

**WATERS CORPORATION**  
**ENGINEERING TEST SPECIFICATION AND PROCEDURE**  
**COMPANY CONFIDENTIAL**

SYSTEM/ASSEMBLY NAME: **2414 Personality PCB**

PART NUMBER AFFECTED: **510000153**

TEST EQUIPMENT NAME: **Racal Freedom System Functional Tester**

TEST EQUIPMENT NUMBER: **TE82698**

TEST EQUIPMENT LOCATION: **Sparton Electronics, FL**

DOCUMENT NUMBER: **735000661**

AUTHOR(S): **E. Jackson**

REV: **A**      DATE: **9/26/02**

LAST REVISION:

REV:      DATE:

| APPROVED BY                            | SIGNATURE | DATE  |
|----------------------------------------|-----------|-------|
| RESPONSIBLE<br>TEST ENGINEER:          | _____     | _____ |
| RESPONSIBLE<br>PRODUCT ENGINEER:       | _____     | _____ |
| RESPONSIBLE<br>MANUFACTURING ENGINEER: | _____     | _____ |

**FOR DATA SERVICES USE ONLY**

ECO NO. \_\_\_\_\_

DOC  
CONTROL \_\_\_\_\_

OFFICIALLY RELEASED TO CORPORATE DOCUMENTATION CENTER

THIS DOCUMENT WAS THE  
CURRENT EFFECTIVE REVISION ON  
THE DATE ISSUED, AND IS VALID  
FOR GENERAL PRODUCTION AND  
TESTING USE ONLY IF STAMPED IN  
RED BY DATA SERVICES.

THIS DOCUMENT AND ITS CONTENTS ARE EXCLUSIVE PROPERTY OF WATERS CORPORATION, MILFORD MASSACHUSETTS.  
THIS DOCUMENT MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER, WITHOUT PRIOR WRITTEN PERMISSION FROM  
WATERS CORPORATION.

DOCUMENT NO.: 735000661 /REV. A

Page 1 of 24

Form Revision 6.0, 12/18/01

## Table of Contents

|      |                                                                           |    |
|------|---------------------------------------------------------------------------|----|
| I.   | Introduction to the 2414 Functional Test.....                             | 3  |
| II.  | Functional Test System Features .....                                     | 3  |
| III. | Test Descriptions .....                                                   | 3  |
| 1.   | DC Power Supply Input Voltages, Currents, and On Board Voltages.....      | 3  |
| 2.   | Xmit Carriage Return to Console Port.....                                 | 4  |
| 3.   | Fan Rear Panel (External Fan) and Leak Detect (J13) Voltage Outputs ..... | 5  |
| 4.   | ID's -Solenoid-LED-Ext ColHtr-Leak Detect.....                            | 5  |
| 5.   | Valve Outputs J11 and J12 .....                                           | 7  |
| 6.   | Oven Heater and Column Heater Outputs .....                               | 8  |
| 7.   | Event Output and Inputs .....                                             | 8  |
| 8.   | GPIB (IEEE488) Handshake Test.....                                        | 11 |
| 9.   | DAC_1 and DAC_2 Outputs .....                                             | 11 |
| 10.  | LED Voltage and Current Select.....                                       | 12 |
| 11.  | Keypad Test .....                                                         | 13 |
| 12.  | Display Test .....                                                        | 14 |
| 13.  | Front End Board Interface Test (J6).....                                  | 16 |
| 14.  | Read DAC_1_OUTPUT and DAC_2_OUTPUT Via Mux.....                           | 16 |
| 15.  | Read RTD Inputs .....                                                     | 18 |
| 16.  | Read LED_I and GND Via Mux.....                                           | 22 |

## **I. Introduction to the 2414 Functional Test**

The 2414 Functional Test Station is designed to functionally test the Water's Corporation 2414 RIU Detector CPU board.

The UUT should have had an in-circuit test performed prior to testing on this station. This will screen out most catastrophic board problems.

The system uses a vacuum fixture to pull the board down into a bed of nails for test access. Most, but not all, test points are accessed at the UUT's connectors.

## **II. Functional Test System Features**

The Racal Functional Test System is capable of testing many different Waters Corporation PCB board assemblies. A User Interface program will instruct the operator to select the proper program to run.

When the program starts, it verifies the fixture ID before the actual testing begins. Failure tags are printed out on the test system printer.

## **III. Test Descriptions**

Note:

- 1) The UUT is powered up with two shorts connecting J17\_11 to J17\_12 and J17\_15 to J17\_16. The first short is required to force the UUT into Diagnostic mode (manufacturing mode). The second short enables the RS232 Communications port on the CPU2000 Board. Both shorts must be present at all times. The UUT reads the shorts upon power up.
- 2) A carriage return sent to the UUT before each test following Section 4.0 is recommended to clear the communications buffers.
- 3) Upon power up, the UUT may take up to 20 seconds to boot-up and allow RS232 communications.
- 4) Some commands such as "acq -i" may take up to 10 seconds to respond.

### **1. DC Power Supply Input Voltages, Currents, and On Board Voltages**

#### **1.1. +5VDC RMTVCC Voltages**

5VDC is applied to J9 pins 3(+) and 4(-). The voltage is measured at those points to verify the proper voltage. Test Limits: Lo= 4.75VDC Hi= 5.25VDC

#### **1.2. +5VDC VCC Voltages**

5VDC is applied to J9 pins 1,7(+) and 2,8(-). The voltage is measured at those points to verify the proper voltage. Test Limits: Lo= 4.75VDC Hi= 5.25VDC

#### **1.3. +12VDC Voltage**

12VDC is applied to J9 pins 12(+) and 5(-). The voltage is measured at those points to verify the proper voltage. Test Limits: Lo= 11.0VDC Hi= 12.6VDC

#### **1.4. -12VDC Voltage**

12VDC is applied to J9 pins 5(+) and 6(-). The voltage is measured at those points to verify the proper voltage. Test Limits: Lo= -12.6VDC Hi= -11.0VDC

#### **1.5. +24VDC Voltage**

24VDC is applied to J9 pins 9,10(+) and 11(-). The voltage is measured at those points to verify the proper voltage. Test Limits: Lo= 23.5VDC Hi= 24.5VDC

#### 1.6. +5VDC RMTVCC Current

The current is measured via the HP power supply inquiry of the +5VDC power supplied as described above. Test Limits: Lo= 57.0mA Hi= 228.0mA

#### 1.7. +5VDC VCC Current

The current is measured via the HP power supply inquiry of the +5VDC power supplied as described above. Test Limits: Lo= 360.0mA Hi= 1440.0mA

#### 1.8. +12VDC Current

The current is measured via the HP power supply inquiry of the +12VDC power supplied as described above. Test Limits: Lo= 6.0mA Hi= 50.0mA

