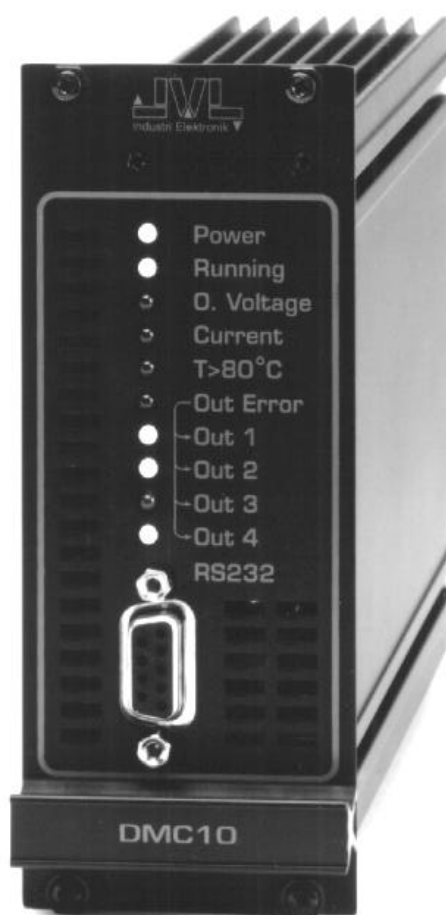


DMC10

DC-Servomotor Controller

User Manual



JVL Industri Elektronik A/S - May 2001

Contents

1	Introduction	4
1.1	Introduction	6
1.2	Overview of Controller Operation Modes	7
1.3	Getting Started — Step-pulse (Mode 1)	8
1.4	Getting Started — Positioning (Mode 2)	9
1.5	Getting Started — Register Mode (Mode 3)	10
1.6	Getting Started — Velocity Mode (Mode 4)	11
1.7	Getting Started — Torque Mode (Mode 5)	12
2	Hardware	14
2.1	Connections	15
2.2	Motor Output	16
2.3	User Inputs	17
2.4	User Outputs	18
2.5	Encoder Input	19
2.6	Power Supply	21
2.7	Step-pulse and Direction Input	22
2.8	Analog Input	23
2.9	Power Dump Output	24
2.10	RS232 Interface	25
2.11	RS485 Interface	28
2.12	Adjustment of Motor Current	29
2.13	Adjustment of Servo Regulation	30
3	Software	32
3.1	Use of RS232 Commands	33
3.2	Step-pulse Mode (MO=1)	34
3.3	Positioning Mode (MO=2)	35
3.4	Register Mode (MO=3)	36
3.5	Velocity Mode (MO=4)	41
3.6	Torque Mode (MO=5)	42
3.7	Zero-point Seek Function	43
3.8	Adjustment of Analog Input	45
3.9	Commands	46
3.10	Error Messages	78
3.11	Alphabetical Command Overview	82
4	Appendix	84
4.1	Specifications	85
4.2	Physical Dimensions	86
4.3	Servo loop	87
4.4	Error Indication	88
4.5	Common Errors	89
4.6	Connector Board	90
4.7	Examples of Motor Connection	91

Copyright 1997-2001, JVL Industri Elektronik A/S.

All rights reserved.

This user manual must not be reproduced in any form without prior written permission of JVL Industri Elektronik A/S.

JVL Industri Elektronik A/S reserves the right to make changes to information contained in this manual without prior notice.

Similarly JVL Industri Elektronik A/S assumes no liability for printing errors or other omissions or discrepancies in this user manual.

This user manual is valid for firmware versions up to 2.8.

MotoWare is a registered trademark

JVL Industri Elektronik A/S

Blokken 42

DK-3460 Birkerød

Denmark

Tel. +45 45 82 44 40

Fax. +45 45 82 55 50

e-mail: jvl@jvl.dk

Internet: <http://www.jvl.dk>





DMC10 is a compact, programmable DC servo motor controller.

The DMC10 features control via either the built-in RS232 interface, or via an analog input ($\pm 10V$)

In addition, the DMC10 can be controlled as a step motor system controller via a step-pulse and direction input.

The DMC10 can be configured for absolute/relative positioning via 4 digital inputs.

DMC10 accepts a balanced or unbalanced signal from a standard 2-channel incremental encoder.

All inputs and outputs are optically isolated and overload protected.

The DMC10 has 4 general-purpose outputs. These can be used to provide a control signal, for example when the motor has reached a required position, or an error signal if a condition arises that prevents motor operation.

The DMC10 can be mounted in a 19" rack system or surface mounted.

Main features:

- Digital servo control loop
- Extremely precise positioning
- Small physical dimensions (128x171x50mm)
- Current: 12A continuous, 25A peak
- Current overload protection
- Absolute/Relative positioning
- EMC protected - CE approved
- Voltage overload protection
- Following input features:
 - Analog $\pm 10V$
 - Step-pulse and direction
 - Digital selection of positioning
 - RS232 Interface
- Windows program available for set-up
- End-of-travel inputs
- RS232 and RS485 Interface
- Set-up stored in EEPROM
- Can handle motors up to 1kW
- Pre-programmed Velocity Profiles
- Automatic zero-point search function
- Short-circuit and thermal protection
- Any DC-motor can be used

1.2 Overview of Controller Operation Modes

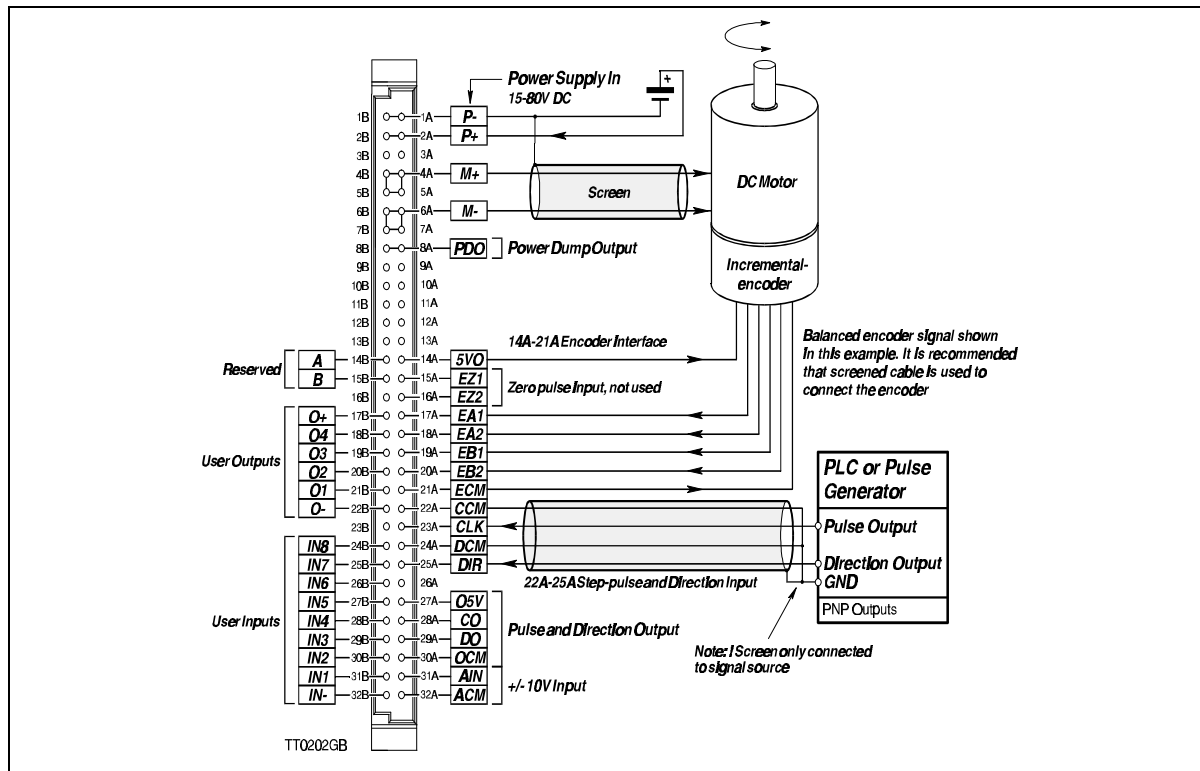
1.2.1 Basic Modes of Controller Operation

The DMC10 Controller includes many individual features for general use in a variety of application configurations. However the Controller is operated in one of five basic modes which are selected using the Mode command (MO). The DMC10's basic modes of operation are:

1. Step-pulse Mode
In Step-pulse Mode, the Controller functions as a step motor system. The motor will move one step each time a voltage pulse is applied to the Controller pulse input. Velocity and acceleration/deceleration are determined by the external frequency.
2. Positioning Mode
In Positioning Mode, the Controller positions the motor via commands transmitted over the RS232 interface. Various operating parameters can be continuously regulated via the RS232 interface while the motor is running.
This mode can be used primarily when the Controller is part of a system which is permanently connected to a PC via the RS232 interface. In addition, it is recommended that this mode is used during installation and commissioning of systems.
3. Register Mode
In this mode, the Controller's set of parameter registers (X0-X15) store the positions and velocities etc. required by the actual system. These registers can be addressed via the Controller's digital inputs and are activated by activating a start input. This mode of operation is especially powerful since it enables the Controller itself to take care of the entire positioning sequence.
4. Velocity Mode
In this mode, the Controller controls the motor velocity via the analogue input. This mode is typically used for simple applications or applications in which another device, such as a PC card, is used for overall control of velocity and position.
5. Torque Mode
In Torque Mode, the Controller controls the motor torque via the analog input. Typical applications which use this mode of operation include, for example, spooling or tensioning foil, cable, etc. No encoder is used in this mode.

The individual modes of operation are illustrated further in the following pages. These pages provide a quick guide to setting up a functional system. For more detailed documentation of operating modes, the individual inputs and outputs and the Controller command set are described in Chapters 2 and 3

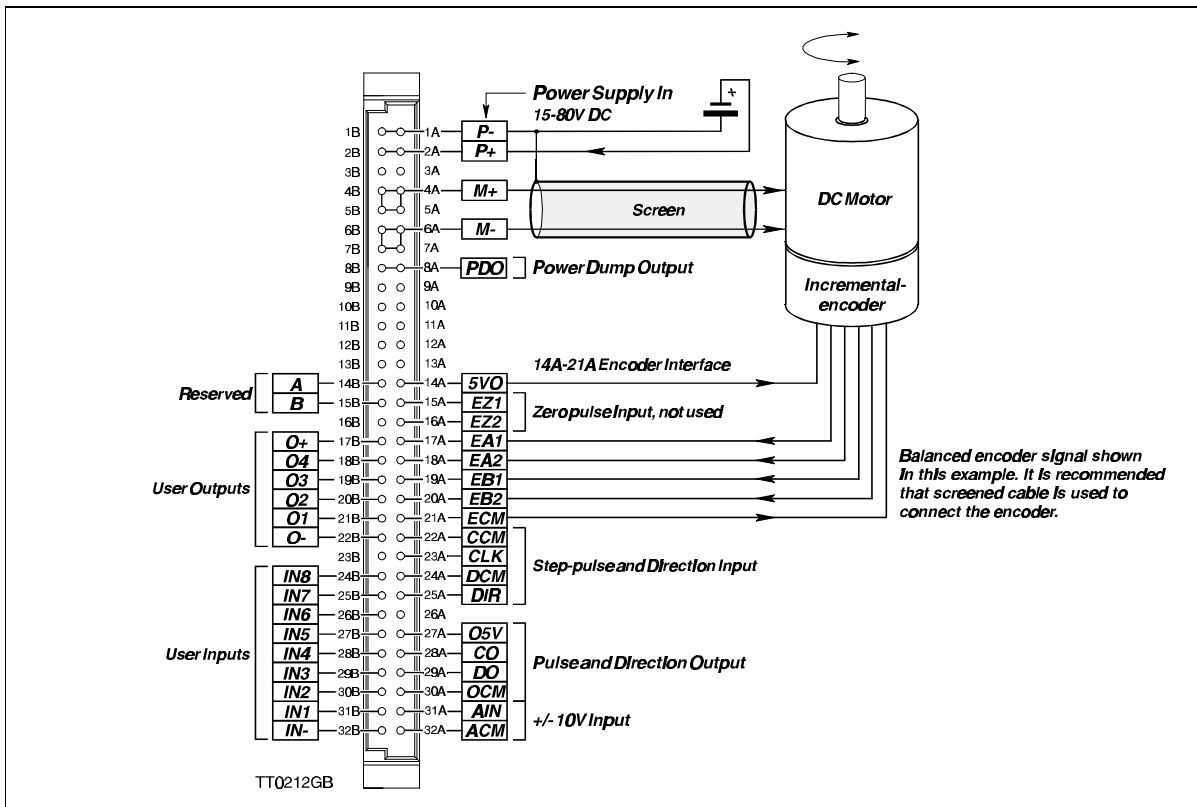
1.3 Getting Started — Step-pulse (Mode 1)



Follow the procedure below for operation of the DMC10 in Mode 1 (Step-pulse and Direction Mode)

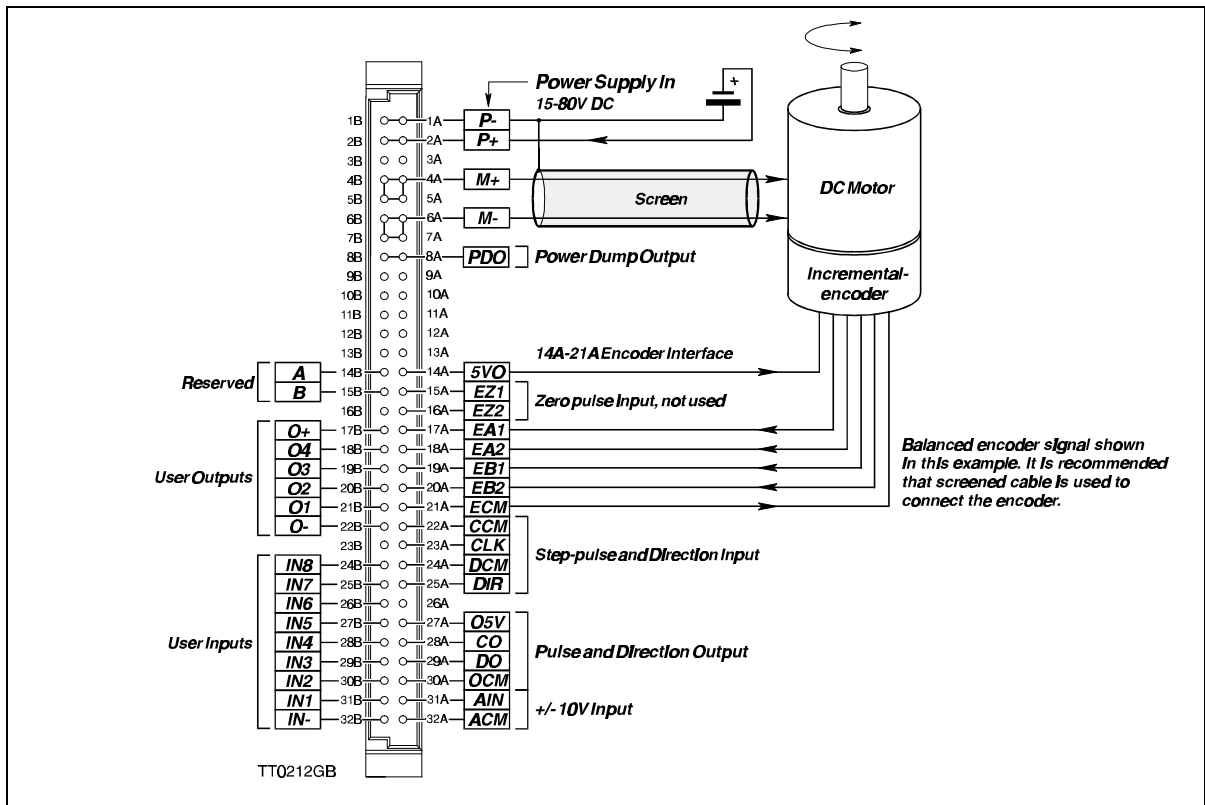
1. Connect the Controller as shown above. For further details, see also the following sections : *Motor Output* page 16 / *Encoder Input* page 19 / *Step-pulse and Direction Input* page 22.
2. Connect the PC via a terminal program (e.g. JVL's *MotoWare* or Windows HyperTerminal). If necessary following the description of the RS232 interface in *RS232 Interface* page 25.
3. Switch on the Controller but ensure that no pulses are applied to the step-pulse input. Only the *Power LED* and possibly *Out 1* may be active. If one or more of the red LEDs is active or blinks, either the motor or encoder is probably connected incorrectly.
4. Send the command ? (enter) to the Controller and wait until the Controller responds with a status overview.
If the status overview is displayed, the RS232 interface and power supply are correctly connected.
5. Set the Controller to Step-pulse and Direction Mode by sending the command *MO=1* (enter).
The Controller should respond *Y*, indicating that Mode 1 has been selected.
6. The motor current must be adjusted for the actual motor used. See *Adjustment of Motor Current* page 29.
7. Servo parameters *KD*, *KP*, and *KI* are set to typical values by default. This means that the motor can be operated without further adjustment. For optimum system operation however, the parameters should be adjusted. See *Adjustment of Servo Regulation* page 30.
8. The Controller is now set to Step-pulse and Direction Mode.

1.4 Getting Started — Positioning (Mode 2)



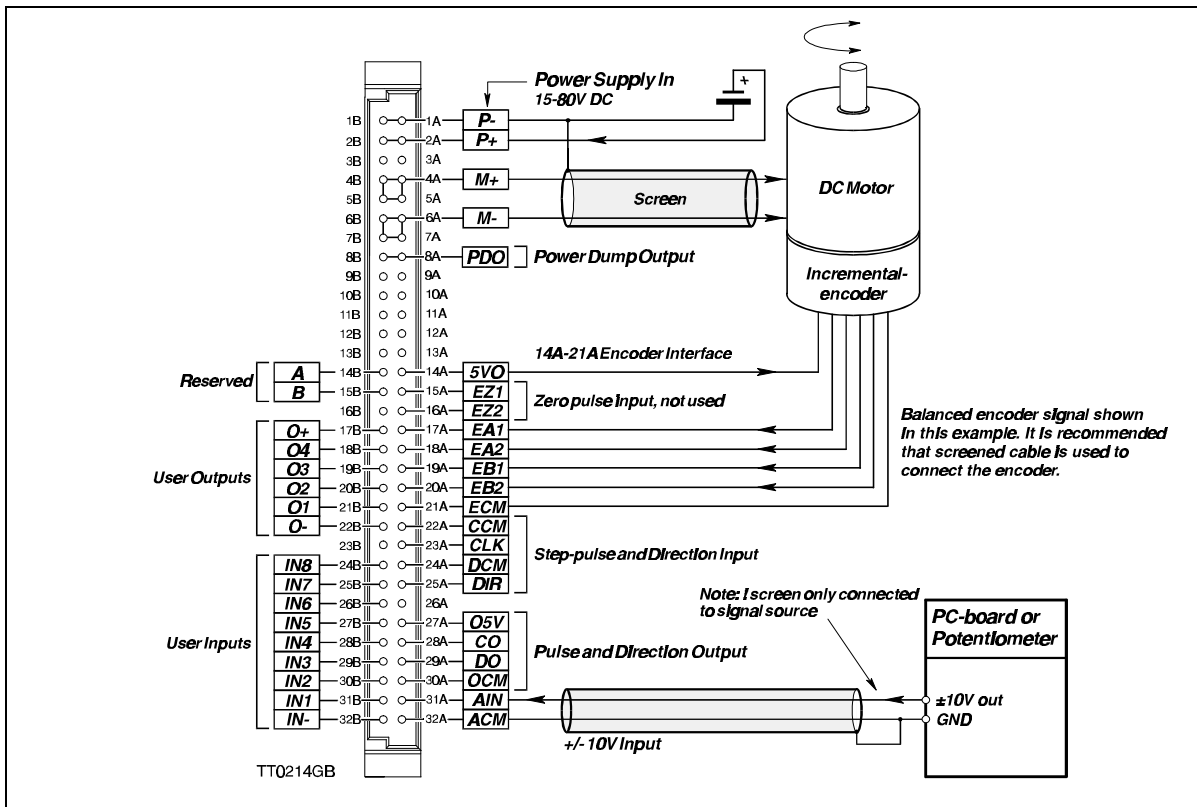
- Follow the procedure below for operation of the DMC10 in Mode 2 (Positioning Mode)
1. Connect the Controller as shown above. For further details, see also the following sections: *Motor Output* page 16 / *Encoder Input* page 19 / *Power Supply* page 21.
 2. Connect the PC via a terminal program (e.g. JVL's *MotoWare* or Windows Hyperterminal), if necessary following the description of the RS232 interface in *RS232 Interface* page 25.
 3. Switch on the Controller. Only the *Power* LED and possibly *Out 1* may be active. If one or more of the red LEDs is active or blinks, either the motor or encoder is probably connected incorrectly.
 4. Send the command `?(enter)` to the Controller and wait until the Controller responds with a status overview.
If the status overview is displayed, the RS232 interface and power supply are correctly connected.
 5. Set the Controller to Positioning Mode by sending the command `MO=2(enter)`.
The Controller should respond `Y`, indicating that Positioning Mode has been selected.
 6. The motor current must be adjusted for the actual motor used. See *Adjustment of Motor Current* page 29.
 7. Servo parameters *KD*, *KP*, and *KI* are set to typical values by default. This means that the motor can be operated without further adjustment. For optimum system operation however, the parameters should be adjusted. See *Adjustment of Servo Regulation* page 30.
 8. The Controller is now set to Positioning Mode. As a test, the motor can be moved to absolute position 1000 by sending the command `SP=1000(enter)`. The motor should move to the specified position. By sending the command `SP=-1000(enter)` the motor will move in the opposite direction to position -1000. If this does not occur, or the motor runs for a very long time, it may be due to the fact that the position counter either was at position 1000, or that the previous position was far from 1000. See the software section for details of other commands.

