

Transferring BASIC Programs From One Computer to Another

Details on an efficient way to transfer BASIC programs written for other computers to yours, using an Apple IIc as an example

By Fred Blechman, K6UGT

There are literally thousands of programs written in BASIC for microcomputers. Should you wish to run one that's written for another computer than your machine, or vice versa, there are some bumps in the road ahead of you. Assuming the program is written in some dialect of Microsoft BASIC, as most are, you have to make appropriate language changes to meet the target computer's requirements. Some translations are straightforward, while others border on the impossible, especially where graphics are concerned.

Before making translations, though, you must get the original BASIC into the target computer's memory so that you can make any changes in syntax and key words to allow the target computer to run the program. You can enter the program on the target computer's keyboard, of course. But this is a tiresome chore that, additionally, invites typing mistakes that have to be debugged. To avoid the drudgery, BASIC pro-

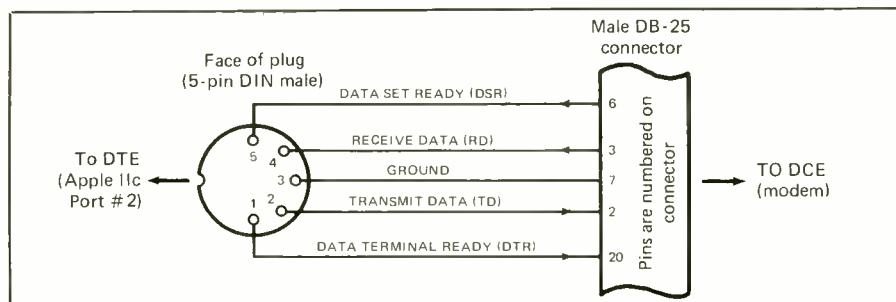


Fig. 1. This drawing shows details of modem interface cable connections between the 5-pin DIN and standard DB-25 connectors.

grams are best transferred in or out of a computer in ASCII (American Standard Code for Information Interchange). This converts BASIC "tokens" (key words) to individual characters so that the program is not machine-specific, saving you considerable time and ensuring an accurate transfer.

Most computers provide a simple method for converting a BASIC program from tokenized form to ASCII. Unfortunately, Applesoft BASIC does not. Since Apple computer models are so widespread, I'll use one of them, the Apple IIc, which is among the more popular personal computers sold today, as an example of how to make the program transfer efficient and painless. With this done, you would run the program, trying out equivalent key words

whenever a line crashes. A bonus for Apple IIc owners will be some technical information, such as connector pin-outs, that the beautiful IIc manual neglects.

Computer-To-Computer Transfer

It is relatively simple to transfer programs directly between two computers if you have the proper equipment and the source and target computers sitting physically close to each other. You can use a null cable to directly connect the two together. If the computers are located distantly from each other, however, you will have to add modems and operate through the telephone lines.

For the sake of illustration, let us assume you wish to transfer a BASIC

(Continued on page 42)

Note: This article is based upon material in the author's book Apple IIc—An Intelligent Guide, soon to be published by Holt, Rinehart & Winston (CBS Computer Books).

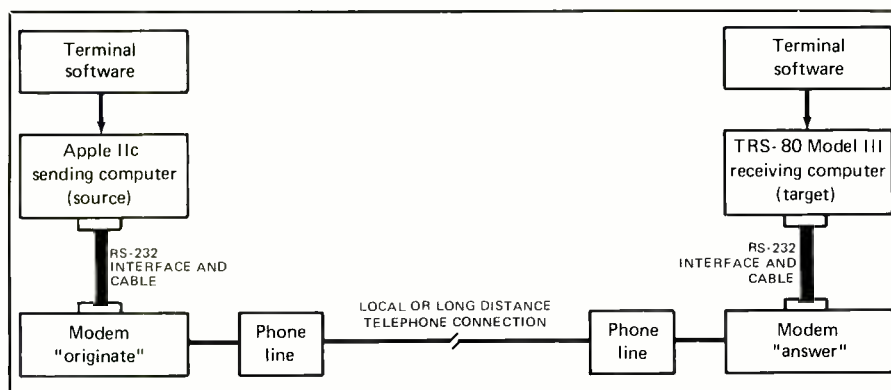


Fig. 2. Diagram illustrates typical telecommunications systems, using RS-232 interface and cable and modem at each end of phone line.

program from a TRS-80 Model III disk system to an Apple IIc. While this example will discuss specifics for transferring data between these two computers, the general approach can be used with just about any other combination of microcomputer.

