Includes barcode for easy software entry.

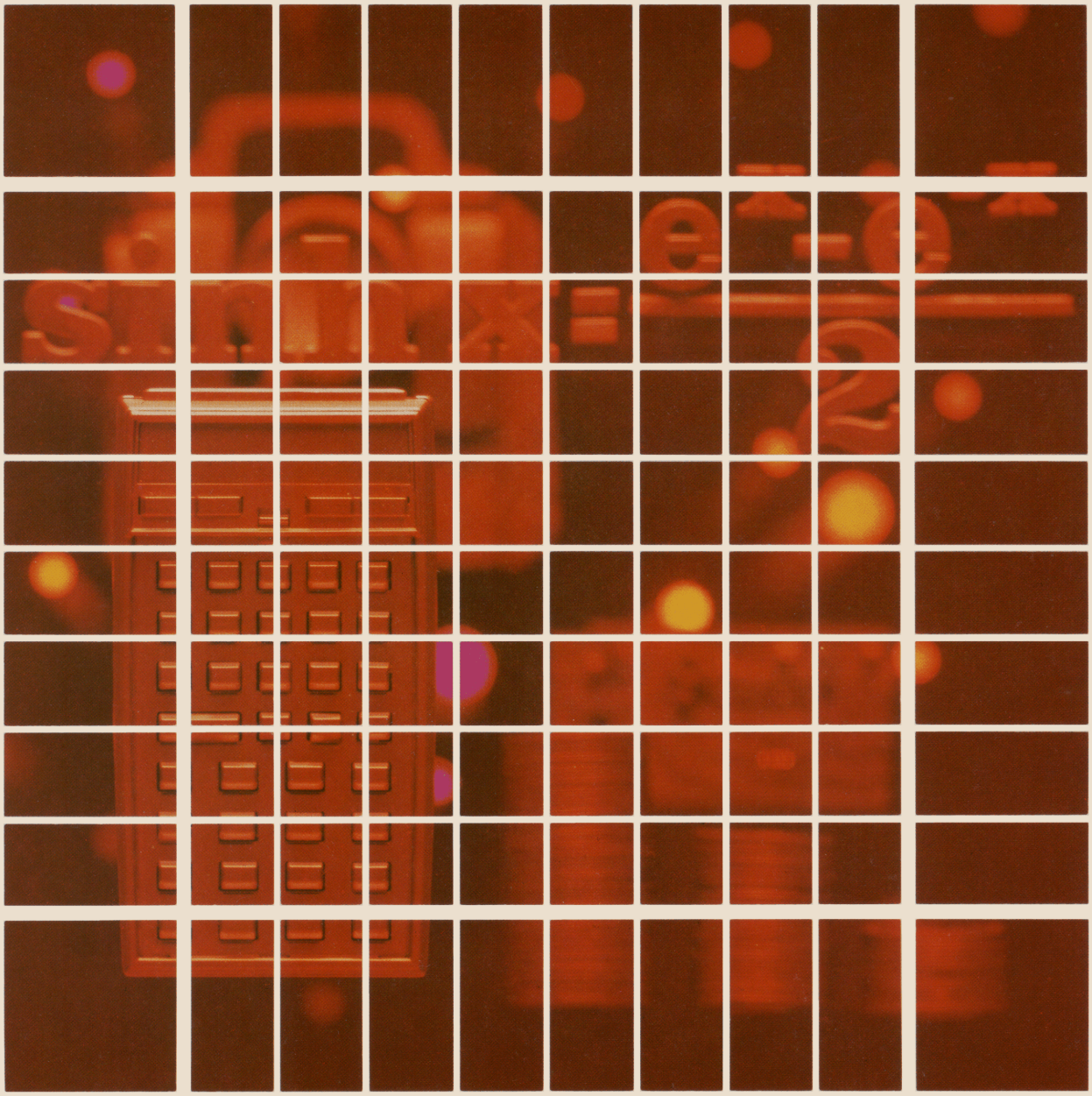
HEWLETT-PACKARD

HP-41

USERS' LIBRARY

Time Module Solutions I

For the HP 82182A Time Module



NOTICE

The program material contained herein is supplied without representation or warranty of any kind. Hewlett-Packard Company therefore assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.

INTRODUCTION

This HP-41C Solutions book was written to help you get the most from your calculator. The programs were chosen to provide useful calculations for many of the common problems encountered.

They will provide you with immediate capabilities in your everyday calculations and you will find them useful as guides to programming techniques for writing your own customized software. The comments on each program listing describe the approach used to reach the solution and help you follow the programmer's logic as you become and expert on your HP calculator.

KEYING A PROGRAM INTO THE HP-41C

There are several things that you should keep in mind while you are keying in programs from the program listings provided in this book. The output from the HP 82143A printer provides a convenient way of listing and an easily understood method of keying in programs without showing every keystroke. This type of output is what appears in this handbook. Once you understand the procedure for keying programs in from the printed listings, you will find this method simple and fast. Here is the procedure:

1. At the end of each program listing is a listing of status information required to properly execute that program. Included is the SIZE allocation required. Before you begin keying in the program, press [XEQ] [ALPHA] SIZE [ALPHA] and specify the allocation (three digits; e.g., 10 should be specified as 010).

   Also included in the status information is the display format and status of flags important to the program. To ensure proper execution, check to see that the display status of the HP-41C is set as specified and check to see that all applicable flags are set or clear as specified.

2. Set the HP-41C to PRGM mode (press the [PRGM] key) and press [■] [GTO] [•] [•] to prepare the calculator for the new program.

3. Begin keying in the program. Following is a list of hints that will help you when you key in your programs from the program listings in this handbook.

   a. When you see " (quote marks) around a character or group of characters in the program listing, those characters are ALPHA. To key them in, simply press [ALPHA], key in the characters, then press [ALPHA] again. So "SAMPLE" would be keyed in as [ALPHA] "SAMPLE" [ALPHA].

   b. The diamond in front of each LBL instruction is only a visual aid to help you locate labels in the program listings. When you key in a program, ignore the diamond.

   c. The printer indication of divide sign is /. When you see / in the program listing, press [÷].

   d. The printer indication of the multiply sign is ⋇. When you see ⋇ in the program listing, press [×].

   e. The ⊢ character in the program listing is an indication of the [APPEND] function. When you see ⊢, press [■] [APPEND] in ALPHA mode (press [■] and the K key).

   f. All operations requiring register addresses accept those addresses in these forms:

      nn (a two-digit number)
      IND nn (INDIRECT: [■], followed fy a two-digit number)
      X, Y, Z, T, or L (a STACK address: [•] followed by X, Y, Z, T, or L)
      IND X, Y, Z, T or L (INDIRECT stack: [■] [•] followed by X, Y, Z, T, or L)

      Indirect addresses are specified by pressing [■] and then the indirect address. Stack addresses are specified by pressing [•] followed by X, Y, Z, T, or L. Indirect stack addresses are specified by pressing [■] [•] and X, Y, Z, T, or L.

Printer Listing	Keystrokes	Display
01◆LBL "SAMPLE"	[■] [LBL] [ALPHA] SAMPLE [ALPHA]	*01 LBL"SAMPLE*
02 "THIS IS A "	[ALPHA] *THIS IS A* [ALPHA]	*02"THIS IS A*
03 "⊢SAMPLE"	[ALPHA] [■] [APPEND] SAMPLE	*03" ⊢ SAMPLE*
04 AVIEW	[■] AVIEW [ALPHA]	*04 AVIEW*
05 6	6	*05 6*
06 ENTER↑	[ENTER↑]	*06 ENTER ↑*
07 -2	2 [CHS]	*07 -2*
08 /	[÷]	*08 /*
09 ABS	[XEQ] [ALPHA] ABS [ALPHA]	*09 ABS*
10 STO IND L	[STO] [■] [•] L	*10 STO IND L*
11 "R3="	[ALPHA] R3= [■] [ARCL] 03	*11"R3=*
12 ARCL 03	[■] [AVIEW]	*12 ARCL 03*
13 AVIEW	[ALPHA]	*13 AVIEW*
14 RTN	[■] [RTN]	*14 RTN*

TABLE OF CONTENTS

This book was developed and coded
under consultation from
Cary E. Reinstein

PROGRAM DESCRIPTION

APPOINTMENT CALENDAR

"Appointment Calendar" allows you to create in calculator memory a list of time-based appointments and reminders. This list can be copied to mass storage, edited, added to, and can be printed out as an organized memo calendar. Entries may be added or edited at any time and may be viewed, printed and/or set as alarms.

Appointment lists may be any length that is practical for the amount of calculator memory available. Each appointment requires 7 data registers and is composed of a date, time, sorting value (invisible to the user), and ALPHA message. Once a list has been created, it may be modified at any time using the routine labelled "B" (key [1/X] in USER mode) and added to using routine "a" ([shift] [Σ+] in USER MODE). Appointments cannot be deleted, but may be written over.

All appointments that have associated times will be set as alarms by the alarm setting routine (labelled "E", key [LN] in USER mode). Appointments with associated times of Ø are considered to be "REMINDERS" and are not set as alarms. Since the time module considers the time Ø to be midnight, appointments cannot be set for this time. This is no inconvenience since appointments can be set to within one-tenth of a second of midnight and still be set as alarms.

"Appointment Calendar" includes three versions of the "LOAD" and "STORE" routines in order to be compatible with the card reader, cassette drive and extended memory. The appropriate version for your system configuration should be the only version loaded. These routines can be called from the "APPT" program, but if the SIZE function must be performed while in these subroutines, control will not be passed back to "APPT".

The "STORE" routine creates and records the appointment list on a mass storage device.

The "LOAD" routine is used to download data files from the mass storage device to the HP-41's data registers. The cassette version of the routine checks for the necessary SIZE setting and prompts you if the current setting is inadequate. The card reader version assumes the proper SIZE is set. The extended functions version attempts to reset the SIZE if it is not large enough and will generate an error message if there is not enough room in calculator memory for both the programs and required data.

OPERATING LIMITS AND WARNINGS

At least six unused registers must be allowed in program memory for each alarm that is to be set. Fewer registers are required for alarms with fewer than 24-character messages. See the HP 82182A Time Module Owner's Manual for storage requirements of the various alarm types.

Anytime the calculator prompts you for a change in the SIZE setting, be sure you do not alter the stack in any way, or the appointments may be stored in unpredictable places.

SAMPLE PROBLEM

Appointments and reminders for the fourth week of January, 1983.

January 28,	5 p.m.	NICKS BIRTHDAY PRESENT
January 28,	9:15 a.m.	DENTIST APPT
January 28,	11:45 a.m.	LUNCH WITH JIM B.
January 29		MAKE LUNCH RES/NICK
January 29,	2 p.m.	RENEW LICENSE
January 30		NICK 5 TODAY

This example assumes a SIZE setting of less than 49 registers, MDY mode, flags 28 and 29 set and that no assignments have been made to the referenced keys.

DISPLAY	INPUT	FUNCTION
		[XEQ] "APPT"
APPTS	6	[R/S]
SIZE>=49.		[XEQ] "SIZE" 049
48.		[R/S]
DATE? ()/()	1.281983	[R/S]
TIME?	-5	[R/S]
MSSG?	NICKS BIRTHDAY PRESENT	[R/S]
DATE? 01/28		[R/S]
TIME?	9.15	[R/S]
MSSG?	DENTIST APPT	[R/S]
DATE? 01/28		[R/S]
TIME?	11.45	[R/S]
MSSG?	LUNCH WITH JIM B.	[R/S]
DATE? 01/29	1.291983	[R/S]
TIME?		[R/S]
MSSG?	MAKE LUNCH RES/NICK	[R/S]
DATE? 01/29		[R/S]
TIME?	-2	[R/S]
MSSG?	RENEW LICENSE	[R/S]
DATE? 01/30	1.301983	[R/S]
TIME?		[R/S]
MSSG?	NICK 5 TODAY	[R/S]
0.00		[C] Output calendar.
FRIDAY		[R/S]*

DISPLAY	INPUT	FUNCTION
01/28/83		[R/S]*
9:15:00 AM		[R/S]*
DENTIST APPT		[R/S]*
11:45:00 AM		[R/S]*
LUNCH WITH JIM B.		[R/S]*
5:00:00 PM		[R/S]*
NICKS BIRTHDAY PRESENT		[R/S]*
SATURDAY		[R/S]*
01/29/83		[R/S]*
REMINDER		[R/S]*
MAKE LUNCH RES/NICK		[R/S]*
2:00:00 PM		[R/S]*
RENEW LICENSE		[R/S]*
SUNDAY		[R/S]*
01/30/83		[R/S]*
REMINDER		[R/S]*
NICK 5 TODAY		[R/S]*
-1.0000		[R/S]*

* [R/S] is not required if a printer is attached.

STATUS

```
              SIZE  :  appointment * 7 + 7
               FIX  :  2, 4
TOTAL PROGRAM BYTES  :  APPT                    605
                        LOAD/STORE (card)        27
                                   (cassette)   105
                                   (x memory)    73
```

REQUIRED PERIPHERALS

HP 82182A Time Module

1 HP 82106A Memory Module (minimum)

Optional: HP 82104A Card Reader OR
HP 82160A HP-IL Module AND HP 82161A Digital Cassette Drive OR
HP 82180A Extended Functions/Memory Module

DATA REGISTERS

R00	Index for "A", "a", and "B"
R01	Last appointment date input
R02	1.01198 - Used to generate the sorting values
R03	Index for "C"
R04	Current earliest appoint in "C"'s sort
R05	Index constant used to refresh R03 in "C" Also used as index for "E"
R06	Last date output by "C"
R07	The remaining registers are grouped in blocks of 6 and are the appointments Where: R(0+7n) = Date R(1+7n) = Time R(2+7n) = Sort value R(3+7n) to R(6+7n) = Message (24 characters) for the nth appointment

FLAGS USED

00	Set	:	Edit mode
	Clear	:	All other modes
05	Set	:	Single appointment - used by "C"
	Clear	:	All other conditions
12	Set	:	Print double-wide
	Clear	:	Print single-wide
21	Set	:	Enable printer
	Clear	:	Disable printer
25	Set	:	Error detected
	Clear	:	No error detected
27	Set	:	USER mode on
	Clear	:	USER mode off

FUNCTION LABELS

Label	Function
"APPT"	Main global label
"A"	Establish appointment list
"a"	Add appointments to list
"B"	Edit current list
"C"	Memo calendar output
"D"	XEQ "STORE"
"d"	XEQ "LOAD"
"E"	Set alarms in current list

USER INSTRUCTIONS

SIZE: 7n + 7

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load "APPT".		[GTO]..	PACKING
2	Load one of the versions of the "LOAD"			
	program, if desired.		[GTO]..	PACKING
3	Set any desired modes. Display mode is			
	modified by the program and, consequently,			
	any mode the user has selected will not			
	be preserved.			
4	Initialize the program.		[XEQ]"APPT"	APPTS?
5	Input the number of appointments you plan			
	to store. No input or an input of zero			
	causes this prompt to be repeated.	#appts.	[R/S]	SIZE>=()
6	This prompt will only be generated if the			
	currently allocated SIZE is too small. If			
	the prompt does not appear, go to step 7.			
6a	Correct the SIZE.		[XEQ]"SIZE"nnn	
	Note: Do not modify the stack at this			
	time or errors may result.			
6b	Continue.		[R/S]	DATE? ()
7	The prompt "DATE?" is followed initially by			
	today's date in the current MDY or DMY			
	format. All later occurrences of this			
	prompt will contain the last input date			
	rather than today's date. Input the date			
	of the first appointment, be sure to			
	include the year. Omission of the date			
	will cause the use of the displayed date			
	as default. Dates are restricted to			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	those later than January 1, 1980.	date	[R/S]	TIME?
8	Input the time of the appointment. Input			
	time in the same way that you normally			
	would when using the HP 82182 Time			
	Module. No input or an input of zero			
	signifies a "REMINDER" with no associated			
	time. "REMINDER"'s are not set as alarms.			
	The time of midnight cannot be input as Ø,			
	but can be input as a time arbitrarily			
	close to 12 a.m. (i.e. 11:59:59.99 p.m.).	time	[R/S]	MSSG?
9	Input the message you wish, up to 24			
	characters. Keep in mind that all 24			
	characters will be output by the calendar			
	output portion (label "C") of this program,			
	but if set as an alarm only the first 12			
	characters will be displayed as the alarm			
	is activated.	message	[R/S]	DATE? ( )
10	If more appointments remain to be keyed			
	in, go to step 7. Otherwise, input			
	terminates with a display of Ø.ØØ.			
11	To Add Appointments:		[shift][a]	APPTS ADDED?
	Note: This cannot be done unless steps			
	1 - 1Ø have been performed.			
12	Key in the number of appointments you wish			
	to add to the current list. No input or			
	an input of zero causes this prompt to			
	be repeated.	#appts. more	[R/S]	SIZE>=( )

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
13	Go to step 6.			
14	To Edit the current list:		[B]	( )
	Note: This cannot be performed if steps			
	3 - 1Ø have not previously been performed.			
15	The date of the appointment is displayed			
	first. A new date may be entered or the			
	displayed date may be left as is. If the			
	input value is not a proper date, this			
	prompt will be repeated.	date	[R/S]	( )
16	The time of the appointment is displayed			
	next. If the appointment was stored with			
	no time, "REMINDER" will be displayed. A			
	new time (or no time) may be input now or			
	the value left as it is. No error check			
	is made on this value.	time	[R/S]	( )
17	The appointment's message is displayed next.			
	The message may be altered, or left alone.	message	[R/S]	
18	If there are more appointments in the			
	current list, go to step 15, otherwise			
	the program halts here.			
19	To print out a calendar of the current			
	appointment list.		[C]	
	The current appointment list is printed out			
	in chronological order. Different dates			
	are separated by headings of date and			
	day-of-week. All appointments on a			
	certain date are listed by time and			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	message. "REMINDERS" are listed as such.			
20	To set all alarms in the current			
	appointment list. All timed appointments			
	will be set as alarms.		[E]	
21	To store the current appointment list			
	on mass storage:		[D]	
22a	If the cassette drive or extended			
	functions/memory version of "LOAD" is			
	in memory:			FL NAME?
	Key in the file name - 6 characters max.	name	[R/S]	
22b	If the card reader version of "LOAD"			
	is in memory:			RDY nn OF mm
	Load the proper blank cards.			
23	To recall an appointment list from			
	mass storage:		[d]	
24a	If the cassette drive or extended			
	functions/memory version of "LOAD"			
	is in memory:			FL NAME?
	Key in the file name.	name	[R/S]	SIZE>=( )
	If the number of allocated data registers			
	is inadequate, alter the SIZE setting			
	and procede.		[XEQ]"SIZE"nnn	
24b	If the card reader version of "LOAD"			
	is in memory:			CARD
	Load the data cards. Note that the program			
	assumes that you have adequate data			
	space allocated.			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
25	To reset the appointment list:		[A]	APPTS?
	All previously stored appointments are			
	lost (except those that have been saved			
	on mass storage).			
	Go to step 5 of these instructions.			

