HEINZMANN® Energy requires Control

Heinzmann GmbH & Co. KG
Engine & Turbine Management

Am Haselbach 1
D-79677 Schönau/Germany

Phone: +49 7673 8208 - 0
Fax: +49 7673 8208 - 188
E-mail info@heinzmann.de
www.heinzmann.com

V.A.T. No.: DE145551926

HEINZMANN®
Digital Positioners

PANDAROS
Positioner DC 6-06

Manual DG 08 001-e / 10-08

DANGER

The appropriate manuals must be thoroughly studied before installation, initial start-up and maintenance.

All instructions pertaining to the system and safety must be followed in full. Non-observance of the instructions may lead to injury to persons and/or material damage.

HEINZMANN shall not be held liable for any damage caused through non-observance of instructions.

Independent tests and inspections are of particular importance for all applications in which a malfunction could result in injury to persons or material damage.

All examples and data, as well as all other information in this manual are there solely for the purpose of instruction and they may not be used for special application without the operator running independent tests and inspections beforehand.

HEINZMANN does not guarantee, neither expressly nor tacitly, that the examples, data or other information in this manual is free from error, complies with industrial standards or fulfils the requirements of any special application.

WARNING

To avoid any injury to persons and damage to systems, the following monitoring and protective systems must be provided:

- overspeed protection independent of the rpm controller

HEINZMANN shall not be held liable for any damage caused through missing or insufficiently rated overspeed protection.

- thermal overload protection

The following must also be provided for alternator systems:

- Overcurrent protection
- Protection against faulty synchronisation for excessively-large frequency, voltage or phase difference
- Directional contactor

The reasons for overspeeding may be:

- Failure of positioning device, control unit or its auxiliary devices
- Linkage sluggishness and jamming

The following must be observed before an installation:

- Always disconnect the electrical mains supply before any interventions to the system.
- Only use cable screening and mains supply connections that correspond with the *European Union EMC Directive*
- Check the function of all installed protection and monitoring systems

	Please observe the following for electronically controlled injection (MVC): – For **common rail** systems each injector line must be equipped with a separate mechanical flow-rate limiter. – For **unit pump (PLD)** and **pump-injector unit (PDE)** systems, the fuel enable is first made possible by the solenoid valve’s control plunger motion. This means that in the event of the control plunger sticking, the fuel supply to the injection valve is stopped.
	As soon as the positioning device receives power, it can actuate the controller output shaft automatically at any given time. The range of the controller shaft or control linkage must therefore be secured against unauthorised access.
	HEINZMANN expressly rejects any implied guarantee pertaining to any marketability or suitability for a special purpose, including in the event that **HEINZMANN** was notified of such a special purpose or the manual contains a reference to such a special purpose.
	HEINZMANN shall not be held liable for any indirect and direct damage nor for any incidental and consequential damage that results from application of any of the examples, data or miscellaneous information as given in this manual.
	HEINZMANN shall not provide any guarantee for the design and planning of the overall technical system. This is a matter of the operator its planners and its specialist engineers. They are also responsible for checking whether the performances of our devices match the intended purpose. The operator is also responsible for a correct initial start-up of the overall system.

Table of Contents

Page

1 Safety instructions and the signal words and symbols used 6
1.1 Safety measures under normal operation 7
1.2 Safety measures for maintenance and servicing 7

2 General 8
2.1 General System Description 8
2.2 Firmware 8
2.2.1 HEINZMANN Basic Software 9
2.2.2 Custom firmware 9
2.3 Other Information 10
2.4 Conventions 11
2.5 Parameter Lists 12
2.6 Levels 12
2.7 Block Diagram 13

3 Parameterization of the HEINZMANN control units 15
3.1 Parameterization options 15
3.2 Saving the data 16
3.3 DcDesk 2000 16
3.4 Value range of parameters 17
3.5 Activating functions 18
3.6 Parameterization examples 18
3.7 Resetting a control unit 18

4 Setpoint 19
4.1 Design of the setpoint generator 19
4.1.1 Analogue setpoint 19
4.1.2 PWM Setpoint 20
4.2 Determining the actuator setpoint 20

5 System monitoring 22
5.1 Operation mode 22
5.2 Stopped state 22
5.2.1 Stop request 23
5.2.1.1 External stop request by a digital input 23
5.2.1.2 External stop request by an analogue input 24
5.3 Setpoint monitoring 25

5.3.1 Monitoring the analogue setpoint 25
5.3.2 Monitoring the PWM setpoint 26
5.4 Monitoring the analogue stop signal 26
5.5 Electronics monitoring 27
5.5.1 Voltage references 27
5.5.2 RAM memory test 27
5.5.3 Program memory test 28
5.5.4 Stack Depth Test 28
5.5.5 Program flow test 28
5.5.6 Operating voltage monitoring 28

6 Inputs and outputs 30
6.1 Switch functions via digital inputs 30
6.1.1 Assignment of digital inputs 31
6.2 Setpoint selection and stop request via analogue inputs 32
6.2.1 Adjusting the current/voltage inputs 32
6.2.2 Filtering the analogue inputs 33
6.2.3 Error detection with the analogue inputs 33
6.3 Setpoint selection by PWM input 34
6.3.1 Filtering the PWM inputs 35
6.3.2 Error detection with the PWM inputs 36
6.4 Actuator position output with analogue output 36
6.4.1 Value range of the analogue output 36
6.5 Actuator position output with PWM output 37
6.5.1 Value range of the PWM output 37
6.6 Common alarm via alarm output 37
6.7 Auto adjust with pushbutton 38

7 Technical data 39
7.1 General 39
7.2 Inputs and Outputs 40
7.3 Terminal connection diagrams 41
7.3.1 Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor 42
7.3.2 Analogue setpoint, stop signal digital and/or analogue, gearless actuator 43
7.3.3 PWM setpoint, stop signal digital, actuator with disc rotor motor 44
7.3.4 PWM setpoint, stop signal digital, gearless actuator 45
7.3.5 PWM setpoint, stop signal analogue, actuator with disc rotor motor 46
7.3.6 PWM setpoint, stop signal analogue, gearless actuator 47

8 Actuator and feedback 48
8.1 Calibrating the actuator 49
8.1.1 Manual calibration 49

8.1.2 Automatic calibration 50
8.1.2.1 Saving the calibration data 51
8.1.3 Error detection from the feedback 51
8.2 Limiting the actuator stroke 52
8.3 Servo loop 52
8.3.1 Servo control loop 52
8.3.2 Actuator current 53
8.4 Test mode 54

9 Data Management 56
9.1 Serial Number of the Control Unit 56
9.2 Identification of the Control Unit 56
9.3 Identification of the PC Program and Hand Programmer 57

10 Error handling 58
10.1 General 58
10.2 Error types 58
10.3 Error display 59
10.4 Error memory 59
10.4.1 Operating data memory and enhanced error memory 60
10.5 Bootloader 60
10.5.1 Bootloader Start Tests 61
10.5.2 Bootloader Communication 62
10.6 Configuration error 62
10.7 Emergency shutdown errors 63
10.8 Error parameter list 64

11 Parameter Description 71
11.1 General 71
11.2 List 1: Parameter 72
11.3 List 2: Measured values 78
11.4 List 3: Functions 85
11.5 List 4: Curves and Control Maps 88

12 Figures 89

13 Tables 90

14 Index 91

15 Download of Manuals 94

1 Safety instructions and the signal words and symbols used

This publication offers practical safety instructions to indicate the unavoidable residual risks involved when operating the machine. These residual risks involve hazards to

- Personnel
- Product and machine
- The environment

The primary aim of the safety instructions is to prevent personal injury!

The signal words used in this publication are specifically designed to direct your attention to possible damage extent!

DANGER *DANGER indicates a hazardous situation the consequence of which could be fatal or severe injuries if it is not prevented.*

WARNING *WARNING indicates a hazardous situation which could lead to fatal injury or severe injuries if it is not prevented.*

CAUTION *CAUTION indicates a hazardous situation which could lead to minor injuries if it is not prevented.*

NOTICE *NOTICE indicates possible material damage.*

Safety instructions are not only denoted by a signal word but also by hazard warning triangles. Red hazard warning triangles indicate immediate danger to life. Yellow hazard warning triangles indicate a possible risk to life and limb. Hazard warning triangles can contain different symbols to illustrate the danger. However, the symbol used is no substitute for the actual text of the safety instructions. The text must therefore always be read in full!

This symbol does not refer to any safety instructions but offers important notes for better understanding the functions that are being discussed. They should at any rate be observed and practiced. The respective text is printed in italics.

1.1 Safety measures under normal operation

The system may be operated by qualified and authorised personnel only, who are both familiar with the operating instructions and who can carry them out!

Before switching on the system, check and ensure that:

> only authorised personnel are in the machine's operating range

> no-one can be injured by the machine starting up

Before each start of the motor:

> Always check the system for visible damage and ensure it is not put into operation unless it is in perfect condition! Always notify the responsible department immediately about any defects

> Check and ensure that all safety devices are in proper working condition

> Remove all material and objectives surplus to requirements from the operating range of the system or motor

1.2 Safety measures for maintenance and servicing

Before starting maintenance or repair work:

> Block off access to the machine's working area for unauthorised persons! Put up an information board that indicates that such work is underway

> Switch off main switch for mains supply and secure with a padlock! The key to the padlock must be held by the person carrying out the maintenance or repair work

> Ensure that all parts that are capable of being touched have cooled down to ambient temperature and have been isolated from the mains

> Re-fasten loose connections

> Replace any damaged lines or cables immediately

> Keep the switch cabinet closed at all times! Access is solely for authorised persons with key/tools

> Never use a water spray or high-pressure cleaners on switch cabinets and other electrical equipment enclosures for cleaning purposes! Risk of short circuit and corrosion to positioning device

2 General

2.1 General System Description

The HEINZMANN positioner is used to trigger an actuator, with the setpoint being externally selected by either an analogue or a PWM input.

The heart of the control unit is a very rapid and highly powerful microprocessor (CPU). The actual control unit programme – the so-called firmware on which the microprocessor operates – is permanently stored in a Flash ROM. The application-dependent configuration is stored in an E^2PROM.

The actuator is triggered by means of a PWM signal. Both 2-quadrant (electrically single acting) and 4-quadrant (electrically double acting) actuators can be triggered.

The operating states of the positioner are supplied as analogue and digital output signals. The positioner communicates with the DcDesk 2000 parameterization and visualization tool or with a hand programmer across a serial interface.

2.2 Firmware

The control unit software is designed for universal use while providing a high functionality. Consequently the firmware contains more functions than are required for a specific application. The customer can configure the input/output assignment of the control unit and can also activate and parameterize functions.

Each control unit contains a bootloader (↑ *10.5 Bootloader*) with which the actual firmware is loaded to the unit. HEINZMANN normally ships these units with the so-called HEINZMANN basic software which contains the functionality that is available as standard.

Customized firmware variants can be created from this basic software on request.

The software version number xx.y.zz or xxxx.yy.zz in the parameter 3842 *SoftwareVersion* consists of the following parts:

Customer number	xx or xxxx
Variant	y or yy
Change index	zz.

2.2.1 HEINZMANN Basic Software

The HEINZMANN basic software has the customer number x = 0 in each unit.

It is supplied in various different basic variants y = 0..99.

The change index z = 0..99 is a running index and is incremented by variant with every new software release. Each higher index fully includes the lower index and consequently supersedes it. At any given time there is only one valid version of a basic software variant, namely the one with the highest current change index.

The following variants of the HEINZMANN basic software are currently supplied for PANDAROS positioners.

Variant	*Firmware*	**Meaning**
DC 6-07	00.0.zz	*Peripheral module with HZM CAN protocol*
DC 6-06	**00.1.zz**	***Positioner with setpoint selection*** ***via analogue or PWM input***
DC 6-12	00.2.zz	*Elektra gas measuring unit with air/fuel ratio control* *Peripheral module with HZM CAN protocol*
DC 6-13	00.3.zz	*Speed and power-dependent* *Gas positioner with misfire detection*
DC 6-15	00.4.zz	*Elektra gas measuring unit withgas flow control*
DC 6-16	00.5.zz	*Gas positioner with accelerator pedal manipulation* *(idle/final speed governor in the diesel control unit)*
DC 6-17	00.6.zz	*Elektra gas measuring unit with lambda control*
DC 6-18	*00.7.zz*	*Gas positioner with diesel injection time or diesel position control* *(variable speed governor in the diesel control unit)*

Table 1 Basic firmware variants

This brochure describes the DC 6-06 positioner with setpoint selection by analogue or PWM input

2.2.2 Custom firmware

Custom firmware always has an unique customer number x > 0. Once a customer number has been assigned to a customer it remains dedicated to him and is used for each custom software he orders, irrespective of the control unit that is used.

Different software variants y = 0..99 are created at the customer's request, e.g. for different systems or different applications with one and the same control unit.

The change index z = 0..99 is a running index and is incremented per variant with every new software release. Each higher index fully includes the lower index and consequently supersedes it. At any given time there is only one valid version of a customer software variant, namely the one with the highest current change index.

HEINZMANN communication modules such as the PC program ↑*3.3 DcDesk 2000* or a hand programmer allow the customer to access the general HEINZMANN basic software 00.y.zz and his own custom software. This means that although many customers can access the so-called 'zero' software, only one customer (and his authorised representatives) can access his own custom software. If a customer wishes to protect an application against access by other HEINZMANN customers therefore, he must ask HEINZMANN to create custom firmware for him.

2.3 Other Information

This brochure mainly describes the functions of the individual setting parameters. Trouble-shooting is described in detail.

The general operation of the PANDAROS control units, the technical specifications and connections for the control electronics, sensors, setpoint generators and actuators are comprehensively described in the following publications:

PANDAROS

Title	**Order Number**
The PANDAROS VI digital basic system	DG 03 006-d

Table 2 PANDAROS Basic Systems

The sensors available from **HEINZMANN** are described in the following publication.

Title	**Order Number**
Product Overview Sensors	E 99 001-d

Table 3 Product Overview Sensors

The method of operation of the DcDesk 2000 communication program as either a local or remote communication variant will be found in the following publications or in the programs' online help.

Title	**Order Number**
Operating instructions communication program DcDesk 2000	DG 00 003-d
Basic information remote communication program DcDesk 2000/Saturn	DG 05 008-d
Basic information remote communication program SATURN	DG 05 006-d

Table 4 Communication programs

2.4 Conventions

The following typographical conventions are used in this brochure:

1911 *ServoGain*	Parameter names are always shown in italics. No distinction is made between the four ↑ *2.5 Parameter Lists*.
↑1911 *ServoGain*	An arrow with a parameter means that the parameter is explained in detail elsewhere. A brief description of the parameter can be found in Chapter ↑ *11 Parameter Description* which also gives page references to the detailed description of the parameter.
<1911>	In drawings, numbers in angle brackets are used when the point being described corresponds to a parameter number.
↑	An arrow followed by text in italics refers to a chapter in which a function is described in detail.

2.5 Parameter Lists

A certain number of parameters must be set for each function of the firmware. These parameters are divided up into four lists:

1. Parameters — Parameters for configuring and setting (Parameter numbers 1..1999)
2. Measured values — Parameters for displaying the current states (Parameter numbers 2000..3999)
3. Functions — Parameters for activating and switching between functions (Parameter numbers 4000..5999)
4. Curves — Parameters for parameterizing curves and control maps (Parameter numbers 6000..9999)

Each parameter has a number and a name. The parameter number indicates the list to which the parameter belongs. Within the various lists, the parameters are arranged in groups to make them easy to locate.

This brochure explains all of the functions that can be executed by the PANDAROS DC 6-06 positioner. Some of these functions may be absent from special applications if they are not relevant for them. In this case the parameter that belongs to an absent function will also be omitted.

Custom applications can contain new or enhanced functions. These are described in separate brochures.

2.6 Levels

Because the control unit determines a system's operating characteristics, parameterization should be left exclusively to the system manufacturer. However to ensure that the benefits of the digital device can be enjoyed right down to the end customer, the parameters of the **HEINZMANN** control unit are divided into seven levels.

- **Level 1: Level for the end customer**
  On this level, the most important operating values (e.g. setpoints and actual values for filling) and errors can be displayed. The configuration data cannot be changed.
- **Level 2 and 3: Level for the device manufacturer**
  The device manufacturer can activate the actuator test mode.
- **Level 4: Level for the system manufacturer's service department**
  All the parameters for adjusting the system's operating characteristics are available on this level.

- **Level 5 and 6: Level for the system manufacturer**
  Intervention in the control unit functions is possible on this level.
- **Level 7: Level for development**
  This level is reserved for use by HEINZMANN's development department.

As this list shows, higher levels automatically incorporate all lower levels. The particular level of a parameter is listed in Chapter ↑ *11 Parameter Description*. The maximum possible level, i.e. the choice / number of parameters which can be accessed, is determined by the diagnostic tester that is used, and cannot be changed. It is possible however to move down from the currently valid level by means of a menu option in the PC program or by using the parameter ↑ 1800 *Level* and so to reduce the number of currently visible parameters and functions.

2.7 Block Diagram

The block diagram shows a simplified schematic of the control configuration of HEINZMANN positioners. It shows the basic functions and the signal paths of major functions.

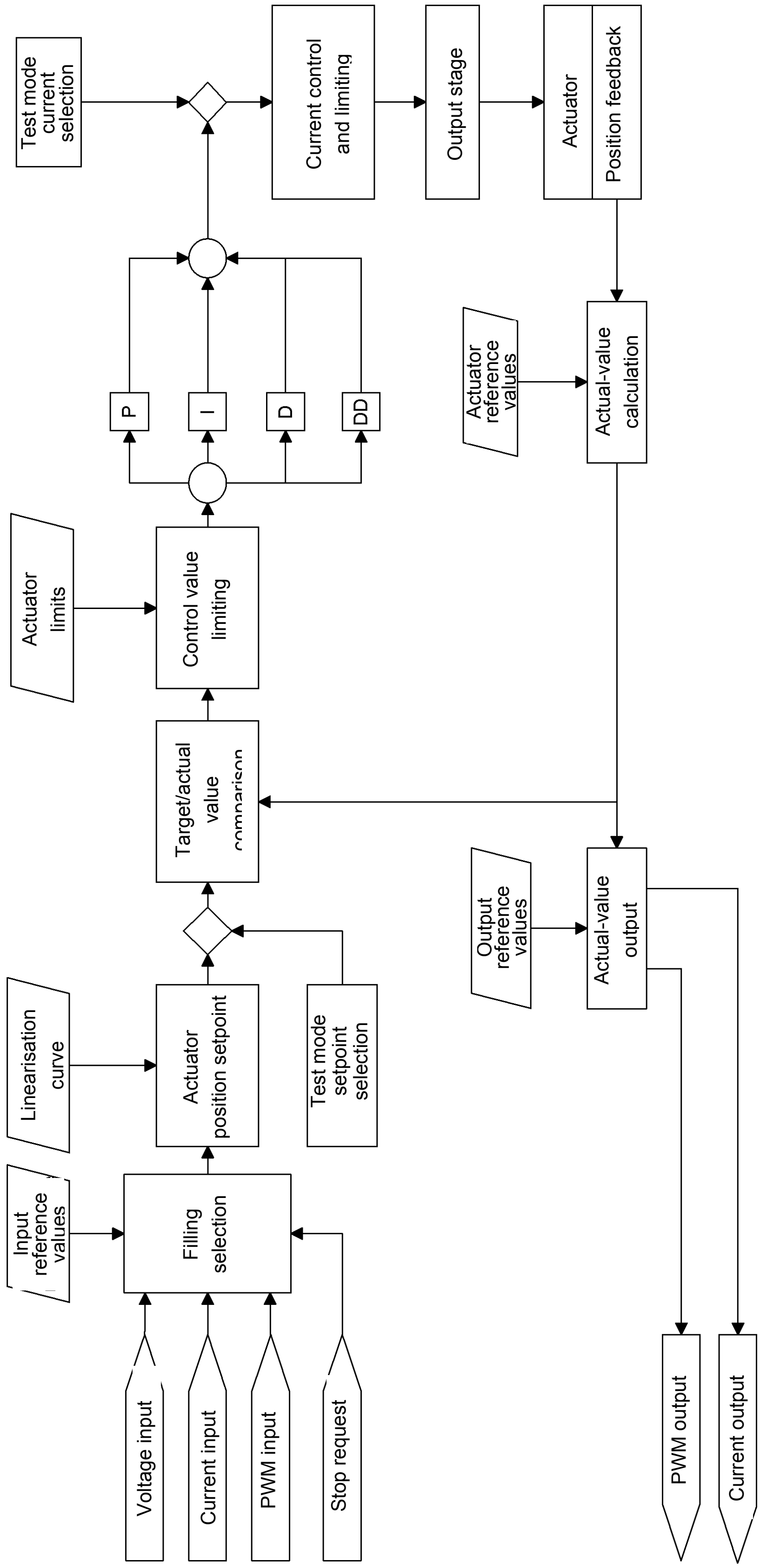

Fig. 1 Block Diagram

3 Parameterization of the HEINZMANN control units

The following chapters describe the functions of the **HEINZMANN** control units and how they are set. Some functions only act together with others, or are influenced by other functions. When functions like these are parameterized and optimized, it is often advisable to deactivate other functions so that only the effect of the required function is seen. The setting of these functions is described in the individual chapters.

