USER - Manual

MACCONTROLLER (MAC)

MAC4-INC

Version 4.2

MAC4-SSI

Version 4.2

MAC4-STP

Version 2.2

MACCON GmbH, June 1996

MACCON GmbH
Kuehbachstr. 9
D-81543 Munich
Tel. +49 89 651220-0
Fax. +49 89 655217

Change Status

Version	Date	Chapter	Changes
1.1e	30.09.96	all	First issue
1.2e	03.03.98	all	Maccon/port/ESO

MACCON would like to thank *port* for their development work on the MAC series of position controllers and the engineers of the European Southern Observatory (ESO) for their part in the generation and correction of this documentation

MACCON GmbH
Kuehbachstr. 9
D-81543 Munich
Tel. +49 89 651220-0
Fax. +49 89 655217
e-mail sales@maccon.de

port Automation
Antonienstr. 3
D-06749 Bitterfeld
Tel. +49 3493 743-10
Fax. +49 3493 743-15
e-mail service@port.de

General

This documentation of the motor-axis controller family MAC4 consists of a User- and a Reference Manual.

This User Manual serves as an introduction in the use of the MAC series of axis controllers. The procedure for the integration of axis controllers in the user system is described here.

The Reference Manual includes an instruction description for each command.

The page references given in the index refer to this User Manual only. Please also refer to index of Reference Manual. Differentiation is made between the manuals by the roman numbers „I“ (User Manual) and „II“ (Reference Manual).

Conventions

- Parameter for the axis controllers are designated in the text as follows:
  <abbreviation for parameter>
  A summary of all these abbreviations is given in the appendix.

- Hexadecimal numbers are designated in the text as follows:
  $hexadecimal number

- Abbreviation for instructions in the form of mnemonics is given in capitals:
  INSTRUCTION

- Instructions, which write parameters to the controller are designated with an additional "_R", if these are relative write instructions:
  WRITE_R relative parameter value

- The given abbreviations for the instructions only serve to simplify documentation. Transmission of instructions to the axis controllers is performed always as a number code.

- Special notes and warning indications are provided as number codes and are provided in a box with an arrow:

⇨ WARNING!

- Descriptions of abbreviations and designations are provided in the glossary.

List of Contents

1. Overview of the VMEbus-axis Controller Family MAC

The MAC VMEbus Controller Family consists of initially three flexible and readily configurable servo- and stepper-axis positioning controllers for application in open control system architectures.

These components were mainly developed by MACCON; this means that we are in a position to extend and adapt the functionality and firmware at any time to match specific application requirements.

MACCON offers engineering capacity alongside normal product support in order to provide optimum solutions for demanding motion control tasks.

1.1 Hardware Concept

An overview over all axis controllers of the MAC family of 6HE-VMEbus-cards is given in Tab.1.

The controllers consist of two sub-cards, the first being a 3HE-CPU-Module, which is either:

an "IPC" intelligent peripheral controller or

an "APC" advanced peripheral controller,

the second being an adapter interface module for interface to the motor power-stages and the axes. This interface module also includes interfaces to a position measurement system, reference and limit switches as well as additional binary inputs and outputs. One controller can serve 4 axes. Each axis can be programmed independently of the others.

Type	CPU	Drive interface	Measurement system
MAC4-INC	MC 68000	±10 V	Incremental encoder
MAC4-INC-A	MC68332/DSP	±10 V	Incremental encoder
MAC4-SSI	MC68000	±10 V	SSI-Absolute encoder
MAC4-SSI-A	MC68332/DSP	±10 V	SSI-Absolute encoder
MAC4-STP	MC68000	Step,	Incremental encoder,
		Direction	Step counter
MAC4-STP-A	MC68332/DSP	Step,	Incremental encoder,
		Direction	Step counter
MAC4-UNI	MC68332/DSP	±10 V,	Incremental encoder,
		Step,	SSI-Absolute encoder,
		Direction	Step counter

Table 1.1: MAC Axis Controller Family

External position measurement systems can be employed in addition to the systems mentioned in Tab.1.1. The axis position is transferred via the VMEbus to the axis controller or read autonomously by the controller over the bus.

All axis controllers of the family possess very similar features with the exception of their drive interface functions.

The master capability of the VMEbus-Interfaces of both types of CPU-modules used (IPC and APC) extends the possibility for system integration. This also applies for the upward scaling of the APC when implementing the DSP-option.

In addition to the powerful communications capability via VMEbus the APC offers a CAN-Interface for the close integration of further measurement systems or process I/O.

The axis interfaces of the controller are potentially isolated from the processor module, in order to ensure a high level of safety and noise immunity. Further, the similar connector pinning employed by all the types of the controller ensure the greatest possible standardisation in the application and wiring of the drives and the axes when using these cards.

A number of display functions on the front-panel of the controller indicate the state of the control of the individual axes, in order to assist in commissioning and fault-finding.

This documentation specifically describes the axis controller types: MAC4-INC, MAC4-SSI and MAC4-STP.

1.2 Software Concept

The extensive system software with more than 250 instructions (with and without parameters) for the control and parameterisation of motions functions is addressed by means of a numerical command set, which has been optimised for field-bus applications. The functionality of the firmware is illustrated in the signal flow chart of Diagram 1.1 (The CAN-interface is not implemented in all versions of the MAC-family).

The instructions, that have been implemented in these axis controllers may be split into 6 groups:

1. General instructions
   (Initialisation, confirmation of parameter changes, error confirmation)

2. Instructions for mode change
   (Positioning etc.)

3. Instructions for system set-up
   (Direction of motor rotation, "on-the-fly"-operation etc.)

4. Instructions to read/write system specific parameters
   (position measurement systems, axis type etc.)

5. Instructions to read/write mode dependent parameters
   (mode specific velocities, accelerations, delays etc.)

6. Diagnosis instructions
   (reading of status and position information etc.)

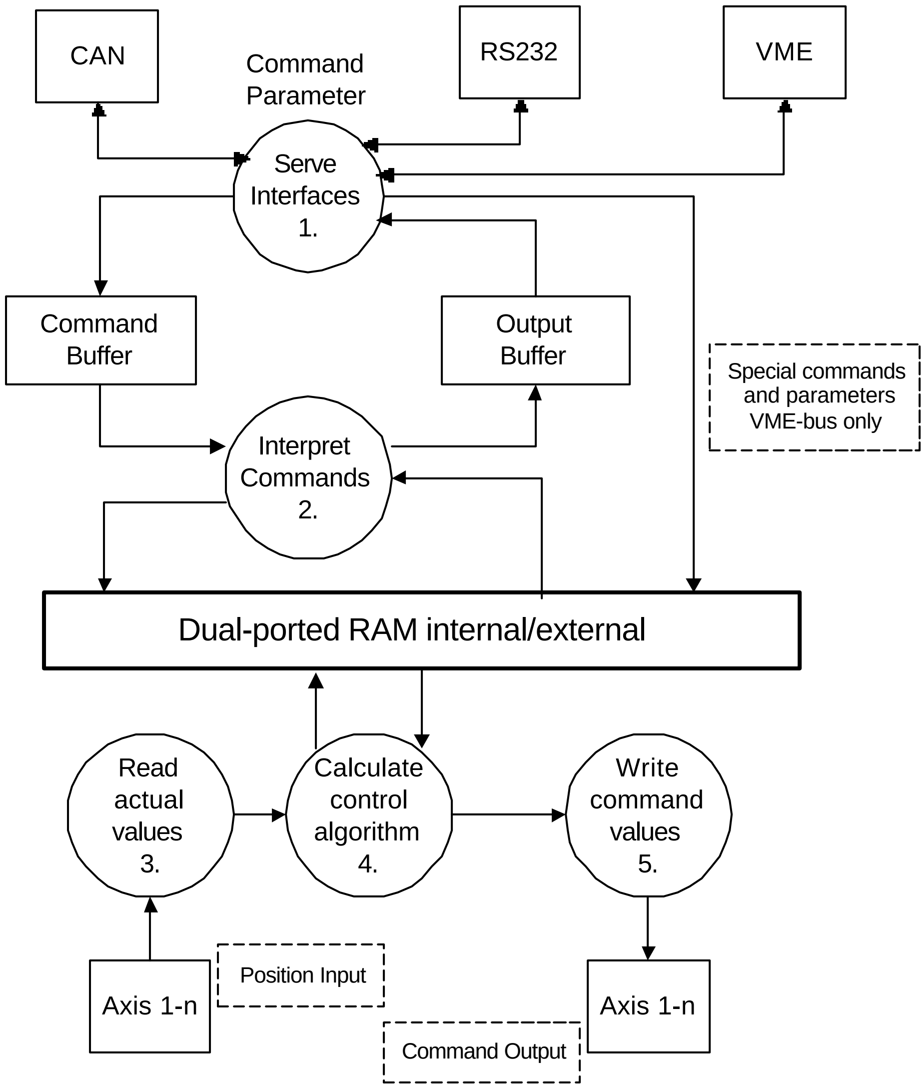


Diagram 1.1 Data flow Chart for MAC-Firmware

The parameter set includes approx. 50 parameters, that are needed for the definition or change of values such as:

- Motion definition,
- Measurement system parameters,
- Limit switches,
- Servocontrol coefficients and
- Interrupt vectors.

A number of different operating modes have been implemented:

- Axis disabled ("disable"),
- Axis enabled ("enable"),
- Velocity control ("speed"),
- Axis stopped ("brake"),
- Positioning ("positioning"),
- Coarse search for the index pulse ("search index coarse"),
- Fine search for the index pulse ("search index"),
- Search for reference switch ("home"),
- Search for reference switch edge ("find edge"),
- Velocity trajectory ("velocity tracking"),
- Position trajectory ("position tracking"),
- Test ("test").

An overview over the operating modes, that can be activated by a corresponding command is provided by the state change Diagram 1.2. The term "Move" covers all motion modes.

1.3 Axis Control

The MAC4 family can control the following motor and encoder configurations:

- DC-Servomotor with incremental encoder (MAC4-INC) or
- DC-Servomotor with absolute SSI-encoder(MAC4-SSI) or
- Stepper and microstepper drives in normal- or TURBO-operation, incremental encoder feedback is possible (MAC4-STP)

The axes can be:

- linear
- rotary or
- 360° rotary (with shortest path algorithm)

Motion control is performed either closed-loop (MAC4-INC/SSI) or open-loop with the option of feedback position correction (MAC4-STP).

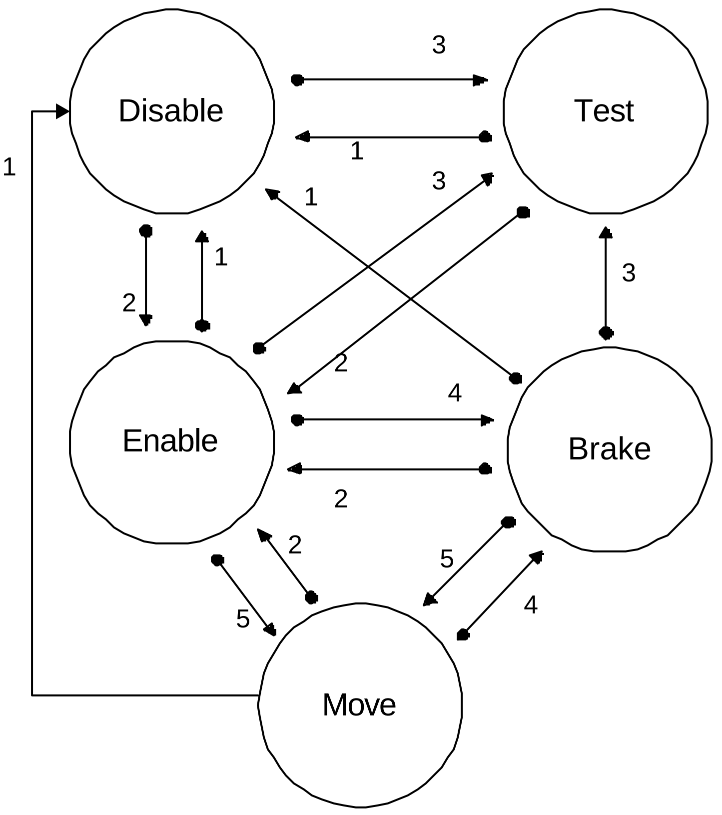


Diagram 1.2: State Change Diagram (simplified)

1	MD
2	ME
3	UE
4	MB
5	MC, MF, MH, MI, MP, MS, PT, VT

2. Hardware

2.1 Processor Module IPC

The processor module of the axis controllers consists of the functional blocks illustrated in Diagram2.1.

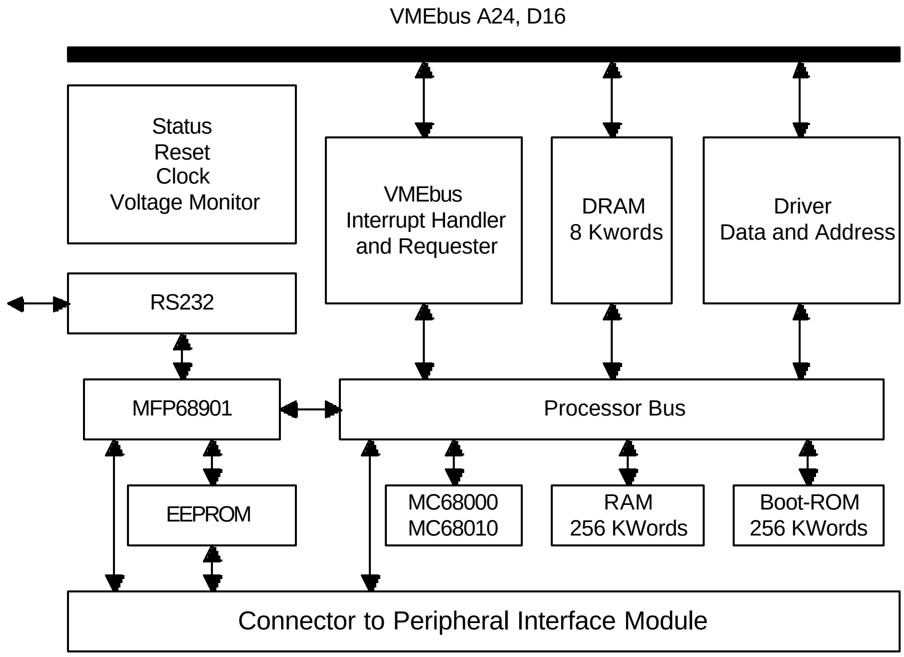


Diagram 2.1: Block Diagram of IPC

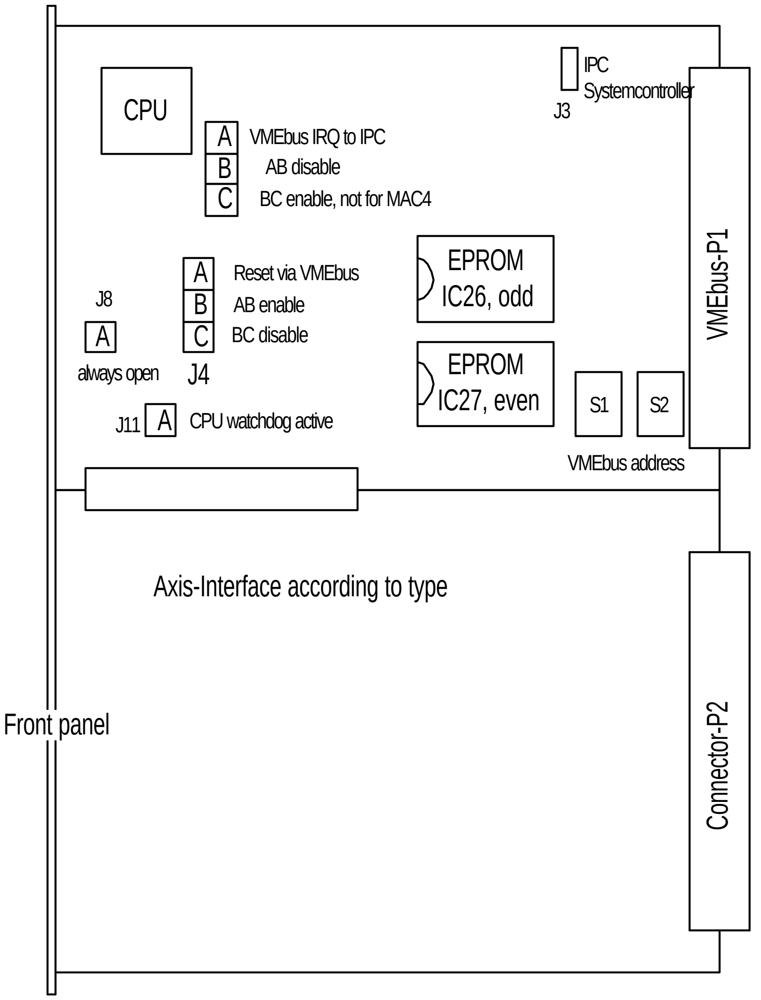
The necessary set-up conditions for the operation of the overall module on the VMEbus are described in the following sections. Detailed information is provided in [1].

2.1.1 Address Allocation

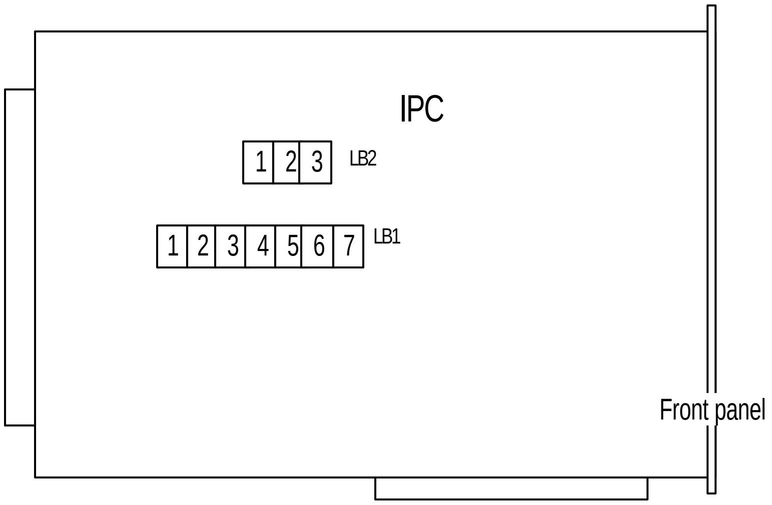
The axis controller DPRAM occupies 64 KByte of address space on the VMEbus. The base address of the DPRAM-space can be adjusted within the address range of the VMEbus, $100000 to $BEFFFF, with switches [S1] and [S2] on the IPC (Diagram2.2). The address is defined by the switch settings as follows:

$[S1][S2]0000

Before commissioning the card it is necessary to check whether the DPRAM address space lies within the addressable address space of the host system and whether possible address conflicts may occur.



component-view



rear-view

J ... Jumper, designation of set jumpers

Diagram 2.2: Front and Rear View of the IPC

2.1.2 Host-Reset

Jumper J4 (see Diagram2.2) allows the user to define whether a reset of host computer causes the axis controller to reset or not:

A-B: A host reset causes the axis controller to reset

B-C: A host reset has no influence on the axis controller

2.1.3 Software-Reset

The axis controller can also be reset with the instruction MR (see Reference manual). This command uses the "watchdog" of the IPC and functions only, when the Jumper J11 is in place (see Diagram2.2).

2.1.4 Interrupts

The axis controller supports interrupt vectors at the VMEbus. The interrupt level can be selected by means of the solder bridges LB1 and LB2 on the IPC (Diagram2.2) according to Tab.2.1.

A detailed description of the interrupts and their use is provided in Section 4.

Level	LB2-1	LB2-2	LB2-3	LB1
7	x	x	x	7
6	x	x	-	6
5	x	-	x	5
4	x	-	-	4
3	-	x	x	3
2	-	x	-	2
1	-	-	x	1

x ... closed
- ... open

Table 2.1: Setting of interrupt levels

The axis controller does not support interrupts generated by the host.

⇨	The jumper J7 on the IPC must be in position A-B.

2.2 Peripheral Connector MAC4-INC

The axis interface of the MAC4-INC comprises the functional blocks shown in Diagram2.3.

