PCM-3612
PC/104 Dual-port
RS-422/485 Module

PCM-3612 PC/104 RS-422/485

Introduction

The PCM-3612 is a PC/104-compatible RS-422/485 serial interface module. It works with PC/104 CPU modules or CPU cards which accept PC/104 expansion modules. It provides two independent serial interfaces, accessed through two male DB-9 connectors. You can configure both ports for RS-422 or RS-485 operation.

The module's industry-standard 16C550 asynchronous communication chip is fully programmable. The module requires no special commands or control nodes if you use the standard COM1 and COM2 port addresses.

The module's RS-485 function uses an automatic direction control circuit, so you don't need to change any jumpers to switch the module between driver and receiver.

An additional surge protection circuit protects other devices on the RS-485 network.

Features

- Two RS-422/485 interfaces
- Long distance cmmunication — up to 4000 feet (1.2 km)
- High speed data transmission—up to 115,200 bps
- Switch selectable addresses (COM1, COM2, or any other address from hex 200 to 3F8)
- 16 bytes FIFOs
- Jumper selectable interrupt level
- Four LEDs indicate status of TX, RX lines (red LED represents TX, green LED represents RX)

RS-422/485

- Supports TX, RX, RTS and CTS signals
- 2-wire or 4-wire operation
- Auto direction control for RS-422/485
- Surge protection for TX and RX lines (option)

Specifications

- **Dimensions:** 3.775" x 3.550" (9.6 cm x 9.0 cm)
- **Bus:** PC/104
- **Baud rate:** 50 to 115,200 bps
- **Character length:** 5, 6, 7, or 8 bits
- **Parity:** Even, odd, or none
- **Stop bit:** 1, 1.5 (5-bit data only) or 2
- **I/O connectors:** Dual male DB-9
- **Interrupt levels**: IRQ 3, 4, 5, 6, 7, 9, 10, 11, 12 or 15
- **Clock input:** 1.8432 MHz
- **Driver / receiver**
  Device: LTC485
  Differential input threshold: 0.2 V max.
  Hysteresis: 50 mA typical
  Input impedance: < 12 kW without terminators
- **Power consumption (+5 V):** 400mA typical, 950 mA maximum
- **MTBF:** 251319 hrs @ 25° C Ground-Fix environment
- **Operating temperature:** 0 ~ 60° C (refer to IEC-68-1.2.3 item)
- For technical support and service please visit our support website at:
  http://support.advantech.com
  and visit the "IA group" and "FAQ" sections

Part no. 2000361200 1st Edition Printed in Taiwan Nov1998

Initial inspection

We carefully inspected the PCM-3612 both mechanically and electrically before we shipped it. It should be free of marks and scratches and in perfect electrical order upon receipt.

Handle the board only by its edges. The static charge on your body may damage its integrated circuits. Keep the card in its anti-static package whenever it is not installed. You can use this package to return the card if it should need repair.

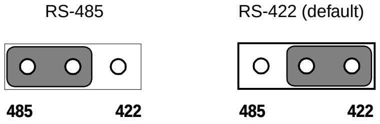
Switches and jumpers

The following chart shows the switches and jumpers corresponding to each serial interface channel:

Ch. 1	Ch.2	Function
SW1	SW2	I/O base address
JP1	JP2	Interrupt level
JP10	JP11	RS-485 OR RS-422
JP5	JP7	Enable mode selection

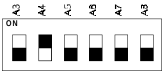
RS-422/485 selection (JP10 and JP11)

You can set each port individually for either RS-422 (the default) or RS-485 operation. Jumper JP10 configures Port 1, and JP11 configures Port 2. The figure below shows the jumper settings.