PROGRAM LISTING

```
 01*LBL "APP
T"
 02 SF 21          Enable printer and
 03 SF 27            USER mode
 04*LBL A          Set up initial list
 05 CF 00          Not EDIT mode
 06 CLX
 07 "APPTS?"       No. of appointments
 08 PROMPT
 09 X=0?           No input?
 10 GTO A
 11 7
 12 *
 13 7
 14 +
 15 XEQ 99
 16 FC?C 25        Check SIZE
 17 PROMPT         Bad SIZE?
 18 .1             Set up index
 19 %
 20 7
 21 +
 22 STO 00
 23 DATE           Default date
 24 STO 01
 25 1.01198        Jan. 1, 1980
 26 STO 02
 27*LBL 20         Input loop
 28 "DATE? "
 29 RCL 01         Default
 30 FS? 00         EDIT mode?
 31 RCL IND
00
 32 FS? 00
 33 CLA
 34 FIX 2
 35 FS? 00
 36 FIX 4
 37 ADATE          Format prompt
 38 PROMPT
 39 RCL 02         Test for valid date
 40 X<>Y
 41 SF 25
 42 DDAYS
 43 FC?C 25        Not valid?
 44 GTO 20
 45 LASTX          Get date
 46 STO IND
00
 47 STO 01         New default
 48 "TIME?"
 49 ISG 00
 50 RCL IND        Last time
00
 51 FC? 00         Not EDIT mode?
 52 CLX
 53 FC? 00
 54 GTO 09
 55 CLA
 56 ATIME          Last time
 57 X=0?
 58 "REMINDE       No time
R"
 59*LBL 09
 60 PROMPT
 61 STO IND        New time
00
 62 X>0?           AM?
 63 GTO 10
 64 ABS
 65 12
 66 +
 67*LBL 10
 68 1 E2
 69 /
 70 RCL 02         Jan. 1, 1980
 71 RCL 01         Date
 72 DDAYS
 73 +
 74 ISG 00
 75 STO IND        Sort value
00
 76 "MSSG?"
 77 CF 21
 78 FC? 00         Not EDIT mode?
 79 AVIEW          Display prompt
 80 CLA
 81 FC? 00         Not EDIT mode?
 82 GTO 08
 83 ISG 00
 84 ARCL IND       Last message
00
 85 ISG 00
 86 ARCL IND
00
 87 ISG 00
 88 ARCL IND
00
 89 ISG 00
 90 ARCL IND
00
 91 4              Decrement index
```

PROGRAM LISTING

```
 92 ST- 00
 93*LBL 08          Prompt for message
 94 AON
 95 STOP
 96 AOFF
 97 4
 98*LBL 07          Store new message
 99 ISG 00
100 ASTO IND
 00
101 ASHF
102 DSE X
103 GTO 07
104 ISG 00          Next appointment
105 GTO 20
106 CF 00
107 RTN
108*LBL a           Add appointments
109 CF 00           Not EDIT mode
110 "APPTS A
DDED?"
111*LBL 14          Add loop
112 PROMPT          More?
113 X=0?            Invalid input?
114 GTO a
115 7
116 *
117 RCL 00
118 FRC
119 1 E3
120 *
121 STO 00          Last register used
122 +
123 1
124 +
125 XEQ 99          Check SIZE
126 FC?C 25         Bad SIZE?
127 PROMPT
128 .1
129 %
130 1
131 +
132 ST+ 00          New index
133 GTO 20          Goto input loop
134 RTN
135*LBL B            EDIT mode
136 RCL 00
137 FRC
138 7
139 +
140 STO 00          Reset index
```

```
141 FIX 4
142 SF 00
143 GTO 20          Goto input loop
144*LBL C           Calendar output
145 RCL 00
146 FRC
147 16
148 +
149 7 E-5
150 +
151 STO 05          New index
152 INT
153 LASTX
154 FRC
155 1 E3
156 *
157 INT
158 CF 05
159 X<Y?            Only one appointment?
160 SF 05           One appointment
161 CLX
162 STO 06          Last date used
163*LBL 30          Sort loop
164 9
165 STO 04          First sort number location
166 FS? 05          Only one appointment?
167 GTO 36
168 RCL 05
169 STO 03          Refresh index
170 RCL 12          Second sort number
171*LBL 31          Search loop
172 RCL IND         Last sort number
04
173 X<=0?           Already used?
174 GTO 34
175 RCL IND         Next sort number
03
176 X<=0?           Already used?
177 GTO 33
178 X<Y?            X>= Y?
179 X=Y?
180 GTO 33
181*LBL 34          New last number
182 RCL 03
183 STO 04
184*LBL 33          Next sort number
185 ISG 03
186 GTO 31
187 RCL IND
04
188 X>0?            Not used?
```

PROGRAM LISTING

```
189 GTO 36
190◆LBL 38          Re-establish sort
191 RCL 05            numbers as positive
192 FC? 05
193 STO 04
194 -1
195 ST* 09
196◆LBL 37
197 ST* IND
04
198 ISG 04
199 GTO 37
200 CLD
201 RTN
202◆LBL 36
203 -1              Flag the number as
204 ST* IND           used
04
205 RCL 04          Index of appointments
206 INT
207 2
208 -
209 RCL 06          Last date
210 RCL IND         This date
Y
211 STO 06
212 X=Y?            Same?
213 GTO 35
214 SF 21
215 ENTER↑
216 DOW             New day
217 XEQ IND
X
218 "⊢DAY"
219 ADV
220 SF 12
221 AVIEW
222 RDN
223 CLA
224 FIX 4
225 ADATE           New date
226 AVIEW
227◆LBL 35
228 ADV
229 CF 12
230 CLA
231 RCL 04
232 INT
233 1
234 -
235 RCL IND         Appointment time
X
236 ATIME
237 X=0?
238 "REMINDE        No time
R"
239 AVIEW
240 CLX
241 2
242 +
243 CLA
244 ARCL IND        Message
 X
245 1
246 +
247 ARCL IND
 X
248 1
249 +
250 ARCL IND
 X
251 1
252 +
253 ARCL IND
 X
254 AVIEW
255 FS? 05          One appointment
256 GTO 38
257 GTO 30
258◆LBL 00          Days
259 "SUN"
260 RTN
261◆LBL 01
262 "MON"
263 RTN
264◆LBL 02
265 "TUES"
266 RTN
267◆LBL 03
268 "WEDNES"
269 RTN
270◆LBL 04
271 "THURS"
272 RTN
273◆LBL 05
274 "FRI"
275 RTN
276◆LBL 06
277 "SATUR"
278 RTN
279◆LBL E           Set alarms
280 RCL 00
281 FRC
```

PROGRAM LISTING

```
282 7
283 +
284 STO 05          New Index
285◆LBL 50          Alarm set loop
286 CLST
287 CLA
288 RCL IND         Date
05
289 ISG 05
290 RCL IND         Time
05
291 ISG 05          Sort value
292 ISG 05
293 ARCL IND        Message
 05
294 ISG 05
295 ARCL IND
 05
296 ISG 05
297 ARCL IND
 05
298 ISG 05
299 ARCL IND
 05
300 X≠0?            No time?
301 XYZALM          Set alarm
302 ISG 05
303 GTO 50          Loop
304 RTN
305◆LBL D           XEQ 'STORE'
306 CF 25
307 XEQ "STO
RE"
308 RTN
309◆LBL d           XEQ 'LOAD'
310 CF 25
311 XEQ "LOA
D"
312 RTN
313◆LBL 99
314 "SIZE>="        SIZE test
315 FIX 0
316 ARCL X          Desired SIZE
317 1
318 -
319 SF 25
320 STO IND         Highest register
X
321 END
```

PROGRAM LISTING

Cassette Version

```
 01◆LBL "LOA
D"                "LOAD" routine
 02 XEQ 20        "FL NAME?" routine
 03 CLX
 04 SEEKR         Find file
 05 READRX        Read register zero
 06 XEQ 21        Find file size
 07 ASTO Y        Save file name
 08 FIX 0
 09 "SIZE>="
 10 ARCL X        Desired SIZE
 11 1
 12 -
 13 SF 25
 14 STO IND
X                 Highest register
 15 FC?C 25
 16 PROMPT        Bad SIZE
 17 CLA
 18 ARCL Y        File name
 19 CLX
 20 SEEKR         Reset file to R00
 21 READR         Read file contents
 22 RTN
 23◆LBL "STO
RE"               "STORE" routine
 24 XEQ 20        "FL NAME?" routine
 25 XEQ 21        Find file size rqd.
 26 CF 25         Allow errors
 27 CREATE
 28 0
 29 SEEKR         Position to file
 30 RCL 00        Index
 31 FRC
 32 WRTRX         Write list
 33 RTN
 34◆LBL 20        Name prompt routine
 35 "FL NAME
?"
 36 AON
 37 STOP
 38 AOFF
 39 RTN
 40◆LBL 21        Desired file size
 41 RCL 00        from index
 42 FRC
 43 1 E3
 44 *
 45 1
 46 +
 47 END
```

X Function/Memory Version

```
 01◆LBL "LOA
D"                "LOAD" routine
 02 XEQ 00        "FL NAME?" routine
 03 SIZE?         Allocated registers
 04 FLSIZE        File size
 05 CF 25         Allow errors
 06 X>Y?          Need more room?
 07 PSIZE         Allocate registers
 08 CLX
 09 SEEKPT        Position to file
 10 GETR          Load file
 11 RTN
 12◆LBL "STO
RE"               "STORE" routine
 13 XEQ 00        "FL NAME?" routine
 14 RCL 00        Index
 15 FRC           Determine necessary
 16 1 E3          file size
 17 *
 18 1
 19 +
 20 CF 25         Allow errors
 21 CRFLD         Create the data file
 22 RCL 00        Index
 23 FRC
 24 SEEKPT        Position to the file
 25 SAVERX        Store the registers
 26 RTN
 27◆LBL 00        "FL NAME?" routine
 28 "FL NAME
?"
 29 AON
 30 STOP
 31 AOFF
 32 END
```

Card Reader Version

```
 01◆LBL "LOA
D"                "LOAD" routine
 02 RDTA          Read data
 03 RTN
 04◆LBL "STO
RE"               "STORE" routine
 05 RCL 00        Index
 06 FRC
 07 WDTAX         Write registers
 08 END
```

PROGRAM DESCRIPTION

WORLD TIME CONVERTER

While it is useful to know the time in a foreign city, that time will not always correspond to business hours or other convenient calling or arrival time. "World Time Converter" is a programmable alarm clock that displays the time and date of a foreign city and can set an alarm that corresponds to a time in the destination city's time zone.

"WTIME" does a straightforward time conversion by adding the time differences between each city of interest and Greenwich time. Each location must be keyed into program memory by the user so as to eliminate the need for a large data base of locations and time offsets.

"T2" prompts for the time of interest in the destination city, and then sets an alarm that is activated in the home city's time when the desired time in the destination city is reached. The name of the city is flashed as a message.

City	Offset	City	Offset	City	Offset
Alexandria	2	Gdansk	1	Oslo	1
Amsterdam	1	Geneva	1	Paris	1
Athens	2	Haifa	2	Prague	1
Auckland	0	Havana	-5	Rangoon	6.3
Baghdad	3	Helsinki	2	Rio de Janeiro	-3
Bangkok	7	Hong Kong	8	Rome	1
Beijing	8	Istanbul	2	Saigon	8
Belfast	0	Jerusalem	2	Santiago, Chile	-4
Berlin	1	Johannesburg	3	Seoul	9
Bogota	-5	Karachi	5	Shanghai	8
Bombay	5.3	Kyoto	9	Singapore	7.3
Brussels	1	Leningrad	3	Stockholm	1
Bucharest	2	Lima	-5	Sydney	10
Budapest	1	Lisbon	1	Teheran	3.3
Buenos Aires	-3	London	0	Tel Aviv	2
Calcutta	5.3	Madrid	1	Tokyo	9
Capetown	2	Manila	8	Vancouver, BC	-8
Caracas	-4	Melbourne	10	Vienna	1
Copenhagen	1	Mexico City	-6	Warsaw	1
Dacca	6	Montevideo	-3	Wellington	12
Delhi	5.3	Montreal	-5	Yokohama	9
Djakarta	7	Moscow	3	Zurich	1
Dublin	0	Nagasaki	9		
North American Time Zones:					
Atlantic	-4	Mountain	-7	Alaska-Hawaii	-10
Eastern	-5	Pacific	-8	Bering	-11
Central	-6	Yukon	-9		

OPERATING LIMITS AND WARNINGS

The time adjustment chart does not take into account Daylight Savings Time or any other adjustments due to local laws and ordinances. These must be known and input by the user of the program. If your appointment time is critical, please verify the offsets in this against some standard reference source for correct time.

SAMPLE PROBLEM

An electronics manufacturer in Corvallis, Oregon frequently finds it necessary to call Singapore. The caller needs to phone at a convenient business hour in Singapore's time zone.

To run this example the user must key the example cities, Corvallis and Singapore, into the program. The actual answer will depend upon the time of day in which the example is run. If the output time is 15-1/2 hours later than the input Corvallis time, the program has been run correctly.

DISPLAY	INPUT	FUNCTION
The program steps to insert would be as follows:		
		[GTO] ..
		[PRGM]
		01 LBL "CORV"
		02 -8
		03 "CORVALLIS"
		04 GTO "WT"
		05 LBL "SING"
		06 7.3
		07 "SINGAPORE"
		08 GTO "WT"
		09 END
		[PRGM]
Find the Corvallis offset		[XEQ] "CORV"
Find the Singapore offset		[XEQ] "SING"
Find the time in Singapore		[XEQ] "WTIME"
Select a time of day in Singapore that is 2 or 3 minutes later than the time of day on your clock for the next portion of the problem.		
You would like to be reminded to call Singapore at nn:nn a.m. or p.m. their time.		

DISPLAY	INPUT	FUNCTION
		[XEQ] "T2"
SINGAP. TIME?	HH.MMSS	[R/S]

The time of day that the alarm is to be activated is left in the X register.

STATUS

SIZE : 003

FIX : 2

TOTAL PROGRAM BYTES : 201

DATA REGISTERS

00	Time difference between the input city and Greenwich
01-02	City name

FLAGS USED

21	Set	: Printer enabled
	Clear	: Printer disabled
55	Set	: Printer exists
	Clear	: Printer does not exist

FUNCTION LABELS

Label	Function
"WTIME"	Calculates and displays time of day in foreign city
"T2"	Sets relative time alarm
"WT"	Accessed by user written routines

USER INSTRUCTIONS

SIZE: 003

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load program "WTIME".		[GTO] ..	PACKING
	If you have not yet programmed the cities			
	that you require, go to step 7.			
2	Find the home city. Do not alter the			
	stack at this point.		[XEQ] city	city's name
3	Find the destination city. Do not alter			
	the stack at this point.		[XEQ] city	city's name
4	Calculate the time of day in the			
	destination city.		[XEQ] "WTIME"	(time,day)
5	Set an alarm in the home city's time			
	that will activate at a time in the			
	destination city.		[XEQ] "T2" or	
			[R/S]	(city). TIME?
6	Input the time of day in the destination			
	city that you require.	HH.MMSS	[R/S]	( ) : ( )
	The alarm time is left in the X-register.			
7	To program cities of interest: Position			
	the calculator to the end of program			
	memory.		[GTO] ..	PACKING
	Label the routine. Use a global label.			
	Key in the number taken from the time			
	zone chart including the sign (if any).			
	Key in the name of the city using no			
	more than 12 characters.			
	Key in as the last line for each city:			
	[GTO] "WT".			

PROGRAM LISTING

```
 01◆LBL "WTI
ME"
 02 X<0?          Is destination city earlier
                  than Greenwich time?
 03 GTO 08
 04 X<>Y          If not, then make home
                  city positive and add
 05 ABS           numbers
 06 +
 07 GTO 09
 08◆LBL 08
 09 -             If destination city earlier,
                  subtract
 10 CHS
 11◆LBL 09
 12 STO 00        Save time adjustment for
                  LABEL 'T2'
 13 TIME
 14 XEQ 10        Do relative time sub-
                  routine to get clock and
 15 CLA           date in stack
 16 FIX 2
 17 ATIME         Get time into ALPHA
 18 "⊢ "
 19 X<>Y
 20 DOW           Calculate day from date
                  in Y
 21 X<> L
 22 X<>Y
 23 XEQ IND
L                 Get day string
 24 AVIEW
 25 RTN
 26◆LBL "T2"
 27 CLA
 28 ARCL 01       Get city name as message
                  (destination city)
 29 "⊢. TIME
?"
 30 PROMPT        Allows negative input to
                  be recognized as PM
 31 -12           time, to be consistent
                  with Tim Module in-
 32 X<>Y          put formats
 33 X<0?          If negative, convert to
                  24-hour format
 34 +
 35 ABS
 36 24            Allows routine 10 to cal-
                  culate time if next day
 37 +
 38 RCL 00        Recall adjustment and
                  subtract from midnight
 39 CHS
 40 XEQ 10
 41 CLA
 42 ARCL 01       Get city name as alarm
                  message
 43 ARCL 02
 44 XYZALM        Set
 45 RTN
 46◆LBL 10        Almrel routine to calcu-
                  late time offset and
 47 HMS+          leave alarm parameters
 48 ENTER↑        in X
 49 ENTER↑
 50 24
 51 /             Number of days
 52 INT
 53 DATE
 54 X<>Y
 55 DATE+
 56 LASTX
 57 24
 58 *
 59 ST- Z         Set up stack for XYZALM
                  time, date, no reset
 60 CLX
 61 STO T
 62 RDN
 63 X<>Y
 64 RTN
 65◆LBL 00
 66 "⊢SUN"        Day of the week strings
 67 RTN
 68◆LBL 01
 69 "⊢MON"
 70 RTN
 71◆LBL 02
 72 "⊢TUE"
 73 RTN
 74◆LBL 03
 75 "⊢WED"
 76 RTN
 77◆LBL 04
 78 "⊢THU"
 79 RTN
 80◆LBL 05
 81 "⊢FRI"
 82 RTN
 83◆LBL 06
 84 "⊢SAT"
 85 RTN
 86◆LBL "WT"
 87 FC? 55
 88 CF 21         If no printer, clear printer
                  enable flag to prevent
 89 AVIEW         halt
 90 ASTO 01       Save city name for
                  messages
 91 ASHF
 92 ASTO 02
 93 END
```

PROGRAM DESCRIPTION

EXERCISE MONITOR

This program can be used for timing periods of aerobic exercise preceded by a pulse count and followed by pulse counts at one-and five-minute intervals. Runners can input the various distance markers and an overall time goal for the course and alarms will signal when each marker should be reached so as to remain on target. Splits may be stored and later replayed and compared to the goals. The course is easily set up before the exercise period and remains in the HP-41 until ready. A course can also be saved on any storage medium.

The program consists of six main segments which use many of the capabilities of the HP 82182A Time Module: message alarms, stopwatch and control alarms.

Label "RUN" identifies the program, initializes flags and interactively sets up the running course and options. If the user needs to record a pulse count, the program will provide a fifteen-second pulse timing interval at the start and end of the course, followed at the end in five minutes by another pulse count to monitor recovery. Any prompted for option that is not wanted can be skipped by pressing [R/S] with no input. The user is prompted for a course goal, if any, and the successive distances of the various course. The goal time is divided into segments that will activate a control alarm, "M", at the moment that the distance marker should be passed to remain on target. When all markers have been input, the number of splits desired is prompted for. The usefulness of storing splits is best realized by having a coach or friend take them.

The initialized program halts at label "GO" (which is a global label to facilitate a key assignment). "GO" will activate the stopwatch, the initial pulse count (if chosen), and the time limit alarm.

"SPLIT" uses the programmable stopwatch command to store splits in successive registers by incrementing a counter in register 12. If storing splits was chosen as an option during the input portion of the program, the marker alarms are of short duration to allow for greater accuracy.

Label "FIN" simply stops the stopwatch, and trips the pulse intervals after 15 seconds and 04:45 minutes. The three pulse counts remain temporarily in registers 01, 02, and 03.

"REPLAY" recalls the stored pulse counts and all stored splits formatted for printing (may be replayed without printer).

"M" is the control alarm that is activated at each marker in proportion to the total goal time for the course. If the option of storing splits was chosen, "M" sounds two high pitch tones only and resets to the next marker. If splits were not chosen, eight tones would sound and the marker number would be displayed. If the program were to run for several seconds to output audio feedback signals and format an alpha display, a split could not be stored close to the "marker" (if they coincided). The keyboard and user functions are only available when a program is not running.