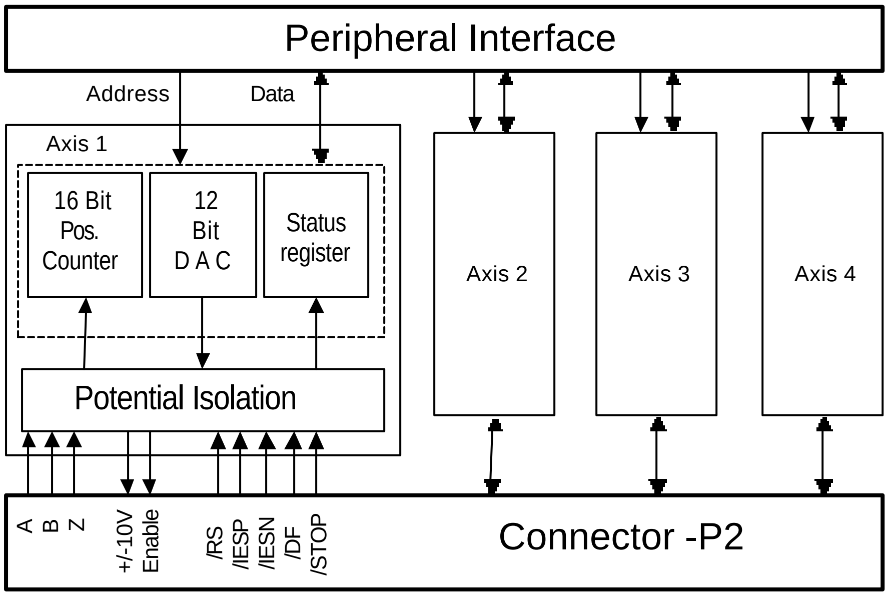


Diagram 2.3 MAC4-INC Block Diagram of Axis Interface

The connection of the axis controllers to the drive is made via the peripheral connector -P2 (Diagram2.4). The Pin allocation is given in the appendix.

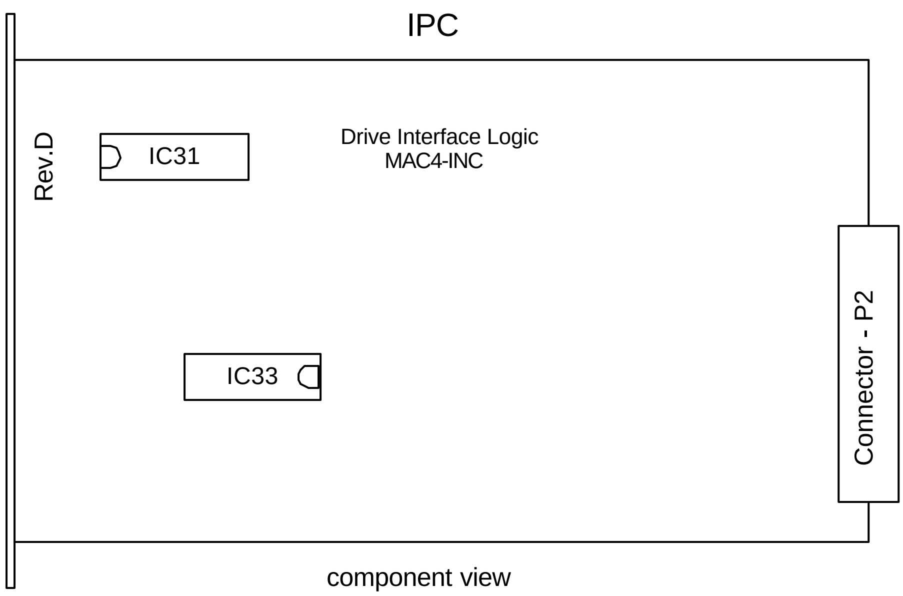


Diagram 2.4 MAC4-INC: Component view

2.2.1 Motor

The motors are controlled by means of the signals listed in Tab. 2.2.

Designation	Signal name	Signal level	Signal level after Reset
Enable	/EN(1...4)	0 V - axis enabled o.c - axis disabled	H
DAC-Output	AOUT(1...4)	±10V	0 V

o.c. ... „open Collector“ - Output open

Table 2.2: MAC4-INC: Motor Signals

2.2.2 Position Measurement Systems

There are three ways of measuring position:

- Direct connection of an incremental encoder via the P2-connector (= internal position measurement),
- Position measurement (incremental or absolute) by access through the VMEbus to an external position counter (= external position measurement),
- Supply of the position information (incremental or absolute) by a host via the DPRAM of the axis controller (= external position measurement)

The connection of an incremental encoder to the axis controller is made according to Tab. 2.3 (See appendix for the pin allocation of the connectors P2-connectors).

Designation	Signal name	Signal level
Encoder inputs	Ch_A(1..4) /Ch_A(1..4) Ch_B(1..4) /Ch_B(1..4) Ch_Z(1..4) /Ch_Z(1..4)	TTL (differential or unipolar)

Table 2.3: MAC4-INC: Inputs of Position Counter

The A- and B-tracks supply the basic counting pulses. The position counter can however generate a four times higher resolution by decoding the 90° shift between the two signals. The Z or reference input (index pulse) however can only register a single pulse per encoder revolution. This signal serves for referencing or justification of the absolute position information within the axis controller. The index pulse is recognised per interrupt.

2.2.3 Input Signals

Various switches can be connected via the P2-connector (see Tab.2.4). The emergency stop signal must be present. The signal level of all other switches are configurable (see Section 4).

Designation	Signal name	Signal level low active on	Signal level low active off	Signal level high active on	Signal level low active off
Positive limit switch	/IESP(1..4)	0 V	24 V	24 V	0 V
Negative limit switch	/IESN(1..4)	0 V	24 V	24 V	0 V
Reference switch	/RS(1..4)	0 V	24 V	24 V	0 V
„drive fault“	/DF(1..4)	0 V	24 V	24 V	0 V
Emergency stop	/STOP	0..12 V	12..24 V	-	-

Table 2.4: MAC4-INC: Switch Signals

The emergency stop signal is effective for all axes. Activation of the emergency stop switch causes all axes to be deactivated immediately.

⇨ It must be ensured, that the hardware limit switch remain active over the complete forbidden range. There is no protection against movement in the forbidden direction, if the axis overruns a hardware limit switch and comes to halt behind the switch.

2.2.4 Fault Signals

The MAC4-INC can recognise three fault code signals (FC-Signals) at the P2-connector. This information can be made available to the host system by reading the axes general status information. They do not indicate any status of the axis controller itself.

These inputs can also be used for other signals, according to the user’s choice - typically the status of the drives being controlled.

Designation	Signal name	Signal level
Error code	FC(0..2)	TTL

Table 2.5: MAC4-INC: FC-Signals

2.3 Peripheral Connector MAC4-SSI

The axis interface of the MAC4-SSI consists of the functional blocks shown in Diagram 2.5.

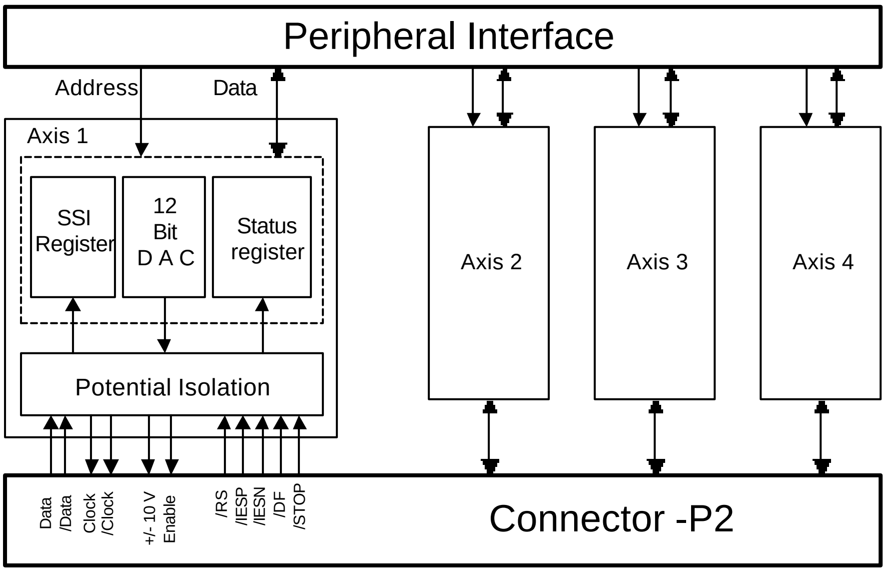
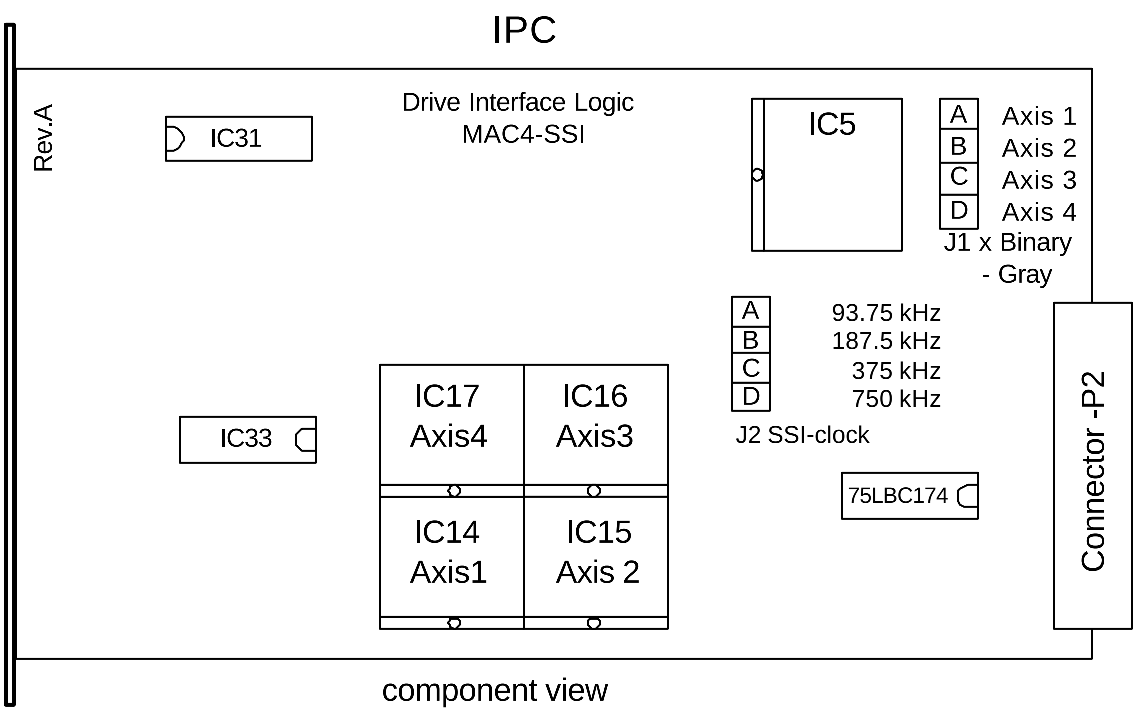


Diagram: 2.5 MAC4-SSI: Block Diagram of the Axis Interface

The connection of the axis controller to the drives is made via the connector -P2 (Diagram 2.6).
The pin allocation is provided in the appendix.



J ... Jumper, designation of set jumpers (see Diagram 2.2)
x ... closed
- ... open

Diagram: 2.6 MAC4-SSI: Component View

2.3.1 Motor

The control of the motors is made by means of the signals listed in Tab. 2.6.

Designation	Signal name	Signal level	Signal level after Reset
Enable	/EN(1...4)	0 V - axis enabled o.c - axis disabled	H
DAC - Output	AOUT(1...4)	±10 V	0 V

o.c ... „open collector“ - Output open

Table 2.6: MAC4-SSI: Motor Control Signals

2.3.2 Position Measurement Systems

There are three possibilities of measuring the actual position of the motor:

- Direct connection of a SSI-absolute encoder via the connector -P2 (= internal position measurement),
- Position measurement (incremental or absolute) by access over the VMEbus to a separate position counter (= external position measurement),
- Supply of the position information (incremental or absolute) by the host to the DPRAM of the axis controller (= external position measurement).

An absolute position sensor with a resolution of up to 24 Bit can be connected to the axis controller using internal position measurement.

The connection of the Clock - and Data lines is via the peripheral connector -P2 (see Tab.2.7). The pin allocation of the connectors -P2 is given in the appendix.

Designation	Signal name	Signal level
SSI-Clock	T(1..4) /T(1..4)	TTL-differential
SSI-Data	DAT(1..4) /DAT(1..4)	TTL-differential

Table 2.7: SSI-Encoder Connections

The MAC4-SSI transmits clock signals to the connected position sensor (Diagram4.4). At the first H-L-transition (high-low) the position information is buffered and then transmitted over the data lines to the MAC4-SSI with the most significant data bit first, starting at the first L-H transition. At each following L-H-transition the next less significant bit is transmitted until the last bit has been sent. After completion of transmission and an additional recovery time the position sensor returns to its original condition. Thereafter the next read cycle can be initiated. The time period between two position interrogations by the axis controller lies well above the recovery time. Diagrams for the recovery time of an SSI-encoder are provided in the data sheet of the sensor manufacturer.

The clock signal to the absolute encoder is provided by the axis controller via the Jumper J2 at the clock rates stated in Tab.2.8 (Diagram2.6). 26 clock pulses are always generated.

⇨	Only **one** Jumper for the SSI-Baud rate may be set.

The maximum permissible Clock rate is conditioned by the cable length.

J2	Clock rate in kHz	Cable lengths in m
A	93.75	<400
B	187.5	<200
C	375	<100
D	750	<50

Table 2.8: MAC4-SSI: Setting of the Clock Rate

The required SSI clock rate is chosen by closing the appropriate the corresponding jumper.

The position information can be coded in Binary- or Gray code. The selection is made for each axis separately with the jumper J1 (Diagram2.6) according to Tab.2.9

J1	Axis	Gray-Code	Binary-Code
A	1	open	closed
B	2	open	closed
C	3	open	closed
D	4	open	closed

Table 2.9: Setting of the Code Selection

2.3.3 Input signals

Various switches can be connected via the connector -P2 (see Tab.2.10). The emergency stop signal must be connected. The signal levels of all other switches are configurable (see Section 4)

Designation	Signal name	Signal level	Signal level	Signal level	Signal level
		low active	low active	high active	high active
		on	off	on	off
Positive limit switch	/IESP(1..4)	0 V	24 V	24 V	0 V
Negative limit switch	/IESN(1..4)	0 V	24 V	24 V	0 V
Reference switch	/RS(1..4)	0 V	24 V	24 V	0 V
„drive fault“	/DF(1..4)	0 V	24 V	24 V	0 V
Emergency stop	/STOP	0..12 V	12..24 V	-	-

Table 2.10: MAC4-SSI: Switch Signal Levels

The emergency stop signal is effective for all axes simultaneously. Activation of the emergency stop switch causes all axes to be disabled immediately.

⇨ It must be ensured, that the hardware limit switch remain active over the complete forbidden range. There is no protection against movement in the forbidden direction, if the axis overruns a hardware limit switch and comes to halt behind the switch.

2.3.4 Fault Signals

The MAC4-SSI can interrogate three fault code signals (FC-Signals) via the connector -P2. These codes can be made available to the host system for all axes as joint status information.

These inputs can also be used to interrogate other user signals.

Designation	Signal name	Signal level
Error code	FC(0..2)	TTL

Table 2.11: MAC4-SSI: FC-Signals

2.4 Peripheral Connections MAC4-STP

The axis interface of the MAC4-STP consists of the functional blocks shown in Diagram2.7.

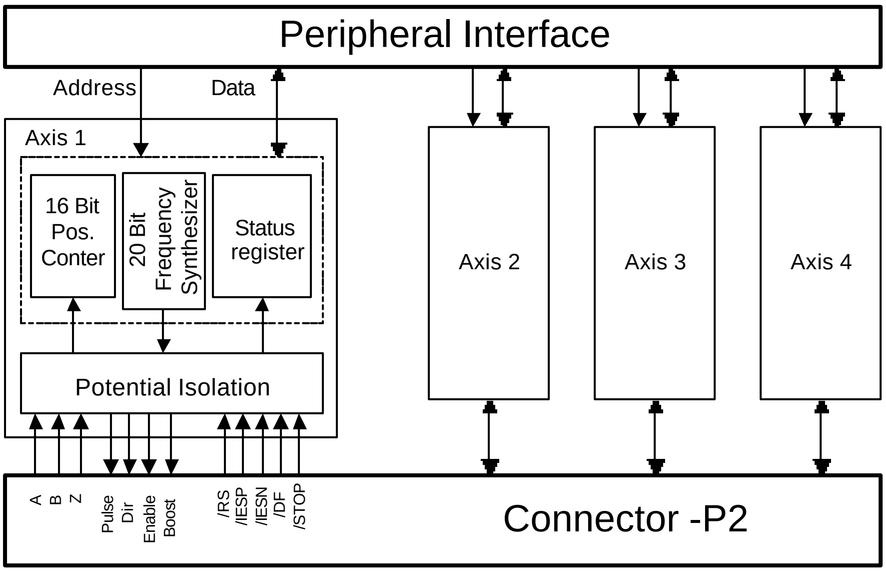


Diagram 2.7: MAC4-STP: Block Diagram of Axis Interface

The connection of the axis controller to the stepper drivers is made via the connector -P2 (Diagram2.8). The pin allocation is given in the appendix.

2.4.1 Motor

The control of the stepper drives is made by means of the signals listed in Tab.2.12.

Designation	Signal name	Signal level	Signal level after Reset
Direction	DIR(1..4)	0 V - positive oc - negative	oc
Enable	/EN(1..4)	0 V - axis enabled oc - axis disabled	oc
Pulse	P_OUT(1..4)	step on the falling transition (negative pulse)	oc
Boost	BOOST(1..4)	0 V - „hold“ oc - „run“	oc

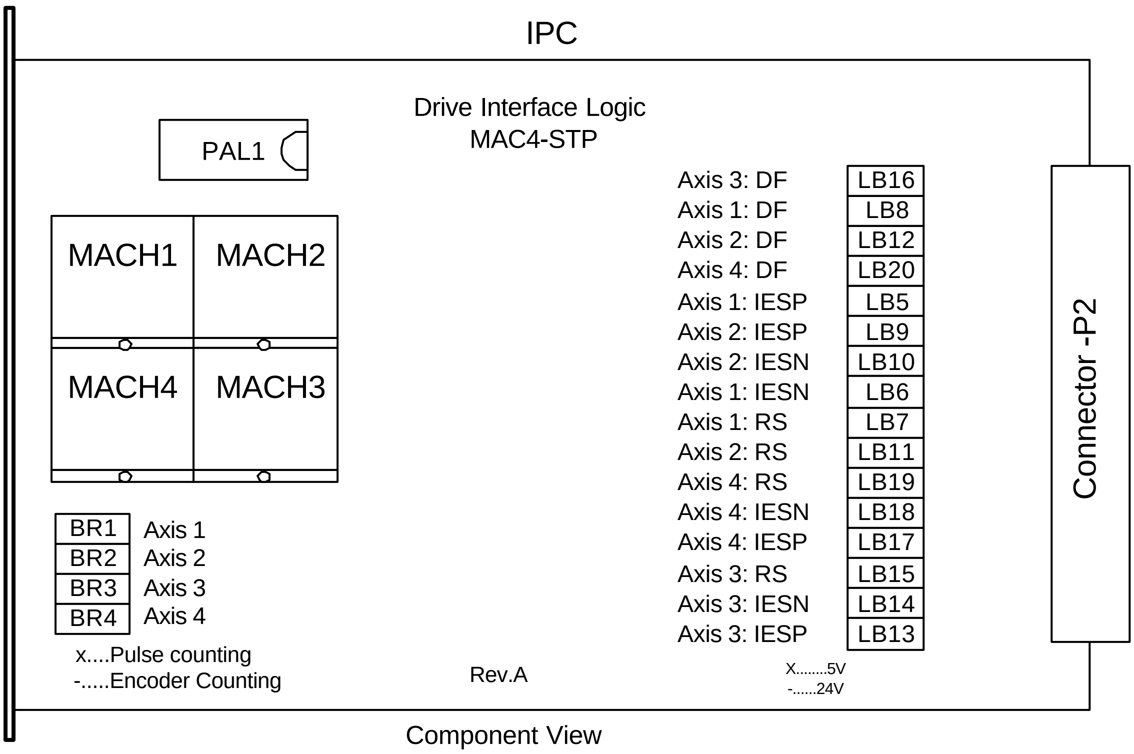
oc ... „open collector“ - Output open

Table 2.12: MAC4-STP: Motor Signals

2.4.2 Position Measurement Systems

There are three ways of measuring position:

- Direct connection of an incremental encoder via the peripheral connector -P2-connector (= internal position measurement),
- Position measurement by counting the number of steps generated (= internal position measurement),
- Position measurement (incremental or absolute) by access through the VMEbus to an external position counter (= external position measurement),
- Supply of the position information (incremental or absolute) by a host via the DPRAM of the axis controller (= external position measurement).



