P620 6-CHANNEL

ISOLATED RESISTANCE

SIMULATOR

User Manual

January 13, 2022

650 Potrero Avenue San Francisco, CA 94110 www.highlandtechnology.com tel: 415 551-1700 fax: 415 551-5129

650 Potrero Ave, San Francisco, CA 94110
Phone 415-551-1700 • Fax 415-551-5129
www.highlandtechnology.com

NOTICE

HIGHLAND TECHNOLOGY, INC. PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

(Disclaimer of expressed or implied warranties in certain transactions is not allowed in some states. Therefore, the above statement may not apply to you.)

This manual may contain technical inaccuracies and/or typographical errors. Changes are periodically made to this manual, which are incorporated in later editions. Highland Technology, Inc. may make changes and improvements to the product(s) and/or programs described in this publication at any time without notice.

This product has finite failure rates associated with its hardware, firmware, design, and documentation. Do not use the product in applications where a failure or defect in the instrument may result in injury, loss of life, or property damage.

IN NO EVENT WILL HIGHLAND TECHNOLOGY, INC. BE LIABLE FOR DAMAGES, INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE SUCH PRODUCT, EVEN IF HIGHLAND TECHNOLOGY, INC. OR AN APPROVED RESELLER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

Table of Contents

1 *Introduction*

This is the user manual for the Highland model P620 six-channel resistor/RTD simulator.

The P620 is a standalone unit which simulates 6 isolated, high precision resistors or RTDs. Each resistor is independently programmable to one of five resistance ranges, or four RTD types. Within each range, resistance changes are smooth, monotonic, and glitch free.

Some features of the P620 include:

- 6 channels of fully isolated precision resistance/RTD simulation
- True 4-wire resistance simulation on all channels
- Resistance programmable from 5Ω to 5MΩ in five ranges
- Programmable for 100Ω and 1kΩ platinum RTDs
- RTD mode accepts temperatures directly in Celsius
- Monotonic resistance changes allow for RTD simulation without transient errors
- Channels may be used in series or parallel for additional voltage and current capability
- Separate TEST port allows access to channels for in-system calibration check
- 4 versatile digital input/outputs
- Built-in self-test (BIST)
- Ethernet and USB interfaces
- Access via serial ASCII commands or web interface

2 Specifications

FUNCTION	6-channel isolated resistance simulator
RESISTANCE SIMULATION RANGES	Standard resistance ranges: R5 5Ω to 500Ω R50 50Ω to 5kΩ R500 500Ω to 50kΩ R5K 5kΩ to 500kΩ R50K 50kΩ to 5MΩ Platinum RTD types: R385 -125°C to 700°C 100Ω Pt 385 K385 -125°C to 700°C 1kΩ Pt 385 R392 -125°C to 650°C 100Ω Pt 392 K392 -125°C to 650°C 1kΩ Pt 392
OUTPUT RESOLUTION	Standard resistance ranges: Variable from 9-15 bits with higher resolution at the lower end of the range (see section 6.1 for details) RTD ranges: Better than 0.05°C
INPUT OPERATING RANGE	±10 V differential, ±20 mA
OUTPUT PROTECTION	±30 V differential ±750 V common-mode ESD to 5 kV, human body model
DIGITAL INPUT/OUTPUT	4 DIO channels, TTL/CMOS compatible Outputs can sink up to 32 volts at 250 mA Direct LED drive
CONNECTORS	D25 female for 6 channels D9 male for test D9 female for DIO 1 securable power barrel connector
PACKAGING	7.0" (L) x 8.5" (W) x 2.25" (H) Aluminum enclosure
INDICATORS	LEDs indicate Ethernet/Serial access, CPU activity, error conditions Additional user programmable LED

COMMUNICATIONS	10/100 Ethernet, USB
OPERATING TEMPERATURE	0 to 70°C
CALIBRATION INTERVAL	One year
POWER	24 volts DC, 500mA max model J24, 24 VDC, 30 watt external adapter furnished 2.1 x 5.5 mm barrel connector, center pin positive
CONFORMANCE	DIN 43760

Resistance Output Accuracy

OUTPUT	TYPICAL	LIMIT 15-35°C[1]	LIMIT 0-70°C
100Ω[2], 1kΩ[3] RTD, -125°C to 250°C[4]	±0.05°C	±0.25°C	±0.7°C
Resistance 1x to 20x of base resistance	R5-R5K[5]: ±0.1%	±0.4%	±1.25%
	R50K[6]: ±0.25%	±1.0%	±4%

For a more in-depth discussion of accuracy, see section 6.

---

[1] P620 ambient temperature

[2] 1mA minimum test current

[3] 100μA minimum test current

[4] Simulated RTD Temperature

[5] 10mV minimum test voltage

[6] 100mV minimum test voltage

3 *Overview*

The P620 is a programmable resistor simulator. It can be used to simulate a resistor or resistive sensor, such as an RTD. It provides 6 independent, isolated simulation channels. The equivalent circuit of each channel is:

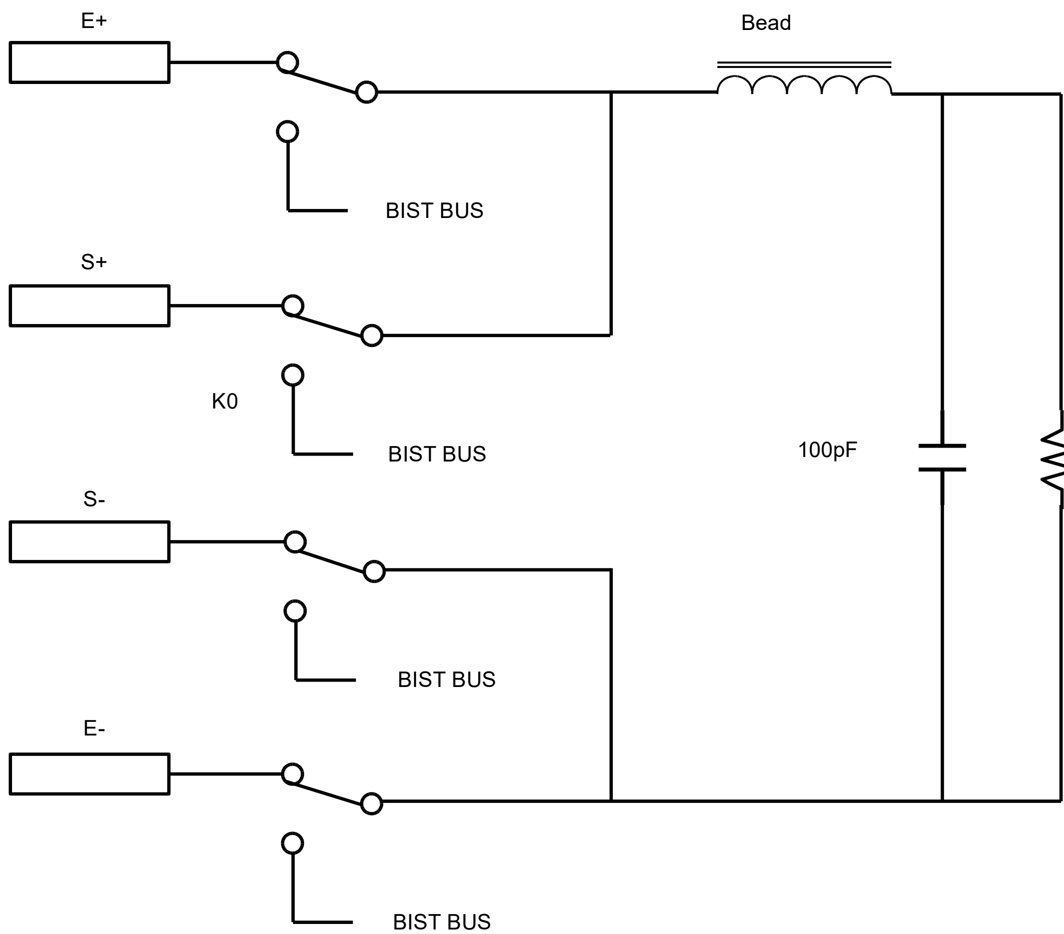

Figure 1. P620 Equivalent Circuit

Each channel has a DC power supply, a digital interface, resistance simulation circuitry, signal conditioning, and input protection circuits.

Resistances are electronically simulated and are accurate for DC, AC, or pulsed excitation. Unlike modules using relay-switched series or parallel combinations of discrete resistors, this entirely solid-state topology means that changes of resistance are monotonic, with no glitches during transition.

On-board relays can connect any input channel to the dedicated D9 calibration connector for accurate 4-wire verification of channel performance, or to the optional built-in self-test circuitry for automatic testing.

The BIST feature allows for quick verification of channel function by measuring the channel resistance using an internal ADC. When activated, each channel in turn is disconnected from the front panel and connected to the BIST ADC. Each range is then verified and finally the channel is returned to normal function. BIST also continuously checks the onboard power supplies for proper function.

The P620 can be connected in 4-wire mode or, if accuracy is not as great a concern, in 3-wire or 2-wire mode.

4 *Front and Rear Panel*

Figure 2. Front Panel

Figure 3. Rear Panel

4.1 *LEDs*

There are 4 front-panel LED indicators.

The green PWR LED will blink once every two seconds to indicate that the unit is powered and that the internal firmware is operating normally.

The blue COMM LED will flash whenever the P620 receives a command over either the USB or Ethernet ports.

The red ERR LED will blink to indicate a module error. The blinking pattern indicates a category of error:

One blink	Channel programming error or channel overload
Two blinks	Calibration table error
Three blinks	Power supplies error

Calibration table and power supply errors indicate fatal hardware problems.

