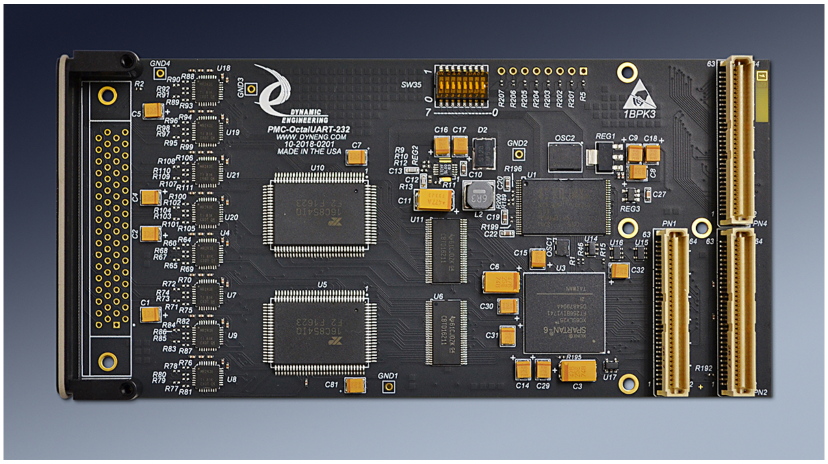
DYNAMIC ENGINEERING
150 DuBois St., Suite C Santa Cruz, CA 95060
(831) 457-8891
https://www.dyneng.com
sales@dyneng.com
Est. 1988

User Manual

PMC-OctalUART-232 Hardware Manual

8 Port RS-232 UART Interface



"-RIO" version shown

Corresponding Hardware: 10-2018-0201
Manual Revision 1p2

PMC-OctalUART-232

8-Channel UART Interface

Dynamic Engineering
150 DuBois St., Suite B/C
Santa Cruz, CA 95060
(831) 457-8891

Manual Revised 4/29/22

This document contains information of proprietary interest to Dynamic Engineering. It has been supplied in confidence and the recipient, by accepting this material, agrees that the subject matter will not be copied or reproduced, in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered.

Dynamic Engineering has made every effort to ensure that this manual is accurate and complete. Still, the company reserves the right to make improvements or changes in the product described in this document at any time and without notice. Furthermore, Dynamic Engineering assumes no liability arising out of the application or use of the device described herein.

The electronic equipment described herein generates, uses, and can radiate radio frequency energy. Operation of this equipment in a residential area is likely to cause radio interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Dynamic Engineering.

Connection of incompatible hardware is likely to cause serious damage.

Table of Contents

List of Figures

Product Description

PMC-OctalUART-232 is part of the Dynamic Engineering family of modular I/O. PMC-OctalUART-232 is a PMC with options for bezel and rear IO, up to 480 KHz. RS-232 signaling, 128 bytes of storage per each of the 8 Tx or Rx ports. Rx, Tx, RTS, CTS, DTR, DSR are supported. *PMC-BiSerial-VI-UART provides a higher performance differential interface should that be required.*

PMC-OctalUART-232 uses a 10 mm inter-board spacing for the front panel, standoffs, and PMC connectors. The 10 mm height is the "standard" height and will work in most systems with most carriers.

Feature Table:
1. 128 byte FIFOs for Rx and Tx data storage per port
2. Industry Standard UART interface with 2 x ‘854 devices installed for the UART ports
3. 8 position Switch for board identification and user purposes
4. Windows driver and reference software. Linux by request.
5. Industrial temperature components [-40 ⇔ +85C]
6. Select between two oscillators for standard and non-standard baud rates.
7. Industry standard pinout with build option to support bezel and rear IO. Resistor selectors eliminate stubs.
8. ESD Protection:
  a) 15 KV Human Body Model
  b) 8 KV IEC61000-4-2 Contact Discharge
  c) 15 KV IEC61000-4-2 Air-Gap Discharge

The following diagram shows the PMC-OctalUART-232 configuration:

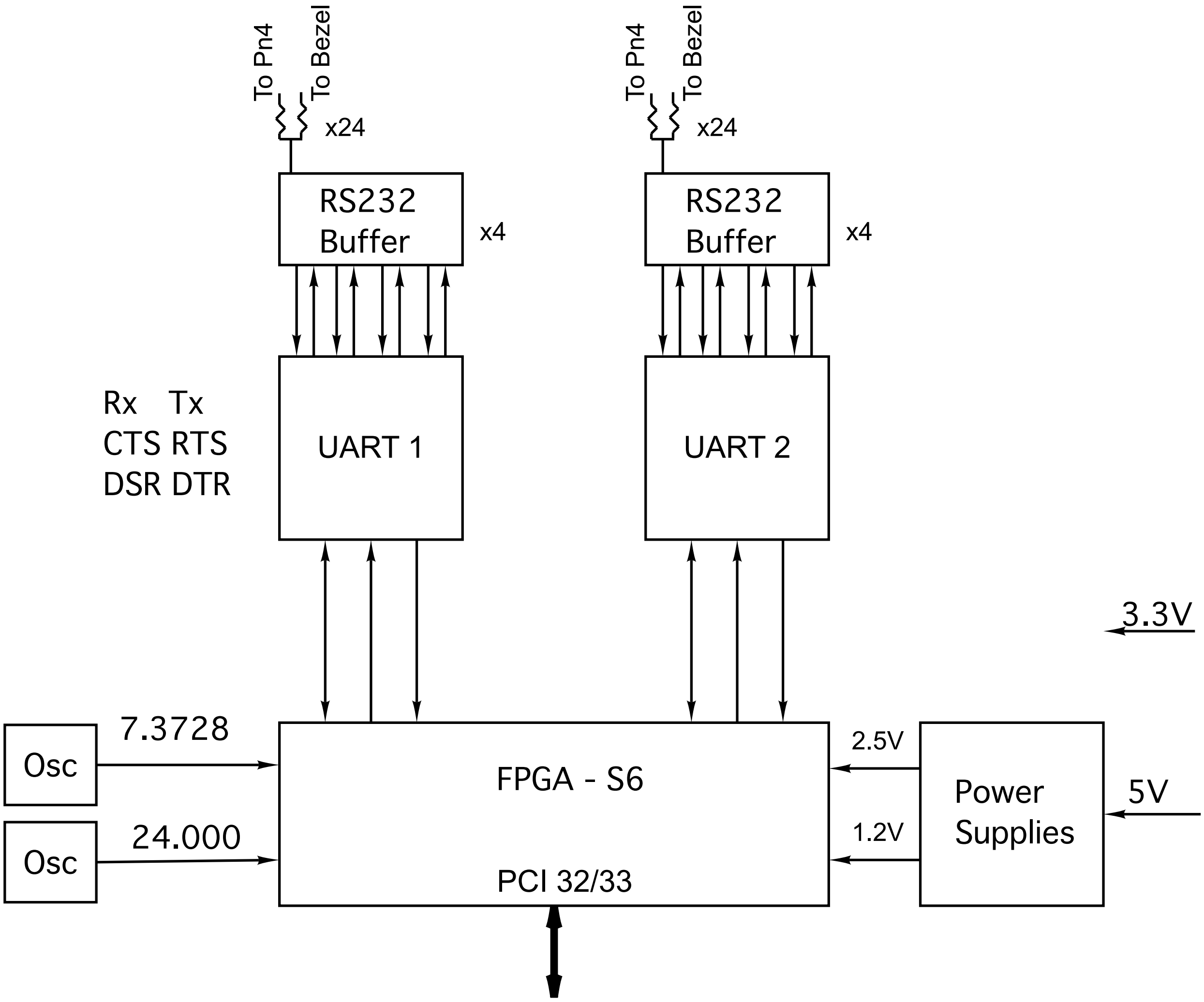


FIGURE 1 PMC-OCTALUART-232 BLOCK DIAGRAM

Please note: The default configuration has bezel IO installed. See ordering options to select rear IO via selection resistors. Both or selective IO are options by special request.

The UART protocol is implemented using industry standard 16C854 devices. Each UART is separately connected to the FPGA to allow for special programming options. The base version of this design uses the industry standard memory map and will port easily into your system. Please see the memory map for the address ranges etc.

Baud rates are programmed for each transmitter and receiver. Two oscillators are supplied with 7.3728 MHz and 24 MHz available to each UART device separately. Further, the UART “C” on each quad UART is supplied with a separate clock also selectable.

Interrupts are maskable within the UARTs and with the master enable. The status register within the FPGA allows for quick detection of which port needs service. In addition the Tx and Rx ready signals are available as status within the FPGA.

The ‘854 devices are byte oriented. The devices have 3 address bits which are generated by the FPGA based on the address presented by the PCI access. To remain common with the industry standard the second nibble is used for this purpose. The result is bytes on LW compatible addresses which works well with PCI systems.

TS3243 RS232 buffers are used with the QFN package which has the advantage of a thermally enhanced package. The parts are capable of 500 Kbits/sec.

