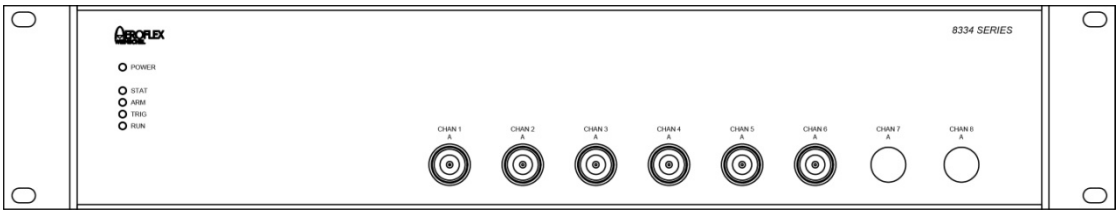
Operation & Installation Manual



Model 8334 Series
Attenuator Profile Simulator Unit

api technologies corp.
› WEINSCHEL

Frederick, Maryland
2013

SAFETY SUMMARY

DEFINITIONS.

The following definitions apply to WARNINGS, CAUTIONS, and NOTES found throughout this manual.

WARNING

An operating or maintenance procedure, practice, statement, condition, etc., which, if not strictly observed, could result in injury and/or death of personnel. Do not proceed beyond a WARNING symbol until all the indicated conditions have been fully understood and/or met.

CAUTION

An operating or maintenance procedure, practice, statement, condition, etc., which, if not strictly observed, could result in damage or destruction of the equipment or long-term health hazards to personnel. Do not proceed beyond a CAUTION symbol until all the indicated conditions have been fully understood and/or met.

NOTE

An essential operating or maintenance procedure, condition, or statement that must be highlighted.

GENERAL PRECAUTIONS.

The following are general precautions that are not related to any specific procedure and, therefore, do not appear elsewhere in this publication. These are precautions that personnel must understand and apply during various phases of instrument operation or service.

WARNING

- Potentially lethal voltages are present in this instrument. Serious shock hazards from voltages above 70 volts may exist in any connector, chassis, or circuit board. Observe the following precautions:
- To minimize shock hazard, the instrument chassis must be connected to an electrical ground. Using the supplied three-conductor power cable ensures that the instrument can be firmly connected to the ac power source and electrical ground at a grounded power outlet. If using a 3-2 wire adapter be sure to connect the ground lead to earth ground.
- Use the buddy system any time work involving active high voltage components is required. Turn OFF the power before making/breaking any electrical connection. Regard any exposed connector, terminal board, or circuit board as a possible shock hazard. DO NOT replace any component or module with power applied.
- If test conditions to live equipment are required, ground the test equipment before probing the voltage or signal to be tested.
- Personnel working with or near high voltage should be familiar with modern methods of resuscitation.
- DO NOT wear jewelry (rings, bracelets, metal watches, and/or neck chains) while working on exposed equipment. Be very cautious about using hand tools near exposed backplanes, bus bars, and/or power supply terminals. Use properly insulated tools. When making test connections to the power supply terminals and bus bars, use only insulated probe tips.
- Verify that the instrument is set to match the available line voltage and the correct fuse is installed.
- DO NOT install substitute parts or perform any unauthorized modification to this instrument. Contact Weinschel Corporation to acquire any information on replacement parts or returning the instrument for repair. Unauthorized modification can cause injury to personnel and/or destruction of the instrument.
- Operating personnel must not remove instrument covers. Component replacement or adjustments MUST BE performed by qualified service personnel.
- DO NOT operate the instrument near or in the presence of flammable gases or fumes.

DETAILED PRECAUTIONS.

The following WARNINGS, CAUTIONS and NOTES appear throughout the text of this manual and are repeated here for emphasis.

CAUTION

- All procedures and/or steps identified as must be followed exactly as written and according to industry accepted ESDS device handling procedures. Failure to comply WILL RESULT in ESDS damage.
- DO NOT use a nylon bristle brush in the solvent as the bristles may dissolve and cause damage to the circuit card or component.
- DO NOT use ultrasonic cleaning on parts or assemblies containing electrical or electronic components.
- DO NOT bend pins of electrical connectors when using fiber-bristle brush.
- Compressed air used for cleaning and/or drying can create airborne particles that may enter the eye. Goggles/faceshields should be worn. DO NOT direct air stream towards self or other personnel. Pressure should be restricted to a maximum of 15 psi to avoid personal injury.
- Under no circumstances should a wire brush, steel wool, or abrasive compound be used on any surface. Using these items will cause extensive damage to the instruments surface.

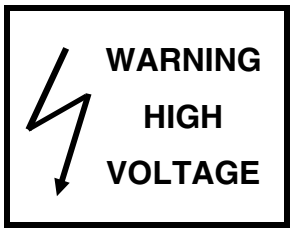
NOTE

DO NOT return any instrument or component to Weinschel Corporation without receiving prior factory authorization.

SAFETY SYMBOLS.

The following symbols are used to identify safety hazards found throughout this publication and/or located on the instrument.

CAUTION
HIGH VOLTAGE



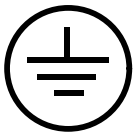


TABLE OF CONTENTS

1. GENERAL INFORMATION 4

1-1. PURPOSE 4
1-2. SCOPE 4
1-3. EQUIPMENT DESCRIPTION 4
1-4. UNPACKING AND INSPECTION 5
1-5. RESHIPMENT INSTRUCTIONS 5
1-6. STORAGE INSTRUCTIONS 5
1-7. RELATED MANUALS 6
1-8. ELECTROSTATIC DISCHARGE SENSITIVE (ESD) 6
1-9. ABBREVIATIONS & ACRONYMS 6
1-10. SAFETY CONSIDERATIONS 6

2. SPECIFICATIONS 7-8

2-1. GENERAL SPECIFICATIONS 7
2-2. CONFIGURATION MATRIX / RF SPECIFICATIONS 7

3. INSTALLATION 8-9

3-1. RACKMOUNTING 8
3-2. INTIAL SETUP 8
3-3. INPUT/OUTPUT OPTIONS 8-9
3-3.1. POWER ENTRY MODULE ASSEMBLY 9
3-3.2. CONTROL CONNECTORS 9
3-3.3 RF PORT CONNECTORS 9

4. OPERATION 10-54

4-1. SYSTEM BLOCK DIAGRAM 10-11
4-2. FRONT PANEL INDICATORS 11
4-3. SIMULATION SWEEP SETUP 12-13
4-4. RS232 SERIAL/CONSOLE PORT 14
4-5. 10/100BASET ETHERNET 15
4-6. USB 15
4-7. COMMAND OPERATION 16
4-8. COMMAND EXECUTION AND BUFFERING 16
4-9. COMMAND REFERENCE 17
4-10. APPLICATION SPECIFIC COMMANDS 17
4-10.1 ATTENUATION COMMANDS (DIRECT I/O) 17
4-10-2 SWEEP SETUP COMMANDS 18-22
4-10-3. SWEEP CONTROL/STATUS COMMANDS 22
4-10.4 TABLE MEMORY COMMANDS 23-24
4-11. SHOW COMMANDS 25-26
4-12. MISC SETUP COMMANDS 27
4-13. 488.2 COMMON COMMANDS 28-29
4-14. SETUP AND CONFIGURATION COMMANDS 30-34
4-14.1 SERIAL PORT COMMANDS 30
4-14.2 NETWORK COMMANDS 31-34
4-15. SHOW COMMANDS 35-37

4-16. MISC COMMANDS 38-41
4-17. MISC NETWORK COMMANDS 41
4-18. UPDATING DEVICE FIRMWARE VIA SERIAL PORT 42-43
4-19. USB DRIVER FILE (AW83XXCDC.INF) 44-45
4-20. INSTALLING API WEINSCHEL USB CDC DRIVER 46-48
4-21. USING THE MICROCHIP ETHERNET DISCOVERER TOOL 49
4-22. CREATING TABLES WITH EXCEL 50-54

5. MAINTENCE 55-56

5-1. INSPECTION 55
5-2. PREVENTIVE MAINTENANCE 55
5-3. SPECIAL CLEANING INSTRUCTIONS 55
5-3.1. MICROWAVE COAXIAL CABLE ASSEMBLIES 55
5-3.2. MACHINED SURFACES AND HARDWARE 56
5-3.3. CONNECTOR CLEANING 56
5-4. LINE VOLTAGE FUSE REPLACEMENT 56

6. REPLACABLE PARTS LIST 56

6-1 REPLACEABLE PARTS LIST (RPL) 56
6-2 ASSEMBLY AND COMPONENT LOCATION 56

7.CONTACTING API / WEINSCHEL. 57

8.API / WEINSCHEL WARRANTY 57

9. SPECIFICATION/ICD DRAWINGS

ICD, ATTENUATOR UNIT, TYPE N THRU, MECHANICAL SPECIFICATION, MODEL 8334 SERIES 089-4442A

ICD, ELECTRICAL SPECIFICATIONS, ATTENUATOR PROFILE SIMULATOR (8834) 089-4318A

1. GENERAL INFORMATION:

1-1 PURPOSE: This manual contains setup and operation information for the API / Weinschel Model 8334 Series, Attenuator Profile Simulator Unit, P/N 193-8300-XX-XX.

1-2 SCOPE: This manual is to be used in conjunction with the operation and installation of the Model 8334 Series. The manual also provides a description of the assembly; block diagrams: and general maintenance procedures to maintain the instrument.

1-3 EQUIPMENT DESCRIPTION: API / Weinschel Model 8334 series of Attenuation Profile Simulators (Figure 1) provide multi-channel high-speed attenuation control with synchronous output update capability. The unit allows for programming of up to 128K (131072) attenuation data points per attenuator and sweeping through those data points at user-programmable intervals from 100us to 1sec per point. The system provides for non-volatile storage of up to four data point tables which may be later recalled under user control. Status and control TTL signals are available for external monitoring and sweep control via a rear-panel DE9 connector.

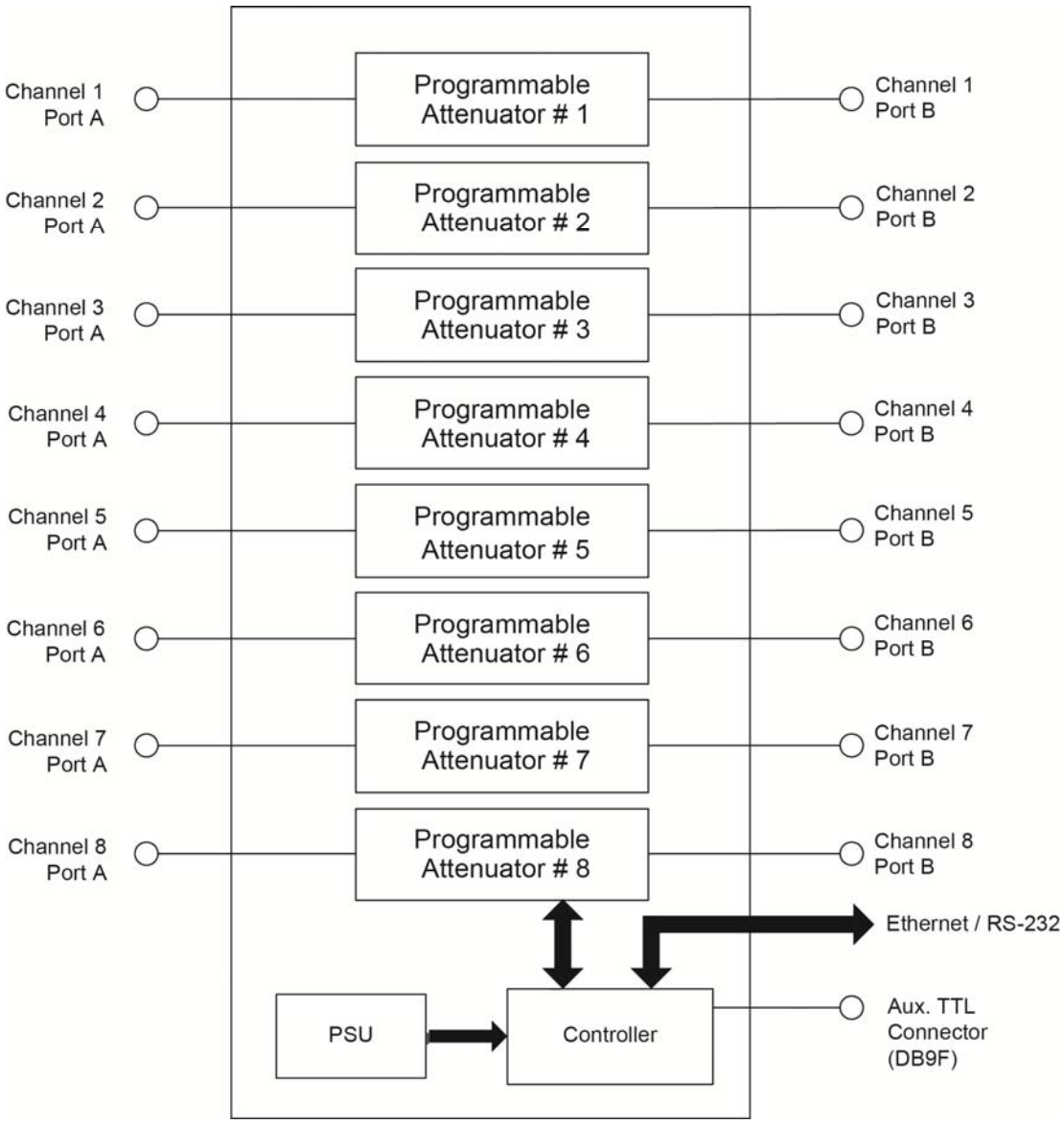


Figure 1. Model 8334 Simplified Block Diagram

1-4. UNPACKING AND INSPECTION: Upon unpacking the equipment, retain the shipping container and packing material for future shipment for recalibration. Perform the following initial inspection:

a. Carefully look at the outside of the shipping container for discoloration, stains, charring, or other signs of exposure to excessive heat, moisture, or liquid chemicals. Check for any physical damage to the shipping container such as dents, snags, rips, crushed sections or areas, or similar signs of excessive shock or careless handling.

b. With the equipment and any accessory package removed from the shipping container, check each item against the packing list or Items Supplied List. If any items are missing, contact the API / Weinschel Customer Service Department.

c. Carefully inspect the equipment looking for dents, deep scratches, damaged or loose connector, or any other signs of physical abuse or careless handling. If damage is found, forward an immediate request to the delivering carrier to perform an inspection and prepare a concealed-damage report. DO NOT destroy any packing material until it has been examined by an agent of the carrier. Concurrently, report the nature and extent of damage to API / Weinschel, giving equipment model and serial numbers, so that necessary action can be taken. Under U.S. shipping regulations, damage claims must be collected by the consignee; DO NOT return the equipment to API / Weinschel until a claim for damages has been established.

2-5. RESHIPMENT: Use the best packaging materials available to protect the unit during storage or reshipment. When possible, use the original packing container and cushioning material. If the original packing materials are not available, use the following procedure:

a. Wrap the storage cases in sturdy paper or plastic;

b. Place the wrapped storage cases in a strong shipping container and place a layer of shock-absorbing material (3/4 inch minimum thickness) around all sides of the unit to provide a firm cushion and to prevent movement inside the container.

c. If shipping the unit for service, attach a tag to indicate:

1. model and serial numbers
2. service required
3. description of malfunction
4. return address
5. authorization to conduct repairs
6. return authorization number

d. Thoroughly seal the shipping container and mark it FRAGILE. Ship to:

API / Weinschel, Inc.
Attn: Customer Service Department
5305 Spectrum Drive
Frederick, MD 21703-7362
or to an authorized sales representative.

1-6. STORAGE: Storage of the Model 8334 is possible for extended periods without incurring damage to internal circuitry if the Model 8334 is packaged according to the instructions above. The safe limits for storage environment are as follows:

Temperature: -67° to +167 °F (-55° to +75°C)
Humidity: 5% to 85% (non-condensing)

1-7. RELATED MANUALS: The following manuals contain information that may be used in conjunction with this manual to operate, service, or calibrate this instrument.

Manual	Title
IM-609-1	Manual, Drawings & Replaceable Parts List, Model 8334 Series & Specials

1-8. ELECTROSTATIC DISCHARGE SENSITIVE: The equipment documented in this manual contains certain Electrostatic Discharge Sensitive (ESDS) components or parts. Therefore, certain procedures/steps are identified by the use of the symbol . This symbol is used in two ways:

CAUTION

All procedures and/or steps identified as must be followed exactly as written and according to accepted ESDS device handling procedures. Failure to comply **WILL RESULT** in ESDS damage.

a. When the ESDS symbol is placed between a paragraph number and title all of that paragraph, including all subparagraphs, is considered ESDS device handling procedure.

b. When the ESDS symbol is placed between a procedure/step number and the text , all of that procedure is considered an ESDS device handling procedure.

1-9. ABBREVIATIONS AND ACRONYMS: The following list contains abbreviations used throughout this manual. Abbreviations and acronyms that are not listed conform to MIL-STD-12D.

ESDS	Electrostatic Discharge Sensitive
TBD	To Be Determined

1-10. SAFETY CONSIDERATIONS: The Model 8334 Series and all related documentation must be reviewed for familiarization with safety markings and procedures before any operation and/or service. Refer to the SAFETY SUMMARY located at the beginning of this manual for a summary of safety information and procedures. Following these simple safety precautions will ensure safe operation and service of the unit.

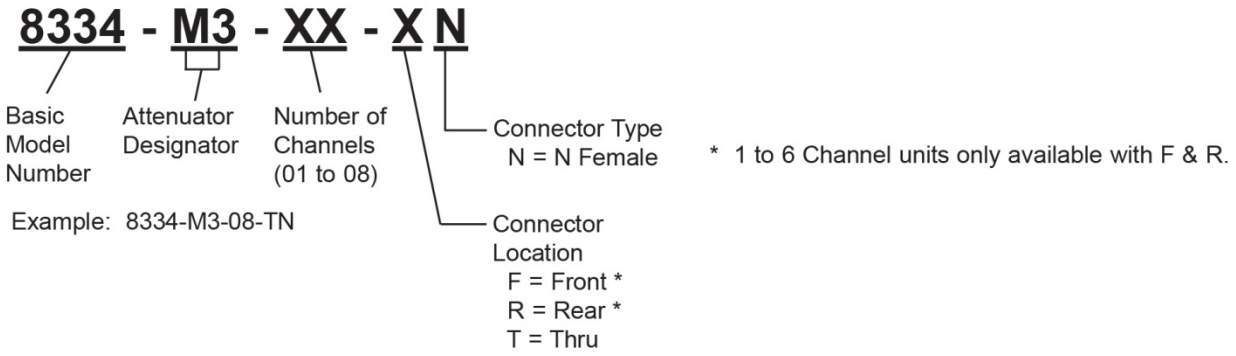
2. SPECIFICATIONS:

2-1. GENERAL SPECIFICATIONS:

Input Power Requirements	AC	100 to 240 Vac, 50/60 Hz, 180 Watts
Environmental	Operating Temperature Storage Temperature: Humidity: Altitude:	0° to +50°C -67° to +167 °F (-55° to +75°C) 5% to 85% (non-condensing) 40,000' (12,192M)
RS232 Bus[1] Serial I/O / Console Port	Connector: Signals: Baud Rates: Data Bits: Handshaking: Parity:	9-pin male D TXD, RXD, RTS, CTS, GND 9600 to 230400 8 None, RTS/CTS None
AUX	Connector: Signals:	9-pin male D SYNC, EXT TRIG, INTERVAL UPDATE, RUN, GND
Ethernet	10/100 Base T Connector:	Standard RJ45
USB 2.0	Connector:	USB, MINI B
RF Characteristics[2]	Refer to API / Weinschel ICD/Specification Drawing 089-4442.	
Physical Dimensions	Refer to API / Weinschel ICD/Specification Drawing 089-4443.	

NOTES:
1. RS-232 can be used with standard PC serial port for short and medium distances (up to approximately 50 ft).
2. Refer to Individual data sheet (Appendix B) for detailed specifications on internal programmable attenuators.

2-2. CONFIGURATION MATRIX:



Solid State (Only)								
Frequency Range	Attenuator Designation		Attenuator Model	Range (dB)	Step Size (dB)	Insertion Loss (maximum)	VSWR (maximum)	☑ RoHs
0.2 to 6 GHz	M	3	4205-95.5	95.5	0.5	8.50 dB	2.0	✓

☑ **RoHs** compliance dependent on attenuator installed. Some attenuators may NOT be compliant.

3. INSTALLATION:

3-1. RACKMOUNTING: The Model 8334 Series is shipped with front panel rack mounting ears that will allow the unit to be mounted in any rack or cabinet that is designed according to EIA RS-310 or MIL-STD-189.

3-2. INITIAL SETUP: The following initial setup procedures should be performed prior to operating the Model 8334 Series .

a. Perform inspection per paragraph 1-4 prior to connecting the 8334 Series to any power source.
b. Install the 8334 Series into a cabinet or rack, as required using customer supplied rack mount slides.
c. Connect all power, RF and system cables as required to the Model 8334 Series.

3-3 INPUT/OUTPUT OPTIONS: The following paragraphs provide a description of the connections that can be made to the Model 8334 Series. Figure 2 shows the location of these connectors and switches.

WARNING

Sufficient power levels are present at the Power Input Assembly to cause personal injury. Ensure that the instrument power cord is DISCONNECTED before attempting to change fuses.

