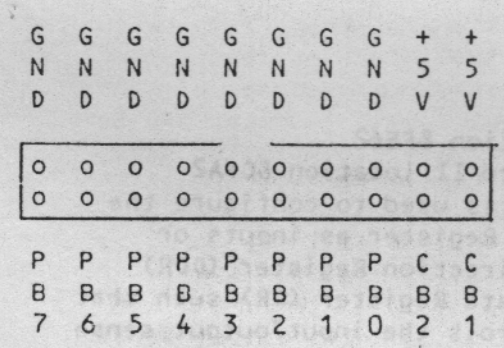
PRELIMINARY/REV4
++++++++++++++++

The Nightingale Accessory Board

The Nightingale Accessory Board gives Nightingale a number of advanced features; Software Mode-Selection,
Auto Dial,
Auto Answer,
Audible Call-Progress Monitoring.
Fitting the Accessory Board is covered in a separate manual.

Controlling the Accessory Board

The Accessory Board is controlled through an 8 bit parallel interface, which must also have two 'control' lines
[illegible] This [illegible] by
a 'user port'.
This manual assumes the use of a BBC computer, in which the user port is provided by one half of a 6522 Versatile Interface Adapter (VIA). This device is in common usage and so the details given could be the basis of software for a variety of computers. The Pace Mastercard II (for Apple II computers) provides a 6522 user port.
NOTE: The pin-out diagram given on page 426 of the BBC 'Advanced User Guide' (edition 3) is INCORRECT. Other editions may or may not be correct !
The correct pin-out is as shown below:



If using the Pace Mastercard II it is necessary to select the card workspace using two BIT instructions as per page 10 of the Mastercard II manual:

```
BIT $CFFF     ;switch off peripheral ROM space
BIT $CnOO     ;switch on Mastercard II
              ;where n = slot number.
```

The 6522 VIA

The 6522 provides two 8 bit bi-directional ports and 4 control lines, although in most applications, e.g. the BBC computer only one port and two control lines are available to the user. Both the BBC computer and Mastercard II provide the upper (B) half of the VIA together with the B-half control lines. The 8 data lines are referred to as PB0-PB7 and the control lines as CB1 and CB2.
The registers of the 6522 are memory mapped so that control of its various functions is achieved by simply writing data into specific addresses. These addresses will obviously vary between different types of computer but should be listed in your computer documentation.

Data Register- BBC location &FE60
Mastercard II location &CFA0

The Data Register (DR) is used to send/receive data to the Accessory Board. The 8 bits of the register correspond to the 8 port lines, so, if the port was configured to output, and a bit within the register (say bit 3) was set, then the corresponding output line (in this case PB3) would be high.

eg. Bit 3 = 1 gives PB3 = 5 Volts
Bit 3 = 0 gives PB3 = 0 Volts

If the port was configured as an input then the converse would apply,

eg. PB3 = 5 Volts gives Bit 3 = 1
PB3 = 0 Volts gives Bit 3 = 0

Data Direction Register- BBC location &FE62
Mastercard II location &CFA2

The Data Direction Register (DDR) is used to configure the individual bits/lines of the Data Register as inputs or outputs. The 8 bits of the Data Direction Register (DDR) correspond to the 8 bits of the Data Register (DR) such that a bit in the DDR (say Bit 3) controls the input/output sense of the corresponding bit/line (in this case PB3) of the DR. If a bit in the DDR is set then the corresponding bit/line of the DR will be an output, conversely if a bit in the DDR is cleared then the corresponding bit/line in the DR will be an input.

Control Register- BBC location &FE6C
Mastercard II location &CFAC

The Control Register (CR) can be used in an almost bewildering variety of ways, so only those areas of specific use will be discussed.

CB1 as an Interrupt Input.

The Accessory Board makes use of CB1 as an interrupt request (IRQ) line into the computer to signal on/off line status. Since CB1 is edge sensitive rather than level sensitive it is necessary to be able to tell the VIA whether it should trigger an interrupt request on the positive edge or on the negative edge of a CB1 transition. This is done using bit 4 of the Control Register. If bit 4 is set then the VIA will respond to a positive going edge, if bit 4 is cleared then the VIA will respond to a negative going edge.

CB2 as a controlled output.

