# Radstone PIO-2 Manual

RADSTONE
TECHNOLOGY
*Why settle for less*

**Publication No. HH681PIO2**
Rev D, March 2002

# Copyright Notice



# Trademarks



# Document History

The document revision state refers *only* to the manual, *not* the board.

| Revision | Date |
|---|---|
| Rev - | 10/92 |
| Rev A | 1/95 |
| Rev B | 4/97 |
| Rev C | 1/98 |
| Rev D | 3/02 |
| | |
| | |
| | |

Changes from the previous revision are sidelined.

# Contents

# Chapter 1 - Introduction

## Scope of this Manual

This manual describes Radstone's PIO-2 Parallel I/O Interface board. The PIO-2, in common with other Radstone boards, is available in five environmental build standards.

## Conventions Used in This Manual

This manual adopts the following conventions:

As with the 680x0, bits are numbered from 0 to *n*, where 0 is the LSB and *n* is the MSB.

Signal names follow the ANSI/VITA 1-1994 Specification. Signal names ending in "~" deonte active low signals, all others are active high.

The prefix 0x indicates a hexadecimal value following the 'C' programming language convention.

## Related Documents

ANSI/VITA 1-1994 VMEbus Specification.

IEEE Std 1101.2-1992 Standard for Mechanical Core Specifications for Conduction-Cooled Eurocards.

Radstone BIT V3 User Manual, publication number HH681BITE3.

Radstone Glossary, publication number RT5116.

BIT on Radstone PPCx User Guide, publication number RT5107.

## Features

- Fully ANSI/VITA 1-1994 VMEbus Specification compatible
- 32 digital I/O bits for monitoring and control applications
- A variety of population options
- Opto-isolation
- Input filtering and high voltage spike protection
- Six programmable 16-bit counter/timers
- Change of state monitoring for all inputs
- All I/O through the P2 connector
- VMEbus slave
- BIT facilities

# Build Styles

The PIO-2 is available in Radstone's five electrically compatible build levels. These have two basic mechanical configurations:

- Air (convection)-cooled (build levels 1 to 3) in accordance with ANSI/VITA 1-1994 specification, designed to be used in standard industrial VME chassis
- Conduction-cooled (build levels 4 and 5) in accordance with IEEE Std 1101.2-1992, for use in Radstone or third party ATR style enclosures.

In addition to these COTS configurations, PIO-2 may be supplied to meet the mechanical and thermal requirements of specific platforms with the addition of mission specific, to-type mechanics.

Radstone uses advanced thermal and mechanical design in the PCB, metal work and assembly process to build-in the required levels of ruggedness. Build level 2 and higher circuit card assemblies include conformal coating as standard.

All five styles fully support the power and versatility of the VMEbus, so no matter how large or diversified your project, absolute compatibility is assured at all stages of development.

A brief description of each build style follows:

### Level 1

Intended for use in benign environments, level 1 also provides the ideal cost effective method of complete system development. The level 1 assembly comprises a double-Eurocard size printed wiring board with high quality commercial (plastic encapsulated) components.

As software compatibility throughout the build styles is absolute, a system intended for final implementation in a severe tactical environment can be developed and debugged at low cost, switching over to target style only in the final stages of system integration.

### Level 2

As Level 1, but tested in manufacture to provide an extended operating range.

### Level 3

Level 3 boards are intended for applications that have extended temperature, shock and vibration requirements, but can be served by conventional, forced-air-cooled, racking systems. These rugged boards comprise a double-Eurocard size printed wiring board fitted with wide temperature range, industrial grade components.

### Level 4

Designed primarily for use in sealed ATR chassis and other conduction-cooled environments, the level 4 board features wide temperature range, industrial grade devices, an integral thermal management layer. It also incorporates a central stiffening bar for additional strength.

Cooling is achieved through conduction of heat from the thermal management layer to the cold wall of the rack to which the boards are secured by screw driven wedgelocks. Level 4 boards are temperature characterised during manufacture.

### Level 5

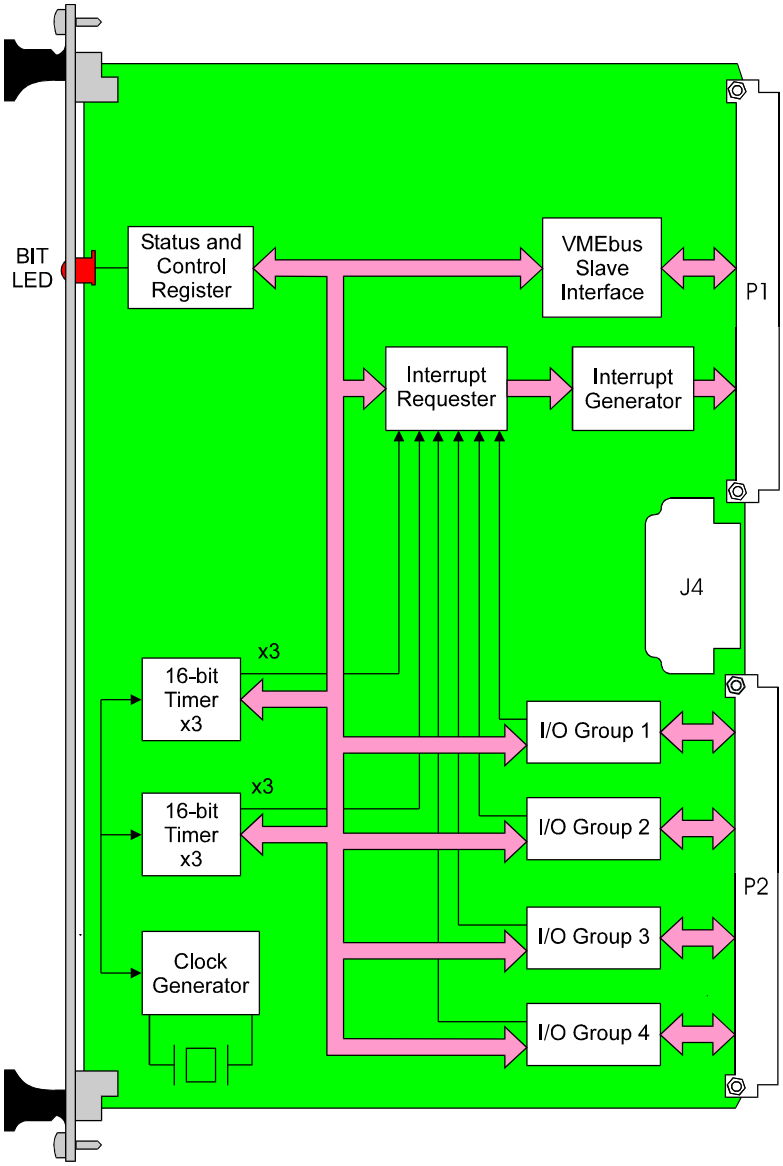
As Level 4, but tested in manufacture to provide an extended operating range.

See Chapter 2 for more details of the environmental specifications of each build style.

# Functional Overview

The PIO-2 provides an efficient means of monitoring and controlling remotely located machinery and equipment from a VMEbus computer system. Solid state relays permit the PIO-2 to switch low power DC voltages to provide fully isolated control of external relays or amplifiers of electro-mechanical systems. This offers the system designer the flexibility to interface with the real world in such application areas as Naval machinery control, vehicle systems control, power plants, avionics and many others.

**Figure 1-1. PIO-2 Functional Block Diagram**




## VMEbus Interface

The PIO-2 is addressed as a VMEbus slave, occupying a block of locations in A24 address space. Either 8 or 16 bit data transfers may be used between the VMEbus host processor and the PIO-2 to transfer status, control and data. The PIO-2 is a VME interrupter with interrupts generated on any change of state of an input bit or by the expiration of any one of the timer/counters. The basic Status/ID interrupt word is set into the PIO-2 by the VMEbus host processor and the source of interrupt is encoded to permit simple identification.

## I/O Connections

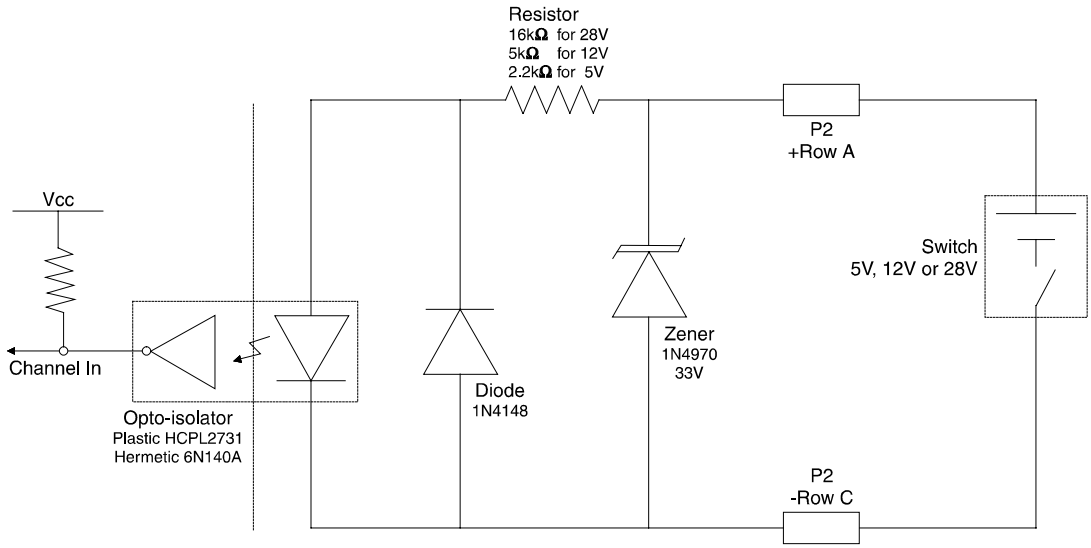
All I/O connections to the PIO-2 are made through the system backplane to the user I/O pins of the P2 connector. Each I/O bit is assigned 2 pins, with no common connections. Input bits are always optically isolated and must be driven from an external DC current source.

## I/O Cells

The PIO-2 has 32 identical I/O cells, arranged in 4 groups of 8 cells. The configuration of these groups depends on the board variant. Input and output components are fitted to, or omitted from the cells in the groups to give the required configuration of inputs and outputs. The three basic build configurations are:

- Input only
- Output only
- Monitored output

**Figure 1-2. Circuit of Input Only Cell**

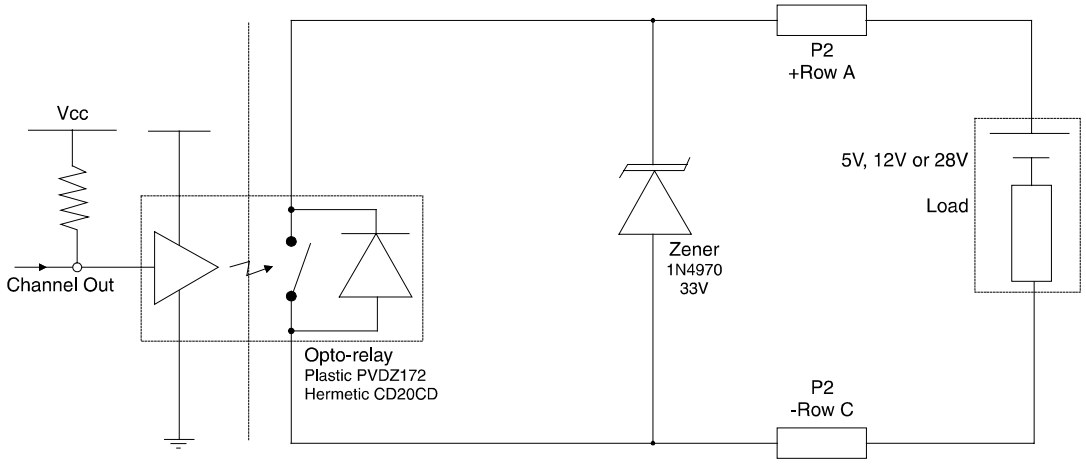



The diode protects the opto-coupler from reverse voltages caused by contact bounce in the load when the protective zener has not been fitted.

**Note:** The zener diode is only fitted on level 5 boards.

The resistor value depends on the configured load voltage (+5V, +12V or +28V) and is chosen to give a current of approximately 1.6 mA through the opto-coupler. The input terminals are polarised so that the opto-coupler and transorb function correctly.

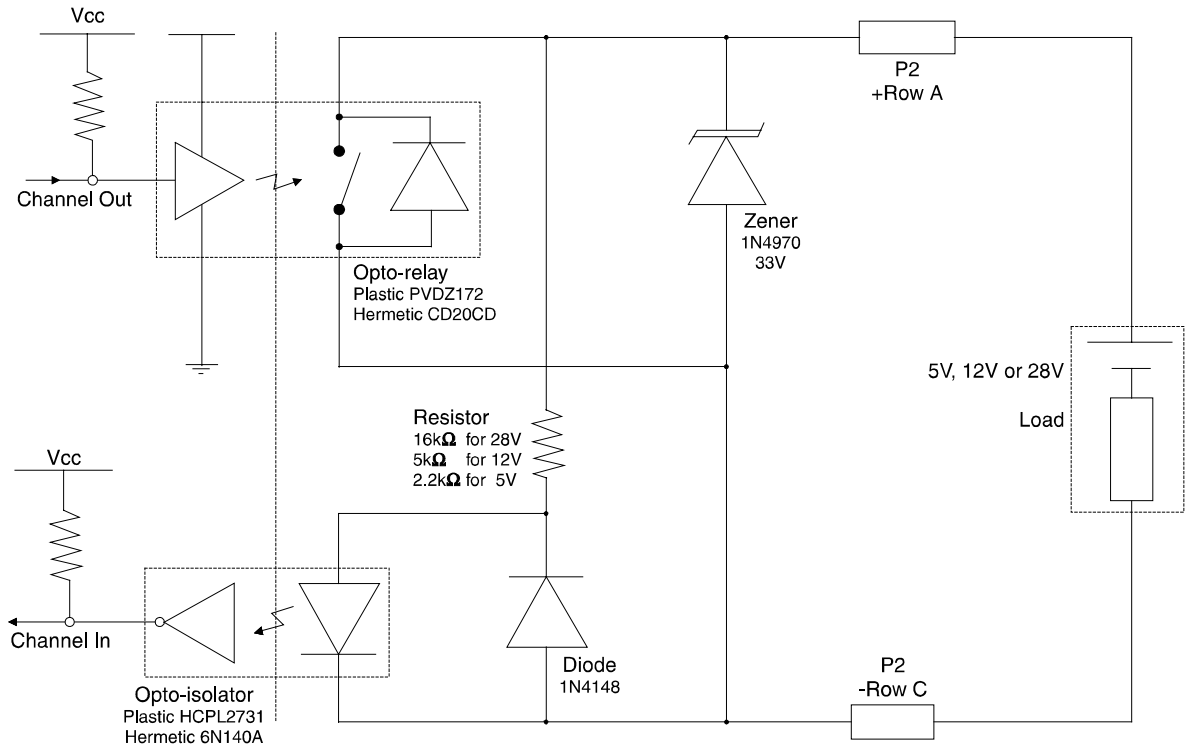
**Figure 1-3. Circuit of Output Only Cell**




The zener, rated at 33V, is included to protect the relay from high back-EMF generated by inductive loads.

**Note:** The zener diode is only fitted on level 5 boards.

**Figure 1-4. Circuit for Monitored Output Cell**




This type of cell is formed by fitting the components for both Input and Output cells. The input section of the monitored output circuit works in the opposite way to the input-only circuit. In the quiescent state, a small current (1.6 mA), large enough to operate the opto-coupler but not enough to operate *most* loads, is drawn by the opto-coupler. Consideration should be given to this when operating sensitive loads such as some solid state relays.

The operation of the opto-relay short-circuits the output terminals, the load is energised and the opto-coupler is turned off. This means that the data read in from the opto-coupler is opposite to that returned by the input-only circuit.

**Note:** The zener diode is only fitted on level 5 boards.

## Counter/Timers

The PIO-2 has six 16-bit programmable timers. These timers are independent of the I/O bits and may be used in any system application that requires enhanced counter/timer capability. All timers are fed from a common on-board crystal clock running at 5 MHz, divided to give a timer range up to 1 second. The counter/timers may be programmed by the host VMEbus CPU to generate regular clock “ticks” or to time-out after a pre-programmed time period. Each counter/timer, when it expires, makes a local interrupt request, which the PIO-2 then presents as a VMEbus interrupt. The VMEbus interrupt level and the top 4 bits of the Status/ID byte are preset by the host VMEbus CPU during the initialization of the PIO-2. The on-board interrupt generator prioritizes local interrupt requests and adds the lower 4 bits of the Status/ID byte to identify the source of the interrupt request uniquely.