1.5 Getting Started — Register Mode (Mode 3)



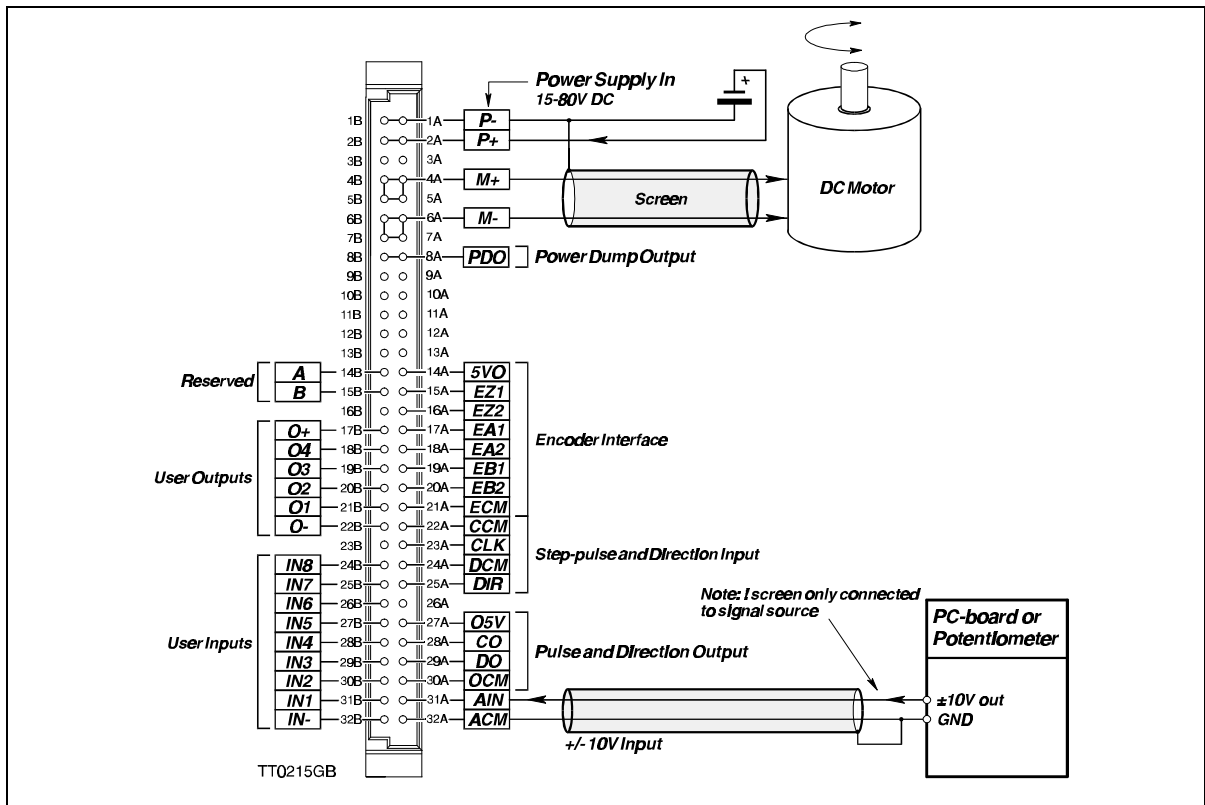
- Follow the procedure below for operation of the DMC10 in Mode 3 (Register Mode)
1. Connect the Controller as shown above. For further details, see also the following sections : *Motor Output* page 16 / *User Inputs* page 17 / *User Outputs* page 18 / *Encoder Input* page 19 / *Power Supply* page 21.
 2. Connect the PC via a terminal program (e.g. JVL's *MotoWare* or Windows *HyperTerminal*), if necessary following the description of the RS232 interface in *RS232 Interface* page 25.
 3. Switch on the Controller, but ensure that all inputs are logic "0". Only the *Power* LED and possibly *Out 1* may be active. If one or more of the red LEDs is active or blinks, either the motor or encoder is probably connected incorrectly.
 4. Send the command ? (enter) to the Controller and wait until the Controller responds with a status overview.
If the status overview is displayed, the RS232 interface and power supply are correctly connected.
 5. Set the Controller to Register Mode by sending the command *MO=3* (enter).
The Controller should respond *Y*, indicating that Register Mode has been selected.
 6. The motor current must be adjusted for the actual motor used. See *Adjustment of Motor Current* page 29.
 7. Servo parameters *KD*, *KP*, and *KI* are set to typical values by default. This means that the motor can be operated without further adjustment. For optimum system operation however, the parameters should be adjusted. See *Adjustment of Servo Regulation* page 30.
 8. The Controller is now set to Register Mode.

1.6 Getting Started — Velocity Mode (Mode 4)



- Follow the procedure below for operation of the DMC10 in Mode 4 (Velocity Mode)
1. Connect the Controller as shown above. For further details, see also the following sections : *Motor Output* page 16 / *Encoder Input* page 19 / *Power Supply* page 21 / *Analog Input* page 23
 2. Connect the PC via a terminal program (e.g. JVL's *MotoWare* or Windows *HyperTerminal*), if necessary following the description of the RS232 interface in *RS232 Interface* page 25.
 3. Switch on the Controller, but ensure that the analog input is 0 V. Only the *Power* LED and possibly *Out 1* may be active. If one or more of the red LEDs is active or blinks, either the motor or encoder is probably connected incorrectly.
 4. Send the command ?(enter) to the Controller and wait until the Controller responds with a status overview. If the status overview is displayed, the RS232 interface and power supply are correctly connected.
 5. Set the Controller to Velocity Mode by sending the command *MO=4* (enter). The Controller should respond *Y*, indicating that Velocity Mode has been selected.
 6. The motor current must be adjusted for the actual motor used. See *Adjustment of Motor Current* page 29.
 7. Servo parameters *KD*, *KP*, and *KI* are set to typical values by default. This means that the motor can be operated without further adjustment. For optimum system operation however, the parameters should be adjusted. See *Adjustment of Servo Regulation* page 30. If the motor is inoperative, first try setting *KI* to a high value (100-1000).
 8. The Controller is now set to Velocity Mode. When the voltage at the input is greater than 0V, the motor will move at a velocity which is proportional to the applied voltage. If the input voltage is less than 0V (negative), the motor will move in the opposite direction.
See section *Adjustment of Analog Input* page 45.

1.7 Getting Started — Torque Mode (Mode 5)

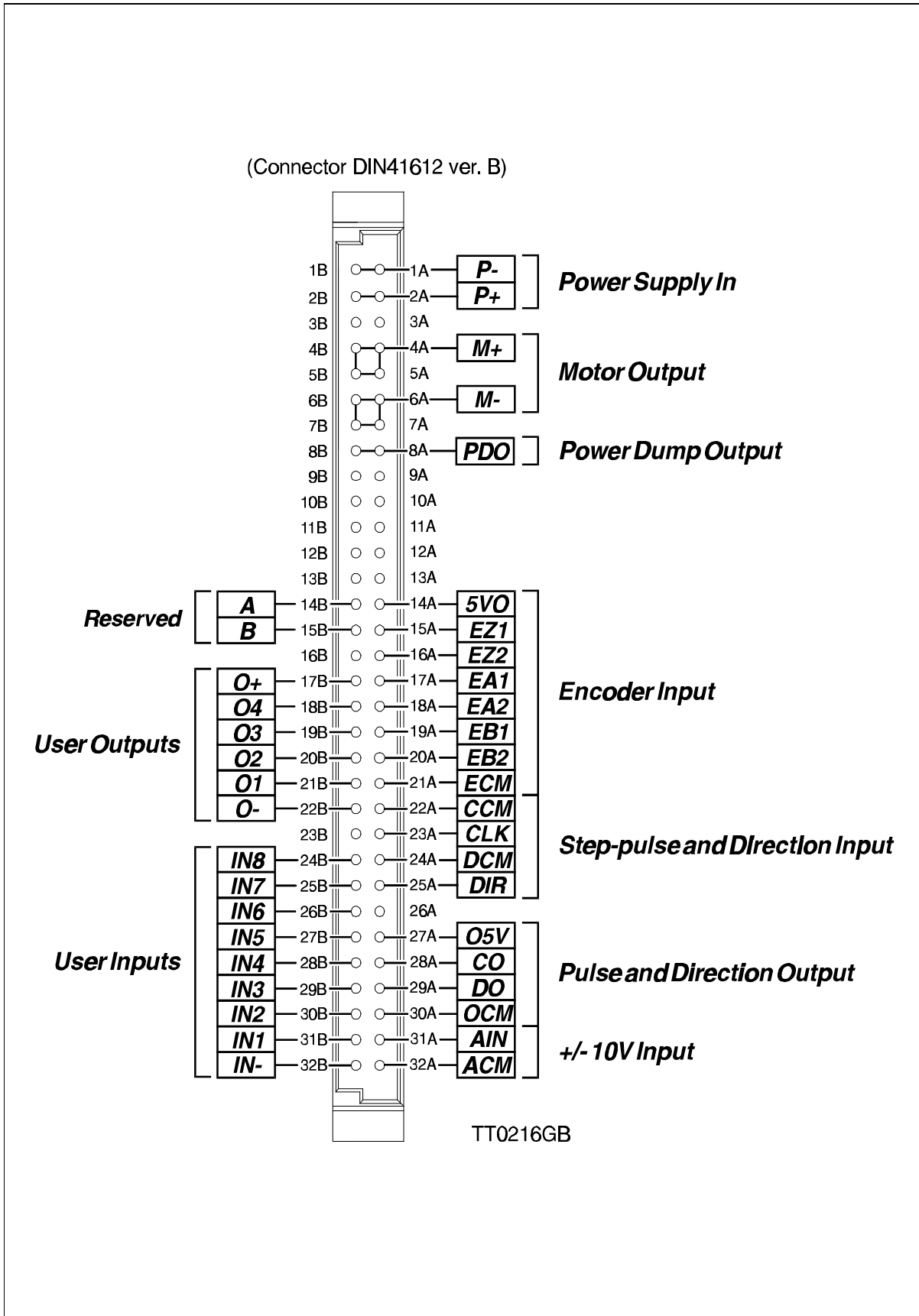


Follow the procedure below for operation of the DMC10 in Mode 5 (Torque Mode)

1. Connect the Controller as shown above. For further details, see also the following sections : *Motor Output* page 16 / *Power Supply* page 21 / *Analog Input* page 23.
2. Connect the PC via a terminal program (e.g. JVL's *MotoWare* or Windows *HyperTerminal*), if necessary following the description of the RS232 interface in *RS232 Interface* page 25.
3. Switch on the Controller, but ensure that the analog input is 0 V. Only the Power LED and possibly *Out 1* may be active. If one or more of the red LEDs is active or blinks, the motor is probably connected incorrectly.
4. Send the command ?(enter) to the Controller and wait until the Controller responds with a status overview.
If the status overview is displayed, the RS232 interface and power supply are correctly connected.
5. Set the Controller to Torque Mode by sending the command *MO=5* (enter).
The Controller should respond *Y*, indicating that Torque Mode has been selected.
6. The motor current must be adjusted for the actual motor used.
See *Adjustment of Motor Current* page 29.
7. Servo parameters *KD*, *KP*, and *KI* are set to typical values by default. This means that the motor can be operated without further adjustment. For optimum system operation however, the parameters should be adjusted.
See *Adjustment of Servo Regulation* page 30. If the motor is inoperative, first try setting *KI* to a high value (100-1000).
8. The Controller is now set to Torque Mode. When the voltage at the input is greater than 0V, the motor will produce a positive torque which is proportional to the applied voltage. When the input voltage is less than 0V (negative), the motor will produce a negative torque proportional to the applied voltage.
See section *Adjustment of Analog Input* page 45.

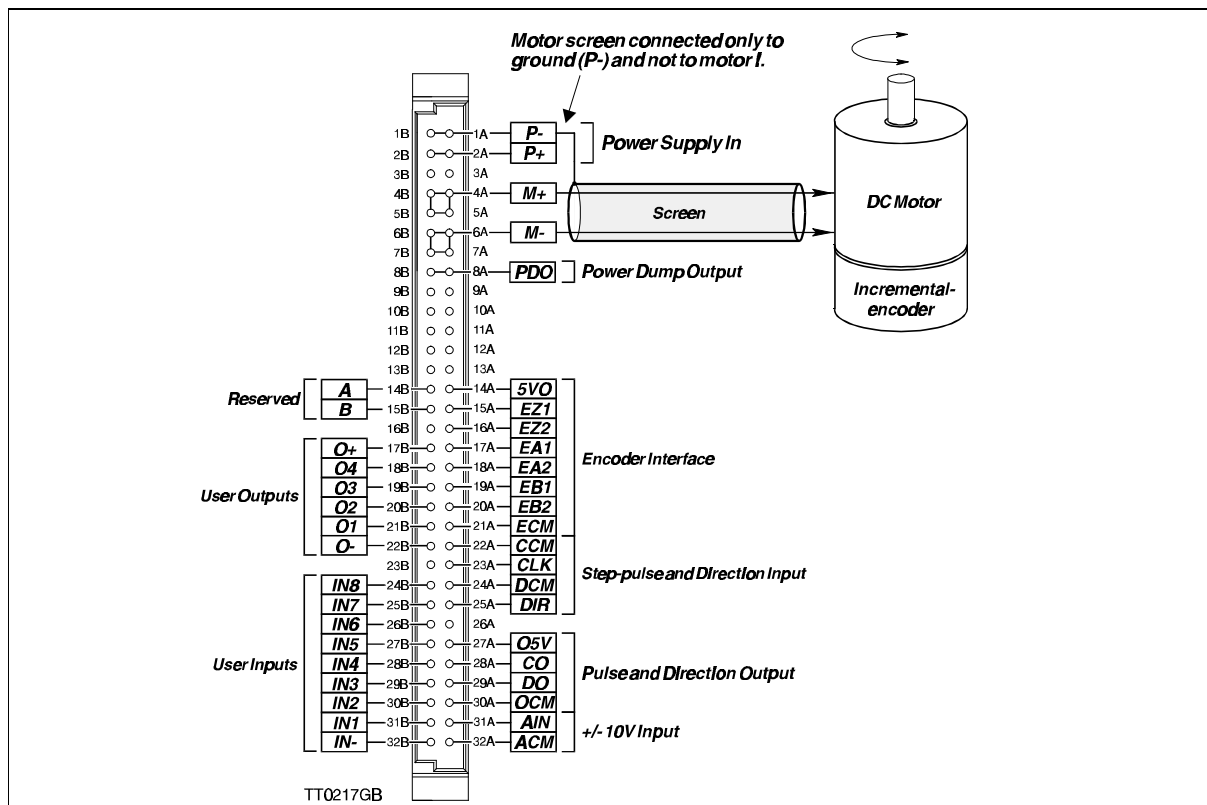
2.1

Connections



2.2

Motor Output



2.2.1 General Aspects of Motor Connection

The Controller is designed for use with common DC servo motors. The Controller can supply currents of up to 12 Amp (continuous) and 25 Amp peak.

These current values must be set using the software commands *CA* and *CP*.

The Controller Driver uses Mos-Fet transistors, which give especially good performance. The motor voltage is regulated at a frequency of 19.75kHz, which ensures that the motor does not produce any audible noise as a result of regulation.

The Driver's switching time is very short (<200nS), which can result in high-frequency noise components in the cables between the driver and the motor.

In certain situations this can result in undesirable influences on other electronic equipment in proximity to the step motor system. To avoid this problem, the connection between the Controller and the motor should be made using screened cable, as illustrated above. Further, it is recommended that screened cable is also used for the encoder cable to avoid influences from the motor cable affecting the encoder signal.

2.2.2 Short-circuiting of the Motor Output

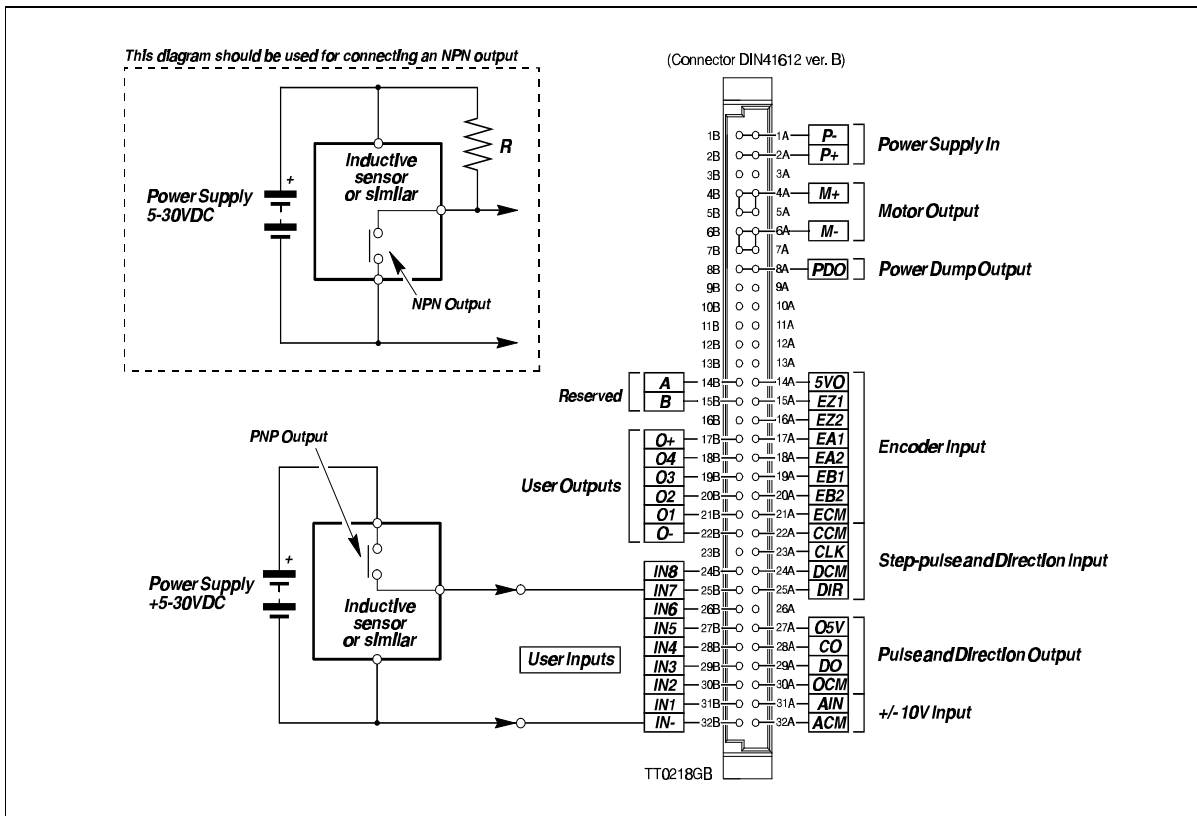
The motor output can withstand short-circuiting between the M+ and M- terminals. In addition, both motor output terminals will withstand short-circuiting to ground or to the positive supply.

If a short circuit occurs, the Controller will stop all activity and report an error condition by activating the *Current* LED. In addition, the Controller's error registers will be activated. See the ES and EST commands.

2.2.3 Protection of the Driver

With certain motors, it may be necessary to connect an inductance coil in series with one of the motor cables. This coil functions as an "elastic" buffer for sudden current transients which would otherwise activate the Controller's short circuit protection.

2.3 User Inputs



2.3.1 General

The Controller is equipped with 8 digital inputs. Each input can be used for various purposes depending on the basic mode of operation selected.

The User Inputs are optically isolated from other circuitry in the DMC10.

All of the Inputs have a common ground terminal denoted *IN-*.

Each of the User Inputs can operate with voltages in the range 5 to 30VDC.

Note that the User Inputs should normally be connected to a PNP output, since a positive current must be applied to the input for it to be activated.

2.3.2 Connection of NPN Output

If an Input is connected to an NPN output, a Pull-Up resistor must be connected between the Input and the + supply. See above illustration.

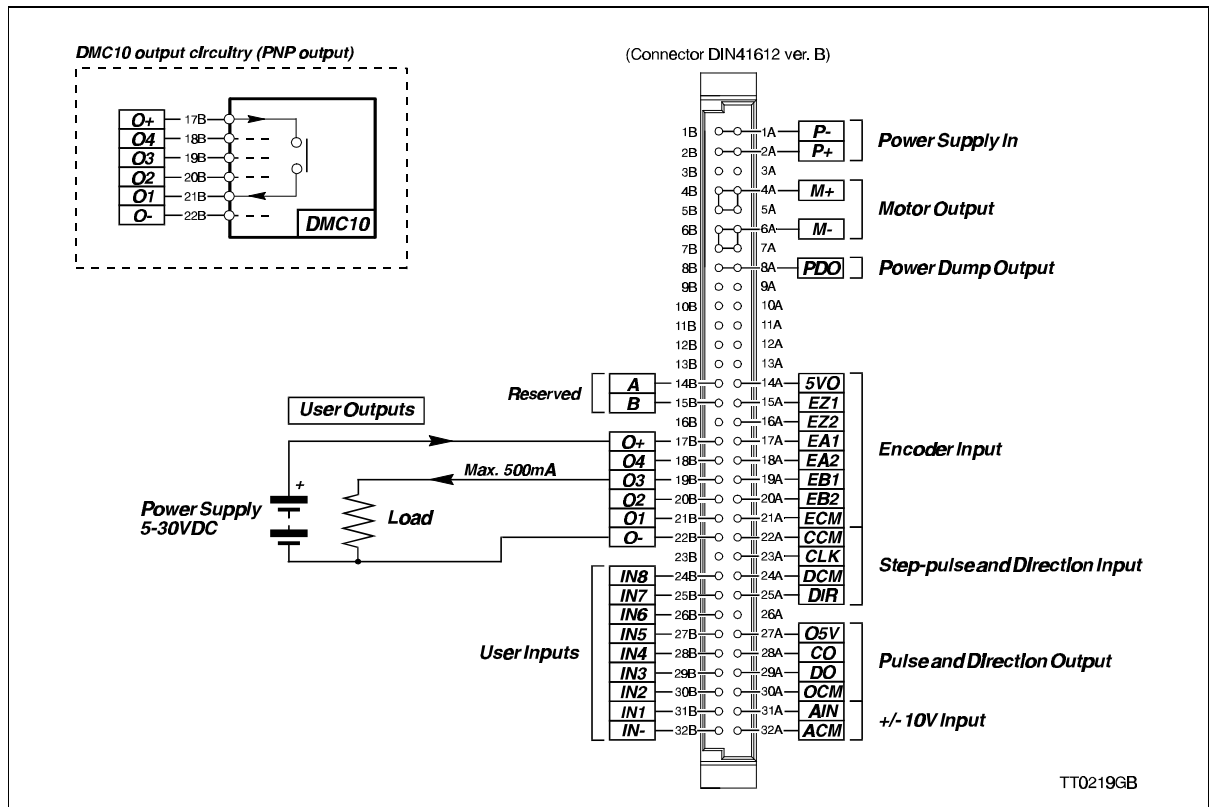
The value of the resistance depends on the supply voltage used.

The following resistances are recommended:

Supply Voltage	Recommended Resistor
5-12VDC	1kOhm / 0.4W
12-18VDC	2.2kOhm / 0.4W
18-24VDC	3.3kOhm / 0.4W
24-30VDC	4.7kOhm / 0.4W

2.4

User Outputs



2.4.1 General

The Controller is equipped with a total of 4 digital outputs. Each of the Outputs can be used for various purposes depending on the basic mode of operation selected.

The Outputs are optically isolated from other Controller circuitry.

The output circuitry must be powered from an external supply. This external power supply is connected to the O+ and O- terminals. The circuitry accepts voltages in the range 5-28VDC.

Each output can supply a continuous current of 500mA.

The Outputs are all source drivers, i.e. if one of the Outputs is activated, contact is made between the + supply (O+) and the respective output terminal. See above illustration.

To indicate the level of each of the User Outputs, the Controller front panel is equipped with 4 LEDs, denoted O1, O2, O3 and O4. These LEDs are activated when the corresponding User Output is active.