3.1 Parameterization options

There are a number ways in which **HEINZMANN** control units can be parameterized. For experimental work and first commissioning HEINZMANN recommends the use of ↑ *3.3 DcDesk 2000* as a diagnostic and parameterization tool. DcDesk 2000 can also be used for service purposes, although the hand programmer units PG 02 and HP 03 are also available. The DcDesk 2000/Saturn remote communication option also provides considerable assistance for servicing.

The following list gives an overview of all the parameterization options that are available.

- **Parameterization by HEINZMANN**
  Control unit operation is tested with a test routine during final inspection at the factory.
- **Parameterization with the hand programmer**
  Complete parameterization can be carried out with the hand programmer PG 02 or HP 03 depending on the level. This handy device is suitable for use mainly by service personnel.
- **Parameterization with DcDesk 2000 or DcDesk 2000/Saturn**
  Depending on the level, a number of parameters can be continuously displayed and edited with a DcDesk 2000 PC program. The PC program also allows the graphic display and simple setting of curves. Control unit data can be stored on the PC or uploaded from the PC back to the control unit. Another advantage of the PC program is the high resolution visualisation of measured values against time or against one another.
- **Transferring records**
  Once the parameterization for an application has been finalised, the data record can be stored in the hand programmer or on the PC. If there are other applications of the same type, these data records can be transferred to the new control units.
- **End-of-line parameterization**
  This option is used for test bench runs by the system manufacturer. The control unit is adapted to the requirements specified in the order. Both the control unit firmware and a delivery data record can be programmed during the end-of-line run with a command line call from the DcDesk 2000; operator action is not required.

3.2 Saving the data

Basically the communication programs/devices that have been referred to only change the parameters in the volatile memory of the control unit. Although the control unit will immediately operate with the new settings, any values changed in this way will be lost when the voltage supply is turned off. A storage command must be used to permanently save the parameter settings in the control unit. DcDesk 2000 uses the function key F6 for this, while the hand programmers work with the “Save Parameter” key or menu option. It is this operation that is meant whenever there is a reference to the need to save parameters in the following descriptions.

3.3 DcDesk 2000

The **HEINZMANN** DcDesk 2000 PC program is used to set up and distribute operating data in all digital **HEINZMANN** systems, in particular the systems described here.

The PC and the control unit are connected by a serial interface. A remote communication variant allows “extended” access over the Internet, Intranet or by direct modem connection.

As a Windows® program it offers all of the numerical and graphic features that are needed for experimental work, first commissioning and servicing, and facilitates the associated documentation work.

DcDesk 2000 also allows the user to print off screen displays and data recordings. The data are stored in a standard text format for the purposes of further processing, importing into reports etc.

The data record of a connected control unit can be edited and the response to parameter changes can be observed at the same time. A parameter set can also be edited and recorded data can be evaluated without a control unit. A parameter set that is created in this way can be transferred to the control unit later on.

All settings can be made directly by accessing parameter numbers. However there are also additional windows which can significantly simplify special functions, especially configuring the system and parameterizing curves.

Current measured values are displayed numerically and/or graphically. There is a separate window in which up to ten freely selectable measured values can be displayed simultaneously as a function of time. There is another window in which nine measured values can be displayed as a function of a tenth. Data recordings can also be logged for analysis and printing at a later time.

All of the curves that are available in the control unit can be shown in separate windows two-dimensionally. This can provide an immediate impression of the curve profile. The current point where the system is on the curve is displayed online. A knowledge of the relationship between the parameter number and the point on the curve is not necessary for setting purposes, as there is a special input section. Here all the particular aspects of curve parameterization are taken into consideration, so input errors cannot occur.

DcDesk 2000 is being continuously developed and enhanced with additional functions.

HEINZMANN strongly recommends the use of DcDesk 2000 for experimental work and first commissioning. The program is also of major benefit to service engineers conducting diagnosis and trouble-shooting.

3.4 Value range of parameters

Each parameter is assigned a particular value range. The large number of parameters and functions means that there is also a large number of value ranges. The value range for each parameter is listed in Chapter ↑ *11 Parameter Description*. The value range of the parameters is also displayed by the PC or hand programmer (↑ *3.1 Parameterization options*).

Some parameters have a value range that can have only two states: 0 or 1. These parameters are used to activate or toggle individual functions or to display the status of errors, external switches etc. Parameters with this value range can only occur in List 2 and List 3 (↑*11.3 List 2: Measured values* and ↑*11.4 List 3: Functions*).

State "1" means that the function is active or that the error is present, whereas with state "0" the function is inactive or the error is not present.

With toggles or parameters that choose between two functions, the parameter name always contains an "*Or*" (example: 4330 *FuelSetpPWM**Or**Analog*). The function before the "*Or*" is active when the parameter value corresponds to the maximum, the function after the "*Or*" when the parameter value corresponds to the minimum.

3.5 Activating functions

A function can be activated in one of two ways:

- **Parameters**
  Parameters on List 3 (↑*11.4 List 3: Functions*) activate functions which the user selects and they are then always active.
- **Switch functions**
  External switches (↑*6.1 Switch functions via digital inputs*) can tell the control unit about desired operating states which change frequently during continuous operation, such as a stop request, or which are to be activated by external switches (e.g. resetting errors). The state of the switch functions can be seen from the parameters starting from number 2810.

3.6 Parameterization examples

A parameterization example exists for most functions. This example contains all the parameters that are needed for the described function. However the settings that are given may vary depending on the application and should only be taken as examples. Sensible settings which are appropriate for the specific application should therefore be used when setting a function.

3.7 Resetting a control unit

A reset involves resetting and restarting the control unit. This can be achieved by briefly turning off the power supply or by using a special command from DcDesk 2000 or the hand programmer HP 03.

Note

A reset loses all the data which were not stored in the control unit's read-only memory. It is absolutely essential therefore to save the data to the control unit's read-only memory before a reset if you wish to retain the data.

Certain parameters or functions of control units only become active after a reset. These are mainly functions which can put the control unit in a different operating state, or parameters which must not be changed during continuous operation for safety reasons.

WARNING

Positioning device shaft may start up unexpectedly!
Control unit is briefly non-operational during a reset.

Risk of injury through moving elements

> A reset may only be carried out when the application is stopped!

4 Setpoint

The **HEINZMANN** control units distinguish between analogue or PWM inputs and setpoints or sensors respectively. This means that the application control is determined by the current value of setpoints or sensors but it is configured separately, which is how they obtain their value.

4.1 Design of the setpoint generator

The setpoint generators supply either an analogue current or voltage signal or a PWM signal (↑ *6.2 Setpoint selection and stop request via analogue inputs* and ↑ *6.3 Setpoint selection by PWM input*).

The choice of setpoint generator as an analogue or PWM value is made by parameter 4330 *FuelSetpPWMOrAnalog*. One of the following values must be entered in it:

Channel type	Setpoint source
0	Analogue signal (current or voltage)
1	PWM signal

Table 5: Setpoint sources

4.1.1 Analogue setpoint

The analogue signal is connected to Pin 7. The analogue value is defined as a current or voltage value in parameter 5530 *FuelSetpCurrOrVolt*:

Signal type	Signal range
1	Voltage
2	Current

Table 6: Setpoint type

The referencing of the analogue input and the definition of the error limits is described in ↑ *6.2 Setpoint selection and stop request via analogue inputs*.

Terminal connection diagrams with an analogue setpoint can be seen in

↑ *7.3.1 Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor*

↑ *7.3.2 Analogue setpoint, stop signal digital and/or analogue, gearless actuator*

4.1.2 PWM Setpoint

Two inputs are available for the PWM setpoint. The choice depends on the use and on the type of stop signal. If no stop signal is connected or if the stop is transmitted to the control unit with a digital signal (↑*5.2.1.1 External stop request by a digital input*), then PWM input 1 at Pin 2 and 3 is automatically used.

If the stop has to be received via an analogue input however (↑*5.2.1.2 External stop request by an analogue input*), then Pin 11 and 12 at PWM input 3 are automatically available for the PWM setpoint.

If an analogue setpoint is requested and not a PWM setpoint, then Pin 11 is basically used as digital input 5.

The configuration of the PWM inputs is described in detail in ↑*6.3 Setpoint selection by PWM input*.

The following drawings illustrate the terminal connections

↑*7.3.3 PWM setpoint, stop signal digital, actuator with disc rotor motor*

↑*7.3.4 PWM setpoint, stop signal digital, gearless actuator*

↑*7.3.5 PWM setpoint, stop signal analogue, actuator with disc rotor motor*

↑*7.3.6 PWM setpoint, stop signal analogue, gearless actuator.*

4.2 Determining the actuator setpoint

The setpoint generator connected to an analogue or PWM input determines the value of parameter 2350 *FuelSetpoint*. This parameter always has the value range 0..100 % (↑*6.2 Setpoint selection and stop request via analogue inputs* or ↑*6.3 Setpoint selection by PWM input*).

If the externally connected analogue setpoint generator supplies an inverted signal, this must be taken into consideration when configuring the analogue input. The voltage and current values for 0 and 100 % of the setpoint must therefore always be entered in reference values 1530 *FuelSetAna_Ref_0%* and 1531 *FuelSetAna_Ref_100%* whether or not the physical values are rising or falling (↑*6.2 Setpoint selection and stop request via analogue inputs*).

The actual actuator setpoint 2330 *ActPosSetpoint* is determined from Parameter 2350 *FuelSetpoint*. If the full actuator stroke of 0..100 % is not going to be used, a curve can be interposed which determines the actuator setpoint from the current filling setpoint. If no correction curve is used, 2350 *FuelSetpoint* and 2330 *ActPosSetpoint* are identical.

7300 *FuelToActSetp:f*	Set filling values of the curve
7315 *FuelToActSp:Pos*	Actuator setpoints of the curve
4721 *FuelToActPosCurveOn*	1 = Activation of the curve

The correction curve must be parameterized as strictly rising, which means that both x and y values must increase as the index increases. Inverting filling setpoint over this path is not permitted. This parameterization is checked by the control unit and in case of error the configuration error 3000 *ConfigurationError* = 40 is output. The curve is ignored in this case.

The actuator setpoint is used on the control loop to trigger the actuator (*↑ 8.3.1 Servo control loop*).

5 System monitoring

5.1 Operation mode

When the control unit has started (booting routine) it automatically assumes that the system is stopped. 3830 *Phase* indicates the value 0.

The system is defined as running when there is no stop request present and a setpoint 2350 *FuelSetpoint* greater than 0 % is selected. This state is indicated by 3802 *SystemStopRequest* = 0 and 3803 *SystemStopped* = 0. 3830 *Phase* indicates the operation phase with the value 4.

For safety reasons neither automatic actuator calibration nor the actuator test mode is possible during operation. The control unit will also reject attempts to load firmware and an automatic system reset by the PC program.

5.2 Stopped state

When the control unit has started (booting routine) it automatically assumes that the system is stopped. 3830 *Phase* indicates the value 0.

Following an operation phase, the system is detected as being stopped again if either a stop request is present for 5 s (↑*5.2.1 Stop request*) or the setpoint generator 2350 *FuelSetpoint* selects the setpoint 0 % for 5 s.

During the 5 s stop request, a 4Q-actuator is positioned by force onto the lower limit stop, whereas without a stop request and with a setpoint selection of 0 % it is controlled to the lower limit stop. 2Q-actuators are always pulled to 0 by a strong spring.

After the 5 s, 3803 *SystemStopped* changes from 0 to 1 and 3830 *Phase* from 4 to 0. Both indicate the active stopped state. The energization of the actuator can also be switched off after this. This is achieved by entering a seconds value greater than 0 in parameter 1915 *ServoCurrentOffDelay*. After the 5 s referred to above and the delay in this parameter, the energization is switched off. If the parameter value is on 0 s, then the actuator is continuously energized.

Consequently the system is “stopped” when parameter 3803 *SystemStopped* shows the value 1. The system must be stopped before automatic actuator calibration can be performed or the actuator test mode can be activated. Similarly, a new firmware download and an automatic control unit reset by PC or hand programmer is only possible in this state.

The system runs in 3830 *Phase* = 8 during automatic actuator calibration, and in 3830 *Phase* = 9 during the actuator test. Both phases are quit immediately when a setpoint 2350 *FuelSetpoint* > 0 % is selected and there is no stop request present.

5.2.1 Stop request

A stop request is present when 3802 *SystemStopRequest* = 1. This means that either an externally selected stop signal is active or a fatal system error has occurred (↑ *10.7 Emergency shutdown errors*).

An external stop signal is selected by 2810 *SwitchStopRequest* (↑ *5.2.1.1 External stop request by a digital input*) or 2809 *AnaStopRequest* (↑ *5.2.1.2 External stop request by an analogue input*). Both signals are of equal value in the program flow: If one of the two values is present, then the stop request 3802 *SystemStopRequest* is active, if both values are inactive, then 3802 *SystemStopRequest* is also deactivated unless a fatal error is present.

5.2.1.1 External stop request by a digital input

A digital stop request is indicated by 2810 *SwitchStopRequest* = 1. The assignment of a certain digital input to this function is described in detail in ↑ *6.1.1 Assignment of digital inputs*.

This stop request can be executed as high-active or low-active by the software. High-active means that the stop request is only active when the signal = 1. Low-active means that the stop request is activated when the signal = 0. The desired signal type is defined with 4811 *DigStopOpenOrClose*.

4811 *DigStopOpenOrClose* = 0	Stop request is only active when the switch is closed
4811 *DigStopOpenOrClose* = 1	Stop request is only active when the switch is open

Parameter 4810 *DigStopImplsOrSwitch* can also be used to select whether the stop will only last for as long as the request itself is active, or whether a one-off switch impulse is sufficient to activate the stop. In the latter case the stop request is only terminated when the system is stopped.

4810 *DigStopImplsOrSwitch* = 0	Stop request is only active when stop command is present
4810 *DigStopImplsOrSwitch* = 1	A one-off switch impulse keeps the stop request active until the system stops

Digital inputs are not monitored for errors and must not be used for safety-related functions as defined by classification societies

> The use of a monitored analogue input instead of a digital input for the external stop request is therefore possible.

The terminal connection diagrams

↑ *7.3.1 Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor*

↑ *7.3.2 Analogue setpoint, stop signal digital and/or analogue, gearless actuator*

↑ *7.3.3 PWM setpoint, stop signal digital, actuator with disc rotor motor*

↑ *7.3.4 PWM setpoint, stop signal digital, gearless actuator*

operate with a digital stop signal.

5.2.1.2 External stop request by an analogue input

To be able to meet demands for the stop input to be monitored, the stop request can be also be selected by a monitored analogue input instead of by a digital input. Parameter 4340 *AnaStopRequestUsed* must be set to 1.

ADC 1 on 2 is the used automatically as an analogue input. The input can be used as a current or voltage input.

5510 *AnaStopCurrOrVolt* = 1	Voltage input 0..5 V
5510 *AnaStopCurrOrVolt* = 2	Current input 0..22 mA

It should be remembered that only a part of the range can be used for the signal, e.g. 4..20 mA, so as to be able to define error limits. Parameterization of the analogue input is described in detail in ↑ *6.2 Setpoint selection and stop request via analogue inputs*.

The measured value of the analogue stop request is displayed in the range 0..100 % in 2900 *AnaStopSensor*. The two thresholds 340 *AnaStopThresholdLow* and 341 *AnaStopThresholdHigh* are used to generate the actual stop signal 2809 *AnaStopRequest*:

2900 *AnaStopSensor* <= 340 *AnaStopThresholdLow* → 2809 *AnaStopRequest* = 0

2900 *AnaStopSensor* >= 341 *AnaStopThresholdHigh* → 2809 *AnaStopRequest* = 1.

Note

If the analogue stop request is used, a PWM selection of the setpoint (if any) must be connected to PWM input 3 (Pins 11 and 12). If the analogue stop request is not used on the other hand, the PWM setpoint is automatically expected at PWM input 1 (Pins 2 and 3). This does not affect the analogue setpoint selection.

The following terminal connection diagrams contain an analogue stop signal:

↑ *7.3.1 Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor*

↑ *7.3.2 Analogue setpoint, stop signal digital and/or analogue, gearless actuator*

↑ *7.3.5 PWM setpoint, stop signal analogue, actuator with disc rotor motor*

↑ *7.3.6 PWM setpoint, stop signal analogue, gearless actuator.*

5.3 Setpoint monitoring

Depending on the configuration in 3007 *ErrFuelSetpAna*, 3009 *ErrFuelSetpPWMIn1* or 3011 *ErrFuelSetpPWMIn3*, a setpoint error is displayed and the common alarm 3801 *CommonAlarm* is activated (↑ *6.6 Common alarm via alarm output*).

In the configuration phase you can decide how the system should respond to such an error.

4331 *FuelSetpSubstOrLast* = 0	continue working with the last valid percentage
4331 *FuelSetpSubstOrLast* = 1	continue working with 331 *FuelSetpSubst*
4332 *FuelSetpHoldOrReset* = 0	when the measured value is back within the permitted limits, reset the error and continue with the new valid value
4332 *FuelSetpHoldOrReset* = 1	the error remains active until the control unit is switched off or until an error reset, whether or not the measured value returns to within a valid range

5.3.1 Monitoring the analogue setpoint

If the measured value of the analogue setpoint generator 3531 *FuelSetpAna_Value* exceeds or falls below the defined error limits 1532 *FuelSetAna_ErrorLow* or 1533 *FuelSetAna_ErrorHigh* (↑ *6.2.3 Error detection with the analogue inputs*), the error 3007 *ErrFuelSetpAna* is set. The common alarm 3801 *CommonAlarm* is activated at the same time.

