HEWLETT-PACKARD

HP-41C

USERS' LIBRARY SOLUTIONS

TIMER SOLUTIONS I

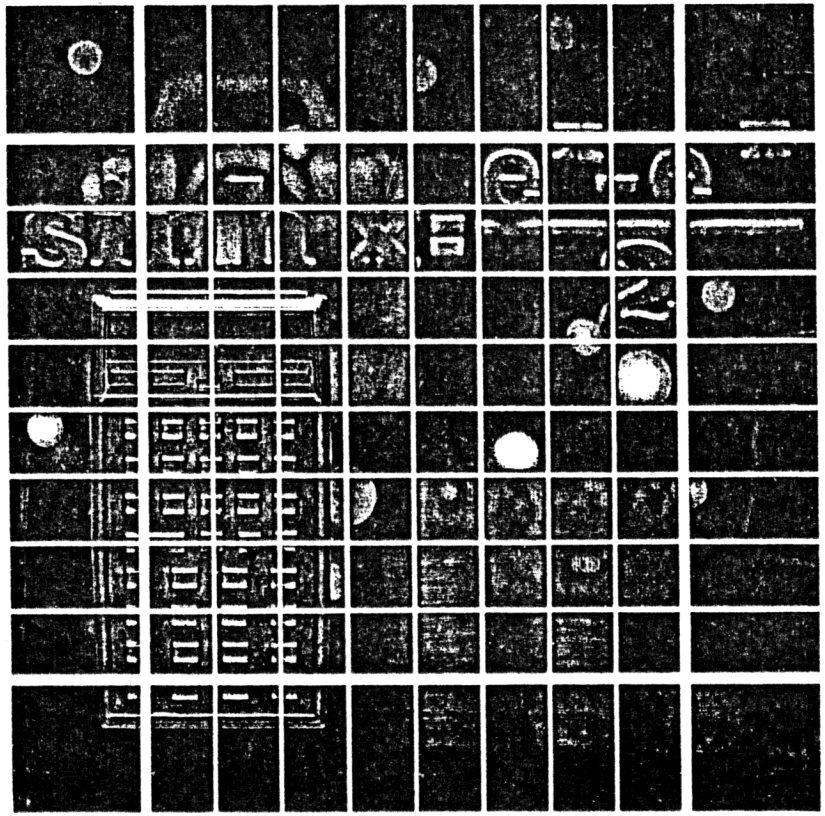


TABLE OF CONTENTS

APPOINTMENT CALENDAR

"Appointment Calendar" is an interactive program using the card reader or mass storage to create files of upcoming events. Entries may be added or deleted at any time and either viewed, printed or set as alarms either selectively or as a group. The program prints out a memo page calendar starting on any date. Using mass storage, the calculator can search the database automatically every morning and set the alarms for the day without the presence of the user. It will not stop even if the printer is turned off.

"Appointment Calendar" consists of three main routines: Initialize, Add Alarms and Read Alarms.

When the program is initialized, it prompts for the number of appointments in order to set up an index for incremental storage of data. The program attempts to execute the program "AMS" - Alarms - Mass Storage - to create the file on a digital cassette. With this method, users who do not require mass storage need not have the program occupying space in the calculator. Alarm files may be any length that is practical for the amount of RAM available, although a week of appointments is usually reasonable. Digital cassette files may be as short as one day due to the advantage of being able to name each file.

Once the file has been initialized by executing the global label "APPT", it may be added to at any time using the routine labeled by (C). Routine "ACAL" which is a portion of routine (C) is primarily for access by the mass storage routine, discussed below.

Two additional functions determine the way that the alarms and appointments will be accessed. Normally, each alarm is viewed for an input date followed by the prompt "SET ALM? Y/N". Alarms may be set at that time by responding "Y", Yes. If "N", No, the prompt "EDIT? Y/N" is given. An alarm may be edited or deleted in the file this way. The first register of the alarm record is set to zero so that it will not be accessed again. Using the function key [///] (d) causes the alarms for an input date to be viewed rapidly or printed when accessed if a printer is in the system. No prompts are given. Function [///] (e) causes all of the alarms in the file for an input date to be set immediately without prompting when accessed until the end of the file.

The routine labeled by 13, based on a routine by Roger Hill, will view requested items in the file without the halt normally encountered if Flag 21 is set and a printer is not present or turned off. With this facility, the user may have the database searched automatically on a pre-input date, without fear that the calculator will print only the first line or fail to set alarms because the printer was not left on Standby.

The program "AMS" - Alarms - Mass Storage has four section. "AMS is called only when the "Appointment Calendar" is initialized to create a data file of the necessary length on a digital cassette.

"SAVE" records the file when it is finished. It can be used whenever the file has some additional data.

"INW" - "Initialize Wake" - initially clears the view only mode and initializes the automatic mode to set all alarms for the input date. The file name is prompted for and the program reads the third register of the file to determine its length. If there is insufficient data memory allocated, the correct SIZE is prompted for. The user is prompted "SET ALL? Y/N". A "Y" response will set Flag 04 enabling the program to be run without

input/output halts. A control alarm is set to call the program "WAKE" on the input date and time followed by the creation of a temporary status file to save the status of Flag 04 and the File Name.

"LOAD" is a utility to download the data file from the digital cassette drive to the HP41's data registers. The routine prompts for "FL NAME?" and checks for sufficient size.

"WAKE" is a control alarm that "wakes" the calculator and HP82162A Printer, reads the status file to get the name of the appointment file, and loads the data file. The program calls "ACAL" in "APPT" and prints or sets all of the alarms for the input date. The Printer is then turned off as well as the Digital Cassette Drive. If the printer was not an HPIL type printer or was not left on standby, the alarms will still be set.

"AMS" actually consists of four independent utilities, all of which need not be in memory at the same time. Each routine may be terminated with an END statement. "WAKE" does not require any other routine, once initialized. The calculator may be used for other purposes after "Initialize Wake" has been run.

OPERATING LIMITS AND WARNINGS

At least six unused registers must be allowed in program memory for each alarm that is to be automatically set. Fewer registers are required for alarms with fewer than 24 character messages. See the Time Module Owners Manual, page ___ for the storage requirements of various alarm types.

Users with HP82180A Extended Functions/Memory Module may wish to include a size checking routine in the initialization portion of "APPT". Page ___ of the Owners' Manual demonstrates how this is done.

Suggestions and Modifications.

Users may wish to utilize the "WAKE" program to load programs or data that they may need on a particular day, such as for a class or assignment. "WAKE" can also be modified to read the main program, "Appointment Calendar" if desired. Using the PCLPS function in the Extended Functions Module the program can also be cleared and others read to replace it once the alarms are set. The routine labeled by "ACAL" may be used without the initialization portion of the program by placing an END after line 77 and deleting lines 78 through 82. The program can be considerably shortened by removing various options, including the mode toggles [///] (d) and [///] (e) if the program will always be used in the same way or called only by "WAKE" which contains the same options. Routine 13 may be removed only if the program is used with a Printer and the Printer is always left on or on Standby. It may be replaced with PRA. Flags 03 and 04 may be replaced by any User flags that will not be cleared until the program has finished its run.

Finally REGSWAP and REGMOVE may be used to sort the alarms before they are printed in strict order, rather than merely by morning and evening as is presently done. This will extend the length and run time of the routine, however.

DATA REGISTERS

00	Index
01	Date MM.DDYYYY
02	Index for Alarm storage
03	Index for recalling alarms
04	Index for date recall and compare (every sixth register)
05	File Name (ALPHA)
06	Time of alarm } saved after recall/view for setting alarms
07	Date of alarm }
08	Date of first alarm, MM.DDYYYY
09	Time " " " , HH.MMSSS
10	Alpha message
11	" " continued
12	"
13	"
14	each subsequent alarm requires six registers

FLAGS

00	Separate AM and PM alarms for memo page printout 00 Set=AM
01	EDIT mode/restore index when finished, RTN to "ACAL".
03	View and/or print alarms only
04	Set all alarms for input date automatically, also prints
NOTE:	if neither flag 03 nor 04 are set all alarms for input date are output followed by prompts before continuing
12	Print DOW banner double wide
21	Enable printer
23	Test message input
25	attempts PRA whether or not printer in system
55	Printer existence

USER KEYS and Function labels

APPT	Initialize file
(B)	Store appointments
(C)	View appointments, print or set alarms per flag status; EDIT
(///) (d)	Toggle View Only mode on or off
(///) (e)	Set alarms automatically - toggle on or off
AMS	Create appointment file on digital cassette
SAVE	Save appointment file on cassette
INW	Initialize "WAKE" program
WAKE	Control alarm to search data base and find appointments

Status.

Size (n (Alarms) * 6) + 8

TOTAL PROGRAM BYTES

APPT	538
AMS	199
WAKE	43

FIX 2, 4, 6

USER mode ON

To see how the "Appointment Calendar" works, we will create a data file that can be used with either the Card Read 82104A (with or without a printer) or with the HPIL Module 82160A and mass storage with a printer.

If you will be using the program with mass storage, you can follow the entire example. However, you should substitute a future date for the date given in the example or the alarms will immediately become past due, sound two tones and terminate the program run. A suggested date would be the date following the date that you are trying the sample problem.

Load the "APPT" program. If you are using mass storate GTO...and load the "AMS" program. GTO.. and load the "WAKE" program.

Appointments and reminders for the fourth week of January, 1982.

January 28	NICKS BIRTHDAY PRESENT?
"	DENTIST APPT
"	LUNCH WITH JIM B.
"	RENEW AUTO CLUB MEM
"	HILL ST BLUES/TV
January 29	MAKE LUNCH RES/NICK
"	RENEW LICENSE
January 30	TUNE UP PORSCHE
"	NICK 5 TODAY

PROMPT	INPUT	FUNCTION	DISPLAY
Initialize the program. [XEQ] "APPT"			
N APPTS?	9	[R/S]	You will need to SIZE your HP41C/V to at least 62.
If you are not using mass storage, skip the next line and continue running the example.			
FL NAME?	JAN/4	[R/S]	
DATE?	1.28 1982	[R/S]	
TIME?	-5	[R/S]	
MESSAGE?	NICKS BIRTHDAY PRESENT	[R/S] or [XEQ] (B)	It is not necessary to store all of the appointments at once. The file may have additions on another date until the limit of memory is reached.

DATE?	1.28 1982	[R/S]
TIME?	9.15	[R/S]
MESSAGE?	DENTIST APPT	[R/S]
DATE?	1.28 1982	[R/S]
TIME?	11.45	[R/S]
MESSAGE?	LUNCH WITH JIM B	[R/S]

Continue inputting the messages until the list is finished. If you are using mass storage you may wish to temporarily save the example file on a digital cassette to see how it will be used later. [XEQ] "SAVE".

If you are using the magnetic card reader and wish to save the data file, use WDTA or .062 WDTAX. Two blank cards are required. It is not necessary to save the file to run the example with either storage option.

To read the file. (C)

DATE?	1.28 1982	[R/S]

We could have chosen to output the list of appointments for the input date in either of three ways.

The "Normal" mode would view each alarm and prompt SET ALM? Y/N. In this interactive mode all would be viewed or printed but only selected records would be set as alarms.

The "View Only" mode is illustrated by the printer tape here. All alarms are immediately viewed or printed without I/O halts. The result is a printed memo page.

The third mode, "Auto", causes all alarms for the input date to be set whether or not they are also printed.

You may wish to try the "Normal" mode to see how it works.

If you run the example in the "Auto" mode, your calculator's memory will rapidly be filled with fictitious alarms. Refer to section ___ on page ___ of the Time Module Owners Manual to clear these examples.

```
------------
THURSDAY
1/28/1982
------------

9:15:00 AM
DENTIST APPT

11:45:00 AM
LUNCH WITH JIM B

------------

5:00:00 PM
NICKS BIRTHDAY PRESENT?

5:30:00 PM
RENEW AUTO CLUB MEM

10:00:00 PM
HILL ST BLUES/TV

------------
FRIDAY
1/29/1982
------------

11:00:00 AM
MAKE LUNCH RES/NICK

10:00:00 AM
RENEW LICENSE

------------
```

If you would like to try the "Wake" program, the steps are as follows:

	[XEQ] "INW"	
FILE NAME?	JAN/4	[R/S]
SET ALL? Y/N	N	[R/S]
DATE?	1.28 1982	[R/S]

If you are using mass storage, you can see how the system would "wake up" with your personal information file loaded and ready to start the day.

Initialize the program.

	[XEQ] "INW"		
FL NAME?	JAN/4	[R/S]	
SET ALL? Y/N	Y	[R/S]	If you had input "N", No, the alarms would be printed but not set.
DATE?	1.28 1982	[R/S]	
TIME?	7 (7 AM)	[R/S]	

A status file will be created on the digital cassette to preserve the size, file name and status of Flags 03 and 04. If your display reads RESIZE 62, you must SIZE the HP41 to 62 data registers and return to the first step, [XEQ] "INW". The status file will not be created.

If you have set your system to "WAKE" in a few minutes, you can set it to standby and turn the calculator off.

To see that the control alarm "WAKE" has been set:

[XEQ] ALMCAT	[R/S] when you see WAKE in the display.	

In a few minutes, the system will turn on and either print or set the alarms in the example for the date you have chosen. This is the way your "Automatic" system would work in actual practice.

You might like, at this point, to refer to the section heading "Operating Limits and Warnings" and "Suggestions and Modifications".

