Radiometrix

Hartcran House, 231 Kenton Lane, Harrow, HA3 8RP, England

Tel: +44 (0) 20 8909 9595, Fax: +44 (0) 20 8909 2233

Issue D, 13 July 2001 **RPC dev kit**

Development kit for the RPC transceiver

Introduction

RPC-DEV is a designers development kit for the RPC transceiver. It contains 2 complete sets of hardware for a point to point link, each end may be operated in one of 3 operating modes:

1. *Stand alone*
   Selects the RPC's internal diagnostic modes 0 to 8 Enables self-test, range testing and software debugging.

2. *PC driven link*
   RPC is controlled directly by a standard PC printer port (SPP with Open Collector / pull up control lines) or Extended Capabilities Port (ECP) set to Byte Mode or PS/2 Mode. PC software supplied with the kit enables display and editing all RPC internal registers.
   Data packets may be sent to and received from other RPCs. Software is supplied on 3.5" disk and runs under MS-DOS.

3. *PIC development*
   The RPC-DEV PCB will accept a wide range of 18 & 28 pin Arizona Microsystems PIC micro controllers for product development. RPC driver subroutine source code is included.

Kit contents

• 2 of RPC-418 or RPC-433	418MHz or 433.92MHz packet transceivers + Antennas.
• 2 of Development PCB's	Fully populated
• 2 of RPCPS2CAB cables	SPP to PS/2 Interface Cables
• 1 of 3.5" Floppy disc	MS-DOS PC Driver software for the RPC. RPC utility software including sample drivers.
• 1 of set of data	BiM & RPC data sheets Circuit / layout diagrams and documentation for the Development PCB's.

Required but not supplied:

1. A power supply of 7V to 12V DC @ <60mA (typ 20 to 40mA). (PSU or PP3)

2. An MS-DOS PC or laptop with a Standard Printer Port (SPP with Open Collector / pull-up) or Extended Capabilities Port (ECP) set to Byte Mode or PS/2 Mode).

3. PIC microcontrollers / PIC development tools.(only required for mode 3)

1. Standalone Operation

This mode selects the internal diagnostics built into the RPC.

Set-up

- Connect the antenna into the Antenna (R.H.) terminal on the RPC and plug the RPC into the development PCB.
- Connect a DC supply/9V battery to the supply input terminals and switch on.
- Select 'Debug' on the debug/normal switch and press 'RPC Reset'
- The Hex switch selects the required debug mode 0 to 8. An RPC reset is not required after a mode change. Debug must not be used when a PIC is inserted or a PC is connected to the Development PCB.

Test modes:

SELF TEST *(debug mode 8)*

8 on the hex switch selects the RPC's internal self test. The RPC puts the BiM transceiver into local loop back (both TX & RX on), a test code is continuously sent and recovered. The RXR LED on the development PCB will light to indicate a pass.

RADAR *(debug mode 7)*

7 on the hex switch will cause the RPC to repetitively send an ASCII test packet - "**Radiometrix RPC v 1.0 XX**", where XX is an incrementing packet counter 00 to 63. A remote unit in Radar mode is very useful for checking and debugging user RPC upload (RX) host software.

Additionally, Radar mode listens for a short time between each packet for an echo (see mode 6) and will light the RXR LED if a valid packet is received. This 'ping-pong' function is very effective for range / antenna tests.

ECHO *(debug mode 6)*

6 on the hex switch will cause the RPC to ECHO (retransmit) any valid packets it receives (transponder). Echo is very useful for remote loop-back testing of user host software and for "ping-pong" range testing in conjunction with the other development PCB in RADAR mode. The RXR LED flashes each time a packet is echoed.

Debug modes 0 - 5

These modes are primarily designed for radio performance testing with RF test gear and are described in the RPC data sheet.

RPC Dev Kit contains pairs of RPCPS2CAB (SPP to PS/2 Interface) cables with Black Assembled Cover on Male connector for PS/2 port or ECP port set to Byte Mode or PS/2 compatible mode.

2. PC or Laptop Operation

Set-up

- Connect the antenna into the Antenna (R.H.) terminal on the RPC and plug the RPC into the development PCB.

- Connect a DC supply/9V battery to the supply input terminals and switch on.

- Connect the development PCB to the LPT port of a PC or Laptop with the 25 way lead supplied. (Standard 25 way 1:1 'D' cable).

