PINX - 5

USER'S MANUAL

P I N X - 5

USER'S MANUAL

****

Table of Contents

PINX-5

1. INTRODUCTION

PINX-5 is a freely programmable stepping motor controller for one or two axes. Free programmability, compact structure with integrated power amplifiers and logic input/outputs provide the controller with undreamt-of efficiency and application possibilities.

PINX-5 knows 10 different programme instructions dealt with in this manual in detail. Pay attention to the fact that this unit uses a sequential processing of programmes just like any computer. On principle, this means that an instruction is not executed until the previous instruction is completed.

Thus, the general rules used for computer programmes also apply to the structure of a PINX-5 programme:

* Subdivision of the total task into partial tasks
* Definition of desired operating possibilities
* Programme representation by means of flow card, structure diagram, etc.
* Programming
* Test
* Documentation

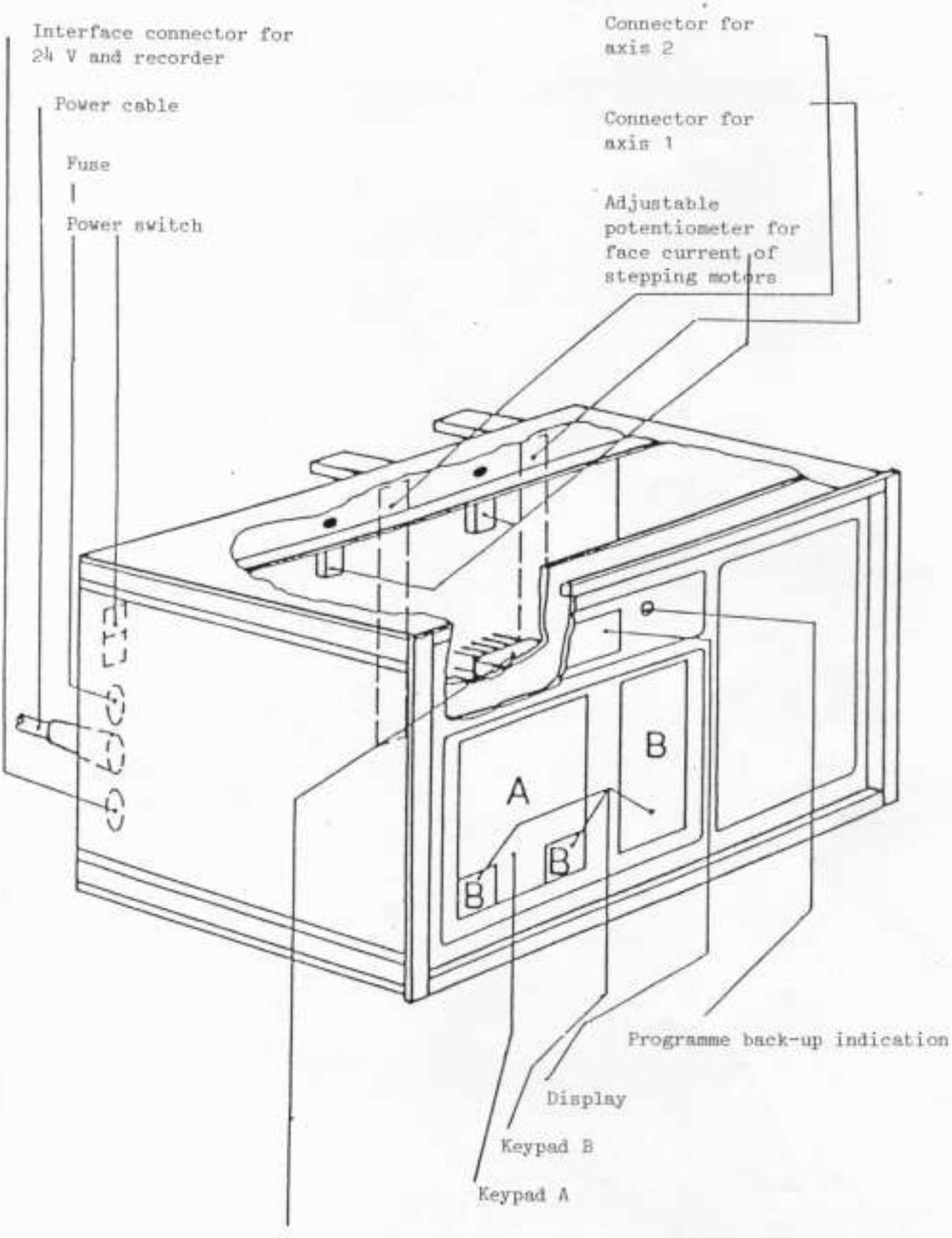
TECHNICAL CHARACTERISTICS

Arrangement:	In a compact aluminium casing with key-board and programming table at the front panel. 32-pin connector according to DIN 41612/d for each axis as well as installation set for front panel installation are included.			
Dimensions:	Width:	228 mm	Depth of the one axis version:	170 mm
	Weight:	114 mm	Depth of the two axes version:	230 mm
Supply:	220 V AC / 50 Hz			
	one axis version:	160 VA		
	two axes version:	250 VA		
Inputs:	one axis version:	4 (24 V DC)		
	two axes version:	8 (24 V DC)		
Outputs:	one axis version:	4 (24 V DC / 2 A)		
	two axes version:	8 (24 V DC / 2 A)		
Counters:	10 (0 .... 99999 each)			
Motor speed:	12000 steps/s max. (= 1800 r.p.m.)			
Storage capacity:	600 programme lines (RAM battery backed, 5 year life)			

Please pay attention to date of battery change.

Type of battery: TL 2150 size, 1/2 AA

PINX-5 UNIT LAYOUT



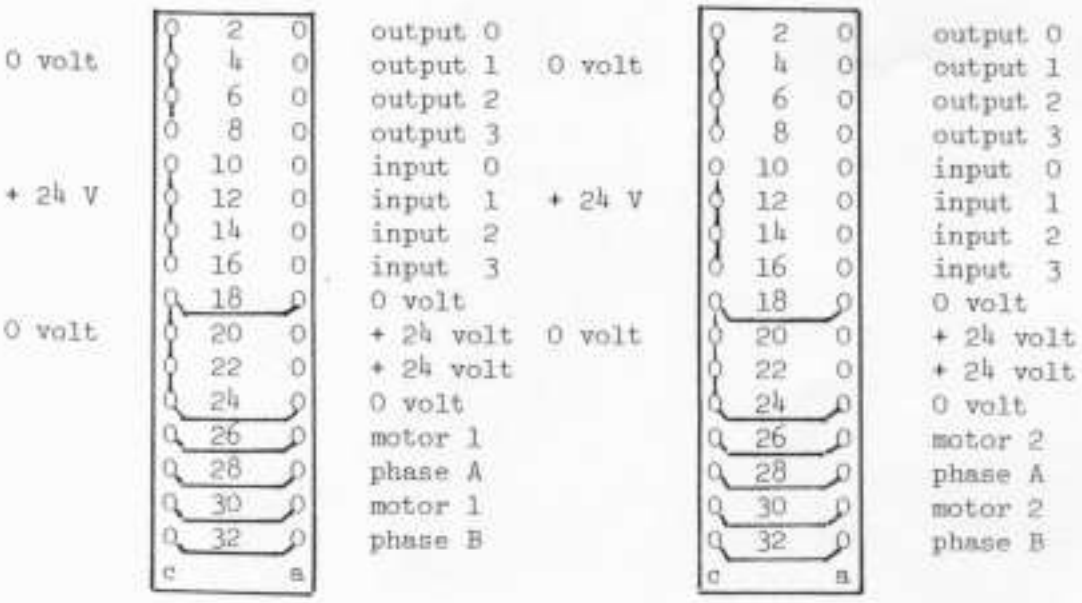
2. Putting the unit into operation

The PINX-5 stepping motor controller is very easily put into operation by means of the following steps:

* connection of signal transducers and loads at inputs/outputs
* connection of stepping motors
* selection of the desired frequency range
* adjustment of the stepping motor current

2.1. Connectors

One connector is installed for PINX-5 one axis version and two axes version. The left connector seen from the rear is used to connect axis 1.



Rear view

Connections shown are already realized within the unit.

Outputs

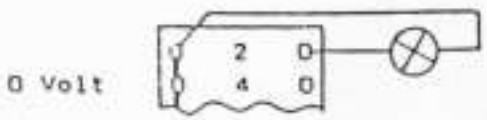
Four logic outputs are available per axis. The outputs are pulse switching and are capable of driving 2 A each. The sum current per group (4 outputs) is 2.5 A. Each output is short-circuit-proof and protected against induction peaks. If the short-circuit fuse is blown, the respective output or the total unit has to be disconnected.

Inputs

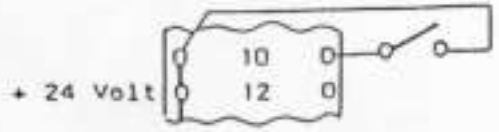
Four inputs are available per axis. The inputs are pulse switching (PNP). The control current is 5 mA and the switching threshold is approx. 14 V.

