Preen®
AC Power Corp.

AFV-P Series
Programmable AC & DC Power Supply

Programming Manual

AC Power Corp.

CONTENTS

SECTION 1
SCPI CONFORMANCE INFORMATION

1.1 Introduction

This manual provides programming information for AFV-P series programmable AC power sources.

The AFV-P Series power sources conform to all specifications for devices as defined in IEEE 488.2, and comply with SCPI command syntax version 1999.0.

1.2 Parameter Definitions

Type	Valid Arguments
<boolean>	ON, OFF, 0, or 1, HIgh, LOw
<value>	Integer or Floating point number
<string>	String enclosed by single or double quotes

1.3 Units

The AFV-P Series accepts the following units as suffixes to numeric values:

Type of Unit	Valid Suffix
Voltage	“Volts” or “V”
Current	“Amps” or “A”
Frequency	“Hz”
Time	“ms” (milliseconds), “s” (seconds), or “min” (minutes)

1.4 Conventions

Commands enclosed by “[ ]” are optional.

For example, [**SOURce:]VOLTage[:LEVel][:IMMediate][:AMPLitude] 120.0**
can be written as, **VOLTage 120.0** or **VOLT 120.0**

1.5 Queries

The query syntax is identical to the command syntax, with a ”?” appended.

For example, to query the programmed voltage, send the string:
VOLTage? or VOLT?

A subsequent device read will return a value such as “120.00”. All queries are terminated with a carriage return and line feed (0x0D 0x0A) for those GPIB controllers that require termination characters.

SECTION 2
POWER-ON AND RESET CONDITIONS

The following sections define the factory and reset power-on conditions of the unit.

2.1 Factory Defaults

When the AFV-P unit is first powered up, the following factory defaults will be in place:

- GPIB Address: 1
- RS-232 Baud Rate: 115200
- RS-232 Data Bits: 8
- RS-232 Stop Bits: 1
- RS-232 Parity: N

2.2 Power-On Conditions

When the AFV-P unit is first powered up, the system reads in the configuration that was last saved.

2.3 Reset Conditions

When the *RST command is sent via SCPI, the following conditions apply:

- Faults are cleared
- The error queue is cleared
- The output relay is opened

2.4 Changing Voltage Ranges Via Command Interface

When the voltage range is changed using the command interface:
SOUR:VOLT:RANG 0 or VOLT:RANG AUTO

2.5 Any commands to change parameters will not save to eeprom.

It should send below command to save all parameter.
*SAV

SECTION 3
SCPI COMMANDS

3.1 MEASURE Commands

:MEASure	Measurement meter
:ALL?	Return all meters of testing.
:ALL? <value>	Return results meters of the value.
:ALL? <value1>,<value2>	Return results meters from value1 to value2.
:CURRent?	Return measured current.
:PEAK?	Return the peak current measured.
:FREQuency?	Return the frequency measured.
:PEAKCURRent?	Return the peak current measured.
:POWer?	Return presently calculated power in Watts.
:POWer[:TOTAL]?	Return presently calculated power in Watts.
:POWERFACtor[:TOTAL]?	Return the power factor measured.
:CRESTFACtor?	Return the crest factor measured.
:VA[:TOTAL]?	Return the Volt/Amps measured/calculated.
:VOLTage?	Return the RMS voltage.
:VAR?	Return the reactive power.
:TIMEr	Return the timer.

3.2 OUTPUT Commands

:OUTPut[:STATe]?	Return the state of the output relay for general mode.
:OUTPut[:STATe] <boolean>	Set the output relay to OFF or ON for general mode.
:OUTPut[:GENeral][:STATe]?	Return the state of the output relay for general mode.
:OUTPut:GENeral[:STATe] <boolean>	Set the output relay to OFF or ON for general mode.
:OUTPut[:ADVanced][:STATe]?	Return the state of the output relay for advanced mode.
:OUTPut: ADVanced [:STATe] <boolean>	Set the output relay to OFF or ON for advanced mode.
:OUTPut[:BASic][:STATe]?	Return the state of the output relay for basic mode.
:OUTPut: BASic [:STATe] <boolean>	Set the output relay to OFF or ON for basic mode.