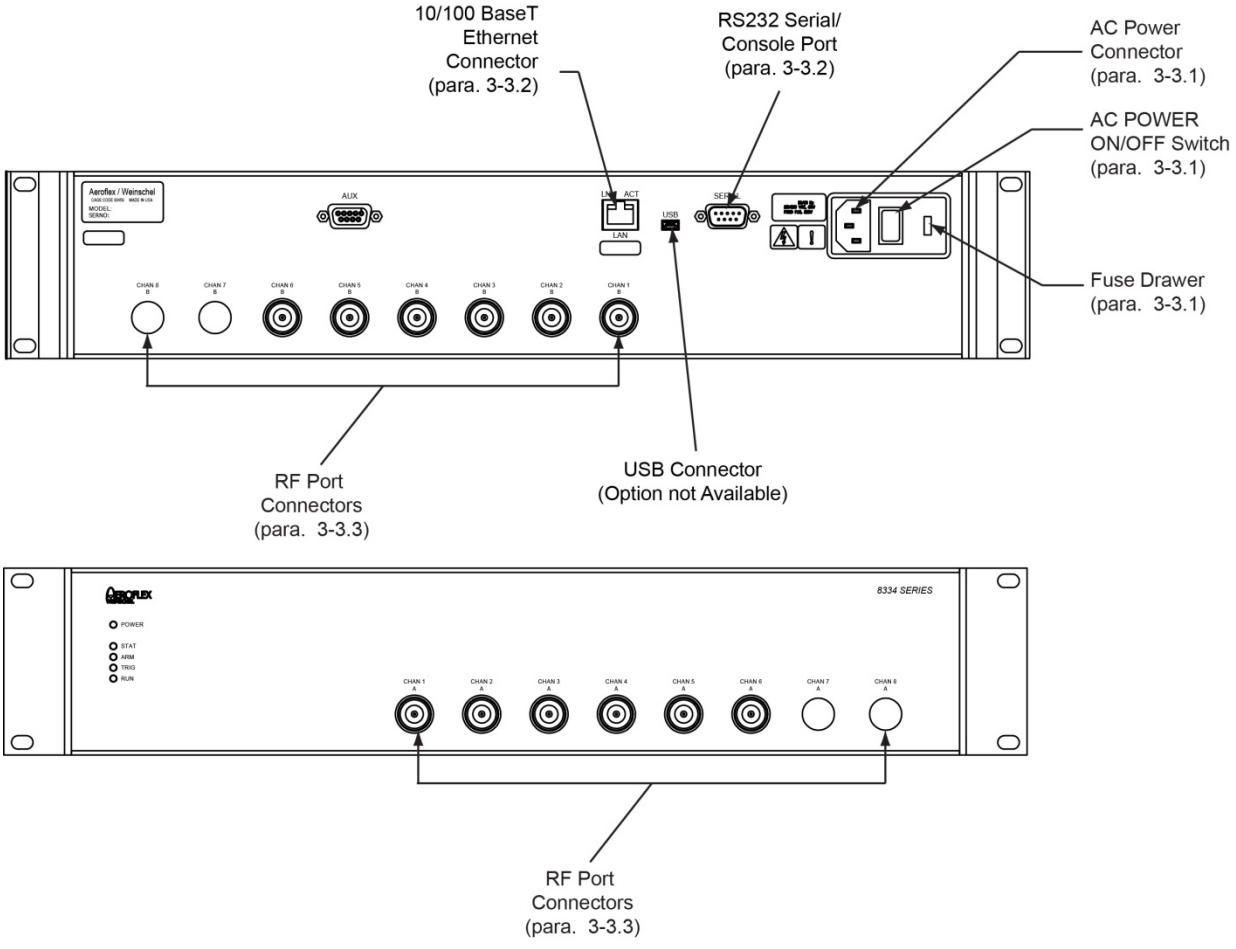


Figure 2. Front/Rear Panel Connectors

3-3.1 POWER ENTRY MODULE ASSEMBLY: The Power Entry Module Assembly located on the rear panel contains a three-prong AC power input connector and a fuse drawer assembly (Figure 2). The **Fuse Drawer Assembly** contains the line voltage fuse (API / Weinschel P/N 052-1-0.5). The Model 8334 Series uses a T 0.5A, 250 Vac fuse which is 5 x 20 mm in size.

The **AC Power Connector**, located on the left side of XF1 (Figure 2), is a plug-type, prong insert connector with three conductors for connection of the power cord (P/N 068-21) to the Power Supply Assembly located within the Unit. This connector also grounds the chassis of the Unit when the ac power cord is connected to a grounded wall outlet. If necessary, use a three prong to two-prong adapter and connect the adapter's ground lead to the outlet plate retaining screw.

The **Power ON/OFF Switch** is located on the front panel and in part of the Power Entry Module Assembly. Placing the POWER ON/OFF switch in the ON position applies power to the instrument.

CAUTION

All electrical rack or chassis and machine elements should be Earth Grounded in installations where high level of electrical noise can be expected. The rack or chassis should be grounded with a rod or attached to a nearby earth structure such as a steel beam support beam. Connect each apparatus to a single ground point in a star configuration with low impedance cable. Scrape away paint and other nonconductive material from the area where a chassis makes contact with the enclosure. In addition to the ground connection made through the mounting bolt or stud, use a one-inch metal braid or size #8 AWG wire to connect between each chassis and the enclosure at the mounting bolt or stud.

3-3.2. CONTROL CONNECTORS: These connectors are located on the middle of the rear panel and consists of one is a standard 9 pin D connector and is identified as Serial, a USB Mini B connector supports USB 2.0 and a standard RJ45 female connector to provide LAN connectivity in support of 10/100baseT. Also there is an AUX connector that is a standard 9 pin D connector (DB9F) which makes available various TTL-level status and control signals. For more details about these connectors and their Pin outs can be located in paragraph 4 (Operation).

CAUTION

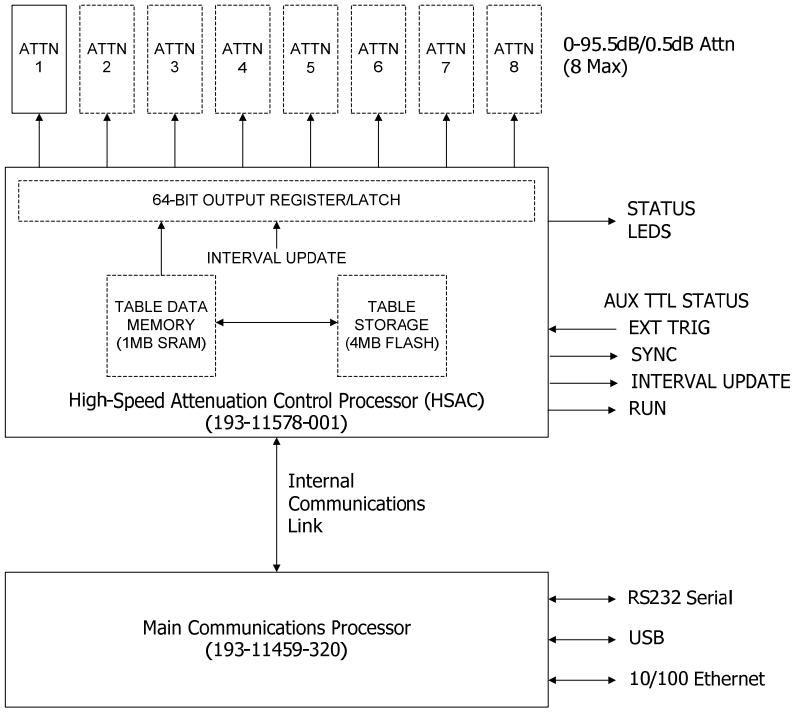
When applying an RF signal to the RF INPUT connector, DO NOT exceed the maximum allowable power level specifications of the unit.

Do not over torque the SMA connectors more than 7 inch pounds. Damage may occur.

3-3.3. RF PORT CONNECTORS: A typical Model 8334 Series Attenuator Unit contains 8 standard D holes on the front and rear panel allowing for single or multichannel configurations. Standard Model 8334 are supplied with either SMA or Type N connectors that can be mounted on the front or rear panel. These connectors provide an input and output port where various types of RF signals can be applied to the devices internally mounted in the Model 8334 (Connector location specified by customer when ordering). These connectors are per MIL-STD-348 interface dimensions and mate nondestructively per MIL-STD-212.

4. OPERATION:

4-1. System Block Diagram



The system is comprised of two main assemblies: the Main Communications Processor (MCP) and the High-Speed Attenuation Control Processor (HSAC). The MCP is responsible for handling communications among the various external interfaces, and provides for control via the RS232 serial, USB, and 10/100Mbit Ethernet interfaces. The HSAC processor is responsible for the data storage and sweep control logic capabilities that interface to the RF attenuators.

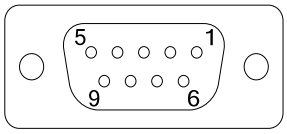
The unit supports up to eight channels of programmable attenuation. The standard unit is supplied with Model 4205-95.5 8-bit MMIC digital step attenuators which provide for a range of 0-95.5dB with 0.5dB resolution. These attenuators are controlled via an output register/latch circuit on the HSAC which allows for simultaneous control of the attenuator settings so that there will be minimal skew between channels as a sweep is performed.

Attenuation control can be performed manually using direct I/O commands (see ATTN command) or via the simulation Table Data memory. Sweep attenuation data is loaded into the Table Data memory of the HSAC which provides for up to 128K points of data per attenuator (1MB total). All sweep parameters, including the table size, sweep/triggering modes, and update interval are user-programmable and can be stored in non-volatile flash memory. Flash table storage allows for up to 4 complete attenuation profiles to be stored locally on the HSAC processor.

When a sweep is performed, information from the Table Data Memory is transferred to the attenuators at a programmable rate, or time interval. This update interval (see INTERVAL command) can be set from 100us to 1sec per data point, and is programmable in 100us increments (the INTERVAL is specified in msecs, so it has a range of 0.1 – 1000.0 in 0.1 msec steps). For a full simulation sweep of 128K data points this provides a sweep duration ranging from 13.1072 secs (@100us) to 131072 secs (@1sec), which is over 36 hours. The INTERVAL setting information is used to setup a hardware-based timer that controls the sweep updates and provides minimal jitter in the rate at which the attenuators are updated.

Sweep modes of operation include single sweep, continuous sweep, and single step mode. Single sweep performs one simulation run and then stops, while continuous sweep mode will automatically repeat the simulation once the end of a sweep is reached until commanded to stop. Single-step mode can be used to manually step through the table one entry at a time under software command (or via an external trigger signal).

Various TTL-level status and control signals are available via the rear-panel AUX connector. These include an EXT TRIG input with programmable polarity, a SYNC output which can be set to provide a programmable width and polarity signal at user-defined simulation points, an INTERVAL UPDATE status output that pulses high for approx 30us at each attenuation update INTERVAL, and a RUN status output that is asserted high for the duration of a simulation sweep.



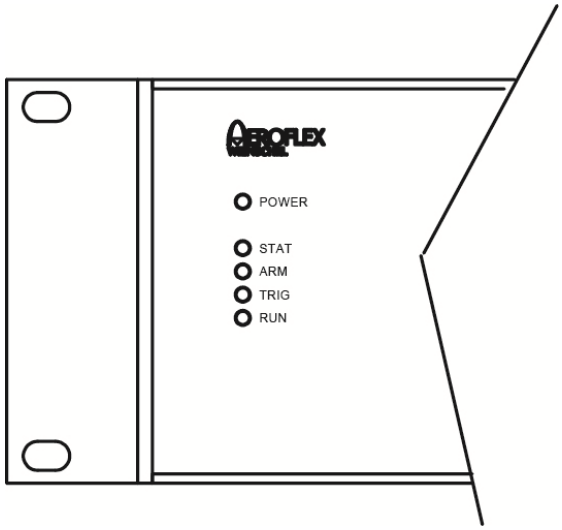
AUX Connector Signals (DE9F)

Pin	Signal	Direction and Type
1	SYNC	Out, 5V CMOS
2	EXT TRIG	In, TTL, 10K pullup to 5V
3	INTERVAL UPDATE	Out, open-drain, 2.5K pullup to 5V
4	RUN	Out, open-drain, 2.5K pullup to 5V
5	unused	
6	GND	Signal Ground
7	GND	
8	GND	
9	GND	

The simulation sweep is controlled via software command, and includes the ability to START, STOP, STEP, PAUSE, and CONT sweep operations (see SWEEP command). The sweep triggering mode is selected using the TRIG MODE command, and supports both internal operation (TRIG MODE INT), or external operation (TRIG MODE EXT). When set for internal triggering, sweep operation is automatic. External trigger mode allows the sweep to be synchronized to external hardware events using the EXT TRIG IN signal, which can be programmed for either active-high or active-low operation (see TRIG POL).

4-2. FRONT PANEL INDICATORS:

Front panel Status LEDs provide an indication of the various operational states and modes. These include:



POWER: Illuminates when power is applied to the unit.

ACT: Indicates MCP communications activity/busy status. Flashes briefly when characters are received via the RS232 serial port, or while the MCP is busy executing a command.

ARM: Illuminates when a sweep is started (SWEEP START) and the sweep control logic is armed waiting for a trigger event. In Internal trigger mode the sweep will automatically transition to the Triggered state, so this LED will flash briefly. In External trigger mode the sweep control logic will wait in this state until an EXT TRIG input signal is detected or the sweep is stopped (SWEEP STOP). This LED is also used to indicate HSAC busy status, and will illuminate when flash memory operations such as a TABLE SAVE or TABLE RECALL operation are in process.

TRIG: Illuminates for one INTERVAL time period when the sweep control logic detects a Trigger event and the sweep transitions to the Running state. As with the ARM state, in Internal trigger mode this transition is automatic, while in External trigger mode this event will occur when the EXT TRIG input signal is asserted.

RUN: Illuminates while the sweep is in the Running state, and will remain on for the duration of the sweep time.

4-3. SIMULATION SWEEP SETUP:

The steps involved in setting up a simulation include

- Set sweep parameters (sweep mode, triggering mode, signal polarities, time interval)
- Set number of simulation points to use (TABLESIZE)
- Load Table Data memory with the attenuation data points
- (optionally save the setup and table data if desired)
- Start the sweep

As an example, the following shows setting up a single 10-point sweep in internal trigger mode. The simulation will be set to update at 100ms intervals. The SYNC output will be set to provide an active-high pulse at data points 3-5 (two INTERVALs in duration). The Table Data memory will be setup assuming there are 4 attenuators, and the data will ramp up in attenuation by 0.5 dB steps, with each attenuator offset by 1dB from each other. The table will be saved in memory #2.

```
SWEEP MODE SING
TRIG MODE INT
INTERVAL 100
SYNC POL 1
SYNC 3 5
TABLESIZE 10
TABLEDATA 0 0.5 1.5 2.5 3.5
TABLEDATA 1 1.0 2.0 3.0 4.0
TABLEDATA 2 1.5 2.5 3.5 4.5
TABLEDATA 3 2.0 3.0 4.0 5.0
TABLEDATA 4 2.5 3.5 4.5 5.5
TABLEDATA 5 3.0 4.0 5.0 6.0
TABLEDATA 6 3.5 4.5 5.5 6.5
TABLEDATA 7 4.0 5.0 6.0 7.0
TABLEDATA 8 4.5 5.5 6.5 7.5
TABLEDATA 9 5.0 6.0 7.0 8.0

TABLE DESC "10-point example”
TABLE SAVE 2

SWEEP START
```

Setting up a sweep and loading the table data can be done either programmatically, or by creating an ASCII text file containing the commands and then using a program such as a Terminal Emulator to transfer the file. This transfer must be done as a plain “send text file” operation. Note that if using an async interface such as the RS232 port, you should have the terminal emulator serial port set with Hardware handshaking enabled (RTS/CTS), or allow sufficient time after each line is sent for the information to be processed. Most terminal emulators include a setting for End Of Line delay when transferring text files. A few msec is typically sufficient.

When using the ASCII text file method, creating a file that loads all 131072 data points will result in a rather large file… approx 7MB in size. Transferring this file will take some time, especially if loaded via the serial port at 115K baud. While the time can be reduced by disabling console mode during the transfer (so there is no echo) , using the short form command TDATA instead of TABLEDATA, removing unneeded decimal points (ie using "10" instead of "10.0"), it will still likely take in excess of 10 minutes to perform the transfer. A faster method of transfer would be to use a TCP connection if the terminal emulator supports it. To do this, connect to the unit using a raw TCP socket to port 10001 (the default TCP server port on the unit) and send the text file using that connection. This should result in transfers of at least twice the speed of the serial port.

The table data can also be sent via the serial port using an XMODEM-CRC transfer protocol, but with this method you are limited to sending only the TABLEDATA information, and the file must be a pure binary data file formatted in the

correct fashion. See the XMODEM TABLEDATA command for more information on the required data format.

All sweep setup and table data commands operate on information contained in the working Table Data Memory located in volatile static SRAM. Once the table data and sweep parameter settings have been programmed, you can save the complete simulation setup in non-volatile flash memory for later recall. The unit provides storage for up to four simulation data sets (see TABLE SAVE/TABLE RECALL), and you can also specify a default table memory to load at startup (see TABLE DEFAULT). Note that when a simulation table is saved, the entire contents of the current Table Data memory (all 128K data point sets, 1MB total) and settings are saved to flash, but that when a recall operation is performed the table header information (which includes the sweep settings and TABLESIZE) is recalled first, and then that TABLESIZE information is used to control how much data is recalled from the flash into the working Table Data memory.

A simple example of using Excel to create and load a data point text file is shown in Paragraph 4-22.

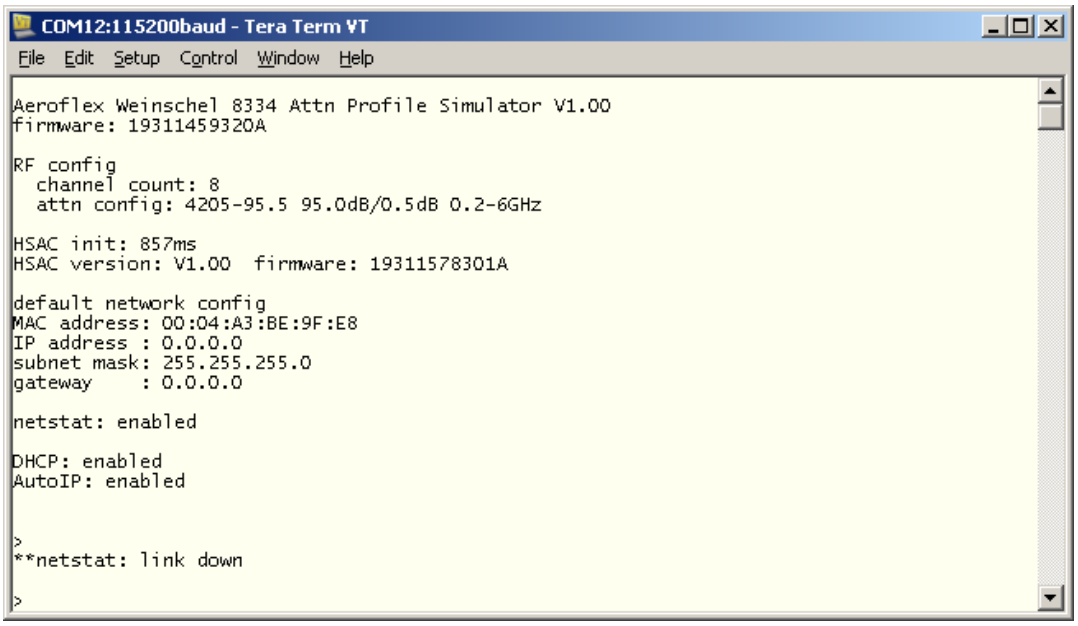


Figure 3. Typical serial console startup screen

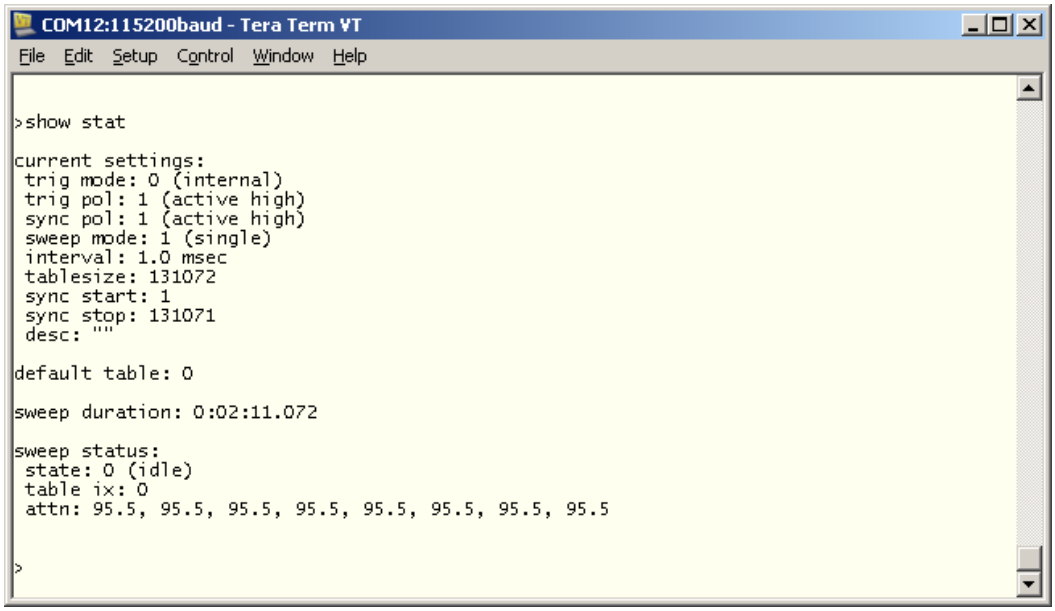
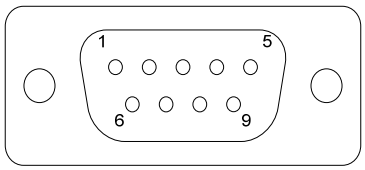


Figure 4. Sweep settings and status display

4-4. RS-232 SERIAL/CONSOLE PORT: The RS232 port is available on a DE9M connector and utilizes three signals: TXD, RXD and GND, along with the optional RTS and CTS handshaking signals. The connector pinout is configured as a DTE device, so connections to an external DTE device (such as a PC) would require the use of a null-modem cable. The format is fixed at N81 (no parity, 8 data bits, 1 stop bit), and the baud rate is selectable via software command at rates of from 9600 to 230400. By default, the unit is shipped to operate at 115K baud with hardware flow control on (RTS/CTS), and with Console mode non-volatile memory (NVM) setting enabled (see below).