PMC-OctalUART-232 is an industrial temperature design.

Dynamic Engineering offers drivers and reference software for Windows®, and Linux. Drivers and reference SW are available AS-IS to clients of the PMC-OctalUART-232. Support contracts are encouraged to help with integration and enhancements. www.dyneng.com/TechnicalSupportFromDE.pdf

Theory of Operation

PMC-OctalUART-232 provides UARTs for transferring data from one point to another using the standard UART transfer protocol.

While UARTs are mature devices.   PMC-OctalUART-232 features the ability to reprogram the FPGA storage FLASH to allow updates via software. A programming adapter is required to use this feature on this HW set.

A logic block within the Xilinx controls the PCI interface to the host CPU. PMC-OctalUART-232 design requires two wait states for read or writes cycles to any address. The wait states refer to the number of clocks after the PCI-core decode before the “terminate with data” state is reached. The additional clock periods account for the clock delay to decode the signals from the PCI bus and to convert to IO controls to the UARTs. The UARTs are operated from the 5V supply to decrease the pulse width requirement allowing operation with fewer wait-states.

There are multiple UARTs each with separate Receiver and Transmitter.   Each pair is organized into a port within the UART.   Frequency of operation [Baud rate], mode of operation, Parity, Stop bits, interrupt conditions are all programmable on a port basis.

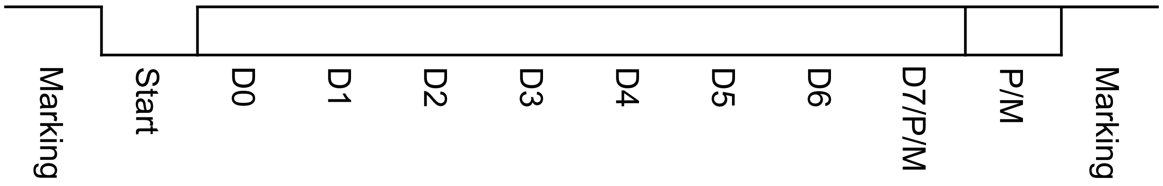


FIGURE 2 UART TRANSFER ENCODING

The Transmit state-machine will transmit a high level followed by the first falling edge of the transmission. The falling edge is the leading edge of the start bit. The start bit is 1 period wide and followed by the first data bit [LSB] of the byte being transmitted. D1-D6 follow. If the UART is programmed for 8 bit data the next period is D7. If programmed for 7 bit data the next position can be Parity if that is enabled or the marking state. The shortest transfer of a byte is 7 bit data, no parity and 1 stop bit for a total of 1[start]+7[data]+1[stop] = 9 bits. If 8 bit data is selected and parity is enabled the length becomes 1+8+1+1 = 11 bits. If 2 stop bits are selected an extra clock period is inserted between byte transfers.

The receiver does not have a clock to work with and uses over-sampling to detect the transitions and the programmed expected transfer rate to count into the bit periods to determine the bit value.

Parity can be programmed to be odd, even or forced. When odd the parity bit is set/cleared to make the number of 1's odd. For example if the data is "AA" an even number of bits are set in the data so the parity would be set "0 01010101 1 1" would be the string with start, data, parity and stop shown. Please note the lsb first nature of the data. The spaces are added for clarity. For even parity the reverse is true, with parity set/cleared to make the total of the data and parity fields an even count.

In addition to the framing and parity errors, FIFO over-run is flagged. When the Rx FIFO is full and a write is attempted the error is captured. A full FIFO will not accept the new write so that data is lost.

Break characters are detected by the RX state-machine. Data received, Break, Frame, Parity, Data over-run status can all be programmed to generate an interrupt request.

When Break or Frame is detected the receiver resynchronizes before looking for new characters. With parity errors the error is flagged and processing continues without resynchronization.

Over reading the Rx FIFO is not an error condition. The FIFO will continue to provide the last read data multiple times. The FIFO count should be read, prior to doing read multiple commands to prevent under-run.

On the Tx side an empty FIFO causes the transmitter to go to the marking state once the last word read has been transmitted. When more data is available that data will be transmitted. No under-run error is generated for this situation.

A break condition can be generated from the Tx control register within the UART.

Many other features are supported with the UART employed by PMC-OctalUART-232. The previous description was meant to summarize the more commonly used features. For all of the options please refer to the EXAR 16C854 data sheet.

Programming

Programming PMC-OctalUART-232 board requires only the ability to read and write data from the host. The base address is determined during system configuration of the PCI bus. The base address refers to the first user address for the slot in which the board is installed. The VendorId = 0xDCBA. The CardId = 0x0066.

