Socapel ST1

A Digital Motion Controller

Software for the synchronization of axes

Ordering Number: 024.8068.A

Rev. August 1993

This upgraded and improved version replaces all the previous. We reserve the right to amend this document without prior notice and decline all responsibilities for eventual errors.

Atlas Copco Controls SA
En Montillier 4
CH-1303 PENTHAZ
Switzerland

Doc. No. 024.8068.A/BS

CONTENTS

1. INTRODUCTION

In addition to the power stage, the ST1 digital motion controller comprises a microprocessor-based control unit. It therefore requires the software described in the present manual.

Whether delivered with the ST1 motion controller or on a separate non-volatile EPROM memory board (or any other support), the present software remains the exclusive property of Socapel SA, Penthaz, Switzerland. Copyright law prohibits the copying of this software, even partially, by whatever method, without the written consent of the Owner. It is equally prohibited to disassemble the software content.

By purchasing the ST1 digital motion controller, the User receives the right to employ the software for its intended purpose.

2. HARDWARE

2.1 MICROPROCESSOR

The ST1 digital motion controller uses the TMS 320C15 microprocessor manufactured by Texas Instruments. This processor is particularly well adapted for fast numerical handling of analog signals.

2.2 PROGRAM MEMORY

The ST1 operating software is stored in a non-volatile EPROM memory marked with the following information :

- The version number : V0001, V0002 etc.
- The check-sum.
- The programming date of the component.

Each new edition of the software has a separate version number, even when only minor changes have been implemented. The version number is also stored in the memory as the VSOFT parameter.

The program memory is mounted in a socket permitting replacement. To avoid damaging the chip, we recommend the use of an extraction tool (for example : Bernstein 2-601).

Figure 2.1 shows the location and positioning of the memory. Also shown is bridge S4, which has the following function :

- Pins 1 and 2 shorted : Memory size 8 or 16 kbytes (standard).
- Pins 2 and 3 shorted : Memory size 32 kbytes (option).

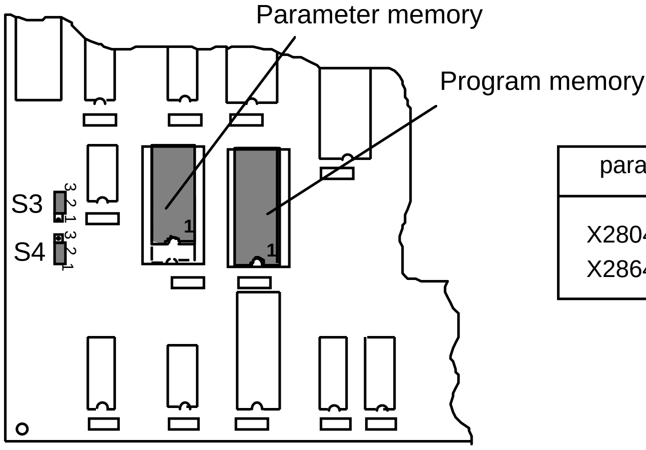
Note : All ST1 digital motion controllers shipped before mid-1990 were fitted with TMS 32010 microprocessors. All software versions are compatible with both processor types, except those with version number V0500, which require the TMS 320C15. If necessary, the processor may be exchanged : please consult your supplier.

The software can also be downloaded : refer to the correspon-ding documentation.

When the ST1 is fitted with the software-downloading option, the program memory provides for a boot program.

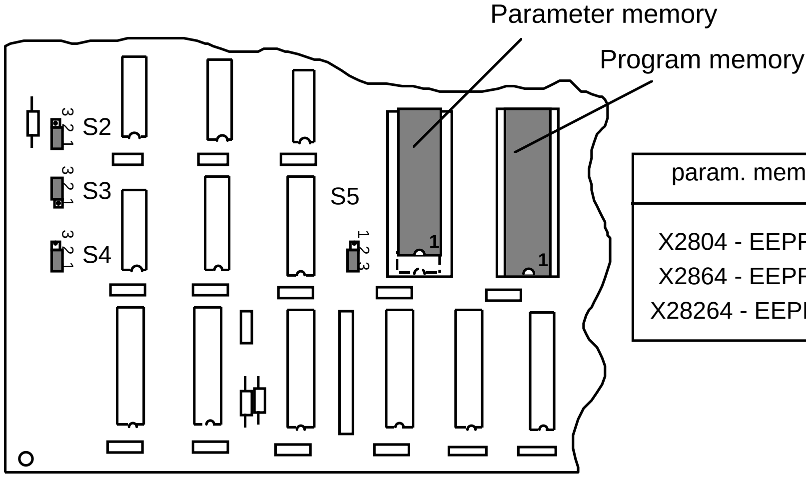
(See § 4.3.)

a



param. memory	S3	S4
X2804 - EEPROM	2-3	1-2
X2864 - EEPROM	2-3	2-3

b



param. memory	S3	S5
X2804 - EEPROM	2-3	2-3
X2864 - EEPROM	1-2	1-2
X28264 - EEPROM	1-2	1-2

Figure 2.1 : Location of program and parameter memories.

The drawing shows the lower part of the R-board which is located on the right-hand side of the ST1.

a) Equipment delivered up to 1991 (part numbers 024.77xx and 024.77xx.A)

b) Equipment delivered as from 1991 (part numbers 024.77xx.B and later)

2.3 PARAMETER MEMORY

Chapter 3.3 of this manual defines the notion of "parameter" in general, and the notion of "user's initial values" in particular. In the standard version, parameters are stored in an electrically erasable/programmable memory (EEPROM type XICOR X2804 or X2864 or equivalent). For the downloaded-software version, only a XICOR X2804 EEPROM (or equivalent)may be used.

As for the program memory, the parameter memory is mounted in a socket. The same precautions should be taken in case of any exchange. Figure 2.1 shows the location and positioning of this memory, and the correct insertion method for memories fitted with 24 or 28 pins :

- 28 pins : X2864 or X28256 EEPROM (standard),

- 24 pins : X2804 EEPROM (exceptions).

Positioning of mini-bridge S3 and S5 depends on the memory type, as shown in Figure 2.1.

2.4 CYCLE TIME

The internal oscillator of the microprocessor produces a clock signal whose frequency is determined by a crystal with an accuracy of ± 100 ppm.

A PAL (programmable array logic) counter circuit determines the cycle time independently from the software. This time is set at exactly 1/3 millisecond for the standard-version ST1.

Certain software versions may require different cycle times. The PAL circuit is therefore mounted on a socket to allow later exchange. Figure 2.2 shows the location as well as the installation method. To distinguish between versions, the PAL chip bears a label which may read, for example, "333V4", which is the cycle time in microseconds, and the version number.

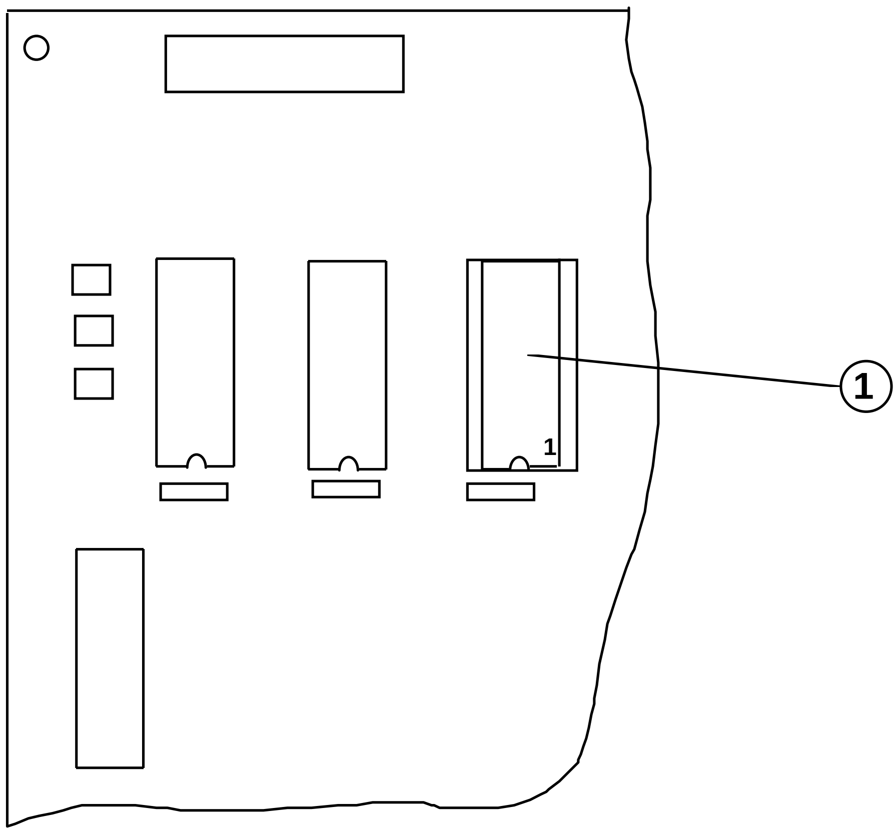


Figure 2.2 : Location of the PAL circuit.

The drawing shows the upper part of the R-board (located to right-hand side of ST1), with PAL circuit (1) .

3. SOFTWARE DESCRIPTION

3.1 PROGRAM

Due to the numerical regulator algorithm used in the microprocessor, the ST1 digital motion controller can be adapted to very varied applications by simply exchanging the software. The software is structured by functions, to facilitate the creation of new application-dependent versions as required.

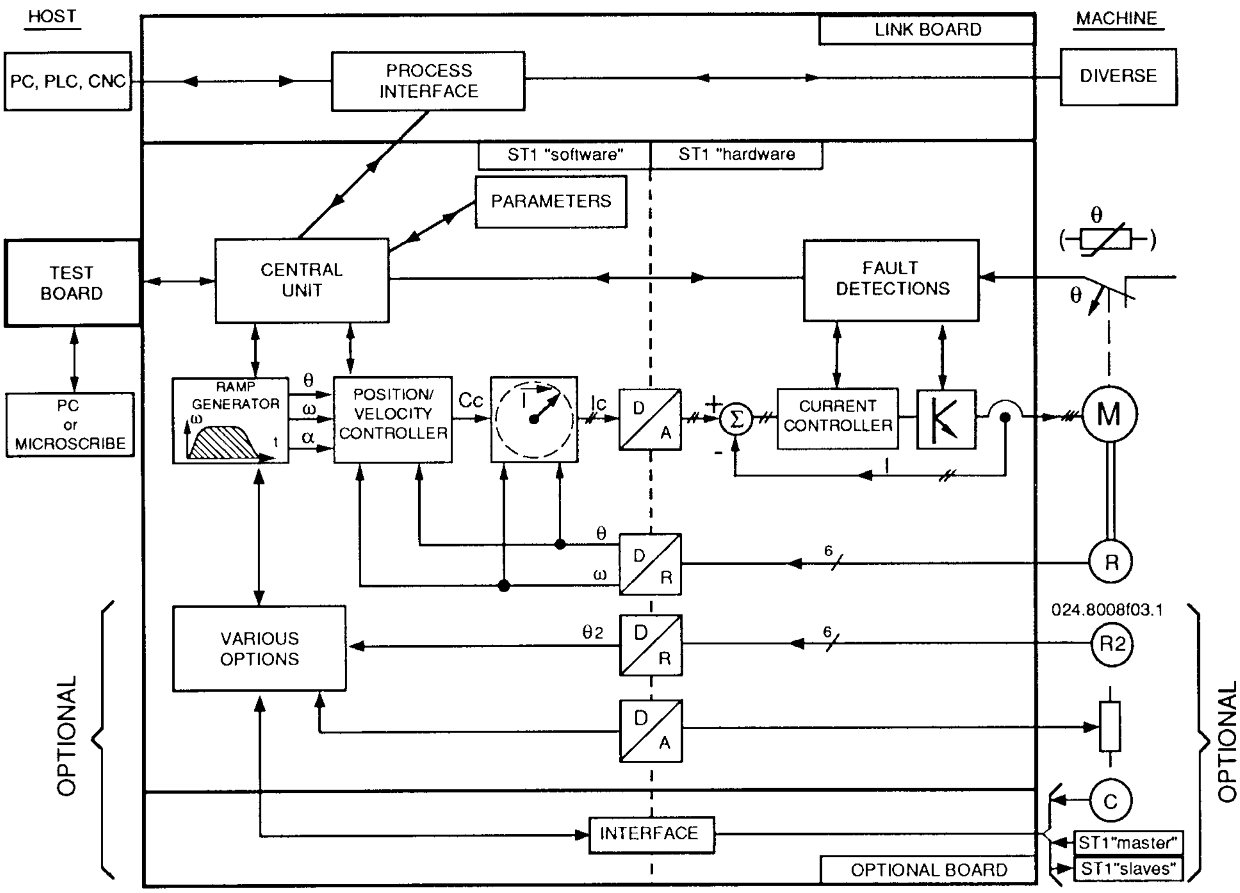


Figure 3.1 : Functional blocks.

Shown are the ST1 software (left) and hardware (right), as well as the link board and test unit.

The most important functions of the basic software are :

- The start-up sequence.
- Handling of variables and parameters.
- Fault monitoring.
- Measurement of the absolute position and the speed of the motor, using the resolver.
- Axis control.
- Current distribution to three-phase synchronous motor.
- Output of variables through a test card.

Added to this are special application-dependent functions such as :

- Absolute positioning using a dual resolver (allows overriding limit switches in certain cases).
- Control of three-phase induction motors with magnetizing current reduction, which permits speeds up to four times the nominal speed.
- Serial communication with a host computer, allowing interactive dialog and downloading of parameters.
- Generation of acceleration and deceleration ramps, for motor speed-control or point-to-point movements.
- Mutual synchronizing of axes, or synchronizing of one axis in relation to a reference axis.

3.2 VARIABLES

Variables are numerical magnitudes, whose value at any instant depends on the software, the initialization parameters, the interpretation of external events, and running of the process.

Except for a few exceptions, the variables of the ST1 digital motion controller are stored in the volatile memory (RAM), internal to the microprocessor. They respond to a specific address bus whose address ranges from 0 to 255 (hexa : 0...FF). (The TMS 320C10 microprocessor limits this range to 143 (hexa : 8F).

3.3 PARAMETERS

Parameters are numerical magnitudes with values defined by the user. They influence the initialization process as well as the effect of the software and the associated variables. Their value may be modified by the user or by the command host at any time, including during operation.

The ST1 digital motion controller distinguishes between the following values for each parameter :

- The "default value", as defined by the firmware;
- The user's "initial value", as defined by the user and stored in the parameter memory;
- The "current value" which is the only one used in the calculation by the software; this value is stored in a volatile memory (RAM) - the so-called 'working storage'. Its is located at address 52 to 255 (hexa : 34 ... FF) of the main address bus.

During each start-up as well as after a reset command, the current value of each parameter is determined as follows :

- If the parameter memory is in place, and if the compatibility codes correspond, the user's stored values are valid. The seven-segment status display on the ST1 indicates an "8".
- In the opposite case (and particularly if the ST1 is new or if the EEPROM is missing), the default values are valid. The status display will indicate a "9".

> ***Comment*** : If the ST1 digital motion controller detects a fault, the status display may indicate another code. See paragraph 5.3.

Subsequently, the current value may be changed :

- By the user, using a programming terminal or a personal computer + test unit;
- By the host computer, during downloading through the serial link;
- By the built-in sequencer (if using the LIO link board).

The current values of all parameters in the parameter memory can be transferred at any time, under one condition : the corresponding memory must be of the standard type (EEPROM). It is thus possible to define new initial values.

The saving procedure starts with the "SAVE" command from the user or from a host computer at the end of a remote loading sequence. Saving takes a maximum of eight seconds. For certain software versions, bit 3 of status STATC indicates the beginning and the end of the procedure. See paragraph 6.2.3.

3.4 PRECISION FORMATS

The ST1 microprocessor processes all values (whether variables or parameters) in fixed-point, 16-bit format.

Certain values are processed with double (or even triple) precision. The variable or the parameter is then represented by two (or three) words of 16 bits, hence extends over 32 (or 48) bits. In this case the suffix "A" indicates the least significant word of the variable or the parameter considered; the suffixes "B" and "C" indicate more significant words in the command.

Examples :

VIREF		=	+ 1,890,071,254	(hexa : 70A8,32D6)
gives	VIREFB	=	28,840	(hexa:70A8)
	VIREFA	=	13,014	(hexa : 32D6)
PHIRE		=	-164,831,062,250	(hexa: FFD9,9F4D,95EA)
gives	PHIREC	=	- 39	(hexa : FFD9)
	PHIREB	=	40,781	(hexa : 9F4D)
	PHIREA	=	38,378	(hexa : 95EA)

The "ST1-EXPERT", which is a PC-DOS based software for commissioning of ST1 digital motion controllers, is able to work using either decimal, hexadecimal or binary form as required. The "Microscribe" terminal only knows about hexa-decimal numbers.

4. TOOLS FOR PARAMETER SET UP AND COMMISSIONING

4.1 TEST DEVICE

The characteristic of every numerical controller is its ability to handle data by transferring it from one memory location to another. The data is therefore inaccessible without the addition of supplementary hardware.

In the case of the ST1 digital motion controller, this hardware is contained on a special test device which is used when the equipment is being set up or tested. The device permits direct access to any byte (8 bits) handled by the microprocessor. It also links two analog values for oscilloscope display. The choice of byte or analog value is made by entering parameters as described in paragraph 7.7.

The test device also includes a serial RS232 link for connection to a personal computer (PC) or a programming terminal.

4.2 USE OF A PERSONAL COMPUTER (PC)

The MS-DOS compatible ST1-EXPERT software includes features for linking a personal computer (PC) to a ST1 digital motion controller. Please read manual No. 024.8038 "ST1-EXPERT" for further details.

4.3 SOFTWARE DOWNLOADING

Standard ST1 digital motion controller is fitted with an EPROM which supports the appropriate software (see § 3.1).

As an option, it can be fitted with a software-download facility. This provides for a better interchargeability in applications where different softwares are to be used. Software-updates are also simpler to implement.

Upon delivery (and unless otherwise stated), no downloaded software is stored. The ST1 runs a "boot" program which provides for the following functions :

- Operating system and fault detection :

  The standard operating system of ST1 is running, with its fault detection, display and status. But these are no functions for controlling the motor. The power stage cannot be enabled. The resolver interface is not running.

- Serial link :

  Two interfaces for serial link are available :

  - **over the test device**, using the fixed length protocole (see document 024.8020 "Link board LIO"). This part is to be used with a PC and "The Socasin EXPERT" software.

  - **over the link board,** using following protocoles :

    - LS and LIO boards : fixed frame protocole (see document 024.8020 "Link board LIO")

    - LPO board : PAM-ring protocole

    - LA boards : no interface available.

- Specifical status bits :

  § 5.3 gives all necessary informations.

- Reduced and specific instruction set :

  Only those standard instructions specific to the reading and the setting of variables parameters and similar values are available. They are descripted in § 10.1 through 10.4, whereas their availability in the boot software is mentioned.

  Further instructions which are specific to the downloading procedure are available, but only in the boot software. They are descripted in § 10.9.

4.4 DOWNLOADING PROCEDURE

This paragraph describes the software download procedure, which has to be implemented into the host command when ST1's link board is of LS or LIO type.

In such cases, the communication for downloading protocole is always ST1's "fixed frame protocole", as described in document 024.8020 "Link board LIO".

Should the ST1 be fitted with an LS link board, (and only in that case), then most downloaded softwares use ST1's "optimal length protocole", as described in document 024.8012 "Link boards LS". The host command must then be able to use both protocoles.

All downloadable firmwares are available as PC-DOS files (JEDEC format).

The first line of JED file start with '0' or 'K' and the first 13 characters are not available. Then the format of Jedec"Comand" is made of 5 characters strings:

x a b c d

where : x = string type

'0' or 'K' = skip 13 characters

'9' = new address pointer

'B' = data

'7' or '8' = end of line

':' = end of file

a b c d = address pointer or data (16 bits, in hexadecimal code)

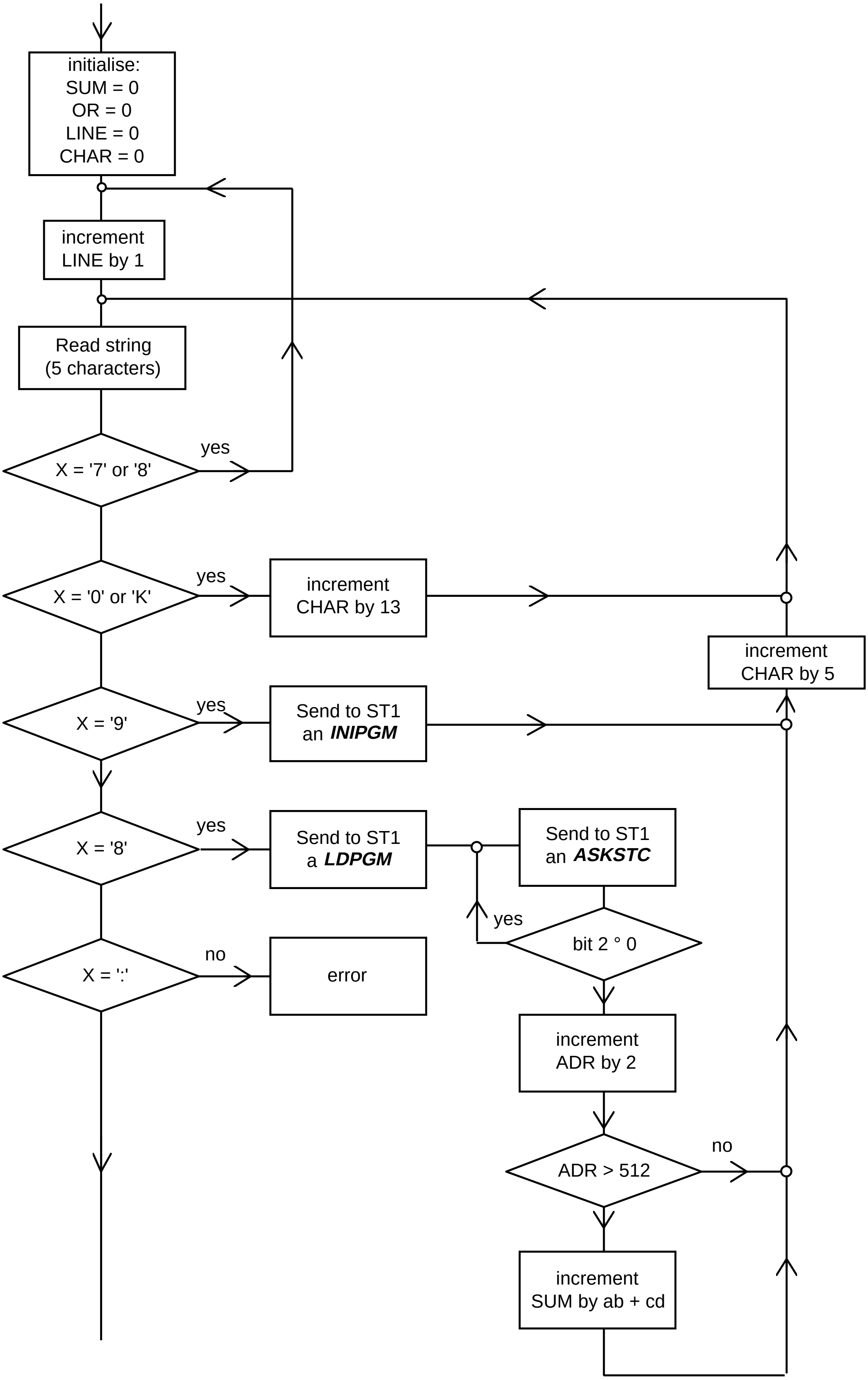
The corresponding "Jedec Comand" and ST1 instructions are listed below:
Ubon reading on "9abcd" string, the host has to send a "INIPGM" instruction to the ST1 (6 bytes) as follows.