USER INSTRUCTIONS

SIZE: (n x 6)+8 (HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load "Appointment Calendar" and set USER.		[USER]	
2	Load "Appointment - Mass Storage" if you will be using the PIL option in the main program.		[GOTO] . .	
3	Initialize the program.		[XEQ] "APPT"	
4	Input maximum number of appointments to be in file.	n	[R/S]	n APPTS?
5	If you will not be using the mass storage (PIL) option or did not load the program "AMS" skip to step number 7.			FL NAME?
6	Input the File Name for the Appointment Period. Any descriptive ALPHA characters may be input up to a maximum of 6.	[ALPHA]	[R/S]	DATE?
7	Input the date of the appointments.	MM.DDYYYY	[R/S]	TIME?
8	Input the time of the first appointment. (The time may be input as a negative number to signify PM or in 24 hour time)	HH.MMSSS	[R/S]	MESSAGE?
9	Input your message, up to 24 ALPHA characters. If the alarms will not be printed and the message contains more than 12 ALPHA characters, you should bear in mind that when the alarm is triggered only the first 12 characters will be displayed until the alarm is acknowledged (message alarm only).	[ALPHA]	[R/S]	nn (display is random)
	If you do not wish to store a message with the alarm, press no ALPHA keys.	no input	[R/S]	
10	To store further alarms and appointments it is not necessary to store all of your appointments at one time. Other appointments may be added at any time until the last record (6 registers) is full.		[R/S] or [B]	DATE?
11	Recalling Appointments			
	Appointments may be recalled in either of 30 ways, interactively, View/Print only or automatically set.			
	If Flag 03 is set, alarms will be viewed and printed if a printer is in the system without any halts. If Flag 04 is			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY (SIZE: (HP-41C))
	set, all of the alarms for the requested date will be set without prompting. If neither Flag is set, all alarms for the input date will be presented interactively.			
12a	To save an appointment file on magnetic cards.		[RCL] 02 [FRC] [WDTAX]	
12b	To save an appointment file on mass storage.		[XEQ] "SAVE"	
	This operation should only be performed after all inputs are complete. "SAVE" is a utility and not a subroutine and does not return to the APPT program. Appointments may be added at any time and "SAVE" executed again to add the new data.			
13	To read a mass storage file into the calculator's data registers.		[XEQ] "LOAD"	FL NAME?
	Input the name of the data file.	[ALPHA]	[R/S]	
	If the display "SIZE=(nn)" is seen, "SIZE" the calculator and repeat step 13.			
14	Set and clear "View Only" mode.		[///] [d] [///] [d]	VIEW VIEW ONLY
15	Set and clear "Auto" mode to set all alarms.		[///] [e] [///] [e]	AUTO ON AUTO OFF
16	Recall Appointments for a given date.		[C]	DATE?
	Input date. If no input, the current date will be a default.	MM.DDYYYY	[R/S]	" DAY" "MM/DD/YYYY" "nn:nn:nn -M" SET ALM? Y/N
	This prompt would not be seen in the Auto mode (Flag 04 set). Input Y for Yes or N for No.	"Y" or "N"	[R/S]	EDIT? Y/N
	This prompt would only be seen if the response was N.	"Y" or "N"	[R/S]	...next alarm...
	"EDIT" allows the option of deleting an alarm from the file reflecting a cancelled appointment or error.			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY (SIZE (HP-41C))
	If your response to EDIT? Y/N was "Y"			DATE?
	You are positioned to the current alarm to delete	0	[R/S]	NEXT ALARM
	To change, input new date, time and/or message.			TIME? MESSAGE?
	In EDIT mode you must replace all 3 prompted for values.			
17	Use PIL module HP82160A to "WAKE" the system on a given date.			
	Initialize the program.		[XEQ] "INW"	FILE NAME?
18	Input the mass storage file name.	[ALPHA]	[R/S]	SET ALL? Y/N
	This allows the "Auto" mode option.	"Y" or "N"	[R/S]	DATE?
19	Input the Date of the filed appointments.	MM.DDYYYY	[R/S]	TIME?
	Input the time that the system is to "WAKE".	HH.MMSSS	[R/S]	
	If the prompt "RESIZE" (nn) is seen, SIZE nnn and return to step 19. Peripheral mass storage and printer-type devices must be left on standby with the controller, the HP41C/V turned off. The "WAKE" program and the "ACAL" portion of "APPT" must be in program memory. See operating limits and warnings.			

Program Listings

```
 01◆LBL "APP
T"
 02 CLX
 03 STO 01
 04 "N APPTS
?"
 05 PROMPT
 06 6
 07 *
 08 7
 09 +
 10 .1
 11 %
 12 7
 13 +
 14 STO 02
 15 SF 25
 16 XEQ "AMS
"
 17◆LBL B
 18 CF 01
 19 "END"
 20 ISG 02
 21 GTO 20
 22 PROMPT
 23◆LBL 20
 24 "DATE?"
 25 RCL 01
 26 PROMPT
 27 FS? 01
 28 GTO 00
 29 STO 01
 30◆LBL 00
 31 STO IND
02
 32 X≠0?
 33 GTO 00
 34 FS? 01
 35 GTO 30
 36◆LBL 00
 37 "TIME?"
 38 PROMPT
 39 -12
 40 X<>Y
 41 X<0?
 42 +
 43 ABS
 44 ISG 02
 45 STO IND
02
 46 "MESSAGE
?"
```

Prompt to determine size requirements and calc. index

Number of registers in each alarm record

Plus the number of scratch registers

/1000 to set upper limit

Places index in R02

SF 25 in case 'AMS' not in memory or needed Gosub Mass Storage Pgm

Store alarms

Prompt if end of data registers

If no input, repeat original date

Editing alarm?

Then date should not change

If 0 input, alarm will not be accessed again

EDIT mode? Then update index

Input time, HH.MMSSS

In case PM time input as negative number

If negative, convert to 24-hour time, 24-hour format required for sorting AM/PM alarms

Increment index, store time

```
 47 CF 23
 48 AON
 49 STOP
 50 AOFF
 51 FC?C 23
 52 CLA
 53 ISG 02
 54 ASTO IND
02
 55 ISG 02
 56 ASHF
 57 ASTO IND
02
 58 ISG 02
 59 ASHF
 60 ASTO IND
02
 61 ISG 02
 62 ASHF
 63 ASTO IND
02
 64◆LBL 30
 65 RCL 03
 66 FS?C 01
 67 STO 02
 68 CLX
 69 RTN
 70 GTO B
 71◆LBL d
 72 "VIEW"
 73 FC?C 03
 74 SF 03
 75 FC? 03
 76 PROMPT
 77 "ONLY"
 78 PROMPT
 79◆LBL e
 80 "AUTO 0"
 81 FC?C 04
 82 SF 04
 83 FS? 04
 84 "ON"
 85 FC? 04
 86 "OFF"
 87 PROMPT
 88◆LBL C
 89 SF 21
 90 "DATE?"
 91 DATE
 92 PROMPT
 93◆LBL "ACA
L"
```

Turn on ALPHA

Test message input

No input? Then CLA. When alarms are accessed 0.00 in data registers becomes 'message' if not alpha null

Store entire alpha register

Cycle back

View toggle

If flag 03 set, alarms are viewed/printed only without I/O halts if off

If on, auto set toggle

First portion of display identical

Toggle flag

On

Off

Alarm recall routine

Enable printer (printer existence tested in Label 13)

Date to be searched

Label for access by Mass Storage routine

Program Listings

```
 94 STO 01
 95 ADV
 96 FS? 55
 97 XEQ 12
 98 RCL 01
 99 DOW
100 XEQ IND
X
101 "DAY"
102 XEQ 13
103 LASTX
104 FIX 6
105 CLA
106 ADATE
107 XEQ 13
108 FS? 55
109 XEQ 12
110 CF 00
111 XEQ 07
112 ADV
113 FS? 55
114 XEQ 12
115 SF 00
116◆LBL 07
117 2
118 RCL 02
119 FRC
120 6 E-5
121 +
122 +
123 STO 04
124 CF 12
125◆LBL 08
126 ISG 04
127 GTO 09
128 CLX
129 CLD
130 RTN
131 RCL 01
132 1
133 DATE+
134 GTO "ACA
L"
135◆LBL 09
136 RCL 01
137 RCL IND
04
138 X≠Y?
139 GTO 08
140 RCL 04
141 INT
```

If printer, print dashed line

Get day of week ALPHA string

Completes string

Execute print if possible routine

Formats correct date display

Print?

If printer, print dashed line header for memo page

Index subroutine

Printer?

Set up index to recall every sixth register starting with first appointment record

Index for date recall and compare

End double-wide print

Get next date

Go to compare routine, if done, clear display

Increment date

Cycle to next on R/S and skip input prompt

Input date

Recall filed date

Not same, try next one

```
142 RCL 02
143 FRC
144 +
145 STO 03
146 12
147 ISG 03
148 1 E-8
149 -
150 RCL IND
03
151 STO 06
152 FC? 00
153 X<>Y
154 X<Y?
155 GTO 08
156 RCL 06
157 FIX 4
158 ADV
159 CLA
160 ATIME
161 XEQ 13
162 XEQ 11
163 5
164 ST- 03
165 CLX
166 XEQ 13
167 FS? 03
168 GTO 08
169 FS? 04
170 GTO 10
171 "N"
172 ASTO Y
173 "SET ALM
? Y/N"
174 AON
175 PROMPT
176 ASTO X
177 AOFF
178 X≠Y?
179 GTO 10
180 "EDIT? Y
/N"
181 AON
182 STOP
183 ASTO X
184 AOFF
185 X=Y?
186 GTO 08
187 SF 01
188 RCL 03
189 X<> 02
```

Set up index to recall appointment record

Record index

Save date and time for stack if alarm to be set

AM or PM, get AM appointments only on first pass

Format time in ALPHA

Print if possible

Get ALPHA portion of appointment record, 4 registers

Restore record index

Print if possible

If view only flag set, then skip (get next appointment)

If alarms to be auto-set, skip prompts

If not, then prepare to query

Store for compare in stack

If no, is alarm still valid?

Yes, valid

Not valid

Save R02 index

```
190 STO 03
191 XEQ 20
192 GTO 08
193*LBL 10
194 CLST
195 RCL IND
03
196 STO 07
197 ISG 03
198 RCL IND
03
199 STO 06
200 XEQ 11
201 0
202 RCL 07
203 RCL 06
204 XYZALM
205 GTO 08
206*LBL 11
207 CLA
208 ASTO Y
209 ISG 03
210 ARCL IND
 03
211 ASTO X
212 X=Y?
213 RTN
214 ISG 03
215 ARCL IND
 03
216 ISG 03
217 ARCL IND
 03
218 ISG 03
219 ARCL IND
 03
220 RTN
221*LBL 12
222 SF 12
223 "------"
224 ASTO X
225 ARCL X
226*LBL 13
227 SF 25
228 PRA
229 SF 25
230 FS?C 21
231 CF 25
232 FS? 55
233 GTO 14
234 AVIEW
235 PSE
```

Recall alarm parameters

Get ALPHA message

Get ALPHA message, will be nulls if no message

Double-wide print

Gets six more dashes (saves two bytes)

Print if possible

Maintain status of flag 21

If no printer, slow output to allow viewing

```
236*LBL 14
237 FC?C 25
238 SF 21
239 RTN
240*LBL 00
241 "SUN"
242 RTN
243*LBL 01
244 "MON"
245 RTN
246*LBL 02
247 "TUES"
248 RTN
249*LBL 03
250 "WEDNES"
251 RTN
252*LBL 04
253 "THURS"
254 RTN
255*LBL 05
256 "FRI"
257 RTN
258*LBL 06
259 "SATUR"
260 END
```

Reset flag 21 (would have caused halt on AVIEW if printer not present, or would print twice otherwise)
If printer, no need to slow output with AVIEW (scrolling)

Day of week strings for memo page header

```
01*LBL "WAK
E"
02 PWRUP
03 "WAKE-S"
04 READS
05 PURGE
06 CLA
07 ARCL X
08 CLX
09 SEEKR
10 READR
11 DATE
12 XEQ "ACA
L"
13 PWRDN
14 PWRUP
15 PWRDN
16 .END.
```

Turn on Printer and Digital Cassette

Read Status File

Get File Name from X-register

Position medium

Read file

Date of alarm input to ACAL routine

Called as subroutine so that Loop devices can be turned off when finished

Command sequence to rewind tape


```
01*LBL "AMS
"
02 XEQ 00
03 RCL 02
04 FRC
05 1 E3
06 *
07 1
08 +
09 CREATE
10 RTN
11*LBL "LOA
D"
12 XEQ 00
13 XEQ 01
14 CLX
15 SEEKR
16 RCL 02
17 READRX
18 RTN
19*LBL "SAV
E"
20 CLA
21 ARCL 05
22 CLX
23 SEEKR
24 RCL 02
25 FRC
26 WRTRX
27 RTN
28*LBL "INW
"
29 CF 03
30 SF 04
31 XEQ 00
32 "Y"
33 ASTO Y
34 "SET ALL
? Y/N"
35 AON
36 STOP
37 AOFF
38 ASTO X
39 X≠Y?
40 CF 04
41 FC? 04
42 SF 03
43 CLST
44 "DATE?"
45 PROMPT
46 "TIME?"
47 PROMPT
48 "↑↑WAKE"
```

Prompt for File Name

R02=ISG index number for appointment storage

calculate (n) registers

Add register 00

Prompt for File Name

Check size

Position medium

Download

Utility routine to transfer data from RAM to storage medium

Get File Name

Position medium

Index

'INITIALIZE WAKE'

Clear view only flag

Set auto flag as default

Get File Name

Yes

Auto Mode?

No?

Set view only mode to print alarms and disable interactive mode

Get alarm parameters

Control alarm

```
49 XYZALM
50 XEQ 01
51 CLST
52 RCL 05
53 "WAKE-S"
54 WRTS
55 RTN
56*LBL 00
57 "FL NAME
?"
58 AON
59 STOP
60 AOFF
61 ASTO 05
62 RTN
63*LBL 01
64 CLA
65 ARCL 05
66 CLX
67 SEEKR
68 2 E-3
69 READRX
70 1 E3
71 RCL 02
72 FRC
73 *
74 1
75 +
76 "SIZE="
77 FIX 0
78 ARCL X
79 SF 25
80 RCL IND
X
81 FC?C 25
82 PROMPT
83 .END.
```

Check if sufficient SIZE to read file

Get File Name in X

Name Status File

Write Status to preserve flag status (03 or 04), S ZE and File Name in X.

Input prompt

Get File Name

Position medium

R02 contains index for storage, fractional part reflects highest numbered data register required

Format SIZE prompt

Test for existence of highest numbered data register

Display if test failed

WORLD TIME CONVERTER

"World Time Converter" is a programmable alarm clock that will not only display the time and day for any cities that the user has keyed into program memory but also set a message alarm in the destination city's time. While it is useful to know the time in a foreign city, that time will often not correspond to business hours or other convenient calling or arrival time. The user inputs the time of day in the foreign city and an alarm is triggered in the users time zone that will correspond to the foreign time and flash the city's name for a message.

"WTIME" does a straightforward time conversion by comparing and adding the differences between the home city's time and Greenwich time and that of the destination city. The cities of interest to the user are keyed into program memory by reference to a chart and may take local ALPHA labels or Global labels for convenient key assignments. A branch is made to get the day of the week added to the time display and the city name saved for an alarm message, if needed.

"T2" prompts for the time of interest in the destination city and then sets an alarm that will trigger 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.

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, C.	-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, B.C.	-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		

Additionally, there are eight time zones in North America, Atlantic, Eastern, Central, Mountain, Pacific, Yukon, Alaska-Hawaii and Bering which are respectively -4, -5, -6, -7, -8, -9, -10 and -11 hours earlier than Greenwich time.

Reference.

DATA REGISTERS

00	time difference between input cities and Greenwich
01	Alpha string: first 6 characters of city name
02	last " " " " " "

FLAGS

21	Cleared if no printer in system to prevent halt on AVIEW
55	Printer existence

FUNCTION LABELS

WTIME	Calculates and displays time of day in foreign city
T2	Sets relative time alarm

STATUS

Size 03 Total Registers 28 + User Labels

Fix 2 USER Mode dependent on Label type

Operating Limits and Warnings.

The time adjustment chart does not take into account Daylight Savings time nor any other adjustments due to local laws and ordinances which must be known and input by the user of the program.

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.

It is assumed for the purpose of running this example that the user has keyed in the example cities in the printed program, Corvallis and Singapore. The actual answer will depend upon the time of day in which the example is run. If the output time is 15 and ½ hours later than Corvallis time, then the program has been run correctly.

PROMPT	INPUT	FUNCTION	DISPLAY
Get the home city.		[XEQ] "CORV"	CORVALLIS
Get the destination city.		[XEQ] "SING"	SINGAPORE
Get the time in Singapore.		[XEQ] "WTIME"	nn:nn AM or PM (DAY)

Select a time of day in Singapore that is 2 or 3 minutes later than the time of day on your clock for this portion of the problem.

You would like to be reminded to call Singapore at nn:nn_M their time.

		[XEQ] "T2"	
SINGAP. TIME?	Input a time	HH.MMSS [R/S]	

To verify that the alarm has been set, you may use the ALMCAT function. Press [R/S] when "SINGAPORE" is displayed and purge the alarm. The time of day that the alarm will be triggered is left in the X register.

You would like to include certain North American cities in your "World Time Converter" program. The cities are Willmette, Illinois, Mexico City, D.F. and Oakland, California.

Willmette, Illinois is in the Central time zone, Mexico City, D.F. is also in the Central time zone and Oakland, California is in the Pacific time zone.

The program steps to insert will be as follows:

```
GTO . 0 8 5
(PRGM)          LABEL "WILM" -6
                "WILLMETTE" GTO 08
                LABEL "MEXDF" -6
                "MEXICO, DF" GTO 08
                LABEL "OAK" -8
                "OAKLAND" GTO 08
```

USER INSTRUCTIONS

SIZE: (HP-41C) 3

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the Program and set USER mode. If you have not yet programmed the cities that you require, go to step ____		USER	
2	Get the home city.		[XEQ] "(CITY)"	CITY NAME
3	Get the destination city.		[XEQ] "(CITY)"	CITY NAME
4	Calculate the time of day in the destination city.		[XEQ] "WTIME"	nn:nn_M (DAY)
5	Set an alarm in the home city's time that will trigger at a time in the destination city.		[XEQ] "T2" or [R/S]	(CITY). TIME?
	Input the time of day in the destination city that you require.	HH.MMSSS	[R/S]	nn.nn (Alarm Time in X reg.)
6	Program cities of interest Any local ALPHA label may be used [///] (a) through (J) or any unused Global Label to facilitate key assignments. Label _____(descriptive or local ALPHA). Key into program memory a number taken from the chart on page ___ or page ___ including the sign if any. Key in the name of the city using no more than 12 ALPHA characters. Key in as the last line for each city: GTO 08 These labels must be keyed into the main program and not as independent routines. They should follow line 85 and must terminate with GTO 08.		"ALPHA"	

Program Listings

```
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 numbers
05 ABS
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 subroutine to get clock and date in stack
15 CLA
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       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 time, to be consistent with Tim Module input formats
31 -12           If negative, convert to 24-hour format
32 X<>Y
33 X<0?
34 +
35 ABS
36 24            Allows routine 10 to calculate 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 calculate time offset and leave alarm parameters in X
47 HMS+
48 ENTER↑
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 "COR
V"
87 -8            Corvallis is eight hours earlier than Greenwich time
88 "CORVALL
IS"
89 GTO 07        Go to display routine
90◆LBL "SIN
G"
91 7.3           7½ hours later than Greenwich time
92 "SINGAPO
RE"
93 GTO 07
94◆LBL "GNV
A"
```

Program Listings


```
 95 1                    1 hour later
 96 "GENEVA"
 97 GTO 07
 98*LBL "HAI
FA"
 99 2                    2 hours later
100 "HAIFA"
101*LBL 07
102 FC? 55
103 CF 21               If no printer, clear printer
104 AVIEW                 enable flag to prevent
105 ASTO 01               halt
106 ASHF                Save city name for
107 ASTO 02               messages
108 .END.
```


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. It can also time a warm-up period and overall time limit. Runners can input the various distance markers for any course and overall time goal for the course and alarms will signal when each marker should be reached to remain on target. Splits may be stored and later "replayed" and compared to goals. The course is easily set up before the exercise period and remains in the HP41C till 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 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, Flag 01 is set and will trigger a fifteen second pulse timing interval at the start and end of the course, followed at the end by another pulse count in five minutes to monitor recovery. Warm-up periods and time limits are also established during the input phase. Any option that is not wanted can be skipped by pressing [R/S] with no input. The user is prompted for their course goal, if any, and the successive distances of the various course legs are input. The goal time is divided into segments that will trigger a control alarm, "M", explained in paragraph 7 below, at the moment that the distance marker should be passed to remain on target. When all distance markers have been input, the number of splits desired is prompted. The utility of storing splits is best realized by having a coach or friend take the splits.

The initialized program comes to "rest" at Label "GO" which has a global label to facilitate a key assignment, if desired. "GO" will trigger the warmup period timing, initial pulse count, if chosen, and time limit alarm.

"SPLIT" uses the programmable stopwatch command to store splits in successive registers by incrementing a pre-limited counter in register 12. If storing splits was chosen as an option during the input portion of the program, the marker alarms are very short to allow 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 triggered 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 a display of the marker number. If the program were running 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 or paused.

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 using the PRPLOT program in the HP82143A Printer and HPIL Module 82160A. Using this feature, you would be able to plot goals against actual performance allowing any arbitrary distance increments to represent the X Axis and %CH to be the Y Axis with the center of the paper representing exact compliance with the goal. It would then be possible to see a visual graph of performance, where you slowed down, ran too fast or any segment that needs work.

DATA REGISTERS

00	Index for storing pulse counts .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
09	"WARMUP"
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
17	Warmup period
18	Loop control for "M" control alarm
19	Store distance marker intervals
20	"
21	"
22	"
23	
24	Splits begin after last "M" register
25	

FLAGS

01	Set pulse alarm and wait interval
02	toggle set/reset of "FIN" control alarm for pulse intervals
05	store splits, if set skip marker display, shorten tones to 2
21	enable printer, set and cleared when necessary to avoid output halt
25	error detect, used to check for sufficient size to store splits
26	enable tone signalling
27	User mode
29	formats output display without decimal points
55	Printer?

FUNCTION INDEX

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

Status.

SIZE n Distance markers + n splits + 18 PROGRAM REGISTERS 83

FIX 0, 2, 4, 6 USER mode ON

To see how the program works without running it for an excessively long time, we shall simulate the course of training undertaken by Wonder Woman on her secret island, who, with her HP41C to time her goals and performance, desires to run an irregularly marked course of ten kilometers in six minutes.

PROMPT	INPUT	FUNCTION	DISPLAY
Initialize the program.			
	"Y"	[R/S]	
WARMUP HMS?	Input 20 seconds .002	[R/S]	(.MMSS)
LIMIT HMS?	Not needed	[R/S]	
GOAL HMS?	6 minutes .06	[R/S]	(.MMSS)
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?	no input	[R/S]	
N SPLITS?			
To see the full marker display for this example, we must allow at least one split.			
	1	[R/S]	READY

In actual practice, a runner desiring to run a ten kilometer course, for example, in 00:40:00 minutes would input the distances at known points, or at arbitrarily spaced or even points, till the end of the course. The distance might have been paced by automobile odometer or other means if not an actual track. If we were to recall the registers, starting with R19, where Wonder Woman's distance markers were stored, we would see how they were divided into linear time intervals.

R19 = 00:00:36.00
R20 = 00:01:30.00
R21 = 00:02:31.20
R22 = 00:04:12.00
R23 = 00:05:24.00
R24 = 00:06:00.00

When ready to begin [R/S] or [XEQ] "GO"

The first display seen will be "PULSE", signaling the start of a 15 second interval to be used for taking a pulse count. After 15 seconds, a message alarm is triggered, BEATS=?. Acknowledge the alarm by pressing (←) and input, for example, 17 beats counted.

17 [R/S] RATE=68

WARMUP /

A warmup period will be timed when [R/S] is pressed. (Allow time to prepare).

[R/S] WARMUP

At the end of the warmup period another message alarm will trigger, displaying READY. Acknowledge the alarm by pressing (←) or any key.

READY [R/S] 0

As each marker should be passed, a series of 8 high-pitched tones will sound, followed by a display of

MARK (n)

Let this example run till completion at 6 minutes. After the display MARK 6 is seen,

[XEQ] "FIN"

In practice, you might like to select key assignments for these functions, or using Extended Functions/Memory Module HP82180A, modify the program by inserting PASN at appropriate points during initialization of the program.

WAIT

"WAIT" is displayed for 15 seconds followed by PULSE

BEATS=? 29 [R/S]

After 04:45 minutes the WAIT and PULSE alarms are repeated, to allow you to monitor pulse recovery.

Replay the stored pulse counts. No splits have been stored or they would also be replayed.

```
[XEQ] "REPLAY" PULSE 1=68
[R/S]          PULSE 2=116
[R/S]          PULSE 3=(n)
[R/S]          00:00:00.00 (no split)
```

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. Flag 21 would be cleared and have no effect on the displays.

USER INSTRUCTIONS

SIZE: m+s+18 * (HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the "Exercise Monitor" ("RUN")		USER	
2	Initialize the program.		[XEQ] "RUN"	PULSE? Y/N
3	If you require a timed period to check your pulse. If not required	"Y" "N" or	[R/S] [R/S]	WARMUP HMS?
4	Input warmup period length, or skip if not needed.	HH.MMSS	[R/S]	LIMIT HMS?
5	Input the maximum amount of time you want the exercise period to last. May also be skipped if not applicable.	HH.MMSS	[R/S]	GOAL HMS?
6	Input the course goal, total time for the complete course. Skip if n/a.	HH.MMSS	[R/S]	N SPLITS?
7	Input the number of splits you anticipate storing. If skipped, distance marker output display will be shortened to allow greater accuracy of splits. To see the full output display, input at least one split.	n splits	[R/S]	READY
8	When ready to begin, if the machine has not been touched. Else,		[R/S] or [XEQ] "GO"	
9	Pulse option. When "PULSE" display is seen, take a pulse count until message alarm is displayed.			PULSE BEATS=?
	Input count.		[R/S]	RATE= (nn)
10	(This display will not be seen if this option was not chosen).		[R/S]	WARMUP
11	When ready to begin Warmup period. Acknowledge the alarm.		[R/S] (←) or any key	WARMUP
12	Ready to start the Exercise period.		[R/S]	0
13	To store splits. (Must be assigned to a USER key to be effective).		[XEQ] "SPLIT"	Audible feedback; .nnnn

* markers + splits + 18

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: (HP-41C) DISPLAY
14	During the exercise period, if the Goal option was chosen, there are two possible outputs.			
	Split option. Feedback at the check-point or distance marker consists of 2 high pitch tones and a display.			0 or 0.0000
	No splits. Feedback consists of 8 high pitch tones and an ALPHA display.			MARK (n)
	No acknowledgement is required.			
15	End of exercise period. A USER mode key assignment would also be helpful to access this function.		[XEQ] "FIN"	
	There are two possible results of this action.			
	No pulse option chosen.			Two tones 0 or 0.000
16	Pulse option.			WAIT
	A 00:00:15 second interval is timed. Prepare for a pulse count. No acknowledgement is necessary.			PULSE two tones
	Take a pulse count till the alarm is heard. Acknowledge the alarm.		(←) or any key	BEATS=?
	Input count.	n beats	[R/S]	RATE= (nn)
17	To read back splits and/or pulse.		[XEQ] "REPLAY"	PULSE n= (nn) .... nn:nn:nn.nn .... till end
	Determining the SIZE requirement. The data size required is 18 + n splits + n distance markers. Alternatively, you may choose to let the program calculate the required minimum size. If a prompt of RESIZE = (nnn) is seen, RESIZE the HP41C and press [R/S] to continue. If you have an HP82180A Extended Function/ Memory Module, see page ___ of the Owners Manual for the method of having your program check and set its own size.			

Program Listings

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

Program Listings

```
 92 "RESIZE>
="
 93 FIX 0
 94 SF 25
 95 RCL 12
 96 +
 97 1
 98 +
 99 ARCL X
100 LASTX
101 -
102 STO 04
103 RCL IND
X
104 FC? 25
105 PROMPT
106 FC?C 25
107 GTO 02
108 RCL 04
109 RCL 05
110 *
111 ST+ 12
112 RCL 12
113 STO 06
114 SF 27
115 FIX 4
116 "READY"
117 ASTO 10
118 RCL 10
119 RTN
120*LBL "GO"
121 CF 02
122 FC? 55
123 CF 21
124 SF 26
125 FS? 01
126 XEQ 05
127 FS? 01
128 RTN
129 SF 02
130 RCL 11
131 STO 13
132 CLST
133 RCL 17
134 X=0?
135 GTO 03
136 "WARMUP
↑"
137 ASTO 09
138 PROMPT
139 CLA
```

Test for sufficient size, to ensure that program will not fail to store required n splits
Format prompt
Set error Flag

Compute size requirement

Format ALPHA display if needed

SIZE

Test if register exists by attempting a recall. A recall would not place unwanted value in register.
If test failed, then display prompt

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

Pressing R/S will cycle back and continue

Store index

Split index
Set fix mode to display time

Save prompt for reuse
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 at proper times. If no printer (likely in this application) Clear Flag 21 to prevent halt on output (AVIEW)
Set tone Flag for feedback messages at goal markers
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
Prepare stack for alarm command
No warmup required? Then skip
Prompt: warmup period next
Save display to place in X during warmup act.
Initialize ALPHA

```
140 ARCL 10
141 TIME
142 HMS+
143 XYZALM
144 TONE 9
145 TONE 9
146 RCL 10
147 CLA
148 ARCL 09
149 AVIEW
150 RTN
151*LBL 03
152 RUNSW
153 CLST
154 SETSW
155 TIME
156 TONE 9
157 TONE 9
158 STO 15
159 RCL 16
160 X=0?
161 GTO 04
162 CLA
163 FIX 4
164 ATIME24
165 HMS+
166 XYZALM
167*LBL 04
168 RCL 14
169 X≠0?
170 GTO 20
171 RTN
172*LBL "SPL
IT"
173 RCLSW
174 FIX 4
175 TONE 9
176 TONE 8
177 ISG 12
178 STO IND
12
179 RTN
180*LBL "FIN
"
181 STOPSW
182 TONE 9
183 TONE 9
184 FC? 01
185 RTN
186 CLST
187 TIME
```

Get 'READY' string
Time offset for alarm

Message alarm set

Feedback: start activity displayed

Get 'READY' string in X
Get 'WARMUP' string in ALPHA and display. When alarm is acknowledged, the message will remain in X
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

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

Program Listings

```
188 RCL 07
189 HMS+
190 "↑↑"
191 XYZALM
192 "WAIT"
193 PROMPT
194*LBL 05
195 FC? 55
196 CF 21
197 "PULSE"
198 AVIEW
199 "BEATS=?
"
200 CLST
201 RCL 07
202 TIME
203 HMS+
204 TONE 9
205 TONE 9
206 XYZALM
207 .043
208 HMS+
209 "↑↑FIN"
210 FS?C 02
211 XYZALM
212 CLX
213 STOP
214 4
215 *
216 ISG 00
217 STO IND
00
218 "RATE="
219 FIX 0
220 ARCL X
221 AVIEW
222 RTN
223*LBL "REP
LAY"
224 SF 21
225 1.003
226 STO 00
227 FIX 0
228*LBL 06
229 "PULSE "
230 ARCL 00
231 "⊢="
232 RCL IND
00
233 ARCL X
234 X≠0?
```

+ 15 seconds (stored at initialization)

ALPHA alarm command

Display required activity

No printer?

Be sure AVIEW will not halt program

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=?'

Multiply pulse count by 4 to get rate/minute
Save pulse count
R01, 02, 03

Display

Recall splits and pulse counts

Enable printing or force halt on output display, AVIEW
Index for pulse count
Display significant digits only

Index number

Display only if not 0

```
235 AVIEW
236 ISG 00
237 GTO 06
238 RCL 06
239 1
240 +
241 FIX 6
242 STO 04
243*LBL 07
244 RCL IND
04
245 CLA
246 ATIME24
247 AVIEW
248 ISG 04
249 GTO 07
250 CLX
251 FIX 2
252 RTN
253*LBL "M"
254 8
255 FS? 05
256 GTO 08
257 TONE 9
258 TONE 9
259 GTO 20
260*LBL 08
261 TONE 9
262 DSE X
263 GTO 08
264 FIX 0
265 "MARK "
266 RCL 13
267 19
268 -
269 INT
270 ARCL X
271 CF 21
272 AVIEW
273*LBL 20
274 CLST
275 RCL 15
276 RCL IND
13
277 HMS+
278 "↑↑M"
279 ISG 18
280 XYZALM
281 CLX
282 ISG 13
283 END.
```

Display and get next, do not stop if printer attached
Number of splits

Time format to ALPHA
Recall index

Get splits

Into ALPHA
Print, or stop and get next

Leave display clear when done
Control alarm entry point
If splits option chosen, two two quick tones only

Set up next control alarm for marker
Tone loop, sound eight high pitch tones

Format ALPHA display

Get number of marker

AVIEW must not halt program or next control alarm will not be set
Two byte label because it is also called earlier in program
Set up stack for alarm
Start time
Next goal point

Control alarm
Loop control

Leave in X

Increment goal register

AUTOMOBILE TRIP COMPUTER and SPEED CALIBRATION

"TRIP" and "CAL" are two programs designed to work together to perform timing functions on automobile trips. "TRIP" may be used without "CAL" though local label functions [///] (c), [///] (d) and [///] (e) require data calculated by the "CAL" program. Users of the "TRIP" program can calculate their estimated time of arrival and required speed to a planned destination. The program has routines for setting periodic alarms, converting tachometer RPMs to speed in a given gear and correcting a speedometer reading. Alarms may be set, cleared, changed or merely silenced at any time. A feature of the program is its ability to be interrupted and restarted as often as needed. Either the HP82180A Extended Functions/Memory Module or Card Reader 82104A may be used to save data needed to restart the program. 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 "blocks" and may easily be extracted or modified for other use. There are only two subroutines in the program, an ALPHA prompt and a version of the "ALMREL" program in the Time Module Owners' Manual, page ___. Its use prevents data errors when alarms are set relative to the current clock time and produce times greater than 24 hours.

Label (A), the Chime routine, sets a periodic tone signal that may be used for signaling or as a keep alert signal on monotonous stretches of highway. To maintain a constant interval between alarms, the interval is added to the last alarm time rather than the clock time, negating the time lost by label searches and the alarm calculations themselves. Like all of the alarms in the program, the chime alarm is always set by the program and stored in the alarm stack without an automatic reset. The chime may be set at initialization or bypassed and set later. It may also be changed or cancelled. When a change is desired, the existing in the alarm stack must be cleared. To enable the program to perform that task, the clock is temporarily advanced using the T+X function to immediately before the alarm triggers. When the chime triggers it encounters a disable flag and returns during a pause. The clock is then retarded to normal time. This routine assumes that a great number of alarms will not conflict with the chime. Finally, when the alarm is activated, it first saves X, Y and Z in storage registers and later restores them to prevent calculations in progress, if any, from being disturbed.

Label "2" is set as a control alarm at the start of the program to keep tract of 100s of hours when the stopwatch rolls over. Because it is set whenever the program is initialized, false starts should be avoided. Superflous alarms may be cleared manually by using the ALMCAT function, page ___ of the Time Module Owners' Manual.

Label (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. Label (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!

Label [///] (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 100ths of a second may gradually be introduced. The Calibration program, "CAL", repeats the routine to allow the two programs to be used independently.

Label (E) is a D=RT calculation to estimate time of arrival and format the output in clock time; a date display is also seen if different from the current date. A straight line calculation will give fairly credible results for typical interstate highway travel. Inputs can be in either miles or kilometers as long as they are consistent. The program treats them only as units and does not convert or name them to allow easier input and faster calculations. If you wish to add conversion routines to this program, two conversion factors may 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 calculates a vehicle's speed accurately by translating engine revolutions in a given gear to miles per hour. 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 a vehicle. Tachometers on manual transmission vehicles are generally a more reliable and linear gauge of speed than speedometers. However, speedometers may also be calibrated by the program but commonly have non-linear errors. A correction factor calibrated at 50 miles per hour may be virtually useless at 35 miles per hour depending upon the vehicle and individual gauge. 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 hundreths of a second from mile to mile to be reliable. If this is of importance, routine might be added to convert the different splits to decimal form and accumulate them in a statistical block of registers (REG =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 and the program only counts from Ø through 10. Registers 05 through 10 serve primarily as an overflow bugger though they may also be used for additional splits such as on a ten mile odometer calibration run in a Time and Distance Rally.

When the running program executes line 39, the calculator is put into SW mode and the keyboard is redefined to 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 the stopwatch mode is exited by manually pressing [///] (←) the program will resume normal execution and trigger past due Control Alarms, if any, that may have come due but could not execute during the single line, SW. If the chime routine were set to a very 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 reset. This inconvenience is rarely encountered with Control Alarms.

"CAL" does not use flags or registers that are used by "TRIP" although 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.

DATA REGISTERS

00 - 10	Split and overflow buffer
11	save X
12	save Y
13	save Z
14	Counter, greater than Ø 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 occurence: 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, " " " "
31	gear 5, " " " "

FLAGS

03	gear 3
04	gear 4
05	gear 5
07	cancel/disable Control alarms, "CS" and "T-"
08	chime alarm is set, used by Reset routine
09	calibrate tachometer
10	Time-out in progress, enable and maintain, not cancelled by 07
21	printer enable
26	clear tone flag, first pass through chime routine
55	Printer existence

Flags 00-07 may be used during the INSTAT command, if HPIL is used without disturbing program operation.

USER KEYS

(///) (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

INSERT

22	data entry, tachometer calibration

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/hour.

PROMPT	INPUT	FUNCTION	DISPLAY
Load the "TRIP" program. GTO . . Load the "CAL" program. Set size 32.			
Initialize the program. [XEQ] "TRIP"			
ODOMETER?	Input odometer at start of trip. 16000	[R/S]	
COREX?	Correction is presently un-known. No input.	[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.			
	(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 problem and the time elapsed since the beginning of the trip. This example assumes that the trip was started at 7 AM and the answer will be close to ½ hour later than the start time.

You have been driving 25 minutes and wish to know your projected arrival time.

	(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 10/15

If you input a fictitious distance, the calculator may assume you are traveling at unusually high speed.

You would like to stop at the Golden Arches.

(B)		TIME OUT ... TIME OUT

Later, when you return to your vehicle:

(B)		RESTART

How long have you been driving?

(C)		DRIVE=00: (minutes)

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

(A)		RESET ...
CHIME/HMS?	Input any interval or continue to next operation. No keys need to be pressed to ignore prompt.	

If the chime is not being reset, you might press (J). The alarm would be triggered once, but not heard or reset in this case.

Use the "Calibration" program.

(F)
Use of the (F) key 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 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 stored at mileposts.

Press [///] (←) to exit SW mode and resume program execution.

If either of the Control Alarms, Chime "CS", or Time-Out "T-" came due it would be triggered 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/hour for every 1000 RPMS in fourth gear.

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

What should your speedometer indicate if you wish to drive exactly 55 miles/hour?

55 [///] (e) INDIC=50.8

When the speedometer indicates 50.8 miles/hour, you are actually travelling at 55 miles/hour.

[///] (c)

GEAR	4	[R/S]	
SPEED	No input	[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.

Operating Limits and Warnings

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

If any alarms will be triggered before the chime, with the exception of Time-Out, the label (A) reset routine will trigger them prematurely. Before resetting the chime, use the ALMCAT mode of Time Module operation to purge unwanted chime alarms. Flag 08 should be clear. In most cases, the reset routine will clear the chime without disturbing other alarms.

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

"CAL" is written to calibrate gauges to miles/hour. For kilometers a conversion constant, 1.61, must be added.

Status:

SIZE: 32 TOTAL REGISTERS: Min. 66, "CAL" only
114, "TRIP" only
Max.148, both programs

FIX: 1,2,4 USER mode ON

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
				SIZE: (HP-41C) 32
1	Load "TRIP" program and set USER mode		USER GTO . .	
2	Load "CAL" program.			
3	Execute "TRIP" only immediately prior to start of planned trip.		[XEQ] "TRIP"	ODOMETER?
4	Input current odometer reading. NOTE: Units of distance may be in either miles or kilometers and must be consistent.	distance reading	[R/S]	COREX?
5	Input speedometer correction factor if known.	factor	[R/S]	DISTANCE?
6	Input distance to destination.	units of distance	[R/S]	CHIME/HMS?
7	If a periodic tone signal is desired, input time period.	HH.MMSS	[R/S]	0.00
	The following functions may be performed at any time except as noted.			
8	Calculate speed required to reach destination in (n) hours.		(D)	ODOMETER?
	Input odometer reading.	units of distance	[R/S]	ARRIVE HMS?
9	Reset Stopwatch. If the trip timer maintained by the stopwatch must be used temporarily for another purpose, reset the stopwatch when ready, allowing for interruption.		[///] (a)	RESET SW
10	Chime change or set.		(A) [R/S]	
	The calculator will execute the ALMCAT function to display alarms if any. Purge any alarm displaying the message "//CS". Refer to Time Module Owners Manual, page ____.			RESET . . .
				CHIME/HMS?
	Input period of tone signal desired.	HH.MMSS	[R/S]	HH:MM:SS
11	To stop the trip timer and initiate a rest period of any duration.		(B)	TIME-OUT ... TIME-OUT

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
				SIZE: (HP-41C)
12	To restart trip timer.		(B)	RESTART
13	Calculate total driving time. Note that time-outs are not accounted for until restart routine accumulates them.		(C)	DRIVE=nn:nn
14	To disable periodic "TIME-OUT" display and chime and prevent reset.		(J)	CANCEL
	Note that TIME-OUT, if in progress, must be restarted to update time accumulator.			
15	Estimated time of arrival.	units of distance	(E) [R/S]	ODOMETER? nn:nn AM (or PM) MM/DD
	Display of date, MM/DD, not seen if same as current date. Input distance may not equal start distance or DATA ERROR will be generated. Rate cannot be calculated without distance ≠ 0.			
16	USING THE "CALIBRATION" PROGRAM.			
	If called as a subroutine from "TRIP". If used independently, [XEQ] "TRIP".		(F)	TACH RPM?
	Input tachometer RPM, revolutions per minute as read on gauge. No input if not applicable.	RPM	[R/S]	GEAR? 3/4/5
	Input gear, third, fourth or fifth.			
17	If "TACH RPM?" input bypassed.			SPEED?
	Acknowledge by pressing any digits. If both prompts are ignored, routine is repeated.		[R/S]	
18	Operate Stopwatch. To use this mode, you may wish to review page ___ of the Time Module Owners' Manual.			
	To start Stopwatch.		[R/S]	(running SW)
	To store a split at a mile marker. Up to 11 splits may be stored. No programmable test is possible in SW mode		ENTER	

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY (SIZE: (HP-41C))
	to test and limit number of inputs, so avoid taking extra splits.			
	To exit SW mode and continue running program.		[///] (←)	
	Past due control alarms, if any, will be run immediately. "TIME-OUT" and chime HH.MMSS may be seen.			
	Alternatively, calculator may be turned off and on again and [R/S] pressed.		[R/S]	S/COREX= (n.n)X 1K RPM, (n)=(n)
	Display signifies: One thousand RPMs in input gear = (nn.n) miles/hour.			
19	To convert between RPM and speed.		[///] (c)	GEAR?
	Input gear.	3/4 or 5	[R/S]	SPEED?
	If speed unknown		[R/S]	RPM?
	Input RPM.	RPM	[R/S]	SPEED= (n.n) or RPM= (nnnn)
20	To calculate actual speed for an observed speedometer reading.	speed	[///] (d)	ACTUAL= (n.n)
21	To calculate speedometer reading that will give actual speed.	speed	[///] (e)	INDIC= (n.n)
22	To suspend program temporarily.			
	Using HP82180A Extended Functions/Memory Module.		ALPHA "TRIPED" 32 CRFLD .031 SAVERX	ALPHA
	To restore data.		ALPHA "TRIPD" Ø SEEDPTA .031 GETRX	ALPHA
	Using card reader HP82104A	.031	WDTAX	
	Refer to Flag table, page ___ to see effect of various flags upon program operation.			

Program Listings

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

```
  98 ENTER↑
  99 RCLSW          Get trip elapsed time and
                    adjustments for 100s of
                    hours and time-outs
                    totalled
 100 RCL 21
 101 HMS+
 102 HR             Calculate rate
 103 /
 104 RCL 26         Planned distance
 105 R↑
 106 -
 107 X<>Y
 108 /
 109 HMS
 110 TIME           Add projected time to
                    clock time
 111 XEQ 04
 112 FIX 2          Execute routine to get
                    time and date in X and
                    Y
 113 CLA
 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
                    ALPHA
 121 AVIEW
 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
                    previous adjustments
 132 FC?C 10        Time-out mode?
 133 GTO 02
 134 RCL 27
 135 HMS-           If Flag 10 clear, then ini-
                    tiate a time-out, else
                    adjust accumulator
 136 RCL 21         register
 137 X<>Y
 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--
                    reminder
 152 XEQ 04         Set up alarm parameters
                    in stack
 153 XYZALM
 154 FS? 07         If alarm 'silenced' quit
                    here, if triggered by
                    chime reset, then reset
 155 RTN            alarm to next occurrence
 156 XEQ 03         omit display this time.
 157*LBL 03          Repeat display and tone
 158 "TIME OU       twice.
T"                  Reminder routine
 159 AVIEW
 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
 166 TIME           Accumulator register
 167 XEQ 04
 168 XYZALM
 169 RTN            Next hundred
 170*LBL 04
 171 HMS+           ALMREL type routine
 172 ENTER↑         similar to Timer Module
 173 ENTER↑         Owner's Manual, here
 174 24             without I/O halts
 175 /              Number of days
 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
 191 DATE           Reset stopwatch
 192 RCL 18
 193 DDAYS
 194 24             Date stopwatch started
 195 *              running
 196 TIME           n days running
                    n hours
```



```
 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
 205 "RESET S       error
W"                  Confirm
 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
 221 +              of gear data
 222 "SPEED?"       Calculate RPM if speed
 223 0              known
 224 PROMPT         Set to 0 for input test
 225 X≠0?           No input, try next
 226 GTO 05         If input speed, go to
 227 "RPM?"         calculate routine
 228 PROMPT         Calculate speed from
 229 RCL IND        known RPM?
Z                   Recall gear data
 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
 249 ARCL X         digit
 250 AVIEW
 251 RTN
 252*LBL J
 253 SF 07          Set Flag to disable chime
 254 "CANCEL"       at entry point and silence
 255 AVIEW          Time-out, if any, after
 256 RTN            resetting
 257*LBL F
 258 FC? 55         Call global subroutine and
 259 CF 21          return here
 260 XEQ "CAL       Printer?
"
 261 .END.
```

```
 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 redafined 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.)
                   Reset stopwatch
 51 10
 52 STO 20         Store index, register ad-
 53 9                dress of highest allowed
 54 STO 19           split
 55 CLX            Store index, register ad-
 56 STO 14           dress of lowest allowed
 57 STO 15           split
 58*LBL 02         Set counter to 0
 59 RCL IND
20
 60 RCL IND        Calculate delta splits
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, ret
                     or stop
```



```
 95 STOP           Stop if no printer, display
 96 60               register unchanged
 97 *
 98 RCL IND        Recall gear RPM
16
 99 1 E3           Convert to 1 RPM
100 /
101 /
102 "1K RPM/
"                  Format output display:
103 FIX 0            1,000 RPM in (n) gear
104 RCL 16           = (n)/hour
105 26
106 -
107 ARCL X         Get gear number into
108 "*="             ALPHA
109 FIX 1
110 CF.IND X
111 ARCL Y         Clear gear Flag
112 X<>Y
113 .1
114 %              Divide by 1,000
115 STO IND
16                 Store in gear (n) register
116 AVIEW
117 .END.
```

FOUR CHANNEL CONTROLLER

"Four Channel Controller" provides a means of maintaining up to four controller programs for HPIL devices, or for manual activities that must be performed at precise intervals. The program allows control and timing of four independent "channels". Each channel may have its own six character ALPHA identifier; time, data and note files; simple message alarms; and user defined "control" alarms. Alarms and callable sub-programs 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 stopwatch (SW) with any channel time and of keeping tract of the next available data register to which splits may be stored. Nine "free" registers have been provided for user defined program 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 01 through 04. Flags 01 through 04 are not used by the program but are provided as indicators to the user of which channel is currently being accessed. The states of these flags are preserved in register 06 and therefore will suffer no ill effects if INSTAT or some other flag destroying function is performed.

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

The subroutine labeled [///] (b) initiates the channel whose number is taken from the X register.

The subroutine labeled [///] (c) adds alarms to any running channel without reinitializing it. The channel number is taken from the x register.

The subroutine labeled [///] (e) clears the registers used by the "DATA" program.

The subroutine labeled (C) recalls and prints all of the stored data pertinent to the channel whose number is taken from the X register.

The subroutine labeled (E) synchronizes the 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. Manually setting the stopwatch split register pointer to this value will avoid overwriting important data registers.

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

The user may write programs that access these blocks or registers in the same fashion that "DATA" does. The data is stored in five register records where the lowest numbered register 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 datum, such as the pemperature of the observation at that time.

This program is intended to be compatible with all HPIL controllable devises. All of the routines will allow at least four additional levels of subroutine calls. The display mode used by routine labeled (C) is not controlled by the program and therefore is up to the discretion of the user.

Operating Limits and Warnings

Obsolete alarms must be cleared with the ALMCAT function of the TIME module.

Space must be available in program memory for storage of alarms. Refer to the Time Module Owner's Manual, page ___ for details of the memory requirements of the various alarms.

Failure to adjust the stopwatch pointer when in SW 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 defined programs. User defined program interrupts - control alarms - should preserve the stack whenever possible.

See steop 10 of the User Instructions, page __.

Suggestions and Modifications

All of the local ALPHA labels may be given Global labels to enable access from any defined control program. All of the routines will allow at least four additional levels of subroutine calls. The display mode used by Label (C) is not set in the program and should be added by the user to appropriately display the data.

The HP82180A Extended Functions/Memory Module may be used as suggested in paragraph 3 of "Operating Limits and Warnings" (above) as well as providing a direct means of checking and setting size, storing flag status and sending control characters to various devices in the loop by means of the XTOA function in the Module. Additionally, data may be copied to Extended Memory when the records are full.

DATA REGISTERS

00	Scratch register for User defined program
01	start time of channel 1
02	" " " " 2
03	" " " " 3
04	" " " " 4
05	Store index, returned to R07 after block clear, Label (//) (e)
06	channel flag 01 - 04
07	"DATA" storage index
08	recall index, Label (C); time parameter for XYZALM
09	reset parameter for XYZALM
10	Channel name index
11	start date of channel 1
12	" " " " 2
13	" " " " 3
14	" " " " 4
15 - 22	Scratch registers for User defined programs
23	Name of channel 1
24	" " " 2
25	" " " 3
26	" " " 4
27	start of first "DATA" record: channel number 1 - 4
28	elapsed time
29	Alpha note, first 6 characters
30	" " last 6 characters
31	data or scratch register
32 - nnn	additional records

FLAGS

00	Enables a RTN to Label DATA from line 215 of Label 08.
01	Visual identification for channel 1
02	" " " " " 2
03	" " " " " 3
04	" " " " " 4
21	Printer enable, set to print or halt output if no Printer
23	Test for Program name, Alpha message; stores nulls if no message
25	Prevent error on recalling nonexistent register

LABEL AND FUNCTION INDEX

insert below	(//) (b)	Start timer for channel (n) in X register
	(//) (e)	Clear block of registers allocated to "DATA" routine
	(C)	Recall, view or print "DATA" records for channel (n)
	(E)	Synchronize Stopwatch to channel (n)
	DATA	Store data interactively
**	(//) (c)	Add alarms to channel (n) in the X register without resetting the timer to zero.

Three examples are given to illustrate possible uses of "4CON"/

1. A medical office administers Glucose Tolerance Tests to many of their patients. The tests are four to six hours in length and are used to test for Diabetes and Hypoglycemia. Blood samples must be taken at exact intervals and measurements recorded carefully.

2. Line voltage to a microcomputer must be measured periodically to test the effectiveness of a new Voltage Regulator. Using Channel 3 of "4CON", sample and print the voltage at five minute intervals with the HP3468A Voltmeter.

3. A photographer needs to monitor 3 activities simultaneously including keeping tract of how long a model is working on a particular assignment, how long a batch of prints are being rinsed in his darkroom, and when to leave for an appointment.

PROMPT	INPUT	FUNCTION	DISPLAY

Load the "4CON" program. Prior to running the example problems, be sure that there are at least 8 unused program registers in your calculator. The registers will be used by the "Alarm Stack". The minimum size required to run first problem is SIZE 42.

1. Initialize the program. [XEQ] "4CON"

DATA X?			
	In practice, at least 7 data points would be stored for a 6 hour test. However, for this example, we only require 3 points. (The third point will be used in another example).		
	3	[R/S]	
NAME	BLTEST	[R/S]	
ALARM/HMS?			
	The first test is run at ½ hour. The balance are run at 1 hour intervals.		
	.3	[R/S]	
RESET?	1	[R/S]	
PRGM?			
	We will not be using a special program.		
	no input	[R/S]	BLTEST

If you use the ALMCAT function, you can confirm that the alarm has been set as a message alarm with a 1 hour reset. Use the "DATA" function to store the initial blood sugar level in the patient.

		[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 triggered. 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 1 channel being run by "4CON".

	MOE SUGAR	[R/S]	
DATE?	162	[R/S]	162.00

Recalling the stored date. The recall function in the program can provide immediate hard copy if there is a printer in the HP41 system. As an alternative to manual recall, let the program do the work. A Global label could be inserted after label (C), for example, LABEL "RC". A separate short program would contain these lines:

LABEL "RC-1"	GTO "RC"	
	1	[///] (c)
NAME?		[R/S]
ALARM/HMS?	.4	[R/S]

Setting the second alarm on channel 1 ten minutes (00:10:00) later than the first alarm assures they will not conflict.

RESET?	1	[R/S]	
PRGM?	RC-1	[R/S]	BLTEST

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 printed here, you are running the problem correctly.

1	(C)	BLTEST
		00:01:45.41 **
		FASTING BL/S
		80.0
		00:31:50.41 **
		MOE SUGAR
		162.000000
		END

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

PROMPT	INPUT	FUNCTION	DISPLAY
Initialize Channel 2			
	Input 2	[///] (b)	
NAME?	VRTEST	[R/S]	
ALARM/HMS?	.3	[R/S]	
RESET?	.05	[R/S]	
PRGM?	V	[R/S]	VRTEST

The first alarm will trigger 30 minutes after the timer was started and thereafter every 5 minutes until manually purged. Note that the sample problem count have been written to store certain values in the unused registers, R15 - R22 for comparisms. Additionally, the ALPHA strings used to send instructions to the HP3468A Voltmeter could have been prestored by an initialization routine to speed program execution. 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.

PROMPT	INPUT	FUNCTION	DISPLAY
Initialize Channel 4			
	4	[///] (b)	
NAME?	PRINTS	[R/S]	
ALARM/HMS?	.3	[R/S]	
RESET?	Ø or no input	[R/S]	
PRGM?	no input	[R/S]	PRINTS

The prints must be delivered in one hour. Add an alarm to channel 4 to trigger in one hour.

ALARM/HMS?	1 input	[R/S]	
RESET?	.05	[R/S]	(The prints must be delivered!)
PRGM?	no input	[R/S]	PRINTS

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 function [///] (c). Input the channel number and press [///] (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?	MS (MODEL NAME)	[R/S]	


```
01*LBL "V"
02 AUTOIO
03 2
04 SELECT
05 REMOTE
06 "F2"
07 OUTA
08 "R4"
09 OUTA
10 "T2"
11 ATIME
12 OUTA
13 IND
14 X<>Y
15 FIX 4
16 CLA
17 ATIME
18 PRA
19 X<>Y
20 FIX 2
21 PRX
22 .END.
```

The Voltmeter is the second device in the Loop. Make it the primary device.

AC Volts function is selected.

300 Volt range is selected.

Single Trigger mode causes a single reading to be taken.

Voltage reading is sent to the X register.

Swap with time.

Format and print time.

Print voltage.

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.

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 00:30:00 from the time that the alarm was set. Press [///] (c) to PURGE the alarm.

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.

PROMPT	INPUT	FUNCTION	DISPLAY
Initialize Channel 3			
	3	[///] (b)	
NAME?	MODEL	[R/S]	
ALARM/HMS?	no input	[R/S]	

We are giving the channel an identification that will be used by the "DATA" function.

DATA?	25	[R/S]	25 (the data might reflect an hourly rate).
Recall the data. 3		(C)	MODEL 00:00:05.41 ** MS (MODEL NAME) 25.000000

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

USER INSTRUCTIONS

SIZE: (nxdata) (HP-41C) +27

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the program "Four Channel Controller".			
2	Initialize the program.		[XEQ] "4CON".	DATA X?
3	Input the number of data points to be stored during program operation.	n points	[R/S]	NAME?
4	Input the name or code that identifies the channel (six character maximum).	[ALPHA]	[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.MMSSx	[R/S]	RESET?
6	Input the reset interval.	HH.MMSSx	[R/S]	PRGM?
	If the alarm is a control alarm, input the name of the peripheral function or program name that is to be triggered.	[ALPHA]	[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. (n)	[///] (b)	NAME? (ALPHA)
	Return to step 4.			
10	To add alarms to a channel that has already been initialized.	Ch. (n)	[///] (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. To avoid past-due alarms, particularly past-due control alarms, input the initial alarm HH.MMSS so that it will occur at a future time and input the			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY (SIZE: (HP-41C))
	reset HH.MMSS equivalent to the required cycle. For example, if the timer has been running for (m) hours and the alarm must trigger every (n) hours the first interval must be (m)+(n) hours. The reset will be (n) hours.			
	Input channel identification, if change desired.	[ALPHA]	[R/S]	
11	To clear an alarm that has been set by the program.		ALMCAT	
	Alarms must be cleared manually in the ALMCAT mode (see page ___ of the Time Module Owners' Manual).			
	An alternative is to set a USER Flag that will not be set by any other running program, such as Flag 10. The first line of your control program could be:		FS? 10 [RTN]	
12	To store time-related data interactively.		[XEQ] "DATA"	CHANNEL?
	Input the channel number 1 - 4.	Ch. (n)	[R/S]	NOTES?
	Input a descriptive ALPHA note, up to 12 characters in length.	[ALPHA]	[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]	(ALPHA)
	View data.		[R/S]	data (appears in ALPHA and X)
14	To continue sequentially.		[R/S]	similar to step 13
	It is not necessary to press [R/S] if a Printer is in the system. Display will continue till no more data is found			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY (SIZE: (HP-41C))
	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.		[///] (e)	0.00
16	To display a running timer for any channel.	Ch. (n)	(E)	SYCE (n) R(nn)
	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.		SW	
17	NOTE: If splits are stored at the pointer address displayed in the ALPHA register, those splits cannot be read by function (C) which expects data to be formatted by Label "DATA".			
18	To determine the constant to be inserted in place of line 216 in the "4CON" program a temporary change must be made in the program itself.		GTO . 2 1 9 (PRGM) STOPSW STOP	
	After keying in the additional two lines of code "STOPSW" and "STOP"		[XEQ] PACK	
		0 1	SETSW (E)	
	Place the calculator in SW mode [XEQ]SW and read the display. The displayed value will replace line 216 Be sure that the procedure has been performed more than one time and that the Stopwatch was initially cleared each time.			
	Delete the two temporary lines, STOPSW and STOP.			

```
 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 XEQ e
 29 1
 30 GTO c
 31*LBL b
 32 TIME
 33 STO IND
Y
 34 10
 35 RCL Z
 36 +
 37 DATE
 38 STO IND
Y
 39 LASTX
 40*LBL c
 41 XEQ 09
 42 22
 43 STO 10
 44 +
 45 "NAME?"
 46 AON
 47 STOP
```

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? Flag 01 will be initially set. Flag 04 will be cleared on first pass. Label 09 User mode to enable local labels. Store Flag 04 in Flag register (06) to overwrite possible nonexistent Flag nn and clear Input number of data points Data points*5+26 additional registers Calculate ISG 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

Input channel alpha i.d.

```
 48 AOFF
 49 ASTO IND
 X
 50*LBL 01
 51 "ALARM/H
MS?"
 52 CLX
 53 PROMPT
 54 X=0?
 55 GTO 04
 56 STO 03
 57 "RESET?"
 58 CLX
 59 PROMPT
 60 STO 09
 61 CF 23
 62 "PRGM?"
 63 AON
 64 STOP
 65 AOFF
 66 FC?C 23
 67 GTO 02
 68 ASTO X
 69 ASHF
 70 ASTO Y
 71 "↑↑"
 72 ARCL X
 73 ARCL Y
 74 GTO 03
 75*LBL 02
 76 CLA
 77 RCL 06
 78 RCL 10
 79 +
 80 ARCL IND
 X
 81*LBL 03
 82 DATE
 83 RCL IND
06
 84 RCL 08
 85 HMS+
 86 ENTER↑
 87 ENTER↑
 88 24
 89 /
 90 INT
 91 DATE
 92 X<>Y
 93 DATE+
 94 LASTX
```

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)

Number of hours

and Number of days


```
 95 24
 96 *
 97 ST- Z
 98 CLX
 99 RCL 09
100 X<>Y
101 R↑
102 XYZALM
103*LBL 04
104 RCL 06
105 RCL 10
106 +
107 RCL IND
X
108 RTN
109 RCL 06
110 GTO 01
111*LBL C
112 RCL 10
113 X<>Y
114 +
115 VIEW IND
 X
116 LASTX
117 XEQ 09
118 RCL 10
119 RCL 07
120 FRC
121 +
122 STO 08
123 FIX 6
124*LBL 05
125 5
126 ST+ 08
127*LBL 06
128 SF 21
129 SF 25
130 "END"
131 RCL IND
08
132 RCL 06
133 FC?C 25
134 PROMPT
135 X≠Y?
136 GTO 05
137 ISG 08
138 RCL IND
08
139 ADV
140 CLA
141 ATIME24
```

R09=reset

Place XYZALM parameters in stack in correct sequence

Display routine Channel number+22= ALPHA name

Continue storing alarms on this channel Recall data records Get name of channel (file)

Display or print

Channel number Clear old Flag, display channel on annunciator R10=22 R07=index .see portion of index (highest register)

Initialize R08 as recall index

Get every 5th register

Enable printer or force stop on output display AVIEW Error ignore No more data or no data found Get first register of 'DATA' record Compare to channel number If a nonexistent register tried, end of file Compare to channel Try again Format elapsed time display Get 2nd register of record 2nd register=elapsed time

```
142 AVIEW
143 CLA
144 ASTO X
145 ISG 08
146 RCL IND
08
147 ISG 08
148 X=Y?
149 GTO 07
150 ARCL X
151 ARCL IND
 08
152 AVIEW
153*LBL 07
154 ISG 08
155 RCL IND
08
156 ISG 08
157 CLA
158 ARCL X
159 X≠0?
160 AVIEW
161 GTO 06
162*LBL "DAT
A"
163 TIME
164 TONE 9
165 "NO ROOM
"
166 ISG 07
167 "CHANNEL
?"
168 PROMPT
169 XEQ 09
170 STO IND
07
171 SF 00
172 X<>Y
173 XEQ 08
174 ISG 07
175 STO IND
07
176 ISG 07
177 "NOTES?"
178 CF 23
179 AON
180 STOP
181 AOFF
182 FC?C 23
183 CLA
184 ASTO IND
 07
```

ALPHA null string for comparison

Get first 6 characters of ALPHA note

If no message, then skip blank display

Else, get entire string

Continue display

Get 5th register of record

Display if not 0, place in ALPHA register to maintain uniform print format

Store data interactively

Get time immediately

Feedback

End of file

Increment storage index

Which timer?

Go to Flag update routine

Flag 00 will enable subroutine return

Gosub elapsed time calculation

Store ET in 2nd register

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

Program Listings


```
185 ASHF
186 ISG 07
187 ASTO IND
 07
188 "DATA?"          Any numeric data or
                      measurement?
189 CLX              Default=0
190 PROMPT
191 ISG 07
192 STO IND
07
193 RTN
194*LBL E            Synchronize stopwatch
195 CF 00             to channel (n) whose
196 XEQ 09            number is in X
197 RUNSW            Clear subroutine Flag
198 TIME             Update channel Flag
                     Set stopwatch on the fly
199*LBL 08           Routine 'written into'
200 RCL IND           sync routine to avoid
Y                     label search time on
201 HMS-              first execution if it
202 RCL 06            were a subroutine.
203 10               Search time would make
204 +                 constant on line 216
205 RCL IND           inaccurate.
X                    Get date timer channel
206 DATE              started
207 X<>Y             Current date
208 DDAYS
209 24               Positive difference
210 *                Max. number of hours
211 RCL Z
212 HMS+             Plus time difference
213 FS?C 00          =total number of hours
214 RTN
215 158 E-6          If called by 'DATA' rou-
216 HMS+              tine, return
217 1 E2             Time taken by lines 199
218 MOD               through 219 must be
219 SETSW             added to elapsed time
220 FIX 0            Avoid data error when
221 "SYNC "           stopwatch is set
222 ARCL 06          Format prompt
223 "+  R"
224 RCL 07           Get 'DATA' index and
225 INT               truncate to calculate
226 1                 last used register
227 +
228 ARCL X            Output: Synchronize
229 AVIEW             channel (n); set pointer
230 RTN               to R(nnn)
231*LBL 09
```

5 byte sequence called 5 times saves 8 bytes

```
232 CF IND 0         Clear previous flag (at
6                     initialization set to 04
233 STO 06            to avoid possible error
234 SF IND X          if nonexistent or ALPHA
235 RTN               data)
236*LBL e            Set current channel
237 RCL 05            annunciator. These
238 STO 07            Flags are not tested,
239 ISG X             only displayed.
240 0                Clear block of registers
241*LBL 10            allocated to data. Use-
242 STO IND           ful as a local ALPHA
Y                     label only if block will
243 ISG Y             be used, dumped to
244 GTO 10            storage medium and re-
245 .END.             initialized. Original
                      index restored on com-
                      pletion.
                     0 registers bbb-eee
```


L O G B O O K

"Logbook" uses the HP82180A Extended Functions 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 registers 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. Useful in professional offices and any application where time must be stored in a flexible format.

"Logbook" consists of four main parts: Create File, Start Clock, Stop Clock and 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 timer 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 compares.

Label [///] (b) stops the timer 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 register 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.

DATA REGISTERS

00 n(days) * 24
01 Start/Finish time; Loop control=n records/account (Label C)
02 Start/Finish Date

ASCII FILE VARIABLES

Record	Variable
0	Account Name
1	Account Code, Billing number, Purchase Order, etc.
2	Remarks or description
3	each succeeding pair of Records contains a Start Date MM.DDYYYY in the
4	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 till
:	end of File
:	
:	

FLAGS

21 Enable Printer or halt on AVIEW
23 Alpha Input? Allows skipping File Name input if already positioned to Working File
55 Printer existence.

FUNCTION INDEX

$TIME Initializes a new File and prompts for Variables
(///) (b) Stops the timer 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

---

Status.

SIZE 3 TOTAL REGISTERS 57

ASCII FILE SIZE $\frac{(nA \cdot 60) + (nA \cdot n(D+1) \cdot 20) + 30}{7}$ A=Accounts D=Days

FIX 4, 6 USER MODE ON

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 the travel time would be helpful.

PROMPT	INPUT	FUNCTION	DISPLAY
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		
REMARKS?	BUILDING WRECKERS	[R/S]	
NAME?	BENJAMIN ELIAS	[R/S]	
CODE?	JOB NO 97330-2.5	[R/S]	
REMARKS?	PLUMBING CONTRACTOR	[R/S]	0.000000

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

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.

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

The following routine will print or view the entire ASCII file. Substitute OUTA for PRA if you are using an HPIL printer that responds to this command.

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

To provide a reasonable time for this example, use the T+X functions in the Time Module to set your clock back four hours.

```
          -4              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.

```
                          (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 occurence of the string "NICHOLAS".

```
                          START
```

The display confirms that the clock has started running.

Start the clock for the second client.

```
                          (B)

FL NAME                   [R/S]

ACCT?     BEN             [R/S]       START
```

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

```
          1               T+X         1.000000
```

Stop the clock for one client.

```
                          [///] (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.

After the clock has been stopped for the second client, the ASCII file would now look something like this:

```
NICHOLAS NABIL
JOB NO 97321-4.5
BUILDING WRECKERS
11.011981
1.021345
-
11.011981
20.364782
BENJAMIN ELIAS
JOB NO 97330-2.5
PLUMBING CONTRACTOR
11.011981
1.020761
-
11.011981
20.371007
```

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

```
                          (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               T+X
```

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

Stop the clock and display the time file.

		[///] (b)	
FL NAME?		[R/S]	
ACCT?	NICHOLAS	[R/S]	03:00:00.00

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

Print the files.		(C)	
FL NAME?		[R/S]	
ACCT?	NICHOLAS	[R/S]	

```
NICHOLAS NABIL
JOB NO 97321-4.5
BUILDING WRECKERS
11.011981
1.021345
11.011981
3.004699
-
11.011981
21.401292
BENJAMIN ELIAS
JOB NO 97330-2.5
PLUMBING CONTRACTOR
11.011981
1.020761
11.011981
3.004364
-
11.011981
21.403439
```

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

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

Whenever a time is output in ALPHA, it is also in the X register in decimal form and the Y register in HMS form.

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

11/01/1981
01:02:13
11/01/1981
03:00:46
========
04:05:41

BENJAMIN ELIAS
JOB NO 97330-2.5
PLUMBING CONTRACTOR

11/01/1981
01:02:07
11/01/1981
03:00:43
========
04:05:15
```


USER INSTRUCTIONS

SIZE (HP-41C) 3

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the "Logbook" and initialize.		[XEQ] "$TIME"	N ACCTS?
2	Input the number of individual clients or accounts that are to be timed.	n	[R/S]	N DAYS?
3	Input the number of days or times that each account will be timed.	n	[R/S]	FL NAME?
4	Select a name for the ASCII file, up to 7 ALPHA characters.	[ALPHA]	[R/S]	NAME?
5	Input the name of the first account in the file. Up to 60 ALPHA characters are allowed for the name and the following two lines.	[ALPHA]	[R/S]	CODE?
6	Input a Purchase Order number, billing code or any selected descriptive characters. Note that only character No. 95 is illegal in the file.	[ALPHA]	[R/S]	REMARKS?
7	Any descriptive ALPHA string may be input. If no remarks are necessary, press [R/S].	[ALPHA]	[R/S]	NAME?
8	If the ASCII file is to contain more than one account, the prompt "NAME?" will be repeated as many times as necessary. Return to step 5 for the input format.			
9	To start the clock for any account.		(B)	FL NAME?
	If the ASCII file is your working file (refer to page ___ of the HP82180A Extended Functions/Memory Module Owners Manual for details and definitions) then no input is necessary. If uncertain,	(FL NAME)	[R/S]	ACCT?
	Input the name of the account to be timed.	(NAME)	[R/S]	START
10	To stop the clock for any account.		[///] (b)	FL NAME?
	Refer to step 9 for input instructions.	(FL NAME)	[R/S]	ACCT?
11	Input the name of the account being timed.	(NAME)	[R/S]	(nn:nn:nn.nn)

USER INSTRUCTIONS

SIZE: (HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
12	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 input at initialization (step 3) the file may become prematurely full and not accept further data.			
13	Error recovery. If a timer was started accidentally, 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 9 to start.			
14	To display the timer data stored under a particular account name.		(C)	FL NAME?
	Input the file name if the file is not your working file.	(FL NAME)	[R/S]	ACCT?
	Input the name of the account. If the name begins with distinctive characters that are not repeated elsewhere in the file, only the first few characters need be input.	(NAME)	[R/S]	(NAME) (CODE) (REMARKS) (MM.DDYYYY) (HH.MMSS)
	The elapsed time for the displayed date also is in the Y register in HH.MMSShh form and in the X register in decimal form for any rate calculations that may be required. 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. If the temporary use of registers 00, 01 and 02 will disturb a working program or valuable data, they may be reprogrammed to any			

USER INSTRUCTIONS

SIZE: (HP-41C)

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	other register numbers. They are only in use temporarily and contain no data required to maintain the timers.			
15	To output the entire ASCII file, see page ___.			

Program Listings

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


Program Listings

```
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
111 "START"        at end of file from becoming first record (0).
112 PROMPT         Confirm action
113+LBL 02         Temporary storage of time and date while user is prompted for file name prevents loss of data in stack
114 STO 01         Format for full precision of time/date data
115 DATE           Prompt account name, return -1 if misspelled
116 STO 02         Position to end of data space by finding character 95
117 FIX 6
118 XEQ 06
119 95
120 XTOA
121 POSFL
122 1
123 +
124 SEEKPT         Position file to scratchpad area
125 RTN
126+LBL C           Routine to display time
127 SF 21          Enable printer or cause halt on AVIEW
128 XEQ 06         Prompt account name
129 95             Position to scratch space delineator
130 XTOA
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             Last portion, RTN not necessary
172 GTO 05
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.
```

B I C Y C L E C A L C U L A T O R

This program is based on "Bicycle Commuter Computer", HP41C Users' Library number _____C by Durwin A. Schmitt.

Reference: Sloan, E. A.: The Complete Book of Bicycling, Trident Press, New York, N.Y. 1970

The program stores data on magnetic cards on the number of teeth on each chainwheel and rear sprocket cog as well as the wheel diameter. If unchanged, the data need not be reentered with each use. Desired velocity may be input to determine required pedal cadence. Alternatively, speed may be calculated by inputting chainwheel and rear cog number in use. The Time Module is used to compute speed and to set a pace for a desired speed. Inputs may be in miles or kilometers.

The distance traveled for each wheel revolution is:

Distance, inches = PI (wheel diameter, inches). The wheel diameter is usually 27 inches.

The number of wheel revolutions per pedal revolution is equal to the ratio of the teeth on the front gear being used, the chainwheel, divided by the number of teeth on the rear gear in use, the rear cog.

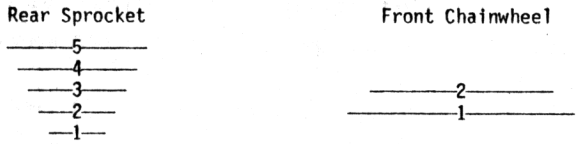
$$\text{Velocity (V)} = (\text{Distance per wheel rev}) \cdot \frac{(\text{No. of wheel revs})}{(\text{No. of pedal revs})} \cdot \frac{(\text{Pedal revs})}{(\text{Time interval})}$$

$$= \text{PI} \left\{ (\text{Wheel diameter}) \cdot \frac{(\text{Teeth on chainwheel})}{(\text{Teeth on rear cog})} \right\} \cdot \frac{(\text{Pedal revs})}{(\text{Time interval})}$$

The gear ratio, the term in square brackets, is normally expressed in inches, so that converting to a velocity in miles/hour the equation becomes:

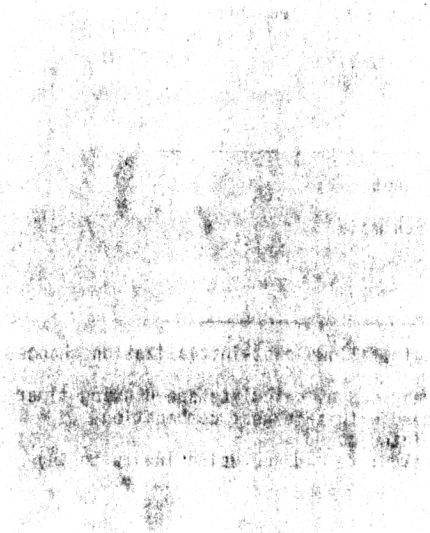
$$= \text{PI (GR)} \cdot \frac{(\text{revs} \cdot 1 \text{ foot} \cdot 1 \text{ mile} \cdot 3600 \text{ sec})}{(12 \text{ in} \cdot 5280 \text{ ft} \cdot 1 \text{ hr})} \quad \text{or} \quad \text{PI (GR)} \frac{\text{rev} \cdot 3600}{63660}$$
$$\text{seconds}$$

View looking down from rider's position:



The program accepts from one to three chainwheels and one to seven rear cogs. The number of teeth are input interactively and may be saved on magnetic cards for later use.

The Pace subroutine is similar to the "PACE" program, page ___. Input is limited to 60 RPMs to eliminate the possibility of the routine causing a past due alarm and stopping the audible cadence. Although the Time Module will exhibit virtually no variance from one HP41 to another, the time to calculate the alarm interval and reset does vary somewhat and is not quartz controlled. The routine can be used to establish an audible or visual cadence for the left leg, i.e., half pace.



DATA REGISTERS

00	Index for rear cog input, storage
01	Rear cog 1
02	" " 2
03	" " 3
04	" " 4
05	" " 5
06	" " 6
07	" " 7
08	Chainwheel 1
09	" " 2
10	" " 3
11	Wheel diameter
12	scratch
13	M or K for output labelling, miles, kilometers
14	Miles to kilometers conversion factor, or 1 if no conversion
15	"READY"
16	Index for chainwheel input, storage

FLAGS

05	kilometers if set, miles if not
21	Cleared if no printer
27	Enable local labels, set USER mode
55	Printer existence

USER KEYS

(///) (a)	Miles - kilometers toggle, allows post-initialization change
(A)	Input Chainwheel and Rear Cog n, calculate speed using timer
(B)	calculate different speed with same gear combination
(C)	Cadence (RPM) of calculated speed
(D)	Input speed desired, calculate cadence using inputs to (A)
(E)	Cadence/2 routine
"P"	Pace subroutine

Your bicycle had 27 inch wheels, with 52 and 47 teeth on the front chainwheels and 14, 17, 21, 26 and 32 teeth on the rear sprocket cogs. The chain is positioned on the 47 tooth chainwheel and the 17 tooth rear cog. What is the velocity? What is the pedal cadence (RPM)? What cadence is necessary to achieve a velocity of 20 miles/hour?

PROMPT	INPUT	FUNCTION	DISPLAY
Load the "BIKE" program. GTO .. Load the Pace subroutine "P".			
Initialize the program.			
	[XEQ] "BIKE"		
R/COG 1?	14	[R/S]	
R/COG 2?	17	[R/S]	
R/COG 3?	21	[R/S]	
R/COG 4?	26	[R/S]	
R/COG 5?	32	[R/S]	
R/COG 6?	no input	[R/S]	
CHAIN WL 1?	52	[R/S]	
CHAIN WL 2?	47	[R/S]	
CHAIN WL 3?	no input	[R/S]	
WL DIAM?	27	[R/S]	
MILE/KM? M/K			
Program has stopped in ALPHA mode.			
	M	[R/S]	READY 1 of 1 Prompts for insertion of a data card.
Insert card.		[R/S]	READY

Calculate your speed.

(A)

CHAIN WL? 2 [R/S]

R/COG? 2 [R/S] COUNT REVS

Count the number of full pedal revolutions between successive pressing of the [R/S] key. In practice, you may count as many as you like after pressing [R/S] and press the key again when you are finished. To provide a common answer for this example and to be sure that you are running the program correctly, you should manually set the stopwatch with the following key strokes:

.0014 STOPSW SETSW (PRGM). You should see 86 TONE 9. Single step in program mode to line 90 and press (PRGM) again. Now continue running the program.

[R/S]

REVS? 15 [R/S] MPH=14.3

You may wish to try this step again. To run the problem again without changing the chainwheel and rear cog already input, Press (B).

To see your pedal cadence after calculating velocity:

(C) RPM=64.6

If you reran the example for a different length of time, your answer would not be the same.

Use the Pace routine to establish a pedal cadence.

(E)

RPM/2? 40 [R/S] READY

The actual display you see will reflect the time of day plus the time offset. To see the full display:

FIX 6

If you use the routine at an interval near its limit, the display will barely be seen. To see the time display

CF 26

USER INSTRUCTIONS

SIZE: (HP-41C) 17

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the "Bicycle Calculator" program.		GTO . .	
2	Load the Pace subroutine, "P".		GTO . .	
	Do not merge or otherwise combine with the main program. "P" works best if it is the last program in USER memory.			
3	Initialize the "Bicycle Calculator".		[XEQ] "BIKE"	R/COG 1?
	If you have previously stored data that does not require a change, such as a different chainwheel, go to step			
	Input the number of teeth on rear wheel cog number 1, counting from the inside to outside from the rider's perspective, see diagram, page ___.	n	[R/S]	R/COG 2?
	Continue input, up to seven times, for each rear cog on your bicycle.	n	[R/S]	R/COG (n)?
	When you have input all of the rear cog data for the particular bicycle:	no input	[R/S]	CHAIN WL 1?
4	Input number of teeth on chainwheel.	n	[R/S]	CHAIN WL 2?
	Continue input. Program accepts up to 3 chainwheels.	n	[R/S]	CHAIN WL 3?
5	If you do not have a third chainwheel:	no input	[R/S]	WL DIAM?
6	Input wheel diameter in inches.	inches	[R/S]	READY 1 of 1
7	If you prefer to use the HP82180A Extended Functions? memory module.		[R/S] ALPHA "YOUR FILE NAME" ALPHA 16 CRFLD .015 SAVERX	
8	To load data from Extended Memory.		ALPHA "YOUR FILE NAME" ALPHA Ø SEEKPTA .015 GETRX	
9	To load data from a magnetic card, merely insert card in reader.			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: (HP-41C) DISPLAY
10	Chainwheel and rear cog positions will be represented by the following convention, looking at the gears from the readers perspective and counting from the inside to the outside. Rear Sprocket — Front Chainwheel 5 — 2 4 — 1 3 2 1			
11	Calculate speed in miles or kilometers/hour.		(A)	CHAIN WL?
	Input chainwheel number 1, 2 or 3.	position	[R/S]	R/COG?
	Input rear cog position 1 through 7	position	[R/S]	COUNT REVS
12	When ready, start counting complete pedal revolutions at tone signal.		[R/S]	tone signal ... COUNT
	When you have counted several cycles		[R/S]	REVS?
	Input number counted between tones.	n	[R/S]	MPH= (nn.n)
	If you had initialized the program for kilometers.			KPH= (nn.n)
13	To change the original units of distance.		[///] (a)	MILE/KM? M/K
	A card reader prompt will appear and may be ignored if change is temporary.		( )	
14	To calculate a different rate of speed using the same gear combination.		(B)	COUNT REVS
	Go to step 12.			
15	To calculate the pedal cadence (revolutions per minute) for the calculated speed.		(C)	RPM= (nn.n)
16	To calculate pedal revolutions required to travel at a desired rate, with the same gear combination as before.		(D)	SPEED?

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: (HP-41C) DISPLAY
17	To use the Pace subroutine to establish an audio and/or visual pace indicator		(E)	RPM/2?
	Input rate to be paced, in (n) beats per nimute.	rate	[R/S]	READY
	When ready (do not press any other keys)		[R/S]	Tones.....
18	To return to the "Bicycle" program.		GTO "BIKE" or use the assigned key of your choice.	
	Clear prompt		(←)	
	If "P" were called as a subroutine, it would only execute one time before returning to "BIKE". If it were written as part of the main program it could not be as fast.			

Program Listings

```
01*LBL "BIK
E"
02 SF 27                Enable local labels
03 FC? 55               Printer?
04 CF 21
05 1.007                Set index for rear cog
06 STO 00                 input and storage
07 9.01                 Set index for chainwheel
08 STO 16                 storage
09 FIX 0                Set input display format
10*LBL 00               Input loop
11 "R/COG "             Prompt: Input number
12 ARCL 00                of teeth on rear cog
13 "+?"                 Get number
14 CLX
15 PROMPT               Set up input test
16 X=0?
17 GTO 01               No more?
18 STO IND              Index
00
19 ISG 00
20 GTO 00               Continue loop
21*LBL 01
22 "CHAIN W             Next loop, input chain-
L "                       wheel data
23 RCL 16               R16 is index register
24 7
25 -
26 ARCL X               Calculate number
27 "+?"
28 CLX
29 PROMPT               Set up input test
30 X=0?
31 GTO 02               No input?
32 STO IND              Index
16
33 ISG 16
34 GTO 01               Continue loop
35*LBL 02
36 "WL DIAM             Next parameter
?"                        Wheel diameter
37 PROMPT
38 STO 11
39*LBL a
40 "M"                  Set up test for input=
41 ASTO Y                 miles
42 "MILE/KM             I/O units in metric or
? M/K"                    english?
43 AON                  ALPHA mode
44 STOP
45 AOFF
```

```
46 ASTO X
47 STO 13               Place in stack for test
48 CF 05                Clear metric flag
49 X=Y?
50 SF 05                Not M? Then set metric
51 1                      flag
52 ENTER↑               Default of 1=miles
53 5
54 LN
55 FC?C 05              Miles to kilometers
56 X<>Y                   shortcut conversion
57 STO 14               Store correct value
58 "READY"
59 ASTO 15              Store prompt for later
60 .015                   reuse
61 WDTAX                Turn on card reader
62 RCL 15
63 RTN                  Display READY
64*LBL A
65 "CHAIN W             Input label, inputs remain
L?"                       for all further calcula-
66 7                      tions, Labels B, C, and D
67 PROMPT               Chainwheel position?
68 +                    Get location
69 RCL IND              Get data, number of
X                         teeth on wheel
70 RCL 11               Get diameter of wheel
71 *
72 "R/COG?"             Input rear cog
73 PROMPT
74 X<>Y
75 RCL IND              Teeth on chainwheel/
Y                         teeth on rear cog *PI
76 /
77 PI
78 *                    Times the wheel diameter
79 STO 12                 = PI (gear ratio)
80*LBL B
81 STOPSW
82 0
83 SETSW                Initialize stopwatch
84 "COUNT R             Cue next activity
EVS"
85 ASTO X               Leave cue in X
86 PROMPT
87 TONE 5               Signal start counting
88 RUNSW
89 STOP                 User counts arbitrary
90 STOPSW                 number of pedal revo-
91 TONE 9                 lutions, presses R/S
                          End of count confirm
```


Program Listings

```
92 PSHF                 Build output prompt
93 "+?"
94 PROMPT
95 36 E2                3600
96 STO 00               Save constant for later
97 *                    Times 12 times 5280
98 63360
99 /
100 RCL 12              Divided by the gear ratio
101 *
102 RCLSW
103 HR
104 RCL 00
105 *                   Calculates velocity and
106 STO 00                saves answer
107 /
108 CLA                 Build output display
109 ARCL 13             R13 contains either K for
110 "+PH="                kilometers or M for
111 RCL 14                miles
112 *                   Kilometers to miles or
113 GTO 04                default=1
114*LBL C
115 900
116 RCL 00              Recall calculated RPM
117 /
118 GTO 03              To output
119*LBL D
120 "SPEED?"            Calculate required
121 PROMPT                cadence for input
122 RCL 14                speed
123 /
124 RCL 12              Metric conversion, if
125 PI                    optioned
126 /                   Recall gear ratio
127 /
128 PI                  Simplify equation
129 1056
130 /
131 /
132*LBL 03
133 "RPM="
134*LBL 04              Output displays
135 FIX 1
136 ARCL X
137 AVIEW
138 RTN
139*LBL E
140 60
141 "RPM/2 ?            Initialize pace, set up
"                         input error trap
                        Reminder that half the
                          required pace should
                          be input
```

```
142 PROMPT
143 X>Y?                Input too large may cause
144 GTO E                 routine to fail--
145 .01                 Try again
146 HR
147 X<>Y                Convert pace per minute
148 /                     to clock time
149 HMS
150 RUNSW
151 TIME                Set stopwatch to time
152 HMS+                  plus alarm offset to
153 SETSW                 reduce algorithm to
154 RCL 15                simplest form
155 "↑↑"                Displays READY
156 STOP                Set up control alarm
157 GTO "P"             Wait
158 .END.               Go to global routine
```

PLAYBACK - PROGRAMMABLE TIMER

This Timer may be interactively programmed by users to playback a series of ALPHA messages up to 12 characters in length for specified lengths of time. Optionally, user defined programs may be run during any segment of the playback sequence. The programs may scroll longer displays, print or perform any other functions using available registers. Periodic tones may be played to accompany messages. The program's features include routines to store, recall, save and edit playback segments, treated as records, and is compatible with any HP41C storage medium. Two modes are available: Manual stop-start or auto sequence without pause through entire playback. Mode may be chosen when initializing run.

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

Label "NM", Next Manual, is set in manual mode when user desires to press [R/S] to run each step. It is also used to start the playback and works by storing time in an accumulator register, R00, and setting the control alarm to trigger relative to that time. The display flag annunciator for Flag 00 signifies manual mode.

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 till the last record.

Label "T" triggers 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. The number of periods is controlled by a decrement loop.

Label "AR", called in the program as Label 10 (because the numeric label distance is compiled) is the "ALMREL" program in the Time Module Owners' Manual, page ___. 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 inputs must all be trapped to prevent playback errors, such as tone intervals longer than the step interval or short enough to cause a past due alarm, failure to input a playback message or program name which would cause the display to be blanked by clear ALPHA on playback and non-existent tones. If invalid inputs are detected, the input prompt is repeated.

User defined programs are flagged and 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, wait till the alarms "NA" or "NM" are triggered.

"Playback-Programmable Timer" may be used in a variety of applications and amusements inlcuding childrens games and activities, photographic darkroom, calisthenics and warmup 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 rallies, timing long distance phone calls.

For this example, "PB" will be programmed with simple messages to show how it works in practice.

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	available for user program
10	available for user program
11	Alpha message, first six characters Ø if user program optioned
12	Alpha message, last six characters Program name, indirect GTO
13	step HH.MMSS total time till start of next step
14	tone pitch, 1 to 9 - decimal point - tone cycle .MMSS
15	begin next record (step two)

FLAGS

00	Manual run
04	Edit
05	Skip tone input, user defined program
06	Run user program
07	No tone
08	First pass, skip prompt
21	Printer enable, print or cause halt on AVIEW during review
23	Test alpha input
25	Check insufficient size, trap invalid tone parameter
55	Printer existence

USER KEYS

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

Status.

Size (n steps * 4) + 11 TOTAL REGISTERS 77 + Size

FIX 0, 4 USER mode ON

This example will require a minimum SIZE of 3 registers. Program the "Playback 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.

PROMPT	INPUT	FUNCTION	DISPLAY
Initialize the program.			
	[XEQ] "PB"		
N STEPS?	5	[R/S]	
MESSAGE?	BODY TWISTS	[R/S]	
STEP HMS?	.01	[R/S]	one minute
TONE/HMS?	.001	[R/S]	ten seconds
TONE N?	9	[R/S]	Tone 9
MESSAGE?	FOOT CIRCLES	[R/S]	
STEP HMS?	.003	[R/S]	
TONE/HMS?	.001	[R/S]	
TONE N?	8	[R/S]	
		[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?	9	[R/S]	
		[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
HMS=00:01:00
TN 9/00:00:10
2=FOOT CIRCLES
HMS=00:00:30
TN 8/00:00:10
3=PRGM, FASTP
HMS=00:01:00
4=CALF STRETCH
HMS=00:00:45
TN= 9/00:00:10
5=PRGM, FASTP
"END" (does not print out) HMS=00:00:30

We will use a modification of the "PACE" subroutine to establish a faster cadence for steps 3 and 5.

Enter the program "FASTP" in your calculator's program memory.

Initialize the Auto Run "Playback" mode.

(E) AUTO

If "AUTO" is not seen in the display, try again.

(E) AUTO

[R/S]


```
01*LBL "FAS
TP"
02 CF 02        Initialize Flag not used by "PB"
03 23           test recall index to determine if earlier or
04 RCL 03       later occurence of "FASTP"
05 X<Y?
06 SF 02        earlier, one minute timing
07 11
08 ENTER↑
09 FS?C 02
10 +
11 STO 09       R09 unused by "PB"
12 RUNSW
13 2 E-4        apply 2 second time offset to stopwatch for
14 TIME         fast loop
15 HMS+         set on the fly
16 SETSW        Control Alarm entry point
17 "↑↑FP"
18*LBL "FP"
19 TONE 9       Control Alarm .
20 CLST
21 RCLSW        Initialize the stack for alarm w/o reset.
22 DSE 09
23 XYZALM
24 .END.
```

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: (steps x 4) (HP-41C)+ 11 DISPLAY
1	Load the "Playback" program and set USER mode.		USER	
2	Initialize the program.		[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?
	If you have an HP82180A Extended Functions/Memory Module, you should refer to page ___ of the Owners' Manual for PSIZE.			
4	Input message to be played back, up to 12 ALPHA characters.	ABCDE...	[R/S]	
	If no message is to be run, but rather a program,	no input	[R/S]	PROGRAM?
5	Key in the name of your playback program.	ABCDEF...	[R/S]	STEP HMS?
6	Input the length of time that the message or program message is to be displayed.	HH.MMSS	[R/S]	TONE/HMS?
7	If a periodic tone is wanted, input the time interval.	HH.MMSS	[R/S]	TONE N?
8	Input tone pitch, from 0 to 9.	TONE n	[R/S]	0.00 MESSAGE?
9	Return to step 4 of instructions till done.			
10	To recall input steps at any time, starting with nth step.	Step n	(C)	n= (ABCDEF...)
	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)
11	To view following steps.		[R/S]	TN n/(nn:nn:nn)

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: (HP-41C) DISPLAY
12	To correct or change any step.	Step n	(D)	MESSAGE?
13	Refer to step 4 of instructions for input parameters. Once edit is started, follow through entire steop or subsequent steps will be stored in unplanned locations.			
14	The entire Playback sequence may be saved on any HP41C storage medium.			
	Magnetic Cards		RCL 02 FRC WDTAX	
	Mass storage, such as HP82161A Digital Cassette Drive.		ALPHA "YOUR FILE NAME" ALPHA 1000 RCL 02 FRC * CREATE Ø SEEKR LASTX WRTRX	
15	To playback the stored messages		(E)	AUTO OR MANUAL
	If the desired mode is not seen, press again.		(E)	
	AUTO mode will cycle continuously after starting from step to step without stopping.			
	MANUAL mode requires the user to restart at each step.		[R/S]	
16	Guidelines for user defined programs to be run in place of playback messages:			
	Run time of user program should not exceed the programmed step time.			
	Five levels of subroutines are available.			
	Two lower numbered data registers are			

USER INSTRUCTIONS

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: (HP-41C) DISPLAY
	available, R09 and R10. Other data must be stored in higher numbered data registers. 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 function that will interfere with the operation of a Time Module based program. The stack and ALPHA register 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 Listings

```
01*LBL "PB"
02 "N STEPS
03 PROMPT
04 4
05 *
06 11
07 +
08 "RESIZE>
="
09 FIX 0
10 ARCL X
11 1
12 -
13 SF 25
14 STO IND
X
15 FC? 25
16 PROMPT
17 FC?C 25
18 GTO "PB"
19 .1
20 %
21 10
22 +
23 STO 02
24 0
25*LBL 00
26 STO IND
Y
27 ISG Y
28 GTO 00
29*LBL B
30 CF 04
31*LBL 20
32 "END"
33 ISG 02
34 GTO 01
35 PROMPT
36*LBL 01
37 CF 05
38 CF 23
39 "MESSAGE
?"
40 AON
41 STOP
42 ASTO X
43 FS?C 23
44 GTO 02
45 CLX
```

Number of steps required to calculate index

Each step record consists of 4 registers

Add 10 housekeeping registers

Format size prompt

Test for existence of highest number R required

Display if necessary

Repeat initialization prompts if size inadequate /1000

Add scratch registers

Save index

Clear block of registers to prevent recall of garbage playback

Store routine

Clear edit Flag

Display END (no more data registers) if Indexx=eee

Flag 05 will be tested for 'skip tone input' on line 63

Prompt message input

No input?

Then try next possibility

```
46 "PROGRAM
?"
47 STOP
48 FC? 23
49 GTO 01
50 SF 05
51*LBL 02
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?"
60 PROMPT
61 STO IND
02
62 CLX
63 FS?C 05
64 GTO 04
65*LBL 03
66 "TONE/HM
S?"
67 CLX
68 PROMPT
69 X=0?
70 GTO 04
71 6 E-4
72 X>Y?
73 GTO 03
74 X<>Y
75 RCL IND
02
76 X<=Y?
77 GTO 03
78 X<>Y
79*LBL 04
80 ISG 02
81 STO IND
02
82 X=0?
83 GTO 06
84*LBL 05
85 9
86 "TONE N?
"
```

Input program name if chosen in place of message
If no input here, cycle back to message prompt
If program name input, skip tone prompt
Store message or program name, first part is in X, if program name x=0

Input step time HH.MMSSx

Save in next register, 3rd register of step record

Skip tone flag set?

Tone interval desired, if any

No input?

6 seconds is min. time allowed to prevent possible past-due alarm due to slow processing or label search
Cycle back if error

Test if greater than step HMS
Cycle back on error
Return input to X
If no tone input prompted execution resumes here
Update index

Tone 9 supplied as default

Prompt pitch

```
 87 SF 25
 88 PROMPT
 89 TONE IND
  X
 90 FC?C 25
 91 GTO 05
 92 ST+ IND
08
 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' c first pass, skip manual mode prompt


```
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 pe
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

PACER

"Pacer" establishes an accurate beep at intervals as short as 60 beeps per minute or faster if Flag 26 is clear. The program consists of two independent routines, each with their own END, and is written for maximum speed and accuracy.

"PACE" is the initializing program for the actual "Pacer" program, Label "P". "PACE" can be located anywhere in program memory, but "P" is initialized faster, particularly on short intervals, if it is the last program in program memory. This is due to the method of global label search in which the calculator seeks the last label first. "PACE" error traps inputs greater than 60, signifying loops shorter that 1 second, to ensure that the routine will not cause a past-due alarm and stop prematurely. "P" uses no flags, data registers or GTOs. It simply sets a control alarm. All lines of code that could have been removed from the routine were placed in the initialization program, including the control alarm code [///]. It is not critical to set the stopwatch, from which "P" gets its trigger time, to exact clock time because only the first loop will be affected. Using the stopwatch to "store" the trigger time avoids the HMS+ arithmetic and the need for data registers and additional code. The stack is cleared to prevent random data from setting reset or date parameters, the stopwatch placed in X and the alarm set. The program executes a single tone and stops, waiting for the next occurrence of the control alarm. "P" is defined as a control alarm by the two "/" characters left in ALPHA by the initialization program. "P" is also used by the "Bicycle Computer", page ___ although the initialization is in the "BIKE" program itself.

PROMPT	INPUT	FUNCTION	DISPLAY

Use the Pacer program to set a pace for exercising at 30 beats per minute.

Initialize the program.

[XEQ] "PACE"

PACE/MIN?	30	[R/S]	RUN
Run the program.		[R/S]	

Operating Limits and Warnings.

Due to slight variations in the operating speed of most HP41C/Vs the routine will not operate faster than one tone per second. Do not clear Alpha or Control Alarm characters will be lost.

USER INSTRUCTIONS

SIZE: (HP-41C) Ø

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the initialization routine "PACE"		GTO . .	
2	Load the Pacer program, "P"		GTO . .	
3	Initialize the program.		[XEQ] "PACE"	BEATS/MIN?
4	Input the number of times per minute that you would like to have the pace prompted audibly. (Less than or equal to 60).	n times	[R/S]	
	If the input number was greater than 60, the program will cycle back to the original prompt.			RUN
5	Start the Pacer program.		[R/S]	
	The display seen reflects the clock time offset by the pace interval with an insignificant error.			
6	To stop the Pacer, quickly press ALPHA and backarrow. The alarm becomes a simple alarm without message and can be acknowledged by ( ). It will not reset.			
	If you are using label P as part of another program, do not call it a subroutine. As a subroutine, it would only run one loop before returning.			

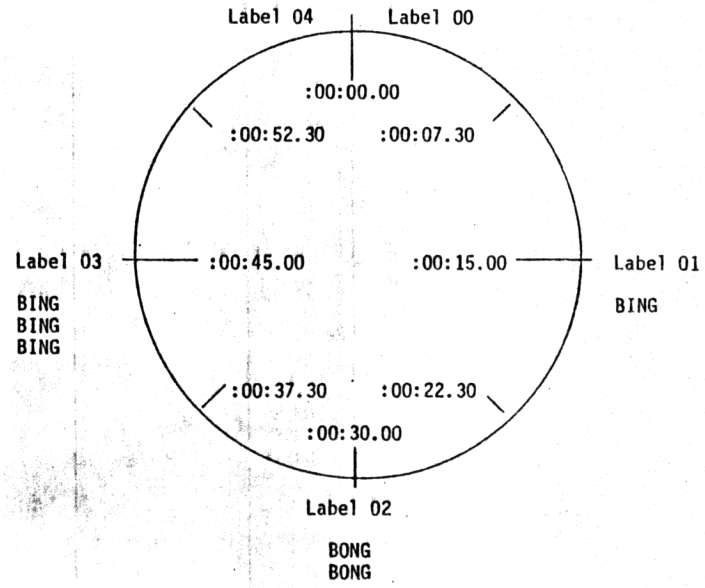
Program Listings


```
01*LBL "PAC
E"
02 60                    Set input limit
03 "PACE/MI
N?"
04 PROMPT
05 X>Y?                  Test if greater than 60
06 GTO "PAC
E"                       Repeat if failed test
07 .01                   Convert to time format
08 HR
09 X<>Y
10 /
11 HMS
12 RUNSW
13 TIME                  Synchronize stopwatch
                           to clock
14 HMS+
15 SETSW
16 "RUN"                 Leave prompt in X
17 ASTO X
18 "↑↑"                  Control alarm command
                           placed in ALPHA
19 STOP
20 GTO "P"               Go to Pace routine
21 .END.
```

```
01*LBL "P"
02 CLST                  Initialize stack for alarm
                           parameters (clear reset)
03 RCLSW                 Get time of next occur-
                           rence
04 MYZALM
05 TONE 9                Pace audible feedback
06 .END.                 Program stops here, waits
                           for control alarm to
                           trigger and falls through
                           END or .END. to first
                           line of program
```


GRANDFATHER CLOCK

"Grandfather Clock" allows the User to tell time to the nearest quarter-hour in the dark by sounding deep "bongs" for the hour, followed by 1 "bing" if nearest to the quarter-hour, 2 "bongs" if one-hour or 3 "bings" if three-quarter-hour. The routine uses only the stack and Flag 25.



Status.

SIZE any TOTAL REGISTERS 13

FIX 2 USER mode n/a

"Grandfather Clock" allows the User to tell time in the dark, or periodically by setting "GC" as a Conditional or Control alarm. The time of day is output audibly in mimicry of a Grandfather Clock, rounded to the nearest 1/4 hour. The clock face is rotated 45 degrees so that the time period within 7 and 1/2 minutes before the hour will be rounded up to the next whole hour. Time output is always in 12 hour format. The routine sets Flag 25 and searches for a non-existent label to place slight pauses between the tones to allow counting.

USER INSTRUCTIONS

SIZE: (HP-41C) Ø

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Load the program. Any mode may be used. Only the status of Flag 25 will be affected and the stack will be lost.			
2	Listen to the time.		[XEQ] "GC"	Audible feedback.
	The audio output is interpreted as follows:			
	Number of hours (MOD 12) = repetitions of tone 3.			
	If nearest to quarter-hour: tone 8 half-hour: tone 4 (2X) 3/4 hour: tone 8 (3X)			
		Reset		
3	To use as an alarm (either a Conditional or Control Alarm).	HH.MMSS HH.MMSS MM.DDYYYY	ENTER ENTER XYZALM	
	(Precede "GC" in ALPHA with one or two "/" as required for alarm type).			
	To see how the routine works at various times of day, GTO . 003, and key in the time (must be positive). [R/S] to hear the time.			

Program Listings

```
01 ◆LBL "GC"      Grandfather Clock
02  TIME
03  FIX 2
04  CLA
05  ATIME         Format time display
06  .073          Add 7½ minutes
07  HMS+
08  12
09  MOD           Convert time to 12 hour
10  INT             clock face
11  X=0?          If zero, then 12 o'clock
12  12
13 ◆LBL 05
14  TONE 4        Low 'bong' tone for
15  SF 25           hours
16  GTO 06        Slow loop slightly
17  DSE X         By attempting to find a
18  GTO 05          non-existent label
19  LASTX         Repeat for each hour
20  FRC           Get minutes
21  1 E2          To integer
22  *
23  HR
24  15            Integer divide to round
25  /               to nearest ¼ hour
26  GTO IND       Go to tone routine, ac-
                    cording to time
27 ◆LBL 03
28  TONE 8        ¾ hour = 3 'bings'
29  SF 25
30  GTO 06        Slow loop slightly
31  TONE 8
32  SF 25
33  GTO 06
34 ◆LBL 01        ¼ hour = 1 'bing'
35  TONE 8
36  SF 25
37  GTO 06
38  AVIEW
39  RTN
40 ◆LBL 02        ½ hour = 2 'bongs'
41  TONE 6          slightly higher in
42  SF 25           pitch than hour
43  GTO 06
44  TONE 6
45 ◆LBL 00        Exact hour
46 ◆LBL 04
47  AVIEW         Display time
48  .END.
```

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 HP41C 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 last 6 significant digits exhibited the most fandomness 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 HP41C 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 (speed). 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.

"RZ", in conjunction with a good random number generator, will mimic the randomize function in the HP85 computer, which also utilizes the internal clock for the seed, though to greater precision.

Example of use demonstrates that the "RZ" subroutine works by using it to generate a seed for a random number generator and using the random number produced to set an alarm that will re-trigger the routine several times. This would simulate "random" occasional use of "RZ".

(see listings, comments and examples on following page).


```
                                                    XEQ "TESTR"
                                                  8:22:20.66 PM 472192199
                                                  8:22:53.18 PM 64939423
                                                  8:24:00.88 PM 51815129
                                                  8:24:13.98 PM 317185809
                                                  8:24:36.38 PM 750232819
                                                  8:25:11.78 PM 272786021
                                                  8:26:12.08 PM 258534251
                                                  8:26:30.08 PM 623293109
                                                  8:27:21.58 PM 526480071
                                                  8:28:09.68 PM 210111833
                                                  8:28:16.18 PM 400366719
                                                  8:28:43.08 PM 908137333
                                                  8:28:56.28 PM 75480291
                                                  8:29:40.28 PM 869002397
                                                  8:30:40.58 PM 854605377
                    Demonstration routine
8:39 PM 10/17
 01*LBL "TESTR"
15  STO 01  0  STO 00

 06*LBL "S"
PWRUP  TIME  FIX 6  CLA
ATIME  "+ "  XEQ "RZ"
STO 02  1.E9  *  FIX 0
ARCL X  PRA  RCL 02
XEQ 00  ABS  CLST  LASTX
HMS  1 E2  /  TIME  HMS+
"††S"  DSE 01  GTO 03
15  STO 01  ADV  ADV

 37*LBL 02
"EQ 00  FIX 6  VIEW X
/S      GTO 02  RTN

 44*LBL 03
XYZALM  PWRDN  RTN

 48*LBL 00
RCL 00                  0.221617
        STO 00  END     0.943024
                        0.881170
                        0.413040
                        0.908307
                        0.925510
                        0.876180
                        0.406250
                        0.223717
                        0.567124
                        0.167271
                        0.210958
                        0.260985
                        0.576152
                        0.831259
```

set up 15 loops and initialize registers

control alarm label
turn on printer
format alpha display to show full precision of time and nine digits of seed. Use seed to create time offset for setting control alarm and rerunning routine.

when done, print 15 random numbers

alarm, turn off printer

random number generator

9821 * .211327 + FRC

```
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