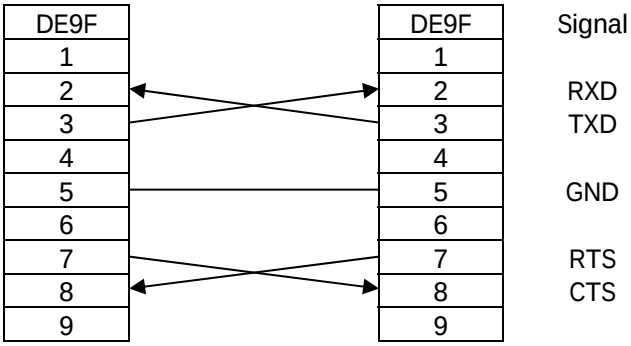
The serial port has two modes of operation: console and raw mode. Console mode provides a command-line interface (CLI) and is useful for interactive control via a terminal emulator. Console mode sends command prompts ('>'), echoes received characters, supports the backspace key for simple editing, provides command line recall (CTRL-R), and issues error messages, while raw mode is more suitable for programming. Console mode may be enabled in one of two ways: either by using the CONSOLE command or by setting the controller card internal CONF DIP switch S1-2 ON. The CONSOLE command stores the enable parameter in non-volatile memory, and is recalled during power up. Setting switch S1-2 ON overrides the non-volatile setting for the baud rate and console mode parameters, allowing the user to force the usage of default parameters (console mode ON and baud rate 115200) if desired. The units are initially shipped with the DIP switch OFF, so console mode and baud rate settings are determined by the NVM setting (see the CONSOLE command for additional options).



RS-232 DE9M (DTE Pinout)

Pin	Signal	Name	Direction
1			
2	RXD	Receive Data	In
3	TXD	Transmit Data	Out
4			
5	GND	Signal Ground	
6			
7	RTS	Request To Send	Out
8	CTS	Clear To Send	In
9			

RS-232 Null-Modem Cable (DTE to DTE)



4-5. 10/100BASET ETHERNET The Ethernet port supports 10/100BaseT operation, with auto-negotiation of the interface speed and duplex mode. LED indicators are provided to indicate network LINK status (green) and TX/RX activity (YELLOW). Supported network protocols include: IP, UDP, TCP, ICMP, ARP, DHCP, and AUTOIP. A TELNET server is provided for a command-line interface that implements many of the functions of the serial console CLI. Both TCP and UDP servers allow connections to be established for general programming purposes. Additionally, the Microchip Announce protocol is implemented to provide support for the Microchip Ethernet Discovery tool, which is a UDP-based protocol used to detect ethernet devices on the network.

IP addressing modes supported include the use of a statically assigned fixed address, or dynamic address assignment using either DHCP or AutoIP. The use of DHCP requires a DHCP server to reside on the network. AutoIP is an address mode that can be used when no DHCP server is available. It automatically allocates an address from the special block of addresses 169.254.1.0 to 169.254.254.255 reserved for link-local addressing. These addresses are only valid on the link that the host is connected to, such as a local network segment or point-to-point connection, and are unroutable. See the SET IPADDR, SET DHCP, SET AUTOIP, and SHOW IPADDR commands for more information.

The TELNET server communicates using the standard port 23 typically used by the TELNET protocol. The implementation is a reduced-functionality version and does not support the full protocol, but it should function properly with many clients. The server only supports a single connection, does not require any login, and does not support options negotiation except for the initial state of the echo setting. The server does support an inactivity timeout, and allows the use of TELNET NOP commands sent by the client to keep a session open. By default, the timeout is set for 300 seconds (5 minutes), after which the server will automatically close the session if no activity has occurred. Many of these features are configurable by the user (see SET TELNET in the command reference). The status of the server can be seen using the SHOW NET TELNET command. An active TELNET connection may be closed from the telnet application on the client using the 'QUIT' command.

The unit provides a TCP server that can be used for control and status of the unit using the same text-based messages used by the serial port. By default, the server is configured to listen on a single port (port 10001), and supports a single connection. The server can support multiple ports, each with multiple connections for multi-user environments, however this functionality must be configured by the factory. Server settings, such as the port number, keepalive timeout, inactivity close timer, and character echoing are programmable by the user (see SET TCP in the command reference). The status of the server(s) can be seen using the SHOW NET TCP command.

A UDP server is also provided that will accept command messages sent via UDP protocol using the same text-based messaging. UDP is a connection-less based protocol that is simpler and has less overhead than TCP. By default, the internal UDP server listens on port 20000, but this can be changed via the SET UDP SERVER command.

Various network events generate status messages shown on the serial Console port. These events include TCP and TELNET server connect/disconnect messages and DHCP/AUTOIP address assignment changes. The status messages may be disabled if desired (see SET NETSTAT), but are enabled by default.

For simple device discovery, the Microchip Announce protocol is used. The Microchip Announce protocol is a UDP-based scheme used to detect devices supporting the protocol. The protocol broadcasts UDP packets to port 30303 containing the message, "Discovery: Who is out there?", and supporting devices respond with a UDP packet which provides the device IP and MAC address, as well as other info such as the firmware version. A copy of the Ethernet Device Discoverer application for MS Windows-based systems is provided on the CD supplied with the unit, or can be downloaded from the Microchip website.

4-6. USB: The USB port provides a USB Communications Device Class device (CDC) interface that allows programming via a virtual COM port using the same text-based commands as the serial port. A console mode command-line interface (CLI) is provided for ease of use, and is user-configurable via the USB CONSOLE command.

See paragraph 4-20 for information on installing the USB CDC driver file.

4-7. COMMAND OPERATION: Commands are comprised of text-based ASCII strings. The command parser is case-insensitive, so either upper or lower case characters are acceptable. Command parameters may be separated with either an ASCII SPACE char (0x20) or an ASCII COMMA char (0x2E), but the separator character used must be the same within an individual command string. Additional SPACE characters are ignored. Typically, input program messages may be terminated using either an ASCII CR character (0x0D) or an ASCII LF character (0x0A), however this can be changed by using the SET EOS command. Command message strings are limited to 128 characters total, including the terminator. Multiple commands can be included in one message by separating the individual commands with an ASCII SEMICOLON character ';' (0x3B), up to the 128 character message limit. Response messages are terminated differently depending on the source of the command. Response messages sent over the serial and USB CDC ports default to using both a CR (0x0D) and LF (0x0A) to terminate the line, while messages sent via a network TCP or UDP connection default to using a single CR (0x0D) terminator. The output terminator sequence may be changed using the SET EOS command. A list of supported commands can be seen by typing 'HELP' or '?' at the Console prompt.

The command structure/operation is similar to that used in IEEE 488.2, and includes some of the 488.2 Common Commands such as *IDN?, *RST, *CLS, and *OPC?, in addition to device specific commands. In 488.2, programming commands take one of two forms: a Program message or a Query message. Program messages are used to send commands to the device, while Query messages are used to elicit a response. Query commands are those that contain a '?' character. In general, the device does not generate any response to a program message unless the message contains a valid Query command. (Note that this does not apply when operating in Console mode, or when using some commands such as SHOW which are designed to provide the user general information). You can use this feature to provide a method to synchronize command execution with the controller by appending a Query to the desired command, and waiting for the response. For example, sending "*CLS;*OPC?" will place a "1" in the output queue when the *CLS command has been executed. Query commands that return multiple values will have the values separated by an ASCII COMMA character (0x2E). If multiple Query commands are included in the same message, the individual query responses will be separated with an ASCII SEMICOLON character (0x3B).

An Error Queue is provided that logs the results of command/execution errors in a FIFO fashion. The queue entries can be read using the ERR? command, which returns both an error code and a descriptive text message, such as

101, "invalid command"

When the queue is empty, ERR? returns the message **0, "no error"**. The queue can be emptied by repeatedly sending ERR? until all entries are read from the queue, or via sending the *CLS message. Note: There is a single Error Queue shared by all the command interfaces, such as the network socket connections, TELNET, USB, and the serial Console. Since the TELNET, USB and serial Console interfaces operate in an interactive fashion, if you are using multiple interfaces simultaneously the error messages may not appear on the expected interface. For example, errors generated by messages sent to the TCP server port may be shown on the serial console if it is in active use.

Unless otherwise specified, commands revert to their default setting at system reset/poweron, with the exception of the system setup and configuration commands (see SET). The various SET commands are used to update the settings in non-volatile memory (NVM), and do not typically take effect until the next system reset event unless otherwise noted.

4-8. COMMAND EXECUTION AND BUFFERING: Typically, simple commands execute in 1-2 msecs, however certain commands such as switching an electromechanical relay-based attenuator may take significantly longer than this. During this time, input commands are buffered for later execution. Buffering typically provides space for approximately 20 commands, but this is command and interface dependent. When operating at fast communication rates it is possible to exceed the buffering ability causing commands to be missed. For interfaces such as RS232, you can use hardware flow control to prevent this from occurring. For other interfaces, you can use a command/query scheme as a synchronization method.

4-9. COMMAND REFERENCE: In the command descriptions that follow, argument types are described using the following additional conventions to indicate the relative size of the parameter:

byte	- used to indicate an 8-bit unsigned integer
word	- used to indicate a 16-bit unsigned integer
int8	- 8-bit integer
int16	- 16-bit integer
int32	- 32-bit integer
string	- character data, including the max number of characters allowable. (ie string8 has a max of 8 chars)

Numeric arguments default to decimal (base 10) notation, but may optionally be provided in hex or binary if appropriate by using a "0x" prefix for hex or "0b" for binary. In addition, commands that accept a '0' or '1' argument will also accept the text strings 'OFF' and 'ON' in place of the numeric parameter. For example, "CONSOLE 1" and "CONSOLE ON" are equivalent.

Required command keywords are shown in CAPITAL letters, and arguments are shown in *italics*. Square brackets '[]' may be used to indicate an optional parameter, for example [*select*]. Optional parameters, if not supplied by the user, assume the default setting specified in the text

4-10. APPLICATION SPECIFIC COMMANDS: The HSAC hardware provides support for up to eight attenuators. Many commands will allow you to specify values for all eight attenuators even though the actual number of attenuators installed in the unit may be less. In the commands below, *N* is used to indicate the number of installed attenuators.

4-10.1. Attenuation Commands (Direct I/O)

ATTN
Function: set attenuator(s)
Syntax: ATTN *select setting*
ATTN ALL *setting*
ATTN ALL *setting1 setting2... settingN*
Argument(s): *select* attenuator select 1-*N* or ALL
setting attenuator setting, in dB. s*etting*=0-max attenuation value
Remarks: This command sets the specified RF attenuator(s) to the dB value provided by *setting*. If *select* is specified as ALL then the synchronous output hardware is used, and all attenuators will change simultaneously. In addition, ALL allows specifying a list of up to *N* individual dB settings. This command will cause any active sweep to stop.
Return Value: none
Example(s):

```
ATTN 1 10                       // sets attn 1 to 10 dB
ATTN ALL 20.5                   // sets all attenuators to 20.5 dB
ATTN ALL 10 20 30 40 50 60      // sets attn 1 to 10dB, attn 2 to 20dB, etc (up to N values)
```

ATTN?
Function: read attenuator setting
Syntax: ATTN? *select*
Argument(s): *select* attenuator select 1-*N* or ALL
Remarks: This command returns the current setting of the specified attenuator
Return Value: attenuator setting, in dB
Example(s):

```
ATTN ALL 10 20 30 40 50 60      // sets attn 1 to 10dB, attn 2 to 20dB, etc
ATTN? 1                         // request attn 1 setting
10.0                            // returns attn 1 setting (10 dB)
ATTN? ALL                       // request all attn settings
10.0, 20.0, 30.0, 40.0, 50.0, 60.0   // returns all attn settings (up to N values)
```

4-10-2. Sweep Setup Commands:

TABLESIZE
Function: set the data table max size
Syntax: TABLESIZE *num_points*
Argument(s): *num_points* table size, 1 - 131072 (128K)
Remarks: This command sets the size of the data table (the number of points in use). This value is used as the end index for a sweep. Table index values are zero-based, so they range from 0 to *num_points* - 1. This parameter has a default value of 131072, and is stored in NVM when a TABLE SAVE operation is performed
Return Value: none
Example(s):

```
TABLESIZE 100          // sets tablesize to 100 points (0-99)
```

TABLESIZE?
Function: returns data table size
Syntax: TABLESIZE?
Argument(s): none
Remarks: This command returns the current number of points in use (TABLESIZE setting)
Return Value: 1-131072
Example(s):

```
TABLESIZE 1000         // sets tablesize to 1000 points (0-999)
TABLESIZE?             // request current number of points
1000                   // returns TABLESIZE
```

TABLEDATA
TDATA
Function: set table data point attenuation values
Syntax: TABLEDATA *ix db1 db2 ... db8*
Argument(s): *ix* data point index, 0-131071
db1...db8 list of attenuation values
Remarks: This command sets a table data point attenuation value settings, where *db1* = attn 1, *db2* = attn 2, up to the max number of supported attenuators. Any unspecified values will be set to 0, so you only need to specify values for the number of attenuators installed in the unit. The keyword TDATA may also be used in place of TABLEDATA. This command will cause any active sweep to stop.
Return Value: none
Example(s):

```
TABLEDATA 0 10.5 11.0 11.5 12.0 12.5 13.0    // sets data point 0 attn1-attn6 values (attn7-attn8 set to 0)
TDATA 131071 20 30.5                         // sets data point 131071 attn1-attn2 values (others set to 0)
```

TABLEDATA?
TDATA?
Function: read a table data point attenuation values
Syntax: TABLEDATA? *Ix*
Argument(s): *ix* data point index, 0-131071
Remarks: This command returns the table data point attenuation values for the specified table index *ix*. It will return values for all eight attenuators. The keyword TDATA? may also be used in place of TABLEDATA?. This command will cause any active sweep to stop.
Return Value: eight attenuation data values
Example(s):

```
TABLEDATA 0 10.5 11.0 11.5 12.0 12.5 13.0    // sets data point 0 attn1-attn6 values (attn7-attn8 set to 0)
TABLEDATA? 0                                 // request data point 0
10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 0.0, 0.0 // returns all eight attn values for table index 0
```

INTERVAL
Function: set sweep update interval
Syntax: INTERVAL *msecs*
Argument(s): *msecs* update interval time, in msecs. 0.1 – 1000.0 in 0.1 msec steps (100us)
Remarks: This command sets the sweep update interval time, in msecs. This parameter has a default value of 1, and is stored in NVM when a TABLE SAVE operation is performed.
Return Value: none
Example(s):

```
INTERVAL  1          // sets update interval to 1 msec/point
INTERVAL  100.2      // sets update interval to 100.2 msec/point
```

INTERVAL?
Function: read sweep update interval
Syntax: INTERVAL?
Argument(s): none
Remarks: This command returns the current sweep update interval time, in msecs.
Return Value: *msecs*
Example(s):

```
INTERVAL  100.2      // sets update interval to 100.2 msec/point
INTERVAL?            // request interval
100.2                // returns current setting
```

TRIG MODE
Function: set sweep trigger mode
Syntax: TRIG MODE *select*
Argument(s): *select* triggering mode. 0 (or INT) and 1 (or EXT)
Remarks: This command sets the sweep trigger mode for either internal or external sweep control. The *select* parameter accepts the strings INT and EXT in addition to the values 0 and 1. When the trig mode is set for internal operation, sweeping is controlled via software command, while external mode relies on the TRIG IN TTL signal on the AUX rear-panel DE9 connector. This parameter has a default value of 0 (internal), and is stored in NVM when a TABLE SAVE operation is performed.
Return Value: none
Example(s):

```
TRIG MODE 0          // sets triggering to internal
TRIG MODE EXT        // sets triggering for external operation
```

TRIG? MODE
Function: read sweep trigger mode
Syntax: TRIG? MODE
Argument(s): none
Remarks: This command returns the current sweep trigger mode.
Return Value: 0 (internal), 1 (external)
Example(s):

```
TRIG MODE INT        // sets trig mode to internal
TRIG? MODE           // request trig mode
0                    // returns current setting (0=internal, 1=external)
```

TRIG POL
Function: set external trigger in polarity
Syntax: TRIG POL *level*
Argument(s): *level* ext trigger polarity. 0=active-low, 1=active-high
Remarks: This command sets the polarity of the external sweep trigger TRIG IN TTL signal, allowing the user to select between an active-high or active-low trigger. This parameter is only used when TRIG MODE EXT is selected, and is ignored when trigger mode is set for internal operation. The TRIG IN TTL signal is available on the AUX rear-panel DE9 connector. There is an internal 10K pullup to 5V on the TRIG IN signal, so the signal will be seen high if there is no connection to this pin. This parameter has a default value of 1 (active-high), and is stored in NVM when a TABLE SAVE operation is performed.
Return Value: none
Example(s):

```
TRIG POL 0          // sets active-low external trigger
TRIG POL 1          // sets active-high external trigger
```

TRIG? POL
Function: read sweep trigger polarity
Syntax: TRIG? POL
Argument(s): none
Remarks: This command returns the current ext trigger in polarity setting.
Return Value: 0 (active-low), 1 (active-high)
Example(s):

```
TRIG POL 0          // sets ext trig polarity to 0 (active-low)
TRIG? POL           // request trig polarity
0                   // returns current setting (0=low, 1=high)
```

SYNC
Function: set external SYNC output point
Syntax: SYNC *start_ix [end_ix]*
Argument(s): *start_ix* ext SYNC output starting table index. 1-131071
end_ix ext SYNC output ending table index (optional)
Remarks: This command controls the timing of the external SYNC output signal. The SYNC output will be asserted when the active sweep current table index matches the *start_ix* parameter, and will remain asserted until the *end_ix* is reached or the sweep ends. If the optional *end_ix* value is not specified, then *end_ix* will automatically be set to *start_ix* + 1, providing a pulse 1 INTERVAL time in duration. The SYNC OUT TTL signal is available on the AUX rear-panel DE9 connector. This parameter has a default value of 1, 131071, and is stored in NVM when a TABLE SAVE operation is performed.
Return Value: none
Example(s):

```
SYNC 10             // assert SYNC from index 10 to 11 (1 INTERVAL in duration)
SYNC 100 120        // assert SYNC from index 100 to 120 (20 INTERVAL in duration)
```

SYNC?
Function: read SYNC output points
Syntax: SYNC?
Argument(s): none
Remarks: This command returns the current SYNC output index points.
Return Value: start_ix, end_ix
Example(s):

```
SYNC 10          // assert SYNC from index 10 to 11 (1 INTERVAL in duration)
SYNC?            // request SYNC points
10, 11           // returns sync start and end points
```

SYNC POL
Function: set external SYNC output polarity
Syntax: SYNC POL *level*
Argument(s): *level* ext SYNC polarity. 0=active-low, 1=active-high
Remarks: This command sets the polarity of the external SYNC output TTL signal, allowing the user to select between an active-high or active-low sync pulse. The SYNC OUT TTL signal is available on the AUX rear-panel DE9 connector. This parameter has a default value of 1 (active-high), and is stored in NVM when a TABLE SAVE operation is performed.
Return Value: none
Example(s):

```
SYNC POL 0       // sets active-low external SYNC output
SYNC POL 1       // sets active-high external SYNC output
```

SYNC? POL
Function: read SYNC output polarity
Syntax: SYNC? POL
Argument(s): none
Remarks: This command returns the current SYNC output polarity setting.
Return Value: 0 (active-low), 1 (active-high)
Example(s):

```
SYNC POL 0       // sets active-low external SYNC output
SYNC? POL        // request sync polarity
0                // returns current setting (0=low, 1=high)
```

SWEEP MODE
Function: set sweep mode
Syntax: SWEEP MODE *select*
Argument(s): *select* sweep mode. 0 (or CONT), 1 (or SING), 2 (or STEP)
Remarks: This command sets the sweep mode, allowing the user to select between continuous, single, or step mode operation. The *select* parameter accepts the strings CONT, SING, and STEP in addition to the values 0, 1 and 2. This parameter has a default value of 1 (single), and is stored in NVM when a TABLE SAVE operation is performed.
Return Value: none
Example(s):

```
SWEEP MODE 0          // sets continuous sweep mode
SWEEP MODE STEP       // sets step sweep mode
```

SWEEP? MODE
Function: read sweep mode setting
Syntax: SWEEP? MODE
Argument(s): none
Remarks: This command returns the current SWEEP MODE setting.
Return Value: 0, 1, or 2 (see SWEEP MODE)
Example(s):

```
SWEEP MODE CONT                  // sets continuous sweep mode
SWEEP? MODE                      // request sweep mode
0                                // returns current setting (0, 1, or 2)
```

4-10-3. Sweep Control/Status Commands:

SWEEP
Function: sweep state control
Syntax: SWEEP *ctrl*
Argument(s): *ctrl* sweep state control, consisting of:
START
STOP
PAUSE
CONT
STEP
Remarks: This command controls sweep operation. To view the current status of a sweep, you can use the SWEEP? or SHOW SWEEP commands.
Return Value: none
Example(s):

```
SWEEP START          // starts a sweep
SWEEP STOP           // stops a running sweep
```

