It is user RESPONSIBILITY to check that this manual (in PDF format) refers to product model and version that will be used.

In any case, regarding installation, use and maintenance, the paper-made manual given together with the product TAKES PRIORITY.

R.T.A. srl

MOTION CONTROL SYSTEMS

X-MIND K SERIES

STEPPING MOTOR DRIVES

PROGRAMMER'S MANUAL

R.T.A. s.r.l.

Via E. Mattei – Frazione DIVISA
27020 MARCIGNAGO (PV)
Tel. +39.0382.929.855 - Fax +39.0382.929.150
Internet: *http://www.rta.it* - e-mail: *info@rta.it*

R.T.A.
DEUTSCHLAND GmbH

Bublitzer Straße, 34
40599 DÜSSELDORF (Germany)
Tel. +49.211.749.668.60-Fax +49.211.749.668.66
Internet: *http://www.rta-deutschland.de*
e-mail: *info@rta-deutschland.de*

R.T.A. IBERICA
MOTION CONTROL SYSTEMS S.L.

C/Generalitat 22, 1° 3°
08850 GAVA – BARCELONA (Spain)
Tel. +34.936.388.805-Fax +34.936.334.595
Internet: *http://www.rta-iberica.es*
e-mail: *info@rta-iberica.es*

R.T.A. s.r.l. – SW03ME03 – 02/09

INDEX

1- WARNINGS AND MANUAL USE

This manual has been realized in order to give, together with the instruction manual, all the necessary information to program and manage X-MIND K series drives. R.T.A. recommends a **careful reading first of the instruction manual and then of this manual.**

R.T.A. deeply suggests to new users of X-MIND K series drive the purchase of **starter kit** including:

- a software (**X-MIND K application program**), equipped with a complete and easy to consult on line help, that can be installed on a PC using Windows 98/XP. X-MIND K application program is a powerful and easy to use instrument to program X-MIND K series drives and to make proofs.
- an adapter to connect the drive to a USB port of a PC.

Using X-*MIND K application program,* the managing of the serial communication and the programs writing is very easy.

This manual is organized as follows:

- chapter 2: main features of X-MIND K series drives.
- chapter 3 and 4: memory storing, commands and instructions.
- chapter 5: X-MIND K series drives operation modes.
- chapter 6: code tables.

2- MAIN FEATURES, INPUTS, OUTPUTS AND SETTING OF X-MIND K SERIES DRIVES

2.1- MAIN FEATURES

Most important features of X-MIND K series drives are:

- 48 available drive addresse setting by user. Therefore it is possible to connect up to 48 drives in parallel on a single serial line (daisy-chain).
- Connection using RS485.
- Permanent storing of 128 instructions in 128 memory cells, each defined by its address: an integer number between 100 and 227.
- Storing of 3 instructions in RAM in 3 memory cells, each defined by its address: an integer number between 228 and 230.
- 20 different types of recognized instructions.
- Execution of all previously stored instructions both singly and in prearranged sequences (programs) through a start command on the serial line.
- Execution of one of the first 16 stored instructions (from 100 to 115) both singly and in prearranged sequences (programs) using a start hardware command.
- 8 hardware inputs with a fixed use and 3 hardware inputs whose status can be set by the user.
- 2 hardware outputs with a fixed use and 4 hardware outputs whose status can be set by the use.
- 8 dummy outputs (flags) whose status can be set by the user.

2.2- INPUTS AND OUTPUTS

In this chapter the signal part of the front panel of X-MIND K series drive is described: connectors RJ-12 S1 and S2, connectors C2 IN and C3 OUT. Inputs and outputs hardware characteristics and the power part (LEDs, connector C1) are described in the instruction manual.

2.2.1- Connectors RJ-12 S1 and S2 (Serial input/output)

Connectors RJ-12 are used to establish the serial communication between the drive and a PC or a PLC. S1 and S2 are identical and in parallel, so there is no difference between them: the presence of two connectors instead of one allows a daisy-chain connection. Every single drive in a daisy-chain connection is identified by its drive address which is an integer number between 0 and 47. The drive address appears in the first position of all the commands transmitted to the drive. Drive address is composed by two characters (therefore, for example, 7 has to be written as 07) and can be set by means of “ADDRESS PRESET” command. A transmission with drive address 99 corresponds to a broadcast command, that is a command recognized and executed by all X-MIND K series drives connected to serial line. In case of broadcast command, X-MIND K series drives connected to serial line don’t answer with any message. Consider that broadcast transmission is not allowed for all the commands; all command characteristics are described in chapter 4. Hardware characteristics of S1 and S2 connectors are described in the instruction manual.

2.2.2- Connector C2 (inputs)

Connector C2 includes all hardware inputs. Through them the user can activate some commands or previously stored instructions. For all inputs in the following we indicate the corresponding numbers of connector C2.

COMMON OF INPUTS.

12, 13, 14, 15: (SEL0, SEL1, SEL2, SEL3): selection of one of the 16 programs that can be activated through the hardware input ST.

16 (ST): START: start of previously stored programs. The program starts on OFF-ON transition of this input at these conditions:
- Start command is rejected when EE is ON
- Start command can be rejected or not accordingly to setting of "PRESET OF ES PRIORITY" when ES is ON (see chap. 4.2.2.7).

17 (EE): EMERGENCY STOP. Stop of the execution of any instruction or command, immediate block of any program running.

18 (ES): FREE RUN STOP. Stop of the execution of a free run instruction (both with and without ramp).

19 (I0): I0 INPUT. Input terminal whose status can be read by a command or instruction.

20 (I1): I1 INPUT. Input terminal whose status can be read by a command or instruction.

21 (PX): PROXIMITY input. Used for zero research procedure.

22 (I2): I2 INPUT. Input terminal whose status can be read by a command or instruction.

23: Drive GND. PAY ATTENTION: this terminal is internally connected to 6 terminal of C1 connector (see chap. 5) and insulated from input terminals. It can be used when shield of input signals has to be connected to the earth.

2.2.3- Hardware inputs use

Consider following information about hardware inputs use:

- An input is ON when at its terminals is present a voltage within the following limits:

  Vin_{MIN} = 5 Volt

  Vin_{MAX} = 24 Volt

- **I0, I1** and **I2** inputs can be read directly using the serial line or with a proper instruction previously stored in the drive.
- **PX** input can be used only for the execution of a *Zero Search Procedure*.
- **ES** input can be used only for a *Free Run Stop* and for an instruction *Delayed Stop*.
- **EE** input can be used only for the blocking of the execution of any single instruction or program (in this case the whole program is blocked).
- **SEL0, SEL1, SEL2, SEL3** inputs together with ST input are used for the via hardware activation of previously stored programs. It is possible to store up to 16 runs (or programs) that can be activated using ST input. In this context a program is defined as a sequence of stored instructions; the syntax of all the instructions includes the address of the next instruction in the sequence or the indication that the sequence ends with the current instruction. The first instruction of a sequence has to be stored in a memory cell whose address is between 100 and 115. To start a stored program the user must first select (using SEL0, SEL1, SEL2, SEL3) the memory cell containing the first instruction of the sequence and then he has to activate ST input that starts the program execution. The relationship between memory cells and SEL0, SEL1, SEL2, SEL3 status is showed in Table 1:

Inputs				Memory cell
SEL0	**SEL1**	**SEL2**	**SEL3**	
OFF	*OFF*	*OFF*	*OFF*	100
ON	*OFF*	*OFF*	*OFF*	101
OFF	ON	*OFF*	*OFF*	102
ON	ON	*OFF*	*OFF*	103
OFF	*OFF*	ON	*OFF*	104
ON	*OFF*	ON	*OFF*	105
OFF	ON	ON	*OFF*	106
ON	ON	ON	*OFF*	107
OFF	*OFF*	*OFF*	ON	108
ON	*OFF*	*OFF*	ON	109
OFF	ON	*OFF*	ON	110
ON	ON	*OFF*	ON	111
OFF	*OFF*	ON	ON	112
ON	*OFF*	ON	ON	113
OFF	ON	ON	ON	114
ON	ON	ON	ON	115

Table 1: Selection inputs

The stored program activation timing is described in Figure 1.

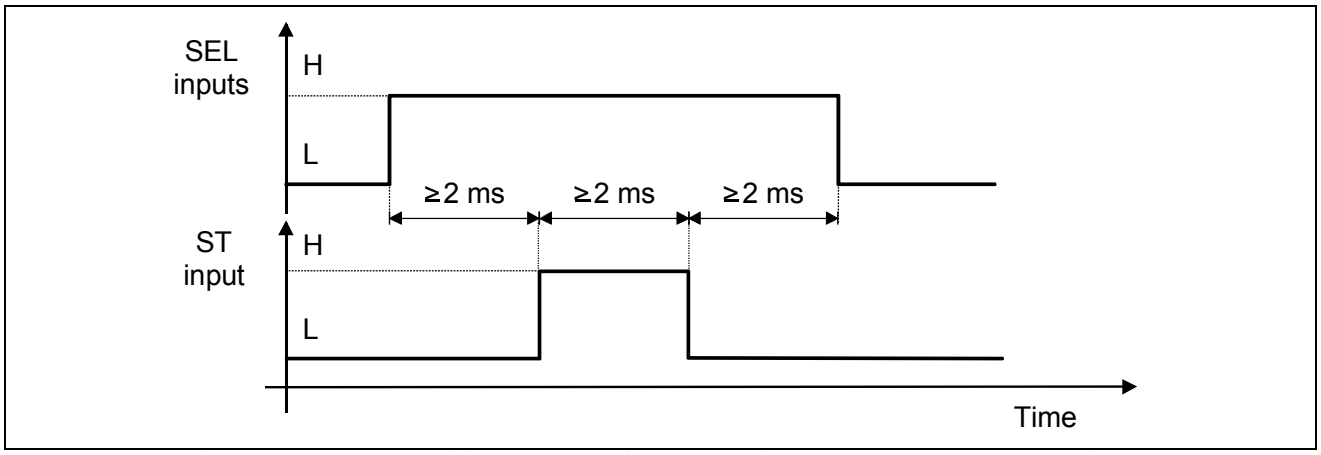


Fig. 1: Timing of the status change of ST input and SEL inputs for via hardware activation of previously stored programs.

