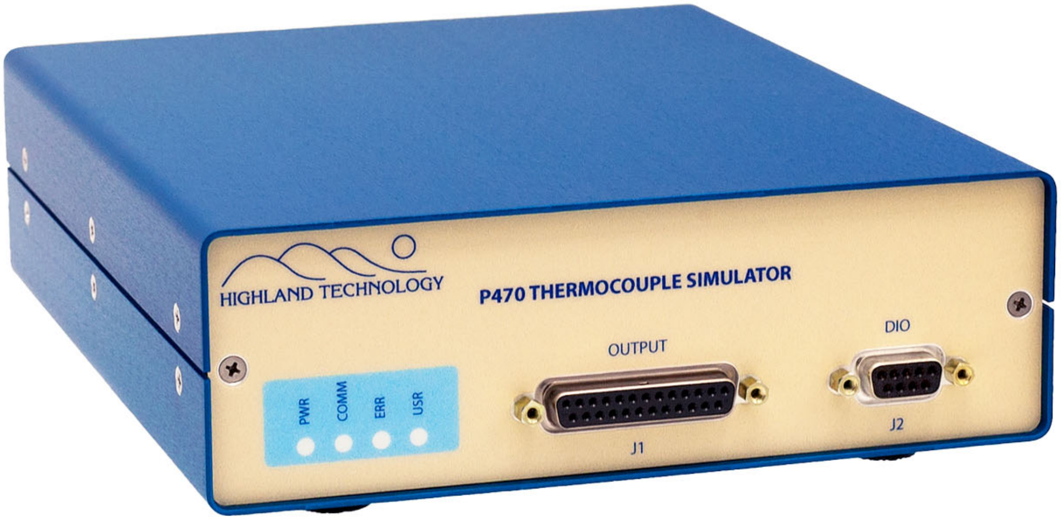
HIGHLAND TECHNOLOGY

# P470 8-CHANNEL BENCHTOP

# THERMOCOUPLE SIMULATOR



Technical Manual

April 26, 2022

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

Copyright © Highland Technology
650 Potrero Avenue, 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.

## Contents

# 1 *Introduction*

This is the technical manual for the Highland Model P470 thermocouple simulator.

Features of the P470 include:

- 8 fully isolated analog output channels, each independently programmable for mode and output range
- Voltage mode: provides 20-bit resolution with programmable output voltage range of ± 100 millivolts
- Thermocouple mode: simulates most common thermocouples (types J, K, E, T, R, S, B and N)
- NIST-standard lookup tables are included to allow direct entry of simulated temperatures in °C
- Two precision RTD signal conditioners for external reference junction temperature sensing, plus one on-board RTD reference junction sensor
- Any thermocouple simulation channel can be associated with any reference
- Provision for open and reverse thermocouple simulation
- Separate calibration port allows users to check calibration without moving connectors
- Compatible with optional J470 isothermal junction box with integrated RTD reference junction sensor or J475 DIN rail field wiring interface board
- 4 versatile digital input/output lines
- Full-function Built-In Self-Test (BIST)
- Ethernet and USB interfaces
- Controllable via serial ASCII commands or web interface

# 2 *Specifications*

## 2.1 *General Specifications*

| | |
|---|---|
| FUNCTION | 8-channel isolated thermocouple simulation module |
| RANGES | Programmable per channel<br>Bipolar voltage range: ± 100 mV<br>Thermocouples: Types J K E T R S B N |
| RESOLUTION | Voltage mode, 20 bits: 0.2 µV/LSB<br>Temperature simulation mode, 0.1°C |
| OUTPUT IMPEDANCE | 45 Ω max |
| OUTPUT CHANNEL PROTECTION | ± 350 volts peak (250 volts RMS) differential or common-mode, 5 KV ESD |
| RTD INPUTS | Two non-isolated thermocouple reference junction inputs for 100 ohm 4-wire PT 385 RTD |
| ALLOWABLE RTD TEMPERATURE RANGE | -40 to 120°C |
| RTD PROTECTION | Shorts to ground, 5 KV ESD |
| DIGITAL INPUT/OUTPUT | 4 digital I/O channels, TTL/CMOS compatible<br>32 V max input voltage<br>250 mA max sink current<br>Direct LED drive |
| ONBOARD SENSOR | 100R Platinum 4-wire RTD<br>± 0.5°C typical accuracy |
| OPERATING TEMPERATURE | 0 to 60°C |
| STORAGE TEMPERATURE | -40 to 70°C |

| | |
|---|---|
| CALIBRATION INTERVAL | One year |
| POWER | 24 VDC typical (23.5 to 25.5 V)<br>Nominal 200 mA<br>model J24, 24 VDC, 30 watt external adapter furnished |
| CONNECTORS | 1 D25 female for 8 channels and two RTDs<br>1 D9 male for test<br>1 D9 female for DIO<br>1 securable 2.1 x 5.5 mm power barrel connector with center pin positive |
| PACKAGING | 7.0" (L) x 8.5" (W) x 2.25" (H) Aluminum enclosure |
| INDICATORS | LEDs indicate power, communications and error<br>Additional LED is user programmable |
| COMMUNICATIONS | Micro-USB serial port emulator, 115.2 kbaud<br>10/100 Ethernet |
| CONFORMANCE | Thermocouple tables based on NIST/ITS-90<br>RTD tables per IEC-751 for 385 curve RTDs |

## 2.2 *Accuracy Specifications*

| | TYPICAL | LIMIT 15/35°C | LIMIT 0/60°C |
|---|---|---|---|
| Channel Output Accuracy | ± 5 µV Offset<br>± 50 ppm Gain | ± 35 µV Offset<br>± 500 ppm Gain | ± 70 µV Offset<br>± 1200 ppm Gain |
| RTD Measurement Accuracy | - | ± 500 ppm of RTD resistance | ± 1000 ppm of RTD resistance |

# 3 *Overview*

The P470 includes eight independent, isolated output channels. Each channel has an isolated DC power supply, isolated data interface, D/A converter, output amplifier, and a calibration bus switching relay.

Each channel may be user programmed to operate in either voltage output mode or thermocouple simulation mode. In each mode, channel operating parameters and output levels are programmed per-channel. User may write temperature or voltage values and the microprocessor will do all necessary calculations and update the channel electronics.

Two separate, non-isolated, 4-wire RTD signal conditioners are provided for external reference junction sensing. A third internal reference-junction temperature sensor is isothermal to the D25 output connector. Users may also declare a reference junction temperature.

The P470 includes lookup tables for common thermocouple types so that users may directly request temperatures in °C. In thermocouple simulation mode, any channel can be associated with any of the reference junction sensors. Cold-junction compensation is done via table lookup of thermocouple potential for the type currently selected. Reference junction temperatures are readable.

The front panel D25 connector is used to interface to external analog devices. The connector provides eight differential outputs and connections for two 4-wire RTDs.

A back panel male D9 test connector is provided for in-system calibration check. Each isolated channel incorporates a software-controlled relay which allows the channel output to be diverted to two pins of the test connector. Other pins of the test connector allow verification of the accuracy of the reference-junction RTD measurement subsystem.

BIST allows the invocation of a full closed-loop test of all 8 channels.

The front-panel female D9 connector provides four digital inputs/outputs. Each can be a TTL or switch input, and each can drive TTL loads or relay coils up to 32 volts, 250 mA.

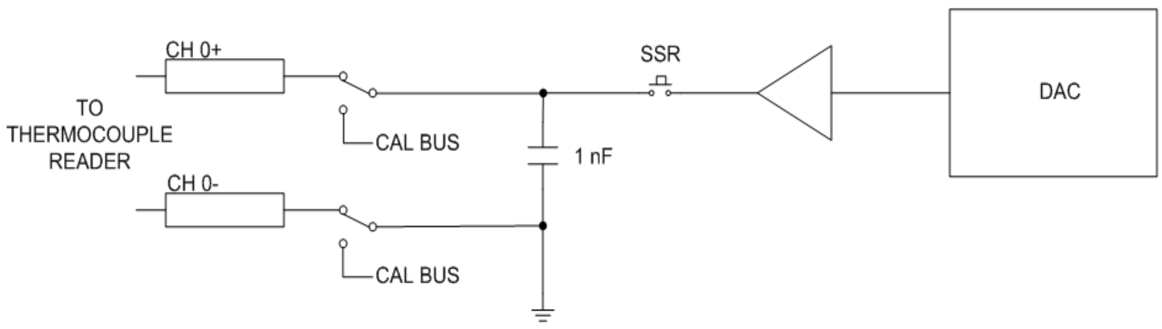


**Figure 1: P470 Output Channel**

# 4 *Connectors*

## 4.1 *D25 Output Connector*

One front-panel female D25 connector is provided to interface to external analog devices. The connector accommodates eight isolated differential outputs and two 4-wire RTD reference junction sensors.

The connector shell is bonded to the front panel, which connects to the aluminum enclosure through the front panel screws.

The pinout of the D25 connector is as follows:

| Pin number | Description | |
|---|---|---|
| 1 | ch 0+ | |
| 14 | ch 0- | |
| | | |
| 2 | ch 1+ | |
| 15 | ch 1- | |
| | | |
| 3 | ch 2+ | |
| 16 | ch 2- | |
| | | |
| 4 | ch 3+ | |
| 17 | ch 3- | |
| | | |
| 5 | ch 4+ | |
| 18 | ch 4- | |
| | | |
| 6 | ch 5+ | |
| 19 | ch 5- | |
| | | |
| 7 | ch 6+ | |
| 20 | ch 6- | |
| | | |
| 8 | ch 7+ | |
| 21 | ch 7- | |
| | | |
| 9 | RTD_A E+ | RTD excitation |
| 22 | RTD_A E- | |
| 10 | RTD_A S+ | RTD sense |
| 23 | RTD_A S- | |
| | | |
| 11 | RTD_B E+ | RTD excitation |
| 24 | RTD_B E- | |
| 12 | RTD_B S+ | RTD sense |

| Pin number | Description |
|---|---|
| 25 | RTD_B S- |
| 13 | GND |

## 4.2 *D9 Test Connector*

A male D9 Test connector is provided for connection to an external precision DVM. This makes it possible to externally verify the channel output and reference junction measurement accuracy.

