SBCOMM

SuperBox Communications Program

USER'S GUIDE

version 1.0

Visual Database Systems

TABLE OF CONTENTS

SBCOMM - Introduction

SuperBox COMMunications Program
The following is an introduction to the Visual Database Systems **SBCOMM** SuperBox **COMM**unications programming tool. Installation and use of the various features are covered. For a more extensive tutorial on programming the SuperBox please read the separate SuperBox Programming Manual.

What is SBCOMM?

SBCOMM is a DOS utility program that makes it easy for PC users to create programs on the SuperBox. It has the following capabilities:

1. Very easy to set up communications program. The only setting is selection of COM1: or COM2: and that is only necessary if your computer has two ports installed. Data communication parameters are preset for the SuperBox. You can quickly establish communications between your computer and the SuperBox and also use PASS-THRU communications to the player.

2. Extended buffer. Communication programs on many slower PCs can not handle the extended EPROM data dump when it comes from the SuperBox. Because of SBCOMM's extended buffer size this is not a problem, even with old 8086 based machines.

3. Easy utilities built-in for uploading and downloading program files between the PC and SuperBox. Automatic support of Xon/Xoff protocol for full speed programming.

4. Complete on-line help reference for SuperBox programming commands and basic Pioneer player commands.

SBCOMM can be used at the DOS prompt or in a DOS window in MS Windows.

SBCOMM is used with your choice of editors. We use Borland's original Sidekick on smaller PC's running DOS. With faster Windows machines, we use either the notepad editor shipped with Windows or Brief running in a DOS window. Any ASCII mode editor, including many word processors can be used.

While it is possible to program a SuperBox with any computer or dumb terminal with RS232 communications capabilities, it is much simpler to use SBCOMM for your player communications.

SBCOMM Kit Contents

SBCOMM is shipped on a 3-1/2" floppy diskette with the following files:

- **SBCOMM.EXE** -- The SBCOMM executable program

- **SB42HELP.TXT**
- **SB20HELP.TXT**
- **SB_HELP.TXT** -- SBCOMM on-line help files

- **SB8kDEMO.SBX** -- A sample program. Other samples may also be provided.

- **SB2_IO.TXT** -- A text file showing the I/O pinouts for the SuperBox connectors.

SBCOMM is also shipped with the following accessories:

- **Programming cable** - RJ-11 to DB25 female (or optionally a DB9 female).

- **This User's Guide.**

- **The SuperBox II Programing Manual.**

Installing SBCOMM

The minimum installation is to copy the SBCOMM.EXE file to a directory in your search path.

We recommend the following somewhat more complex installation:

1. Copy the SBCOMM.EXE file to your \DOS directory or some other directory in your search path where you story your executable programs.

2. Make a directory \SBCOMM and copy the remaining files into this directory.

When programming, change directory to the SBCOMM directory and start the SBCOMM program from there. Also save your programming files in the SBCOMM directory.

```
C:> CD \SBCOMM
C:> SBCOMM
```

IMPORTANT: To use the SBCOMM Help function, the SBCOMM help files must be in the current directory.

File Naming Conventions

SBCOMM uses the following file suffixes when saving files:

.SBX - A SuperBox text format program file
.SBE - A SuperBox EPROM dump file
.SBC - A SuperBox text format capture file

Operation of SBCOMM

Setting up.