Terminal Programs. Each computer will require a "smart" terminal program that can upload and download (send and receive) programs either directly from disk or from a memory-storage area called the *buffer*. Many such programs are available for the TRS-80 Model III and for most other micros. This is not true for the Apple IIc, however.

Although there are many terminal programs for earlier Apple II-series computers, most of them will not work in the IIc. Fortunately, by the time you read this, several are expected to be available.

It is best if the terminal programs support the Christensen protocol, also known as the XMODEM, MODEM7, and CP/M protocols. This method of transferring files, using 128-byte verified blocks, assures virtually error-free transfers.

RS-232 Interface. You need an RS-232 (serial) interface installed in each computer. The Apple IIc has this built in and uses Port No. 2 for this purpose. An RS-232 interface is frequently included with the TRS-80 Model III or is available as an option.

Similarly, most common micro-

computers have RS-232 interfaces built in or available as options for \$75 to \$100.

Cables & Time. Getting the equipment you need to transfer programs should not be a problem. However, plan on spending considerable time finding or making the special cables you will probably require for use with the Apple IIc. Also, getting acquainted with the terminal software for each computer can be very time-consuming. Each comes with documentation that describes its particular features and operation, and it seems that no two are the same!

The Computer Connection

Entire books have been written on the subject of telecommunications, so it would be presumptuous to attempt to cover this subject in detail in this article. If you need information on telecommunications, consult any of the books on the market.

Interfacing an Apple IIc to a modem is simple and straightforward—if you use an Apple modem. Unfortunately, Apple charges a premium price for its equipment. Too, you may already have a modem from a source other than Apple. If you wish to avoid the challenge of interfacing a non-Apple modem to your IIc, with its resultant uncertainties, I suggest that you pay the premium

price and get an Apple modem. For those who are more adventurous (or are on a tighter budget) I'll cover some alternatives.

Example Only. For the example of transferring programs from the TRS-80 Model III to the Apple IIc, I will assume you have a modem, an RS-232 interface for the TRS-80, and a terminal program for each computer, and that you know how to operate the modems and use the terminal programs. This being the case, I will simply review the procedures to use.

Modem Connection. Figure 1 illustrates a typical modem interface cable for the IIc, while Fig. 2 shows the two computers connected together via modems and the telephone line. The latter approach requires two different telephone numbers, a necessity if the two computers are not near each other.

Hard-Wire Connection. Since my TRS-80 and IIc were just a few feet apart and not wanting to tie up my phone for hours as I tested the programs and procedures, I eliminated the phone lines and both modems by making a direct RS-232-to-RS-232 connection (Fig. 3). I did this by using the Port No. 2 cable shown in Fig. 1 and a standard RS-232 cable. In most cases, this works better than using modems and somewhat simplifies the procedure, since the phones and modems are bypassed. Also transfers at higher rates—9600 baud, for example—are practical with direct wiring, if the terminal programs can handle the higher transfer rate.

You may be puzzled by the "null modem adapter" shown in Fig. 3. The RS-232 interface wires 2 (transmitted data) and 3 (received data), normally connected straight-through to a modem, must be interchanged if you directly wire two computers together. Otherwise, you will get no response.

The foregoing is done because both computers are acting like terminals (DTE, or data terminal equipment). The RS-232 standard expects

a DTE to be connected to a modem (DCE, or data communication equipment), with wires 2 and 3 connected straight-through.

Signals are transmitted from the DTE to the modem on pin 2 and are received on pin 3 of the DTE. However, if two DTEs are connected in this manner, they will both be attempting to send on pin 2 and receive on pin 3. This problem is solved simply by swapping the wires on pins 2 and 3 of the conventional DB-25 RS-232 interface connector. You can buy a commercial unit, called a *null modem adapter* for \$10 to \$20, or make your own from male and female DB-25 connectors.

Incidentally, this same wire interchange is necessary when using a printer. So a serial printer cable made for use with the Apple IIc can be used for direct wiring between two computers, eliminating the need for a null modem adapter.