#### 1.9. -12VDC Current

The current is measured via the HP power supply inquiry of the -12DC power supplied as described above. Test Limits: Lo= 6.0mA Hi= 50.0mA

#### 1.10. +24VDC Current

The current is measured via the HP power supply inquiry of the +24VDC power supplied as described above. Test Limits: Lo= -10.0mA Hi= 10.0mA

#### 1.11. POS5 Regulator Voltage

The voltage is measured at through-hole point POS5 and connector J9 pin 5 AGND. This is the output of U17. Test Limits: Lo= 4.75VDC Hi= 5.25VDC

#### 1.12. NEG5 Regulator Voltage

The voltage is measured at through-hole point NEG5 and connector J9 pin 5 AGND. This is the output of U18. Test Limits: Lo= -5.25VDC Hi= -4.75VDC

#### 1.13. POS11 Regulator Voltage

The voltage is measured at through-hole point POS11 and connector J9 pin 5 AGND. This is the output of U24. Test Limits: Lo= 10.0VDC Hi= 12.0VDC

#### 1.14. NEG11 Regulator Voltage

The voltage is measured at through-hole point NEG11 and connector J9 pin 5 AGND. This is the output of U25. Test Limits: Lo= -12.0VDC Hi= -10.0VDC

#### 1.15. DACVCC Regulator Voltage

The voltage is measured at through-hole point DACVCC and connector J9 pin 5 AGND. This is the output of U16. Test Limits: Lo= 4.75VDC Hi= 5.25VDC

## 2. Xmit Carriage Return to Console Port

A carriage return is sent to the UUT via the RS232 console port J1 on the CPU2000 board (embedded in the fixture, J1\_1=Tx, J1\_2=Rx, J1\_3=Gnd). The port is set default for 38400 baud, 1 stop, 8 bit, no parity, no flow control. A prompt from the CPU2000 is expected to verify the CPU2000's

microprocessor is up alive and talking when interfaced with the UUT. The prompt expected is "CONSOLE:>".

### **3. Fan Rear Panel (External Fan) and Leak Detect (J13) Voltage Outputs**

#### **3.1. External Fan**

The voltage is measured at connector J8\_1 (+) and J8\_2 (-). +24VDC is expected at those points.

Test Limits: Lo= 23.0VDC Hi= 25.0VDC

#### **3.2. Leak Detect 24VDC Output Voltage**

The voltage output is measured at J13\_1 (+) and J13\_6 (-).

Limits: Lo= 23.0VDC Hi= 25.0VDC

#### **3.3. Leak Detect Vcc Output Voltage**

The voltage output is measured at J13\_2 (+) and J13\_5 (-).

Limits: Lo= 4.60VDC Hi= 5.30VDC

### **4. ID's -Solenoid-LED-Ext ColHtr-Leak Detect**

A (peek *hexaddress*) command is sent through the RS232 port to the CPU2000 Bd. A hex value will be returned (XX) showing the results. The inputs are memory mapped and can read via a peek command followed by the address. The uut inputs are stimulated one by one and read each time to verify the inputs are functioning. Most inputs have a digital input and a ground available on the uut connector used. By using a relay to short these points, usually pulled high on the uut, the high input can be tested and then the low input, via a good ground point on the uut, can be tested when the relay shorts these points. Refer to the 2414 Performance Specification 810000153.

Actual example:

CONSOLE:>peek 0x50000F

5242895 (0x50000f): 255 (0xff)

CONSOLE:>

#### **4.1. Solenoid Driver1 ID0 Bit High**

A "peek 0x50000F" command is sent to the CPU2000.

SDRV1\_B0 bit reads high. (0xff)

#### **4.2. Solenoid Driver1 ID0 Bit Low**

A relay shorts J11\_5 to J11\_2. A "peek 50000F" command is sent to the CPU2000.

SDRV1\_B0 bit reads low. (0xfe)

#### **4.3. Solenoid Driver1 ID1 Bit High**

A "peek 50000F" command is sent to the CPU2000.

SDRV1\_B1 bit reads high. (0xff)

#### **4.4. Solenoid Driver1 ID1 Bit Low**

A relay shorts J11\_6 to J11\_3. A “peek 50000F” command is sent to the CPU2000.  
SDRV1\_B1 bit reads low. (0xfd)

4.5. Solenoid Driver2 ID0 Bit High

A “peek 50000F” command is sent to the CPU2000.  
SDRV2\_B0 bit reads high. (0xff)

4.6. Solenoid Driver2 ID0 Bit Low

A relay shorts J12\_5 to J12\_2. A “peek 50000F” command is sent to the CPU2000.  
SDRV2\_B0 bit reads low. (0xfb)

4.7. Solenoid Driver2 ID1 Bit High

A “peek 50000F” command is sent to the CPU2000.  
SDRV2\_B1 bit reads high. (0xff)

4.8. Solenoid Driver2 ID1 Bit Low

A relay shorts J12\_6 to J12\_3. A “peek 50000F” command is sent to the CPU2000.  
SDRV2\_B1 bit reads low. (0xf7)

4.9. Leak Detect Present Bit High

A “peek 50000F” command is sent to the CPU2000.  
\_LK\_DET\_PRESENT bit reads high. (0xff)

4.10. Leak Detect Present Bit Low

A relay shorts J13\_3 to J13\_5. A “peek 50000F” command is sent to the CPU2000.  
\_LK\_DET\_PRESENT bit reads low. (0xfd)

4.11. Leak Detect Bit High

A “peek 50000F” command is sent to the CPU2000.  
\_LEAK bit reads high. (0xff)

4.12. Leak Detect Bit Low

A relay shorts J13\_4 to J13\_6. A “peek 50000F” command is sent to the CPU2000.  
\_LEAK bit reads low. (0xef)

4.13. LED ID0 Bit High

A “peek 50000E” command is sent to the CPU2000.  
LED\_ID\_B0 bit reads high. (0xf8)

4.14. LED ID0 Bit Low

A relay shorts J3\_1 to J3\_3. A “peek 50000E” command is sent to the CPU2000.  
LED\_IB\_B0 bit reads low. (0xb8)

4.15. LED ID1 Bit High

A “peek 50000E” command is sent to the CPU2000.  
LED\_ID\_B1 bit reads high. (0xf8)

#### 4.16. LED ID1 Bit Low

A relay shorts J3\_2 to J3\_4. A “peek 50000E” command is sent to the CPU2000.  
LED\_ID\_B1 bit reads low. (0x78)

#### 4.17. Column Heater1 ID0 Bit High

A “peek 50000E” command is sent to the CPU2000.  
COL\_ID\_B0 bit reads high. (0xf8)