SWEEP?
Function: returns the current sweep status
Syntax: SWEEP?
Argument(s): none
Remarks: This command returns the current sweep status, including the state and current table index.
Return Value: sweep_state, table_ix
Where sweep_state is an integer value from 0-6 with the following meaning:
0 = SWEEP_IDLE
1 = SWEEP_INIT
2 = SWEEP_STARTED
3 = SWEEP_ARMED
4 = SWEEP_TRIGGERED
5 = SWEEP_RUNNING
6 = SWEEP_PAUSED
table_ix is the current table data point index. 0-131071
Example(s):

```
SWEEP START          // starts a sweep running
SWEEP?               // request sweep status
5, 101               // state=5 (RUNNING), table_ix=101
SWEEP PAUSE
SWEEP?               // request sweep status
6, 558               // state=6 (PAUSED), table_ix=558
SWEEP CONT
SWEEP STOP
SWEEP?               // request sweep status
0, 623               // state=0 (IDLE), table_ix=623
```

4-10.4. Table Memory Commands:

TABLE SAVE
TABLE RECALL
TABLE ERASE
Function: table flash memory operations
Syntax: TABLE *oper mem_num*
Argument(s): *oper* table memory operation (SAVE, RECALL, or ERASE)
mem_num memory number. 1-4
Remarks: This command performs table memory operations on the flash memory region specified by *mem_num*, which can specify flash memory regions 1-4. TABLE SAVE copies the current working tabledata and sweep parameters to flash memory, TABLE RECALL loads the working memory (including both tabledata and sweep parameters) from a previously saved flash region, and ERASE clears the specified flash tabledata region. Flash memory operations can take several seconds to complete. Some typical execution times are:

TABLE SAVE: 8-9 secs
TABLE RECALL: 2-3 secs
TABLE ERASE: 500ms

Note that a TABLE SAVE operation will automatically perform a TABLE ERASE on the specified flash tabledata region, so this does not need to be performed as a separate step. Also note that a TABLE ERASE operation only clears the tabledata flash memory, and does not affect the table header (sweep parameters) information.
Return Value: none
Example(s):

```
TABLE SAVE 2        // save current data into table 2
TABLE RECALL 3      // recall previously saved table into working mem
```

TABLE DESC
Function: set a table description string
Syntax: TABLE DESC *string*
Argument(s): *string* user-defined ASCII string, 1-22 chars in length.
Remarks: This command can be used to set a user-defined description string that will be stored along with the tabledata and sweep parameters when a TABLE SAVE operation is performed. The string value should be delimited using either the ASCII Double-Quote (0x22, 34 dec) or Single-Quote (0x27, 39 dec) characters.
Return Value: none
Example(s):

```
TABLE DESC "192 point table"        // sets description string
```

TABLE DEFAULT
Function: set default table number
Syntax: TABLE DEFAULT *mem_num*
Argument(s): *mem_num* memory number. 1-4, or 0
Remarks: This command sets the default table number to automatically load at startup. Specifying a value of 0 disables the auto load function. This parameter has a default value of 0 (disable autoload), and is saved to NVM.
Return Value: none
Example(s):

```
TABLE DEFAULT 2     // set autoload table 2 at startup
```

TABLE FILL
Function: fills table data working memory
Syntax: TABLE FILL *val*
Argument(s): *val* byte fill value
Remarks: This command fills the contents of the working tabledata memory with the specified byte value (0-255). It does not alter any flash memory contents. Note that the value is not an attenuation value, but a raw 8-bit data value.
Return Value: none
Example(s):

```
TABLE FILL 0        // fill tabledata with 0
```

TABLE DUMP
Function: displays tabledata working memory
Syntax: TABLE DUMP *start_ix [end_ix]*
Argument(s): *start_ix* starting table data point index
enc_ix ending table data point index (optional)
Remarks: This command displays the raw table data memory for the specified data point index range. If *end_ix* is not specified, then *end_ix* will be set to display 16 sets of tabledata. Each data point displayed is the raw 8-bit table data value for each of the eight attenuators at that point.
Return Value: none
Example(s):

```
>TABLE DUMP 0 10

0x00000:  00 00 00 00 00 00 00 00
0x00001:  01 01 01 01 01 01 01 01
0x00002:  02 02 02 02 02 02 02 02
0x00003:  03 03 03 03 03 03 03 03
0x00004:  04 04 04 04 04 04 04 04
0x00005:  05 05 05 05 05 05 05 05
0x00006:  06 06 06 06 06 06 06 06
0x00007:  07 07 07 07 07 07 07 07
0x00008:  08 08 08 08 08 08 08 08
0x00009:  09 09 09 09 09 09 09 09
0x0000A:  0A 0A 0A 0A 0A 0A 0A 0A
```

XMODEM TABLEDATA
Function: XMODEM table data download
Syntax: XMODEM TABLEDATA
Argument(s): none
Remarks: This command can be used to perform an XMODEM-CRC download of raw table data via the RS232 serial port. The tabledata file must be an unformatted binary data file containing eight raw values for each tabledata point, and can be up to a maximum of 1MB in size (128K points x 8 bytes/point). The binary data for each attn value is a single byte representing the attenuation value for that point, and is dependent on the installed attenuator. For a 4205-95.5dB attn, the bit values (7-0) represent: 32dB, 32, 16, 8, 4, 2, 1, 0.5dB. This command is primarily for use with an external terminal emulator capable of performing XMODEM-CRC transfers via the serial RS232 port.
Example(s):

```
>xmodem tabledata
begin xmodem-crc download...CC                    // initiate XMODEM-CRC SEND via terminal
emulator
transfer complete. 8192 packets (1048576 bytes)
elap time: 147.9 sec
```

Binary file format:

File Offset	0	1	2	3	4	5	6	7
0	ATTN1	ATTN2	ATTN3	ATTN4	ATTN5	ATTN6	ATTN7	ATTN8
8	ATTN1	ATTN2	ATTN3	ATTN4	ATTN5	ATTN6	ATTN7	ATTN8
16								
<…>	ATTN1	ATTN2	ATTN3	ATTN4	ATTN5	ATTN6	ATTN7	ATTN8
datapoint N-1	ATTN1	ATTN2	ATTN3	ATTN4	ATTN5	ATTN6	ATTN7	ATTN8

4-11. SHOW Commands: SHOW commands provide a method of viewing a variety of system settings and information. Unlike regular query commands, they are primarily meant for interactive CLI usage such as the serial console mode or telnet, as the contents contain multiple lines of formatted information, and may change depending on the current mode, settings, and installed implementation features.

SHOW TABLE
Function: displays table header (sweep parameter) information
Syntax: SHOW TABLE [1-4]
Argument(s): none (displays current settings in memory), or table number 1-4
Remarks: This command can be used to display the table header (sweep parameter) information. SHOW TABLE will display the current settings actively in use in memory, while SHOW TABLE n will display the parameters for the specified flash table memory region.

```
>SHOW TABLE

current settings:
 trig mode: 0 (internal)
 trig pol: 1 (active high)
 sync pol: 1 (active high)
 sweep mode: 1 (single)
 interval: 1.0 msec
 tablesize: 131072
 sync start: 1
 sync stop: 131071
 desc: ""

default table: 0

>SHOW TABLE 1

table header 1:
 trig mode: 0 (internal)
 trig pol: 1 (active high)
 sync pol: 1 (active high)
 sweep mode: 1 (single)
 interval: 100.0 msec
 tablesize: 192
 sync start: 1
 sync stop: 2
 desc: "192x8 points"
```

SHOW SWEEP
Function: displays sweep status
Syntax: SHOW SWEEP
Argument(s): none
Remarks: This command can be used to display sweep status information, including the current sweep state, table data point index, and current attn settings

```
>SHOW SWEEP
sweep status:
 state: 0 (idle)
 table ix: 0
 attn: 95.5, 95.5, 95.5, 95.5, 95.5, 95.5, 95.5, 95.5
```

SHOW STAT
Function: displays current status
Syntax: SHOW STAT
Argument(s): none
Remarks: This command can be used to display the current status, including the SHOW TABLE, sweep duration, and SHOW SWEEP information.

```
>SHOW STAT

current settings:
 trig mode: 0 (internal)
 trig pol: 1 (active high)
 sync pol: 1 (active high)
 sweep mode: 1 (single)
 interval: 100.0 msec
 tablesize: 192
 sync start: 1
 sync stop: 2
 desc: "192x8 points"

default table: 0

sweep duration: 0:00:19.200

sweep status:
 state: 0 (idle)
 table ix: 0
 attn: 95.5, 95.5, 95.5, 95.5, 95.5, 95.5, 95.5, 95.5
```

4-12. Misc Setup Commands:

SHOW RFCONFIG
Function: displays current RF configuration
Syntax: SHOW RFCONFIG
Argument(s): none
Remarks: This command displays the current RF hardware configuration
Example(s):

```
>show rfconfig
RF config
  channel count: 6
  attn config: 4205-95.5 95.0dB/0.5dB 0.2-6GHz
```

SET RFCONFIG
Function: set rf hardware installation
Syntax: SET RFCONFIG CHAN *numchannels* // set the number of installed attenuator channels
SET RFCONFIG ATTN *type* // sets the installed attenuator type
Remarks: This command can be used to change the RF configuration to support different chassis configurations
Return Value: none
Example(s):

```
SET RFCONFIG CHAN 6    // sets number of installed attn channels to 6
```

RFCONFIG?
Function: read chassis configuration items
Syntax: RFCONFIG? CHAN // returns the number of installed attenuator channels
RFCONFIG? ATTN *n* // returns configuration info for attn *n* (model, range, etc)
Remarks: These commands can be used to retrieve various chassis configuration settings. For RFCONFIG? ATTN, the command returns the attn type, max attenuation, stepsize, switching speed (msec), cycle rate (msec), and a descriptive string.

HSAC CONSOLE
Function: redirect RS232 serial port
Syntax: HSAC CONSOLE
Argument(s): none
Remarks: This command redirects the main rear-panel RS232 serial port, and connects it to the serial port of the HSAC processor. When running the CLI of the HSAC, the command prompt changes from '>' to '$'. This command is used primarily for troubleshooting and diagnostic usage, or to perform updates of the HSAC firmware. To exit redirect mode, send an RS232 BREAK signal or cycle power.
Example(s):

```
>HSAC CONSOLE
$*IDN?
API Weinschel, HSAC, 0, V1.00

$
```

4-13. 488.2 COMMON COMMANDS:

***CLS**
Function: clears the error status
Syntax: *CLS
Argument(s): none
Remarks: This function is used to clear the Error Queue
Return Value: none
Example(s):
*CLS

***IDN?**
Function: Reads the system identification information
Syntax: *IDN?
Argument(s): none
Remarks: This function is used to read the system identification info, which is a string consisting of the following data: manufacturer, model, serial number, and firmware version.
Return Value: *idstr* string id info
Example(s):
*IDN?
API Weinschel, 6990, 002, V1.00

***OPC?**
Function: Operation complete query
Syntax: *OPC?
Argument(s): none
Remarks: This function loads a '1' into the output queue when the Program Message Unit is executed. It's primary use is to provide an indication of command completion by including the command as the last one in a series of commands. It can be useful to synchronize operation and to prevent input buffer overflow.
Return Value: 1 integer constant command completed
Example(s):
CMD1 1; CMD2 2; *OPC?
1 // sends a '1' response when the three commands have been parsed and executed

***ESR?**
Function: Event Status Register query
Syntax: *ESR?
Argument(s): none
Remarks: This function reads the 488.2 Event Status Register. Reading the register also clears it.
Return Value: *int8* integer status register
Example(s):
*ESR?
32 // indicates a Command Error

***RST**
Function: Performs a device application level reset.
Syntax: *RST
Argument(s): none
Remarks: This function is used to reset the device application settings. For a full device reset, see the REBOOT command.
Return Value: none
Example(s):
*RST

***TST?**
Function: Self-test query
Syntax: *TST?
Argument(s): none
Remarks: This function performs an internal self-test. Upon completion, the results of the test are loaded into the output queue.
Return Value: *testresults* integer '0' indicates test passed. Non-zero indicates test failed.
Example(s):

```
*TST?
0                    // returns a '0' when the test completes successfully.
```

ERR?
Function: Read the Error Queue
Syntax: ERR?
Argument(s): none
Remarks: This function returns the last entry in the error status queue, and a string description of the error code. Repeating the command will return the next entry, until the error queue is empty and returns a zero. The error queue may be cleared via the *CLS command. Note that when using the command-line interface the Error Queue contents are automatically displayed after each command prior to issuing the CLI prompt.
Return Value: error number, "error description"
Example(s):

```
ERR?
101, "invalid command"
ERR?
0, "no error"
```

4-14. Setup and Configuration Commands:

NOTE: The SET commands are used to update settings which are stored in non-volatile memory (NVM), and do not typically take effect until the next system restart event (see REBOOT) unless otherwise noted. The settings listed here are dependent on the installed hardware, so not all settings are available on some implementations.

SET EOS
Function: sets the Program Message Terminator and/or Response Message Terminator end of string characters
Syntax: SET EOS *interface inout val*
Argument(s): *interface* protocol selection SERIAL, USB, GPIB, TCP, UDP, or ALL
inout PMT (input) or RMT (output)
val word, eos characters
Remarks: This function sets the input Program Message Terminator (PMT) or the output Response Message Terminator (RMT) sequences. Each communications port/protocol can have separate definitions. The *val* parameter specifies the character sequence used, and can specify up to two characters, typically as a hex word high byte-low byte pair. Common definitions for the terminators include the ASCII CR (0x0D) and LF (0x0A) characters. A single character may be specified either by using 0 for the high byte, such as 0x000D, or by only specifying a single character (ie 0x0D). On input, the message will terminate when either of the two character codes are received, while for output the characters are sent low byte then high byte, unless it is specified as 0. Note that the serial CONSOLE and network TELNET servers are excluded from this selection and always use a fixed CRLF (0x0A0D) sequence. The current settings may be viewed using the SHOW EOS command.
Return Value: none
Example(s):

```
SET EOS SERIAL PMT 0x0A0D     // set serial input to terminate on either CR or LF
SET EOS SERIAL RMT 0x0A0D     // set serial output sequence as CR-LF
SET EOS USB PMT 0x0A0D        // set usb input to terminate on either CR or LF
SET EOS USB RMT 0x0A0D        // set usb output sequence as CR-LF
SET EOS TCP PMT 0x0A0D        // set tcp socket input to terminate on either CR or LF
SET EOS TCP RMT 0x0D          // set tcp output sequence as a single CR character
SET EOS UDP PMT 0x0A          // set udp socket input to terminate on Lf character only
SET EOS UDP RMT 0x0D          // set udp output sequence as a single LF character
```

4-14.1. Serial Port Commands

SET SERIAL BAUDRATE
Function: RS232 serial port baud rate setting
Syntax: SET SERIAL BAUDRATE *rate*
Argument(s): *rate* int32
Remarks: This function sets the baud rate for the RS232 serial port. The *rate* parameter may be any value from 9600 to 230400, with the standard rates being 9600, 19200, 38400, 57600, 115200, and 230400. This command will take effect immediately, and does not require a reboot. Note that this setting may be overridden by hardware DIP switches located on the controller assy.
Return Value: none
Example(s):

```
SET SERIAL BAUDRATE 115200
```

SET SERIAL FLOW
Function: RS232 flow control
Syntax: SET SERIAL FLOW *enable*
Argument(s): *enable* byte, 0-1 (or OFF/ON)
Remarks: This function can be used to selectively enable or disable the serial port hardware RTS/CTS flow control signals. A value of 0 (or OFF) disables flow control, while any other value (or ON) enables RTS/CTS handshaking. Flow control can be used to prevent loss of input data while the system is busy executing commands.
Return Value: none
Example(s): SET SERIAL FLOW 1 // enable RTS/CTS flow control

4-14.2. Network Commands

SET IPADDR
Function: Sets the network IP address/mode
Syntax: SET IPADDR [*ipaddr* | DHCP | AUTOIP]
Argument(s): *ipaddr* static IP address, in the form DDD.DDD.DDD.DDD
DHCP selects DHCP address mode (default)
AUTOIP selects AUTOIP mode
Remarks: This function sets the default IP address mode, allowing the choice between static or dynamic modes. There is some interaction between the various settings, but typically selecting one mode disables the others as follows:

Static IP

Setting a static fixed IP address automatically disables DHCP and AutoIP operation.

DHCP

Setting DHCP mode will enable both the DHCP and AutoIP modes. The existing static IP address (if any) will be erased. DHCP takes preference over any other selected mode. If the system is unable to obtain an address from a DHCP server on the network, it will switch over to AutoIP mode, where it will attempt to assign a link-local address.

AutoIP

Setting AUTOIP mode will enable the AutoIP function and disable DHCP operation for networks where a DHCP server is not available. The existing static IP address (if any) will be erased.

You can also use the SET DHCP and SET AUTOIP commands to combine modes and override the default addressing mode operation selected by this command. If doing so, you should use the SET IPADDR command prior to using SET DHCP or SET AUTOIP, as it has precedence. For example, you can use the SET IPADDR *ipaddr* to set a fixed IP, followed by SET DHCP ON to enable DHCP. The system would attempt to use DHCP, and if unable to obtain an address would use the static IP address *ipaddr*. Likewise, you can use SET IPADDR DHCP followed by SET AUTOIP OFF, in which case the system would only use DHCP and would never switch over to AutoIP mode. You can use the SHOW IPADDR command to view the current address in use, as well as the status of the DHCP and AUTOIP clients.

Return Value: none
Example(s):

```
SET IPADDR 10.0.0.2          // sets static IP, disables DHCP and AUTOIP
SET IPADDR DHCP              // enables DHCP (and AutoIP)
SET IPADDR AUTOIP            // enables AutoIP (disables DHCP)
```

SET DHCP
Function: DHCP client control
Syntax: SET DHCP *enable*
Argument(s): *enable* byte, 0-1 (or OFF/ON)
Remarks: This function can be used to selectively enable or disable the DHCP client. A value of 0 (or OFF) disables DHCP, while any other value (or ON) enables DHCP.
Return Value: none
Example(s):

```
SET DHCP 1          // enable DHCP
SET DHCP ON         // enable DHCP
SET DHCP OFF        // disables DHCP
```

SET AUTOIP
Function: AutoIP client control
Syntax: SET AUTOIP *enable*
Argument(s): *enable* byte, 0-1 (or OFF/ON)
Remarks: This function can be used to selectively enable or disable the AutoIP client. A value of 0 (or OFF) disables AutoIP, while any other value (or ON) enables AutoIP.
Return Value: none
Example(s):

```
SET AUTOIP 1        // enable  AutoIP
SET AUTOIP ON       // enable  AutoIP
SET AUTOIP 0        // disables  AutoIP
```

SET NETMASK
Function: Sets the network IP address subnet mask
Syntax: SET NETMASK *ipmask*
Argument(s): *ipmask* subnet mask, in the form DDD.DDD.DDD.DDD
Remarks: This function sets the default IP subnet mask used when static IP addressing is selected. The default value is 255.255.255.0
Return Value: none
Example(s):

```
SET NETMASK 255.255.255.0
```

SET GATEWAY
Function: Sets the network Gateway IP address
Syntax: SET GATEWAY *ipaddr*
Argument(s): *ipaddr* IP address, in the form DDD.DDD.DDD.DDD
Remarks: This function sets the default gateway/router IP address. Network packets that have a destination not reachable by the current IP configuration are sent to this address. The default value is 0.0.0.0, which disables the gateway function.
Return Value: none
Example(s):

```
SET GATEWAY 10.0.0.100
```

SET TCP SERVER
Function: Sets the TCP server port number
Syntax: SET TCP SERVER *portno*
Argument(s): *portno* initial server port, 1024-65530
Remarks: This function sets the port number used to communicate with the internal TCP server(s). If multiple servers are installed then each server opens successive ports from this initial setting. For example, if 3 servers are installed, and *portno* = 10001, then server #1 opens port 10001, server #2 opens port 10002, and server #3 opens port 10003 (The system is set to use a single server by default. Consult the factory if multiple servers are required). The default port is 10001.
Return Value: none
Example(s):

```
SET TCP SERVER 1024          // sets the server to listen for connections on port 1024
```

SET TCP KEEPALIVE
Function: Sets the TCP keepalive rate
Syntax: SET TCP KEEPALIVE *tout*
Argument(s): *tout* keepalive timer value, in seconds (0, 10-7200)
Remarks: This function sets the value of the TCP keepalive timeout parameter. The TCP server uses this setting in order to keep a socket open by periodically sending keepalive packets during periods of inactivity. The value can be set for 10 to 7200 seconds (2 hours), or 0 to disable the keepalive function. The default is 30 seconds.
Return Value: none
Example(s):

```
SET TCP KEEPALIVE 60          // sets the keepalive timer to 60 seconds
```

SET TCP TIMEOUT
Function: Sets the TCP server inactivity timeout
Syntax: SET TCP TIMEOUT *tout*
Argument(s): *tout* inactivity timer value, in seconds (0 - 60)
Remarks: This function sets the value of the TCP server inactivity timeout parameter. The TCP server uses this setting in order to automatically close a connection if the client is inactive for a period of time. The value can be set for 0 to 60 seconds, with 0 (or OFF) disabling the inactivity timeout function. The default is 0 (inactivity timeout disabled).
Return Value: none
Example(s):

```
SET TCP TIMEOUT 10            // close connection if host is inactive for 10 seconds
```

SET TCP ECHO
Function: Sets the TCP server character echoing
Syntax: SET TCP ECHO *onoff*
Argument(s): *onoff* byte, 0-1 (or OFF/ON)
Remarks: This function controls the setting of character echoing for the TCP server(s). With echo on, the server echos each received character back to the sender on a character by character basis, while with the setting off no such echoing occurs. This is useful for testing connectivity, but can result in a large number of packets transactions and degrade performance. By default, TCP echo is OFF.
Return Value: none
Example(s):

```
SET TCP ECHO OFF              // disables character echoing
SET TCP ECHO 1                // enables character echoing
```

SET UDP SERVER
Function: Sets the UDP server port number
Syntax: SET UDP SERVER *portno*
Argument(s): *portno* server port, 1024-65530
Remarks: This function sets the port number used to communicate with the internal UDP server. The default port is 20000.
Return Value: none
Example(s):

```
SET UDP SERVER 1024           // sets the server to listen for messages on port 1024
```

SET TELNET
Function: Set TELNET server controls
Syntax: SET TELNET ECHO *onoff* // local server echo
SET TELNET OPTNEG *onoff* // TELNET options negotiation
SET TELNET KEEPALIVE *onoff* // TELNET NOP keepalive
SET TELNET LOGIN *onoff* // require login
SET TELNET TIMEOUT *secs* // session inactivity timeout
Argument(s): *onoff* byte, 0-1 (or OFF/ON)
Remarks: This function controls various settings of the TELNET server operation. The current settings can be viewed using the SHOW NET TELNET command. Note that there are two TELNET server implementations available, a full and a reduced-functionality version, and not all parameters are supported by the reduced version. The default settings are: echo on, optneg on, NOP keepalive on, logon off, and an inactivity timeout of 300 seconds.
Return Value: none

SET NETSTAT
Function: network status message events control
Syntax: SET NETSTAT *enable*
Argument(s): *enable* byte, 0-1 (or OFF/ON)
Remarks: This function can be used to control the display of network status messages on the serial console, including link up/down and port connect/disconnect messages. A value of 0 (or OFF) disables messages, while any other value (or ON) enables them. The default setting is on.
Return Value: none
Example(s):
SET NETSTAT 1 // enables logging of network events to the console

Example NETSTAT messages:

**netstat: link up
**netstat: link down
**netstat: port 23: socket 2 connected to 10.0.0.101
**netstat: port 23: disconnected

IPCONFIG?
Function: return network settings
Syntax: IPCONFIG?
Argument(s): none
Remarks: This query command returns the current network settings, including the IP address, subnet mask, gateway address, DHCP enable, AutoIP enable, TCP server port, and the UDP server port.
Return Value: IP addr, netmask, gateway, DHCP enable, AutoIP enable, TCP port, UDP port
Example(s):
IPCONFIG?
10.100.103.80, 255.255.255.0, 0.0.0.0, 0, 0, 10001, 20000

4-15. SHOW Commands: The SHOW commands provide a method to view a variety of system settings and information. They are primarily meant for CLI usage such as the console mode or telnet, as the contents are system dependent and may change depending on the current mode, settings, and installed implementation features.

