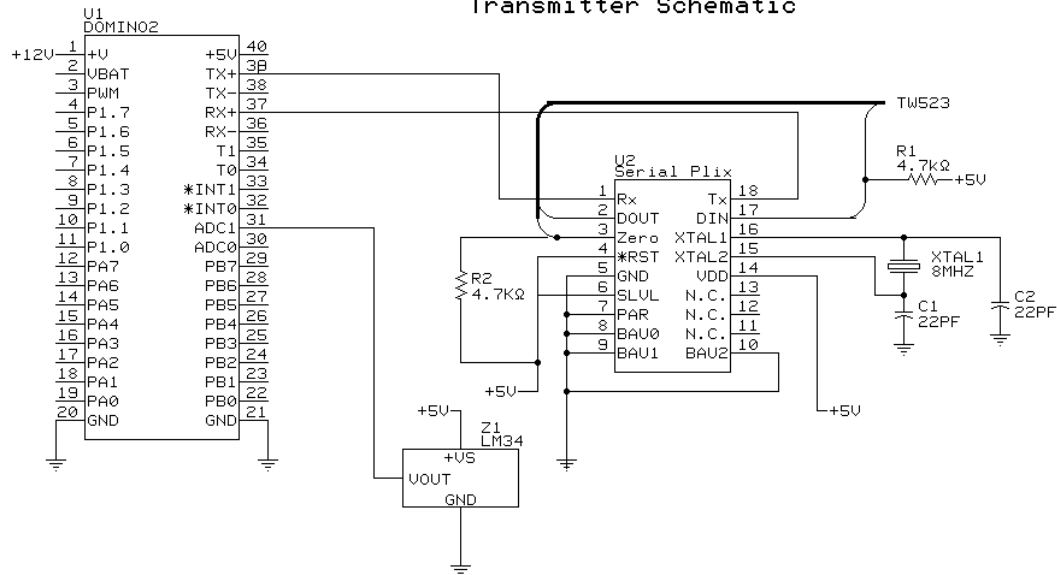
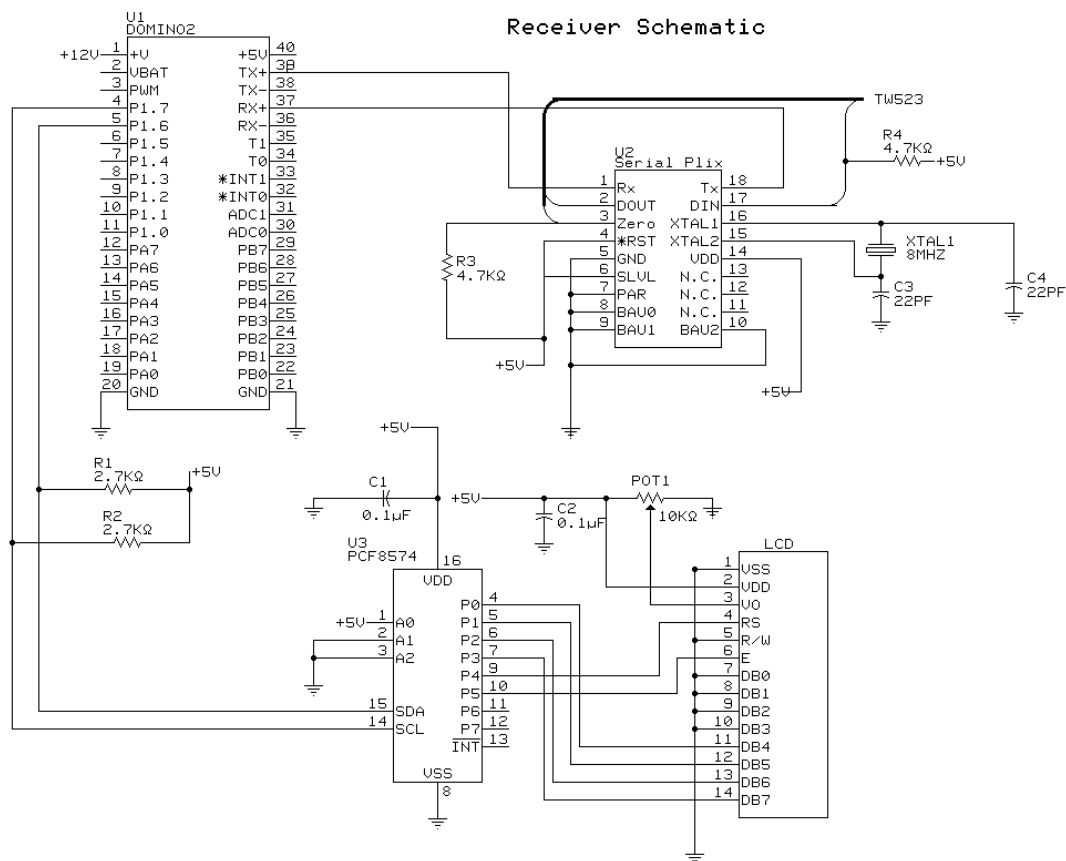
	Application Note
	Serial Plix
Temperature Transmission Via Serial Plix	Date: 08/14/00
<p>Introduction: The following procedure shows how to use the Serial Plix in conjunction with the Domino 2 microcontroller and the LM34 temperature sensor. This application will monitor the temperature from a remote location and transmit the temperature data via the AC power lines.</p>	
<p>Background: The Serial Plix has been used as a simple interface to X-10 control modules. In more complex systems, the Serial Plix can be used to transmit information through the AC power lines. The microcontroller chosen for this application was the Domino 2. It is used to encode the information into plain ASCII text that the Serial Plix will understand. The data is sent across the AC power lines and received at a remote location using a Serial Plix chip and a second Domino 2 controller.</p>	
<p>How it works: This application is broken down into two parts, a transmitter and a receiver. Both portions consist of a Domino 2 and a Serial Plix. The transmitter also has an LM34 Temperature sensor. The receiver uses a PCF8574, an I2C I/O expander chip, to interface the Domino 2 with a parallel LCD display. A Domino 2A is used for the transmitter controller because it has a 12-bit ADC, which is needed to read the output of the LM34. The output is sampled 10 times by the ADC and then averaged for smoothing. The result is converted into the temperature. The temperature range of the LM34 is between 0°F and 300°F. If the temperature is not within these limits the ADC