

	Application Note																		
	Product: 80C52 BASIC																		
Reading 80C52 Port 3 Bits	Date: 7/28/99																		
<p>Introduction: While BASIC-52 allows you to access Port 1 bits directly through BASIC, it does not allow direct access of the Port 3 bits available on many of the Micromint boards and modules</p>																			
<p>Background: Port 3 is an 8 bit bi-directional I/O port with internal pullups.</p> <table> <tr> <th><u>Port Pin</u></th><th><u>Alternate Function</u></th></tr> <tr> <td>P3.0</td><td>RXD (serial input port)</td></tr> <tr> <td>P3.1</td><td>TXD (serial output port)</td></tr> <tr> <td>P3.2</td><td>*INT0 (external interrupt 0)</td></tr> <tr> <td>P3.3</td><td>*INT1 (external interrupt 1)</td></tr> <tr> <td>P3.4</td><td>T0 (timer 0 external input)</td></tr> <tr> <td>P3.5</td><td>T1 (timer 1 external input)</td></tr> <tr> <td>P3.6</td><td>*WR (external Data Memory write strobe)</td></tr> <tr> <td>P3.7</td><td>RD (external Data Memory read strobe)</td></tr> </table>		<u>Port Pin</u>	<u>Alternate Function</u>	P3.0	RXD (serial input port)	P3.1	TXD (serial output port)	P3.2	*INT0 (external interrupt 0)	P3.3	*INT1 (external interrupt 1)	P3.4	T0 (timer 0 external input)	P3.5	T1 (timer 1 external input)	P3.6	*WR (external Data Memory write strobe)	P3.7	RD (external Data Memory read strobe)
<u>Port Pin</u>	<u>Alternate Function</u>																		
P3.0	RXD (serial input port)																		
P3.1	TXD (serial output port)																		
P3.2	*INT0 (external interrupt 0)																		
P3.3	*INT1 (external interrupt 1)																		
P3.4	T0 (timer 0 external input)																		
P3.5	T1 (timer 1 external input)																		
P3.6	*WR (external Data Memory write strobe)																		
P3.7	RD (external Data Memory read strobe)																		
<p>Program Listing:</p> <p>Example 1: Waiting for a bit to be set or cleared</p> <p>BASIC-52 program</p> <pre> . . . CALL xxxxh Assembly routine xxxxh Label1: JB P3.x, Label1 ;Loop until bit is set RET ;Return to BASIC program or xxxxh Label1: JNB P3.x,Label1 ;Loop until bit is cleared RET ;Return to BASIC program </pre> <p>Example 2: Return the state of a bit to BASIC-52</p> <p>BASIC-52 program</p> <pre> . . . CALL xxxxh Assembly routine Xxxxh Label1: JB P3.x, B_SET ;Test the bit, jump if set B_CLR: MOV 20h, #0 ;Store 0 if cleared RET B_SET: MOV 20h, #1 ;Store 1 if set RET </pre> <p>Once you are back in the BASIC program use the following line to read what was stored:</p> <pre>IF DBY(20h)=0 THEN PRINT "P3.x = Low" ELSE PRINT "P3.x = High"</pre>																			