The PHANTOM

INSTRUCTIONS
Version 0.6

PLEASE READ THIS
BOOKLET BEFORE
USING THE PHANTOM

INTRODUCTION:

Thank-you for buying (or building your very own) the NEW! Phantom. The parallel disk turbo system every 1541 wants but nobody's been able to get since 1987!

- Expert Compatible – Program the Expert in less than 2 seconds
- Speeds up ALL drive functions
- Full error checking retained – essential for reliability. Other systems sacrifice this crucial function to achieve speed increases but are in fact NO FASTER, only LESS RELIABLE.
- 8 function key commands for major functions.
- 60 additional commands including
    - File lock and unlock
    - Write protect ignore, no need to notch disks
    - Set device number
    - Screen on/off
    - GO1541 – reverts to standard 1541
    - Many more useful commands aid compatibility (disable function keys, disable extra ram etc.)
    - Drive monitor – commands include disassemble, fill, compare, hunt, assemble, execute etc etc.
- Compatible with most commercial software.
- Reliable – does not corrupt disks unlike rival products.
- Switchable kernel replacement included for 64 or 128 (in 64 mode)
- British designed and manufactured, available exclusively from Trilogic
- Upgradable – DOS, Kernal & Copier upgrades will be available soon to give extra features for a nominal sum.

- The LOAD command can load files under the I/O area, so theoretically you can load 248 blocks (from the start of basic at $0801) or even 252 block files (starting from the tape buffer). This has never been seen in any other speeder ever created for the Commodore.

The Phantom consists of these components.

1. Phantom Drive Board
2. Phantom Drive Board Sync Cable
3. Phantom Kernal Replacement with Switch
4. Parallel Cable

Please read/reference the installation section of this manual for correct installation of your new Phantom. Trilogic cannot be held responsible for any damage caused due to incorrect installation.

Compatibility:

The Trilogic Phantom board is compatible with all “long board” Commodore 1541 disk drives (110v & 220v models). These are the drives manufactured around 1987 or prior. You can tell the drive is correct as it will look like the following picture.

<<< INSERT PHOTO HERE >>>

In terms of Commodore 64 compatibility, so long as the computer has a socketed kernal ROM, it can be deemed compatible. If your computer doesn’t have a socketed kernal ROM, then you’re going to have to remove the ROM and replace it with a socket.

If this is beyond your capability, seek competent professional assistance.

Keyboard Shortcuts:

The following are various keyboard shortcuts incorporated into the Phantom hardware to assist in making your computing experience more productive.

<CTRL-RESTORE>:

Resets the screen colors to the original factory default state of dark blue screen, light blue border and light blue text. Does not affect the contents of memory.

<RUNSTOP-RESTORE>:

Resets the screen colors to the PHANTOM default state of dark grey screen, light grey border and yellow text. Does not affect the contents of memory.

<RUNSTOP-(← key)>:

Resets the computer as though you pressed a hardware reset button or issued an SYS64738 reset command.

<CommodoreKey-RUNSTOP>:

Loads the first file on your diskette into basic memory space. The command that's issued is LOAD"0:*",8

<SHIFT-RUNSTOP>:

Loads the first file on your diskette into the memory space where the program was originally saved from. The command that's issued is LOAD"0:*",8,1

Function Keys:

f1 - @8$:*
f2 - @9$:*

f3 - LOAD
f4 - SAVE"
f5 - RUN:
f6 - LIST
f7 - @
f8 - sys61916

Built-In Functions:

Custom Directory Function: £

This feature allows you to create a customized directory of your diskette. To further explain I will illustrate with an example.

As we all know, it's possible to have a LOT of different programs on a single side of a diskette given it's large capacity of 170KB of storage space. And each of these programs can be made up of many different files. As such, it can be difficult to quickly determine what exactly is on the disk. Well, the Phantom to the rescue.

Let's assume that you have the following programs on your disk.

1 Jumpman
2 Gyruss
3 Ms.Pacman
4 Ft. Apocalypse
5 Dig Dug
6 Drol
7 BCs Quest for Tires

So, that's a lot of files. To make a custom directory, you've just seen the first step. Enter your list of programs that you'd find on your disk as shown above. It looks like a little program with line numbers right? Good.

Now with your diskette in the drive, and the write protect open, type the following:

SAVE “£”,8 (assuming your drive is 8)

That’s it.

Now to use the custom directory, type the following:

$£

Instead of displaying the actual directory, it will show you that list you just created. And the beauty of it is that you can put anything in that list you want to make it as meaningful as you’d like!

Interactive Editor Function:

Very little at this point is known about this function. I will highlight what is currently known, and update as more information becomes available.

@U0S<FILENAME>

This command opens the “FILENAME” for write output, and creates that file as a PRG file on the floppy disk. When this command is issued, the drive starts spinning, and the FILENAME <prg extension> is created on the diskette and opened for writing.

@U0L<FILENAME>

This command is presumably to load the above noted “FILENAME” and display it on the screen.

*** more testing and investigation is required for this command.

Commands:

@ : Command Character:

This is the character that prefixes all DOS commands. It is interchangeable with the **>** command.

> : Command Character:

This is the character that prefixes all DOS commands. It is interchangeable with the **@** command.

@& : Utility Loader:

The utility loader is the command which will load a USR file from disk into disk drive memory where it will then execute. The filename must be of a USR type, and the filename must be prefixed with the & character.

It should further be noted that when using this command, the <filename> you enter in the syntax below must have the & omitted.

Syntax/variants:

@&<filename>

The syntax assumes you have a file on floppy disk of a USR format, prefixed with the & character.

@£ : <<< unknown >>>:

This command generates 62, file not found, 00, 00 error.

Syntax/variants:

Have yet to determine the syntax for this.

@= : System Status:

This command displays a list of current option status' as well as the drive rom identification. This may also do a soft reset on the drive at the same time (unconfirmed)

Syntax/variants:

Displays the following information.

```
73,8K PHANTOM DOS V1.1E,08,08
90, RAM ENABLED,00,00
91,TURBO ENABLED,00,00
92,SCREEN BLANKING ENABLED,00,00
98,BUMPS ENABLED,00,00
99,WRITE PROTECT ENABLED,00,00
```

@$: Directory:

This command will display the contents of the directory on the diskette. The <Commodore-key> will pause the directory listing, and the RUNSTOP key will stop the directory listing. These are ideally suited for when you're looking at a very long directory or you want to stop part way through when you've found what you've been looking for.

Syntax/variants:

```
@$ (directory of drive 0 of current device)
@$$ (shows directory header & blocks free only)
@0:$ (directory of drive 0 of current device)
@0:$pa (only shows files starting with "pa")
```

@1 : Enable TAPE Support:

I'm listing this separately to bring it to your attention that this is how you enable TAPE support. Normally, when powered up, TAPE support is not available, as the default commands are focused directly to the disk drive. By issuing this command you are redirecting focus to the TAPE drive. All subsequent I/O commands that would normally be directed to the drive now go to TAPE.

Syntax:

@1

Functional Example:

Once the command is issued, LOAD or SAVE will access the plugged in datasette recorder.

@n : Directing DOS commands to a specific Drive number:

In order to "focus" your machine on a specific drive number, you can issue the @<number> command. This will direct focus of all future DOS commands for your computing session or until you change it to a specific drive number.

Syntax:

@<drive number>

Functional Example:

@8 – Will direct all DOS commands to drive 8
@9 – Will direct all DOS commands to drive 9
@10 – Will direct all DOS commands to drive 10

@An : Soft Re-Numbering your Drive:

The disk drive, by default from the factory, is device #8. Unless you've physically changed this on the 1541 logic board, your drive will have that drive number assigned to it when you power it on. Soft renumbering allows you to change the device number of your drive without any physical changes.

Syntax:

@A <device-number>

Functional Example:

In order to use this command, DOS must have focus on your current active drive. By default, we will assume that your drive is device 8. To ensure DOS focus on drive 8, you can issue the @8 command. After that you can change the device to another number.

@A9 will then soft renumber your 1541 to device 9.

To change it back, or to issue any DOS commands on the newly renumbered drive, you need to first issue an @9 to direct your DOS focus to that drive.

Then, you can issue any DOS command and it will be directed by your computer to device 9.

To change it back, or to any other device number, simply issue the @A8 command to change your drives number back to device 8. Don't forget to re-focus DOS to device 8 now using the @8 command.

@BH : Seek Drive Head to Track Zero:

The purpose of this is to force the drive to find track 0 on your floppy drive. There are times when your floppy drive head may become "lost", that could prevent you from loading a directory or any software for that matter. Issuing an @BH will correct that problem.

Syntax:

@BH

Functional Example:

@BH <return>

From the basic prompt, type @BH and press return. The drive will spin, and you will hear a slight

banging in your drive. This is a normal sound. Once the drive stops spinning, the process is complete.

@BQ : <<<unknown>>>:

The purpose of this command is unknown.

Syntax / Functional Example:

This command issued as @BQ will generate a machine response of:

10, no track 1 sensor, 00, 00

@CD : <<<unknown>>>:

The purpose of this command is unknown.

Syntax / Functional Example:

This command issued as @BQ will generate a machine response of:

26, write protect on, 18, 01

@EA : Enable ALL:

The purpose of this command is to enable all of the listed system functions. Regardless of what their current state is, this command will enable them all and display their status on the screen.

Syntax :

@EA

Functional Example:

Enables all functions and returns a response on the screen as the example shows:

@EA <return>

90,RAM ENABLED, 00, 00
91,TURBO ENABLED, 00, 00
92,SCREEN BLANKING ENABLED, 00, 00
93,PHANTOM COMMANDS ENABLED, 00, 00
94,FAST SAVER ENABLED, 00, 00
95,FAST FORMAT ENABLED, 00, 00
96,PARALLEL BUS ENABLED, 00, 00
98,BUMPS ENABLED, 00, 00

It should also be noted that the @EA command can be stacked with other single commands that follow: Examples are as follows:

@EBC <return> will enable Bumps and Phantom Commands the following

98,BUMPS ENABLED, 00, 00
93,PHANTOM COMMANDS ENABLED, 00, 00

@ENP <return> will enable Fast Save and Parallel Bus with the following

94,FAST SAVER ENABLED, 00, 00
96,PARALLEL BUS ENABLED, 00, 00

EB : Enable Bumps:

The purpose of this command is to activate the "Bumps Enabled" to the current state.

Syntax :

@EB

Functional Example:

Returns a response on the screen as the example shows:

@EB <return>

98,BUMPS ENABLED, 00, 00

@EC : Enable Phantom Commands:

The purpose of this command is to set the "Phantom Commands" as enabled to be the current state.

Syntax :

@EC

Functional Example:

Returns a response on the screen as the example shows:

@EC <return>

93,PHANTOM COMMANDS ENABLED, 00, 00

@EF : Enable Fast Saver:

The purpose of this command is to set the "Fast Saver" as enabled to be the current state.

Syntax :

@EF

Functional Example:

Returns a response on the screen as the example shows:

@EF <return>

94,FAST SAVER ENABLED, 00, 00

@EN : Enable Fast Format:

The purpose of this command is to set the "Fast Format" as enabled to be the current state.

Syntax :

@EN

Functional Example:

Returns a response on the screen as the example shows:

@EN <return>

95,FAST FORMAT ENABLED, 00, 00

@EP : Enable Parallel Bus:

The purpose of this command is to set the "Parallel Bus" as enabled to be the current state.

Syntax :

@EP

Functional Example:

Returns a response on the screen as the example shows:

@EP <return>

96,PARALLEL BUS ENABLED, 00, 00

@ER : Enable RAM:

The purpose of this command is to set the "RAM" as enabled to be the current state.

Syntax :

@ER

Functional Example:

Returns a response on the screen as the example shows:

@ER <return>

90,RAM ENABLED, 00, 00

@ES : Enable Screen Blanking:

The purpose of this command is to set the “Screen Blanking” as enabled to be the current state.

Syntax :

@ES

Functional Example:

Returns a response on the screen as the example shows:

@ES <return>

92,SCREEN BLANKING ENABLED, 00, 00

@ET : Enable Turbo:

The purpose of this command is to set the “Turbo” as enabled to be the current state.

Syntax :

@ET
@+
@-

Functional Example:

Returns a response on the screen as the example shows:

@ET <return>

91,TURBO ENABLED, 00, 00

To toggle Turbo mode, perform the following commands

@+ enables Turbo mode
@- disables Turbo mode

@FL : File Lock:

The purpose of this command is to lock one or more files on diskette. This prevents deletion (scratching) of the file. The only way to wipe out a locked file is to unlock it first, or format the diskette.

File locking can be done to a single file, multiple specified files up to a maximum of 5 at once, or with a wildcard (*). Using a wildcard will allow you to do all of the files on a diskette.

When doing a list of up to five filenames, you cannot allow them to wrap to the next line, as this command does not support wrapping.

Syntax :

@FL:<file1>,<file2>,<file3>,<file4>,<file5>
@FL:<filename>
@FL:*
@FL:F* (lock all files starting with F)

@FU : File UnLock:

The purpose of this command is to unlock one or more files on diskette. This allows deletion (scratching) of the file to be done if the user wishes to do so.

File unlocking can be done to a single file, multiple specified files up to a maximum of 5 at once, or with a wildcard (*). Using a wildcard will allow you to do all of the files on a diskette.

When doing a list of up to five filenames, you cannot allow them to wrap to the next line, as this command does not support wrapping.

Syntax :

@FU:<file1>,<file2>,<file3>,<file4>,<file5>
@FU:<filename>
@FU:*
@FU:F* (unlock all files starting with F)

@GO1541 : Revert to Stock ROM:

The purpose of this command is to allow the user to revert the drive back to the original factory Commodore 1541 Drive ROM.

Syntax :

GO1541

Functional Example:

Returns a response on the screen as the example shows:

@GO1541 <return>

73,CBM DOS V2.6 1541,00,00

@H : Header:

The purpose of this command is to rename the header (diskette name) of the diskette. This is the name that is used when formatting a diskette, and the name that appears reversed when listing a directory.

This is a non-destructive process with respect to the data on the diskette. You may perform this function without risk of loss of data.

Syntax :

@H0:<header name>

Functional Example:

@H0:Trilogic Phantom

Take note that the header name you choose must be at least 1 character long and no longer than 16 characters.

@I : Initialize:

The purpose of this command is to initialize the diskette.

Syntax :

@I0:

@KA : Disable ALL:

The purpose of this command is to disable all of the listed system functions. Regardless of what their current state is, this command will disable them all and display their status on the screen.

Syntax :

@KA

Functional Example:

Enables all functions and returns a response on the screen as the example shows:

@KA <return>

90,RAM DISABLED, 00, 00
91,TURBO DISABLED, 00, 00
92,SCREEN BLANKING DISABLED, 00, 00
93,PHANTOM COMMANDS DISABLED, 00, 00
94,FAST SAVER DISABLED, 00, 00
95,FAST FORMAT DISABLED, 00, 00
96,PARALLEL BUS DISABLED, 00, 00
98,BUMPS DISABLED, 00, 00

It should also be noted that the @KA command can be stacked with other single commands that follow: Examples are as follows:

@KBC <return> will disable Bumps and Phantom Commands the following

98,BUMPS DISABLED, 00, 00
93,PHANTOM COMMANDS DISABLED, 00, 00

@KNP <return> will disable Fast Save and Parallel Bus with the following

94,FAST SAVER DISABLED, 00, 00
96,PARALLEL BUS DISABLED, 00, 00

@KB : Disable Bumps:

The purpose of this command is to de-activate the "Bumps Enabled" to the current state.

Syntax :

@KB

Functional Example:

Returns a response on the screen as the example shows:

@KB <return>

98,BUMPS DISABLED, 00, 00

@KC : Disable Phantom Commands:

The purpose of this command is to set the "Phantom Commands" as disabled to be the current state.

Syntax :

@KC

Functional Example:

Returns a response on the screen as the example shows:

@KC <return>

93,PHANTOM COMMANDS DISABLED, 00, 00

@KF : Disable Fast Saver:

The purpose of this command is to set the “Fast Saver” as disabled to be the current state.

Syntax :

@KF

Functional Example:

Returns a response on the screen as the example shows:

@KF <return>

94,FAST SAVER DISABLED, 00, 00

@KN : Disable Fast Format:

The purpose of this command is to set the “Fast Format” as enabled to be the current state.

Syntax :

@KN

Functional Example:

Returns a response on the screen as the example shows:

@NN <return>

95,FAST FORMAT DISABLED, 00, 00

@KP : Disable Parallel Bus:

The purpose of this command is to set the “Parallel Bus” as disabled to be the current state.

Syntax :

@KP

Functional Example:

Returns a response on the screen as the example shows:

@KP <return>

96,PARALLEL BUS DISABLED, 00, 00

@KR : Disable RAM:

The purpose of this command is to set the “RAM” as disabled to be the current state.

Syntax :

@KR

Functional Example:

Returns a response on the screen as the example shows:

@KR <return>

90,RAM DISABLED, 00, 00

@KS : Disable Screen Blanking:

The purpose of this command is to set the “Screen Blanking” as disabled to be the current state.

Syntax :

@KS

Functional Example:

Returns a response on the screen as the example shows:

@KS <return>

92,SCREEN BLANKING DISABLED, 00, 00

@KT : Disable Turbo:

The purpose of this command is to set the “Turbo” as disabled to be the current state.

Syntax :

@KT
@+
@-

Functional Example:

Returns a response on the screen as the example shows:

@KT <return>

91,TURBO DISABLED, 00, 00

To toggle Turbo mode, perform the following commands

@+ enables Turbo mode
@- disables Turbo mode

@ME : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@ME:

Functional Example:

Returns a response on the screen as the example shows:

@ME <return>

The command accesses the 1541, then reports 62,FILE NOT FOUND,00,00

@ML : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@ML:

Functional Example:

Returns a response on the screen as the example shows:

@ML <return>

Reports “Memory Located and Protected”

@MN : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@MN:

Functional Example:

Returns a response on the screen as the example shows:

@MN <return>

Reports “Memory Protection Enabled”

@MP : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@MP:

Functional Example:

Returns a response on the screen as the example shows:

@MP <return>

Reports “Memory Protection Enabled”

@MU : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@MU:

Functional Example:

Returns a response on the screen as the example shows:

@MU <return>

Reports “Memory Protection Disabled”

@N : Format Diskette:

The purpose of this command is to format/prepare a diskette such that files can be stored on the magnetic medium.

This command has an enhanced feature that allows you to format a diskette to 40 tracks. Typical capacity on a stock 1541 is formatting to 35 tracks.

35 tracks equates to 664 blocks free
40 tracks equates to 749 blocks free

This higher capacity works fine on standard DSDD diskettes. This additional storage space is only accessible when the computer is in full Phantom mode.

The header name may only be 1 to 16 characters in length. The ID may only be 2 characters in length.

To enable the enhanced format, add a + after the ID value when formatting.

Syntax :

@N0:<header name>,<id>{+}

Functional Example:

Formatting a standard 35 track/664 block free diskette.

@N0:TRILOGIC PHANTOM,35 <return>

Formatting a standard 40 track/749 block free diskette.

@N0:TRILOGIC PHANTOM,40+ <return>

@P0-9, PA-N : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@P0-9, @PA-N:

Functional Example:

Returns a response on the screen as the example shows:

@P0-9, @PA-N <return>

Reports 70,NO CHANNEL,00,00

@PO : <<< unknown >>>:

The purpose of this command is unknown.

Syntax :

@po:

Functional Example:

Returns a response on the screen as the example shows:

@PO <return>

Reports 64,FILE TYPE MISMATCH,00,00

@Q : Kill DOS / Disable Function Keys:

The overall purpose of this command is unknown. What is known is that the pre-programmed function key assignments go away, and the built in DOS functions also disappear. No further testing has been done on this function at this time.

Syntax :

@Q:

Functional Example:

Returns a response on the screen as the example shows:

@Q <return>

Reports DOS KILLED

@R : Display Drive Registers:

The overall purpose of this command is unknown. What is known is that this command displays the 1541 drives registers it is believed.

Syntax :

@R:

Functional Example:

Returns a response on the screen as the example shows:

@R <return>

```
RST PC  RA RX RY SP NV.BDIZC
>; C05C 10 00 FF 41 11111110
```

@RI : Display Drive Registers:

The overall purpose of this command is unknown. What is known is that this command displays the 1541 drives I/O registers it is believed.

