Meet Your Color Computer

Send a message from Moon Base, visit two planets, and make a quiz a friend can play. Bring your Mac and your curiosity.



Tandy Color Computer 3 keyboard and computer

Brian K. Holman

Book 1 · VCC Plus 3.7.1

© Brian K. Holman 2026

Contents

Meet the machine 3
Give one instruction 7
Send a message from Moon Base 9
Pick a planet 11
Send tomorrow a message 14
Catch the computer in the act 17
Give your friend the controls 21
Your field guide 24
On the bookshelf 26
Publication and artwork notices 27

Meet the machine

Chapter 1 of 7

Picture a family in 1986 making room for a computer. Someone clears the table in front of the television. Someone else sets down a cream-colored keyboard. Where does the computer go? It's already there, inside the keyboard case.

Find the keyboard and monitor

Tandy's Color Computer 3 arrived in 1986. It was the newest member of a family of computers that owners called the **CoCo**. Switch it on and you could give it an instruction in a language called **BASIC**.



Figure 1: Illustration of the Color Computer 3. The keys are built into the case that houses the computer.

A family could use its television for the picture or give the CoCo a monitor of its own. A separate drive could sit beside the computer. Slip a floppy disk into that drive and you had a place to keep programs for another day.

The original CoCo 3 had **128 KB of RAM**. RAM is the computer's working memory. KB stands for kilobytes, a way to measure how much it can hold. Owners could expand that space to **512 KB**, four times as much. Look for that number on the machine in this book.

Remember using that floppy drive? Find its controls on the Mac in *Back at the CoCo*.

Set a CoCo on your Mac desk

Open VCC Plus on your Mac. It's an **emulator**. Your Mac does the work of running a CoCo, right down to accepting the keys you press. For these lessons, start with the **Disk BASIC** setup and its supplied **Hello World** disk in **FD0**, the first floppy drive.

On a fresh setup, go straight to the main emulator window. If **Using your CoCo** is open, read the short guide, then click outside it to get back to your computer.

Already use VCC Plus? Make a practice setup

Save any CoCo work you want to keep before loading a setup. Loading one clears the CoCo's working memory. Here's how to give your book experiments a starter disk of their own.

1. Open **Machine → Configuration Profiles...**.
2. Choose **Disk BASIC** under **System Profiles**.
3. Choose **Create Fresh Starter Media...**.
4. Name your setup **My First CoCo** and choose **Save**.
5. With your new setup selected, choose **Load...**, then **Load Profile**.
6. Return to the main window.

If the machine is paused, choose **Run**.

Choose **Display → Interactive System**. Find the Color Computer 3 nameplate and the RAM badge. The case underneath the keys houses the computer. Look above the keys for the monitor, then beside the monitor for the disk drives. Find **FD0**. If you'd like a list of the drives, open **Media → Floppy Disks...**. FD0 should show **Hello World.dsk**.

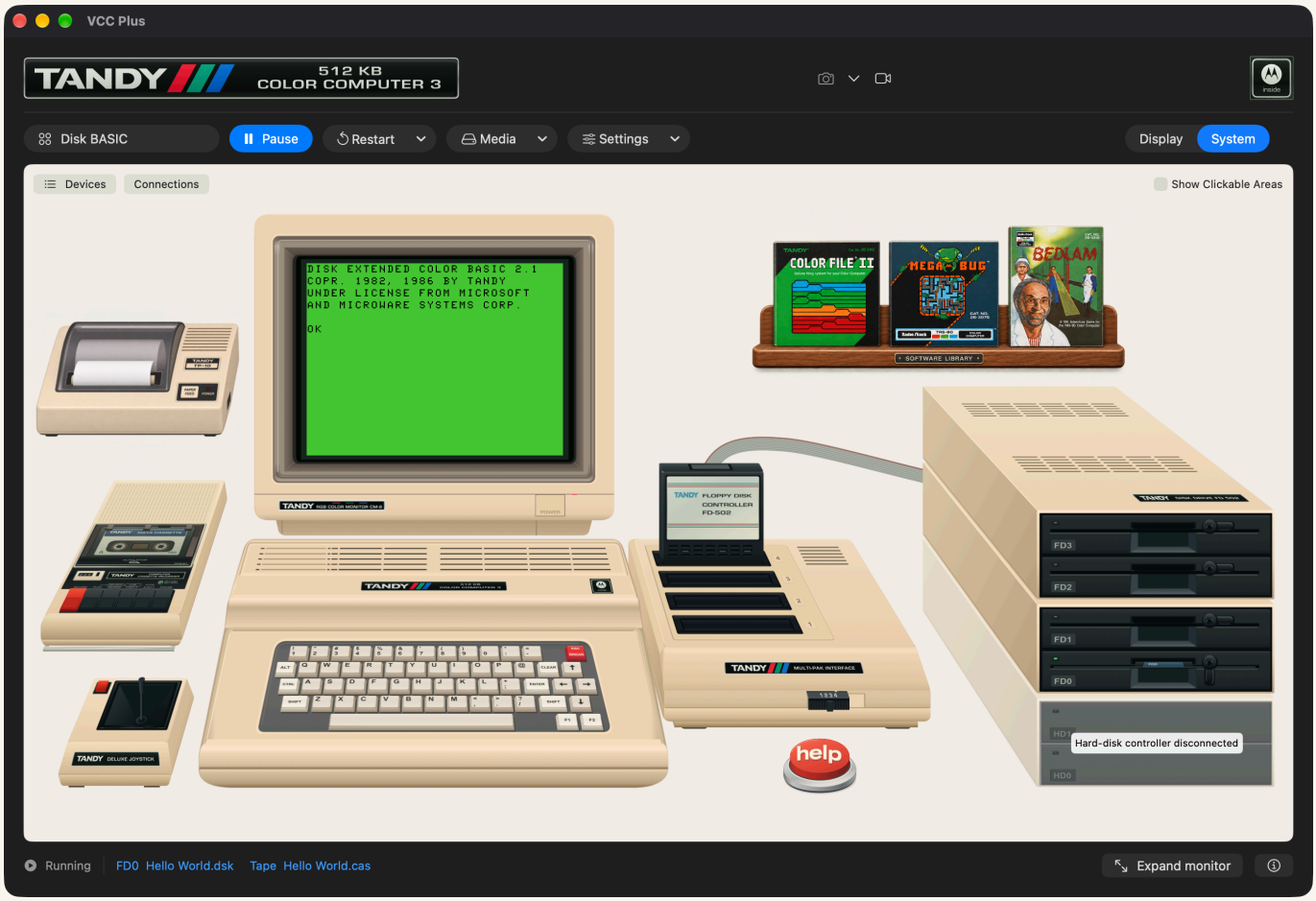


Figure 2: VCC Plus System view. Find the CoCo keyboard case below the monitor and the disk drives beside it.

Try pointing to three places in the picture. **Below the monitor**, find the keyboard where a person at a real CoCo would type. **Inside the monitor**, find the green screen where the answer appears. **To the right**, find the drive labeled FD0. A real floppy drive had a slot for a thin disk. Later you'll slip a whole disk into this one by choosing a file on your Mac.

Get a reply

Look for **OK** on the CoCo screen. That's the **prompt**. BASIC is ready for an instruction. The blinking square shows where the next character will appear.

1. Click inside the CoCo screen. From the keyboard, choose **Input → Focus CoCo Display** or press **Option-Command-K**. This directs your typing to the CoCo.
2. Type the line below, including both double quotation marks. Pause with your finger over Return.

```
PRINT "HELLO, COCO!"
```

So far, every letter on that line came from your fingers. Press **Return**. Look just below the line you typed.

```
HELLO, COCO!
OK
```

