

# Command Manual

for  
INTAB

PC-loggers

Copyright (c) INTAB Interface-Teknik AB,  
July 2003



# MANUALS of the PC-logger System:

- <b>How to Use 3100 Hardware</b> "	How to connect sensors and transducers correctly
- <b>How to Use 3150 Hardware</b> "	How to connect sensors and transducers correctly
- <b>User's Manual PC-Logger 2100"</b>	How to connect sensors and transducers correctly
- <b>EasyView</b> "	A tutorial guide to a Windows based user-friendly expert data analysis program
- <b>Command Manual</b> "	Describes the ASCII strings (used by software) that make the PC-logger tick



# Table of contents

---

---

Table of contents.....	v
General information.....	1
About this manual.....	1
Communications standard.....	1
Command Syntax Notation.....	2
Commands grouped by function.....	3
ONLINE Measurement.....	3
OFFLINE Measurement.....	3
Data retrieval/identification.....	3
Configuration.....	3
Communications and special.....	4
Control program commands.....	4
Input/Output commands.....	4
Clock commands.....	4
Commands in Alphabetical order.....	5
ANSWER.....	5
ANSWER.....	5
AOUT.....	5
BOOT.....	6
CF.....	6
CHECK.....	6
CONFIG.....	7
CONFIG.....	8
COUNTER.....	9
COUNTMODE.....	10
COUNTMODE.....	10
CT.....	11
CUSTOM.....	11
DATA.....	12
DATE.....	13
DELAY.....	13
DIALRING.....	14
DIGINP.....	15
DIGOUT.....	15
DIGOUT.....	16
EXTTRIG.....	17
GUARD.....	17
HANGUP.....	17
HERTZ.....	18
INFO.....	19
INITANSWER.....	20
INITANSWER.....	20
INTERVAL.....	20
MEMSIZE.....	21
MEMTYPE.....	21
MODE.....	22
NAME.....	22
OFF.....	22
OFF.....	23
OPEN.....	23
PAUSE.....	24
PAUSECOUNT.....	24
PROG.....	25
QUIT.....	25

RADIO .....	26
REDUCE .....	26
REPEAT .....	27
RING .....	27
SEND .....	28
SET .....	28
START .....	29
STARTDATE.....	29
STARTTIME .....	29
STOP .....	30
STORE .....	30
TERMCHAR.....	30
TERMO.....	31
TIME .....	31
TITLE.....	31
TRIG.....	32
TRIGLEVEL.....	32
TRIGWINDOW .....	32
VERSION.....	32
Programming example in GW-BASIC .....	34

---



---

---

---

## General information

---

### About this manual

---

This manual is produced exclusively for users who write their own measuring/logging software. It describes the commands the INTAB PC-loggers like the AAC-2 and the "2100" will recognise. Most of them are used by the "**AAC-2 PC Soft**" logger software that runs under MS-DOS.

The "**AAC-2 PC Soft**" should cover 99% of all logging needs, but occasionally requirements are so specialised that customised software is a must. That is where this manual has its given place!

The "**AAC-2 PC Soft**" has a feature that enables the programmer to try these commands if in doubt about how they work. Its "*terminal emulator*" will let the programmer transmit ASCII character strings (by simply typing them) to the AAC-2 and then read the response (if any).

---

### Communications standard

---

The PC-Logger's uses RS-232 with 19200 baud, 8 data bits, no parity and 1 stop bit. Older models may run on 9600 baud.

Communication with a PC-Logger is straightforward, the format is straight ASCII strings, followed by a termination character ( CR ).

The PC-Logger's response string termination sequence can be changed with the **TERMCHAR:** command.

Commands that the AAC cannot understand or perform result in the PC-Logger sending an "ERR" string.

Commands that sets parameters respond with an "OK" string if the command was successful.

Commands that contain a ? performs some sort of query, in which case the PC-Logger will respond by sending one or more strings as a reply, depending on which command it received.

The PC-Logger's are equipped with a mechanism that causes its power consumption to be reduced to a very low level in order to extend the duration of the internal rechargeable batteries. If the PC-Logger hasn't received any commands for the last 2 minutes, it will turn itself off. Because of this, the communication output lines from the PC-Logger may be unpowered from time to time. This implies that special considerations must be taken into account when you intend to communicate with a PC-Logger.

Since there are several different models of the PC-Loggers, the commands listed in this manual may or may not work with your model of the PC-Logger. Therefore, we have placed a list of one letter codes following the command headings, to inform you of which models support a particular command.

The AAC-2 uses the character **2**, the AAC-2 with software switchable input ranges uses the character **3** or **AAC-3**, the AAC-2f and the 3150 use the character **F** and AAC-2's with a ROM-version number greater than 5.00 uses the character **5**.

Units with higher baudrate = 19600 and 3100 are given the number **4** as designator.

The "**2100**" has been given the letter **M**

## **Command Syntax Notation**

---

The syntax of all commands have been represented (or noted) in a generally accepted format.

This means in short that :

- Brackets are not part of any commands or arguments. They are included for clarity.
- Square brackets [ ] separate argument strings.
- Normal brackets ( ) surround optional arguments.
- Forward slashes / separate alternative arguments or responses.
- Backslash \ separates two ASCII (control)characters in one argument. Only included for clarity.
- Commas , separate arguments in the same string. They must be included in the command.
- A colon : must be included in all commands. It separates the command from its arguments.