Preparation. Connect the hardware together as described. However, before you actually start transferring data or programs between computers, determine what data or program you want to transfer and establish the target computer. The most important thing to realize is that the transmitted file, whether a BASIC program or a data file, should be in ASCII format when transmitted. Since most text and data files are in ASCII, they are normally not a problem. BASIC programs, however, do pose a problem.

Note: Strictly speaking, some terminal programs allow direct transmission of BASIC programs between like computers, using the Christensen protocol. However, this can lead to immense confusion and problems between different computers. So plan on direct transmission of BASIC programs only if you really know what you are doing. ASCII files are much more compatible.

Since BASIC programs are normally saved in binary (also called

tokenized or *compressed*) format, you must first change any BASIC file to the ASCII format. You do this by loading the program into your BASIC memory and then saving it on disk with the filename followed by a comma and an A, for most computers. Unfortunately, you cannot do this in Applesoft BASIC, but we will get to this later.

For example, ORDER/BAS on a TRS-80 would be loaded with the LOAD"ORDER/BAS" command and then saved with the SAVE"ORDER/ASC",A command. The extension /ASC could actually be any three-letter code, but this is the most commonly used to designate that the file is in ASCII format, to distinguish it from the /BAS binary file. The ,A after the filename tells the computer to save this program in ASCII format. *This is very important!*

Matching Baud Rates. It is extremely important that the baud rate and word format (signal rate and sequence) be the same at both the transmitting and the receiving ends of the telecommunications link. Accomplishing this is a matter of using merely whatever commands are required in the terminal program to set baud rate

(usually 300 or 1200 when using modems) and word format (usually eight data bits, one stop bit and one parity bit).

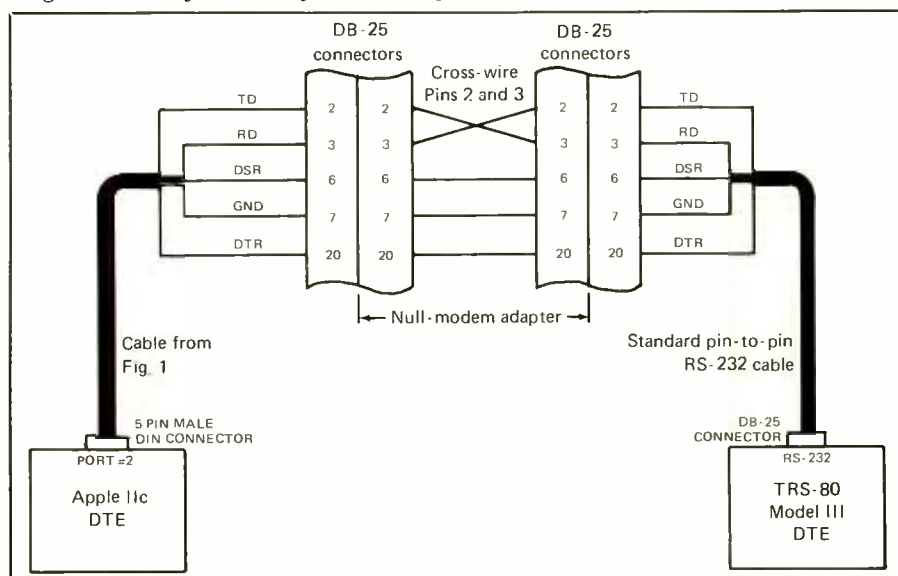
Unfortunately, as is so common with Apple equipment, the IIc departs from the norm, with a default serial port setting of two stop bits. My advice is to leave the IIc settings alone, since they are inconvenient to change with some IIc terminal programs. Change the other computer, if necessary—and it might not be, since many computers ignore the difference between one and two stop bits.

Baud rate must be matched. There should be no problem communicating at 1200 baud over the phone lines, but use 300 baud if a lot of errors interrupt program transfer. Furthermore, some early Apple IIcs have a problem with 1200-baud transmission, due to a design oversight. To communicate at 1200 baud, the modems at both ends must be designed for that protocol, or 300 baud must be used. Fortunately, the default for IIc's Port No. 2 is 300 baud.

Step-By-Step

Perhaps the best way to describe the procedure for transferring a BASIC

Fig. 3. Details for directly connecting two computers, both acting as DTEs.



program is to “walk” you through an actual example, using the TRS-80 Model III as the source computer. Remember that you would use a similar procedure with another computer. While the following describes data transfer using direct wiring, transfer using modems is essentially the same.