3.3 SOURCE Commands

[:SOURce]	General mode parameters
:VOLTage	
[:LEVel]	
[:IMMediate]	
[:AMPLitude]?	Return output voltage value
[:AMPLitude] <value>	Set output voltage value
:RANGe?	Return output voltage range value
:RANGe <value>	Set output voltage range value<HIGH | AUTO>.
:FREQuency	
[:IMMediate]? <value>	Return output frequency value
[:IMMediate] <value>,<value1>	Set output frequency value
:CURRent	
[:LIMit]	
[:HI]?	Return current hi limit value
[:HI] <value>	Set current hi limit value
:POWer	
[:LIMit]	
[:HI]? <value>	Return power hi limit value
[:HI] <value>,<value1>	Set power hi limit value

3.4 BASIC Commands

:BASIc	Basic mode parameters
:MEMory	
:NAME? <value>	Return memory<1~50>'s file name
:NAME <value>,<string>	Set memory<1~50>'s, file name<"name">
:VOLTage	
[:LIMit]	
:HI?	Return voltage hi limit for basic mode
:HI <value>	Set voltage hi limit for basic mode
:LO?	Return voltage lo limit for basic mode
:LO <value>	Set voltage lo limit for basic mode
[:LEVel]	
[:IMMediate]	
[:AMPLitude]? <value>	Return memory<1~50>'s output voltage value
[:AMPLitude] <value>,<value1>	Set memory<1~50>'s, output voltage value
:RANGe? <value>	Return memory<1~50>'s output voltage range value
:RANGe <value>,<value1>	Set memory<1~50>'s, output voltage range value <HIGH | AUTO>
:FREQuency	
[:LIMit]	
:HI?	Return frequency hi limit for basic mode
:HI <value>	Set frequency hi limit for basic mode
:LO?	Return frequency lo limit for basic mode
:LO <value>	Set frequency lo limit for basic mode
[:IMMediate]? <value>	Return memory<1~50>'s output frequency value
[:IMMediate] <value>,<value1>	Set memory<1~50>'s, output frequency value
:CURRent	
[:LIMit]	
[:HI]? <value>	Return memory<1~50>'s current hi limit value
[:HI] <value>,<value1>	Set memory<1~50>'s, current hi limit value
:POWer	
[:LIMit]	
[:HI]? <value>	Return memory<1~50>'s power hi limit value
[:HI] <value>,<value1>	Set memory<1~50>'s, power hi limit value