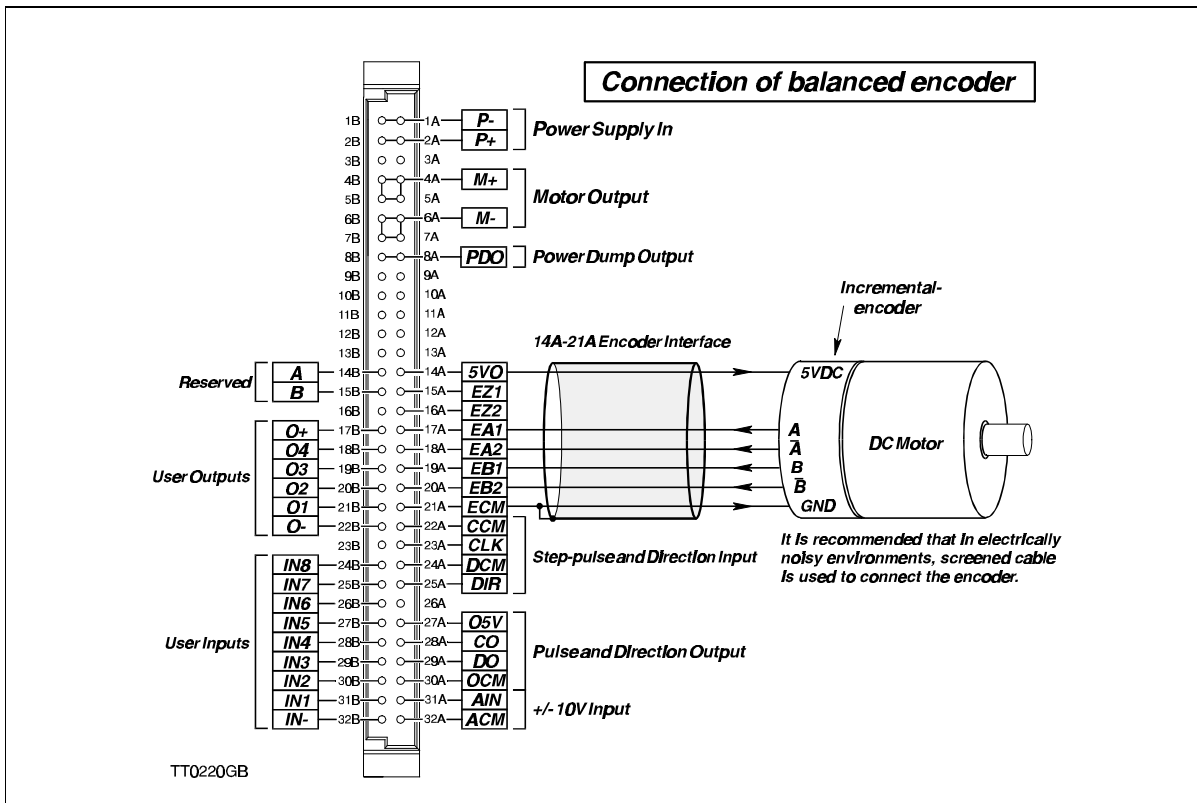
Note ! LEDs O1-O4 do not indicate the actual level at the User Outputs. They are coupled directly to the internal microprocessor and are not connected to the output terminals themselves.

2.4.2 Overload of User Outputs

All of the outputs are short-circuit protected, which means that the output is automatically disconnected in the event of a short-circuit. The output will first function normally again when the short-circuit is removed and the Controller has been switched off for a period of 5 seconds.

The *Out Error* LED on the front panel of the Controller is activated when one or more of the outputs has been short-circuited. This LED is also used to indicate if the output circuitry has overheated due to an overload.

2.5 Encoder Input



2.5.1 General Aspects of the Encoder Input

The Controller must normally be connected to an incremental encoder to detect the velocity and position of the connected motor. Almost all types of encoder can be used, providing they are equipped with one of the following types of output: NPN, PNP, Push-Pull or Balanced.

The Encoder Input can read an encoder signal up to 500kHz. The encoder signal voltage must be in the range 0 to 5V.

Note ! - the cable used between the encoder and the Controller must be screened and the screen must only be connected to the encoder chassis terminal (ECM).

If the Controller is used in Torque Mode (Mode 5), no encoder is used.

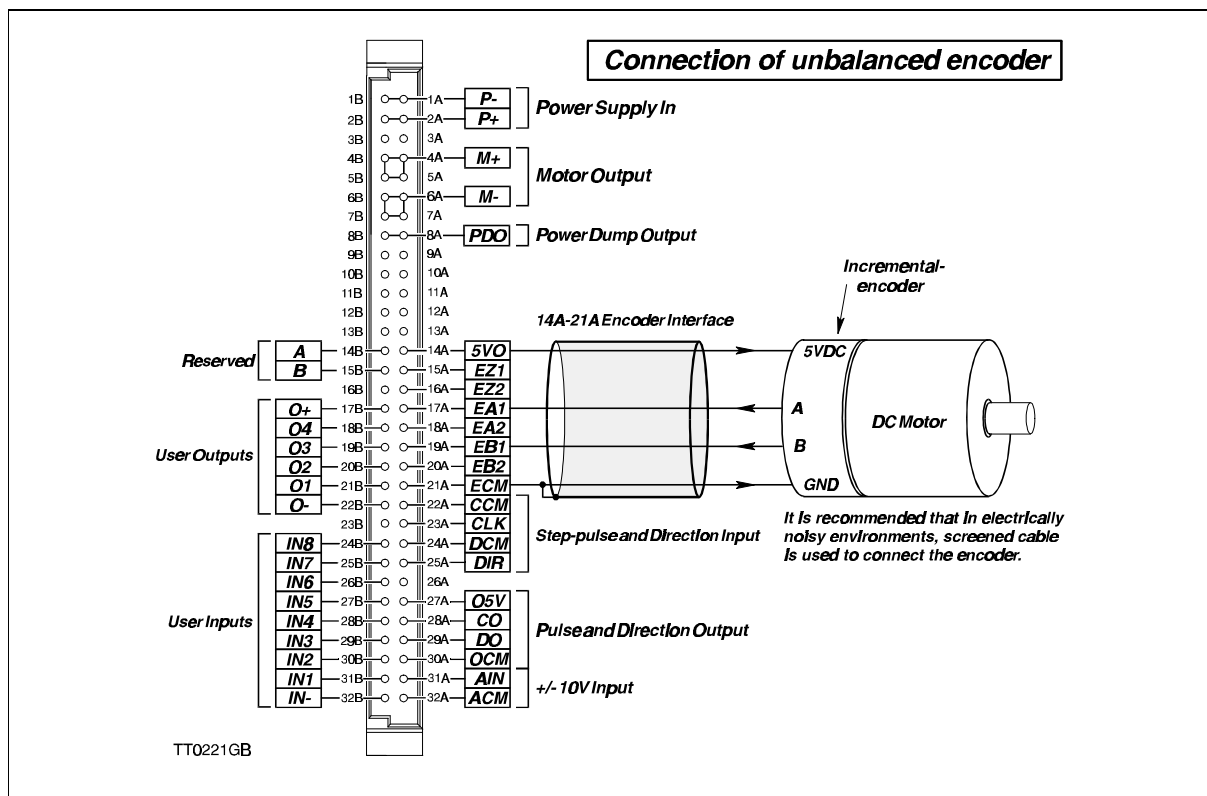
Terminals EZ1 and EZ2 are not used and are reserved for future use.

For details of general encoder set-up, see the ET command.

2.5.2 Encoders with Balanced Output

If an encoder with a balanced output is connected to the Controller, follow the above illustration.

2.5 Encoder Input



2.5.3 Encoders with Unbalanced Output

As mentioned above, the Controller can be used with almost all types of encoder, including encoders with an unbalanced output.

Some types of encoder have an NPN or a PNP output. In such cases the Controller's ET command is used to configure the Encoder Input for the specific type of encoder. See also *Encoder type (ET)* page 59.

If encoders with balanced/unbalanced outputs of the type push-pull or source/sink are used, this configuration can be omitted.

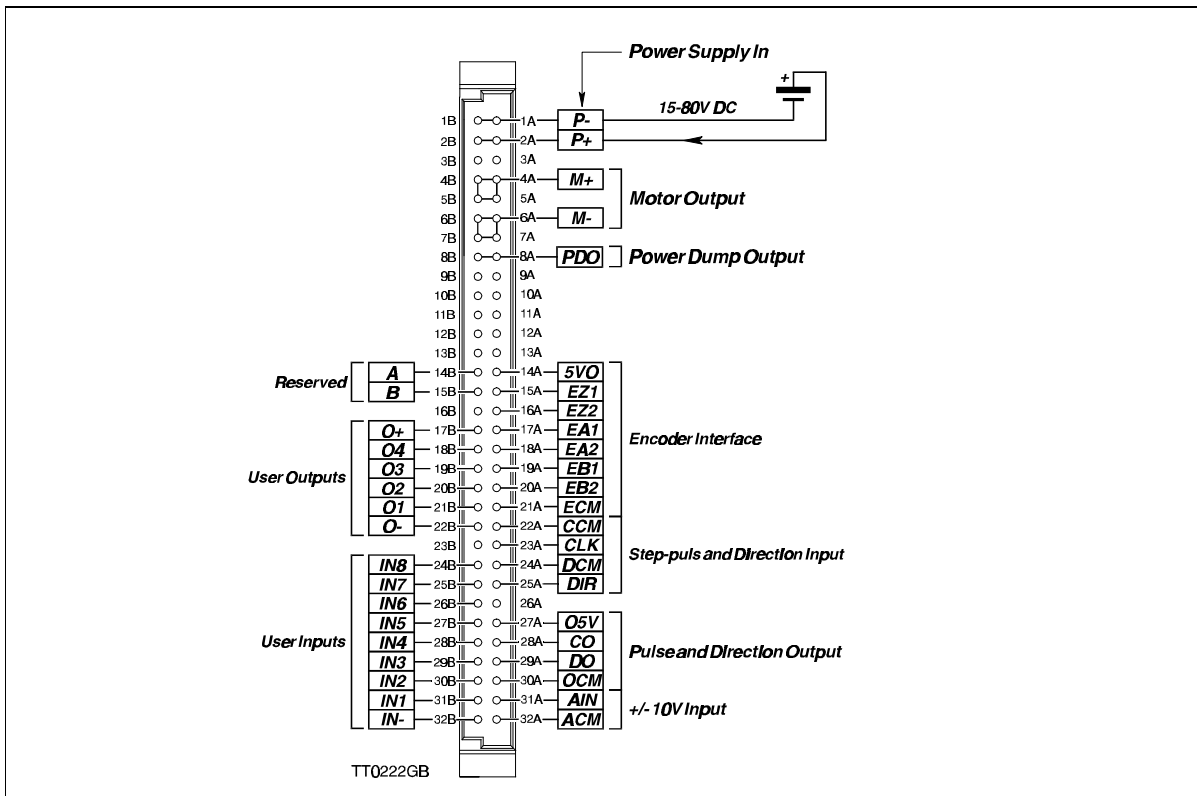
Important : The unconnected terminals EA2, EB2, and EZ2 must be left open. The encoder input will not work properly if they are connected.

2.5.4 Special Encoders/Sensors

JVL currently plans to supply other adaptor modules for other types of encoder and sensor. Contact JVL Industri Elektronik for further details.

2.6

Power Supply



2.6.1 Power Supply

To ensure that powering of the Controller is as simple as possible, only a single supply voltage is connected to the Controller. Internal power supply circuitry ensures correct supplies to the driver and control circuitry, etc.

For optimum driver performance, it is recommended that the external power supply has a capacitance of minimum 2000-5000 μF . Similarly, it is recommended that 1.5mm cables (minimum) are used to connect the power supply to the Controller. If the driver supply voltage falls below 10V, the internal reset-circuitry will re-set the driver. Provision should therefore be made to ensure that the supply voltage is always maintained at a minimum of 12-15V, even in the event of a mains voltage drop.

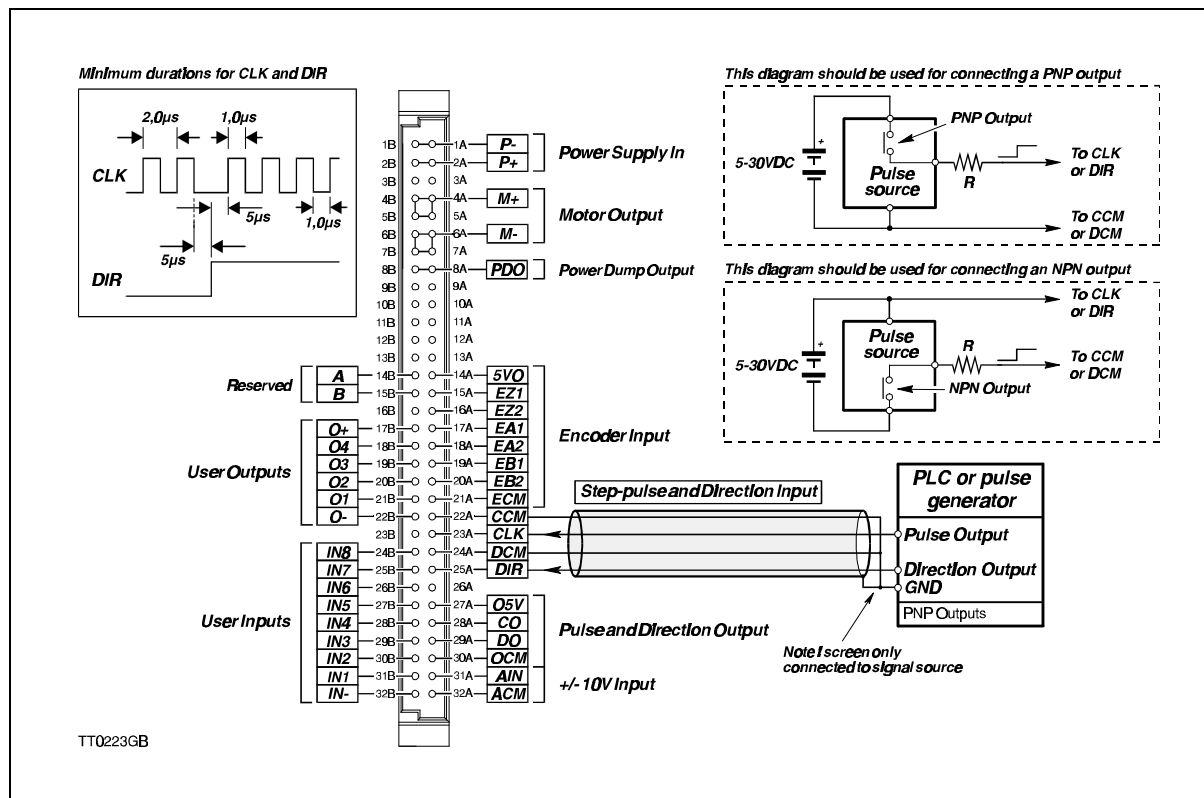
2.6.2 Power Supply Faults

The Controller is protected against incorrect polarity connection and overload.

If a voltage overload occurs, or the supply is connected with incorrect polarity, the Controller's internal fuse will be blown.

The fuse can only be replaced by an authorised service centre.

2.7 Step-pulse and Direction Input



2.7.1 General

The Step-pulse and Direction Inputs are used in conjunction with Mode 1. Each time a pulse is applied to the step-pulse input *CLK*, the motor moves a certain amount. The direction signal *DIR* determines the direction of motor movement. Both signal inputs are equipped with a built-in noise filter which cuts-off frequencies above 625kHz.

The above figure illustrates connections and minimum durations for the signals. Note that if the source used for the step-pulse and direction signals has a PNP output, the input requires special connection, as illustrated above. Similarly if an NPN signal source is used, the output must be connected as shown above. It is recommended that screened cable is used.

2.7.2 Input Voltage

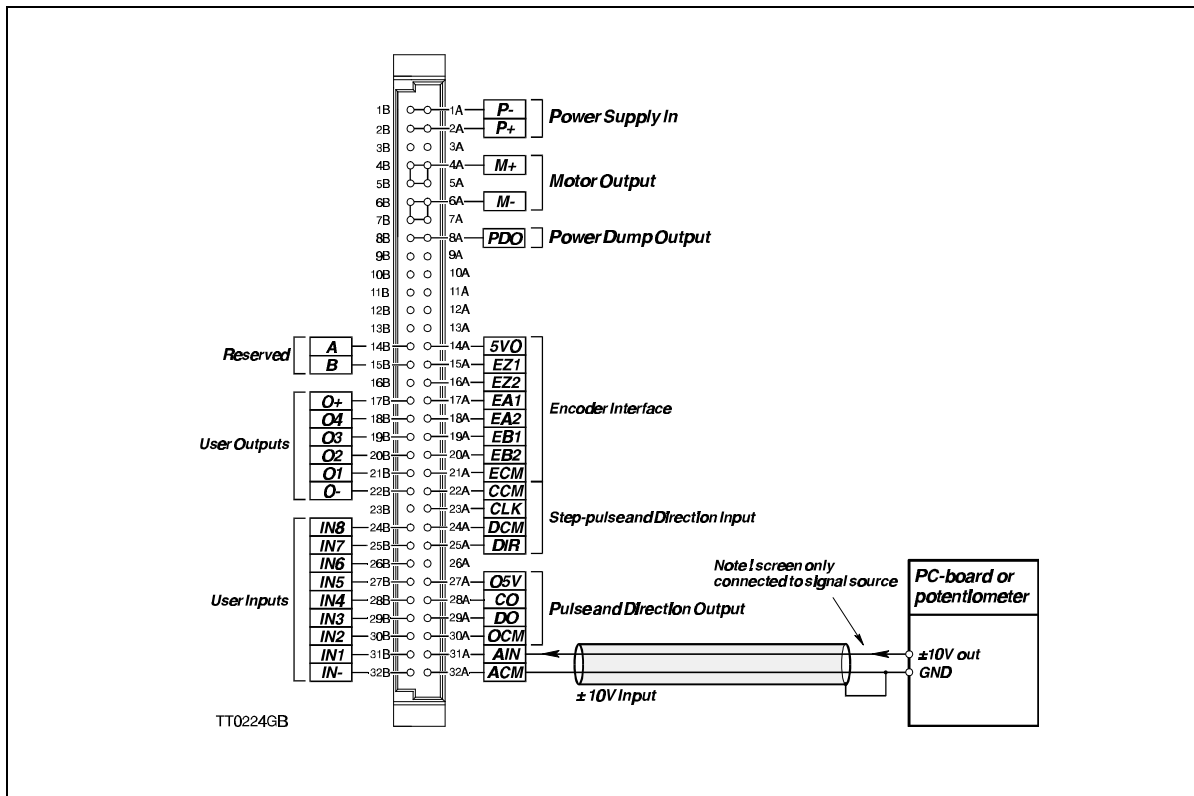
As standard, the inputs are designed to operate with voltages of 5V. If greater input voltages are used, a resistor R must be inserted as illustrated above. The size of the required resistor is given in the following table.

Voltage	Resistance
5-8V	0 Ohm (Short-circuited)
8-12V	470 Ohm / 0,4W
12-18V	1,2 kOhm / 0,4W
18-24V	1,8 kOhm / 0,4W
24-30V	2,2 kOhm / 0,4W

See also the description of Mode 1 : *Step-pulse Mode (MO=1)* page 34.

2.8

Analog Input



2.8.1 General

The Analog Input is used for example when the Controller is operated in Velocity Mode (Mode 4) or Torque Mode (Mode 5).

In this mode, the Controller reads the Analog Input and controls the motor velocity in accordance with the voltage applied at the Input.

The Analog Input accepts input voltages in the range -10V to +10V and is galvanically isolated from all other inputs and outputs, including power supply terminals. Note however that the Input shares a common internal supply with the RS232 interface and is therefore not galvanically isolated from the interface.

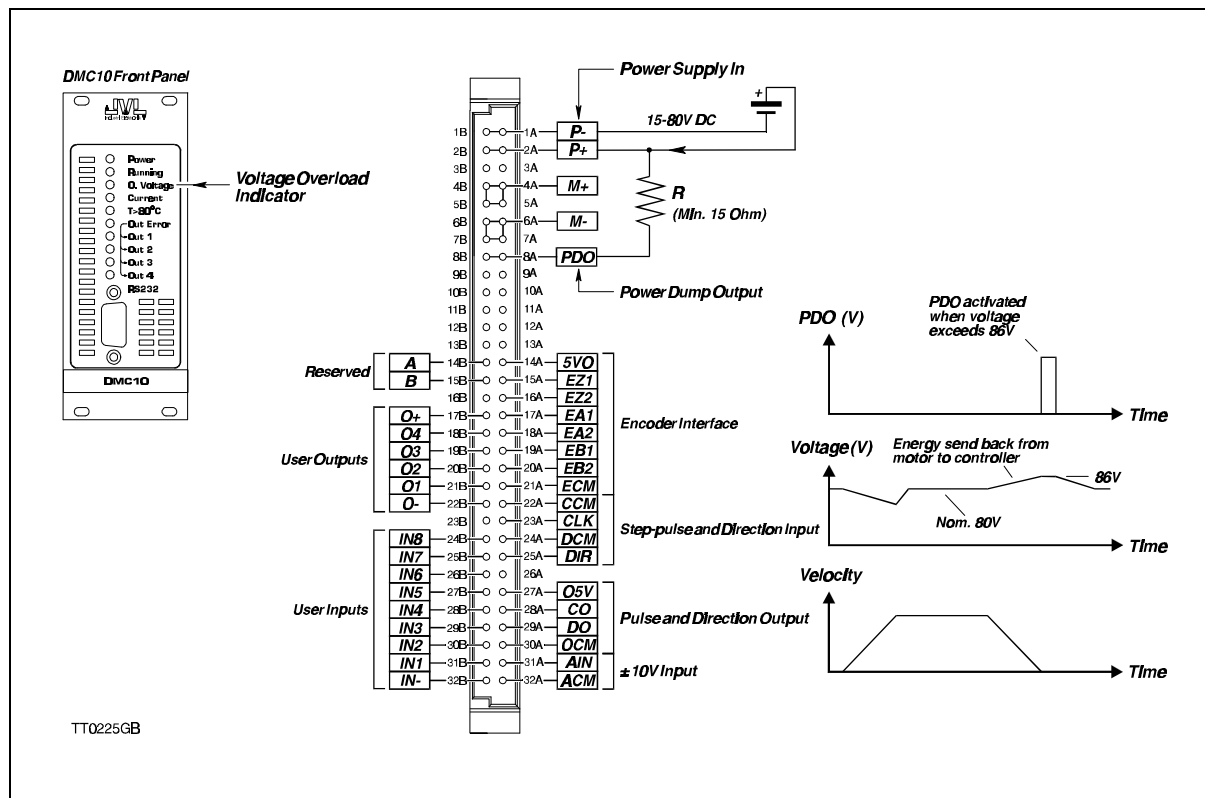
The Analog Input is protected against overloads up to 100V peak and has a built-in filter which removes input signal noise.

Always use screened cable for connecting the source to control the Analog Input since the motor, etc. can easily interfere with the analog signal and result in instability.

The Controller is equipped with an analog to digital converter (ADC) which converts the measured analog signal level. The ADC has a resolution of 11 bits, which gives a total working range of 2048 steps.

2.9

Power Dump Output



2.9.1 General Aspects of the “Power Dump” Output

If the Controller is used in systems in which there are very large inertial loads (flywheels, etc.), a problem can arise during deceleration in that energy is sent back from the motor to the Controller supply.

This can result in increases in the supply voltage to a critically high level, above the Controller’s maximum working range. To alleviate this problem, the Controller’s “Power Dump” Output (PDO) can be used. This Output can sink the energy to an external resistance and thus avoid the Controller shutting down and reporting an error.

2.9.2 Detailed Description of “Power Dump”

The value of the PDO resistance will depend on many parameters, including the max. rpm of the motor, the supply voltage, how rapidly the motor decelerates, etc.

It is recommended however that the shunt resistor has a minimum value of 15 Ohm / 50W. The resistor’s rated power can be greater or less, depending on the actual load.

1. When the Controller registers that the supply voltage exceeds 89V, the PDO-Output is activated and the “O. Voltage” LED is activated.

2. If activation of the PDO output and thus the PD shunt resistor does not stop the increase in supply voltage, the following occurs:

When the supply voltage exceeds 95V, the Controller automatically shuts down and the motor is short-circuited (and thus stops instantaneously).

The Controller transmits error message “E29”. The PDO remains active until the voltage falls below 89V, and the Controller will remain in this error state until it receives the RE-SET command. See section *Reset Controller (RESET)* page 69.

If the supply voltage continues to increase due to other circumstances, the Controller’s internal voltage overload protection will be activated and short-circuit the supply so that the internal fuse is blown. See section *Power Supply Faults* page 21.

