Topo™ Owner's Manual

Apple II, Apple II+ and Apple IIe Computer Version

ANDROBOT INC™

1287 Lawrence Station Road
Sunnyvale, California 94086
(408) 745-1084

Contents

SOFTWARE DISCLAIMER: Although we have made every attempt to verify the accuracy of this document and accompanying software, we cannot assume any liability for errors or omissions. No warranty or other guarantee can be given as to the accuracy or suitability of this software for a particular purpose, nor can we be liable for any loss or damage arising from the use of the same.

Background

In 1950, science fiction author Issac Asimov published a collection of earlier stories in a book called I, Robot. In the introduction of this book, Asimov stated that the major manufacturer of robots would come into existence in 1982.

He was right about the date but not about the company's name. Our name is Androbot, Inc., and like Asimov's U.S. Robot and Mechanical Men, Inc., we came into existence in 1982. There is, however, a major difference between the two companies -- we are real, and so are our products. Robots are no longer just science fiction!

So allow us to welcome you to a new era -- the robotic age!

Introducing Topo -- and a new age!

You have purchased a product that was pure science fiction a few short years ago -- a domestic robot. As you learn about your new robot (we have named him Topo from topography or topology), you will come to realize that you are a pioneeer in a totally new and exciting field. Whether your applications are practical, educational, or purely for entertainment, you will soon be making Topo do things that no one else has made him do before.

Regardless of your applications for Topo, we know one thing -- you will enjoy using him. Topo will seem like a family member as he scampers around the house. You can

even teach Topo to do some tricks. Unlike most new pets, however, Topo is already house-broken.

We know that you want to get Topo unpacked and on his way, and that you don't want to read this whole manual first.

Fair enough -- we want you to learn how to use Topo too.

In order to help make this process as practical as possible, we divided this manual into sections. Each section begins with a question (How do I unpack Topo?). You may want to look for the section that deals with the question that is foremost on your mind and proceed from there. This way, you will have Topo up and running in no time. Once Topo is connected to your computer and is operational, we encourage you to read this entire manual. Each section has valuable information.

There is one last comment to be made before you unpack Topo.

Topo is not a toy -- Topo is a full-fledged robot.

Topo uses industrial grade components throughout, ranging from the highest quality rechargeable batteries and a professional crystal controlled radio link, to high quality motors and cast aluminum gearboxes, a fabricated steel framework, and a high impact body. With proper care, Topo will provide years of trouble-free service.

And that matters to us because providing service to humans is what robots are all about!

What do I need to use Topo?

Your Topo robot needs a computer to send him instructions. The version you have purchased is designed to be used with the computers made by Apple, specifically with the Apple II, II+, or IIe.

To use Topo, you should have an Apple II, II+, or IIe computer system. If you are using cassette tape to store you programs, you will need 16K of RAM. If you have the disk system, you will need 48K of RAM. If you intend to use the Logo language to control Topo, you will need a full 64K of RAM.

To use the enclosed disk, you need one disk drive (16 sector) with the disk controller card plugged into slot #6.

You should also have a pair of paddle controllers or a smooth action joystick.

How do I unpack Topo?

Your Topo robot system comes packed in two boxes. If you have followed the directions on the boxes, the only one you have opened

at this time is the smaller box marked "Topo Apple Controller". As you unpack this box you will find:

this manual and owner registration cards,

Topo's battery charger (the black box with the built-in wall plug),

a plastic foam block to prop Topo up during calibration,

the Topo/Apple transmitter controller card (this is the printed circuit board with the row of contacts along one edge),

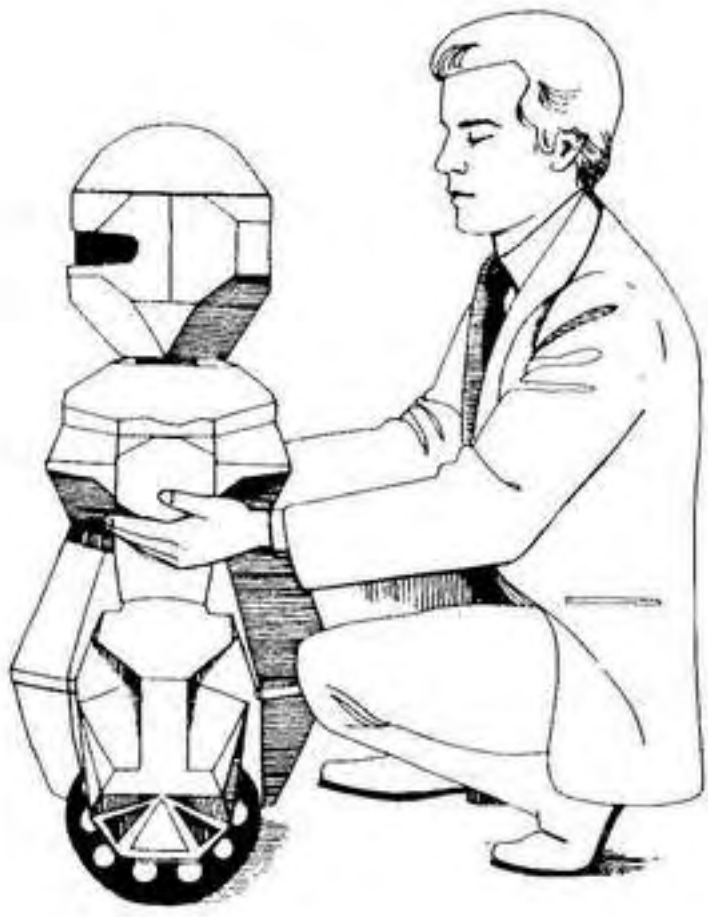
the Topo radio transmitter (a plastic case with a rod antenna and a cable coming out of one end),

and a floppy disk containing the TOPO joystick programs and a set of BASIC Topo controller routines.

Once you have verified that you have all these items, you are ready to open the larger box containing Topo itself. Topo has been very securely packed to avoid damage in transit. It is important that you unpack Topo carefully and that you save his box in case you ever want to ship him somewhere.

To unpack Topo, carefully open the top flaps of the box and slide out the top foam

packing insert. You should be able to see Topo as you look in the box. In lifting Topo out of the box, be careful to not lift him by his head. Neither robots nor human beings should ever be lifted by their head! Instead, reach down around both sides of Topo. Midway down his body you will feel two flat areas under which you can hook you hands. Gently lift Topo from the carton and gently set him on the floor. Never drop Topo! Even though he can take a lot of rough and tumble action, Topo can be damaged if he is dropped onto a hard surface.



Once Topo is out of his box, continue to lift him by grabbing him underneath the shoulder area with both hands as shown in the illustration. Topo can also be lifted from underneath his base.

Remember to never lift Topo by his head. If you look near the bottom of Topo, you can see that he moves with the aid of two large wheels positioned at an angle to each other. This is the proper orientation for these wheels. Topo balances in an upright position of his own accord, and if you give him a slight push, you will see that Topo rocks back and forth a little bit without falling over. This rocking motion is normal and keeps Topo from damaging itself or anything into which he might run by accident.

Now check Topo over to be sure he hasn't been damaged in transit. In the unlikely event that Topo has been damaged, immediately contact your authorized Androbot dealer for instructions.

Gently move Topo closer to an electrical outlet so you can charge his batteries. While Topo's batteries are being charged overnight for the first time (as shown in the next section), you should fill out and mail the enclosed registration card (we will send you a FREE Topo T-SHIRT in exchange!), and use this time to read the rest of your manual and connect the radio link to your Apple computer.

How do I charge Topo's batteries?

To charge Topo's batteries, you should plug in the cable for the battery charger in the connector located on Topo's back control panel. This panel is located near the bottom of Topo and contains a power-on button, a power-off button (which should be pressed), a power-on light, a battery status light, and the battery charger connector. After connecting the charger cable to Topo, plug the charger housing into a standard 115V 60 cycle power outlet.

When Topo is operating, the battery status light should be green, indicating that the batteries are still charged. If this light turns red, Topo's batteries are low and Topo should be plugged into his charger. The batteries can be completely recharged overnight. Once charged, they will provide many hours of use before needing recharging. When Topo is not in use, we suggest that you leave him plugged into his charger to keep his batteries ready.

Your Topo has special battery protection circuitry. The overcharge protection circuit allows a full charge to be maintained by leaving the charger connected, with no possibility of damage. The discharge protection circuit prevents the batteries from being damaged by fully discharging. If you continue to operate Topo after the battery status light turns red, the protection circuit will engage automatically and turn him off. If this happens, simply charge Topo overnight.

How do I get Topo running for the first time?

Installing the transmitter controller card...

If this is the first time you are using Topo, you will have to insert the transmitter controller card in the Apple computer. This interface circuit is the printed circuit board that was packed with the transmitter.