## Input Change of State

Input bits can be programmed either to ignore changes of state or to generate an interrupt request as a result of the following conditions:

- A high to low transition
- A low to high transition
- On either transition

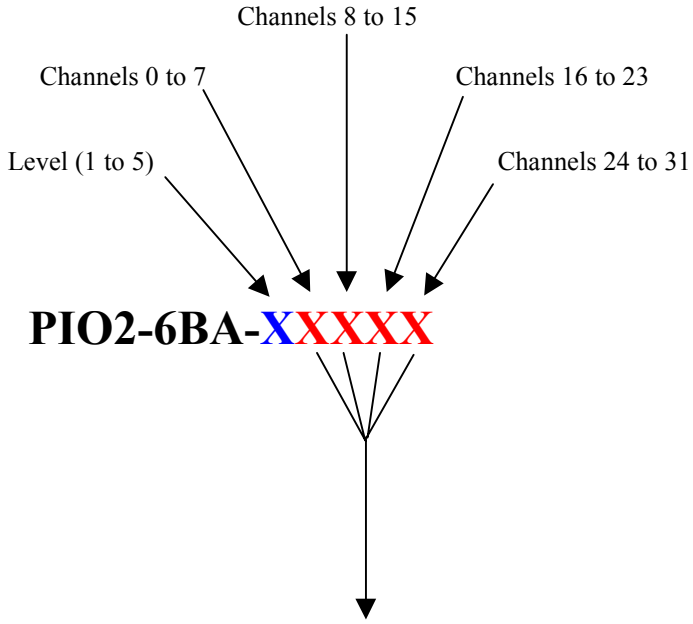
Whenever an interrupt is generated, the Interrupt Status Registers (ISRs) are frozen until they are read over the VMEbus during the interrupt service routine. Any input changes that are detected between the freezing and the reading of the ISRs are stored in temporary registers and then loaded into the ISRs after they have been read, to generate further interrupts. This ensures that no input state transition goes unnoticed. Input filtering prevents spurious detections caused by noise or contact bounce. The detection of a change of state in a particular group of 8 bits causes a local interrupt request and the generation of a VMEbus interrupt. The lower 4 bits of the Status/ID byte identify the group in which the change was detected. The VMEbus host processor may then read the ISRs to determine which bit or bits have changed.

## Built In Test

The PIO-2 is tested, on power-up, over the VMEbus by a host CPU card. Radstone provides this test package as part of Built In Test (BIT) to allow a complete, configured BIT system to be constructed. The BIT firmware is located on the VMEbus master processor card and is used to test the system configuration with the PIO-2 card(s) installed.

## PIO-2 Numbering System

The PIO-2 is numbered as follows:




| | 5V | 12V | 28V | Other |
|---|---|---|---|---|
| Not fitted | 0 | 0 | 0 | 0 |
| Input | 1 | 2 | 3 | 4 |
| Output | 5 | 5 | 5 | 5 |
| Input/Output | 6 | 7 | 8 | 9 |

For example **PIO2-6BA-33560** = Level 3 board, Channels 0 to 7 28V inputs, Channels 8 to 11 outputs, Channels 12 to 23 5V input/outputs, Channels 24 to 32 not fitted.

# Chapter 2 - Specifications

This section details the specifications common to all versions of the PIO-2.

## General Specifications

| | |
|---|---|
| **VMEbus Compliance** | ANSI/VITA 1-1994 Specification<br>A24<br>D16, D08 (EO)<br>Supports Read-Modify-Write (RMW) and<br>Supports Address Modifier Codes 0x39, 0x3A, 0x3D, 0x3E<br>Release on Register Access (RORA) interrupter. |
| **Front Panel** | Single width VME containing a single LED |

## Electrical Specifications

Two 96-way DIN41612 compatible connectors that mate with rack mounted backplane connectors form the electrical connection to the board.

### Current Consumption

| Current | At +5V |
|---|---|
| Typical | 0.6A |
| Maximum | 0.9A |

### Power Dissipation

| VME +5V | | Remote Supply per Active Input | Remote Supply per Active Output at 1A load current |
|---|---|---|---|
| Levels 1, 2 | 3W | 50 mW @ 28V<br>20 mW @12V<br>8 mW @5V | 415 mW |
| Levels 3 to 5 | 3W | 50 mW @ 28V<br>20 mW @12V<br>8 mW @5V | 500 mW |

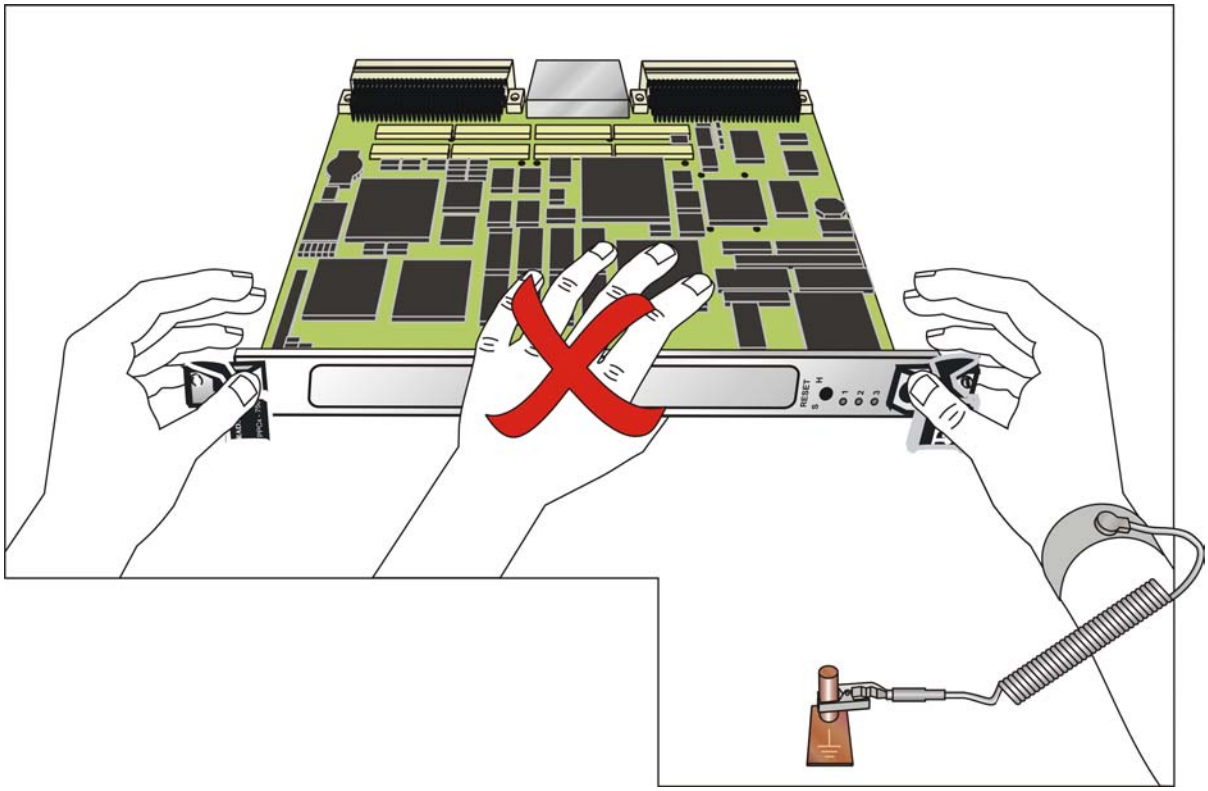
### Voltage Supply Requirements

+5V ±0.25V DC total excursion, including all transients.

Vripple (5V) = 50 mV RMS (maximum) contained within the total excursion.

# Safety Requirements

**WARNING**
**Do not exceed the maximum rated input voltages or apply reversed bias to the assembly.**
**If such conditions occur, toxic fumes may be produced due to the destruction of components**

# Mechanical Specifications

## Weight

The approximate weights of the different styles of PIO-2 are as follows:

| | |
|---|---|
| Level 4 and 5 | ≈ 450g |
| Levels 1 to 3 | ≈ 375g |

## Dimensions

Figures 2-1 and 2-2 show basic VME board dimensions in millimetres with inches (in parentheses) for guidance only. All build standards of the PIO-2 board are ANSI/VITA 1-1994 VMEbus Specification compatible. This allows all styles of assembly to be fitted into any commercial VMEbus development chassis. Level 4 and 5 boards comply with IEEE Std 1101.2-1992.

**Figure 2-1. VME Dimensions for Build Levels 1 to 3**

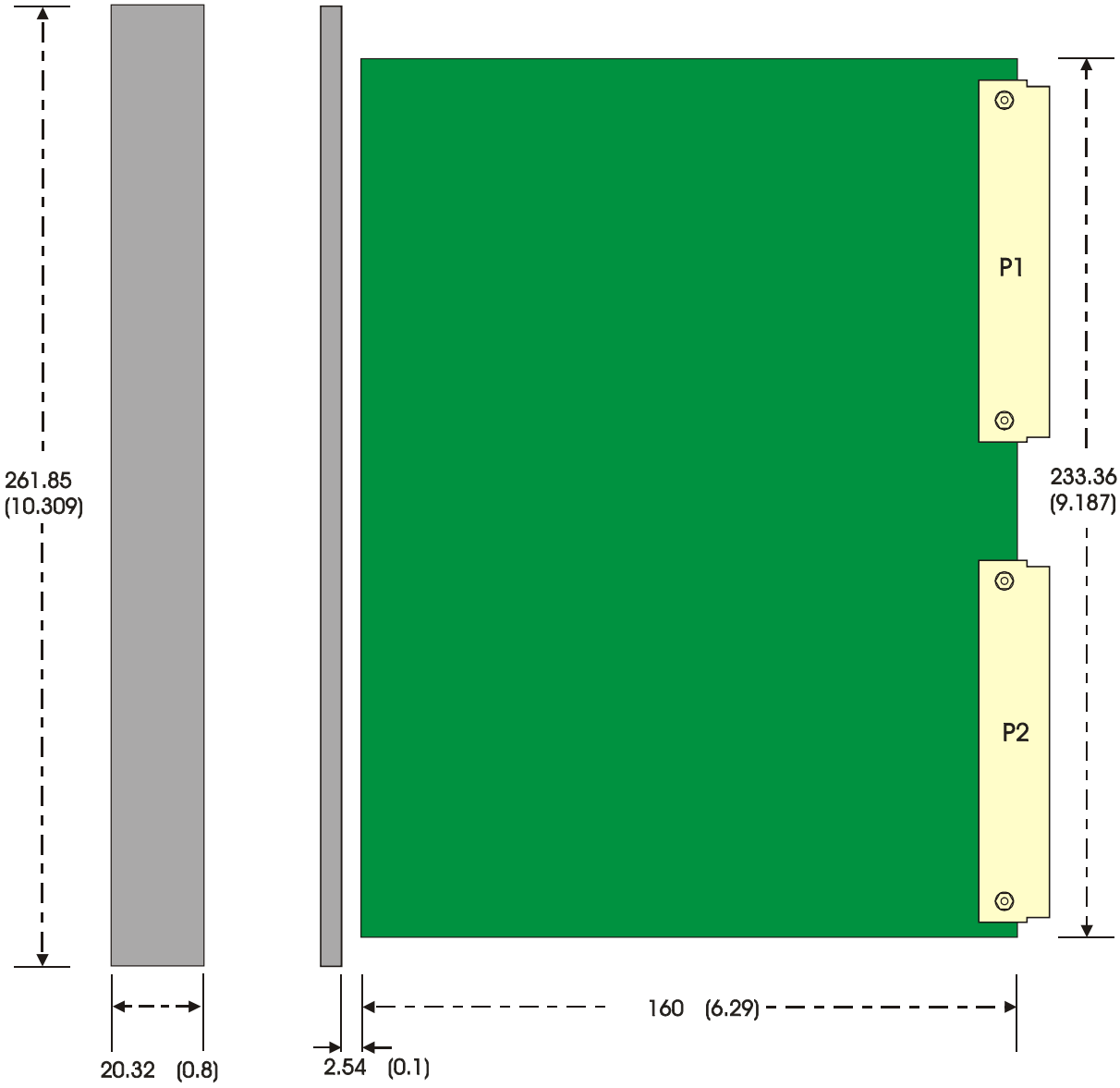
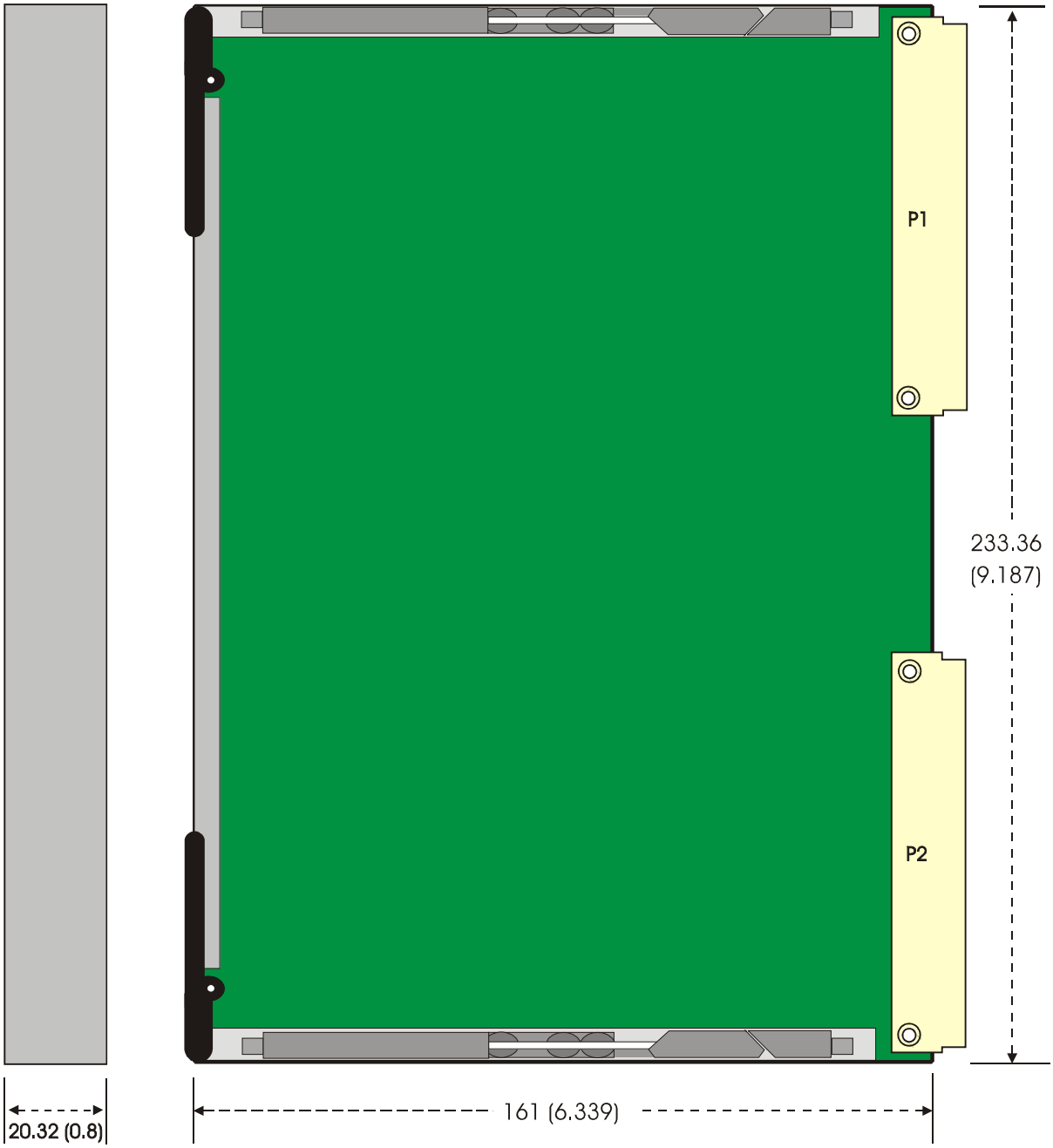


**Figure 2-2. VME Dimensions for Build Levels 4 and 5**



# Environmental Specification

## Convection-cooled Boards

| Build Style | Temperature (°C) | Low Pressure (Feet) | Vibration | Shock | Humidity/ Salt Fog | Comments |
|---|---|---|---|---|---|---|
| **Standard (Level 1)** | **Operating:** 0 to +55 with airflow of 300 feet/minute. **Storage:** -40 to +85 **Shock**: 10°C/minute over range | **Operating:** 15,000 **Storage:** 50,000 | **Random:** 0.002$g^2$/Hz from 10 to 2000 Hz **Sine:** 2g from 10 to 500 Hz | 20g peak sawtooth, 11ms duration | Up to 95% RH | Commercial grade cooled by forced air, for use in benign environments and software development applications. Optional conformal coating |
| **Extended Temp (Level 2)** | **Operating:** -20 to +65 with airflow of 300 feet/minute **Storage:** -40 to +85 **Shock**: 10°C/minute over range | **Operating:** 15,000 **Storage:** 50,000 | **Random:** 0.002$g^2$/Hz from 10 to 2000 Hz **Sine:** 2g from 10 to 500 Hz | 20g peak sawtooth, 11ms duration | Up to 95% RH with varying temperature. 10 cycles, 240 hours | As Standard but conformally coated and temperature characterised. |
| **Rugged Air-cooled (Level 3)** | **Operating:** -40 to +75 with airflow of 600 feet/minute **Storage:** -50 to +100 **Shock**: 10°C/minute over range | **Operating:** 15,000 **Storage:** 50,000 **Rapid Decompression:** 0 to 50,000 | **Random:** 0.04$g^2$/Hz with a flat response to 1000 Hz. 6db/Octave roll-off from 1000 to 2000 Hz. **Sine:** 5g from 5 to 2000 Hz | 20g peak sawtooth, 11ms duration. Bench handling | Up to 95% RH with varying temperature. 10 cycles, 240 hours. 5% salt 48 hours | Wide temperature rugged, cooled by forced air. Conformally coated for additional protection. |

## Conduction-cooled Boards

| Build Style | Temperature (°C) | Low Pressure (Feet) | Vibration | Shock | Humidity/ Salt Fog | Comments |
|---|---|---|---|---|---|---|
| **Rugged Conduction-cooled (Level 4)** | **Operating:** -40 to +75 at the thermal interface **Storage:** -50 to +100 **Shock**: 10°C/minute over range | **Operating:** 70,000 **Storage:** 70,000 **Rapid Decompression:** 0 to 70,000 | **Random:** 0.1$g^2$/Hz with a flat response to 1000 Hz. 6db/Octave roll-off from 1000 to 2000 Hz. **Sine:** 5g from 5 to 2000 Hz | 40g peak sawtooth, 11ms duration. Bench handling | Up to 95% RH with varying temperature. 10 cycles, 240 hours. 5% salt 48 hours | Mechanically compliant with IEEE 1101.2-1992. Designed for severe environment applications with high levels of shock and vibration, small space envelope and restricted cooling supplies. Conformally-coated as standard. Optional ESS. |
| **Rugged Conduction -cooled (Level 5)** | **Operating:** -40 to +85 at the thermal interface **Storage:** -50 to +100 **Shock**: 10°C/minute over range | **Operating:** 70,000 **Storage:** 70,000 **Rapid Decompression:** 0 to 70,000 | **Random**: 0.1$g^2$/Hz with a flat response to 1000 Hz. 6db/Octave roll-off from 1000 to 2000 Hz. **Sine:** 5g from 5 to 2000 Hz | 40g peak sawtooth, 11ms duration. Bench handling | Up to 95% RH with varying temperature. 10 cycles, 240 hours. 5% salt 48 hours | Mechanically compliant with IEEE 1101.2-1992. Designed for severe environment applications with high levels of shock and vibration, small space envelope and restricted cooling supplies. Conformally-coated as standard. Optional ESS. |

# Reliability

Using MIL-HBK-217F notice 1 as a data base and the parts count method, the following estimates have been made of the MTBF for the PIO-2.