This error is automatically reset when the signal is back within the valid range and the reset is permitted by 4332 *FuelSetpHoldOrReset* = 0.

Even if the value returns to within a valid range, the error remains continuously present if the 'freeze' is requested with 4332 *FuelSetpHoldOrReset* = 1.

5.3.2 Monitoring the PWM setpoint

If the setpoint is connected to PWM input 1, a possible error in 3009 *ErrFuelSetpPWMIn1* is activated, when connected to PWM input 3 on the other hand the error 3011 *ErrFuelSetpPWMIn3* is set. As with every other error, the common alarm 3801 *CommonAlarm* is simultaneously activated.

An error at the PWM setpoint generator is detected when signal 3500 *FuelSetpPWMIn1* or 3504 *FuelSetpPWMIn3* fails or when the frequency at 3501 *FuelSetpFrequencyIn1* or 3505 *FuelSetpFrequencyIn3* rises above 625 Hz or when a high-phase is measured below the mean between 0 % and the lower reference or above the mean between the upper reference and 100 % (↑*6.3.2 Error detection with the PWM inputs*).

The particular error is automatically reset when the signal is back within the valid range and the reset is permitted by 4332 *FuelSetpHoldOrReset* = 0.

Even if the value returns to within a valid range, the error remains continuously present if the 'freeze' is requested with 4332 *FuelSetpHoldOrReset* = 1.

5.4 Monitoring the analogue stop signal

If the measured value of the analogue stop signal 3511 *AnaStop_Value* exceeds or falls below the specified error limits 1512 *AnaStop_ErrorLow* and 1513 *AnaStop_ErrorHigh* (↑*6.2.3 Error detection with the analogue inputs*), the error 3005 *ErrAnaStopRequest* and the common alarm 3801 *CommonAlarm* are activated simultaneously (↑ *6.6 Common alarm via alarm output*).

As with the setpoint input, you can decide in the parameterization phase how the system will respond to such an error.

4341 *AnaStopSubstOrLast* = 0	continue working with the last valid percentage
4341 *AnaStopSubstOrLast* = 1	continue working with 342 *AnaStopSensorSubst*
4342 *AnaStopHoldOrReset* = 0	when the measured value is back within the permitted limits, reset the error and continue with the valid value
4342 *AnaStopHoldOrReset* = 1	the error remains active until the control unit is switched off or until an error reset, whether or not the measured value comes back to within a valid range

It should be remembered that error handling manipulates the percentage 2900 *AnaStopSensor* and that it therefore also indirectly affects 2809 *AnaStopRequest* - via the thresholds, but does not directly affect the stop request.

An error is therefore automatically reset when the signal is back within the valid range and the reset is permitted by 4849 *AnaStopHoldOrReset* = 0.

However even if the value returns to within a valid range, the error remains continuously present if the 'freeze' is requested with4849 *AnaStopHoldOrReset* = 1.

With these parameterization options, the appropriate error handling can be found for any system.

5.5 Electronics monitoring

Electronics self-tests are performed to monitor the safe operating state. The following table shows what is monitored and which errors are set in each case. The tests that are run once when the control unit is booted are described in ↑ *10.5.1 Bootloader Start Tests*. ↑*10.7 Emergency shutdown errors* shows which errors lead to an emergency shutdown of the system or which disable a system start. Each individual error is described in detail in ↑*10.8 Error parameter list*.

Error	**Cause**
3075 *ErrClearFlash*	Error when clearing the flash memory (display in the bootloader)
3076 *ErrParamStore*	Error when storing parameters
3077 *ErrProgramTest*	Error in the ongoing test of the program memory
3078 *ErrRAMTest*	Error in the ongoing test of the RAM memory
3081 *Err5V_Ref*	Error at voltage reference values
3085 *ErrVoltage*	Operating voltage too high or too low
3089 *ErrWatchdog*	Undefined program flow, internal program error (display in the bootloader)
3090 *ErrData*	No parameters available, or checksum of parameters is incorrect (always active after a program download)
3093 *ErrStack*	Stack overflow, internal program error
3094 *ErrIntern*	Exception, internal program error

5.5.1 Voltage references

The PANDAROS DC 6-06 uses a voltage reference value for ratiometric measurement at analogue inputs. The value must be within software-internal fixed limits or the error 3081 *Err5V_Ref* is output and the setpoint input cannot be corrected.

5.5.2 RAM memory test

All of the used RAM memory is tested while the program is running. The address of the cell that is being currently tested is shown in 3895 *RAMTestAddr*. The test value that is currently running can be taken from 3896 *RAMTestPattern*. If a fault cell is detected, these two displays are held, the error 3078 *ErrRAMTest* is activated and a fatal error is generated.

5.5.3 Program memory test

The program memory is tested while the program is running. The checksum is gradually computed over the entire program memory and compared with the stored checksum. If the checksums do not match, error 3077 *ErrProgramTest* is displayed and a fatal error generated.

5.5.4 Stack Depth Test

During ongoing program execution, a so-called stack memory is needed to execute subroutines and interrupt-service routines. The utilisation of this memory is continuously monitored, and error message 3093 *ErrStack*is output if it would run too deep. At the same time a fatal error is generated because the program flow is no longer protected.

5.5.5 Program flow test

While the program is running, a test is executed to check whether the software is running through valid memory areas. If it is not, the exception error 3094 *ErrIntern* is displayed and the system is stopped. HEINZMANN can draw inferences about the program error from the displayed values starting from 3195 *ExceptionNumber*.

The amount of calculation time which the running program needs can be taken from the displayed value 3865 *CalculationTime*. The value 3870 *Timer* is a rolling millisecond display that is used internally for time-controlled functions and externally to influence the graphic display in DcDesk 2000.

5.5.6 Operating voltage monitoring

The operating voltage 3600 *PowerSupply* is monitored by every control unit. PANDAROS type control units can tolerate a battery power dip for a certain length of time before setting an error.

Normally these control units go into reset when the voltage dips below 9 V. If the function 5600 *LowPowerEnable* is activated and with suitable hardware, the control unit will accept the power falling below a limit of 8.5 V for 20 s and even below a limit of 7 V for 1 s according to the 12 V battery standard. Subsequently the voltage must be above 9 V for twice the length of time it was below 9 V before the low power can be enabled again.

If the low power enable function is not activated or if the unfiltered voltage 3602 *PowerSupplyRaw* dips for longer than permitted, then error 3085 *ErrVoltage* is set.

Note

Activation/deactivation of function 5600 LowPowerEnable is only accepted after a control unit reset. Whether the control unit hardware that

is being used can actually execute the function is indicated in 3601 LowPowerEnabled.

6 Inputs and outputs

This section describes the inputs and outputs of the control unit. For inputs/outputs whose type is configured, please refer to the chapters indicated.

The terminal for the particular signal is underscored and shown in bold print.

6.1 Switch functions via digital inputs

One fixed digital input is available. The system automatically configures a second input as a digital input unless a PWM setpoint selection and an analogue stop request are simultaneously provided, i.e. either 4330 *FuelSetpPWMOrAnalog* = 0 or 4340 *AnaStopRequestUsed* = 0.

Inputs	Use	Name	Terminal
Digital input 4	unassigned	SpD	**9**, 21
Digital input 5*	Digital input unless PWM setpoint and analogue stop are simultaneously requested	Stp	**11**, 21

* optional PWM input(↑*4.1.2 PWM Setpoint*)

Table 7: Digital Inputs

In the **HEINZMANN** control units a distinction is made between external switches (digital inputs) and internal switch functions. This means that although the system control is determined by the current value of switch functions, the actual configuration which gives these switch functions their value takes place separately.

For every switch function there is a display parameter which indicates whether the function is activated. A "1" always means that the function is active, whereas a "0" means it is inactive. This display is independent of the hardware configuration of the switches (high side/low side).

The switch functions used in the PANDAROS DC 6-06 are on/off switches. The name of the switch function corresponds to the meaning *On* or *Active*, i.e. 2810 *SwitchStopRequest* = 1 means that a stop request is present, for example. The state "1" always defines *On* and the state "0" stands for *Off* or *Inactive*.

The following table gives an overview of the existing switch functions. Explanations of the individual functions and switch priorities will be found in the corresponding chapters of the function descriptions.

Switch function	Meaning
2810 *SwitchStopRequest*	0 = no stop request via digital input 1 = stop request via digital input
2828 *SwitchErrorReset*	0→1 = clear current error (only with edge change)
2845 *SwitchAutoAdjust*	0→1 = automatic actuator adjustment (only with edge change)

Table 8: Switch functions

6.1.1 Assignment of digital inputs

A digital input can be assigned to a switch function by entering the number of the digital input in the corresponding assignment parameter of the function from 810 *Funct....*

An assignment of 0 means that the switch function is not being used by a digital input. Such a switch function always has a value of 0 and is therefore always inactive.

A maximum of two digital inputs are available in the PANDAROS DC 6-06: Input 4 on terminal 9 and input 5 on terminal 11. Consequently a maximum of two of the possible switch functions can actually be used.

Switch function	Input	Remarks
810 *FunctStopRequest*	0	not used, but analogue stop request is possible
	5	only possible when PWM setpoint and analogue stop request are not simultaneously requested
828 *FunctErrorReset*	0	not used, but error can be cleared with the PC program or hand programmer
	4	always possible
	5	only possible when no digital stop is requested
845 *FunctAutoAdjust*	0	not used, but auto adjust is possible with the PC program or hand programmer or pressure switch
	4	always possible
	5	only possible when no digital stop is requested

Table 9: Switch function assignment

Note

A switch pulse must be at least 20 ms long to be detected by the control electronics.

CAUTION

Digital inputs are not monitored for errors and must not be used for safety-related functions as defined by classification societies

> Therefore it is recommended to use a monitored analogue input instead of a digital input for the external stop request (↑ 5.2.1.2 External stop request by an analogue input).

6.2 Setpoint selection and stop request via analogue inputs

Analogue input 3 is permanently assigned by the system for the connection of an analogue setpoint generator 4330 *FuelSetpPWMOrAnalog* = 0. The input stays open when a PWM setpoint is used 4330 *FuelSetpPWMOrAnalog* = 1.

If an analogue stop request is necessary (4847 *AnaStopRequestUsed* = 1), analogue input 1 must always be used because it is permanently assigned by the system.

Inputs	Use	Name	Terminal	Range
Analogue input 1*	Analogue stop request	P1	2, 3, 6	0..5 V or 0..22 mA
Analogue input 3	Analogue setpoint	SpA	6, 7, 8	0..5 V or 0..22 mA

* optional PWM input(↑*4.1.2 PWM Setpoint*)

Table 10: Analogue inputs

6.2.1 Adjusting the current/voltage inputs

Each of the current/voltage inputs has a lower (1510 *AnaStop_Ref_0%* or 1530 *FuelSetAna_Ref_0%*) and an upper reference value (1511 *AnaStop_Ref_100%* or 1531 *FuelSetAna_Ref_100%*). These two values in this order basically correspond to the 0 % and 100 % value of the sensor/setpoint generator irrespective of which current or voltage is to be externally applied. The current unfiltered value of the analogue input is displayed in 3511 *AnaStop_Value* or 3531 *FuelSetpAna_Value* and the filtered value in 3510 *AnaStop_Percent* or 3530 *FuelSetpAna_Percent*. The filtered values correspond directly to the displayed values in 2900 *AnaStopSensor* or 2350 *FuelSetpoint*.

Parameterization example

The setpoint generator is connected to analogue input 3 as a current signal. It has an inverter measurement range of 20 to 4 mA corresponding to 0 to 100 %. Parameter 3531 *FuelSetpAna_Value* shows the current measured value in mA and Parameter 3530 *FuelSetpAna_Percent* shows the measured value as a percentage. The resulting 2350 *FuelSetpoint* takes error handling into account.

Number	Parameter	Value	Unit
1530	*FuelSetAna_Ref_0%*	*20*	*mA*
1531	*FuelSetAna_Ref_100%*	*4*	*mA*
4330	*FuelSetpPWMOrAnalog*	*0*	

6.2.2 Filtering the analogue inputs

The measured value of the analogue input can be filtered with a digital filter. The relevant parameters are on numbers 1514 *AnaStop_Filter* or 1534 *FuelSetAna_Filter*.

A filter value between 1 and 255 is entered in these parameters. A value of 1 means no filtering. The following formula applies

$$\tau = \frac{Filter\_value}{62.5} \text{ [s].}$$

A filter value of 8 must be used for normally fast sensor changes. The time constant for filtering should be approximately the same as the time constant of the setpoint generator/sensor.

Parameterization example:

Number	Parameter	Value	Unit
1534	*FuelSetAna_Filter*	*8*	

Time constant

$$\tau = \frac{8}{62.5} \text{ [s]} = 0.128 \text{ s}$$

6.2.3 Error detection with the analogue inputs

If a setpoint generator/sensor fails (e.g. due to short or open circuit), the control unit measures voltages or currents that are outside the normal measurement range. These measured values outside the normal measurement range can be defined as an inadmissible operating range in which the control unit detects a failure of the sensor.

Like the reference values, the error limits are indicated in the electrical unit.

The parameters 1512 *AnaStop_ErrorLow* or 1532 *FuelSetAna_ErrorLow* determine the lower error limits that define when the minimum permissible value is undershot.

The parameters 1513 *AnaStop_ErrorHigh* or 1533 *FuelSetAna_ErrorHigh* determine the upper error limits that define when the maximum permissible value is exceeded.

It makes no difference whether the minimum or maximum value is assigned to the 0 % or the 100 % value of the setpoint generator/sensor for referencing, only the value is of interest, i.e. 0.5 V and 4.5 V for example.

Parameterization example:

The setpoint generator at analogue input 3 normally supplies a measured value between 4 and 20 mA. These values are undershot or exceeded when a short or open circuit occurs. The range below a measured value of 2.5 mA and above 21.5 mA is defined as inadmissible by the following parameters:

Number	Parameter	Value	Unit
1530	*FuelSetAna_Ref_0%*	*20*	*mA*
1531	*FuelSetAna_Ref_100%*	*4*	*mA*
1532	*FuelSetAna_ErrorLow*	*2.5*	*mA*
1533	*FuelSetAna_ErrorHigh*	*21.5*	*mA*
4330	*FuelSetpPWMOrAnalog*	*0*	

The error limits should not be too close to the minimum or maximum value to prevent natural measured value fluctuations in the sensor resulting in an error detection. Nevertheless a short or open circuit must be positively detected.

When an error is detected, the corresponding error parameter of the associated sensor is set. The response to this error is described in Chapter ↑*10.8 Error parameter list*. If an analogue input is not used, i.e. it is not assigned to any sensor, it is also not monitored for errors.

6.3 Setpoint selection by PWM input

PWM input 1 is used for PWM setpoint selection if an analogue stop is not used, and PWM input 3 if an analogue stop request has to be executed.

Inputs	Use	Name	Terminal	Maximum frequency
PWM input 1*	PWM setpoint if no analogue stop request exists	P1	**2**, 3	500 Hz
PWM input 3+	PWM setpoint if an analogue stop request exists	Stp	**11**, 12	500 Hz

* optional analogue input (↑*5.2.1.2 External stop request by an analogue input*)

+ optional digital input (↑*4.1.2 PWM Setpoint*)

Table 11: PWM Inputs

The PWM signal is usually transmitted with a range of 5 to 95 % PWM. To scale the measurement ranges, the lower reference values must be entered in parameters 1500 *FuelSetPWMIn1RefLow* or 1504 *FuelSetPWMIn3RefLow* and the upper reference values in parameters 1501 *FuelSetPWMIn1RefHigh* or 1505 *FuelSetPWMIn3RefHigh*.

The measured value parameters 3500 *FuelSetpPWMIn1* or 3504 *FuelSetpPWMIn3* show the PWM ratio as a percentage, and the measured value parameters 3501 *FuelSetpFrequencyIn1* or 3505 *FuelSetpFrequencyIn3* show the PWM frequency.

The selection as a PWM setpoint is made according to ↑ *4.1 Design of the setpoint generator*.

6.3.1 Filtering the PWM inputs

The measured value of the particular PWM input can be filtered with a digital filter. The associated parameters are on numbers 1506 *FuelSetPWMIn1_Filter* or 1508 *FuelSetPWMIn3_Filter*. The inputs are filtered exactly as described in ↑*6.2.2 Filtering the analogue inputs*.

Parameterization example 1:

The setpoint generator selects the filling setpoint with a PWM ratio of between 5 % and 95 %. The analogue stop request is not needed, so the PWM setpoint is automatically present on Pin 2 and 3 at PWM input 1.

Number	Parameter	Value	Unit
1500	*FuelSetPWMIn1RefLow*	*5*	*%*
1501	*FuelSetPWMIn1RefHigh*	*95*	*%*
1506	*FuelSetPWMIn1_Filter*	*4*	
4330	*FuelSetpPWMOrAnalog*	*1*	
4340	*AnaStopRequestUsed*	*0*	

Parameterization example 2:

The setpoint generator selects the filling setpoint with a PWM ratio of between 10 % and 90 %. The analogue stop request is needed, so the PWM setpoint is automatically expected at PWM input 3 on Pins 11 and 12.

Number	Parameter	Value	Unit
1504	*FuelSetPWMIn3RefLow*	*10*	*%*
1505	*FuelSetPWMIn3RefHigh*	*90*	*%*
1508	*FuelSetPWMIn3_Filter*	*8*	
4330	*FuelSetpPWMOrAnalog*	*1*	
4340	*AnaStopRequestUsed*	*1*	

6.3.2 Error detection with the PWM inputs

The following error causes are detected at PWM inputs and displayed as an error on the assigned sensor:

- The PWM signal has failed
- The frequency is 25 % higher than the maximum permitted frequency of 500 Hz. In this case the PWM input is deactivated to minimise the interrupt load on the control unit
- The PWM ratio is outside the error limits which correspond to half the lower reference parameter (parameter from 1500 *FuelSetPWMIn1RefLow)* or the mean between the upper reference parameter (parameter from 1501 *FuelSetPWMIn1RefHigh)* and 100 %.

6.4 Actuator position output with analogue output

The current actuator value can be output as a current on the port at Pin 1 and 3 when 4335 *ActPosOutPWMOrAnalog* = 0 is set.

Output	Use	Name	Terminal	Specification
Analogue output 2*	Current actuator value	P2	**1**, 3	0..22 mA

* optional PWM output

Table 12: Analogue output

6.4.1 Value range of the analogue output

For the current output it is usually not the maximum output range of approx. 0..22 mA that is desired but the standard output range of 4..20 mA.

Parameters 1646 *CurrentOut_RefLow* and 1647 *CurrentOut_RefHigh* are provided for matching the output range. The value to be entered is directly parameterized in electrical unit.

6.5 Actuator position output with PWM output

The current actuator value can be output as a PWM signal on the port at Pin 1 and 3 when 4335 *ActPosOutPWMOrAnalog* = 1 is set.