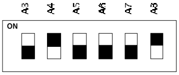


I/O base address (SW1 and SW2)

You set the port base addresses using two six-position DIP switches, one for each port. Switch SW1 sets Port 1, and SW2 sets Port 2. You can choose base addresses from hex 200 to 3F8. The default settings are CH#1 (hex 3E8) for Port 1, and CH#2 (hex 2E8) for Port 2, as shown below:

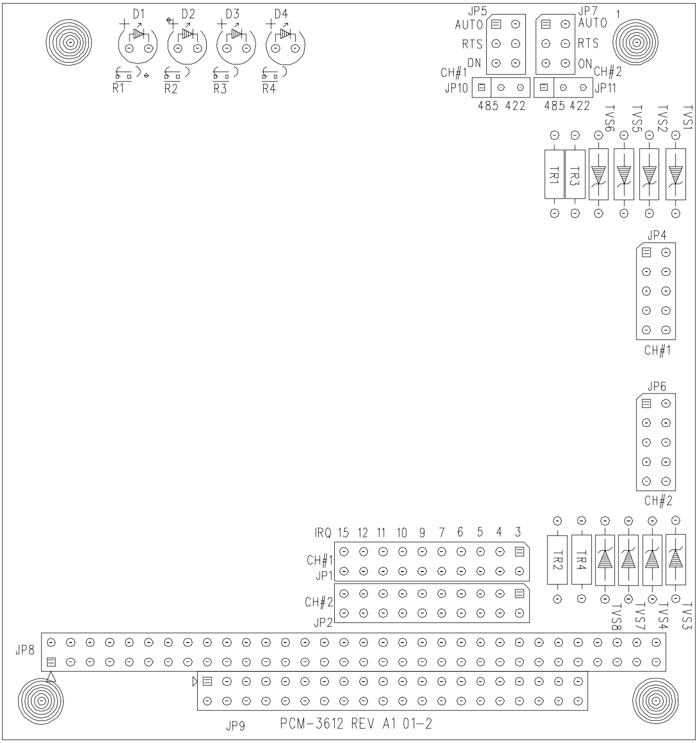


SW1: CH#1
Base address: 3E8



SW2: CH#2
Base address: 2E8

Switch locations appear in the figure below:



The following table shows the switch settings for various base addresses:

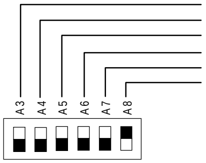
Port base address (SW1 or SW2)

base address	A3	A4	A5	A6	A7	A8
200-207	●	●	●	●	●	●
208-20F	○	●	●	●	●	●
—						
*2E8-2EF	○	●	○	○	○	●
—						
2F8-2EF (COM2)	○	○	○	○	○	●
—						
*3E8-3EF	○	●	○	○	○	○
—						
3F8-3FF (COM1)	○	○	○	○	○	○

○ = Off ● = On * = default

The following example shows how to set the base address to 2F8, the default setting for COM2: The switch sum is set to 2F8 = 200 + 80 + 40 + 20 + 10 + 8, (HEX).

Base address line	Decimal Value	Hex Value
A3	8	8
A4	16	10
A5	32	20
A6	64	40
A7	128	80
A8	256	100
A9	512	200

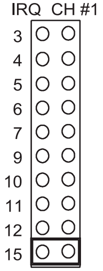


SW2: CH#2
Base address:2F8

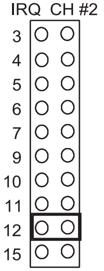
Interrupt level (IRQ) setting (JP1 and JP2)

Jumpers JP1 and JP2 set the interrupt for each port. You can choose any IRQ from 3 to 15 except for 8, 13, and 14. JP1 sets Port 1 and JP2 sets Port 2. If you want to use the card as COM1 and COM2, you must set Port 1 for IRQ4 and Port 2 for IRQ3. (Please disable COM1 and COM2 on-board under BIOS.) If you choose different IRQ's, make sure they are not used by other cards in the system. The following figures show the card default settings:

JP1: Port 1
IRQ15 (default)



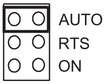
JP2: Port 2
IRQ12 (default)



Enable mode selection (JP5, JP7)

Jumpers JP5 and JP7 set the driver enable mode selections of ports 1 and 2 respectively. If jumpers JP5 or JP7 are set at "AUTO," the driver automatically senses the direction of the data flow and switches the direction of transmission. No handshaking is necessary.

