# Terrapin - Apple Interface User's Manual

Deborah G. Tatar

Terrapin, Inc.
678 Massachusetts Ave.
Cambridge, MA 02139
(617)492-8816

## General Introduction

These instructions should tell you everything you need to know about how to set-up, test and begin using your Terrapin-Apple Interface and Turtle. Each section begins with an overview of the essential instructions contained within that section and continues with details and explanations.

## Parts List

Inside this box you will find two items in addition to these instructions:

1) your power supply -- a small heavy black box
2) a flat white cardboard box containing:
   a) a CCS Parallel Interface card with special ROM -- a flat circuit board packed in foam rubber
   b) the connecting ribbon -- a flat cord about 1 1/2 inches wide
   c) a booklet entitled Owner's Manual for a Parallel Interface Card
   d) two small black rectangular ROM (Read-Only Memory) chips with pointed silver leads

You only need parts 1, 2a and 2b to assemble and run the Turtle with the Terrapin - Apple Interface.

Items 2c and 2d should be stored some place safe in case you ever want to use your Parallel Interface card with a printer. The two chips (items 2d above) that you found packed in the foam rubber below your Interface card were originally on the card. Terrapin has replaced them in order to let you run a Turtle. However, it is possible to reinstall them, and the Owner's Manual... tells you how to manage the Interface card when those chips are in place.

## To Set Up

Basically, all you have to do is plug the connecting ribbon in at both ends, that is, into the Interface card and into the power supply, then place the Interface card in slot 7 of the Apple and plug the power supply into a wall socket. The following instructions are to make sure you get the right ends plugged in the right way. Make especially sure to get the marked side of the ribbon lined up with the column of pins marked 1 on the connector of the Parallel Interface.

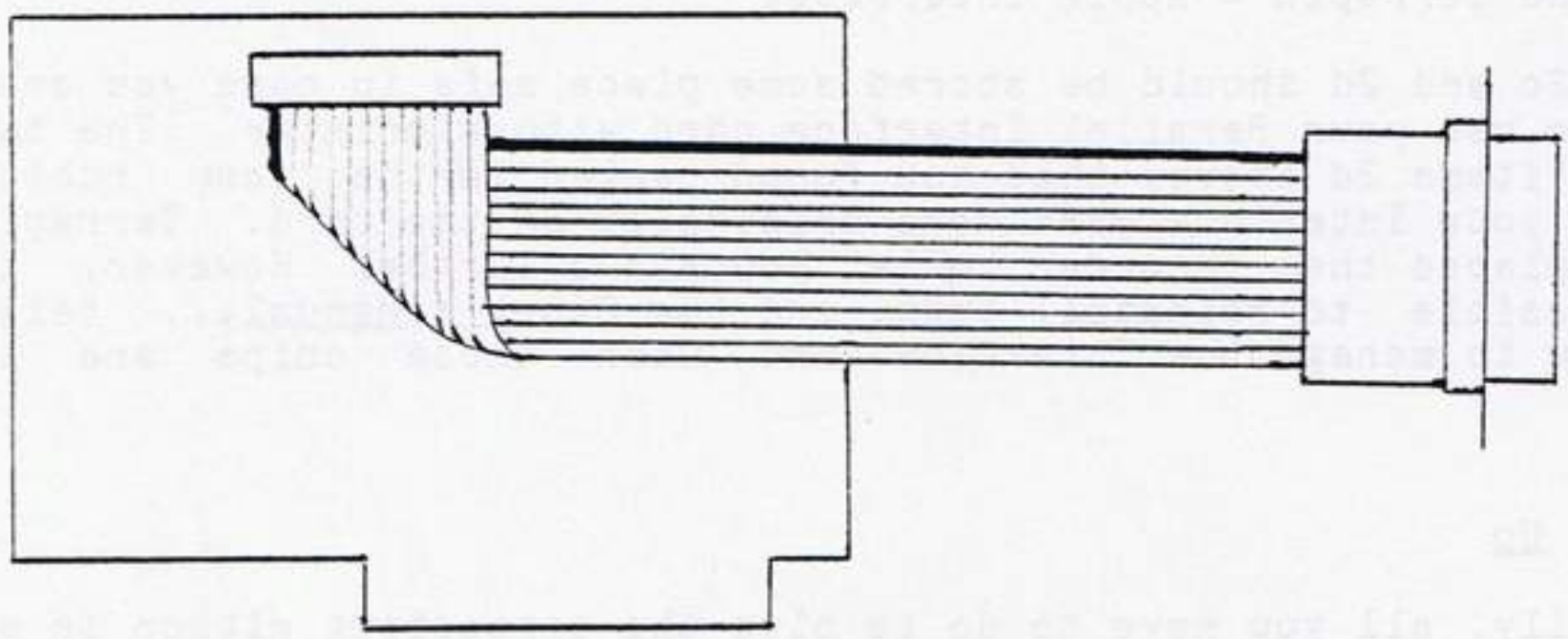
Before you start, turn the power off. Take the lid off your Apple. Touch the large gold box inside to discharge any static.