The Pinout of the D9 test connector is as follows:

| Pin number | Description | |
|---|---|---|
| 7 | CAL bus+ | test resistor source+ |
| 6 | CAL bus- | test resistor source- |
| 9 | | test resistor sense+ |
| 8 | | test resistor sense- |
| | | |
| 1 | ground | |
| 2 | not used | |
| 3 | not used | |
| 4 | not used | |
| 5 | not used | |

## 4.3 *D9 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) |
| | |
| 1 | GND |
| 7 | Not used |
| 8 | Not used |
| 9 | GND |

As an input, each pin can accept a TTL input or switch closure to ground, with threshold about +1.6 volts. As an output, each pin can sink up to 32 volts at 250 mA. Inductive flyback protection is included. The 1K pullup resistor can drive small LEDs or SSRs against ground.

The equivalent circuit of each DIO pin is shown in Figure 2.

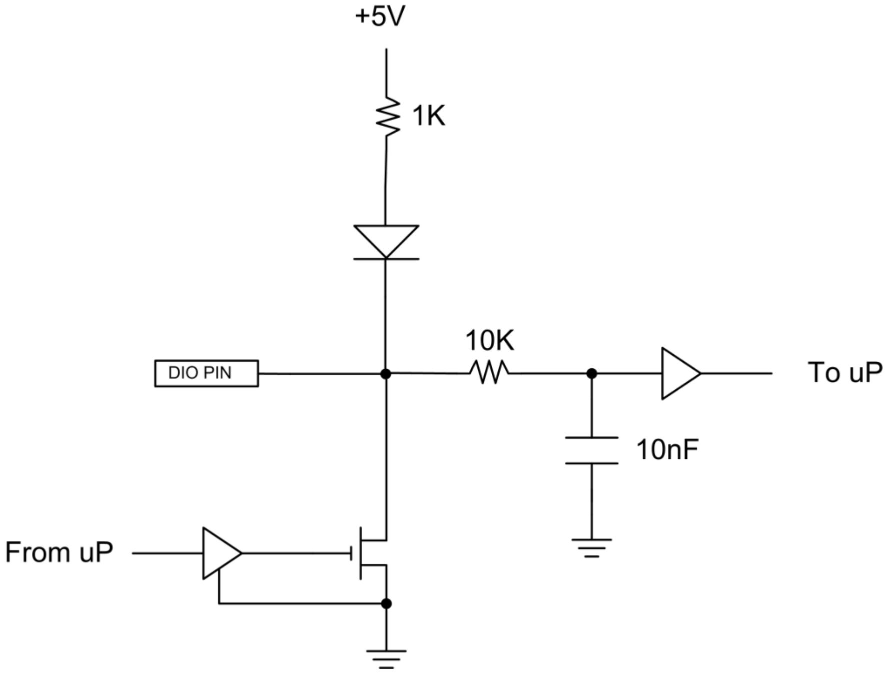


**Figure 2: DIO Circuit.**

## 4.4 *Barrel Power Connector*

The P470 requires +24 volt DC power. A Highland model J24 supply is available. The J24 includes a standard 2.1x5.5 mm DC barrel connector.

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 is bare leads.

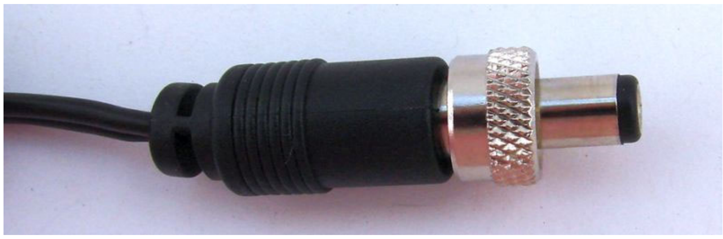


**Figure 3: J27 Power Connector**

The return (negative) side of the +24 power is grounded to the P470 enclosure, and the high side is protected by a polyfuse and a transzorb power zener.

## 4.5 *Ethernet*

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.

## 4.6 *USB*

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 6.3.

# 5 *Operation*

## 5.1 *Front-Panel LEDs*

There are 4 front-panel LED indicators: POWER, COMM, ERROR and USER.

The green POWER LED will blink once every two seconds to indicate that the internal firmware is operating normally.

The blue COMM LED will flash whenever the P470 receives a command over either the USB or Ethernet ports.

The red ERROR LED will normally be off, unless a failure is detected. Blink patterns are:

| | |
|---|---|
| One blink | Channel programming error |
| Two blinks | RTD acquisition error |
| Three blinks | RTD system self-check error |
| Four blinks | Calibration table error |
| Five blinks | Power supplies error |

The last three blink patterns indicate fatal hardware problems.

The orange USER LED may be programmed with the `USer` serial command.

## 5.2 *Power up Defaults*

The first time that a P470 it is powered up, the defaults for all 8 Output Channels are:

| Setting | Value |
|---|---|
| Thermocouple Type | K |
| Reference | Internal |
| Name | "" (None) |
| Output Impedance | Normal |
| Output Value | 100.000 °C |

The DIO will power up with all its channels as Inputs. Nothing is connected to the CAL BUS or the D9 Test Connector.

## 5.3 *Quick Start Procedure*

Basic Operation of the P470 can be demonstrated by the following steps:

Power on the unit by plugging the barrel connector of the power supply into the P470 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 P470 USB port. The computer should detect the P470 as a USB to Serial converter and automatically install all the necessary drivers. A new comport should be available in 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 6.3 and open the connection.

In the new terminal, type in the command

```
  USER 0XFF00
```

The front panel USR LED should begin to blink.

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

```
  SET 0 TYPE M
  VALUE 0 100
```

Connect a volt-meter to the D25 Connector. The High line should connect to pin 1; the low should connect to pin 14. The volt-meter should read 100 mV.

# 6 *Communication*

## 6.1 *TCP Interface*

By default the P470 IP address is set by DHCP. This may be changed using the `IPADD` and `SUBNET` commands, issued through the Ethernet or USB ports. Static addressing is supported.

The P470 can be accessed in TCP/Telnet mode, using standard ASCII commands. User software can open the P470 at its IP address, port 2000. Only one TCP session may connect to this port at a time. Port 80 is used for the web interface access.

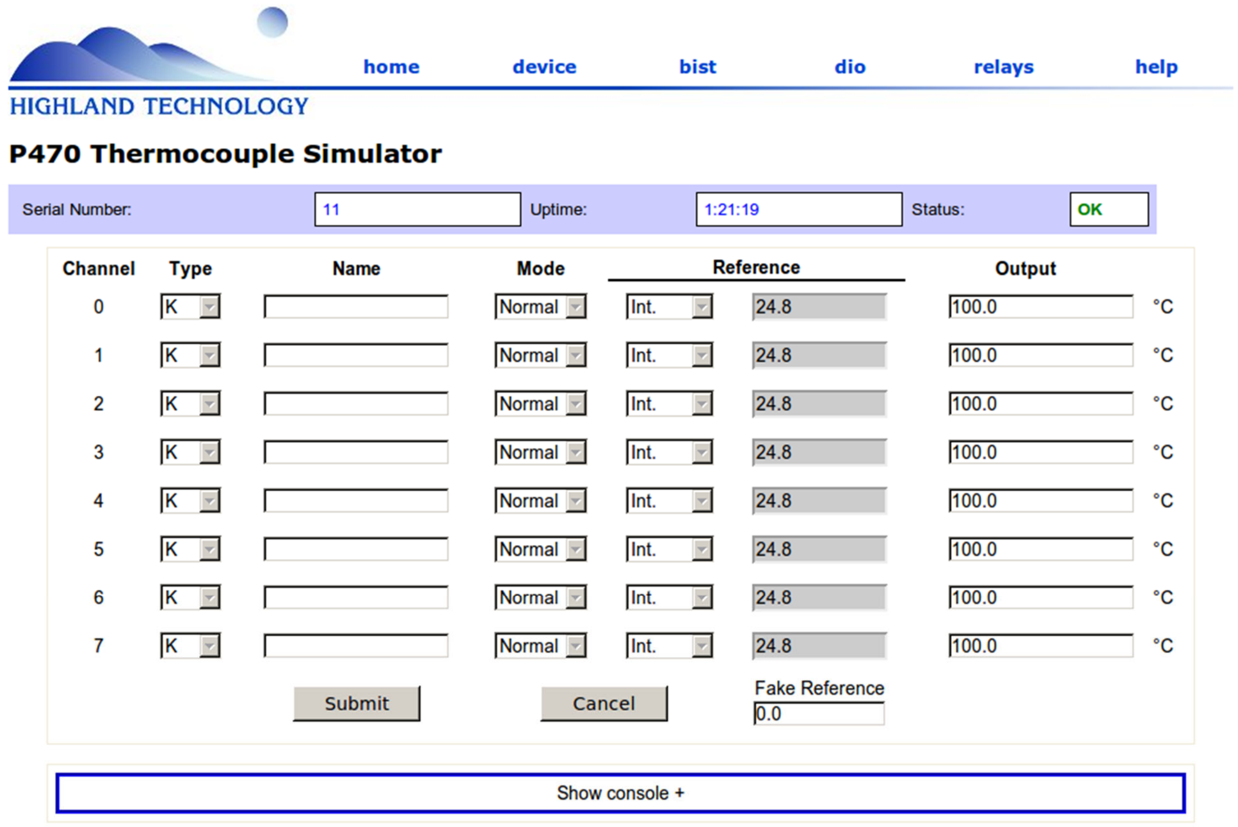
## 6.2 *Web Browser User Interface*

The P470 can be accessed from a web browser. Firefox or Chrome is recommended. Enter the IP address of the P470 in the browser address bar.