RPCPS2CAB cable has to be used for new PCs with ECP port

- Select 'Normal' on the debug/normal switch. Select 'F' (open circuit) on the hex switch

  Note: During PC operation, the hex debug switch must be on 'F' and no PIC microcontroller inserted (or it must be tri-stated).

- Copy all the supplied files into a suitable directory on your hard disk.

Software Overview:

The software supplied has been written to allow an RPC developer to gain immediate access to a Radiometrix RPC module and enables message send/receive and RPC memory examine/modify .

The 3.5” disk contains two versions of the software:

1. For Standard Parallel Port (SPP with Open Collector / pull-up control lines) in old PCs

RPC.DOC	RPC DEV KIT documentation.
RPC.EXE	main SPP RPC driver program (Run this file)
RPC.DAT	ASCII data file holds system information used by RPC.EXE
DATA-A.TXT	sample ASCII test files
DATA-B.TXT	
DATA-C.TXT	
DATA-D.TXT	
DATA-E.TXT	

2. For Bidirectional PS/2 or Extended Capabilities Port (ECP) in new PCs

Bidirectional version of the software is in the subdirectory \RPC_BI

It contains the following files:

DEMO.BAT	Batch file to set the Byte Mode or PS/2 Mode in ECP before running RPC software (Run this file)
README.DOC	RPC DEV KIT documentation.
RPC-BI.EXE	main PS/2 RPC driver program
RPC.DAT	ASCII data file holds system information used by RPC-BI.EXE
D.A	sample ASCII test files
D.B	
D.C	
D.D	
D.E	

Note: *SPP version of the RPC Dev Kit software (RPC.EXE) will not work with new PCs. Bidirectional PS/2 version of the RPC Dev Kit software (RPC-BI.EXE) need to be used in this case. Extended Capabilities Port (ECP) can be set to Byte Mode or PS/2 Mode. ECP_PS2.EXE program will change the mode to PS/2 compatible mode. PS2_ECP.EXE will change it back to ECP mode.*

The main program RPC.EXE is designed to run under MS-DOS 3.3 or higher on any IBM PC or compatible with a Standard Printer Port (SPP with Open Collector / pull-up control lines).
RPC-BI.EXE is designed for Extended Capabilities Port (ECP) set to Byte Mode or PS/2 Mode, set-up as LPT1.

Using this program it is possible gain access to the onboard EEPROM to evaluate the extended functionality provided via the Reserved Memory settings and to also read/write the User EEPROM area.

The program provides a set of commands allowing the user to operate the RPC module. These command functions enable the user to include send and receive messages, write to EEPROM memory, send continuous messages to the RPC and enable the 7 RPC reset modes.

Upon start-up of the utility if an RPC is connected to the parallel port and is working, the Reserved Memory and User Memory areas of the display will be updated.

The Outgoing and Incoming message area can be expanded or reduced as the user requires by pressing the TAB key (or alternatively entering switcH at the command prompt). This has the effect of either hiding or showing the user memory area. In order to view the help list properly the display needs to be in the expanded mode.

Messages sent from the RPC are displayed under the Outgoing (TX) message heading to the left of the display. Messages received by the RPC are displayed under the Incoming (RX) heading to the right of the display.

Command set:

The following list has been taken from the RPC demonstration program and details the commands which are available for evaluating the RPC.

COMMAND	*DESCRIPTION*
Reset [n]	Reset the RPC; Test mode (n = 0 - 8)
rea**D** address	Read from memory address (Addr = 00 - 3F)
Send [$] string	Transmit string via RPC; $ selects preamble
Write address data	Write data to RPC memory address (00 - 3F)
Clear	Clear the display output window
switc**H** or <TAB>	toggle memory display window On/Off
File [delay] file [file...]	Send a file(s) to the RPC
	delay = delay between files (0.25s increments)
Test	execute the file send test list
sto**P** or <^X>	to stop repeating file send (Test & File)
Help	Display this help information
e**X**it or <F3>	Exit from this demo program
<ESC>	Erase current command line
<TAB>	Switch display between memory and output

note: 1. The capital letter in each command may be used in place of the full word.
2. All values required by specific commands should be entered in hex.

