\

From Components to Systems

www.isel.com

4-Axis Controller

iMC-M / iMC-MP
iMC-S8

Programming Guide

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ +49 (0)6659 981-0 🖷 +49 (0)6659 981-776

\

From Components to Systems

www.isel.com

4-Axis Controller

iMC-M / iMC-MP
iMC-S8

Programming Guide

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ +49 (0)6659 981-0 🖷 +49 (0)6659 981-776

\

From Components to Systems

www.isel.com

4-Axis Controller

iMC-M / iMC-MP
iMC-S8

Programming Guide

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ +49 (0)6659 981-0 🖷 +49 (0)6659 981-776

\

From Components to Systems

www.isel.com

4-Axis Controller

iMC-M / iMC-MP
iMC-S8

Programming Guide

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ +49 (0)6659 981-0 🖷 +49 (0)6659 981-776

The information, technical data and measurements in this printed text are up to date at the time of publication. There may nevertheless be typographical and other errors. If you have any suggestions for improving the text or have identified any errors, we would be grateful for your feedback.

It should be noted that the software and hardwarenames used in our printed text are the property of their manufacturers and universally protected by brand name, trademark and intellectual property law.

All rights reserved. No part of our printed text may be reproduced in any form (by printing, photocopying or any other process) or processed, copied or distributed using electronic systems without the prior written permission of isel Germany AG.

Manufacturer: **isel Germany AG**
Bürgermeister-Ebert-Straße 40
D-36124 Eichenzell
Germany

Tel.: +49 (0)6659 981-0
Fax: +49 (0)6659 981-776
Email: automation@isel.com
http://www.isel.com

Item no.:

Version: 05/2014

The right to make technical changes is reserved. Current user guides and manuals can be downloaded at: www.isel-data.de/manuals

Contents

1 Overview of commands 5

2 DNC mode and commands 7

2.1 Structure of DNC commands 7

2.2 Commands in DNC mode 8

2.2.1 Initialisation and set number of axes 8
2.2.2 Execute a relative movement 8
2.2.3 Read ports 9
2.2.4 Write ports 10
2.2.5 Set reference speed 12
2.2.6 Diagnosis 12
2.2.7 Set circle interpolation plane 13
2.2.8 Set circle interpolation direction 14
2.2.9 Release 14
2.2.10 Brake switch 15
2.2.11 Set device number 15
2.2.12 Request controller status data 16
2.2.13 Save CNC data field 16
2.2.14 Initialise parameters 17
2.2.15 Set start-stop frequency 18
2.2.16 Set acceleration 18
2.2.17 Delete CNC program 19
2.2.18 Execute an absolute movement 19
2.2.19 Set origin 20
2.2.20 Simulate reference movement 21
2.2.21 Request current position 22
2.2.22 Reference run 23
2.2.23 Restart a movement that has been stopped or start a CNC program 24
2.2.24 Switch test mode on/off 25
2.2.25 Request controller version data 25
2.2.26 Helix interpolation 26
2.2.27 Circle interpolation 27
2.2.28 Switch 3D interpolation on/off 27
2.2.29 Move to a port event 28
2.2.30 Monitor and control codes 29

2.3 Calculating circle parameters 30

2.3.1 Parameters for circle and helix interpolations 30
2.3.2 Calculating the arc 30
2.3.3 Calculating the interpolation parameter 31
2.3.4 Arc start position 31
2.3.5 Directions in start position of circle interpolation 31
2.3.6 Example circle interpolation calculation 31

3 The CNC mode and its commands 34

3.1 Structure of the CNC commands 34

3.2 The CNC mode commands 34

3.2.1 Save CNC data field 34

3.2.2 Relative movement in CNC mode ... 35
3.2.3 Send synchronisation character ... 36
3.2.4 Wait for synchronisation character ... 36
3.2.5 Loops and jump to in CNC mode ... 37
3.2.6 Time delay in CNC mode ... 37
3.2.7 Move to a port event in CNC mode ... 38
3.2.8 Reference movement in CNC mode ... 38
3.2.9 Data fields in CNC mode ... 39
3.2.10 Define reference speed in CNC mode ... 40
3.2.11 Select plane for circle interpolation in CNC mode ... 40
3.2.12 Set circle direction for circle interpolation in CNC mode ... 41
3.2.13 Brake switch in CNC mode ... 42
3.2.14 Set start-stop frequency in CNC mode ... 42
3.2.15 Set acceleration in CNC mode ... 42
3.2.16 Absolute movement in CNC mode ... 43
3.2.17 Set origin in CNC mode ... 44
3.2.18 Simulate reference movement in CNC mode ... 45
3.2.19 Read port and jump to in CNC mode ... 45
3.2.20 Set port in CNC mode ... 47
3.2.21 Turn test mode on/off in CNC mode ... 48
3.2.22 Helix interpolation in CNC mode ... 48
3.2.23 Circle interpolation in CNC mode ... 49
3.2.24 Turn 3D interpolation on/off in CNC mode ... 50

4 Initialisation commands ... 51

4.1 Fundamentals ... 51

4.2 Details of the individual commands ... 51

4.2.1 Set default reference speed ... 51
4.2.2 Number of steps for releasing the reference switch ... 51
4.2.3 Set default acceleration ... 52
4.2.4 Set default start-stop frequency ... 53
4.2.5 Enable IO-only mode ... 53
4.2.6 Enable/disable end switch and set active level ... 54
4.2.7 Invert end switch ... 55
4.2.8 Invert axis direction ... 56
4.2.9 Invert reference directions ... 56
4.2.10 Set axis type ... 57
4.2.11 Abort initialisation via port input in CNC mode ... 58
4.2.12 Read initialisation values from flash ... 58
4.2.13 Write initialisation values to flash ... 59
4.2.14 Load default initialisation values ... 58
4.2.15 Initialization error signal via output port in CNC mode ... 60
4.2.16 Initialization hood monitoring via port input in CNC mode ... 61
4.2.17 Initialization of an external emgerncy OFF via port input in CNC mode ... 61
4.2.18 Set an UserInfo ... 62

5 iMC-xx error messages ... 63

1 Overview of commands

The following *isel*-@ format commands have been implemented for iMC-xx controllers:

DNC commands

Command	*Meaning*	*Chapter*
@01	x-axis initialisation	2.2.1
@03	x-y-axis initialisation	2.2.1
@07	x-y-z-axis initialisation	2.2.1
@08	a-axis initialisation	2.2.1
@0a / @0A	Relative movement	2.2.2
@0b	Read port	2.2.3
@0B	Write port	2.2.4
@0d	Set reference speed	2.2.5
@0D	Diagnosis (only for activation/debugging)	2.2.6
@0e	Set circle interpolation plane	2.2.7
@0f	Set circle interpolation direction	2.2.8
@0F	Release	2.2.9
@0g	Brake switch	2.2.10
@0G	Set device number	2.2.11
@0h	Read inputs (only for activation/debugging)	2.2.12
@0H	Read status (only for activation/debugging)	2.2.12
@0i	Save CNC program	2.2.13
@0I	Initialising parameters	2.2.14
@0j	Set start-stop frequency	2.2.15
@0J	Set acceleration	2.2.16
@0k	Delete CNC program	2.2.17
@0m / @0M	Absolute movement	2.2.18
@0n	Set origin	2.2.19
@0N	Simulate reference movement	2.2.20
@0P	Request position	2.2.21
@0r / @0R	Reference run	2.2.22
@0s / @0S	Start CNC program or movement	2.2.23
@0T	Switch test mode on/off	2.2.24
@0V / @0?	Request version	2.2.25
@0w	Helix interpolation	2.2.26
@0y	Circle interpolation	2.2.27
@0z	3D interpolation on/off	2.2.28
@0Z	Movement to input port	2.2.29

CNC commands

Command code	*Meaning*	*Chapter*
0	Relative movement	3.2.2
1	Send synchronisation character	3.2.3
2	Wait for synchronisation character	3.2.4
3	Loop, jump to	3.2.5
5	Time delay	3.2.6

6	Condition for movement to port event	3.2.7
7	Reference movement	3.2.8
9	End data field	3.2.9
D	Set reference speed	3.2.10
e	Set circle interpolation plane	3.2.11
F	Set circle interpolation direction	3.2.12
G	Brake switch	3.2.13
J	Set start-stop frequency	3.2.14
J	Set acceleration	3.2.15
M	Absolute movement	3.2.16
N	Set origin	3.2.17
N	Simulate reference movement	3.2.18
O	Read port and jump to	3.2.19
P	Set output port	3.2.20
T	Turn test mode on/off	3.2.21
W	Helix interpolation	3.2.22
Y	Circle interpolation	3.2.23
Z	Turn 3D interpolation on/off	3.2.24

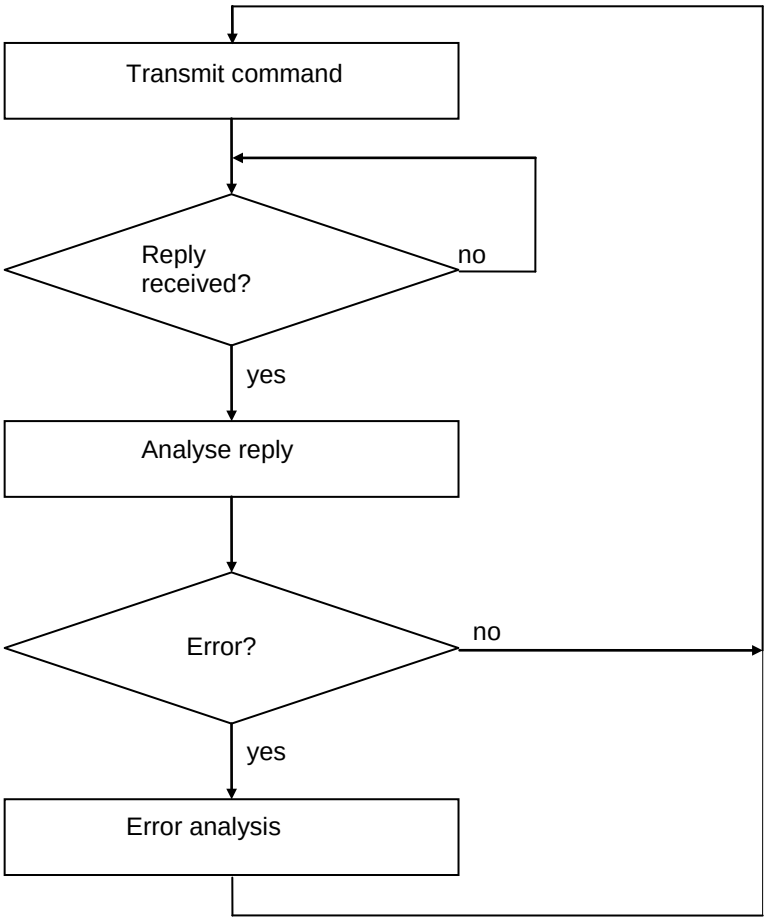
Initialisation commands

Command	*Meaning*	*Chapter*
@0Id	Set default reference speed	4.2.1
@0IF	Define the steps for releasing the reference switch	4.2.2
@0IJ	Set default acceleration	4.2.3
@0Ij	Set default start-stop frequency	4.2.4
@0IO	Enable IO-only mode	4.2.5
@0IE	Enable/disable end switch and set active level	4.2.6
@0Ie	Invert end switch	4.2.7
@0ID	Invert axis direction	4.2.8
@0IR	Invert reference directions	4.2.9
@0IT	Set axis type	4.2.10
@0IA	Abort command in CNC mode	4.2.11
@0IL	Read initialisation values from the flash memory	4.2.12
@0IW	Write initialisation values to the flash memory	4.2.12
@0IX	Load default initialisation values	4.2.14

2 DNC mode and commands

2.1 Structure of DNC commands

Operating in DNC mode, data sets or commands transmitted from a control computer are analysed and executed directly. For this purpose, an initialisation is required at the start of the data communication process. The initialisation comprises the data opening character @, the device number (0=standard) and the number of axes to be processed. Then the program steps are transmitted individually to the controller, which executes them directly. In order to verify the transmission of data and report any errors, the appropriate ASCII characters are sent back to the control computer via the interface. This software handshake procedure is implemented as follows:



Initially, a command is sent to the controller. The command is decoded and executed by the controller, which then generates an appropriate confirmation or error indicator. The control computer analyses this reply. If an error has occurred, the appropriate error analysis and correction must be executed. The next command can then be transmitted to the controller in the same way.