To install this board, carefully remove the lid from your Apple computer (BE SURE THE POWER TO YOUR APPLE COMPUTER HAS BEEN TURNED OFF FIRST). As you look inside the Apple computer, you will see a series of connectors along the back part of the Apple circuit board. Each connector has a number printed next to, starting with 0 next to the power supply (the long metal box) and running to 7 as you move from left to right. If you are using the disk memory system with the Apple, you should refer to your Apple instruction manual and make sure that the disk controller interface is installed in connector #6. Next, you are ready to install the transmitter controller card. To do this, you must first connect it to the cable from the transmitter unit. This cable has a connector on its end that should be plugged into the corresponding socket on the rear edge of the transmitter controller card. This connector will only fit one way, and it will slip into place without much force. If the connector doesn't seem to fit, turn it around and try again.

The Topo/Apple transmitter controller card should be inserted in the Apple computer connector #5 in such a manner that the circuit components on the card are facing away from the power supply and that the transmitter cable runs out the back side of the computer through one of the available slots.

If the card is inserted backwards, it will fail to work, so be sure that you insert this circuit in connector #5 with the components facing away from the power supply. Be sure this card is properly seated in the connector. You can press it in firmly if the connector is stiff. You won't break anything.

Once you have completed this installation procedure, replace the lid to your Apple computer and snap it into place.

Locating the transmitter...

You should locate the transmitter module in a convenient out-of-the-way place near the computer. Be sure that the rod antenna is extended fully, and that the transmitter isn't located in a place where it can be easily bumped. The antenna should point straight up, and should not be blocked by any large metal or electronic articles such as televisions, etc. To get maximum range for your Topo robot, you will want to be sure that the transmitter is in an unobstructed location.

The transmitter derives its power from the

Apple computer and should require little attention on your part. When using your Apple computer for tasks that don't involve Topo, be sure the switch on the front of the transmitter unit is turned off.

Using Topo with the joystick or paddle controllers...

Now that you have installed the transmitter controller in your computer and have charged Topo's batteries, you are finally ready to get Topo moving! The easiest way to do this is with the JOYSTICK program that is stored on the diskette. To use this program you must have an Apple-compatible joystick or a pair of paddle controllers plugged into the Apple game controller port.

When turning Topo on for the first time, you will want to prop him up so his wheels are free to rotate without touching the floor. There is a foam plastic block in the bottom of Topo's shipping box that is designed for just this purpose.

Once you have set Topo on his perch, (and are sure his power is turned off), insert the TOPOBASIC diskette in your computer's #1 disk drive (with the label facing upward and towards the front of the drive) and turn on your computer system. In a few seconds, you will see the following message on the display:

ANDROBOT TOPOBASIC VERSION 1.1

(C) COPYRIGHT 1983, ANDROBOT, INC. 2-1-83

(LOADING TOPO DRIVER PROGRAM)

(LOADING TOPOBASIC PROGRAM)

When the BASIC prompt (]) has appeared on the screen, enter:

GOSUB 6000

This activates the transmitter interface card and enables your Apple to talk to the transmitter (for the beginner: always press the RETURN key after entering each line on your Apple keyboard to tell it that you are through typing the line). Now type:

GOSUB 5000

When you press the RETURN key, the program will run and you will gain manual control of Topo. The screen will display:

RUNNING JOYSTICK .....
PRESS ANY KEY TO STOP

Follow these instructions:

1. Locate the joystick handle near the center position. If you are using the paddle controllers instead, set both knobs near the center setting.

2. Turn on the transmitter unit (using the switch on the transmitter cabinet).

3. Turn Topo on by depressing the green button on the control panel (the red button will turn him off).

4. If his wheels are turning, adjust the joystick or paddle controllers so both wheels have stopped.

5. Now you can take Topo off his perch and set him on the floor.

Place Topo in an area where he will be able to travel at least a few feet in any direction. As you slowly push the joystick handle forward, Topo will start to move forward. You can steer Topo by pushing the handle to the right or left. To make Topo go backwards, pull the handle towards you.

If you are using the paddle controllers, you will find that one controller adjust Topo's forward and backward motion, and that you can steer with the other controller.

Now that you have seen how to make Topo move and steer with the joystick or paddle controllers, you should practice moving him around the room to get the feeling for his motion and control. WHENEVER THE TRANSMITTER IS TURNED OFF, TOPO WILL STOP MOVING. There is no harm in turning the transmitter on or off whether a program is running or not. By pressing any key, you may enter new commands such as the Topo

commands described in the next section or return to the joystick mode by typing in again:

```
GOSUB 5000
```

The program is written in such a way that one main direction of the joystick handle will make Topo go forward and backwards and the other direction will make it go left and right. If your joystick responds in a reverse manner, that is, it switches left for right, you can swap the sides by typing in:

```
5050 DI = 255
```

To revert to the original mode, use the statement:

```
5050 DI = 0
```

Operating Topo with a joystick is a fine way to get started, but an even more powerful way is to write your own programs that teach Topo some complete routines to follow.

The next section shows some ways of doing this using some programs we provided on the diskette and the Applesoft Basic language in your Apple computer. You should now make a copy of your diskette for daily use and store the original in a safe place.

How can I make Topo operate on his own?

The disk that was packed with your Topo Apple controller card contains all the BASIC subroutines you will need for creating your own robot control programs.

When you first turn on your Apple computer and the screen displays the title lines, all the necessary routines to run Topo under program or joystick control are loaded in. The machine language routines are not visible but do reside in memory. The BASIC routines start at statement number 5000 and extend to numbers past 7300. You can see them all by typing:

```
LIST
```

Your Apple will respond by scrolling the program on the screen. To temporarily stop the scrolling, type CTRL S (for the beginner: that means you hold down the CTRL key and press the S key). To continue, use CTRL S again. If you want to stop the listing process completely, type CTRL C and you get the BASIC prompt back. The system contains the following routines:

BASIC_ROUTINES

Line number		Function
5000	JOYSTICK	Makes joystick run Topo
5100	TFD:	Moves Topo forward
5200	TBK:	Moves Topo back
5300	TLT:	Turns Topo left
5400	TRT:	Turns Topo right
5500	TSTOP:	Stops Topo
5600	TSTEP:	Makes Topo step
5700	TSTEPSTOP:	Makes Topo step, stop
5800	TCOUNT:	Generates a delay
5900	TSET:	Sets all channels
6000	TRESET:	Resets all channels
6300	HOLD:	Stores channel values
6450	STORESTP:	Stores step value
6500	TFDX:	Extended forward move
6600	TBKX:	Extended back move
6700	TLTX:	Extended left turn
6800	TLTX:	Extended right turn
7000	INIT:	Sets initial values
7200	CALIBRATE NEUTRAL VALUES	
7300	CALIBRATE LINEAR MOTION	

Remember: You need to activate the transmitter interface card after you run on the computer before you can already, type GOSUB 6000. Now you can use typed-in commands to run Topo. For example, to go forward 200 steps, enter the following:

```
N = 200
GOSUB 5100
```

To turn 90 degrees to the right, type:

```
N = 90
GOSUB 5400
```

Try the other two motion commands, at lines 5200 and 5300 too. If you want to stop Topo while he is executing a command, hit any key with the joystick routine. Your Topo may not go very straight or turn very precise corners at this point, but don't worry, you can fix that later when we get to the calibration procedures. For now, just have some fun with him!

Writing Topo programs

Remember, in order for the Apple to talk to the transmitter after power-up, you have to type in the following line once:

```
GOSUB 6000
```

Now you are ready to write a program that can combine many of the commands that we have entered one by one in the immediate mode. A program to control Topo always has to start with a call to the INIT subroutine. You will find that writing programs for Topo is a snap. Let's start with something quite simple -- making Topo move forward and backwards by 50 centimeters. The program to do this is shown below:

```
10 GOSUB 7000:  REM INIT
20 N = 50
30 GOSUB 5100:  REM TFD
40 GOSUB 5200:  REM TBK
50 END
```

Turn on Topo and the transmitter and type RUN. There will be a slight pause and then

Topo will move forward 50 centimeters, stop, and move backwards 50 centimeters. You should be able to see how this program works by looking at the listing. Line 10 initializes Topo. Line 20 sets the distance we want Topo to travel. Line 30 uses the routine at line 5100 -- this routine moves Topo forward by N centimeters. Line 40 moves Topo back by this amount by using the routine at line 5200

The key to using the routines for linear movement (5100 and 5200) and turning (5300 and 5400) is to be sure and set the value you want used in N before using the routine. Otherwise, the movement or turning procedures will use the previously set value for N.

Now let's see how Topo turns! Enter the following program:

```
10 GOSUB 7000:  REM INIT
20 DIST = 50
30 ANGL = 90
40 FOR I = 1 TO 4
50 N = DIST : GOSUB 5100:  REM TFD
60 N = ANGL : GOSUB 5400:  REM TRT
70 NEXT I
80 END
```

When you run this program, Topo will move in a square path! The routine at line 5400 turns Topo to the right. You should modify the program above to have Topo turn to the left (using the routine at line 5300) and see how that works.