Commands in Detail:

Note: Square brackets [] means the argument(s) are optional.
Angle brackets <> means the argument(s) are required.
The capital letter in the command represents an abbreviation of that command.

Reset [0-8] — send a RESET to the RPC.
sets the RPC into the specified demo mode.
- *example:* >R reset RPC
  >R7 reset RPC into debug mode 7 (RADAR)

rea**D** <address> — read the EEPROM memory at location <address>
the address should be in the range of 0x00 - 0x3F
- *e.g.:* >read 20

Send [$] <string> — Transmit the given string via the RPC.
Preceding the string with a $ sign will enable extended preamble to be used when transmitting the packet. If more than 27 bytes of data are entered on the command line, the string will be broken into 2 RPC packets and transmitted.
- *example:* >S THIS IS A TEST send 'THIS IS A TEST'
  >S$Is any body out there ? ; extended preamble

Write <addr> <data> — Write data to the specified RPC EEPROM location.
The allowable memory range is from 0x00 to 0x3F. Data represents a single byte between 0x00 - 0xFF.
- *example:* >W 00 4 writes 04h into SWITCHES (00h)
  >W 08 80 will set PS1 on reset

e**X**it or <F3> — Typing either EXIT or X, or pressing F3 will exit from the demonstration program back to the command prompt.

Clear — Clear the display output window.

sto**P** or <^X> — Stop the repeating file send (Test & File commands).

Help — Display this help information as shown in section RPCDEMO commands.

<TAB> — Switches the display between the memory display and the expanded output display.

<ESC> — Clear the current command line

File [delay] file [file...] Send a file to the RPC. A maximum of 3 files can be given on the command line. The file names should contain only alpha characters (e.g. fred1.txt <- not allowed. freda.txt <- allowed). The [dly] enables a delay of between 0x00 and 0xFF seconds between files being sent. Using the delay will enable the files to be repeated continuously using the delay value between them. Without the delay value the file(s) will only be sent once.
This is NOT a file transfer function. i.e.; it will not copy the file to the destination.
If any lines in the file contain more than 27 bytes, the line will be broken into multiple blocks of 27 bytes. They will NOT be reassembled into full lines by the receiver.

- *e.g.* file 2 autoexec.bat config.sys
- *example:* >F RPC.DOC transmits this file

Test Execute the test file command line from the RPC.DAT file.
See RPC Configuration File details later for a description of RPC.DAT command line used with this command.

- *example:* >T repetitively sends the test files.

RPC Configuration File:

The demonstration program requires a configuration file. This file is called RPC.DAT. Following is an example RPC.DAT file:

```
PORT = 378
COLOUR = 0
FILE = 2 rpc.dat c:\config.sys
```

PORT = <xxx>
This entry determines the base address of the PC printer port.
The value is entered as a hex value.

COLOUR = <0 | 1>
This is a Boolean of either 0 or 1.
Setting this entry to 0 disables the colour display. i.e.; all output will be in black and white.
This is suitable for a monochrome display device such as a laptop.
Setting this entry to 1 will enable the coloured output.

FILE = <delay> <file1> <file2> <file3>
This entry has the same format as the FILE command.
It is the command line used when the TEST command is entered.

Interrupt selection:

The RPC driver program uses interrupts therefor the two IRQ jumpers on the RPC Development PCB should be fitted. If a PC based RPC driver is being developed it is often desirable to use interrupt driven software. If the IRQ jumpers are fitted a +ve going interrupt (std) will be generated when either/or RXR or TXA from the RPC become active.

3. PIC Microcontroller Development

The PIC range of microcontrollers from Arizona Microsystems are well suited to act as host controllers for the RPC. The RPC Development kit will accommodate the following PIC's:

- low cost 18pin - PIC16C54, PIC16C56 & PIC16C58
- mid performance 18pin - PIC16C61, PIC16C62x, PIC16C71, PIC16F84, & PIC16F84A
- larger I/O 28pin - PIC16C62, PIC16C63, PIC16C73A & PIC16C76.

In all cases port B (RB0 - RB7) is dedicated to driving the RPC, sample driver subroutines are included on disk.

Port A and Port C (28pin PIC's) are brought out to a standard pin headers together with supply and reset lines to allow connection of user developed hardware. A 10.240MHz XTAL is fitted as standard but may be changed to suit user requirements.