The range of commands in the DNC mode of iMC-xx controllers is described below.

2.2 Commands in DNC mode

2.2.1 Initialisation and set number of axes

Command: Set number of axes

Purpose: Transmitting the number of axes reinitialises the controller.

Structure: @<GN><Axes><CR>

@	= Data opening character
<GN>	= Device number, standard=0
<Axes>	= Axes entered (see below)
<CR>	= Carriage return to indicate end of command

Application: @07, @08

Description: The controller is addressed via "@0", and the following numerical value contains the axis configuration. Each axis is internally represented by one bit of a binary value, giving the following values:

1 --> x-axis
3 --> x+y-axis
7 --> x+y+z-axis
8 --> a-axis

Restrictions:

The combinations, "@00", "@02", "@04", "@06" and "@09" are not permitted.
NOTE: The a-axis must always be initialised separately.

2.2.2 Execute a relative movement

Command: Relative movement

Purpose: The controller will generate a relative movement corresponding to the number of steps and step speed transmitted. The movement is executed immediately.

Structure: @<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
@<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

@	= Data opening character
<GN>	= Device number, standard=0
A or a	= Relative movement command
<Sx>	= Number of steps x
<Gx>	= Speed
<Sy>	= Number of steps y
<Gy>	= Speed
<Sz>,<Sz1>	= Number of steps z
<Sz2>	= Number of steps z, second movt. in 2.5D, three axes
<Gz>,<Gz1>	= Speed

<Gz2> = Speed z, second movement in 2.5D, three axes
<Sa> = Number of steps a, with four axes
<Ga> = Speed, with four axes
<CR> = Carriage return to indicate end of command

Application:
@0A 5000,900 (x-axis only)
@0A 50,900,20,9000 (x and y-axis)
@0A 30,800,10,900,4,90,-4,30 (x, y and z-axis, with three axes)
@0A 30,800,10,900,4,90,-4,30 (x, y, z and a-axis, with four axes)

Description: The controller is addressed via "@0"; "A" or "a" indicates that a relative movement is to be executed. The controller then expects a coordinate pair for each axis, consisting of the number of steps and the speed. Distances are defined using relative dimensions, i.e. based on the most recent position. The number of entries must agree with the number of axes, i.e. one pair of parameters for an x operation, two parameter pairs for an xy operation, three parameter pairs for an xyz operation, and four parameter pairs for an xyza operation. The individual values are separated using commas. When working with three axes and 2.5D, two pairs of values are expected for the z-axis, as the situation, “Move, lower tool, then raise tool”, is very common in processing applications. In 2.5D interpolation operation, the movement of the x and y-axes is carried out first (linear interpolation) and the z-axis is then moved using the values stated in z1, then by those stated in z2. Even if only one axis needs to be moved, values must still be submitted for all initialised axes. A value of 0 should be entered for the number of steps relating to the stationary the axes. After the movement has been executed, the controller responds with the handshake character ("0"). The controller can only process new commands after the existing command has been executed.

Restrictions:
The command can only be used after the number of axes has been set. Moreover, the controller does not verify that the movement remains within the permissible range for the connected mechanism.
When using 2.5D interpolation, the speed parameter for the axis with the longest path is defined as the track speed and the speed of the other axes is adapted depending on the path ratio. In contrast, in 3D interpolation operation the speed value for the x-axis is used as the standard value for specifying the track speed.

2.2.3 Read ports

Command: Read port

Purpose: This command makes it possible to ascertain the current status of logical or physical input ports via the serial interface.

Structure: @<GN>b<Portnr><CR>

@ = Data opening character

<GN> = Device number, standard=0
b = “Read port” command
<Portnr> = Port number (see below)
<CR> = Carriage return to indicate end of command

Application: @0b0, @0b1

Description: The controller is addressed via "@0". "b" indicates that the status of an input port is to be ascertained. The port number is then entered, and the command is completed by carriage return. The controller responds with the software handshake, “0”, followed by two characters giving a hexadecimal value which corresponds to the current status of the input port. For iMC-xx controllers, the ports and their functions are defined as follows:

Port	Status	Function
0	00 - FF	User I/O
1	00 - FF	Status inputs
		Bit0 Emergency stop circuit 1 input
		Bit1 Emergency stop circuit 2 input
		Bit2 Overheat end stages input
		Bit3 Start button input
		Bit4 Stop button input
		Bit5 Length measurement button input
		Bit6 not in use
		Bit7 not in use
2	00 - FF	Status inputs
		Bit0 spindle input
		Bit1 PowerOk input
		Bit2 not in use
		Bit3 not in use
		Bit4 not in use
		Bit5 “Close cover” and automatic input
		Bit6 “Close cover” or setup input
		Bit7 Key switch input
3	00 - FF	End switch inputs
		Bit0 End switch X1
		Bit1 End switch X2
		Bit2 End switch Y1
		Bit3 End switch Y2
		Bit4 End switch Z1
		Bit5 End switch Z2
		Bit6 End switch A1
		Bit7 End switch A2

Restrictions:

Port status information will be given only if the controller responds with the software handshake “0”. This command cannot be used if a movement is being executed.

2.2.4 Write ports

Command: Write port

Purpose: This command makes it possible to describe logical or physical output ports with defined values via the serial interface.

Structure: @<GN>B<Portnr>,<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
B	= “Write port” command
<Portnr>	= Port number (see below)
<Value>	= New port value
<CR>	= Carriage return to indicate end of command

Application: @0B1,1

Description: The controller is addressed via "@0". "B" indicates that the value of an output port is to be set. Then the port number and the new port value are entered, separated by a comma, and the command is completed by carriage return. The controller responds either with the software handshake “0”, if the command was successfully executed, or with an error message, if erroneous port numbers and/or values have been entered. For iMC-xx controllers, the ports and their functions are defined as follows:

Port	Value	Function
0	0 - 255	User I/O
1	0	Cover must not be opened
	1	Cover can be opened
2	0	Switch off spindle
	1	Switch on spindle
3	0	Switch off motor currents
	1	Switch on motor currents
4	0 - 255	Analogue output 0 – 10V
5	0	Switch off current reduction
	1	Switch on current reduction
6	0	Switch off brake
	1	Switch on brake
100	0 - 255	Control out
		Bit0 Current reduction output
		Bit1 not in use
		Bit2 “Reset end stages” output
		Bit3 “Enable end stages” output
		Bit4 Spindle output
		Bit5 “Enable cover” output
		Bit6 Brake output
		Bit7 SoftwareOk output
101	0 - 255	Signal out
		Bit0 not in use
		Bit1 not in use
		Bit2 not in use
		Bit3 not in use
		Bit4 not in use

Bit5 “Stop lamp” output
Bit6 “Start lamp” output
Bit7 Lamp error output

Restrictions:

Port values are overwritten only if the controller responds with the software handshake “0”. This command cannot be used if a movement is being executed.

2.2.5 Set reference speed

Command: Set reference speed

Purpose: This command defines, separately for each axis, the speed at which a reference movement is to be executed.

Structure:

@<GN>d<Gx><CR>	(x)
@<GN>d<Gx>,<Gy><CR>	(x-y)
@<GN>d<Gx>,<Gy>,<Gz><CR>	(x-y-z)
@<GN>d<Gx>,<Gy>,<Gz>,<Ga><CR>	(x-y-z-a)

@	= Data opening character
<GN>	= Device number, standard=0
d	= “Set reference speed” command
<Gx>	= Reference speed x
<Gy>	= Reference speed y
<Gz>	= Reference speed z
<Ga>	= Reference speed a
<CR>	= Carriage return to indicate end of command

Application:
@0d2500
@0d2400,3000
@0d1000,3000,2000
@0d1000,3000,2000,2000

Description: If the controller is not provided with any information on the reference speed, it will execute the movement using a default value. Changed values are not saved when the controller is switched off.

Restrictions:

The defined speed must lie within the valid range of values for speeds
If the reference speed selected is too high and the spindle pitch is too high, the inertia may damage the reference switches. The controller requires the connected origin switch to exhibit hysteresis. Please bear this in mind when connection electronic null sensors!

2.2.6 Diagnosis

Command: Diagnosis

Purpose: Requesting controller diagnosis data.

Structure: @<GN>D<Code1><Code2><CR>

@	= Data opening character
<GN>	= Device number, standard=0
D	= Diagnosis command
<Code1>	= Command code 2
<Code2>	= Command code 3
<CR>	= Carriage return to indicate end of command

Application: @0DRp, @0DRn, @0DS0

Description: The data opening character "@0" prepares the controller for a new command. “D” communicates to the controller that a diagnosis is to be carried out. Then there are two codes for the parameter, and carriage return completes the command. The controller responds with the software handshake “0”, followed by two characters giving a hexadecimal value corresponding to the current value of the parameter. The following parameters can be requested:

Code1	Code2	Function
R	p	Request positive end switches
R	n	Request negative end switches

End switch bit configuration:

Bit0:	x-axis
Bit1:	y-axis
Bit2:	z-axis
Bit3:	a-axis

Restrictions:

This command has been listed here for the sake of completeness and is used by the operating programs of isel Germany AG for the purpose of diagnosing specific controller statuses.

2.2.7 Set circle interpolation plane

Command: Set plane

Purpose: Setting the circle interpolation plane. Circles are only defined within a plane. The default plane for circle interpolation is the xy plane. The “select plane” command, however, makes it possible to define any of the other plane configurations as the circle plane.

Structure: @<GN>e<Plane><CR>

@	= Data opening character
<GN>	= Device number, standard=0
e	= Set circle plane command
<Plane>	= Plane entered (see below)
<CR>	= Carriage return to indicate end of command

Application: @0e1, @0e0

Description: The controller is addressed via "@0". "e" indicates that the circle interpolation plane is to be set. The ensuing numerical value defines the plane as follows:

0 --> XY plane
1 --> XZ plane
2 --> YZ plane

Restrictions:

This command is modal, i.e. the plane selected for circle interpolation is retained until it is overwritten by a new plane selection.

2.2.8 Set circle interpolation direction

Command: Set circle direction

Purpose: Setting the circle interpolation direction. Circle interpolation is executed by two successive commands. The first command establishes the circle direction, and the second gives the interpolation parameters (see 2.2.27).

Structure: @<GN>f<Direction><CR>

@	= Data opening character
<GN>	= Device number, standard=0
f	= “Set circle direction” command
<Direction>	= 0 → CW (clockwise)
	-1 → ACW (anticlockwise)
<CR>	= Carriage return to indicate end of command

Application: @0f-1, @0f0

Description: The controller is addressed via "@0". "f" indicates that the circle interpolation direction is to be set. The ensuing numerical value defines the direction as follows:

0 → CW (clockwise circle interpolation)
-1 → ACW (anticlockwise circle interpolation)

Restrictions:

The circle interpolation direction must always be programmed before any circular movement.

2.2.9 Release

Command: Release a connected axis

Purpose: The controller will release the axis from the reference switch.

Structure: @<GN>F<Axis><CR>

@	= Data opening character
<GN>	= Device number, standard=0
F	= Release command
<Axis>	= Axis entered
<CR>	= Carriage return to indicate end of command

Application: @0F1

Description: The controller is addressed via "@0". "F" indicates that a connected axis is to be “released” from the end switch. The following numerical value specifies which axis is to be released.
After the movement has taken place, the controller will send its confirmation indicator and wait for the next commands. The controller can only execute further commands after a reference movement has been carried out.

2.2.10 Brake switch

Command: Switch brake

Purpose: Enabling the operation of a brake, if there is one.

