HP E1535A
Watchdog Timer
Signal Conditioning Plug-on

User's Manual

Enclosed is the User's Manual for the HP E1535 Signal Conditioning Plug-on. Insert this manual in your VXI Module's User's Manual behind the "Signal Conditionining Plug-ons" divider.

APPLICABILITY
This SCP is used with the HP E1415.

HEWLETT PACKARD

E1535-90002

Manual Part Number: E1535-90002

Printed: October 1996 Edition 2
Printed in U.S.A. E 1096

HP E1535
Watchdog Timer
Signal Conditioning Plug-on

Introduction

The HP E1535 Watchdog Timer is used with the HP E1415 Algorithmic Closed Loop Controller to provide alarm signals in the event of a control failure. The HP E1535 provides:

- Four algorithm controllable SPST relays
- A watchdog timer that must be "petted" at intervals by an algorithm or it will time-out, de-energizing (opening) all four of the relays. The oscillator for this timer is implemented locally on the SCP and will continue to run until time-out even if the HP E1415 internal system clocks should fail.
- Jumpers to set different maximum "petting" (i.e. time-out) periods for the watchdog timer.
- A signal input –"I/O Disconnect"– that can be used to cause all of the HP E1415's measurement inputs and control outputs to disconnect from your system.
- A jumper to enable/disable the "I/O Disconnect" signal.

The HP E1535's watchdog timer, along with Algorithm Language control of its four relays can be used to signal various levels of process control alarms. See "Programming with the Algorithm Language" for examples.

About this Manual

This manual shows you how to configure the Signal Conditioning Plug-on (SCP) option jumpers, determine the SCP's ID using a SCPI command, and control the SCP with Algorithm Language statements. The contents of this manual are:

Identifying the Plug-on

You'll find the HP part number on the side opposite the connectors, to the left of the serial number bar code. For the HP E1535, the part number is : E1535-66501

Installation

Installation of this SCP involves setting its configuration jumpers and then installing it in the HP E1415.

Jumper Configuration

The configuration jumpers are located on the top of the SCP (side opposite the connectors). There are two jumpers and eight positions for them on the HP E1535. Five of the positions (labeled as indicated below) perform a function, the other three can be used to store a jumper when its function is not required. There are four positions that control the watchdog time-out period, and one position that enables or disables the "I/O Disconnect" input.

Watchdog Period Jumper

The watchdog period jumper can be placed in one of four positions. Each position is labeled according to the watchdog time-out period:

- 12ms for the .012 second period.
- 120ms for the 0.12 second period.
- 1.2s for the 1.2 second period.
- 12s for the 12 second period.

Note If a jumper is not place in one of the four time-out period positions, or more than one jumper is placed in these positions, the relays will not be programmable, and will remain open.

"I/O Disconnect" Enable Jumper

When a jumper is installed in the "DISC" position, "I/O Disconnect" is enabled. Removing the jumper (placing it in an unlabeled position) disables "I/O Disconnect".

Installing into the HP E1415

Installation of the HP E1535 into the HP E1415 is common to other SCPs and is covered in Chapter 1 of your VXI Module User’s Manual.

Field Wiring

The Watchdog Timer has one input for the "I/O Disconnect" signal, a pull-up supply output, and terminal connections for 4 single-poll, single-throw relays. See I/O diagram that follows.