SHOW EOS
Function: Shows a summary of the EOS PMT and RMT message terminator settings
Example(s):
>show eos
serial pmt: 0x0A0D
serial rmt: 0x0A0D
usb pmt: 0x0A0D
usb rmt: 0x0A0D
tcp pmt: 0x0A0D
tcp rmt: 0x000D
udp pmt: 0x0A0D
udp rmt: 0x000D

SHOW SET
Function: Shows a summary of various configuration SET command settings
Example(s):
>show set
serial baudrate: 115200

IP address = 10.0.0.2
subnet mask = 255.255.255.0
gateway = 0.0.0.0
addr_conf = 0x03 (DHCP)
tcp keepalive timeout = 30
server port = 10001
telnet_conf = 0x0F
telnet timeout = 300
hostname = core18

SHOW USB
Function: Shows USB settings
Example(s):
>show usb

protocol: CDC
VID: 0x25EA
PID: 0x206C
ver: 0.10
status: 0xC1

SHOW NET
Function: Shows general network settings
Example(s):
>show net

link status: up
phy speed: 100
phy duplex: full

MAC address: 00:04:A3:12:2D:45
IP address : 10.0.0.2
subnet mask: 255.255.255.0
gateway : 0.0.0.0

DHCP: enabled

SHOW IPADDR
Function: Shows IP address mode and status
Example(s):
// Example #1: SET IPADDR DHCP status: no DHCP server found, using AUTOIP
>show ipaddr

IP address : 169.254.127.57
subnet mask: 255.255.0.0
default IP : 0.0.0.0

DHCP client: enabled
server IP: none detected
addr stat: not bound

AutoIP client: enabled
addr stat: bound

// Example #2: SET IPADDR 10.0.0.2, SET DHCP ON status: no DHCP server found, using static IP
>show ipaddr

IP address : 10.0.0.2
subnet mask: 255.255.255.0
default IP : 10.0.0.2

DHCP client: enabled
server IP: none detected
addr stat: not bound

AutoIP client: disabled

// Example #2: SET IPADDR 10.0.0.2, SET DHCP ON status:, DHCP server detected, using DHCP
>show ipaddr

IP address : 192.168.0.2
subnet mask: 255.255.255.0
default IP : 10.0.0.2

DHCP client: enabled
server IP: 192.168.0.1
addr stat: bound

AutoIP client: disabled

SHOW NET TCP
Function: Shows TCP server settings/status
Example(s):
>show net tcp

tcp keepalive: 30
tcp echo: on
tcp server timeout: 0
tcp server port: 10001

number of servers: 1
connections per server: 1
port 10001: no connection
port 10001: socket 1 connected to 10.100.103.113

SHOW NET UDP
Function: Shows UDP server settings/status
Example(s):
>show net udp

udp server port: 20000

SHOW NET TELNET
Function: Shows TELNET server settings/status
Example(s):
>show net telnet

timeout: 300
flags
  echo: 1
  keepalive: 1
  neg options: 1
  login: 0

max connections: 1
port 23: no connection

4-16. Misc Commands

CONSOLE
Function: Console mode enable
Syntax: CONSOLE *mode*
Argument(s): *mode* byte 0, 1, 2, 3 or OFF, ON, ENABLE, DISABLE
Remarks: This function enables/disables the serial port console mode command-line interface and optionally updates the nvm setting. Setting *mode*=0 turns console off, *mode*=1 turns console on, *mode*=2 enables the console, and *mode*=3 disables the console. Modes 0 and 1 (OFF and ON) update the nvm setting, while modes 2 and 3 (ENABLE and DISABLE) do not. Note: This setting may be overridden by a hardware DIP switch located on the controller assembly.
Return Value: none
Example(s):

```
CONSOLE ON          // turns on the console and updates nvm setting
CONSOLE ENABLE      // turns on console for this session only
CONSOLE 0           // turns off the console and updates nvm setting
CONSOLE DISABLE     // turns off console for this session only
```

CONSOLE?
Function: Console mode query
Syntax: CONSOLE?
Argument(s): none
Remarks: This function returns the serial console mode nvm and DIP switch settings
Return Value: *nvm, dipsw* integer, integer
Example(s):

```
CONSOLE?
1, 0                // console nvm flag = 1, DIP switch = 0
```

USB CONSOLE
Function: USB Console mode enable
Syntax: USB CONSOLE *mode*
Argument(s): *mode* byte 0, 1, 2, 3 or OFF, ON, ENABLE, DISABLE
Remarks: This function enables/disables the USB CDC console mode command-line interface and optionally updates the nvm setting. Setting *mode*=0 turns console off, *mode*=1 turns console on, *mode*=2 enables the console, and *mode*=3 disables the console. Modes 0 and 1 (OFF and ON) update the nvm setting, while modes 2 and 3 (ENABLE and DISABLE) do not.
Return Value: none
Example(s):

```
USB CONSOLE ON          // turns on the USB console and updates nvm setting
USB CONSOLE ENABLE      // turns on USB console for this session only
USB CONSOLE 0           // turns off the USB console and updates nvm setting
USB CONSOLE DISABLE     // turns off USB console for this session only
```

DELAY
Function: Delays execution (Pause)
Syntax: DELAY *msecs*
Argument(s): *msecs* word, 0-65535 in msecs
Remarks: This command pauses execution for the specified time in msecs.
Return Value: none
Example(s):

```
ATTN 1 0; DELAY 100; ATTN 1 10      // waits 100 msecs between attn commands
```

REBOOT
Function: system reset
Syntax: REBOOT
Argument(s): none
Remarks: This command performs a system reboot, similar to a poweron reset.
Return Value: none
Example(s):
REBOOT

RUN
Function: run an auxiliary program function
Syntax: RUN *cmd*
Argument(s): *cmd* command function
LOADER
USB LOADER
Remarks: This command runs an external function, such as the Flash Bootloader for downloading program updates
Return Value: none
Example(s):
RUN LOADER // invokes the flash bootloader for update
RUN USB LOADER // invokes the usb bootloader for updating USB functionality

LCD
Function: Adjust LCD display (if installed)
Syntax: LCD CONTRAST n
LCD BKLIGHT n
Argument(s): *n* byte, 0-255
Remarks: This function can be used to adjust the viewing parameters of the LCD display. Default values are CONTRAST 128 and BKLIGHT 255.
Return Value: none
Example(s):
LCD CONTRAST 128
LCD BKLIGHT 0 // turns off backlight

TEMP?
Function: reads internal temperature sensor
Syntax: TEMP? [*sensor*]
Argument(s): *sensor* byte optional sensor number, 0-2. default=0 (internal)
Remarks: This function returns the current temperature, in degrees C. Resolution is 0.5 degrees
Return Value: degC
Example(s):
TEMP?
30.0

TIME?
Function: reads execution time
Syntax: TIME?
Argument(s): none
Remarks: This function returns the current execution time from the start of the command message, in msecs.
Return Value: msecs integer32
Example(s):
CMDSTATS 0; TIME?; DELAY 10; TIME?
1;11

TIMESTAMP
TIMESTAMP?
Function: set/reads system timestamp timer
Syntax: TIMESTAMP [0]
TIMESTAMP?
Argument(s): see text
Remarks: These commands can be used to time multiple events spanning a long time period. TIMESTAMP records the current system tick counter, and TIMESTAMP? returns the number of ticks since the last TIMESTAMP command. The tick counter is a 1ms 32-bit value, and will rollover when the max count is reached. The command TIMESTAMP 0 allows you to reset any current timestamp, after which TIMESTAMP? will return the current system tick counter directly.
Return Value: ticks integer32
Example(s):

```
>TIMESTAMP?                      // read current system ticks (sys uptime)
1193502
>TIMESTAMP; TIMESTAMP?           // set a new timestamp and read the time
1
>TIMESTAMP 0                     // reset timestamp
>TIMESTAMP?                      // returns current system tick counter
1214141
```

REPEAT
Function: Enables command repetition/looping
Syntax: REPEAT *count*
Argument(s): *count* word, 1-65535
Remarks: This function causes the remainder of the current command message to be repeated count number of times. Any commands included prior to REPEAT are executed a single time.
Return Value: none
Example(s):

```
ATTN 1 0; REPEAT 50; INCR 1; DELAY 100      // repeats INCR and DELAY 50 times
```

SYSTEST
Function: performs a low-level system test
Syntax: SYSTEST *select*
Argument(s): *select* test select, varies by platform

WDT	// test the watchdog timer function
STACK	// test the stack over/underflow reset function
ALL	// test aux hardware board
FP	// front-panel (if installed)

Remarks: This command performs a low-level test on the selected hardware. NOTE: These tests should be used with great caution, and should typically NOT be performed with any attached RF devices/hardware, as they may exercise them in an invalid fashion. They are typically used in serial console mode only. Consult the factory prior to performing any of these tests.
Return Value: various status messages

CMDSTATS
CMDSTATS?
Function: command statistics
Syntax: CMDSTATS 0
CMDSTATS?
Remarks: These commands are used to control and query the internal command statistic counters, which include a count of the total number of commands processed and a count of errors. CMDSTATS 0 resets the counters, while CMDSTATS? returns the current counter settings since last reset. Note: these commands are not counted in the statistics. **Return Values:** total_cmds, number_of_errors
Example(s):

```
>CMDSTATS 0                    // reset counters
>*CLS                          // send a valid command
>CMDSTATS?                     // query counters
1, 0                           // command count: 1, error count: 0
>FRED                          // send an invalid command
error 101: invalid cmd
>CMDSTATS?                     // query counters
2, 1                           // command count: 2, error count: 1
```

4-17. Misc Network Commands

PING
Function: Sends ICMP ECHO packets
Syntax: PING *ipaddr*
Argument(s): *ipaddr* destination IP address, in the form DDD.DDD.DDD.DDD
Remarks: This function performs a ping of the specified network address. It is primarily for console usage
Example(s):

```
>ping 10.100.103.113
pinging 10.100.103.113...
reply time: 1 ms
reply time: 1 ms
reply time: 1 ms
reply time: 1 ms
```

TCP SERVER
Function: TCP server control
Syntax: TCP SERVER *cmd*
INIT // reinitializes all server ports
CLOSE // closes all server ports
OPEN // reopens server ports
Remarks: This function allows control of the TCP server. It is primarily for testing/troubleshooting purposes

4-13. UPDATING DEVICE FIRMWARE VIA SERIAL PORT

Requirements:

- a terminal emulator that supports XMODEM-CRC transfer (such as Windows Hyperterm, or the open-source TeraTerm)
- hex programming file (19311459xxx.HEX), where ‘xxx’ represents the specific file/revision.

Note: any CRC and version numbers displayed will change depending on the actual data file used.

The internal bootloader allows for downloading and programming of the unit via an RS232 serial port. It features a simple command-line interface, and supports downloading of standard Intel HEX format files that contain application Program Code memory (flash), Configuration memory settings, and Data memory (EEPROM) initialization sections. The bootloader supports transfers using the XMODEM-CRC protocol, so it can be used with any standard terminal emulator software such as Windows Hyperterminal. By default, transfers are done with a serial rate/format of 115200 N81.

To get into bootloader mode, from the main application command prompt, use the command RUN LOADER.

API Weinschel Model xxxxx Vx.xx
firmware: 19311459xxx

>run loader

When the bootloader executes in 'loader' mode, you should see a sign-on message similar to

>run loader

Weinschel core18K loader V0.01
firmware: 19311459301x

osc stat: 00000011

:

The loader uses the ':' colon character as it's prompt. Because of program space constraints of the bootloader, there are minimal messages and responses to most commands. A '#' character indicates an error was detected. Other status messages are typically issued in the form

stat: 11000000

The commands to the bootloader are short two or three ASCII character sequences, and can be either upper or lower-case. A simple line-editor allows the use of the BACKSPACE key, and commands are executed when a carriage-return (CR) is detected. Commands include:

AC	allow Configuration word update
AD	allow Data Memory update
BL	blank check
DL	download and program HEX file
DLW	download and program with 10 second wait
ED	erase Data Memory
EP	erase Program Memory
RS	reset (reboot)
RN	run application in Program Memory
XS	checksum Program Memory (display only)
XSU	checksum Program Memory and update ID locations

Since there is limited memory available to the microcontroller, there is only space for a single application. This requires the program memory to be erased prior to downloading a new file, and the programming must be performed as the HEX file is being transferred. Because of this, any failure in the download/programming process will leave the PIC without an application, and the process will have to be repeated. To simplify the operation, the DL download command performs all the steps typically required to reprogram an application. It performs a blank check, erases the ID checksum location (to signify that the program memory is invalid), erases the program code memory if necessary, downloads (and programs) the code, and, if successful, computes and updates the checksum IDLOC. A typical display of the DL operation looks like the following (note that the actual crc displayed will vary with the file downloaded)

:dl
blank check...*not blank*
erasing program...
begin download...CCCCCCCC <begin XMODEM-CRC file transfer on PC>
stat: 11000000
crc: B6E3

Here you see the resulting status warnings indicating that Config and Data memory updates were disabled, and that programming was successful. Any other status result should produce a '***fail***' message in place of the crc display.

After issuing the DL command, you will see the '**begin download...**' message and a series of '**C**' characters as the loader attempts to initiate an XMODEM download. At this point, send the HEX file via XMODEM-CRC using your terminal emulator program. Many terminal emulators will automatically detect the XMODEM transfer type, but if it gives a choice, select XMODEM-CRC . For example, using Windows Hyperterm you would use the '**Transfer | Send File...**' menu, enter the HEX file in the '**Filename**' box (19311459xxx.HEX), select '**Xmodem**' as the Protocol, and click 'Send'. Hyperterm will detect that the loader is using CRC error checking (as opposed to checksum). If you need to abort the download operation, send a few CTRL-C characters, and the loader should terminate the download.

Use XS command to check the status of the Program memory, which will perform a CRC checksum on the program memory contents as compared to the IDLOC CRC contents, which should match.

:xs
crc: B6E3
idloc: B6E3

To exit the bootloader and run the main application, use the RN command (or cycle power). When the unit reboots, you should see the new version signon message.

:rn
running app...
API Weinschel Model xxxxx Vx.xx
firmware: 19311459xxx

4-19. USB driver file (AW83xxCDC.inf):

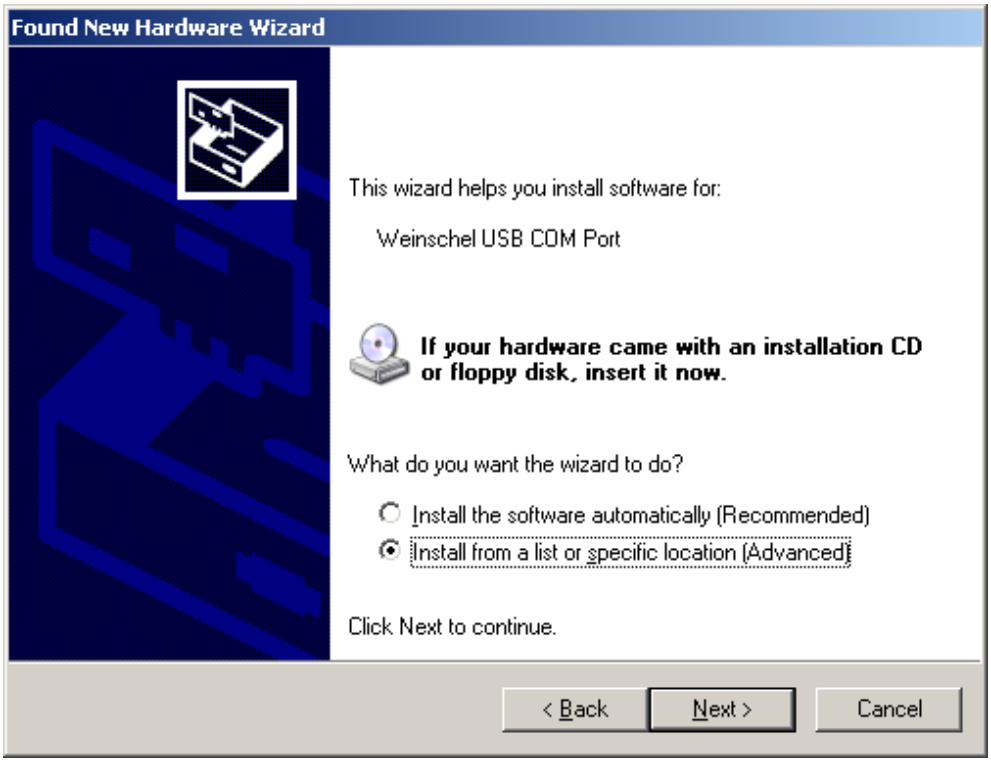
```
; Windows USB CDC ACM Setup File
; Copyright (c) 2000 Microsoft Corporation
; Copyright (C) 2007 Microchip Technology Inc.

[Version]
Signature="$Windows NT$"
Class=Ports
ClassGuid={4D36E978-E325-11CE-BFC1-08002BE10318}
Provider=%MFGNAME%
LayoutFile=layout.inf
;CatalogFile=%MFGFILENAME%.cat
DriverVer=03/11/2010,5.1.2600.2

[Manufacturer]
%MFGNAME%=DeviceList, NTamd64

[DestinationDirs]
DefaultDestDir=12


;------------------------------------------------------------------------------
;  Windows 2000/XP/Server2003/Vista/Server2008/7 - 32bit Sections
;------------------------------------------------------------------------------

[DriverInstall.nt]
include=mdmcpq.inf
CopyFiles=DriverCopyFiles.nt
AddReg=DriverInstall.nt.AddReg

[DriverCopyFiles.nt]
usbser.sys,,,0x20

[DriverInstall.nt.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,%DRIVERFILENAME%.sys
HKR,,EnumPropPages32,,"MsPorts.dll,SerialPortPropPageProvider"

[DriverInstall.nt.Services]
AddService=usbser, 0x00000002, DriverService.nt

[DriverService.nt]
DisplayName=%SERVICE%
ServiceType=1
StartType=3
ErrorControl=1
ServiceBinary=%12%\%DRIVERFILENAME%.sys

;------------------------------------------------------------------------------
;  Windows XP/Server2003/Vista/Server2008/7 - 64bit Sections
;------------------------------------------------------------------------------

[DriverInstall.NTamd64]
include=mdmcpq.inf
CopyFiles=DriverCopyFiles.NTamd64
AddReg=DriverInstall.NTamd64.AddReg
```

```
[DriverCopyFiles.NTamd64]
%DRIVERFILENAME%.sys,,,0x20

[DriverInstall.NTamd64.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,%DRIVERFILENAME%.sys
HKR,,EnumPropPages32,,"MsPorts.dll,SerialPortPropPageProvider"

[DriverInstall.NTamd64.Services]
AddService=usbser, 0x00000002, DriverService.NTamd64

[DriverService.NTamd64]
DisplayName=%SERVICE%
ServiceType=1
StartType=3
ErrorControl=1
ServiceBinary=%12%\%DRIVERFILENAME%.sys


;------------------------------------------------------------------------------
;  Vendor and Product ID Definitions
;------------------------------------------------------------------------------
[SourceDisksFiles]
[SourceDisksNames]
[DeviceList]
%DESCRIPTION%=DriverInstall, USB\VID_25EA&PID_206C

[DeviceList.NTamd64]
%DESCRIPTION%=DriverInstall, USB\VID_25EA&PID_206C


;------------------------------------------------------------------------------
;  String Definitions
;------------------------------------------------------------------------------
;Modify these strings to customize your device
;------------------------------------------------------------------------------
[Strings]
MFGFILENAME="aw83xxcdc"
DRIVERFILENAME ="usbser"
MFGNAME="AeroflexWeinschel"
DESCRIPTION="Weinschel USB COM Port"
SERVICE="USB RS232 Emulation Driver"
```

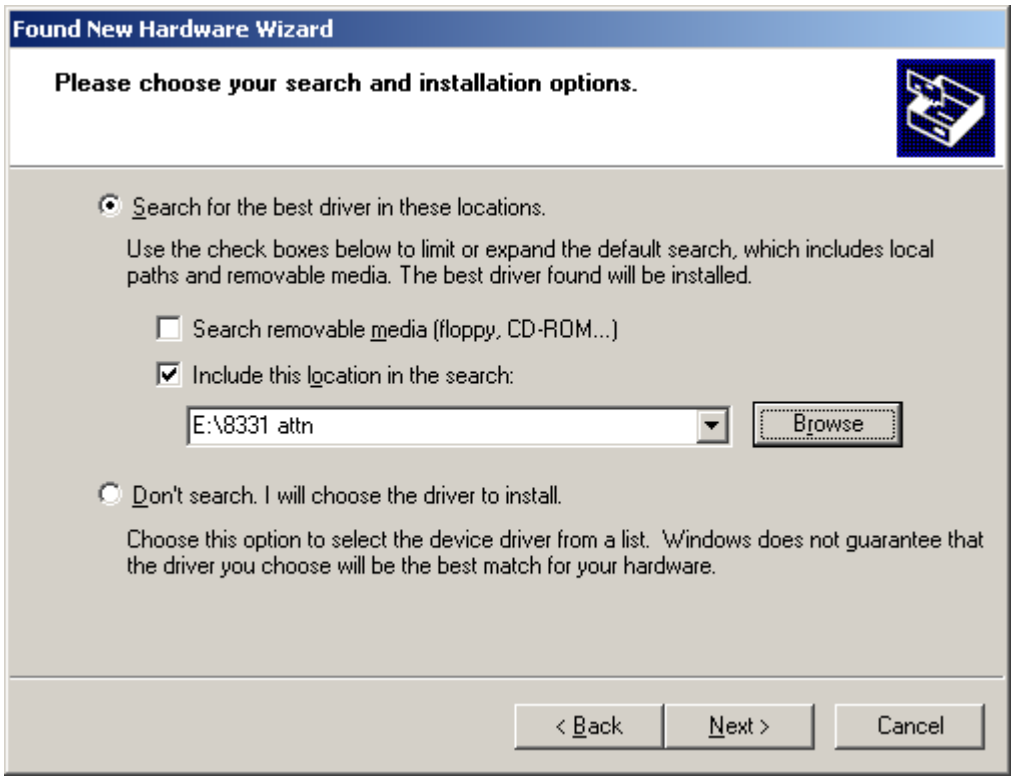
4-20. INSTALLING API WEINSCHEL USB CDC DRIVER:

Connect a USB cable from the unit to a USB port on the PC. Windows should detect the device and the New Hardware wizard should run.