Registers 00 through 11 are only used during the input phase of the program. When the course splits and pulse counts have been displayed and/or printed, they are available for plotting routines using the PRPLOT program of the HP 82143A printer and HP 82160A HP-IL Module.

SAMPLE PROBLEM

Simulate the training course undertaken by Wonder Woman on her secret island, who, with her HP-41 to time her goals and performance, desires to run an irregularly marked course of ten kilometers in six minutes.

DISPLAY	INPUT	FUNCTION
		[XEQ] "RUN"
PULSE? Y/N	Y	[R/S]
LIMIT HMS?		[R/S]
GOAL HMS?	.06	[R/S]
DISTANCE 1?	1	[R/S]
DISTANCE 2?	2.5	[R/S]
DISTANCE 3?	4.2	[R/S]
DISTANCE 4?	7	[R/S]
DISTANCE 5?	9	[R/S]
DISTANCE 6?	10	[R/S]
DISTANCE 7?		[R/S]
N SPLITS?	1	[R/S]
READY		

In actual practice, a runner desiring to run a ten kilometer course in 40 minutes would input the distance to known points to the end of the course. If we were to recall the data registers, beginning with R19, where Wonder Woman's distance markers were stored, we could see how they were divided into linear time intervals.

```
R19 = 0.003600  (00:00:36.00)
R20 = 0.013000  (00:01:30.00)
R21 = 0.023120  (00:02:31.00)
R22 = 0.041200  (00:04:12.00)
R23 = 0.052400  (00:05:24.00)
R24 = 0.060000  (00:06:00.00)
```

To begin the course: [R/S] or [XEQ] "GO"

The first display is "PULSE", signalling the start of a 15 second interval to be used for taking a pulse count. After 15 seconds, a message alarm is activated and you are prompted with "BEATS=?". Acknowledge the alarm and input the number of beats counted, in this case: 17.

DISPLAY	INPUT	FUNCTION
	17	[R/S]
RATE=68		
A message alarm will be activated with the display: "READY". Acknowledge the alarm and proceed.		
		[R/S]
0		
As each marker should be passed, a series of eight high pitched tones will sound followed by the display: "MARK (n)". When the end of the course is reached (in this case, after "MARK 6") execute "FIN". (in practice, the user might wish to select key assignments for functions like "GO" and "FIN").		
MARK 6		[XEQ] "FIN"
WAIT		
PULSE		
BEATS=?	29	[R/S]
After 4:45 minutes, the "WAIT" and "PULSE" alarms are repeated to allow you to monitor pulse recovery. To replay the stored pulse counts (no splits have been stored or they, too, would be displayed):		
		[XEQ] "REPLAY"
PULSE 1=68		[R/S]
PULSE 2=116		[R/S]
PULSE 3=(n)		[R/S]
00:00:00.00 (no split)		[R/S]
The various options offered by the program may be used in any combination. For example, store splits under program control by skipping all of the initial prompts until "N SPLITS?".		
If the program "times out" and the machine turns off automatically after 10 minutes, no flag used by the program will be affected.		

STATUS

SIZE : # of Distance Markers + # of Splits + 19

FIX : 0, 2, 4, 6

TOTAL PROGRAM BYTES : 528

DATA REGISTERS

00	Index for storing pulse counts (1.003)
01	Start pulse
02	Finish pulse
03	Recovery pulse
04	SIZE (also used as recall split index)
05	1 E-3 (.001) a constant repeated in the program
06	Split index
07	15 E-4 used to time pulse and wait intervals
08	Marker n, during input
10	"READY"
11	Index for recalling distance, converting to time intervals
12	Highest distance register n, used to compute size required, store splits
13	Distance marker store index
14	Goal HH.MMSS
15	Start "GO" time
16	Time limit
18	Loop control for "M" control alarm
19-23	Store distance marker intervals
24-25	Splits begin after last "M" register

FLAGS USED

01	Set	:	Set pulse alarm and wait interval
	Clear	:	No pulse count
02	Set	:	Allow pulse alarm
	Clear	:	Disallow alarm
05	Set	:	Skip marker display, shorten tones to 2
	Clear	:	Allow marker display, allow all tones
21	Set	:	Printer enabled
	Clear	:	Printer disabled
25	Set	:	No error detected
	Clear	:	Error detected
26	Set	:	Audio enabled
	Clear	:	Audio disabled
27	Set	:	User mode on
	Clear	:	User mode off
29	Set	:	Radix separators displayed
	Clear	:	No separators
55	Set	:	Printer exists
	Clear	:	No printer

FUNCTION LABELS

Label	FUNCTION
"RUN	Initialize, set indices and flags for options, check size
"GO"	Start activity
"SPLIT"	Store splits
"FIN	End activity, trigger pulse counts, if optioned - stops stopwatch
"REPLAY"	Read pulse counts and splits

USER INSTRUCTIONS

SIZE: *

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load "RUN"		[GTO] . .	PACKING
2	Initialize "RUN"		[XEQ]"RUN"	PULSE? Y/N
3	If you wish to check your pulse:	"Y" or "N"	[R/S]	LIMIT HMS?
4	Input the maximum duration of the exercise			
	period. This step may be skipped if it is			
	not applicable.	HH.MMSS	[R/S]	GOAL HMS?
5	Input the goal time for completion of the			
	course. This step may be skipped if it			
	is not applicable.	HH.MMSS	[R/S]	DISTANCE N?
6	Input the elapsed distance to the			
	first marker.	distance	[R/S]	DISTANCE N?
7	Input the elapsed distance to next marker.	distance	[R/S]	DISTANCE N?
8	Repeat line 7 until all distance markers			
	are inputted.		[R/S]	N. SPLITS?
9	Input the number of splits you plan to			
	store. If skipped, the distance marker			
	output display will be shortened to			
	allow greater accuracy in the displayed			
	splits. To see the full output display,			
	input at least one split.	HH.MMSS	[R/S]	READY
10	When ready to begin:			
10a	Immediately following step 9:		[R/S]	PULSE
10b	Otherwise:		[XEQ] "GO"	PULSE
11	Take a pulse count until the program			
	prompts for the total.			BEATS=?
12	Stop counting your pulse when the			
	* # of splits + # of distance markers + 19			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	calculator flashes the display. The			
	display will flash until the user			
	acknowledges the alarm.		[R/S]	Ø.ØØØØ
13	Input total beats counted.	beats	[R/S]	RATE=( )
14	Start your exercise period.		[R/S]	Ø
15	To store splits: ("SPLITS" should be			
	assigned to a key to be effective).		[XEQ] "SPLIT"	.nnn
16	During the exercise period, if the goal			
	option was chosen, there are two possible			
	outputs:			
16a	Split option: feedback at the checkpoint			
	consist of two high pitched tones and			
	no display.			
16b	No split option: feedback consist of 8			
	high pitch tones and an alpha display.			MARK( )
	Either way, no acknowledgement			
	is required.			
17	To end the exercise period:		[XEQ]"FIN"	
	There are two possible results of			
	this action:			
17a	No pulse option: two tones are generated			
	and the program terminates.			
17b	Pulse option: A 15-second interval is			
	timed in order to prepare for a pulse			
	count.			WAIT
18	An alarm is sounded at the end of the			
	period.			PULSE

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
19	The display will flash until the user			
	acknowledges the alarm.		[R/S]	Ø.ØØØØ
20	Input total beats counted.	beats	[R/S]	RATE=( )
21	To read back splits and/or pulse:		[XEQ]"REPLAY"	PULSE N=( )
			[R/S]	.
			[R/S]	.
				nn:nn:nn:nn
				.
				.
				.

PROGRAM LISTING

```
01◆LBL "RUN
"
02 CF 01          Initialize the flags that
03 CF 05            will be tested by the
04 CF 29            program
05 "Y"
06 ASTO Y         Prepare stack for option
                    test
07 "PULSE?        Pulse option?
Y/N"
08 AON
09 STOP           Stop in ALPHA mode
10 AOFF
11 ASTO X         ALPHA to stack for
12 X=Y?             comparison
13 SF 01
14 CLRG           Prevents recall of use-
15 1 E-3            less splits or goal
16 STO 05           points
17 3              E-3 is a constant that
18 *                will be used by the
19 STO 00           program to form ISG
                    loop control numbers
20 2              Register 00 will store
21 /                pulse counts
22 STO 07
23 "LIMIT H       Time Limit Option?
MS?"
24 CLX            0 is default
25 PROMPT
26 STO 16
27 "GOAL HM       Goal for the course
S?"                 (option) - may also
                    be skipped
28 CLX            Default is 0
29 PROMPT
30 STO 14
31 X=0?           If no goal, skip next
32 GTO 02           input sequence
33 17
34 STO 13         Store index for distance
35◆LBL 00           marker input and store
36 1              Loop entry point
37 ST+ 13         Increment counter,
                    distance
38 "DISTANC
E "
39 FIX 0
40 RCL 13         Get increment value
41 17
42 -
43 ARCL X         Format ALPHA
44 "⊢?"             prompt
45 FIX 2
46 CLX
47 PROMPT
48 STO IND        Store distance
13
49 X≠0?           If not 0, save number of
50 STO 08           highest input
51 X≠0?           If not 0, continue input
52 GTO 00           loop
53 RCL 13         If 0, format index for
54 18               recall of distances and
55 -                conversion to time for
                    alarms
56 RCL 05
57 *
58 STO 17         Loop control
59 RCL 13
60 STO 12         Save to form split index
                    later
61 1
62 -
63 RCL 05         R05=1E-3, build ISG
64 *                loop number
65 18             Get start register
66 +
67 STO 11
68 STO 13
69◆LBL 01
70 RCL 14         R14=course goal
71 HR             To decimal time
72 RCL IND        Recall first distance
11                  marker
73 RCL 08         Recall number of
74 /                markers
75 *              Divide into goal and
76 HMS              convert to time
77 STO IND        Replace distance with
11                  time
78 ISG 11         Increment counter,
79 GTO 01           repeat loop
80 RCL 13         Loop ended, get start
81 STO 11           register saved at line 68
                  Re-use R11
82◆LBL 02
83 "N SPLIT       Split option?
S?"
84 CLX
85 PROMPT         Prepare input test
86 X≠0?
87 SF 05          Option chosen, set split
                    flag
88 "RESIZE>       Test for sufficient size,
="                  to ensure that program
89 FIX 0            will not fail to store
90 SF 25            required n splits
91 RCL 12         Format prompt
92 +              Set error flag
```

PROGRAM LISTING

```
 93 1
 94 +
 95 ARCL X
 96 LASTX
 97 -
 98 STO 04
 99 RCL IND
X
100 FC? 25
101 PROMPT
102 FC?C 25
103 GTO 02
104 RCL 04
105 RCL 05
106 *
107 ST+ 12
108 RCL 12
109 STO 06
110 SF 27
111 FIX 4
112 "READY"
113 ASTO 10
114 RCL 10
115 RTN
116*LBL "GO"
117 CF 02
118 FC? 55
119 CF 21
120 SF 26
121 FS? 01
122 XEQ 05
123 RCL 10
124 FS? 01
125 RTN
126 SF 02
127 RCL 11
128 STO 13
129 RUNSW
130 CLST
131 SETSW
132 TIME
133 TONE 9
134 TONE 9
135 STO 15
136 RCL 16
137 X=0?
138 GTO 04
139 CLA
140 FIX 4
141 ATIME24
142 HMS+
```

Compute size requirement

Format ALPHA display if needed

SIZE

Test if register exists

If test failed, then display prompt

If the flag was cleared, assume that size will be changed

Store index

Split index

Set fix mode to display time

Save prompt for re-use

Leave in X register to be seen when machine is turned back on

Global label allows a key assignment

Flag 02 will be used to toggle pulse alarm

Set tone flag

If pulse option chosen, do pulse alarm

Repeat flag test as no op or return skipped if no flag

Now toggle flag 02 on for later test by FIN routine

Get index

Start activity timer

Set stopwatch to 0

Save start time

Signal activity start

Time will be used for setting alarms

If time limit chosen, start here

If not, skip

Time limit message will be time formatted nn:nn:nn

```
143 XYZALM
144*LBL 04
145 RCL 14
146 X≠0?
147 GTO 20
148 RTN
149*LBL "SPL
IT"
150 RCLSW
151 FIX 4
152 TONE 9
153 TONE 8
154 ISG 12
155 STO IND
12
156 RTN
157*LBL "FIN
"
158 STOPSW
159 TONE 9
160 TONE 9
161 FC? 01
162 RTN
163 CLST
164 TIME
165 RCL 07
166 HMS+
167 "↑↑"
168 XYZALM
169 "WAIT"
170 PROMPT
171*LBL 05
172 FC? 55
173 CF 21
174 "PULSE"
175 AVIEW
176 "BEATS=?
"
177 CLST
178 RCL 07
179 TIME
180 HMS+
181 TONE 9
182 TONE 9
183 XYZALM
184 .043
185 HMS+
186 "↑↑FIN"
187 FS?C 02
188 XYZALM
189 CLX
```

Time limit alarm

Goal time

If there is a goal, set marker alarms, if not, do nothing

Global label allows faster execution if assigned to USER key

Get stopwatch

Display when done

Tick-tock feedback that split was actually stored

Increment split counter

Stop

Global label for end routine

Terminate activity timing

Audible feedback

Pulse not wanted? STOP

Else, set up stack for alarm command clock

+15 seconds (stored at initialization)

ALPHA alarm command

Display required activity

No printer?

Now take pulse

Message for alarm

Set up stack for alarm

15 seconds added to clock

Start signal

Do again in 4½ minutes

Alarm entry point

Alarm toggle flag

Clear display. Wait for pulse alarm to display 'BEATS=?'

PROGRAM LISTING

```
190 STOP
191 4                    Multiply pulse count by
192 *                      4 to get rate/minute
193 ISG 00
194 STO IND              Save pulse count R01,
00                         R02, R03
195 "RATE="              Display
196 FIX 0
197 ARCL X               Recall splits and pulse
198 AVIEW                  counts
199 RTN
200◆LBL "REP
LAY"
201 SF 21                Enable printer
202 1.003
203 STO 00               Index for pulse count
204 FIX 0                Display significant digits
205◆LBL 06                 only
206 "PULSE "
207 ARCL 00              Index number
208 "⊢="
209 RCL IND
00
210 ARCL X
211 X≠0?                 Display only if not 0
212 AVIEW                Display and get next, do
213 ISG 00                 not stop if printer is
214 GTO 06                 attached
215 RCL 06               Number of splits
216 1
217 +
218 FIX 6                Time format to ALPHA
219 STO 04               Store index
220◆LBL 07
221 RCL IND              Get splits
04
222 CLA
223 ATIME24              Into ALPHA
224 AVIEW                Print, or stop and get
225 ISG 04                 next
226 GTO 07
227 CLX                  Leave display clear when
228 FIX 2                  done
229 RTN
230◆LBL "M"              Control alarm entry
231 8                      point
232 FS? 05               If splits option chosen,
233 GTO 08                 two quick tones only
234 TONE 9
235 TONE 9               Set up next control
236 GTO 20                 alarm for marker
237◆LBL 08               Tone loop, sound eight
238 TONE 9                 high pitch tones
239 DSE X
240 GTO 08
241 FIX 0                Format ALPHA display
242 "MARK "
243 RCL 13               Get number of marker
244 18
245 -
246 INT
247 ARCL X               AVIEW must not halt
248 CF 21                  program or next control
249 AVIEW                  alarm will not be set
250◆LBL 20
251 CLST                 Set up stack for alarm
252 RCL 15               Start time
253 RCL IND              Next goal point
13
254 HMS+
255 "↑↑M"                Control alarm
256 ISG 17               Loop control
257 XYZALM
258 CLX
259 ISG 13               Increment goal register
260 END
```

PROGRAM DESCRIPTION

AUTOMOBILE TRIP COMPUTER AND SPEED CALIBRATOR

"TRIP" and "CAL" are two programs designed to work together to perform time related functions on automobile trips. Users of the "TRIP" program can calculate their estimated time of arrival based on their travel speed and/or the speed required to arrive at a planned destination in a certain amount of time. The program has routines for setting periodic alarms, converting tachometer readings to speed in a given gear and correcting a speedometer reading. Alarms may be set, cleared, changed or merely silenced at any time. One feature of the program is its ability to be interrupted and restarted as often as needed. Travel time-outs are also provided.

The programs contain several routines that may be useful in other applications. All of the routines in the program are written in functional blocks and may easily be extracted or modified for other uses. There are only two subroutines in the program: an alpha prompt and a version of the "ALMREL" program in the HP 82182A Time Module Owner's Manual. The use of the latter prevents data errors when alarms set relative to current clock time produce times greater than 24 hours.

The chime routine, "A", sets a periodic tone that may be used for signaling or as a keep-alert device on monotonous stretches of highway. To maintain a constant time lapse between alarms, the repeat interval is added to the last alarm time rather than the current clock time, eliminating the time taken by label searches and the alarm calculations themselves. Like all of the alarms in the program, the chime alarm is set by the program and stored in the alarm stack without an automatic reset. The chime may be set at program initialization or bypassed and set at a later time. It may also be changed or cancelled. When a change is desired, the existing alarm in the alarm stack is not cleared. The alarm catalog is displayed to enable the user to purge the obsolete alarm. When the alarm is activated, the routine first saves the X-, Y- and Z- registers and later restores them to prevent calculations in progress, if any, from being disturbed.

The global label "2" is called by a control alarm at the start of the program to keep track of 100's of hours if the stopwatch rolls over.

Routine "G" is a time-out feature that, when activated, subtracts hours and minutes of rest time from the total driving time. An alert is sounded approximately every ten minutes to remind the user that a time-out is in progress. It will remain set until cleared by restart - toggled by pressing the "G" key again. When restart is initiated, although the time-out alarm becomes due, it does not sound a tone or display a prompt. The alarm will not be reset until toggled again. Routine "J" prevents the alarm from appearing, but does not cause time to be accumulated again. This way a time-out can last overnight, for example, without hearing alarms every ten minutes.

Routine "a" resets the running stopwatch, presumed to have been interrupted, to where it would have been if not stopped. It uses clock data to perform the necessary calculation and then adds the amount of time used by the routine itself to restore the stopwatch. The necessary data are the time the trip clock started and the current clock time. The difference is calculated and multiplied by the number of days and the stopwatch is set MOD 100. This may be performed at any time after starting the program, although a trivial error of a few hundredths of a second may gradually be introduced. The calibration program "CAL", contains an identical routine to allow the two programs to be used independently.

Label "E" is a Distance = Rate * Time calculation that estimates the time of arrival and formats the output in clock time. A date display is also seen if the arrival date differs from the current date. This linear calculation will give credible results for typical interstate highway travel. Inputs can be in either miles or kilometers, as long as the units are consistent. The program treats them only as units and does not convert or name them, allowing easier input and faster calculations. (If you wish to add conversion routines to this program, two conversion factors will be useful: miles to kilometers, miles [ENTER^] 5 [LN] [*] (accurate to 2 decimal places) and liters to gallons, liters [ENTER^] 3.785 [/]).

"Calibration" is a routine that accurately calculates a vehicle's speed, in miles per hour, translated from engine revolutions in a given gear. To adapt the program to kilometers, a conversion factor must be added. Although mathematical routines exist that calculate road speed from the tire rolling diameter and transmission ratios, it is more realistic and practical to actually time the vehicle. Tachometers on vehicles with manual transmission are generally a more reliable and linear gauge of speed than are speedometers. Speedometers may also be calibrated by the program, but they commonly have non-linear errors. A correction factor calibrated at 50 miles per hour may be virtually useless at 35 miles per hour. Therefore, calculated factors should be trusted only in a range close to the calibration speed.

It should be noted that a typical reaction time, wherein a mile marker is seen and the [ENTER^] key is pressed to take a split is about 100 milliseconds. This means that the splits taken on a measured stretch of road should not vary more than a few hundredths of a second from mile to mile to be reliable. If this is of importance, a routine might be added to convert the different splits to decimal form and accumulate them in a statistical block of registers (ΣREG 32). An acceptable standard deviation could be chosen and tested.