1. Install SBCOMM on your computer as described above.

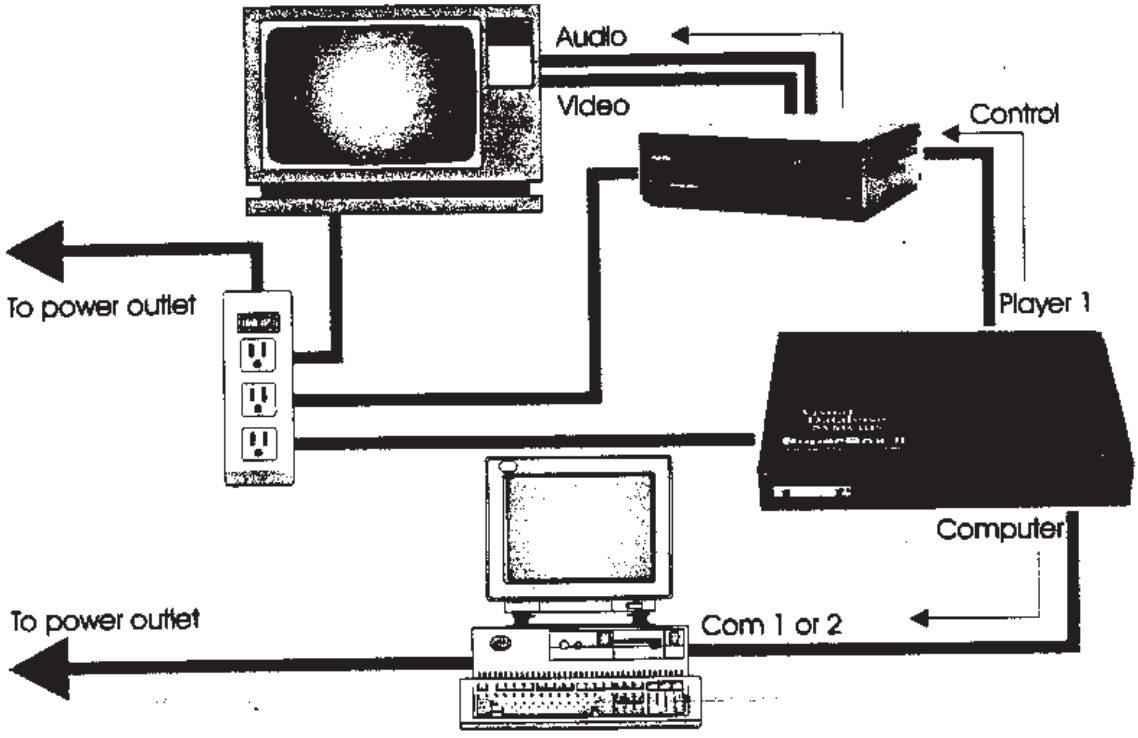
2. Connect the programming cable from your PC's COM1 or COM2 port and the SuperBox's front panel Computer port.

3. Connect the SuperBox to the Laserdisc player.
   This example is assuming a Pioneer player with one of the following model numbers: LD-V2200/2200/4400/8000 or CLD-V2400/2600.

4. It is convenient to have the Laserdisc player and SuperBox power supply plugged into a switched outlet strip for testing.

5. Make the necessary audio and video connections between the Laserdisc player and your monitor.

Your setup should look like the following diagram:



6. Set the player's communication parameters to match the SuperBox.
   - For LD-V4200 - open front panel access door and slide all switches UP
   - For LD-V2200 and CLD-V2400/2600 - on back panel slide all switches UP
   - For LD-V4400/8000 - Hold display button in while turning player on. Press the open button to restore factory defaults and then set for 4800 baud, all other settings are to defaults. See the SuperBox manual or the Pioneer player instructions for detailed instructions.

7. Start the monitor and player and get a picture playing on the screen.

Programming with SBCOMM

The Sample Program.

We will use a DOS based PC as an example. The program that we are going to be creating is for the following:

1. Set the player to search with a black background.
2. Play an attract loop and allow the user input to interrupt from one input line.
3. Play a video segment. Do not allow this to be interrupted.
4. Return to the attract loop and repeat steps 2 - 4 forever or until the power is shut down.

We will use the PASS-THRU function of the SuperBox to search for the desired frame numbers.

Typographical Conventions

When you are to type something on your PC it is shown it like this:

<Esc>B1?<CR>

which means type the **escape** key, "**Shift B**", "**1**", "**?**" and press the **enter** or return key.

Non-bold face type is used to show what the computer responds:

```
BTN 1:
100SE
[
200PL
[
Ok
```

This is a zero: **0**
This is the letter "O": **O**

Starting SBCOMM.

From your computer, change the directory to SBCOMM and boot the program:

```
CD \SBCOMM<CR>
SBCOMM<CR>
```

SBCOMM starts and displays its command screen. The following should be visible:

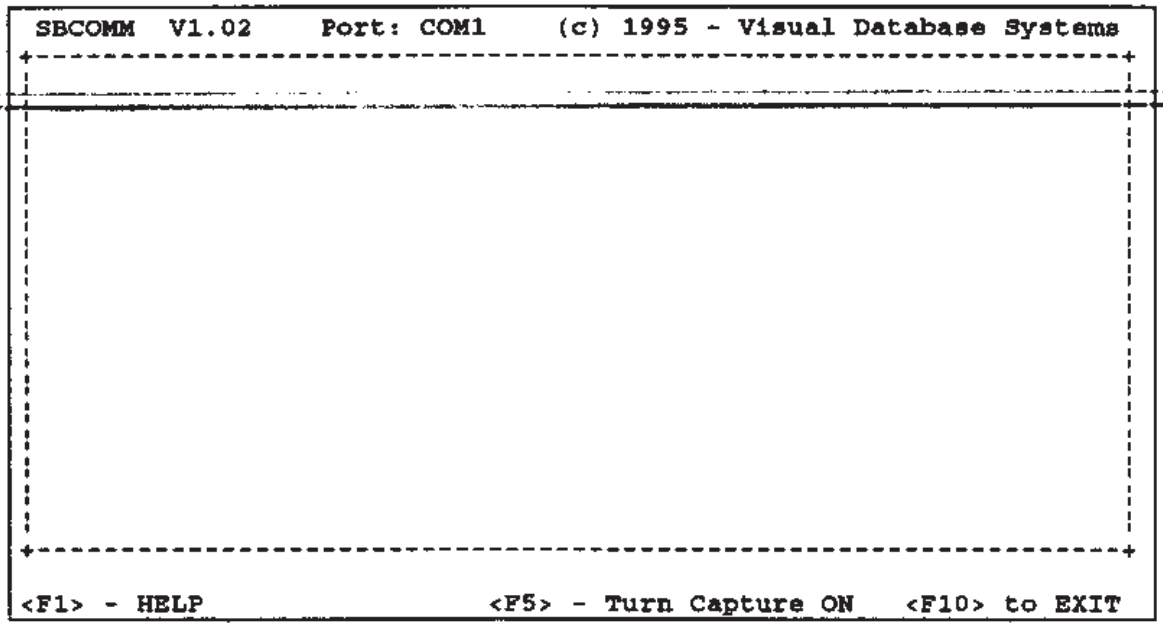
```
SBCOMM   V1.02   Port: COM1    (c) 1995 -  Visual Database Systems



                        OPTIONS

        1)  Communicate with the SuperBox
        2)  DownLoad a Text File to the SuperBox
        3)  DownLoad an EEPROM Dump File to the SuperBox
        4)  UpLoad SuperBox Button Contents to a Text File
        5)  UpLoad SuperBox EEPROM Contents to a File
        6)  Change I/O Port From COM1 to COM2

        0)  Exit to DOS


              Response : (1,2,3,4,5,6 or 0) :
```

1. Select COM port

Press **6** to change the COM port to COM2 if the SuperBox is connected to COM2 (If you only have one COM port, selection 6 is not displayed.) If the SuperBox is connected to COM1 you don't need to do anything. If you don't know which COM port you are connected to, you can toggle back and forth using selection 6 at the command screen. The color of the bar at the top of the screen changes between blue for COM1 and green for COM2 and the text for selection 6 changes as you toggle the port.