will resample the LM34. The temperature will be converted into a series of ASCII commands statements that the Serial Plix will be able to understand. The Serial Plix has house codes from A-P and unit codes 1-16 which can be used to transmit X-10 commands. For this program we decided to reserve house codes L-P and their unit codes 1-10 for use in encoding the temperature information for transmission. The Domino 2A encodes the four digit temperature (###.#) and sends those commands to the Serial Plix. The commands are then sent to the TW523 module and transmitted across the AC power lines. The remote Serial Plix receives the commands from its TW523 and sends the data to a Domino 2 which stores the commands as strings in memory. The Domino 2 will compare the strings to verify that the appropriate house codes were received in the correct order. The receiving Domino 2 is responsible for converting the information into the appropriate place values to represent the temperature. When the temperature has been reconstructed, the receiving Domino 2 will send the temperature to the LCD, using the PCF8574. After the temperature has been displayed, a completion string is transmitted back to the transmitter to signify the end of the temperature transmission. Once the completion command is received, the transmitter will begin sampling the LM34 for any temperature changes.</p>	

Transmitter Schematic



Receiver Schematic



Program Listing:

```

10 REM *** Program 1 of 2
11 REM *** This is the temperature sensor/transmitter program which uses
12 REM *** a Domino2A, an LM34 temperature sensor and a Serial Plix. The

```

```