If jumpers JP5 or JP7 are set to RTS (request to send), a high RTS signal can enable the driver. Otherwise, the driver output remains in high impedance status. If jumpers JP5 or JP7 are set to "On," the driver is always enabled, and always in high or low status. The user must select a mode before beginning RS-422 applications.



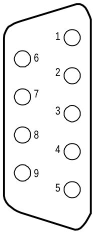
Note: *If your CPU module or card has serial interface ports, you will need to adjust the I/O port addresses (or disable the ports) to avoid conflicts.*

Connector pin assignments

You access the PCM-3612's ports through two external male DB-9 connctors. Ground pins are not connected to the DB-9 connector housing for the sake of isolation. With channel 1, you must attach the external cable to the proper connector.

RS-422/485 pin assignments appear below:

RS-422
RS-485



Pin description	
1	TX-(DATA-) or send data - (DTE)
2	TX+(DATA+) or send data + (DTE)
3	RX+ or receive data + (DTE)
4	RX - or receive data - (DTE)
5	GROUND
6	RTS - or ready to ssend -
7	RTS+ or ready to send +
8	CTS+ or clear to send +
9	CTS- or clear to send -

Hardware installation

Warning! *TURN OFF your PC power supply whenever you install or remove the PCM-3612 or connect and disconnect cables.*

Installing the module on a CPU card

1. Turn the PC's power off. Turn the power off to any peripheral devices, such as printers and monitors.

2. Disconnect the power cord and any other cables from the back of the computer.

3. Remove the system unit cover (see the user's guide for your chassis if necessary).

4. Remove the CPU card from the chassis (if necessary) to gain access to the card's PC/104 connector.

5. Screw the brass spacer (included with the module) into the threaded hole on the CPU card. Do not tighten too much, or the threads may be damaged.

6. Carefully align the pins of the PCM-3612 witht the PC/104 connector. Slide the module into the connector. The module pins may not slide all the way into the connector; do not push too hard, or the module may be damaged.

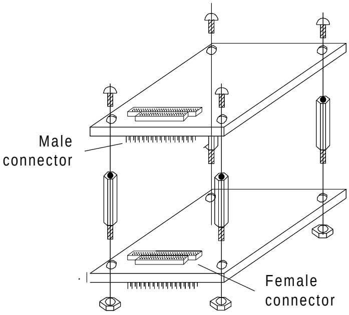
7. Secure the module to the CPU card to the threaded hole in the CPU card using the screw included.

8. Attach any accessories to the PCM-3612.

9. Reinstall the CPU card and replace the system unit cover. Reconnect the cables you removed in step 2. Turn the power on.

Connecting to another PC/104 module

1. Insert the pins fo connector JP8 on the end of the PCM-3612 module) into the piggyback connector on the other PC/104 module.



2. Screw the PCM-3612 to the brass spacer.

Signal wiring

The RS-422 has separate transmit and receive lines so that both devices can transmit at the same time. The transmit lines from one device connect to the receive lines on the other device. Typical connections are as follows:

Computer A		**Computer B**	
1	TX-	4	RX-
2	TX+	3	RX+
3	RX+	2	TX+
4	RX-	1	TX-
5	GND	5	GND
6	RTS-	9	CTS-
7	RTS+	8	CTS+
8	CTS+	7	RTS+
9	CTS-	6	RTS-

In the RS-485 the two devices share a single pair of data lines. One device transmits while the other receives.

Typical connections are as follows:

Device A		**Device B**	
1	TX-(Data-)	1	TX-(Data-)
2	TX+(Data+)	2	TX+(Data+)
5	GND	5	GND

Termination resistors

A user can solder in termination resistors for impedance matching if necessary. The card has two mounting spaces for termination resistors but no resistors are installed at the factory. Each pair of signal lines has a separate resistor.

