

	<b>Application Note</b>
	Product: 80C52 Based Products
Controlling 80C52 Port 1 Bits with Assembly Language Calls	Date: 5/26/98
<p><b>Introduction:</b> Although the Port 1 bits can be controlled directly through BASIC using the PORT1 command, sometimes its easier to use assembly language calls. This method allows the individual bits to be changed instead of the whole port.</p>	
<p><b>Background:</b> The following table indicates the addresses for the Port 1 bits:  P1.0 = 90h P1.4 = 94h P1.1 = 91h P1.5 = 95h  P1.2 = 92h P1.6 = 96h P1.3 = 93h P1.7 = 97h</p> <p>If only a few of the bits are being controlled then the easiest way to implement this is by storing the machine code using data statements. The code to set, clear, or toggle a bit takes three bytes of memory (command, address, return).</p> <p>The machine code commands are used as follows: <b>NOTE:</b> Assignments must begin with an integer, otherwise the processor will interpret them as a variable.  0D2h = set bit  0C2h = clear bit  0B2h = toggle bit (if set, clear it or if cleared, set it)</p>	
<p><b>Program Listing:</b> The sample program below shows how the commands are stored and then called:</p> <pre> 10 DATA 0D2h, 94h, 22h : REM code to set bit 3 of port 1 20 DATA 0C2h, 94h, 22h : REM code to clear bit 3 of port 1 30 DATA 0B2h, 90h, 22h : REM code to toggle bit 0 of port 1 40 FOR ADR = 4000h to 4008h : REM store the codes - this must be a free area of RAM! 50 READ DT 60 XBY(ADR) = DT 70 NEXT ADR 80 REM all commands are now stored in RAM 200 CALL 4000h : REM set bit 3 of port 1 210 CALL 4003h : REM clear bit 3 of port 1 220 CALL 4006h : REM toggle bit 0 of port 1 230 GOTO 200 : REM continuous loop through the three commands </pre>	