

Circle 135 on inquiry card.

# **WARBLE ALARM CAR-VAN CLOCK** WITH HEADLIGHT ALARM



COMPLETE KIT \$35.95  
ASSEMBLED \$45.95

- ELAPSED TIMER
- SECONDS DISPLAY SWITCH
- 4 MINUTE SNOOZE ALARM
- SIMPLE AM/PM HOUR-UP
- JUMBO 7-SEG DISPLAY
- 1 TO 10 MINUTE COUNTDOWN TIMER (HOURS DISPLAYS ANGLE WITH CLOCK)
- ROSEWOOD ABS CASE
- QUARTZ CRYSTAL ACCURACY

## **DIGITAL AUTO INSTRUMENTS SEVEN MODELS!**

- #1 TACHOMETER
- #2 WATER TEMP.
- #3 FUEL LEVEL
- #4 SPEEDOMETER
- #5 OIL PRESSURE
- #6 OIL TEMP.
- #7 BATTERY MONITOR



- KIT INCLUDES: CASE & ALL HARDWARE
- PREWIRED W/AMP SENSORS
- ADDRESSABLE SWITCHES
- FEATURES: 10 RANGE LED'S
- 67" W x 3.7" H x 3.0" D

ADD \$10 FOR REGULATED SPEED SENSOR. \$15 FOR SPEED SENSOR ALONE  
KIT: \$49.95. . . . . ASSEMBLED: \$59.95

## **ELECTRONIC 'PENDULUM' CLOCK**



- SWING PENDULUM
- 7" HOURS AND MINUTES DISPLAY
- TIME SET PUSH BUTTONS
- ALARM FEATURE

KIT-UNFINISHED CASE . . . . . \$59.95  
ASSEMBLED-STAINED CASE . . . . . \$69.95

## **QUARTZ DIGITAL AUTO CLOCK OR ELAPSED TIMER!**

- ELAPSED TIMER: HRS, MINS & SECS
- SIMPLE PUSHBUTTON RESET & HOLD TOGGLE SWITCH
- KIT INCLUDES EVERYTHING
- NOTHING ELSE TO BUY! (LESS 10% OFF!)
- INTERVAL: 10 SECONDS BACKOUT
- NON-POLAR INPUT
- 12 OR 24 HR MODE
- CONVERSIONS: 12" = 5" x 10"
- 4 DIGIT VERSION: 10" x 5" x 10"



KIT: \$27.95. . . . . ASSEMBLED: \$37.95

## **NOW WITH ELAPSED TIME!**

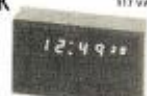


**3 1/2" DIGITAL CLOCK**  
• 4 DIGIT KIT . . . \$49.95 • 4 DIGIT ASSEMBLED . . . \$59.95  
• 6 DIGIT KIT . . . \$69.95 • 6 DIGIT ASSEMBLED . . . \$79.95

117VAC, 12 OR 24 HR MODE . . . . . KIT COMES COMPLETE!  
6 DIGIT VERSION: 27" x 5" x 10" . . . 4 DIGIT VERSION: 10" x 5" x 10"

## **TV-WALL CLOCK**

- 25" VIEWING DISTANCE
- 3" HOURS & MINUTES
- 3" SECONDS
- COMPLETE WITH WOOD CASE



KIT: \$34.95. . . . . ASSEMBLED: \$39.95

## **ECONOMY CAR CLOCK**

- 3" LED MODULE!
- COMPLETE WITH CASE, BRACKET & TIME SET PUSHBUTTONS
- ALARM OPTION



KIT: \$19.95. . . . . ASSEMBLED: \$26.95

## **PENDULUM**

- Give your digital clock a PENDULUM SWINGING 8 TO 12" (24" ADD-A-INPUT)
- SIMPLE HOOK UP TO ANY CLOCK

\$14.95 CASE WITH BRACKET \$3.75

## **MARK FOSKETS' SOLID STATE TIME**

P.O. BOX 2159  
DUBLIN, CALIF. 94866

ORDERS (415) 828-1923



CALIFORNIA RESIDENTS - ADD 6% SALES TAX

BR ea 0 1 d d (Branch Always)

An effective address (ea) is calculated by adding the signed displacement byte (dd) to the program counter. The program counter contains the address of the instruction immediately following the BR, or the address of the BR operation plus 2. The displacement is a signed two's complement value from -128 to +127. Branch conditions are not changed. Note that effective address calculation is identical to that for 6502 relative branches.

Some examples:

dd = \$80 ea = PC + 2 - 128  
dd = \$B1 ea = PC + 2 - 127  
dd = \$FF ea = PC + 2 - 1  
dd = \$00 ea = PC + 2 + 0  
dd = \$01 ea = PC + 2 + 1  
dd = \$7E ea = PC + 2 + 126  
dd = \$7F ea = PC + 2 + 127

Example:  
\$300: 01 50 BR \$352

BNC ea 0 2 d d (Branch if No Carry)

A branch to the effective address is taken only if the carry is clear, otherwise execution resumes as normal with the next instruction. Branch conditions are not changed.