Structure: @<GN>g<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
g	= Switch brake
<Value>	= 0 → brake not supplied with power 1 → brake supplied with power
<CR>	= Carriage return to indicate end of command

Application: @0g1

Description: The “@0g1” command can be used to power, i.e. open, the brake. “@0g0” closes the brake.

2.2.11 Set device number

Command: Set device number

Purpose: Specifying the number of the device by which the controller can be addressed.

Structure: @<GN>G<N><CR>

@	= Data opening character

<GN> = Device number, standard=0
G = Set device number
<N> = Device number
<CR> = Carriage return to indicate end of command

Application: @0G2

Description: The controller is assigned device number 2.

Restrictions:

2.2.12 Request controller status data

Command: Status data request

Purpose: Requesting important controller status data in order to see the current status and to find and diagnose errors.

Structure: @<GN>H<CR>
@<GN>h<CR>

@ = Data opening character
<GN> = Device number, standard=0
H = “Request status data from inputs” command
h = “Request status data from end switch inputs” command
<CR> = Carriage return to indicate end of command

Application: @0H, @0h

Description: The data opening character "@0" prepares the controller for a new command. "H" or "h" makes the controller reply with information in plaintext format about the current status. At the end of this information, the controller responds with the handshake character ("0"). The information is given in ASCII format, formatted in lines so it can be represented directly on a control computer’s monitor screen, e.g. in a terminal window. This information covers the status of the end switches or of the operational components of a system.

Restrictions:
For this function to be used, the control computer must have a sufficiently large receiving buffer so that no information is lost.

2.2.13 Save CNC data field

Command: Save CNC data field

Purpose: Initialising the transmission of storable commands. This command is compulsory when starting CNC mode.

Structure: @<GN>i<CR>

@	= Data opening character
<GN>	= Device number, standard=0
i	= “Save CNC data field” command
<CR>	= Carriage return to indicate end of command

Application: @0i

Description: The controller is addressed via "@0". "i" indicates that a CNC data field is to be saved. The command is completed by carriage return. Until it receives the “end data field” command, or until an error occurs, the controller will now only accept CNC commands. The command is acknowledged with an appropriate response. All subsequent storable commands are then saved in the flash memory.

Restrictions:

This command can only be used once the controller has already been initialised, and as long as no movement is being executed. If a program has already been saved, this will result in an error message.

2.2.14 Initialise parameters

Command: Initialise parameters

Purpose: Initialising axis and controller parameters.

Structure: @<GN>I<Code><Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= Initialise command
<Code>	= ASCII character parameter code (see below)
<Value>	= New parameter value
<CR>	= Carriage return to indicate end of command

Application: @0ID3, @0IR1, @0Id1000,900,500,900, @0IW

Description: The data opening character “@0” prepares the controller for a new command. “I” notifies the controller that an initialisation is to be executed. This is followed by a code for the parameter and then by the parameter value(s), and the command is completed by carriage return. The following parameters can be initialised:

Code	Function
d	Default reference speeds for x, y, z, a axis
F	Steps to release the switches for x, y, z, a axis
J	Default acceleration
j	Default start-stop frequency

O	Enable IO only
E	Enable end switch + active level
e	Change end switch
D	Axis directions
R	Reference directions
T	Axis types
L	Read initialisation values from the flash memory
W	Write initialisation values into the flash memory
X	Restore default controller values

Restrictions:

This command has been listed here for the sake of completeness and is used by the operating programs of isel Germany AG to initialise and operate controllers.

2.2.15 Set start-stop frequency

Command: Set start-stop frequency

Purpose: Specifying the frequency with which the ramps necessary for movements begin and end.

Structure: @<GN>j<F><CR>

@	= Data opening character
<GN>	= Device number, standard=0
j	= Set start-stop frequency
<F>	= Frequency
<CR>	= Carriage return to indicate end of command

Application: @0j300

Description: If the controller is not provided with any information on start-stop frequency, it will execute the command using a default value. Changed values are not saved when the controller is switched off.
The start-stop frequency is stated in Hz, the default value being 300 Hz. The expected range of frequencies lies between 20 Hz and 4000 Hz.

Restrictions: The defined start-stop frequency must lie within the valid range of values for step frequencies.
A start-stop frequency that is too high can lead to step losses in the acceleration or braking phase.

2.2.16 Set acceleration

Command: Set acceleration

Purpose: Specifying the acceleration with which the ramps necessary for movements are produced.

Structure: @<GN>J<B><CR>

@	= Data opening character
<GN>	= Device number, standard=0
J	= Set acceleration
<B>	= Acceleration
<CR>	= Carriage return to indicate end of command

Application: @0J100

Description: If the controller is not provided with any information on acceleration, it will execute the command using a default value. Changed values are not saved when the controller is switched off.
The acceleration is stated in Hz/ms, the default value being 100 Hz/ms. The expected range of acceleration lies between 1Hz/ms and 4000 Hz/ms.

Restrictions: The defined acceleration must lie within the valid range of values for acceleration.
Acceleration that is too high can lead to step losses in the acceleration or braking phase.

2.2.17 Delete CNC program

Command: Delete CNC program in flash memory

Purpose: Deleting a CNC program saved in the flash memory.

Structure: @<GN>k<CR>

@	= Data opening character
<GN>	= Device number, standard=0
k	= Delete flash memory command
<CR>	= Carriage return to indicate end of command

Application: @0k

Description: The controller is addressed by “@0”. “k” indicates that the controller’s flash memory is to be wiped. The command is completed by carriage return. If the command has been properly understood by the controller, the flash memory is wiped. This takes about two seconds. The controller will then confirm that the memory has been wiped by sending the reply “0”.

2.2.18 Execute an absolute movement

Command: Movement to an absolute position

Purpose: The controller will move at the specified speeds to the specified position. The movement is executed immediately.

Structure: @<GN>M<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
@<GN>M<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

@	= Data opening character
<GN>	= Device number, standard=0
M	= Absolute movement command
<Sx>	= Position x
<Gx>	= Speed
<Sy>	= Position y
<Gy>	= Speed
<Sz>,<Sz1>	= Position z
<Sz2>	= Position z, second movement always = 0
<Gz>,<Gz1>	= Speed
<Gz2>	= Speed
<Sa>	= Position a, with four axes
<Ga>	= Speed, with four axes
<CR>	= Carriage return to indicate end of command

Application:	@0M 5000,900	(only x-axis)
	@0M 50,900,20,9000	(x and y-axis)
	@0M 30,800,10,900,4,90,0,30	(x, y and z-axis, with three axes)
	@0M 30,800,10,900,4,90,4,30	(x, y, z and a-axis, with four axes)

Description: The controller is addressed by "@0". "M" indicates that an absolute position is about to be stated. In order to provide compatibility with the relative position command, two pairs of figures are expected for the z-axis in the event of three axes being involved. The second set of positional information for the z position must, however, be set to zero and is ignored. Once the command has been executed, the controller responds with the handshake character. The controller cannot process further commands until this command has been executed.

Restrictions:
This command can only be used once the number of axes has been set. The controller does not verify that the movement remains within the permissible range for the connected mechanism.

2.2.19 Set origin

Command: Set current position as origin

Purpose: The controller will save the current position as the virtual origin for the axis or axes entered. The next “absolute movement” instructions will take this virtual origin as the new reference point.

Structure: @<GN>n<Axes><CR>

@ = Data opening character
<GN> = Device number, standard=0
n = “Set origin” command
<Axes> = Axes entered (see below)
<CR> = Carriage return to indicate end of command

Application: @0n7, @0n1, @0n8

Description: The controller is addressed by "@0". "n" indicates that the origin is to be set. After this, the controller is informed of the axes for which the origin is to be set. Each axis is represented internally by one bit of a binary value, giving the following values:

1 --> x-axis
2 --> y-axis
3 --> x+y-axis
4 --> z-axis
5 --> x+z axis
6 --> y+z axis
7 --> x+y+z axis
8 --> a-axis

After the command has been executed, the controller will send a reply.

Restrictions: The virtual origin is only significant for the command, “absolute movement”. Relative positioning is not influenced by the virtual origin, as a relative movement vector is entered in this case.

NOTE: The origin for the a-axis must always be set separately.

2.2.20 Simulate reference movement

Command: Set current position as reference point

Purpose: The controller simulates a reference movement – no axis movement is executed.

Structure: @<GN>N<Axes><CR>

@ = Data opening character
<GN> = Device number, standard=0
N = “Simulate reference movement” command
<Axes> = Axes entered (see below)
<CR> = Carriage return to indicate end of command

Application: @0N1

Description: The controller is addressed by "@0". "N" indicates that a reference movement is to be simulated. After this, the controller is informed of the axes for which a reference movement is to be simulated. Each axis is represented internally by one bit of a binary value, giving the following values:

1 --> x-axis
2 --> y-axis
3 --> x+y-axis
4 --> z-axis
5 --> x+z-axis
6 --> y+z-axis
7 --> x+y+z axis
8 --> a-axis

After the command has been executed, the controller will send a reply.

Restrictions:

The virtual reference point is only significant for the command, “absolute movement”. Relative positioning is not influenced by the virtual reference point, as a relative movement vector is entered in this case.

NOTE: The reference point for the a-axis must always be set separately.

2.2.21 Request current position

Command: Request position

Purpose: The controller replies to the control computer with the current position of all axes.

Structure: @<GN>P<CR>

@	= Data opening character
<GN>	= Device number, standard=0
P	= “Request position” command
<CR>	= Carriage return to indicate end of command

Application: @0P

Description: The controller is addressed by "@0". "P" indicates that a position request is being made. The controller responds with the handshake character and gives the positional values of all axes in hexadecimal format (18 hexadecimal characters for up to three axes, 24 hexadecimal characters for four axes).

The position given in the controller’s reply is structured as follows:

e.g.: 00010002000FFFFFFE for three axes

position x =	000100	hexadecimal in second complement
corresponds to	256	decimal
position y =	02000F	hexadecimal in second complement
corresponds to	131087	decimal
position z =	FFFFFE	hexadecimal in second complement

corresponds to	-2	decimal

e. g.: 000B00044000000FFE003040 for four axes

position x =	000B00	hexadecimal in second complement
corresponds to	2816	decimal
position y =	044000	hexadecimal in second complement
corresponds to	278528	decimal
position z =	000FFE	hexadecimal in second complement
corresponds to	4094	decimal
position a =	003040	hexadecimal in second complement
corresponds to	12352	decimal

Restrictions:

This command can only be used if no movement is being executed (if the system is in stop mode). The controller cannot verify whether the current position corresponds to the mechanism's real position, as there is no control circuit.

NOTE: With up to three axes, this function always gives positions for three axes, irrespective of the number of specified axes. With four axes, positions for four axes are always given.

2.2.22 Reference run

Command: Reference run

Purpose: The controller will move all axes entered to their origins (reference points). The reference points of the axes are always arranged in a reasonable default order in isel systems, but they can be adjusted using the appropriate initialisation commands.

Structure: @<GN>R<axes><CR>

@	= Data opening character
<GN>	= Device number, standard=0
R or r	= Reference run command
<Axes>	= Axes entered (see below)
<CR>	= Carriage return to indicate end of command

Application: @0R7, @0r7, @0R8

Description: The controller is addressed by "@0". "R" indicates that a reference movement is to be executed. The following numerical value specifies the axes which should execute a reference movement. Each axis is represented internally by one bit of a binary value, giving the following values:

1 --> x-axis
2 --> y-axis
3 --> x+y-axis
4 --> z-axis
5 --> x+z-axis
6 --> y+z-axis
7 --> x+y+z-axis
8 --> a-axis

The command is executed in the following order:
z-axis --> y-axis --> x-axis --> a-axis

After the reference movement has been executed, the controller will send its confirmation character and wait for the next commands. The controller can only process new commands once the mechanism has carried out the reference movement.

Restrictions:

This command can only be used once the controller has been initialised by the command, “set number of axes”, and it is limited by the axis configuration given there. If axes are entered erroneously, the error message “3” appears. If the controller is in 3D mode, the command will switch it back into 2.5D mode.

NOTE: The a-axis must always be referenced separately.