As shown in figure 1, the user has to set selection (SEL) inputs to the established value at least 2 milliseconds before the activation of ST input which has to be maintained at the high status for at least 2 milliseconds. SEL inputs have to remain in this state for at least 2 milliseconds after the change to the low level of ST input.

2.2.4- Connector C3 (outputs)

Connector C3 includes all the hardware outputs. Through them the user can obtain information about the internal functioning of the drive. The drive has four hardware outputs that can be changed by the user. For all outputs in the following we indicate the corresponding numbers of connector C3.

31, 36: **COMMON OF OUTPUTS.**

32 (FA): **FAULT.** FA output is ON when drive is working correctly, OFF when it is blocked by a protection.

33 (BS): **BUSY.** BS output is ON when drive is executing an instruction, OFF when drive is waiting or ready to receive a command.

34 (O0): **O0 OUTPUT.** Hardware output whose status can be set by the user using a command or an instruction.

35 (O1): **O1 OUTPUT.** Hardware output whose status can be set by the user using a command or an instruction.

37 (O2): **O2 OUTPUT.** Hardware output whose status can be set by the user using a command or an instruction.

38 (O3): **O3 OUTPUT.** Hardware output whose status can be set by the user using a command or an instruction.

39: **Drive GND.** PAY ATTENTION: this terminal is internally connected to 6 terminal of C1 connector (see chap. 5) and insulated from input terminals. It can be used when shield of output signals has to be connected to the earth.

Consider the following information about hardware outputs:

- an output is ON when its terminal is closed to common of outputs.
- FA (FAULT) output is ON when drive is working correctly, OFF when it is blocked by a protection.
- BS (BUSY) output is ON when drive is executing an instruction, OFF when drive is waiting and it is ready to receive a command.
- **O0, O1, O2** and **O3** status can be changed by the user using whether a command through the serial line or an instruction previously stored in the drive.

NOTE: in X-MIND K series drives eight dummy outputs are available: F0, F1, F2, F3, F4, F5, F6 and F7. For commands and instructions dummy outputs have exactly the same properties of O0 ÷ O3 hardware outputs; even if flags, then outputs F0÷F7, are not related to any pin of C3 connector.

3- SERIAL COMMUNICATION MANAGEMENT

The communication through the serial line RS485 between control system (PC or PLC) and X-MIND K series drives have a baud rate of 9600 (half duplex). Every transmitted single element of a string (corresponding to an ASCII character) is composed by ten bits: one start bit, eight bit of the ASCII code and a stop bit. There is no parity bit.
Control system transmits to drives ASCII character strings whose last character has always to be an end-of-transmission character (ASCII code 13). Drives can hold the serial line only if it has been specifically requested by the control system. If the control system transmits a command or an instruction to a busy drive (for example during the execution of an instruction) the drive gives no answer unless the transmitted command is a STOP command (EE or ES code, see next chapters).
It is very important a correct managing of the serial line timing. Particularly, for any character string transmitted, the control system has to wait for the answer of the drive. Consider that any single character holds the serial line for approximately 1.04 milliseconds. The characters of a string have to be transmitted in sequence without a waiting time among them. Every string has to end with an end-of-transmission character (ASCII code 13) that has to be counted as a transmitted character. Between the transmission of the last character of the string and the transmission of the first character of the corresponding answer there is a time of 1.5 milliseconds. That is true for all the commands with the exception of the memory storing command (WN code): in this case this waiting time is 12 milliseconds. Also the answer strings end with the end-of-transmission character (ASCII code 13) that has to be counted as a transmitted character. To the total calculated time of serial line holding the user has to add a 4 milliseconds margin before the transmission of another string. Consider that the duration of the answer depends (as the transmitted command duration) on the number of characters composing the string: if the user does not know the length of an answer, he has to consider the time corresponding to the maximum possible length.
However the problem of the timing computation can be easily solved simply waiting for the answer of a transmitted command before sending the next one.
In the following there are four examples of strings transmission with correct timing computation:

1) Suppose to transmit the following string (stored instruction execution):

00PS,100↵

where ↵ = is the end-of-transmission character (ASCII code 13).
As a consequence of the reception of this string, the drive transmits the following confirmation message:

00Y↵

Let's calculate the timing of the whole operation: the string transmitted by the control system is composed by nine characters (end-of-transmission character included), therefore it holds the serial line for approximately $1.04 \times 9 = 9.36$ milliseconds. After this time there is a 1.5 milliseconds interval before the transmission of the answer. The answer holds the serial line for about $4 \times 1.04 = 4.16$ milliseconds (four characters including the ASCII code 13 character). Adding the 4 milliseconds margin, the whole operation needs about:

9.36 + 1.5 + 4.16 + 4 = 19.02 ms

2) As a second example suppose to transmit the following character string (memory storing command):

00WN,100,01,10,18,+10000,x1,000↵

whose correct answer from the drive is the following confirmation message:

00Y↵

The transmission of the command, composed by 32 characters (including the end-of-transmission character) holds the serial line for approximately 1.04 x 32 = 33.28 milliseconds. After this time there is a waiting time of 12 milliseconds (it is a memory storing command) before the transmission of the answer from the drive. The answer holds the serial line for about 4 x 1.04 = 4.16 milliseconds (4 characters including the end-of-transmission character). To complete the computation we have to consider the 4 milliseconds margin. Therefore the whole operation needs about:

33.28 + 12 + 4.16 + 4 = 53.44 ms

3) As a third example, suppose to transmit the following character string (content of a memory cell request):

00QM,140↵

whose answer could be, for example:

00QM,140,01,05,15,+100000,x1,141↵

The transmission of the command string (nine character including the end-of-transmission character) holds the serial line for about 1.04 × 9 = 9.36 milliseconds. After that, a waiting time of 1.5 milliseconds has to be counted before the transmission by the drive of the corresponding answer. The latter holds the serial line for about 33 × 1.04 = 34.32 milliseconds (33 characters including the end-of-transmission character). Considering also the 4 milliseconds margin, the whole operation needs approximately:

9.36 + 1.5 + 34.32 + 4 = 49.18 ms

4) As a fourth example, suppose to transmit the following string (absolute position request):

08QA↵

whose corresponding answer could be, for example:

08QA,+10854000↵

The transmission of the first string needs approximately 5 × 1.04 = 5.2 milliseconds. After 1.5 milliseconds the drive transmits the answer which needs about 15 × 1.04 = 15.6 milliseconds. To this time we add the 4 milliseconds margin. Therefore the total time is about 26.3 milliseconds.

In case of the receiving of this answer:

08QA,+0↵

the total time is about 19.02 milliseconds.

4- INSTRUCTIONS, COMMANDS, DATA REQUESTS AND MESSAGES

To classify the strings that can be transmitted to the drive we can define 4 different typologies of strings:

- **INSTRUCTIONS** (from control system to drive): strings containing parameters that can be stored in the drive. An instruction can be recalled from the memory and executed using a proper command.
- **COMMANDS** (from control system to drive): strings corresponding to immediately executed actions. Preset commands belong to this type of string. Different preset parameters values change the instructions' and commands' behaviour in accordance with assigned value, as then it will be described in detail.
- **DATA REQUESTS AND ANSWERS** (*data requests*: from control system to drive, *answer*: from drive to control system): *Data requests* are strings containing a request of the value of parameters, status of outputs etc... *Answers* are strings containing the data requested from the control system.
- **CONFIRMATION MESSAGES** (from the drive to the control system): strings communicating the correct (or not correct) reception by the drive of a command or instruction.

All the strings has to begin with the drive address and end with the end-of-transmission character (ASCII code number 13).

4.1- INSTRUCTIONS

An instruction is an action (run, conditional jump, etc…) whose parameters can be stored in one of 131 memory cells of the drive using a memory storing command (WN code). An instruction can be transmitted only when drive is not executing a command. Broadcast transmission is not allowed for instructions. The syntax of all the instructions is the following:

AAWN,BBB,CC,ppppp…,DDD↵

where:

- **AA** = drive address (number between 00 and 47).
- **WN** = memory storing command code.
- **BBB** = memory cell address where the instruction has to be stored (number between 100 and 227 included for storing in E2prom, between 228 and 230 included for storing in RAM).
- **CC** = instruction code (see chap.4.1.1).
- **ppppp…** = instruction parameters (depending on the particular instruction).
- **DDD** = memory cell address where it is stored the next instruction in the sequence (number between 100 and 230 included). If the program ends with this instruction, then DDD = 000.
- ↵ = end-of-transmission character (ASCII code 13).

After the receiving of an instruction the drive transmits a confirmation message whose syntax is described in par. 4.4.
In the following the list of all the instructions that can be stored in the drive is showed. For all the instructions we indicate the code, the parameters and an example.

4.1.1- INDEXED RUN WITH RAMP INSTRUCTION

NOTE: This instruction has been kept for retro-compatibility with MIND and HI-MOD series drives. RTA suggests to use the new instruction "HIGH RESOLUTION INDEXED RUN WITH RAMP".

Movement with the following parameters: number of steps, direction, frequency, acceleration and deceleration ramp time and x4 multiplier setting. If an *Indexed run with ramp* is stopped by an emergency stop command (hardware or through the serial line), at the receiving of the stop command the drive begins the deceleration ramp.