The orange USR LED may be programmed with the `USer` serial command. This LED is useful for distinguishing between multiple P-series instruments in the field.

4.2 *OUTPUT Connector*

The OUTPUT connector provides access to the 6 channel resistance outputs. Pinout of the D25 OUTPUT connector is as follows:

Pin Number	Description
2	Ch 0 E+
1	Ch 0 E-
15	Ch 0 S+
14	Ch 0 S-
4	Ch 1 E+
3	Ch 1 E-
17	Ch 1 S+
16	Ch 1 S-
6	Ch 2 E+
5	Ch 2 E-
19	Ch 2 S+
18	Ch 2 S-
8	Ch 3 E+
7	Ch 3 E-
21	Ch 3 S+
20	Ch 3 S-
10	Ch 4 E+
9	Ch 4 E-
23	Ch 4 S+
22	Ch 4 S-
12	Ch 5 E+
11	Ch 5 E-
25	Ch 5 S+

Pin Number	Description
24	Ch 5 S-
13	GND

Excite pins are electrically equivalent to sense pins of the same polarity.

4.3 *DIO (Digital Input/Output) Connector*

A female D9 DIO connector provides four digital input/output levels. Pinout is as follows:

Pin number	Description
2	DIO 0
3	DIO 1
4	DIO 2
5	DIO 3
6	+5V out, 100 mA max
7	not used
8	not used
1	GND
9	GND

As an input, each pin can accept a TTL input or switch closure to ground, with a threshold of approximately +1.6 volts. As an output, each pin can sink up to 32 volts at 250 mA. Inductive flyback is limited by a voltage clamp. A 1K pullup resistor can drive small LEDs or SSRs against ground.

The equivalent circuit of each DIO pin is as follows:

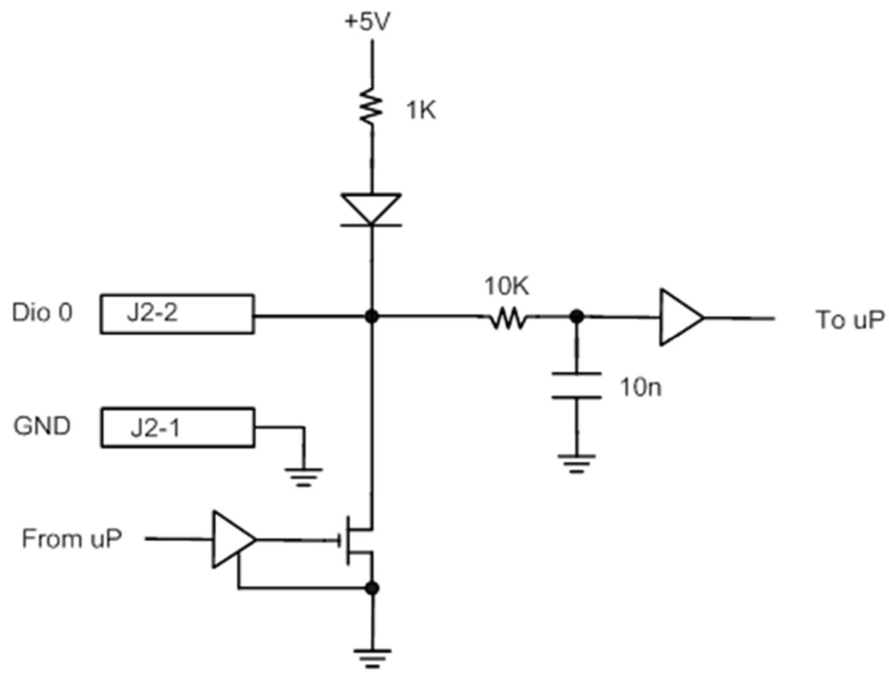

Figure 4. DIO Equivalent Circuit

4.4 *+24VDC Power Connector*

The P620 requires +24 volt DC power to its 2.1x5.5 mm DC barrel connector. The Highland model J24 supply is available.

If users furnish their own 24 volt power, the model J27 cable is available, with a screw-thread barrel connector for a durable, vibration-proof connection. The other end of the J27 terminates with bare leads.

4.5 *TEST Connector*

A male D9 TEST connector is provided for connection to an external precision DVM. Each output channel incorporates a relay (K0 in Figure 1) which allows it to be switched to the BIST BUS using the RELAYS command. The BIST BUS can also be connected to the TEST connector using the MUX command. This makes it possible to externally verify the channel output accuracy.

The Pinout of the TEST connector is as follows:

Pin number	Description
7	BIST BUS E+

Pin number	Description
6	BIST BUS E-
9	BIST BUS S+
8	BIST BUS S-
1	not used
2	not used
3	not used
4	not used
5	not used

4.6 *USB Connector*

The USB port is a micro A/B connector. It is not used for power. The USB port acts as a serial port emulator. For more information, please refer to section 7.1.

4.7 *ETHERNET Connector*

The Ethernet connector is a standard RJ45 10/100 connection and is used for communication using the web interface or using ASCII commands through TCP/IP. For more information, please refer to sections 7.2 and 7.3.

5 Operation

5.1 Simulating Resistance

The P620 can simulate any resistance from 5Ω to 5MΩ. This span is divided up into five overlapping spans as follows:

Range Name	Span (Ω)
R5	5 - 500
R50	50 - 5k
R500	500 - 50k
R5K	5k - 500k
R50K	50k - 5M

To set a resistance output on a channel, first set the resistance range using the SET command with the TYPE setting. Then set the output value using the VALUE command.

5.2 Simulating RTDs

The P620 allows the simulation of RTDs using internal lookup tables to convert a desired temperature into a resistance. The available RTD types are:

Type	RTD
R385	100Ω Platinum 385
K385	1kΩ Platinum 385
R392	100Ω Platinum 392
K392	1kΩ Platinum 392

Setting the channel output as an RTD is similar as for a resistance. But in this case, the VALUE command takes the temperature setpoint as an argument.

5.3 Power-up Defaults

The first time that a P620 powers up, the defaults for all 6 outputs are:

Setting	Value
Resistance Range	R50K
Resistance Value	50kΩ
Name	(blank)

The DIO will power up with all its channels set as inputs. The BIST BUS is connected to the D9 TEST connector and no channels are connected to the BIST BUS.

5.4 Quick Start Procedure

Basic operation of the P620 can be demonstrated by the following steps:

Power on the unit by plugging the barrel connector of the power supply into the P620 power connector. Plug the power supply into an available outlet.

Plug a USB-A to micro-USB cable into a computer and plug the other end into the P620 USB port. The computer should detect the P620 as a USB to Serial converter and automatically install the necessary drivers. A new comport should appear on the computer.

On the computer, start a terminal program such as PuTTY. Configure the session as serial using the new comport with the settings listed in section 7.1 and open the connection.

In the new terminal, type in the command

```
IDENT
```

You should get a reply something like this:

```
P620-1A SN xxxx FIRMWARE xxxx IP x.x.x.x MAC
xx:xx:xx:xx:xx:xx
```

Now type in this command:

```
USER 0xFF00
```

The front panel USR LED should begin to blink.

To set up channel 0 to output 100kΩ, send the following commands:

```
SET 0 TYPE R50K
```

```
VALUE 0 100000
```

Connect an ohm-meter to the D25 OUTPUT Connector. The high lines should connect to pins 2 and 15; the low to pins 1 and 14. The meter should read 100kΩ.

5.5 *Overload Protection*

A channel operated outside of the voltage or current limitations will go into overload protection mode. In this mode, the affected channel will become 50kΩ or greater to attempt to avoid damage. Once per second, the channel will return briefly to the user-requested settings to see whether the fault condition has been cleared. If the fault is cleared, the channel will remain as programmed. If the fault remains, it will return to overload protection.

6 *Performance*

6.1 *Resistance Resolution*

Though at all times resistance is specified as a decimal, the P620 will round this to the nearest value that can be represented by the hardware. The available resolution is a function of both the requested resistance and the selected range. While the ranges are capable of simulating resistances over a 100:1 span, on any given range the available resolution for a requested output resistance R_{OUT} increases as R_{OUT}^2 / R_{MIN}. Therefore for best performance use the highest range that will allow for the full simulation range. The resolution is described quantitatively in the figures below.