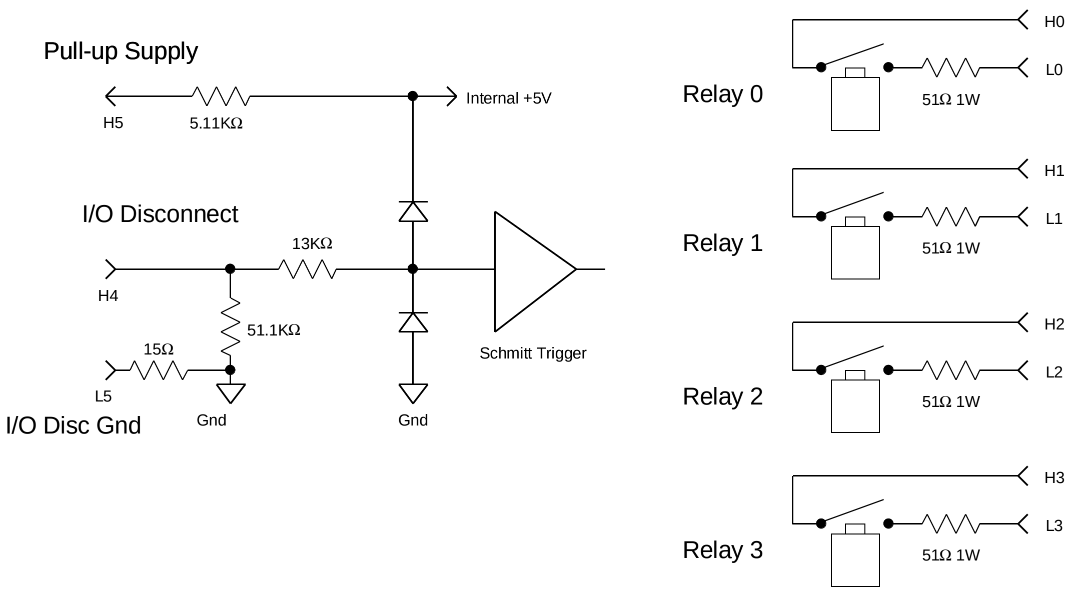

Figure 1 HP E1535 Input/Output Diagram

Connecting at the Terminal Module

The relationship of the Watchdog Timer’s signals to the HP E1415’s channel HI and channel LO terminals at the Terminal Module is dependent on the SCP’s installed position. Refer to the supplied SCP position labels or the following table for this information.

SCP’s Terminal	SCP 0 channels	SCP 1 channels	SCP 2 channels	SCP3 channels	SCP 4 channels	SCP 5 channels	SCP 6 channels	SCP 7 channels
Relay 0	0 H & L	8 H & L	16 H & L	24 H & L	32 H & L	40 H & L	48 H & L	56 H & L
Relay 1	1 H & L	9 H & L	17 H & L	25 H & L	33 H & L	41 H & L	49 H & L	57 H & L
Relay 2	2 H & L	10 H & L	18 H & L	26 H & L	34 H & L	42 H & L	50 H & L	58 H & L
Relay 3	3 H & L	11 H & L	19 H & L	27 H & L	35 H & L	43 H & L	51 H & L	59 H & L
I/O Disc.	4 H	12 H	20 H	28 H	36 H	44 H	52 H	60 H
Pull-Up +	5 H	13 H	21 H	29 H	37 H	45 H	53 H	61 H
I/O Disc -	5 L	13 L	21 L	29 L	37 L	45 L	53 L	61 L

Programming With SCPI Commands

The SCPI command shown here queries the SCP's identification string. The HP E1415 doesn't use SCPI commands to control its relays or watchdog timer. These functions are programmed using the HP E1415's Algorithm Language. See example command sequences that follow.

Checking the ID of the SCP

To verify the SCP type(s) installed on your VXI module, use the SYSTem:CTYPE? (@<channel>) command.

- The *channel* parameter specifies a single channel in the channel range covered by the SCP of interest. The first channel number for each of the eight SCP positions are; 0,8,16,24,32,40,48, and 56.

The value returned for the HP E1535 SCP is:
HEWLETT-PACKARD,E1535A Watchdog Timer SCP,0,0

To determine the type of SCP installed on channels 0 through 7 send

SYST:CTYPE? (@100) *query SCP type @ ch 0*
enter statement here *enter response string*

Reading the Watchdog Configuration/Status Register

The Watchdog SCP provides a way to determine the current positions of the configuration jumpers without removing the HP E1415 or its cover. Also provided is a way to read the SCP's current operating status (see Operating States section that follows). Use the command DIAGnostic:QUERy:SCPREAD? <SCP_reg_addr> to return the contents of the configuration/status register. Address is "whole SCP register address"+2. The formula to calculate the address for each SCP position is (64*SCP_position)+2. To read the config/status register from the Watchdog SCP at position 1 send:

DIAG:QUERY:SCPREAD? 66
execute "enter" statement *return conf/stat register*

The following table shows the meaning of the bit positions in this register.

Register Bit	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Meaning	I/O Disc.	12s	1.2s	120ms	12ms	state1	state0

Bits 0 and 1 show the SCP's current state: 00=IDLE, 10=RUN, 01=ALARM, and 11=ALARM. See the "SCP Operating States" section that follows.

The other bits indicate the current jumper configuration: 0=Jumper Installed.