## Configured as Input Board

| Environment | Temp. (°C) | Level 2 (-21111) | | Level 4 (-41111) | |
|---|---|---|---|---|---|
| | | Fail Rate (FPMH) | MTBF (Hours) | Fail Rate (FPMH) | MTBF (Hours) |
| Ground Benign | 30 | 3.807148 | 262,664 | 1.700408 | 588,094 |
| Ground Fixed | 40 | 6.79689 | 147,126 | 4.98312 | 200,677 |
| Ground Mobile | 45 | 11.41862 | 87,576 | 8.88552 | 112,543 |
| Naval Sheltered | 40 | 8.27089 | 120,906 | 7.36302 | 135,814 |
| Naval Unsheltered | 45 | 12.72184 | 78,605 | 13.80574 | 72,434 |
| Airborne Inhabited Cargo | 55 | 12.58174 | 79,480 | 12.0295 | 83,129 |
| Airborne Inhabited Fighter | 55 | 13.46782 | 74,251 | 20.61802 | 48,501 |
| Airborne Uninhabited Cargo | 70 | 23.51722 | 42,522 | 24.06652 | 41,551 |
| Airborne Uninhabited Fighter | 70 | 26.63558 | 37,544 | 44.13698 | 22,657 |
| Airborne Rotary Wing | 55 | 16.35648 | 61,138 | 21.26868 | 47,017 |
| Space Flight | 30 | 3.77227 | 265,092 | 1.541514 | 648,713 |
| Missile Flight | 45 | 12.28732 | 81,385 | 11.55612 | 86,534 |
| Missile Launch | 55 | 19.414 | 51,509 | 28.219 | 35,437 |

## Configured as Output Board

| Environment | Temp. (°C) | Level 2 (-25555) | | Level 4 (-45555) | |
|---|---|---|---|---|---|
| | | Fail Rate (FPMH) | MTBF (Hours) | Fail Rate (FPMH) | MTBF (Hours) |
| Ground Benign | 30 | 3.027184 | 330,340 | 1.4894 | 671,411 |
| Ground Fixed | 40 | 6.34272 | 157,661 | 3.56328 | 280,640 |
| Ground Mobile | 45 | 14.6604 | 68,211 | 7.45288 | 134,176 |
| Naval Sheltered | 40 | 11.44704 | 87,359 | 6.447372 | 155,102 |
| Naval Unsheltered | 45 | 39.15198 | 25,541 | 14.81342 | 67,506 |
| Airborne Inhabited Cargo | 55 | 11.886 | 84,133 | 6.909372 | 144,731 |
| Airborne Inhabited Fighter | 55 | 16.0057 | 62,478 | 8.86858 | 112,758 |
| Airborne Uninhabited Cargo | 70 | 24.77843 | 40,358 | 12.08786 | 82,728 |
| Airborne Uninhabited Fighter | 70 | 164.1383 | 6,092 | 48.13314 | 20,776 |
| Airborne Rotary Wing | 55 | 32.80141 | 30,486 | 15.91902 | 62,818 |
| Space Flight | 30 | 2.717338 | 386,007 | 1.38257 | 723,291 |
| Missile Flight | 45 | 28.90645 | 34,594 | 11.75686 | 85,057 |
| Missile Launch | 55 | 46.43864 | 21,534 | 22.55308 | 44,340 |

# Chapter 3 - Configuration

This chapter provides the information required to configure the PIO-2. The board is delivered with push-on jumper links, but for rugged or Military applications, you are recommended to connect the posts using wire wraps.

**Note:** To maintain the environmental integrity of cards that have been conformally coated during manufacture, user-made wire-wrap links should be given a conformal coating.

**Figure 3-1. Link Locations**

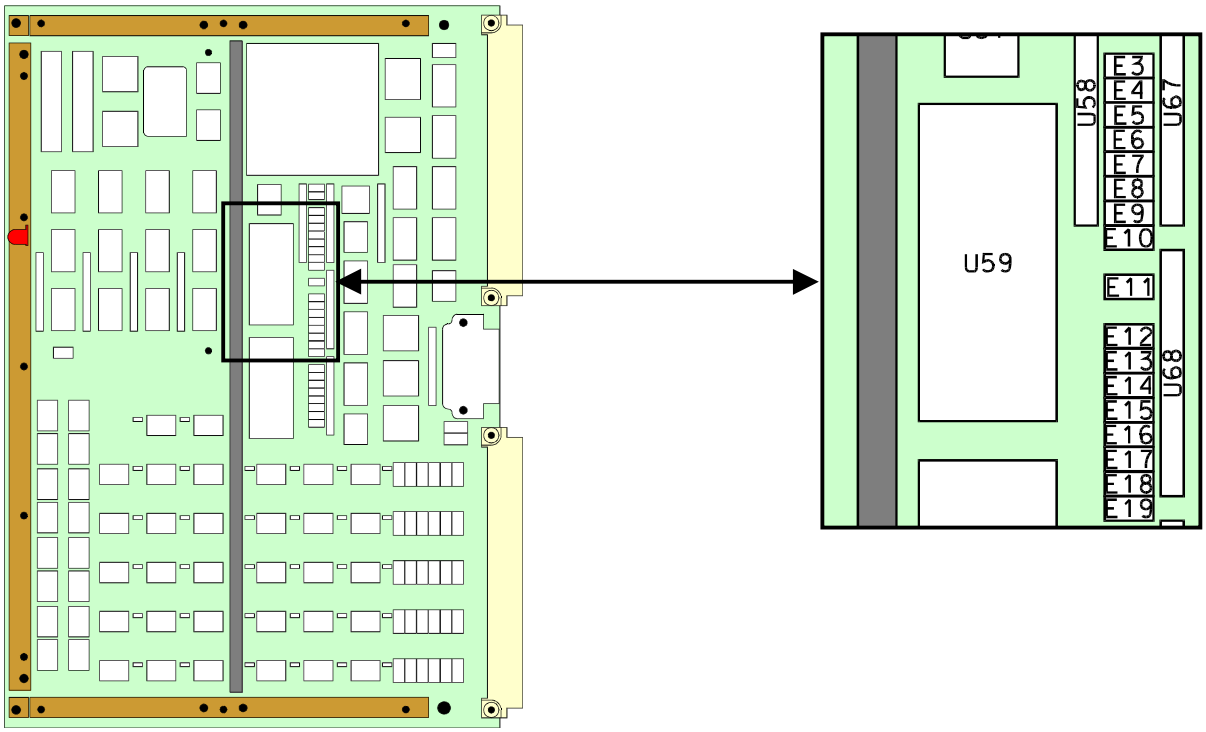
# Setting the PIO-2 Base Address

To set the PIO-2 base address using on-board links, link E11 **must be fitted**.

If link E11 is fitted, links E3 to E10 and E12 to E19 set the VME address bits A23 to A8 as follows:

| A23 | A22 | A21 | A20 | A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E18 | E16 | E14 | E12 | E13 | E15 | E17 | E19 | E10 | E8 | E6 | E4 | E3 | E5 | E7 | E9 |

Fitting a link makes the corresponding address bit a “0”; leaving the link out makes the corresponding address bit a “1”. The granularity sets the board base address to a 256-byte boundary.

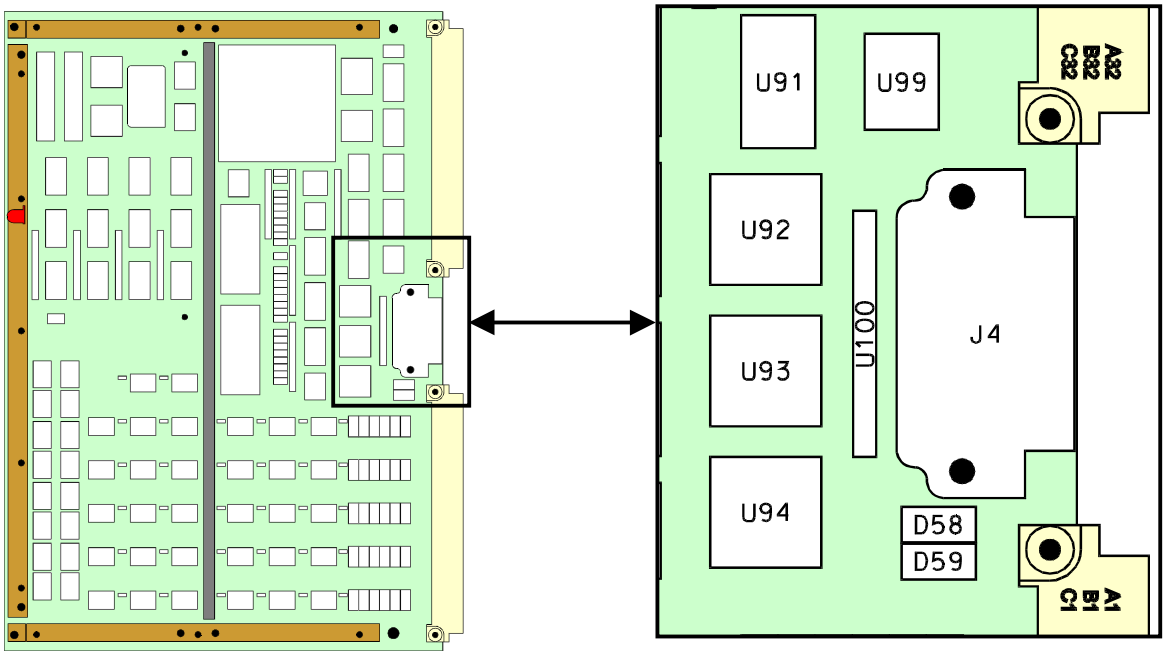


If E11 is not fitted, the address decoding is done by U94, which compares addresses against a code set via pins 17 (the LSB) to 20 (the MSB) of the J4 connector as shown in the table below. This is only intended to be used on level 5 boards where Mean Time To Repair (MTTR) requirements rule out configuration via numerous links.

**Table 3-1. Code Set via J4**

| J4 Pin | | | | Code | Address |
|---|---|---|---|---|---|
| 20 | 19 | 18 | 17 | | |
| 0 | 0 | 0 | 0 | 0 | 0x000000 |
| 0 | 0 | 0 | 1 | 1 | 0x100000 |
| 0 | 0 | 1 | 0 | 2 | 0x200000 |
| 0 | 0 | 1 | 1 | 3 | 0x300000 |
| 0 | 1 | 0 | 0 | 4 | 0x400000 |
| 0 | 1 | 0 | 1 | 5 | 0x500000 |
| 0 | 1 | 1 | 0 | 6 | 0x600000 |
| 0 | 1 | 1 | 1 | 7 | 0x700000 |
| 1 | 0 | 0 | 0 | 8 | 0x800000 |
| 1 | 0 | 0 | 1 | 9 | 0x900000 |
| 1 | 0 | 1 | 0 | A | 0xA00000 |
| 1 | 0 | 1 | 1 | B | 0xB00000 |
| 1 | 1 | 0 | 0 | C | 0xC00000 |
| 1 | 1 | 0 | 1 | D | 0xD00000 |
| 1 | 1 | 1 | 0 | E | 0xE00000 |
| 1 | 1 | 1 | 1 | F | 0xF00000 |

If none of the addresses listed above meet your requirement, please consult Radstone Technology for alternatives.



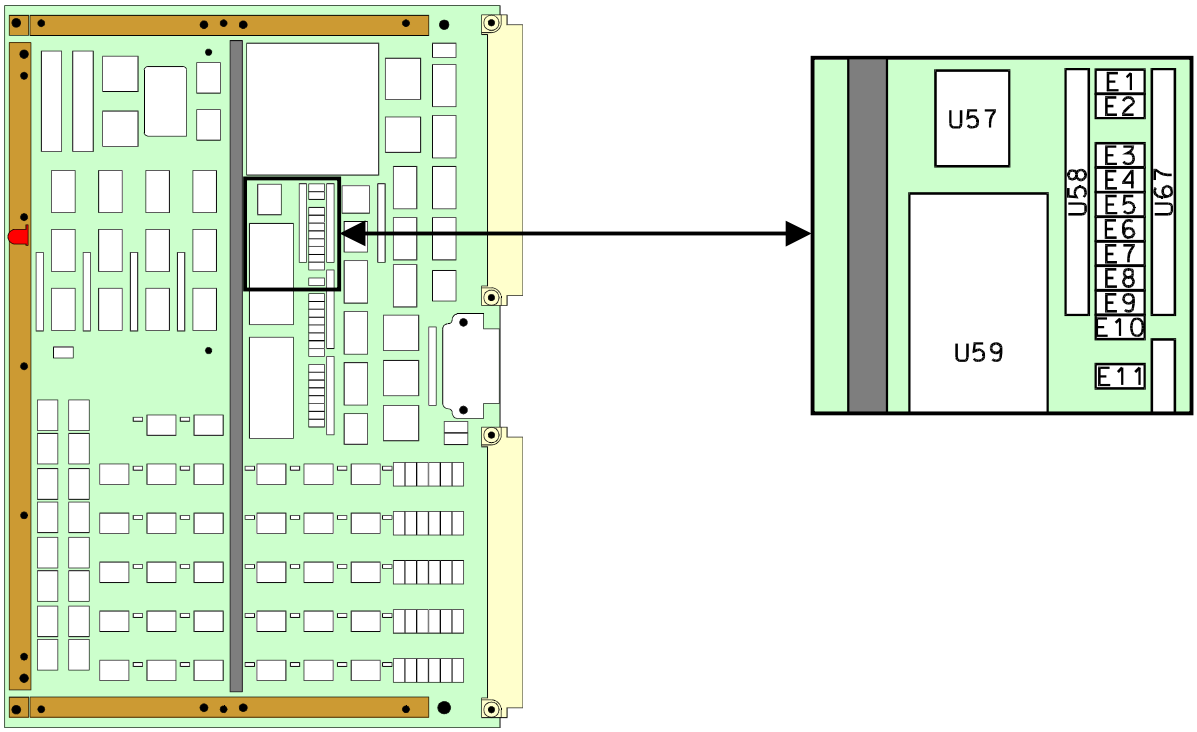
# Debounce Clock Selection

Links E2 and E1 select the clock that is used to debounce input signals from the opto-couplers, as shown in the following table:

**Table 3-2. Debounce Clock Selection**

| E2 | E1 | Debounce Clock Rate | Response Time† (worst case) |
|---|---|---|---|
| Out | Out | 4.76 Hz | 420 ms |
| Out | In | 76 Hz | 27 ms |
| In | Out | 1220 Hz | 1.7 ms |
| In | In | 19.53 kHz | 160 µs |

† The measured total time from the opto-coupler current flow to an interrupt generated on the VMEbus (if enabled).