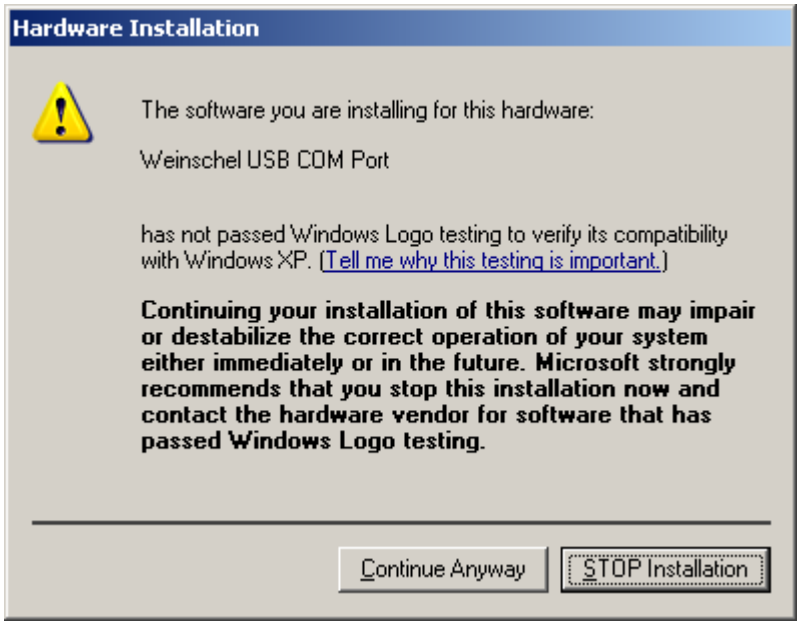
Select 'No, not this time' and click 'Next'



Select ‘Install from a specific location’ and click ‘Next’



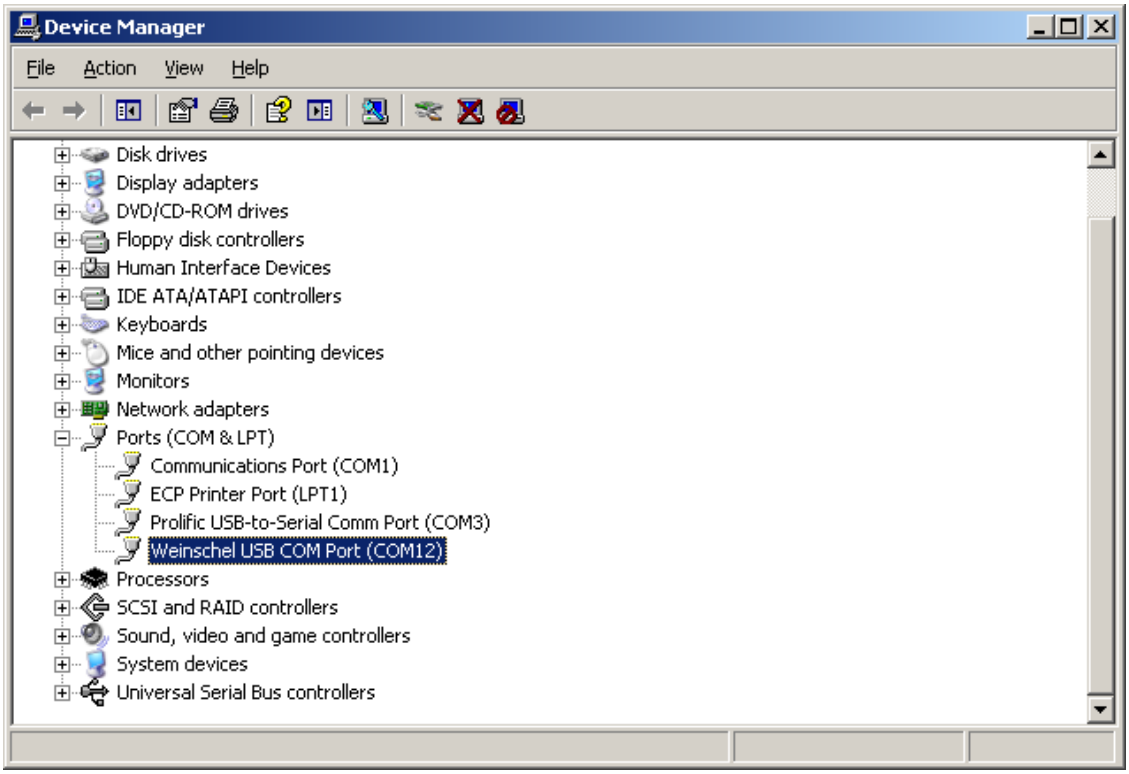
Using the ‘Browse’ button, navigate to the drive/folder containing the AW83xxCDC.inf file.
Select ‘Next’



You will get a message stating that the driver is not signed. Select ‘Continue Anyway’

The hardware wizard should complete. Select ‘Finish’, and you should get a message that your new hardware is installed any ready to be used.

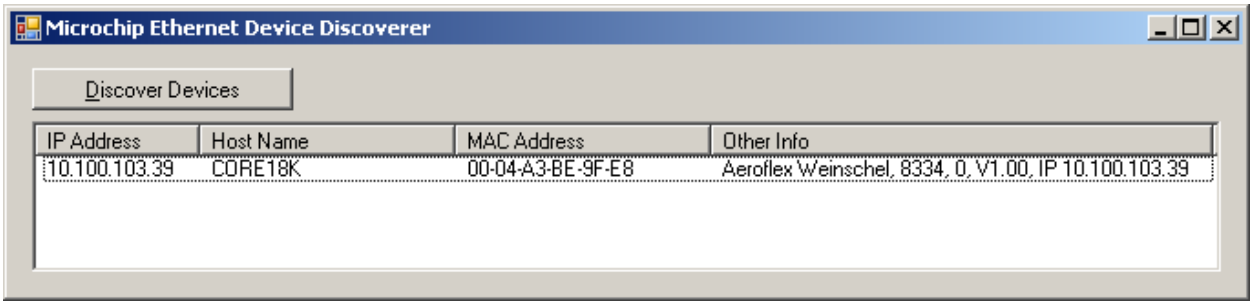
To verify that the driver is installed properly, go to Device Manager and you should see an entry under Ports (COM & LPT) for Weinschel USB COM Port, and the assigned COM port number



4-21. USING THE MICROCHIP ETHERNET DISCOVERER TOOL

The Microchip Ethernet Discoverer Tool is a PC application that aids in locating Ethernet devices on the network.

When the "Discover Devices" button is clicked, this application will transmit a broadcast UDP packet containing the message, "Discovery: Who is out there?" on the local network to port 30303. If any embedded devices with the Microchip Announce protocol enabled are connected to the network, they will respond with a UDP packet containing their host name, MAC address, and other info such as the model number and firmware version.

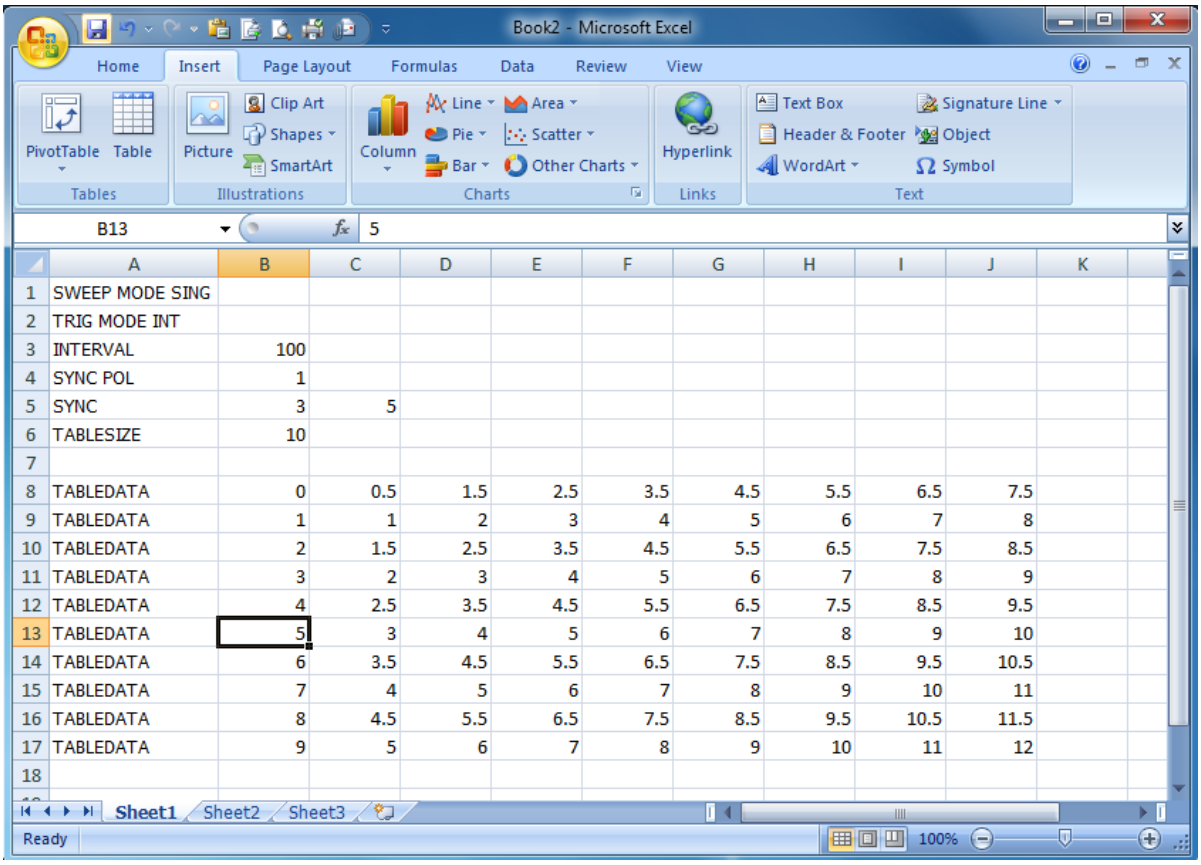


If you are unable to view the network setup with a serial or USB connection, this utility can be used to find the current IP address of any attached API Weinschel devices. You can then use TELNET to connect to the device and change the address using the SET IPADDR command.

4-22. CREATING TABLES WITH EXCEL

Example of how you can create and download a table using MS Excel:

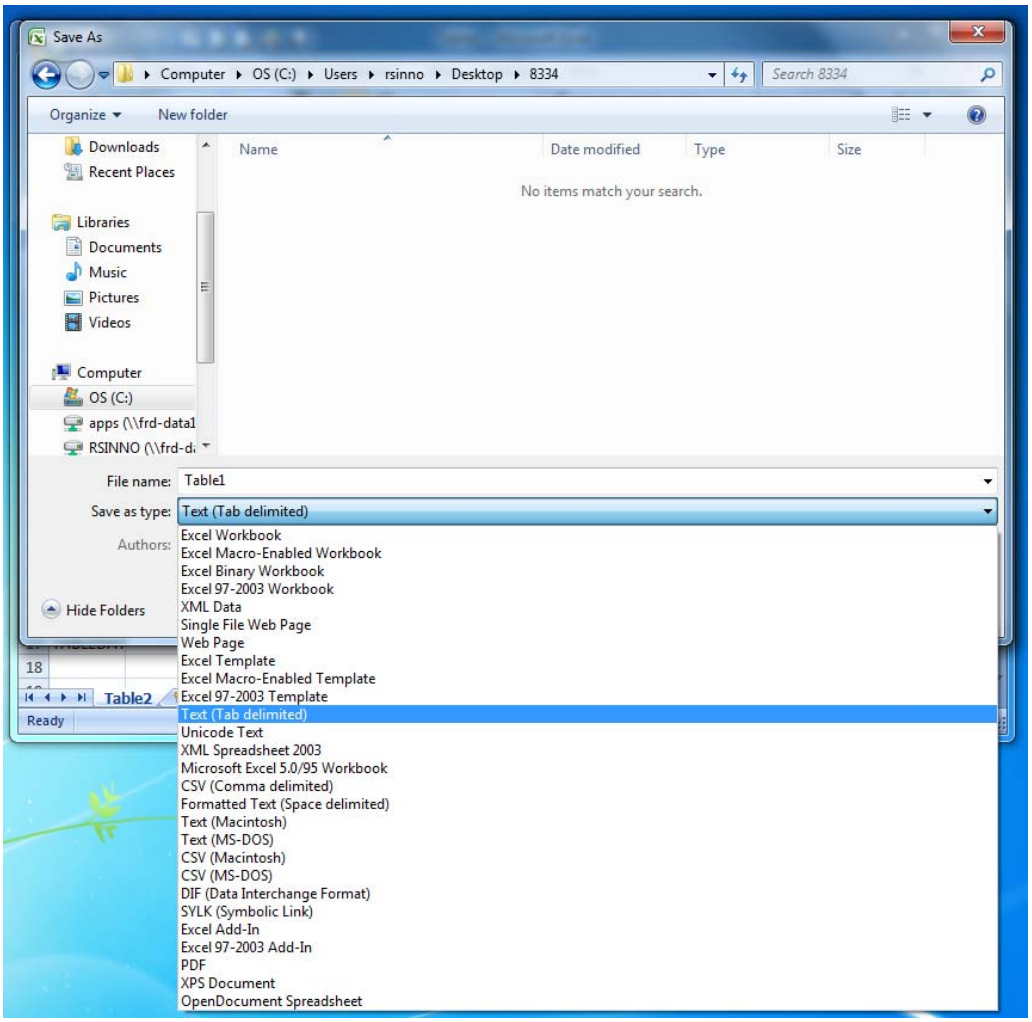
Step 1: Input data to Excel



	A	B	C	D	E	F	G	H	I	J	K
1	SWEEP MODE SING										
2	TRIG MODE INT										
3	INTERVAL	100									
4	SYNC POL	1									
5	SYNC	3	5								
6	TABLESIZE	10									
7											
8	TABLEDATA	0	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	
9	TABLEDATA	1	1	2	3	4	5	6	7	8	
10	TABLEDATA	2	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	
11	TABLEDATA	3	2	3	4	5	6	7	8	9	
12	TABLEDATA	4	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	
13	TABLEDATA	5	3	4	5	6	7	8	9	10	
14	TABLEDATA	6	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	
15	TABLEDATA	7	4	5	6	7	8	9	10	11	
16	TABLEDATA	8	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	
17	TABLEDATA	9	5	6	7	8	9	10	11	12	
18											

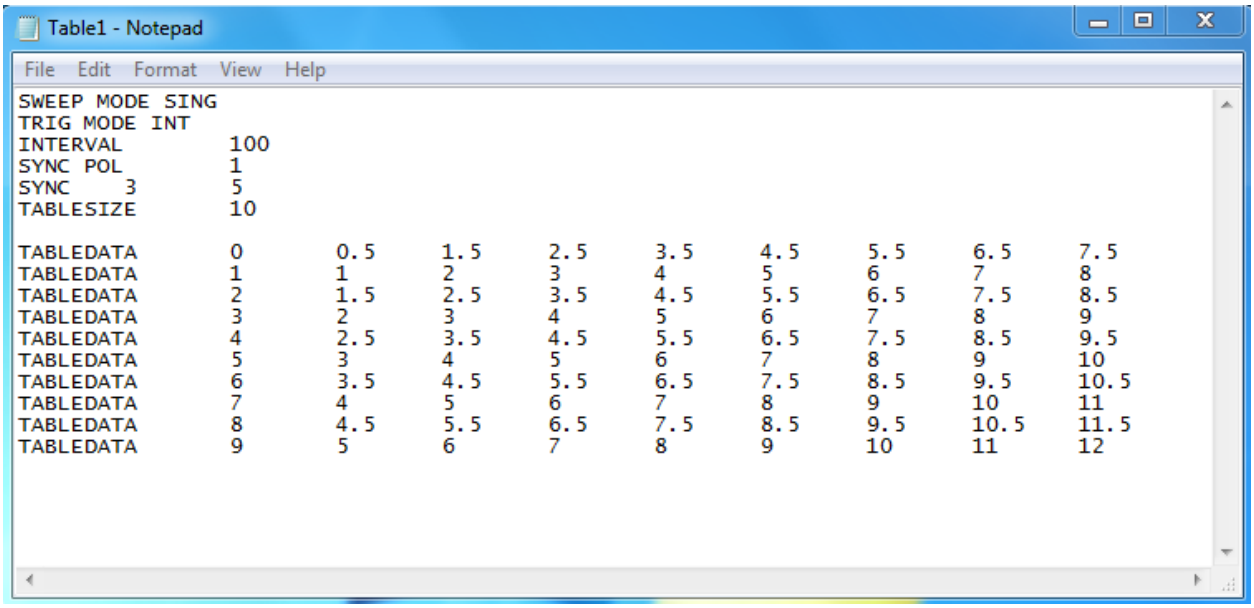
In the Excel sheet showing above, the sweep parameters are set in rows 1 to 5, the table size is set in row 6, and the table data are recorded in rows 8 to 17. The alphabetical and the numerical parts of each command are separated into multiple cells. In this case, the user has the capability of creating these numbers using Excel functions.

Step 2: Save File



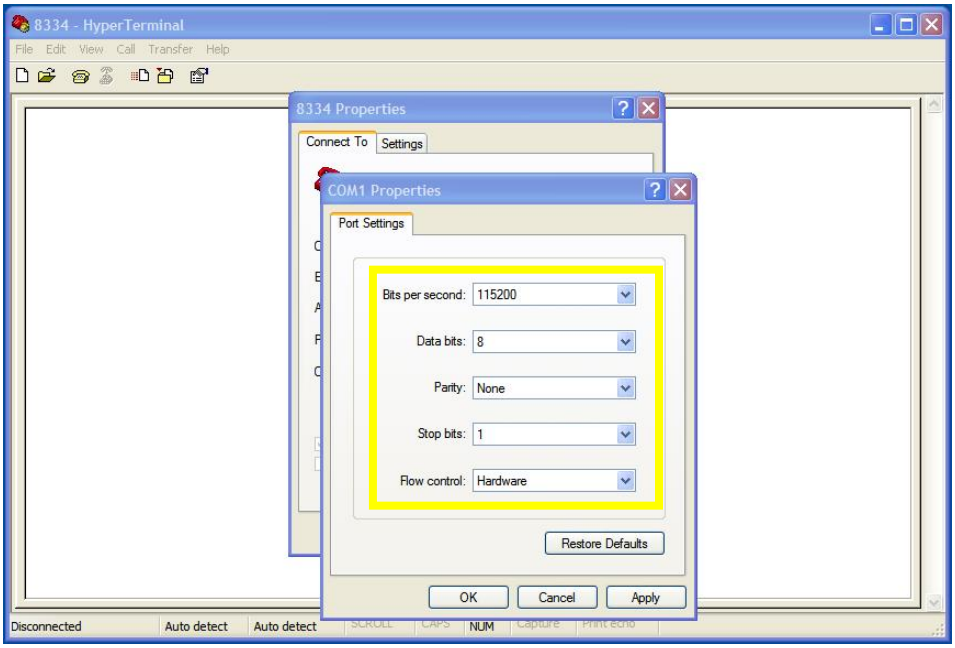
After recording all the required data, the file must be saved as a text (.txt) file and that can be done directly from Excel by choosing from the “save as type” drop down menu the text (Tab delimited) format. Another copy of this file could be saved as an Excel Workbook file (.xls or xlsx) as a reference for future editing.