BR ... Bridge/Jumper
LB ... Solder bridge
x ... closed
- ... open

Diagram 2.8: MAC4-STP: Component View

Using the solder bridge BR1-4 (Diagram2.8) it is possible to distinguish whether the actual position is measured by counting the steps generated by the controller axis or by the encoder signals from the peripheral connector -P2.

If position measurement is made via a separate position encoder, the position of the solder bridge BR1-4 is not relevant.

In the mode „Internal position measurement“ an internal position counter is used to count the input steps or the incremental encoder signals.

The connection of an incremental encoders to the axis controller is made according to Tab. 2.13 (The pin allocation of the peripheral connector -P2 is in the appendix).

Designation	Signal name	Signal level
Encoder inputs	Ch_A(1..4) /Ch_A(1..4) Ch_B(1..4) /Ch_B(1..4) Ch_Z(1..4) /Ch_Z(1..4)	TTL (differential or unipolar)

Table 2.13 : MAC4-STP: Inputs of Position Counter

The index pulse of the incremental encoder is generally monitored per interrupt. However during positioning moves the stepper pulse outputs are monitored by the interrupt inputs to ensure a high positioning accuracy; no output pulse is lost.

2.4.3 Input Signals

Various switches can be connected via the peripheral connector -P2 (see Tab. 2.14). The emergency stop signal must be connected. The signal levels of all other switches are configurable (see Section 4).

Designation	Signal name	Signal level low active on	Signal level low active off	Signal level high active on	Signal level high active off
Positive limit switch	/IESP(1..4)	0 V	5/24 V	5/24 V	0 V
Negative limit switch	/IESN(1..4)	0 V	5/24 V	5/24 V	0 V
Reference switch	/RS(1..4)	0 V	5/24 V	5/24 V	0 V
„drive fault“	/DF(1..4)	0 V	5/24 V	5/24 V	0 V
Emergency stop	/STOP	0..12 V	12..24 V	-	-

Table 2.14: MAC4-STP Switch Signals

The emergency stop signal is effective on all axes simultaneously. Activation of the emergency stop switch causes all axes to be disabled immediately.

⇨ It must be ensured, that the hardware limit switch remain active over the complete forbidden range. There is no protection against movement in the forbidden direction, if the axis overruns a hardware limit switch and comes to halt behind the switch.

The setting of the positive signal levels of the switches (5 V or 24 V) is made for each signal by using the solder bridges BR5-20 (Diagram2.8).

2.4.4 Fault Signals

The MAC4-STP can interrogate three fault code signals (FC-Signals) via the peripheral connector -P2. These codes can be made available to the host system for all axes as joint status information.

These inputs can also be used to interrogate other user signals.

Designation	Signal name	Signal level

Error code	FC(0..2)	TTL

Table 2.15: MAC4-STP: FC-Signals

2.5 Front Panel

There is a 7-segment display at the front panel of the axis controller (Diagram2.9), including specific LED-displays, a reset button and a connector for the serial communication interface.

2.5.1 Reset

After a reset of the axis controller all output signals return to the default states given in tables Tab.2.2, Tab.2.6 or Tab.2.12. In addition the velocity command register is reset, i.e. no signals are sent to the motor driver.

2.5.2 7- Segment Display

The 7-segment display indicates the operating condition of the axis controllers by displaying a hexadecimal number between $0 and $F (Tab.2.16).

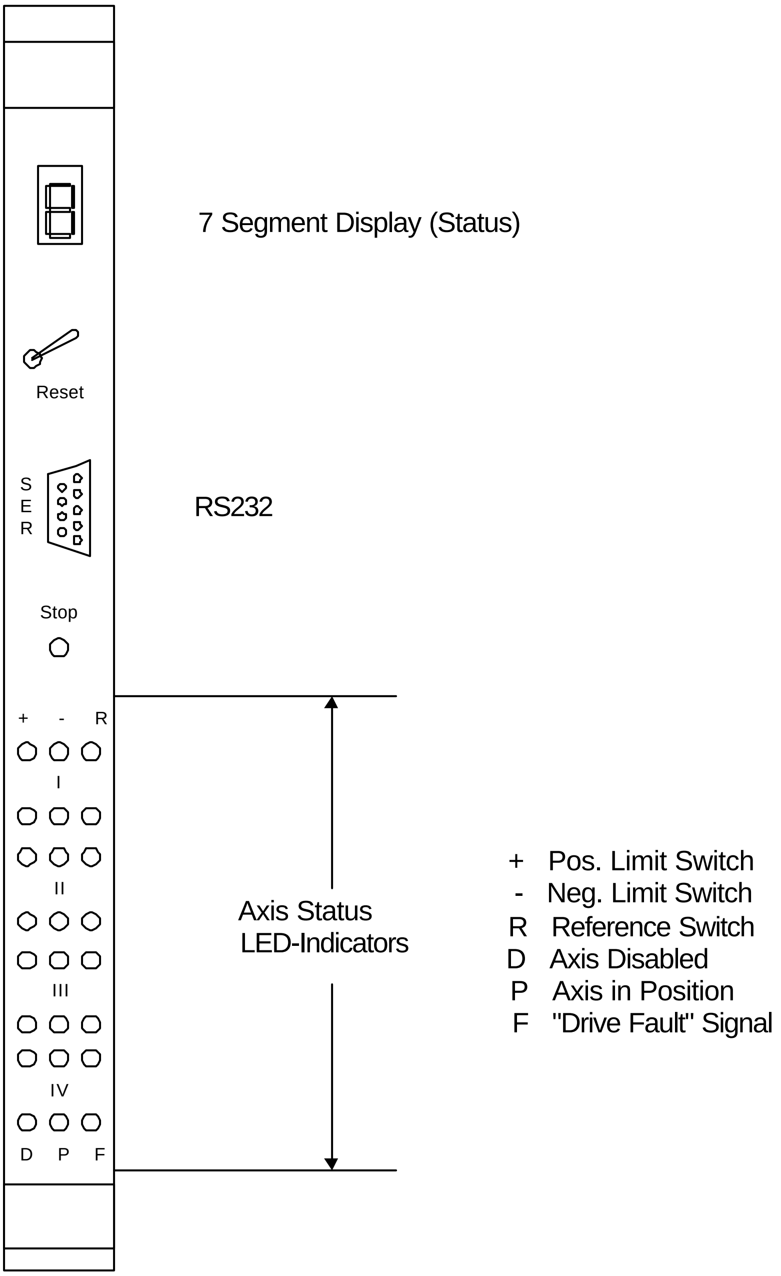


Diagram 2.9: Front Panel

Display	Meaning
$0	axis controller operational
$1	basic initialisation of the CPU performed
$2	RAM-test
$3	calculation of the CR check sum for the EPROM
$5	axis controller ready for communication
$7	limit value error
$8	logical error
$9	syntax error
$C	error of the position measurement system

$D	axis disabled by „watchdog“
$E	hardware emergency stop condition

Table 2.16: 7-Segment Display

2.5.3 LED- Display

The hardware emergency stop is effective for all axes and is displayed by the LED "STOP".

The display of the switch positions and of the axis operating conditions is made by six LEDs per axis (see Diagram2.9).

The LEDs for the limit switch and the reference switch light up, when the signal level drops to 0 V.
They are disabled or extinguished at a signal level of 5 V (MAC4-STP) or 24 V.

The LED "D" is turned on when the axis is disabled. It is turned off when the axis controller is instructed to go in to the "enable" mode.

During a position move, the LED „P“ indicates that the target position has been reached. The LED turns off on changing to another mode or commanding a new target position.

The user can modify the polarity of the "drive fault" signal according to his application. However the LED always turns on, when the system recognises the signal condition as "switch on".

3. Communications

The axis controller executes instructions, that must be transmitted to it in sequence. Communication can either be via:

- the VMEbus or
- the serial communication interface.

Communications are bidirectional. Each command transmitted is acknowledged by the axis controller.

3.1 Address Range of the DPRAM

Communications between the host and the axis controller is made over a special address range in the DPRAM. Access to other address can lead to an unexpected reaction of the axis controller.

Addressing is made in hexadecimal form without including the base address defined by the switches [S1] and [S2].

The axis controller uses the range of $3000 to $FFEE in the DPRAM. The range of $0000 to $3000 is reserved for local IRQ-vectors, monitor- and debugger variables.

The axis controller is based on a Motorola CPU. A 32-bit-number in the DPRAM is used according to Diagram3.1 to define the byte sequence:

Address	→	Byte HIGH
Address + 1	→	Byte MID-HIGH
Address + 2	→	Byte MID-LOW
Address + 3	→	Byte LOW

Diagram 3.1: Byte Sequence

Axis	Address	Content
1	$3000	internal axis data
	$3308	flag-field (host)
	$330A	counter content of an external position measurement system (host)
2	$3400	internal axis data
	$3708	flag-field(Host)
	$370A	counter content of an external position measurement system (host)
3	$3800	internal axis data
	$3B08	Flag-Field (Host)
	$3B0A	counter content of an external position measurement system (host)
4	$3C00	internal axis data
	$3F08	flag-field (Host)
	$3F0A	counter content of an external position measurement system (host)
1-4	$4000	internal axis data
1-4	$4600	stop bits

Axis	Address	Content
1-4	$4700	command buffer-queue-administration
	$4718	queue base address
1-4	$4800	output buffer-queue-administration
	$4818	queue base address

Table:3.1 Address Allocation in the DPRAM

3.2 Communication via Queues

Communication with the axis controller is made via the command- and output buffer in the DPRAM. Both are implemented and structured as ring storage - so called queues.

A queue consists of a queue-administration and a queue-base address. The elements of the queue (included in queue-base address) are numbered in sequence. For administration purposes two pointers are used, which point to the start and end address of the actual queue contents.

The start address of the command- or output-queue in the DPRAM is fixed (see Tab 3.1). The administration structure for the queue is held at this address, the so called queue-header. The actual queue begins after the header with an offset of 24 byte ($18) to the start address. Each queue can accommodate a maximum of 10 messages.

The basic structure of the queue is shown in Diagram3.2.

Offset to queue-Base address $18	Write-index	Read-index	free	Element-size 10 Byte	Number of elements 10

Element 1		.....		Element 10

Diagram 3.2: Queue-Structure

A detailed description of the queue is included in the appendix.

3.3 Communication via the Serial Interface

The serial connection (SUB-D, 9pin) at the front panel can be used for communication with the axis controller. The transmission rate is fixed at 4800 Baud (one stop bit, no parity).

The serial interface is served by a software driver, which transfers messages to the queue. From there they are read by the axis controller and processed accordingly.

The protocol employed is based on BSC-protocol (binary synchronous communication) IBM GA 27-3004-2.

Each telegram consists of three component parts (Diagram3.3).

Telegram header 2 Byte	Data n Byte	Check sum 1 Byte

Diagram 3.3: Structure of a Serial Telegram

Telegram Header
The telegram header (Diagram3.4) introduces a new telegram and begins with DLE (Data Link Escape, $10), followed by a STX (Start type of TeXt, $02).

DLE 1.Byte	STX 2.Byte

Diagram 3.4: Structure of the Telegram Header

Data
The data field (Diagram3.5) includes the axis number, the instruction code and possible parameters for the axis controller. This dictates the length of 7 bytes. If a byte with the value $10 (DLE) has to be transmitted, this character must be repeated and the length of the data field increases by one byte.

Axis number	Instruction code	Parameter
1. Byte	2.-3. Byte	4.-7. Byte

Diagram 3.5 :Structure of the Data Fields (excl. transmitted DLE)

Check sum
A check sum is added at the end of the serial telegram. This check sum is generated by adding the contents of all bytes of the data fields, with the exception of the repeated DLE-character. The result is truncated to one byte, i.e. the more significant bytes are ignored. If the check sum does not agree with the contents of the telegram, the receiver responds with a NAK-character ($15, Negative AcKnowledgement).

3.4 Message Structure

Parameters and modes of the axis controller are set by means of commands, that are acknowledged by the axis controller. Commands and responses possess the same structure and are designated in the following as messages.

A message (Diagram3.6) consists of a message header (Diagram3.7), a message parameter and a definition of origin.

Message header 4 Byte	Message parameter 4 Byte	Source 1 Byte	empty 1 Byte

Diagram 3.6: Structure of a Message

Axis number 1 Byte	empty 1 Byte	Instruction code 2 Byte

Diagram 3.7: Structure of the Message Header

The empty bytes are needed for reasons of portability.

The axis controller counts the axes 0...3. In this documentation the axes are counted from the number 1. When communicating with the axis controller, 1 must be deducted from the axis number.

All messages are entered in the queue exactly as stated.

If communications are made via the serial interface, the interface software driver generates the structure. When communicating via the VMEbus, i.e. by means of a direct write access to the DPRAM, the user must ensure that this structure is observed.

The contents of the field "Source" indicate to the axis controller, via which interface the command was received, i.e. where the response must be sent to. When accessing via the VMEbus the user must take care that a zero is put in the "Source" field. In all other cases the interface driver enters the "Source".

Commands messages usually include parameter values, which are associated with the instruction code. If the command requires no parameters, the remaining bytes for the message parameter remain empty.

The response of the axis controllers include the instruction code of the command received. The user status is included in the message parameters (see Section 6.). If the axis controller received a read command, the recognised value is repeated in the message parameters instead of the user status.

In the case of an error the appropriate error code is included in the response of the axis controller in place of the instruction code, the user or general status is included in the message parameters (see Section 6).

3.5 Direct Access to the DPRAM

There are two further possibilities of accessing the controller directly over the VMEbus via the DPRAM. These possibilities are in addition to communication via the queue.

3.5.1 Software Stop

The instruction "brake" (see Section 5) serves to bring the axis to a stand-still. As this command is entered in the queue and is not immediately executed, there is a possibility, to stop the axis immediately per software. The user must set the corresponding stop-bit in the DPRAM (at address $4600, Tab.3.2).

The stop-bits are reset by the axis controller.

The software stop does not disable the axis; instead it executes the mode "brake".

If the software stop is initiated in the mode "test", the axis is disabled (mode "disable").

Bit	Axis
1 set	Stop axis 1
3 set	Stop axis 2
5 set	Stop axis 3
7 set	Stop axis 4

Table 3.2: Stop by Means of Software

3.5.2 External Position Information via the DPRAM

When the position information is made available via the DPRAM, the host must write this information directly in the DPRAM of the axis controller.

The addresses for the entry of the position information are fixed (see Tab.3.3) and must be identified by the parameter *<ea>* in the initialisation phase. Further explanations regarding the configuration of the position measurement system can be found in Section 4.

There is an interlocking flag in order to avoid access conflicts between host and axis controller when entering the position information in the DPRAM.

Axis	Address of the position information	Address of the interlocking flag	Bit number of the interlocking flag
1	$330A	$3308	15
2	$370A	$3708	15
3	$3B0A	$3B08	15
4	$3F0A	$3F08	15

Table 3.3: DPRAM Addresses for the Position Information

The length of the position information is always 4 bytes. The flag-field has a length of 2 bytes.

Example
The position information for axis 1 has to be supplied by the host.

1. The host checks, whether the interlocking flag at address $3308 is set to $0000. If this is the case, the axis controller has already read out the current position information.

2. Now the host writes the new value for the position information to address $330A of the DPRAM. In this case the base addresses set with the switches[S1] and [S2] of the IPC have to be observed. Afterwards the host sets the interlocking flag at addresses $3308 to $8000.

3.The axis controller reads the value of position information at the address $330A and resets the interlocking flag to $0000.

4. System Settings

After switch-on or reset the axis controller executes a basic initialisation, in which all of the parameters are set to their default values. The axis controller is operational when:

- A "5" is shown in the LED status display and
- The copyright statement "MACCON ©1993" appears in the output queue at address $4818.

The command interface is now active.

The axis controller must be configured and initialised before actual operation (axis initialisation). The configuration process includes:

- MAC4-STP: setting of motor and drive parameters,
- setting of the motor direction of motion,
- setting of the axis type and definition of the axis parameters,
- description of the position measurement system,
- identification of switches and their settings,
- entry of interrupt vectors,
- setting of parameters for the operating modes.

All non-configurable parameters retain their default values.

After completion of the configuration, each axis of the controller must be initialised with the instruction INIT. Operation can first begin after the initialisation.

4.1 Motor and Drive Parameters for the MAC4-STP

The parameters of the connected stepper drives and their configuration must be communicated to the axis controller.

4.1.1 Step Number of the Motor

The step number of the motors indicates how many steps it executes per revolution. This number is stored in the parameter *<ms>*.

The step number of the motor is an important factor, for the conversion of units in position measurement.

The setting is made using the command UWMS (see Reference Manual, URMS).

4.1.2 Start/Stop Frequency

The start/stop frequency *<ssf>* indicates, what maximum velocity jump the motor can follow from stand-still without loosing steps or from what velocity the motor can be brought to an abrupt halt. This value is dependent on the motor load.

Velocities above the start/stop frequency can only be reached using the acceleration ramps. When stopping braking ramp is used.

The setting is made by means of command WSSF (see Reference Manual RSSF).

4.1.3 TURBO-Signal

A distinction is made between two operating conditions - normal- and TURBO-operation - in order to achieve a greater dynamic velocity range.

If the axis is in the TURBO-mode, velocities are multiplied internally by the factor 10. This property must only be allowed for when generating user velocity profiles in the modes "velocity tracking" and "position tracking“.

The possible velocity range in the operational modes normal-/TURBO-operation are summarised in Tab 4.1.

	TURBO	Normal
Lowest frequency	0 Hz	0 Hz
Highest frequency	500 kHz	50 kHz
Frequency steps	7.629 Hz	0.7629 Hz
Length of output pulses	1 μs	10 μs

Table 4.1: Normal- and TURBO-Operation

Only the frequencies, that lie exactly on the frequency steps (i.e. 0.7629 Hz, 1.5258 Hz etc. in normal operation), are generated precisely. All other values are subject to rounding effects with a minimum of modulation, which is however practically insignificant.

The setting is made by means of the command TURBO (see Reference Manual). The bit "Turbo" in the system status is set or deleted accordingly.

4.1.4 BOOST-Signal

The BOOST-signal is generated before each motor move and cleared during stand-still. The application of the signal increases the torque output of the stepper motor during the critical acceleration and deceleration phases of movement. The bit "boost" is set in the system status. The signal must be generated for a period prior to motion depending on the drive (Diagram5.2). This time is variable and can be set by means of the parameter *<bt>*.

The setting is made by means of the command WBT (see Reference Manual RBT).

4.2 Motor Direction of Motion

If the motor is correctly connected, the motor turns in a clockwise direction when a positive velocity values is output. This sense of orientation can be reversed by means of the instruction USR.

Using this instruction the direction of motor motion can be synchronised with direction of counting of the position measurement system, i.e. the output of a positive velocity value causes the actual position value to increase.

The initialised condition of motor direction orientation is re-established by repeating the instruction USN.

The instruction USR has no effect in the mode "test".

> ⇨ The correct motor-/encoder connections should be established during the commissioning phase. Otherwise the controller may respond falsely on reaching the hardware limit switches.

4.3 Axes

4.3.1 Axis Types

The following axis types can be set up:

- linear axes
- circular axes (360°)
- circular-optimised axes (360°)

This axis type is identified by the parameter *<sh>*.

