the MOP control Default value: FALSE
E2_iMOPSetp*	INT	0...1000	0.1 °C/°F	MOP setpoint Default value: 320
E2_usiContModType*	USINT	0...2	-	Continuous modulation type 0=disable, 1=cool, 2=heat Default value: 0
E2_iContModSetp*	INT	-1000...1000	0.1 °C/°F/bar/psi	Continuous modulation setpoint Default value: 50
E2_uiMaxValue*	UINT	0...1000	0.1 %	Maximum opening value Default value: 1000
E2_uiDeadZoneValueMin*	UINT	0...1000	0.1 %	Dead zone minimum opening value Default value: 0
E2_uiDeadZoneValueMax*	UINT	0...1000	0.1 %	Dead zone maximum opening value Default value: 1000
E2_uiMaxVarOut*	UINT	0...1000	0.1 %	Maximum variation of output Default value: 0 (no limitation)
E2_uiValueOnStart*	UINT	0...1000	0.1 %	Fixed output value after restart from stop, hold for the time uiTimeOnStart Default value: 0 If 0, use value stored in E2_uiLastValue to restart
E2_uiTimeOnStart*	UINT	0...65535	s	Time with uiValueOnStart before starting the regulation Default value: 0
E2_uiValueOnStop*	UINT	0...1000	0.1 %	Fixed value on stop, hold for the time uiTimeOnStop (if 0 abrupt closing) Default value: 0
E2_uiTimeOnStop*	UINT	0...65535	s	Time with the fixed value before closing Default value: 0
E2_uiValueOnDefrost*	UINT	0...1000	0.1 %	Fixed value on restart from defrost, hold for the time uiTimeOnStart Default value: 0
E2_uiHighLoadDelay*	UINT	0...65535	s	Time to wait for MOP control/alarm and dynamic setpoint Default value: 0
E2_uiMOPAlarmDelay*	UINT	0...65535	s	Time to trigger the MOP alarm Default value: 180

*: If not connected, the default value is used.

Parameter	Data type	Range	Scaling/Unit	Description
E2_uiOpenAlarmDelay*	UINT	0...65535	s	Time to trigger the full opening alert Default value: 3600
E2_uiSuperHeatPb*	UINT	0...1000	0.1 K/°R	Super heat PID proportional band Default value: 100
E2_uiSuperHeatTi*	UINT	0...1000	s	Super heat PID integral time constant Default value: 40
E2_uiSuperHeatTd*	UINT	0...1000	s	Super heat PID derivative time constant Default value: 0
E2_xEnSuperHeatEvo*	BOOL	TRUE / FALSE	-	Enable evolved superheat algorithm Default value: FALSE
E2_uiSuperHeatDz*	UINT	0...1000	0.1 K/°R	Super heat regulation dead band Default value: 5
E2_uiMOPPb*	UINT	0...1000	0.1 °C/°F	MOP PID proportional band Default value: 10
E2_uiMOPTi*	UINT	0...1000	s	MOP PID integral time constant Default value: 10
E2_uiMOPTd	UINT	0...1000	s	MOP PID derivative time constant Default value: 0
E2_uiContModPb*	UINT	0...1000	0.1 °C/°F/bar/psi	Continuous modulation PID proportional band Default value: 10
E2_uiContModTi*	UINT	0...1000	s	Continuous modulation PID integral time constant Default value: 10
E2_uiContModTd*	UINT	0...1000	s	Continuous modulation PID derivative time constant Default value: 0
E2_xEnDinSetpCalc*	BOOL	TRUE / FALSE	-	Enable the dynamical super heat set point calculation Default value: FALSE
E2_uiDinSetpMaxOffset*	UINT	0...1000	0.1 K/°R	Maximum offset in the dynamical set point calculation Default value: 100
E2_uiSetpIncrStep*	UINT	0...1000	0.1 K/°R	Incremental step in the dynamical set point calculation Default value: 1
E2_uiSetpIncrTime*	UINT	0...1000	s	Step time in the dynamical set point calculation Default value: 20
E2_iLastValue	INT	0...1000	0.1 %	Last stored value
*: If not connected, the default value is used.				

E2_iSuperHeatSetp

Set point of the super heat regulation.

E2_uiPbAlarmValue

If an invalid value on the superheat or continuous modulation probe is detected, a flag is set and the super heat or continuous modulation (if enabled) PID regulator gives as output the constant value E2_uiPbAlarmValue.

