3.2 Entering Level II Code via the RS-232 Port

The user can enter (download) Level II code into the player's memory from an external computer by using the RS-232C port. This downloading of data is accomplished by using new Level III commands. Thus, the player must be ON or in *Manual Mode* (such that Level III commands will control the player) before entering Level II code using this *Downloading Mode.*

3.2.1 Downloading Level II Codes

To use *Downloading Mode* to send data (as Hex codes) from a computer to the player's memory, use the following two Level III commands:

· 1. Set the Memory Address pointer

Before sending any data to the player, it is usual to specify the memory location (address) where the first byte of program code is to be written. Use the *S command, as shown in the example below, to start writing data at memory location 100. The argument specifies the memory location as a decimal number. If the *S command is successful, an "R" will be sent by the player. If the memory location has already been specified by other means, this step may be omitted.

```
100*S<CR>
```

Note that memory locations are almost identical to program addresses. See ***Section 2.5, Random Access Memory*** for details about memory locations.

• 2. Download data

Specify a data length (the number of code bytes to be sent to the player) and then send the data. This is a two-step Level III command.

Step 1: Use the *W command to specify a maximum number of data bytes to be written into the player's memory by the next step. Without this, the player would try to interpret the following data as Level III commands. If the *W command is successful, the player will transmit an "R". The computer should wait for this response from the player before proceeding.

Step 2: Send the data, two Hex digits for each code byte, followed by a carriage return. A maximum of 64 bytes may be specified in Step 1. Thus, a maximum of 128 Hex characters should be transmitted in Step 2.

For example, transfer 8 bytes of data:

```
8*W<CR>
            R
AFBFCFDF01020304<CR>
            R
```

Since a maximum of 64 bytes of Level II code can be sent to the player in one download operation, it will usually be necessary to download many times. Since the memory location pointer is incremented by one each time one byte of program code is written, it is usually not necessary to use the *S command except at the beginning of a sequence of download operations.

Example: The Level II program codes for1000 SEARCH, 1200 AUTOSTOP, HALT are sent to the player's RAM beginning at address 120. Obtain the Hex codes for each code byte - use ***Chapter 4*** or ***Appendix B*** and ***E***. This 11-byte program is represented by the following Hex codes:

```
1000  SEARCH      -->    0F, 3F, 3F, 3F, F7
1200  AUTOSTOP    -->    0F, 8F, 3F, 3F, F3
      HALT        -->    BF
```

Set the program memory pointer to 120:

```
>     120*S<CR>
<                        R<CR>
```

Set the data length to 11 bytes:

```
>      11*W<CR>
<                          R<CR>
```

Send the Hex codes for the 11 bytes:

```
>      0F3F3F3FF70F8F3F3FF3BF<CR>
<                          R<CR>
```

(The memory location pointer will now be 131.)

When downloading Level II code into the player's memory, the code can be written to any byte of any page, even if the page is not active. Even though they are essentially identical (except for referring to Register 0), Memory Locations (Memory Addresses) are used instead of Program Addresses. See **Figure 2-D**

If one wishes to download data into specific registers, the user must take into account the number of pages that will be active when the registers are accessed by the Level II program. For example, Register 2 would be at Program Address 1018 and 1019 when there is only one page active. However, it will be at 2040 and 2041 if there are just two pages active. Register 0 is always accessed as Memory Locations 7154 and 7155.

Recall that each register uses two bytes of memory. The most significant byte of the register data is written at an even-numbered address (lower program or memory address). The least significant byte is written at an odd-numbered address (higher program or memory address). The formula to calculate the memory locations that a register occupies is shown below.

Memory Location of Register Y, assuming it is accessed by a Level II program after an X PAGE command was executed (with 0 ≤ X ≤ 6, and 0 ≤ Y ≤ (511 x (X + 1))):

If Y = 0, then (for R 0):
a.) The most significant byte of R 0 is at memory location 7154.
b.) The least significant byte of R 0 is at memory location 7155.

If Y >0, then (for R Y):
a.) The most significant byte is at memory location (((511 x (X +1)) - Y) x 2.
b.) The least significant byte is at memory location (((511 x (X +1)) - Y) x 2 + 1.

3.2.2 Reading Level II Codes

Program codes can be read from the player's memory by an external computer in a manner similar to that used to send them to the player.

First, set the memory location pointer with the *S command just as you would for sending code to the player. The location specified will be the first code byte read.

Second, use the *D command to tell the player now many code bytes to transmit from it's memory. The maximum transmission is 64 bytes of code, followed by a carriage return. Each byte is sent as two Hex characters, so a maximum of 128 characters are transmitted.

To aid reading sequential sections of the player's memory, the memory location pointer is increased by one every time a byte of data is output.

Example: Read the first 9 bytes of the program downloaded in the previous section.

- **Set the memory location pointer.**

As before, set the memory location pointer to 120:

```
>      120*S<CR>
<                          R<CR>
```

- **Transfer Data.**

Ask the player to transmit 9 bytes of code:

```
>      9*D<CR>
<      0F3F3F3FF70F8F3F3F<CR>
```

The memory location pointer will be at 129 when the player's transmission is complete.

At this point the programmer should note that the player's Program Address Pointer is used for two different purposes:

First, it is used during the execution of a Level II program to indicate the instruction being processed. While being used in this manner, the Program Address counter will point to a location in active memory.

Second, it is set by the *S command and is used as a Memory Location Pointer by the Level III downloading and code reading commands. At this time, it may point to any of the byte locations (addresses) in memory, even if only one page is active.

3.2.2 Reading Level II Codes

Program codes can be read from the player's memory by an external computer in a manner similar to that used to send them to the player.

First, set the memory location pointer with the *S command just as you would for sending code to the player. The location specified will be the first code byte read.

Second, use the *D command to tell the player now many code bytes to transmit from it's memory. The maximum transmission is 64 bytes of code, followed by a carriage return. Each byte is sent as two Hex characters, so a maximum of 128 characters are transmitted.

To aid reading sequential sections of the player's memory, the memory location pointer is increased by one every time a byte of data is output.

Example: Read the first 9 bytes of the program downloaded in the previous section.

- **Set the memory location pointer.**

As before, set the memory location pointer to 120:

```
>       120*S<CR>
<                          R<CR>
```

- **Transfer Data.**

Ask the player to transmit 9 bytes of code:

```
>       9*D<CR>
<       0F3F3F3FF70F8F3F3F<CR>
```

The memory location pointer will be at 129 when the player's transmission is complete.

At this point the programmer should note that the player's Program Address Pointer is used for two different purposes:

First, it is used during the execution of a Level II program to indicate the instruction being processed. While being used in this manner, the Program Address counter will point to a location in active memory.

Second, it is set by the *S command and is used as a Memory Location Pointer by the Level III downloading and code reading commands. At this time, it may point to any of the byte locations (addresses) in memory, even if only one page is active.

Since it might be useful for an external computer to monitor the execution of a Level II program, the player has a new *P command (Level III) to ask the player to transmit the value of the Program Address Pointer.

• Transmit the value of the Program Address Pointer
The player transmits four decimal digits followed by a carriage return. For example:

```
        >       *P<CR>
        <       0129<CR>
```