4.3.1.1 Linear Axes

Linear axes are limited by the parameter *<ln>* in the negative direction and *<lp>* in the positive direction (Diagram 4.1). The positive software limit switch must be above the negative limit:

<lp> > *<ln>*.

Hardware limits are defined by the limit switches.

All target positions must lie between the range defined by *<ln>* and *<lp>* (total axis travel). If the axis is moved out of this range the motor is braked and the axis disabled.

These settings are made with the commands UWSH, UWLP, UWLN (see Reference Manual URSH, URLP, URLN).

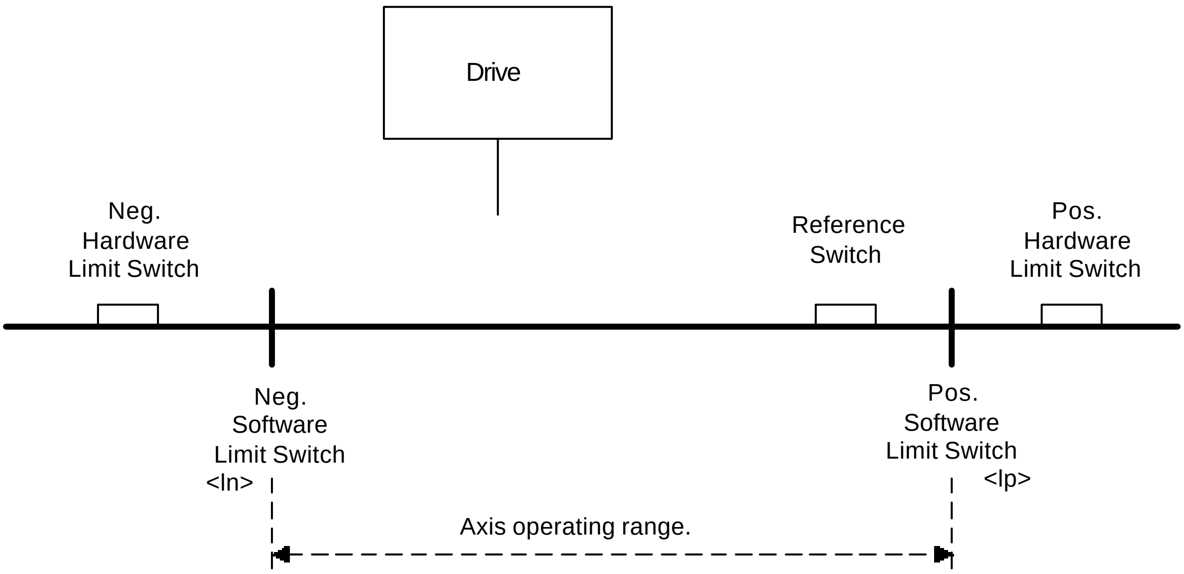


Diagram 4.1: Linear Axes

4.3.1.2 Circular and Circular-optimised axes

In the case of circular axes there are no limits that may cause the axis to be disabled (Diagram 4.2).

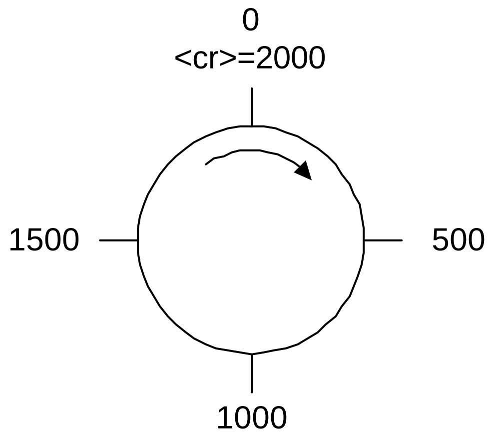


Diagram 4.2: Circular Axes

The travel of the circular axis (one revolution) is defined in the parameter *<cr>*. This value must be greater than zero:

<*cr*> > 0.

The position counter is set to zero when the position of the axis arrives at the value <*cr*> during motion in the positive direction. During movement in the negative direction the position counter is set to <*cr*>-1, when the position value drops below zero.

Circular-optimised axes differ from circular axes in that the shortest distance for an absolute move is automatically selected by the axis controller.

The settings are made with the commands UWSH, UWCR (see Reference Manual: URSH, URCR).

4.3.2 Axis Offset

The position information read normally identifies the position of the axis. This value may be modified with an offset, defined by the parameter <*ao*> ,axis offset, (see Diagram 4.3).

The position information processed by the axis controller is the sum of the actual measured position information and the axis offset.

The axis offset can also be set indirectly by allocating a user value to the actual position by means of the instruction WZP.

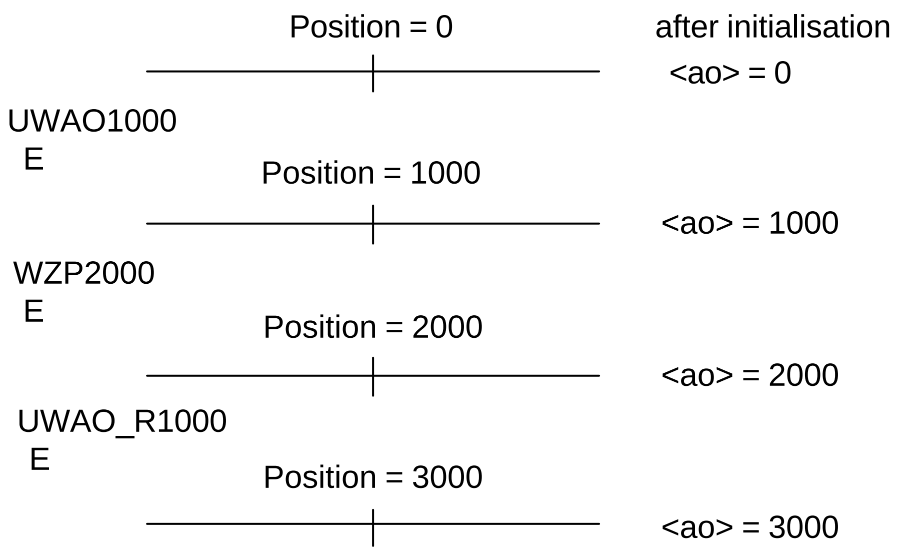


Diagram 4.3: Axis Offset

4.4 Position Measurement System

4.4.1 Types of Position Measurement

There are various possibilities of measuring the axis position:

Position measurement	INC	SSI	STP	Attribute
step counter			x	internal
incremental encoder at the peripheral connector -P2	x		x	internal
absolute encoder at the P2-connector in relative measurement mode		x		internal
absolute encoder at the peripheral connector -P2 in absolute measurement mode		x		internal
position sensor over the VMEbus in relative measurement mode	x	x	x	external
position sensor over the DPRAM in relative measurement mode	x	x	x	external
position sensor over the VMEbus in absolute measurement mode	x	x	x	external
position sensor over the DPRAM in absolute measurement mode	x	x	x	external

x ... position measurement available

Table 4.2: Types of Position Measurement

4.4.1.1 Step Counter

If there is no position measurement system connected, the output pulse counts generated by the axis controller are counted and used for the value of the actual position. The real position is not monitored by the controller. The position value is available in units of motor or driver steps. The solder bridges selecting the counting of the outputs steps must be closed for the individual axes (see Section 2)

4.4.1.2 Incremental Encoder at the Peripheral Connector -P2

An incremental encoder is directly connected at the peripheral connector -P2. The signals received from the incremental encoder are counted in the axis controller and are used to generate the value of position.

In the case of MAC4-STP the axis solder bridges for the counting of increments must remain open (see Section 2). The position value is now in units of the incremental encoder and must be converted to motor or driver steps. The ratio of steps/increment (<*ms*>/<*es*>) must be known.

4.4.1.3 Absolute Encoder at the Peripheral Connector -P2

An absolute encoder with SSI interface is connected via the peripheral connector -P2 directly to the axis controller. The position received from the sensor can be registered either as absolute or relative position information. In the case of relative registration only the change is position value since the last postion interrogation is processed. In the case of absolute position measurement the operating range of the absolute encoder may not be exceeded.

4.4.1.4 External Position Encoder

Any position measurement system of choice is connected to the host system. Position measurement is not made by the axis controller but by another module. The transfer of the position information to the axis controller is made either via the DPRAM (see Section 3) or the axis controller reads the measurement value over the VMEbus as a bus-master. The position read can either be processed in absolute or relative form.

The position value is now in units of the incremental encoder and must be converted in the case of MAC4-STP to motor or driver steps.

4.4.2 Configuration of the Position Measurement System

4.4.2.1 Address - and Range Settings

The following parameters are used to set the address - and range settings of the position measurement system:

<*et*>	encoder type
<*ea*>	encoder address
<*eb*>	encoder bit size
<*ec*>	encoder counter range

The type of position measurement system is defined by the parameter <*et*>.

The parameter <*ea*> indicates the address from which the axis controller should read the position information.

> ⇨ Before initialisation (INIT) it must be ensured, that plausible measurement values are present at these address, as the axis controller will immediately start processing these values after initialisation.

The counting range of the storage address, identified by the parameter <*ea*>, is defined by the parameter <*ec*>. This storage address can be the register of a counter or the register of an encoder with SSI-interface. In the relative counting mode this value is important to be able to recognise counter overflow. In the absolute encoder mode this value limits the axis range, values outside this range <*ec*> can be recognised as errors.

The width of the position information in bits is identified by the parameter <*eb*>. It indicates the number of significant bits of the values read. All higher value bits are ignored.

If the axis controller reads the position information externally over the VMEbus (*<et>* = 3,4), the parameter *<eb>* further defines, whether the axis controller accesses one or two 16-bit-words via the VMEbus.

When two 16-bit accesses are made it must be ensured that the measurement values does not change between them. The first 16-bit access reads the LSW (least significant word) of the position measurement value at the address *<ea>* + 2 and the second 16-bit-access reads the MSW (most significant word) at the address *<ea>*.

Tab. 4.3 summarises all the settings needed. The counter of the axis controller is used when output steps are counted or an incremental encoder is used at the peripheral connector -P2. All necessary information (address of the counter registers, counter range) are automatically generated during initialisation (INIT) and need not be set by the user.

These settings are made by the commands UWET, UWEA, UWEC and UWEB (see Reference Manual: URET, UREA, UREC, UREB).

Position measurement	*<et>*	*<ea>*	*<ec>*	*<eb>*	Value read
step counter	0	def.	def.	def.	relative position in steps
incremental encoder at the peripheral connector -P2	1	def.	def.	def.	relative position in increments
absolute encoder at the peripheral connector -P2 in relative counting mode	1	def.	def.	def.	relative position in increments
absolute sensor at the peripheral connector -P2 in absolute counting mode	2	def.	def.	def.	absolute position in increments
position sensor over the VMEbus in relative counting mode	3	VMEbus-address	value	value	relative position in increments
position sensor over the DPRAM in relative counting mode	3	DPRAM-address	value	value	relative position in increments
position sensor over the VMEbus in absolute counting mode	4	VMEbus-address	value	value	absolute position in increments
position sensor over the DPRAM in absolute counting mode	4	DPRAM-address	value	value	absolute position in increments

def. ... Default value (automatic entry)
Value ... Parameter value, within the permissible limits

Table 4.3: Settings of the Position Measurement System

4.4.2.2 Scaling of the SSI-Encoders (MAC4-SSI)

The protocol of the SSI-encoders must be specified, in order to be able to connect different SSI-encoders to the axis controller MAC4-SSI. The protocol is dependent on the manufacturer and can be identified in the encoder documentation. An example is shown in Diagram4.4

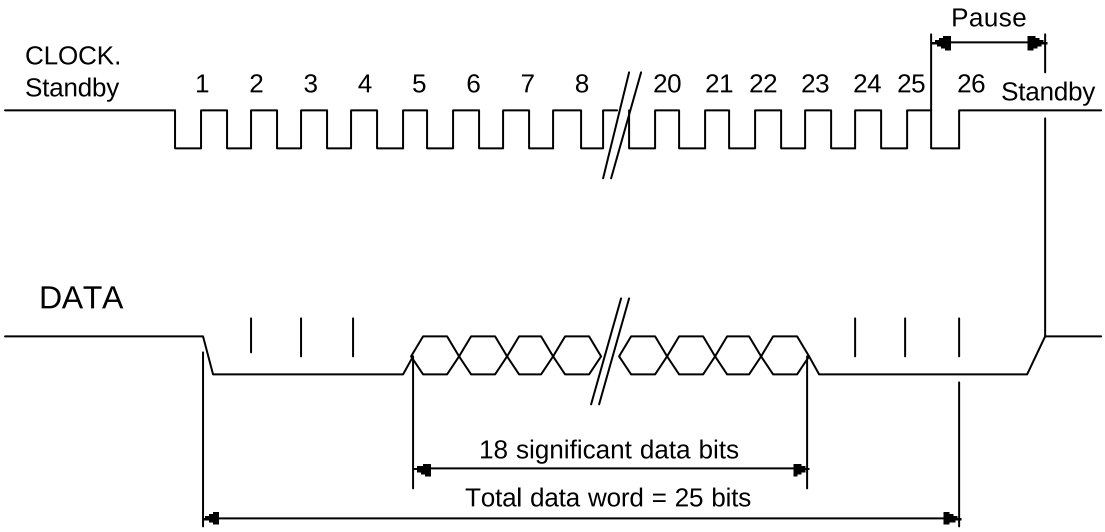


Diagram 4.4: Signals of the SSI

As a rule the axis controller does not read position information from the register with right-handed justification, so that a binary shift operation is needed before further processing. The axis controller can only process the position information when it is correctly justified. (see Diagram 4.5)

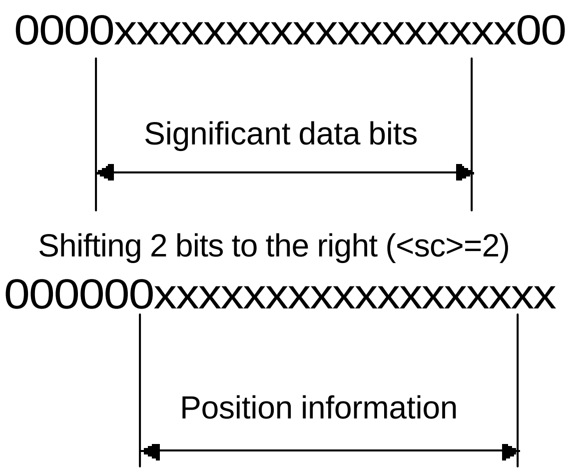


Diagram 4.5: Shifting of the Position Information for SSI-Encoder

The parameter <*sc*> indicates the number of the binary shift steps required. The setting is made by the command UWSC (see Reference Manual: URSC).

4.4.2.3 Units Conversion for MAC4-STP

The position units used by the MAC4-STP are motor or driver steps.

Position measurement systems that generate increments can be connected to MAC4-STP (see Tab. 4.3). The ratio of increments/step is needed for the conversion between the two frames of reference.

This ratio is defined by means of the parameter encoder resolution <*es*> and number of motor steps <*ms*>. The encoder resolution is defined as the number of encoder lines per revolution. The 90° phase difference between the A- and B-tracks allow the effective resolution to be multiplied by four. The number of lines (before quadrature multiplication) is entered in the parameter <*es*>. The axis controller multiplies this value by four internally.

4<*es*> is the number of increments that correspond to <*ms*> step.

⇨ Rounding errors can occur if the ratio of motor steps to encoder resolution in non-integral. As a guideline it should be ensured that the quotient *es/ms* and *ms/es* should not go beyond two significant bits behind the decimal point.

1st Example
A motor/driver with 25,000 steps/rev is directly connected to an incremental encoder with 1,000 lines/rev (after quadrature discrimination: 4,000 increments/rev). The following parameter values are set-up in the initialisation phase:

<*es*> = 1,000

<*ms*> = 25,000

The axis controller derives the following coefficients :

<*ms*> /(4 * <*es*>*)* = 25,000 / (4 * 1,000)= 6.25

or

4 <*es*> / <*ms*> = 4 * 1,000 / 25,000 = 0.16

There is no rounding error as there is no third figure behind the decimal point.

2nd Example
A motor/driver with 25,000 step/rev is connected with an external positioning system with 2.500 inc./rev. via a gearbox with a reduction of 5:1. After five revolutions of the motor the encoder has made just one. 1 motor revolution corresponding to 25,000 steps; 5 motor revolutions correspond to 125,000 steps. 1 encoder revolution corresponds to 2,500 increments = 625 lines. The relation for the conversion factor is:

<*es*> = 625

<*ms*> = 125,000

<*ms*> /(4 * <*es*>*)* = 125,000 / (4 *625) = 50

or

4* <*es*> / <*ms*> = 4 * 2,000 / 125,000 = 0.02

The settings are made by the commands UWES and UWMS (see Reference Manual , URES, URMS).

4.5 Switch Configuration per Software

Connections are available for each axis at the peripheral connector -P2 of the axis controller for:

- a positive limit switch,
- a negative limit switch,
- a reference switch and
- a "drive-fault"-signal (see Section 2)

The active signal level can be configured separately for each switch. The following parameters are available for this purpose setting of:

<*ps*> positive hardware limit switch
<*ns*> negative hardware limit switch
<*rs*> Reference switch
<*df*> "drive fault"-Signal

⇨ The configuration of the limit- and reference switches has no influence on the corresponding LED- display (see Section 2).

The settings are made by the commands UWPS, UWNS, UWRS and UWSO (see Reference Manual, URPS, URNS, URRS, URSO).

4.6 Interrupt Vectors

The axis controller is able to generate interrupts dependent on various events (see Tab.4.4). The interrupt vectors can be configured accordingly.

The interrupt generation can be turned off by choosing vector 0 (Default). Only axis controllers with interrupt vectors defined as greater than 0 actually initiate interrupts.

The interrupts 1-4 and 6 can be set up separately for each axis. The interrupt 5 is effective for all axes.

Nr.	Short designation	Description	Mnemonic
1	command output queue empty	The axis controller has read an instruction from the command buffer that came from the VMEbus.	RIRQ1
2	answer available	The axis controller has written a response to the output buffer to a command from the VMEbus.	RIRQ2
3	drive fault	The signal „drive fault“ is active.	RIRQ3
4	end of positioning	The actual positioning or justification is completed.	RIRQ4
5	emergency stop	The emergency stop switch has been activated.	RIRQ5
6	encoder value demand	A new position command value may be written to the DPRAM.	RIRQ6

Table 4.4 : Interrupts

⇨ Interrupt 2 is not generated, when the output queue is full. The response is lost.

4.7 Watchdog

A "watchdog", generated by software, surveys communications between the host and axis controller.

It disables all axes if the axis controller receives no instruction from the host during a period set in the parameter *<wd>*. This facility is to ensure that the axes do not go out of control if there is a fault in the host-system or in the connection between the host and axis controller.

The set value is valid for all axes.

4.8 Global Parameters

The following parameters are used in various modes.

4.8.1 Maximum Following Error

The following error is defined as the difference between the commanded position and the actual position of the axis. There may be several reasons why the axis does not reach the commanded position:

- cable breakage,
- the commanded velocity or acceleration may be too high for the motor,
- error of the position measurement system,
- MAC4-STP: the given value for motor steps <*ms*> or encoder lines <*es*> is false,
- MAC4-STP: the ratio of <*es*> to <*ms*> has led to rounding errors,
- bad mechanical connection between motor and encoder,
- the motor or encoder are incorrectly connected, i.e. in the positive motor direction of motion counts corresponds to a down count of the encoder,
- MAC4-INC/SSI: bad set of control parameters
- MAC4-SSI: false shift factor <*sc*>

The parameter <*mf*> indicates the size of the following error that may be tolerated. The axis is braked and disabled on exceeding the given value. The error message "Following error overflow" is generated.

> ⇨ MAC4-STP: A following error can also occur when counting position pulses internally. This effect results from the time delay between position measurement and command output. In this case the following error does not truly indicate an error in motor motion, only the generated steps to the motor/driver have been counted.

4.8.2 Emergency Braking

Emergency braking <*ed*> serves to stop the axis as quickly as possible. It is employed in the following cases:

- reset instruction,
- "watchdog"-error,
- software- or hardware-limit switch activated,
- following error overflow,
- signal "drive fault" active.