In order to transfer data to another UART, several steps must be performed. First a physical connection must be established with the appropriate interface cable. The port of interest must be programmed with the appropriate UART parameters for transmit and or receive operations. Each port has a separate register set for control bits, baud rate and other parameters. Once programmed you can load data into the Tx buffer for transmission or read from the Rx when data becomes available.

Be sure to select the correct mode of operation, and note the Rx and Tx do not need to be the same.

The hardware supports several modes of operation. Choose the right mode based on your environment. For example if you are operating with a console program and need to remain compatible with other standard UART's the "8N1" mode will typically be the right choice.

The baud rate is programmable, and should be set correctly. The jitter tolerance will allow slightly off frequencies to work, but will effectively have no jitter tolerance when operating in this manner.

Firmware Updates

Revision 1.1: First release

PCB Updates

Revision 01: First release
Revision 02: Misc Clean-up
Revions 03 : Replace EOL FLASH with QSPI device

Base Address Map

```
          Register Name                Offset         Description

#define     UART 1                     0x0000 // Port 1 base offset
#define     UART 2                     0x0100 // Port 2 base offset
#define     UART 3                     0x0200 // Port 3 base offset
#define     UART 4                     0x0300 // Port 4 base offset

#define     UartCntl                   0x0500 // Control Register also @ x1500
#define     FR1_4                      0x0600 // FIFO Ready ports 1-4
#define     IntVector                  0x0700 // Interrupt Status also @ x1700

#define     BaseRev                    0x0800// Revision and Switch
#define     BaseCntl                   0x0900// additional control register for enhanced features

#define     UART 5                     0x1000 // Port 5 base offset
#define     UART 6                     0x1100 // Port 6 base offset
#define     UART 7                     0x1200 // Port 7 base offset
#define     UART 8                     0x1300 // Port 8 base offset

#define     FR5_8                      0x1600 // FIFO Ready ports 5-8
```

FIGURE 3 PMC-OCTALUART-232 BASE ADDRESS MAP

UART Port Address Map

Register Name	Offset	Description
DATA	0x0000	UART read/write data
IEN	0x0010	UART write interrupt enable
ISTAT	0x0020	UART read interrupt status
FCNTL	0x0020	UART write FIFO control
LCNTL	0x0030	UART write line control
MCNTL	0x0040	UART write modem control
LSTAT	0x0050	UART read line status
MSTAT	0x0060	UART read modem status
SPAD	0x0070	UART read/write scratchpad
UART baud rate register defines (enabled when *LCNTL bit-7 = 1, ≠ 0xbf)		
DLL	0x0000	UART read/write LSB divisor
DLM	0x0010	UART read/write MSB divisor
UART enhanced register offsets (enabled when *LCNTL = 0xbf)		
FTC	0x0000	UART FIFO read count/write trigger level
FEAT	0x0010	UART write feature control
ENF	0x0020	UART read/write enhanced features
XON1	0x0040	UART read/write Xon-1 word
XON2	0x0050	UART read/write Xon-2 word
XOFF1	0x0060	UART read/write Xoff-1 word
XOFF2	0x0070	UART read/write Xoff-2 word
UART EMS registers (enabled when *FEAT bit-6 = 1, *LCNTL bit-7 = 0)		
EMS	0x0070	UART write enhanced mode select
FLV	0x0070	UART read FIFO level counter

FIGURE 4 PMC-OCTALUART-232 UART CHANNEL ADDRESS MAP

There are 8 UART Ports. Each port/channel has a separate set of control registers as shown in Figure 3. The offset within each of the ports is shown in Figure 4.

The base address for PMC-OctalUART-232 is set by the system. For Base features the base address is added to the base feature offset. For Port features the base address is added to the Port Offset and to the Port Feature. In some cases the UART is indirect with register configurations affecting the meaning of a particular address. Address = Base + Port Offset + Port Feature. All addresses are on LW boundaries and all accesses affect the entire LW. Writing a byte still affects the other three bytes.

Register Definitions

UartCntl

Base Control Register (read/write)

Base Control

Data Bit	Description
7-5	Spare
4	UART Reset
3	Osc Sel
2	Interrupt Enable
1	UART 2 ClockSEL
0	UART 1 ClockSEL

FIGURE 5 PMC-OCTALUART-232 UART CONTROL REGISTER

All bits are active high and are reset on system power-up or reset.

