

CSS

Black

BOX

Version 1.0

10/12/1989

By Robert Puff

The BLACK BOX!

Version 1.0 10/12/89 By: Robert Puff

(C) 1989 by Computer Software Services

1350 Buffalo Rd. #2 Rochester, NY 14624

Telephone: (585) 429-5639

INTRODUCTION

The Black Box is a parallel bus device designed for the Atari 600XL, 800XL, and 130XE home computers, which greatly enhances their capabilities and usefulness. The Black Box (we will refer to it as 'BB' throughout this document) provides several operating system enhancements.

The BB started out to be a simple hard disk interface, but soon evolved into a much more powerful product. We have striven to support even some of the less common peripherals that until now, have not been able to be used with an 8-bit (including some modems and hard disks). Computer Software Services believes in the original Ataris, and will, with your help, continue to develop innovative products for the 8-bit line.

The following is a brief list of the BB's features:

- * RS232 serial port with built-in handler
- * SASI/SCSI hard disk bus
- * Parallel printer port definable of printer IDs 1-8
- * 64k Printer spooler using XE memory or optional BB memory
- * Text or Graphics printer dump of most screens
- * Hard disk floppy drive emulation (Single or Double density)
- * 512 byte sector support for large hard disks
- * 96 entry partition list for ease of drive manipulation
- * Configuration menu allowing manipulation of drive numbers, swapping drives, write protection, and density (hd only)
- * All hard disks may be write-protected with the flip of a switch
- * Ultra high speed disk I/O with modified 1050s, Happy 810s, and all XF-551 floppy drives
- * Built-in machine language monitor allowing you to see the program counter and registers, change them, disassemble memory, and search for bytes

A separate product for the BB is **The Floppy Board**, a floppy disk interface. This will permit the connection of 'bare-bones' 3.5 and 5.25 inch floppy drives, without the need for a special

controller. Because this interface board will be on the computer's parallel bus, the data transfer rate will be MUCH faster than that of even a floppy drive in high speed mode!

PORT OVERVIEWS

The RS232 Port

The BB's RS232 port provides the full RS232C specification signal levels for modems, or other serial devices. It emulates the Atari 850 interface very closely. The handler (a program that tells the computer how to talk to the port) is built-in, and takes up NO user memory. The BB's RS232 port handler can be accessed using the "R:" device, and works with all tested terminal and BBS programs.

If you are already using another serial interface, you can disable the BB's port by using the configuration menu (explained later). Note that although you may use this port to drive a serial printer, we do not guarantee compatibility with existing printer software. Unfortunately, the Atari was never designed to use a serial printer.

The RS232 port is located on the left side of the BB, next to the push-button switches. It is a 34 pin card edge connector, with pin one located on the lower corner. You need a cable with a DB-25 connector on one end, and a 34 pin edge connector on the other. CSS has modem cables that may be ordered, but you can also make one out of parts from Radio Shack. We suggest using ribbon cable and crimp-on connectors. 34 conductor ribbon cable is widely available, but for this port you only need 25 conductors. Simply line up pin one of both connectors with the index wire (usually one of the edge wires have a red stripe), and crimp each connector onto the cable using large pliers or a vice.

These are the pinouts and signal names supported in the BB:

Signal Name	Originator	RS232 Pin #	BB Pin #
Transmit Data	BB	2	3
Receive Data	Modem	3	5
Req To Send	BB	4	7
Clear To Send	Modem	5	9
Data Set Ready	Modem	6	11
Signal Ground		7	13
Carrier Detect	Modem	8	15
Data Term Ready	BB	20	14

Note: the apparent discrepancy between the BB's numbering system and the RS232 connector's pinouts is due to the way the connectors are numbered. The BB was designed so that no special wiring order was necessary; it matches perfectly when using crimp-on connectors.

The Parallel Printer Port

The BB's printer port interfaces to most all Centronics-type printers. It emulates the 850's parallel printer port for full compatibility. The BB provides a longer timeout (approximately 30 seconds as

opposed to the 850's 5 seconds), so printing data with a lot of head movement will not cause spurious errors.

The BB may respond as any single printer ID 1 through 8, or any printer number (this is all set by using the config menu's Port Statuses page). This is to provide for those who have more than one printer online (the second printer being connected through a SIO interface). Setting the printer ID to ALL will account for programs that improperly call the printer directly (BUG/65 is an example of this).

Another way to have two printers connected to your system is to enable the FAULT line (sometimes called the ERROR line). This is done on the BB by turning dip switch #1 OFF. (Normally we recommend this switch be ON.) If the BB sees that FAULT is high, it will not use its own printer port, but rather let the print command go to a SIO device (850, P:R:, etc.). FAULT can be set high by simply taking the printer offline (most printer have an online switch or button). So by taking the printer connected to the BB offline, all printing data will go to the printer connected through the SIO daisy chain.

The BB provides the capability of a 'print spooler'. This is a large buffer (approximately 60,000 characters long) that will allow you to quickly 'print' your document. What happens is, the BB stores all the data to be printed in the memory you specify, then sends it to your printer all by itself, while you are doing other things with your computer. You must, however, have your computer turned on while this is happening, since the BB uses your computer's processor to perform its tasks. This spooler works only with the BB's printer port.

You may also perform a "print screen" function, by pressing one of the buttons on the BB. In text mode, only text messages will be printed. In graphics mode, a mirror image of the screen will be dumped (in black/white only). Again, this function works only with the BB's port.

CSS sells BB Printer cables, but you can build your own from Radio Shack parts. All you need is a 36 pin male centronics crimp-on connector, a 34 pin edge connector, and some 34 conductor ribbon cable. Align pin one of both connectors to the same wire of the ribbon, and crimp the connectors on. The maximum cable length should be 20 feet. Note that there will be two unused pins on the centronics connector.

These are the BB's pinouts for the Parallel Printer Port:

Signal Name	Originator	Cent. Pin #	BB Pin #
-Data Strobe	BB	1	1
Data bit 0	BB	2	3
Data bit 1	BB	3	5
Data bit 2	BB	4	7
Data bit 3	BB	5	9
Data bit 4	BB	6	11
Data bit 5	BB	7	13
Data bit 6	BB	8	15
Data bit 7	BB	9	17
Busy	Printer	11	21
-Fault	Printer	32	28
Signal Ground		16	31

Once again, the apparent mismatch in the numbering is because the 36 pin centronics D-type connector is numbered differently than the 34 pin edge connector. It is designed so that you can use ribbon cable and crimp-on connectors, with no soldering required. Pin one of the BB's printer connector is located on the top, next to the hard disk port.

The Hard Disk Port

The BB's main attraction is its SASI/SCSI hard disk interface. The BB supports many types of hard disks and controllers; a list is given in Appendix A of ones known to be compatible.

The typical hard disk system consists of a hard disk drive, a hard disk controller, a host adaptor (the Black Box), and a power supply. The compatibility issue primarily is the controller-to-host connections. For this reason, only SASI or SCSI controllers can be

used. Controllers designed for IBM PCs will NOT work. SASI controllers (pronounced 'sassy'), which include the Xebec 1410 and Western Digital WD1002SHD, were the 'pioneers' in hard disk technology. Since then, SCSI controllers (pronounced 'scuzzy') have become more popular. The Adaptec 4000 series controllers are examples of SCSI devices.

Some hard disks have built-in controllers. These are called Embedded SCSI drives. Seagate drives ending with the suffix N (such as the ST-125N) are embedded drives. They save on space and power requirements, since there are less components (no external controller is needed). Their only disadvantage is if you want to hook up another hard disk (that is not an embedded SCSI), you will need a controller card.

The BB can handle up to 8 controllers, the limit of the SCSI specification. Most controller cards can handle two hard disks. Embedded SCSI drives count as one controller, but only control the single drive attached to them. A 220/330 ohm resistor termination pack should be installed in the last controller on the cable. Remember that the hard disk bus cable should not be longer than 20 feet.

The following are the pinouts for the SASI/SCSI Bus:

Signal Name	Originator	Pin #	Signal Name	Originator	Pin #
-Data bit 0	Either	2	-Busy	Controller	36
-Data bit 1	Either	4	-Acknowledge	BB	38
-Data bit 2	Either	6	-Reset	BB	40
-Data bit 3	Either	8	-Message	Controller	42
-Data bit 4	Either	10	-Select	BB	44
-Data bit 5	Either	12	-Cmd/Data	Controller	46
-Data bit 6	Either	14	-Request	Controller	48
-Data bit 7	Either	16	-In/Out	Controller	50

All lines are inverted; low means true. All odd numbered pins are connected to ground. There are two other signals supported by some controllers, but these are not needed by the BB.