NOTE: If the reference switch is not connected, the corresponding axis will be permanently activated. It is, however, possible to simulate the reference switch by operating the stop switch.

2.2.23 Restart a movement that has been stopped or start a CNC program

Command: Start

Purpose: Continuing a movement that has been stopped or starting a CNC program.

Structure: @<GN>S<CR>

@	= Data opening character
<GN>	= Device number, standard=0
S or s	= Start command
<CR>	= Carriage return to indicate end of command

Description: The controller is addressed by "@0". "S" indicates that a movement that has been stopped is to be restarted and that the remainder of the movement is to be completed. If no movement has been stopped, a saved CNC program is started. After the command has been executed, the controller responds with the handshake character (“0”) or sends an error message if there is no saved movement to be completed or if no CNC program has been saved.

2.2.24 Switch test mode on/off

Command: Switch test mode on/off

Purpose: Switching test mode on and off.

Structure: @<GN>T<Status><CR>

@	= Data opening character
<GN>	= Device number, standard=0
T	= “Switch test mode on/off” command
<Status>	= 0 --> switch off, 1 --> switch on
<CR>	= Carriage return to indicate end of command

Application: @0T1, @0T0

Description: The data opening character “@0” prepares the controller for a new command. “T1” turns to test mode on and “T0” turns it off. Following successful completion, the controller issues a handshake character ("0"). The controller handles the end switch and reference movement differently in test mode than in normal operational mode. If a reference movement command is received in test mode, the controller does not actually execute a reference movement as such but just sets the current position as the reference point. The end switch can still be monitored but will be overridden. This is very useful if an axis is at the end switch when the system is turned on and needs to be released.

Restrictions:

This command cannot be executed when a movement is being executed.

2.2.25 Request controller version data

Command: Request version data

Purpose: Requesting important controller version data.

Structure: @<GN>V<CR>

@	= Data opening character
<GN>	= Device number, standard=0
V	= “Request version data” command
<CR>	= Carriage return to indicate end of command

Application: @0V

Description: The data opening character "@0" prepares the controller for a new command. “V” makes the controller respond with information about the controller version in plaintext format. At the end of this information, the controller responds with the handshake character ("0"). The information is given in ASCII format, formatted in lines so it can be represented

directly on a control computer's monitor screen, e.g. in a terminal window.

Restrictions:

For this function to be used, the control computer must have a sufficiently large receiving buffer so that no information is lost.

2.2.26 Helix interpolation

Command: Helix interpolation

Purpose: Producing a helical movement.

Structure: @<GN>w<B>,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry>,<S3><CR>

@	= Data opening character
<GN>	= Device number, standard=0
<B>	= Arc length in steps
<V>	= Speed
<D>	= Interpolation parameter
<Xs>	= Start position axis 1
<Ys>	= Start position axis 2
<Rx>	= Direction axis 1
<Ry>	= Direction axis 2
<S3>	= Number of steps axis 3
<CR>	= Carriage return to indicate end of command

Application: @0w32000,600,-1000,-2000,0,1,-1,6000

Description: The controller is addressed by "@0". "w" indicates that a helical movement is to be executed. The arc length defines the length of the arc section in steps between the start and end positions of the circle segment. All integer values within the valid range of values are permitted for the speed. The interpolation parameter serves as a starting value for the interpolation algorithm used. The parameters Xs and Ys define the starting point of the arc section relative to the helix centre point. Rx and Ry indicate the quadrant in which the interpolation will start. S3 specifies the number of steps for the third axis. After the command has been executed, the controller responds with the handshake character ("0"). The controller cannot process new commands until this command has been executed.

NOTE: To calculate the parameters for the arc, please read the chapter, "Calculating circle parameters".

Restrictions:

This command can only be used once the number of axes has been set. Moreover, the controller does not verify that the movement remains within the permissible range for the connected mechanism.

2.2.27 Circle interpolation

Command: Circle interpolation

Purpose: Processing circles and arcs at constant speed. Circle interpolation is executed by two successive commands. The first sets the circle direction (see 2.2.8); the second gives the interpolation parameters.

Structure: @<GN>y<B>,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry><CR>

@	= Data opening character
<GN>	= Device number, standard=0
<B>	= Arc length in steps
<V>	= Speed
<D>	= Interpolation parameter
<Xs>	= Start position x
<Ys>	= Start position y
<Rx>	= Direction x
<Ry>	= Direction y
<CR>	= Carriage return to indicate end of command

Application: @0y400,1500,119,-141,141,-1,-1

Description: The controller is addressed by "@0". "y" indicates that a circle interpolation is to be executed. The arc length is the length of the arc in steps between the start and end positions of the circle interpolation. All integer values within the valid range of values are permitted for the speed The interpolation parameter serves as a starting value for the interpolation algorithm used. The parameters Xs and Ys define the starting point of the arc relative to the centre of the circle. Rx and Ry indicate the quadrant of the circle in which the interpolation will start. After the command has been executed, the controller responds with the handshake character ("0"). The controller cannot process new commands until this command has been executed.

NOTE: To calculate the parameters, please read the chapter, “Calculating circle parameters”.

Restrictions:

This command can only be used once the number of axes has been set. Moreover, the controller does not verify that the movement remains within the permissible range for the connected mechanism.

2.2.28 Switch 3D interpolation on/off

Command: Switch 3D linear interpolation on/off

Purpose: The controller will extend the interpolation from 2.5D to 3D. With this command, 3D interpolation can be switched on and off.

Structure: @<GN>z<Status><CR>

@	= Data opening character
<GN>	= Device number, standard=0
z	= “3D interpolation” command
<Status>	= 0 --> switch off, 1 --> switch on
<CR>	= Carriage return to indicate end of command

Application: @0z1, @0z0

Description: The data opening character "@0" prepares the controller for a new command. "z1" changes interpolation from 2D to 3D mode. This instruction works modally, i.e. all relative and absolute movements are executed three-dimensionally. With three axes, any z2 parameters entered for these movements are ignored. The interpolation speed must be entered with the x entry. With four axes, the fourth axis is positioned accordingly.

2.2.29 Move to a port event

Command: Move to port event

Purpose: Relative linear movement to port event or movement end.

Structure: @<GN>Z<Portno>,<Mask>,<Value>,<G>,<Sx>,<Sy>,<Sz>,<Sa><CR>

@	= Data opening character
<GN>	= Device number, standard=0
Z	= “Movement to port” command
<Portno>	= Port address
<Mask>	= Mask for masking bits
<Value>	= Desired value as condition for end of movement
<G>	= Speed
<Sx>	= Number of steps X
<Sy>	= Number of steps Y
<Sz>	= Number of steps Z
<Sa>	= Number of steps A
<CR>	= Carriage return to indicate end of command

Application: @0Z0,4,4,600,0,0,3000,0

Description: The controller is addressed by "@0"; Z indicates that a relative movement which can be ended by a port event is to be executed. The movement is carried out with a step frequency of 600Hz. The movement is ended when the numbers of steps entered have been reached, or when bit2 has been set to 1 at input port 0.

Comments: Please note that the number of parameters is dependent on the number of initialised axes. Hence, the following command structure applies for the different numbers of axes:

x-axis:	@<GN>Z<A>,<M>,<W>,<G>,<X><CR>
x-y-axis:	@<GN>Z<A>,<M>,<W>,<G>,<X>,<Y><CR>
x-y-z-axis:	@<GN>Z<A>,<M>,<W>,<G>,<X>,<Y>,<Z><CR>
x-y-z-a-axis:	@<GN>Z<A>,<M>,<W>,<G>,<X>,<Y>,<Z>,<A><CR>

2.2.30 Monitor and control codes

Monitoring and control codes make it possible to intervene directly in the functioning of the controller via the serial interface. Any commands sent in this way are directly and immediately analysed in the controller's receive routine and then executed. Specific control codes are available for the following functions:

Function: Software stop char(253)

A positioning movement, whether relative or absolute, in DNC mode can be stopped by a stop command without causing step losses. If a start command is subsequently executed (by entering "@0S"), the interrupted functioning process is ended. Moreover, after a stop command has been executed, the "request position" command can be used to read the position that has been reached. This function can also be used by operating the stop switch. If a movement has been successfully stopped, the controller will give an additional response, "F".

This function is operated by entering char(253) via the serial RS232 interface.

Function: Software reset char(254)

The controller will immediately interrupt all activities and execute a software reset internally. Then the system must be reinitialised and a reference movement must be executed.

This function is operated by entering char(254) via the serial RS232 interface.

Function: Software break char(255)

A positioning movement, whether relative or absolute, in DNC mode can be ended by a break command. This means that the rest of the movement will be forgotten.

This function is operated by entering char(255) via the serial RS232 interface.

2.3 Calculating circle parameters

2.3.1 Parameters for circle and helix interpolations

Circle interpolation is executed by two successive commands (see 2.2.8, 2.2.27). The first command sets the circle direction, and the second gives the interpolation parameters.

Circle direction: @<GN>f<Direction><CR>

@	= Data opening character
<GN>	= Device number, standard=0
f	= "Set circle direction" command
<Direction>	= 0 → CW (clockwise)
	-1 → ACW (anticlockwise)
<CR>	= Carriage return to indicate end of command

Circle interpolation: @<GN>y<B>,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry><CR>

@	= Data opening character
<GN>	= Device number, standard=0
<B>	= Arc length in steps
<V>	= Speed
<D>	= Interpolation parameter
<Xs>	= Start position x
<Ys>	= Start position y
<Rx>	= Direction x
<Ry>	= Direction y
<CR>	= Carriage return to indicate end of command

Inside the controller, a specially adjusted Bresenham differential algorithm is used to produce arcs. This kind of algorithm is very often used in microprocessor applications, because its low computational effort results in high speeds of execution.
The meaning and calculation of parameters for circle interpolation is explained below. See point 2.3.6 of this chapter for an example calculation.

2.3.2 Calculating the arc

Arc length <B> is the length of the arc in steps between the start and end positions of the arc, and it is used by the controller internally as the control variable for the differential algorithm. The arc length in steps can be calculated in different ways, as explained below:

- Simple approximate formula

  For simple circle applications, e.g. those which contain only a quarter-circle, semi-circle or whole circle, the arc length can be calculated using the following formula:

  B – Arc length in steps
  R – Arc radius in steps
  A – Start angle for arc measurement
  E – End angle for arc measurement

$$B = 4 * R * \frac{E - A}{Pi}$$

The result must be rounded to the nearest whole number. In order to eliminate any imprecision in the command to be executed by the controller, the next positioning is programmed as an absolute movement.

· Calculation using software routine

The arc length in steps can be calculated exactly using simple software routines.

2.3.3 Calculating the interpolation parameter

The controller uses the interpolation parameter as the start value for the differential register of the algorithm it uses to create circles. The PC calculates the parameter using the appropriate software routines. This increases the speed of execution by relieving the controller of unnecessary computational work.

2.3.4 Arc start position

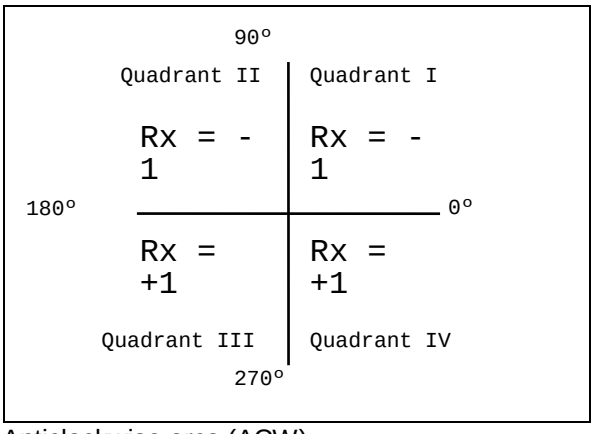
The arc start position represents the distance in steps along the x and y axes from the centre of the circle in relative coordinates (i.e. the centre of the circle is treated as the origin for this calculation). This can be calculated using the appropriate circle functions.

