isel - IMC4-M

Programming the IMC4-M stepper control card

isel-automation GmbH & Co.KG ● D-36466 Dermbach ● Untere Röde 2 ● ☎ (036964)844 ● 🖷 (036964)84510

The information, technical data and dimensions contained in this publication comply with the latest state of the art (9/2005). Nevertheless, any printing errors and mistakes cannot be ruled out.

Suggestions for improvement and indications to errors are always welcomed.

The software and hardware designations of the corresponding companies, which are used in our publications, are generally either patented or protected by trademarks.

Table of Contents

1. Introduction 5
1.1. Preface 5
1.2. Connecting the control system to a PC 6
1.3. Conversion Factors 7
1.4. Coordinate System and Reference Points 8
1.5. Relative and Absolute Coordinate Specifications 9

2. DNC Mode and its Commands 10
2.1. DNC Command Structure 10
2.2. IMC4-M commands in DNC-mode 11
2.2.1. Initialisation; Setting the Number of Axes 11
2.2.2. Reference Point Approach 12
2.2.3. Defining the Referencing Velocities 13
2.2.4. Executing a Relative Movement 14
2.2.5. Executing an Absolute Movement 15
2.2.6. Interrogating the Actual Position 16
2.2.7. Setting Zero 17
2.2.8. Enabling/Disabling the 3D Interpolation 18
2.2.9. Plane Selection for Circular Interpolation 19
2.2.10. Setting the Circle Direction for the Circular Interpolation 20
2.2.11. Circular Interpolation 21
2.2.12. Starting a Stopped Movement 22
2.2.13. Reading Ports 23
2.2.14. Writing Ports 24
2.2.15. Turning On/Off Test Mode 25
2.2.16. Polling the Status Data of the Control System 26
2.2.17. Requesting the Version Data of the Control System 27
2.2.18. Initialising Parameters 28
2.2.19. Diagnosis 29
2.2.20. Carrying Out a Self-test 31
2.2.21. Check and Control Codes 32
2.2.22. Deleting a stored CNC-programme 33
2.2.23. Move until portevent 34
2.3. Calculating the Circle Parameters 35
2.3.1. The Parameters for the Circular Interpolation 35
2.3.2. Calculating the Arc Length 35
2.3.3. Calculating the Interpolation Parameter 36
2.3.4. Arc Starting Point 36
2.3.5. Directions in the Starting Point of the Circular Interpolation 36
2.3.6. Calculation Example for Circular Interpolation 37

3. CNC Mode and Its Commands 38
3.1. CNC Command Structure 38
3.2. The Commands of the IMC4-M in CNC Mode 39
3.2.1. Saving the CNC Data Field 39
3.2.2. Reference Point Approach in CNC Mode 40
3.2.3. Relative Movement in CNC Mode 41
3.2.4. Absolute Movement in CNC Mode 42
3.2.5. Setting Zero in CNC Mode 43
3.2.6. Enabling/Disabling 3D Interpolation in CNC Mode 44
3.2.7. Plane Selection for Circular Interpolation in CNC Mode 45
3.2.8. Setting the Circle Direction for the Circular Interpolation in CNC Mode 46
3.2.9. Circular Interpolation in CNC Mode 47
3.2.10. Loops and Branches in CNC Mode 48
3.2.11. Time Delays in CNC Mode 49

3.2.12. Setting the Port in CNC Mode 50
3.2.13. Reading a Port and Branching in CNC Mode 51
3.2.14. End of Data Field in CNC Mode 52

4. Error Messages of the IMC4-M 53

5. Default port specifications for IMC4-M 55

6. Pin assignment of IMC4-M 56

1. Introduction

1.1. Preface

Our company *iselautomation* has been known for drive systems and computerised numerical control systems (CNC) with stepper motors for many years. Stepper motors in the lower and medium performance ranges have generally gained acceptance all over the world. We can virtually not imagine state-of-the-art automation technology without drive systems in the medium accuracy and dynamic response range. As a rule, stepper motor control systems do not use closed-loop control circuits so that expensive sensor and electronic evaluation systems are not necessary. In addition to a simple design and start-up, this also provides a good price-performance ratio.
We have been offering control systems based on IMC4 control system for many years; they are used in many areas of production, automation, research and further education. In continuation of this product line, the IMC4-M control system has been developed. The IMC4-M is a micro-processor based control system designed to control up to 4 axes using step direction signals. For data transfer, the IMC4-M is coupled via a serial interface, as forerunner IMC4 too. To maintain a certain compatibility, the commands to control the control system are used similarly to the CNC and DNC modes of the IMC4 control system. This description will give you an overview of the CNC and DNC commands implemented in the IMC4-M control.

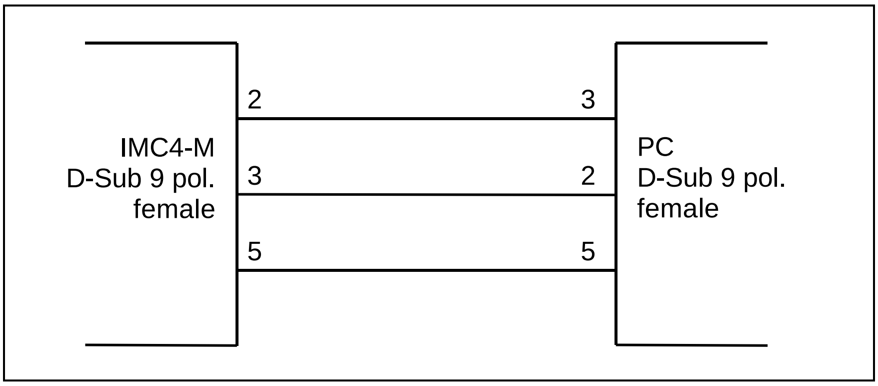
1.2. Connecting the control system to a PC

For data transfer between IMC4-M and control computer, a serial interface to RS232 is used. The connection is provided via a 3-wire line.

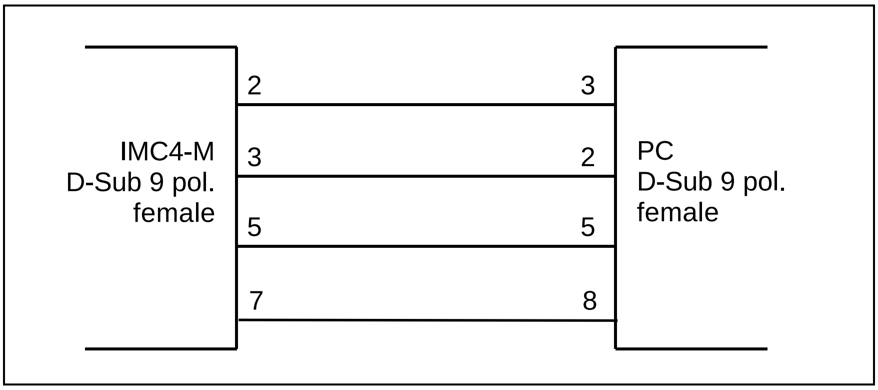
The following parameters are defined on the IMC4 as the data transfer parameters:

baudrate:	19200
data-bits:	8
stop-bist:	1
parity:	none

A special 3-wire connection is used as the connection line to the PC:



In conjunction with isel-switch cabinets and isel-terminals the following connection is used:



The additional wire is used to control the status of operators panel start-key directly from isel-operators software (ProNC, Remote).

1.3. Conversion Factors

The distances to be traversed are transferred to the control system in steps. In contrast to the transfer of metric dimensions [mm], this kind of representation allows faster decoding and execution, since the control system represents all positions internally as steps. The conversion of the distances to be traversed has to be carried out by the control computer. For this conversion, the following information about the connected mechanical system is required:

spindle lead:
- lead of the spindle installed in the mechanical system
- specifies the distance traversed by the slide (at a revolution of the spindle)
- leads of 2.5, 4, 5 and 10mm are typical

steps/revolution:
- number of increments per motor revolution

gear ratio:
- is only needed when geared stepper motor drives are used

The conversion from mm into steps can be carried out using the following formula:

$$steps = \frac{dist.\ to\ be\ traversed\,(mm)}{spindle\ lead\,(mm)} * steps / revolution * gear\ ratio$$

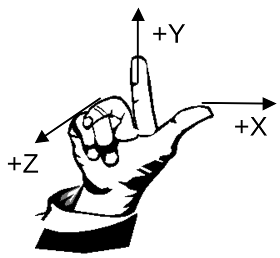
For velocities, the control system expects a specification in steps (increments) per second. To convert the velocity specifications usually used for mechanical systems (mm/s or m/min), the following formulas can be used:

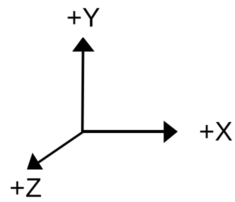
$$V(steps/s) = \frac{V(mm/s) * steps/revolution * gear\ ratio}{spindle\ lead\,(mm)}$$

$$V(steps/s) = \frac{V(mm/min) * steps/revolution * gear\ ratio}{spindle\ lead\,(mm)} * \frac{60}{100}$$

1.4. Coordinate System and Reference Points

The definition of the coordinate systems and the reference points is an essential prerequisite for programming machining movements within the working area of machine tools. Acc. to DIN 66217, a right-handed, right-angled coordinate system with the axes X, Y and Z is used. The Z axis is identical with the axis of the machining spindle. The positive direction of the Z axis extends from the workpiece towards the tool.





To create the machining movements, either the tool or the workpiece can be moved. But irrespective of that, the coordinate system will always refer to the workpiece. It is thus not essential for the programming whether the tool or the workpiece is moved. The programmer will always assume that the tool moves relatively to the workpiece not moving.