Setup

- Connect the antenna into the Antenna (R.H.) terminal on the RPC and plug the RPC into the development PCB.
- Connect a DC supply/9V battery to the supply input terminals.
- Select 'Normal' on the debug/normal switch, select 'F' on the hex switch.
- Insert a PIC or ICE into either the 18 or 28 pin PIC socket + any custom hardware.

note: The 'F' on the hex switch is all off (open circuit) During PIC operation, the hex debug switch must be on 'F' and the PC must not be connected.

Appendix A *Using a printer port to drive the RPC.*

For Old PCs: Standard Parallel Port (SPP with open collector / pull-up control lines)

This is the default RPC Dev Kit configuration. It will work properly only on old PCs which have Standard Parallel Port (SPP) with open collector / pull-up control lines. RPC.EXE has to be used with this configuration.

Bidirectional PS/2 port configuration has to be used for new PCs which have PS/2 compatible ports or Extended Capabilities Port (ECP). DEMO.BAT (RPC-BI.EXE) in \RPC-BI subdirectory has to used with PS/2 mode configuration.

25 way Male D Connector on the RPC Dev Kit is connected to the RPC with the connections shown on the table below. The cable is standard one to one Female to Male 25 way cable (not included). It can only be used with the default configuration.

To use the Bidirectional PS/2 mode, RPCPS2CAB has to be used to change the connections as shown in Figure 1 on page 11.

Connections: RPC to 25 WAY 'D' printer port

RPC name	*End pin*		*pin*	*SPP bit*	*Port End pin labels*
GND	1	–	18 to 25		Ground
D0	2	↔	1	C0	$\overline{\text{Strobe}}$
D1	3	↔	14	C1	$\overline{\text{Auto Linefeed}}$
D2	4	↔	16	C2	$\overline{\text{Initialise Printer}}$
D3	5	↔	17	C3	$\overline{\text{Select Printer}}$
TXR	6	←	2	D0	Data 0
TXA	7	→	12	S5	Paper Out
RXR	8	→	13	S4	Printer Selected
RXA	9	←	3	D1	Data 1
RES	10	←	4	D2	Data 2
5V	11	←	+5V supply		
GND	12	←	0V supply		
+ve interrupt		→	10	S6	$\overline{\text{Acknowledge}}$

PC printer port registers (addresses given for base address of 0378hex)

0378	data register	b7	b6	b5	b4	b3	b2	b1	b0
		-	-	-	-	-	RES	RXA	TXR
0379	status register	b7	b6	b5	b4	b3	b2	b1	b0
			Int	TXA	RXR				
037A	control register	b7	b6	b5	b4	b3	b2	b1	b0
		-	-	dir	Ien	D3	D2	D1	D0

int +ve transition interrupt bit, see "interrupt drive"
Ien bit is internal interrupt enable, not used if polling used
1 = interrupt enable , 0 = disabled (polled operation)

note D0, D1 and D3 pins are inverted drives from the register i.e. a 1 in the control register gives a 0 on the pin all other registers / bits are true.

Cautions:

1. Many laptops use 3.3V logic on the printer port pins. This will work directly with a 5V RPC with the exception of the RPC's reset line which must be level shifted to 5V logic.(the RPC EVAL kit does this)

2. If you are writing PC Drivers beware of the long noise suppression time constants (upto 10µs) often used on some/all of the pins of the printer port. Beware of AND/OR instructions which act directly on the port registers as the data may not have settled from previous writes' to the port, it's safer to keep and use an indirect copy of the port registers in memory.

For New PCs: Bi-directional Port (PS/2)

Port requirement 8 bit bi-directional PS/2
(PS/2 or ECP set to PS/2 Mode / Byte Mode)

This configuration is needed for new PCs because the Standard Parallel Port (SPP) in new PCs does not contain open collector / pull-up control lines as in older PCs. SPP version of the software uses Printer Port Control lines as RPC Data lines. Data lines are pulled low by RPC when sending bit '0' to host. SPP in new PCs are hard driven logic outputs which cannot be pulled low by RPC.

Bidirectional version of the hardware & software overcomes this by using PS/2 Mode. In PS/2 Mode, Status Lines are used for Control line input from RPC (RXR, TXA) and Printer Port Control Lines are used to output the RPC Control signals (RXA & TXR). Default RPC Dev Kit connector is setup for SPP mode. RPCPS2CAB (SPP to PS/2 interface cable) provided has to be used for Bidirectional PS/2 mode.