Output	Use	Name	Terminal	Specification
PWM output 2*	Current actuator value	P2	**1**, 3	0.3 A low side 50…500 Hz

* optional analogue output

Table 13: PWM output

6.5.1 Value range of the PWM output

Normally only a PWM ratio of between 5 and 95 % is desired.

Parameters 1601 *PWMOut_RefLow* and 1602 *PWMOut_RefHigh* must be used to match the output range. The limits are entered directly in percent PWM ratio.

The frequency of the PWM signals can be set with parameter 1625 *PWMOutFrequency*.

6.6 Common alarm via alarm output

The port on Pin 10 to which the signal 3801 *CommonAlarm* is automatically assigned is used for error display. This common alarm is always active when at least one error is present in the system. The output can be used for an optical or an acoustic signal.

Output	Use	Name	Terminal	Specification
Error output	Common alarm	Err	**10**, 21	0.3 A low side

Table 14: Alarm output

The common alarm can be configured high or low active depending on how 5000 *CommonAlarmLowActive* is set.

5000 *CommonAlarmLowActive* = 0 — Output = 1 when at least one error is present, otherwise 0

5000 *CommonAlarmLowActive* = 1 — Output = 0 when at least one error is present, otherwise 1

In order to tell a higher-level system that a new error has occurred, 5102 *CommonAlarmResetOn* = 1 is set. Consequently with each new error the active common alarm output is reset for 500 ms.

If the higher-level system must also be told that an error that was previously present has now cleared, then 5103 *CommonAlarmResetBoth* must also be set. The common alarm output will now be reset for 500 ms with each cleared error.

6.7 Auto adjust with pushbutton

At the left-hand edge of the control unit PCB – above the 9-pin communication plug – is a pushbutton which can be used to request the automatic actuator adjustment (↑ *8.1.2 Automatic calibration*).

7 Technical data

The PANDAROS system is based on type DC 6 control units.

The following technical data apply to the positioner control units.

7.1 General

Nominal voltage		12 V DC or 24 V DC
Min. voltage		9 V DC (short-time for starting)
Max. voltage		32 V DC
Current consumption		max. 7 A, max. 11 A for max. 60 seconds
Control unit fusing		12 A
Storage temperature		-40 °C to +85 °C
Operating temperature		-40 °C to +80 °C
Humidity		up to 98 % at 55 °C, condensing
Vibration resistance		max. ± 1.75 mm at 10 to 21 Hz max. 0.24 m/s at 21 to 45 Hz max. 7 g at 45 to 400 Hz
Shock		30 g, 11 ms - half-sine
Degree of protection		IP 20
Insulation resistance		> 1 MΩ at 48 V DC
Weight		approx. 0.5 kg
ESD	IEC 61000-4-2	6 kV contact discharge 8 kV air discharge
EMC	IEC 61000-4-3	Electromagnetic fields: 80 MHz - 2 GHz (3 s/dec)
	IEC 61000-4-4	Burst: 2 kV supply leads / 1 kV signal
	IEC 61000-4-6	Conducted high-frequency EMI: 150 kHz - 80 MHz (3 V_{rms})
	IEC 61000-4-5	Surge: 0.5 kV signal/signal, 1 kV signal/frame
	CISPR 16-1 / 16-2	Conducted emissions, Radiated emissions from enclosure port: 10 kHz - 30 MHz (conducted) 150 kHz - 2 GHz (emitted) 156 – 165 MHz (24 dB µV/m)

7.2 Inputs and Outputs

All inputs and outputs are protected against polarity reversal and short circuit-proof against battery positive and battery negative.

Pin 1	Position (analogue output)	I_a = 4 .. 20 mA, R_a = 20 Ω
	or Position (PWM output)	I_{sink} < 0.3 A, ground switching 50..500Hz, external R_{pu} required
Pin 2	Stop (analogue input)	U = 0..5 V, R_e = 100 kΩ, f_g = 15 Hz or I = 4 .. 20 mA, R_e = 200 Ω, f_g = 15 Hz
	or Setpoint (PWM input 1)	U_0 < 1 V, U_1 > 5 V, R_{pd} = 100 kΩ 50..500Hz
Pin 6	Reference voltage	U_{ref} = 5 V ±125mV, I_{ref} < 30 mA
Pin 7	Setpoint (analogue input)	U = 0..5 V, R_e = 100 kΩ, f_g = 15 Hz or I = 4 .. 20 mA, R_e = 200 Ω, f_g = 15 Hz
Pin 10	Error lamp (digital output)	I_{sink} < 0,3 A, ground switching
Pin 11	Stop (digital input)	U_0 < 1 V, U_1 > 5 V, R_{pd} = 100 kΩ, optional $R_{pu/pd}$ = 4.75 kΩ
	or Setpoint (PWM input 3)	U_0 < 1 V, U_1 > 5 V, R_{pd} = 100 kΩ, 50..500Hz
Pin 15/16/17	control path measurement	internal in the actuator with reference feedback
Pin 18/19	Actuator solenoid output	I < 7 A, I < 11 A for T < 60 s, PWM
CAN H/L	CAN communication	ISO/DIS 11898, standard/extended identifier, baud rate up to 1 Mbps
9-pole	Serial communication	HZM interface up to 57600 baud

Note

Pin 2 and Pin 11 are used alternately as PWM input for the setpoint depending on whether the stop is induced digitally or analoguely. With an analogue setpoint this makes no difference.

7.3 Terminal connection diagrams

All of the terminal connection variants described in this publication are illustrated schematically below.

They differ in the type of setpoint selection, the stop signal and the actuator.

The setpoint selection can be analogue using a setpoint potentiometer or a current or voltage signal. PWM setpoint selection is also possible. Setpoint selection by a CAN protocol can also be implemented on request.

The stop signal can be selected digitally or analoguely by a current or voltage signal.

Digital inputs are not monitored for errors and must not be used for safety-related functions as defined by classification societies

> Therefore it is recommended to use a monitored analogue input instead of a digital input for the external stop request (↑ 5.2.1.2 External stop request by an analogue input).

In the diagrams, the actuators are distinguished according to those with a disc rotor motor and gearless actuators. The former type include StG 6 to StG 40, for example. The fast 2000 series actuators, linear actuators or the Bosch-EDC can be used in the gearless version.

7.3.1 Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor

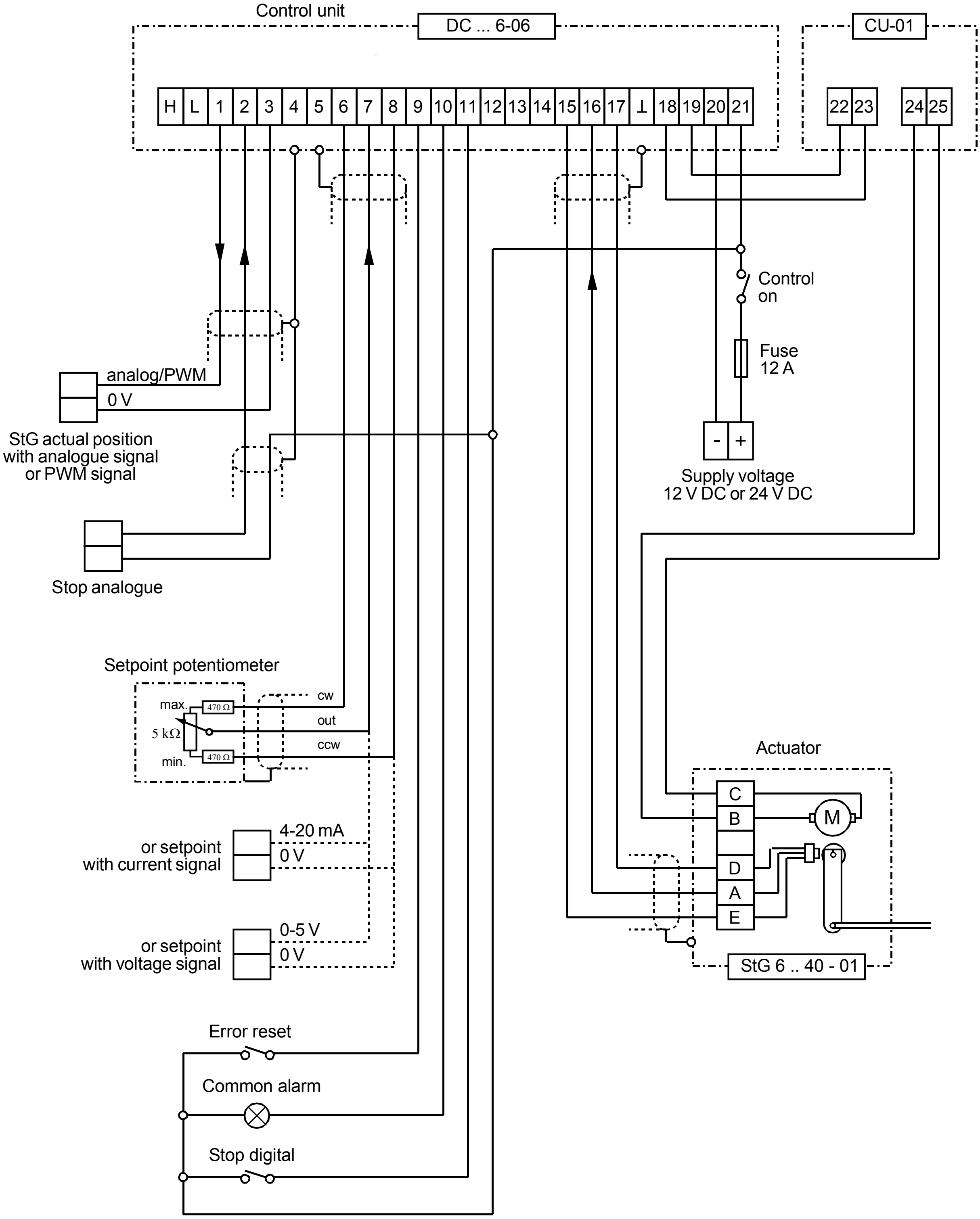

Fig. 2 Terminal connection diagram 1
– Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor

7.3.2 Analogue setpoint, stop signal digital and/or analogue, gearless actuator

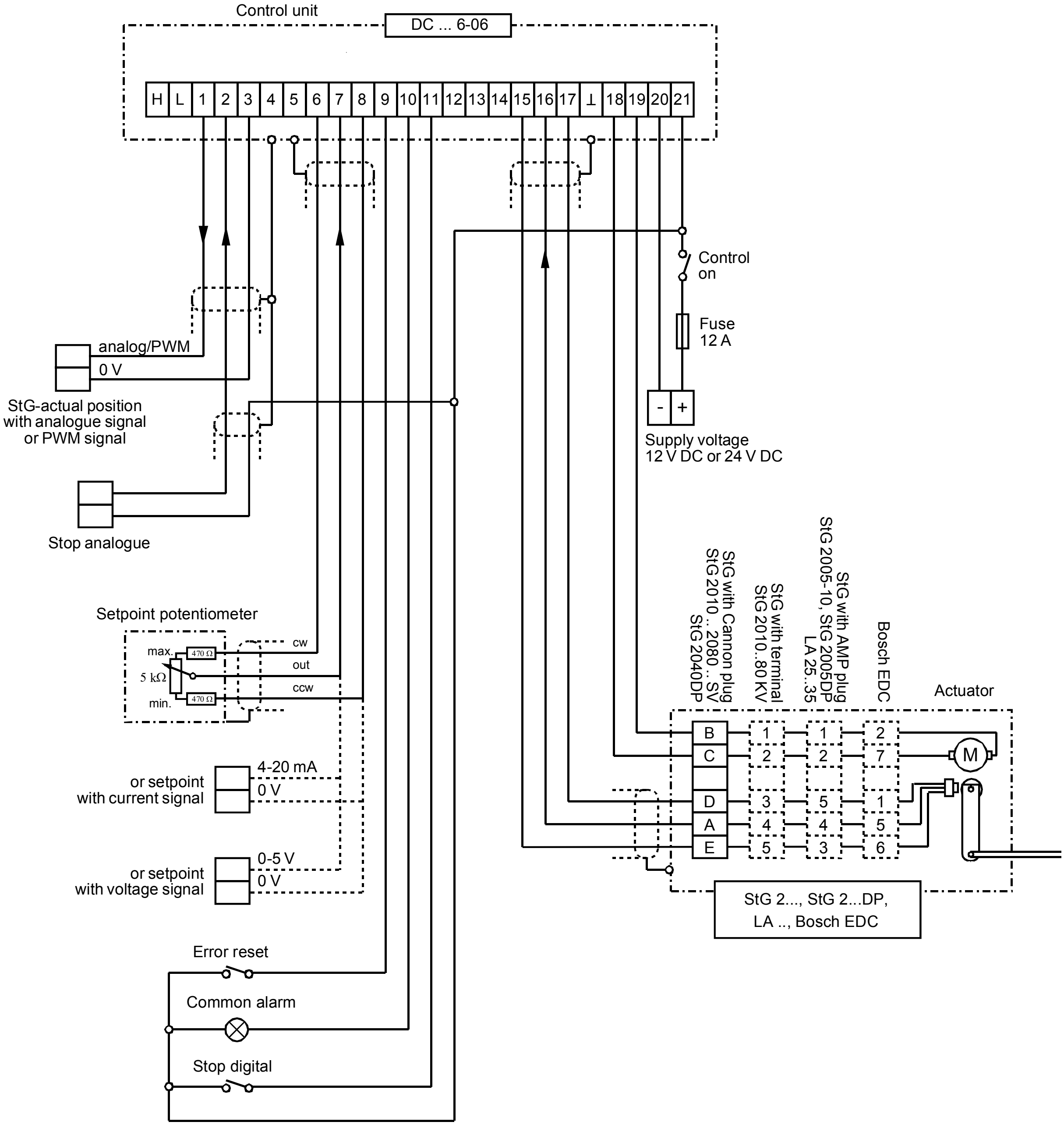

Fig. 3 Terminal connection diagram 2
– Analogue setpoint, stop signal digital and/or analogue, gearless actuator

7.3.3 PWM setpoint, stop signal digital, actuator with disc rotor motor

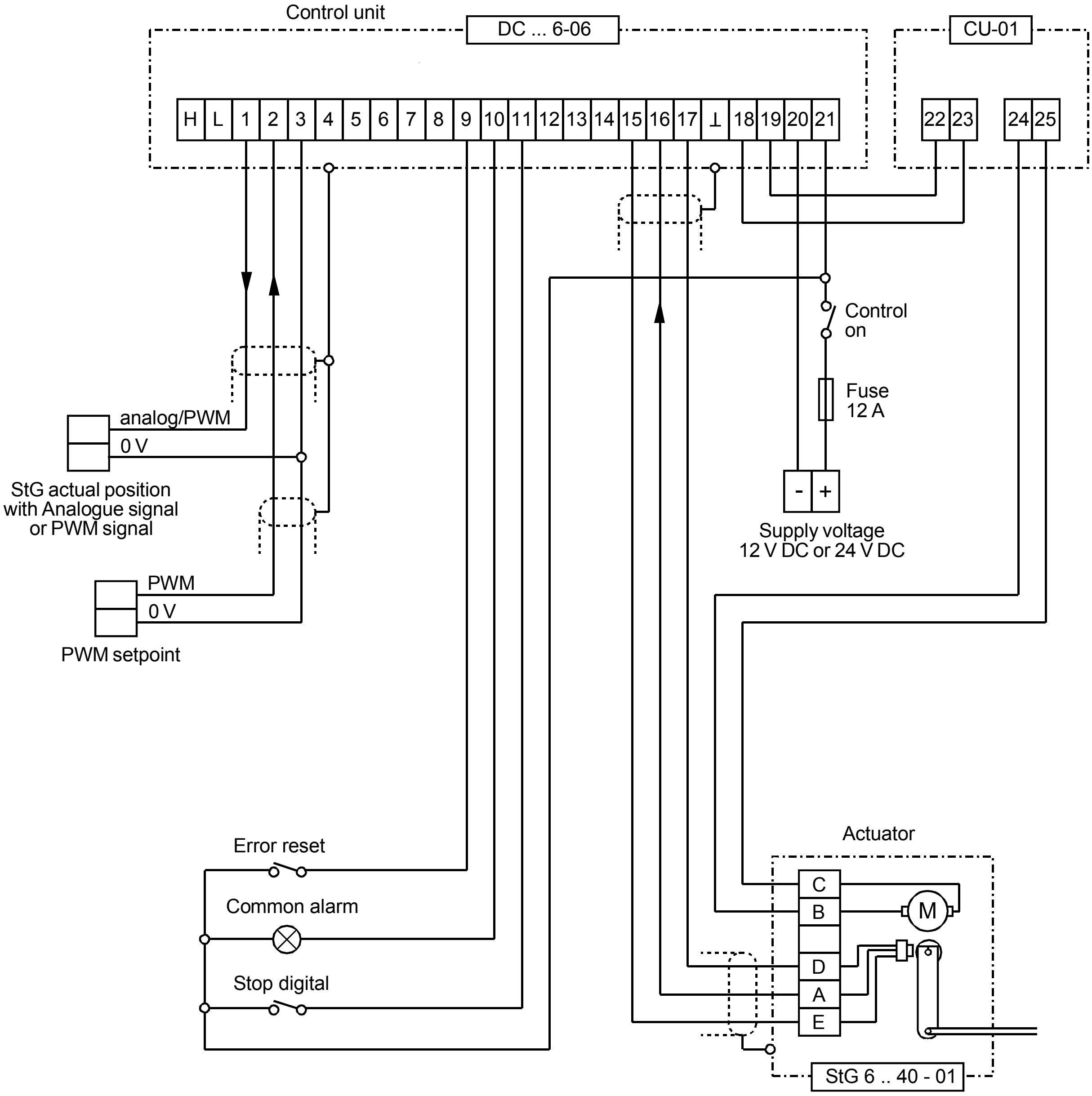

Fig. 4 Terminal connection diagram 3 – PWM setpoint, stop signal digital, actuator with disc rotor motor