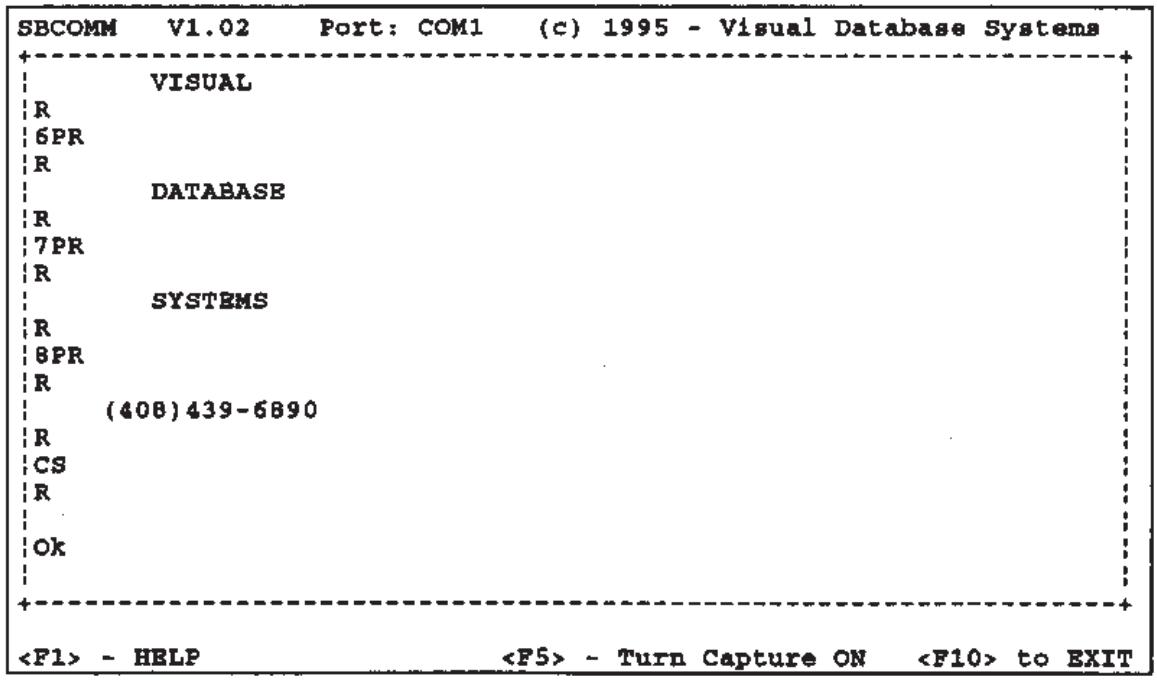
2. Communicate with SuperBox

Select **1** - Communicate with the SuperBox by pressing the **1** key on your PC. The screen will change and look like this:



3. Turn the SuperBox on

If it is already on, turn it off and back on. You should see the following displayed on your screen:



In addition to the display on your screen, you should see the video monitor display PLAYER SPINUP for a few seconds, additional startup information and then clear the screen. Additional things may happen depending on the previous programming of the SuperBox.
If your screen does not show the SuperBox startup, press the **<F10>** then **6** to toggle ports then start again at #3 above.

CONGRATULATIONS -
You have now established communications with the player, computer and SuperBox. The rest is easy! (just a small matter of programming...)

4. Enter PASS-THRU mode and communicate with the player

<Esc>P<CR>

NOTE: The commands to the SuperBox must be typed in UPPER CASE or they will be ignored.

You will see:

```
PASS-THRU MODE
```

Now you are talking directly to the player. You can use this to test any of the player commands and see what happens. We are going to use it to display the frame numbers and search around the disc to select an attract loop and program.

To turn the frame display on:

1DS3RA<CR>

A "R" comes back from the player when the command has been completed. This is usually immediately after a command is entered but for search or autostop playback commands it may take from a few seconds to many minutes.

Now start the disc playing:

PL<CR>

To stop the disc give the still command:

ST<CR>

Also try these additional player commands:

SF<CR>	Step Forward
SR<CR>	Step Reverse
MF<CR>	Multi-speed forward
10SP<CR>	Set speed to 10 fields/second (1/6 speed slow)
120SP<CR>	Set speed to 120 fields/second (double speed)
MR<CR>	Multi-speed reverse

Using these player commands, identify starting and stopping frame numbers for our test program. (HINT: make them short for our example.)

When you finish sending commands to the player return the player's display setting to its starting value (the player command 0DS).

0DS<CR>

end the PASS-THRU mode:

<Esc>

and see displayed:

COMMAND MODE

Lets assume the following frame numbers for finishing this example, you may substitute others if you wish.

	Starting Frame #	Ending Frame #
Attract Loop	123	456
Main Video	9876	10012

5. Initialize the SuperBox

The first step is to prepare the box by enabling the EEPROM write line. On a newer box this is accessible by removing the Program Memory Access panel. Older boxes require the removal of the entire top cover. Short the jumper block marked "EEPROM Write Disable" by plugging the jumper between the two pins. This enables the EEPROM to be modified. Unplugging and rotating this jumper when we are finished will make the program EEPROM write only until re-jumpered. This protects against power-up glitches and any other accidental erasure.

Since we don't know what state the SuperBox was last configured to, we will start by performing a complete re-initialization.