In Bidirectional PS/2 mode, Printer port data lines can be used as RPC data lines in bidirectional mode.
Open collector / pull-up control lines are not needed for RPC data lines.

Most of the new PCs come with Extended Capabilities Port (ECP). ECP can be set to operate in PS/2 compatible bidirectional mode. Program supplied with bidirectional version will automatically change the mode from ECP to PS/2 and change it back to ECP when the RPC Dev Kit software is closed.

Connections: RPC to 25 WAY 'D' printer port

RPC name	***End pin***		***Bidirectional pin***	***PS/2 Printer bit***	***Port End pin labels***
GND	1	—	18 to 25		Ground
D0	2	↔	2	D0	Data 0
D1	3	↔	3	D1	Data 1
D2	4	↔	4	D2	Data 2
D3	5	↔	5	D3	Data 3
TXR	6	←	1	C0	Strobe
TXA	7	→	12	S5	Paper Out
RXR	8	→	13	S4	Printer Selected
RXA	9	←	14	C1	Auto Linefeed
RES	10	←	16	C2	Initialise Printer
5V	11	←	+5V supply		
GND	12	←	0V supply		
+ve interrupt		→	10	S6	Acknowledge

PC Printer port registers
(addresses given for base address of 0378h)

0378	data register	b7	b6	b5	b4	b3	b2	b1	b0
		-	-	-	-	D3	D2	D1	D0
0379	status register	b7	b6	b5	b4	b3	b2	b1	b0
			int	TXA	RXR				
037A	control register	b7	b6	b5	b4	b3	b2	b1	b0
		-	-	dir	Ien	-	RES	RXA	TXR

For Extended Capabilities Port (ECP) only

077A	Extended Control Register (ECR)	b7	b6	b5	b4	b3	b2	b1	b0
		0	0	1	-	-	-	-	-

Printer Port can be configured to operate in ECP mode by changing the Printer Port setting in BIOS from SPP or EPP to ECP. BIOS setup can be accessed by pressing DEL key for AWARD BIOS or F1 for AMI BIOS when booting the computer. Parallel Port settings can be changed in the Integrated Peripherals section of the BIOS setup.

However, it may be necessary to change it back to SPP or EPP mode for some printers to operate properly.

int +ve transition interrupt bit, see "interrupt drive"
Ien bit is internal interrupt enable, not used if polling used
1 = interrupt enable , 0 = disabled (polled operation)

note RXA and TXR pins are inverted drives from the register
ie a 1 in the control register gives a 0 on the pin
all other registers / bits are true.

RPCPS2CAB (SPP to PS/2 Interface Cable) is available with RPC Dev Kit.

RPCPS2CAB cable has the following cross connections.