E2_uiExtAlarmValue

If i_xAlarm = TRUE, an alarm flag is set and the open value (q_uiOut) is set at E2_uiExtAlarmValue until i_xAlarm = FALSE when there is a bumpless restart of regulators from this value.

E2_uiComAlarmValue

If `i_xComAlarm = TRUE`, an alarm flag is set and the open value (`q_uiOut`) is set at `E2_uiComAlarmValue` until `i_xComAlarm = FALSE` when there is a bumpless restart of regulators from this value.

E2_usiUnitType

Type of unit to be controlled.

If `E2_usiUnitType = 0`, parameters `E2_uiSuperHeatPb`, `E2_uiSuperHeatTi`, `E2_uiSuperHeatTd`, `E2_xEnSuperHeatEvo`, `E2_uiSuperHeatDz`, `E2_uimOPPb`, `E2_uimOPTi`, `E2_uimOPTd`, `E2_xEnDinSetpCalc`, `E2_uiDinSetpMaxOffset`, `E2_uiSetpIncrStep`, `E2_uiSetpIncrTime` are used as input for regulation.

If `E2_usiUnitType` is different from 0, those parameters are automatically set (the used values are available on output) in order to cope with different type of unit or machine that can be found in HVAC or refrigeration:

E2_usiUnitType	Description of Unit
1	Ducted refrigeration unit and evaporation pressure quickly variable (for example step control)
2	Ducted refrigeration unit and evaporation pressure controlled (for example INVERTER control)
3	Refrigeration unit with on-board compressor
4	Refrigeration unit with on-board compressor and regenerative heat exchanger
5	HVAC unit with plate heat exchanger (slow reaction)
6	HVAC unit with shell and tube heat exchanger (medium reaction)
7	HVAC unit with finned coil heat exchanger (moderately fast reaction)
8	HVAC unit with variable cooling capacity (fast reaction)
9	Perturbed HVAC unit (very fast reaction)

It is possible to customize the list of predefined units. To do this, you can modify values in **Init value** field of the array `aauiEEVCntrlParams`, inside the `FB_MyEEVDCntrl` AFB Local variables list.

Each row in the array corresponds to the `E2_usiUnitType` value (for example, the first row corresponds to `E2_usiUnitType = 1`) and the values in each row correspond to the values assumed from the parameters in the previous list.

E2_xEnMOP

It enables the Maximum Operating Pressure (MOP) regulation.

E2_iMOPSetp

The maximum saturated vapor temperature. This is the set point of the MOP PID.

When approaching this value, the MOP regulation starts to close the valve in order to come back to a safety operating mode.

This is a protection for the compressors in the refrigeration circuit when there is a significant thermal load. In that case, superheat control is abandoned but closing action is kept at minimum in order to start again to regulate when this load situation disappears.

E2_usiContModType

This parameter enables the continuous modulation regulation and permit to set a heat or cool PID regulation. Heat regulation means that the output (`q_uiOut`) increases as the process variable is far below the setpoint, while cool regulation means that the output increases as the process variable is below the setpoint.

E2_iContModSetp

The continuous modulation setpoint.

When approaching this value, the continuous modulation regulation starts to close the valve in order to maintain the setpoint while abandoning the superheat control. In that case, superheat control is abandoned but closing action is kept at minimum in order to start again to regulate superheat when this setpoint is far.

This permits to use the EEV to control another variable for example the cabinet air temperature or the water temperature in a chiller; while keeping superheat as low as possible and preventing at the same time to go beyond the MOP.

E2_uiMaxValue, E2_uiDeadZoneValueMin, E2_uiDeadZoneValueMax

The open value (q_{uiOut}) is never greater than $E2_uiMaxValue$.

If the calculated opening value is greater than $E2_uiDeadZoneValueMax$, $q_{uiOut} = E2_uiMaxValue$ (if $E2_uiMaxValue < E2_uiDeadZoneValueMax$ this function is ignored).

If the calculated opening value is smaller than $E2_uiDeadZoneValueMin$, $q_{uiOut} = 0.0$.

E2_uiMaxVarOut

To limit too quick variations of open value (q_{uiOut}) that can create oscillations in superheat and mechanics issues, it is possible to set a maximum variation of opening degree (%) per second through the parameter $E2_uiMaxVarOut$. This value should be arranged depending on the mechanics of EEV, for example it can be set less than maximum admitted speed for the particular EEV. If $E2_uiMaxVarOut = 0$, q_{uiOut} is not limited in speed.