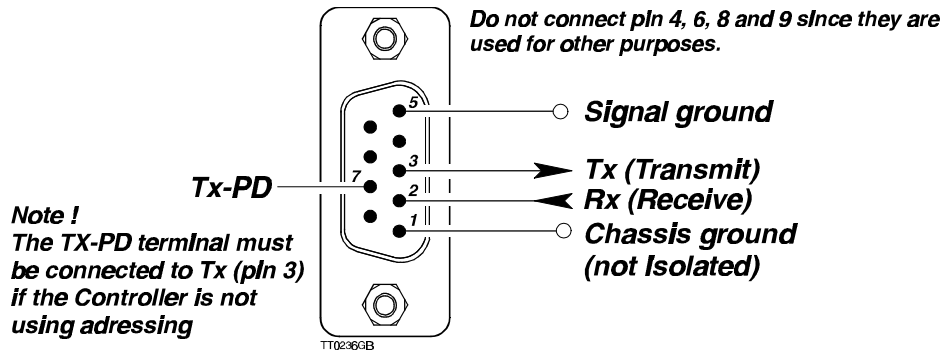
2.10

RS232 Interface

2.10.1 Interface Connection

The Controller Interface uses the widespread RS232C standard, offering the advantage that all Personal Computers and standard terminals can be connected via the interface. The 3 interface signals Rx, Tx and ground are used. The interface cable length should not exceed 10 meters.

Controller interface:



2.10.2 Communication Protocol

The Controller uses the following format: (1 startbit), 7 databits, Odd parity, 1 Stop bit. Note that a startbit is always used in the RS232C/V24 protocol.

2.10.3 Communication Rate

The Controller operates at a fixed communication rate (Baud rate) of 9600 Baud. The Baud Rate must be set accordingly on the terminal or PC used to communicate with the Controller.

2.10.4 Command Syntax

Communication with the Controller must follow a specific command syntax:

[Address] Command [=Argument] [;Command [=Argument] [Checksum] <CR>

Text in square brackets [] may be included or omitted depending on the set-up.

Address: This address must be used when more than one Controller is used on the interface. See also the ADDR command - *Address (ADDR)* page 47.

Command: The command itself.

Argument: The numeric argument for the command. An argument always begins with the equal-to sign "=". Certain commands do not use arguments (e.g. commands that display set-ups).

; More than 1 command can be included in a command line. A semi-colon ";" is used to delimit multiple commands.

Checksum: In situations where long communication lines are used, a checksum can be used to ensure that commands are received correctly. If an error occurs, the error message E9 is received and the command must be re-transmitted. See also the CHS command.

<CR>: ASCII value 13. This character terminates the command line.

2.10

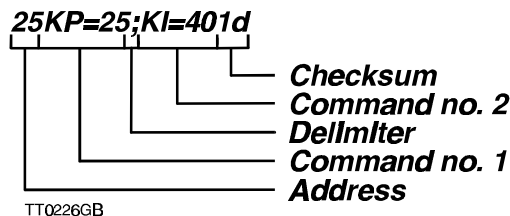
RS232 Interface

2.10.5 Synchronisation

When communicating with the Controller, each command must be terminated either by a <CR>, (ASCII 13) or a semi-colon ";". This tells the Controller that the command string is complete and interpretation can begin. When a checksum is used, command interpretation will not begin until the entire command line has been received, i.e. is terminated by a <CR>. A maximum of 80 characters may be sent in a single command line.

2.10.6 Checksum

In industrial applications, electrical noise from motors, etc. often occurs. This noise is quite arbitrary and cannot be eliminated 100% even by effective electrical filtration. To ensure correct transmission of Controller commands, a facility is included for using checksums in the command string. A typical command line may be as follows:



In this example, addressing is used (address 25). Two commands are transmitted and are delimited by the semi-colon ";". Finally a checksum is transmitted. The checksum consists of two characters. The checksum is a 'simple' checksum and is calculated in the following manner. First the ASCII value of each of the characters in the command line is determined. These values are summed and the two least significant characters (the least significant byte) of the result's hexadecimal value are used. The two least significant digits are converted to ASCII values and transmitted together with the command string. The actual calculation is as follows:

$$50 + 53 + 75 + 80 + 61 + 50 + 53 + 59 + 75 + 73 + 61 + 52 + 55 = 797 \text{ (decimal)} = 31d \text{ (hexadecimal)}$$

The checksum is thus 1d, which is sent as ASCII 49 (decimal) and 100 (decimal). Hex. characters a-f can also be sent as capitals, i.e. d can also be sent as ASCII 68 (decimal).

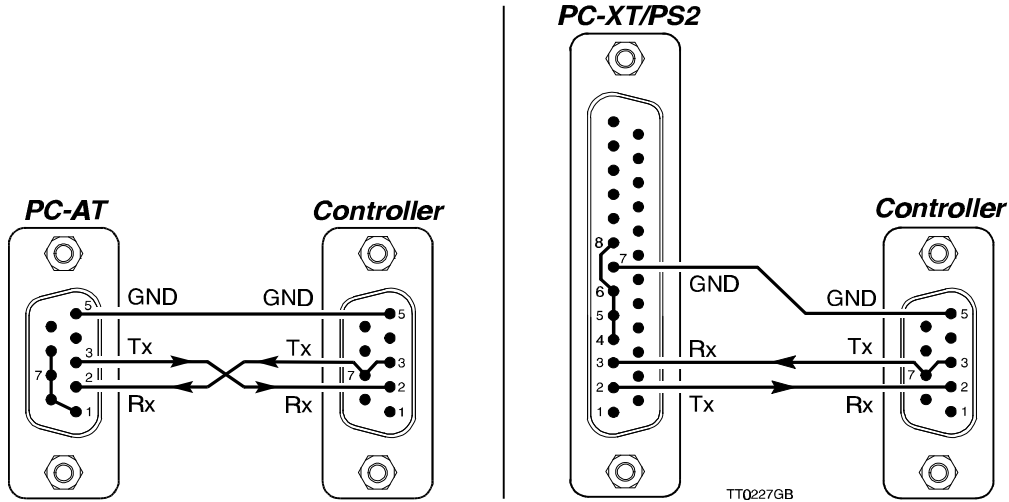
In the event that the command string is affected during transmission, the checksum will not correspond and the Controller will send an error message "E9" indicating that a checksum error has occurred. The command string must then be re-transmitted. The checksum function is activated using the CHS command.

2.10

RS232 Interface

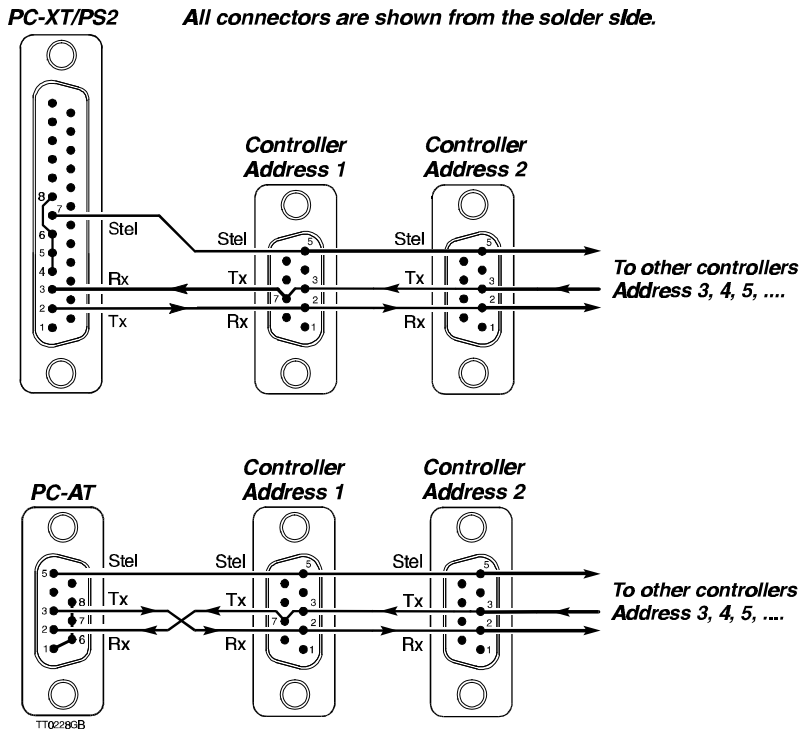
2.10.7 Connection to PC

For communication from a PC, the following connection diagrams can be used. These show the connections between the Controller and an IBM AT or IBM-XT/PS2:



2.10.8 Connection of Several Controllers to a PC

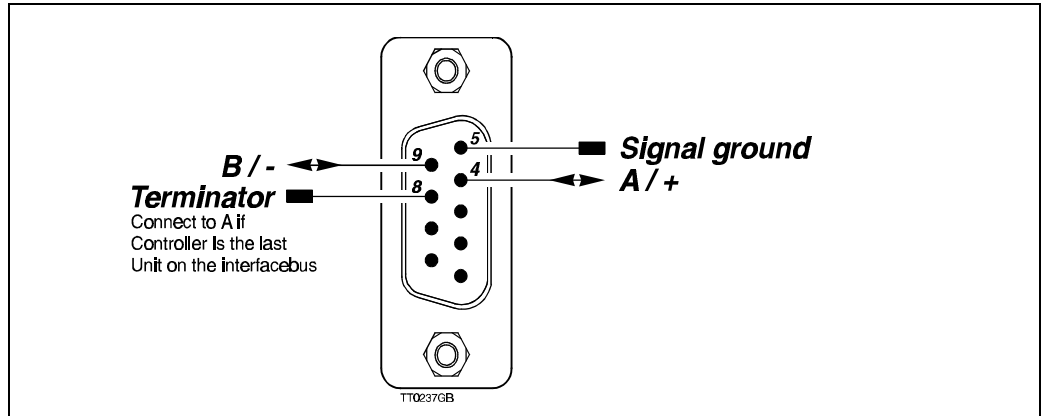
For connection of more than 1 Controller to a PC (i.e. using addressing), the connection diagrams given below can be used. Note that Tx (pin 3) must be connected to TX-PD (pin 7) on one of the Controllers included in the system. The address for each controller must be set-up by using the ADDR command - see also *Address (ADDR)* page 47. The diagrams show the connections between Controllers and an PC with 25 or 9 pole RS232 connector:



2.11

RS485 Interface

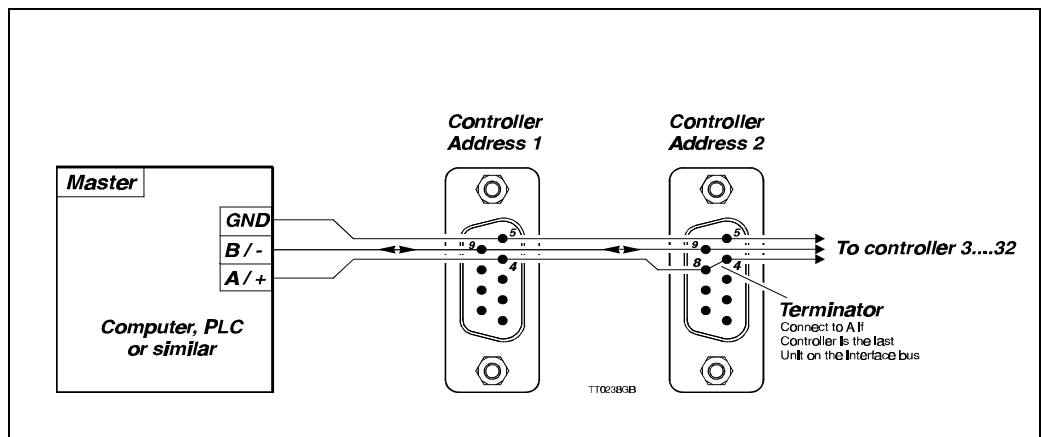
The Controller also includes an RS485 interface, in addition to the normal RS232 interface. The RS485 interface is intended for purposes where 1 to 32 controllers are connected on the same interface in a noisy environment.



The communication protocol is exactly the same as that for RS232 communication. The only difference is the balanced signal lines, and the fact that all communication is half-duplex, which means that the Controller cannot send and receive at the same time, unlike RS232 communication.

The RS485 interface makes it possible for up to 32 units to be connected to the same interface bus. On the last Controller on the interface, the terminal marked Terminator (pin 8) must be shorted to the A terminal (pin 4). If more than 1 Controller is used on the same RS485 line, the ADDR command must be used to set-up each of the Controllers for an individual address - see also *Address (ADDR)* page 47.

The following illustration shows a typical system with 2 or more units connected to a computer or similar system.



2.12 Adjustment of Motor Current

2.12.1 General

Certain basic control parameters must always be adjusted to suit the actual system. The most significant of these are allowable motor current and the servo regulation parameters.

2.12.2 Adjustment of Motor Current

A DC motor has 2 current limits which must not be exceeded in order not to overheat the motor or reduce its operating lifetime. These current limits are the motor's allowable continuous rated current and peak current, and are adjusted in the following manner.

Adjustment of Rated Current.

Consult the actual motor's data sheet to determine the specification of the motor's rated current. This parameter can be specified as "Continuous Current", "Rated Current", or "Nominal Current".

The Rated Current is set using the Controller's *CA* parameter.

Example :

To adjust the Controller's setting for a rated current of 1Amp.: *CA=1* (enter).

The Controller will under no circumstances allow the motor to draw a continuous current greater than 1.0 Amp.

Note that the rated current value can only be adjusted to an integer value (1, 2, 3....).

Adjustment of Peak Current.

The Controller is adjusted for the motor's specified peak current rating as follows:

Consult the actual motor's data sheet to determine the specification for the motor's allowable peak current. This may be specified as "Peak Current", "Instantaneous max. Current", or "Stall Current". Most types of motor can withstand a peak current that is 3-4 times greater than the continuous current.

The peak current is adjusted on the Controller using the *CP* parameter.

Example :

If the peak current is to be adjusted to 4 Amp.: *CP=4* (enter).

The Controller will then under no circumstances allow the motor to draw a peak current greater than 4.0 Amp.

Note that the peak current can only be adjusted to an integer value (1, 2, 3....).

2.13 Adjustment of Servo Regulation

2.13.1 Adjustment of Servo Parameters

The DMC10 servo regulator is of the PID-type and has therefore 3 parameters that must be adjusted.

The function of the servo loop is to ensure that the motor operates smoothly with stable movements and stops at its intended location. The 3 servo parameters must be adjusted according to the actual conditions of the specific system, since the motor type, load, supply voltage and other factors all have a decisive influence on the value of the parameters required.

The 3 regulator parameters are denoted as follows and have the following function:

- KP Determines the system's proportional amplification. This parameter is the most important of the 3 since the system will function using this parameter alone.
- KD Determines the system's differential amplification. This parameter determines how aggressively the system will react to sudden changes in load or a sudden change in velocity.
- KI Determines the system's integral amplification. This parameter determines the extent to which a persistent positioning error will influence the system.

The 3 parameters can be quickly adjusted in the following manner:

1. All 3 parameter are set to 0 by keying $KP=0$ (enter) followed by $KI=0$ (enter) followed by $KD=0$ (enter).
2. The value of KP is then slowly increased until the system begins to become unstable. KP is then adjusted to half this value.
3. To make the system response quicker, KD can be adjusted. The value of KD is increased until the system becomes unstable. KD is then adjusted to approximately 0.5 to 0.7 of this value.
4. The value of KI is then increased until the system is unstable. KI is then set to approximately 0.5 to 0.75 of this value.
5. If a system is required which reacts quickly when a positioning error occurs but in which the summed error does not increase, the integral summation limit IL should be used.
6. Adjustment of the servo parameters can be completed by fine-tuning the individual parameters.
7. Remember to store the parameter values in permanent memory by keying MS (enter)

3.1

Use of RS232 Commands

The DMC10 DC Motor Controller can be controlled via the Controller's RS232 interface. Controller commands are sent as ASCII characters terminated by <CR> ASCII 13 (decimal) or ";". See also section RS232 Interface page 25.

Some of the Controller commands have associated command parameters, others do not. For those commands with associated parameters, transmitting the command alone will provoke the Controller to respond with the command and the currently associated value. When no addressing is used, the Controller always responds when a command has been received. If the purpose of the command is to display a value or set-up, the required information will be sent as a reply or a 'Y' will be transmitted to indicate that the command has been received. In the event that incorrect information has been sent to the Controller, for example a command that does not exist or a value that is out of range, etc., the Controller will respond with an error message. The error message consists of an 'E' followed by a number, followed by an explanatory text. See also *Error Messages* page 78.

Example:	Sent to Controller	VM<CR>
	Received from Controller	VM=500<CR>
	Sent to Controller	VM=600<CR>
	Received from Controller	Y<CR>
	Sent to Controller	VM=-5<CR>
	Received from Controller	E2: Out of range<CR>

When addressing is used, the Controller will not acknowledge the receipt of a command. Any errors in communication will be registered by the error status register 0.

Commands may be sent both as upper-case and lower-case characters. With the exception of error messages, replies from the Controller are always upper-case.

The following sections describe all of the RS232 commands. As mentioned above, all commands must be terminated by a carriage return character <CR> or a semi-colon ";" before they will be interpreted. This is not included in the description of each individual command.

3.2 Step-pulse Mode (MO=1)

The system can be controlled as a step motor system via the step-pulse and direction input. The motor moves a single step each time a voltage pulse is applied to the step-pulse input. This feature enables the DC-servo to replace a step motor in many applications. The velocity and acceleration /deceleration will be determined by the external voltage pulses. MO must be set to 1 for operation in this mode. See section Getting Started — Step-pulse (Mode 1) page 8.

In this mode, the IN1 Input is used to select whether the incoming step-pulses will be multiplied by a factor specified by the MP command. If the Input is high (1) the pulses are multiplied by the specified factor. If the Input is low (0) the pulses are used directly.

Example of the use of step pulse mode:

Select step pulse mode using MO=1

Adjust the servo-loop. See section *Adjustment of Servo Regulation* page 30.

Select whether the step pulses will be multiplied by a given factor. See the MP command.

The motor can now be controlled via the step-pulse (CLK) and direction (DIR) input terminals.

Commands of particular interest for operation in this mode are:
MU, MP, ET, PR, PE

3.3 Positioning Mode (MO=2)

In this mode the Controller will position the motor via commands transmitted over the RS232 interface. Various operating parameters can be continuously adjusted via the interface while the motor is running. This mode is primarily used in systems in which the Controller is permanently connected to a PC via the RS232 interface. MO must be set to 2 for operation in this mode.

See also section Getting Started — Positioning (Mode 2) page 9.

The position is specified in encoder pulses. If for example the specific encoder type multiplied by the encoder multiplication factor (MU) gives 4000 pulses per revolution, a specified change in position of 8000 pulses means that the motor will rotate 2 revolutions. The motor's instantaneous position can be read regardless of whether the motor is running or stationary. When a new position is set up, the motor moves to the new position using the programmed velocity profile. See AC and VM.

Motor operation can use a programmed velocity profile by programming a maximum velocity and acceleration. In this mode, when the motor is operated to move to a new position, it will operate using the programmed velocity profile and the profile will always follow the acceleration/deceleration values. This means that the motor will not always attain maximum velocity if the distance is short. Motor status can be read using the RS command.

At any time the motor can be stopped using either the H or SH commands.

Note that in order to achieve the correct velocity and acceleration, the number of encoder pulses per revolution must be set up using the PR command.

Example of the use of Positioning Mode:

Select Positioning Mode using MO=2

Set a maximum velocity using VM

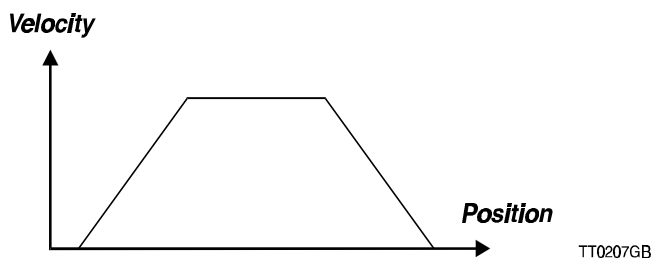
Set an acceleration using AC

Adjust the servo-loop. See section *Adjustment of Servo Regulation* page 30.

The motor can now be set to move using either the SP or SR commands.

Commands of particular interest for operation in this mode are:

MU, ET, PR, SP, SR, VM, AC, PE



3.4 Register Mode (MO=3)

The Controller can also be configured for absolute or relative positioning via 4 digital inputs. See also section Getting Started — Register Mode (Mode 3) page 10.

The Controller has 16 programmable parameter sets. Each parameter set can be used to store information about acceleration, position (relative or absolute) and velocity. Selection of a parameter set is made using inputs IN1-IN4.

Register set	Binary inputs				Function			
	IN4	IN3	IN2	IN1	Acceleration	Velocity	Position	Relative
0	0	0	0	0	XA0	XV0	XP0*	XR0**
1	0	0	0	1	XA1	XV1	XP1	XR1
2	0	0	1	0	XA2	XV2	XP2	XR2
3	0	0	1	1	XA3	XV3	XP3	XR3
4	0	1	0	0	XA4	XV4	XP4	XR4
5	0	1	0	1	XA5	XV5	XP5	XR5
6	0	1	1	0	XA6	XV6	XP6	XR6
7	0	1	1	1	XA7	XV7	XP7	XR7
8	1	0	0	0	XA8	XV8	XP8	XR8
9	1	0	0	1	XA9	XV9	XP9	XR9
10	1	0	1	0	XA10	XV10	XP10	XR10
11	1	0	1	1	XA11	XV11	XP11	XR11
12	1	1	0	0	XA12	XV12	XP12	XR12
13	1	1	0	1	XA13	XV13	XP13	XR13
14	1	1	1	0	XA14	XV14	XP14	XR14
15	1	1	1	1	XA15	XV15	XP15	XR15

0 = Low (Inactive) 1 = High (Active)

* XP0 indicates the direction for the zero-point seek function -1 = negative, 1 = positive

** XR0 indicates whether automatic zero-point seek will occur.

The motor is started by user input 5 (IN5).

If User input IN5 is activated, a register set is selected and the motor moves to the position specified in the register set. The velocity and acceleration are also determined by the actual register set.

To stop the motor, 2 different possibilities exist. The DIF (digital input format) register decides which method is used, as described in the following sections.

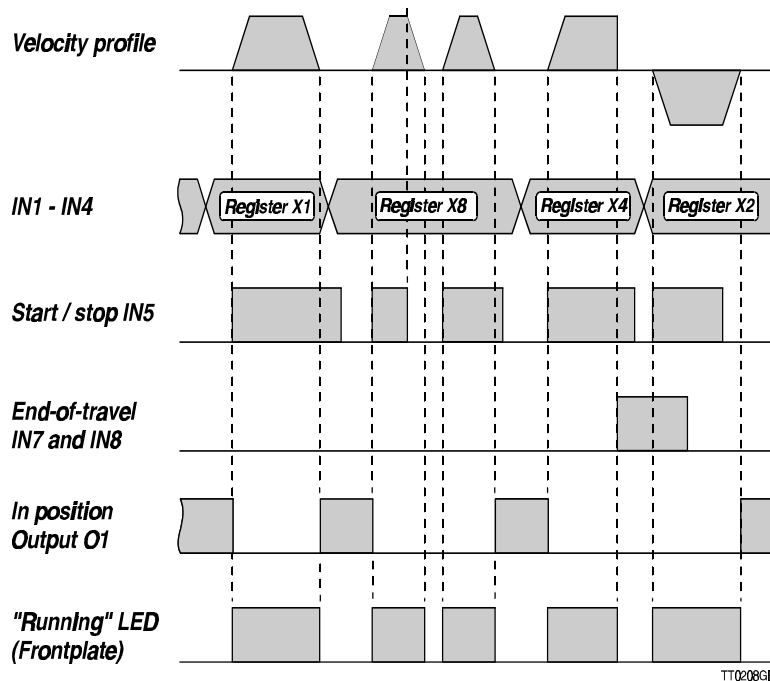
3.4 Register Mode (MO=3)

3.4.1 Digital Input format 1 (default)

If the DIF register is set to 1, start and stop of the motor will take place as follows. Start of the motor occurs by activating User Input IN5. If IN5 is high, a parameter set is selected and the motor moves to the new position in accordance with the selected velocity profile.