(1) ASCII. If you are planning to transfer just text or data (not programs), this step is not necessary. However, if you are going to send a BASIC program, you can eliminate a lot of problems by first converting it to ASCII format.

This is easy to do on most microcomputers, since a program can be saved in either tokenized form or in ASCII. Unfortunately, the Apple does not provide this convenience. (A method to do this will be covered later in this article).

I loaded into the TRS-80 Model III's memory the BASIC program I wanted to transfer to my Apple IIc and then saved it in ASCII with the SAVE“PROGRAM/ASC”, A command.

(2). Terminal Programs. With the program to be transferred in ASCII format, I got the terminal programs up and running in both computers. I used OMNITERM in the Model III and EasyCom/EasyGo in the IIc. (OMNITERM sells for \$95 from Lindberg Systems, 41 Fairhill Rd., Holden, MA 01520. EasyCom/EasyGo sells for \$129 from Transend Corp., 2190 Paragon Dr., San Jose, CA 95131.)

I used a direct cable connection between the two computers. The procedure to make the transfer would be the same with modems, except for the additional task of setting one modem to “originate” and the other to “answer,” and making the call from one to the other to establish the link.

(3) Setting Parameters. With OMNITERM running in the Model III, I used the various menus (on-screen choices) to set the computer's trans-

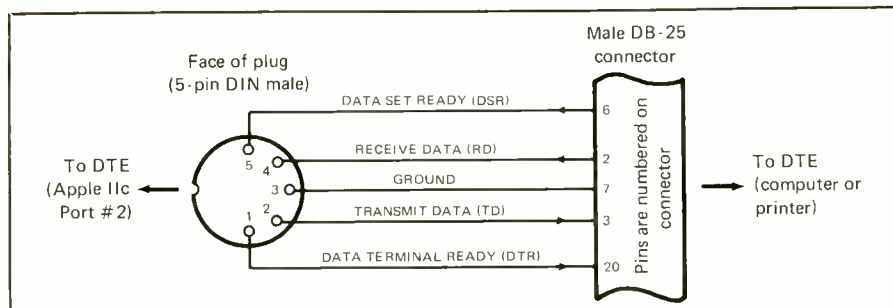


Fig. 4. This drawing shows how a printer cable can be used to eliminate the need for a null modem adapter.

fer rate at 300 baud, since that is the default setting for the IIc's terminal program. You can get fancy after you have learned to do transfers, but I suggest that you leave everything “plain vanilla” at the start.

(4) File Transfer. Next, I used the appropriate commands in OMNITERM to load PROGRAM/ASC into the Model III's buffer, where it is resided waiting to be transferred.

I then put the Apple IIc into the terminal mode so that it could receive and display incoming text. Returning to the Model III, I selected the OUTPUT FROM BUFFER function. It was a pleasure to watch the IIc screen as the program zipped over to it through the cable coming from the Model III. In a short time, each screen indicated that program transfer had been completed successfully.

Some terminal programs will automatically save this incoming program as a TXT (text) file. Others will save it in a buffer for you to modify and then save to disk—still as a TXT file. In any case, you cannot use this file as a program until it is converted to a BASIC program file; otherwise, it will not load into BASIC.

(5) File Conversion. Some terminal programs for the IIc have provision for converting an ASCII text file to a BASIC file. If you have such a conversion program, and learn how to use it, fine. However, there is a very simple and effective alternate way to do this:

(A) Save the incoming program on

a IIc disk, using whatever means your terminal program provides.

(B) Get out of the terminal program and into Applesoft BASIC.

(C) Type “EXEC filename” (without the quotes), using the disk filename you assigned the incoming program when you capture it from the source computer.

(D) The disk drive will load (but not run) the text file into memory just as if you had typed it in from the keyboard. You will see only right-bracketed symbols (carriage returns) on the screen at this point, ending with a blinking cursor.

(E) Do not attempt to run the program yet. Simply save it on disk with another filename, and it will be saved as a BASIC program.

(F) Now you can attempt to run it and modify it any way you please, since it is in its “virgin” form on-disk under a new filename, in case it crashes during translation. If it was originally an Applesoft program, it might run as-is. If not, you might have to make extensive revisions to make it run.