BC ea 0 3 d d (Branch if Carry set)

A branch is effected only if the carry is set. Branch conditions are not changed.

BP ea 0 4 d d (Branch if Plus)

A branch is effected only if the prior "result" for most recently transferred data was positive. Branch conditions are not changed.

Example: (Clear mem from loc A034 to A03F)

15 34 A0	SET	R5, A034	Init pointer.
14 3F A0	SET	R4, A03F	Init limit.
10 00 00	LOOP	R0, 0	
55	ST	@R5	Clear mem byte, incr R5.
24	LD	R4	Compare limit to
05	CPR	R5	pointer.
04 F8	BP	LOOP	Loop until done.

BM ea 0 5 d d (Branch if Minus)

A branch is effected only if the prior "result" was minus (negative, MSB = 1). Branch conditions are not changed.

BZ ea 0 6 d d (Branch if Zero)

A branch is effected only if the prior "result" was zero. Branch conditions are not changed.

BNZ ea 0 7 d d (Branch if NonZero)

A branch is effected only if the prior "result" was nonzero. Branch conditions are not changed.

BM1 ea 0 8 d d (Branch if Minus 1)

A branch is effected only if the prior "result" was minus 1 (\$FFFF hexadecimal). Branch conditions are not changed.

BNM1 ea 0 9 d d (Branch if Not Minus 1)

A branch is effected only if the prior "result" was not minus 1 (\$FFFF hexadecimal). Branch conditions are not changed.

Text continued from page 154

instructions to this implementation of SWEET16. If you use the unassigned op codes \$0E and \$0F, remember that SWEET16 treats these as 2 byte instructions. You may wish to handle the break instruction as a SWEET16 call, saving two bytes of code each time you transfer into SWEET16 mode. Or you may wish to use the SWEET16 BK (Break) operation as a "CHAROUT" call in the interrupt handler. You can perform absolute jumps within SWEET16 by loading the ACC (R0) with the address you wish to jump to (minus 1) and executing a ST R15 instruction.

And as a final thought, the ultimate modification for those who do not use the 6502 processor would be to implement a version of SWEET16 for some other microprocessor design. The idea of a low level interpretive processor can be fruitfully implemented for a number of purposes, and achieves a limited sort of machine independence for the interpretive execution strings. I found this technique most useful for the implementation of much of the software of the Apple II computer; I leave it to readers to explore further possibilities for SWEET16.■

BRK 0 A (Break)

A 6502 BRK (break) instruction is executed. SWEET16 may be reentered non-destructively at SW16D after correcting the stack pointer to its value prior to executing the BRK.

RS 0 B (Return from SWEET16 Subroutine)

RS terminates execution of a SWEET16 subroutine and returns to the SWEET16 calling program which resumes execution (in SWEET16 mode). R12, which is the SWEET16 subroutine return stack pointer, is decremented twice. Branch conditions are not changed.

BS ea 0 C d d Branch to SWEET16 Subroutine

A branch to the effective address (PC + 2 + d) is taken and execution is resumed in SWEET16 mode. The current PC is pushed onto a "SWEET16 subroutine return address" stack whose pointer is R12, and R12 is incremented by 2. The carry is cleared and branch conditions set to indicate the current ACC contents.

Example: (Calling a "memory move" subroutine to move A034-A03B to 3000-3007)

300: 15 34 A0	SET	R5, A034	Init pointer 1.
303: 14 3B A0	SET	R4, A03B	Init limit 1.
306: 16 00 30	SET	R6, 3000	Init pointer 2.
309: 0C 15	BS	MOVE	Call move subroutine.
320: 45	MOVE	LD @R5	Move one
321: 56	ST	@R6	byte.
322: 24	LD	R4	
323: 05	CPR	R5	Test if done.
324: 04 FA	BP	MOVE	
326: 0B	RS		Return.

## **The Best of BYTE, Volume 1**



The volume we have all been waiting for! The answer to those unavailable early issues of BYTE. Best of BYTE, edited by Carl Helmers Jr and David Ahl. This 384 page book is packed with a majority of material from the first 12 issues. Included are 146 pages devoted to "Hardware" and how-to articles ranging from TV displays to joysticks to cassette interfaces, along with a section devoted to kit building which describes seven major kits. "Software and Applications" is the other side of the coin: on-line debuggers to games to a complete small business accounting system is included in this 125 page section. A section on "Theory" examines the how and why behind the circuits and programs. "Opinion" closes the book with a look ahead, as to where this new hobby is heading. It is now available through BITS Inc for only \$11.95 and 50 cents postage.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

The Best of BYTE, Volume 1 Price of Book \$ \_\_\_\_\_

Postage, 50 cents \$ \_\_\_\_\_

Total \$ \_\_\_\_\_

☐ Check enclosed

☐ Bill MC # \_\_\_\_\_ Exp. Date \_\_\_\_\_

☐ Bill BA # \_\_\_\_\_ Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_

In unusual cases, processing may exceed 30 days. All orders must be prepaid.

You may photocopy this page if you wish to leave your BYTE intact.

Circle 12 on inquiry card.

BYTE November 1977 159