Popular Electronics®

WORLD'S LARGEST SELLING ELECTRONICS MAGAZINE JUNE 1982/$1.25

How to Get Stereo Sound from TV & VCRs

Wireless System Stops Burglars Before They Enter

Add a Status Monitor to Extension Telephones

Big-Computer Performance from a Pocket Computer

ssue:

e Computer System

502 Audio/Video Control Amplifier

Dolby-C Noise Reduction Processor

Popular Electronics
JUNE 1982

BIG COMPUTER PERFORMANCE FROM A POCKET COMPUTER

THE WIDE PRINTOUT

A simple interface and some software added to a Pocket Computer enable it to "talk" to the larger TRS-80 Model I, producing easy-to-read hard copy at low cost

BY CASS AND DAN LEWART

THE handheld computer, programmable in BASIC, brought portable computing power to us in late 1980. Called the TRS-80 "Pocket Computer" from Radio Shack (also Sharp's PC-1211), it has a wide repertoire of instructions and is user friendly. Although an accessory printer is available, it prints only 16 characters per line on narrow adding-machine paper. This is fine for portable use, but is obviously deficient for serious work at home since the format is so difficult to read.

Hardware/software information presented here enables anyone with a TRS-80 Model I with two disk drives and a standard printer to transfer list-

ings from the Pocket Computer to the big machine. The result is hard copy in conventional width that can also be viewed on a video screen. And the cost should be less than $50.

Description. When a program is CSAVEd from the Pocket Computer, instead of being fed into the cassette interface and tape recorder, it outputs to an inexpensive "black box" that plugs into the expansion interface of the TRS-80 Model I. When CSAVEing, the Pocket Computer sends binary pulses representing "tokens" corresponding to the BASIC statements and line numbers. The TRS-80, in turn, reads the characters off the data bus, groups them into tokens, and translates these tokens into line numbers and BASIC statements. The BASIC program can then be formatted, printed, or stored for future use. (Statements could possibly be executed on the Model I if the user can resolve differences between the two BASICs.)

Reading and interpreting the output of the Pocket Computer then becomes a purely software problem. A machine-language program reads the binary pulses off the bus, a BASIC program does the token conversion and formatting, and listings are displayed on the screen. Hard copy can then be produced by a standard printer.

```
10:"Z"PAUSE "DE
   MONSTRATION
   PROGRAM"
20:PAUSE "SPECI
   AL NON-ASCII
    CHARACTERS"
   :PAUSE "π,^,
   √,¥ AND E"
30:FOR I=1TO 10
   :INPUT "HOW
   MANY $?";Z
40:B=Z*3E2:
   PRINT "EQUAL
   S ";B;" ¥":
   NEXT I
```

Fig. 1. A demonstration program printed on the regular Radio Shack/Sharp printer is shown at left. The same program printed on a Paper Tiger IDS440G printer using the circuit described in this article is below.

```
10:"Z"PAUSE "DEMONSTRATION PROGRAM"
20:PAUSE "SPECIAL NON-ASCII CHARACTERS":
  PAUSE "π,^,√,¥ AND E"
30:FOR I=1TO 10:INPUT "HOW MANY $?";Z
40:B=Z*3E2:PRINT "EQUALS ";B;" ¥":NEXT I
```

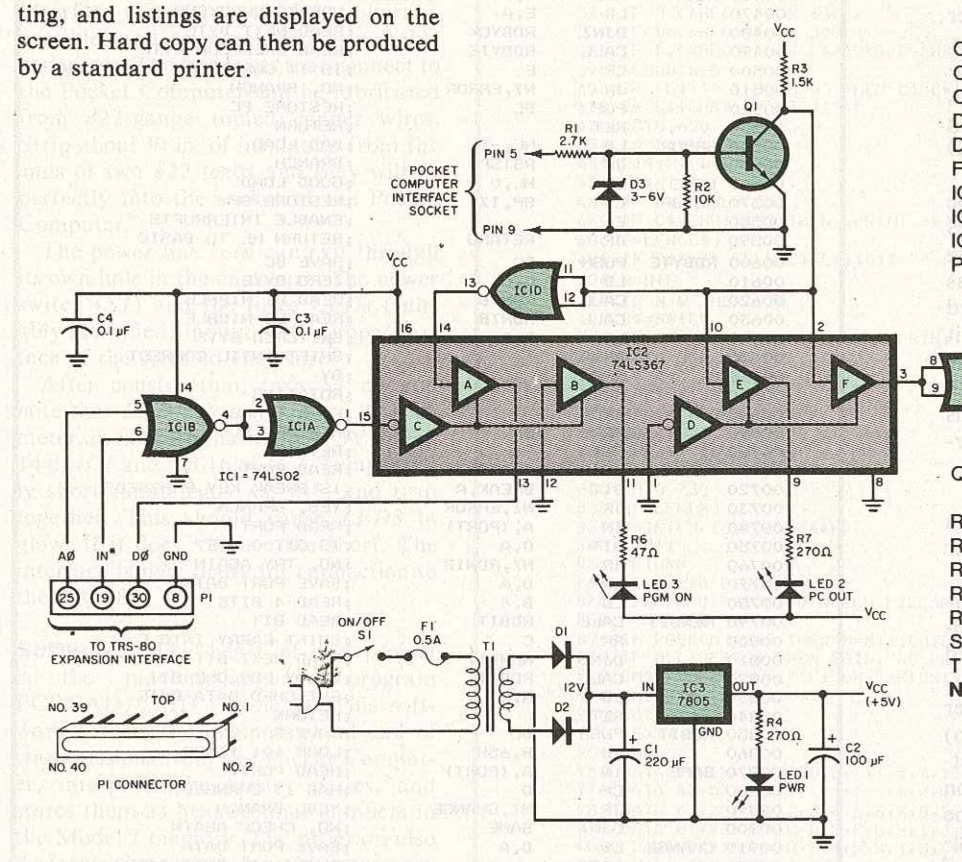

Fig. 2. Schematic of the interface circuit. Included is a regulated power supply.

PARTS LIST

C1—220-µF, 35-V electrolytic
C2—100-µF, 35-V electrolytic
C3, C4,—0.1-µF disc capacitor
D1, D2—50-V, 1-A silicon rectifier
D3—3-to-6-V zener diode
F1—0.5-A fuse with holder
IC1—74LS02 quad NOR gate
IC2—74LS367 hex tri-state buffer
IC3—7805 5-V regulator
P1—2-by-20-pin edge connector Radio Shack (276-1558)
Q1—Npn silicon transistor (Radio Shack 276-2014 or similar)
R1—2.7-kΩ resistor
R2—10-kΩ resistor
R3—1.5-kΩ resistor
R4, R5, R7—270-Ω resistor
R6—47-Ω resistor
S1—Spst switch
T1—12-V, 1-A center-tapped transformer

Note: The following is available from C&R Electronics, P.O. Box 217, Holmdel, NJ 07733: a drilled, glass-epoxy, silk-screened pc board for $12.95, and a 5¼" single-density disk (without DOS) with PCREAD/CMD and three BASIC driver programs for $16.95. Add $1.00 postage and handling. New Jersey residents add 5% sales tax. If you have a single-drive system, enclose a disk with TRSDOS, NEWDOS+ or NEWDOS80 and deduct $2.50. Your disk will then be returned with the PCREAD programs.

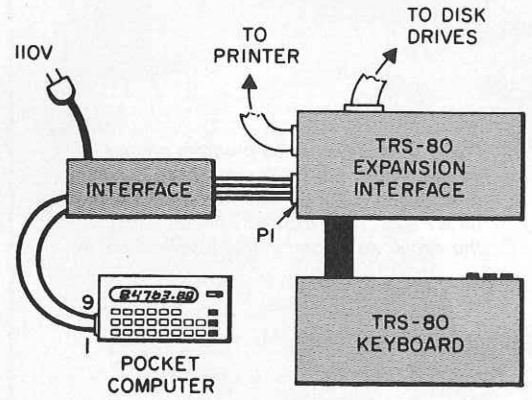

Fig. 3. Use this arrangement to connect the interface circuit between the Pocket Computer and the TRS-80. Keep the four leads between the new circuit and the TRS-80 Expansion Interface as short as possible.

The printer the authors used normally operates in the character (ASCII) mode. However, when a token describing a special non-ASCII or graphics character is encountered, the printer switches from ASCII to the graphics mode, where each matrix dot can be individually addressed. Thus, the printer can create a replica of such a character. If your printer does not provide a graphics mode, then π could be printed as Pi, $\sqrt{\ }$ as Sqr, etc. An example of a listing obtained with a Paper Tiger IDS440G printer is shown in Fig. 1.

The Pocket Computer tokens can be saved on tape or disk with the DOS DUMP command for future listing. The BASIC statements can also be saved as ASCII files for formatting with a word processor. (Using the low-cost circuit with appropriate software allows the Pocket Computer to communicate with the Model I, but not vice versa. This would require a more complicated interface and software.)

Circuit Operation. The circuit shown in Fig. 2 interfaces the Pocket Computer with the TRS-80 Model I.

Data pulses from the Pocket Computer, representing BASIC tokens, are amplified by *Q1*, buffered by *IC1D*, and applied via pin 14 to one three-state buffer within *IC2*. The output of this buffer (pin 13) is applied to the DO (data-zero) line of the TRS-80 via a 40-pin connector or plugged into the TRS-80 Expansion Interface. The A0 address line and the IN* line from the TRS-80 are also coupled via this connector to the inputs of *IC1B*. When these two inputs go low simultaneously, which happens when the TRS-80 issues an IN A (port) command, the output of *IC1B* goes high. This signal is inverted by *IC1A* and passed to pin 15 of *IC2* to "open" gates A and B via gate C. Gate A then passes the Pocket Computer data to the DO bus of the TRS-80. Gate B, in turn, sinks current from *LED3* (PGM ON) which, when glowing, indicates that the TRS-80 is ready to accept data from the Pocket Computer.

Gates E and F of *IC2* are always "open" due to control gate D being kept "on" by the ground at its input. Thus, gate E allows *LED2* (PC OUT) to glow as data flows from the Pocket Computer, while gate F, after inversion by *IC1C*, drives test point 1 (*TP1*).

Power is derived from a conventional full-wave rectifier (*D1* and *D2*). The rectifier output is filtered by *C1*, and regulated to 5 V by *IC3*. An indication that power is applied to the circuit is provided by *LED1* (PWR).

