Appendix—A

Table of Contents

PCSS–8FA and PCSS–8FX Operating System

Before describing the rich set of communications commands available to the programmer, a few notes are in order. Take a moment to read them.

1—In the following command descriptions, "CSC" means Currently Selected Channel.

2—Some of the commands have operands. Command operands are written to the data register. If a command returns information, then that information is read from the data register.

3—Communicating with the PCSS–8FA/FX is accomplished through two of the board's four contiguous I/O addresses in the computer's I/O map. The default base address is 2E0H. See Chapter 3 for other available base addresses.

3.a—The register at the boards' base address is known as the DATA REGISTER.

3.b—The register at the base address+1 is known as the COMMAND/STATUS Register. When written to, the base+1 address is the command register. On reads, the base address + 1 is a status register as follows:

Bit 7 Command Register Ready. When set (this is normally the case), the board is ready to accept a command. Note: ON PCSS–8FA only, status bits 0 through 6 must be qualified by bit 7. That is, they are only valid when bit 7 is high. On the PCSS–8FX, however, bit 7 need only be high prior to issuing a command. Data (and command operands if desired) may always be qualified by the status bits TxRdy, TxEmp, RxRdy. That is to say, on the PCSS-8FX, that while a command is in progress, not only will bit 7 be low, but TxRdy=0, RxRdy=0, Txemp=0 so that you would not transfer any data even if you were just monitoring these 8250 compatible status bits.

Bit 6 TxEmpty. Set when the transmit buffer is empty.

Bit 5 TXready. Set when the transmit buffer is ready, clear when the transmit buffer is full.

Bit 4 Break detected.

Bit 3 Framing error on the received byte.

Bit 2 Parity error on the received byte.

Bit 1 Overrun error on the received byte.

Bit 0 RXready. Set when there is received data. The data is already available on the data port at the board base address. When you read the data port, the status port is updated, and new data is applied to the data port if more is available.

3.d—On the PCSS–8FA/FX boards, the base address + 2 (2E2H) is reserved. Do not use it.

3.e—The base address + 3 is the RESET REGISTER. A write to this register of any data will cause a continuous hardware reset of the PCSS–8FA/FX. The reset condition is cleared by a write of any data to the DATA REGISTER. A hardware reset of the computer also resets the board, but a warm boot (ctl-alt-del) does not.

3.f—The following subroutines can be used to communicate with the PCSS–8FA/FX. You can assemble them into a linkable .OBJ file and use them with any high level language. They only use the AL register in your computer. They are used in the examples in the command descriptions.

```
;to issue command, call this routine with the command number in
AL.
xcom:   push    dx      ;execute command subroutine
        push    ax
        call    waitr   ;make sure card is ready, wait if nec.
```

```
        pop     ax      ;waitr sets dx=com base address+1
        out     dx,al   ;send command to card
        pop     dx
        ret
;to transfer data to the board, call this routine with the data in AL.
xdat:   push    dx      ;execute command to send
        push    ax      ;data to card
        call    waitr   ;wait until card is ready
        pop     ax
        dec     dx      ;point to data port
        out     dx,al   ;send data to card
        pop     dx
        ret
;to transfer data from board, call this routine. Data is returned in AL.
rdat:   push    dx
        call    waitr   ;make sure card is ready, wait if nec.
        dec     dx
        in      al,dx   ;get data
        pop     dx
        ret
;waitr returns status if board is ready or waits until it is ready.
waitr:  push    cx              ;wait until board is ready
        mov     dx,base_addr+1  ;point to cmd register
        mov     cx,0h           ;maximum wait count
waitr2: in      al,dx           ;get status
        test    al,80h          ;check ready bit (bit 7)
        jz      waitr1          ;uh oh, board not ready
        pop     cx
        ret                     ;return when board ready
waitr1: loop    waitr2          ;try again if cx not timed out
        ;If we get here, the board failed to become ready in a
        ;reasonable period of time. This means the board has
        ;failed, so output an error message (see following)
        mov     dx,offset ermsg
        mov     ah,9    ;display message function
        int     21h     ;display the message
        mov     ax,4C01h;exit with errorlevel
        int     21H     ;abort program all way back to DOS.

ermsg:  db      cr,lf,7,'Card Fault - Card not ready.',cr,lf,'$'
```

Following is an example:

```
;assume ah=channel, al=data
xmit:   xchg al,ah          ;get channel , save data in ah
        call    xcom        ;select channel
xmit0:  call    waitr       ;get status
        test    al,20h      ;check txrdy bit
        jz      xmit0       ;ok to xmit
;here if ok to xmit data
        xchg    al,ah       ;get data back to AL
        call    xdat        ;send the data
        ret
```

OK, enough said. Now for the command descriptions.

Commands 00h Thru 07h
Channel Select.

These commands select the channel (CSC) upon which the following commands operate (except global commands). usage:

```
        mov     al,chan_no      ;desired channel
        call    xcom            ;select the channel
```

Command 08h
Force PCSS-8I compatiblilty mode.
Only available on the PCSS-8F

Command 09h
(MCSS–9IM Only)
Select Channel 9

Reserved for the MCSS–9IM.

Command 0Ah
(MCSS–9IM Only)
Change Baud Reserved for the MCSS–9IM.

Reserved for the MCSS–9IM.

Commands 0Bh and 0Ch
Reserved

Command 0Dh
Get The Currently Selected Channel.

```
mov     al,0dh
call    xcom
call    rdat      ;returns csc in AL
```

Block Commands 0Eh

Command 0Eh, Data 00h

Turns off special mode to transmit the the same data to all eight channels.

Command 0Eh, Data 01h

Turns on special mode to transmit the same data to all eight channels.

Command 0Eh, Data 02h

Toggles between the standard and alternate baud rate sets.

Issuing command 0Eh followed by data 02h toggles the baud rate set between the standard set and the extended set. The standard set is the default set selected when the

card is initialized. This command affects all channels on the PCSS-8FA/FX. The baud rate selected by commands 18h and 1Ah are modified as shown following:

Standard Set	Extended Set
75	7200
110	880
150	14.4k
300	**28.8k**
600	57.6k
1200	**115.2k**
2000	2000
2400	57.6k
4800	4800
1800	**14.4k**
9600	9600
19.2k	19.2k
38.4k	38.4k
57.6k	57.6k

Command 0Eh, Data 03h

Enable DMA transfers from the PC to the PCSS–8FX. DMA NOTE: DMA transfers are only available on the PCSS-8FX versions 2.27 and above. DMA transfers must be enabled by selecting a DACK line with jumper JP2 and a DRQ line with jumper JP1.

Command 0Eh, Data 04h

Disable DMA transfers from the PC to the PCSS–8FX. See DMA note above.

Command 0Eh, Data 05h

Enable DMA transfers from the PCSS–8FX to the PC. See DMA note above.

Command 0Eh, Data 06h

Disable DMA transfers from the PCSS–8FX to the PC. See DMA note above.

Command 0Fh
Enable PCSS–8FA Mode
(PCSS–8F Only)

This command will make a PCSS–8F behave exactly like the PCSS–8FA described in this manual.

Command 10h
Start Break.

The transmit data line on CSC goes to +12 volts. The break is continuous until the stop break commmand is issued.

```
usage:  mov     al,10h   ;start break
        call    xcom     ;do it.
```

Command 11h
End Break.

The transmit data line on the CSC is restored to –12 volts.

usage:

```
        mov     al,11h   ;stop break
        call    xcom     ;do it.
```

Command 12h
Assert DTR (+12v)

Sets the DTR line on the currently selected channel to allow external device to send. The default is DTR inactive (–12v). If you are using auto DTR handshaking, do not use this command.

usage:

```
mov     al,12h    ;set dtr high
call    xcom      ;do it.
```

Command 13h
Negate DTR (–12V—Inactive)

Resets DTR to tell external device to stop sending. This happens in real time and is not inserted in the transmit queue. The default is DTR inactive. If you are using auto DTR handsahaking, then you should not use this command. Updated channel status for the CSC is placed on the data port.

usage:

```
mov     al,13h    ;set dtr low
call    xcom      ;do it.
```

Command 14h
Get Global CD Conditions.

This command will return a bit pattern on the data port showing the condition of all 8 CD lines. Bit 7 is channel 7, bit 6 is channel 6, bit 0 is channel 0, etc. If the bit is set, this means that an external device is asserting that line to +12 volts. Bits reset (0) mean that the CD line is –12 volts. This information is "real time" showing the condition of the CD lines, not from the queue. See also command 16h. After reading the the global cd bit pattern on the port, updated channel status for the CSC is placed on the data port.

usage:

```
mov     al, 14h   ;command
call    xcom      ;give command
call    rdat      ;get info
```

Command 15h
Get Global CTS Conditions.

This command will return a bit pattern on the data port showing the condition of all 8 CTS lines. Bit 7 is channel 7, bit 6 is channel 6, bit 0 is channel 0, etc. If the bit is set, this means that an external device is asserting that line to +12 volts. Bits reset (0) mean that the CTS line is –12 volts. This information is "real time" showing the condition of the CTS lines, not from the queue. See also command 16h. After reading the the global CTS bit pattern on the port, updated channel status for the CSC is placed on the data port.

usage:

```
        mov     al, 15h   ;command
        call    xcom      ;give command.
        call    rdat      ;get info
```

Command 16h
Get Modem Status.

This command will get the handshaking status of the CSC. A byte of data is placed on the data port in the following format.

Modem Status

bit 7	CD status. If set then CD currently being asserted (+12)
bit 6	reserved
bit 5	Remote TX status. If set, card has inhibited the remote transmitter using automatic handshaking, i.e., Xoff sent or DTR negated.
bit 4	CTS status. If set, CTS is currently asserted (+12)
bit 3	Delta CD. Set if CD changed since the last time read.
bit 2	reserved
bit 1	Received Xoff. If set, the card has received an Xoff, cleared when the card receives an Xon (OK to transmit). Valid only if Xon/Xoff TX flow control is on.
bit 0	Delta CTS. CTS changed since last time read.

Bit five set indicates that the card has attempted to inhibit the equipment that is transmitting to the card by the currently selected automatic handshaking method. Bit one set indicates that the card has received an Xoff if that method of receive handshaking has been enabled. After reading the the MODEM STATUS bit pattern on the port, updated channel status for the CSC is placed on the data port.

usage:

```
        mov     al, 16h   ;command
        call    xcom      ;give command.
        call    rdat      ;get info
```

Command 17h
Return Version Number.

This command will return the version of the operating system running in the card. Write the command number (17h) to the command port. Successive reads of the data port will return the version number. The first byte of data returned will be the number of bytes (n) in the version number. The next n bytes returned will be the version number. The last byte returned is the checksum of all the previous bytes.

```
;the following routine returns the version in ax. Returns ax=0 if
;version no. is bad
getvers:  push    cx
          push    bx
          mov     al, 17h
          call    xcom      ;give command
          call    rdat      ;get data byte count (currently 2)
          mov     bl, al    ;generate check sum
          call    rdat      ;get data
          mov     cl, al
          add     bl, al    ;generate check sum
          call    rdat      ;get data
          mov     ch, al
          add     bl, al    ;generate check sum
          call    rdat      ;get check sum
          add     bl, al    ;finish check sum
          cmp     bl, 0
          jz      ident1    ;check sum is ok
          mov     cx, 0     ;invalid information in cx
ident1:   mov     ax, cx    ;return version in ax
          pop     bx
          pop     cx
          ret
```

The current version reads like this: ax=126Fh, which means Operating System Version 1.26F.

Command 18h
Initialize Port (IBM Style)

Baud Rate, parity, stop bit and word size initialization for the CSC. This command also has the effect of flushing the currently selected channel queue. The data byte written to the data port following the command is in the following format:

```
7       6       5       4       3       2               1       0
—baud rate—             -parity-        stop            word length
000- 110                x0-none         0-1 stop        10-7 bits
001-19200*              01-odd          1-2 stop        11-8 bits
010-300                 11-even
011-600
100-1200        (*)150 bps in IBM bios. (Who needs that?)
101-2400        you can use the dos MODE command to
110-4800        set 19.2k by using 150 in the command
111-9600        MODE COM1:150 will yield 19.2K baud
```

See Command 0Eh Data 02h to select between the standard and extended baud rate set. Extended set follows:

```
7       6       5       4       3       2               1       0
---baud rate---         -parity-        stop            word length
000-    880             x0-none         0-1 stop        10-7 bits
001- 19,200             01-odd          1-2 stop        11-8 bits
010- 28,800             11-even
011- 57,600
100-115,200
101- 57,600
110-  4,800
111-  9,600
```

As an example, to set 1200 baud, even parity, 7 data bits with 1 stop bit, the data byte would be 10011010B or 9AH. Here is an example:

```
mov     al, 18h   ;the command
call    xcom
mov     al, 9Ah   ;the operand
```

call xdat

DTR is not affected by this command.

Command 19h
Handshake Init of the CSC.

You can use either Xon/Xoff handshaking or DTR/CTS handshaking. When this command is performed, it also has the effect of flushing the currently selected channel queue.

Auto Xon/Xoff handshaking controls the transmitting and receiving of characters to and from the PCSS–8FA/FX with Xon and Xoff characters.

If Auto Xon/Xoff Transmit Handshaking is enabled, an Xoff character received from an external device will automatically cause the PCSS–8FA/FX to stop sending any characters from its buffer until an Xon is received.

If Auto Xon/Xoff Receive Handshaking is enabled then when the "high water mark" is reached, an Xoff is automatically sent. This condition remains until characters are taken out of the receive queue for that channel and the "low water mark" is reached. A Xon is then sent.

Auto DTR/CTS handshaking controls the transmitting and receiving of characters to and from the PCSS–8FA/FX using the DTR and CTS hardware lines.

If Auto CTS Tx Handshaking is enabled and the CTS line goes low (–12), then no more characters will be transmitted from the PCSS–8FA/FX until it goes high again (+12).

If Auto DTR Rx Handshaking is enabled, then DTR is asserted automatically. Then when the "high water mark" is reached in the PCSS–8FA/FX receive buffer, DTR is set low (–12) until the receive buffer "low water mark" is reached by taking characters out of the receive queue. DTR is then set high (+12). If you select this method of handshaking, DO NOT use commands 12h or 13h since DTR is being controlled automatically.

The byte written to data register after the write to the command register is as follows (by bit):

7—reserved

6—reserved

5—Special Multi–Drop mode for GTEK's RS–485 adapter, on operating system versions 1.18d and later.

4—Enable Auto CTS Tx Handshake. Stops PCSS–8FA/FX from transmitting when CTS goes low (–12).

3—Enable Auto Xon/Xoff Tx Handshake. Stops PCSS–8FA/FX from transmitting when Xoff is received.

2—Enable Auto DTR Rx Handshake. Stops external device from transmitting when "high water mark" in the receive buffer is reached by setting DTR low (–12).

1—Enable Auto Xon/Xoff Rx Handshake. Stops external device from transmitting by sending Xoff when the "high water mark" in the receive buffer is reached.

0—reserved

usage:

```
mov     al, 19h    ;the command
call    xcom
mov     al, 14h    ;cts/dtr both directions
call    xdat       ;give data—DTR is now asserted.
```

NOTE: You **CANNOT** use both auto DTR and auto Xon/Xoff Rx handshaking at the same time. You **SHOULD NOT** use auto CTS and auto Xon/Xoff Tx handshaking at the same time.

Command 1Ah
Extended Baud Rates.

The parameter byte written to the data register following the command allows a greater degree of flexibility in setting baud rates than does command 18h. It is composed of two baud rate specification nibbles from the following table. Bits 7-4 control the baud rate of the receiver while bits 3-0 control that of the transmitter. See Command 0Eh to select the Normal or Extended Baud Rate Set.

Nibble data	Baud Rate Standard	Baud Rate Extended
0	75	7,200
1	110	880
3	150	14,400
4	300	28,800
5	600	57,600
6	1,200	115,200
7	2,000	2,000
8	2,400	57,600
9	4,800	4,800
A	1,800	14,400
B	9,600	9,600
C	19,200	19,200
2	38,400	38,400
D	57,600	57,600
E	reserved	reserved
F	reserved	reserved

For example, to transmit at 300 baud and receive at 1200 baud on the same channel, write 64h as the data to base address + 0 after the write of 1Ah to base address +1.

usage:

```
mov     al,1Ah    ;command
call    xcom
mov     al,64h
call    xdat      ;give the parameter
```

Command 1Bh
Set Handshaking Low Water Mark

This command and command 1Ch are used to adjust the levels for the automatic flow control on the receive queue. To set the receive low water mark on the csc, issue command 1Bh followed by the LSB and then the MSB of the new low water mark. There is a minimum value of one. If some form of handshaking is enabled and if high water mark is reached (upon which DTR goes low or XOFF is transmitted), then the re-enabling of the handshake signal (raising DTR or sending XON) will occur when enough characters are removed from the receive queue to reach the low water mark again.

IF the sum of all the transmit and receive limits exceeds the available system memory then there is the possibility that when a queue attempts to allocate a memory block, there will be none available. If this happens, the handshake flow control will take place (DTR low or XOFF sent) and the RESERVE block will be allocated to the channel. Each receive channel keeps a reserved block in case this condition occurs. If this happens, then the flow control will be re-enabled as soon as system memory becomes avail-

able. That is the DTR high or XON will occur as soon as the memory shortage clears rather than wait until the low water mark is reached.

```
;Set the receive low water mark
        mov     al,1Bh    ;command to set the low water mark
        call    xcom      ;issue the command
        mov     al,2h     ;low byte of the low water mark
        call    xdat      ;send the low byte
        mov     al,0      ;high byte of the low water mark
        call    xdat      ;send the high byte
;now the low water mark on the csc is set to two
```

Command 1Ch
Set The Receive High Water Mark

This command and command 1Bh are used to adjust the levels for the automatic flow control on the receive queue. To set the receive high water mark on the csc, issue command 1Ch followed by the LSB and then the MSB of the new high water mark. The high water mark is the place where DTR is lowered or an XOFF is sent if automatic handshaking is enabled with command 19h.

```
;Set the receive high water mark
        mov     al,1Ch    ;command to set the high water mark
        call    xcom      ;issue the command
        mov     al,10h    ;low byte of the high water mark
        call    xdat      ;send the low byte
        mov     al,0      ;high byte of the high water mark
        call    xdat      ;send the high byte
;now the high water mark on the csc is set to ten
```

Command 1Dh
Set Non–standard Baud Rate

Command 1Dh allows the user to select non–standard baud rates up to 57,600 on the csc. The command is issued followed by the LSB and MSB, in order, of the count to be calculated as follows:

COUNT = $^{115,200}/_{Desired\_Baud\_Rate}$

Note that the count must be greater than or equal to 2. Both bytes must be sent, even if one is zero. This command flushes both queues on the csc.

It is not possible to set different non–standard rates on pairs of channels such as 0–1, 2–3, etc. The second channel would have to use either the first one's rate or a standard baud rate, eg. channel 0 set to 57,600 would cause you to use either 57,600 for channel 1 or a standard baud rate such as 1200, 2400, etc. for channel 1.

usage:

```
;Set the baud rate to 1920 baud
;we calculated the count for the above data and formula
;to be 003Ch.
        mov     al,1Dh    ;command to set non standard baud
        call    xcom      ;issue the command
        mov     al, 3Ch   ;the low byte of the count
        call    xdat      ;send the low byte
        mov     al, 0     ;the high byte of the count
        call    xdat      ;send the high byte
;now the baud rate is 1920 on the csc, RX and TX
;the data for 2400 baud would be 0030h
```

Command 1Eh
Block commands

This command is issued followed by the function number on the data port as follows:

Command 1Eh, Data 00h
Return Free Memory Blocks

Issuing command 1Eh followed by a zero to the data port will cause the operating system to return the number of blocks of buffer memory that are currently not in use to the data port. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 01h
Return Buffer Memory Size

Issuing command 1Eh followed by a one to the data port will cause the operating system to return the number of blocks that can be used by the buffers in this system. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 02h
Return Reserved Code Size

Issuing command 1Eh followed by a two to the data port will cause the operating system to return the number of blocks that are reserved for code. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 03h
Return number of TX Blocks in Use

Issuing command 1Eh followed by a three to the data port will cause the operating system to return the number of blocks that are in use in the transmit queue on the currently selected channel. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 04h
Return number of RX Blocks in Use

Issuing command 1Eh followed by a four to the data port will cause the operating system to return the number of blocks that are in use in the receive queue on the currently selected channel. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 05h
Return TX Limit

Issuing command 1Eh followed by a five to the data port will cause the operating system to return the maximum number of blocks of memory that the transmit queue on the currently selected channel can use. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 06h
Return RX Limit

Issuing command 1Eh followed by a six to the data port will cause the operating system to return the maximum number of blocks of memory that the receive queue on the currently selected channel can use. If hand shaking is enabled, the remote transmitter will be inhibited from transmitting when the receive queue reaches this limit. The remote transmitter will also be inhibited when there is no

available memory even though the receive queue has not reached the limit. If this happens, the remote transmitter will be allowed to transmit as soon as memory becomes available. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 07h
Return RX Low Limit

Issuing command 1Eh followed by a seven to the data port will cause the operating system to return the "low water mark." This is the size in blocks that the receive queue has to go down to in order for the remote transmitter to continue transmitting if it had been inhibited from transmitting because the receive queue had reached the limit. If the low limit is set such that it is within one block of the receive limit, the remote transmitter will be inhibited/enabled as the last block is allocated/de–allocated to the receive queue. This will result in no spread between the high water and low water limits and could slow operation. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 08h
Return Total Number of Blocks in Use

Issuing command 1Eh followed by a eight to the data port will cause the operating system to return the total number of blocks of memory that are in use in all the queues. This number includes the eight blocks held in reserve by the receive queues. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 09h
Return System Memory Size

Issuing command 1Eh followed by a nine to the data port will cause the operating system to return the number of 32k "segments" of memory that are installed on the board. The low byte is returned first, followed by the high byte on successive reads.

Command 1Eh, Data 0Ah
Set TX Limit

This function will set the maximum number of blocks available for the transmit queue on the currently selected channel. This is accomplished by first issuing command 1Eh and then writing 0Ah to the data latch followed by the limit that you would like to use. Write the low byte first followed by the high byte even if the high byte is zero. The default transmit limit is 6 times the number of 32K memory blocks. Remember there is a minimum of one.

Command 1Eh, Data 0Bh
Set RX Limit

This function will set the maximum number of blocks available for the receive queue on the currently selected channel. This is accomplished by first issuing command 1Eh and then writing 0Bh to the data latch followed by the limit that you would like to use. Write the low byte first followed by the high byte even if the high byte is zero. The default receive limit is 10 times the number of 32K memory blocks. Remember there is a minimum of two because each channel holds one block in reserve to use if there are no free blocks available as the queue grows. When the queue uses this or its reserved block, the remote transmitter is XOFFed or DTR is negated if either type of hand-

shaking is enabled. As soon as memory becomes available it is allocated to the receive queue and the remote transmitter is allowed to begin sending again.

Command 1Eh, Data 0Ch
Set RX Low Water Mark

This function will set the "low water mark" in blocks for the receive queue on the currently selected channel. This is accomplished by first issuing command 1Eh and then writing 0Ch to the data latch followed by the lower limit that you would like to use. Write the low byte first followed by the high byte even if the high byte is zero. The default lower limit is 8 times the number of 32K memory blocks. Remember there is a minimum of one. If the lower limit is set to a number one lower than the RX limit or greater, the remote transmitter will be allowed to continue sending as soon as the queue drops below the high water mark.

Command 1Eh, Data 0Dh
Enable Irq line

This function will enable the IRQ line on the board. The IRQ number is selected by jumper as discussed in chapter 3. The selected IRQ is enabled by issuing command 1Eh and then writing 0Dh to the data latch followed by data 02h. If this command is not issued or if it is issued with a zero for the data byte, the board will not interrupt the host PC at all. Events that are enabled by command 30 will still be stored in the event queue and command 2Fh can be used to detect their occurrence.

Command 1Eh, Data 0Eh
Enable Timed interrupt

Command 1Eh sub 0Eh enables interrupt type 8, which is a timed interrupt. Write 1Eh to the command port followed byt 0Eh to the data port. Follow this with the low and then the high byte of the number of milliseconds that you wish to wait before an interrupt occurs. This timed intterupt must be enabled each time that you want the interrupt to occur.

usage:

```
;Enable an interrupt after 520 milliseconds.
        mov     al,1Eh    ;major command
        call    xcom      ;issue the command
        mov     al,0Eh    ;sub command for timed interrupt
        call    xdat      ;send the sub command number
        mov     al, 8     ;the low byte of the count
        call    xdat      ;send the low byte
        mov     al, 2     ;the high byte of the count
        call    xdat      ;send the high byte
;interrupt type 8 will occur in 208h or 520 milliseconds.
```

Command 1Fh
Poll Event ID Queue

Command 1Fh has been added to check for an interrupt ID in the interrupt ID queue. This command will return the oldest interrupt ID or 0FFh if no events have occurred. These interrupt events are enabled with command 30h. This command, unlike command 2Fh, does not remove the interrupt ID from the queue or cause the board to interrupt the PC if more interrupts are pending. This may be useful in your interrupt service routine in order to decide what to process without having to disable interrupts. You should

still do command 2Fh at the end of your interrupt service routine so that the board can interrupt the PC on the next event.

Command 20h
Flush Tx Queue On CSC.

A write of 20h to the command register immediately empties the currently selected channel's transmit queue. It has no effect on any other channel.

usage:

```
mov     al, 20h
call    xcom;that's all there is to it.
```

Command 21h
Flush Rx Queue On CSC.

A write of 21h to the command register immediately empties the currently selected channel's receive queue. It has no effect on any other channel.

usage:

```
mov     al, 21h
call    xcom    ;that's all there is to it.
```

Command 22h
Flush Tx/Rx Queues.

A write of 22h to base address + 1 immediately empties the currently selected channel's receive and transmit queue. It has no effect on any other channel.

usage:

```
mov     al, 22h
call    xcom    ;that's all there is to it.
```

Command 23h
Get Rx Queue Count

This command will give you the number of characters in the csc receive queue . The data returned to the data port (base + 0) is the low byte of the count followed by the high byte of the count.

usage:

```
mov     al, 23h   ;command
call    xcom
call    rdat
mov     ah, al    ;save low byte
call    rdat      ;get msb
xchg    al, ah    ;now have count in ax
```

The default value of 10 blocks on a 32K "segment" would allow for a maximum of 1680 characters.

Command 24h
Get Tx Queue Count

This command will give you the number of characters in the csc transmit queue that are waiting to be transmitted. The format is identical to command 23H.

Command 25h
Get Global Rxrdy Status.

This command allows you to check to see if any queue has received characters available. Bit 7 of the byte corresponds to channel 7, bit 6 is channel 6, etc.

usage:

```
mov     al, 25h
call    xcom
call    rdat      ;get bit pattern
```

If the data byte read is FFh, that would mean that there are characters in every channel. You could then determine how many in a particular channel by performing a channel select command (0–7h for channels 0–7) and then a command 23h.

Command 26h
Get Global Txrdy Status.

This command returns a bit pattern corresponding to the transmitter queue ready bits. A bit that is set indicates the transmit queue for that channel can accept more data (ie bit 7 set means that channel 7 transmit queue is not full).

usage:

```
mov     al, 26h
call    xcom
call    rdat      ;get bit pattern
```

Command 27h
Get Global Txemp Status.

This command returns a bit pattern corresponding to the transmitter queue empty bits. This actually is telling you there are no characters at all waiting to be sent in the transmit queue of each channel. Bit 7 is channel 7, bit 6 is channel 6, etc.

usage:

```
mov     al, 27h
call    xcom
call    rdat    ;get bit pattern
```

Command 28h
Pre–emptive Transmit.

This command allows you to put a character directly into the transmit holding register of the uart on the csc. More than likely, the character in the uart's transmit holding register will be over–written and thus that character will never be sent. Use this command with that in mind. It could even over–write an Xoff or Xon character in the tx holding buffer.

usage:

```
xchg    al, ah    ;save char in ah
mov     al, 28h
call    xcom
xchg    al, ah    ;get data
call    xdat      ;give pcss8fa/fx the data
```

Command 29h
Sample Most Recently Received Character.

This command allows you to sample the last character and status put into the receive queue. Status is returned first followed by the data. NOTE: This command reads the latest received character and does not remove the character from the queue. The command is thus non–destructive.

usage:

```
mov     al, 29h
call    xcom
call    rdat    ;get status
mov     ah, al  ;save it
call    rdat    ;now have stat in ah, data in al
```

Command 2Ah
Send Xon.

This command sends an Xon character on the csc. This command is used to send an xon character if you have auto xon/xoff rx handshaking in effect and you think that the remote device may have missed his xon. You can tell that the remote device is not supposed to be xoffed by examining bit 5 in the modem status.(see command 16h)

usage:

```
mov     al, 2ah
call    xcom    ;do it
```

Command 2Bh
Receive Xon.

This command allows you to simulate reception of Xon in the receive buffer. This will allow the resumption of the transmission of data if you have Auto Xon/Xoff Tx Handshaking enabled. It is acted on immediately, as if an Xon had been received. You can tell if you are in an Xoffed condition by examining bit 1 in the modem status (command 16h).

usage:

```
mov     al, 2bh
call    xcom    ;ok
```

Command 2Ch
Reserved

Command 2Dh
Reserved

Command 2Eh
Scan Receive Queue

This command is used to search the receive queue without removing any characters from the queue. Issue the command followed by the low and then the high bytes of the number **N** of the character that you wish to examine. Byte number 1 is the oldest byte in the queue. Be sure to use command 23h first to determine how many characters are in the receive queue. This command will return first the status and then the data that was in the queue at position **N**.