13 REM *** temperature in a certain area is monitored by the LM34 and
14 REM *** transmitted through a standard wall outlet via the Serial Plix.
15 REM *** The temp. is sampled ten times and averaged for smoothing.
16 REM *** The temperature can range from 0 to 300 degrees Fahrenheit,
17 REM *** accurate to one tenth of a degree.
20 MTOP = 3000H
30 REM *** Reserves memory for string use.
40 STRING 20,4 : $(1)= "$OFF" : $(2)= "01"
50 REM *** Sample loop for the ADC. Takes 10 samples & saves in CH1.
55 CLOCK 1
65 CH1 = 0
70 FOR SMP = 0 TO 9
80 CALL 0F008H
90 POP KEV
110 CH1 =CH1 +KEV
120 NEXT SMP
130 REM *** Takes the average of the ten samples
140 CH1=CH1/10
150 REM *** Converts the ADC count to a voltage, using a 5.1 ref voltage.
160 CH1= (CH1*(5.1/4096))
170 REM *** Every 10mV on the output of the LM34 represents 1 degree F.
180 TEMP=CH1/.01
190 REM *** The max. range of the LM34 is 300°F. If the temp is larger than
200 REM *** 300 degrees then there has been an error. New samples will be
210 REM *** Taken.
220 IF TEMP>300 THEN 70
230 REM *** Sends the start command to the receiver.
240 PRINT "$OFFL0501"
250 REM *** Allows a 1 sec delay so serial plix can complete transmission
255 DELAY = TIME + 1
260 IF TIME<DELAY THEN 260
270 REM *** Jumps to appropriate subroutine for specific temp ranges.
280 IF (TEMP>=100.AND.TEMP<=300) THEN GOSUB 560
290 IF (TEMP>=10.AND.TEMP<100) THEN GOSUB 1330
300 IF (TEMP>=1.AND.TEMP<10) THEN GOSUB 1800
310 IF TEMP<1 THEN GOSUB 2150
320 REM *** Sends completion command to receiver.
330 PRINT "$OFFL0901"
340 REM *** ALLOWS A 2 SEC DELAY
345 DELAY = TIME + 2
350 IF TIME<DELAY THEN 350
360 REM *** Serial Plix polls the power outlet for confirmation from the
370 REM *** receiver before the sampling can continue.
380 PRINT "$DINA9999"
390 REM *** Saves confirmation in string zero so a comparison can be done.
400 INPUT $(0)
410 W=XBY(2FFAH)
420 X=XBY(2FFBH)
430 Y=XBY(2FFCH)
440 Z=XBY(2FFDH)
450 REM *** The Domino must see a house code of "L" and a unit code of "1"
460 REM *** followed by a carriage return, to restart the sampling.
470 IF (W<>76.OR.Y<>49.OR.Z<>13) THEN 380
475 DELAY = TIME+4
480 IF TIME<DELAY THEN 480 ELSE 70
490 REM *** If the temperature has a value greater than 100 degrees, this
500 REM *** subroutine is used to transmit appropriate house and unit codes
510 REM *** used to represent specific digits and thier place values.
520 REM *** The variable "A" represents the hundreds place.
530 REM *** The variable "B" represents the tens place.
540 REM *** The variable "C" represents the ones place.
550 REM *** The variable "D" represents the tenths place.
560 A=TEMP/100

```

```