#### 4.18. Column Heater1 ID0 Bit Low

A relay shorts J2\_1 to J2\_8. A “peek 50000E” command is sent to the CPU2000.  
COL\_ID\_B0 bit reads low. (0xf0)

#### 4.19. Column Heater1 ID1 Bit High

A “peek 50000E” command is sent to the CPU2000.  
COL\_ID\_B1 bit reads high. (0xf8)

#### 4.20. Column Heater1 ID1 Bit Low

A relay shorts J2\_2 to J2\_8. A “peek 50000E” command is sent to the CPU2000.  
COL\_ID\_B1 bit reads low. (0xe8)

#### 4.21. Column Heater1 ID2 Bit High

A “peek 50000E” command is sent to the CPU2000.  
COL\_ID\_B2 bit reads high. (0xf8)

#### 4.22. Column Heater1 ID2 Bit Low

A relay shorts J2\_3 to J2\_8. A “peek 50000E” command is sent to the CPU2000.  
COL\_ID\_B2 bit reads low. (0xd8)

### 5. Valve Outputs J11 and J12

These tests verify the open and shut states of the recycle and purge valves. A 70-ohm 25W resistor is connected to J11\_1 and J11\_4, and to J12\_1 and J12\_2, to act as a simulated valve load. A short is made between J11\_2 to J11\_5, and J12\_2 to J12\_5, to simulate Jumpers ID0. Note: Power must be cycled off and on before J11 and J12 testing due to a bug in the 2414 diagnostic firmware.

#### 5.1. Valve Output J11 Open Voltage

The voltage is measured at connector J11\_4(+) and J9\_11(24Gnd). U36 is used to drive the output. An Initialize Command “purge –i” is sent to the UUT via the CPU2000 RS232 Port to initialize and Open the valve.

Test Limits: Lo= 10.0VDC Hi= 18.0VDC

#### 5.2. Valve Output J11 Shut Voltage

The voltage is measured at connector J11\_4(+) and J9\_11(24Gnd). U36 is used to drive the output. A Shut Command “purge –s” is sent to the UUT via the CPU2000 RS232 Port.

Test Limits: Lo= 22.0VDC Hi= 25.0VDC

### 5.3. Valve Output J12 Open Voltage

The voltage is measured at connector J12\_4(+) and J9\_11(24Gnd). U36 is used to drive the output. An Initialize Command “purge –i” is sent to the UUT via the CPU2000 RS232 Port to initialize and Open the valve.

Test Limits: Lo= 10.0VDC Hi= 18.0VDC

### 5.4. Valve Output J12 Shut Voltage

The voltage is measured at connector J12\_4(+) and J9\_11(24Gnd). U36 is used to drive the output. A Shut Command “purge –s” is sent to the UUT via the CPU2000 RS232 Port.

Test Limits: Lo= 22.0VDC Hi= 25.0VDC

## 6. Oven Heater and Column Heater Outputs

The following tests verify the on and off states of the oven heater output. A 50-ohm 50W resistor is connected to J16\_1 and J16\_2 to act as a simulated load. A jumper is connected across J14\_1 and J14\_2 to bypass the thermal switch and complete the current path.

### 6.1. Oven Heater Output Off

The voltage is measured at connector J16\_1(+) and J9\_11(24Gnd). Q3 is used to drive the output. An Oven Heat Off Command “poke 0x500021 0x00” (new firmware send “poke –w 0xFFFFD4 0x7800”) is sent to the UUT via the CPU2000 RS232 Port. +24VDC is expected at those points.

Test Limits: Lo= 22.0VDC Hi= 25.0VDC

### 6.2. Oven Heater Output On

The voltage is measured at connector J16\_1(+) and J9\_11(24Gnd). Q3 is used to drive the output. An Oven Heat On Command “poke 0x500021 0x04” (new firmware send “poke –w 0xFFFFD4 0x0000”) is sent to the UUT via the CPU2000 RS232 Port. 0VDC is expected at those points.

Test Limits: Lo= -1.0VDC Hi= 1.0VDC

The following tests verify the open and shut states of the column heater output. A 5 volt 10K pull up is tied to J2\_A5. A digital input is connected to J2\_A5 to detect the column heater output.

### 6.3. Column Heater Output Low

A column heater off command “poke 0x500021 0x20” is sent to the UUT via the CPU2000 RS232 port.

A digital input reads J2\_A5 as Low.

### 6.4. Column Heater Output High

A column heater on command “poke 0x500021 0x00” is sent to the UUT via the CPU2000 RS232 port.

A digital input reads J2\_A5 as High.

## 7. Event Output and Inputs

A 5-volt pull up through 10K is tied to J1\_A12. The 5-volt ground is tied to J1\_A13. A digital input is connected to J1\_A12 to detect the Event Out relay closure.

#### 7.1. Event Out Relay Open

An event out open command "switch -0" is sent to the UUT via the CPU2000 RS232 port. A digital input reads J1\_A12.

Expect to read high with relay open.

#### 7.2. Event Out Relay closed

An event out close command "switch -1" is sent to the UUT via the CPU2000 RS232 port. A digital input reads J1\_A12.

Expect to read low with relay closed.

Below, the UUT Purge Input consists of J1 A4 and A5. A Test System relay is used to short or open these points. J1 A4 is normally at a Hi logic level. Shorting J1 A4 to A5 puts the input of A4 at the UUT ground potential (Low) of A5. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

Actual example:

```
CONSOLE:> show PurgeInput.curState
```

```
PurgeInput.curState = True
```

```
CONSOLE:>
```

#### 7.3. Purge Input Hi

Input open, expect port command "show PurgeInput.curState\r" result to be TRUE.

#### 7.4. Purge Input Lo

Input closed, expect port command "show PurgeInput.curState\r" result to be FALSE.

Below, the UUT Polarity 1 Input consists of J1 A6 and A7. A Test System relay is used to short or open these points. J1 A6 is normally at a Hi logic level. Shorting J1 A6 to A7 puts the input of A6 at the UUT ground potential (Low) of A7. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

#### 7.5. Polarity 1 Input Hi

Input open, expect port command "show Polarity1Input.curState\r" result to be TRUE.

#### 7.6. Polarity 1 Input Lo

Input closed, expect port command "show Polarity1Input.curState\r" result to be FALSE.

Below, the UUT Polarity 2 Input consists of J1 B6 and B7. A Test System relay is used to short or open these points. J1 B6 is normally at a Hi logic level. Shorting J1 B6 to B7 puts the input of B6 at the UUT ground potential (Low) of B7. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

#### 7.7. Polarity 2 Input Hi

Input open, expect port command "show Polarity2Input.curState\r" result to be TRUE.