3.5 Advanced Command

:ADVanced	Advanced mode parameters
:MEMory	
:NAME? <value>	Return memory<1~50> file name
:NAME <value>,<string>	Set memory<1~50>,file name<"name">
:STARt?	Return start memory value
:STARt <value>	Set start memory <1~50>
:END?	Return end memory value
:END <value>	Set end memory <1~50>
:LOOP?	Return memory loop value
:LOOP <value>	Set memory loop <1~999>
:SWITch?	Return memory loop switch value
:SWITch <boolean>	Set memory loop swutch <ON | OFF>
:STEP	
:STARt? <value>	Return start step of memory<1-50>
:STARt <value>,<value1>	Set memory<1~50>'s start step<1-24>
:END? <value>	Return end step of memory<1-50>
:END <value>,<value1>	Set memory<1~50>'s end step<1-24>
:LOOP? <value>	Return step loop of memory<1-50>
:LOOP <value>,<value1>	Set memory<1~50>'s step loop<1-999>
:DELTa	
:VOLTage? <value>,<value1>	Return memory<1~50>,step<1-24>'s delta voltage value
:VOLTage <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s delta voltage value
:FREQuency? <value>,<value1>	Return memory<1~50>,step<1-24>'s delta frequency value
:FREQuency <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s delta frequency value
:TIME? <value>,<value1>	Return memory<1~50>,step<1-24>'s delta time value
:TIME <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s delta time value
:UNIT? <value>,<value1>	Return memory<1~50>,step<1-24>'s delta time unit value
:UNIT <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s delta time unit value < MILLisecond | SECond | CYCLe>
:TRANsiment	
:SWITch? <value>,<value1>	Return memory<1~50>,step<1-24>'s transiment switch value
:SWITch <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s transiment switch value <ON | OFF>
:VOLTage? <value>,<value1>	Return memory<1~50>,step<1-24>'s transiment voltage value
:VOLTage <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s transiment voltage value
:SITE? <value>,<value1>	Return memory<1~50>,step<1-24>'s transiment site value
:SITE <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s transiment site value
:TIME? <value>,<value1>	Return memory<1~50>,step<1-24>'s transiment time value
:TIME <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s transiment time value
:CYCLe? <value>,<value1>	Return memory<1~50>,step<1-24>'s transiment cycle value
:CYCLe <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s transiment cycle value
:TIME? <value>,<value1>	Return memory<1~50>,step<1-24>'s test time value
:TIME <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s test time value
:UNIT? <value>,<value1>	Return memory<1~50>,step<1-24>'s test time unit value
:UNIT <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s test time unit value <SECond | MINUte | HOUR>
:DELay? <value>,<value1>	Return memory<1~50>,step<1-24>'s delay time value
:DELay <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s delay time value
:CONNect? <value>,<value1>	Return memory<1~50>,step<1-24>'s connect value
:CONNect <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s connect value <ON | OFF>

:VOLTage	
[:LEVel]	
[:IMMediate]	
[:AMPLitude]? <value>,<value1>	Return memory<1~50>,step<1-24>'s output voltage value
[:AMPLitude] <value>,<value1>, <value2>	Set memory<1~50>,step<1-24>'s, output voltage value
:RANGe? <value>	Return memory<1~50>,step<1-24>'s output voltage range value
:RANGe <value>,<value1>	Set memory<1~50>,step<1-24>'s, output voltage range value <HIGH | AUTO>
:FREQuency	
[:IMMediate]? <value>,<value1>	Return memory<1~50>,step<1-24>'s output frequency value
[:IMMediate] <value>,<value1>, <value2>	Set memory<1~50>,step<1-24>'s, output frequency value
:CURRent	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s current hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, current hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s current lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, current lo limit value
:PEAK	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s peak hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, peak hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s peak lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, peak lo limit value
:CREast	
:FACTor	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s cf hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, cf hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s cf lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, cf lo limit value
:POWer	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s power hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, power hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s power lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, power lo limit value
:FACTor	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s pf hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, pf hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s pf lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, pf lo limit value
:APParent	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s va hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, va hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s va lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, va lo limit value

:RESistive	
[:LIMit]	
[:HI]? <value>,<value1>	Return memory<1~50>,step<1-24>'s q hi limit value
[:HI] <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, q hi limit value
:LO? <value>,<value1>	Return memory<1~50>,step<1-24>'s q lo limit value
:LO <value>,<value1>,<value2>	Set memory<1~50>,step<1-24>'s, q lo limit value

3.6 STATUS Commands

:STATus	The status commands tree.
:OPERation?	SCPI commands returns „1".
:OPERation	The SCPI Operation sub-tree.
:[EVENt?]	Returns „1".
:CONDition?	Returns „1".
:ENABle <value>	Sets the value in the Operations Enable register.
:ENABle?	Returns the value in the Operations Enable register.
:PRESet	Set the value of the Operations Enable register.
:QUEStionable?	Returns „1".
:QUEStionable	
:CONDition?	Returns „1".
[:EVENt]?	Returns „1".
:ENABle <value>	Set the value of the Operations Enable register.
:ENABle?	Returns the value of the Operations Enable register.