Code: **CC = 01**

Instruction parameters: **ppppp… = FF,RR,+nnn..,XX** where:

- **FF =** maximum frequency code (number between 03 e 24, see Table 2).
- **RR =** ramp code (number between 10 and 42, see Table 5).
- **+nnn.. =** integer number between –8388607 e +8388607 indicating the total number of steps. This field can be composed by a variable number of characters.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

  Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

Example

07WN,156,01,09,15,+31754,x1,121↵

Meaning of the string: in memory cell number 156 of the drive whose address is 07 store an *Indexed run with ramp* instruction of 31754 steps with a maximum frequency of 9 kHz, ramp code 15, positive direction and no activation of x4 multiplier. Afterwards continue executing the content of memory cell number 121.

4.1.2- HIGH RESOLUTION INDEXED RUN WITH RAMP INSTRUCTION

Movement with the following parameters: number of steps, direction, frequency, acceleration and deceleration ramp time and x4 multiplier setting. If an *High resolution indexed run with ramp* is stopped by an emergency stop command (hardware or through the serial line), at the receiving of the stop command the drive begins the deceleration ramp.

Code: **CC = 31**

Instruction parameters: **ppppp… = FFF,RR,+nnn..,XX** where:

- **FFF =** maximum step frequency in Hz × 100 (number between 030 and 240 included).
- **RR =** ramp code (number between 10 and 42, see Table 5).
- **+nnn.. =** integer number between –8388607 e +8388607 indicating the total number of steps. This field can be composed by a variable number of characters.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

  Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

Example

07WN,156,31,091,15,+31754,x1,121↵

Meaning of the string: in memory cell number 156 of the drive whose address is 07 store an *High resolution indexed run with ramp* instruction of 31754 steps with a maximum frequency of 9100 Hz, ramp code 15, positive direction and no activation of x4 multiplier. Afterwards continue executing the content of memory cell number 121.

4.1.3- FREE RUN WITH RAMP INSTRUCTION

NOTE: This instruction has been kept for retro-compatibility with MIND and HI-MOD series drives. RTA suggests to use the new instruction "HIGH RESOLUTION FREE RUN WITH RAMP".

Movement with the following parameters: direction, frequency, acceleration and deceleration ramp time, x4 multiplier setting. It is different from the indexed run because there is no setting of the number of steps, so a free run ends only with a free run stop command or an emergency stop command. At the receiving of the stop command the drive begins the deceleration ramp.

Code: **CC = 02**

Instruction parameters: **ppppp… = FF,RR,+,XX** where:

- **FF =** maximum frequency code (number between 03 e 24, see Table 2).

- **RR =** ramp code (number between 10 and 42 included, see Table 5).
- **+ =** movement direction.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

  Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

Example

03WN,140,02,10,18,+,x1,210↵

Meaning of the string: in memory cell number 140 of the drive whose address is 03 store a free run with ramp with a maximum frequency of 10 kHz, ramp code 18, positive direction and no activation of x4 multiplier. Afterwards continue executing the content of memory cell number 210.

4.1.4- HIGH RESOLUTION FREE RUN WITH RAMP INSTRUCTION

Movement with the following parameters: direction, frequency, acceleration and deceleration ramp time, x4 multiplier setting. It is different from the indexed run because there is no setting of the number of steps, so a free run ends only with a free run stop command or an emergency stop command. At the receiving of the stop command the drive begins the deceleration ramp.

Code: **CC = 32**

Instruction parameters: **ppppp… = FFF,RR,+,XX** where:

- **FFF =** maximum step frequency in Hz × 100 (number between 030 and 240 included).
- **RR =** ramp code (number between 10 and 42 included, see Table 5).
- **+ =** movement direction.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

  Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

Example

03WN,140,32,113,18,+,x1,210↵

Meaning of the string: in memory cell number 140 of the drive whose address is 03 store an high resolution free run with ramp with a maximum frequency of 11300 Hz, ramp code 18, positive direction and no activation of x4 multiplier. Afterwards continue executing the content of memory cell number 210.

4.1.5- INDEXED RUN WITHOUT RAMP INSTRUCTION

NOTE: This instruction has been kept for retro-compatibility with MIND and HI-MOD series drives. RTA suggests to use the new instruction “HIGH RESOLUTION INDEXED RUN WITHOUT RAMP”.

Movement with the following parameters: number of steps, direction, frequency, x4 multiplier setting. The difference between the *Indexed run without ramp* and the *Indexed run with ramp* is the different values of frequencies that can be selected, lower for the first one. In this case the run does not need an acceleration/deceleration ramp.

Code: **CC = 11**

Instruction parameters: **ppppp… = FF,+nnn..,XX** where:

- **FF =** maximum frequency code (number between 02 e 30, see Table 3).
- **+nnn.. =** integer number between –8388607 e +8388607 indicating the total number of steps. This field can be composed by a varying number of characters.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

  Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

Example

08WN,122,11,02,+1246,x4,188↵

Meaning of the string: in memory cell number 122 of the drive whose address is 08 store an *Indexed run without ramp* with a maximum frequency of 200 Hz, positive direction, 1246 steps and activation of x4 multiplier. Afterwards continue executing the content of memory cell number 188.

4.1.6- HIGH RESOLUTION INDEXED RUN WITHOUT RAMP INSTRUCTION

Movement with the following parameters: number of steps, direction, frequency, x4 multiplier setting. The difference between the *High resolution indexed run without ramp* and the *High resolution indexed run with ramp* is the different values of frequencies that can be selected, lower for the first one. In this case the run does not need an acceleration/deceleration ramp.

Code: **CC = 21**

Instruction parameters: **ppppp… = FFFF,+nnn..,XX** where:

- **FFFF =** step frequency in Hz (number between 0001 and 4800 (see Note).
- **+nnn.. =** integer number between –8388607 e +8388607 indicating the total number of steps. This field can be composed by a varying number of characters.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

    Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

Example

08WN,122,21,0224,+1246,x4,188↵

Meaning of the string: in memory cell number 122 of the drive whose address is 08 store an *High resolution indexed run without ramp* with a maximum frequency of 224 Hz, positive direction, 1246 steps and activation of x4 multiplier. Afterwards continue executing the content of memory cell number 188.

Note: Frequency values set by the user in the string are rounded to the nearest and lower value shown in Table 4 (see Chap. 6.1). Reading again memory cell where instruction is stored (in the example cell number 122) with "CONTENT OF A MEMORY CELL REQUEST" instruction (see Chap. 4.3.5), the real frequency of the drive is shown.

4.1.7- FREE RUN WITHOUT RAMP INSTRUCTION

NOTE: This instruction has been kept for retro-compatibility with MIND and HI-MOD series drives. RTA suggests to use the new instruction "HIGH RESOLUTION FREE RUN WITHOUT RAMP".

Movement with the following parameters: direction, frequency, x4 multiplier setting. It is different from the indexed run because there is no setting of the number of steps, so a free run ends only with a free run stop command or an emergency stop command. The difference between a free run without ramp and a free run with ramp is the different values of frequencies that can be selected, lower for the first one. In this case the run does not need an acceleration/deceleration ramp.

Code: **CC = 12**

Instruction parameters: **ppppp… = FF,+,XX** where:

- **FF =** maximum frequency code (number between 02 e 30, see Table 3).
- **+ =** direction
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

    Note: x4 multiplier enabling is available only for some operation mode settings (see Chap. 4.2.2.3).

Example

02WN,176,12,22,+,x4,133↵

Meaning of the string: in memory cell number 176 of the drive whose address is 02 store a free run without ramp with a maximum frequency of 2200 Hz, positive direction and activation of x4 multiplier. Afterwards continue executing the content of memory cell number 133.

4.1.8- HIGH RESOLUTION FREE RUN WITHOUT RAMP INSTRUCTION

Movement with the following parameters: direction, frequency, x4 multiplier setting. It is different from the High resolution indexed run because there is no setting of the number of steps, so a free run ends only with a free run stop command or an emergency stop command. The difference between a free run without ramp and a free run with ramp is the different values of frequencies that can be selected, lower for the first one. In this case the run does not need an acceleration/deceleration ramp.

Code: **CC = 22**

Instruction parameters: **ppppp… = FFFF,+,XX** where:

- **FFFF =** step frequency in Hz (number between 0001 and 4800, see Note).
- **+ =** direction
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.
    Note: x4 multiplier enabling is available only for some operation mode settings (see Chap. 4.2.2.3).

Example

02WN,176,22,2200,+,x4,133↵

Meaning of the string: in memory cell number 176 of the drive whose address is 02 store a free run without ramp with a maximum frequency of 2200 Hz, positive direction and activation of x4 multiplier. Afterwards continue executing the content of memory cell number 133.

Note: Frequency values set by the user in the string are rounded to the nearest and lower value shown in Table 4 (see Chap. 6.1). Reading again memory cell where instruction is stored (in the example cell number 122) with "CONTENT OF A MEMORY CELL REQUEST" instruction (see Chap. 4.3.5), the real frequency of the drive is shown.

4.1.9- ZERO SEARCH PROCEDURE INSTRUCTION

Free run with ramp with the possibility to stop it in a definite position depending on a proximity sensor connected to the hardware input PX (proximity input).

Code: **CC = 03**

Instruction parameters: **ppppp… = FF,RR,+,LL** where (see Figure 2):

- **FF** = maximum frequency code (number between 03 and 24, see Table 2).
- **RR** = ramp code (number between 10 and 42, see Table 5).
- **+** = direction.
- **LL** = slow approaching frequency code (number between 02 and 30, see Table 3).

Example

11WN,146,03,20,10,+,05,182↵

Meaning of the string: in memory cell number 146 of the drive whose address is 11 store a zero search procedure with a maximum frequency of 20 kHz, ramp code 10, positive direction and slow approaching frequency of 500 Hz. Afterwards continue executing the content of memory cell number 182.