E2_uiValueOnStart, E2_uiTimeOnStart

Start procedure ($i_xStart = TRUE$ from FALSE).

If $E2_uiValueOnStart$ differs from zero, the regulator fixes the opening value to $E2_uiValueOnStart$ for $E2_uiTimeOnStart$ seconds. After this time is elapsed the regulation starts from this opening value, as for the super heat, as for the MOP (if enabled).

If $E2_uiValueOnStart$ is equal to zero, the regulator fixes the opening value to the opening value recorded before the stop and stored in EEPROM into the address $ptriLastValue$.

After the $E2_uiTimeOnStart$ time is elapsed, the regulation starts.

E2_uiValueOnStop, E2_uiTimeOnStop

Stop procedure

When a stop instance occurs ($i_xStart = FALSE$ from TRUE), the system saves the last opening value and it stores it into the EEPROM, at the address passed with $ptriLastValue$.

After this the closure is performed, particularly:

- If $E2_uiValueOnStop$ is different from 0, the EEV stay in a fixed opening value ($E2_uiValueOnStop$) for a certain time ($E2_uiTimeOnStop$) before the complete closure is performed ($q_{uiOut} = 0$),
- If $E2_uiValueOnStop = 0$, the abrupt closure is performed ($q_{uiOut} = 0$).

E2_uiValueOnDefrost

Start from defrost procedure

The Start from defrost consists on keeping the value $E2_uiValueOnDefrost$ if it is different from zero, for $E2_uiTimeOnStart$ seconds. If $E2_uiValueOnDefrost = 0$, the fixed opening value on restart is the last opening value stored in EEPROM. After the time is elapsed the regulation can restart from this value.

E2_uiHighLoadDelay

After the block activation (i_xEn transition to TRUE) or after a defrost ending condition ($i_xDefrost$ transition to FALSE) for a time $E2_uiHighLoadDelay$ MOP alarm is not monitored, and MOP control is not performed. Also dynamic setpoint calculation is frozen.

E2_uiMOPAlarmDelay

If $E2_iMOPSetp$ is passed for a time longer than $E2_uiMOPAlarmDelay$, the function block puts the output to zero ($q_{uiOut} = 0$) and the MOP alarm flag is set. This alarm is not monitored for a time $E2_uiHighLoadDelay$ after the block activation (i_xEn transition to TRUE) or after a defrost ending condition ($i_xDefrost$ transition to FALSE).

E2_uiOpenAlarmDelay

if `i_xManualMode = FALSE` and the valve stays open at maximum value (`q_uiOut = E2_uiMaxValue`) for a time longer than `E2_uiOpenAlarmDelay` an alarm flag occurs. The alarm reset automatically if the output changes to a smaller value.

E2_uiSuperHeatPb, E2_uiSuperHeatDz, E2_uiSuperHeatTi, E2_uiSuperHeatTd

Super heat standard PID controller parameters.

E2_xEnSuperHeatEvo, E2_uiSuperHeatPb, E2_uiSuperHeatDz, E2_uiSuperHeatTi

Evolved super heat controller parameters. To enable this controller instead of standard PID, `E2_xEnSuperHeatEvo` have to be set to TRUE.

E2_uiMOPFb, E2_uiMOPTi, E2_uiMOPTd

Maximum Operating Pressure PID controller parameters.

E2_uiContModPb, E2_uiContModTi, E2_uiContModTd

Continuous modulation PID controller parameters.

E2_xEnDinSetpCalc, E2_uiDinSetpMaxOffset, E2_uiSetpIncrStep, E2_uiSetpIncrTime

if `E2_xEnDinSetpCalc = FALSE` the super heat set point (`E2_iSuperHeatSetp`) is given directly to the super heat PID (`E2_iSuperHeatSetpCalc`) and only the parameter check is performed

If `E2_xEnDinSetpCalc = TRUE` the dynamic set point calculation is enabled and the super heat set point is calculated with the following routine.

For a time `E2_uiHighLoadDelay` after the block activation (`i_xEn` transition to TRUE) or after a defrost ending condition (`i_xDefrost` transition to FALSE):

$$E2_iSuperHeatSetpCalc = E2_iSuperHeatSetp + E2_uiDinSetpMaxOffset.$$

After that there is a dynamic set point calculation if superheat regulator is acting (`q_usiRegulStatus = 1`) with a timing of `E2_uiSetpIncrTime`:

If `i_iSuperHeatPv > E2_iSuperHeatSetp`, `q_iSuperHeatSetpCalc = iSuperHeatRegSetp - E2_uiSetpIncrStep`

if `i_iSuperHeatPv < E2_iSuperHeatSetp`, `q_iSuperHeatSetpCalc = iSuperHeatRegSetp + E2_uiSetpIncrStep`

Lastly, the calculated set point is forced to stay above `E2_iSuperHeatSetp` and under `E2_iSuperHeatSetp + E2_uiDinSetpMaxOffset`.