What about triangles, pentagons and other paths? We can write one program to let Topo move in any of these paths. To see how we do this, enter the following program:

```
10 GOSUB 7000: REM INIT
20 PRINT "ENTER STEP SIZE":
30 INPUT DIST
40 PRINT "ENTER NUMBER OF SIDES":
50 INPUT A
60 ANGL = 360 / A
70 FOR I = 1 TO A
80 N = DIST:GOSUB 5100:  REM TFD
90 N = ANGL:GOSUB 5400:  REM TRT
100 NEXT I
110 GOTO 20
120 END
```

When you run this program, you can enter any step size and any number of sides you want. Topo will do exactly as you ask, and then will wait patiently for your further instructions! You can stop the program with CTRL C.

You should do some experimenting on your own at this point. There are lots of projects you could do with Topo. For example, set up a small obstacle course with a series of evenly spaced chairs and write a program to have Topo weave his way around them.

Write a program to make Topo dance!

The following program makes Topo move in the pattern of a five-pointed star:

```
10 GOSUB 7000:  REM INIT
20 DIST = 50
30 ANGL = 144
40 FOR I = 1 TO 5
50 N = DIST:GOSUB 5100:  REM TFD
60 N = ANGL:GOSUB 5300:  REM TLT
70 NEXT I
80 END
```

You should also try the extended motion commands TFDX, TBTX and TRTX in lines 6500, 6600, 6700 and 6800 for smoother motion and better precision.

The choreographic repertoire of Topo is by no means limited to what you can do by combining the four simple motion commands. Routines TSTEP, TSTOP, TSTEPSTOP, TCOUNT AND TSET give you the freedom to control the speed of each wheel at any given instant. Topo can dance through smooth curves just as well as he can draw angles. For example, the following routine makes Topo go round and round in a circle, clockwise:

```
10 REM CIRCLES
20 GOSUB 7000: REM INIT
30 C1 = 1500:C2 =1250
40 GOSUB 5900:  REM TSET
50 END
```

Topo will keep going until you give it a TSTOP command by entering GOSUB 5500. Now draw one circle:

```
10 REM CIRCLE
20 GOSUB 7000: REM INIT
30 STP = 600:C1 = 1500:C2 = 1250
40 GOSUB 5700:  REM TSTEPSTOP
50 END
```

You may need to adjust STP to make a perfect circle. Next, make Topo do a counterclockwise circle of the same size. The program is the same except for line 30 which may be something like:

```
30 STP = 650:C1 = 1280:C2 = 1500
```

Note that you may not get exactly the same values in both cases and that it takes some trial and error to get a good combination. Now combine the two circles for a figure eight:

```
10 REM FIGURE 8
20 GOSUB 7000: REM INIT
30 STEP = 600:C1 = 1500:C2 = 1250
40 GOSUB 5600: REM TSTEP
50 STP = 650:C1 = 1280:C2 = 1500
60 GOSUB 5700: REM TSTEPSTOP
70 END
```

If you want Topo to keep doing the figure eight, replace the last two lines with:

```
60 GOSUB 5600: REM TSTEP
70 GOTO 30
80 END
```

Let your imagination be your guide -- the possibilities for Topo are endless!

How do I calibrate Topo?

The routines provided on the TOPOBASIC disk include two types of motion commands. The easiest commands to use are the ones with which we are already familiar: TFD, TBK, TLT, and TRT. TOPOBASIC also has a set of extended commands that give Topo smoother motion. These commands are TFDX, TBKX, TLTX, and TRTX. The extended commands are harder to calibrate than the first set, but they do give more accurate motion, so they might be worth the extra effort, once you are completely comfortable with Topo's easy commands.

This section includes instructions on calibrating Topo for use with the simple commands. The calibration procedures for the extended commands are covered in the advanced section at the end of this manual.

Topo can be calibrated in two steps:

1. Find the zero settings for the motors.

2. Calibrate the turning angle and movement increment.

Zeroing the motors...

If you look at the listing of the INIT routine, you will find these lines:

```
7005 Z1 = 1024
7010 Z2 = 1024
```

Z1 and Z2 are the neutral settings for each drive wheel motor. While the value 1024 is close to the right value for your robot, you may want to adjust these values so they are exactly right. To do this, you will use the calibration routine at line 7200.

Prop Topo up on his perch so his wheels are free to rotate are aren't touching the floor. Next enter the line:

```
7005 Z1 = A
```

This temporarily allows the value of Z1 to be set by the calibration program while we zero the left motor. Type:

```
GOSUB 7200
```

Turn on Topo, and turn on the transmitter. If the footlights on Topo's left wheel are either both off or both flickering, then the wheel is zeroed. If only the front lights are on, the wheel has a tendency to go forward. Type a number smaller than 1024, say 1010 in response to the question on the screen and press RETURN. If the back lights come on, you have gone too far and the wheel now has a tendency to go backwards. Type in a number slightly larger the the previous one. Repeat until the wheel is zeroed, and stop the program by entering 0. Suppose you have found number 1020 to work best.

Now enter:

```
7005 Z1 =1020
```

(or whatever you found to be right), and repeat the process for the other wheel. To do this, you should change line 7010 to:

```
7010 Z2 = A
```

and run the program. Once you have zeroed the right wheel, retype line 7010 with this new value, e.g.,

```
7010 Z2 = 1026
```

This completes the zeroing procedure.

Calibrating distance and angle...

Topo will move and turn by different amounts depending on the suface on which he is running. A carpeted floor will produce results different from a tile floor, for example. Also, Topo's inertia will influence the precision of his motion. Low batteries will affect Topo's performance too. If his battery indicator shows red, or if he has been running for an extended period, give Topo a good charge before attempting to calibrate him. He'll be glad you did, and he will behave better, too!

The routine at line 7300 can be used to determine the speed values for Topo to go in a straight line. We will start with forward motion. Set Topo on the floor where he will be able to go a few feet in any direction. Type:

```
GOSUB 7300
```

In answer to the questions on the screen, enter 1536 for both wheels. Topo will go a few feet forward. If he veers to one side, determine which wheel is going faster than the other. You need to decrease the drive on that wheel, in answer to the same questions displayed again on the screen. For example, if Topo veers to the left, then the right wheel is too fast. On the next set of questions, input 1536 for the left and 1450 for the right wheel. For reliable performance, do not exceed 1536 on either wheel.

After a few trials, you will find the set of values that balance the forward speeds. Suppose they are 1536 (left) and 1480 (right). Save these in the INIT routine by entering:

```
7025 F1 = 1536
7030 F2 = 1480
```

Topo now goes straight when moving forward. You can adjust the distance factor SFD to make him go a known distance for a given input. SFD is nominally 1.15 for Topo to go one centimeter for a unit input. For example, if you run:

```
10 REM DISTANCE TEST
20 GOSUB 7000: REM INIT
30 N = 200
40 GOSUB 5100: REM TFD
50 END
```

Topo should travel 2 meters (approximately 6 feet). If he goes too far, you can try decreasing SFD to 1.05 for example. To do

so, change the value of SFD in the INIT routine with:

```
7085 SFD = 1.05
```

Run the program again. After a few trials, you will find the correct value of SFD.

Now repeat the calibration procedure for backwards motion, starting with speed values 512 on both wheels. Note that you decrease the backward speed by increasing numerical value. Store these in the variable B1 and B2, and adjust the distance travelled with SBK. Again, for reliable performance, do not use values below the limit of 512.

The left turn angle is adjusted with SLT. Make Topo do a 360 degrees turn by the following program:

```
10 REM ANGLE TEST
20 GOSUB 700: REM INIT
30 N = 360
40 GOSUB 5300: REM TLT
50 END
```

Change SLT as necessary with, for example:

```
7095 SLT = .48
```

Increasing SLT will increase the amount of turn. After a few trials, you will get the correct value for SLT. Repeat the same procedure for the right turn by adjusting SRT. This completes the steps for calibration of the regular simple motion commands. The extended commands are

calibrated in a similar manner, shown in the Details section of this manual. You may want to go into those procedures when you have gained more familiarity with your Topo.

Once you have customized the calibration values of your robot, delete any undesired programs you have entered (such as the short test routines) and type:

SAVE TOPOBASIC

This saves your new values on the disk so they will be loaded automatically each time the computer is turned on.

At this point you should make a back-up copy of your working diskette and save it together with the original Androbot TOPOBASIC diskette.