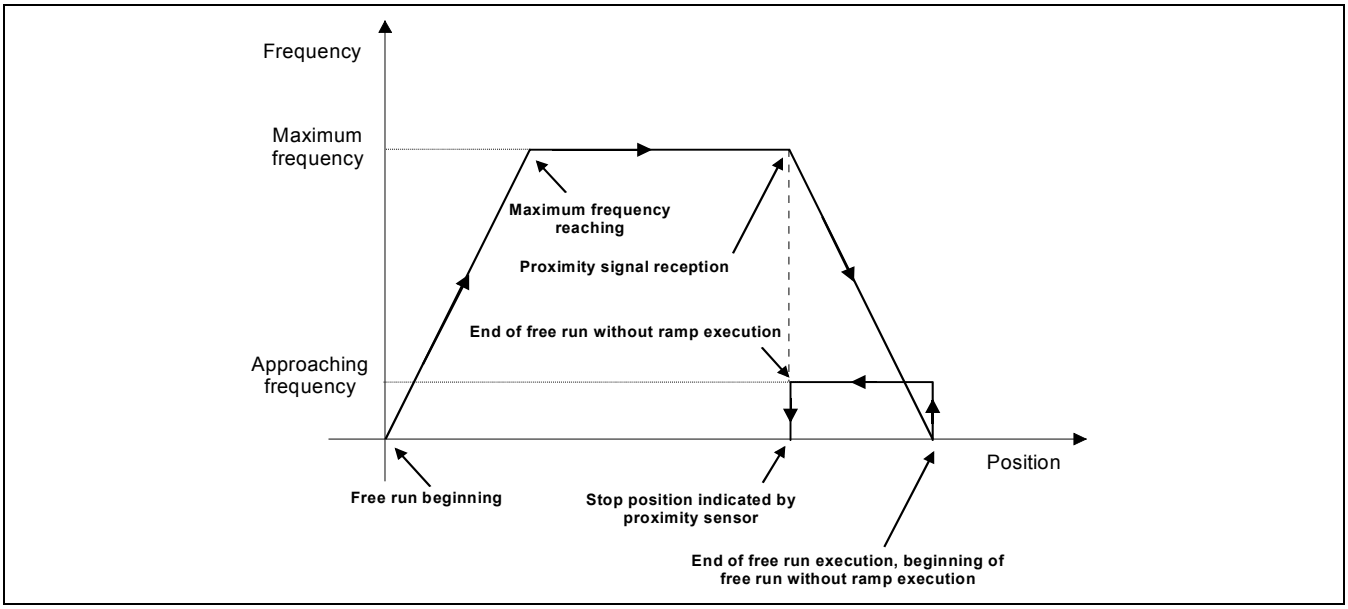


Fig. 2: Motor frequency VS position during a zero search procedure execution.

Figure 2 shows motor frequency VS position during a zero search procedure execution.
Notice that till the proximity sensor signal reception the drive executes a free RUN with ramp.

4.1.10- ELECTRONIC CAM SETTING INSTRUCTION

When setting position is reached (number between -2.147.483.648 and 2.147.483.647) logic value of O3 output is complemented

Code: **CC = 57**

Instruction parameters: **ppppp… = V, +nnn..** where:

- **V =** value to assign:
  - **V=0-**disabled electronic cam
  - **V=1-**enabled electronic cam
- **+nnn.. =** integer number between -2.147.483.648 and 2.147.483.647 indicating the position where O3 output is complemented. This field can be composed by a different number of characters.

Example

06WN,177,57,1,+13000, 191↵

Meaning of the string: in memory cell number 177 of the drive whose address is 06 store an instruction *Electronic cam setting instruction*: When position +13.000 is reached O3 output is complemented. Afterwards continue executing the content of memory cell number 191.

Note 1: When set position is reached (in the example +13.000) electronic cam function is automatically disabled until a new similar instruction is set.

Note 2: To read on O3 output the right logic transition when set position is reached (in the example +13.000), O3 must be set with a suitable value before enabling electronic cam.

Note 3: Electronic cam isn't available with free run in repetitive cyclic operation mode.

Note 4: "ELECTRONIC CAM" function is available from 5000 serial number.

4.1.11- HARDWARE OUTPUT SETTING INSTRUCTION

User can set to logic values 0 or 1 the hardware outputs O0÷O3, the outputs F0÷F7 and enabling/disabling Full Current.

Code: **CC = 58**

Instruction parameters: **ppppp… = OO,V** where:

- **OO =** can be whether O0÷O3, FC, F0÷F7.
- **V =** value to assign; it can be whether 0 or 1.

Example

06WN,170,58,O1,1,194↵

Meaning of the string: in memory cell number 170 of the drive whose address is 06 store an instruction *Hardware output setting*: the output O1 will be set to 1. Afterwards continue executing the content of memory cell number 194.

Note 1:

hardware outputs are:

- FA: driver fault.
- BS: busy.
- O0÷O3: auxiliary output.
- FC: enabling/disabling Full Current.
- F0÷F7 dummy outputs (flags).

These outputs can be set to logic value whether 0 or 1. The outputs O0÷O3 and F0÷F7 can be set by the user. The outputs FA and BS depend on internal signals of the drive and can not be modified with *Hardware output setting* instruction.

Note 2:

FC = 1 Always keep the drive in Full Current

FC = 0 enabling automatic current reduction

When automatic current reduction is active, the current which flows in each motor windings is reduced to 50% of nominal current, 50 ms after the end of run.

Except uncommon cases we recommend to set active the automatic current reduction to avoid unnecessary overheating.

4.1.12- TIME DELAY INSTRUCTION

Time delay of a duration selected by the user. Maximum allowed time delay is 65535 milliseconds.

Code: **CC = 65**

Instruction parameters: **ppppp… = nnn..** where:

- **nnn.. =** number between 0 and 65535 corresponding to the time delay in milliseconds.

Example

12WN,120,65,4000,192↵

Meaning of the string: in memory cell number 120 of the drive whose address is 12 store an instruction *Time delay* with a waiting time of 4 seconds: after 4 seconds the sequence of instruction continues with the instruction stored in memory cell number 192.

Note

The presence of a time delay between the end of a movement and the beginning of another one is strictly required in order to control mechanical problems of system assessment (settling time).

4.1.13- NULL INSTRUCTION

It has no effects.

Code: **CC = 00**

Instruction parameters: **none**

Example

15WN,135,00,196↵

Meaning of the string: in memory cell number 135 of the drive whose address is 15 store a *Null* instruction. The sequence of instruction continues with the instruction stored in memory cell number 196.

Note

Null instruction can be used to do a reset of a memory cell, getting the certainty of its content. For example:

00WN,130,00,000 reset of the content of memory cell 130

Another application is the easy variation of via hardware instruction selection, that is if we write:

00WN,100,00,116

00WN,116,.....

We obtain that as a consequence of the selection of the hardware inputs corresponding to memory cell number 100, the program passes to the instruction in memory cell number 116. The difference between this situation and the writing of the content of 116 directly in 100 is the easier change of the instruction pointed by the hardware input corresponding to 100: we would only have to change the next address of memory cell number 100. That could be useful in the presence of different sequences of instructions.

4.1.14- CONDITIONAL JUMP INSTRUCTION

Depending on a hardware input status, the sequence of instructions continues to next address 1 if the condition is true or to next address 2 if the condition is false.

Code: **CC = 63**

Instruction parameters: **ppppp… = II,V,EEE** where:

- **II =** hardware input whose value determines the condition (I0÷I2, PX, S0÷S3, F0÷F7, where S0÷S3 correspond to selection inputs on C2 connector).
- **V =** input status (0 or 1).
- **EEE =** memory cell address which will be considered the next address if the condition on the hardware input is true; if the condition is false the sequence of instructions continues with the instruction stored in DDD.

Example

00WN,180,63,I1,0,130,155↵

Meaning of the string: in memory cell number 180 of the drive whose address is 00 store an instruction *Conditional jump*: if I1 = 0 continue to memory cell 130, if I1 = 1 continue to memory cell 155.

Note

Using the instruction *Conditional jump* the user can obtain loops whose end condition is the change of an hardware input logic value.

4.1.15- INTERRUPTED PROGRAM RECOVERY INSTRUCTION

Recovery of a previously interrupted instruction. The instruction execution is ended, beginning from the point where it has been stopped.

Code: **CC = 69**

Instruction parameters: **none**

Example

14WN,220,69↵

Meaning of the string: in memory cell number 220 of the drive whose address is 14 store an instruction *Interrupted program recovery*.

Note

It's possible to stop any instruction using the emergency stop command (EE code). The information related to the interrupted action are stored in a RAM memory, so the user has the possibility to recall it and to complete the execution of the interrupted instruction.

The *Interrupted program recovery* instruction works only on the last instruction which has been stopped: if many instructions have been stopped, only the last one will be ended. If nothing is stored in the RAM memory, the *Recovery instruction* ends the program and the drive executes a null action with next address 000.

4.1.16- DELAYED STOP INSTRUCTION

Free run with ramp with a definite braking distance after the reception of the signal of a STOP sensor (connected to ES pin).

Code: **CC = 05**

Instruction parameters: **ppppp… = FF,RR,+,nnn..,XX** where:

- **FF =** maximum frequency code (number between 03 and 24, see Table 2)
- **RR =** ramp code (number between 10 and 42, see Table 5).
- **+ =** movement direction.
- **nnn..** = integer number lower than 8.388.608 corresponding to the number of steps covered between the reception of the free run STOP signal and the stopping of the motor. This field can be composed by a variable number of characters.
- **XX :** two different expressions are possible:
  - **XX = x1 –** Steps and frequencies multiplier by one.
  - **XX = x4 –** Steps and frequencies multiplier by four.