1) Find the flat ribbon and the Interface card. Turn the Interface so that the letters PI (for Parallel Interface) are upright and read correctly. In the top center of the card you will now find a silver-colored rectangle made of two rows of

small densely-packed pins. This is a connector. Underneath the first and last column of pins, you will find a small number printed on the board, labeling the pins 1 and 25. Locate number 1. Now find the small plastic end of the connecting ribbon. There will be either a colored strip on one side of the cable, or a triangular mark or some other tick that will distinguish between one side of the small end and the other. The side that is marked should be lined up with the column of pins marked 1 on the Interface and the ribbon should be (carefully!) plugged in. Be careful not to bend the pins -- make sure the sockets and pins are aligned. It requires a firm squeeze to slid into place. It is not completely plugged in if you can still see the metal connecting pins. There is usually a small snapping sound as it finally slips into place.

2) Orient the ribbon so that when you put the card into the computer, the ribbon will fit relatively smoothly into the slot in the back wall. To do this, fold the ribbon underneath itself and to the right. Do not crease the fold.

Illustration 1



Now plug the card into slot 7 in the computer. As you stand in front of the computer, this is the last slot on the right. The side of the card with the ribbon attached to it should face the right wall of the computer. There is an opening in the back wall of the computer behind slot 7. You should be able to slide the ribbon into it as you lower the card into place. Again, although one must exercise caution not to bend any components, it requires some firmness to get the card into the slot.

Now put the lid back on your Apple.

**********
**********

EUROPEAN MODEL Apples take the Interface card in slot 5 instead of slot 7. Therefore, whenever the slot number is invoked--in programming, as well as installation--owners of European model Apples must substitute 5 for 7.

**********
**********

3) Find the small (6.5" x 4.5" x 2.5") heavy black box that is your power supply. It has two ends. The front end is distinguished by a switch, and the back end by a small screw with a slot in the end. Put the power supply on top of the Apple so that the front end faces the keyboard. Now plug the loose end of the connecting ribbon, which should be hanging out of the slit behind slot number 7, into the back end of the power supply. The plug is not symmetrical; if it won't go in, turn it over and try again. Don't crease the cable.

4) Now plug the Turtle's cord into the front end of the power supply. Again, the plug is not symmetrical; if it won't go in, turn it over. You may want to secure the hooks on either side of the plug over the connection; however, you don't need to.

5) Finally, plug the power supply into a standard wall socket.

## To Test the System

To test your Turtle and Interface card, you need to run each Turtle command and test all the touch sensors. The following programs show you how to do this.

If you plan to use your Turtle in teaching, this first experience can provide valuable data. Now how you approach the Turtle the first time. Your experience can be helpful when introducing the Turtle to the students.

### Part 1: Checking the Turtle Commands

Flip the on-off switch on your power supply to the on position. It should glow red. If it doesn't, make sure you have plugged your power supply into a wall socket.