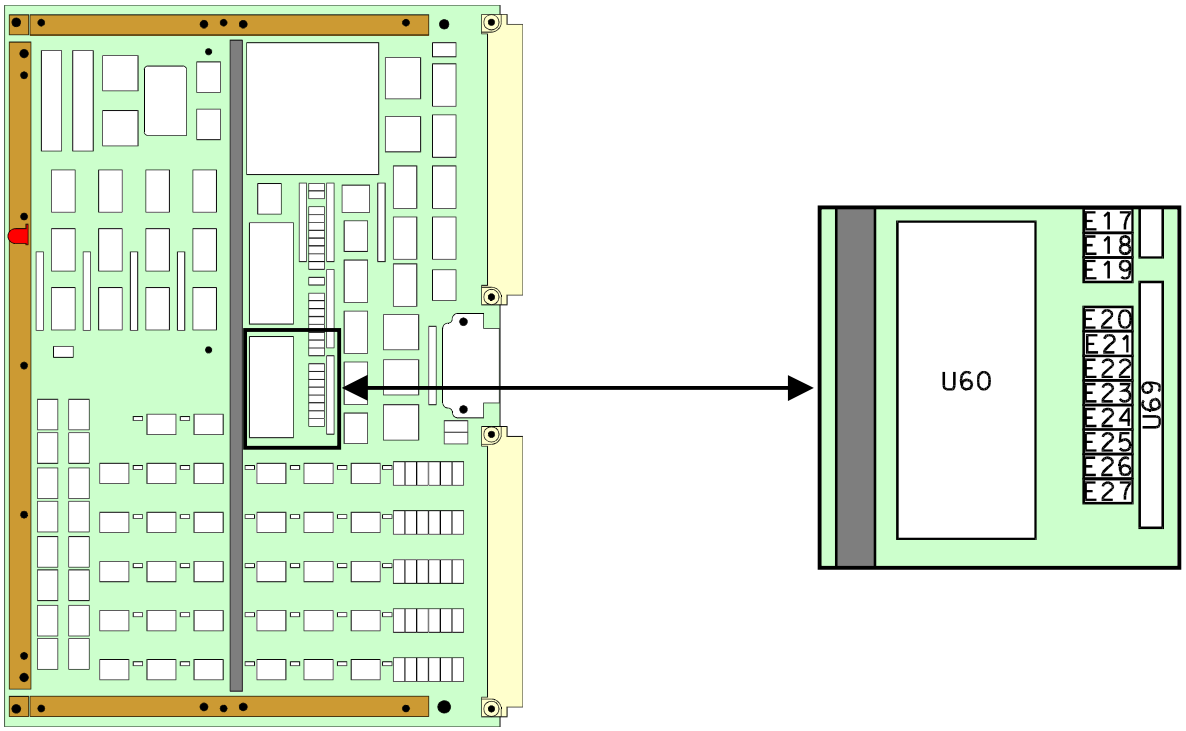
# ID Register

The user can set up a value on links E20, E21, E22, E23, E25 and E27. Any combination of these links can be fitted, as they have no effect on the function of the board, but can be used to identify the particular configuration of a PIO-2 to the host (how many inputs and outputs are fitted, voltage range etc.). The mapping of values to configuration is user dependent. The value set up by the links can be read by the host in the ID register at VME address BASEADD+0x30, where a '0' bit means that the corresponding link is fitted and a '1' means that the corresponding link is not fitted.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| E2 | E1 | E22 | E20 | E27 | E25 | E23 | E21 |
| E2 and E1 read-back | | User defined link area | | | | | |

Bits 7 and 6 can be used to read back the state of links E2 and E1 respectively. See the previous description of links E2 and E1.

**Note:** **Do *not*** use links E24 and E26.



## Connector Pinouts

In Table 3.3, the signals enclosed in braces { } are not active on PIO-2. However they are passed on to the next VME slot.

**Table 3-3. VME Interface (P1 Connector and P2 Connector Row B)**

| Pin Number | P1 Row a | P1 Row b | P1 Row c | P2 Row b |
|---|---|---|---|---|
| 1 | D00 | BBSY~ | D08 | +5V |
| 2 | D01 | BCLR~ | D09 | GND |
| 3 | D02 | ACFAIL~ | D10 | Reserved |
| 4 | D03 | {BG0IN~} | D11 | A24 |
| 5 | D04 | {BG0OUT~} | D12 | A25 |
| 6 | D05 | {BG1IN~} | D13 | A26 |
| 7 | D06 | {BG1OUT~} | D14 | A27 |
| 8 | D07 | {BG2IN~} | D15 | A28 |
| 9 | GND | {BG2OUT~} | GND | A29 |
| 10 | SYSCLK | {BG3IN~} | SYSFAIL~ | A30 |
| 11 | GND | {BG3OUT~} | BERR~ | A31 |
| 12 | DS1~ | BR0~ | SYSRESET~ | GND |
| 13 | DS0~ | BR1~ | LWORD~ | +5V |
| 14 | WRITE~ | BR2~ | AM5 | D16 |
| 15 | GND | BR3~ | A23 | D17 |
| 16 | DTACK~ | AM0 | A22 | D18 |
| 17 | GND | AM1 | A21 | D19 |
| 18 | AS~ | AM2 | A20 | D20 |
| 19 | GND | AM3 | A19 | D21 |
| 20 | IACK~ | GND | A18 | D22 |
| 21 | IACKIN~ | SERCLK | A17 | D23 |
| 22 | IACKOUT~ | SERDAT | A16 | GND |
| 23 | AM4 | GND | A15 | D24 |
| 24 | A07 | IRQ7~ | A14 | D25 |
| 25 | A06 | IRQ6~ | A13 | D26 |
| 26 | A05 | IRQ5~ | A12 | D27 |
| 27 | A04 | IRQ4~ | A11 | D28 |
| 28 | A03 | IRQ3~ | A10 | D29 |
| 29 | A02 | IRQ2~ | A09 | D30 |
| 30 | A01 | IRQ1~ | A08 | D31 |
| 31 | -12V | +5VSTDBY | +12V | GND |
| 32 | +5V | +5V | +5V | +5V |

**Table 3-4. PIO-2 Interface (P2 Connector)**

This table details the PIO-2 function allocated to each pin.

| Pin | Row A | Row C |
|---|---|---|
| 1 | CH0+ | CH0- |
| 2 | CH1+ | CH1- |
| 3 | CH2+ | CH2- |
| 4 | CH3+ | CH3- |
| 5 | CH4+ | CH4- |
| 6 | CH5+ | CH5- |
| 7 | CH6+ | CH6- |
| 8 | CH7+ | CH7- |
| 9 | CH8+ | CH8- |
| 10 | CH9+ | CH9- |
| 11 | CH10+ | CH10- |
| 12 | CH11+ | CH11- |
| 13 | CH12+ | CH12- |
| 14 | CH13+ | CH13- |
| 15 | CH14+ | CH14- |
| 16 | CH15+ | CH15- |
| 17 | CH16+ | CH16- |
| 18 | CH17+ | CH17- |
| 19 | CH18+ | CH18- |
| 20 | CH19+ | CH19- |
| 21 | CH20+ | CH20- |
| 22 | CH21+ | CH21- |
| 23 | CH22+ | CH22- |
| 24 | CH23+ | CH23- |
| 25 | CH24+ | CH24- |
| 26 | CH25+ | CH25- |
| 27 | CH26+ | CH26- |
| 28 | CH27+ | CH27- |
| 29 | CH28+ | CH28- |
| 30 | CH29+ | CH29- |
| 31 | CH30+ | CH30- |
| 32 | CH31+ | CH31- |

# Chapter 4 - Registers

## Register Map

PIO-2 is driven via the set of registers shown below. These are mapped onto the VMEbus at the board base address, BASEADD, as set up by either on-board links or the P4 connector. The register map is shadowed 4 times in the first 256 bytes of address space.

| Offset From BASEADD | Function | Access |
|---|---|---|
| 0x00 | Data, channels 0 to 7 | Read/Write |
| 0x01 | Data, channels 8 to 15 | Read/Write |
| 0x02 | Data, channels 16 to 23 | Read/Write |
| 0x03 | Data, channels 24 to 31 | Read/Write |
| 0x04 to 0x07 | Not used | |
| 0x08 | Interrupt status, channels 0 to 7 | Read Only |
| 0x09 | Interrupt status, channels 8 to 15 | Read Only |
| 0x0A | Interrupt status, channels 16 to 23 | Read Only |
| 0x0B | Interrupt status, channels 24 to 31 | Read Only |
| 0x0C | Interrupt status, counters 0 to 5 | Read Only |
| 0x0D to 0x0F | Not used | |
| 0x10 | Interrupt mask set, channels 0 to 3 | Write Only |
| 0x11 | Interrupt mask set, channels 4 to 7 | Write Only |
| 0x12 | Interrupt mask set, channels 8 to 11 | Write Only |
| 0x13 | Interrupt mask set, channels 12 to 15 | Write Only |
| 0x14 | Interrupt mask set, channels 16 to 19 | Write Only |
| 0x15 | Interrupt mask set, channels 20 to 23 | Write Only |
| 0x16 | Interrupt mask set, channels 24 to 27 | Write Only |
| 0x17 | Interrupt mask set, channels 28 to 31 | Write Only |
| 0x18 | BIT/VME Interrupt Control | Write Only |
| 0x19 | VME Interrupt Vector | Read/Write |
| 0x1A to 0x1F | Not used | |
| 0x20 | Counter 0 | Read/Write |
| 0x22 | Counter 1 | Read/Write |
| 0x24 | Counter 2 | Read/Write |
| 0x26 | Control Word Register 0 | Write Only |
| 0x28 | Counter 3 | Read/Write |
| 0x2A | Counter 4 | Read/Write |
| 0x2C | Counter 5 | Read/Write |
| 0x2E | Control Word Register 1 | Write Only |
| 0x30 | ID Register | Read Only |
| 0x31 to 0x3F | Not used | |

These registers may be accessed individually (as bytes) or in pairs (as words). Any attempt to access an unused location is bus errored by the PIO-2.

Due to the way in which the VMEbus organises its addresses, even addresses end up on the most significant data bits (D8 to D15), and odd addresses are on the least significant data bits (D0 to D7). PIO-2 always routes data bit 0 to channel 8 (or channel 24), data bit 1 to channel 9 (or channel 25), and so on up to data bit 15 to channel 7 (or channel 23). This leads to a difference between byte and word accesses to PIO-2, in that the byte order for word accesses is the reverse of what you might expect.

**Examples:**

1. To operate the relay on channel 3 using a byte access:

   write 0x08 to BASEADD + 0x0

2. To operate the relay on channel 14 using a byte access:

   write 0x40 to BASEADD + 0x1

**Note**: As all the registers on PIO-2 are implemented in a Logic Cell Array (LCA), there is an initialization period following power-on, where the board does not respond to a VMEbus access. This period lasts for approximately 100ms after the removal of SYSRESET~ on the VMEbus.

# Data Registers 0 to 3 (Offsets 0 to 3)

| Register | 7 | | | | | | | 0 | Offset |
|---|---|---|---|---|---|---|---|---|---|
| 0 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 | 0 |
| 1 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | 1 |
| 2 | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 | 2 |
| 3 | CH31 | CH30 | CH29 | CH28 | CH27 | CH26 | CH25 | CH24 | 3 |

There are four 8-bit read/write registers that give access to the 32 I/O channels by reading from or writing to bits in the corresponding register, as follows:

| **Bit Write** | | **Bit Read** | |
|---|---|---|---|
| Affects associated output channel | | Gives status of associated input channel | |
| **0** | **1** | **0** | **1** |
| Deactivates relay for associated channel | Activates relay for associated channel | Indicates that associated input channel is inactive, i.e. no current is flowing through input opto-coupler associated with that channel | Indicates that associated input channel is active, i.e. current is flowing through the input opto-coupler associated with that channel |

If a channel or group of channels is input only, then writing to the corresponding bits has no effect. Similarly, if a channel or group of channels is output only, then reading the corresponding bits gives an invalid result.

**Note:** In the case of a combined I/O cell, or where loopback testing mode has been invoked, the state of the input bit is the inverse of the output bit, i.e. writing a 1 results in a 0 being read and vice versa. This is because an activated relay provides a very low impedance path between the P2 connector pins, which prevents any current from flowing through the input opto-coupler.

# Interrupt Status Registers (Offsets 8 to 0xC)

| Register | 7 | | | | | | | 0 | Offset |
|---|---|---|---|---|---|---|---|---|---|
| Status 0 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CH0 | 8 |
| Status 1 | CH15 | CH14 | CH13 | CH12 | CH11 | CH10 | CH9 | CH8 | 9 |
| Status 2 | CH23 | CH22 | CH21 | CH20 | CH19 | CH18 | CH17 | CH16 | 0xA |
| Status 3 | CH31 | CH30 | CH29 | CH28 | CH27 | CH26 | CH25 | CH24 | 0xB |
| Counters | N/U | N/U | CNTR5 | CNTR4 | CNTR3 | CNTR2 | CNTR1 | CNTR0 | 0xC |

These are a group of five 8-bit read-only registers. The first four registers store the interrupt status of the 32 channels, while the fifth stores the interrupt status of the six timer/counters.

If a change is detected in an I/O cell and the cell is configured to generate an interrupt on that change, then the interrupt is recorded in the appropriate interrupt status register. At any time, any or all of the channels can detect a change, but only one interrupt may be generated on the VMEbus. Once an interrupt has been serviced, the status register is read, which resets it ready for the next change. Since all the changes are recorded in the register, this method ensures that no interrupts are missed. If a change is detected after an interrupt service routine has started, then it is stored until after the status register has been reset, at which time the bit is set in the status register and a new VMEbus interrupt may be generated.

Interrupt generation on a change of state of a cell is controlled by the Interrupt Mask Registers, described overleaf

# Interrupt Mask Registers 0 to 7 (Offsets 0x10 to 0x17)

| Register | 7 | | | | | | | 0 | Offset |
|---|---|---|---|---|---|---|---|---|---|
| 0 | CH3 | CH3 | CH2 | CH2 | CH1 | CH1 | CH0 | CH0 | 0x10 |
| 1 | CH7 | CH7 | CH6 | CH6 | CH5 | CH5 | CH4 | CH4 | 0x11 |
| 2 | CH11 | CH11 | CH10 | CH10 | CH9 | CH9 | CH8 | CH8 | 0x12 |
| 3 | CH15 | CH15 | CH14 | CH14 | CH13 | CH13 | CH12 | CH12 | 0x13 |
| 4 | CH19 | CH19 | CH18 | CH18 | CH17 | CH17 | CH16 | CH16 | 0x14 |
| 5 | CH23 | CH23 | CH22 | CH22 | CH21 | CH21 | CH20 | CH20 | 0x15 |
| 6 | CH27 | CH27 | CH26 | CH26 | CH25 | CH25 | CH24 | CH24 | 0x16 |
| 7 | CH31 | CH31 | CH30 | CH30 | CH29 | CH29 | CH28 | CH28 | 0x17 |

There are eight 8-bit write-only registers that control the generation of interrupts from the input channels. Two bits are allocated per channel, where the LSB is bit 0 and the MSB is bit 1.

Using the two bits allocated per channel, each channel can be independently programmed as to whether it generates an interrupt on an input change and, if so, whether this occurs on a positive or negative edge (or both), as follows:

| Bit 1 | Bit 0 | Action |
|---|---|---|
| 0 | 0 | No interrupt generated |
| 0 | 1 | Interrupt on low to high transition |
| 1 | 0 | Interrupt on high to low transition |
| 1 | 1 | Interrupt on either transition |

Where high = current flow through the input opto-coupler,
low = no current flow through the input opto-coupler.

# BIT/VME Interrupt Control Register (Offset 0x18)

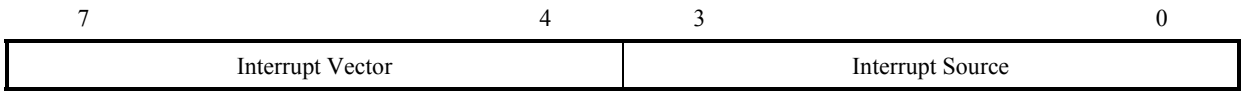
| 7 | 6 | 5 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|
| EL | BIT LED | Reserved | IL2 | IL1 | IL0 |

This 8-bit write-only register controls the level at which a VME interrupt can be generated in response to an input channel or timer change.

After a reset, bits 4 to 7 are clear and bits 0 to 3 are ‘don’t care’.

| Bits | Mnemonic | Function |
|---|---|---|
| 0 to 2 | IL0 to IL2 | Used to select which VME interrupt level is used when the PIO-2 generates interrupts, as follows:<br>0 = Disable VME interrupt generation<br>1 = IRQ1~<br>2 = IRQ2~<br>3 = IRQ3~<br>4 = IRQ4~<br>5 = IRQ5~<br>6 = IRQ6~<br>7 = IRQ7~ |
| 3 to 5 | | ***Do not write to these bits*** |
| 6 | BIT LED | Turns the front panel LED on and off as follows:<br>0 = LED on<br>1 = LED off<br>This bit has no effect on other areas of the PIO-2 circuitry. It is cleared (LED on) during power-up or system reset. A host CPU should BIT test the PIO-2 before setting it (LED off) |
| 7 | EL | Enable Loopback<br>0 = Normal operation<br>1 = Loopback mode<br>Setting this bit disables the outputs and inputs, and enables an internal loopback path between them. All outputs are off (relays open) and the inputs are ignored. This feature is intended for use during BIT testing of the card by the host CPU.<br>**Note:** In loopback mode, ***all*** output channels, including those that are not populated on a particular build version, are looped back to their corresponding inputs. A 32-channel input-only card can therefore still use this feature.<br>In loopback mode, the inputs are inverted. A “1” written to a channel is read back as “0”, and a “0” written to a channel is read back as “1” (after waiting for the input debounce circuitry to pass the change through) |

# VME Interrupt Vector Register (Offset 0x19)




This is a 4-bit read/write register that contains the most significant four bits of the interrupt vector that is returned during the VMEbus interrupt acknowledge cycles. This is user programmable to a value that is compatible with the host processor requirements.