E2_uiLastValue

This is not a parameter, this is variable (EEPROM or RAM) that stores the last opening value of the valve, before a stop, a defrost or an alarm instance occurs. This value is used in the start procedure if `uiValueOnStart = 0` and in start from defrost procedure if `uiValueOnDefrost = 0`.

Troubleshooting

Alarm ID Description

The `q_wAlarmID` output represents a value from 0 to 65535, whereby each bit represents an alarm. The bits and their description are described in the following table:

Bit	Alarm condition	Effect
0	Super heat PID parameter (Pb, Ti, Td) out of range	block disabled: <code>uiOut</code> goes to zero
1	MOP PID parameter (Pb, Ti, Td) out of range	
2	Continuous modulation PID parameter (Pb, Ti, Td) out of range	
3	Super heat dead band	
4	<code>E2_iSuperHeatSetp</code> , <code>E2_iMopSetp</code> or <code>iContModSetp</code> out of range	
5	<code>E2_uiPbAlarmValue</code> or <code>E2_uiExtAlarmValue</code> or <code>uiComAlarmValue</code> out of range	
6	<code>E2_uiTimeOnStart</code> or <code>E2_uiTimeOnStop</code> out of range	
7	<code>E2_uiValueOnDefrost</code> or <code>E2_uiValueOnStart</code> or <code>E2_uiValueOnStop</code> or <code>E2_uiExtLimitValue</code> or <code>E2_uiManualValue</code> out of range	
8	Dynamical set point parameters out of range	
9	Maximum open value or maximum output variation or dead zone values out of range	
10	<code>E2_usiUnitType</code> or <code>E2_usiContModType</code> out of range	
11	Saturated temperature probe out of range	
12	MOP alarm	block disabled: <code>uiOut</code> goes to fixed value
13	External alarm	
14	Communication alarm	
15	Error detected on EEPROM writing or incorrect task	

Alarm ID Probable Cause

The probable cause linked to each alarm bit is presented in the following table:

Alarm bit	Alarm Details
0...10	To re-enable the function block, you should modify the wrong parameter with a valid value, disable the function block (<code>i_xEn = FALSE</code>), then enable again (<code>i_xEn = TRUE</code>).
11	When the probe has an invalid value (less than -32760 and more than 32760) and <code>E2_xEnMop</code> is set to TRUE, the output goes to zero. If the probe returns to admitted values, the regulation restarts from zero and goes with a bump-less pass to the new output of the regulation.
12	The MOP alarm occurs if <code>E2_xEnMop</code> is set to TRUE, if <code>i_xStart</code> is TRUE and the <code>E2_iMOPSetp</code> is exceeded for a time longer than <code>E2_uiMOPAlarmDelay</code> . The alarm automatically disables if the saturated temperature measure (<code>i_iMopPv</code>) goes under the limit (<code>E2_iMOPSetp</code>) with <code>i_xStart</code> is TRUE or if the <code>E2_xEnMop</code> is disabled. When it restarts the output goes to the regulation value with a bump-less pass, starting from zero.
13	The alarm occurs if <code>i_xAlarm = TRUE</code> and is automatically reset if <code>i_xAlarm = FALSE</code> .
14	The alarm occurs if <code>i_xComAlarm = TRUE</code> and is automatically reset if <code>i_xComAlarm = FALSE</code> .
15	The alarm occurs if <code>E2_uiValueOnStart</code> or <code>E2_uiValueOnDefrost</code> are set to 0 and there is an issue while writing in EEPROM at beginning of stop or defrost procedure, for example if <code>prtiLastValue</code> is not connected or pointer is not correctly assigned, or cause of a memory issue. To let the block work again, the EEPROM issues shall be fixed and the error flag shall be set by disabling the function block (<code>i_xEn = FALSE</code>) an enabling it again (<code>i_xEn = TRUE</code>).