Turn on your Apple. Type in the following program: (1)

---

(1) To stop this program, press control-C when it asks for a command. Should you need to stop the turtle immediately, press the RESET key.

```
1Ø PR#7 : PRINT "*"
2Ø PR#Ø
3Ø INPUT "TURTLE COMMAND?"; A$
4Ø PR#7 : PRINT A$
5Ø GOTO 2Ø
RUN
```

**********
**********

European Model Apples---substitute PR#5 for PR#7 in the above program.

**********
**********

This program asks for and executes Turtle commands. It prints out the words "TURTLE COMMAND?" When you type in your instructions and press RETURN, the Turtle should exhibit the desired behavior and the words "TURTLE COMMAND?" should appear once more.

The Turtle has thirteen built-in commands, falling into two categories: motion commands and auxiliary commands.

Motion commands:

| Command | Means | Example | Produces |
|---|---|---|---|
| F <some#> | Forward <some#> | F 1Ø | Turtle moves Forward 1Ø inches |
| B <some#> | Back <some#> | B 1Ø | Turtle moves Back 1Ø inches |
| R <some#> | Right <some#> | R 9Ø | Turtle turns 9Ø degrees Right |
| L <some#> | Left <some#> | L 9Ø | Turtle turns 9Ø degrees Left |
| M <some#> | Right wheel forward <some#> left wheel still | M 9Ø | Turtle pivots 9Ø degrees forward around left side |
| S <some#> | Left wheel forward <some#> Right wheel still | S 9Ø | Turtle pivots 9Ø degrees around right side |
| Q <some#> | Right wheel back <some#> Left wheel still | Q 9Ø | Turtle pivots 9Ø degrees back about left side |
| K <some#> | Left wheel back <some#> Right wheel still | K 9Ø | Turtle pivots 9Ø degrees back around right side. |

In other words, each letter command must be followed by a numerical argument, i.e. you must tell the Turtle how far to go forward or how much to rotate. For example, F 1Ø should make

your Turtle move forward 1Ø turtle units (roughly 1Ø inches) and R 9Ø should make it turn right 9Ø degrees.

None of the <some#>s may be greater than 255; larger numbers will produce generally unpredictable results.

Notice that rotating right or left is caused by both wheels turning in opposite directions. The four pivot commands, M, S, Q and K, involve only one wheel moving. Right and left rotations cause the center to stay still, while in pivoting one of the wheels remains stationary. All turning is caused by the relationship between the actions of the right and left motors.

If your Turtle does not behave as expected, there are several possible reasons. If it does nothing at all, you should double-check all the connections. Then try turning the screw on the power-supply all the way first in one direction, then in the other.

You will be told how to correct any curves produced by Forward or Back and any deficiencies in the amount of rotation produced by turn commands in the Adjustments section.

Hint: Try to use a smooth floor rather than a rug.

Auxiliary commands:

| Command | Means |
|---|---|
| P <some#> | Pen-- Numbers > Ø put the pen down, Ø raises the pen |
| E <some#> | Eyes--Numbers > Ø make the eyes glow red, Ø turns them off |
| H <some#> | Horn--Numbers greater than Ø turn the horn on, Ø turns it off |
| T <some#> | Tone--Numbers > Ø switch the horn higher Ø restores its original tone |
| O | Off-- Raises the pen and turns off the eyes, horn and motor. Can take <some#>, but does not require one. |

To get out of the program you typed at the beginning of this section, press control-C. If you wish to start it up again, type RUN.

## Part 2 --Checking the Touch Sensors

Your Turtle has four touch sensors, left, right, back and front. To check them out, type in the following program:

```
10 PR#7 : PRINT "*"
20 PR#0
30 DEF FN TUCH(DUMMY) = 255-PEEK (49392)
40 X = FN TUCH(0)
50 PRINT X
60 GOTO 40
RUN
```

When you press the left side of your Turtle, a series of 1s should appear on the screen. The right side should produce 2s, the front 4s, and the back 8s. These numbers are additive, so that pushing the right and the front will produce 6s, the left and back 9s and so on. Different combinations of one or two sensors being activated permit recognition of contact in any of 8 directions.

