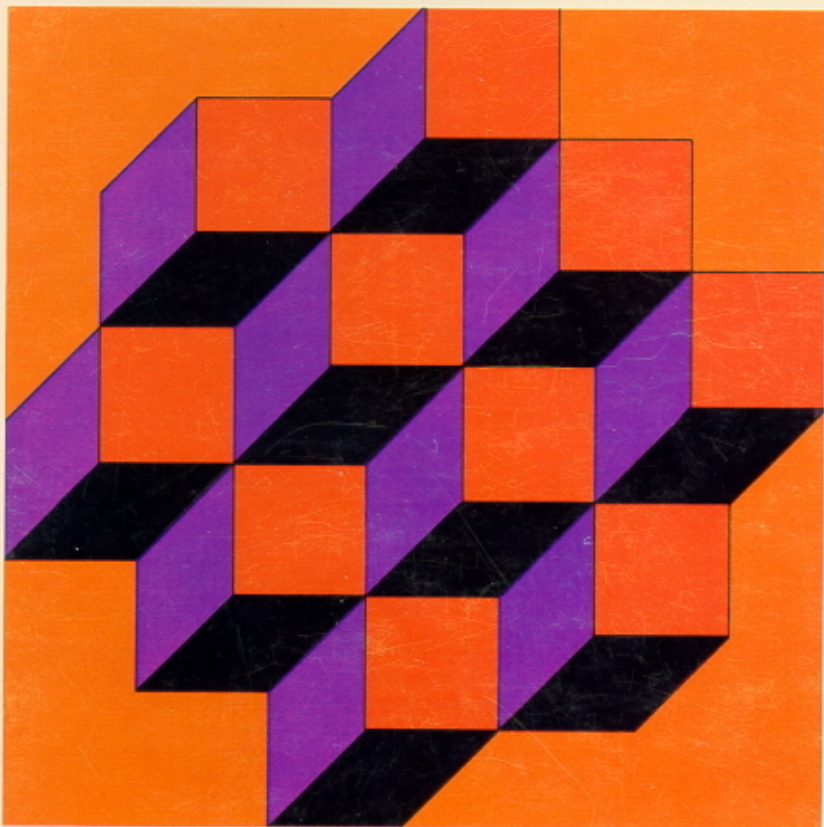


Apple II



Apple Language Card

Installation and Operation Manual



Apple II

Apple Language Card

Installation and Operation Manual

CHAPTER 2

INSTALLATION

CHAPTER 3

USING THE LANGUAGE CARD

CHAPTER 4

THE AUTOSTART ROM

PLEASE READ THIS MANUAL BEFORE ATTEMPTING TO INSTALL THE LANGUAGE CARD IN THE APPLE. INCORRECT INSTALLATION COULD CAUSE PERMANENT DAMAGE TO BOTH THE LANGUAGE CARD AND THE APPLE.

TABLE OF CONTENTS

INTRODUCTION v

CHAPTER 1

SYSTEM REQUIREMENTS 1

CHAPTER 2

INSTALLATION 3

CHAPTER 3

USING THE LANGUAGE CARD 7

- 7 Using the Extra Memory
- 8 Using DOS 3.3 and BASIC
- 8 Using 13-Sector Diskettes
- 9 Using Other Languages

CHAPTER 4

THE AUTOSTART ROM 11

- 11 The Autostart Features
- 12 The RESET Key
- 12 The BASIC-Editing Features
- 18 The Stop-List Feature

- 21 Appendix A: Summary of Autostart ROM BASIC Features
- 22 Appendix B: Autostart ROM Technical Information
 - 22 Subroutines Added to the Autostart ROM
 - 23 Subroutines Deleted
 - 24 Subroutines Changed
 - 24 Limitations of the Autostart ROM
- 25 Appendix C: Memory Map of Language Card
- 26 Appendix D: Language Card Control Codes
- 28 Appendix E: Functional Block Diagram of Language Card

RADIO AND TELEVISION INTERFERENCE

The equipment described in this manual generates and uses radio frequency energy. If it is not installed and used properly, that is in strict accordance with our instructions, it may cause interference to radio and television reception.

This equipment has been tested and complies with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules. These rules are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation.

You can determine whether your computer is causing interference by turning it off. If the interference stops, it was probably caused by the computer. If your computer does cause interference to radio or television reception, you can try to correct the interference by using one or more of the following measures:

- Turn the TV or radio antenna until the interference stops.
- Move the computer to one side or the other of the TV or radio.
- Move the computer farther away from the TV or radio.
- Plug the computer into an outlet that is on a different circuit from the TV or radio. (That is, make certain the computer and the TV or radio are on circuits controlled by different circuit breakers or fuses.)

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find the following booklet prepared by the Federal Communications Commission helpful:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, DC 20540, Stock number 004-000-00345-4.

CHAPTER 1

SYSTEM REQUIREMENTS

The Language Card should be used only on an Apple II or Apple II Plus containing 48K of Apple Random-Access Memory (RAM) on its main printed-circuit board. If your Apple does not contain 48K of memory, you need to install the proper amount of additional memory to bring the Apple's memory up to 48K. Parts and instructions for adding RAM in increments of 16K may be obtained by ordering one or more Apple 16K Memory Expansion Modules (Apple Product No. A2M0016). Your Apple dealer can tell you how to obtain what you need.

Although you can use the Language Card without it, many of the languages and applications that take advantage of the extra 16K of memory also require at least one Disk II drive and disk controller card. The Language Card will work with up to three controller cards, each running one or two drives.

If you're using the Disk II drive, we recommend that your Apple be set up for "16 sector format" -- a particular way in which data is arranged on a diskette. You can use the Language Card with DOS 3.2 or DOS 3.2.1, which use 13 sector format, but you won't be able to use the many application programs that are in the 16 sector format and utilize the extra memory provided by the Language Card. You cannot use the Language Card with versions of DOS earlier (lower-numbered) than 3.2.

To set up your Apple for 16 sector format, you should obtain DOS 3.3 (APPLE Product number A2D0023). This product contains everything you need to convert your Apple to a 16 sector system.

* * * WARNING * * *

Apple Computer cannot guarantee your Apple's compatibility with memory ICs from other vendors. Use of ICs from other vendors is not recommended and may void your warranty.