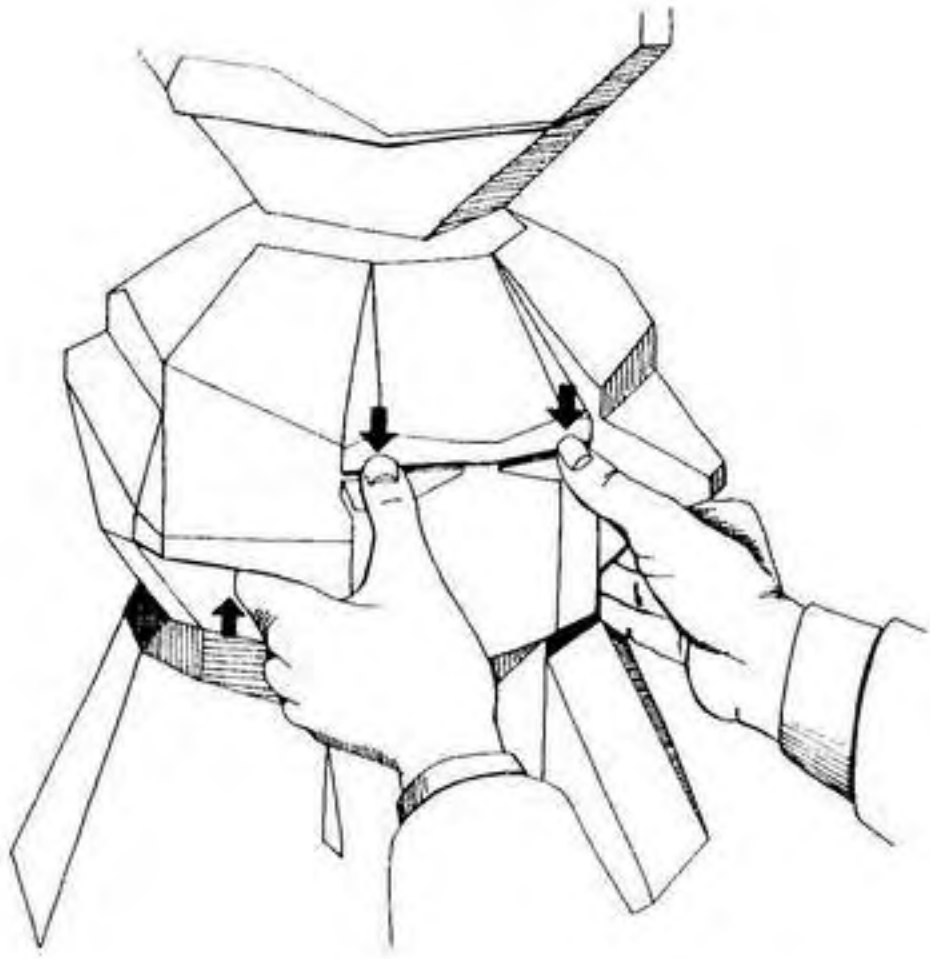
One note of caution: variable values are not saved, only program statements are saved. To preserve new calibration values, be sure to record the values in the INIT routine, for example with:

7085 SFD = 1.12

Simply typing SFD = 1.12 will change the current value, but it will not be preserved on disk.

What_kinds_of_things_can_I_make_Topo_do?

As we showed in the previous section, you can make Topo move around the house in repsonse to commands you give him from your computer keyboard. You can make Topo do useful things, such as carry snacks from the kitchen to the living room. You can make Topo do fun things such as dance. You can even use Topo to help teach geometry to children!



Opening Topo's storage compartments is a snap! With your hands in the position shown, gently press down with your thumbs until the latch opens and the panel swings down. The same hand motion is repeated when you are ready to close the latch.

The scope of Topo's capabilities is as broad as your own imagination. But in order to get Topo to do anything, you must learn how to have the computer give him instructions. Your instructions will take the form of a computer program. To make this task easier for you, we have developed three ways for you to create your own Topo instructions. The first of these uses the language Applesoft BASIC, and is covered in the previous section.

The other two ways involve the use of more powerful computer languages called Logo and FORTH. Both of these Androbot products will be available from your dealer. Either of these last two languages has the advantage that you will be able to quickly create programs that not only send Topo on a trip, but which draws a picture of this trip on your Apple display screen at the same time!

Features in the Topo extension to Logo ...

Logo is a computer language that is both easy to learn and almost limitless in its application. Of particular value to Topo owners is a feature of Logo called "Turtle Graphics". Turtle graphics lets you create pictures on the display by sending messages to an imaginary "turtle" as he draws lines on the screen. To draw a straight line 100 units long, you would give the command FORWARD 100. To turn the turtle by 90 degrees, you would enter RIGHT 90. By using the commands FORWARD, BACK, RIGHT and LEFT, you can send the display turtle on almost any imaginary path imaginable. For

example, a square path 50 units on a side would be created by the command REPEAT 4 [FORWARD 50 RIGHT 90].

While turtle graphics is of temendous power in its own right, Androbot has created its own modifications for Logo that allows these same commands to send your Topo robot on trips around the house. By using the Logo procedures provided by Androbot, anyone -- child or adult -- can be creating dances, games, maze following programs, or even learning geometry in a new way.

Both the screen image and Topo can be controlled separately or simultaneously in several modes including using the joystick, instant repsonse to key presses, and programmed commands. In addition, a TEACH mode lets the computer remember a sequence of commands you have entered with key strokes and save these commands on your disk under a name of your choosing. By just typing this name, Topo will be instructed to carry out this sequence of commands. The Androbot extensions to Logo include calibration routines and a set of demonstration programs for getting you up to speed quickly.

The Androbot extensions to the Logo language provide one of the most powerful ways of putting Topo to work.

Features in TopoFORTH ...

TopoFORTH is a version of the FORTH computer language designed especially for use with Topo. It has Logo-like screen turtle graphics and provides the same functions as described above for the Androbot extensions to Logo.

Experienced computer users will be able to take advantage of the extremely versatile and efficient FORTH programming environment underlying the Topo commands. FORTH can run very fast and its memory requirements are modest, so that there is ample room left in the Apple for your own programs. TopoFORTH includes calibration routines and a set of demonstration programs for getting you up to speed quickly.

What do I do if Topo doesn't work?

The programs won't load ...

Be sure that:

the disk drive controller card is located in the Apple expansion slot #6.

the dooor to the disk drive is closed after inserting the disk.

the disk is inserted properly, with the label facing up and to the front of the drive.

Topo doesn't move ...

Be sure that:

the power switch on Topo is turned on.

the power switch to the transmitter is turned on.

Topo's batteries are charged.

Topo is less than 75 feet from the transmitter.

the transmitter antenna is extended and is not blocked by any large metal objects.

the Topo controller card is plugged into slot #5 on the Apple

computer with the components facing away from the power supply.

the joystick or game paddle controller is plugged in properly.

Topo moves erratically ...

Be sure that:

the power switch to the transmitter is turned on

Topo is within 75 feet of the transmitter

Topo's batteries are fully charged

the transmitter antenna is fully extended

the charger is not connected to Topo

What do I do if Topo gets into mischief?

Once you have followed the calibration procedures that make Topo's motors respond properly to your commands, Topo will do exactly what you ask him to do.

Unfortunately, you might ask Topo to do something that might cause him damage, or cause him to bump into someone or something. Because you are just learning how to pilot Topo around the house, and are just learning what kinds of motions Topo can make, it is a very good idea for you to follow Topo around as he goes on his excursions.

If you see Topo starting to do something you don't want him to, you can stop him immediately by pressing down gently on the top of his head. This motion activates a switch that turns Topo off.

The computer will have the transmitter continue to send Topo commands, but Topo will just stand there waiting to be turned on. Of course, Topo will not respond or even remember any of the commands you sent him while the power is off, so you will want to go back to the computer and modify your program before turning Topo back and and re-submitting it.

Remember to press gently on Topo's head to activate the emergency OFF switch. Especially be careful to not bang Topo's head -- after all, he is only following your instructions. If you aren't too comfortable with your skills as a Topo

programmer, you might want to work out your Topo tasks with pencil and paper before trying them with the real thing!

How should I take care of Topo?

Topo has been designed to require a minimum of service. There are no user serviceable parts inside Topo's body, so all your care and attention should be devoted to Topo's exposed parts. If Topo's body gets dirty, you can clean him off with a soft cloth dampened with a mild soap. You should avoid using harsh soaps, abrasive cleansers, or solvents, as these may damage the finish on Topo's high-impact plastic parts. After washing the body, remove the soap with a cloth dampened with water and dry the surface with a cloth towel.

The drive wheels should be checked periodically to remove any carpet lint that may have accumulated around the axle. In checking Topo's wheels, be sure that Topo and the computer system are both turned off first and that you don't wedge your fingers in between the wheels and the housing. Any dirt that has accumulated near the axles can be removed with a dust cloth.

Topo is permanently lubricated during manufacuture, and should never need to be oiled. In fact, the major service Topo will require from you is making sure that his batteries are charged.

What are Topo's specifications?

Topo characteristics ...

height: 36.5
width 22"
depth: 17.5"
weight: 34.5 lbs.

storage temperature range: 32° F to 110° F ambient

operational temperature range: 45° F to 100° F ambient

Note: Topo will feel comfortable where you feel comfortable. Do not put him next to a fireplace or stove and do not leave him in direct sunlight.

humidity, storage and operational ambient: 8 - 80 %
internal power source: two 12 V sealed lead acid rechargeable batteries (6 amp/hours)
battery charger: 24 VAC transformer

radio receiver: 8 channel AM superhet 27.145 MHz
drive system: independent dc motors and gearboxes
maximum speed: 12 inches per second

Apple Topo controller and interface characteristics...

storage temperature range: 32° F to 110° F ambient

operational temperature range: 45° F to 100° F ambient

humidity, storage and operational ambient: 8 - 80%

interface card slot number (for Apple computer): 5

interface card power requirement: 5 VDC @ 270 mA and 12 VDC @ 35 mA provided by the computer

transmitter power requirement: 9 VDC @ 15 mA provided by the interface card

transmitter characteristics: 100 mW 4 channel AM 27.145 MHz

transmitter certification number: pending FCC approval

Limited Warranty

Who is protected

Your Androbot warranty protects the original purchaser and all subsequent owners, during the warranty period, from any failure as a result of an original manufacturing defect so long as the original dated bill of sale is presented whenever service is required during the warranty period.

