MBBIOS is an assembler language program that captures INT 14 (the COM service call) and extends it with support for the 8250 async port in buffered mode and adds support for the 8530 PACCOMM PC-100 packet board. It is configured by the file MBCNF.ASM. MBBIOS is a resident extension to DOS and only has to be invoked once.

PC-100 support is NOT normally included in the files.

## Changes

V 3.5 -- 16550A support, Desqview support, improved break.
Thanks to N2GTE for the latter two items. See notes further down for enabling Desqview support

V 3.6 -- High speed handshaking. Remove PC-100 support as default.

V 3.7 -- Fix bugs in handshake. Remove use of DTR in handshake as default.

V 3.8 -- Variable handshaking. Add control function.

## File contents

Files included are:

| | |
|---|---|
| MBBIOS  .COM | -- The resident program. Comes configured for COM1 as standard ASYNC |
| MBBIOSDT.COM | -- The resident program but for units that use DTR for handshaking. See note below under hardware handshaking. |
| MBBCONFG.COM | -- Configuration control program |
| MS400   .MOD | -- How you share interrupts on a DFI MS-400 card |

This ZIP file may contain the source code. If not, you can get it by sending $5 US (or equivalent) to:

Dave Larton, N6JQJ
353-C East 10th Street, Box 124
Gilroy, CA  95020-4149
(408) 778-7265

| | |
|---|---|
| 8250con .asm | -- 8250 hardware constants |
| 8250ini .asm | -- 8250 initialization |
| 8250int .asm | -- 8250 interrupt handler |
| 8250svc .asm | -- 8250 service call handler |
| 8250svcx.asm | -- 8250 service call handler (extended) |
| 8250tmr .asm | -- 8250 timer pop handler |
| qrqpcon .asm | -- Quadram QuadPort hardware constants |
| qrqpini .asm | -- Quadram QuadPort initialization |
| 8530con .asm | -- 8530 hardware constants |
| 8530ini .asm | -- 8530 initialization |
| 8530int .asm | -- 8530 interrupt handler |
| 8530svc .asm | -- 8530 service routines |
| 8530tmr .asm | -- 8530 timer pop handler |
| kiss    .asm | -- KISS constants |

```
asmmb   .bat   -- BAT routine to assemble MBBIOS

mbbios  .asm   -- Main line
mbbcom  .asm   -- Control block format
mbbinit .asm   -- Initialization routine
mbbinter.asm   -- Interrupt handler
mbbload .asm   -- Load MBBIOS
mbbsubr .asm   -- General Subroutines
mbbtimer.asm   -- Timer pop handler
mbbusvc .asm   -- User request service routine
desqview.asm   -- Interface to desqview

mbcnf   .asm   -- ASM file configured for COM1 and two PC-100 ports
```

## Configuring

Use MBBCONFG to configure the tables inside MBBIOS with regards to your particular hardware configuration. All ports which share interrupts must be defined! Thus if you a port on a PC-1XX, define both ports. If you use a port on a MS-400 with the diode modification, you must define all the ports with diodes on them.

## Parameter explanations

The HIGH SPEED option is used to shift the data rate table. See the explanation on AH=0 below.

TRANSMIT BUFFERING allows MBBIOS to return control to the calling program before the character is fully sent.

HARDWARE HANDSHAKING makes MBBIOS follow the RTS/CTS handshaking system used by many TNCs. ***NOTE*** TNC-1s are no longer supported for handshaking. If this is a problem, use the MBBIOSDT.COM file instead. This does not affect TNC-1 with AA4RE BBS since that uses host mode.

Recommended settings:

| Type BBS --> | RLI/MBL/BQE | Host mode (e.g. AA4RE BB) |
|---|---|---|
| Transmit Buffering | Try ON. If you get hangs, OFF | Must be ON |
| Hardware Handshake | Must be ON | Must be OFF |

## Using

Simply execute MBBIOS before running the program that needs it. If you wish to unload MBBIOS for some reason, issue MBBIOS /U.

Desqview users should specify /d when loading. Example: MBBIOS /D

## Software Notes

```
--------------------------------------------------------------------------------
!                                                                              !
!  ASYNC + Extended Support for BBS                                            !
!                                                                              !
--------------------------------------------------------------------------------
```

MBBIOS supports the standard INT 14 calls plus the extended ones as defined by Jeff Jacobsen, WA7MBL for his BBS program. These are explained below. AH=0 to AH=3 are identical to the IBM BIOS definition.

## AH = 0     Initialize

This initializes the communications port. AL contains parity, stop bit, and word length as in the standard BIOS call. On return, AH contains the Line Status Register information (except bit 0 - Data ready tells whether there is information in the extended buffer rather than the SIO chip receive buffer). AL contains the Modem Status Register.

The baud rate is selected via AL. If the high speed option is off, the rate is a specified for the standard BIOS call. If the high speed option is on the speed settings are from the alternate table

High speed option off:

```
   MSB                                                     LSB
    7       6       5       4       3       2       1       0
    ----data-rate----       -Parity--    Stopbits   --Word length--

    000 -   110             X0 - none      0 - 1    10 - 7 bits
    001 -   150             01 - odd       1 - 2    11 - 8 bits
    010 -   300             11 - even
    011 -   600
    100 -  1200
    101 -  2400
    110 -  4800
    111 -  9600
```

High speed option on:

```
   MSB                                                     LSB
    7       6       5       4       3       2       1       0
    ----data-rate----       -Parity--    Stopbits   --Word length--

    000 -    9600           X0 - none      0 - 1    10 - 7 bits
    001 -   14400           01 - odd       1 - 2    11 - 8 bits
    010 -   19200           11 - even
    011 -   28800
    100 -   38400
    101 -   57600
    110 -  115200
    111 -  330400
```

The high speed option can be set either by MBBCONFG or by software using AH=9 (see below).