# Commands grouped by function

---

## ONLINE Measurement

---

SEND: Orders the logger to measure and send channel values.  
REPEAT: Starts measurement and sends channel values at set intervals.  
QUIT: Cancels a REPEAT: command.

## OFFLINE Measurement

---

CT: Shortform for CUSTOM: (see below)  
CUSTOM: Linear X-form data  
EXTTRIG: Controls measurement trig source.  
INTERVAL: Controls the measurement interval.  
MEMTYPE: Controls how data is stored.  
PAUSE Data storage can be inhibited by digital input.  
PAUSECOUNT: Inhibits storage of data when counter-channel reads low counts.  
REDUCE: Controls data reduction.  
START: Starts recordings.  
STOP: Stops recordings.  
STORE: Selects channels to use when measuring "off line".  
TERMO: Selects thermocouple used for temperature channels.  
TRIG: Controls start and stop triggers for AAC-2f units.  
TRIGLEVEL: Controls triglevels for AAC-2f units.  
TRIGWINDOW: Specifies post and pre trig duration of "f"-units

## Data retrieval/identification

---

CT: Shortform for CUSTOM: (see below)  
CUSTOM: Linear X-form data  
DATA: Activates data retrieval mode.  
NAME: Channel naming/identification support.  
STARTDATE: Reads the timestamp of the first recorded set of data.  
STARTTIME: Reads the timestamp of the first recorded set of data.  
TITLE: Title or name support for recording.

## Configuration

---

CF: Shortform for CONFIG: (see below)  
CONFIG: Gives information about or controls channel configuration.  
COUNTMODE: Controls how the counter channels are to be used.  
INFO: Gives information about of an AAC-3.  
MODE: Controls a PC-Logger's mode of operation.  
TERMCHAR: Set communication termination sequence.  
VERSION: Returns version no etc. of the PC-Logger.

---

## Communications and special

ANSWER:	Command string for modem to answer.
DIAL:	Controls AAC-2 initiated modem communications.
DIALRING:	Limits alarm initiated DIALs
GUARD:	Manages guard ( event ) support.
HANGUP	Terminates modem communication.
INITANSWER:	Modem initiation string at "ring".
OPEN:	Controls the RS-422 addressing.
OFF:	Turns off the power = sleep mode.
RADIO:	Provides guard-based relay operation.
RING:	Determines the number of ring signals that must occur before an AAC-2 lifts the handle ( answers ).

---

## Control program commands

DELAY:	Timing control.
PROG:	Control program management.
DIGIN:	Reads digital inputs directly ( channel 31 ).
DIGOUT:	Controls digital outputs directly ( channel 32 ).
COUNTER:	Reads counters directly ( channels 25-30 ).
STORE:	Performs measurement and storage of data.

---

## Input/Output commands

AOUT:	Sets analog output levels
COUNTER:	Reads counter channels.
DIGINP:	Reads digital inputs.
DIGOUT:	Controls digital outputs.
SEND:	Measures and sends selected channels.

---

## Clock commands

DATE:	Sets/Gets the real time clock date.
TIME:	Sets/Gets the real time clock time.

## Commands in Alphabetical order

---

---

### ANSWER

(4)

Purpose	Defines answer command to be sent to MODEM.
Syntax	ANSWER:[?]/[STRING][CR\LF]  Where " <b>STRING</b> " consists of modem command characters.
Response	[STRING]/[OK][CR\LF]

---

### ANSWER

(5)

Purpose	This command is not recognised by the above model.
Syntax	Not applicable!  A "ring" will in these models result in ATA being sent to connected modem.
Response	N.A.

---

### AOUT

(4,5)

Purpose	Sets analog output levels.
Syntax	AOUT:[Val1,Val2,Val3,Val4][CR\LF] or AOUT:[n,Val][CR\LF] Where Val is a number between -2048 and +2047 and n = 1,2,3 or 4 corresponding to one of four outputs.
Response	[OK][CR\LF]

---

**BOOT****(2,3,4,5,F,M)**

Purpose	Loads controller program (= load modules : *.LDM files) into loggers.
Syntax	BOOT:[MSFF]  <b>MSFF</b> = Motorola S Format File  Must be downloaded in "binary" format as is. No extra ASCII control characters may be added.  <i>This command is only operational in mode "A".</i>
Response	N.A.

---

**CF****(3,4,M)**

Purpose	Provides channel configuration control for AAC-3:s. This is a shortform notation for CONFIG. See CONFIG: for versions 3 and 4 below for complete details.
---------	--

---

**CHECK****(4,F,M)**

Purpose	Checks system configuration. Returns code representing encountered hardware.
Syntax	CHECK:
Response	[Number][CR\LF] [OK][ERR][CR\LF]  [OK] if there is a functional analog board [ERR] if there is a hardware error [Number] is a decimal representation of the binary weight sum of codes for encountered devices : bit 0 : PIA bit 1 : A/D1 bit 2 : A/D2 bit 3 : A/D3 bit 4 : Dig. I/O bit 5 : Counters

Purpose	Retrieves channel configuration information.
Syntax 1	CONFIG:[?][CR\LF]
Response	count[CR\LF] ch,scalefactor,type[CR\LF] . . . ch,scalefactor,type[CR\LF]  <b>count</b> gives you the number of configuration strings that follow.  <b>ch</b> is the channel identification number.  <b>scalefactor</b> is the weight adjust factor that when multiplied with the value read by a <b>DATA</b> session gives you the actual value measured.  <b>type</b> is the engineering unit of the channel.

Purpose	Provides channel configuration control for AAC-2:s with software selectable ranges.
Syntax 2	<p>CONFIG:[nCh,Type,Gain(,ThermoType)][CR\LF]</p> <p>where:</p> <p><b>nCh</b> identifies the channel to configure ( <b>1-24</b> ).</p> <p><b>Type</b> is the type of input you want for the channel ( <b>U,I, T or F</b> ). If <b>T</b> or <b>F</b> (f for Fahrenheit) is selected, the <b>ThermoType</b> parameter must also be provided.</p> <p><b>Gain</b> selects the input range ( <b>1, 2, 3 or 4</b> ). Actual ranges are determined by the hardware configuration of the AAC-3, such information is acquired by the <b>INFO:</b> command.</p> <p><b>ThermoType</b> selects the thermocouple that should be used when the channels is configured for temperature measurements ( types <b>B, E, J, K, R, S, T, Y or Z</b> ).</p>
Syntax 3	<p>CONFIG:[nCh,?][CR\LF]</p> <p>where:</p> <p><b>nCh</b> identifies the channel of interest ( <b>1-32</b> ).</p> <p>The PC-Logger responds by sending a string with the following layout:</p> <p>[ nCh, WeightFactor, Unit, Gain, ThermoType ]</p> <p>The non-analogue channels 25-32 have the <b>Gain</b> field set to <b>-1</b>, and the <b>ThermoType</b> field set to <b>-</b>.</p> <p>If an analogue channel has a <b>Type</b> setting other than <b>T</b> or <b>F</b>, it's <b>ThermoType</b> field will be <b>-</b>.</p>
Syntax 4	<p>CONFIG:[?][CR\LF]</p> <p>The PC-Logger's response on this version of the CONFIG command is similar to that of the <b>CONFIG:nCh,?</b> form, but it sends information for all 32 channels.</p>

Syntax 5

CONFIG:[A,?][CR\LF]

The response on this form is a sequence of strings with the following layout:

```
[NoOfChannels]
[ nCh, Type, Gain, Thermocouple ]
.
.
.
[ nCh, Type, Gain, Thermocouple ]
```

**NoOfChannels** gives information about how many analogue channels the PC-Logger-3 is equipped with.

**nCh** identifies a channel.

**Type** tells you what kind of input the channels is set up to measure; **U**, **I**, **T** or **F** are possible values for this field.

**Gain** tells you of the channels accuracy setting; **1**, **2**, **3** or **4** are possible values for this field.

**Thermocouple** gives you information about which thermocouple type the channel is set up to use, if the **Type** field is **T** or **F**. If the channels is configured with type **U** or **I**, this field will always be a -.

Note that information about channels 25-32 are not provided by this version of the CONFIG: command.

---

**COUNTER**

(2,3,4,5)

Purpose	Reads the value of a counter channel, reset of the counter is done if the optional <b>R</b> parameter is specified.
Syntax	COUNTER:([R],[Rn])[CR\LF]
Response	[ValueOfCounter][CR\LF]

---

**COUNTMODE****(2,3,4,5)**

Purpose	Controls the behaviour and source of the counter channels values.
Syntax	<p>COUNTMODE:[?]/[m,m,m,m,m,m][CR\LF]</p> <p>where <b>m</b> is one of the following: ( the first <b>m</b> corresponds to counter no 1/channel 25, the last to counter no 6/channel 30)</p> <p><b>R</b> Read counter with reset after read. <b>A</b> Read counter with reset after store. <b>N</b> Read counter without reset ( accumulating ). <b>h</b> Read clock hours (00-23). <b>m</b> Read clock minutes (00-59). <b>s</b> Read clock seconds (00-59). <b>D</b> Read clock day (01-32). <b>M</b> Read clock month (01-12). <b>Y</b> Read clock year (00-99).</p>
Response	[m,m,m,m,m,m]/[OK][CR\LF]

---

**COUNTMODE****(M)**

Purpose	Controls the behaviour and source of the counter channels values.
Syntax	<p>COUNTMODE:[?]/[m,m,m][CR\LF]</p> <p>where <b>m</b> is one of the following: ( the first <b>m</b> corresponds to counter no 1/channel 25, the last to counter no 3/channel 27)</p> <p><b>R</b> Read counter with reset after read. <b>A</b> Read counter with reset after store. <b>N</b> Read counter without reset ( accumulating ). <b>h</b> Read clock hours (00-23). <b>m</b> Read clock minutes (00-59). <b>s</b> Read clock seconds (00-59). <b>D</b> Read clock day (01-32). <b>M</b> Read clock month (01-12). <b>Y</b> Read clock year (00-99).</p>
Response	[m,m,m]/[OK][CR\LF]

---

**CT****(3,4,F,M)**

Purpose	Storage area for linear x-form parameters.
Syntax	CT:[nCh,uuuu,X2,Y2,X1,Y1][CR\LF] or CT:[nCh,?][CR\LF]  This is a shortform notation for CUSTOM. See CUSTOM below for complete details.  CT:[C][CR\LF] will clear all custom settings.
Response	[OK][CR\LF] or when ?: [uuuu,X2,Y2,X1,Y1][CR\LF]

---

**CUSTOM****(2,3,4,5,F,M)**

Purpose	Storage area for linear x-form parameters.
Syntax	CUSTOM:[nCh,uuuu,X2,Y2,X1,Y1][CR\LF] or CUSTOM:[nCh,?][CR\LF]  Where <b>nCh</b> specifies channel number for x-form parameters. The unit after transformation is specified by four characters: <b>uuuu</b> . The straight line is specified by pairs of numbers: <b>X1/Y1</b> and <b>X2/Y2</b> . They are each four byte float numbers in Hex ASCII format.
Response	[OK][CR\LF] or when ?: [uuuu,X2,Y2,X1,Y1][CR\LF]

Purpose	Enters data transfer mode or returns the number of values that has been stored so far.
Syntax	DATA:[?]/[BlockSize][CR\LF]  <b>BlockSize</b> is the number data bytes in a block that you wish to use when receiving data from the PC-Logger. It has to be an even number between 10 and 1000. Other values cause an "ERR" string to be returned.
Response on <b>DATA:?</b>	The number of values that have been stored in the PC-Logger's memory. Data transfer mode is <b>not</b> activated.
Response on <b>DATA:BlockSize</b>	[No of Overwritten batches][CR\LF] and then activates data transfer mode (described below). When data transfer mode is terminated with [CAN\CAN], the PC-Logger will send an "OK" [CR\LF] string.

A dialogue to transfer data begins when the PC-program sends a **NAK** code, followed by a block sequence byte, which must be zero for the first block. The PC-Logger responds by sending a sequence block which has the following layout:

```
SYN          8 bit synch character
NUM          8 bit binary block sequence number
LEN LO      bit 1-8 of the 16 bit block length
LEN HI      bit 9-16 of the 16 bit block length
DATA LO
DATA HI
.
.          A block of LEN bytes that holds the data.
.
DATA LO
DATA HI
CHECK LO    16 bit sum of all bytes in the block except the
CHECK HI    SYN character.
```

Consecutive block are transferred when the program sends **NAK** followed by an larger value sequence byte. If an error occurs during transfer (block sequence number does not correspond to the last block sequence byte, or unequal checksums), you should repeat the last sent **NAK SequenceByte**. Data transfer mode is **terminated** by [CAN\CAN]. The PC-Logger will respond with an [OK][CR\LF] string.

---

**DATE****(2,3,4,5,F,M)**

Purpose	Sets or reads the date of the PC-Logger internal clock.
Syntax	DATE:[?]/[YY:MM:DD][CR\LF]
Response	[YY:MM:DD]/[OK][CR\LF]

---

**DELAY****(2,3,4,5,M)**

Purpose	Pause command used to delay a PC-Logger control program. ( See the PROG command ).
Syntax	DELAY:[MilliSeconds][CR\LF]
Response	Not applicable.

---

**DIAL**  
**(2,3,4,5)**

Purpose	<p>Controls modem communication <b>initiated</b> by the AAC-2.</p> <p>If this command is activated, the AAC-2 will monitor the guard time set by the GUARD: command, and the state of the Alarms. If the guard is active, or an alarm has occurred, the AAC-2 will try to establish contact with a PC through an external modem connected to the AAC-2.</p> <p>In the case of a guard call, the call should be acknowledged, otherwise the AAC-2 will continue to call for the duration of the guard time.</p>
Syntax	<p>DIAL:([?]/[1]/[0]/[ModemCmdStr])[CR\LF]</p> <p><b>DIAL:</b> acknowledges a call caused by guard and prevents further calls during the same guard time.</p> <p><b>DIAL:1</b> enables the dial function.</p> <p><b>DIAL:0</b> disables the dial function.</p> <p><b>DIAL:"ModemCmdStr"</b> set the commands that the AAC-2 will send to the modem when it tries to call.</p> <p><b>DIAL:?</b></p>
Response on ?	<p>[Enabled,Reason,ModemCmdString][CR\LF]</p> <p><b>Enabled</b> = 1 if the feature is enabled</p> <p><b>Reason</b> is 0 if guard is inactive or if no alarm condition has occurred. It will be <b>1</b> if the guardtime was active, <b>2</b> if an alarm has been triggered</p>

---

**DIALRING**  
**(4)**

Purpose	<p>Limits the number of times an alarm condition will cause a DIAL: when no acknowledge has been received.</p>
Syntax	<p>DIALRING:([?]/[n][CR\LF]</p> <p>n = 0 .. 9</p>
Response	<p>[n][CR\LF] [OK][CR\LF]</p>

---

**DIGINP**  
**(2,3,4,5,F,M)**

Purpose	Reads digital inputs.
Syntax	DIGINP:([n])[CR\LF]  where <b>n</b> is an optional number between 1 and 8, used to identify a single digital input bit.
Response with <b>n</b>	the decimal bit weight value for the digital input bit, i.e. if the digital input 3 is 'high', the value sent will be a "4".
Response without <b>n</b>	[BS][CR\LF] BS=The bit weight sum of all active inputs.

---

**DIGOUT**  
**(2,3,5,F)**

Purpose	Sets digital outputs.
Syntax	DIGOUT:[A],[0 .. 15][CR\LF] or DIGOUT:[1 .. 4],[0]/[1][CR\LF] or DIGOUT:[?][CR\LF] where:  <b>A</b> means all outputs (bitsum needed). <b>or</b> <b>1.. 4</b> is the relay number (one of four). <b>0 .. 15</b> bitsum if <b>A</b> is used. <b>or</b> <b>1/0</b> on/off if relay number is used.
Response	[OK]/[BS]/[CR\LF] BS=The bit weight sum of all active outputs.

**DIGOUT**  
**(4,M)**

---

Purpose	Sets digital outputs.
Syntax	DIGOUT:[A],[0 .. 31][CR\LF] or DIGOUT:[1 .. 8],[0]/[1][CR\LF] or DIGOUT:[?][CR\LF] where: <b>A</b> means all outputs (bitsum needed). <b>or</b> <b>1.. 8</b> is the relay number ( one of eight ). <b>0.. 31</b> bitsum if <b>A</b> is used. <b>or</b> <b>1/0</b> on/off if relay number is used.
Response	[OK]/[BS]/[CR\LF] BS=The bit weight sum of all active outputs.

---

**EXTTRIG**  
**(2,3,4,5,M)**

Purpose	Sets sampling-command source.
Syntax	EXTTRIG:[?]/[0]/[1][CR\LF]  where:  <b>0</b> is the real time clock. <b>1</b> is digital input no 1.
Response	[0]/[1]/[OK]/[ERR][CR\LF]

---

**GUARD**  
**(3,4,5)**

Purpose	Provides AAC-2 activity control.
Syntax	GUARD:[?]/[1]/[0]/[WakeUp,ToSleep]][CR\LF]  where  <b>WakeUp</b> is the wakeup time in the format <b>HH,MM,SS</b>  <b>ToSleep</b> is the turn off allowed time in the format <b>HH,MM,SS</b>  <b>GUARD:1</b> enables the guard facility. <b>GUARD:0</b> disables the guard facility.  <b>GUARD:HH,MM,SS,HH,MM,SS</b> sets the time span of the guard function.
Response	[Enabled,Active,WakeUpTime,SleepTime][CR\LF] or [OK]/[ERR][CR\LF]

---

**HANGUP**  
**(3,4,5)**

Purpose	Terminates modem communication.
Syntax	HANGUP:[CR\LF]
Response	[OK]/[ERR][CR\LF]

---

**HERTZ****(M,4)**

Purpose	Sets filter factor of A/D converter to reject either 50 or 60Hertz mains interference
Syntax	HERTZ:[50]/[60][CR\LF]
Response	[OK]/[ERR][CR\LF]

Purpose Returns hardware configuration information of the AAC-2 models with software range switching.

Syntax INFO:[?][CR\LF]

Response [NoOfGroups]  
[Group1info] ( if present )  
[Group2info] ( if present )  
[Group3info] ( if present )  
[ THERMO TYPES:ListOfTypes]  
[ DIGI/O:Digital I/OPresentFlag]  
[ COUNTERS:CountersPresentFlag]

**NoOfGroups** is either **1**, **2** or **3**.

The **GroupNInfo** fields has the following layout:

[ FullscaleGain1U, Unit, ScaleFactor ]  
[ FullscaleGain2U, Unit, ScaleFactor ]  
[ FullscaleGain3U, Unit, ScaleFactor ]  
[ FullscaleGain4U, Unit, ScaleFactor ]  
[ FullscaleGain1I, Unit, ScaleFactor ]  
[ FullscaleGain2I, Unit, ScaleFactor ]  
[ FullscaleGain3I, Unit, ScaleFactor ]  
[ FullscaleGain4I, Unit, ScaleFactor ]

**ListOfTypes** is a sequence of characters that identifies the thermocouple types that the AAC-2 supports.

**DigitalI/OPresentFlag** can be **0** or **1**, a **1** indicates that the AAC-2-3 supports digital I/O.

**CountersPresentFlag** can be **0** or **1**, a **1** indicates that the AAC-2-3 supports counters.

Output example

```
1
1000 ,mV ,1 . 0e-1
100 ,mV ,1 . 0e-2
50 ,mV ,1 . 0e-2
10 ,mV ,1 . 0e-3
20 ,mA ,1 . 0e-3
2000 ,uA ,1 . 0e-1
1000 ,uA ,1 . 0e-1
200 ,uA ,1 . 0e-2
THERMO TYPES : BEJKRSTYZ
DIGI / O : 1
COUNTERS : 0
```

---

**INITANSWER****(3)**

Purpose	Defines initialising commands to be sent to MODEM at "ring".
Syntax	INITANSWER:[?]/["STRING"]  Where " <b>STRING</b> " consists of modem command characters. Quotation marks must be included. Default string is: "ATZ!£ATE0!"
Response	[OK]/["STRING"]

---

**INITANSWER****(4)**

Purpose	Defines initialising commands to be sent to MODEM at "ring".
Syntax	INITANSWER:[?]/["STRING"]  Where " <b>STRING</b> " consists of modem command characters. Quotation marks must be included. Default string is: "ATE0!"
Response	[OK]/["STRING"]

---

**INTERVAL****(2,3,4,5,F,M)**

Purpose	Sets or reads the measuring interval for off-line recordings timed by the AAC-2 clock.
Syntax	INTERVAL:[?]/[TimingInterval]  The AAC-2 and "2100" uses the following format for <b>TimingInterval: HH:MM:SS</b> .  The AAC-2f uses a <b>1-1000</b> ms format.
Response	[OK]/[TimingInterval]

---

**MEMSIZE**  
**(2,3,4,5,F,M)**

Purpose	Returns information about how much data storage memory the AAC-2 is fitted with.
Syntax	MEMSIZE:[?][CR\LF]
Response	The AAC-2 responds with a string that may look like this:  32 kByte ram present  To calculate the amount of measurement values that can be stored, you divide this number by 2.

---

**MEMTYPE**  
**(2,3,4,5,F,M)**

Purpose	Controls or retrieves the method used when storing sampled data.
Syntax	MEMTYPE:[?]/[0/1][CR\LF]  where  <b>?</b> returns the method code.  <b>0</b> memory size is set and limited. Stops recordings when memory is filled with data.  <b>1</b> regard the memory as if it were of an unlimited size: loop back to overwrite oldest data when memory gets filled and continue to store data in a "rotating" manner.
Response	[OK]/[MethodCode][CR\LF]

		<b>MODE</b> (2,3,4,5,F,M)
Purpose	Controls the operational mode of an AAC-2.	
Syntax	MODE:[?]/[A]/[B][CR\LF]	
	<p>An <b>A</b> argument switches the AAC-2 into it's primitive mode. ( the program for this mode resides in ROM ). All normal functions of the unit are inoperational in this mode.</p> <p>A <b>B</b> argument switches the AAC-2 into it's normal mode ( a valid RAM program must be present in the AAC-2's code memory ), the AAC-2 can perform measurements in this mode.</p> <p><b>N.B.:</b>The commands listed in this manual can only be used when the AAC-2 is in <b>B</b> mode.</p>	
Response	[OK]/[ERR]/[A]/[B][CR\LF]	

		<b>NAME</b> (2,3,4,5,F,M)
Purpose	Reads or Stores channel "naming" information.	
Syntax	NAME:[nCH],[?]/[Text][CR\LF]	
	<p>where:</p> <p><b>nCH</b> is the channel identification number.</p> <p><b>Text</b> is the text that is to be assigned to the channel.</p>	
Response	[OK]/[Text][CR\LF]	

		<b>OFF</b> (2,3,5)
Purpose	Puts the AAC-2 to sleep. Power consumption is reduced to a minimum.	
Syntax	OFF:[CR\LF]	
Response	[OK]/[ERR]	
	<p>This command is not functional if the AAC-2 is recording, .</p>	

---

**OFF****(4,M)**

Purpose	Puts logger to sleep
Syntax	OFF:([0]/[1]/[?])[CR\LF] where (no arg) puts logger to "sleep" <b>0</b> disables sleep mode <b>1</b> enables sleep mode <b>?</b> returns code representing reason that OFF is not allowed: 1 - command in process 2 - logo-key active 4 - Warm Up in progress 8 - sampling in progress 16 - on line recording in progress 32 - short interv. off-line in progress 64 - PROG mode active 128 - digital output active 256 - EXTTRIG:1 active 512 - MODEM active 1024 - Alarm active 2048 - OFF:0 in effect 4096 - GUARD: in progress 8192 - DIAL: in progress
Response	[OK]/[ERR]/[Reason][CR\LF]

---

**OPEN****(2,3,4,5)**

Purpose	Addresses one of several AAC-2:s fitted with RS-422 interfaces connected to a RS-422 multi-drop line.
Syntax	OPEN:([?]/[A]/[XXXXXX])[CR\LF] where: <b>?</b> the currently 'talking' AAC-2 responds with it's serial number.  <b>A</b> All units accept commands without responding.  <b>XXXXXX</b> AAC-2 serial number - Addressed unit activated as talker, all other units are disabled. Higher order digits are only necessary if lowest digit coincides with that of other unit(s.)  (No par) All units disabled.

---

**PAUSE**  
**(4,F,M)**

Purpose	Digital input no. 1 inhibits storage of data.
Syntax	PAUSE:[0]/[1]/[2]/[?]
	Where "0" disables this function
	A "1" inhibits storage of data when digital input no. 1 is "high".
	A "2" inhibits storage of data when digital input no. 1 is "low".
Response	[OK]/[0]/[1]/[2][CR\LF]

---

**PAUSECOUNT**  
**(3,4,M)**

Purpose	Inhibit storing of data online when the pulse frequency on selected channel is below set level.
Syntax	PAUSECOUNT:[cCh,Count,Delay]/[?][CR\LF] where <b>cCh</b> is counter channel no (25 - 30) <b>Count</b> is count below which storing of data suspends <b>Delay</b> no of stored data batches after count has gone below set value (=Count)
Response	[OK]/[ERR]/[cCH,Count,Delay][CR\LF]

---

**PROG**  
**(2,3,4,5,M)**

Purpose	Enters AAC-2 "control program" entry mode or switches the AAC-2's control program on or off.
Syntax	<p>PROG:([?]/[1]/[0]/[L])[CR\LF]</p> <p><b>PROG:?</b> responds with either a <b>1</b> or a <b>0</b>. A <b>1</b> indicates that the AAC-2 is in control program measuring mode. In this mode, the AAC-2 no longer uses the information that has been set by the STORE command. Instead, on each set time interval, the AAC-2 executes a control program that has been stored by the PROG: command.</p> <p><b>PROG:1</b> turns on the "control program" measurement mode. The control program is activated when the START: command is sent or when the AAC-2 is started by the use of the display keys.</p> <p><b>PROG:0</b> disables the control program measurement mode.</p> <p><b>PROG:</b> enters control program entry mode. In this mode, the commands sent to the AAC-2 will not be immediately executed, they are stored in a control program area. To return to normal command mode, an <b>X</b> followed by [CR\LF] must be sent to the AAC-2.</p> <p><b>PROG:L</b> causes the AAC-2 to send the command strings that are stored in the control program area.</p>

---

**QUIT**  
**(2,3,4,5,F,M)**

Purpose	Cancels online transmission of data.
Syntax	QUIT:[CR\LF]
Response	NA

---

**RADIO**  
**(3,4,5)**

Purpose	Controls the AAC-2 external power control facility by following the state of the GUARD.
Syntax	RADIO:[?]/[1]/[0][CR\LF]  <b>RADIO:?</b> responds with the power control state as follows:
Response	[ <i>Enabled,Active</i> ][CR\LF]  <i>Enabled</i> is a <b>1</b> if the AAC-2 currently monitors the state of the GUARD, otherwise it's <b>0</b> .  <i>Active</i> is a <b>1</b> if the GUARD state currently has caused the output relay (4) to be active

---

**REDUCE**  
**(2,3,4,5,F,M)**

Purpose	Sets/Reads data reduction (averaging) parameters for offline recordings.
Syntax	REDUCE:[?]/[dL,AVE,MIN,MAX][CR\LF]  where:  <b>dL</b> samples/storage <b>AVE</b> 1 if average is to be stored, else 0. <b>MIN</b> 1 if minimum is to be stored, else 0. <b>MAX</b> 1 if maximum is to be stored, else 0.  At least one of <b>AVE</b> , <b>MIN</b> and <b>MAX</b> must be a <b>1</b> .
Response	[Parameters]/[OK][CR\LF]

---

**REPEAT**  
**(2,3,4,5,F,M)**

Purpose	Sets measuring interval for on-line measurements, and starts measurements.
Syntax	<p>REPEAT:[TimingInterval(,S)(,B)][CR\LF]</p> <p>This command orders repeats of last SEND: Counters, if present and active, will be reset.</p> <p>The AAC-2 uses a <b>TimingInterval</b> of <b>00-59</b> seconds.</p> <p>The AAC-2f uses a <b>TimingInterval</b> of <b>100-1000</b> milliseconds.</p> <p><b>S</b> forces a synchronising string to precede all data batches, i.e. one string every <i>delta-t</i> ( =timing interval)</p> <p><b>B</b> switches to binary output format instead of normal ASCII. The binary format is : SOH/N/V1/ ... /Vn/CS/EOT (Slashes are not present. They are typed for the sake of clarity.) SOH - Start of heading: ASCII control character N - Two byte Hex batch number V1 - Four byte Hex (first) channel value Vn - Four byte Hex (last) channel value CS - Two byte Hex "overflowed" one byte checksum, excluding SOH and EOT EOT - End of text: ASCII control character</p>
Response	<p>Repeated batches of channel value strings. Each channel value is terminated by CR/LF If <b>S</b> is active, the string batches are preceded by [SYN/Batch No/CR/LF} where "Batch no" is a numeric string (three ascii digits) between 0 and 255. "Batch no" restarts at 0 at overflow. See above when <b>B</b> is active.</p>

---

**RING**  
**(2,3,4,5)**

Purpose	Sets or reads the number of ring signals that must pass before the AAC-2 answers via an attached or internal modem.
Syntax	RING:[1.. 5]/[?][CR\LF]
Response	[OK]/[NumberOfSignals][CR\LF]

---

**SEND**  
**(2,3,4,5,F,M)**

Purpose	Sends channel values over the RS-232 line. J,K etc. specifies T/C type in use (2)
Syntax	SEND:(J/K/T/S)(,)[K1,...,K32][CR\LF]
Response	[Value][CR\LF] . . . [Value][CR\LF]

---

**SET**  
**(2,3,4,5,M)**

Purpose	Sets alarms, Output relay and state, "Upper" and "lower" alarm levels.
Syntax	SET:[?]/[1]/[0]/[0,nCH][CR\LF] or SET:[dOut,StateBetween,nCH,Min,Max][CR\LF]
	SET:?      Lists all settings. Multiple level settings for each analog channel, directed to the same or different outputs, are allowed.
	SET:0      Disables level checking for all channels.
	SET:1      Enables level checking for all channels.
	SET:0,nCH    Removes level checking for channel nCH
	SET:dOut,StateBetween,nCH,Min,Max Adds a level setting for nCH directed to output dOut. State (0 or 1) between Min and Max must be given. Min and Max must be given as normalized levels as : NormLevel=TrueLevel/Nf where Nf is the normalization factor which can be fetched by the CONFIG:? command (for example 3.3E-4 from a model "2100" 10V range)
Response	[OK]/[ERR]/[dOut,StateBetween,nCH,Min,Max][CR\LF]

---

**START**  
**(2,3,4,5,F,M)**

Purpose	Starts an off-line recording.
Syntax	START:[?]/[YY,MM,DD,hh,mm,ss("STRING")][CR\LF] or START:[SerNo1,SerNo2,.....SerNon][CR\LF]  Where YYMMDD is date and hhhmmss is time for when recording shall start.  "STRING" (inverted commas are part of string and must be present) is an access-code or password specified by user. Recordings can only be stopped and restarted by using this password. See also STOP:  SerNo specifies serial number of several units connected to same RS-422 line  Counters, if present and active, will be reset.
Response	[OK/ERR][CR\LF] or [ON]/[OFF]/[ERR][CR\LF]

---

**STARTDATE**  
**(2,3,4,5,F,M)**

Purpose	Reads date of start of recording.
Syntax	STARTDATE:[?][CR\LF]
Response	[YY:MM:SS][CR\LF]