	Port 1	**Port 2**
RS-422	TR1,TR3	TR2,TR4
RS-485	TR1	TR2

Resistors are usually installed at both ends of the communication lines. The value of each resistor should equal the characteristic impedance of the signal wires used (approximately 120 Ohms).

Programming

Programming with COM1 OR COM2

C language test program

You can use the following C program to test the PCM-3612's send and receive functions.

```
/********************************************/
/* Program:  DEMO01.C  (For RS485/RS422)    */
/* Description: This demo program transmits a*/
/* string  to COM1 and receives a string from */
/* COM2                                       */
/* Compiler: Turbo C 2.0                    */
/*                                            */
/* RS-485 jumper and switch settings, signal */
/* wiring                                     */
/* SW1  - 3F8       COM1:           COM2:     */
/* SW2  - 2F8       1   DATA- <——>  1   DATA- */
/* JP1  - IRQ4      2   DATA+ <——>  2   DATA+ */
/* JP2  - IRQ3      5   GND   <——>  5   GND   */
/* JP10 - 485                                 */
/* JP11 - 485                                 */
/* JP5  - 485/422                             */
/*                                            */
/* RS-422 jumper and switch settings, signal */ /
*  wiring                                     */
/* SW1  - 3F8       COM1:           COM2:     */
/* SW2  - 2F8       1   TX-   <——>  4   RX-   */
/* JP1  - IRQ4      2   TX+   <——>  3   RX+   */
/* JP2  - IRQ3      3   RX+   <——>  2   TX+   */
/* JP10 - 422       4   RX-   <——>  1   TX-   */
/* JP11 - 422       5   GND   <——>  5   GND   */
/* JP7  - 485/422                             */
/********************************************/
```

```
#   include <dos.h>
#   include <io.h>
#   include <stdio.h>
#   include <conio.h>

#   define  TIME_OUT  10000

static  int base0 = 0x3f8;  /* Base address of
        port 0  */
static  int base1 = 0x2f8;  /* Base address of
        port 1  */
static  char rec[16];  /* Buffer for received
   string  */
static  char cmd[16];  /* Buffer for transmit
   ted string  */
void main ()
{
   int i; /* Counter for character being sent/
        received  */
   char flag; /* Flag for end of output/input
   data  */
   int timeout;  /* Timeout counter  */
   outport((base0+2), 0xc9); /*enable port 0
   FIFO  */
   outport((base1+2), 0xc9); /*enable port 1
   FIFO  */

/* Set communciation parameters for port 0  */
   outp(base0+3, 0x80);  /* Set DLAB=1  */
   /* Set baud = 115200  */
   outp(base0, 0x01);
   /* Set data=8, stop=1, no parity */
   outp(base0+3, 0x03);
   /* Disable port 0 interrupt   */
   outp (base0+1, 0x00);

/* Set communciation parameters for port 1  */
   outp(base1+3, 0x80);  /* Set DLAB=1  */
   /* Set baud = 115200  */
   outp(base1, 0x01);
   /* Set data=8, stop=1, no parity */
   outp(base1+3, 0x03);
   /* Disable port 1 interrupt   */
   outp (base1+1, 0x00);

   printf("\nEnter a string to be transmitted"
   "(15 characters or less) or Q to quit:");
   gets(cmd);
   while (cmd[0] != 'q' && cmd[0] != 'Q')
   {
        i=0;
        cmd[strlen(cmd)] = 0x0d;
        flag=1;
        while (flag)
        {
           outportb(base0, cmd[i]);  /*Send
           data*/
           if (cmd[i] == 0x0d)
                 flag=0;
           i++;
        }

        i=0;
        flag=1;
        timeout=TIME_OUT;
        while (flag)
        {
           /* Check if receiver data is ready */
           if ((inportb(base1+5) & 1) !=0)
           {
                 rec[i]=inportb(base1); /*
                 Receive data */
                 if (rec [i] == 0x0d)
                 {
                     rec[i+1]='\0';
                     flag=0;
                     printf("\nReceived data:
                 %s\n",rec);
                 }
                 i++;
           }
           else
           {     /* Check timeout */
                 timeout--;
                 if (timeout == 0)
                 {
                     flag = 0;
                     printf("\nTimeout error\n");
                 }
           }
        }
   printf("\nEnter a string to be transmitted"
   "(15 characters or less) or Q to quit:");
   gets(cmd);
   }
}
```

Register structure and format

This section gives a short description of each of the module's registers. For more information, please refer to the data book for the STARTECT 16C550 UART chip.

