Heuristics
INC

CONTROLLER 70
FOR THE
APPLE® COMPUTER

1285 Hammerwood Avenue
Sunnyvale, California
(408) 734-8532

Welcome to the Heuristics Family of Users

We hope you will profit from and enjoy using this fine Heuristics Product. Please feel free to send us your comments and suggestions.

Getting Started with Controller 70

If you are **already familiar** with your Apple® computer and associated programs:

- Turn off power and install card in selected peripheral slot (any slot, No. 1 thru 7).
- Make connections to supplied female connector and connect to card. (See Figure 1 and Table 1 for required information.)
- Turn on power, load desired program, and RUN! (See Table 1 for programming control statements for Controller 70.)

If you are **new to the Apple® computer,** please read the entire manual before doing anything.

> **WARNING**
>
> **Do not connect 120 VAC** (or higher voltage) circuits to the Controller 70 relays. Neither they, the connectors, or the card, are rated for operation at these voltages or potential currents. Exposed traces and components will be hazardous to the user if connected to 120 VAC.

INTRODUCTION

The Controller 70 is an inexpensive easy-to-use controller card with which you can use your Apple® computer to control external devices — peripherals. It provides four (4) isolated, dry relay contact closures which may be used to control low-voltage low-current circuits by opening or closing them.

Specific applications for which Model 70 is recommended include:

- Control of low-voltage, low current AC or DC circuits requiring electrical isolation from the Apple® computer.
- Providing low speed on-off signaling to external circuits via contact closure (circuit completion) or contact openings (circuit interruption). Pulse rates to 30 pulse/second can be obtained.

Applications for which the Model 70 is suitable include:

- Control of model railroads and ships
- Low speed computer signaling on telephone 0-75 baud and Type 1000 circuits (so called "burglar-alarm" or DC signaling circuits)
- Control of auto-dialer
- Control of battery operated devices
- Additional control of other peripherals attached to the Apple® where circuit isolation providing relays is important.

Apple® is a Registered Trademark of the Apple Computer Co.

The Model 70 is easily programmed in BASIC. A POKE (X), N command is given to select the relay action desired.

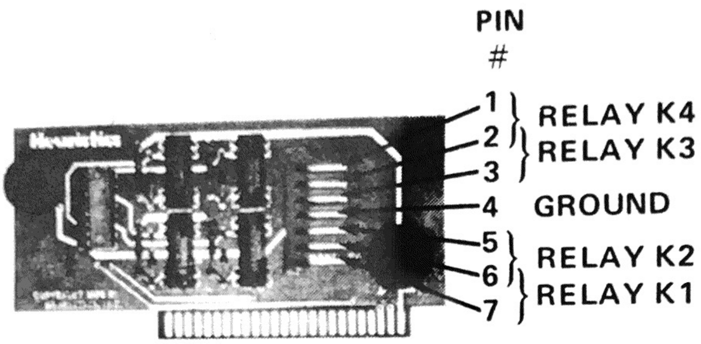


FIGURE 1
MODEL 20A CONTROLLER

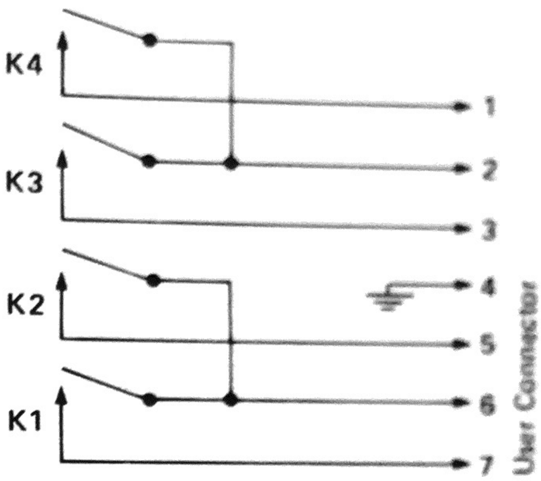


FIGURE 2
SCHEMATIC DIAGRAM OF USER FUNCTIONS

Heuristics™ voice input peripherals for the Apple® computer can be used to specify the action controlled by the Model 70. Thus direct voice control over external devices may be obtained, completely programmed in BASIC. Commands are given in functional form such as "ON" or "OFF" or "NUMBER," "6", and translated into appropriate contact closures in the BASIC program.