This connector is located on the top right of the BB, and is a right-angle 50 pin header connector. Pin one is nearest the right corner of the BB (right next to pin one of the parallel printer port). Some header connectors have an index tab on one side of them; this tab will need to be broken off before the connector can be inserted.

SYSTEM INSTALLATION

The BB will hook up to any 600XL, 800XL, or 130XE computer. To connect the BB to an XL machine, first remove the little plastic piece that covers the Parallel Bus connector on the rear of the machine. This will expose a finger connector. Place the BB so that the bottom of the T shape is by this connector. Now push the loose end of the short ribbon connector into your computer's Parallel Bus connector. The metal shielding may cause slight difficulty; but be careful not to apply too much pressure that might cause damage to the circuit board of the computer.

The parallel port cable is (and should be) very short, because larger cables will cause RF interference. This cable will cover up the cartridge connector on the BB. This is intentional, because the BB's cartridge connector is not usable on XL machines.

If you are installing the BB on a 130XE machine, you do not need this little cable. It may be removed by gently pulling the connector on the BB straight up, with equal pressure on both sides. The BB's bottom edge is designed to fit right into the Cartridge and ECI ports. You'll notice we have provided a vertical cartridge port on the BB to replace the now-occupied horizontal cartridge port being used by the BB (a welcome change for most XE users!). The majority of cartridges will simply plug in, with the exception of the spring-loaded cases on Atari brand cartridges. To insert these, you'll have to trip the spring mechanism by inserting a thin screwdriver or object into the bottom of the cartridge case and retract the bottom portion of the case, thereby exposing the edge fingers of the cartridge's circuit board. With the bottom of the case retracted, go ahead and insert your Atari cartridge.

The BB should be on whenever your computer is on, or else the computer will not function. For this reason, there is no power switch. The power connector will only work when it is installed the proper way. If it is reversed, the BB will simply not work. The BB's power pack draws minimal current, and nothing can be damaged by leaving it plugged in at all times.

Now you're ready to plug in the peripheral cables. Here's a crude drawing identifying the ports:

(oops, haven't scanned the pic yet!)

The * represents the location of pin one for each connector. The floppy port is for a future floppy disk interface card.

Switch Description

The first light on the BB is the power indicator; the second will glow whenever the BB's ROM memory is being accessed.

There are two push-button switches, and two toggle switches on the BB board. The left-most push switch is used to enter the BB's menu. From here you may edit the system configuration, or use the

6502 monitor program. This menu can be invoked from most anywhere in any program. The only time you should not enter the menu is when reading or writing to a disk drive.

The second button and the second toggle switch (the right-most switch) are used for the built-in printer screen dump. When the right button is pushed, the BB will send a mirror of the screen image to your printer, if it is connected to the BB's printer port, and if it is online. The second switch is the graphics switch. If it is off, the BB will only print text information, translating any non-printable characters to periods. If the switch is on, an exact copy of the screen will be printed, including ALL characters. We only intended this function to work with standard text screens, but as it turned out, it does a fair job with some graphics modes as well. Due to the fact that a printer cannot print colors, color graphics may not print very well. Pressing [BREAK] will abort the dump. If you abort a graphics dump, turn your printer off then back on to eliminate any graphics mode settings.

The left toggle switch is used to write protect all hard disk partitions. If you're running a program that you are unsure of, its a good idea to write protect your drives so that if the program decides to write some bad data, or worse yet, format your hard disk, you are safe. You may of course write protect a single part of your hard disk with the menu, but this switch takes care of all drives.

The dip switches are used to set the default configuration, and various other parameters. These are their meanings, and factory settings in parenthesis:

- Switch 1: Force printer FAULT line to NO FAULT (ON)
- Switch 2: Enable hard disk port/high speed floppy SIO (ON)
- Switch 3: Enable parallel printer port (ON)
- Switch 4: Enable RS232 modem port (ON)
- Switch 5: Enable printer linefeeds (OFF)
- Switch 6: Printer is Prowriter (OFF)
- Switch 7: MIO compatibility mode (OFF)
- Switch 8: unused (OFF)

The above table describes the action taken when the switch is UP, or in the on position.

Switch 1 controls the FAULT line coming from the printer. This signal basically reflects an off-line condition of the printer. If the BB sees that FAULT is true, it will NOT send any data to the printer, but rather will drop down to the SIO level, and use any other printer (interfaces) you may have connected. This way you can have two printers on at the same time. If no other printers are connected, then an error 138 will be returned. We recommend if you only have one printer to leave this switch on, because if the printer is offline when you go to print, the BB will wait for 30 seconds for it to return back online before returning an error. If you are using the print spooler option, there is no timeout value until you fill up the buffer.

Switch 2 enables the main BB functions of disk functions. By turning this switch off, you disable the BB's influence on disk operations. This is a quick way of disabling all hard disks or the high speed code for floppy disks, in the event of a compatibility problem. This should normally be on.

Switch 3 enables the BB's printer port. This switch, along with switches 4 and 5, are checked by the BB upon initialization, and are really only used for non-hard disk systems. The BB stores a system configuration on the first hard disk in the system. If you have a hard disk connected, the BB will use the parameters in the config rather than the dip switches.

Switch 4 enables the BB's RS232 port. The handler entry will automatically be placed in the HATABS table upon every RESET.

Switch 5 will cause line feeds to be sent to the printer after each carriage return. Some printers require this; others have their own dip switch settings to control this. If your printer is printing every line on top of the preceding line without advancing the paper, then you need to enable line feeds. On the other hand, if you are getting a blank line between each line of text, line feeds need to be turned off.

Switch 6 is used to define the printer type used for the graphics screen printer dump. If you have a Prowriter printer, then turn this switch on. If you have an Epson, Epson compatible, or your printer does not support graphics, then leave this switch off.

Switch 7 is used to enable MIO compatibility. The BB's hard disk port differs slightly from the MIO board. The MIO stores all data on hard disks inverted. This is fine as long as the MIO reads it, but a host adaptor that functions properly will get garbage data. Setting this switch on will cause the BB to invert all data when doing hard disk I/O. This pertains to ALL hard disks connected. When this switch is on, you may not have any partitions larger than 16 megs. We provided this for MIO users who now use the BB, and already have hard drives full of data. If you reformat your hard disks, we recommend you format with this switch off.

Connection of the BB to a modem or printer is straightforward; connecting hard disks and their controllers requires some thought.

Hard Disk Installation

The BB supports up to 8 controllers, numbered 0-7. All controllers connected must be set to a different ID (since they are all connected in parallel), with the first starting at 0. Hard disks with embedded SCSI controllers will have the jumpers on its board for setting the ID. Some controllers will have a jumper to enable parity checking; **MAKE SURE THIS IS DISABLED!** Most embedded SCSI drives are shipped with this jumper installed, making the drive appear not to work. Most SASI controllers also have a jumper to select the sector size. Set this as you desire. Its always a good idea to read carefully the documentation of your controller and drive to make sure everything is set up correctly.

The last controller on the 50 pin SASI/SCSI bus must be terminated with a resistor pack. Usually this consists of a single resistor pack looking much like an integrated circuit chip, however, some controllers use two or more single inline resistor packs (these are normally yellow). Controllers shipped from the factory should have the termination packs already installed. The only time you need to remove these are when you have more than one controller: the controller(s) between the BB and the last unit should have their termination removed. Some SCSI embedded drives have a jumper for terminator power; this should be set to the drive's power supply, not the bus line power.

So far all we've been concerned about is the controller card. The controller is the only thing to which the BB directly interfaces. Your hard disk mechanism(s) connect up to the controller. A typical SASI or SCSI controller has a 50 pin header connector (the SASI/SCSI bus connector, which

hooks up to the BB), a 34 pin connector (which goes to the 34 pin connectors of your hard disks), and two or more 20 pin header connectors (which connect to the individual hard disks).