If the P470 is set up to operate in DHCP mode, it may also be accessed by its host name, which will be of the form:

P470-xxxxx

Where xxxxx is the 5-digit unit serial number. If the serial number is fewer than five digits long, pad it on the left with zeros.



P470 Thermocouple Simulator

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Serial Number: | 11 | Uptime: | 1:21:19 | Status: | OK |

| Channel | Type | Name | Mode | Reference |  | Output |  |
|---|---|---|---|---|---|---|---|
| 0 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 1 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 2 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 3 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 4 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 5 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 6 | K |  | Normal | Int. | 24.8 | 100.0 | °C |
| 7 | K |  | Normal | Int. | 24.8 | 100.0 | °C |

Fake Reference: 0.0

## 6.3 *USB Interface*

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

The P470 emulates a serial port using the FTDI FT230XS USB interface chip. This is a very common USB/serial interface chip, and many operating systems include appropriate drivers by default.

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 *ASCII Command Reference*

## 7.1 *General Comments*

The P470 accepts ASCII serial commands from the USB interface or from the 10/100 Ethernet port.

A family of HELP commands is available, summarizing serial commands and operating modes.

The P470 ignores serial input while it is processing the current command line.

In the following section, text using normal typewriter font, such as

```
SEt<cr>
```

represents a command string sent to the P470, terminated with a carriage return character. Italicized typewriter text such as

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

represents the reply from the P470. All commands must be terminated with a semicolon or <cr>; all reply lines are terminated by <cr><lf>.

“Query command” refers to a serial command that requests information about the P470’s state without changing it. “Set command” refers to a command that changes the P470’s state. (The specific SET command is always written in uppercase.)

### 7.1.1 *Command Strings*

Users send serial ASCII command strings to the P470, to which the P470 replies. Because the P470 may spend milliseconds to process commands, user software must wait for a response to each command line before sending another command.

Each command consists of a command keyword 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> character, the buffered line is executed in the order received.

All commands must be terminated by either an end of line indicator (carriage return, ASCII 13, denoted <cr>) or a semicolon separator (;). Linefeeds (ASCII 10, denoted <lf>) are ignored.

Keywords may be fully spelled out or may be sent as their first two letters; only the first two letters are significant. 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. The actual P470 protocol is case insensitive.

Examples:

| | |
|---|---|
| `VAlue` | Indicates that the short form is `VA`, and the long form |

is `VALUE`, both of which are recognized commands.

All forms are case insensitive. At least one space is required to separate command words from arguments.

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

| | |
|---|---|
| `VAlue 4 347.2` | Sets the channel 4 output to 347.2 °C |
| `VAlue 4` | Queries channel 4 output, which returns:<br>*347.2* |

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 P470 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 notation, not exponential or engineering notation. For example, use "0.123", not "123e-3" or "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.

### 7.1.2 *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 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 K; SET 4 TYPE J; SET 1 TYPE T
```

To which it would reply

```
OK; OK; OK
```

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 P470 will abort without performing the remaining commands on the line.

The errors are:

| Error message | Comments |
|---|---|

| Error message | Comments |
|---|---|
| *E01: Command not found* | The P470 did not recognize a keyword as valid. |
| *E02: Argument missing or invalid* | The P470 did not recognize an argument as valid for the keyword. |
| *E03: Invalid range* | A specific version of E02. |
| *E04: Hardware error* | Always considered a critical error. |
| *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 9, Security Provisions). |

## 7.2 *P470 Command Summary*

The following is a summary of ASCII commands which may be sent to the P470:

| LONG FORM | SHORT FORM | FUNCTION |
|---|---|---|
| SET | SE | set channel operating modes |
| GET | GE | return channel operating modes |
| VALUE | VA | set channel output, temperature or mV |
| STATUS | ST | return status reports |
| BIST | BI | control self-test |
| RELAYS | RE | control internal test relays |
| IPADD | IP | manage IP address |
| SUBNET | SU | manage subnet mask |
| SAVE ALL | SA AL | save all user settings in nonvolatile memory |
| SAVE SETUPS | SA SE | save only channel setups |
| SAVE VALUES | SA VA | save only channel output values |
| SAVE DIO | SA DI | save only the DIO output setting, 0 to 15 |
| SAVE IPADD | SA IP | save the IP address |
| 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 P470 to the default setup |
| DIO | DI | set/report digital i/o levels |
| IDENT | ID | identify unit hardware and firmware |
| USER | US | load "user" LED blink pattern |
| BOOT | BO | restart code |
| MAC | MA | query MAC address |
| FAKE | FA | set fake reference junction temperature |
| HELP | HE | return command summary |
| 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 a file line to flash |
| FLASH CHECKSUM | FL CH | run checksum check of firmware images |

## 7.3 *Command Details*

### 7.3.1 *Channel SET Command*

The `SEt` command establishes channel operating modes. Use the `SAve SEtups` command to preserve these after power up.

Usage is of the pattern:

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

<channel-list> is a list of channels without spaces (EG “234” for channels 2, 3, and 4), or “ALL” for all channels (equivalent to “01234567”).

<setting> is one of `TYpe`, `REf`, `NAme`, or `ZOut`, discussed in the sections below.

As with command words, setting names are case-insensitive, and only their first two letters are evaluated.

Typical SET commands might be:

| | |
|---|---|
| `SEt 234 TYpe K REf A` | Set channels 2, 3, and 4 to type K thermocouples using the external "A" RTD as the reference junction sensor. |
| `SEt ALl ZOut NOrm` | Set all channels’ output impedance to normal. |

The settings are discussed individually below.

### 7.3.2 *TYPE Setting*

Set the thermocouple type. Value for `TYpe` is one of the following:

| | |
|---|---|
| `J K E T R S B` or `N` | standard types |
| `M` | millivolt output |

For Example:

| | |
|---|---|
| `SEt 6 TYpe J` | Sets the thermocouple type of channel 6 to J |

### 7.3.3 *REF Setting*

Select the reference junction sensor to be used for the channel.

| | |
|---|---|
| `A` | use external RTD A |
| `B` | use external RTD B |
| `I` | use internal RTD |
| `Z` | assume 0°C reference temperature |

| | |
|---|---|
| `F` | use fake reference temperature; see `FAke` command |

For Example:

| | |
|---|---|
| `SEt 6 REf I` | Selects the internal reference junction sensor for channel 6 |

### 7.3.4 *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 "Pump 4 Inlet"` | Declares the name "Pump 4 Inlet" for channel 2 |

### 7.3.5 *ZOUT Setting*

Set the channel's output impedance. Valid values are `NOrm`, `OPen`, and `REv`. Only the first two letters are evaluated. They may be either uppercase or lowercase.

| | |
|---|---|
| `SEt 2 ZOut NOrm` | Sets output impedance normal on channel 2 |
| `SEt 2 ZOut OPen` | Simulates open thermocouple on channel 2 |
| `SEt 2 ZOut REv` | Simulates reversed thermocouple on channel 2 |

### 7.3.6 *Channel GET Command*

`GEt` is the query equivalent of `SEt`.

Usage is of the pattern:

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

| | |
|---|---|
| `GEt All` | Returns all channels' settings, delimited by semicolons for each channel |
| `GEt 5 Type` | Returns thermocouple type of channel 5 |

The reply is of the pattern:

```
CHANNEL <n> <setting> <value> [<setting> <value> […]] [; CHANNEL …]
```

For example, `GEt 12 NAme TYpe` might reply:

```
CHANNEL 1 NAME "Pump 4" TYPE J; CHANNEL 2 NAME "" TYPE M
```

The empty quotes (`""`) in this example indicate that the name for channel 2 has not been set by the user. This ensures consistency in the number of returned string tokens.

If an argument does not follow the channel-list argument, then all channel settings (TYPE, REF, NAME, and ZOUT) will be returned for each channel in the channel-list argument.

### 7.3.7 *Channel VALUE Command*

The `VAlue` command sets the real-time output of a channel, with the numeric argument being degrees C or millivolts, appropriate to the channel TYPE setting.

| | |
|---|---|
| `VAlue 6 235.7` | Sets channel 6 (already in thermocouple mode) to 235.7°C |
| `VAlue 2 -91.271` | Sets channel 2 (already in voltage mode) to -91.271 mV |
| `VAlue 5` | Queries the current setting of channel 5 |

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

For channels with `TYpe` set to be thermocouples, the value is limited to the range of -270 to +2000 °C, with resolution of 0.1 °C. If the programmed value exceeds the defined range of the selected thermocouple type, the simulated output will be clipped to the lower or upper limit of that thermocouple type and the red LED will make single flashes to indicate a channel programming error.

For channels that are declared to be millivolt output types, the value is limited to ± 100.000 mV, with resolution of 0.001 mV (1 µV). Values beyond this range will be clipped and invoke the red LED flash.

Replies to queries will print to a precision of 1 µV.

### 7.3.8 *DIO Command*

The `DIo` command controls the digital inputs/outputs on the D9 DIO connector. The numeric argument can be a decimal or hexadecimal number, which represents a bitfield of the 4 DIO channels. DIO 0 is the LSB.

`DIo 15` Turns all four pulldown drivers on, electrically low.

`DIo 2` Turns DIO 1 low/on. Others high/inputs.

`DIo` Queries DIO states, returning set levels and actual levels, as decimal values. The sequence:

| | |
|---|---|
| `DIo 15` | Returns *OK* |
| `DIo` | Returns *15 0* |

Since 15 sets all DIO drivers ON, actual input levels are all low.

To use all DIOs as inputs, first send `DIO 0`. Then the query `DIO` will return *`0 X`* where X is the decimal encoded logic levels of the four inputs. X would be 8 if only DIO3 was low.

### 7.3.9 *IDENT Command*

The `IDent` command returns an ID string of the form