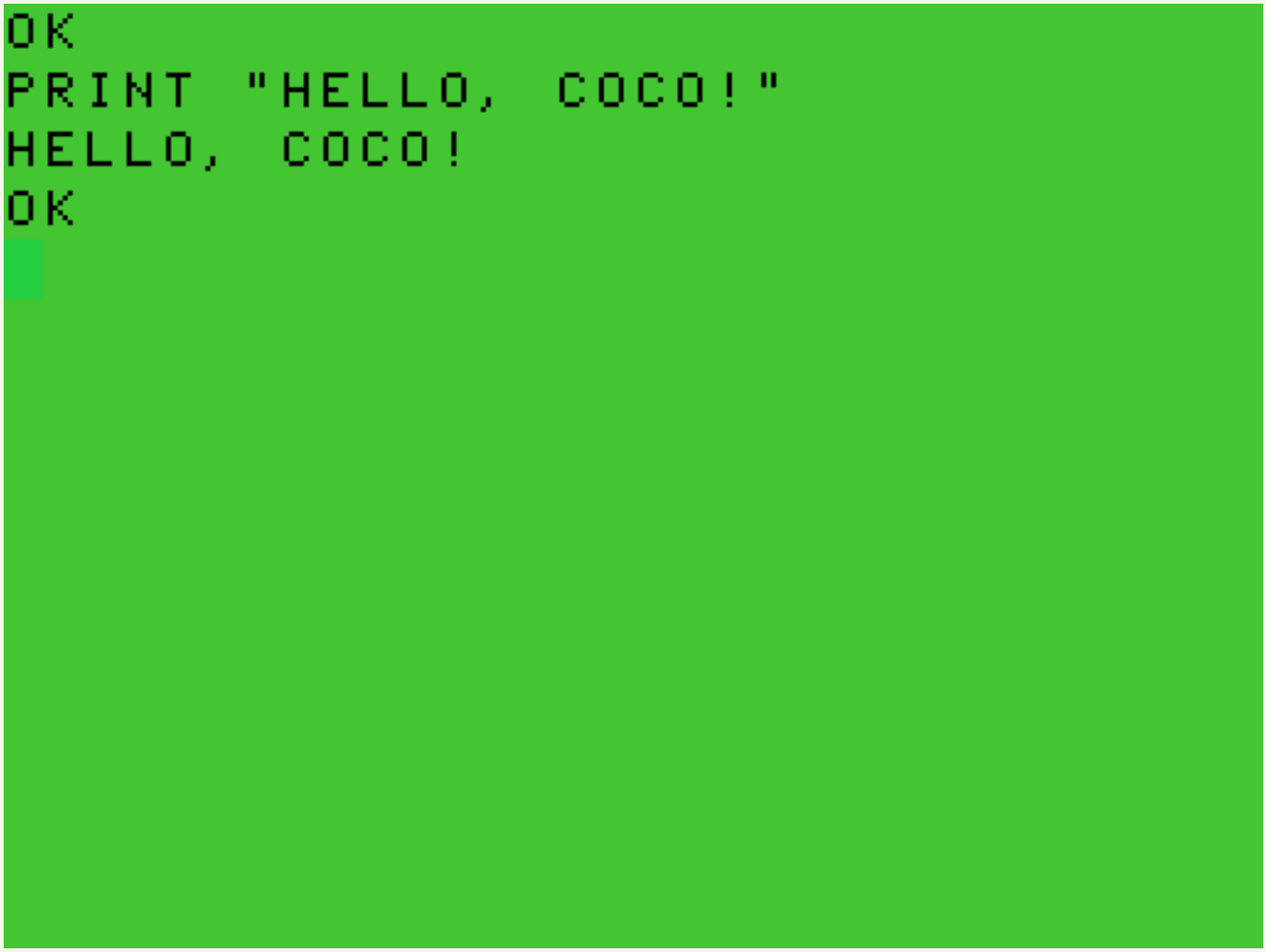


Figure 3: VCC Plus showing the typed PRINT instruction, the greeting on its own line, and the next OK prompt.

Your keystrokes were **input**. The CoCo followed your instruction and sent the greeting to the monitor. That greeting was **output**. Find the exclamation mark twice on your screen. Can you tell which one belongs to the instruction and which one belongs to the reply?

If typing goes to a Mac control, press **Option-Command-K** and try the line again. To go the other way, choose **Input → Focus Mac Controls** or press **Option-Command-U**. **Tab** moves between Mac controls and **Space**

activates a focused button. If Tab skips buttons, turn on **Keyboard navigation** in **System Settings → Keyboard** on your Mac.

Try leaving an answer half typed, then bringing the same window back in the handbook's flight-number exercise.

Where to go next

You supplied every letter of the greeting. Can you get the CoCo to print an answer you never typed? Leave it at the prompt for chapter 2.

Tandy Corporation. *Introducing Your Color Computer 3*. 1986. "To Our Customers" and the opening hardware tour.

CoCopedia contributors. "Color Computer 3." *CoCopedia*, revision 11134.

Give one instruction

Chapter 2 of 7

You've asked BASIC to show words you supplied. Now give it a plus sign. Can you make the sign disappear in one answer and show up in the next?

Try it on your CoCo

At the **OK** prompt, click the CoCo screen or press **Option-Command-K**. Type this line. Guess what will appear beneath it, then press **Return**.

```
PRINT 2+3
```

Check your prediction

The answer is **5**. Look closely and you'll see a space before the digit. BASIC leaves room for the sign when it prints a positive number.

Here are the same numbers and the same plus sign. This time, type a double quotation mark on each side of them. Make your prediction, then press **Return**.

```
PRINT "2+3"
```

You should see **2+3**. Inside the quotes, each character is a piece of text to show. The plus sign gets a spot on the screen along with the two digits. The quote marks themselves have done their job by telling BASIC where the text begins and ends.

Put the question and answer on one line

There are **two** plus signs in this next instruction. Point to the one you think will reach the screen. Type the line at the prompt and press **Return**.

```
PRINT "2+3=";2+3
```

```
2+3= 5
```

Read it from left to right. First comes the text between the quotes, right through the equals sign. The semicolon tells PRINT to carry straight on with the next item. Outside the quotes, the second plus sign tells BASIC to add. The result is a number, so it gets that space in front of it.

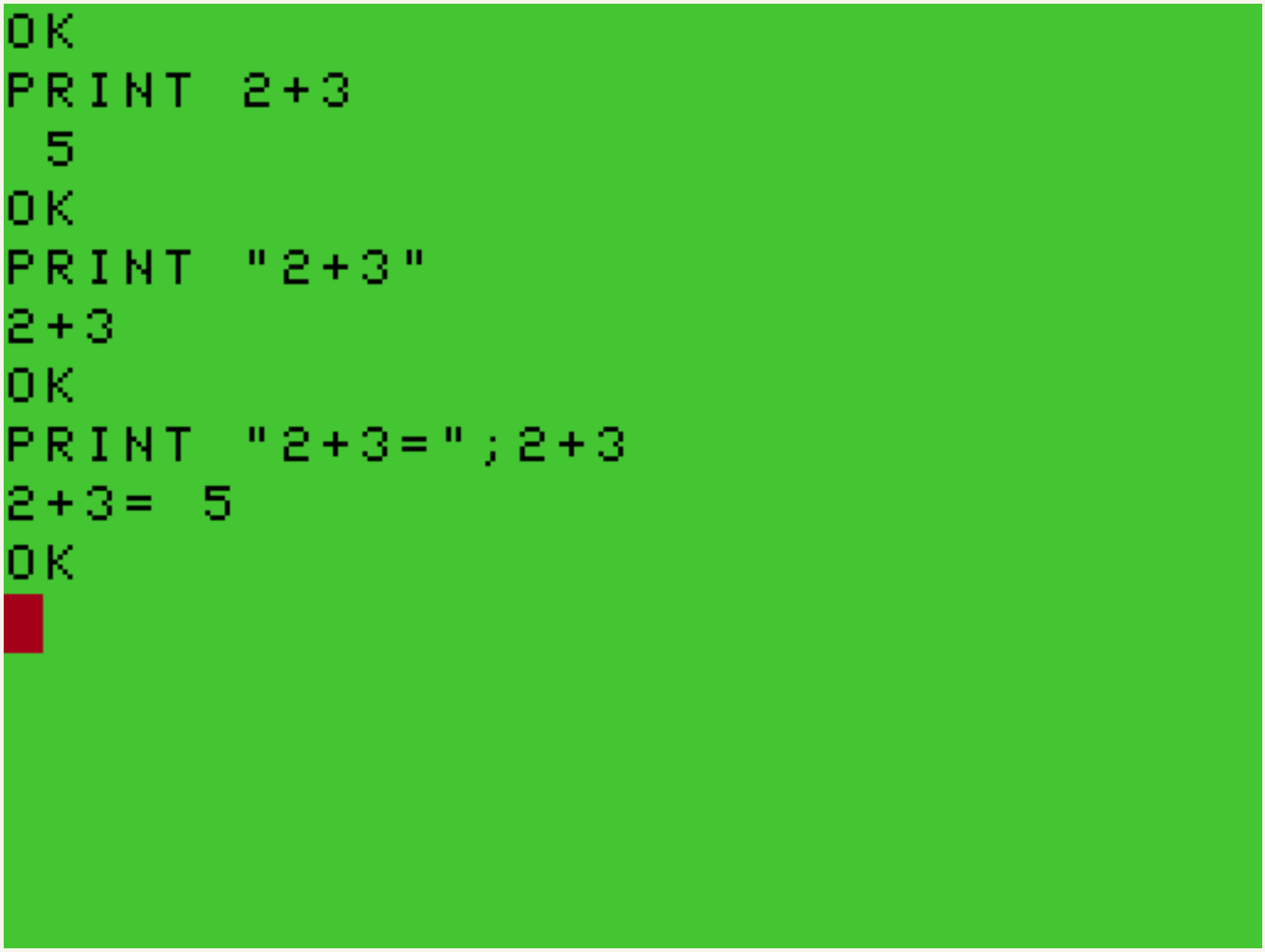


Figure 4: The CoCo shows three answers in order. First comes the number 5, then the text 2+3, then 2+3= 5 together on one line.