The emergency braking ramp is used in all other modes for the above exception conditions with the exception of the mode "test". In the „test“ mode the axis is stopped without a ramp.

4.8.3 Justification Velocity

The justification velocity <*lv*> serves for the final and exact positioning to switch transitions or to the encoder index pulse in the modes "search index", "home", "find edge" and "positioning".

The value should be chosen to be as low as possible, in order that the axis can stop as precisely as possible at the target position. This justification velocity may not exceed the start/stop frequency <*ssf*> in the case of MAC4-STP. If the start/stop frequency is reduced to a value below <*lv*>, this is then automatically reduced to the start/stop frequency value.

4.8.4 Scaling of the Command Signal

The axis controller only accepts integer values. Velocities with fractional values may be realised by binary shifting by the parameter <*sf*>. Scaling obeys the equation:

$<Parameter\sim scaled> = <Parameter> / 2^{<sf>}$

> ⇨ The output values for all modes (velocity-, acceleration- and braking values) can be scaled.

4.9 Axis Initialisation

The system- and parameter settings are activated by the INIT-command. This command acts on each axis individually. Parameters which are not defined by the user are set to their default values on initialisation.

When the internal position measurement system is set by the parameter *<et>*, the additional parameters *<ea>*, *<eb>* and *<ec>* are automatically generated.

These parameters, which are activated by the E-command have no effect prior to axis initialisation, as the controller and the profile generator are disabled. Exceptions are the axis independent parameters:

- watchdog *<wd>*
- vector for the emergency stop-interrupt *<irq_vec5>*

After basic initialisation the axis controller starts to monitor the emergency stop condition. All other connected switches are first interrogated after axis initialisation. This only happens, when no emergency stop condition is recognised.

The signals to the FC-inputs [0..2] are available immediately after basic initialisation. The same applies for the status display of the gray-/binary-jumpers on MAC4-SSI controllers.

Axis initialisation activates the position measurement. After this actual position values are available, that may be read by commands RCP or RCPI (MAC4-STP). Until then only the position value 0 is seen. A repeat initialisation causes the position value to be reset in the case of incremental position measurement. This must be allowed for when using the software limit-switches.

The axis controller responds to the INIT-command with the user status after axis initialisation has been completed. The "init"-bit in the user status is set.

The axis initialisation process can be reversed with the command DEINIT. The axis is then brought to a stand-still and deactivated("disable"). The axis controller clears the „init"-bit in the user status. The system parameters may now be changed.

5. Operating Modes

In addition to the modes listed in Section 1 there is the mode "reset", which strictly speaking is not a mode at all. The behaviour of the axis controllers in the various modes is defined by a differing number of parameters.

In addition the axis is influenced by the settings:

- "on-fly"   or
- "off-fly"

as well as the controller parameter in the servo mode.

The setting of a mode as well as changes of the parameter values only become effective after execution of an "execute"-command.

A motion mode can only be activated, when an axis is in the mode "enable" or "brake".

The axis controller reads the position, generates a command value and controls or monitors the motion of each axis within the set system sampling period.

Within each system sampling period a profile generator calculates the command value for each axis .

1. System sampling period Controller for the axes 1, 2, 3 and 4
   Profile generator for axis 1
   Command interpreter
2. System sampling period Controller for the axes 1, 2, 3 and 4
   Profile generator for axis 2
   Command interpreter
3. System sampling period Controller for the axes 1, 2, 3 and 4
   Profile generator for axis 3
   Command interpreter
4. System sampling period Controller for the axes 1, 2, 3 and 4
   Profile generator for axis 4
   Command interpreter

The following sampling periods are set up:

	MAC4-INC	MAC4-SSI	MAC4-STP
system sampling period	2.5 ms	2.5 ms	8 ms
generator sampling period	10 ms	10 ms	32 ms

Table 5.1: Sampling Periods

In the following the generator sampling period is simply called sampling period (SP).

5.1 Mode Reset

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

With a reset all axes are brought to a stand-still with the emergency braking ramp <*ed*> and are transferred to the mode "disable". All parameters are set to their default values.

If a reset is made in the mode "test", the axes are disabled without first being braked.

A reset is executed without the need for an "execute"-command. After a reset all axes are deinitialised.

This function is only guaranteed, when the jumper J11 on the IPC is set (Section 2)

5.2 Mode Disable

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The addressed axis is disabled and an output motor command of zero is generated.

5.3 Mode Enable

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The addressed axis is enabled. It is held at its actual position. Only after an enable can a motion mode be selected.

The axis remains enabled until it is transferred again to the mode "disable".

5.4 Mode Speed

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The axis moves with the velocity given by the parameter <*sv*>. This velocity is reached with an acceleration ramp defined by <*sa*>. Thereafter the axis moves continuously with the velocity <*sv*>.

The sign of the velocity <*sv*> determines the direction of motion of the axis .

5.5 Mode Brake

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The axis is brought to a stand-still with the brake ramp <*sd*>.

Thereafter the axis is held at the actual position.

5.6 Mode Positioning

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The axis moves to the target position defined with the parameter <*ap*>.

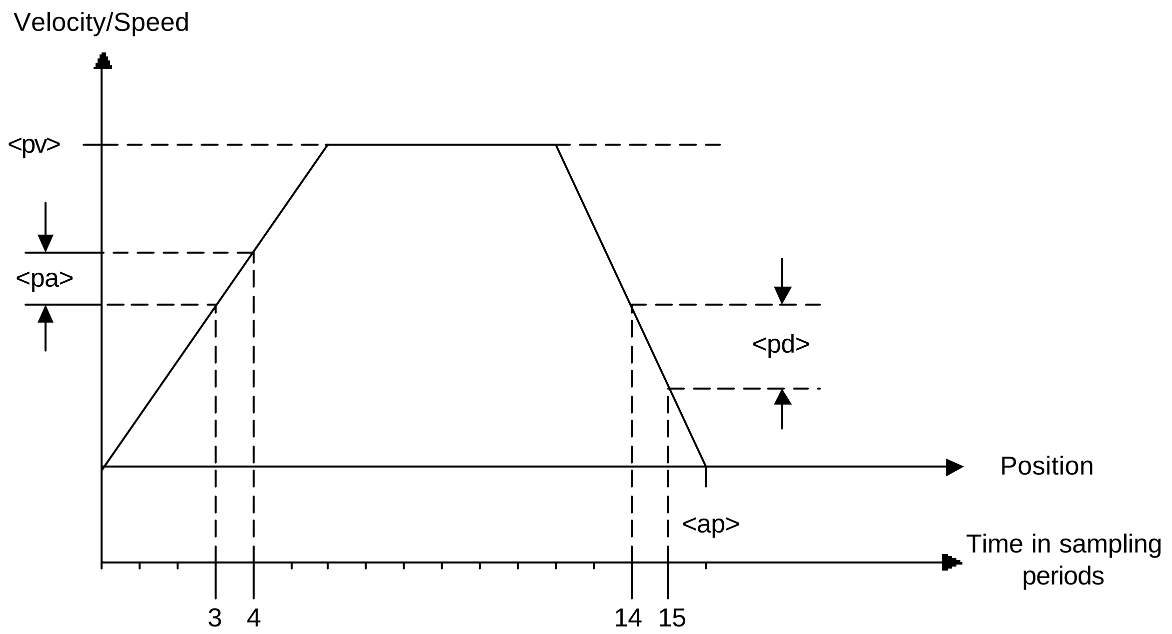


Diagram 5.1: Mode "positioning" (MAC4-INC/SSI)

The axis controller accelerates the axis with the acceleration *<pa>* to the velocity *<pv>*. Thereafter the axis moves with the velocity *<pv>* until it enters the braking phase. Braking is made with the braking ramp *<pd>*.

The velocity profile is trapezoidal. Depending on the velocity set, the acceleration and braking ramps as well as the move distance, the velocity profile may be triangular or rectangular.

In the case of MAC4-STP the braking ramp is followed by creep phase (Diagram 5.2). In the creep phase the axis moves *<stc>* steps with the justification velocity *<lv>*. Each pulse generated is counted by means of an interrupt. This ensures that the axis positions exactly to one step.

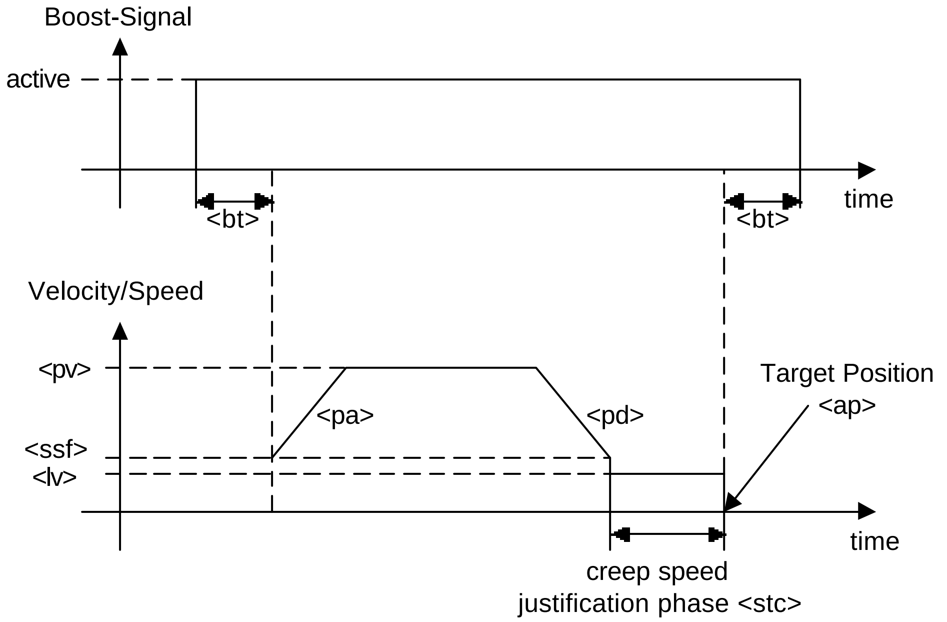


Diagram 5.2: Mode „positioning“ (MAC4-STP)

The direction of the motion depends on the target position.

On arriving at the target oscillations around this position must be avoided. A time variable recognition of the target serves to dampen oscillations and thereby stabilises the recognition of the end of the move (see Diagram 5.4).

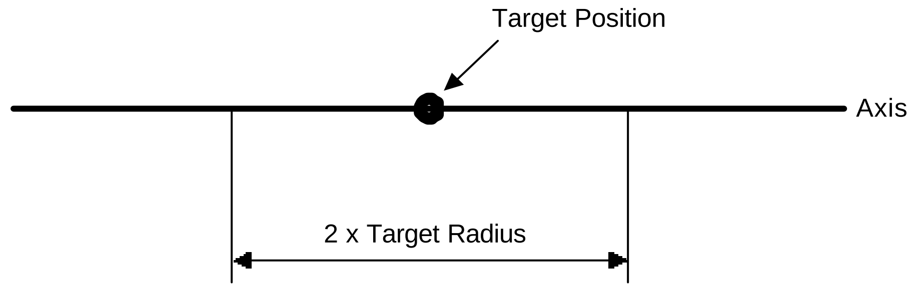


Diagram 5.3: Target Position and Radius

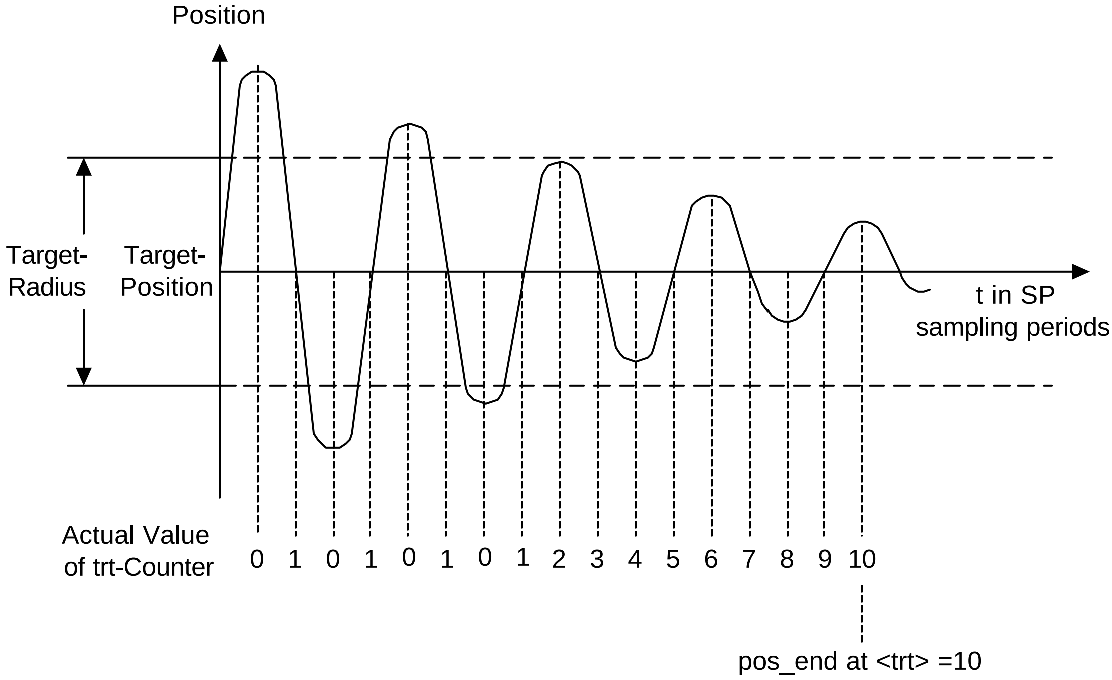


Diagram 5.4: Significance of <*trt*>

The parameter target radius <*tr*> puts a window around the target position <*ap*> (see Diagram5.3). After reaching the target position a target time counter is started. During each SP of the axis controller the actual position is compared with the target position. If the actual position lies within the target radius, the target time counter is increased by one. In all other cases the contents of the target time counter are reset. If the counter for the target time reaches the value given by the parameter <*trt*>, the axis controller recognises that the positioning move has been completed.

In the case of MAC4-STP tolerances resulting from the ratio <*es*> to <*ms*> are allowed for automatically.

For a target radius of zero the axis controller only recognises the end of the positioning move, when the actual position has been the same as the target position for the time duration <*trt*>. If the parameter <*trt*> is set to zero, the end of a positioning move is therefore recognised immediately on reaching the target position. <*trt*> can be set up in steps of the sampling period. The effective value is automatically adapted to this time frame through rounding up.

The end of a position move is indicated by the bit "pos_end" in the user status. The LED POS at the display of the axis controller is turned on. If the interrupt vectors "end of positioning" has been selected this is generated once at the VMEbus. The LED display and the bit "pos_end" are cleared only at the next "execute" instruction.

5.7 Mode Search Index Coarse

Axis controllers: MAC4-INC, MAC4-STP

The axis is approximately positioned to the next index pulse of the incremental encoder (Diagram 5.5)

⇨ These mode is only executable, when an incremental encoder is connected at the peripheral connector -P2.

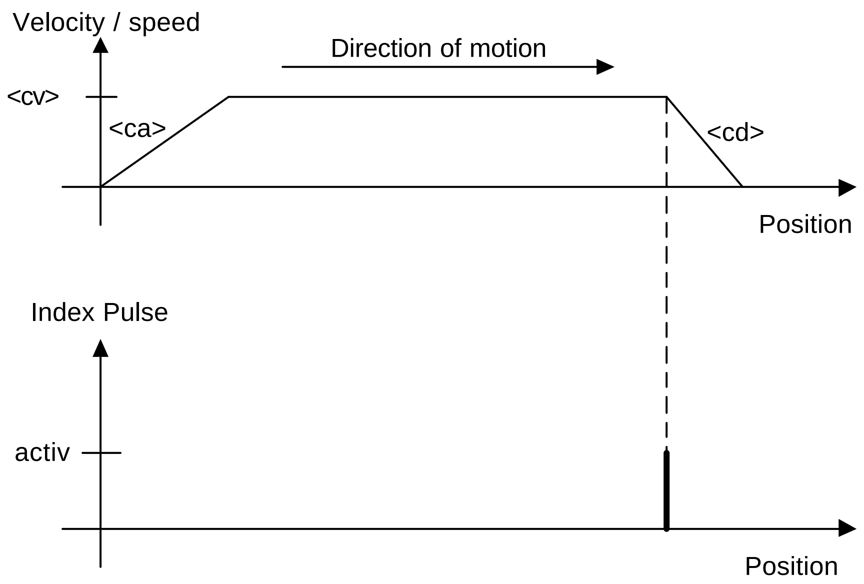


Diagram 5.5: Mode „search index coarse“

The axis controller moves the axis with the acceleration <*ca*> to the maximum permissible velocity <*cv*>. Then the axis moves with this constant velocity <*cv*> to the next index pulse. After registration of the index pulse a braking phase with the braking ramp <*cd*> follows. The axis is thus brought to a stand-still near to the index pulse.

The final deviation of the stationary axis position from the index pulse depends on the parameter settings.

The direction of motion is determined by the sign of the velocity <*cv*>.

5.8 Mode Search Index

Axis controllers: MAC4-INC, MAC4-STP

The mode „search index“ executes a fine justification (alignment) to the index pulse (Diagram 5.6)

⇨ These mode is only executable, when an incremental encoder is connected at the peripheral connector -P2.

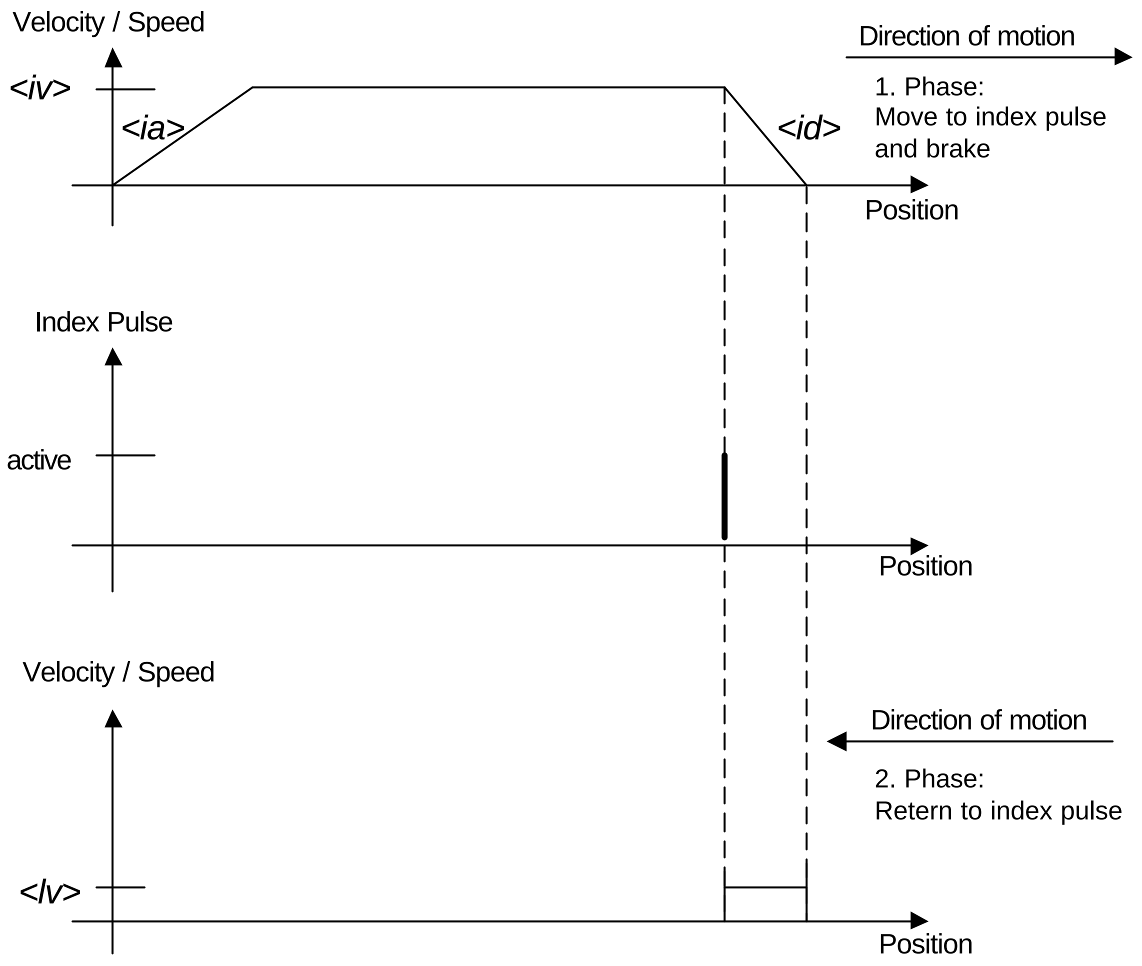


Diagram 5.6 Mode "search index " (MAC4-INC)

The first phase is the same as for the mode "search index coarse"; however "search index" uses acceleration *<ia>*, maximum velocity *<iv>* and braking ramp *<id>*.