7.3.4 PWM setpoint, stop signal digital, gearless actuator

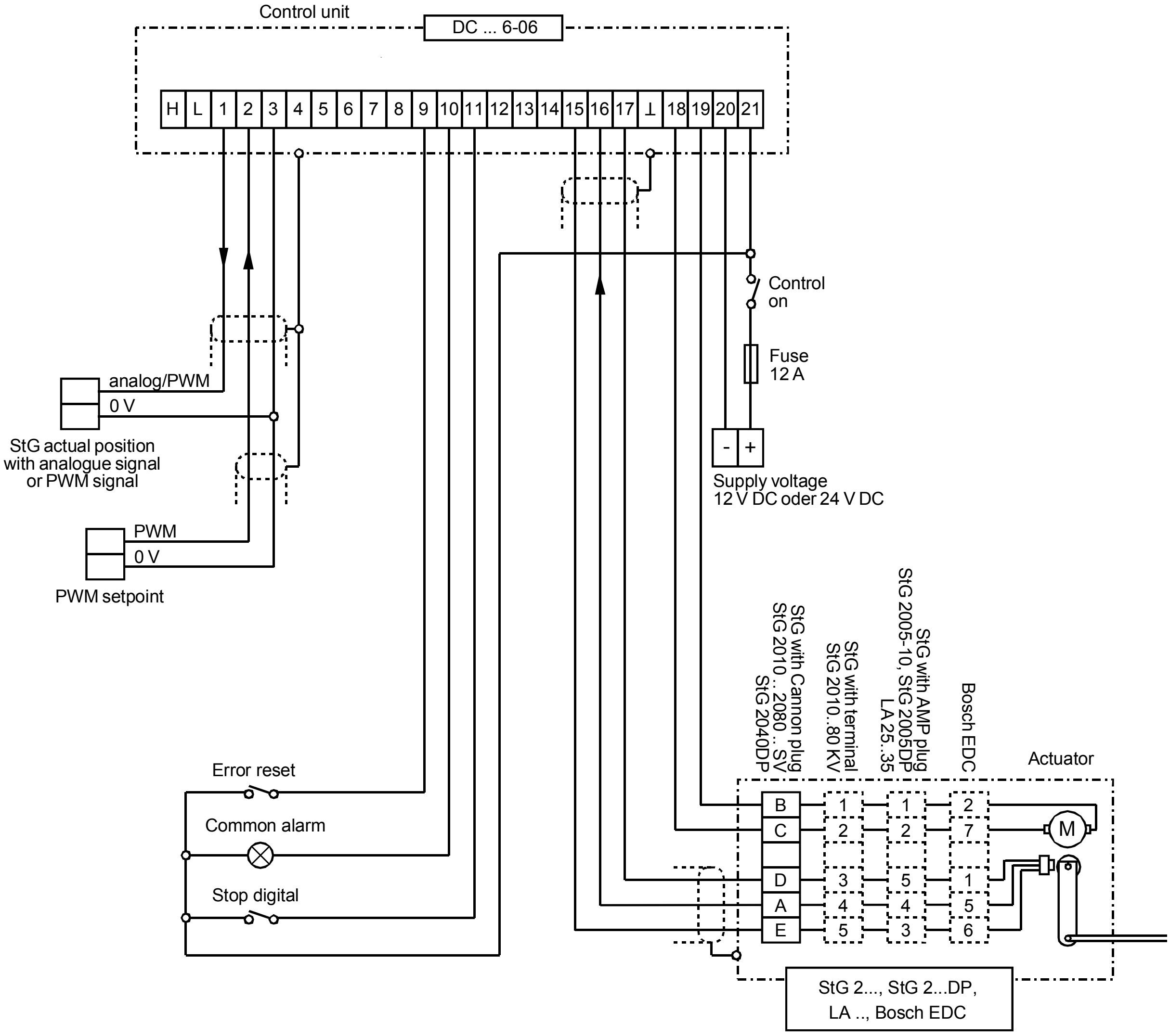

Fig. 5 Terminal connection diagram 4 – PWM setpoint, stop signal digital, gearless actuator

7.3.5 PWM setpoint, stop signal analogue, actuator with disc rotor motor

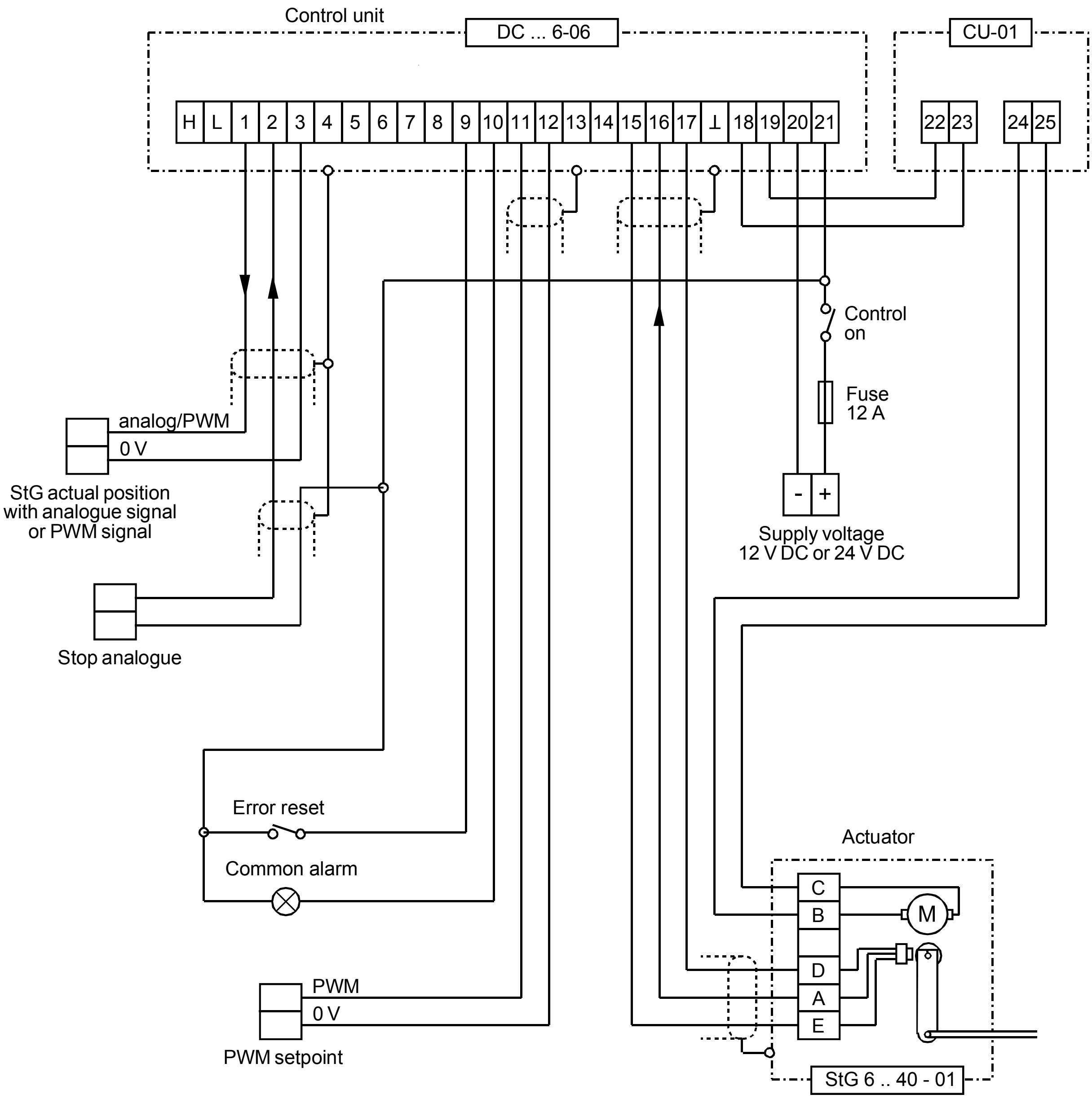

Fig. 6 Terminal connection diagram 3 – PWM setpoint, stop signal analogue, actuator with disc rotor motor

7.3.6 PWM setpoint, stop signal analogue, gearless actuator

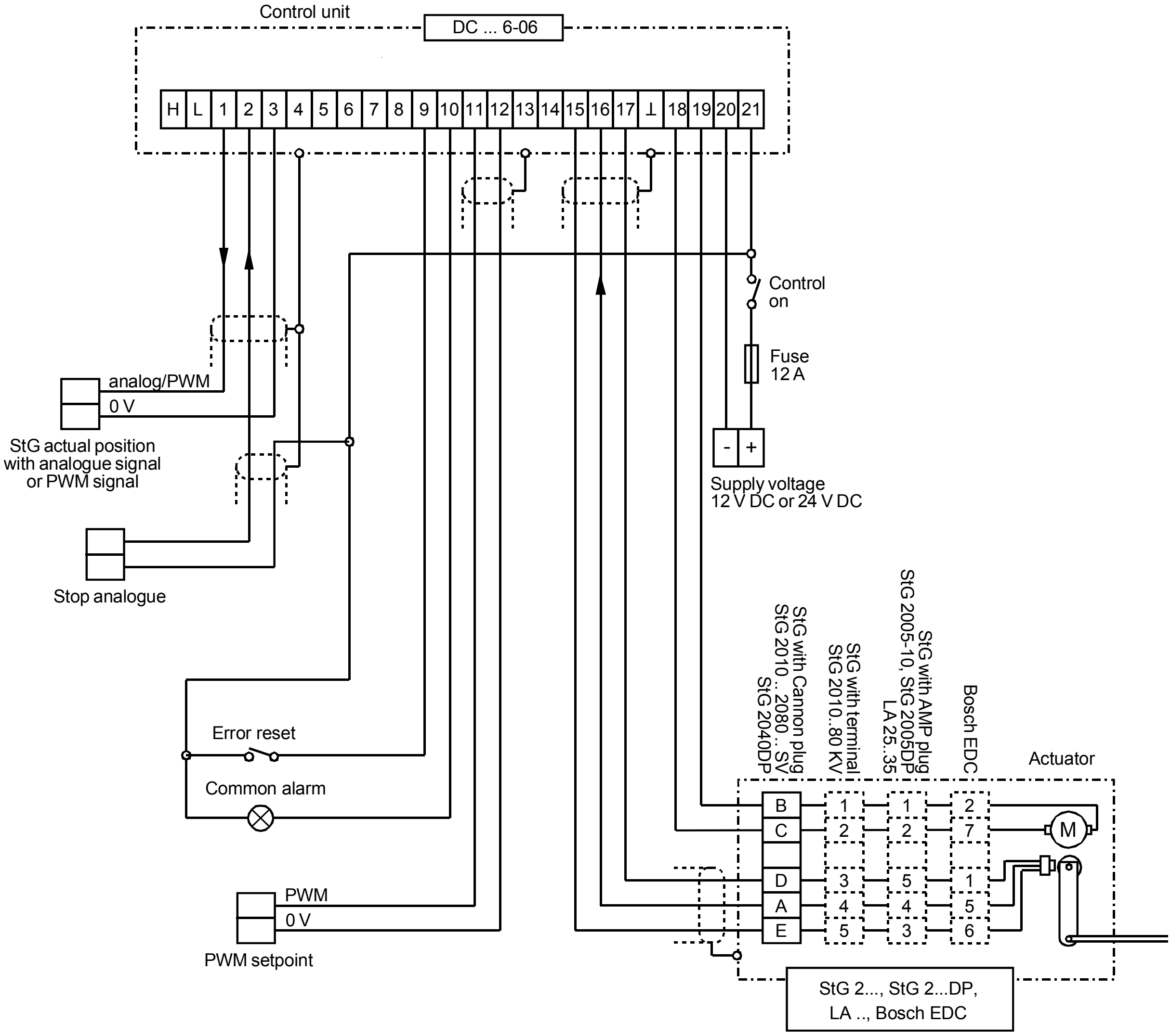

Fig. 7 Terminal connection diagram 4 – PWM setpoint, stop signal analogue, gearless actuator

8 Actuator and feedback

The **HEINZMANN** control units with conventional injection can work with actuators with different methods of operation and feedback. They can trigger actuators with either 2-quadrant operation (electrically energized on one side, with a strong spring return) or with 4-quadrant operation (electrically energized on both sides).

Depending on the actuator type, an analogue or a digital signal is used to feed back the actuator position. With an analogue signal the information about the actuator position is contained in the size of the d.c. voltage signal, whereas with a digital feedback the actuator position is computed from time intervals between pulses. As well as the measurement signal for the actuator position, actuators with digital feedback have a reference signal. The reference signal is used to compensate for temperature variations in the feedback which might affect the measurement signal.

For some actuators, the evaluation of the feedback must be inverted (small feedback signal with a large filling and vice versa).

The actuator is activated with the parameter 5910 *ActuatorOn*.

The operation of the amplifier is selected with the parameter

5911 *Amplifier2QOr4Q* = 0	4-quadrant amplifier
5911 *Amplifier2QOr4Q* = 1	2-quadrant amplifier (with return spring)

and the type of feedback of the actuator is set with the parameter

5950 *FeedbDigitalOrAnalog* = 0	Analogue feedback
5950 *FeedbDigitalOrAnalog* = 1	Digital feedback.

Whether the feedback signal falls or rises as the filling increases can be set separately with the following parameter:

5951 *FeedbSlopeFallOrRise* = 0	Rising feedback signal for actuator 1 as filling increases
5951 *FeedbSlopeFallOrRise* = 1	Falling feedback signal for actuator 1 as filling increases

Note

Parameters 5911 Amplifier2QOr4Q, 5950 FeedbDigitalOrAnalog and 5951 FeedbSlopeFallOrRise are not active until after ↑ 3.2 Saving the data and ↑ 3.7 Resetting a control unit.

After a re-start, a reset or a stop request, the control unit usually energizes the actuator for safety for 5 seconds with the simultaneous output of an actuator position setpoint 0 %, and then switches the current off.

8.1 Calibrating the actuator

Before the control unit can compute the exact position of the control path, reference values must tell it the relationship between the control path measured value and the position of the actuator. These reference values correspond to the control unit's measured values at the minimum and maximum position of the actuators. In the case of actuators with digital feedback, the reference signal must also be measured.

Note

Actuator calibration must be carried out for every control unit with its associated actuator, otherwise component tolerances of both the actuator and the control unit will affect control quality and especially the compliance of the limiting functions. Actuators must be re-calibrated if either the actuator or the control unit have been replaced.

Calibration can be carried out automatically or manually. The actuators must be capable of reaching the minimum and maximum position in all cases. Only if the actuator is calibrated to its full stroke can exact positioning be expected subsequently. The greater the difference between the measured values at the 0 % and the 100 % positions, the more accurately the set position can be controlled. The actuator should therefore be calibrated without its linkage if possible.

Note

Calibration determines the 0 % and 100 % positions of the actuator. The subsequent approach to these positions and to all intermediate values is initiated by the external setpoint in the range 0..100 %. The 0 % position is approached with the 0 % setpoint, the 100 % actuator position is approached with the 100 % setpoint. If the external setpoint and the actuator setpoint are not identical, an adaptation curve must be interposed (↑4.2 Determining the actuator setpoint).

8.1.1 Manual calibration

Manual calibration is performed similarly to the calibration of analogue inputs.

The actuator must be moved to the minimum and maximum position, with the control unit's measured value

3950 *Feedback*

being entered in the parameter

1950 *FeedbackRef_0%*

for the minimum position and in the parameter

1951 *FeedbackRef_100%*

for the maximum position.

For actuators with digital feedback, the reference signal 3955 *FeedbackRef* must also be entered in parameter 1955 *FeedbackReference*. The reference signal does not change over the entire range of the actuator.

Note

With manual calibration, these parameters are only active after ↑ 3.2 Saving the data and ↑3.7 Resetting a control unit.

The control unit can now scale the measured value of the feedback and display in exactly in 3960 *FeedbackCorrection*. The actuator position can be checked with parameter 2300 *ActPos,* which displays the current control path of the actuator.

8.1.2 Automatic calibration

If necessary, automatic calibration can be performed by the PC or hand programmer (*↑3.3 DcDesk 2000*), with a pushbutton on the PCB or the switch function 2845 *SwitchAutoAdjust* which is activated by a digital input. Please refer to *↑8.1.2.1 Saving the calibration data* without fail!

Note

Automatic actuator calibration cannot be started if a fatal error is present or a setpoint not equal to 0 % is selected.

With automatic calibration, the control unit measures the reference values by itself. It does this by energizing the actuator for a certain time to ensure that the minimum or maximum position is positively reached, then measures the reference values. The measured values are entered in the corresponding parameters

1950 *FeedbackRef_0%*

for the minimum position and

1951 *FeedbackRef_100%*

for the maximum position and unlike manual calibration they are immediately available.

The time during which the control unit energizes the actuator and waits for the minimum or maximum position to be positively reached is defined by the parameter

1900 *Feedback-AdjustTime*

The level of current for automatic calibration is determined by the parameter

1919 *ServoCurrentAdjust*

The error

3059 *ErrFeedbackAdjust*

can occur during automatic calibration. The cause of this error is described in Chapter *↑10.8 Error parameter list* which also gives guidance on remedying the problem.

8.1.2.1 Saving the calibration data

When automatic calibration is initiated with DcDesk 2000 or the hand programmer, the device asks the operator if he wishes to save the results in the read-only memory. It is therefore the responsibility of the operator to decide whether to accept the settings as final. If calibration is initiated with the pushbutton or a switch function however, the settings will be automatically saved following successful automatic calibration.

Note

The time for the calibration itself is determined by 1900 FeedbackAdjustTime – once each at 0 % and 100 % position plus 1 s. The control unit needs an extra 30 s approx. to automatically save the measured settings. This means that after calibration is complete, at least twice the calibration time + 31 s must be allowed to elapse before the control unit can be de-energized or before a reset can be initiated in some other way!

8.1.3 Error detection from the feedback

Similarly to sensors at analogue inputs, the feedback has error limits by which the control unit can detect when a measured value is prohibited. The error limits must be entered by hand with both manual and automatic calibration.

In this process, a measured value that is below the lower error limit

1952 *FeedbackErrorLow*

and above the upper error limit

1953 *FeedbackErrorHigh*

is defined as prohibited. The error limits are designated “Low/High” and refer to the minimum and maximum measured value respectively – not to the actuator position.

Similarly, the error limits 1956 *FeedbackRefErrLow* and 1957 *FeedbackRefErrHigh* apply to the reference value in the case of digital feedback.

These error limits should not be too close to the minimum and maximum value to prevent natural measured value fluctuations in the feedback causing error detection. A short circuit or open circuit in the supply or signal line must be positively detected however.

A detected error sets the corresponding error parameter of the feedback. The response to this error is described in ↑*10.8 Error parameter list*.

8.2 Limiting the actuator stroke

Absolute limiting of the actuator's control path is also provided to protect the actuator from mechanical and thermal overload. This limiting provides a safety distance ahead of the actuator's mechanical limit stops.

The minimum position of the control path is limited by parameter 310 *ActPosSecureMin*. Parameter 312 *ActPosSecureMax* limits the maximum position of the control path. The following values are usually used for these two parameters:

Number	Parameter	Value	Unit
310	*ActPosSecureMin*	*3.0*	*%*
312	*ActPosSecureMax*	*97.0*	*%*

8.3 Servo loop

The control unit specifies a filling setpoint 2350 *FuelQuantity* from which a set position for the actuator has to be calculated. Because 2350 *FuelQuantity* is provided as a percentage, with a linear linkage the actuator setpoint 2330 *ActPosSetpoint* can be taken directly from 2350 *FuelQuantity*.

In certain cases however it may be necessary to take the actuator setpoint from a filling-dependent curve. This is especially useful with nonlinear linkages or throttles, or if the actuator is mounted directly without any linkage (*↑4.2 Determining the actuator setpoint*).

8.3.1 Servo control loop

The task of the servo or position control loop is to approach the selected set position 2330 *ActPosSetpoint* for the actuator. Parameters for P, I and D components are available for this control loop. There is also an additional parameter which counteracts the actuator's acceleration. This parameter is used primarily with the particularly fast actuators in the 2000 series.

1911 *ServoGain*	P component for the servo loop of the actuator
1912 *ServoStability*	I component for the servo loop of the actuator
1913 *ServoDerivative*	D component for the servo loop of the actuator
1914 *ServoAcceleration*	DD component for the servo loop of the actuator
2300 *ActPos*	Current value of the actuator
2330 *ActPosSetpoint*	Set position for the actuator

A simple correction of the PID parameters of the servo loop in the static state which is defined by the range 1906 *ServoCorrRange* is possible:

If the actuator position deviation between target and actual position is within the range 1906 *ServoCorrRange*, then the PID parameters are corrected with the value 1905 *ServoCorrFactor*, i.e. they are usually reduced.