Wiring examples

output 0 with lamp



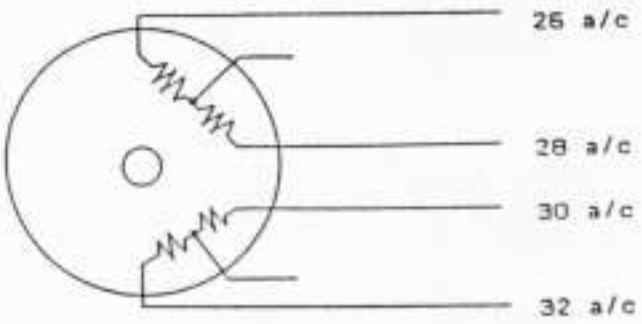
Input 0 with switch



2.2 Motor connection

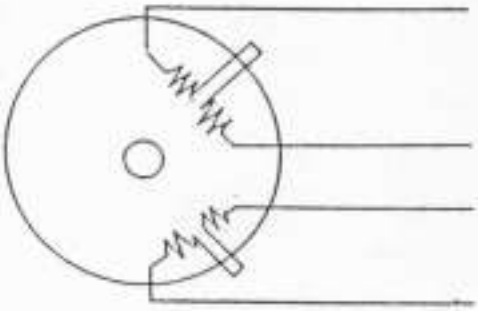
The bipolar stepping motor amplifiers installed are capable of driving virtually all commercial two-phase hydride stepping motors. The stepping angle is 0.9° corresponding to 400 steps per revolution.

Connection of 6 wire motors

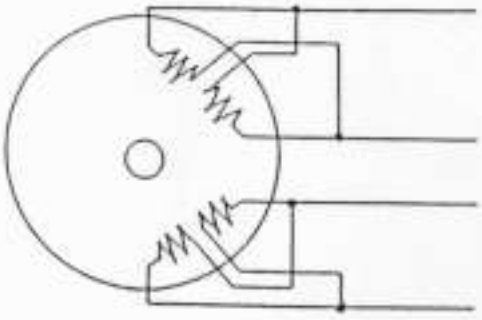


Conncection of 8 wire motors

Windings in series for high torque/low speed



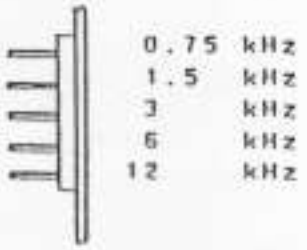
Windings in parallel for high speed/smaller torque



2.3. Selection of desired frequency range

5 coding bridges are available within the unit for selection of the frequency range (refer to chapter 1 unit layout). The following frequency ranges may be selected by changing the code bridges:

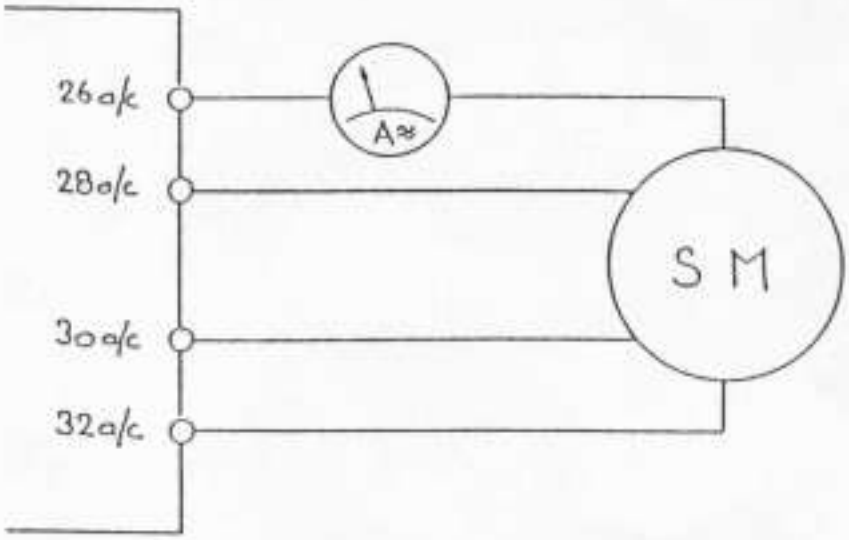
(view from the top:)



2.4. Face current adjustment

After removal of the top cover plate, the adjustable potentiomter is accessible at the amplifier PCB (at the heat sink) (refer to chapter 1, unit layout). The current is increased by turning clockwise. The face current shall be measured with slowly rotating motor (400 Hz). Since a bipolar amplifier is considered in case of PINX-5 the current is to be set to 65 % of the rated face current of the stepping motor used.

Caution: Set ammeter to alternating current.



The 400 stepping frequency results in the following programme motor velocities (VSM) depending on the frequency range (FB) selected :

FB	VSM
12'000 Hz	8
6'000 Hz	17
3'000 Hz	34
1'500 Hz	68
750 Hz	136

Programming examples for motor movement by 400 Hz

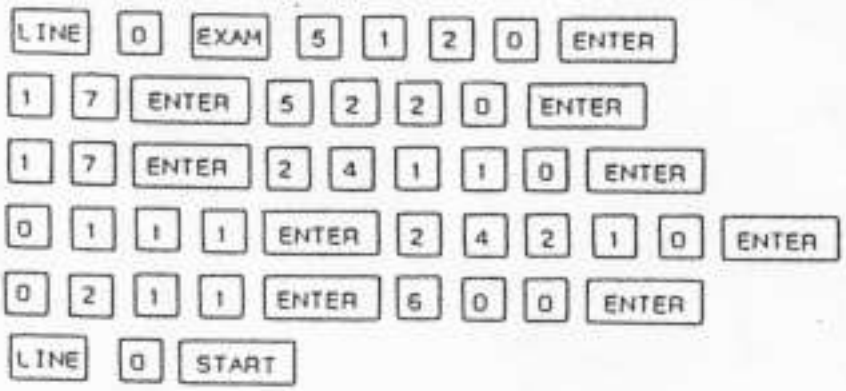
The stepping motors 1 and 2 may be moved by the keys 1 and 2 of the keyplate A. The present example uses a frequency range of 6 kHz. If other frequency ranges have been coded, motor velocities according to the table shown above are to be programmed in the programme lines 1 and 3.

Programme

LINE	INSTRUCTION		COMMENT
0	5	- 120 -	motor velocity for motor 1
1		17	
2	5	- 220 -	motor velocity for motor 2
3		17	
4	2	- 4110 -	calling of key 1
5	0	- 1-11	motor rotation 1
6	2	- 4210	calling of key 2
7	0	- 2-11	motor rotation 2
8	6	- 0 0	jump to line 0
9			

Key sequence for programme input

(enter all numbers by keypad A)



Then, the stepping motors may be moved by keys 1 and 2 enabling the current to be adjusted with running motor.

3. Key functions

Keypad A

The keypad A (keys 0 ... 9) is used to enter the line numbers, programme instructions, and data values. Each key may be scanned as function key by the programme.

Keypad B

The keys of the keypad B are operating keys for the input and the test of PINX-5 programmes. Each key may be scanned as function key by the programme. The key functions described below are efficient with disconnected programme lock (LOCK) only.

3.1 LINE

Each programme instruction is written to one programme line. Max. 600 programme lines (0 ... 599) are available. The programme line presently processed by PINX-5 is displayed upon actuation of the "LINE" key. At the same time, the programme is stopped after execution of that line. If a new programme line shall be selected, it is entered by means of the keypad A (...9).

Example: Display of actual programme line / programme stop

Key sequence	Display	
L I N E	XXX	actual programme line

Example: Select new programme line

Key sequence	Display	
L I N E	XXX	actual programme line
Y Y Y	YYY	programme line YYY

3.2 EXAM

The EXAM key is used to display the instruction entered into an arbitrary programme line. All instructions entered may be displayed in increasing order by repeated actuation of the key. If individual instructions or complete programme parts are to be modified, the EXAM key has to be actuated after having entered the desired programme line (refer to 3.3 ENTER).

Example : Display of instruction code

Key sequence	Display	
LINE	XXX	actual programme line
YYY	YYY	programme line YYY
EXAM	A-AAAA	instruction

3.3. ENTER

To input programme instructions, first write them into the PINX-5 display. Terminate any input by ENTER to transfer it into the programme memory. Upon actuation of this key, the programme line number is automatically incremented by 1. Thus, the next instruction can be stored immediately.

Example: Input of programme instructions

Key sequence	Display	
L I N E	XXX	actual programme line
Y Y Y	YYY	new programme line
EXAM	A-AAAA	display of present instruction
BBBBB	B-BBBB	new instruction
ENTER	------	storing
CCCCC	C-CCCC	next instruction
ENTER	------	storing

LINE and EXAM are to be entered at the beginning of the input of programme instructions only.

3.4. CLEAR

If you make a mistake while keying in the programme instructions, you can CLEAR the display and make a new input.

Key sequence	Display	
LINE	XXX	actual programme line
Y Y Y	YYY	new programme line
EXAM	A-AAAA	display of instruction
BBBBB	B-BBBB	new, incorrect instruction
CLEAR	A-AAAA	former instruction
CCCCC	C-CCCC	new, correct instruction
ENTER	------	storing

3.5 START

This key is to be used to start the programme provided that the programme is not LOCKED. Actuate LINE to select the programme line from which the programme has to be executed and start subsequently.

Example:	Key sequence	Display	
	L I N E	XXX	actual programme line
	Y Y Y	YYY	new programme line
	S T A R T		programme is executed from programme line YYY

The execution of the programme may be stopped by actuation of the LINE key.

3.6. STEP

To be able to test a programme step by step, start by STEP instead of START. After execution of the first programme line, press STEP again to initiate the next programme line. The actual programme line can be followed in the display.

3.7. BREAK

To be able to have a programme executed block by block, you may define arbitrary stops by BREAK without changing the programme stored. A stop is set by means of LINE and BREAK and reset by switching off the unit.

Example:	Key sequence	Display	
	LINE	XXX	actual programme line
	Y Y Y	YYY	end of programme block
	B R E A K	YYY	set stop
	L I N E		
	Z Z Z	ZZZ	start programme block ZZZ to YYY

3.8. LOCK

A programme entered may be protected against incidental changes by entering a certain code.