In the first phase of motion the axis is driven over and beyond the index pulse. In the second phase it moves with the creep velocity *<lv>* back to the index pulse. On reaching the index pulse the axis is halted. A braking ramp is not generated.

The direction of motion depends on the sign of the parameter *<iv>*.

The measurement of the index pulse employs interrupts.

5.9 Mode Home

Axis controllers: MAC4-INC, MAC4-STP

The axis is positioned to the first index pulse following (as shown) the left active signal transition of the reference switch (Diagram 5.7)

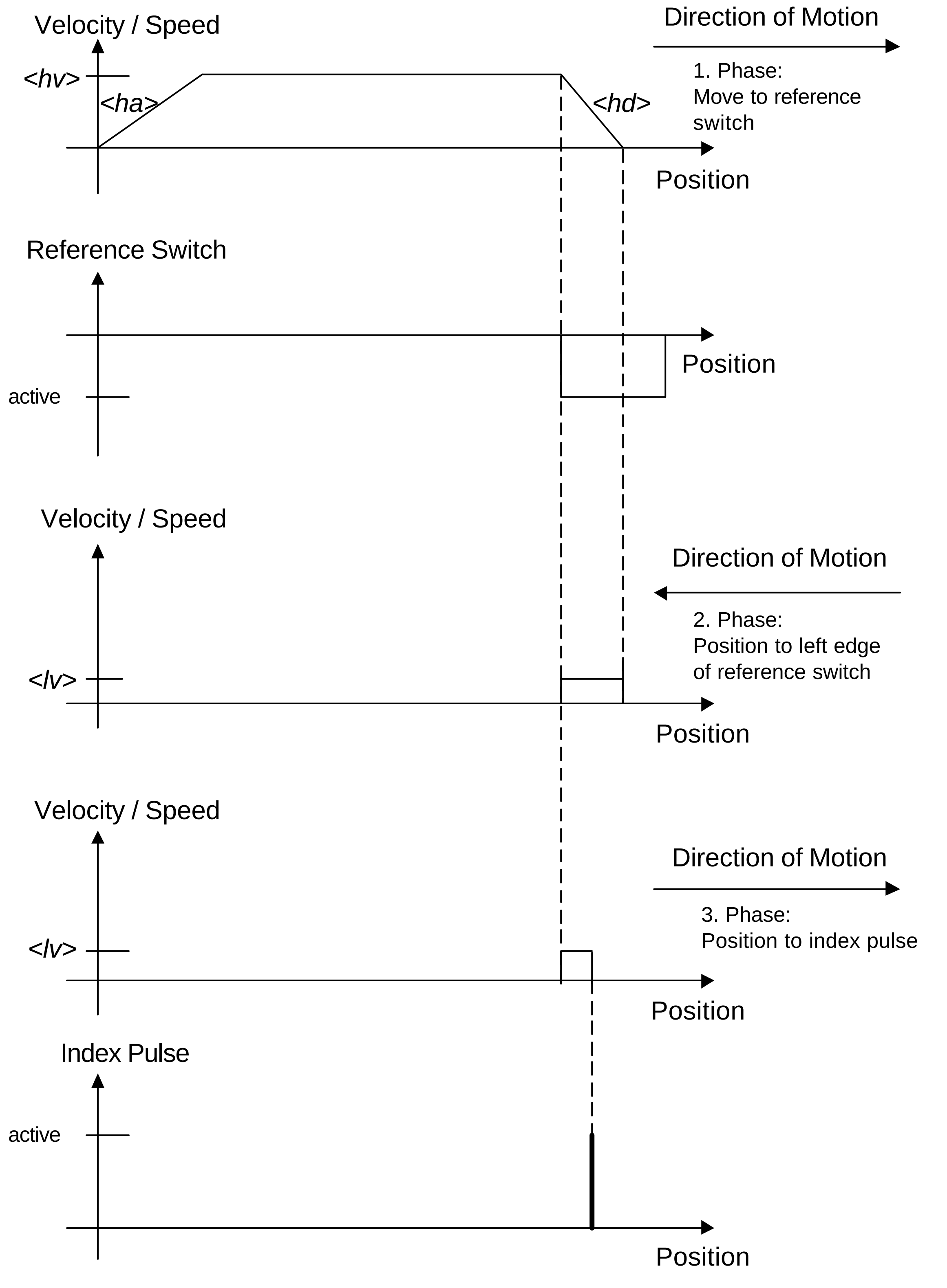


Diagram 5.7: Mode "home"(MAC4-INC)

> ⇨ These mode is only executable, when an incremental encoder is connected to the peripheral connector -P2.

In the first phase the reference switch is looked for with the acceleration <*ha*> and the velocity <*hv*>. The axis is then braked with the braking ramp <*hd*>. In the second phase the left signal transition of the reference switch is sought for. The left signal transition of the reference switch lies closest to the negative limit of axis movement.

Finally in the third phase the axis is positioned to the next index pulse in a positive direction. In the latter two phases the velocity <*lv*> is used. Braking from velocity <*lv*> occurs without a braking ramp.

The direction or search is determined by the sign of the velocity <*hv*>.

5.10 Mode Find Edge

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

With this mode it is possible, to justify (align) an axis to a switch defined by the parameter <*dl*> (Diagram5.8).

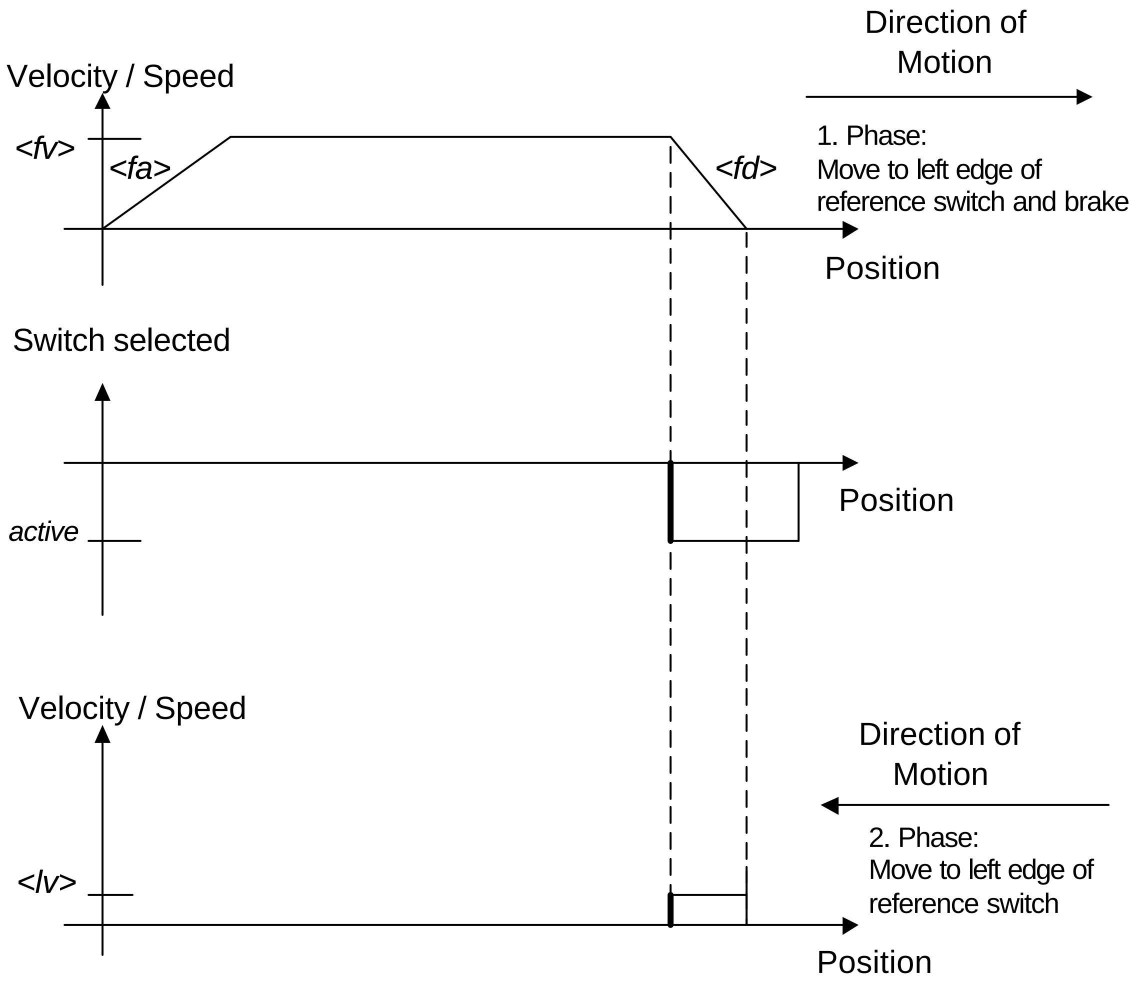


Diagram 5.8. mode „find edge“ (MAC4-INC)

The following switches can be selected:

<*dl*> = 0	reference switch
<*dl*> = 1	negative hardware limit switch
<*dl*> = 2	positive hardware limit switch

In the case of circular axes it is only possible to justify to the reference switch.

The mode "find edge" is executed in two phases. The axis controller accelerates the axis with the acceleration <*fa*> to the maximum permissible velocity <*fv*>. The axis moves with constant velocity <*fv*> to the switch transition and brakes with <*fd*>. The justification to the switch transition is made in opposite direction of motion with the velocity <*lv*>. Braking from the velocity <*lv*> is made without a braking ramp.

The direction of motion during a justification move to the reference switch is determined by the sign of the parameter <*fv*>. In the other configurations the direction of motion depends on the switch positions. On moving to the reference switch the axis controller justifies to the first recognised switch transition (Diagram 5.9)

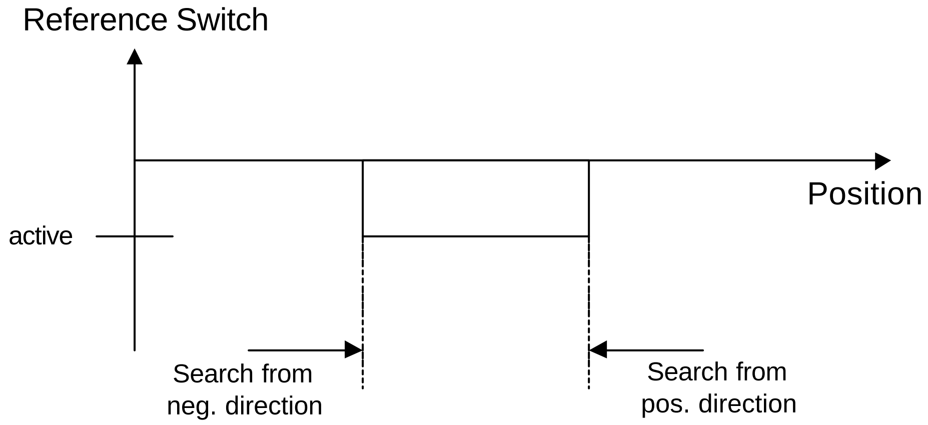


Diagram 5.9: Reference Switch Transition "find edge"

5.11 Mode Velocity Tracking

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The user can generate his own ramps in this mode. The velocity is defined by the parameter <*vt*> and can be changed at any time.

> ⇨ The acceleration is not monitored, it is the responsibility of the user to remain within the permissible acceleration limits.

5.12 Mode Position Tracking

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

The position that the axis should reach within the next SP is defined by the parameter <*at*>. This position is held until the parameter <*at*> receives a new value.

> ⇨ Velocity and acceleration are not limited, the user is responsible for ensuring that not too large position jumps are made.
>
> When the distance between target and actual positions becomes too large a following error overflow may occur. The axis controller generates a braking ramp depending on the distance. This also applies on recognising the limit switches, the signal "drive-fault", a communication or an encoder error .

5.13 Mode Test

Axis controllers: MAC4-INC, MAC4-SSI, MAC4-STP

In this mode it is possible to recognise whether the axis is correctly connected. To this purpose the axis is activated and the command value given in the parameter <*da*> is output to the motor.

The profile generator and the position controller (servomode) are not active in this mode.

The instruction USR has no effect in the mode "test".

⇨ If the axis controller recognises a hardware limit switch in the mode "test" the axis turns off ("disable"). Thereafter it can be returned to mode "test" and driven in either direction.

5.14 Special Characteristics of the Profile Generator in the MAC4-STP

The BOOST signal is generated before each motor move and deactivated after move completion. The MAC4-STP uses the start/stop frequency in all modes as the first and last acceleration value in a profile (Diagram5.2).

5.15 On-fly Operation

If the axis controller receives a new instruction it normally executes it immediately, even when the instruction currently being processed has not been completed. This behaviour is known as "on-fly"-operation.

Alternatively "off-fly" operation, in which instructions are strictly processed in series, can be selected in the modes

- "positioning"
- "search index coarse"
- "search index"
- "home"
- "find edge".

All other modes operate exclusively "on-fly".

Two examples illustrate the behaviour of the axis in "on-fly" operation:

1st Example
The axis moves in the mode "find edge" and looks for the positive hardware limit switch. During movement the user issues the instruction code for the mode "brake" and the "execute" command. In "on-fly" operation the axis switches to the mode "brake" immediately and stops without having found the limit switch.

2nd Example
Another axis is positioning to the target position 20,000. Currently it is at the position 0. The input and activation of the new target position 10,000 causes the axis to already stop at the new target, without have first moved to the position 20,000.

After deactivation of the "on-fly" operation by means of the command FLYOFF the axis controller processes each command in series.

The axis can then only be directly accessed and stopped via the stop-bits in the DPRAM (see Section 3).

The behaviour of the axes for the above examples changes as follows:

1st Example
The axis looks for the limit. It only transfers to the mode "brake" after the switch has been found.

2nd Example
The other axis moves to the position 20,000; only then does it accept the new command and starts to generate a ramp. It moves to the position 10,000 for the second time and stops there.

"on-fly" operation is activated with the command FLYON. The axis controller goes automatically to "on-fly" operation on switch on.

5.16 Position Controller of MAC4-INC/SSI

5.16.1 Controller

A Lead/Lag-Filter (with PD behaviour) and a parallel integrator (I behaviour) has been implemented in these controllers.

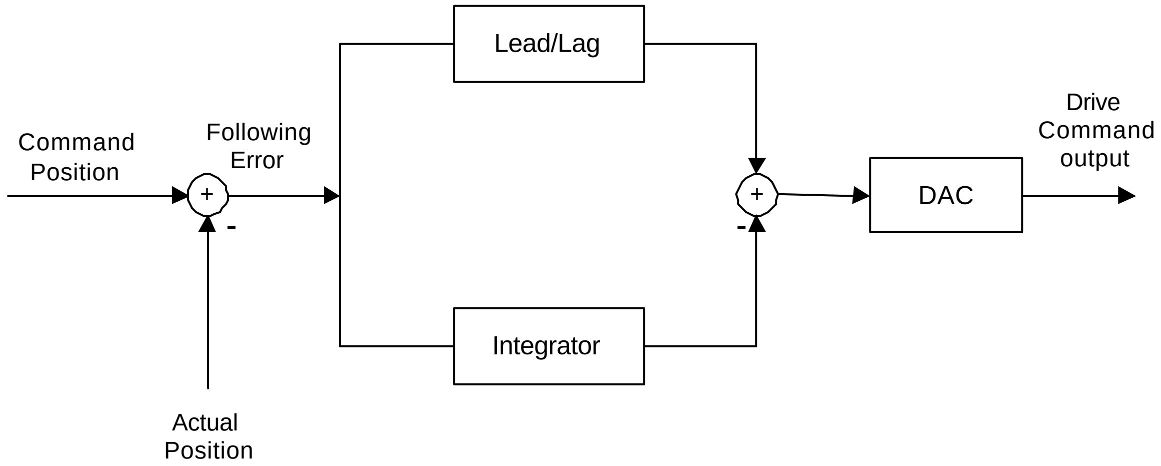


Diagram 5.10: Controller Structure

The basic controller structure is shown in Diagram5.10.

The z transfer function of the controller is:

Lead/Lag-Filter:

$$G_1(z^{-1}) = K\ (1-Az^{-1})/(1+Bz^{-1})$$

with the corresponding recursive algorithm:

$$u_1(k) = K * e(k) - K * A * e(k-1) - B * u_1(k-1)\ .$$

Integrator:

$$G_2\ (z^{-1}) = C\ (z^{-1})\ /\ (1 - z^{-1})$$

with the corresponding recursive algorithm:

$$u_2\ (k) = Ce(k-1) + u_2\ (k-1).$$

The output command value of the controller is given by the following expression:

$$u_R (k) = u_1 (k) + u_2 (k) .$$

where e(k) is the following error and u(k) the command value at the sampling time k.

The integral component of the controller is limited, to avoid excessive command values (anti „wind up“).

The controller coefficients can set by the parameters *<ga>* ("gain"), *<ze>* ("zero"), *<po>* ("pole"), *<ki>* ("integral gain"). These values are converted as follows:
.

$$<ga> = 4 * K$$

$$<ze> = 256 * A$$

$$<po> = 256 * B$$

$$<ki> = 256 * C.$$

In order to influence the integral amplification more exactly it is possible to shift the integral coefficient *<ki>* internally a number *<ki_sc>* of binary steps to the right. Each shift corresponds to a halving of the value:

$$\text{Integral factor} = <ki> /2^{<ki\_sc>}$$

Example: The actual integral amplification is 8 /(256*2*2) with *<ki>* = 8 and *<ki_sc>* = 2

⇨	The default value of *<ki_sc>* is 8.

The qualitative relationship between the values of the controller parameters and system behaviour is shown in Tab.5.2.

Increase of the parameter	Stability	Response time	Stiffness
„zero“	better	shorter	reduced
„pole“	less better	shorter	reduced
„integral gain“	worse	shorter	increased
„gain“	worse	shorter	increased

Table 5.2: Influence of the Controller Parameters

In setting of controller parameter the following procedure is recommended:

1.Increase "gain" until the motor starts to oscillate gently.

2.Reduce the "gain" parameter about 20%.

3.Reduce the "zero" during continuous motion until no further improvement in the following error can be recognised.

4.Again increase the "zero" value slightly.

5.The parameter "pole" can be set up in a similar way to the "zero". However this may reduce the stiffness of the system.

6.As the last step the integral amplification is set up as high as possible in order that the target position is reached as quickly as possible with minimum following error.

⇨ A too high integral component will cause stability problems.

5.16.2 Torque (Force) Limitation

The command signal generated by the controller is converted by the 12-bit out DAC and the motor amplifier into a current value in the motor, which generates a torque (or force). The torque is limited by the controller to the value *<lt>*.

A too strong limitation of the command value or torque may lead to an overflow of following error.

⇨ If the motor driver is set to voltage mode instead of current the command values output via the DAC are converted to a voltage at the driver output and thus to a velocity. The torque limit then has the significance of velocity limit.

5.17 Servomode

Axis controller: MAC4-STP

Stepper drives are usually controlled without a control loop. Step loss may occur and the motor target position then lost.

If a position measurement system at the motor is connected to the axis controller, it recognises the discrepancy and can correct it. The servomode is active after motion in the modes:

- "enable",
- "positioning",
- "search index coarse",
- "search index",
- "home",
- "find edge",
- "brake"

In the servomode the actual and target positions are compared after the time *<sw>* has lapsed. If the axis is not within the set tolerance band *<tr>* around the target position, a new position command is started. The waiting time *<sw>* becomes effective after reaching the target position.