INSTALLING THE CONTROLLER 70

- Turn the Apple® Computer <u>OFF!</u>
  (Note: Power should always be OFF when inserting or removing a card. If the power is on, removal or insertion of a card can cause permanent damage to both the card and the computer!)
- Connect the 7-pin flat female cable connector which you have wired to its male counterpart on the Controller 70 board.
- Remove the Apple® top cover and look at the row of eight connectors at the rear. Each connector is assigned a number (0 to 7) from left to right as viewed from the front. The number is located at the back edge of the connector. These numbered connectors are called **slots**.
- Plug the Controller 70 into any slot except Slot # Ø; the component side of controller will be to your right as you insert the card from the front of the Apple®.
- Direct the cable through a convenient cable opening in back of the Apple®.
- Replace the cover.

PROGRAMMING THE CONTROLLER 70

A particular relay contact is closed by POKEing a "1" into its corresponding bit position in a location in memory associated with the controller slot. The same relay contact is opened by POKEing a "Ø" into the same bit position.

This location for POKEing is specified by the number X, where X = −16256 + SLOT * 16. This location is in the group of memory locations specifically reserved for use of the peripheral card installed in the position SLOT (1 to 7).

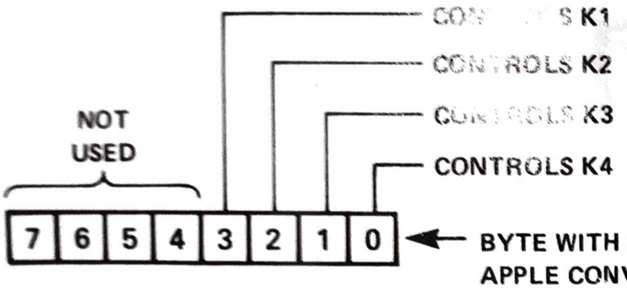
Truth Table

	Relay (O=Open; C=Closed)			
POKE (X),N (Note 1)	K1	K2	K3	K4
N=Ø	O	O	O	O
N=1	O	O	O	C
N=2	O	O	C	O
N=3	O	O	C	C
N=4	O	C	O	O
N=5	O	C	O	C
N=6	O	C	C	O
N=7	O	C	C	C
N=8	C	O	O	O
N=9	C	O	O	C
N=10	C	O	C	O
N=11	C	O	C	C
N=12	C	C	O	O
N=13	C	C	O	C
N=14	C	C	C	O
N=15	C	C	C	C

Note 1: X= −16256 + S*16; S = 1 to 7, number of slot of Controller 70.

TABLE 1
RELAY OPERATION

(Note that **all relays** are re-established open or closed each time a new POKE command is given. Thus to leave a given relay in a particular control position it is important to leave the bit associated with it **unchanged** in the control byte that is POKE'd. As an example, suppose bit Ø is set by giving a POKE(X), 1 closing relay K4, and it is now desired to close Relay K3, leaving K4 closed. In order to do this, a POKE(X), 3 must be given which puts 1's in both bit positions Ø and 1 in the Byte (where X is the location specified by the formula X = −16256 + SLOT * 16). See Figure 3 below.



BYTE WITH BITS NUMBERED ACCORDING TO APPLE CONVENTION. BIT Ø IS LEAST SIGNIFICANT BIT. BYTE IS IN LOCATION X, WHERE X = 16256 + SLOT * 16.

FIGURE 3
USE OF BITS IN CONTROL BYTE

Programming Hint

It may be useful to think in terms of a control byte and of modifying that as required, as follows.

First, initialize the control byte by setting BC = Ø, where BC is the name of the control byte.

Then:

```
To Set Control Bit Ø          (K4)
        BC = BC + 1
        POKE X, BC
To Set Control Bit 1          (K3)
        BC = BC + 2
        POKE X, BC
To Set Control Bit 2          (K2)
        BC = BC + 4
        POKE X, BC
To Set Control Bit 3          (K1)
        BC = BC + 8
        POKE X, BC
```

```
NOW:  To Clear Control Bit Ø          (K4)
                BC = BC -1
                POKE X, BC
      To Clear Control Bit 1          (K3)
                BC = BC -2
                POKE X, BC
      To Clear Control Bit 2          (K2)
                BC = BC -4
                POKE X, BC
      To Clear Control Bit 3          (K1)
                BC = BC -8
                POKE X, BC
      To RESET (open) all relays, simply
                POKE X, Ø
```

Naturally more than one modification operation as given above may be combined before POKE-ing the modified control byte back into location X.