Important data used by the "TRIP" program is retained in higher numbered data registers and will not be overwritten by accidentally storing up to eight splits too many. Normally, four or five miles would suffice for accurate calibration and the program only counts from Ø to 1Ø. Registers Ø5 through 1Ø serve primarily as an overflow buffer though they may also be used for additional splits such as on a ten-mile odometer calibration run in a time and distance rallye.

When the running program executes line 39, the calculator is put into stopwatch (SW) mode and the keyboard is redefined with stopwatch functions. Although a number of functions may be performed in this mode and the stopwatch may be controlled by the [R/S] key, the program is still running. When stopwatch mode is exited by manually pressing [<--] the program will resume normal execution and activate past due alarms, if any, that may have come due but could not be executed while the stopwatch was running. If the chime routine had been set to a short interval, such as five minutes, and had come due more than once, it would execute one time and become past due. It would not automatically be reset. This inconvenience is rarely encountered with control alarms.

"CAL" uses neither flags nor registers that are used by "TRIP", though it does provide data for tachometer RPM to speed conversion and its inverse. "CAL" resets the running stopwatch as if it had not been interrupted. It is not designed to correct the stopwatch if its use has bracketed the midnight hour.

OPERATING LIMITS AND WARNINGS

Estimated time of arrival cannot be calculated if the input odometer reading is equal to the start odometer reading.

Calibrated constants for any vehicle will vary slightly with environmental factors and tire inflation.

"CAL" is written to calibrate gauges in miles per hour. For kilometers, a conversion constant, 1.61, must be multiplied.

SAMPLE PROBLEM

(Some of the answers displayed in the following example are dependent upon the time of day that the example is run).

Use the "Automobile Trip Computer" on a vacation trip, driving 575 miles between Corvallis, Oregon and Palo Alto, California.

Use the "Calibration" program to calibrate the tachometer of your vehicle on a five-mile stretch of interstate highway. Use the calculated correction factor to be sure you are actually driving 55 miles per hour.

DISPLAY	INPUT	FUNCTION
Load the "TRIP" program		[GTO] ..
Load the "CAL" program		[XEQ] "SIZE" 032
Initialize the program		[XEQ] "TRIP"
ODOMETER?	Input odometer at start of trip 16000	[R/S]
COREX?	The correction factor is not presently known	[R/S]
DISTANCE?	575	[R/S]
CHIME/HMS?	Input a 5-minute interval to see how the chime works .05	[R/S]
You might like to drive "straight-through" and be there in ten hours.		
		[XEQ] "D"
ODOMETER?	16000	[R/S]
ARRIVE/HMS?	10	[R/S]
REQ SPD=57.5		
The result of the next operation will depend upon the actual time of day that you are running this example and the time elapsed since the beginning of the trip. This example assumes that the trip was started at 7 a.m.		
You have been driving 25 minutes, and wish to know your projected arrival time.		
		[XEQ] "E"
ODOMETER?	16023	[R/S]
5:25 PM		
If the projected time of arrival would be on the following day, the date would also be displayed, for example:		
5:25 AM 10/25		
If you input a fictitious distance, the calculator may assume you are travelling at an unusually high speed.		
You would like to stop at the Golden Arches.		
		[XEQ] "B"
TIME OUT		
TIME OUT		
Later, when you return to your vehicle:		
		[XEQ] "B"
RESTART		

DISPLAY	INPUT	FUNCTION

How long have you actually been on the road, barring time-outs?

[XEQ] "C"

DRIVE=( ) : ( )

The chime alarm is no longer useful to you and should be changed to a different interval.

[XEQ] "A"

RESET

CHIME/HMS?

Input any interval and press [R/S] or continue to the next operation. No keys need to be pressed to ignore the prompt. If the chime is not to be reset, you might execute "J" so that the alarm would be activated once, but not heard or reset.

Use the "Calibration" program.

[XEQ] "F"

Use of the function labelled "F" enables a return to the "TRIP" program. The remainder of the examples are in the "TRIP" program.

TACH RPM?	3000	[R/S]
GEAR? 3/4/5	4	[R/S]
00:00:00.00-->R00		[R/S] (when ready)

Press [ENTER^] at approximately the following intervals to simulate the passing of milepost in actual operation:

00:00:55.4	[ENTER^]
00:01:50.8	[ENTER^]
00:02:46.2	[ENTER^]
00:03:41.6	[ENTER^]
00:04:37.0	[ENTER^]
	[R/S]

The splits in actual use would be taken at mileposts. At least five are required for reliability.

Press [shift] [<--] to exit stopwatch mode and resume program execution.

If either of the control alarms, "chime" (CS), or "time-out" (T-) came due each would be activated before the program resumed execution.

Your output display should be similar to:

S/COREX=1.08X [R/S]

1K RPM, 4=21.7

Your vehicle travels at a rate of 21.7 miles per hour for every 1000 RPM's in fourth gear.

If you have executed the program properly, you will again be positioned to the "TRIP" program. Do not press [R/S].

DISPLAY	INPUT	FUNCTION
What should your speedometer indicate if you wish to drive exactly 55 miles per hour?		
55		[XEQ] "e"
INDIC=50.8		
When the speedometer indicates 50.8 miles per hour, you are actually travelling at 55 miles per hour.		
		[XEQ] "c"
GEAR	4	[R/S]
SPEED		[R/S]
RPM?	2700	[R/S]
SPEED=58.5		
The actual results may vary slightly due to the variation in the stored splits. If they are close, you have been running the program correctly.		

STATUS

SIZE	:	032
FIX	:	1, 2, 4
TOTAL PROGRAM BYTES	:	"CAL" 237
		"TRIP" 575

DATA REGISTERS

00-10	Split and overflow buffer
11	Save X
12	Save Y
13	Save Z
14	Counter, greater than 0 splits
15	Total time, calibration routine
16	Store gear, index for "CAL"
17	Scratch
18	Date start, MM.DDYYYY
19	Index, high split
20	Index, low split
21	SW n(100)

22	Start of trip, clock time
23	Chime, next occurrence: interval, HMS+
24	Chime interval
25	Odometer start
26	Planned distance
27	Time-out
28	Speedometer correction factor: default is 1.00
29	Gear 3, miles/hour/one RPM
30	Gear 4, miles/hour/one RPM
31	Gear 5, miles/hour/one RPM

FLAGS USED

00-07	May be used during the INSTAT command, if HP 82160A HP-IL Module is used without disturbing program operation	
03	Set	: Calibrating gear 3
	Clear	: All other states
04	Set	: Calibrating gear 4
	Clear	: All other states
05	Set	: Calibrating gear 5
	Clear	: All other states
07	Set	: Control alarms "CS" and "T-" disabled
	Clear	: Control alarms "CS" and "T-" active
08	Set	: Chime alarm is set
	Clear	: No chime alarm
09	Set	: Tachometer calibration
	Clear	: No calibrating
10	Set	: Time-out in progress
	Clear	: No time-out
21	Set	: Printer enabled
	Clear	: Printer disabled
22	Set	: Numeric input detected
	Clear	: No numeric input
26	Set	: Audio enabled
	Clear	: Audio disabled
55	Set	: Printer exists
	Clear	: Printer doesn't exist

FUNCTION LABELS

Label	Function
"a"	Reset stopwatch
"c"	Tachometer/gear/speed conversions
"d"	Indicated to actual speedometer reading
"e"	Actual to indicated speedometer reading
"A"	Chime, change and/or reset
"B"	Time-out/restart toggle
"C"	Total driving time
"D"	Required speed to distance in (n) HMS
"E"	Estimated clock time of arrival
"F"	Execute "Calibration" program and return
"J"	Cancel, disable control alarms

USER INSTRUCTIONS

SIZE: 032

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Loan Program "TRIP".		[GTO] ..	PACKING
2	Loan Program "CAL".		[GTO] ..	PACKING
3	Execute "TRIP" immediately prior to			
	the start of the trip.		[XEQ] "TRIP"	ODOMETER?
4	Input the current odometer reading. Any			
	units may be used as long as they are			
	consistent throughout the program.	reading	[R/S]	COREX?
5	Input the speedometer correction factor,			
	if known.	factor	[R/S]	DISTANCE?
6	Input the total distance to the			
	destination.	distance	[R/S]	CHIME/HMS?
7	If you want a periodic chime:	HH.MMSS	[R/S]	N.NN 0.00
	The following functions may be performed			
	at any time except as noted.			
8	Calculate the driving speed required			
	to reach the destination in n hours.		[XEQ] "D"	ODOMETER?
9	Input the odometer reading. If the			
	odometer passes through 1000,000 miles			
	and rolls over, you must input the actual			
	mileage it would read.	odometer	[R/S]	ARRIVE HMS?
10	Input the arrival time. The final output			
	can be miles or kilometers (or whatever)			
	depending on units that your odometer			
	reading is expressed in.	HH.MMSS	[R/S]	REQ SPD=( )
11	To reset the stopwatch: If the trip timer			
	maintained by the stopwatch must be used			
	temporarily for another purpose, reset			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	the stopwatch when ready, allowing for			
	the interruption.		[XEQ] "a"	RESET SW
12	To change or set the chime interval.		[XEQ] "A"	RESET CHIME/HMS?
	Input the chime repeat period.	HH.MMSS	[R/S]	
13	To temporarily interrupt the running			
	trip timer:		[XEQ] "B"	TIME OUT TIME OUT
14	To restart the trip timer:		[XEQ] "B"	RESTART
15	To calculate the total driving time:			
	Note that time-outs are not accounted for			
	until the restart routine accumulates			
	them.		[XEQ] "C"	DRIVE=nn:nn
16	To disable the periodic "TIME-OUT"			
	display and chime and to prevent its reset:		[XEQ] "J"	CANCEL
	Note that a time-out, if in progress, must			
	be restarted to update the time accumulator.			
17	Find the estimated time of arrival based			
	on the time you've been on the move and			
	your current distance from your			
	destination.		[XEQ] "E"	ODOMETER?
		distance	[R/S]	time and date
	The date display will not be seen if it			
	is the same as today's date.			
	The input distance may not equal the start			
	distance or a "DATA ERROR" will result.			
	USING THE CALIBRATION PROGRAM			
18a	If called as a subroutine from "TRIP":		[XEQ] "F"	TACH RPM?
18b	If run independently:		[XEQ] "TRIP"	TACH RPM?

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
19	Input engine RPM's: Skip this input if it			
	is not applicable, press [R/S] and go to			
	step 21.	RPM	[R/S]	GEAR? 3/4/5
20	Input the gear.	gear	[R/S]	00:00:00.00->R00
21	If "TACH RPM?" input bypassed, input your			
	speed. If both prompts are ignored, the			
	routine is repeated.	speed	[R/S]	00:00:00.00->R00
	OPERATING THE STOPWATCH			
22	To start the stopwatch:		[R/S]	
23	To store a split at a mile marker: Up to			
	11 splits may be stored. No programmable			
	test for the number of total splits is			
	possible in SW mode, so be cautious.		[ENTER^]	
	To exit SW mode and continue running			
	the program.		[shift] [<--]	
	Past due control alarms, if any, will run			
	immediately. Therefore, "TIME-OUT" and			
	chime HH.MMSS may be seen.			
	As an alternative, the calculator may be			
	turned off and on again and [R/S] pressed.		[R/S]	S/COREX= ( )X
			[R/S]	1K RPM, ( )=( )
	Display Signifies: 1000 RPM's in the input			
	gear = (nn.n) miles or kilometers			
	per hour.			
24	To calculate the speedometer reading			
	that corresponds to a speed:	speed	[XEQ] "e"	INDIC=( )
	To convert between RPM and speed:		[XEQ] "c"	GEAR?

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
25	Input the gear.	3, 4 or 5	[R/S]	SPEED?
26	Input the speed, if known. No input if			
	not known.	speed	[R/S]	RPM?
27	Input RPM, if known. No input if not known.	RPM's	[R/S]	SPEED=( ) or RPM=( )
28	To calculate actual speed for an observed			
	speedometer reading:	speed	[XEQ] "d"	ACTUAL=( )

PROGRAM LISTING

```
 01*LBL "CAL
"
 02 CF 03          Clear Flags used by pro-
 03 CF 04           gram
 04 SF 09          Set calibrate tachometer
 05 ΣREG 00         Flag
 06 CLΣ            Clear block of registers
 07 ΣREG 05         used to store splits
 08 CLΣ
 09 CLX
 10 "TACH RP       Place 0 in X for test input
M?"
 11 PROMPT         Prompt calibrate tach?
 12 X=0?            User to input RPM
 13 GTO 00         No input? Then next
 14 STO 16          question
                   Store calibration RPM
 15 "GEAR? 3       Input gear to be cali-
/4/5"               brated
 16 PROMPT
 17 SF IND X       Set gear flag, serves only
 18 26              as a visual aid
 19 +
 20 X<> 16         Set index register for
 21 STO IND         gear
16
 22 GTO 01         Go to start
 23*LBL 00
 24 CF 09
 25 CF 22          No tachometer input,
 26 "SPEED?"        clear Flag
 27 PROMPT         Try speedometer
 28 FC?C 22
 29 GTO "CAL       No input? Repeat
"                   sequence
 30*LBL 01
 31 RCLSW          Save stopwatch and
 32 TIME            clock
 33 STO 20
 34 X<>Y
 35 STO 19
 36 STOPSW         Initialize stopwatch
 37 CLX
 38 SETSW          Set to 0
 39 SW             Enter redefined keyboard
 40 CLD             mode: Stopwatch
 41 ALMNOW         Clear display on exit
 42 RUNSW
 43 TIME
 44 RCL 20         Calculate time taken by
 45 HMS-            stopwatch calibration
 46 RCL 19         And reset running stop-
                    watch to where it would
                    have been
 47 HMS+
 48 56 E-6         Allow for time taken by
 49 HMS-            reset code lines them-
 50 SETSW           selves (approx.)
 51 10             Reset stopwatch
 52 STO 20         Store index, register ad-
 53 9               dress of highest allowed
 54 STO 19          split
                   Store index, register ad-
 55 CLX             dress of lowest allowed
 56 STO 14          split
 57 STO 15         Set counter to 0
 58*LBL 02
 59 RCL IND        Calculate delta splits
20
 60 RCL IND
19
 61 HMS-
 62 X<=0?
 63 GTO 03         Test for end of splits
 64 RCL 15
 65 HMS+           Sum total time
 66 STO 15
 67 ISG 14         Increment counter
 68*LBL 03
 69 DSE 20
 70 DSE 19         Continue calculation of
 71 GTO 02          delta splits
 72 RCL 01
 73 RCL 00         Calculate last delta split
 74 HMS-
 75 RCL 15
 76 HMS+           Add to total time
 77 HR
 78 RCL 14
 79 1              Convert to decimal
 80 +
 81 /              Get number of good
 82 .01             splits
 83 HR             Divide into one minute,
 84 X<>Y            calibration routine
 85 /               written for 'miles', i.e.,
 86 STO 28          60 miles per hour is
 87 "S/COREX        basis
="                 Store correction factor
 88 FIX 2           for speedometer
 89 ARCL X         Output display
 90 "*X"
 91 AVIEW          Identify number as a
 92 FC?C 09         multiplication factor
 93 RTN
 94 FC? 55         If tach not wanted, return
                    or stop
```

PROGRAM LISTING

```
 95 STOP
 96 60
 97 *
 98 RCL IND
16
 99 1 E3
100 /
101 /
102 "1K RPM,
 "
103 FIX 0
104 RCL 16
105 26
106 -
107 ARCL X
108 "⊢="
109 FIX 1
110 CF IND X
111 ARCL Y
112 X<>Y
113 .1
114 %
115 STO IND
16
116 AVIEW
117 .END.
```

Stop if no printer, display register unchanged

Recall gear RPM

Convert to 1 RPM

Format output display: 1,000 RPM in (n) gear = (n)/hour

Get gear number into ALPHA

Clear gear Flag

Divide by 1,000

Store in gear (n) register

PROGRAM LISTING

```
01*LBL "TRI
P"
02 TONE 9          Signal timer start
03 RUNSW
04 CLX
05 SETSW           Set stopwatch to 0
06 TIME            Save clock start time
07 STO 22
08 STO 23          Clock saved for chime
09 -1 E2             HMS+
10 STO 21          First pass through con-
11 XEQ "2"           trol alarm '2' will set
12 DATE              to 0
13 STO 18          Initialize routine to total
14 ΣREG 25           100s of hours
15 CLΣ             Start date, used for re-
16 CF 07             setting stopwatch
17 CF 08           Clear block of registers
18 CF 10           Clear Flags that will ini-
19 XEQ 01            tially be tested by prgm.
20 STO 25
21 "COREX?"        Get ODOMETER?
22 1                 Prompt
23 PROMPT          Input factor, if any
24 STO 28            default=1, stored if
25 "DISTANC          no input
E?"                Input planned distance
26 CLX
27 PROMPT          Default is 0
28 STO 26
29 GTO 00          Jump over chime reset
30*LBL A             routine
31 SF 07           disable chime temporarily
32 RCL 23          Next occurrence
33 TIME
34 STO 23          Reset occurrence offset
35 FC?C 08           from present time
36 GTO 00          If chime not active, skip
37 HMS-              reset routine
38 1 E-4           Calculate time till alarm
39 HMS-              triggers
40 STO 17          Set back one second
41 T+X             Save clock adjustment
42 "RESET"         Advance clock
43 AVIEW           Use ALPHA display to
44 PSE               mask alarm activity
45 PSE             Be sure that triggered
46 RCL 17            chime occurs during
47 CHS               space
48 T+X             Set clock back to present
49*LBL 00          Initial entry point to
                     chime set routine,
                     bypass reset
50 CF 07           Flag 07 used to disable
51 CLX               chime
52 "CHIME/H        Default is 0
MS?"               Input periodic interval
53 PROMPT
54 STO 24
55 X=0?            No input?
56 RTN
57 CF 26           Silence tones on first
58*LBL "CS"          pass through chime
59 FS? 07          Label for control alarm
60 RTN               Disable?
61 SF 08           Return if called by reset,
62 TONE 9            else stop
63 TONE 8          Chime active Flag, used
64 SF 26             to test reset
65 STO 11          Chime
66 RDN             Re-enable audio for future
67 STO 12            occurrences
68 RDN             Save stack in case a calcu-
69 STO 13            lation in progress is
70 "↑↑CS"            interrupted
71 RCL 24          X, Y, and Z are saved
72 RCL 23          Set up control alarm
73 XEQ 04          Get interval
74 XYZALM          Get clock time of last
75 STO 23            occurrence and execute
76 RCL 13            ALMREL routine
77 RCL 12          Store next occurrence
78 RCL 11          Restore Z
79 RTN             Restore Y
80*LBL D           Restore X
81 XEQ 01          Required speed routine
82 RCL 25          Get ODOMETER?
83 RCL 26            input prompt
84 +               Odometer start
85 X<>Y            Planned distance
86 -               Added
87 "ARRIVE         Less the distance covered
HMS?"                to this point
88 PROMPT          Input desired travel time,
89 HR                n hours
90 /
91 "REQ SPD        Calculate rate of travel
="                 Output display
92 GTO 06          Output routine
93*LBL E           Estimated time of arrival
94 XEQ 01          ODOMETER? Prompt
95 RCL 25          Odometer start
96 -               Distance covered
97 ENTER↑          Save in stack
```