3.7 SYSTEM Commands

:SYSTem	The system command tree.
:ERRor	
[:NEXT]?	Return error message.
:COUNt?	Return error counts.
:VERSion?	Return SCPI V1999.0.
:LOCal	The instrument return to local mode.
:KLOCK?	Returns ON, if the front panel programmable keys are locked out.
:KLOCK <boolean>	ON locks the front panel programmable keys.
:MEMory:LOCK?	Return OFF | ON
:MEMory:LOCK <boolean>	Set ON to lock memory to load.
:COMMunicate	The communications sub-tree.
:GPIB	The GPIB communications channel.
[:SELF]	SCPI sub-command.
:ADDRess <value>	Set the GPIB address to a value of 1..31.
:ADDRess?	Returns the GPIB address presently set.
:RESUlts?	Return ALL | PF | LAST
:RESUlts <value>	Set the test end screen.
:BACKlight?	Return 1-9, 9 is brightest.
:BACKlight <value>	Set the LCD backlight
:ALArm?	Return 0-9, 0 is off the buzzer.
:ALArm <value>	Set the buzzer sound.
:LANGuage?	Return EN | TW | CN | JP
:LANGuage <value>	Set the user interface to mutilanguage.
:MODE?	Return ADVanced | BASic
:MODE <value>	Set the instrument to advanced or basic mode.
:OUTPut?	Return AC | DC
:OUTPut <value>	Set the instrument to output AC or DC.
:OC:FOLD?	Return OFF | ON
:OC:FOLD <boolean>	Set ON the instrument can work on CC mode.
:POWer:UP?	Return OFF | ON | LAST
:POWer:UP <value>	Set the instrument boot on status.
:VOLTage:SENSe?	Return INTernal | EXTernal
:VOLTage:SENSe <value>	Set the instrument voltage sense locate.

:ANGLe	The system command tree.
:STARt?	Return 0-359
:STARt <value>	Set the instrument start angle.
:END?	Return 0-359
:END <value>	Set the instrument end angle.
:SYNChronous:SINGnal?	Return OFF | ON | EVENt
:SYNChronous:SINGnal <value>	Set the instrument synchronous singnal
:FAIL:STOP?	Return OFF | ON
:FAIL:STOP <boolean>	Set the instrument fail stop in testing on advanced mode.
:CONSecutive:STEP?	Return OFF | ON
:CONSecutive:STEP <boolean>	Set the instrument consecutive step in testing on advanced mode.
:PLC:REMote?	Return OFF | ON
:PLC:REMote <boolean>	Set the instrument plc remote.
:COMMand:FORMate?	Return MODBus | SCPi
:COMMand:FORMate <value>	Set the instrument bus command format.
:MODBus:ID?	Return 1 - 255
:MODBus:ID <value>	Set modbus id.
:BAUDrate?	Return 9600 | 19200 | 38400 | 57600 | 115200
:BAUDrate <value>	Set RS232 & RS485 baudrate.
:IP	
: MODE?	Return MANUal | AUTO
: MODE <value>	Set LAN mode.
:ADDRess?	Return 192.168.1.8
:ADDRess <string>	Set LAN ip address. “192.168.1.8”
:PORT?	Return 0 – 65535
:PORT <value>	Set LAN IP port number.
:SUBNet:MASK?	Return 255.255.255.0
:SUBNet:MASK <string>	Set LAN subnet mask. “255.255.255.0”
:DEFault:GATeway?	Return 192.168.1.1
:DEFault:GATeway <string>	Set LAN gatew. “192.168.1.1”
:MAC:ADDRess?	Return D8-FC-5C-AA-00-08

:GPIB:ADDRess?	Return 1 - 30
:GPIB:ADDRess <value>	Set the gpib card address.
:ANALogy?	Return OFF | ON
:ANALogy <boolean>	Set the analogy card on or off.
:SERIALNO?	D1234567890

3.6 Common Commands

The following commands are common to all SCPI instruments and declared mandatory by IEEE 488.2. In the following table, the AFV-P Series will be defined as the “device” on the GPIB bus. These commands are further supported on the RS-232 command interface.