#### 7.8. Polarity 2 Input Lo

Input closed, expect port command "show Polarity2Input.curState\r" result to be FALSE.

Below, the UUT Auto Zero Input consists of J1 B1 and B2. A Test System relay is used to short or open these points. J1 B1 is normally at a Hi logic level. Shorting J1 B1 to B2 puts the input of B1 at the UUT ground potential (Low) of B2. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

#### 7.9. Auto Zero Input Hi

Input open, expect port command "show AutozeroInput.curState\r" result to be TRUE.

#### 7.10. Auto Zero Input Lo

Input closed, expect port command "show AutozeroInput.curState\r" result to be FALSE.

Below, the UUT Chart Mark Input consists of J1 A9 and A10. A Test System relay is used to short or open these points. J1 A9 is normally at a Hi logic level. Shorting J1 A9 to A10 puts the input of A9 at the UUT ground potential (Low) of A10. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

#### 7.11. Chart Mark Input Hi

Input open, expect port command "show ChartMarkInput.curState\r" result to be TRUE.

#### 7.12. Chart Mark Input Lo

Input closed, expect port command "show ChartMarkInput.curState\r" result to be FALSE.

Below, the UUT Recycle Valve Input consists of J1 B4 and B5. A Test System relay is used to short or open these points. J1 B4 is normally at a Hi logic level. Shorting J1 B4 to B5 puts the input of B4 at the UUT ground potential (Low) of B5. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

#### 7.13. Recycle Valve Input Hi

Input open, expect port command "show RecycleInput.curState\r" result to be TRUE.

#### 7.14. Recycle Valve Input Lo

Input closed, expect port command "show RecycleInput.curState\r" result to be FALSE.

Below, the UUT Inject Start Input consists of J1 A1 and A2. A Test System relay is used to short or open these points. J1 A1 is normally at a Hi logic level. Shorting J1 A1 to A2 puts the input of A1 at

the UUT ground potential (Low) of A2. The results are read by a show command sent to the UUT via the CPU2000 RS232 port.

#### 7.15. Inject Start Input Hi

Input open, expect port command "show InjectInput.curState\r" result to be TRUE.

#### 7.16. Inject Start Input Lo

Input closed, expect port command "show InjectInput.curState\r" result to be FALSE.

### 8. GPIO (IEEE488) Handshake Test

An RS232 command "setieeeaddress 30" is sent to the UUT to set the GPIB address to 30. An ID command is sent to the UUT via the IEEE port J7, send "ID". A response is expected from that command via port J7, receive "DET410".

### 9. DAC 1 and DAC 2 Outputs

These tests are measured via the connector J1 B9(+) and B10(-). The UUT is sent an analog out command "setoutput 1 x" (where x ranges from -2.0 to 2.0) via the CPU2000 RS232 port to output the various voltages.

Actual example:

CONSOLE:> setoutput 1 1.0

CONSOLE:>

#### 9.1. DAC\_1 Output to 0VDC

Test Limits: Lo= -0.05VDC Hi= 0.05VDC

#### 9.2. DAC\_1 Output to 2VDC

Test Limits: Lo= 1.9VDC Hi= 2.1VDC

#### 9.3. DAC\_1 Output to 1VDC

Test Limits: Lo= 0.95VDC Hi= 1.05VDC

#### 9.4. DAC\_1 Output to -2VDC

Test Limits: Lo= -2.1VDC Hi= -1.9VDC

#### 9.5. DAC\_1 Output to -1VDC

Test Limits: Lo= -1.05VDC Hi= -0.95VDC

These tests are measured via the connector J1 B12(+) and B13(-). The UUT is sent an analog out command "setoutput 2 x" (where x ranges from -2.0 to 2.0) via the CPU2000 RS232 port to output the various voltages.

Actual example:

CONSOLE:> setoutput 2 1.0

CONSOLE:>

9.6. DAC\_2 Output to 0VDC

Test Limits: Lo= -0.05VDC Hi= 0.05VDC

9.7. DAC\_2 Output to 2VDC

Test Limits: Lo= 1.9VDC Hi= 2.1VDC

9.8. DAC\_2 Output to 1VDC

Test Limits: Lo= 0.95VDC Hi= 1.05VDC

9.9. DAC\_2 Output to -2VDC

Test Limits: Lo= -2.1VDC Hi= -1.9VDC

9.10. DAC\_2 Output to -1VDC

Test Limits: Lo= -1.05VDC Hi= -0.95VDC

**10. LED Voltage and Current Select**

The UUT is sent a command through the CPU2000 RS232 port. The LED drive current is controlled by the RS232 command and measured via a DC ammeter placed at the connector J3 pin 8(+) and 7(-). The Current select voltages are measured at the connector J3.

10.1. Current Select J3\_5 Output

The voltage is measured at J3\_5(+) and J9\_5(-)AGND.

Test Limits: Lo= 2.40VDC Hi= 2.60VDC

10.2. Current Select J3\_6 Output

The voltage is measured at J3\_6(+) and J9\_5(-)AGND.

Test Limits: Lo= 0.30VDC Hi= 0.55VDC

10.3. LED ON with Lowest Current Level

The UUT is sent a “poke 0x500021 0xbf” command to turn on the LED circuitry. An “led -d 100” command is sent to force current to the lowest level. A DC ammeter on J3\_8(+) and J3\_7(-) is used to measure the current.

Test Limits: Lo= 44.0mA Hi= 53.0mA

10.4. LED ON with Highest Current Level

The UUT is sent a “poke 0x500021 0xbf” command to turn on the LED circuitry. An “led -i 100” command is sent to force current to the highest level. A DC ammeter on J3\_8(+) and J3\_7(-) is used to measure the current.

Test Limits: Lo= 158.0mA Hi= 195.0mA

10.5. LED ON with 115mA Current Level

The UUT is sent a “poke 0x500021 0xbf” command to turn on the LED circuitry. An “led -d 100” command is sent to force current to the lowest level. Then an “led -i 50” command is sent to force current to the 115mA level. A DC ammeter on J3\_8(+) and J3\_7(-) is used to measure the current. Test Limits: Lo= 105.0mA Hi= 125.0mA

#### 10.6. LED OFF Current Check

The UUT is sent a “poke 0x500021 0xff” command to turn off the LED circuitry. A DC ammeter on J3\_8(+) and J3\_7(-) is used to measure the current. Test Limits: Lo= -1.0mA Hi= 1.0mA

### 11. Keypad Test

The UUT keypad interface circuitry consists of connector J22 pins 1 through 10. The five Y inputs are pulled high to 5 volts on the UUT. A digital output driver is connected to each of these inputs to verify their status by driving only one low at a time. A “peek 0x500002” command via the CPU2000 RS232 port is used to query the input.

The five X outputs are pulled high via fixture pull up resistors. The UUT outputs are connected to five test system digital inputs. A “poke 0x500002 *value*” command via the CPU2000 RS232 port is used to drive the board outputs high or low.

NOTE: A “suspend keyscan” command must be sent prior to starting these tests in order to keep the UUT CPU from resetting the memory locations during this test.

Actual example: (Y)

CONSOLE:> peek 0x500002

5242882 (0x500002): 031 (0x1f)

CONSOLE:>

Actual example: (X)

CONSOLE:> poke 0x500002 0x1f

CONSOLE:>

#### 11.1. Y1 through Y5 All Inputs High

All test system digital drivers to Y1 – Y5 are set high. A “peek 0x500002” command is sent to the CPU2000 RS232 port.

All inputs read high by the serial command response (0x1f).

#### 11.2. Y1 Low All Other Inputs High

Y1 input is set low. A “peek 0x500002” command is sent to the CPU2000 RS232 port.

Y1 reads low and all other inputs read high by the serial command response (0x1e).

#### 11.3. Y2 Low All Other Inputs High

Y2 input is set low. A “peek 0x500002” command is sent to the CPU2000 RS232 port.

Y2 reads low and all other inputs read high by the serial command response (0x1d).

#### 11.4. Y3 Low All Other Inputs High

Y3 input is set low. A “peek 0x500002” command is sent to the CPU2000 RS232 port.

THIS DOCUMENT AND ITS CONTENTS ARE EXCLUSIVE PROPERTY OF WATERS CORPORATION, MILFORD MASSACHUSETTS. THIS DOCUMENT MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER, WITHOUT PRIOR WRITTEN PERMISSION FROM WATERS CORPORATION.

Y3 reads low and all other inputs read high by the serial command response (0x1b).

11.5. Y4 Low All Other Inputs High

Y4 input is set low. A “peek 0x500002” command is sent to the CPU2000 RS232 port.  
Y4 reads low and all other inputs read high by the serial command response (0x17).

11.6. Y5 Low All Other Inputs High

Y5 input is set low. A “peek 0x500002” command is sent to the CPU2000 RS232 port.  
Y5 reads low and all other inputs read high by the serial command response (0x0f).

11.7. X1 through X5 All Outputs High

A “poke 0x500002 0x1f” command is sent to the CPU2000 RS232 port.  
All UUT outputs X1 – X5 read high via the test system digital inputs.

11.8. X1 Low All Other Inputs High

A “poke 0x500002 0x1e” command is sent to the CPU2000 RS232 port.  
X1 output reads low and all other outputs read high via the test system digital inputs.

11.9. X2 Low All Other Inputs High

A “poke 0x500002 0x1d” command is sent to the CPU2000 RS232 port.  
X2 output reads low and all other outputs read high via the test system digital inputs.

11.10. X3 Low All Other Inputs High

A “poke 0x500002 0x1b” command is sent to the CPU2000 RS232 port.  
X3 output reads low and all other outputs read high via the test system digital inputs.

11.11. X4 Low All Other Inputs High

A “poke 0x500002 0x17” command is sent to the CPU2000 RS232 port.  
X4 output reads low and all other outputs read high via the test system digital inputs.

11.12. X5 Low All Other Inputs High

A “poke 0x500002 0x0f” command is sent to the CPU2000 RS232 port.  
X5 output reads low and all other outputs read high via the test system digital inputs.

## 12. Display Test

12.1. VCC Output

The voltage is measured at connector J19\_2(VCC) and J19\_1(GND). +5VDC is expected at those points.

Test Limits: Lo= 4.60VDC Hi= 5.30VDC

12.2. \_RS Output

The voltage is measured at connector J19\_17(\_RS) and J19\_1(GND). +5VDC is expected at those points.

Test Limits: Lo= 4.60VDC Hi= 5.30VDC

### 12.3. P12V Output

The voltage is measured at connector J19\_20(+) and J19\_19(GND). +12VDC is expected at those points.

Test Limits: Lo= 10.0VDC Hi= 13.0VDC

### 12.4. DCS2 Output

The voltage is measured at connector J19\_16(DCS2) and J19\_1(GND). 0VDC is expected at those points.

Test Limits: Lo= -0.10VDC Hi= 0.10VDC

### 12.5. DCS1 Output

The voltage is measured at connector J19\_15(DCS1) and J19\_1(GND). +5VDC is expected at those points.

Test Limits: Lo= 4.60VDC Hi= 5.30VDC

### 12.6. VL Output

The contrast output adjust circuit requires a test system supplied voltage. Connect a negative 9.3VDC to J19\_18(- VEE) and J19\_1(+ GND). Connect a dc voltmeter to J19\_3(+ VL) and J19\_1(- GND).

Test Limits: Lo= -9.80VDC Hi= -5.70VDC

### 12.7. VL Output Contrast Adjust Up

The contrast output adjust circuit requires a test system supplied voltage. Connect a negative 9.3VDC to J19\_18(- VEE) and J19\_1(+ GND). Connect a dc voltmeter to J19\_3(+ VL) and J19\_1(- GND). An RS232 command "contrast -i 650" (new firmware send "contrast -i 1000") is sent to adjust U44 to the highest output level for VL output.

Test Limits: Lo= -9.80VDC Hi= -8.80VDC

### 12.8. VL Output Contrast Adjust Down

The contrast output adjust circuit requires a test system supplied voltage. Connect a negative 9.3VDC to J19\_18(- VEE) and J19\_1(+ GND). Connect a dc voltmeter to J19\_3(+ VL) and J19\_1(- GND). An RS232 command "contrast -d 650" (new firmware send "contrast -d 1000") is sent to adjust U44 to the lowest output level for VL output.

Test Limits: Lo= -6.70VDC Hi= -5.70VDC

### 12.9. VL Output Contrast Adjust Middle

The contrast output adjust circuit requires a test system supplied voltage. Connect a negative 9.3VDC to J19\_18(- VEE) and J19\_1(+ GND). Connect a dc voltmeter to J19\_3(+ VL) and J19\_1(- GND). An RS232 command "contrast -i 325" (new firmware send "contrast -i 500") is sent to adjust U44 to the midpoint output level for VL output.

Test Limits: Lo= -8.80VDC Hi= -7.50VDC

### 12.10. E Output

A 0 to 5 volt digital signal burst is sent out on J19\_6. The Counter measures the average signal voltage level. A signal must be present in order for the level to not be railed high or low.

Test Limits: Lo= 1.30VDC Hi= 3.30VDC

#### 12.11. RD\_WR Output

A 0 to 5 volt digital signal burst is sent out on J19\_5. The Counter measures the average signal voltage level. A signal must be present in order for the level to not be railed high or low.

Test Limits: Lo= 2.0VDC Hi= 4.30VDC

#### 12.12. D\_I Output

A 0 to 5 volt digital signal burst is sent out on J19\_4. The Counter measures the average signal voltage level. A signal must be present in order for the level to not be railed high or low.

Test Limits: Lo= 2.0VDC Hi= 4.30VDC

### 13. Front End Board Interface Test (J6)

The Front End connector J6 is measured for all DC voltage outputs. The CPU data lines are verified by attempting communications to the Front End Analog board located in the fixture. J3\_2 is shorted to J3\_4 to provide the specified LED ID value required.

#### 13.1. POS5 Voltage

The voltage is measured at connector J6\_5(+) and J6\_11(-)AGND.

Test Limits: Lo= 4.75VDC Hi= 5.25VDC

#### 13.2. NEG5 Voltage

The voltage is measured at connector J6\_7(+) and J6\_12(-)AGND.

Test Limits: Lo= -5.25VDC Hi= -4.75VDC

#### 13.3. POS11 Voltage

The voltage is measured at connector J6\_1(+) and J6\_9(-)AGND.

Test Limits: Lo= 10.0VDC Hi= 13.0VDC

#### 13.4. NEG11 Voltage

The voltage is measured at connector J6\_3(+) and J6\_10(-)AGND.

Test Limits: Lo= -13.0VDC Hi= -10.0VDC

#### 13.5. RMTVCC Voltage

The voltage is measured connector J6\_15(+) and J6\_19(-)RMTGND.

Test Limits: Lo= 4.75VDC Hi= 5.25VDC

#### 13.6. QSPI Communication

The CPU2000 RS232 port is sent an "acq -i" command to initialize the UUT. If there are NO errors to report, the UUT will return just a prompt "CONSOLE:>". If the QSPI bus has communication problems, the UUT will return a long string of errors back. The string will contain the substring "QSPI:ERR\_COMM\_FAILURE".

Close this test by sending "acq -x" to suspend the task.

### 14. Read DAC 1 OUTPUT and DAC 2 OUTPUT Via Mux

The UUT is sent via the console port a 'setoutput 1 *value*' command to output a voltage to J1 B9. The UUT generated voltage is also routed to U14 and called DAC\_1\_OUTPUT. A setmux command is sent to the UUT via the console port for U14 to select the DAC\_1\_OUTPUT input and an xrawread command is sent to get the value from A/D U4. The value is returned in Bit counts that represent the A/D's output. The Test System program calculates the measured voltage at the 1.0 volt level.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command 'acq -i' is sent to initialize the task. Then 'acq -x' is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example:

```

CONSOLE:> acq -i
CONSOLE:> acq -x
CONSOLE:> setoutput 1 1.0
CONSOLE:> setmux dac1

```

```

CONSOLE:> Temp Signal → DAC1_OUT
CONSOLE:> xrawread 1 1

```

```

CONSOLE:> Sec. DiodeA      DiodeB      Temp Counts
      0.00   xxxxxx      xxxxxx      3332952 {this is the 'bits' used in the calc.}

```

#### 14.1. Read Setmux Command

An RS232 "setmux dac1" command is sent.  
Expected to receive "DAC1\_OUT" as a response.

#### 14.2. Read Xrawread Command

An RS232 "xrawread 1 1" command is sent.  
Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

#### 14.3. DAC1 Read 1 Volt Via Mux

The captured bit count from the previous test is converted to volts by the following formula.  

$$\text{Volts} = [ \text{bits}(.3\mu\text{V/bit}) ] . \{ \text{to calculate bits : } \text{bits} = [ (\text{Volt}) / .3\mu\text{V/bit} ] \} , \text{ where } .3 \text{ is derived from } 5 / 2^{24} , \text{ where } 5 \text{ is the A/D's full scale unipolar input and } 2^{24} \text{ is for a 24 bit device.}$$
  
 Test Limits: Lo= 0.850VDC Hi= 1.150VDC

The UUT is sent via the console port a 'setoutput 2 *value*' command to output a voltage to J1 B12. The UUT generated voltage is also routed to U14 and called DAC\_2\_OUTPUT. A setmux command is sent to the UUT via the console port for U14 to select the DAC\_2\_OUTPUT input and an xrawread command is sent to get the value from A/D U4. The value is returned in Bit counts that represent the A/D's output. The Test System program calculates the measured voltage at the 1.0 volt level.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command 'acq -i' is sent to initialize the task. Then 'acq -x' is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example:

THIS DOCUMENT AND ITS CONTENTS ARE EXCLUSIVE PROPERTY OF WATERS CORPORATION, MILFORD MASSACHUSETTS.  
THIS DOCUMENT MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER, WITHOUT PRIOR WRITTEN PERMISSION FROM WATERS CORPORATION.

```
CONSOLE:> acq -i
CONSOLE:> acq -x
CONSOLE:> setoutput 2 1.0
CONSOLE:> setmux dac2
```

```
CONSOLE:> Temp Signal → DAC2_OUT
CONSOLE:> xrawread 1 1
```

```
CONSOLE:> Sec. DiodeA      DiodeB      Temp Counts
          0.00   xxxxxx      xxxxxx      3332932 {this is the 'bits' used in the calc.}
```

#### 14.4. Read Setmux Command

An RS232 “setmux dac2” command is sent.  
Expected to receive “DAC2\_OUT” as a response.

#### 14.5. Read Xrawread Command

An RS232 “xrawread 1 1” command is sent.  
Expected to receive “Temp Counts” as a response. (The bit count is also captured for the next test.)

#### 14.6. DAC2 Read 1 Volt Via Mux

The captured bit count from the previous test is converted to volts by the following formula.  
 $\text{Volts} = [ \text{bits}(.3\text{uV/bit}) ]$ . {to calculate bits :  $\text{bits} = [ (\text{Volt})/.3\text{uV/bit} ]$  }, where .3 is derived from  $5 / 2^{24}$ , where 5 is the A/D's full scale unipolar input and  $2^{24}$  is for a 24 bit device.  
Test Limits: Lo= 0.850VDC Hi= 1.150VDC

### 15. Read RTD Inputs

The CAL input is at mux U14. It is measured by the CPU when sent a ‘calrtd -i’ command. The results are read via the console port. The Test System program calculates the measured ohms by the following equation::  $\text{Ohms} = \text{bits} * 1000 / 8391448$ , where 1000 represents the calibration resistor and 8391448 is the expected bit count for that calibration resistor when ‘calrtd -i’ is selected.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command ‘acq -i’ is sent to initialize the task. Then ‘acq -x’ is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example:

```
CONSOLE:> acq -i
CONSOLE:> acq -x
CONSOLE:> calrtd -i
```

```
Calibration counts →8391311
CONSOLE:>
```

#### 15.1. Read calrtd Command

An RS232 command “calrtd -i” is sent.  
Expected to receive “Calibration counts” as a response. (The bit count is also captured for the next test.)

## 15.2. Read CAL. Ohm Mux Value

The bit count captured from the previous test is calculated into ohms by the following formula.

$\text{Ohms} = \text{bits} * 1000 / 8391448.$

Test Limits: Lo= 990.0 Ohms Hi= 1010.0 Ohms

The RTD\_AIR input is at connector J5 pins 1,1 and pins 3,4. It is measured in the open state and also with a 1000-ohm resistor across it. The results are read via the console port. First a setmux command is given to set mux U14 to RTD\_AIR input. Then an xrawread command gets the value from A/D U14. The Test System program calculates the measured ohms and then degrees Celsius by the following equation::  $\text{Ohms} = \text{bits} * 1000 / \text{calrtdbits}$ , where 1000 represents the calibration resistor and calrtdbits is the measured bit count for the CAL input as a reference (see INT\_CAL\_RTD above). Then degrees Celsius is then calculated by::

$C = (\text{ohms} - 1000) / 3.85$ , where 1000 is ohms for 0 degrees Celsius and 3.85 is the ohms per degree Celsius for the specified RTD when used.

Note that the CAL reference is also taken into consideration in this test. If the INT\_CAL\_RTD test fails, a theoretical calrtdbit count is substituted for this reference. However, if the RTD\_AIR test fails by only a margin, then the CAL reference circuitry may be slightly out of spec. and should be checked also.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command 'acq -i' is sent to initialize the task.

Then 'acq -x' is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example reading 1000-ohm RTD resistance:

CONSOLE:> acq -i

CONSOLE:> acq -x

CONSOLE:> setmux oven

CONSOLE:> Temp Signal → INT\_OVEN

CONSOLE:> xrawread 1 1

| CONSOLE:> Sec. DiodeA | DiodeB | Temp Counts                                    |
|-----------------------|--------|------------------------------------------------|
| 0.00    xxxxxx        | xxxxxx | 8375342 {this is the 'bits' used in the calc.} |

## 15.3. Read Setmux Command

An RS232 "setmux oven" command is sent.

Expected to receive "INT\_OVEN" as a response.

## 15.4. Read Xrawread Command

An RS232 "xrawread 1 1" command is sent.

Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

## 15.5. RTD\_AIR Read Open via Mux

The bit count captured from the previous test is calculated into degrees C by the following formula.

$\text{Ohms} = \text{bits} * 1000 / 8391448.$   $C = (\text{ohms} - 1000) / 3.85$ , where 1000 is ohms for 0 degrees Celsius and 3.85 is the ohms per degree Celsius for the specified RTD when used.

Expecting greater than 100 degrees Celsius or (>1385 ohms).

Test Limits: Lo= 100.0°C Hi= 1000.0°C

#### 15.6. Read Xrawread Command

A 1K-ohm resistor is connected between J5\_1,2 and J5\_3,4. An RS232 “xrawread 1 1” command is sent.

Expected to receive “Temp Counts” as a response. (The bit count is also captured for the next test.)

#### 15.7. RTD\_AIR Read 1000 Ohms via Mux

The bit count captured from the previous test is calculated into degrees C by the following formula.  
 $\text{Ohms} = \text{bits} * 1000 / 8391448$ .  $C = (\text{ohms} - 1000) / 3.85$ , where 1000 is ohms for 0 degrees Celsius and 3.85 is the ohms per degree Celsius for the specified RTD when used.

Expecting approximately 0 degrees Celsius or (1000 ohms).

Test Limits: Lo= -2.0°C Hi= 2.0°C

The CAL input is at mux U14. It is measured by the CPU when sent a ‘calrtd -x’ command. The results are read via the console port. The Test System program calculates the measured ohms by the following equation:  $\text{Ohms} = \text{bits} * 1000 / 3473364$ , where 1000 represents the calibration resistor and 3473364 is the expected bit count for that calibration resistor when ‘calrtd -x’ is selected.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command ‘acq -i’ is sent to initialize the task.

Then ‘acq -x’ is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example:

CONSOLE:> acq -i

CONSOLE:> acq -x

CONSOLE:> calrtd -x

Calibration counts →3473395

CONSOLE:>

#### 15.8. Read calrtd Command

An RS232 command “calrtd -x” is sent.

Expected to receive “Calibration counts” as a response. (The bit count is also captured for the next test.)

#### 15.9. Read CAL. Ohm Mux Value

The bit count captured from the previous test is calculated into ohms by the following formula.

$\text{Ohms} = \text{bits} * 1000 / 3473364$ .

Test Limits: Lo= 990.0 Ohms Hi= 1010.0 Ohms

The COL\_RTD input is at connector J2 pin 6 and pin 7. It is measured in the open state and also with a 1000-ohm resistor across it. The results are read via the console port. First a setmux command is given

THIS DOCUMENT AND ITS CONTENTS ARE EXCLUSIVE PROPERTY OF WATERS CORPORATION, MILFORD MASSACHUSETTS.  
THIS DOCUMENT MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER, WITHOUT PRIOR WRITTEN PERMISSION FROM  
WATERS CORPORATION.

to set mux U14 to COL\_RTD input. Then an xrawread command gets the value from A/D U14. The Test System program calculates the measured ohms and then degrees Celsius by the following equation::  
$$\text{Ohms} = \text{bits} * 1000 / \text{calrtdbits}$$
, where 1000 represents the calibration resistor and calrtdbits is the measured bit count for the CAL input as a reference (see EXT\_CAL\_RTD above). Then degrees Celsius is then calculated by::

$$C = (\text{ohms} - 1000) / 3.85$$
, where 1000 is ohms for 0 degrees Celsius and 3.85 is the ohms per degree Celsius for the specified RTD when used.

Note that the CAL reference is also taken into consideration in this test. If the EXT\_CAL\_RTD test fails, a theoretical calrtdbit count is substituted for this reference. However, if the COL\_RTD test fails by only a margin, then the CAL reference circuitry may be slightly out of spec. and should be checked also.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command 'acq -i' is sent to initialize the task. Then 'acq -x' is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example reading 1000-ohm RTD resistance:

```
CONSOLE:> acq -i
```

```
CONSOLE:> acq -x
```

```
CONSOLE:> setmux ext
```

```
CONSOLE:> Temp Signal → EXT_COL
```

```
CONSOLE:> xrawread 1 1
```

```
CONSOLE:> Sec. DiodeA      DiodeB      Temp Counts
      0.00   xxxxxx      xxxxxx      3473494 {this is the 'bits' used in the calc.}
```

#### 15.10. Read Setmux Command

An RS232 "setmux ext" command is sent.