Actuation of the programme lock

1) Press LINE
2) Press lock; the display is cleared
3) Enter 3 1 4 1 5
4) Press LOCK again; the programme is automatically started at LINE 000 and the "LOCKED" LED lights.

The programme is also started automatically upon switching on the PINX-5.

Release of the programme lock

1) Switch off PINX-5
2) Press and hold LOCK
3) Switch on PINX-5
4) Release LOCK and enter 3 1 4 1 5 after approx. 1 s.
5) Press LOCK again. The display shows programme line 000
6) PINX-5 is ready for a programme input.

4. Programming

A PINX-5 programme consists of a series of instructions entered according to the desired sequence into the memory of the control. The available programme instructions can be learned quickly and enable a maximum flexibility of programme preparation. An instruction always starts with a digit between 0 and 9 representing an instruction code. The 4 other possible digits have the following meanings:

S/J = selection / jump - S = selection of the functional group (e.g. axis 1 or 2)

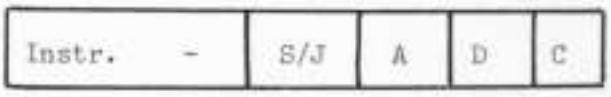
J = jump function

A = address - selection of the functional element (inputs, keys, etc.)

D = definition - definition of the function (e.g. positioning by absolute or incremental dimensions, etc.)

C = condition - condition for the function (e.g. sense of rotation + or - etc.)

Instruction format



Each instruction covers one programme line. The two instructions INDEX and DATA cover two programme lines.

Storage organization

As you know, the programme is protected against changes, if the programme LOCK is switched on. To enable data values for speeds, dwells, positioning distances, etc. to be modified, the memory is subdivided into two partial ranges:

A) Programme memory
b) Data memory

The programme memory remains locked, whilst data of the data memory may be modified by means of the keyboard at any time. The input of data into the data memory is described in detail in chapter 4.7.

The size of the two partitions A and B is freely selectable. For this purpose, enter the desired "borderline" into programme line 600.

Example: Programme line 600: --- 400

This results in the following partition of the memory:

Programme memory range	0 ... 399
Data memory range	400 ... 599

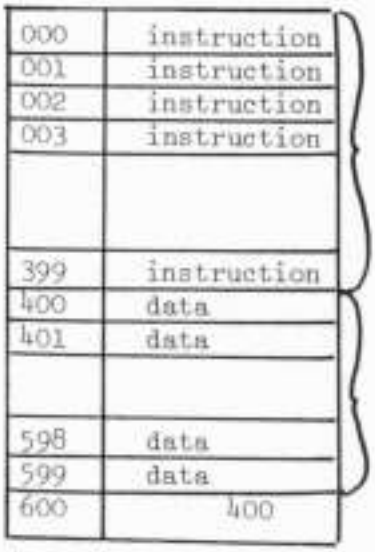
Clearing the memory

The complete memory (programme lines 000 ... 599) may be cleared by entering the code 27182.

Procedure

1) Press LINE
2) Press LOCK, display is switched off
3) Enter 27182
4) Press LOCK
5) By the clearing process, the programme memory is overwritten by ----- and the data memory by 0.

Graphic representation of the storage organization



Programme memory
Here, instructions and permanent data values are stored and protected by the programme LOCK against accidental changes.

Data memory
Here, data to be changed during operation (distances, speeds) are stored.

4.1 JOGGING / indefinite motion (Code 0)

This instruction causes a movement of the stepping motor into a defined direction. The run instruction is executed in combination with an input instruction only. The stepping motor turns as long as the latest input instruction executed in the programme is active.

FORMAT OF RUN CODE 0

0	-	G	-	u/$\overline{U}$	-/+

G	1: motor motion, axis 1 2: motor motion, axis 2
u/$\overline{U}$	0: absolute position register does not detect motion 1: absolute position register detects motion
-/+	0: positive sense of rotation + 1: negative sense of rotation -

Selection G

Selection of axis 1 or 2.

Definition u/$\overline{U}$

D = 1 or 0 defines, whether the absolute position register detects the running motor motion or not. The absolute position register is used to record the absolute actual position of the motor referred to 0.

Condition -/+

Determination of the sense of motor rotation.

Example

Positive sense of rotation as long as function key 3 of keypad A is actuated.

LINE		INSTRUCTION	COMMENT
0	2	- 4300	scanning of key 3
1	0	- 1-00	motion into pos. direction
2	6	- 0 0	jump to line 0
3			
4			
5			
6			
7			
8			
9			

4.2. Programmed Motion

FORMAT OF RUN CODE 1	1	-	G	-	M	E	DATA/ADDR

G	0: motor motion, axis 1 = axis 2 1: motor motion, axis 1 2: motor motion, axis 2
M	0: absolute value; distance entered in the programme 1: incremental value; distance entered in the programme 2: absulute value; distance entered from data memory, adressing of data memory in the programme 3: relative value; distance entered from data memory, adressing of data memory in the programme 4: absolute value; distance entered from data memory, data memory by means of counters 0...9 5: relative value; distance entered from data memory, data memory by means of counters 0...9
E	For E = 0...3, the motor motion is detected by the absolute position register. 0: preparation of positioning without motor motion 1: start positioning and wait until motor motion is completed 2: start positioning and wait until both motor motions are completed 3: start positioning and execute next instruction at once 4: start positioning and wait until motor motion is completed; absolute position register does not detect motor motion

Enter data/addresses subsequent to ENTER, if display is dark (max. 5 digits + sign)

Selection G

S selects the axis. If axis 1 and 2 have to move for the same distance at the same time, this is defined by S = 0.

Definition M

Every index instruction includes a distance input in the following programme line. This may be made either directly in the programme or from the data memory. In the latter case, there are two possibilities to address the desired programme storage location.

M = /1 distance input in the programme

The travelling distance is directly entered in the following programme line.

Example 1

Relative positioning by 200 mm forward and 50 mm backward on axis 1 upon actuation of key 0.

example 1

LINE		INSTRUCTION	COMMENT
0	2	- 4000	scanning of key 0
1	1	- 1-11	position by relative values
2		200	200 steps forward
3	1	- 1-11	position by relative values
4		-50	50 steps backward
5	6	- 0 0	jump to line 0
6			
7			

M = 2/3 Direct addressing of the data memory
Addressing of data memory in the programme

The desired programme memory address is directly written into the second programme line.

Example 2

Relative positioning by the distance stored in memory address of line 500 as soon as key 0 is actuated.

example 2

LINE		INSTRUCTION	COMMENT
0	2	- 4000	scanning of key 0
1	1	- 1-31	position by relative value
2		500	distance of data memory
3			500
4	6	- 0 0	jump to line 0
5			
6			

M = 4/5 Distance input from data memory
Addressing of data memory by means of counters 0...9

The address of a counter is written into the following programme line. The count defines the address of the data memory and the distance to be travelled.

The application of direct addressing by means of counts 0...9 enables the user to travel different distances by the same positioning instruction (e.g. modification of the counts and hence of travelling distances by external inputs).

Example 3

Relative positioning by the distance stored in the data memory (500).

The count (=500) of the counter 0 indicates the address of the data memory. Start by key 0.

LINE		INSTRUCTION	COMMENT
0	5	- 300	advance counter 0 to count
1		500	500
2	2	- 4000	relative positioning by
3			distance of data memory
4	2	- 4000	count of counter 0 =
5			address of data memory
6			(500)
7	1	- 1-51	
8	6	- 0 0	jump to line 0
9			

Condition E

For C = 0 ...3 the current motor motions are detected by the absolute position register.

E = 0 The positioning data required are loaded, but the positioning process is not started for the axis selected.

E = 1 All data required are loaded, the positioning process is started, and the next programme instruction is executed upon reaching the new position.

E = 2 All data required are loaded, the positioning process is started, and the nex programme instruction is executed when both axes have reached their new position.

E = 3 The positioning process is started and the next programme instruction is executed without waiting for the new position to be reached.

E = 4 This condition corresponds to C = 1, but leaved the absolute position register unchanged.

4.3. INPUT TEST (code 2)

Function keys, external imputs, markers, and internal state signals can be scanned and used to control the programme sequence.

FORMAT OF INPUT CODE 2

2	-	G	#	SW	D/I

G

0: inputs of axes 1 and 2 of OR function
1: inputs/internal signals of axis 1
2: input/internal signals of axis 2
3: markers
4: keypad A
5: keypad B
6: status of simultaneous programmes

#

If G = 0, 1 or 2

0...3: inputs

If G = 1 or 2

4: internal signal "motor motion not completed"
5: internal signal "motor speed = 0"

If G = 3

0...9: marker

If G = 4

0...9: keys 0...9

If G = 5

0: '-' key
1: LINE key
2: EXAM key
3: ENTER key
4: CLEAR key
5: START key
6: STEP key
7: BREAK key
8: LOCK key
9: '.' key

If G = 6

1: simultaneous programme 1 active
2: simultaneous programme 2 active
3: simultaneous programme 3 active

SW

0: wait
1...9: jump forward/backward by 1...9 instructions

D/I

0: wait or jump forward, if signal = 0
1: wait or jump forward, if signal = 1
2: jump backward, if signal = 0
3: jump backward, if signal = 1

Selection G

Inputs, markers, status signals and keys may be scanned by G = 0 ...6, G = 0 means that two inputs of axes 1 and 2 are scanned by the same address (0...3) by means of the OR function.

Address #

Depending on the value of G, # selects and scans different functional elements according to the instruction scheme.

Definition SW

For D = 0, the programme waits until a scan signal is active. For 0 = 1...9, the following or previous instruction may be skipped until the scanned signal is active. D states the number of instructions to be skipped.