The termination principle also applies to the 34 pin cable coming from the controller card, attaching to each hard disk. The last hard disk in the line should have a termination pack installed, the drive in the middle should have it removed. You also need to be sure the hard drive's unit number jumpers are set correctly. Some drives have them numbered DS1-DS4, others 0-3. Your first drive should be jumpered accordingly, and so on. You will not have a 3 or 4, since most controllers only support a maximum of two hard disks. Some drives are shipped with all drive number jumpers installed; make sure only one is enabled.

Each hard disk has its own 20 pin connector on the controller card; consult your controller manual to find the proper connector for each drive.

THE BLACK BOX MENU

The menu is the heart of the BB. It is where all BB functions are controlled. To enter the menu, press the left push button. You should be able to do this in just about any program, at any time. It is not recommended you enter the menu in the middle of disk operations. The BB does not destroy any of the computer's memory for this menu, so your program remains intact when you exit.

Once in the menu, you will see these options:

- [A] Drive Configuration
- [B] Port Statuses
- [C] 6502 Monitor
- [ESC] Return to program

Pressing A-C will take you into sub-menus. Pressing [ESC] will return you to the next higher level menu, and back to your program if you're at the above main menu.

In any of the sub-menus, the method of editing is the same. Use the cursor arrow keys (you do not need to hold the CONTROL key) to position the cursor bar on the desired item, then press [RETURN] to modify. If it is a toggle (such as SASI/SCSI) or a single digit change (such as the floppy drive number), pressing [RETURN] will cycle through the possibilities. If it is a text or numerical input, press [RETURN], type in the new data, then press [RETURN] again. Pressing [ESC] will leave the field unchanged.

The Drive Configuration Page: Fields

Pressing [A] from the main menu will take you to this page. You will now see a list numbered from one to nine. This is the main drive configuration, where each of the nine drives (D1:-D9:, as the computer sees them) are defined. We will call these the logical drive numbers. There are two sub-menus from here: the Controller list, and the Partition list. They are sub-menus off this menu,

because they will not be used as often as this main drive config menu. First though, we will discuss the drive config.

The first vertical field is the partition name. If the drive is a hard disk, you may assign a name to the partition, and enter it into the partition list (more on this later). If the drive is defined as a floppy drive (connected to the computer's SIO port), then the name should be 'Floppy Drv'. You can flip back and forth between floppy and hard disk by pressing [RETURN] twice when the cursor bar is in this field.

The Partition name may be altered only when the drive is configured as a hard disk. See the WRITE PARTITION command for information on how the name interacts with the partition list. The reason for the partition name is so you can easily identify each partition of your hard disk. This may not be needed if you only use one or two partitions, but becomes invaluable when a large number of 720-sector floppy style partitions are used.

The Partition ID field is set whenever you execute the WRITE PARTITION command, and identifies where it is in the 96-entry partition list. This is only for cosmetic appearance, and does not actually control anything. For this reason, it cannot be edited. If this partition has not been written to the list, this field will contain "??".

The Controller number field identifies which controller and hard disk is associated with this logical drive. Each hard disk has its own entry in the Controller config (explained below). They are indexed by letters, so as not to be confused with floppy drive numbers. Your first hard disk would typically be defined as the 'A' entry of the controller list; so when setting up a partition for that drive, you would set the Controller number to 'A'.

This field also indicates the drive number if the logical drive is set to be a floppy. With this field, you can redefine a logical drive to actually use any floppy drive, one through nine. For example, you can set logical drive 1 to be floppy drive #2, and then boot a program from drive 2!

The fourth field is the Starting Sector number. Sector 0 on a hard disk is reserved for BB use; all the rest are free for partitioning. See the section PARTITIONING HARD DISKS for more information. If you are using 512 byte sectors, the starting sector number MUST be an odd number. If you enter an even number, the BB will round to the next higher odd number.

If the logical drive is configured as a floppy drive, this field will display "US:" and a Y or N. The BB has ultra high speed SIO routines for modified 1050 drives, Happy 810s, and XF-551 drives. For most all purposes, this will be a great benefit, speeding up floppy disk access considerably. If you run into an incompatible program, you may turn this feature off by setting the field to 'N'. The speed may also be toggled by pressing SELECT and OPTION together while the drive is being accessed. This will take it down to standard speed. The high speed may then be re-enabled by holding START and OPTION together while the disk is being accessed, or by hitting RESET. Note that the remaining three fields are not applicable to floppy drives.

The Length field is the number of sectors the partition contains. Note that there is a minimum of 720; this is to prevent potential problems with certain programs. Most DOSes were written for floppy drives only, and will not be able to access anything above 720 sectors. MYDOS and SpartaDOS are the only two that are designed to work with hard disks. MYDOS 4.5 (and all previous versions) and SpartaDOS have a limit of 16 megabytes (65535 sectors) per logical drive. MYDOS 4.60 (soon to be released) has an upper limit of 48 megabytes (196,607 sectors) per logical drive. The

limit of 16 megs may seem to pose a problem for hard disks 20 megabytes and larger, but because a hard disk can be partitioned into smaller drives, this eliminates the problem.

The next field is the Write Protect flag. By setting this to a 'Y', that logical drive will be write protected. Reads are allowed, but writes to the drive will generate an error 139. Note that all hard disk partitions may be write protected by turning on the left toggle switch.

The last field is the Density selection. All partitions larger than 720 sectors should use double density (256 bytes/sector). One of the unique features of the BB is the ability to copy boot programs to partitions. Simply set up a 720 sector partition, and use this flag to set the density to single. The partition will now act just like a single density floppy drive, only much faster! In single density mode, the second half of each sector is not used.

The Drive Configuration Page: Commands

You may exchange any of the logical drive numbers with the [E] command. Simply place the cursor bar on the line of one of the drives, press [E], then type the number of the drive with which you want to 'swap'. This is handy for changing drive 1, so that you can boot different partitions. You could edit the configuration of logical drive 1, but using the [E] command is much easier.

The [W]rite Partition command will add the current hard disk entry to the partition list, if the list has been set up. See THE PARTITION LIST PAGE for more information. If the list has not been formatted or there is no more room in the list (all 96 entries are filled), the error "No more room in list" will be displayed. If you have already entered the current partition into the list (characterized by a number and letter in the Partition ID column), you may change the name in the list by first changing the name of the current logical drive, then pressing [W]. If you try to re-write a partition entry having modified any of the data (such as controller number, starting sector, etc.), the BB will warn you that an entry already exists. You will be given the choice of overwriting the old entry, or adding this as a new entry.

Once you have established the drive configuration you desire, it should be saved it by pressing [S]. This will write to sector 0 of hard disk 0 with a controller ID of 0. The BB will alert you if it could not write to this drive. The BB will read in this configuration whenever the unit and computer are powered up. It is retained for as long as the BB is on.

If you want to restore your default config (incase you inadvertently altered some important data), press [L]. This will reload the saved configuration. (Note that this command is not shown on the BB menu.)

Pressing [C] will take you to the Controller List page, which we will cover next. Pressing [P] will take you to the Partition List page, which is covered below.

The Controller List Page

The controller list contains one entry per physical hard disk. They are identified by letters instead of numbers, so as not to be confused with logical drive numbers. The controller list identifies which physical drive will be used for each entry in the drive configuration.

The first field is the interface type. Most controllers are the SCSI variety; the Xebec 1410, Konan DJ210, and the Western Digital 1002-SHD are SASI. The SMS FWD5001 controllers are SASI only when in 512 byte sector mode.

The second field contains two values: the Bus ID number, and the drive Logical Unit Number (LUN). The bus ID number is the ID of the controller. Your first controller should always be 0. See your controller manual on how to set its bus ID if you need to alter its default. Note that some controllers (namely the SMS FWD5001) have no way to change its ID; they are fixed at 0.

The Logical Unit Number is the ID of your physical hard disk. Most controllers can handle two drives, numbered 0 and 1. Most hard disks have jumpers on their analog board to set the ID. Beware that some manufacturers use 0-3, others use 1-4. The BB numbers them 0-3. Use the first ID for your first drive, and so on.

The Heads and Cylinders are characteristics about your hard disk; consult the data you have on your hard disk, or check the table in appendix B for some of the more common drives. This data is important, for it tells the BB the capacity of your drives.

The Step Rate field is drive-dependent. Most all drives can handle what's called buffered seeks. To use buffered seeks, use a value of 7. One of the few drives that cannot handle buffered seeks is the Seagate ST506; Use a value of 0 for these.