Step 3: Text file verification



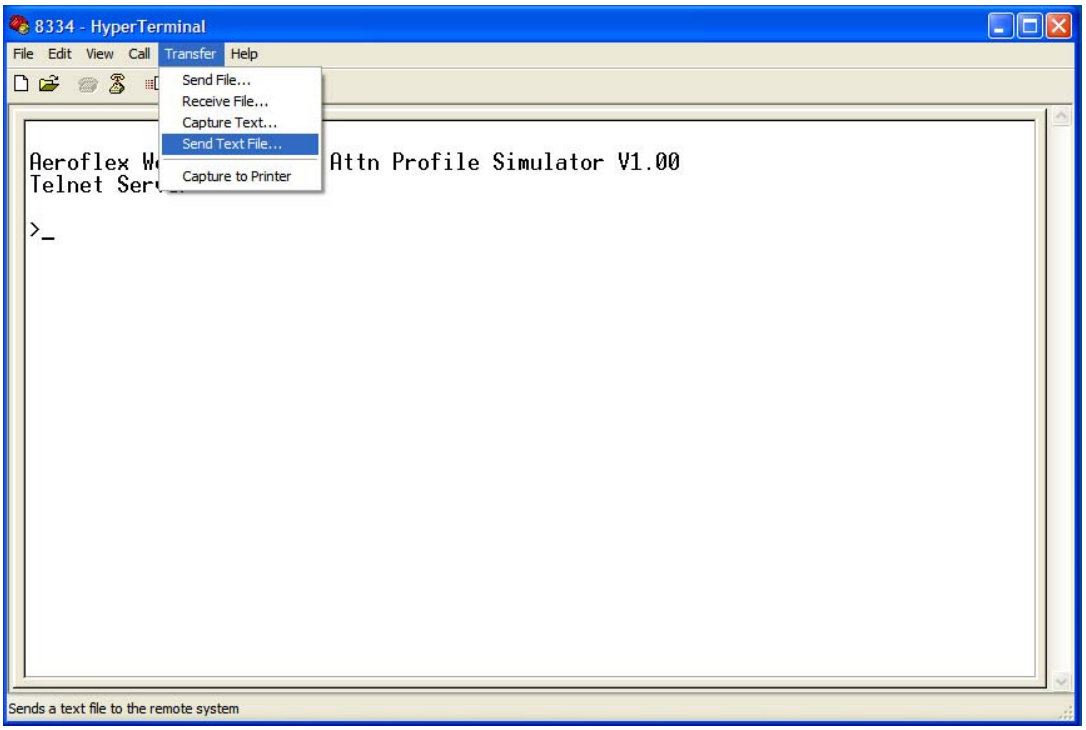
The .txt file format that was saved from the previous step is TAB delimited. The 8334 can accept a text file with space or tab delimited formats.

Step 4: Setting a Terminal Emulator Connection

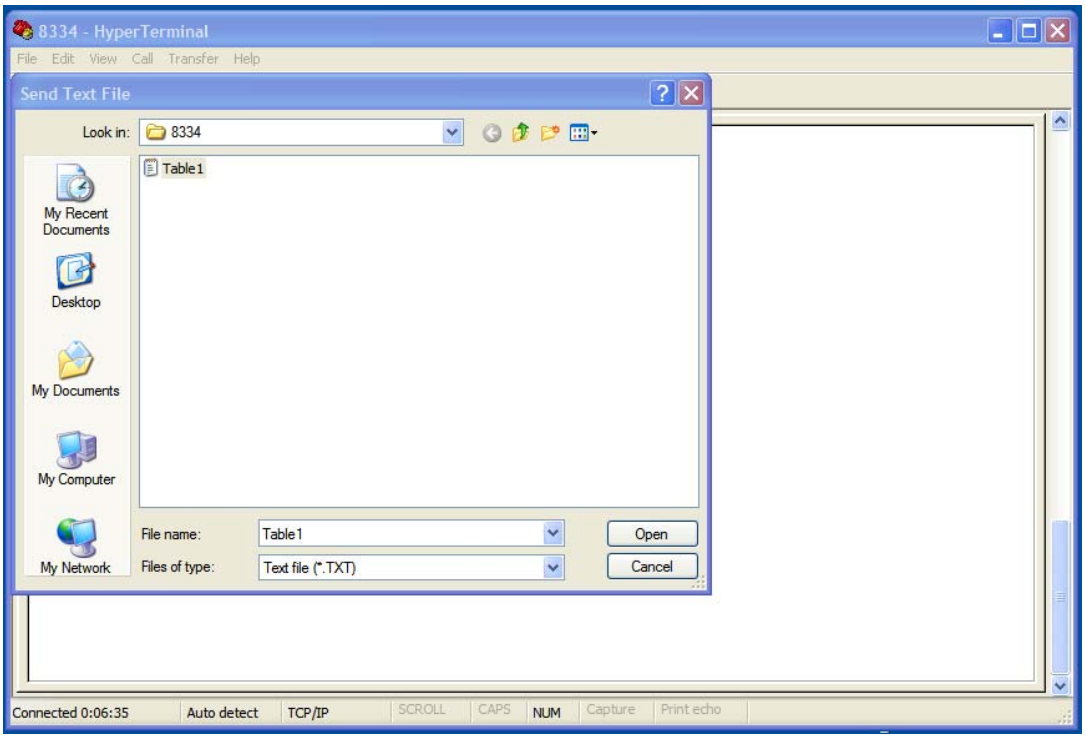


In this Example, HyperTerminal is used to create a serial connection between a PC and an 8334. The parameters for this connection must be set as shown above. Be sure to select Hardware Flow control if connecting via a serial COM port.

Step 5: Importing the Saved Text File

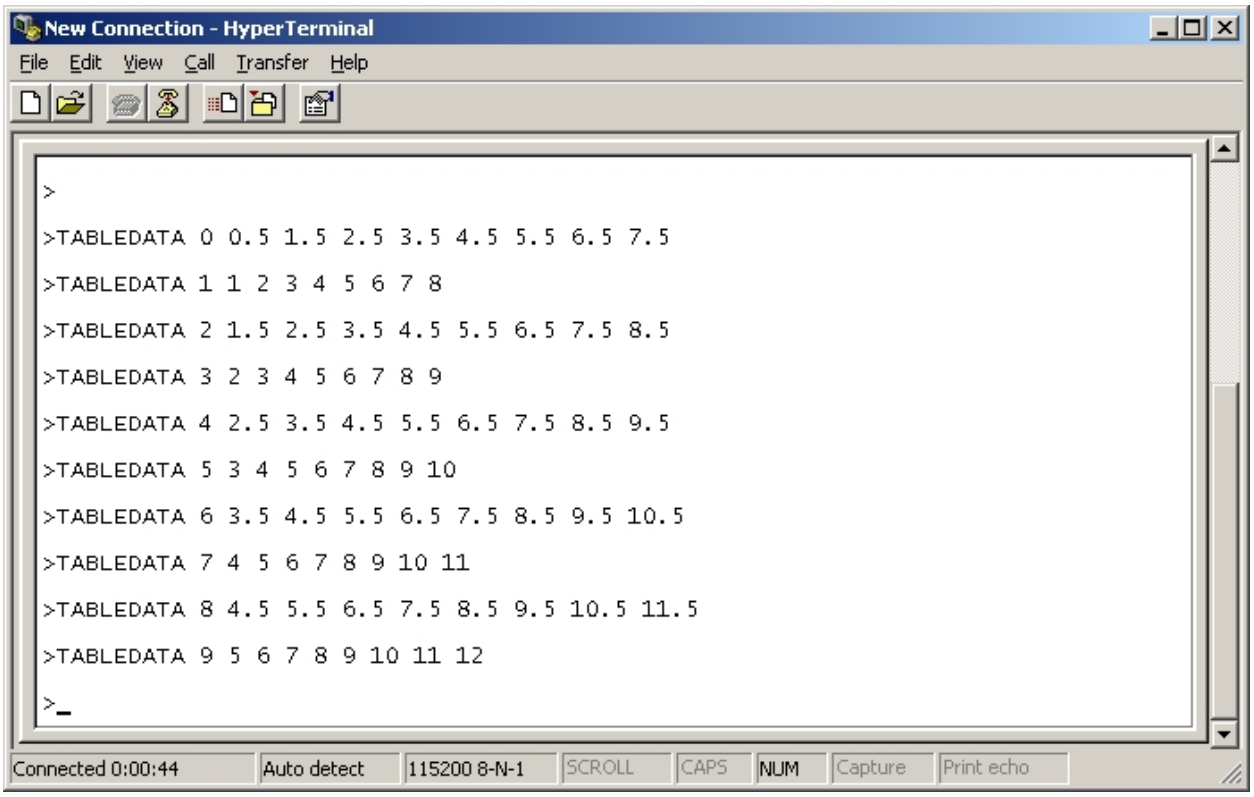


To import the saved tab delimited text file, select “Transfer” then “Send Text File” from the toolbar.



From the pop-up browsing window, locate the saved text files from step 2 (in this case it is table1.txt) then click on “open”. Import process time will vary depending on the number of tabledata points.

Step 6: Sample screen echo from the table transfer



Echo from the imported table will be displaced on the main HyperTerminal window while it is reading the text file. If desired, the user can achieve higher transfer rate by turning the console off before the first table point and turning it back on after the last point. TABLEDATA points will not be displayed on the main window, but a higher transfer rate could be achieved.

Once the table data and setup has been sent, you can save it to non-volatile memory (TABLE SAVE), or run a simulation (SWEEP START).

5. MAINTENANCE:

The following paragraphs provide general inspection and maintenance guide-lines for the Model 8334 Series.

5-1. INSPECTION: Perform a visual inspection in conjunction with the maintenance activities schedule when a malfunction is suspected, or whenever an assembly is removed or replaced.

5-2. PREVENTIVE MAINTENANCE: While the 8334 Series requires very little preventive maintenance, it should not be subjected to physical abuse, severe mechanical shock, high humidity, or operating temperatures outside the specification range. The instrument should be kept free of excessive dirt and dust, since these can interfere with connector functions and with normal heat dissipation. The following paragraphs provide the preventive maintenance that is to be performed on the Unit.

Care should be taken to prevent strain on the interconnecting cables, since damage here may not always be apparent. Occasionally check the external cables and connectors for signs of cracked insulation and/or bent or worn pins. Tests show that connectors must be clean for accuracy and stability. This requires an inspection and cleaning of each connector immediately before use. For connector cleaning instructions, refer to paragraph 4-3. When cleaning precautions are observed regularly, connectors can maintain their stability for over several thousand connection cycles. Refer to Appendix A for more information about cables and connectors.

5-3. SPECIAL CLEANING INSTRUCTIONS: The cleaning procedures for 8334 Series are divided into five general groups: microwave coaxial cable assemblies, circuit card and modules; machined surfaces and hardware, chassis cleaning, and connector cleaning.

5-3.1. MICROWAVE COAXIAL CABLE ASSEMBLIES: Appendix A (located at the end of this manual) provides all the necessary procedures for care, cleaning, and handling of microwave coaxial cable assemblies.

5-3.2 MACHINED SURFACES AND HARDWARE: To remove light dirt and dust from mechanical parts such as castings, covers and other hardware proceed as follows:

WARNING

Compressed air used for cleaning and/or drying can create airborne particles that may enter the eye. Goggles/ face-shields should be worn. DO NOT direct air stream towards self or other personnel. Pressure should be restricted to a maximum 15 psi to avoid personal injury.

CAUTION

Under no circumstances use a wire brush, steel wool, or abrasive compound. Using these items will cause extensive damage to the instrument's surface.

DO NOT use a nylon bristle brush in solvent as the bristles may dissolve and cause damage to the circuit card or component.

a. Use 5 psi of clean, moisture-free compressed air or preferably dry nitrogen to blow loose dirt and dust from surface of item.

b. Briskly brush isopropyl alcohol onto area to be cleaned with a fiber-bristle brush.

c. Remove residue with lint-free cloth and repeat step "b" as a rinse.

d. When parts are thoroughly clean, dry parts using 5 psi of clean, moisture-free compressed air or preferably dry nitrogen.

e. Clean smaller mechanical parts or hardware by dipping into a container of isopropyl alcohol. Remove dirt by brushing with fiber-bristle brush after parts have been immersed for several hours.

f. Remove parts from isopropyl alcohol and rinse by immersing into a different container of isopropyl alcohol.

g. When parts are thoroughly cleaned, dry parts using 5 psi of clean, moisture-free compressed air or preferably dry nitrogen.

5-3.3 CONNECTOR CLEANING: Where small amounts of rust, corrosion, and/or oxide deposits are present on connectors, clean externally with a soft-bristle brush, aluminum wool, or internally with an acid brush; then wash with a non-corrosive solvent. Exercise care to ensure no metal filing or residue remains inside the connector and the connector is thoroughly dry. Where rust, corrosion, and/or oxide deposits are present in large quantities, replace the connector.

5-4. LINE VOLTAGE FUSE REPLACEMENT: The following steps provide procedures to replace the line voltage Fuse Assembly. This unit accepts a T0.5A, 250 Vac fuse.

WARNING

Sufficient power levels are present at the Power Input Assembly to cause personal injury. Ensure that the instrument power cord is DISCONNECTED before attempting to change fuses.

CAUTION

DO NOT connect or apply power to this instrument until the Power Entry Module Assembly has been adjusted to the operational line voltage.

a. Disconnect the power cord from the Power Entry Module Assembly.

b. Use a small screwdriver to pry open the Fuse Drawer.

c. Slide out Fuse Drawer located in the center of the Power Entry Module Assembly.

d. Remove defective fuse and replace with the correct fuse listed in the parts list.

e. Snap the Fuse Drawer shut and re-connect ac power cord.

6. REPLACEABLE PARTS LIST & DRAWINGS:

6-1 REPLACEABLE PARTS LIST (RPL): Refer to IM-609-1 for the Model 8334 Series RPL. The IM-609-1 manual contains a parts breakdown for the different configurations of the Model 8334 Series into its assemblies and detailed parts.

6-2 ASSEMBLY AND COMPONENT LOCATION: The assembly/component location and schematic diagrams for the Model 8334 Series is located in the IM-609-1 manual by the drawing number. Drawing find numbers have also been included in the manuals RPL to help locate components or hardware.

This manual can be downloaded from the API / Weinschel website at:

http://www.weinschel.apitech.com/weinschel/pdfiles/IM-609-Model-8334.pdf

7. CONTACTING API / WEINSCHEL:

In the event of a malfunction, contact API / Weinschel. An apparent malfunction of an instrument or component may be diagnosed over the phone by first contacting the Customer Service Department at API / Weinschel. DO NOT send the instrument or component back to the factory without prior authorization. When it is necessary to return an item, state the symptoms, catalog and type number of the instrument or component, and date of original purchase. Also write the Company name and your name and phone number on a card and tape the card to the item returned. Page provides further information regarding preparation of a unit for reshipment. Contact API / Weinschel Customer Service Department as follows:

Via mail: API / Weinschel, Inc. 5305 Spectrum Drive Frederick, MD 21703-7362 U.S.A.

Via Telefax: 301-846-9116

Via Phone: Call TOLL FREE 800-638-2048
Toll call # 301-846-9222

Via Website: www.API.com/weinschel

Via e-mail: weinschel-sales@API.com

8. API / WEINSCHEL WARRANTY:

PRODUCTS - API / Weinschel warrants each product it manufactures to be free from defects in material and workmanship under normal use and service anywhere in the world. API / Weinschel's only obligation under this Warranty is to repair or replace, at its plant, any product or part thereof that is returned with transportation charges prepaid to API / Weinschel by the original purchaser within ONE YEAR from the date of shipment.

The foregoing Warranty does not apply in API / Weinschel's sole opinion to products that have been subject to improper or inadequate maintenance, unauthorized modifications, misuse, or operation outside the environmental specifications for the product.

SOFTWARE PRODUCTS- API / Weinschel software products are supplied without representation or Warranty of any kind. API / Weinschel, therefore, assume no responsibility and will not accept liability (consequential or otherwise) arising from the use of program materials, disk, or tape.

The Warranty period is controlled by the Warranty document furnished with each product and begins on the date of shipment. All Warranty returns must be authorized by API / Weinschel prior to their return.

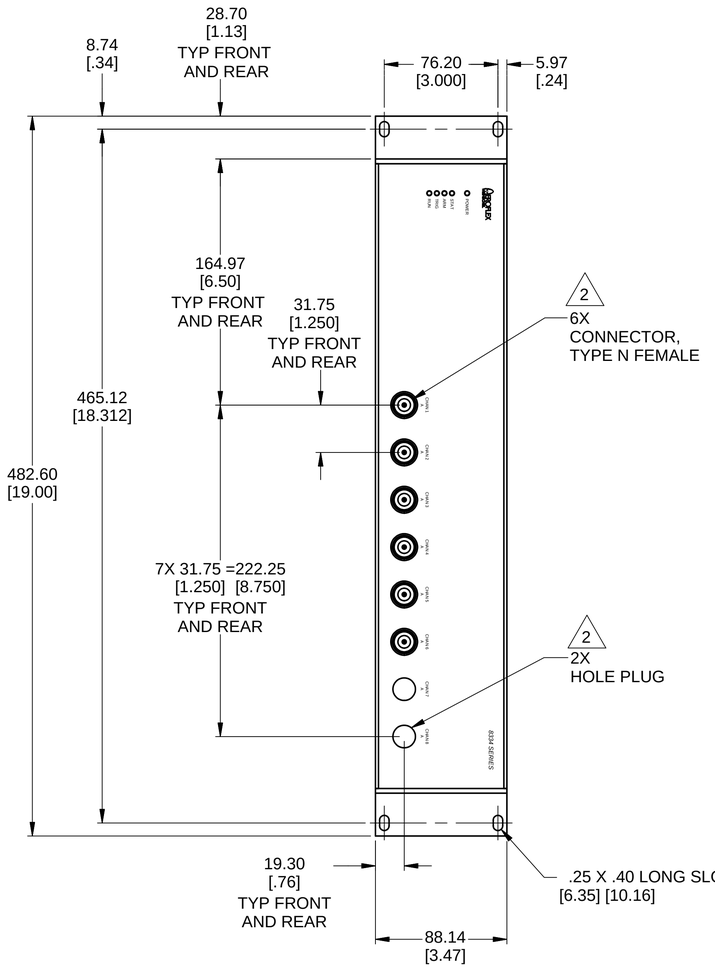
API / Weinschel's Quality System Certified to:

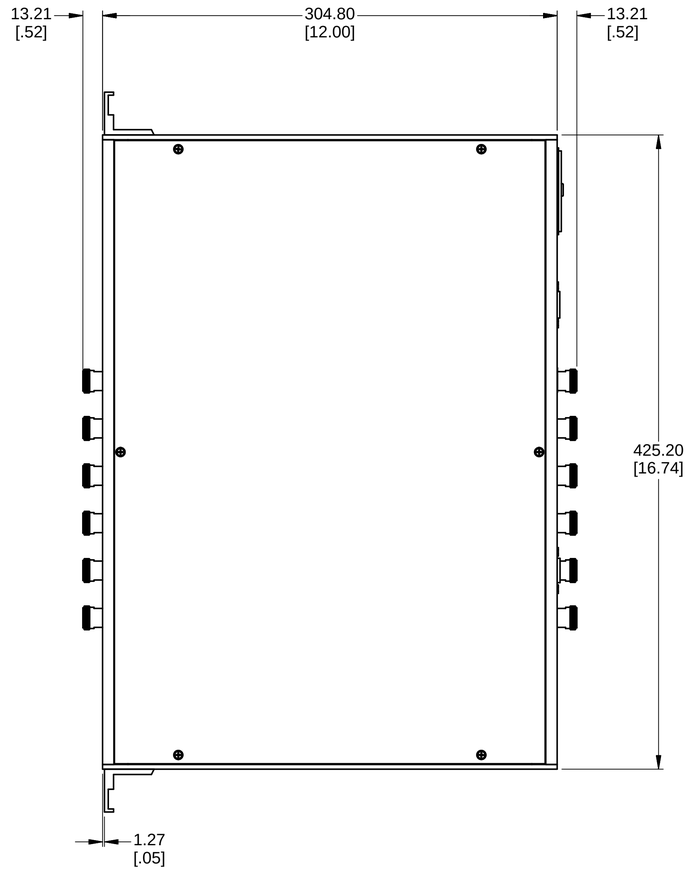
ISO 9001:2008 CERTIFICATION TM
Intertek
Certificate 94-289k

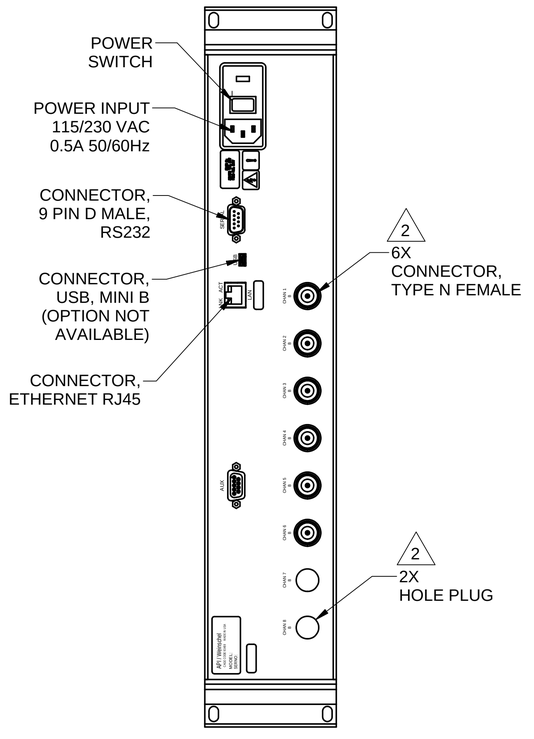
ANAB ACCREDITED

REVISION HISTORY				
ZONE	REV	DESCRIPTION	DATE	APPROVED
	A	ERN 13-037	3/19/2013	R. SINNO

NOTES:
1. ALL DIMENSIONS GIVEN IN MM [INCHES].
2. CONNECTORS AND HOLE PLUGS INSTALLED AS REQUIRED AND DETERMINED BY NUMBER OF CHANNELS IN UNIT. 6 CHANNEL UNIT SHOWN.
3. SEE A/W DWG 089-4443 FOR ELECTRICAL SPECIFICATIONS.