Condition D/I

The condition C = 0/1 defines the execution of the functions wait and jump forward for active (1) or inactive (0) signal. D/I = 2/3 defines the function jump backward active (3) or inactive (2) signal.

Example

A motion into positive direction shall be executed as long as key 3 is actuated. If key 5 is actuated, a motion into negative direction shall be made.

LINE		INSTRUCTION	COMMENT
0	2	- 4310	skip next instruction, if
1			key 3 = 0
2	0	- 1-10	motion of axis 1 into pos.
3			direction
4	2	- 4510	skip next instruction, if
5			key 5 = 0
6	0	- 1-11	motion of axis 1 in neg.
7			direction
8	6	- 0 0	jump to line 0
9			

4.4. OUTPUT (code 3)

External outputs, markers, and certain control instructions for stepping motors may be allocated by this instruction.

FORMAT OF OUTPUT CODE 3: | 3 | - | G | # | - | 1/0 |

G	1: outputs/motor control of axis 1 2: outputs/motor control of axis 2 3: markers
#	If G = 1 or 2 0...3: outputs 4: interruption of motor motion by ramps (stop 1/0 = 0/start 1/0 = 1) 5: disconnect (1/0 = 1) / connect (1/0 = 0) motor current 6: stop motor motion wothout ramp (1/0 = 1) If G = 3 0...9: markers
1/0	0: allocation OFF 1: allocation ON

Selection G

Selection of stepping motor axes or markers.

Address #

Depending on the value G, outputs, markers, and control signals for stepping motors can be addressed.
A running motor motion may be interrupted by A = 4.
The motion is stopped by 1/0 = 0 and restarted and completed by 1/0 = 1.
A prepared positioning may be started by # = 4 and 1/0 = 1 (refer to 4.2. Index).
The stepping motor can be switched on and off by # = 5 to enable e.g. handwheel control.
= 6 initiates an immediate stop of motor run without ramps.

Example 1

The stepping motor 1 shall be switched on by key 1 and switched off by key 0. A lamp at output 0, axis 1 lights if the motor is switched off.

LINE		INSTRUCTION	COMMENT
0	2	- 4020	skip the next 2 instruc-
1			tions, if key 0 = 0
2	3	- 15-1	disconnect motor
3	3	- 10-1	connect output 0, axis 1
4	2	- 4120	skip the next 2 instruc-
5			tions, if key 1 = 0
6	3	- 15-0	connect motor
7	3	- 10-0	disconnect output 0,
8			axis 1
9	6	- 0 0	jump to line 0
0			

Example 2

A positioning is started by functionning key 0. The running motor motion may be stopped by key 1. Key 2 starts the interrupted motor motion and completes it without loss of steps.

LINE		INSTRUCTION	COMMENT
0	6	- 3 50	start of concurrent pro-
1			gramme "interruption" at
2			line 50
3	2	- 4000	wait, if key 0 = 0
4	1	- 1-11	incremental positioning
5		2000	distance: 2000 steps
6	6	- 0 1	jump to line 1
7			
8			
9			
0			
1			
2	2	- 4110	skip next instruction if
3			key 1 = 0
4	3	- 14-0	stop motor motion with
5			ramp
6	2	- 4210	skip next instruction, if
7			key 2 = 0
8	3	- 14-1	start motor motion with
9			ramp
0	6	- 0 50	jump to line 50
1			

4.5. COUNTER and display (Code 4)

This instruction has two functions:
- counting function
- indicating function

For counting tasks, 10 counters are available with a maximum count of 99999 each. Each counter can individually be incremented, decremented, and displayed. The counters are advanced by the data instruction, code 5 (refer to 4.6). The counters used are event counters or loop counters for programme loops.

The indicating function enables individual counts or the contents of the absolute position register of the two stepping motor axes to be indicated.

FORMAT OF COUNTER CODE 4

4	-	G	#	F	D/D

G	1: display of absolute position register, axis 1 2: display of absolute position register, axis 2 3: counter 0...9
#	If S = 1 or 2 1: display of absolute position register If S = 3 0...9: counter 0...9
F	0: display function without count change 1: increment counter 2: decrement counter (count = 0 causes jump to the next but 1 instruction) 3: clear display
D/D	0: display OFF 1: display ON

Selection G

G = 1 or 2 — The display of the absolute position register (actual position) may be selected by 1 or 2 for the two stepping motor axes.

G = 3 — The counting function is realized by G = 3.

Address #

G = 1/2 A = 1 is used to display the absolute position register.

G = 3 The 10 counters available are selected by address A = 0...9.

Definition E

The count can be incremented, decremented, or left unchanged (display function only) according to need. The next instruction is skipped, when the count 0 is reached.

Condition D/D

D/D = 1 causes the count or the contents of an absolute position register to be displayed.

Example

Any actuation of key 0 shall be counted and displayed.

```
5       -300-          set counter 0 to count 0
                  0
2       -4000          wait as long as key 0 = 0
4       -3011          increment and display
                       counter 0
2       -4001          wait as long as key 0 = 1
6       -0  2          jump to line 2
```

4.6 DATA (code 5)

The following data values can be defined by the data instruction:

- scale factor
- absolute position register
- motor speed
- acceleration/decelaration
- count of counters 0...9

FORMAT OF DATA CODE 5: | 5 | - | G | # | M | - | DATA/ADDR.

G	1: data for axis 1 2: data for axis 2 3: data for counters 0...9
#	If # = 1 or 2 0: scale factor 1: absolute position register 2: programmable motor speed (VSM) 3: programme time base for acceleration (TB) 4: programmable acceleration (Acc) 5: programmable deceleration (Dec) If # = 3 0...9: counters 0...9
M	0: data entered in the programme 2: data entered from data memory addressing of data memory in the programme 3: read and write of absolute position register or counters 0...9 in the data memory addressing of data memory in the programme 4: data entered from data memory addressing of data memory by means of counters 0...9 5: read and write of absolute position register or counters 0...9 in the data memory addressing of data memory by means of counter 0...9 (refer to explanations on page 20)

Enter data/addresses after ENTER, when display is dark (max. 5 digits plus sign).

Selection G

G= 1 or 2 defines the following data for axis 1 or 2 and S = 3 for counters 0...9.

Address #

Parameters for the stepping motor motion are defined by means of the value A, if S = 1 or 2.

= 0 scale factor

To enable operation with usual units independent of spindle pitches, step-down ratios etc., a corresponding scale factor is programmed. This factor can be entered with sign and decimals. The actually performed number of motor steps is defined as follows:

Step number = distance entered x scale factor

Example:

Distance entered:	200 mm
Scale factor:	3,5
Step number:	200 x 3,5 = 700 Steps

= 1 set absolute position register

To enable the determination of the absolue position referred to a reference point, if required, the absolute position register has to be set to the defined value once (e.g. to 0 at zero).
The value entered may have none, one or two decimals.
The ~~the~~ subsequent display of the absolute position register has the same number of digits behind the decimal point.

= 2 programmable motor speed

The programmable motor speed VSM may be entered by values of 1...255.
The resulting stepping frequency of the stepping motor depends on frequency range FB selected. FB may be internally programmed by means of coding bridges (refer to chapter 1, introduction, unit layout). The following ranges are possible:

FB = 0 to	12 kHz
	6 kHz (factory setting)
	3 kHz
	1.5 kHz
	0.75 kHz

The resulting stepping frequency FR is calculated by:

$$FR\ (Hz) = \frac{FB\ (Hz) \times VSM}{256}$$

Example:

Programmable motor speed VSM = 64
Frequency range FB = 12 000 Hz

$$\text{Resulting stepping frequency FR} = \frac{12\,000\ Hz \times 64}{256} = 3\,000\ Hz$$

= 3 programmable time base for acceleration / deceleration

The acceleration ramp is formed by a time base independent of the stepping frequency. Values from 0 to 7 may be programmed for this time base TB. In almost all applications, the time base does not need to be programmed, since the control automatically selects the value 0 (= 1 ms) when power is put on.

0 = time base TB 1.0 ms
1 = " " 1.125 ms
2 = " " 1.25 ms
3 = " " 1.375 ms
4 = " " 1.5 ms
5 = " " 1.625 ms
6 = " " 1.75 ms
7 = " " 1.875 ms

= 4 programmable acceleration

The programmable acceleration Acc is individually programmed for each motor and can take any value between 1 (minimum acceleration time) and 255 (maximum acceleration time). The acceleration time Tacc is as follows:

$$T_{acc}\ (msec) = VSM \times Acc \times TB\ (msec)$$

Example

Programmable motor speed VSM = 200
Programmable acceleration Acc = 2
Programmable time base TB = 2 = 1.25 ms
Acceleration time Tacc = 200 x 2 x 1.25 ms = 500 ms

= 5 programmable deceleration

The deceleration Dec is individually programmed for each motor and may take any value between 1 (maximum deceleration time) and 255 (minimum deceleration time). The deceleration time Tdec depends on the speed and is as follows:

$$T_{dec}\ (msec) = \frac{VSM \times 5.1 \times 10^5}{Dec \times FB\ (Hz)}$$

Example

Programmable motor speed for VSM = 250
Programmable deceleration Dec = 50
Frequency range FB = 6 kHz

$$\text{Deceleration time } T_{dec} = \frac{250 \times 5.1 \times 10^5}{50 \times 6000} = 425 \text{ msec}$$

In pratice, the optimum speed accelerations and decelerations are determined by tests at the object, since the behaviour of stepping motors greatly depends on mass moments of inertia, friction forces, etc.