If IN5 is set low before the desired position is reached, the motor will stop according to the programmed deceleration (acceleration). When IN5 is again set high, the motor continues to the required position. When the required position is reached, O1 is set high to indicate that the motor has reached its destination. In this format, IN6 is only used for the zero search sensor.

Timing diagram for selection of register set etc. when DIF = 1



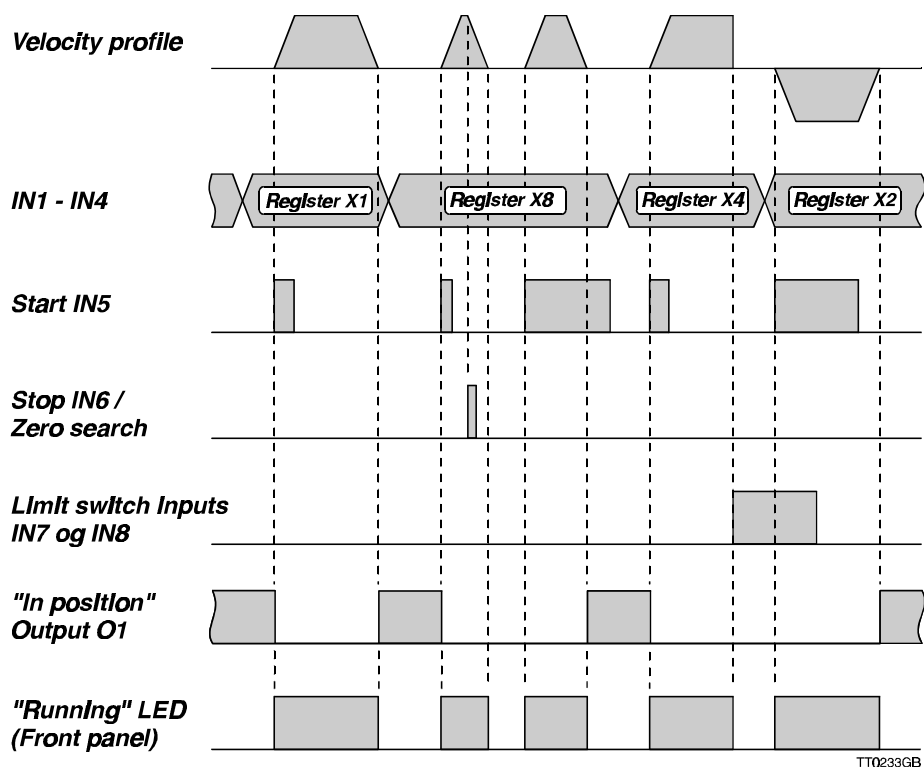
Please note that the motor is stopped if one of the End-of travel inputs is activated.

3.4 Register Mode (MO=3)

3.4.2 Digital Input format 2.

If the DIF register is set to 2, start and stop of the motor will take place as follows. Starting the motor occurs in the same way as when DIF=1, i.e. by activating User Input IN5. However, when Digital Input Format 2 is used, the motor can only be stopped if the final position is reached or if IN6 is activated. Every time the motor is started, the position counter will be set to zero (0). This function makes sure that the position counter never reaches its maximum. This input format typically is used for feeding mechanisms where it is typical to make moves in one direction only. If the final position (specified by the XP register) is reached, the user output O1 is activated to indicate the position/distance is reached.

In this input format, IN6 is not only used as a stop input but can also be used as a zero sensor input.



3.4.3 General about Register Mode setup

Commands of particular interest for operation in this mode are:
MU, ET, PR, XR, XA, XP, XV, PE

The setup of these registers can take place in 2 different ways.

- Via direct communication using a terminal program like Windows HyperTerminal.
- Via JVL's windows software *MotoWare*.

Both methods are described in the following sections.

3.4 Register Mode (MO=3)

3.4.4 Setup using direct communication

Use a standard terminal program (like Windows HyperTerminal) or JVL's MotoWare (On line editor) for the DMC10 setup.

Example 1: Send to the Controller XV1=1000 Set the velocity in register set 1 to 1000 RPM.

Received from Controller √

Example 2: Send to the Controller XV1 Show register set 1

Received from Controller XV1=1000

Example 3: Send to the Controller XV Show all the velocity register sets

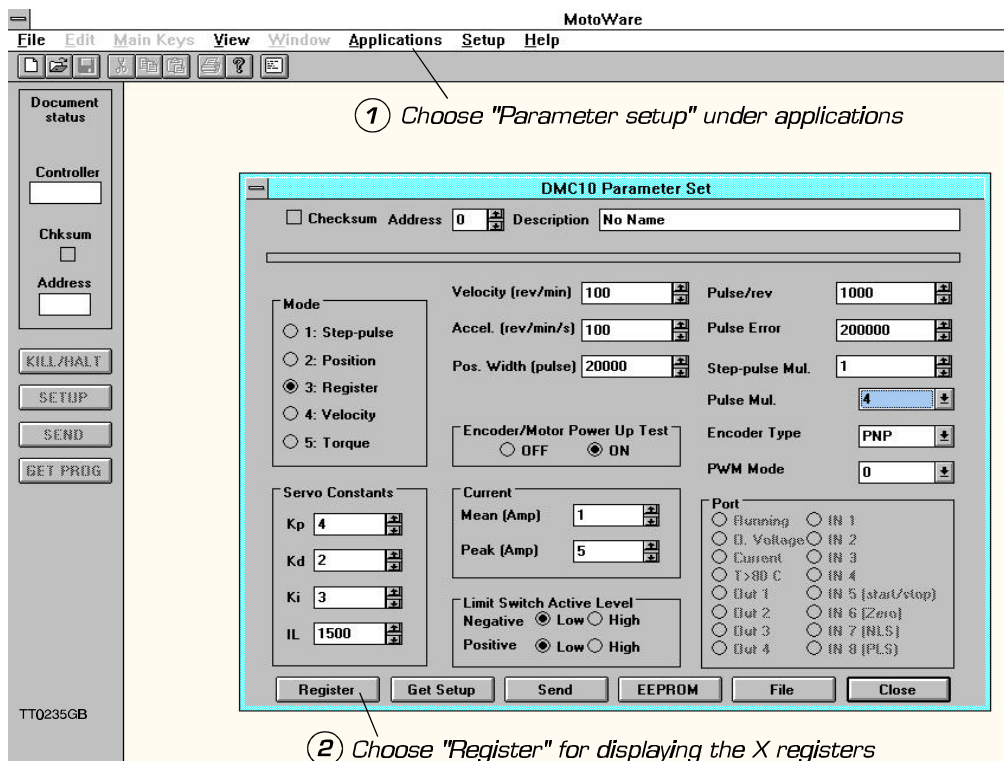
Received from Controller XV0=0
 XV1=1000
 XV2=200

 XV15=0

Once the X-registers have been set up, the contents can be saved permanently in memory using the MS command. The Controller will thus store all the new values, even if the Controller is powered down.

3.4.5 Setup using MotoWare

All X-registers are available in MotoWare under the Applications menu. Select Register to view all of the X-registers.



3.4 Register Mode (MO=3)

The Register Setup window shows all the X-registers. All of the values can be edited. If the *Acceleration* and *Velocity* fields are set at 0, the default setup from the main setup window will be used.

Press OK when the X-registers have been edited as required.

If the *Relative* field is activated, the positioning sequence for that specific register will be made relative to the position before the start signal, and thus the movement, was started. Register 0 (XA0, XV0, XP0, and XR0) is especially dedicated to zero searching.

If *Relative* under register 0 is activated, the DMC10 will automatically start a zero search after each power up. The direction is determined by XP0.

See also *Zero-point Seek Function* page 43.

Register	Acceleration	Velocity	Position	Relative	Zero Search Reg.
0	0	0	-1	- Dir	<input type="checkbox"/>
1	0	3000	1230		<input type="checkbox"/>
2	0	0	6540		<input checked="" type="checkbox"/>
3	0	0	10000		<input checked="" type="checkbox"/>
4	0	1000	1300		<input type="checkbox"/>
5	0	0	0		<input type="checkbox"/>
6	0	0	50000		<input type="checkbox"/>
7	0	0	2500		<input type="checkbox"/>

! If Acceleration or Velocity is equal to zero, Default values are used!

Cancel OK

Press OK here when all values has been setup

3.5

Velocity Mode (MO=4)

Analogue control of the motor velocity can be achieved using the analog input (AIN). The input voltage must be in the range -10V to +10V, with negative voltages producing movement in a negative direction and positive voltages producing movement in a positive direction. The VM command specifies the maximum velocity, i.e. the velocity at which the motor will rotate for maximum voltage applied to the input.

The numeric value of the full-scale voltage does not have to be the same in both positive and negative directions. Use the VVx commands for adjustment of the Analog Input.

Once the servo-loop has been adjusted, the Controller will ensure that the required velocity is maintained, regardless of whether the motor is loaded or not. The load however must not be so great that the current limits are exceeded. If the rated current or peak current limits start to regulate, motor operation will be very unsmooth and in extreme circumstances the motor will resonate.

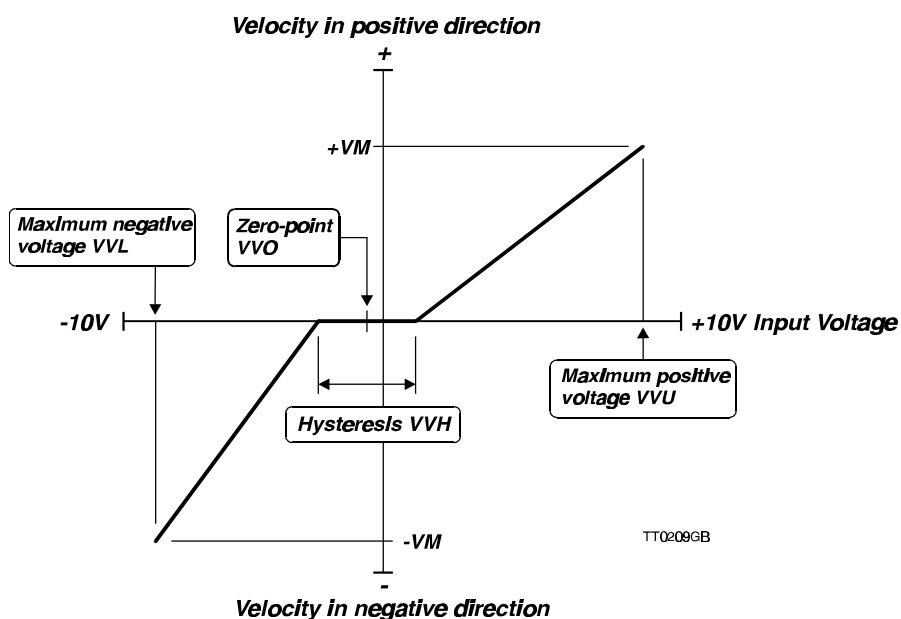
If for example VM=500 rpm and the analog input voltage is set to 5V, the motor will rotate at 250 rpm in the positive direction. See also section *Maximum Velocity (VM)* page 73

Use of Velocity Mode:

1. Select Velocity Mode (MO=4)
2. Adjust the servo loop, see *Adjustment of Servo Regulation* page 30
3. If necessary, adjust the analog input. See *Adjustment of Analog Input* page 45
4. Set the maximum velocity using VM

The motor can now be controlled via the analog input (AIN).

Commands of particular interest for operation in this mode are:
MU, ET, PR, VM, VVH, VVL, VVO, VVU



3.6

Torque Mode (MO=5)

The motor torque can be controlled by an analogue signal using the Analog Input (AIN). The input voltage must be in the range -10V to +10V, with negative voltages producing a negative torque and positive voltages producing a positive torque. The value of the torque is specified in Amps. CA is used to specify the maximum torque, i.e. the torque provided by the motor when a maximum input voltage is applied.

The numeric value of the full-scale voltage does not need to be the same in both the positive and negative directions. Use the VVx commands to adjust the Analog Input.

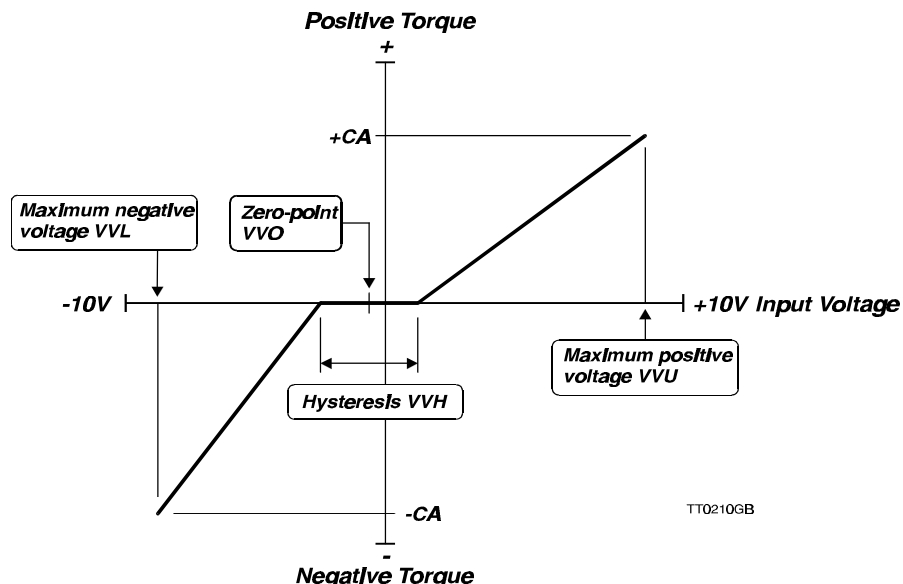
If for example CA is set to 6 Amp and the analog input voltage applied is set to 5V, a torque corresponding to 3 Amp will be produced. The torque is directly proportional to the motor current. The relationship is specified by a torque constant that is normally denoted by K_T or K_A .

Use of Torque Mode:

1. Select Torque Mode (MO=5).
2. Adjust the servo-loop. See *Adjustment of Servo Regulation* page 30.
3. If necessary, adjust the Analog Input. See *Adjustment of Analog Input* page 45.
4. If required, set maximum velocity using VM.
5. Set maximum torque using CA.

The motor can now be controlled via the Analog Input (AIN). In this mode, VM is used to ensure that the motor does not exceed some critical velocity. The velocity limit in this mode is a precautionary measure and not a precise control. The velocity can only be measured if an encoder is used.

Commands of particular interest for operation in this mode are:
CA, VM, VVH, VVL, VVO, VVU



3.7

Zero-point Seek Function

3.7.1 Zero-point Seek Function

The motor can be brought to a known mechanical reference position. This is done using a sensor connected to the HM input. Parameter set 0 (IN1 IN2 IN3 IN4 = 0000) differs from the other parameter sets in that it stores information about how this zero-point seek function will be carried out. The parameter set *XA0*, *XP0*, *XV0* and *XR0* determines how the zero-point seek is done. Parameter *ZL* determines the HM input's active level. These parameters have the following function:

Parameter	Function	
<i>XA0</i>	Specifies acceleration/deceleration during zero-point seek. The specified value is expressed in rpm/second. If <i>XA0</i> is set to 0, the Controller will use the AC parameter during zero-point seek.	
<i>XP0</i>	<i>XP0</i> =-1 specifies that the zero-point seek will occur in a negative direction.	<i>XP0</i> =1 specifies that the zero-point seek will occur in a positive direction.
<i>XV0</i>	Specifies the nominal velocity during zero-point seek. If <i>XV0</i> is set to 0, the Controller will use the VM parameter during zero-point seek.	
<i>XR0</i>	<i>XR0</i> =0 specifies that the Controller does not automatically perform a zero-point seek after start-up.	<i>XR0</i> =1 specifies that the Controller does automatically perform a zero-point seek after start-up.
<i>ZL</i>	<i>ZL</i> =0 specifies that HM is activated when low	<i>ZL</i> =1 specifies that HM is activated when high.
<i>SZT</i>	<i>SZT</i> =0 specifies normal reset	<i>SZT</i> =1 specifies reset with index channel.
<i>CB11</i>	<i>CB11</i> =0 Reset AP after zero-search	<i>CB11</i> =1 Do not reset AP after zero-search

A reset will occur when one of the following conditions is met:

After start-up or after the Controller has received the *RESET* command.

This will only occur if *XR0*=1 (see above table).

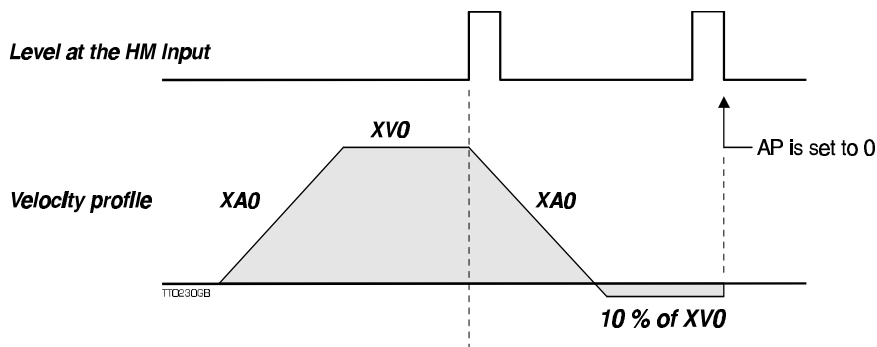
If the Controller receives the zero-point seek command *SZ*.

If the Controller is set to Mode 3 (Register Mode) and register set *X0* is selected.

3.7.2 Reset Sequence

When the zero-point seek function is activated, the motor will move in the specified direction at the specified velocity until the *HM* input becomes active. The motor then decelerates and stops, after which it moves back until the *HM* sensor becomes passive.

The result of the seek process is that the motor is positioned precisely at the zero-point contact. The zero-point is thus located and the motor's AP (Actual Position) will be set to 0. If *SZT*=1, the zero-point seek sequence will be extended as described in section *Extended Zero-point Seek Function* page 44.

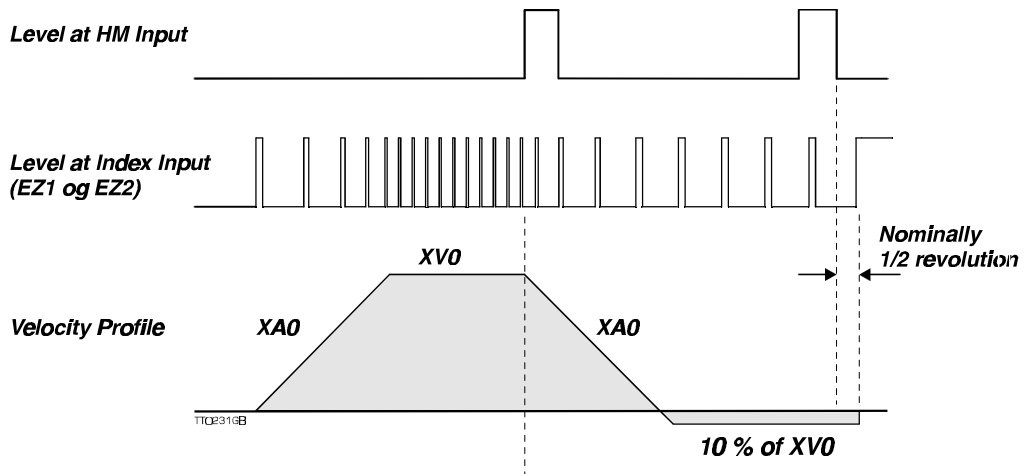


3.7 Zero-point Seek Function

3.7.3 Extended Zero-point Seek Function

If the SZT parameter is activated ($SZT=1$), the reset sequence is carried out normally as described above, but after the HM input is deactivated again, the motor will continue to move in the same direction until a signal is received at the index channel from the motor's encoder.

This type of reset is particularly accurate. Note that the zero-point sensor must be located approximately 1/2 motor revolution before activation of the index channel. Otherwise inaccuracies of ± 1 motor revolution may occur.



3.8 Adjustment of Analog Input

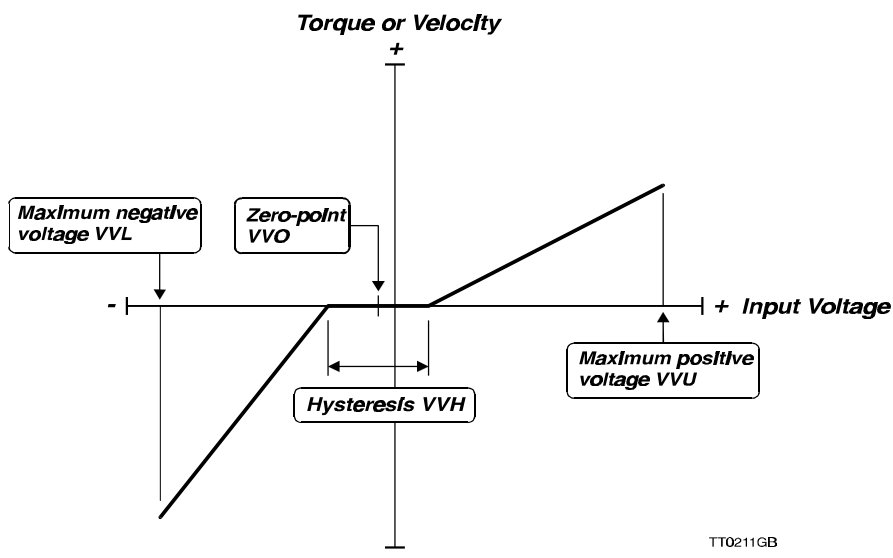
The motor can be controlled directly using the Controller's Analog Input. Voltages applied to the Input must be in the range $\pm 10V$. The Analog Input is used in Velocity Mode (MO=4) and in Torque Mode (MO=5). See section *Analog Input* page 23 for further information about the Input.

Before the Analog Input is used, it must be adjusted for the actual application. This adjustment is necessary because the signal source supplying the control signal to the Controller may have an offset error or may only be able to output $\pm 9.5V$ or less.

1. Select Velocity Mode (MO=4) or Torque Mode (MO=5).
2. Remove the voltage to the motor using the command PO=1, so that the motor does not move during the adjustment procedure.
3. Adjust the zero point by setting the input to 0V, and send the command VVO.
4. Set the input voltage to the maximum negative value (max. -10V) and send the command VVL.
5. Set the input voltage to the maximum positive value (max. +10V) and send the command VVU.
6. Set a hysteresis value using VVH. VVH is set to the number of ADC steps around the 0V zero point where the motor must not move.
7. Reset the input voltage (apply 0V).
8. Set the voltage to the motor using the command PO=0.

The motor can now be controlled within the limits set by VVL and VVU, with a range around the zero point given by VVO and VVH in which the motor remains stationary. The motor is controlled linearly in the range from the maximum negative voltage to the hysteresis level below the zero point, and in the range from the zero point plus hysteresis to the maximum positive voltage. Note that if the zero-point is not 0V, and the negative voltage is not numerically the same as the positive voltage, the control profile will be asymmetric.

See also *CB6 - Stop motor (set velocity to 0)* page 50.
CB7 - Invert the analog input page 51.
CB8 - Use O1 to indicate hysteresis window page 51.



TT0211GB

3.9 Commands

3.9.1 Show set-up (?)

Command ?

Modes 1, 2, 3, 4, 5

Description The most important details of status and set-up can be displayed using this single command.

Usage ? Displays values.

