revision 2
Prog version 1.02

INSTRUCTION MANUAL

DC900 COMPUTER INTERFACE UNIT FOR IEEE-488 AND RS232C

RV-Elektroniikka Oy
Veromiehentie 14
SF-01510 VANTAA, FINLAND

phone 358-0-822087
telex 19100449 vdx sf
fax 358-0-822184

TABLE OF CONTENTS **PAGE**

1. COMMAND SUMMARY

P	RESET TO DEFAULT SETTINGS	overlapped
E0	ECHO OFF Commands are not echoed	sequential
E1	ECHO ON Commands are echoed	sequential
E?	GIVE CURRENT ECHO STATE	sequential
C0	MANUAL CONTROL MODE Manual switches override	sequential
C1	COMPUTER CONTROL MODE Computer overrides	sequential
C?	GIVE CURRENT CONTROL MODE	sequential
R?	GIVE CURRENT RANGE	sequential
R<1...7>	SELECT RANGE	overlapped
X?	GIVE CURRENT EXCITATION	sequential
X<1...6>	SELECT EXCITATION	overlapped
D?	TRANSMIT RESULT OF NEXT COMPLETED A/D CONV.	overlapped
H?	TRANSMIT RESULT OF LAST COMPLETED A/D CONV.	sequential

MULTIPLEXER COMMANDS

M?	GIVE CURRENT INPUT CHANNEL	overlapped
M0	DISABLE MULTIPLEXER	sequential
M<1...15>	SELECT INPUT CHANNEL	overlapped

INDIRECT READOUT COMMANDS

O8	SELECT INDIRECT READOUT MODE	sequential
O0	SELECT DIRECT READOUT MODE	sequential
O?	GIVE CURRENT READOUT MODE	sequential
Snnnnn	DISPLAY STRING nnnnn (n = 0...9, other characters blanked)	sequential

STATUS REPORTING (all sequential)

*STB?	PRINT STATUS BYTE
*SRE <1...255>	SET SERVICE REQUEST ENABLE REGISTER
*SRE?	PRINT CONTENTS OF SERVICE REQUEST ENABLE REGISTER
*ESR?	PRINT CONTENTS OF STANDARD EVENT STATUS REGISTER
*ESE <1...255>	SET STANDARD EVENT ENABLE REGISTER
*ESE?	PRINT CONTENTS OF STANDARD EVENT ENABLE REGISTER
/AEC?	PRINT CONTENTS OF AVS-46 ERROR CONDITION REGISTER
/AEE?	PRINT CONTENTS OF AVS-46 ERROR EVENT REGISTER
/AEM <1...255>	SET ERROR EVENT ENABLE REGISTER
/AEM?	PRINT CONTENTS OF ERROR EVENT ENABLE REGISTER
/ASC?	PRINT CONTENTS OF AVS-46 STATUS CONDITION REGISTER
/ASE?	PRINT CONTENTS OF AVS-46 STATUS EVENT REGISTER
/ASM <1...255>	SET STATUS EVENT ENABLE REGISTER
/ASM?	PRINT CONTENTS OF STATUS EVENT ENABLE REGISTER

MISCELLANEOUS

/IBA <1..31>	SET IB ADDRESS NONVOLATILE REGISTER
/IBA?	PRINT IB ADDRESS NONVOLATILE REGISTER
/BPS <1...5>	SET BIT RATE NONVOLATILE REGISTER
/BPS?	PRINT BIT RATE NONVOLATILE REGISTER

2. INTRODUCTION

The DC900 computer interface for the AVS-46 Resistance Bridge has been designed for maximum flexibility of operation. It can be connected to either an RS232C serial interface or to the GPIB (HPIB, IEEE-488 etc.) bus.

The AVS-46 is a sensitive instrument which is mostly used for resistance thermometry at very low temperatures, and one usually tries to measure the sensor resistance with the lowest possible excitation current. The GPIB, on the other hand, is a very fast interface, carrying large current transients between instruments and the computer. In order to prevent these currents from coupling to the sensor and heating it, the DC900 interface has been made completely isolated from the bridge circuits. All information between the AVS-46 and the DC900 is transferred via two pulse transformers.

The DC900 interface is relatively easy to use. It can be configured by software, without opening the instrument. Most commands are exactly the same for both RS232 and GPIB operation, so that programs written for RS232 can be modified for GPIB with a minor effort, and vice versa. Further, programs originally written for the AVSI2 Serial Interface need only a small change to use the DC900 in RS232 mode.

The original IEEE-488 standard was replaced in June 1987 by IEEE-488.1, with only minor changes. In addition, a new standard, IEEE-488.2 was published, and this defines the codes, formats, protocols and commands that should be used by a bus-interfaceable instrument.