For example

```
BC = BC +1      (Will Close Relay K4)
BC = BC -2      (Will Open Relay K3)
POKE X, BC
```

will close Relay K4 (Bit Ø = 1) and open Relay K3 (Bit 1 = Ø).

Programmers Caution:

When following the above examples make sure that a relay to be turned off is **already** on, or that one to be turned on is **already** off. The reason follows:

The above rules are a simple arithmetic approach to setting and re-setting individual bits from INTEGER BASIC, since bit-level AND, OR, NOT and XOR operations are not part of BASIC.

As a result, if a bit is **already set** and then an addition operation is done to set it, there will be a carry to the next higher bit position causing the higher order relays to be turned on or off. Similarly if a bit is **already clear** and a subtraction is done to clear a bit there will be borrowing from the next higher bit position which will affect the higher order relays.

Timing of Relay Pulse Duration

To produce a pulse of a specified duration, W, the relay should be turned on and then off by POKEing first a "1" and then, after a suitable delay, D, a "Ø".

This program will do that:

```
100 REM SLOT IS MODEL 70 ADDRESS
110 X = −16256 + SLOT * 16
120 POKE X, 1
130 REM DELAY
140 FOR K = 1 TO D: NEXT K
150 POKE X, Ø
```

To get a pulse of approximately width W (See Figure 4), set D as follows:

Width(W)	Delay(D)	Remarks
3 ms	1	
4 ms	2	
7 ms	5	
9 ms	8	TTY Rate
12.5 ms	10	
16.0 ms	15	
22.0 ms	20	

and in general W(in ms) = 2.5 + 0.9 D

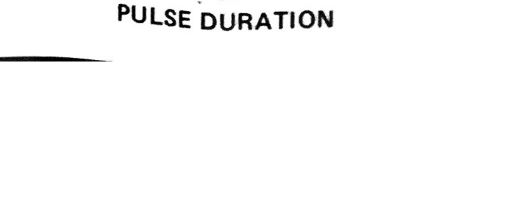


FIGURE 4
PULSE DURATION

USE WI

The Co
verbal co

Progra
for DOS I
periphera
paramete
training a

To train
then print
INTEGER
string. "ST

After Spe
"RESET". "
program wi

If W$ = "S
ITEM", rela

In this wa

For exam

USE WITH SPEECHLAB

The Controller 70 may be used with Speechlab Model 20A or the Speechlink H2000 to allow verbal commands to control the relay closing.

Programming Speechlab is simple from INTEGER BASIC . . . and not much more complicated for DOS INTEGER BASIC (or DOS APPLESOFT for the H2000). Following the standard Apple II peripheral conventions, Speechlab is selected either for input (recognition) or output (passing parameters to Speechlab) via the IN# and PR# commands. Speechlab is used in two modes: training and recognition.

To train Speechlab on a word like "STOP", simply select the board with a PR# command and then print the string, "STOP", to it. For example with Speechlab in SLOT #3, the following INTEGER BASIC program will cause Speechlab to listen for a word and associate it with the string, "STOP":

```
200 PR#3
210 PRINT "STOP'
```

After Speechlab has been trained on the desired words and phrases, for instance, "STOP", "RESET", "NEXT ITEM", it is ready to recognize spoken words. The following INTEGER BASIC program will recognize a spoken word and return the string assigned during training.

```
300 IN#3
310 INPUT W$
```

If W$ = "STOP" for example, then relay K1 might be opened as explained above. If W$ = "NEXT ITEM", relay K1 might be set (or pulsed—turned on for 1 second and then off again).

In this way you can work through the CONTROLLER 70 using voice commands.

For example (in INTEGER BASIC, for the Model 20A)

```
 70 REM '20A' in SLOT 3, '70' in 4
 80 X = (-16256 + 4 * 16)
 85 BC = Ø:REM INITIALIZE CONTROL
 90 DIM W$(20)
100 REM TRAINING
110 W$ = "STOP"
120 PRINT "SAY", W$
130 PR#3
140 PRINT W$
150 PR#Ø
```