Example Sent to Controller ?
Received from Controller

```
Max. Velocity (RPM):      VM=3000
Acceleration (RPM/S):    AC=30000
Average Current (AMP):    CA=2
Peak Current (AMP):       CP=8
Constant Kd:              KD=100
Constant Ki:              KI=50
Constant Kp:              KP=40
Constant IL:              IL=1500
Pulses/Revolution:       PR=500
Multiply Control Pulses:  MP=1
Multiply Encoder Pulses:  MU=4
Position Width (PULSES): PW=30
Mode:                     MO=2
Encoder Type:             ET=1
Input (IN8-IN1):         IN=00011110
Output (4 LEDs - O4-O1): OUT=00000001
Actual Position (PULSES): AP=2
Max. Position Error:      PE=1073741823
Power Disabled:           PO=0
Control Flag CB (1..n)   CB=0,0,0,0,0,0,0,0,0
Motor startup test:      EMT=1
Error status:
E0: No errors
```

3.9.2 Controller type (!)

Command !

Modes 1, 2, 3, 4, 5

Description This command (exclamation mark) can be used to obtain information about the Controller's type and address. The Controller will reply to this command regardless of whether addressing or checksum are used. Thus there must only be 1 Controller connected to the interface if this command is used without an address. The command can be used alone, i.e. ! or together with an address.

Usage ! Show Controller type and address.

Example Sent to Controller !
Received from Controller DMC10 : ADDR=24

3.9

Commands

3.9.3 Acceleration (AC)

<u>Command</u>	AC
<u>Modes</u>	2, 3
<u>Range</u>	100 - 100000 rpm/sec.
<u>Description</u>	This command is used to specify the acceleration/deceleration profile. If the motor is running when the acceleration is changed, the acceleration will first be changed when the motor has stopped.
<u>Usage</u>	AC = x Set acceleration in rpm/sec. AC Show acceleration.

3.9.4 Address (ADDR)

<u>Command</u>	ADDR
<u>Modes</u>	1, 2, 3, 4, 5
<u>Range</u>	0 - 255
<u>Description</u>	The Controller is as default configured to react to all commands on the interface bus (Point to Point communication). In this case the address is set to 0 (ADDR=0). When the address is 0, the address must not be sent when communicating with the Controller.

It is also possible to connect several Controllers to the same interface bus. In this case each Controller must be assigned its own unique address. Addresses in the range 1-255 can be used. The number of Controllers that can be simultaneously controlled is however dependent on the system hardware.

Changing the ADDR register will have immediate effect, which means that no answer will be given after sending the ADDR command. No answer is given since using addressing will disable all answers permanently.

If responses are needed, the CB5 control bit can be set. See also *CB5 - Send "Y" when using addressing* page 50.

Note: If a Controller's address has been forgotten, the ! command can be used.

<u>Usage</u>	ADDR=x Set address to x. ADDR Show address (if ADDR=0). If ADDR>0 use the ! command.
--------------	---

<u>Example</u>	The address for a specific Controller must be set to 5. This is done by sending the following command Sent to the Controller ADDR=5 The Controller will now use the new address and no answers will be given if commands are sent to the Controller.
	Answer from the controller (none)

(Continued)

3.9

Commands

(Continued)

Now a positioning command must be sent to the Controller. This is done as follows:

Sent to the Controller	5SP=1000	Go to position 1000. The address (5) must be inserted before the command to make sure that the Controller accepts the command.
Received from Controller	(none)	No response from the Controller but the command will be executed.

If the ADDRESS needs to be changed back, this can be done in the following manner.

Sent to the Controller	5ADDR=0	Set address=0. The current address must be inserted before the address command to make sure that the Controller accepts the command.
Received from Controller	Y	Yes - The Controller has received the command and is now setup for the new address (0).

3.9.5 Actual Position (AP)

Command AP

Modes 1, 2, 3, 4, 5

Range -1073741824 to 1073741823 pulses

Description The motor position can be read at any given time. The position is given in terms of encoder pulses relative to the zero-point. The motor position can also be "reset" by specifying an argument to the AP command. It is recommended that the position is only changed when the motor is stationary.

Usage **AP = x** Set motor's current position to x.
AP Show motor's position in pulses.

3.9.6 Rated Current (CA)

Command CA

Modes 1, 2, 3, 4, 5

Range 1 - 12 Amp.

Description To protect the motor from overload and to ensure that the lifetime is not reduced, a maximum rated current and maximum peak current are specified. The CA command is used to specify the maximum rated current. The system will automatically reduce the motor voltage when the specified rated current is exceeded. The voltage is reduced linearly until the current is exceeded by 10%, after which the motor stop completely. CA is also used to set the maximum torque in Torque Mode.
See also : *CB9 - Multiply all current settings/readings by 10* page 51

Usage **CA=xx** Set Rated Current in Amp.
CA Show actual rated current value.

3.9

Commands

3.9.7 Control Bits in General (CB)

The Controller contains a number of control bits. These bits control some basic parameters/functions in the Controller. Mostly these parameters or functions are special compared to the standard functions described in this chapter. For example, a bit can control whether a certain input should be activated at logic 1 or logic 0. Some of the bits can only be read. These bits show the status of different conditions during operation — for example, in which direction the motor is moving.

The following Control bits are available.

3.9.8 CB1 - Disable motor using IN2

Command CB1

Modes 1, 2, 4, 5

Selection CB1=0 Function disabled (default).
CB1=1 Function active. Motor will be disabled if IN2=1.

Description The motor output can be disabled by activating user input 2 (IN2). If CB1=0 (default) the function is disabled and IN2 can be used as a standard input. If CB1=1 the motor will operate normally as long as IN2 is passive. If IN2 is activated, the motor output will immediately be set current less. Please note that CB1 does not have any function when the Controller is set to mode 3.

3.9.9 CB2 - Reset Controller using IN3

Command CB2

Modes 1, 2, 4, 5

Selection CB2=0 Function disabled (default).
CB2=1 Function active. Controller will be reset if IN3=1.

Description If a fatal error occurs, such as a motor blockage, it can be an advantage to be able to reset the system from an external source. Normally a reset can only be done by sending the RESET command via the RS232/RS485 interface or alternatively switching off the power. If CB2 is set to 1, the Controller can be reset using IN3. Please note that CB2 does not have any function when the Controller is set to mode 3.

3.9.10 CB3 - Invert error output (O2)

Command CB3

Modes 1, 2, 3, 4, 5

Selection CB3=0 Output 2 is only active if a fault has occurred (default).
CB3=1 Output 2 is inverted.

Description The user output 2 is used by default as an error output. The output is activated if a fatal error has occurred. The output can be inverted, which means that it is set passive if an error has occurred. Set CB3 = 1 to invert the output. Please note that changing CB3 will first have effect after saving and resetting the Controller using the MS and RESET command.

3.9

Commands

3.9.11 CB4 - Send Y+Address if IN4 changes

Command CB4

Modes 1, 2, 3, 4, 5

Selection CB4=0 Function disabled (default).
CB4=1 Function enabled. Y + Address sent if IN4 changes from logic 0 to logic 1.
CB4=2 Function enabled. Y + Address sent if IN4 changes from logic 1 to logic 0.

Description The CB4 control bit can enable a Y and the Address number to be sent on the RS232/RS485 interface every time user input 4 (IN4) changes level. CB4 set to 1 or 2 is used to select if the transmission is done at the rising or falling edge.
If no address is used (Address=0), only a Y is send.

3.9.12 CB5 - Send "Y" when using addressing

Command CB5

Modes 1, 2, 3, 4, 5

Selection CB5=0 An answer (Y) is never sent when using addressing (default).
CB5=1 An answer (Y) is always sent when using addressing.

Description The Controller always sends an answer (typically Y) when receiving commands and when addressing is not used (Address=0).
If addressing is used (Address>0), by default the Controller will not respond when commands are sent.
Setting CB5=1 will enable an answer to be sent every time a command is received by the Controller, regardless of whether the Controller is set up for addressing or not.

3.9.13 CB6 - Stop motor (set velocity to 0)

Command CB6

Modes 4, 5

Selection CB6=0 Function disabled (default).
CB6=1 Stop motor if IN5 is activated.

Description If this function is selected (CB6=1), the motor will be stopped if user input 5 (IN5) is activated.
The function internally sets the value measured from the analog input to 0.
Note that the CB6 function is only active in modes 4 and 5 (speed and torque mode).

3.9

Commands

3.9.14 CB7 - Invert the analog input

Command CB7

Modes 4, 5

Selection CB7=0 Function disabled (default).
CB7=1 Invert the analog input if IN6 is activated.

Description If this function is selected, activation of user input 6 (IN6) will invert the polarity of the analog input, which will make the motor move in the opposite direction.
Note that the CB7 function is only active in modes 4 and 5 (speed and torque mode).

3.9.15 CB8 - Use O1 to indicate hysteresis window

Command CB8

Modes 4, 5

Selection CB8=0 Function disabled (default).
CB8=1 Activate O1 if the voltage at the analog input is within the hysteresis window specified by VVH.

Description When CB8 is set to 1, the user output 1 (O1) will be activated if the voltage at the analog input is within the hysteresis window specified in the VVH register.
This function makes it possible to detect if the voltage at the analog input is high enough to make the motor run.

3.9.16 CB9 - Multiply all current settings/readings by 10

Command CB9

Modes 1, 2, 3, 4, 5

Selection CB9=0 Function disabled (default).
CB9=1 Function enabled - All current settings/readings are multiplied by 10.

Description If CB9 is set to 1, all current settings and readings will be multiplied by 10.
The following commands are covered:
- CA *Rated Current (CA)* page 48.
- CP *Peak Current (CP)* page 54.
- CU *Motor Current (CU)* page 54.
Please note that if CB9 is changed, the MS and RESET command must be used before the change is effected.

Example A motor is specified for an average current of 5.3A. CA can only be set to 5A so CB9 is set to 1 and register CA is set at 53, which is normally an illegal setting (53A is above maximum) but by setting CB9=1, the internal value will be 5.3A.
If for example the CU command (show actual motor current) is used and a reading of CU is 23 is obtained, this means that the actual motor current is 2.3A.

3.9

Commands

3.9.17 CB10 - Replace servo loop with voltage control

Command CB10

Modes 4

Selection CB10=0 Function disabled (default).
CB10=1 Function enabled. Control motor voltage directly from analog input.

Description When CB10=1, the normal servo loop is bypassed and the voltage applied to the motor is directly proportional to the voltage at the analog input. The velocity at the motor will thus be controllable but the performance is very poor compared to normal operation using the complete servo loop.
The voltage control feature does not use the encoder at the motor which means that a pure DC motor without feed-back can be used.
The basic idea of using this feature is lower cost - there is no need for encoder feedback at the motor.

Example CB10 is set to 1 (function active).
The velocity register VM is set to 1000 RPM.
A voltage of +5V is applied to the analog input. The input is calibrated for +/-10 volt as full scale limits (default). Set the Controller in mode 4.
With this set-up, the motor will now be applied with 50% of the supply voltage. The voltage will be applied with a positive sign which means that the motor rotates clockwise.
If -2V is applied to the input, the Controller will generate 20% voltage to the motor but with negative sign, which means that the motor rotates counter-clockwise.
When the CB10 function is active, the motor test feature (EMT) is not supported and must therefore be disabled by setting EMT=0 - see also *Encoder/Motor Test flag (EMT)* page 56.

3.9.18 CB11 - Avoid changing AP after zero-search

Command CB11

Modes 2, 3

Selection CB11=0 Function disabled (default).
CB11=1 Do not reset AP after zero-search.

Description When a zero-search takes place, AP will by default be set to 0 once the zero-search sensor is found and the motor is stationary. If it is required that only the motor is to be moved to the mechanical zero position but AP should not be reset or manipulated, the CB11 function can be used.

3.9

Commands

3.9.19 CB12 - Soft Halt using IN5

Command CB12

Modes 2

Selection CB12=0 Function disabled (default).
CB12=1 Function enabled. Motor will be soft halted by rising edge at IN5.
CB12=2 Function enabled. Motor will be soft halted by falling edge at IN5.

Description This function is equivalent to the SH command but instead of sending the command via the RS232/RS485 interface, user input 5 is used.
If a rising or falling edge is applied to this input, the motor will decelerate according to the setting in the AC register.
See also *Soft Halt (SH)* page 71.

3.9.20 Interface checksum (CHS)

Command CHS

Modes 1, 2, 3, 4, 5

Selection 0 = no, 1 = yes

Description As described in *Checksum* page 26, a checksum facility can be used for communication via the interface.

Usage **CHS=x** 0=do not use checksum, 1=use checksum.

CHS Show checksum status.

3.9

Commands

3.9.21 Peak Current (CP)

Command CP

Modes 1, 2, 3, 4, 5

Range 1 - 25 Amp.

Description To protect the motor from overload and to ensure that the lifetime is not reduced, a maximum rated current and maximum peak current are specified. The system can withstand currents for short periods that are higher than the maximum allowable rated current. The motor must however be protected from high current peaks. The CP command is used to set the maximum peak current to the motor.
See also : *CB9 - Multiply all current settings/readings by 10* page 51

Usage **CP=xx** Set peak current in Amp.
CP Show actual peak current setting.

3.9.22 Motor Current (CU)

Command CU

Modes 1, 2, 3, 4, 5

Description The motor current consumption in Amps can be read using this command.
See also : *CB9 - Multiply all current settings/readings by 10* page 51

Usage **CU** Show motor current consumption in Amps.

3.9.23 Current Velocity (CV)

Command CV

Modes 1, 2, 3, 4, 5

Description The motor velocity can be read at any given time.

Usage **CV** Show velocity in rpm.

3.9

Commands

3.9.24 Digital Input Format (DIF)

Command DIF

Mode 3 (1, 2, 3, 4, 5)

Selection 1 - 3 (default 1)

Description The DIF command makes it possible to select the functionality of user inputs IN5 and IN6 when the Controller is operating in mode 3 - See also *Register Mode (MO=3)* page 36 for further details.

DIF=1 As default, the DIF register is set to 1, which means that the motor runs when IN5 is activated and the motor will stop if IN5 is passive.

When using this format, all positioning will be relative or absolute.

DIF=2 Used in applications where a separate start and stop input is preferred.

When DIF is set to 2, the position counter will be set to 0 every time a new start is made. This means that every move is relative.

The motor starts when IN5 is activated and stopped when IN6 is activated regardless if IN5 is still active.

DIF=3 Same function as DIF=1 (if using mode 3) but user output 4 (O4) is controlled by the relationship between the actual position (AP) and the value contained in the X-register 15 (XP15).

If the motor is moving in a positive direction and AP becomes a higher value than XP15, user output 4 will be activated.

If the motor is moving in a negative direction and AP becomes a lower value than XP15, user output 4 will be activated.

In both situations, user output 4 will stay active until the command OUT4=0 is sent via the RS232/485 interface or user input 4 is activated.

If user output 4 is cleared by one of these methods, it will not be activated again unless AP passes the value in XP15 again or the contents of XP15 are re-defined/overwritten.

If DIF=3 is used in mode 3, only X0 to X7 are usable. X8 to X14 (X15) are passive and cannot be accessed.

DIF=3 can be used in all modes.

Input	DIF=1	DIF=2	DIF=3
IN4	Normal use	Normal use	Clear output 4 (O4)
IN5	Start / Stop	Start	Start/Stop
IN6	Home sensor	Stop / Home sensor	Home sensor
OUT4	Normal use	Normal use	Controlled by AP / XP15

Usage **DIF** Show actual input format.

DIF=x Set input format.

3.9

Commands

3.9.25 Encoder/Motor Test flag (EMT)

Command EMT

Modes 1, 2, 3, 4, 5

Selection

Setting	Function	Test direction
EMT=0	Motor test disabled	-
EMT=1	Motor test enabled	Counter clockwise
EMT=2	Motor test enabled	Clockwise

Description If EMT is set to 1, the Controller will automatically test the encoder and motor connections at power up. The Controller must move the motor to perform the test. If e.g. the motor is blocked by a brake, the test will fail, and EMT must be set to 0.

Usage **EMT** Show Encoder/motor test flag.
EMT=x Set Encoder/motor test flag.

3.9

Commands

3.9.26 Read-out of Error Status (ES)

Command ES

Modes 1, 2, 3, 4, 5

Selection 0 and 1

Description During operation of a system, various error conditions can arise. Some errors can be attributed to communication and set-up (error status register 0) and others attributed to hardware and motor control errors. The error status can be read using the ES (Error Status) command. The command invokes the Controller to transmit a series of zeroes (0) and ones (1). A quick overview of error messages is thus obtained which can also be interpreted by other software. Using the command EST an overview of text responses is obtained.

There are two error status registers.

Register 0 provides information about RS232 communication and set-up errors. This register stores all errors since the register was last read. When the register is read, the information is automatically cleared.

Bit no.	E no.	Explanation
0	E1	Error
1	E2	Value out of allowable range
2	E3	Incorrect number of parameters
3	E4	Unknown command (command does not exist)
4	E5	Not a command
5	E6	Error in parameter or value out of allowable range
6	E7	Error in register number or value out of allowable range
7	E8	Data cannot be stored in EEPROM
8	E9	Error in command checksum
15	E16	See status register 1

Register 1 provides information about Controller and motor errors. Some error conditions may be temporary, for example the maximum peak current may have been exceeded for a short duration and the corresponding bit is set in the status register. The error indication is cleared after reading the error status. For critical errors, motor operation is interrupted and the error information remains in the register, and O2 is set high (=1). The user must then either switch the system off and on to reset the error status, or use the RESET command.

3.9

Commands

Bit no.	Error status cleared by reading	O2 is set high	System must be reset	E no.	Explanation
0	No	Yes	Yes	E20	Temperature > 80°C
1	No	Yes	Yes	E21	Current overload
2	Yes	No	No	E22	Peak current exceeded
3	Yes	No	No	E23	Rated current exceeded
4	Yes	No	No	E24	Supply voltage exceeded 89 V
5	No	No	No	E25	Negative end-of-travel active
6	No	No	No	E26	Positive end-of-travel active
7	No	Yes	Yes	E27	Motor incorrectly connected
8	No	Yes	Yes	E28	Encoder not connected
9	No	Yes	Yes	E29	Supply voltage exceeded 95 V
10	No	Yes	Yes	E30	Motor not connected
11	No	No	No	E31	Rated current cannot be read
15	No	No	No	E16	See status register 0

Usage **ES0** Show error status register 0.

Example Sent to Controller `ES0`
Received from Controller `ES0=000000001000101`

Note: bit 0 is the last bit to the right.

3.9

Commands

3.9.27 Error Status Text (EST)

Command EST

Modes 1, 2, 3, 4, 5

Selection 0 or 1

Description The EST command has exactly the same function as the ES command described above, with the exception that the error status is reported as text. Using the EST command produces a list of the various errors in English. If there are no errors, the error response is *EO: No errors*. A list of error messages is given in *Error Messages* page 78.

Usage

EST0	Read out error status register 0 as text.
EST1	Read out error status register 1 as text.
EST	Read out all errors in register 0 and register 1 as text.

3.9.28 Encoder type (ET)

Command ET

Modes 1,2, 3, 4, 5

Selection 0=PNP or 1=NPN

Description To achieve correct positioning and precise velocity and acceleration, it is important that the encoder set-up is correct. The encoder may be either a PNP or an NPN type. In addition, both a balanced and unbalanced signal from a standard 2-channel incremental encoder may be used. For details of encoder connection, see *Encoder Input* page 19.

This command is used to specify the type of encoder connected to the Controller. If an encoder with a balanced output is used, the setting of ET can be omitted. If however an unbalanced NPN encoder is used, ET must be set to 1 (ET=1). If an unbalanced encoder of the type PNP is used, ET must be set to 0 (ET=0).

Usage

ET=x	Set encoder type.
ET	Show encoder type.

3.9.29 Halt of motor (H)

Command H

Modes 2, 3

Description This command is used to instantaneously stop the motor, regardless of velocity, etc. Please notice that the Halt (H) command is disabled during zero-search.

Usage

H	Halt motor.
----------	-------------

3.9

Commands

3.9.30 Command Overview (HELP)

<u>Command</u>	HELP
<u>Modes</u>	1, 2, 3, 4, 5
<u>Description</u>	The HELP command is used to display an alphabetic list of the commands that can be used with the Controller.
<u>Usage</u>	HELP Show commands.
<u>Example</u>	Sent to Controller HELP Received from Controller Following Instructions can be used AC ADDR AP CA CHS

3.9.31 Integral Summation Limit IL

<u>Command</u>	IL
<u>Modes</u>	1, 2, 3, 4, 5
<u>Range</u>	0 - 25000
<u>Description</u>	K_i has the same effect on instantaneous errors as K_p . K_i normally determines how the system reacts to persistent errors, but in order for the system to react quickly to changes, the value of K_i must be increased. In order that the summed error does not exceed an unacceptably high level, an integral summation limit (IL) is specified. For example in a system with large mass, it can be difficult for the motor to respond to the desired velocity profile. It may therefore be desirable to set K_i to a high value so that a quick response to a positioning error is obtained. When the motor does not follow the required velocity profile, and a high value for K_i is used, the accumulated error must be limited. If the accumulated error is not limited, the system will become unstable and the motor velocity will be too high when the required position is reached. The system should first be adjusted without adjusting I_L . I_L can then initially be set to 1500 in pulse mode (MO=1), positioning mode (MO=2) and register mode (MO=3) and to 10000 in velocity mode (MO=4) and torque mode (MO=5).
<u>Usage</u>	IL=x Where x specifies the integral summation limit. IL Show value of IL.

3.9 Commands

3.9.32 Read Status of Inputs (IN1 - IN8)

Command IN

Modes 1, 2, 3, 4, 5

Description The DMC10 has 8 inputs. The status of these can be read using the following commands.

Input	Function		
	Step-pulse Mode (MO=1)	Register Mode (MO=3)	All other modes
IN1	Multiplication on/off	Least significant bit	
IN2		Next least significant bit	
IN3		Next most significant bit	
IN4		Most significant bit	
IN5		DIF=1 : Start/stop (pause) DIF=2 : Start	
IN6		DIF=1 : Home sensor DIF=2 : Stop	
IN7	Negative end-of-travel	Negative end-of-travel	Negative end-of-travel
IN8	Positive end-of-travel	Positive end-of-travel	Positive end-of-travel

Usage **IN** Read inputs.

Example Sent to Controller IN
Received from Controller IN=00010100 Note that IN8 is the leftmost digit (MSB)

3.9.33 Show Servo Constants K

Command K

Modes 1, 2, 3, 4, 5

Description The **K** command can be used to check the values of the system K and IL values.

Usage **K** Show all three K values and the integral summation limit IL.

3.9.34 Constant K_d

Command KD

Modes 1, 2, 3, 4, 5

Range 0 - 65535

Description The Differential Constant has particular influence on the system when changes occur, such as during an acceleration or deceleration.

Usage **KD=x** Set KD to value x.
KD Show KD set-up value.

3.9

Commands

3.9.35 Constant K_i

Command KI

Modes 1, 2, 3, 4, 5

Range 0 - 65535

Description The Integral Constant influences how aggressively the system reacts to persistent errors.

Usage **KI=x** Set K_i to value x.
KI Show K_i set-up value.

3.9.36 Constant K_p

Command KP

Modes 1, 2, 3, 4, 5

Range 0 - 65535

Description The Proportional Constant determines how the system will react to instantaneous errors. A high K_p value causes the system to react quickly to a given error.

Usage **KP=x** Set K_p to value x.
KP Show K_p set-up value.

3.9.37 Mode Selection (MO)

Command MO

Modes 1, 2, 3, 4, 5

Range 1 - 5

Description Control of a motor can be made in one of 5 basic modes of operation, as outlined in the table below. The MO command is used to select the Controller Mode.