The Sector Size is the last field, and will toggle between 256 and 512. You may use 512 byte sectors with the BB. Since Atari 8-bit DOSes cannot handle anything greater than 256 byte sectors, the BB's 512 byte sector mode works by splitting each 512 byte sector into two 256 byte sectors. In this mode write operations are a little slower, but reads can be slightly faster than normal, because the sector is stored in the BB's memory (a 1 sector cache). Some embedded drives will not support anything BUT 512 byte sectors, so many more drives and controllers can be used on the BB. Another advantage to using the larger sector size is you will ALWAYS get more storage. Due to the overhead necessary for each sector, more data can be fit on a cylinder using 512 byte sectors than 256. The added amount is generally 2 sectors per head per cylinder, but can be more with some controllers (especially RLL).

The Partition List Page

The Partition List page allows you to manage several drive configurations. As previously explained, a partition is a section of a hard disk (it can be the whole drive, if your drive is small). The most common use of the partition list is for setting up small 720 sector floppy look-alikes. By doing this, you can sector-copy your favorite boot programs to the hard disk, and run them at hard disk speed! Since a typical hard disk has several thousand sectors, you can fit several of these 720 sector partitions on it. The Partition list allows you to have them all "on call", so that they may be used at any time, quickly.

To create a partition entry, first use the Configuration page to enter the parameters for a partition (starting sector, length, etc.). Now copy the desired program to this drive. Enter back into the Configuration page, position the cursor bar over the drive number you just configured, and change the name. You can enter up to 10 upper case letters or numbers, 9 of which will show up in the partition list. Now press [W]. This will save this partition into the partition list. You may now change the parameters for your next partition, and repeat the process until you are finished.

Whenever you want to use this partition, it must be copied to one of the logical drives in the main configuration. To do this, first enter the partition list. (From the main menu, type [A] for the drive config, then [P] to get to the partition list.) The list can hold 96 entries; 48 per page. Press [N] to go back and forth between pages.

Position the cursor over the desired name, then press [RETURN]. You will now be shown the main drive config, and prompted for a drive number. When you type a valid number, the information from the partition you selected will be copied into that drive's config, replacing what was there.

There are a few other maintenance commands in the partition list. Pressing [D] will delete the entry currently highlighted. Pressing [K] will delete all entries FROM THE CURSOR LOCATION to the end of the list. When you initially set up the partition list starting sector number, the list should be initialized. To do this, position the cursor on the first entry (it defaults to this location when you first enter the partition list), and press [K].

To use the partition list, it first must be defined. See the Port Statuses page (below), about setting the partition list starting sector. If this number is not defined, an error message will be displayed whenever the partition list is accessed.

The Port Statuses Page

This page enables you to define miscellaneous parameters relating to the BB's functions.

I/O Sound: Setting this to YES will enable "noisy I/O". The BB will produce a tone whenever the hard disk or printer are accessed, generating a different tone for writes to the hard disk. If you do not want the tones, set this to OFF.

RS232 Port: If you wish to use an 850 or other serial interface, setting this to OFF will disable the BB's RS232 port. This may also be needed if a program will not run with the BB attached.

Printer Port: The BB's parallel printer port may be set to any printer device number between 1 and 8, ALL, or OFF. If set to ALL, it will respond to any printer number. This is needed for compatibility with some software. Press [RETURN] to cycle through the possibilities.

P: Line Feeds: This pertains to the BB's printer port. If your printer is printing everything on one line without advancing the paper, set this to YES. If it is leaving a blank line between every printed line, set this to NO.

Spooler Stat: This toggles the printer spooler feature, and defines what memory will be used for the buffer. Setting this to OFF will disable the spooler. If this is set to XE BANKS, it will use the

extra 64k the standard 130XE has, allowing use of the rest of the memory for other applications. If this is set to INTERNAL, then the 64k of memory in the BB will be used. Note that only what your system is capable of will be displayed: If you have no extra memory in the BB or in your computer, then you will not be able to enable this feature.

You may empty the printer buffer by toggling this to off, then to back on (on being either XE BANKS or INTERNAL).

HD Part. Start Sec: This is the starting sector for the partition list. The partition list takes up 14 sectors, and should be considered when initially partitioning your drive. The sector number is not DOS related; it is a sector on your hard disk #0 of controller ID 0. If you enter 0, it will be 'undefined', and the partition list will not be functional. If you are using 512 byte sectors, this must be an even number. If your drive has already been partitioned without accounting for the 14 sector partition list, you may be able to squeeze it at the end of your disk, perhaps taking a few sectors away from the last partition. See the section on Partitioning Hard Disks for more information.

Note that the RS232 and Printer port settings in this menu override the dip switches that correspond to the same.

The 6502 Monitor

A machine language monitor program has been built into the BB. Its feel is similar to the DDT debugger in the MAC/65 Cartridge. All values are displayed in hexadecimal, and must be entered in the same.

Once you enter the monitor, the processor registers (at the time you pressed the left button to enter the menu) are displayed. The program counter will point to the instruction the processor was just about to execute. The disassembly window shows the next 11 instructions, their hex values, the ATASCII equivalent of the memory at the location shown, and the internal screen display codes.

The top lines display the 6502's program counter, Accumulator, X register, Y register, stack pointer (add \$0100 to this value to compute the actual memory address of the stack, since the 6502's stack is on page 1), and the processor status register flags. You may change the A, X, and Y registers by pressing [R] followed by the register letter, and the new hex value. The stack pointer and processor status register may not be changed, because doing so would produce unpredictable results.

Use the up and down arrow keys to scroll one byte/instruction at a time in the disassembly window. You may change the disassembly address by typing either [E] or [*], followed by the new address. If you want to change the program counter, type [P] followed by the address. Be EXTREMELY careful when modifying the program counter! When you exit the menu, the CPU will pick up at this address, and if it is pointing to invalid code (or even perhaps a different subroutine than what was being executed), your computer WILL lock up.

Use the [D] command to deposit bytes at the location of the top of the disassembly window. You may deposit up to 6 bytes per command. All bytes must be entered in hex.

The [S] command will allow you to search for a string of bytes, starting at the current location at the top of the disassembly window, up to \$FFFF. Enter the byte(s) in the order they are to be found (up to 6 maximum). The disassembly window will be set to the location for each match found. To keep searching, press [S] [RETURN]. The window address will not change when more matches are found. The search routine will search everything except the \$D000-\$D7FF hardware area.

SETTING UP A HARD DISK SYSTEM

There are five phases to setting up a hard disk system. It is not that difficult, but take your time. It is easy to overlook something small, and spend hours trying to find the problem.

Step 1: Connect all drives and controllers. Most controllers will need two cables running to their drive(s). Be sure to line up pin one on all cables to their appropriate connector. Pin one is usually identified by a '1' next to one side of the connector, other times it is identified by a notch or missing pin in the connector. Also remember to insure the termination packs are installed in the proper places: on the last controller, and on the last drive connected to each controller. Be sure the drive IDs are jumpered correctly. Connect your power supply connectors to all drives and controllers. Hook up the 50 pin SCSI bus cable to all controllers. (If you are using embedded SCSI drives, you only have the SCSI bus and power supply connectors.)

Before you connect the 50 pin SCSI bus to the BB, apply power to your hard disk(s) and controller(s). You should hear the drives spin up. If not, power down (quickly!) and check your connections. After everything appears ok, connect the SCSI bus to the BB.

Step 2: Hook up your BB to your computer, and power them up (plugging the BB in first). You do not need a floppy drive connected yet. Press the BB's left push button to enter the menu.

Press [A] for the Configuration menu, then hit [C] for the Controller list. Now we will enter the data for the hard disks and controllers used.

The controller type (SASI or SCSI) depends on your controller. The Konan DJ210, Western Digital WD1002-SHD, and the Xebec are SASI. Most all others (including embedded drives) are SCSI.

The ID,LUN specifies which controller, and what drive on that controller is to be used. The default values in the BB provide for 8 drives (2 drives per controller), but these values can be modified to your specifications. The first number is the controller id (0-7), and the second is the drive id. Note that with embedded SCSI drives, the drive id is always 0; the controller id is the only thing you can change (with jumpers). If you only have one drive on a controller, it should be jumpered as drive 0.

The number of heads and cylinders must be derived from your hard disk's manual. If you do not know what these values are, consult the table listed in Appendix B for some of the more common hard disks and their values.

The step rate is also dependent on the drive. Most all hard disks (all embedded drives) can handle buffered steps. The only drive we've found that does not is the Seagate ST-506. So for this

particular drive, use a value of 0. For all other drives, use 7. Note: the SMS FWD5001 controller needs a value of 2 for this field.

The sector size field determines how the drive will be formatted. Please see the explanation of 512 byte sectors in the beginning of this document, but it basically boils down to this: 512 byte sectors give you more storage at the expense of slower disk writes. If your drive has already been formatted, this field **MUST** be set to the drive's format, or your hard disk will pause after each sector read.

Step 3: Press [ESC] a couple times to get out of the menu, and boot the BB master disk. (Your drive 1 should be configured as Floppy drive #1; if it is not, enter into the menu, press [A], and change it.) Type DOS if you are not at the DOS menu. We want to run the physical format program, so type [L], then 'HDFMT.COM' [RETURN]. This physical format program will initialize each drive, wiping out any data that was on it before.

You will now be presented with a list just like the controller list. Press the letter of the desired hard disk to format. Now select the controller type. If you have a controller not listed, call us. We may have an updated format program to handle your controller. Since each controller usually requires a different format routine, we cannot cover all the bases, but we've provided support for the popular ones. Any new format programs will be posted on our BBS.

A confirmation prompt should now be displayed. Type [Y] to begin the format process. The format program will first initialize the drive, then it will go through and verify each sector. This can take a **LONG** time, especially with large hard disks, so don't hold your breath.

If the cable going to the controller is plugged in backwards (or is defective), you will see a message telling you to "Check your cables". Bad or twisted cables are the cause of 90% of setup problems, so much attention must be paid to them. Note that some embedded SCSI drives (the Seagate 3.5" drives, in particular) send power through the SCSI bus, and if it is plugged in backwards, can be shorted out, causing possible power supply and drive damage.

After the format process has finished, you will be given a total sector count, as well as a possible number of unresolvable bad sectors. Write this total sector count down! This will be used when partitioning the drive. Also note if there was a bad sector count at the end. The format program will attempt to re-assign bad sectors to other surrounding sectors, so your drive will appear error-free. Some controllers, however, do not support this; others partially support it. We will use this information in step 5.

Repeat this format process for all the hard disks connected, then proceed to step 4.

Step 4: Now each drive must be partitioned. See the section below entitled "Partitioning Hard Disks".

Step 5: The hard disk is now low-level formatted, but needs to be 'DOS formatted'. Exit the format program, and [L]oad the FMTDIR.COM program. You will now be using the BB menu to swap in each partition, then DOS-format it with this program.

The first question the FMTDIR program will ask you is the DOS type. MYDOS and SpartaDOS are the only two current DOSes that can handle any large partition (more than 1 megabyte). Type [M] or [S], depending on which DOS you will be using for that partition. There is no reason why you can't mix formats; but remember that MYDOS 4.5 will not read SpartaDOS formatted drives, and vice

versa. This program need not be run on 720 sector partitions; since they are the size of a standard floppy drive, any DOS can be used to format, read, and write to them.

If you saved all your partitions in the partition list (which we recommend), it will make life easier. Otherwise, you will need to enter the data for each partition that is not on the main configuration.

Start with the first partition. If you have already configured it as one of the current nine drives, then simply type its drive number at the "Drive to Format" prompt. Otherwise, you need to enter the BB's config menu, and enter the partition information into an unused drive number. If you have all your partitions stored in the partition list, then you can use the menu to assign the desired partition to an unused logical drive. If you only have the partition information on paper, then use the Configuration page to enter this data into the unused drive.

Now exit back to the DOS format program, and type in the drive number you used for the partition. You will be prompted for verification. The program will now ask you if you wish to verify the disk. This verification only needs to be done to partitions of drives that reported a bad sector count at the end of the physical format. This verification process is another lengthy one, but is the only way a bad sector can be found. If any are found, they will be mapped out at the DOS level, so you should never hit a bad sector with normal DOS usage.

Now repeat this step for all your partitions (that are longer than 720 sectors long). After you are finished, your hard disk(s) are ready for use!

If you ever want to reformat a logical drive (partition), you need only run the FMTDIR.COM program; the HDFMT.COM physical format program should only need to be run once.

After you are done, you may see a filename called "SECTORS BAD" on your drive. This is a file containing a list of all the bad sectors for that particular partition. It is written in a special format, so that programs like the VTOC FIXER and CLEANUP will see the bad sectors, and not free them in the table of contents area. This file should NOT be deleted. If you do develop additional bad sectors on your hard disk after use, it is best to reformat the partition(s) with the FMTDIR program to map out the bad sectors.

You will probably want to have one of your hard disk partitions be drive 1, so that DOS can be booted quickly. With MYDOS, simply use the [H] command to Write DOS files. If you are using SpartaDOS, you must first copy the DOS (X32D.DOS) to your hard disk, then use the BOOT command to enable DOS to boot automatically on the drive.

Partitioning Hard Disks

A typical 20 megabyte hard disk will contain approximately 78,000 sectors, but most DOSes have a limit of 65535 sectors per drive, or 16 megabytes. (MYDOS 4.60 will address this and provide for 48 megs/drive, but this is not yet complete.) To make use of all 20 megs, you need to 'partition' the drive. Partitioning involves splitting the drive into different drives. If your hard disk is smaller than 16 megs, you can either opt to use it all as one drive, or partition it into smaller drives.

The first step is deciding into how many drives you would like your hard disk split. Remember that the BB will let you use 720 sector partitions that will function just like floppy drives, so you may want to have a few of these. Also another factor to consider is this: if you put all your information on one partition, Murphy's law says the likelihood of that partition to be accidentally wiped out is much higher than if the data were split across two or more drives.

Anyway, decide on the number of drives, and the approximate size. Taking the total number of sectors the physical format program reported into account, calculate the starting sector number, and length of each partition. Remember the first sector on a drive is 1. The partition list takes up 14 sectors, and should be assigned on your first hard disk. You may want to leave a spare sector or two between each partition, for safety. If you are using 512 byte sectors on the drive, starting sector numbers must be odd numbers, so leaving a sector spare when calculating your partition information will allow for this.

The best way to show you how partitioning is done is by an example. In this example, we have a 20 megabyte hard disk, having a total of 83231 sectors. Let's partition this drive into two 720 sector floppy-type drives, and two 9 (or so) meg drives. We'll use one of the 720 sector drives as drive 1, the large partitions as drives 3 and 5, and the other 720 sector partition we'll store in the partition list (for use later). Here's our layout:

Graphic representation of sample hard disk:

Oops, another pic yet to scan.

Each partition start where the last one left off. We also followed the 512 byte sector rule: the starting sector must be an odd number, and the partition list starting sector must be even. The ending sector number is equal to the length added to the starting sector minus 1.

Now here's how this information will look on the BB's Drive Configuration Page:

Drive Configuration Page

D:	Partition	Ctl	Start	Pro
#	Name:	ID #	Sec.	Length Dns
1	HD 1	?? A	17	720 N D
2	Floppy Drv # 1	US:Y	-----	- -
3	HD 3	?? A	1457	36000 N D
4	Floppy Drv # 4	US:Y	-----	- -
5	HD 5	?? A	37457	45760 N D
6	Floppy Drv # 6	US:Y	-----	- -
7	Floppy Drv # 7	US:Y	-----	- -
8	Floppy Drv # 8	US:Y	-----	- -
9	HD 9	?? A	737	720 N D

You'll notice we configured drive 9 to be the spare 720 sector partition. We need to configure this in the menu before writing it to the partition list.

We would now go to the Port Statuses page, and set the starting sector for the Partition List to be 2.

The Partition List needs to be initialized, so we would go to it by pressing [ESC], [A], then [P]. Typing [K] will delete any junk entries, making the partition list useable. Now we would go back to the main drive configuration, and save each hard disk entry to the partition list by pressing [W] after the cursor bar is positioned on each hard disk entry.

You could now change drive 9 back to being a floppy drive. Now we would type [S] to save this config. Each time the system is booted, it will load the drive configuration and the port statuses information from the hard disk.