Alert ID Description

The `uiAlertID` output represents a value from 0 to 65535, whereby each bit represents an alert. The bits and their description are described in the following table:

Bit	Alert condition	Effect
0	Super heat probe out of range	<code>iOut</code> goes to <code>uiPbAlarmValue</code>
1	Continuous modulation probe out of range	<code>iOut</code> goes to <code>uiPbAlarmValue</code>
2	Open alert	Alert the user of the open situation
3	<code>E2_uiTimeOnStop</code> changed runtime	The block runs with the old value
4	<code>E2_uiTimeOnStart</code> changed runtime	The block runs with the old value
5	<code>E2_uiMopAlarmDelay</code> changed runtime	The block runs with the old value
6	<code>E2_uiOpenAlarmDelay</code> changed runtime	The block runs with the old value
7	<code>E2_uiHighLoadDelay</code> changed runtime	The block runs with the old value
8	<code>E2_usiUnitType</code> changed runtime	Only alert (parameters are changed so)
9...15	Reserved	-

Alert ID Probable Cause

The probable cause linked to each alert bit is presented in the following table:

Alert bit	Alert Details
0...1	When the probe has an invalid value (less than -32760 and more than 32760), the output goes to <code>E2_uiPbAlarmValue</code> . If the probe returns to admitted values, the regulation restart from <code>E2_uiPbAlarmValue</code> and goes with a bump-less pass to the new output of the regulation.
2	The opening alert appears when the output stays to the maximum value (<code>E2_uiMaxValue</code>) for a time longer than <code>E2_uiOpenAlarmDelay</code> . It is disabled if the parameter <code>E2_uiOpenAlarmDelay</code> is zero. This alarm produces only a flag and it does not affect the function of the machine. The flag resets by itself when the output fall from the maximum value to another.
3...7	To modify one of these parameters, you should disable and re-enable the block through <code>i_xEn</code> .
8...15	Reserved

Implementation

Internal Variables Description

These variables are automatically initialized to right values if automatic instantiation is used. Otherwise, you must provide your own values.

Parameter	Data Type	Description
ptriSuperHeatSetp	@INT	Pointer to parameter
ptruiPbAlarmValue	@UINT	Pointer to parameter
ptruiExtAlarmValue	@UINT	Pointer to parameter
ptruiComAlarmValue	@UINT	Pointer to parameter
ptrusiUnitType	@USINT	Pointer to parameter
ptrxEnMOP	@BOOL	Pointer to parameter
ptriMOPSetp	@INT	Pointer to parameter
ptrusiContModType	@USINT	Pointer to parameter
ptriContModSetp	@INT	Pointer to parameter
ptruiMaxValue	@UINT	Pointer to parameter
ptruiDeadZoneValueMin	@UINT	Pointer to parameter
ptruiDeadZoneValueMax	@UINT	Pointer to parameter
ptruiMaxVarOut	@UINT	Pointer to parameter
ptruiValueOnStart	@UINT	Pointer to parameter
ptruiTimeOnStart	@UINT	Pointer to parameter
ptruiValueOnStop	@UINT	Pointer to parameter
ptruiTimeOnStop	@UINT	Pointer to parameter
ptruiValueOnDefrost	@UINT	Pointer to parameter
ptruiHighLoadDelay	@UINT	Pointer to parameter
ptruiMOPAlarmDelay	@UINT	Pointer to parameter
ptruiOpenAlarmDelay	@UINT	Pointer to parameter
ptruiSuperHeatPb	@UINT	Pointer to parameter
ptruiSuperHeatTi	@UINT	Pointer to parameter
ptruiSuperHeatTd	@UINT	Pointer to parameter
ptrxEnSuperHeatEvo	@BOOL	Pointer to parameter
ptruiSuperHeatDz	@UINT	Pointer to parameter
ptruiMOPPb	@UINT	Pointer to parameter
ptruiMOPTi	@UINT	Pointer to parameter
ptruiMOPTd	@UINT	Pointer to parameter
ptruiContModPb	@UINT	Pointer to parameter
ptruiContModTi	@UINT	Pointer to parameter
ptruiContModTd	@UINT	Pointer to parameter
ptrxEnDinSetpCalc	@BOOL	Pointer to parameter
ptruiDinSetpMaxOffset	@UINT	Pointer to parameter
ptruiSetpIncrStep	@UINT	Pointer to parameter
ptruiSetpIncrTime	@UINT	Pointer to parameter
ptriLastValue	@INT	Pointer to parameter