All registers are one byte. Bit 0 is the least significant bit, and bit 7 is the most significant bit. The address of each register is specified as an offset from the port base address (BASE), selected with DIP switch SW1 or SW2.

DLAB is the "Divisor Latch Access Bit," bit 7 of BASE+3.

BASE+0 Receiver buffer register when DLAB=0 and the operation is a read.

BASE+0 Transmitter holding register when DLAB+0 and the operation is a write.

BASE+0 Divisor latch bits 0 - 7 when DLAB=1

BASE+1 Divisor latch bits 8 - 15 when DLAB=1.

The two bytes BASE+0 and BASE+1 together form a 16-bit number, the divisor, which determines the baud rate. Set the divisor as follows:

Baud rate	Divisor
50	2304
75	1536
150	768
300	384
600	192
1200	96
1800	64
2400	48
3600	32
4800	24
7200	16
9600	12
19200	6

38400	3
57600	2
115200	1

BASE+1 DLAB=0 Interrupt Status Register (ISR) when

bit 0 Enable received-data-available interrupt

bit 1 Enable transmitter-holding-register empty interrupt

bit 2 Enable receiver-line-status-interrupt

bit 3 Enable modem-status-interrupt

BASE+2 FIFO Control Register (FCR)

bit 0 Enable transmit and receive FIFO's

bit 1 Clear contents of receive FIFO

bit 2 Clear contents of transmit FIFO

bit 3 Change RXRDY and TXRDY from mode 0 to mode 1

bits 6-7 Set trigger level for receiver FIFO interrupt

Bit 7	Bit 6	FIFO trigger level
0	0	01
0	1	04
1	0	08
1	1	14

BASE+3 Line Control Register (LCR)

bit 0 Word length select bit 0

bit 1 Word length select bit 1

Bit 1	Bit 0	Word length (bits)
0	0	5
0	1	6
1	0	7
1	1	8

bit 2 Number of stop bits

bit 3 Parity enable

bit 4 Even parity select

bit 5 Stick parity

bit 6 Set break

bit 7 Divisor Latch Access Bit (DLA)

BASE+4 Modem Control Register (MCR)

bit 0 DTR

bit 1 RTS

BASE+5 Line Status Register (LSR)

bit 0 Receiver data ready

bit 1 Overrun error

bit 2 Parity error

bit 3 Framing error

bit 4 Break interrupt

bit 5 Transmitter holding register empty

bit 6 Transmitter shifting register empty

bit 7 At least one parity error, framing error, or break indicator in the FIFO.

BASE+6 Modem Status Register (MSR)

bit 0 Delta CTS

bit 1 Delta DSR

bit 2 Trailing edge ring indicator

bit 3 Delta received line signal detect

bit 4 CTS

bit 5 DSR

bit 6 RI

bit 7 Received line signal detect

BASE+7 Temporary data register

Standard PC I/O port assignments

The following chart shows the I/O addresses used by standard PC peripheral devices:

I/O address (hex)	Assignment
000-1FF	used by base system board
200	not used
201	game control
202-277	not used
278-27F	second printer port
280-2F7	not used
2F8-2FF	COM2
300-377	not used
378-37F	printer port
380-3AF	not used
3BO-3BF	monochrome adapter and printer
3CO-3CF	not used
3DO-3DF	color graphics adaptor
3EO-3EF	not used
3FO-3F7	floppy diskette drive
3F8-3FF	COM1: