PHILIPS

PHILIPS

PHILIPS INTERACTIVE MEDIA of America

Technical Note #76

Differences between the Philips 18x and 605 Development Platforms

Charles Golvin

April 16, 1992

This note summarizes the major differences between the Philips 18x and 605 development platforms. This note is based on revision 1.1 of the 605 ROM.

Differences between the Philips 18x and 605 Development Platforms

Recently, Philips announced the commercial availability of its 605 series CD-I player. This player serves as the replacement for the 18x player, which has until now served as the stalwart CD-I development platform. This note attempts to summarize the differences between these two platforms.

> *Note: This technical note is based on revision 1.1 ROM in the 605 player. Therefore, it is possible that, should you be in possession of a 605 player, the behavior you observe may differ from that described herein. The 1.1 ROM updates for the 605 player will be distributed in the early May, 1992, time frame.*

y Components

The key components of the 605 player are identical to those of the Philips 910, or consumer, player. However, with the exception of the VSC chips, the key components of the 605 player are different from those of the 18x player. (That is, the 605 contains new versions of the CDIC, CPU, and DSP; and, the VSD replaces the VSR.) The following problems (which do not comprise a comprehensive list) are corrected on the 605 by virtue of these changes:

- Loading of CLUT banks 2 and 3 did not conform with the Green Book
- Matte registers were reset to false one pixel prior to the end of each scan line, rather than at the start of each scan line
- Matte ICF changes at offset 0 changed the ICF of the entire plane's display
- Occurrence of the sync pattern in a sector's unscrambled data could cause seek problems
- Possible loss of selected sectors being delivered to memory because of the combination of seeks and sound map plays
- The "twilight zone" problem: this relates to file positions that fall between main channel times of an integral number of minutes and two seconds thereafter [xx:00.00 to xx:01.74]. There were two manifestations of this problem:

  (1) Incorrect initial play position if an explicit seek was not issued prior to a play and the current file position was in the described two second window
  (2) Incorrect file position reporting within this two-second window.

The latter two problems are related to Philips-specific hardware in the 18x player. However, the former three problems are related to one of the video display chips that is used by various other hardware manufacturers. Therefore, we strongly urge that title applications continue to employ the PIMA-recommended workarounds for these problems. Consult PIMA Technical Note #62, *Living with Reality: Workarounds for Remaining Bugs in CD-RTOS 1.1* for an explanation of these workarounds. With respect the "twilight zone" problem, it is a recommended practice to force an explicit seek seek(); i.e., I$Seek suffices) prior to issuing a play command.

he 18x player derives its video and ticker rates from a single crystal, however, the 605 ayer uses different crystals for NTSC and PAL operations. The video and ticker rates the 605 player, for both PAL and NTSC, are different from the corresponding rates the 18x player.

ne known problem on the 605 that relates to the hardware components is an incorrect ack level. This is the source of the "12 pixel offset" problem that also existed on the 3x players.

layer Shell

ome features of the player shell are new; others are the same as those of the 18x player owever, the button descriptions for these common functions are different. Two atures that remain are the ability to start a "floppy application" and the ability to start ie -9 shell (which is conveniently in ROM). These two options are available from ttings menu. To start a floppy application, place a properly formatted floppy disk e drive and select **Load** from the **Settings** menu. To start the OS-9 shell, select the ystem button from the **Settings** menu.

> *Note: To use either of these features, the diskette must be formatted as an OS-9 device, not a PC device. More information regarding PC devices follows below.*

this case, the OS-9 shell is chained; there is no player shell ability to fork the OS-9 ell. However, this can be easily achieved by chaining to the OS-9 shell, setting the efault data and execution directories to the CD device, and then starting the player ell in the background via the following command:

```
play >>>/nil &
```

he shell prompt will return, and the player shell will start up. However, it is not ossible to start the CD-I application from an invocation of the player shell from the S- hell. To start the application use the "launcher" command: use the -c option to n the application on the optical disc (emulates selecting the **Play CD-I** button); use e -f option to launch the application on the floppy disc (emulates selecting the **Load** utton at the **Settings** menu).

> *Note: If the System button is not present, then either a serial cable is not connected to serial port 3 on the back of the player or the serial connection is not correct.*

There is a program in the player ROM called rdvid that may be used to examine and modify the settings currently in use. Values changed using this program are recorded in the NVRAM file, and are, hence, re-used with the **Previous** option until **Randomize** is once again selected. To a large extent using the **Randomize** and **Previous** features eliminates the need for the test programs **cdi_deinit** and **cdi_chgtix**.

- **Seek Delay**
  If you select **On** for this option, the player forces the greatest seek time allowed by the Green Book for all seek requests. This results in a seek time of one second for a seek to any address within twenty megabytes of the current position, increasing linearly to a maximum of three seconds across the entire disc. This feature allows you to test whether your application tolerates the worst case specified by the Green Book.

- **Additional Memory**
  The 605 player has a total of five megabytes of RAM, consisting of the base-case one megabyte of colored memory and four additional megabyte of system RAM. The value in the rectangle of the **Additional Memory** menu displays the amount of system memory accessible at boot time. Thus, the possible values range from 0.0 to 4.0. The + and - buttons above and below this display allow you to change the available memory in 0.5 megabyte increments. Modifying the current value causes the player shell to exit. Thus, if the player shell is the only process running in the system, the player resets and the newly modified memory setting is obtained when the player shell restarts.

Note: The decision by the player whether or not to reset is determined by whether the +/- buttons are used—the reset is triggered whether or not the actual memory setting changed. The current additional memory setting is visible from the main screen of the player shell, displayed above the ***Tools*** *menu button as RAM: N.M.*

When the amount of additional memory is set to 0.0 the player is, with respect to available colored memory, a worst case, base-case player. Prior to loading the application into plane B, there are two free blocks of 480 Kilobytes—one in each plane. This replaces the functionality of the test program **cdi_basecase** (in fact, it improves on it, because the annoying 512 byte block in plane A is eliminated).

When the additional memory is set to less than the maximum of 4.0, the remaining RAM is treated as ROM. This memory is searched at start up, and any modules found are entered into the module directory. This is a very powerful mechanism. You may leave your favorite utilities in memory so you do not have to load them off a floppy disk; or you may introduce a newer version of a system module without having to *deiniz* the associated device, etc. (The **le0** ethernet device descriptor is a perfect example.) The process for this follows:

◊ Set the additional memory to 4.0.
◊ Start the OS-9 shell.
◊ Load the module(s) into memory.
◊ Start the player shell from the command line by entering play<CR>.
◊ Change the memory setting to a value less than 4.0 and sufficiently large to assure that the loaded modules are in the protected area. (Use mdir and knowledge that system memory allocation is always first fit from high to low addresses to determine this. For example, when no RAM is reserved and the OS-9 shell is chained, the largest free block is at address $601AB0.)
◊ Reset the player.

When the player restarts, the modules will be in the module directory. Another powerful use of this memory is to employ it as a debug buffer for problems that cause the player to crash. Since the memory is treated as ROM, information written to this memory is intact after a reset.

aware that the more memory you reserve to be treated as ROM, the longer the rt-up time will be, because the memory has to be searched for any modules to be loaded into the module directory. In order to save time at start up, only the parity of found modules is checked, not the CRC. Therefore, it is possible to find modules in the module directory that do not have a correct CRC value.

er Differences

Default Device

When the OS-9 shell is chained, the default data and execution directories are not set to anything meaningful. That is, the directory "." is not well defined. You must explicitly chd and chx to the directories you intend to use for these respective purposes.

Diskette Density

The 605 player has different device names for different density diskettes. Thus, unlike the 18x player where /d0 worked for most of the formats commonly used, ne 605 player d0 describes only a normal density OS-9 "standard" formatted floppy. If the diskette in the floppy drive is high density, the proper device name is /d0h. Other device names and the corresponding formats are:

«»	/d0hd	Microware "highest density" format
«»	/d0uv	Microware "universal" format
«»	/d0tb	Microware "testbed" format
«»	/pcd0	Low density PC format
«»	/pcd0h	High density PC format (more on these two below)

Device Names

The names of some of the devices were changed in revision 1.0, but revision 1.1 has returned them to the original names known from the 18x player, such as: /cd,

/ap, /vid, /ptr, etc. The author maintains that the 1.0 scheme was a good one, because it prevented developers from hard coding the "standard" device names into an application. It is possible that some median ground will be struck on this point in the future, provided there is a solution whereby the odd names may be imposed strictly at will for testing.

- CSD
  There are several entries in the 605 CSD that you will not find on the 18x. There is an entry for the player control keys (for example, the front panel controls or the buttons on the remote controller other than the two action buttons) which, though functional on the 18x player, are not reflected in the CSD. For more information on using the player control keys, please consult PIMA Technical Note #73: *Reading the Player Control Keys*. In addition, the CSD now reflects the extension features the player possesses. These include device 20 (SCF for RS232 communication), device 40 (RBF for Microware format floppies), and device 60 (PCF for PC format floppy drives).

- Ram Disks
  There is a RAM disk in the 18x player (seldom used though it may be). In addition, the 605 contains several different RAM disks of various sizes: 32K, 64K, 128K, 256K, and 512K. These are named r32, r64, etc., and must be initialized (with iniz) prior to use. These can be very useful in lieu of the second floppy drive present on a 18x player.

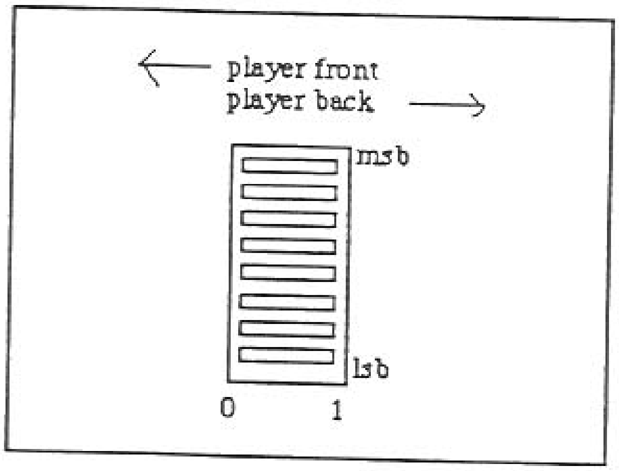
- PCF
  There is a new file manager in the 605 player called **pcf**, the PC file manager. This allows the player to read and write PC-formatted floppy disks, both normal and high density. The devices are initialized at start up so the only necessary proviso is the same as the one outlined above: use the correct device name, making sure to take into account the density of the diskette you're using.

  Note: There was a problem in the previous version of pcf in which high density diskettes could provide correct directory listings, but would not correctly read the dat. This bug has been fixed in both versions 1.0 and 1.1 of the 605 ROM.

- Pointing Device Initialization
  The device that ends up in the CSD entry for the pointing device depends on which pointing devices are connected and to which port. If only one device is connected, either to the front or rear port, it will be the first device in the CSD and will be called /ptr. In the latter case, there will also be an entry in the CSD for a second pointing device called /pt2. This corresponds to the remote control unit, which is always assumed. In the case that no devices are connected, the entry in the CSD reflects the remote control unit.

- **Ethernet**
  In addition to the extra four megabytes of RAM, the extension card for the 605 player also has ethernet. The ethernet modules are in the ROM, although the versions in the ROM are not the most up-to-date editions. To connect the player to your ethernet network (assuming you have one), it is necessary to create a correct **le0** module for yo'ır player and to properly set the dip switches on the board.

  The dip switches are found on the left side of the ethernet board as you look at the board from the player rear panel. From this view, the switches from left to right move from least significant bit to most significant bit. A dip switch set toward the back of the player sets the corresponding bit, and a dip switch set toward the front of the player clears the corresponding bit. The following diagram illustrates this:



- **Customized Start Up Environment**
  It is possible to set up a customized start-up environment so that default data and execution directories can be set, environmental variables may be defined, ethernet may be started, etc. The environment is described by a file in NVRAM called **startup.prf**. This file is used to set the environment of the shell chained to from the player shell, provided:

  (1) the file exists;
  (2) and the "system" button is clicked twice rapidly ("double-clicked").

  If the file **startup.prf** does not exist when the system button is double-clicked, the shell is not started and an error message is written to the serial port.