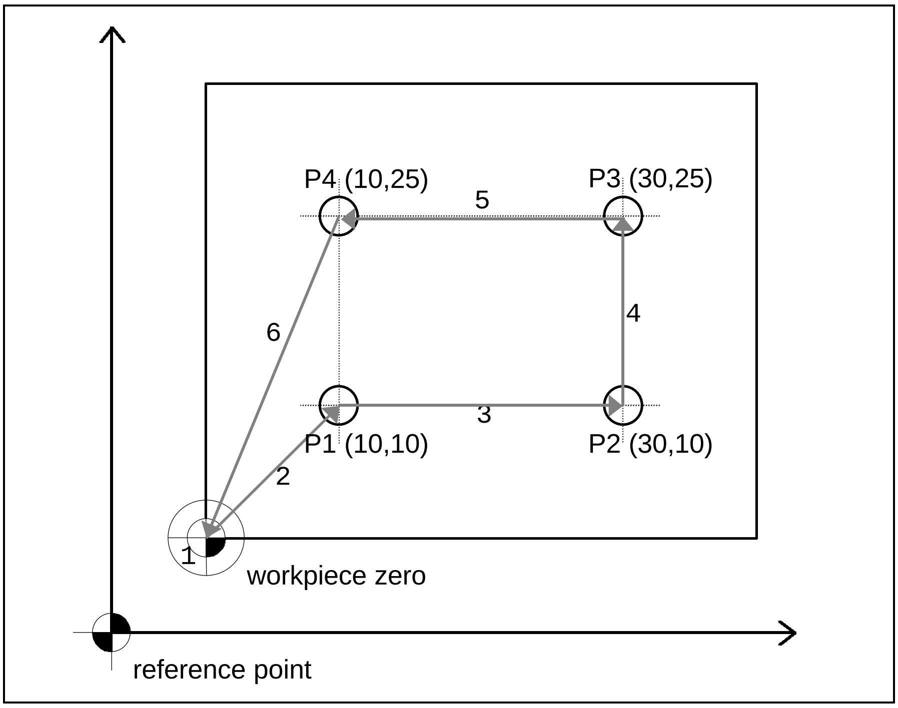
In addition to the coordinate system, reference points also play an important part in the programming of machine tools. The most important reference points will be mentioned here:

machine zero:	- is fixed in the origin of the plant coordinate system - is given by the plant design and cannot be changed
reference point:	- point which is defined by limit switches and can be located by them; is in most cases situated in an outer corner of the working area and often identical with machine zero - can also be set to a fixed distance to the machine zero and will then remain unchanged - reference point approach is in most cases possible using an accuracy of one path increment
workpiece zero:	- origin of the workpiece coordinate system - can be freely selected by the programmer and is used as the reference point for the programming

For particulars and specifications regarding your plant, please refer to the relevant Operator's Guides and/or hardware and mechanical manuals.

1.5. Relative and Absolute Coordinate Specifications

Distances to be traversed can be programmed using either relative (incremental dimensions) or absolute coordinates. When specifying relative coordinates, the reference point will be the current tool position. The coordinates will then constitute the distance to the point of the current position to be approached next. In contrast, when specifying absolute coordinates, the workpiece zero point will be used as the reference point. The coordinates will then constitute the distance from the point to be next approached from workpiece zero. This is illustrated by the example below:



4 drill holes are to be made in a workpiece at the points P1, P2, P3 and P4. The points are to be approached in the order P1 --> P2 --> P3 --> P4.

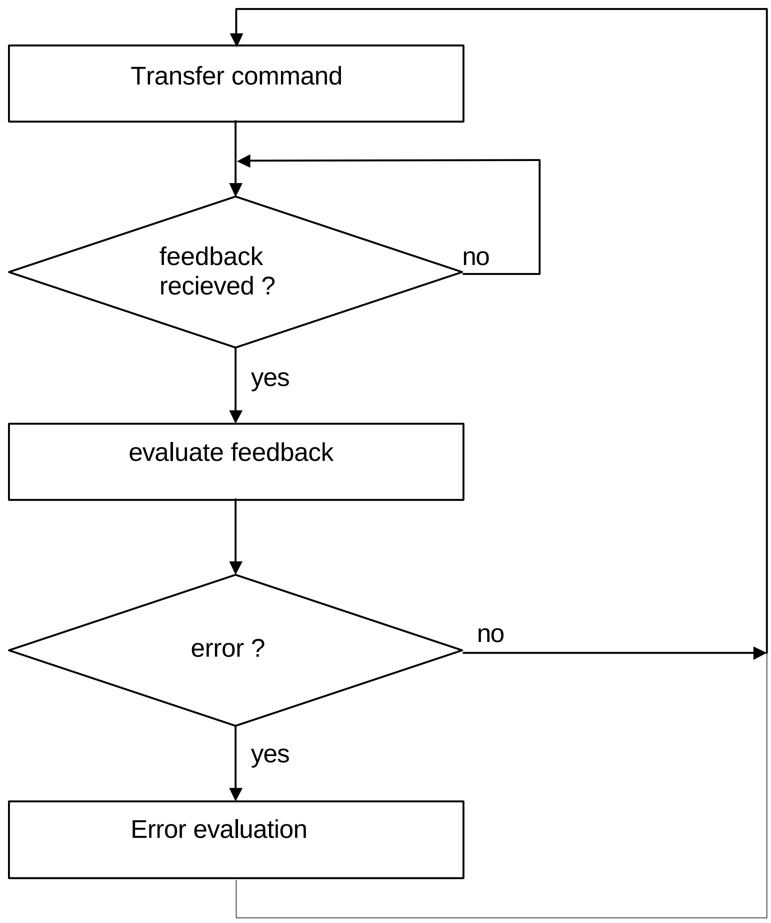
For machining, the following relative and absolute coordinates will be used:

	absolute	relative
1	0, 0	0, 0
2	10, 10	10, 10
3	30, 10	20, 0
4	30, 25	0, 15
5	10, 25	-20, 0
6	0, 0	-10, -25

2. DNC Mode and its Commands

2.1. DNC Command Structure

In DNC mode, the data records and commands transferred by a control computer are evaluated and executed directly. To this aim, a so-called initialisation is required in the beginning of the data communication. This initialisation consists of the data opening character @, the device number (0 = default) and the number of the axes to be traversed. After the initialisation, the program steps are transferred to the control system individually and are directly executed by the control system. To check the data transfer or to signal any errors during the data transfer, appropriate ASCII characters are transmitted to the control computer as a feedback. This so-called software handshake procedure is realised as follows:



First, a command is transferred to the control system. The control system will decode and execute the command and will then generate either an appropriate acknowledgement or an error character. This feedback is evaluated by the control computer. If an error has occurred, an error evaluation and elimination must be carried out. Then the next command can be transferred to the control system in the same way.

The scope of commands available in DNC mode of the IMC4 control system is described in the following.

2.2. IMC4-M commands in DNC-mode

2.2.1. Initialisation; Setting the Number of Axes

command: Set number of axes

application: The control system is re-initialised by transferring the number of axes.

structure: @<GN><axes><CR>

@	= data opening character
<GN>	= device number, default = 0
<axes>	= axis specification, see below
<CR>	= Carriage Return to compete the command

notation: @07, @08

explanation: The control system is addressed using @0; the numerical value following contains the axis configuration. Each axis is internally represented by a bit of a binary value, resulting in the following values:

1 --> X axis
3 --> X+Y axes
7 --> X+Y+Z axes
8 --> A axis

restriction: The combinations @00, @02, @04, @06 and @09 are not permitted.
CAUTION: The A axis must always be initialised separately.

2.2.2. Reference Point Approach

command: Reference point approach

application: The control system will traverse all specified axes to their zero points (reference points). With *isel* systems, the reference points are always defined in a reasonable order, but can also be adapted accordingly using appropriate initialisation commands.

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

@	= data opening character
<GN>	= device number, default = 0
R or r	= reference point approach command
<axes>	= axis specification, see below
<CR>	= Carriage Return to compete the command

notation: @0R7, @0r7, @0R8

explanation: The control system is addressed using @0. "R" specifies that referencing is required. The following numerical value defines the axes to be referenced. Each axis is internally represented by a bit of the binary value, resulting 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 order of execution is defined as follows:
Z axis --> Y axis --> X axis --> A axis
After the reference point approach has been carried out, the control system will send its acknowledgement character and will wait for the commands to come. The control system will only be able again to execute commands after the reference point approach has been carried out by the connected mechanical system.

restriction: This command can only be used after the control system has been initialised using the command “Set number of axes” and is limited to the axis configuration specified there. If the axes are specified not correctly, error message "3" will appear. If the control system is operated in 3D mode, the command will switch back to 2.5-dimensional mode.
CAUTION: The A axis must always be referenced separately.
CAUTION: If the reference point switch is not connected, the appropriate axis is selected continuously. It is, however, possible to simulate the reference point switch by pressing the STOP button.

2.2.3. Defining the Referencing Velocities

command: Set referencing velocity

application: This command defines the reference point approach velocity (further referred to as referencing velocity) for each axis separately.

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, default = 0
d	= command “Set referencing velocity”
<Gx>	= referencing velocity x
<Gy>	= referencing velocity y
<Gz>	= referencing velocity z
<Ga>	= referencing velocity a
<CR>	= Carriage Return to complete the command

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

explanation: If no information regarding the referencing velocity is transferred by the control system, a default value will be used. Any values that have been changed will not be stored when the control system is turned off.

restriction: The specified velocities must be within the valid range of values for the velocities. A referencing velocity chosen too high, in conjunction with a spindle lead also chosen too high, may damage the reference point switches due to the existing mass inertia. The control system needs a switching hysteresis from the connected zero-position switch. This should be observed when connecting electronic zero sensors.