UART x ClockSEL When set to a one, the corresponding ClockSEL control to the UART is set. This signal is sampled by the UART when removed from reset or power up. ‘1’ uses a divide by 1 and ‘0’ for divide by 4 for the baud rate generator clock. Also can be controlled via MCR[7].

Interupt Enable: This bit acts as the Master Interrupt Enable. When set ‘1’ UART interrupts are enabled to the system. When ‘0’ the UART interrupt requests are available as status but are not driven to the system.

Osc Sel: This bit selects between the 7.3728 and 24.000 MHz reference clocks for the UART. ‘0’ corresponds to the 7.3728 and ‘1’ to the 24.000 MHz. Please note: bit 7 is returned with a ‘1’ to indicate the second oscillator is installed.

UART Reset: This bit provides direct control over the UART reset line. This bit must be returned to ‘0’ for normal operation. When changing external inputs to the UART [ClockSEL, Osc SEL] it is necessary to assert and deassert reset to insure proper operation. A 40 nS pulse is required. Back to back set and clear instructions can be used.

FR1_4, FR5_8

UART FIFO Status (read)

Base Control

Data Bit	**Description**
7	RxRdyD
6	RxRdyC
5	RxRdyB
4	RxRdyA
3	TxRdyD
2	TxRdyC
1	TxRdyB
0	TxRdyA

FIGURE 6 PMC-OCTALUART-232 UART FIFO READY STATUS

TxRdyA-D and RxRdyA-D are registered and presented within the status. This decription covers both UART ports [offsets x0060 and x0160]

RXRDY_A - D when ‘0’ indicates data is present in the Receive Holding Register for the corresponding UART channel. When ‘1’ indicates that no data is available to be read. The status change is based on the programmed threshold.

TXRDY_A - D when ‘0’ indicates there is room for data to be written into the Transmit Holding Register for the corresponding UART channel. When ‘1’ indicates that no more data can be written. The status change is based on the programmed threshold.

IntVector

UART FIFO Status (read)

Base Control

Data Bit	**Description**
8	Force Int
7	Port8 Int
6	Port7 Int
5	Port6 Int
4	Port5 Int
3	Port4 Int
2	Port3 Int
1	Port2 Int
0	Port1 Int

FIGURE 7 PMC-OCTALUART-232 UART INTERRUPT STATUS

Each of the 8 UART ports has the capability of creating an interrupt. Depending on how the UART is programmed the interrupt will be asserted or not. INTSEL and 16/68 are set to ‘1’ which puts the device in “Intel mode” with continuously driven interrupts.

If the master interrupt enable [see UartCntl] is set the interrupt requests are converted to system interrupt requests. If cleared this port can be used for polling.

Force Int is shown on bit 8 to allow a single read to determine the cause of the interrupt.

BaseRev

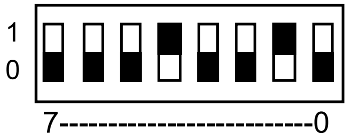
Base Revision Register (read)

Base General Purpose Register

```
#define    BASE_STAT_SW_MASK     0x000000FF // 7-0 are switch bit when installed
#define    BASE_STAT_REV_MAJ     0x0000FF00 // Design major revision
#define    BASE_STAT_REV_MIN     0x00FF0000 // Design minor revision
```

FIGURE 8 PMC-OCTALUART-232 BASE REVISION REGISTER

Switch 7-0: The user switch is read through this port. The bits are read as the lowest byte. Access the read-only port as a long word and mask off the undefined bits. The dip-switch positions are defined in the silkscreen. For example the switch figure below indicates a 0x12. The switch is an optional item. Bits have no meaning if not installed.



The Major Revision is used to track FLASH releases to the client. The revision will be updated when official releases to clients occur to allow the client to tell if a board has been updated. Currently 1.

The Minor Revision is used to track FLASH updates during development and for unofficial releases to clients. This revision may roll over depending on the number of iterations needed. Currently 1.

BaseCntl

Base Control Register (read/write)

Base Interrupt Status

8	Interrupt Active [ready only]
7-1	Spare
0	ForceInt

FIGURE 9 PMC-OCTALUART-232 BASE CONTROL STATUS

ForceInt when set '1' causes PMC-OctalUART-232 to create an interrupt request to the system. Clearing '0' removes the interrupt request. This is an masked bit meaning the master interrupt enable does have to be set to use. This bit can be used for SW development to create an interrupt on demand and check operation.

Spare bits are RW in the register and are reserved for future functionality.

Interrupt Active is the OR of the UART interrupt request bits after the mask. If set the board is requesting an interrupt.