How long is the warranty period?

All Androbot products have a warranty period of 90 days for labor and one year for parts.

What does the warranty cover?

Except as specified below, the Androbot warranty covers all defects in original materials and workmanship. The following are not covered: Damage caused by accident, misuse, abuse, neglect, product modification, damage occuring during shipment; damage caused by failure to follow the instructions in your owner's manual, including failure to perform routine maintenance; damage resulting from repairs by someone not authorized by Androbot; or claims based upon any misrepresentations by the seller.

Who pays for what?

During the period that both parts and labor are covered by this warranty, Androbot will

pay all the labor and material expenses to repair a warrantable defect; during the period the parts ONLY are covered by this warranty, Androbot will pay for all materials to correct a warrantable defect, but you must pay for labor charges.

How_can_warranty_service_be_obtained?

In the unlikely event your Androbot product requires service, you should first contact the dealer from whom the product was purchased. If this is not practical, contact us at Androbot, Customer Service Dept., 1287 Lawrence Station Rd., Sunnyvale, CA 94086 (415-745-1084). We may direct you to an authorized service center or ask that you return the product to use for repair. In any event, you will have to present your original dated bill of sale to establish warranty coverage. Do not send us your product without prior authorization.

You are responsible for transporting your product for repair and for payment of shipping charges. However, Androbot will pay return shipping charges if the repairs are covered by the warranty. All robot products must be returned in the original shipping carton to prevent damage during transport.

LIMITATION OF IMPLIED WARRANTIES: All implied warranties, including fitness for a particular purpose and merchantability are limited in duration to the length of the warranty period for your product.

LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES: Androbot is not responsible for any incidental or consequential damage of any kind. Our liability is limited to the repair or replacement, at our option, of a defective produc

Some states do not allow limitations on how long an implied warranty lasts and/or do not allow the exclusion of incidental or consequential damage, so the above limitations or exclusions may not apply to you.

This warranty gives you specific legal rights and you may also have other rights that vary from state to state.

Details for advanced Topo programmers

Let's take a look at the programs that were automatically loaded into your Apple when the power was turned on. The first nine pages of the Listings section show the BASIC routines that you can also see on the screen by LISTing out, and the last four pages show the assembly language source listing of the binary routines residing at location $300. There are twenty or so BASIC program modules as listed above, all configured as subroutines that are called from your main program or from each other. Some of them make use of the binary routines for talking to the interface card.

Aside from JOYSTICK, the routines that you will be using most frequently are the simple motion commands: TFD, TBK, TLT and TRT (regular forms), and TFDX, TBKX, TLTX and TRTX (extended form). They take as input the variable N representing the number of steps, and also use some motion variables that are set in the INIT routine. These motion variables can be changed for calibrating Topo.

Your computer controls Topo through four channels whose values are contained in variables C1, C2, C3, and C4. These variables may have values in the range of 512 to 1536. All channel values are transmitted once every 20 ms, so there are 50 control frames per second. Channels C3 and C4 are for future expansion. Channels C1 and C2 control the left and right wheels respectively. For example, when C1 = 512, the left wheel rotates at full speed

backwards. When C1 = 1024, it is stopped and when C1 = 1536, it turns at full speed forward. Channels C3 and C4 are spares that may be used later for optional accessories. The variables Z1, Z2, Z3 and Z4 contain the four neutral values, nominally 1024. The routine TRESET uses them to reset all channels. To stop one wheel, say the left, we set C1 = Z1. The routine TSET may be used to communicate that information to the transmitter.

When Topo goes forward at full speed under a TFD command, channels C1 and C2 are set to the proper running values that make both wheels roll forward at the same speed, C1 = F1 for the left and C2 = F2 for the right wheel. If medium or low speed are desired, then one can set C1 = E1 and C2 = E2, or C1 = E3 and C2 = E4 respectively. Similarly, for backwards motion, variables B1; B2 represent full speed and A1, A2 or A3, A4 represent medium and low speed. These values are fine tuned during calibration.

When Topo starts forward on a TFDX command with input N, he first goes at medium speed for so many steps (FUP), then uses full speed for the required number of steps (a function of input N), then goes back to low speed for another number of steps (FDN) and finally stops. This achieves smooth start and stop by ramping the speed.

On a left turn under a TLTX command, no ramping is needed, but the number of steps taken is derived from the input number after accounting for wheel slippage and inertia effects with factor CL.

How to calibrate Topo completely

If Topo veers steadily while travelling full-forward (TFD or TFDX).

Variable F1 controls the left wheel forward; F2 controls the right wheel forward. Determine which wheel needs to be rotating slower and decrease its value until Topo goes straight, or increase the opposite wheel. For reliable performance, do not increase above the factory limit of 1536.

If Topo veers steadily while travelling full-backward (TBK or TBKX).

Variable B1 controls the left wheel backward; B2 controls the right wheel backward. Determine which wheel needs to be rotating slower and increase its value until Topo goes straight, or decrease the opposite wheel. Note that increasing the value will slow down the wheel, which is the opposite effect from the forward adjustment. For reliable performance, do not decrease below the factory limit of 512.

If Topo travels more or less than the commanded distance (TFD, TBK only).

The distance standard is 1 unit = 1 cm. Thus, 100 forward should cause Topo to travel 100 cm, or 1 meter. Variables SFD and SBK are scale factors for forward and backward, respectively. Increasing either

factor will increase the distance travelled. For example, if you request 100 backward and Topo travels 110 cm, then decrease SBK until Topo travels 100 cm.

If Topo turns more or less than the commanded amount (TLT, TRT only).

Variable SLT is the left turn scale factor; SRT is the right turn scale factor. Increasing either factor will increase the amount that Topo turns for any given command. Thus, for example, if you request a 90-degree right turn and Topo only turns 85 degress, then increase SRT until Topo turns 90 degrees.

Further calibration of the extended commands (TFDX, TBKX, TLTX, and TRTX).

If Topo travels straight, but veers during starting or stopping (TFDX or TBKX only).

There are four pairs of variables here - (E1, E2) for forward starting, (E3, E4) for forward stopping, (A1, A2) for backward starting, and (A3, A4) for backward stopping. For each pair, the first is for the left wheel and the second is for the right.

The procedure is actually nearly identical to the full-speed veer adjustment. For simplicity, we will only discuss one case -- say forward stopping (controlled by E3-left and E4-right). Determine which wheel

needs to rotate faster during forward stopping, and increase its value until Topo stops straight. (Or, decrease the opposite wheel. The choice depends on whether you like slower or faster stopping.)

The same procedure applies to forward starting. A similar procedure works for backward values, except that increasing the value will slow down the wheel, which is the opposite to the forward adjustment.

For reliable performance, E1-4 should never exceed the factory limit of 1536, and A1-4 should never be less than the factory limit of 512.

If Topo travels more or less than the commanded distance (TFDX or TBKX only).

Calibrating distances with the extended BASIC commands TFDX and TBKX requires several distinct steps; fortunately each step is quite easy. We will only describe the procedure for forward calibration; the identical steps are used for backward settings.

First, set the minimal ramp distance RF (RB for backward). To do so, set the distance (N) EQUAL to the CURRENT value of RF (whatever it may be) and give a forward (TFDX) command. This causes a minimal ramp. Measure the distance Topo travels and set RF to this value (if Topo moves 47 cm, then set RF = 47). This completes the first step.

Second, set the scale factor XFD (XBK for backward). To do this, give a large distance command (at least 100 or so, the larger the better) and measure the distance travelled by Topo. Increasing the scale factor will increase the distance travelled. Thus, if you request forward 300 with XFD and Topo travels 330 cm, then decrease SFD until Topo travels 300 cm. This completes the second step.

Lastly, adjust the short-distance slippage constant FS (BS for backward) if desired. This is only used to adjust short-distance steps (less than the minimum ramp RF) where the ramping is automatically disabled. Give a distance command just less than the value of RF, and measure how far Topo travels. If Topo comes short of the commanded distance, then increase FS by the difference. Thus, if you command forward (TFDX) 41 and Topo moves 38 cm, then increase FS by 3. This completes the distance calibration procedure for TFDX and TBKX.

What if I want to vary the ramping characteristics? (TFDX or TBKX only).

The variables E1-4 and A1-4 (discussed in "If Topo travels straight, but veers during starting or stopping") control the MOTOR SPEEDS during start-up and slow-down. These can be adjusted to vary the ramped start and stop speed characteristics. Again, increased values will slow down the motors for backward commands.

The variables FUP, FDN, BUP, and BDN control the TIME DURATION of the aforementioned motor speeds, as follows: FUP - forward start (ramp up), FDN - forward stop, BUP - backward start, and BDN backward stop. Although not generally adjusted, they can be varied if desired. A typical value of 30 represents 30/50 or 0.6 seconds duration of ramping time.