***CLS**	Clear Status Command. Clears all status reporting data structures, including the Status Byte, Standard Event Status Register and Error Queue. Enable Masks are not cleared.
***ESE**	Standard Event Status Enable Command. Sets the Standard Event Status Enable Register, which determines which bits can be set in the Standard Event Status Register.
***ESE?**	Standard Event Status Enable Query. Returns value of Standard Event Status Enable register.
***ESR?**	Standard Event Status Register Query. Returns value of Standard Event Status Register. The ESR and the Status Byte ESR are cleared.
***IDN?**	Identification Query. Returns the device identity as an ASCII string: <manufacturer>, <model>, <serial number>, <firmware version level>. Example: Preen,AFV-P-600,D1234567890,VER: 1.04.00
***OPC**	Operation Complete Command. Causes the Operation Complete bit to be set in the Standard Events Status Register when all pending operations are complete.
***OPC?**	Operation Complete Query. Causes an ASCII „1‟ to be placed in the output queue when all pending operations are complete.
***RST**	Reset Command. Resets the device to the state defined in section 2.3. Clears all status reporting data structures, including the Status Byte, Standard Event Status Register and Error Queue. Enable Masks are not cleared.
***SRE**	Service Request Enable Command. Sets the Service Request Enable Register, which determines which bits in the Standard Event Status Register will cause a service request from the device.
***SRE?**	Service Request Enable Query. Returns contents of Service Request Enable Register. Values range from 0 to 63 or 128 to 191.
***STB?**	Read Status Byte Query. Returns the Status Byte with bit 6 representing the Master Summary Status (MSS) instead of RQS. The MSS bit acts as a summary bit for the Status Byte, and indicates whether or not the device has at least one reason to request service, based on the MAV and SESR bits. The return value is in the range of 0-255. The Status Byte is cleared after the read.
***TST?**	Self Test Query. Causes the device to execute an internal self-test and report whether or not it detected any errors. A value of „0‟ indicates the test completed without detecting any errors.
***WAI**	Wait to Continue command. Makes the device wait until all previous commands and queries are completed before executing commands following the *WAI command.
***SAV**	The command will save all memory system to eeprom, if you had send any command to change the instrument ‘s parameters. .

SECTION 4
RS-232 INTERFACE

This section describes how to connect the AFV-P for remote programming with a controller using the RS232 interface. The RS-232 interface provides a simpler 3-wire serial interface (compared to the GPIB parallel interface), while fully supporting the SCPI commands presented in Section 3.

4.1 RS-232 Interface Operation

The AFV-P uses a 3-wire connection to the controller: TxD, transmit data; RxD, receive data; GND, signal ground. When connecting the AFV-P, appropriate consideration is required to ensure that the TxD line of the AFV-P connects to the RxD line of the controller and the RxD line of the AFV-P connects to the TxD line of the controller. See Section 4.3.

The AFV-P implements the XON/XOFF (Transmit On/Transmit Off) software protocol to control the flow of data between it and the controller. Under this protocol, the receiver (either the AFV-P or the controller) controls when data is sent, and requests that the transmitter (controller or AFV-P) stop sending data, if necessary. To enable data flow, the receiver sends an XON (ASCII 19) on its TxD line to the transmitter. To request that data flow be stopped, the receiver sends an XOFF (ASCII 17) to the transmitter.

4.2 RS-232 Characteristics

The interface characteristics are listed below:

- Baud Rate: 9600
- Data Bits: 8
- Stop Bits: 1
- Parity: None
- Protocol: XON/XOFF

4.3 RS-232 Connector

The RS-232 connector is a 9-pin male Subminiature-D type; its pinout is presented below:

Pin Number	Function	Input/Output
2	RxD, Receive Data	Input
3	TxD, Transmit Data	Output
4	Connected to Pin-6	—
5	Signal Ground	—
6	Connected to Pin-4	—
7	Connected to Pin-8	—
8	Connected to Pin-7	—
1,9	Not Used	—

When connecting the AFV-P to a controller, ensure that the RxD line of the AFV-P connects to the TxD line of the controller, and that the TxD line of the AFV-P connects to the RxD line of the controller. This could be accomplished using a null-modem cable.