The normal parameters are used outside the double range. Between these two values, interpolation is used to achieve a smooth transition. This function is always active. A value of 100 % means no influence on the servo loop parameters. The current correction factor is displayed in the parameter 3905 *ServoPIDCorr*.

The values for the servo loop vary depending on the actuator type and must be set accordingly. The initial settings are made at the factory by HEINZMANN when the control unit is shipped and do not normally need to be altered.

8.3.2 Actuator current

The servo loop uses the servo loop parameters to calculate the current 3916 *ServoCurrentSetpoint* for the actuator. To prevent the actuator from being overloaded, the maximum current can be limited with parameter 1917 *ServoCurrentMax*. The maximum current can be tolerated briefly for position changes, but over longer periods the current has to be reduced to prevent the actuator being thermally overloaded. This is why the servo controller reduces the current under static load by an exponential function with a time constant of approx. one minute to the value set with parameter 1918 *ServoCurrentRed.* The reduction only begins after the delay time 1916 *ServoCurrentRedDelay*. If this parameter does not exist, the reduction begins without a delay.

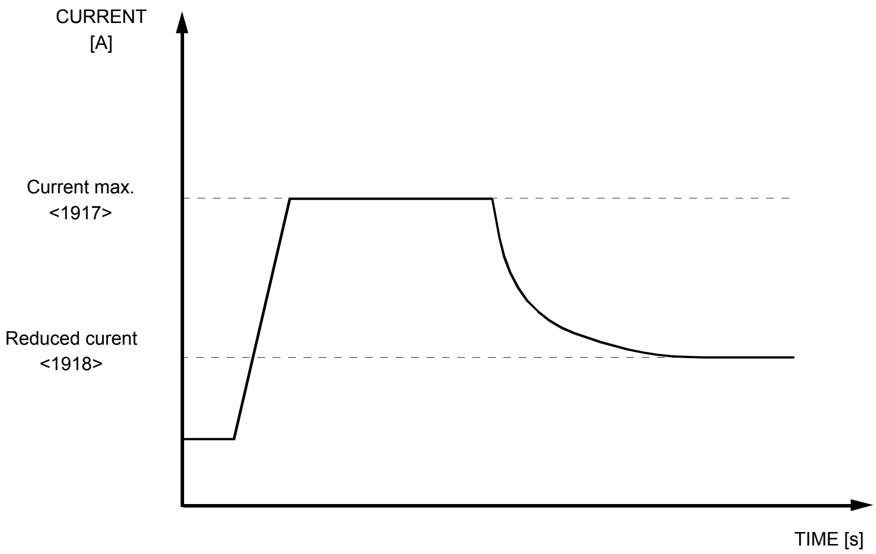

Fig. 8 Current reduction under static load

The maximum current is still available for dynamic position changes. When setting the current reduction, the current must be measured in the supply cable to the actuators, with the corresponding cable length. It must be remembered that the current cannot be measured until the actuators have warmed up (static operation) because the flowing current changes as temperature rises.

Control units of the type PANDAROS 6-06 are equipped with a current-regulated amplifier.

8.4 Test mode

Good system integration is only possible when the actuators position accurately. PID parameters to match the actuator are set at the factory. However when the actuator is installed it is affected by both the mechanical mounting via the linkage and by the dimensions and weights of the control rods.

The purpose of the test mode therefore is to make it possible to optimise the servo loop parameters without any influence from the control unit. It is used solely for testing and setting up purposes.

The test mode can only be started when the actuator selected for the positioning test has been switched on with parameter 5910 *ActuatorOn*. For safety reasons the program automatically exits the positioning mode if the setpoint generator generates values not equal to 0 and there is no external stop signal present. The test mode cannot be stored either, i.e. it is automatically deactivated off after a control unit reset.

The test mode is activated with parameter 5700 *ActuatorTestOn* = 1. If an actuator setpoint is now selected with parameter 1700 *ActuatorTestSetpoint*, the actuator's reaction can be observed and/or measured from the feedback. The feedback value is displayed in 2300 *ActPos*. By changing the setpoint, the step response of the actuator can be monitored and optimised by adjusting the PID values.

In order to check not only the static setting but to also be able to optimise the much more important dynamic response of the actuator, the change in setpoint selection can be automated. A test mode is selected with parameter 5701 *ActuatorTestMode* which can assume the following settings:

5701 *ActuatorTestMode* = 0	the setpoint from 1700 *ActuatorTestSetpoint* is output.
5701 *ActuatorTestMode* = 1	the setpoint from 1700 *ActuatorTestSetpoint* is alternately positively or negatively exposed to the value from 1701 *ActuatorTstAmplitude*. The result is a square-wave function.
5701 *ActuatorTestMode* = 2	a delta function is formed from 1700 *Actuator-TestSetpoint* and 1701 *ActuatorTstAmplitude*.

The frequency for the positional change is defined in 1702 *ActuatorTstFrequency*.

Once the servo loop settings have been completed in test mode, they must be checked with the system running and adjusted if necessary. Owing to vibration, the friction moment may be significantly less and other parameter settings than in a static system may be required. In extreme cases this can even result in a situation where the actuator positions very poorly when the system is stopped but produces very good control results when the system is running.

9 Data Management

The control unit contains a number of parameters which indicate the control unit type and the firmware version.

9.1 Serial Number of the Control Unit

Each individual control unit is uniquely identified by a serial number. The first 4 digits indicate the production year and the shipping month. The remaining digits consist of the sequential production number. The serial number can be seen on the **HEINZMANN** nameplate and in the following parameters:

3844 *SerialDate*	Year and month of manufacture
3845 *SerialNumber*	Sequential production number

9.2 Identification of the Control Unit

The application-related functionality of a control unit is uniquely defined by the firmware which only runs on exactly one particular hardware type.

3840 *HardwareVersion*	Version number of the control unit hardware
3841 *AddHardwareVersion*	Version number for hardware modification
3842 *SoftwareVersion*	Version number of the control unit firmware
3843 *BootSoftwareVersion*	Version number of the bootloader software

The software version consists of an unique two- to four-digit customer number x defined by **HEINZMANN**, a one- to two-digit variant number y and a two-digit change index z.

xx.y.zz or xxxx.yy.zz

DcDesk 2000 and hand programmer only allow the customer to access control units which contain the HEINZMANN basic software 00.y.zz or a custom firmware xx.y.zz with his own customer number x. The variants y define different firmware, e.g. for different applications of a manufacturer. Each variant can exist in different change levels z by means of software enhancements; the next higher change index always incorporates the next lower change index and supersedes it (↑***2.2 Firmware***).

9.3 Identification of the PC Program and Hand Programmer

Each dongle of the **HEINZMANN** DcDesk 2000 PC program and each **HEINZMANN** hand programmer (↑ *3.3 DcDesk 2000*) that is needed to edit parameters has its own identification number which is sent to the control unit. The current identification number of the PC program or hand programmer appears in parameter 3850 *Identifier*. The identification number of the dongle or hand programmer with which the last parameter change was stored in the control unit is indicated by the parameter 3851 *LastIdentifier*. The owner of this identifier is responsible for parameterization.

10 Error handling

10.1 General

The **HEINZMANN** control units have an integral error monitor with which errors in setpoints, sensors etc. can be detected and displayed.

There is a permanently assigned digital error output to which the general error message is applied.

10.2 Error types

The following basic error types can be identified:

- Errors in configuring and parameterizing the control unit

  These errors as caused by incorrect inputs made by the user and which the PC or hand programmer cannot intercept. They do not occur with a standard production control unit.

- Errors during ongoing operation

  These errors are the most important errors in a control unit in serial operation. This category includes sensor errors such as the failure of the setpoint generator.

- Internal computing errors of the control unit

  These errors can be due to faulty components or other unacceptable operating conditions. They do not occur in normal circumstances.

The different errors can be taken from parameter numbers 3000..3099. When an error is currently present the value is set to 1, otherwise it is 0.

When rectifying an error, the cause should be eliminated first and then the current errors should be cleared (reset). Some errors also reset by themselves as soon as the cause has been removed. Errors can be reset with a PC, a hand programmer or, given the appropriate configuration, with the switch function 2828 *SwitchErrorReset*. If the error is still present, you will need to continue looking for the cause.

The control unit always starts with the assumption that no error is present and then checks the error conditions. The control unit can therefore be put in an error-free condition by a ↑*3.7 Resetting a control unit*, however currently present errors are immediately displayed again.

10.3 Error display

The errors are divided into two basic groups. There are errors where operation can be continued although functionality may be restricted (e.g. sensor errors). The other group are so-called fatal errors which result in an emergency shutdown of the system (e.g. program memory errors).

These two error groups are signalled by the following two parameters:

3800 *EmergencyAlarm*

3801 *CommonAlarm*

Parameter 3801 *CommonAlarm* is set with every error that occurs, 3800 *EmergencyAlarm* is only set with fatal errors (↑ *10.7 Emergency shutdown errors)*. Consequently, 3800 *EmergencyAlarm* cannot occur by itself.

The common alarm output can be configured so that it is reset for 0.5 s when a new additional error occurs. A PLC connected to this output can therefore detect the new error. Parameter 5102 *Common-AlarmResetOn* = 1 must be set for this purpose. To also obtain an edge change when an error clears, 5103 *CommonAlarmResetBoth* must also be set to 1.

10.4 Error memory

When the control unit is powered off it loses all its information about the current errors. However the unit has an integral permanent error memory that provides an overview of errors that have occurred. Every error that has occurred at least once is stored in this error memory.

Control units of the type PANDAROS 6-06 contain an enhanced error memory as standard (↑ *10.4.1 Operating data memory and enhanced error memory*) which logs the number of occurrences and the times when the error occurred.

For the firmware, the values in the error memory are just display values and are not otherwise relevant. It only responds to the occurrence of errors during ongoing operation.

The permanent error memory can be viewed using the parameters starting from number 3100. Here the error counter stands for every error that has occurred since the last error memory reset. The numbers of these historic errors are incremented by 100 compared to their associated current error.

The permanent error memory can only be reset with the PC or hand programmer. The control unit then starts to accumulate new errors in the empty error memory.

Once a system has been commissioned, the error memory should always be reset to ensure that errors which occurred during commissioning, e.g. by sensors that are not yet connected, are not subsequently evaluated as errors during system operation.

Note

When parameter 5100 NoStoreSerrOn = 1 is set and the error memory is then cleared, no errors will be stored in the error memory until the next ↑3.7 Resetting a control unit. This means that a control unit with a customised data record can be shipped in error-free condition without the inputs having to be stimulated with the correct values. Parameter 5100 itself cannot be stored.

10.4.1 Operating data memory and enhanced error memory

Control units of the type PANDAROS 6-06 contain an operating data memory and an enhanced error memory as standard.

The operating hours – the hours run by the system with setpoint selection not equal to 0 – in 3871 *OperatingHourMeter* and 3872 *OperatingSecondMeter* are recorded as the operating data.

For each error that occurs since the last error memory reset, there is an error counter and the time of the first and last occurrence. These times are shown in operating hours. Up to 4 environmental data can also be logged with each error, to match the time of the last occurrence. The maximum number of environmental data is set by the firmware that is used, and cannot be changed. However the choice of environmental data can be freely parameterized by the user with DcDesk 2000.

The error counters are displayed in the error memory parameters starting from 3101 *SErr...*. The times of the first and last occurrence and the environmental data can be viewed in the error memory window of DcDesk 2000 or of the hand programmer HP 03.

10.5 Bootloader

The **HEINZMANN** control units contain a so-called bootloader. This part of the program is located in a certain section of the ROM and is programmed at the factory. The bootloader cannot be erased.

When the control unit is started by switching on the operating voltage or a reset, the bootloader program is always run first. It executes important tests which tell it whether the actual control unit program can function or not. The bootloader then decides whether the rest of the program routine can be sent to the control unit program or whether it should stay in bootloader mode for the safety of personnel and the machine. The system cannot be started while the program is still in bootloader mode.

Note

All tests run by the bootloader and the subsequent initialisation of the main program tale approx. 200 ms.

10.5.1 Bootloader Start Tests

The next section describes the tests performed by the bootloader and the resulting actions. Communication with the unit is not possible while the tests are in progress, especially when the program sticks in an endless loop because of a fatal error. This is why different types of display on the control units are used to indicate the current test mode.

- Watchdog Test

  Type of display in the event of an error: Error lamp is on

  The test checks whether the watchdog built into the processor is operational. This purpose of the test is to ensure that in an undefined program flow, the control unit enters a safe state after a defined time. If the watchdog test is negative, the bootloader program stays in an endless loop and the said display stays on.

- Internal RAM Test

  Type of display in the event of an error: Error lamp is on

  Different bit patterns are read to the internal processor RAM memory and read back again. If at least one cell does not contain the expected code, the bootloader program enters an endless loop and the said display stays on.

- Bootloader Program Test

  Type of display in the event of an error: Error lamp is on

  A checksum is calculated over the memory area that contains the bootloader program and is compared with the checksum programmed at the factory. If they do not match, the bootloader program stays in an endless loop and the said display stays on.

- Control Unit Program Test

  Type of display in the event of an error: Error lamp flashes: three times short, long off, three times short, ...

  A checksum is calculated over the memory area that contains the control unit program and is compared with the programmed checksum. If they do not match, the bootloader enters a state in which the error 3087 *ErrMainCheckSum* is displayed via the serial communication (DcDesk 2000 PC program or hand programmer).

- Watchdog Tripped

  The bootloader enters a state in which the watchdog error 3089 *ErrWatchdog* is displayed via the serial communication (DcDesk 2000 PC program or hand programmer).

10.5.2 Bootloader Communication

Communication between DcDesk 2000 or a hand programmer and the bootloader can commence when the error lamp flashes.

In this condition, errors are indicated and it is also the starting point for the download of a new control unit program (only with DcDesk 2000) which is basically performed by the bootloader.

10.6 Configuration error

If the control unit is wrongly configured, this is shown in 3092 *ErrConfiguration*. The unit's configuration may be incorrect, for example, if the channel type is disregarded when inputs or outputs are parameterized.

In addition to 3092 *ErrConfiguration*, an error code appears in 3000 *ConfigurationError* which tells you about the type of error. The display of 3000 *ConfigurationError* changes every second and shows all existing configuration errors.

Note

The communication program DcDesk 2000 displays the error text of configuration errors in the "Current Errors" window.

Although it may be possible to clear a configuration error using the command "Clear Error", this does not remove the cause of the error. Most configuration errors are only checked when the control unit is started, so a reset is required when the parameterization is changed and the parameters are saved in the control unit.

The error codes and their meanings are given in the following tables. Whether one of the listed communication protocols is implemented or not will depend on the particular control unit firmware version. This means that not all of the errors listed here will occur in a special control unit.

Configuration error	
10	Port is used as analogue input but not configured accordingly
15	Port is used as analogue output but not configured accordingly
20	Port is used as PWM input but not configured accordingly
25	Port is used as PWM output but not configured accordingly
30	Port is used as digital input but not configured accordingly
35	Port is used as digital output but not configured accordingly
40	The characteristic curve ActPosSetoint[FuelSetpoint] is not parameterized with a rising characteristic

10.7 Emergency shutdown errors

The following list gives a summary of all the errors that either prevent starting or result in an emergency shutdown during operation. The presence of at least one of these so-called fatal errors will activate 3800 *EmergencyAlarm*. These errors are described in detail in Chapter ↑*10.8 Error parameter list*.

Error	Cause
3050 *ErrFeedback*	Error in the actuator feedback
3056 *ErrFeedbackRef*	Error in the actuator feedback reference
3059 *ErrFeedbackAdjust*	Error in the actuator auto adjustment
3060 *ErrAmplifier*	Error in the amplifier
3075 *ErrClearFlash*	Error when clearing the flash memory (display in the bootloader)
3076 *ErrParamStore*	Error when storing the E^2PROM memory parameters
3077 *ErrProgramTest*	Error in the ongoing test of the program memory
3078 *ErrRAMTest*	Error in the ongoing test of the RAM memory
3089 *ErrWatchdog*	Undefined program flow, internal program error (display in the bootloader)
3090 *ErrData*	No parameters, or checksum of parameters is incorrect (always active after a program download)
3093 *ErrStack*	Stack overflow, internal program error
3094 *ErrIntern*	Exception, internal program error

Note

An actuator auto adjustment (↑ 8.1.2 Automatic calibration) cannot be performed when a fatal error is present.

10.8 Error parameter list

The following error parameter list describes the causes of the individual errors and the reaction of the control unit. It also indicates actions to rectify the errors.

The errors are entered in the current error memory starting from parameter number 3000 and in the permanent error memory starting from number 3100.

The errors are listed by ascending order of numbers, with the parameter on the left being the current error in the volatile error memory and the one on the right the related parameter in the permanent error memory. The control unit only reacts to current errors, the permanent error memory is only used to collect errors which have occurred.

3000 ConfigurationError

Cause: Configuration error.

Reaction: Display

Action: Check the control unit configuration and rectify, save parameters and perform a control unit reset

Reference: *↑10.6 Configuration error*

3005 ErrAnaStopRequest	**3105 SErrAnaStopRequest**
3007 ErrFuelSetpAna	**3107 SErrFuelSetpAna**
3009 ErrFuelSetpPWMIn1	**3109 SErrFuelSetpPWMIn1**
3011 ErrFuelSetpPWMIn3	**3111 SErrFuelSetpPWMIn3**

Cause: An error has occurred at the corresponding setpoint/sensor input (e.g. short circuit or open circuit).

Reaction: Depending on the configuration, the error may clear by itself when the control unit's measured values are within the error limits again.

Action:
- Check the cable for short circuit or open circuit.
- Check the corresponding sensor and replace if necessary.
- Check the error limits for the sensor.

Reference:
- ↑ 6.2.3 Error detection with the analogue inputs
- ↑ 5.3.1 Monitoring the analogue setpoint
- ↑ 6.3.2 Error detection with the PWM inputs
- ↑ 5.3.2 Monitoring the PWM setpoint

3050 ErrFeedback **3150 SErrFeedback**

Cause: Error in the feedback system of the actuator, actuator not connected.

Reaction:
- Control unit cannot be operated.
- Emergency shutdown.

Action:
- Check the feedback cable to the actuator.
- Check the actuator and replace if necessary.
- Check the error limits for the feedback:
  1952 *FeedbackErrLow* / 1953 *FeedbackErrHigh*

Reference: - *↑8 Actuator and feedback*

3053 ErrActuatorDiff **3153 SerrActuatorDiff**

Cause: The difference between the target control path and the actual control path exceeds 10 % of the total control path for more than one second. This situation exists when the injector pump, the throttle, the linkage or the actuator sticks or is not connected.

Reaction:
- Error message.
- The error clears when the difference is below 10 % again.

Action:
- Check the injector pump or throttle, replace if necessary.
- Check the mechanism (linkage).
- Check cable to the actuator.
- Check actuator, replace if necessary.

Reference: *↑8 Actuator and feedback*

3056 ErrFeedbackRef **3156 SerrFeedbackRef**

Cause: Error in the feedback system of the actuator, actuator not connected.

Reaction:
- Control unit cannot be operated.
- Emergency shutdown.