How did the CoCo know the word PRINT? Its **ROM**, short for *read-only memory*, holds a program called the **BASIC interpreter**. The interpreter recognizes BASIC words and symbols. A chip called the **CPU**, or *central processing unit*, carries out the interpreter's instructions. Your two plus signs gave it two different jobs on the very same line.

Try expressions at the BASIC prompt.

Where to go next

So far, pressing Return has meant "do this now." In chapter 3 you'll give the CoCo a message it can repeat on demand.

Curious about fitting more letters across the screen? Pick a planet uses a 40-column CoCo 3 display. You can come back to the Moon Base story whenever you like.

Tandy Corporation. *Tandy Color Computer 3 BASIC Quick Reference Guide*. 1986. Commands, pp. 2–17, particularly PRINT.

Send a message from Moon Base

Chapter 3 of 7
Earth has come into range. Moon Base has a two-line message ready. Can you get the CoCo to hold both lines and send them to the screen on your signal?

Give each line a number

Continue at the CoCo prompt from chapter 2. The calculations you tried there have left you with an empty program. Click the screen or press **Option-Command-K**. Type these two lines, pressing **Return** at the end of each one.

```
10 PRINT "THIS IS MOON BASE."
20 PRINT "CAN YOU HEAR ME?"
```

Where's the reply? Look at the first thing you typed on each line. A **line number** tells BASIC to keep that instruction in RAM. Together the stored instructions make a **program**. Your words are ready in memory, waiting for the signal to start.

Send the signal

Type this command on a fresh line and press **Return**.

```
RUN
```

```
THIS IS MOON BASE.
CAN YOU HEAR ME?
OK
```

BASIC starts with the lowest line number. It carries out line 10, moves on to line 20, then hands the keyboard back to you at OK. The prompt is how you know this transmission is over.

Change the request

The crew has a more urgent question. Type a fresh line 20 exactly as shown and press **Return**.

```
20 PRINT "SEND SNACKS!"
```

How many instructions does the CoCo have now? Type this and press **Return** to see the stored program.

```
LIST
```

```
10 PRINT "THIS IS MOON BASE."
20 PRINT "SEND SNACKS!"
```

Count the line numbers. There are still two. BASIC used the number 20 to find the instruction you meant to replace. Line 10 was already in RAM, ready to use again.
Type **RUN** and press **Return**. Read the new transmission on your screen.

```
THIS IS MOON BASE.
SEND SNACKS!
OK
```

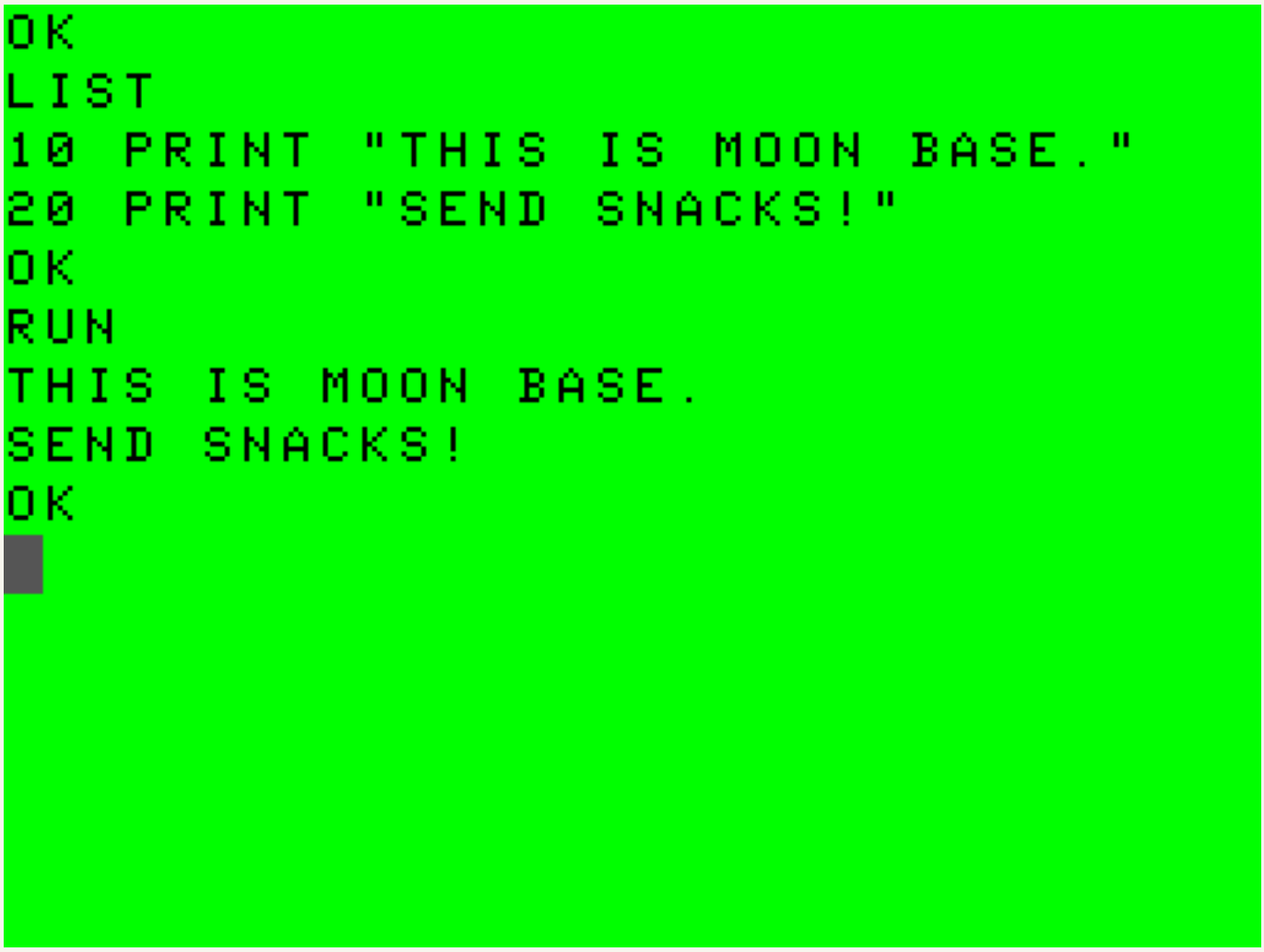


Figure 5: LIST shows the instructions with their line numbers. RUN sends the two messages, including the crew's new request for snacks.

If the old question still appears, use **LIST** again. Check the number and the words on the second line. You can retype that whole numbered line, press **Return**, and try **RUN** again.

Renumber a program and follow its branches.

Keep the beacon handy

At the moment your beacon lives in the CoCo's RAM. Chapter 4 will use that space for a different program. You can rebuild the beacon from the two lines shown under LIST, including the crew's snack order. In chapter 5 you'll give it a home on disk.

Tandy Corporation. *Tandy Color Computer 3 BASIC Quick Reference Guide*. 1986. Commands, pp. 2–17, particularly LIST, PRINT, and RUN.

Pick a planet

Chapter 4 of 7

Two travelers run the same program. One gets a greeting from Earth and the other gets a greeting from Mars. Can you find the moment their trips take different paths?

Make two trips

Use the Disk BASIC session from chapter 3, with the starter disk in FD0, the first floppy drive. Loading the next program uses the RAM that held your beacon. Its two lines are in chapter 3 for when we need them again.

1. Click the CoCo display, or choose **Input → Focus CoCo Display** (Option-Command-K). At `OK`, type `LOAD"HELLO"` and press **Return**. Wait for `OK`.
2. Type `RUN` and press **Return**. The program asks `What planet do you live on?` and offers two choices.
3. Tap **3** by itself. The choices stay on screen. Now tap **1**. Read the greeting on the blue strip. What will change if you take the second trip?
4. At `OK`, type `RUN` and press **Return** again. This time tap **2**. Read the greeting on the red strip.