If the axis lies within the target radius after the period *<sw>*, the axis controller turn off the "BOOST" signal. The axis resets the time value zero, and the time measurement is restarted. The basic function of the servomode is illustrated in Diagram 5.11.

The waiting period *<sw>* should be chosen such that oscillations of the motor-encoder system have time to damp.

The parameter *<tr>* must allow for the achievable step accuracy as well as the ratio of the position encoder resolution to motor steps.

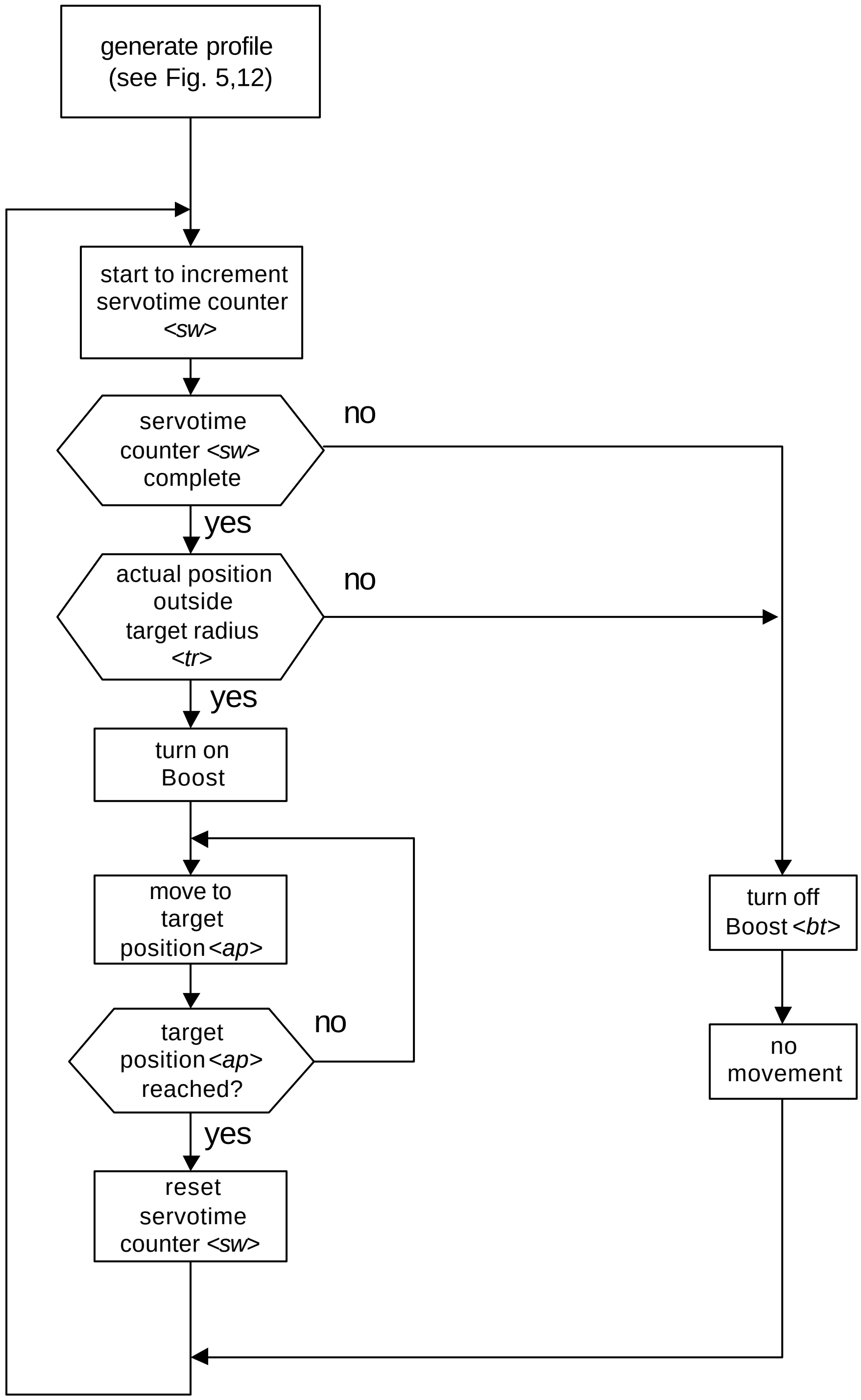
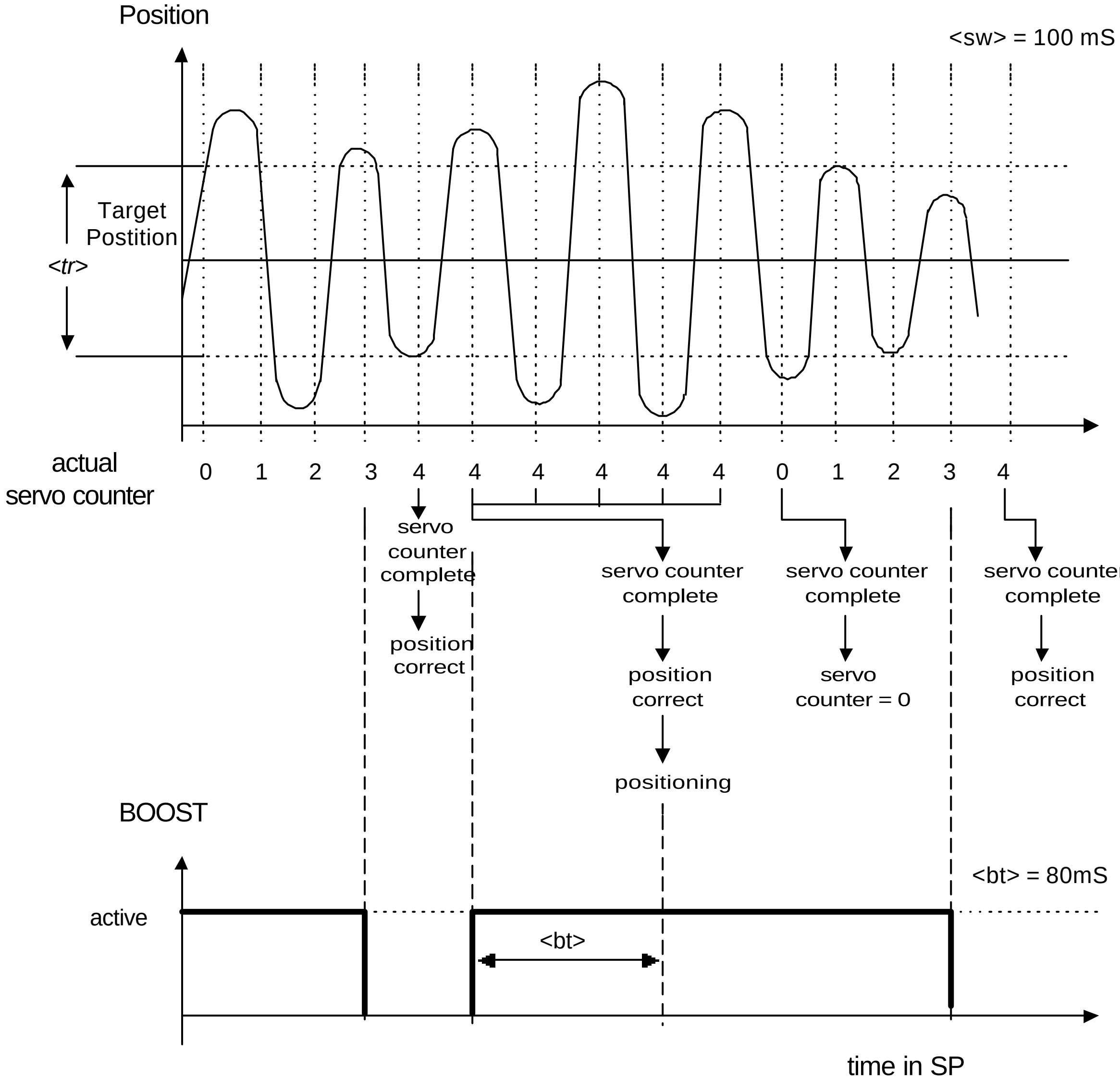


Diagram 5.11: Basic Functions of the Servomode



Significance of <sw>

Diagram 5.12: Significance of <sw>

6. Diagnosis and Monitoring

6.1 System Displays

Version

The hard- and software version of the axis controller can be read with the instruction VER.

A 32-bit value is returned with the version coded according to Tab. 6.1

Position	Meaning
Bit 0-7	software release number
Bit 8-15	software version number
Bit 16-23	hardware release umber (coded as a letter)
Bit 24-31	card type (0..INC, 1..SSI, 2..STP)

Table 6.1: Version Coding

System Data
The read instructions listed in Tab. 6.2. have been implemented to monitor motor motion. The direct output of the position measurement systems can be accessed as well as the actual position of the motor. This option allows the internal processing of the position measurement system to be checked by the user.

Abbrev.	Meaning	Units MAC4 -INC	Units MAC4-SSI	Units MAC4-STP
RAV	actual velocity	Inc/SSP	Inc/SSP	Steps/SSP
RCP	actual position	Inc	Inc	Steps
RCPI	actual position	*	*	Inc
RCT	processed counter content of the position measurement system (after bit suppression)	-	-	-
RED	value of the position measurement system counter (hardware counter)	-	-	-
RFE	actual following error	Inc	Inc	Steps
ROUT	output value	-	-	Steps/SP
RPS	command velocity	Inc/SP	Inc/SP	Steps/SP

SSP ... System sampling period
* ... Instruction does not exist
- ... no units

Table 6.2: System Data

Using the instruction RFE (read following error) it is possible to identify by how much the position of the motor differs from the target value. A detailed description of each instruction is provided in the Reference Manual.

6.2 Status Information

The operating condition of the axis controller can be read by using the following status commands:

Overall Status RAS	User status RBS	System status RSS
axis independent information	specific axis information for the user	specific information for internal control

Table 6.3: Status Information

A detailed description of status is provided in the appendix.

6.3 Fault Processing

If the axis controller recognises an error, it returns a different code to the received instruction (excepting read instructions). In addition an error dependent bit is set in the user- or overall status.
A reported error must first be acknowledged with the instruction C before the axis controller can accept further commands. Read commands are an exception, these are always executed.

6.3.1 False Axis Number

returned instruction code: 160
user status: bit "axis_f" set
cause of error: input of a false axis number
effect of error: the transmitted command is ignored.
the interrupt "command buffer empty" is not generated. The response with the error code 160 includes the falsely given axis number.
the interrupt "answer available" is generated with the interrupt vector of the last valid axis.
continued operation of all axes is only possible after an error acknowledgement (for any axis).

6.3.2 Syntax Error

returned instruction code: 161
user status: bit "syn_f" set
cause of error: the instruction code does not exist.
effect of error: the transmitted command is ignored.

6.3.3 Limit Value Error

returned instruction code: 162
user status: bit "limit_f" set
cause of error: the parameter value lies outside of the permissible limits.
the given target position lies outside of the software axis limits
the given position is outside of the value range of the absolute encoder.
effect of error: the transmitted command is ignored.

6.3.4 Drive Fault

returned instruction code: 163
user status: bit "dr_f" set
cause of error: the "drive fault" signal is active.
effect of error: the axis is braked with the emergency braking ramp *<ed>* and transferred to the mode "disable". In the mode "test" the axis is disabled without a braking ramp.
the error message is given as the response to the next instruction.

6.3.5 Hardware Emergency Stop

returned instruction code: 165
user status: bit "h_stop" set
cause of error: the emergency stop switch was activated.
effect of error: all axes are disabled (without braking ramp) and put in the mode "disable".
the error message is given as the response to the next instruction.

6.3.6 Watchdog Error

returned instruction code: 164
user status: bit "watch_f" set
cause of error: the set "watchdog"- time has expired without communications between host and axis controller.
effect of error: all axes are braked with the emergency braking ramp *<ed>* and are put in the mode "disable". In the mode "test" the axis is disabled without a braking ramp.
the error message is given as the response to the next instruction.

6.3.7 Logical Error

returned instruction code: 163
user status bit "logo_F"
cause of error: a system parameter has been set, although the axis is not deinitialised or in the mode "disable".
a justification to the index pulse has been commanded although an incremental encoder has not been specified and connected at the peripheral connector -P2.
an error has not been acknowledged with the "clear" command.
a change to a motion mode is only possible from the modes "enable" or "brake".
a circular axis cannot be justified to the limit switch.
the axis position can only be set, when the axis is in the stand-still condition.
the mode "find edge" was requested with a condition that was declared OFF for the switch.
the mode "home" was requested although the reference switch was declared OFF.
the parameter *<ec>* may not be set in circular axes using an absolute encoder.
effect of error: the transmitted command is ignored.

6.3.8 Error of Position Measurement System

returned instruction code: 163
user status: bits "encoder_f", "logo_f" set
cause of error the position value read from an externally connected position sensor or the SSI absolute encoder exceeds the value range given by <*ec*>.
effect of error: the affected axis is disabled (without braking ramp) and remains in the mode "disable".

6.3.9 Following Error Overflow

returned instruction code: 163
user status: bit „ferr_f“ set
cause of error: the difference between the actual position of the axis and the commanded position exceeds the maximum values <*mf*> allowed. The cause can be:

- a bad cable connection
- the value of velocity or acceleration is too large; the motor cannot follow.
- an error of the position measurement system,
- MAC4-STP: a false input value for the motor steps <*ms*> or the encoder lines <*es*>,
- MAC4-STP: a bad ratio of <*es*> to <*ms*>, that leads to rounding errors,
- bad mechanical connection between motor and encoder,
- a falsely connected motor or encoder, i.e. in the positive direction of motion the encoder counts downwards,
- MAC4-INC/SSI: insufficient controller parameters,
- MAC4-SSI: a false shift factor <*sc*>.

effect of error: the axis is braked with the emergency braking ramp <*ed*> and put in the mode „disable“.

7. Appendix

7.1 Status

7.1.1 User Status

Bit	Name	Meaning (Bit=1)
0-7	"mode“	mode (see below)
8	"dr_en“	axis enabled
9	"rev_o“	reversing of the output
10	"pos_end“	positioning completed, i.e. axis is in the target radius
11	"jus_end“	justification completed
12	"init“	axis initialised
13	"syn_f“	syntax error
14	"limit_f“	limit value error
15	"logo_f“	logical error
16	"encoder_f“	error of the position measurement systems
17	"dr_f“	"drive fault“ signal on
18	"ferr_f“	following error overflow
19	"hwl_n“	negative hardware limit switch on
20	"hwl_p“	positive hardware limit switch on
21	"swl_n“	negative software limit switch on
22	"swl_p“	positive software limit switch on
23	"ref“	reference switch on
24	"free“	not used
25	"on_fly“	"on-fly“ operation on
26	"free“	not used
27	"gray“	MAC4-SSI: SSI sensor with Gray-code or not used
28-31	"free“	not used

Mode Codes

Code	Mode	Code	Mode
0	"disable“	7	"brake“
1	"enable“	8	"positioning“
2	"search index coarse“	9	"velocity tracking“
3	"search index“	10	"test“
4	"home“	11	"positioning tracking“
5	"find edge“	12	"reset“
6	"speed“		

7.1.2 System Status

Bit	Name	Meaning (Bit=1)
0	"m_rdy“	command operational
1	"phase1“	internal control bit
2	"phase2“	internal control bit
3	"phase3“	internal control bit
4	"phase4“	internal control bit
5	"z_irq“	internal control bit
6	"z_stop“	internal control bit
7	"init_weg _sys“	synchronisation bit
8	"in_tr“	MAC4-STP: internal control bit, otherwise not used
9	"servo_on“	MAC4-STP: internal control bit, otherwise not used
10	"dr_disable“	MAC4-INC/SSI: interlocking bit
10	"direction“	MAC4-STP: motion in negative direction
11	"boost“	MAC4-STP: BOOST signal on, otherwise not used
12	"boost_en“	MAC4-STP: BOOST signal permits move, otherwise not used
13	"turbo“	MAC4-STP: TURBO operation is turned on, otherwise not used
14	"irq-line“	MAC4-STP: internal control bit, otherwise not used
15	"in_pos“	MAC4-STP: LED ´P´on, otherwise not used
16	"led_fault“	MAC4-STP: LED ´F´on, otherwise not used
17	"phase5“	MAC4-STP: internal control bit, otherwise not used
18	"boost_lock“	MAC4-STP: BOOST signal , otherwise not used
19	"phase6“	MAC4-STP: internal control bit, otherwise not used
20	"bits_valid“	MAC4-STP: internal control bit, otherwise not used
21	"wzp_bit“	MAC4-STP: internal control bit, otherwise not used
22	"dr_disable“	MAC4-STP: internal control bit, otherwise not used
23	"brake_stop“	MAC4-STP: internal control bit, otherwise not used
24	"phase7“	MAC4-STP: internal control bit, otherwise not used
25-31	"free“	not used

7.1.3 Overall Status

Bit	Name	Meaning (Bit=1)
0	"h_stop“	hardware-emergency stop
1	"axis_f“	false axis number
2	"watch_f“	"watchdog“ generated
3	"fcode0“	signal FC0 active
4	"fcode1“	signal FC1 active
5	"fcode2“	signalFC2 active
6	"serv_en“	internal control bit
7-31	"free“	not used

7.2 Example Configuration

The various configuration possibilities of the axis controllers are illustrated below with the help of examples.
Default values apply for parameters not listed.

MAC4-STP

Setting	Mnemonic	
Motor with 500 steps per revolution ,	UWMS	500
Start-/stop frequency of the motor 30 Hz ,	WSSF	30
BOOST- time 80 ms	WBT	80
Linear axis with limits of 0 and 1,000,000 Steps	UWSH UWLP UWLN	1 1000000 0
Internal counting of steps generated (no position sensor connected)	UWET	0
Activation of the interrupt, "drive fault" (Vector $A0)	WIRQ3	$A0
Maximum following error 1,000 steps	WMF	1000
Emergency braking ramp 2,000 Hz/SP	WAA	2000
Justification velocity 8 Hz	WLV	8
Axis initialisation	INIT	

MAC4-INC

Setting	**Mnemonic**	
Motor to turn in anti-clockwise direction	USR	
Linear axis with the limits of -2,000,000 and 2,000,000 Inc	UWSH UWLP UWLN	1 2000000 -2000000
All switch levels active high	UWPS UWNS UWRS UWSO	2 2 2 2
Activation of the interrupt "drive fault" (Vector $A2) and "end of positioning" (Vector $A4)	WIRQ3 WIRQ4	$A2 $A4
"on-fly" operation		
Maximum following error 2,000 Inc	WMF	2000

Setting	Mnemonic	
Controller set-up:		
Controller amplification 20	WGA	80
Zero 0.5	WZE	128
Pole 0	WPO	0
Integral amplification 0.02	UWKISC	3
	WKI	41
Emergency braking ramp 5,000 Inc/SP/SP	WAA	5000
Justification velocity 20 Inc/SP	WLV	20
Axis initialisation	INIT	

MAC4-SSI

Setting	Mnemonic	
Circular axis with the size 10,000 Inc	UWSH	2
	UWCR	10000
Absolute encoder (resolution 10,000) connected	UWET	4
via the VMEbus, access via address $80C000	UWEA	$80C000
	UWEB	14
No switches connected	UWPS	0
	UWNS	0
	UWRS	0
	UWSO	0
"off-fly" operation	FLYOFF	
Maximum following error 500	WMF	500
Controller set-up:		
Controller amplification 50	WGA	200
Zero 0.99	WZE	253
Pole 0	WPO	0
Integral amplification 0.05	UWKISC	0
	WKI	13
Emergency braking ramp 5,000 Inc/SP/SP	WAA	5000
Set "watchdog" to 1 minute	WWD	60000
Axis initialisation	INIT	

7.3 Queue Description

7.3.1 Message Structure for VMEbus or RS232

The conversion of commands into corresponding queue entries or serial messages is explained using examples. The following commands have to be transferred to the axis controller MAC4-INC:

Code	Parameter	Mnemonic	Meaning
70	16	WAP 16	target position 16 Inc
59		MP	mode „positioning“
3		E	activate positioning
116		RCP	actual axis position is read
1		(Error)	non implemented command

These commands should be transferred to axis 1(axis number 0).