Too short deceleration times are to be avoided, since otherwise the programme execution may be blocked. Therefore, the following approximate values shall not be exceeded for Dec:

VSM	FB	DEC	TDEC
255	3 kHz	144	300 ms
255	6 kHz	72	300 ms
255	12 kHz	36	300 ms

Remark

Speeds, accelerations, decelerations, and the time base must not necessarily be defined by programme.
If power is connected, PINX-5 automatically selects the following parameters:

Scale factor:	1
Programmable motor speed for SM:	128
Programmable time base TB:	0
Programmable acceleration ramp Acc:	2
Programmable deceleration Dec:	24

Definition M

The "DATA" instruction includes a data input. This can be made either directly in the programme or from the data memory. In the latter case, there are two possibilities of addressing the data storage location desired:
- addressing of the data memory in the programme
- addressing of the data memory by means of counters 0...9

M = 0 data input in the programme

The data value desired is written into the programme line following the "DATA" instruction.

M = 2 data input from data memory
Addressing of the data memory in the programme

In constrast to the direct data input in the programme, a line number of the data storage range is entered into the following programme line. The data value desired is located in the storage location.

M = 4 data input from data memory
Addressing of the data memory by means of counters 0...9

The address of a counter (0...9) is entered into the following programme line. The count determines the address of the data memory with the data value desired.

The addressing of data memories by means of counters 0...9 enables the user to process different data values by the same data instruction (e.g. by influencing the count and hence the data values by means of external inputs).

M = 3/5 reading and writing of absolute position registers or counters 0...9

The contents of an absolute position register or the count of a counter can be read and stored in the data memory. The addressing of the data memory may be made in the programme (M = 3) or by means of a counter 0...9 (M = 5). This function enables the preparation of a real teach-in programme. Programme examples may be requested.

Example 1

Subsequent to starting by key 0, a positioning (2000 steps) shall be made in fast motion first, and subsequently a second positioning (400 steps) shall be executed in slow motion.

LINE		INSTRUCTION	COMMENT
0	2	- 4000	wait, if key 0 = 0
1	5	- 120-	data input of programmed
2			motor speed
3		255	fast motion value 255
4	1	- 1-11	position by rel. values
5		2000	distance 2000 steps
6			forward
7	5	- 120	data input of programmable
8			motor speed
9		10	slow motion value 10
0	1	- 1-11	position by rel. values
1		-400	distance 400 steps
2			backwards
3	6	- 0 0	jump to line 0
4			
5			

Example 2

A distance input for the positioning by revolutions is desired. After the start by key 0, the stepping motor shall execute 2.5 revolutions.

Calculation of the scale factor:
1 revolution corresponds to 400 steps, thus the scale factor has to be programmed as 400.

LINE		INSTRUCTION	COMMENT
0	5	-100	set scale factor to 400
1		400	
2	2	- 4000	wait, if key 0 = 0
3	1	- 1-11	position by rel. values
4		2.5	2.5 revolutions
5	6	- 0 2	jump to line 2
6			
7			
8			
9			
0			

4.7. JUMP (code 6)

A simple and clear programme structure can be achieved by means of jump functions and subroutines.

FORMAT OF JUMP CODE 6:

6	-	I	X	X	X

I	0: jump to line XXX 1: jump to subroutine in line XXX 2: return from subroutine 3: start simultaneous programme 1 in line XXX 4: start simultaneous programme 2 in line XXX 5: start simultaneous programme 3 in line XXX 6: stop simultaneous programme 1 7: stop simultaneous programme 2 8: stop simultaneous programme 3 9: end of simultaneous programme

Jump I

I = 0 jump to an arbitrary programme line

I = 1 jump into a subroutine starting in "LINE"

I = 2 any subroutine is completed by "RETURN" (2). Returning the programme to the point from which the jump into the subroutine was made.

I = 3/4/5 As you know, PINX-5 sequentially processes the complete programme line for line. To enable a simple realization of functions running in parallel such a monitoring of limit switches, display functions, etc., 3 simultaneous programmes are available that can be processes concurrently with the main programme. I = 3/4/5 means the start of a simultaneous programme, starting in "LINE".

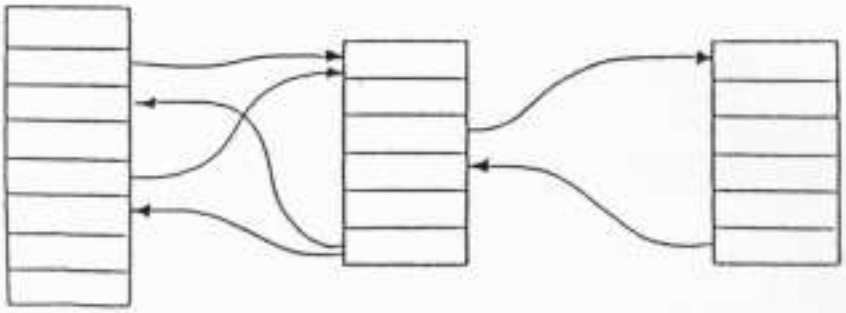
Caution: a simultaneous programme already started must not be started again.

I = 6/7/8 causes the stop of the execution of the respective simultaneous programme

I = 9 a simultaneous programme may be completed by I = 9 (end of simultaneous programme)

A PINX-5 programme may include any number of simultaneous programmes.Take care not to activate more than 3 simultaneous programmes at the same time.

PINX-5 is capable of processing up to 5 nested subroutines.



main programme subroutine 1 subroutine 2

Example

The actual position of the stepping motor shall be displayed by the absolute position register simultaneously to the positioning of 2000 steps. Starting is made by key 0.

LINE		INSTRUCTION		COMMENT
0	5	- 110-		set absolute position
1			0.00	register of axis 1
2	6	- 3 10		start concurrent program
3				display in line 10
4	2	- 4000		wait, if key 0 = 0
5	1	- 1-11		position by rel. values
6			2000	2000 steps forward
7	6	- 0 3		jump to line 3
8				
9				
0				
1	4	- 1101		display of absolute
2				position register, axis 1
3	6	- 0 10		jump to line 10
4				

Data input into the data memory by means of keyboard

The complete storage area (0...599) is locked against modification as long as the locked LED lights. The data memories may be changed by the instruction 6-1600 without disconnection of the programme locked.

If this jump to a subroutine is executed, the first address line of the data memory is displayed and the locked LED is starting to flash at the same time. Then the data in the complete data memory may be changed by means of the keyboard. The input procedure is completed by the start key. Thereby, the programme execution is restarted simultaneously. During the complete process, the programme memory remains locked against changes.

4.8. TIMER (code 7/8)

Any timing function can easily be realized by the "TIMER" instruction. The time is directly entered by seconds. The maximum time is 9999 s.

FORMAT OF TIMER CODE 7:	7	-	X	X	X	X
XXX	time in seconds (from 0.000 to 9999s)					

Code 8

Instruction code 8 is used to realize adjustable times. The line number of the data memory area contains the time value (seconds).

FORMAT OF TIMER CODE 8:	8	-	-	X	X	X
XXX	time from data memory in line XXX					

Example

The positioning of axis 1 by 1500 steps is started by key 0. A fixed waiting period of 1.5 s starts, when this position is reached. Then, a positioning of axis 2 by 1000 steps is executed, and finally, a waiting period starts, the duration of which is defined in data memory 500.

LINE		INSTRUCTION	COMMENT
0	2	- 4000	wait, if key 0 = 0
1	1	- 1-11	position axis 1 by rel.
2			values
3		1500	1500 steps
4	7	1.5	waiting period 1.5 s
5	1	- 2-11	position axis 2 by rel.
6			values
7		1000	1000 steps
8	8	- -500	waiting period with time
9			time value from data
0			memory 500
1	6	- 0 0	jump to line 0
2			

4.9. NOP (code 9)

The following instruction is used as blank comment (NOP)

FORMAT OF NOP CODE 9:	9	-	-	-	-	-

5. General considerations on the programme's structure

As already mentioned in the introduction, the general rules of programme preparation apply to PINX-5 programming.

- o subdivision of the total task into partial tasks
  examples: manual operation
  automatic operation
  reference point travel etc.

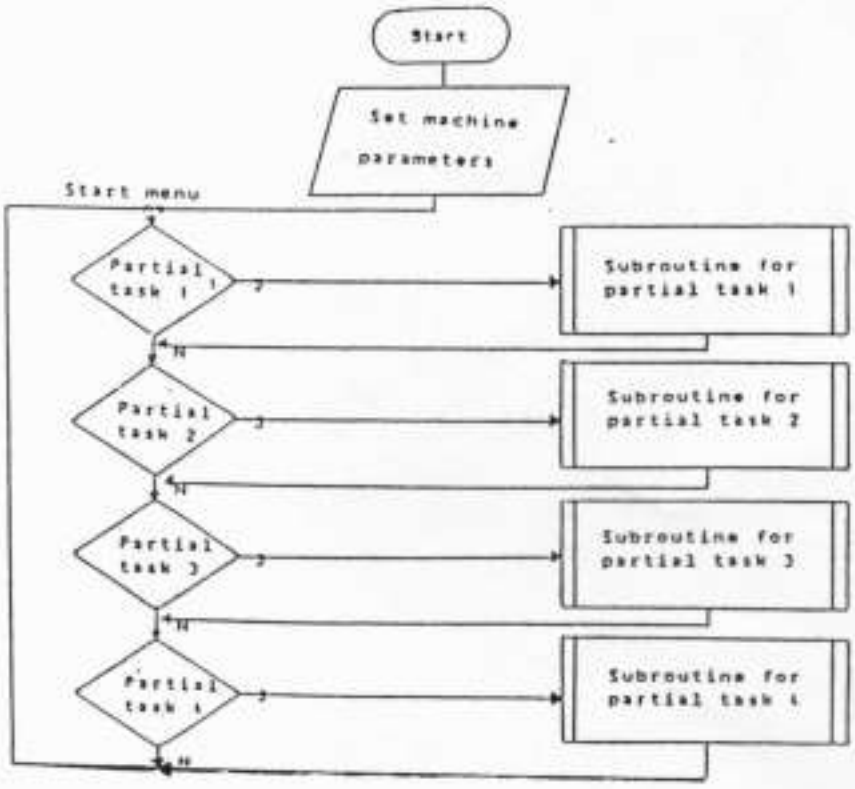
- o definition of the operating possibilities desired (start, top, manual/automatic operation etc.). This question is of great importance, since the structure of the programme can be influenced essentially by the type of operating functions.