  Note: x4 multiplier enabling is available only for some operation mode settings (see chap. 4.2.2.3).

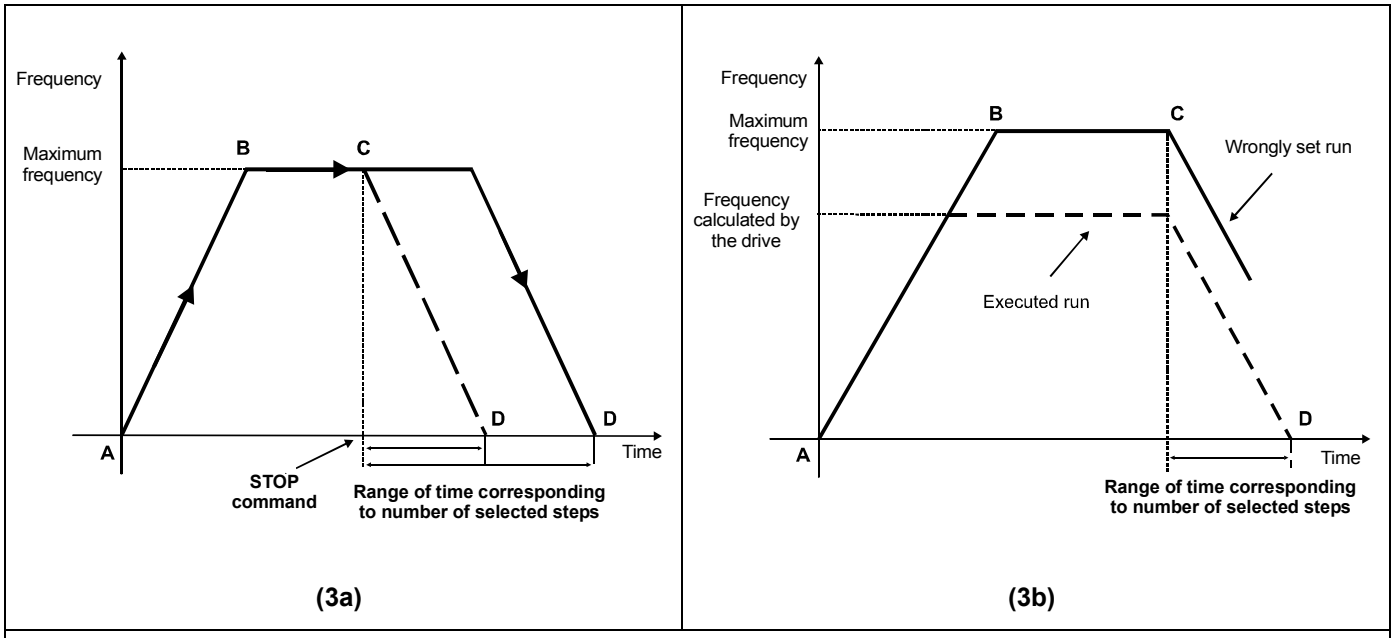
Example

01WN,165,05,10,20,+,3000,x1,205↵

Meaning of the string: in memory cell number 165 of the drive whose address is 01 store an instruction *Delayed stop* with a braking distance of 3000 steps, a maximum frequency of 10 kHz, code ramp 20, positive direction without x4 multiplier. The sequence of instruction continues with the instruction stored in memory cell number 205.

Note

1) The braking distance (number of steps covered by the motor between the reception of the free run STOP signal and the stop of the movement) is defined as an instruction parameter, as above explained. The parameters defining the deceleration ramp are: maximum frequency, number of steps and ramp time. Varying this three parameters, there are three possible cases:
   - The borderline case is when the number of steps is the same of the one required by the deceleration ramp (hatched line in figure 3a).
   - The case described by the continuous line in figure 3a corresponds to a condition where the deceleration ramp requires a number of steps lower than the one selected. In this case the motor continues the running at the same frequency until it begins the deceleration ramp in such a way that the sum of the number of steps required by the deceleration ramp and the number of steps covered at constant speed is the selected number of steps; this is the condition of correct system working.
   - When the selected number of steps is lower than the one required by the deceleration ramp (continuous line in figure 3b), speed is automatically decreased to such a value that the deceleration ramp requires the selected number of steps. So the motor behaviour is the one described in figure 3b with the hatched line. Depending on the selected number of steps, the new frequency calculated by the drive in the case above described can be lower than the minimum frequency of the free run with ramp (3 kHz).



Meaning of symbols: **A** = Beginning of free run; **B** = Reaching of maximum frequency; **C** = Reception of stop signal; **D** = Stop of motor movement.

Fig. 3: Motor frequency VS time during the execution of an instruction *Delayed stop* for different values of the number of steps, the frequency and the ramp time.

2) If the selected number of steps is 0 (setting not coherent with the requirements of a free run with ramp), the motor covers one step between the reception of the STOP signal and the actual stop. Obviously in this case the drive has to calculate a new frequency as explained in note 1.
3) For the instruction *Delayed stop,* the instruction *Recovery of an interrupted program* is not available. So if the last interrupted instruction has been a *Delayed stop*, it is not possible, differently from all the other instructions, to recover it.
4) Free run STOP is ignored during the execution of the acceleration ramp. The user has to check the number of steps required by the acceleration ramp in order to know if the STOP can be accepted or not.

4.1.17- CURRENT ON / CURRENT OFF INSTRUCTION

Enable or disable of motor current.
Code: **CC = 70**
Parameters: **ppppp…= S** where:

- **S = 1 or 0.**
  - S = 1 CURRENT ON
  - S = 0 CURRENT OFF

EXAMPLE

03WN,180,70,1,190↵

Meaning of the string: in the memory cell number 180 of the drive, whose address is 03, store a *Current ON* instruction. Then continue executing the content of the memory cell number 190.

4.1.18- OPENING LOOP INSTRUCTION

Execution of a group of instructions repeated a fixed number of times.

Code: **CC = 61**

Parameters: **ppppp… = nnnn** where

- **nnnn =** repetitions number: integer number without sign included between1 and 65.535.

Example

03WN,180,61,12,181↵

Meaning of the string: in the memory cell number 180 of the drive , whose address is 03 store an OPENING LOOP instruction and load counter to 12 repetitions. Then continue executing the content of the memory cell number 181.

Note:

Opening Loop instruction has to be used always together with *Closing Loop* instruction (see next paragraph). The group of instructions, that is all the instructions that have to be repeated, has to have the first instruction stored in the memory cell pointed by *Opening Loop* instruction (in the previous example, the first instruction of the group of instructions has to be stored in memory cell number 181).

4.1.19- CLOSING LOOP INSTRUCTION

Execution of a group of instructions repeated a fixed number of times.

Code = **CC = 62**

Parameters = **ppppp… = EEE** where

- **EEE =** Memory address of the first instruction of the block of instructions. When a *Closing Loop* instruction is executed, the counter, loaded by the *Opening Loop* instruction, is decreased of one unit. If counter = 0 then memory cell indicated by DDD is executed, if counter ≠ 0 then memory cell indicated by EEE is executed and a new repetition of the block begins (see 4.1).

Example

03WN,183,62,181,201↵

Meaning of the string: in the memory cell number 183 of the drive, whose address is 03 store a *Closing Loop* instruction. Decrease the repetitions counter and if the counter is different from zero, continue executing the content of the memory cell number 181 (group of instructions) while if the counter is zero, execute memory cell number 201.

Note:

Closing Loop instruction points to two different instructions, and the execution of one or the other depends on the value of the counter when *Closing Loop* instruction execution begins. To obtain a correct loop structure, the user have to define the total number of instructions of the block and their addresses. Moreover, the instructions of the block have to be linked together, so that the end of an instruction executes automatically the following instruction. The last instruction of the block has to point to the *Closing Loop* instruction, so that the latter, depending on the value of the counter, points either to the first instruction of the block, or to the instruction to execute after the end of the block.

It is not possible the writing of a loop in another loop.

Examples of Loop

1) Suppose to execute for five times a group of four instructions constituted by two runs *Indexed run with ramp* and two *Time Delay* instructions (respectively instructions with code 01 and code 65). The programming of memory cells could have the following structure:

```
06WN,120,61,5,121↵          Opening Loop
06WN,121,01,12,30,+12500,x1,122↵
06WN,122,65,400,123↵        Block of instruction
06WN,123,01,12,30,-12500,x1,124↵
06WN,124,65,400,125↵
06WN,125,62,121,000↵        Closing Loop
```

Note that the block of instruction, that is constituted by instructions in memory cells 121, 122, 123, and 124, the first time is executed after the execution of the *Opening Loop* instruction. After the block of instruction execution, the instruction contained in cell number 125 (*Closing Loop*) is executed, so the counter is decreased of one unit and checked; if counter ≠ 0, memory cell number 121 is executed, otherwise if counter = 0 the second memory cell pointed by *Closing Loop* instruction is executed. In the example, program ends because the address is 000.

2) Suppose to enable and disable the output O0 for 10 times running (10 flashings of a led if this is connected to output O0) so that the ignition frequency is 1 Hz (so for 500 msec the output O0 has to be 0 and for the same time it has to be 1). At the end of the loop is executed a *Indexed Run Without Ramp* instruction.

```
09WN,100,61,10,121↵         Opening loop
09WN,121,58,O0,1,122↵
09WN,122,65,500,123↵        Block of instruction
09WN,123,58,O0,0,124↵
09WN,124,65,500,125↵
09WN,125,62,121,130↵        Closing Loop
09WN,130,11,05,+6000,x1,000↵  INDEXED RUN WITHOUT RAMP instruction
```

4.1.20- ABSOLUTE POSITION COUNTER RESET INSTRUCTION

Reset of the internal steps counter.