	Instr.Code		Data bytes		
address	INIPGM	0	a b	c d	checksum

Upon reading on "B a b c d" string, the host has to send a "LDPGM" instruction to the ST1 (6 bytes) as follows :

	Instr.Code		Data bytes		
address	LDPGM	0	a b	c d	checksum

The whole procedure is shown in the following flow-chart :


initialise:
SUM = 0
OR = 0
LINE = 0
CHAR = 0
increment
LINE by 1
Read string
(5 characters)
X = '7' or '8'
yes
X = '0' or 'K'
yes
increment
CHAR by 13
increment
CHAR by 5
X = '9'
yes
Send to ST1
an *INIPGM*
X = '8'
yes
Send to ST1
a *LDPGM*
Send to ST1
an *ASKSTC*
yes
bit 2 ° 0
X = ':'
no
error
increment
ADR by 2
ADR > 512
no
increment
SUM by ab + cd

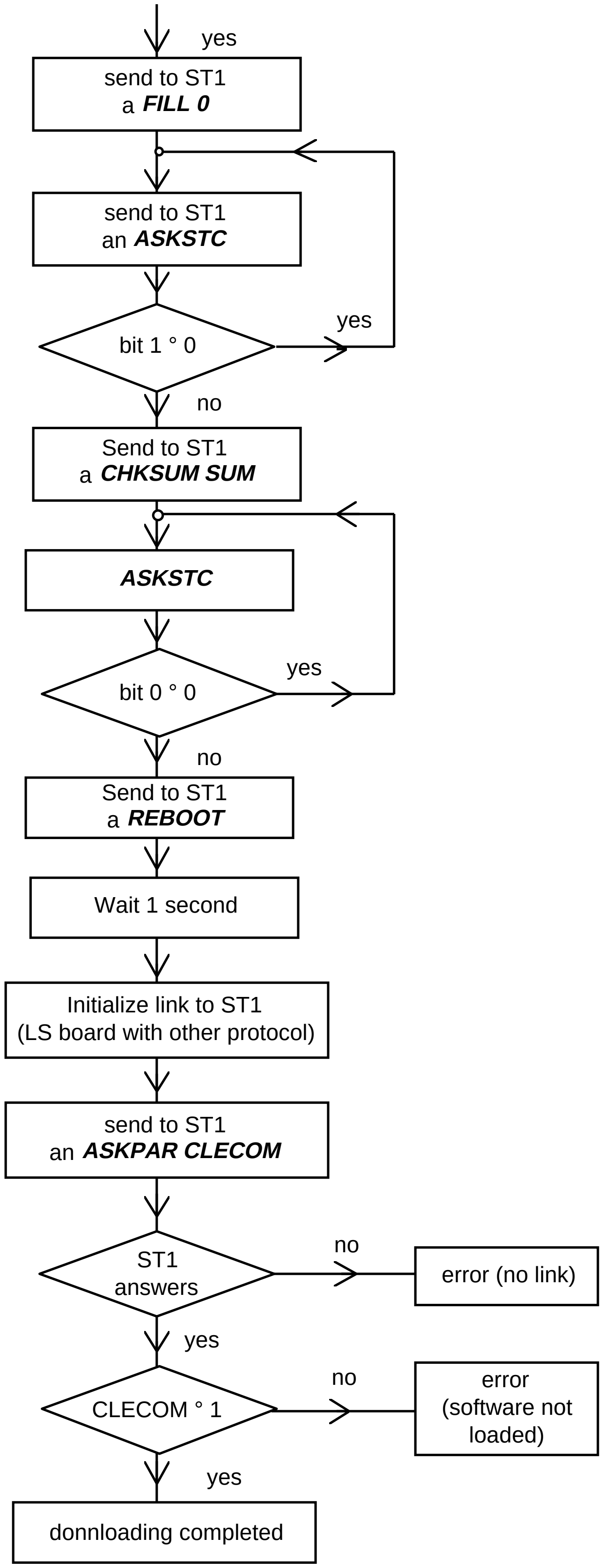


Figure 4.1 *: Dounloading flow-chart*

Before downloading a software to a ST1, this one must be running the "boot" software. If not, the previously loaded software has to be "killed", as follows :

- initialise link to ST1 (with the proper protocole)
- send to ST1 a "SETOUT >4055 0" instruction (Desable Download program)
- send to ST1 a "SETPAR >23 0" instruction (Reset ST1)
- wait for about 1 second. (Until ST1 Display "6")
- initialise link to ST1 (with fixed frame protocol)

It is also possible to activate the "boot" software momentarily by pushing the "reset" button on test device while pressing the ST1 "reset" button. In that case, the software runs its "boot" software and displays a "6".

5. STATUS DISPLAYS

5.1 PRINCIPLE

The ST1 digital motion controller has a 7-segment LED status display on the front panel. This display indicates in coded form the last important event which occurred in the unit.

This may be a matter of normal events (for example : enabling the power stage) or abnormal events (for example : lack of auxiliary supply). Paragraph 5.3 lists the significance of each code.

5.2 SIGNIFICANCE OF THE DECIMAL POINT

In addition to the coded display, the decimal point indicates status of the ST1 digital motion controller :

- Decimal point extinguished : This signals that the equipment is not being fed with correct auxiliary supply (+ 5 VDC shorted or out of spec.).

- Decimal point dimmed : Indicates that the equipment is being fed correctly, but that the software does not function correctly (watch-dog error in the microprocessor).

- Decimal point strongly lit : The equipment is fed correctly and the software is functioning.

In case of doubt, it is possible to make a distinction between the last two states by resetting the microprocessor :

- Press the "RESET" button on the ST1 digital motion controller. The decimal point should be dimmed.

- Release the "RESET" button. The brightness of the decimal point should increase significantly.

5.3. STATUS CODES

The following codes are valid only when the decimal point is lit strongly.

"0" : The power stage has been disabled as a result of an external command and not because of a fault.

"1" : The power stage has been enabled.

"6" : No firmware available; boot program is running (§ 4.3).

"7" : Reboot and load the downloading software

"8" : Unit initialized, with the transfer of the user's initial values as current values of the parameters.

"9" : Unit initialized, with the transfer of the default values as current values of the parameters.

"A" : The power stage has been disabled as a result of a failure in the 48 VDC auxiliary supply supply.

"C" : The power stage has been disabled as a result of a fault in the resolver feedback circuit.

"d" : The power stage has been disabled as a result of an internal failure :

- Auxiliary supplies + 5 V, + 15 V, - 15 V are out of tolerances.
- Defective software ("watch-dog" has been triggered).

"E" : Parameter memory or jumpers not ok (§2.3).

"H" : A checksum error has been seen after downloading a software (§4.4).

"L" : The power stage has been disabled as a result of exceeding parameter programmed limit positions.

"M" : Internal hardware error.

"o" : The power stage has been disabled by the variables STATC/STATD as set by the mask parameter CMASKU. See further explanations in paragraph 6.3.

"P" : The power stage has been disabled as a result of an over voltage condition in the DC bus.

"r" : Fault test completed successfully.

"S" : The power stage has been blocked as a result of an overload or a short condition.

"U" : Memory overrun while downloading a software (§ 4.4).

Other codes may appear, depending on the configuration options (particularly, the "software downloading" option). Refer to the relevant documentation.

6. STATUS BYTES AND MASKS

6.1 GENERAL

The general condition of the ST1 operation can be observed by looking at four status bytes which summarizes the conditions of the unit. Each piece of information is assigned to one bit.

The fault indicating bits are latched. These bits can trigger an interrupt in the fault handling routine of the ST1 digital motion controller. Those bits which simply indicates a state or condition are not latched.

The status bytes are called STATA, STATB, STATC and STATD. Figure 6.1 shows the numeration of the corresponding bits, bytes and words.

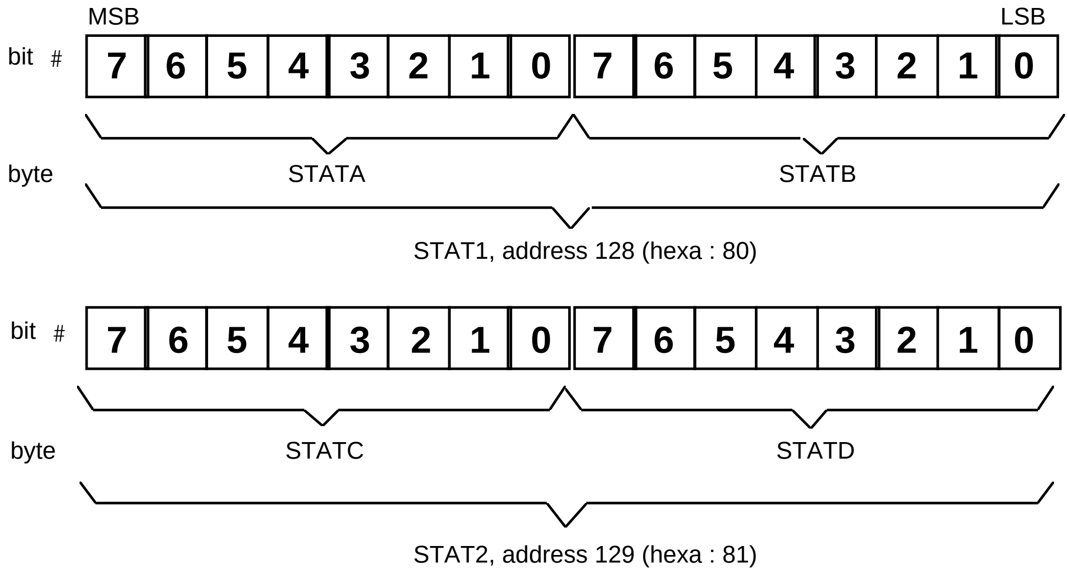


Figure 6.1 : *Status bytes.*

Those status bits which are significent while the boot program is running, or while downloading a software are marked with a " * " . (see also § 4.4).

6.2 STATUS BITS

6.2.1 Status STATA

The STATA byte provides essential information on the status of the ST1 digital motion controller and its software.

Bit 7 (latched) signals that the power stage has been disabled as a result of a fault. Normal disabling will not have any effect on this bit.

Bit 6 (latched) informs that there has been a change in the two bytes of the STATC and STATD status as a function of the mask MASKS. See paragraph 6.3.

Bit 5 indicates whether the real position is less than ("1") or greater than ("0") the value set by the instruction SPWARN (see paragraph 10.2.4).

Bit 4 signals that the position lag (variable DPOS) is greater than ("1") or less than ("0") the value set by the parameter SEUIL1. It provides an overall functional control of the axis, allowing the disclosure of any occurrence of excessive friction or a cable rupture, for example. It is resumed under the memorized format in STATD. In certain versions of the software, velocity lag DVIT is monitored instead of the position lag if parameter KTEGR = 32,768 (hexa : 8000).

Bit 3 signals that the move requested by an instruction ERMOV, ERUN, START, STOP etc. is in progress. The bit is set to "1" when one of the above instructions is received and returns to "0" when the corresponding move is finished.

Bit 2 is used by those software versions which require computation time between the receipt of a move order and the execution of it. It is set to "1" when the computation is finished, which means that a START order will follow immediately. It returns to "0" as soon as the move has started

Bits 1 and 0 give information about the status of the motor speed during the execution of a move of the axis :

Bit 1	*Bit 0*	*Meaning*
0	*0*	*Stop (reference speed zero)*
0	*1*	*Acceleration*
1	*1*	*At speed (reference speed not zero)*
1	*0*	*Deceleration*

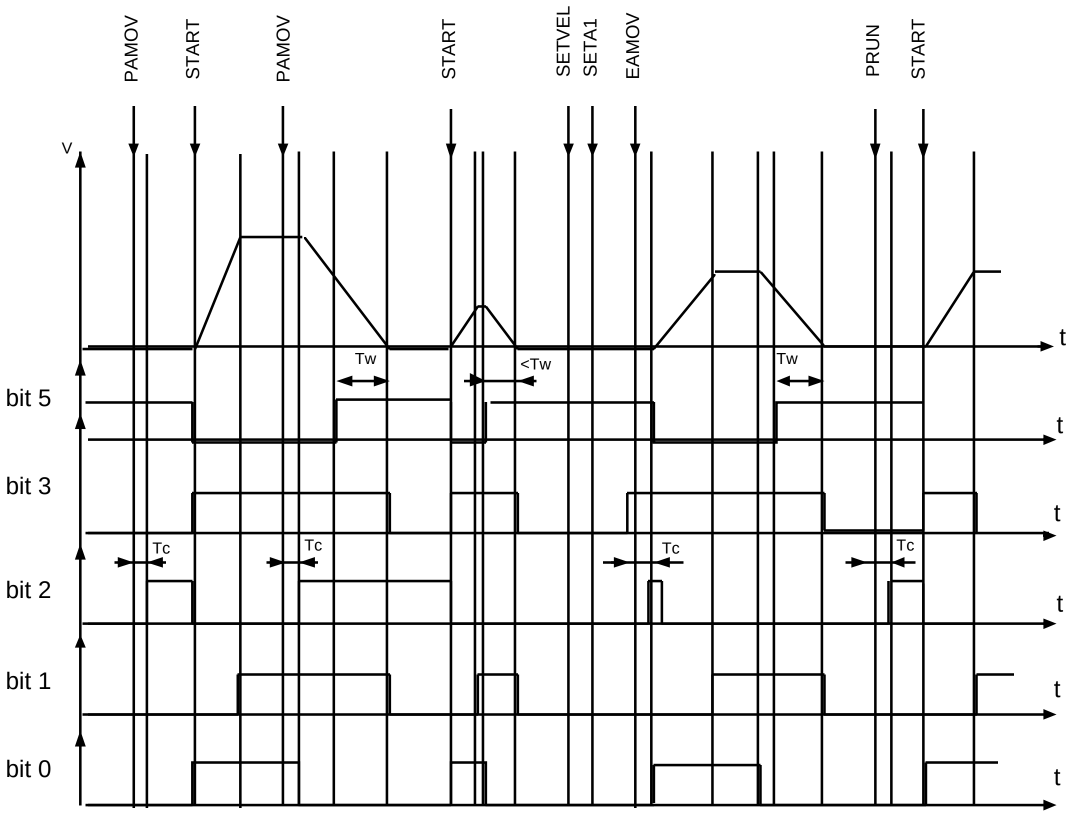


Figure 6.2 : *Development of bits 5, 3, 2, 1 and 0 of STATA.*

Abbreviations : *Tw = delay of warning TWARN;*
Tc = duration of computation (< 30 ms).

6.2.2 Status STATB

The STATB byte indicates such faults which have immediately resulted in the disabling of the power stage. It allows you to find the reasons for an undesired stop of the axis.

Bit 7 (latched) indicates the amplifier has been disabled due to a failure in the resolver feedback signal.

* ***Bit 6 (latched)*** indicates that the DC bus voltage UA has exceeded its upper limit.

* ***Bit 5 (latched)*** indicates that a saturation fault in the power transistors is the source of the disabling. Possible causes : short circuit or overload.

* ***Bit 4 (latched)*** signals that an internal failure in the auxiliary voltage is the source of the disabling.

* ***Bit 3 (latched)*** indicates that a momentary interruption of the auxiliary supply is the source of the disabling.

* ***Bit 0*** indicates that the boot program is running (see § 4.3 and 4.4).

6.2.3 Status STATC

The effects of the bits of variable STATC depend on the masks, as described in paragraph 6.3. The listing which accompanies every software version, shows all bits having really a meaning. All the other bits remain set to "0".

- ***Bit 7 (latched)*** indicates that a not valid instruction code has been received.

- ***Bit 6 (latched)*** indicates that implausible instruction code or data has been received.

- ***Bit 5 (latched)*** indicates a firmware fault and must always be "0". Should it be set to "1" then please call immediately your supplier.

- ***Bit 4 (latched)*** indicates after powering up or resetting that the present link card is not compatible with the present software version, or that the card is missing. Afterwards, it indicates a serial link transmission failure.

Bit 3 is "1" as long as the SAVE routine is in progress, which saves all parameter actual values into the EEPROM parameter memory. It returns to "0" as soon as saving has been completed.

- ***Bit 2*** indicates that the master shaft has overrun the limit of one period, or that the zero track of an incremental encoder has been detected. Refer to paragraphs 8.5.6 and 8.8.1.

- ***Bit 1*** reproduces the state of input No. 2 of some link cards. If the entry is open (no current, zero voltage), the bit is at 1. If the entry is fed, it is at 0. While downloading a software (§ 4.4 and § 10.9.3) bit 1 indicates that the EEPROM memory filling in under way.

- ***Bit 0*** depends on the state of input No. 1 of some link cards, as bit 1. While downloading a software (§ 4.4 and § 10.0.4), bit 0 indicates that the checksum computation is under way.

6.2.4 Status STATD

The effects of the bits of variable STATD depend on the masks as STATC. The listing which accompanies every software version, shows all bits having really a meaning. All other bits remain set to "0".

- ***Bit 7 (latched)*** indicates excessive motor temperature, or that the corresponding input of the ST1 digital motion controller is not connected.

Bit 6 (latched) indicates that the internal overheating protection circuitry is active and reducing the current to the motor.

Bit 5 (latched) is a latched copy of bit 4 of STATA.

Bit 4 (latched) indicates that the real position has exceeded (even if only briefly) the limits set by parameters PHIL1/PHIL2. Refer to paragraph 7.8.

Bit 3 (latched) indicates that the ST1 digital motion controller has initiated a STOP procedure.

Bit 2 (latched) indicates a failure of the optional board.

* ***Bit 1 (latched)*** is used to determine if the microprocessorhas been reinitialized ("RESET") or not. The validity of the position measurement (number of turns) and the other variables depends on it. After each restart of the microprocessor, this bit is worth 1. It is put back to 0 by one of the initialization commands of status described in paragraph 6.4, and particularly when the power stage is enabled. Then, it can only go back to 1 with a new restart of the microprocessor.

Bit 0 indicates if the power stage is enabled ("0"), or disabled ("1").

6.2.5 Status STATE

Bit 7 indicates that a motor speed has exceeded the limit set by the parameter VITMAX · 1.125

6.3 STATUS MASKS

The bits of variables STATC and STATD may be used to automatically start one of the emergency procedures described below. The masks are the parameters matching these procedures to each bit.

In determining the value of the masks, it is necessary to recall that the STATC and STATD bytes are collected in the form of only one word. Bits 0 to 7 correspond to STATD, and bits 8 to 15 correspond to bits 0 to 7 of STATC (see Figure 6.1).

For every mask, a logical "AND" combination is executed bit to bit. If the result is not zero, the corresponding procedure is started.

> ***Example :*** If bit 7 of the mask is "1", any excessive motor temperature (bit 7 of STATD) starts the desired procedure. If the bit is "0", the motor temperature does not have any effect.

Three masks are available, each one corresponding to a specific procedure :

- ***CMASKU*** selects those bits which, when they are "1", cause a disabling of the power stage (no current to the motor).

  Address : 124 (hexa : 7C).

  Default value : 128 (hexa : 80)

- ***CMASKA*** selects those bits which, when they are "1", produce a controlled deceleration of the motor, that means a "STOP" (reference speed of the motor at zero). This mask is only active in software versions which support point to point move or the speed control by serial communication.

  With the LIO link board and software versions including sequencer functions, this CMASKA status initiates no STOP. It opens instead the output OUT7. See corresponding manual.

  Address : 125 (hexa : 7D).

  Default value : 0.

- ***CMASKS*** selects those bits which, in case of a change of status, set bit 6 of STATA to "1". Thus, in case of a serial communication, the command host is quickly informed about the occurrence of a change. The host can then inquire about the new status of the variables STATC and STATD.

  Address : 126 (hexa : 7E).

  Default value : 61,680 (hexa : F0F0).

  As the boot program is running, only CMASKS (with its default value) mask is active.

6.4 RESET OF STATUS BITS

The latched bits are automatically set to "0" at powering up and after the microprocessor has been reset (with the exception of bit 1 of STATD, which is set to "1"). They only return to "1" if the corresponding fault is detected.

The serial link or the sequencer function, if available, make it possible to reset all status bits using one of the RESSTS, PWRRES, PWRONS, PWRONI or PWRONR instructions. See also the corresponding paragraphs of chapter 10.

As the boot program is running, only instruction RESSTS is available for resetting the status bits.

7. DESCRIPTION OF BASIC PARAMETERS

7.1 GENERAL

The performance of the ST1 digital motion controller may be enhanced in the future. It is therefore wise to maintain a certain versatility in the number of the parameters.

Those described in this document concern most of the software versions in use today. For some particular functions additional parameters may be introduced; see the corresponding notices. In any case there is only one valid reference : the listing which describes each version.

When a user's initial value is not determined, it is essential to keep the default value planned by the creator of the software. All default values are indicated in the listing.

7.2 EEPROM MEMORY CONTROL PARAMETERS

7.2.1 Parameter CLECOM

This is a compatibility key. The parameter determines the validity of the default values or of the user's initial values, depending on the software introduced. If the value of CLECOM contained in the parameter memory (user's initial value) is the same as the one contained in the program memory (default value), all user's initial values are admitted as current values. In the opposite case, the default values are kept. See also paragraph 3.3.

Address : 52 (hexa : 34).

7.2.2 Parameter VSOFT

This parameter shows which software version is active.

Address : 55 (hexa : 37).

7.3 PARAMETERS FOR POSITION FEEDBACK

7.3.1 Parameter CURES

This parameter defines the resolver excitation voltage. The value is only input as a starting value. The real voltage value is automatically determined by the controller (variable URES). The CURES parameter value is only used for resolver fault supervision. The value depends on the type of resolver used.

Manufacturer	Type	CURES
Harowe	21 BRCX-310-F-8	1,280 (hexa : 500)
Harowe	21 BRCX-310-H-7	1,536 (hexa : 600)
Harowe	11 BRCX-300-A-95	256 (hexa : 100)
Précilec	26V 21 RN45PE-09	1,280 (hexa : 500)
Précilec	26V 20 RN45PE-03	1,280 (hexa : 500)
Thomson	26 SM 19 RX 452 C 0/F/00	1,536 (hexa : 600)
Thomson	26 TM 11 RX 4a ... k	256 (hexa : 100)
Sagem	22 RX 04 02 04 (A)	1,280 (hexa : 500)
Sagem	21 RX 34 02 01	1,280 (hexa : 500)
Sagem	21 RX 34 03 08	1,280 (hexa : 500)
Sagem	15 RX 37 01 18	1,280 (hexa : 500)

When using other resolver types proceed as follows :

- Turn on ST1 supplies.
- Read the value of the URES variable.
- Introduce and save this value as the CURES parameter.
- A difference not greater than 500 (hexa : 200) between the values URES and CURES is allowed.