```
P470-1A SN 1234 FIRMWARE xxxxx IP xxx.xxx.xxx.xxx MAC xx:xx:xx:xx:xx:xx
```

The *`-1A`* following *`P470`* means dash 1, hardware revision A. Firmware will usually be 23E470 followed by the firmware revision letter, for example `23E470A`.

### 7.3.10 *USER Command*

The `USer` command loads a blink pattern for the orange USER LED. This LED is useful in identifying multiple P-series instruments in the field. The argument is a 16-bit-wide integer, whose bits are shifted at 8 Hz. It may be either decimal or hexadecimal (“0x”) notation.

| | |
|---|---|
| `USer 0` | Turns the LED off |
| `USer 0xFFFF` | Turns the LED full on |
| `USer 0xF0F0` | Blink continuously at 1 Hz |

### 7.3.11 *IPADD Command*

The `IPadd` command sets the Ethernet IP address.

| | |
|---|---|
| `IPadd 192.168.254.183` | Sets a static IP address |
| `IPadd 0.0.0.0` | Sets DHCP mode |
| `IPadd DHCP` | Sets DHCP mode |
| `IPadd` | With no argument is a query |

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.

### 7.3.12 *SUBNET Command*

The `SUbnet` command sets the Ethernet subnet mask.

| | |
|---|---|
| `SUbnet 255.255.255.0` | Sets subnet mask |
| `SUbnet` | With no argument is a query. |

### 7.3.13 *MAC Command*

The query-only `MAc` command returns the factory-set MAC address in colon-delimited notation.

### 7.3.14 *BOOT Command*

The `BOot` command reboots the P470 software, simulating a power up. All SAVED items will be restored.

### 7.3.15 *SAVE and LOAD Commands*

`SAve` commands save current settings in nonvolatile memory. A subsequent power up or a `LOad` command restores those values.

| | |
|---|---|
| `SAve ALl` | Saves everything |
| `SAve SEtups` | Saves only channel setups |
| `SAve VAlues` | Saves only channel output values |
| `SAve DIo` | Saves the DIO output setting, 0 to 15 |
| `SAve IPadd` | Saves the IP address |

| | |
|---|---|
| `LOad ALl` | Restores all saved items |
| `LOad SEtups` | Restores saved channel setups |
| `LOad VAlues` | Restores saved channel values |
| `LOad DIo` | Restores saved DIO output state |
| `LOad IPadd` | Restores saved IP address |

At power up, all saved items are restored, equivalent to `LOad ALl`.

| | |
|---|---|
| `LOad DEfaults` | Sets the P470 to a default setup. The defaults are: |

- All outputs type K thermocouple set to 100 °C.

- All use the internal reference junction sensor
- The IP address is set to DHCP.
- FAKE is set to 0 °C.

### 7.3.16 *RELAYS Command*

**The `RElays` command controls test relays internal to the P470. Refer to**

Figure 5 for a summary of signal routing.

`RElays OFf`

Turns off all test relays

`RElays Kn`

Turns on Kn, where n is a number from 0-9. For n= 0 through 7, this connects an output channel to the internal BIST bus. All other channels are disconnected.

For n = 8, this connects the BIST bus to pins 7 and 6 of the D9 test connector. This action turns off all other relays.

For n = 9, this establishes a 4-wire connection from the 100 ohm test resistor to the D9 test connector. This action turns off all other relays.

To set any combination of a channel relay, K8, and K9, multiple arguments may be listed.

For example, the command `RElays K8 K5` will connect channel 5 output to pins 7 and 6 of the D9 test connector.

### 7.3.17 *BIST Command*

The `BIst` command controls the Built in Self-Test of the P470. `BIst GO` runs a full self-test and returns an *`OK`* statement. The user can verify test completion with the `BIst STatus` command.

BIST tests internal power supplies, all eight channel outputs, and the RTD measurement subsystem. This is a functional test and cannot verify P470 calibration, which would require external NIST traceable references.

Control over the relays of the P470 will not be available until completion of BIST. If a `RElays` command or any other command that requires writing relays, such as certain `LOad` commands, is sent, the request will be saved, *"`OK`"* will be replied, and the relays will be set to the new value when BIST is completed.

The `BIst` command can be followed by an argument; a single-line response will follow with exception of `BIst REport`. Arguments are case-insensitive, and only their first two characters will be evaluated.

`BIst GO` Runs a complete Self-Test. Replies *OK*.

`BIst STatus` Returns the status of BIST. Replies are: *NOT RUN* BIST has never run. *RUNNING* BIST is in operation. *PASS* all BIST tests were successful. *FAIL* one or more tests of BIST were not successful.

`BIst CAlr` Returns the last recorded value of the 100 Ω calibration test resistor in ohms.

`BIst BUs` Returns the last recorded value of the calibration bus in millivolts. This value is refreshed approximately every 1.32 seconds. It is necessary to wait longer than this when putting a channel in the Cal Bus and reading its value with the `BIst BUs` command.

`BIst POwer` Returns the voltage of each power supply; 5V, 3.3V, 1.2V and 24V; with an "*OK*" or "*ERROR*" statement. The reply might look like:

*"4.998 OK 3.280 OK 1.201 OK 23.999 OK"*

`BIst Report` Returns a report of BIST tests. Channel voltages and Isolation Fault Detect measurements are in millivolts, power supplies are reported in volts, and RTD measurement is reported in ohms. The report might look like:

```
BIST REPORT
CH0      -100mV  =  -100.000   OK
CH0      0mV     =  0.000      OK
CH0      +100mV  =  100.000    OK
CH0      ISOF    =  0          OK
CH1      -100mV  =  -100.000   OK
CH1      0mV     =  0.000      OK
CH1      +100mV  =  100.000    OK
CH1      ISOF    =  0          OK
CH2      -100mV  =  -100.000   OK
CH2      0mV     =  0.000      OK
CH2      +100mV  =  100.000    OK
CH2      ISOF    =  0          OK
CH3      -100mV  =  -100.000   OK
CH3      0mV     =  0.000      OK
```

```
CH3     +100mV  = 100.000    OK
CH3     ISOF    = 0          OK
CH4     -100mV  = -100.000   OK
CH4     0mV     = 0.000      OK
CH4     +100mV  = 100.000    OK
CH4     ISOF    = 0          OK
CH5     -100mV  = -100.000   OK
CH5     0mV     = 0.000      OK
CH5     +100mV  = 100.000    OK
CH5     ISOF    = 0          OK
CH6     -100mV  = -100.000   OK
CH6     0mV     = 0.000      OK
CH6     +100mV  = 100.000    OK
CH6     ISOF    = 0          OK
CH7     -100mV  = -100.000   OK
CH7     0mV     = 0.000      OK
CH7     +100mV  = 100.000    OK
CH7     ISOF    = 0          OK
5V              = 4.998      OK
3.3V            = 3.280      OK
1.2V            = 1.201      OK
24V             = 23999      OK
RES100          = 100        OK
BIST            = PASS
```

### 7.3.18 *HELP Command*

The `HElp` command returns documentation about the serial command set. With no argument, `HELP` returns a summary of available commands. The user can type `HELP` <command> for details about specific commands.

### 7.3.19 *STATUS Command*

The `STatus` command returns a summary report of the state of the P470. Report looks like this:

```
P470-1A  SN 0012   Firmware 23E470A1   Uptime 268
Hostname “P470-0012”
MAC  AB:CD:EF:01:02:03   IP  192.168.254.183 (STATIC)
Cal Date: 11/04/2015 OK
CHAN  0  TYPE  K  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  -180.500
CHAN  1  TYPE  J  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  10.000
CHAN  2  TYPE  E  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  740.000
CHAN  3  TYPE  E  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  100.560
CHAN  4  TYPE  M  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  -80.654
CHAN  5  TYPE  J  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  -30.500
CHAN  6  TYPE  K  VALUE  REF  I  NAME  “ ”  ZOUT  NORM  VALUE  1000.600
CHAN  7  TYPE  E  VALUE  REF  I  NAME  " "   ZOUT  NORM  VALUE  500.250
BIST  PASS
```

If the IP address is configured for DHCP, the IP address in the status report will show the IP as set by the DHCP server (0.0.0.0 if the server has not provided an IP yet), followed by *(DHCP)* instead of *(STATIC)*.

The calibration date as shown in the status should always be followed by *OK*. If calibration date is followed by *DEFAULT* or *DIRTY,* something is wrong with the CAL table and the P470 should be sent for recalibration.

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.

`STatus DIp`

Shows state of DIP switches. Reply is a number 0 to 15, representing a bitfield of the four DIP switches. A reply of *15* means "all on". A reply of *0* means "all off". Switch 1 is the LSB.

A reply of *8* indicates that the calibration table may not be written (switch 4 on) but the user settings may be written (switch 1 off). See section 9, Security Provisions.

`STatus IMage`

Shows which firmware image is currently loaded (see section 11, Boot Flow and Firmware Upgrade). Reply is either *FACTORY* or *UPGRADE*.

`STatus UPtime`

Shows number of seconds of uptime.

`STatus POwer`

Shows the voltage of each power supply: 5V, 3.3V, 1.2V and 24V. The reply might look like:

*"4.998 3.280 1.201 23.999"*

`STatus RTd X`

Reads one of the A, B or I (internal) RTDs. Reply is the measured resistance and the temperature in a single line:

*"R: 108.829, T: 22.663"*

### 7.3.20 *FAKE Command*

The `FAke` command sets or queries the internal FAKE temperature parameter. Any thermocouple simulation channel can be set to use the FAKE temperature as its reference junction temperature. The allowed range is -40 to 120 °C.

`FAke 52.5`

Sets the FAKE value to 52.5 °C

| | |
|---|---|
| `FAke` | Queries the value. |

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

### 7.3.21 *EXIT Command*

The `EXit` command, whether executed from the USB interface or the TCP interface, will close any open TCP session.  To ensure that a TCP session ends cleanly, use this command before closing the session on the client side.

### 7.3.22 *FLASH UNLOCK Command*

The `FLash UNlock` command enables firmware upgrade operations.  The DIP switches must be configured to permit writes to nonvolatile memory (see section 9, Security Provisions).

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.

### 7.3.23 *FLASH ERASE Command*

The `FLash ERase` command erases the upgrade portion of the boot flash (see section 11, Boot Flow and Firmware Upgrade).  The FLASH UNLOCK command must have been called first, and the DIP switches must be configured to permit writes to nonvolatile memory (see section 9, Security Provisions).

This command may take up to thirty seconds to execute.

| | |
|---|---|
| `FLash ERase` | replies *`OK`* when finished. |
| `FLash ERase VErbose` | replies a line of the form:<br>*`ERASING SECTOR i OF n`*`<cr><lf>`<br>for every flash sector being erased. |

### 7.3.24 *FLASH WRITE Command*

The `FLash WRite` command writes a line of an S-Record File, to the upgrade portion of the boot flash (see section 11, Boot Flow and Firmware Upgrade).  The `FLash UNlock` command must have been called first, and the DIP switches must be configured to permit writes to nonvolatile memory (see section 9, Security Provisions).

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.

A typical S-Record line is less than 80 characters. A command might look like:

```
  FLash WRite S22410010018F09FE518F09FE<remainder of line...>
```

### 7.3.25 *FLASH CHECKSUM Command*

The `FLash Checksum` command performs a checksum of the factory, and upgrades firmware images in the flash and reports their status (see section 11, Boot Flow and Firmware Upgrade). This may take up to five seconds.

The reply is of the form:

```
  FF=<result>,FH=<result>,UF=<result>,UH=<result>
```

*FF* is for “factory firmware”. *FH* is for “factory html” (the Web pages, see section 6.1, TCP Interface

By default the P470 IP address is set by DHCP. This may be changed using the `IPADD` and `SUBNET` commands, issued through the Ethernet or USB ports. Static addressing is supported.

The P470 can be accessed in TCP/Telnet mode, using standard ASCII commands. User software can open the P470 at its IP address, port 2000. Only one TCP session may connect to this port at a time. Port 80 is used for the web interface access.

Web ). *UF* is for “upgrade firmware”. *UH* is for “upgrade html”.

*<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 be either NP (if never upgraded) or both *OK*. The expected default result for a new P470 is:

```
  FF=OK, FH=OK, UF=NP, UH=NP
```

# 8 *Built-in Self-Test (BIST)*

The BIST facility allows users to verify functionality and accuracy of the P470 with a high degree of confidence. Note that BIST does not provide absolute verification of module accuracy, as external NIST-traceable standards are required for formal calibration.

Automatic 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, RTD subsystem, internal reference temperature sensor and channel isolation are checked. Each of the 8 channels are tested using three points of their full output range (+/-100mv).

Total test time is about 30 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.
- Failure of certain relay and RTD multiplexer paths.

# 9 *Security Provisions*

There may be cases where it is desirable to disallow any changes to the nonvolatile memory internal to the P470. 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 top cover can be sealed with tamper-evident stickers if desired.

The switch looks like:

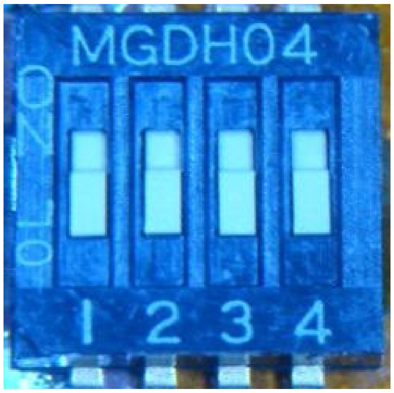


**Figure 4: 4 position DIP switch**

All four switches are shown OFF in this picture.

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

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

The factory default has switches 1-4 OFF.

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

# 10 *Calibration Verification*

The calibration of one or more P470 modules may be checked using the D9 test connectors. Multiple Highland P-series units may be bussed to a common test cable and test instruments.

The P470 module or modules may be connected to a precision digital voltmeter, such as a Fluke 8845A or equivalent. The required connections are:

| D9 PIN | DVM CONNECTION | FUNCTION |
|---|---|---|
| 7 | INPUT HI | Channel output (+), test resistor source (+) |
| 6 | INPUT LO | Channel output (-), test resistor source (-) |
| 9 | SENSE HI | Test resistor sense (+) |
| 8 | SENSE LO | Test resistor sense (-) |

To verify a channel output, use the `RElays` command to route a channel out through the D9 test connector into the DVM voltage input leads, pins 7 and 6. Then program the channel and verify accuracy using DVM measurements. Channels can be programmed as millivolt outputs, or can be programmed to simulate thermocouples, perhaps using the ZERO or FAKE reference junction options.

The serial commands to monitor the channel 3 output would be

| | |
|---|---|
| `RElays K8` | routes CAL bus to D9 connector |
| `RElays K3` | connects channel 3 to the CAL bus |

Send `RElays OFf` when done.