PROGRAM LISTING

```
 98 ENTER↑
 99 RCLSW          Get trip elapsed time and
100 RCL 21         adjustments for 100s of
101 HMS+           hours and time-outs
                   totalled
102 HR             Calculate rate
103 /
104 RCL 26         Planned distance
105 R↑
106 -
107 X<>Y
108 /
109 HMS            Add projected time to
110 TIME           clock time
111 XEQ 04
112 FIX 2          Execute routine to get
113 CLA            time and date in X and
                   Y
114 ATIME          Initialize ALPHA
115 "⊢ "           Format time display
116 CLX
117 DATE           Test if today's date
118 X<>Y
119 X≠Y?
120 ADATE          If different, append to
121 AVIEW          ALPHA
122 RTN
123◆LBL 01         Input prompt subroutine
124 CLX            Set to 0 for testing input
125 "ODOMETE
R?"
126 PROMPT
127 RTN
128◆LBL B          Time-out toggle
129 RCLSW          Trip time to date
130 RCL 21
131 HMS+           Plus 100s of hours and
132 FC?C 10        previous adjustments
                   Time-out mode?
133 GTO 02
134 RCL 27
135 HMS-           If Flag 10 clear, then ini-
136 RCL 21         tiate a time-out, else
137 X<>Y           adjust accumulator
                   register
138 HMS-
139 STO 21
140 "RESTART       Result of action
"
141 AVIEW
142 RTN
143◆LBL 02         Initiate a time-out
144 SF 10          Set time-out Flag
145 STO 27         Adjusted trip time
146◆LBL "T-"       Label for control alarm
147 FC? 10         Time-out period ended?
148 RTN
149 "↑↑T-"         Control alarm
150 TIME
151 .1             Ten minute interval--
152 XEQ 04         reminder
153 XYZALM         Set up alarm parameters
154 FS? 07         in stack
                   If alarm 'silenced' quit
155 RTN            here, if triggered by
156 XEQ 03         chime reset, then reset
157◆LBL 03         alarm to next occurrence
158 "TIME OU       omit display this time.
T"                 Repeat display and tone
                   twice.
159 AVIEW          Reminder routine
160 TONE 5
161 RTN
162◆LBL "2"        Add 100 hours to accumu-
163 "↑↑2"          lator when stopwatch
164 1 E2           rolls over
165 ST+ 21         Accumulator register
166 TIME
167 XEQ 04
168 XYZALM         Next hundred
169 RTN
170◆LBL 04         ALMREL type routine,
171 HMS+           similar to Timer Module
172 ENTER↑         Owner's Manual, here
173 ENTER↑         without I/O halts
174 24             Number of days
175 /
176 INT
177 DATE
178 X<>Y
179 DATE+
180 LASTX
181 24
182 *
183 ST- Z
184 CLX
185 STO T          Set up parameters in
186 RDN            alarm, no reset, time
187 X<>Y           less than 24 hours
188 RTN
189◆LBL a
190 RUNSW          Reset stopwatch
191 DATE
192 RCL 18
193 DDAYS          Date stopwatch started
194 24             running
195 *              n days running
196 TIME           n hours
```

PROGRAM LISTING

```
197 RCL 22
198 7 E-5          Time stopwatch started
                     on first day
199 HMS-           Approx. amount of time
200 HMS-             taken to execute this
201 HMS+             block of code
202 1 E2           Subtracted from total
203 MOD            Total hours since start
204 SETSW          MOD 100 to avoid data
                     error
205 "RESET S       Confirm
W"
206 AVIEW
207 RTN
208*LBL C
209 RCLSW          Total driving time
210 RCL 21         Trip timer
211 HMS+           Accumulator register
212 "DRIVE="
213 FIX 2
214 ATIME24
215 AVIEW          Format output
216 RTN
217*LBL c
218 "GEAR?"        Input prompt
219 PROMPT
220 26             Index number, location
                     of gear data
221 +
222 "SPEED?"       Calculate RPM if speed
                     known
223 0              Set to 0 for input test
224 PROMPT
225 X≠0?           No input, try next
226 GTO 05         If input speed, go to
                     calculate routine
227 "RPM?"         Calculate speed from
                     known RPM?
228 PROMPT
229 RCL IND        Recall gear data
Z
230 *
231 "SPEED="
232 GTO 06         Go to output routine
233*LBL 05
234 RCL IND        Gear data
Z
235 /
236 "RPM="         Output
237 GTO 06
238*LBL d
239 "ACTUAL=       Observed reading input,
"                    calculate actual speed
240 RCL 28         Constant was stored by
241 *                'CAL' or input at start
242 GTO 06         Go to output
243*LBL e
244 "INDIC="       Speedometer display
245 RCL 28
246 /              Correction factor
247*LBL 06
248 FIX 1          Display one significant
                     digit
249 ARCL X
250 AVIEW
251 RTN
252*LBL J
253 SF 07          Set Flag to disable chime
                     at entry point and silence
254 "CANCEL"       Time-out, if any, after
255 AVIEW            resetting
256 RTN
257*LBL F
258 FC? 55         Call global subroutine and
259 CF 21            return here
260 XEQ "CAL       Printer?
"
261 .END.
```

PROGRAM DESCRIPTION

FOUR CHANNEL CONTROLLER

"Four Channel Controller" provides a means of maintaining up to four user-written controller programs or "channels". Each channel may have its own six character identifier; time, data and note files; simple message alarms; and user defined control alarms. Alarms and callable subprograms may be added to or deleted from memory at any time. The data associated with a channel may be recalled and/or printed when desired. The program is capable of synchronizing the HP 82182A Time Module's stopwatch with any channel time and of keeping track of the next available data register into which splits may be stored. Nine free registers are provided by the program for user-defined channel use.

"4CON" is the initializing portion of the program. With it, all four channels are initialized by storing the current clock time in data registers Ø1 through Ø4. Flags Ø1 through Ø4 are not tested by the program, but are provided as indicators of which channel is currently being accessed. The states of these flags are preserved in register Ø6 and, therefore, will not be adversely affected by the HP 82160A HP-IL Module function INSTAT or any other flag-altering function.

All of the following routines may be given global labels, to follow or replace their respective local alpha labels, to provide global access, i.e., from control programs.

The subroutine labelled "b" initiates the channel whose number is taken from the X-register.

The subroutine labelled "c" adds alarms to any running channel without reinitializing it. The channel number is taken from the X-register.

The subroutine labelled "e" clears the registers used by the "DATA" program.

The subroutine labelled "C" recalls and prints all of the stored data pertinent to the channel whose number is taken from the X-registers.

The subroutine labelled "E" syncronizes the HP 82182A Time Module's stopwatch with any channel. The actual time taken to run the routine is considered in the calculation. The routine displays the first empty data record. When using the stopwatch, manually setting the stopwatch's split-register pointer to this value will avoid overwriting important data registers.

The subroutine labelled "DATA" allows the storage of time related data in each channel. If the calculator does not have sufficient room to store the next group of data, "NO ROOM" will be displayed and the program will halt.

The user may write programs that access these blocks of registers in the same fashion that "DATA" does. The data is stored in five-register records in which the lowest numbered record contains the channel number. The second register contains the time from initialization at which the data set was recorded. The third and fourth registers contain a message of up to twelve characters. The fifth register contains any pertinent numeric data.

This program is intended to be compatible with all HP-IL controllable devices. All of the routines will allow at least four additional levels of subroutine calls. The display mode used by routine labelled "C" is not controlled by the program and is therefore up to the discretion of the user.

OPERATING LIMITS AND WARNINGS

Obsolete reset alarms must be cleared manually via the "ALMCAT" function of the HP 82182A Time Module.

Space must be available in memory for storage of alarms. Refer to the HP 82182A Time Module Owner's Manual for details of the memory requirements of the various alarms.

Failure to adjust the stopwatch pointer when in stopwatch mode may result in the destruction of valuable data.

Default conditions (such as the states of flags 17 and 21) should not be assumed by any user-written programs. User defined program interrupts -- control alarms -- should preserve the stack whenever possible.

See step 10 of the User Instructions.

SAMPLE PROBLEM

1. A medical office administers glucose tolerance tests to many of their patients. The tests are from four to six hours in length and are used to test for Diabetes and Hypoglycemia. Blood samples must be taken at exact intervals and the measurements must be recorded. Use the program to accommodate this need.

2. Line voltage to a microcomputer must be measured periodically to test the effectiveness of a voltage regulator. Use channel 3 to sample and print the voltage at five minute intervals with the HP 3468A Voltmeter.

3. A Photographer needs to monitor three activities simultaneously including the time during which a model is on a particular assignment, how long a batch of prints is being rinsed in the darkroom and when to leave for an appointment.

DISPLAY	INPUT	FUNCTION
Load the "4CON" program. Prior to running the example problems, be sure that there are at least eight unused program registers in your calculator. These registers will be used by the alarm stack. The minimum SIZE required by the first problem is 042.		
		[XEQ] "4CON"
DATA X?		
In practice, at least seven samples would be drawn for a six hour test. However, for this example, we will take only three readings. (The third reading will be used in another example).		
	3	[R/S]
NAME	BLTEST	[R/S]
ALARM/HMS?		
The first test is run at .5 hour. The remaining tests are run at 1 hour intervals.		
	.3	[R/S]
RESET?	1	[R/S]
PRGM?		
We will not be controlling a program.		
		[R/S]
BLTEST		
If you use ALMCAT, you can confirm that the alarm has been set with a one hour reset. Use the "DATA" routine to store the patient's initial blood sugar level.		
		[XEQ] "DATA"
CHANNEL?	1	[R/S]
NOTES?	FASTING BL/S	[R/S]
DATA?	80	[R/S]
80.00		
Let's assume that the first alarm has been activated. The next measurement is 162 MG%.		
		[XEQ] "DATA"
CHANNEL?	1	[R/S]
NOTES?		
The patient's name could be input here to prevent confusion if there is more than one channel being run by "4CON".		
	MOE SUGAR	[R/S]
DATA?	162	[R/S]
162.00		
Recall the data. The outputs marked ** are approximate. The displayed times depend upon how long it took to run the sample problem. If your displays are similar in appearance to those given here, you are running the program correctly.		

DISPLAY	INPUT	FUNCTION
	1	[XEQ] "C"
BLTEST		
00:01:45.41 **		[R/S]*
FASTING BL/S		[R/S]*
80.0		[R/S]*
00:31:50.41 **		[R/S]*
MOE SUGAR		[R/S]*
162.000000		[R/S]*
END		

* [R/S] is not necessary if printer exists.

2. Before running this example, you should purge the alarms set from the previous problem.

	2	[XEQ] "b"
NAME?	VRTEST	[R/S]
ALARM/HMS?	.3	[R/S]
RESET?	.05	[R/S]
PRGM?	V	[R/S]
VRTEST		

The first alarm will go off 30 minutes after the timer was started and will repeat every five minutes thereafter until it is manually purged. Note that the sample problem could have been written to store certain values in the unused registers, R15 - R22, for comparisons. Additionally, the ALPHA strings used to send instructions to the HP3468A Voltmeter could have been prestored by an initialization routine to speed program execution. They are normally programmed on the same line but are separated here for clarity. The program could be designed to test for a minimum or maximum voltage and either sound an alarm or shut down the entire system via a (hypothetical) relay interface.

```
01*LBL "V"
02 AUTOIO
03 2              The voltmeter is the 2nd device in the loop
04 SELECT
05 REMOTE
06 "F2"           AC Volts function is selected
07 OUTA
08 "R4"           300 Volt range is selected
09 OUTA
10 "T2"           Single trigger mode causes a single reading to be taken
11 ATIME
12 OUTA
13 IND            Voltage reading is sent to the X register
14 X<>Y
15 FIX 4          Swap with time
16 CLA
17 ATIME          Format and print time
18 PRA
19 X<>Y
20 FIX 2
21 PRX            Print voltage
22 END
```

Whether or not the example program was keyed in, use the ALMCAT function to confirm that the control alarm was set and purge the alarm.

[XEQ] "ALMCAT"

Press [R/S] when the display "↑↑V" is seen. Press [R] to see the 00:05:00 reset. Press [T] to see the time 30 minutes from the time that the alarm was set. Press [shift] [c] to purge the alarm.

DISPLAY	INPUT	FUNCTION
3. Sophisticated equipment is not required to use the "Four Channel Controller". It may be used to monitor various activities as well as timekeeping chores.		
Initialize channel 3.		
	3	[XEQ] "b"
NAME?	MODEL	[R/S]
ALARM/HMS?	no input	
We are giving the channel an identification that will be used by the "DATA" function.		
Initialize channel 4.		
	4	[XEQ] "b"
NAME?	PRINTS	[R/S]
ALARM/HMS?	.3	[R/S]
RESET?	0 or no input	[R/S]
PRGM?		[R/S]
		[R/S]
The prints must be delivered in one hour. Add an alarm to channel 4 to activate in one hour.		
ALARM/HMS?	1	[R/S]
RESET?	.05	[R/S]
PRGM?		[R/S]
Note that by pressing [R/S] a series of alarms could be placed on the same channel. To give each alarm a different "message", use routine "c". Input the channel number and execute "c". Respond to the prompt "NAME?" with a 6-character code or message. Although this also has the effect of changing the channel identification, each message alarm will have its own unique message. Channel 3 retains its original identification, "MODEL".		
Your model has arrived. Use the "DATA" function to keep track of the amount of time that the model works in your studio.		
		[XEQ] "DATA"
CHANNEL?	3	[R/S]
NOTES?	(model name)	[R/S]
DATA?	25	[R/S]
Recall the data	3	[XEQ] "C"

When your model has finished, you might use the "DATA" function to record the finish time and other important information.

STATUS

SIZE : 042

TOTAL PROGRAM BYTES : 470

DATA REGISTERS

00	Scratch register for User defined program
01	Start time of channel 1
02	Start time of channel 2
03	Start time of channel 3
04	Start time of channel 4
05	Storage index - returned to R07 after block clear - used by routine "e"
06	Channel flag - 1, 2, 3 or 4
07	"DATA" storage index
08	Recall index used by routine "C" - time parameter for "XYZALM"
09	Reset parameter for "XYZALM"
10	Channel name index
11	Start date of channel 1
12	Start date of channel 2
13	Start date of channel 3
14	Start date of channel 4
15-22	Scratch registers for user defined programs
23	Name of channel 1
24	Name of channel 2
25	Name of channel 3
26	Name of channel 4
27	Start of first "DATA" record: channel number (1-4)
28	Elapsed time
29-30	Message
31	Data or scratch
32->	Additional records

FLAGS USED

00	Set	:	Return to "DATA" from routine 08
	Clear	:	No return
01-04	Set	:	Visual identification of channel 1
	Clear	:	Not channel 1
21	Set	:	Printer enabled
	Clear	:	Printer disabled
23	Set	:	ALPHA input detected
	Clear	:	No ALPHA input
25	Set	:	No error detected
	Clear	:	Error detected

FUNCTION LABELS

Label	Function
"b"	Starts the time for the channel indicated by the X-register
"c"	Add alarms to the channel indicated by the X-register without resetting the timer to zero
"e"	Clear the block of registers dedicated to the "DATA" routine
"C"	Recall, view or print "DATA" records for the channel
"E"	Synchronize the stopwatch with the channel
"DATA"	Store data interactively

USER INSTRUCTIONS

SIZE: 042

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load program "4CON".		[GTO] ..	PACKING
2	Initialize "4CON".		[XEQ]"4CON"	DATA X?
3	Input the number of data points to be			
	stored during program operation.	data	[R/S]	NAME?
4	Input the name or code that identifies			
	the channel (6 character maximum).	name	[R/S]	ALARM/HMS?
5	Input the time interval of the first			
	alarm on channel 1. (Can be either a			
	control alarm or message alarm where the			
	message is the channel name).	HH.MMSS	[R/S]	RESET
6	Input the RESET interval.	HH.MMSS	[R/S]	PRGM?
7	If the alarm is a control alarm, input			
	the name of the peripheral function			
	or program name that is to be activated.	prgm	[R/S]	
	If the alarm is a message alarm.	no input	[R/S]	channel name
8	To set further alarms on the same channel.		[R/S]	ALARM/HMS?
	Return to step 5.			
9	To start any channel timer 1 - 4:	ch. (4)	[shift][b]	NAME?
	Return to step 4.			
10	To add alarms to a channel that has already			
	been initialized.	ch. (n)	[shift][c]	NAME?
	The channel name may be changed, if			
	desired, by keying in a different name.			
	Note: When adding intervals to timers			
	that are currently running, it is			
	important to remember that they are time			
	offsets from the original start time.			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	To avoid past-due alarms, input the initial			
	alarm HH.MMSS so that it will occur at a			
	future time and input the rest HH.MMSS			
	equivalent to the required cycle. For			
	example, if the timer has been running			
	for (m) hours and the alarm must activate			
	every (n) hours, the first interval must			
	be (m) + (n) hours. The reset will be			
	(n) hours. Input channel identification			
	if change desired.	name	[R/S]	
11	To clear an alarm that has been set by			
	the program.		[XEQ]"ALMCAT"	
	Alarms must be cleared manually in			
	ALMCAT mode.			
12	To store time-related data interactively.		[XEQ]"DATA"	CHANNELS?
	Input the channel number 1 - 4.	ch. (n)	[R/S]	NOTES?
	Input a descriptive alpha note, up to			
	12 characters in length.	notes	[R/S]	DATA?
	Input any numeric data or measurement			
	taken at the time "split".	data	[R/S]	
13	To view or print data	ch. (n)	[C]	(channel name)
	View elapsed time		[R/S]	nn:nn:nn.nn
	View notes		[R/S]	(notes)
	View data		[R/S]	data (appears in alpha and x)
14	To continue sequentially:		[R/S]	
	It is not necessary to press [R/S] if a			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	printer is in the system. Display will			
	continue until no more data is found filed			
	under the channel number and the display			
	"END" is seen.			END
15	To clear the entire data file and reset			
	the index to the first record.		[shift][e]	Ø.ØØØØ
16	To display a running time for any channel,			
	user must manually go into SW mode to see			
	running time.	ch. (n)	[E]	SYNC( )R( )
	To avoid overwriting valuable data by			
	accidentally storing splits (pressing			
	the ENTER key) while in the SW mode,			
	manually set the SW pointer to the value			
	of "R" displayed in the alpha register.		[XEQ]"SW"	nn:nn:nn.n
	Note: If splits are desired and stored at			
	the pointer address displayed in the alpha			
	registers, those splits cannot be read			
	by function [c] which expects data to			
	be formatted by label "DATA".			
17	To determine the constant to be inserted			
	in place of line 215 in the "4CON" program			
	a temporary change must be made in the			
	program itself.		[GTO] .219	
			[PRGM]	
			[XEQ]"STOPSW"	
			[R/S]	
	After keying in the additional two lines			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	of code "STOPSW" and "STOP".		[XEQ]"PACK"	
		0	[XEQ]"SETSW"	
		1	[E]	
	Place the calculator in SW mode [XEQ]"SW"			
	and read the display. The displayed value			
	will replace line 215. Be sure that the			
	procedure has been performed more than			
	one time for accuracy and that the stop-			
	watch was initially cleared each time.			

PROGRAM LISTING

```
 01◆LBL "4CO
N"
 02 TIME
 03 STO 01
 04 STO 02
 05 STO 03
 06 STO 04
 07 DATE
 08 STO 11
 09 STO 12
 10 STO 13
 11 STO 14
 12 CF 02
 13 CF 03
 14 SF 27
 15 4
 16 STO 06
 17 "DATA X?
"
 18 PROMPT
 19 5
 20 *
 21 26
 22 +
 23 .1
 24 %
 25 26
 26 +
 27 STO 05
 28 CLA
 29 ASTO 23
 30 ASTO 24
 31 ASTO 25
 32 ASTO 26
 33 XEQ e
 34 1
 35 GTO c
 36◆LBL b
 37 TIME
 38 STO IND
Y
 39 10
 40 RCL Z
 41 +
 42 DATE
 43 STO IND
Y
 44 LASTX
 45◆LBL c
 46 XEQ 09
 47 22
```

Store clock time to initialize all 4 channels. Channel numbers and register numbers correspond.