Transferring Apple IIc BASIC

You may have occasion to transfer an Apple IIc's Applesoft BASIC program to another computer. Here, again, plan on using ASCII for the transfer. Although many terminal programs support the Christensen protocol, a remarkably effective “error-free” transmission method,

ASCII transfers are more universally utilized.

To convert an Applesoft BASIC program into ASCII text format, follow these steps, suggested by Harold H. Stuart of Sun Data Software:

(A) In Applesoft BASIC, load the target program into memory. Then list it and note the lowest program number. If this number is less than or greater than 10, retype and/or renumber the lines so that the program starts with line number 10.

(B) Using the keyboard, type in the listing:

```
1 D$ = CHR$(4)
2 HOME : POKE 33,33
3 PRINT "CONVERT BASIC FILE TO
  TEXT FILE."
4 PRINT : INPUT "NEW FILE
  NAME? ";F$
5 PRINT D$;"OPEN";F$
6 PRINT D$;"WRITE";F$
7 LIST 10 -
8 PRINT D$;"CLOSE"
9 POKE 33,30: END
```

This short program is designed to create a text file of your BASIC program, starting with program line 10.

(C) Run the new integrated program. The screen will clear, announce its purpose, and prompt with "NEW FILENAME?" Type in a different filename from the original program name so the new ASCII file to be created will not destroy the original BASIC file.

(D) After entering the new filename, press RETURN to write a new file to the disk in ASCII text format.

(E) This new file can then be transmitted with your terminal program and be converted back to BASIC by the target computer. For an Apple, the EXEC command described above does the job. With other companies, this conversion is frequently done with a word processor.

EXEC ASCII File

If you plan to do a lot of the foregoing work, you will not want to type in the above listing for every conversion. Here is a neat way to create an

ASCII version of the listing, suggested by Jeffrey Mazur:

(A) While in BASIC, type NEW and hit RETURN. Then type in lines 1 through 9 from above.

(B) Add the following:

```
10 D$ = CHR$(4)
20 PRINT D$;"OPEN ASCII"
30 PRINT D$;"WRITE ASCII"
40 LIST 1-9
50 PRINT D$;"CLOSE"
```

(C) Check the above carefully. Then type RUN 10 and hit RETURN. Lines 10 through 50 will create a text file of lines 1 through 9, with a filename of ASCII.

(D) To use this file, first load in the BASIC file you want converted to ASCII. Make sure the starting line number of the program is 10. If it is not, change it so that it is. Then type EXEC ASCII and hit RETURN to add lines 1 through 9 to the beginning of

your BASIC program. This puts you right where you want to be, in the position to run the new integrated program.

A Lot Of Work?

This may seem like a lot of work just to transfer a BASIC program, and it is. It is also a lot *more* work to hand key in a long program, not to mention the extra time you will have to spend finding all your typing errors!

Though the transfer of programs to or from an Apple computer is inherently more complicated than it is with most other computers (since there are no direct ASCII conversions), once you learn the procedure you can save a lot of time, energy and frustration. Of course, you will still face the challenge of translation if the program is not Apple IIc-compatible. But that is another story. **ME**



DON'T LET THE ELECTRONICS REVOLUTION PASS YOU BY!

SUBSCRIBE TO



Now you can reach out every month and learn about the latest innovations in consumer electronics and personal computers—what's new, how products and circuits work, and what leading authorities think of them.

***Stereo *Video *Home & Car Electronics
*Test Instruments *Circuitry & Devices *Servicing
*Personal Computers *Communications *Construction Projects**

In every issue, authors lucidly explain how exciting, new electronics ideas are applied . . . issue forthright buying counsel . . . present complete plans to build electronic devices to enhance products, even unique ones that are not yet being sold in the marketplace.

Just fill out the coupon and mail with your check, money order, or credit card information (Visa or MasterCard) to enter your subscription. Do it today to ensure earliest delivery!

MODERN ELECTRONICS 76 North Broadway, Hicksville, NY 11801

Please send me Modern Electronics for ☐ 2 Years ☐ 1 Year

Name _____
Street _____
City _____
State _____ Zip _____

Charge My Order To: Payment Enclosed \$ _____ 2 Years (24 issues) \$22.97

☐ MasterCard  My account number is: 1 Year (12 issues) \$12.97

☐ VISA

| | | | | | | | | | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|