There are two versions of the null-modem cable: a 9-pin/9-pin and a 9-pin/25-pin. The 9-pin/9-pin cable swaps Pin-2 and Pin-3 between the two ends of the cable, so that the proper RxD and TxD connections are made, while the 9-pin/25-pin cable connects Pin-2 to Pin-2 and Pin-3 to Pin-3. Also, because the AFV-P connector has Pin-4 connected to Pin-6, and Pin-7 connected to Pin-8, the null-modem cables would connect the following control lines: DSR to DTR and CTS to RTS.

APPENDIX A
STATUS REGISTER DEFINITIONS

The AFV-P Series supports the IEEE 488.2 and SCPI 1993.0 status reporting data structures. These structures are comprised of status registers and status register enable mask pairs. These pairs are described below:

A.1 Status Byte

The Status Byte status register can be read by the *STB? command or by issuing a GPIB serial poll. Either operation clears the contents of the Status Byte. The *CLS command clears the Status Byte.

The AFV-P Series unit can be configured to request service from the GPIB controller by setting the appropriate bits in the Service Request Enable register. The SRE register has the same bit pattern as the Status Byte. The SRE register is modified using the *SRE command, and can be read with the *SRE? command. For example, if the SRE register is set to 0x10 (MAV), when the AFV-P unit has a message available, the Status Byte register will contain 0x50 (RQS and MAV) and the SRQ line of the GPIB bus will be pulled low indicating a request for service.

Bit	Hex Value	Description
0	01	Not used.
1	02	Not used.
2	04	Error/event queue message available.
3	08	Questionable Status flag. Indicates quality of current data being acquired. This bit is not used.
4	10	Message available (MAV).
5	20	Standard Event Status Register (ESR).
6	40	Request Service flag (RQS) for serial polling, or Master Summary Status (MSS) in response to ***STB?**
7	80	Operation Status flag. Indicates the current operational state of the unit. This bit is not used.

Bit 2, Error/Event Queue Information Available

This bit is set when any error/event is entered in the System Error queue. It is read using the SYSTem:ERRor? query.

Bit 4, Message Available

Indicates a message is available in the GPIB output queue. This bit is cleared after the GPIB output buffer is read.

Bit 5, Standard Event Status Register

This is a summary bit for the ESR. It is set when any of the ESR bits are set, and cleared when the ESR is read.

Bit 6, Request Service/Master Summary Status

If service requests are enabled (with the *SRE command), this bit represents the RQS and will be sent in response to a serial poll, then cleared. If RQS is not enabled, the bit represents the MSS bit and indicates the device has at least one reason to request service. Although the device sends the MSS bit in response to a status query (*STB?), it is not sent in response to a serial poll. It is not considered part of the IEEE 488.1 Status Byte.

A.2 Standard Event Status Register

The ESR can be read by the *ESR? command. Reading this register, or using the *CLS command will clear the ESR.

Bits in the ESR will be set only when the corresponding bit in the Standard Events Status Enable register is set. Use the *ESE to set bits, and the *ESE? to read this register. To configure the AFV-P Series to generate SRQ's based on the ESR, both the Standard Event Status Enable register and the Service Request Enable registers must be programmed.

Bit	Hex Value	Description
0	01	Operation Complete.
1	02	Request Control - Not used.
2	04	Query Error.
3	08	Device-Dependent Error.
4	10	Execution Error (e.g., range error).
5	20	Command Error (e.g., syntax error).
6	40	User Request - Not used.
7	80	Power On.

Operation Complete
Set whenever the last command is completed and the CW is ready to accept another command, or when query results are available.

Query Error
Set when a query is made for which no response is available.

Device-Dependent Error
Set for device-specific errors. These errors are entered in the System Error Queue and have error codes greater than 0. See Appendix B for error descriptions.

Execution Error
Set when a parameter exceeds its allowed range.

Command Error
Set for a syntax error.

Power On
Set once at power-up. The Status Byte ESR bit is not set.