Syntax :

@RI:

Functional Example:

Returns a response on the screen as the example shows:

@R <return>

```
>#1800 C1 FF 1A 00 C6 01 DF 01
>#1808 D9 AD FF 00 0B 40 82 FF
>#1C00 D2 11 6F 00 58 16 00 3A
>#1C08 2F 1B 01 41 EC 02 C0 22
```

@S : Scratch/Delete files from Diskette:

The purpose of this command is to delete individual or groups of files from a diskette that's inserted in your 1541 drive. Take note that if the files have been locked, then this command will not delete them. Instead, it will report an error. Also, if the diskette is write protected, the files also cannot be deleted.

Syntax :

@S0:<filename>

Functional Example:

```
@S0:PACMAN    ← deletes file named PACMAN
@S0:P*        ← deletes all files starting with P
```

@U9 : Disk Drive Reset:

The purpose of this command is to reset your disk drive to a powered on state, with the Phantom drive rom active.

Syntax :

@U9:

Functional Example:

Returns a response on the screen as the example shows:

@U9 <return>

Reports 73,8K PHANTOM DOS V1.1E,08,08

@V : Validate Diskette:

The purpose of this command is to rebuild the BAM (block allocation map) on the floppy diskette. The BAM is located on track 18. This is the only way of removing a corrupted file from the directory. A corrupted file may have a file length of zero, and have an asterisk (*) beside it. Once the validation command has completed, the BAM will have been rebuilt, and the corrupt file(s) will have been removed. This is a permanent modification that occurs.

Syntax :

@V:I ← Validate and ignore read errors
@V:L ← Validate and learn read errors

@WC : Write Protect Ignore until disk change:

The purpose of this command is to tell the disk drive that it should not observe the write protect sensor on the drive, even if the write protect tab is in place. This command will

remain in place only while the current diskette is in the drive. As soon as it is removed from the drive, the drive sensor will return to normal function.

Syntax :

@WC

@WI : Write Protect Ignore:

The purpose of this command is to tell the disk drive that it should not observe the write protect sensor on the drive, even if the write protect tab is in place. This command will remain in place until it is either switched off or the drive is powered down.

Syntax :

@WI

@WN : Write Protect Return to Normal:

The purpose of this command is to tell the disk drive to return to normal detection mode. Typically this command would be used if you wanted to reverse the @WI command.

Syntax :

@WN

@WU : <<< unknown >>>:

Not a lot is known about this function. What it appears to do is modify the diskette in some fashion such that it is no longer possible to write to the diskette. No reversal process has been found other than the destructive FORMAT (@N) command. Use with caution.

Syntax :

@WU

Functional Example:

Returns a response on the screen as the example shows:

26,WRITE PROTECT ON,18,00

SYS61916 : RENEWED:

Not a lot is known about this function. What it appears to do is some sort of “NEW” function, but not a lot is known about it.

Appendix: INSTALLATION:

Installation of this board into your drive will take approximately 30 minutes. Make sure you have a well lit and tidy work area before starting.

Additionally, your Commodore 64 will need to have a socketed kernal. If your Commodore 64 Kernal ROM (the chip marked as 901227-03) is soldered onto your computers mainboard, then please seek a qualified professional to remove the chip and install a socket. Attempting this yourself can cause irreparable damage to your computer if you don't know what you're doing. Nobody including Trilogic cannot be held responsible for any damage to your Commodore 64.

Tools required are:

- Phillips head screw driver
- Small flat head screwdriver
- Anti-static grounding strap for yourself.

***STEP 1** : Remove / Replace the Kernal ROM in your 64.*

Estimated Install time: 10 minutes

In this step, you will be removing your Kernal ROM from your machine and replacing it with THE PHANTOM rom provided in your package. This Phantom kernal replacement rom actually contains TWO ROMS.

1. Original Commodore Kernal ROM
2. Custom PHANTOM ROM. The most current version of the PHANTOM ROM is v1.07

- Remove the screws from your Commodore 64 (the three screws along the bottom front edge of the machine, and lift up the top half of the computer (like you would the hood of your car).

- Locate the kernal rom location on your Commodore 64 and remove it, taking note of the notch location on the chip.
- Insert the Phantom kernal rom adapter board into the same location, taking care to ensure the notch on the Phantom rom is oriented in the same fashion.
- Locate and install the provided switch in the case, or use the provided jumper if you prefer to always use the Phantom kernal on your machine.
- If you choose the jumper option, then unplug the switch and discard, and place the jumper across the two pins closest to the notch end of the chip on the Phantom kernal board.
- Close up your Commodore 64, and do a power on test of your Commodore 64. One of the switch positions will provide you with the factory start up screen, and the other position will be the Phantom start up screen that will look like figure 1.