<Esc>CINIT<CR>

Now power-down the box and start it up again. This resets everything to factory defaults. When the computer display says "Ok", you are ready to continue.

Next wipe the memory clean.

<Esc>CALL<CR>

Again wait for the "Ok".

Program layout

Let's review the program outline mentioned at the beginning of this section.

The program that we are going to be creating is for the following:

1. Set the player to search with a black background.
2. Play an attract loop and allow the user input to interrupt from one input line.
3. Play a video segment. Do not allow this to be interrupted.
4. Return to the attract loop and repeat steps 2 - 4 forever or until the power is shut down.

We will now program the first event, called a button in SuperBox programming terminology. When the SuperBox is first powered up, it does the following:

a) Looks to see if a player is connected and turned on.
b) Displays "Starting player" on the Laserdisc player character generator.
c) Sends the player a spin-up command.
d) Displays the VDbS name, address and phone number for a few seconds.
e) Clears the screen sets the player's frame number and display registers to their defaults (RA, RC and DS registers to 0) and executes button 0.

Button 0 (zero) is used to send any user specified startup commands and begin the user program. We want to set the search background to black as a default and start the attract loop. We will set the background in button 0 and do the attract loop as button 2. Our program will use button 1 as the user input for starting the main program. Buttons 3 through 15 will not be used.

1. Programming the first Button (Button Zero)

Type

```
<Esc>B0=<CR>
```

You will be prompted by a "-" to enter a line of program. Each time you enter a line, you will be prompted again until you enter a blank line to finish programming the button. You can backspace to correct errors on the line that you are entering but you can not go back to a previous line. If you have already pressed enter and made a mistake, just press enter again, end the button and start over again.

Now type the program in. (You type what is shown after the "-" prompt.)

```
- 16RC<CR>
- [<CR>
- >2<CR>
- <CR>
```

The SuperBox returns an "Ok".

Now examine the contents of button 0:

```
<Esc>B0?<CR>
BTN 0:
16RC
[
>2

BYTES USED : 9 / 511
Ok
```

If the button looks like this you are ready to continue, otherwise start again by typing **<Esc>B0=<CR>** and re-entering the button's contents until it is correct.

What this does: Button 0 sends the player command "**16RC**" which sets the player's C register to use a black background while searching. The "**[**" is a SuperBox function to wait until the player responds with an "R" before continuing. This is used to synchronize the player and SuperBox timing as commands are given that take some time to execute. A "**[**" is used after every player command. The "**>2**" is a SuperBox function to branch to button 2 and continue execution. While a button is executing, the user inputs are disabled unless specifically enabled by the program.

2. Programming the Attract Loop

Enter Button 2 as follows:

```
<Esc>B2=<CR>
- FR123SE<CR>
- [<CR>
- ;,456PL<CR>
- [<CR>
- ;,>2<CR>
- <CR>

Ok
```

And check it for mistakes:

```
<Esc>B2?<CR>
BTN 2:
FR123SE
[
;,456PL
[
:,>2

BYTES USED : 23 / 511
Ok
```

What this does: Button 2 sends the player command "**FR123SE**" which sets the frame mode and searches for frame #123. The "**[**" tells the SuperBox to suspend execution until the player has completed the search and signaled with an "R" response. Next the user inputs are enabled by "**;,**" and the player is commanded to begin playing and stop on frame 456. The next "**[**" causes the SuperBox to wait until the player has stopped on frame #456. "**:,**" causes the user inputs to be disabled while the next command is sent and the "**>2**" says go back to the beginning of button 2 and do it all over again. The player will continue this until it is interrupted by a user input.

3. The Main Video program

Enter:

```
<Esc>B1=<CR>
- FR9876SE<CR>
- [<CR>
- 10012PL<CR>
- [<CR>
- >2<CR>
- <CR>

Ok
```

and check:

```
<Esc>B1?<CR>
BTN 1:
FR9876SE
[
10012PL
[
>2
```

```
BYTES USED : 22 / 511
Ok
```

What this does: Button 1 sends the player a command to search to frame 9876, wait for response, and play to frame 10012, waiting for a response and then branch to Button 2 (the attract loop).

That's it! You have now completed your first SuperBox program. Now we will test it.