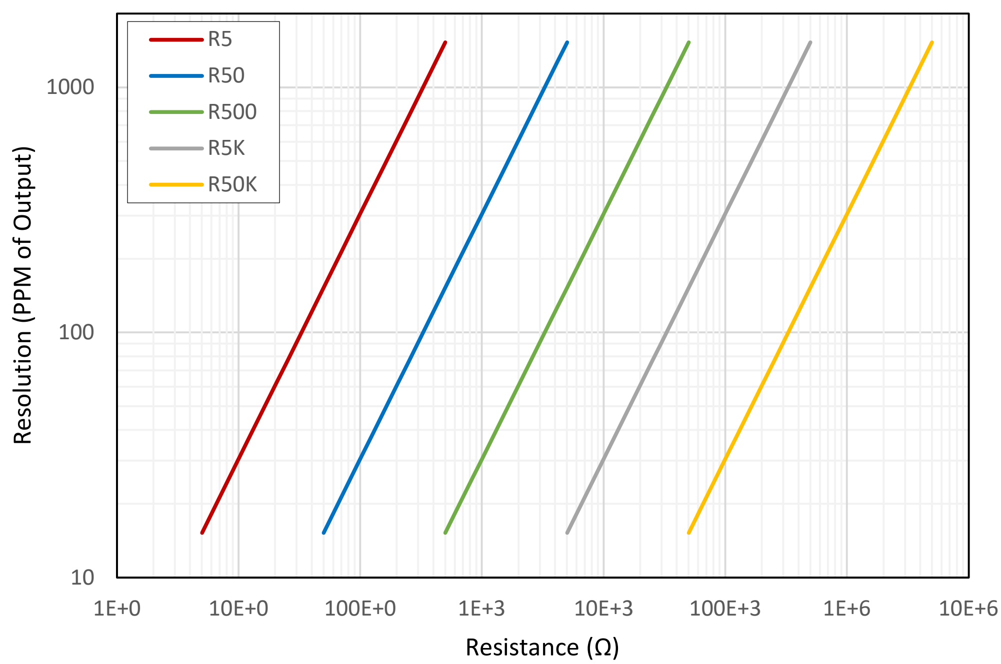

Figure 5. Resolution (PPM) as a function of output resistance for each range

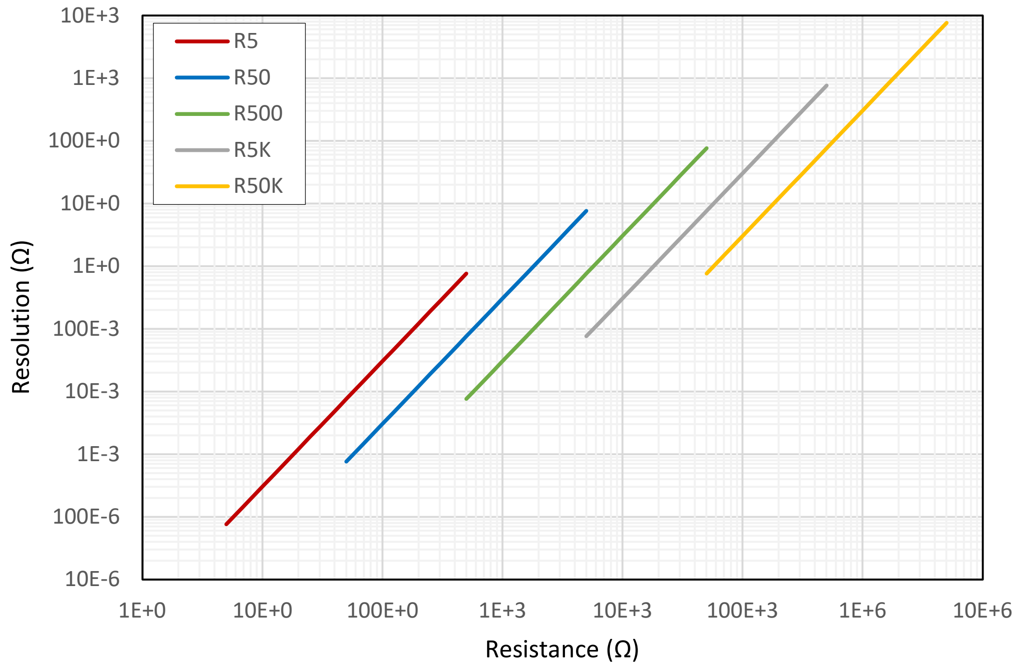

Figure 6. Resolution (Ω) as a function of output resistance for each range

6.2 *Resistance Simulation*

Resistance simulation accuracy is dependent on a number of factors. P620 unit temperature, selected range, and output setpoint have the biggest impact. In general, it is best to operate the P620 near room temperature and to use the highest applicable range for a given output. For example, if requesting 100Ω, it's better to use range R50 than R5. The following figures illustrate these points.

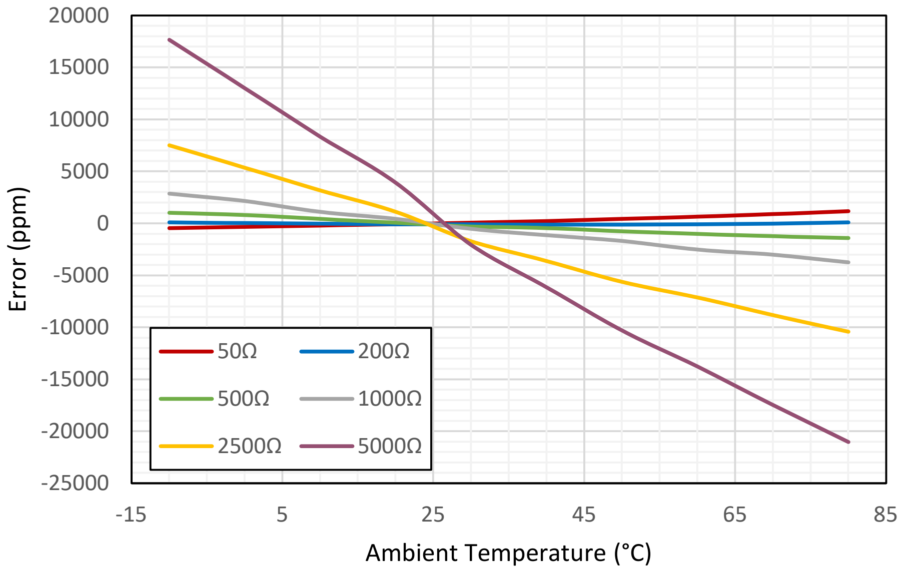

Figure 7. Typical output accuracy for range R50 over temperature for various setpoints within that range.

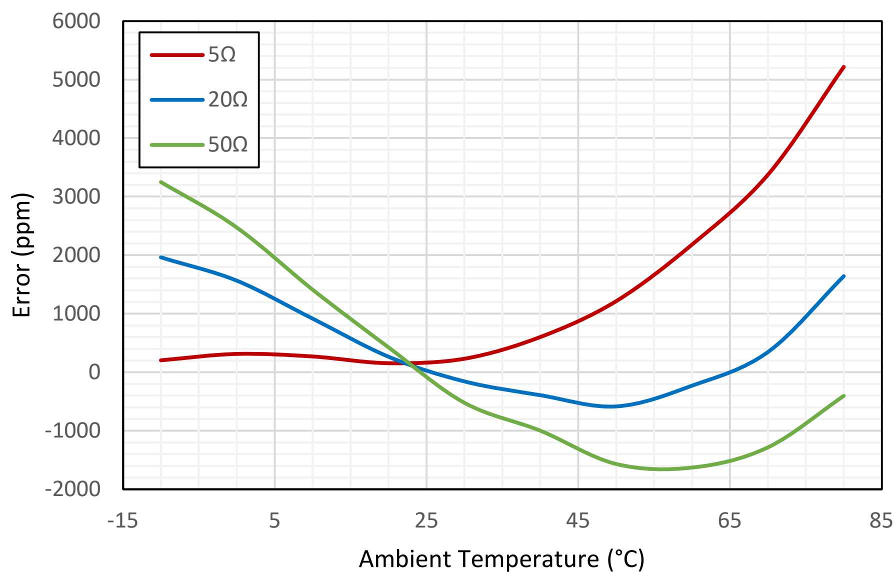

Figure 8. Typical output accuracy for range R5 over temperature for setpoints 1x, 4x, and 10x of the base resistance.

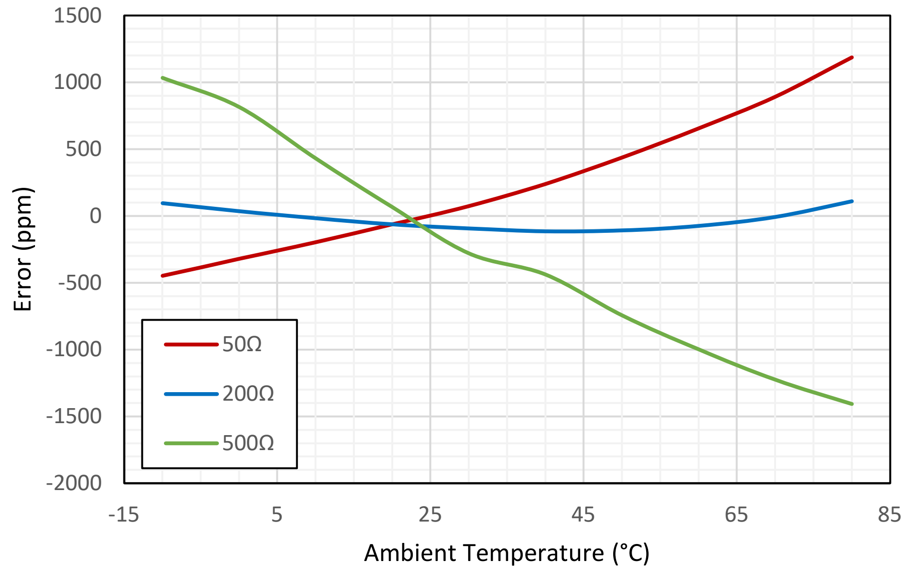

Figure 9. Typical output accuracy for range R50 over temperature for setpoints 1x, 4x, and 10x of the base resistance.