After a reset, bits 4 to 7 are clear; bits 0 to 3 are ‘don’t care’.

During IACK cycles or reads of this register, the least significant four bits of the vector contain a priority encoded indication of the source of the interrupt, as follows:

0 = Spurious interrupt

1 = Channels 7 to 0

2 = Channels 15 to 8

3 = Channels 23 to 16

4 = Channels 31 to 24

5 = Counter 0

6 = Counter 1

7 = Counter 2

8 = Counter 3

9 = Counter 4

10 = Counter 5

# Programmable Interval Timer Registers (Offsets 0x20 to 0x2E)

The PIO-2 has two Programmable Interval Timers (PITs). These are either Intel or Harris 82C54 devices. Each PIT has three independent Counters.

## Counter Registers

Each of the three Counters is a 16-bit down-counter and has a separate 8-bit read/write register for loading it and reading it back (to access all 16 bits of the Counter requires two consecutive accesses to the Counter register). The addresses of these Counter registers are as follows:

| PIT | Counter | Offset |
|---|---|---|
| 1 | 0 | 0x20 |
| | 1 | 0x22 |
| | 2 | 0x24 |
| 2 | 3 | 0x28 |
| | 4 | 0x2A |
| | 5 | 0x2C |

## Control Word Registers

These are 8-bit write-only registers at offset 0x26 for PIT 1 and 0x2E for PIT 2. The Counters are programmed by writing a Control Word (which specifies which Counter is being programmed) into the Control Word register.

**Note:** The Control Word has 8 bits, whereas the term 'word' usually means 16 bits. The term 'Control Word' is part of the manufacturer's terminology for the PIT.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SC1 | SC0 | RW1 | RW0 | M2 | M1 | M0 | BCD |

| Bits | Mnemonic | Function |
|---|---|---|
| 7 and 6 | SC1 and SC0 | Select which of the three Counters is to be programmed, as follows: |

| SC1 | SC0 | Counter |
|---|---|---|
| 0 | 0 | Counter 0 or 3 |
| 0 | 1 | Counter 1 or 4 |
| 1 | 0 | Counter 2 or 5 |
| 1 | 1 | Read-back command |

| Bits | Mnemonic | Function |
|---|---|---|
| 5 and 4 | RW1 and RW0 | Select the initial count format, which can be one or two-byte, as follows: |

| RW1 | RW0 | Format |
|---|---|---|
| 0 | 0 | Counter Latch command |
| 0 | 1 | Read/write least significant byte only |
| 1 | 0 | Read/write most significant byte only |
| 1 | 1 | Read/write least significant byte then most significant byte |

| Bits | Mnemonic | Function |
|---|---|---|
| 3 to 1 | M2, M1 and M0 | Selects the Mode in which the Counter is to operate, as follows: |

| M2 | M1 | M0 | Mode |
|---|---|---|---|
| 0 | 0 | 0 | Mode 0 |
| 0 | 0 | 1 | Mode 1 |
| 0 | 1 | 0 | Mode 2 |
| 0 | 1 | 1 | Mode 3 |
| 1 | 0 | 0 | Mode 4 |
| 1 | 0 | 1 | Mode 5 |

| Bits | Mnemonic | Function |
|---|---|---|
| 0 | BCD | Controls whether the initial count is in binary or BCD form, as follows: |

| | |
|---|---|
| 0 | Binary Counter (16 bits) |
| 1 | BCD Counter (4 decades) |

## PIT Programming Overview

Each PIT contains several internal 8-bit registers that allow it to be programmed to operate in various ways. Each Counter is programmed by writing a Control Word and then an initial count. The Counter counts down from this initial count to zero. The action taken when the Counter reaches zero depends on the Mode and the interrupt generation capabilities set up.

Software for setting up a Counter in the PIT with an initial count value must obey the following restrictions:

- For each Counter, the Control Word must be written before the initial count is written
- The Control Word must be written to the Control Word register for the Counter being initialised, and the initial count must be written to the Counter register for the Counter being initialised
- Initial counts of length 16 bits are set by writing the least significant 8 bits to the Counter register, then writing the most significant 8 bits to the same Counter register
- The initial count must follow the count format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte)

Since the Control Word register and the three Counters have separate addresses, and each Control Word specifies the Counter to which it applies (with the SC0, SC1 bits), no special instruction sequence is required. Any programming sequence that follows the above restrictions is acceptable.

A new initial count may be written to a Counter at any time without affecting the Counter's programmed Mode in any way. Counting is affected as described in the Mode definitions. The new count must follow the programmed count format.

**Caution**

If a Counter is programmed to read or write two-byte (16-bit) counts, software must not transfer control between writing the first and second byte to another routine which also writes into that same Counter. Otherwise, the Counter will be loaded with an incorrect count

## PIT Read-back Operations

It is often desirable to read the value of a Counter without disturbing the count in progress. The Counter registers can be read directly, but the Counter may be in the process of changing when it is read, giving an undefined result. To avoid this problem, a Counter Latch Command should be written to the PIT before reading the Counter register (alternatively, the Read-Back command can be used, see page 4-12).

### Counter Latch Command

The Counter Latch command is written to the Control Word register of the PIT. This causes the selected Counter's value to be latched into the Counter register and held until the register has been read (the Counter register is separate from the Counter itself, so the Counter register can hold the latched count while the Counter continues to decrement). This allows the contents of the Counters to be read without affecting counting in progress.

Multiple Counter Latch commands may be used to latch more than one Counter. Each latched Counter value is held in its own Counter register until it is read. Counter Latch commands do not affect the programmed Mode of the Counter in any way. If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch command is ignored. The count read is the count at the time the first Counter Latch command was issued.

The count must be read according to the format (least significant byte only, most significant byte only or least significant byte then most significant byte) which was set in the Control Word when the Counter was programmed. Specifically, if the Counter is programmed for two-byte counts, two bytes must be read. The two bytes do not have to be read one immediately after the other; read or write or programming operations of other Counters may be inserted between them.

The format of the Counter Latch command is shown below.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SC1 | SC0 | 0 | 0 | Don't care (0) | | | |

Where SC1, and SC0 specify the counter to be latched, as follows:

| SC1 | SC0 | Counter |
|---|---|---|
| 0 | 0 | 0 or 3 |
| 0 | 1 | 1 or 4 |
| 1 | 0 | 2 or 5 |

### Read-back Command

The Read-back command allows the user to check the count value, programmed Mode, and current state of the Out pin and Null Count flag of the selected counter(s). The command is written into the Control Word register and has the format shown below:

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | COUNT~ | STATUS~ | CNT2 | CNT1 | CNT0 | 0 |

| Bit | Mnemonic | Function |
|---|---|---|
| 5 | COUNT~ | Clear to latch count of selected Counter(s) |
| 4 | STATUS~ | Clear to latch status of selected Counter(s) |
| 3 | CNT2 | Set to latch Counter 2 or 5 |
| 2 | CNT1 | Set to latch Counter 1 or 4 |
| 1 | CNT0 | Set to latch Counter 0 or 3 |
| 0 | BCD | Must be clear |

The Read-back command may be used to latch more than one Counter by clearing the COUNT~ bit (bit 5) and selecting the desired Counters (more than one of the CNT bits can be set). This single command is functionally equivalent to several Counter Latch commands. The value of each Counter is held separately in its Counter register until it is read (the Counter registers are separate from the Counters themselves, so the Counter registers can hold the latched counts while the Counters continue to decrement). If multiple count Read-back commands are issued to the same Counter without reading the count, all but the first are ignored, i.e. the count that is read is the count at the time the first Read-back command was issued.

The Read-back command may also be used to latch status information on the selected Counter(s) by clearing the STATUS~ bit (bit 4). The status of a Counter can then be accessed by a read from its Counter register (this status information cannot be read without being latched first). The format of the Counter status is shown below:

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OUTPUT | NULL COUNT | RW1 | RW0 | M2 | M1 | M0 | BCD |

| Bits | Mnemonic | Function |
|---|---|---|
| 7 | OUTPUT | The current state of the OUT pin, as follows:<br>1 = OUT pin is 1<br>0 = OUT pin is 0<br>The PIT has an OUT pin for each Counter. This allows software to monitor the Counter's output |
| 6 | NULL COUNT | 1 = Null count has been reached<br>0 = Count available for reading<br>Indicates whether the last initial count written to the Counter register was loaded into the Counter itself. The exact time this happens depends on the Mode in which the Counter was programmed to operate, but until the count is loaded into the Counter, it cannot be read from the Counter. If the count is latched or read before this time, the count value will not reflect the new initial count just written |
| 5 to 0 | M2 to M0, BCD | Counter's programmed Mode and BCD bits as programmed in the last Control Word written for that Counter |

If multiple status Read-back commands are issued to the same Counter(s) without reading the status, all but the first are ignored, i.e. the status that is read is the status of the Counter at the time the first status read-back command was issued.

Both the count and the status of the selected Counter(s) may be latched simultaneously by clearing both the COUNT~ and STATUS~ bits (bits 5 and 4) in the Read-back command. This is functionally equivalent to issuing two separate Read-back commands at once. If both the count and the status of a Counter are latched in this way, the first read of its Counter register returns the latched status. The next one or two reads of the Counter register (depending on whether the counter is programmed for one or two byte counts) returns the latched count. If multiple count and/or status Read-back commands are issued to the same Counter(s) without any intervening reads, all but the first are ignored.

## Modes

The Control Word used to program the PIT includes bits that control the Mode in which the Counter operates. The available Modes are described in this section.

### Definitions

The following terms are used in describing the operation of the PIT:

| | |
|---|---|
| CLK: | The clock input pin of a Counter, which causes it to count. |
| GATE: | The GATE input pin of a Counter, which enables counting. |
| OUT: | The OUT output pin which is controlled by a counter. |
| CLK pulse: | A rising edge, then a falling edge, of a Counter's CLK input. |
| Trigger: | A rising edge of a Counter's GATE input. |
| Counter Loading: | The transfer of a count from the Counter Register to the Counter. |

**Note:** Each of the three Counters has a CLK pin, a GATE pin and an OUT pin of its own on the PIT. The GATE pins, however, are not used on the PIO-2 and are permanently pulled up into the active state.

### Mode 0: Interrupt on Terminal Count

After the Control Word is written, OUT is initially low, and remains low until the Counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written to the Counter.

After the Control Word and initial count are written to a Counter, the initial count is loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of *n*, OUT does not go high until *n* + 1 CLK pulses after the initial count is written.

If a new count is written to the Counter, it is loaded on the next CLK pulse and counting continues from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte disables counting. OUT is set low immediately (no clock pulse required).

2) Writing the second byte allows the new count to be loaded on next CLK pulse.

This allows the counting sequence to be synchronized by software. Again OUT does not go high until *n* + 1 CLK pulses after the new count of *n* is written.

### Mode 1: Hardware Retriggerable One-Shot

Not available on PIO-2 as the GATE inputs to the PIT are tied high.

### Mode 2: Rate Generator

This Mode functions like a divide-by- *n* counter. OUT is initially high. When the initial count has decremented to 1, OUT goes low for one CLK pulse. OUT then goes high again, the Counter reloads the initial count and the process is repeated. Mode 2 is periodic; the same sequence is repeated indefinitely. For an initial count of *n*, the sequence repeats every *n* CLK cycles.

After writing a Control Word and initial count, the Counter is loaded on the next CLK pulse. OUT goes low *n* CLK pulses after the initial count is written. This also allows the Counter to be synchronized by software. Writing a new count while counting does not affect the current counting sequence.

### Mode 3: Square Wave Mode

Interrupt generating capabilities are identical to Mode 2, use Mode 2.

### Mode 4: Software Triggered Mode

OUT is initially high. When the initial count expires, OUT goes low for one CLK pulse then goes high again. The counting sequence is triggered by writing the initial count.

After writing a Control Word and initial count, the Counter is loaded on the next CLK pulse. This CLK pulse does not decrement the count, so for an initial count of *n*, OUT does not strobe low until *n* + 1 CLK pulses after the initial count is written.

If a new count is written during counting, it is loaded on the next CLK pulse and counting continues from the new count. If a two-byte count is written, the following happens:

1) Writing the first byte has no effect on counting.

2) Writing the second byte allows the new count to be loaded on the next CLK pulse.

This allows the sequence to be retriggered by software. OUT strobes low *n* + 1 CLK pulses after the new count of *n* is written.

### Mode 5: Hardware Triggered Strobe (Retriggerable)

This Mode is not available on PIO-2 as GATE inputs to the PITs are tied high.

## Operation Common to All Modes

This section contains general information on the operation of the PIT.

### Programming

When a Control Word is written to a Counter, all Control Logic is immediately reset and OUT goes to a known initial state; no CLK pulses are required for this.

### GATE

The GATE input is always sampled on the rising edge of CLK. In Modes 0, 2, 3 and 4, the GATE input is level sensitive, and the logic level is sampled on the rising edge of CLK. In modes 1, 2, 3 and 5 the GATE input is rising-edge sensitive. In these Modes, a trigger sets an edge-sensitive flip-flop in the Counter. This flip-flop is reset immediately after it is sampled. In this way, a trigger is detected no matter when it occurs; a high logic level does not have to be maintained until the next rising edge of CLK.

**Notes**: In Modes 2 and 3, the GATE input is both edge-and-level- sensitive. The GATE inputs of all 6 counters are permanently tied high on PIO-2 so Modes 1 and 5 are not available.

### Counter

New counts are loaded and Counters are decremented on the falling edge of CLK.
The largest possible initial count is 0 (see the Note below).

The counter does not stop when it reaches zero. In Modes 0, 1, 4 and 5, the Counter wraps around to the highest count, either 0xFFFF for binary counting or 9999 for BCD counting. Modes 2 and 3 are periodic; the Counter reloads itself with the initial count and continues counting from there. The minimum and maximum count values for each Mode are shown below:

| Mode | Minimum Count | Maximum Count |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 0 |
| 2 | 2 | 0 |
| 3 | 2 | 0 |
| 4 | 1 | 0 |
| 5 | 1 | 0 |

**Note:** 0 is equivalent to $2^{16}$ for binary counting and $10^4$ for BCD counting.

### OUT

The OUT outputs for each of the 6 counters is connected to the LCA, which generates an interrupt on a low to high transition of these pins.

### CLK

The CLK inputs for all six counters are connected together and supplied with a 610.35 Hz clock signal from the LCA. This allows time intervals to be programmed in multiples of 1.64ms.

## PIT Programming Examples

### Example 1

To initialize Counter 2 of PIT 1 in mode 0 with an initial count of 0xABCD, the host software would perform the following steps:

1) Write a byte of value 0xB0 to address BASEADD+0x26 to set the Control Word for Counter 2 for writing a 16-bit binary count, and setting Counter 2 in Mode 0.
2) Write a byte of value 0xCD to address BASEADD+0x24 to set the least significant byte of the initial count for Counter 2.
3) Write a byte of value 0xAB to address BASEADD+0x24 to set the most significant byte of the initial count for Counter 2.

### Example 2

To initialize Counter 0 of PIT 2 in mode 0 with an initial count of 0xABCD, the host software would perform the following steps:

1) Write a byte of value 0x30 to address BASEADD+0x2E to set the Control Word for Counter 0 for writing a 16-bit binary count, and setting Counter 0 in Mode 0.
2) Write a byte of value 0xCD to address BASEADD+0x28 to set the least significant byte of the initial count for Counter 0.
3) Write a byte of value 0xAB to address BASEADD+0x28 to set the most significant byte of the initial count for Counter 0.

### Example 3

To initialize Counter 1 of PIT 2 in mode 4 with an initial count of 0xFFFF, the host software would perform the following steps:

1) Write a byte of value 0x78 to address BASEADD+0x2E to set the Control Word for Counter 1 for writing a 16-bit binary count, and setting Counter 1 in Mode 4.
2) Write a byte of value 0xFF to address BASEADD+0x2A to set the least significant byte of the initial count for Counter 1.
3) Write a byte of value 0xFF to address BASEADD+0x2A to set the most significant byte of the initial count for Counter 1.

### Example 4

To initialize Counter 0 of PIT 1 in mode 2 with an initial count of 0xFFFF, the host software would perform the following steps:

1) Write a byte of value 0x34 to address BASEADD+0x26 to set the Control Word for Counter 0 for writing a 16-bit binary count, and setting Counter 0 in Mode 3.
2) Write a byte of value 0xFF to address BASEADD+0x20 to set the least significant byte of the initial count for Counter 0.
3) Write a byte of value 0xFF to address BASEADD+0x20 to set the most significant byte of the initial count for Counter 0.

# ID Register (Offset 0x30)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| E2 | E1 | E22 | E20 | E27 | E25 | E23 | E21 |

This is an 8-bit read-only register that returns the value preset on links E20, E21, E22, E23, E25 and E27. It also returns the debounce clock selection on links E2 and E1.