Code: **CC = 60**

Parameters: **none**

Example

```
03WN,180,60,190↵
```

Meaning of the string: in the memory cell number 180 of the drive whose drive address is 03, store an *Absolute Position Counter Reset* instruction. Then continue executing the content of memory cell number 190.

4.2- COMMANDS

A command is an action (emergency stop, activation of motor current…) which is not stored in the drive but immediately executed at the reception of it by the drive. Some commands can be activated through proper hardware inputs (see chapter 2). In addition to these ones, other commands can be transmitted through the serial line.
They can modify some actions of the drive (see par. 4.2.2).
The only commands recognized by the drive even if it is executing an action (instruction, command…) are:

- **EE** = Emergency stop.
- **ES** = Free run stop.

All the others can be recognized only when the drive is free. At the end of the reception of a command through the serial line, the drive transmits a confirmation message whose syntax is described in par. 4.4. Remember that, in case of broadcast commands, the drive doesn't transmit any confirmation message.

4.2.1- COMMANDS THROUGH SERIAL LINE

Common syntax of all commands is the following:

AACC,pp.. ↵

where:

- **AA** = address (number between 00 and 47 or 99 for broadcast transmission).
- **CC** = command code.
- **pp..** = possible command parameters.

Available commands are the following:

4.2.2- PRESET COMMAND

This command causes the variation of a preset parameter. Broadcast transmission is allowed for this command using 99 as drive address.

Code **CC = WS**
Parameters: **pp.. = VV,nnn,kkk**

4.2.2.1 ADDRESS Preset

VV = AD
nnn = address (number between 0 and 47).
kkk = it isn't necessary.

Default setting: nnn=0

Example: 03WS,AD,38↵ Changing drive address from 03 to 38.
This configuration is stored in Eeprom.

Note: The drive addresses range changing is immediate at the proper command transmission, but the corresponding confirmation message transmitted by the drive contains the drive address of the string of the transmitted preset command.

4.2.2.2 DRIVE CURRENT Preset

VV = IN
nnn = Following values are accepted:

- 0: corresponding to current setting as 55% of nominal current
- 1: corresponding to current setting as 70% of nominal current
- 2: corresponding to current setting as 85% of nominal current
- 3: corresponding to current setting as 100% of nominal current

kkk = it isn't necessary.

Default setting: nnn=1

Example: **13WS,IN,3↵** Current in drive 13 is set to 100%
This configuration is stored in Eeprom.

4.2.2.3 OPERATION MODE Preset

VV = RS
nnn = Following values are accepted (D: decimal; B: binary):

- D0: corresponding to setting of motor 500 step/rev
- D1: corresponding to setting of motor 1000 step/rev
- D2: corresponding to setting of motor 2000 step/rev
- D3: corresponding to setting of motor 4000 step/rev

- B0: corresponding to setting of motor 400 step/rev
- B1: corresponding to setting of motor 800 step/rev
- B2: corresponding to setting of motor 1600 step/rev
- B3: corresponding to setting of motor 3200 step/rev

kkk = it isn't necessary.

Default setting: nnn=D2

Example: **13WS,RS,B2↵** Set resolution: 1600 motor step/rev
This configuration is stored in Eeprom.

Note 1: Step × 4 function can be enabled in D2, D3, B2 and B3 operation mode.
Note 2: following this command "Operation mode preset" the drive is set in "Current Off for at least 200 ms and in this time it isn't able to receive any instructions. This command mustn't be set during the usual operation mode of the drive because could cause position and holding torque loss of the motor.

4.2.2.4 EQUALIZATION Preset

VV = EQ
nnn = Following values are accepted:
- 0: equalization excluded
- 1: equalization active

kkk = it isn't necessary.

Default setting: nnn=0

Example: **23WS,EQ,0↵** Equalization excluded on number 23 drive
This configuration is stored in Eeprom.

Note: When equalization is active, the motor has more torque at medium speed. We recommend to keep equalization excluded in case of movements with short run repeated to high rhythm.

4.2.2.5 ANSWER DELAY DRIVE Preset

This parameter corresponds to a time delay of the answer on serial communication (RS485) that drive transmits to control system when requested.

VV = RD
nnn = can be an integer number between 0 and 255, as the delay of the answer in msec
kkk = it isn't necessary.

Default setting: nnn=10

Example: **03WS,RD,98↵** Answer delay preset parameter = 98 msec.
This configuration is stored in Eeprom.

Note: The increasing of the delay on the serial communication, delays the execution of all the commands transmitted through serial line. For example, an emergency stop transmitted through the serial line is executed only after a time corresponding to the delay set by **RD**.

4.2.2.6 COORDINATES OPERATION MODE Preset

On the basis of the coordinates operation mode used correspond different behaviours of *Indexed run with and without ramp* (standard and high resolution) instructions. In chapter 5 all the operation modes are described.

VV = CM then **nnn** can be 0, 1 or 2:

- **nnn = 0**: relative coordinates operation mode; in this case **kkk** isn't necessary
- **nnn = 1:** absolute coordinates operation mode; in this case **kkk** isn't necessary
- **nnn = 2**: repetitive cyclic operation mode. In this case you have to specify the **kkk** parameter corresponding to the working range. The valid values for **kkk** are included between 1 and 8.388.607.

Default setting: nnn=0

Example: **03WS,CM,0↵** Coordinates mode preset parameter = relative coordinates setting
This configuration is stored in Eeprom.

4.2.2.7 ES PRIORITY Preset

This command allows to know the behaviour of the drive after the execution of a free RUN stopped with ES, when ES remains set (ES = 1)after the motor has finished its revolution.

VV = ES
nnn = following values are accepted:

- 0: if input ES=1, START HARDWARE commands on input terminal **16 (ST)** aren't accepted (retro-compatible with MIND and HI-MOD series drives); the motor is stopped until ES signal is set to zero.
- 1: even if ES=1, START HARDWARE commands on input terminal **16 (ST)** are accepted.

kkk = it isn't necessary.

Default setting: nnn=1

Example: **23WS,ES,1↵** New START HARDWARE command is accepted also with ES=1.
This configuration is stored in Eeprom.

4.2.3- CURRENT ON

Enable of motor current. Broadcast transmission is allowed for this command using 99 as drive address.

Code **CC = CY**
Parameters: **pp.. = none**
Example: **12CY↵**

Note: after a CURRENT ON command execution, the presence of current in the motor will remain until whether a CURRENT OFF command or the drive switching off, or until the execution of *CURRENT ON / CURRENT OFF* instruction (code 70). At the drive switching on, drive default condition is CURRENT ON.

4.2.4- CURRENT OFF

Disable of motor current. Broadcast transmission is allowed for this command using 99 as drive address.

Code **CC = CN**
Parameters: **pp.. = none**
Example: **08CN↵**

4.2.5- STORED INSTRUCTION EXECUTION

Execution of a previously stored instruction. Broadcast transmission is allowed for this command using 99 as drive address.

Code **CC = PS**
Parameters: **pp.. = BBB**
BBB = memory cell address where it is stored the instruction to execute.
Example: **05PS,147↵**

4.2.6- EMERGENCY STOP

Stop of the execution of any instruction or command, immediate block of any program running. Broadcast transmission is allowed for this command using 99 as drive address.

Code **CC = EE**
Parameters: **pp.. = none**
Example: **05EE↵**

4.2.7- FREE RUN STOP

Stop of the execution of a free run instruction (both with and without ramp). Broadcast transmission is allowed for this command using 99 as drive address.

Code **CC = ES**
Parameters: **pp.. = none**
Example: **02ES↵**

4.2.8- ABSOLUTE POSITION COUNTER RESET

Reset of internal steps counter. Broadcast transmission is not allowed for this command, so in this paragraph drive address can not be 99.

Code **CC = RA**
Parameters: **pp.. = none**
Example: **08RA↵**

4.2.9- ABSOLUTE POSITION COUNTER SETTING

Loading a value between -2.147.483.648 and +2.147.483.647 into internal absolute position counter. Broadcast transmission is not allowed for this command, so in this paragraph drive address can not be 99.

Code **CC = SA**
Parameters: **pp.. = none**
Example: **03SA,3200↵**

4.2.10- HARDWARE OUTPUT SETTING

Setting of an hardware output logic status. Broadcast transmission is not allowed for this command, so in this paragraph drive address can not be 99.

Code **CC = SO**

Parameters: **pp.. = OO,V**

OO can be whether

-O0÷O3: pin of auxiliary output

-FC: enabling/disabling Full Current.

-F0÷F7: one of the eight dummy outputs (flag), (see chap. 2.2.2.4)

V can be whether 0 or 1.

Example: **14SO,O1,1↵**

Note:

When FC = 1 X-MIND K is always in Full Current (at stand still motor too).

FC = 0 automatic current reduction is enabled.

When automatic current reduction is active, the current which flows in each motor windings is reduced to 50% of nominal current, 50 ms after the end of run.

Except uncommon cases we recommend to set active the automatic current reduction to avoid unnecessary overheating.

4.2.11- ELECTRONIC CAM SETTING

When setting position is reached (number between -2.147.483.648 and 2.147.483.647) logic value of O3 output is complemented. Broadcast transmission is not allowed for this command, so in this paragraph drive address can not be 99.

Code: **CC = SC**

Parameters: **ppppp… = V, +nnn..** where:

- **V =** value to assign:
  - **V=0-**disabled electronic cam
  - **V=1-**enabled electronic cam
- **+nnn.. =** integer number between -2.147.483.648 and 2.147.483.647 indicating the position where O3 output is complemented. This field can be composed by a different number of characters.

  Example: **06SC,1,+117500↵**

Note 1: When set position is reached (in the example +117.500) electronic cam function is automatically disabled until a new similar instruction is set.

Note 2: To read on O3 output the right logic transition when set position is reached (in the example +117.500), O3 must be set with a suitable value before enabling electronic cam.