Store starting date for all channels. Will be used by Label 08 to calculate total elapsed time and add n(days) * 24 to total

Clear channel flags

Store Flag 04 in flag register (06) to overwrite possible non-existent flag

Input number of data points

Data points *5+26 additional registers

Calculate index for 'DATA' routine

÷1000

Save index in R05 in case it will be used again

Channel 1 go to alarm input routine

Start channel (n) whose number is in X register

Corresponding register

Calculate register address of date, ch.(n) in Z register

Save start date

Get flag and channel number back

Add alarms to a running timer channel

```
 48 STO 10
 49 +
 50 CF 23
 51 "NAME? "
 52 ARCL IND
X
 53 AON
 54 STOP
 55 AOFF
 56 FS?C 23
 57 ASTO IND
X
 58◆LBL 01
 59 "ALARM/H
MS?"
 60 CLX
 61 PROMPT
 62 X=0?
 63 GTO 04
 64 STO 08
 65 "RESET?"
 66 CLX
 67 PROMPT
 68 STO 09
 69 CF 23
 70 "PRGM?"
 71 AON
 72 STOP
 73 AOFF
 74 FC?C 23
 75 GTO 02
 76 ASTO X
 77 ASHF
 78 ASTO Y
 79 "↑↑"
 80 ARCL X
 81 ARCL Y
 82 GTO 03
 83◆LBL 02
 84 CLA
 85 RCL 06
 86 RCL 10
 87 +
 88 ARCL IND
X
 89◆LBL 03
 90 DATE
 91 RCL IND
06
 92 RCL 08
 93 HMS+
```

Input channel ALPHA i.d.

Store in register channel (n)+22

Input alarm interval

Default=0

No input?
Exit

Save temporarily

Default=0

Save reset

Test input

Control alarm?

If not a control alarm, then get the name of the channel

Save the control alarm name in stack and put control characters in ALPHA, then append name

Go to time offset routine

Get channel name for default message

Time offset routine

Clock time

+ alarm interval (time offset)

PROGRAM LISTING

```
 94 ENTER↑
 95 ENTER↑
 96 24                Number of hours
 97 /
 98 INT               and number of days
 99 DATE
100 X<>Y
101 DATE+
102 LASTX
103 24
104 *
105 ST- Z
106 CLX
107 RCL 09           R09=reset
108 X<> Z
109 XYZALM           Place XYZALM parameters in stack in correct sequence
110◆LBL 04
111 RCL 06
112 RCL 10           Display routine
113 +                Channel number +22= ALPHA name
114 RCL IND
X
115 RTN
116 RCL 06           Continue storing alarms on this channel
117 GTO 01           Recall data records
118◆LBL C
119 RCL 10           Get name of channel (file)
120 X<>Y
121 +
122 VIEW IND          Display or print
 X
123 LASTX            Channel number
124 XEQ 09           Clear old flag, display channel on annunciator
125 RCL 10           R10=22
126 RCL 07           R07=index
127 FRC              .eee portion of index (highest register)
128 +
129 STO 08           Initialize R08 as recall index
130 FIX 6
131◆LBL 05           Get every 5th register
132 5
133 ST+ 08           Enable printer or force stop on output display AVIEW
134◆LBL 06
135 SF 21            Error ignore
136 SF 25            No more data or no data found
137 "END"
138 RCL IND          Get first register of 'DATA' record
08
139 RCL 06           Compare to channel number
140 FC?C 25          If a nonexistent register tried, end of file
141 PROMPT
142 X≠Y?             Compare to channel
143 GTO 05           Try again
144 ISG 08           Format elpased time display
145 RCL IND          Get 2nd register of record
08
146 ADV              2nd register=elapsed time
147 CLA
148 ATIME24
149 AVIEW            ALPHA null string for comparison
150 CLA
151 ASTO X
152 ISG 08
153 RCL IND          Get first 6 characters of ALPHA note
08
154 ISG 08           If no message, then skip blank display
155 X=Y?
156 GTO 07
157 ARCL X           Else, get entire string
158 ARCL IND
 08
159 AVIEW
160◆LBL 07           Continue display
161 ISG 08
162 RCL IND
08                   Get 5th register of record
163 ISG 08
164 CLA
165 ARCL X
166 X≠0?             Display if not 0, place in ALPHA register to maintain uniform print format
167 AVIEW
168 GTO 06
169◆LBL "DAT
A"                   Store data interactively
170 TIME             Get time immediately
171 TONE 9           Feedback
172 "NO ROOM
"                    End of file
173 ISG 07           Increment storage index
174 "CHANNEL
?"                   Which timer?
175 PROMPT
176 XEQ 09           Go to flag update routine
177 STO IND
07
178 SF 00            Flag 00 will enable subroutine return
179 X<>Y
180 XEQ 08           Gosub elapsed time calculation
181 ISG 07
182 STO IND
07                   Store ET in 2nd register
183 ISG 07
```

PROGRAM LISTING

```
184 "NOTES?"
185 CF 23
186 AON
187 STOP
188 AOFF
189 FC?C 23
190 CLA
191 ASTO IND
 07
192 ASHF
193 ISG 07
194 ASTO IND
 07
195 "DATA?"
196 CLX
197 PROMPT
198 ISG 07
199 STO IND
07
200 RTN
201◆LBL E
202 CF 00
203 XEQ 09
204 RUNSW
205 TIME
206◆LBL 08
207 RCL IND
Y
208 HMS-
209 RCL 06
210 10
211 +
212 RCL IND
X
213 DATE
214 DDAYS
215 24
216 *
217 RCL Z
218 HMS+
219 FS?C 00
220 RTN
221 158 E-6
222 HMS+
223 1 E2
224 MOD
225 SETSW
226 FIX 0
227 "SYNC "
228 ARCL 06
229 "⊢  R"
```

Input up to 12 characters to name or describe data

Test for input

If no input, routine would store 'NOTES?' string, so CLA

Save note

Any numeric data or measurement?
Default=0

Synchronize stopwatch to channel (n) whose number is in X
Clear subroutine flag
Update channel flag Set
Set stopwatch on the fly
Routine 'written into' sync routine to avoid label search time on first execution if it were a subroutine
Search time would make constant on line 216 inaccurate
Get date timer channel started

Current date

Positive difference
Max. number of hours

Plus time difference

=total number of hours

If called by 'DATA' routine, return
Time taken by lines 199 through 219 must be added to elapsed time
Avoid data error when stopwatch is set

Format prompt

Get 'DATA' index and truncate to calculate last used register

```
230 RCL 07
231 INT
232 1
233 +
234 ARCL X
235 AVIEW
236 RTN
237◆LBL 09
238 CF IND 0
6
239 STO 06
240 SF IND X
241 RTN
242◆LBL e
243 RCL 05
244 STO 07
245 ISG X
246 0
247◆LBL 10
248 STO IND
Y
249 ISG Y
250 GTO 10
251 END
```

Output: Synchronize channel (n); set pointer to R(nn)

5 byte sequence called 5 times saves 8 bytes
Clear previous flag (at initialization set to 04 to avoid possible error if nonexistent or ALPHA data)
Set current channel annunciator. These flags are not tested, only displayed
Clear block of registers allocated to data. Usefuleas local ALPHA label only if block will be used, dumped to storage medium and reinitialized. Original index restored on completion
0 registers bbb-eee

PROGRAM DESCRIPTION

LOGBOOK

"Logbook" uses the HP 82180A Extended Functions/Memory Module to store the name and times worked for accounts that are billed at an hourly rate. Additionally, purchase order numbers, billing codes and remarks can be stored without regard to their length or format. The starting time and date for each account is saved in an ASCII file and requires no data register to maintain. Files can be printed or viewed at any time and total time worked can be updated on a daily basis. Access to stored information is by account name or the first few letters of the name. The program is useful in professional offices and any application where time must be stored in a flexible format.

"Logbook" consists of four main parts:

1. Create File
2. Start Clock
3. Stop Clock
4. Output

"$TIME" initializes an ASCII file by prompting for the file name, number of accounts and number of days to be recorded. Sixty characters are allowed for the account name and description and 9 characters each for time and date. Including file overhead, the calculation of approximate file size is: (number of accounts * 60) + (number of accounts * (number days + 1) * 20) + 30. A scratchpad space is created in each file to store the starting time of each job. This means that the user need not be concerned with data being erased by other programs or constant updating of magnetic cards. Only the file name and the first few characters of the account name are required to access the time data.

Label "B" starts the timer for each account. The start time (clock) and date is immediately stored in main memory scratch registers and the user is prompted for the file name and account name. Only the first few unique characters of the account name need to be input to locate the account in the ASCII file, a process considerably faster than label search or data recall and comparisons.

Label "b" stops the time for the named account and displays the total elapsed time for the current day. The time is displayed in HH:MM:SS.hh format in the alpha registers and HH.MMSShh format in the Y register and decimal format in the X register for easy time/rate calculations. The elapsed time is inserted in the ASCII file at the bottom of the list.

Label "C" prints or views the account name and descriptive data and chronologically outputs each date and time worked for the account. The data is formatted in the alpha register and in HH.MMSShh and decimal format in the Y and X registers. At the end of the list the total time worked is output.

OPERATING LIMITS AND WARNINGS

Character number 95 "_" should not be used in any input alpha string as the program interprets that character as an end of account record delineator. Using substantially more descriptive characters than 60 may cause the file to reach the end prematurely. If more characters are required, then change line 05 of the program accordingly.

SAMPLE PROBLEM

An accountant will be working in the offices of two of his clients on three successive days. As time worked is charged at an hourly rate, a convenient way of logging the time worked including travel time would be helpful.

DISPLAY	INPUT	FUNCTION
Load the "$TIME" program		
Initialize the program		[XEQ] "$TIME"
N ACCTS?	2	[R/S]
N DAYS?	3	[R/S]
FL NAME?	$TDEMO	[R/S]
NAME?	NICHOLAS NABIL	[R/S]
CODE?	JOB NO 97321-4.5	[R/S]
REMARKS?	BUILDING WRECKERS	[R/S]
NAME?	BENJAMIN ELIAS	[R/S]
CODE?	JOB NO 97330-2.5	[R/S]
REMARKS?	PLUMBING CONTRACTORS	[R/S]

The ASCII file has been created. If the file were to be printed out at this point, it would look like this:

```
NICHOLAS NABIL
JOB NO 97321-4.5
BUILDING WRECKERS
_
0.000000
0.000000
BENJAMIN ELIAS
JOB NO 97330-2.5
PLUMBING CONTRACTORS
_
0.000000
0.000000
```

The 0.000000 data represents the work spaces in the file. These spaces are overwritten by the start date and time whenever function [B] is used.

The following routine will print or view the entire ASCII file.

```
01   LBL "PA"
02   SF 25
03   SF 21
04   CLX
05   SEEKPTA
06   LBL 01
07   GETREC
08   FS? 25
09   AVIEW
10   FS? 25
11   GTO 01
12   END
```

DISPLAY	INPUT	FUNCTION

To provide a reasonable time for this example, use the T+X function of the the HP 82182A Time module to set your clock back four hours.

	-4	[XEQ] "T+X"

Start the clock running for the first client to keep tract of travel time which is billed at a different rate than office time.

		[XEQ] "B"
FILE NAME?		

If the ASCII file is your working file, no input is necessary. For this example, we will assume that the file "$TDEMO" is not yet the working file.

	"$TDEMO"	[R/S]
ACCT?	NICHOLAS	[R/S]

The entire account or client name need not be input each time the file is accessed. The file will be positioned to the first occurrence of the string "NICHOLAS".

START

The display confirms that the clock has started running.

Start the clock for the second client.

		[XEQ] "B"
FL NAME?		[R/S]
ACCT?	BEN	[R/S]

Advance your clock by one hour. Note that this is not part of the program. It is performed to give a typical output.

	1	[XEQ] "T+X"

Stop the clock for one client.

		[XEQ] "b"
FL NAME?		[R/S]
ACCT?	NICHOLAS	[R/S]
01:05:00.00?		

If your display has a similar time-formatted output then you are running the example correctly. Press [<--] to see the elapsed time in decimal form in the X-register. This allows an immediate rate calculation if desired. Press [X<>Y] to see the elapsed time in HH.MMSShh format where it may be used for HMS+ addition if needed.

Start the clock for the first client to reflect the time spent working in the clients office.

		[XEQ] "B"
FL NAME?		[R/S]
ACCT?	NICHOLAS	[R/S]
START		

Now your clock can be advanced to the correct time. Note that this is not necessary in order to run this program. It simply gives a more realistic appearance to the output

	3	[XEQ] "T+X"

Stop the clock for this account. The very first can be called by just one initial if desired. Any account can be called by a second name. For example, the records in the file pertaining to "NICHOLAS" can also be called by inputting "NABIL".

		[XEQ] "b"
FL NAME?		[R/S]
ACCT?	NICHOLAS	[R/S]
03:00:00.0		

The actual time that is in your display will depend on how long you spent running the example.

Print the files		[XEQ] "C"
FL NAME?		[R/S]
ACCT?	NICHOLAS	[R/S]

```
NICHOLAS NABIL
JOB NO 97321-4.5
BUILDING WRECKERS
3.151982
1.013860
3.151982
3.003861
-
3.151982
14.382076
BENJAMIN ELIAS
JOB NO 97330-2.5
PLUMBING CONTRACTORS
-
3.151982
13.364754
```

If the entire ASCII file were printed, it would be similar to the printout at the left.

If there is no printer in the system, [R/S] will advance the display to each successive line.

Whenever a time is output as an alpha-numeric display, its decimal form will be in the X-register and its HMS form will be in the Y-register.

```
NICHOLAS NABIL
JOB NO 97321-4.5
BUILDING WRECKERS

03/15/1982
01:01:38
03/15/1982
03:00:38
========
04:04:21
```

STATUS

SIZE : 003

FIX : 4, 6

TOTAL PROGRAM BYTES : 381

ASCII FILE SIZE : $\frac{(nA * 60) + (nA * n(D + 1) * 20) + 30}{7}$

DATA REGISTERS

00	n(days) * 24
01	Start/Finish time; Loop control = n records/account (Label [c])
02	Start/Finish date

FLAGS USED

21	Set	: Printer enabled
	Clear	: Printer disabled
23	Set	: Alpha data detected
	Clear	: No alpha data detected
55	Set	: Printer exists
	Clear	: No printer

FUNCTION LABELS

Label	Function
"$TIME"	Initializes a new file and prompts for variables
"b"	Stops the time for the named account and outputs total (stored)
"B"	Starts timer for named account
"C"	Prints/views file for named account, outputs file header information, each start date and elapsed time and total elapsed time

ASCII FILE RECORDS

0	Account name
1	Account code, billing number, purchase order, etc.
2	Remarks or description
3-4	Each succeeding pair of records contains a start date (MM.DDYYYY) in the first record and a total elapsed time (HH.MMSShh) in the second record
n	"_" character 95, delineates end of data space, start of work space
n	Start Date of running timer for this account
n	Start clock time (HH.MMSShh) of running timer for this account
n	Next account repeats the same format as the first account until the end of file

USER INSTRUCTIONS

SIZE: 003

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load program "$TIME".		[GTO] ..	PACKING
2	Initialize program "$TIME".		[XEQ]"$TIME"	N. ACCTS?
3	Input the number of individual clients			
	or accounts that are to be timed.	n.accts	[R/S]	N. DAYS?
4	Input the number of days or times that			
	each account will be timed.	n.days	[R/S]	FL NAME?
5	Select a name for the ASCII file, up			
	to 7 characters.	fl name	[R/S]	NAME?
6	Input the name of the first account in the			
	file. Up to 60 characters are allowed			
	for the name and following 2 lines.	name	[R/S]	CODE?
7	Input the purchase order number, billing			
	code or any other descriptive text.			
	Note that only character 95 is illegal			
	in the file.	code	[R/S]	REMARKS?
8	Any descriptive text may be input. If no			
	remarks are required, press [R/S].	remarks	[R/S]	NAME?
9	If the ASCII file is to contain more than			
	one account, the prompt "NAME?" will			
	recur as many times as are necessary.			
	Return to step 6.			
10	To start the clock for any account:		[B]	FL NAME?
11	If the ASCII File is the working file			
	(refer to HP 82180A Extended Functions/			
	Memory Module owner's manual for a			
	description of the "Working File"), then			
	no input is necessary. If not, or you			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	are uncertain:	fl name	[R/S]	ACCT?
12	Input the name of the account to be timed.	acct	[R/S]	START
13	To stop the clock for any account.		[shift][b]	FL NAME?
13a	Input the file name if it is not the			
	working file.	fl name	[R/S]	ACCT?
13b	Input the name of the account being timed.	name	[R/S]	nn:nn:nn.n
14	All of the named accounts may be timed			
	independently and started or stopped			
	as necessary. However, for each account,			
	only one starting time is stored until			
	the clock has been stopped and restarted.			
	If you start and stop the clock more			
	times than the number of days or times			
	inputted at initialization (step 4), the			
	file may become prematurely full and			
	not accept further data.			
15	Error recovery. If a timer was stored			
	accidently, it is not necessary to correct			
	the error. Simply restart the individual			
	timer when ready and the new time will			
	replace the previous starting time.			
	Refer to step 10 to start.			
16	To display the timer data stored under			
	a particular account name:		[C]	FL NAME?
16a	Input the file name if the file is not			
	the working file.	fl name	[R/S]	ACCT?
16b	Input the name of the account. If the			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	name begins with distinctive characters			
	that are not repeated elsewhere in the			
	file, only the first few characters need			
	to be inputted.	name	[R/S]	name
				code
				remarks
				date
				HH.MMSS
	The elapsed time for the displayed date			
	is also in the Y-register in HMS format			
	and in the X-register in decimal form.			
	When each complete cycle has been			
	displayed or printed, the total elapsed			
	time will be output preceded by:			--------
				HH.MMSS
	If a printer is not in the system, it			
	will be necessary to press [R/S] to			
	advance the display.			

