Sefram

PROGRAMMING MANUAL

DAS30 – DAS50 – DAS60

Edition October 2020

CONTENT

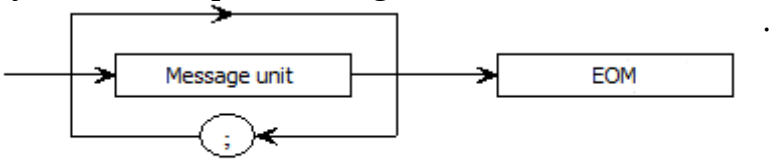
1. Programming language

1.1. Format of the reception messages

In all following examples, the blank character is displayed as a space.

Exchanges from a controller to the recorder are made of messages as successive ASCII characters (and possibly binary octets) with an EOM at the end.

Syntax of a reception message



Message unit: if the message includes several message units, they are separated by a semi-colon " ; " and with possible one or several "filling" characters before and after in ASCII code (0 to 32, decimal, except 10 and 13).

The EOM is designed for the Ethernet link:

- LF: Line Feed (10 in decimal)

The EOM may be preceded with one or several "filling" characters in ASCII code (0 to 32, decimal, except 10 and 13).

Message example made of 3 message units:

MESSAGE 1; MESSAGE 2; MESSAGE 3; EOM

CHANNEL 1; TYPE:VOLTAGE DC; CALDEC ? EOM

Syntax of a message unit

A message unit (example: REAR:SETUP 1) is made of several fields:

- *Header:*

For command messages (example: **REAR:SETUP** 1) or interrogation messages (example: **REAR** ?), it is made of a chain of characters (simple header) or of several chains separated with the ":" character (composite header).

A chain includes 1 to 12 alphanumerical characters or "_" (ASCII code 95 in decimal).

Recommended chain length: 4 characters.

A header chain must start with an alphanumerical character. It may be preceded by 2 dots ":" (composite header) or finish with a question mark "?" (interrogative message).

An interrogative message must be followed by an EOM.

- *Header separator:*
One or several ASCII characters (0 to 32, decimal, except 10 and 13).

- *One or several data items:*
(example: SPEED **1**, **MM_S**), alphanumerical, numerical or made of any characters and binary octets.

- *Data separator:*
A comma "," possibly followed and/or preceded with one or several "filling" characters in ASCII code (0 to 32, decimal, except 10 and 13).

Data:
There are several types of data items:

- *Alphanumerical data:*
1 to 12-character words that can be alphabetical (upper or lower case), digital or the "-" character (95d).
A word always starts with an alphabetical character.
For example, for a non-digital parameter: S1M.
- *Decimal digital data:*
Made of a significand and, possibly, an exponent, and displayed as a chain of ASCII-coded characters starting with a digit or a sign (+ or -). It is of NR1 (integer), NR2 (decimal) or NR3 (with exponent) type or a combination of these three types.
- *Text:*
Any chain of characters under 7-bit ASCII code, between quotation marks (") or apostrophes (').
For example: "Channel 1"

1.2. Formats of the emited messages

Exchanges from the recorder to a controller are made of messages as successive ASCII characters (and possibly binary octets) with an EOM at the end.
The format of the emission messages is identical to the reception messages. However, their structure is stricter.

The syntax of an emission message is: **message unit + EOM**

Message unit:
If the message includes several message units, they will be separated with a semicolon ";".

EOM:
- LF: Line Feed (10 in decimal)

Syntax of a message unit:

A message unit (for example: TYP:THE J, COMP) is made of several fields

- *Header:*
(for example: **TYP:THE**) is made of one (simple header) or several (composite header) 1 to 12-character alphabetical chains (upper case only or digital or "_" (ASCII code 95 in decimal)
A header chain starts with an alphabetical character.
In a composite header, the chains of characters are separated with the ":" character (for example: TYP:THE).

- *Header separator:*
"Space" character (32d) only.

- *One or several data items:*
(for example: **J**, **COMP**) alphanumerical, numerical or made of any characters and binary octets.

- *One data separator:*
A comma ",".
Data:
There are several types of data items:

- *Alphanumerical data:*
1 to 12-character words that can be alphabetical (upper case only), digital or the "-" character (95d) (example: **J**).

- *Decimal digital data:*
Made of a chain of ASCII-coded characters starting with a digit or a sign (+ or -). It is of NR1 (integer), NR2 (decimal) or NR3 (with exponent) type.
For example, for a digital character: -25.02.

- *Text data:*
Any chain of characters under 7-bit ASCII code, between quotation marks (") or apostrophes (').
For example: "A".

- *Any chain of ASCII characters:* ends with the EOM.

2. Standard instructions

All these instructions start with an asterisk "*".

***IDN ?** IDENTIFICATION REQUEST OF AN APPLIANCE
answer by the appliance: 4 data items separated with ',':

- the trademark of the appliance
- the name of the appliance followed with _nn, where nn is the number of inputs of the recorder
- the serial number of the appliance (0 if unknown)
- the software version number as x.xx x

***OPT ?** IDENTIFICATION REQUEST OF THE OPTIONS OF AN APPLIANCE
answer by the appliance: n data items separated with ';':

- number of cards
- number of channels per card

***RST** INITIALIZATION OF AN APPLIANCE
action: initialization of the recorder in a fix configuration (inputs under voltage, caliber: 10 V, center: 0 V...)