Note 3: Electronic cam isn't available with free run in repetitive cyclic operation mode.

Note 4: "ELECTRONIC CAM" function is available from 5000 serial number.

4.3- DATA REQUESTS AND ANSWERS

Control system can transmit data requests to drive through the serial line in order to get information about drive status. Drive answer depends on the particular data request, as showed in following examples. Anyway the answer always begins with drive address followed by data request code. For all data requests broadcast transmission is not allowed.
The common syntax of all the instructions is the following:

AACC,pp.. ↵

where:

- **AA** = drive address (number between 00 and 47).
- **CC** = data request code.
- **pp.. =** data request parameters (if necessary).

Data requests are:

4.3.1- ABSOLUTE POSITION REQUEST

Request of steps counter content to get the total number of covered steps from the last power on or counter reset command. Steps in negative direction are subtracted to steps in positive direction. Drive answer, as indicated in the example, contains a number with sign. Broadcast transmission is not allowed.

Code **CC = QA**
Command parameters: **pp.. = none**
Example of data request: **14QA↵**
Example of answer: **14QA,+5121↵**

Note.: counting limits are:

- higher limit = + 2.147.483.647
- lower limit = - 2.147.483.648

4.3.2- PRESET VALUE REQUEST

Request of a preset parameter value. Broadcast transmission is not allowed.
Code **CC = QS**
Parameters **pp.. = VV**
VV can be:

- AD: drive address
- IN: set current
- RS: operation mode (resolution)
- EQ: equalization selection
- ES: STOP RUN priority
- RD: time delay on serial communication answer
- CM: coordinate system

Data request example:	**14QS,IN↵**	**14QS,RS↵**
Answer example:	**14QS,IN,0↵**	**14QS,RS,B1↵**
Data request example:	**14QS,RD↵**	**14QS,ES↵**
Answer example:	**14QS,RD,102↵**	**14QS,ES,1↵**
Data request example:	**14QS,CM↵**	**14QS,EQ,↵**
Answer data request:	**14QS,CM,0↵**	**14QS,EQ,1↵**

4.3.3- HARDWARE INPUT STATUS REQUEST

Request of a hardware input status. Broadcast transmission is not allowed.

Code **CC = QI**

Command parameters: **pp.. = IN**

IN can be:

- ST : start hardware
- ES : free run stop
- EE : emergency stop
- I0÷I2 : auxiliary input pin
- PX : proximity input
- S0÷S3: one of the four selection inputs

Example of data request: **14QI,I0↵**

Example of answer: **14QI,I0,1↵** (supposing I0 = 1)

4.3.4- HARDWARE OUTPUT STATUS REQUEST

Request of a hardware output status. Broadcast transmission is not allowed.

code **CC = QO**

Command parameters: **pp.. = OO**

OO can be:

- FA: driver fault
- BS: busy
- O0÷O3: auxiliary output pin
- FC: enabling/disabling Full Current.
- F0÷F7one of the eight dummy outputs (flag) (see chap. 2.2.4)

Example of data request: **14QO,O1↵**

Example of answer: **14QO,O1,0↵** (supposing O1 = 0)

Note: Supposing BS=1, at the output BS status request, the drive doesn't transmit any answer: BS=1 means that drive is busy in a command execution.

4.3.5- CONTENT OF A MEMORY CELL REQUEST

Request of the content of a memory cell whose address is specified as a parameter. Broadcast transmission is not allowed.

code **CC = QM**

Command parameters: **pp.. = BBB**

BBB = memory cell address whose content is asked.

Example of data request: **11QM,172↵**

Example of answer N°1: **11QM,172,01,09,15,+31754,X1,121↵**

Example of answer N°2: **11QM,172,65,3400,159↵**

Note: the *example of answer N°1* shows the case in which in memory cell number 172 is located an instruction *Indexed run with ramp*; in *example of answer N°2* we have an instruction *Time delay*. Notice that the answer structure is the same of the instruction stored in the memory cell.

4.3.6- ELECTRONIC CAM SETTING REQUEST

Request of the electronic cam setting. Broadcast transmission is not allowed.

code **CC = QC**

Command parameters: **pp.. = nothing**

BBB = memory cell address whose content is asked.

Example of data request: **14QC↵**

Example of answer N°1: **14QC,1,+56564↵**

Example of answer N°2: **14QC,0↵**

Note 1: "ELECTRONIC CAM" function is available from 5000 serial number.

4.3.7- KIND OF ERROR REQUEST

Request of the kind of the error and the relative address of the memory cell whose execution generated the error. Broadcast transmission is not allowed. The error can be generated trying the execution of a stored instruction that can not be executed. In the answer the drive transmits to the system the error code (00=no error, 50=FIELD overflow error) followed by the address of memory cell (000 if code error is 00).

Code **CC = QE**
Parameters: **pp.. = none**

Example of data request: **11QE↵**
Example of answer N°1: **11QE,00,000↵***:*
Example of answer N°2: **11QE,50,212↵**

4.4- CONFIRMATION MESSAGES

The drive answers to commands or instructions transmitting a short message immediately afterwards having received a command or an instruction giving the possibility to the control system to check the correct command or instruction reception (see also chapter 3). Possible confirmation messages are described in 4.4.1, 4.4.2 and 4.4.3. In the case of a broadcast transmission, the drive transmits no confirmation message. So in this paragraph we never refer to a drive address as 99.

4.4.1- AAY↵

(AA = drive address). The reception of this message means that:

- The command (or instruction) has been recognized by the drive whose drive address is AA as addressed to it.
- The command (or instruction) is written using the correct syntax.
- Parameters were in the correct range.
- The command (or instruction) has been transmitted in a moment in which this operation was allowed (any moment for EE and ES commands, free drive for the others).
- Command (or instruction) has been executed. Obviously a transmitted command can have no effect, for instance a STOP command with motor already stopped, a CURRENT OFF command with current already zero, etc…

4.4.2- AAN↵

(AA = drive address). The receiving of this message means that:

- The command (or instruction) has been recognized by the drive whose drive address is AA as addressed to it.
- There was a syntax error or some parameter was out of range.
- The command (or instruction) has been transmitted in a moment in which this operation was allowed (any moment for EE and ES commands, free drive for the others).
- The command (or instruction) has not been executed.

4.4.3- NO CONFIRMATION MESSAGES

No answer from the drive is due to one of the following causes:

- The drive is off or the serial line is not connected.
- Drive address not correct or no connected drives have the drive address written in the command (or instruction).
- The command (or instruction) has been transmitted in a moment in which this operation was not allowed (drive working for all the commands or instructions but EE or ES commands).
- Drive address refers to broadcast communication.

5- OPERATION MODE

In this chapter are described the three operation modes that can be set through WS command and the different behaviours of X-MIND K series drives according to the setting. The different behaviours refers to *Indexed run with ramp* and *Indexed run without ramp* instructions (see par. 4.1.1, 4.1.2, 4.1.5, 4.1.6), where the interpretation of the number of steps depends on the operation mode.

5.1- RELATIVE COORDINATES MODE

- Relative coordinates operation mode is set through WS command, whose syntax (see par. 4.2.2.6) is the following: **xxWS,CM,0**↵ (**xx** = drive address).
- The range of the position absolute counter (see par. 4.3.1, 4.2.8, 4.2.9, 4.1.20) is between -2.147.483.648 and +2.147.483.647 steps.
- The maximum number of steps that can be executed with one Index instruction is ± 8.388.607 steps.
- During the execution, an *Indexed run* (both with and without ramp) interprets the number of steps like a relative shift starting from current motor position.
- If the multiplier X4 is enabled, the maximum frequency, acceleration and number of the steps are multiplied by 4, if X4 multiplier enabling is available according to the resolution set (4000, 3200, 2000, 1600 steps/rev.), (see instruction manual).

EXAMPLE: Executing an *Indexed run with ramp* previously stored using the command 00WN,100,01,20,30,+2400,X1,000↵, motor covers 2400 steps in the positive direction, independently on the starting position of the motor. The position absolute counter is increased of 2400 steps.

5.2- ABSOLUTE COORDINATES MODE

- Absolute coordinates operation mode is set through WS command, whose syntax (see par. 4.2.2.6) is the following: **xxWS,CM,1**↵ (**xx** = drive address).
- The range of the position absolute counter (see par. 4.3.1, 4.2.8, 4.2.9, 4.1.20) is between -2.147.483.648 and +2.147.483.647 steps.
- During the execution, an *Indexed run* (both with and without ramp) interprets the number of steps like an absolute position to reach; motor covers a number of steps depending on the starting position, that can be checked reading the position absolute counter.
- The addressing range in absolute coordinates mode is between - 8.388.607 and + 8.388.607 steps.
- The maximum number of steps that can be executed with one Index instruction is ± 8.388.607 steps. If a relative shift out of range is set, the execution of the instruction is not allowed and a FIELD overflow error is raised (error code 50). Error analysis can be done through the proper command (par 4.3.7).
- Working with absolute coordinates operation mode requires the definition of an origin, corresponding to coordinate 0. Usually coordinate 0 can be defined through the execution of a zero search procedure (see par. 4.1.9) followed by the internal steps counter reset (see par. 4.1.20 and 4.2.8).
- If the step multiplier is enabled and with a right setting of operation mode (4000, 3200, 2000, 1600 step/rev). only maximum frequency and acceleration are multiplied by four. In this case the position reached is the one set as instruction parameter, with a maximum error range of ± 3 steps. That's shown by the value of the internal position absolute counter, which corresponds to the exact position of the motor. Maximum position error is always ± 3 steps, even when more than one run is executed with X4 set.