Xs - x-coordinate of start position relative to centre
Ys - y-coordinate of start position relative to centre
R - Radius in steps
A - Start angle in arc measurement

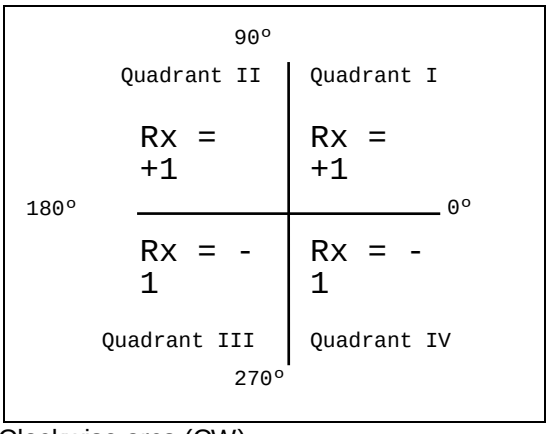
$$Xs = R * \cos(A)$$
$$Ys = R * \sin(A)$$

2.3.5 Directions in start position of circle interpolation

In order to execute the interpolation algorithm, the controller requires the following information: the quadrant in which the arc begins, and the characters to be used internally by the controller for specific calculations. This information is provided to the controller in the form of parameters Rx and Ry. The following definitions apply:



Anticlockwise arcs (ACW)



Clockwise arcs (CW)

2.3.6 Example circle interpolation calculation

To illustrate the calculation of parameters for the circle command, a brief example is given here.

An anticlockwise arc with a radius of 200 steps and a speed of 1500 steps per second is executed. The start angle is 135º, and the end angle is 225º. Please note that all distances and speeds in steps and all angles for arc measurement must be available.

Known:		To be established:	
Radius	R = 200	Arc length	B
Start angle	A = 135*Pi/180 = 2.3562	Start position X	Xs
End angle	E = 225*Pi/180 = 3.9267	Start position Y	Ys
Speed	V = 1500	Direction X	Rx
Direction	ACW	Direction Y	Ry
		Interpolation parameter	D

Arc length B (see 2.3.2):

B = 4 * R * (E - A)/Pi
B = 4 * 200 * (3.9267 - 2.3562) / Pi
B = 4 * 200 * 0.4999 = 399.9245
B = 400

Start position Xs and Ys (see 2.3.4):

Xs = R * cos(A) = 200 * cos(2.3562) = -141.4221
Xs = -141

Ys = R * sin(A) = 200 * sin(2.3562) = 141.4205
Ys = 141

Direction Rx and Ry (see 2.3.5):

Start angle 135º Direction of rotation ACW
Rx = -1 Ry = -1

Interpolation parameter D (see 2.3.3.):

D = (Rx*Ry*R + Rx*Ry*sum(R-1)
- Rx*sum(Xs+(Rx-Ry)/2)
+ Ry*sum(Ys+(Rx+Ry)/2))/2

sum(R-1) = sum(199)
= 199*(199+1)
= 39800

sum (Xs+(Rx-Ry)/2) = sum(-141+(-1-(-1))/2)
= sum(-141)
= 141*(-141+1)
= -19740

sum(Ys+(Rx+Ry)/2) = sum(141+(-1-(-1))/2)
= sum(141)
=141*(141+1)
= 20022

D = ((-1)*(-1)*200 + (-1)*(-1)*39800 - (-1)*(-19740) + (-1)*20022) / 2
D = (200 +39800 - 19740 - 20022) / 2 = 119

The commands would read:

@0f-1
@0y400,1500,119,-141,141,-1,-1

3 The CNC mode and its commands

3.1 Structure of the CNC commands

When running in CNC mode, the controller saves all the commands transmitted in an internal memory. Activation is executed by transmitting the “Save CNC data field” command following the standard initialisation. The data field is subsequently transferred and the process ends with the “Data field end” command.
The programme can now be activated by an external start command (pressing the start button) without the need for further communication with the control computer.

The storable iMC-xx controller commands are listed and briefly described below. More comprehensive information for certain commands can be looked up under the corresponding command in DNC mode, as the description and number of parameters equates to the relevant DNC mode command.

If an error occurs during the transmission and saving of a CNC data field, the CNC programme saved up until that point is marked as invalid and cannot be processed further. The error in the programme must then be rectified and the flash memory deleted before the data field can be re-submitted for saving.

3.2 The CNC mode commands

3.2.1 Save CNC data field

Command: Save CNC data field

Purpose: Initialising the transmission of storable commands. This command is required in order to start the CNC mode.

Structure: @<GN>i<CR>

@	= Data opening character
<GN>	= Device number, standard=0
i	= “Save CNC data field” command
<CR>	= Carriage return to indicate end of command

Example: @0i

Description: The controller is addressed via "@0". "i" indicates that a CNC data field is to be saved. The command is ended with a carriage return. The controller will then only accept CNC commands until the “End data field” command is received or an error occurs. The command is acknowledged with an appropriate response. All subsequent storable commands are then saved in the flash memory.

Restrictions:
The command can only be used when the controller has previously been initiated and no movements are being processed. An error message will be issued if a programmed is already stored in the memory.

3.2.2 Relative movement in CNC mode

Command: Relative movement

Purpose: The controller saves a relative movement corresponding to the submitted number of steps and step speed.

Structure: 0<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
0<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

0	= “Relative movement” command code
<Sx>	= Number of steps x
<Gx>	= Speed
<Sy>	= Number of steps y
<Gy>	= Speed
<Sz>,<Sz1>	= Number of steps z
<Sz2>	= Number of steps z, second movt. with 2.5D, three axes
<Gz>,<Gz1>	= Speed
<Gz2>	= Speed z, second movement with 2.5D, three axes
<Sa>	= Number of steps a, with four axes
<Ga>	= Speed, with four axes
<CR>	= Carriage return to indicate end of command

Example: 05000,900 (x-axis only)
050,900,20,9000 (x and y-axis)
030,800,10,900,4,90,-4,30(x, y and z-axis, with three axes)
030,800,10,900,4,90,-4,30(x, y, z and a-axis, with four axes)

Description: "0" indicates that a relative movement is to be executed. The controller then expects a coordinate pair for each axis, consisting of the number of steps and the speed. Distances are defined using relative dimensions i.e. based on the most recent position. The number of entries must agree with the number of axes i.e. one pair of parameters for an x-axis operation, two parameter pairs for xy operation, three parameter pairs for xyz operations and four parameter pairs for xyza operations. The individual values are separated using commas. Two pairs of values are expected for the z-axis when working with three axes and 2.5D, as the situation “Move, lower tool and then raise it when finished” is very common in processing applications. In 2.5D interpolation operation, the movement of the x and y axes is carried out first (linear interpolation), and the z-axis is then moved using the values stated in z1 and z2. If only one axis is to be moved, values must still be submitted for all initialised axes. A value of 0 should be entered for the number of steps relating to the stationary axes. The controller issues a handshake character ("0") after it has successfully saved the command.

Restrictions:
The controller does not verify that the movement remains within the permissible range for the connected mechanism.

When using 2.5D interpolation, the speed parameter for the axis with the longest path is defined as the track speed and the speed of the other axes is adapted depending on the path ratio. In contrast, in 3D interpolation operation the speed value for the x-axis is used as the standard value for specifying the track speed.

3.2.3 Send synchronisation character

Command: Send synchronisation character

Purpose: The controller sends an ASCII character via the serial interface.

Structure: 1<Z><CR>

7	= “Reference movement” command code
<Z>	=ASCII code (decimal value between 33 and 126)
<CR>	= Carriage return to indicate end of command

Example: 133

Description: "1" indicates that a synchronisation character is to be sent. “33” represents the ASCII character "!". The controller issues a handshake character (“0”) after it has successfully saved the command.

3.2.4 Wait for synchronisation character

Command: Wait for synchronisation character

Purpose: The controller waits to receive an ASCII character.

Structure: 2<Z>,<Offset><CR>

7	= “Reference movement” command code
<Z>	= ASCII code (decimal value between 33 and 126)
<Offset>	= Offset, jump destination when correct character received
<CR>	= Carriage return to indicate end of command

Example: 133,-10

Description: "1" indicates that a synchronisation character is expected. "33" represents the ASCII character "!". The controller interrupts the command processing until a character is received. If the character received equates to the defined synchronisation character (in this case "!"), the offset (-10 in this case) is applied, otherwise the processing is continued with the next command in the CNC programme. The controller issues a handshake character (“0”) after it has successfully saved the command.

3.2.5 Loops and jump to in CNC mode

Command: Loop, jump to (offset)

Purpose: Save loops and jump to commands. Loops are used to repeat identical types of movements. This allows the controller's memory to be used more efficiently. Jump to commands can be used to jump directly to a particular place in the programme based on a logical decision.

Structure: 3<Number>,<Offset><CR>

3	= "Loop, offset" command code	
<Number>	= Number of loops	
	Loop:	0 < number of loops < 32768
	Offset:	always 0
<Offset>	= Jump to row	
	Loop:	-1 >= Jump target >= -32768
	Offset:	-32768 <= Jump target <= 32767
<CR>	= Carriage return to indicate end of command	

Example:	3 25,-1	Repeat the most recent command 25 times
	3 0,-5	Always jump back five steps
	3 0,5	Hop over the next four commands
	3 6,-5	Repeat the five most recent commands six times

Description: If the controller encounters the "loop/offset" command in a CNC programme, a decision is first made, by checking the number of loops, to see if the command relates to loops or offsets. If it is a loop command, a loop counter is established and set and the command counter is corrected by the stated offset. The commands executed up to the loop counter are repeated and the loop counter decreases until it reaches it zero. The first command after the loop is then executed. Loops can be nested within each other with a nest depth of seven levels. The counters required are administered on a loop stack. If the command is a jump to command, the offset is interpreted as a relative jump within the CNC programme and the command counter is adjusted in line with the offset.

Restrictions:

It is not permissible to make a jump that goes to a place before the beginning or after the end of the data field. Forwards loops are not permitted. A loop always repeats the last n commands. At least one command must be repeated. Loops may be nested and the maximum nesting depth is seven levels. A loop cannot be exited via a jump to command.

3.2.6 Time delay in CNC mode

Command: Time delay

Purpose: Saving of time delays.

Structure: 5<Time><CR>

5	= “Time delay” command code
<Zeit>	= Time in 1/10 sec
<CR>	= Carriage return to indicate end of command

Example: 3 50 Time delay five seconds

Description: If the controller encounters the “time delay” command in a CNC programme, the next command in the CNC programme is executed only after the time specified in the delay has elapsed. Times are stated in tenths of seconds.

Restrictions:
A time delay cannot be stopped by pressing the controller’s stop button.

3.2.7 Move to a port event in CNC mode

Command: Definition of a port condition for movement until a port event occurs

Purpose: The controller saves the condition for a movement until a port event occurs. This condition is applied to the next subsequent movement.

Structure: 6<Portnr>,<Bitnr>,<Value><CR>

6	= “Port condition” command code
<Portnr>	= Port number
<Bitnr>	= Bit number, 1 - 8 --> bitwise, 128 --> bytewise
<Wert>	= Comparison value
<CR>	= Carriage return to indicate end of command

Example:	60,128,1	following movement will be ended, if port 0 == 1
	60,1,0	following movement will be ended, if Port0, Bit1 == 0

Description: "6" indicates that the condition for a movement until a port event occurs is to be defined. This condition is applied to the next movement. During the movement, the relevant port is interrogated and logically compared with the comparison value in terms of bits and bytes. If the logical comparison gives a positive result, the movement is terminated.

Port	Bit	Condition	Function
0	0 - 7	00 - FF	User E/A

3.2.8 Reference movement in CNC mode

Command: Reference movement

Purpose: The controller stores a movement of all the defined axes to their origin (reference points). With isel systems, the reference points for the axes

are always defined in a suitable default configuration, but can be adapted using relevant initialisation commands.