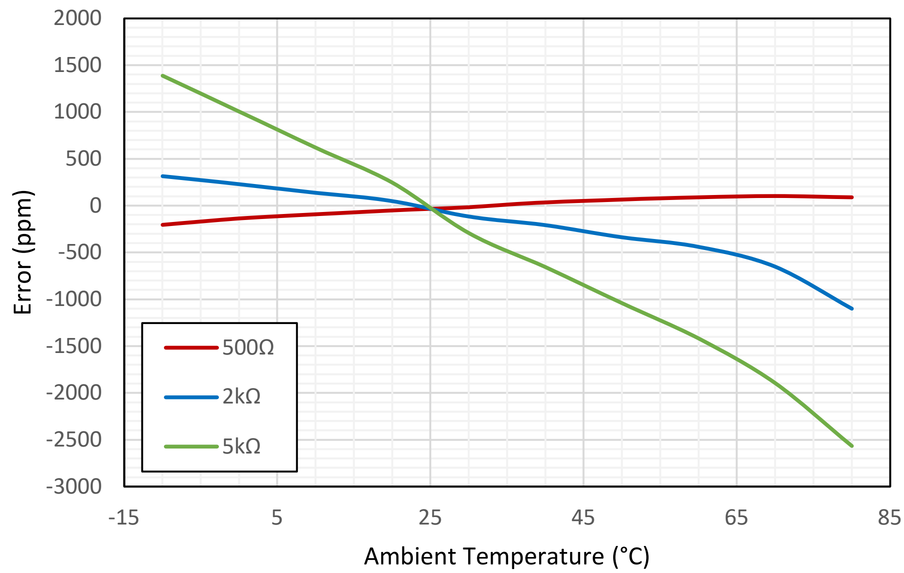

Figure 10. Typical output accuracy for range R500 over temperature for setpoints 1x, 4x, and 10x of the base resistance.

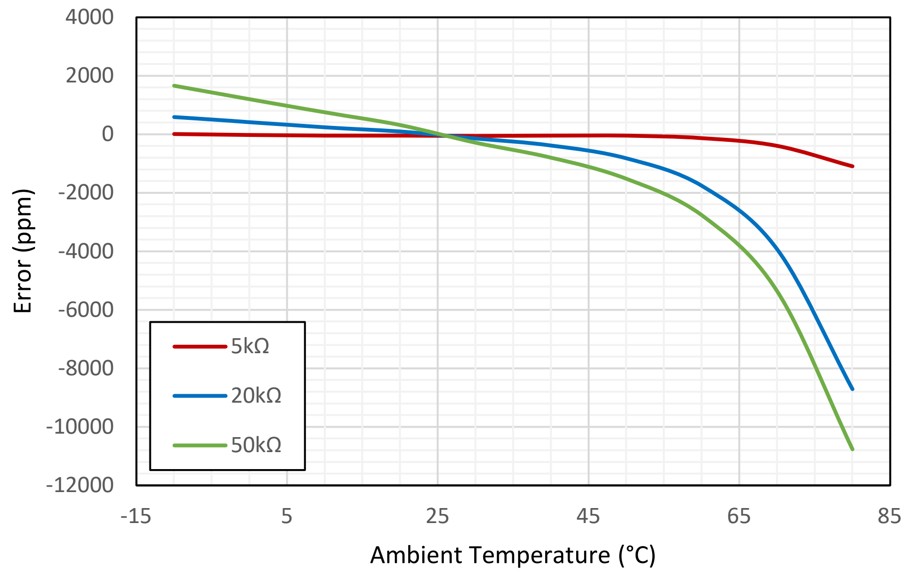

Figure 11. Typical output accuracy for range R5K over temperature for setpoints 1x, 4x, and 10x of the base resistance.

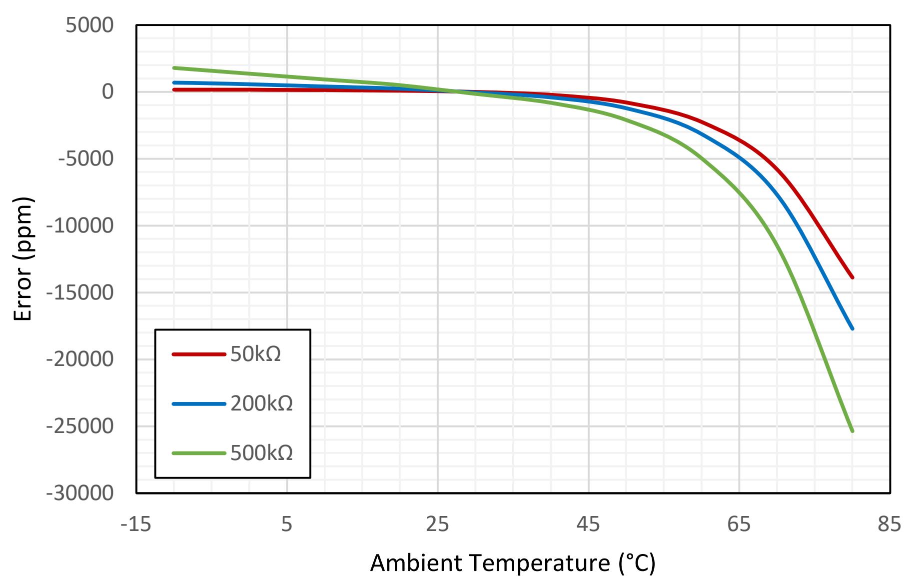

Figure 12. Typical output accuracy for range R50K over temperature for setpoints 1x, 4x, and 10x of the base resistance.

6.3 *Platinum RTD Simulation*

For Pt100 and Pt1000 RTDs, resolution is better than 0.02°C for RTD temperatures between -125°C and 250°C.

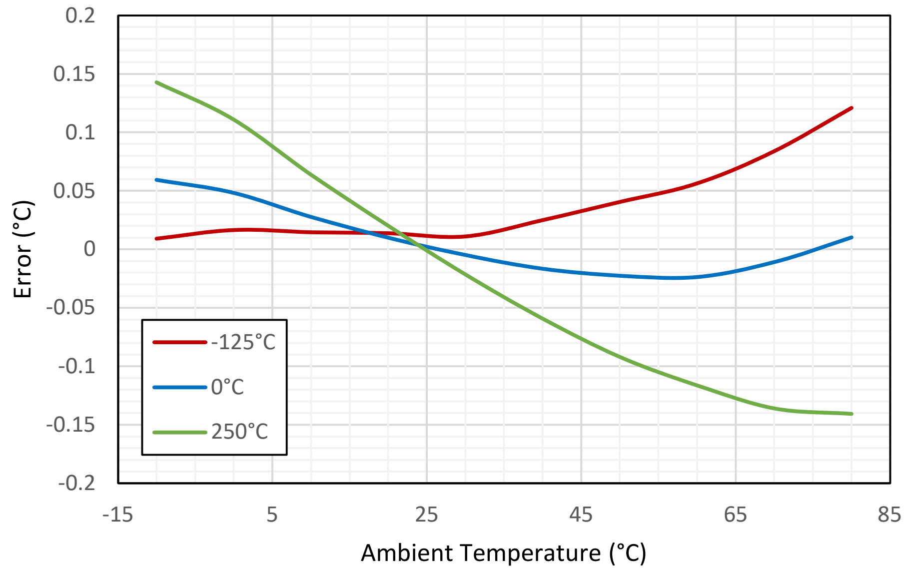

Figure 13. Typical output accuracy for 100Ω platinum RTD over ambient temperature for setpoints spanning -125°C to 250°C.

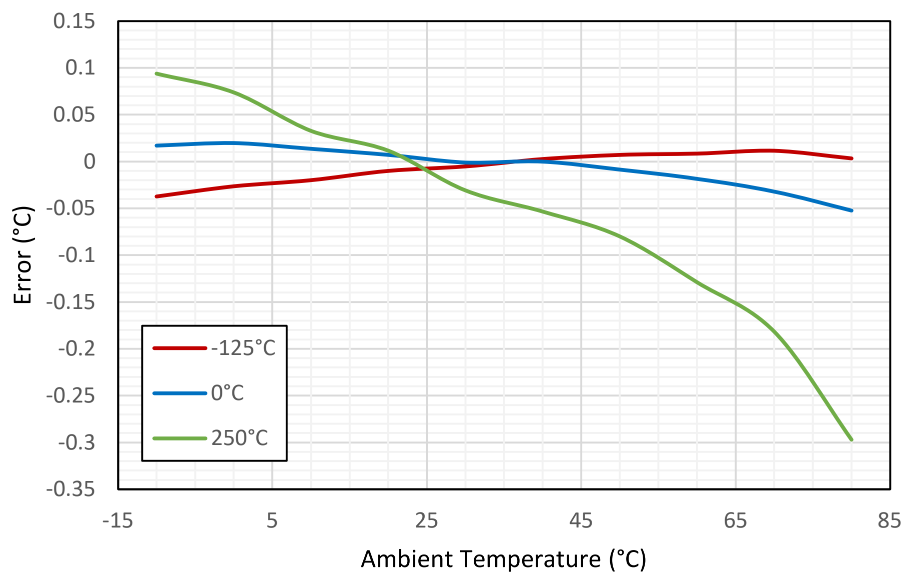

Figure 14. Typical output accuracy for 1kΩ platinum RTD over ambient temperature for setpoints spanning -125°C to 250°C.

6.4 *Monotonic Resistance Changes*

Most resistor and RTD simulators work by switching relays and connecting to a bank of available resistors. This means that there are dips and spikes when a new value is set. Unlike relay-controlled resistor simulators, the P620 output transitions smoothly to a new setpoint.