The RS232 Handler Functions

The BB's serial interface follows the protocol of the 850. There are a number of parameters in setting up the port, so it is not as easy to use as is the printer port. You must consider the BAUD rate, number of bits per character, number of stop bits, the handshake lines (DTR, CD, DSR, etc...), and translation.

This section will list all of the calls to the RS232 port in Atari BASIC's format. If you are using another language, check with its manual for the proper syntax.

The device name for this port is "R:". The handler ignores any port numbers that may be specified.

Default Conditions

If no configuring XIOs are performed, the set of default parameters will be used. Once you set a parameter, it will retain that value until a SYSTEM RESET is done. These are the default values: 300 Baud, 8 bits per word, 1 stop bit, no parity, light translation, no line-feeds after CRs, and all outgoing handshake lines high (DTR, RTS, and SD).

Mode: Block or Concurrent

There are two modes in which the RS232 port may be used: **Block** and **Concurrent**. Block mode is seldom used, because you can only send data to the port, not receive. The reason it was used is because the 850 would not allow disk access in concurrent mode, only block. The BB will allow you to access any SIO bus device (floppy, etc) while in block OR concurrent, so there is not a limitation.

Concurrent mode is what is used for all terminal programs. It lets you get and put data to the modem. The only disadvantage is you cannot check for a carrier (directly; you can look at a location in memory, but that is only valid for the BB). The BB has an input and an output buffer for the data. The input buffer is 192 bytes long, so data that comes in when you are not waiting for it

will be stored, and hopefully nothing will be lost. The output buffer is 32 bytes long, because a large buffer is not needed for outputting data.

The Status command is where the two modes also enter the picture. When in block mode, a status will return the state of the input handshake lines (CD, DSR, and CTS). When in concurrent mode, it returns the number of bytes present in the input and output buffers.

Opening the Port

Syntax: OPEN #Channel,Aux1,0,"R:"

Sample: OPEN #5,13,0,"R:"

This command will open an I/O control block to the RS232 port. The value specified in Aux1. A value of 4 is read only, 8 is write only, and 13 is read and write (the mode normally used). The Channel is the IOCB number, 1-7.

When OPEN is performed, the port is opened in block mode. See XIO 40 on setting concurrent mode. None of the states of the handshake lines will be modified by an open.

Closing the Port

Syntax: CLOSE #Channel

Sample: CLOSE #5

The close command will close the IOCB used by the RS232 handler. This will disable the RS232 interrupts, so any data that may come in will be lost. This needed to be done with the 850 to allow the SIO bus to be used for floppy disk access. On the BB, you may access the floppy while the RS232 port is still active, but in writing programs compatible with all interfaces, the 850's rule should be followed. The close will not affect the state of any handshake lines.

If there are any characters in the output buffer, the close will wait until they have all been sent. There will never be any 'hit' characters at the end of a close, as does the 850.

Statusing the Port

Syntax: STATUS #Channel,Junk

Sample: STATUS #5,X

The Status command places four bytes into memory locations 746-749 (\$02EA-\$02ED) with information pertinent to port operation. There are two different sets of data returned, depending on if you are in block or concurrent mode. Location 746 always returns the error status bits relating to the last byte received. These are meanings to the error bits set in location 746:

Bit	Dec	Equiv	Error
7	128	Received data framing error	
6	64	Received data byte overrun error	
5	32	Received data parity error	

When in block mode, location 747 contains the information pertaining to the input handshake lines. Locations 748 and 749 are not meaningful. These are the meanings of the values in location 747 when in block mode:

Bit	Dec	Equiv	Meaning
7	128	DSR (Data Set Ready) now on	
6	64	(history of DSR from previous status)	
5	32	CTS (Clear To Send) now on	
4	16	history of CTS from previous status)	
3	8	CD (Carrier Detect) now on	
2	4	(history of CD from previous status)	

You'll notice each bit has a 'history' bit. This history bit can be compared to the actual bit value to determine if there has been a change since the last status. Note that the history bits are all zero when the port is first opened, regardless of the state of the handshake lines.

When in concurrent mode, location 747 holds the number of characters in the input buffer, and location 749 holds the number of characters in the output buffer waiting to be sent.

Setting Concurrent Mode

Syntax: XIO 40,#Channel,0,0,"R:"

Sample: XIO 40,#5,0,0,"R:"

This will enable concurrent mode, necessary for getting any data from the RS232 port. There are no parameters needed. Once you are in concurrent mode, the only way to exit is by closing the channel.

If you have used XIO 36 to specified lines to monitor, they will be checked when this command is executed. If any line fails the check, then an error 139 will be returned.

The suggested order for accessing the RS232 port is this: first open a channel to the R: device, set the parameters needed (BAUD rate, handshake lines, etc.), do a status for a carrier check, then set concurrent mode. Now you can do the puts and gets to access the serial device.

Flushing the Output Buffer

Syntax: XIO 32,#Channel,0,0,"R:"

Sample: XIO 32,#5,0,0,"R:"

This command will simply pause until any data in the output buffer has been sent. This is not a real useful function, but is provided for 850 compatibility.

Controlling the Handshake Lines

Syntax: XIO 34,#Channel,Aux1,0,"R:"

Sample: XIO 34,#5,192,0,"R:"

This function allows you to alter the state of the outgoing handshake lines: Data Terminal Ready (DTR), Ready To Send (RTS), and allows indirect control of the transmit data line. The following are the values added to obtain a value for Aux1:

<u>Line</u>	<u>Bit</u>	<u>Dec</u>	<u>Equiv</u>	<u>Meaning</u>
DTR	7	128	Set DTR to bit 6	
	6	64	Turn DTR ON (turn off if this is 0)	
RTS	5	32	Set RTS to bit 4	
	4	16	Turn RTS ON (turn off if this is 0)	
SD	1	2	Set SD to bit 0	
	0	1	SD is normal (send BREAK if 0)	

The BB's handler treats bits 0 and 1 of this value slightly different. If bit 0 is 0 and bit 1 is 1 (that is, a value of 2 is placed in Aux1), a 500ms break signal will be sent. To be compatible with the 850, any program using this feature should set bits 0 and 1 in Aux1 and issue another XIO 34 1/2 seconds after the first is sent.

There are two bits per handshake line: the first bit tells whether we will change it or not, and the second bit is what we will change it to, but only if the first bit is set. For example, to turn DTR on, we would set bits 7 and 6, so adding 64 and 128, we get 192 (which was used in the sample). To turn DTR off, we would set bit 7, but have bit 6 clear. This would result in an Aux1 value of 128.

Setting the BAUD, Stop Bits, and Ready Monitoring

Syntax: XIO 36,#Channel,Aux1,Aux2,"R:"

Sample: XIO 36,#5,12,0,"R:"

This command controls three separate functions: setting the BAUD rate, the number of stop bits for each byte, the number of bits per byte, and identifying which handshake lines (if any) are to be checked when entering concurrent mode (with a XIO 40). The following three tables are values that are added together to produce a value for Aux1:

<u>Add</u>	<u>BAUD</u>	<u>Add</u>	<u>BAUD</u>	<u>Add</u>	<u>Word Size</u>
0	300	9	600	0	8 bits
2	50	10	1200	16	7 bits
4	75	11	1800	32	6 bits
5	110	12	2400	48	5 bits
6	134.5	13	4800		
7	150	14	9600	<u>Add</u>	<u>Stop Bits</u>
8	300	15	19200	0	1
				128	2

Note that BAUD rates 1 and 3 are not present; the BB hardware cannot produce these. Rate 15, 19,200 BAUD, is a BB exclusive: the 850 could not support anything higher than 9600 BAUD.

The following is a table of numbers that are used for the Aux2 values:

<u>Value</u>	<u>To Monitor</u>	<u>Value</u>	<u>To Monitor</u>
0	None	4	DSR
1	CD	5	DSR, CD
2	CTS	6	DSR, CTS
3	CD, CTS	7	DSR, CD, CTS

The "To Monitor" lines are the handshake lines of the RS232 port. DSR is Data Set Ready, CD is Carrier Detect, and CTS is Clear To Send. The checking is done whenever the XIO 40 (set concurrent mode) is preformed. If all the lines defined in the selected value of Aux2 are TRUE (on), then the XIO 40 will return without errors. If one or more of the lines are not set (and you specified them to be checked), then an error 139 will be generated. We've never seen this function actually used, but it is provided to insure compatibility.