TABLE I—PCREAD/CMD MACHINE-LANGUAGE PROGRAM

```
00100 BREAK   EQU   2            ;BREAK KEY BIT
00110 CORNER  EQU   3C3FH        ;CORNER OF SCREEN
00120 EOF     EQU   0F0H         ;END CODE OF RECORDING
00130 FIRST   EQU   0F100H       ;FIRST BYTE OF BUFFER
00140 NEWDOS  EQU   402DH        ;ENTRY TO NEWDOS
00150 PORT    EQU   00H          ;CASSETTE INPUT PORT
00160 RETARG  EQU   0A9AH        ;RETURN HL TO BASIC
00170 ROW7    EQU   3840H        ;KEYBOARD ROW 7
00180         ORG   0F000H       ;61440 DECIMAL
00190 PCREAD  DI                 ;DISABLE INTERRUPTS
00200         LD    IX,0         ;ZERO IX
00210         ADD   IX,SP        ;SAVE SP IN IX
00220         LD    HL,FIRST-1   ;POINT TO BUFFER-1
00230         CALL  RDBYTE       ;READ CHECKSUM BYTE
00240         LD    E,0          ;ZERO CHECKSUM
00250         CALL  RD8B         ;READ PASSWORD
00260 RD80B   LD    E,0          ;ZERO CHECKSUM
00270         LD    B,10         ;READ 10*8 BYTES
00280 NXT8    CALL  RD8B         ;READ 8 BYTES
00290         DJNZ  NXT8         ;READ NEXT GROUP
00300         JR    RD80B        ;ZERO CHECKSUM
00310 RD8B    PUSH  BC           ;SAVE BC
00320         LD    B,8          ;READ 8 BYTES
00330 RDBYCK  CALL  RDBYTE       ;READ BYTE
00340         LD    (CORNER),A   ;DISPLAY ON SCREEN
00350         INC   HL           ;MOVE BUFFER POINTER
00360         LD    (HL),A       ;PUT BYTE IN BUFFER
00370         CP    EOF          ;IS IT END CODE?
00380         JR    Z,DONE       ;YES, BRANCH
00390         AND   0FH          ;LESS SIGNIFICANT NIBBLE
00400         LD    C,A          ;SAVE IN C
00410         LD    A,D          ;SWITCHED BYTE
00420         AND   0FH          ;MORE SIGNIFICANT NIBBLE
00430         ADD   A,E          ;ADD MSN TO CHECKSUM
00440         JR    NC,NOOV      ;SKIP ON NO OVERFLOW
00450         INC   A            ;INCREMENT CHECKSUM
00460 NOOV    ADD   A,C          ;ADD LSN TO CHECKSUM
00470         LD    E,A          ;UPDATE CHECKSUM
00480         DJNZ  RDBYCK       ;READ NEXT BYTE
00490         CALL  RDBYTE       ;READ CHECKSUM BYTE
00500         CP    E            ;IS IT CORRECT?
00510         JR    NZ,ERROR     ;NO, BRANCH
00520         POP   BC           ;RESTORE BC
00530         RET                ;RETURN
00540 ERROR   LD    HL,-1        ;BAD LOAD
00550         JR    RSTSP        ;BRANCH
00560 DONE    LD    HL,0         ;GOOD LOAD
00570 RSTSP   LD    SP,IX        ;RESTORE SP
00580         EI                 ;ENABLE INTERRUPTS
00590         JP    RETARG       ;RETURN HL TO BASIC
00600 RDBYTE  PUSH  BC           ;SAVE BC
00610         LD    C,0          ;ZERO BYTE
00620         CALL  RDNIB        ;READ HI NIBBLE
00630         CALL  RDNIB        ;READ LO NIBBLE
00640         LD    D,A          ;SWITCHED BYTE
00650         RRCA               ;SHIFT UNTIL CORRECT
00660         RRCA               ;BY
00670         RRCA               ;ROTATING
00680         RRCA               ;FOUR TIMES
00690         POP   BC           ;RESTORE BC
00700         RET                ;RETURN
00710 RDNIB   LD    A,(ROW7)     ;READ PORT
00720         BIT   BREAK,A      ;IS BREAK KEY PRESSED?
00730         JR    NZ,ERROR     ;YES, BRANCH
00740         IN    A,(PORT)     ;READ PORT
00750         BIT   0,A          ;IS BIT 0 OFF?
00760         JR    NZ,RDNIB     ;NO, TRY AGAIN
00770         LD    D,A          ;SAVE PORT DATA
00780         LD    B,4          ;READ 4 BITS
00790 NEWBIT  CALL  RDBIT        ;READ BIT
00800         RR    C            ;SHIFT CARRY INTO C
00810         DJNZ  NEWBIT       ;READ NEXT BIT
00820         CALL  RDBIT        ;DELAY FOR ONE BIT
00830         LD    A,C          ;SWITCHED DATA BYTE
00840         RET                ;RETURN
00850 RDBIT   PUSH  BC           ;SAVE B
00860         LD    B,65H        ;LOOP 101 TIMES
00870 SAME    IN    A,(PORT)     ;READ PORT
00880         CP    D            ;HAS IT CHANGED?
00890         JR    NZ,CHANGE    ;YES, BRANCH
00900         DJNZ  SAME         ;NO, CHECK AGAIN
00910 CHANGE  LD    D,A          ;SAVE PORT DATA
00920         RRCA               ;SHIFT BIT INTO CARRY
00930         POP   BC           ;RESTORE B
00940         RET                ;RETURN
00950         END   NEWDOS       ;BY DANIEL S. LEWART
```


Construction. The circuit can be fabricated on a small piece of perf board using point-to-point wiring, or a small pc board can be made. Use sockets for *IC1* and *IC2*.

The pc board and the small associated power supply can be mounted within almost any desired enclosure. The four leads going to the TRS-80 Expansion Interface (Fig. 3) should be as short as possible, and terminated in a suitable connector. The two leads that connect to the Pocket Computer can be fabricated from #22-gauge tinned copper wires. Strip about 1/4 in. of insulation from the ends of two #22 leads and they will fit perfectly into the socket of the Pocket Computer.

The power line cord can exit through its own hole in the enclosure. The power switch (*S1*) and the three LEDs (suitably identified) mount on the upper surface of the selected enclosure.

After construction, turn *S1* on and note that *LED1* glows. Using a dc voltmeter, make sure that there is 5V on pin 14 of *IC1* and pin 16 of *IC2*. Temporarily short input leads A0, IN*, and GND together. This should cause *LED3* to glow. If it does, remove the short. The interface is now ready for connection to the TRS-80.

Software. The heart of the project is the machine-language program PCREAD/CMD (Table I). This software detects the beginning and end of transmissions from the Pocket Computer, interprets the binary pulses, and stores them as hexadecimal numbers in the Model I memory. The program also performs the timing function necessary to read the 500-baud signals from the Pocket Computer, and sets a blinking display block in the upper-right corner of the video monitor screen to indicate when the data is being read. Detailed

TABLE II—PRINTER DRIVER PROGRAM FOR MX-80

```
100 POKE &H40B2,&HEF:POKE &H40B1,&HFE:CLEAR 2000
110 DEFINT A-Z:DIM D$(255),P$(255):DEFUSR=&HF000
120 CMD"PCREAD/CMD"
130 FOR I=1 TO 11
140     READ A,B
150     FOR J=A TO B
160         IF I<7 THEN READ D$(J):ELSE READ T$:D$(J)=T$+" "
170         NEXT
180     NEXT
190 D$(0)=CHR$(13)
200 D$(18)=CHR$(34)
210 A$=CHR$(27)+CHR$(75)+CHR$(5)+CHR$(255)
220 '23=YEN,25=PI,26=SQR,57=CARET,75=EXP
230 P$(23)=A$+CHR$(168)+CHR$(104)+CHR$(62)+CHR$(104)+CHR$(168)
240 P$(25)=A$+CHR$(130)+CHR$(252)+CHR$(128)+CHR$(252)+CHR$(130)
250 P$(26)=A$+CHR$(8)+CHR$(4)+CHR$(254)+CHR$(128)+CHR$(128)
260 P$(57)=A$+CHR$(24)+CHR$(32)+CHR$(64)+CHR$(32)+CHR$(24)
270 P$(75)=A$+CHR$(254)+CHR$(254)+CHR$(146)+CHR$(146)+CHR$(146)
280 INPUT"HOW MANY CHARACTERS PER LINE";W
290 IF W<11 THEN 280
300 PRINT"1)PC LOAD  2)DISK LOAD  3)LIST  ";
310 INPUT"4)PRINT  5)DISK SAVE  6)EXIT";G
320 ON G GOTO 350,420,470,470,870,950
330 GOTO 300
340 'LOAD FROM PC
350 IF (INP(0) AND 1)=1 THEN 380
360 PRINT"TURN ON POCKET COMPUTER"
370 IF (INP(0) AND 1)=0 THEN 370
380 PRINT"CSAVE PC PROGRAM"
390 IF USR(0) THEN PRINT"BAD LOAD" ELSE PRINT"GOOD LOAD"
400 GOTO 300
410 'LOAD FROM DISK
420 LINEINPUT"LOAD FROM WHICH FILESPEC? ";F$
430 CM$="LOAD "+F$
440 CMD CM$
450 GOTO 300
460 'LIST (OR PRINT)
470 M=&HF100
480 C$="":N=0:K=0
490 FOR I=1 TO 7
500     GOSUB 820
510     IF B>0 THEN C$=D$(16*L+H)+C$
520     NEXT
530 M=M+1
540 CLS:PRINT C$
550 B=PEEK(M):M=M+1
560 IF B<(&HE0) THEN 630
570 IF B=(&HF0) THEN 300
580 IF B>(&HE0) THEN C$=CHR$(B-&HB0) ELSE C$=" "
590 GOSUB 820
600 IF (C$<>" " OR H>0) THEN C$=C$+CHR$(H+&H30) ELSE C$="  "
610 C$=C$+CHR$(L+&H30)+":"
620 GOTO 650
630 C$=D$(B)
640 C1$=P$(B)
650 PRINT C$;                                :'OUTPUT STRING
660 IF G=3 THEN 550
670 IF C$=CHR$(13) THEN LPRINT S$:S$="":N=0:K=0:GOTO 550
680 E=LEN(C$)
690 IF C1$>"" THEN C$=C1$:C1$="":E=1
700 N=N+E
710 IF N>W THEN 750
720 S$=S$+C$
730 IF (C$=":" OR N=4) THEN LPRINT S$;:K=N-4:S$=""
740 GOTO 550
750 IF K=0 THEN 790
760 LPRINT:LPRINT"    ";
770 N=N-K:K=0
780 GOTO 720
790 LPRINT S$:LPRINT"    ";
800 S$=C$:N=4+E:K=0
810 GOTO 730
820 B=PEEK(M)                                :'READ BYTE
830 H=INT(B/16):L=B-16*H
840 M=M+1
850 RETURN
860 'SAVE TO DISK
870 LINEINPUT"SAVE TO WHICH FILESPEC? ";F$
880 M=&HF108
890 IF PEEK(M)<240 THEN M=M+1:GOTO 890
900 IF M>(&HF698) THEN PRINT"NO PROGRAM LOADED":GOTO 300
910 CM$="DUMP "+F$+" F100H,"+RIGHT$(STR$(65536+M),5)+",402DH"
920 CMD CM$
930 GOTO 300
940 'EXIT
950 END
960 DATA 17,29," ",QUOTE,?,!,#,%,YEN,$,PI,SQR,",",;,":"
970 DATA 48,57,(,),>,<,=,+,-,*,/,[
980 DATA 64,75,0,1,2,3,4,5,6,7,8,9,.,EXP
990 DATA 81,93,A,B,C,D,E,F,G,H,I,J,K,L,M
1000 DATA 94,106,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
1010 DATA 130,132,>=,<=,<>
1020 DATA 144,146,TO,STEP,THEN
1030 DATA 160,173,SIN,COS,TAN,ASN,ACS,ATN,EXP,LN,LOG,INT,ABS
1040 DATA         SGN,DEG,DMS
1050 DATA 176,183,RUN,NEW,MEM,LIST,CONT,DEBUG,CSAVE,CLOAD
1060 DATA 192,197,GRAD,PRINT,INPUT,RADIAN,DEGREE,CLEAR
1070 DATA 208,222,IF,FOR,LET,REM,END,NEXT,STOP,GOTO,GOSUB,CHAIN
1080 DATA         PAUSE,BEEP,AREAD,USING,RETURN
```

17	SPACE	100	T
18	QUOTE	101	U
19	?	102	V
20	!	103	W
21	#	104	X
22	%	105	Y
23	YEN	106	Z
24	$	130	>=
25	PI	131	<=
26	SQR	132	<>
27	,	144	TO
28	;	145	STEP
29	:	146	THEN
48	(	160	SIN
49	)	161	COS
50	>	162	TAN
51	<	163	ASN
52	=	164	ACS
53	+	165	ATN
54	−	166	EXP
55	*	167	LN
56	/	168	LOG
57	^	169	INT
64	0	170	ABS
65	1	171	SGN
66	2	172	DEG
67	3	173	DMS
68	4	176	RUN
69	5	177	NEW
70	6	178	MEM
71	7	179	LIST
72	8	180	CONT
73	9	181	DEBUG
74	.	182	CSAVE
75	E	183	CLOAD
81	A	192	GRAD
82	B	193	PRINT
83	C	194	INPUT
84	D	195	RADIAN
85	E	196	DEGREE
86	F	197	CLEAR
87	G	208	IF
88	H	209	FOR
89	I	210	LET
90	J	211	REM
91	K	212	END
92	L	213	NEXT
93	M	214	STOP
94	N	215	GOTO
95	O	216	GOSUB
96	P	217	CHAIN
97	Q	218	PAUSE
98	R	219	BEEP
99	S	220	AREAD
		221	USING
		222	RETURN

Fig. 4. Token equivalents for converting hexadecimal number characters into BASIC statements.

function of the program is shown in the "Remark" column in the listing.

A BASIC driver program (PC-READ/BAS) then takes the hexadecimal numbers characters stored in the Model I computer memory (equivalent to the Pocket Computer program listing), forms them into tokens, and translates the tokens into BASIC statements according to Fig. 4. (This was first derived by Norlin Rober.)