Testing the Program

From the PC you can force the SuperBox program to begin execution by typing

<Esc>0<CR>

This causes execution to begin at button 0 then branches to button 2 and will loop forever. You can interrupt this from the PC by typing

<Esc><Esc>1<CR>

This causes the main video to play then return to the attract loop.

1. Attaching the User Input

If you specified your input requirements when your SuperBox was ordered, it was shipped with an input cable and attached buttons or a cable stub with the inputs ready for attachment to your buttons / cable / etc.

The cable usually has an arrow embossed in the plastic connector or a polarizing key. This indicates pin 1 on the connector and is plugged into the right side of the connector, near the polarization dot painted on the chassis. If the arrow is not visible, pin one is on the side with the red stripe on the ribbon cable or in the case of a cable with only a few wires, it is usually the side that the wires are on. (See Button 1 Wiring Diagram)

Plug the cable into the User Input connector on the rear of the SuperBox.

If the SuperBox is already running, touch the ground (pin 1 or 2) and button 1 (pin 4) wires together for a moment. The attract loop should end and the main video begin. While the main video is playing, touch the wires together again and nothing should happen. Try this a few more times and then again when the attract loop returns.

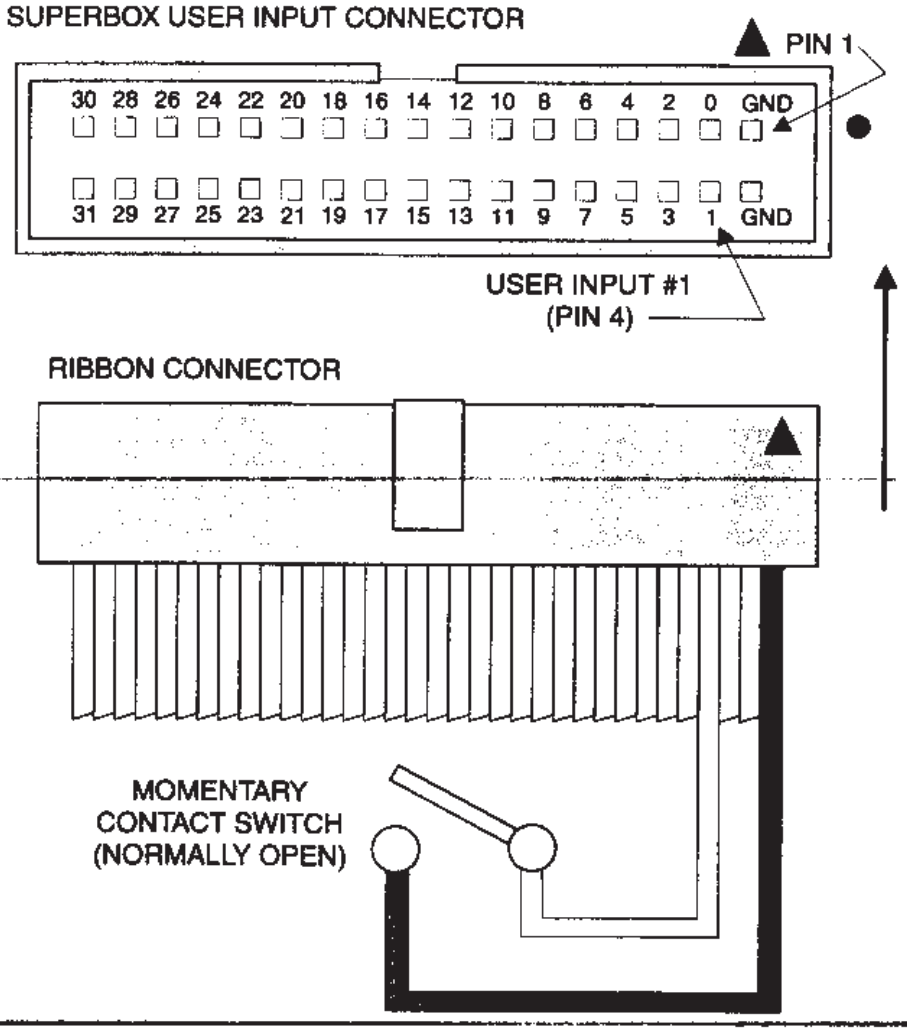
Testing User Input with PASS-THRU Mode

Using SBCOMM's mode 1 - Communicate with the SuperBox, command the SuperBox to enter PASS-THRU mode by typing **<Esc>P<CR>.**

Now press an input button or simulate the button closure by shorting from ground to one of the inputs on the User Input connector. You will see the number of the closed contact displayed on the monitor. Because you are in PASS-THRU mode no other action is taken.