---

**STARTTIME**  
**(2,3,4,5,F,M)**

Purpose	Reads time of day of start of recording.
Syntax	STARTTIME:[?][CR\LF]
Response	[HH:MM:SS][CR\LF]

---

**STOP**  
**(2,3,4,5,F,M)**

Purpose	Stops an off-line recording.
Syntax	STOP:[?]/(["STRING"])[CR\LF]  "STRING" (inverted commas are part of string and must be present) is an access-code or password specified by user in START: (see START above). Recordings can then only be stopped and restarted by using this password.
Response	[OK]/[ERR][CR\LF] or [ON]/[OFF]/[ERR][CR\LF]

---

**STORE**  
**(2,3,4,5,F,M)**

Purpose	Sets the channels to be measured off-line. If used in a control program, this command actually performs the measurement and stores the data in the AAC-2 data memory.
Syntax	STORE:[?]/[Ka(,Kn+)...(,K32)][CR\LF] where K signifies channel(s) with numbers a, n ... 32
Response	[a, n, .. 32] or [OK]/[ERR][CR\LF]

---

**TERMCHAR**  
**(2,3,4,5,F,M)**

Purpose	Sets string termination character sequence for serial communication.
Syntax	TERMCHAR:[0D]/[0A]/[0D0A]/[0A0D][CR\LF]
Response	[OK]/[ERR][CR\LF]

---

**TERMO**  
**(2,3,4,5,M)**

Purpose	Sets correct thermocouple linearization for off-line recordings in units <u>not</u> equipped with "software range switching". (On line : specify "send" parameter)
Syntax	TERMO:[J]/[K]/[T]/[S]/../[?][CR\LF]
Response	[OK]/[J]/[K]/[T]/[S][CR\LF]

---

**TIME**  
**(2,3,4,5,F,M)**

Purpose	Sets or reads the AAC-2's real-time clock.
Syntax	TIME:[?]/[HH:MM:SS][CR\LF]
Response	[HH:MM:SS]/[OK][CR\LF]

---

**TITLE**  
**(2,3,4,5,F,M)**

Purpose	Reads or sets the title of the AAC-2's currently stored recording.
Syntax	TITLE:[?]/[TitleText][CR\LF]
Response	[TitleText]/[OK][CR\LF]

---

**TRIG**  
(F)

Purpose	Set switches for start/stop on trig.
Syntax	TRIG:[1/0,1/0,1/0,1/0]/[?][CR\LF]  where:  1 = on and 0 = off for the parameters in this order:  Start on external trig. Stop on external trig. Start on level trig. Stop on level trig.
Response	[OK]/[ERR]/[Switches][CR\LF]

---

**TRIGLEVEL**  
(F)

Purpose	Sets or reads triglevels for analogue channels.
Syntax	TRIGLEVEL:[?]/[K,SL,SH,EL,EH][CR\LF]  where:  <b>K</b> is the channel ID number. <b>SL</b> is the low start-trig level. <b>SH</b> is the high start-trig level. <b>EL</b> is the low stop-trig level. <b>EH</b> is the high stop-trig level.
Response	[Levels]/[OK][CR\LF]

---

**TRIGWINDOW**  
(F)

Purpose	Set pre and post trig times.
Syntax	TRIGWINDOW:[Pre,Post]/[?][CR\LF]  where: <b>Pre</b> is number of storage intervals prior to stop trig and <b>Post</b> is number of storage intervals after trig.
Response	[OK]/[ERR]/[Pre,Post][CR\LF]

---

**VERSION**  
(2,3,4,5,F,M)

Purpose	Reads AAC-2 version information.
Syntax	VERSION:[?]
Response	[No of strings] [Rom version string] [Ram version string] [Serial number] [Fabrication date]

## Programming example in GW-BASIC

---

```
100 REM *****
110 REM *
120 REM *          PROGRAM IN GW-BASIC THAT DEMONSTRATES *
130 REM *          COMMUNICATION WITH INTABS AAC-2's *
140 REM *
150 REM *****
160 REM
170 REM The AAC-2 is woken up when there's activity on the communications line.
180 REM We send a CR character directly to the ACIA and read the garbage.
190 REM
200 REM 1016 = COM1, 760 = COM2 !
210 PORT = 760 : OUT PORT,13 : FOR I=1 TO 10000 : NEXT : GARBAGE = INP( PORT )
220 REM
230 REM Activate BASIC error handling and
240 REM open the BASIC serial I/O
250 REM
260 ON ERROR GOTO 460
270 OPEN "COM2:9600,N,8,1,CS0,DS0" AS #1
280 REM
290 REM Set a proper BASIC termination sequence.
300 REM
310 PRINT #1, "TERMCHAR:0D" : INPUT #1,REPLY$
320 REM
330 REM Tell the AAC-2 to send channel 1, 2 and 3's present readings.
340 REM
350 PRINT #1, "SEND:1,2,3" ; SEND:K,1,2.. or SEND:T,1,2... For T/C channels
(J;K;T;S;:etc.)
360 FOR I = 1 TO 3
370 INPUT #1, V$
380 PRINT "Value";I;" ";V$
390 NEXT
400 REM
410 REM Turn off the AAC-2.
420 REM
430 PRINT #1,"OFF:"
440 CLOSE 1
450 END
460 CLOSE #1 : GOTO 270
```

□