2.2.4. Executing a Relative Movement

command: Relative movement

application: The control system will generate a relative movement depending on the transferred number of steps and on the transferred stepping velocity. The traversing movement is carried out 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, default = 0
A or a	= command “Relative Movement”
<Sx>	= number of steps x
<Gx>	= velocity
<Sy>	= number of steps y
<Gy>	= velocity
<Sz>, <Sz1>	= number of steps z
<Sz2>	= number of steps z, 2nd movement with 2.5D, 3 axes
<Gz>, <Gz1>	= velocity
<Gz2>	= velocity z, 2nd movement with 2.5D, 3 axes
<Sa>	= number of steps a, with 4 axes
<Ga>	= velocity, with 4 axes
<CR>	= Carriage Return to complete the command

notation:

@0A 5000,900	(x axis only)
@0A 50,900,20,9000	(x and y axes)
@0A 30,800,10,900,4,90,-4,30	(x, y and z axes, with 3 axes)
@0A 30,800,10,900,4,90,-4,30	(x, y, z and a axes, with 4 axes)

explanation: The control system is addressed using @0; "A" or "a" specify that a relative movement is to be carried out. The control system expects a number of pairs consisting of the number of steps and the velocity for each axis. The movement is carried out using relative dimensions, i.e. referred to the last position. The number of axes must match with the number of axes, i.e. one parameter pair for x operation, two parameter pairs for xy operation, four parameter pairs for xyz operation and four parameter pairs for xyza operation. The individual numbers must be separated by commas. For the z axis, two pairs of numbers are expected when 3 axes are used in 2.5D mode, since for machining applications the situation "Traverse, lower tool and then lift" often occurs. In 2.5D interpolation mode, first the movements of the x and y axes are carried out (interpolated linearly), then the z axis is traversed first by the values specified in z1 and then by the values specified in z2. If only one axis is required to be moved, the values for all axes initialised must nevertheless be transferred. When doing so, "0" must be specified as the number of steps for the axes not moved. After execution of this command, the control system will send the handshake character ("0") as the feedback. The control system will only be able again to execute new commands after this command has been executed.

restriction: This command can only be used after the number of axes has been set. Apart from this, the control system will not check whether the movement goes beyond the admissible range of the connected mechanical system.

In 2.5D interpolation mode, the velocity specification of the axis with the longest way is accepted as the path velocity, and the velocity of the remaining axes is adapted accordingly depending on the distance ratio. In contrast to that, in 3D interpolation mode, the velocity specification of the x axis is used as the setting value for the path velocity.

2.2.5. Executing an Absolute Movement

command: Move to absolute position

application: The control system will approach the specified position at the velocities specified. The traversing movement will be carried out 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, default = 0
M	= command "Absolute Movement"
<Sx>	= position x
<Gx>	= velocity
<Sy>	= position y
<Gy>	= velocity
<Sz>, <Sz1>	= position z
<Sz2>	= position z, 2nd movement always = 0
<Gz>, <Gz1>	= velocity
<Gz2>	= velocity
<Sa>	= position a, with 4 axes
<Ga>	= velocity, with 4 axes
<CR>	= Carriage Return to complete the command

notation:

@0M 5000,900	(x axis only)
@0M 50,900,20,9000	(x and y axes)
@0M 30,800,10,900,4,90,0,30	(x,y and z axes, with 3 axes)
@0M 30,800,10,900,4,90,4,30	(x,y,z and a axes, with 4 axes)

explanation: The control system is addressed using @0. "M" specifies that an absolute position will follow. For reasons of compatibility with the relative position command, two pairs of numbers are also here expected for the z axis when 3 axes are used. The second position specification of the z position must then, however, be zero and will be ignored. After the command has been carried out, the control system will response with the handshake character. The control system will only be able again to execute a new command if this command has been carried out.

Restriction: This command can only be used if the number of axes has been set. The control system will not check whether the movement goes beyond the range of the connected mechanical system.

2.2.6. Interrogating the Actual Position

command: Position interrogation

application: The control system will return the current actual position of all axes to the higher-level computer.

structure: @<GN>P<CR>

@	= data opening character
<GN>	= device number, default = 0
P	= position inquiry command
<CR>	= Carriage Return to complete the command

notation: @0P

explanation: The control system is addressed using @0. "P" specifies that a position interrogation is carried out. The control system confirms this with the handshake character and then outputs the position values of all axes in the hexadecimal format (up to 3 axes: 18 hexadecimal digits, with 4 axes: 24 hexadecimal digits).

The structure of the returned position is as follows:

e.g.: 00010002000FFFFFFE for 3 axes

Position x = 000100, hexadec. in the complement on 2, corresponds to 256 dec.
Position y = 02000F, hexadec. in the complement on 2, corresponds to 131087 dec.
Position z = FFFFFE, hexadec. in the complement on 2, corresponds to -2 dec.

e.g.: 000B00044000000FFE003040 for 4 axes

Position x = 000B00, hexadec. in the complement on 2, corresponds to 2816 dec.
Position y = 044000, hexadec. in the complement on 2, corresponds to 278528 dec.
Position z = 000FFE, hexadec. in the complement on 2, corresponds to 4094 dec.
Position a = 003040, hexadec. in the complement on 2, corresponds to 12352 dec.

restriction: This command can only be used if no traversing command takes place (if the plant is at a standstill). The control system cannot check whether the actual position corresponds to the current position of the connected mechanical system, since no control loop exists.

CAUTION: With a maximum of 3 axes, in all cases the positions for three axes are returned by the function, irrespective of the number of axes defined. In the case of four axes, in all cases positions for 4 axes are returned.

2.2.7. Setting Zero

command: Set zero at the current position

application: The control system will save the current position as the virtual zero point for the specified axis/axes. The next Absolute Movement commands will use this virtual zero point as the new reference point.

structure: @<GN>n<axes><CR>

@ = data opening character
<GN> = device number, default = 0
n = command "Set Zero"
<axes> = axis specification, see below
<CR> = Carriage Return to complete the command

notation: @0n7, @0n1, @0n8

explanation: The control system is addressed using @0. "n" specifies that a zero offset is to be carried out. After the command, the control system are reported the axes for which a zero offset is to be carried out. Each axis is internally represented by a bit of a binary value, resulting in the following values:

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

After the command has been executed, the control system will send an appropriate feedback.

restriction: The virtual zero point is only important for the Absolute Movement command. The virtual zero point has no influence on the relative positioning, since a relative traversing vector is specified here.
CAUTION: The zero offset for the A axis must always be carried out separately.

2.2.8. Enabling/Disabling the 3D Interpolation

command: Enable/disable 3D linear interpolation

application: The control system extends the 2.5D interpolation of the standard operating system to 3-dimensional interpolation. This command can be used to enable/disable this interpolation depending on the particular task.

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

@	= data opening character
<GN>	= device number, default = 0
z	= command "3D Interpolation"
<status>	= 0 --> disable, 1 --> enable
<CR>	= Carriage Return to complete the command

notation: @0z1, @0z0

explanation: The control system is prepared for the new command using the data opening character @0. "z1" will change the interpolation from 2D to 3D operation. This statement is modal, i.e. all relative and absolute movements are carried out threedimensionally. The specification of z2 parameters with 3 axes are ignored in the case of these traversing movements. The velocity specification of the interpolation must be carried out with the x specification. In the case of 4 axes, the 4th axis is traced.

2.2.9. Plane Selection for Circular Interpolation

command: Plane selection

application: Setting of the interpolation plane for the circular interpolation. Circles are only defined in one plane. The default plane for the circular interpolation is the XY plane. However, it is also possible to define any other plane configuration as the circle plane using the Plane Selection command.

structure: @<GN>e<plane><CR>

@	= data opening character
<GN>	= device number, default = 0
e	= command "Set circle plane"
<plane>	= plane specification, see below
<CR>	= Carriage Return to complete the command

notation: @0e1, @0e0

explanation: The control system is addressed using @0. "e" specifies that the plane for circular interpolation is to be set. The following numerical value defines the plane as follows:

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

restriction: This command has modal effect, i.e. a plane selection for the circular interpolation remains stored until it is overwritten by a new plane selection.

2.2.10. Setting the Circle Direction for the Circular Interpolation

command: Set circle direction

application: Setting of the circle direction for the circular interpolation. The circular interpolation is initiated by two successive commands. The first command defines the direction of the circle, and the second command (see 2.2.11.) transfers the interpolation parameters.

structure: @<GN>f<direction><CR>

@	= data opening character
<GN>	= device number, default = 0
f	= command "Set "Circle Direction"
<direction>	= 0 --> CW (clockwise), -1 --> CCW (counter-clockwise)
<CR>	= Carriage Return to complete the command

notation: @0f-1, @0f0

explanation: The control system is addressed using @0. "f" specifies that the direction for the circular interpolation is to be set. The following numerical value defines the direction as follows:

0 --> CW (circular interpolation arc CW)
-1 --> CCW (circular interpolation arc CCW)

restriction: The direction for the circular interpolation must always be programmed before any circular movement is programmed.

2.2.11. Circular Interpolation

command: Circular interpolation

application: Processing of circles and arcs at constant path velocity. The circular interpolation is initiated by two successive commands. The first command defines the circle direction (see 2.2.10.), and the second command transfers the interpolation parameters.

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

@	= data opening character
<GN>	= device number, default = 0
<B>	= arc length in steps
<V>	= velocity
<D>	= interpolation parameters
<Xs>	= starting point x
<Ys>	= starting point y
<Rx>	= direction x
<Ry>	= direction y
<CR>	= Carriage Return to complete the command

ntation: 0y400,1500,119,-141,141,-1,-1

eplanation: The control system is addressed using @0. "y" specifies that a circular interpolation is to be carried out. The arc length specifies the length of the arc between starting and end points of the circular interpolation in steps. For the velocity, all integer values within the valid range of values for velocities are permitted. The interpolation parameter has to be transferred, because the control system cannot calculate this parameter by itself due to its memory capacity. The parameters Xs and Ys specify the starting point of the arc relatively to the circle centre point. Rx and Ry specify in which quadrant of the circle the interpolation starts. After the command has been executed, the control system will response with the handshake character ("0") as the feedback. The control system will only be able again to execute new commands if this command has been carried out.