The Accessory Board makes use of CB2 as a standard output line/bit that is achieved through the use of bits 7,6,5 of the control register.

viz; Bit 7	Bit 6	Bit 5	CB2
1	1	0	Low
1	1	1	High

Interrupt Enable Register -BBC Location &FE6E
Mastercard II location &CFAE

The Interrupt Enable Register (IER) is used in conjunction with the Control Register and is used to ensure that only the desired peripheral is capable of interrupting the processor. Although conceptually fairly tortuous it is reasonably easy to use in this application. Each of bits 0-6 is used to enable or disable one of the various interrupt sources. The interrupt capability of CB1 is controlled with bit 4 of the IER. eg. If IER bit 4 = 1 CB1 interrupts are enabled, if IER bit 4 = 0 CB1 interrupts are disabled. However, the setting/clearing of the bits in the IER cannot be accomplished by straightforward writing to the register. Bit 7 of the IER is used to specify the way in which writing to bits 0-6 is carried out. Clearing a bit in the IER is accomplished by writing an appropriate word to the IER with bit 7 cleared. In this case each set bit in the written word will clear the corresponding bit in the IER. Cleared bits in the written word have no effect on the corresponding bits in the IER. Setting a bit in the IER is accomplished by writing an appropriate word to the IER with bit 7 set. In this case each bit set in the written word will set the corresponding bit in the IER. Again, cleared bits in the written word have no effect on the corresponding bits in the IER. Bit 7 of the IER is write only, if read it always appears to be set. The other bits of the IER may be read normally.

Interrupt Flag Register -BBC Location &FE6D
Mastercard II location &CFAD

The Interrupt Flag Register (IFR) is used to communicate the source of an interrupt request to the 6502. When an interrupt request is accepted by the 6502 it reads the IFR's (or their equivalents) in each of its peripheral chips to firstly determine which chip caused the interrupt, and then to determine which of the potential interrupt sources within that chip is responsible for the request. Bit 7 of the IFR is used as a 'this chip caused the interrupt' flag, whilst bits 0-6 are used to indicate which interrupt source is requesting the interrupt. CB1 interrupts are associated with bit 4 of the IFR. The flag is cleared (ready for the next interrupt) either by a read/write of the Data Register or by a direct clear of the bit in the IFR.

Other Control Lines.

The Accessory Board's operation is also affected by the status of the Request To Send (RTS) line on the serial port. If RTS is low then the serial port is judged to be 'inactive' and the Accessory Board will ignore incoming telephone calls.

Signal Descriptions.
++++++++++++++++++++

CB1 - Carrier Detect, positive edge indicates carrier detect, negative edge indicates carrier lost.
Note there is limited hardware debounce on this line so further software debounce is essential to prevent noise giving spurious results.

CB2 - Hook Select, this line is analogous to the 'hook switch' on a telephone. The telephone line is 'grabbed' ready for dialling when this line is taken low.

PB0-7 have two modes of operation, firstly when CB2 is high, eg. when not dialling, and secondly when CB2 is low, eg. when dialling.

Mode 1 : CB2 = 1 (not dialling).
PB0 - Latch Data. Provided PB7 is high then a positive going edge (transition from 0 to 1) on PB0 will latch the data on PB1-PB6. To pass data to the Accessory Board the data must first be set up on the port, and then be latched into the board.
PB1 - Switch Select. For the Nightingale front panel switches to be active both PB1 and PB2 must be high. If PB1 is low then the host computer has control over the modem and the front panel switches will be inactive.
PB2 - Channel Select. When in a V23 mode this line controls the primary (high-speed) channel direction. If this line is set high then transmission will be at high speed and reception will be at low speed. If the line is set low then transmission will be slow but reception will be at high speed. This line should then be low for Prestel reception.
PB3 - MC1. This line is used to select the modem transmission mode (see table 1).
PB4 - MC0. As above.
PB5 - MC3. As above
PB6 - Speaker Control. This line is active low. e.g. If the line is taken high then the loudspeaker will be turned off, if the line is taken low then the loudspeaker will be turned on.
PB7 - DTR. This line should be be held high in mode 1.

Mode 2 : CB2 = 0 (dialling).
PB0-PB6 - Dial Code inputs. These lines are used to pass dialling information to the Accessory Board.(see table 2).
PB7 - DTR Override. This line is effectively a 'reset' line for the modem. It is only available when CB2 = 0 and is used whilst dialling to eliminate the problem of spurious carrier detection. This line is active low e.g. to reset, take CB2 and PB7 low.

Modem Control : Mode 1

As mentioned in the signal descriptions above, in mode 1 data must be latched into the modem under control of PB0. This is done so that the modem will remember the required settings without the data remaining on the port. eg. after setting up the modem in software the port can be used for dialling.

Transmission modes.
One of the most useful features gained with the Accessory Board is the ability to set up the transmission/reception mode of the Nightingale. This is done using PB3-PB5 and PB7. PB3-PB5 are used (when CB1 is high) to control the 'Mode Select' pins of the AM7910 chip in Nightingale and this makes it possible to set up a range of transmission modes including some which are not provided by the front panel switches.

Table 1 : AM7910 Mode Selection.

Port Bit	PB2	PB3	PB4	PB5		
7910 pin		MC1	MC0	MC3		
					Tx	Rx
V21 Originate	1	0	0	0	300	300
V21 Answer	1	0	1	0	300	300
V23 mode 2	1	1	0	0	1200	75
V23 mode 2	0	1	0	0	75	1200
V23 mode 2 equalised	1	1	1	0	1200	75
V23 mode 2 equalised	0	1	1	0	75	1200
V23 mode 1	1	0	0	1	600	75
V23 mode 1	0	0	0	1	75	600

Note. The V23 back channel (75 Baud) is optional. This means that Nightingale is also compatible with 600/1200 half duplex systems. However, since in actual use 1200 Baud half duplex is rare, and 600 Baud virtualy non-existent, it is strongly recommended that any future systems should support duplex operation.

Auto Answer

The Auto Answer system implemented on the Nightingale Accessory Board allows the implementation of a simple message system without software support, or a sophisticated bulletin board (such as OBBS) with software support.

A number of conditions must be met for the Accessory Board to be capable of answering the telephone:

1. The modem connect button must be left IN.
2. There must be 5 volts present on the User port's power supply pin.
3. The RTS line on the serial port must be high.

NB. It is conventional to leave Nightingale in either 300 Baud Answer mode OR 1200/75 Baud Originate mode.

Practically conditions 2 and 3 can be met by leaving the host computer turned on in 'chat' mode. When an incoming call is detected Nightingale will grab the line and send out carrier. If the incoming call was from another modem, set to the correct transmission mode then the two modems will lock on to each others carrier and, being in chat mode, the caller will be able to leave a message on your screen.
This system is obviously rather limited in that it takes no account of the 'Carrier Detect' line (CB1). This means that the host software has no means of determining whether it is ON or OFF line. Listing 1(A/B), for the BBC, correctly interprets the signals on this line and is designed for bulletin board use. The accumulator is set on 'carrier detect' and BASIC error No 255 is set on 'carrier lost' This may be readily incorporated into a BASIC program. The *AUTO command should be left on disc.

Auto Dial : Mode 2

For most users the auto dial facility will be the most used of the boards features and for this reason the Commstar Auto Dial Emulation disc is available. This disc will prove ideal for most uses and custom software will only be necessary in applications requiring unattended operation or where the dialling software requires to be integrated with other functions.

The process of Auto Dialling is closely analogous to the process of manual dialling, but a terminology is needed:

1. Take the handset off the 'hook switch' (pick up the phone)
2. Dial a digit
3. Release the dial
4. Repeat until finished dialling
5. Listen for an answer
6. If there is an answer go into Commstar (etc) else put the receiver back on the 'hook switch' (put the phone down)

Stage 1 is accomplished by taking CB2 low. (see 'Control Register). It is a good idea to clear the port before taking CB2 down.

Stage 2 involves sending out a 7 bit code along PB0-PB6. The codes required for each digit are as below.

Table 2 - Digit Codes.

Digit to Dial	Value to send on PB0-PB6 Hex	Decimal
0	28	40
1	44	68
2	24	36
3	14	20
4	42	66
5	22	34
6	12	18
7	41	65
8	21	33
9	11	17
*	48	72
# (pause)	18	24

The code should be held for a few mS.

Stage 3 is accomplished by clearing PB0-PB6 and waiting a few centiseconds.

Stage 5 is the most complex stage since it is easy for the modem to see noise on the phone line as carrier. This can even happen when actually dialling unless the DTR override line is used as mentioned under PB7. The override should be enforced at least 5 seconds before the first digit is sent (to allow time for the modems internal circuitry to reset) and released soon after the final digit has been dialled. To do this it is necessary to calculate the time required by the Accessory Board to dial the digits. Each digit is sent as a series of pulses on the phone line. The pulse rate is 10 per second. For all digits other than 0 the number of pulses required corresponds to the value of the digit, eg. to dial a 5, the Accessory Board sends 5 pulses. The digit 0 requires 10 pulses. Between each digit there is an 'inter-digit pause' of 0.8 seconds.

Thus the time taken to dial a digit is:

(0.1*digit value) + 0.8 Seconds.

The Accessory Board carries a dedicated dialling chip which automatically carries out these timings.

Once the DTR override has been released there are two particular ways of 'listening' for a connection, to ensure success it is best to use a combination of the two techniques.
Firstly the carrier detect line can be used to check for carrier using the *AUTO command set up with listing 2.
Secondly it is possible to look for valid characters on the serial port. eg.

```
*FX 21,1      flush buffer
*FX 2,1       Set RS432 as input stream
TIME =0
REPEAT
A$=GET$
UNTIL A$<>"" OR TIME = 100
IF A$="" THEN drop line etc ELSE talk
```

To be sure of a connection, not only must carrier be detected but valid characters should be seen. The value of the wait time will typically be 5-10 seconds depending upon the age of your local exchange.
Having established a connection you may require a jump into Commstar. eg.

```
*FX 21,0      flush buffer
*FX 138,0,67 put a 'C' in buffer (Chat)
*FX 142,x     jump to rom in socket x
```

****

Simon Pearse, 24/9/85

Listing 1a

```
  5 REM *AUTO SOURCE FILE : Version 1.4 24/09/85
 10 OSBYTE=&FFF4:oscli=&FFF7
 20 DDRB=&FE62:REM Data Direction Register 'B'
 30 OVEC=&3A0:REM Location of stored interrupt vector
 40 orb=&FE60:REM Output/Input Register 'B'
 50 PCR=&FE6C:REM Peripheral Control Register
 60 ier=&FE6E:REM Interrupt Enable Register
 70 ifr=&FE6D:REM Interrupt Flag Register
 80 irq2=&206:REM Interrupt vector
 90 Destination=&C00:REM Place code at &C00
100 FOR I%=0TO2STEP2:P%=Destination
110   [ OPT I%
130    LDA #0:STA&404:STA&405:STA&406:STA&407 \ Clear A%
150    LDA #2:LDX #2:LDY #0:JSR OSBYTE \ Enable RS423
160    LDA #&FF:STA DDRB \ Make PB0-PB7 outputs
170    LDA ier
180    ORA #&90
190    STA ier \ to enable interrupt on CB1
200    LDA PCR
210    ORA #&10:STA PCR : \ Set bit 4 of PCR (ie. Look for a positive edge)
220   .Start SEI \ Disable interrupts
230    LDA irq2 \ Alter the interrupt vector.....
240    STA OVEC \ To piont to our routine
250    LDA irq2+1
260    STA OVEC+1
270    LDA #int1 MOD 256
280    STA irq2
290    LDA #int1 DIV 256
300    STA irq2+1
310    CLI /Re-enable interrupts
320    RTS \...and return to basic
330   .int1 LDA &FC \ Save register
340    PHP:PHA:TXA:PHA:TYA:PHA
350    LDA ifr
360    AND #&10 \ Check if the interrupt is the one we want......
370    BNE yes \ yes it is so branch
380    JMP exit \ no so leave routine
390   .yes LDA PCR:AND #&10 \ Check which edge we are looking for.....
400    BEQ clear \ negative edge
410    LDA orb \ Clear interrupt by reading data register 'B'....
420    LDA PCR:AND #&EF:STA PCR:LDA#1:STA&404:JMP exit \ Set A% ..
430   .clear LDA PCR:ORA #10:STA PCR \ We were looking for a positive edge so
now we look for a negative edge....
440    LDA orb \ Clear interrupt by reading data register 'B'
450    PLA:TAY:PLA:TAX:PLA:PLP \ Clear the stack
460    CLI \....Enable interrupts....
470    BRK:EQUB 255:EQUS "Line dropped":BRK \......and cause an error
480   .exit PLA:TAY:PLA:TAX:PLA:PLP \ Restore status quo
490    STA &FC
500    JMP (OVEC) \ Not our interrupt so let the system deal with it
510    ]
520   NEXT I%
530 CLS
535 IF P%=Destination+&9D THEN 540 ELSE PRINT"ERROR.. Please check your listi
ng"
537 END
540 $&A00="SAVE AUTO "+STR$~(Destination)+" "+STR$~(Destination+&A6)
550 X%=&A00 MOD 256:Y%=&A00 DIV 256:CALL oscli:REM Save object code as AUTO
560 CLS:PRINT"Assembled and saved OK.To run type *AUTO":END
```

Listing 1b

```
10 MODE7:ON ERROR GOTO 60
20 *AUTO
30 PRINT"Waiting.....";:REPEAT UNTIL A%<>0:PRINT "Answered":A%=0
40 REM Insert your program here
50 REPEAT UNTIL 0
60 IF ERR=255 THEN PRINT"Line dropped":?&FE6C=&10:A%=0:GOTO30
70 REPORT:PRINT"  at  ";ERL
>
```

Listing 2

```
  10 REM
  20 REM
  30 REM              DTRING4
  40 REM            SIMON PEARSE
  50 REM              13/9/85
  60 REM
  70 REM
  80 REM     **********************************
  90 REM     *                                *
 100 REM     * DTRING4 ALLOWS BAUD RATE       *
 110 REM     * SELECTION AS WELL AS DIALING.  *
 120 REM     * THE PROGRAM MAKES USE OF THE   *
 130 REM     * ACCESSORY BOARD'S 'DTR-        *
 140 REM     * OVERIDE' FACILITY.            *
 150 REM     * TO DISABLE THE MODEM TAKE PB7  *
 160 REM     * LOW AT THE SAME TIME AS CB2.   *
 170 REM     * I.E TAKE PB7 DOWN JUST BEFORE  *
 180 REM     * STARTING TO DIAL, RESTORE PB7  *
 190 REM     * WHEN IT IS CALCULATED THAT     *
 200 REM     * DIALING IS COMPLETED           *
 210 REM     *                                *
 220 REM     **********************************
 230 REM
 240 REM
 250 MODE0
 260 CLS
 270 auto=0
 280 CONREG=&FE6C
 290 DATREG=&FE60
 300 DIRREG=&FE62
 310 DROP=&E0
 320 GRAB=&C0
 330 ?CONREG=?CONREG AND &1F:?CONREG=?CONREG OR DROP
 340 PB7=&80
 350 DIM CODE(11),SETNG(8)
 360 ?DIRREG=&FF
 370 PROCtables
 380 ?CONREG=?CONREG AND &1F:?CONREG=?CONREG OR DROP:CLS
 390 INPUT "PLEASE ENTER THE NUMBER YOU WISH TO DIAL";NUMBER$
 400 INPUT"IS THIS A PRESTEL NUMBER";PRESTEL$
 410 INPUT"SPEAKER ON?";SPEAKER$
 420 PROCmenu
 430 CLS
 440 PROCsetmode
 450 PRINT"CHECK THAT THE 'MODEM CONNECT' BUTTON IS IN, THEN PRESS ANY KEY":G$=
GET$
 460 CLS
 470 PRINT"ENSURING CARRIER DROPPED"
 480 TIME=0
 490 IF auto=0 THEN auto=1 :*AUTO
 500 ?DATREG=0
 510 ?CONREG=?CONREG AND &1F:?CONREG=?CONREG OR GRAB
 520 REPEAT UNTIL TIME > 500
 530 PRINT"CARRIER DROPPED"
 540 PROCtimetodial
 550 dialstart=TIME
 560 PRINT"DIALING"
 570 FOR DIGIT =1 TO LEN(NUMBER$)
 580   DIGIT$=MID$(NUMBER$,DIGIT,1)
```

```
590 IF DIGIT$<>"*" AND DIGIT$<>"#" THEN KEY = VAL(DIGIT$)
600 IF DIGIT$="*" THEN KEY=10
610 IF DIGIT$="#" THEN KEY=11
620 opdata=CODE(KEY)
630 dialstart=TIME
640 PROCopbyte
650 NEXT DIGIT
660 REPEAT UNTIL TIME >(dialstart + dialtime)
670 ?DATREG=PB7   :REM DTR now released
680 TIME=0
690 REPEAT:PRINTTAB(10,10);TIME:UNTIL A%<>0 OR TIME >1500
700 IF INSTR("YESyes",SPEAKER$)=1 THEN ?DATREG=SETNG(mode)+PB7+64:?DATREG=?DAT
REG+1
710 ?CONREG=?CONREG AND &1F:?CONREG=?CONREG OR DROP
720 PRINT
730 IF TIME>1500 THEN PRINT"SORRY, NO CONNECTION":time=TIME:REPEAT:UNTILTIME=t
ime+200:GOTO380
740 REM flush keyboard buffer
750 *FX21,0
760 IF INSTR("YESyes",PRESTEL$)=0 THEN 860:REM if not Prestel selected then go
to 300 Baud entry
770 REM set 1200/75 baud
780 *FX7,4
790 *FX8,1
800 REM insert '#' & 'C' into keyboard buffer
810 *FX138,0,35
820 *FX138,0,67
830 REM call Commstar
840 *C*
850 REM set 300/300 baud
860 *FX7,3
870 *FX8,3
880 REM insert 'C' into keyboard buffer.
890 *FX138,0,67
900 REM call Commstar
910 *C*
920 END
930 DEFPROCtables
940 FOR KEY=0 TO 11
950   READopdata
960   CODE(KEY)=opdata
970   NEXT KEY
980 FOR mode =1 TO 8
990   READmodata
1000  SETNG(mode)=modata
1010  NEXT mode
1020 ENDPROC
1030 DEFPROCtimetodial
1040 dialtime=0
1050 FOR KEY=1 TO LEN(NUMBER$)
1060   DIGIT$=MID$(NUMBER$,KEY,1)
1070   IF DIGIT$<>"*" AND DIGIT$<>"#" AND DIGIT$<>"0" THEN dialtime=dialtime+10
*(VAL(DIGIT$))
1080   IF DIGIT$="0" THEN dialtime=dialtime+100
1090   IF DIGIT$="#" THEN dialtime=dialtime+320
1100   NEXT KEY
1110 dialtime=dialtime+(80*((LEN(NUMBER$))-1))
1120 dialtime=dialtime+100:REM Allow time taken to send data to board
1130 ENDPROC
1140 DEFPROCopbyte
>
```

```
1150 ?DATREG=0
1160 time=TIME
1170 REPEAT UNTIL TIME>time+5
1180 ?DATREG=opdata
1190 PRINTDIGIT$;
1200 time=TIME
1210 REPEAT UNTIL TIME>time+5
1220 ?DATREG=0
1230 ENDPROC
1240 DEFPROCmenu
1250 PRINT:PRINT
1260 PRINT"      MODE          Tx      Rx      Equaliser"
1270 PRINT
1280 PRINT" 1 => V21 ORIG      300     300     OFF "
1290 PRINT" 2 => V21 ANS       300     300     OFF "
1300 PRINT" 3 => V23.2         1200    75      OFF"
1310 PRINT" 4 => V23.2         75      1200    OFF"
1320 PRINT" 5 => V23.2         1200    75      ON"
1330 PRINT" 6 => V23.2         75      1200    ON"
1340 PRINT" 7 => V23.1         600     600     OFF"
1350 PRINT" 8 => V23.1         75      600     OFF"
1360 PRINT:PRINT
1370 PRINT"PLEASE SELECT COMMUNICATIONS MODE"
1380 REPEAT:G$=GET$:UNTILVAL(G$)>0 AND VAL(G$)<9
1390 mode=VAL(G$)
1400 ENDPROC
1410 DEFPROCsetmode
1420 *FX2,2
1430 ?DATREG=PB7+SETNG(mode)
1440 IF INSTR("YESyes",SPEAKER$)=0 THEN ?DATREG=?DATREG+64
1450 ?DATREG=?DATREG+1:REM Latch data
1460 ENDPROC
1470 DATA40,68,36,20,66,34,18,65,33,17,72,24:REM Dial digit codes
1480 DATA4,20,12,8,28,24,36,32:REM Communication modes
>
```