Button 1 Wiring Diagram

The diagram below illustrates a simple User Input connection for Button 1. Ground (GND) is available at both pins 1 and 2 on the SuperBox User Input connector. Button 1 is located at pin 4. The corresponding ribbon connector shows pin 1 (typically indicated by a red stripe on the ribbon cable) and pin 4 in a normally open circuit. Momentary shorting of these two contacts will trigger Button 1.



2. Saving the Program

To save a copy of this program to your computer, interrupt the program that is running and wait until it has reached it's autostop or go into PASS-THRU and stop the player. (This is so that the player does not send an extraneous "R" back in the middle of our download.)

```
<Esc><Esc>P<CR>
PASS-THRU.
ST<CR>
R
<Esc>

COMMAND MODE
Ok
```

Now go to the SBCOMM command mode by pressing **<F10>** .
Next select **4** - Upload SuperBox Button Contents to a Text File.

Your screen will look like similar to this:

```
SBCOMM   V1.02    Port: COM1   (c) 1995 - Visual Database Systems
          UpLoad SuperBox Button Contents to a Text File

 Directory File Spec: *.SBX

    No Files Found.




 Default Directory: C:\WIN3\
  Default File Ext: ".SBX"

     Enter File Name :
```

Press return to copy the contents of all buttons and give the program a name when prompted. Call this program **TEST1**. SBCOMM will save the button contents to your computer.

You will see the contents of the SuperBox listed as it is saved to the PC's disc drive.

Saved File Format

The file is automatically named with a .SBX suffix and saved on the default drive and directory unless you have specified differently.

The file consists of a first line that tells the SuperBox's button number setting (the **<Esc>Nn<CR>** command.) in the form \Nn where n is 0,1,2,3...

Each button's content's are next listed beginning with a line of the form
```
BTN #:
```
where the # is the button number.

After the button's content's are listed, a blank line is appended and then the button's size is listed:
```
BYTES USED : 55 / 255
```

Two more blank lines are added and then the next button is listed.

At the end of the file an Ok is added.

The file does not have to contain all of the buttons in the program. This is often useful in editing a longer program. Only the buttons contained in the saved file are changed when the program is again downloaded.

3. Ending SBCOMM

End the SBCOMM program by pressing **<F10>** and then selecting **0**. You will now be back at the DOS prompt.

Editing your new program

The program that you have created can now be edited using an ASCII mode text editor. Your computer probably has one named EDIT under DOS or NOTEPAD under Windows. Other editors and even word processing programs can be used but you must use them in the raw ASCII format mode.

Open the saved file **TEST1.SBX** using your choice of editor. The file should look like this:

```
\N0


BTN 0:
16RC
[
>2

BYTES USED : 9 / 511


BTN 1:
FR9876SE
[
10012PL
[
>2

BYTES USED : 22 / 511


BTN 2:
FR123SE
[
;,456PL
[
:,>2

BYTES USED : 23 / 511
Ok
```

Now change the ending frame number for the main video loop from 10012 to 10123. Save this file as TEST2.SBX in the \SBCOMM directory.

1. Loading a program from the computer to the SuperBox

Restart SBCOMM and from the main screen select 2 - Download a Text File to the SuperBox.
You will see a screen like this:

```
SBCOMM   V1.02    Port: COM1   (c) 1995 - Visual Database Systems

               DownLoad a Text File to the SuperBox


 Directory File Spec: *.SBX

    TEST1.SBX



 Default Directory: C:\WIN3\
  Default File Ext: ".SBX"

     Enter File Name :


 <F1> - Directory                                    <F10> to EXIT
```