***REM** TRANSITION TO PROGRAMMING (REMOTE)
compulsory with RS232C before sending any other program command.

***LOC** RETURN TO LOCAL MODE

***CLS** CLEARING THE STATE REGISTERS
action: the appliance reinitializes the state registers.

***ESE** VALIDATION OF THE STANDARD EVENT BITS OF AN APPLIANCE
*ESE is followed with a number between 0 and 255
action: changes the standard event validation register and updates the ESB bit in the state register of service requests (see the following paragraph).

***ESE ?** REQUEST OF THE CONTENT OF THE STANDARD EVENT VALIDATION REGISTER OF AN APPLIANCE
Answer by the appliance: NR1 number from 0 to 255 (see the following paragraph).

***ESR ?** REQUEST OF THE CONTENT OF THE STANDARD EVENT VALIDATION REGISTER OF AN APPLIANCE
Answer by the appliance: NR1 number from 0 to 255
All events are erased and the register is cleared (see the following paragraph).

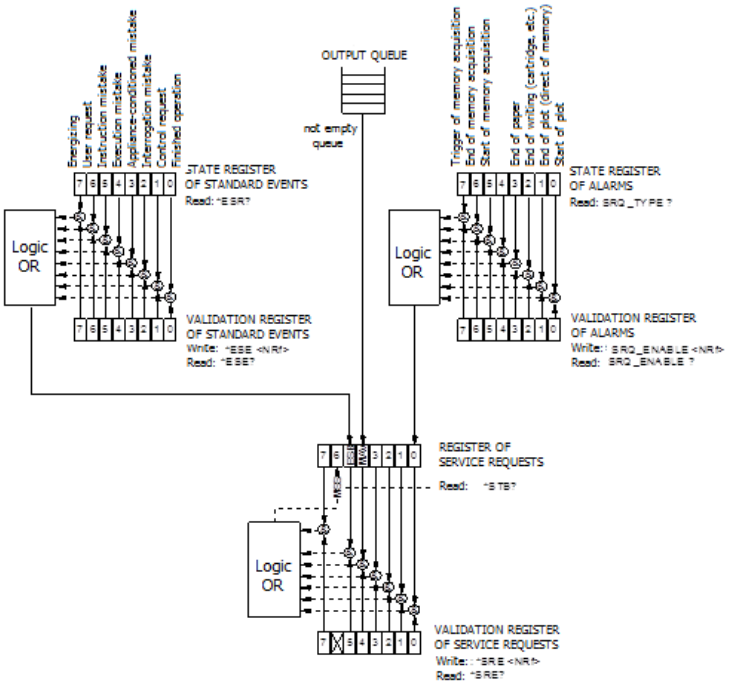
***SRE** VALIDATION OF THE SERVICE REQUESTS OF AN APPLIANCE
*SRE is followed with a number between 0 and 63 or from 128 to 191.
action : the appliance changes the validation register of service requests (see the following paragraph).

***SRE ?** INTERROGATION OF THE VALIDATION REGISTER OF THE SERVICE REQUESTS OF AN APPLIANCE
answer by the appliance: NR1 number from 0 to 63 or from 128 to 191 (see the following paragraph).

***STB ?** READING THE SERVICE REQUEST REGISTER OF AN APPLIANCE
answer by the appliance: NR1 number from 0 to 255: state word with bit 6 MSS (Master Summary Status) (see the following paragraph)

2.1. State indication of the appliance

Here is the model of structure of the state data that documents state changes in the appliance (energizing, printing launch...).



Overview of the structures of the state data of the register:

4 registers are used:
- the register of service request (STB) associated with its validation register
- the register of standard events (ESR) associated with its validation register

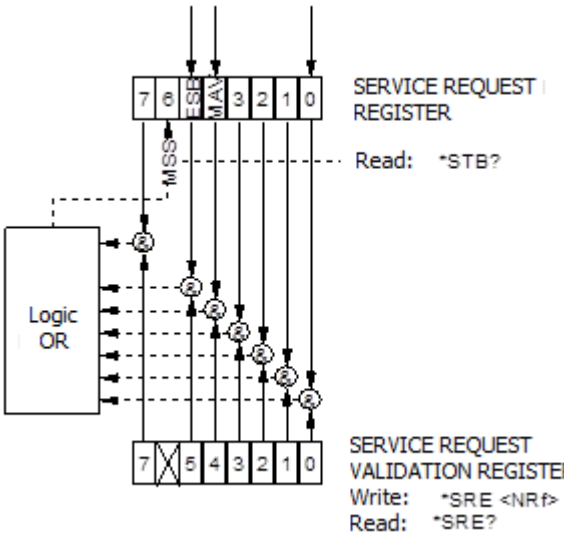
The bits #0, 1, 2 and 7 of the STB register are available as sum-up messages specific to the appliance. Each of these bits can be associated with a data structure, whose model is defined and manages the events of the appliance that may induce a service request.