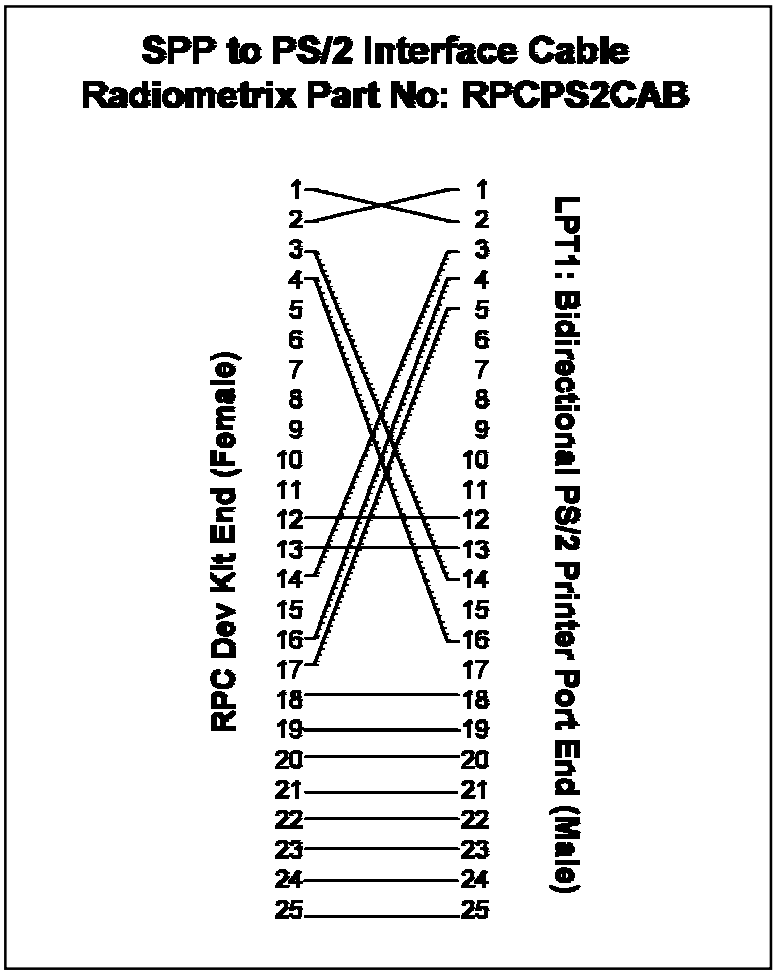

Figure 1: RPCPS2CAB (SPP to PS/2 Interface Cable) wiring diagram

Appendix B
SAMPLE RPC DRIVE SUBROUTINES FOR A PIC HOST

These subroutines may by used by a PIC Host in the RPC development PCB.
Packet transfers to / from the RPC are handled by two subroutines :-
OUT_BYTE & IN_BYTE
Additionally LISTEN_BUS is called on completion of a packet transfer
to the RPC to return the data bus to inputs (default state).

```
TITLE 'RPC DRIVERS FOR A PIC'
SUBTITLE 'VERSION1 GRAHAM SHARPLES'
;
            list  p=16F84     r=hex
;
; STANDARD EQUATES - dedicated data file locations - PAGE 0
;
INDF         EQU     0           ; INDIRECT CALL , OPCODE WILL USE FSR
                                 (4H) AS FILE POINTER
TMR          EQU     1           ; 8bit timer
PCL          EQU     2           ; low order program counter
STATUS       EQU     3           ; STATUS BITS
IRP          EQU     7           ; Not used on 16c84 , indirect addressing
                                 page bit
RP1          EQU     6           ; Not used on 16c84 , high order file
                                 addressing page bit
RP0          EQU     5           ; file page 0=PAGE 0 , 1=PAGE 1 e.g.
                                 DDR's etc.
TO           EQU     4           ; cleared on watchdog time-out
PD           EQU     3           ; cleared on sleep instruction
Z            EQU     2           ; ZERO flag
DC           EQU     1           ; DIGIT CARRY flag (4th bit)
C            EQU     0           ; CARRY flag
;
FSR          EQU     4           ; file pointer (indirect file address
                                 register)
;
PORTA        EQU     5           ; i/o port A - 5 Bits , Free for
                                 applications program use.
;
RPC          EQU     86          ; USE PORT B ON PIC

; Bit assignments for RPC PORT
D7           EQU     7           ; Bi-Dir data , D3
D6           EQU     6           ; Bi-Dir data , D2
D5           EQU     5           ; Bi-Dir data , D1
D4           EQU     4           ; Bi-Dir data , D0
TXA          EQU     3           ; INPUT  , active low TX accept from RPC
TXR          EQU     2           ; OUTPUT , active low TX request to RPC
RXA          EQU     1           ; OUTPUT , active low RX accept to RPC
RXR          EQU     0           ; INPUT  , active low RX request from
                                 RPC,(interrupt if required)
;
PCH          EQU     0A          ; Holding reg for program counter high
                                 byte
INTCON       EQU     0B          ; GIE,EEIE,TOIE,INTE,RBIE,TOIF,INTF,RBIF
                                 interrupts
;
```

```
; STANDARD EQUATES - dedicated data file locations - PAGE 1
;
OPTION       EQU     01          ; RBPU,INTEDG,RTS,RTE,PSA,PS2,PS1,PS0
DDRA         EQU     05          ; I/O direction reg ,portA 1= i/p  0= o/p
RPC_DDR      EQU     06          ; Data direction register for portB (RPC)
                                 ; This register is in BANK 1 of the
                                 register file
;
;-----------------------------------------------------------------------
             ORG     0
ONRESET      GOTO    START       ; jump to main program
;
; Initialise PORT B to drive RPC.
;
START        BSF     STATUS,RP0  ; select page 1
             MOVLW   B'11111001' ; TXR & RXA O/P , Rest as inputs
             MOVWF   RPC_DDR
             BCF     STATUS,RP0  ; select page 0
;

;  SUBROUTINE - IN_BYTE
;
;      IN_BYTE  - READ A BYTE FROM THE RPC INTO FILE POINTED TO BY FSR
;                 W IS DESTROYED
;
;          NOTE - THIS ROUTINE WILL HANG THE HOST UNTIL THE HOST
;                 COMPLETES THE TRANSFER OF TWO NIBBLES
;
;             - THIS SUBROUTINE CAN BE CONFIGURES TO RUN
;               AS PART OF AN INTERUPT HANDLER IF THE RXR
;               LINE FROM THE RPC IS USED TO TRIGGER A HOST INTERUPT
;
IN_BYTE      BTFSC   RPC,RXR     ; WE GOT A RX REQUEST YET ?
             GOTO    IN_BYTE     ; NO , SO LOOP BACK AND WAIT
;
;               READ THE LS NIBBLE FROM THE RPC
             BCF     RPC,RXA     ; ACCEPT THE REQUEST (SET ACCEPT LOW)
;
AWAITDATA    BTFSS   RPC,RXR     ; HAS REQUEST GONE UP ? i.e. data is