**********
**********

European model Apple users-- remember: use PR#5 instead of PR#7 for the PR#. Also, the PEEK number is not 49392 but 49360.

**********
**********

Again, to get out of this program, type control-C.

Adjustments

There are two ways to adjust your Turtle. The first is mechanical, and the second computational. Mechanical adjustments include the slotted screw on the back of your power supply which adjusts both motors equally, and the four screws in the small blocks (called "pots") on top of your Turtle which allow you to address the forward and back functions of each motor separately. Computational adjustments allow you to modify any Turtle motion through changing the delay increment in the Apple's memory.

The slotted screw on the back of your power supply adjusts both motors equally. Use this to adjust the amount of rotation per unit command. Test for the correct setting by turning your Turtle some constant amount such as 90 degrees. For example, type R 90 to the program listed on page 5 and then adjust the screw. Stop when the Turtle actually moves 90 degrees. (This change will also affect forward/backward movement slightly.)

Additionally, you can control the amount that each motor turns in either direction. Because the adjustments are separate for each motor, you can use them to correct any curvature in Forward or Back commands. (If the motors are not working equally, you will get curvature.) Because the adjustments are separate for Forward and Back motion, you must be sure to check both functions; it is possible to have both motors working equally to go Forward 10,

but not for Back 1Ø.

To make the adjustment, unscrew the dome of your Turtle. On top, in a line just front of the center, you will find four rectangular plastic blocks with small screws in them. They are the pots (short for "potentiometers") referred to above. Next to them on the printed-circuit board is written FD or BK. The two on the left (from the Turtle's point of view) affect the left motor and the two on the right, the right motor. The inside screws adjust the forward motion and the outside screws, the backward. Turning any screw towards the center of the Turtle causes the speed of that particular motor and direction to increase. For example, turning the rightmost screw (leftmost as you face it) toward the center causes the right motor to increase speed in the backwards direction. If your Turtle draws a curve when you command it forward or back, these adjustments allow you to make sure that your Turtle goes in a straight line, i.e. that the motors are working equally. You can also use these commands to ensure that the Turtle moves the same amount forward and backward per unit command.

Occasionally, the Turtle will move in a tight circle, or move intermittently when commanded to move forward or back. There is a slight possibility that a wheel has worked loose and needs to be tightened with the allen wrench which is supplied.

The computer does not actually measure how far the Turtle goes when it moves but rather how long the motors are on. Therefore, you can adjust the delay constant stored in the Apple's memory. This number represents the amount of time the computer allows the motor to be on for each unit of command. There are three constants which allow you to do this, FCONST, TCONST and ACONST affecting respectively, forward/backward motion, left/right commands, and K/M/S/Q commands. Initially, they are set to the highest constant possible. If this does not work out, you may lower them.

The way to do this in your programs is to follow the initialization of the Interface card ( PR#7 : PRINT "*") with lines that express the desired modification. Just as the initialization must be repeated each time the Apple has been turned off or the RESET button pushed, so to must these modifications be added each time to the original program.

FCONST controls the amount of time the motor is on per unit command in forward and backward motion. It is normally set at 255. To change it, you must type in: POKE 1535, <some#-between-1-and-255>.

ACONST controls the amount of time the motor is on in right or left commands. It is normally set at 87. To address it type: POKE 1663, <some#-between-1-and-255>.

TCONST controls the amount of time the motor is on for the commands K, M, S and Q. It is normally set at 62 and may be changed by typing: POKE 1791, <some#-between-1-and-255>.

When trying to determine how much to change these constants, remember that the effect of an adjustment is not linearly proportional to the numerical change; the effect is proportional to the square of the ratio. For example, typing in POKE 1663, 174 will make turning angles not two times greater, but four times.

Format

Programming for your Turtle is just like any other programming in BASIC except for a few special considerations.