Programming with the HP E1415 Algorithm Language

There are only two types of operations that your algorithm can perform on the HP E1535; controlling the relays and "petting" the watchdog timer. Both operations require setting the state of digital bits on the SCP.

Channel/Bit Addressing

All functions of the SCP are accessed by the lowest 5 digital bits of its first channel. This first channel number is dependent on the SCP position at which it is installed. The first channel number for each SCP position is:

SCP Position	SCP0	SCP1	SCP2	SCP3	SCP4	SCP5	SCP6	SCP7
First Channel	0	8	16	24	32	40	48	56

The functions are controlled by bits 0 through 4 as follows:

Function	Timer "pet"	Relay 3	Relay 2	Relay 1	Relay 0
Bit Number	4	3	2	1	0

SCP Operating States

The HP E1535 has four operating states:

1. **OFF State**. The SCP is in this state when its VXI mainframe experiences a power failure. In this state of course all four relays are open.

2. **IDLE State**. The SCP is in this state before the INIT command is sent to start algorithm execution, and after ABORT or *RST stops algorithm execution. In this state the watchdog timer clock is disabled and all four relays are open.

3. **RUN State**. The SCP is in this state after an INIT command starts algorithm execution. When entering this state, the watchdog timer clock starts running and all four relays are open but can then be programmed open or closed. Bit 4 must now be toggled by an algorithm at a period that does not exceed the timeout period set by watchdog period jumpers. If the timer times-out, the SCP enters its ALARM State

4. **ALARM State.** The SCP is in this state if the watchdog timer has timed-out (not "petted" often enough). In this state all four relays are open regardless of their programmed states. Only a *RST command will remove this ALARM condition. An ABORT command and a subsequent INIT command can stop and restart algorithm execution but the SCP will still be in its ALARM state, its relays still open and not programmable.

Note The time between watchdog pets should not exceed (period jumper setting)-1.5ms. IF longer, the SCP could enter the ALARM State when you stop your algorithms with the ABORT command. This is because of the time delay between when the ABORT command stops your algorithm and when the command returns the watchdog to its IDLE State.

Controlling Relays

Relays are controlled using the HP E1415's Algorithm Language. To close a relay, assign its bit identifier a 1. To open a relay, assign its bit a 0 (zero). For example:

Close relay 0 and 2, open relay 1 and 3 on HP E1535 in SCP position 1:

```
O108.B0 = 1;        /* close relay 0 */
O108.B2 = 1;        /* close relay 2 */
O108.B1 = 0;        /* open relay 1 */
O108.B3 = 0;        /* open relay 3 */
```

Petting the Watchdog

The term "petting" means to compliment the state of the watchdog bit (bit 4) each pass through your algorithm. Your algorithm must "pet" the watchdog often enough to keep it from timing-out, or the SCP will enter the ALARM state. You can see that if your algorithm fails to execute for any reason, the watchdog timer will time-out. Note that as soon as the HP E1415 receives the INIT command the watchdog clock starts running, so your algorithm must begin "petting" it before it times-out. For example:

```
if( first_loop ) O108.B0 = 1;      /* close relay 0 on first execution */

/************** pet the watchdog ***************/
O108.B4 = ! O108.B4;    /* compliment bit 4 */
/***************************************************/
```

Programming Other Alarm Levels

The watchdog time-out alarm signals that one or all algorithms have failed to execute (no trigger, or disabled by ALG:STATE, or hardware failure). Another potential control problem that can be signaled with the HP E1535's relays is that the VXIbus controller cannot communicate with algorithms. The HP E1415's algorithms, once started, can execute without supervision from the VXIbus controller. Because of this, the controller could fail, while the HP E1415 continues to control one or more processes. To detect this, you can create a custom algorithm that decrements a counter on each execution and opens an HP E1535 relay if the count reaches zero. This forces the application program to keep restoring this counter (an algorithm variable) or a "VXIbus Controller Failure Alarm" (as we'll call it here), will occur (relay opens). Example for ALG24:

```
static float exec_counter = 10;  /* declare counter variable and assign start value */

if( First_loop ) O116.B1 = 1;     /* on first execution close relay 1 @ SCP pos 2 */
```

```
exec_counter = exec_counter - 1;        /* decrement counter */
if( exec_counter <= 0 ) O116.B1 = 0;   /* if counter exhausted, open relay 1 */
```

The VXIbus controller then needs to send

```
ALG:SCALAR  'ALG24','exec_counter',10
ALG:UPDATE
```

before the algorithm executes 10 times or relay 1 will open.