Setting Translation and Parity

Syntax: XIO 38,#Channel,Aux1,Aux2,"R:"

Sample: XIO 38,#5,32,0,"R:"

This command is another multi-purpose function. It sets up the translation type, line feed options, and parity modes for the RS232 port.

There are three types of translation: None, where all incoming and outgoing data is untouched; Light, where the 8th bit is stripped and any return characters are translated; and heavy, where light translation is performed, plus any characters less than 32 will be translated to the byte specified in Aux2. In light and heavy translation, any ASCII return characters (13, or \$0D) will be translated to the ATASCII return character (155, or \$9B). The same thing, except in reverse, will happen with outgoing characters. If you specify Linefeeds appended to CRs, then a line feed character will be sent after a converted CR character has been sent. Note that no other ATASCII codes other than the CR character will be translated.

The following are the values added to generate a value for Aux1:

<u>Add</u>	<u>To Get</u>
0	Light ATASCII/ASCII translation
16	Heavy ATASCII/ASCII translation
32	No translation
0	Do not append linefeeds
64	Append linefeeds after CR (for translation)
0	No Parity
4	Odd Parity
8	Even Parity

The BB does not check the parity of incoming data; but it will send data with the correct parity.

Possible Errors from the RS232 Handler

These are the error codes that can occur when using the BB's RS232 port, and their explanations:

Error 129: IOCB Channel already open. You tried to open an IOCB that is in use by another device. Remember that a command processor DOS can use a couple IOCBs when it is executing batch files. CLOSE the channel first.

Error 130: Nonexistent device. You tried to use the RS232 port, but it was not enabled. Go into the BB menu, select Port Statuses, and enable it.

Error 131: IOCB opened for Write only. You tried to read from the port that was not opened for read and write.

Error 132: Invalid XIO. The XIO number you specified was not valid.

Error 133: IOCB Channel not open. You tried to access a device that was not opened.

Error 135: IOCB opened for Read only. You tried to write to the port that was not opened for read and write.

Error 139: Readiness check failed. The handshake lines specified in Aux2 of the XIO 36 command were not all true (on).

Error 153: Command not valid in Concurrent mode. You tried to perform a XIO command that cannot be done while in concurrent mode, or you tried to execute a XIO 40 when you're already in concurrent mode. First close the IOCB, re-open it, then do the command.

Error 154: Concurrent mode not active. You must enable concurrent mode with an XIO 40 in order to read data from the RS232 port.

Black Box Technical Information

This technical information is for those who wish to write programs that use the data used in the BB. This information need not be understood for the average user; it is for programmers.

The BB consists of several peripheral I/O chips mapped into the \$D1xx address. The BB's RAM is split up into 256 byte pages, which are mapped into a window at \$D600-\$D6FF. The high byte of the RAM address is at location \$D1BC, named RAMPAGE. RAMPAGE is normally set to \$FF; values \$FE through \$EE are used for system use, and the rest for the internal RAM printer buffer. On 8k RAM versions, RAMPAGE may only go as low as \$E0.

These are the registers of the BB hardware that may be useful:

Address Name Description

\$D1BC RAMPAGE This is the high address byte of the BB's internal RAM. Memory is mapped in at page \$D6xx. This register is read/write.

\$D1BE SWITCH bit 0=dump type switch, bit 1=HD write protect switch, bits 2-7 are dip switches 2-7. This is a read-only register. Each bit is inverted.

\$D1C0 SENSEL Bit 5=DSR, bit 6=CTS, bit 7=CD from RS232 Port. This is a read-only register.

\$D1FF PBIBANK Bits 0-3 select which bank of the BB's ROM will be mapped into the \$D800-\$DFFF memory segment.

Location \$0248 is the shadow of this write-only register.

\$D6xx BBRAMW This 256 byte window is where the BB's internal RAM is accessed. The high byte of the 64k address range is set in RAMPAGE.

There are two pages of the BB's memory that may be useful. RAM page \$FE is where all the miscellaneous working variables are stored. Here are a few locations of interest:

Address Description

\$D608-\$D610 Floppy high speed table. Each byte corresponds to drives 1-9. If they are configured as a floppy in high speed, these bytes will identify the type of high speed. \$80=standard speed, \$10=Happy warp, \$0F=XF551, \$0A=UltraSpeed, \$00=not yet determined.

\$D653 Partition Index. This byte identifies which entry (\$10-\$6F) will be used when accessing the partition list read/write routines.

\$D656-\$D667 Partition data buffer. This is the 18 byte buffer that is used to read/write to the partition list.

Page \$F5 holds the main system configuration. Here are the variables used in it:

Address Description

- Drive Configuration -

\$D600-\$D609 ASCII text of partition name. Only valid when drive is configured as a hard disk.

\$D60A Bit 7=Floppy flag. Bits 6-0 are the partition index value. The index value starts at \$10, so if this value is less than \$10, it is unknown.

\$D60B Bit 7=HD write protect flag. Bit 6=Ultra Speed flag (valid if floppy flag=1). Bit 5=HD double density mode (0 means single). Bits 2-0 are the controller index (0-7, which translates to A-H on the menu).

\$D60C-\$D60E HD starting sector number -1. Low byte first. This is the actual sector number, so when using 512 byte sectors, this number is multiplied by 2 before being displayed on the menu.

\$D60F-\$D610 HD length +1, bits 0-15. Low byte first. Note that this value does not change for 512 byte sectors.

\$D611 Bits 7-4 are the floppy number used if this drive was configured as a floppy. Bits 1-0 are bits 16-17 of the HD length +1 value.

\$D600-\$D611 The above definition was for logical drive 1. This is repeated 8 more times, for all 9 drives.

\$D612-\$D623 Configuration for D2:

\$D624-\$D635 Configuration for D3:

\$D636-\$D647 Configuration for D4:

\$D648-\$D659 Configuration for D5:

\$D65A-\$D66B Configuration for D6:

\$D66C-\$D67D Configuration for D7:

\$D67E-\$D68F Configuration for D8:

\$D690-\$D6A1 Configuration for D9:

- Controller Configuration -

\$D6AB Bits 7-5: Logical Unit Number of hard disk. Bit 4 is the SASI flag (0=SCSI). Bit 3 the 512 byte sector flag (0=256 byte sectors). Bits 2-0: SCSI bus controller ID.

\$D6AC-\$D6AD Number of Cylinders, low byte first.

\$D6AE Bits 7-4: number of heads. Bits 3-0: step rate.

\$D6AF-\$D6B0 Reduce Write Current cylinder (low byte first).

\$D6B1-\$D6B2 Write Precompensation cylinder (low byte first).

\$D6AB-\$D6B2 The above definition was for controller 1. This is repeated 7 more times for the 8 controllers possible.

\$D6B3-\$D6BA Configuration for controller 2

\$D6BB-\$D6C2 Configuration for controller 3

\$D6C3-\$D6CA Configuration for controller 4

\$D6CB-\$D6D2 Configuration for controller 5

\$D6D3-\$D6DA Configuration for controller 6

\$D6DB-\$D6E2 Configuration for controller 7

\$D6E3-\$D6EA Configuration for controller 8

- Port Statuses -

\$D6ED Bit 7: I/O sound flag. Bit 6: Enable RS232 port. Bit 5: Enable printer port. Bit 4: Send LF after CR. Bit 3: Make printer respond to all IDs. Bits 2-0: Printer ID+1 to use (if bit 3=0).

\$D6EE Bit 7: Spooler enabled. Bit 6: Spooler using XE RAM (0=spooler using internal BB RAM). Bit 5: XE RAM available. Bit 4: Internal RAM available.

\$D6EF-\$D6F1 Partition list starting sector number, low byte first. This is the actual sector number, so it is multiplied by two if the drive uses 512 byte sectors.

\$D6FE-\$D6FF Configuration check number. This 16 bit number is calculated based on the information of \$D600-\$D6FD. It is checked every time the system is reset. If the calculated number does not match the number here, the configuration from hard disk #0,0 is loaded. If that cannot be done, the config is initialized to all floppy drives.