Address : 56 (hexa : 38).

Default value : 1,536 (hexa : 600).

7.3.2 Parameters KVIRB, KVIRIN, KSIN and KSIINV

These parameters depend on the resolver secondary voltages. For the types listed above, the default values should not be modified, namely :

Address			Default value	
KVIRB	57	(hexa : 39)	19,700	(hexa : 4CF4)
KVIRIN	58	(hexa : 3A)	6,954	(hexa : 1B2A)
KSIN	59	(hexa : 3B)	3,277	(hexa : CCD)
KSIINV	60	(hexa : 3C)	5,120	(hexa : 1400)

7.3.3 Parameter NPPOL

See paragraph 7.4.1 : Motor file libraries. This parameter defines the relation between the number of motor poles and the number of resolver poles.

Example : Resolver having 2 poles (1-speed), and a motor with 8 poles : NPPOL = 4.

Address : 61 (hexa : 3D).

Default value : 4.

7.3.4 Parameter NVERN

This parameter is used when dual resolvers are used to obtain an absolute positioning system. In this case, NVERN is equal to the length of absolute measurement of the position, expressed in number of turns.

Example : For 2 resolvers (1-speed) connected with one ratio reducer 256/255, the measurement is absolute over 256 turns.

Then NVERN = 256 (hexa : 100).

Address : 62 (hexa : 3E).

Default value : 1.

7.4 MOTOR PARAMETERS

7.4.1 Motor File Libraries

For synchronous and induction motor types which have been used successfully with ST1 digital motion controllers, all corresponding parameters have already been determined and checked. They were brought into MS-DOS compatible files and libraries. Read manual No. 024.8038, "ST1-EXPERT", for further details.

7.4.2 Parameter CDEPHA

This parameter defines the phase relationship between the zero of the resolver and zero of the motor (if synchronous). The resolver should usually be installed according to a procedure described in the operation manual. CDEPHA is then set to 0. When system constraints prevent the resolver from being ideally aligned to the motor poles, the angular difference can be compensated by the CDEPHA parameter.

Address : 68 (hexa : 44).

Default value : 0.

7.4.3 Parameter KTINV

See paragraph 7.4.1 : Motor file libraries. This parameter defines the torque constant of the motor winding. The value 4,096 (hexa : 1000) corresponds to a motor which supplies a torque of 1 Nm when the current is 1A (peak value) or 0.71 A_{RMS}

If the motor supplier specifies the value K_T [Nm/A] using RMS value of the current, KTINV is calculated as follows :

$$KTINV = \sqrt{2} \cdot 4{,}096 / K_T$$

Example : If K_T = 0.76 Nm/A_{RMS}, KTINV = 7,622 (hexa : 1DC6).

For induction (asynchronous) motor the K_T value is generally not available. Proceed then as follows :

1) Nominal torque : M_n [Nm] = $\frac{P_n[W]}{n_n[rpm]} \cdot \frac{30}{\pi}$

2) "Active" nominal current : Ia_{RMS} [A] = $In_{RMS} \cdot \cos\varphi$

3) Constant torque : K_T = $\frac{M_n[Nm]}{Ia_{RMS}[A]}$

Address : 69 (hexa : 45).

Default value : 8,192 (hexa : 2000).

7.4.4 Parameter KIGLIN

See paragraph 7.4.1 : Motor file libraries. If the *motor* is of induction type (*asynchronous*), this parameter defines the ratio between the slip frequency and the actual current at nominal speed.

If s_N is the nominal slip [revs/second], I_N the nominal current [in A_{RMS}] and cos j the power factor, then KIGLIN is given by :

$$\text{KIGLIN} = \frac{p \cdot sN \cdot 494.3}{IN \cdot \cos\varphi}$$

It is recommended to increase the calculated value by 10%.

Example : A 4-pole motor (p = 2), with I_N = 11.4 A_{RMS}, cos φ = 0.85, v_N = 1,700 rpm and f_n = 60 Hz.

The slip s_N = (60/2) - (1,700/60) = 1.667 revs/s.

KIGLIN = (2 · 1.667 · 494.3 / (11.4 · 0.85)
= 170 (hexa : A8).

If the *motor* is of *synchronous* type, KIGLIN must be 0.

Address : 73 (hexa : 49).

Default value : 0.

7.4.5 Parameter IMAGNN (induction motor only)

See paragraph 7.4.1 : Motor file libraries. This parameter defines the nominal magnetizing (reactive) current rating If IN is the nominal current of the motor expressed in A_{RMS}, and cos j its nominal power factor, then the value of IMAGNN is obtained by :

$$\text{IMAGNN} = 128 \cdot \sqrt{2} \cdot I_N \cdot \sin\varphi$$

Note : Due to the effect of saturation a 10 to 50% lower value may be considered.

Example : A motor with I_N= 17 A_{RMS} and cos φ = 0.85. The following is obtained :

$\sin\varphi = \sqrt{1 - \cos^2\varphi} = 0.527$

IMAGNN = 1,621 (hexa : 655)

Address : 74 (hexa : 4A).

Default value : 384 (hexa : 180)

7.4.6 Parameter VITN (induction motor only)

See paragraph 7.4.1 : Motor file libraries. This parameter defines the highest (base) speed of the motor at rated nominal magnetizing current. The ST1 digital motion controller provides rated magnetizing current up to that speed. For higher speeds the ST1 digital motion controller automatically reduces the magnetizing current.

The value of VITN is given by :

VITN = base speed [rpm] . 256 / 704

Example : The base speed is 1400 rpm.

VITN = 1,400 · 256 / 704 = 509 = (hexa : 1FD).

Address : 75 (hexa : 4B).

Default value : 1,092 (hexa : 444).

7.4.7 Parameter VITMAX

See paragraph 7.4.1 : Motor file libraries. This parameter defines the maximum speed of the motor as determined by the motor manufacturer. The ST1 digital motion controller limits the frequency in such a way that this speed is never exceeded, no matter which reference speed has been set. Bit 7 of STATUS E is set, if the speed is limited. However, a lower value may be used if required.

The value of VITMAX is given by :

VITMAX = maximum speed [rpm] · 256 / 704

Address : 76 (hexa : 4C).

Default value : 1,092 (hexa : 444).

7.5 POSITION CONTROLLER PARAMETERS

7.5.1 Feed-forward Commands

In all conventional controllers a torque command can only be produced when a velocity or position lag exists. Usually, an integrator is used to eliminate this lag. However, this method is ineffective for dynamic speed changes and produces disturbing overshoots.

Nevertheless, the torque required from the motor to produce a given degree of movement can largely be estimated : even where the inertia load varies, torque can usually be estimated to within 20 percent. Friction loads (dry or sliding), or the influence of gravity (on vertical axes), can be calculated or approximated empirically, allowing the load to be "modeled".

Modeling consists in knowing the desired position, speed and acceleration at any moment, and then calculating the required motor torque for this model. As it is applied as an open loop, this feed-forward control method never produces instabilities.

Of course, a real system performs differently from the model. But the position controller only has to correct for the difference between the theoretical torque and the actual system parameters. Since this difference is always the lesser value, positioning error is reduced accordingly.

In each application, the obvious aim is to model as accurately as possible, either by theoretical calculations or testing. Only afterwards is control coefficient I determined. In many cases this coefficient will be zero.

7.5.2 Position Controller

The ST1 digital motor controller software includes a PID position controller with feed-forward commands. The position feedback is derived from a resolver mounted at the end of the motor shaft. No tachometer is required. The regulation parameters can be adjusted experimentally or can be premodeled as shown later.

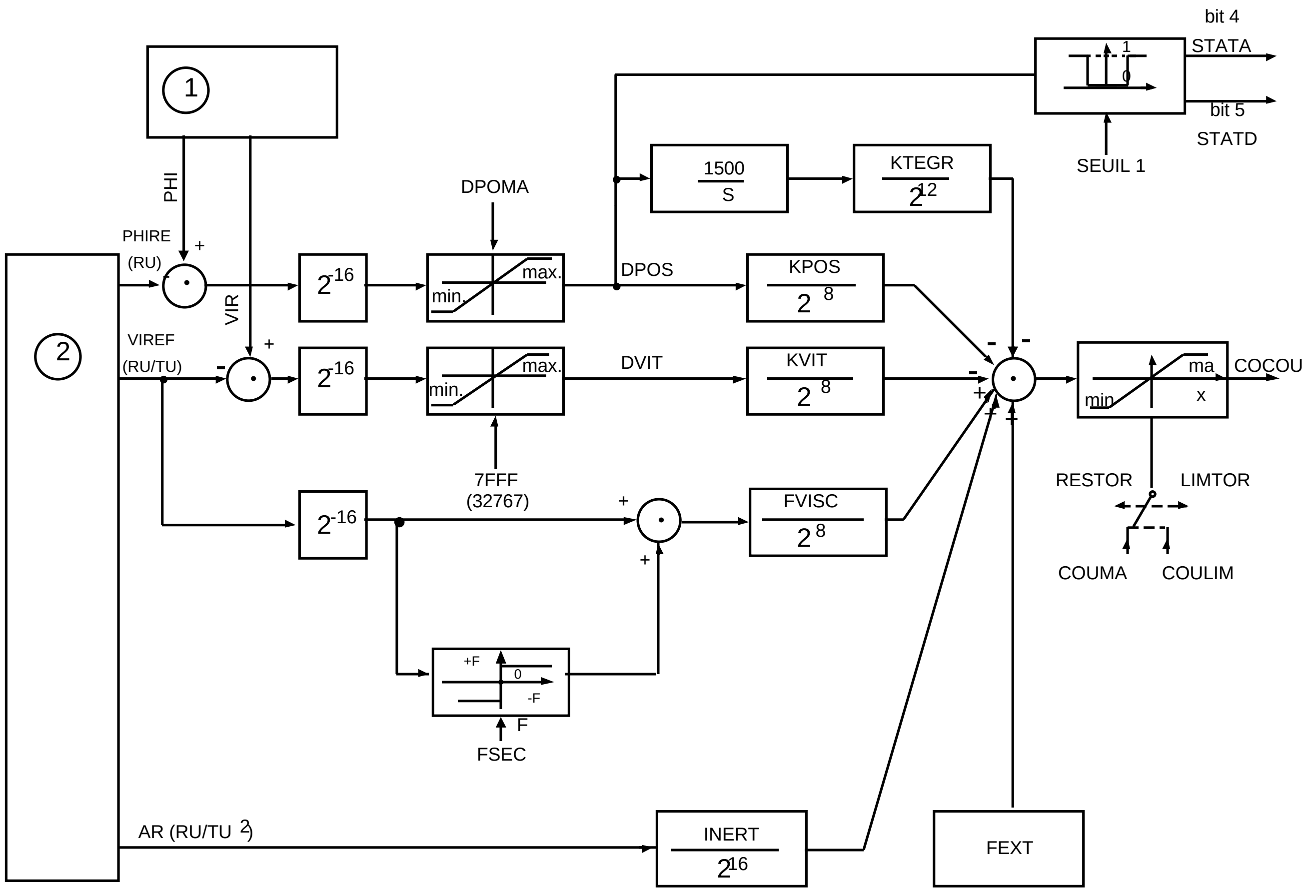


Figure 7.1 : *PID position controller.*

The position and velocity feedbacks are provided through a resolver to digital converter (1). The set values come from the "host" command or from the internal velocity profile generator (2). The position lag (DPOS) and the velocity lag (DVIT) are reversed compared to the usual conventions; a positive value corresponds to an advance.

7.5.3 PI Speed Controller

In many software versions, the PID position controller can be replaced with a PI speed controller by setting the KTEGR parameter to 32,768 (hexa : 8000). The motor speed is then regulated independently of the position. Figure 7.2 shows this configuration.

This feature is useful when only the speed of the axis is to be controlled and a large velocity lag is expected, due to a torque limit for example. In this mode of operation the position controller normally adds up the entire lag, and cannot stop the motor before bringing it back to zero. With the PI controller this problem is eliminated.

When operating in this mode, therefore, it is the velocity lag (DVIT) that is checked. The KPOS parameter controls the integration constant of the speed loop and the DPOMA parameter controls the level of integration.

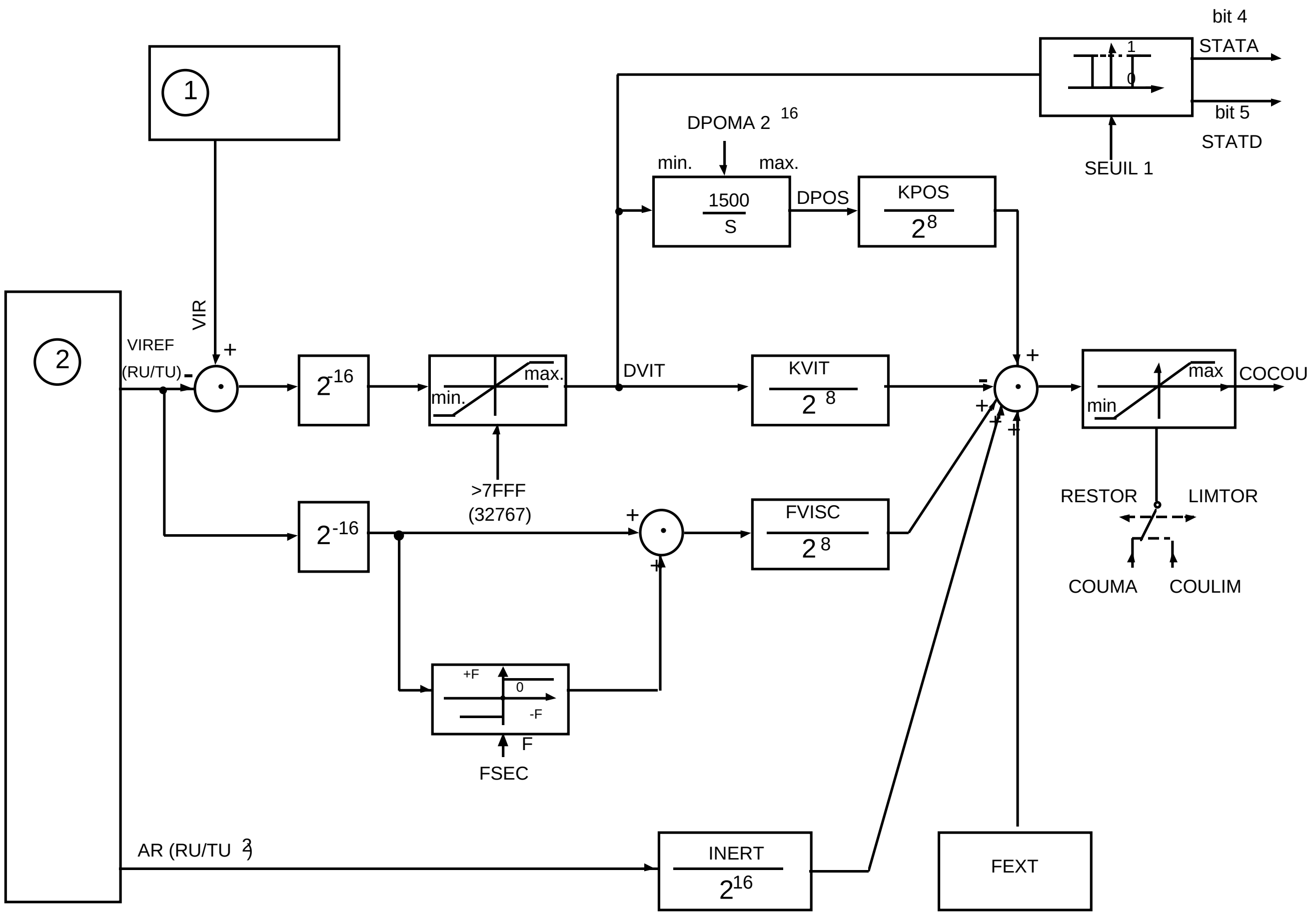


Figure 7.2 : *PI speed controller. Position lag DPOS has no effect.*

7.5.4 Parameter KPOS

When the unit is operated as a position controller the KPOS term determines the proportional gain (block P). When operated as a speed control KTEGR is set to 32,768 (hexa : 8000), and KPOS determines the integration constant (I) of the speed controller. In both cases, it influences directly the relation between the output torque and the corresponding angular distance.

The KPOS value should be positive between 1 and 32,767 (hexa : 1 ... 7FFF). When KPOS = 256 (hexa : 100), an angular distance or position error of 0.0123 radians produces a counter-torque of 1Nm.

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 80 (hexa : 50).

Default value : 48 (hexa : 30).

7.5.5 Parameter KVIT

This parameter determines the derivative gain (D) of the position controller, or the proportional gain (block P) of the speed controller. It is required to ensure the stability of the system.

The KVIT value should be positive between 1 and 32,767 (hexa : 1 ... 7FFF). With KVIT = 256 (hexa : 100), a velocity error of 2.30 rad/s produces a counter torque of 1 Nm.

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 81 (hexa : 51).

Default value : 96 (hexa : 60).

7.5.6 Parameter KTEGR

This parameter determines the integration constant of the position controller (I) when the amplifier is operated as a position control. It also serves as a choice whether the unit should operate as a position or velocity controller as explained in paragraph 7.5.3.

When the position regulator is active, the KTEGR value should be positive between 0 and 32,767 (hexa : 0 ... 7FFF).

To select the speed regulator, set KTEGR = 32,768 (hexa : 8000).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 82 (hexa : 52).

Default value : 0.

7.5.7 Parameter INERT

This parameter indicates the total system inertia. It greatly reduces the second-order lag due to the load inertia during the acceleration. If the total value of the load inertia as seen by the motor shaft JT, expressed in kgm2, is known, the value is obtained by :

INERT = 119,000 · J_T.

INERT should be positive or zero between 0 and 32,767 (hexa : 0 ... 7FFF).

Example : If inertia J_T = 0.012 kgm^2, the following is obtained :

INERT = 1,428 (hexa : 594).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 83 (hexa : 53).

Default value : 0.

7.5.8 Parameter FVISC

This parameter specifies the viscous damping constant. It allows a reduction of the first order error without introducing additional integration. FVISC should be positive or zero between 0 and 32,767 (hexa : 0 ... 7FFF).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 84 (hexa : 54).

Default value : 0.

7.5.9 Parameter FSEC

This parameter compensates for static friction. It must be adjusted after FVISC has been determined.

It is without effect when the reference speed is zero and it produces a constant feed forward torque, positive or negative, when the reference speed is not zero.

FSEC should be a positive value between 0 and 32,767 (hexa : 0 ... 7FFF).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 85 (hexa : 55).

Default value : 0.

7.5.10 Parameter FEXT

This parameter allows for the compensation of loads that are overhauling or not counterbalanced. It allows for a reduction of the position error when such a torque is applied to the motor such as in the load of a vertical axis. FEXT can also be continuously readjusted by a "host" command when a serial connection exists.

FEXT can have any positive or negative value, except 32,768 (hexa : 8000). A value FEXT = 128 (hexa : 80) compensates for a torque of 1 Nm.

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 86 (hexa : 56).

Default value : 0.

7.5.11 Parameter DPOMA

This parameter defines the linearity limit for the position lag. When the value DPOS is less than DPOMA, the torque produced by the controller is a function of the lag. Beyond, it remains constant.

The value of DPOMA is given by :

$$\text{DPOMA} = 2^{16} \cdot \frac{lag_{\max}[rad]}{2 \cdot \pi} = 2^{16} \frac{lag_{\max}[\deg]}{360}$$

When the position controller is disabled by KTEGR = 32,768 (hexa : 8000), DPOMA is used as limit for the integrator of the speed controller.

The value of DPOMA is then given by :

$$\text{DPOMA} = \frac{2^{20}}{3000} \cdot \frac{\max.speed\ lag\ [rad/s]}{2 \cdot \pi}$$

$$= \frac{2^{20}}{3000} \cdot \frac{\max.speed\ lag\,[\deg/s]}{2 \cdot \pi}$$

DPOMA should be positive between 1 and 32,767 (hexa : 1 ... 7FFF).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 87 (hexa : 57).

Default value : 4,096 (hexa : 1000).

This value corresponds to a lag of 0.39 [rad] or 0.0625 [rev], respectively 73.6 [rad/s] or 703 [rpm].

7.5.12 Parameter SEUIL1

This parameter determines the warning level for excessive lag (position-control mode) or excessive speed (speed-control mode) : KTEGR = 32,768 (hexa : 8000). See paragraphs 6.2.1 and 6.2.4 for descriptions of bit 4 of STATA and bit 5 of STATD.

The value of SEUIL1 is expressed in the same way as DPOMA (described above).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 88 (hexa : 58).

Default value : 2,048 (hexa : 800).

This value corresponds to a lag of 0.20 [rad] or 0.0313 [rev], respectively 36.8 [rad/s] or 351 [rpm].

7.5.13 Parameter COUMA

This parameter determines the maximum torque output of the drive system. The value may be set based on mechanical limitations, to prevent demagnetization of a synchronous motor, or to the maximum current which the ST1 can produce.

The value of COUMA is given by :

$$COUMA = \text{max. torque [Nm]} \cdot 128$$

COUMA shall be positive between 1 and 32,767 (hexa : 1 ... 7FFF).

Please refer to application note AN 5 : "The speed and position controlling regulator" No. 024.8042 for the adjustment of this parameter.

Address : 89 (hexa : 59).

Default value : 512 (hexa : 800).

This value corresponds to a max. torque of 4 Nm.

Note : The default value of previous versions was 8,192 (hexa : 2000).

7.5.14 Parameter COULIM

This parameter determines the level of the 2nd torque limit. This limit is activated by the instruction "LIMTOR".

The value is expressed in the same way as COUMA.

Address : 90 (hexa : 5A).

Default value : 0.

7.6 STATUS MASKS CMASKU, CMASKA AND CMASKS

These parameters are explained in paragraph 6.3.

7.7 PARAMETERS FOR THE TEST DEVICE

7.7.1 General

The functions of the test device are described in paragraph 4. This card allows measuring of variables in digital (one byte of 8 bits) or analog form (two outputs of ± 10 V). The choice of variables as well as the scaling is determined by the parameters described below. The default condition of the software measuring these variables is OFF. It is changed to ON by the parameter CMPBIN. It is recommended to deactivate the software (that is, set CMPBIN = 0) after taking the measurements, to avoid overloading the processor.

7.7.2 Parameters CADBIN and CMPBIN

The parameter CADBIN defines the address of the variable to be monitored digitally. Pins No. 0 to 7 of the device are for the eight bits of the chosen byte (pin No. 0 = lsb).

The name and address of each variable are given in the listing which describes each software version. See also paragraph 9. The address extends from 0 to 255 (hexa : 0 ... FF).