Initialization: every time you begin to use the Turtle or press the RESET button run a program beginning with the following lines:

```
10 PR#7 : PRINT "*"
20 PR#0
```

PR#7 tells the computer to send commands to slot #7 (where you inserted the Interface card). PRINT "*" initializes the control program on the Interface card. PR#0 tells it to make PRINT print on the tv-screen again.

Line 10 should be included in every program that is written for the Turtle. (1)

RESET: Pressing RESET has nearly the same effect as typing PR#0. You can press RESET at any time, even if the turtle is running. It will stop the turtle (and your program) immediately, and redirect printing to the screen. Note that pressing RESET while the turtle is running may disable DOS commands; type CALL 1002 if attempts to use DOS commands result in "SYNTAX ERROR." Turning the computer off and back on will fix most unusual conditions.

Touch-Sensors: Note that "TUCH", not "TOUCH" must be used in the touch-sensor example program because "to" is a restricted word in BASIC, and appears in the word "touch."

Touch-sensor numbers: Memory location 49392 (49360 on European Apples) is associated with the Turtle's touch sensor. PEEKing this location in BASIC returns a number from 0 to 255. If the turtle not touching anything, the number will be 255. For convenience, the TUCH function provided in the program on page 6 inverts these numbers so that PRINT FN TUCH(0) prints 0 when the

---

(1) European model Apples--here as always #5, not #7.

Turtle isn't touching anything. Here are the numbers returned by FN TUCH(Ø) when various sensors are touching something:

| | |
|---|---|
| Left | 1 |
| Right | 2 |
| Front | 4 |
| Back | 8 |

When two sensors are on, the number will be the sum of the two values for the respective sensors.

Other Programs: Turtle functions may be included in any program, even programs that also do entirely different things, by intializing the Interface card and giving a standard Turtle command of the form PR#7: Print "<somecommand> <some#>". A typical subroutine that could take an input produced by the rest of the program might look like this:

```
1ØØ PR#7 : PRINT "*"
11Ø PRINT TC$;TA : PR#Ø
12Ø RETURN
```

To use this as part of another program, assign the desired command to the string variable TC$, and the argument to the numeric variable TA. Then GOSUB 1ØØ.

## Using Your Turtle

In general, you will probably want to use your Turtle in conjunction with the programs on pages 4 and 6 that allow you to give direct motion commands and receive direct feedback from the touch-sensors; however, it is easy to modify these programs and create new ones.

The program listed below is the one you used to test motion commands. It prints out the words "TURTLE COMMAND?", receives an input, executes that input, and requests a new input.

```
1Ø PR#7 : PRINT "*"
2Ø PR#Ø
3Ø INPUT "TURTLE COMMAND?" ; A$
4Ø PR#7 : PRINT A$
5Ø GOTO 2Ø
RUN
```

It accepts the commands discussed in the Testing section with the amounts outlined there:

| Command | Means | Example | Produces |
|---|---|---|---|
| F <some#> | Forward <some#> | F 1Ø | Turtle moves Forward 1Ø inches |
| B <some#> | Back <some#> | B 1Ø | Turtle moves Back 1Ø inches |
| R <some#> | Right <some#> | R 9Ø | Turtle turns 9Ø degrees Right |
| L <some#> | Left <some#> | L 9Ø | Turtle turns 9Ø degrees Left |
| M <some#> | Right wheel forward <some#> left wheel still | M 9Ø | Turtle pivots 9Ø degrees forward around left side |
| S <some#> | Left wheel forward <some#> Right wheel still | S 9Ø | Turtle pivots 9Ø degrees around right side |
| Q <some#> | Right wheel back <some#> Left wheel still | Q 9Ø | Turtle pivots 9Ø degrees back about left side |
| K <some#> | Left wheel back <some#> Right wheel still | K 9Ø | Turtle pivots 9Ø degrees back around right side. |
| P <some#> | Pen-- Numbers > Ø put the pen down, Ø raises the pen | | |
| E <some#> | Eyes--Numbers > Ø make the eyes glow red, Ø turns them off | | |
| H <some#> | Horn--Numbers greater than Ø turn the horn on, Ø turns it off | | |
| T <some#> | Tone--Numbers > Ø switch the horn higher Ø restores its original tone | | |
| O <some#> | Off-- The letter "O" can be followed by a number, but does not require one. | | |

An example of something you might like to do for young children is to create an environment in which a single letter command produces an entire movement, i.e. instead of typing F 1Ø all the child would have to do is press F or any other key (perhaps with an arrow taped to it) to produce a forward movement of some predetermined amount. Such a program could look like this:

```
1Ø PR#7 : PRINT "*"
2Ø PR#Ø
3Ø PRINT "WHICH WAY?"
4Ø GET A$
5Ø C$ = "Z"
6Ø IF A$ = "F" THEN C$ = "F 1Ø"
7Ø IF A$ = "B" THEN C$ = "B 1Ø"
8Ø IF A$ = "R" THEN C$ = "R 3Ø"
9Ø IF A$ = "L" THEN C$ = "L 3Ø"
1ØØ IF C$ = "Z" THEN 13Ø
11Ø PR#7 : PRINT C$
12Ø GOTO 2Ø
13Ø PRINT "WRONG LETTER, PLEASE TRY AGAIN"
```

```
140 GOTO 20
RUN
```

Another program causes the Turtle to move or turn away from whichever sensor is activated :

```
10 PR#7 : PRINT "*"
20 DEF FN TUCH(DUMMY) = 255 - PEEK (49392)
30 X = FN TUCH(0)
40 IF X = 4 OR X = 5 OR X = 6 THEN PRINT "B 10"
50 IF X = 8 OR X = 9 OR X = 10 THEN PRINT "F 10"
60 IF X = 1 THEN PRINT "R 90"
70 IF X = 2 THEN PRINT "L 90"
80 GOTO 30
```

Be creative! May you find your Turtle intellectually stimulating, and lots of fun!

## Appendix

### Using the Terrapin - Apple Interface from Assembly Language

The following information is a sample program and listing of machine locations relevant to the use of the Terrapin - Apple interface with 6502 machine-language programs.

A full listing of the assembly language program that runs the Turtle is available from Terrapin at no cost.

$47F Command: Contains the current command character in ASCII. the MSB is used as a flag for argument accumulation (the MSB should be low for all commands except "*" (INIT)).

$4FF Argument: Contains a binary integer that determines the delay for the motor commands.

$57F AUX: contains a bit pattern that preserves the state of the auxiliary functions (Eyes, Pen, Horn, Tone).

The contents of these three locations regulate the software delay for Turtle motor commands:

$5FF FCONST: Integer constant that determines delay for F(orward) and B(ackward) (normally $FF).

$67F ACONST: Controls delay for L(eft) and R(ight) commands (normally $3E).

$6FF TCONST: Delay for K, M, Q, and S commands (normally $57).

```
TITLE SAMPLE
;A sample assembly language program for controlling the Turtle.

FCONST=$5FF
ACONST=$67F
TCONST=$6FF

COMMND=$47F
ARGMNT=$4FF
AUX=$57F

SNDCMD=$C7ØØ          ;Turtle routine on Terrapin - Apple Interface Card.

;Call this routine to initialize the interface.
INIT:     LDA #$8Ø+'*
          STA COMMND
          JSR SNDCMD
          LDA #$8Ø+$ØD
          JSR SNDCMD          ;Send the command to the turtle.
                               (* for initialization).
;Store any desired values of the motion delay constants in FCONST, TCONS
;and ACONST at this point.
          RTS

;Call this routine with the command name in A, the argument
;in X, and the bit pattern for auxiliary devices in Y.
;(See the Turtle owner's manual for bit masks.)
DOCMDA:   STY AUX
;...
;Enter at this point to keep the same auxiliary bits.
DOCMD:    AND #$7F            ;Turn off MSB, in case it's present.
          STA COMMND
          STX ARGMNT
          LDA #$8Ø+$ØD
          JMP SNDCMD          ;Do it.
END
```