                                 present
             GOTO    AWAITDATA   ; LOOP BACK TILL IT DOES
;
             NOP                 ; TIME DELAY TO ENSURE DATA STABLE BEFOR
                                 READ
             MOVF    RPC,W       ; READ THE LS NIBBLE FROM THE BUS
             BSF     RPC,RXA     ; TELL RPC WE GOT NIBBLE (ACCEPT = 1)
             ANDLW   B'11110000' ; JUST THE DATA
             MOVWF   INDF        ; SAVE LS NIBBLE IN TARGET FILE (VIA FSR)
             SWAPF   INDF        ; RIGHT JUSTIFY LS NIBBLE
;
;               NOW GET MS NIBBLE FROM THE RPC
INNIBBLE     BTFSC   RPC,RXR     ; WE GOT NEXT RX REQUEST YET ?
             GOTO    INNIBBLE    ; NO , SO LOOP BACK AND WAIT
;
             BCF     RPC,RXA     ; ACCEPT REQUEST (SET ACCEPT LOW)
;
AWAITD1      BTFSS   RPC,RXR     ; HAS REQUEST GONE UP ? i.e. data is
                                 present
             GOTO    AWAITD1     ; LOOP BACK TILL IT DOES
;
             NOP                 ; TIME DELAY TO ENSURE DATA STABLE BEFORE
                                 READ
             MOVF    RPC,W       ; READ THE MS NIBBLE FROM THE BUS
             BSF     RPC,RXA     ; TELL RPC WE GOT NIBBLE (ACCEPT=1)
             ANDLW   B'11110000' ; JUST THE DATA
             IORWF   INDF        ; COMBINE MS NIBBLE WITH LS NIBBLE
```

```
                                ALREADY
;                               ; IN THE FILE (VIA FSR)
            RETURN
;       A BYTE HAS BEEN READ FROM THE RPC INTO ADDRESS POINTED AT BY FSR
;-----------------------------------------------------------------------


; SUBROUTINE - OUT_BYTE
;
;       OUT_BYTE  - WRITE A BYTE FROM FILE POINTED TO BY FSR TO RPC
;                   W IS DESTROYED
;
;       NOTE - THIS ROUTINE WILL HANG THE HOST UNTIL THE RPC
;              ACCEPTS THE TRANSFER OF TWO NIBBLES
;
;       WARNING - OUT_BYTE WILL SET THE DATA BUS TO DRIVE AFTER DETECTING
;                 A TXA FROM THE RPC.
;                 THE CALLING ROUTINE MUST SET 4 DATA LINES BACK TO I/P
;                 ON COMPLETION OF PACKET TRANSFER (i.e. call LISTENBUS)
;
OUT_BYTE     SWAPF    INDF,W        ; GET LS NIBBLE FROM FILE (VIA FSR) INTO
                                    ; BITS 4 to 7 of W
             ANDLW    B'11110000'   ; JUST THE NIBBLE
             IORLW    B'00000010'   ; SET TXR LOW, LEAVE RXA HIGH
             MOVWF    RPC           ; SET TXR LOW , OUTPUT NIBBLE
WACCEPT      BTFSC    RPC,TXA       ; WE GOT A TX ACCEPT BACK YET ?
             GOTO     WACCEPT       ; NO , SO LOOP BACK AND WAIT
;
;       WE GOT ACCEPTANCE SO IT'S OK TO DRIVE BUS
             BSF      STATUS,RP0    ; SELECT PAGE 1
             MOVLW    B'00001001'   ; DRIVE BUS
             MOVWF    RPC_DDR
             BCF      STATUS,RP0    ; SELECT PAGE 0 BUS IS NOW DRIVING
;
             BSF      RPC,TXR       ; REMOVE REQUEST, DATA IS ON BUS
WDUN         BTFSS    RPC,TXA       ; HAS DATA BEEN READ ?
             GOTO     WDUN          ; WAIT TILL RPC REMOVES ACCEPT
;
;        LS NIBBLE OF (FSR) IS SENT , NOW DO MS NIBBLE
             MOVF     INDF,W        ; GET MS NIBBLE FROM FILE (VIA FSR)
             ANDLW    B'11110000'   ; JUST THE MS NIBBLE
             IORLW    B'00000010'   ; SET TXR LOW (BIT 2), RXA STAYS HIGH
             MOVWF    RPC           ; OUTPUT NIBBLE + TXR LOW

WACCEPT1     BTFSC    RPC,TXA       ; WE GOT A TX ACCEPT BACK YET ?
             GOTO     WACCEPT1      ; NO , SO LOOP BACK AND WAIT
             BSF      RPC,TXR       ; REMOVE REQUEST, DATA IS ON BUS

WDUN1        BTFSS    RPC,TXA       ; HAS DATA BEEN READ ?
             GOTO     WDUN1         ; WAIT TILL RPC REMOVES ACCEPT
;
             RETURN                 ;    BYTE IS SENT TO RPC
;-----------------------------------------------------------------------
; SUBROUTINE - LISTEN_BUS , SET DATA BUS TO INPUT
;
LISTEN_BUS   BSF      STATUS,RP0    ; SELECT PAGE 1
             MOVLW    B'11111001'   ; BUS TO INPUT
             MOVWF    RPC_DDR
             BCF      STATUS,RP0    ; SELECT PAGE 0
             RETURN
;               BUS IS LISTENING TO RPC
;-----------------------------------------------------------------------
             END
```