The parameter CMPBIN specifies if the byte monitored is the most significant or the least significant of the selected variable. In addition, a low level (0) completely deactivates the test software and the analog values are no longer available.

CMPBIN = 0	deactivated
CMPBIN = 16 (hexa : 10) :	most significant byte (bits 8 ... 15)
CMPBIN = 4,096 (hexa : 1000) :	least significant byte (bits 0 ... 7)

	Address	Default value
CADBIN	128 (hexa : 80)	128 (hexa : 80)
CMPBIN	129 (hexa : 81)	0 (for byte STATA).

7.7.3 Parameters CADMA1 and CMPMA1

For enabling the test device see paragraph 7.7.1

The parameter CADMA1 defines the address of the analog variable to be monitored on pin A1 of the test device. The address is determined in the same way as the parameter CADBIN. In the case of variables with double or triple precision, only access to one (1) word is possible.

The parameter CMPMA1 defines a scale factor for the A1 analog output. The converter resolution is 8 bits corresponding to a resolution of 78.1 mV. The following table shows the value of CMPMA1 in accordance with the desired full scale and resolution.

- 10 V ... + 10 V	Resolution	CMPMA1
- 32,767 ... + 32,767 (hexa : 8001 ... 7FFF)	256 (hexa : 100)	16 (hexa : 10)
- 2,047 ... + 2,047 (hexa : F801 ... 7FF)	16 (hexa : 10)	256 (hexa : 100)
- 127 ... + 127 (hexa : FF81 ... 7F)	1 (hexa : 1)	4,096 (hexa : 1000)

Intermediate values are allowed. The following relation binds the chosen variable, the parameter CMPMA1 and the output voltage :

$$U_{OUT} \text{ [V]} = \text{value [lsb]} \cdot \text{CMPMA1} \cdot 0.078 \text{ [V]} / 4{,}096$$

	Address		Default value	
CADMA1	130	(hexa : 82)	53	(hexa : 35)
CMPMA1	131	(hexa : 83)	256	(hexa : 100)

These values correspond to the position lag DPOS, where + 10 V represents a deviation of 1/32 of a rotation (11.25°), with a resolution of 1/4096 of a revolution (5.3').

The scale factors could be set according to the following table.

speed reference **VIREFB** address: 47(>2F)		Position lag **DPOS** address: 53 (>35)		torque **COCOU** address: 55 (>37)	
CMPMAi	[t/min/V]	**CMPMAi**	[deg/V]	**CMPMAi**	[Nm/V]
72 (>48)	2000	144 (>90)	2	82 (>52)	5
144 (>90)	1000	288 (>120)	1	205 (> CD)	2
288 (>120)	500	576 (>24)	0.5	410 (>19A)	1
722 (>2D2)	200	1444 (>5A4)	0.2	820 (>334)	0.5
1444 (>5A4)	100	2888 (>B48)	0.1	1444 (>5A4)	0.2

7.7.4 Parameters CADMA2 and CMPMA2

These parameters define the second variable measured in analog form on pin A2 of the test device. They are determined in the same way as the parameters CADMA1 and CMPMA1.

	Address		Default value	
CADMA2	132	(hexa : 84)	41	(hexa : 29)
CMPMA2	133	(hexa : 85)	2,048	(hexa : 800)

These values correspond to the actual speed VIRB, where + 10 V represents a speed of approx. 700 rpm with a resolution of 2.7 rpm.

7.8 ABSOLUTE POSITION LIMITS

7.8.1 General

The two parameters described below define the absolute position limit values. When one of these values is exceeded, the power stage is disabled. See paragraph 6.2.4, bit 4 of STATD.

7.8.2 Parameter PHIL1

This parameter determines the "left" limit for the absolute position. A position lower than the value of PHIL1 is considered as an error, except if parameter PHIL2 equals 32,768 (hexa : 8000).

The value of PHIL1 may be positive or negative. A value PHIL1 = 16 (hexa : 10) corresponds to one revolution.

Address : 134 (hexa : 86).

Default value : 32,768 (hexa : 8000).

7.8.3 Parameter PHIL2

This parameter determines the "right" limit for the absolute position. A position higher than the value of PHIL2 is considered as an error. Setting parameter PHIL2 equals 32,768 (hexa : 8000), disables both overtravel limits for continuous systems with no position restriction.

The value of PHIL2 may be positive or negative, however, it is necessary to verify that PHIL2 > PHIL1. A value PHIL2 = 16 (hexa : 10) corresponds to 1 shaft revolution.

Address : 135 (hexa : 87).

Default value : 32,768 (hexa : 8000).

This value disables both overtravel limits PHIL1 and PHIL2.

8. REFERENCE GENERATOR

8.1 GENERAL

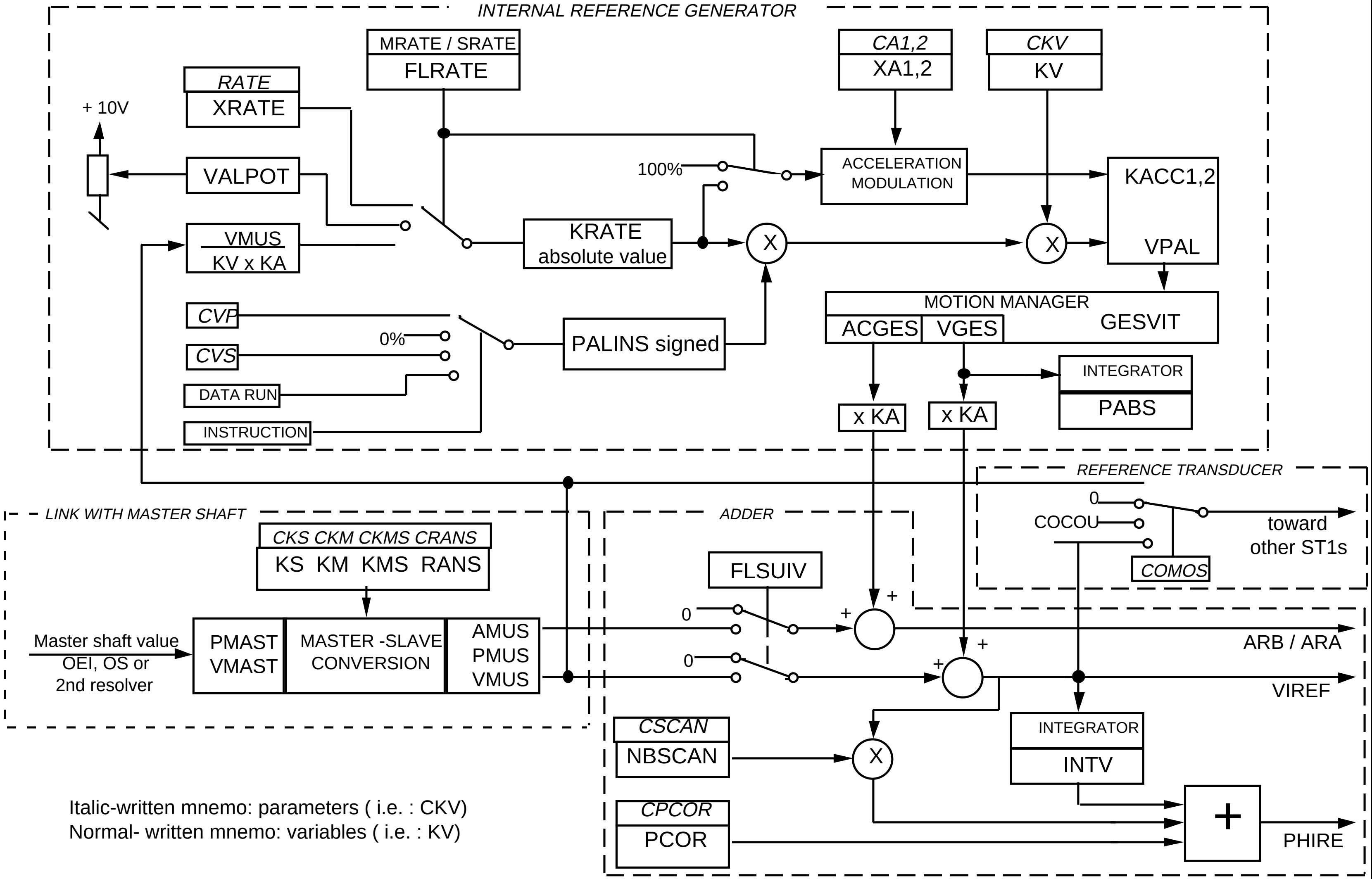
The general block diagram (Figure 8.1) serves as the basis for the present section. Figure 8.2 is a reduced format.

The information contained in this paragraph is general information covering all software versions. For reasons of performance and speed, the various software versions contain only those functions that are required for the particular application.

The term "host computer" employed hereafter refers to equipment external to the ST1 and used for commanding the motion controller. The term "sequencer" refers to a software function performed by the LIO board (if installed) - this board can be used with, or in place of, the host. Each instruction is executed by the ST1 within a particular sampling cycle ("TU").

8.2 PRINCIPLE

The motor can be slaved to various references provided by the internal reference generator or a "master shaft". These two references are switched in turn, or summed to make a single reference.



Italic-written mnemo: parameters (i.e. : CKV)
Normal- written mnemo: variables (i.e. : KV)

Figure 8.1 : *Reference computer*

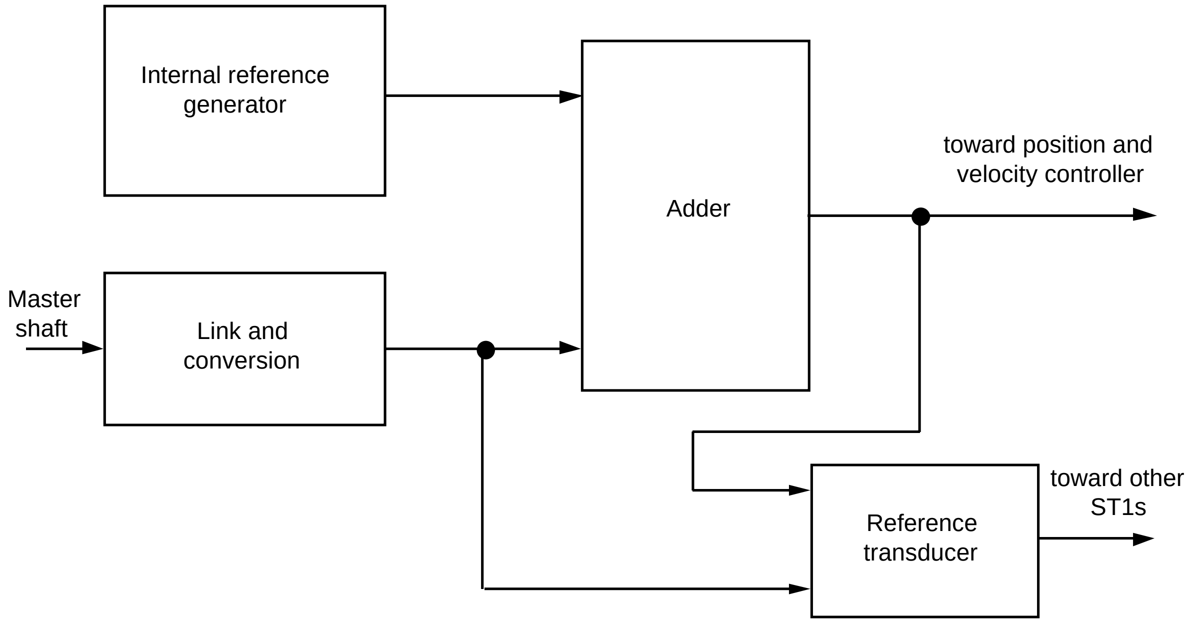


Figure 8.2 : *General Block Diagram (reduced format)*

The **INTERNAL REFERENCE GENERATOR** (abbreviated hereafter to "IRG") transforms instructions from the host or the sequencer into a string of speed and position commands; these are updated at each sampling cycle (TU). The generator is described in paragraph 8.4 below.

The link with the **MASTER SHAFT** supplies speed and position commands updated at each sampling period. These commands undergo unit conversion. The link is described in paragraph 8.5.

The **ADDER** combines the above two commands according to set modes contained in the host and sequencer instructions. This process is described in paragraph 8.6. This summation results in :

- a reference speed (variable "VIREF"),
- a corresponding reference position (variable "PHIRE"),
- a reference acceleration (variable "AR").

The above three components are sent to the position/speed controller, in the same units as those taken at the motor by the resolver. The position measurement is an absolute value within the given period.

The **REFERENCE TRANSDUCER** (see paragraph 8.11) feeds the references to the other axes.

8.3 PRINCIPLE

Time base

The ST1 sampling period is also used as the basic time unit (TU); it is set by a crystal with an accuracy of ± 100 ppm. For most software versions, this period equates to 1/3,000 sec. = 333.3 µs, or a frequency of 3 kHz ± 0.3 Hz.

Motor position

A one-speed resolver is normally employed, that is, having a single pole-pair. Thus the resolver period corresponds to one actual rotation of the motor. Hence the resolver unit (RU) is expressed as :

$$1 \text{ RU} = 2^{-32} \text{ motor revs.}$$

If a higher number of poles is used, the motor rotation period is reduced accordingly.

Example : Resolver with four pole-pairs : 1 RU = 2^{-34} motor revs.

Motor speed

The motor speed measurement is based on the two units defined above. For a resolver having a single pole-pair, the speed unit (RU/TU) is therefore expressed as :

$$1 \text{ RU/TU} = 2^{-32} \text{ motor revs} \cdot 3{,}000 \text{ rps} = 4.39 \cdot 10^{-6} \text{ rad/s.}$$

The effect of crystal precision on speed is described in paragraph 8.10 within the context of synchronization precision.

Motor acceleration

Acceleration measurement is obtained from the position and time units. The acceleration unit (RU/TU^2) is given by :

$$1 \text{ RU/TU}^2 = 2^{-32} \cdot 3{,}000^2 \text{ rp/s}^2 = 13.2 \cdot 10^{-3} \text{ rad/s}^2.$$

8.4 INTERNAL REFERENCE GENERATOR

The internal reference generator has two distinct functions :

- control and monitor machine movements,
- speed- and acceleration-changing (this function prepares data for the monitoring function).

8.4.1 Control and Monitoring of Machine Movements

The basic function of the internal reference generator consists in altering a continuous speed reference over successive time units, while incorporating any instructions received in the meantime, or any variables changing during the movement. This speed reference is the variable VGES. The corresponding acceleration reference is the variable ACGES.

The function applies in two ways :

- motion executed in terms of speed, that is, commanded by instructions of the "RUN" type (such as ERUN, SRUN etc.);
- motion executed in terms of position, that is, commanded by instructions of the "MOV" or "POS" types (such as EMOV, RMOV, SPOS etc.).

Motion executed in terms of speed (referred to hereafter as "RUN motion") :

The speed changes from one value to another (conditioned by the acceleration and deceleration ramps), WITH NO ALLOWANCE for positioning, either before or after the speed change.

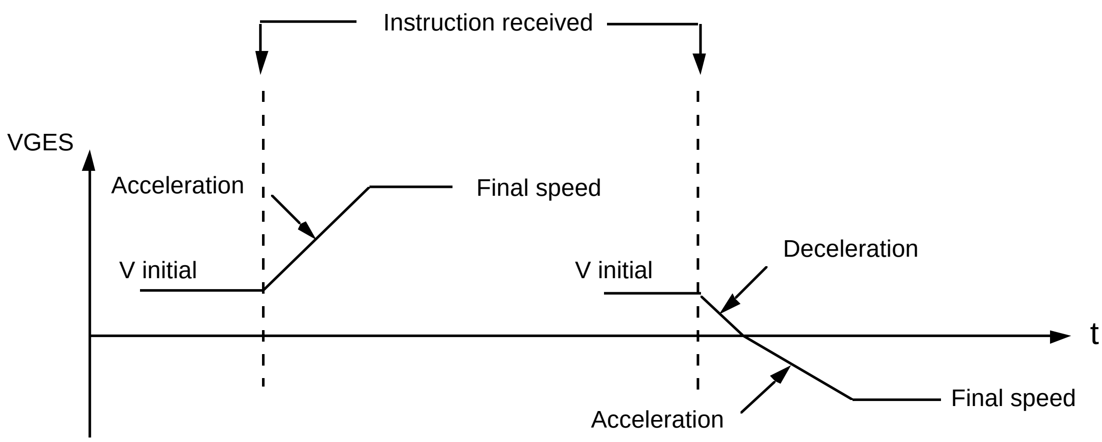


Figure 8.3

Data :

- Initial VGES value; current speed when instruction received;
- VPAL (the terminal speed);
- KACC1, KACC2 (acceleration values).

Results :

- VGES, ACCGES.

Motion executed in terms of position (referred to hereafter as "MOV/POS motion") :

The speed changes from an initial value to a NULL value, transiting through an intermediate speed but with integral speed control (thus motion occurs during the change).

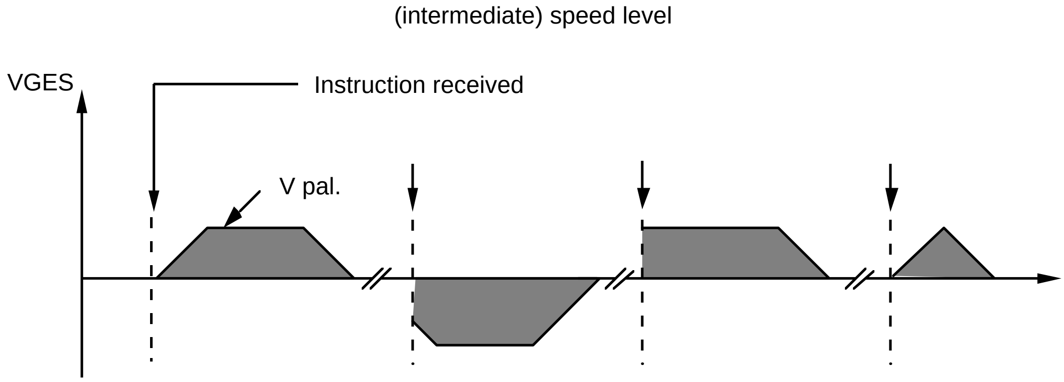


Figure 8.4

Data :

- Initial VGES value; current speed when instruction received;
- VPAL (the terminal speed, or travel speed);
- KACC1, KACC2 (acceleration values) :
- PGES (distance to travel).

Results :

- VGES, ACCGES.

8.4.2 Units

The internal units described above give excellent resolution for both speeds and accelerations. They require the processing of a large number of data bits to cover the limit values, but this is not problematic for the ST1's microprocessor.

In order to process a lower number of bits, thus increase processing speed, the internal reference generator works with its own system of units. Its position unit is the Generator Unit (GU). It employs the basic time unit (TU).

GU's are converted into TU's using the parameters CKA and CKAINV, which are binary coefficients. These coefficients also define the acceleration scale. The parameter CKV defines the speed scale.

8.4.3 Parameters CKA and CKAINV

The parameter CKA sets the scale factor for the acceleration. The choice is limited to the values in the table below.

The parameter also allows the definition of the positive rotation direction of the motor. When this becomes apparent from the resolver, the positive direction corresponds to counterclockwise rotation of the motor shaft ("mathematical" definition).

The parameter CKAINV is only an auxiliary parameter. It is also determined with the help of the following table.

Acceleration	[rad/s^2]	CKA		CKAINV
Full scale	Resolution	Positive direction	Negative direction	
1,710	0.0527	4	- 4 (h : FFFC)	16,384 (h : 4000)
3,420	0.1053	8	- 8 (h : FFF8)	8,192 (h : 2000)
6,850	*0.2107*	*16 (h : 10)*	*- 16 (h : FFF0)*	*4,096 (h : 1000)*
13,700	0.4212	32 (h : 20)	- 32 (h : FFE0)	2,048 (h : 800)
27,400	0.843	64 (h : 40)	- 64 (h : FFC0)	1,024 (h : 400)
54,800	1.685	128 (h : 80)	- 128 (h : FF80)	512 (h : 200)
109,000	3.37	256 (h : 100)	- 256 (h : FF00)	256 (h : 100)

Comments :

- The acceleration values indicated are valid for a resolver with one pole pair ("one-speed"). For other models, the desired acceleration should be multiplied by the number of pole pairs before consulting the table.

- The parameters CKA and CKAINV are only taken into consideration during setup of the initial system parameters.

	Address	Default value
CKA	144 (hexa : 90)	16 (hexa : 10)
CKAINV	145 (hexa : 91)	4,096 (hexa : 1000)

8.4.4 Parameter CKV

Having determined the parameter CKA, the scale for the speed is set with the parameter CKV. After defining (in rad/s) the speed of the motor to the full scale (100 %), and CKA being known, CKV is obtained by :

$$\text{CKV} = \frac{\omega_{PE}[rad/s]\cdot 2^{28}}{|CKA|\cdot 32767\cdot 375\cdot\pi} = \frac{6.95\cdot\omega_{PE}[rad/s]}{|CKA|}$$

Comments :

- CKV should be a positive integer.

- The product of CKV · |CKA| should be less than 16,384, which corresponds to a maximum scale of 22,500 rpm.

- The parameter CKV is only taken into consideration in the initialization of the microprocessor.

Address : 146 (hexa : 92).

Default value : 273 (hexa : 111).

When CKA = 16 (hexa : 10) or - 16 (hexa : FFF0) this default value corresponds to a full scale of 618 rad/s or 6,000 rpm.

8.4.5 Speed and Acceleration Modulation

Speed :

Variable VPAL serves as the terminal speed for RUN motion, or as the intermediate speed for MOV/POS motion. Its value is defined by variable KV, copied from parameter CKV. VPAL can be reduced by two other variables : KRATE and PALINS :

$$VPAL = KV \cdot PALINS \cdot KRATE$$

Variable KRATE depends on the operating mode. It acts as a general coefficient on speeds and sometimes, accelerations. This coefficient can be either parameterized, or set through a potentiometer, or proportional to the speed of the master shaft. This operating mode is defined by the parameters MRATE and SRATE.

Variable PALINS depends on the current instruction (that is, RUN, MOV, STOP, synchronizing engaged/disengaged etc.).

Instruction Type :	PALINS contains :
MOV type and (P)ERPOS	Parameter CVP
STOP	0%
(P)ESPOS	Parameter CVS
RUN type	Data content of RUN instruction

Acceleration :

Variables KACC1 and KACC2 respectively define accelerations and decelerations. They are copied from parameters CA1 and CA2.

Depending on parameter MRATE, the above values also can be modulated by the variable KRATE. In this case, the modulation level is a function of the square of KRATE in order to get the acceleration acting (cam simulation) :

$$KACC1,2 = CA1,2 \cdot KRATE^2$$

Note :

The effect of this modulation feature is limited by the effect of digitizing. If KRATE is deduced from the master shaft and the shaft position is read by an OEI system or a second resolver, resolution will be low and hunting will be high. These measurement problems will be amplified (squared) and the acceleration component will be unusable in certain applications.

8.4.6 Parameter CVP

For MOV or RPOS type motions, this parameter is copied into the variable PALINS, where it determines the absolute value of the travel speed (by applying a percentage of parameter CKV). Parameter CVP must be positive, between 1 and 32,767 (hexa : 1 ... 7FFF). The latter value is the total speed-scale reading (= 100%). The speed hold is determined for both directions of rotation.

Address : 162 (hexa : A2).

Default value : 5,400 (hexa : 1518).

The default value is 16.5% of the total scale reading (1,000 rpm for default values at CKA and CKV).

8.4.7 Parameter CVS

This parameter has the same function and must be calculated as parameter CVP (see above). However, it is active (and thus transferred into variable PALINS) in case of motions requested by instructions ERPOS or PRPOS.

Address : 150 (hexa : 96)

Default value : 32,767 (hexa : 7FFF),
thus 100% of the total scale reading

8.4.8 Parameters CA1 and CA2

These parameters indicate the acceleration desired for each displacement and for each speed transition.

Acceleration (CA1) and deceleration (CA2) are specified separately. The parameter CA2 should be positive, because it acts on the absolute value of the deceleration for both directions.

The terms "acceleration" and "deceleration" actually refer to the sign of the speed transition at the internal reference-speed generator.

When selecting CA1 or CA2, no allowance is made for summated speed of the master shaft (if this factor is present). Details concerning selection of CA1 or CA2 during synchronizing are given in paragraph 8.7.

A value of 32,512 (hexa : 7F00) corresponds to the full acceleration-scale reading (=100 %) determined by parameter CKA.

Limit : To speed up the calculation process, the software processes any deceleration event not exceeding 10.9 seconds' duration. Program CA2 for this limit (which applies to position control only).

	Address	Default value
CA1	166 (hexa : A6)	10,000 (hexa : 2710)
CA2	169 (hexa : A9)	10,000 (hexa : 2710),

thus 30.7 % full scale, or 2,090 rad/s^2 if CKA = 16 (hexa : 10).

8.4.9 Parameter ASTOP

If an acceleration value is contained in a "STOP" command, this value may be specified separately from the normal accelerations by means of the parameter ASTOP. The latter can take any positive value between 1 and 32,767 (hexa : 1 ... 7FFF). The value 32,767 (hexa : 7FFF) corresponds to 100.8 % of the full scale as determined by parameter CKA.

If ASTOP is introduced as = 32,768 (hexa : 8000), or zero, this parameter is ignored. In this case, the acceleration value for this STOP is as contained in variable KACC2.

Address : 155 (hexa : 9B).

Default values : 32,768 (hexa : 8000).

(The latter value disables parameter ASTOP).

8.4.10 Parameters MRATE and SRATE

As discussed in paragraph 8.4.5 above, the variable KRATE acts on speed and acceleration values. The variable FLRATE determines which component is copied into KRATE; it therefore determines the effect of KRATE on the acceleration. At system start-up, or during instructions containing no synchronizing component, the parameter MRATE is copied into variable FLRATE. If the instruction does contain a synchronizing component, the parameter SRATE is copied into variable FLRATE.

FLRATE		Component copied	Modulation
0	(hexa : 0)	Parameter RATE	active
1	(hexa : 1)	Potentiometer setting	active
2	(hexa : 2)	VMUS/KA · KV	active
32,768	(hexa : 8000)	Parameter RATE	inactive
32,769	(hexa : 8001)	Potentiometer setting	inactive
32,770	(hexa : 8002)	VMUS/KA · KV	inactive

It is not allowed for MRATE or SRATE to have values other than those given in the above table.

	Address	Default value
MRATE	172 (hexa : AC)	32,768 (hexa : 8000).
SRATE	173 (hexa : AD)	32,770 (hexa : 8002).

8.4.11 Parameter RATE

RATE can be used in the same way as KRATE. It must be positive, with a value between 1 and 4096 (hexa : 1 ... 1000). The latter value corresponds to a multiplier of 100%.

Address : 170 (hexa : AA).

Default values : 4,096 (hexa : 1000) (= 100%).

8.4.12 Program Position Units

The position unit used by the host processor or the sequencer may differ from the internal unit described in paragraph 8.3. In this case a "length unit" (LU) must be determined, and used throughout the application or function.

Example : 1 LU = 1 µm, 10 µm, 0.1 mil, 0.1° etc.

Standard software versions can process positions expressed in LU in three-byte format (23 bits + sign). This corresponds to values between - 8,388,607 and + 8,388,607 [LU]. Conversion of LU's into GU (generator units) employs parameters CKR and CKH.

8.4.13 Parameters CKR and CKH

The values of these two parameters depends on the following factors :

- The position unit (LU) selected.
- The gearing ratio planned.
- The CKA and CKAINV factors determined beforehand.
- The number of pole pairs in the resolver.

It is recommended to proceed as follows :

a) Determine the exact relation between the deviation corresponding to one turn of the motor and the selected deviation unit LU. N1 and N2 being two positive whole numbers, the result is as follows :

$$N1 \cdot 1\ \text{LU} = N2 \cdot 1\ \text{turn}$$

b) CKR and CKH are then given by :

$$\frac{CKR}{CKH} = \frac{CKAINV \cdot N2}{N1}$$

c) Verify that the result is between 1/1 and 256/1. Verify also that both the numerator and denominator are between 1 and 32,767.

Comments :

1. If the number of pole pairs in the resolver is greater than 1, divide the result obtained under b) by that number.

2. If the conditions in c) are not fulfilled, consider one of the following actions and then redo the calculations from the beginning :

   - Modify the ratio of the gearing (if still feasible).

   - Select another length unit LU (for example : 1' instead of 0.1°).

   - Select other values for CKA, CKAINV and CKV even if the resolution on the acceleration becomes worse.

3. The number ¹ may be replaced by the fractions 22/7 (error 403 ppm, in excess) or 355/113 (error 0.085 ppm in excess).

	Address	Default value
CKR	147 (hexa : 93)	4,096 (hexa : 1000)
CKH	148 (hexa : 94)	4,096 (hexa : 1000)

When CKA = 16 (hexa : 10) or -16 (hexa : FFFO), these values correspond to the definition : 4,096 LU = 1 turn.

8.5 LINK WITH MASTER SHAFT

8.5.1 Principle

An ST1 motion controller can be connected to, hence receive commands from, a master shaft. Three types of connection are possible :

- "OEI connection" : the master shaft is fitted with an incremental encoder, which communicates with the ST1 through an optional OEI board.

- "Second resolver connection" : the shaft is fitted with a resolver connected to the second resolver input of the ST1.

- "OM/OS connection" : the master shaft is controlled by a second ST1 fitted with an OM board. The slave ST1 is fitted with an OS board; references are routed between the boards by an fiber-optic link.

References generated by the master shaft are read and stored in variables named with mnemonics starting with "MAST" ("Master"). They retain the master's system units.

The references are then converted into the unit system of the slave, and stored in variables with mnemonics starting with "MUS" ("Master : Units = Slave"). This conversion is performed using the coefficients CKS, CKM, CKMS and CRANS. These parameters enable converting without cumulative error due to non-integer ratios. Conversion is always applied to the speed data, that is, the variable VMAST is converted, giving the variable VMUS. The new position (variable PMUS) is deduced by summing the various VMUS speed components.

8.5.2 Parameters CKS, CKM, CRANS and CKMS

These parameters allow defining the ratio between the master and the slave shafts, taking into account the respective units of each.

They are deduced from the formula :

$$\frac{CKS}{CKM} \cdot CRANS = \frac{V_{Slave}}{V_{Master}} \cdot RANM$$

The ratio Vslave / Vmaster is the speed ratio to be determined between the master shaft and the slave shaft.

The fraction CKS/CKM allows defining ratios containing several prime numbers such as found in reduction gears : using decimal or binary numbers, such values could not be expressed with absolute precision.

CRANS is employed as a scale coefficient.

The ratio CKS/CKM must always be 1 or less. The nearer it is to 1, the better the conversion resolution. The parameter CRANS is able to meet this criterion, since it can have any value between 1 and 32,767.

It is important for the absolute values of CKM and CKS to be as close as possible to 32,767 also, for reasons of resolution. If necessary, increase CKS and CKM by applying the same multiplication factor.

The parameter CKMS is auxiliary since it expresses the same ratio as CKS and CKM :

$$CKMS = \frac{CKS \cdot 32768}{CKM}$$

If the two shafts rotate in different directions, the negative sign of the ratio is applied to the parameters CKS and CKMS. As for parameters CKM and CRANS, they are always positive.

RANM is neither a parameter nor a variable : it is a mnemonic expressing the scale coefficient in the master shaft's units.

In an OM/OS system, the value of RANM is defined by parameter COMOS on the master ST1, as described in paragraph 8.11.2 below.

In the OEI and second resolver systems, RANM always has a value of 256.

A calculation example is given in paragraph 8.9.

Special case :

If the precision obtained with parameters CKS and CKM is not required (that is, a slight cumulative error is admissible), CKS and CKM can be set to zero. The ratio is then defined by the parameters CKMS and CRANS only, according to the following equation :

$$\frac{CKMS}{32768} \cdot CRANS = \frac{V_{Slave}}{V_{Master}} \cdot RANM$$

Thus the ratio can be adjusted while the shaft is rotating, by altering variable KMS only (which is a copy of parameter CKMS).

Notes :

Prior to use, parameters CKS, CKM, CRANS and CKMS must be transferred in the variables associated using the UPGRAD instruction (see paragraph 10.6.6).

In an OM/OS system, if the transmitted reference is a torque value, these four parameters have another function : see paragraph 8.5.9.

	Adresse	Valeur par défaut
CKS	116 (hexa : 74)	32,767 (hexa : 7FFF)
CKM	115 (hexa : 73)	32,767 (hexa : 7FFF)
CRANS	117 (hexa : 75)	1
CKMS	118 (hexa : 76)	32,767 (hexa : 7FFF).

8.5.3 OEI System

The OEI board filters, counts and shapes pulses delivered by an incremental encoder. These pulses deliver a position that is transferred into the variable PMAST. The ST1 uses variations in this position to deduce speed VMAST, then calculates the converted speed (VMUS) using the parameters CKS, CKM, CKMS and CRANS. Position PMUS is obtained by summing the speeds.

Definition :

Incremental encoders are characterized by the number of marks per revolution. These marks may be referred to as "dashes".

The OEI board counts four pulses per mark. The pulses can be referred to as "dots". One dash = four dots.

Referring back to the conversion formulae :

$$\frac{CKS}{CKM} \cdot CRANS = \frac{V_{Slave}}{V_{Master}} \cdot RANM$$

$$CKMS = \frac{CKS \cdot 32768}{CKM}$$

Vslave is expressed as 1/65,566 motor revolutions in the period chosen.

Vmaster is expressed as the number of encoder dots in the same time units.

RANM is fixed at 256 (hexa : 100).
The following parameters depend on the type of sensor used, and the application.

8.5.4 Parameter FILTEI

The OEI board contains a digital filter, for use when the encoder environment contains a high degree of disturbance. The parameter FILTEI gives the maximum frequency generated by the encoder : the filter removes frequencies above this value. The following table gives the value of FILTEI as a function of the maximum speed of the application :

FILTEI	0	1	2	3	4	5	6
Dashes/s	250,000	100,000	27,500	3,500	875	190	0

Address :102 (hexa 66)

Default value : 0.

8.5.5 Parameter ACCLIM

In an OEI system, if the encoder resolution is coarse, the obtained reference will contain noise, especially if the slave is rotating much faster than the master.

It is possible to reduce this noise by limiting the resulting acceleration using the parameter ACCLIM.

Caution : ACCLIM must not limit the real acceleration measurement on the master shaft : this would have the effect of cutting an electric shaft drive.

ACCLIM is calculated by reference to values CA1 and CA2, using the following formula :

$$ACCLIM = \frac{CA1(ouCA2)}{2 \cdot CRANS} \cdot CKA$$

The value of ACCLIM must always be positive, between 0 and 32,767 (hexa : 0 ...7FFF).

Address :174 (hexa : AE)
Default value : 32,767 (hexa : 7FFF).

8.5.6 Parameter PISTEI

As described above, the (converted) position of the master shaft is obtained by summing the converted speeds : PMUS = sum of VMUS. PMUS can be set to zero by the instructions. In the OEI system, it can also be set to zero at each pass through the encoder's "zero" track, if the user so wishes.

Parameter PISTEI indicates this choice :

PISTEI = 1 : "zero" track active

PISTEI = 0 : "zero" track inactive.

If "zero" track is active, bit 2 of STATC goes high every time the "zero" position is reached. This bit can be tested by the sequencer (that is, asked to trigger a sequence).

Address :103 (hexa : 67)

Default value : 0.

8.5.7 Parameter NIMPEI

This parameter indicates the number of dashes per encoder revolution (master-shaft encoder).

The value of NIMPEI must be positive, between 0 and 65535 (hexa : FFFF). If the number of dashes is greater, it is necessary to subtract 32768 (or a multiple of 32768) from the number, to bring it below 65536.

Example :

For an encoder with 100,000 dashes per revolution :

NIMPEI = 100,000 - 32,768 - 32,768 = 34,464.

Address :104 (hexa : 68)
Default value : 2,000 (hexa : 7D0).

8.5.8 OM/OS System

The references supplied by the master shaft are routed through optical fibers, and handled by the (optional) OS board. After shaping, these references are immediately forwarded to any other optional OS boards installed. See also paragraph 8.11 - "Reference Transducer".

If the reference is a speed, it is copied into the variable VMAST and converted into variable VMUS.

The following conversion rules apply :

$$\frac{CKS}{CKM} \cdot CRANS = \frac{V_{Slave}}{V_{Master}} \cdot RANM$$

$$CKMS = \frac{CKS \cdot 32768}{CKM}$$

RANM depends on the COMOS parameter of the master shaft.

The COMOS parameter of the slave shaft in turn depends on the type of references transmitted by the master shaft - see paragraph 8.11.2.

8.5.9 OM/OS Torque System

In this mode, the master ST1, fitted with an OM board, transmits its torque component through the optical fiber. This component is in fact the content of the variable COCOU. The slave ST1 (fitted with an OS board) reads the torque reference, applies any necessary corrections, and transmits it to its own motor; it also cancels its own speed- and position-controller.

To select this mode, the parameter COMOS on the master ST1 must be set to 7, and the corresponding COMOS on the slave ST1 to 37,769 (hexa : 8001).

The slave ST1 can correct the received reference, as follows :

$$Cs = Cm \cdot \frac{CKMS}{4096} + \frac{CRANS}{128}$$

where : Cs = slave torque (in Nm)
Cm = master torque (in Nm)

Example :

If the slave torque has to be the same as the master torque, then :
CKMS = 4,096 CRANS = 0

8.5.10 "Second resolver" Connection

An angular position reference can be generated by a second resolver connected to the 15-pin sub-D connector of the ST1's "R" board. This position is copied into the variable "PMAST". It allows calculation of speed VMAST, which, after conversion, provides the variable VMUS.

The following conversion factors apply :

$$\frac{CKS}{CKM} \cdot CRANS = \frac{V_{Slave}}{V_{Master}} \cdot RANM$$

$$CKMS = \frac{CKS \cdot 32768}{CKM}$$

It should be noted that the factor RANM is forced to a value of 256 (hexa : 100) when the parameters are selected. The new position of PMUS is determined by summing the speeds. Nevertheless, at power-up or RESET, PMUS takes the absolute resolver value, but can be overridden by later resets.

8.6 SUMMATION AND SYNCHRONIZING

8.6.1 Adder and Integrator

Note : The term "internal reference generator" is abbreviated to "IRG".

Instructions can select one of two operating modes :

Desynchronized mode :

- The master shaft is not considered. In this case, the reference speed (variable 'VIREF') is that delivered at the IRG output (in the form of variable VGES), with application of the coefficient KA : VIREF = VGES · KA.

  Similarly, for acceleration : AR = ACGES · KA.

Synchronized mode :

- The master shaft is considered. In this case, the IRG speed is added to the master-shaft speed : VIREF = VGES · KA + VMUS.

  Similarly, for acceleration : AR = ACGES · KA + AMUS.

In either of these operating modes, the reference position (variable PHIRE) is obtained by adding the reference speed (variable VIREF), corrected by two additional elements :

PHIRE = INTV + NBSCAN · VMUS + PCOR.

In the above equation :

- The variable INTV is obtained by continuously adding VIREF. It can be initialised by various instructions (see paragraph 10.6).

- The variable NBSCAN is for compensating the delay in transmitting the master-shaft speed : this delay depends on the transmission method used. It causes a position shift between the master and slave shafts, proportional to the speed.

- The variable NBSCAN (copied from the parameter CSCAN) translates the above-mentioned delay into a round number of sampling periods.
- The variable PCOR (copied from the parameter CPCOR) allows insertion of a position shift between INTV and PHIRE, useful during initial set-up etc.

8.6.2 Parameter CSCAN

This parameter is the multiplier applied to the speed to be monitored (to compensate for transmission lag).

The value of CSCAN is copied from the variable NBSCAN during execution of the UPGRAD instruction; it must be positive, between 0 and 20.

The respective values of CSCAN in the three systems are as follows :

- OM/OS system : CSCAN = 0

In this system, if the speed of the master shaft depends on a second resolver, the slave ST1's must have the same CSCAN values as the master shaft.

- OEI system : CSCAN = 2

- second resolver system.

In this case, CSCAN depends on the software employed for this shaft, and can be determined empirically by measuring the physical error between the shafts. The size of the error for a zero or low speed is compared against the error for a high speed. The error differential is then divided by the difference between both speeds; the result is a time, which represents the transmission lag. This is expressed in the CSCAN parameter (in TU).

Address :119 (hexa : 77)

Default value : 0.

8.6.3 Parameters CPCORC / CPCORB

These parameters determine the position shift between the variables INTV and PHIRE.

This position shift allows for independence between the resolver zero (= PHIRE zero) and the mechanical zero (= INTV zero).

The two parameters are inserted as for any others, or via the instructions ALIPOS and ALISU (q.v.). They are expressed in internal-system units :

1 lsb = 2^{-16} motor revolution

The sum CPCORC / CPCORB can be positive or negative, with a value between 0 and ± 2,147,483,647 (hexa : 0 ... 7FFF,FFFF).

	Address	Default value
CPCORC	106 (hexa : 6A)	0
CPCORB	107 (hexa : 6B)	0

8.7 SWITCHING THE SYNCHRONIZING FUNCTION

8.7.1 Synchronizing

Paragraph 8.6.1 mentioned two operating modes : synchronized and desynchronized, selected by the variable FLSUIV. This variable stores the mode requested in the "synchronize" and "desynchronize" instructions. Synchronizing can be achieved even if the master shaft is rotating. It will be appreciated that if FLSUIV is switched to follow a master shaft that is already rotating, then the speed reference VIREF will undergo a jump.

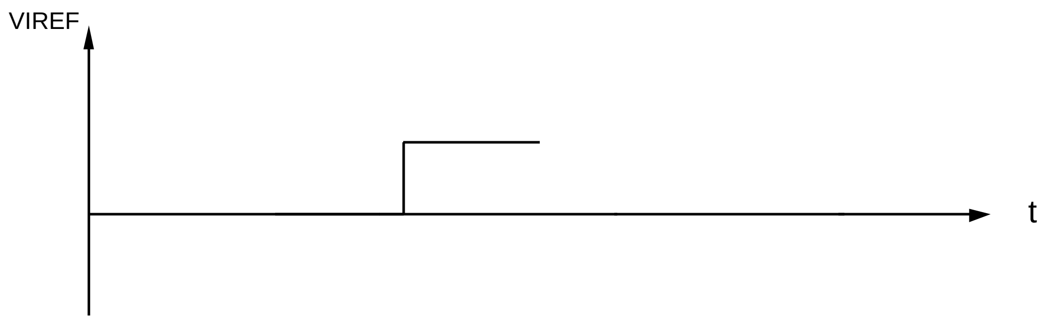


Figure 8.5

This situation is inadmissible since it would cause infinite acceleration and throw out the slaving system. Hence the necessity to "assist" the speed-change by applying a controlled acceleration through the IRG.

Since VIREF = VGES · KA before synchronizing, and VIREF = VMUS + VGES · KA after synchronizing, it is simply necessary to reinitialize VGES to ensure that VIREF remains unchanged at the point of synchronicity.

This is achieved by subtracting the value of VMUS/KA from VGES. Now the sum will be identical at the initial instant. Afterwards the internal reference generator will change VGES to a new value (null or otherwise), respecting the required acceleration values.

Example :

Synchronizing a stopped motor with a rotating master shaft :

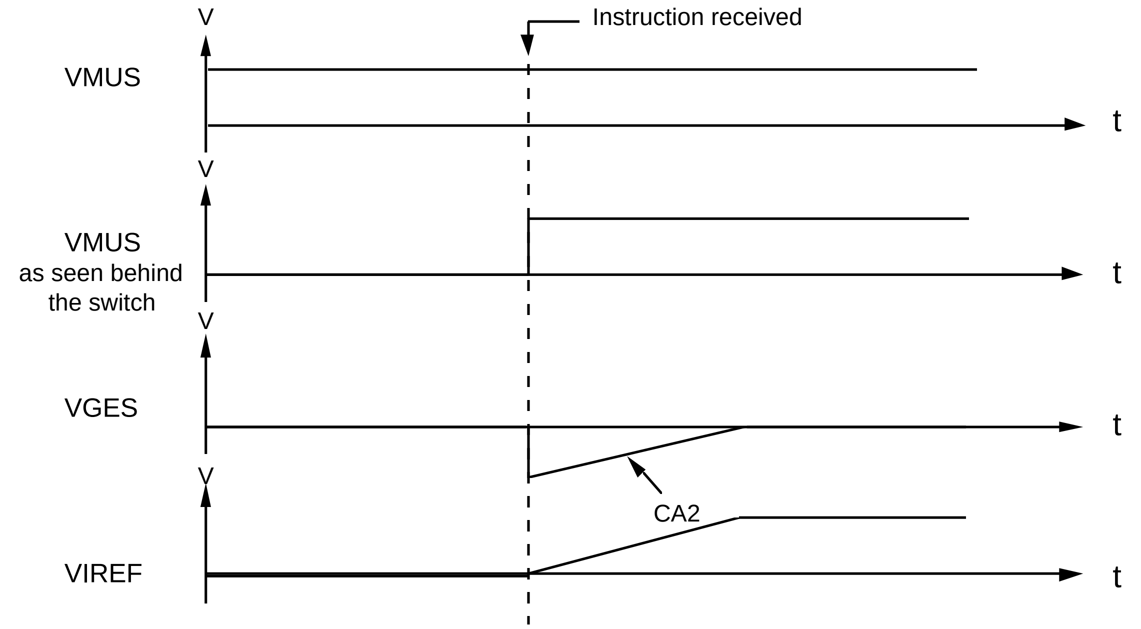


Figure 8.6

The principle allows for a VGES speed other than null before and after synchronizing.

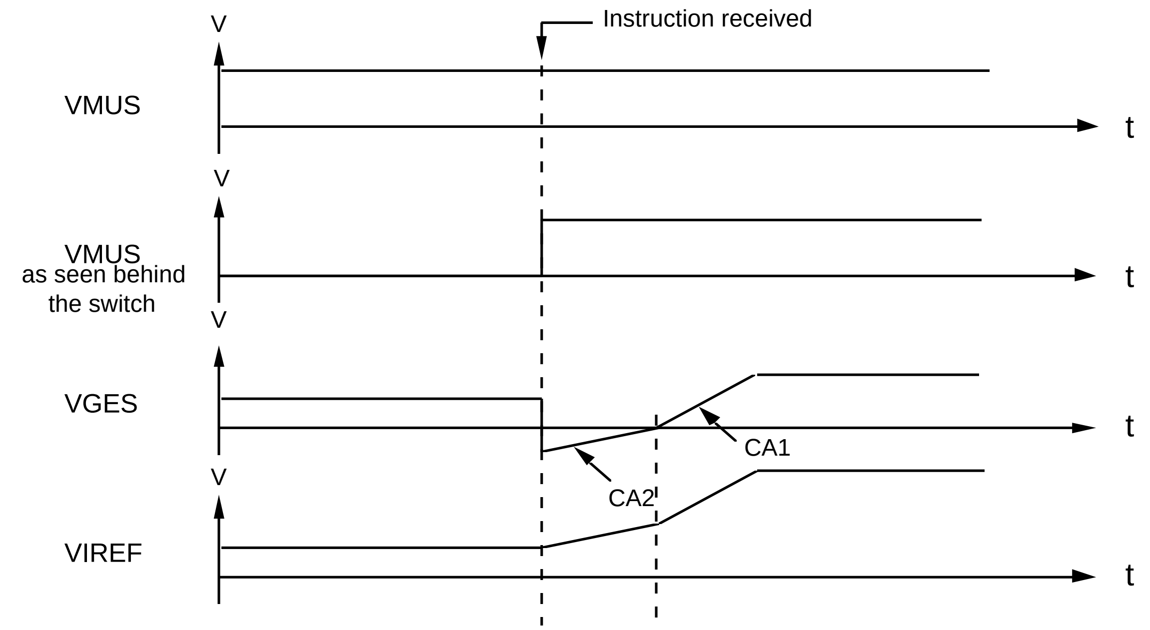


Figure 8.7

8.7.2 Desynchronizing

Desynchronizing relies on a similar principle : VGES is incremented by the value of VMUS/KA, to avoid violent transition on VIREF.

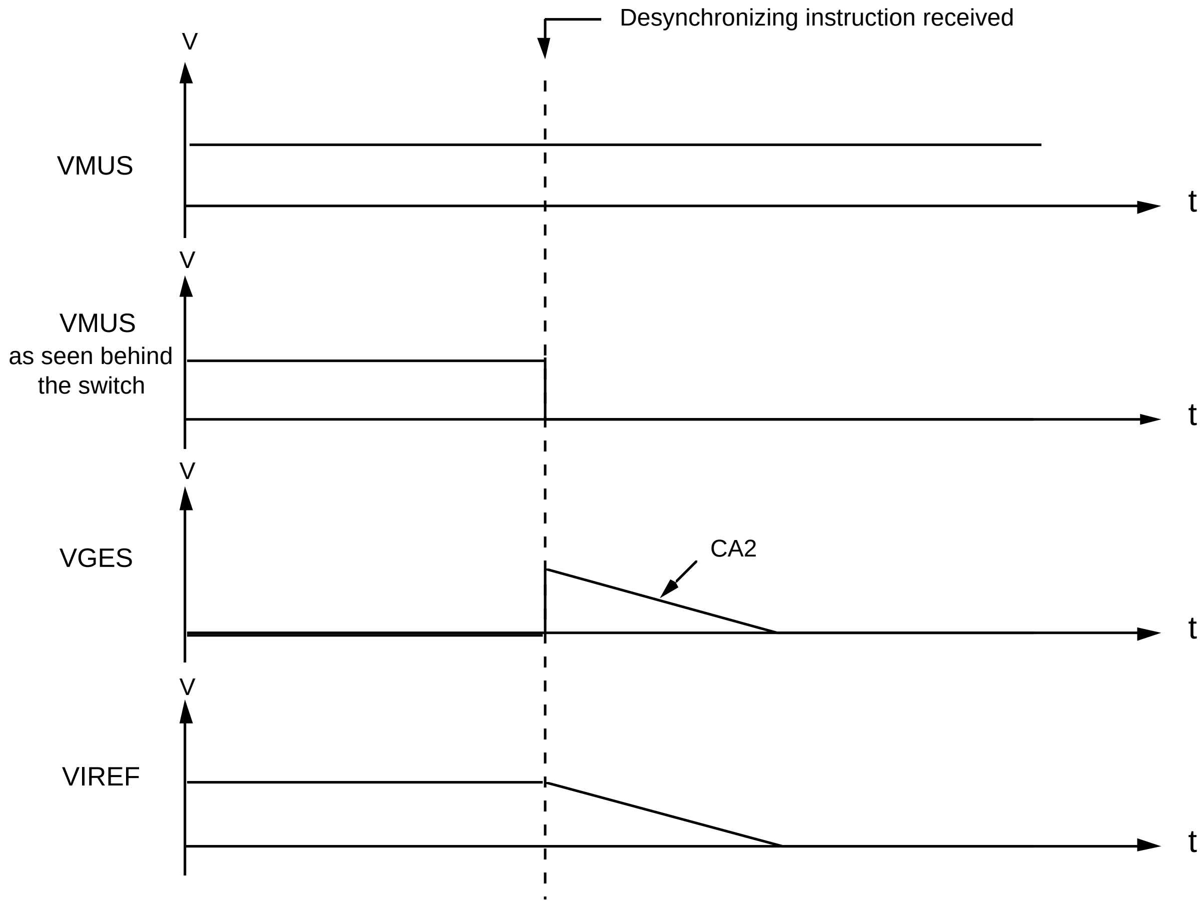


Figure 8.8

As with Figure 8.6, VGES can have a non-null value both before and after desynchronizing; the essential point is that it undergoes a compensatory speed-change to reflect the change in apparent speed VMUS.

8.7.3 Synchronizing and Phase-following

The above discussion concerning synchronizing and desynchronizing considered the factor of speed only. Some instructions allow synchronizing and desynchronizing with emphasis on the respective positions of the master and slave shafts (for example, instruction "ESPOS"). Such instructions are accompanied by a data packet indicating the relative positions of the shafts after synchronizing.

For position-based synchronizing, the software calculates the respective shaft positions before synchronizing, acknowledging the data accompanying the instruction then initializing a movement via the IRG. This movement starts simultaneous with the synchronizing action; when it is over, the shafts are synchronized with the phase commanded by the user.

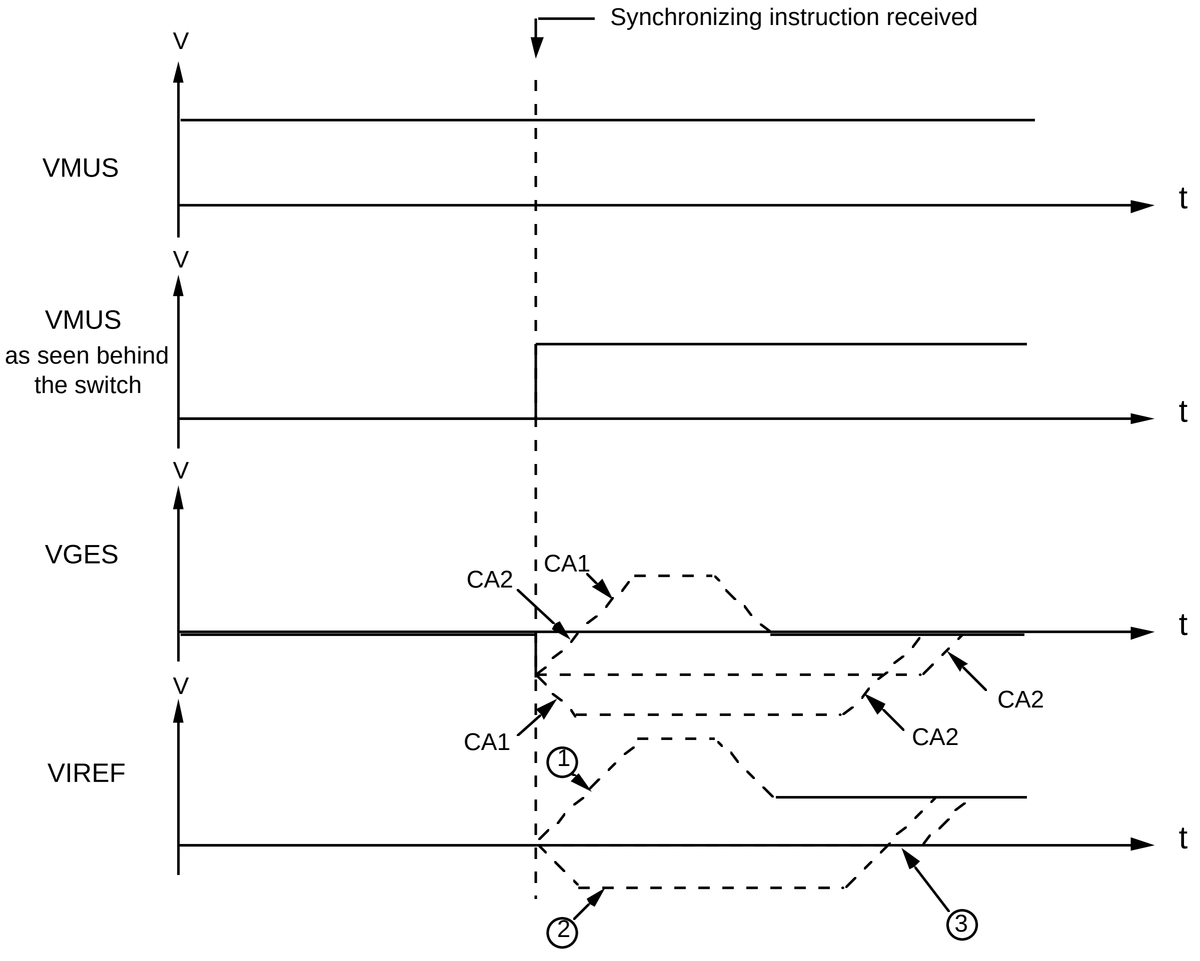


Figure 8.9

Based on the relative shaft positions and the value parametered into VPLA, the slave shaft can act in three ways during a "synchronize" command :

- "Synchronize" command issued "too late" - slave has to rotate faster than master in order to "catch up." This can represent normal operation if the slave is physically capable of supporting the extra speed.

- "Synchronize" command issued "too soon" - slave will meet master.

- Special case where "synchronize" is command issued "too soon" but the IRG's VPAL value is the same as for the master : for a few seconds VIREF will be null due to summing of the two opposed speeds. The slave gives the impression of "waiting" for the master in order to synchronize correctly. To obtain synchronicity, the parameter SRATE takes the master speed (VMUS/KV · KA) as "KRATE".

8.7.4 Desynchronizing with Stop to a Particular Position

When desynchronizing, an additional movement of the IRG can stop the slave to a particular position.

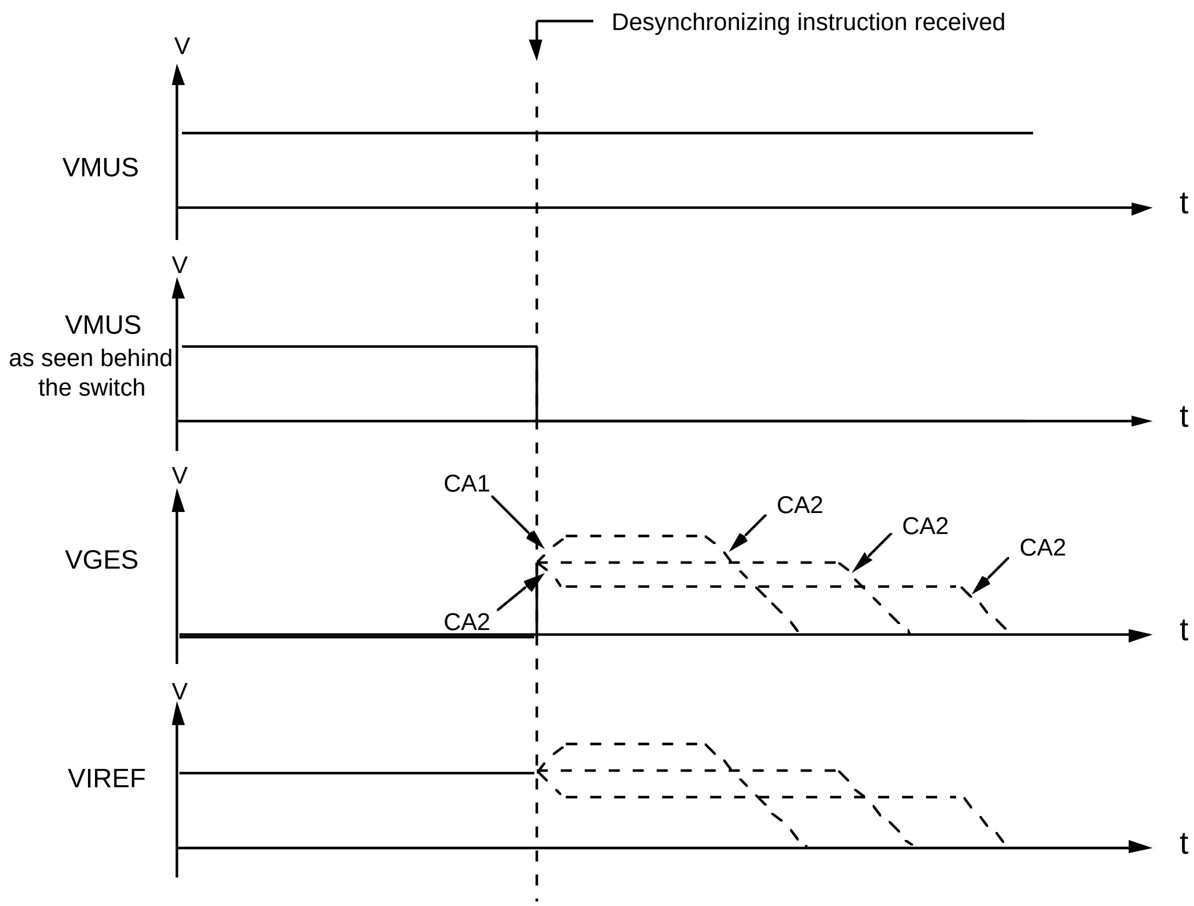


Figure 8.10

8.8 PERIODICITY

8.8.1 Periodic Mode

Motor-driven axes can be classified as belonging in one of two categories

- axes having a limited degree of travel : for example, a machine-tool bed;
- axes having an unlimited degree of travel, in most cases operating in a single direction : for example, printing-press cylinder; conveyor belt; cutting-tool head etc.

The latter types of axes include the notion of "periodicity" : motor revolution, cylinder revolution, space between cars, chain links etc. With these axes, there is no point in synchronizing the slave shaft with the original position of the master shaft : the position of the latter might be far-distant if the slave has been operating out of synch for a long period. Hence synchronizing is performed using the nearest period.

The earlier discussion concerning relative positioning of the master and slave shafts introduced the variable PMUS for the master and the variable INTV for the slave. The notion of periodicity can now be introduced, in the form of two new variables, MOMUS and MOPHI. MOMUS contains an integer value that is the closest expression of the number of counted periods contained in PMUS.

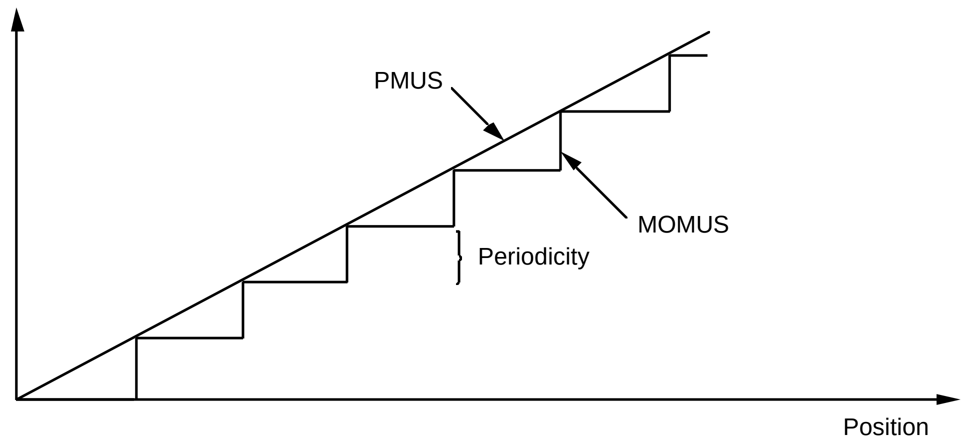


Figure 8.11

More precisely, MOMUS follows PMUS at a distance of one period at the most. A similar arrangement exists between MOPHI and INTV. Thus when synchronizing is required, the difference taken into account is not that between INTV and PMUS, but rather that between (INTV-MOPHI) and (PMUS-MOMUS).

This notion of periodicity can be applied at the master or the slave, or both. The master and slave periods can be equal or different : the choice is made in the "periodicity" parameter groups.
When the system increments MOMUS by one period, bit 2 of STATC is set to "1". This bit can be used by the sequencer to trigger a sequence.

8.8.2 Pseudo-periodic Mode

The variables employed for setting the master-shaft periodicity can be used in another feature known as "pseudo-periodicity".

The master-shaft periodicity mode is disabled with PERIMC = 32,768 (hexa : 8000). When desynchronizing (with ERPOS and ERUN), PMUS is copied into MOMUS. Thus, at the next "synchr*(hexa : 8000,0000) signifies the absence of the periodicity factor.

	Address	Default value
PERISC	111 (hexa : 6F))	) 65,536 (hexa : 1,0000)
PERISB	112 (hexa : 70))	thus 1 motor revolution

8.8.3 Parameters PERIMC / PERIMB

These parameters determine the period of the master shaft analogous to PERISC/B for the slave shaft (see above).

The value of PERIMC / PERIMB must be positive, between 0 and 2,147,483,648 (hexa : 0 ... 8000,0000).

A PERIMC/PERIMB = 2,147,483,648 (hexa : 8000,0000) signifies the absence of periodicity.

	Address	Default value
PERIMC	108 (hexa : 6C))	) 65,536 (hexa : 1,000)
PERIMB	109 (hexa : 6D))	thus 1 motor revolution

8.8.4 Parameter ZMPERS

If the period is a non-binary fraction of a motor revolution, the parameter will be unable to express it exactly and this would lead to cumulative errors in the variable MOPHI. Parameter ZMPERS obviates this problem by ordering MOPHI to reset each time the number of periods equates to a full motor revolution. MOPHIB is compared with a tolerance factor and is reset each time it is within the threshold. ZMPERS expresses this tolerance in the same units as PERIS. One lsb in ZMPERS equals one lsb in PERISB, or 2^{-16} motor revolutions.

Example 1 :

Period = 1/3 motor revolutions. This gives PERISB = 21,845. After three periods, MOPHIB will be at 65,535 (or minus 1). Thus ZMPERS needs to be 2 in order to command a zero-reset at MOPHIB.

Example 2 :

Period = 6/7 motor revolutions. This gives PERISB = 56,173. After seven periods, MOPHIB will be at $6 \cdot 2^{-16}$ revolutions. However : ZMPERS must be less than 9,362 ($\frac{65536}{7}$) otherwise it would already be reset to zero after one period.

The value of ZMPERS must be positive, between 0 and 32,767 (hexa : 0 ... 7FFF).

Address :113 (hexa : 71)

Default value : 256 (hexa : 100)

8.8.5 Parameter ZMPERM

This parameter has the same function as ZMPERS but is attributed to the master shaft. See above.

When ZMPERM = 32,768 (hexa : 8000), the pseudo-periodic mode is selected.

The value of ZMPERM must be positive, between 0 and 32,768 (hexa : 0 ... 8000).

Adress : 110 (hexa : 6E)

Default value : 256 (hexa : 100)

8.9 CALCULATION EXAMPLE

The purpose of this exercise is to calculate the parameters for the following application :

A form-roller bearing an impression is used to print pages rotating on a cylinder. The cylinder is driven by a free motor : this represents the "master shaft". An encoder mounted on the cylinder feeds a reference signal to an ST1, via a multiplier and an OEI board. The ST1 controls the motor driving the form-roller (via a reducing unit with toothed drive belt). The latter assembly represents the "slave shaft".

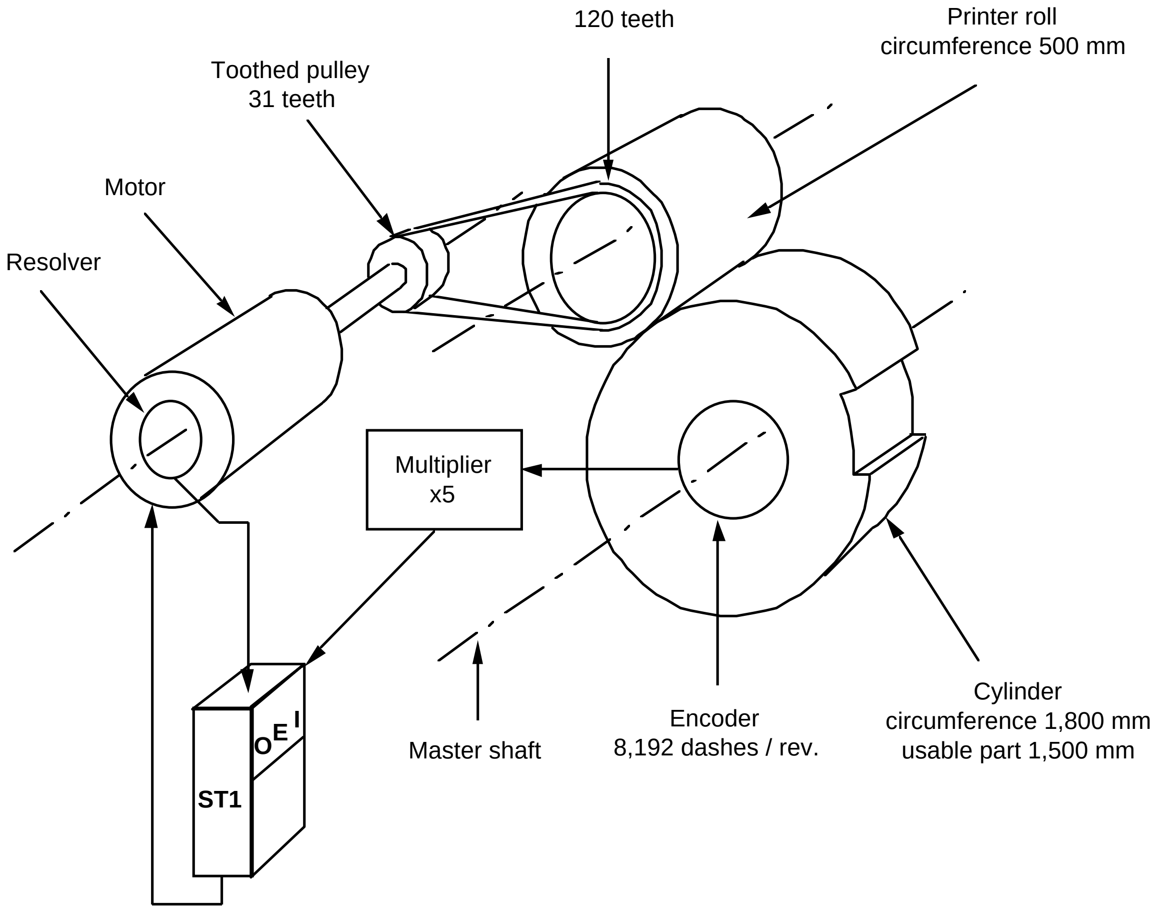


Figure 8.12
Through the action of the multiplier on the OEI board, the ST1 will receive 8,192 · 20 = 163,840 pulses per cylinder rotation. It can be seen from the respective circumferences that the system is synchronized for only three revolutions of the form-roller, after which desynchronizing occurs, with retarding of the slave shaft, then resynchronizing. The periods of the master and slave are not identical.

Calculating the conversion coefficients :

Note : $\frac{CKS}{CKM} \cdot CRANS = \frac{V_{Slave}}{V_{Master}} \cdot RANM$

Expressing speeds in terms of cylinder rotations :

VSLAVE = $\frac{1800}{500} \cdot \frac{120}{31} \cdot 65536$

VMASTER = 8192 · 20

RANM = 256 (OEI),

whence $\frac{CKS}{CKM} \cdot CRANS = \frac{108 \cdot 8}{155} \cdot 256$

By setting CRANS = 8 · 256 = 2,048, the limits of the ratio CKS/CKM are respected.

Now multiply CKS and CKM by 128 to improve the resolution (without exceeding 32,768) :

CKS = 108 · 128 = 13,824

CKM = 155 · 128 = 19,840

CKMS = $\frac{CKS}{CKM} \cdot 32768$ = 22,832.

Calculating the periodicity parameters :

The "slave" period is one rotation of the form-roller expressed in terms of 2^{-16} of a motor revolution :

PERIS = $\frac{120}{31} \cdot 65536$ = 253,687.742

Selecting the higher value of 253,688 (in hexa : 3,DEF8) :

PERISC = 0003 PERISB = DEF8

Check the effect of rounding on the high values : after 31 periods, MOPHI should be expressed as a complete number of motor revolutions (MOPHIB should be 0). Since :

PERIS · 31 = 253,688 · 31 = 7,864,328 (hexa : 78,0008)
then MOPHIC = 0078 MOPHIB = 0008.

MOPHIB should be 0 : the value "8" is due to the effect of rounding. This is corrected by setting ZMPERS = 9. Now the system will automatically reset MOPHIB to zero any time it is below 9, that is, after 31 periods.
The master period is one cylinder rotation expressed as 2^{-16} of a motor revolution.

PERIM = $\frac{18}{5} \cdot \frac{120}{31} \cdot 65536$ or $\frac{432}{31} \cdot 65536$ = 913,275.87

Select the higher value of 913,276 (in hexa : D,EF7C).

PERISC = 0003 PERISB = DEF8

As with the slave shaft, after 31 periods MOMUS should be a complete number of motor revolutions (MOMUSB should be 0). Since :

PERIM · 31 = 28,311,556 (hexa : 1B0,0004).

MOMUSC = 01B0 MOMUSB = 0004.

We can set ZMPERM = 5.

The system will now readjust MOMUS after 31 periods.

8.10 SYNCHRONIZATION PRECISION

Converting the speed of the master shaft into a slave-shaft speed is achieved using the carry method (see paragraph 8.5.2) : VMUS may momentarily be approximate, due to the digitizing process and the complexity of the conversion factors (±1 bit), but there will be no accumulative error.

Sum of VMUS's = constant · sum of VMAST's.

Two axes can therefore be slaved together for several days at a time, without moving by more than one position bit.

During synchronizing or desynchronizing, VMUS and VGES are added. VGES is most often fixed by parameters, and is therefore independent of VMAST. Thus VIREF is neither proportional to nor independent of VMAST.

Special case :

If an IRG RATE dependent on VMAST is selected, VGES will be proportional to VMAST but the conversion carries will not be processed.

Sum of VGES = approx. constant . sum of VMAST.

Similarly, any modulated acceleration values will not be converted exactly : their degree of internal precision will diminish with size. For this reason, if the slave appears to be "waiting" for the master, synchronizing is achieved by setting :

VIREF = C · VMAST - C1 · VMAST

C = approximately C1, whence VIREF = approximately 0.

The above discussion on speed control showed how VGES can be approximate, but that the SUM of the individual VGES values is CONTROLLED, and for this reason the relative positions of the axes after or before synchronizing will be EXACT, without cumulative error.

If the master shaft is an ST1, the fiber-optic transmission system imposes a common time base for both master and slave, hence there will be no relative-positioning error between the two. There is only an absolute error (± 100 ppm) due to a quartz crystal in the transmission system.

Two independent ST1's would operate on such a time-base (error of 100 ppm). If they received the same references at the same time, they would reproduce a maximum error of 0.2 revolutions after 1000 revolutions.

8.11 REFERENCE TRANSDUCER

8.11.1 Principle

One ST1 motion controller can transmit references to other ST1's, and thus becomes the "master shaft" of the latter, as defined above (particularly in paragraph 8.5).

Such a system requires fitting the ST1 with an OM ("optional master") board. The software transmits a given reference to this board in parallel form; the board converts the reference into serial format and transmits it down an optical fiber to the slaves, in the particular format required. An OS ("optional slave") board at each slave transforms the serial code into parallel data, read by a processor in each slave amplifier. Serial data are transmitted at a rate of 3000 values per second. A new reference (24 useful bits) is transmitted in each TU.

The software (V0500 family) configuration allows a given ST1 to act as both slave to a master, and master to other slaves. Nevertheless, since the ST1 can accommodate only one optional board, this configuration is possible only if the first master shaft is fitted with a resolver.

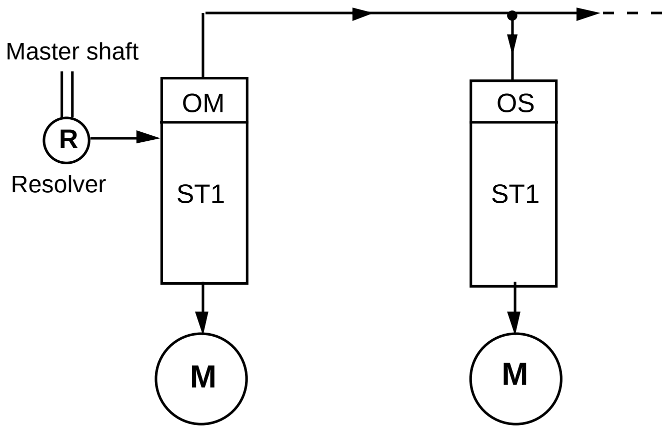


Figure 8.13

The reference transmitted in such a system can be produced by various variables, such as :

VIREF set-value for speed

VMUS translated master speed

VIR actual speed

COCOU torque reference produced at slave output

The selection is made in the parameter COMOS.
Transmission of the actual speed is not recommended, since it contains interference as well as feedback error. In addition, resolution is more coarse than for the reference speed, and is insufficient to allow the slave to deduce an acceleration component for its feed-forward command. Finally, the slaving precision between master and slave is not better. Thus this mode is only justified if the control error on the master shaft can be very great : there will be less control of the slave shaft but the latter will be coupled more closely with the master.

8.11.2 Parameter COMOS

This parameter determines whether the ST1 is master or slave. In either case, it also defines the nature of the reference transmitted, and the scale factor (latter depends on the maximum speeds and accelerations to be transmitted).

A COMOS = 0 indicates no OM or OS option taken up.

A COMOS < 0 (hexa : 8xxx) indicates the OS option.

Table of COMOS values for an ST1 fitted with an OS board

COMOS	Type of reference
32,768 (hexa : 8000)	speed reference
32,769 (hexa : 8001)	torque reference

The following table gives the COMOS values to be inserted in an ST1 employed as a "master shaft" (that is, fitted with an OM board).

RANM is the scale factor; it is used in calculating the conversion coefficients for the slave - see paragraph 8.5.2.

COMOS	RANM	Max.speed [rpm]	Max.accel. [rad/s^2]	Data sent	
1	1	351	430	set value for speed	(VIREF)
2	16	5,620	6,900	set value for speed	(VIREF)
3	256	22,500	110,000	set value for speed	(VIREF)
4	1	351	430	actual speed	(VIRB)
5	16	5,620	6,900	actual speed	(VIRB)
6	256	22,500	110,000	actual speed	(VIRB)
7	-	-	-	torque reference	(COCOU)
8	1	351	430	master speed in slave units	(VMUS)
9	16	5,620	6,900	master speed in slave units	(VMUS)
10	256	22,500	110,000	master speed in slave units	(VMUS)

The parameter COMOS must be transferred inside a variable in order to be used - see instruction UPGRAD, paragraph 10.6.6.

Address :114 (hexa : 72)

Default value : 0.

8.12 POTENTIOMETER INPUT

8.12.1 Principle

When the second resolver option is not used, the potentiometer input can be used for an analog (±10 volt) reference supplied by a potentiometer or a voltage source. This reference is converted into digital format and inserted in variable VALPOT. Conversion is at low speed (between 7 and 9 TU - 2.3 to 3ms, depending on the software installed).

The potentiometer should be connected to the ±SIN2 differential input of the 15-pin sub-D connector on the ST1's "R" board. If second-resolver adaptor P/N 024.7063 is used, the motor resolver should be connected into the X242 connector. The potentiometer or the voltage source are connected into connector X241, per the following diagrams.

SCHEMATIC DIAGRAM

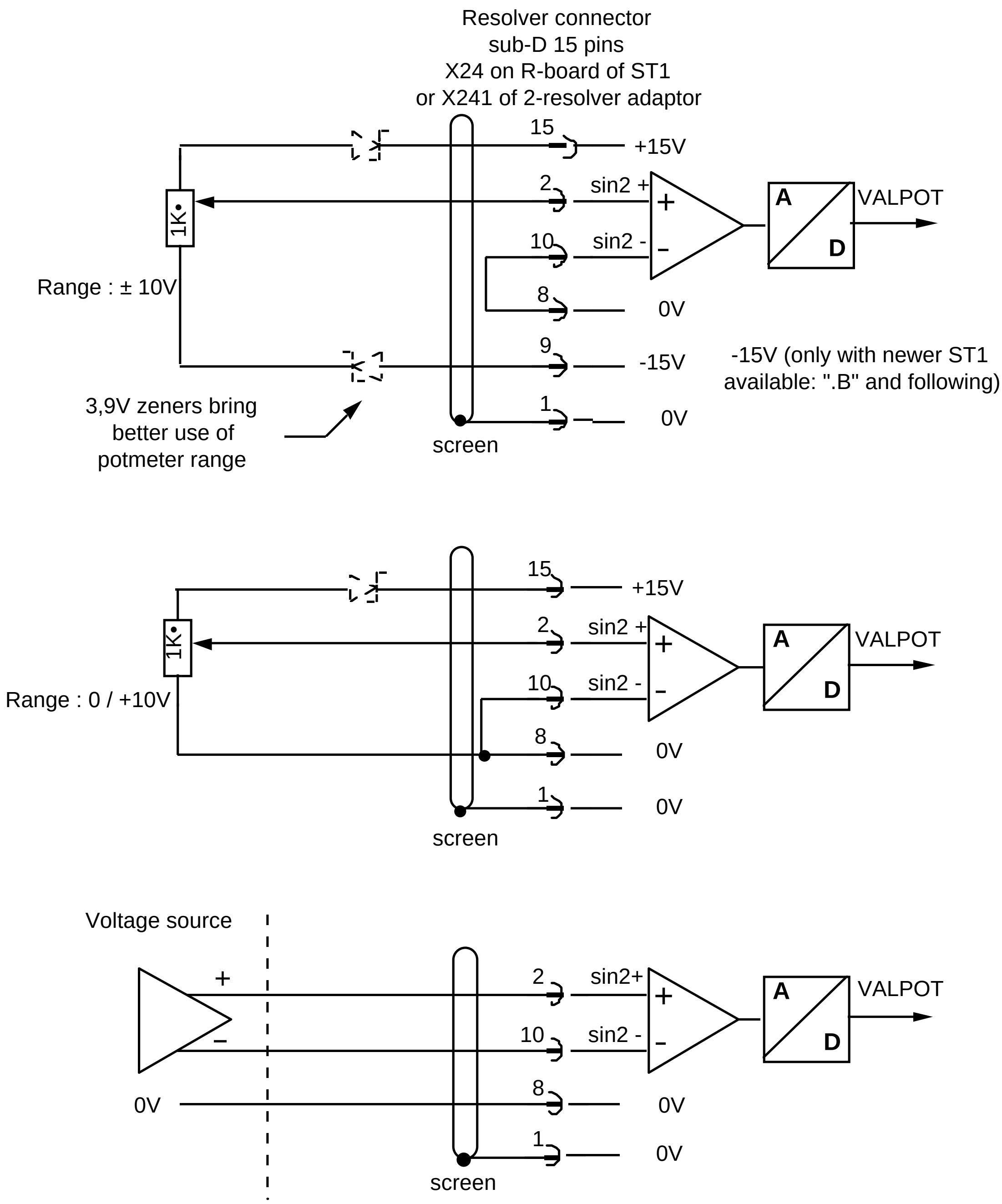


Figure 8.14

The reference VALPOT can be interpreted as the RATE factor by the IRG. In this case, only the VALPOT absolute value is considered. VALPOT can also be considered as a distance to travel - see instructions concerning sequencer.

VALPOT is an analog reference, conditioned by the following parameters :

8.12.2 Parameter OFPOT

This parameter expresses the offset added to the reference.

The value of OFPOT can be positive or negative, between 0 and ± 32,767 (hexa : 0 ... 7FFF).

Address :188 (hexa : BC)

Default value : 0.

8.12.3 Parameter FILTPOT

This parameter limits incrementing of the new reference value from the old value; it therefore performs a filter function.

Incrementing is based on a percentage of the input value :

$$newPOT = oldPOT + newPOT \cdot \frac{FILPOT}{4096}$$

It should be noted that a new conversion occurs once every 2.3 ms only.

The value of FILTPOT is positive, between 64 and 32,767 (hexa : 40 ... 7FFF).

Address :189 (hexa : BD)

Default value : 4,096 hexa : 1000).

8.12.4 Parameter COEPOT

This parameter expresses the coefficient applied to the reference.

$$VALPOT = newPOT \cdot \frac{COEPOT}{4096} + OFPOT$$

The value of COEPOT is positive, between 1 and 32,767 (hexa : 1 ... 7FFF).

Address :190 (hexa : BE)

Default value : 4,096 hexa : 1000).

8.12.5 Parameter ZMPOT

This parameter expresses a dead band around the reference value; a voltage inside the limits is considered as null.

The value of ZMPOT must be positive, between 0 and 32,767 (hexa : 1 ... 7FFF).

Address :191 (hexa : BF)

Default value : 0.

8.13. MISCELLANEOUS PARAMETERS

8.13.1 Parameter TIMTOP

This parameter can be employed to improve the precision of instructions ETOP and INIPM whenever they are available. What TIMTOP actually expresses is the physical detection and transmission delay that triggers these functions. One lsb in TIMTOP equals 0.02844 ms. The software takes account of this delay and the speed to correct the corresponding positions.

Computation of parameter TIMTOP takes account for the reaction time of the sensor and the 96delay dued to the input filter. A 1/6 ms delay must be added, which corresponds to half of the capture-time uncertainly (ST1 software cycle-time is 1/3 ms).

If input INO of the LIO board is used, this uncertainly may be reduced to 0.01422 ms. TIMTOP value must be then incremented by 32,768.

Example 1 :	Sensor delay :	2.5 ms
	Input filter delay :	0.6 ms
	Capture-delay (average) :	0.167 ms
	Total average delay :	3.26 ms
	TIMTOP = 3.26 ms / 0.02844 ms = 115	

Example 2, using input INO :

	Sensor delay :	1.2 ms
	Input filter delay :	0.6 ms
	Capture-delay (average)	0.014 ms
	Total average delay :	1.814 ms
	TIMTOP = 1.814 ms / 0.02844 ms + 32,768	
	= 32,832 (hexa : 8040)	

Address :56 (hexa s : 9C)

Default value : 0.

8.14 TRANSFERRING PARAMETERS

Most parameters are read by the software as required. Hence when any parameter is changed, the software will immediately take account of the new value.

The above principle does not apply to all parameters : certain functions would perform incorrectly if they immediately incorporated changes in certain special parameters. Thus the parameters are transferred into buffer memories, where they can be read and examined by the software. Transfer into buffers occurs at special times : the parameters can be grouped into three categories, depending on when they are bufferized :

- parameters CKA, CKAINV, CKV, CKR and CKH are transferred as soon as the processor is initialized (that is, at powering-up or reset);

- parameters CKS, CKM, CRANS, CKMS, CSCAN and COMOS are transferred simultaneous with execution of the instruction UPGRAD (see paragraph 10.6.6), and when the processor is initialized;

- parameters CVP, CA1, CA2, ASTOP, MSRATE, SRATE and RATE are transferred simultaneous with execution of the corresponding "move" instructions, that is, at the start of the particular movement.

9. DESCRIPTION OF SOME VARIABLES

9.1 GENERAL

All variables can be monitored from the host command or from the programming terminal. This section describes only the most useful variables. For the variables related to speeds and angular positions, it is assumed that the resolver has only one pole pair (single speed).

For each variable described a mnemonic code and its address are indicated.

9.2 BASE VARIABLES

URES : Address : 27 (hexa : 1B).

Primary excitation voltage applied to the resolver. This value is auto-matically adjusted in such a way that the sum of the squares of both secondary voltage levels returned by the resolver are constant, in accordance with the equation $\cos^2 a + \sin^2 a = 1$. The value of URES is compared to the value of the parameter CURES; if the deviation is too extreme, the ST1 digital motion controller software recognizes a fault in the resolver and disables the power stage.

See also in sections 6.2.2 (bit 7 of status STATB) and 7.3.1 (parameter CURES).

PHIC/PHIB : Addresses : 36, resp 37 (hexa : 24, 25).

Actual angular position of the resolver. This variable is expressed with double precision (2 words, being 31 bits + sign). 1 lsb = 2^{-16} resolver turn = 2^{16} RU, which implies that 1 lsb of PHIC = exactly 1 resolver turn.

PHIB corresponds to the absolute position measurement within one turn, the values - 32,768 ... 0 ... + 32,767 (hexa : 8000 ... 0 ... 7FFF) correspond to - 1/2 ... 0 ... + 1/2 turn.

PHIC corresponds to the incremental measurement of the number of turns. With the softwares supporting multi-resolver position measurement, the value of PHIC can be considered as absolute.

PHIREC/PHIREB : Addresses : 49 resp. 50 (hexa : 31, 32).

Angular reference position of the resolver, 1 lsb = 2^{-16} resolver turn = 2^{-16} RU, involving that 1 lsb of PHIREC = 1 resolver turn exactly.

PMUSC/PMUSB Addresses : 72 & 73, respectively (hexa : 48 & 49).

Angular position of master shaft, converted into slave units, hence same units as PHIREC/PHIREB.

DPOS : Address : 53 (hexa : 35).

Angular position lag. 1 lsb = 2^{-16} resolver turn.

VIRB : Address : 41 (hexa : 29).

Actual angular speed of the resolver. 1 lsb = 0.288 rad/s = 2.75 rpm.

VIREFB : Address : 47 (hexa : 2F).

Set angular speed of the resolver. 1 lsb = 0.288 rad/s = 2.75 rpm.

VMUSB : Address : 75 (hexa : 4B).

Speed of master shaft converted into slave units, hence same units as VIREFB.

DVIT : Address : 52 (hexa : 34).

Angular speed lag. 1 lsb = 0.018 rad/s = 0.17 rpm.

KRATE : Address : 195 (hexa : C3).

General coefficient applied to speeds; the value 4096 (hexa : 1000) corresponds to a 100-percent speed.

PALINS : Address : 203 (hexa : CB).

Another coefficient applied to speeds, depending on the instruction. A value of 32,767 (hexa : 7FFF) corresponds to a 100-percent speed.

FLSUIV : Address : 192 (hexa : C0).

Variable indicating the state of synchronicity : 0 = no synchronicity; 1 = synchronicity.

COCOU : Address : 55 (hexa : 37).

Required motor torque delivered from the position and speed controller after the limitation by COUMA or by COULIM. 1 lsb = 1/128 Nm.

MOTB/MOTA : Addresses : 120, resp. 121 (hexa : 78, 79).

Copy in form of two words of bytes of data received in combination with an instruction.

9.3 PSEUDO-PARAMETERS

For technical reasons, a few variables are memorized as the current values of the parameters. These variables are read as if they were parameters, from the programming terminal or by serial communication from a host computer.

TEMP : Address : 249 (hexa : F9)

Temperature of the power stage. 1 lsb = 1/16th of a degree C.

10. EXECUTABLE INSTRUCTIONS

10.1 GENERAL

An essential quality of the ST1 digital motion controller is that it has an informatics kernel. An interactive dialog with the host processor is possible whereby the host processor transmits the instructions (commands) to the ST1 digital motion controller and receives information in return. Moreover, some software releases offer sequencer functions, where preloaded instructions may be called for execution through binary inputs.

The following information is given for each instruction described in this chapter :

- The instruction code.
- The number of data bytes and their meaning.
- The meaning of the response byte.
- The validation test (if any).

The instructions described in this chapter are not all available in each software version. Please refer to the listing provided with each software version for detailed information.

For instructions which are available in the boot program, an indication is given.

Instructions specific to the sequencer function are described in Manual ref. 024.8020 - "LIO Interface Board".

10.2 INSTRUCTIONS MODIFYING THE PARAMETERS

10.2.1 Instruction SETPAR (or SETMEM)

(Available in the boot program)

Instruction code : 42 (hexa : 2A).

Data bytes : 1. Address of the parameter to be modified.
2. New value (msb).
3. New value (lsb).

Answer byte : Status STATA.

Tests : (on some versions only) :

Address belongs to the parameter field.

This instruction allows a modification of the current value of any parameter. It can also be used to load the parameters during initial installation. The new parameter value is not checked for correctness.

10.2.2 Instruction SETVAR

(Available in the boot program)

Instruction code : 61 (hexa : 3D).

Data bytes : 1. Address of the variable to be modified.
2. New value (msb).
3. New value (lsb).

Answer byte : Status STATA.

Tests : None.

This instruction allows a modification of any variable's value.

Some variables are decided by the software. Therefore, a SETVAR change can not always be accepted.

10.2.3 Instruction SETOUT

(Available in the boot program)

Instruction code : 64 (hexa : 40).

Data bytes : 1. New value (1 byte).
2. Address (msb).
3. Address (lsb).

Answer byte : Status STATA.

Tests : None.

With this instruction it is possible to set a value into memories or outputs connected to the the microprocessor I/O bus.

<u>List of addresses on the I/O bus</u>

0 ... 32,767	(hexa : 0 ... 7FFF)	program memory of the ST1.
36,864 ... 38,911	(hexa : 9000 ... 97FF)	parameter memory
40,976	(hexa : A010)	analog output A1 of the test device
40,977	(hexa : A011)	analog output A2 of the test device
40,984	(hexa : A018)	digital output 8-bits of the test device

Other addresses allow the access to the link and optional boards. Please refer to the appropriate documentation.

10.2.4 Instruction SPWARN

Instruction code : 60 (hexa : 3C).

Data bytes :
1. Indication point (msb).
2. Indication point (intermediate byte).
3. Indication point (lsb).

Answer byte : Status STATA.

Tests : None.

Bit 5 of status STATA indicates whether the reference position (PHIRE) is to the left or to the right of a threshold.

In such a case, the instruction SPWARN specifies the distance the axis should travel to the left or right before the indication is issued. The relative indication supplied may be positive or negative, irrespective of the direction of motor rotation.

Bit 5 of STATA is updated before this answer is sent, as a form of acknowledgement.

10.2.5 Instruction SAVE

Instruction code : 57 (hexa : 39).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction allows all parameter current values to be stored into the parameter memory (EEPROM). If data entered during set-up is not saved using this command, it will be lost when system power is removed.

The transfer procedure takes about eight seconds, during which it is necessary to avoid any supply failure as well as resetting the microprocessor. In some software versions STATC bit 3 indicates that the SAVE procedure is in progress.

10.3 READ INSTRUCTIONS

10.3.1 Instruction ASKVAR (or ASKDAT)

(Available in the boot program)

Data bytes : Address of the inquired variable.

Answer byte : Value requested (msb).

Tests : None.

This instruction allows to obtain through the serial link the instantaneous value of a variable. The ST1 digital motion controller sends the most significant byte in response. The least significant byte is memorized and can be obtained later on with the instruction ASKNXT.

10.3.2 Instruction ASKPAR (or ASKMEM)

(Available in the boot program)

Instruction code : 43 (hexa : 2B).

Data bytes : Address of the inquired parameter.

Answer byte : Value requested (msb).

Tests : None.

This instruction allows to obtain through the serial link the current value of a parameter. The ST1 digital motion controller sends the most significant byte in response. The least significant byte is memorized and can be obtained later on with the instruction ASKNXT.

10.3.3 Instruction ASKPRG

(Available in the boot program)

Instruction code : 62 (hexa : 3E).

Data bytes : 1. Address (msb).

2. Address (lsb).

Answer byte : Value requested (msb).

Tests : None.

This instruction is similar to instruction ASKPAR but with an address range of 16 bits. It is only used in special cases after consultation with the software developer.

10.3.4 Instruction ASKIN

(Available in the boot program)

Instruction code : 47 (hexa : 3F).

Data bytes : None.

Answer byte : Value requested (1 byte).

Tests : None.

This instruction allows to read a byte on memories or inputs connected to the microprocessor I/O bus.

The specific addresses are listed in section 10.2.3.

10.3.5 Instruction ASKNXT

(Available in the boot program)

Instruction code : 18 (hexa : 12).

Data bytes : None.

Answer byte : Value requested (following byte).

Tests : None.

This instruction allows to obtain through the serial link the second byte of the value requested by one of the ASK instructions.

It also allows a faster means of reading several parameters, giving the possibility to sequentially view a set of values without providing the subsequent addresses.

10.3.6 Instruction ASKPOS

Instruction code : 44 (hexa : 2C).

Data bytes : None.

Answer byte : Value requested (msb).

Tests : None.

This instruction allows reading through the serial link of the instantaneous position-reference value (PHIRE), converted into LU, in a 2-word or 4-byte format. The ST1 digital motion controller sends the most significant byte in response. The following 3 bytes are memorized and can be obtained in the order with 3 successive ASKNXT instructions.

10.3.7 Instruction ASKACT

Instruction code : 59 (hexa : 3B).

Data bytes : None

Answer byte : Required value (msb).

Tests : None.

This instruction allows the current position (PHIC/B) (in LU's) to be captured through the serial interface board. The ST1 transmits the most significant byte. The following three bytes are memorized and can be written into the command by means of three successive "ASKNXT" instructions.

10.3.8 Instruction ASKABS

Instruction code : 66 (hexa : 42).

Data bytes : None

Answer byte : Required value (msb).

Tests : None.

This instruction allows the absolute position (PABSC/B) (in LU's) to be captured through the serial interface board. The ST1 transmits the most significant byte. The following three bytes are memorized and can be written into the command by means of three successive "ASKNXT" instructions.

10.3.9 Instruction ASKTOR

Instruction code : 59 (hexa : 3B).

Data bytes : None.

Answer byte : Value requested (msb).

Tests : None.

This instruction allows to obtain through the serial link the torque set value for the motor respectively the variable COUCOU. The value 128 (hexa : 80) corresponds to 1 Nm.
The ST1 digital motion controller will respond with the most significant byte. The least significant byte is memorized and can be read by instruction ASKNXT.

10.4 INSTRUCTIONS CONCERNING THE STATUS

10.4.1 Instruction NOOP

(Available in the boot program)

Instruction code : 1.

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction allows to obtain through the serial link the value of STATA, without affecting the operation of the ST1 digital motion controller. When the host control is not sending new instructions to the ST1 digital motion controller, it should periodically send NOOP in order to be informed as fast as possible about a possible fault.

10.4.2 Instruction ASKSTB

Instruction code : 45 (hexa : 2D).

Data bytes : None.

Answer byte : Status STATB.

Tests : None.

This instruction allows to obtain through the serial link the value of status STATB. It is only used for system diagnostics after the power stage has been disabled which is indicated by bit 7 of status STATA or when the ST1 does not enable when commanded.

10.4.3 Instruction ASKSTC

Instruction code : 46 (hexa : 2E).

Data bytes : None.

Answer byte : Status STATC.

Tests : None.

This instruction allows to obtain through the serial link the value of status STATC. The value of status STATD is memorized and can be read by instruction ASKNXT.

The ASKSTC instruction also sets bit 6 of STATA to zero. Any new change in STATC or STATD will thus be signalled by bit 6 of STATA switching again to one. See also section 6.3 on the mask CMASKS

10.4.4 Instruction RESSTS

(Available in the boot program)

Instruction code : 47 (hexa : 2F).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction is used for zeroing the memorized bits of status STATC and STATD as well as bit 6 of STATA.

10.5 INSTRUCTIONS CONTROLLING THE POWER STAGE

10.5.1 Instruction PWROFF

Instruction code : 2.

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction causes the immediate disabling of the power stage. All current to the motor is zeroed thus the motor can not be stopped by the ST1 digital motion controller once this command is given.

Disabling the system with the PWROFF command does not change the state of bit 7 of STATA which signals that a fault has occurred. Only bit 0 of STATD is set to 1. If the value of status mask CMASKS is set accordingly, bit 6 of STATA can signal this change.

10.5.2 Instruction PWRRES

Instruction code : 33 (hexa : 21).

Data bytes : 1. Security code = 31 (hexa : 1F).
2. Security code = 31 (hexa : 1F).
3. Security code = 31 (hexa : 1F).

Answer byte : Status STATA.

Tests : Security code 2,039,583 (hexa : 1F,1F1F).

This instruction allows reinitializing the four status bytes (idem RESSTS) and the power-stage fault circuits, without unblocking the power stage. The security code is to prevent accidental resetting during an undetected transmission error.

10.5.3 Instructions PWRONS and PWRONI

Instruction code : 35, resp. 36 (hexa : 23, 24).

Data bytes :	1. Security code = 31 (hexa : 1F). 2. Security code = 31 (hexa : 1F). 3. Security code = 31 (hexa : 1F).
Answer byte :	Status STATA.
Tests :	Security code 2,039,583 (hexa : 1F,1F1F) correctly received.

Both instructions first call the instruction PWRRES as a subprogram. As long as no system faults are present, the power stage is enabled. As the position actual value is chosen as set value, the ST1's controller holds the motor at start-up position. A security code is used to prevent any undesired enabling due to a not detected transmission error.

Unlike the PWRONS instruction, the PWRONI instruction causes in addition a reinitialization to zero of the incremental part PHIC of the position measurement. When a single resolver is used, the position measurement PHIC/PHIB is then set between -1/2 and + 1/2 turn of the motor. See the description of this variable in section 9.2 for further explanations.

10.5.4 Instruction PWRONR

Instruction code : 34 (hexa : 22).

Data bytes :	1. Safety code = 31 (hexa : 1F). 2. Safety code = 31 (hexa : 1F). 3. Safety code = 31 (hexa : 1F).
Answer byte :	Status STATA.
Tests :	Safety code 2,039,583 (hexa : 1F,1F1F) correctly received.

This instruction is similar to PWRONS. However, the position set value does not change. The motor will therefore not stay at the start-up position, but will start back to the old position set value.

The movement that occurs is not controlled from the ramp generator but from the position and velocity controller. Therefore this move is influenced solely by parameters KPOS, KVIT, KTEGR, and FEXT.

10.6 INITIALIZATION INSTRUCTIONS

10.6.1 Instruction INIPM

Instruction code : 27 (hexa : 1B).

Data bytes : None

Answer byte : Status STATA.

Tests : None.

This instruction initializes PMUS and MOMUS (master-shaft position and period). MOMUS is reset to zero. To allow for the switching delay compensation, PMUS is initialized as :

$$PMUS = TIMTOP \cdot VMUS$$

It is used to complete non-absolute systems such as :

- OM/OS,
- OEI without "zero" track,
- second resolver when its given absolute position is not of use.

10.6.2 Instruction INIPS

Instruction code : 26 (hexa : 1A).

Data bytes : None

Answer byte : Status STATA.

Tests : None.

This instruction causes zero-reset of the incremental part of counters PHIC and PHIREC. They also update INTV and MOPHI.

10.6.3 Instruction ALISU

Instruction code : 28 (hexa : 1C).

Data bytes : None

Answer byte : Status STATA.

Tests : None.

The purpose of this instruction is to save the instantaneous physical position of the slave shaft as a reference for later synchronizing steps. It provokes zero-reset of INTV and as a result corrects variable PCOR and parameter CPCOR, to avoid changes in reference position PHIRE (see description of parameter CPCOR and explanation of independence of resolver zero).

In order to limit the number of motor revolutions in parameter CPCOR, an INIPS instruction can be executed previously.

If instruction ALISU is executed only once (at commissioning), and not after each reset, the CPCOR parameter must be saved in a SAVE instruction. In this case, the instruction ALISU should be preceded by an INIPS instruction to ensure that a particular number of motor revolutions is not taken into account.

10.6.4 Instruction ALIPOS

Instruction code : 20 (hexa : 14).

Data bytes : None

Answer byte : Status STATA.

Tests : None.

Like the instruction ALISU, this instruction saves a physical situation by initializing the variable PCOR and the parameter CPCOR.

For this, ALIPOS takes account not only of the slave position, but of the relative positioning of the master and slave shafts, to ensure their relationship is respected subsequently.

ALIPOS does not reset INTV to zero; it adjusts PCOR in order to get zero-difference between INTV and PMUS, taking account of any periodicities involved.

Apart from this, ALIPOS is used in the same way as the instruction ALISU. If ALISU imposes a prior INIPS value, ALIPOS will also ask for an INIPM.

10.6.5 Instruction INIABS

Instruction code : 32 (hexa : 20).

Data bytes : 1. New value (most significant byte)
2. New value (intermediate byte)
3. New value (least significant byte)

Answer byte : Status STATA.

Tests : None.

This instruction initializes the absolute position. The instructions ERMOV, PRMOV, EAMOV and PAMOV employ (and update) this position. The new value is specified in [LU].

10.6.6 Instruction UPGRAD

Instruction code : 19 (hexa : 13).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction copies certain parameters into the variables, so that modifications to various values will be recognized simultaneously.

The parameters in question are : CKS, CKM, CKMS, CRANS, COMOS and CSCAN.

For example, if it is necessary to change the transmission ratio between the master shaft and the slave shaft, the corresponding parameters (CKS, CKM, CKMS and CRANS) are modified then activated using UPGRAD.

10.6.7 Instruction EQUPOS

Instruction code : 40 (hexa : 28).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction takes the actual motor position as the reference position, and updates the corresponding components of the reference (variables INTV and MOPHI).

10.6.8 Instruction RESPOS

Instruction code : 38 (hexa : 26).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction has no effect : it is saved for compatibility with softwares non-specific for master/slave uses. This instruction is not needed since absolute values are processed differently.

10.7 "MOVE" INSTRUCTIONS

10.7.1 General Information

The instruction set described in this section refers to ST1 software configured for master/slave synchronizing. A similar instruction set exists for earlier ST1's which, although less suited to synchronizing applications, nevertheless offer the possibility of "smoothing" acceleration jumps during machine movements.

The instructions in question generate motion via the internal reference generator (IRG) described in paragraph 8.4.

The instructions operate in one of two ways :

- Those with mnemonic codes beginning with "E" (for "execute") allow "informing" the ST1 of the type of motion required, as well as the desired speed or travel. The movement begins as soon as the instruction is received and an acknowledgement is issued in the form of status STATA. Acknowledgements for subsequent instructions (if none, NOOP's) record the progress of the movement (in particular, the end of the movement).

- Those with mnemonic codes beginning with "P" (for "prepare") are the same, except that the movement does not begin until a START instruction is received. When several axes are to execute different but simultaneous movements, it is advisable first to issue each axis with its travel or speed; only then is the START command sent to initiate simultaneous movement of all the axes.

A new movement can always be commanded, even if the current movement has not finished.

10.7.2 Instruction START

Instruction code : 13 (hexa : D).
Data bytes : None.
Answer byte : Status STATA.
Tests : None.

Causes the start of the preloaded movement. A START is ignored if it occurs before the preceding movement has been finished.

10.7.3 Instruction STOP

Instruction code : 12 (hexa : C).
Data bytes : None.
Answer byte : Status STATA.
Tests : None.

Interrupts the movement in progress. Braking is immediate. This instruction also desynchronizes a slave shaft. The acceleration value depends on the parameter ASTOP. Status STATA evolves in the same way as for an ERUN (null-speed) instruction.

10.7.4 Instructions EAMOV and PAMOV

Instruction code : 7 and 6, respectively.
Data bytes : 1. Final position (msb).
2. Final position (intermediate byte).
3. Final position (lsb).
Answer byte : Status STATA.
Tests : None.

These instructions execute or prepare the movement (acceleration, velocity, deceleration) towards the given final position (specified in LU units). This position can be positive or negative.

As soon as the final position is received, the distance to travel is converted into internal units, and the movement or displacement direction is determined.

The absolute reference is the motor position after the ST1 is powered up. It can be changed to another value at any time, using the instruction "INIABS".

The absolute position only takes account of motion generated by the IRG : it ignores movements commanded by the master shaft.

In the case of a long movement, the velocity profile is trapezoidal; the velocity corresponds to the current values of parameters CVP and RATE.

In the case of a small movement, a lower velocity is calculated, and the velocity profile is almost triangular.

10.7.5 Instructions ERMOV and PRMOV

Instruction code : 5 and 4, respectively.

Data bytes : 1. Distance to run (msb).
2. Distance to run (intermediate byte).
3. Distance to run (lsb).

Answer byte : Status STATA.

Tests : None.

These instructions execute or prepare movement over a given distance, which is specified in "LU" units. The distance can be positive or negative, the reference being the last final position. After calculation and memorizing of the new final position, the ST1 reacts in the same way as for an E/PAMOV instruction.

If a movement is in progress as a new ERMOV instruction is received, a single "general" movement is performed, equal to the sum of the two movements. There is no intervening stop.

If a PRMOV "movement-preparation" instruction has already been sent to the ST1, it will be cancelled by the new PRMOV.
If a "RUN" type movement is in progress, an RMOV will refer to the shaft position at the time the instruction is received.

10.7.6 Instructions ERUN and PRUN

Instruction code : 9 and 8, respectively.

Data bytes : 1. New speed (msb).
2. New speed (lsb).

Answer byte : Status STATA.

Tests : 1. None.

These instructions produce and maintain the motor rotation speed at the value specified, or prepare such a movement. They apply equally to an acceleration or a deceleration.

The new speed is expressed as a percentage of the full scale, as for parameter CVP described in paragraph 8.4.6.

The sign of the value transmitted indicates the desired direction of rotation. Speed reversal is tolerated.

The instructions also desynchronize a slave shaft in relation to a master shaft. After the instruction, the speed will thus be the new speed independent of the master shaft.

Although the displacement value is not taken into account in producing the speed component, the motor can remain slaved to position and speed if the parameters KPOS and KTEGR are correspondingly adjusted.

Similarly, the absolute position is continuously updated to validate later AMOV instructions.

10.7.7 Instructions ESRUN and PSRUN

Instruction code : 17 and 16, respectively (hexa : 11 & 10).

Data bytes : 1. New address (msb)
2. New address (lsb)

Answer byte : Status STATA.

Tests : None.

These instructions allow executing or preparing speed-synchronizing of a slave shaft in relation to a master shaft. The new speed specifies an offset in relation to the master, and can be positive or negative. It is expressed as a percentage of the full-scale reading

10.7.8 Instruction ESPOS and PSPOS

Instruction code : 22 and 21 respectively (hexa : 16 & 15).

Data bytes :
1. Distance to run before synchr. (msb)
2. Distance to run before synchr. (intermediate byte)
3. Distance to run before synchr. (lsb).

Answer byte : Status STATA.

Tests : None.

These instructions allow executing and preparing for synchronizing ("phasing") of a slave shaft with a master shaft. The distance to run represents the difference between the slave and the master shafts after synchronizing, and is given in [LU].

The moment of synchronizing depends on the positions and speeds of both shafts. If these positions are to be independent of speed during the synchronizing period, it will be necessary to modulate acceleration (SRATE = hexa : 000x) (simulates cam operation).

The positions taken into account are described in paragraphs 8.7 and 8.8 "Switching the Synchronizing Function" and "Periodicity".

10.7.9 Instructions ERPOS and PRPOS

Instruction code : 25 and 24 respectively (hexa : 19 & 18).

Data bytes :
1. Distance to run after synchr. (msb)
2. Distance to run after synchr. (intermediate byte)
3. Distance to run after synchr. (lsb).

Answer byte : Status STATA.

Tests : None.

These instructions allow executing and preparing for desynchronizing of a slave shaft with a master shaft. The distance to run represents an amount of travel relative to the slave periodicity such that stopping will always occur at the same point in the period.

The velocity during desynchronizing depends on the value of the parameter MRATE.

If desynchronizing is to occur without the master losing position, parameter ZMPERM should be set to hexa : 8000 (see "pseudo-periodic mode").

10.7.10 Instruction ETOP

Instruction code : 14 (hexa : E).

Data bytes :
1. Distance to travel (msb).
2. Distance to travel (intermediate byte).
3. Distance to travel (lsb).

Answer byte : Status STATA.

Tests : None.

This instruction allows stopping the motor after detection of an external event (pulse); the distance to travel after the event is specified.

The movement continues at the same speed as that defined in the interrupted movement.

The external event is signalled to the ST1 by a transition at the corresponding input on the interface board. It is possible to compensate an average "reaction time" by adjusting parameter TIMTOP described in paragraph 8.13.1.

10.8 TORQUE CONTROL AND LIMITATION

10.8.1 Instruction LIMTOR

Instruction code : 55 (hexa : 37).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction limits the motor torque to a value less than the maximum torque. The value of the new torque limit is specified by the parameter COULIM.

10.8.2 Instruction RESTOR

Instruction code : 56 (hexa : 38).

Data bytes : None.

Answer byte : Status STATA.

Tests : None.

This instruction allows the re-establishment of the maximum motor torque to a value which is specified by the parameter COUMA.

10.9 SOFTWARE DOWNLOADING

10.9.1 Instruction INIPGM

(Only available in the boot program)

Instruction code : 31 (hexa : 1 F)

Data bytes :
1. Word adress in EEPROM (msb)
2. Word adress in EEPROM (lsb)

Answer byte : Status STATA

Tests : None

This instruction sets a pointer which will be used later-on, while receiving an instruction LDPGM.

10.9.2. Instruction LDPGM

(Only available in the boot program)

Instruction code : 29 (hexa : 1 D)

Data bytes :
1. Program word to be stored (msv)
2. Program word to be stored (lsb)

Answer byte : Status STATA

Tests : EEPROM is not over, bit 2 of STATC not equal to 0, this instruction is not plausible. (Bit 6 of STATC will be immediately set and an "U" will be displayed.)

This instruction provides the ST1 with one word of the downloaded software, which is to be stored at the EEPROM location depending on the pointer value. This pointer is to be set before sending the first LDPGM instructions using the INIPGM instruction. It will be automatically incremented by 2 after each reception of an LDPGM instruction.

10.9.3. Instruction FILL

(Only available in the boot program)

Instruction code : 30 (hexa : 1 E)

Data bytes : 1. Filling word to be stored (msb)

2. Filling word to be stored (lsb)

Answer byte : Status STATA

Tests : If the writing of the preceding word into EEPROM is not over, bit 1 of STATC not equal to 0, this instruction is not plausible. (Bit 6 of STATC will be immediately set and an "U" will be displayed.)

This instruction is used in order to fill all unused EEPROM locations with a given value (recommended value is O). This is helpful for computing the checksum value.

As this filling procedure needs some time (up to 10 seconds), bit 1 of status STATC is set to 1 upon receipt. It returns back to 0 as soon as the procedure is over.

10.9.4. Instruction CHKSUM

(Only available in the boot program)

Instruction code : 28 (hexa : 1 C)

Data bytes : 1. Result value (msb)

2. Result value (lsb)

Answer byte : Status STATA

Tests : If the writing of the preceding word into EEPROM is not over, bit 0 of STATC not equal to 0, this instruction is not plausible. (Bit 6 of STATC will be immediately set and an "U" will be displayed.)

This instruction is used in order to check the received software upon its checksum. The intended result value has to be computed and transmitted to the ST1 which compares it with its own checksum. As this checking procedure needs some time (about 200 ms), bit 0 of status STATC is set to 1. As soon as this procedure is over, this bit is set back to 0. Should ST1's checksum procedure lead to an error statement, then the ST1 displays a "H".

10.9.5. Instruction REBOOT

(Only available in the boot program)

Instruction code : 27 (hexa : 1 B)

Data bytes : None

Answer byte : Status STATA

Tests : If no valid downloaded software is present in the ST1, or if the checksum was not correct, bit 6 of STATC is set to 1, thus indicating an unplausible instruction.

Upon receipt of a REBOOT instruction, the ST1 checks if a valid downloaded software is available.

If yes : A reset is initiated, which needs about 50 ms; the downloaded software is then active. Later on, every time the ST1 will be powered-up or resetted, the downloaded software will also be also active.

If no : The boot program remains active. Bit 6 of status STATC indicates the error.

INDEX

(Numbers refer to corresponding pages)