EXAMPLE: Executing an *Indexed run with ramp* previously stored using the command 00WN,100,01,20,30,+2400,X1,000↵, the motor shift is in positive direction if the motor is in a position lower then +2400 steps, or in negative direction, if the position before starting is higher then +2400. The borderline case is when the starting position is + 2400, then motor doesn't move. If the absolute starting position is -8.388.607, the motor shift is not allowed, given that number of steps that should be executed are 8.391.007 and they over the maximum number of steps that can be executed with an Index instruction only (± 8.388.607 steps); the instruction is not allowed and a FIELD overflow error is raised (error code 50).

5.3- REPETITIVE CYCLIC OPERATION MODE

- The repetitive cyclic operation mode can be set through the command WS (see par. 4.2.2.6), whose syntax is **xxWS,CM,2,kkk↵** (**xx** = drive address). **kkk** is an integer number without sign between 1 and 8.388.607 and corresponds to working range of the repetitive cyclic operation mode.
- The range of the absolute counter (see par. 4.3.1, 4.2.8, 4.2.9, 4.1.20) is between 1 and **kkk**.
- During the execution, an *Indexed run* (both with and without ramp) interprets the number of steps like an absolute position to reach and the total movement of the motor is circular (see Fig. 4), without mechanical stops. Motor covers a number of steps depending on the starting position, that can be checked through the internal position absolute counter.
- When the multiplier by 4 is enabled a maximum error of ± 3 steps can affect the position of the motor. Maximum position error is always ± 3 steps, even when more than one run is executed with X4 set.
- In repetitive cyclic operation mode a FIELD overflow error can occur when an *Indexed run* instruction (both with or without ramp) contains a negative coordinate or a number of steps higher then the working range. The analysis of the FIELD overflow error, can be carried out through the proper serial command (par. 4.3.7).

EXAMPLE: The typical application of repetitive cyclic drive operation mode is a tool changer, as the one described in Fig. 4.

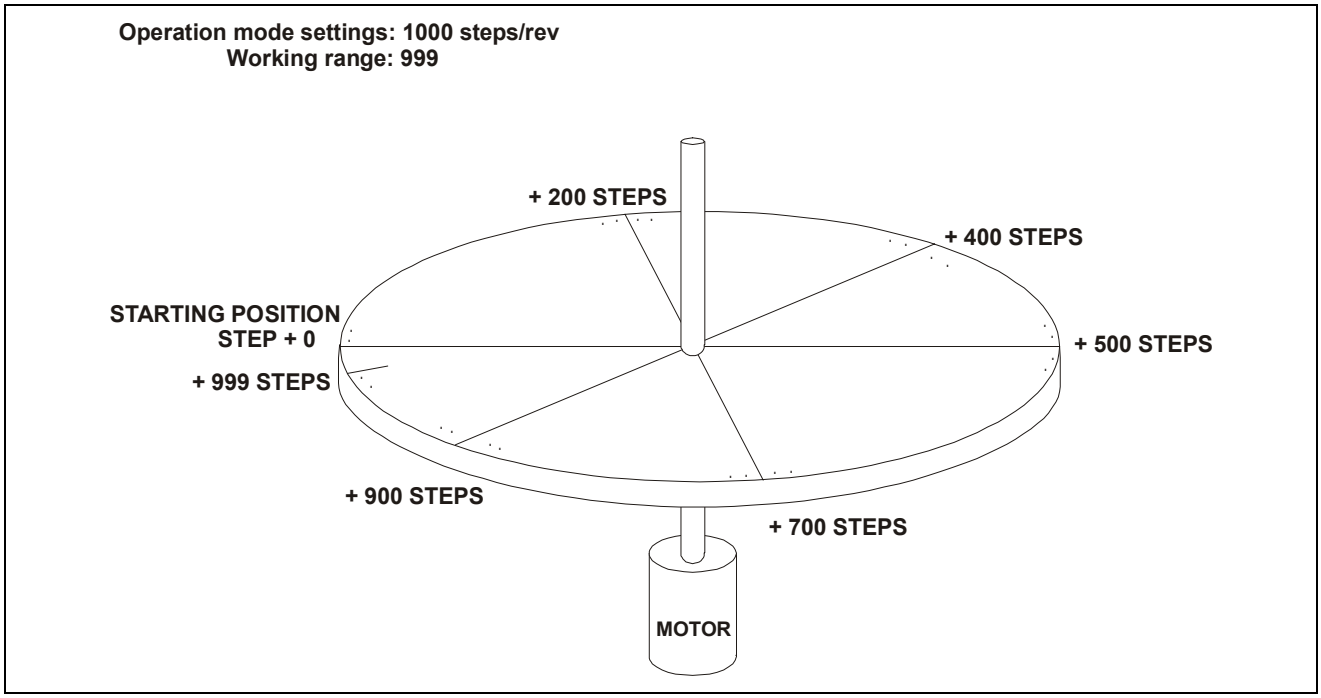


Fig. 4: Tool changer and working range

Suppose to have a system as the one described in the fig. 4: a stepping motor direct drive to a tool changer with a 1.000 steps for revolution setting. The total number of steps covered to return in the starting position is 999 from the starting position. Position next to +999 is +0. All the positions in this range can be reached referring to an absolute coordinates between +0 and +999. In the example showed in fig. 4, all the positions in the range +0 and + 999 can be reached using an *Indexed run (both with and without ramp)* with a number of steps between 0 and 999. Maximum range is 8.388.607, minimum is 1; even if low working range are hard to use.
In the example of fig. 4, six different positions have been defined: +0, +200, +400, +500, +700, +900.

Each of these correspond to indexed run instructions stored in different memory cells. For example: a valid storing of the indexed run instructions can be the following:

00WN,100,11,05,+0,x1,000↵
00WN,101,11,05,+200,x1,000↵
00WN,102,11,05,+400,x1,000↵
00WN,103,11,05,+500,x1,000↵
00WN,104,11,05,+700,x1,000↵
00WN,105,11,05,+900,x1,000↵

Every run agree with one of the six positions defined on the tool changer of fig. 4.
To reach a selected position, drive covers the lowest number of steps, that is the direction of the movement is chosen by the drive in order to obtain the shortest path. The steps counter shows the absolute position in the working range.
As in the case of absolute coordinates operation mode, the absolute counter will be changed only if internal motor is enabled. Otherwise a relative run will be executed on the external motor.

6- FREQUENCY AND RAMP TIME CODE TABLES

In this paragraph the tables of frequency codes and ramp time codes used in some instructions are listed. The instructions where codes appear and the correspondences between codes and quantities are showed in the tables.

6.1- SPEED CODES (step frequencies)

Table 2

Frequency codes for:
INDEXED RUN WITH RAMP
FREE RUN WITH RAMP
High speed in **ZERO SEARCH PROCEDURE**

Code	Frequency (Hz)	Code	Frequency (Hz)
03	3.000	14	14.000
04	4.000	15	15.000
05	5.000	16	16.000
06	6.000	17	17.000
07	7.000	18	18.000
08	8.000	19	19.000
09	9.000	20	20.000
10	10.000	21	21.000
11	11.000	22	22.000
12	12.000	23	23.000
13	13.000	24	24.000

Table 3

Frequency codes for:
INDEXED RUN WITHOUT RAMP
FREE RUN WITHOUT RAMP
Low speed in **ZERO SEARCH PROCEDURE**

Code	Frequency (Hz)	Code	Frequency (Hz)
02	200	17	1.700
03	300	18	1.800
04	400	19	1.900
05	500	20	2.000
06	600	21	2.100
07	700	22	2.200
08	800	23	2.300
09	900	24	2.400
10	1.000	25	2.500
11	1.100	26	2.600
12	1.200	27	2.700
13	1.300	28	2.800
14	1.400	29	2.900
15	1.500	30	3.000
16	1.600		

Table 4

Frequency available values:
HIGH RESOLUTION INDEXED RUN WITHOUT RAMP
HIGH RESOLUTION FREE RUN WITHOUT RAMP

Frequency range	Available frequency step
From 1 Hz to 50 Hz included	1 Hz
From 52 Hz to 100 Hz included	2 Hz
From 105 Hz to 200 Hz included	5 Hz
From 200 Hz to 500 Hz included	10 Hz
From 525 Hz to 1200 Hz included	25 Hz
From 1250 Hz to 2400 Hz included	50 Hz
From 2500 Hz to 4800 Hz included	100 Hz

6.2- RAMP TIME CODES

Table 5		
Ramp time codes for: **INDEXED RUN WITH RAMP** **FREE RUN WITH RAMP** High speed in **ZERO SEARCH PROCEDURE**		
Code	Total ramp time 0 - 24 kHz (mSec)	Average acceleration (Hz / sec)
10	16	1˙500˙000
11	18	1˙333˙333
12	20	1˙200˙000
13	22	1˙090˙909
14	24	1˙000˙000
15	27	888˙888
16	30	800˙000
17	32	750˙000
18	34	705˙882
19	37	648˙640
20	40	600˙000
21	44	545˙454
22	48	500˙000
23	53	452˙830
24	60	400˙000
25	68	352˙941
26	80	300˙000
27	87	275˙862
28	96	250˙000
29	107	224˙299
30	120	200˙000
31	137	175˙182
32	160	150˙000
33	192	125˙000
34	206	116˙504
35	240	100˙000
36	288	83˙333
37	320	75˙000
38	360	66˙667
39	480	50˙000
40	720	33˙333
41	960	25˙000
42	1˙440	16˙667

NOTE: total ramp time as indicated in the table is the duration of the ramp to reach a 24˙000 Hz frequency starting from motor stopped. In case of lower frequency set, the effective time is reduced approximately as the frequencies ratio, as showed in the following example.

EXAMPLE: if an *Indexed run with ramp* instruction has a ramp time code of 34 (206 milliseconds) and a frequency code of 15 (15˙000 Hz), the effective ramp time will be approximately:

$$T = 206 \times (15 / 24) = 129 \text{ milliseconds}$$