If Topo turns more or less than the commanded amount (TLTX or TRTX only).

Variable XLT is the left turn scale factor; XRT is the right turn scale factor. Increasing either factor will increase the amount that Topo turns for any given command. Thus, for example, if you request a 90-degree right turn and Topo only turns 85 degrees, then increase XRT until Topo turns 90 degrees.

If Topo turns 90 degrees correctly, but other angles are off (TLTX or TRTX only).

In addition to XLT and XRT (left and right scale factors described earlier), you can also adjust CL and CR. These are left and right correction factors, respectively. Proceed as follows: Give a 360-degree (1 full turn) command, and adjust XLT (or XRT) as appropriate until Topo turns 360 degrees. Now give four separate, consecutive 90-degree turn commands. If this overshoots 360 degrees, then increase CL (or CR) until four 90-degree turns gives exactly the same as one 360-degree turn. You may need to slightly re-adjust XLT(or XRT) during this process. This procedure

will result in good performance for all angles.

If I change one variable, do I have to readjust the others?

Although small adjustments will usually not produce noticeable effects elsewhere, some large variable changes can affect other nominal settings. Generally, you need not worry about recalibration unless you notice an obvious shortcoming.

How to relocate the Transmitter Controller card...

The standard TOPOBASIC program assumes that the transmitter interface card is plugged into peripheral slot #5 in the Apple computer, in the first statement of TRESET:

```
6005 CD = 49361:DT = 49360
```

You may relocate this card to any other slot but slot #0 by changing the values of the two variables, CD and DT. Here is a table of the values to use:

SLOT #	CD	DT
1	49297	49296
2	49313	49312
3	49329	49328
4	49345	49344
5	49361	49360
6	49377	49376
7	49393	49392

GLOSSARY_OF_BASIC_ROUTINES

TRESET - 6000 (Transmission RESET)
Resets the interface card to its wake-up state, with all control channels set to neutral values. Must be executed once after system power-up to start data transmission to Topo.

JOYSTICK - 5000
Puts Topo under joystick control. To exit, hit any key on the keyboard. Uses none of the calibration variables.

TFD - 5100 (Topo ForwarD)
Takes input N and makes Topo go forward N steps at full speed, typically going N centimeters, then stop. Uses speed variables F1 and F2, and distance factor SFD.

TBK - 5200 (Topo BacK)
Takes input N and makes Topo go backward N steps at full speed, typically going N centimeters, then stop. Uses speed variables B1 and B2, and distance factor SBK.

TLT - 5300 (Topo LefT)
Takes input N and makes Topo turn N steps to the left at full speed, typically going N degrees, then stop. Uses Speed values B1 and F2, and angle factor SLT.

TRT - 5400 (Topo RighT)
Takes input N and makes Topo turn N steps to the right at full speed, typically going N degrees, then stop. Uses speed values F1 and B2, and angle factor SRT.

TFDX - 6500 (Topo ForwarD eXtended)
Similar to TFD, but with smoother and more precise motion. Takes input N and moves Topo forward N centimeters. Correction factors FS and RF together with XFD adjust the distance travelled. Variables FUP, FDN, and E1-E4 control orientation and smoothness during starting and stopping.

TBKX - 6600 (Topo BacK eXtended)
Similar to TBK, but with smoother and more precise motion. Takes input N and moves Topo backward centimeters. Correction factors BS and RB together with XBK adjust the distance travelled. Variables BUP, BDN, and A1-A4 control orientation and smoothness during starting and stopping.

TLTX - 6700 (Topo LefT eXtended)
Similar to TLT, but with more precise turning for any given angle. Takes input N and turns Topo N degrees to the left. Correction factor CL and scale factor XLT adjust the turning angle.

TRTX - 6800 (Topo RighT eXtended)
Similar to TRT, but with more precise turning for any given angle. Takes input N and turns Topo N degrees to the right. Correction factor CR and scale factor XRT adjust the turning angle.

INIT - 7000 (INITialize variables)
Resets all motion variables to their correct start-up values. Should be called at the start of each user program. Latest calibration values should be saved into this routine.

TSET - 5900 (Topo SET control values)
Sets Topo control channels to the values of C1, C2, C3, and C4. Appropriate values for C1 or C2 will engage the motors. Note that they will continue to run until a command is given to stop.

TCOUNT - 5800 (Topo COUNTer delay)
Takes as input the variable STP and generates a delay equal to that number of time frames (1/50 second per frame, so STP=50:GOSUB 5800 delays 1 second).

TSTEP - 5600 (Topo STEP one setting)
Combines the functions of TSET and TCOUNT. Sets control channels to C1-4, and waits for a number of frames equal to STP before returning. Note that the motors will not be stopped on return.

TSTOP - 5500 (Topo STOP)
Makes Topo stop. Uses neutral values Z1 and Z2. Typically used after a TSET or TSTEP.

TSTEPSTOP - 5700 (Topo STEP and STOP)
Combines the functions of TSTEP and TSTOP. Sets control channels, waits for a number of frames equal to STP, and then stops motors.

HOLD - 6300 (HOLD channel data)
Utility routine, used to pass values C1-C4 to the machine language Topo driver program.

STORESTP - 6450 (STORE STeP time value)
Utility routine, used to pass the value of

STP (step delay time) to the machine language Topo driver program.

CALIBRATE NEUTRAL VALUES - 7200
Interactive calibration routine for setting the neutral values Z1-Z4, one at a time.

CALIBRATE LINEAR MOTION - 7300
Interactive routine for setting the speeds on both wheels. Normally used to determine the values that yield straight-line motion.

GLOSSARY OF BASIC VARIABLES

The following variable names are reserved. The ones with descriptions can be calibrated by the user, others are used internally by the various routines. Remember that BASIC only "sees" the first two characters of any name.

AS
A1 - Backward left ramp speed, startup
A2 - Backward right ramp speed, startup
A3 - Backward left ramp speed, stopping
A4 - Backward right ramp speed, stopping
BDN- Backward ramp stopping time
BS Backward short distance slippage
BUP- Backward ramp starting time
B1 - Backward full speed, left wheel
B2 - Backward full speed, right wheel
CD
CH%
CL - Left turn correction factor
CR - Right turn correction factor
C1 - Channel one (left wheel) storage
C2 - Channel two (right wheel) storage
C3 - Channel three storage
C4 - Channel four storage
DI
DT
DL
DH%
E1 - Forward left ramp speed, startup
E2 - Forward right ramp speed, startup
E3 - Forward left ramp speed, stopping
E4 - Forward right ramp speed, stopping
FDN- Forward ramp stopping time
FS - Forward short distance slippage
FUP- Forward ramp starting time
F1 - Forward full speed, left wheel