The user can set up the recorder so that it triggers the bit #6 of the service request register at a few specific events.
In RS232, you have to regularly read the service request register to detect events.
Events are identified by reading the state word, then the associated event register(s).

State of these registers at power-up:

The content of the STB, ESR and alarm registers is systematically cleared at power-up (except the bit #7 of the ESR that specifies a power-up).

2.2. Service request register



State register:
It contains the state word of the appliance.
This state word can be read by request with the instruction "*STB?": In this case, the bit #6 is MSS (Master Summary Status) resulting from the logic operations as in the figure here.
In fact, MSS is 1 when at least one other bit is 1 both in the state register and in the validation register.

Composition of the STB register:
The bit #6 (value 64) contains the sum-up message "MSS" (reading with "*STB?").

The service request takes place in the following cases:

- a bit from the service request state register switches from 0 to 1 while the corresponding bit in its associated validation register is at 1, and reversely
- the bit #5 of the service request validation register is at 1 and an event happens in the following conditions:
- a bit from the service request state register switches from 0 to 1 while the corresponding bit in its associated validation register is at 1
- a bit from the service request validation register switches from 0 to 1 while the corresponding bit in its associated state register is at 1
- the bit #0 of the service request validation register is at 1 and an event happens in the following conditions:
- a bit of the alarm state register switches from 0 to 1 while the corresponding bit in its associated validation register is at 1
- a bit of the alarm state register switches from 0 to 1 while the corresponding bit in its associated state register is at 1.

The bit #5 (ESB: Event Status Bit, value 32) contains the sum-up message of the standard events state register (see the detail of these bits in the description of this register). Its state specifies whether one or several authorized events showed up in the standard events state register after its latest clearing (an event is authorized if the corresponding bit in the event validation register is 1).

The bit #4 (MAV: Message AVailable, value 16) contains the sum-up message of the output queue. Its state specifies if a message or data from the appliance are ready for emission through the interface (ex: answer to an interrogative instruction).

The bits #7 and 3, 2, 1, 0 are used to receive sum-up messages as defined by the appliance. In the case of the recorder, the bit #0 is used while the bits #1, 2, 3 and 7 are always 0.
The bit #0 contains the sum-up message of the alarm state register (see the detail of these bits in the description of this register). Its state specifies whether one or several authorized events showed up in the alarm state register after its latest cleaning.

Validation register:

A state word is associated with a validation register, which makes it possible to control the service request by authorizing only specific cases.
When a bit is 1, it allows that the state 1 of the bit of same rank in the state register (STB) leads to the activation of the bit #6 in the same state register.
Writing into the validation octet is made with the *SRE<NRF> command, where <NRF> is the sum of the binary values of the bits 0 to 5 and 7.
Reading the validation octet is made with the instruction *SRE?. The answer is given in decimal format (NR1).

2.3. Standard events register

See the overview of the structures of the state data.
The structure of the standard events register is assigned to the bit #5 of the service request register.

State register:
This register contains a few specific messages with the following meanings.
You can read its content with the *ESR? command.
Reading leads to the erasing of the register.
The bits of the events state register are assigned to specific events:

- BIT 7: POWER-UP (Value 128)
  Shows that the appliance is energized.
- BIT 6: USE REQUEST (Value 64)
  Not used, positioned at 0
- BIT 5: INSTRUCTION MISTAKE (Value 32)
  Specifies that an unknown or incorrect instruction has been sent to the recorder.
- BIT 4: EXECUTION MISTAKE (Value 16)
  Not used, positioned at 0
- BIT 3: APPLIANCE-CONDITIONED MISTAKE (Value 8)
  Not used, positioned at 0
- BIT 2: INTERROGATION MISTAKE (Value 4)
  Specifies that the output queue is full and some data is or may be lost.
- BIT 1: CONTROL REQUEST (Value 2)
  Not used, positioned at 0
- BIT 0: FINISHED OPERATION (Value 0)
  Not used, positioned at 0.

An event is authorized is the corresponding bit in the event validation register is 1.

Validation register:
It makes it possible to control the standard events state register:
When a bit in this register is 1, it makes it possible that the state 1 of the bit of same rank in the standard events state register leads to the switch to 1 of the **bit #5** of the service request state register (STB).
Writing into this register is made with the *ESE<NRF> command, where <NRF> is the sum of the binary values of the bits inside the validation register.
Reading this register is made with the "*ESE?" command.

2.4. Alarms register

See the overview of the structures of state data.
The structure of alarm registers is assigned to the bit #0 of the service request register.

State register:
This register contains a few specific messages to the recorder with the following meanings.
You can read its content with the SRQ_TYPE ? command
Reading the register leads to the erasing of its content.

The bits of the alarms state register are assigned to specific events:

- BIT 7: TRIGGER OF MEMORY ACQUISITION (Value 128)
  Specifies that the triggering condition of a data acquisition into memory has been achieved.
- BIT 6: END OF MEMORY ACQUISITION (Value 64)
  Specifies that a data acquisition into memory has ended.
- BIT 5: START OF MEMORY ACQUISITION (Value 32)
  Specifies that a data acquisition into memory has started.
- BIT 4: Not used (Value 16)
- BIT 3: END OF PAPER (Value 8)
  Specifies that there is no more paper in the printer.
- BIT 2: END OF WRITING (Value 4)
  Specifies that a writing process has ended: cartridge, programmed text with the instruction WRIte (cf. programming dictionary)...
- BIT 1: END OF PLOT (Value 2)
  Specifies that a printing process has ended.
- BIT 0: START OF PLOT (Value 1)
  Specifies that a printing process has started.

An event is authorized only if the corresponding bit in the event validation register is 1.

Validation register:
It makes it possible to control the alarms state register:
When a bit in this register is 1, it makes it possible that the state 1 of the bit of same rank in the alarms state register leads to the switching to 1 of the **bit #0** of the service request state register (STB).
Writing into this register is made with the *SRQ_ENABLE <NRF> command, where <NRF> is the sum of the binary values of the bits of the validation register.
Reading this register is made with the "SRQ_ENABLE ?" command.

2.5. Using the structure of state data

Before any use, it is advisable to send the recorder the instruction *CLS that clears all state registers.
You should first determine which events you would like to detect by authorizing them in the validation registers:

- with the "SRQ_ENABLE n" command for events associated to the alarm registers
- with the "*ESE n" command for events associated with standard events registers
- with the "*SRE n" command for events associated with the service request register.

Example:

Programming a service request for a start of end of paper printing, an instruction mistake, the presence of data at the output of the recorder, is made with the commands:

SRQ_ENABLE 3	(Bits 0 and 1 switch to 1)
*ESE 32	(Bit 5 switches to 1)
*SRE 49	(Bits 0, 4 and 5 switch to 1)

In RS232 mode, the controller must regularly read the service request register with the "*STB?" command. Switching the bit #6 (MSS) to 1 shows that an authorized event happened.
When read, the word of state makes it possible to determine the type of event that happened. In the case of a standard or specific event, you must read the associated state register with the "*ESR?" or "SRQ_TYPE ?" command to precisely know the event.

A standard event happened: The user sends the "*ESR?" command:
Answer by the recorder: 160 (Bits 7 and 5 switch to 1)
Two events are displayed (energizing and instruction mistake); the instruction mistake (only event authorized in the validation register) triggered the service request.

Programming dictionary

In the following tables, sending the lower case characters of the headers and parameters is facultative.
As a rule, the digital parameters are integers (NR1); where it is specified "decimal" can be of NR1, NR2 or NR3 type.

1.1. Configuration

HEADER	PARAMETERS	EXAMPLES
MODE	P1 Definition of the mode of use of the appliance P1= DIRect, MEMory, FILE, GONOgo, POWer	MODE FILE
MODE ?	Returns the mode	
PAGe	P1 Displays a screen P1= SETUP : Config CHAN : channel N (see command: CHAN) TRigger : trigger SCOpe : direct display CHArt : paper REPLay : Replay	:CHAN A3 ;:SCREEN CHAN Display of the channel A3
ALArm	P1 Definition of the alarm to change P1= A, B or C	ALARM :VAL A,TR ;TR :CH A1,S1ED GEP The trigger is defined by the TRig command: (see 15.5.8)
ALArm :DEF	P1 P2=NO, Trigger, RECtr or ERRor	
ALArm ?	Display of the alarms	
DATe	P1, P2, P3 changes the current date P1= day (1 to 31) P2= month (1 to 12) P3= year (0 to 99)	DATE 11,12,14 December 11th 2014
DATe ?	Displays the date	
HOUrs	P1, P2, P3 Definition of the current time P1= hour (0 to 23) P2= minutes (0 to 59) P3= second (0 to 59)	HOURS 10, 6, 0 10 hours 6 minutes
HOUrs	Displays the time	
RECAII	P1 recovering a set-up P1= name of the set-up	RECALL “foldercnf/File1” Recover s the set-up
STOre	P1 saving a set-up P1= name of the set-up	:STORE “Conf 2” save the set-up into the file #2 with the name « Conf 2 »
READSETup	Recovering the current set-up in binary format; the appliance sends 4 octets specifying the number of octets and 2 octets specifying the checksum to send and then the setup file : N bytes (N=74600)	
SENDSETup	Send a set-up in binary format: 4 bytes specifying the length of the file and 2 bytes specifying the checksum of the set-up (format little endian)	SENDSET 68 23 01 00 AB 23 00012368= 74600 bytes 23AB = checksum

		add the file
CAPtion	Writing the set-up on paper (cartridge)	
KEYBLock	P1 locking the keyboard (ON or OFF)	

2.6. Parameters of the channels

HEADER	PARAMETERS	EXAMPLES
CHannel	P1 defines the CHANNEL input to change with commands P1= selection of the input A1, A2 etc.	CHAN B3 We selected to change the channel 3 of the card B
Channel ?	Displays the number of the selected input and its value	
VALID	P1, P2 defines the authorization status of each channel P1= ALL for all channels or A1, A2 etc. for each channel LOG for logic channels P2= ON or OFF	VALID ALL,OFF ;VALID A1 ON ; VALID LOG, ON We authorized the channel A1 and the logic channels only
VALID ?	Displays the validity of all channels	
NAMe	P1 changes the name of the CHANNEL input P1= name (26 characters max.) between two ' or "	CHAN B3 ; NAM'four1'
NAMe ?	Displays the name of the channel	
TYPe :VOLtage	P1 changes the channel under voltage P1= DC, RMS DVDT SVDT	TYPE:THERM K, COMP Use of a balanced thermocouple K
TYPe :SHUNT	P1, P2 changes the SHUNT channel P1= DC or RMS P2= S1M, S10M, S01, S1, S10, S50 (pour 1mΩ, ... 50 Ω)	
TYPe :FREQ	Change of the channel under FREQUENCEMETRE	
TYPe :PT100	P1, P2 change of the type of channel for PT100 P1= W2 or W3 for 2 wires or 3 wires P2= Resistance value (in 1/10 Ω)	
TYPe :THErmo	P1, P2 change of the type of channel for Thermocouple P1= Thermocouple= J, K, T, S, B, E, N, W	
TYPe:Gauge	P1, P2, P3 change of the type of channel for constraint gauge P1= HALF, FULL P2= 2 V or 5 V P3= Coefficient (from 1.8 to 2.2)	
TYPe:INTEGRE	P1,P2 change of the type integral or derivative P1= value of the caliber of the channel (en V) P2= integration period (in seconds)	

HEADER	PARAMETERS	EXAMPLES
TYPe :COUNTer	P1 change of the type of channel for counter P1= decision threshold (in V)	TYP :COUNT 1.4 The command initializes the counter to 0
TYPe ?	Displays the type of channel	
UNIt	P1 Temperature unit for thermocouple and PT100 P1: CEL, FAR, KEL	UNIT CEL Unit: Celsius degrees
UNIt ?	Displays the temperature unit of the channel	
FILter	P1,P2 definition of the filter of the channel as defined with the CHANNEL command P1= WOUT, F10KHz, F1KHz, F100Hz,NUM P2 = value on the numeric value	FILTER F10Hz,1 FILTER NUM,50
FILter ?	Displays the filter of the selected input	
RANge	P1, P2, P3 changes the caliber and the center of the input :CHAN P1= caliber in ISO units (Volts or °C) in real time P2= center in ISO units in real time P3= position in percentage	RANGE 12, 3, 0 caliber = 12 V center on 3 V
RANge ?	Displays the caliber and the center of the selected input	
THREshold	P1, P2, P3 definition of thresholds P1= SI or S2 P2= ON or OFF (validity of the plot) P3= value of the threshold	:THRES S1, ON, 10 Threshold S1 is worth 10 V
THREshold ?	Displays the values of the 2 thresholds	

Recovery of the instant values:

HEADER	PARAMETERS	EXAMPLES
RDC ?	Sends the values of all channels and the logic channels or the parameters in network analysis	

2.7. Paper (8460)

HEADER	PARAMETERS	EXAMPLES
DIRECTPLOT	P1 definition of the transcription mode on paper in direct mode direct P1= FT, TEXTe	DIRECTPLOT FT We selected the F(t) real time mode
DIRECTPLOT ?	Displays the paper mode	
SPEed	P1, P2 definition of the paper scroll speed P1= Value - 1,2,5,10,20,25,50,100,200 P2= units : - MM_S (mm/s) - MM_M (mm/min) - MM_H (mm/h)	SPEED 10,MM_S Speed : 10 mm/s
SPEed :LOGEXT	P1: number of impulsions / mm	
SPEed ?	Displays the state of SPEED or SPEED :EXT command	
BASESPeed :NONe	Basis speed zero	BASESP :SPE 1, MM_H Basis speed 1mm/h
BASESPeed :SPEed	P1, P2 modifies the current basis speed P1= value (see speed) P2= unit (see SPEED)	
BASESPeed ?	Displays the basis speed	
TEXTSpeed	P1, P2 defines the period for paper in text mode P1 varies from 1 to 500 P2 = Sec or Min or HOurs	TEXTSPEED 2,SEC
TEXTSpeed :EXT	Defines the external paper scroll speed	
TEXTSpeed ?	Displays the period in text mode	
GRATicule	P1, P2 defines the reticule on the paper P1= WOUT, G5, G10 or DIV defines the type of reticule P2= Fine or Coarse	GRAT G5,C
GRATicule ?	Displays the reticule	

HEADER	PARAMETERS	EXAMPLES
CHART :TITle	P1 definition of the title of acquisition P1= header message	CHART :TITLE « OVEN 12 »
CHART :TITle ?	Displays the title	
CHART :DATe	P1 definition of the type of date on paper P1= ABSolute or RELative	CHART :DAT ABS
CHART :DATe ?	Displays the command	
CHART :BOUndary P1	P1 defines whether the limits are printed at the end of the plot P1= WITH or WOUT	CHART :BOU WITH Printing of the limits
CHART :BOUndary ?	Displays the command	
ANNOte	P1, P2 Definition of the annotation mode P1= WOUT, START, ALarm or LENght P2 number of the alarm (1 to 3) or length of the paper	ANNOT LEN,20 Annotation every 20 cm
ANNOte ?	Displays the command	
ANNOte :TYpe	P1, P2, P3 Writing the names of the channels P1= NONAME or NAME Writing the names of the channels P2= NONUMber, NUMber Writing the numbers of the channels P3= NO, VALue, RANge, SCAle MINmax definition of the type of annotation to write	ANNOT :Type NAME,NUM,VALUE
ANNOte :TYpe ?	Displays the command	
ANNOTe:BMP	P1 Typing the BMP file P1 = WOUT or WITH	You can exchange the file with ftp by using the same name

2.8. Triggers

HEADER	PARAMETERS	EXAMPLES
START :MANual	Manual triggering (stop or start)	SEQ :MANUAL
START :TRIG	Triggering with a combination of thresholds (see 7.3)	start :trig;:trig :chan A1, S1, POS
START :WAIt	P1,P2, P3 Triggering according to a delay P1= number of hours waiting (0 to 23) P2, P3= minutes, seconds (0 to 59)	START :WAIT 0, 2, 10 waiting 2 min 10 s
START :DATe	P1, P2, P3, P4, P5, P6 Triggering with a date P1= day (1 to 31) P2= month (1 to 12) P3= year (0 to 99) P4= hour (0 to 23) P5, P6= minute ? second (0 to 59)	SEQ START ;SEQ :DATE 3, 10,06,15,30,10 Start on 3/10/06 at 15:30:10
START :AUTO	Automatic triggering (except in DIRECT mode)	
START ?	Displays the initial command	
STOP :MANual	Manual stop (direct mode)	
STOP :TRIG	Triggering with a combination of thresholds (see 7.3)	
STOP :WAIt	P1, P2, P3 Triggering according to a delay (see START :WAIt) Only in DIRECT mode	
STOP :DATe	P1, P2, P3 Triggering with a date Only in DIRECT mode	
STOP :LENGth	P1 End of triggering on a length of plot (only in DIRECT mode) P1= Length of plot x 10 cm	
STOP :AUTO	Automatic stop (memory or file mode)	
STOP ?	Displays the command of end of acquisition	

HEADER	PARAMETERS	EXAMPLES
TRIG :TYP	P1 defines the type of general trigger P1= EDGE or LEVEL	
TRIG :LOG P1	P1 Selection of trigger on the logic channels P1= defines the 16 trigger values; add a message delimiter (quotation marks)	TRIG :LOG « XXXXXXXXXXXXXXX1 » Trigger on logic channel VL1
TRIG :Chan P1, P2, P3	P1= number of the channel (A1, A2, etc.) P2= threshold (S1 or S2) P3= POS or NEG For rising or falling edge	TR :CH A1,S1,EDGEP Trigger on the rising edge of the channel A1 (threshold 1)
TRIG :Com P1	Selection of the type of complex trigger P1= OR, AND ou DELta are: • one threshold (OR) • all thresholds (AND) • slope (DELta)	TRIG :CO DEL;CO :DEL 2,S ;RESET;ADD A1,S1,POS;ADD A2,S1,NEG There are 2 thresholds (S1 on A1 and S1 on A2)
TRIG:COm:DELta P1, P2	Selection of the slope P1= value (1 to 500) P2= Sec or MIN or HOURS	
TRIG:COm:REset	ON removes all channels	
TRIG:COm:ADD P1, P2,P3	Adds a threshold to the trigger P1= number of the channel (A1, A2, etc.) P2= threshold (S1 or S2) P3= POS or NEG For a rising or a falling edge	
TRIG ?	Displays the value of the selected trigger	

The programmed trigger depends on the latest sent command (alarm, start/stop trigger, etc.)

2.9. Memory Mode

HEADER	PARAMETERS	EXAMPLES
MEMSpeed	P1, P2 Definition of the sampling period P1= period (1 to 500) P2= MICro, Mlli, Sec, Min, HOur is the unit	MEMSPEED 10,MICRO 10 µs period
MEMSpeed:EXT	Use of an external clock	
MEMSpeed ?	Displays the acquisition speed	
MEMBloc	P1 Definition of the number of blocks P1= 1, 2, 4, 8, 16... 128	MEMBLOC 4 4 blocks
MEMBloc ?	Displays the number of blocks and the validation status of each block	
POSTrig	P1, P2 Definition of the triggering position in the acquisition P1= varies between -100% and +100% P2= ON or OFF: inhibition of the trigger during the pre-triggering phase	:STOP:AUTO;POSTRIG 0 Acquisition after triggering
POSTrig ?	Displays the triggering position	
MEM:CONT	P1, P2 Definition of the sequel P1= Plot, NOPlot: plot P2= File, NOFile: save to a file	
MEM:CONT ?	Displays the sequel	
FILE:NAMe	P1, P2 Name of the save file P1= BINary only P2 : name of the file (20 characters max.)	:FILE :NAME BIN, " FileO" :FILE:LENG 10,MS
FILE :NAMe ?	Displays the name of the save file	
FILE:LENGth	P1, P2 Restriction of the number of samples P1= de 0 à 1000 (0 = no limit) P2= KSample or MSample or GSample	
FILE:LENGth ?	Displays the limitation of the length of file	
CONVERttext	P1 convert the present file or memory in a texte file P1 : NAME of text file The file will be available in the same folder than the Binary file (.rec)	CONVERT "FILE_CONV.csv"
CONVERttext ?	Return the percentage of convertion Or STOP if finished	

2.10. Reloading, real time saving

HEADER	PARAMETERS	EXAMPLES
REARm	P1 Definition of manual reloading P1= SINgle, AUTo, SETup	:REARm SINGLE
REARm:SETup	P1 : name of the setup file	:REARM SETUP;:REARM:SETUP "file1" Go to set-up file1
REARm ?	Displays the type of loading	
SAVE	P1 Real time record P1= NO, DISk or MEMORy NO: no record DISk: record to DD (hard disk) or USBKEY MEMORy: only in DIRECT mode	SAVE DISK
SAVE ?		
SAVE:MEM	P1, P2 Definition of the trigger for saving into memory in DIRECT mode P1= DIRect ? TRIG or MANual P2= CONt, NOCont reloading	SAVE MEM;SAVE:MEM TRIG,NOC;:TRIG:CHAN A2,S1,POS
SAVE:MEM ?		

2.11. Launching plot and acquisitions

HEADER	PARAMETERS	EXAMPLES
RECord	P1 Start or stop of the plot (or of the memory acquisition) P1= ON : launching OFF : stop TRIG : forcing the triggering TRIGREC : forcing the triggering of composite acquisition	RECORD ON In direct mode, the plot will be effective once the start condition is effective; you can force the triggering with RECORD TRIG and the stop with RECORD OFF.
RECord ?	Displays the state of the command and the percentage of memory acquisition	
WRIte	P1 Writing a message on paper In case of record on disk, allows an annotation P1= message (max. 93 characters) between quotation marks (") or apostrophes (')	WRITE 'RECORDER'
LINE	Draws a vertical line	
TEXT	P1, P2 Writing an horizontal text (max. 50 characters) P1= Position between 0 and 252 mm P2= text	TEXT 252, « High position »

2.12. Diagrams

HEADER	PARAMETERS	EXAMPLES
GRID	P1, P2 Definition of the diagrams P1= number of diagrams P2= SEPOLOGON or SEPLOGOFF: separated logic channels	GRID:LOG 50,5,UP;:GRID 2,SEPLOGON The logic channels are on top with 50 mm height, with 2 100-mm screens
GRID ?	Displays the definition of all diagrams	
GRID:LOG	P1, P2, P3 Definition of the diagrams for the logic channels P1= number of logic channels P2= height of the logic channels P3= UP or DOWN: position of the logic channels	
GRID:LOG ?	Displays the definition of all diagrams	
GRID:LENGth	P1, P2, P3 Définition of all diagrams P1= number of the diagram P2= min. value (0 to max.), max. is 250 or 200 according to the appliance P3= max. value (0 to max.)	GRID:LENG 1,0,100 Diagram 1 from 0 to 100 mm
GRID:LENGth ?	Displays the definition of all diagrams	
GRID:CHAnnel	P1, P2, P3 Definition of the position of a channel P1= number of the channel P2= number of the diagram: 1 to max. P3= plot thickness: 1 to 8	GRD:CHA A4,3,2 Channel A4 in the diagram 3 with a plot thickness of 2
GRID:CHAnnel ?	Displays the definition of the selected channel	
COLOR	P1, P2, P3 Color of each channel P1= value for red (0 to 100) P2= value for green P3= value for blue	CHAN A2,COLOR 100,100,100
DEFLOG	P1, P2, P3, P4, P5 Definition of the logic channels P1= number of the logic channel P2= value for red (0 to 100) P2= value for green P4= value for blue	

2.13. Direct display

HEADER	PARAMETERS	EXAMPLES
SCREEN	P1 Definition of the visualization mode P1 = FT, TEXT or XY	SCREEN FT
SCREEN:FT	P1, P2, P3 Definition under F(t) P1= VER or HOR for vertical or horizontal P2= BOUNON or BOUNOFF to display the limits P3= FULLON or FULLOFF to display full screen or not	PAGE SCOPE;SCREEN FT;:SCREEN VER,BOUNON,FULLON Vertical full screen display
SCREEN:XY	P1, P2 Definition under XY P1= channel X of A1, A2, etc. P2= channel Y becomes ALL for all channels ON or A1, A2 for only one channel P3= DOT, VECTor	SCREEN:XY A3,A2,DOT
SCREEN:TIMEBASE	P1,P2 Definition for the time base in Scope mode P1= value (1 to 500) P2= MICRO, MILlisec, Sec, Mln or HOurs	SCOPE:TIMEBASE 500,MS;:SCREEN FT;:PAGE SCOPE;:SCOPE:RESTART
SCREEN:RUN	P1 Start or stop the Scope mode P1= ON or OFF Stop the Scope mode	We change the time base, then we display the scope f(t) screen
SCREEN:RUN ?	Displays the Scope mode	
SCREEN:TRIG	P1, P2, P3, P4 Trigger in f(t) mode for quick speeds P1= number of the channel P2= POS or NEG P3= level (0-100) P4= position (0-100)	

2.14. Mathematical functions

HEADER	PARAMETERS	EXAMPLES
MATH	P1 Number of mathematical functions (0 to 5) P1 = FT, TEXT or XY	MATH 3
MATHDEF	P1, P2, P3 Definition of a function P1= number of the function P2= used channel P3= function MIN MAX PK_PK LOW HIGH AMPL P_OVERSH N_OVERSH FREQ PERIOD R_EDGE F_EDGE P_WIDHT N_WIDTH P_DUTTY_CYCLE N_DUTTY_CYCLE MEAN MEAN_CYC RMS RMS_CYC	MATHDEF 1,A1,MIN
MATH ?	Reading of the function values ON must be in visualization f(t) mode to get the values	

2.15. Replay

HEADER	PARAMETERS	EXAMPLES
OUTBloc	P1, P2, P3 Definition of the block and the output window P1= 1 to 128 block numbers P2= 0 to 100 (real percentage of the start) P3= 0 to 100 (real percentage of the end)	OUTBLOC 1,25.2,80 block 1 ? start at 25.2% and end at 80%
OUTBloc ?	Displays the command	
OUT:REC	P1, P2 Definition of the type of paper output P1= FT or XY type of output P2= defines the reduction rate in output under mode F(t) (from 1 up to 10000 with increments of 1,2,5) or the width of the reticule under XY mode	OUT:REC XY,200 Diagram XY 200x200 on paper OUT:REC FT,100 Mode F(t) 100 samples per mm
OUT:REC ?	Displays the command	
PLOTRec	P1 Starts or stops the plot on screen P1= ON or OFF	
POLTRec ?	Displays the plot and the written percentage	
DEFPACQ	P1, P2 P1 : number of packets to send P2 : number of octets in the packet	OUTBLOC 2,0,100;:DEFPACQ 0,50000 ;READPACK ? Recovery of the 1st packet of the block #3 :DEFBLOC 1,50000;READBLOC ? Recovery of the sequel
READPACK ?	Reading the packet defined by DEFPACQ in binary format *4 octets: length of the data packet *4 octets: number of the received packet *4 octets: checksum of the data packet *4 octets: length of the total file (for the packet #0 only)	

2.16. Additionnals channels

HEADER	PARAMETERS	EXAMPLES
VALIDEXT	P1 : Validity of additional channel ON , OFF	VALIDEXT ON; NBEXT 4; CHAN G1; ONOFF ON; RANGE 100,0,0 VALEXT G1,40 : add 4 channels and give the value 40V to the channel G1
NBEXT	P1 : number of externals channels From G1 to Gx	
VALEXT	P1,P2 P1 : channels (G1 to Gx) P2=value of the channels P1	

The name of additionnals is from G1 to Gx
The number total of channels (analogics and additionnals channels would be <=72)

it’s possible to record , print this channels as an analogic channels.

2.17. Service request

Refers to the explanations about the structure of the state data.

HEADER	PARAMETERS	EXAMPLES
SRQ_ENABLE	P1 Changes the alarm validation register P1= value of the register bit decimal value use 0 1 start of plot 1 2 end of plot 2 4 end of writing 3 8 end of paper 4 16 open table 5 32 start of acquisition 6 64 end of acquisition 7 128 trigger acquisition	SRQ_ENABLE 3 3 = 1 + 2, which means bits #0 and #1 The beginning and the end of the plot are signaled on the service request register
SRQ_ENABLE ?	Displays the value of the alarm validation register	
SRQ_TYPE ?	Displays the value of the alarm state register Then, the register is erased The definition of each bit is the same as for SRQ_ENABLE	SRQ_TYPE ? The recorder displays :SRQ_TYPE 4 which means « one finished writing operation »

3. Error messages

In case of trouble with the programming through the recorder interface, a debugging window shows up on screen to help you identify your mistake:

# error	Explanation
1	Unknown header
2	Unknown parameter
3	Forbidden parameter
4	Absent parameter
5	Wrong parameter separator
6	Wrong message separator
7	Too long word
8	Wrong format for text parameter
9	Forbidden interrogation
10	Digital parameter out of range
11	Text parameter out of range
12	Compulsory interrogation
13	Emission buffer full
14	Impossible in this context
15	Checksum error

At each error matches a line specifying:

- a mistake number
- the received message

When the window is full, the mistakes are displayed from the first line.
The last error line is followed by a blank page.