570 REM *** INT eliminates all numbers to the right of the decimal point so
580 REM *** only a single digit number remains.
590 A=INT(A)
600 REM *** Prints to the Serial Plix a house code of "M" followed by unit
610 REM *** code that corresponds to the value of "A". Because a zero can't
620 REM *** be transmitted by the Serial Plix number ten is used.
630 HC=77 : IF A=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+A
640 GOSUB 9000
730 REM *** Delay of 1 sec. to allow Serial Plix time to transmit.
735 DELAY = TIME + 1
740 IF TIME<DELAY THEN 740
750 REM *** The following mathematics sets "B" as a single digit for the
760 REM *** purpose of transmission.
770 B=TEMP-(A*100)
780 B=B/10
790 B=INT(B)
800 REM *** Prints a house code of "N" and a unit code that corresponds to
810 REM *** value of "B".
820 HC=78 : IF B=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+B
830 GOSUB 9000
920 IF TIME<(DELAY+1) THEN 920
930 REM *** Calculates "C" to a single digit.
940 C=TEMP-(A*100)
950 C=C-(B*10)
960 C=INT(C)
970 REM *** Prints a house code of "O" followed by a unit code that
980 REM *** corresponds to the value of "C".
990 HC=79 : IF C=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+C
1000 GOSUB 9000
1090 IF TIME<(DELAY+2) THEN 1090
1100 REM *** Calculates variable "D" to a single digit interger.
1110 D=TEMP-(A*100)
1120 D=D-(B*10)
1130 D=(D-C)*10
1140 D=INT(D)
1150 REM *** Sends a house code of "P" and a unit code corresponding to the
1160 REM *** the variable "D" to the Serial Plix.
1170 HC=80 : IF D=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+D
1260 GOSUB 9000
1270 IF TIME<(DELAY+3) THEN 1270
1280 RETURN
1290 REM *** If the temperature is between 10 and 100 degrees this
1300 REM *** subroutine is used to configure command sequence. Because the
1310 REM *** digit in the hundreds place will not be needed a zero be sent
1320 REM *** automatically, followed by a one second delay.
1330 PRINT "$OFFM1001"
1335 DELAY = TIME + 1
1340 IF TIME<DELAY THEN 1340
1350 B=TEMP/10
1360 B=INT(B)
1370 HC=78 : IF B=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+B
1380 GOSUB 9000
1470 IF TIME<(DELAY+1) THEN 1470
1480 C=TEMP-(B*10)
1490 C=INT(C)
1500 HC=79 : IF C=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+C
1510 GOSUB 9000
1600 IF TIME<(DELAY+2) THEN 1600
1610 D=TEMP-(B*10)
1620 D=(D-C)*10
1630 D=INT(D)
1640 HC=80 : IF D=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+D
1650 GOSUB 9000

```

```

1740 IF TIME<(DELAY+3) THEN 1740
1750 RETURN
1760 REM *** If the temperature is between 1 and 10 degrees then this is
1770 REM *** is the subroutine that will be used. Both the "A" & "B"
1780 REM *** variables will be zeros so the corresponding command sequences
1790 REM *** will be sent to the Serial Plix followed by a one second delay.
1800 PRINT "$OFFM1001"
1805 DELAY = TIME + 1
1810 IF TIME<DELAY THEN 1810
1820 PRINT "$OFFN1001"
1830 IF TIME<(DELAY+1) THEN 1830
1840 C=TEMP
1850 C=INT(C)
1860 HC=79 : IF C=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+C
1870 GOSUB 9000
1960 IF TIME<(DELAY+2) THEN 1960
1970 D=(TEMP-C)*10
1980 D=INT(D)
1990 HC=80 : IF D=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+D
2000 GOSUB 9000
2090 IF TIME<(DELAY+3) THEN 2090
2100 RETURN
2110 REM *** If the temperature is smaller than 1 then this subroutine will
2120 REM *** be used. Variables "A", "B" and "C" will be zeros, so the
2130 REM *** corresponding command sequences will be sent to the Serial
2140 REM *** Plix, each followed by a one second delay.
2150 PRINT "$OFFM1001"
2155 DELAY = TIME + 1
2160 IF TIME<DELAY THEN 2160
2170 PRINT "$OFFN1001"
2180 IF TIME<(DELAY+1) THEN 2180
2190 PRINT "$OFFO1001"
2200 IF TIME<(DELAY+2) THEN 2200
2210 D=TEMP*10
2220 D=INT(D)
2230 HC=80 : IF D=0 THEN UC1=49 : UC2=48 : ELSE UC1=48 : UC2=UC1+D
2240 GOSUB 9000
2330 IF TIME<(DELAY+3) THEN 2330
2340 RETURN
9000 PRINT $(1),CHR(HC),CHR(UC1),CHR(UC2),$(2)
9010 RETURN
9020 END

```

```