## AH = 1     Send Character

AL contains the character to send. Return values are the same as above.

## AH = 2     Receive Character

On return, AL contains the character received. AH contains the composite receive LSR. This is done by ORing all LSRs on receive interrupts together. Therefore the LSR returned does not necessarily match the LSR for the the character this routine returns). Again, AH bit 0 indicates if there is additional characters in the buffer. This routine does NOT timeout while waiting for a character but will hang indefinitely waiting for a character to appear.

## AH = 3 Return Status

Return values are the same as returned by the Initialize routine.

## AH = 4 Inquiry

AH returns 0AAH. AL returns 055H. This call is used to see if the MBBIOS driver has been loaded for a particular port.

## AH = 5 Drop Handshaking

This drops the handshaking signals. Used to tell the TNC not to send any more information. No return value.

## AH = 6 Raise Handshaking

Turns the handshaking bits back on.

## AH = 7 Send Break

Generates a "Break". Useful for getting TNC-2 or PK-232 into Command mode from Converse or Transparent mode. (However, if the TNC is sending characters at the time, they will be garbled.) No return value.

## AH = 8 Non-destructive Read

Returns the next character in the buffer in AL without removing it from the buffer. AH value defined as in AH=2 so you can use the Data Ready bit to check to see if you really did get a character.

## AH = 9 Set/Get options

Set the current option byte to AL returning the old option byte in AL. The option byte is defined as follows:

| | |
|---|---|
| .......xB | Transmit buffering (1 = on, 0 = off) |
| .....x..B | Hardware handshaking (1 = on, 0 = off) |
| ..x.....B | High speed option (1 = on, 0 = off) |
| xx.xx.x.B | Reserved. May be in use on other types of devices |

## AH = A Write buffer

Sends CX characters starting at DI:ES. Upon return, DI will be updated and CX will contain the unsent character count. Unsent characters could result from either BUFFER FULL or hardware handshake is OFF. AX will be set as in AH=3 (Status) above.

## AH = B Read buffer

Receives up to CX characters starting at DI:ES. Upon return, CX set to actual count read and DI is updated. If the buffer is empty, return is immediate with CX = 0. On return, AL contains the MSR and AH has the composite receive LSR. This is done by ORing all LSRs on receive interrupts together. Therefore the LSR returned does not necessarily match the LSR for the the character this routine returns). Again, AH bit 0 indicates if there is additional characters in the buffer. This could happen if characters were received between the time of the buffer movement and the loading of the LSR into AH.

## AH = D Control

### Subcommand AL = 0

Gives the available bytes in the buffers. Returns AX = bytes available in TX buffer and CX = bytes available in RX buffer.

### Subcommand AL = 1

Turns off all modem controls (DTR RTS, etc).

### Subcommand AL = 2

Turns on all modem controls (DTR RTS, etc).

### Subcommand AL = 3

CL is used to set the handshake byte.
On exit, CL contains the previous handshake byte.

## PACCOM PC-100, PC-110, PC-120 support

This may not be included in this version...

### AH = 0     Initialize

Parms

```
 xxx..... = Data rate.  Use same settings as IBM ROM BIOS.
            300 bps = 2, 1200 bps= 4.

 ...x.... = Full duplex if 1.  Half duplex if 0.

 ....x... = Error packet handler.  0 = throw away all error
            packets.

 .....xxx = Don't care.  Use 0 for now!
```

Special case -- If AL = 0 then the COM port is closed!

The bits shown are used for parity on an ASYNC port and will be zero for a KISS interface so it should be compatible with TCP/IP usage of INT 14.

### AH = 1,2   Send/Receive Characters

See above. Packets are handled in KISS format. Note that all commands which set time are rounded to the nearest 50ms value.

### AH = 3     Status

```
Returns AH.. AL=0
    0.00000. = not assigned.  Zero
    .x...... = If 0 then packets remain in the transmit queue
    .......x = If 1 then data await in the receive queue
```

These bits correspond to THRE and RDA so are compatible to ASYNC ports

### AH = 4     Inquiry

AH returns 0AAH. AL returns 055H. This call is used to see if the MBBIOS driver has been loaded for a particular port.

### AH = 10    Send a packet

ES = Segment of buffer containing the packet to be sent. Note: The buffer pointed to by ES will be freed when the packet is sent unless this is turned off. It should be gotten via Opcode 12. It must be 512 bytes long and conform to the layout of the buffers

```
           Buffer Layout:

               0 - 503 -- data (504 bytes long)
             504 - 505 -- 16 bit data length
             506       -- flags/status #1
                            x........ = If 1 then don't discard
                                        this buffer after xmt.
                            .x....... = If 1 then this buffer has
                                        been transmitted
                            ..0000000 = not used
             507       -- flags/status #2
                            000000000 = not used
             508 - 509 -- User field.  Not used by MBBIOS
             510 - 511 -- MBBIOS uses as a pointer to next buffer in chain
```

### AH = 11    Receive a packet

Returns ES with segment of buffer containing the packet received. If zero, no such packet exists.

### AH = 12    Buffer management

If ES is zero then returns a buffer segment in ES. If ES is non-zero then the buffer pointed to is freed.

### AH = 13    Set TXD -- Time from RTS to start or packet

AL = new setting in clock counts (54.3 ms)

### AH = 14    Set P (persistance) -- Controls time from end of DCD to RTS

AL = new setting. 255 = 1.0

### AH = 15    Set Slot time -- Controls time from end of DCD to RTS

AL = new setting in clock counts (54.3 ms)

### AH = 16    Set CRC wait -- time from start of last character to drop RTS

AL = new setting in clock counts (54.3 ms)

Should be 5 character lengths minimum