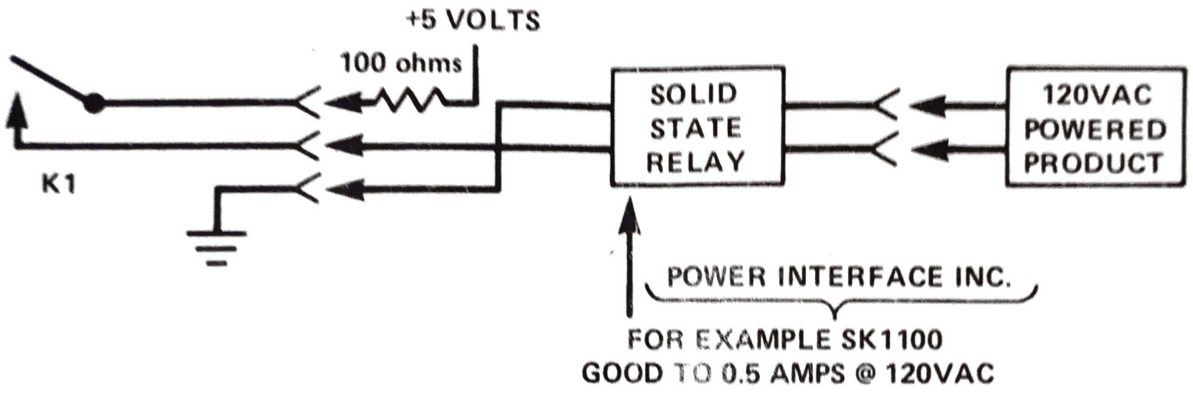
٩

١٨-١١

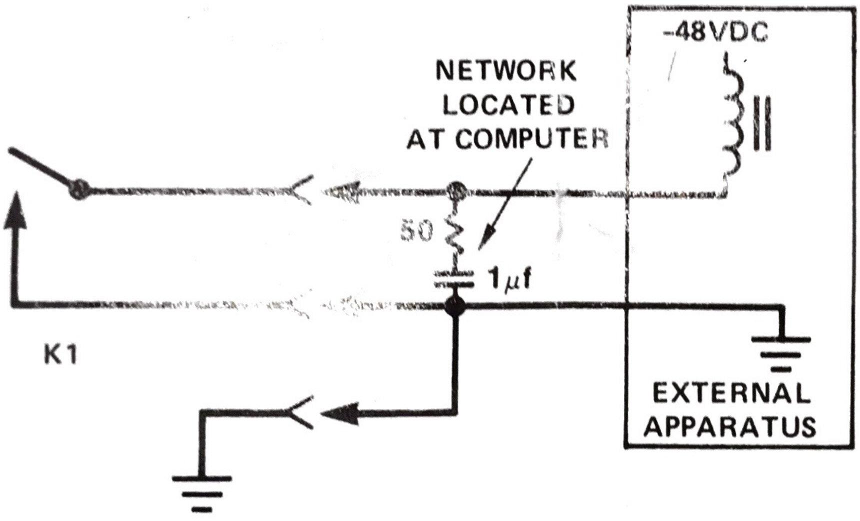
```
160  W$ = "NEXT ITEM"
170  PRINT "SAY", W$
180  PR#3
190  PRINT W$
200  PR#Ø
210  REM RECOGNIZE
220  PRINT "SPEAK . . . ."
230  IN#3
240  INPUT W$
250  IN#Ø:PR#Ø
260  IF W$ = "STOP" Then 300
270  IF W$ = "NEXT ITEM" Then 400
280  GO TO 210
300  REM STOP
310  IF BC = Ø THEN 330
320  BC = BC −1
330  POKE (X), BC
340  GO TO 210
400  REM NEXT ITEM
410  IF BC = 1 THEN 430
420  BC = BC + 1
430  POKE (X) BC
440  GO TO 210
```

Some Applications

- For control of 120VAC powered devices, the following approach may be used:



- For control of external AC or DC powered relays for example (up to 48 VDC maximum)



Note addition of arc suppression RC network.

SPECIFICATIONS

Contact Rating (Resistive Load)

0.5 Amp max.

100 VDC max.

10 Watts max.

Initial Resistance:

200 milliohms

Bounce Time (max):

1.0ms on "operate" and "release"

Life:

20 Million Operations at 20 VDC, 0.25A, resistive.

Repetition Rate:

30 cycles per second, max.

LIMITED WARRANTY

Products manufactured by and purchased from HEURISTICS, INC., or its authorized dealers, are guaranteed to meet specifications in effect at the time of manufacture for a period of at least one year following purchase. These units are additionally guaranteed against defects in materials or workmanship for the same one year period. All in-warranty factory assembled units returned to HEURISTICS, INC., postpaid with proof of purchase date will be repaired and returned without charge. Out-of-warranty units, or units returned without proof-of-purchase, will be repaired and returned COD at prevailing repair charges then in effect.

This warranty is made in lieu of all other warranties expressed or implied and is limited in any case to the repair or replacement of the module involved.