LOOP-BACK & IO Connection Definitions

PMC-OctalUART-232 can be used with direct end point cabling or with an interface. Dynamic Engineering uses HDEterm68 along with loop-back connections to accomplish loop-back.

The following table shows the connections the HDEterm68 used in the loop-back test. Each UART uses 6 IO to create the TX, RX, RTS, CTS, DTR, and DSR connections. The reference SW uses loop-back within the same channel as a test mechanism.

Pins shown for P1 SCSI connector and match on HDEterm68
Numbers shown P1/Pn4 . For loop-back connections with rear IO use this table plus the Rear IO mapping table from the PMC carrier.

<u>UART 1</u>

UART1_TXA	2/2	UART1_RXA	1/1
UART1_RTSA	4/4	UART1_CTSA	3/3
UART1_DTRA	6/6	UART1_DSRA	5/5

<u>UART 2</u>

UART1_TXB	11/10	UART1_RXB	10/9
UART1_RTSB	13/12	UART1_CTSB	12/11
UART1_DTRB	15/14	UART1_DSRB	14/13

<u>UART 3</u>

UART1_TXC	19/18	UART1_RXC	18/17
UART1_RTSC	21/20	UART1_CTSC	20/19
UART1_DTRC	23/22	UART1_DSRC	22/21

<u>UART 4</u>

UART1_TXD	28/26	UART1_RXD	27/25
UART1_RTSD	30/28	UART1_CTSD	29/27
UART1_DTRD	32/30	UART1_DSRD	31/29

UART 5

UART2_TXA	36/34	UART2_RXA	35/33
UART2_RTSA	38/36	UART2_CTSA	37/35
UART2_DTRA	40/38	UART2_DSRA	39/37

UART 6

UART2_TXB	45/42	UART2_RXB	44/41
UART2_RTSB	47/44	UART2_CTSB	46/43
UART2_DTRB	49/46	UART2_DSRB	48/45

UART 7

UART2_TXC	53/50	UART2_RXC	52/49
UART2_RTSC	55/52	UART2_CTSC	54/51
UART2_DTRC	57/54	UART2_DSRC	56/53

UART 8

UART2_TXD	62/58	UART2_RXD	61/57
UART2_RTSD	64/60	UART2_CTSD	63/59
UART2_DTRD	66/62	UART2_DSRD	65/61

P1 Grounds : 8, 9, 17, 25, 26 34, 42, 43, 51, 59, 60, 68.
The shield on P1 can be grounded [default], AC coupled, or open.

Pn4 Grounds: 8,16,24, 32, 40, 48, 56, 64

Dynamic Engineering Drivers and Reference SW include loop-back tests using the above connections.

PMC PCI Pn1 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn1 Interface. See the User Manual for your carrier board for more information. Unused pins may be assigned by the specification and not needed by this design.

TCK	-12V	1	2
GND	INTA#	3	4
		5	6
BUSMODE1#	+5V	7	8
		9	10
GND		11	12
CLK	GND	13	14
GND		15	16
	+5V	17	18
	AD31	19	20
AD28	AD27	21	22
AD25	GND	23	24
GND	C/BE3#	25	26
AD22	AD21	27	28
AD19	+5V	29	30
	AD17	31	32
FRAME#	GND	33	34
GND	IRDY#	35	36
DEVSEL#	+5V	37	38
GND	LOCK#	39	40
		41	42
PAR	GND	43	44
	AD15	45	46
AD12	AD11	47	48
AD9	+5V	49	50
GND	C/BE0#	51	52
AD6	AD5	53	54
AD4	GND	55	56
	AD3	57	58
AD2	AD1	59	60
	+5V	61	62
GND		63	64

FIGURE 10 PMC-OCTALUART-232 PN1 INTERFACE

PMC PCI Pn2 Interface Pin Assignment

The figure below gives the pin assignments for the PMC Module PCI Pn2 Interface. See the User Manual for your carrier board for more information. Unused pins may be assigned by the specification and not needed by this design.