F2 - Forward full speed, right wheel
HN%
HS%
H1%
H2%
H3%
H4%
LN
L1
L2
L3
L4
L5
N - Command variable set by user
RB - Minimum ramp distance, backward
RF - Minimum ramp distance, forward
SBK- Backward distance scale factor
SFD- Forward distance scale factor
SLT- Left turn scale factor
SRT- Right turn scale factor
STP- Step increment, 50 = 1 second
SYNC
XBK- Extended backward scale factor
XFD- Extended forward scale factor
XLT- Extended left turn scale factor
XRT- Extended right turn scale factor
Z1 - Channel one neutral value
Z2 - Channel two neutral value
Z3 - Channel three neutral value
Z4 - Channel four neutral value

```
5000  REM JOYSTICK
5005 AS = 1023
5010 H3% - Z3 / 256
5015 L3 = Z3 - (H3% * 256)
5020  POKE (AS - 6),L3
5025  POKE (AS - 7),H3%
5030 H4% = Z4 / 256
5035 L4 = Z4 - (H4% * 256)
5040  POKE (AS - 8),L4
5045  POKE (AS - 9),H4%
5050 DI = 0: REM 255 INVERTS JOY
     STICK
5055  POKE 888,DI: POKE 928,DI
5060  PRINT
5065  PRINT "RUNNING JOYSTICK....
     ."
5070  PRINT "PRESS ANY KEY TO STOP"
5075  CALL 853: REM BINARY JOYST
     ICK
5080 RETURN : REM

5100  REM TFD
5105 STP = N * SFD
5110 C1 = F1:C2 = F2
5115  GOSUB 5700: REM TSTEPSTOP
5120  RETURN : REM
```

```
5200  REM TBK
5205 STP = N * SBK
5210 C1 = B1:C2 = B2
5215  GOSUB 5700: REM TSTEPSTOP
5220  RETURN : REM
5300  REM TLT
5305 STP = N * SLT
5310 C1 = B1:C2 = F2
5315  GOSUB 5700: REM TSTEPSTOP
5320  RETURN : REM

5400  REM TRT
5405 STP = N * SRT
5410 C1 = F1:C2 = B2
5415  GOSUB 5700: REM TSTEPSTOP
5420  RETURN : REM

5500  REM TSTOP
5505 C1 = Z1:C2 = Z2
5510  GOSUB 5900: REM TSET
5515  RETURN : REM

5600  REM TSTEP
5605  GOSUB 6300: REM HOLD
5610  GOSUB 6450: REM STORESTP
5615  CALL 775: REM BINARY TSTEP

5620  RETURN : REM
```

```
5700  REM TSTEPSTOP
5705  GOSUB 6300: REM HOLD
5710  GOSUB 6450: REM STORESTP
5715  CALL 768: REM BINARY TSTE
    PSTOP
5720  RETURN : REM

5800  REM TCOUNT
5805  GOSUB 6450: REM STORESTP
5810  CALL 807:REM BINARY TCOU
    NT
5815  RETURN : REM

5900  REM TSET
5905  GOSUB 6300: REM HOLD
5910  CALL 785: REM BINARY TSET
5915  RETURN : REM

6000  REM TRESET
6005 CD = 49361:DT = 49360
6010 CH% = CD / 256
6015 CL = CD - (CH% * 256)
6020 DH% = DT / 256
6025 DL = DT - (DH% * 256)
6030  POKE 800,DL: POKE 801,DH%
6035  POKE 827,CL: POKE 828,CH%
6040  POKE 834,CL: POKE 835,CH%
6045  GOSUB 7000: REM INIT
6050 SYNC = Z1 + Z2 + Z3 + Z4 + 2
    560
6055 H1% = Z1 / 256
6060 H2% = Z2 / 256
```

```
6065 H3% = Z3 / 256
6070 H4% = Z4 / 256
6075 HS% = SYNC / 256
6080 L1 = Z1 - (H1% * 256)
6085 L2 = Z2 - (H2% * 256)
6090 L3 = Z3 - (H3% * 256)
6095 L4 = Z4 - (H4% * 256)
6100 LS = SYNC - (HS% * 256)
6105  POKE CD,255: POKE CD,95
6110  POKE CD,23: POKE DT,5
6115 AS = PEEK (DT)
6120 AS = PEEK (DT)
6125 AS = PEEK (DT)
6130  POKE DT,244: POKE CD,1
6135  POKE DT,251: POKE DT,13
6140  POKE DT,HS% POKE DT,251
6145  POKE DT,1: POKE DT, H1%
6150  POKE DT,251: POKE DT,1
6155  POKE DT,H2%: POKE DT,251
6160  POKE DT,1: POKE DT,H3%
6165  POKE DT,251: POKE DT,1
6170  POKE DT,H4%: POKE CD,21
6175 AS = PEEK (DT)
6180  POKE DT,98: POKE DT,172
6185  POKE DT,LS: POKE DT,98
6190  POKE DT,244: POKE DT,L1
6195  POKE DT,98: POKE DT,244
6200  POKE DT,L2: POKE DT,98
6205  POKE DT,244: POKE DT,L3
6210  POKE DT,98: POKE DT,244
6215  POKE DT,L4: POKE CD,233
6220  POKE CD,127: POKE CD,25
```

```
6225 AS = 1013: POKE AS,LS
6230  POKE (AS - 1),HS%
6235  POKE (AS - 2),L1
6240  POKE (AS - 3),H1%
6245  POKE (AS - 4),L2
6250  POKE (AS - 5),H2%
6255  POKE (AS - 6),L3
6260  POKE (AS - 7),H3%
6265  POKE (AS - 8),L4
6270  POKE (AS - 9),H4%
6275  RETURN : REM

6300  REM HOLD
6305 SYNC = C1 + C2 + C3 + C4 + 2
     560
6310 AS = 1023
6315 H1% = C1 / 256
6320 H2% = C2 / 256
6325 H3% = C3 / 256
6330 H4% = C4 / 256
6335 HS% = SYNC / 256
6340 L1 = C1 - (H1% * 256)
6345 L2 = C2 - (H2% * 256)
6350 L3 = C3 - (H3% * 256)
6355 L4 = C4 - (H4% * 256)
6360 LS = SYNC - (HS% * 256)
6365  POKE AS,LS
6370  POKE (AS - 1),HS%
6375  POKE (AS - 2),L1
6380  POKE (AS - 3),H1%
6385  POKE (AS - 4),L2
6390  POKE (AS - 5),H2%
```

```
6395  POKE (AS - 6),L3
6400  POKE (AS - 7),H3%
6405  POKE (AS - 8),L4
6410  POKE (AS - 9),H4%
6415  RETURN : REM

6450  REM STORESTP
6455 HN% = STP /256
6460 LN = STP - (HN% * 256)
6465  POKE 1002,LN + 1
6470  POKE 1003,HN% + 1
6475  RETURN : REM

6500  REM TFDX
6505  IF N > = RF THEN GOTO 652
    0
6510 STP = (N + FS) * XFD
6515  GOTO 5110
6520 C1 = E1:C2 = E2
6525 STP = FUP
6530  GOSUB 5600: REM TSTEP
6535 C1 = F1:C2 = F2
6540 STP = (N - RF) * XFD
6545  GOSUB 5600: REM TSTEP
6550 C1 = E3:C2 = E4
6555 STP = FDN
6560  GOSUB 5700: REM TSTEPSTOP
6565  RETURN : REM
```

```
6600  REM TBKX
6605  IF N > = RB THEN GOTO 662
      0
6610 STP = (N + BS) * XBK
6615  GOTO 5210
6620 C1 = A1:C2 = A2
6625 STP = BUP
6630  GOSUB 5600: REM TSTEP
6635 C1 = B1:C2 = B2
6640 STP = (N - RB) * XBK
6645  GOSUB 5600: REM TSTEP
6650 C1 = A3:C2 = A4
6655 STP = BDN
6660  GOSUB 5700: REM TSTEPSTOP
6665  RETURN : REM

6700  REM TLTX
6705 C1 = B1:C2 = F2
6710 STP = N * XLT * 2 / 3
6715  IF N > (3 * CL) THEN STP =
      (N - CL) * XLT
6720  GOSUB 5700: REM TSTEPSTOP
6725  RETURN : REM

6800  REM TRTX
6805 C1 = F1:C2 = B2
6810 STP = N * XRT * 2 / 3
6815  IF N > (3 * CR) THEN STP =
      (N - CR) * XRT
6820  GOSUB 5700: REM TSTEPSTOP
6825  RETURN : REM
```

```
7000  REM INIT
7005 Z1 = 1024
7010 Z2 = 1024
7015 Z3 = 1024
7020 Z4 = 1024
7025 F1 = 1536
7030 F2 = 1536
7035 B1 = 512
7040 B2 = 512
7045 E1 = 1350
7050 E2 = 1350
7055 E3 = 1250
7060 E4 = 1250
7065 A1 = 700
7070 A2 = 700
7075 A3 = 800
7080 A4 = 800
7085 SFD = 1.15
7090 SBK = 1.15
7095 SLT = .47
7100 SRT = .47
7105 CL = 8
7110 CR = 8
7115 RF = 45
7120 RB = 45
7125 FUP = 30
7130 FDN = 30
7135 BUP = 30
7140 BDN = 30
7145 FS = 4
7150 BS = 4
7155 XFD = 1.15
```

```
7160 XBK = 1.15
7165 XLT = .52
7170 XRT = .52
7175 N = 0:STP = 0
7180 C1 = Z1:C2 = Z2
7185 C3 = Z3:C4 = Z4
7190  RETURN : REM

7200  REM CALIBRATE NEUTRAL  VALU
     ES,  START WITH NEW 7005-
7201  REM "7005 Z1 = A"
7205  GOSUB 7000: REM INIT
7210  PRINT "ENTER VALUE, 0 TO EX
     IT"
7215  INPUT A
7220  IF A = 0 THEN END
7225  GOSUB 6000: REM TRESET
7230  GOTO 7210
7235  REM

7300  REM CALIBRATE LINEAR MOTIO
     N
7305  GOSUB 7000: REM INIT
7310 STP = 200
7315  PRINT "ENTER LEFT DRIVE, 0
     TO EXIT"
7320  INPUT C1
7325  IF C1 = 0 THEN END
7330  PRINT "ENTER RIGHT DRIVE"
7335  INPUT C2
7340  GOSUB 5700: REM TSTEPSTOP
7345  GOTO 7315
```

