Apple II, IIe and II+: Auto-Run Apple w/o DOS (1 of 2)

Revised: 3/2/88
Security: Everyone

Apple II, IIe and II+: Auto-Run Apple w/o DOS (1 of 2)

==========

This article last reviewed: 21 September 1984

Some applications require that an Apple start running an Applesoft program from power-up without human interaction. This is easy with the disk and Auto-Start ROM. Simply initilaize the diskette with the desired program in memory and the disk will boot and run it when the power comes on. But sometimes a disk drive is undesirable, especially where there is only one program to run and the cost must be kept low. So here is a way to have a card that will load and run an Applesoft program automatically on power-up in any Apple II+.

I will assume the use of a card like the Mountain Computer ROM+ that has a 256 byte "control ROM" and room for some larger ROMs to store the Applesoft program. On the ROM+ this is a bank of up to six 2716 type EPROMs. Using EPROMs has the advantage that you can change your Applesoft program later by erasing the EPROMs and reprogramming and the disadvantages of higher cost and using more power from the Apple's power supply. The power consumption of the EPROMs won't be a problem if the Apple isn't filled with cards.

The software in the control ROM is required to do five things:

1. Pretend that it's a disk controller card so that the Auto-Start ROM will execute its code.

2. Initialize Applesoft.

3. Move an image of the Applesoft program down from the ROMs into the proper area of RAM.

4. Set up the required Applesoft pointers for the end of the program.

5. RUN the program.

All that's needed to convince the Auto-Start that there's a disk controller card out there is to have a ROM whose first four odd bytes match the Apple P5

or P5A PROM. If the monitor finds a ROM that matches it in slot N, it will do a jump to $CN00. The routine that does this starts at $FAA6 and is listed on page 144 of the Apple II reference manual. The first eight bytes in the control ROM will be

        24 20 24 00 24 03 24 3C

Note that by having the even numbered bytes equal to $24, (BIT Page zero) when the code is executed starting at $CN00 nothing will happen until the byte after the $3C.

The proper way to initialize Applesoft is to jump to $E000. Unfortunately this entry into Applesoft falls into the normal command level routine. To regain control so that the control ROM can load a program we can use the same trick that DOS uses. As soon as Applesoft reaches its command level it prints a prompt, ("]") and waits for the user to type in a command. Since all input and output in the Apple is handled through two pointers in RAM, we can divert the input routine to point back into the control ROM. This will leave several levels of subroutine on the 6502 stack but Applesoft will re-initialize the stack when we RUN the program so it doesn't matter.

Now the question becomes what address to put into the pointer. The control ROM's address will change depending on which peripheral slot it's plugged into.

The low byte is just the offset from the start of the ROM since the address always starts on a 256 byte boundary but the high byte could be anything from $C1 to $C7. When the Auto-Start ROM looks for a disk controller card it saves the high order byte in $7F8. So the contents of $38 and $39, the input pointer, becomes the offset which is stored at location $7F8. Then we can jump to $E000 to initialize Applesoft confident that we will regain control when it's done.

Tech Info Library Article Number:9