The user may fit any combination of links E20, E21, E22, E23, E25 and E27, as they have no effect on the operation of the PIO-2, but provide the host software with a means of identifying the PIO-2's configuration (how many inputs and outputs are fitted, voltage ranges etc.). For example, if there were three PIO-2s in a rack, two with 32 output-only channels and one with 32 input-only channels, then by setting up different ID values on the output and input boards, the host software could differentiate between the output and input boards by reading this register on each board. The host could then run the appropriate code for each board.

The mapping of values to configuration is user dependent and can be read by the host in the ID register at VME address BASEADD+0x30, where a '0' bit means that the corresponding link is fitted, and a '1' means that the corresponding link is not fitted.

Bits 7 and 6 can be used to read back the state of links E2 and E1 respectively. Links E2 and E1 select the clock that is used to debounce input signals from the opto-couplers, as shown in the following table:

| E2 | E1 | Debounce Clock Rate | Response Time† (worst case) |
|---|---|---|---|
| Out | Out | 4.76 Hz | **420 ms** |
| Out | In | 76 Hz | **27 ms** |
| In | Out | 1220 Hz | **1.7 ms** |
| In | In | 19.53 kHz | **160 µs** |

† The measured total time from the opto-coupler current flow to an interrupt generated on the VMEbus (if enabled).

# Chapter 5 - BIT Configuration and Fail Data

## Configuration Data

The PIO-2 test requires 6 longwords of configuration data, 4 of which are the fixed usage test masks and slave processor timeout.

| Configuration Data Word | Meaning | |
|---|---|---|
| 0 | PIO-2 test ignore mask. Bits are set to ignore tests as follows: | |
| | **Bit** | **Test** |
| | 0 | Internal loopback tests |
| | 1 | External loopback tests |
| | 2 | PIT counter tests |
| | 3 | PIO-2 interrupt tests |
| | 4 | PIO-2 ID register test |
| 1 | PIO-2 test serious failure mask. The bit allocation is as above | |
| 2 | PIO-2 test fatal failure mask. The bit allocation is as above | |
| 3 | PIO-2 slave processor time-out count | |
| 4 | PIO-2-under-test base-address | |
| 5 | PIO-2 I/O blocks fitted | |
| | **Bit** | **I/O Channels Tested if Bit Set** |
| | 0 | Input on channels 0 to 7 |
| | 1 | Output on channels 0 to 7 |
| | 4 | Input on channels 8 to 15 |
| | 5 | Output on channels 8 to 15 |
| | 8 | Input on channels 16 to 23 |
| | 9 | Output on channels 16 to 23 |
| | 12 | Input on channels 24 to 31 |
| | 13 | Output on channels 24 to 31 |

The run time for a fully populated PIO-2 card under test is approximately:

2 seconds for Initial BIT.

12 seconds for Comprehensive BIT.

2 to 3 minutes for Factory Test BIT.

# Fail Data

The PIO-2 test is divided into the following test modules:

1) Internal loopback test
2) External loopback test
3) PIT counter test
4) Interrupt test
5) ID register test

Each test is identified by its respective mask code.

The following sections describe each test module with its fail data. The ‘mask_code’ shows the test that failed, and the ‘fail_code’ shows the fail point within the test module. The remainder of the fail data is fail point specific.

The following table shows the ‘mask_code’ values for the test modules:

| Mask_Code | Test Module |
|---|---|
| 0001 | Internal loopback test |
| 0002 | External loopback test |
| 0004 | PIT counter test |
| 0008 | Interrupt test |
| 0010 | ID register test |

## Reset Routine

This routine is common to all tests, so the following fail codes apply to all Mask_Code values:

| Fail_Code | Meaning |
|---|---|
| 0110 | Write to channels 0 to 7 |
| 0111 | Write to channels 8 to 15 |
| 0112 | Write to channels 16 to 23 |
| 0113 | Write to channels 24 to 31 |
| 0120 | Initialize interrupt masks |
| 0130 | Initialize PIT 0, counters 0, 1 and 2 |
| 0140 | Initialize PIT 1, counters 3, 4 and 5 |
| 0150 | Clear interrupt status for channels 0 to 7 |
| 0160 | Clear interrupt status for channels 8 to 15 |
| 0170 | Clear interrupt status for channels 16 to 23 |
| 0180 | Clear interrupt status for channels 24 to 31 |
| 0190 | Clear interrupt status for counters 0 to 5 |
| 01A0 | Check channels 0 to 7 initialized |
| 01A1 | Check channels 8 to 15 initialized |
| 01A2 | Check channels 16 to 23 initialized |
| 01A3 | Check channels 24 to 31 initialized |

## Internal Loopback Test (Mask_Code = 0001)

The internal loopback test includes an access test, a PIO-2 card reset and an internal loopback checking the speed of loopback. The access test checks access to the BIT/VME Interrupt Register. PIO-2 reset uses the standard reset routine. The internal loopback test is run on all 32 channels of the PIO-2 card under test; this is because internal loopback is a function of the on-board LCA. The test algorithm is: write 1 to all I/O ports, check no exceptions are generated and that ports do not change too quickly, after a suitable delay re-read all ports checking that they are now all changed to the correct values. The test is then repeated writing 0 to all I/O ports and checking they respond in the expected time.

The Fail_Data words contain the following information:

| Fail_Data Word | Contents |
|---|---|
| 0 | Data received from input channel |
| 1 | Data expected from input channel |
| 2 | I/O block under test<br>0 = channels 0 to 7<br>1 = channels 8 to 15<br>2 = channels 16 to 23<br>3 = channels 24 to 31 |

The Fail_Codes are listed in the following table:

| Fail_Code | Meaning |
|---|---|
| 0050 | Access to BIT/VME Interrupt Control Register failed |
| 0200 | Exception generated during write to I/O ports |
| 0X10 | Ports 0 to 7 changed too quickly |
| 0X20 | Ports 8 to 15 changed too quickly |
| 0X30 | Ports 16 to 23 changed too quickly |
| 0X40 | Ports 24 to 31 changed too quickly |
| 0X50 | Ports 0 to 7 failed to change in time |
| 0X60 | Ports 8 to 15 failed to change in time |
| 0X70 | Ports 16 to 23 failed to change in time |
| 0X80 | Ports 24 to 31 failed to change in time |

X = 2 for writes of 1

X = 3 for writes of 0

## External Loopback Test (Mask_Code = 0002)

This test is only run in factory test mode (tertiary BIT), as it requires an additional, Known To Be Good (KTBG) PIO-2, an external loopback cable and external power input, to drive the loopback. If factory test mode is not selected, this routine exits with a pass status.

The external loopback test includes an access test of the card under test, a reset of both PIO-2 cards, a check of the configuration data and an external loopback checking the speed of loopback.

Access is tested by reading from and writing to the BIT/VME Interrupt Register. Configuration Data Word 5 is checked, identifying channels and modes to be tested. Reset of the PIO-2 under test is done using the standard reset routine. A similar routine resets the KTBG PIO-2. The inputs of the card under test are tested by writing 18 test patterns, to the KTBG PIO-2 and checking the data received on the card under test. Outputs are tested after inputs using the reverse transfer, both tests checking the transfer is within the appropriate time window.

The Fail_Data words contain the following information:

| Fail_Data Word | Contents |
|---|---|
| 0 | Data received from input channel |
| 1 | Data transmitted via output channel |
| 2 | Data expected from input channel |
| 3 | I/O block under test<br>0 = channels 0 to 7<br>1 = channels 8 to 15<br>2 = channels 16 to 23<br>3 = channels 24 to 31 |

The Fail_Codes are listed below:

| Fail_Code | Meaning |
|---|---|
| 0010 | KTBG PIO-2 access fail; should be at address 0x100 above card under test |
| 0020 | KTBG PIO-2 setup failed |
| 0050 | Access to BIT/VME Interrupt Control Register failed |
| 0101 | Check Configuration Data Word 5 valid |
| 02X0 or 03Y0 | Error during write of test data to KTBG card |
| 02X1 or 03Y1 | Data on channels 0 to 7 received too fast |
| 02X2 or 03Y2 | Data on channels 8 to 15 received too fast |
| 02X3 or 03Y3 | Data on channels 16 to 23 received too fast |
| 02X4 or 03Y4 | Data on channels 24 to 31 received too fast |
| 02X5 or 03Y5 | Data on channels 0 to 7 not received |
| 02X6 or 03Y6 | Data on channels 8 to 15 not received |
| 02X7 or 03Y7 | Data on channels 16 to 23 not received |
| 02X8 or 03Y8 | Data on channels 24 to 31 not received |
| 04X0 or 05Y0 | Error during write of test data to test card |
| 04X1 or 05Y1 | Data on channels 0 to 7 transmitted too fast |
| 04X2 or 05Y2 | Data on channels 8 to 15 transmitted too fast |
| 04X3 or 05Y3 | Data on channels 16 to 23 transmitted too fast |
| 04X4 or 05Y4 | Data on channels 24 to 31 transmitted too fast |
| 04X5 or 05Y5 | Data on channels 0 to 7 not transmitted |
| 04X6 or 05Y6 | Data on channels 8 to 15 not transmitted |
| 04X7 or 05Y7 | Data on channels 16 to 23 not transmitted |
| 04X8 or 05Y8 | Data on channels 24 to 31 not transmitted |

X is the test pattern number in hex.

Y is the test pattern number less 16.

## PIT Counter Test (Mask_Code = 0004)

The PIT Counter Test checks all modes (0 to 5). Tests are divided into initial, comprehensive and factory, depending on the run time. An access test of the card under test and reset of the card are also included.

Access is tested by reading from and writing to the BIT/VME Interrupt Register, reset of the PIO-2 is done using the standard reset routine. Each counter is tested in turn, all following the same tests, starting with mode 0 and working through to mode 5. All modes are checked by a counter running a status check. Wrap round tests are carried out in modes 0, 2, 3 and 4 during factory test BIT. Mode 3 test checks the OUT signal. Modes 2 and 3 are checked to ensure counter reload while the counter is running is not allowed. Conversely, in the mode 4 test, counter reload while the counter is running, *is* allowed.

The Fail_Data words contain the following information:

| Fail_Data Word | Contents |
|---|---|
| 0 | Counter under test (0 to 5) |
| 1 | PIT status word at fail point |
| 2 | Counter value at fail point |
| 3 | 4 = Mode 2 test<br>6 = Mode 3 test<br>10 = LSByte counter under test<br>20 = MSByte counter under test<br>30 = WRD counter under test |

The Fail_Codes are listed in the following table:

| Fail_Code | Meaning |
|---|---|
| 0050 | Access to BIT/VME Interrupt Control Register failed |
| 0200 | Mode 0 status check, latching status only |
| 0210 | Mode 0 status check, latching status and count |
| 0220 | Mode 0 count check, latching status and count |
| 0230 | Mode 0 status check, after loading count |
| 0240 | Mode 0 count check, after loading count |
| 0250 | Mode 0 status check, MSB/WRD mode |
| 0260 | Mode 0 status check, during MSB/WRD countdown |
| 0270 | Mode 0 count check, during MSB/WRD countdown |
| 0280 | Mode 0 status check, after second write to counter |
| 0290 | Mode 0 count check, after second count write |
| 02A0 | Mode 0 status check, during wrap-around test |
| 02B0 | Mode 0 count check, during wrap-around test |
| 02C0 | Mode 1 status check |
| 0300 | Mode 2/3 status check, during count |
| 0310 | Mode 2/3 check, count running |
| 0320 | Mode 2/3 check, count reload during countdown not allowed in this mode |
| 0330 | Mode 2/3 check, status during wrap-around test |
| 0340 | Mode 2/3 check, count during wrap-around test |
| 0350 | Mode 4 status check |
| 0360 | Mode 4 count, running check |
| 0370 | Mode 4 count, reload during run check |
| 0380 | Mode 4 check, count during wrap-around test |
| 0390 | Mode 5 status check |

## Interrupt Test (Mask_Code = 0008)

After carrying out the access test and resetting the card under test, the Interrupt test checks all PIO-2 interrupt sources from VME interrupt level 7 down to 1 and tests interrupt queuing.

Access is tested by reading from and writing to the BIT/VME Interrupt Register, reset of the PIO-2 is done using the standard reset routine. Interrupts are disabled and cleared, vector base is setup (0x0C0) and a check for premature interrupts is made, before testing PIO-2 generated interrupts.

Interrupts generated by each of the 6 PIT counters are tested, at level 7 down to 1. I/O port interrupts are tested in factory test mode only. This is because with a debounce clock of 4.76 Hz, the test time increases by approximately 1 minute from that taken by the fastest setting (19.53 kHz). Interrupts are tested in all 3 modes, high to low and low to high transition, high to low only and low to high only. For the second and third options a check is made that interrupts are not generated when not expected.

Interrupt queuing is exercised by interrupts generated on both transitions, checking that the correct interrupts are active or inactive as expected during the formation and clearing of the queue.

The Fail_Data words contain the following information:

| Fail_Data Word | Contents |
|---|---|
| 0 | Interrupt mask (LSByte). A bit is set for each active interrupt level |
| 1 | Last test pattern written to I/O port |
| 2 | Address of I/O port block last written to |
| 3 | Timeout count. If 0, test failed due to a timeout |
| 4 | Address of Interrupt Status Register last accessed |
| 5 | Last value written to the BIT/VME Interrupt Control register |

The Fail_Codes are listed in the following tables:

**Where X = 2 to 5**

| Fail_Code | Meaning |
|---|---|
| 0050 | Access to BIT/VME Interrupt Control Register failed |
| 0060 | Premature interrupts found in the test system |
| 0XY0 | PIT timeout error. PIT countdown failed therefore no interrupt generated |
| 0XY1 | Wrong interrupt vector generated |
| 0XY2 | Wrong interrupt status mask generated |
| 0XY3 | Interrupt not cleared by read of interrupt mask |
| 0XY4 | Interrupt not generated on VME bus |

X = 2 for Interrupt levels 7 and 6
X = 3 for Interrupt levels 5 and 4
X = 4 for Interrupt levels 3 and 2
X = 5 for Interrupt level 1

Y = 0 or 8 for PIT counter 0
Y = 1 or 9 for PIT counter 1
Y = 2 or A for PIT counter 2
Y = 3 or B for PIT counter 3
Y = 4 or C for PIT counter 4
Y = 5 or D for PIT counter 5

**Where X = 6 to C**

| Fail_Code | Meaning |
|---|---|
| 0X00 to 0X29 | Transition interrupt check for I/O channels 0 to 7 |
| 0X30 to 0X59 | Transition interrupt check for I/O channels 8 to 15 |
| 0X60 to 0X89 | Transition interrupt check for I/O channels 16 to 23 |
| 0X90 to 0XB9 | Transition interrupt check for I/O channels 24 to 31 |
| 0XC0 to 0XD4 | Interrupt queuing error |

X = 6 for Interrupt level 7
X = 7 for Interrupt level 6
X = 8 for Interrupt level 5
X = 9 for Interrupt level 4
X = A for Interrupt level 3
X = B for Interrupt level 2
X = C for Interrupt level 1

## ID Register Test (Mask_Code = 0010)

The ID register test is in 2 parts: first checking access to the BIT/VME Interrupt Register and secondly reading and checking the ID Register.

The Fail_Codes are listed in the following table:

| Fail_Code | Meaning |
|---|---|
| 0050 | Access to BIT/VME Interrupt Control Register failed |
| 0060 | Read of ID Register failed |

# Appendix A - I/O Device Specifications

This appendix provides input/output electrical specifications for the solid state relays and opto-isolators used on PIO-2.

| Build Level | Solid State Relay Used | Opto-coupler Used |
|---|---|---|
| 1 | PVDZ172 | HCPL 2731 |
| 2 | PVDZ172 | HCPL 2731 |
| 3 | CD20CD | 6N140A |
| 4 | CD20CD | 6N140A |
| 5 | CD20CDY | 6N140A/883B DESC 8302401EC |

**Note:** The HCPL2731 and 6N140A are manufactured by Hewlett Packard.
The CD20CD range is manufactured by Teledyne Solid State.