You may make modifications to the system configuration by changing the parameters in this page, but upon a RESET or a cold start, the BB will reload the saved configuration. If you wish to make a

permanent change, you must calculate the new configuration check number, and place it in the last two bytes. This may be accomplished by the following machine language subroutine:

```
RAMPAGE = $D1BC ;identify hardware registers
BBRAMW = $D600 ;where the window is located
CFG_PAGE = $F5 ;page used for system config
CFGCHKN = $D6FE ;location of the check value
DUMMYPAGE = $FF ;number that should be placed in RAMPAGE after ;we're done so that an
inadvertent store to the ;$D6xx page doesn't wipe us out
;
SET_CHECKSUM ;Now, our subroutine
LDA #CFG_PAGE ;tell the BB we want to mess with
STA RAMPAGE ;the configuration page
LDA #0 ;initialize registers to 0
TAX
TAY
CLC
SET_C1 ADC BBRAMW,Y ;add the value of a byte,
PHA ;save for a sec
TXA
ADC #$0E ;this part does some bit
EOR BBRAMW,Y ;manipulations, so that we
TAX ;will detect any bit change,
PLA ;even if a bit was only moved.
INY
CPY #$FE ;do this for all 254 bytes
BCC SET_C1
STX CFGCHKN ;store our new value
STA CFGCHKN+1
LDA #DUMMYPAGE ;now lets set the dummy page value,
STA RAMPAGE ;so that we don't mess up the config
RTS ;and that's it!
```

This routine could not easily be accomplished using BASIC statements, but it could be embedded within a string and called by a USR statement, since it is relocatable.

The Partition list may be accessed by setting \$D653 in the miscellaneous page (RAMPAGE=\$FE) to the desired partition number + 15 (The first partition is 16, or \$10), then calling the read partition vector. When returning from the read partition subroutine, the carry flag will be set if there was an error. The Partition data buffer at \$D656 of the miscellaneous bank will contain the 18 bytes for that partition. The example below will demonstrate how you can incorporate the partition list into your programs. What it will do is take the partition number (1-96) from the A register, and put its information into logical drive 8.

```
RAMPAGE = $D1BC
CFG_PAGE = $F5
MISC_PAGE = $FE
```

```

DUMMYPAGE = $FF ;These were defined in the previous example
PBIBANK = $D1FF ;BB firmware control register
PBISHADOW = $0248 ;RAM shadow register for PBIBANK
HDCODE_BANK = $08 ;bank where hard disk drivers reside
;
SET_PART ;our subroutine. A=(1-96) which partition we
;want to place into drive 8.
CLC
ADC #$0F ;the BB numbers them $10-$6F
LDX #MISC_PAGE ;set ram window for miscellaneous
STX RAMPAGE
STA $D653 ;we needed to set RAMPAGE first, else this
;store might have ended up in the wrong bank
LDA #HDCODE_BANK ;bank for the hd low level code
STA PBISHADOW ;store in shadow register first
STA PBIBANK ;puts the BB's HD code into $D800-$DFFF
JSR $D82C ;** Address for Read Partition subroutine **
LDA #0 ;restore normal FP back
STA PBISHADOW
STA PBIBANK
BCS SET_ERROR ;carry is set if there was an error
LDA $D656 ;if bit 7 (the minus flag) is set, then
BMI SET_ERROR ;this partition is not in use
LDY #0 ;prepare to copy the bytes
SET_1 LDX #MISC_PAGE ;get one byte from the
STX RAMPAGE ;partition data buffer in the
LDA $D656,Y ;miscellaneous page
LDX #CFG_PAGE ;now put the byte in the main
STX RAMPAGE ;configuration, in the config page.
STA $D67E,Y ;$D67E is the first byte for D8:
INY
CPY #18 ;copy 18 bytes this way
BCC SET_1
CLC ;we've completed with success!
BCC SET_2 ;skip the set carry
SET_ERROR SEC ;set the carry upon an error
SET_2 LDA #DUMMYPAGE ;return with the dummy page
STA RAMPAGE
RTS ;and return

```

This routine will return with the carry flag set if there was an error (no partition list defined or bad sector), or clear if everything was ok.

Random Notes

The BB provides high speed capabilities with some drives. When you format a disk on a modified 1050 or XF551 (not happy drives), it will write out the sectors in a special high speed format. If you read a disk with the high speed format on a standard drive, it will be slower than normal. If you want to format a disk with the standard format, you can either disable the high speed (with the

menu, or holding SELECT and OPTION while the drive is being accessed), or hold the SHIFT key down when the format command is given.

There may be a few game programs that will not allow you to enter the BB's config menu. They do so because they load in their own special operating system. Simply press RESET to re-enable the standard OS.

The BB should be used with XL/XE compatible operating systems only. OS chips based on the old 400/800 operating system will function, but will render the BB useless because they have no drivers for parallel bus devices. The same applies for disk based 'translator' programs.

The BB's physical format program provides allows you to change the format interleave factor. CSS has run tests on all controllers listed and determined the optimum interleave. This number should only be changed if you need to slow down the operation of your hard disk.

CSS would like to thank Jeff Williams for his involvement in the BB firmware development, Ed Bachman for his directory format program, and all those who offered suggestions and comments with regards to BB development. It's great to see 8-bit programmers come together to make a product like this possible.

We have set up a support BBS for those of you who have modems, to provide the latest information and make program updates available. There are no connect charges for this service, just the cost of the call. The BBS is online 24 hours a day, running at 300 and 1200 BAUD. ATASCII, ASCII, and VT-52 translations are available. The number is (585) 247-7157. An interesting note: because we are using multiple lines, you will see another of our fine products, **THE MULTIPLEXER**, in operation!

Appendix A: Compatible Controllers

The BB's SASI/SCSI port should interface to any SASI or SCSI device. However, since most controllers have their own special way of formatting drives, there needs to be separate formatter programs written for each one.

We have combined the format routines for all the controllers we have seen into the HDFMT.COM program. If you have a controller that is not listed, give us a call. We may be able to write a format program for your controller, provided that you have the technical manual for it. Any new format utility programs will be available on our CSS Support BBS.

These are the controllers known to work with the BB:

Adaptec 4000A and 4070 (SCSI) Seagate embedded SCSI drives **
Konan DJ-210 (SASI) SMS 5001 (SASI or SCSI)*
OMTI 3520 and 3527 (SCSI) ** Western Digital 1002-SHD (SASI)
Seagate ST9000 (SASI) (WD clone) Xebec S1410 and S1410A (SASI)

* - The SMS controller needs to be set to SCSI for 256 byte sectors, and to SASI for 512 byte sectors. This is done automatically in the format program; you need only select the sector size.

** - The OMTI controllers can only be used in 512 byte mode.

*** - Some Seagate embedded drives can only be used in 512 byte sector mode. Seagate embedded drives are characterized with a suffix of N.

Appendix B: Hard Drive Data

(The data listed here was so old that it is not relevant. The BB's format program now auto-detects SCSI drive settings, so no real configuration is necessary unless you are using an older drive with a separate controller.)

Appendix C: Files on the BB DISK

The following is a list of the programs on the BB distribution disk, and a brief description of their purpose:

DOS.SYS and DUP.SYS: These are the DOS files to MYDOS version 4.50. This is a freeware DOS by Charles Marslett of Wordmark Systems that can handle hard disks with ease. It is simple to use, yet powerful in features.

VTOCFIX.COM and VTOCFIX.DOC: The VTOCFIX is a MYDOS utility for fixing up any problems that may occur. The .DOC file is the documentation for this program, and may be printed out by copying to P:.

HDFMT.COM: This is the BB physical format program, used to format hard drives.

FMTDIR.COM: This program is used to set your hard disk partitions up so that DOS can use them. It also maps out any bad sectors that were not able to be mapped out at the controller level.

USPRAM9.COM: This is a utility program for those who use the US+ OS (by CSS). The US+ defaults the RAMdisk to be drive 4. This file will set the US+'s RAMdisk drive number to 9, so that it does not get in the way of the BB (if you wanted to use drive 4 as a hard disk). To install this program, first copy it to your drive 1. Now copy any AUTORUN.SYS file onto this WITH APPEND (use the /A modifier). Rename or remove your old AUTORUN.SYS file, and rename this new file to AUTORUN.SYS.

PARK.COM: This program will position the heads of each connected hard disk to the inner cylinder. This is a precautionary step that should be done prior to powering down your hard disks (if they do not have an auto-park feature). It also reports the last sector available on that drive.