The demand for maximum similarity of RS232 and GPIB modes has forced a number of compromises, which have prevented full implementation of the new IEEE-488.2 standard. However, this does not mean incompatibility with a system adhering to the standard, but some omissions have been necessary.

This manual describes operation in both RS232 and GPIB modes. Differences of the two modes are pointed out when necessary. Should you have a question that this manual fails to answer, please feel free to contact us. The facts that we will need are the serial number of your AVS-46, the program version of the interface and, of course, what's the problem.

3. PREPARATIONS FOR RS232 OPERATION

3.1. RS232 CONNECTOR

The interface is configured as DTE, and the following signals are used (pins not mentioned are unconnected):

PIN	SIGNAL	TYPE	DESCRIPTION
1			Cable shield
2	TXD	Output	Asynchronous data
3	RXD	Input	Asynchronous data
4	RTS	Output	DC900 ready to accept data
5	CTS	Input	Enables DC900 to transfer data, Note 1
7	GND		Signal ground
20	DTR	Output	True when DC900 powered

NOTE 1. If not driven, CTS is held true by passive pullup.

3.2. RS232 CABLE FOR THE IBM PC

Following cable has been found suitable for use with the IBM PC, PC/XT and compatibles. We recommend that when you connect the DC900 interface to your PC for the first time, you use the RS232 mode together with the demo programs supplied with the interface.

```
DC900 CONNECTOR                         PC CONNECTOR

PIN 1 ------------------------PIN 1  CABLE SHIELD
PIN 2-------------------------PIN 3  (RXD input)
PIN 3-------------------------PIN 2  (TXD output)
PIN 7-------------------------PIN 7  (GROUND)
                            [ PIN 4  (RTS output)
                            [ PIN 5  (CTS input)

                            [ PIN 6  (DSR input)
                            [ PIN 20 (DTR output)
```

Following cable should be used with the 9-pin connector used in AT-type computers.

```
DC900                                   PC/AT CONNECTOR

PIN 1------------------------- -        CABLE SHIELD
PIN 2-------------------------PIN 2 (RXD input)
PIN 3-------------------------PIN 3 (TXD output)
PIN 7-------------------------PIN 5 (GROUND)
                            [ PIN 7 (RTS output)
                            [ PIN 8 (CTS input)

                            [ PIN 4 (DTR output)
                            [ PIN 6 (DSR input)
```

3.3. BAUD RATE AND CHARACTER FORMAT

When turned on, the AVS-46 will first show the revision number of the DC900 firmware and then, for a while, the current GPIB address and baudrate code. These default settings may be changed with /IBA and /BPS commands, respectively. The new settings are immediately stored in a non-volatile memory, but they become effective after the next power-on.

Please note that some of our demo programs supplied with the interface assume 4800 bits/s, and this is also the factory default setting of the DC900. Refer to /BPS command in section "Miscellaneous Commands".

The character format used for transmission is 8 databits, no parity and 1 stopbit.

3.4. COMPATIBILITY WITH AVSI2

If you have cables or programs made for the AVSI2 Serial Interface, please note the following differences:

1. Jumper between pins 4 and 5 of the DB25 connector is needed with the AVSI2, but is not necessary with the DC900. There is no need to remove the jumper, if it exists.

2. AVSI2 requires some numeric parameters in hexadecimal format (these are commands that are relayed to the TS-530 Temperature Controller). For simplicity, the DC900 accepts parameters in decimal format only.

3. AVSI2 accepts only upper-case letters. The DC900 accepts both upper- and lower case letters.

4. The DC900 does not support optocoupled current-loop mode, as the new internal transformer connection scheme permits standard RS232 operation while retaining complete isolation from the bridge.

4. HANDSHAKING IN RS232 MODE

4.1. SOFTWARE HANDSHAKING

The fact that the DC900 responds to all commands and queries, be they valid or not, provides simple means for implementing software handshaking. Then only two signal wires plus a ground lead are needed.

Transmit only one command at a time from the computer to the DC900. Then read the serial port until the complete response has been received. In many cases, but not always, the next command can be transmitted immediately upon receipt of the response. For example, the following program would take readings at maximum speed and print them on the screen.

```
10 open "com1:4800,n,8,1" as#1
20 print #1, "D?"                :'take next reading
30 line input #1,response$      :'read com1 until CR is received
40 print response$
50 goto 20
```

For almost all applications, software handshaking is quite adequate, as the AVS-46 is a slow instrument, and the amounts of transferred data are small. It also provides best compatibility with the AVSI2 interface.

Note that the LINE INPUT -command of the above example may turn out unreliable. If, for some reason like external interference, the CR character terminating the DC900 response is not received, or is not recognized, the program will wait forever. A much more reliable routine is found in the appendix. This monitors the state of the serial port buffer before attempting to read its contents.

4.2. HARDWARE HANDSHAKING

The DC900 provides an input buffer of 250 bytes. If the buffer becomes full, the RTS signal is set FALSE, and if applied to the CTS input of the computer, this will prevent buffer overflow.

The AVS-46 is a slow instrument, and with the presently supported short commands, there should not be any need for hardware handshaking.

Please note that neither soft- nor hardware handshaking can guarantee that the commanded operation will be succesfully completed.

5. THE IEEE-488 FUNCTIONS

The interface uses a commercial controller IC, the NEC uPD 7210. The functional subset implemented is specified in the table below:

DC900 Device Capability

SH1	Source handshake complete capability
AH1	Acceptor handshake complete capability
T6	Basic talker, with serial poll, unaddress with MLA
TE0	No extended talker
L4	Basic listener, unaddress with MTA
LE0	No extended listener
SR1	Service request complete capability
RL0	No Remote-Local capability
PP0	No parallel poll capability
DC1	Device Clear complete capability
DT0	No Device Trigger capability
C0	No controller capability
E1	Open collector drivers

6. INSTRUCTION SYNTAX

6.1 GENERAL PRINCIPLES OF OPERATION

The DC900 unit recognizes one logic input, one logic output and the interface with the AVS-46 bridge circuit. **The logic input and output are dynamically assigned to that physical interface port (RS232 or GPIB) which has last provided an input character.** The two interfaces can thus be both connected to a host computer if only the user has some means of guaranteeing that they are not used simultaneously, in which case the results are indeterminate.

Both interfaces transfer streams of 7-bit ASCII characters (with an enforced 8th bit) which are buffered on both input and output side (250 characters each). If the input buffer is full, overflow is prevented by turning the RS232 signal RTS FALSE and by NRFD holdoff (GPIB). Characters are buffered without any processing. It is possible to transmit commands at maximum speed without waiting for each command to be executed. This possibility must, however, be used in a disciplined manner, in order to maintain sychronization. This is true especially of the high-speed IB, but a few commands execute slowly enough to cause trouble even with serial transfer, if uncautiously used.

Synchronization is also dealt with below in the detailed descriptions of each command.

The parser extracts one character at a time from the input buffer, converting lower-case alphabet characters to uppercase, and suppressing white space, until a valid delimiter is found. The valid delimiters are listed below:

Interface	Character	ASCII(decimal)	Comment
RS232:	CR	13	Message Terminator
	;	59	Command separator
GPIB:	LF	10	Message Terminator
	EOI		Message Terminator
	;	59	Command Separator

If the resulting character string matches one of the defined command strings, execution of the command is initiated immediately, and the parser starts extracting a new command.

6.2. COMMAND AND QUERIES

This manual distinguishes between command and query messages. A **command** includes one or no parameter, and changes the state of the instrument in some way.

If RS232 is used, a single CR character is written to the output if the command followed by the possible parameter is accepted by the DC900. Otherwise the response is a question mark followed by CR. The response, if affirmative, indicates that the command has been initiated, but it does not mean that the command was succesfully completed.

In case of the GPIB, nothing is written to the output port as response to commands.

A **query** affects the state of the instrument only in the sense that it causes data to be written to the logic output. Presently, all response data is numeric and decimal (IEEE 488.2 <NR1> or <NR2>).

6.3. COMMAND SYNTAX

Most of the legal DC900 messages appear in pairs, with a command for setting a parameter, and a query formed by adding a question mark to the command header to obtain the current value of the parameter.

A command consists of a command mnemonic of one to four characters, followed by no more than one parameter in integer decimal format (<NR1>). The mnemonic and parameter can, but do not have to, be separated by white space (decimal ASCII 32). A query is formed by adding a question mark to the mnemonic. For reasons of compatibility, DC900 will accept a command string which begins with a defined mnemonic, even if it is followed by additional characters before the delimiter. In this case, those characters will be simply discarded by the DC900.

For compatibility with software written for the AVSI2 serial interface, DC900 will recognize a query also if the question mark is missing, as long as the mnemonic is not followed by any numeric data which would cause the message to be interpreted as a command.

6.4. SEQUENTIAL AND OVERLAPPED MESSAGES

It is also possible to fit several commands and queries into one message, by separating them with a semicolon. As explained above, the parser will start extracting and initiating commands as soon as the input buffer is non-empty. One input message will produce no more than one output message. If the input message contains several units separated by semicolons, the response message will also contain units separated in the same manner. But it is necessary to take into account the excecution speeds of various commands. Most queries and some commands -which are referred as **sequential** commands and queries- execute within a few milliseconds, and they can be given in a fast sequence without problems. However, response to the D? query, for example, may take up to 0.4 seconds and execution of the M<n> command can take up to one or two seconds. In order to preserve the execution order of messages, **overlapped** commands should be the last ones in a sequence, followed by delay long enough to allow completion. The versatile status queries then provide means to suspend further commands to the DC900 until they can be executed.

6.5. POWER-ON DEFAULT STATE, P COMMAND

When the AVS-46 is turned on, the DC-900 makes a sequence of initialization actions. The firmware version number of the DC900 is displayed for some seconds, and then the GPIB device address and baud rate code, separated by the decimal point. Refer to the /BPS command for interpretation of these codes.

A typical power-on display would be:

+ or - 1.02	where 1.02 is the program version, followed by
16.00	where 16 is the current GPIB address and 00 indicates baud rate of 4800 bps (see sec. 8)

The range is set to 20 kΩ, excitation to 1 mV, multiplexer is disabled, direct readout mode is selected and echo is turned off. Finally, the AVS-46 is set to manual mode and all registers and buffers are cleared.

The DC900 also provides an external serial program interface for the TS-530 Temperature Controller. Power-on routine includes scanning of the TS-530 addresses and setting them to their corresponding default states.

The AVS-46 (and TS-530) can be set to the default state by command

P (instrument reset command)

This behaves exactly in same way as the power-on routine, except that program version etc. are not displayed, and that a possibly changed GPIB device address or baud rate does not become effective after the P command (for this purpose, it is necessary to switch power off and on).

The P command is of overlapped type and its completion takes a variable time depending on how many times the range, excitation and multiplexer have to be clocked up or down until their default states are achieved. Note also that P command terminates all pending operations, and that the command is finished by clearing the input buffer.

The user shall take care that no further commands are given to the DC900 as long as the P command is being executed. This can be done either by inserting a delay after the P command (this should be at least 3 seconds), or better still, by the *ESR? query. Upon completion of the P command, bit 7 of the ESR register will be set. You can also use the /AEC? query, which will return a non-zero value until default state has been achieved.

6.6. ECHO

In its default state, the DC900 does not echo characters. However, if the DC900 is connected to a "dumb" terminal through the RS232 port, then you can turn echo on by command

E1 (sequential)

and turned off by

E0 (sequential)

Current state of echoing is queried by E?. Note that this command can be useful only in RS232 mode.

7. STATUS QUERIES AND COMMANDS

All commands in the status reporting group are sequential in nature. The query messages in the group cause register contents to be written to the Output Queue in decimal integer (<NR1>) format.

***STB?** PRINT STATUS BYTE

The response to this query is the IEEE-488 status byte written to logic output in <NR1> format. This is a condition register, that is, it reflects the current status of the instrument, and can not be directly changed by the user (reading does not clear this register). The following summary message bits are defined:

```
MSB 7
    6 RQS/MSS    Request Service/Master summary status
                 The device has at least one reason
                 to request service
    5 ESB        Event Status Register non-zero
    4 MAV        Output queue not empty
    1 ASB        AVS-46 status event register not zero
LSB 0 AEB        AVS-46 error event register not zero
```

***SRE <1...255>** SET SERVICE REQUEST ENABLE REGISTER
***SRE?** PRINT CONTENTS OF SERVICE REQUEST ENABLE REGISTER

This register is used to specify to the device which bits of the Status Byte Register may set the local message rsv true to initiate a service request. The bits corresponding to unused bits of the Status Byte Register are silently ignored. Power-on state 0 (no bits enabled). Enabling is achieved by AND operation between STB and SRE registers. For example, in order to enable only the MAV and AEB bits, your command would be *SRE 5.

***ESR?** PRINT CONTENTS OF STANDARD EVENT STATUS REGISTER

The DC900 Event Status Register is a partial implementation of the Standard Event Status Register specified by IEEE standard 488.2. In this register the bits are set when the corresponding conditions arise, and they remain set until the register is read, and are then automatically cleared. The following bits are used:

```
MSB 7 PON    Register has not been read since power-on.
    5 CME    Command Error
    4 EXE    Execution Error
    2 QYE    Query Error
```

***ESE <1...255>** SET STANDARD EVENT ENABLE REGISTER
***ESE?** PRINT CONTENTS OF STANDARD EVENT ENABLE REGISTER

The Standard Event Status Enable Register allows one or more events in the Standard Event Register (ESR) to be reflected in the ESB summary-message bit of the Status Byte Register (STB). Power-on state 0 (none enabled).

/AEC? PRINT CONTENTS OF AVS-46 ERROR CONDITION REGISTER

This register reflects a number of instrument specific error conditions. In this context, it should be noted that the condition of actual values deviating from set values is considered an error only if the C-bit has been set (with C1 command), that is, if the instrument is under active computer control as opposed to being passively monitored via the remote interface. This register is not cleared by reading it.

The following error conditions are reflected by this register:

```
MSB 7
    6 E_MUX  Multiplexer current setting <> set value
    5 E_DVM  A copy of the DVM overrange output bit
    4 E_RNG  Range selector actual value <> set value
    3
    2 E_EXC  Excitation selector actual value <> set value
    1 E_OOF  Output queue overflow error
    0
```

/AEE? PRINT CONTENTS OF AVS-46 ERROR EVENT REGISTER

The AVS-46 error event register is used to capture **changes** in the error condition register. Each time a bit transition in the error condition register occurs (an error condition arises or disappears), the corresponding bit in the event register is set. A nonzero bit will thus indicate that at least one change has occurred since the event register has been read. Consequently, it does NOT give any implication as to the current state of the condition register bit, if there has been no changes. This register is cleared by reading it.

/AEM <1...255> SET AVS-46 ERROR EVENT ENABLE REGISTER
/AEM? PRINT CONTENTS OF AVS-46 ERROR EVENT ENABLE REGISTER

The Standard Event Status Enable Register allows one or more events in the AVS-46 Error Event Register to be reflected in the AEB summary-message bit of the Status Byte Register. Power-on state 0 (none enabled).

/ASC? PRINT CONTENTS OF AVS-46 STATUS CONDITION REGISTER

This register contains some status bits reflecting other than error conditions. Presently, the following bits are defined:

```
MSB 7
    6
    5
    4 S_DPOL  DVM polarity
    3
    2
    1 S_BADS  Bad reading counter nonzero
    0 S_MODB  AVS-46 mode bit
```

An error condition sets a Bad reading counter to a value of 6, and when the error is cleared the counter is decremented by each completed A/D conversion. After the sixth conversion the S_BADS bit is reset.

S_MODB is presenly not used.

/ASE? PRINT CONTENTS OF AVS-46 STATUS EVENT REGISTER

This register is used to record changes in device status in a manner analogous to the AVS-46 Error Event Register described above. In addition to bits defined above for the ASC register, ASE defines unused bit 3 as:

```
3 S_DRDY  DVM reading ready
```

This bit is set by a completed A/D conversion of the AVS-46 digital voltmeter. As reading clears the ASE register, this bit provides means to synchronize to the free-running A/D converter.

/ASM <1...255> SET AVS-46 STATUS EVENT ENABLE REGISTER
/ASM? PRINT CONTENTS OF AVS-46 STATUS EVENT ENABLE REGISTER

The Status Event Enable Register allows one or more events in the AVS-46 Status Event Register to be reflected in the ASB summary-message bit of the Status Byte Register. Power-on state 0 (none enabled).

8. MISCELLANEOUS COMMANDS

In this instrument, there are no switches to set the GPIB address or serial interface baud rate. Instead, they are stored in internal nonvolatile memory, and changed using ordinary remote commands. On power-up, the display will first show the firmware version number for a moment, and after that shift to display GPIB address and bit rate code at the same time (see also sec 6.5). These registers can be rewritten at any time, but changes take effect only at power-on. Both interfaces can be used to change either register.

/IBA <1..31> SET GPIB ADDRESS NONVOLATILE REGISTER
/IBA? PRINT IB ADDRESS NONVOLATILE REGISTER

GPIB talk and listen address. Factory default value is 16. If you use the /IBA command to change the device address, remember to switch the AVS-46 off and on in order to make the new setting effective.

/BPS <0...5> SET BAUD RATE NONVOLATILE REGISTER
/BPS? PRINT BAUD RATE NONVOLATILE REGISTER

RS232 transmit and receive baud rate. The possible settings and corresponding codes are shown below:

Code	Baud rate, bps	
0	4800	Factory default
1	2400	
2	1200	
3	600	
4	300	
5	150	

9. COMPUTER CONTROL OF THE AVS-46

9.1. WHAT CAN BE CONTROLLED

Most AVS-46 functions, although not all, can be remotely controlled through the DC900 interface. For clarity, we list below the operations that are possible with the DC900, and operations that are not possible:

YOU CAN

*Get the DVM reading to your computer
*Display this reading as such, or linearize the data with a computer and send the result to AVS-46 to be displayed.
*Select resistance range
*Select excitation range
*Select multiplexer input channel
*Operate the AVS-46 as a manual instrument and read current range, excitation and input channel to computer
*Obtain complete isolation between AVS-46 and computer.
*Operate your TS-530 Temperature Controller remotely without need for a second interface unit.

YOU CANNOT

*Change positio of following manual switches: CAL/MEAS, FAST/SLOW, DELTA-R/R, REF BUTTON, AUTO/MANUAL RANGE, AUTO/MANUAL EXCITATION, x1/x10 MODE
*Set the AVS-46 deviation reference from computer

9.2. POWER-ON DEFAULT STATE

When the AVS-46 is switched on, the DC900 assumes a state for normal manual operation of the AVS-46, specifically:

* Range, excitation and multiplexer are controlled only by the front panel toggle switches

* Multiplexer is disabled (channel indicator digit is blanked).

* Display is in resistance readout mode.

In the initial state range should be 20 kΩ and excitation 1000 microvolts. The default settings (see descriptions in later paragraphs) are:

X = 5 (excitation)
R = 5 (range)
E = 0 (echo)
C = 0 (manual mode)
M = 0 (multiplexer disabled)

9.3. MANUAL MODE AND COMPUTER MODE

Computer control mode is selected by command

C1 (sequential)

In computer mode, the DC900 overrides selection of range, excitation and multiplexer channel. All commands and queries can be used in computer mode.

Manual mode is restored by command

C0 (sequential)

In manual mode, the AVS-46 front panel switches override, and neither R, X nor M commands will be executed. If any of these commands are given in C0 mode, they will be queued and executed only after a C1 command (beware of limitations with overlapped messages!).

All queries can be used in manual mode, which makes it possible to use the computer as a passive "recorder" rather than a controller.

Current mode is asked by sequential query C?.

10. READING THE RESISTANCE DATA

The A/D converter of the AVS-46 is of free-running type, making about 2.5 conversions per second. It is based on the well-known dual-slope integration, so that exact conversion time cannot be defined. Unlike successive-approximation converters, the ICL7135 does not provide means to start conversion by an external command. Therefore, other devices have to synchronize themselves to this converter.

There are basically two different ways to do that. The first, and simple, is suitable for the RS232 mode. It makes use of the D? query and software handshake. The host computer is forced to wait until the next A/D conversion has been completed and result transmitted to the computer. This method, although easy, makes the whole system quite slow.

A more refined way is needed with the GPIB mode. The H? query returns the result of the last completed A/D conversion. However, status queries must be used to verify that one A/D conversion is read only once. The H? can also be used to speed up operation in RS232 mode.

10.1. D? QUERY

The result of the next A/D conversion is read using query

D? (overlapped)

The DC900 responds by sending a string of characters, starting from polarity (+ or -), followed by the conversion result from MSD (most significant digit) to LSD (least significant digit), and the string ends with a CR (RS232) character or LF (GPIB) . The result is always expressed in ohms, with a decimal point inserted in the string when needed.

A valid result ranges from -19999 to +19999. Overload condition is shown by forcing the response to 9999900.

Please note that D? queries cannot be queued in the input buffer. If there are more than one D? in the buffer, only the last of them will be executed. Avoide giving any further commands to the DC900 while awaiting the resistance data.

10.2. H? QUERY

The result of the last completed A/D conversion is read by query

H? (sequential)

This, unlike D?, provides immediate response. Use this query in connection with one or more status queries to prevent multiple reads of a single conversion. For example, you can check bit 3 of the ASE register and make the H? query only when the bit is set.

10.3. DECIMAL POINT

The data from the A/D converter is a decimal number expressed in Ohms regardless of range setting. If range 4 (2 kΩ) or lower is selected, the resistance data will include decimal point, but readings taken on the three highest ranges are integers.

Note that decimal point position in magnified deviation mode corresponds to that of the original range. It is therefore necessary to divide the readings by 10. This mode is selected only by the AVS-46 hardware, and is not reflected by any of the status bytes.

11. RANGE

11.1 R COMMAND, R? QUERY

In computer mode, range can be selected by command

R<1...7> (overlapped)

Changing range takes some time depending on how many decades the range selector must be stepped. A/D conversion result during this time will show error condition (9999900). Also bit 1 of the ASC register and bit 4 of the AEC register can be checked to verify completion of the R command. Note that the AVS-46 still needs some time to stabilize, although the error bits have been cleared.

Current range can be asked by query

R? (sequential)

Number 1 corresponds to 2 Ω range and number 7 to 2 MΩ range.

11.3. AUTORANGING

The autoranging feature of the AVS-46 is useful only in CO mode. Attempt to use autoranging in computer mode will lead to a never-ending race condition between the autoranging logic of the AVS-46 and the DC900. Beware that when the autoranging action takes place, some readings transmitted to computer may be invalid and your program should be able to detect this.

Note also that a 1 or 5 seconds delay, depending on a jumper setting, is reserved for the bridge to stabilize between successive autoranging operations. In scanning applications with the multiplexer, better results can be obtained in computer mode, without using the autoranging feature, if the last range setting for each sensor is stored in computer memory, and the selection of multiplexer channel is accompanied by selecting this range as a default.

12. EXCITATION

12.1. X COMMAND, X? QUERY

When the DC900 is in computer mode, the excitation can be selected by command

X<1...6> (overlapped)

Number 1 corresponds to 10 μV excitation range and number 6 to 3 mV excitation range.

Like with R command, readings taken when excitation is being clocked up or down will be invalid (9999900). Bit 1 of the ASC register and bit 2 of the AEC register can be used to detect completion of the X command. Note that the readings, although valid, will not be stable immediately after a change of excitation.

Current excitation may be asked by

X? (sequential)

and the response is now an integer 1...6.

12.2. AUTOMATIC CHANGE

One of the AVS-46 features is the possibility to change excitation automatically whenever the autoranging circuitry selects a new resistance range. Autoexcitation can therefore be used only when the bridge is in autorange mode (and, consequently, the DC900 in manual mode).

13. MULTIPLEXER

13.1. M COMMAND, M? QUERY

The AVS-46 may be equipped with one seven-channel input multiplexer. In the reset state after power-on, the multiplexer is disabled and the channel display LED is blanked.
Multiplexer input channel is selected using command

M<1...7> (overlapped)

Command M0 disables the multiplexer. The AVS-46 front panel input switch should be then switched to FRONT position.

NOTE: The AVS-46 needs a long time to recover from a heavy overload. Therefore be careful not to leave multiplexer disabled if the bridge input has been assigned to it.

The selected input channel number is seen at AVS-46 display and can be asked by the sequential query M?.

14. INDIRECT READOUT

14.1 COMMANDS

The DC900 changes over into indirect readout mode by command

O8 (sequential, note: letter O)

This mode is indicated by blanking the decimal point of the AVS-46 display. Now the interface is ready to receive data from the computer, to be displayed. Indirect readout can be used in both CO and C1 modes.

Direct readout mode is restored by command

O0 (sequential)

and current readout mode is read by O? query.

Data to be displayed by AVS-46 is transmitted from computer using command

S<nnnnn> (sequential)

where the string nnnnn consists of one to five digits, each ranging from 0 to 9. Examples:

* S0 and S0000 are displayed as 00000
* S1 and S0001 00001
* S1000 01000
* S99999 99999 (upper limit)

The displayed temperature must be in form of a positive integer (decimal point is blanked in this mode). In cryogenic applications below 100K, the display units may be millikelvins and at higher temperatures the unit can be 10 mK. A given string is displayed until updated by a new S command.

14.2. APPLICATIONS

The indirect readout provides some extra convenience in applications where the AVS-46 is interfaced with a computer, especially if the latter and its screen are located outside the cryostat room.

One useful way to use the indirect readout mode is to display a digital average of many readings, as calculated by the computer. This will provide a more stable display at the lowest sensor power levels.

If the computer calculates temperature from the sensor resistance, indirect readout may be used to display the temperature in units of millikelvins or tens of millikelvins. Whereas the standard AVS-46 readout extends to 19999, full five digits up to 99999 are available for the indirect mode. Thus, temperature display goes up to 999K with a resolution of 10 mK and up to 99K with 1 mK resolution.

In a multiplexed-input application, where several sensors are scanned, you may want to display just one sensor instead of the individual readings that change continuously. Indirect readout makes this possible, as the readout mode does not affect the operation of the A/D converter.

It is also possible to use the indirect readout to show the channel number and temperature. This is made by constructing the S-command string as follows:

S + (channel number digit) + (hex digit greater than 9) + (three digits of data).

Of course, this leaves you only three digits for the temperature, which is enough for 1 mK resolution below 1 K and for 0.1 mK resolution below 0.1 K.

The 4 1/2 digit display range can be extended to 30000 or 40000 with the aid of the indirect readout.

Select deviation display mode and null the resistance display in CAL mode by the REFERENCE potentiometer. Then use the computer to add 10000 to all subsequent readings, and send them back to the AVS-46.

You can also dial the potentiometer to 19990 with the aid of the SET display, and then add 19990 to all readings (range is extended to 39990). This latter method is not quite as accurate as the former one.

15. RS232 PROGRAMMING

15.1 GENERAL PRINCIPLES

Asynchronous communications can sometimes be tricky, and therefore we present here some principles and hints that has been found useful.

DC900 responds to all commands and queries. Any string ending with a <CR> character is interpreted as a message. If this message is not a valid command or query, response will be a question mark (?) followed by <CR>, and if the message was valid, DC900 will take the requested action or send the requested data, and terminate the response with a <CR>. DC900 will never transmit characters without a preceding message from computer.

Reading data from the DC900, through the computer's serial port is not as easy as giving commands to the DC900. First of all, you must make sure there is a character or string you can read. If you fail to do this, you will either get an error message, or your computer will wait for the character. But the character cannot be received, as the DC900 does not send anything without a preceding command. You cannot send the command because your computer is waiting, and there is nothing else to do than to abort the program or reset the computer.

In the next section we describe a BASIC subroutine that has found quite reliable. This routine is used in demo programs DC900.EXE and NTEST.EXE. This routine concentrates only to writing and reading data via the RS232 line. It does not make any conclusions regarding the validity of the received data.

15.2. A READ-WRITE SUBROUTINE

Before entering the subroutine, string variable TXD$ must be set equal to the command or query string to be transmitted. First the routine checks that the communications buffer is empty, and if not, it reads as many characters as there are in the buffer (line 720). Then TXD$ is transmitted to DC900.

It will take some unknown time until the DC900 response is completed. Therefore a cycle counter is initialized (line 740). Next the routine tests whether a character has been received or if the buffer is still empty. If empty, the cycle counter is incremented and the buffer is tested again. After a reasonable number of trials, a time-out error message is set, and control is returned to main program.

It may be necessary to increase the cycle counter value for AT-type or 386-based computers.

If there is a character in the buffer, it is read (790). If this character is a question mark, an error message string is set, and we branch to the end of the subprogram. Branching to end occurs also if a carriage-return character is read (820), otherwise the last character is concatenated to the response string RXD$ (830). Reading procedure is repeated until a CR character is encountered, and RXD$ is complete. The final test of line 840 is actually not necessary as the buffer is flushed in the beginning, but it is good to make this test at least once.

```
690 'SUBROUTINE TO WRITE DATA TO DC900 AND TO READ THE RESPONSE
700 ' INPUT TO ROUTINE IS TXD$ AND OUTPUT FROM IS RXD$
710 '
720 IF LOC(1)>0 THEN FLUSH$=INPUT$(LOC(1),1)
730 '
740 PRINT #1, TXD$:RXD$="":ERRMSG$=""
745 TURNS=0
750 IF LOC(1)=0 THEN TURNS=TURNS+1 ELSE GOTO 790
```

```
760 IF TURNS<10000 THEN 750
770 ERRMSG$="TIMEOUT ERROR IN RECEIVING":GOTO 840
780 '
790 A$=INPUT$(1,1)
800 IF A$="?" THEN ERRMSG$="UNKNOWN COMMAND "+TXD$:GOTO 840
810 '
820 IF ASC(A$)=13 THEN 840
830 RXD$=RXD$+A$:GOTO 745
840 IF LOC(1)>0 THEN FLUSH$=INPUT$(LOC(1),1)
850 RETURN
```

LINEARISATION ALGORITHM FOR PT-100 SENSORS

Following short algorithms can be used to convert PT-100 resistance readings into temperature, and the desired final temperature into resistance (for defining control set point).

Input parameters for the first algorithm are TSTART, which is any Celsius temperature, and R, which is the resistance reading in Ohms. Calculation is made by iteration from the DIN 43760 equations, which are different for positive and negative (Celsius) temperatures.

Iteration converges quite rapidly. If you do not scan many sensors, you can retain the TSTART value from one subroutine call to the next and reduce the number of iteration cycles (row 2940) to make program faster. In scanning applications it is probably wiser to use a fixed start value and a sufficient number of cycles.

Before returning, Celsius degrees are converted into Kelvins, but you can skip lines 2980 and 3040.

The input parameter for the second algorithm is B$, which represents a Kelvin temperature to be converted into resistance. It is first converted into Celsius degrees (line 3100). Positive and negative temperatures have different formulas. Iteration is not needed, as the DIN equation can be used directly.

Output parameter is set point SP in terms of volts.

```
2900 REM SUBROUTINE TO CALCULATE TEMPERATURE FROM RESISTANCE DATA
2910 REM ITERATION IS BASED ON DIN 43760 EQUATION.
2920    T=TSTART
2930    IF R<100 THEN GOTO 3000
2940    FOR I=1 TO 20
2950    T=(R-100+5.80195E-05*T*T)/.390802
2960    NEXT
2970    TSTART=T
2980    T=T+273.15
2990    GOTO 3050
3000    FOR I=1 TO 20
3010    T=(R-100+5.80195E-05*T*T+4.2735E-10*(T-100)*T*T*T)/.390802
3020    NEXT
3030    TSTART=T
3040    T=T+273.15
3050 RETURN
```

```
3080 REM SUBROUTINE TO CALCULATE SET POINT RESISTANCE FROM SET POINT
3090 REM TEMPERATURE.
3100 T=VAL(B$)-273.15
3110 IF T<0 THEN 3140
3120 SP=(1+3.90802E-03*T-5.80195E-07*T*T):REM SP IN VOLTS
3130 GOTO 3150
3140 SP=(1+3.90802E-03*T-5.80195E-07*T*T-4.2735E-12*(T-100)*T*T*T):REM SP IN VOLTS
3150 RETURN
```

INDEX