To verify the accuracy of the RTD measurement subsystem, operate the module normally and note the measured value of the internal test resistor, nominally 100 ohms, as displayed in the status report. Then send the serial command `RElays K9` and use the DVM to make a 4-wire resistance measurement of that same resistor. The reading of the DVM should be 100 ohms ± 0.25% and should agree with the status report value to within ± 0.5%. To return to normal operation send the command `RElays OFf`.

The D9 connector pins of multiple P470 modules may be bussed in parallel to a single precision DVM, provided that only one box is operating in calibration verification mode at any one time.

Each output channel incorporates a relay (K0 through K7) which allows it to be switched, using the BIST command, to the CAL bus. The BIST command can also be used to connect the D9 connector to either the CAL bus (through K8) or the RTD subsystem check resistor (through K9).

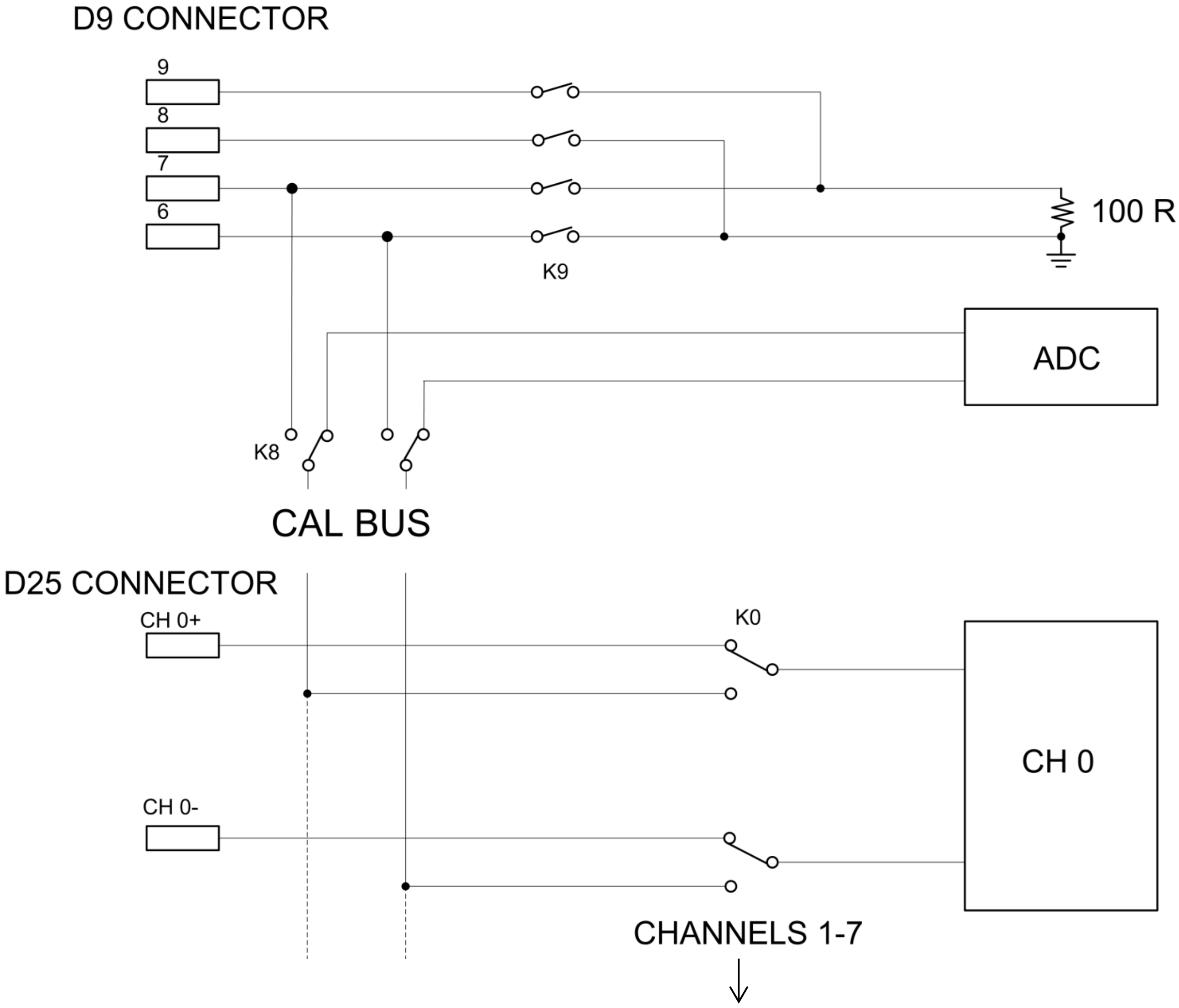


**Figure 5: P470 BIST connections**

# 11 *Boot Flow and Firmware Upgrade*

The P470 contains two major address spaces in its boot ROM (the “flash”): one for the factory image, and one for an upgrade image

The P470’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 P470 will boot into it. Otherwise the P470 will boot into the factory fallback image.

If its DIP switches are configured to permit writes to nonvolatile memory (see section 9, Security Provisions), the P470 may have its firmware upgraded by using the FLASH commands. The P470’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 P470, to enable modifying the flash. Wait for the reply *OK*.

2. Send `FLash ERase` to the P470. 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 P470.

    Wait for the reply *OK*

4. Send `FLash CHecksum` to the P470. 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 P470 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.

New P470 units ship with only a factory firmware image installed.

## 12 *Termination Boxes*

The J470 is an enclosed 8-channel isothermal termination box. It may be located up to 300 feet from the P470 and can be connected using standard D25 cable assemblies.

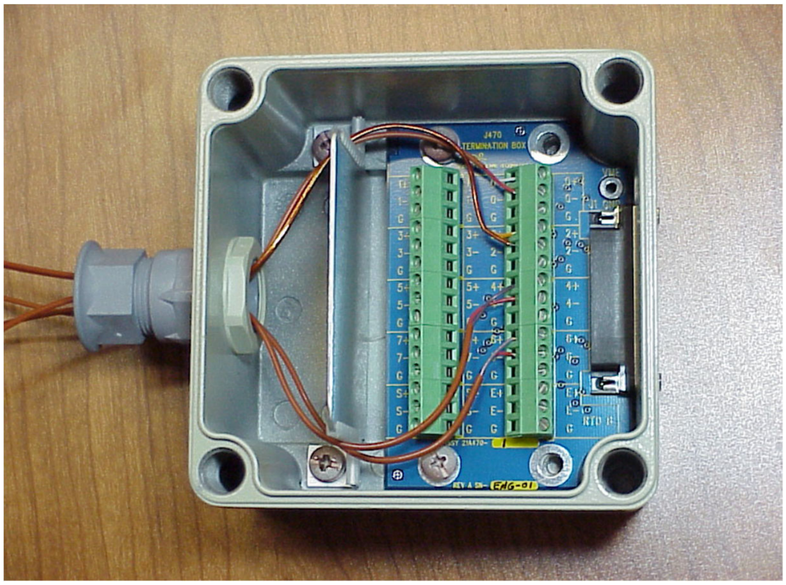


**Figure 6: J470 8-channel isothermal termination box.**

The J475-2 is an 8-channel DIN rail compatible PC-board termination panel that transitions from a D25 connector to screw terminals.

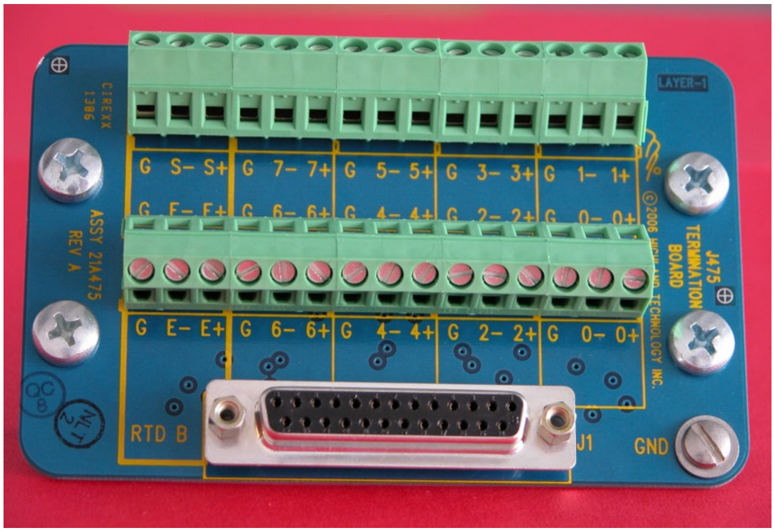

CIREXX
1366
LAYER-1
G S- S+ G 7- 7+ G 5- 5+ G 3- 3+ G 1- 1+
G E- E+ G 6- 6+ G 4- 4+ G 2- 2+ G 0- 0+
ASSY 21A475
REV A
J475
TERMINATION
BOARD
G E- E+ G 6- 6+ G 4- 4+ G 2- 2+ G 0- 0+
RTD B
J1
GND

**Figure 7: J475-2 8-channel DIN rail mount termination board.**

Both units include an on-board reference junction RTD A and terminals to connect an additional external RTD B.

# 13 *Packaging and Dimensions*

The P470 has mounting holes on the bottom in case you want to secure it. They are labeled “B” in Figure 8, and a #4-40 screw can be used. A mounting flange, model P51 is available. If using the mounting flange to secure the P470, Highland provides all screws necessary. All dimensions are in inches.

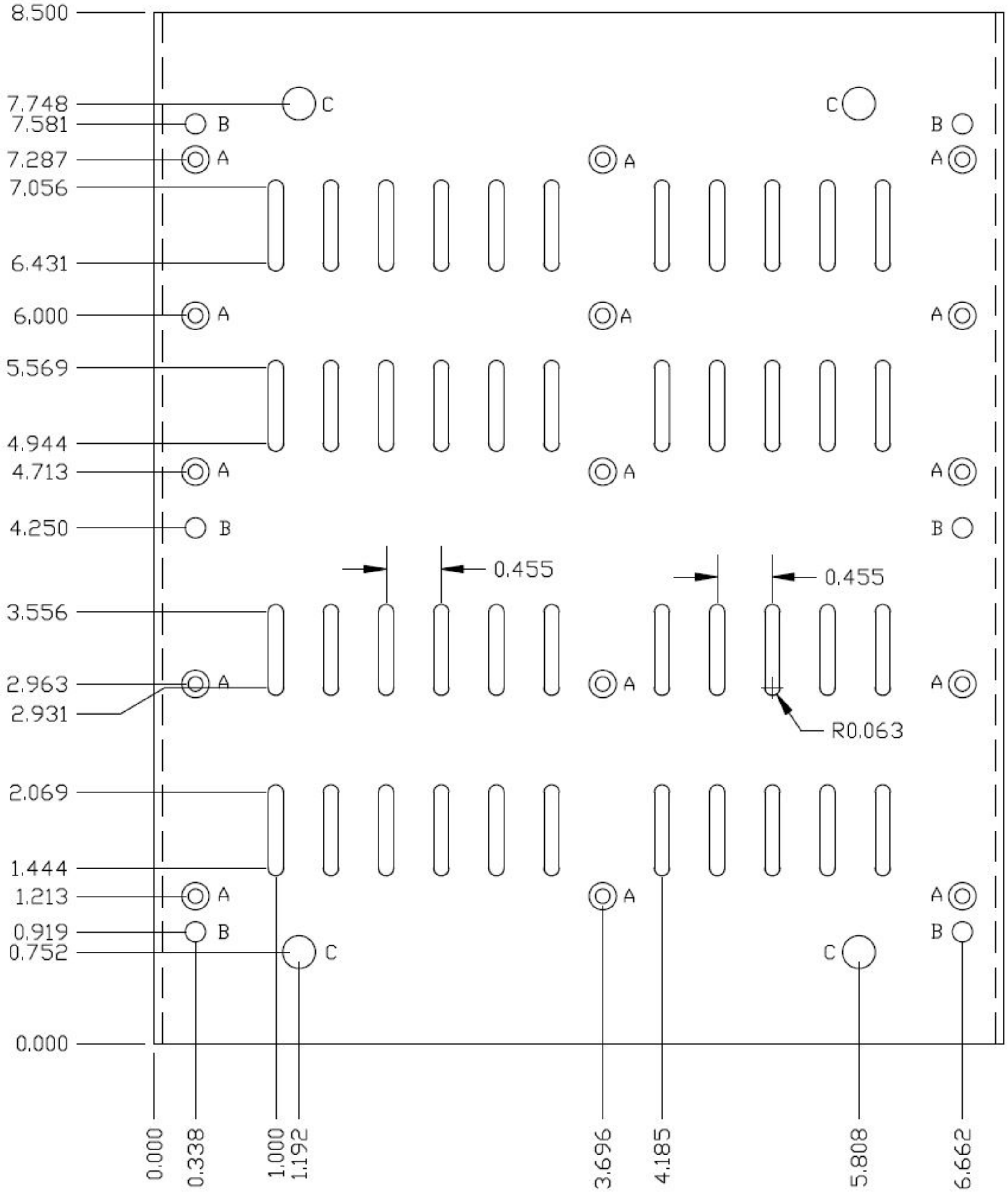


**Figure 8: Dimensions of P470 Enclosure, Bottom view.**

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

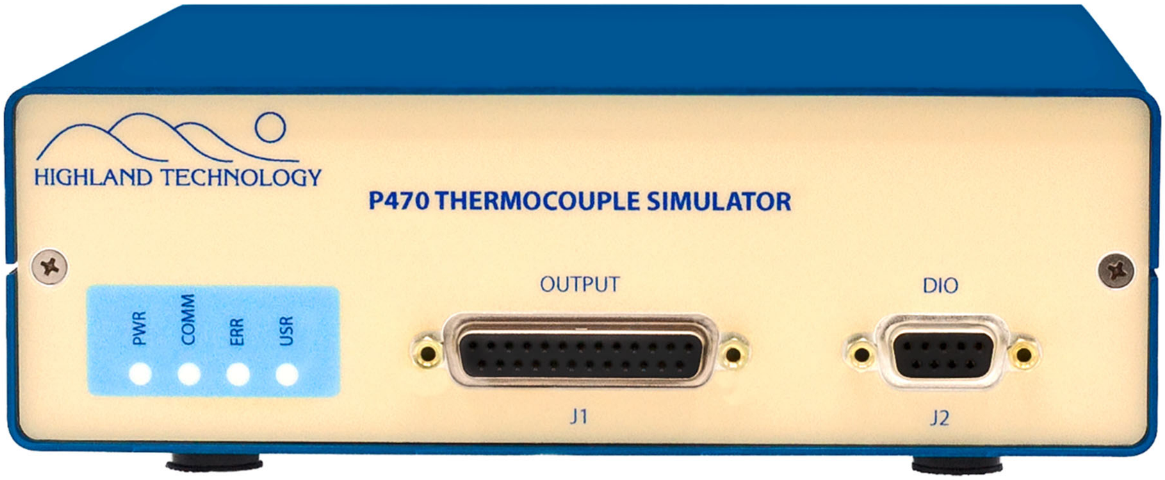


**Figure 9: P470 Front Panel**

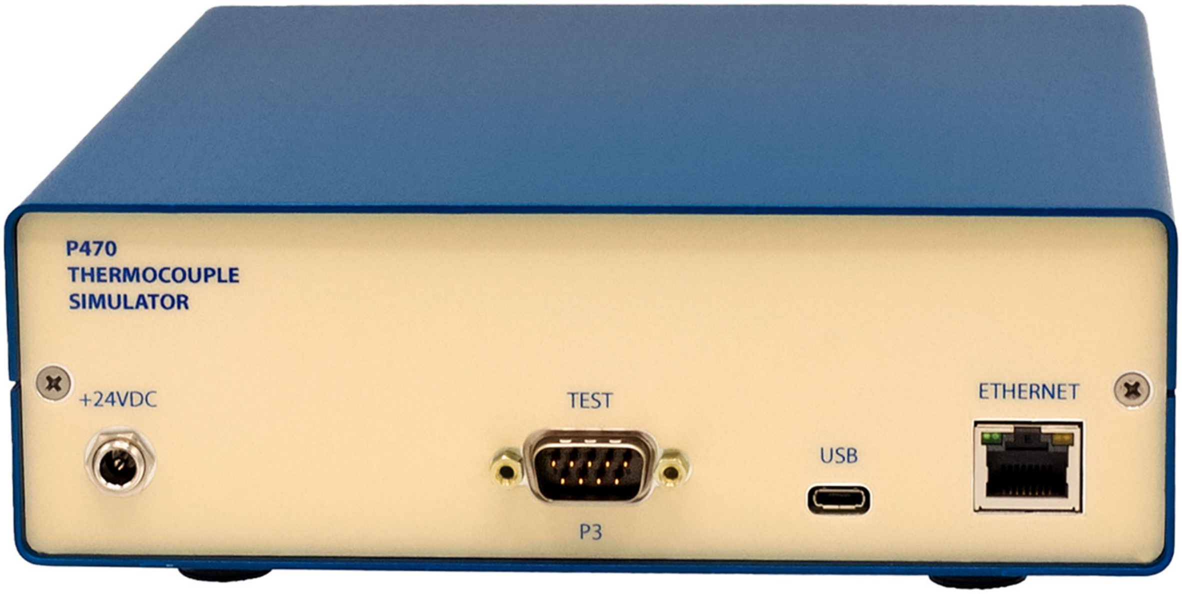


**Figure 10: P470 Back Panel**

# 14 *Thermocouple Theory and Notes*

A thermocouple is a pair of dissimilar-alloy wires used to measure temperature. Any metallic alloy can be thought of as having an intrinsic energy level or voltage that is a nonlinear function of temperature. This voltage is usually in the range of about 5 to 40 microvolts per degree C.

Figure 11 below illustrates a Type K thermocouple. The chromel and alumel alloys have different thermocouple voltage versus temperature functions, so a voltage appears at the ends of the leads that depends on the temperature gradients along the wires.