Structure: 7<Axes><CR>

7	= “Reference movement” command code
<Axes>	= Axes specification, see below
<CR>	= Carriage return to indicate end of command

Example: 77, 78

Description: "7" indicates that a reference movement is to be executed. The associated value defines the axes that are to have a reference movement applied. Each axis is defined by a bit of a binary value, which results in the following values:

1 --> X axis
2 --> Y axis
3 --> X+Y axes
4 --> Z axis
5 --> X+Z axes
6 --> Y+Z axes
7 --> X+Y+Z axes
8 --> A axis

The sequence for the execution is defined as follows:
Z-axis --> Y- axis --> X- axis --> A- axis

Following a successfully executed reference movement, the next CNC command is read and processed from the memory.

Restrictions:

The command is limited to the initialised axis configuration. In the event of an incorrect entry of axes, the "3" error message is issued. If the controller is in 3D mode, the command switches back to the 2.5D model.

NOTE: The A-axis must always be referenced separately.

NOTE: For reference switches that are not connected, the relevant axis is permanently active. However, the stop button can be used to halt the reference movement.

3.2.9 Data fields in CNC mode

Command: End data field

Purpose: Indicating the end of a CNC data field and specifying the termination of a data transmission and the saving of storable commands.

Structure: 9<CR>

9 = “End data field” command code
<CR> = Carriage return to indicate end of command

Example: 9

Description: "9" defines that the end of the CNC data field being transmitted has been reached. The command is ended with a carriage return. The controller responds either with the "0" (software handshake) if the save operation has been successful or with an error message. In addition to the designation of the data field as a valid CNC programme, status information (e.g. the current reference speed) is saved in the flash memory. The controller is subsequently in DNC mode again and accepts the relevant commands.

Restrictions:

A CNC data field must be terminated with the end data field command, otherwise the saved CNC programme is not valid and cannot be processed.

3.2.10 Define reference speed in CNC mode

Purpose: Specifying the speed at which a reference movement is executed.

Structure: d<Gx>,<Gy>,<Gz>,<Ga><CR>

d = “Set reference speed” command
<Gx> = Reference speed for X-axis
<Gy> = Reference speed for Y-axis
<Gz> = Reference speed for Z-axis
<Ga> = Reference speed for A-axis
<CR> = Carriage return to indicate end of command

Example: d2500,1000,1200,800

Description: If the controller is not provided with any information about the reference speed, the movement is executed using a default value. A changed value is not retained after switching off.

Restrictions: The defined speed must lie within the valid range of values for speeds. If the reference speed selected is too high and the spindle pitch is too high, the inertia may damage the reference switches. The controller requires the connected origin switch to exhibit hysteresis. Please bear this in mind when connection electronic null sensors!

3.2.11 Select plane for circle interpolation in CNC mode

Command: Plane selection

Purpose: Saving the interpolation plane for circle interpolation. Circles are only defined within a single plane. The default plane for circle interpolation is the XY plane. The “plane selection” command provides an option to define any other plane configuration as a circle plane.

Structure: e<Ebene><CR>

e	= “Set circle plane” command code
<Ebene>	= Plane definition, see below
<CR>	= Carriage return to indicate end of command

Example: e1, e0

Description: "e" indicates that the level for circle interpolation needs to be defined. The subsequent figure defines the plane as follows:

0 --> XY plane
1 --> XZ plane
2 --> YZ plane

Restrictions:

This command is modal i.e. a mode selection for circle interpolation is retained until it is overwritten by a new definition.

3.2.12 Set circle direction for circle interpolation in CNC mode

Command: Set circle direction

Purpose: Saving the circle direction for circle interpolation. The circle interpolation is programmed using two consecutive commands. The first command defines the circle direction and the second (see 3.2.9.) transmits the interpolation parameters.

Structure: f<Direction><CR>

f	= “Set circle direction” command code
<Direction>	= 0 → CW (clockwise)
	-1 → ACW (anticlockwise)
<CR>	= Carriage return to indicate end of command

Example: f-1, f0

Description: "f" indicates that the direction for circle interpolation is to be set. The subsequent figure defines the direction as follows:

0 → CW (clockwise circle interpolation)
-1 → ACW (anticlockwise circle interpolation)

Restrictions:

The direction for circle interpolation is, in principle, to be programmed prior to each circle movement.

3.2.13 Brake switch in CNC mode

Command: Brake switch

Purpose: Enabling the operation of a brake, if there is one.

Structure: g<Value><CR>

g	= Brake switch
< Value >	= 0 → Brake not powered
	1 → Brake powered
<CR>	= Carriage return to indicate end of command

Example: g1

Description: The “g1” command can be used to power (i.e. open) the brake; “g0” closes the brake.

3.2.14 Set start-stop frequency in CNC mode

Command: Set the start-stop frequency

Purpose: Specifying the frequency with which the ramps necessary for movements begin and end.

Structure: j<F><CR>

j	= Set start-stop frequency
<F>	= Frequency
<CR>	= Carriage return to indicate end of command

Example: j300

Description: If the controller is not provided with any information about frequency, it will use a default value to execute the command. Changed values are not retained when the controller is switched off.
The value for the start-stop frequency is stated in Hz and the default value is 300 Hz. The expected range of frequencies lies between 20Hz and 4000Hz.

Restrictions: The defined start-stop frequency must lie within the valid range of values for step frequencies.
A start-stop frequency that is too high can lead to step losses in the acceleration and braking phases.

3.2.15 Set acceleration in CNC mode

Command: Set acceleration

Purpose: Specifying the acceleration with which the ramps necessary for movements are executed.

Structure: J<B><CR>

J	= Set acceleration
<B>	= Acceleration
<CR>	= Carriage return to indicate end of command

Example: J100

Description: If the controller is not provided with any information about acceleration, it executes the command using a default value. A changed value is not retained after the controller is switched off.
The value for the acceleration is stated in Hz/ms and the default value is 100 Hz/ms. The expected range of acceleration lies between 1Hz/ms and 4000 Hz/ms.

Restrictions: The defined acceleration must lie within the valid range of values for acceleration.
An acceleration that is too high can lead to step losses in the acceleration and braking phases.

3.2.16 Absolute movement in CNC mode

Command: Movement to an absolute position

Purpose: The controller saves an absolute movement defined by the stated speeds and positions.

Structure: m<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
m<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

m	= “Absolute movement” command code
<Sx>	= Position x
<Gx>	= Speed
<Sy>	= Position y
<Gy>	= Speed
<Sz>,<Sz1>	= Position z
<Sz2>	= Position z, 2. Movement always = 0
<Gz>,<Gz1>	= Speed
<Gz2>	= Speed
<Sa>	= Position a, with four axes
<Ga>	= Speed, with four axes
<CR>	= Carriage return to indicate end of command

Example:
m5000,900 (only x-axis)
m50,900,20,9000 (x and y-axis)
m30,800,10,900,4,90,0,30 (x, y and z-axis, with three axes)
m30,800,10,900,4,90,4,30 (x, y, z and a-axis, with three axes)

Description: "m" indicates that an absolute position is about to be stated. In order to provide compatibility with the relative position command, two pairs of figures are expected for the z-axis in the event of three axes being involved. The second set of positional information for the z-axis must be set to zero and is ignored. Following the successful saving, the controller responds with the handshake character.

Restrictions:

The controller does not verify that the movement remains within the permissible range for the connected mechanism.

3.2.17 Set origin in CNC mode

Command: Set the current point as the origin

Purpose: The controller saves a command that sets the current position during the processing of the CNC programme as a virtual origin for the stated axe(s). The following “Absolute movement” instructions then relate to the virtual origin.

Structure: n<Axes><CR>

n	= “Set origin” command code
<Axes>	= Axis definition, see below
<CR>	= Carriage return to indicate end of command

Example: n7, n1, n8

Description: "n" indicates that an origin shift is to be executed. Following this command code, the controller is informed of the axes that are to be allocated a new origin. Each axis is defined by a bit of a binary value, which results in the following values:

1 --> X axis
2 --> Y axis
3 --> X+Y axes
4 --> Z axis
5 --> X+Z axes
6 --> Y+Z axes
7 --> X+Y+Z axes
8 --> A axis

The controller provides confirmation after the command is saved successfully.

Restrictions:

The virtual origin is only relevant for the “Absolute movement” command. Relative position movements are not affected by a virtual origin, as they state the vector to be moved.

NOTE: The origin shift for the A-axis must always be carried out separately.

3.2.18 Simulate reference movement in CNC mode

Command: Simulate reference movement

Purpose: The controller saves a command that sets the current position during the processing of the CNC programme as a reference point.

Structure: N<Axes><CR>

N	= “Simulate reference” command code
<Achsen>	= Axis definition, see below
<CR>	= Carriage return to indicate end of command

Example: N7, N1, N8

Description: "N" indicates that a reference movement is to be simulated. Following this command code, the controller is informed of the axes for which a reference movement is to be simulated. Each axis is defined by a bit of a binary value, which results in the following values

1 --> X axis
2 --> Y axis
3 --> X+Y axes
4 --> Z axis
5 --> X+Z axes
6 --> Y+Z axes
7 --> X+Y+Z axes
8 --> A axis

The controller provides confirmation after the command is saved successfully.

Restrictions:

The virtual reference point is only relevant for the “Absolute movement” command. Relative position movements are not affected by a virtual reference point, as they state the vector to be moved.

NOTE: The reference shift for the A-axis must always be carried out separately.

3.2.19 Read port and jump to in CNC mode

Command: Read input port

Purpose: The input port is read and a jump occurs in the programme sequence. The jump involves a logical comparison with a particular record resulting in a jump (offset) being carried out within the programme.

Structure: o<Port no.>,<Bit no.>,<Value>,<Offset><CR>

o	= “Read port” command code
<Port no.>	= Port number
<Bit no.>	= Bit number, 0 - 7 --> bitwise, 128 --> bytewise
<Value>	= Comparison value
<Offset>	= Jump target-32768 <= Jump target <= 32767
<CR>	= Carriage return to indicate end of command

Example:	o0,128,1,-1	wait until Port0 <> 1
	o0,0,1,-1	wait until Port0, Bit0 = 0
	o0,0,1,3	if Port0, Bit0 == 1, command counter += 3

Description: "o" indicates that the value of an input port is to be read and that the programme sequence is to be adjusted based on the resulting value. The port number, the bit number, the comparison value and the command offset are then all submitted, separated by a comma, before the command is closed with a carriage return. If the command is successfully saved, the controller responds with a "0" software handshake, or with an error message if an incorrect port number and/or value is submitted. The relevant port number is interrogated as the programme runs and is compared logically, in terms of bits or bytes, with the comparison value. If the logical comparison is true, an offset (jump) is applied, otherwise the next command in the sequence is invoked. The following ports and related functionality are defined for iMC-xx controllers:

Port	State	Function
0	00 - FF	User E/A
1	00 - FF	Status inputs
		Bit0 Input emergency stop circle 1
		Bit1 Input emergency stop circle 2
		Bit2 Input excess temperature output stage
		Bit3 Input start button
		Bit4 Input stop button
		Bit5 Input linear measuring sensor
		Bit6 not in use
		Bit7 not in use
2	00 - FF	Status inputs
		Bit0 Input spindle
		Bit1 Input power Ok
		Bit2 not in use
		Bit3 not in use
		Bit4 not in use
		Bit5 Input close cover and automatic
		Bit6 Input close cover or setup
		Bit7 Input key switch
3	00 - FF	End switch inputs
		Bit0 End switch X1
		Bit1 End switch X2
		Bit2 End switch Y1
		Bit3 End switch Y2

Bit4	End switch Z1
Bit5	End switch Z2
Bit6	End switch A1
Bit7	End switch A2

Restrictions:

The port input is interrogated during controlling of the programme sequence. Thus, it is not possible to interrogate inputs during the processing of a command e.g. during a positional movement.

3.2.20 Set port in CNC mode

Command: Set output port

Purpose: Specifying the turning on/off of available output ports.