* * * * *

containing 400 of Apple's random access memory (RAM) in its main memory circuit board. If your Apple does not contain 400 K of memory, you need to install the memory of additional memory to bring the Apple's memory up to 400 K. Parts and instructions for adding RAM to Macintosh are in the "Parts and Instructions for Adding RAM to Macintosh" manual. It may be obtained by ordering one of Apple's Apple II family Expansion Module Apple Product No. A120011. Your Apple dealer can tell you how to obtain what you need.

The Language Card should be used only on an Apple II or Apple II Plus containing 400 K of Apple's random access memory (RAM) in its main memory circuit board. If your Apple does not contain 400 K of memory, you need to install the memory of additional memory to bring the Apple's memory up to 400 K. Parts and instructions for adding RAM to Macintosh are in the "Parts and Instructions for Adding RAM to Macintosh" manual. It may be obtained by ordering one of Apple's Apple II family Expansion Module Apple Product No. A120011. Your Apple dealer can tell you how to obtain what you need.

Although you can use the Language Card without it, any of the programs and applications that take advantage of the extra 100 K of memory also require at least one High II Drive and Disk Controller Card. The Language Card will work up to three controller cards, each handling one or two drives.

If you're using the Disk II drive, we recommend that you write to see us for "The Sector Format" -- a particular way to write data is required on a diskette. You can use the Language Card with the 5.25 or 8" Disk II, which use 13 sector format. But you won't be able to use the many application programs available in the 16 sector format and within the extra memory provided by the Language Card. The format was the Language Card with versions of ROM earlier (lower numbers) than 1.1.

To see if your Apple has 16 sector format, you should obtain the 1.1 Apple Product number A120011. This product contains instructions on how to convert your Apple to a 16 sector system.

* * * * *

Apple Computer cannot guarantee that Apple's compatibility with other 16 bit other vendors. Use of the 16 bit other vendors is not recommended and may void your warranty.

* * * * *

CHAPTER 2

INSTALLATION

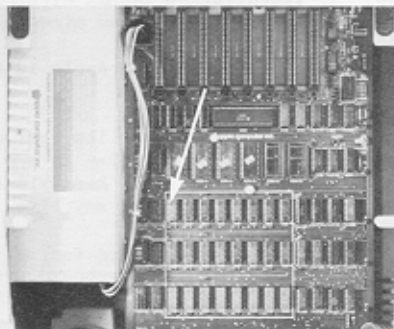
It is easy to install the Language Card, but it is very important to do it carefully. Improper installation can damage the Language Card and the rest of the Apple. Read the instructions up through removing the Apple's cover, remove it, then read the remaining instructions before doing anything more. If installation sounds difficult, ask your dealer to help you do it. It will take only a few minutes.

Before doing anything else, make sure the Apple is turned OFF. The POWER light should not be glowing.

Remove the Apple's cover by pulling up on its rear edge until the two corner fasteners pop apart. Without lifting it any further, slide the cover backward (away from the keyboard). Read all the following instructions before removing or inserting anything.

Look into the open Apple, with the keyboard facing you. On the main printed-circuit board, right behind the keyboard, you will see a white-outlined square, about four inches on each side. This white square contains your Apple's memory: 24 RAM integrated circuits (ICs, or "chips"). Each of these ICs is marked with a lot of numbers, including ones like MK4116, MCM4116, UPD416 or HM4716; and the board beneath it is marked 'RAM'. Each of these ICs has a notch on the end toward the keyboard. The Language Card's small plug will eventually replace the IC in the left rear corner of the square (socket E3), but read all the instructions before attempting to do this.

Across the rear of the circuit board, you will see a row of eight long, narrow edge-connector receptacles, or "slots", green outside (on most Apples), with gold-plated contacts inside. If you lean over the slots, you will see that they are numbered at the far end, on the board. The Language Card will fit into slot 0, the leftmost one. Read the following instructions for details about how to do this. (Your Apple cannot use the Applesoft II Firmware Card and the Apple Language Card at the same time. An Apple with the Language Card will use the same version of Applesoft as one with the Applesoft Card-- not the more limited "cassette version" of Applesoft--but will load it from the DOS System Master diskette when it is booted.)



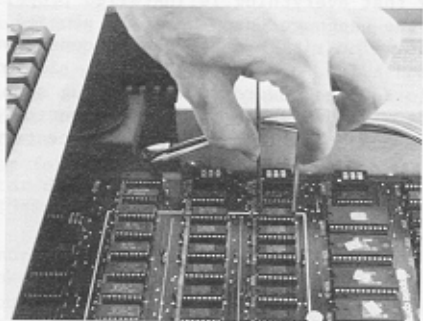
Touch the power-supply cover (the big silver- or gold-colored box along the left side) to discharge any static charge you may be carrying. You need not worry about getting a shock if your Apple is properly grounded, as all high voltages are well insulated inside the grounded power-supply case. (Besides, you have already turned the Apple off, haven't you?)

Find slot 0, the leftmost slot. If there is a card in the slot, remove it by grasping the end nearest you and moving it up and down gently until the card comes out.

Find the white square containing the RAM ICs. In this square, locate the IC in the left rear corner (the RAM IC closest to slot 0). Remove this IC, using the IC puller that came with the Language Card. This tool is U-shaped, with a prong on each of its two "arms". As you look at the IC and its socket from either narrow end, you will see a small gap between the IC and its socket. Use the puller like tweezers, grasping the IC from directly above, fitting the two prongs into the two gaps between the IC and the socket (one at each narrow end of the IC). Holding the IC firmly enough that it doesn't slip, wiggle the puller slightly back and forth (toward the slots, then toward the keyboard). The top of the puller should move only a fraction of an inch. If you pull upward gently as you wiggle it, the IC should come loose. As it starts to move, wiggle the IC even more gently, and relax your upward tension, making sure the IC doesn't tilt more than a few degrees. Both ends and both sides should come free at once. If they

don't, shift the puller slightly to loosen the part that is stuck. When it has been successfully removed, the IC should be stored in a safe place, such as the box in which the Language Card was packed.

If you can't remove the IC safely (for example, if it's hard to remove because it's been in your Apple for a long time) see your dealer. This will protect the IC, your warranty, and your nerves.