Radiometrix Ltd

Hartcran House
231 Kenton Lane
Harrow, Middlesex
HA3 8RP
ENGLAND
Tel: +44 (0) 20 8909 9595
Fax: +44 (0) 20 8909 2233
sales@radiometrix.com
www.radiometrix.com

Copyright notice

Limitation of liability

The information furnished by Radiometrix Ltd is believed to be accurate and reliable. Radiometrix Ltd reserves the right to make changes or improvements in the design, specification or manufacture of its subassembly products without notice. Radiometrix Ltd does not assume any liability arising from the application or use of any product or circuit described herein, nor for any infringements of patents or other rights of third parties which may result from the use of its products. This data sheet neither states nor implies warranty of any kind, including fitness for any particular application. These radio devices may be subject to radio interference and may not function as intended if interference is present. We do NOT recommend their use for life critical applications.
The Intrastat commodity code for all our modules is: 8542 6000.

R&TTE Directive

After 7 April 2001 the manufacturer can only place finished product on the market under the provisions of the R&TTE Directive. Equipment within the scope of the R&TTE Directive may demonstrate compliance to the essential requirements specified in Article 3 of the Directive, as appropriate to the particular equipment.
Further details are available on The Office of Communications (Ofcom) web site:
http://www.ofcom.org.uk/licensing_numbering/radiocomms/licensing/licensing_policy_manual/

Information Requests
Ofcom
Riverside House
2a Southwark Bridge Road
London SE1 9HA
Tel: +44 (*0*)*845 456 3000 or 020 7981 3040*
Fax: +44 (0)*20 7783 4033*
information.requests@ofcom.org.uk

European Radiocommunications Office (ERO)
Peblingehus
Nansensgade 19
DK 1366 Copenhagen
Tel. +45 33896300
Fax +45 33896330
ero@ero.dk
www.ero.dk