# ASCII commands for ED-582

1. %AAnnttccff 3
2. #** 4
3. #AA 5
4. #AAN 6
5. $AA2 7
6. $AA4 8
7. $AA5 9
8. $AARS 10
9. $AA5VV 11
10. $AA6 12
11. $AA7CiRrr 13
12. $AA8Ci 14
13. $AAF 16
14. $AAFC 17
15. $AAFM 18
16. $AAFR3V3 19
17. $AAFR5V 20
18. $AAFU 21
19. $AAFVA 22
20. $AAFVB 23

21. $AAM 24

22. $AAB 25

23. ~AAD 26

24. ~AADVV 27

25. ~AAO(name) 28

26. ~** 29

27. ~AA0 30

28. ~AA1 31

29. ~AA2 32

30. ~AA3EVV 33

31. $AAS1 34

32. $AAM0 35

33. $AAM1 36

34. ~AAL(data) 37

## %AAnnttccff

**Description:**
Command to set the RTD device configuration.

**Command Syntax:**

%**AAnnttccff[CS]**(CR)

<table>
<tr><td>%</td><td>Delimiter character</td></tr>
<tr><td><b>AA</b></td><td>Address of the device to be configured in hexadecimal format (00 to FF)</td></tr>
<tr><td><b>nn</b></td><td>New device address in hexadecimal format (00 to FF)</td></tr>
<tr><td><b>tt</b></td><td>RTD type code, this is not used. Default value is 80.<br>Use the $AA7CiRrr command to set the type of each channel.</td></tr>
<tr><td><b>cc</b></td><td>New Baud Rate code.
<table>
<tr><th>Code:</th><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>0A</td></tr>
<tr><th>Baud Rate:</th><td>1200</td><td>2400</td><td>4800</td><td>9600</td><td>19200</td><td>38400</td><td>57600</td><td>115200</td></tr>
</table></td></tr>
<tr><td><b>ff</b></td><td>Used to set filter settings, data format, temperature unit and checksum
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>FS</td><td>CS</td><td colspan="2">Reserved</td><td colspan="2">TU</td><td colspan="2">DF</td></tr>
</table>
<u>FS: Filter Setting</u><br>0: 60Hz Rejection<br>1: 50Hz Rejection<br><br><u>DF: Data format</u><br>00: Temperature<br>11: Resistance<br><br><u>TU: Temperature Unit</u><br>00: Celsius<br>01: Fahrenheit<br>10: Kelvin<br><br><u>CS: Checksum Setting</u><br>0: Disabled<br>1: Enabled</td></tr>
<tr><td><b>[CS]</b></td><td>Checksum</td></tr>
<tr><td>(CR)</td><td>Carriage Return</td></tr>
</table>

**Response:**
Valid Command: **!AA[CS]**(CR)
Invalid Command: **?AA[CS]**(CR)

| | |
|---|---|
| **!** | Delimiter for a valid command |
| **?** | Delimiter for an invalid command |
| **AA** | Address of the device in hexadecimal format (00 to FF) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Change the device address from 01 to 02. The device returns a valid response.
Command: %0102800600(CR) Response: !02(CR)

Change the Baud Rate of device 01 to 115200. Device returns a valid response.
Command: %0101800A00(CR) Response: !01(CR)

Change the Temperature Unit of the device from Celsius to Fahrenheit. Device returns a valid response.
Command: %0101800604(CR) Response: !01(CR)

Change the Baud Rate of device 01 to 115200. Device returns invalid command because the baud rate code is incorrect.
Command: %010180FF00(CR) Response: ?01(CR)

**Notes:**

*Any change to the address, filter settings, data format direction will take effect once the command has been received. Changes to the Baud Rate, checksum settings and temperature unit will take effect after the device is restarted.*

## #**

**Description:**

Allows the device to read and store data for later retrieval.

**Command Syntax:**

#**[CS](CR)

| | |
|---|---|
| # | Delimiter character |
| ** | Synchronized sampling command |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**
There is no response for this command. To read the data, the $AA4 command is used.

**Examples:**
Sends the synchronized sampling command.

Command: #**(CR)                    No response

## #AA

**Description:**

Command to read data from all the RTD inputs.

**Command Syntax:**

#**AA[CS]**(CR)

| | |
|---|---|
| # | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: >**Data[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| > | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |
| **(Data)** | Data from the RTD input channels |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Read the RTD values of the all the lines

| | |
|---|---|
| Command: | #01(CR) |
| Response: | >+027.31+044.31+101.31+120.31(CR) |

## #AAN

**Description:**

Command to read the RTD input of the specified channel.

**Command Syntax:**

#**AAN[CS]**(CR)

| | |
|---|---|
| # | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| **N** | The channel to be read (0 to F) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: >**(Data)[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| > | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |
| **DDDDD** | Data returned from the RTD channel specified. |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Read the RTD value of the fourth channel.

| | |
|---|---|
| Command: #013(CR) | Response: >+120.31(CR) |

## $AA2

**Description:**

Reads the device configuration.

**Command Syntax:**

$**AA2[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format(00 to FF) |
| **2** | Command to read the device configuration |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AATTCCFF[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

<table>
<tr><td>!</td><td>Delimiter for a valid command</td></tr>
<tr><td>?</td><td>Delimiter for an invalid command</td></tr>
<tr><td><b>AA</b></td><td>Address of the device (00 to FF)</td></tr>
<tr><td>TT</td><td>RTD type code. Default value is 80.</td></tr>
<tr><td>CC</td><td>Baud Rate of the device
<table>
<tr><td><b>Code:</b></td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>0A</td></tr>
<tr><td><b>Baud Rate:</b></td><td>1200</td><td>2400</td><td>4800</td><td>9600</td><td>19200</td><td>38400</td><td>57600</td><td>115200</td></tr>
</table>
</td></tr>
<tr><td>FF</td><td>Data format setting
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>FS</td><td>CS</td><td colspan="2">Reserved</td><td colspan="2">TU</td><td colspan="2">DF</td></tr>
</table>
<br>
FS: Filter Setting<br>
0: 60Hz Rejection<br>
1: 50Hz Rejection<br>
<br>
CS: Checksum Setting</td></tr>
</table>

| | |
|---|---|
| | 0: Disabled<br>1: Enabled<br><br>DF: Data format<br>00: Temperature<br>11: Resistance<br><br>TU: Temperature Unit<br>00: Celsius<br>01: Fahrenheit<br>10: Kelvin |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the configuration of device 01.

Command: $012(CR)　　　　Response: !01800600(CR)

## $AA4

**Description:**

Reads the synchronized data that was retrieved by the last #** command.

**Command Syntax:**

$**AA**4**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format(00 to FF) |
| 4 | Command to read the device configuration |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: >**AAS(Data)[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |
| **S** | Status of the synchronized data:<br>1: first read<br>0: not the first read |
| **(Data)** | Synchronized data |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Read synchronized data. Device returns the synchronized data and sets first byte to 1 to indicate this is the first time the synchronized data has been read. (Assumes the set synchronized data command has been sent)

| | |
|---|---|
| Command: | $014(CR) |
| Response: | >011+027.31+044.31+101.31+120.31(CR) |

Read synchronized data. Device returns synchronized data and sets status byte to 0 to indicate the data has been read before.

| | |
|---|---|
| Command: | $014(CR) |
| Response: | >010+027.31+044.31+101.31+120.31(CR) |

## $AA5

**Description:**

Command to read the reset status of a module.

**Command Syntax:**

$**AA**5**[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |

| | |
|---|---|
| 5 | Command to read the reset status |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AAs[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the device |
| **s** | Reset status of the device.<br>Device has been reset = 1<br>Device has not been reset = 0 |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Send command to read reset status of device

Command: $015(CR) Response: !011(CR)

## $AARS

**Description:**

Reset the device to power on state.

**Command Syntax:**

$AARS**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| RS | Command to reset device |

| | |
|---|---|
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

No Response

**Examples:**

Reset the device to power on state. Command sent to the ED-582 and no response is expected.

| | | |
|---|---|---|
| Command: | $01RS(CR) | No Response |

## $AA5VV

**Description:**

Command to enable a specific channel.

**Command Syntax:**

$**AA**5**VV[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 5 | Command to read the reset status of the device |
| **VV** | Two digit hexadecimal value representing RTD channels where bit 0 is channel 0, bit 1 is channel 1 etc.<br>1: enable<br>0: disable |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**
Valid Command: **!AA[CS]**(CR)
Invalid Command: **?AA[CS]**(CR)

| | |
|---|---|
| **!** | Delimiter for a valid command |

| | |
|---|---|
| **?** | Delimiter for an invalid command |
| **AA** | Address of the device in hexadecimal format (00 to FF) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Enable only the first channel on the ED-582.

| | |
|---|---|
| Command: $01501(CR) | Response: !01(CR) |

Enable all the 4 input channels on the ED-582

| | |
|---|---|
| Command: $0150F(CR) | Response: !01(CR) |

## $AA6

**Description:**

Reads the channels enable/disable status.

**Command Syntax:**

$**AA**6**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 6 | Command to read the channels enable/disable status |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AAVV[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |

| | |
|---|---|
| **VV** | Two digit hexadecimal value representing RTD channels where bit 0 is channel 0, bit 1 is channel 1 etc.<br>1: enable<br>0: disable |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Read the enable/disable status of the channels.

Command: $016(CR) Response: !010F(CR)

## $AA7CiRrr

**Description:**

Command to set the specified channel RTD type.

**Command Syntax:**

$**AA**7CiR**rr[CS]**(CR)

<table>
<tr><td>$</td><td>Delimiter character</td></tr>
<tr><td><b>AA</b></td><td>Address of the device to be configured in hexadecimal format (00 to FF)</td></tr>
<tr><td>7</td><td>Command to set the channel RTD type</td></tr>
<tr><td>Ci</td><td>Command to specify the input channel where <b>i</b> is the input channel</td></tr>
<tr><td>R<b>rr</b></td><td>Command to specify the type code where <b>rr</b> is the RTD type code to be set
<table>
<tr><th>Type code</th><th>RTD input type</th><th>Temperature Range</th></tr>
<tr><td>80</td><td>PT-100 with α=0.00385 (European Standard)</td><td>-200°C to +600°C</td></tr>
<tr><td>89</td><td>PT-100 with α=0.003911 (American standard)</td><td>-200°C to +600°C</td></tr>
<tr><td>81</td><td>PT-100 with α=0.003916 (Japanese Standard)</td><td>-200°C to +600°C</td></tr>
<tr><td>8B</td><td>PT-100 with α=0.003926 (ITS-90)</td><td>-200°C to +600°C</td></tr>
<tr><td>8D</td><td>PT-1000 with α=0.00375 (RTD1000-375)</td><td>-200°C to +600°C</td></tr>
<tr><td>2A</td><td>PT-1000 with α=0.00385 (European Standard)</td><td>-200°C to +600°C</td></tr>
</table>
</td></tr>
</table>

| | | | |
|---|---|---|---|
| | 8A | PT-1000 with α=0.003911 (American standard) | -200°C to +600°C |
| | 88 | PT-1000 with α=0.003916 (Japanese standard) | -200°C to +600°C |
| | 8C | PT-1000 with α=0.003926 (ITS-90) | -200°C to +600°C |
| | 29 | Ni-120 with α=0.00672 | 0°C to +100°C |
| **[CS]** | Checksum | | |
| (CR) | Carriage Return | | |

**Response:**

Valid Command: !AA**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Set the RTD type of channel 0 to 2A(PT-1000 with α=0.00385) and module returns a valid command.

Command: $017C0R2A(CR)    Response: !01(CR)

Sets the RTD type for channel 1 of module 03 to be 30. The module returns an invalid response because the type code is invalid.

Command: $037C1R30 (CR)    Response: ?03(CR)

## $AA8Ci

**Description:**

Command to read the specified channel RTD type.

**Command Syntax:**

$**AA**8C**i[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 8 | Command to read channel RTD type |
| C | Command to specify the channel |
| **i** | Channel to be read |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**C**i**R**rr[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the device which responded |
| C | Channel read command |
| **i** | Specifies the channel the data is from |
| R | RTD type command |
| **rr** | RTD type data from the specified channel. Please refer command $AA7CiRrr for the RTD type codes table. |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Read the RTD type of the first channel.

Command: $018C0(CR)    Response: !01C0R80 (CR)

Read the RTD type of the fourth channel.

Command: $018C3(CR) Response: !01C3R2A(CR)

## $AAF

**Description:**

Command to read the firmware version on the device.

**Command Syntax:**

$**AA**F**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| F | Command to read the firmware version |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**(Data)**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the device which responded |
| **(Data)** | Firmware version of the responding device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the firmware version of the device and shows it as version 1.9.

Command: $01F(CR)    Response: !011.9(CR)

## $AAFC

**Description:**

Command to read the CPU temperature on the device.

**Command Syntax:**

$**AA**FC**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FC | Command to read CPU temperature |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**tt.tt**C[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| tt.tt | CPU temperature of the responding device in Celsius |
| C | Temperature unit in Celsius |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the CPU temperature of the device and shows it as 30.25C

Command: $01FC(CR)

Response: !0130.25C(CR)

## $AAFM

**Description:**

Command to read the MAC address of the device.

**Command Syntax:**

$**AA**FM**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FM | Command to read MAC Address |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**UU:VV:WW:XX:YY:ZZ**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| UU:VV:WW:XX:YY:ZZ | MAC address of the responding device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the MAC address of the device and shows it as 00:0A:4F:05:05:88

Command: $01FM(CR)

Response: !0100:0A:4F:05:05:88(CR)

## $AAFR3V3

**Description:**

Command to read 3V3 rail voltage of the device.

**Command Syntax:**

$**AA**FR3V3**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FR3V3 | Command to read 3V3 rail voltage. |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**XX.YY**V[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| XX.YY | 3V3 rail voltage of the responding device |
| V | Voltage Unit |
| **[CS]** | Checksum |

| | |
|---|---|
| (CR) | Carriage Return |

**Examples:**

Reads the 3V3 rail voltage of the device and shows it as 3.21V

Command: $01FR3V3(CR)

Response: !0103.21V(CR)

## $AAFR5V

**Description:**

Command to read 5V rail voltage of the device.

**Command Syntax:**

$**AA**FR5V**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FR5V | Command to read 5V rail voltage. |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**XX.YY**V[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| XX.YY | 5V rail voltage of the responding device |
| V | Voltage Unit |

| | |
|---|---|
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the 5V rail voltage of the device and shows it as 4.85V

Command: $01FR5V(CR)

Response: !0104.85V(CR)

## $AAFU

**Description:**

Command to read up time.

**Command Syntax:**

$**AA**FU**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FU | Command to read up time |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**DD,HH,MM,SS**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |

| | |
|---|---|
| DD,HH,MM,SS | Up time of the responding device in Days, Hours, Minutes and Seconds as comma separated format |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the up time of the device and shows it as 0 days, 1 hour, 15 minutes, 20 sec

Command: $01FU(CR)

Response: !0100,01,15,20(CR)

## $AAFVA

**Description:**

Command to read power supply VA of the device.

**Command Syntax:**

$**AA**FVA**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FVA | Command to read power supply VA |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**XX.YY**V[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |

| | |
|---|---|
| **AA** | Address of the responding device |
| XX.YY | Power supply VA of the responding device |
| V | Voltage Unit |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the power supply VA of the device and shows it as 4.94V

Command: $01FVA(CR)

Response: !0104.94V(CR)

## $AAFVB

**Description:**

Command to read power supply VB of the device.

**Command Syntax:**

$**AA**FVB**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| FVB | Command to read power supply VB |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**XX.YY**V[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |

| | |
|---|---|
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| XX.YY | Power supply VB of the responding device |
| V | Voltage Unit |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the power supply VB of the device and shows it as 4.92V

Command: $01FVB(CR)

Response: !0104.92V(CR)

## $AAM

**Description:**

Reads the name of the device.

**Command Syntax:**

$**AA**M**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| M | Command to read the device's name |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA(Data)[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| **(Data)** | Name of the device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the device name. Valid response is returned with the device's name, ED-582.

Command: $01M(CR) Response: !01ED-582(CR)

## $AAB

**Description:**

Command to read the channel diagnostic status of the RTD inputs.

**Command Syntax:**

$**AA**B**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| B | Command to read the channel diagnostic status |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**NN**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |

| | |
|---|---|
| **AA** | Address of the device which responded |
| **NN** | Diagnostic result of all the RTD input channels. |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads diagnostic code. The module returns a valid response denoting that channel 0 is in either over-range, under-range or wire opening condition.

Command: $01B(CR) Response: !0101(CR)

## ~AAD

**Description:**

Command to read the miscellaneous settings of the device.

**Command Syntax:**

$**AA**D**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| D | Command to read the miscellaneous settings |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA**VV**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the device which responded |
| VV | Two hexadecimal digits that represent the miscellaneous settings as follows: |

<table>
<tr><td></td><td>
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Reserved</td><td>SU</td><td>SR</td><td colspan="2">Reserved</td></tr>
</table>
SR – Setting for over range/under range readings
<table>
<tr><td>SR</td><td>Data Format</td><td>Over range</td><td>Under Range</td></tr>
<tr><td>0</td><td>Temperature</td><td>+9999</td><td>-0000</td></tr>
<tr><td>1</td><td>Temperature</td><td>+9999.0</td><td>-9999.9</td></tr>
</table>
<br>
SU – Setting for under range reading<br>
0 – The reading of under range is as usual<br>
1 – Force the under range reading to be the same as over range
</td></tr>
<tr><td><b><u>[CS]</u></b></td><td>Checksum</td></tr>
<tr><td>(CR)</td><td>Carriage Return</td></tr>
</table>

**Examples:**

Reads the miscellaneous settings of the module.

Command: ~01D(CR) Response: !0104(CR)

## ~AADVV

**Description:**

Command to set the miscellaneous settings of the device.

**Command Syntax:**

$**AA**DVV**[CS]**(CR)

<table>
<tr><td>$</td><td>Delimiter character</td></tr>
<tr><td><b><u>AA</u></b></td><td>Address of the device to be configured in hexadecimal format (00 to FF)</td></tr>
<tr><td>D</td><td>Command to read the miscellaneous settings</td></tr>
<tr><td>VV</td><td>Two hexadecimal digits that represent the miscellaneous settings as follows:
<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Reserved</td><td>SU</td><td>SR</td><td colspan="2">Reserved</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>SR – Setting for over range/under range readings
<table>
<tr><td>SR</td><td>Data Format</td><td>Over range</td><td>Under Range</td></tr>
<tr><td>0</td><td>Temperature</td><td>+9999</td><td>-0000</td></tr>
<tr><td>1</td><td>Temperature</td><td>+9999.0</td><td>-9999.9</td></tr>
</table>
SU – Setting for under range reading<br>0 – The reading of under range is as usual<br>1 – Force the under range reading to be the same as over range</td></tr>
<tr><td><b>[CS]</b></td><td>Checksum</td></tr>
<tr><td>(CR)</td><td>Carriage Return</td></tr>
</table>

**Response:**

Valid Command: !AA**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Sets the miscellaneous settings of module 01 to 04, and returns a valid response

Command: ~01D04(CR) Response: !01(CR)

## ~AAO(name)

**Description:**

Sets a the name of the device to the name specified.

**Command Syntax:**

~**AA**O**(Name)[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| O | Command to read the device’s name |
| (Name) | New device name (10 characters max) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !AA**[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter for an invalid command |
| **AA** | Address of the responding device (00 to FF) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Set the device name to 582 and receives a valid command response.

Command: ~01O582(CR)     Response: !01(CR)

## ~**

**Description:**

Command sent to all devices to say the host is OK.

**Command Syntax:**

~****[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| ** | Command to check the host is OK |

| | |
|---|---|
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

No response.

**Examples:**

Send a “Host OK” command to all the devices

Command: ~**(CR)

No response

## ~AA0

**Description:**

Command to read the watchdog status of the device.

**Command Syntax:**

~**AA**0**[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 0 | Command to read the device watchdog status |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AASS[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the device |

| | |
|---|---|
| **SS** | Two Hexadecimal digits indicating the host watchdog status<br>SS=00 – Watchdog timeout is cleared<br>SS=04 – Watchdog timeout is set |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Read the host watchdog status and response is 00 meaning the host watchdog is disabled

Command: ~010(CR)　　　　　Response: !0100(CR)

Read the host watchdog status and response is 04 meaning that a host watchdog timeout has occurred.

Command: ~010(CR)　　　　　Response: !0104(CR)

## ~AA1

**Description:**

Command to reset the watchdog status of the device.

**Command Syntax:**

~**AA**1**[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 1 | Command to reset the watchdog timeout status |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |

| | |
|---|---|
| **AA** | Address of the device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Send command to reset the watchdog status and return valid response.

Command: ~011 Response: !01

## ~AA2

**Description:**

Command to read the watchdog timeout value of the device.

**Command Syntax:**

~**AA**2**[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 2 | Command to read the watchdog timeout value |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AAEVV[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the device |

| | |
|---|---|
| **E** | Watchdog enabled status<br>1: Watchdog enabled<br>0: Watchdog enabled |
| **VV** | Two hexadecimal digits representing watchdog timeout value in tenths of a second.<br>01 = 0.1 seconds, FF=25.5 seconds |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Send command to read the watchdog timeout value return valid response with FF meaning the watchdog timeout value is 25.5 seconds.

Command: ~012(CR)                    Response: !011FF(CR)

## ~AA3EVV

**Description:**

Command to enable/disable the watchdog and set the watchdog timeout value.

**Command Syntax:**

~**AA**3**EVV[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| 3 | Command to read the watchdog timeout value |
| **E** | Watchdog enabled status<br>1: enable the host watchdog<br>0: disable the host watchdog |
| **VV** | Two hexadecimal digits representing watchdog timeout value in tenths of a second.<br>01 = 0.1 seconds, FF = 25.5 seconds |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Send command to enable watchdog and set the timeout value to 25.5 seconds.

Command: ~0131FF(CR) Response: !01(CR)

## $AAS1

**Description:**

Command to restore factory setting and preserve the calibration value.

**Command Syntax:**

$**AA**S1**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| S1 | Command to restore factory setting and preserve the calibration value |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid response |
| **AA** | Address of the responding device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reset to the factory default while preserving the calibration value

Command: $01S1 Response: !01

## $AAM0

**Description:**

Command to read the device model.

**Command Syntax:**

$**AA**M0**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| M0 | Command to read the device’s model |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA(Data)[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |

| | |
|---|---|
| **(Data)** | Model of the device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples:**

Reads the device model. Command sent to the ED-582 and a valid response is returned with the device’s model, ED-582.

Command: $01M0(CR)    Response: !01ED-582(CR)

## $AAM1

**Description:**

Command to read the device location.

**Command Syntax:**

$**AA**M1**[CS]**(CR)

| | |
|---|---|
| $ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| M1 | Command to read the device’s location |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA(Data)[CS]**(CR)<br>
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command |
| **AA** | Address of the responding device |
| **(Data)** | Location of the device |
| **[CS]** | Checksum |

| | |
|---|---|
| (CR) | Carriage Return |

**Examples:**

Reads the device's location. Command sent to the ED-582 and a valid response is returned with the device’s location, machine1.

Command: $01M1(CR) Response: !01machine1(CR)

## ~AAL(data)

**Description:**
Sets a the location of the device.

**Command Syntax:**

~**AA**L**(Location)[CS]**(CR)

| | |
|---|---|
| ~ | Delimiter character |
| **AA** | Address of the device to be configured in hexadecimal format (00 to FF) |
| L | Command to set the location of the device |
| **(Location)** | New location name (10 characters max) |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Response:**

Valid Command: !**AA[CS]**(CR)
Invalid Command: ?**AA[CS]**(CR)

| | |
|---|---|
| ! | Delimiter for a valid command |
| ? | Delimiter character for an invalid command. |
| **AA** | Address of the device |
| **[CS]** | Checksum |
| (CR) | Carriage Return |

**Examples**:

Set the device location to Office and receives a valid command response.

Command: ~01LOffice(CR)    Response: !01(CR)