Usage **MO=x**

Mode no.	Mode
1	Step-pulse
2	Positioning
3	Register
4	Velocity
5	Torque

MO Show current mode of operation.

3.9

Commands

3.9.38 Multiply Pulses - Step-pulse Control (MP)

Command MP

Mode 1

Range 1 - 10

Description In situations where the control system (PLC, PC-board, etc.) cannot supply a sufficiently high pulse frequency, the DMC 10 can be used to increase the number of control pulses. The DMC10 can multiply the number of incoming pulses by a factor of between 1 and 10, as specified by the MP command. In Pulse Mode (Mode=1) the multiplication factor is enabled or disabled using the IN1 input. If IN1 is 0 (low), the pulses are not multiplied. If IN1 is 1 (high), the pulses are multiplied by the factor specified by the MP command.

Usage **MP = x** Set the pulse multiplication factor x.

MP Show the current value of the multiplication factor.

3.9.39 Save Set-up (MS)

Command MS

Modes 1, 2, 3, 4, 5

Description The Controller set-up data can be permanently stored in non-volatile EEPROM memory, i.e. without the need for an external supply to store data. The MS command is used to store the Controller set-up.

Usage **MS** Save set-up in EEPROM

3.9.40 Recall Setup (MR)

Command MR

Modes 1, 2, 3, 4, 5

Range 0 - 3

Description Controller set-up data can be permanently stored in non-volatile EEPROM memory, i.e. without the need for an external supply to store data. The Memory Recall command MR is used to recall set-up data from the EEPROM memory. It is possible to recall the set-up data automatically at power up.

Usage **MR=x** Restore status: 0=do not recall data on power up.
1=recall data at power up.
2=recall data now, but not at power up.
3=recall data now and at power up.

MR Show restore status. The response given is 0 or 1 only.

3.9

Commands

3.9.41 Encoder Control (MU)

Command MU

Modes 1, 2, 3, 4, 5

Selection 1, 2 or 4

Description To achieve greater precision in positioning with any given encoder, the system can provide increased encoder resolution. The system can multiply the incoming encoder pulses by a factor of 1, 2, or 4, using the value specified by the MU command. If for example MU is set to 4 (MU=4), an encoder with 500 pulses/revolution will have an effective resolution of 2000 pulses/revolution.

Usage **MU**=x Where x is the encoder pulse multiplication factor.

MU Show current multiplication factor.

Example An encoder with 1000 pulses/revolution is used. MU=2 and MP=1. In Step-pulse Mode (MO=1) 2000 pulses must be applied to the pulse input to rotate the motor 1 revolution.

3.9.42 Negative Limit Switch (NLS)

Command NLS

Modes 1, 2, 3, 4, 5

Selection 0=low or 1=high

Description Inputs IN7 and IN8 function as end-of-travel limit switches. If the motor is moving in a negative direction and IN7 is activated, the motor is stopped instantaneously. IN8 is the positive end-of-travel limit. The two limit switches can be independently programmed to be active high (1=PNP sensor) or active low (0=NPN sensor). For connection of the end-of-travel limits, see *Connections* page 15.

Usage **NLS**=x Set Negative Limit Switch to level 0=low 1=high.

NLS Show Negative Limit Switch level.

3.9

Commands

3.9.43 Read Status of Outputs (O1 - O4)

Command OUT

Modes 1, 2, 3, 4, 5

Description The DMC10 has 4 outputs. Via the RS232 interface it is possible to reset Output 3 (O3) and Output 4 (O4) using the OUT3 and OUT4 commands respectively. Output 1 (O1) and Output 2 (O2) are set automatically by the Controller. The status of the outputs can be read using the OUT command. When the status of the outputs is read, information is also given about the status of the 4 control LEDs.

Bit no.	Output	Function
0	O1	1 = in position (only used in modes 2 and 3)
1	O2	0 = no error, 1 = vital error
2	O3	Output 3 can be set via the OUT3 command
3	O4	Output 4 can be set via the OUT4 command
4	T > 80°C LED	The Controller's internal temperature is too high
5	Current LED	Current limitation has been exceeded
6	O. Voltage LED	Limit for supply voltage has been exceeded
7	Running LED	Motor operating

Usage

OUT	Read status of outputs
OUTn	Read status of output n
OUTn=x	Set output n (On) to x (0 or 1)
OUT=xxxxxxxx	Set all outputs to x, where x is 0 or 1 (Only bit 2 and 3 are changed).

Example

Sent to Controller	OUT	Read outputs
Received from Controller	OUT=00000000	Note bit 0 is the rightmost digit (LSB).

Sent to Controller	OUT3=1	Set O3 to 1
Received from Controller	Y	Controller has accepted the command and output 3 is now activated.

3.9

Commands

3.9.44 Maximum Pulse Error (PE)

Command PE

Modes 1, 2, 3

Range 0 - 1073741823 pulses

Description As an additional precaution, a maximum allowable pulse error can be specified. If the error between the desired position and the actual position is too large, the encoder may be at fault. If the pulse error exceeds the specified maximum, the motor is stopped and 3 LEDs flash. The PE command can be used in Step-pulse Mode (MO=1), Positioning Mode (MO=2) and Register Mode (MO=3).

Usage **PE=x** Set Pulse Error

PE Show current maximum Pulse Error limit

3.9.45 Positive Limit Switch (PLS)

Command PLS

Modes 1, 2, 3, 4, 5

Selection 0=low or 1=high

Description Inputs IN7 and IN8 function as end-of-travel limit switches. If the motor is moving in a negative direction and IN7 is activated, the motor is stopped instantaneously. IN8 is the positive limit switch. The two end-of-travel limits can be independently programmed to be active high (1=PNP sensor) or active low (0=NPN sensor). For connection of the end-of-travel inputs, see *Connections* page 15.

Usage **PLS=x** Set Positive Limit Switch to level 0=low 1=high.

PLS Show current PLS level.

3.9.46 PWM Mode (PM)

Command PM

Modes 1, 2, 3, 4, 5

Selection 0 or 1

Description This parameter determines the way in which the driver regulates current to the motor. Normally the PM command is set to 0.

Usage **PM=x** Set pwm

PM Show current PM value

3.9

Commands

3.9.47 Motor Power ON/OFF (PO)

Command PO

Modes 1, 2, 3, 4, 5

Selection 0 = ON (motor power on) or 1 = OFF (motor power off)

Description This command enables the motor power to be switched off, for example during adjustment of the analog input in Velocity Mode (MO=4) and Torque Mode (MO=5).
An alternative to the PO register is the CB1 control bit which makes it possible to de-energize the motor using an input. See also *CB1 - Disable motor using IN2* page 49.

Usage **PO**=x Set voltage on/off

PO Show current value.

3.9 Commands

3.9.48 Encoder Pulses (PR)

Command PR

Modes 1, 2, 3, 4, 5

Range 50 - 20000

Description To obtain correct velocity and acceleration, the number of encoder pulses per revolution must be programmed. The PR command is used to set the Controller for the encoder's specified pulse rate (independent of any setting using the MU command).

Important ! : When changing the value of PR, the Controller must be reset before the change takes full effect. Use the MS and RESET commands to restart the Controller.

Usage PR = x Set pulses per revolution.
PR Show current setting.

3.9.49 Position Width (PW)

Command PW

Modes 2, 3

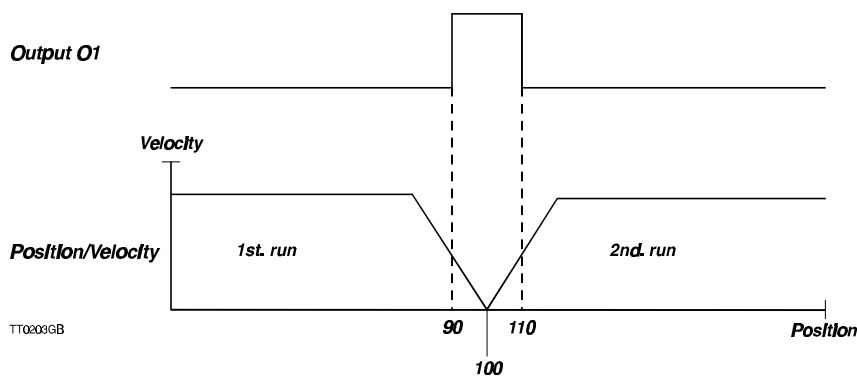
Range 0 - 20000 pulses

Description How precisely a motor can be positioned depends on load, the K parameters, the encoder resolution etc. The Controller therefore allows a Position Width to be specified. If the actual motor position is within the range of desired position (Position Width, output O11 is high).

Usage PW=x Where x specifies the Position Width
PW Show Position Width

Example Sent to Controller PW=10 Set Position Width to ± 10 pulses, i.e. output O1 is high providing the encoder position is within the required position ± 10 .

Received from Controller \surd



3.9

Commands

3.9.50 Reset Controller (RESET)

Command Reset

Modes 1, 2, 3, 4, 5

Description If a system overload occurs, for example if the supply voltage is too high (see ES), the system must be reset before motor control is possible again. The Reset command has the same function as turning the Controller off and then on again. The set-up values can be stored (using MS) before the Reset command is used.
See also *CB2 - Reset Controller using IN3* page 49.

Usage **RESET** Reset Controller.

3.9.51 Report Motor Status (RS)

Command RS

Modes 1, 2, 3, 4, 5

Range 0 - 3

Description During operation, the system can report the status of the motor (stationary, running, etc.).

Usage **RS** Motor Status: 0=stationary, 1=accelerating, 2=max velocity, 3=decelerating

3.9

Commands

3.9.52 Default Setup (SD)

Command SD

Modes 1, 2, 3, 4, 5

Description The SD command is used to recall the Controller's factory default set-up.

Factory default set-up is as follows:

AC = 500	MO = 2	VVH = 0
ADDR = 0	MP = 1	VVL = -500
CA = 5	MU = 4	VVO = 0
CHS = 0	NLS = 1	VVU = 500
CP = 10	PE = 1073741823	XA0-15 = 0
ET = 1	PLS = 1	XP0 = 1
IL = 1500	PM = 0	XP1-15 = 0
KD = 2	PR = 5000	XR0 - 15 = 0
KI = 2	PW = 30	XV0 - 15 = 0
KP = 2	SZT = 0	VM = 100
		ZL = 1

Usage **SD** Recall system default set-up

3.9

Commands

3.9.53 Soft Halt (SH)

Command SH

Modes 2, 3

Description This command is used to initiate a controlled halt of the system. The motor is stopped according to the pre-programmed deceleration (acceleration).
See also *CB12 - Soft Halt using IN5* page 53.

Usage **SH** Soft halt.

3.9.54 Set new Position (SP)

Command SP

Modes 2, 3

Range -1073741824 - 1073741823 pulses

Description In Positioning Mode (MO=2) and Register Mode (MO=3), the motor can be set to move to a new position specified in pulses.

Usage **SP = x** Move to new Position.

SP Show new position.

Example Sent to Controller `SP=-1000` Move to absolute position -1000
Received from Controller `Y`

3.9.55 Relative Positioning (SR)

Command SR

Modes 2, 3

Range -1073741824 - 1073741823 pulses

Description In position mode the motor can be set to move a specified number of pulses in a positive or negative direction, relative to the current position.

Usage **SR = x** Set relative position

Example Sent to Controller `SR=5000` Move 5000 pulses in a positive direction
Received from Controller `Y`

3.9

Commands

3.9.56 Zero-point Seek (SZ)

Command SZ

Mode 2 and 3

Description In modes 2 and 3, a zero-point seek can be initiated using the command SZ. The zero-point seek is started in accordance with a profile determined by register 0. See therefore the XR0 command for further information. When the SZ command is sent, the zero-point seek is initiated immediately.
See also *CB11 - Avoid changing AP after zero-search* page 52.

Please notice that the Halt (H) command is disabled during zero-search.

Usage **SZ** Perform zero-point seek

3.9.57 Zero-point Seek Type (SZT)

Command SZT

Modes 2 and 3

Selection 0= zero-point contact or 1=index pulse

Description The motor can be brought to a known mechanical reference position. This is done using a sensor connected to IN6 (zero-point contact). If SZT is set to 0, the result of the seek process is that the motor is positioned precisely at the zero-point contact. If SZT is set to 1 the result of the seek process is that the motor is positioned precisely at the first index pulse after zero-point contact have been found.

Usage **SZT=x** Set Zero-point Seek Type to x

SZT Show current Zero-point Seek type.

3.9.58 Firmware Version (VE)

Command VE

Modes 1, 2, 3, 4, 5

Description Provides information on the Controller firmware version and date.

Usage **VE** Show version and date.

3.9

Commands

3.9.59 Maximum Velocity (VM)

Command VM

Modes 2, 3, 4, 5

Range 3 - 65535 rpm

Description The VM command is used to set the maximum velocity.

In Positioning Mode (MO=2) and Register Mode (MO=3), VM is used to set the velocity that the motor is accelerated to and maintains until it is decelerated. See also *Acceleration (AC)* page 47.

In Velocity Mode (MO=4) VM sets the limit for velocity for maximum input at the analog input. If for example VM is set to 1000 and the analog input is adjusted to an input voltage in the range -10V to +10V, the motor will rotate at 500 rpm in a negative direction for an applied voltage of -5V.

In Torque Mode (MO=5) VM is used to set a limit for the motor. Control of the velocity in this mode is not precise and is used only as an additional measure.

VM has no effect in Pulse Mode (MO=1).

Usage **VM** = x Set maximum velocity in rpm.

VM Show current max. velocity

3.9.60 Supply Voltage (VOL)

Command VOL

Modes 1, 2, 3, 4, 5

Range 12 - 100

Description The VOL command is used to check the supply voltage applied to the Controller.

Usage **VOL** Show voltage in Volts

3.9

Commands

3.9.61 Analog Input - Hysteresis (VVH)

Command VVH

Modes 4, 5

Range 0 - 200 ADC steps

Description The VVH command is used to define a region around the analog voltage zero point in which the motor must not move. The range is symmetrical around the zero point (twice the value specified). The value is specified as the number of AD converter steps. The converter has an operating range of 2048 steps (11 bit), i.e. with an adjustment of -10V to +10V at the input, approximately 10 mV per step is obtained. See *Adjustment of Analog Input* page 45 for more information on the use of this command. See also *CB8 - Use O1 to indicate hysteresis window* page 51

Usage **VVH**=x Where x specifies the hysteresis value

VVH Show hysteresis value and value of the three calibration commands.

3.9.62 Analog Input - maximum Negative (-10V) Value (VVL)

Command VVL

Modes 4, 5

Description Calibrate full-scale — set negative voltage (max. -10V) at the analog input and send the VVL command. The Controller will then calibrate the analog input's negative value. The negative input voltage must not be greater than the zero-point voltage. See *Adjustment of Analog Input* page 45 for more information on the use of this command.

Usage **VVL** Maximum negative voltage is calibrated.

3.9.63 Analog Input - Zero-point Voltage (VVO)

Command VVO

Modes 4, 5

Description This command is used to determine the analog input zero-point voltage. To calibrate the Controller, the zero-point voltage should be applied to the input and the command sent. In the majority of cases, the zero-point voltage will be 0 Volt, but this is not a requirement however. The zero-point voltage must lie in the range from the maximum negative voltage to the maximum positive voltage. See *Adjustment of Analog Input* page 45 for more information on the use of this command.

Usage **VVO** Zero-point voltage is calibrated

3.9

Commands

3.9.64 Analog Input - Maximum Positive (+10V) Value (VVU)

Command VVU

Modes 4, 5

Description Calibrate full-scale — set positive voltage (max. +10V) at the analog input and send the VVL command. The Controller will then calibrate the analog input's positive voltage. The positive voltage must not be less than the zero-point voltage. See *Adjustment of Analog Input* page 45 for more information on the use of this command.

Usage **VVU** Maximum positive voltage is calibrated

3.9.65 Read Analog Input (VV)

Command VV

Modes 1, 2, 3, 4, 5

Description This command is used to read the Controller's analog input directly. The returned value is given in AD-converter steps.

Usage **VV** Read analog input in ADC steps

3.9.66 Show all Parameter Set Values (X)

Command X

Mode 3

Description The X command can be used to obtain a quick overview of all the values in the 16 parameter sets.

Usage **X** Show all parameter sets

3.9.67 Acceleration in Parameter Sets (XA)

Command XA

Mode 3

Range (0) 100 - 100000 rpm/sec.

Description A required acceleration can be set for each parameter set. If the acceleration is set to 0, the acceleration will not be changed by selecting the parameter set, i.e. the previous acceleration value will be used.

Usage **XAn**=xxxxx Set acceleration in parameter set n to xxxxx rpm/sec.
XAn Show acceleration in parameter set n
XA Show all acceleration values

3.9

Commands

3.9.68 Position in Parameter Sets (XP)

Command XP

Mode 3

Range -1073741824 to 1073741823 pulses
Position register 0 (XP0): -1 = negative direction, 1 = positive direction.

Description A required position can be set for each parameter set. If the position is set to null, no change in position will occur but the acceleration and velocity will be changed. Note! XP1-15 do not have the same meaning as XP0, which is used in conjunction with zero-point seek. The set-up of the position variable in parameter set 0 (XP0) determines in which direction the zero-point seek will occur. See section Zero-point Seek Function page 43.

Usage **XPn=xxxxx** Set Position parameter to xxxxx pulses for parameter set n
XPn Show position
XP Show position values for all parameter sets

3.9.69 Relative Positioning in Parameter Sets (XR)

Command XR

Modes 3

Selection 0=absolute, 1=relative, for register 0: 0=do not seek, 1=seek zero-point

Description The relative positioning parameter set (XR) contains information about whether the required position is relative or absolute. XR0 has a different function than the other registers. XR0 determines whether a zero-point seek should be carried out when the DMC is switched on: 0=do not seek, 1=seek. If the Controller is in Register Mode (MO=3) and XR0=1, at power up an automatic zero-point seek will be carried out in a positive or negative direction (as specified by XP0). See section Zero-point Seek Function page 43.

Usage **XRn=x** x specifies whether the position is absolute (0) or relative (1)
XRn Show relative positioning set-up in parameter set n
XR Show all relative values

3.9

Commands

3.9.70 Velocity in Parameter Set (XV)

Command XV

Mode 3

Range (0) 3 - 65535 rpm

Description A required maximum velocity can be set for each parameter set. If the velocity is set to null, the velocity will not be changed when the parameter set is selected, i.e. the previous velocity setting will be re-used.

Usage **XVn=xxxxx** Set maximum velocity in parameter set n to xxxxx rpm.

XVn Show velocity value in parameter set n

XV Show all velocity values for all parameter sets.

3.9.71 Zero-point Input (ZL)

Command ZL

Mode 3

Selection 0 or 1

Description The zero-point contact is connected to input IN6. The input may be active high (1), if a PNP sensor is used or low (0) if it is an NPN sensor.

Usage **ZL=x** Set level for zero-point contact, 0 = low, 1 = high.

ZL Show level.

3.10

Error Messages

When an error occurs in communication with the DMC 10 or an internal error occurs, the Controller transmits an error message. The error message consists of an "E" followed by an error number, followed by a colon ":", followed by a descriptive text. The following illustrates an example of an error message:

Example: E2: Out of range

3.10.1 Description of Error Messages

E0: No errors

No errors have occurred since the last request.

E1: Error

The command string cannot be understood.

Example:

KP 8 K

Results in error E1.

Correction:

Carefully check the command sent to the Controller and compare with the command description given in this manual.

E2: Out of range

The command Parameter Value is not within the legal range.

Example:

CP=100

The above command attempts to set the peak current to 100 Amp, which is outside the allowable range. The Controller therefore reports an E2 error.

Correction:

Select a value within the allowable range.

E3: Number of parameters is wrong

The number of parameters specified is incorrect.

Example:

KP8 or ES0=9

Both of the above commands will produce an E3 error.

Correction:

The KP has only one register and can therefore only be called by specifying KP.

The ES0 command is only used to show information and therefore specifying a parameter has no meaning.

E4: Instruction does not exist

There is no such command.

Example:

ABCDEF

Correction:

Use a valid Controller command. See the description of the command for details of the required syntax.

E5: It is not an instruction

The Controller has received an incorrect command.

Example:

4R

If the Controller is not using addressing, this command will result in error message E5.

Correction:

Use correct command.

E6: Parameter error or out of range

The Controller has received an incorrect command.

Example:

4R

If the Controller is not using addressing, this command will result in error message E5.

Correction:

Use correct command.

E7: Register number error or out of range

Error register number.

Example:

XP7777 or *XP4F*

Correction:

Use a register number in the allowable range.

Register numbers may not contain alphabetic characters.

E8: Data can not be saved in EEPROM

The set-up cannot be saved in EEPROM. A hardware error has occurred that prevents the CPU communicating with EEPROM.

E9: Checksum error

The Controller's (receiver's) calculated checksum is not the same as the transmitted checksum.

Example:

255KP=25F3

Correction:

Send the command as 255KP=25DB.

E16: Check other Status Register

An error has been detected in the other status register. Read this register.

E20: Temperature too high

The Controller's internal temperature is too high.

Correction:

Turn off the Controller. Ensure better ventilation/cooling of the Controller environment or reduce the allowable rated current value (CA command).

E21: Current Overload

The Controller has been overloaded/short-circuited.

Correction:

Use a different motor or insert an inductance of approximately 1mH in series with the motor.

E22: Peak Current limit exceeded

The maximum allowable peak current has been exceeded.

Example:

The acceleration is very high, causing the motor to draw a high peak current.

Correction:

Reduce the acceleration until the error disappears.

E23: Average Current limit exceeded

The maximum allowable rated current has been exceeded.

Example:

The velocity is too high.

Correction:

Reduce velocity until the error disappears.

E24: Supply Voltage exceeds 89V

The power supply voltage has exceeded 89V.

Example:

The power supply voltage is too high or the motor has been decelerated too quickly.

Correction:

If the supply voltage is too high, it should be reduced.

During deceleration the motor can send current back to the Controller, causing the supply voltage to increase. The deceleration (AC) should be reduced until the error disappears. If required, a "Power Dump" shunt resistor should be inserted as described in section *Power Dump Output* page 24.

E25: Negative Limit Switch active

The negative end-of-travel limit is active. Motor movement in the negative direction is stopped. Only positive movement is now possible.

E26: Positive Limit Switch active

The positive end-of-travel limit is active. Motor movement in the positive direction is stopped. Only negative movement is now possible.

E27: The motor is not mounted correctly

The motor is not connected correctly.

Example:

The motor is moving in the wrong direction.

Correction:

Switch the positive and negative cables (terminals M+ and M-).

E28: The encoder is not connected or the motor is blocked

The encoder is not connected or the motor is jammed.

Example:

The motor is blocked by a brake when the Controller is switched on. The encoder can therefore not be checked. The encoder may also be connected incorrectly or not connected at all.

Correction:

Ensure that the motor is free to move when the Controller is switched on. Also check the encoder connections. The test can be switched off by the command EMT=0.

E29: Supply Voltage exceeds 95 V

The power supply voltage has exceeded 95V.

Example:

The power supply voltage is too high or the motor is being decelerated too rapidly.

Correction:

If the power supply voltage is too high, it must be reduced.

During deceleration, the motor can send current back to the Controller, causing an increase in the supply voltage. The deceleration must be reduced until the error disappears.

This problem can also be alleviated by using a "Power Dump" shunt resistor. See section *Power Dump Output* page 24.

E30: The motor is not connected

The motor is not connected.

Example:

The motor does not move.

Correction:

Check the motor connections.

E31: Average Current cannot be measured correctly

The rated current value cannot be measured correctly.