Now you are ready to install the Language Card. First examine the Language Card. Set it in front of you so that the lettering is right-side up. At the bottom left corner you will see a ribbon cable, with one edge colored differently from the rest of the cable. The cable ends in a plug, which will replace the IC you have just removed. The plug has three black plastic prongs, two round and one rectangular, which will make it hard to put it in wrong. You will find an edge-connector tongue protruding from the bottom edge of the plug at the right end. This tongue will go into slot 0.

Hold the Card in your right hand, IC side down and cable end toward you. Set the Card down gently on the board and place the plug over the left rear RAM socket, now empty, making sure that the differently colored edge of the ribbon cable is toward the keyboard and that all the pins are aligned with their holes. Each pin should have a hole, and vice versa. Press down gently on the plug: it should seat (settle into the correct position) easily. If it won't go in, check to see that the plug is facing the right way and is lined up properly.



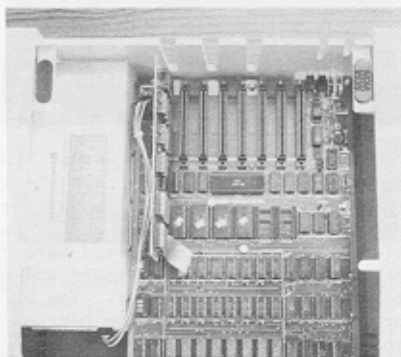
If the RAM socket in your Apple does not have two round holes, the plug will not seat properly. Take your Language Card to your dealer and have him or her remove the two round black plastic prongs on the plug. An alternate solution is to have the dealer insert a second RAM socket -- this one with the correctly-placed two round holes -- into the RAM socket in your Apple, piggy-back style. Then you can insert your language card plug into the socket on top.

If the prongs on the plug are lined up with their holes on the socket, the pins will go into their own holes. The cable should come out of

the right side of the plug, with the differently colored edge toward you.

Insert the Card's edge-connector tongue into the leftmost slot, slot 0, by pressing down gently and wiggling it slightly until the Card is firmly seated.

Now check carefully that everything is installed properly. As you look over the keyboard, the Language Card should be plugged firmly into slot 0, the leftmost slot. The ribbon cable should come down from the rear end of the Card, bend twice, and enter the right side of its plug, with the differently colored edge toward you. The plug should be firmly inserted in the left rear RAM socket (in the corner of the white-outlined square on the main board) and should be centered in its row and column.



When you are satisfied that the Language Card is installed properly, you can replace the Apple's cover and go to the next chapter.

CHAPTER 3

USING THE LANGUAGE CARD

To make full use of the capabilities of the Language Card, we recommend that you use it only with DOS 3.3 and later (higher-numbered) versions of DOS. DOS 3.3 uses the 16 sector format, which allows you to store more information on a diskette than the 13 sector format. It also provides you with whichever BASIC language was not part of your system. Finally, many of the application programs and languages you may want require 16 sector format. With DOS 3.3 you will still be able to use your old 13 sector diskettes by following the directions in the section, Using 13 Sector Diskettes.

If you already have DOS 3.2 or 3.2.1 and don't want to obtain DOS 3.3, you can still take advantage of the additional 16K bytes of memory provided when you install the Language Card. You just won't be able to use any language or program requiring 16 sector diskettes.

Before getting into procedures, let's make sure we're using the same terms and the same setup. If you have a Disk II drive, it should be plugged into the Drive 1 connector pins of the Disk II Interface Card, and the Interface Card should be in slot 6. If you have more than one disk drive, install the drives and the Interface Cards as explained in your DOS manual. Every system should have a Disk II in the drive 1, slot 6 position. In the following discussion, if the drive number is not specified, it is assumed to be drive 1. If the slot number is not specified, it is assumed to be slot 6.

The Apple II comes in two flavors: Apple II (standard) and Apple II Plus. The standard Apple II has Integer BASIC in Read-Only Memory (ROM) on the main printed-circuit board. The Apple II Plus has Applesoft BASIC in ROM on the main board. The language in ROM is the language the Apple displays when it is turned on, if it has no disk drives or accessory cards. As the Apple BASIC manuals describe, a > on the screen indicates that your Apple is in Integer BASIC, and a) means your Apple is in Applesoft BASIC.

USING THE EXTRA MEMORY

Some application programs contain special statements to determine if the Language Card is installed in the system and automatically use the extra memory provided there. Programs that do not contain these special statements do NOT automatically use the extra memory. Your dealer can tell you which application programs can use the extra memory provided.

If you're planning to write your own application programs to take advantage of the extra memory, be sure to review the memory map and the Language Card control codes in the Appendices. Special memory mapping techniques are needed to write programs for the Language Card. These techniques are not difficult to use, but you need to understand them before you start to write your program.

USING DOS 3.3 AND BASIC

To use DOS 3.3, simply place the System Master diskette in drive 1 and turn on your Apple's power. The System Master diskette now automatically loads either Applesoft or Integer BASIC -- whichever is NOT available on your main board -- into the storage space available on your Language Card. If you have a standard Apple II, when the System Master is booted, Applesoft is loaded into your Language Card. If you have an Apple II Plus, Integer BASIC is loaded. In either case, a message appears, telling you which basic is being loaded during the booting of your System Master. All of the DOS commands now work when using and writing programs in both Applesoft and Integer BASIC.

The Apple can now run any program in either BASIC. Once you have DOS running, you can switch from one BASIC to the other by using the INT and FP commands, as explained in the DOS manual. As always, you can tell which BASIC you are in by the prompt you see on the screen: > for Integer BASIC and | for Applesoft BASIC.

You can reboot DOS without turning the power off. Simply insert the System Master diskette, type PR#6 or IN#6 and press RETURN.

USING 13 SECTOR DISKETTES

The "BASICS" diskette that came with the DOS System Master can be used to boot 13 sector diskettes on your 16 sector system. Versions of DOS earlier than 3.3 are 13 sector DOS.

To use the 13 sector version of either Integer or Applesoft BASIC, insert the BASICS diskette into the drive, and turn the Apple on. In about five seconds the screen displays

```
INSERT YOUR 13 SECTOR DISKETTE  
AND PRESS RETURN
```

Insert any 13 sector DOS/BASIC diskette and press RETURN. The Apple will now behave as described in the DOS manual, except that one step is added to the boot (startup) procedure:

Whenever you turn the Apple on, or type PR#6, IN#6, C600G, or 6CTRL-P, and you want to boot a 13 sector diskette, the BASICS diskette must be in the drive.