- o programme representation by usual types of representations
  examples: flow chart
  structural diagram, etc.

The previous graphic representation of the task is recommended for various reasons:

- less errors in reasoning
- compulsion to structured programme design
- better understanding
- good documentation

- o programming
  It is advisible to enter enough spaces between the individual programme blocks during the programme input to be able to incorporate possible subsequent instructions into the programme. In many cases, the following programme design has been proved:



Set machine parameters

Speeds, accelerations, decelerations, starting of continuously activated concurrent programmes, etc. are programmed in the first programme lines to advantage. Due to the above mentioned programme design, these instructions are positively executed once, when the control is started, and remain unconsidered subsequently.

Start menu

This programme part enables the user to have the desired partial tasks executed. Since all partial tasks are stored in subroutines, a return to the start menu is always ensured. This prevents the programme execution running on endlessly without it being possible to intervene.

Subroutines for partial tasks

The use of subroutines mainly offers the possibility to realize a clear programme design. Furthermore, programme parts used several times, may be started from an arbitrary position without impairing the clearness.

A sample programme ensuring the described programme parts follows.

In lines 0 to 7, the machine parameters are set. In lines 1 to 16, the start menu is located, enabling the call of individual subroutines. Manual operation forward/backward, automatic operation with two positioning distances as well as running to zero are programmed in the subroutines.

Operation of sample programme

- 'STEP' key: manual operation ON/OFF
  '.' key: forward motor motion
  '-' key: backwardmotor motion
- 'START'key: start of automatic operation
- 'CLEAR'key: start of running to zero

Programme structure

LINE		INSTRUCTIONS	COMMENTS
00	5	- 100-	Scale factor
1		10	
2	5	- 140-	progr. acceleration machine
3		5	para-
4	5	- 150-	progr. deceleration meter
5		20	
6	5	- 110-	set absolute position
7		0.00	register to 0
8			
9			"start menu"
10	2	- 5610	"STEP" key = man. operation (ON)
1	6	- 1 20	jump to subroutine manual
2	2	- 5510	"START" key = autom. operation
3	6	- 1 30	jump to subroutine auto
4	2	- 5410	"CLEAR" key = back to zero
5	6	- 1 40	jump to subroutine zero
6	6	- 0 10	jump to line 10
7			
8			
9		subroutine manual operation	
20	2	- 5601	"STEP" key = man. operation
1	2	- 5010	"," key = Move motor forward
2	0	- 1-10	forward motor motion
3	2	- 5910	"-" key = move motor backward
4	0	- 1-11	backward motor motion
5	2	- 5611	"STEP" key = man. operation (OFF)
6	6	- 0 20	jump to line 20
7	2	- 5601	"STEP" key = man. operation
8	6	- 2---	return to start menu
9		subroutine automatic operation	
30	1	- 1-11	position axis 1 by rel. values
1		1250	distance 1250
2	7	1.50	waiting period 1.5 s
3	1	- 1-11	position axis 1 by rel. values
4		-500	distance -500
5	6	- 2	return to start menu
6			
7			
8			
9		subroutine return to zero	
40	1	- 1-1	position axis 1 by absolute values
1		0	to zero
2	6	- 2	return to start menu
3			
4			
5			
6			
7			

6. PINX-5: Liaison adapter, interface

6.1. General

The adapter, or interface has 2 channels:

serial interface RS 232
audio interface for tape recorder connection

The serial interface is used for communication with printers or other peripheral units.

The output forward is designed for an easily readable documentation. Baud rate and transmission format may be selected by the user and are entered my means of the frontal keyboard.
Moreover, kind of representation (50 or 100 lines per A4 page) as well as extent of programme to be printed may be defined at the beginning.

The audio interface installed transfers data by means of frequency modulated signals. Thereby, a high reliability of data transfer is achieved even for simple tape recorders. In contrast to the RS 232 interface, the complete memory contents of the control is transfered through the audio interface.

Important: The programme lock has to be switched off for all data transfert functions ("LOCKED" LED dark)

6.2. Data transmission of audio interface

Three different transmission functions are possible on principle:

- write storage contents onto tape
- read storage contents from tape
- compare storage contents to tape

Write storage contents onto tape

Procedure:
- connect tape recorder at interface connector according to item 6.4, connector
- select play mode at tape recorder
- select line 902 at PINX-5
- press "START" key

Read storage contents from tape

Procedure:
- connect tape recorder at interface connector according to item 6.4, connector
- select play mode at tape recorder
- select line 903 at PINX-5
- press "START" key

Compare storage contents to tape

Procedure:
- o connect tape recorder at interface connector according to item 6.4, connector
- o select play mode on tape recorder
- o select line 904 on PINX-5
- o press "START" key

General remarks

During all transmission procedures, the programme line currently transmitted is displayed. If a transmission error occurs, the transmission process is stopped and the incorrectly transmitted programme line is displayed.

If several programmes are to be stored on the same tape cassette, it is advisible to prepare an index with the corresponding tape counts. Any recording starts with a leader without data transmission with a duration of approx. 1 s.

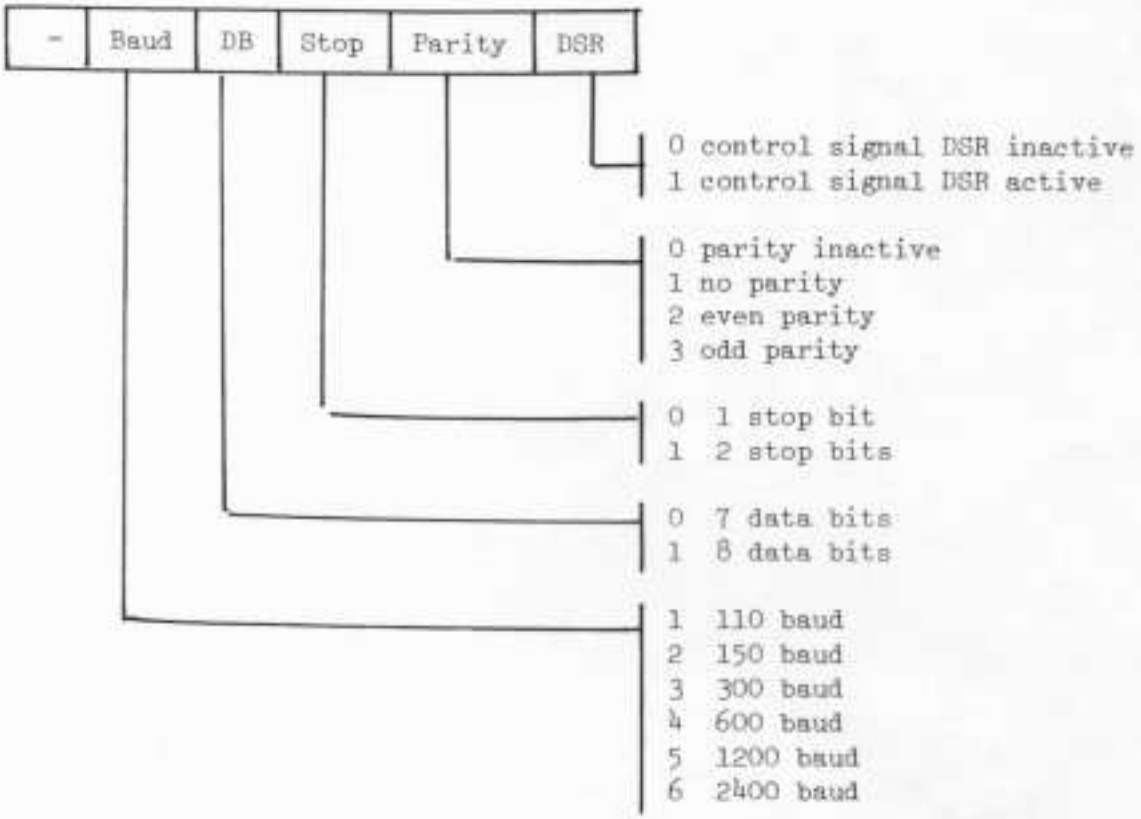
Electric specifications

Kind of transmission: 2-phase frequency modulation
Baud rate: 300 baud
Carrier frequency: 5 kHz
Transmission level: 300 mVpp
Received level: 200...8000 mVpp

6.3. Data transmission, RS 232 interface

Transmission parameters

All important transmission parameters are defined in programme line 601.



Control signal DSR

The control signal DSR is exclusively used, when data are transmitted by PINX-5. DSR is used to stop the transmission process, if the peripheral unit connected is not ready to receive data.
Connection is made according to item 6.4, connectors.

Kind of representation

A programme is printed page by page in the format A4.
The following kinds of representations may be selected.