Entry in the Command Queue via the VMEbus

When directly accessing the queue via the VMEbus the following entries must be made in sequence in the command queue address area:

Axis Nr.	empty	Code		Parameter				Source	empty
0	*	0	70	0	0	0	16	0	*
0	*	0	59	*	*	*	*	0	*
0	*	0	3	*	*	*	*	0	*
0	*	0	116	*	*	*	*	0	*
0	*	0	1	*	*	*	*	0	*

The entries marked with a "*" may be chosen freely. These entries are ignored by the axis controller.

Entries in the Response Queue over the VMEbus

The responses of the axis controller as a reaction to commands are entered in sequence in the response queue address area:

Axis Nr.	empty	Code		Parameter				Source	empty
0	*	0	70	$02	$00	$01	$01	0	*
0	*	0	59	$02	$00	$01	$01	0	*
0	*	0	3	$02	$00	$01	$08	0	*
0	*	0	116	$00	$00	$03	$E8	0	*
0	*	0	161	$02	$00	$21	$08	0	*

Explanation of the responses:

- at the start of the instruction sequence the axis is in the mode "enable".
- the change to the mode "positioning" is made with the "execute" instruction.
- the position read by the fourth instruction is 1,000 Inc.

A syntax error is shown as the response to the last instruction.

Transmit Message over the Serial Interface

When communicating via the serial interface the queue entries given above are made by the interface driver of the axis controller. In the field "Source" there is a One instead of a Zero. The message to the axis controller must be structured as follows:

DLE	STX	Axis Nr.	Code		Parameter					Check Sum
16	2	0	0	70	0	0	0	16	16	86
16	2	0	0	59	0	0	0	0		59
16	2	0	0	3	0	0	0	0		3
16	2	0	0	116	0	0	0	0		116
16	2	0	0	1	0	0	0	0		1

As the parameter "16" is identical with the value for DLE, this byte must be transmitted twice in the first message.

Response Message via the Serial Interface

By means of the source entry the axis controller recognises that it must return the message in the response queues via the serial Interface. The response message has the form:

DLE	STX	Axis Nr.	Code		Parameter				Check Sum
16	2	0	0	70	2	0	1	1	74
16	2	0	0	59	2	0	1	1	63
16	2	0	0	3	2	0	1	8	14
16	2	0	0	116	0	0	3	232	95
16	2	0	0	161	2	0	33	8	204

Control Characters in Serial Protocol

Control character	Code
DLE	$10
NAK	$15
STX	$02

7.3.2 Queue Implementation under OS-9

C-functions under the operation system OS-9 are available for the entry of the messages in the command queue, that may be easily transferred to another platform.

A queue is implemented as ring store array. The elements of the queue are numbered in sequence. Two pointers, a read- and a write pointer, are used for administration of the queue.

> ⇨ Only one process - write or read - may access the queue at one time. A simultaneous write access falsifies the message. Simultaneous reading may also lead to false values.

The queues of the axis controller must be accessed with absolute addresses, i.e. the access must be made to fixed storage space. An absolute address must be made available to the queue management function.

> ⇨ The user must ensure that other data address ranges are not overwritten by the queue.

The following steps are necessary when transmitting messages via the queue:

- generate queue
- enter message
- read message
- close queue.

A C routine is available for each step, that can be obtained through MACCON.GmbH

```
queue_t *queueopen (unsigned int nel, int width, void *loc)
```

This function generates a queue, initialises the queue header and returns a pointer of the data type queue_t. queue_t is the structure definition for the queue header (queue.h):

```
typedef struct {
        int base;       /* Offset queue-header - queue-address range */
        int front;      /* Write index */
        int rear;       /* Read index */
        int free;       /* frei */
        int width;      /* Size of a queue entry in Bytes */
        int nel;        /* max. number of elements in the queue */
} queue_t;
```

The absolute start address of the queue is transferred to the routine by means of the parameter *loc*. The queue header must follow the queue-base address directly.

> ⇨ This function is executed by the axis controller. It may not be called by the host system: This ensures that only one process can write to the queue header.

The other communications process only needs a pointer of the type queue_t to define the absolute start address of the queue.

Response values:

loc is returned.

The axis controller defines with this function a command- and response queue of 10 queue elements of each 10 byte length.

Example: Define a queue of 10 elements, each of 10 bytes, at address 0x854700

```
#define NUMBER   10
#define SIZE     10
#define LOCATION 0x854700

queue_t *myqueue;

myqueue = queueopen(NUMBER, SIZE, (void *)LOCATION);

void queueclose(queue_t *q)
```

With this function the queue *q* is cleared. Elements, that are not yet in the queue are lost.

Response value:

none

The routine queue_close() should only be called by a process, after other communication processes have ceased to access the queue.

```
int enqueue(queue_t *q, char *s)
```

The same number of bytes are written from the buffer *s* in the queue *q* as were given as the size of an element (*width*) on opening the queue.

Response value:

0 = queue full, no entry possible
1 = entry possible

Of the maximum predefined number of storage addresses in the queue *nel* only a maximum of *nel*-1 may actually be occupied with data. For programming reasons one element must remain free between the read- and write pointers of a queue.

```
int dequeue(queue_t *q, char *s)
```

The same number of bytes are written from the queue in the buffer *s* as were given as the size of an element (*width*).

Response value:

0 = queue empty, no element taken
1 = element taken

```
unsigned int qu_avail(queue_t *q)
```

This routine interrogates the number of available elements in the queue *q*.

Response value:

Number of available elements

```
unsigned int qu_used(queue_t *q)
```

This routine interrogates the number of occupied queue elements in the queue *q*.

Response value:

Number of occupied addresses

Example

```
#include <stdio.h>
#include "queue.h"

#define QUEUEELEMENTS  10
#define ELEMENTSIZE        10
#define OUT_BUF_ADR                    0x853000
#define IN_BUF_ADR                     0x854000

main()
{
char i;                              /* counter variable */
queue_t *outqueue_ptr;               /* pointer to output queue */
queue_t *inqueue_ptr;                /* pointer to input queue */
char *data_ptr[10];                  /* pointer to data buffer */

        /* Generate a pointer to a queue generated by another process
         * The queue begins with the queue header pointing to
         * address OUT_BUF_ADR.
        */
        outqueue_ptr = (queue_t *)OUT_BUF_ADR;

        /* Open a queue at address IN_BUF_ADR.
         * The queue should have 10 elements of 10 Bytes each.
         */
        inqueue_ptr = queueopen(QUEUEELEMENTS, ELEMENTSIZE,
                    (void *)IN_BUF_ADR);
        /* an attempt is made to read data from the queue.
         */
        if(dequeue(outqueue_ptr, (char *)data_ptr)) {
                printf("Data ");
                for(i = 0; i < 10; i++)
                        printf("%c ", *data_ptr[i]);
                printf("\n");
        }
        else
                printf("Queue empty!\n");

        /* The previously read date are written to another queue
         */
        if(enqueue(inqueue_ptr, (char *)data_ptr))
                printf("Data written in queue.\n");
        else
                printf("Queue full!\n");

        /* close queue. */
        queueclose(inqueue_ptr);
```

}

7.3.3 Queue Address List

The command queue of the axis controllers is the basis for the address list in Tab. 7..1.

Address	Contents	Comment
$xx4700	base	**Begin of the queue header**
		Value is always $18 (= Offset to queue base address).
		This value may not be changed.
$xx4704	front	Set by enqueue()
$xx4708	rear	Set by dequeue()
$xx470C	free	
$xx4710	width	The value is always $A.
		This value may not be changed.
$xx4714	nel	The value is always $A.
		This value may not be changed.
		End of the queue header
	entry 1	**Begin of the queue base address**
$xx4718		Axis number
$xx4719		free
$xx471A		MSB - instruction code
$xx471B		LSB - instruction code
$xx471C		MSB - parameter value
$xx471D		I- parameter value
$xx471E		I- parameter value
$xx471F		LSB - parameter value
$xx4720		Source
$xx4721		empty
	entry 2	
$xx4722		axis number
$xx4723		MSB - instruction code
$xx4724		LSB - instruction code
$xx4725		MSB - parameter value
$xx4726		I- parameter value
$xx4727		I- parameter value
$xx4728		LSB - parameter value
$xx4729		Source
$xx472A		empty
$xx472C entry 3	analogue entry 1 and 2	
$xx4736 entry 4	analogue entry 1 and 2	
$xx4740 entry 5	analogue entry 1 and 2	
$xx474A entry 6	analogue entry 1 and 2	
$xx4754 entry 7	analogue entry 1 and 2	
$xx475E entry 8	analogue entry 1 and 2	
$xx4768 entry 9	analogue entry 1 and 2	
$xx4772 entry 10	analogue entry 1 and 2	
$xx477C		**End of the queue base address**

Table 7.1: Queue Address List

7.4 Technical Data

6HE-VMEbus-card for the control of:

4 DC servodrives, brush and brushless, (MAC4-INC/SSI)

OR

4 stepper drives (MAC4-STP)

- local Motorola-CPU 68000, with 16 MHz clock and 64 kByte DPRAM to VMEbus
- VMEbus-Master/Slave A24 D16
- standard-EPROM firmware or user programmable
- generates and processes interrupts
- process peripherals potentially isolated from the CPU
- supply voltage 5 V
- current consumption 2.5 A
- temperature range
  operational environment: 0 - 45 °C
  storage and transport: -25 - 80 °C
- VMEbus- connection via connector VMEbus-P1
- peripheral connections via a two row peripheral connector -P2 according to DIN 41612, style C via a backplane or a 64-pole flat-band cable
- RS232 with SUB-D-9m connector at the front panel
- 4 enable outputs
- 3 user definable inputs (error code inputs, TTL)
- 1 emergency stop input with LED-display
- 6 Status-LEDs per axis
- reset switch at the front panel or reset via the VMEbus

MAC4-INC:

- 8 inputs - limit switches (24 V)
- 4 inputs- reference switches (24 V)
- 4 inputs -"drive fault" signal (24 V)
- 4 analog motor outputs +- 10 V (12 bit)
- 6 incremental encoder inputs - unipolar or bipolar with index pulse
- controller sampling rate 2.5 ms (all axes)
- cycle time of the profile generator 10 ms (all axes)

MAC4-SSI:

- 8 inputs - limit switches (24 V)
- 4 inputs - reference switches (24 V)
- 4 inputs - "drive fault" signals (24 V)
- 4 analog motor outputs +- 10 V (12 bit)
- 4 SSI-inputs - bipolar (TTL)
- 4 SSI outputs - bipolar to the clock (TTL)
- Gray-/binary sensor can be connected, configuration per jumper
- SSI-clock rates (750,375,187.5 or 93.75 kHz) selectable per jumper
- controller sampling rate 2.5 ms (all axes)
- cycle time of the profile generator 10 ms (all axes)

MAC4-STP:

- 8 inputs limit switches (5/24 V)
- 4 inputs reference switches (5/24 V)
- 4 inputs "drive fault" signals (5/24 V)
- 4 motor outputs with boost, direction and pulse output
- maximum step frequency 500 kHz
- 4 incremental encoder inputs - unipolar or bipolar with index pulse
- controller sampling period 8 ms (all axes)
- cycle time of the profile generator 32 ms (all axes)

7.5 MAC4-INC: Pin Allocation of the Peripheral Connector -P2

Row A	Meaning	Row C	Meaning
A1	+ 5 V (Input)	C1	+ 15 V (Input)
A2	GND2	C2	- 15 V (Input)
A3	GND2	C3	AOUT1
A4	AOUT3	C4	AOUT2
A5	AOUT4	C5	Ch_A1
A6	Ch_B1	C6	-Ch_A1
A7	-Ch_B1	C7	Ch_Z1
A8	Ch_A2	C8	-Ch_Z1
A9	-Ch_A2	C9	Ch_B2
A10	Ch_Z2	C10	-Ch_B2
A11	-Ch_Z2	C11	Ch_A3
A12	Ch_B3	C12	-Ch_A3
A13	-Ch_B3	C13	Ch_Z3
A14	Ch_A4	C14	-Ch_Z3
A15	-Ch_A4	C15	Ch_B4
A16	Ch_Z4	C16	-Ch_B4
A17	-Ch_Z4	C17	AG1
A18	AG3	C18	AG2
A19	AG4	C19	/EN1
A20	/EN3	C20	/EN2
A21	/EN4	C21	/DF1
A22	/DF3	C22	/DF2
A23	/DF4	C23	FC0
A24	FC2	C24	FC1
A25	n.c.	C25	n.c.
A26	/STOP	C26	GND2
A27	/IESP1	C27	/IESN1
A28	/IESP2	C28	/RS1
A29	/IESN2	C29	/RS2
A30	/IESN3	C30	/IESP3
A31	/RS3	C31	/IESP4
A32	/RS4	C32	/IESN4

AG1-4 ... not implemented

7.6 MAC4-SSI: Pin Allocation of the Peripheral Connector -P2

Row A	Meaning	Row C	Meaning
A1	+ 5V (Input)	C1	+ 15 V(Input)
A2	GND2	C2	- 15 V (Input)
A3	GND2	C3	AOUT1
A4	AOUT3	C4	AOUT2
A5	AOUT4	C5	T1
A6	DAT1	C6	-T1
A7	-DAT1	C7	n.c.
A8	T2	C8	n.c.
A9	-T2	C9	DAT2
A10	n.c.	C10	-DAT2
A11	n.c.	C11	T3
A12	DAT3	C12	-T3
A13	-DAT3	C13	n.c.
A14	T4	C14	n.c.
A15	-T4	C15	DAT4
A16	n.c.	C16	-DAT4
A17	n.c.	C17	n.c.
A18	n.c.	C18	n.c.
A19	n.c.	C19	/EN1
A20	/EN3	C20	/EN2
A21	/EN4	C21	/DF1
A22	/DF3	C22	/DF2
A23	/DF4	C23	FC0
A24	FC2	C24	FC1
A25	n.c.	C25	n.c
A26	/STOP	C26	GND2
A27	/IESP1	C27	/IESN1
A28	/IESP2	C28	/RS1
A29	/IESN2	C29	/RS2
A30	/IESN3	C30	/IESP3
A31	/RS3	C31	/IESP4
A32	/RS4	C32	/IESN4

7.7 MAC4-STP: Pin Allocation of the Peripheral Connector -P2

Row A	Meaning	Row C	Meaning
A1	+ 5 V (Input)	C1	DIR3
A2	GND2	C2	DIR1
A3	GND2	C3	OUTP1
A4	OUTP3	C4	OUTP2
A5	OUTP4	C5	Ch_A1
A6	Ch_B1	C6	-Ch_A1
A7	-Ch_B1	C7	Ch_Z1
A8	Ch_A2	C8	-Ch_Z1
A9	-Ch_A2	C9	Ch_B2
A10	Ch_Z2	C10	-Ch_B2
A11	-Ch_Z2	C11	Ch_A3
A12	Ch-B3	C12	-Ch_A3
A13	-Ch_B3	C13	Ch_Z3
A14	Ch_A4	C14	-Ch_Z3
A15	-Ch_A4	C15	Ch_B4
A16	Ch_Z4	C16	-Ch_B4
A17	-Ch_Z4	C17	BOOST1
A18	BOOST3	C18	BOOST2
A19	BOOST4	C19	/EN1
A20	/EN3	C20	/EN2
A21	/EN4	C21	/DF1
A22	/DF3	C22	/DF2
A23	/DF4	C23	FC0
A24	FC2	C24	FC1
A25	DIR4	C25	DIR2
A26	/STOP	C26	GND2
A27	/IESP1	C27	/IESN1
A28	/IESP2	C28	/RS1
A29	/IESN2	C29	/RS2
A30	/IESN3	C30	/IESP3
A31	/RS3	C31	/IESP4
A32	/RS4	C32	/IESN4

8. Glossary

APC	Advanced peripheral controller
axis initialisation	Adaptation of the axis controller to a specific application by means of configuring system parameters
basic initialisation	After switch on the axis controller initialises the internal data fields, enters the default values in the parameter fields and sets the status and displays.
circular-optimised	Returns the position at the axis limits to the base value (motion over 360° repeats infinitely). When positioning the shortest route to the target is automatically selected.
circular axis	returns the position at the axis limits to the base value (motion over 360° repeats infinitely). Positioning is only possible within the range of one load revolution.
DAC	Digital to analogue Converter.
DPRAM	Dual-Port-RAM
Index pulse	Reference signal (z-track) of an incremental encoder
IPC	Intelligent peripheral controller
justification	The process of aligning the axis to a marker, usually the reference pulse. The justification velocity is a low speed used to approach the final position in order that the axis can stop immediately on recognition of the target.
linear axis	Axis movement is linear instead of rotary. The range of movement is physically limited. The range of operation is limited by software limits and limit switches.
message	Commands and responses in communication between the axis controller and the host.
off_fly	Instructions transmitted to the axis controller are executed individually. The instructions that follow wait until completion.
on_fly	Cause the immediate execution of instructions.
position measurement, external:	The axis controller receives the position information via the VMEbus or the DRAM. There is no position sensor at the peripheral connector -P2.
position measurement, internal	The axis controller receives its position information from an encoder connected to peripheral connector -P2.
queue	Ring buffer for the intermediate storage of messages between the axis
queue administration	Addresses for the attributes: read- and write pointer of a queue
queue base address	Start address for the messages in a queue
SP	Generator sampling period

SSI	Synchronous serial interface: interface standard for absolute encoders
start-/stop frequency	is the maximum possible acceleration or braking step from or to velocity Zero of a stepper motor.

9. References

(1) Dorsch Mikrosystem GmbH IPC V4.0
Technical Manual (in German), Ref. Nr. 942.I346.50

10. Index

11. Hardware Drawings

Circuit diagrammes are included with this document. These may help the user to analyse the hardware functions of the MAC4 series of axis control cards. There may also be of assistance when interface circuits are to be specified for these cards.

MAC4-INC

1. ST2-Connector (ST2-Stecker)
2. Microprocessor Interface (Rechner-Interface)
3. Opto-couplers (Optokoppler)
4. Counters (Zaehler)
5. Enable Logic (Freigabe)
6. Absolute Position Sensor (Absolutgeber)
7. Selection Logic (Select)
8. DC/DC Converter (DC-DC-Wandler)
9. DAC Output to Drives (Sollertgeber)
10. Axis Interface and Status (Synchro)
11. Axis Status (Synchro)
12. Interconnections Processor/Peripherals (MAC-64POL-Stecker)
13. Component Layout (Rev. D)

MAC4-SSI

1. ST2-Connector (ST2-Stecker)
2. Microprocessor Interface (Rechner-Interface)
3. Opto-couplers (Optokoppler)
4. SSI Clock and Receivers (Pulsgenerator und Empfänger)
5. Enable Logic (Freigabe)
6. SSI-Counter
7. Selection Logic (Select)
8. DC/DC Converter (DC-DC-Wandler)
9. DAC Output to Drives (Sollertgeber)
10. Axis Interface and Status (Synchro)
11. Axis Status (Synchro)
12. Interconnections Processor/Peripherals (MAC-64POL-Stecker)
13. Component Layout (Rev. A)

MAC4-STP

1. Block Diagramme (MAC4STP)
2. Microprocessor Interface (Rechner-Interface)
3. Block Diagramme, Axis Control (MAC4STP Achsen)
4. Drive Interface, Axis 1 (MAC4STP Achsteuerung 1)
5. Drive Interface, Axis 2 (MAC4STP Achsteuerung 2)
6. Drive Interfcae, Axis 3 (MAC4STP Achsteuerung 3)
7. Drive Interface, Axis 4 (MAC4STP Achsteuerung 4)
8. Block Diagramme, Connector VME P2 (MAC4STP)
9. Axis Input Circuits (MAC4STP, Lesevestärker)
10. Status Inputs (MAC4STP, Status)
11. Opto-couplers, Axis 1 (MAC4STP, Trennung 1)
12. Opto-couplers, Axis 2 (MAC4STP, Trennung 2)
13. Opto-couplers, Axis 3 (MAC4STP, Trennung 3)
14. Opto-couplers, Axis 4 (MAC4STP, Trennung 4)
15. Component Layout, Component Side (Rev. A)
16. Component Layout, Solder Side (Rev. A)