24 REM *** PROGRAM 2 OF 2.
26 REM *** THIS IS THE RECEIVER'S PROGRAM WHICH WILL USE A SERIAL PLIX TO
28 REM *** RETRIEVE THE TEMPERATURE FROM THE POWER OUTLET AND PRINT IT TO
30 REM *** A PARALLEL LCD SCREEN USING AN I2C I/O EXPANDER CHIP.
35 MTOP=3000H
40 REM *** RESERVES MEMORY FOR STRING USE
60 STRING 6,4
182 REM *** SERIAL PLIX USES THE 'DINA' COMMAND TO POLL THE OUTLET FOR INFO
200 PRINT "$DINA0550"
220 INPUT $(0)
222 REM *** THE INFO RECEIVED BY THE SERIAL PLIX WILL BE STORED INTO
224 REM *** THESE MEMORY LOCATIONS AND INTO THE APPROPRIATE VARIABLES
240 A=XBY(2FFAH) : B=XBY(2FFBH) : C=XBY(2FFCH) : D=XBY(2FFDH)
302 REM *** THE DOMINO2 COMPARES THE INFO. RECEIVED, THE DOM2 NEEDS TO
304 REM *** SEE A HOUSE CODE OF 'L' AND A UNIT CODE OF '5', IF THIS IS NOT
306 REM *** RECEIVED THEN THE DOM2 LOOPS BACK TO THE DINA COMMAND. IF L5 IS
308 REM *** RECEIVED THEN IS STORES IT IN VARIABLES B AND C.
320 IF (B<>76.OR.C<>53) THEN 200

```

```