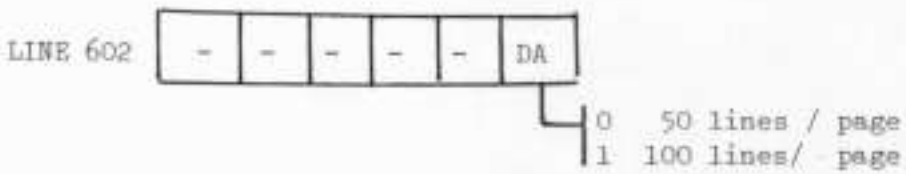
o 50 programme lines per page

LINE	INST./DATA	COMMENTS
000	3-10-0	
001	2-1010	
002	3-10-1	
003	3-11-0	
004	2-1110	
005	2-11-1	
006	3-12-0	

o 100 programme lines per page

LINE	INST./DATA	COMMENTS	LINE	INST./DATA	COMMENTS
100	9-----		150	9-----	
101	9-----		151	9-----	
102	9-----		152	9-----	
103	9-----		153	9-----	
104	9-----		154	9-----	
105	9-----		155	9-----	

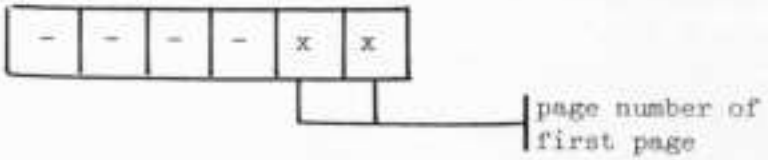
The kind of representation is defined in line 602:



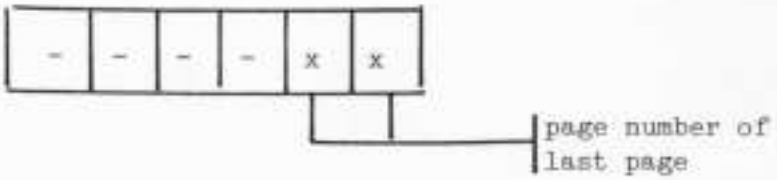
Extent of print out

The extent of the programme to be printed out can be defined by an input in lines 603 and 604:

LINE 603:



LINE 604:



Reading to programmes via RS 232 C

"Carriage return" (CR) has to be transmitted after the transmission of any instruction. The programme line does not need to be transmitted, too. Spaces are ignored.

Example

Transmitted ASCII-characters:

0:6320(CR)5:41101(CR):7.05(CR):605(CR)

PINX-5 PROGRAMME

LINE		INSTRUCTION	COMMENTS
0	6	- 3 20	
1	6	- 0 1	
2			
3			
4			
5	4	- 1101	
6	7	- 0.05	
7	6	- 0 5	
8			
9			

ASCII-characters (e.g. comments) not relevant to the PINX-5 programme may be transmitted between semicolon (;) and carriage return (CR).

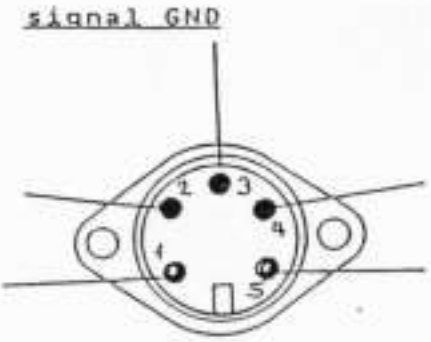
Start of transmission process

Procedure:

- o connect periphery unit according to item 6.4 connectors, to PINX-5
- o select line 900 to print out data
  select line 901 to read data via RS 232
- o press "START" key

6.4. Connector

The connector for the two interfaces is located at the rear of the unit below the power cord.



data output audio interface

TXD data output RS 232

data input audio interface

RXD data input RS 232 or DSR, if data are transmitted via RS 232

Note on wiring

The data output of the audio interface is connected to the microphone input (MIC) of the tape recorder and the data input of the audio interface is connected to the earphone output (EAR) of the tape recorder.

PINX-5: One axis and two axes

The new PINX-5 Version NC completes the family of proven versatile controllers.

Direct communication with master control computer.

There is an increasing requirement for communication between computers and intelligent positionning devices. For example:

- o Large scale systems with central control and satelite positionning devices (DNC operation).
- o Production systems requiring frequent changes.
- o Computer controlled measuring and test systems.

PINX-5, one axis version and two axes version, meet these requirements by providing handshake facilities. This allows programme instructions and data to be transmitted from the control computer to the controller via integral RS 232 C channel.

PROGRAMMING

To switch from programme execution to data reception, the following instruction must be included in the programme whenever new programme instructions or data need to be transmitted.

6 - 0901

When the PINX-5 processor meets this instruction it initializes the data reception routine and transmits a $ character to the control computer to start the transmission of data.

NOTE: The computer must delay transmission for a minimum of 2msec after receiving the $ character.

DATA TRANSMISSION FROM THE COMPUTER

TRANSMISSION PARAMETERS

For the method of entering transmission parameters see PINX-5 Manual, paragraph 6.3.

DATA TRANSMISSION

The following can be transmitted:

- Complete programmes
- Programme sections
- Individual instructions
- Data sections
- Individual data value

To terminate data transmission and restart programme execution a line number followed by a $ character must be transmitted. Programme execution will start at the defined line number.

EXPLANATION OF CONTROL CHARACTERS

Character transmitted to PINX-5	Equivalent to pressing KEY
CR (Carriage return)	ENTER followed by LINE
: (colon)	EXAM
, (coma)	ENTER followed by EXAM
$ (Dollar)	START

NOTE: All characters between a semi colon (;) and a carriage return (CR) are ignored by PINX-5 (e.g. computer programme commentary).

Examples:

Transmitting an individual positionning instruction

(CR) 10: 1111 (CR): 2500 (CR): 60901 (CR) 10 $

PINX-5 memory contents:

```
010  1-1-11  Increment motor / (relative data)
011    2500  Data 2500 steps
012  6-6901  Jump line 901 (receive new data)
```

Programme starts at line 10.

Transmitting data to the data memory section

(CR) 500: 175.5 CR: 1500 (CR): -220 (CR) 100 $

PINX-5 memory contents:

500	175.5
501	1500
502	-220

Programme starts at line 100.

STARTING DIFFERENT PROGRAMME SECTIONS IN PINX-5

- Computer transmits to PINX-5

  (CR) 10 $
  (Programme execution starts at line 10 and stops at first 6-0901 instruction).

- PINX-5 transmits to computer:

  $
  (DSR, instruction to computer to transmit new data)

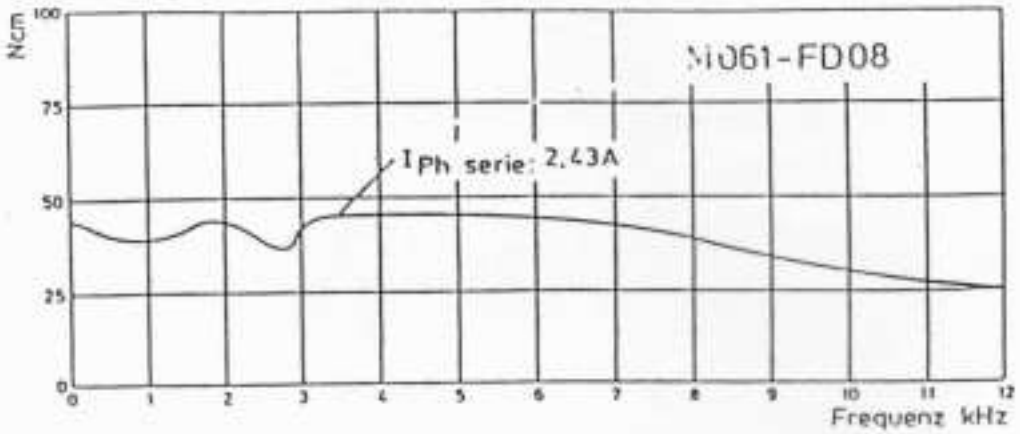
- Compter transmits to PINX-5
  (CR) 100 $
  (Programme execution starts at line 100 and stops at next instruction 6-0901)

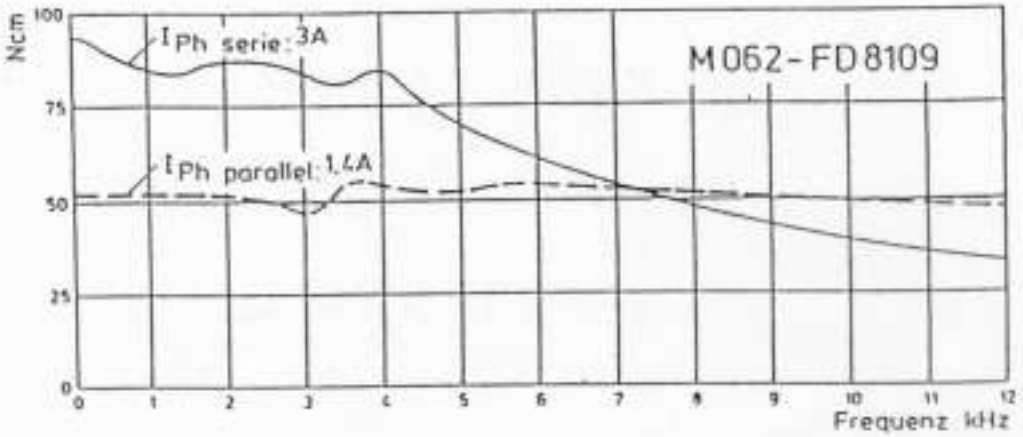
- PINX-5 transmits to computer:
  $
  (DSR, instruction to computer to transmit new data).