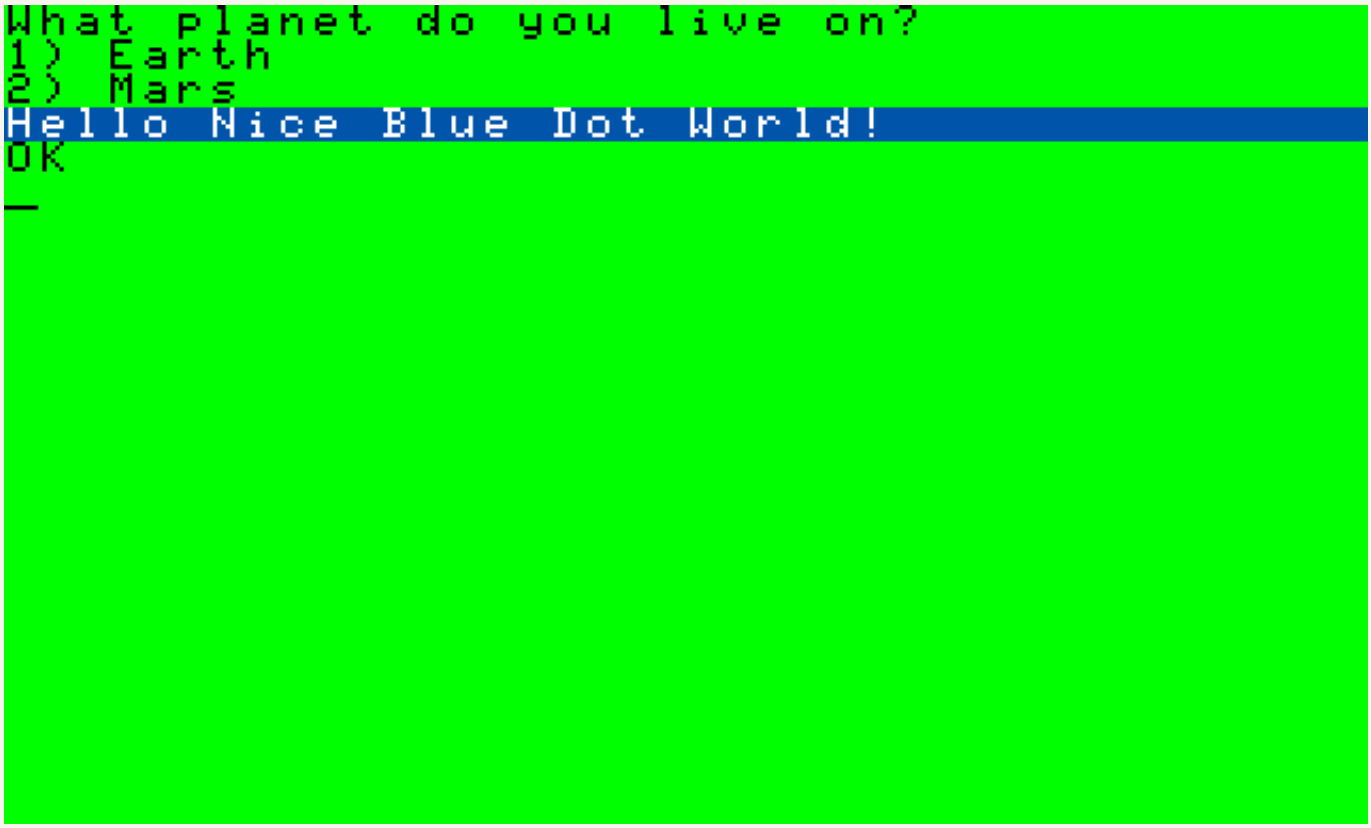


Figure 6: Choosing 1 for Earth prints “Hello Nice Blue Dot World!”



Figure 7: Choosing 2 for Mars prints “Hello Cold Red Dot World!”

At the prompt, **Return** tells BASIC that your command is complete. During the planet question, the running program reads one key at a time. It can act on **1** or **2** the instant you press it.

Follow the choice in the listing

A **listing** is the set of instructions you can read and edit. In VCC Plus choose **Develop** → **BASIC Development**. The CoCo is at `OK` with the planet program still in RAM, so choose **Read from CoCo**. If you opened the lesson separately, click the planet listing in the Manual Library to work on an editable copy. In a web browser, save the file and use **Open…** in BASIC Development. In the editor, choose **More file actions** (the three dots) → **Save As…** and give your working copy a fresh name such as `My Planet.bas`.

Find line 60. `INKEY$` checks the keyboard and `K$` holds the character it found. The dollar sign marks a name that holds text. Line 70 asks whether that text is `"1"`. If it is, BASIC jumps straight to line 110, the Earth greeting. If you pressed **3**, neither planet matches and line 80 sends BASIC back to line 60 to check again. That's what it was doing while the screen seemed still.

Look up at line 20. `WIDTH 40` gives the CoCo 3 room for 40 characters on a row, eight more than the 32 across the first screen you used. Each route chooses a color with `PALETTE` and puts a greeting in `M$`. Both routes meet at line 120. `ATTR 3,6` selects the colors for the letters and their background, and `PRINT M$` displays the chosen greeting.

Try 32, 40 and 80 columns.

Send a greeting of your own

1. Find line 110 in your Mac copy. Select only the words between its quotation marks and replace them with `Greetings from Earth!`. Your whole line should now read like this.

```
110 PALETTE 6,10:M$="Greetings from Earth!"
```

2. Choose the editor's **Save**, then its **Run**. If it asks about replacing the BASIC program in CoCo memory, check that you're working in the practice session and confirm.
3. Press **Option-Command-K** to return to the display. Tap **1** at the planet question. Look for `Greetings from Earth!` on the blue strip.

You changed the words while keeping the color choice. Where would you look for the words a visitor to Mars sees?

Find the Mars greeting

Line 90 pairs the Mars color with the words `Hello Cold Red Dot World!`. Both planets use the same `PRINT` on line 130.

For more on the display commands, see Tandy Corporation, *Tandy Color Computer 3 BASIC Quick Reference Guide* (1986), commands, pp. 2–17.

Send tomorrow a message

Chapter 5 of 7

Suppose someone comes to Moon Base tomorrow and wants to hear your beacon. Can you bring the message back after you empty BASIC's working memory?

Make a disk for Moon Base

A **disk image** is a Mac file that holds the contents of a whole CoCo disk. We'll copy the starter disk. It already has space prepared for BASIC files and a program called HELLO. Choose a fresh filename so you can recognize your disk tomorrow.

1. In your Disk BASIC practice session, choose **Media → Floppy Disks…**. Find **FD0** and check that it holds the starter disk from chapter 4. If you see a personal disk there, return to your practice setup before continuing.
2. In FD0 choose **Actions → Save As…**. Choose an easy-to-find folder and enter a fresh Mac filename such as `Moon Base.dsk`. If that name already exists, choose an unused one. Keep a note of the folder and name.
3. Read FD0 again. Look for **your new filename** and **Saves automatically**. If the new copy says **Temporary changes** or **Write protected**, open **Disk options** in its row and turn off that setting on your copy. Wait for any save to finish. Resolve any save error before continuing.
4. Return to the main window. If the CoCo is paused, choose its **Run**. Click the display or press **Option-Command-K** to type at the CoCo.

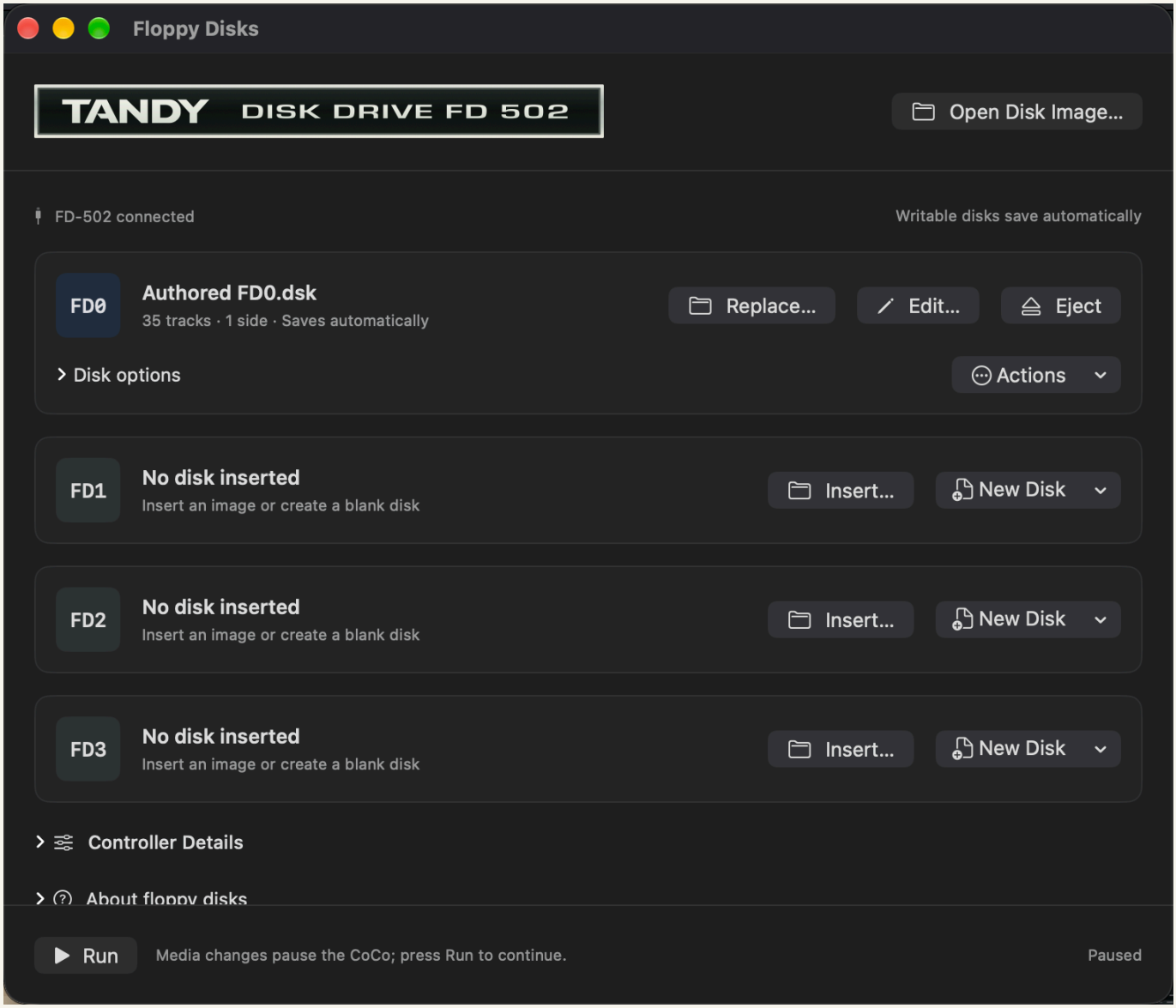


Figure 8: The current Floppy Disks window shows the filename and save mode in FD0. This example uses Authored FD0.dsk; your practice copy should show Moon Base.dsk (or the fresh name you chose) and Saves automatically.

To use these controls from the keyboard, **Input → Focus Mac Controls** (Option-Command-U) takes you out of the display. **Tab** and **Shift-Tab** move between Mac controls, and **Space** operates a focused button. If Tab skips buttons, turn on **Keyboard navigation** in **System Settings → Keyboard**.

Rebuild the beacon and save it

1. At `OK`, type `NEW` and press **Return**. This clears the planet program from BASIC's working memory.
2. Enter both numbered lines below, pressing **Return** after each one. Include the new snack request from chapter 3.

```
10 PRINT "THIS IS MOON BASE."
20 PRINT "SEND SNACKS!"
```

3. Type `RUN` and press **Return**. Check both lines of the message. If one needs a fix, retype its whole numbered line and run it again.

4. At `OK`, type `SAVE"BEACON"` and press **Return**. Wait for `OK`. This tells Disk BASIC to store the program on the disk in FD0.
5. Type `DIR` and press **Return**. This asks the disk for its directory, the list of files it holds. Find `BEACON` with `BAS` beside it. You should also see `HELLO`. Check the FD0 row for a completed save before clearing RAM.

If BASIC says `?WP ERROR`, the disk is write protected. Your two lines are still in RAM. Check the filename on FD0, open **Disk options** on your practice copy and turn off **Write protected**. If the switch is unavailable, use **Actions → Save As…** to make a fresh copy in your own folder. When the save finishes, use FD0's **Replace…** to open that new file. Turn off **Write protected** on the new copy if needed. When FD0 says **Saves automatically**, choose **Run** if paused and press **Option-Command-K**. Repeat the save and directory steps. For a Mac save error, resolve it using **Retry Save** or **Save As…** before going on.

Empty RAM and bring the message back

1. At `OK`, type `NEW` and press **Return**. Type `LIST` and press **Return**. The next `OK` appears with no numbered lines from that listing. Where could the two lines still be?
2. Type `LOAD"BEACON"` and press **Return**. Wait for `OK`. BASIC has copied the program from disk into RAM.
3. Type `RUN` and press **Return**. Check the reply.

```
THIS IS MOON BASE.
SEND SNACKS!
OK
```

Point to the three places involved. **RAM** holds the lines BASIC can run right now. **BEACON** is the name you look up with `DIR` on the CoCo disk. **Moon Base.dsk** is the file you'd look for in your Mac folder to find the whole disk again. Keep that disk in FD0 for your quiz. To see the same distinction with another program, find CARGO inside its disk image.

The editor's **Save** in chapter 4 wrote readable BASIC text to a Mac document. Here you typed `SAVE"BEACON"` at the CoCo prompt, so BEACON appears *inside* the disk image alongside HELLO.

For the disk commands and write-protection error, see *Color Computer Disk BASIC Quick Reference Guide* (InterTAN Canada Ltd., 1987), pp. 3–9.

Catch the computer in the act

Chapter 6 of 7

The beacon keeper wants three blinks. If the CoCo prints all three before you can look, how can you find out when it changed from blink one to blink two?

Give the signal a counter

This program prints a numbered message for each blink. Read from line 20 to line 40 with your finger. How many times do you think your finger will pass line 30?

```
10 CLS
20 FOR I=1 TO 3
30 PRINT "BLINK";I
40 NEXT I
50 PRINT "SIGNAL SENT!"
```

`I` is a **variable**, a name BASIC uses to keep a value in RAM. Here its job is to count. `FOR I=1 TO 3` starts the count at 1. `PRINT "BLINK";I` puts a word and the current number on the screen. `NEXT I` adds one to the counter and sends BASIC around again while there is another turn left. A group of instructions that repeats this way is a **loop**.

In **Help → Manual Library...**, click SIGNAL.bas. An editable copy opens in a **BASIC Example** window. Keep the book beside it so you can read and step through the program together. The CoCo keeps doing what it was doing until you choose **Run**.

Reading in a web browser? Save a copy of the linked file on your Mac, then choose **Develop → BASIC Development → Open...** in VCC Plus to open it. Save any previous work if the editor asks.

Count by twos or count backward.

Stop before the first blink

1. In the editor, click the circle in the left margin beside line **30**. You've set a **breakpoint**, a request for the CoCo to pause when it reaches that line. Using the keyboard, move the insertion point to line 30 and activate the toolbar button **Toggle BASIC breakpoint at insertion point**. Use either method once.
2. Expand **Variables and Watches** below the listing if it is closed. Choose **Watches**, type the capital letter `I` in its field and choose **Add** or press **Return**. A **watch** lets you read a value at each stop.
3. Choose the editor's **Run**. Choose **Replace and Run** if asked about replacing your previous practice program in RAM. Look for the highlight at line 30 and the value **1** beside `I`. Line 10 has cleared the CoCo screen. The first BLINK is still to come.

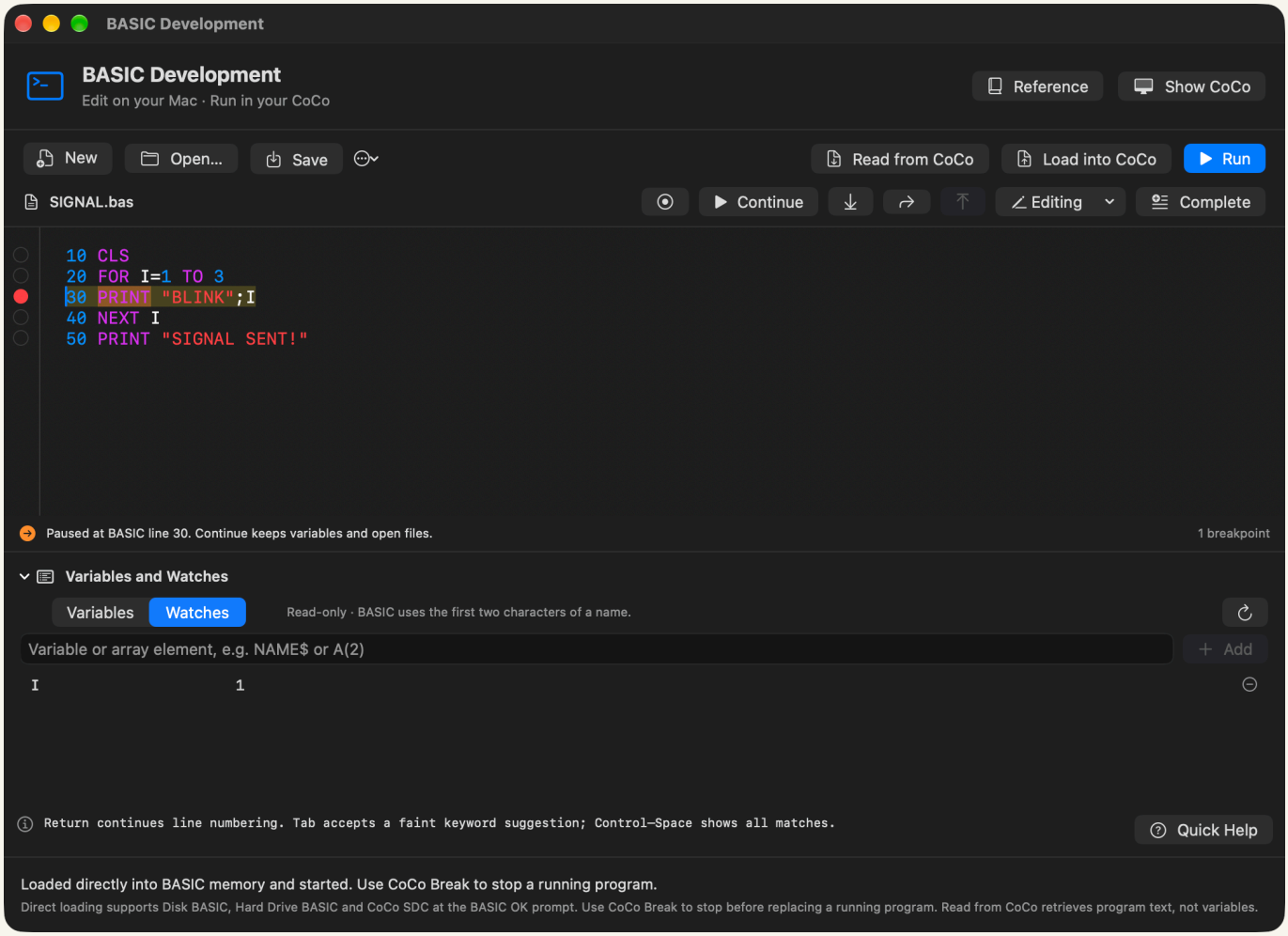


Figure 9: In BASIC Development the line highlight shows where execution paused. The watch gives the value of I at that stop.

Choose **Step Into BASIC statement**, the down-arrow button beside **Continue**, once. The highlight moves to line 40. On the CoCo display, `BLINK 1` has appeared. Read the watch again. It still says **1**. Which instruction will change it?

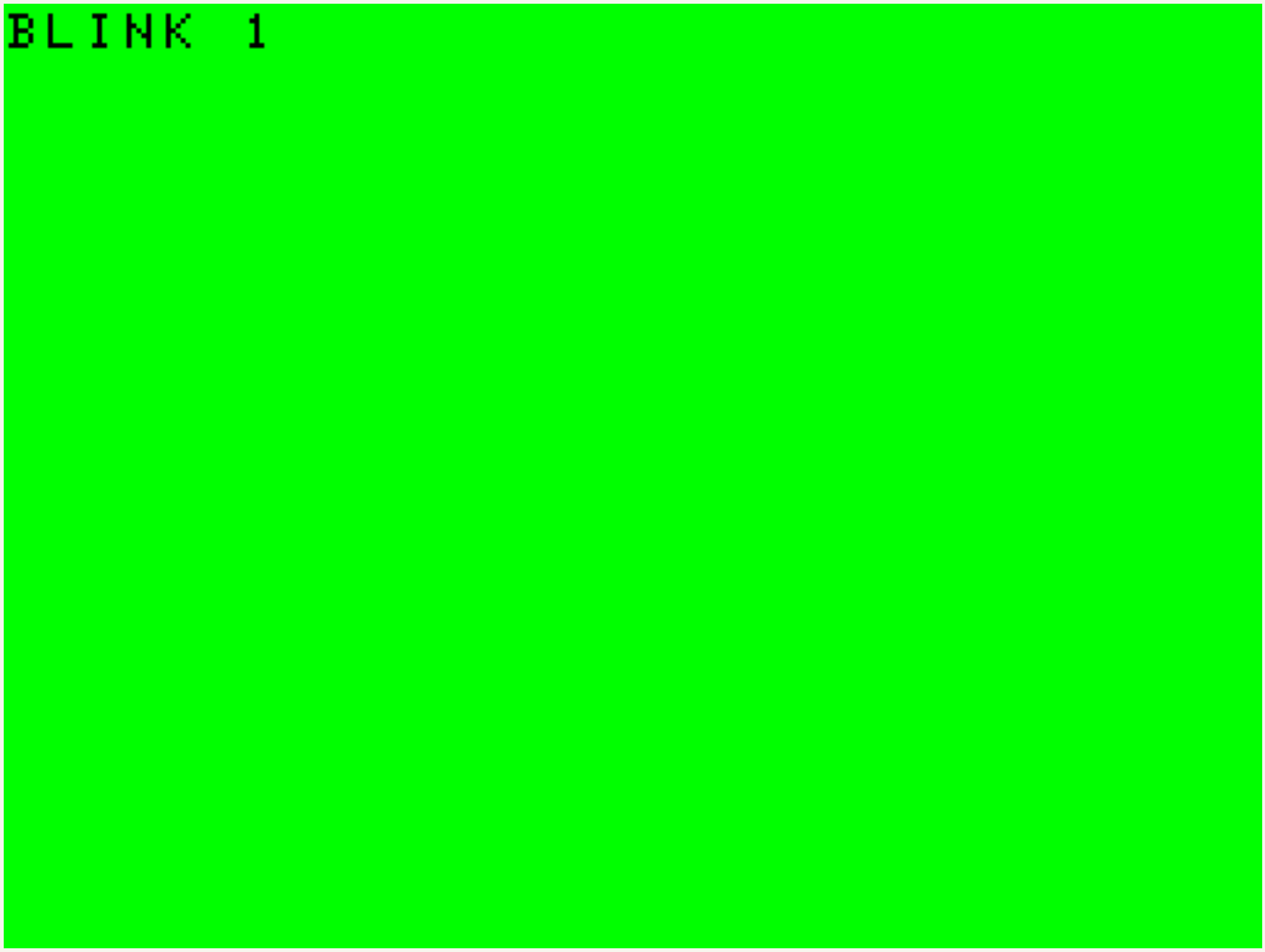


Figure 10: After one step the CoCo shows BLINK 1. BASIC has printed the first message and is waiting before NEXT I.

If all three blinks print in one burst, your program has finished. First check that the marker is beside line 30 and you used **Run** in BASIC Development. If it happens again, save any work you want to keep. In your practice session, choose **Cold Restart – Clear RAM…**, confirm, and wait for BASIC's **OK**. Open SIGNAL in the editor, check the line 30 breakpoint and run it from there.

To see the guest screen if the editor covers it, choose **Show CoCo**. Use the **Window** menu to bring the editor with SIGNAL back to the front. Choose **Help → Manual Library…** to bring this lesson back. The Mac controls work while the emulated CoCo waits.

Predict the next two stops

Choose **Continue** once. This resumes your paused program. Before looking at the screen, predict what has changed. Line 30 is highlighted again. The watch says **2**, but only `BLINK 1` has printed. The highlighted line is the instruction BASIC is *about to* carry out.

Choose **Continue** once more. Stop and count the messages before you peek at the watch. Now you should have two messages and a counter of **3**. What will appear if you continue one last time? Make your prediction, then choose **Continue** a third time.

Check the finished signal

The CoCo prints the last blink, finishes the loop and carries on to line 50. Here is the complete screen text.

```
BLINK 1
BLINK 2
BLINK 3
SIGNAL SENT!
OK
```

When you see `SIGNAL SENT!` followed by `OK`, the CoCo is ready for the quiz.
Watch a subtraction one statement at a time.

How a number gets onto the screen

At the second stop on line 30, RAM held the number 2 under the name `I`. The screen still showed the earlier number 1. When BASIC carried out `PRINT`, it turned the current number into a character you could read. You saw those two jobs happen at different times.

Memory stores patterns of **bits**. A bit can be 0 or 1, and eight bits make one **byte**. The CoCo uses patterns in memory to represent the characters on its screen. BASIC also uses space in RAM to keep a variable's name, the kind of value it holds, and the value itself. A one-letter name on your watch is enough to look up all that information.

To watch a number change one instruction at a time, follow a byte through register A. To see how RAM becomes a picture, look for a dot on a hidden page.

For the loop commands, see Tandy Corporation, *Tandy Color Computer 3 BASIC Quick Reference Guide* (1986), commands, pp. 2–17.

Give your friend the controls

Chapter 7 of 7

You get to make the entrance quiz for the Moon Base crew. A friend gets the keyboard. Can the CoCo ask the questions, keep the score, and play your game again after a restart?

Make the first point count

Start in your Disk BASIC practice session with the Moon Base disk from chapter 5 in FD0. Finish the signal program so you can see `OK`. Click the CoCo display or press **Option-Command-K**. This time you'll build the game at the CoCo keyboard.

1. Type `NEW` and press **Return** to make room in RAM for the quiz.
2. Type these seven lines, pressing **Return** after each one. Use the digit zero in `S=0`. Use straight double quotation marks around each piece of text.

```
10 CLS
20 S=0
30 PRINT "MOON BASE CREW QUIZ"
40 PRINT "WHICH PART SHOWS A MESSAGE?"
50 PRINT "1 SCREEN   2 KEYBOARD"
60 INPUT A
70 IF A=1 THEN S=S+1
```

3. Type `RUN` and press **Return**. At the question mark, type **1** for SCREEN and press **Return**. This short version ends here and gives you `OK`.
4. At `OK`, type `PRINT S` and press **Return**. Look for **1**. The program kept a point even though you hadn't yet written the line to display the score.

Line 60, `INPUT A`, waits for a number and **Return**, then stores the number under the name `A`. Line 70, `IF A=1 THEN S=S+1`, makes a decision. Read its first part as a question. “Is the answer in A equal to 1?” If it is, BASIC does the part after `THEN`. It takes the old score in `S`, adds one and stores the result back in `S`. A different answer skips that addition. The first equals sign **compares**. The second **stores**.

Follow an answer through a two-planet menu.

Add two more questions

Keep the first seven lines in RAM. At `OK`, enter the remaining lines below, one at a time. The higher line numbers put them after line 70. Line 110 awards a point for answer **2** because DISK is the second choice on that question.

```
80 PRINT "WHICH KEEPS A SAVED PROGRAM?"
90 PRINT "1 RAM   2 DISK"
100 INPUT A
110 IF A=2 THEN S=S+1
120 PRINT "WHAT DOES PRINT 2+3 SHOW?"
130 PRINT "1 FIVE   2 THE TEXT 2+3"
140 INPUT A
150 IF A=1 THEN S=S+1
160 PRINT "YOUR SCORE";S;"OUT OF 3"
170 END
```

Run the game twice. Each time, type `RUN` at `OK` and press **Return**. Give *one number and Return per question*. Try **1, 2, 1** on the first run. Try **2, 1, 2** on the second. Predict each final score.

Check the two scores

On the first run, look for `YOUR SCORE 3 OUT OF 3`. On the second, look for `YOUR SCORE 0 OUT OF 3`. Line 20 sets the score to zero at the start of every game. Each `INPUT A` replaces the previous answer, while `S` keeps the points earned so far. Line 160 prints the score.

If a score surprises you, type `LIST` at `OK`. Compare lines 70, 110 and 150 with the listing. Retype a whole numbered line to correct it, then run both tests again.

Open the complete runnable quiz

QUIZ.bas contains the entire original program. Click its name in the Manual Library to read the complete listing in its own editable window. In a web browser, save a copy and use **Open...** in BASIC Development.

```
10 CLS
20 S=0
30 PRINT "MOON BASE CREW QUIZ"
40 PRINT "WHICH PART SHOWS A MESSAGE?"
50 PRINT "1 SCREEN   2 KEYBOARD"
60 INPUT A
70 IF A=1 THEN S=S+1
80 PRINT "WHICH KEEPS A SAVED PROGRAM?"
90 PRINT "1 RAM   2 DISK"
100 INPUT A
110 IF A=2 THEN S=S+1
120 PRINT "WHAT DOES PRINT 2+3 SHOW?"
130 PRINT "1 FIVE   2 THE TEXT 2+3"
140 INPUT A
150 IF A=1 THEN S=S+1
160 PRINT "YOUR SCORE";S;"OUT OF 3"
170 END
```

Give the game a title and a home

1. At `OK`, type this replacement line and press **Return**.

   ```
   30 PRINT "SPACE CADET QUIZ"
   ```

2. Type `RUN` and press **Return**. Look for `SPACE CADET QUIZ`, then answer all three questions to return to `OK`.
3. In **Media → Floppy Disks...**, check the filename in FD0 against the practice disk you made. Check for **Saves automatically**. Press **Option-Command-K** to focus the CoCo, then type `DIR` and press **Return**. On your chapter 5 disk you'll see HELLO and BEACON. If there's already a QUIZ you want to keep, make a fresh disk copy with FD0's **Actions → Save As...** before saving this version. If you made a copy, choose **Run** if paused, then press **Option-Command-K**.
4. At `OK`, type `SAVE"QUIZ"` and press **Return**. Wait for `OK`. Type `DIR`, press **Return**, and find the entry with `QUIZ` and `BAS`.
5. Check FD0 once more and wait for the disk save to finish. For a write-protection or save error, keep your game in RAM and follow the disk recovery steps before you restart.

Let someone play after a restart

1. In the main VCC Plus window open **Restart → Cold Restart – Clear RAM...**. Now that your game is saved, confirm **Cold Restart**. This restarts the CoCo and clears its working memory.
2. Check that your practice disk is still in FD0. If the machine is paused, choose the main window's **Run**. Return to the CoCo display and wait for `OK`.

3. Type `LOAD"QUIZ"` and press **Return**. Wait for `OK`, then type `RUN` and press **Return**. Your new title should appear.
4. Move the book and listing out of sight. Give your friend the keyboard. Tell them, “Choose 1 or 2 and press Return for each answer.” Let the CoCo tell them their score.