CAUTION: To calculate the parameters, please refer to the Section "Calculating the Parameters for Circular Interpolation".

Restriction: This command can only be used if the number of axes has been set. Apart from that, the control system will not check whether the movement goes beyond the range of the connected mechanical system.

2.2.12. Starting a Stopped Movement

command: Start

application: A stopped movement will be continued.

structure: @<GN>S<CR>

@	= data opening character
<GN>	= device number, default = 0
S or s	= Start command
<CR>	= Carriage Return to complete the command

explanation: The control system is addressed using @0. "S" specifies that a stopped movement is to be restarted, completing the execution of the movement remaining. After the command has been executed, the control system will either response with the handshake character ("0") as the feedback or output an error message if no movement to be traversed is left in the memory. The control system will only be able again to execute new commands after the remaining movement has been traversed.

restriction: This command makes only sense if a movement has been stopped and the control system is in Stop mode. In this mode, the current status of the START button can be interrogated using the Diagnosis command (see 2.2.19 Diagnosis), and it is also possible to create a Start command by the operating program when the START button is pressed. In DNC mode, the operation using the keys of the control system is only possible in this way, since the control system does not directly react to the START button.

2.2.13. Reading Ports

command: Read port

application: This command can be used to determine the current status of logical or physical input ports via the serial interface.

structure: @<GN>b<port No.><CR>

@	= data opening character
<GN>	= device number, default = 0
b	= command "Read Port"
<port No.>	= port number, see below
<CR>	= Carriage Return to complete the command

notation: @0b0, @0b1

explanation: The control system is addressed using @0. "b" specifies that the status of an input port is to be determined. Then the port number is transferred and the command is completed with Carriage Return. The control system will response with the software handshake "0" followed by two characters that specify a hexadecimal value corresponding to the current status of the input port. For the IMC4-M control system, the following ports are defined with the following functionalities:

Port	Status	Function	
0	00 - FF	User I/O (also still possible with 65531)	
		Bit0	Input Stop button
		Bit1	Input Start button
		Bit2	User Input 1
		Bit3	Input Power OK
		Bit4	Input Emergency Stop switch
		Bit5	Input ON button
		Bit6	Input Cover switch
		Bit7	Input Keyswitch
1	00	Cover is open	
	01	Cover is closed	
2	00	Spindle is switched off	
	01	Spindle is switched on	
3	00	Motor currents are switched off	
	01	Motor currents are switched on	
5	00 - FF	User I/O I2C-port extention (I2C-address=0)	
6	00 - 1F	User I/O	
		Bit0	User Input 1
		Bit1	User Input 2
		Bit2	User Input 3
		Bit3	User Input 4
		Bit4	User Input 5

Restriction: The port states are only returned if the control system responses with the software handshake "0". The command can only be used if no movement is executed.

2.2.14. Writing Ports

command: Write port

application: This command can be used to write defined values to logical or physical output ports via the serial interface.

structure: @<GN>B<port No.>,<value><CR>

@	= data opening character
<GN>	= device number, default = 0
B	= command "Write Port"
<port No.>	= port number, see below
<Value>	= new port value
<CR>	= Carriage Return to complete the command

notation: @0B1,1

explanation: The control system is addressed using @0. "B" specifies that the value of an output port is to be set. The port number and the new port value are then transferred separated by a comma and the command is completed with Carriage Return. The control system responses with the software handshake "0" if the execution has been successful or with an error message if wrong port numbers and/or wrong values have been transferred. For the IMC4-M control system, the following ports are defined with the following functionalities:

Port	Value	Function	
0	0 - 255	User I/O (also still possible with 65529)	
		Bit0	reserved
		Bit1	reserved
		Bit2	reserved (relay, spindle)
		Bit3	User Output 1
		Bit4	User Output 2
		Bit5	Output Brake
		Bit6	User Output 3
		Bit7	reserved
1	0	Cover may not be opened	
	1	Cover may be opened	
2	0	Spindle OFF	
	1	Spindle ON	
3	0	Motor currents OFF	
	1	Motor currents ON	
4	0 - 255	analog output 0 – 10V	
5	0 - 255	User I/O I2C-port extention (I2C-address=0)	
6	0 - 255	User E/A	
		Bit4	User Output 5
		Bit5	User Output 6
		Bit6	User Output 7
		Bit7	User Output 8

Restriction: The port values will only be overwritten if the control system responses with software handshake "0". This command can only be used if no movement is carried out. The reserved bits of port 0 cannot be overwritten by the user.

2.2.15. Turning On/Off Test Mode

command: Test mode ON/OFF

application: This command can be used to turn on/turn off the test mode depending on the particular task.

structure: @<GN>T<status><CR>

@	= data opening character
<GN>	= device number, default = 0
T	= command "Test Mode ON/OFF"
<status>	= 0 --> OFF, 1 --> ON
<CR>	= Carriage Return to complete the command

notation: @0T1, @0T0

explanation: The control system is prepared for a new command using the data opening character @0. "T1" is used to turn on the test mode, and "T0" is used to turn it off. After the command has been executed, the control system will response with the handshake character ("0"). In test mode, the control system will treat the reference point approach and the limit switches other than in normal mode. If a reference point approach command is received in normal mode, the control system will not carry out a reference point approach in the actual sense, but will set the current point as the reference point. The limit switches are continued to be monitored, but can be overtravelled. This is very useful if an axis stands on a limit switch after the plant has been turned on and must be cleared.

Restriction: This command can only be used if no movement is executed.

2.2.16. Polling the Status Data of the Control System

command: Poll status data

application: Polling important status data of the control system in order to represent the current status, as well as for error location and diagnosis.

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

@	= data opening character
<GN>	= device number, default = 0
H	= command "poll status of inputs and keys"
h	= command "poll status of endswitches"
<CR>	= Carriage Return to complete the command

notation: @0H, @0h

explanation: The control system is prepared for a new command using the data opening character @0. "H" or "h" causes the control system to send back information about the current status in plaintext format. At the end of this information, the control system will response with the handshake character ("0"). The information is already output in ASCII format formatted line by line so that it can be displayed, e.g. in a terminal window directly on the screen of a control computer. This information includes the status of the limit switches and of the plant control elements.

restriction: To call this function, a sufficiently large receive buffer (min. 512bytes) must be provided in the control computer to make sure that no information is lost.

2.2.17. Requesting the Version Data of the Control System

command: Request version data

application: Requesting important version data of the control system.

structure: @<GN>V<CR>

@	= data opening character
<GN>	= device number, default = 0
V	= command "Interrogate Version Data"
<CR>	= Carriage Return to complete the command

notation: @0V

explanation: The control system is prepared for a new command using the data opening character @0. "V" causes the control system to send back information about the version of the control system in plaintext format. At the end of this information, the control system will response with the handshake character ("0"). The information is already output in ASCII format formatted line by line so that it can be displayed, e.g. in a terminal window directly on the screen of a control computer. This information includes the status of the limit switches and of the plant control elements.

restriction: To call this function, a sufficiently large receive buffer (min. 512bytes) must be provided in the control computer to make sure that no information is lost.

2.2.18. Initialising Parameters

command: Initialise Parameters

application: Initialising axis and referencing directions

structure: @<GN>I<code><value><CR>

@	= data opening character
<GN>	= device number, default = 0
I	= Initialisation command
<code>	= ASCII characters to differ different parameters, see below
<value>	= new value for the parameter
<CR>	= Carriage Return to complete the command

notation: @0ID3, @0IR1, @0IS1

explanation: The control system is prepared for a new command using the data opening character @0. "I" tells the control system that an initialisation is to be carried out. The initialisation is followed by an identifier for the parameter and the new value, as well as by Carriage Return as the command end. The following parameters can be initialised:

Code	Value	Function
D	1	Negate direction of X axis
	2	Negate direction of Y axis
	4	Negate direction of Z axis
	8	Negate direction of A axis
		(all combinations are possible by adding the values above)
R	1	Negate referencing direction of X axis
	2	Negate referencing direction of Y axis
	4	Negate referencing direction of Z axis
	8	Negate referencing direction of A axis
		(all combinations are possible by adding the values above)
L	1	X-axis is axis of rotation
	2	Y-axis is axis of rotation
	4	Z-axis is axis of rotation
	8	A-axis is axis of rotation
		(all combinations are possible by adding the values above)
S	0-255	Set internal hardware status byte
		Bit assignment: Bit0: Enable/disable cover opening
		Bit1-7: Not assigned

Restriction: The internal hardware status byte should not be used by the user and is only mentioned here for the sake of completeness. Its use is reserved for appropriate software from *iselautomation*.

2.2.19. Diagnosis

command: Diagnosis

application: Polling diagnosis data of the control system.

structure: @<GN>D<code1><code2><CR>

@ = data opening character
<GN> = device number, default = 0
D = Diagnosis command
<code1> = ASCII characters to differ different parameters, see below
<code2> = ASCII characters to differ different parameters, see below
<CR> = Carriage Return to complete the command

notation: @0DRp, @0DRn, @0DS0

explanation: The control system is prepared for a new command using the data opening character @0. "D" tells the control system that a diagnosis is to be carried out. The diagnosis is followed by two identifiers for the parameter and by Carriage Return as the command end. The control system will response with the software handshake "0" followed by two characters specifying a hexadecimal value that corresponds to the current value of the parameter. The following parameters can be polled:

Code1	Code2	Function
R	p	Poll positive limit switches
R	n	Poll negative limit switches
P	0	Poll input port
P	1	Poll endswitch port
P	2	Poll user input port
O	0	Poll output port
O	1	Poll user output port
S	0	Poll hardware status input port
H	0	Poll hardware status output port
S	L	Poll axis configuration

Bit assignment of the limit switches:

Bit0: X axis
Bit1: Y axis
Bit2: Z axis
Bit3: A axis

Bit assignment of the input port:

Bit0: User Input 1
Bit1: Input Stop button
Bit2: User Input 2
Bit3: Input PowerOK
Bit4: Input Emergency Stop button
Bit5: Inpt PowerOn button
Bit6: Input cover
Bit7: Input keyswitch

Bit assignment of the endswitch port:

Bit0: positive endswitch A-axis
Bit1: negative endswitch A-axis
Bit2: positive endswitch Z-axis
Bit3: negative endswitch Z-axis
Bit4: positive endswitch Y-axis
Bit5: negative endswitch Y-axis
Bit6: positive endswitch X-axis

Bit7: negative endswitch X-axis

Bit assignment of the user input port:

Bit0: User Input3
Bit1: User Input4
Bit2: User Input5
Bit3: Input Start button

Bit assignment of the output port

Bit0: enable power stage
Bit1: SoftwareOK
Bit2: spindle
Bit3: User Output 1
Bit4: User Output 2
Bit5: Output Brake
Bit6: User Output 3
Bit7: enable cover

Bit assignment of the user output port:

Bit4: User Output 5
Bit5: User Output 6
Bit6: User Output 7
Bit7: User Output 8

Bit assignment of the hardware status input port:

Bit0: Status of spindle
Bit1: Status of Start button
Bit2: Status of Stop button
Bit3: Status of Emergency Stop button
Bit4: Status of keyswitch
Bit5: Status of cover
Bit6: Not used
Bit7: Status of output stages/power supply

Bit assignment of the hardware status output port:

Bit0: enable/disable cover
Bit1-7: not used

Bit assignment of the axis configuration:

Bit0: X-axis 1=enabled, 0=disabled
Bit1: Y-axis 1=enabled, 0=disabled
Bit2: Z-axis 1=enabled, 0=disabled
Bit3: A-axis 1=enabled, 0=disabled
Bit4: X-axis 1=axis of rotation, 0=linear axis
Bit5: Y-axis 1=axis of rotation, 0=linear axis
Bit6: Z-axis 1=axis of rotation, 0=linear axis
Bit7: A-axis 1=axis of rotation, 0=linear axis

restriction: The diagnostic functions should not be used by the user and are only mentioned here for the sake of completeness. The use is reserved for appropriate software from *iselautomation*.

2.2.20. Carrying Out a Self-test

command: Self-test

application: This statement will initiate a self-test of the control system. This test includes the movement of the axes, as well as the interface test and the output of information about the version.

structure: @<GN>?<CR>

@	= data opening character
<GN>	= device number, default = 0
?	= Self-test command
<CR>	= Carriage Return to complete the command

notation: @0?

explanation: The control system is addressed using @0. "?" specifies that a self-test is to be carried out. The command is completed with Carriage Return. The control system will then output information about the version, will test the movement of the motors and carry out an interface test. To test the interface, the control system will first output the ASCII character set. If a character is received via the interface, the control system will change to Echo mode and then return all characters received to the control computer.

restriction: To call this function, a sufficiently large receive buffer must be provided in the control computer to make sure that no information is lost. Therefore, it only makes sense to call this function within a terminal program or a terminal function. The self-test can only be quitted by turning off the control system or by software reset (char(254)). The plant must then be re-initialised.

The self-test can also be initiated if the Start button is pressed when turning on the control system and is released only after the self-test has been started.

2.2.21. Check and Control Codes

Check and control codes provide direct access to the functional sequence of the control system via the serial interface. The commands sent are evaluated directly in the control system's receive routine and are then executed. Special control codes are provided for the following functionalities:

function: Software stop char(253)

A positioning movement in DNC mode (relative or absolute) can be stopped by a stop command without any step losses. A start command executed thereafter (by transferring @0S, see 2.2.12 "Starting a Stopped Movement") will complete the interrupted functional sequence. Furthermore, it is possible to read back the currently reached position after a stop command using the command "Interrogate Position". This functionality can also be achieved by pressing the Stop button. If a movement has been successfully stopped, the control system will create an additional feedback signal "F".

The function is called by transferring char(253) via the serial RS232 interface.

function: Software reset char(254)

The control system interrupts all activities immediately and carries out a software reset internally. The plant must then be re-initialised and referencing be carried out.

The function is called by transferring char(254) via the serial RS232 interface.

function: Software break char(255)

A positioning movement in DNC mode (relative or absolute) can be completed using a break command. This means that the remainder of the movement will be lost.

The function is called by transferring char(255) via the serial RS232 interface.

see also Chapter 4 "Error Messages of the IMC4-M"

2.2.22. Deleting a stored CNC-programme

command: Deleting a stored CNC-programme from FlashPROM

application: This statement will remove the stored CNC-programme from FlashPROM.

structure: @<GN>k<CR>

@	= data opening character
<GN>	= device number, default = 0
k	= delete stored CNC-programme
<CR>	= Carriage Return to complete the command

notation: @0k

explanation: The control system is addressed using @0. "k" specifies that the CNC-programme stored in the FlashPROM should be removed. The command is completed with Carriage Return.

2.2.23. Move until portevent

command: move until portevent

application: Use this command to specify a relative move without exact travel information. When doing so, a maximum distance to be traversed is specified, which is then cancelled by the given portevent.

structure: @<GN>Z<port No.>,<bit mask>,<value>,<G>,<Sx>,<Sy>,<Sz>,<Sa><CR>

@	= data opening character
<GN>	= device number, default = 0
Z	= command "Port move"
<port No.>	= port number
<bit mask>	= bit mask to mask selected bits
<value>	= comparison value
<G>	= velocity
<Sx>	= number of steps x
<Sy>	= number of steps x
<Sz>	= number of steps x
<Sa>	= number of steps x
<CR>	= Carriage Return to complete the command

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

explanation: The control system is addressed using @0; "Z" specify that a relative movement until portevent is to be carried out with the given velocity. The movement is carried out using relative dimensions, i.e. referred to the last position. The movement will be cancelled if the bit 2 on port 0 is read as 1 or if the given numbers of steps are executed. The control system will only be able again to execute new commands after this command has been executed.

CAUTION: The number of parameters must match with the number of axes, i.e. one parameter for x operation, two parameters for xy operation, four parameters for xyz operation and four parameters for xyza operation. The individual numbers must be separated by commas. Examples:

X-axis only:	@<GN>Z<P>,<M>,<V>,<G>,<Sx><CR>
XY-axes:	@<GN>Z<P>,<M>,<V>,<G>,<Sx>,<Sy><CR>
XYZ-axes:	@<GN>Z<P>,<M>,<V>,<G>,<Sx>,<Sy>,<Sz><CR>
XYZA-axes:	@<GN>Z<P>,<M>,<V>,<G>,<Sx>,<Sy>,<Sz>,<Sa><CR>

restriction: This command can only be used if the number of axes has been set. The control system will not check whether the movement goes beyond the range of the connected mechanical system.

2.3. Calculating the Circle Parameters

2.3.1. The Parameters for the Circular Interpolation

The circular interpolation is initiated by two successive commands (see 2.2.10, 2.2.11). The first command defines the circle direction, and the second command is used to transfer the interpolation parameters.

circle direction: @<GN>f<direction><CR>

@	= data opening character
<GN>	= device number, default = 0
f	= command "Set Circle Direction"
<direction>	= 0 --> CW (clockwise), -1 --> CCW (counter-clockwise)
<CR>	= Carriage Return to complete the command

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

@	= data opening character
<GN>	= device number, default = 0
<B>	= arc length in steps
<V>	= velocity
<D>	= interpolation parameter
<Xs>	= starting point x
<Ys>	= starting point y
<Rx>	= direction x
<Ry>	= direction y
<CR>	= Carriage Return to complete the command

An especially adapted differential algorithm acc. to Bresenham is used internally in the control system to create arcs. This kind of algorithms is very often used in micro-processor applications since high processing speeds are achieved here with a low calculation expenditure.
The meaning and how to calculate the parameters for circular interpolation will be explained in the following. An appropriate example is to be found in Section 2.3.6. of this Description.

2.3.2. Calculating the Arc Length

The arc length <B> specifies the length of the arc between the starting point and the end point of the arc in steps and is used internally in the control system as a running variable for the differential algorithm. The calculation of the arc length can be carried out in different ways and will be explained in the following.

- simple approximation formula

  For simple circle applications that contain, e.g. only quadrants, semi or full circles, the arc length can be calculated using the following formula:

  B – arc length in steps
  R – arc radius in steps
  A – starting angle in arc dimension
  E – end angle in arc dimension

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

  The result must be rounded to the next integer value. To eliminate the inaccuracy for processing on the control system, the next positioning movement should be programmed as an absolute movement.

- calculation using the software routine

  An exact calculation of the arc length can be achieved using simple software routines.

2.3.3. Calculating the Interpolation Parameter

The interpolation parameter is used by the control system as the starting value for the differential register of the algorithm used by the control system to create a circle. The calculation of the parameter is carried out on the PC side using appropriate software routines. This will take unnecessary burden of calculation from the control system, thus increasing the processing speed.

2.3.4. Arc Starting Point

The starting point of the arc constitutes the distance from the circle centre along X and Y in steps using relative coordinates (i.e. the circle centre point is assumed as an imaginary centre point for the calculation). The calculation can be carried out using the appropriate circle functions.

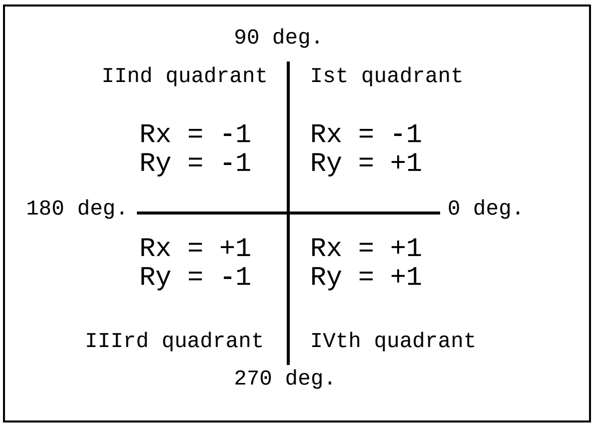
Xs - X coordinate of the starting point relative to the centre point
Ys - Y coordinate of the starting point relative to the centre point
R - radius in steps
A - starting angle as an arc

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

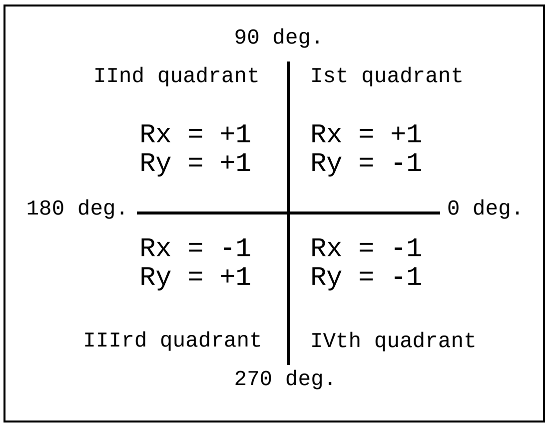
2.3.5. Directions in the Starting Point of the Circular Interpolation

To carry out the interpolation algorithm, the control system needs an information on in which quadrant the arc starts and which signs are to be used internally in the control system for certain calculations. This information is provided to the control in the form of the parameters Rx and Ry. This is carried out based on the following definitions:

ArcsCCW



Arcs CW



2.3.6. Calculation Example for Circular Interpolation

An example will be calculated in the following to illustrate how to calculate the circle command parameter.
An arc CCW with a radius of 200 steps is to be traversed at a velocity of 1,500 steps/s. The starting angle is 135 degrees, the end angle 225 degrees. Please note that all path specifications must be specified in steps and all angle specifications in arc dimension to carry out the calculation.

given:	Radius	R = 200	searched:	Arc length	B
	Starting angle	A = 135*Pi/180 = 2.3562		Starting point X	Xs
	End angle	E = 225*Pi/180 = 3.9267		Starting point Y	Ys
	Speed	V = 1500		Direction X	Rx
	Direction	CCW		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
<u>B = 400</u>

Starting point Xs and Ys (see 2.3.4):

Xs = R * cos(A) = 200 * cos(2.3562) = -141.4221
<u>Xs = -141</u>

Ys = R * sin(A) = 200 * sin(2.3562) = 141.4205
<u>Ys = 141</u>

Directions Rx and Ry (see 2.3.5):

Starting angle 135 degrees, direction of rotation CCW

<u>Rx = -1</u> <u>Ry = -1</u>

Interpolation parameter D (see 2.3.3.):

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

Total(R-1) = total(199) = 199*(199+1) = <u>39800</u>

Total(Xs+(Rx-Ry)/2) = total(-141+(-1-(-1))/2) = total(-141) = 141*(-141+1) = <u>-19740</u>

Total(Ys+(Rx+Ry)/2) = total(141+(-1-(-1))/2) = total(141) =141*(141+1) = <u>20022</u>

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

The appropriate commands would be:

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

3. CNC Mode and Its Commands

3.1. CNC Command Structure

Operated in CNC mode, the control system stores all transmitted commands in the internal data memory. For activation, the command "Store CNC data field" must be transferred after the standard initialisation. Then the data field is transferred and completed with the command "End of data field".
The program can now be enabled without any further communication with the control computer using an external Start command (pressing the Start button).
As a memory medium, FlashPROMs (non-volatile, electrically programmable and erasable memories) are used on the IMC4-M. Similarly to EPROMs, these memories are programmed in the system with the appropriate information by certain programming cycles. Erasing them corresponds to programming them with a default value. Before these chips are reprogrammed, they have generally to be deleted. To do so, the command "Deleting a stored CNC-programme" ("@0k") must be carried out.
After erasure, the new program can be written to the memories. If a program or part of the program has already been stored in the memories, the command "Store CNC data field" will result in an error message.

The commands of the IMC4-M control system will be listed below and be explained in brief. For a detailed explanation of some commands, please refer to the appropriate DNC mode command, since the meanings and number of the parameters often correspond to those in DNC mode.

If an error has occurred when transmitting and storing a CNC data field, the CNC program stored until then will be marked as invalid and can not be executed. In this case, the error in the program must be eliminated accordingly and the FlashPROM be erased before the data field can be transferred anew for saving.

3.2. The Commands of the IMC4-M in CNC Mode

3.2.1. Saving the CNC Data Field

command: Save CNC data field

application: This statement is intended to initialise the transfer of storable commands and must be programmed in the beginning of CNC mode.

structure: @<GN>i<CR>

@	= data opening character
<GN>	= device number, default = 0
i	= command "Save CNC data field"
<CR>	= Carriage Return to complete the command

notation: @0i

explanation: The control system is addressed using @0. "i" specifies that a CNC data field is to be stored. The command is completed with Carriage Return. Then, until the End-of-Data Field command comes or an error occurs, the control system will only accept CNC commands. The command is acknowledged with an appropriate feedback signal. All of the following storable commands are stored in the FlashPROM.

restriction: This command can only be used if the control system has been initialised and no movement is being carried out. If a program is already stored in the memories, an error message is output.

3.2.2. Reference Point Approach in CNC Mode

command: Reference point approach

application: The control system will save a movement of all axes towards their zero points (reference points). With *isel* systems, the reference points of the axes are always defined in a reasonable default order, but can be adapted accordingly using appropriate initialisation commands.

structure: 7<axes><CR>

7	= command code for reference point approach
<axes>	= axis specification, see below
<CR>	= Carriage Return to complete the command

notation: 77, 78

explanation: "7" specifies that referencing is to be carried out. The following numerical value defines the axes that will carry out a reference point approach. When doing so, each axis is internally represented by a bit of a binary value, resulting 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 order of execution is defined as follows:
Z axis --> Y axis --> X axis --> A axis

After the reference point approach has been carried out, the next CNC command is read from the memory and executed.

restriction: This command is limited to the axis configuration initialised. If wrong axes are specified, error message "3" will occur. If the control system is operated in 3D mode, this command will switch back to 2.5-dimensional mode.

CAUTION: The A axis must always be referenced separately.
CAUTION: If the reference point switch is not connected, the appropriate axis is selected continuously. It is, however, possible to cancel referencing by pressing the Stop key.

3.2.3. Relative Movement in CNC Mode

command:	Relative movement
application:	The control system will store a relative movement according to the transferred number of steps and the transferred stepping velocity.
structure:	0<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR> 0<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

0	= command code for relative movement
<Sx>	= number of steps x
<Gx>	= velocity
<Sy>	= number of steps y
<Gy>	= velocity
<Sz>, <Sz1>	= number of steps z
<Sz2>	= number of steps z, 2nd movement with 2.5D, 3 axes
<Gz>, <Gz1>	= velocity
<Gz2>	= velocity z, 2nd movement with 2.5D, 3 axes
<Sa>	= number of steps a, with 4 axes
<Ga>	= velocity, with 4 axes
<CR>	= Carriage Return to complete the command

notation:	05000,900	(x axis only)
	050,900,20,9000	(x and y axes)
	030,800,10,900,4,90,-4,30	(x,y and z axes, with 3 axes)
	030,800,10,900,4,90,-4,30	(x,y,z and a axis, with 4 axes)

explanation: "0" specifies that a relative movement is to be carried out. The control system will now expect a number of pairs for each axis, which consists of number of steps and velocity. The distances are specified using relative dimensions, i.e. with reference to the last position. The number of positions must match with the number of axes, i.e. one parameter pair in x mode, two parameter pairs in xy mode, four parameter pairs for xyz mode and four parameter pairs for xyza mode. The individual numbers must be separated by commas. For the z axis, two pairs of numbers are expected when 3 axes are used in 2.5D mode, since for machining applications the situation "Traverse, lower tool and then lift" often occurs. In 2.5D interpolation mode, first the movements of the x and y axes are carried out (interpolated linearly), then the z axis is traversed first by the values given in z1 and then by the values specified in z2. If only one axis is required to be moved, the values for all axes initialised must nevertheless be transferred. When doing so, "0" must be specified as the number of steps for the axes not moved. After execution of this command, the control system will send the handshake character ("0") as the feedback.

restriction: The control system will not check whether the movement goes beyond the admissible range of the connected mechanical system.
In 2.5D interpolation mode, the velocity specification of the axis with the longest way is accepted as the path velocity, and the velocity of the remaining axes is adapted accordingly depending on the distance ratio. In contrast to that, in 3D interpolation mode, the velocity specification of the x axis is used as the setting value for the path velocity.

3.2.4. Absolute Movement in CNC Mode

command: Move to absolute position

application: The control system will store an absolute movement according to the specified velocities and positions.