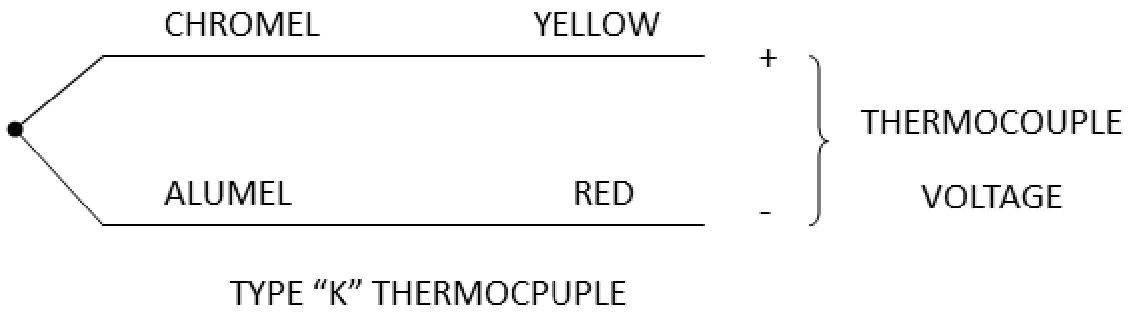


**Figure 11: Thermocouple type "K"**

If we want to measure the junction temperature, we connect a voltmeter to the ends of the thermocouple leads.

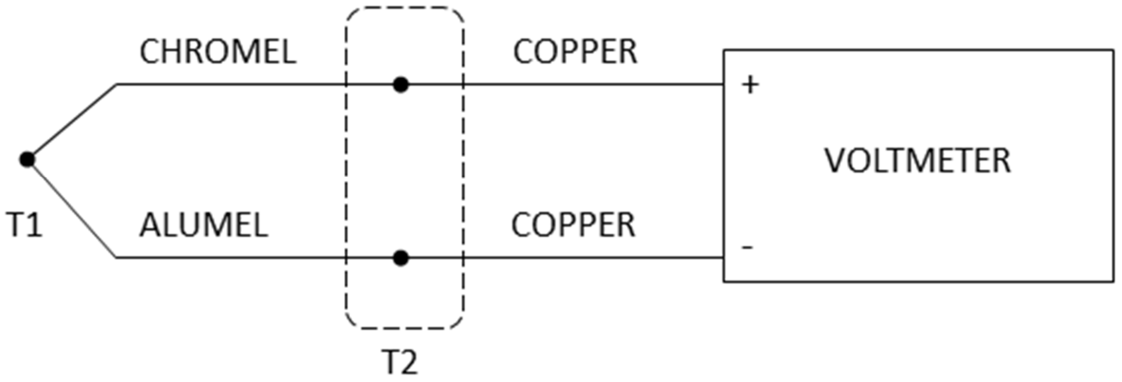


**Figure 12: Junction temperature**

In Figure 12, T1 is the temperature to be measured and T2 is the reference junction temperature. NIST publishes thermocouple temperature/voltage tables which assume that the reference junction is kept at ice point, 0°C.

If the reference junction cannot be maintained at 0°C, its temperature can be measured and a correction factor applied to the indicated voltmeter reading, so the ice-point-based lookup tables can then be used. Note that, because the thermocouple effect is nonlinear, the voltmeter reading is not an accurate function of the temperature difference between T1 and T2: it is more correctly a function of the difference in thermocouple potentials at T1 and T2.

Many thermocouple instruments apply a linear correction to the measured voltage, as a function of reference junction temperature, but this is only an approximation. Ideal thermocouple reference junction compensation corrects the voltmeter reading by the actual, nonlinear thermocouple potential at point T2, which is the technique used in the P470.

A thermocouple simulator, like the P470, is configured as shown in Figure 13.

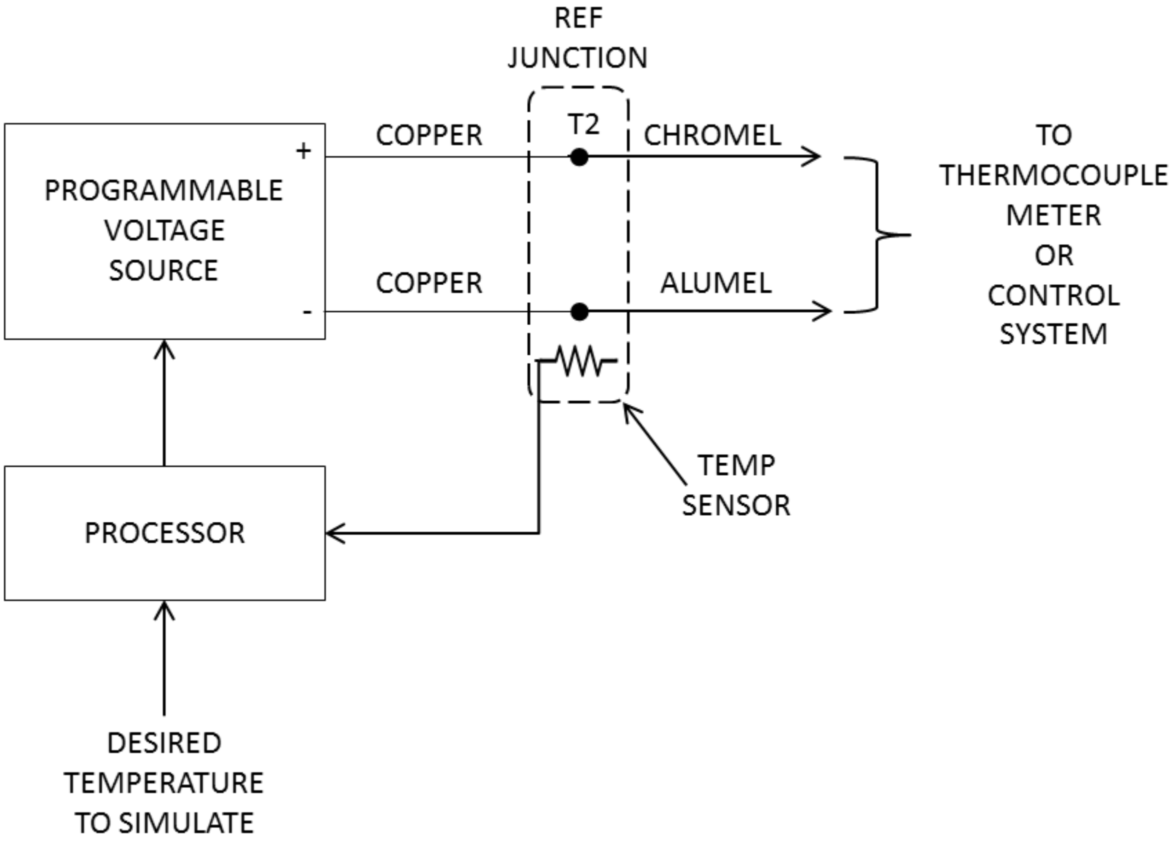


**Figure 13: P470 Configuration**

Here, a microprocessor is given a thermocouple type and a temperature to simulate. It looks up the proper microvolt level for the specified temperature, and also reads the temperature sensor at the reference junction temperature T2 and applies the appropriate nonlinear offset voltage to compensate for reference junction temperature. Within the reference junction box are screw terminal where the transition is made from copper-copper leads to the actual thermocouple alloys. It is important that both screw terminals and the temperature sensor be at the same temperature for the compensation to be accurate.

Note that the device being driven, a thermocouple meter or control system, generally has its own reference junction and associated compensation.

The P470 includes the capability to measure up to two platinum RTD temperature sensors to perform the reference junction math. Any of the 8 output channels may be assigned to use any one desired reference junction sensor. The module assumes that the RTDs are calibrated to 100 at zero degrees C, and that they follow the standard ISO 385 nonlinearity curve. RTDs are connected as 4-wire devices to eliminate errors caused by the resistance of connection wires from the P470 to the reference junction.

A typical P470 channel setup is shown in Figure 14.

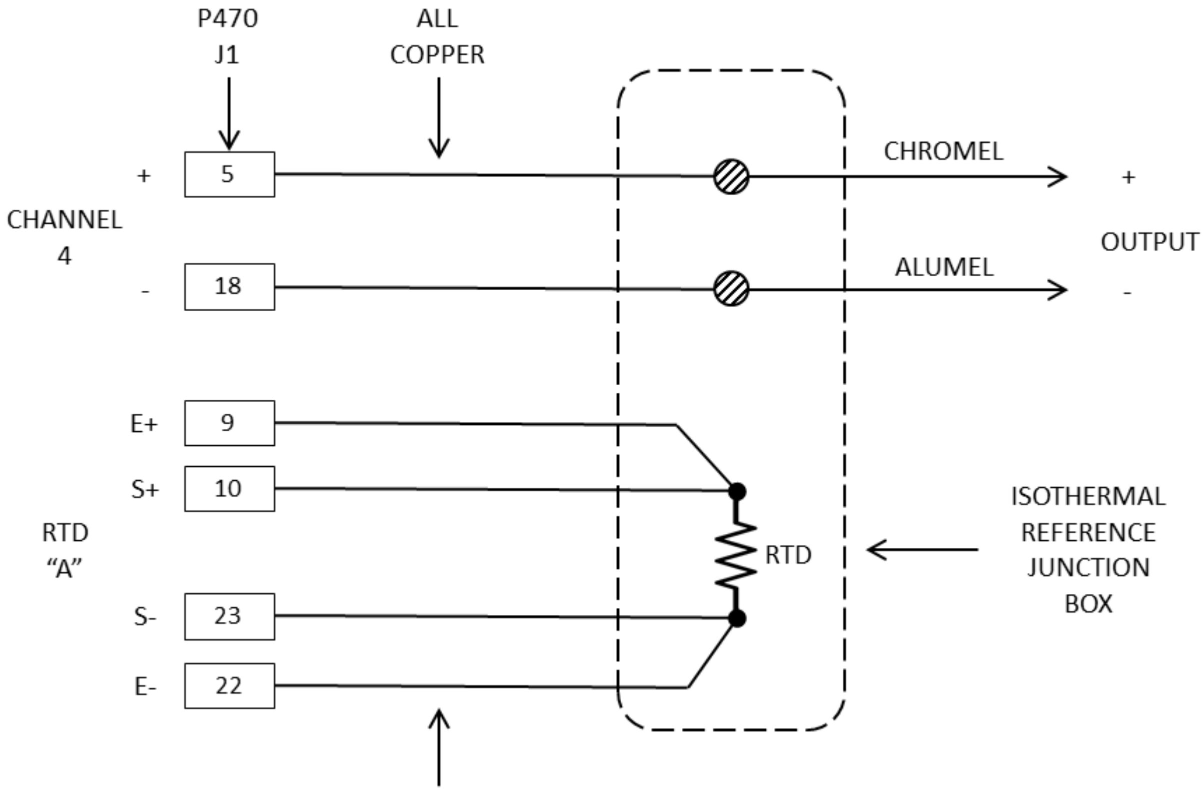


**Figure 14: P470 channel setup**

Channel 4 is used in Figure 14 to simulate a type K thermocouple.

The cabling from the front panel of the P470 to the reference junction can be ordinary connectors and copper wiring. Appropriate connector shells should be in place to ensure that all pins within the mating connector are reasonably isothermal and protected against air currents and thermal transients. Shielded wiring is preferred, with the shield bonded to the connector shell, which is the P470 front panel and enclosure.

If the internal temperature sensor of the P470 is to be used, the thermocouple wires should be extended all the way to the front panel of the box, soldered or crimped into the pins of the D25 connector. Program the channel to specify the internal sensor. Because the sensor is on the printed-circuit board, but the reference junction is effectively within the mating connector, isothermal conditions will be poor and errors on the order of tenths of a degree C may be expected.

There are many potential sources of error in thermocouple simulation:

- Imperfection in the P470 programmable voltage source.
- Imperfection in the reference-sensing RTD and the P470 RTD acquisition subsystem.
- Non-isothermal reference junction.
- Voltage drop in thermocouple wires.
- Imperfect thermocouple wire chemistry.

A P470 using the J470 termination box will typically have errors below +/-0.2°C over the full type K range. Type K thermocouples are available with standard accuracy limits of ± 2.2°C or 0.75% of temperature, whichever is greater. “Special limit" thermocouples are available with errors of about half that. Therefore, the P470 will usually not be a significant error contributor compared to the inherent thermocouple accuracy.

Thermocouple wire generally has high resistance compared to copper. For example, a Type J pair, #24 wire, is about 0.9 ohms per foot, and Type K is about 1.5 ohms/foot, as compared to 0.05 ohms/foot for a pair of #24 copper wires –a 30:1 ratio. Thermocouple wire resistance can be a significant source of error. For example, 200 feet of #24 type K extension wire has about 300 ohms resistance. If a measurement device had a low input impedance of, say, 100K ohms, the resulting error would be about 0.3%, 3°C at 1000°C. If a measuring device applied a constant +1 microampere DC current to detect an open thermocouple, the voltage drop in the lead wire would be about 300 microvolts, equivalent to an error of +14 degrees C.

Finer-gage wire has correspondingly higher resistance. Type K extension wire is available in #30 (6 ohms/foot) and #40 (63 ohms/foot.)

Common thermocouple types include:

| Type | +/- Alloys | Range °C | uV/°C nom | USA Colors | IEC Colors |
|---|---|---|---|---|---|
| J | Iron/Constantan | -210 to 1200 | 50 | white/red | black/white |
| K | Chromel/Alumel | -270 to 1372 | 39 | yellow/red | green/white |
| E | Chromel/Constantan | -270 to 1000 | 59 | violet/red | violet/white |
| T | Copper/Constantan | -270 to 400 | 39 | blue/red | brown/white |
| R | Pt-13Rh/Platinum | -50 to 1768 | 5 | black/red | orange/white |
| S | Pt-10Rh/Platinum | -50 to 1768 | 5 | black/red | orange/white |
| B | Pt-30Rh/Pt-6Rh | 0 to 1820 | 10 | grey/red | grey/white |

| | | | | | |
|---|---|---|---|---|---|
| N | Nicrosil/Nisil | 70 to 1300 | 26 | orange/red | pink/white |

Note that the negative wire is always red in the USA/ANSI color code, and always white for the IEC colors.

Type B has a near-zero thermal coefficient at room temperature, so is often used without a reference junction sensor, namely programmed to use a simulated ice point reference.

Type J is not recommended below 0 °C

# 15 *Versions*

P470-1: Standard 8-channel isolated thermocouple simulator, 0°C to 60°C

# 16 *Customization*

Consult factory for information about additional custom versions.

# 17 *Revision History*

## 17.1 *Hardware Revision History*

| | |
|---|---|
| 23A470-1A | April 2015<br>Initial P470 release |
| 23A470-1B | July 2018<br>Improved manufacturability<br>Functionality equivalent to Rev. A |
| 23A470-1C | April 2022<br>Improved manufacturability<br>Functionality equivalent to Rev. B |

## 17.2 *Firmware Revision History*

| | |
|---|---|
| 23E470A1 | April 2015<br>Initial firmware release |
| 23E470B1 | July 2017<br>Fixed bug that caused lost characters when sending long strings via UART. |
| 23E470C1 | July 2017<br>Expanded BIST Limits. |
| 23E471A | July 2018<br>For hardware Rev. B |
| 23E471B | April 2022<br>Allows the use of two different ethernet chips. |

# 18 *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 |
| J71-1: | Dual D9 female to Agilent 34104A cable |
| J75-1: | D9 female to two(2) dual banana plug cable |
| J470-1: | 8-channel D25 female isothermal termination box |
| J475-2: | 8-channel D25 female Din rail termination panel w/ reference junction sensor |
| P10-1: | 19" rack mount shelf (two p-boxes per rack) |
| P51-1: | Mounting Flange |