Chapter 19

EVEVD••000500 Evaporator Temperature and Superheat Calculation: SuperHeatCalc

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Evaporator Temperature and Superheat Calculation (SuperHeatCalc)	158
Input Pin Description	159
Output Pin Description	161

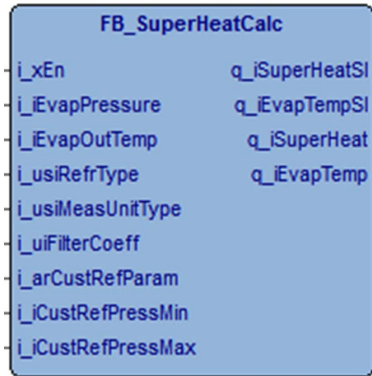
Evaporator Temperature and Superheat Calculation (SuperHeatCalc)

Functional Overview

This Application Function Block calculates the evaporator temperature and superheat , also providing measurement unit conversion depending on the measurement system selected.

Input Pin Description

I/O Description:



Input	Data type	Range	Scaling/Unit	Description
i_xEn	BOOL	TRUE / FALSE	-	Enables the Function Block
i_EvaporatorPressure	INT	-32760...32760	0.01 bar/psi	Evaporator pressure Values lower than -32760 and greater than 32760 are invalid
i_iEvapOutTemp	UINT	-32760...32760	0.1 °C/°F	Evaporator output temperature Values lower than -32760 and greater than 32760 are invalid
i_usiRefrType	USINT	0...255	-	Refrigerant type 255 = custom refrigerant Default value:0
i_usiMeasUnitType	USINT	0...1	-	Measurement unit type: <ul style="list-style-type: none"> ● 0 = International System (SI) (default value) ● 1 = Imperial System
i_uiFilterCoeff*	UINT	0...10000	-	Coefficient of filter: Default value: 10000
i_arCustRefParam*	REAL[5]	-	-	Custom Refrigerant Parameters Array
i_iCustRefPressMin*	INT	-	0.01 Bar	Custom Refrigerant minimum pressure
i_iCustRefPressMax*	UINT	-	0.01 Bar	Custom Refrigerant maximum pressure Default value:0

*: If not connected, the default value is used.

i_xEn

If the input i_xEn is FALSE, the outputs are set to zero.

i_iEvaporatorPressure

The value of evaporating pressure. It comes from an evaporator pressure sensor in the suction line of the refrigeration circuit.

If an invalid value on the pressure sensor is detected, the outputs are set to the 8000_h.

It is used to calculate q_iEvapTemp and q_iSuperHeat, q_iEvapTempSI, and q_iSuperHeatSI depending on i_usiRefrType.

i_iEvapOutTemp

The value of evaporator output temperature. It comes from temperature sensor placed in the pipe of the suction line of the refrigeration circuit between the evaporator and the compressor usually.

If an invalid value on the temperature probe is detected, the outputs q_iSuperHeat and q_iSuperHeatSI are set to the 8000_h.

This value is internally used to calculate $q_{iSuperHeat}$ and $q_{iSuperHeatSI}$ depending on $i_usiRefrType$.

$i_usiRefrType$

Indicates the refrigerant to be considered for pressure/temperature conversion. The valid range of values is indicated in `Press2Temp` Application Function Block of Application library. A value of 255 permits to you to implement custom refrigerant.

$i_iMeasUnitType$

Indicates which measurement units must be considered for inputs and outputs representing pressures, temperatures, or temperature difference.

A value of 0 indicates the International System (SI) that means bar, °C, and K; a value of 1 indicates the Imperial System that means psi, °F, °R.

$i_uiFilterCoeff$

The calculated evaporator temperature can be filtered through a low pass filter having the formula:

$$y(n) = \alpha * x(n) + (1 - \alpha) * y(n - 1)$$

With:

$$\alpha = \frac{i_uiFilterCoeff}{10000}$$

This can be useful when the dynamics of pressure sensor are faster than those of temperature sensor, determining a dynamic faulty superheat estimation.

$i_arCustRefParam$

Array containing coefficients of polynomial to be used with custom refrigerant calculation (see description of `Press2Temp` (see *EcoStruxure Machine Expert - HVAC, Application Function, HVAC Library Guide*)).

If $i_usiRefrType \neq 255$, this input could be not connected.

$i_iCustRefPressMin$

Minimum valid pressure value to be used with custom refrigerant calculation (see description of `Press2Temp` (see *EcoStruxure Machine Expert - HVAC, Application Function, HVAC Library Guide*)).