# Input Specifications

## HCPL2731 (-40°C to +85°C Ambient Temperature)

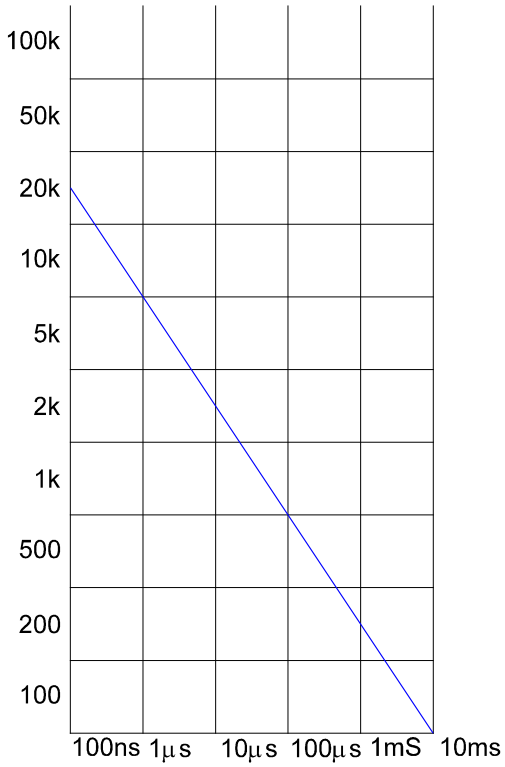
| Parameter | Minimum | Maximum | Units |
|---|---|---|---|
| Continuous operating voltage (all variants) | | 32 (see Note 1) | V DC |
| Voltage for logic 1 output | 14 for 28V version<br>5.7 for 12V version<br>3.6 for 5V version | | V DC |
| Voltage for logic 0 output | | 1.1 | V DC |
| Current for a logic 1 output ($I_F$) | 0.75 | | mA |
| Reverse voltage | | 3.0 | V DC |
| Isolation voltage | 250 | | V DC |
| Load resistance | | 15 for 28V version<br>6.8 for 12V version<br>1.2 for 5V version | kΩ (see note 2) |
| Common load transient immunity $\frac{dv_F}{dt}$ | 500 | | V/µs |

**Notes:**

1 Surge above 32V can be tolerated in line with the protective zener diode ratings.

2 This assumes a 10% load power supply ($V_{LOAD}$) - see Figure A.2. Normally $R_{LOAD}$ would only represent cabling resistance.

**Figure A-1. Zener Rating**



## 6N140A (-55°C to +125°C Ambient Temperature)

| Parameter | Minimum | Maximum | Units |
|---|---|---|---|
| **Continuous operating voltage (all variants)** | | 32 for 28V version<br>32 for 12V version<br>24 for 5V version<br>(see Note 1) | V DC |
| **Voltage for logic 1 output** | 14 for 28V version<br>5.7 for 12V version<br>3.6 for 5V version | | V DC |
| **Voltage for logic 0 output** | | 1.1 | V DC |
| **Current for a logic 1 output ($I_F$)** | 0.75 | | mA |
| **Reverse voltage** | | 3.0 | V DC |
| **Isolation voltage** | 250 | | V DC |
| **Load resistance** | | 15 for 28V version<br>6.8 for 12V version<br>1.2 for 5V version | kΩ (see note 2) |
| **Common load transient immunity $\frac{dv_F}{dt}$** | 500 | | V/µs |

**Notes:**

1 Surge above 32V can be tolerated in line with the protective zener diode ratings.

2 This assumes a 10% load power supply ($V_{LOAD}$) - see Figure A.2. Normally $R_{LOAD}$ would only represent cabling resistance.

**Figure A-2. Test Circuit**

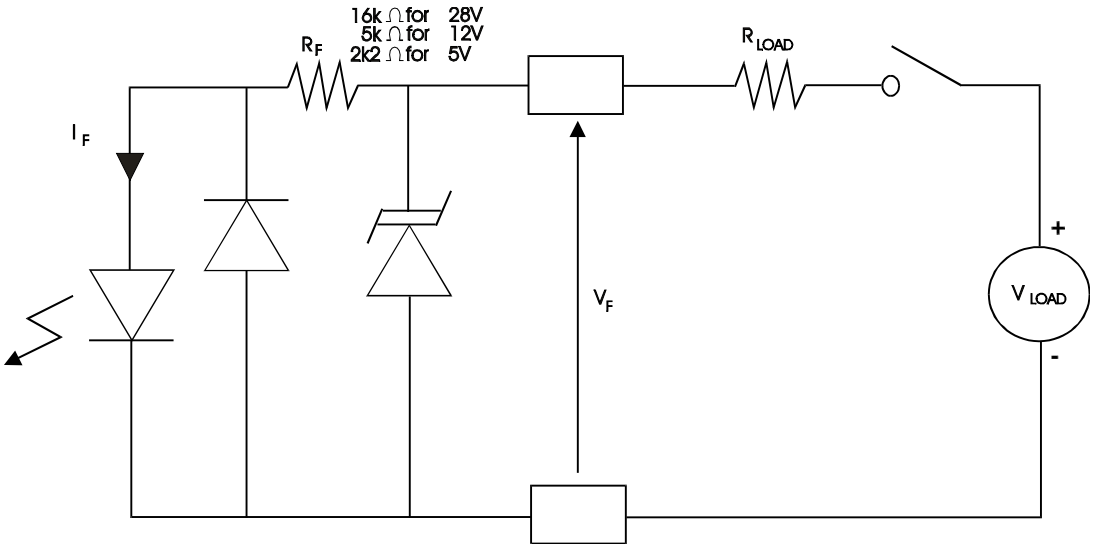
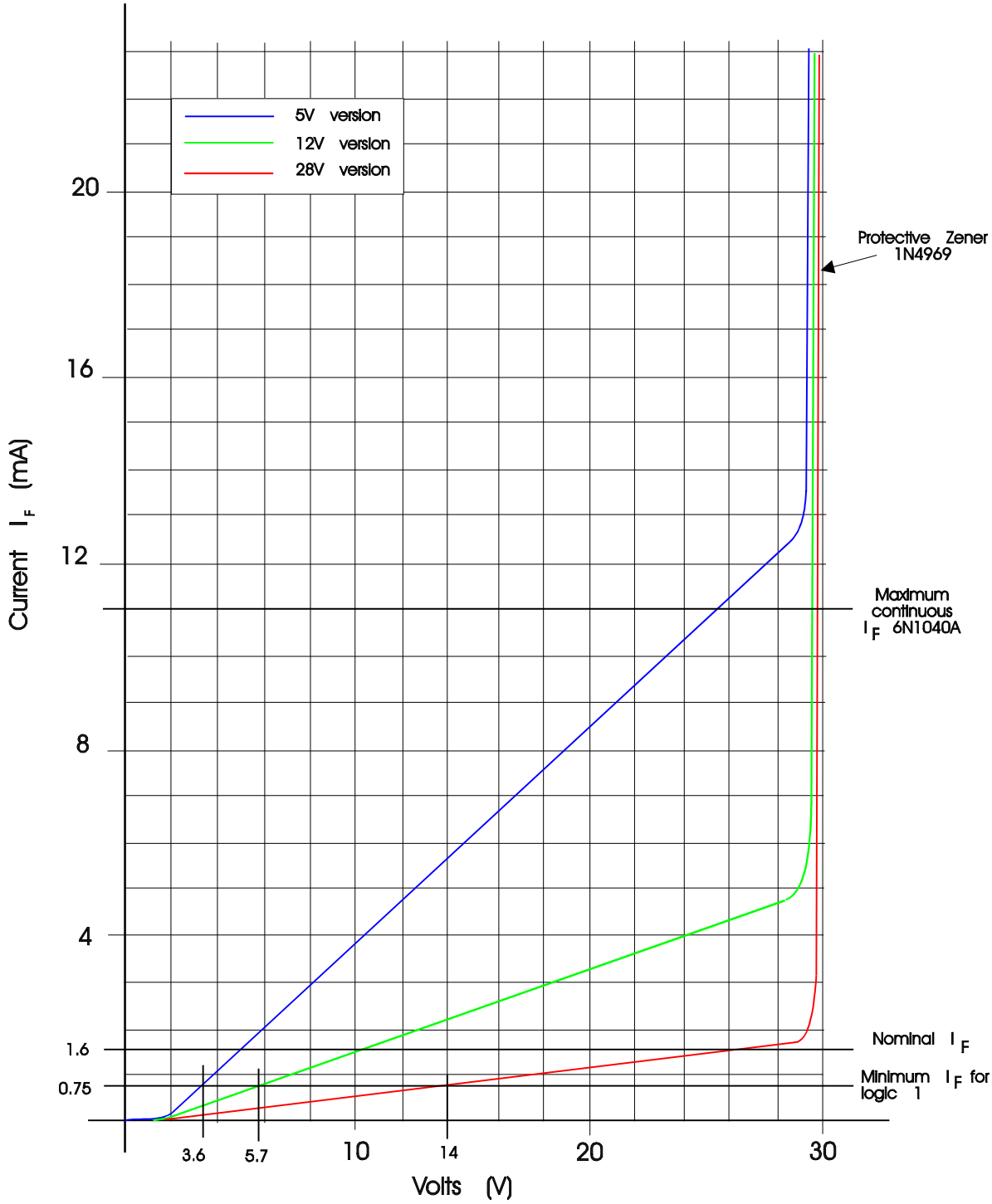


**Figure A-3. Input Circuit Characteristics HCPL 2731 & 6N140A**



# Output Specifications

## PVDZ172 (-40°C to +85°C Ambient Temperature)

| Parameter | Minimum | Typical | Maximum | Units |
|---|---|---|---|---|
| Continuous load current | | | 1.2 | A DC (see note 1) |
| Leakage current at $V_{LOAD}$ = 32 | | | 40 | µA |
| Output voltage drop at $I_{LOAD}$ = 1.2A, T = 85°C | | | 0.65 | V DC |
| Continuous operating voltage | | | 32 | V DC (see note 2) |
| On resistance Rds (on) at Tj = 25°C $I_{LOAD}$ = 100 mA DC | | 0.4 | 0.5 | Ω |
| Reverse voltage | | | 3 | V |
| Turn On time | | | 0.5 | ms (see Fig A-4) |
| Turn Off time | | | 8 | ms (see Fig A-4) |
| $\frac{dV}{dt}$ | 1000 | | | V/µs |
| Output capacitance | | | 1200 | pF |
| Isolation | 250 | | | V DC |

**Notes:**

1 There is no in-built current limiting feature in the PVDZ172.

2 This is limited by the protective zener diode. The relay itself can withstand 60 V DC.

**Figure A-4. Output Turn-on/Turn-off**

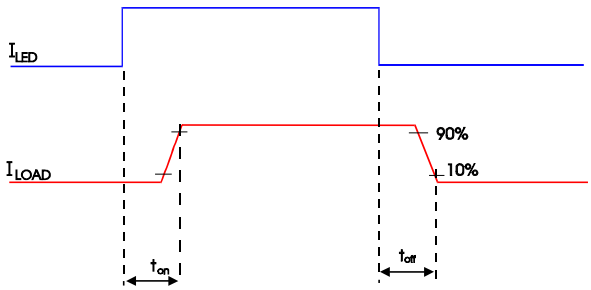
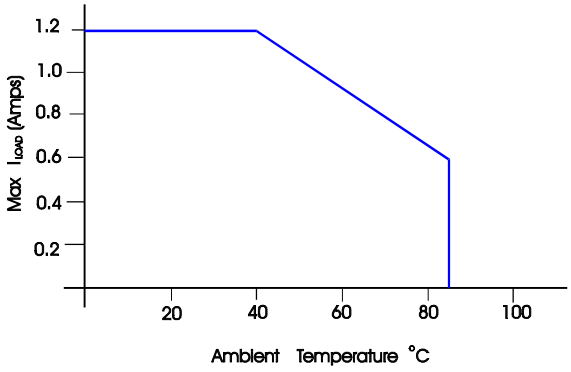



**Figure A-5. Thermal Derating Curve**



## CD20CD (-55°C to +105°C Ambient Temperature)

| Parameter | Minimum | Typical | Maximum | Units |
|---|---|---|---|---|
| Continuous load current | | | 1.0 | A DC (see note) |
| Leakage current at $V_{LOAD}$ = 32 | | | 50 | µA |
| Output voltage drop at $I_{LOAD}$ = 1.2A, T = 85°C | | | 0.8 | V DC |
| Continuous operating voltage | | | 32 | V DC (see note) |
| On resistance Rds (on) at Tj = 25°C $I_{LOAD}$ = 100 mA DC | | 0.52 | 0.65 | Ω |
| Reverse voltage | | | 3 | V |
| Turn On time | | | 1.5 | ms (see Fig A-6) |
| Turn Off time | | | 8 | ms (see Fig A-6) |
| $\frac{dV}{dt}$ | 100 | | | V/µs |
| Output capacitance | | | 1200 | pF |
| Isolation | 250 | | | V DC |

**Note:** This is limited by the protective zener diode. The relay can withstand 60 V DC continuous.

**Figure A-6. Output Turn-on/Turn-off Timing**

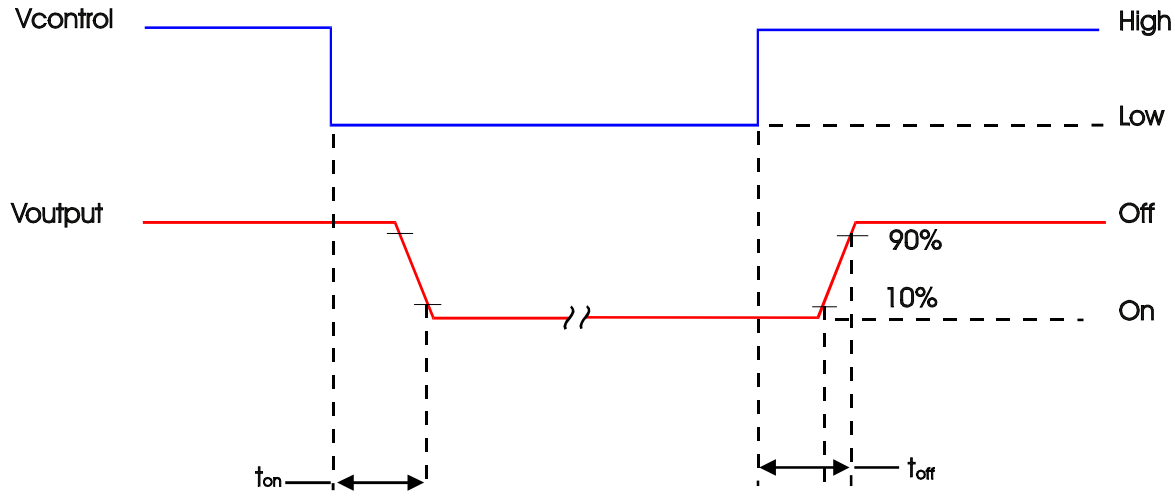


**Figure A-7. Thermal Derating Curve**

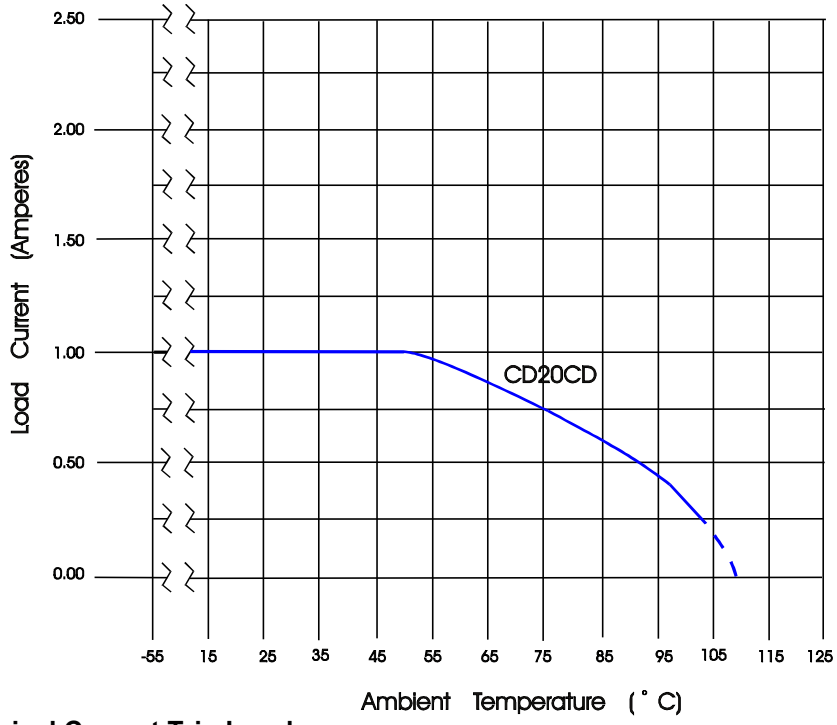
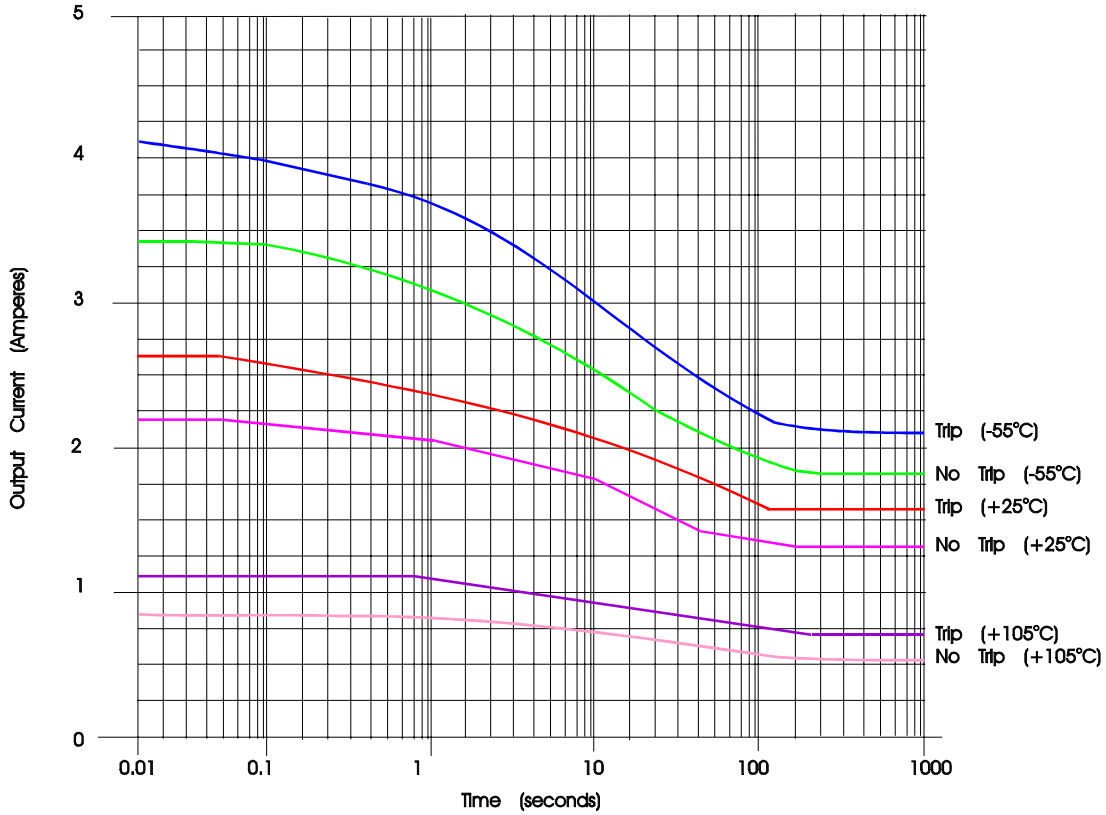