322 REM *** THE SERIAL PLIX WILL POLL THE POWER OUTLET FOR MORE INFORMATION
340 PRINT "$DINA0550"
342 REM *** ONCE SOME INFORMATION IS RECEIVED THEN IT WILL BE STORED IN
344 REM *** MEMORY FOR COMPARASION.
360 INPUT $(0)
380 E=XBY(2FFAH): F=XBY(2FFBH): G=XBY(2FFCH): H=XBY(2FFDH)
442 REM *** THE DOM2 COMPARES THE INFORMATION. THE DOM2 NEEDS TO SEE A
444 REM *** HOUSE CODE OF 'M' AND A UNIT CODE LESS THEN OR EQUAL TO '10'
445 REM *** IF VALUES ARE NOT RECEIVED THEN THE DOM2 WILL LOOP BACK TO THE
446 REM *** SECOND DINA COMMAND.
460 IF (F<>77.OR.G>57) THEN 340
462 REM *** THE FOLLOWING COMMAND WILL VARIABLES G AND H. IF THE UNIT CODE
464 REM *** RECEIVED IS TEN OR MORE THE TWO DIGITS WILL BE STORED IN THESE
466 REM *** VARIABLES. IF THE VALUE IS GREATER THEN 10 THEN THE PROGRAM
468 REM *** LOOPS BACK TO THE SECOND DINA COMMAND. IF THE VALUE EQUALS 10
470 REM *** THEN THE VARIABLE G IS SET EQUAL TO ZERO.
480 IF (G=50.AND.H=48) THEN 340
500 IF H=48 THEN G=48
502 REM *** THE SERIAL PLIX POLLS THE POWER OUTLET FOR INFORMATION.
520 PRINT "$DINA0550"
540 INPUT $(0)
560 I=XBY(2FFAH) : J=XBY(2FFBH) : K=XBY(2FFCH) : L=XBY(2FFDH)
622 REM *** THE INFORMATION GATHERED IS COMPARED, THE DOM2 NEEDS TO SEE
624 REM *** A HOUSE CODE OF 'N', AND A UNIT CODE DEFINED IN THE ABOVE REM.
640 IF (J<>78.OR.K>57) THEN 520
660 IF (K=50.AND.L=48) THEN 520
680 IF L=48 THEN K=48
682 REM *** THE SERIAL PLIX POLLS THE OUTLET FOR INFORMATION.
700 PRINT "$DINA0550"
720 INPUT $(0)
740 M=XBY(2FFAH) : N=XBY(2FFBH) : O=XBY(2FFCH) : P=XBY(2FFDH)
802 REM *** WHEN THE INFORMATION IS COMPARED THE DOM2 NEEDS TO SEE A HOUSE
804 REM *** CODE OF 'O' AND A UNIT CODE AS DEFINED IN ABOVE REM.
820 IF (N<>79.OR.O>57) THEN 700
840 IF (O=50.AND.P=48) THEN 700
860 IF P=48 THEN O=48
862 REM *** THE SERIAL PLIX POLLS THE OUTLET FOR INFORMATION.
880 PRINT "$DINA0550"
900 INPUT $(0)
920 Q=XBY(2FFAH) : R=XBY(2FFBH) : S=XBY(2FFCH) : T=XBY(2FFDH)
982 REM *** WHEN THE INFORMATION IS COMPARED THE DOM2 NEEDS TO SEE A HOUSE
984 REM *** CODE OF 'P' AND A UNIT CODE AS DEFINED IN ABOVE REM.
1000 IF (R<>80.OR.S>57) THEN 880
1020 IF (S=50.AND.T=48) THEN 880
1040 IF T=48 THEN S=48
1042 REM *** THE SERIAL PLIX POLLS THE OUTLET FOR THE FINAL COMMAND SEQUENCE
1060 PRINT "$DINA0550"
1080 INPUT $(0)
1100 U=XBY(2FFAH) : V=XBY(2FFBH) : W=XBY(2FFCH) : X=XBY(2FFDH)
1162 REM *** WHEN THE INFORMATION IS COMPARED THE DOM2 NEEDS TO SEE A HOUSE
1164 REM *** OF 'L' AND A UNIT CODE OF '9' WHICH WILL INDICATE THAT THE
1166 REM *** TRANSMITTER HAS FINISHED SENDING THE TEMPERATURE. IF THIS
1168 REM *** IS NOT RECEIVED THEN THE TEMPERATURE WILL NEVER BE DISPLAYED.
1180 IF (V<>76.OR.W<>57) THEN 1060
1182 REM *** VALUES STORED IN DESIGNATED VARIABLES ARE NOW CONVERTED FROM
1184 REM *** ASCII TO DECIMAL VALUES.
1200 G=G-48 : K=K-48 : O=O-48 : S=S-48
1262 REM *** THE NUMBERS MUST NOW BE CONVERTED INTO THIER APPROPRIATE
1264 REM *** POSITIONS IN THE TEMPERATURE. 'G' REPRESENTS THE HUNDREDS PLACE
1266 REM *** 'K' REPRESENTS THE TENS PLACE, 'O' REPRESENTS THE ONES PLACE,
1270 REM *** AND 'S' REPRESENTS THE TENTHS PLACE IN THE TEMPERATURE.(GKO.S)
1280 G=G*100 : K=K*10 : S=S*0.1
1322 REM *** WHEN ALL VARIABLES ARE ADDED TOGETHER THE SUM IS STORED IN THE

```

```

1324 REM *** VARIABLE Z. Z IS EQUAL TO THE TEMPERATURE.
1340 Z=G+K+O+S
1342 REM *** CALLS DOM2 UTILITIES TO INITIALIZE THE I2C LCD
1360 CALL 0F030H
1362 REM *** CLEARS THE LCD DISPLAY
1380 CALL 0F038H
1382 REM *** ALLOWS USER TO USE U01/U00 COMMAND
1400 CALL 0F050H
1402 REM *** ALLOWS USER TO USE A SECONDARY CONSOLE OUTPUT
1420 UO 1
1430 REM *** PRINTS TO THE LCD 'THE TEMPERATURE IS ###.#' WHERE ###.# IS THE
1435 REM *** ACTUAL TEMPERATURE.
1440 PRINT " THE TEMPERATURE IS"
1460 PRINT "      ",
1480 PRINT USING(###.#),Z,
1500 PRINT "      "
1502 REM *** RETURNS CONSOLE CONTROL BACK TO BASIC-52
1520 UO 0
1522 REM *** STARTS THE INTERNAL CLOCK ON THE DOM2 AND SETS IT TO ZERO
1540 TIME=0 : CLOCK 1
1542 REM *** PROVIDES A 10Sec DELAY BEFORE TRANSMITTING THE CONFIRMATION
1544 REM *** COMMAND BACK TO THE TEMPERATURE SENSOR.
1560 IF TIME<10 THEN 1560
1590 REM *** TRANSMITS A FINISH CODE BACK TO THE TRANSMITTER SO THAT IT CAN
1600 REM *** START THE SAMPLING PROCESS OVER AGAIN.
1610 PRINT "$OFFL0101"
1620 REM *** TIME DELAY SO THE SERIAL PLIX CAN FINISH TRANSMITTING THE
1630 REM *** COMMAND
1640 IF TIME<13 THEN 1640
1650 REM *** STOP THE CLOCK
1660 CLOCK 0 : TIME=0
1670 IF TIME=0 THEN 182
1680 END

```