Using the HP E1535 with Standard PIDs

Sensing PID Limit Alarms

When defining a PID algorithm, any of the <*_channel*> parameters can specify variable names instead of actual channels. If you first define a global variable, then specify that variable as the PID's <*alarm_channel*>, the PID will send the alarm value to that variable instead of an output channel. The example below shows a custom algorithm, ALG32. ALG32 closes the relay at O116 on the algorithm's first execution, then watches the alarm globals from ALG1 and ALG2. If either of these alarm variables is true, ALG32 opens the relay controlled by O116. This way the relay will be opened if an alarm limit is exceeded but won't re-close if the PID later retracts the alarm condition.

```
ALG:DEF  'GLOBALS','static float pid_alarm1, pid_alarm2;  /* 2 alarm variables defined */'
ALG:DEF  'ALG1','PIDB(I100,O108,pid_alarm1)'
ALG:DEF  'ALG2','PIDB(I101,O109,pid_alarm2)'
ALG:DEF  'ALG32','if( First_loop ) O116 = 1; if( ( pid_alarm1 || pid_alarm2 ) ) O116 = 0;' /* cntl relay */
```

Standard PIDs and the Watchdog Timer

The standard PID algorithms don't "pet" the watchdog timer. If you want to use the watchdog timer feature with standard PID algorithms, you must create a small custom algorithm who's only function is to "pet" the watchdog. Since this algorithm will be executed along with the PIDs (must remain enabled with ALG:STATE ALGn,ON), the watchdog timer will be satisfied. Example for an HP E1535 in SCP position 1:

Close relay 0 on the first execution of the algorithm and "pet" every time the algorithm executes (HP E1535 in SCP position 1)

```
ALG:DEF 'ALG24','if(First_loop) O108.B0 = 1; O108.B4 = ! O108.B4;  /* pet timer */
```

Using I/O Disconnect

By sourcing a voltage level of 2.5 volts into "I/O Disconnect" input (with jumper "DISC" installed), the HP E1415 will disconnect all of its measurement inputs and control outputs from your process. The HP E1415 has a pair of relay contacts for each of its channels (one contact for HI and one for LO). These relays are used to connect an SCP's channels to the CAL BUS during calibration (*CAL? execution). While in the CAL position each SCP channel is disconnected from the user’s wiring. "I/O Disconnect" simply causes all relays to switch to their CAL position.

Figure 2 shows an example of wiring the HP E1535 to drive "I/O Disconnect" if the SCP enters its ALARM State.

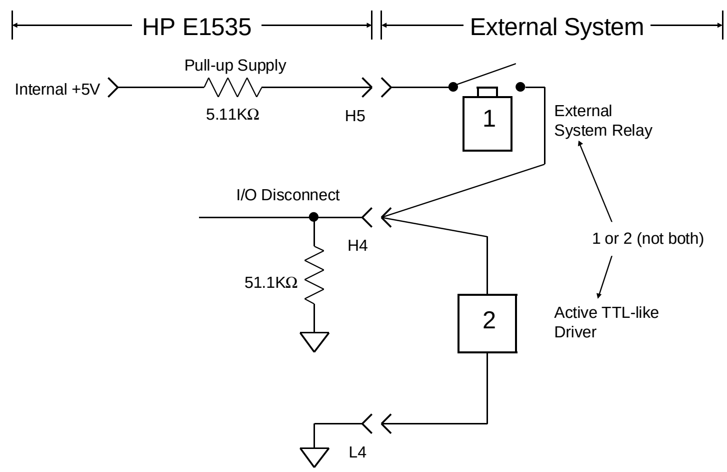

Figure 2 Driving I/O Disconnect

Specifications

General Specifications

Maximum voltage applied to input terminal		Operating: < ±5 V peak
Relay Characteristics	Relay Switching Current Relay switching Voltage Contact Power Rating Contact Resistance Relay Life - 1V @ 10mA (reduced life if signal higher than 1V@10mA)	0.5 A 60 VDC 42VAC 10 W max 0.2 Ω 100E6 operations
Input Characteristics for "I/O Disconnect"	Minimum True Level Maximum False Level Transition Delay Time	2 Volts 0.8 Volts 1 msec