If $i_usiRefrType \neq 255$, this input could be not connected.

$i_iCustRefPressMax$

Maximum valid pressure value to be used with custom refrigerant calculation (see description of `Press2Temp` (see *EcoStruxure Machine Expert - HVAC, Application Function, HVAC Library Guide*)).

If $i_usiRefrType \neq 255$, this input could be not connected.

Output Pin Description

Output	Data type	Range	Scaling/Unit	Description
q_iSuperHeatSI	INT	-32760...32760	0.1 K	Current super heat calculated
q_iEvapTempSI	INT	-32760...32760	0.1 °C	Current evaporation temperature calculated
q_iSuperHeat	INT	-32760...32760	0.1 K/°R	Current super heat calculated
q_iEvapTemp	INT	-32760...32760	0.1 °C/°F	Current evaporation temperature calculated

q_iSuperHeatSI

Value of current superheat calculated; it is equal to the difference between `i_iEvapOutTemp` and `q_iEvapTemp`. The measurement unit is K, `i_usiMeasUnitType` is not considered.

q_iEvapTempSI

Value of current evaporation temperature calculated; it is the saturation temperature in the suction line of the refrigeration circuit, calculated depending on evaporation pressure and refrigerant (see description of `Press2Temp` (see *EcoStruxure Machine Expert - HVAC, Application Function, HVAC Library Guide*)). The measurement unit is °C, independently from `i_usiMeasUnitType`.

q_iSuperHeat

Value of current superheat calculated; it is equal to the difference between `i_iEvapOutTemp` and `q_iEvapTemp`.

q_iEvapTemp

Value of current evaporation temperature calculated; it is the saturation temperature in the suction line of the refrigeration circuit, calculated depending on evaporation pressure and refrigerant (see description of `Press2Temp` (see *EcoStruxure Machine Expert - HVAC, Application Function, HVAC Library Guide*)).

Chapter 20

EVEVD••000500 Alarm Management: EEVDAlarmMgmt

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Alarm Management (EEVDAlarmMgmt)	164
Input Pin Description	165
Output Pin Description	167

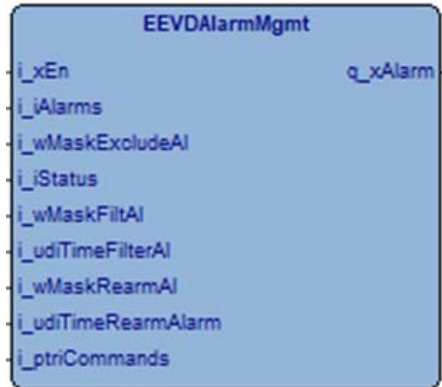
Alarm Management (EEVDAlarmMgmt)

Functional Overview

This Application Function Block permits to exclude, filter, and automatically rearm alarms in the EEV driver.

Input Pin Description

I/O Description:



Input	Data type	Range	Scaling/Unit	Description
i_xEn	BOOL	TRUE / FALSE	-	Enables the function block
i_iAlarms	INT	-	-	Diagnostics from EEVD
i_wMaskExcludeAl *	WORD	-	-	Bitmask to exclude alarms Default value:0
i_iStatus *	INT	-	-	Status from EEVD Default value:0
i_wMaskFiltAl *	WORD	-	-	Bitmask to filter alarms Default value:0
i_udiTimeFilterAl *	UDINT	-	ms	Filter time for Alarms Default value:10000
i_wMaskRearmAl *	WORD	-	-	Bitmask to rearm alarms Default value:0
i_udiTimeRearmAlarm *	UDINT	-	ms	Repeat time for rearming of alarms Default value:30000
I_ptrCommands *	@INT	-	-	Pointer to Commands for EEVD Default value:0

*: If not connected, the default value is used.

i_xEn

If the input i_xEn is False, the output is FALSE and the timers for filter and rearm are reset.

i_iAlarms

Alarms variable, bit-addressable, read from EEV driver.

For example, the field variable iEEVDAlarmsEEVx.

For more details, refer to Alarms description ([see page 77](#)).

i_wMaskExcludeAl

Bitmask that indicates which bit alarms must be excluded from detection. For example to exclude bit 11, 12 and bit 13 alarms set 16#3800 (=2#0011 1000 0000 0000).

i_iStatus

Status variable, read from EEV driver.

For example, the field variable iEEVDStatusEEVx (see Status description).