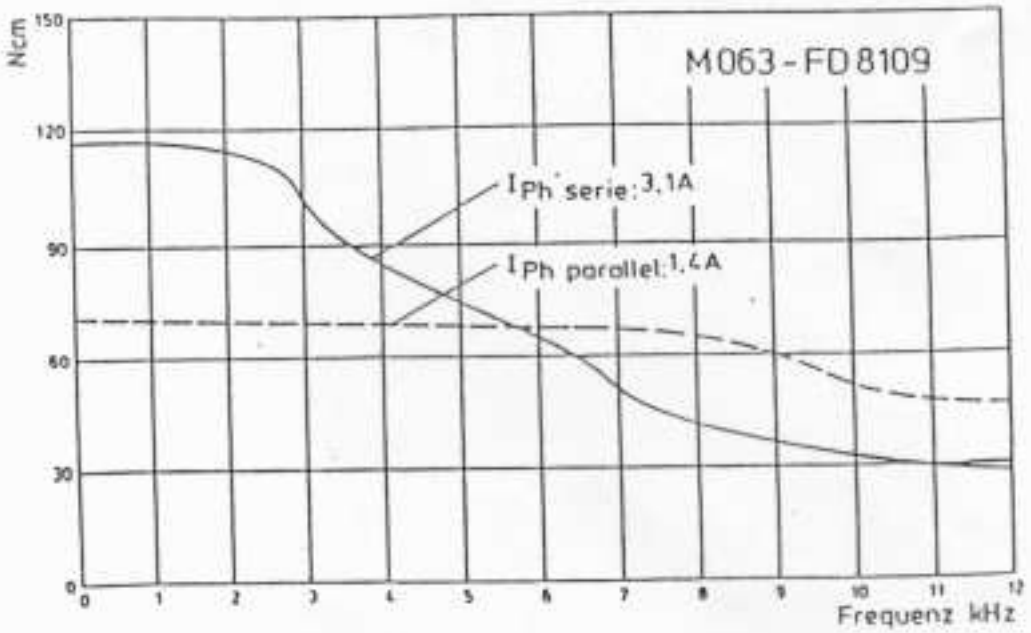
7. Troubleshooting list

Fault	Possible cause	Remedy
PINX-5 does not accept inputs by keys	Programme lock switched on	Switch off programme lock (refer to chapter 3.8. LOCK)
PINX-5 does not accept any input of instructions	Display shows programme line	Switch over to instruction display by means of EXAM key
An instruction entered does not appear in the correct format in the display	An index or data instruction without corresponding data or address value was programmed in the previous programme line	Correct programme
PINX-5 does not accept instruction inputs form a certain line	This line is intended as border line between programme and data memory	Enter correct border line number in line 600 (refer to chapter 4.7 JUMP)
The programme execution is incorrect	Subroutines used are not completed by the instruction 6-2---	Check and modify programme
	Incorrect programme input	
	Memory was not cleared prior to programme input	Owerwrite superfluous instruction by NOP
Programme execution is blocked after positioning process	Deceleration selected is to great	Programme new deceleration (refer to chapter 4.6 DATA)
The programme is executed but once, subsequently a programme line appears in the display	The programme is not closed in itself	A return to the start of the programme has to be entered at the end of the programme
Concurrent programmes are not executed correctly	The concurrent programme is restartet during execution	Modify programme
Positioning is executed during the first programme run only	A positioning to be performed several times is programmed by absolute values	Programme positioning by relative values

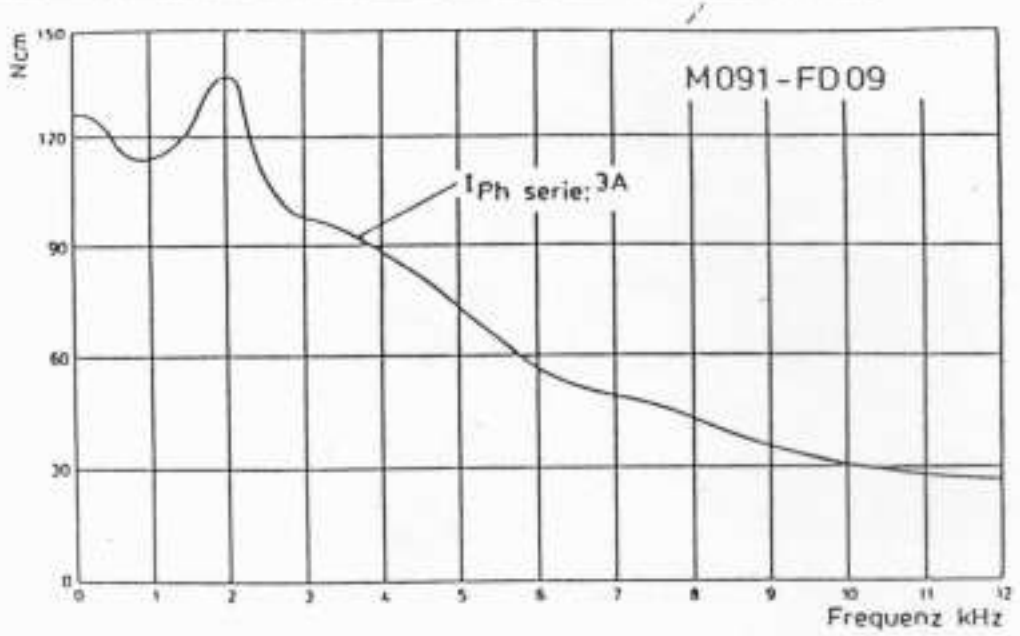
Torque characteristics of PINX-5, one and two axes versions

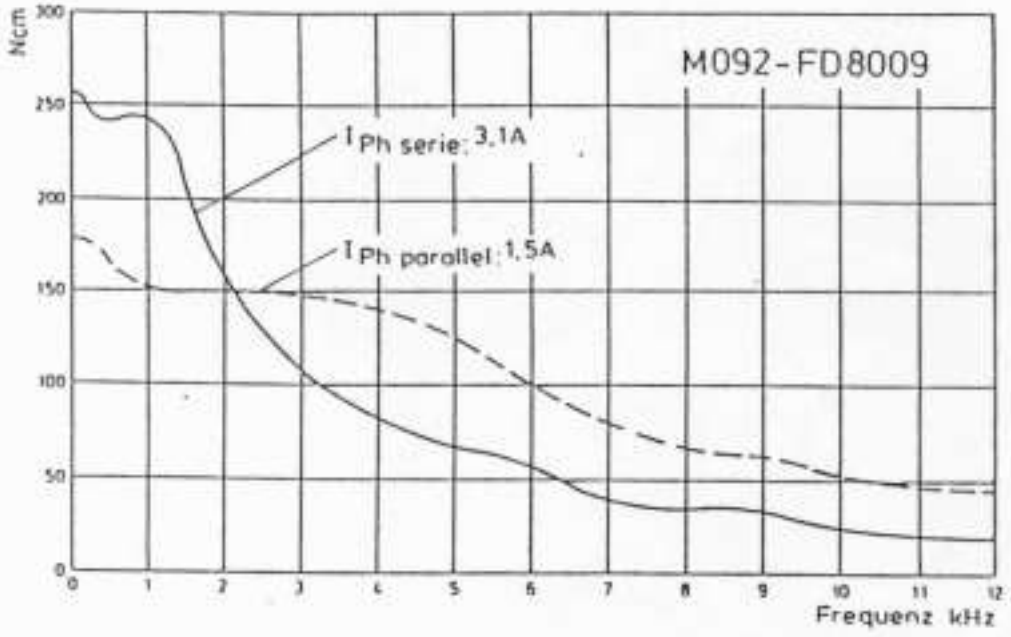


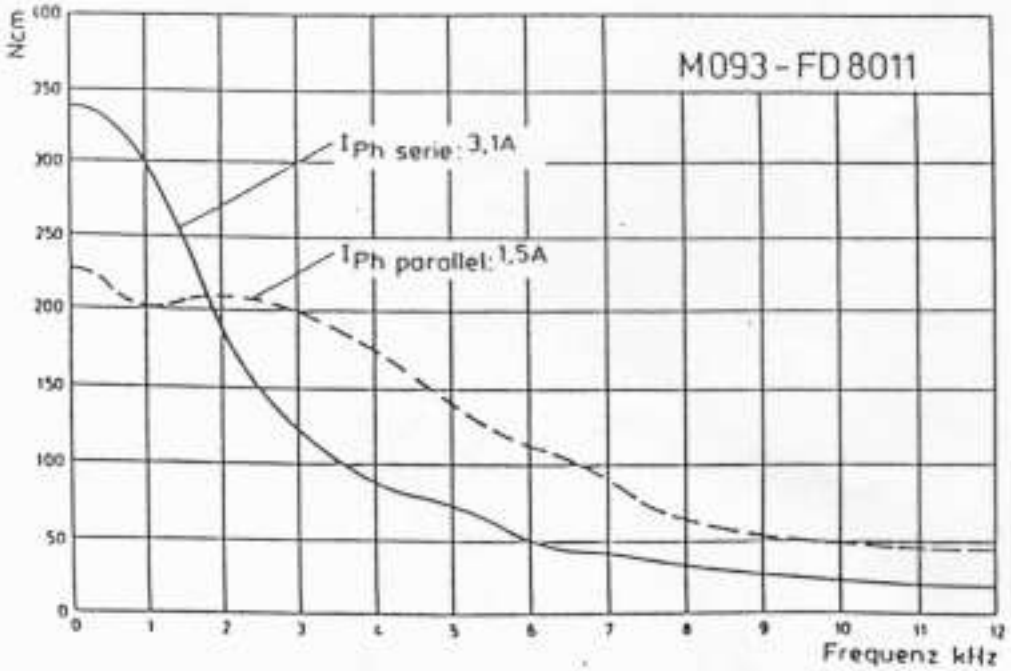




Torque characteristics of PINX-5, one and two axes versions







Torque characteristics of PINX-5, one and two axes versions