Now type **TEST2**<CR>

You will see the SBCOMM program listing the contents of TEST2.SBX as it is downloaded to the SuperBox. When it is complete, you are returned to SBCOMM's mode 1 - Communicate with the SuperBox and ready to continue. The screen should look similar to this:

```
  SBCOMM   V1.02   Port: COM1   (c) 1995 - Visual Database Systems
+-------------------------------------------------------------------+
|                                                                   |
|- FR9876SE                                                         |
|- [                                                                |
|- 10012PL                                                          |
|- [                                                                |
|- >2                                                               |
|-                                                                  |
|Ok                                                                 |
|                                                                   |
|B2=                                                                |
|                                                                   |
|- FR123SE                                                          |
|- [                                                                |
|- ;,456PL                                                          |
|- [                                                                |
|- :,>2                                                             |
|-                                                                  |
|Ok                                                                 |
|                                                                   |
+-------------------------------------------------------------------+
 DownLoad Complete
<F1> - HELP                  <F5> - Turn Capture ON   <F10> to EXIT
```

Download file format

The requirements for a .SBX file that is to be downloaded are somewhat simpler than exactly following the format of a file saved by SBCOMM. Here are the requirements:

Anything in the file before the keyword `BTN n:` is ignored. (n is 0,1,2,3...) This allows comments to be added before and between buttons. the button's size is treated as a comment at load time.
A button must begin with

```
BTN n:
```

on a single line. n is 0, 1, 2 ... to maximum number of buttons set by the <Esc>Nn<CR> command.
No blank lines are allowed within a button.
One or more blank line(s) are used at the end of each button to signal the button's end.
Comments are allowed between buttons. They are discarded when uploaded so if you use comments, you should only upload your edited file and not plan on downloading, editing and having your comments retained.
At least one blank line must be given at the end of the file.
Buttons need not be in order in the download file.
If a button is listed twice, the second occurrence is what will remain when the download is completed.

2. Check your Edited Button contents

List the contents of the edited button to see if the new contents have been downloaded.

```
<Esc>B1?<CR>
BTN 1:
FR9876SE
[
10123SE
[
>2
Ok
```

That is all there is to doing the most common operations of programming a SuperBox.

3. Write protecting the SuperBox

Disable the EEPROM write line. This is the reverse of the procedure which you performed at the beginning of this tutorial. Access the "EEPROM Write Disable" jumper by removing the Program Memory Access panel. Older boxes require the removal of the entire top cover. Open the jumper block marked "EEPROM Write Disable" by plugging the jumper on only one of the two pins. This prevents the EEPROM from being modified until it is re-enabled. This protects against power-up glitches and any other accidental erasure. when you have finished with your programming, you should replace the panel or cover.

4. Saving the SuperBox Contents as an EPROM dump format

If you are going to create an EPROM from the SuperBox program that you have currently loaded in the EEPROM you simply enter mode 5 of SBCOMM and save the file. Enter mode 5 by pressing **<F10>** then **5** --Upload SuperBox EEPROM Contents to a File. You will be prompted to enter the name of a file to save the data to and then the dump will occur. This will take a couple of minutes.

The resulting file will be given a .SBX suffix. For easy file housekeeping we recommend that you use the same name for the .SBX and matching .SBE files.

Making an EPROM

If you are distributing multiple copies of a program or creating a program that is to be used on a SuperBox III, you should use an EPROM for the copies that will go to the field. EPROMs are less expensive than EEPROMs and can not be modified as easily.

To make an EPROM you need a binary image or hex format data file of the desired contents. The SuperBox will produce this data and it can be saved using SBCOMM's mode 5. The .SBX file that is produced is an Intel hex format image of the memory and can be used by any EPROM writer. This file can also be used by the appropriate software to create a binary image of the memory data if that is necessary.

5. Restoring EPROM data to a SuperBox

It is possible to reload EPROM data that has been saved in the .SBX (Intel hex) format by using SBCOMM's mode 3 -- Download and EEPROM Dump File -- in a manner similar to that used to upload the text contents. This mode is really only used to verify existing .SBX files.

6. Capture Mode

When debugging lengthy programs, the SBCOMM utility offers a convenient capture utility. From SBCOMM mode 1 -- Communicate with the SuperBox -- screen, press **<F5>** and you will be prompted to enter the name of a file to save the forthcoming capture to. Enter a name, and a new file with a .SBC suffix will be opened and all communications from the SuperBox are written until the **<F5>** key is again pressed to close the file. This file can be examined by your editor to see exactly what is occurring as the SuperBox program is run.