A.3 Operation Status/ Questionable Status Registers

The Operation Status and Questionable Status registers always return 0 when queried. The Operation Status Enable and Questionable Status Enable registers can be programmed and queried to allow SCPI compatibility, but have no effect on the Operation Status and Questionable Status registers.

A.4 Error/ Event Queue

The CW Series maintains an Error/Event Queue as defined by SCPI. The queue holds up to 10 errors and events. It is queried using the SYSTem:ERRor? command, which reads in a first in, first out manner. The read operation removes the entry from the queue. The *CLS command clears all entries from the queue.

A.5 Serial Poll Operation

Performing a serial poll will not modify the Status Byte other than to clear the RQS (bit 6) for a CW requesting service. Queries affecting the status registers and subsequent serial polls are described below:

- *STB? clears the Status Byte
- *ESR? clears the ESR and bit 5 of the Status Register
- SYSTem:ERRor? clears bit 2 of the Status Register if the queue is empty.

This page intentionally left blank.

APPENDIX B
ERROR CODES

B.1 Error Codes Returned by SYSTem:ERRor? Query

The following error codes are defined in the SCPI 1993.0 specification, and are supported by the AFV-P Series. Error codes are in the range of [-32768, 32767]. SCPI reserves the negative error codes and 0, while error codes greater than 0 are device-specific errors.

Additionally, in multiphase system applications where multiple units are interconnected for generating polyphase outputs, the error codes will also note which unit of the system experienced the fault condition. An example is a response to the SYST:ERR? Command is: <Fault Code Number>, [Fault Condition] ; source #x where x=the unit in a multiple unit system.

B.2 SCPI Error Codes

0, No error

The error queue is empty.

-102, Syntax error

An unrecognized command or data type was encountered.

-200, Execution error

An error/event number in the range [-299, -200] indicates that an error has been detected by the instrument's execution control block. The occurrence of any error in this class shall cause the execution error bit (bit 4) in the event status register to be set.

An execution error can be the result of:

- A <program data> element out of range, such as programming 200 volts in low (156 volt) range.
- A command could not be executed due to the current condition of the device, such as attempting to change ranges while the output relay is closed. The output relay must be opened first.

-292, Referenced name does not exist

-330, Self-test failed

-345, Overcurrent Occurred; source #x

Indicates that the CW shutdown because the programmed current limit levels were met or exceeded. Source #x where x=CW Unit that saw the overcurrent condition in a multiphase system application.

-346, Overvoltage Occurred; source #x

Indicates that the CW shutdown because the programmed voltage limit levels were met or exceeded. Source #x where x=CW Unit that saw the overvoltage condition in a multiphase system application.

-347, Hardware Fault; source #x

Indicates that the CW shutdown because of hardware protection circuits internal to the CW Unit or others in a multiphase system application. The hardware fault condition may be caused from a number of internally detected faults. Source #x where x=CW Unit that saw the overvoltage condition in a multiphase system application.

-350, Queue overflow

The error queue can contain up to 10 entries. If more than 10 error/event conditions are logged before the SYSTem:ERRor? query, an overflow will occur; the last queue entry will be overwritten with error -350. When the queue overflows, the least recent error/events remain in the queue and the most recent errors/events are discarded.

APPENDIX C
SAMPLE PROGRAMS

The following examples will illustrate how to perform certain functions on the AFV-P using SCPI via the GPIB interface or the RS-232 channel.

NOTE: See the *AFV-P Series Operation Manual* for examples of how to use Slave commands.

1. Source AC from power up
   - SOUR:VOLT:RANGE LOW Specify low range 0-156V 13 Amps
   - SOUR:CURR 3 Specify 3 Amps current limit; foldback mode
   - SOUR:VOLT 120 Specify 120 VAC setpoint
   - SOUR:FREQ 60 Specify 60 Hz frequency
   - OUTP ON Source power to the output

2. Turn off Front Panel Programmability
   - SYST:KLOCK ON (1) Lock the program keys out

3. Turn on Front Panel Programmability
   - SYST:KLOCK OFF (0) Front panel program keys now functi