+12V		1	2
TMS	TDO	3	4
TDI	GND	5	6
GND		7	8
		9	10
	+3.3V	11	12
RST#	BUSMODE3#	13	14
+3.3V	BUSMODE4#	15	16
	GND	17	18
AD30	AD29	19	20
GND	AD26	21	22
AD24	+3.3V	23	24
IDSEL	AD23	25	26
+3.3V	AD20	27	28
AD18		29	30
AD16	C/BE2#	31	32
GND		33	34
TRDY#	+3.3V	35	36
GND	STOP#	37	38
PERR#	GND	39	40
+3.3V	SERR#	41	42
C/BE1#	GND	43	44
AD14	AD13	45	46
GND	AD10	47	48
AD8	+3.3V	49	50
AD7		51	52
+3.3V		53	54
	GND	55	56
		57	58
GND		59	60
	+3.3V	61	62
GND		63	64

FIGURE 11 PMC-OCTALUART-232 PN2 INTERFACE

Applications Guide

Interfacing

Some general interfacing guidelines are presented below. Do not hesitate to contact the factory if you need more assistance.

ESD
Proper ESD handling procedures must be followed when handling the PMC-OctalUART-232. The card is shipped in an anti-static, shielded bag. The card should remain in the bag until ready for use. When installing the card the installer must be properly grounded and the hardware should be on an anti-static workstation.

Start-up
Make sure that the "system" can see your hardware before trying to access it. Many BIOS will display the PCI devices found at boot up on a "splash screen" with the VendorID and CardId and an interrupt level. Look quickly, if the information is not available from the BIOS then a third party PCI device cataloging tool will be helpful.

Watch the system grounds
All electrically connected equipment should have a fail-safe common ground that is large enough to handle all current loads without affecting noise immunity. Power supplies and power consuming loads should all have their own ground wires back to a common point.

We provide the components. You provide the system. Only careful planning and practice can achieve safety and reliability. Inputs can be damaged by static discharge, or by applying voltage outside of the device rated voltages.

Construction and Reliability

Dynamic Engineering Modules are conceived and engineered for rugged industrial environments. PMC-OctalUART-232 is constructed out of 0.062-inch thick High-Temp ROHS compliant FR4 material.

ROHS and standard processing are available options.

Through-hole and surface-mount components are used. PMC connectors are rated at 1 Amp per pin, 100 insertion cycles minimum. These connectors make consistent, correct insertion easy and reliable.

PMCs are secured against the carrier with four screws attached to the 2 stand-offs and 2 locations on the front panel. The four screws provide significant protection against shock, vibration, and incomplete insertion.

The PCB provides a (typical based on PMC) low temperature coefficient of 2.17 W/°C for uniform heat. This is based upon the temperature coefficient of the base FR4 material of 0.31 W/m-°C, and taking into account the thickness and area of the board. The coefficient means that if 2.17 Watts are applied uniformly on the component side, then the temperature difference between the component side and solder side is one degree Celsius.

PMC-OctalUART-232 has internal thermal planes made up of heavy copper power and ground planes. The planes will spread the thermal load over the entire board to minimize hotspots and increase the “coolability”. The components are Industrial temperature rated or better. Where possible devices with thermal ties were chosen to allow direct connection to the ground plane.

Thermal Considerations

The PMC-OctalUART-232 design consists of CMOS circuits. The power dissipation due to internal circuitry is very low. It is possible to create higher power dissipation with the externally connected logic. If more than one Watt is required to be dissipated due to external loading, then forced-air cooling is recommended. With the one degree differential temperature to the solder side of the board, external cooling is easily accomplished.

Warranty and Repair

Please refer to the warranty page on our website for the current warranty offered and options. https://www.dyneng.com/warranty.html

Service Policy

Before returning a product for repair, verify as well as possible that the suspected unit is at fault. Then call the Customer Service Department for a RETURN MATERIAL AUTHORIZATION (RMA) number. Carefully package the unit, in the original shipping carton if this is available, and ship prepaid and insured with the RMA number clearly written on the outside of the package. Include a return address and the telephone number of a technical contact. For out-of-warranty repairs, a purchase order for repair charges must accompany the return. Dynamic Engineering will not be responsible for damages due to improper packaging of returned items. For service on Dynamic Engineering Products not purchased directly from Dynamic Engineering contact your reseller. Products returned to Dynamic Engineering for repair by other than the original customer will be treated as out-of-warranty.

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. Customer approval will be obtained before repairing any item if the repair charges will exceed one half of the quantity one list price for that unit. Return transportation and insurance will be billed as part of the repair and is in addition to the minimum charge.