Expected to receive "EXT\_COL" as a response.

#### 15.11. Read Xrawread Command

An RS232 "xrawread 1 1" command is sent.

Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

#### 15.12. COL\_RTD Read Open via Mux

The bit count captured from the previous test is calculated into degrees C by the following formula.  
$$\text{Ohms} = \text{bits} * 1000 / 3473364$$
. 
$$C = (\text{ohms} - 1000) / 3.85$$
, where 1000 is ohms for 0 degrees Celsius and 3.85 is the ohms per degree Celsius for the specified RTD when used.

Expecting greater than 100 degrees Celsius or (>1385 ohms).

Test Limits: Lo= 100.0°C Hi= 1000.0°C

#### 15.13. Read Xrawread Command

A 1K-ohm resistor is connected between J2\_6 and J2\_7. An RS232 "xrawread 1 1" command is sent.

Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

#### 15.14. COL\_RTD Read 1000 Ohms via Mux

The bit count captured from the previous test is calculated into degrees C by the following formula.  
$$\text{Ohms} = \text{bits} * 1000 / 3473364. \quad C = (\text{ohms} - 1000) / 3.85,$$
 where 1000 is ohms for 0 degrees Celsius and 3.85 is the ohms per degree Celsius for the specified RTD when used.

Expecting approximately 0 degrees Celsius or (1000 ohms).

Test Limits: Lo= -2.0°C Hi= 2.0°C

#### 16. Read LED I and GND Via Mux

Connect a jumper from J3\_8 to J3\_7 to complete the led current path. Connect a jumper from J3\_2 to J3\_4 to simulate the LED ID jumper.

The UUT is sent via the console port a 'poke 0x500021 0xbf' command to turn on the LED circuitry. An 'led -d 100' command is sent to force current to the lowest level. Then an 'led -i 50' command is sent to force the current to the 115mA level. The UUT generated current is also routed to U14 and called LED\_I. A setmux command is sent to the UUT via the console port for U14 to select the LED\_I input and an xrawread command is sent to get the value from A/D U21. The value is returned in Bit counts that represent the A/D's output. The Test System program calculates the measured current at the low and 100mA level.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. The command 'acq -i' is sent to initialize the task. Then 'acq -x' is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example:

```
CONSOLE:> acq -i
CONSOLE:> acq -x
CONSOLE:> poke 0x500021 0xbf
CONSOLE:> led -d 100
CONSOLE:> led -i 50
CONSOLE:> setmux led
```

```
CONSOLE:> Temp Signal → I_LED
CONSOLE:> xrawread 1 1
```

| CONSOLE:> Sec. DiodeA | DiodeB | Temp Counts                                    |
|-----------------------|--------|------------------------------------------------|
| 0.00    xxxxxx        | xxxxxx | 1676838 {this is the 'bits' used in the calc.} |

##### 16.1. Read Setmux Command

An RS232 "setmux led" command is sent.  
Expected to receive "I\_LED" as a response.

##### 16.2. Read Xrawread Command

The UUT is sent a "poke 0x500021 0xbf" command to turn on the LED circuitry. An "led -d 100" command is sent to force the current to the lowest level.

An RS232 "xrawread 1 1" command is sent.

Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

##### 16.3. Lowest Current Output Via Mux

THIS DOCUMENT AND ITS CONTENTS ARE EXCLUSIVE PROPERTY OF WATERS CORPORATION, MILFORD MASSACHUSETTS. THIS DOCUMENT MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER, WITHOUT PRIOR WRITTEN PERMISSION FROM WATERS CORPORATION.

The bit count captured from the previous test is calculated into mADC by the following formula.  
Current = [ bits(.059uV/bit) ] . {to calculate bits : bits = [ (Volt) /.059uV/bit ]} .  
Expecting near zero mA or (0 Bits).  
Test Limits: Lo= 38.0mA Hi= 58.0mA

#### 16.4. Read Xrawread Command

An "led -i 50" command is sent to force the current to the mid level.  
An RS232 "xrawread 1 1" command is sent.  
Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

#### 16.5. 115mAmp Output via Mux

The bit count captured from the previous test is calculated into mADC by the following formula.  
Current = [ bits(.059uV/bit) ] : {to calculate bits : bits = [ (Volt) /.059uV/bit ]} .  
Expecting near 115mA.  
Test Limits: Lo= 100.0mA Hi= 130.0mA

The AGND voltage is measured via U14 to A/D U21. A setmux command is sent to the UUT via the console port for U14 to select the GND input and an xrawread command is sent to get the value from A/D U21. The value is returned in Bit counts that represent the A/D's output. The Test System program calculates the measured voltage.

Note: This test assumes the acqman task runs without errors. Acqman is tested in a previous test. J3\_2 is shorted to J3\_4 to simulate the LED ID jumper. The command 'acq -i' is sent to initialize the task. Then 'acq -x' is sent to suspend the task. These steps must be performed prior to testing and must not report any errors in order to run this test properly.

Actual example:

```
CONSOLE:> acq -i
CONSOLE:> acq -x
CONSOLE:> setmux gnd
```

```
CONSOLE:> Temp Signal → GND
CONSOLE:> xrawread 1 1
```

| CONSOLE:> Sec. DiodeA | DiodeB | Temp Counts                                |
|-----------------------|--------|--------------------------------------------|
| 0.00 xxxxxx           | xxxxxx | 131 {this is the 'bits' used in the calc.} |

#### 16.6. Read Setmux Command

An RS232 "setmux gnd" command is sent.  
Expected to receive "GND" as a response.

#### 16.7. Read Xrawread Command

An RS232 "xrawread 1 1" command is sent.  
Expected to receive "Temp Counts" as a response. (The bit count is also captured for the next test.)

#### 16.8. GND Voltage Via Mux

THIS DOCUMENT AND ITS CONTENTS ARE EXCLUSIVE PROPERTY OF WATERS CORPORATION, MILFORD MASSACHUSETTS.  
THIS DOCUMENT MAY NOT BE REPRODUCED IN ANY FORM WHATSOEVER, WITHOUT PRIOR WRITTEN PERMISSION FROM  
WATERS CORPORATION.

The captured bit count from the previous test is converted to volts by the following formula.  
Volts = [ bits(.3uV/bit) ] . {to calculate bits : bits = [ (Volt) /.3uV/bit ] } , where .3 is derived from  $5 / 2^{24}$  , where 5 is the A/D's full scale unipolar input and  $2^{24}$  is for a 24 bit device.  
Get GND reading from console port. Expecting zero volts or (0 Bits).  
Test Limits: Lo= -0.1VDC Hi= 0.1VDC