Examples

The following example is an assembly language program to initialize a channel on the PCSS-8FA/FX.

```
csc:    db      0       ;current com channel selected 0-7
IUART:  mov     al, csc ;comc= current com channel
        call    xcom    ;give command to select channel
        mov     al, 18h ;initialize baud/parity/stop
        call    xcom    ;send command
        mov     al, 23h ;19,200/8data/1stop/no parity
        call    xdat    ;send data to finish
        mov     al, 1Ah ;set new baud of 57,600
        call    xcom    ;send command
        mov     al, 0DDh ;parameter for 57,600 tx/rx
        call    xdat    ;send data to finish
        mov     al, 19h ;set handshaking command
        call    xcom    ;give command
        mov     al, 14h ;data to set hardware handshaking
        call    xdat    ;send data to finish
        ret     ;we are finished initializing the channel
```

Command 2Fh
Interrupt Acknowledge

This command should be issued after an interrupt has occurred and you have acknowledged the 8259 interrupt controller. It will return a data byte indicating what type of interrupt has occurred as outlined in the documentation of command 30h. If more interrupts are pending when this command is issued, the selected interrupt will be activated again. You should disable PC interrupts while in your interrupt service routine to keep from having nested interrupts.

Command 30h
Enable Interrupt on Event

Command 30h is used to enable the PCSS–8FA/FX to interrupt the Host PC on one of four events. You can select the type of event for the PCSS–8FA/FX to interrupt the PC with the ID that you write to the data port after the command.

Event	ID for channel 0—7
Transmitter Empty	00h—07h
Received data available	10h—17h
Receive buffer at limit	20h—27h
Transmitter Ready	30h—37h
Receive character match	40h—47h
Receive queue at level	50h—57h

For example if you wish to have the PC interrupted when there is received data on channel 3, the data byte to send would be 13h. Each of these interrupts is a one time event marker. If you wish to have the interrupt again on a similar event, you must enable it again. Command 2Fh is the interrupt acknowledge and it will return to you the identification number of the interrupt that has occurred in the same format as outlined here. The "Receive buffer at limit" interrupt depends on some type of receive handshaking being enabled with command 19h first. If you do not have receive hand shaking enabled, this event will never occur. The "Transmitter ready" interrupt can be used to mark the change from transmit buffer full (not ready) to ready. If you enable it before the buffer is full, the condition is true! Enabling any of the interrupts while the condition is true will cause the interrupt to occur right away. When this interrupt occurs you are assured of being able to transmit 252 more characters.

Multiple types of events on multiple channels can be enabled at any time. As the events occur the associated IDs are stored in a first in first out buffer so that they can be serviced in the order that they occur. If a condition is true when that event is enabled with this command the ID is immediately put into the event buffer, except type 40–47h.

Event Types 40h–47h occur when a received character matches the compare character stored with command 3ch. Event types 40h–47h , unlike the other event types, do not have to be enabled after each time the matched character is received. This command toggles the enabling of this type of interrupt. You simply enable this type of interrupt during the initialization of the board and each time a match receive character comes in, the interrupt ID will be stored in the interrupt ID queue. If you ever want to disable this interrupt type after it is enabled, simple issue command 30h followed by data 40-47h again to turn it back off.

Event Types 50–57 signal the receive count has reached the count supplied for the appropriate channel by command 3ch. The PCSS–8F/FX will continue to buffer receive characters even after the count is reached.

Command 3Bh
Return the Number of Pending Interrupts

Command 3Bh will return a data byte on the data port indicating how many events have been detected and are awaiting service. The count that is returned is the sum of events that have been enabled with command 30h that have occurred and have not been acknowledged with command 2Fh.

Command 3Ch
Store Compare Character

Command 3Ch is used to store the compare character or the receive level count on the currently selected channnel. Issue 3Ch to the command port at the base address plus one, followed by the byte you wish to look for or the number of received characters to the data port at the base address.

Command 3Dh
Clear Parity Bit On The Csc.

Command 3Dh forces the transmitter on the csc to transmit with the parity bit cleared. This command is sometimes used with RS–485 communication to indicate data bytes.

Command 3Eh
Set Parity Bit On The CSC.

Command 3Eh forces the transmitter on the csc to transmit with the parity bit set. This command is sometimes used with RS–485 communications to indicate address bytes for slaves.

Technical information and specifications provided in this document are ***SUBJECT TO CHANGE WITHOUT NOTICE.***

—Notes—