INTERFACE CONTROL DRAWING

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS ±1/64 DECIMALS .XX ± .01 .XXX ± .005 ANGLES ± 1/2°		CONTRACT NO.		API / Weinschel
		APPROVALS	DATE	ICD, ATTEN UNIT, TYPE N THRU MECH SPEC, MODEL 8334 SERIES
		MATERIAL	DRAWN G. THOMAS	12/11/2012
193-8300-X-X	8334-M3-XX-TN	FINISH	CHECKED A. HOPKINS	12/11/2012
NEXT ASSY	USED ON		ENGR. R. SINNO	12/11/2012
APPLICATION	DO NOT SCALE DRAWING		ISSUED	

SIZE	FSCM NO.	DWG NO	REV
D	93459	089-4442	A
SCALE 1/2		SHEET 1 OF 1	

REVISION HISTORY				
ZONE	REV	DESCRIPTION	DATE	APPROVED
	A	ERN 13-038	3/19/2013	R. SINNO

Specifications:

Frequency Range: 200 to 6000 MHz
Attenuator Model: 4205-95.5
Attenuation Range: 0 to 95.5 dB in 0.5 dB steps
Impedance: 50 ohms nominal
Insertion Loss: 8.5 dB max
VSWR: 2.0:1 max
Attenuation Update Rate: Software programmable from 100 usec./point to 1 sec./point in 100 usec. increment.
Channel to Channel Timing Skew: 1.5 usec. max. (all attenuators updated simultaneously)
Number of Attenuation Data Points: 1 to 128K (131072) per attenuator (programmable)
Non-volatile Data Table Storage: Up to 4 tables
Simulation Modes: Single sweep continuous run, & step.
Trigger modes: Internal (via software command) or External TTL signal
Status outputs (TTL): Running
Programmable Sync out
Trigger/Status TTL Connector: 9-pin D-sub
Control Interface: Ethernet (10/100), RS-232, USB
AC Power Requirements: 100-240 VAC, 47-63 Hz, 1.5 A
Operating Temperature Range: 0 to +50 degrees C

Aux TTL Connector (DB9F):

PIN No.	Description
1	SYNC OUT
2	EXT TRIG IN
3	INTERVAL UPDATE (OUT)
4	RUN OUT
5	Unused (no connection)
6	GND
7	GND
8	GND
9	GND

MODEL NUMBER CONFIGURATION MATRIX

8334-M3-XX-XN *UP TO 6 CHANNELS

- XX: 01 TO 08 CHANNELS
- XN: F = FRONT*, R = REAR*, T = THRU

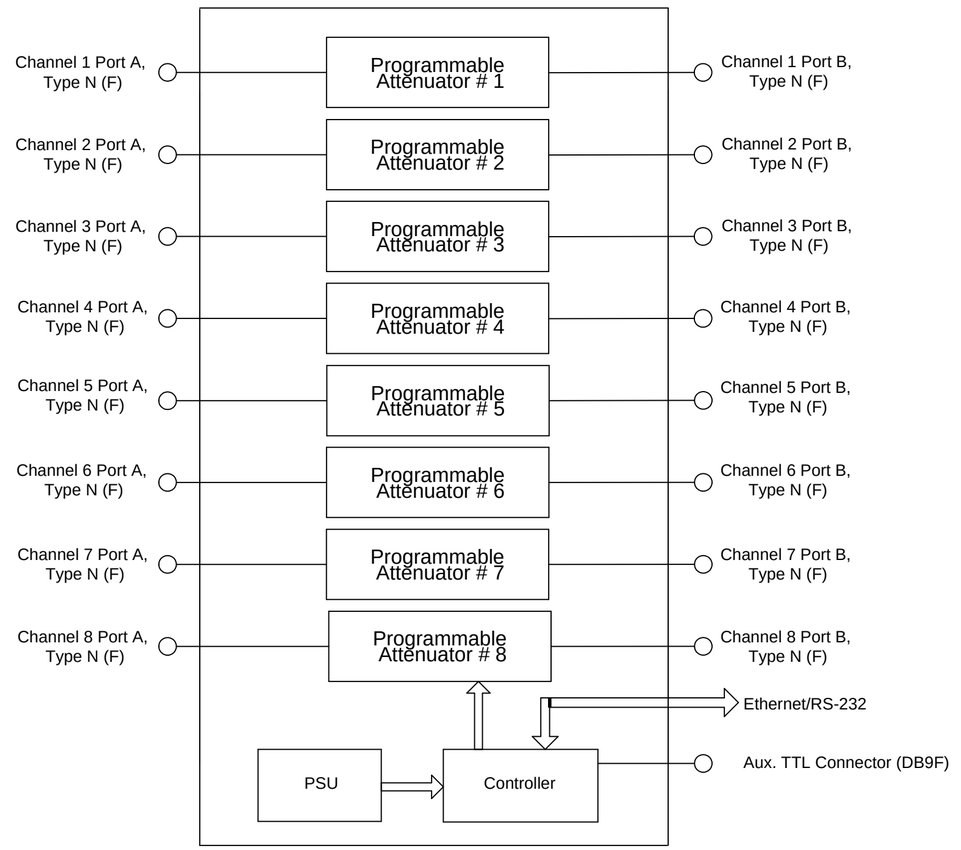


Figure 1

8 Channel Attenuator Profile Simulator, 0.2 to 6 GHz

NOTE:
ALL MATERIALS & PROCESSES ARE TO BE IN COMPLIANCE WITH THE EUROPEAN DIRECTIVE (RoHS) REF: A/W 080-638.

INTERFACE CONTROL DRAWING

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ARE:
FRACTIONS ±1/64 DECIMALS .XX ± .01 .XXX ± .005 ANGLES ± 1/2°

CONTRACT NO.

API / Weinschel

ICD, ELEC SPEC, ATTEN PROFILE SIMULATOR (8334)

	APPROVALS	DATE
DRAWN	G. THOMAS	3/19/2013
CHECKED	A. HOPKINS	3/19/2013
ENGR.	R. SINNO	3/19/2013
ISSUED		

	8334
NEXT ASSY	USED ON
APPLICATION	DO NOT SCALE DRAWING

MATERIAL
FINISH

SIZE C | FSCM NO. 93459 | DWG NO 089-4443 | REV A
SCALE 1/2 | SHEET 1 OF 1

APPENDIX A
CARE AND HANDLING OF MICROWAVE COAXIAL CABLE ASSEMBLIES

A-1 CARE AND HANDLING OF ASSEMBLIES.

To ensure accurate measurements and optimal performance of Weinschel products, the microwave coaxial cable assemblies used in system and test setups must be properly used and maintained. Proper connections, routine inspection of all cables, and cleaning of the connectors are extremely important procedures which can prolong the longevity and accuracy of equipment.

A-2 CABLE INSPECTION.

Routinely check external cables for signs of cracked insulation, dents, twists, flattening, signs of jacket abrasion, or other signs of abuse. Wrinkles in the jacket indicate that the minimum bend radius has been exceeded. Most often, this occurs near the marker tubes and connectors.

Also inspect the connector interfaces for the following:

- Bent pins (male).
- Bent or missing tines (female).
- Worn or chipped plating.
- Damaged or displaced dielectric inserts.
- Thread damage.
- Folded or mushroomed outer interface rims.
- Mushroomed pin shoulders (male) or tine ends (female).
- Score lines on pins and outer interface rims visible to the unaided eye.
- Recessed or protruding pins.

It is advisable to clean the connectors prior to inspection to make subtle damage more apparent. If any of the above is noted, replace the assembly before its further use results in equipment damage. Also inspect the mating connectors for similar damage.

Inspect the connector interface for signs of debris. Debris may be in the form of:

- Plating chips or other metal particles.
- Dust or dirt.
- Oily films.
- Other miscellaneous foreign particles.

If signs of debris are present, clean the connector interface as directed in Paragraph A-6.

A-3 MAKING INITIAL CONNECTIONS.

Exercise caution when mating cables. Poor connections lead to poor system performance. They can also damage not only the cable assembly, but more significantly, front or rear panel connectors on the equipment itself which may be more difficult to repair.

A-3.1 ALIGNING CONNECTORS. Align the center lines of two connectors before actual mating. Male retaining nuts contain a small amount of necessary play which may make it possible to mate the threads without the pins being properly aligned. Pin misalignment can damage pins and dielectric inserts.

A-3.2 MATING CONNECTORS. Gently mate the connectors by hand, taking care not to force the coupling nut at the slightest resistance. It is often possible to feel whether or not the pins are mated. If the coupling nut is difficult to turn, either the pins are not mated, the coupling nut is cross-threaded, or one of the connectors has been damaged by excess torque.

Never hold a male connector coupling nut stationary while screwing a female connector into it. This rotation can erode the plating and damage both the outer interface rim as well as the pin. If the pins become locked, serious damage can result to both the equipment and the cable assembly.

A-4 ENSURING PROPER CONNECTOR TORQUE.

A-4.1 OVERTORQUING. Once connectors have been properly mated, apply only the proper amount of torque. Overtorquing damages both connectors involved. Also, a connector which has been damaged by overtorquing, in turn, damages every connector to which it is subsequently mated. It usually leads to poor system performance as well. Overtorque can cause:

- Bent pins.
- Recessed or protruding pins.
- Recessed or protruding dielectrics.
- Chipped plating.
- Damaged coupling threads.
- Coupling nut retaining ring damage.
- Mushroomed outer interface shells.
- Mushroomed pin shoulders.

A-4.2 HEX-NUT TYPES. To mate a connector of the hex-nut type, always use a torque wrench set to the correct torque value. Tighten the connector slowly until the wrench snaps. Tightening too quickly can cause the wrench to exceed its set limit. Do not snap the wrench more than once as this also causes overtorque.

A-4.3 KNURLED NUTS. Tighten connectors with knurled nuts by hand. If this does not provide sufficient tightness use a hex-nut connector and torque wrench instead. Never use pliers to tighten a connector. Table A-1 recommends torque specifications for the various types of connectors.

Table A-1. Recommended Torque Values

Connector	Recommended Torque
GPC-7 (7mm) w/hex nut	14 in/lbs ± 1 in/lbs
Type N w/hex nut	14 in/lbs ± 1 in/lbs
SMA, 2.92mm, 3.5mm 2.4mm, WPM, WPM-3 WPM-4	7.5 in/lbs ± 0.5 in/lbs
Type N & TNC (knurled)	Hand-tight
BNC (knurled)	Hand-tight

A-5 PROPER CABLE HANDLING.

Never exceed the minimum bend radius specified for a cable. Guard against tight bends at the end of connector strain relief tubing, or at the ends of marker tubing where they may be less noticeable. Although cable bend may seem slight, the actual radius of the bend at the point of angular departure may be far smaller than the acceptable radius.

Never pinch, crush or drop objects on cable assemblies. Also, do not drag a cable over sharp edges as this will pinch it and cause it to exceed the minimum bend radius.

Never use a cable assembly to pull a piece of equipment. Cables and connectors are not designed to support or move equipment.

A-5.1 SECURING CABLES. Use toothed, rubber-lined "P-clamps" to hold cables in place. If it is necessary to use tie-wraps, use the widest possible wrap and the lowest setting on the gun to ensure the minimum pressure on the cable.

A-5.2 STORING CABLES. When storing cables, minimize cable "set" by coiling them in large diameters (1 or 2 feet). Unroll the cable properly when it is ready to be used; do not pull the loops out hastily. Similarly, re-roll them when storing them away again.

A-6 CLEANING CONNECTOR INTERFACES.

Use the following guidelines in cleaning connector interfaces:

a. Do not use chlorinated solvents including common tap water. These solvents are extremely penetrating and sometimes ruin otherwise good devices and assemblies.

b. Moisten a cotton swab with isopropyl alcohol. Roll the swab on a paper towel to remove excess.

c. Use the moistened cotton swab to wipe away debris. Do not try to dissolve the debris by overwetting the swab.

d. Repeat the cleaning process using additional swabs as necessary. If metallic particles are embedded in the dielectric, use an eyeglass and a sharp pick in an attempt to dislodge them. Swab again.

e. When satisfied that the interfaces are clean, blow them dry with dry compressed air, or preferably dry nitrogen (pressurized spray cans work well). Do not use breath.

f. Clean the mating connectors. These may be the source of the debris.

Appendix B - Component Datasheets

Model 4205 Series, Digital Attenuator

AEROFLEX WEINSCHEL

Model 4205
Digital Attenuator
TTL & USB Control, SMA Connectors

0.2 to 6 GHz

NEW

RoHS

Features

- **Ideal for Automated Test Equipment (ATE), WiMAX, 3G Fading Simulators, Engineering/Production Test Lab environments**
- **Excellent Repeatability & Performance**
- **Custom Configurations Available Upon Request**
- **Ruggedized Construction**

Description

API / Weinschel's new line of MMIC Digital Attenuator operates over the 0.2 to 6 GHz frequency range and is in a variety of attenuation ranges. These unit can be controled using either standand TTL or USB interfaces.

Specifications

NOMINAL IMPEDANCE: 50 Ω

FREQUENCY RANGE: 0.2 to 6.0 GHz

CELL CONFIGURATIONS:

Model Number	Attenuation Range/Steps (dB)	Attenuation Increments (dB)
4205-31.5	0 -31.5 / 0.5	0.5, 1, 2, 4, 8, 16
4205-63.5	0-63.5 / 0.5	0.5, 1, 2, 4, 8, 16, 32
4205-95.5	0-95.5 / 0.5	0.5, 1, 2, 4, 8, 16, 32, 32

ATTENUATION ACCURACY (dB):

dB range	4205-31.5	4205-63.5	4205-95.5
1 to 7.5 dB	± 0.5	TBD	± 0.5
8 to 11.5 dB	± 1.0		± 1.0
12 to 31.5	± 1.25		± 1.25 or 4%
32 to 85 dB	- - -		± 1.25 or 4%
86 to 95 dB	- - -	- - -	± 5 %

MAXIMUM INSERTION LOSS (dB):

Frequency (GHz)	4205-31.5	4205-63.5	4205-95.5
0.2 - 3.0	3.0	TBD	
3.0 - 6.0	4.0		8.0

MAXIMUM SWR:

Frequency (GHz)	4205-31.5	4205-63.5	4205-95.5
0.2 - 5.0	1.50	TBD	
5.0 - 6.0	1.70		2.00

POWER RATING: +23 dBm maximum

SWITCHING SPEED: 1 µSec maximum

CONTROL LOGIC: TTL or USB

OPERATING VOLTAGE: +5 V @ 50 mA

TEMPERATURE RANGE: 0°C to + 70°C

TEST DATA: Test data can be provided at additional cost.

CONNECTORS: SMA female connector - mates nondestructively with other SMA connector per MIL-C-39012, 3.5mm and other 2.92mm connector.

CONTROL CONNECTOR: The TTL control connector is an AMP-Latch 10 pin ribbon cable connector mates with AMP P/N 746285-1 (supplied with each unit). The USB is a 5-pin female series B mini socket and mates with most standard USB 5-pin male series B mini plug connectors.

WEIGHT: 83 g (2.92 oz)

Units are supplied with both a built-in TTL and USB 2 interfaces. The unit internal intelligents senses which control configuration is connected to the unit.

5305 Spectrum Drive, Frederick, MD 21703-7362 • TEL: 301-846-9222, 800-638-2048 • Fax: 301-846-9116 web: www.API.com/weinschel • email: weinschel-sales@API.com

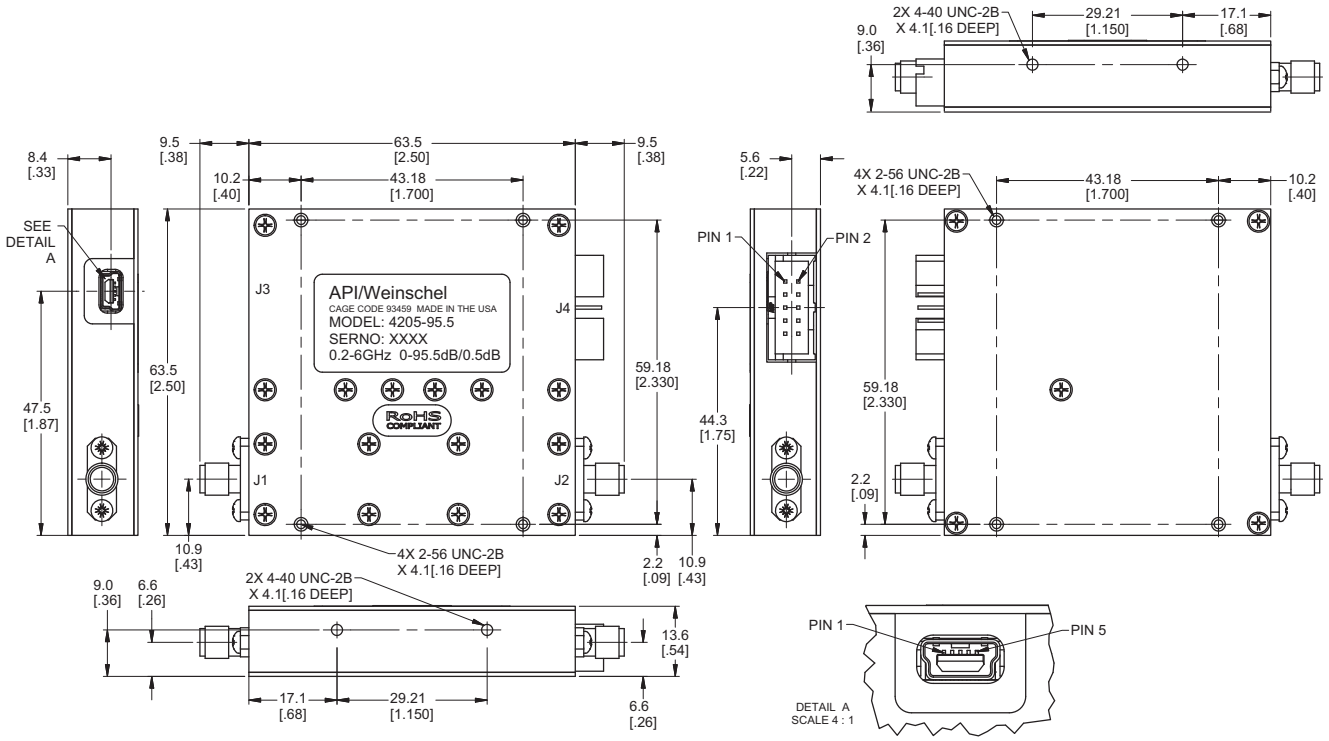
Revision Date: 3/6/2013

CONTROL CONFIGURATION:

USING TTL CONTROL: Units are supplied with a built-in TTL interface. Each unit is supplied with a mating 10 pin connector (Amp 746285-1). Refer to Physical Dimensions for mating connector pin/wiring details. Two wires are specified for supply voltage and ground. The remaining wires will accept TTL control signals to activate or de-activate a particular attenuation cell. A TTL high will energize a cell to the high attenuation state, whereas a TTL low will maintain a cell in its zero attenuation state.

USING USB CONFIGURATION: To use USB interface, your computer must have the capability built-in USB or USB expansion card installed. The computer's operating system must support USB as well. This USB port supports USB 2.0 Full-Speed, which is 12Mb Mbps and is also compatible with 1.0 (1.5 Mbps) and 1.1 (12.5 Mbps). Refer to Physical Dimensions for mating connector pin/wiring details.

PHYSICAL DIMENSIONS:



USB Control Connector J3 Pin Locations:

USB Conn PIN No. (J3)	Function
1	V BUS +5 V
2	Data-
3	Data+
4	ID (NC)
5	GND

TTL Control Connector J4 Pin Locations:

TTL Conn PIN No. (J3)	4205-31.5 dB (Cell)	4205-63.5 dB (Cell)	4205-95.5 dB (Cell)
1	0.5	0.5	0.5
2	1	1	1
3	2	2	2
4	4	4	4
5	8	8	8
6	16	16	16
7	NC	32	32
8	NC	NC	32
9	+5V	+5V	+5V
10	GND	GND	GND

NC = Not Connected.

NOTE: All dimensions are given in mm (inches) and are maximum, unless otherwise specified.

5305 Spectrum Drive, Frederick, MD 21703-7362 • TEL: 301-846-9222, 800-638-2048 • Fax: 301-846-9116 web: www.API.com/weinschel • email: weinschel-sales@API.com

Revision Date: 3/6/2013

Model 8334 Series (IM-609) Revision Record

REVISION	DATE	DESCRIPTION	APPLICABLE SERIAL NUMBERS
A	3-28-2013	ERN 13-039: Initial Issue	All Units