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

m	= command code for absolute movement
<Sx>	= position x
<Gx>	= velocity
<Sy>	= position y
<Gy>	= velocity
<Sz>, <Sz1>	= position z
<Sz2>	= position z, 2nd movement always = 0
<Gz>, <Gz1>	= velocity
<Gz2>	= velocity
<Sa>	= position a, with 4 axes
<Ga>	= velocity, with 4 axes
<CR>	= Carriage Return to complete the command

notation:

m5000,900	(x axis only)
m50,900,20,9000	(x and y axes)
m30,800,10,900,4,90,0,30	(x,y and z axis, with 3 axes)
m30,800,10,900,4,90,4,30	(x,y,z and a axes, with 4 axes)

explanation: "m" specifies that an absolute position will follow. For reasons of compatibility with the relative positioning command, for the z axis two pairs of numbers are expected also here when 3 axes are used. The second position specification of the z position, however, must then be zero and will be ignored. After storing, the control system will response with the handshake character.

restriction: The control system will not check whether the movement goes beyond the admissible range of the connected mechanical system.

3.2.5. Setting Zero in CNC Mode

command: Set zero at virtual point

application: The control system will store a command to set the current position as the virtual zero point for the specified axis/axes when executing the CNC program. The next following Traverse Absolutely movements will then refer to this virtual zero.

structure: n<axes><CR>

n	= Set Zero command code
<axes>	= axis specification, see below
<CR>	= Carriage Return to complete the command

notation: n7, n1, n8

explanation: "n" specifies that a zero offset is to be carried out. After the command code, the control system are communicated the axes for which a zero offset is to be carried out. Each axis is internally represented by a bit of a binary value, resulting 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

After storing, the control system will response with an appropriate feedback.

restriction: The virtual zero point is only important for the Absolute Movement command. The virtual zero point has no influence on the relative positioning, since a relative traversing vector is specified here.
CAUTION: The zero offset for the A axis must always be carried out separately.

3.2.6. Enabling/Disabling 3D Interpolation in CNC Mode

command: Enable/disable 3D-linear interpolation

application: The control system will save the command in order to be able to extend the 2.5D interpolation of the operating system to 3-dimensional interpolation. This command can be used to enable/disable this interpolation depending on the particular task.

structure: z<status><CR>

z	= 3D interpolation command code
<status>	= 0 --> disable, 1 --> enable
<CR>	= Carriage Return to complete the command

notation: z1, z0

explanation: "z1" is used to change the interpolation from 2D to 3D operation. The statement is modal, i.e. all relative and absolute movements are carried out three-dimensionally. The specification of z2 parameters with 3 axes will be ignored in the case of these traversing movements. The velocity specification of the interpolation must be carried out with the x specification. In the case of 4 axes, the 4th axis are traced.

3.2.7. Plane Selection for Circular Interpolation in CNC Mode

command: Plane selection

application: Setting of the interpolation plane for the circular interpolation. Circles are only defined within one plane. The default plane for the circular interpolation is the XY plane. However, it is also possible to define any plane configuration other than the circle plane using the Plane Selection command.

structure: e<plane><CR>

e = command code for "Set Circle Plane"
<plane> = plane specification, see below
<CR> = Carriage Return to complete the command

notation: e1, e0

explanation: "e" specifies that the plane for the circular interpolation is to be set. The following numerical value defines the plane as follows:

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

restriction: This command has modal effect, i.e. a plane selection for the circular interpolation remains stored until it is overwritten by a new plane selection.

3.2.8. Setting the Circle Direction for the Circular Interpolation in CNC Mode

command: Set Circle Direction

application: Setting the circle direction for the circular interpolation. The circular interpolation is initiated by two successive commands. The first command defines the direction of the circle, and the second command (see 3.2.9.) transfers the interpolation parameters.

structure: f<direction><CR>

f = command "Set "Circle Direction"
<direction> = 0 --> CW (clockwise), -1 --> CCW (counter-clockwise)
<CR> = Carriage Return to complete the command

notation: f-1, 0f0

explanation: The control system is addressed using @0. "f" specifies that the direction for the circular interpolation is to be set. The following numerical value defines the direction as follows:

0 --> CW (circular interpolation arc CW)
-1 --> CCW (circular interpolation arc CCW)

restriction: The direction for the circular interpolation must always be programmed before any circular movement is programmed.

3.2.9. Circular Interpolation in CNC Mode

command: Circular interpolation

application: Saving motion commands for circles and arcs at constant path velocity. The circular interpolation is initiated by two successive commands. The first command defines the circle direction (see 3.2.8.), and the second command transfers the interpolation parameters.

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

y	= command code for circular interpolation
<B>	= arc length in steps
<V>	= velocity
<D>	= interpolation parameter
<Xs>	= starting point x
<Ys>	= starting point y
<Rx>	= direction x
<Ry>	= direction y
<CR>	= Carriage Return to complete the command

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

explanation: "y" specifies that a circular interpolation is to be saved. The arc length specifies the length of the arc between starting and end points of the circular interpolation in steps. For the velocity, all integer values within the valid range of values for velocities are permitted. The interpolation parameter has to be transferred, because the control system cannot calculate this parameter by itself due to its memory capacity. The parameters Xs and Ys specify the starting point of the arc relatively to the circle centre point. Rx and Ry specify in which quadrant of the circle the interpolation starts. After the command has been executed, the control system will response with the handshake character ("0") as the feedback.
CAUTION: For calculating the parameters, please read the Section "Calculating the Parameters for Circular Interpolation".

restriction: The control system will not check whether the movement goes beyond the admissible range of the connected mechanical system.

3.2.10. Loops and Branches in CNC Mode

command: Loop, branch

application: Saving loops and branches. Loops are intended to summarise movement sequences of the same kind. Thanks to this feature, the memory available in the control system is used more efficiently. Branches can be used to jump after a logical decision to a certain block in the program.

structure: 3<number>,<offset><CR>

3	= command code for loop, branch
<number>	= number of loops Loop: 0 < number of loops < 32768 Branch: always 0
<offset>	= branch destination Loop: -1 >= branch destination >= -32768 Branch: -32768 <= branch destination <= 32767
<CR>	= Carriage Return to complete the command

notation:

3 25,-1	Repeat the last command 25 times
3 0,-5	Jump 5 steps back
3 0,5	Skip the next 4 commands
3 6,-5	Repeat the last 5 commands 6 times

explanation: If the control system finds the command "Loop/branch" during the execution of the CNC program, first the number of loops is checked to decide whether it is a loop or branch command. In the case of a loop command, a loop counter is set up, loaded with default values, and the command counter is corrected by the offset specified. The commands up to the next loop counter are now repeated, and the loop counter is decremented until it has reached zero. Then the program continues with the execution of the first command after the loop. Loops may be nested with a nesting depth of 15. The required counters are managed on an appropriate loop stack. In the case of a branch, the offset will be understood as a relative branch destination within the NC program and the command counter be corrected by the offset.

restriction: Branching before the start or after the end of the data field is not permitted. Forward loops are permitted. A loop will always repeat the last n commands. At least one command must be repeated. Loops may be nested; the maximum nesting depth is 15. Leaving a loop via a branch is not permitted.

3.2.11. Time Delays in CNC Mode

command:	Time delay
application:	Storing time delays.
structure:	5<time><CR>

5	= time delay command code
<time>	= time in 1/10 sec
<CR>	= Carriage Return to complete the command

notation: 3 50 delay of 5 seconds

explanation: If the control system finds the Time delay command during the execution of the CNC program, the next command in the CNC program will only be executed after the delay time has expired. The appropriate time will be specified in 1/10 seconds.

restriction: A time delay cannot be cancelled by pressing the Stop key of the control system.

3.2.12. Setting the Port in CNC Mode

command: Set output port

application: Defined enabling/disabling of existing output ports.

structure: p<port No.>,<bit No.>,<value><CR>

p = command code "Set Port"
<port No.> = port number
<bit No.> = bit number, 0 - 7 --> bit by bit, 128 --> byte by byte
<Value> = new value
<CR> = Carriage Return to complete the command

notation:

p2,128,1	Port 0, setting to "1" by bytes
p2,0,1	Port 0 , setting bit 0 to 1

explanation: "p" specifies that the value of an output port is to be set. Then port number, bit number and new port value are transferred separated by commas, and the command is completed with Carriage Return. The control system will respond with the software handshake "0" if the saving operation has been successfully completed or with an error message if wrong port numbers and/or values have been transferred. For the IMC4-M control system, the following ports are defined with appropriate functionalities:

Port	Bit	Value	Function	
0	0 - 7	0 - 255	User I/O (also still possible with 65529)	
			Bit0	reserved
			Bit1	reserved
			Bit2	reserved (relay, spindle)
			Bit3	User Output 1
			Bit4	User Output 2
			Bit5	Output Brake
			Bit6	User Output 3
			Bit7	reserved
1	0	0	Cover may not be opened	
	0	1	Cover may be opened	
2	0	0	Spindle OFF	
	0	1	Spindle ON	
3	0	0	Motor currents OFF	
	0	1	Motor current ON	
4	0 - 7	0 - 255	analog output 0 – 10V	
5	0 - 7	0 - 255	User I/O I2C-port extention (I2C-address=0)	
6	0 - 7	0 - 255	User E/A	
			Bit4	User Output 5
			Bit5	User Output 6
			Bit6	User Output 7
			Bit7	User Output 8

restriction: The setting of the port outputs is carried out in the control system as defined in the program. Setting or deleting outputs while a command is being executed, e.g. during a positioning movement, is not possible.