LISTING OF BINARY ROUTINES

SOURCE FILE: TOPOMAC.SRC

```
0000:             1 ************************************
0000:             2 *                                  *
0000:             3 *  ANDROBOT TOPO DRIVER PROGRAM    *
0000:             4 *        (C) COPYRIGHT BY          *
0000:             5 *       ANDROBOT INC. 1983         *
0000:             6 *                                  *
0000:             7 ************************************
----NEXT OBJECT FILE NAME IS TOPOMAC.OBJ0

0300:             8           ORG $300
FB1E:             9 PDLINPUT  EQU $FB1E      ; GAME CNTRL MONITOR ROUTINE
C0C0:            10 DATREG    EQU $C0D0      ; AMD 9513 DATA REGISTER
C0D1:            11 CMDREG    EQU $C0D1      ; AMD 9513 COMMAND REGISTER
C000:            12 KEYDAT    EQU $C000      ; APPLE KEYBOARD DATA
E003:            13 BASICRET  EQU $E003      ; BASIC RETURN ADDR
03FC:            14 CH1DAT    EQU $3FC       ; CHANNEL 1 DATA LOCATION
03FA:            15 CH2DAT    EQU $3FA       ; CHANNEL 2 DATA LOCATION
03EA:            16 TEMPLOC   EQU $3EA       ; TEMPORARY STORAGE LOCATION
03FE:            17 SYNC      EQU $3FE       ; SYNC DATA LOCATION
03F4:            18 NEUTBASE  EQU $3F4       ; NEUTRAL DATA BASE ADDRESS
0300:20 07 03    19 TSTPSTP   JSR TSTEP      ; TSTEPSTOP ENTRY POINT
0303:20 15 03    20           JSR NEUTRAL    ; STUFF NEUTRAL DATA
0306:60          21           RTS
0307:20 3A 03    22 TSTEP     JSR EDGE?      ; TSTEP ENTRY POINT
030A:20 11 03    23           JSR TSET       ; SET 9513 DATA
030D:20 27 03    24           JSR TCOUNT     ; COUNT LOOP
0310:60          25           RTS
0311:A0 0A       26 TSET      LDY #$A        ; TSET ENTRY POINT
0313:D0 02       27           BNE CRNT       ; LOAD CURRENT VALUES
0315:A0 14       28 NEUTRAL   LDY #$14       ; LOAD NEUTRAL VALUE
0317:20 3A 03    29 CRNT      JSR EDGE?
031A:A2 14       30           LDX #$14       ; LOAD CURRENT VALUES
031C:BD EB 03    31 COPY      LDA TEMPLOC+1,X
031F:8D D0 C0    32           STA DATREG     ; COPY TO AMD 9513
0322:CA          33           DEX
```

```
0323:88          34            DEY
0324:D0 F6       35            BNE COPY       ; DONE?
0326:60          36            RTS
0327:AC EB 03    37 TCOUNT     LDY TEMPLOC+1  ; TCOUNT ENTRY POINT
032A:AE EA 03    38            LDX TEMPLOC    ; X,Y GET COUNT VALUE
032D:20 3A 03    39 LOOP       JSR EDGE?      ; WAIT FOR EDGE
0330:20 49 03    40            JSR KEY?       ; KEYBOARD TOUCH?
0333:CA          41            DEX
0334:D0 F7       42            BNE LOOP       ; X LOOP
0336:88          43            DEY
0337:D0 F4       44            BNE LOOP       ; Y LOOP
0339:60          45            RTS
033A:AD D1 C0    46 EDGE?      LDA CMDREG     ; HISYNC PERIOD
033D:29 02       47            AND #$2
033F:D0 F9       48            BNE EDGE?
0341:AD D1 C0    49 LOSYNC     LDA CMDREG     ; LOSYNC PERIOD
0344:29 02       50            AND #$2
0346:F0 F9       51            BEQ LOSYNC
0348:60          52            RTS            ; NOW AT EDGE OF SYNC
0349:AD 00 C0    53 KEY?       LDA KEYDAT     ; SEE IF ANY KEY
034C:30 01       54            BMI KEYTOUCH   ; IS TOUCHED?
034E:60          55            RTS
034F:20 15 03    56 KEYTOUCH   JSR NEUTRAL    ; STUFF NEUTRAL DATA
0352:4C 03 E0    57            JMP BASICRET   ; RETURN TO BASIC
0355:A2 01       58 JOYSTICK   LDX #$1        ; JOYSTICK ENTRY POINT
0357:8A          59 ONCE       TXA
03581A2 06       60            LDX #$6        ; DELAY 2MS
035A:A0 FF       61 XLOOP      LDY #$FF
035C:88          62 YLOOP      DEY
035D:D0 FD       63            BNE YLOOP
035F:CA          64            DEX
0360:D0 F8       65            BNE XLOOP
0362:AA          66            TAX
0363:20 1E FB    67            JSR PDLINPUT   ; READ PADDLE VALUES (0-255)
0366:98          68            TYA
0367:4A          69            LSR A          ; EACH PADDLE = 0-127
0368:9D EA 03    70            STA TEMPLOC,X
036B:CA          71            DEX
036C:F0 E9       72            BEQ ONCE       ; READ SECOND PADDLE?
```

```
036E:A0 00      73           LDY #$0        ; CLEAR HIGH BYTES
0370:8C FA 03   74           STY CH2DAT
0373:8C FC 03   75           STY CH1DAT
0376:0A         76           ASL A
0377:49 00      77           EOR #$0        ; INVERT PADDLE 0
0379:4A         78           LSR A
037A:18         79           CLC            ;  CALCULATE CH1 = 1-255
037B:69 80      80           ADC #$80
037D:38         81           SEC
037E:ED EB 03   82           SBC TEMPLOC+1
0381:A2 02      83           LDX #$2
0383:C9 40      84 SECOND    CMP #$40       ; TRUNCATE TO 64-192
0385:B0 02      85           BCS LARGE
0387:A9 40      86           LDA #$40
0389:C9 C0      87 LARGE     CMP #$C0
038B:90 02      88           BCC RANGE
038D:A9 C0      89           LDA #$C0
038F:A0 03      90 RANGE     LDY #$3        ; SHIFT LEFT 3
0391:0A         91 SHIFT     ASL A
0392:3E FA 03   92           ROL CH2DAT,X
0395:88         93           DEY
0396:D0 F9      94           BNE SHIFT      ;
0398:9D FB 03   95           STA CH2DAT+1,X ; SAVE CHANNEL
039B:AD EA 03   96           LDA TEMPLOC    ; CALCULATE CHANNEL 2
039E:0A         97           ASL A
039F:49 00      98           EOR #$0        ; INVERT PADDLE 0
03A1:4A         99           LSR A
03A2:18        100           CLC
03A3:6D EB 03  101           ADC TEMPLOC+1
03A6:49 FF     102           EOR #$FF
03A8:CA        103           DEX
03A9:CA        104           DEX
03AA:F0 D7     105           BEQ SECOND     ; DONE?
03AC:AD FC 03  106           LDA CH1DAT     ; CALCULATE HIGH SYNC
03AF:A2 06     107           LDX #$6
03B1:7D F4 03  108 CALCHS    ADC NEUTBASE,X
03B4:CA        109           DEX
03B5:CA        110           DEX
03B6:D0 F9     111           BNE CALCHS
```

```
03B8:69 0A     112           ADC #$A
)3BA:8D FE 03  113           STA SYNC       ; SAVE HIGH SYNC
03BD:AD FD 03  114           LDA CH1DAT+1   ; CALCULATE LO SYNC
03C0:A2 06     115           LDX #$6
03C2:18        116 CALCLS    CLC
03C3:7D F5 03  117           ADC NEUTBASE+1,X
03C6:90 03     118           BCC VTBT
03C8:EE FE 03  119           INC $3FE       ; CARRY INTO HI SYNC
03CB:CA        120 VTBT      DEX
03CC:CA        121           DEX
03CD:D0 F3     122           BNE CALCLS
03CF:8D FF 03  123           STA SYNC+1     ; SAVE LO SYNC
03D2:20 11 03  124           JSR TSET       ; STUFF 9513
03D5:20 49 03  125           JSR KEY?       ; KEYBOARD TOUCH?
03D8:4C 55 03  126           JMP JOYSTICK   ; NO, CONTINUE
03DB:60        127           RTS

*** SUCCESSFUL ASSEMBLY: NO ERRORS

0300 TSTP       0307 TSTEP      0311 TSET       0315 NEUTRAL
0317 CRNT       031C COPY       0327 TCOUNT     032D LOOP
033A EDGE       0341 LOSYNC     0349 KEY        034F KEYTOUCH
0355 JOYSTICK   0357 ONCE       035A XLOOP      035C YLOOP
0383 SECOND     0389 LARGE      038F RANGE      0391 SHIFT
03B1 CALCHS     03C2 CALCLS     03CB VTBT       03EA TEMPLOC
03F4 NEUTBASE   03FA CH2DAT     03FC CH1DAT     03FE SYNC
C000 KEYDAT     C0D0 DATREG     C0D1 CMDREG     E003 BASICRET
FB1E PDLINPUT
```

ASSEMBLY_PROGRAM_MEMORY_MAP

$300 - $3DB : Topo machine language routines

$3EA - $3FF : Topo machine language variables (see below)

$3EA - $3EB : STEPS counter - Lo, Hi: used by TCOUNT

$3EC - $3FS : Neutral values storage: set by HOLD (see below)

$3EC - $3ED : CH4 Neutral - Hi, Lo

$3EF - $3EF : CH3 Neutral - Hi, Lo

$3FO - $3F1 : CH2 Neutral - Hi, Lo

$3F2 - $3F3 : CH1 Neutral - Hi, Lo

$3F4 - $3F5 : SYNC Neutral - Hi, Lo

$3F6 - $3FF : Running values storage: set by HOLD (see below)

$3F6 - $3F7 : CH4 Running - Hi, Lo

$3F8 - $3F9 : CH3 Running - Hi, Lo

$3FA - $3FB : CH2 Running - Hi, Lo

$3FC - $3FD : CH1 Running - Hi, Lo

$3FE - $3FF : SYNC Running - Hi, Lo

INDEX