PROGRAM LISTING

```
 01*LBL "$TI
ME"
 02 "N ACCTS
?"
 03 PROMPT
 04 STO 00          Temporary storage for
 05 60                file size calculation
 06 *               Allow approx. 60 char-
 07 "N DAYS?          acters for description
"                     and name
 08 PROMPT
 09 1
 10 +               Add one day to allow for
 11 RCL 00            overhead and scratch
 12 *                 registers
 13 20              Allow 9 characters for
 14 *                 each time and date and
 15 +                 2 characters for each
 16 30                record
 17 +               Allow 30 characters for
 18 7                 overhead
 19 /               Number of registers
 20 INT
 21 XEQ 07          Gosub file name
 22 CRFLAS          Create ASCII file
 23*LBL 00
 24 XEQ 01          Go to input prompt
 25 DSE 00            routine once for each
 26 GTO 00            account
 27 RTN
 28*LBL 01          Input prompting routine
 29 "NAME?"           to initialize accounts in
 30 AON               file
 31 STOP
 32 APPREC          Place in successive records
 33 "CODE?"
 34 STOP
 35 APPREC
 36 CF 23           Test for input remarks
 37 "REMARKS
?"
 38 STOP
 39 FC?C 23
 40 "--"            Store default message to
 41 APPREC            have expected number
 42 AOFF              of records
 43 CLA
 44 95              Place underscore charac-
 45 XTOA              ter at end of account
 46 APPREC            data space as end of
 47 FIX 6             account delineator
 48 CLX
 49 CLA
 50 ARCL X
 51 APPREC
 52 APPREC          Accumulate two records
 53 RTN               of zeros to mark the
 54*LBL b             work (scratchpad) space
 55 TIME
 56 XEQ 02          Immediately saves time
 57 GETREC            for accuracy
 58 ANUM            Gosub to store time and
 59 DATE              date in main memory
 60 DDAYS             scratch area
 61 1               Set display mode and
 62 -                 position file
 63 24              Convert first date to
 64 *                 numeric data
 65 STO 00          n days
 66 X<>Y
 67 1               Number of hours calcu-
 68 -                 lation
 69 SEEKPT
 70 INSREC          Temporary
 71 3               Swap with previous
 72 +                 pointer value
 73 SEEKPT
 74 GETREC
 75 24              Insert time into next
 76 ANUM              space in file
 77 HMS-
 78 RCL 01
 79 HMS+            Get time of start clock
 80 RCL 00            from scratch space
 81 +
 82 CLA             Convert to numeric data
 83 ARCL X
 84 X<>Y
 85 2               n hours in current day
 86 -
 87 SEEKPT
 88 INSREC
 89 RDN             + number of days * 24
 90 ENTER↑
 91 CLA
 92 ATIME24
 93 HR              Insert elapsed time into
 94 AVIEW             file
 95 RTN
 96*LBL B           Load stack
 97 TIME
 98 TONE 9          Format printer display
 99 XEQ 02
100 CLA             Leave decimal value of
101 ARCL 01            time in X register for
102 INSREC             calculations

                    Start clock

                    Temporary storage done
                      immediately for accuracy
                    Gosub temporary storage
                      and file position

                    Insert 6 digit unformatted
                      (numeric) date in scratch
                      space
```

PROGRAM LISTING

```
103 CLA
104 ARCL 02
105 INSREC          Insert time in scratch space
106 2
107 +
108 SEEKPT
109 DELREC          Delete previous scratch
110 DELREC          times. Insert must be performed before deletion to prevent insert at end of file from becoming first record (0).
111 "START"
112 PROMPT
113*LBL 02          Confirm action
114 STO 01          Temporary storage of time and date while user is prompted for file name prevents loss of data in stack
115 DATE
116 STO 02
117 FIX 6
118 XEQ 06          Format for full precision of time/date data
119 95              Prompt account name, return -1 if misspelled
120 XTOA
121 POSFL           Position to end of data space by finding character 95
122 1
123 +
124 SEEKPT          Position file to scratch-pad area
125 RTN
126*LBL C           Routine to display time
127 SF 21           Enable printer or cause halt on AVIEW
128 XEQ 06
129 95              Prompt account name
130 XTOA            Position to scratch space delineator
131 POSFL
132 RCL Z
133 -
134 3               Subtract 3 records for file descriptive data
135 -
136 2               Divide into pairs
137 /
138 STO 01
139 RCL Z           Set index for loop
140 SEEKPT
141 XEQ 04          Print/view loop will execute three times
142 ADV
143 CLX
144 STO 00
145*LBL 03
146 FIX 6           Full precision format
147 GETREC
148 ANUM            Convert to numeric data
149 CLA
150 ADATE           Format printer display
151 XEQ 05
152 GETREC
153 ANUM
154 ENTER↑          Load X and Y with time
155 FIX 4           Sufficient precision for time display
156 CLA
157 ATIME24         Format time
158 HR              Leave decimal value of time in X for calculations
159 XEQ 05
160 RCL 00
161 HMS+            Running total of elapsed times
162 STO 00
163 DSE 01
164 GTO 03
165 "=======        Signify addition
="
166 XEQ 05
167 RCL 00          Print or view
168 CLA
169 ATIME24
170 ENTER↑          Leave decimal form of total elapsed time in X register
171 HR
172 GTO 05          Last portion, RTN not necessary
173*LBL 04
174 XEQ 04          Speeds loops and saves 1 byte
175 XEQ 04
176*LBL 04
177 GETREC
178*LBL 05
179 FS? 55          If printer
180 PRA
181 FC? 55          If not
182 AVIEW
183 RTN
184*LBL 06
185 XEQ 07          Gosub file name
186 CLX
187 SEEKPT          Go to top of file
188 "ACCT?"
189 AON             ALPHA mode
190 STOP
191 AOFF
192 POSFL
193 X<0?            If not found, indicate error
194 STOP
195 CLA
196 RTN
197*LBL 07
198 CF 23
199 "FL NAME
?"
200 AON
201 STOP
202 AOFF
203 FC?C 23         Test input, if no input, calculator is positioned to working file
204 CLA
205 .END.
```

PROGRAM DESCRIPTION

PLAYBACK PROGRAMMABLE TIMER

This timer may be interactively programmed to playback a series of messages or run user-defined programs at specified intervals. The program's features include compatibility with any HP-41 mass storage device and routines to store, recall, save and edit playback segments. Manual and auto playback modes are available.

"Playback" uses three control alarms to perform timing functions:

1. A control alarm which calls routine "NM" (next manual) is set in manual mode. It is used to start the playback and works by storing time in an accumulator register, R00, and setting the control alarm to activate relative to that time. The display flag annunciator for Flag 00 signifies manual mode.

2. Label "NA" (next automatic) is the control alarm entry point in the auto mode. In this mode, no visible flag is seen. Label "NM" falls through "NA" during its execution. "NA" does not store a new starting time with each playback step, but references all time offsets to the starting time of the first step, increments the step counter and continues until the last record.

3. Label "T" activates a periodic tone of a frequency chosen by the user during initial input and variable for every step. Period must be less than the step period and greater than 6 seconds to avoid the possibility of a past due alarm due to slow processing or search through a long table of global labels.

4. Label "AR" (called in the program as label 10 is the "ALMREL" program in the HP 82182A Time Module Owner's Manual. It is given a global label to enable it to be called as a subroutine in a user-defined program.

A NOTE ON INPUT

Invalid erroneous inputs must all be trapped to prevent playback errors. Tone intervals longer than the step interval, or short enough to cause a past due alarm, and failure to input a playback message or program name are examples of such inputs. If invalid inputs are detected, the input prompt is repeated.

User-defined programs are detected by the presence of zero in the first register of the step record. A flag is set, the next alarm set and the balance of the playback sequence skipped. The user function or program is addressed by an indirect GTO. It becomes a subroutine by virtue of being called by a control alarm. In other words, the user program will be executed and finished, and depending upon its structure, will wait until the alarms "NA" or "NM" are activated.

"Playback Programmable Timer" may be used in a variety of applications and amusements including children's games and activities, photographic darkroom, calisthenics and warm-up exercises, laboratory experiments, baking bread, debates (speaker name 1, speaker name 2, rebuttal 1, rebuttal 2, etc), timing dramatic scripts and routines, prompting signposts and speed changes in auto rallyes, timing long distance phone calls.

SAMPLE PROBLEM

This example will require a minimum SIZE of 3 registers. Program the "Playback Programmable Timer" to display a series of five flexibility exercises. Two of the steps require a faster cadence than the pace available in the program. Three of the steps will sound a tone every ten seconds indicating a change of direction.

DISPLAY	INPUT	FUNCTION
Initialize the program		[XEQ] "PB"
N STEPS?	5	[R/S]
MESSAGE?	BODY TWISTS	[R/S]
STEP HMS?	.01	[R/S]
TONE/HMS?	.001	[R/S]
TONE N?	9	[R/S]
MESSAGE?	FOOT CIRCLES	[R/S]
STEP HMS?	.003	[R/S]
TONE/HMS?	.001	[R/S]
TONE N?	8	[R/S]
MESSAGE?	no input	[R/S]
PROGRAM?	FASTP	[R/S]
STEP HMS?	.01	[R/S]
MESSAGE?	CALF STRETCH	[R/S]
STEP HMS?	.0045	[R/S]
TONE/HMS?	.001	[R/S]
TONE N?	7	[R/S]
MESSAGE?	no input	[R/S]
PROGRAM?	FASTP	[R/S]
STEP HMS?	.003	[R/S]
END		
Recall the Playback series to confirm correct entry.		
	1	[C]
1=BODY TWISTS		[R/S]*
HMS=00:01:00		[R/S]*
TN 9/00:00:10		[R/S]* [R/S]
2=FOOT CIRCLES		[R/S]*
HMS=00:00:30		[R/S]*
TN 8/00:00:10		[R/S]* [R/S]
3=PRGM, FASTP		[R/S]*

DISPLAY	INPUT	FUNCTION
HMS=00:01:00		[R/S]* [R/S]
4=CALF STRETCH		[R/S]*
HMS=00:00:45		[R/S]*
TN=0/00:00:10		[R/S]* [R/S]
5=PRGM, FASTP		[R/S]*
HMS=00:00:30		[R/S]*
Load the fast-pace program, "FASTP".		
Initialize the Auto Run "Playback" mode.		
		[E]
If "AUTO" is not seen in the display, try again.		
		[E]
		[R/S]
[AUTO]		

* [R/S] is not necessary if printer exists.

```
 01◆LBL "FAS
TP"                  Initialize flag not used by "PB"
 02 CF 02            Test recall index to determine
 03 23                if earlier or later occurrence
 04 RCL 03            of "FASTP"
 05 X<Y?
 06 SF 02
 07 11
 08 ENTER↑
 09 FS?C 02
 10 +                Earlier, one minute timing
 11 STO 09           R09 unused by "PB"
 12 RUNSW
 13 2 E-4
 14 TIME
 15 HMS+             Apply 2 second time offset to
 16 SETSW             stopwatch for fast loop
 17 "↑↑FP"           Control alarm entry point
 18◆LBL "FP"         Control alarm
 19 TONE 9
 20 CLST
 21 RCLSW            Initialize the stack for alarm
 22 DSE 09            without reset
 23 XYZALM
 24 END
```

STATUS

SIZE : n * 4 + 11

FIX : 0, 4

TOTAL PROGRAM BYTES : 538

USER MODE : ON

DATA REGISTERS

00	Total time, run message
01	Total time, run tone
02	Store index
03	Recall and run index
04	Edit, then restore index to R02, indirect tone in run mode
05	Alpha message for playback
06	Alpha message, next six characters
07	Recall n, (Label C), tone HH.MMSS
08	DSE index for tone, label "T"
09-10	Available for user program
11	Alpha message, first six characters
12	Alpha message, last six characters NOTE: 0 if user program optioned program name, indirect GTO
13	Step HH.MMSS, total time until start of next step
14	Tone pitch, 1 to 9 - decimal point - tone cycle .MMSS
15	Begin next record (step two)

FLAGS USED

00	Set	:	Manual mode
	Clear	:	Not manual mode
04	Set	:	Edit mode
	Clear	:	Not edit mode
05	Set	:	Skip tone input if a user-defined program
	Clear	:	Allow tone input
06	Set	:	Run user's program
	Clear	:	Don't run the user's program
07	Set	:	No tone
	Clear	:	Tone
08	Set	:	First pass - skip prompt
	Clear	:	All other cases - allow prompt
21	Set	:	Printer enabled
	Clear	:	Printer disabled
23	Set	:	ALPHA input detected
	Clear	:	No ALPHA input detected
25	Set	:	No error detected
	Clear	:	Error detected
55	Set	:	Printer exists
	Clear	:	No printer

FUNCTION LABELS

Label	FUNCTION
"B"	Store playback message and parameters
"C"	Recall playback message and parameters
"D"	Edit playback message and parameters
"E"	Toggle Auto or Manual mode, initialize run

USER INSTRUCTIONS

SIZE: 4n+11

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load program "PB". Set User Mode.		[USER]	
2	Initialize "PB".		[XEQ]"PB"	N STEPS?
	Input number of steps or instructions.	n	[R/S]	MESSAGE? or
				RESIZE = (nn)
	If necessary, resize and start over.			
	The size is determined by multiplying			
	the number of steps by 4 and adding 11.			MESSAGE?
3	Input message to be played back, up to			
	12 alpha characters.	message	[R/S]	
	If no message is to be run, but rather			
	a program.	no input	[R/S]	
4	Key in the name of your playback program.	program	[R/S]	STEP HMS?
5	Input the length of time that the			
	message or program message is to be			
	displayed.	HH.MMSS	[R/S]	TONE/HMS?
6	If a periodic tone is wanted, input the			
	time interval.	HH.MMSS	[R/S]	TONE N?
7	Input tone pitch, from 0 to 9.	n	[R/S]	MESSAGE?
8	Return to step 3 of instructions			
	until done.			
9	To recall input steps at any time,			
	starting with nth step.	n	[C]	n= (...)
	To continue viewing step.		[R/S]	HMS= (nn:nn:nn)
	Note: [R/S] is not necessary with a			
	printer attached.		[R/S]	TN n/(nn:nn:nn)
10	To view following steps.		[R/S]	TN n/(nn:nn:nn)
11	To correct or change any step.	n	[D]	MESSAGE?

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
12	Refer to step 3 of instructions for input			
	parameters. Once edit is started, follow			
	through entire step or subsequent steps			
	will be stored in unplanned locations.			
13	The entire playback sequence may be			
	saved on any HP-41 storage medium.			
	Magnetic cards:		[RCL] 02	
			[FRC]	
			[WDTAX]	
	Mass storage, such as HP 82161A Digital			
	Cassette Drive:		[ALPHA] YOUR	
			FILE NAME	
			[ALPHA] 1000	
			[RCL] 02	
			[FRC][*]	
			[CREATE] 0	
			[SEEKR]	
			[LASTX]	
			[WRTRX]	
14	To play back the stored messages:		[E]	AUTO OR MANUAL
	If the desired mode is not seen,			
	press again.		[E]	
	Auto mode will cycle continuously from			
	step to step after starting. Manual mode			
	requires the user to restart at each step.		[R/S]	
15	Guidelines for user-defined programs to			
	be run in place of playback messages:			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	Run time of user program should not			
	exceed the programmed step time.			
	Five levels of subroutines are available.			
	Registers 9 and up are available for			
	data storage.			
	Flag 21 should not be set unless a printer			
	is present, flag 00 should not be used,			
	nor should flag 26 or 27.			
	The program should not use the T+X			
	function nor any other functions that			
	will interfere with the operation of an			
	HP 82182A Time Module based program.			
	The stack and alpha registers are			
	available for program use.			
	Do not use the function OFF.			
	The global label should not have more than			
	6 characters for indirect addressing.			
	Avoid key assignments to the top row of			
	user keys.			

PROGRAM LISTING

```
01♦LBL "PB"
02 "N STEPS
?"                    Number of steps required to calculate index
03 PROMPT
04 4                  Each step record consists of 4 registers
05 *
06 11                 Add 10 housekeeping registers
07 +
08 "RESIZE>
="                    Format size prompt
09 FIX 0
10 ARCL X
11 1
12 -
13 SF 25
14 STO IND
X                     Test for existence of highest number R required
15 FC? 25
16 PROMPT             Display if necessary
17 FC?C 25
18 GTO "PB"           Repeat initialization prompts if size inadequate
19 .1
20 %                  /1000
21 10
22 +                  Add scratch registers
23 STO 02             Save index
24 0
25♦LBL 00
26 STO IND
Y                     Clear block of registers to prevent recall of garbage playback
27 ISG Y
28 GTO 00
29♦LBL B              Store routine
30 CF 04              Clear edit Flag
31♦LBL 20
32 "END"
33 ISG 02
34 GTO 01             Display END (no more data registers) if indexx=eee
35 PROMPT
36♦LBL 01
37 CF 05              Flag 05 will be tested for 'skip tone input' on line 63
38 CF 23
39 "MESSAGE           Prompt message input
?"
40 AON
41 STOP
42 ASTO X
43 FS?C 23            No input?
44 GTO 02             Then try next possibility
45 CLX
46 "PROGRAM
?"                    Input program name if chosen in place of message
47 STOP
48 FC? 23             If no input here, cycle back to message prompt
49 GTO 01
50 SF 05              If program name input, skip tone prompt
51♦LBL 02             Store message or program name, first part is in X, if program name x=0
52 AOFF
53 STO IND
02
54 ISG 02
55 FC?C 23
56 ASHF
57 ASTO IND
02
58 ISG 02
59 "STEP HM
S?"                   Input step time HH.MMSSx
60 PROMPT
61 STO IND
02                    Save in next register, 3rd register of step record
62 CLX
63 FS?C 05            Skip tone flag set?
64 GTO 04
65♦LBL 03
66 "TONE/HM
S?"                   Tone interval desired, if any
67 CLX
68 PROMPT
69 X=0?               No input?
70 GTO 04
71 6 E-4              6 seconds is min. time allowed to prevent possible past-due alarm due to slow processing or label search
72 X>Y?
73 GTO 03             Cycle back if error
74 X<>Y
75 RCL IND
02
76 X<=Y?              Test if greater than step HMS
77 GTO 03             Cycle back on error
78 X<>Y               Return input to X
79♦LBL 04             If no tone input prompted execution resumes here
80 ISG 02             Update index
81 STO IND
02
82 X=0?
83 GTO 06
84♦LBL 05
85 9                  Tone 9 supplied as default
86 "TONE N?           Prompt pitch
"
```

PROGRAM LISTING

```
 87 SF 25
 88 PROMPT
 89 TONE IND
 X
 90 FC?C 25
 91 GTO 05
 92 ST+ IND
02
 93*LBL 06
 94 RCL 04
 95 FS?C 04
 96 STO 02
 97 GTO B
 98*LBL D
 99 4
100 *
101 6
102 +
103 RCL 02
104 FRC
105 +
106 X<> 02
107 STO 04
108 SF 04
109 GTO 20
110*LBL C
111 SF 21
112 CF 29
113 STO 07
114*LBL 07
115 ADV
116 RCL 07
117 4
118 *
119 7
120 +
121 RCL 02
122 FRC
123 +
124 STO 03
125 FIX 0
126 CLA
127 ARCL 07
128 "⊢="
129 RCL IND
03
130 SIGN
131 X≠0?
132 "⊢PRGM.
"
```

Prepare to test parameter

Does tone exist?