3.2.13. Reading a Port and Branching in CNC Mode

command: Read input port

application: Reading an input port and branching within the program. Thanks to the branching, it is possible to branch to a certain block after a logic comparison.

structure: o<port No.>,<bit No.>,<value>,<offset><CR>

o	= command code "Set Port"
<port No.>	= port number
<bit No.>	= bit number, 0 - 7 --> by bits, 128 --> by bytes
<value>	= comparison value
<Offset>	= branch destination -32768 <= branch destination <= 32767
<CR>	= Carriage Return to complete the command

notation:

o2,128,1,-1	Wait until port 0 <> 1
o2,0,1,-1	Wait until Port0, bit0 = 0
o2,0,1,3	If Port0, Bit0 == 1, command counter += 3

explanation: "o" specifies that the value of an input port is to be read and the program will be adapted according to the value. Then port number, bit number, comparison value and command offset are transferred separated by commas, and the command is completed with Carriage Return. The control system will respond with the software handshake "0" if the saving operation has been successfully completed or with an error message if wrong port numbers and/or values have been transferred. During the program execution, the appropriate port is polled and logically compared with the intended value either bit by bit or byte by byte. If the logic comparison is true, branching by the offset is carried out; otherwise, the next command is executed as intended by the program. For the IMC4 control system, the following ports are defined with the appropriate functionalities:

Port	Bit	Status	Function	
0	0 - 7	00 - FF	User I/O (also still possible with 65531)	
			Bit0	Input Stop button
			Bit1	Input Start button
			Bit2	User Input 1
			Bit3	Input Power OK
			Bit4	Input Emergency Stop switch
			Bit5	Input ON button
			Bit6	Input Cover switch
			Bit7	Input Keyswitch
1	0	00	Cover is open	
	0	01	Cover is closed	
2	0	00	Spindle is turned off	
	0	01	Spindle is turned on	
3	0	00	Motor currents are turned off	
	0	01	Motor currents are turned on	
5	0 - 7	00 - FF	User I/O I2C-port extention (I2C-address=0)	
6	0 - 7	00 - 1F	User I/O	
			Bit0	User Input 1
			Bit1	User Input 2
			Bit2	User Input 3
			Bit3	User Input 4
			Bit4	User Input 5

restriction: The port inputs are polled in the control system as defined in the program. Polling inputs while a command is being executed, e.g. during a positioning movement, is thus not possible.

3.2.14. End of Data Field in CNC Mode

command: End of data field

application: This command is used to mark the end of a CNC data field and serves to complete the data transfer and to save storable commands.

structure: 9<CR>

9	= command code "End of Data Field"
<CR>	= Carriage Return to complete the command

notation: 9

explanation: "9" specifies that the end of the CNC data field transferred is reached. The command is completed with Carriage Return. The control system will response with the software handshake "0" if the saving operation has been successfully carried out or with an error message. In addition to the marking of the data field as a valid CNC program, status information (e.g. the current referencing velocity) are stored in the FlashProm. The control system will then be in DNC mode again and will accept the appropriate commands.

restriction: A CNC data field must be completed with the end-of-data field command; otherwise, the saved CNC program will not be valid and cannot be executed.

4. Error Messages of the IMC4-M

After each transferred command, the control system will response with an appropriate feedback. These codes are transferred as ASCII characters and can thus be evaluated easily. Based on the character transferred, error sources and causes can be determined. The individual error codes are described in the following.

Code	Description
0	Handshake character - No error; the command has been executed correctly. - The next command can be transferred.
1	Error in transmitted number - The control system has received a numerical specification that could not be interpreted correctly. - The numerical value transmitted is outside the admissible range or contains illegal characters.
2	Limit switch error - As a result of the traversing movement, a limit switch has responded. The current movement has been cancelled. This is done by stopping the movement without braking ramp. As a result, the actual positions of the control system are no longer correct; step losses may have occurred. - The reference point approach of a stepper motor axis has been carried out not correctly or has not yet been carried out at all. **CAUTION:** After a limit switch error, the control system has to be re-initialised and a reference point approach to be carried out.
3	Illegal axis specification - The control system has been transmitted an axis specification for a command to be executed, which contains an axis not defined. - In commands that contain axis specifications you should only use combinations of axes that are initialised.
4	No axes defined - Before movements or, generally, commands are transferred to the control system which have a number of parameters which depends on the number of axes, the Set Axes command must be transferred in order to be able to set the internal axis parameters correctly.
5	Syntax error - A command has been transmitted with errors. - The command used does not exist or cannot be executed by this control system. - Check whether all transferred commands are correct.
6	End of memory - You have tried to transfer more commands in CNC mode, than can be stored in the control system.
7	Illegal number of parameters - The control system has received more or less parameters for the command than needed. - Check whether the number of parameters required for the command is correct in conjunction with the number of axes.
8	Command to be saved is correct - The control system has been transferred a command that is not available as a CNC command.
9	Plant error - The power supply of the plant is not yet turned on. - The safety relay of the plant is not active. - The output stages and/or the safety circuit could not be turned on, since the cover is still open. - An Emergency Stop situation has occurred. **CAUTION:** After an Emergency Stop situation, the control system must be re-initialised and a reference point approach be carried out.

A	not used by this control system
B	not used by this control system
C	not used by this control system
D	Illegal velocity - The admissible limits for velocity specifications have not been observed. - Check whether all velocity specifications are correct.
E	not used by this control system
F	User stop - The user has actuated the Stop key on the control system and the movement currently active has been stopped. The command execution can be resumed either by pressing the Start key or using the Start command @0s.
G	Invalid data field - The control system has been transferred a Start command although no remaining movement to be traversed is left in the memory, i.e. although no stop function has been carried out beforehand. - You have tried to transfer a CNC program although a program or parts of a program are still contained in the memory.
H	Cover error - You have tried to execute a command that is not permissible with open cover.
=	not used by this control system

5. Default port specifications for IMC4-M

Default I/O settings for IMC4-M to work in conjunction with Geckoboard

signal / specification default setting active level	logic port* in ProNC / Remote	local Port ** at IMC4-M	connector	location	terminal- command	connection at IMC4-M IO-interface
tool holder: 1=open / 0=close	A1.4	1	X30	Geckoboard	@0B0,8	
workpiece holder: 1=on / 0=off	A1.5	1	X31	Geckoboard	@0B0,16	
cooling: 1=on / 0=off	A1.7	1	X32	Geckoboard	@0B0,64	
toolchange cover: 1=open / 0=close	A2.5	7	CN106.6	IMC4M	@0B6,16	Out5
output 2.6 (free)	A2.6	7	CN106.2	IMC4M	@0B6,32	Out6
output 2.7 (free)	A2.7	7	CN106.7	IMC4M	@0B6,64	Out7
output 2.8 (free)	A2.8	7	CN106.3	IMC4M	@0B6,128	Out8
spindle: 1=on / 0=off	X	3	X19	Geckoboard	@0B2,1	
stop-key operators panel: 0=pressed	E1.1	1	X21	Geckoboard	@0b0	
start-key operators panel: 1=pressed	E1.2	1	X34	Geckoboard	@0b0	
tool legth sensor: 0=pressed	E1.3	1	X27	Geckoboard	@0b0	
PowerOK: 1=power stage voltage ok	E1.4	1	X29	Geckoboard	@0b0	
input 2.2 (free)	E2.2	7	X28	Geckoboard	@0b6	
tool change cover: 1=closed	E2.3	7	CN106.8	IMC4M	@0b6	In1
compressed air: 1=pressure ok	E2.4	7	CN106.6	IMC4M	@0b6	In2
input 2.5 (free)	E2.5	7	CN106.9	IMC4M	@0b6	In3
start-key operators panel: 1=pressed	E3.4***	16	X34	Geckoboard		

* "logic port" starting with 1 ** "local port on IMC4-M" starting with 1 *** depending on configuration

note: these are binding specifications since 39/2005 to work in conjunction with IMC4-M-firmware version V3.2.0

6. Pin assignment of IMC4-M

CN1 SubD-37 male

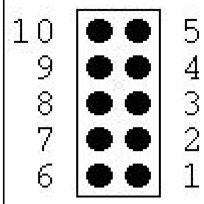
Pin SubD37 IMC4-M	Signal	Terminal at board
1	step X	
2	step Z	
3	nc	
4	direction Y	
5	direction A	
6	+24V	
7	User Output 1	X30
8	User Output 3	X32
9	Output Enable Powerstage	
10	GND 24V	
11	Reference switch A (negative end switch)	
12	Reference switch Z (negative end switch)	
13	Reference switch Y (negative end switch)	
14	Reference switch X (negative end switch)	
15	nc	
16	GND analog (0-10V)	X20
17	Input PowerOK	X29
18	UserInput 2	X28
19	Input StartKey	X34
20	step Y	
21	step A	
22	direction X	
23	direction Z	
24	nc	
25	+5V	
26	UserOutput 2	X31
27	UserOutput Spindle	X19
28	Output brake	
29	positive end switch A	
30	positive end switch Z	
31	positive end switch Y	
32	positive end switch X	
33	nc	
34	analog output 0-10V	X20
35	Input StopKey	X21
36	User Input 1	X27
37	GND 24V	

CN104 SubD-9 male

Pin	Signal
1	
2	RXD
3	TXD
4	
5	Digital GND
6	
7	start key
8	
9	

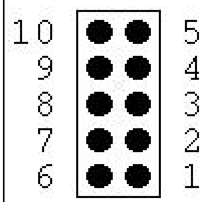
CN105 terminal 2X5pole

Pin	Signal
1	24V-supply
2	output cover enable
3	input emergency stop
4	input key switch
5	
6	output SoftwareOK
7	input power on key
8	input cover closed
9	
10	GND-24V



CN106 terminal 2X5pole

Pin	Signal
1	24V-supply
2	User Output 6
3	User Output 8
4	User Input 4
5	
6	User Output 5
7	User Output 7
8	User Input 3
9	User Input 5
10	GND-24V



CN201 terminal 2X2pole

Pin	Signal
1	5V-supply
2	I2C-Bus data
3	I2C-Bus clock
4	Digital GND