Correction:

Turn the Controller off and then on again. If the error persists, a hardware error has occurred.

It is important to note that the motor must not be moving when the Controller is switched on.

3.11 Alphabetical Command Overview

Com- mand	Description	Limits		Mode					Unit	Page	
		Min	Max	1	2	3	4	5			
AC	Acceleration	100	100000		x	x				RPM/s	47
ADDR	Address	0	255	x	x	x	x	x			47
AP	Motor's Actual Position	-1073741824	1073741823	x	x	x	x	x		Pulses	48
CA	Rated current value	1	12	x	x	x	x	x		Amp	48
CB	Control Bits	1	12	x	x	x	x	x		-	49
CHS	Use Checksum	0=No	1=Yes	x	x	x	x	x			53
CP	Peak Current	1	25	x	x	x	x	x		Amp	54
CU	Show Motor Current			x	x	x	x	x		Amp	54
CV	Current Velocity (actual)			x	x	x	x	x		RPM	54
DIF	Digital Input Format	1	3			x				-	55
EMT	Encoder/motor Test flag (after power up)	0=No	1 or 2 =Yes	x	x	x	x				56
ES	Error Status	0	1	x	x	x	x	x			57
EST	Error Status in Text	0	1	x	x	x	x	x			59
ET	Encoder Type	0=PNP	1=NPN	x	x	x	x	x			59
H	Halt Motor				x	x					59
HELP	Show Commands			x	x	x	x	x			60
IL	Integral Summation Limit	0	25000	x	x	x	x	x			60
IN	Read input port status			x	x	x	x	x		bit	61
K	Show all K and IL values			x	x	x	x	x			61
KD	Constant Kd	0	65535	x	x	x	x	x			61
KI	Constant Ki	0	65535	x	x	x	x	x			62
KP	Constant Kp	0	65535	x	x	x	x	x			62
MO	Mode: 1=pulse, 2=position, 3=register 4=velocity, 5=moment	1	5	x	x	x	x	x			62
MP	Multiplication of incoming pulses	1	10	x							63
MR	Recall data from EEPROM	0	3	x	x	x	x	x			63
MS	Save set-up in EEPROM			x	x	x	x	x			63
MU	Multiplication of encoder pulses	1	4	x	x	x	x	x			64
NLS	Negative Limit Switch (active level)	0=Low	1=High	x	x	x	x	x			64
OUT	Read output port status			x	x	x	x	x		bit	65
PE	Maximum Pulse Error	0	1073741823	x	x	x				Pulses	66
PLS	Positive Limit Switch (active level)	0=Low	1=High	x	x	x	x	x			66
PM	PWM mode	0	1	x	x	x	x	x			66
PO	Motor Voltage	0=On	1=Off	x	x	x	x	x			67
PR	No. of encoder pulses per revolution	50	20000	x	x	x	x	x		Pulses/rev.	68
PW	Position Width	0	20000		x	x				Pulses	68
RESET	Reset Controller			x	x	x	x	x			69
RS	Status: 0=stop, 1=acc, 2=max, 3=dec	0	3	x	x	x	x	x			69
SD	Default Set-up			x	x	x	x	x			70
SH	Smooth Halt of motor				x	x					71
SP	Set new Position	-1073741824	1073741823		x	x				Pulses	71
SR	Set Relative position	-1073741824	1073741823		x	x				Pulses	71
SZ	Zero-point Seek				x	x					72
SZT	Zero-point Seek Type	0	1		x	x					72
VE	Show firmware version and date			x	x	x	x	x			72
VM	Maximum Velocity	3	65535		x	x	x	x		RPM	73
VOL	Show supply voltage	12	100	x	x	x	x	x		Volt	73
VV	Show analog input value	-1024	1023	x	x	x	x	x		ADC Step	75
Continued next page ...											

3.11 Alphabetical Command Overview

Com- mand	Description	Limits		Mode					Unit	Page
		Min	Max	1	2	3	4	5		
VVH	Hysteresis for analog input	0	200				x	x	ADC Step	74
VVL	Negative voltage for analog input	-10V	Zero-point				x	x	ADC Step	74
VVO	Zero-point for analog input	-10V	+10V				x	x	ADC Step	74
VVU	Positive voltage for analog input	Zero-point	+10V				x	x	ADC Step	75
X	Show parameter set	none or 0	15			x				75
XA	Acceleration in parameter set	100	100000			x			RPM/sec.	75
XP	Position in parameter set	-1073741824	1073741823			x			Pulses	76
XR	Relative positioning	0=No	1=Yes			x			Pulses	76
XV	Velocity in parameter set	3	65535			x			RPM	77
ZL	Level for zero-point contact	0	1		x	x				77
!	Show controller type and address			x	x	x	x	x		46
?	Show set-up			x	x	x	x	x		46

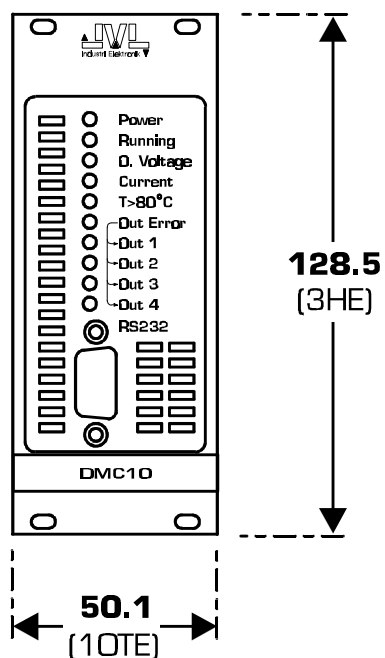
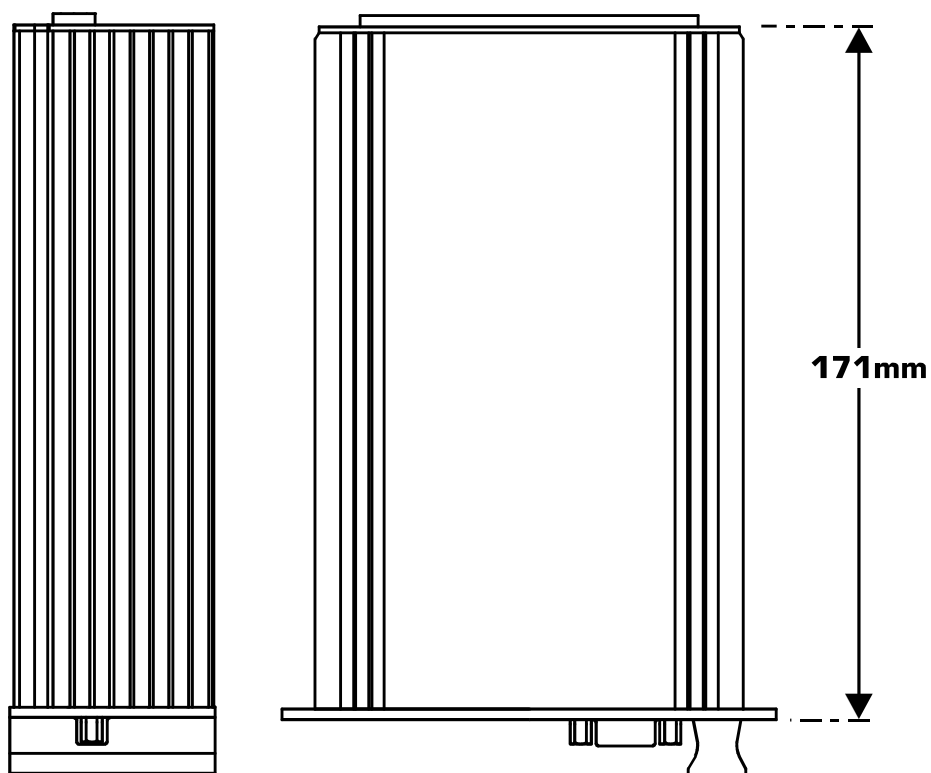
4.1

Specifications

Description	Min.	Typical	Max.	Unit
Supply				
Supply Voltage (P+ and P-)	15		80	V DC
Power Consumption (unloaded)		5		W
Motor Output M+ and M-				
Output Voltage (depends on supply voltage)	0		85	V RMS
Continuous Motor Current (Rated Current)	0		12	A
Peak Current	0		25	A
Power Loss in Driver (at full motor current)			25	W
PWM Frequency		19.75		kHz
Encoder Input :				
Supply to Encoder (pin 14A/5VO)	4.8		5.2	V DC
Allowable load on encoder supply (pin 14A/5VO)			200	mA
Encoder Frequency (50% duty-cycle)	0		500	kHz
Step- and Direction Input (pins 22A to 25A)				
Allowable Input Frequency (50% duty-cycle)			500	kHz
Positive pulse-width	1.0			µs
Negative pulse-width	1.0			µs
Logic "0"			1.8	V DC
Logic "1"	3.8			V DC
User Inputs IN1-IN8 :				
Input Impedance	3.2		3.6	kOhm
Logic "0"	-1		2.5	V DC
Logic "1"	4.5		30	V DC
Logic "0"	-		1.0	mA DC
Logic "1"	2.0		-	mA DC
User Outputs O1 - O4 :				
Supply Voltage	6		28	V DC
Loaded Current per Output			500	mA DC
Analog Input AIN :				
Input voltage (nominal)	-10.0		10.0	V DC
Input impedance		20		kOhm
Power Dump Output PDO :				
Voltage	0		100	V DC
Allowable shunt resistor	15			Ohm
Diverse :				
Operating Temperature Range	0		45	°C
Weight		720		gram

4.2

Physical Dimensions



TT0204GB

4.3

Servo loop

The DMC 10 Controller uses a PID (Proportional-Integral-Differential) servo loop, as illustrated in the figure below

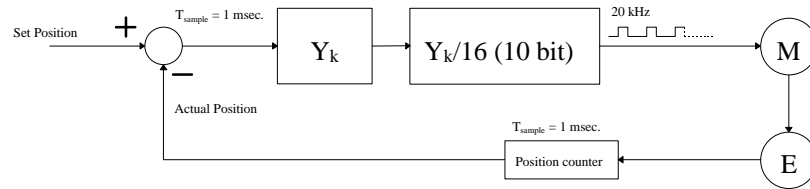


Figure -8 - DC servo

4.3.1 Mathematical Description of the Servo Loop

The servo loop can be mathematically described as follows:

$$Y_k = K_p(e + K_i(e_i + K_d(e_d$$

where:

Error = Set Position - Actual Position = required position minus actual position

e = Error = The instantaneous error between the required and the actual position.

e_i = Error_{Total} = Accumulated error

e_d = Error - Error_{Previous} = Difference between the actual error and the preceding error.

Error_{Total} is limited by the integral summation limit IL.

The servo loop can be adjusted using the parameters IL, K_p , K_i and K_d .

4.4

Error Indication

In addition to their normal function as indicated on the Controller panel, the DMC10's LEDs are also used to indicate vital error conditions. The following describes the normal functions of the LEDs and then their additional functions.

4.4.1 O. Voltage LED

The O. Voltage LED is activated when the supply voltage exceeds 89V. The Power Dump shunt resistor is simultaneously switched in when the voltage is exceeded.

4.4.2 Current LED

The Current LED is lit if the specified rated current (CA) or peak current (CP) is being exceeded. The LED is switched off again if current consumption falls below the specified limit. The LED is also lit if an overload has occurred. The system must be reset after an overload. See *Reset Controller (RESET)* page 69.

4.4.3 T>80°C LED

The T>80°C LED is lit when the Controller's internal temperature exceeds 80°C The Controller must be reset.

4.4.4 Out Error LED

The Out Error LED is lit when an error has occurred at one of the four outputs O1-O4.

4.4.5 Four LEDs Blinking in Sequence

If the four LEDs *Running*, *O. Voltage*, *Current* and *T>80°C* blink in sequence, it is an indication of a PROM error. When the Controller is switched on, the checksum in the Controller's program memory (PROM) is verified. If the pre-programmed checksum does not match the calculated checksum, the Controller will not operate the motor. The PROM may be defective. Try resetting the Controller.

4.4.6 Three LEDs Blinking Simultaneously

If the three LEDs *Running*, *Current* and *T>80°C* blink simultaneously, a motor error or encoder error has occurred. When the Controller is switched on, a check is carried out to ensure that the motor and encoder are connected correctly. The PWM signal to the motor is gradually increased until movement is registered or the PWM signals reach 50%.

In this way the Controller can check whether:

1. The motor is correctly connected, i.e. moves in the right direction.
2. The motor is blocked, i.e. draws a lot of current without the motor moving.
3. The encoder is not connected correctly.

Check that the motor or encoder is connected correctly. Use the EST command (*Error Status Text (EST)* page 59) for further information from the Controller.

During installation and use of the DMC10, various errors can occur. Information about many of these can be obtained from the Controller itself using the EST command. Some error conditions are similar to other errors. The following describes some of the most common errors and possible solutions.

The Encoder is not connected.

Use the ET command and select the correct encoder type.

Four LEDs Blinking in Sequence.

See preceding section.

Incorrect Velocity.

It is important that the servo constants are adjusted. The system cannot maintain the correct velocity if the servo loop is not adjusted.

The Velocity is incorrect, even though the servo constants have been adjusted.

It is important that the specification of the encoder pulses/revolution is set correctly. Use the PR command, see *Encoder Pulses (PR)* page 68.

The motor moves in the wrong direction.

Reverse the connections to the motor.

The motor does not move to the correct position by selecting XP0.

XP0 is used for the zero-point seek function and has therefore a different function than the other position registers.

The motor and encoder are connected correctly but still report an error.

Check that the encoder type is set correctly using the ET command.

The motor does not supply the correct torque

It is important that the servo constants are adjusted. The system cannot produce the correct torque if the servo loop is not adjusted.

Three LEDs Blink.

A problem has occurred with either the encoder or the motor. If the error occurs when the system is switched on, see section Three LEDs Blinking Simultaneously page 88. The error may also occur during operation. In cases where the encoder and motor appear to be connected correctly, check the maximum allowable pulse error using the PE command (Maximum Pulse Error (PE) page 66). Check also the encoder type using the ET command, see *Encoder type (ET)* page 59.

4.6

Connector Board

As an accessory to the Controller, JVL can supply a Connector Board type CON12. This connector board enables connection via snap-lock terminals. The following illustration shows the connection facilities. The Connector Board can either be mounted at the rear of a 19" rack or on the Controller itself.

Connections Connector Board for DMC10 Type: "CON12"

Controller Supply
P+ +15-80V In
P- Ground (power)

Motor Outputs
M+ Motor +
M- Motor -

Power dump
PDO Power dump output

Encoder Input
5V0 +5V Out for encoder
EZ1 Reserved for future use
EZ2 Reserved for future use
EA1 Encoder, input A1
EA2 Encoder, Input A2
EB1 Encoder, Input B1
EB2 Encoder, Input B2
ECM Encoder ground

Step-pulse and Direction Inputs
CCM Ground for pulse Input
CLK Pulse input
DCM Ground for direction input
DIR Direction input

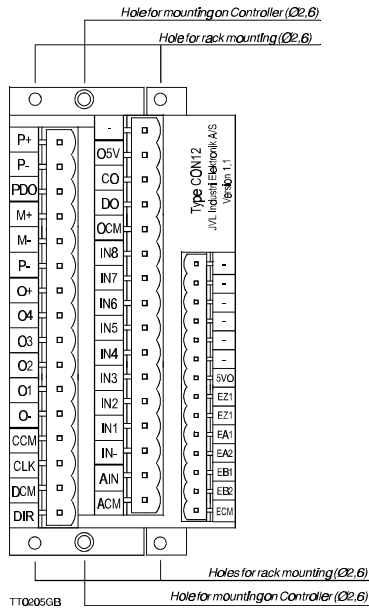
Step-pulse and Direction Outputs
O5V +5V In for output supply
CO Pulse output
DO Direction output
OCM Ground for outputs

Analogue Inputs
AIN +/-10V Analogue input
ACM Ground for analogue Input

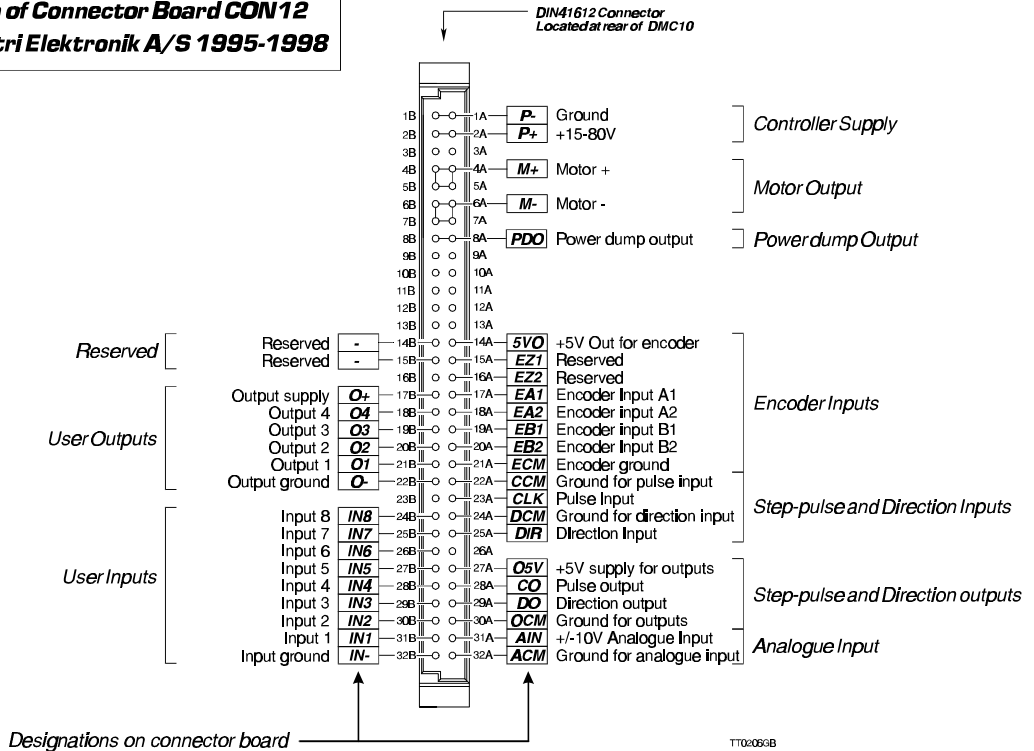
User Outputs
O+ Output supply
O1 Output 1
O2 Output 2
O3 Output 3
O4 Output 4
O- Output ground

User Inputs
IN1 Input 1
IN2 Input 2
IN3 Input 3
IN4 Input 4
IN5 Input 5
IN6 Input 6
IN7 Input 7
IN8 Input 8
IN- Input ground

Reserveret
- Reserved for future use

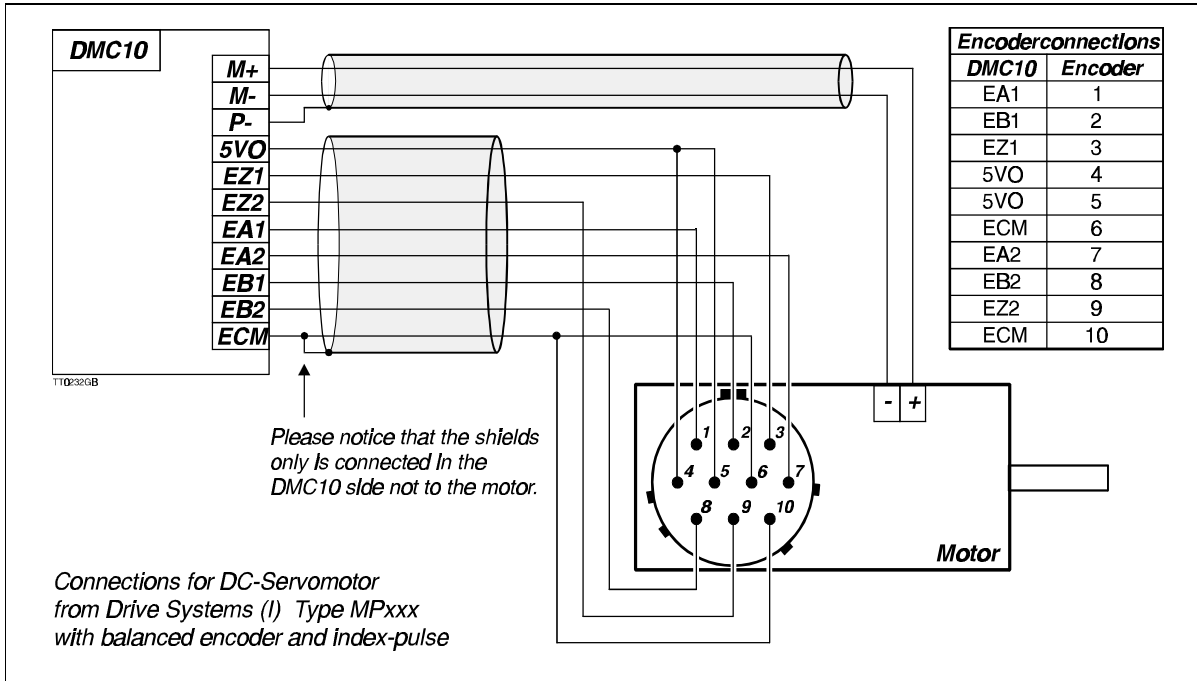


**Diagram of Connector Board CON12
JVL Industri Elektronik A/S 1995-1998**



4.7 Examples of Motor Connection

This section illustrates examples of motor connection for different DC-motors to the DMC10. Please feel free to consult JVL or one of our distributors if you have problems by connecting motors to DMC10.



-
- Symbols**
! 46
? 46
- A**
AC 47
Acceleration 47, 68, 75
Actual Position 48
ADC 23
ADDR 47
AIN 42
Analog Input 45
AP 48
- C**
CA 42, 48
CB Flags 49
CB1 49
Checksum 26, 53
CHS 53
Controller type 46
CP 54
CU 54
- D**
Deceleration 47
DIF 55
- E**
EEPROM 63
EMT 56
Encoder 64
Encoder pulses 68
End-of-travel 64, 66
Error Messages 78
ES 58
EST 59
ET 59
Exclamation mark 47
- F**
Firmware 72
- G**
Ground 21
- H**
H 59
Hardware 14
HELP 60
- I**
IL 60
IN 61
IN1 36
IN2 43
IN3 43
IN4 43
IN6 43, 72
IN7 64, 66
IN8 64, 66
Inputs 61
- K**
K 61
KD 61
KI 62
KP 62
Kp 62
- M**
Maximum Velocity 73
MO 62
MP 63
MR 63
MS 63
MU 64
- N**
Negative Limit Switch 64
NLS 64
- O**
O1 37, 38, 65
O4 65
OUT 65
Outputs 61
- P**
PE 66
Peak Current 54
PID 87
PLS 66, 72
PM 66
PO 67
Position 76
Positioning Mode 9, 35
Positive Limit Switch 66
Power Dump Output 24
PR 68
PROM error 88
PW 68
- R**
Rated Current 48
Register Mode 10, 38
Relative Positioning 71
Reset 69
RS 69
RS232 25
RS232/RS485 Answer 49
- S**
SD 70
Servo Loop 87
SH 71
SP 71
SR 71
Step-pulse 34
Stop motor 59
SZ 72
SZT 72
- T**
Torque Mode 12, 42, 45
- V**
VE 72
Velocity 68, 73, 77
Velocity Mode 11, 45
VM 73
VOL 73
VV 75
VVH 74
VVL 74
VVO 74
VVU 75
- X**
XAn 75
XPn 76
XRn 76
XVn 77
- Z**
Zero point contact 77
Zero-point Seek Function 43
Zero-point Seek Type 72
ZL 77