Structure: p<Port no.>,<Bit no.>,<Value><CR>

p	= “Set port” command code
<Portnr>	= Port number
<Bitnr>	= Bit number, 0 - 7 --> bitwise, 128 --> bytewise
<Wert>	= New value
<CR>	= Carriage return to indicate end of command

Example:

p0,128,1	Port 0, set bytewise to 1
p0,0,1	Port 0 , set Bit 0 to 1

Description: "p" indicates that the value of an output port is to be set. The port number, the bit number and the new port value are then all submitted, separated by a comma, before the command is closed with a carriage return. If the command is successfully saved, the controller responds with a "0" software handshake, or with an error message if an incorrect port number and/or value is submitted. The following ports and related functionality are defined for iMC-xx controllers:

Port	Value	Function
0	0 - 255	User E/A
1	0	Cover may not be opened
	1	Cover may be opened
2	0	Turn off spindle
	1	Turn on spindle
3	0	Turn off motor currents
	1	Turn on motor currents
4	0 - 255	Analog output 0 – 10V
5	0	Turn off current reduction
	1	Turn on current reduction
6	0	Turn off brake
	1	Turn on brake
100	0 - 255	Control Out
		Bit0 Output for current reduction
		Bit1 not in use
		Bit2 Output for reset output stages

		Bit3 Output enable output stages
		Bit4 Output spindle
		Bit5 Output enable cover
		Bit6 Output brake
		Bit7 Output SoftwareOk
101	0 - 255	Signal Out
		Bit0 not in use
		Bit1 not in use
		Bit2 not in use
		Bit3 not in use
		Bit4 not in use
		Bit5 Output stop lamp
		Bit6 Output start lamp
		Bit7 Output error lamp

Restrictions: The port output is set during controlling of the programme sequence. Thus, it is not possible to set or delete outputs during the processing of a command e.g. during a positional movement.

3.2.21 Turn test mode on/off in CNC mode

Command: Turn test mode on/off

Purpose: Turning the test mode on or off, as required.

Structure: T<Status><CR>

T	= Command turn on/off test mode
<Status>	= 0 --> turn off, 1 --> turn on
<CR>	= Carriage return to indicate end of command

Example: T1, T0

Description: "T1" turns the test mode on and "T0" turns it off. Following successful completion, the controller issues a handshake character ("0"). The controller handles the end switch and reference movement differently in test mode than in normal operational mode. If a reference movement command is received while in test mode, the controller does not actually execute a reference movement as such but just sets the current position as the reference point. The end switch can still be monitored but will be overridden. This is very useful if an axis is at the end switch when the system is turned on and needs to be released.

3.2.22 Helix interpolation in CNC mode

Command: Helix interpolation

Purpose: Saving movement commands for helix interpolation.

Structure: w<B>,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry>,<S3><CR>

w	= “Circle interpolation” command code
<B>	= Arc length in steps
<V>	= Speed
<D>	= Interpolation parameter
<Xs>	= Start point axis1
<Ys>	= Start point axis2
<Rx>	= Direction axis1
<Ry>	= Direction axis2
<S3>	= Number of steps axis3
<CR>	= Carriage return to indicate end of command

Example: w32000,600,-1000,-2000,0,1,-1,6000

Description: "w" indicates that a helix interpolation is to be saved. The arc length defines the length of the arc section in steps between the start and end points of the circle segment. All integer values within the valid range of values are permitted for the speed. The interpolation parameter serves as a starting value for the interpolation algorithm used. The parameters Xs and Ys define the starting point of the arc section relative to the helix centre point. Rx and Ry indicate the quadrant in which the interpolation is to be initiated. Following successful completion, the controller issues a handshake character ("0").

NOTE: Regarding calculation of the parameters, please read the chapter "Calculation of parameters for circle interpolation".

Restrictions:

The controller does not verify that the movement remains within the permissible range for the connected mechanism

3.2.23 Circle interpolation in CNC mode

Command: Circle interpolation

Purpose: Saving movement commands for circles and circular arcs with constant track speed. The circle interpolation is triggered by two consecutive commands. The first defines the circle direction (see 3.2.12.) and the second transmits the interpolation parameters.

Structure: y<B>,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry><CR>

y	= “Circle interpolation” command code
<B>	= Arc length in steps
<V>	= Speed
<D>	= Interpolation parameter
<Xs>	= Start point x
<Ys>	= Start point y
<Rx>	= Direction x
<Ry>	= Direction y

<CR> = Carriage return to indicate end of command

Example: y400,1500,119,-141,141,-1,-1

Description: "y" indicates that a circle interpolation is to be saved. The arc length defines the length of the arc section in steps between the start and end points of the circle interpolation. All integer values within the valid range of values are permitted for the speed. The interpolation parameter serves as a starting value for the interpolation algorithm used. The parameters Xs and Ys define the starting point of the arc section relative to the circle centre point. Rx and Ry indicate the quadrant in which the interpolation is to be initiated. Following successful completion, the controller issues a handshake character ("0").

NOTE: Regarding calculation of the parameters, please read the chapter "Calculation of parameters for circle interpolation".

Restrictions:

The controller does not verify that the movement remains within the permissible range for the connected mechanism.

3.2.24 Turn 3D interpolation on/off in CNC mode

Command: Turn 3D linear interpolation on/off

Purpose: The controller saves a command that expands the 2.5D interpolation of the operating system to a three-dimensional interpolation. This command allows the interpolation to be turned on and off as desired.

Structure: z<Status><CR>

z = “3D interpolation” command code
<Status> = 0 --> turn off, 1 --> turn on
<CR> = Carriage return to indicate end of command

Example: z1, z0

Description: "z1" changes the interpolation from 2D to 3D operation. The instruction is modal i.e. all relative and absolute movements are executed three dimensionally. The specification of z2-parameters in the event of three axes is ignored for these movements. The speed of the interpolation must be specified in conjunction with the x specification. If there are four axes, the fourth axis is updated accordingly.

4 Initialisation commands

4.1 Fundamentals

iMC series controllers are equipped with an internal flash memory so that initialisation values can be saved in the controller. This makes it possible to adapt the controller's operating system more effectively to the hardware and operational environment. The initialisation and default values can be set using relevant commands and then saved on a permanent basis in the controller's flash memory using the "Write initialisation values to flash" command. When the controller is turned on, these values are then automatically used as initialisation and default values for the relevant controller parameters.

4.2 Details of the individual commands

4.2.1 Set default reference speed

Command: Set the default reference speed

Purpose: Initialising axis and controller parameters.

Structure: @<GN>Id<ValueX>,<ValueY>,<ValueZ>,<ValueA><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= "Initialisation" command
d	= Command code
<ValueX>	= Default reference speed X-axis
<ValueY>	= Default reference speed Y-axis
<ValueZ>	= Default reference speed Z-axis
<ValueA>	= Default reference speed A-axis
<CR>	= Carriage return to indicate end of command

Example: @0Id1000,900,500,900

Description: The controller is prepared for a new command with the issuing of the data opening segment "@0". "I" informs the controller that an initialisation is to be carried out. This is followed by the command code "d", which defines the default reference speed. Step frequencies separated by commas are specified for each axis. The carriage return indicates the end of the command.

Restrictions:
The defined step frequencies must lie within the valid range for speeds.

4.2.2 Number of steps for releasing the reference switch

Command: Define the steps for releasing the reference switch

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IF<ValueX>,<ValueY>,<ValueZ>,<ValueA><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
F	= Command code
<ValueX>	= Default reference speed X-axis
<ValueY>	= Default reference speed Y-axis
<ValueZ>	= Default reference speed Z-axis
<ValueA>	= Default reference speed A-axis
<CR>	= Carriage return to indicate end of command

Example: @0IF20,50,50,10

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “F”, which defines the number of steps for releasing the reference switch. The number of steps, separated by commas, is specified as a parameter for each axis. The carriage return indicates the end of the command. A reference movement proceeds as follows:

- movement required in the reference switch (with reference speed) in order for the switch to open,
- step-wise movement of the switch until the switch closes again,
- retraction of the switch with the number of steps and the reference speed specified here.

Restrictions:

4.2.3 Set default acceleration

Command: Set default acceleration

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IJ<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
J	= Command code
<Value>	= Default value for acceleration
<CR>	= Carriage return to indicate end of command

Example: @0IJ120

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “J”, which defines the default acceleration. The acceleration parameter is stated in Hz/ms. The carriage return indicates the end of the command.

Restrictions:

4.2.4 Set default start-stop frequency

Command: Set default start-stop frequency

Purpose: Initialising axis and controller parameters.

Structure: @<GN>Ij<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
j	= Command code
<Value>	= Default value for the start-stop frequency
<CR>	= Carriage return to indicate end of command

Example: @0Ij300

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “j”, which defines the default start-stop frequency. The start-stop frequency parameter is stated in Hz. The carriage return indicates the end of the command.

Restrictions:

4.2.5 Enable IO-only mode

Command: Enable IO-only mode

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IO<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
O	= Command code
<Value>	= Enable/Disable IO-Only 1 = Enable 0 = Disable
<CR>	= Carriage return to indicate end of command

Example: @0IO1

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an

initialisation is to be carried out. This is followed by the command code “O”, which enables/disables the IO-only mode. A parameter of 1 enables the mode and 0 disables it. The carriage return indicates the end of the command.

Restrictions:

4.2.6 Enable/disable end switch and set active level

Command: Enable/disable end switch and set active level

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IE<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
E	= Command code
<Value>	= 16 bit value as code to enable/disable the end switch and set the active level
<CR>	= Carriage return to indicate end of command

Example: @0IE57343

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “E”, which defines the end switch functionality. A 16 bit value is transmitted as the parameter in the form of a decimal number. The carriage return indicates the end of the command. The following coding is available for <Value>.

Bit	Value	Function
0	0	End switch1 X-Axis disabled
	1	End switch1 X-Axis enabled
1	0	End switch2 X-Axis disabled
	1	End switch2 X-Axis enabled
2	0	End switch1 X-Axis high-active
	1	End switch1 X-Axis low-active
3	0	End switch2 X-Axis high-active
	1	End switch2 X-Axis low-active
4	0	End switch1 Y-Axis disabled
	1	End switch1 Y-Axis enabled
5	0	End switch2 Y-Axis disabled
	1	End switch2 Y-Axis enabled
6	0	End switch1 Y-Axis high-active
	1	End switch1 Y-Axis low-active
7	0	End switch2 Y-Axis high-active
	1	End switch2 Y-Axis low-active
8	0	End switch1 Z-Axis disabled

	1	End switch1 Z-Axis enabled
9	0	End switch2 Z-Axis disabled
	1	End switch2 Z-Axis enabled
10	0	End switch1 Z-Axis high-active
	1	End switch1 Z-Axis low-active
11	0	End switch2 Z-Axis high-active
	1	End switch2 Z-Axis low-active
12	0	End switch1 A-Axis disabled
	1	End switch1 A-Axis enabled
13	0	End switch2 A-Axis disabled
	1	End switch2 A-Axis enabled
14	0	End switch1 A-Axis high-active
	1	End switch1 A-Axis low-active
15	0	End switch2 A-Axis high-active
	1	End switch2 A-Axis low-active

Restrictions:

4.2.7 Invert end switch

Command: Invert end switch

Purpose: Initialising axis and controller parameters.

Structure: @<GN>Ie<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
e	= Command code
<Value>	= Axis encoding
<CR>	= Carriage return to indicate end of command

Example: @0Ie4

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “e”, which inverts end switches. Axis encoding is entered as a parameter in the form of a decimal number. The carriage return indicates the end of the command. The axis encoding is defined as follows:

Bit0	→	X-axis
Bit1	→	Y-axis
Bit2	→	Z-axis
Bit3	→	A-axis

The decimal value results from adding the values of the individual bits. For example, if only the end switch for the x-axis is to be changed, the

value is “1”, whereas the value if the end switches for all the axes are to be changed is “15”.

Restrictions:

4.2.8 Invert axis direction

Command: Invert axis direction

Purpose: Initialising axis and controller parameters.

Structure: @<GN>ID<Value><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
D	= Command code
<Value>	= Axis encoding
<CR>	= Carriage return to indicate end of command

Example: @0ID4

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “D”, which inverts the axis direction. Axis encoding is entered as a parameter in the form of a decimal number. The carriage return indicates the end of the command. The axis encoding is defined as follows:

Bit0	→	X-axis
Bit1	→	Y-axis
Bit2	→	Z-axis
Bit3	→	A-axis

The decimal value results from adding the values of the individual bits. For example, if only the direction of the z-axis is to be changed, the value is “4”, whereas the corresponding value if the directions of all the axes are to be changed is “15”.