Action:
- Check the feedback cable to the actuator.
- Check the actuator and replace if necessary.
- Check the error limits for the reference of the feedback:
  1956 *FeedbackRefErrLow* / 1957 *FeedbackRefErrHigh*

Reference: *↑8 Actuator and feedback*

3059 ErrFeedbackAdjust **3159 SErrFeedbackAdjust**

Cause: Automatic adjustment of the actuator could not be executed, incorrect input in the reference values for the actuator (↑*8.1 Calibrating the actuator*).

Reaction: Control unit cannot be operated.

Action:
- Check the voltage supply and the supply leads to the actuator.
- Check the feedback cable to the actuator.
- Check the actuator, replace if necessary.
- To be able to run an automatic adjustment, the system must be stopped
- Check the reference values and error limits for the feedback
- Set the error limits
  1952 *FeedbackErrorLow* = 0
  1953 *FeedbackErrorHigh* = 65535
  1956 *FeedbackRefErrLow* = 0
  1957 *FeedBackRefErrHigh* = 65535
  for the feedback, save the settings
  Restart control unit and repeat the automatic adjustment.
- Set error limits again

Reference: ↑*8.1.2 Automatic calibration*

3060 ErrAmplifier **3160 SErrAmplifier**

Cause: Overcurrent in the actuator triggering.

Reaction:
- System cannot be started.
- Emergency shutdown.

Action: Check the actuator, replace if necessary.

3075 ErrClearFlash **3175 SErrClearFlash**

Cause: An error has occurred while clearing the control unit's ROM.

Reaction:
- System cannot be started.
- Emergency shutdown.

Action:
- Restart control unit with ↑*3.7 Resetting a control unit*.
- Notify **HEINZMANN**.

Reference: ↑*10.5 Bootloader*

Note: *This error can only be seen in the bootloader.*

3076 ErrParamStore **3176 SErrParamStore**

Cause: An error has occurred while programming the control unit's parameter memory.

Reaction: - System cannot be started.
- Emergency shutdown.

Action: - Restart control unit with ↑*3.7 Resetting a control unit*.
- Notify **HEINZMANN**.

3077 ErrProgramTest **3177 SErrProgramTest**

Cause: The continuous monitoring of the program memory returned an error.

Reaction: - System cannot be started.
- Emergency shutdown.

Action: - Restart control unit with ↑*3.7 Resetting a control unit*.
- Notify **HEINZMANN**.

3078 ErrRAMTest **3178 SErrRAMTest**

Cause: The continuous monitoring of the main memory (RAM) returned an error.

Reaction: - System cannot be started.
- Emergency shutdown.

Action: - Make a note of the settings of parameters 3895 *RAMTestAddrHigh* and 3896 *RAMTestAddrLow*
- Restart control unit with ↑*3.7 Resetting a control unit*.
- Notify **HEINZMANN**.

3081 Err5V_Ref **3181 SErr5V_Ref**

Cause: The 5V sensor reference voltage 3603 *5V_Ref* is not within the permitted range of 4.5 to 5.5 V.

Reaction: - Error message.
- Errors clears itself if the voltage is within the normal range again.

Action: Check the sensor supply.

3085 ErrVoltage **3185 SErrVoltage**

Cause: The supply voltage for the control unit is not within the permitted range of 9 to 32 V.

Reaction: - Error message.
- Errors clears itself if the voltage is within the normal range again.

Action: Check the voltage supply.

3087 ErrMainCheckSum

Cause: The checksum of the control unit program is wrong.

Reaction: Control unit cannot be operated

Action: - Restart control unit with ↑*3.7 Resetting a control unit*.
- Notify **HEINZMANN**.

Reference: ↑*10.5 Bootloader*

Note: This error can only be seen in the bootloader.

3089 ErrWatchdog

Cause: Internal computing error, so-called “Watchdog error”

Reaction: - Control unit cannot be operated
- Emergency shutdown.

Action: - Restart control unit with ↑*3.7 Resetting a control unit*.
- Notify **HEINZMANN**.

Reference: ↑*10.5 Bootloader*

Note: This error can only be seen in the bootloader.

3090 ErrData **3190 SErrData**

Cause: Parameter 3099 *EEPROMErrorCode* provides more information about the type of error:

Bit	Meaning
0	Program identifier in the E²PROM is invalid, the program stays in the bootloader, start is not possible
1	Operating data memory 2 in the E¹PROM is invalid, the operating data are cleared, start is possible
2	Operating data memory 2 in the E²PROM is invalid, the operating data are cleared, start is possible
3	Serial number memory in the E²PROM is invalid, start is possible
4	Error memory in the E²PROM is invalid, the error memory is cleared, start is possible
5	Parameter memory in the E²PROM is invalid, standard parameters of the firmware are transferred, start is not possible
6	Exception memory in the E²PROM is invalid, start is possible

Action: - Check data for correct settings,
- Restart control unit with ↑*3.7 Resetting a control unit*
- Notify **HEINZMANN**.

Note: The error only occurs after control unit reset. Bit 5 is signalled after each program download.

3092 ErrConfiguration **3192 SErrConfiguration**

Cause: Configuration error

Reaction: - System cannot be started.
- Control unit operates with standard parameters

Action: - Check configuration for correct settings,
- Restart control unit with ↑*3.7 Resetting a control unit*

Reference: ↑*10.6 Configuration error*

3093 ErrStack **3193 SErrStack**

Cause: Internal program or computing error, so-called "Stack Overflow" error

Reaction: - System cannot be started.
- Emergency shutdown.

Action: - Make a note of the value of parameters 3897 *CStackTestFreeBytes* and 3898 *IStackTestFreeBytes*
- Notify **HEINZMANN**.
- Restart control unit with ↑*3.7 Resetting a control unit*.

3094ErrIntern **3194 SErrIntern**
3195 SExceptionNumber
3196 SExceptionAddrHigh
3197 SExceptionAddrLow
3198 SExceptionFlag

Cause: Internal program or computing error, a so-called "EXCEPTION" error

Reaction: - System cannot be started.
- Emergency shutdown.

Action: - Make a note of the values of parameters 3195 to 3199 and notify **HEINZMANN**.
- Restart control unit with ↑*3.7 Resetting a control unit*.

11 Parameter Description

11.1 General

All the parameter groups are listed in ↑ *Table 15: Parameter groups*. This provides an overview of the number ranges on which certain functions can be found. The four subsequent parameter tables (↑ *Table 16: Parameter* ↑ *Table 17: Measured Values and Display Values* ↑ *Table 18: Functions* and ↑ *Table 19: Curves and Control Map*) list each individual parameter with a brief description and a reference to related chapters.

These four parameter lists explains all of the parameters which are defined in the positioner. The defined level is given for each parameter. An operating tool such as DcDesk 2000 or a hand programmer can only be used to view parameters whose level is no higher than the level of the tool.

Parameters that require a save and control unit reset after a change are identified by (RESET).

Only the first field parameter is given for curves and control maps; the parameter number is identified by "ff" (and following).

No. Parameter		Measured values		Functions		Curves	
300	Control path/setpoint	2300	Control path/setpoint	4300	Setpoint	7300	Actuator curve
800	Switch functions	2800	Switch functions	4800	Configuration of the input/output channels		
		3000	Current errors	5000	Error handling		
		3100	Error memory				
1500	PWM inputs Analogue inputs	3500	PWM inputs Analogue inputs	5500	Configuration of analogue input/output channels		
1600	PWM outputs Analogue outputs	3600	Internal measured values	5600	Analogue outputs		
1700	Positioner	3700		5700	Positioner		
1800	Status	3800	Status				
1900	Servo loop	3900	Servo loop	5900	Servo loop		
1950	Feedback	3950	Feedback	5950	Feedback		

Table 15: Parameter groups

11.2 List 1: Parameter

No.	Name		Meaning
310	**ActPosSecureMin**		
	Level:	6	Minimum control path to protect the actuator from mechanical and thermal stress (approx. 3 %)
	Range:	0..100 %	
	Page(s):	52	
312	**ActPosSecureMax**		
	Level:	6	Maximum control path to protect the actuator from mechanical and thermal stress (approx. 97 %)
	Range:	0..100 %	
	Page(s):	52	
331	**FuelSetpSubst**		
	Level:	4	Equivalent value for the setpoint generator in case of error
	Range:	0..100 %	
	Page(s):	25	
340	**AnaStopThresholdLow**		
	Level:	6	Definition of the lower threshold for the stop request via analogue input
	Range:	0..100 %	
	Page(s):	24	
341	**AnaStopThresholdHigh**		
	Level:	6	Definition of the upper threshold for the stop request via analogue input
	Range:	0..100 %	
	Page(s):	24	
342	**AnaStopSensorSubst**		
	Level:	6	Equivalent value for the analogue stop sensor in case of error
	Range:	0..100 %	
	Page(s):	26	
810	**FunctStopRequest**		
	Level:	6	Configuration of the stop request via a digital input
	Range:	0..5	
	Page(s):	31	
828	**FunctErrorReset**		
	Level:	6	Configuration of the error reset function via a digital input
	Range:	-5..5	
	Page(s):	31	
845	**FunctAutoAdjust**		
	Level:	6	Configuration of the auto adjust function via a digital input
	Range:	-5..5	
	Page(s):	31	
1500	**FuelSetPWMIn1RefLow**		
	Level:	4	Lower reference value of PWM input 1 for setpoint selection
	Range:	0..100 %	
	Page(s):	35, 36	

No.	Name		Meaning
1501	**FuelSetPWMIn1RefHigh**		
	Level:	4	Upper reference value of PWM input 1 for setpoint selection
	Range:	0..100 %	
	Page(s):	35, 36	
1504	**FuelSetPWMIn3RefLow**		
	Level:	4	Lower reference value of PWM input 3 for setpoint selection
	Range:	0..100 %	
	Page(s):	35	
1505	**FuelSetPWMIn3RefHigh**		
	Level:	4	Upper reference value of PWM input 3 for setpoint selection
	Range:	0..100 %	
	Page(s):	35	
1506	**FuelSetPWMIn1_Filter**		
	Level:	4	Filter for PWM input 1 for setpoint selection
	Range:	1..255	
	Page(s):	35	
1508	**FuelSetPWMIn3_Filter**		
	Level:	4	Filter for PWM input 3 for setpoint selection
	Range:	1..255	
	Page(s):	35	
1510	**AnaStop_Ref_0%**		
	Level:	4	0 % reference value of the analogue input for the stop request
	Range:	0.. 5V or 0..22 mA	
	Page(s):	32	
1511	**AnaStop_Ref_100%**		
	Level:	4	100 % reference value of the analogue input for the stop request
	Range:	0.. 5V or 0..22 mA	
	Page(s):	32	
1512	**AnaStop_ErrorLow**		
	Level:	4	Error limit for minimum value of the analogue input for the stop request
	Range:	0.. 5V or 0..22 mA	
	Page(s):	26, 34	
1513	**AnaStop_ErrorHigh**		
	Level:	4	Error limit for maximum value of the analogue input for the stop request
	Range:	0.. 5V or 0..22 mA	
	Page(s):	26, 34	
1514	**AnaStop_Filter**		
	Level:	4	Filter value of the analogue input for the stop request
	Range:	1..255	
	Page(s):	33	

No.	Name		Meaning
1530	**FuelSetAna_Ref_0%**		
	Level:	4	0 % reference value of the analogue input for the setpoint selection
	Range:	0.. 5V or 0..22 mA	
	Page(s):	32	
1531	**FuelSetAna_Ref_100%**		
	Level:	4	100 % reference value of the analogue input for the setpoint selection
	Range:	0.. 5V or 0..22 mA	
	Page(s):	32	
1532	**FuelSetAna_ErrorLow**		
	Level:	4	Error limit for minimum value of the analogue input for the setpoint selection
	Range:	0.. 5V or 0..22 mA	
	Page(s):	25, 34	
1533	**FuelSetAna_ErrorHigh**		
	Level:	4	Error limit for maximum value of the analogue input for the setpoint selection
	Range:	0.. 5V or 0..22 mA	
	Page(s):	25, 34	
1534	**FuelSetAna_Filter**		
	Level:	4	Filter value of the analogue input for the setpoint selection
	Range:	1..255	
	Page(s):	33	
1601	**PWMOut_RefLow**		
	Level:	4	Minimum value of the PWM output for transmitting the actuator actual-value
	Range:	0..100 %	
	Page(s):	37	
1602	**PWMOut_RefHigh**		
	Level:	4	Maximum value of the PWM output for transmitting the actuator actual-value
	Range:	0..100 %	
	Page(s):	37	
1625	**PWMOutFrequency**		
	Level:	4	Frequency of the PWM output for transmitting the actuator actual-value
	Range:	0..100 %	
	Page(s):	37	
1646	**CurrentOut_ RefLow**		
	Level:	4	Minimum value of the analogue output for transmitting the actuator actual-value
	Range:	0..22 mA	
	Page(s):	36	
1647	**CurrentOut_RefHigh**		
	Level:	4	Maximum value of the analogue output for transmitting the actuator actual-value
	Range:	0..22 mA	
	Page(s):	36	

No.	Name		Meaning
1700	**ActuatorTestSetpoint**		
	Level:	2	Setpoint for the control path in test mode for setting and testing the actuator
	Range:	0..100 %	
	Page(s):	54	
1701	**ActuatorTstAmplitude**		
	Level:	2	Amplitude of the control path step generator in test mode for setting and testing the actuator
	Range:	0..100 %	
	Page(s):	54	
1702	**ActuatorTstFrequency**		
	Level:	2	Frequency setting for test mode for setting and testing the actuator
	Range:	0..100 %	
	Page(s):	54	
1800	**Level**		
	Level:	1	User level
	Range:	1..7	Maximum level of the dongle or hand programmer, can be reduced
	Page(s):	13	
1876	**ValueStep**		*for hand programmers only*
	Level:	2	Increment for value changes
	Range:	0..65535	
1900	**FeedbackAdjustTime**		
	Level:	6	Position hold time with the auto adjust
	Range:	0..100 s	
	Page(s):	50	
1905	**ServoCorrFactor**		
	Level:	6	Correction factor of the PID value for servo loop
	Range:	0..400 %	
	Page(s):	52	
1906	**ServoCorrRange**		
	Level:	6	Position range for correction factor
	Range:	0..50 %	
	Page(s):	52	
1911	**ServoGain**		
	Level:	6	Proportional factor for the servo loop
	Range:	0..100 %	
	Page(s):	11, 52	
1912	**ServoStability**		
	Level:	6	Integral factor for the servo loop
	Range:	0..100 %	
	Page(s):	52	
1913	**ServoDerivative**		
	Level:	6	Differential factor for the servo loop
	Range:	0..100 %	
	Page(s):	52	

No.	Name		Meaning
1914	**ServoAcceleration**		
	Level:	6	DD factor for the servo loop
	Range:	0..100 %	
	Page(s):	52	
1915	**ServoCurrentOffDelay**		
	Level:	6	Delay time for the start of the current shutoff of the actuator when no more setpoints are selected or a stop request is present
	Range:	0..100 s	
	Page(s):	22	
1916	**ServoCurrentRedDelay**		
	Level:	6	Delay time for the start of current reduction of the actuator
	Range:	0..100 s	
	Page(s):	53	
1917	**ServoCurrentMax**		
	Level:	6	Maximum current for the actuator (during motion)
	Range:	0..12.5 A	
	Page(s):	53	
1918	**ServoCurrentRed**		
	Level:	6	Reduced current for the static state of the actuator
	Range:	0..12.5 A	
	Page(s):	53	
1919	**ServoCurrentAdjust**		
	Level:	6	Current during automatic adjustment of the actuator
	Range:	0..12.5 A	
	Page(s):	50	
1920	**ServoCurrentPC**		
	Level:	6	Current input from the PC
	Range:	-12.5..12.5 A	
	Page(s):		
1950	**FeedbackRef_0%**		
	Level:	4	0 % reference value for the feedback
	Range:	0..65535	(RESET)
	Page(s):	49	
1951	**FeedbackRef_100%**		
	Level:	4	100 % reference value for the feedback
	Range:	0..65535	(RESET)
	Page(s):	49	
1952	**FeedbackErrLow**		
	Level:	4	Lower error value for the feedback
	Range:	0..65535	
	Page(s):	51, 65	
1953	**FeedbackErrHigh**		
	Level:	4	Upper error value for the feedback
	Range:	0..65535	
	Page(s):	51, 65	

No.	Name		Meaning
1955	**FeedbackReference**		
	Level:	4	Reference value for the reference coil
	Range:	0..65535	(RESET)
	Page(s):	50	
1956	**FeedbackRefErrLow**		
	Level:	4	Lower error value for the reference coil
	Range:	0..65535	
	Page(s):	51, 65	
1957	**FeedbackRefErrHigh**		
	Level:	4	Higher error value for the reference coil
	Range:	0..65535	
	Page(s):	51, 65	

Table 16: Parameter

11.3 List 2: Measured values

	Name		Meaning
2300	**ActPos**		
	Level:	1	Current actuator position
	Range:	0..100 %	
	Page(s):	50, 52, 54	
2330	**ActPosSetpoint**		
	Level:	1	Control path setpoint for the actuator
	Range:	0..100 %	
	Page(s):	20, 52	
2350	**FuelSetpoint**		
	Level:	1	Setpoint
	Range:	0..100 %	
	Page(s):	20, 22, 32, 52	
2809	**AnaStopRequest**		
	Level:	1	State of the stop request via analogue input
	Range:	0/1	
	Page(s):	23, 26	
2810	**SwitchStopRequest**		
	Level:	1	State of the stop request via digital input
	Range:	0/1	
	Page(s):	23	
2828	**SwitchErrorReset**		
	Level:	1	State of the switch function "Error reset"
	Range:	0/1	
	Page(s):	31, 58	
2845	**SwitchAutoAdjust**		
	Level:	1	State of the switch function "Automatic actuator adjustment"
	Range:	0/1	
	Page(s):	31, 50	
2900	**AnaStopSensor**		
	Level:	1	Current measured value of the analogue stop input
	Range:	0..100 %	
	Page(s):	24, 26, 32	
3000	**ConfigurationError**		
	Level:	1	Displays configuration errors
	Range:	0..65535	
	Page(s):	62, 64	
3005	**ErrAnaStopRequest**		
	Level:	1	Error display of the analogue stop input
	Range:	0/1	
	Page(s):	26, 64	

3007	**ErrFuelSetpAna**			
	Level:	1	Error display of the analogue setpoint generator
	Range:	0/1	
	Page(s):	25, 64	
3009	**ErrFuelSetpPWMIn1**		
	Level:	1	Error display of the PWM setpoint generator at PWM input 1
	Range:	0/1	
	Page(s):	25, 64	
3011	**ErrFuelSetpPWMIn3**		
	Level:	1	Error display of the PWM setpoint generator at PWM input 3
	Range:	0/1	
	Page(s):	25, 64	
3050	**ErrFeedback**		
	Level:	1	Error in the measured value of the actuator feedback
	Range:	0/1	
	Page(s):	63, 64	
3053	**ErrActuatorDiff**		
	Level:	1	Excessive difference between control path setpoint and actual-value
	Range:	0/1	
	Page(s):	65	
3056	**ErrFeedbackRef**		
	Level:	1	Error in the reference value of the actuator feedback
	Range:	0/1	
	Page(s):	63, 65	
3059	**ErrFeedbackAdjust**		
	Level:	1	Error in the auto adjust or when entering the reference values of the feedback
	Range:	0/1	
	Page(s):	50, 63, 66	
3060	**ErrAmplifier**		
	Level:	1	Amplifier error
	Range:	0/1	
	Page(s):	63, 66	
3075	**ErrClearFlash**		
	Level:	1	Error display when clearing the read-only memory (displayed in the bootloader)
	Range:	0/1	
	Page(s):	27, 63, 66	
3076	**ErrParamStore**		
	Level:	1	Error display when programming the read-only memory
	Range:	0/1	
	Page(s):	27, 63, 67	
3077	**ErrProgramTest**		
	Level:	1	Error display when checking the checksum of the firmware program
	Range:	0/1	
	Page(s):	27, 63, 67	