If not connected, rearm action will not work.

i_wMaskFiltAl

Bitmask that indicates which bit alarms must be filtered before detection. For example to filter bit 11, 12, and bit 13 alarms set 16#3800 (=2#0011 1000 0000 0000).

i_udiTimeFilterAl

This value indicates how long (ms) to wait for alarms selected by `i_wMaskFiltAl` (and not excluded by `i_wMaskExcludeAl`) to become stable before triggering the transition in the output.

That means for example that, starting from a condition of no alarm (`q_xAlarm = FALSE`), if one of alarms indicated by `i_wMaskFiltAl` becomes TRUE and stay stable for at least `i_udiTimeFilterAl` ms then the output `q_xAlarm` becomes TRUE. In the opposite way, once all selected alarms become FALSE staying stable for at least `i_udiTimeFilterAl` ms, and no other alarm (not excluded) is active, then the output `q_xAlarm` becomes FALSE.

i_wMaskRearmAl

Bitmask that indicates which bit alarms must trigger a rearm action. For example, to trigger the rearm for bit 11, 12 and bit 13 alarms set 16#3800 (=2#0011 1000 0000 0000).

i_udiTimeRearmAl

This value indicates the period of rearming in ms. That means that if an alarm (not excluded), selected by the mask `i_wMaskRearmAl` is triggered (after filtering, if selected), a rearm action is started after `i_udiTimeRearmAl` ms and repeated periodically.

The rearm action consists in setting the proper bit (bit 0) in the variable pointed by `i_ptrCommands`, then checking status by the input `i_iStatus` and waiting proper timing before resetting that bit.

i_ptrCommands

Pointer to the commands variable, bit-addressable, written in the EEV driver, for example the field variable `iEEVDCommandsEEVx` (see Command description). If not connected, rearm action is not possible.

Output Pin Description

Output	Data type	Range	Scaling/Unit	Description
q_xAlarm	BOOL	TRUE / FALSE	-	Alarm from EEVD

q_xAlarm

Indicates that a blocking alarm is detected in the EEVD and control algorithm must manage it.



A

AWG

(*American wire gauge*) The standard that specifies wire section sizes in North America.

C

controller

Automates industrial processes (also known as programmable logic controller or programmable controller).

COP

(*Coefficient Of Performance*) It is a ratio of cooling provided to work required.

E

EEV

(*Electronic Expansion Valve*)

EEVD

(*Electronic Expansion Valve Driver*)

EIA

(*electronic industries alliance*) The trade organization for establishing electrical/electronic and data communication standards (including RS-232 and RS-485) in the United States.

EN

EN identifies one of many European standards maintained by CEN (*European Committee for Standardization*), CENELEC (*European Committee for Electrotechnical Standardization*), or ETSI (*European Telecommunications Standards Institute*).

expansion bus

An electronic communication bus between expansion I/O modules and a controller.

F

firmware

Represents the BIOS, data parameters, and programming instructions that constitute the operating system on a controller. The firmware is stored in non-volatile memory within the controller.

H

HVAC&R

(*heating, ventilation, and air conditioning and refrigeration*)

I

IEC 61131-3

Part 3 of a 3-part IEC standard for industrial automation equipment. IEC 61131-3 is concerned with controller programming languages and defines 2 graphical and 2 textual programming language standards. The graphical programming languages are ladder diagram and function block diagram. The textual programming languages include structured text and instruction list.

IP20

(*ingress protection*) The protection classification according to IEC 60529 offered by an enclosure, shown by the letter IP and 2 digits. The first digit indicates 2 factors: helping protect persons and for equipment. The second digit indicates helping protect against water. IP20 devices help protect against electric contact of objects larger than 12.5 mm, but not against water.

L

LAN

(local area network) A short-distance communications network that is implemented in a home, office, or institutional environment.

M

MOP

(Maximum Operating Pressure)

N

network

A system of interconnected devices that share a common data path and protocol for communications.

S

SELV

(safety extra low voltage) A system that follows IEC 61140 guidelines for power supplies is protected in such a way that voltage between any 2 accessible parts (or between 1 accessible part and the PE terminal for class 1 equipment) does not exceed a specified value under normal conditions or under inoperable conditions.



C

CAN expansion bus, *60*
Communication, *59*

E

EVEVD1B000500, *46*
EVEVD1U000500, *44*
EVEVD2B000500, *48*
EVEVDBA000000, *51*

P

Power Supply, *54*

S

Specifications
 Battery Backup, *41*
 Electronic Valve Drivers, *41*