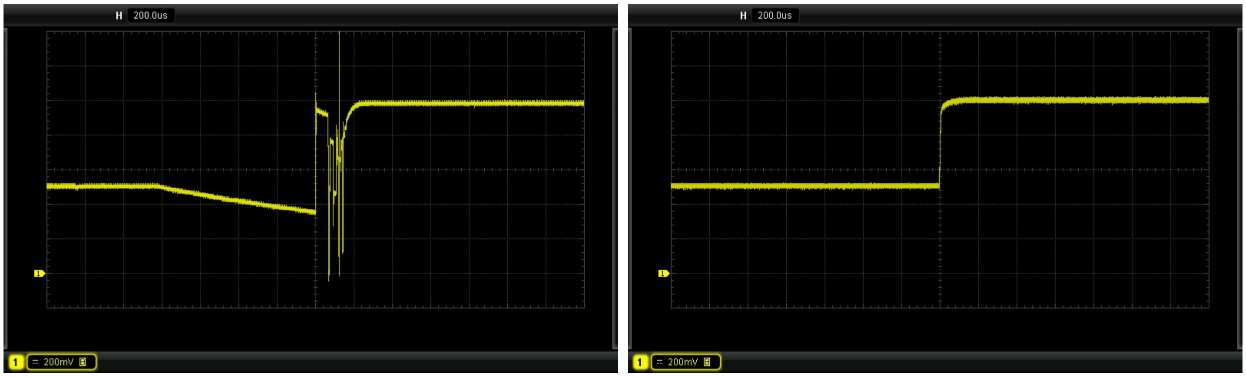

Figure 15. Transition from 50Ω to 100Ω for a relay-controlled simulator (left) and the P620 (right)

6.5 *Settling Time*

Given a rapidly changing input, as when multiplexing an ohmmeter between multiple channels, it is necessary to allow some time for the signal to settle before making a measurement. The settling time depends on the selected range and setpoint.

Base Resistance Setpoint Multiplier	Resistance Range				
	R5	R50	R500	R5K	R50K
1x	50	25	25	25	250
10x	300	150	150	150	1200
100x	1000	1000	1000	1000	1500

Table 1. Typical settling time (μS)

For RTDs, allow a settling time of 150μS.

For simplicity, it is easiest to allow a 2mS or greater settling time in all cases.

6.6 *Bandwidth*

It is possible to apply a waveform to the output of the P620, but there is a limit to the bandwidth if you wish to limit error in resistance. The bandwidth depends on the selected range and setpoint.

Base Resistance Setpoint Multiplier	Resistance Range				
	R5	R50	R500	R5K	R50K
1x	500	2k	14k	20k	1.5k
10x	500	2k	3k	1.5k	200
100x	200	300	200	150	100

Table 2. Typical bandwidth to limit DC-AC difference to 1% (Hz)

Typical bandwidth is 2kHz for 100Ω and 14kHz for 1kΩ RTDs.

7 *Communication*

7.1 *USB Interface*

All standard ASCII commands can be sent through the USB interface.

The P620 emulates a serial port using the FTDI FT230XS USB interface chip. This is the most common USB/serial interface chip so many operating systems will include appropriate drivers.

Documentation and drivers are available at http://www.ftdichip.com/

The serial port protocol is:

Baud	115,200
Data bits	8
Stop Bits	1
Parity	None

7.2 *TCP/IP Interface*

By default, the P620 IP address is set by DHCP. This may be changed using the IPADD and SUBNET commands. Static addressing is supported.

The P620 has a default hostname of “P620-xxxxx”, where xxxxx is the serial number of the unit, padded with zeros on the left to 5 digits. For example P620-00001.

Port 2000 is used for ASCII commands. Only one TCP session may be connected to Port 2000 at a time. Port 80 is used for web interface access.

7.3 *Web Browser User Interface*

The P620 can be accessed from a web browser. Recommended web browsers are Firefox and Chrome. Enter the IP address of the P620 in the browser address bar. It can also be accessed by its hostname if it is set to DHCP mode.

8 *ASCII Command Reference*

8.1 *General Comments*

The P620 accepts ASCII serial commands from the USB interface or from the Ethernet port.

In the following sections, text using this font

```
VAlue ALl 5<cr>
```

represents a command string sent to the P620, terminated with a carriage return character.

Italic text

```
OK<cr><lf>
```

represents the reply from the P620. All commands must be terminated with semicolon or carriage return (<cr>); all reply lines are terminated by carriage return + line feed (<cr><lf>).

Passing references to commands outside of examples will otherwise be referenced with this sentence’s font, all capitalized, e.g. VALUE.

A “query” command refers to a serial command that replies information about the P620’s state without changing it. A “set” command refers to a command that changes the P620’s state, usually replying OK.

8.1.1 *Command Strings*

Users send serial ASCII command strings to the P620, to which the P620 immediately replies. Because the P620 may spend milliseconds to process commands (and longer for the commands used for upgrading firmware), user software must wait for a response to each line of commands before sending another line.

Each command consists of one or more command keywords, followed by optional arguments. Multiple commands may be sent in a single line, separated by semicolons. When the full line is received, indicated by the final <cr>, the commands are executed in the order received.

Only the first two letters of a command keyword are significant. Keywords may be fully spelled out or sent as their first two letters. In this documentation, a word that has two possible forms is written with the short form capitalized, and the rest of the word in lower-case letters.

Examples:

`VAlue`	indicates that the short form is VA and the long form is VALUE, both of which are recognized commands.

All commands are case insensitive. At least one space or tab is required to separate command keywords from arguments.

Most set commands may be sent without an argument, in which case they become queries of the associated value.

`VAlue 4 347.2` *`OK`*	sets the channel 4 temperature to 347.2 °C
`VAlue 4` *`347.2`*	queries that setting

Certain incoming ASCII characters are treated specially:

- A blank input line (whitespace and <cr> only) evokes the response *`<cr><lf>`*.
- Double-quotation marks (ASCII 34, '`"`') are not ignored. They are used in the SET command for names that have spaces in them. Do not use them elsewhere or the P620's command parser may confuse them as part of the command or argument.

Numbers are evaluated as described in specific commands. When a floating point number is expected, use decimal or exponential notation. For example, use "0.123" or "123e-3", not "123m". When an integer is expected, a leading "0x" will be evaluated as a base-16 (hexadecimal) number, and other formats will be evaluated as base-10 (decimal). It will never be evaluated as octal.

8.1.2 *Reply Strings*

Each received command will evoke a reply indicating the execution status of the command. For query commands, the reply is the requested data. For most other commands, successful completion will yield a reply of *`OK`*. If multiple commands are issued on one line, multiple responses will be sent back on a single line, separated by semicolons. For example:

```
SET 1 TYPE K385; SET 4 TYPE R393; GET 1 TYPE
```

This command would reply

`OK; OK; CHAN 1 TYPE K385`

All reply strings are terminated with carriage return/linefeed <cr><lf> characters.

If an error occurs while processing a command or an incorrect command was entered, the reply will be an error number and the type of the error. If the error occurred for a line with multiple commands, the P620 will abort without performing the remaining commands on the line.

The errors are:

Error message	Comments
`E01: Command not found`	
`E02: Argument missing or invalid`	
`E03: Invalid range`	A specific version of E02.
`E04: Hardware error`	Always considered a critical error.

Error message	Comments
E05: Flash locked	Returned if either FLASH WRITE or FLASH ERASE is used before FLASH UNLOCK.
E07: Checksum fail	Returned if a LOAD command is used but the user settings as read from flash fail a checksum. Unless a SAVE command has never been used, this is a critical error.
E08: Help file not found	Replied if the help file in flash cannot be accessed. This is a critical error, because it indicates either an invalid firmware image in the flash or a flash hardware error.
E10: Not permitted	Replied if the commands SAVE, FLASH UNLOCK, FLASH WRITE, or FLASH ERASE are used while the DIP switches do not permit modification of nonvolatile memory (see section 10, Security Provisions).

8.2 *Command Summary*

The following is a summary of ASCII commands which may be sent to the P620:

LONG FORM	SHORT FORM	FUNCTION
SET	SE	set channel operating modes
GET	GE	return channel operating modes
VALUE	VA	set channel output, resistance or °C
DIO	DI	set/report digital i/o levels
IDENT	ID	identify unit hardware and firmware
USER	US	load "user" LED blink pattern
IPADD	IP	manage IP address
SUBNET	SU	manage subnet mask
MAC	MA	query MAC address
BOOT	BO	restart unit
SAVE ALL	SA AL	save all user settings in nonvolatile memory
SAVE SETUPS	SA SE	save channel setups
SAVE VALUES	SA VA	save channel output values
SAVE DIO	SA DI	save the DIO output setting, 0 to 15
SAVE IPADD	SA IP	save the IP settings
LOAD ALL	LO AL	restore all saved items
LOAD SETUPS	LO SE	restore saved channel setups
LOAD VALUES	LO VA	restore saved channel values
LOAD DIO	LO DI	restore saved DIO output state
LOAD IPADD	LO IP	restore saved IP address
LOAD DEFAULTS	LO DE	set the P620 to the default setup
RELAYS	RE	control channel test relays
MUX	MU	control BIST BUS test relays
BIST	BI	control self-test
HELP	HE	return command summary
STATUS	ST	return status reports
NETSTAT	NE	return TCPIP status
EXIT	EX	close TCP session
FLASH UNLOCK	FL UN	enable firmware upgrade operations
FLASH ERASE	FL ER	erase upgraded firmware in flash
FLASH WRITE	FL WR	write S-Record line to flash
FLASH CHECKSUM	FL CH	run checksum check of firmware images

8.3 *Command Details*

8.3.1 *Channel SET*

Description:

The SET command establishes channel operating modes.

Command Format:

```
SEt <channel-list> <setting> <value> [<setting> <value> [. . .]]
```

Arguments:

`<channel-list>`	A list of channels without spaces (EG “234” for channels 2, 3, and 4), or “ALL” for all channels (equivalent to “012345”).
`<setting>`	Either TYPE or NAME, discussed in the sections below. As with command words, setting names are case-insensitive, and only their first two letters are evaluated.
`<value>`	Dependent on <setting>, see following sections.

Example:

`SEt 234 TYpe R385 NAme "Ref temp"` *OK*	Set channels 2, 3, and 4 to type 100 Ω (Pt385) RTD, and name them all “Ref temp”
`SEt ALl TYpe R500` *OK*	Set all channels to simulate a resistor in the range 500 Ω to 50 kΩ.