3078	**ErrRAMTest**			
	Level:	1	Error display of the RAM test	
	Range:	0/1		
	Page(s):	27, 63, 67		
3081	**Err5V_Ref**			
	Level:	1	Error display of the 5 V sensor supply	
	Range:	0/1		
	Page(s):	27, 67		
3085	**ErrVoltage**			
	Level:	1	Error display of the voltage supply	
	Range:	0/1		
	Page(s):	27, 68		
3087	**ErrMainCheckSum**			
	Level:	1	Error display of the data record	
	Range:	0/1	(displayed in the bootloader)	
	Page(s):	61, 68		
3089	**ErrWatchdog**			
	Level:	1	Error display of the data record	
	Range:	0/1	(displayed in the bootloader)	
	Page(s):	27, 61, 68		
3090	**ErrData**			
	Level:	1	Error display of the data record	
	Range:	0/1		
	Page(s):	27, 63, 68		
3092	**ErrConfiguration**			
	Level:	1	Configuration error	
	Range:	0/1		
	Page(s):	62, 69		
3093	**ErrStack**			
	Level:	1	Error display of the “Stack Overflow” error	
	Range:	0/1		
	Page(s):	27, 63, 70		
3094	**ErrIntern**			
	Level:	1	Error display for internal software error	
	Range:	0/1		
	Page(s):	27, 63, 70		
3099	**EEPROMErrorCode**			
	Level:	1	Display of incorrect E^2PROM pages	
	Range:	0000..FFFF Hex		
	Page(s):	68		
3105	**SErr...**			
to	Level:	1	Error counter for current error starting from 3005 Err...	
3194	Range:	0..255		
	Page(s):	60, 70		

3195 to 3198	**SExceptionNumber**		
	Level:	1	Identifiers for the internal software error 3094
	Range:	0..FFFF Hex	
	Page(s):	28, 70	
3500	**FuelSetpPWMIn1**		
	Level:	1	Current value of PWM input 1 for setpoint selection
	Range:	0..100 %	
	Page(s):	26, 35	
3501	**FuelSetpFrequencyIn1**		
	Level:	1	Current frequency at PWM input 1 for setpoint selection
	Range:	0..1000 Hz	
	Page(s):	26, 35	
3504	**FuelSetpPWMIn3**		
	Level:	1	Current value of PWM input 3 for setpoint selection
	Range:	0..100 %	
	Page(s):	26	
3505	**FuelSetpFrequencyIn3**		
	Level:	1	Current frequency at PWM input 3 for setpoint selection
	Range:	0..1000 Hz	
	Page(s):	26	
3510	**AnaStop_Percent**		
	Level:	1	Scaled value of the analogue input for the stop request
	Range:	0..100 %	
	Page(s):	32	
3511	**AnaStop_Value**		
	Level:	1	Unscaled value of the analogue input for the stop request
	Range:	0..5 V or 0..22 mA	
	Page(s):	26, 32	
3530	**FuelSetpAna_Percent**		
	Level:	1	Scaled value of the analogue input for the setpoint selection
	Range:	0..100 %	
	Page(s):	32	
3531	**FuelSetpAna_Value**		
	Level:	1	Unscaled value of the analogue input for the setpoint selection
	Range:	0..5 V or 0..22 mA	
	Page(s):	25, 32	
3600	**PowerSupply**		
	Level:	1	Current value of the filtered supply voltage
	Range:	0..55 V	
	Page(s):	28	
3601	**LowPowerEnabled**		
	Level:	1	1: Undervoltage is supported by the control unit and is requested by parameter 5600 *LowPowerEnable*
	Range:	0..1	
	Page(s):	29	

3602	**PowerSupplyRaw**			
	Level:	1	Current measured value of supply voltage x:
	Range:	0..55 V	
	Page(s):	28	
3603	**5V_Ref**		
	Level:	1	Current value of the 5V reference voltage
	Range:	0..10 V	
	Page(s):	67	
3800	**EmergencyAlarm**		
	Level:	1	Display of the emergency alarm
	Range:	0/1	
	Page(s):	59, 63	
3801	**CommonAlarm**		
	Level:	1	Display of the common alarm
	Range:	0/1	
	Page(s):	25, 26, 37, 59	
3802	**SystemStopRequest**		
	Level:	1	Shows that the system is stopped by an internally or
	Range:	0/1	externally present stop (1 = stop request is active)
	Page(s):	22	
3803	**SystemStopped**		
	Level:	1	1 = shows that the system is stopped
	Range:	0/1	
	Page(s):	22	
3830	**Phase**		
	Level:	1	Current phase of the system
	Range:	0..9	0: Waiting for start
	Page(s):	22	4: Setpoint selection active
			8: Auto adjust
			9: Positioner operation
3840	**HardwareVersion**		
	Level:	1	Version number of the control unit hardware
	Range:	00.00..99.99	
	Page(s):	56	
3841	**AddHardwareVersion**		
	Level:	1	Additional version number of the control unit hardware
	Range:	00.00..99.99	
	Page(s):	56	
3842	**SoftwareVersion**		
	Level:	1	Version number of the software (firmware)
	Range:	00.0.00...64.9.99	2 digits customer number
		or 0000.00.00...6552.99.99	1 digit variant
	Page(s):	8, 56	2 digits change index

3843	**BootSoftwareVersion**		
	Level:	1	Version number of the bootloader software
	Range:	65.0.00...65.5.35	
	Page(s):	56	
3844	**SerialDate**		
	Level:	1	Serial date of the control unit hardware
	Range:	0000..9912	
	Page(s):	56	
3845	**SerialNumber**		
	Level:	1	Serial number of the control unit hardware
	Range:	00000..65535	
	Page(s):	56	
3850	**Identifier**		
	Level:	1	Identification number of the PC program \ hand programmer
	Range:	0..65535	
	Page(s):	57	
3851	**LastIdentifier**		
	Level:	1	Identification number of the PC program \ hand programmer of the last stored parameter change
	Range:	0..65535	
	Page(s):	57	
3865	**CalculationTime**		
	Level:	1	Necessary calculation time of the main loop
	Range:	0..16.384 ms	
	Page(s):	28	
3870	**Timer**		
	Level:	1	Internal millisecond timer
	Range:	0..65.535 s	
	Page(s):	28	
3871	**OperatingHourMeter**		
	Level:	1	Number of operating hours of the running system
	Range:	0..65535 h	
	Page(s):	60	
3872	**OperatingSecondMeter**		
	Level:	1	Seconds of the running system until carryover to an operating hour
	Range:	0..3599 s	
	Page(s):	60	
3895	**RAMTestAddr**		
	Level:	6	Current tested memory address
	Range:	0000..FFFF Hex	
	Page(s):	27, 67	
3896	**RAMTestPattern**		
	Level:	6	Test value
	Range:	0000..FFFF Hex	
	Page(s):	27	

3897	**CStackTestFreeBytes**		
3898	**IStackTestFreeBytes**		
	Level:	6	Displays the free bytes in the stack memory
	Range:	0000..FFFF Hex	
	Page(s):	70	
3905	**ServoPIDCorr**		
	Level:	6	Correction factor for the PID parameters of the servo loop
	Range:	0..400 %	
	Page(s):	53	
3916	**ServoCurrentSetpoint**		
	Level:	1	Setpoint for the current through the actuator
	Range:	-12.5..12.5 A	
	Page(s):	53	
3950	**Feedback**		
	Level:	1	Unscaled value of the feedback
	Range:	0..65535	
	Page(s):	49	
3955	**FeedbackReference**		
	Level:	1	Unscaled value of the reference coil
	Range:	0..65535	
	Page(s):	50	
3960	**FeedbackCorrection**		
	Level:	1	Value of the feedback corrected by the reference
	Range:	0..65535	
	Page(s):	50	

Table 17: Measured Values and Display Values

11.4 List 3: Functions

	Name		Meaning
4330	**ChanTyp_FuelSetp**		
	Level:	6	Configuration of the input channel type of the setpoint generator
	Range:	0..10	0 = analogue
	Page(s):	17, 19, 30, 32	1 = PWM (RESET)
4331	**SubstOrLastFuelSetp**		
	Level:	4	Selection of the equivalent value for the setpoint generator in case of error (0 = last valid value,
	Range:	0/1	1 = equivalent value 331 *FuelSetpSubst*)
	Page(s):	25	(RESET)
4332	**HoldOrResetFuelSetp**		
	Level:	4	Selects whether the error at the setpoint generator is reset or held when the signal returns
	Range:	0/1	(0 = error cleared, 1 = error retained)
	Page(s):	25	(RESET)
4335	**ActPosOutPWMOrAnalog**		
	Level:	4	Configuration of the output channel type for the display of the current actuator position
	Range:	0/1	0 = analogue
	Page(s):	36	1 = PWM (RESET)
4340	**AnaStopRequestUsed**		
	Level:	6	Activates the analogue stop request
	Range:	0/1	(RESET)
	Page(s):	24, 30, 32	
4341	**AnaStopSubstOrLast**		
	Level:	6	Selects the equivalent value for the analogue stop request in case of error (0 = last valid value,
	Range:	0/1	1 = equivalent value 1000 *AnaStopSensorSubst*)
	Page(s):	26	(RESET)
4342	**AnaStopHoldOrReset**		
	Level:	6	Selects whether the error at the analogue stop request is reset or held when the signal returns
	Range:	0/1	(0 = error cleared, 1 = error retained)
	Page(s):	26	(RESET)
4721	**FuelToActPosCurveOn**		
	Level:	4	Activates the filling-dependent actuator nominal position (curve)
	Range:	0/1	
	Page(s):	20	(RESET)

	Name		Meaning
4810	**DigStopImplsOrSwitch**		
	Level:	6	Activates the digital stop request as an impulse (1) or as
	Range:	0/1	a switch (0)
	Page(s):	23	(RESET)
4811	**DigStopOpenOrClose**		
	Level:	6	Activates the digital stop request as an NC (1) or NO
	Range:	0/1	(0) contact
	Page(s):	23	(RESET)
5000	**CommAlarmLowActive**		
	Level:	2	Selects whether the common alarm display will be low
	Range:	0/1	active (0 when at least one error is present) or high
	Page(s):	37	active (1 when at least one error is present)
5100	**NoStoreSErrOn**		
	Level:	6	Suppresses the storage of errors in the error memory
	Range:	0/1	until the control unit is restarted
	Page(s):	60	
5102	**CommonAlarmResetOn**		
	Level:	2	Selects whether the common alarm display will be
	Range:	0/1	briefly reset (edge change) when a new error is added
	Page(s):	37, 59	
5103	**CommonAlarmResetBoth**		
	Level:	2	Selects whether the edge change (5102
	Range:	0/1	*CommonAlarmResetOn* = 1) is also generated when an
	Page(s):	38, 59	error disappears (generally with every error change)
5510	**AnaStopCurrOrVolt**		
	Level:	2	Selects the signal type of the analogue input for the stop
	Range:	1..2	request
	Page(s):	24	1: 0..5 V
			2: 0..22.7 mA (RESET)
5530	**FuelSetpCurrOrVolt**		
	Level:	2	Selects the signal type of the analogue input for the
	Range:	1..2	setpoint generator
	Page(s):	19	1: 0..5 V
			2: 0..22.7 mA (RESET)
5600	**LowPowerEnable**		
	Level:	2	Activates the enabling of undervoltage
	Range:	0/1	
	Page(s):	28	
5700	**ActuatorTestOn**		
	Level:	2	Activates the test mode for the actuator
	Range:	0/1	This function cannot be stored
	Page(s):	54	

	Name		Meaning
5701	**ActuatorTestMode**		
	Level:	2	Selects the test mode for the actuator
	Range:	0..3	0 = input from 1700
	Page(s):	54	1 = rectangular from 1700 ± 1701
			2 = delta from 1700 ± 1701
			3 = sinusoidal from 1700 ± 1701
5910	**ActuatorOn**		
	Level:	6	Switches the servo loop on/off
	Range:	0/1	
	Page(s):	48, 54	
5911	**Amplifier2QOr4Q**		
	Level:	6	Operation of the amplifier
	Range:	0/1	0 = 4-quadrant (energizing in both directions)
	Page(s):	48	1 = 2-quadrant (energizing in direction 100 %) (RESET)
5920	**ServoCurrentPCOn**		
	Level:	6	Activates the actuator test mode to output the current from 1920 *ServoCurrentPC* as a test setpoint to the actuator
	Range:	0/1	
	Page(s):		This function cannot be stored
5950	**FeedbDigitalOrAnalog**		
	Level:	6	Type of actuator feedback
	Range:	0/1	0 = DC voltage signal
	Page(s):	48	1 = coil feedback (RESET)
5951	**FeedbSlopeFallOrRise**		
	Level:	6	Type of feedback signal curve
	Range:	0/1	0 = rising output signal with rising filling
	Page(s):	48	1 = falling output signal with rising filling (RESET)

Table 18: Functions

11.5 List 4: Curves and Control Maps

	Name		Meaning
7300ff	**FuelToActSp:f(x)**		*Position curve*
	Level:	6	Filling values for the filling-dependent actuator position curve
	Range:	0..100 %	
	Page(s):	20	(RESET)
7315ff	**FuelToActSp:Pos(x)**		
	Level:	6	Position for the filling-dependent actuator position curve
	Range:	0..100 %	
	Page(s):	20	(RESET)

Table 19: Curves and Control Maps

12 Figures

Fig. 1 Block Diagram 14

Fig. 2 Terminal connection diagram 1 - Analogue setpoint, stop signal digital and/or analogue, actuator with disc rotor motor 42

Fig. 3 Terminal connection diagram 2 Analogue setpoint, stop signal digital and/or analogue, gearless actuator 43

Fig. 4 Terminal connection diagram 3 PWM setpoint, stop signal digital, actuator with disc rotor motor 44

Fig. 5 Terminal connection diagram 4 PWM setpoint, stop signal digital, gearless actuator 45

Fig. 6 Terminal connection diagram 3 PWM setpoint, stop signal analogue, actuator with disc rotor motor 46

Fig. 7 Terminal connection diagram 4 PWM setpoint, stop signal analogue, gearless actuator 47

Fig. 8 Current reduction under static load 53

13 Tables

Table 1 Basic firmware variants ... 9
Table 2 PANDAROS Basic Systems ... 10
Table 3 Product Overview Sensors ... 10
Table 4 Communication programs ... 11
Table 5: Setpoint sources ... 19
Table 6: Setpoint type ... 19
Table 7: Digital Inputs ... 30
Table 8: Switch functions ... 31
Table 9: Switch function assignment ... 31
Table 10: Analogue inputs ... 32
Table 11: PWM Inputs ... 35
Table 12: Analogue output ... 36
Table 13: PWM output ... 37
Table 14: Alarm output ... 37
Table 15: Parameter groups ... 71
Table 16: Parameter ... 77
Table 17: Measured Values and Display Values ... 84
Table 18: Functions ... 87
Table 19: Curves and Control Maps ... 88

14 Index

Activating functions 14
Actuator 44, 61, 62
  2-quadrant operation 44
  4-quadrant operation 44
  Analogue feedback 44
  Calibration 45
  Current 49
  Digital feedback 44
  Maximum current 49
  Positioning mode 50
  Reference values 45
Actuator limiting
  absolute 48
Actuator type 49
Alarm outputs 33
Analogue inputs
  Calibration 28
  Error detection 29
  Filtering 29, 31
  Value range 32
Auto calibration 46
Block Diagram 9
Bootloader 56
Calibrating the actuator 45
  automatic 46
  manual 45
Common alarm 55
Configuration
  PWM inputs 31
Control unit
  Reset 14
Control Unit
  Identification 52
  Serial number 52
Conventions 7
Current errors 55
Data Management 52
Data saving 12
DcDesk 2000 .... 7, 12
Digital outputs
  Alarm outputs 33
Electronics monitoring 23
Emergency Alarm 55
Emergency shutdown errors 59
Error handling 54
  Clear error memory 55
  Common alarm 55
  Configuration errors 58
  Emergency Alarm 55
  Emergency shutdown 59
  Enhanced error memory 56
  Error display 55
  Error memory 55
  Error parameter list 60
  Error types 54
Exception 66
Fatal errors 59
Feedback 44, 61
  Error detection 47
  Error limits 47
  Reference values 45
Firmware 4
  Basic software 5
  Custom 5
  Versions 5
Flash ROM 4
Functions
  Activating 14

Identification
Communication program.................................. 53
Control unit 52
Serial number 52
Levels.. 8
Limiting functions
Actuator protection.. 48
Microprocessor 4
Monitoring
Program flow... 24
Program memory... 24
RAM memory 23
Stack depth 24
Voltage references... 23
Monitoring functions
Electronics... 23
Motor shutdown 59
Motor stop request 19
ORION
Pushbuttons 34
PANDAROS
Basic systems 6
Control unit data.. 35
Pushbuttons 34
Parameter
Levels 8
Parameterization
DcDesk 2000... 11
End-of-line 11
EOL... 11
Hand programmer 11
Presetting... 11
Parameters... 8
Control map list... 84
Curve list... 84
Function list.. 81
Measured-value list 74
Overview table 67
Parameter list... 67
Saving 12
Value range 13
Positioner 50
Program flow
Monitoring 24
Program memory
Monitoring 24
Publications... 6
Pushbuttons.. 34
PWM inputs
Configuration 31
Error detection... 32
PWM outputs
Value range 33
RAM memory
Monitoring 23
Reset 14
ROM 4
Safety instructions.. 1
Normal operation 2
Servicing and maintenance................................ 2
Sensors.. 6, 15
Input type 15
Sensor errors 60
Type 15
Serial Number 52
Servo loop... 48
Current 49
Stability... 48
Software.. 4
Version number... 4

Versions 5
Stack depth
Monitoring... 24
Switch functions
Motor stop 19
Switch Functions
Allocation.. 27
System description 4
Technical data 35
PANDAROS 35
Value range 13
Voltage references
Monitoring of the 23

15 Download of Manuals

Technical manuals can be downloaded in pdf-format from our homepage:

www.heinzmann.com

If the desired manual should not be available there, send an e-mail to:

info@heinzmann.de

or write to:

HEINZMANN GmbH & Co. KG

Technische Redaktion

Am Haselbach 1

D-79677 Schönau/Germany

Please include the following information:

- your name
- name and address of your company
- e-mail or address where you want the manuals to be sent to (if different from above)
- item code (as on front page, bottom right) and title of the desired manual
  or alternatively the technical data of your HEINZMANN equipment

We would be pleased to receive your comments about the contents and presentation of our publications. Send your comments to the e-mail or address shown above please.