Restrictions:

4.2.9 Invert reference directions

Command: Invert reference directions

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IR<Value><CR>

@	= Data opening character

<GN> = Device number, standard=0
I = “Initialisation” command
R = Command code
<Value> = Axis encoding
<CR> = Carriage return to indicate end of command

Example: @0IR2

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “R”, which inverts reference directions. Axis encoding is entered as a parameter in the form of a decimal number. The carriage return indicates the end of the command. The axis encoding is defined as follows:

Bit0 → X-axis
Bit1 → Y-axis
Bit2 → Z-axis
Bit3 → A-axis

The decimal value results from adding the values of the individual bits. For example, if only the reference direction of the y-axis is to be changed, the value is “2”, whereas the corresponding value if the reference directions of all the axes are to be changed is “15”.

Restrictions:

4.2.10 Set axis type

Command: Set axis type

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IT<Value><CR>

@ = Data opening character
<GN> = Device number, standard=0
I = “Initialisation” command
R = Command code
<Value> = Axis encoding
<CR> = Carriage return to indicate end of command

Example: @0IT8

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “T”, which is for defining axis types. Axis encoding is entered as a parameter in the form of a decimal number. The carriage return indicates the end of the command. The axis encoding is defined as follows:

Bit0	→	X-axis	(0=linear, 1=rotational)
Bit1	→	Y-axis	(0=linear, 1= rotational)
Bit2	→	Z-axis	(0=linear, 1= rotational)
Bit3	→	A-axis	(0=linear, 1= rotational)

The decimal value results from adding the values of the individual bits. For example, if the a-axis is to be handled as a rotational axis, the value is “8”.

Restrictions:

4.2.11 Abort initialisation via port input in CNC mode

Command: Define port input allocation for which an abort command will be executed in CNC mode

Purpose: Initialising the abort command.

Structure: @<GN>IA<Port>,<Mask>,<Target><CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
A	= Command code
<Port>	= Port number
<Mask>	= Mask
<Target>	= Target value
<CR>	= Carriage return to indicate end of command

Example: @0IA0,1,1

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “A”, which is for defining the port for the generation of an abort command in CNC mode. The parameters are a port number, a mask and a target value. When the CNC mode is active, the port is read in cycles, and the actual and target values are masked using the mask (UND link) and compared. An abort command is generated if the two values are not the same.
If this setting is to be used on a regular basis, the desired setting should be saved on a more permanent basis in the flash memory using the “Write initialisation values to flash” command. The function is disabled if the mask is set to “0”.

Restrictions:

4.2.12 Read initialisation values from flash

Command: Read initialisation values from flash

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IL<CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
L	= Command code
<CR>	= Carriage return to indicate end of command

Example: @0IL

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “L”, which is for the reading of initialisation values from the flash memory. The carriage return indicates the end of the command. The values are read from the flash and the respective parameters are initialised.

Restrictions:

4.2.13 Write initialisation values to flash

Command: Write initialisation values to flash

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IW<CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
W	= Command code
<CR>	= Carriage return to indicate end of command

Example: @0IW

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “W”, which is for the writing of initialisation values to the flash memory. The carriage return indicates the end of the command. The current initialisation parameter values are thus saved to the flash memory.

Restrictions:

4.2.14 Load default initialisation values

Command: Load default initialisation values

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IX<CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
X	= Command code
<CR>	= Carriage return to indicate end of command

Example: @0IX

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “X”, which is for the loading of default initialis ation values. The carriage return indicates the end of the command. The parameters are initialised using default values defined internally in the controller.

Restrictions:

4.2.15 Initialization error signal via output port in CNC mode

Command: Define the port output assignment, which provides an error signal in CNC mode

Purpose: Initialization error signal

Structure: @<GN>Ia<Port>,<Mask>,<Value><CR>

@ = Data opening character
<GN> = Device number, standard=0
I = „Initialization“ command
a = Command code
<Port> = Port number
<Mask> = Mask
<Value> = Value
<CR> = Carriage return to indicate end of command

Example: @0Ia0,1,1

Description: The controller is prepared for a new command with the issuing of the data opening segment "@0". "I" informs the controller that an initialisation is to be carried out. This is followed by the command code "A", which is for defining the initialization error signal in CNC mode. The parameters are a port number, a mask and a target value. When the CNC mode is active, the port is write in cycles, the value is masked with the mask and output . In case of error the specified value is output, otherwise the bitwise negated value.
Should this setting be used permanently, the current setting with the command must be "initialization write in Flash" are permanently stored in

Command: Load default initialisation values

Purpose: Initialising axis and controller parameters.

Structure: @<GN>IX<CR>

@	= Data opening character
<GN>	= Device number, standard=0
I	= “Initialisation” command
X	= Command code
<CR>	= Carriage return to indicate end of command

Example: @0IX

Description: The controller is prepared for a new command with the issuing of the data opening segment “@0”. “I” informs the controller that an initialisation is to be carried out. This is followed by the command code “X”, which is for the loading of default initialis ation values. The carriage return indicates the end of the command. The parameters are initialised using default values defined internally in the controller.

Restrictions:

4.2.15 Initialization error signal via output port in CNC mode

Command: Define the port output assignment, which provides an error signal in CNC mode

Purpose: Initialization error signal

Structure: @<GN>Ia<Port>,<Mask>,<Value><CR>

@ = Data opening character
<GN> = Device number, standard=0
I = „Initialization“ command
a = Command code
<Port> = Port number
<Mask> = Mask
<Value> = Value
<CR> = Carriage return to indicate end of command

Example: @0Ia0,1,1

Description: The controller is prepared for a new command with the issuing of the data opening segment "@0". "I" informs the controller that an initialisation is to be carried out. This is followed by the command code "A", which is for defining the initialization error signal in CNC mode. The parameters are a port number, a mask and a target value. When the CNC mode is active, the port is write in cycles, the value is masked with the mask and output . In case of error the specified value is output, otherwise the bitwise negated value.
Should this setting be used permanently, the current setting with the command must be "initialization write in Flash" are permanently stored in

the flash. If the mask is set to "0", this function is turned off.

Restrictions:

4.2.16 Initialization hood monitoring via port input in CNC mode

Command: Define port input assignment, in which an externally attached hood can be monitored in CNC mode.

Purpose: Initialization hood monitoring

Strucutre: @<GN>IH<Port>,<Mask>,<Traget><CR>

@ = Data opening character
<GN> = Device number, standard=0
I = „Initialization“ command
H = Command code
<Port> = Port number
<Mask> = Mask
<Target> = Target value
<CR> = Carriage return to indicate end of command

Example: @0IH0,2,1

Description: The controller is prepared for a new command with the issuing of the data opening segment "@0". „"I" informs the controller that an initialisation is to be carried out. Then, the command code is followed for setting the ports to monitor an external hood in CNC mode "H". The parameters are a port number, a mask and a target value. When the CNC mode is active, the port is read in cycles, and the actual and target values are masked using the mask (UND link) and compared. The hood is open if the two values are not the same.
If this setting is to be used on a regular basis, the desired setting should be saved on a more permanent basis in the flash memory using the "Write initialisation values to flash" command. The function is disabled if the mask is set to "0".

Restrictions:

4.2.17 Initialization of an external emergency OFF via port input in CNC mode

Command: Define port input assignment, in which an external emergency OFF can be monitored in CNC mode.

Purpose: Initialization emergency OFF

Structure: @<GN>IN<Port>,<Mask>,<Taregt><CR>

@ = Data opening character
<GN> = Device number, standard=0
I = „Initialization“ command

N = Command code
<Port> = Port number
<Mask> = Mask
<Target> = Target value
<CR> = Carriage return to indicate end of command

Example: @0IN0,3,1

Description: The controller is prepared for a new command with the issuing of the data opening segment "@0". „"I" informs the controller that an initialisation is to be carried out. Then, the command code is followed for setting the ports to monitor an external emergency OFF in CNC mode "N".
The parameters are a port number, a mask and a target value. When the CNC mode is active, the port is read in cycles, and the actual and target values are masked using the mask (UND link) and compared. The emergency OFF signal is recognized if the two values are not the same.
If this setting is to be used on a regular basis, the desired setting should be saved on a more permanent basis in the flash memory using the "Write initialisation values to flash" command. The function is disabled if the mask is set to "0".

Restrictions:

4.2.18 Set an UserInfo

Command: set an UserInfo

Purpose: Initialization of a customized UserInfo

Structure: @<GN>IU<Text><CR>

@ = Data opening character
<GN> = Device number, standard=0
I = „Initialization“ command
U = Command code
<Text> = Plaintext max. 32 characters
<CR> = Carriage return to indicate end of command

Example: @0IUisel_1234_aabbcc_24122012

Description: The controller is prepared for a new command with the issuing of the data opening segment "@0". „"I" informs the controller that an initialisation is to be carried out. Then, the command code is followed for setting the UserInfo in CNC mode "U". The parameter a plain text up to 32 characters (ASCII) is transmitted. Carriage Return acts as end of command. If the the initialization values are saved with the command „@0IW“ to flash, this info is also available after power off or reset again. The query of this UserInfo can be done with the command "@ 0D".

Restrictions:
The length of the user info is limited to 32 characters.

5 iMC-xx error messages

The controller provides a response following the transmission of each command. These codes are issued as ASCII characters in order to make interpretation easy. The characters issued can be used to identify error sources and causes. The individual error codes are described below.

Code	Description
0	Handshake character - No error, the command was executed correctly. - The next command can be transmitted.
1	Error in numeric value provided - The controller received a numeric value that could not be interpreted correctly. – The numeric value provided is beyond the permitted range or the numeric value provided contains invalid characters.
2	End switch error - An end switch has been activated during the movement procedure. The current movement has been interrupted and stopped without a braking ramp. The actual position of the controller is thus no longer correct, and step losses have probably occurred. - The reference movement of a step motor axis have been incorrectly executed or not executed at all. **NOTE:** The controller must be initialised again and a reference movement carried out following an end switch error.
3	Incorrect axis specification - The controller is provided with an axis specification for a command that is to be executed, but the axis specification contains an undefined axis. - For commands containing axis specifications , only use combinations of axes that have also been initialised.
4	No axis defined - Before the controller is provided with movements or general commands for which the number of parameters depends on the number of axes, the “set number of axes” command must be submitted in order to correctly set the axis parameters.
5	Syntax error - A command has been incorrectly submitted. - The command used does not exist or cannot be processed by this controller. - Check that all the commands submitted are correct.
6	End of memory - An attempt has been made to transmit more commands in CNC mode than can be stored in the controller.
7	incorrect number of parameters - The controller has received too few or too many parameters for the particular command. - Check to see if the number of parameters for the command is correct with regard to the number of axes.
8	Command to be stored is incorrect - The controller has received a command that is not available as a CNC command.
9	System error

	- The system's power supply has not yet been turned on. - The system's safety circuit has not yet been activated. - The output stages and/or the safety circuit could not be turned on, as the cover is still open. - The output stages are overheated. - An emergency shutdown situation has occurred. **NOTE:** The controller must be initialised again and a reference movement carried out following an emergency shutdown situation.
A	Not used by this controller
B	Not used by this controller
C	Not used by this controller
D	Speed not permitted - Non-compliance with the permitted thresholds for speed values. - Check that all speed values are correct.
E	Not used by this controller
F	User stop - The user has pressed the controller's stop button and the current movement has therefore been stopped. The command can continue to be executed by pressing the start button or with the "@0s" start command.
G	Invalid data field - The controller has received a start order, but no movement remains in the memory to be initiated i.e. no stop function was previously sent and executed. - An attempt has been made to transmit a CNC programme although there is still a programme or part of a programme in the memory.
H	Cover error - An attempt has been made to execute a command that is not permitted when the cover is open.
R	Reference error - A movement cannot be executed, because it is first necessary to carry out a reference movement.
=	Not used by this controller