Three versions of a printer driver program were written in disk BASIC for Epson MX-80 with Graftrax, Paper Tiger IDS/440G, and for printers without graphic capabilities. The version for the MX-80 is listed here (Table II). Minor modifications would be required for other graphic printers. These programs are available on diskette (see Parts List). The BASIC programs give the option to specify the maximum number of characters per line. The text will break at whole tokens when the maximum number of characters per line is exceeded (again see Fig. 1).

Checkout and Operation. Connect the Pocket Computer to the interface as shown in Fig. 3. With both units powered up, CSAVE a program and note that *LED2* on the interface blinks. If you have a scope, observe the rectangular pulses at *TP1* as the Pocket Computer outputs in the CSAVE mode.

Now turn everything off and plug the interface *P1* connector into the Model I Expansion Interface as shown in Fig. 3. Turn on the power for the TRS-80 Model I, the interface, and the pocket computer, run a BASIC driver program that calls PCREAD/CMD. When prompted, CSAVE a program from the Pocket Computer. You then have the options of displaying, printing, or saving the program. ◇

A NEW, EFFECTIVE ANTI-BURGLARY SYSTEM

Here's a wireless theft-prevention system that sounds an alarm before a burglar enters the home

BY RAYMOND L. KIRBY

INTRUSION systems to protect one's valuables and life have become increasingly popular as burglaries continue to rise. The theft-prevention system presented here operates in an effective manner using unusual means to thwart thieves.

Firstly, it's a pre-entry deterrent since it is activated by attempted forced-entry noises such as the sound of breaking glass, splintering wood, metal striking metal, etc. At the same time, it ignores such "normal" sounds as talking, telephone ringing, and the doorbell. Secondly, it does not require running of wires and installation of switches, and so on. As a consequence, the system alerts the user to forced-entry efforts *before* the thief can enter the home, and installation is utterly simple.

The pre-entry warning system described in this article features a sound discriminator that can cover a 3000-square-foot area, a 110-dB siren, battery backup in case of power-line failure, the ability to turn on lights, an exit delay, duration control, remote arm/disarm, and compatibility with many other instrusion alert systems. Best of all, this system can be built for under $200.

Circuit Operation. As shown in Fig. 1, the ceramic microphone picks up the entire gamut of sounds within the protected area. The microphone signal is amplified by *IC1A*, which has some frequency rolloff due to *R2* and *C2*. Stage gain is determined by *R4*, which is set during calibration.

After level selection by *R5*, the audio signal is fed to bandpass filter *IC1B*. The feedback network consisting of *C5*, *C6*, *R8*, and *R9* peaks this circuit at approximately 16 kHz. Thus, conventional speech and household noises will not pass the *IC1B* filter—but, the high-frequency (mechanical) noises caused by attempted forced entry will. The filtered audio signal is then applied to timer *IC2A* of Fig. 2.

When the filtered audio signal applied to pin 6 exceeds about 4 V, *IC2A* is activated. The timing circuit consists of *R13*, *R14*, and *C8*, with *R13* permitting a duration adjustment between 10 seconds and two minutes. This range can be shortened or lengthened by changing the value of *R14*.

When pin 5 of *IC2A* goes high, *C9* charges and causes pin 13 of *IC2B* to go high. This, in turn, causes *Q1* to turn on, thus keeping pin 6 of *IC2A* high to prevent self-activation. Diode *D1* prevents rapid discharge of *C9* via pin 5 of *IC2A*. This hold-off lasts approximately five seconds. Since pin 9 of *IC2B* goes low during this time, *LED1* remains off. However, after the brief timing interval, pin 13 goes low and pin 9 goes high. This shuts off *Q1*, arming the circuit, and the high at pin 9 turns on *LED1*, signifying that the system is ready to activate.

Transistor *Q9* is also coupled to pin 6 of *IC2A*. As long as a short exists across the N.C. LOOP terminals, *Q9* is cut off. When the short is removed, *Q9* goes into conduction and shuts down *IC2A*, similar to the action of *Q1*. To prevent false activation under low-voltage conditions, transistors *Q2* and *Q3* of Fig. 3 form a Schmitt trigger. The collector of *Q3* is connected