8.3.1.1 *TYPE Setting*

Set the channel range/RTD type. Value for TYPE is one of the following:

Value Argument	**Channel type**	**Range**
R385	RTD 100Ω (Pt385)	-125°C to 700°C
R392	RTD 100Ω (Pt392)	-120°C to 650°C
K385	RTD 1 kΩ (Pt385)	-125°C to 700°C
K392	RTD 1 kΩ (Pt392)	-120°C to 650°C
R5	Resistor	5 Ω to 500 Ω
R50	Resistor	50 Ω to 5 kΩ
R500	Resistor	500 Ω to 50 kΩ

`R5K`	Resistor	5 kΩ to 500 kΩ
`R50K`	Resistor	50kΩ to 5MΩ

The expression is case-insensitive, but all characters are expected.

8.3.1.2 *NAME Setting*

Declare a channel name, 63 characters max. If there are spaces, use double quotes (‘`"`’) or only the first word will be used. Do not include semicolons or newline characters in the name. The name’s letter case will be preserved.

`SEt 2 NAme "Load 4"` declares the name “Load 4” for channel 2.

By default, channels are not named. To clear a name, use empty double-quotes. For example:

`SEt 2 NAme ""`

8.3.2 *Channel GET*

Description:

GET is the query equivalent of SET.

Command Format:

`GEt <channel-list> [<setting> [<setting>]]`

Arguments:

`<channel-list>`	A list of channels without spaces (EG “234” for channels 2, 3, and 4), or “ALL” for all channels (equivalent to “012345”).
`<setting>`	Either TYPE or NAME. As with command words, setting names are case-insensitive, and only their first two letters are evaluated.

Query Reply Format:

`CHANNEL <n> <setting> <value> [<setting> <value>] [, CHANNEL <n> ...]`

Example:

`GEt ALl` *`CHAN 0 TYPE R50 NAME "", [...]`*	returns all channels’ settings, delimited by commas for each channel.
`GEt 5 TYpe` *`CHAN 5 TYPE R500`*	returns thermocouple type of channel 5.

The empty quotes (*`""`*) in this example indicate that the name for channel 0 has not been set by the user. This ensures consistency in the number of replied string tokens.

If no setting arguments are given, than all settings will be given for each channel requested.

8.3.3 *Channel VALUE*

Description:

The VALUE command sets or queries the output setpoint of a channel.

Command Format:

```
VAlue <channel-list> [<value>]
```

Arguments:

`<channel-list>`	A list of channels without spaces (EG “234” for channels 2, 3, and 4), or “ALL” for all channels (equivalent to “012345”).
`<value>`	The numeric setpoint value in °C or ohms, appropriate to the channel TYPE setting.

Use decimal notation, not exponential or engineering notation. For example, use “0.22”, not “220e-3” or “220m”.

The value is limited to the range determined by the channel’s TYPE setting. If the programmed value exceeds the defined range, the simulated output will be clipped to the lower or upper limit of that type and the red LED will indicate a channel programming error.

Query Reply Format:

```
<value>[, <value>[…]]
```

Queries are replied for channels in the order in which they were written in the channel list.

Replies to queries will print to a precision of three decimal places.

Example:

`VAlue 4 -25.7`	sets channel 4 to -25.7°C, RTD mode.
`VAlue 3 725.8`	sets channel 3 to 725.8Ω.
`VAlue 34` *`725.800; -25.700`*	queries the current setting of channels 3 and 4, in that order.

8.3.4 *DIO*

Description:

The DIO command controls the digital inputs/outputs on the D9 DIO connector.

Command Format:

```
DIo [<value>]
```

Command Arguments:

`<value>`	Decimal or hex representation of the DIO channel settings as bits. To set a channel as an input, set its bit OFF. To set it as and output, logic high, set its bit OFF. To set it as an output, logic low, set its bit ON. For example: Set channels ON — <value> 0 — 1 1 — 2 2 — 4 2 and 3 — 12

Query Reply Format:

`<output> <input>`

Reply Arguments:

`<output>`	Integer representation of the DIO channel settings as bits. A bit set ON represents the output set logic low.
`<input>`	Integer representation of the DIO channel inputs as bits. A bit set ON represents logic high.

Example:

`DIo 15` (or `DIo 0xF`) *OK*	Turns all four pull-down drivers on, electrically low.
`DIo 2` *OK*	Turns DIO 1 on/low, others high/inputs
`DIo` *2 13*	Queries DIO states, returning set levels and actual levels, as decimal values.

8.3.5 *IDENT*

Description:

The IDENT command returns an ID string.

Command Format:

`IDent`

Query Reply Format:

`P620-1A SN xxxx FIRMWARE xxxx IP x.x.x.x MAC xx:xx:xx:xx:xx:xx`

The “-1A” following “P620” means “dash 1, hardware revision A”. The number following “SN” is the serial number. Firmware will usually be 23E620 followed by the firmware revision letter, for example 23E620A.

8.3.6 *USER*

Description:

The USER command loads or queries the blink pattern for the orange USER LED.

Command Format:

`USer [<value>]`

Arguments:

`<value>`	A 16-bit-wide integer, whose bits are shifted at 8 Hz. It may be either decimal or hexadecimal notation.

Query Reply Format:

`<value>`

Replies in hex format.

Example:

`USer 0` *OK*	Turns the LED off
`USer 0xFFFF` *OK*	Turns the LED full on
`USer 0xFF00` *OK*	Would blink continuously at 0.5 Hz
`USer` *0xFF00*	Queries the last set state of the user led. Reply is in hex notation.

8.3.7 *IPADD*

Description:

The IPADD command sets or queries the Ethernet IP address. The setting takes place immediately. Use the SAVE IPADD command to save a new IP address in nonvolatile memory, for use after a power-down/power-up cycle.

Command Format:

`IPadd [<value>]`

Arguments:

`<value>`	Either the static IP address in the format xxx.xxx.xxx.xxx or “DHCP” for DHCP mode. An address of 0.0.0.0 sets DHCP mode.

Query Reply Format:

`xxx.xxx.xxx.xxx`

Example:

`IPadd 192.168.254.183` *OK*	Sets a static IP address to 192.168.254.183
`IPadd 0.0.0.0` *OK*	Sets DHCP mode
`IPadd DHCP` *OK*	Sets DHCP mode
`IPadd` *0.0.0.0*	Queries the IP. If in DHCP mode, replies “0.0.0.0”.

8.3.8 *SUBNET*

Description:

The SUBNET command sets or queries the Ethernet subnet mask. This only takes effect when not in DHCP mode.

Command Format:

`SUbnet [<value>]`

Arguments:

`<value>`	The subnet mask in the format xxx.xxx.xxx.xxx

Query Reply Format:

xxx.xxx.xxx.xxx

Example:

`SUbnet 255.255.255.0` *OK*	Sets the subnet mask
`SUbnet` *255.255.255.0*	Queries the subnet mask

8.3.9 *MAC*

Description:

The MAC command queries the factory-set MAC address in colon-delimited notation.

Command Format:

`MAc`

Query Reply Format:

xx:xx:xx:xx:xx:xx

Example:

`MAc` *`00:0A:12:34:56:78`*	Queries the MAC address

8.3.10 *BOOT*

Description:

The BOOT command reboots the P620 software, simulating a power-up. All items saved by the SAVE commands will be restored. The P620 will not send an acknowledgement for this command.

Command Format:

`BOot`

Example:

`BOot`	Reboots the P620

8.3.11 *SAVE and LOAD*

Description:

SAVE commands save current settings into nonvolatile memory. Load commands set current settings to match those saved in nonvolatile memory. At power-up, all saved items are restored, equivalent to LOAD ALL.

Command Format:

`SAve <value>`

`LOad <value>`

Arguments:

`<value>`	One of the following: SEtups – Saves/loads channel setups VAlues – Saves/loads channel output values DIo – Saves/loads DIO output setting IPadd – Saves/loads the IP Address and subnet mask All – Saves/loads all of the above DEfaults – Load only. Sets all outputs to type R50K, 50kΩ; All names set blank; All DIO set OFF; IP address set to DHCP; Subnet mask set to 255.255.255.0

Example:

`SAve ALl` *`OK`*	Saves all settings
`LOad DEfaults` *`OK`*	Loads defaults

8.3.12 *RELAYS*

Description:

The RELAYS command sets or queries the channel BIST relays. A channel's BIST relay reroutes its output to the BIST BUS. This is useful for calibration verification (see section 11). Only one channel may be connected at a time.

Command Format:

`RElays [<value>]`

Arguments:

`<value>` *OK*	Either a channel number, 0-5, to connect that channel to the BIST BUS; or "OFF" to return all channels to normal function.

Query Reply Format:

<value>

Example:

`RElays 1` *OK*	Connects channel 1 to the BIST BUS
`RElays 2` *OK*	Disconnects channel 1 and connects channel 2 to the BIST BUS
`RElays` *2*	Queries the channel BIST relays
`RElays Off` *OK*	Sets all channels to normal function

8.3.13 *MUX*

Description:

The MUX command sets or queries relays which connect the BIST BUS in different configurations. This is useful for calibration verification (see section 11).

Command Format:

`MUx [<value>]`

Arguments:

`<value>`	One of the following: D9 – Route the calibration bus to the D9 TEST connector. This is the power-on default. ADc – Route the calibration bus to the internal BIST ADC. HIgh - Apply a +24V pullup to the calibration bus E+/E-. LOw - Apply a -24V pulldown to the calibration bus E+/E-.

Query Reply Format:

<value>

Example:

`MUx D9` *`OK`*	Connects the BIST BUS to the D9 TEST connector
`MUx` *`D9`*	Queries the BIST BUS configuration

8.3.14 *BIST*

Description:

The BIST command runs a full self-test, checks the status of a currently running test, or checks the result of a previously run test.

BIST tests internal power supplies and all channel outputs. This is a functional test and cannot verify P620 calibration, which would require external NIST traceable references.

The control over the relays of the P620 will not be available until completion of BIST. If a RELAYS command or any other command that requires setting relays is sent, such as certain LOAD commands, the request will be saved, OK will be replied, and the relays will be set to the new value when BIST is complete.

Except for BIST REPORT, which is more verbose, the reply will be a single line.

Command Format:

`BIst <value>`

Arguments:

`<value>`	One of the following: GO – Starts a BIST sequence ABORT – Stops a running BIST sequence POWER – Shows the values of measured power supplies STATUS – Shows current BIST state: NOT RUN, RUNNING, ABORTED, PASS, or FAIL REPORT <chan> – If BIST is complete, shows the verbose results of BIST for the listed channel.

Example:

`BIst REport` *`NOT RUN`*	Queries the previous BIST result, which does not exist
`BIst GO` *`OK`*	Starts a BIST sequence

`BIst STatus` *RUNNING*	Queries the current BIST sequence status, which is still running
`BIst STatus` *PASS*	Queries the current BIST sequence status, which is now complete
`BIst REport 0` *range=5, expect=5.0, read=5.0* *PASS* *[…]*	Queries the BIST results for channel 0

8.3.15 *HELP*

Description:

The HELP command returns documentation about the serial command set.

The HELP command replies a variable number of lines, and is intended to be a visual aide in an evaluation terminal such as TerraTerm or PuTTY.

Command Format:

`HElp [<value>]`

Arguments:

`<value>`	For documentation of a specific command, use that command as an argument. With no argument, HELP returns a summary of available commands.

8.3.16 *STATUS*

Description:

The STATUS command returns a summary report on the state of the P620.

If STATUS is followed by an argument, a single-line report will follow. Arguments are case-insensitive, and only their first two characters will be evaluated.

Command Format:

`STatus [<value>]`

Arguments:

`<value>`	One of the following: DIp – Show state of DIP switches. Reply is a number 0 to 15, representing a bitfield of the four DIP switches. A value of 15 means “all on”. A value of 0 means “all off”. Switch 1 is the LSB. IMage – Show which firmware image is currently loaded. Reply is either *FACTORY* or *UPGRADE*. If the upgrade image is running and the flash was modified by either a FLASH ERASE or FLASH WRITE command since

	the last boot, then UPGRADE will be followed by DIRTY. This means that the current running image no longer matches any image in the flash; a reboot is necessary for a new upgrade image to take effect. UPtime – Show the number of seconds of uptime. CAl – Show the status of calibration. Reply is `OK`, `DEFAULT`, or `DIRTY`. POwer – Shows the voltage of each power supply: 3.3V and 1.2V. The reply might look like: *"3.280 1.201"* SErial – Shows the unit's serial number. TEmperature – Shows the unit's temperature in °C Error – Show the unit's error status. 1 for error, 0 for none.

Query Reply Format:

<value> [<value>]

Example:

STatus DIp *0*	Show state of DIP switches
STatus IMage *FACTORY*	Show which firmware image is currently loaded
STatus UPtime *1234.12*	Show the number of seconds of uptime
STatus CAl *OK*	Show the status of calibration
STatus POwer *3.280 1.201*	Show the voltage of each power supply. Numbers are: <table><tr><th>#</th><th>Monitor</th><th>Units</th></tr><tr><td>1</td><td>+24V Input</td><td>V</td></tr><tr><td>2</td><td>+5V</td><td>V</td></tr><tr><td>3</td><td>+3.3V</td><td>V</td></tr><tr><td>4</td><td>+1.2V</td><td>V</td></tr><tr><td>5</td><td>+24V Current</td><td>A</td></tr><tr><td>6</td><td>Relay Power</td><td>V</td></tr><tr><td>7</td><td>Customer +5V</td><td>V</td></tr><tr><td>8</td><td>Unused</td><td>NA</td></tr></table> Note: revision A hardware gives only 3.3 and 1.2V data.
STatus SErial *1*	Show the unit's serial number

STatus TEmperature *29.8*	Show the unit's temperature
STatus Error *0*	Show the unit's error status

8.3.17 *NETSTAT*

Description:

The NETSTAT query returns a summary report of the TCPIP connection.

NETSTAT IP differs from IPADD command in that IP replies 0.0.0.0 when in DHCP mode, while NETSTAT IP replies the actual IP address. If DHCP is set (NETSTAT DHCP replies 1) but the NETSTAT IP keeps replying 0.0.0.0 without ever changing, there may not be a DHCP server on the network.

Command Format:

`Netstat [<value>]`

Arguments:

<value>	One of the following: IP – Shows the current IP address HOst – Shows the hostname DHcp – Replies 1 if DHCP, 0 if static IP Link – Replies 1 if the 100-Mbit link is up

Query Reply Format:

<value> (if given an argument)

<IP> <hostname> <DHCP> <link> (if no argument given)

Example:

NEtstat *192.168.1.20 P620-00001 1 1*	Show full TCPIP status
NEtstat IP *192.168.1.20*	Show the current IP address
NEtstat HOst *P620-00001*	Show hostname

8.3.18 *EXIT*

Description:

The EXIT command, whether executed from the USB interface or the TCP interface, will close any open TCP session on port 2000. To ensure that a TCP session ends cleanly, use this command before closing the session on the client side. The P620 will not send an acknowledgement for this command.

Command Format:

`EXit`

Example:

`Exit`	Closes the TCP session.

8.3.19 *FLASH UNLOCK*

Description:

The FLASH UNLOCK command enables firmware upgrade operation. The DIP switches must be configured to permit writes to nonvolatile memory.

This command is required before the FLASH ERASE and FLASH WRITE commands, as a safety against accidental calls to either from taking effect. This command only needs to be called once. It will remain in effect until a reboot.

Command Format:

`FLash UNlock`

Example:

`FLash Unlock` `OK`	Enables firmware upgrade

8.3.20 *FLASH ERASE*

Description:

The FLASH ERASE command erases the upgrade portion of the boot flash. The FLASH UNLOCK command must have been called first, and the DIP switches must be configured to permit writes to nonvolatile memory.

This command may take up to thirty seconds to execute. Most of the P620's normal operation, including channel outputs and the web server, will be stalled during this time.

Command Format:

`FLash ERase [VErbose]`

Arguments:

`VErbose`	Replies a line of the form: *`ERASING SECTOR i OF n<cr><lf>`* for every flash sector being erased, and then *`OK`* when all sectors have been erased.

Example:

`FLash Erase` *OK*	Erases upgrade flash
`FLash ERase VErbose` *ERASING SECTOR 16 OF 31* *ERASING SECTOR 17 OF 31* *[...]* *OK*	Erases upgrade flash

8.3.21 *FLASH WRITE*

Description:

The FLASH WRITE command writes a line of an S-Record to the upgrade portion of the boot flash. The FLASH UNLOCK command must have been called first, and the DIP switches must be configured to permit writes to nonvolatile memory. A typical S-Record line is less than 80 characters.

Only data-type S-Record lines (those that start with "S1", "S2", or "S3") will be evaluated. Others will be ignored and OK will be replied. If the address field of the S-Record is out of the range of the upgrade firmware flash address, an error will be returned. Do not ignore this error. It could imply that the wrong file is being used for upgrading the firmware.

Command Format:

```
FLash WRite <s-record line>
```

Arguments:

`<s-record line>`	A single line of the S-record.

Example:

`FLash WRite S2241001001<remainder of line...>` *OK*	Writes a single line of the S-record

8.3.22 *FLASH CHECKSUM*

Description:

The FLASH CHECKSUM command initiates a checksum of the factory and upgrade firmware images in the P620's flash and queries their status. This may take up to five seconds.

Command Format:

```
FLash CHecksum
```

Query Reply Format:

FF:<result>, FH:<result>, UF:<result>, UH:<result>

FF stands for “factory firmware”.

FH stands for “factory html” (the files used for the P620’s web interface).

UF stands for “upgrade firmware”.

UH stands for “upgrade “html”.

The result is one of the following:

OK Image is present and the checksum passed

NP Image was not found

ER Image was found but the checksum failed

The reply for the factory images should always be OK. The reply for the upgrade images should both be either NP (if never upgraded) or OK. The expected default result for a new P620 is:

FF:OK, FH:OK, UF:NP, UH:NP

Example:

FLash CHecksum *FF:OK, FH:OK, UF:NP, UH:NP*	Queries the flash checksum status

9 *Built-In Self-Test (BIST)*

The BIST facility allows users to verify functionality and rough accuracy of the P620 without any external equipment. Note that BIST does not provide absolute verification of module accuracy, as external NIST-traceable standards are required for formal calibration.

Self-tests are provided by means of the `BIST` command. After running BIST a report will be returned summarizing test results.

During the BIST run, the internal power supplies, error detection, and channel outputs are checked. Each of the 6 channels are tested using three points of their full output range. While running, the PWR LED will blink twice each second, versus once every 2 seconds during normal operation.

Total test time is about 15 seconds. At the end of the test, the BIST report will indicate if any errors were detected and the result of each test.

While a channel is being tested, its test relay is operated, disconnecting it from the front-panel D25 connector. When a channel test is finished, the channel output is restored to its programmed level and the relay de-energized to restore normal operation.

Although the BIST operations can detect most module failures, certain errors can be missed. They include:

- Failure of a channel test relay
- Failure of a connector pin or associated printed-circuit traces.

A sample BIST report looks like the following:

```
range=5, expect=5.0, read=5.0 PASS
range=5, expect=250.0, read=250.0 PASS
range=5, expect=500.0, read=500.0 PASS
range=50, expect=50.0, read=50.0 PASS
range=50, expect=2500.0, read=2500.0 PASS
range=50, expect=5000.0, read=5000.0 PASS
range=500, expect=500.0, read=500.0 PASS
range=500, expect=25000.0, read=25000.0 PASS
range=500, expect=50000.0, read=50000.0 PASS
range=5K, expect=5000.0, read=5000.0 PASS
range=5K, expect=250000.0, read=250000.0 PASS
range=5K, expect=500000.0, read=500000.0 PASS
```

```
range=50K, expect=50000.0, read=50000.0 PASS
range=50K, expect=250000.0, read=250000.0 PASS
range=50K, expect=1000000.0, read=1000000.0 PASS
HIGH-V, expect=1, read=1 PASS
LOW-V, expect=1, read=1 PASS
PASS
```

10 *Security Provisions*

There may be cases where it is desirable to disallow any changes to the nonvolatile memory internal to the P620. The nonvolatile memory saves channel setups, including channel names, and the calibration table.

There is an internal DIP switch that can be accessed by removing the top cover, and the cover can be sealed with tamper-evident stickers if desired.

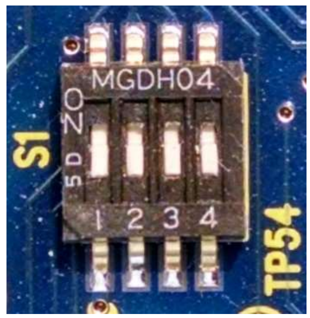

Figure 16. DIP Switch with all sections OFF

To disable all write access to nonvolatile memory, set section 1 ON.

To disable write access to the calibration table, set section 4 ON.

The factory default is for all switches to be OFF.

To view the switch state, use the STATUS DIP command.

11 *Calibration Verification*

Calibration of one or more P620 modules may be verified using the D9 TEST connector. Multiple Highland P-series units may be bussed to a common test cable and test instruments provided that only one box is operating in calibration verification mode at any one time.

The P620 module or modules may be connected to a precision digital multimeter, such as a Fluke 8845A, Keithley 2000, or equivalent. The required connections are:

TEST D9 PIN	P620 FUNCTION	DVM CONNECTION
7	Channel output (+)	INPUT HI
6	Channel output (-)	INPUT LO
9	Channel output (+)	SENSE HI
8	Channel output (-)	SENSE LO

To verify a channel output, use the RELAYS and MUX commands to route a channel out through the D9 test connector into the DMM voltage input leads. Users may then program the channel and verify accuracy using DMM measurements.

The serial commands to monitor the channel 3 output would be

`MUx D9`	routes BIST BUS to D9 connector
`RElays 3`	connects channel 3 to the BIST BUS

Send RELAYS OFF when done.

The J75-1 D9 female to two dual banana plug cable is available as an accessory to easily connect the D9 test connector to a multimeter.

12 *Boot Flow and Firmware Upgrade*

The P620 contains two major address spaces in its boot ROM (the "flash"): one for the factory image, and one for an upgrade image.

The P620's boot loader will search for and checksum the upgrade image in the flash. If the upgrade image exists and it has a valid checksum, the P620 will boot into it. Otherwise the P620 will boot into the factory fallback image.

If its DIP switches are configured to permit writes to nonvolatile memory, the P620 may have its firmware upgraded by using the FLASH commands. The P620's software only permits modification of the upgrade firmware, not the factory firmware.

The firmware upgrade procedure requires an upgrade image, in the form of an S-Record file (with a ".s28" extension) provided by Highland Technology. The procedure is as follows:

1. Send FLASH UNLOCK to the P620, to enable modifying the flash. Wait for the reply OK.

2. Send FLASH ERASE to the P620. Wait for the reply OK. This may take up to thirty seconds.

3. Open the S-Record file. Perform the following loop until you reach end-of-file:

   Read a line from the S-Record.

   Send FLASH WRITE, with the line from the S-Record as an argument, to the P620.

   Wait for the reply OK.

4. Send FLASH CHECKSUM to the P620. Wait up to five seconds for the reply. The expected reply is:

   FF=OK, FH=OK, UF=OK, UH=OK

   This indicates that the upgrade image is now present and its firmware and web-page checksum tests (UF and UH) passed. (FF and FH should always be followed with OK.)

5. Send BOOT to the P620 to reboot it. If the upgrade procedure was successful, then the reply to STATUS IMAGE will be UPGRADE.

Software used for upgrading the firmware should also check for replies of the type:

Enn: description of error

The flash upgrade procedure should abort upon such errors.

A new P620 unit will ship with only its factory firmware image installed.

13 *Dimensions and Mounting*

The P620 has mounting holes on the bottom in case you want to secure it. They are labeled “B” in Figure 17 and are threaded 4-40. Dimensions are in inches.

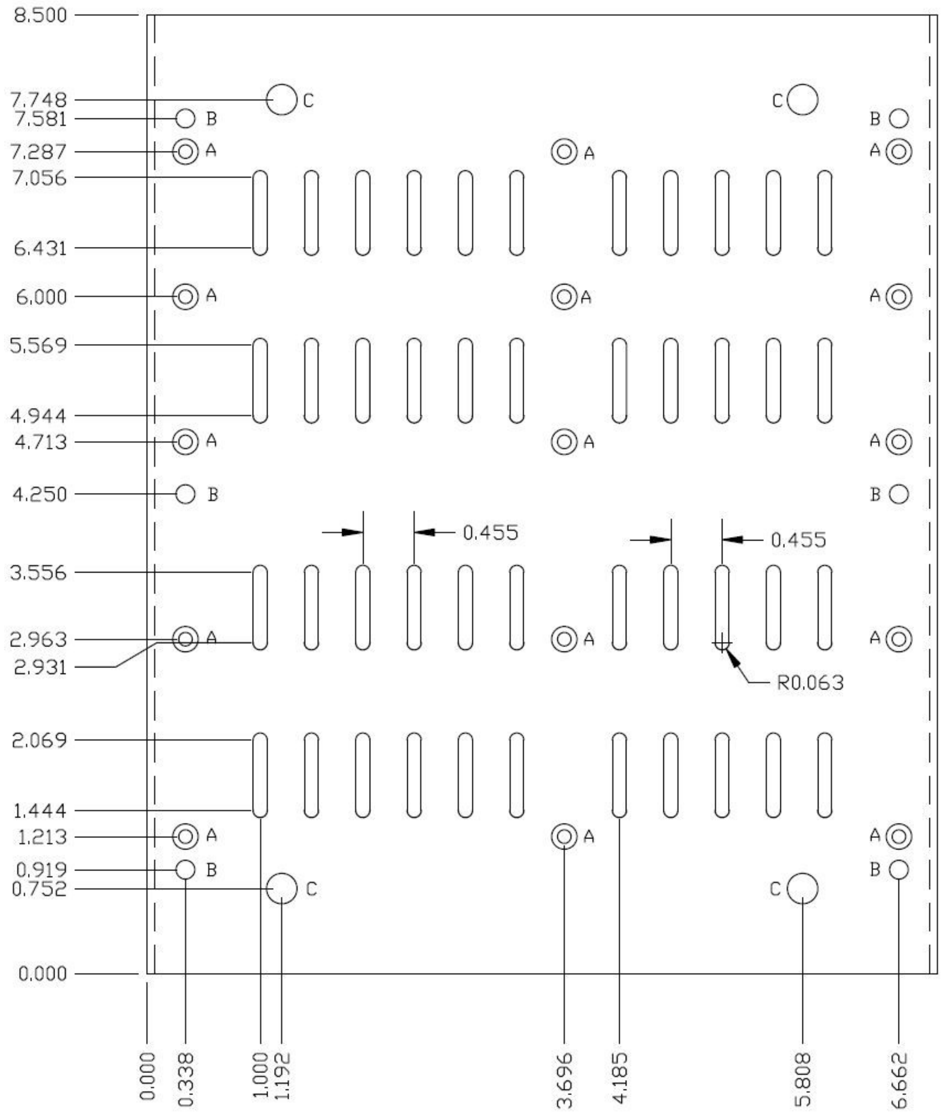

Figure 17. Dimensions and mounting of P620 Enclosure, bottom view

The unit is 2.25 inches tall, not including the rubber feet.

14 *Versions*

P620-1: 6-channel isolated resistance simulator

15 *Customization*

Consult factory for information about additional custom versions.

16 *Revision History*

16.1 *Hardware Revision History*

Revision C	September 2019 Changes for manufacturability. (23A620C)
Revision B	September 2018 Improved accuracy and on-board monitoring. (23A620B)
Revision A	October 2015 Initial PCB release (23A620A)

16.2 *Firmware Revision History*

The firmware is provided as a plug-in EEPROM chip which can be field upgraded.

Revision C	September 2019 Fixes a bug that caused HTTP server to improperly label its power supply voltages on its page “bist.html” (23E620C)
Revision B	September 2018 Support for revision B hardware changes (23E620B)
Revision A	April 2017 Initial firmware release (23E620A)

17 *Accessories*

J24-1: 24 volt 1.2 amp power supply (furnished with purchase)

J27-1: 2.1 x 5.5 mm barrel to pigtail power cable

J55-1: 6′ shielded D25 male to D25 male cable

J56-1: 10′ shielded D25 male to D25 male cable

J75-1: D9 female to two(2) dual banana plug cable

J475-1: 8-channel D25 female Din rail termination panel

P10-1: 19" rack mount shelf (two p-boxes per rack)

P51-1: mounting flange