**Figure A-8. Typical Current Trip Levels**



# Appendix B - Example 'C' Code Subroutines

These example 'C' code sub-routines make use of a structure called **pio2cmdblk,** which declares the various registers that make up the PIO-2 memory map. This structure assumes that accesses to the PIO-2 are made as bytes, which is the preferred method.

```
typedef unsigned long Ulong;
typedef unsigned char Uchar;
typedef unsigned short Ushort

struct pio2cmdblk {
      Uchar ch0_7;
      Uchar ch8_15;
      Uchar ch16_23;
      Uchar ch24_31;
      unsigned :32;
      Uchar intstat0_7;
      Uchar intstat8_15;
      Uchar intstat16_23;
      Uchar intstat24_31;
      Uchar cntintstat;
      unsigned :24;
      Uchar intmask0_3;
      Uchar intmask4_7;
      Uchar intmask8_11;
      Uchar intmask12_15;
      Uchar intmask16_19;
      Uchar intmask20_23;
      Uchar intmask24_27;
      Uchar intmask28_31;
      Uchar ctrlreg;
      Uchar intvec;
      unsigned :32;
      unsigned :16;
      Uchar counter0;
      unsigned :8;
      Uchar counter1;
      unsigned :8;
      Uchar counter2;
      unsigned :8;
      Uchar cntctrl0_2;
      unsigned :8;
      Uchar counter3;
      unsigned :8;
      Uchar counter4;
      unsigned :8;
      Uchar counter5;
      unsigned :8;
      Uchar cntctrl3_5;
      unsigned :8;
      Uchar IDreg;
      unsigned :32;
      unsigned :32;
      unsigned :32;
      unsigned :24;
      }     *cmdblk;
```

The following example is a reset routine, a common requirement when writing code for I/O cards. This routine clears all the outputs, disables and clears out all interrupts and programs the six counter/timers into an inactive mode.

```
void reset_pio2() }

/* Clear out all output registers, disable all input channel
   interrupts, clear control register (i.e. disable VME interrupts,
   turn on front panel LED and disable loopback mode), and clear
   board's VME interrupt vector for completeness */

      cmdblk->ctrlreg = 0;
      cmdblk->ch0_7 = 0;
      cmdblk->ch8_15 = 0;
      cmdblk->ch16_23 = 0;
      cmdblk->ch24_31 = 0;
      cmdblk->intmask0_3 = 0;
      cmdblk->intmask4_7 = 0;
      cmdblk->intmask8_11 = 0;
      cmdblk->intmask12_15 = 0;
      cmdblk->intmask16_19 = 0;
      cmdblk->intmask20_23 = 0;
      cmdblk->intmask24_27 = 0;
      cmdblk->intmask28_31 = 0;
      cmdblk-> = 0;

/* Initialize counters by programming them into a mode of operation
   where they will not generate interrupts (mode 5 in this case) */

      cmdblk->cntctrl0_2 = 0x3A
      cmdblk->counter0 = 0;
      cmdblk->counter0 = 0;
      cmdblk->cntctrl0_2 = 0x7A
      cmdblk->counter1 = 0;
      cmdblk->counter1 = 0;
      cmdblk->cntctrl0_2 = 0xBA
      cmdblk->counter2 = 0;
      cmdblk->counter2 = 0;
      cmdblk->cntctrl3_5 = 0x3A
      cmdblk->counter3 = 0;
      cmdblk->counter3 = 0;
      cmdblk->cntctrl3_5 = 0x7A
      cmdblk->counter4 = 0;
      cmdblk->counter4 = 0;
      cmdblk->cntctrl3_5 = 0xBA
      cmdblk->counter5 = 0;
      cmdblk->counter5 = 0;

/* Clear out any interrupts remaining from input channel and/or
   counter activity before initialisation. There may be up to 2
   interrupts pending for each */

      while (cmdblk->intstat0_7) {}
      while (cmdblk->intstat8_15) {}
      while (cmdblk->intstat16)23) {}
      while (cmdblk->intstat24_31) {}
      while (cmdblk->cntintstat) {}

}
```

The counter/timers are designed to provide simple time-interval generation, terminating with an interrupt to the host processor over the VMEbus. The following example contains a sample counter/timer routine that is intended to generate such intervals. This routine requires three parameters:

**cntr** - the counter to be used (in the range 0 to 5)
**mode** - the counter mode to use (mode 0 is recommended)
**count** - a 16 bit count value representing the number of 1.64ms time periods - 1

For this routine to generate a VME interrupt properly, the VME Interrupt Vector and BIT/VME Interrupt Control Registers must be set up before it is run.

When setting or clearing bits in the data registers, remember that when written to, these bits control the solid state relays (if any) and when read, reflect the status of the opto-couplers (if any). This means that for an output only configuration where only a few bits within a data register are to be set/cleared, this must first be carried out on a RAM copy of the data register ***before*** the actual register is written to, if the other bits of the data register are not to be affected.

```
void counter (Uchar cntr, Uchar mode, Ushort count)

{

      Uchar cntlo, cnthi;
      typedef union {
                  Uchar complete;
                  struct {
                        unsigned counter :2;
                        unsigned RWmode :2;
                        unsigned mode :3;
                        unsigned bcd :1;
                  } bit;
      } TIMERCTRL;

      TIMERCTRL ctrlbyte;
      cmdblk = (struct pio2cmdblk *)BASEADD;
      ctrlbyte.bit.bcd = 0;
      ctrlbyte.bit.RWmode = 3;
      switch (mode) {
      case  0:
            ctrlbyte.bit.mode = 0
            break;
      case  1:
            ctrlbyte.bit.mode = 1
            break;
      case  2:
            ctrlbyte.bit.mode = 2
            break;
      case  3:
            ctrlbyte.bit.mode = 3
            break;
      case  4:
            ctrlbyte.bit.mode = 4
            break;
      case  5:
            ctrlbyte.bit.mode = 5
            break;
      }
```

```
        switch (cntr) {
        case  0:
              ctrlbyte.bit.counter = 0;
              break;
        case  1:
              ctrlbyte.bit.counter = 1;
              break;
        case  2:
              ctrlbyte.bit.counter = 2;
              break;
        case  3:
              ctrlbyte.bit.counter = 3;
              break;
        case  4:
              ctrlbyte.bit.counter = 4;
              break;
        case  5:
              ctrlbyte.bit.counter = 5;
              break;
        }

        cntlo = (Uchar) count;
        cnthi = (Uchar) (count >> 8);
        if (cntr <= 2) cmdblk->cntctrl0_2 = ctrlbyte.complete;
              else cmdblk->cntctrl3_5 = ctrlbyte.complete;
        switch (cntr) {
        case 0:
              cmdblk->counter0 = cntlo;
              cmdblk->counter0 = cnthi;
              break;
        case 1:
              cmdblk->counter1 = cntlo;
              cmdblk->counter1 = cnthi;
              break;
        case 2:
              cmdblk->counter2 = cntlo;
              cmdblk->counter2 = cnthi;
              break;
        case 3:
              cmdblk->counter3 = cntlo;
              cmdblk->counter3 = cnthi;
              break;
        case 4:
              cmdblk->counter4 = cntlo;
              cmdblk->counter4 = cnthi;
              break;
        case 5:
              cmdblk->counter5 = cntlo;
              cmdblk->counter5 = cnthi;
              break;
        }
}
```

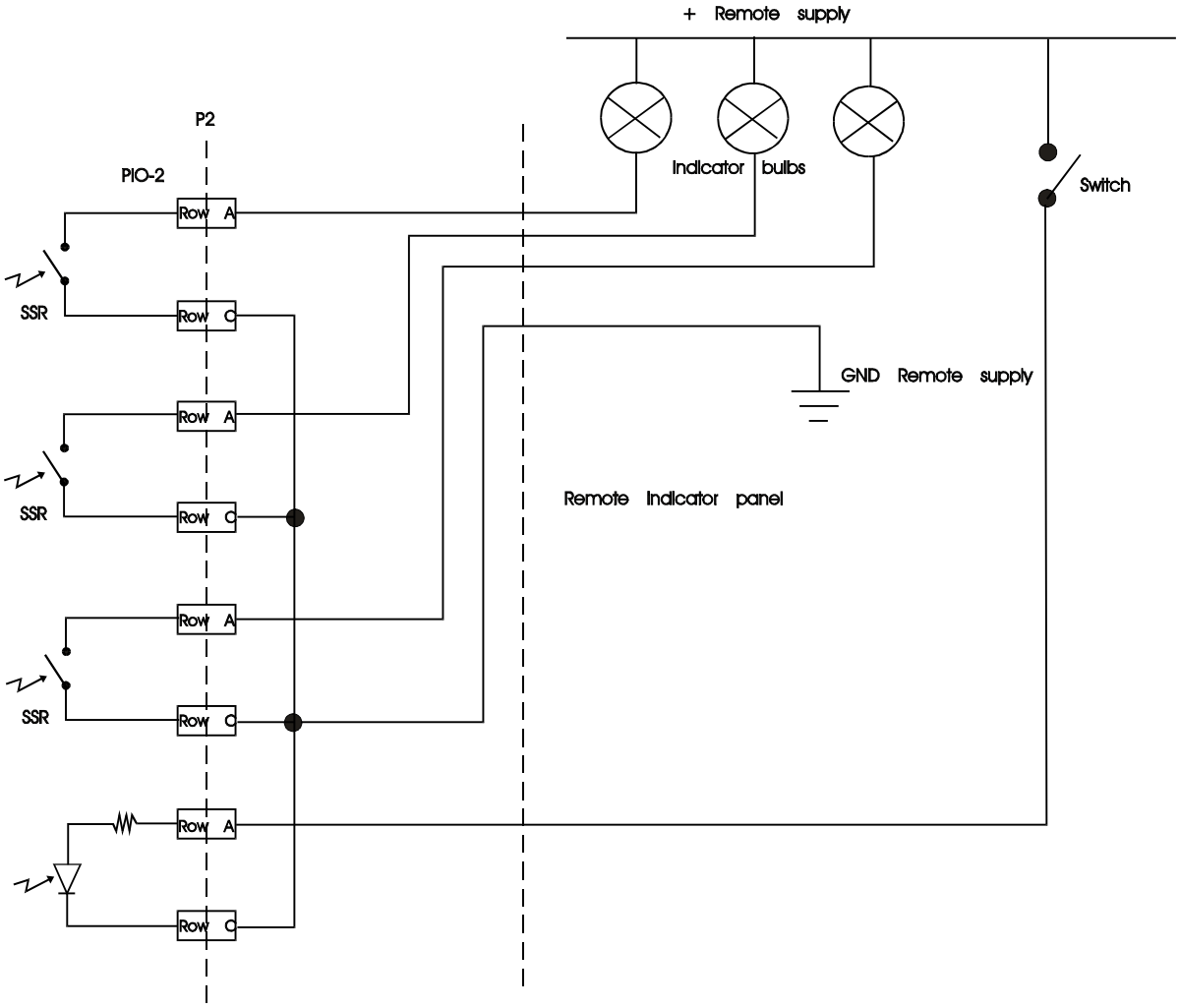
# Appendix C - Typical Load Connection Schemes

## Example Circuits

The following example circuits are included to show typical load connection schemes.

### Example 1

This example describes connection to a simple remote indicator panel containing a few switches and indicator bulbs and their connection to the remote DC supply (+28V, +12V, +5V).

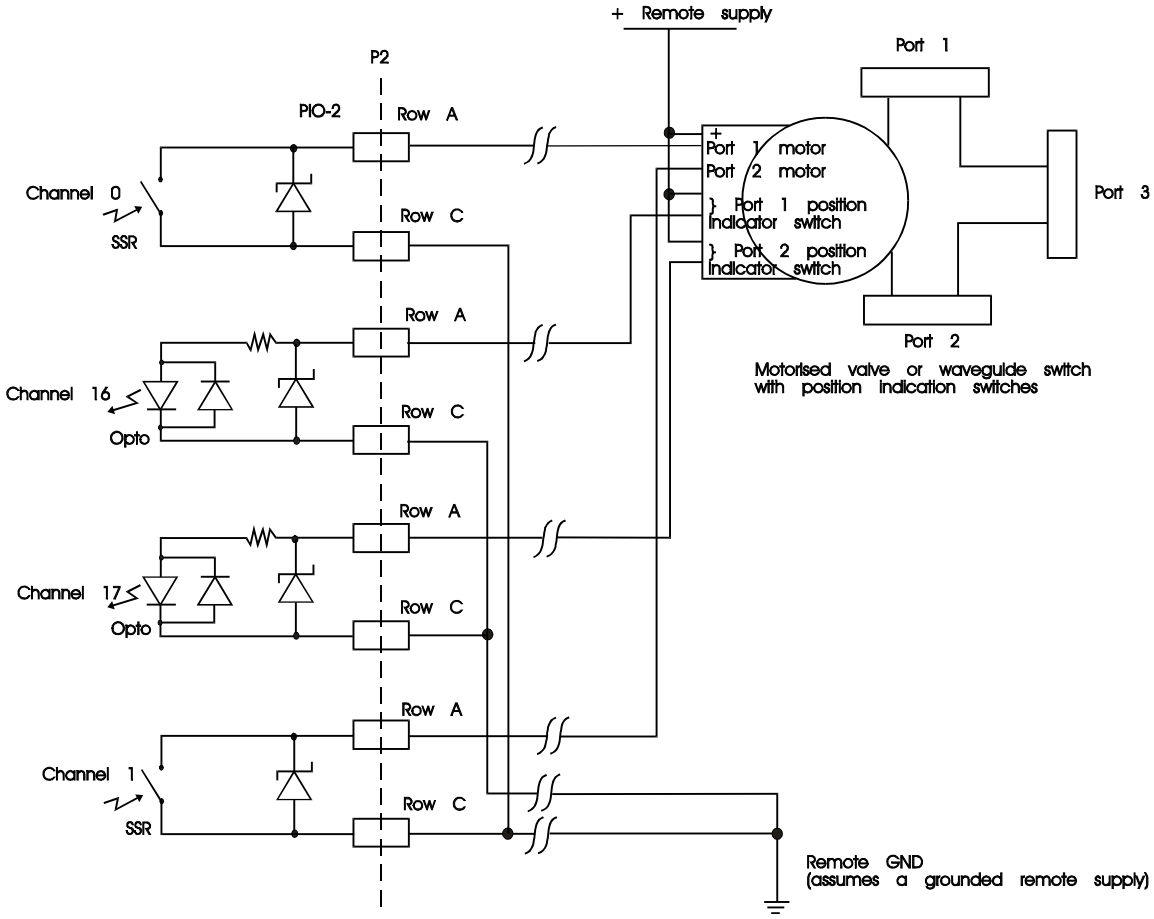



**Note**: To maintain the full 250V isolation, the remote supply GND must be connected in the vicinity of the remote indicator panel and ***not*** at the PIO-2/VME rack.

## Example 2

This example describes a typical setup for driving a motorised valve, where any one of two ports can be connected to a third port, a common setup in hydraulics, pneumatics and waveguide switches for example.

Channel 0 is an output and used to connect port 1 to port 3
Channel 1 is an output and used to connect port 2 to port 3
Channel 16 is an input and used to sense when port 1 is connected to port 3
Channel 17 is an input and used to sense when port 2 is connected to port 3




**Notes:**

1) Separate return wires have been used for the relays and opto-couplers. If the relay current is no greater than a few 10s of mA, the return wires could be commoned.

2) The configuration shown is for a low-side switch. A high-side switch could be implemented just as easily.

3) If the motor/solenoid requires more than 1A drive current, it is possible to connect outputs in parallel to provide the extra capability. Typically if two relays are paralleled, both would be turned on simultaneously to provide the peak initial current for the motor/solenoid, then one would be turned off several milliseconds later leaving the other one to provide the hold current.