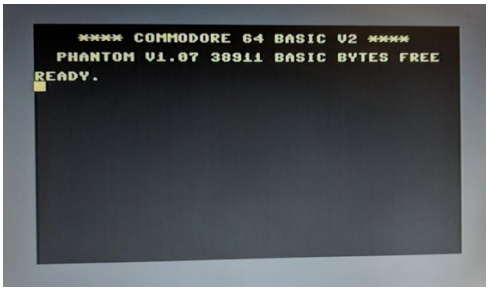

Figure 1

Step 2: *Install The Phantom Drive Board*

Estimated Install time: 30 minutes

In this step, you will be installing the brains and brawn of The Phantom. It may seem daunting, but take your time and make sure you have good

lighting in your work area to ensure everything is plugged in as it should when you're done.

Taking photos of the process as you go along before and after you remove/install components will help you should you forget how things were before and after during the installation process.

1. Open up the 1541 drive by removing the four screws on the underside of the drive. This requires that you use a Philips head screwdriver.
2. Once the top has been removed from the drive, if there is a metal shield installed above the main logic board, remove that as well. You won't be able to use that shield if it is place after the Phantom is installed, as the Phantom occupies that space.
3. Remove the 6502 CPU at location UC4, and the 6522 VIA at location UC3 On the 1541 main board.
4. Remove the cables from locations P4, P5, P6 on the edge of the board taking note of their orientation. It helps to mark these connectors appropriately with a felt marker before removing them so you don't forget how they were installed.
5. Connect the "sync" cable to the three pins on UC1 and UC2. Ensure you connect them to the correct pins as marked on the cable.

   MOS325572-01 (UC1) : 1 – Pin19, 2 – Pin21
   MOS6522 (UC2) : 3 – Pin 17
6. The next part is a little tricky, so make sure you really take your time lining up all the pins. Bending the pins on the underside of the Phantom board would be catastrophic, as that is not easily repaired.

It is for this reason it is HIGHLY RECOMMENDED that you install the two 40 pin sockets onto the male pin headers on the underside of the board BEFORE installing the board. In this way, if you happen to bend a pin installing the Phantom in your 1541, it will only be on the socket, not the male pin header on the Phantom. The socket is easier to replace than the pin headers on the board. **If you purchased an assembled board, that is how it was delivered to you.**

It helps to set it in place, then before pressing it in, use a flashlight to look between the boards and visually inspect where all the pins are... that they are all positioned where they should be. If you do that, you shouldn't have any problems.

If you're finding you're having troubles with it working, gently lift and lower the edge of the phantom where it plugs into the pins on the edge of the board. I've found that on some drives, those pins are somewhat corroded and you have to work them a bit to make all the connections.

7. Once the board is successfully pressed into the socket and pin locations on the board, insert the cables you removed from the logic board in step 4 and sync cable you installed in step 5 to their respective points on the Phantom.
8. Insert the parallel cable connector onto the Phantom board (if you have not already done so), and route the ribbon cable out the back of the drive.
9. Before closing up the drive, connect the parallel cable to the Commodore 64 user port

(taking note of which side is up), connect your serial cable from the Commodore 64 to the 1541 drive containing your newly installed Phantom board, and power on the system.

10. You should be greeted with the Phantom start screen when you power on your Commodore 64.
11. Press the '@' symbol and press return. The computer should respond with the startup DOS response from the drive as follows:

```
 73,8K PHANTOM DOS V1.1E,00,00
```

If you received this message then the board is successfully installed, and you may close up your 1541 drive.

If you do not, and/or the computer locks up when you do that, then you have not properly installed the board in the drive. More than likely, you probably have not pressed the board down sufficiently onto the 1541 logic board, or you have bend a pin or pins.

At this point, you should remove the Phantom logic board, and examine all of the pins. If you bent some, very carefully and slowly try to straighten them. In most cases, unless the pins have been severely kinked, you can straighten the pins once if you're careful before they break off. If you break off any pins, the board will need to be repaired and the connector strip will need to be replaced.

Assuming that you've gotten the correct DOS response, congratulations!