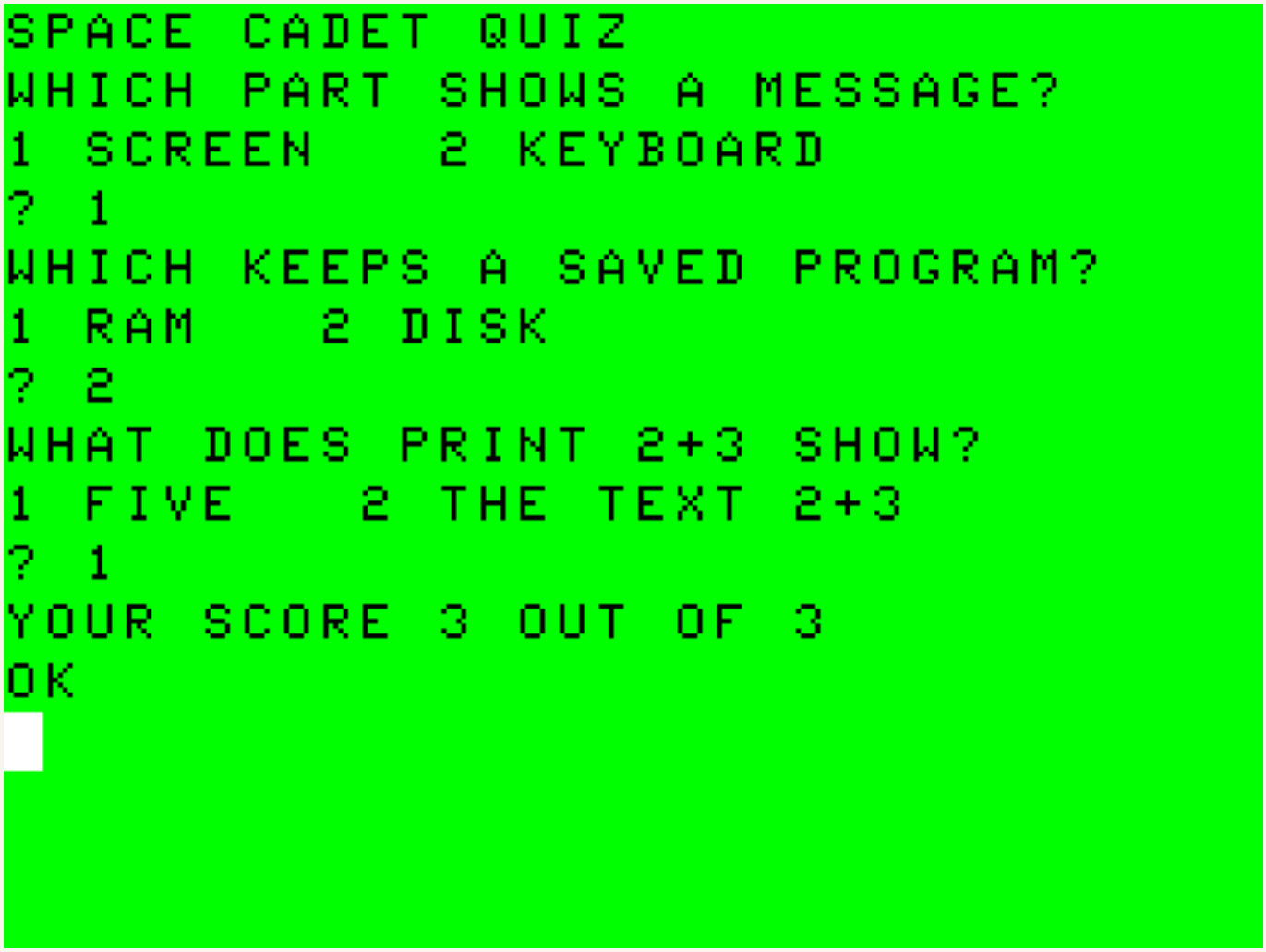


Figure 11: The customized game runs from disk with the title SPACE CADET QUIZ and prints a final score.

Ask your friend to pick one answer they gave. Can you tell its story? The keyboard supplied a number, `INPUT` put it in RAM, and the CPU followed your `IF` instruction to decide whether to add a point. The screen showed the total. And when the restart cleared RAM, your disk held the program, including the title you wrote.

If you want to write a question about your friend, keep this working game on the Moon Base disk. Make another practice copy, change one question and its choices, and decide which answer earns the point. Play it once each way to see whether your rule works.

For the language commands, see Tandy Corporation, *Tandy Color Computer 3 BASIC Quick Reference Guide* (1986), pp. 2–17. For saving and loading, see *Color Computer Disk BASIC Quick Reference Guide* (InterTAN Canada Ltd., 1987), pp. 3–8.

Your field guide

Appendix A
Come here if you remember what you wanted to do but have forgotten what BASIC calls it.

Find a command

When you want to…	Use	Try it in
Put words or a number on screen	`PRINT`	Give one instruction
See the program in memory	`LIST`	Moon Base
Start the program again	`RUN`	Moon Base
Wait for one key	`INKEY$` inside the program	Pick a planet
Ask for a number and wait for Return	`INPUT` inside the program	The crew quiz
Make a choice	`IF` and `THEN`	The crew quiz
Count trips through a loop	`FOR` and `NEXT`	Catch the computer
Find files on the CoCo disk	`DIR`	Tomorrow's message
Keep the program or bring it back	`SAVE` or `LOAD`	Tomorrow's message
Clear the program in RAM	`NEW`, **after saving**	Tomorrow's message

When the screen surprises you

Typing goes nowhere. Look at the main emulator window. If the toolbar offers **Run**, the CoCo is paused. Resume it, then choose **Input → Focus CoCo Display**. If the BASIC editor is stopped at a breakpoint, use **Continue** there to keep your place in the program.

The planet menu waits; the quiz shows a question mark. The planet program watches for one key. Press 1 or 2. The quiz uses INPUT. Type a number and press Return so BASIC knows your answer is finished.

BASIC reports a syntax error. Compare the spelling, the quotes, and the number at the start of the line with the listing. Retype the whole numbered line. Run the program again. At the prompt, an instruction without a number takes effect as soon as you press Return.

The disk seems empty. Check that the selected profile is Disk BASIC and that FD0 shows your *practice* disk's Mac filename. Use DIR to see the names inside it. A new disk image and a copy of the initialized starter disk are different things. These lessons use a copy. Go back to the safe copy steps before writing.

The screen says ?WP ERROR. Stop before clearing RAM. Check the practice disk's name and write protection in Floppy Disks. The program you tried to save is still in RAM. Fix the destination and try the save again.

Files you can open

Reading in **Help → Manual Library…**? Click a filename to open an editable BASIC example beside the book. If you're reading in a browser, save the file to your Mac and choose **Open…** in **Develop → BASIC Development**. BASIC's LOAD command on the green screen looks *inside the mounted CoCo disk*.

- BEACON.bas – the first two lines from Moon Base.
- planet.bas – the two-color Earth or Mars starter.
- SIGNAL.bas – the three-blink counter.
- QUIZ.bas – the completed crew quiz with its original title.

Words from the workbench

Input is information going in, like the key you pressed. **Output** is information coming out, like the greeting you saw. A **prompt** is a sign the computer is ready for you.

An **instruction** tells a computer to do something. A **program** gives it a set of instructions to follow. The **CPU**, or central processing unit, carries out machine instructions. The **interpreter** is the program that understands your BASIC and directs the work.

RAM is read/write working memory. Your beacon lived there while you edited it. **ROM** is read-only memory. It holds built-in software, including the interpreter that understands PRINT. A **bit** holds one of two values. Eight bits make a **byte**.

A **file** is a named collection of data you can come back to. A CoCo **disk drive** reads and writes files on a disk. VCC Plus puts the contents of a whole disk in a **disk image**, one file on the Mac.

A **variable** has a name and holds a value. A **loop** repeats instructions. A **breakpoint** tells the debugger to pause before a chosen line runs. The debugger's **watch** shows a variable's value while the program is stopped.

On the bookshelf

Appendix B

The CoCo in this book has two histories. One belongs to the Tandy computer people took home in the 1980s. The other belongs to the people who made it possible to sit down at that keyboard again on a Mac.

If you want to go further

For more projects with color, loops and choices, open Programming in CoCo BASIC on the Color Computer 3. If you are curious what the processor did between two blinks, follow a byte inside the 6809. For disks, tapes and cartridges, carry a program between devices. See the manual library.

Already inside VCC Plus? **Develop → BASIC Quick Reference** gives you command syntax and small examples. You can keep your own program open while you look something up.

Bibliography

- CoCopedia contributors. “Color Computer 3.” *CoCopedia*, revision 11134. Community history of the machine and its hardware.
- Tandy Corporation. *Introducing Your Color Computer 3*. 1986. “To Our Customers” and the introduction to the hardware.
- Tandy Corporation. *Tandy Color Computer 3 BASIC Quick Reference Guide*. 1986. Commands, pp. 2–17; functions, pp. 18–22.
- *Color Computer Disk BASIC Quick Reference Guide*. InterTAN Canada Ltd., 1987. Disk commands, pp. 3–8; errors, p. 9.

The Color Computer Archive preserves historical documentation that informed this book. The lessons and programs here were written for VCC Plus.

About the pictures

The hardware drawing uses VCC Plus’s original artwork. The application credits explain its origins and permissions. Artwork credits.

The window pictures show VCC Plus on a Mac. A CoCo screen capture shows just what the guest computer draws. Sizes vary with your Mac window and display settings. The commands and results are also given as text so you can read them at your preferred size.

Publication and artwork notices

Original VCC Plus prose and instructional images © 2026 Brian K. Holman. Original component and example notices remain with their respective material.

Except where otherwise identified, original artwork owned by Brian K. Holman is © 2026 Brian K. Holman. All rights reserved.

Permission is granted to use and redistribute this artwork unchanged with VCC Plus, including modified versions of its GPL-licensed software. Other reproduction, modification or redistribution requires prior written permission, except as permitted by applicable law. Third-party materials remain subject to their respective licenses and rights.

The CoCo 3 and CM-8 monitor presentation in VCC Plus's skeuomorphic System view was inspired by SockMaster's "pic of CoCo 3 and CM-8 monitor," published on CoCopedia. VCC Plus's artwork was designed and rendered independently. The remaining hardware illustrations were independently designed using historical photographs and renderings as references.

Read this book online

CMOC license notice · LWTools COPYING notice