

# Vectors and the Challenger 1P

**Vector is one more computer buzzword usually found in conjunction with confusion. This article will try to clear up what vectors are and will show how to use them effectively.**

Mike Bassman  
39-65 52nd Street  
Woodside, New York 11377

Computers have subroutines for every command and for every other necessary function. Computers also need places in memory to look up the address of a particular subroutine. A vector is a place in memory in which the computer finds the address of a subroutine. The vector will consist of two bytes containing any address from \$0000 to \$FFFF in low-byte, high-byte format.

I'll try to clear this up with an example. Let's say you type in a SAVE command. The C1P must know where to go to find the SAVE subroutine. The C1P looks at the SAVE vector, which is at \$220 and \$221 (544 and 545 decimal), and in it finds \$96 and \$FF, which is \$FF96, the address of the SAVE routine.

What good are vectors? They are used if you want to add to BASIC or change it or any of its commands. Vectors have been used to create shorthand (see MICRO 24:25), do a true backspace, and ensure program security. Let's take program security as an example. A good way to prevent copying of tapes is to disable the SAVE command. We want the SAVE command to coldstart BASIC if someone tries to save a program. The SAVE vector must be changed from pointing at the SAVE routine at \$FF96 to the coldstart address at \$BD11. Therefore, the SAVE vector at 544 and 545 must point at

\$BD11. These two poke commands will take care of it:

```
POKE 544,17
POKE 545,189
```

Now, if you type SAVE, the C1P will respond with "MEMORY SIZE?". Although the C1P has vectors for every command, a large portion of them are in ROM rather than in RAM. Since only RAM and not ROM can be changed, only those vectors residing in RAM can be used. Vectors in RAM, their addresses, and the address that they point at initially are listed below.

Vector	Address		Initial Value	
	Hex	Decimal	Hex	Decimal
SAVE	220,221	544,545	FF96	65430
LOAD	21E,21F	542,543	FF8B	65419
CTRL C	21C,21D	540,541	FF9B	65435
OUTPUT	21A,21B	538,539	FF69	65385
INPUT	218,219	536,537	FFBA	65466
WARM	001,002	001,002	A274	41588
START				

The first two vectors, the SAVE vector and the LOAD vector are used whenever the SAVE and LOAD commands are executed. The Control-C vector is somewhat more subtle. For

every line of BASIC executed, BASIC checks the keyboard for a Control-C. If one has been typed, program execution ends. This vector is the one that is used for every line executed. Therefore, we can use this vector, if our function is one that would be executed once for every line, just as we would use a program tracer. The output vector is used each time BASIC wants to type a character. The input vector is used each time BASIC wants to input a character.

Let's try a more involved example. This time, we'll change the cursor from the underline to, say, a tank character. The program will require use of the input vector, because we will have to change the cursor each time a character is inputted. The program will be in the free part of page 2, starting at \$222 (546 decimal). BASIC stores the cursor location at \$200 (513 decimal). This is used as an index from screen location \$D300 (54016 decimal). Since BASIC resets the input vector after carriage return is hit, we will set the input vector to \$222 after every character. The program follows.

Using these techniques, Ohio Scientific C1P users can customize BASIC to their convenience.

```

BASSMAN:VECTORS AND THE CHALLENGER 1P
0222          ORG      $0222
0222 48      PHA      SAVE ACCUMULATOR
0223 8A      TXA      PUT X IN A
0224 48      PHA      AND SAVE
0225 AE 00 02 LDX      $0200  GET CURSOR LOCATION
0228 A9 FA   LDA      #$FA  GET TANK CHARACTER
022A 9D 00 D3 STA      $D300,X  STORE WITH INDEX
022D 68      PLA      GET X FROM STACK
022E AA      TAX      AND PUT INTO X
022F A9 22   LDA      #$22  LOW BYTE--INPUT VECTOR
0231 8D 18 02 STA      $0218  AND SAVE
0234 A9 02   LDA      #$02  HIGH BYTE--INPUT VECTOR
0236 8D 19 02 STA      $0219  AND SAVE
0239 68      PLA      GET A
023A 4C BA FF JMP      $FFBA  AND RETURN

```

MICRO