When the cue appears on the screen, insert the 13 sector DOS/BASIC diskette you wish to boot. If you are in either 13 sector BASIC and you want to continue in 13 sector, you must have BASICS in the drive each time you type PR#6 or IN#6, and then reinsert your previous diskette.

Once you have DOS running, you can switch from one BASIC to the other by using the INT and FP commands as you normally would.

To update your data diskettes to 16 sectors, use the MUFFIN program as described in The DOS Manual. Under some circumstances, a diskette converted by MUFFIN may be un-runnable. This happens when the program on the diskette expects to find instructions at specific locations on the diskette itself. Most programs aren't like this, but a few are. Remember, MUFFINING a diskette won't harm your 13-sector diskette, and you can always boot 13 sector diskettes using the BASICS diskette.

USING OTHER LANGUAGES

To use Pascal, FORTRAN, or another language that requires the Language Card, you must obtain the appropriate software package. The manual supplied with the package contains complete directions for using that language with the Language Card.



Before switching from one language to another on the Language Card, first SAVE any program you may be working on, as it will be erased when you switch. Be sure to consult the manual supplied with each language for the exact startup procedure required.

USING DOS 2.0 AND BASIC

To update your data diskette to DOS 2.0, you must use the **UPDATE** command. The **UPDATE** command is located in the **DISK** menu. To use the **UPDATE** command, you must first format the diskette with DOS 2.0. To format a diskette with DOS 2.0, you must use the **FORMAT** command. The **FORMAT** command is located in the **DISK** menu. To use the **FORMAT** command, you must first insert the diskette into the floppy disk drive. Then, you must type the command **FORMAT** at the prompt. The **FORMAT** command will format the diskette with DOS 2.0. After the diskette is formatted, you can use the **UPDATE** command to update your data diskette to DOS 2.0.

USING OTHER LANGUAGES

Apple II computers can use a variety of languages. The most popular are BASIC, PASCAL, and FORTRAN. Each language has its own set of commands and syntax. To use a language other than BASIC, you must first load the language into the computer. This is done by typing the name of the language at the prompt. For example, to load BASIC, you would type **BASIC**. Once the language is loaded, you can use it to write and run programs. The Apple II computer is designed to be easy to use, and it can handle a wide variety of languages. This makes it a popular choice for students and hobbyists alike.



USING 13 SECTOR DISKETTES

The "BASICS" diskette that comes with the Apple II system is a 13 sector diskette. This means that it can hold 13 sectors of data. The Apple II system is designed to use 13 sector diskettes. To use a 13 sector diskette, you must first format it with the **FORMAT** command. The **FORMAT** command is located in the **DISK** menu. To use the **FORMAT** command, you must first insert the diskette into the floppy disk drive. Then, you must type the command **FORMAT** at the prompt. The **FORMAT** command will format the diskette with 13 sectors. After the diskette is formatted, you can use it to store and retrieve data.

To use the 13 sector diskette, you must first format it with the **FORMAT** command. The **FORMAT** command is located in the **DISK** menu. To use the **FORMAT** command, you must first insert the diskette into the floppy disk drive. Then, you must type the command **FORMAT** at the prompt. The **FORMAT** command will format the diskette with 13 sectors. After the diskette is formatted, you can use it to store and retrieve data.

INSERT THE 13 SECTOR DISKETTE
AND PRESS RETURN

Insert one 13 sector DISK BASIC diskette and press RETURN. The Apple II will now behave as if it had a 13 sector diskette in the floppy disk drive. You can now use the **FORMAT** command to format the diskette with 13 sectors. The **FORMAT** command is located in the **DISK** menu. To use the **FORMAT** command, you must first insert the diskette into the floppy disk drive. Then, you must type the command **FORMAT** at the prompt. The **FORMAT** command will format the diskette with 13 sectors. After the diskette is formatted, you can use it to store and retrieve data.

Whenever you load the Apple II, use type BASIC, PASCAL, FORTRAN, or GCN-9, and the user is sent a 13 sector diskette, the BASICS diskette must be in the drive.

CHAPTER 4

THE AUTOSTART ROM

When you install the Apple Language Card in your Apple II, your computer may gain some new features that make several things easier. With these new features, it is easier to edit and use BASIC programs, and to use Disk II drives. This is because the Language Card, in addition to containing an extra 16K of write-protectable memory, also contains the "Autostart ROM".

The Autostart ROM is the largest IC on the Language Card, and it automatically takes over from the Monitor ROM IC that your Apple may contain in the socket labelled "ROM-F8" on the main printed-circuit board. This IC controls your Apple's behavior at the most elementary level, determining features used by machine-language programs and higher-level languages of all kinds.

If your computer already contains the Autostart ROM, you will be familiar with its features and will not notice any change after installing the Apple Language Card. But if your Apple contains the "old" Monitor ROM in socket ROM-F8 on the main board, read the following sections to discover your computer's new powers.

THE AUTOSTART FEATURES

First of all, as its name implies, the Autostart ROM helps get the Apple started and allows it to act as a "turnkey" system. This means that the Apple can begin executing a particular program the instant the power is turned on. It requires no special effort or knowledge on the part of the user to begin a program which has been properly set up for turnkey operation.

When you turn on the Apple with the Language Card installed, the message APPLE II is displayed. Then the diskette in drive 1, slot 6 is booted, if it is the DOS 3.3 System Master or a diskette containing the essential startup (boot) files of a language which can use the Language Card. Then your diskette's greeting (HELLO) program, if any, is run. (If you removed an Applesoft II Firmware Card to install the Language Card, your Apple will now load the same Applesoft from the System Master.)

Usually, everything will work just as described above, the first time you turn on your Apple with the Language Card installed. If there is

any problem, it may result from one of the following conditions:

1. If the Disk II Interface Card in slot 6 has nothing connected to drive 1, the Apple will just sit there until you press the RESET key.
2. If drive 1 is connected but has no diskette in it, or has an open door, the Apple will spin the drive until the end of time, or until you press the RESET key, whichever comes first.
3. If the Apple tries to boot a diskette that cannot be booted, it will keep spinning the diskette forever, and will try to boot it again each time you press the RESET key. You can stop it by pressing the RESET key several times in rapid succession, or by turning the Apple off.
4. If the Apple finds no Interface Card, or if it tries unsuccessfully to boot and you press the RESET key, the Apple will go into whichever Apple BASIC is installed in your computer, but will not have DOS and will act as if it had no disk drive.

THE RESET KEY

If you turn the Apple off and then on, it will start up as described above. Any program in memory will be lost whenever the Apple is turned off. (Turning the Apple off and on very quickly will have odd results, as the contents of memory will be only partially cleared.)

If you press the RESET key while in BASIC, the Apple will stop whatever program it may be running, and display the prompt for whatever BASIC you were in, without losing your program or DOS.

If you are in either BASIC with the Autostart ROM in control, the RESET key has a function similar to CTRL-C (produced by typing C while holding down the CTRL key). Both of these will stop the program running and allow you to run it again. In addition, RESET restores the computer to the state it was in before the program was first run. CTRL-C does not restore all numbers in memory to their starting values, so when you RUN the program again it may not do the same things it did when it was first run.

Note: Assembly-language programmers may wish RESET to leave them in monitor mode. See the description of SETPWC in "Subroutines Added to the Autostart ROM" in Appendix B.

THE BASIC-EDITING FEATURES

The Apple Language Card's Autostart ROM makes it easier to edit your BASIC programs. With the Autostart ROM in control, it becomes very

easy to move the cursor around on the screen. Look at the I, J, K and M keys on the keyboard. They look like this:



Imagine four arrows drawn on them (or make little labels and actually stick them on).



To begin moving the cursor in the direction of one of the arrows, first press and release the key marked "ESC" (which stands for "ESCAPE") at the left edge of the keyboard. Pressing the ESC key once puts the Apple into what we call "edit mode". With your Apple in edit mode, you can then use the four cursor-moving keys. Type I to move the cursor up, M to move it down, J to move it left and K to move it right.

To move the cursor repeatedly, just keep typing the cursor-moving keys. For faster cursor motion, hold down one of the cursor-moving keys (I, J, K or M) and then hold down the key marked "REPT" (for "REPEAT") at the same time. The cursor will glide up, left, right, or down.

If the cursor reaches the top of the screen, it will stop. If it reaches the bottom of the screen, the cursor will stop, and the lines will start to move upward. If it reaches the right edge, the cursor will disappear, reappearing at the left edge, but on the next line down. If it reaches the left edge, the cursor will reappear at the right, one line up.

To leave edit mode, press the space bar once. The Apple will now be in what we call "normal mode", the mode it was in before you pressed ESC.

Once you have pressed ESC to begin cursor moves in edit mode, typing almost any key except I, J, K, and M will cause the Apple to leave edit mode. This means that you cannot move the cursor, type new information, and move the cursor again, all in edit mode. You must leave edit mode, by pressing the space bar once, before starting to type normally.

From edit mode, it is also possible to move the cursor one space and leave edit mode in a single keystroke. This is done by pressing the ESC key to enter the edit mode, and then typing A (to move right), or B (to move left), or C (to move down), or D (to move up). After any of these moves the Apple is immediately returned to normal mode, and the ESC key must be pressed again before another move can be made. These are the cursor moves familiar to users of the "Old Monitor ROM".

How to clear one or more characters from a line on the screen. In edit mode, move the cursor onto the first character. The character is now blinking. Then press the space bar to get the Apple out of edit mode, and press the space bar once again for each character you wish to clear. Each of these characters will be replaced by a space.

How to clear to the end of a line on the screen. In edit mode, move the cursor onto the first character you wish to erase, then press E. The blinking character and all the characters after it on the line will disappear, and the Apple will return to normal mode.

How to clear to the end of the screen. In edit mode, move the cursor onto the first character you wish to erase, then press F. The blinking character and all the characters after it on the screen will disappear, and the Apple will return to normal mode.

How to clear the whole screen. If you are in edit mode, type @. All the characters on the screen will disappear, the cursor will return to the top left corner ("home"), and the Apple will return to normal mode. To get a BASIC prompt (> or |), press RETURN. If you are not in edit mode and wish to clear the screen, first press the ESC key, then type @.

In edit mode, when you press I, J, K or M the Apple remains in edit mode (because you may want to continue moving the cursor). But if you press A, B, C, D, E, F or @, the Apple leaves the edit mode immediately after carrying out that command. Pressing the space bar also returns the Apple to normal mode.

All these features affect only what is displayed on the screen, not what is stored in memory. Two other keys allow you to change what is in memory, and are very useful in editing programs. These are the left-pointing and the right-pointing arrow keys, which can also be called the Delete and Retype keys.



These keys affect only the line currently being typed. When you press the Delete (left-pointing arrow) key, the cursor moves back (left) one space, erasing the character last typed. If the last character you typed appeared on the screen, it will be deleted. If you last typed a non-printing control character, the Apple will behave as if it had never been typed, but will not change the display. If the last character typed was a pure cursor move (I, J, K or M in edit mode), that move would not be the last character entered, and would never have been part of the current line.

When you press the Retype (right-pointing arrow) key, the cursor moves forward (right), copying the character it just passed over. If you copy a line with the Retype key, then press the RETURN key, the Apple behaves exactly as if you had retyped the line by hand.

You can "fast-reatype" by holding down the Retype key, then the REPT key. The Apple will retype all characters and spaces over which the cursor passes, until one of the two keys is released. Any time you use the REPT key, remember that it repeats the last key typed before the REPT key was pressed. For example, if you press the space bar to leave edit mode, press and hold the REPT key, and then press the Retype key, the Apple will delete the blinking character (as if you had pressed the space bar again). It will not retype the blinking character as you may have intended. This can be disconcerting if you do not understand what is happening. Always remember to press the REPT key after holding down the key you wish repeated.

You can use these features to replace, insert, or delete lines or parts of lines in your program. For example, let's look at a simple program in Applesoft:

```
10 PRINT "JABBERWOCK"
```

To replace 'JABBER' with 'JIBBER', you can do as follows:

1. LIST the program on the screen.

```
]LIST  
10 PRINT "JABBERWOCK"  
]
```

2. Press ESC, then move the cursor onto the very first character (the digit '1') of line 10, using the J and I keys.

```
10 PRINT "JABBERWOCK"
```

3. Press the space bar to leave edit mode, then fast-reatype the line up through the 'J' of 'JABBERWOCK', using the Retype and REPT keys. The blinking cursor should cover the 'A'.

```
10 PRINT "JABBERWOCK"
```

3a. If the 'l' disappeared when you tried to fast-reatype, you pressed the REPT key before pressing the Retype key. Remember, the REPT key must be pressed after holding down the key you wish to repeat. To restore it, back up the cursor to the beginning of the line with the Delete key, type in a 'l', then fast-reatype through "J".

4. Type 'I', then fast-reatype the rest of the line.

```
lØ PRINT "JIBBERWOCK"
```

5. Press RETURN, to tell the machine to replace the line in memory.

```
lØ PRINT "JIBBERWOCK"  
]
```

6. LIST the program to see what you have done:

```
]LIST  
lØ PRINT "JIBBERWOCK"  
]
```

To insert 'JABBER' after 'JIBBER', do this:

1. List the program:

```
]LIST  
lØ PRINT "JIBBERWOCK"  
]
```

2. Fast-reatype it through 'JIBBER'.

```
lØ PRINT "JIBBERWOCK"
```

3. Press ESC, then move the cursor back to 'J', using the edit-mode keys.

```
lØ PRINT "JIBBERWOCK"
```

4. Leave edit mode by pressing the space bar, then type 'JABBER' where 'JIBBER' was, using fast-reatype if desired. 'JABBER' will replace 'JIBBER' on the screen, but both words have now been typed into your current line.

```
lØ PRINT "JABBERWOCK"
```

5. Fast-reatype the rest of the line and press RETURN.

```
lØ PRINT "JABBERWOCK"  
]
```

6. LIST the program to see what you have done:

```
]LIST  
10 PRINT "JIBBERJABBERWOCK"  
]
```

To delete 'JABBER' from the current line, do this:

1. LIST the program:

```
]LIST  
10 PRINT "JIBBERJABBERWOCK"  
]
```

2. Fast-retchype it through 'JIBBER'.

```
10 PRINT "JIBBERJABBERWOCK"
```

3. Press ESC, then use the cursor-moving keys to pass the cursor over 'JABBER'. This will avoid retyping 'JABBER' into the current line, although 'JABBER' will remain on the screen.

```
10 PRINT "JIBBERJABBERWOCK"
```

4. Press the space bar to leave edit mode, then fast-retchype the rest of the line and press RETURN.

```
10 PRINT "JIBBERJABBERWOCK"  
]
```

5. LIST the program to see what you have done:

```
]LIST  
10 PRINT "JIBBERWOCK"  
]
```

Experiment with these methods, watching carefully what happens on the screen, then listing the line to see what happened to it in memory. After some practice you will become more familiar with the relationship between these two processes.

Here is a trick that may be helpful in editing programs. If you have tried using the fast-retchype feature on statements that took up more than one line on the screen, you may have noticed that unwanted spaces crept in. The Apple uses these spaces to make the listings appear on the screen in their usual, indented format. To avoid retyping these extra spaces into the current line, do this:

1. Clear the screen by pressing ESC, then typing @.

2. Type

POKE 33, 33

3. Press the RETURN key.

4. LIST the program.

This makes the text window on the screen 33 characters wide and ensures that a statement longer than 33 characters will "wrap around" to the very beginning of the next screen line, without indentation. Consequently, if you fast-retype the statement, the cursor will go to the end of one screen line and immediately jump to the beginning of the next, without picking up any extra blanks.

When you have done all the fast-retyping you need to do, type

TEXT

and then press RETURN to restore the text window to normal.

* * * WARNING * * *

Do not POKE a number larger than 40 into location 33 or you will clobber Applesoft and your program. (The Applesoft Reference Manual explains this in detail.)

* * * * *

THE STOP-LIST FEATURE

The Apple Language Card's Autostart ROM gives your Apple another useful feature: it lets you stop and start a program listing at will. You do this by typing CTRL-S when you wish to interrupt a listing, and typing CTRL-S again to continue. (CTRL-S is produced by typing S while holding down the CTRL key.) Let's try it.

1. LOAD a program of twenty lines or more from diskette or cassette, or type a new program at the keyboard.
2. Type LIST, but don't press the RETURN key yet.
3. With your left hand, press the CTRL key and hold it down. Place another finger of the same hand over the S key, ready to press it.
4. Now, with your right hand press the RETURN key: the program will begin listing on the screen.
5. Press the S key (are you still holding the CTRL key down?): the listing will stop.

6. Type CTRL-S again (or any other key except SHIFT, RESET or CTRL-C) to resume listing.
7. Press the RESET key to quit listing entirely.

This procedure takes a little practice. The Apple lists a program quickly, and it's easy to overshoot the line you want. That is why we suggest holding down the CTRL key while you are waiting for the right line.

The Stop-list feature will also suspend execution of any program that displays text on the screen. Let's try it on a simple program:

```
10 I = I + 1
20 PRINT I
30 GOTO 10
```

1. RUN the program.
2. Type CTRL-S. When the program sends a "Return" to the screen, to start a new line, the Apple looks at the keyboard. If it sees CTRL-S, it suspends execution of the program and waits for further instructions.
3. Type CTRL-S again (or any other key except SHIFT, RESET, or CTRL-C). The program will resume execution.
4. To stop the program and get back to BASIC, type CTRL-C or press the RESET key.

For some programs, CTRL-S will not stop the display. For example:

```
10 I = I + 1
20 PRINT I;
30 GOTO 10
```

This program differs from the previous one by the addition of a semicolon at the end of line 20. If you try to suspend this program by typing CTRL-S, The Apple will not respond. This is because this program sends a continuous stream of characters to the screen (because of the semicolon), rather than separate lines. It does not send any RETURNS to the screen, so the Apple never looks at the keyboard.

1. Press the NEXT key to start listening actively.

This procedure takes a little practice. The first time a program is run, and it's easy to overlook the line you want. That is why we suggest to take your first run while you are waiting for the signal. In the second run, the program will wait until you press the NEXT key. In the third run, you will be waiting for the signal. In the fourth run, you will be waiting for the signal. In the fifth run, you will be waiting for the signal. In the sixth run, you will be waiting for the signal. In the seventh run, you will be waiting for the signal. In the eighth run, you will be waiting for the signal. In the ninth run, you will be waiting for the signal. In the tenth run, you will be waiting for the signal.

2. Press the NEXT key to start listening actively.

3. Press the NEXT key to start listening actively.

THE STOP LIST FEATURE

This feature allows you to stop a program at any time. To use this feature, you must first set up a stop list. This is done by pressing the STOP key when the program is running. The stop list will then be created. When you press the STOP key again, the program will stop. This feature is useful when you want to stop a program that is running in the background.

4. Press the NEXT key to start listening actively.

5. Press the NEXT key to start listening actively.

APPENDICES

APPENDIX A: SUMMARY OF AUTOSTART ROM BASIC FEATURES

Command	Effect
Insert System Master and turn on power	Boots DOS and loads the BASIC not in ROM into Language Card RAM
Press RESET key	Stops and restarts, keeping program
Press ESC key	Enters edit mode
Press space bar	Leaves edit mode
Press I, J, K, or M	In edit mode, moves cursor, remaining in edit mode
Press A, B, C, or D	In edit mode, moves cursor one space, leaving edit mode
Press left-arrow key	Deletes last character typed
Press right-arrow key	Retypes character the cursor is over
Press ESC, then E	Clears line from cursor on
Press ESC, then F	Clears screen from cursor on
Press ESC, then @	Clears entire screen
Type CTRL-S	Stops listing or screen display at next RETURN
Type CTRL-S	Resumes listing

SUBROUTINES DELETED

APPENDIX B: AUTOSTART ROM TECHNICAL INFORMATION

This appendix describes the differences between the Autostart ROM and the "old" Monitor ROM.

SUBROUTINES ADDED TO THE AUTOSTART ROM

NAME: APPLE II (\$FB60) CALL -1184

PURPOSE: This routine clears the screen and POKEs the string 'APPLE][' into line 1 of the text buffer, so that the screen will display 'APPLE][' on RESET.

ENTRY: Before CALLing this routine, make sure the scrolling window parameters have been set for maximum screen limits, by using the TEXT command or

```
10 POKE 32, 0
20 POKE 33, 40
30 POKE 34, 0
40 POKE 35, 24
```

EXIT: A=\$C1; Y=0; X=entry

STATUS: N=0, Z=1, C=X, V=X

NAME: SETPWRC (\$FB6F) CALL -1169

PURPOSE: This routine lets the user make the Apple go into monitor mode when RESET is pressed. It sets location \$3F4 to the exclusive OR of the contents of location \$3F3 with \$A5.

ENTRY: No conditions; location \$3F3 (POKE 1011) should be set with the high-order address of the RESET vector for this call to be meaningful.

EXIT: A=EOR#\$A5, \$3F3; Y=entry; X=entry

STATUS: N=X, Z=X, C=entry, V=entry

COMMENTS: To get the Apple to go into monitor mode upon RESET, do a CALL -155. This will get you into monitor mode. After the monitor prompt (*), type

```
3F2:69 FF 5A
```

and press RETURN.

After you have done this, the Apple will go into monitor mode whenever you press RESET.

Monitor mode is described in the Apple II Reference Manual.

This feature allows you to cause the Apple to execute any machine-language subroutine when RESET is pressed, by setting the RESET vector to point to the address of that subroutine. For example, suppose we have a subroutine with a warm entry point of \$803. Let's set a break point (BRK) at that point, so that if we jump to that point the computer will stop at \$805. (The computer always stops two bytes beyond a break point.) First get into monitor mode by typing

```
CALL -151
```

and pressing RETURN .

Now, after the Prompt (*), type

```
803:0
```

and press RETURN .

Now let's set the 2-byte RESET vector to point to \$803. After the * type

```
3F2:3 8
```

and press RETURN , typing the low-order byte first, then the high-order byte.

Now call the routine that prevents the Apple from rebooting when RESET is pressed, by typing

```
FB6FG
```

and pressing RETURN .

Now, when you press RESET , the routine at \$803 will be executed, displaying

```
$805- A=03 X=38 Y=1B P=35 S=EB
```

SUBROUTINES DELETED

STEP	\$FA40-\$FA85
	\$FAA5-\$FAD6
	\$FAFD-\$FB18
MULPM, DIVPM	\$FB60-\$FBC0

SUBROUTINES CHANGED

This routine has been moved:

IRQ/BREAK was at \$FA86, is now at \$FA40

PAGE 3 MEMORY USAGE:

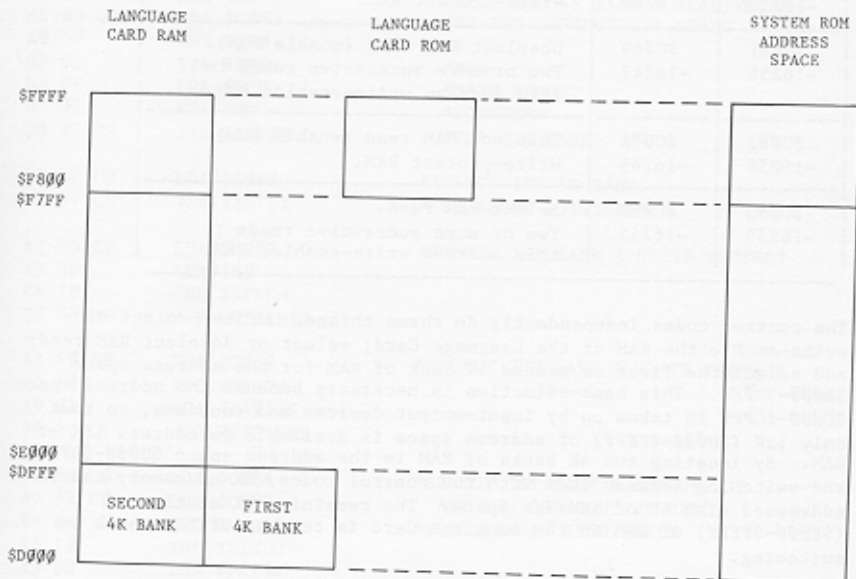
LOCATION	USAGE	DEFAULT VALUE
\$3F0, \$3F1	BREAK VECTOR	\$59, \$FA
\$3F2	RESET VECTOR LOW	\$03*
\$3F3	RESET VECTOR HIGH	\$E0*
\$3F4	POWERED-UP MASK	\$45*

LIMITATIONS OF THE AUTOSTART ROM

1. Any programs using the 16-bit integer multiply/divide routines of the old monitor will fail, as these are not in the Autostart ROM.
2. Any programs using any part of the single-step simulator code will fail for the same reason.

*For non-disk systems after power-up reset.

APPENDIX C: MEMORY MAP OF LANGUAGE CARD



APPENDIX D: LANGUAGE CARD CONTROL CODES

RAM selection for \$D000-\$DFFF		RAM/ROM selection, and RAM write-protect selection:
Second 4K Bank	First 4K Bank	
\$C080 -16256	\$C088 -16248	Select RAM read. Write-protect RAM.
\$C081 -16255	\$C089 -16247	Deselect RAM read (enable ROM). Two or more successive reads to this address write-enables RAM.
\$C082 -16254	\$C08A -16246	Deselect RAM read (enable ROM). Write-protect RAM.
\$C083 -16253	\$C08B -16245	Select RAM read. Two or more successive reads to this address write-enables RAM.

The control codes independently do three things: write-protect or write-enable the RAM of the Language Card; select or deselect RAM read; and select the first or second 4K bank of RAM for the address space \$D000-\$DFFF. This bank-selection is necessary because the address space \$C000-\$CFFF is taken up by input-output devices and routines, so that only 12K (\$D000-\$FFFF) of address space is available to address 16K of RAM. By locating two 4K banks of RAM in the address space \$D000-\$DFFF, and switching between them with the control codes, 8K of memory can be addressed with 4K of address space. The remaining 8K section (\$E000-\$FFFF) of RAM in the Language Card is reached without bank switching.

Control codes (special locations) are shown in both hexadecimal and decimal forms. The hexadecimal form (denoted by \$) is used in assembly-language programs; the decimal form, in BASIC programs. Control codes are of the form \$C08x, where x is a hexadecimal digit.

The control codes \$C084-\$C087 and \$C08C-\$C08F have the same effect as the control codes \$C080-\$C083 and \$C088-\$C08B, as bit 2 of each address is ignored.

Bit 3 in a control code selects which 4K block of RAM is located in the address space \$D000-\$DFFF. If bit 3 = 0, the first 4K bank of RAM is mapped into \$D000-\$DFFF; if bit 3 = 1, the second 4K bank of RAM is mapped into \$D000-\$DFFF.

When RAM is deselected, the ROM on the Language Card is selected for the top 2K (\$F800-\$FFFF), and ROM on the Apple printed-circuit board is selected for the address space \$D000-\$F7FF.

When RAM is deselected (and write-enabled), it may still be written but not read.

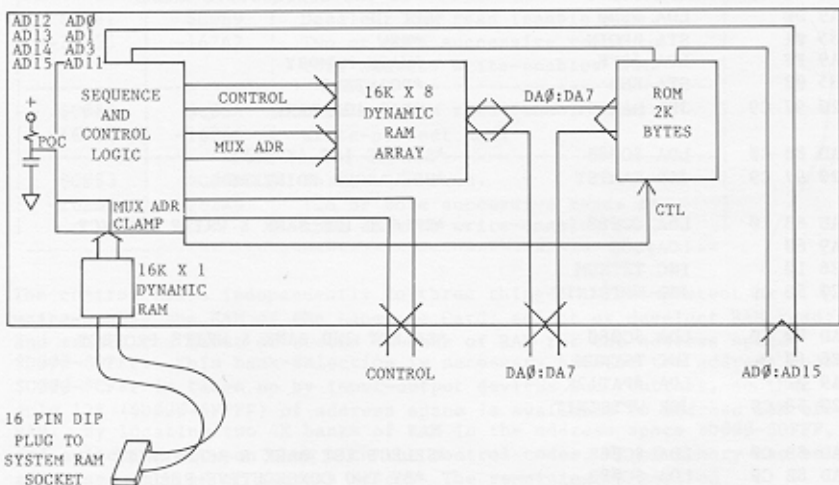
Power-on RESET initializes ROM to read mode and RAM to write mode, and selects the second 4K bank to map into \$D000-\$DFFF.

As an example of the use of these codes, let's look at the following section of an assembly-language program:

AD 83 C0	LDA \$C083	*SELECT 2ND 4K BANK & READ/WRITE
AD 83 C0	LDA \$C083	*BY TWO CONSECUTIVE READS
A9 D0	LDA #\$D0	*SET UP
85 01	STA BEGIN	*NEW
A9 FF	LDA #\$FF	*MAIN MEMORY
85 02	STA END	*POINTERS
20 97 C9	JSR RAMTST	*FOR 12K BANK
AD 8B C0	LDA \$C08B	*SELECT 1ST 4K BANK
20 97 C9	JSR RAMTST	*USE ABOVE POINTERS
AD 83 C0	LDA \$C088	*SELECT 1ST BANK & WRITE PROTECT
A9 80	LDA#\$80	
E6 10	INC TSTNUM	
20 58 C9	JSR WPTSINIT	
AD 80 C0	LDA \$C080	*SELECT 2ND BANK & WRITE PROTECT
E6 10	INC TSTNUM	
A9 01	LDA #PAT12K	
20 58 C9	JSR WPTSINIT	
AD 8B C0	LDA \$C08B	*SELECT 1ST BANK & READ/WRITE
AD 8B C0	LDA \$C08B	*BY TWO CONSECUTIVE READS
E6 0E	INC RWMODE	*FLAG RAM IN READ/WRITE
E6 10	INC TSTNUM	
A9 08	LDA #PAT4K	
20 58 C9	JSR WPTSINIT	

The LDA instruction causes a read to the specified memory location. The data in the accumulator are not used. This explains the unusual sequence of two consecutive LDA instructions.

APPENDIX E: FUNCTIONAL BLOCK DIAGRAM OF LANGUAGE CARD





10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010