For Service Contact:

Customer Service Department
Dynamic Engineering
150 Dubois Street, Suite B/C
Santa Cruz, CA 95060
831-457-8891
support@dyneng.com

Specifications

Host Interface (PCI):	PCI Interface 33 MHz. 32-bit
Serial Interfaces:	8 UART channels each with Rx, Tx, RTS, CTS, DTR, DSR signals
TX Bit-rates generated:	user programmable for each UART channel with the standard baud rates up to 500 KHz. and custom programmed rates.
Software Interface:	Control Registers, FIFO’s, and Status Ports
Initialization:	Hardware reset forces all registers to 0 except as noted
Access Modes:	Long-word boundary space (see memory map)
Wait States:	Two for all addresses
Interrupt:	Multiple programmable interrupts per port for flow control and error recognition.
DMA:	not supported on this version
Onboard Options:	All Options are Software Programmable
Interface Options :	Front or Rear IO. Front IO via P1 SCSI connector. Rear IO through Pn4. Isolation resistors to eliminate stubs.
Dimensions:	Standard Single PMC.
Construction:	High Temp ROHS compliant FR4 Multi-Layer Printed Circuit, Through-Hole and Surface-Mount Components
Temperature Coefficient:	2.17 W/^{O}C for uniform heat across PMC [similar for other formats]
Power	5V and 3.3V ⇔ no requirement for +/- 12V.

Order Information

Please refer to our PMC-OctalUART-232 webpage for the most up to date information:
https://www.dyneng.com/PMC-OctalUART-232.html

PMC-OctalUART-232

Standard version with 8 UARTs, each with Rx,Tx, RTS, CTS, DTR, & DSR RS-232 signals supported. Programmable for any standard baud rate up to 500 KHz. Non-standard rates too. Programmable character length[7,8], stop bits[1,2], parity[odd, even, level, none]. 128 byte FIFO per Tx and Rx. Industrial temperature components standard. 7.3728 & 24 MHz references

-AC, -OPN

Change to AC coupling or Open for P1 shield.

-RIO

Change to Pn4 IO instead of SCSI connector.

-FRP

Change to Pn4 & SCSI connector IO.

-CC

Add conformal coating option. Recommended for condensing or near condensing environments

-ROHS

Leaded solder is standard on this product.
Add -ROHS for ROHS processing.

HDEterm68

https://www.dyneng.com/HDEterm68.html is available as a breakout or for loop-back purposes. Available with several options including connector orientation, DIN rails, Terminal Block, header strip.

HDEcabl68

SCSI cable suitable to interconnect PMC-OctalUART-232 and HDEterm68. Available in various lengths. Twisted shielded construction.

Glossary

Acronyms and other specialized names and their meaning:

PMC

PCI Mezzanine Card - establishes common connectors, connections, size and other mechanical features.

PCI

Peripheral Component Interconnect – parallel bus from host to this device.

VendorID

Manufacturers number for PCI/PCIe boards. DCBA is Dynamic Engineering's ID.

CardID

Unique number assigned to design to distinguish between all designs of a particular vendor.

UART

Universal Asynchronous Receiver Transmitter. Common serialized data transfer with start bit, stop bit, optional parity, optional 7/8 bit data. Can be over any electrical interface. RS232 and RS422 are most common.

Baud

Used as the bit period for this document. Not strictly correct but is the common usage when talking about UART's.

FIFO

First In First Out Memory

JTAG

Joint Test Action Group – a standard used to control serial data transfer for test and programming operations.

TAP

Test Access Port – basically a multi-state port that can be controlled with JTAG [TMS, TDI, TDO, TCK]. The TAP States are the states in the State machine controlled by the commands received over the JTAG link.

TMS

Test Mode State – this serial line provides the state switching controls.   ‘1’ indicates to move to the next state, ‘0’ means stay put in cases where delays can happen, otherwise 0,1 are used to choose which branch to take. Due to complexity of state manipulation the instructions are usually precompiled. Rising edge of TCK valid.

TDI

Test Data In - this serial line provides the data input to the device controlled by the TMS commands. For example the data to program the FLASH comes on the TDI line while the commands to the state-machine to move through the necessary states comes over TMS. Rising edge of TCK valid.

TCK

Test Clock provides the synchronization for the TDI, TDO and TMS signals

TDO

Test Data Out is the shifted data out. Valid on the falling edge of TCK. Not all states output data.

Packet

Group of characters transferred.   When the characteristics of a group of characters is known the data can be stored in packets, transferred as such and the system optimized as a result. Any number of characters can be sent.

Packed

When UART characters are always sent/received in groups of 4 allowing full use of host bus / FIFO bandwidth.

UnPacked

When UART characters are sent on an unknown basis requiring single character storage and transfer over the host bus.

MUX

Multiplexor – multiple signals multiplexed to one with a selection mechanism to control which path is active.

Flash

Non-volatile memory used on Dynamic Engineering boards to store FPGA configurations or BIOS.