If invalid parameter, repeat prompt

Add to time register

R02 was stored here at beginning of routine, if this cycle was an edit, then restore original index

Continue

Edit routine

Calculate beginning register and store temporary index

Add end register and store temporary index

Exchange temporary index with actual index and save

Set flag to signal edit

Bypass clear edit flag at Label (B) entry point

Enable printer or halt on AVIEW

Display formatting

Save step n to permit SST to next step on completion

For printer

Step n (4)

Calculate recall index

Get step n to format output display

Get first part of message

Not ALPHA? Then program name

```
133 X=0?
134 ARCL IND
 03
135 ISG 03
136 ARCL IND
 03
137 AVIEW
138 ISG 03
139 FIX 4
140 "HMS= "
141 RCL IND
03
142 ATIME24
143 AVIEW
144 "TN "
145 FIX 0
146 ISG 03
147 RCL IND
03
148 INT
149 ARCL X
150 "⊢/"
151 LASTX
152 FRC
153 FIX 4
154 ATIME24
155 X≠0?
156 AVIEW
157 RTN
158 ISG 07
159 CLX
160 GTO 07
161*LBL E
162 FC? 55
163 CF 21
164 FC?C 00
165 SF 00
166 "AUTO"
167 FS? 00
168 "MANUAL"
169 AVIEW
170 10
171 RCL 02
172 FRC
173 +
174 STO 03
175 CF 06
176 SF 08
177 RTN
178*LBL "NM"
```

Display ALPHA message

Display step time

Display tone (n) in format: TONE n every .MMSS

Integer part is tone pitch

Fractional part is .MMSS

Only display if parameters exist

Add 1 to step number continue sequentially

Initialize Playback, clear Flag 21 to prevent halt on AVIEW if no printer and toggle playback mo mode

Toggle mode flag. Alternate use of routine sets other mode

In 'AUTO' mode, playback is continuous

In 'MANUAL' mode, playback stops after every complete step

Calculate recall index, starting with first playback record

Index register

Flag 06 will be tested during playback for 'run user program' on first pass, skip manual mode prompt

PROGRAM LISTING

```
179 "RUN"
180 FC?C 08
181 PROMPT
182 TONE 9
183 TIME
184 STO 00
185 STO 01
186◆LBL "NA"
187 ISG 03
188 GTO 08
189 "END"
190 PROMPT
191◆LBL 08
192 CF 07
193 RCL IND
03
194 STO 05
195 SIGN
196 X≠0?
197 SF 06
198 ISG 03
199 RCL IND
03
200 STO 06
201 ISG 03
202 "↑↑NA"
203 FS? 00
204 "↑↑NM"
205 RCL IND
03
206 RCL 00
207 XEQ 10
208 XYZALM
209 STO 00
210 ISG 03
211 RCL IND
03
212 X=0?
213 SF 07
214 X=0?
215 GTO 09
216 INT
217 STO 04
218 LASTX
219 FRC
220 STO 07
221 HR
222 RCL 03
223 1
224 -
```

Control alarm entry point, 'Next, Manual'

Skip prompt on first pass

Audible feedback: Clock has started running

Save start clock time

Entry point for Control Alarm, 'Next, Automatically'

End of playback

Clear tone flag

Get message or program name

SIGN test can test for alpha characters whereas X=0? would be ERROR if ALPHA requiring additional flag test

Set 'get User program' flag

Get last 6 characters and save

Control alarm dependent upon status of Flag 00, auto/manual

Get step time

Add to finish time of last step and calculate time offset pa

Parameters in stack, go

Save finish time

Get tone parameters, if any

Clear tone flag

Get tone cycle time

```
225 RCL IND
X
226 HR
227 RCL Z
228 /
229 FIX 0
230 RND
231 STO 08
232 GTO 09
233◆LBL "T"
234 TONE IND
 04
235◆LBL 09
236 CLA
237 ARCL 05
238 ARCL 06
239 FC? 06
240 AVIEW
241 RCL 00
242 FS? 07
243 STO 01
244 FS?C 06
245 GTO IND
06
246 FS?C 07
247 RTN
248 "↑↑T"
249 RCL 01
250 RCL 07
251 XEQ 10
252 STO 01
253 DSE 08
254 XYZALM
255 RTN
256◆LBL 10
257◆LBL "AR"
258 HMS+
259 ENTER↑
260 ENTER↑
261 24
262 /
263 INT
264 DATE
265 X<>Y
266 DATE+
267 LASTX
268 24
269 *
270 ST- Z
271 CLX
272 STO T
273 RDN
274 X<>Y
275 .END.
```

Calculate index for number of tone cycles, not to exceed time of playback step

'Tone' control alarm

Audible

Get message and display

Display only if message, not program name

If tone is part of playback step

If User Program, run it

If no tone, stop and wait for next control alarm

If tone, set tone control alarm for next occurrence

Tone interval

Time offset subroutine

Save finish time of tone period for next occurrence

Set alarm if index permits

ALMREL time offset subroutine

Global label allows User program (Flag 06 set) to use

Label 10 as a subroutine

Calculate number of days and set up stack with XYZ parameters

PROGRAM DESCRIPTION

RANDOM SEED GENERATOR

The random seed generator may be used with any random number generator to provide an automatic seed, different every time, that will assure a long non-repetitive period. The routine takes the seconds and hundredths of seconds at the time it is called and multiplies them by the clock time to get a larger, unpredictable number. The number is then increased to a large value that will not exceed the precision of the HP-41 and tested to check for multiples of five or two which would appreciably shorten the period of the random number generator. If the number passes, it is converted to a fraction and returned to the calling routine. If it fails, the cycle is repeated with a new time until the number passes. The result will be a series of accidental digits, at least seven in length, ending in 1, 3, 7 or 9.

On test runs of up to 5000 iterations the 6 least significant digits exhibited the most randomness with a typical mean of .4995 to .501 and a standard deviation of .27.

The routine uses only the stack and requires no other subroutines, flags or data registers. It is used in the example problem with the random generator from the HP-41 Standard Pac, page 24, developed by Don Malm. It will generate one million distinct random numbers between 0 and 1 regardless of the initial starting value (seed). An excellent article listing several references on random number generation may be found in the July, 1980 issue of the Hewlett-Packard Journal, written by Homer Russell.

PROGRAM LISTING

```
01◆LBL "RZ"
02◆LBL 01
03 TIME
04 1 E2
05 *
06 FRC
07 TIME
08 *
09 1 E8
10 *
11 INT
12 STO Y
13 5
14 MOD
15 X=0?
16 GTO 01
17 RCL Y
18 2
19 MOD
20 X=0?
21 GTO 01
22 R↑
23 1 E9
24 /
25 FRC
26 .END.
```

Use of local label will speed iterations if number fails test and routine must be re-executed one or more times

Separate seconds and hundredths from clock time

Multiply by clock time to get larger number (more digits)

Enlarge to a value that will remain within the precision of the HP-41

Truncate for test

Save in stack

Test if divisible by 5

If a multiple of 5, try another number

Get large integer back

Test for multiple of 2

If number fails, try again

Get large number back if passed test

Back to fraction, seed formed, exit

NOTES

HEWLETT-PACKARD

HP-41

USERS' LIBRARY

Time Module Solutions I

For the HP 82182A Time Module

Bar Codes

TIMER SOLUTIONS I

APPOINTMENT CALENDAR
APPT
PROGRAM REGISTERS NEEDED: 87

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 4)

ROW 2 (4 : 9)

ROW 3 (10 : 17)

ROW 4 (18 : 25)

ROW 5 (25 : 28)

ROW 6 (29 : 36)

ROW 7 (36 : 44)

ROW 8 (44 : 49)

ROW 9 (49 : 56)

ROW 10 (57 : 61)

ROW 11 (61 : 69)

ROW 12 (70 : 76)

ROW 13 (76 : 83)

ROW 14 (83 : 89)

ROW 15 (90 : 99)

ROW 16 (100 : 106)

ROW 17 (106 : 110)

ROW 18 (110 : 115)

APPOINTMENT CALENDAR
APPT

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 19 (116 : 125)

ROW 20 (125 : 133)

ROW 21 (133 : 142)

ROW 22 (143 : 149)

ROW 23 (149 : 158)

ROW 24 (159 : 167)

ROW 25 (167 : 175)

ROW 26 (175 : 182)

ROW 27 (183 : 189)

ROW 28 (189 : 196)

ROW 29 (197 : 203)

ROW 30 (204 : 213)

ROW 31 (213 : 218)

ROW 32 (219 : 227)

ROW 33 (228 : 237)

ROW 34 (238 : 242)

ROW 35 (243 : 252)

ROW 36 (253 : 259)

ROW 37 (259 : 265)

ROW 38 (265 : 269)

ROW 39 (270 : 274)

ROW 40 (275 : 281)

ROW 41 (282 : 290)

ROW 42 (291 : 297)

ROW 43 (297 : 303)

ROW 44 (304 : 308)

ROW 45 (309 : 313)

ROW 46 (314 : 318)

ROW 47 (319 : 321)

APPOINTMENT CALENDAR
LOAD–CASSETTE VERSION
PROGRAM REGISTERS NEEDED: 15

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 4)

ROW 2 (4 : 9)

ROW 3 (9 : 15)

ROW 4 (15 : 23)

ROW 5 (23 : 25)

ROW 6 (26 : 34)

ROW 7 (34 : 38)

ROW 8 (39 : 47)

ROW 9 (47 : 47)

APPOINTMENT CALENDAR
LOAD–X FUNCTIONS
PROGRAM REGISTERS NEEDED: 11

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 3)

ROW 2 (4 : 11)

ROW 3 (12 : 14)

ROW 4 (15 : 23)

ROW 5 (24 : 28)

ROW 6 (28 : 32)

APPOINTMENT CALENDAR
LOAD–CARD READER VERSION
PROGRAM REGISTERS NEEDED: 4

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 4)

ROW 2 (4 : 8)

ROW 3 (8 : 8)

ROW 1 (1 : 4)

ROW 2 (5 : 14)

ROW 3 (14 : 21)

ROW 4 (21 : 27)

ROW 5 (28 : 31)

ROW 6 (31 : 40)

ROW 7 (40 : 49)

ROW 8 (50 : 58)

ROW 9 (59 : 66)

ROW 10 (66 : 72)

ROW 11 (72 : 78)

ROW 12 (78 : 84)

ROW 13 (84 : 87)

ROW 14 (88 : 93)

ROW 1 (1 : 4)

ROW 2 (5 : 7)

ROW 3 (7 : 15)

ROW 4 (15 : 23)

ROW 5 (23 : 27)

ROW 6 (27 : 32)

ROW 7 (32 : 38)

ROW 8 (38 : 43)

ROW 9 (44 : 52)

ROW 10 (52 : 62)

ROW 11 (63 : 73)

ROW 12 (74 : 83)

ROW 13 (83 : 87)

ROW 14 (87 : 90)

ROW 15 (90 : 100)

ROW 16 (100 : 109)

ROW 17 (110 : 114)

ROW 18 (115 : 119)

ROW 19 (120 : 126)

ROW 20 (127 : 134)

ROW 21 (135 : 143)

ROW 22 (143 : 149)

ROW 23 (149 : 154)

ROW 24 (154 : 158)

ROW 25 (159 : 167)

ROW 26 (167 : 172)

ROW 27 (173 : 176)

ROW 28 (176 : 182)

ROW 29 (183 : 186)

ROW 30 (187 : 195)

ROW 31 (195 : 200)

ROW 32 (200 : 202)

ROW 33 (202 : 206)

ROW 34 (207 : 213)

ROW 35 (214 : 223)

ROW 36 (223 : 230)

ROW 37 (230 : 236)

ROW 38 (236 : 242)

ROW 39 (242 : 249)

ROW 40 (250 : 256)

ROW 41 (257 : 260)

ROW 1 (1 : 4)

ROW 2 (5 : 10)

ROW 3 (10 : 15)

ROW 4 (15 : 18)

ROW 5 (18 : 25)

ROW 6 (26 : 29)

ROW 7 (29 : 36)

ROW 8 (36 : 43)

ROW 9 (44 : 50)

ROW 10 (50 : 59)

ROW 11 (59 : 68)

ROW 12 (69 : 78)

ROW 13 (79 : 87)

ROW 14 (87 : 90)

ROW 15 (90 : 98)

ROW 16 (99 : 102)

ROW 17 (102 : 108)

ROW 18 (109 : 116)

AUTO TRIP COMPUTER AND
SPEED CALIBRATION CAL

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 19 (117 : 117)

AUTO TRIP COMPUTER AND
SPEED CALIBRATION TRIP
PROGRAM REGISTERS NEEDED: 83

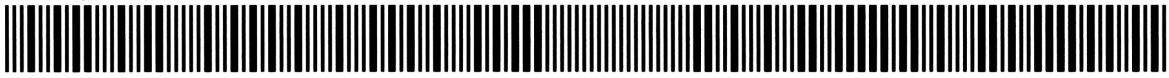
HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 4)

ROW 2 (5 : 10)

ROW 3 (10 : 16)

ROW 4 (17 : 21)



ROW 5 (21 : 25)

ROW 6 (25 : 31)

ROW 7 (31 : 38)

ROW 8 (38 : 42)

ROW 9 (42 : 51)

ROW 10 (52 : 54)

ROW 11 (54 : 59)

ROW 12 (60 : 68)

ROW 13 (69 : 73)

ROW 14 (74 : 81)

ROW 15 (82 : 87)

ROW 16 (87 : 91)

ROW 17 (91 : 95)

ROW 18 (95 : 104)

ROW 19 (105 : 113)

ROW 20 (114 : 121)

ROW 21 (122 : 125)

ROW 22 (126 : 133)

ROW 23 (134 : 140)

ROW 24 (140 : 146)

ROW 25 (146 : 150)

ROW 26 (150 : 156)

ROW 27 (156 : 160)

ROW 28 (160 : 164)

ROW 29 (164 : 171)

ROW 30 (172 : 181)

ROW 31 (181 : 190)

ROW 32 (190 : 197)

ROW 33 (197 : 204)

ROW 34 (204 : 208)

ROW 35 (208 : 212)

ROW 36 (213 : 218)

ROW 37 (218 : 223)

ROW 38 (224 : 230)

ROW 39 (231 : 235)

ROW 40 (236 : 239)

ROW 41 (239 : 244)

ROW 42 (244 : 249)

ROW 43 (250 : 254)

ROW 44 (255 : 260)

ROW 45 (261 : 261)

FOUR CHANNEL CONTROLLER
4CON
PROGRAM REGISTERS NEEDED: 68

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 5)

ROW 2 (6 : 14)

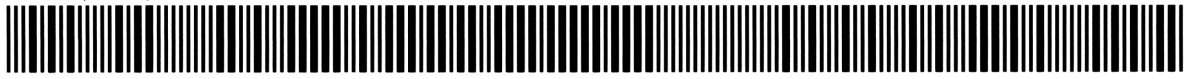
ROW 3 (15 : 20)

ROW 4 (21 : 29)

ROW 5 (30 : 35)

ROW 6 (36 : 42)

ROW 7 (43 : 50)



ROW 8 (50 : 55)

ROW 9 (56 : 59)

ROW 10 (59 : 65)

ROW 11 (65 : 70)

ROW 12 (70 : 78)

ROW 13 (79 : 86)

ROW 14 (87 : 96)

ROW 15 (96 : 105)

ROW 16 (105 : 114)

ROW 17 (115 : 124)

ROW 18 (124 : 133)

ROW 19 (134 : 140)

ROW 20 (140 : 148)

ROW 21 (149 : 156)

ROW 22 (157 : 164)

ROW 23 (165 : 169)

ROW 24 (169 : 172)

ROW 25 (173 : 176)

ROW 26 (176 : 182)

ROW 27 (182 : 186)

ROW 28 (187 : 195)

ROW 29 (195 : 201)

ROW 30 (201 : 207)

ROW 31 (208 : 215)

ROW 32 (216 : 221)

ROW 33 (222 : 227)

ROW 34 (227 : 233)

ROW 35 (234 : 242)

ROW 36 (243 : 251)

ROW 37 (251 : 251)

ROW 1 (1 : 6)

ROW 2 (6 : 11)

ROW 3 (11 : 18)

ROW 4 (19 : 22)

LOGBOOK
$TIME
PROGRAM REGISTERS NEEDED: 55

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 2)

ROW 2 (2 : 7)

ROW 3 (7 : 14)

ROW 4 (15 : 23)

ROW 5 (24 : 29)

ROW 6 (29 : 34)

ROW 7 (35 : 37)

ROW 8 (38 : 45)

ROW 9 (45 : 52)

ROW 10 (53 : 59)

ROW 11 (59 : 69)

ROW 12 (69 : 76)

ROW 13 (77 : 87)

ROW 14 (88 : 97)

ROW 15 (97 : 104)

ROW 16 (104 : 111)

ROW 17 (111 : 118)

ROW 18 (118 : 125)

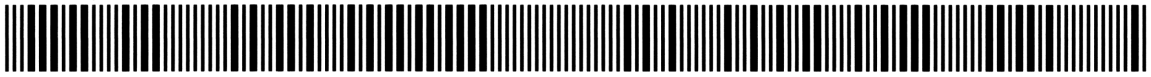
ROW 19 (126 : 131)

ROW 20 (132 : 141)

ROW 21 (141 : 149)

ROW 22 (150 : 156)

ROW 23 (157 : 164)



ROW 24 (165 : 167)

ROW 25 (168 : 175)

ROW 26 (175 : 183)

ROW 27 (184 : 188)

ROW 28 (189 : 199)

ROW 29 (199 : 203)

ROW 30 (204 : 205)

PLAYBACK PROGRAMMABLE TIMER
PB
PROGRAM REGISTERS NEEDED: 77

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 2)

ROW 2 (2 : 8)

ROW 3 (8 : 14)

ROW 4 (14 : 20)

ROW 5 (21 : 29)

ROW 6 (29 : 34)

ROW 7 (35 : 39)

ROW 8 (39 : 46)

ROW 9 (46 : 50)

ROW 10 (50 : 58)

ROW 11 (58 : 61)

ROW 12 (61 : 66)

ROW 13 (66 : 71)

ROW 14 (72 : 80)

ROW 15 (81 : 86)

ROW 16 (86 : 92)

ROW 17 (93 : 101)

ROW 18 (102 : 110)

ROW 19 (110 : 120)

ROW 20 (121 : 129)

ROW 21 (129 : 134)

ROW 22 (134 : 140)

ROW 23 (140 : 145)

ROW 24 (145 : 152)

ROW 25 (153 : 161)

ROW 26 (161 : 166)

ROW 27 (166 : 170)

ROW 28 (171 : 178)

ROW 29 (178 : 183)

ROW 30 (184 : 189)

ROW 31 (189 : 197)

ROW 32 (197 : 202)

ROW 33 (203 : 207)

ROW 34 (208 : 215)

ROW 35 (216 : 227)

ROW 36 (227 : 233)

ROW 37 (234 : 242)

ROW 38 (242 : 248)

ROW 39 (249 : 257)

ROW 40 (257 : 264)

ROW 41 (264 : 272)

ROW 42 (273 : 275)

PLAYBACK PROGRAMMABLE TIMER
FASTP
PROGRAM REGISTERS NEEDED: 9

HEWLETT PACKARD
TIMER SOLUTION BOOK

ROW 1 (1 : 3)

ROW 2 (4 : 12)

ROW 3 (13 : 17)

ROW 4 (17 : 22)

ROW 5 (22 : 24)

RANDOM SEED GENERATOR
RZ

PROGRAM REGISTERS NEEDED: 7

ROW 1 (1 : 5)

ROW 2 (6 : 14)

ROW 3 (15 : 23)

ROW 4 (23 : 26)

NOTES

NOTES

NOTES


HEWLETT
PACKARD

Hewlett-Packard Software

In terms of power and flexibility, the problem-solving potential of the HP-41C programmable calculator is nearly limitless. And in order to see the practical side of this potential, HP has different types of software to help save you time and programming effort. Every one of our software solutions has been carefully selected to effectively increase your problem-solving potential. Chances are, we already have the solutions you're looking for.

Application Pacs

To increase the versatility of your HP-41C, HP has an extensive library of "Application Pacs". These programs transform your HP-41C into a specialized calculator in seconds. Included in these pacs are detailed manuals with examples, minature plug-in Application Modules, and keyboard overlays. Every Application Pac has been designed to extend the capabilities of the HP-41C.

You can choose from:

Aviation
Clinical Lab
Circuit Analysis
Financial Decisions
Mathematics
Structural Analysis
Surveying
Securities
Statistics
Stress Analysis
Games
Home Management
Machine Design
Navigation
Real Estate
Thermal and Transport Science

Users' Library

The Users' Library provides the best programs from contributors and makes them available to you. By subscribing to the HP-41C Users' Library you'll have at your fingertips literally hundreds of different programs from many different application areas.

* Users' Library Solutions Books

Hewlett-Packard offers a wide selection of Solutions Books complete with user instructions, examples, and listings. These solution books will complement our other software offerings and provide you with a valuable tool for program solutions.

You can choose from:

Business Stat/Marketing/Sales
Home Construction Estimating
Lending, Saving and Leasing
Real Estate
Small Business
Geometry
High-Level Math
Test Statistics
Antennas
Chemical Engineering
Control Systems
Electrical Engineering
Fluid Dynamics and Hydraulics
Civil Engineering
Heating, Ventilating & Air Conditioning
Mechanical Engineering
Solar Engineering
Calendars
Cardiac/Pulmonary
Chemistry
Games
Optometry I (General)
Optometry II (Contact Lens)
Physics
Surveying

* Some books require additional memory modules to accomodate all programs.

TIME MODULE SOLUTIONS I

APPOINTMENT CALENDAR
WORLD TIME CONVERTER
EXERCISE MONITOR
AUTOMOBILE TRIP COMPUTER AND SPEED CALIBRATION
FOUR-CHANNEL CONTROLLER
LOGBOOK
''PLAYBACK'' PROGRAMMABLE TIMER
RANDOM SEED GENERATOR

THIS BOOK INCLUDES BARCODE FOR QUICK, EASY, ACCURATE ENTRY OF THESE PROGRAMS.

00041-90395 Printed in U.S.A.